SLA Management for Big Data Analytical Applications in Clouds: A Taxonomy Study

XUEZHI ZENG, Australian National University, Australia
SAURABH GARG, University of Tasmania, Australia
MUTAZ BARIKA, University of Tasmania, Australia
ALBERT Y. ZOMAYA, University of Sydney, Australia
LIZHE WANG, China University of Geoscience (Wuhan), China
MASSIMO VILLARI, University of Messina, Italy
DAN CHEN, Wuhan University, China
RAJIV RANJAN, China University of Geoscience (Wuhan, China) and Newcastle University, UK

Recent years have witnessed the booming of big data analytical applications (BDAAs). This trend provides unrivalled opportunities to reveal the latent patterns and correlations embedded in the data thus productive decisions may be made. This was previously a grand challenge due to the notoriously high dimensionality and scale of big data while the quality of service (QoS) offered by providers is the first priority. As BDAAs are routinely deployed on Clouds with great complexities & uncertainties, it is a critical task to manage the service level agreements (SLAs) thus a high QoS can then be guaranteed. This study performs a systematic literature review (SLR) of the state-of-the-art of SLA-specific management for Cloud-hosted BDAAs. The review surveys the challenges and contemporary approaches along this direction centering on SLA. A research taxonomy is proposed to formulate the results of the systematic literature review. A new conceptual SLA model is defined and a multi-dimensional categorization scheme is proposed on its basis to apply the SLA metrics for 1) an in-depth understanding of managing SLAs and 2) the motivation of trends for future research.

CCS Concepts: • **Computer systems organization** → **Embedded systems**; *Redundancy*; Robotics; • **Networks** → Network reliability.

Additional Key Words and Phrases: Big Data; Big Data Analytics Application; Service Level Agreement; Service Layer; SLA Metrics; SLA

ACM Reference Format:
Xuezhi Zeng, Saurabh Garg, Mutaz Barika, Albert Y. Zomaya, Lizhe Wang, Massimo Villari, Dan Chen, and Rajiv Ranjan. 2019. SLA Management for Big Data Analytical Applications in Clouds: A Taxonomy Study. 1, 1 (November 2019), 35 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

Authors' addresses: Xuezhi Zeng, Australian National University, Research School of Computer Science, Australia; Saurabh Garg, University of Tasmania, School of TED, Australia; Mutaz Barika, University of Tasmania, School of TED, Australia; Albert Y. Zomaya, University of Sydney, School of IT, Australia; Lizhe Wang, China University of Geoscience (Wuhan), School of Computer Science, China; Massimo Villari, University of Messina, Italy; Dan Chen, Wuhan University, School of Computer Science, China; Rajiv Ranjan, China University of Geoscience (Wuhan, China) and Newcastle University, UK.

XXXX-XXXX/2019/11-ART $15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 Introduction

Recent years have witnessed the booming of BDAAs in Clouds [1, 2, 13, 19, 89, 135, 143]. For example, Google utilizes Google BigQuery [5] to offer inventory management system [2], an abundant, highly scalable, low cost and pay-as-you-go Cloud-hosted BDAA to make inventory management productive and efficient. Amazon provides natural language processing-based BDAA in Cloud that identifies the language of voluminous texts, extracts vital entities such as people, organizations, locations or events, and analyze sentiments in texts using Amazon Comprehend [1]. Salesforce builds their social media monitoring service in their marketing Cloud [3] that can collect social media data in close to real-time and run in it through their underpinning big data technologies and algorithms to produce insights such as near-instant feedback on the effectiveness of new marketing campaigns or alerts about emerging problems with products. These applications offer organizations the capabilities of constructing valuable information and extracting actionable insight for enhancing the evidence-based decision-making process. Many leading providers such as Google and Amazon provision such analytics capabilities in the form of service to customers in a pay-per-use economic model.

In today's competitive world, the potential business values of these applications depend a lot on the quality of service (QoS) offered by providers. Hence, to gain competitive advantages, providers should focus on the needs of their customers and respond proactively to their marketing strategies, not only to build and raise customers awareness of their services but also meet customers' best expectations for service quality. That is to say, providers must provide the required and promised services to their customers, and these services must achieve the requirements of users (ex. availability, elasticity and scalability).

Given these circumstances, it is very important and necessary for efficient methods to manage and guarantee the QoS promised. Service Level Agreement (SLA) represents a formal contract among service providers and customers, which captures agreements in the sense of QoS. SLAs play an integral role in governing the relationships between providers and customers in the context of Cloud-hosted BDAAs. Furthermore, SLAs can be considered as a strong differentiator, which allows service providers to provide various levels of guarantees for services offered to customers as well as to distinguish itself from competitors. Therefore, it has become a critical task to manage the SLAs thus a high QoS of Cloud-hosted BDAAs may then be guaranteed.

Existing surveys focus on SLAs management in grid computing (for Sim [115], Sim [114] and Wieder et al. [132]) or Cloud computing (for Mohamadi et al. [81], Faniyi et al. [33], Hussain et al. [49] and Whaiduzzaman et al. [131]). Internet of Things (IoT) emerges with the recent advancements in computing. Mubeen et al. [82] investigated the existing work on SLAs management for IoT-based applications in Clouds. To the best of our knowledge, there exists only one preliminary review with a simple taxonomy of SLA management for Cloud computing and Cloud-based BDAAs [99], and it does not suffice in any in-depth understanding of managing SLAs or the trends for future research in this area. SLA-specific management for BDAAs in Clouds has largely been ignored.

To bridge gap in this field and spot out the trends for future work, this study performs a systematic literature review (SLR) of the state-of-the-art of SLA-specific management for Cloud-hosted BDAAs. The review mainly concerns the requirements and characteristics across Cloud computing stacks. In particular, a taxonomy-based study is emphasized for the following reasons:

- The existing works on SLA management for Cloud-hosted BDAAs manifest a wide range of thematic perspectives (e.g., techniques used, Cloud models deployed, and layers considered). Further, in each different perspective, various subcategories have been discussed. Take the technique perspective as an example, researchers proposed multiple techniques to address SLA management for Cloud-hosted BDAAs (e.g., simulation and machine learning). It is important to form a hierarchy of these categories for a comprehensive understanding.

- Taxonomy is a scientific tool that is able to provide a global and universal categorization addressing the above needs and continues effectively to accommodate new knowledge when applicable. A well-developed taxonomy of SLA management for Cloud-hosted BDAAs aims to support researchers and practitioners from academia and industry by organizing SLA-specific management concepts and terminologies for Cloud-hosted BDAAs. It provides researchers with a scientific tool to focus on all the aspects bridging research gaps.
- There exists no taxonomy enabling an extensive review of SLA management for Cloud-hosted BDAAs. This is evidenced from the aforementioned brief of existing survey works (more details can be seen in section 2). This study formulates a taxonomy categorize the existing research works from multiple perspectives and then to enable readers a better understanding of the state-of-the-art.

This paper aims not only to present researchers an outlook on SLA-specific management for BDAAs in Clouds, but also to give new insights through a global thematic taxonomy in this research area. The main contributions of this survey include:

- A systematic literature review of SLA Management for Cloud-hosted BDAAs with build-up thematic taxonomy covering six core dimensions including actors, Service layers, techniques, Cloud service and deployment models, SLA metrics and conceptualization;
- A unified SLA model for Cloud-hosted BDAAs from a layer-based perspective to link different types of SLAs in a vertical motion;
- A multi-dimensional categorization scheme regarding SLA metrics dedicated for Cloud-hosted BDAAs, which allows systematically categorization of both common metrics and niche metrics for each layer with respect to requirements of BDDAs; An SLA template is provided for a representative Cloud-hosted BDAA to aid understanding SLAs conversation across its different layers.
- Identification of open issues and future directions of Cloud-hosted BDAA based on the systematic literature review.

The rest of this paper is organized as follows. Section 2 discusses the related works and our motivations to construct a taxonomy-based survey. In Section 3 discusses how the SLR applies to the research field of SLA management for Cloud-hosted BDAAs and proposes a novel thematic taxonomy. Section 4 presents the details of the review results and discusses the findings according to the taxonomy. Section 5 presents a dedicated conceptual SLA model and proposes a multi-dimensional categorization scheme of SLA metrics for BDAAs in Clouds and give an illustrative example of SLAs template for a real Cloud-hosted BDAA. We conclude the paper with open issues and future directions in Section 6.

2 Related Work

To figure-out the considerable difference between our survey and the previous studies in the literature, we present in this section the related research works that have been done by others and highlight their limitations.

A few previous studies focused on the literature review of SLA management in the context of Cloud environment. Mohamadi et al. [81] conducted a very preliminary review of SLA management approaches and compared them with respect to improved parameters, implementation/simulation and its environment, and workload/application. Faniyi et al. [33] surveyed the research landscape of SLA-based Cloud systematically with the focus on specific phase of SLA life cycle (i.e. resource allocation phase) and outlining consequences on the others. From Cloud service provider perspective with small to medium-sized enterprise level. Hussain et al. [49] presented a comprehensive overview of existing approaches of SLA management in Clouds and highlighted the features and limitations of these approaches to tackle the issue of creating a viable SLAs in Cloud computing from the viewpoint of service provider's. While from another perspective, Whaiduzzaman et al. [131] focused on

SLA-based service provisioning techniques and methods that assist in evaluating Cloud services provisioned with regards to user-specific requirements and cost.

With the recent advancements in computing, internet of thing (IoT) has been introduced as an emerging and promising technology. Thus, Saad Mubeen et al. [82] conducted a survey to investigate the existing research on SLAs management for IoT-based applications in Clouds. This survey used a systematic mapping study for the purpose of identifying the results of the published research works that are related to SLA management in IoT context.

Existing surveys either focus on SLA management in Cloud environment or IoT environment, which are not specific and sufficient for SLA management in the context of Cloud-hosted BDAAs. To the best of our knowledge, there is only one research work has been done on SLA management for BDAAs in Clouds. Sahal et al. [99] conducted a preliminary survey and divided SLA management into two types, which are SLA management for Cloud computing and SLA management for Cloud-hosted BDAAs. Regarding to latter type, SLA management approaches are categorized into two groups comprising MapReduce scheduler and Cloud Layer, which we argue that this categorization is oversimplified and fails to give a holistic landscape towards SLA management for BDAAs in Clouds.

From the above existing research works, it is clearly seen that a taxonomy-based survey on SLA management for Cloud-hosted BDAAs is in its infancy. Therefore, we conduct a taxonomic survey in this field by using a systematic literature review (SLR) method to fill this research gap. Our work differentiate existing research works in multiple dimensions: (i) we propose a novel thematic taxonomy that covers six core dimensions including actors, Service layers, techniques, Cloud service and deployment models, SLA metrics and conceptualization; (ii) we design a cross-layer SLA model for Cloud-hosted BDAAs (CL-SLAfBDAAs) to provide a unified and structured way to understand SLAs at different layers with different attributes and their strong dependencies relationship. (iii) we propose a general categorization scheme of SLA metrics consisting of common metrics and niche metrics for each different layer of SLA, which fully embodies the characteristics of Cloud-hosted BDAAs; (iv) we elaborate our proposed model and SLA metrics categorization by giving an example of SLA template for a real Cloud-hosted BDAA. Our work not only provides a comprehensive review of the state-of-the-art landscape, but also finds insights into understanding the research themes/patterns in this field.

3 Systematic Literature Review Of SLA Management in Cloud-Hosted BDAAs

The field of SLA management is broad. It involves several phases that depicted into SLA life cycle [33] [49] to clarify expectations and responsibilities, and streamline communications among the parities involved in the agreement. In [33], five phases of the SLA life cycle are described. The first phase is SLA negotiation to define and agree on given service terms and levels. The second phase is SLA establishment/deployment to implement and deliver the service in accordance to the agreed SLA. The third phase is to SLA monitoring to observe and monitor the service after being deployed and under its execution. The fourth phase is violation management to detect and manage SLA violations. The fifth phase is SLA reporting and termination to provide detailed reports for audit activities that happened during service provisioning, and provide a method to terminate the service at the end of SLA agreement or in case of violations as defined in the agreed SLA. From the aforementioned SLA life cycle and after studying the characteristics of Cloud-based BDAAs, we identify the following requirements for adopting SLA management for Cloud-based BDAAs:

- Cloud provider and customer responsibilities. In SLA management, different parities are involved. Thus, each party role, obligation and penalty in the context of Cloud-based BDAAs should be stated.
- SLA description and commitments. During the establishment of agreement, the service terms and levels, and QoS commitments are defined. This includes SLA metrics that will be specified and monitored to ensure that SLA agreement is respected.

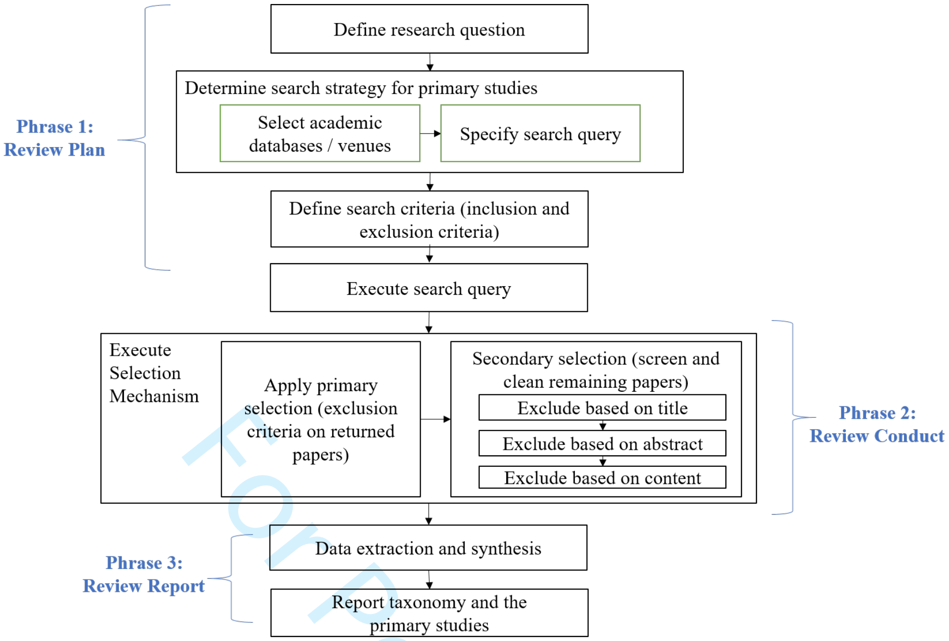
Phrase 1:
Review Plan
Define research question
Determine search strategy for primary studies
Select academic databases / venues
Specify search query
Define search criteria (inclusion and exclusion criteria)
Execute search query
Execute Selection Mechanism
Apply primary selection (exclusion criteria on returned papers)
Secondary selection (screen and clean remaining papers)
Exclude based on title
Exclude based on abstract
Exclude based on content
Phrase 2:
Review Conduct
Phrase 3:
Review Report
Data extraction and synthesis
Report taxonomy and the primary studies

Fig. 1. Systematic literature methodology applied on the survey of SLA management for Cloud-hosted BDAAs

- SLA enforcement. During the deployment of BDAAs in Cloud, different models of Cloud infrastructure and different layers should be considered since the measures and requirements of QoS defined in SLAs differ With each model and layer.
- SLA monitoring and management. After the agreed SLA parameters and metrices are defined, different approaches and techniques to be taken to meet SLA and service providers' procedures to be invoked in the event that SLA guarantees are not fulfilled. Also, policies for applying compensation and penalty are taken place in case of un-fulfilment of SLA terms.

To meet the above requirements, we deliberately intended to cover multiple categories and subcategories in order to acquire a comprehensive understanding of SLA management for Cloud-based BDAAs. To achieve that, we need a a taxonomy-based study by leveraging systematic literature review (SLR) method that provides rigorous way of reviewing the landscape towards SLA-specific management for BDAAs in Cloud to develop comprehensive taxonomy as a key result of this survey work. Our study in this paper uses a SLR methodology proposed by Kitchenham et al. [58] that is an objective, transparent and reproducible methodological method of reviewing extant literature to answer and deduce particular research question(s) in such a way that is unprejudiced. Specifically, our SLR consists of three main phrases as shown in Figure 1.

The first phase is planning for review, where we define our research questions that will be considered and tackled in this study and determine the strategy of search being used for primary studies with lists of inclusion and exclusion criteria. After the executing search query on the selected database sources, the relevant papers obtained will be input for the next phase. The second phase is about conducting the review by applying the selection mechanism including primary selection (exclusion and inclusion criteria) on the obtained papers to get all relevant papers, and the secondary selection for performing a further evaluation for each remaining paper to get the most relevant papers. In the final phase, we further analyze the remaining papers to report a thematic taxonomy of SLA management for BDAAs in Clouds and review these papers based on this taxonomy.

Table 1. Academic database sources

Source	URL
IEEE Explore Digital Library	http://ieeexplore.ieee.org
ACM Digital Library	http://portal.acm.org
Springer	http://springerlink.com
Science Direct	http://sciencedirect.com
Web of Science	http://webofknowledge.com
Google Scholar	http://scholar.google.com

3.1 Research Questions

This study aimed to portray the research landscape in SLA management for BDAAs in Clouds by addressing the following research questions:

- *RQ1*: What are the actors involved in making conversations and engineering SLAs in the context of Cloud-hosted BDAAs?
- *RQ2*: What is the status of addressing SLA management for Cloud-hosted BDAAs from different service layers (i.e. BDSaaS, BDPaaS and CIaaS) and Cloud deployment models (i.e. private, public, hybrid and community Cloud)?
- *RQ3*:What are the techniques applied to address SLA management for Cloud-hosted BDAAs?
- *RQ4*: What SLA metrics are of interest to stakeholders and being discussed in the context of Cloud-hosted BDAAs?
- *RQ5*: To what extent the SLA model for Cloud-hosted BDAAs is conceptualized?

3.2 Search and Selection Strategy for Primary Studies

3.2.1 Selection of Academic Databases and Search String To search for research publications in the areas of computer science, there are well-known databases that are being used as primary sources for these publications [58]. For our study, the academic databases selected are shown in Table 1. These databases provide advanced search options with a set of Boolean functions to make concise search based on certain fields such as abstract, title and keywords, which return the most relevant results in comparison to search all fields.

We then construct our search string that will be executed over the aforementioned databases to search relevant publications. To provide comprehensive coverage of the relevant research works in the literature and state-of-the-art studies, we need to select keywords carefully. Thus, we consider the terms "service level agreement", "Quality of Service", "big data" , "big data analytics", "big data analysis" and "big data analytics application" as the primary keywords along with a range of related abbreviations, plural or synonyms, namely "service level agreements", "service-level agreements", "SLA", "SLAs", "SLM", "SLA management", "QoS", "BDA" and "BDAA". Since a typical BDAA comprises data ingestion, data storage, data processing and data analysis framework, we also consider "MapReduce", "batch processing", "batch computing", "stream processing", "stream computing", "data ingestion" and "NoSQL" as additional keywords. To join the primary keywords and the additional keywords with their synonyms in the search string, Boolean functions (AND and OR) are used. Moreover, to make sure that our search string returns as many relevant studies as possible in the selected databases, we conduct several tests. As a result, we select the following search string:

(("service level agreement" OR "service-level agreement" OR "SLA") OR ("service level agreements" OR "SLAs") OR ("SLA management") OR ("service level management" OR "SLM") OR ("SLA conformance") OR ("quality of service" OR "QoS")) AND (("big data") OR ("big data analytics") OR ("big data analysis") OR ("big data analytical application" OR "BDAA") OR ("big data analytical applications" OR "BDAAs") OR ("big data analytics " OR "BDA") OR "MapReduce" OR ("batch processing") OR ("batch computing") OR ("stream processing") OR ("stream computing") OR "NoSQL" OR "ingestion"))

Table 2. List of Inclusion Criteria

Criterion	Description
The type of study is peer-reviewed	We chose peer-reviewed publications including conference/workshop and journal papers, and peer-reviewed book chapters.
The writing language for the study is English	We restrain the language to English because some databases such as Springer returns publications in another language like German
The publication year for the study is published from 2010 to 2018	We search all publications in accordance to our search string that have been published in the databases between 2010 and 2018.

Table 3. List of Exclusion Criteria

Criterion	Description
The focus of study is not SLA management	We consider the publications in the field of SLA management for Cloud-hosted BDAAs.
Abstracts and publications that are not pass through refereeing process	We exclude the study with only abstract, not in the form of full paper and is not peer reviewed.
Duplicate publication	We remove duplication for the same study found in different databases

3.2.2 Search Criteria and Selection Mechanism To evaluate the publications that will be obtained after applying the search string in next phase, we need to define the search criteria that being applied. Table 2 and Table 3 show inclusion and exclusion criteria that are performed in our SLR.

After executing the search string and applying inclusion and exclusion criteria on the obtained relevant publications in the field of SLA management for Cloud-hosted BDAAs, we get the initial result of 1098 papers. Then, we conduct the stringent selection (secondary selection) on them based their titles, abstracts and contents using the following rule. We strictly select publications that consider SLA management specifically in the context of BDAAs in Clouds and exclude those publications that only discuss SLAs in general Cloud computing environment. This is because, our focus in this paper is given to the evolutionary stage of SLAs for Cloud-hosted BDAAs rather than the evolutionary stage of SLAs for Cloud Computing. Moreover, SLA management in Cloud Computing is a well-explored area with lots of papers. However, SLA management for Cloud-hosted BDAAs is still a young research area, which demands more necessity and urgencies study on. At the end of this phase, we get 109 papers that will be systematically reviewed in this study.

3.3 Data Extraction and Synthesis

The quality assessment strategy used in this study is subjective analysis, where we evaluated the collected papers from primary studies to assess their relevance to the landscape of the survey. From this analysis, the thematic taxonomy of SLA management for BDAAs in Clouds has been emerged (see Figure 2). The following are the detailed descriptions for the proposed taxonomy elements:

- Actors – This element considers the different actors (service providers, consumers and Cloud end users) involved in Cloud-hosted BDAAs. Service providers provide the consumers resources that can be provisioned and metered on demand including big data platform resources and Cloud infrastructure. In particular, it could further classified into BDSaaS, BDPaaS and CIaaS providers. These providers care more about the efficient resource utilization, energy efficiency, profit maximization, cost reduction, and performance enhancement. The service consumers are those actors that utilize services offered by service providers and are liable for their resource consumption's, where they are more concerned about the budget, pricing, and customer satisfaction. The Cloud end users (or for short end users) are those actors that use the applications or services offered by service customers. They are more interested in QoS and SLA constraints such as service quality, performance and response time.
- Service Layers – This element examines SLA management from various levels of abstraction including CIaaS, BDPaaS and BDSaas. At CIaaS layer, SLA management takes care of guaranteeing SLA requirements on virtualized resources such as VMs, storage or network. At BDPaaS layer, such management guarantee SLA requirements for different big data frameworks including data ingestion, data processing, data analysis

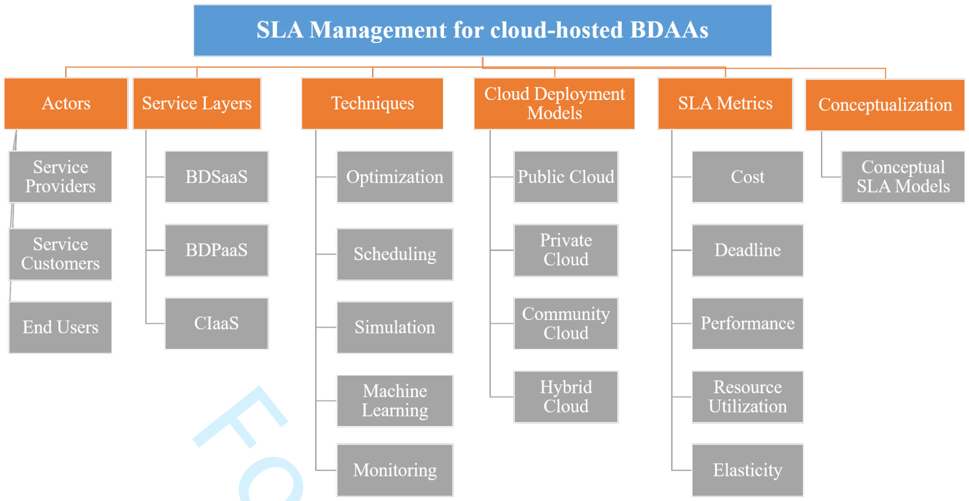

Fig. 2. Taxonomy of SLA management for Cloud-hosted BDAAs

and data storage), and at BDSaaS, such management cares about guaranteeing user-specific application requirements.

- Techniques – In the context of Cloud-hosted BDAAs, one or more techniques (such as optimization-based, scheduling-based and simulation-based) can be included into SLA management to guarantee SLA requirements at specific or across service layers. A technique is a method that aims to address SLA management for Cloud-hosted BDAAs.
- Cloud Deployment Models – When proposing a new SLA management, the activities like implementation, deployment, validation and evaluation come to the picture in order to access the validity and practicality of such management in Cloud infrastructure. Thus, these activities need to be carried-out in private, public, community or/and hybrid Cloud deployment model, where each one of them has its own requirements and challenges.
- SLA Metrics – This element considers SLA items that are defined as quantitative targets in the contract and they must be maintained by the service provided. Measuring these items/metrics (such as cost, deadline and performance) is critical for SLA management to avoid any SLA violation.
- Conceptualization – This element examines SLA management from conceptualization perspective instead of concrete techniques. It mainly consists of designing conceptual SLA models that offer researchers a fundamental and clear way to describe actors, activities and entities involved in a Cloud-hosted BDAA scenario and understand the context of SLA including the conversation and relationship between providers and customers.

4 Review Results in Thematic Taxonomy

4.1 Actors (In response to *RQ1*)

4.1.1 Providers In terms of providers' profit maximization, the authors [29] provide an SLA-based PaaS architecture that can support Cloud-hosted BDAAs. In their paper, a disperse optimization policy is proposed, which aims at maximizing providers' profit and considers to pay penalties incurred when SLA are unsatisfied. Then, the proposed optimization policy is applied to Cloud-hosted BDAAs (e.g., MapReduce applications). In paper [145], the authors designed and implemented automated and elastic resource scheduling algorithms with the objective of profit optimization. Their algorithms can deliver BDAAs to users and optimize profits of platforms while guaranteeing SLAs for query requests in terms of deadlines and budgets and allowing prompt responses with manageable financial costs.

Unlike the above works, paper [18] focuses on the optimization of energy consumption from providers' perspective. The authors take into account sharing MapReduce-based applications in an environment of Hadoop YARN and introduce an SLA-driven energy-saving scheduling algorithm for them [4]. Job profiling is performed to capture the characteristics of performance for diverse stages of a MapReduce-based BDAA. The obtained characteristics of performance will be considered as input to resource provisioning phrase with the purpose of guaranteeing application's SLA such as the completion deadlines. Their experiments demonstrate that their approach enhances the conformance of SLA in terms of reduced energy consumption and resource expenditure.

4.1.2 Customers The authors in [144] focused on their study on Cloud-hosted databases from the customer perspective and addressed the challenge of SLA-driven provisioning and cost management for them. In their paper, a comprehensive framework is proposed, which can flexibly and dynamically provisioning Cloud-hosted database of BDAAs. According to application-defined policies, their proposed framework can satisfy SLAs in terms of performance requirements, avoid penalties when SLA violations occur and control expenses when allocating computing resources.

4.1.3 End Users The authors in [12] proposed an improved resource revenue optimization model. The model defines the constraint mechanism that describes quality of service (QoS) problems. They sliced the requests of end users, modeled the process of requesting service, evaluated the time of response and processing, and allocated resources based on the specified objective function while considering end users' requirements for QoS in this model. They designed a parallel and distributed algorithm based on the working mechanism of MapReduce to solve their proposed model while guaranteeing end users' needs on QoS as much as possible.

4.2 Service Layers (In response to *RQ2*)

Figure 3 presents the statistics of SLA management works in the reviewed papers by the service abstraction level and their breakdown in each layer. It is observed that most of the reviewed papers (67%) fall into the BDPaaS sector. This demonstrates that BDPaaS is the core part of BDAaaS and attracts more interest from researchers. When drilling down into the BDPaaS layer, we found that the top-ranked framework at this layer is data processing with a percentage of 57%. This is because that distributed data processing technologies such as MapReduce receive lots of attention in academia since 2010. Also, it is seen that the data storage framework is ranked secondly, occupying 8%. This indicates that representative data storage technologies such as NoSQL are of increasing interest by researchers in recent years.

Moreover, from the BDSaaS sector, there are 11% reviewed papers discussing SLA management for general BDAAs, while 5% reviewed papers providing domain-specific BDAAs. Interesting, it is further noted that among these domain-specific BDAAs, healthcare application is of the most interest for researchers with four reviewed papers [25, 83, 100, 145] in total and only one reviewed paper takes banking application as the case study [97]. Besides, in CIaaS sector, it is seen that the computing, storage, and network components share the balanced percentage, which means they receive even attention in academia.

Additionally, we present the works of SLA management for Cloud-hosted BDAAs by layers with the corresponding references in Table 4. It has been seen that the quantity of publications regarding batch processing is much higher than that regarding stream processing. The reason is that typical batch processing paradigm like MapReduce featured by its automatic parallelization and distribution, fault tolerance and simplicity becomes ubiquitous programming framework to parallelize the processing of large dataset, which gained significant interest both industry and academia since its emergence in 2007. However, stream processing such as Spark or Storm has been earning improving attention in the last years due to the emerging need for supporting a real-time

Table 4. Classification of the reviewed papers on SLA Management for BDAAs in Clouds by layers

Service Layer (# of papers)	Category (# of papers)	Sub-category and References
BDSaaS (17)	Applications (17)	• General applications: [8, 11, 61, 73, 74, 80, 103, 107, 108, 110, 112, 146] • Domain-specific applications: [25, 83, 97, 100, 145]
BDPaaS	Data Processing Framework (63)	• Batch processing: [10, 12, 16, 18, 29, 31, 35–39, 41, 43, 44, 50, 54, 57, 60, 63, 66–71, 75, 77–79, 85, 87, 88, 90, 93, 95, 105, 106, 109, 111, 120, 124, 125, 128, 129, 133, 134, 137, 141, 142, 145–147] • Stream processing: [15, 40, 42, 46, 51, 59, 83, 117, 121–123]
	Data Storage Framework (9)	[27, 34, 47, 84, 91, 101, 104, 113, 144]
	Data Analysis Framework (2)	[65, 83]
CIaaS	Computing resources (8)	[22, 23, 53, 55, 72, 76, 102, 104]
	Storage resources (6)	[9, 24, 30, 96, 119, 136]
	Network resources (6)	[48, 86, 94, 138, 139, 148]

or near-real-time processing task. The representative works in each layer will be discussed in the following subsections.

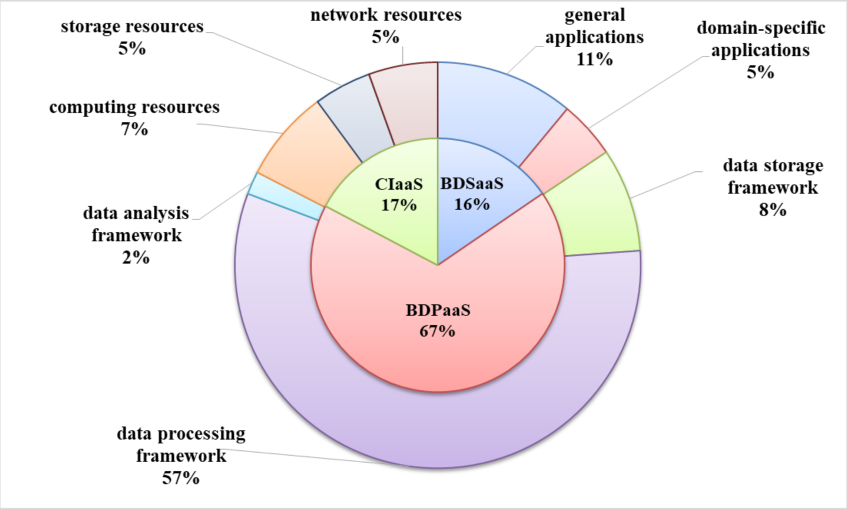

Fig. 3. Statistics of the reviewed papers by layer and their breakdown

4.2.1 *BDSaaS layer* **General applications**: The authors [74] addressed the challenge of situations where numerous job instances in BDAAs should be concurrently deployed at runtime. They introduced DepWare that is a specialized middleware capable of offering an autonomic deployment decision making. They then designed DepPolicy that is a novel language to specify fundamental deployment information. Moreover, an algorithm of deployment decision making is proposed to achieve the optimum deployment for each job instance. Experiments shows that their algorithm of deployment decision making can simultaneously make diverse decisions of deployment at runtime for different job instances. Meanwhile, optimal overall utility is achieved, all given constraints (e.g., cost limit) is satisfied and SLAs (e.g., feasibility, functional correctness, performance, and scalability) is guaranteed.

Domain-specific applications: The authors in [100] focus their work on healthcare BDAA where missing any SLA can generate significant influence on the data analysis of emergency patient thanks to the disease severity. They proposed a computing model for SLA-based healthcare BDAA and implemented a single API to manipulate and analyze both stream-based and batch-based data over Spark platform. They then presented a probabilistic method based on parallel semi-Naive Bayes (PSNB) and designed a modified conjunctive attribute algorithm for dimensionality reduction to improve the accuracy. For those jobs with high priority, they proposed an adaptive job scheduling algorithm to optimize their execution time that satisfies SLA. Experiments results show that their proposed model for SLA-based healthcare BDAA outperforms extant parallel processing models. Also, their proposed PSNB-based approach enhances accuracy compared to the original Naive Bayes algorithm. Differently, the authors [97] focus on how to schedule BDAA workflows in both single Cloud and federated inter-Cloud environments. A workflow consists multiple tasks that need storage, computing and bandwidth resources to transmit and process data. The resources in the workflow need satisfying SLA requirements (e.g., optimizing time to meet deadlines, optimizing cost and managing budgets). A Cloud or inter-Cloud provider could provide resources for executing tasks in the workflow according to specified SLAs criteria. A case study of banking application demonstrates that single or federated Cloud resources are very necessary in terms of executing BDAA workflows.

4.2.2 BDPaaS layer **Data Storage Framework:**

Sakr et. al. [101] considers cost management and SLA-based provisioning for Cloud-hosted NoSQL databases. In their paper, they proposed an end-to-end framework that is represented as middleware residing between the Cloud-hosted databases and consumer applications. The proposed framework aims to flexibly and dynamically provisioning one database function in BDAAs while satisfying their SLA performance requirements (e.g., variability, scalability, elasticity, and performance) according to application-defined policies and avoiding the monetary cost when SLA violations happen as well as controlling the expenses when allocating computing resources. In the context of data security in NoSQL database, Crypt-NoSQL [113] is the first prototype that can encrypt data and execute queries on NoSQL databases while providing high performance. The authors in this paper proposed three different types of models for Crypt-NoSQL and evaluated its performance using Yahoo! Cloud Service Benchmark. Their experiments demonstrate that Crypt-NoSQL is able to efficiently execute queries while guaranteeing SLA requirements (e.g., scalability, high performance). Moreover, they proposed guidance for providers to establish Crypt-NoSQL in the form of a Cloud service and set up pertinent SLA conventions.

Data Processing Framework:

On the one hand, some papers focus on batch-based MapReduce jobs in Cloud. For example, the authors [16] aim at minimizing SLA metrics (e.g., response time) and keeping deadlines set in the pSLA (platform-level SLAs) in this context. First, they developed a so-called grey-box model that can accurately obtain the characteristics of MapReduce behavior. They then proposed a control theory-based framework to satisfy the objectives of SLA. A feed-forward controller was designed and implemented to assure constraints such as service time and improve control response time. The experiments illustrate that the controller is valid in meeting the specified deadlines in the SLAs. Lim et al. [66] propose a novel MapReduce resource manager using constraint programming-based method. In this paper, each MapReduce job is featured by a set of metrics (e.g., the time of earliest start, the time of execution, and deadline) specified in an SLA document. The authors evaluated the performance of their resource manager through an open and discrete event-based simulator where a stream of jobs arrive at intervals. The experiments show that the resource manager can achieve good performance in matchmaking and scheduling MapReduce jobs and give insights into the behavior and performance of system.

On the other hand, some papers focus on stream-based jobs. For example, Rafael et al. [121] take into account simultaneously executing stream workload over shared Cloud infrastructures where each stream is characterized by specific quality of service (QoS) objectives (e.g., throughput, latency) specified in an SLA. They consider classifying customers who submit streams workload into three different classes (Gold/Silver/Bronze/). Each class differentiates by a unique penalty and revenue from providers' side. Their proposed profit model can consider both the cost of provisioning and penalties when the violations of SLA occur. Experiments show that their approach can apply the enforcement of QoS for each application. Paper [46] discusses provisioning resources for stream-based jobs at a granularity of VMs level at runtime. They proposed a novel method to provisioning resources in a cost-efficient way while optimizing the resource usage of VMs (SLA metrics at CIaaS layer).

Moreover, their method is integrated into the Vienna ecosystem at runtime environment for scalable stream processing. The evaluation shows that their method achieves better conformance of SLA by up to 25% and the operation cost reduction up to 36% compared to the extant threshold-based method.

Data Analysis Framework:

The authors [65] developed an extensive model for predictive analysis regarding performance and cost in Cloud. They collected data of resource consumption and placed them in readiness state to enable fast analysis. They stored time series data and various kinds of data regarding performance and events of BDAAs in a layered object store, which can provide the abilities of fast retrieving and pattern analysis. Meanwhile, the authors took into account the data aggregations regarding the interrelation between performance and cost as well as their dynamic tendency over time. Hence, through the application of real-time predictive analysis techniques, the framework achieves an accurate prediction of the current status (i.e., cost and performance) and prospective status. This provides effective support for providers to make decision based on resource configuration regarding the guarantee of SLA requirements.

With regards to analytical capability on prediction accuracy, Lekha et al. [83] focus on developing a real-time stream-based system for the prediction of health status of patients. The system is implemented and deployed on a Cloud-hosted Spark platform which leverages the power of multiple machine learning algorithms. In this scalable system, first, the health information tweeted by users are captured. Then, the proposed system can receive the same health information in real time. Next, the system preprocesses and extracts valid health information from those unstructured stream data and utilizes machine learning algorithms to forecast the health status of users with the purpose of maximizing the accuracy of prediction. By the virtue of the availability of high-quality training datasets of healthcare and the computing power of steam processing of Spark, the process of analyzing huge healthcare samples by applying machine learning techniques becomes significantly more efficient than ever, resulting in enhanced prediction accuracy.

4.2.3 CIaaS layer In terms of Cloud storage, paper [24] addressed various requirements of customers regarding Cloud storage. They first defined a set of realistic and concrete SLA elements. Then, they designed the short-secret-sharing Cloud storage system that applies the defined SLA elements and provides customers with a protected and steady storage service in Cloud. Their proposed system can capture applicable parameters to offer customers with their wanted services while respecting SLAs (e.g., minimal costs). The authors in [136] focus on the scheduling policies of volume request that can shorten the violations of SLA in terms of I/O throughput in Cloud storage systems. To this end, they propose various SLA-driven scheduling policies that consider both I/O throughput and available capacity of backend nodes. The designed scheduling policies can considerably reduce monetary cost of Cloud storage from providers' perspective.

Unlike the above works, Yassine et al. [138] discussed the challenge of transferring multimedia big data over Cloud data centers that are geographically distributed. Since the multimedia volume increases, there is an increasing demand to transfer large datasets across data centers. Therefore, the surplus bandwidth that occurs at different times and for different period in backbone network turns to be inadequate to meet speedily increasing demand for transferring multimedia big data. In this paper, they designed multi-rate Bandwidth-on-Demand (BoD) service to communicate among geo-distributed Cloud datacenters. They also developed a scheduling algorithm that is employed by a BoD broker, which considers transferring multimedia big data requests that are featured by various deadlines.

4.3 Cloud Deployment Models (In response to *RQ2*)

In the reviewed papers, researchers select a particular type of Cloud deployment model (private / public / community / hybrid) to carry out various activities such as implementing the proposed framework, deploying prototype or tools, evaluate approaches or simulate testing environments. Figure 4 shows the distribution of Cloud deployment models in the reviewed papers.

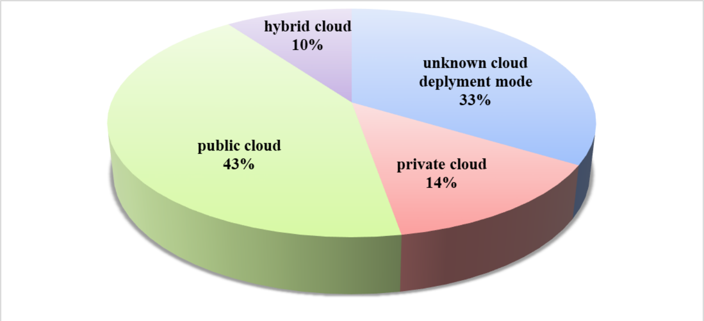

Fig. 4. Statistics of the reviewed papers by Cloud deployment models

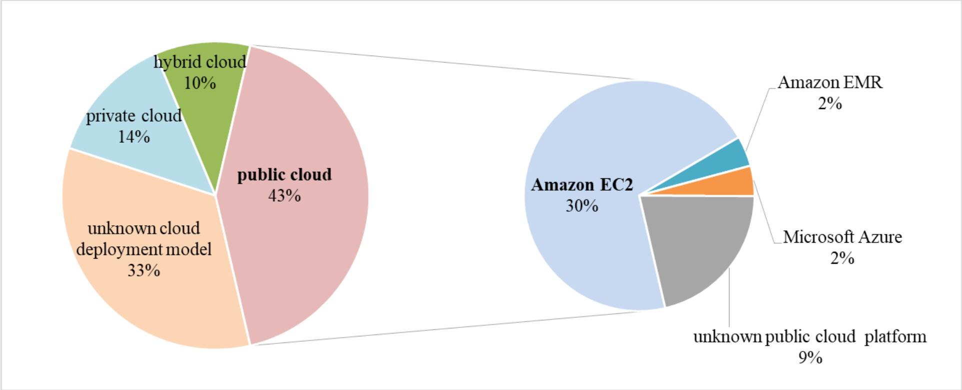

Fig. 5. Breakdown of public Cloud deployment model

It is worth noting that a significant fraction of the reviewed papers only mention Cloud service in general without explicitly telling the Cloud deployment model they used. In this case, we label it as "unknown Cloud deployment model". Apart from the section of "unknown Cloud deployment model", it is observed from Figure 4 that public Cloud is accredited as the principal Cloud deployment model. This finding is in line with our understanding that the public Cloud is the most common and well-known deployment model in comparison with the others. Accordingly, researchers are prone to choose public Cloud to deploy their proposed prototype, applications or tools and evaluate their proposed techniques. The second and third-ranked section is private Cloud and hybrid Cloud, occupying 14% and 10% respectively. Interestingly, community Cloud is not used in the reviewed papers.

Regarding public Cloud, we further investigate what specific public Cloud platforms were selected. Figure 5 gives an apparent breakdown of the public Cloud. It is found that some papers fail to state what specific public Cloud platform was used. Hence we mark them as "unknown public Cloud platform" in Figure 5. It is seen that Amazon EC2 is the preferable public Cloud platform, occupying 29%. Comparatively, only 2% of the reviewed papers use Microsoft Azure as their deployment platform.

Next, Figure 6 presents the breakdown of private Cloud. Similarly, some papers fail to state what specific private Cloud platform was used. Hence we label them as "unknown private Cloud platform" in the figure. It is observed that OpenStack is the most favorable private Cloud platform, occupying 8%, while its competitor OpenNebula and VMware vSphere equally share 1%.

Moreover, the breakdown of the hybrid Cloud is shown in Figure 7. The section of "unknown private and public Cloud platform combination" denotes that it is not deducible what specific mixture of public and private Cloud platform used according to the reviewed papers. It is interesting to find that OpenStack and Amazon

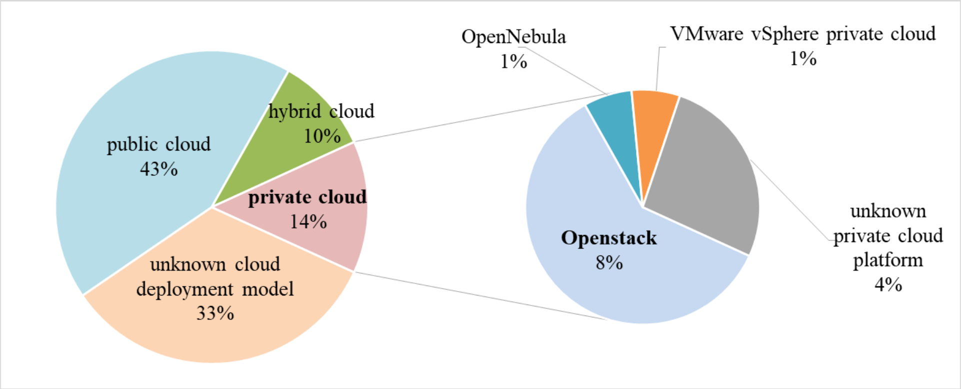

Fig. 6. Breakdown of private Cloud deployment model

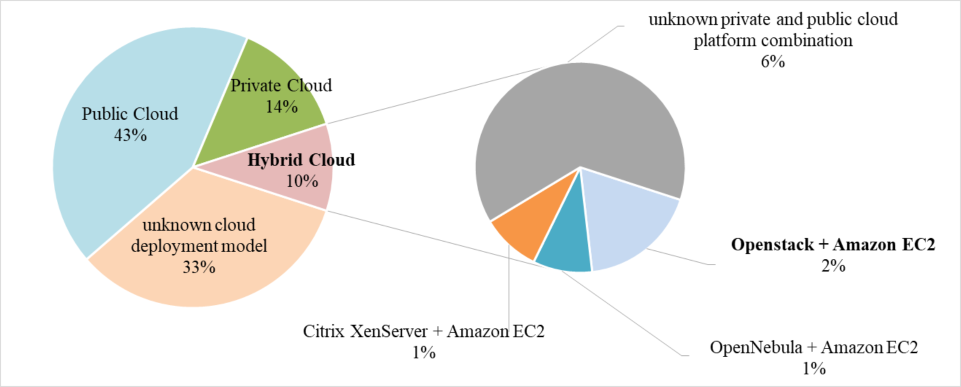

Fig. 7. Breakdown of hybrid Cloud deployment model

EC2 is the more popular combination despite having a small percentage (2%) than other combination such as OpenNebula plus Amazon EC2, Citrix XenServer plus Amazon EC2 occupying 1% respectively.

4.4 Techniques (In response to *RQ3*)

Figure 8 shows the statistics of the reviewed papers by different SLA management techniques. Based on this figure, it can be deduced that the dominant techniques are optimization, scheduling, simulation, monitoring, machine learning, constraint programming, and scaling. Further, Table 5 shows these techniques used to address SLA management for Cloud-hosted BDAAs and their breakdown by layer. From this table, it is clear that the most common techniques used in SLA management are Optimized-based, Simulation-based, Scheduling-based, Machine learning-based and Monitoring-based techniques. There are few research works that investigated other techniques such as scaling, fuzzy logic and error-handling, showing that these techniques are uncommon in the landscape of SLA management for Cloud-hosted BDAAs.

It is deserving to note that some authors combine more than one technique to address SLA management for Cloud-hosted BDAAs. They might use scheduling and machine learning-based technique, or fuzzy logic and machine learning-based technique, or monitoring and scaling technique. For example, Rajinder et al. [103] propose a overall architecture regarding SLA-aware scheduling of BDAAs across geo-distributed Cloud datacenter. Their proposed scheduling algorithms have two levels including coarse-grained and fine-grained. In this paper, firstly, they employ Naive Bayes algorithm to predict which category a user's BDAA belongs to. They then apply Adaptive K-nearest neighboring-based scheduling algorithm to discover which regional datacenter is appropriate

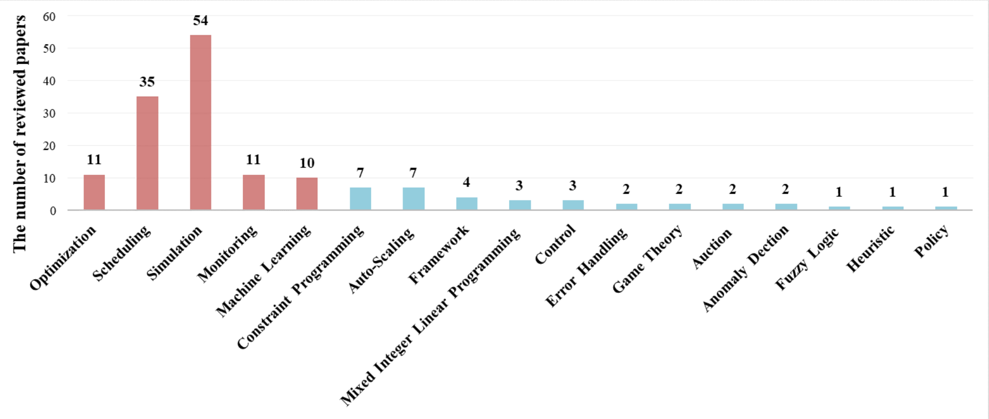

Fig. 8. Statistics of the reviewed papers by techniques

Table 5. Techniques used for SLA management for BDAAs in Clouds

Technique (papers quantity)	Service Layer (papers quantity)	References
Optimization-based (11)	BDPaaS (7) CIaaS (4)	[12, 29, 40, 41, 46, 63, 134] [48, 76, 94, 138]
Scheduling-based (35)	BDSaaS (8) BDPaaS (24) CIaaS (3)	[56, 73, 78, 85, 97, 103, 110, 111] [10, 15, 18, 31, 39, 43, 44, 50, 57, 60, 68, 77, 87, 90, 93, 95, 96, 109, 124, 128, 129, 133, 145, 146] [86, 119, 136]
Simulation-based (54)	BDSaaS (3) BDPaaS (36) CIaaS (15)	[25, 100, 110] [10, 12, 14, 29, 37–41, 43, 46, 50, 51, 54, 59, 60, 66, 68–70, 78, 87, 90, 93, 94, 109, 117, 121, 124, 129, 133, 134, 141, 142, 145, 146] [22, 23, 30, 48, 55, 72, 76, 86, 94, 104, 119, 136, 138, 139, 148]
Monitoring-based (11)	BDSaaS (5) BDPaaS (4) CIaaS (2)	[8, 11, 107, 108, 112] [59, 84, 122, 123] [102, 104]
Machine learning (10)	BDSaaS (5) BDPaaS (3) CIaaS (2)	[11, 25, 100, 103, 107] [51, 63, 122] [34, 104]
Control-based (3)	BDPaaS (3)	[16, 105, 106]
Constraint Programming (7)	BDSaaS (1) BDPaaS (6)	[74] [37, 38, 66, 67, 69, 137]
Scaling (7)	BDPaaS (5) CIaaS (2)	[35, 47, 75, 88, 117] [55, 102]
Mixed Integer Linear Programming (3)	BDPaaS (2) CIaaS (1)	[41, 67] [138]
Fuzzy Logic (1)	BDSaaS (1)	[25]
Framework-based (4)	BDSaaS (1) BDPaaS (2) CIaaS (1)	[118] [101, 144] [9]
Anomaly Detection (2)	BDPaaS (2)	[53, 104]
Auction (2)	CIaaS (2)	[22, 23]
Model Checking (1)	BDSaaS (1)	[61]
Heuristic-based (1)	BDPaaS (1)	[147]
Policy-based (1)	BDPaaS (1)	[79]
Game Theory (2)	BDPaaS (1) CIaaS (1)	[36] [139]
Error Handling (2)	BDPaaS (2)	[70, 71]

based on locations and requirements of users. Next, they performed the optimal scheduling of big data jobs based on their designed scheduling architecture and typical Amazon scheduling policies in the local server. In this paper, they investigated SLA metrics such as the time of waiting, the utilization of CPU, availability, estimated time to complete and response time. Experiments shows the efficacy of their coarse- and fine-grained scheduling algorithms. In [25], the authors focus on big media healthcare BDAAs in Clouds, which must satisfy SLAs for medical users. In this work, they exploited fuzzy logic to orchestrate a local- and global-based Cloud federation model that optimizes the selection decision making regarding target Cloud data centers. The model considers the trades off between the users' application service quality and providers' profit when choosing federated data centers. Also, the model acknowledges the dynamic behavior that user requests posses and system environments. Through the precise estimation of resource requirements for processing big data jobs using multiple linear regression algorithms, the accuracy of selection decision is significantly enhanced. In the subsequent subsections, we will give the details of the fundamental properties of some representative techniques and their application in some of the reviewed papers.

4.4.1 Optimization-based Generally speaking, the appropriate utilization of resources makes the tasks of SLA management more favorable for providers. As a result, providers continuously demands optimization-based algorithms that can optimally allocate/reallocate resource to maximize resources utilization and providers' profit. It makes sense that an optimization-based technique is essential in this context.

For Cloud-hosted BDAAs, the vast configuration diversity and dependency across different layers makes it difficult for customers to choose appropriate configurations or even decide an applicable background regarding their decisions. Moreover, the extant simple optimization algorithms fail to meet the requirements of most BDAAs that are often featured by different objectives, either because one of the objectives is unsatisfied, or the results appear far from the optimum [28]. Consequently, allocating Cloud resources (at CIaaS level) to big data platforms (BDPaaS level) is not any more a conventional single objective problem (e.g., minimizing time, maximizing resource) but involves multiple contradictory objective functions expressed by SLA metrics such as the maximization of classification accuracy using Apache Spark MLlib, the minimization of response time of MapReduce tasks using Apache Hadoop, the minimization of stream processing latency using Apache Storm and the maximization of CPU utilization and so on. Further, the formulated multi-objective optimization problem demands a considerable amount of computation that is increasing exponentially with the problem size in order to find optimum solutions.

Take the energy consumption optimization for BDAAs as an example, the authors [76] propose a multi-objective optimization-based technique that is aware of both energy and SLA requirements when placing and consolidating VMs. Their proposed technique considers to balance the performance and energy utilization of such system as well as SLA-compliance (e.g., availability and reliability). The results demonstrate that their technique achieves better performance on saving energy, reducing resource consumption and communication cost, minimizing the quantities of VM movements and SLA violations in comparison with the other extant tested approaches.

In terms of optimizing PaaS providers' profit, Dib et al. [29] propose a decentralized optimization-based policy and consider to pay the penalties when SLA violations occur. Their proposed policy achieves optimally exploiting private resources, especially at peak time, before leasing any public Cloud resources. The paper applies their proposed optimization-based policy into a realistic batch-based BDAA. Similarly, the authors [23] addressed the challenges of allocating resource while guaranteeing SLAs and maximizing providers' profit. In this paper, the penalty cost incurred by SLA violations is considered in order to increase providers' profit. They take into account SLA metrics such as execution time and deadline of jobs (i.e., urgency) in a combinatorial auction system and propose a new winner determined algorithm (an optimization-based technique). Experiments proves the efficacy of their approach on the reduction of the penalty payment incurred by SLA violation and maximization of providers' profit.

Unlike the above works, paper [48] addresses how to optimize the distribution of big data and allocation of computing resources on mobile Cloud platforms. As such, the authors propose a new network architecture and algorithms. They discuss an end-to-end thin-thick client collaboration to efficiently distributing data by transferring large dataset into splits depending on the bandwidth of Internet connection. Also, this paper details the procedure of selecting suitable algorithms that can efficiently enhance the utilization and allocation of resources and improve users experience by meeting expected SLA requirements (e.g., minimized VMs quantity, shortened execution time and budget).

4.4.2 Scheduling-based Scheduling is one of the fundamental techniques in addressing SLA management for Cloud-hosted BDAAs. Primarily, this technique is based on the above optimization technique where an Non-determin istic Polynomial-time Hardness (NP-hard) optimization problem has been formulated. Unlike the optimization-based technique, scheduling takes a further step of allocation works based on optimal solutions. Depending on different purposes, such allocation works include assigning physical resources (e.g., machines) to virtualized resources (e.g., VMs), or allocating VMs resources to particular batch or stream processing tasks, or designating platform resources to various BDAAs in a smart way while respecting SLA requirements.

Scheduling has been widely used for traditional applications or workflows in Cloud computing (CC) environment. However, unlike them, distributed data processing technologies such as MapReduce paradigm are now often utilized by many organizations to deploy their big data analytical applications (BDAAs). Therefore, scheduling algorithms used for traditional applications or workflows in CC environment cannot be applied directly to BDAAs in Clouds due to the complexities that data processing frameworks incur and the difference of resource allocation mechanisms that big data brings. As a result, various SLA-based scheduling mechanism and algorithms have been proposed, which primarily aims to optimize resource utilization and provides optimal resource allocation/reallocation solutions for Cloud-hosted BDAAs while meeting multiple SLA requirements.

When addressing SLA management for BDAAs in Clouds, scheduling technique can be applied at different layers. Hence, it can be classified into three classes, i.e., "scheduling at the BDSaaS layer", "scheduling at the BDPaaS layer" and "scheduling at CIaaS layer".

Scheduling at the BDSaaS layer

Optimally and strategically providing low-level resources to support BDAAs, jobs or workflows while guaranteeing agreed SLAs between providers and customers is the fundamental objective for the tasks of scheduling at BDSaaS layer. Scheduling at this layer has twofold consideration. On the one hand, it should satisfy users' SLA requirements and optimize objectives such as complete time, makespan, user capital expenditure, and application performance from the customers' perspective. On the other hand, it should efficiently schedule big data platform resources to the application layer to maximize profit or reduce the carbon cost or energy consumption by Cloud centers from the providers' perspective [17].

Verma et al., [124] consider SLA violation concerning the performance of MapReduce-based BDAAs and propose an automatic framework of resource inference and allocation. According to their proposed framework, they firstly profiled some common performance features such as soft deadline and then estimate the number of resources required for completing jobs to meet the deadline. Their proposed algorithm can efficiently schedule the execution sequence of jobs and determine the resources quantities allocated to these jobs while meeting job deadlines.

The authors [145] addressed the challenge of resource scheduling to optimize profit of providers. To this end, they first proposed a scalable and adaptive policy of admission control. Then, they developed a novel algorithm that can optimally schedule resources according to users' query requests while guaranteeing SLAs on deadlines and budgets, and prompt responses with manageable monetary expense.

Scheduling at the BDPaaS layer

Scheduling at the BDPaaS layer aims at allocating some dependent and independent tasks to VMs in Hadoop clusters. A valid scheduling algorithm can provide the optimal solution of task distribution over various VMs in a cluster depending on the requirements of execution time and availability of resources. An optimal distribution of tasks can minimize the scheduled tasks' average execution time and maximize the utilization of allocated resources. As such, the response time of tasks that are to be processed is minimized and resources consumption is reduced [116].

Wang et al. [128] developed a scheduling algorithm at platform-level for MapReduce-based BDAAs that have two practical SLA constraints (i.e., budget and deadline) over the heterogeneous Cloud datacenters. They designed a greedy-based optimization algorithm that can find appropriate VMs from an established pool of different VMs to minimize the job completion time and monetary cost of executing jobs.

Tian and Chen in [120] took into account the entire processing phrases for MapReduce jobs. In this paper, they designed a cost model that formulates the correlation between the input data volume, the MapReduce resources availability, and the Reduce tasks complexity. They performed testing over a limited number of machines to learn model parameters. The proposed cost model can facilitate to make decisions in terms of the optimal amount of resources, the minimization of time under particular financial budget and the minimization of monetary cost under a specific time deadline. Experiments demonstrates that this cost model achieves decent performance and satisfies SLAs for MapReduce-based BDAAs.

Scheduling at the CIaaS layer

Scheduling at this layer is more relevant with the optimal mapping virtualized resources onto physical resources in a homogeneous or heterogeneous environment and with the optimal use of the underlying Cloud resources.

Nita et al. [86] discuss the challenge of transferring big data across various Cloud datacenters where the performance of VMs migration and data transfers are affected. They describe an optimized method that can transfer large dataset according to the characteristics of network and take into account the SLA constraints such as minimizing the duration of individual VM migration. They proposed a scheduling policy that consists of two greedy-based algorithms to transfer large dataset. It manages and maintains an SLA-aware network that impacts the performance of Cloud. They evaluated the proposed scheduling policy by means of the simulation of SLA constraints at CIaaS layer.

In order to address the issue of file requests' tail latency, the authors [119] employed information flow queue theory to provide an optimal scheduling algorithm for erasure codes-based Cloud storage systems (at CIaaS layer). They first designed a model based on k-marriage flow queue. They then built a multi-objective based scheduling strategy to find the optimum depending on SLAs preferences of users. Their solution is featured by the decentralization of the queue form that outperforms the centralization of queue format in terms of the elimination of the overhead of block. Their simulated results showed their approach decently improves tail latency in comparison with the extant approaches of data displacement .

4.4.3 Simulation-based Simulation is also a popular technique used to address SLA management for Cloud-hosted BDAAs, which allows providers to evaluate a broad spectrum of components such as workload, processing elements (e.g., MapReduce, storm), data centers, storage, networking, and SLA constraints.

When providers offer their big data analytics service to customers with awareness of relevant SLAs, they can identify potential issues before introducing them into the operations and focus on service meeting the agreed SLAs. As they define this service, coarse SLAs can be identified and decomposed to identify more targeted SLAs that in turn drive qualification of the feasibility of proposed solutions to meet SLA commitments. This can then be verified through simulation to identify further how other resources are impacted by any shortfall to inform prioritization in addressing any gaps to guarantee SLAs such as maximizing overall resource utilization or reducing idle time. Moreover, the simulation technique is used to predict system performance and further to study SLA impacts in a production environment.

The simulation technique abstracts, models and emulates BDAAs with a wide of components such as workload, or SLA metrics. In cases where the performance of application does not satisfy pre-specified SLAs, algorithms, scheduling, or monitoring are adjusted and further optimized, and corrective and proactive measures are adopted before an issue occurs. Hence, simulation is an essential technique to facilitate SLA management for Cloud-hosted BDAAs. Also, the real-world Cloud-hosted BDAAs covers a wide array of application domains including healthcare, social media, energy and so on. Each type of these application is characterized by diverse architecture, configuration, implementation and deployment requirements. Experimentation in a real environment such as Amazon EC2 or Microsoft Azure for different BDAAs can be challenging for manifold reasons:

- It is not economical to purchase or lease large-scale datacenter infrastructure that will precisely indicate realistic deployment of BDAA and allow researchers conducting experiments with changing hardware resource and dynamic framework configurations, as well as big data diversities in terms of volume, variety and velocity.
- The experiments are not repeatable, because some variables that are not under the control of the tester may affect experimental results.
- Much manual configuration effort involved especially in a real large testbed experiment environment that needs dynamic configurations significantly slows down the performance analysis and makes it almost impractical. As a consequence, it is remarkably challenging to reproduce the experiments outcomes.
- The experiments on a real large distributed platform are unrealistic to some degree due to a huge cluster where a considerable of nodes run in different conditions.

In this case, the simulation technique offers significant advantages to SLA management for Cloud-hosted BDAAs. For example, researchers can conduct controllable and repeatable experiments by means of simulation technique. Also, it becomes easier to study if SLAs met or breached, and investigate how SLAs is impacted by various resources configuration from different layers in a simulated testbed as compared to a real experiment. Simulation technique makes experiments under various configurations of hardware resources easier and provides insights for practitioners to understand the impact that each design choice is upon to SLA guarantees. They also improve the possibility that researchers can share their simulation environment, which contributes to better hypothesis evaluation and results reproducibility. Finally, researchers can instantiate various processing frameworks of BDAAs and multiple workload scenarios as needed by the virtue of simulation-based technique.

We find that 53 papers among the 109 reviewed papers have applied simulation technique, occupying 49%. In order to investigate what particular simulation tools used, we further examine these 53 papers and find interesting results that (i) some papers generally mention that a simulation-based experiment has been conducted without explicitly stating what particular simulation tool used [12, 37, 38, 51, 53, 70]; (ii) some papers simply state that they developed their simulation tools by Java programming and keep them as proprietary code without giving details [90, 119, 136]; (iii) Other papers give specific description regarding simulation tools used [10, 30, 66, 87, 110, 124, 145]. For the first two cases, we label them as "unknown simulation tools". For the third case, we further find that three types of simulation tools are often used in the reviewed papers. They are discrete event simulator (DES) [45, 140], MRPerf [126] and Yarn Scheduler Load Simulator (SLS) [4]. Figure 9 shows the distribution of these three types of simulation tools.

It is observed that DES is the preferable simulation tool, with a percentage of 30%. Among DES, Cloudsim is the dominant one (26%) and widely used by researchers in their papers. Lots of authors use Cloudsim to implement their algorithms to emulate the CC environment or further implement additional logic to mimic the behavior of the MapReduce model. The second-ranked simulation tool is SLS [43, 87, 109, 110], which can support the simulation of large Yarn clusters and application loads in an individual machine. It is interesting to find that two papers use MRPerf [54, 124], a simulator dedicatedly designed for MapReduce jobs to understand how they perform and study the impact of SLA on various Hadoop configuration settings.

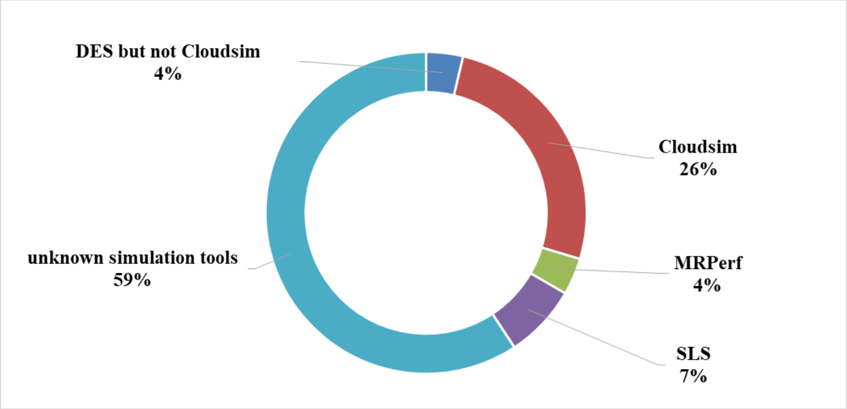

Fig. 9. Statistics of simulation technique used in the reviewed papers

4.4.4 Monitoring-based Monitoring is also an important technique used to manage SLAs for BDAAs. Generally, customers rely on SLAs to deliver the promised quality and level of service. Al through it is readily to notice non-availability or downtime, other types of SLA violation such as performance degradation of VMs and high error rates of APIs are not always easily detected, which can considerably impact the experience of end users. Therefore, monitoring is critical to assure the conformance of SLA and produce fundamental audit trail when SLA violation happens. Moreover, monitoring is essential for providers to guarantee SLA and offer the satisfactory experience to customers [130].

Monitoring of Cloud-hosted BDAAs involves dynamically tracking SLA metrics related to physical resources they share, virtualized resources at CIaaS level (i..e, VM, network and storage), big data processing framework such as Hadoop cluster at BDPaaS level as well as various applications (e.g., smart health, stock recommendation system) running at BDSaaS level. Monitoring is an essential technique to manage SLAs, assisting providers in (i) optimizing the operation of their applications and resources; (ii) capturing performance deviation of application and resources consumed; (iii) monitoring key performance indicators of the applications; (iv) accounting SLA violations regarding specific SLA metrics.

Andreolini et al. [11] take into account the cost minimization regarding computation and communication while assuring peak accuracy in detecting pertinent variations of system behavior guaranteed. They developed an algorithm that can elastically and reliably monitoring big data, which can adapt to update frequencies and sampling intervals. They used real-time series to perform experiments, which shows that their proposed algorithm outperforms extant algorithms in terms of reducing monitoring overheads and maintaining data quality.

The authors [108] develop a method that exploits runtime monitoring to guarantee the applications' performance. They implement a monitoring framework, which collects monitoring data at runtime in a realistic Cloud environment. They then design a performance model that uses data mining techniques to extract from authentic monitoring data at runtime. This model sheds lights on how to adjust the strategy of provisioning resources under specified performance-based SLA requirements.

In the context of monitoring stream-based events in a complicated and time-constrained system, the authors [59] design a framework that can real-time monitor and process large-scale log file streams from various sources. They applied the central limit theory to verify soft deadlines in a real-time system and used the probabilistic deadline to ensure SLA satisfied regarding deadline. Flume is used to collect, aggregate, and transfer voluminous stream-based data from multiple sources to a centralized place where Hadoop HDFS is operated. They extended a generic monitoring architecture and illustrated how to calculate the likelihood of SLA violation. This solution is beneficial for a system of real-time monitoring to determine the deadline for SLAs compliance.

4.4.5 Machine Learning-based Some of reviewed papers use machine learning techniques to study SLA management for Cloud-hosted BDAAs from different aspects. Not only is machine learning used to predict the prospective behavior of resources, but also to detect SLA violations. Machine learning-based technique provides machine derived intelligence to the task of SLA-driven optimization and configuration dependencies across multiple layers. It allows continuously learning many complex behaviors and interactions among interrelated objects/entities in BDAAs scenario and taking the guesswork out of many aspects involved in meeting SLAs more efficiently and cost-effectively. For example, collecting large data regarding VMs, storage, and network at CIaaS layer, Hadoop cluster at BDPaaS layer, and applications at BDSaaS layer, then feeding these data to machine learning-based system, finding the hidden patterns and fixing issues before they might violate SLA guarantee. As long as this wealth of data is gathered, processed and analyzed, machine learning-based technique can learn what constitutes normal behaviors, and it is this baseline that gives the system the ability to detect anomalies and find causes automatically. Thus SLA violation can be avoided. Also, it can simulate and predict the impact of making certain changes in resources and their allocations, which can be particularly useful for meeting SLA objectives such as maximizing resource utilization.

Lama et al. [63] developed a Hadoop-based system that can allocate diverse Cloud resources and automate multiple Hadoop parameters configuration while guaranteeing SLAs requirements such as minimal monetary cost incurred. It addressed the major issue of providing MapReduce-based BDAAs under different performance deadlines. Their approach consists of a novel framework including two phrases (machine learning-based offline phrase and optimization-based online phrase). The offline phrase clusters various Hadoop jobs by using Support Vector Machine algorithm. The clustering result is then regarded as an input and fed into the subsequent online phrase. The online phrase exploits optimization-based techniques to assign Cloud resources and automate the configuration of Hadoop parameters.

The authors [122] address the challenge of optimal resource provisioning for scalable BDAAs. Firstly, they identified that most of the applications running inside JVMs such as Spark highly demand effective memory resources. Then, they consider applications that is featured by their SLAs (i.e., relative delay) and apply Random Forest algorithm to predict their valid memory requirements. The prediction approach can uncover the hidden behavior of BDAAs' memory consumption and forecast dynamic prospective memory utilization in distributed Cloud environments.

4.5 SLA Metrics (In response to *RQ4*)

In this element, we examine SLA metrics accounted for in the reviewed papers to find out what SLA metrics have been discussed and how often they are discussed. Table 6 summarizes SLA metrics and describes their measurements.

Further, Figure 10 present the pictorial representation of the frequency of the above SLA metrics that have been discussed in the reviewed papers. It is observed that the most studied SLA metrics are performance, deadline, resource utilization, and cost. This is consistency with our understanding that actors (i.e., Providers, Customers, and End Users) care more about SLA metrics regarding deadline, cost, performance and resource utilization in Cloud-hosted BDAAs. The least studied SLA metrics are serviceability, consistency, elasticity, security, capacity, reliability, and scalability. This is because these SLA metrics have limitations regarding measurability [92]. The medium level discussed SLA metrics include profit, budget, energy, availability, fault tolerance, and accuracy. These category of SLA metrics are attracting increasing interest from researchers.

It is also found that the above SLA metrics scatter in the reviewed papers, without an organized and clear categorization. Therefore, it is necessary to examine SLA metrics for Cloud-hosted BDAAs through the consideration of building a clear categorization scheme while respecting BDAA characteristics, such that providers and customers will benefit from this categorization scheme when making conventions and engineering SLAs between them.

Table 6. SLA metrics and their measurement for BDAAs in Clouds

SLA Metrics	Unit of Measure
reliability	the number of concurrent failures that are tolerable, MTTF and MTTR and produces the storage system MTTF, number of successful responses in percentage
energy	cost per kWh, power (watts)
capacity	the capacity that the Cloud storage system can store. The options range from kilobytes to zettabytes
availability	percentage of service uptime or downtime
serviceability	period of an outage, duration between consecutive service failures, time to switch over from a failure, time to completely recover from a service failure
security	the ability to detect or tolerate malicious attack
consistency	the degree of equality between responses to queries issued by BDAAs
scalability	the ability to horizontally increase the storage or processing capacity or throughput, and the ability to add more resources (e.g., more processors, memory, bandwidth) to each node to increase capacity or throughput vertically
elasticity	the ability to dynamically and rapidly adjust resources to absorb the demand
fault tolerance	the percentage of continuing operating properly when failures (e.g., data node of Hadoop is down, Map or Reduce task fails) occur
accuracy	percentage of accurate prediction or analysis
cost	monetary cost in terms of VM computing per time unit, electricity prices
profit	revenue made per request
budget	upper bound on monetary cost (dollars) to complete data processing tasks
deadline	upper bound on time (hour) to complete data processing tasks
performance	• throughput: MB/sec • throughput: MB/sec • data freshness • time pertinent: waiting time/ response time/execution time /job processing time/job completion time
resource utilization	• CPU pertinent: MIPS, number of cores regarding CPU or vCPU, CPU utilization etc,. • memory pertinent: MB/GB, memory utilization etc,. • storage pertinent: storage size, I/O throughput etc,. • network pertinent: bandwidth, data transfer time etc,.
request arrival rate	request per second, arrival rate factor (user side) etc.,
quantity pertinent	• the quantity of working node allocated for batch-based or stream-based processing • the quantity of replicas • the quantity of required parallel threads • the quantity of tasks (i.e., Map or Reduce) • the quantity of jobs • the quantity of disks • input data size • the quantity of data blocks • the quantity of VMs

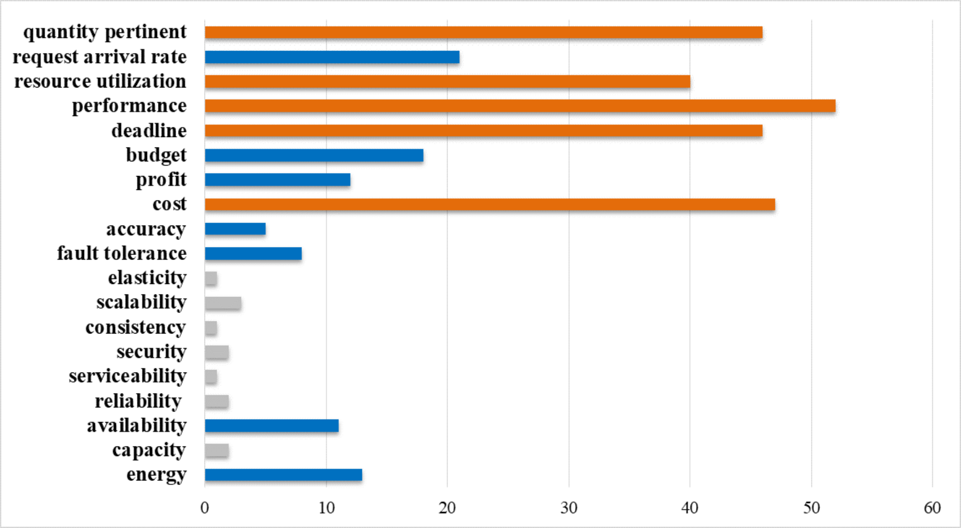

Fig. 10. Frequency of SLA metrics in the reviewed paper

4.6 Conceptualization (In response to *RQ5*)

4.6.1 Conceputal SLA Model In Cloud computing (CC) environment, designing conceptual SLA models or frameworks are often discussed. Alhamad et al. [7] proposed a conceptual SLA framework for CC environment. They consider four types of Cloud service (i.e., IaaS, PaaS, SaaS, storage as a service). For each different SLA, they present the fundamental parameters that are needed to establish a steady form of negotiation and conversation between customers and providers. Based on the above work, the authors [106] developed a new conceptual SLA model in CC environment called SLA as a Service (SLAaaS). SLAaaS can systematically and transparently integrate service levels and SLAs into Cloud. It considers the quality of service levels and SLA as the most superior elements in Cloud services.

Also, Labidi et al. [62] proposed a generic and semantic-rich model that is based on ontology theory in their paper. They developed a prototype to validate their proposed model. Through this prototype, the evaluation and triggered guarantee actions of SLAs can be automatically achieved during their monitoring process.

Moreover, in order to seek an optimal trade-off between revenues and costs while meeting SLA constraints, the authors [64] designed a service-based model that consolidates the major characteristics and SLA objectives of Cloud services. Although this model is generic and abstract, it is beneficial to derive a universal and automatic manager with the capability of managing any Cloud service, no matter what the layer.

In addition, the authors [26] proposed a formal model to describe SLA contents in CC environment and design autonomic mechanism of predicting SLA violation. Their proposed SLA model is devoted to formalizing a capability to manage SLAs violation detections for Cloud services. The proposed approach concerns the representation of information from both the SLAs and Cloud logs in a specific format.

All the above-mentioned works are confined to common Cloud service without specific consideration of BDAAs. To the best of our understanding, the conceptualization of SLA model dedicated to Cloud-hosted BDAAs is rare. The next section will proposes a new conceptual SLA model.

5 A New Model and Categorization Scheme of SLA Metrics

In this section, we design a conceptual SLA model dedicated for Cloud-hosted BDAAs. We further elaborate on this model to propose a multi-dimensional categorization scheme of SLA metrics for BDAAs in Clouds.

5.1 Cross-layer SLA Model for Cloud-hosted BDAAs

5.1.1 Design Principle and Requirements According to the layered architecture of Cloud-hosted BDAAs in Figure 1 of Appendix A, it is easy to figure out that each layer has two kinds of requirements that are crucial to service composition, which are functional and non functional requirements (FRs and NFRs). These requirements clearly define what the service provider should meet and provide to customers. The categorization of requirements for layer-based BDAAs in Clouds is shown in Figure 11.

On the one hand, the focus of FRs is on the functionality of the composed service. For instance, a sentiment analysis service from customer reviews using Amazon Comprehend detects sentiments in the text and extracts information about users' sentiment polarity (Positive, Negative, Neutral or Mixed) [32]. One of FRs in this case is that a sentiment analysis accuracy lower than 80% will never be purchased. A user requests FRs at the top layer (i.e., BDSaaS) and these requirements will be drilled down to the bottom layer (i.e., CIaaS), which provides concrete, scalable and on-demand Cloud resources. In other words, upper layer demands resources from a lower layer while a lower layer supplies resources to an upper layer. Thereby, each service in a layer is featured by unified interfaces by which Cloud-hosted BDAAs invoke possible functions. For example, Amazon EC2 acts as basic Cloud infrastructure and provides a functional interface that supplies its client (i.e., BDPaaS) with computing instances, to install and run software on these instances. By demanding scalable Cloud resources from Amazon EC2, Amazon EMR located at BDPaaS layer can supply its client (i.e., BDSaaS) a fully managed Hadoop cluster

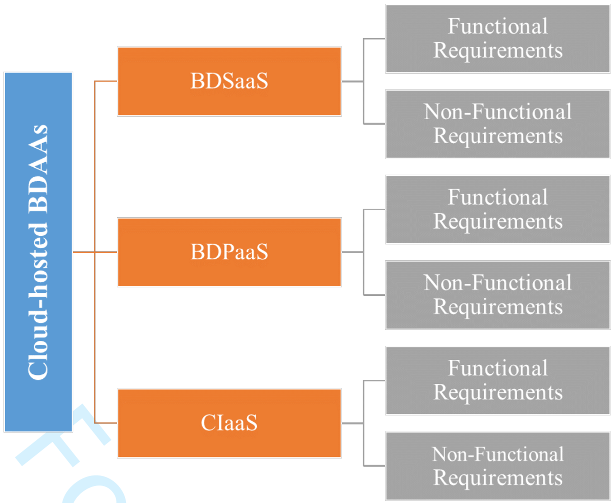

Fig. 11. Categorization of requirements for layer-based BDAAs in Clouds

in minutes, and then Amazon Comprehend uses advanced techniques such as machine learning and natural language processing to predict sentiments as much accurate as it can. As a result, FRs specified by users could be met.

On the other hand, NFRs are concerned with SLA metrics, An instance of a NFR for the above Amazon Comprehend sentiment analysis service is that the service response time to a user should be no more than 5 seconds. NFRs are encapsulated and incorporated into SLAs, where multiple metrics are considered such as maximum data transfer ratio, maximum availability, and minimum network latency. NFRs are important to big data analytics service composition and are often formally expressed in SLAs as part of contracts agreed between providers and customers. Even though FRs are met, unsatisfied NFRs such as slow or unreliable service may still not be adopted for BDAaaS.

In addition to functional and non-functional requirements, the dependency relationships between SLAs across different layers of BDAAs is another critical aspect in designing SLA model for the applications. A sole layer is impossible or struggles to provide either FRs or NFRs, thus it is bound to compromise service quality for customers. Having all layers work jointly, the agreed service quality can be guaranteed in the end. Accordingly, we need a novel SLA model for Cloud-hosted BDAAs that should meet the following essential design principles:

- Allowing the definition of both functional and non-functional interfaces that expose SLAs by layer for big data analytics service.
- Representing a seamless integration between SLAs and BDAaaS across layers.
- Considering SLAs for Cloud-hosted BDAAs in a unified and structured way.
- Reflecting strong dependency relationships between those SLAs.
- Possessing an universal applicability regardless of BDAAs.

5.1.2 Proposed Cross-layer SLA Model for Cloud-hosted BDAAs Keeping the aforementioned design principles in mind, we propose a novel cross-layer SLA model for Cloud-hosted BDAAs (named CL-SLAMfBDAAs) shown in Figure 12. As shown in this figure, there are four interacting actors located at different layers. They are end users (e.g., Business Users, Subject Matter Expertise, Data Scientist or Data Analyst), BDSaaS provider (e.g., Salesforce or BrandsEye), BDPaaS provider (e.g., Google Cloud Dataflow, Amazon EMR or Microsoft Azure HDInsight), and CIaaS provider (e.g., Google Compute Engine, Amazon EC2 or HP Cloud). These actors are involved in a set of activities, for instance, negotiating, clarifying and specifying FRs and NFRs, and formulating NFRs into agreed SLAs in each layer. Moreover, it is observed from this figure that SLAs are dedicatedly divided into three

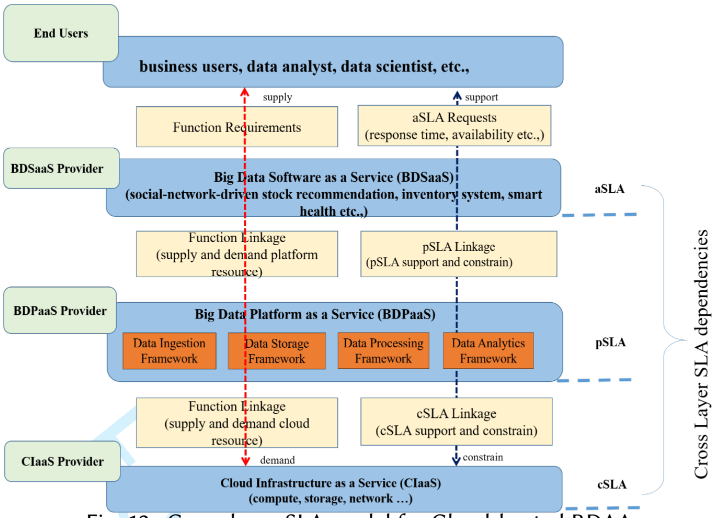
End Users
business users, data analyst, data scientist, etc.,
supply
support
Function Requirements
aSLA Requests
(response time, availability etc.,)
BDSaaS Provider
Big Data Software as a Service (BDSaaS)
(social-network-driven stock recommendation, inventory system, smart health etc.,)
aSLA
Function Linkage
(supply and demand platform resource)
pSLA Linkage
(pSLA support and constrain)
BDPaaS Provider
Big Data Platform as a Service (BDPaaS)
Data Ingestion Framework
Data Storage Framework
Data Processing Framework
Data Analytics Framework
pSLA
Function Linkage
(supply and demand cloud resource)
cSLA Linkage
(cSLA support and constrain)
CIaaS Provider
demand
constrain
Cloud Infrastructure as a Service (CIaaS)
(compute, storage, network …)
cSLA
Cross Layer SLA dependencies

Fig. 12. Cross-layer SLA model for Cloud-hosted BDAAs

categories including application-level SLAs (aSLA), platform-level SLAs (pSLA) and Cloud infrastructure-level SLAs (cSLA).

Furthermore, there are strong bonds between the upper layer and the corresponding lower layer of SLAs. From the perspective of FRs, the actor at each layer has a bidirectional relationship with its neighbor's layers either by demanding or supplying behavior. In other words, the upper-layer actor demands or requests resources from the lower-layer actor while lower-layer actor supplies the resources requested to upper-layer actor. From another perspective, a two-way relationship is existed, where SLAs at each layer either constrain or support SLAs in its adjacent layers. Concretely, an end user requests FRs along with NFRs through the interface with BDSaaS provider. The NFRs will be negotiated and defined into aSLAs between them. Then, an aSLA will be interpreted and formulated into a set of pSLAs. After that, a pSLA will be transformed into a set of cSLAs. From topmost to bottom, the upper-layer SLA sets the constraints into its lower-layer SLA and decides how well the lower-layer SLA must work to meet service-level objectives in the end. In turn, the low-layer SLA works hard to support its upper-layer SLA. For instance, if the availability in aSLA is 99% (a 99% availability at this layer means that users will be able to access the application at least 99% of the time), then, the availability in pSLA must be hover somewhere between 99% and 99.99%, and the availability in cSLA must be higher than the availability in pSLA. In the absence of meeting this, it might fail to guarantee SLA constraints.

In terms of SLA metrics, cSLA metrics such as VMs quantity, CPU and memory resources utilization, or the availability of VMs affects the pSLA metrics such as the quantity of map and reduce nodes in Hadoop platform (Data Processing Framework at BDPaaS layer) or the other pSLA metrics such as the quantity of data nodes, transfer rate, and replication factors of NoSQL database service (Data Storage Framework at BDPaaS layer). Certainly, this at last affects aSLA metrics such as capital cost, availability and reliability of the applications. This exemplifies the strong cross-layer SLA dependency relationship. To guarantee the final SLA to customers, BDSaaS provider should guarantee aSLAs by interweaving pSLAs and cSLAs.

Finally, our proposed CL-SLAMfBDAAs model presents a unified and structured scheme to describe and interpret SLAs for Cloud-hosted BDAAs. In this novel model, SLAs are exposed and linked in a vertical motion, which is orthogonal to the layers and may apply to any of them. Based on this model, users know what different types of SLAs with various attributes exist and how they work collaboratively across layers to ensure the delivery of SLA guarantee. This model meets all the aforementioned design principles and requirements. It is further elaborated to propose a new categorization scheme of SLA metrics for Cloud-hosted BDAAs.

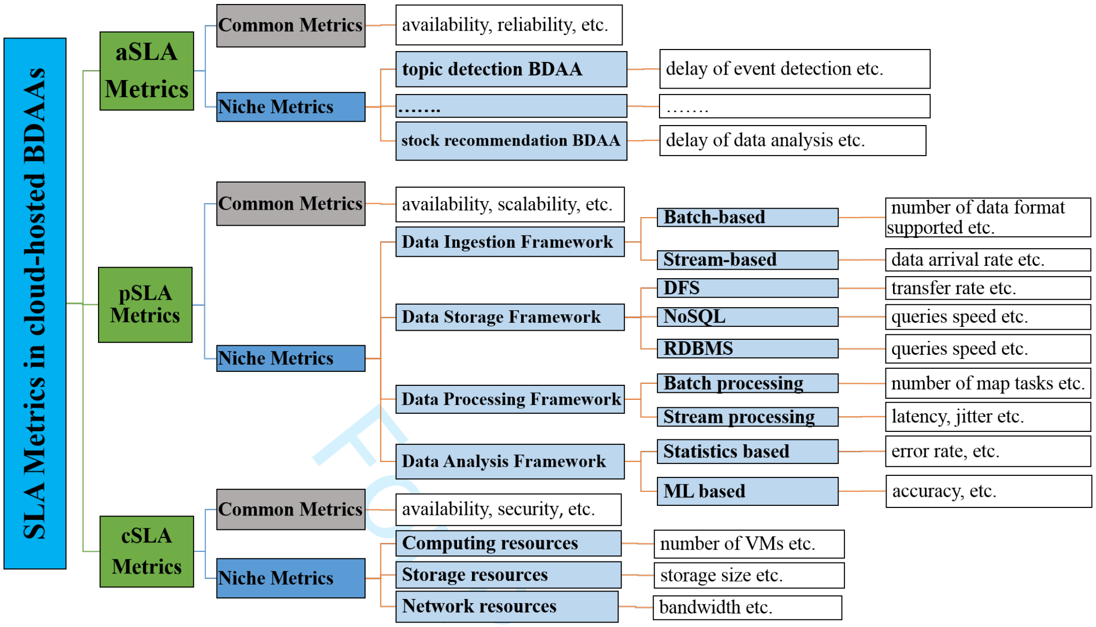

Fig. 13. Categorization scheme of SLA metrics for BDAAs in Clouds

5.2 Multi-dimensional Categorization Scheme of SLA Metrics for Cloud-hosted BDAAs

The typical SLA metrics at CIaaS layer that customers expect are the number of VMs, memory size, CPU usage, hard disk utilization, memory usage, additional network parameters and so on. While at BDPaaS layer, the example of SLA metrics include throughput, response time, and availability. For instance, in the case of a Hadoop cluster (at BDPaaS layer), we have metrics such as execution time, job turnaround and makespan in terms of MapReduce tasks [52]. At BDSaaS level, a particular SLA metric is highly determined by the genre of BDAA. For example, the rate and quality of data transfer are important for any video streaming-oriented BDAA, while latency of processing and network generally interests a batch-based BDAA. It is worth to note that SLAs at each layer might have an endless variety of metrics depending on different components and the nature of applications.

Figure 13 shows our proposed extensible and multi-dimensional categorization scheme of SLA metrics for Cloud-hosted BDAAs. This schema not only defines the most commonly used metrics for each type of SLA (i.e. aSLA, pSLA and cSLA), but also presents niche SLA metrics to be consistent with different components at each layer.

5.2.1 aSLA Metrics There are lots of different types of BDAAs across a wide range of industries. For instance, topic detection and tracking applications, large-scale log analysis applications and business intelligence. Due to their wide variations, listing all SLA metrics at this level is impracticable. Hence, we select some typical Cloud-hosted BDAAs and present common aSLA metrics for them as shown in Table 7. To embody the unique features of these BDAAs, we provide niche aSLA metrics for them in Table 8.

5.2.2 pSLA Metrics There are four main components/frameworks at BDPaaS layer (i.e., data ingestion, storage, processing and analysis). Selection of the specific software or tool as an instantiation of the aforementioned different frameworks is based on many aspects such as flexibility, control and ease of use. Considering the differential nature and role-playing, each framework at BDPaaS layer has different pSLA metrics. Table 9 lists some common pSLA metrics, while some niche pSLA metrics for each framework are given in Table 10.

5.2.3 cSLA Metrics Companies like Microsoft, Google and Amazon offer infrastructure as a service. With this diverse range of Cloud infrastructures, most customers are perplexed to choose which SLA metrics should be defined and specified as the hardware section of cSLAs. To clear away this confusion, we give the most common and niche SLA metrics that interest customers when using Cloud resources in Table 11 and Table 12 respectively.

To better understand SLAs across different layers of BDAA, we present a SLAs template using a real Cloud-hosted BDAA in Appendix B.

6 Open Issues and Future Trends

In this paper, we performed a taxonomic study on SLA-specific management for big data analytical applications (BDAAs) in Clouds. Particularly, we addressed the most pertinent survey questions in this field.

The taxonomy-based study suggested that (i) The BDSaaS and CIaaS layers have received much less interests from researchers in comparison with the BDPaaS layer. Hence, one of future trends could be paying more attention to the former two layers. Taking BDSaaS layer as an example, future researchers could investigate how to manage SLAs by using more domain-specific BDAAs such as healthcare, banking, and smart city; (ii) In terms of BDPaaS layer, researchers put less attention to data storage (e.g., NoSQL) compared to data processing (e.g., Hadoop). Since NoSQL attracted significant interest in recent years and also play an important role in Cloud-hosted BDAAs. Hence, future work could study NoSQL-specific SLA management in Clouds; (iii) Considering data processing technologies (batch or stream), the minority of papers discuss SLA management for stream-based applications compared to batch-based applications. Since stream processing has recently received increasing attention because of technological innovations which have facilitated the creation, maintenance, and processing of massive data with lower latency and better resilience. Therefore, future trend could particularly focus on SLA management for stream-based BDAAs; (iv) Techniques such as constraint programming, auto-scaling, error handling and so on received less attentions by extant researchers compared with optimization, scheduling, simulation, monitoring and machine learning techniques, which points out another possible future work.

As a conclusion, we explored the current research state in the field of SLA management for Cloud-hosted BDAAs and provided some future works. Provisionally, future researchers would take advantage of the ideas from this taxonomy-based survey as an entry point to address some gaps and most importantly enhance the maturity of the research field on SLA management for Cloud-hosted BDAAs.

Table 7. Common aSLA metrics

aSLA Metrics	Description
availability	The uptime of BDAA for end users in a specific time frame
financial cost	The total financial cost of using BDAA
respone time	Time to complete and receive the analysis result
usability	The degree to be easily used by end users through built-in interfaces
deadline	The total time of executing a BDAA and returning final results to its end user
reliability	The ability to maintain operational status in the majority of cases
integration	The degree of simplicity for integrating with applications and services require data from BDAA
capacity	The capacity the BDAA can provision
scalability	The ability to scale when expanding large volume of data or vast number of users
customizability	The flexibility to use with diverse kinds of users
pay-per-use billing	The ability to charge based on the usage of resources or duration
security	The degree to be exempt from malicious attack incurred by the network, software, tools, process or human, which results in significant damage or loss
energy efficiency	The degree of overall energy consumption on a per unit level (e.g., per capita, per customer, per hour)
the ratio of the admitted workloads	The proportion between the permitted workloads quantity and the submitted workloads quantity by end users

Table 8. Niche aSLA Metrics by different types of BDAAs

BDAA	aSLA Metrics	Description
Topic detection and tracking application [21, 127]	• event detection delay • input throughput • output throughput	• the delay of detecting events such as earthquake, football matches • the number of input events that are processed during a period • the number of derived events that are produced during a period
Big data-based traffic congestion detection system [?]	• alert sending delay	• the delay of sending alerts of existing traffics
Large-scale ingestion of analytics events and logs application [6]	• log integrity • alert sending speed	• the percentage of logs that can be seen • the number of alerts sent per second
Business intelligence on big data [20]	• the delay of decision making process	• the delay of a decision that is made based on business intelligence
Social network driven stock recommendation system [98, 143]	• data analysis delay	• the delay of completing stock analysis based on social network data
SLA based healthcare big data analysis and computing in Cloud network [100]	• disease prediction accuracy	• the degree to accurately forecast patients' prospective disease condition
Google smart inventory management system [2]	• inventory accuracy	• the degree to grasp accurate information regarding inventory and product at any time

Table 9. Common pSLA metrics

pSLA Metrics	Description
availability	The uptime of each framework in BDPaaS in a specific time
integration	The abilities to integrate with other frameworks and platforms
capacity	The capacity the BD platform can provision
scalability	The abilities to expand platform-level resources as requests or workloads increase
pay-per-user billing	The ability of the charging based on which framework or time of utilization
energy efficiency	The degree of energy consumption on a per unit level (e.g., per capita, per customer, per hour) for each framework
security	The degree to be free from malicious attack incurred by software or tools in each framework at BDPaaS layer, which brings damage or loss
fault tolerance	The ability to maintain an appropriate operational status even in the case of failures within its components

Table 10. Niche pSLA metrics by the different framework at BDPaaS layer

Framework	Sub Category	pSLA Metrics
Data Ingestion	Batch-based	• data size • the number of chunks • chunk size • throughput • the number of data format supported
	Stream-based	• data size • data arrival rate • latency • the number of data format supported
Data Storage	Distributed File Systems (HDFS)	• transfer rate (read, write) • latency (read, write, update) • the number of data nodes • replication number • the overall size of input data • the size of split data • network throughput
	NoSQL	• transfer rate (read, write) • latency (read, write, update) • the quantity of data nodes • replication factors • queries speed • transaction response time • data freshness
	RDBMS	• queries speed • query throughput • the number of connections • buffer pool usage • transfer rate (read, write) • latency (read, write, update) • replication factors • batch requests/sec • disk read I/O per sec • disk write I/O per sec
Data Processing	Batch-based	• the number of map tasks • the number of reduce tasks • the number of unhealthy nodes • the number of active nodes • the number of instances • upper bound on the time finishing the data processing task • block size • job turnaround • maximum allowed completion time
	Stream-based	• response time to streaming data • stream processing latency • peak system resource usage • system start-up time • Jitter (the variance of processing times)
Data Analysis	Statistics-based	how good is the statistical method (error rate, sensitivity, validity)
	Machine learning-based	• how good is the machine learning model (precision, recall, accuracy, sensitivity, specificity) • model training time • model training speed • the size of the machine learning model • average response speed for individual prediction requests • number of algorithms supported for data analysis

Table 11. Common cSLA metrics

cSLA Metrics	Description
availability	the uptime of Cloud infrastructure in specific time
capacity	The capacity that the Cloud infrastructure can provision
scalability	The ability to expand infrastructure-level resources (e.g., VMs) requested from BDPaaS level
pay as you go billing	The ability to charge based on time of utilization of VMs or storages
energy efficiency	The degree of energy consumption for data centers
security	the degree to be exempt from malicious attack incurred by Cloud infrastructure, which causes damage or loss

Table 12. Niche cSLA metrics by different components at CIaaS layer

Component	cSLA Metrics
Computing resources	• response time • CPU utilization • memory utilization • system load • scale up time • number of VMs • number of cores per CPU • memory size • duration of individual VM migration
Storage resources	• number of units of data storage • storage size • privacy • backup • hard disk utilization • I/O speed (bytes per second) • failure frequency • maximum downtime
Network resources	• network throughput • network bandwidth • network latency • accessibility to the Internet across the firewall

7 Acknowledgments

Dr. Chen's work was supported in part by the National Natural Science Foundation of China (No. 61772380), Major Project for Technological Innovation of Hubei Province (No. 2019AAA044), and the Science and Technology Major Project of Hubei Province (Next-Generation AI Technologies, No. 2019AEA170).

References

[1] 2019. Amazon Comprehend. https://aws.amazon.com/comprehend/

[2] 2019. Building Real-Time Inventory Systems for Retail. https://cloud.google.com/solutions/building-real-time-inventory-systems-retail

[3] 2019. Marketing Cloud Platform Overview. https://www.salesforce.com/au/products/marketing-cloud/platform

[4] June 2014. Yarn Scheduler Load Simulator (SLS). https://hadoop.apache.org/docs/r2.4.1/hadoop-sls/SchedulerLoadSimulator.html/

[5] November 2017. Google Prediction API and Google BigQuery SLA. https://cloud.google.com/bigquery/sla

[6] October 2018. Architecture: Optimizing Large-Scale Ingestion of Analytics Events and Logs. https://cloud.google.com/solutions/architecture/optimized-large-scale-analytics-ingestion/

[7] Mohammed Alhamad, Tharam Dillon, and Elizabeth Chang. 2010. Conceptual SLA framework for cloud computing. In *Digital Ecosystems and Technologies (DEST), 2010 4th IEEE International Conference on*. IEEE, 606–610.

[8] Khalid Alhamazani, Rajiv Ranjan, Prem Prakash Jayaraman, Karan Mitra, Meisong Wang, Zhiqiang George Huang, Lizhe Wang, and Fethi Rabhi. 2014. Real-time qos monitoring for cloud-based big data analytics applications in mobile environments. In *Mobile Data Management (MDM), 2014 IEEE 15th International Conference on*, Vol. 1. IEEE, 337–340.

[9] Ahmad B Alnafoosi and Theresa Steinbach. 2013. An integrated framework for evaluating big-data storage solutions-IDA case study. In *Science and Information Conference (SAI), 2013*. IEEE, 947–956.

[10] Mohammed Alrokayan, Amir Vahid Dastjerdi, and Rajkumar Buyya. 2014. Sla-aware provisioning and scheduling of cloud resources for big data analytics. In *2014 IEEE International Conference on Cloud Computing in Emerging Markets (CCEM)*. IEEE, 1–8.

[11] Mauro Andreolini, Michele Colajanni, Marcello Pietri, and Stefania Tosi. 2015. Adaptive, scalable and reliable monitoring of big data on clouds. *J. Parallel and Distrib. Comput.* 79 (2015), 67–79.

[12] Zhi-guang Ao, Ming-hai Jiao, Ke-ning Gao, and Xing-wei Wang. 2016. Research on Cloud Resource Optimization Model Based on Users' Satisfaction. In *Web Information Systems and Applications Conference, 2016 13th*. IEEE, 99–102.

[13] Marcos D Assunção, Rodrigo N Calheiros, Silvia Bianchi, Marco AS Netto, and Rajkumar Buyya. 2015. Big Data computing and clouds: Trends and future directions. *J. Parallel and Distrib. Comput.* 79, 3–15.

[14] William H Bell, David G Cameron, Luigi Capozza, A Paul Millar, Kurt Stockinger, and Floriano Zini. 2002. Simulation of dynamic grid replication strategies in optorsim. In *International Workshop on Grid Computing*. Springer, 46–57.

[15] Paolo Bellavista, Antonio Corradi, Andrea Reale, and Nicola Ticca. 2014. Priority-based resource scheduling in distributed stream processing systems for big data applications. In *Proceedings of the 2014 IEEE/ACM 7th International Conference on Utility and Cloud Computing*. IEEE Computer Society, 363–370.

[16] Mihaly Berekmeri, Damián Serrano, Sara Bouchenak, Nicolas Marchand, and Bogdan Robu. 2014. A control approach for performance of big data systems. *IFAC Proceedings Volumes* 47, 3, 152–157.

[17] Rajkumar Buyya, Chee Shin Yeo, Srikumar Venugopal, James Broberg, and Ivona Brandic. 2009. Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility. *Future Generation computer systems* 25, 6, 599–616.

[18] Xiaojun Cai, Feng Li, Ping Li, Lei Ju, and Zhiping Jia. 2017. SLA-aware energy-efficient scheduling scheme for Hadoop YARN. *The Journal of Supercomputing* 73, 8, 3526–3546.

[19] CL Philip Chen and Chun-Yang Zhang. 2014. Data-intensive applications, challenges, techniques and technologies: A survey on Big Data. *Information Sciences* 275, 314–347.

[20] Hsinchun Chen, Roger HL Chiang, and Veda C Storey. 2012. Business intelligence and analytics: from big data to big impact. *MIS quarterly*, 1165–1188.

[21] Tao Cheng and Thomas Wicks. 2014. Event detection using Twitter: a spatio-temporal approach. *PloS one* 9, 6, e97807.

[22] Yeongho Choi and Yujin Lim. 2015. Resource management mechanism for SLA provisioning on cloud computing for IoT. In *2015 International Conference on Information and Communication Technology Convergence (ICTC)*. IEEE, 500–502.

[23] Yeongho Choi and Yujin Lim. 2016. Optimization approach for resource allocation on cloud computing for iot. *International Journal of Distributed Sensor Networks* 2016, 23.

[24] I-Hsun Chuang, Yu-Ting Huang, Wei-Tsung Su, Tung-Sheng Lin, and Yau-Hwang Kuo. 2015. S4: An SLA-aware Short-Secret-Sharing cloud storage system. In *2015 Seventh International Conference on Ubiquitous and Future Networks*. IEEE, 401–406.

[25] Amit Kumar Das, Tamal Adhikary, Md Abdur Razzaque, Majed Alrubaian, Mohammad Mehedi Hassan, Md Zia Uddin, and Biao Song. 2017. Big media healthcare data processing in cloud: a collaborative resource management perspective. *Cluster Computing* 20, 2, 1599–1614.

[26] Lucia De Marco, Filomena Ferrucci, and Tahar Kechadi. 2015. SLAFM: A Service Level Agreements Formal Model for Cloud Computing. In *The 5th International Conference on Cloud Computing and Service Science (CLOSER 2015), Lisbon, Portugal, 20-22 May 2015*.

[27] Ramon Hugo de Souza, Paulo Arion Flores, Mário Antônio Ribeiro Dantas, and Frank Siqueira. 2016. Architectural recovering model for Distributed Databases: A reliability, availability and serviceability approach. In *2016 IEEE Symposium on Computers and Communication (ISCC)*. IEEE, 575–580.

[28] Laouratou Diallo, Aisha-Hassan A Hashim, Rashidah Funke Olanrewaju, Shayla Islam, and Abdullah Ahmad Zarir. 2016. Two objectives big data task scheduling using swarm intelligence in cloud computing. *Indian Journal of Science and Technology* 9, 28.

[29] Djawida Dib, Nikos Parlavantzas, and Christine Morin. 2014. SLA-based profit optimization in cloud bursting PaaS. In *Cluster, Cloud and Grid Computing (CCGrid), 2014 14th IEEE/ACM International Symposium on*. IEEE, 141–150.

[30] Mouhamad Dieye, Mohamed Faten Zhani, and Halima Elbiaze. 2017. On achieving high data availability in heterogeneous cloud storage systems. In *2017 IFIP/IEEE Symposium on Integrated Network and Service Management (IM)*. IEEE, 326–334.

[31] Sofia D'Souza and K Chandrasekaran. 2015. Analysis of MapReduce scheduling and its improvements in cloud environment. In *Signal Processing, Informatics, Communication and Energy Systems (SPICES), 2015 IEEE International Conference on*. IEEE, 1–5.

[32] Todd Escalona. January 2018. Detect sentiment from customer reviews using Amazon Comprehend. https://aws.amazon.com/blogs/machine-learning/detect-sentiment-from-customer-reviews-using-amazon-comprehend/

[33] Funmilade Faniyi and Rami Bahsoon. 2016. A systematic review of service level management in the cloud. *ACM Computing Surveys (CSUR)* 48, 3, 43.

[34] Victor AE Farias, Flavio RC Sousa, Jose Gilvan R Maia, Joao Paulo P Gomes, and Javam C Machado. 2018. Regression based performance modeling and provisioning for NoSQL cloud databases. *Future Generation Computer Systems* 79, 72–81.

[35] Anshul Gandhi, Sidhartha Thota, Parijat Dube, Andrzej Kochut, and Li Zhang. 2016. Autoscaling for Hadoop clusters. In *2016 IEEE International Conference on Cloud Engineering (IC2E)*. IEEE, 109–118.

[36] Eugenio Gianniti, Danilo Ardagna, Michele Ciavotta, and Mauro Passacantando. 2017. A game-theoretic approach for runtime capacity allocation in MapReduce. In *Proceedings of the 17th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing*. IEEE Press, 1080–1089.

[37] Adam Gregory and Shikharesh Majumdar. 2016. A configurable energy aware resource management technique for optimization of performance and energy consumption on clouds. In *2016 IEEE International Conference on Cloud Computing Technology and Science (CloudCom)*. IEEE, 184–192.

[38] Adam Gregory and Shikharesh Majumdar. 2016. A constraint programming based energy aware resource management middleware for clouds processing MapReduce jobs with deadlines. In *Companion Publication for ACM/SPEC on International Conference on Performance Engineering*. ACM, 15–20.

[39] Adam Gregory and Shikharesh Majumdar. 2016. Energy aware resource management for MapReduce jobs with service level agreements in cloud data centers. In *2016 IEEE International Conference on Computer and Information Technology (CIT)*. IEEE, 568–577.

[40] Lin Gu, Deze Zeng, Song Guo, Yong Xiang, and Jiankun Hu. 2016. A general communication cost optimization framework for big data stream processing in geo-distributed data centers. *IEEE Trans. Comput.* 65, 1, 19–29.

[41] Lin Gu, Deze Zeng, Peng Li, and Song Guo. 2014. Cost minimization for big data processing in geo-distributed data centers. *IEEE transactions on Emerging topics in Computing* 2, 3, 314–323.

[42] Muhammad Hanif, Hyungduk Yoon, Sunglim Jang, and Choonhwa Lee. 2017. An adaptive SLA-based data flow mechanism for stream processing engines. In *Information and Communication Technology Convergence (ICTC), 2017 International Conference on*. IEEE, 81–86.

[43] Ibrahim Abaker Targio Hashem, Nor Badrul Anuar, Mohsen Marjani, Abdullah Gani, Arun Kumar Sangaiah, and Adewole Kayode Sakariyah. 2018. Multi-objective scheduling of MapReduce jobs in big data processing. *Multimedia Tools and Applications* 77, 8, 9979–9994.

[44] S Hemalatha and S Valarmathi. 2016. Efficient Hybrid framework for parallel Resource and task scheduling in the Map reduce programming. In *2016 International Conference on Computer Communication and Informatics (ICCCI)*. IEEE, 1–7.

[45] Raymond Hoare, Jiyong Ahn, and Jesse Graves. 2002 of Conference. Discrete event simulator. Google Patents.

[46] Christoph Hochreiner, Michael Vögler, Stefan Schulte, and Schahram Dustdar. 2017. Cost-efficient enactment of stream processing topologies. *PeerJ Computer Science* 3, e141.

[47] Chao-Wen Huang, Wan-Hsun Hu, Chia-Chun Shih, Bo-Ting Lin, and Chien-Wei Cheng. 2013. The improvement of auto-scaling mechanism for distributed database-A case study for MongoDB. In *2013 15th Asia-Pacific Network Operations and Management Symposium (APNOMS)*. IEEE, 1–3.

[48] Pham Phuoc Hung, Tuan-Anh Bui, Kwon Soonil, and Eui-Nam Huh. 2016. A new technique for optimizing resource allocation and data distribution in mobile cloud computing. *Elektronika ir Elektrotechnika* 22, 1, 73–80.

[49] Walayat Hussain, Farookh Khadeer Hussain, Omar K Hussain, Ernesto Damiani, and Elizabeth Chang. 2017. Formulating and managing viable SLAs in cloud computing from a small to medium service provider's viewpoint: A state-of-the-art review. *Information Systems* 71, 240–259.

[50] Eunji Hwang and Kyong Hoon Kim. 2012. Minimizing cost of virtual machines for deadline-constrained mapreduce applications in the cloud. In *Proceedings of the 2012 ACM/IEEE 13th International Conference on Grid Computing*. IEEE Computer Society, 130–138.

[51] Shigeru Imai, Stacy Patterson, and Carlos A Varela. 2017. Maximum sustainable throughput prediction for data stream processing over public clouds. In *Proceedings of the 17th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing*. IEEE Press, 504–513.

[52] Gabriel Iuhasz and Ioan Dragan. 2015. An overview of monitoring tools for big data and cloud applications. In *Symbolic and Numeric Algorithms for Scientific Computing (SYNASC), 2015 17th International Symposium on*. IEEE, 363–366.

[53] Ali Imran Jehangiri, Ramin Yahyapour, Philipp Wieder, Edwin Yaqub, and Kuan Lu. 2014. Diagnosing cloud performance anomalies using large time series dataset analysis. In *Cloud Computing (CLOUD), 2014 IEEE 7th International Conference on*. IEEE, 930–933.

[54] Selvi Kadirvel and José AB Fortes. 2011. Towards self-caring mapreduce: Proactively reducing fault-induced execution-time penalties. In *High Performance Computing and Simulation (HPCS), 2011 International Conference on*. IEEE, 63–71.

[55] Hyejeong Kang, Jung-in Koh, Yoonhee Kim, and Jaegyoon Hahm. 2013. A SLA driven VM auto-scaling method in hybrid cloud environment. In *Network Operations and Management Symposium (APNOMS), 2013 15th Asia-Pacific*. IEEE, 1–6.

[56] Karim Kanoun, Cem Tekin, David Atienza, and Mihaela Van Der Schaar. 2016. Big-data streaming applications scheduling based on staged multi-armed bandits. *IEEE Trans. Comput.* 65, 12, 3591–3605.

[57] Banpreet Kaur and Ankit Grover. 2016. Optimizing VM Provisioning of MapReduce Tasks on Public Cloud. In *Proceedings of the International Conference on Advances in Information Communication Technology & Computing*. ACM, 79.

[58] Barbara Kitchenham, O Pearl Brereton, David Budgen, Mark Turner, John Bailey, and Stephen Linkman. 2009. Systematic literature reviews in software engineering–a systematic literature review. *Information and software technology* 51, 1 (2009), 7–15.

[59] Panya Kittipipattanathaworn and Natawut Nupairoj. 2014. SLA guarantee real-time monitoring system with soft deadline constraint. In *Computer Science and Software Engineering (JCSSE), 2014 11th International Joint Conference on*. IEEE, 52–57.

[60] KR Krish, M Safdar Iqbal, M Mustafa Rafique, and Ali R Butt. 2014. Towards energy awareness in hadoop. In *Network-Aware Data Management (NDM), 2014 Fourth International Workshop on*. IEEE, 16–22.

[61] Maria Krotsiani, Christos Kloukinas, and George Spanoudakis. 2017. Validation of Service Level Agreements using Probabilistic Model Checking. In *Services Computing (SCC), 2017 IEEE International Conference on*. IEEE, 148–155.

[62] Taher Labidi, Achraf Mtibaa, and Hayet Brabra. 2016. CSLAOnto: a comprehensive ontological SLA model in cloud computing. *Journal on Data Semantics* 5, 3, 179–193.

[63] Palden Lama and Xiaobo Zhou. 2012. Aroma: Automated resource allocation and configuration of mapreduce environment in the cloud. In *Proceedings of the 9th international conference on Autonomic computing*. ACM, 63–72.

[64] Jonathan Lejeune, Frederico Alvares, and Thomas Ledoux. 2017. Towards a generic autonomic model to manage Cloud Services. In *The 7th International Conference on Cloud Computing and Services Science (CLOSER 2017)*.

[65] Sanping Li, Yu Cao, Simon Tao, Xiaoyan Guo, Zhe Dong, and Ricky Sun. 2015. An extensible framework for predictive analytics on cost and performance in the cloud. In *2015 International Conference on Cloud Computing and Big Data (CCBD)*. IEEE, 13–20.

[66] Norman Lim, Shikharesh Majumdar, and Peter Ashwood-Smith. 2014. A constraint programming-based resource management technique for processing MapReduce jobs with SLAs on clouds. In *Parallel Processing (ICPP), 2014 43rd International Conference on*. IEEE, 411–421.

[67] Norman Lim, Shikharesh Majumdar, and Peter Ashwood-Smith. 2014. Engineering resource management middleware for optimizing the performance of clouds processing mapreduce jobs with deadlines. In *Proceedings of the 5th ACM/SPEC international conference on Performance engineering*. ACM, 161–172.

[68] Norman Lim, Shikharesh Majumdar, and Peter Ashwood-Smith. 2014. Resource management techniques for handling requests with service level agreements. In *International Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPECTS 2014)*. IEEE, 618–625.

[69] Norman Lim, Shikharesh Majumdar, and Peter Ashwood-Smith. 2017. MRCP-RM: a technique for resource allocation and scheduling of MapReduce jobs with deadlines. *IEEE Transactions on Parallel and Distributed Systems* 28, 5, 1375–1389.

[70] Norman Lim, Shikharesh Majumdar, and Peter Ashwood-Smith. 2017. A Run Time Technique for Handling Error in User-Estimated Execution Times on Systems Processing MapReduce Jobs with Deadlines. In *Future Internet of Things and Cloud (FiCloud), 2017 IEEE 5th International Conference on*. IEEE, 1–9.

[71] Norman Lim, Shikharesh Majumdar, and Peter Ashwood-Smith. 2017. Techniques for Handling Error in User-estimated Execution Times During Resource Management on Systems Processing MapReduce Jobs. In *Proceedings of the 17th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing*. IEEE Press, 788–793.

[72] Fotios K Liotopoulos and Petros Lampsas. 2015. Energy-efficient simulation and performance evaluation of large-scale data centers. In *2015 IEEE International Conference on Industrial Technology (ICIT)*. IEEE, 3121–3127.

[73] Qinghua Lu, Shanshan Li, Weishan Zhang, and Lei Zhang. 2016. A genetic algorithm-based job scheduling model for big data analytics. *EURASIP journal on wireless communications and networking* 2016, 1, 152.

[74] Qinghua Lu, Zheng Li, Weishan Zhang, and Laurence T Yang. 2017. Autonomic deployment decision making for big data analytics applications in the cloud. *Soft Computing* 21, 16, 4501–4512.

[75] Yang Lu. 2017. Industry 4.0: A survey on technologies, applications and open research issues. *Journal of Industrial Information Integration* 6, 1–10.

[76] Mohammad-Hossein Malekloo, Nadjia Kara, and May El Barachi. 2018. An energy efficient and SLA compliant approach for resource allocation and consolidation in cloud computing environments. *Sustainable Computing: Informatics and Systems* 17, 9–24.

[77] Xijun Mao, Chunlin Li, Wei Yan, and Shumeng Du. 2016. Optimal Scheduling Algorithm of MapReduce Tasks Based on QoS in the Hybrid Cloud. In *Parallel and Distributed Computing, Applications and Technologies (PDCAT), 2016 17th International Conference on.* IEEE, 119–124.

[78] Lena Mashayekhy, Mahyar Movahed Nejad, Daniel Grosu, Quan Zhang, and Weisong Shi. 2015. Energy-aware scheduling of mapreduce jobs for big data applications. *IEEE Transactions on Parallel & Distributed Systems* 1, 1–1.

[79] Michael Mattess, Rodrigo N Calheiros, and Rajkumar Buyya. 2013. Scaling mapreduce applications across hybrid clouds to meet soft deadlines. In *2013 IEEE 27th International Conference on Advanced Information Networking and Applications (AINA).* IEEE, 629–636.

[80] Rizwan Mian, Patrick Martin, and Jose Luis Vazquez-Poletti. 2013. Provisioning data analytic workloads in a cloud. *Future Generation Computer Systems* 29, 6, 1452–1458.

[81] Akram Mohamadi and Sedigheh Barani. 2015. A review on approaches in service level agreement in cloud computing environment. In *2015 4th Iranian Joint Congress on Fuzzy and Intelligent Systems (CFIS).* IEEE, 1–4.

[82] Saad Mubeen, Sara Abbaspour Asadollah, Alessandro V Papadopoulos, Mohammad Ashjaei, Hongyu Pei-Breivold, and Moris Behnam. 2017. Management of Service Level Agreements for Cloud Services in IoT: A Systematic Mapping Study. *IEEE Access.*

[83] Lekha R Nair, Sujala D Shetty, and Siddhanth D Shetty. 2018. Applying spark based machine learning model on streaming big data for health status prediction. *Computers & Electrical Engineering* 65, 393–399.

[84] Dimas C Nascimento, Carlos Eduardo Pires, and Demetrio Mestre. 2015. Data quality monitoring of cloud databases based on data quality slas. In *Big-Data Analytics and Cloud Computing.* Springer, 3–20.

[85] Deveeshree Nayak, Venkata Swamy Martha, David Threm, Srini Ramaswamy, Summer Prince, and Günter Fahrnberger. 2015. Adaptive scheduling in the cloud-SLA for Hadoop job scheduling. In *2015 Science and Information Conference (SAI).* IEEE, 832–837.

[86] Mihaela-Catalina Nita, Cristian Chilipirea, Ciprian Dobre, and Florin Pop. 2013. A SLA-based method for big-data transfers with multi-criteria optimization constraints for IaaS. In *2013 11th RoEduNet International Conference.* IEEE, 1–6.

[87] Mihaela-Catalina Nita, Florin Pop, Cristiana Voicu, Ciprian Dobre, and Fatos Xhafa. 2015. MOMTH: multi-objective scheduling algorithm of many tasks in Hadoop. *Cluster computing* 18, 3, 1011–1024.

[88] Yoori Oh, Jieun Choi, Eunjung Song, Moonji Kim, and Yoonhee Kim. 2016. A SLA-based Spark cluster scaling method in cloud environment. In *2016 18th Asia-Pacific Network Operations and Management Symposium (APNOMS).* IEEE, 1–4.

[89] Ahmed Oussous, Fatima-Zahra Benjelloun, Ayoub Ait Lahcen, and Samir Belfkih. 2017. Big Data technologies: A survey. *Journal of King Saud University-Computer and Information Sciences.*

[90] Balaji Palanisamy, Aameek Singh, and Ling Liu. 2015. Cost-effective resource provisioning for mapreduce in a cloud. *IEEE Transactions on Parallel and Distributed Systems* 26, 5, 1265–1279.

[91] Tadeusz Pankowski. 2015. Consistency and availability of Data in replicated NoSQL databases. In *Evaluation of Novel Approaches to Software Engineering (ENASE), 2015 International Conference on.* IEEE, 102–109.

[92] Adrian Paschke and Elisabeth Schnappinger-Gerull. 2006. A Categorization Scheme for SLA Metrics. *Service Oriented Electronic Commerce* 80, 25-40, 14.

[93] Jorda Polo, Yolanda Becerra, David Carrera, Malgorzata Steinder, Ian Whalley, Jordi Torres, and Eduard Ayguade. 2013. Deadline-based MapReduce workload management. *IEEE Transactions on Network and Service Management* 10, 2, 231–244.

[94] K Hima Prasad, Tanveer A Faruquie, L Venkata Subramaniam, Mukesh Mohania, and Girish Venkatachaliah. 2010. Resource allocation and SLA determination for large data processing services over cloud. In *2010 IEEE International Conference on Services Computing.* IEEE, 522–529.

[95] Xuanjia Qiu, Wai Leong Yeow, Chuan Wu, and Francis CM Lau. 2013. Cost-minimizing preemptive scheduling of mapreduce workloads on hybrid clouds. In *2013 IEEE/ACM 21st International Symposium on Quality of Service (IWQoS).* IEEE, 1–6.

[96] Joy Rahman and Palden Lama. 2017. MPLEX: In-Situ Big Data Processing with Compute-Storage Multiplexing. In *Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS), 2017 IEEE 25th International Symposium on.* IEEE, 43–52.

[97] B Kezia Rani and A Vinaya Babu. 2015. Scheduling of Big Data application workflows in cloud and inter-cloud environments. In *2015 IEEE International Conference on Big Data (Big Data).* IEEE, 2862–2864.

[98] Rajiv Ranjan, Joanna Kolodziej, Lizhe Wang, and Albert Y Zomaya. 2015. Cross-layer cloud resource configuration selection in the big data era. *IEEE Cloud Computing* 3, 16–22.

[99] Radhya Sahal, Mohamed H Khafagy, and Fatma A Omara. 2016. A Survey on SLA Management for Cloud Computing and Cloud-Hosted Big Data Analytic Applications. *International Journal of Database Theory and Application* 9, 4, 107–118.

[100] Prasan Kumar Sahoo, Suvendu Kumar Mohapatra, and Shih-Lin Wu. 2018. SLA based healthcare big data analysis and computing in cloud network. *J. Parallel and Distrib. Comput.* 119, 121–135.

[101] Sherif Sakr and Anna Liu. 2012. Sla-based and consumer-centric dynamic provisioning for cloud databases. In *Cloud Computing (CLOUD), 2012 IEEE 5th International Conference on*. IEEE, 360–367.

[102] Omran Saleh, Francis Gropengieβer, Heiko Betz, Waseem Mandarawi, and Kai-Uwe Sattler. 2013. Monitoring and autoscaling IaaS clouds: a case for complex event processing on data streams. In *Proceedings of the 2013 IEEE/ACM 6th International Conference on Utility and Cloud Computing*. IEEE Computer Society, 387–392.

[103] Rajinder Sandhu and Sandeep K Sood. 2015. Scheduling of big data applications on distributed cloud based on QoS parameters. *Cluster Computing* 18, 2, 817–828.

[104] Carla Sauvanaud, Mohamed Kaâniche, Karama Kanoun, Kahina Lazri, and Guthemberg Da Silva Silvestre. 2018. Anomaly Detection and Diagnosis for Cloud services: Practical experiments and lessons learned. *Journal of Systems and Software* 139 (2018), 84–106.

[105] Damián Serrano, Sara Bouchenak, Yousri Kouki, Frederico Alvares de Oliveira Jr, Thomas Ledoux, Jonathan Lejeune, Julien Sopena, Luciana Arantes, and Pierre Sens. 2016. SLA guarantees for cloud services. *Future Generation Computer Systems* 54, 233–246.

[106] Damián Serrano, Sara Bouchenak, Yousri Kouki, Thomas Ledoux, Jonathan Lejeune, Julien Sopena, Luciana Arantes, and Pierre Sens. 2013. Towards qos-oriented sla guarantees for online cloud services. In *2013 13th IEEE/ACM International Symposium on Cluster, Cloud, and Grid Computing*. IEEE, 50–57.

[107] M Omair Shafiq and Eric Torunski. 2017. Towards MapReduce based Bayesian deep learning network for monitoring big data applications. In *Big Data (Big Data), 2017 IEEE International Conference on*. IEEE, 2112–2121.

[108] Jin Shao and Qianxiang Wang. 2011. A performance guarantee approach for cloud applications based on monitoring. In *2011 35th IEEE Annual Computer Software and Applications Conference Workshops*. IEEE, 25–30.

[109] Yanling Shao, Chunlin Li, Wenyong Dong, and Yunchang Liu. 2016. Energy-aware dynamic resource allocation on Hadoop YARN cluster. In *High Performance Computing and Communications; IEEE 14th International Conference on Smart City; IEEE 2nd International Conference on Data Science and Systems (HPCC/SmartCity/DSS), 2016 IEEE 18th International Conference on*. IEEE, 364–371.

[110] Yanling Shao, Chunlin Li, Jinguang Gu, Jing Zhang, and Youlong Luo. 2018. Efficient jobs scheduling approach for big data applications. *Computers & Industrial Engineering* 117, 249–261.

[111] Bikash Sharma, Timothy Wood, and Chita R Das. 2013. Hybridmr: A hierarchical mapreduce scheduler for hybrid data centers. In *2013 IEEE 33rd International Conference on Distributed Computing Systems*. IEEE, 102–111.

[112] Mingruo Shi and Ruiping Yuan. 2015. Mad: A monitor system for big data applications. In *International Conference on Intelligent Science and Big Data Engineering*. Springer, 308–315.

[113] Ming-Hung Shih and J Morris Chang. 2017. Design and analysis of high performance crypt-NoSQL. In *Dependable and Secure Computing, 2017 IEEE Conference on*. IEEE, 52–59.

[114] Kwang Mong Sim. 2006. A survey of bargaining models for grid resource allocation. *ACM SIGecom Exchanges* 5, 5, 22–32.

[115] Kwang Mong Sim. 2010. Grid resource negotiation: survey and new directions. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* 40, 3, 245–257.

[116] Mbarka Soualhia, Foutse Khomh, and Sofiène Tahar. 2017. Task scheduling in big data platforms: a systematic literature review. *Journal of Systems and Software* 134, 170–189.

[117] Andre Abrantes DP Souza and Marco AS Netto. 2015. Using application data for sla-aware auto-scaling in cloud environments. In *Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS), 2015 IEEE 23rd International Symposium on*. IEEE, 252–255.

[118] Xi Sun, Bo Gao, Liya Fan, and Wenhao An. 2012. A cost-effective approach to delivering analytics as a service. In *Web services (icws), 2012 ieee 19th international conference on*. IEEE, 512–519.

[119] Yangyang Tao, Shucheng Yu, and Junxiu Zhou. 2018. Information Flow Queue Optimization in EC Cloud. In *2018 International Conference on Computing, Networking and Communications (ICNC)*. IEEE, 888–892.

[120] Fengguang Tian and Keke Chen. 2011. Towards optimal resource provisioning for running mapreduce programs in public clouds. In *Cloud Computing (CLOUD), 2011 IEEE International Conference on*. IEEE, 155–162.

[121] Rafael Tolosana-Calasanz, José Ángel Bañares, Congduc Pham, and Omer F Rana. 2016. Resource management for bursty streams on multi-tenancy cloud environments. *Future Generation Computer Systems* 55, 444–459.

[122] Linjiun Tsai, Hubertus Franke, Chung-Sheng Li, and Wanjiun Liao. 2018. Learning-Based Memory Allocation Optimization for Delay-Sensitive Big Data Processing. *IEEE Transactions on Parallel and Distributed Systems* 29, 6, 1332–1341.

[123] Radu Tudoran, Olivier Nano, Ivo Santos, Alexandru Costan, Hakan Soncu, Luc Bougé, and Gabriel Antoniu. 2014. Jetstream: Enabling high performance event streaming across cloud data-centers. In *Proceedings of the 8th ACM International Conference on Distributed Event-Based Systems*. ACM, 23–34.

[124] Abhishek Verma, Ludmila Cherkasova, and Roy H Campbell. 2011. ARIA: automatic resource inference and allocation for mapreduce environments. In *Proceedings of the 8th ACM international conference on Autonomic computing*. ACM, 235–244.

[125] Chen Wang, Junliang Chen, Bing Bing Zhou, and Albert Y Zomaya. 2012. Just satisfactory resource provisioning for parallel applications in the cloud. In *2012 IEEE Eighth World Congress on Services*. IEEE, 285–292.

[126] Guanying Wang, Ali R Butt, Prashant Pandey, and Karan Gupta. 2009. Using realistic simulation for performance analysis of mapreduce setups. In *Proceedings of the 1st ACM workshop on Large-Scale system and application performance*. ACM, 19–26.

[127] Meisong Wang, Rajiv Ranjan, Prem Prakash Jayaraman, Peter Strazdins, Pete Burnap, Omer Rana, and Dimitrios Georgakopulos. 2015. A Case for Understanding End-to-End Performance of Topic Detection and Tracking Based Big Data Applications in the Cloud. In *International Internet of Things Summit*. Springer, 315–325.

[128] Yang Wang and Wei Shi. 2013. On optimal budget-driven scheduling algorithms for MapReduce jobs in the hetereogeneous cloud. *Technical Report TR-13–02, Carleton Univ.* (2013).

[129] Yang Wang and Wei Shi. 2014. Budget-driven scheduling algorithms for batches of MapReduce jobs in heterogeneous clouds. *IEEE Transactions on Cloud Computing* 2, 3, 306–319.

[130] Jonathan Stuart Ward and Adam Barker. 2014. Observing the clouds: a survey and taxonomy of cloud monitoring. *Journal of Cloud Computing* 3, 1, 24.

[131] Md Whaiduzzaman, Mohammad Nazmul Haque, Md Rejaul Karim Chowdhury, and Abdullah Gani. 2014. A study on strategic provisioning of cloud computing services. *The Scientific World Journal* 2014.

[132] Philipp Wieder, Jan Seidel, Oliver Wäldrich, Wolfgang Ziegler, and Ramin Yahyapour. 2008. Using sla for resource management and scheduling-a survey. In *Grid Middleware and Services*. Springer, 335–347.

[133] Xiaoyong Xu, Maolin Tang, and Yu-Chu Tian. 2016. Theoretical results of QoS-guaranteed resource scaling for cloud-based MapReduce. *IEEE Transactions on Cloud Computing*.

[134] Xiaoyong Xu, Maolin Tang, and Yu-Chu Tian. 2018. QoS-guaranteed resource provisioning for cloud-based MapReduce in dynamical environments. *Future Generation Computer Systems* 78, 18–30.

[135] Chaowei Yang, Qunying Huang, Zhenlong Li, Kai Liu, and Fei Hu. 2017. Big Data and cloud computing: innovation opportunities and challenges. *International Journal of Digital Earth* 10, 1, 13–53.

[136] Zhihao Yao, Ioannis Papapanagiotou, and Robert D Callaway. 2014. SLA-aware resource scheduling for cloud storage. In *Cloud Networking (CloudNet), 2014 IEEE 3rd International Conference on*. IEEE, 14–19.

[137] S Yasmin and S Jessica Sritha. 2017. A constraint programming-based resource allocation and scheduling of map reduce jobs with service level agreement. In *2017 International Conference on Energy, Communication, Data Analytics and Soft Computing (ICECDS)*. IEEE, 3589–3594.

[138] Abdulsalam Yassine, Ali Asghar Nazari Shirehjini, and Shervin Shirmohammadi. 2016. Bandwidth on-demand for multimedia big data transfer across geo-distributed cloud data centers. *IEEE Transactions on Cloud Computing*.

[139] Xiaoqun Yuan, Geyong Min, Laurence T Yang, Yi Ding, and Qing Fang. 2017. A game theory-based dynamic resource allocation strategy in geo-distributed datacenter clouds. *Future Generation Computer Systems* 76, 63–72.

[140] Bernard P Zeigler, Tag Gon Kim, and Herbert Praehofer. 2000. *Theory of modeling and simulation*. Academic press.

[141] Xuezhi Zeng, Saurabh Garg, Zhenyu Wen, Peter Strazdins, Lizhe Wang, and Rajiv Ranjan. 2016. SLA-aware scheduling of map-Reduce applications on public clouds. In *2016 IEEE 18th International Conference on High Performance Computing and Communications; IEEE 14th International Conference on Smart City; IEEE 2nd International Conference on Data Science and Systems (HPCC/SmartCity/DSS)*. IEEE, 655–662.

[142] Xuezhi Zeng, Saurabh Kumar Garg, Zhenyu Wen, Peter Strazdins, Albert Y Zomaya, and Rajiv Ranjan. 2017. Cost efficient scheduling of MapReduce applications on public clouds. *Journal of computational science*.

[143] Rui Zhang, Reshu Jain, Prasenjit Sarkar, and Lukas Rupprecht. 2014. Getting your big data priorities straight: a demonstration of priority-based qos using social-network-driven stock recommendation. *Proceedings of the VLDB endowment* 7, 13, 1665–1668.

[144] Liang Zhao, Sherif Sakr, and Anna Liu. 2015. A framework for consumer-centric SLA management of cloud-hosted databases. *IEEE Transactions on Services Computing* 8, 4, 534–549.

[145] Yali Zhao, Rodrigo N Calheiros, James Bailey, and Richard Sinnott. 2016. SLA-based profit optimization for resource management of big data analytics-as-a-service platforms in cloud computing environments. In *2016 IEEE International Conference on Big Data (Big Data)*. IEEE, 432–441.

[146] Yali Zhao, Rodrigo N Calheiros, Graeme Gange, Kotagiri Ramamohanarao, and Rajkumar Buyya. 2015. SLA-based resource scheduling for big data analytics as a service in cloud computing environments. In *2015 44th International Conference on Parallel Processing*. IEEE, 510–519.

[147] Qin Zheng. 2010. Improving MapReduce fault tolerance in the cloud. In *2010 IEEE International Symposium on Parallel & Distributed Processing, Workshops and Phd Forum (IPDPSW)*. IEEE, 1–6.

[148] Liudong Zuo and Michelle M Zhu. 2015. Concurrent bandwidth reservation strategies for big data transfers in high-performance networks. *IEEE Transactions on Network and Service Management* 12, 2, 232–247.

SLA Management for Big Data Analytical Applications in Clouds: A Taxonomy Study

XUEZHI ZENG, Australian National University, Australia
SAURABH GARG, University of Tasmania, Australia
MUTAZ BARIKA, University of Tasmania, Australia
ALBERT Y. ZOMAYA, University of Sydney, Australia
LIZHE WANG, China University of Geoscience (Wuhan), China
MASSIMO VILLARI, University of Messina, Italy
DAN CHEN, Wuhan University, China
RAJIV RANJAN, China University of Geoscience (Wuhan, China) and Newcastle University, UK

Recent years have witnessed the booming of big data analytical applications (BDAAs). This trend provides unrivalled opportunities to reveal the latent patterns and correlations embedded in the data thus productive decisions may be made. This was previously a grand challenge due to the notoriously high dimensionality and scale of big data while the quality of service (QoS) offered by providers is the first priority. As BDAAs are routinely deployed on Clouds with great complexities & uncertainties, it is a critical task to manage the service level agreements (SLAs) thus a high QoS can then be guaranteed. This study performs a systematic literature review (SLR) of the state-of-the-art of SLA-specific management for Cloud-hosted BDAAs. The review surveys the challenges and contemporary approaches along this direction centering on SLA. A research taxonomy is proposed to formulate the results of the systematic literature review. A new conceptual SLA model is defined and a multi-dimensional categorization scheme is proposed on its basis to apply the SLA metrics for 1) an in-depth understanding of managing SLAs and 2) the motivation of trends for future research.

CCS Concepts: • **Computer systems organization** → **Embedded systems**; *Redundancy*; Robotics; • **Networks** → Network reliability.

Additional Key Words and Phrases: Big Data; Big Data Analytics Application; Service Level Agreement; Service Layer; SLA Metrics; SLA

ACM Reference Format:
Xuezhi Zeng, Saurabh Garg, Mutaz Barika, Albert Y. Zomaya, Lizhe Wang, Massimo Villari, Dan Chen, and Rajiv Ranjan. 2019. SLA Management for Big Data Analytical Applications in Clouds: A Taxonomy Study. 1, 1 (November 2019), 35 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

Authors' addresses: Xuezhi Zeng, Australian National University, Research School of Computer Science, Australia; Saurabh Garg, University of Tasmania, School of TED, Australia; Mutaz Barika, University of Tasmania, School of TED, Australia; Albert Y. Zomaya, University of Sydney, School of IT, Australia; Lizhe Wang, China University of Geoscience (Wuhan), School of Computer Science, China; Massimo Villari, University of Messina, Italy; Dan Chen, Wuhan University, School of Computer Science, China; Rajiv Ranjan, China University of Geoscience (Wuhan, China) and Newcastle University, UK.

XXXX-XXXX/2019/11-ART $15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 Introduction

Recent years have witnessed the booming of BDAAs in Clouds [1, 2, 13, 19, 89, 135, 143]. For example, Google utilizes Google BigQuery [5] to offer inventory management system [2], an abundant, highly scalable, low cost and pay-as-you-go Cloud-hosted BDAA to make inventory management productive and efficient. Amazon provides natural language processing-based BDAA in Cloud that identifies the language of voluminous texts, extracts vital entities such as people, organizations, locations or events, and analyze sentiments in texts using Amazon Comprehend [1]. Salesforce builds their social media monitoring service in their marketing Cloud [3] that can collect social media data in close to real-time and run in it through their underpinning big data technologies and algorithms to produce insights such as near-instant feedback on the effectiveness of new marketing campaigns or alerts about emerging problems with products. These applications offer organizations the capabilities of constructing valuable information and extracting actionable insight for enhancing the evidence-based decision-making process. Many leading providers such as Google and Amazon provision such analytics capabilities in the form of service to customers in a pay-per-use economic model.

In today's competitive world, the potential business values of these applications depend a lot on the quality of service (QoS) offered by providers. Hence, to gain competitive advantages, providers should focus on the needs of their customers and respond proactively to their marketing strategies, not only to build and raise customers awareness of their services but also meet customers' best expectations for service quality. That is to say, providers must provide the required and promised services to their customers, and these services must achieve the requirements of users (ex. availability, elasticity and scalability).

Given these circumstances, it is very important and necessary for efficient methods to manage and guarantee the QoS promised. Service Level Agreement (SLA) represents a formal contract among service providers and customers, which captures agreements in the sense of QoS. SLAs play an integral role in governing the relationships between providers and customers in the context of Cloud-hosted BDAAs. Furthermore, SLAs can be considered as a strong differentiator, which allows service providers to provide various levels of guarantees for services offered to customers as well as to distinguish itself from competitors. Therefore, it has become a critical task to manage the SLAs thus a high QoS of Cloud-hosted BDAAs may then be guaranteed.

Existing surveys focus on SLAs management in grid computing (for Sim [115], Sim [114] and Wieder et al. [132]) or Cloud computing (for Mohamadi et al. [81], Faniyi et al. [33], Hussain et al. [49] and Whaiduzzaman et al. [131]). Internet of Things (IoT) emerges with the recent advancements in computing. Mubeen et al. [82] investigated the existing work on SLAs management for IoT-based applications in Clouds. To the best of our knowledge, there exists only one preliminary review with a simple taxonomy of SLA management for Cloud computing and Cloud-based BDAAs [99], and it does not suffice in any in-depth understanding of managing SLAs or the trends for future research in this area. SLA-specific management for BDAAs in Clouds has largely been ignored.

To bridge gap in this field and spot out the trends for future work, this study performs a systematic literature review (SLR) of the state-of-the-art of SLA-specific management for Cloud-hosted BDAAs. The review mainly concerns the requirements and characteristics across Cloud computing stacks. In particular, a taxonomy-based study is emphasized for the following reasons:

- The existing works on SLA management for Cloud-hosted BDAAs manifest a wide range of thematic perspectives (e.g., techniques used, Cloud models deployed, and layers considered). Further, in each different perspective, various subcategories have been discussed. Take the technique perspective as an example, researchers proposed multiple techniques to address SLA management for Cloud-hosted BDAAs (e.g., simulation and machine learning). It is important to form a hierarchy of these categories for a comprehensive understanding.

- Taxonomy is a scientific tool that is able to provide a global and universal categorization addressing the above needs and continues effectively to accommodate new knowledge when applicable. A well-developed taxonomy of SLA management for Cloud-hosted BDAAs aims to support researchers and practitioners from academia and industry by organizing SLA-specific management concepts and terminologies for Cloud-hosted BDAAs. It provides researchers with a scientific tool to focus on all the aspects bridging research gaps.
- There exists no taxonomy enabling an extensive review of SLA management for Cloud-hosted BDAAs. This is evidenced from the aforementioned brief of existing survey works (more details can be seen in section 2). This study formulates a taxonomy categorize the existing research works from multiple perspectives and then to enable readers a better understanding of the state-of-the-art.

This paper aims not only to present researchers an outlook on SLA-specific management for BDAAs in Clouds, but also to give new insights through a global thematic taxonomy in this research area. The main contributions of this survey include:

- A systematic literature review of SLA Management for Cloud-hosted BDAAs with build-up thematic taxonomy covering six core dimensions including actors, Service layers, techniques, Cloud service and deployment models, SLA metrics and conceptualization;
- A unified SLA model for Cloud-hosted BDAAs from a layer-based perspective to link different types of SLAs in a vertical motion;
- A multi-dimensional categorization scheme regarding SLA metrics dedicated for Cloud-hosted BDAAs, which allows systematically categorization of both common metrics and niche metrics for each layer with respect to requirements of BDDAs; An SLA template is provided for a representative Cloud-hosted BDAA to aid understanding SLAs conversation across its different layers.
- Identification of open issues and future directions of Cloud-hosted BDAA based on the systematic literature review.

The rest of this paper is organized as follows. Section 2 discusses the related works and our motivations to construct a taxonomy-based survey. In Section 3 discusses how the SLR applies to the research field of SLA management for Cloud-hosted BDAAs and proposes a novel thematic taxonomy. Section 4 presents the details of the review results and discusses the findings according to the taxonomy. Section 5 presents a dedicated conceptual SLA model and proposes a multi-dimensional categorization scheme of SLA metrics for BDAAs in Clouds and give an illustrative example of SLAs template for a real Cloud-hosted BDAA. We conclude the paper with open issues and future directions in Section 6.

2 Related Work

To figure-out the considerable difference between our survey and the previous studies in the literature, we present in this section the related research works that have been done by others and highlight their limitations.

A few previous studies focused on the literature review of SLA management in the context of Cloud environment. Mohamadi et al. [81] conducted a very preliminary review of SLA management approaches and compared them with respect to improved parameters, implementation/simulation and its environment, and workload/application. Faniyi et al. [33] surveyed the research landscape of SLA-based Cloud systematically with the focus on specific phase of SLA life cycle (i.e. resource allocation phase) and outlining consequences on the others. From Cloud service provider perspective with small to medium-sized enterprise level. Hussain et al. [49] presented a comprehensive overview of existing approaches of SLA management in Clouds and highlighted the features and limitations of these approaches to tackle the issue of creating a viable SLAs in Cloud computing from the viewpoint of service provider's. While from another perspective, Whaiduzzaman et al. [131] focused on

SLA-based service provisioning techniques and methods that assist in evaluating Cloud services provisioned with regards to user-specific requirements and cost.

With the recent advancements in computing, internet of thing (IoT) has been introduced as an emerging and promising technology. Thus, Saad Mubeen et al. [82] conducted a survey to investigate the existing research on SLAs management for IoT-based applications in Clouds. This survey used a systematic mapping study for the purpose of identifying the results of the published research works that are related to SLA management in IoT context.

Existing surveys either focus on SLA management in Cloud environment or IoT environment, which are not specific and sufficient for SLA management in the context of Cloud-hosted BDAAs. To the best of our knowledge, there is only one research work has been done on SLA management for BDAAs in Clouds. Sahal et al. [99] conducted a preliminary survey and divided SLA management into two types, which are SLA management for Cloud computing and SLA management for Cloud-hosted BDAAs. Regarding to latter type, SLA management approaches are categorized into two groups comprising MapReduce scheduler and Cloud Layer, which we argue that this categorization is oversimplified and fails to give a holistic landscape towards SLA management for BDAAs in Clouds.

From the above existing research works, it is clearly seen that a taxonomy-based survey on SLA management for Cloud-hosted BDAAs is in its infancy. Therefore, we conduct a taxonomic survey in this field by using a systematic literature review (SLR) method to fill this research gap. Our work differentiate existing research works in multiple dimensions: (i) we propose a novel thematic taxonomy that covers six core dimensions including actors, Service layers, techniques, Cloud service and deployment models, SLA metrics and conceptualization; (ii) we design a cross-layer SLA model for Cloud-hosted BDAAs (CL-SLAfBDAAs) to provide a unified and structured way to understand SLAs at different layers with different attributes and their strong dependencies relationship. (iii) we propose a general categorization scheme of SLA metrics consisting of common metrics and niche metrics for each different layer of SLA, which fully embodies the characteristics of Cloud-hosted BDAAs; (iv) we elaborate our proposed model and SLA metrics categorization by giving an example of SLA template for a real Cloud-hosted BDAA. Our work not only provides a comprehensive review of the state-of-the-art landscape, but also finds insights into understanding the research themes/patterns in this field.

3 Systematic Literature Review Of SLA Management in Cloud-Hosted BDAAs

The field of SLA management is broad. It involves several phases that depicted into SLA life cycle [33] [49] to clarify expectations and responsibilities, and streamline communications among the parities involved in the agreement. In [33], five phases of the SLA life cycle are described. The first phase is SLA negotiation to define and agree on given service terms and levels. The second phase is SLA establishment/deployment to implement and deliver the service in accordance to the agreed SLA. The third phase is to SLA monitoring to observe and monitor the service after being deployed and under its execution. The fourth phase is violation management to detect and manage SLA violations. The fifth phase is SLA reporting and termination to provide detailed reports for audit activities that happened during service provisioning, and provide a method to terminate the service at the end of SLA agreement or in case of violations as defined in the agreed SLA. From the aforementioned SLA life cycle and after studying the characteristics of Cloud-based BDAAs, we identify the following requirements for adopting SLA management for Cloud-based BDAAs:

- Cloud provider and customer responsibilities. In SLA management, different parities are involved. Thus, each party role, obligation and penalty in the context of Cloud-based BDAAs should be stated.
- SLA description and commitments. During the establishment of agreement, the service terms and levels, and QoS commitments are defined. This includes SLA metrics that will be specified and monitored to ensure that SLA agreement is respected.

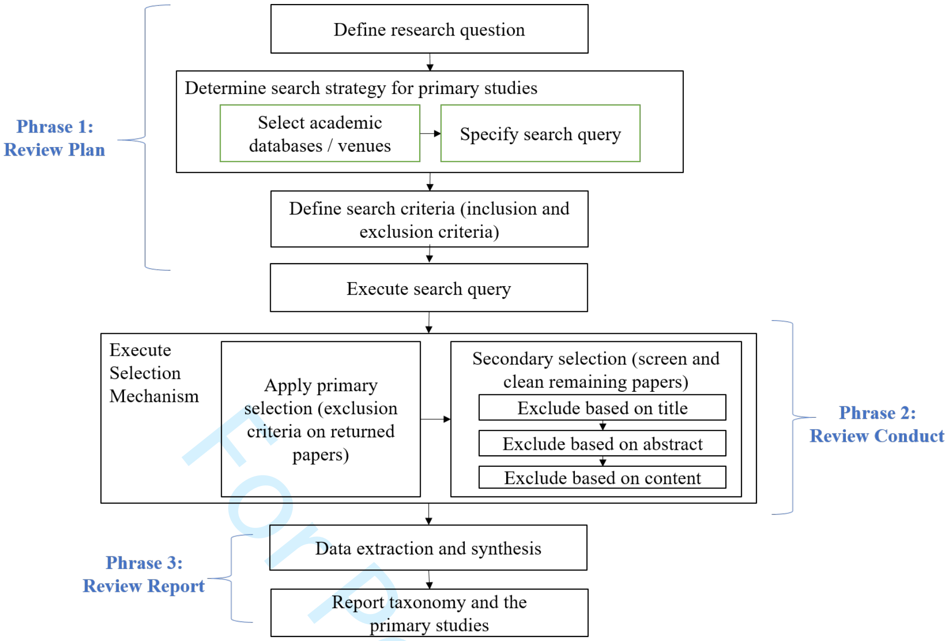
Phrase 1:
Review Plan
Define research question
Determine search strategy for primary studies
Select academic databases / venues
Specify search query
Define search criteria (inclusion and exclusion criteria)
Execute search query
Execute Selection Mechanism
Apply primary selection (exclusion criteria on returned papers)
Secondary selection (screen and clean remaining papers)
Exclude based on title
Exclude based on abstract
Exclude based on content
Phrase 2:
Review Conduct
Phrase 3:
Review Report
Data extraction and synthesis
Report taxonomy and the primary studies

Fig. 1. Systematic literature methodology applied on the survey of SLA management for Cloud-hosted BDAAs

- SLA enforcement. During the deployment of BDAAs in Cloud, different models of Cloud infrastructure and different layers should be considered since the measures and requirements of QoS defined in SLAs differ With each model and layer.
- SLA monitoring and management. After the agreed SLA parameters and metrices are defined, different approaches and techniques to be taken to meet SLA and service providers' procedures to be invoked in the event that SLA guarantees are not fulfilled. Also, policies for applying compensation and penalty are taken place in case of un-fulfilment of SLA terms.

To meet the above requirements, we deliberately intended to cover multiple categories and subcategories in order to acquire a comprehensive understanding of SLA management for Cloud-based BDAAs. To achieve that, we need a a taxonomy-based study by leveraging systematic literature review (SLR) method that provides rigorous way of reviewing the landscape towards SLA-specific management for BDAAs in Cloud to develop comprehensive taxonomy as a key result of this survey work. Our study in this paper uses a SLR methodology proposed by Kitchenham et al. [58] that is an objective, transparent and reproducible methodological method of reviewing extant literature to answer and deduce particular research question(s) in such a way that is unprejudiced. Specifically, our SLR consists of three main phrases as shown in Figure 1.

The first phase is planning for review, where we define our research questions that will be considered and tackled in this study and determine the strategy of search being used for primary studies with lists of inclusion and exclusion criteria. After the executing search query on the selected database sources, the relevant papers obtained will be input for the next phase. The second phase is about conducting the review by applying the selection mechanism including primary selection (exclusion and inclusion criteria) on the obtained papers to get all relevant papers, and the secondary selection for performing a further evaluation for each remaining paper to get the most relevant papers. In the final phase, we further analyze the remaining papers to report a thematic taxonomy of SLA management for BDAAs in Clouds and review these papers based on this taxonomy.

Table 1. Academic database sources

Source	URL
IEEE Explore Digital Library	http://ieeexplore.ieee.org
ACM Digital Library	http://portal.acm.org
Springer	http://springerlink.com
Science Direct	http://sciencedirect.com
Web of Science	http://webofknowledge.com
Google Scholar	http://scholar.google.com

3.1 Research Questions

This study aimed to portray the research landscape in SLA management for BDAAs in Clouds by addressing the following research questions:

- *RQ1*: What are the actors involved in making conversations and engineering SLAs in the context of Cloud-hosted BDAAs?
- *RQ2*: What is the status of addressing SLA management for Cloud-hosted BDAAs from different service layers (i.e. BDSaaS, BDPaaS and CIaaS) and Cloud deployment models (i.e. private, public, hybrid and community Cloud)?
- *RQ3*:What are the techniques applied to address SLA management for Cloud-hosted BDAAs?
- *RQ4*: What SLA metrics are of interest to stakeholders and being discussed in the context of Cloud-hosted BDAAs?
- *RQ5*: To what extent the SLA model for Cloud-hosted BDAAs is conceptualized?

3.2 Search and Selection Strategy for Primary Studies

3.2.1 Selection of Academic Databases and Search String To search for research publications in the areas of computer science, there are well-known databases that are being used as primary sources for these publications [58]. For our study, the academic databases selected are shown in Table 1. These databases provide advanced search options with a set of Boolean functions to make concise search based on certain fields such as abstract, title and keywords, which return the most relevant results in comparison to search all fields.

We then construct our search string that will be executed over the aforementioned databases to search relevant publications. To provide comprehensive coverage of the relevant research works in the literature and state-of-the-art studies, we need to select keywords carefully. Thus, we consider the terms "service level agreement", "Quality of Service", "big data" , "big data analytics", "big data analysis" and "big data analytics application" as the primary keywords along with a range of related abbreviations, plural or synonyms, namely "service level agreements", "service-level agreements", "SLA", "SLAs", "SLM", "SLA management", "QoS", "BDA" and "BDAA". Since a typical BDAA comprises data ingestion, data storage, data processing and data analysis framework, we also consider "MapReduce", "batch processing", "batch computing", "stream processing", "stream computing", "data ingestion" and "NoSQL" as additional keywords. To join the primary keywords and the additional keywords with their synonyms in the search string, Boolean functions (AND and OR) are used. Moreover, to make sure that our search string returns as many relevant studies as possible in the selected databases, we conduct several tests. As a result, we select the following search string:

(("service level agreement" OR "service-level agreement" OR "SLA") OR ("service level agreements" OR "SLAs") OR ("SLA management") OR ("service level management" OR "SLM") OR ("SLA conformance") OR ("quality of service" OR "QoS")) AND (("big data") OR ("big data analytics") OR ("big data analysis") OR ("big data analytical application" OR "BDAA") OR ("big data analytical applications" OR "BDAAs") OR ("big data analytics " OR "BDA") OR "MapReduce" OR ("batch processing") OR ("batch computing") OR ("stream processing") OR ("stream computing") OR "NoSQL" OR "ingestion"))

Table 2. List of Inclusion Criteria

Criterion	Description
The type of study is peer-reviewed	We chose peer-reviewed publications including conference/workshop and journal papers, and peer-reviewed book chapters.
The writing language for the study is English	We restrain the language to English because some databases such as Springer returns publications in another language like German
The publication year for the study is published from 2010 to 2018	We search all publications in accordance to our search string that have been published in the databases between 2010 and 2018.

Table 3. List of Exclusion Criteria

Criterion	Description
The focus of study is not SLA management	We consider the publications in the field of SLA management for Cloud-hosted BDAAs.
Abstracts and publications that are not pass through refereeing process	We exclude the study with only abstract, not in the form of full paper and is not peer reviewed.
Duplicate publication	We remove duplication for the same study found in different databases

3.2.2 Search Criteria and Selection Mechanism To evaluate the publications that will be obtained after applying the search string in next phase, we need to define the search criteria that being applied. Table 2 and Table 3 show inclusion and exclusion criteria that are performed in our SLR.

After executing the search string and applying inclusion and exclusion criteria on the obtained relevant publications in the field of SLA management for Cloud-hosted BDAAs, we get the initial result of 1098 papers. Then, we conduct the stringent selection (secondary selection) on them based their titles, abstracts and contents using the following rule. We strictly select publications that consider SLA management specifically in the context of BDAAs in Clouds and exclude those publications that only discuss SLAs in general Cloud computing environment. This is because, our focus in this paper is given to the evolutionary stage of SLAs for Cloud-hosted BDAAs rather than the evolutionary stage of SLAs for Cloud Computing. Moreover, SLA management in Cloud Computing is a well-explored area with lots of papers. However, SLA management for Cloud-hosted BDAAs is still a young research area, which demands more necessity and urgencies study on. At the end of this phase, we get 109 papers that will be systematically reviewed in this study.

3.3 Data Extraction and Synthesis

The quality assessment strategy used in this study is subjective analysis, where we evaluated the collected papers from primary studies to assess their relevance to the landscape of the survey. From this analysis, the thematic taxonomy of SLA management for BDAAs in Clouds has been emerged (see Figure 2). The following are the detailed descriptions for the proposed taxonomy elements:

- Actors – This element considers the different actors (service providers, consumers and Cloud end users) involved in Cloud-hosted BDAAs. Service providers provide the consumers resources that can be provisioned and metered on demand including big data platform resources and Cloud infrastructure. In particular, it could further classified into BDSaaS, BDPaaS and CIaaS providers. These providers care more about the efficient resource utilization, energy efficiency, profit maximization, cost reduction, and performance enhancement. The service consumers are those actors that utilize services offered by service providers and are liable for their resource consumption's, where they are more concerned about the budget, pricing, and customer satisfaction. The Cloud end users (or for short end users) are those actors that use the applications or services offered by service customers. They are more interested in QoS and SLA constraints such as service quality, performance and response time.
- Service Layers – This element examines SLA management from various levels of abstraction including CIaaS, BDPaaS and BDSaas. At CIaaS layer, SLA management takes care of guaranteeing SLA requirements on virtualized resources such as VMs, storage or network. At BDPaaS layer, such management guarantee SLA requirements for different big data frameworks including data ingestion, data processing, data analysis

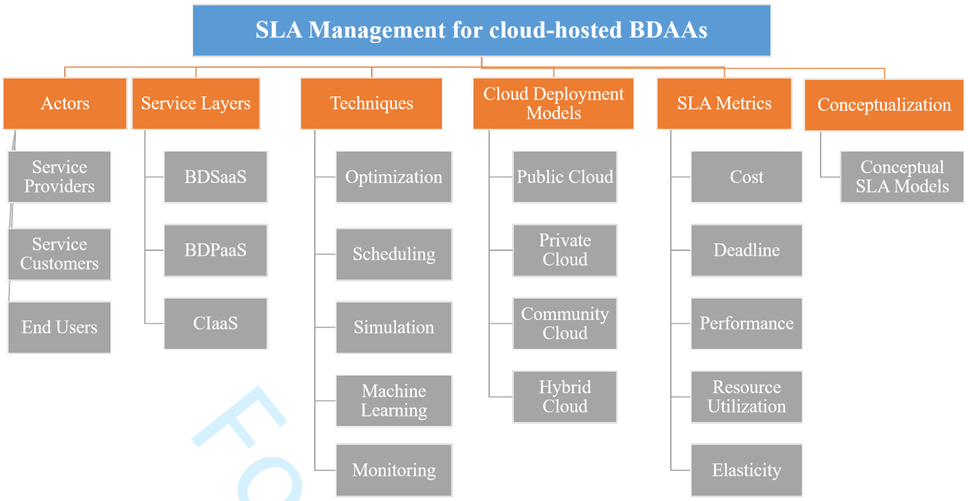

Fig. 2. Taxonomy of SLA management for Cloud-hosted BDAAs

and data storage), and at BDSaaS, such management cares about guaranteeing user-specific application requirements.

- Techniques – In the context of Cloud-hosted BDAAs, one or more techniques (such as optimization-based, scheduling-based and simulation-based) can be included into SLA management to guarantee SLA requirements at specific or across service layers. A technique is a method that aims to address SLA management for Cloud-hosted BDAAs.
- Cloud Deployment Models – When proposing a new SLA management, the activities like implementation, deployment, validation and evaluation come to the picture in order to access the validity and practicality of such management in Cloud infrastructure. Thus, these activities need to be carried-out in private, public, community or/and hybrid Cloud deployment model, where each one of them has its own requirements and challenges.
- SLA Metrics – This element considers SLA items that are defined as quantitative targets in the contract and they must be maintained by the service provided. Measuring these items/metrics (such as cost, deadline and performance) is critical for SLA management to avoid any SLA violation.
- Conceptualization – This element examines SLA management from conceptualization perspective instead of concrete techniques. It mainly consists of designing conceptual SLA models that offer researchers a fundamental and clear way to describe actors, activities and entities involved in a Cloud-hosted BDAA scenario and understand the context of SLA including the conversation and relationship between providers and customers.

4 Review Results in Thematic Taxonomy

4.1 Actors (In response to *RQ1*)

4.1.1 Providers In terms of providers' profit maximization, the authors [29] provide an SLA-based PaaS architecture that can support Cloud-hosted BDAAs. In their paper, a disperse optimization policy is proposed, which aims at maximizing providers' profit and considers to pay penalties incurred when SLA are unsatisfied. Then, the proposed optimization policy is applied to Cloud-hosted BDAAs (e.g., MapReduce applications). In paper [145], the authors designed and implemented automated and elastic resource scheduling algorithms with the objective of profit optimization. Their algorithms can deliver BDAAs to users and optimize profits of platforms while guaranteeing SLAs for query requests in terms of deadlines and budgets and allowing prompt responses with manageable financial costs.

Unlike the above works, paper [18] focuses on the optimization of energy consumption from providers' perspective. The authors take into account sharing MapReduce-based applications in an environment of Hadoop YARN and introduce an SLA-driven energy-saving scheduling algorithm for them [4]. Job profiling is performed to capture the characteristics of performance for diverse stages of a MapReduce-based BDAA. The obtained characteristics of performance will be considered as input to resource provisioning phrase with the purpose of guaranteeing application's SLA such as the completion deadlines. Their experiments demonstrate that their approach enhances the conformance of SLA in terms of reduced energy consumption and resource expenditure.

4.1.2 Customers The authors in [144] focused on their study on Cloud-hosted databases from the customer perspective and addressed the challenge of SLA-driven provisioning and cost management for them. In their paper, a comprehensive framework is proposed, which can flexibly and dynamically provisioning Cloud-hosted database of BDAAs. According to application-defined policies, their proposed framework can satisfy SLAs in terms of performance requirements, avoid penalties when SLA violations occur and control expenses when allocating computing resources.

4.1.3 End Users The authors in [12] proposed an improved resource revenue optimization model. The model defines the constraint mechanism that describes quality of service (QoS) problems. They sliced the requests of end users, modeled the process of requesting service, evaluated the time of response and processing, and allocated resources based on the specified objective function while considering end users' requirements for QoS in this model. They designed a parallel and distributed algorithm based on the working mechanism of MapReduce to solve their proposed model while guaranteeing end users' needs on QoS as much as possible.

4.2 Service Layers (In response to *RQ2*)

Figure 3 presents the statistics of SLA management works in the reviewed papers by the service abstraction level and their breakdown in each layer. It is observed that most of the reviewed papers (67%) fall into the BDPaaS sector. This demonstrates that BDPaaS is the core part of BDAaaS and attracts more interest from researchers. When drilling down into the BDPaaS layer, we found that the top-ranked framework at this layer is data processing with a percentage of 57%. This is because that distributed data processing technologies such as MapReduce receive lots of attention in academia since 2010. Also, it is seen that the data storage framework is ranked secondly, occupying 8%. This indicates that representative data storage technologies such as NoSQL are of increasing interest by researchers in recent years.

Moreover, from the BDSaaS sector, there are 11% reviewed papers discussing SLA management for general BDAAs, while 5% reviewed papers providing domain-specific BDAAs. Interesting, it is further noted that among these domain-specific BDAAs, healthcare application is of the most interest for researchers with four reviewed papers [25, 83, 100, 145] in total and only one reviewed paper takes banking application as the case study [97]. Besides, in CIaaS sector, it is seen that the computing, storage, and network components share the balanced percentage, which means they receive even attention in academia.

Additionally, we present the works of SLA management for Cloud-hosted BDAAs by layers with the corresponding references in Table 4. It has been seen that the quantity of publications regarding batch processing is much higher than that regarding stream processing. The reason is that typical batch processing paradigm like MapReduce featured by its automatic parallelization and distribution, fault tolerance and simplicity becomes ubiquitous programming framework to parallelize the processing of large dataset, which gained significant interest both industry and academia since its emergence in 2007. However, stream processing such as Spark or Storm has been earning improving attention in the last years due to the emerging need for supporting a real-time

Table 4. Classification of the reviewed papers on SLA Management for BDAAs in Clouds by layers

Service Layer (# of papers)	Category (# of papers)	Sub-category and References
BDSaaS (17)	Applications (17)	• General applications: [8, 11, 61, 73, 74, 80, 103, 107, 108, 110, 112, 146] • Domain-specific applications: [25, 83, 97, 100, 145]
BDPaaS	Data Processing Framework (63)	• Batch processing: [10, 12, 16, 18, 29, 31, 35–39, 41, 43, 44, 50, 54, 57, 60, 63, 66–71, 75, 77–79, 85, 87, 88, 90, 93, 95, 105, 106, 109, 111, 120, 124, 125, 128, 129, 133, 134, 137, 141, 142, 145–147] • Stream processing: [15, 40, 42, 46, 51, 59, 83, 117, 121–123]
	Data Storage Framework (9)	[27, 34, 47, 84, 91, 101, 104, 113, 144]
	Data Analysis Framework (2)	[65, 83]
CIaaS	Computing resources (8)	[22, 23, 53, 55, 72, 76, 102, 104]
	Storage resources (6)	[9, 24, 30, 96, 119, 136]
	Network resources (6)	[48, 86, 94, 138, 139, 148]

or near-real-time processing task. The representative works in each layer will be discussed in the following subsections.

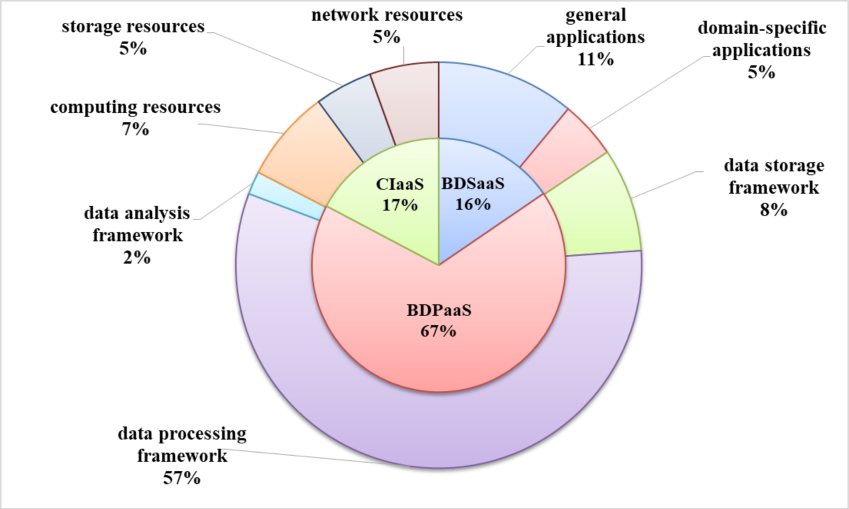

Fig. 3. Statistics of the reviewed papers by layer and their breakdown

4.2.1 BDSaaS layer **General applications**: The authors [74] addressed the challenge of situations where numerous job instances in BDAAs should be concurrently deployed at runtime. They introduced DepWare that is a specialized middleware capable of offering an autonomic deployment decision making. They then designed DepPolicy that is a novel language to specify fundamental deployment information. Moreover, an algorithm of deployment decision making is proposed to achieve the optimum deployment for each job instance. Experiments shows that their algorithm of deployment decision making can simultaneously make diverse decisions of deployment at runtime for different job instances. Meanwhile, optimal overall utility is achieved, all given constraints (e.g., cost limit) is satisfied and SLAs (e.g., feasibility, functional correctness, performance, and scalability) is guaranteed.

Domain-specific applications: The authors in [100] focus their work on healthcare BDAA where missing any SLA can generate significant influence on the data analysis of emergency patient thanks to the disease severity. They proposed a computing model for SLA-based healthcare BDAA and implemented a single API to manipulate and analyze both stream-based and batch-based data over Spark platform. They then presented a probabilistic method based on parallel semi-Naive Bayes (PSNB) and designed a modified conjunctive attribute algorithm for dimensionality reduction to improve the accuracy. For those jobs with high priority, they proposed an adaptive job scheduling algorithm to optimize their execution time that satisfies SLA. Experiments results show that their proposed model for SLA-based healthcare BDAA outperforms extant parallel processing models. Also, their proposed PSNB-based approach enhances accuracy compared to the original Naive Bayes algorithm. Differently, the authors [97] focus on how to schedule BDAA workflows in both single Cloud and federated inter-Cloud environments. A workflow consists multiple tasks that need storage, computing and bandwidth resources to transmit and process data. The resources in the workflow need satisfying SLA requirements (e.g., optimizing time to meet deadlines, optimizing cost and managing budgets). A Cloud or inter-Cloud provider could provide resources for executing tasks in the workflow according to specified SLAs criteria. A case study of banking application demonstrates that single or federated Cloud resources are very necessary in terms of executing BDAA workflows.

4.2.2 *BDPaaS layer* **Data Storage Framework:**

Sakr et. al. [101] considers cost management and SLA-based provisioning for Cloud-hosted NoSQL databases. In their paper, they proposed an end-to-end framework that is represented as middleware residing between the Cloud-hosted databases and consumer applications. The proposed framework aims to flexibly and dynamically provisioning one database function in BDAAs while satisfying their SLA performance requirements (e.g., variability, scalability, elasticity, and performance) according to application-defined policies and avoiding the monetary cost when SLA violations happen as well as controlling the expenses when allocating computing resources. In the context of data security in NoSQL database, Crypt-NoSQL [113] is the first prototype that can encrypt data and execute queries on NoSQL databases while providing high performance. The authors in this paper proposed three different types of models for Crypt-NoSQL and evaluated its performance using Yahoo! Cloud Service Benchmark. Their experiments demonstrate that Crypt-NoSQL is able to efficiently execute queries while guaranteeing SLA requirements (e.g., scalability, high performance). Moreover, they proposed guidance for providers to establish Crypt-NoSQL in the form of a Cloud service and set up pertinent SLA conventions.

Data Processing Framework:

On the one hand, some papers focus on batch-based MapReduce jobs in Cloud. For example, the authors [16] aim at minimizing SLA metrics (e.g., response time) and keeping deadlines set in the pSLA (platform-level SLAs) in this context. First, they developed a so-called grey-box model that can accurately obtain the characteristics of MapReduce behavior. They then proposed a control theory-based framework to satisfy the objectives of SLA. A feed-forward controller was designed and implemented to assure constraints such as service time and improve control response time. The experiments illustrate that the controller is valid in meeting the specified deadlines in the SLAs. Lim et al. [66] propose a novel MapReduce resource manager using constraint programming-based method. In this paper, each MapReduce job is featured by a set of metrics (e.g., the time of earliest start, the time of execution, and deadline) specified in an SLA document. The authors evaluated the performance of their resource manager through an open and discrete event-based simulator where a stream of jobs arrive at intervals. The experiments show that the resource manager can achieve good performance in matchmaking and scheduling MapReduce jobs and give insights into the behavior and performance of system.

On the other hand, some papers focus on stream-based jobs. For example, Rafael et al. [121] take into account simultaneously executing stream workload over shared Cloud infrastructures where each stream is characterized by specific quality of service (QoS) objectives (e.g., throughput, latency) specified in an SLA. They consider classifying customers who submit streams workload into three different classes (Gold/Silver/Bronze/). Each class differentiates by a unique penalty and revenue from providers' side. Their proposed profit model can consider both the cost of provisioning and penalties when the violations of SLA occur. Experiments show that their approach can apply the enforcement of QoS for each application. Paper [46] discusses provisioning resources for stream-based jobs at a granularity of VMs level at runtime. They proposed a novel method to provisioning resources in a cost-efficient way while optimizing the resource usage of VMs (SLA metrics at CIaaS layer).

Moreover, their method is integrated into the Vienna ecosystem at runtime environment for scalable stream processing. The evaluation shows that their method achieves better conformance of SLA by up to 25% and the operation cost reduction up to 36% compared to the extant threshold-based method.

Data Analysis Framework:

The authors [65] developed an extensive model for predictive analysis regarding performance and cost in Cloud. They collected data of resource consumption and placed them in readiness state to enable fast analysis. They stored time series data and various kinds of data regarding performance and events of BDAAs in a layered object store, which can provide the abilities of fast retrieving and pattern analysis. Meanwhile, the authors took into account the data aggregations regarding the interrelation between performance and cost as well as their dynamic tendency over time. Hence, through the application of real-time predictive analysis techniques, the framework achieves an accurate prediction of the current status (i.e., cost and performance) and prospective status. This provides effective support for providers to make decision based on resource configuration regarding the guarantee of SLA requirements.

With regards to analytical capability on prediction accuracy, Lekha et al. [83] focus on developing a real-time stream-based system for the prediction of health status of patients. The system is implemented and deployed on a Cloud-hosted Spark platform which leverages the power of multiple machine learning algorithms. In this scalable system, first, the health information tweeted by users are captured. Then, the proposed system can receive the same health information in real time. Next, the system preprocesses and extracts valid health information from those unstructured stream data and utilizes machine learning algorithms to forecast the health status of users with the purpose of maximizing the accuracy of prediction. By the virtue of the availability of high-quality training datasets of healthcare and the computing power of steam processing of Spark, the process of analyzing huge healthcare samples by applying machine learning techniques becomes significantly more efficient than ever, resulting in enhanced prediction accuracy.

4.2.3 CIaaS layer In terms of Cloud storage, paper [24] addressed various requirements of customers regarding Cloud storage. They first defined a set of realistic and concrete SLA elements. Then, they designed the short-secret-sharing Cloud storage system that applies the defined SLA elements and provides customers with a protected and steady storage service in Cloud. Their proposed system can capture applicable parameters to offer customers with their wanted services while respecting SLAs (e.g., minimal costs). The authors in [136] focus on the scheduling policies of volume request that can shorten the violations of SLA in terms of I/O throughput in Cloud storage systems. To this end, they propose various SLA-driven scheduling policies that consider both I/O throughput and available capacity of backend nodes. The designed scheduling policies can considerably reduce monetary cost of Cloud storage from providers' perspective.

Unlike the above works, Yassine et al. [138] discussed the challenge of transferring multimedia big data over Cloud data centers that are geographically distributed. Since the multimedia volume increases, there is an increasing demand to transfer large datasets across data centers. Therefore, the surplus bandwidth that occurs at different times and for different period in backbone network turns to be inadequate to meet speedily increasing demand for transferring multimedia big data. In this paper, they designed multi-rate Bandwidth-on-Demand (BoD) service to communicate among geo-distributed Cloud datacenters. They also developed a scheduling algorithm that is employed by a BoD broker, which considers transferring multimedia big data requests that are featured by various deadlines.

4.3 Cloud Deployment Models (In response to *RQ2*)

In the reviewed papers, researchers select a particular type of Cloud deployment model (private / public / community / hybrid) to carry out various activities such as implementing the proposed framework, deploying prototype or tools, evaluate approaches or simulate testing environments. Figure 4 shows the distribution of Cloud deployment models in the reviewed papers.

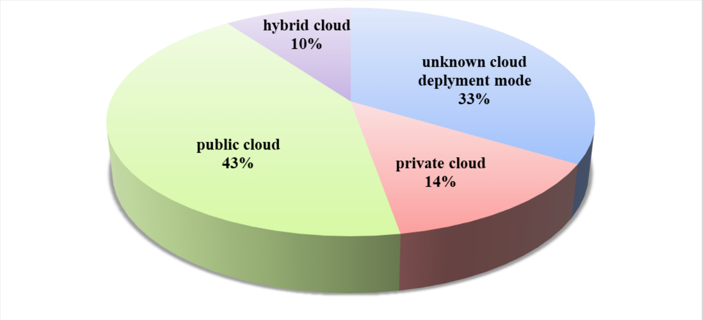

Fig. 4. Statistics of the reviewed papers by Cloud deployment models

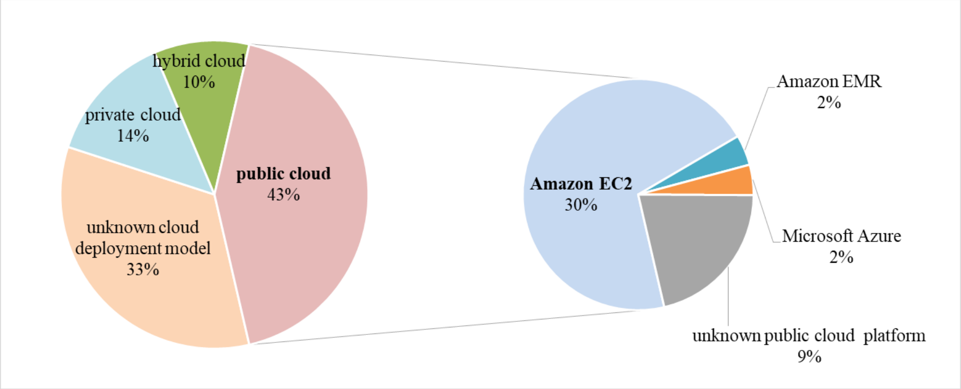

Fig. 5. Breakdown of public Cloud deployment model

It is worth noting that a significant fraction of the reviewed papers only mention Cloud service in general without explicitly telling the Cloud deployment model they used. In this case, we label it as "unknown Cloud deployment model". Apart from the section of "unknown Cloud deployment model", it is observed from Figure 4 that public Cloud is accredited as the principal Cloud deployment model. This finding is in line with our understanding that the public Cloud is the most common and well-known deployment model in comparison with the others. Accordingly, researchers are prone to choose public Cloud to deploy their proposed prototype, applications or tools and evaluate their proposed techniques. The second and third-ranked section is private Cloud and hybrid Cloud, occupying 14% and 10% respectively. Interestingly, community Cloud is not used in the reviewed papers.

Regarding public Cloud, we further investigate what specific public Cloud platforms were selected. Figure 5 gives an apparent breakdown of the public Cloud. It is found that some papers fail to state what specific public Cloud platform was used. Hence we mark them as "unknown public Cloud platform" in Figure 5. It is seen that Amazon EC2 is the preferable public Cloud platform, occupying 29%. Comparatively, only 2% of the reviewed papers use Microsoft Azure as their deployment platform.

Next, Figure 6 presents the breakdown of private Cloud. Similarly, some papers fail to state what specific private Cloud platform was used. Hence we label them as "unknown private Cloud platform" in the figure. It is observed that OpenStack is the most favorable private Cloud platform, occupying 8%, while its competitor OpenNebula and VMware vSphere equally share 1%.

Moreover, the breakdown of the hybrid Cloud is shown in Figure 7. The section of "unknown private and public Cloud platform combination" denotes that it is not deducible what specific mixture of public and private Cloud platform used according to the reviewed papers. It is interesting to find that OpenStack and Amazon

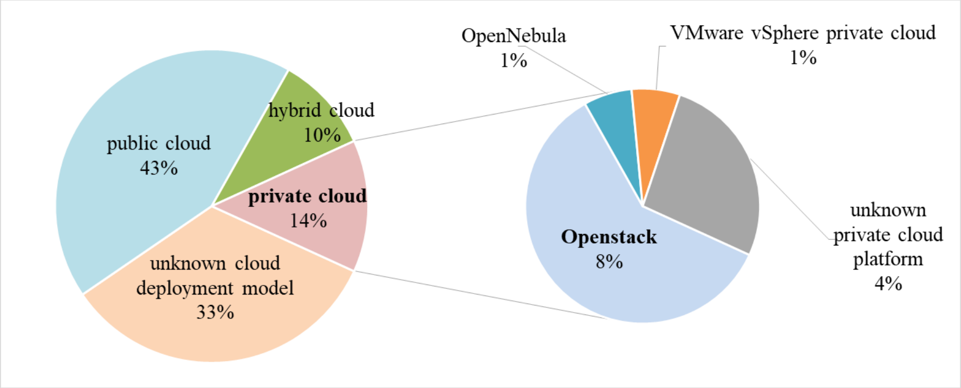

Fig. 6. Breakdown of private Cloud deployment model

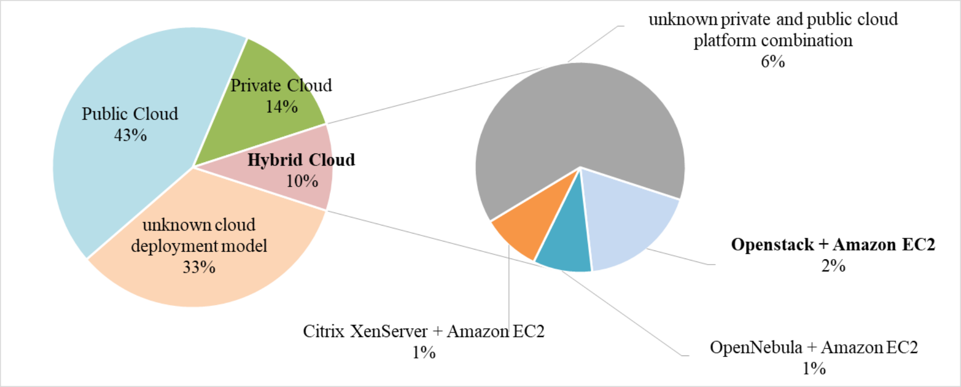

Fig. 7. Breakdown of hybrid Cloud deployment model

EC2 is the more popular combination despite having a small percentage (2%) than other combination such as OpenNebula plus Amazon EC2, Citrix XenServer plus Amazon EC2 occupying 1% respectively.

4.4 Techniques (In response to *RQ3*)

Figure 8 shows the statistics of the reviewed papers by different SLA management techniques. Based on this figure, it can be deduced that the dominant techniques are optimization, scheduling, simulation, monitoring, machine learning, constraint programming, and scaling. Further, Table 5 shows these techniques used to address SLA management for Cloud-hosted BDAAs and their breakdown by layer. From this table, it is clear that the most common techniques used in SLA management are Optimized-based, Simulation-based, Scheduling-based, Machine learning-based and Monitoring-based techniques. There are few research works that investigated other techniques such as scaling, fuzzy logic and error-handling, showing that these techniques are uncommon in the landscape of SLA management for Cloud-hosted BDAAs.

It is deserving to note that some authors combine more than one technique to address SLA management for Cloud-hosted BDAAs. They might use scheduling and machine learning-based technique, or fuzzy logic and machine learning-based technique, or monitoring and scaling technique. For example, Rajinder et al. [103] propose a overall architecture regarding SLA-aware scheduling of BDAAs across geo-distributed Cloud datacenter. Their proposed scheduling algorithms have two levels including coarse-grained and fine-grained. In this paper, firstly, they employ Naive Bayes algorithm to predict which category a user's BDAA belongs to. They then apply Adaptive K-nearest neighboring-based scheduling algorithm to discover which regional datacenter is appropriate

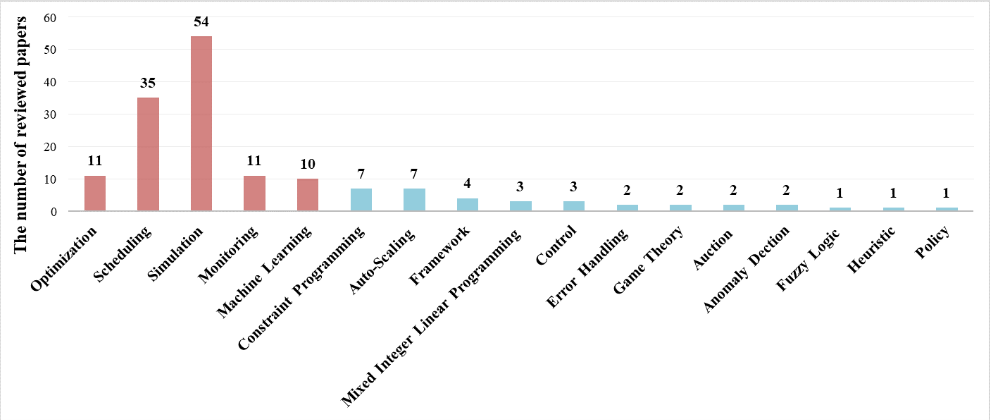

Fig. 8. Statistics of the reviewed papers by techniques

Table 5. Techniques used for SLA management for BDAAs in Clouds

Technique (papers quantity)	**Service Layer (papers quantity)**	**References**
Optimization-based (11)	BDPaaS (7) CIaaS (4)	[12, 29, 40, 41, 46, 63, 134] [48, 76, 94, 138]
Scheduling-based (35)	BDSaaS (8) BDPaaS (24) CIaaS (3)	[56, 73, 78, 85, 97, 103, 110, 111] [10, 15, 18, 31, 39, 43, 44, 50, 57, 60, 68, 77, 87, 90, 93, 95, 96, 109, 124, 128, 129, 133, 145, 146] [86, 119, 136]
Simulation-based (54)	BDSaaS (3) BDPaaS (36) CIaaS (15)	[25, 100, 110] [10, 12, 14, 29, 37–41, 43, 46, 50, 51, 54, 59, 60, 66, 68–70, 78, 87, 90, 93, 94, 109, 117, 121, 124, 129, 133, 134, 141, 142, 145, 146] [22, 23, 30, 48, 55, 72, 76, 86, 94, 104, 119, 136, 138, 139, 148]
Monitoring-based (11)	BDSaaS (5) BDPaaS (4) CIaaS (2)	[8, 11, 107, 108, 112] [59, 84, 122, 123] [102, 104]
Machine learning (10)	BDSaaS (5) BDPaaS (3) CIaaS (2)	[11, 25, 100, 103, 107] [51, 63, 122] [34, 104]
Control-based (3)	BDPaaS (3)	[16, 105, 106]
Constraint Programming (7)	BDSaaS (1) BDPaaS (6)	[74] [37, 38, 66, 67, 69, 137]
Scaling (7)	BDPaaS (5) CIaaS (2)	[35, 47, 75, 88, 117] [55, 102]
Mixed Integer Linear Programming (3)	BDPaaS (2) CIaaS (1)	[41, 67] [138]
Fuzzy Logic (1)	BDSaaS (1)	[25]
Framework-based (4)	BDSaaS (1) BDPaaS (2) CIaaS (1)	[118] [101, 144] [9]
Anomaly Detection (2)	BDPaaS (2)	[53, 104]
Auction (2)	CIaaS (2)	[22, 23]
Model Checking (1)	BDSaaS (1)	[61]
Heuristic-based (1)	BDPaaS (1)	[147]
Policy-based (1)	BDPaaS (1)	[79]
Game Theory (2)	BDPaaS (1) CIaaS (1)	[36] [139]
Error Handling (2)	BDPaaS (2)	[70, 71]

based on locations and requirements of users. Next, they performed the optimal scheduling of big data jobs based on their designed scheduling architecture and typical Amazon scheduling policies in the local server. In this paper, they investigated SLA metrics such as the time of waiting, the utilization of CPU, availability, estimated time to complete and response time. Experiments shows the efficacy of their coarse- and fine-grained scheduling algorithms. In [25], the authors focus on big media healthcare BDAAs in Clouds, which must satisfy SLAs for medical users. In this work, they exploited fuzzy logic to orchestrate a local- and global-based Cloud federation model that optimizes the selection decision making regarding target Cloud data centers. The model considers the trades off between the users' application service quality and providers' profit when choosing federated data centers. Also, the model acknowledges the dynamic behavior that user requests posses and system environments. Through the precise estimation of resource requirements for processing big data jobs using multiple linear regression algorithms, the accuracy of selection decision is significantly enhanced. In the subsequent subsections, we will give the details of the fundamental properties of some representative techniques and their application in some of the reviewed papers.

4.4.1 Optimization-based Generally speaking, the appropriate utilization of resources makes the tasks of SLA management more favorable for providers. As a result, providers continuously demands optimization-based algorithms that can optimally allocate/reallocate resource to maximize resources utilization and providers' profit. It makes sense that an optimization-based technique is essential in this context.

For Cloud-hosted BDAAs, the vast configuration diversity and dependency across different layers makes it difficult for customers to choose appropriate configurations or even decide an applicable background regarding their decisions. Moreover, the extant simple optimization algorithms fail to meet the requirements of most BDAAs that are often featured by different objectives, either because one of the objectives is unsatisfied, or the results appear far from the optimum [28]. Consequently, allocating Cloud resources (at CIaaS level) to big data platforms (BDPaaS level) is not any more a conventional single objective problem (e.g., minimizing time, maximizing resource) but involves multiple contradictory objective functions expressed by SLA metrics such as the maximization of classification accuracy using Apache Spark MLlib, the minimization of response time of MapReduce tasks using Apache Hadoop, the minimization of stream processing latency using Apache Storm and the maximization of CPU utilization and so on. Further, the formulated multi-objective optimization problem demands a considerable amount of computation that is increasing exponentially with the problem size in order to find optimum solutions.

Take the energy consumption optimization for BDAAs as an example, the authors [76] propose a multi-objective optimization-based technique that is aware of both energy and SLA requirements when placing and consolidating VMs. Their proposed technique considers to balance the performance and energy utilization of such system as well as SLA-compliance (e.g., availability and reliability). The results demonstrate that their technique achieves better performance on saving energy, reducing resource consumption and communication cost, minimizing the quantities of VM movements and SLA violations in comparison with the other extant tested approaches.

In terms of optimizing PaaS providers' profit, Dib et al. [29] propose a decentralized optimization-based policy and consider to pay the penalties when SLA violations occur. Their proposed policy achieves optimally exploiting private resources, especially at peak time, before leasing any public Cloud resources. The paper applies their proposed optimization-based policy into a realistic batch-based BDAA. Similarly, the authors [23] addressed the challenges of allocating resource while guaranteeing SLAs and maximizing providers' profit. In this paper, the penalty cost incurred by SLA violations is considered in order to increase providers' profit. They take into account SLA metrics such as execution time and deadline of jobs (i.e., urgency) in a combinatorial auction system and propose a new winner determined algorithm (an optimization-based technique). Experiments proves the efficacy of their approach on the reduction of the penalty payment incurred by SLA violation and maximization of providers' profit.

Unlike the above works, paper [48] addresses how to optimize the distribution of big data and allocation of computing resources on mobile Cloud platforms. As such, the authors propose a new network architecture and algorithms. They discuss an end-to-end thin-thick client collaboration to efficiently distributing data by transferring large dataset into splits depending on the bandwidth of Internet connection. Also, this paper details the procedure of selecting suitable algorithms that can efficiently enhance the utilization and allocation of resources and improve users experience by meeting expected SLA requirements (e.g., minimized VMs quantity, shortened execution time and budget).

4.4.2 Scheduling-based Scheduling is one of the fundamental techniques in addressing SLA management for Cloud-hosted BDAAs. Primarily, this technique is based on the above optimization technique where an Non-determin istic Polynomial-time Hardness (NP-hard) optimization problem has been formulated. Unlike the optimization-based technique, scheduling takes a further step of allocation works based on optimal solutions. Depending on different purposes, such allocation works include assigning physical resources (e.g., machines) to virtualized resources (e.g., VMs), or allocating VMs resources to particular batch or stream processing tasks, or designating platform resources to various BDAAs in a smart way while respecting SLA requirements.

Scheduling has been widely used for traditional applications or workflows in Cloud computing (CC) environment. However, unlike them, distributed data processing technologies such as MapReduce paradigm are now often utilized by many organizations to deploy their big data analytical applications (BDAAs). Therefore, scheduling algorithms used for traditional applications or workflows in CC environment cannot be applied directly to BDAAs in Clouds due to the complexities that data processing frameworks incur and the difference of resource allocation mechanisms that big data brings. As a result, various SLA-based scheduling mechanism and algorithms have been proposed, which primarily aims to optimize resource utilization and provides optimal resource allocation/reallocation solutions for Cloud-hosted BDAAs while meeting multiple SLA requirements.

When addressing SLA management for BDAAs in Clouds, scheduling technique can be applied at different layers. Hence, it can be classified into three classes, i.e., “scheduling at the BDSaaS layer”, “scheduling at the BDPaaS layer” and “scheduling at CIaaS layer”.

Scheduling at the BDSaaS layer

Optimally and strategically providing low-level resources to support BDAAs, jobs or workflows while guaranteeing agreed SLAs between providers and customers is the fundamental objective for the tasks of scheduling at BDSaaS layer. Scheduling at this layer has twofold consideration. On the one hand, it should satisfy users’ SLA requirements and optimize objectives such as complete time, makespan, user capital expenditure, and application performance from the customers’ perspective. On the other hand, it should efficiently schedule big data platform resources to the application layer to maximize profit or reduce the carbon cost or energy consumption by Cloud centers from the providers’ perspective [17].

Verma et al., [124] consider SLA violation concerning the performance of MapReduce-based BDAAs and propose an automatic framework of resource inference and allocation. According to their proposed framework, they firstly profiled some common performance features such as soft deadline and then estimate the number of resources required for completing jobs to meet the deadline. Their proposed algorithm can efficiently schedule the execution sequence of jobs and determine the resources quantities allocated to these jobs while meeting job deadlines.

The authors [145] addressed the challenge of resource scheduling to optimize profit of providers. To this end, they first proposed a scalable and adaptive policy of admission control. Then, they developed a novel algorithm that can optimally schedule resources according to users’ query requests while guaranteeing SLAs on deadlines and budgets, and prompt responses with manageable monetary expense.

Scheduling at the BDPaaS layer

Scheduling at the BDPaaS layer aims at allocating some dependent and independent tasks to VMs in Hadoop clusters. A valid scheduling algorithm can provide the optimal solution of task distribution over various VMs in a cluster depending on the requirements of execution time and availability of resources. An optimal distribution of tasks can minimize the scheduled tasks' average execution time and maximize the utilization of allocated resources. As such, the response time of tasks that are to be processed is minimized and resources consumption is reduced [116].

Wang et al. [128] developed a scheduling algorithm at platform-level for MapReduce-based BDAAs that have two practical SLA constraints (i.e., budget and deadline) over the heterogeneous Cloud datacenters. They designed a greedy-based optimization algorithm that can find appropriate VMs from an established pool of different VMs to minimize the job completion time and monetary cost of executing jobs.

Tian and Chen in [120] took into account the entire processing phrases for MapReduce jobs. In this paper, they designed a cost model that formulates the correlation between the input data volume, the MapReduce resources availability, and the Reduce tasks complexity. They performed testing over a limited number of machines to learn model parameters. The proposed cost model can facilitate to make decisions in terms of the optimal amount of resources, the minimization of time under particular financial budget and the minimization of monetary cost under a specific time deadline. Experiments demonstrates that this cost model achieves decent performance and satisfies SLAs for MapReduce-based BDAAs.

Scheduling at the CIaaS layer

Scheduling at this layer is more relevant with the optimal mapping virtualized resources onto physical resources in a homogeneous or heterogeneous environment and with the optimal use of the underlying Cloud resources.

Nita et al. [86] discuss the challenge of transferring big data across various Cloud datacenters where the performance of VMs migration and data transfers are affected. They describe an optimized method that can transfer large dataset according to the characteristics of network and take into account the SLA constraints such as minimizing the duration of individual VM migration. They proposed a scheduling policy that consists of two greedy-based algorithms to transfer large dataset. It manages and maintains an SLA-aware network that impacts the performance of Cloud. They evaluated the proposed scheduling policy by means of the simulation of SLA constraints at CIaaS layer.

In order to address the issue of file requests' tail latency, the authors [119] employed information flow queue theory to provide an optimal scheduling algorithm for erasure codes-based Cloud storage systems (at CIaaS layer). They first designed a model based on k-marriage flow queue. They then built a multi-objective based scheduling strategy to find the optimum depending on SLAs preferences of users. Their solution is featured by the decentralization of the queue form that outperforms the centralization of queue format in terms of the elimination of the overhead of block. Their simulated results showed their approach decently improves tail latency in comparison with the extant approaches of data displacement .

4.4.3 Simulation-based Simulation is also a popular technique used to address SLA management for Cloud-hosted BDAAs, which allows providers to evaluate a broad spectrum of components such as workload, processing elements (e.g., MapReduce, storm), data centers, storage, networking, and SLA constraints.

When providers offer their big data analytics service to customers with awareness of relevant SLAs, they can identify potential issues before introducing them into the operations and focus on service meeting the agreed SLAs. As they define this service, coarse SLAs can be identified and decomposed to identify more targeted SLAs that in turn drive qualification of the feasibility of proposed solutions to meet SLA commitments. This can then be verified through simulation to identify further how other resources are impacted by any shortfall to inform prioritization in addressing any gaps to guarantee SLAs such as maximizing overall resource utilization or reducing idle time. Moreover, the simulation technique is used to predict system performance and further to study SLA impacts in a production environment.

The simulation technique abstracts, models and emulates BDAAs with a wide of components such as workload, or SLA metrics. In cases where the performance of application does not satisfy pre-specified SLAs, algorithms, scheduling, or monitoring are adjusted and further optimized, and corrective and proactive measures are adopted before an issue occurs. Hence, simulation is an essential technique to facilitate SLA management for Cloud-hosted BDAAs. Also, the real-world Cloud-hosted BDAAs covers a wide array of application domains including healthcare, social media, energy and so on. Each type of these application is characterized by diverse architecture, configuration, implementation and deployment requirements. Experimentation in a real environment such as Amazon EC2 or Microsoft Azure for different BDAAs can be challenging for manifold reasons:

- It is not economical to purchase or lease large-scale datacenter infrastructure that will precisely indicate realistic deployment of BDAA and allow researchers conducting experiments with changing hardware resource and dynamic framework configurations, as well as big data diversities in terms of volume, variety and velocity.
- The experiments are not repeatable, because some variables that are not under the control of the tester may affect experimental results.
- Much manual configuration effort involved especially in a real large testbed experiment environment that needs dynamic configurations significantly slows down the performance analysis and makes it almost impractical. As a consequence, it is remarkably challenging to reproduce the experiments outcomes.
- The experiments on a real large distributed platform are unrealistic to some degree due to a huge cluster where a considerable of nodes run in different conditions.

In this case, the simulation technique offers significant advantages to SLA management for Cloud-hosted BDAAs. For example, researchers can conduct controllable and repeatable experiments by means of simulation technique. Also, it becomes easier to study if SLAs met or breached, and investigate how SLAs is impacted by various resources configuration from different layers in a simulated testbed as compared to a real experiment. Simulation technique makes experiments under various configurations of hardware resources easier and provides insights for practitioners to understand the impact that each design choice is upon to SLA guarantees. They also improve the possibility that researchers can share their simulation environment, which contributes to better hypothesis evaluation and results reproducibility. Finally, researchers can instantiate various processing frameworks of BDAAs and multiple workload scenarios as needed by the virtue of simulation-based technique.

We find that 53 papers among the 109 reviewed papers have applied simulation technique, occupying 49%. In order to investigate what particular simulation tools used, we further examine these 53 papers and find interesting results that (i) some papers generally mention that a simulation-based experiment has been conducted without explicitly stating what particular simulation tool used [12, 37, 38, 51, 53, 70]; (ii) some papers simply state that they developed their simulation tools by Java programming and keep them as proprietary code without giving details [90, 119, 136]; (iii) Other papers give specific description regarding simulation tools used [10, 30, 66, 87, 110, 124, 145]. For the first two cases, we label them as "unknown simulation tools". For the third case, we further find that three types of simulation tools are often used in the reviewed papers. They are discrete event simulator (DES) [45, 140], MRPerf [126] and Yarn Scheduler Load Simulator (SLS) [4]. Figure 9 shows the distribution of these three types of simulation tools.

It is observed that DES is the preferable simulation tool, with a percentage of 30%. Among DES, Cloudsim is the dominant one (26%) and widely used by researchers in their papers. Lots of authors use Cloudsim to implement their algorithms to emulate the CC environment or further implement additional logic to mimic the behavior of the MapReduce model. The second-ranked simulation tool is SLS [43, 87, 109, 110], which can support the simulation of large Yarn clusters and application loads in an individual machine. It is interesting to find that two papers use MRPerf [54, 124], a simulator dedicatedly designed for MapReduce jobs to understand how they perform and study the impact of SLA on various Hadoop configuration settings.

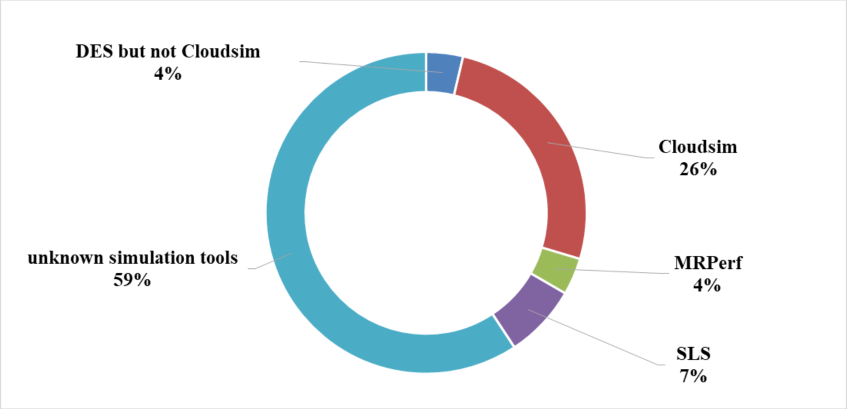

Fig. 9. Statistics of simulation technique used in the reviewed papers

4.4.4 Monitoring-based Monitoring is also an important technique used to manage SLAs for BDAAs. Generally, customers rely on SLAs to deliver the promised quality and level of service. Al through it is readily to notice non-availability or downtime, other types of SLA violation such as performance degradation of VMs and high error rates of APIs are not always easily detected, which can considerably impact the experience of end users. Therefore, monitoring is critical to assure the conformance of SLA and produce fundamental audit trail when SLA violation happens. Moreover, monitoring is essential for providers to guarantee SLA and offer the satisfactory experience to customers [130].

Monitoring of Cloud-hosted BDAAs involves dynamically tracking SLA metrics related to physical resources they share, virtualized resources at CIaaS level (i..e, VM, network and storage), big data processing framework such as Hadoop cluster at BDPaaS level as well as various applications (e.g., smart health, stock recommendation system) running at BDSaaS level. Monitoring is an essential technique to manage SLAs, assisting providers in (i) optimizing the operation of their applications and resources; (ii) capturing performance deviation of application and resources consumed; (iii) monitoring key performance indicators of the applications; (iv) accounting SLA violations regarding specific SLA metrics.

Andreolini et al. [11] take into account the cost minimization regarding computation and communication while assuring peak accuracy in detecting pertinent variations of system behavior guaranteed. They developed an algorithm that can elastically and reliably monitoring big data, which can adapt to update frequencies and sampling intervals. They used real-time series to perform experiments, which shows that their proposed algorithm outperforms extant algorithms in terms of reducing monitoring overheads and maintaining data quality.

The authors [108] develop a method that exploits runtime monitoring to guarantee the applications' performance. They implement a monitoring framework, which collects monitoring data at runtime in a realistic Cloud environment. They then design a performance model that uses data mining techniques to extract from authentic monitoring data at runtime. This model sheds lights on how to adjust the strategy of provisioning resources under specified performance-based SLA requirements.

In the context of monitoring stream-based events in a complicated and time-constrained system, the authors [59] design a framework that can real-time monitor and process large-scale log file streams from various sources. They applied the central limit theory to verify soft deadlines in a real-time system and used the probabilistic deadline to ensure SLA satisfied regarding deadline. Flume is used to collect, aggregate, and transfer voluminous stream-based data from multiple sources to a centralized place where Hadoop HDFS is operated. They extended a generic monitoring architecture and illustrated how to calculate the likelihood of SLA violation. This solution is beneficial for a system of real-time monitoring to determine the deadline for SLAs compliance.

4.4.5 Machine Learning-based Some of reviewed papers use machine learning techniques to study SLA management for Cloud-hosted BDAAs from different aspects. Not only is machine learning used to predict the prospective behavior of resources, but also to detect SLA violations. Machine learning-based technique provides machine derived intelligence to the task of SLA-driven optimization and configuration dependencies across multiple layers. It allows continuously learning many complex behaviors and interactions among interrelated objects/entities in BDAAs scenario and taking the guesswork out of many aspects involved in meeting SLAs more efficiently and cost-effectively. For example, collecting large data regarding VMs, storage, and network at CIaaS layer, Hadoop cluster at BDPaaS layer, and applications at BDSaaS layer, then feeding these data to machine learning-based system, finding the hidden patterns and fixing issues before they might violate SLA guarantee. As long as this wealth of data is gathered, processed and analyzed, machine learning-based technique can learn what constitutes normal behaviors, and it is this baseline that gives the system the ability to detect anomalies and find causes automatically. Thus SLA violation can be avoided. Also, it can simulate and predict the impact of making certain changes in resources and their allocations, which can be particularly useful for meeting SLA objectives such as maximizing resource utilization.

Lama et al. [63] developed a Hadoop-based system that can allocate diverse Cloud resources and automate multiple Hadoop parameters configuration while guaranteeing SLAs requirements such as minimal monetary cost incurred. It addressed the major issue of providing MapReduce-based BDAAs under different performance deadlines. Their approach consists of a novel framework including two phrases (machine learning-based offline phrase and optimization-based online phrase). The offline phrase clusters various Hadoop jobs by using Support Vector Machine algorithm. The clustering result is then regarded as an input and fed into the subsequent online phrase. The online phrase exploits optimization-based techniques to assign Cloud resources and automate the configuration of Hadoop parameters.

The authors [122] address the challenge of optimal resource provisioning for scalable BDAAs. Firstly, they identified that most of the applications running inside JVMs such as Spark highly demand effective memory resources. Then, they consider applications that is featured by their SLAs (i.e., relative delay) and apply Random Forest algorithm to predict their valid memory requirements. The prediction approach can uncover the hidden behavior of BDAAs' memory consumption and forecast dynamic prospective memory utilization in distributed Cloud environments.

4.5 SLA Metrics (In response to *RQ4*)

In this element, we examine SLA metrics accounted for in the reviewed papers to find out what SLA metrics have been discussed and how often they are discussed. Table 6 summarizes SLA metrics and describes their measurements.

Further, Figure 10 present the pictorial representation of the frequency of the above SLA metrics that have been discussed in the reviewed papers. It is observed that the most studied SLA metrics are performance, deadline, resource utilization, and cost. This is consistency with our understanding that actors (i.e., Providers, Customers, and End Users) care more about SLA metrics regarding deadline, cost, performance and resource utilization in Cloud-hosted BDAAs. The least studied SLA metrics are serviceability, consistency, elasticity, security, capacity, reliability, and scalability. This is because these SLA metrics have limitations regarding measurability [92]. The medium level discussed SLA metrics include profit, budget, energy, availability, fault tolerance, and accuracy. These category of SLA metrics are attracting increasing interest from researchers.

It is also found that the above SLA metrics scatter in the reviewed papers, without an organized and clear categorization. Therefore, it is necessary to examine SLA metrics for Cloud-hosted BDAAs through the consideration of building a clear categorization scheme while respecting BDAA characteristics, such that providers and customers will benefit from this categorization scheme when making conventions and engineering SLAs between them.

Table 6. SLA metrics and their measurement for BDAAs in Clouds

SLA Metrics	Unit of Measure
reliability	the number of concurrent failures that are tolerable, MTTF and MTTR and produces the storage system MTTF, number of successful responses in percentage
energy	cost per kWh, power (watts)
capacity	the capacity that the Cloud storage system can store. The options range from kilobytes to zettabytes
availability	percentage of service uptime or downtime
serviceability	period of an outage, duration between consecutive service failures, time to switch over from a failure, time to completely recover from a service failure
security	the ability to detect or tolerate malicious attack
consistency	the degree of equality between responses to queries issued by BDAAs
scalability	the ability to horizontally increase the storage or processing capacity or throughput, and the ability to add more resources (e.g., more processors, memory, bandwidth) to each node to increase capacity or throughput vertically
elasticity	the ability to dynamically and rapidly adjust resources to absorb the demand
fault tolerance	the percentage of continuing operating properly when failures (e.g., data node of Hadoop is down, Map or Reduce task fails) occur
accuracy	percentage of accurate prediction or analysis
cost	monetary cost in terms of VM computing per time unit, electricity prices
profit	revenue made per request
budget	upper bound on monetary cost (dollars) to complete data processing tasks
deadline	upper bound on time (hour) to complete data processing tasks
performance	• throughput: MB/sec • throughput: MB/sec • data freshness • time pertinent: waiting time/ response time/execution time /job processing time/job completion time
resource utilization	• CPU pertinent: MIPS, number of cores regarding CPU or vCPU, CPU utilization etc,. • memory pertinent: MB/GB, memory utilization etc,. • storage pertinent: storage size, I/O throughput etc,. • network pertinent: bandwidth, data transfer time etc,.
request arrival rate	request per second, arrival rate factor (user side) etc.,
quantity pertinent	• the quantity of working node allocated for batch-based or stream-based processing • the quantity of replicas • the quantity of required parallel threads • the quantity of tasks (i.e., Map or Reduce) • the quantity of jobs • the quantity of disks • input data size • the quantity of data blocks • the quantity of VMs

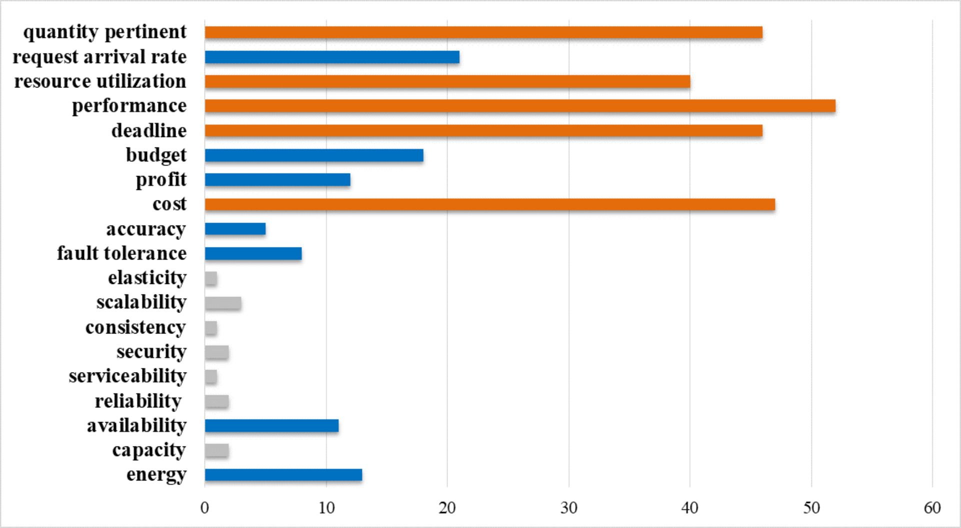

Fig. 10. Frequency of SLA metrics in the reviewed paper

4.6 Conceptualization (In response to *RQ5*)

4.6.1 Conceptual SLA Model In Cloud computing (CC) environment, designing conceptual SLA models or frameworks are often discussed. Alhamad et al. [7] proposed a conceptual SLA framework for CC environment. They consider four types of Cloud service (i.e., IaaS, PaaS, SaaS, storage as a service). For each different SLA, they present the fundamental parameters that are needed to establish a steady form of negotiation and conversation between customers and providers. Based on the above work, the authors [106] developed a new conceptual SLA model in CC environment called SLA as a Service (SLAaaS). SLAaaS can systematically and transparently integrate service levels and SLAs into Cloud. It considers the quality of service levels and SLA as the most superior elements in Cloud services.

Also, Labidi et al. [62] proposed a generic and semantic-rich model that is based on ontology theory in their paper. They developed a prototype to validate their proposed model. Through this prototype, the evaluation and triggered guarantee actions of SLAs can be automatically achieved during their monitoring process.

Moreover, in order to seek an optimal trade-off between revenues and costs while meeting SLA constraints, the authors [64] designed a service-based model that consolidates the major characteristics and SLA objectives of Cloud services. Although this model is generic and abstract, it is beneficial to derive a universal and automatic manager with the capability of managing any Cloud service, no matter what the layer.

In addition, the authors [26] proposed a formal model to describe SLA contents in CC environment and design autonomic mechanism of predicting SLA violation. Their proposed SLA model is devoted to formalizing a capability to manage SLAs violation detections for Cloud services. The proposed approach concerns the representation of information from both the SLAs and Cloud logs in a specific format.

All the above-mentioned works are confined to common Cloud service without specific consideration of BDAAs. To the best of our understanding, the conceptualization of SLA model dedicated to Cloud-hosted BDAAs is rare. The next section will proposes a new conceptual SLA model.

5 A New Model and Categorization Scheme of SLA Metrics

In this section, we design a conceptual SLA model dedicated for Cloud-hosted BDAAs. We further elaborate on this model to propose a multi-dimensional categorization scheme of SLA metrics for BDAAs in Clouds.

5.1 Cross-layer SLA Model for Cloud-hosted BDAAs

5.1.1 Design Principle and Requirements According to the layered architecture of Cloud-hosted BDAAs in Figure 1 of Appendix A, it is easy to figure out that each layer has two kinds of requirements that are crucial to service composition, which are functional and non functional requirements (FRs and NFRs). These requirements clearly define what the service provider should meet and provide to customers. The categorization of requirements for layer-based BDAAs in Clouds is shown in Figure 11.

On the one hand, the focus of FRs is on the functionality of the composed service. For instance, a sentiment analysis service from customer reviews using Amazon Comprehend detects sentiments in the text and extracts information about users' sentiment polarity (Positive, Negative, Neutral or Mixed) [32]. One of FRs in this case is that a sentiment analysis accuracy lower than 80% will never be purchased. A user requests FRs at the top layer (i.e., BDSaaS) and these requirements will be drilled down to the bottom layer (i.e., CIaaS), which provides concrete, scalable and on-demand Cloud resources. In other words, upper layer demands resources from a lower layer while a lower layer supplies resources to an upper layer. Thereby, each service in a layer is featured by unified interfaces by which Cloud-hosted BDAAs invoke possible functions. For example, Amazon EC2 acts as basic Cloud infrastructure and provides a functional interface that supplies its client (i.e., BDPaaS) with computing instances, to install and run software on these instances. By demanding scalable Cloud resources from Amazon EC2, Amazon EMR located at BDPaaS layer can supply its client (i.e., BDSaaS) a fully managed Hadoop cluster

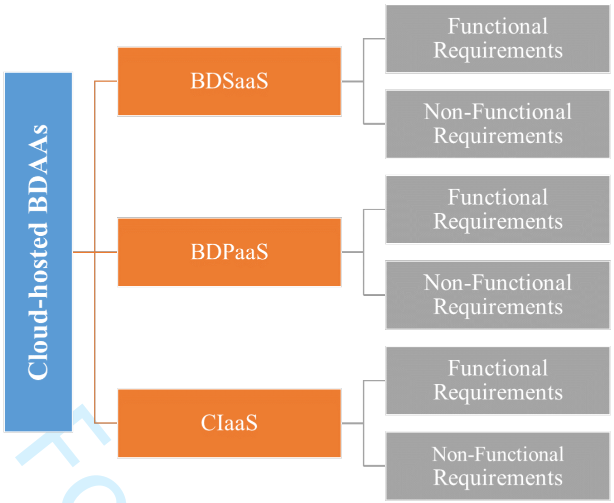

Fig. 11. Categorization of requirements for layer-based BDAAs in Clouds

in minutes, and then Amazon Comprehend uses advanced techniques such as machine learning and natural language processing to predict sentiments as much accurate as it can. As a result, FRs specified by users could be met.

On the other hand, NFRs are concerned with SLA metrics, An instance of a NFR for the above Amazon Comprehend sentiment analysis service is that the service response time to a user should be no more than 5 seconds. NFRs are encapsulated and incorporated into SLAs, where multiple metrics are considered such as maximum data transfer ratio, maximum availability, and minimum network latency. NFRs are important to big data analytics service composition and are often formally expressed in SLAs as part of contracts agreed between providers and customers. Even though FRs are met, unsatisfied NFRs such as slow or unreliable service may still not be adopted for BDAaaS.

In addition to functional and non-functional requirements, the dependency relationships between SLAs across different layers of BDAAs is another critical aspect in designing SLA model for the applications. A sole layer is impossible or struggles to provide either FRs or NFRs, thus it is bound to compromise service quality for customers. Having all layers work jointly, the agreed service quality can be guaranteed in the end. Accordingly, we need a novel SLA model for Cloud-hosted BDAAs that should meet the following essential design principles:

- Allowing the definition of both functional and non-functional interfaces that expose SLAs by layer for big data analytics service.
- Representing a seamless integration between SLAs and BDAaaS across layers.
- Considering SLAs for Cloud-hosted BDAAs in a unified and structured way.
- Reflecting strong dependency relationships between those SLAs.
- Possessing an universal applicability regardless of BDAAs.

5.1.2 Proposed Cross-layer SLA Model for Cloud-hosted BDAAs Keeping the aforementioned design principles in mind, we propose a novel cross-layer SLA model for Cloud-hosted BDAAs (named CL-SLAMfBDAAs) shown in Figure 12. As shown in this figure, there are four interacting actors located at different layers. They are end users (e.g., Business Users, Subject Matter Expertise, Data Scientist or Data Analyst), BDSaaS provider (e.g., Salesforce or BrandsEye), BDPaaS provider (e.g., Google Cloud Dataflow, Amazon EMR or Microsoft Azure HDInsight), and CIaaS provider (e.g., Google Compute Engine, Amazon EC2 or HP Cloud). These actors are involved in a set of activities, for instance, negotiating, clarifying and specifying FRs and NFRs, and formulating NFRs into agreed SLAs in each layer. Moreover, it is observed from this figure that SLAs are dedicatedly divided into three

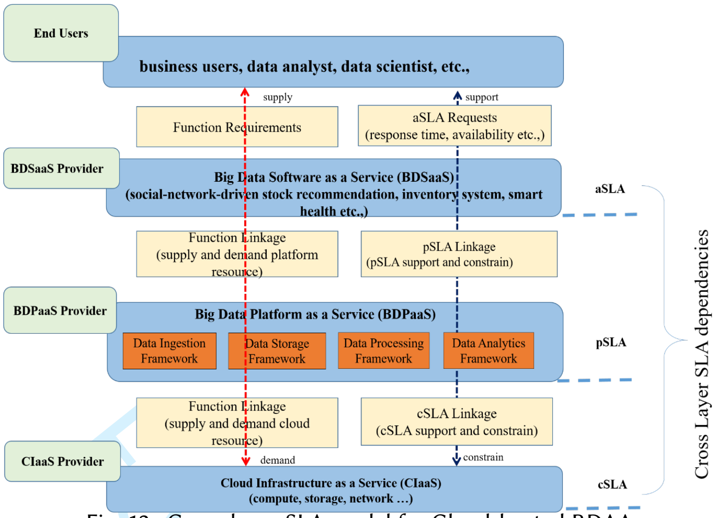
End Users
business users, data analyst, data scientist, etc.,
supply
support
Function Requirements
aSLA Requests
(response time, availability etc.,)
BDSaaS Provider
Big Data Software as a Service (BDSaaS)
(social-network-driven stock recommendation, inventory system, smart health etc.,)
aSLA
Function Linkage
(supply and demand platform resource)
pSLA Linkage
(pSLA support and constrain)
BDPaaS Provider
Big Data Platform as a Service (BDPaaS)
Data Ingestion Framework
Data Storage Framework
Data Processing Framework
Data Analytics Framework
pSLA
Function Linkage
(supply and demand cloud resource)
cSLA Linkage
(cSLA support and constrain)
demand
constrain
CIaaS Provider
Cloud Infrastructure as a Service (CIaaS)
(compute, storage, network …)
cSLA
Cross Layer SLA dependencies

Fig. 12. Cross-layer SLA model for Cloud-hosted BDAAs

categories including application-level SLAs (aSLA), platform-level SLAs (pSLA) and Cloud infrastructure-level SLAs (cSLA).

Furthermore, there are strong bonds between the upper layer and the corresponding lower layer of SLAs. From the perspective of FRs, the actor at each layer has a bidirectional relationship with its neighbor's layers either by demanding or supplying behavior. In other words, the upper-layer actor demands or requests resources from the lower-layer actor while lower-layer actor supplies the resources requested to upper-layer actor. From another perspective, a two-way relationship is existed, where SLAs at each layer either constrain or support SLAs in its adjacent layers. Concretely, an end user requests FRs along with NFRs through the interface with BDSaaS provider. The NFRs will be negotiated and defined into aSLAs between them. Then, an aSLA will be interpreted and formulated into a set of pSLAs. After that, a pSLA will be transformed into a set of cSLAs. From topmost to bottom, the upper-layer SLA sets the constraints into its lower-layer SLA and decides how well the lower-layer SLA must work to meet service-level objectives in the end. In turn, the low-layer SLA works hard to support its upper-layer SLA. For instance, if the availability in aSLA is 99% (a 99% availability at this layer means that users will be able to access the application at least 99% of the time), then, the availability in pSLA must be hover somewhere between 99% and 99.99%, and the availability in cSLA must be higher than the availability in pSLA. In the absence of meeting this, it might fail to guarantee SLA constraints.

In terms of SLA metrics, cSLA metrics such as VMs quantity, CPU and memory resources utilization, or the availability of VMs affects the pSLA metrics such as the quantity of map and reduce nodes in Hadoop platform (Data Processing Framework at BDPaaS layer) or the other pSLA metrics such as the quantity of data nodes, transfer rate, and replication factors of NoSQL database service (Data Storage Framework at BDPaaS layer). Certainly, this at last affects aSLA metrics such as capital cost, availability and reliability of the applications. This exemplifies the strong cross-layer SLA dependency relationship. To guarantee the final SLA to customers, BDSaaS provider should guarantee aSLAs by interweaving pSLAs and cSLAs.

Finally, our proposed CL-SLAMfBDAAs model presents a unified and structured scheme to describe and interpret SLAs for Cloud-hosted BDAAs. In this novel model, SLAs are exposed and linked in a vertical motion, which is orthogonal to the layers and may apply to any of them. Based on this model, users know what different types of SLAs with various attributes exist and how they work collaboratively across layers to ensure the delivery of SLA guarantee. This model meets all the aforementioned design principles and requirements. It is further elaborated to propose a new categorization scheme of SLA metrics for Cloud-hosted BDAAs.

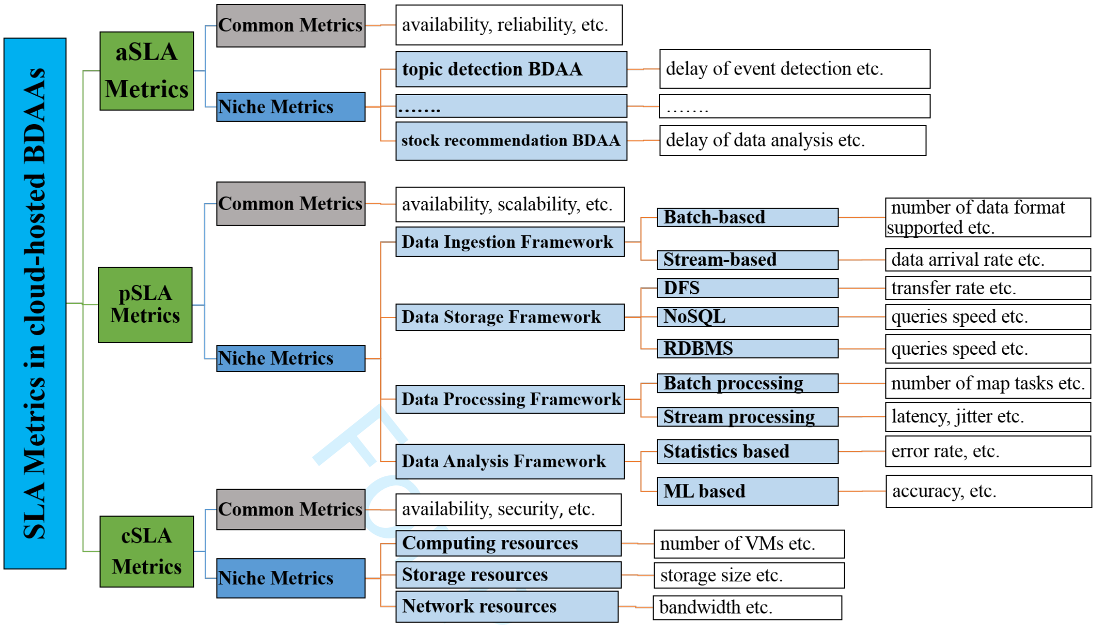

Fig. 13. Categorization scheme of SLA metrics for BDAAs in Clouds

5.2 Multi-dimensional Categorization Scheme of SLA Metrics for Cloud-hosted BDAAs

The typical SLA metrics at CIaaS layer that customers expect are the number of VMs, memory size, CPU usage, hard disk utilization, memory usage, additional network parameters and so on. While at BDPaaS layer, the example of SLA metrics include throughput, response time, and availability. For instance, in the case of a Hadoop cluster (at BDPaaS layer), we have metrics such as execution time, job turnaround and makespan in terms of MapReduce tasks [52]. At BDSaaS level, a particular SLA metric is highly determined by the genre of BDAA. For example, the rate and quality of data transfer are important for any video streaming-oriented BDAA, while latency of processing and network generally interests a batch-based BDAA. It is worth to note that SLAs at each layer might have an endless variety of metrics depending on different components and the nature of applications.

Figure 13 shows our proposed extensible and multi-dimensional categorization scheme of SLA metrics for Cloud-hosted BDAAs. This schema not only defines the most commonly used metrics for each type of SLA (i.e. aSLA, pSLA and cSLA), but also presents niche SLA metrics to be consistent with different components at each layer.

5.2.1 aSLA Metrics There are lots of different types of BDAAs across a wide range of industries. For instance, topic detection and tracking applications, large-scale log analysis applications and business intelligence. Due to their wide variations, listing all SLA metrics at this level is impracticable. Hence, we select some typical Cloud-hosted BDAAs and present common aSLA metrics for them as shown in Table 7. To embody the unique features of these BDAAs, we provide niche aSLA metrics for them in Table 8.

5.2.2 pSLA Metrics There are four main components/frameworks at BDPaaS layer (i.e., data ingestion, storage, processing and analysis). Selection of the specific software or tool as an instantiation of the aforementioned different frameworks is based on many aspects such as flexibility, control and ease of use. Considering the differential nature and role-playing, each framework at BDPaaS layer has different pSLA metrics. Table 9 lists some common pSLA metrics, while some niche pSLA metrics for each framework are given in Table 10.

5.2.3 cSLA Metrics Companies like Microsoft, Google and Amazon offer infrastructure as a service. With this diverse range of Cloud infrastructures, most customers are perplexed to choose which SLA metrics should be defined and specified as the hardware section of cSLAs. To clear away this confusion, we give the most common and niche SLA metrics that interest customers when using Cloud resources in Table 11 and Table 12 respectively.

To better understand SLAs across different layers of BDAA, we present a SLAs template using a real Cloud-hosted BDAA in Appendix B.

6 Open Issues and Future Trends

In this paper, we performed a taxonomic study on SLA-specific management for big data analytical applications (BDAAs) in Clouds. Particularly, we addressed the most pertinent survey questions in this field.

The taxonomy-based study suggested that (i) The BDSaaS and CIaaS layers have received much less interests from researchers in comparison with the BDPaaS layer. Hence, one of future trends could be paying more attention to the former two layers. Taking BDSaaS layer as an example, future researchers could investigate how to manage SLAs by using more domain-specific BDAAs such as healthcare, banking, and smart city; (ii) In terms of BDPaaS layer, researchers put less attention to data storage (e.g., NoSQL) compared to data processing (e.g., Hadoop). Since NoSQL attracted significant interest in recent years and also play an important role in Cloud-hosted BDAAs. Hence, future work could study NoSQL-specific SLA management in Clouds; (iii) Considering data processing technologies (batch or stream), the minority of papers discuss SLA management for stream-based applications compared to batch-based applications. Since stream processing has recently received increasing attention because of technological innovations which have facilitated the creation, maintenance, and processing of massive data with lower latency and better resilience. Therefore, future trend could particularly focus on SLA management for stream-based BDAAs; (iv) Techniques such as constraint programming, auto-scaling, error handling and so on received less attentions by extant researchers compared with optimization, scheduling, simulation, monitoring and machine learning techniques, which points out another possible future work.

As a conclusion, we explored the current research state in the field of SLA management for Cloud-hosted BDAAs and provided some future works. Provisionally, future researchers would take advantage of the ideas from this taxonomy-based survey as an entry point to address some gaps and most importantly enhance the maturity of the research field on SLA management for Cloud-hosted BDAAs.

Table 7. Common aSLA metrics

aSLA Metrics	Description
availability	The uptime of BDAA for end users in a specific time frame
financial cost	The total financial cost of using BDAA
respone time	Time to complete and receive the analysis result
usability	The degree to be easily used by end users through built-in interfaces
deadline	The total time of executing a BDAA and returning final results to its end user
reliability	The ability to maintain operational status in the majority of cases
integration	The degree of simplicity for integrating with applications and services require data from BDAA
capacity	The capacity the BDAA can provision
scalability	The ability to scale when expanding large volume of data or vast number of users
customizability	The flexibility to use with diverse kinds of users
pay-per-use billing	The ability to charge based on the usage of resources or duration
security	The degree to be exempt from malicious attack incurred by the network, software, tools, process or human, which results in significant damage or loss
energy efficiency	The degree of overall energy consumption on a per unit level (e.g., per capita, per customer, per hour)
the ratio of the admitted workloads	The proportion between the permitted workloads quantity and the submitted workloads quantity by end users

Table 8. Niche aSLA Metrics by different types of BDAAs

BDAA	aSLA Metrics	Description
Topic detection and tracking application [21, 127]	• event detection delay • input throughput • output throughput	• the delay of detecting events such as earthquake, football matches • the number of input events that are processed during a period • the number of derived events that are produced during a period
Big data-based traffic congestion detection system [?]	• alert sending delay	• the delay of sending alerts of existing traffics
Large-scale ingestion of analytics events and logs application [6]	• log integrity • alert sending speed	• the percentage of logs that can be seen • the number of alerts sent per second
Business intelligence on big data [20]	• the delay of decision making process	• the delay of a decision that is made based on business intelligence
Social network driven stock recommendation system [98, 143]	• data analysis delay	• the delay of completing stock analysis based on social network data
SLA based healthcare big data analysis and computing in Cloud network [100]	• disease prediction accuracy	• the degree to accurately forecast patients' prospective disease condition
Google smart inventory management system [2]	• inventory accuracy	• the degree to grasp accurate information regarding inventory and product at any time

Table 9. Common pSLA metrics

pSLA Metrics	Description
availability	The uptime of each framework in BDPaaS in a specific time
integration	The abilities to integrate with other frameworks and platforms
capacity	The capacity the BD platform can provision
scalability	The abilities to expand platform-level resources as requests or workloads increase
pay-per-user billing	The ability of the charging based on which framework or time of utilization
energy efficiency	The degree of energy consumption on a per unit level (e.g., per capita, per customer, per hour) for each framework
security	The degree to be free from malicious attack incurred by software or tools in each framework at BDPaaS layer, which brings damage or loss
fault tolerance	The ability to maintain an appropriate operational status even in the case of failures within its components

Table 10. Niche pSLA metrics by the different framework at BDPaaS layer

Framework	Sub Category	pSLA Metrics
Data Ingestion	Batch-based	● data size ● the number of chunks ● chunk size ● throughput ● the number of data format supported
	Stream-based	● data size ● data arrival rate ● latency ● the number of data format supported
Data Storage	Distributed File Systems (HDFS)	● transfer rate (read, write) ● latency (read, write, update) ● the number of data nodes ● replication number ● the overall size of input data ● the size of split data ● network throughput
	NoSQL	● transfer rate (read, write) ● latency (read, write, update) ● the quantity of data nodes ● replication factors ● queries speed ● transaction response time ● data freshness
	RDBMS	● queries speed ● query throughput ● the number of connections ● buffer pool usage ● transfer rate (read, write) ● latency (read, write, update) ● replication factors ● batch requests/sec ● disk read I/O per sec ● disk write I/O per sec
Data Processing	Batch-based	● the number of map tasks ● the number of reduce tasks ● the number of unhealthy nodes ● the number of active nodes ● the number of instances ● upper bound on the time finishing the data processing task ● block size ● job turnaround ● maximum allowed completion time
	Stream-based	● response time to streaming data ● stream processing latency ● peak system resource usage ● system start-up time ● Jitter (the variance of processing times)
Data Analysis	Statistics-based	how good is the statistical method (error rate, sensitivity, validity)
	Machine learning-based	● how good is the machine learning model (precision, recall, accuracy, sensitivity, specificity) ● model training time ● model training speed ● the size of the machine learning model ● average response speed for individual prediction requests ● number of algorithms supported for data analysis

Table 11. Common cSLA metrics

cSLA Metrics	Description
availability	the uptime of Cloud infrastructure in specific time
capacity	The capacity that the Cloud infrastructure can provision
scalability	The ability to expand infrastructure-level resources (e.g., VMs) requested from BDPaaS level
pay as you go billing	The ability to charge based on time of utilization of VMs or storages
energy efficiency	The degree of energy consumption for data centers
security	the degree to be exempt from malicious attack incurred by Cloud infrastructure, which causes damage or loss

Table 12. Niche cSLA metrics by different components at CIaaS layer

Component	cSLA Metrics
Computing resources	● response time ● CPU utilization ● memory utilization ● system load ● scale up time ● number of VMs ● number of cores per CPU ● memory size ● duration of individual VM migration
Storage resources	● number of units of data storage ● storage size ● privacy ● backup ● hard disk utilization ● I/O speed (bytes per second) ● failure frequency ● maximum downtime
Network resources	● network throughput ● network bandwidth ● network latency ● accessibility to the Internet across the firewall

7 Acknowledgments

Dr. Chen's work was supported in part by the National Natural Science Foundation of China (No. 61772380), Major Project for Technological Innovation of Hubei Province (No. 2019AAA044), and the Science and Technology Major Project of Hubei Province (Next-Generation AI Technologies, No. 2019AEA170).

References

[1] 2019. Amazon Comprehend. https://aws.amazon.com/comprehend/

[2] 2019. Building Real-Time Inventory Systems for Retail. https://cloud.google.com/solutions/building-real-time-inventory-systems-retail

[3] 2019. Marketing Cloud Platform Overview. https://www.salesforce.com/au/products/marketing-cloud/platform

[4] June 2014. Yarn Scheduler Load Simulator (SLS). https://hadoop.apache.org/docs/r2.4.1/hadoop-sls/SchedulerLoadSimulator.html/

[5] November 2017. Google Prediction API and Google BigQuery SLA. https://cloud.google.com/bigquery/sla

[6] October 2018. Architecture: Optimizing Large-Scale Ingestion of Analytics Events and Logs. https://cloud.google.com/solutions/architecture/optimized-large-scale-analytics-ingestion/

[7] Mohammed Alhamad, Tharam Dillon, and Elizabeth Chang. 2010. Conceptual SLA framework for cloud computing. In *Digital Ecosystems and Technologies (DEST), 2010 4th IEEE International Conference on*. IEEE, 606–610.

[8] Khalid Alhamazani, Rajiv Ranjan, Prem Prakash Jayaraman, Karan Mitra, Meisong Wang, Zhiqiang George Huang, Lizhe Wang, and Fethi Rabhi. 2014. Real-time qos monitoring for cloud-based big data analytics applications in mobile environments. In *Mobile Data Management (MDM), 2014 IEEE 15th International Conference on*, Vol. 1. IEEE, 337–340.

[9] Ahmad B Alnafoosi and Theresa Steinbach. 2013. An integrated framework for evaluating big-data storage solutions-IDA case study. In *Science and Information Conference (SAI), 2013*. IEEE, 947–956.

[10] Mohammed Alrokayan, Amir Vahid Dastjerdi, and Rajkumar Buyya. 2014. Sla-aware provisioning and scheduling of cloud resources for big data analytics. In *2014 IEEE International Conference on Cloud Computing in Emerging Markets (CCEM)*. IEEE, 1–8.

[11] Mauro Andreolini, Michele Colajanni, Marcello Pietri, and Stefania Tosi. 2015. Adaptive, scalable and reliable monitoring of big data on clouds. *J. Parallel and Distrib. Comput.* 79 (2015), 67–79.

[12] Zhi-guang Ao, Ming-hai Jiao, Ke-ning Gao, and Xing-wei Wang. 2016. Research on Cloud Resource Optimization Model Based on Users' Satisfaction. In *Web Information Systems and Applications Conference, 2016 13th*. IEEE, 99–102.

[13] Marcos D Assunção, Rodrigo N Calheiros, Silvia Bianchi, Marco AS Netto, and Rajkumar Buyya. 2015. Big Data computing and clouds: Trends and future directions. *J. Parallel and Distrib. Comput.* 79, 3–15.

[14] William H Bell, David G Cameron, Luigi Capozza, A Paul Millar, Kurt Stockinger, and Floriano Zini. 2002. Simulation of dynamic grid replication strategies in optorsim. In *International Workshop on Grid Computing*. Springer, 46–57.

[15] Paolo Bellavista, Antonio Corradi, Andrea Reale, and Nicola Ticca. 2014. Priority-based resource scheduling in distributed stream processing systems for big data applications. In *Proceedings of the 2014 IEEE/ACM 7th International Conference on Utility and Cloud Computing*. IEEE Computer Society, 363–370.

[16] Mihaly Berekmeri, Damián Serrano, Sara Bouchenak, Nicolas Marchand, and Bogdan Robu. 2014. A control approach for performance of big data systems. *IFAC Proceedings Volumes* 47, 3, 152–157.

[17] Rajkumar Buyya, Chee Shin Yeo, Srikumar Venugopal, James Broberg, and Ivona Brandic. 2009. Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility. *Future Generation computer systems* 25, 6, 599–616.

[18] Xiaojun Cai, Feng Li, Ping Li, Lei Ju, and Zhiping Jia. 2017. SLA-aware energy-efficient scheduling scheme for Hadoop YARN. *The Journal of Supercomputing* 73, 8, 3526–3546.

[19] CL Philip Chen and Chun-Yang Zhang. 2014. Data-intensive applications, challenges, techniques and technologies: A survey on Big Data. *Information Sciences* 275, 314–347.

[20] Hsinchun Chen, Roger HL Chiang, and Veda C Storey. 2012. Business intelligence and analytics: from big data to big impact. *MIS quarterly*, 1165–1188.

[21] Tao Cheng and Thomas Wicks. 2014. Event detection using Twitter: a spatio-temporal approach. *PloS one* 9, 6, e97807.

[22] Yeongho Choi and Yujin Lim. 2015. Resource management mechanism for SLA provisioning on cloud computing for IoT. In *2015 International Conference on Information and Communication Technology Convergence (ICTC)*. IEEE, 500–502.

[23] Yeongho Choi and Yujin Lim. 2016. Optimization approach for resource allocation on cloud computing for iot. *International Journal of Distributed Sensor Networks* 2016, 23.

[24] I-Hsun Chuang, Yu-Ting Huang, Wei-Tsung Su, Tung-Sheng Lin, and Yau-Hwang Kuo. 2015. S4: An SLA-aware Short-Secret-Sharing cloud storage system. In *2015 Seventh International Conference on Ubiquitous and Future Networks*. IEEE, 401–406.

[25] Amit Kumar Das, Tamal Adhikary, Md Abdur Razzaque, Majed Alrubaian, Mohammad Mehedi Hassan, Md Zia Uddin, and Biao Song. 2017. Big media healthcare data processing in cloud: a collaborative resource management perspective. *Cluster Computing* 20, 2, 1599–1614.

[26] Lucia De Marco, Filomena Ferrucci, and Tahar Kechadi. 2015. SLAFM: A Service Level Agreements Formal Model for Cloud Computing. In *The 5th International Conference on Cloud Computing and Service Science (CLOSER 2015), Lisbon, Portugal, 20-22 May 2015.*

[27] Ramon Hugo de Souza, Paulo Arion Flores, Mário Antônio Ribeiro Dantas, and Frank Siqueira. 2016. Architectural recovering model for Distributed Databases: A reliability, availability and serviceability approach. In *2016 IEEE Symposium on Computers and Communication (ISCC)*. IEEE, 575–580.

[28] Laouratou Diallo, Aisha-Hassan A Hashim, Rashidah Funke Olanrewaju, Shayla Islam, and Abdullah Ahmad Zarir. 2016. Two objectives big data task scheduling using swarm intelligence in cloud computing. *Indian Journal of Science and Technology* 9, 28.

[29] Djawida Dib, Nikos Parlavantzas, and Christine Morin. 2014. SLA-based profit optimization in cloud bursting PaaS. In *Cluster, Cloud and Grid Computing (CCGrid), 2014 14th IEEE/ACM International Symposium on*. IEEE, 141–150.

[30] Mouhamad Dieye, Mohamed Faten Zhani, and Halima Elbiaze. 2017. On achieving high data availability in heterogeneous cloud storage systems. In *2017 IFIP/IEEE Symposium on Integrated Network and Service Management (IM)*. IEEE, 326–334.

[31] Sofia D'Souza and K Chandrasekaran. 2015. Analysis of MapReduce scheduling and its improvements in cloud environment. In *Signal Processing, Informatics, Communication and Energy Systems (SPICES), 2015 IEEE International Conference on*. IEEE, 1–5.

[32] Todd Escalona. January 2018. Detect sentiment from customer reviews using Amazon Comprehend. https://aws.amazon.com/blogs/machine-learning/detect-sentiment-from-customer-reviews-using-amazon-comprehend/

[33] Funmilade Faniyi and Rami Bahsoon. 2016. A systematic review of service level management in the cloud. *ACM Computing Surveys (CSUR)* 48, 3, 43.

[34] Victor AE Farias, Flavio RC Sousa, Jose Gilvan R Maia, Joao Paulo P Gomes, and Javam C Machado. 2018. Regression based performance modeling and provisioning for NoSQL cloud databases. *Future Generation Computer Systems* 79, 72–81.

[35] Anshul Gandhi, Sidhartha Thota, Parijat Dube, Andrzej Kochut, and Li Zhang. 2016. Autoscaling for Hadoop clusters. In *2016 IEEE International Conference on Cloud Engineering (IC2E)*. IEEE, 109–118.

[36] Eugenio Gianniti, Danilo Ardagna, Michele Ciavotta, and Mauro Passacantando. 2017. A game-theoretic approach for runtime capacity allocation in MapReduce. In *Proceedings of the 17th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing*. IEEE Press, 1080–1089.

[37] Adam Gregory and Shikharesh Majumdar. 2016. A configurable energy aware resource management technique for optimization of performance and energy consumption on clouds. In *2016 IEEE International Conference on Cloud Computing Technology and Science (CloudCom)*. IEEE, 184–192.

[38] Adam Gregory and Shikharesh Majumdar. 2016. A constraint programming based energy aware resource management middleware for clouds processing MapReduce jobs with deadlines. In *Companion Publication for ACM/SPEC on International Conference on Performance Engineering*. ACM, 15–20.

[39] Adam Gregory and Shikharesh Majumdar. 2016. Energy aware resource management for MapReduce jobs with service level agreements in cloud data centers. In *2016 IEEE International Conference on Computer and Information Technology (CIT)*. IEEE, 568–577.

[40] Lin Gu, Deze Zeng, Song Guo, Yong Xiang, and Jiankun Hu. 2016. A general communication cost optimization framework for big data stream processing in geo-distributed data centers. *IEEE Trans. Comput.* 65, 1, 19–29.

[41] Lin Gu, Deze Zeng, Peng Li, and Song Guo. 2014. Cost minimization for big data processing in geo-distributed data centers. *IEEE transactions on Emerging topics in Computing* 2, 3, 314–323.

[42] Muhammad Hanif, Hyungduk Yoon, Sunglim Jang, and Choonhwa Lee. 2017. An adaptive SLA-based data flow mechanism for stream processing engines. In *Information and Communication Technology Convergence (ICTC), 2017 International Conference on*. IEEE, 81–86.

[43] Ibrahim Abaker Targio Hashem, Nor Badrul Anuar, Mohsen Marjani, Abdullah Gani, Arun Kumar Sangaiah, and Adewole Kayode Sakariyah. 2018. Multi-objective scheduling of MapReduce jobs in big data processing. *Multimedia Tools and Applications* 77, 8, 9979–9994.

[44] S Hemalatha and S Valarmathi. 2016. Efficient Hybrid framework for parallel Resource and task scheduling in the Map reduce programming. In *2016 International Conference on Computer Communication and Informatics (ICCCI)*. IEEE, 1–7.

[45] Raymond Hoare, Jiyong Ahn, and Jesse Graves. 2002 of Conference. Discrete event simulator. Google Patents.

[46] Christoph Hochreiner, Michael Vögler, Stefan Schulte, and Schahram Dustdar. 2017. Cost-efficient enactment of stream processing topologies. *PeerJ Computer Science* 3, e141.

[47] Chao-Wen Huang, Wan-Hsun Hu, Chia-Chun Shih, Bo-Ting Lin, and Chien-Wei Cheng. 2013. The improvement of auto-scaling mechanism for distributed database-A case study for MongoDB. In *2013 15th Asia-Pacific Network Operations and Management Symposium (APNOMS)*. IEEE, 1–3.

[48] Pham Phuoc Hung, Tuan-Anh Bui, Kwon Soonil, and Eui-Nam Huh. 2016. A new technique for optimizing resource allocation and data distribution in mobile cloud computing. *Elektronika ir Elektrotechnika* 22, 1, 73–80.

[49] Walayat Hussain, Farookh Khadeer Hussain, Omar K Hussain, Ernesto Damiani, and Elizabeth Chang. 2017. Formulating and managing viable SLAs in cloud computing from a small to medium service provider's viewpoint: A state-of-the-art review. *Information Systems* 71, 240–259.

[50] Eunji Hwang and Kyong Hoon Kim. 2012. Minimizing cost of virtual machines for deadline-constrained mapreduce applications in the cloud. In *Proceedings of the 2012 ACM/IEEE 13th International Conference on Grid Computing*. IEEE Computer Society, 130–138.

[51] Shigeru Imai, Stacy Patterson, and Carlos A Varela. 2017. Maximum sustainable throughput prediction for data stream processing over public clouds. In *Proceedings of the 17th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing*. IEEE Press, 504–513.

[52] Gabriel Iuhasz and Ioan Dragan. 2015. An overview of monitoring tools for big data and cloud applications. In *Symbolic and Numeric Algorithms for Scientific Computing (SYNASC), 2015 17th International Symposium on*. IEEE, 363–366.

[53] Ali Imran Jehangiri, Ramin Yahyapour, Philipp Wieder, Edwin Yaqub, and Kuan Lu. 2014. Diagnosing cloud performance anomalies using large time series dataset analysis. In *Cloud Computing (CLOUD), 2014 IEEE 7th International Conference on*. IEEE, 930–933.

[54] Selvi Kadirvel and José AB Fortes. 2011. Towards self-caring mapreduce: Proactively reducing fault-induced execution-time penalties. In *High Performance Computing and Simulation (HPCS), 2011 International Conference on*. IEEE, 63–71.

[55] Hyejeong Kang, Jung-in Koh, Yoonhee Kim, and Jaegyoon Hahm. 2013. A SLA driven VM auto-scaling method in hybrid cloud environment. In *Network Operations and Management Symposium (APNOMS), 2013 15th Asia-Pacific*. IEEE, 1–6.

[56] Karim Kanoun, Cem Tekin, David Atienza, and Mihaela Van Der Schaar. 2016. Big-data streaming applications scheduling based on staged multi-armed bandits. *IEEE Trans. Comput.* 65, 12, 3591–3605.

[57] Banpreet Kaur and Ankit Grover. 2016. Optimizing VM Provisioning of MapReduce Tasks on Public Cloud. In *Proceedings of the International Conference on Advances in Information Communication Technology & Computing*. ACM, 79.

[58] Barbara Kitchenham, O Pearl Brereton, David Budgen, Mark Turner, John Bailey, and Stephen Linkman. 2009. Systematic literature reviews in software engineering–a systematic literature review. *Information and software technology* 51, 1 (2009), 7–15.

[59] Panya Kittipipattanathaworn and Natawut Nupairoj. 2014. SLA guarantee real-time monitoring system with soft deadline constraint. In *Computer Science and Software Engineering (JCSSE), 2014 11th International Joint Conference on*. IEEE, 52–57.

[60] KR Krish, M Safdar Iqbal, M Mustafa Rafique, and Ali R Butt. 2014. Towards energy awareness in hadoop. In *Network-Aware Data Management (NDM), 2014 Fourth International Workshop on*. IEEE, 16–22.

[61] Maria Krotsiani, Christos Kloukinas, and George Spanoudakis. 2017. Validation of Service Level Agreements using Probabilistic Model Checking. In *Services Computing (SCC), 2017 IEEE International Conference on*. IEEE, 148–155.

[62] Taher Labidi, Achraf Mtibaa, and Hayet Brabra. 2016. CSLAOnto: a comprehensive ontological SLA model in cloud computing. *Journal on Data Semantics* 5, 3, 179–193.

[63] Palden Lama and Xiaobo Zhou. 2012. Aroma: Automated resource allocation and configuration of mapreduce environment in the cloud. In *Proceedings of the 9th international conference on Autonomic computing*. ACM, 63–72.

[64] Jonathan Lejeune, Frederico Alvares, and Thomas Ledoux. 2017. Towards a generic autonomic model to manage Cloud Services. In *The 7th International Conference on Cloud Computing and Services Science (CLOSER 2017)*.

[65] Sanping Li, Yu Cao, Simon Tao, Xiaoyan Guo, Zhe Dong, and Ricky Sun. 2015. An extensible framework for predictive analytics on cost and performance in the cloud. In *2015 International Conference on Cloud Computing and Big Data (CCBD)*. IEEE, 13–20.

[66] Norman Lim, Shikharesh Majumdar, and Peter Ashwood-Smith. 2014. A constraint programming-based resource management technique for processing MapReduce jobs with SLAs on clouds. In *Parallel Processing (ICPP), 2014 43rd International Conference on*. IEEE, 411–421.

[67] Norman Lim, Shikharesh Majumdar, and Peter Ashwood-Smith. 2014. Engineering resource management middleware for optimizing the performance of clouds processing mapreduce jobs with deadlines. In *Proceedings of the 5th ACM/SPEC international conference on Performance engineering*. ACM, 161–172.

[68] Norman Lim, Shikharesh Majumdar, and Peter Ashwood-Smith. 2014. Resource management techniques for handling requests with service level agreements. In *International Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPECTS 2014)*. IEEE, 618–625.

[69] Norman Lim, Shikharesh Majumdar, and Peter Ashwood-Smith. 2017. MRCP-RM: a technique for resource allocation and scheduling of MapReduce jobs with deadlines. *IEEE Transactions on Parallel and Distributed Systems* 28, 5, 1375–1389.

[70] Norman Lim, Shikharesh Majumdar, and Peter Ashwood-Smith. 2017. A Run Time Technique for Handling Error in User-Estimated Execution Times on Systems Processing MapReduce Jobs with Deadlines. In *Future Internet of Things and Cloud (FiCloud), 2017 IEEE 5th International Conference on*. IEEE, 1–9.

[71] Norman Lim, Shikharesh Majumdar, and Peter Ashwood-Smith. 2017. Techniques for Handling Error in User-estimated Execution Times During Resource Management on Systems Processing MapReduce Jobs. In *Proceedings of the 17th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing*. IEEE Press, 788–793.

[72] Fotios K Liotopoulos and Petros Lampsas. 2015. Energy-efficient simulation and performance evaluation of large-scale data centers. In *2015 IEEE International Conference on Industrial Technology (ICIT)*. IEEE, 3121–3127.

[73] Qinghua Lu, Shanshan Li, Weishan Zhang, and Lei Zhang. 2016. A genetic algorithm-based job scheduling model for big data analytics. *EURASIP journal on wireless communications and networking* 2016, 1, 152.

[74] Qinghua Lu, Zheng Li, Weishan Zhang, and Laurence T Yang. 2017. Autonomic deployment decision making for big data analytics applications in the cloud. *Soft Computing* 21, 16, 4501–4512.

[75] Yang Lu. 2017. Industry 4.0: A survey on technologies, applications and open research issues. *Journal of Industrial Information Integration* 6, 1–10.

[76] Mohammad-Hossein Malekloo, Nadjia Kara, and May El Barachi. 2018. An energy efficient and SLA compliant approach for resource allocation and consolidation in cloud computing environments. *Sustainable Computing: Informatics and Systems* 17, 9–24.

[77] Xijun Mao, Chunlin Li, Wei Yan, and Shumeng Du. 2016. Optimal Scheduling Algorithm of MapReduce Tasks Based on QoS in the Hybrid Cloud. In *Parallel and Distributed Computing, Applications and Technologies (PDCAT), 2016 17th International Conference on*. IEEE, 119–124.

[78] Lena Mashayekhy, Mahyar Movahed Nejad, Daniel Grosu, Quan Zhang, and Weisong Shi. 2015. Energy-aware scheduling of mapreduce jobs for big data applications. *IEEE Transactions on Parallel & Distributed Systems* 1, 1–1.

[79] Michael Mattess, Rodrigo N Calheiros, and Rajkumar Buyya. 2013. Scaling mapreduce applications across hybrid clouds to meet soft deadlines. In *2013 IEEE 27th International Conference on Advanced Information Networking and Applications (AINA)*. IEEE, 629–636.

[80] Rizwan Mian, Patrick Martin, and Jose Luis Vazquez-Poletti. 2013. Provisioning data analytic workloads in a cloud. *Future Generation Computer Systems* 29, 6, 1452–1458.

[81] Akram Mohamadi and Sedigheh Barani. 2015. A review on approaches in service level agreement in cloud computing environment. In *2015 4th Iranian Joint Congress on Fuzzy and Intelligent Systems (CFIS)*. IEEE, 1–4.

[82] Saad Mubeen, Sara Abbaspour Asadollah, Alessandro V Papadopoulos, Mohammad Ashjaei, Hongyu Pei-Breivold, and Moris Behnam. 2017. Management of Service Level Agreements for Cloud Services in IoT: A Systematic Mapping Study. *IEEE Access*.

[83] Lekha R Nair, Sujala D Shetty, and Siddhanth D Shetty. 2018. Applying spark based machine learning model on streaming big data for health status prediction. *Computers & Electrical Engineering* 65, 393–399.

[84] Dimas C Nascimento, Carlos Eduardo Pires, and Demetrio Mestre. 2015. Data quality monitoring of cloud databases based on data quality slas. In *Big-Data Analytics and Cloud Computing*. Springer, 3–20.

[85] Deveeshree Nayak, Venkata Swamy Martha, David Threm, Srini Ramaswamy, Summer Prince, and Günter Fahrnberger. 2015. Adaptive scheduling in the cloud-SLA for Hadoop job scheduling. In *2015 Science and Information Conference (SAI)*. IEEE, 832–837.

[86] Mihaela-Catalina Nita, Cristian Chilipirea, Ciprian Dobre, and Florin Pop. 2013. A SLA-based method for big-data transfers with multi-criteria optimization constraints for IaaS. In *2013 11th RoEduNet International Conference*. IEEE, 1–6.

[87] Mihaela-Catalina Nita, Florin Pop, Cristiana Voicu, Ciprian Dobre, and Fatos Xhafa. 2015. MOMTH: multi-objective scheduling algorithm of many tasks in Hadoop. *Cluster computing* 18, 3, 1011–1024.

[88] Yoori Oh, Jieun Choi, Eunjung Song, Moonji Kim, and Yoonhee Kim. 2016. A SLA-based Spark cluster scaling method in cloud environment. In *2016 18th Asia-Pacific Network Operations and Management Symposium (APNOMS)*. IEEE, 1–4.

[89] Ahmed Oussous, Fatima-Zahra Benjelloun, Ayoub Ait Lahcen, and Samir Belfkih. 2017. Big Data technologies: A survey. *Journal of King Saud University-Computer and Information Sciences*.

[90] Balaji Palanisamy, Aameek Singh, and Ling Liu. 2015. Cost-effective resource provisioning for mapreduce in a cloud. *IEEE Transactions on Parallel and Distributed Systems* 26, 5, 1265–1279.

[91] Tadeusz Pankowski. 2015. Consistency and availability of Data in replicated NoSQL databases. In *Evaluation of Novel Approaches to Software Engineering (ENASE), 2015 International Conference on*. IEEE, 102–109.

[92] Adrian Paschke and Elisabeth Schnappinger-Gerull. 2006. A Categorization Scheme for SLA Metrics. *Service Oriented Electronic Commerce* 80, 25-40, 14.

[93] Jorda Polo, Yolanda Becerra, David Carrera, Malgorzata Steinder, Ian Whalley, Jordi Torres, and Eduard Ayguade. 2013. Deadline-based MapReduce workload management. *IEEE Transactions on Network and Service Management* 10, 2, 231–244.

[94] K Hima Prasad, Tanveer A Faruquie, L Venkata Subramaniam, Mukesh Mohania, and Girish Venkatachaliah. 2010. Resource allocation and SLA determination for large data processing services over cloud. In *2010 IEEE International Conference on Services Computing*. IEEE, 522–529.

[95] Xuanjia Qiu, Wai Leong Yeow, Chuan Wu, and Francis CM Lau. 2013. Cost-minimizing preemptive scheduling of mapreduce workloads on hybrid clouds. In *2013 IEEE/ACM 21st International Symposium on Quality of Service (IWQoS)*. IEEE, 1–6.

[96] Joy Rahman and Palden Lama. 2017. MPLEX: In-Situ Big Data Processing with Compute-Storage Multiplexing. In *Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS), 2017 IEEE 25th International Symposium on*. IEEE, 43–52.

[97] B Kezia Rani and A Vinaya Babu. 2015. Scheduling of Big Data application workflows in cloud and inter-cloud environments. In *2015 IEEE International Conference on Big Data (Big Data)*. IEEE, 2862–2864.

[98] Rajiv Ranjan, Joanna Kolodziej, Lizhe Wang, and Albert Y Zomaya. 2015. Cross-layer cloud resource configuration selection in the big data era. *IEEE Cloud Computing* 3, 16–22.

[99] Radhya Sahal, Mohamed H Khafagy, and Fatma A Omara. 2016. A Survey on SLA Management for Cloud Computing and Cloud-Hosted Big Data Analytic Applications. *International Journal of Database Theory and Application* 9, 4, 107–118.

[100] Prasan Kumar Sahoo, Suvendu Kumar Mohapatra, and Shih-Lin Wu. 2018. SLA based healthcare big data analysis and computing in cloud network. *J. Parallel and Distrib. Comput.* 119, 121–135.

[101] Sherif Sakr and Anna Liu. 2012. Sla-based and consumer-centric dynamic provisioning for cloud databases. In *Cloud Computing (CLOUD), 2012 IEEE 5th International Conference on*. IEEE, 360–367.

[102] Omran Saleh, Francis Gropengieβer, Heiko Betz, Waseem Mandarawi, and Kai-Uwe Sattler. 2013. Monitoring and autoscaling IaaS clouds: a case for complex event processing on data streams. In *Proceedings of the 2013 IEEE/ACM 6th International Conference on Utility and Cloud Computing*. IEEE Computer Society, 387–392.

[103] Rajinder Sandhu and Sandeep K Sood. 2015. Scheduling of big data applications on distributed cloud based on QoS parameters. *Cluster Computing* 18, 2, 817–828.

[104] Carla Sauvanaud, Mohamed Kaâniche, Karama Kanoun, Kahina Lazri, and Guthemberg Da Silva Silvestre. 2018. Anomaly Detection and Diagnosis for Cloud services: Practical experiments and lessons learned. *Journal of Systems and Software* 139 (2018), 84–106.

[105] Damián Serrano, Sara Bouchenak, Yousri Kouki, Frederico Alvares de Oliveira Jr, Thomas Ledoux, Jonathan Lejeune, Julien Sopena, Luciana Arantes, and Pierre Sens. 2016. SLA guarantees for cloud services. *Future Generation Computer Systems* 54, 233–246.

[106] Damián Serrano, Sara Bouchenak, Yousri Kouki, Thomas Ledoux, Jonathan Lejeune, Julien Sopena, Luciana Arantes, and Pierre Sens. 2013. Towards qos-oriented sla guarantees for online cloud services. In *2013 13th IEEE/ACM International Symposium on Cluster, Cloud, and Grid Computing*. IEEE, 50–57.

[107] M Omair Shafiq and Eric Torunski. 2017. Towards MapReduce based Bayesian deep learning network for monitoring big data applications. In *Big Data (Big Data), 2017 IEEE International Conference on*. IEEE, 2112–2121.

[108] Jin Shao and Qianxiang Wang. 2011. A performance guarantee approach for cloud applications based on monitoring. In *2011 35th IEEE Annual Computer Software and Applications Conference Workshops*. IEEE, 25–30.

[109] Yanling Shao, Chunlin Li, Wenyong Dong, and Yunchang Liu. 2016. Energy-aware dynamic resource allocation on Hadoop YARN cluster. In *High Performance Computing and Communications; IEEE 14th International Conference on Smart City; IEEE 2nd International Conference on Data Science and Systems (HPCC/SmartCity/DSS), 2016 IEEE 18th International Conference on*. IEEE, 364–371.

[110] Yanling Shao, Chunlin Li, Jinguang Gu, Jing Zhang, and Youlong Luo. 2018. Efficient jobs scheduling approach for big data applications. *Computers & Industrial Engineering* 117, 249–261.

[111] Bikash Sharma, Timothy Wood, and Chita R Das. 2013. Hybridmr: A hierarchical mapreduce scheduler for hybrid data centers. In *2013 IEEE 33rd International Conference on Distributed Computing Systems*. IEEE, 102–111.

[112] Mingruo Shi and Ruiping Yuan. 2015. Mad: A monitor system for big data applications. In *International Conference on Intelligent Science and Big Data Engineering*. Springer, 308–315.

[113] Ming-Hung Shih and J Morris Chang. 2017. Design and analysis of high performance crypt-NoSQL. In *Dependable and Secure Computing, 2017 IEEE Conference on*. IEEE, 52–59.

[114] Kwang Mong Sim. 2006. A survey of bargaining models for grid resource allocation. *ACM SIGecom Exchanges* 5, 5, 22–32.

[115] Kwang Mong Sim. 2010. Grid resource negotiation: survey and new directions. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* 40, 3, 245–257.

[116] Mbarka Soualhia, Foutse Khomh, and Sofiène Tahar. 2017. Task scheduling in big data platforms: a systematic literature review. *Journal of Systems and Software* 134, 170–189.

[117] Andre Abrantes DP Souza and Marco AS Netto. 2015. Using application data for sla-aware auto-scaling in cloud environments. In *Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS), 2015 IEEE 23rd International Symposium on*. IEEE, 252–255.

[118] Xi Sun, Bo Gao, Liya Fan, and Wenhao An. 2012. A cost-effective approach to delivering analytics as a service. In *Web services (icws), 2012 ieee 19th international conference on*. IEEE, 512–519.

[119] Yangyang Tao, Shucheng Yu, and Junxiu Zhou. 2018. Information Flow Queue Optimization in EC Cloud. In *2018 International Conference on Computing, Networking and Communications (ICNC)*. IEEE, 888–892.

[120] Fengguang Tian and Keke Chen. 2011. Towards optimal resource provisioning for running mapreduce programs in public clouds. In *Cloud Computing (CLOUD), 2011 IEEE International Conference on*. IEEE, 155–162.

[121] Rafael Tolosana-Calasanz, José Ángel Bañares, Congduc Pham, and Omer F Rana. 2016. Resource management for bursty streams on multi-tenancy cloud environments. *Future Generation Computer Systems* 55, 444–459.

[122] Linjiun Tsai, Hubertus Franke, Chung-Sheng Li, and Wanjiun Liao. 2018. Learning-Based Memory Allocation Optimization for Delay-Sensitive Big Data Processing. *IEEE Transactions on Parallel and Distributed Systems* 29, 6, 1332–1341.

[123] Radu Tudoran, Olivier Nano, Ivo Santos, Alexandru Costan, Hakan Soncu, Luc Bougé, and Gabriel Antoniu. 2014. Jetstream: Enabling high performance event streaming across cloud data-centers. In *Proceedings of the 8th ACM International Conference on Distributed Event-Based Systems*. ACM, 23–34.

[124] Abhishek Verma, Ludmila Cherkasova, and Roy H Campbell. 2011. ARIA: automatic resource inference and allocation for mapreduce environments. In *Proceedings of the 8th ACM international conference on Autonomic computing*. ACM, 235–244.

[125] Chen Wang, Junliang Chen, Bing Bing Zhou, and Albert Y Zomaya. 2012. Just satisfactory resource provisioning for parallel applications in the cloud. In *2012 IEEE Eighth World Congress on Services*. IEEE, 285–292.

[126] Guanying Wang, Ali R Butt, Prashant Pandey, and Karan Gupta. 2009. Using realistic simulation for performance analysis of mapreduce setups. In *Proceedings of the 1st ACM workshop on Large-Scale system and application performance*. ACM, 19–26.

[127] Meisong Wang, Rajiv Ranjan, Prem Prakash Jayaraman, Peter Strazdins, Pete Burnap, Omer Rana, and Dimitrios Georgakopulos. 2015. A Case for Understanding End-to-End Performance of Topic Detection and Tracking Based Big Data Applications in the Cloud. In *International Internet of Things Summit*. Springer, 315–325.

[128] Yang Wang and Wei Shi. 2013. On optimal budget-driven scheduling algorithms for MapReduce jobs in the hetereogeneous cloud. *Technical Report TR-13–02, Carleton Univ.* (2013).

[129] Yang Wang and Wei Shi. 2014. Budget-driven scheduling algorithms for batches of MapReduce jobs in heterogeneous clouds. *IEEE Transactions on Cloud Computing* 2, 3, 306–319.

[130] Jonathan Stuart Ward and Adam Barker. 2014. Observing the clouds: a survey and taxonomy of cloud monitoring. *Journal of Cloud Computing* 3, 1, 24.

[131] Md Whaiduzzaman, Mohammad Nazmul Haque, Md Rejaul Karim Chowdhury, and Abdullah Gani. 2014. A study on strategic provisioning of cloud computing services. *The Scientific World Journal* 2014.

[132] Philipp Wieder, Jan Seidel, Oliver Wäldrich, Wolfgang Ziegler, and Ramin Yahyapour. 2008. Using sla for resource management and scheduling-a survey. In *Grid Middleware and Services*. Springer, 335–347.

[133] Xiaoyong Xu, Maolin Tang, and Yu-Chu Tian. 2016. Theoretical results of QoS-guaranteed resource scaling for cloud-based MapReduce. *IEEE Transactions on Cloud Computing*.

[134] Xiaoyong Xu, Maolin Tang, and Yu-Chu Tian. 2018. QoS-guaranteed resource provisioning for cloud-based MapReduce in dynamical environments. *Future Generation Computer Systems* 78, 18–30.

[135] Chaowei Yang, Qunying Huang, Zhenlong Li, Kai Liu, and Fei Hu. 2017. Big Data and cloud computing: innovation opportunities and challenges. *International Journal of Digital Earth* 10, 1, 13–53.

[136] Zhihao Yao, Ioannis Papapanagiotou, and Robert D Callaway. 2014. SLA-aware resource scheduling for cloud storage. In *Cloud Networking (CloudNet), 2014 IEEE 3rd International Conference on*. IEEE, 14–19.

[137] S Yasmin and S Jessica Sritha. 2017. A constraint programming-based resource allocation and scheduling of map reduce jobs with service level agreement. In *2017 International Conference on Energy, Communication, Data Analytics and Soft Computing (ICECDS)*. IEEE, 3589–3594.

[138] Abdulsalam Yassine, Ali Asghar Nazari Shirehjini, and Shervin Shirmohammadi. 2016. Bandwidth on-demand for multimedia big data transfer across geo-distributed cloud data centers. *IEEE Transactions on Cloud Computing*.

[139] Xiaoqun Yuan, Geyong Min, Laurence T Yang, Yi Ding, and Qing Fang. 2017. A game theory-based dynamic resource allocation strategy in geo-distributed datacenter clouds. *Future Generation Computer Systems* 76, 63–72.

[140] Bernard P Zeigler, Tag Gon Kim, and Herbert Praehofer. 2000. *Theory of modeling and simulation*. Academic press.

[141] Xuezhi Zeng, Saurabh Garg, Zhenyu Wen, Peter Strazdins, Lizhe Wang, and Rajiv Ranjan. 2016. SLA-aware scheduling of map-Reduce applications on public clouds. In *2016 IEEE 18th International Conference on High Performance Computing and Communications; IEEE 14th International Conference on Smart City; IEEE 2nd International Conference on Data Science and Systems (HPCC/SmartCity/DSS)*. IEEE, 655–662.

[142] Xuezhi Zeng, Saurabh Kumar Garg, Zhenyu Wen, Peter Strazdins, Albert Y Zomaya, and Rajiv Ranjan. 2017. Cost efficient scheduling of MapReduce applications on public clouds. *Journal of computational science*.

[143] Rui Zhang, Reshu Jain, Prasenjit Sarkar, and Lukas Rupprecht. 2014. Getting your big data priorities straight: a demonstration of priority-based qos using social-network-driven stock recommendation. *Proceedings of the VLDB endowment* 7, 13, 1665–1668.

[144] Liang Zhao, Sherif Sakr, and Anna Liu. 2015. A framework for consumer-centric SLA management of cloud-hosted databases. *IEEE Transactions on Services Computing* 8, 4, 534–549.

[145] Yali Zhao, Rodrigo N Calheiros, James Bailey, and Richard Sinnott. 2016. SLA-based profit optimization for resource management of big data analytics-as-a-service platforms in cloud computing environments. In *2016 IEEE International Conference on Big Data (Big Data)*. IEEE, 432–441.

[146] Yali Zhao, Rodrigo N Calheiros, Graeme Gange, Kotagiri Ramamohanarao, and Rajkumar Buyya. 2015. SLA-based resource scheduling for big data analytics as a service in cloud computing environments. In *2015 44th International Conference on Parallel Processing*. IEEE, 510–519.

[147] Qin Zheng. 2010. Improving MapReduce fault tolerance in the cloud. In *2010 IEEE International Symposium on Parallel & Distributed Processing, Workshops and Phd Forum (IPDPSW)*. IEEE, 1–6.

[148] Liudong Zuo and Michelle M Zhu. 2015. Concurrent bandwidth reservation strategies for big data transfers in high-performance networks. *IEEE Transactions on Network and Service Management* 12, 2, 232–247.

Appendix:
SLA Management for Big Data Analytical Applications in Clouds: A Taxonomy Study

ZENG XUEZHI, Australian National University, Australia
SAURABH GARG, University of Tasmania, Australia
MUTAZ BARIKA, University of Tasmania, Australia
ALBERT Y. ZOMAYA, University of Sydney, Australia
LIZHE WANG, China University of Geoscience (Wuhan), China
MASSIMO VILLARI, University of Messina, Italy
DAN CHEN, Wuhan University, China
RAJIV RANJAN, China University of Geoscience (Wuhan, China) and Newcastle University, UK

ACM Reference Format:
Zeng Xuezhi, Saurabh Garg, Mutaz Barika, Albert Y. Zomaya, Lizhe Wang, Massimo Villari, Dan Chen, and Rajiv Ranjan. 2019. Appendix:
SLA Management for Big Data Analytical Applications in Clouds: A Taxonomy Study. 1, 1 (November 2019), 13 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

A Background

A.1 Concept: Big Data, Big Data Analytics and Applications

While it is ubiquitous today, big data does not have an absolute, precise and agreed upon definition [42, 92, 98]. In this paper, we considered the definition issued by NIST on September 2015 [24] that Big Data contains large amount of datasets in the four principal characteristics (i.e., volume, variety, velocity, variability) that highly demands advanced technologies and architectures to efficiently store, manipulate, and analyze.

The potential of big data is unleashed when it becomes possible to mine analysis and insights from it. This results in the emergence of big data analytics that provides vast opportunities for extant and prospective organizations to construct valuable information from big data [67?]. According to [30, 33, 71], big data analytics is an advanced technology to examine and transform a significant amount of datasets that consists of multiple data types for improving the comprehension of data. After analyzing data, correct tools and approaches are used to create decent visualizations of the findings in various format (e.g., tables, 2D and 3D graph) for superior decision making. The primary purpose of big data analytics is to enable organizations to uncover hidden patterns and reveal unseen correlations from huge datasets and thus make effective decisions.

The power of big data analytics is usually released through an emerging type of software applications namely big data analytical applications (BDAAs). Generally, BDAAs leverage large-scale distributed processing frameworks (e.g., Hadoop [76]) to analyze big data. Big data analytics has found applications across a wide array of areas

Authors' addresses: Zeng Xuezhi, Australian National University, Research School of Computer Science, Australia; Saurabh Garg, University of Tasmania, School of TED, Australia; Mutaz Barika, University of Tasmania, School of TED, Australia; Albert Y. Zomaya, University of Sydney, School of IT, Australia; Lizhe Wang, China University of Geoscience (Wuhan), School of Computer Science, China; Massimo Villari, University of Messina, Italy; Dan Chen, Wuhan University, School of Computer Science, China; Rajiv Ranjan, China University of Geoscience (Wuhan, China) and Newcastle University, UK.

2019. XXXX-XXXX/2019/11-ART $15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

such as financial and banking services, telecommunications, digital media, healthcare, manufacturing, and others [37, 72]. Leading providers provide a wide scope of different types of BDAAs. For example, Google utilizes Google BigQuery [11] to offer inventory management system [9], an abundant, highly scalable, low cost and pay-as-you-go BDAA to make inventory management productive and efficient. Amazon provides natural language processing-based BDAA in clouds that identifies the language of voluminous texts, extracts vital entities such as people, organizations, locations or events, and analyze sentiments in texts using Amazon Comprehend [3].

A.2 Cloud-hosted big data analytical applications

Nowadays, we have witnessed an increasing trending practice that many organizations deploy and operate their big data analytical applications (BDAAs) in clouds to reap even more benefits of big data. This is because big data (BD) and cloud computing (CC) are two technologies that are often conjoined and increasingly incorporated together [38, 57]. BD is a method for optimizing the data analytic platforms in terms of implementation scalability, deployment flexibility, execution robustness, cost effectiveness and so on [64]. CC is the next revolution in computing paradigm that can efficiently utilize resources without compromising scalability, security, the time and cost of execution and so on. CC complements and benefits BD in the form of promptly provisioning and releasing a shared and configurable infrastructure (i.e., computing resources, resources and network resources) that are readily accessible in a pay-as-you-go economic model with minimal cost of management or interaction [73]. Therefore, it makes sense that organizations should look to CC as the resource provisioning platform to support their BDAAs.

The provision of such BDAAs in cloud could bring benefits to organizations by easing adoption and saving considerable cost. Also, it could generate useful insights and construct them different types of competitive advantages. In fact, increasing organizations opt for cloud nowadays. According to the blog post in August 2017 written by Brian Hopkins who is the vice president of Forrester [2], Global capital spending on big data solutions through cloud subscriptions will expand roughly 7.5 times speedier than the traditional on-premises subscriptions. Moreover, the public cloud ranked the top technology for big data according to the surveys of data analytics professionals over year 2016 and 2017". Also, a business intelligence to Hadoop/big data connection company AtScale [16] conducted a three-year survey on how global companies use big data and the cloud. The survey findings unsurprisingly revealed that cloud is effectively taking center stage for big data use. The survey has found that there has been an increased amount of attention to the deployment of BDAAs in clouds, with 59% of respondents has deployed BDAAs in clouds already. Additionally, over three quarters (77%) are projecting they would use the cloud for big data up 5% from last year. These demonstrate that cloud-based BDAAs is a widespread phenomenon nowadays.

A.3 Layer-based Architecture of Cloud-hosted Big Data Analytic Applications

According to the works in [57, 59, 69, 82, 109], a typical cloud-hosted BDAA spans multiple layers. Each layer serves different function and consists of different components/frameworks. We give a pictorial representation of a layer-based architecture for cloud-hosted BDAAs, which is shown in Figure 1.

It is observed that there exist three layers from top to bottom: Big Data Software as a Service (BDSaaS), Big Data Platform as a Service (BDPaaS) and Cloud Infrastructure as a Service (CIaaS). Beyond the top level are usually end users who request analytics service through the interface. It is not difficult to understand that an end user is served as the client of the BDSaaS. Then, BDSaaS is served as the client of the BDPaaS. Next, BDPaaS is served as the client of the CIaaS. Under the level of CIaaS exists scalable hardware resources in datacenters. Each layer are detailed as follows.

Cloud Infrastructure as a Service (CIaaS): Cloud computing plays an integral role of underlying infrastructure in this framework located at the bottom layer. In this layer, users access three main elastic resources (i.e., computing-based, storage-based, and network-based resources) in a pay-per-use mode. This layer provides

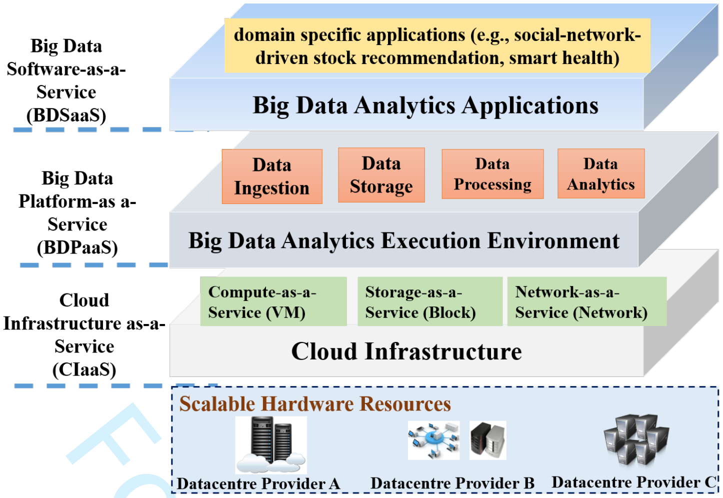
Big Data Software-as-a-Service (BDSaaS)
domain specific applications (e.g., social-network-driven stock recommendation, smart health)
Big Data Analytics Applications
Big Data Platform-as a-Service (BDPaaS)
Data Ingestion
Data Storage
Data Processing
Data Analytics
Big Data Analytics Execution Environment
Cloud Infrastructure as-a-Service (CIaaS)
Compute-as-a-Service (VM)
Storage-as-a-Service (Block)
Network-as-a-Service (Network)
Cloud Infrastructure
Scalable Hardware Resources
Datacentre Provider A
Datacentre Provider B
Datacentre Provider C

Fig. 1. The layer-based architecture of big data analytical applications in clouds

the most opportunities for direct influence on big data technology (scalability, availability, computing, and accessibility of raw data). Amazon EC2 [4] is an excellent choice in this layer offering high-performance and on-demand computing resources, analogous to a server in a datacenter. Amazon Simple Storage Service [6] is storage resources analogous to an array of hard drives.

Big Data Platform as a Service (BDPaaS): The platform layer includes software, tools or frameworks dedicated to BDAAs. For instance, Hadoop and its ecosystem [76]. BDPaaS emphasizes on the provision of a specialized execution environment for processing the applications, which are often instantiated over CIaaS to ensure scalability, availability, reliability, and so on. BDPaaS typically includes multiple frameworks/components as follows.

- Data Ingestion: This is the first step for the data coming from variable sources to start its journey, which is responsible for ingesting data into BDAAs. It includes two types of ingestion (i.e., batch and stream). Batch ingestion represents traditional data integration tools and software (e.g.,extract, transform, load). Stream ingestion consists of simple messaging systems and complicated event-based processing engines [69]. Sample technologies for batch ingestion are Apache Sqoop [91]. Apache Kafka is a good example of stream ingestion [50]
- Data Storage: Represents technologies for persistent data storage for BDAAs. It is characterized by different types of storage such as traditional RDBMS (e.g., MySQL, Oracle etc.,), distributed file systems (e.g., Hadoop HDFS [62], Google File System [52]), and NoSQL (e.g., MongoDB [40], Hadoop HBase [95]) data stores. RDBMSs are mature and well support structured data, while unstructured data prefer NoSQL or distributed file systems, which is a reasonable choice for BDAAs
- Data Processing: Deals with technologies for the execution and computation of voluminous data from data ingestion framework. It is mainly classified into batch-based processing and stream-based processing paradigm. For batch processing, voluminous data is first stored and then processed and analyzed at once [59]. The algorithm exploits an efficient divide-and-conquer approach that split the dataset into chunks and processes each chunk on a individual machine where intermediate output are generated and eventually aggregated to a final result. Apache Hadoop MapReduce [43] has become the dominant batch processing model. The streaming processing paradigm refers to the technology that queries continuous data streams and detects conditions quickly from the time of receiving the data at the second or even millisecond level. Apache Spark [105] and Apache Storm [7] are representative examples of stream processing

Table 1. Summary of the layered architecture of big data analytical applications in clouds

Layer	Component	Representative Examples	Characteristics of Big Data
BDSaaS	A wide range of domains	• Salesforce.com's Marketing Cloud [13] • BrandsEye [8] • Google's inventory management system [9]	Value, Validity
BDPaaS	Data Ingestion	• Batch (e.g., Sqoop) • Stream (e.g., Kafka)	Velocity
	Data Storage	• DFS (e.g.,Hadoop HDFS) • NoSQL (e.g., MongoDB) • RDBMS (e.g., Oracle)	Volume, Variety
	Data Processing	• Batch (e.g., Hadoop MapReduce) • Stream (e.g., Apache Storm)	Velocity, Variety
	Data Analysis	• the type of data analytics (descriptive, predictive and prescriptive) • the type of approaches (statistics-based method and machine learning-based method)	Value, Veracity
CIaaS	Computing	Amazon EC2, Google Compute Engine, Azure Virtual Machines	N/A
	Storage	Amazon S3, Google Cloud Storage, Azure Storage (Block Blob)	N/A
	Network	Amazon VPC, Google VPC, Azure Virtual Network	N/A

- Data Analytics: Comprises technologies that are responsible for uncovering unknown patterns and unraveling invisible correlations. Hence useful insights for better decision making are extracted and real value is generated, which is the ultimate objective of BDAAs. This framework is differentiated by two dimensions: the genre of data analytics and the genre of approaches. The genre of data analytics comprises descriptive, predictive and prescriptive [31, 32, 67, 97]. The genre of approaches includes statistics-based method [74, 78, 104] and machine learning-based methods [45, 99, 104]. Some representative examples of BDSaaS include Google BigQuery [11], Microsoft Data Lake Analytics [14] and Amazon Redshift [5]

Big Data Software as a Service (BDSaaS): This top-tiered layer typically provides specific BDAAs interfaces that enable users to focus on one particular domain of business or private concern and do not mention any underlying Cloud resources nor BDPaaS level components. Typically, BDSaaS is a web-based and multi-tenant system that analyzes and interprets massive amounts of data to deliver more insightful results for their subscribers. Users would develop and execute scripts and queries to analyze and generate reports and visualizations [109]. BDSaaS is an extension of the common SaaS model, with the same delivery model benefits, but better because BDSaaS leverages the best of the underlying BDPaaS-level technologies (i.e., data ingestion, storage, processing, and analysis) in clouds to construct valuable information and gain real insight for users. Representative examples of BDSaaS are Salesforce.com's Marketing Cloud [13] or BrandsEye [8]. These BDSaaS can collect real-time social media data, process and analyze them through their featured data analytics platform and eventually produce insights regarding feedback on the effectiveness of new marketing promotions or forecast of prospective products problems [84].

Table 1 gives a summary of the above layered architecture. In comparison with on-premises environments such as traditional server farms, this architecture demonstrates high level of modularity and granularity. Each layer communicates with the neighboring layers and has the flexibility to evolve separately. The architecture integrates cloud computing (CC) that acts the underlying infrastructure in the whole architecture, which highlights the integral role that CC plays in BDAAs.

A.4 SLA and its Evolution

A.4.1 SLA Definition The term of SLA remains multifarious definitions so far.

Marilly et al. [70] denote SLA as a contract between customers and providers with measurable terms specified, what services customers will consume and what penalties provider will be credited if SLA violations incur".

Gartner defines SLA as an agreement between providers and customers that sets the expectations and specifies their delivered services in terms of SLA metrics, penalties and responsibilities of all parties etc., by which the process is monitored and approved [10].

IBM states that an SLA is a contract among providers and customers that defines the expectations regarding the promised service level and quality concerning multiple measurable objectives such as availability and performance. SLA establishes a shared understanding about the provisioned services content, service quality, obligations, guarantees, and penalties between parties [15].

As a collection of detailed best practices for IT service management, Information Technology Infrastructure Library (ITIL) [87] contributes a lot in spreading SLAs. It defines SLA as an agreement between customers and providers that specifies the content of IT service, service level objectives, and documents the obligations of all parties.

Based on these diversified definitions, in this paper, we consider SLAs as established agreements that govern the relationship between different actors. It encloses comprehensive aspects regarding the services to be provisioned. These include the promised quality of service (QoS) - expressed through multiple different terms, the service level objectives (SLOs) that the service must guarantee in the form of constraints on QoS metrics, individual actor's responsibilities and obligations, as well as the penalties incurred when SLAs are breached [65].

A.4.2 The Evolution of SLA Over the last thirty years, SLA has undergone significant evolution driven by the advancement of distributed computing paradigm in order to adapt for changes and new challenges in different computing environments per requirements. Based on previous works in [55, 61, 63, 79, 80, 85, 96, 106], the distributed computing paradigm has evolved several major phases that include Internet Computing [89], Peer-to-Peer Computing [88], Cluster Computing [36], Grid Computing [34], Utility Computing [83], Cloud Computing (CC) [73] and Big Data (BD) [25].

In the 1970s, the emergence of computer networks resulted in the introduction of distributed systems [28]. ARPANET (the predecessor of the Internet) was firstly developed as a network to serve academic institutions including some government-funded research laboratories and universities. Commercially, Internet service providers commence by the late 1980s [12]. Until now, some certain technologies began to emerge in the distributed systems. Peer-to-Peer network is regarded as one of fundamental distributed systems during this time with the purpose to enable sharing of data, such as streaming audio or video [55]. In the 1980s, Cluster Computing has emerged, which is used for high-performance computing tasks. Another well-known distributed computing paradigm is Grid Computing that appears in the mid-1990s as an evolution of Cluster Computing [55]. In the 2000s, Utility Computing was proposed based on the idea of providing computing solutions in a very similar way as conventional public utilities (i.e., gas, electricity, phone, water) in everyday life [80]. Utility Computing was the first step towards pay-by-use philosophy. Around 2007, CC has emerged as a popular distributed computing paradigm [93]. Recently, the further advancement of computer technologies and distributed processing paradigms have generated a new paradigm over the cloud at the forefront of BD. A representative example of such paradigm is MapReduce [66] programming model that is built for paralleled data-intensive computing in clouds. This has inspired an open source distributed computing framework called Apache Hadoop [76] and its ecosystem for cloud-hosted BDAAs.

As an essential and efficient method of managing relationships between providers and customers and guaranteeing the level of service, we examine that SLA has been successfully used in all the aforementioned distributed computing environments over the last thirty years. Accordingly, SLA has experienced remarkable evolution

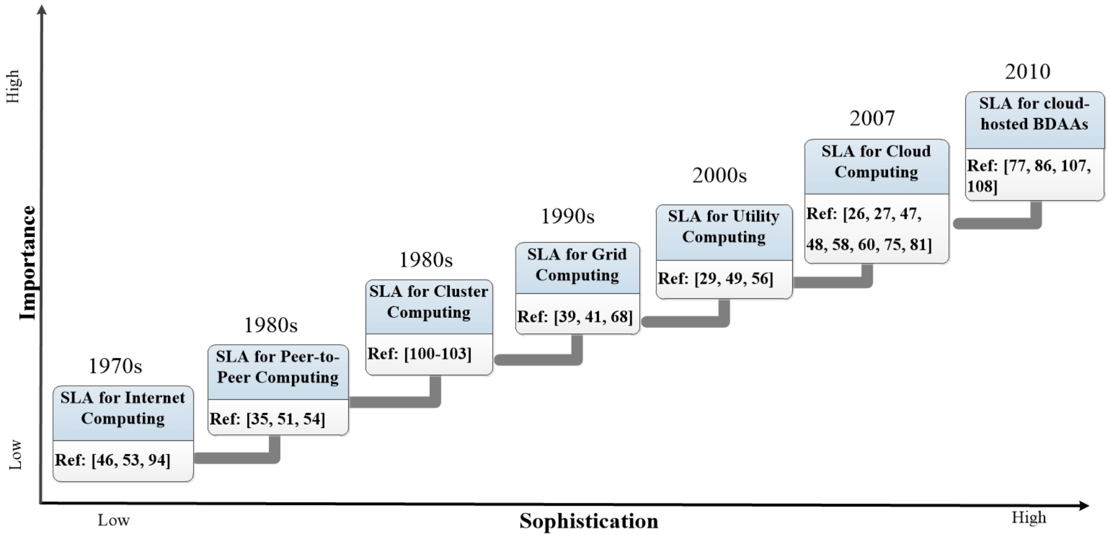

Fig. 2. SLA evolution stages with selected references

as the distributed computing paradigm advances to cater to changes and new challenges in each distinct computing environment. Concretely, the main stages in the SLA evolutionary roadmap include SLAs for Internet Computing [46, 53, 94], SLAs for Peer-to-Peer Computing [35, 51, 54], SLAs for Cluster Computing [100–103], SLA for Grid Computing [39, 41, 68], SLAs for Utility Computing [29, 49, 56], SLAs for Cloud Computing [26, 27, 47, 48, 58, 60, 75, 81, 90] and SLAs for cloud-hosted BDAAs [77, 86, 107, 108]. In this paper, we are not going to detail each stage considering space. Instead, we give a brief explanation regarding SLAs for Internet Computing, SLAs for Cloud Computing and SLAs for cloud-hosted BDAAs. Figure 2 proposes a pictorial representation of SLA evolution and some representative references in each evolutional stage.

Historically, SLAs have originated with internet service providers in the late 1980s, which forms the first stage (i.e., SLAs for Internet Computing). From this time, SLAs have been highly demanded by various providers and customers in the telecommunication marketplace. Hence, Internet service providers and telecoms have commonly incorporated SLAs into their contracts with corporate customers, where they use plain language to specify the service level being provisioned to customers [44]. Further, in order to provide better practice advice, the Tele Management Forum had published the NGOSS SLA Management Handbook in 2001 which represents a milestone of SLA in its evolution [1]. Up to this time, NGOSS SLA Management Handbook is the most complete publication regarding the management of SLAs with a focus on the Telecommunication Industry [87]. The drawbacks in this very early SLAs stage lie in the fact that their SLA metrics are limited to the performance measurement on IP-based network (i.e., packet loss and latency), and the specifications of their SLAs are too rigid to embody SLA terms values once they were established between providers and customers.

Since then, the SLA evolution continues from the stage of SLA for Peer-to-Peer computing to the stage of SLA for Utility Computing. During this evolution, significant advancements of SLA specification are particularly triggered by the emergence of Utility Computing and Grid Computing, as the autonomy and openness of these two computing paradigms required SLA specifications in the formats of adapting any semantic of application domains in any organization. Further, the fast growth of cloud application market leads to the rise of new types of services and forms of providing these services among cloud ecosystems, which drives the extensive exploitation of SLAs for cloud computing environment. The continuous technological advancement of distributed computing stimulates the application of SLA to the current stage of SLAs for cloud-hosted BDAAs. For the stage of SLA for cloud-hosted BDAAs, we will use a real cloud-hosted BDAA as an example to elaborate SLA importance and challenges in the next subsection.

B Detailed Example of SLAs Template for a Real Cloud-hosted BDAA

To aid understanding of the SLAs across different layers of BDAA, we take a real cloud-hosted BDAA as an illustrative example to present SLAs template. The selected BDAA example is a smart inventory management system for retail offered by Google as one of their referred BDAA solutions and use cases [9].

According to Google, a smart inventory system for retail are built to maintain an accurate and update information as anything changes in the inventory. For example, as soon as a retail sold an item or even when moved one part of store to another, inventory data should be refreshed in backend automatically. This brings lots of challenges for retails who increasingly struggle to work in scenarios such as online and offline mode as well as multiple channels. Hence, smart inventory management system comes into the picture to address these challenges, such that, retailers can efficiently work with buyers and assist customers locate products faster. Also, retailers can significantly benefit from operational efficiency by precisely perceiving the location and availability of product at any time. Moreover, the inventory accuracy and movement offers useful insights for retailers to fine-tune and better promote their products in market campaigns. To this end, retailers need to leverage the power of cloud-based BDAAs that offer scalable infrastructure, reduced administrative complicatedness, and the state-of-art techniques in terms of data ingestion, storage, process and analysis.

Figure 3 presents the architect of Google's smart inventory system for retail. The components at each layer, and the interactions and SLA requirements of each component are described as follows:

CIaaS layer

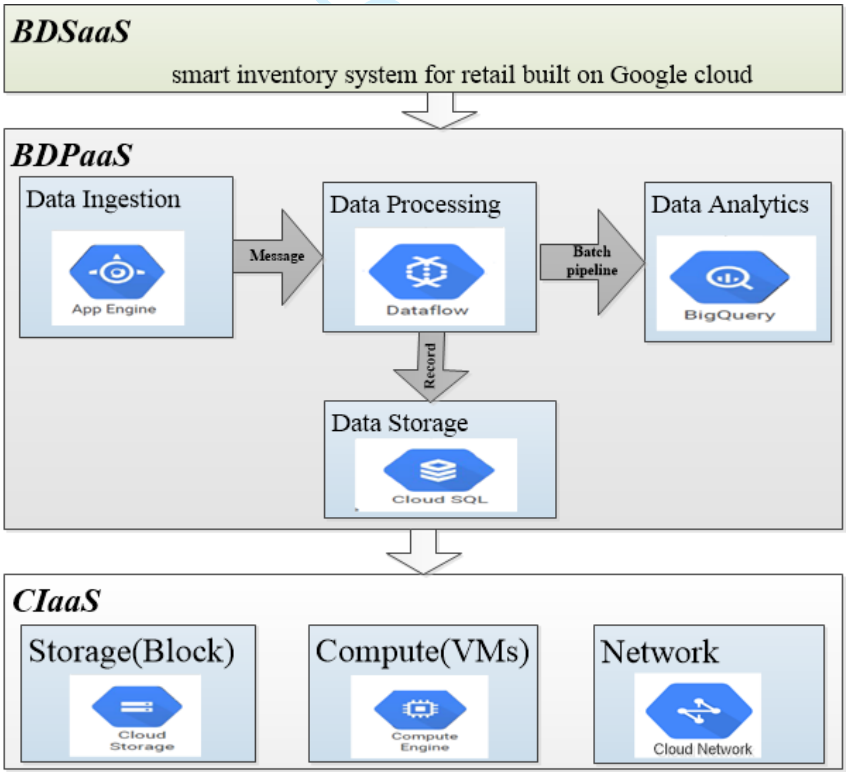

Fig. 3. Architecture of Google's inventory system for retail

- Compute Engine: It delivers multiple different types of VMs. Users can access Google's VMs using predefined sizes or creating customized machine types to suit particular business needs. These VMs serve the Cloud Dataflow at BDPaaS layer. This component must satisfy SLA requirements. For example, it must ensure availability to the customer at minimum 99.99% [18]
- Cloud Storage: Users can access data stored on Google cloud platform infrastructure through an online RESTful-based file storage web interface. This component must satisfy SLA requirements as well. For instance, it must guarantee at least 99.95% availability to customers [23]. Also, it must guarantee the security and confidentiality of an application and customer data
- Cloud Network: Cloud virtual private network (VPN) offers a secure VPN tunnel to connect customers' own network and Google's global network. One of SLA requirements that Google VPN must meet is to assure minimum 99.9% network availability to customers [22]

BDPaaS layer

- Data Ingestion: This regularly collects inventory data from multiple stores and proceeds it to subsequent services. A good example of data ingestion is Google App Engine that offers the automatic and real-time scaling abilities according to the future traffic patterns. SLA requirements are mandatory for this component. It must provide 99.95% above availability [20]. Moreover, it needs to meet the requirements regarding elastic scaling and minimum response time latency.
- Data Processing: This service includes two popular processing paradigms (i.e., batch or stream) for a smart inventory system. A good example of data processing is Google's Cloud Dataflow that offers the abilities to transform and enrich data both in batch-based and stream-based workloads, and distribute these processing workloads across many VMs instances. In terms of SLA requirements, it must efficiently process batch or stream jobs with low-latency response time. The monthly uptime percentage (availability) defined by Google is 99.5% [21].
- Data Storage: This service is dedicated to record and maintain accurate inventory data at any time. Once new inventory events either through purchase or shipping happen, inventory database is then updated in an automatic and real-time form. Google Cloud SQL is a well-suited choice for data storage, which is deployed in MySQL and provides a native support regarding backup, replication and recovery. Still, SLA requirements are mandatory for this component. For example, it must support high-speed queries for counts from the inventory. In the documentation of Google cloud SQL SLA, the availability is set to be greater than 99.95% [17].
- Data Analytics: This collects incoming inventory streams and load them into BigQuery where specific analysis is performed and actionable insights are generated. BigQuery is a state-of-art Google's data warehouse solution. It can execute queries over TB amount of data within seconds. Without exception, this service must satisfy SLA requirements as well. For example, it must guarantee the availability that is greater than 99.9% [19]. Also, it must guarantee queries speed across TB volume data in seconds, and the high accuracy of the queries result regarding the inventory at all times.

BDSaaS layer. Google's inventory system is designed to provide a high accurate, visible, and analytical platform for inventory movements throughout the supply chain at any time. This service must satisfy SLA requirements such as availability, usability, scalability, integration, response time, financial cost, accuracy and query speed. Customers can then choose service level objectives (SLOs) they want to apply for aSLA (application-level SLA). For instance, response time is managed lower than a designated threshold, financial cost is efficiently controlled without exceeding a given bar, agreed availability level is greater than 95%, and specification of penalties applied when SLA violations incur.

For this application, its SLAs consist of aSLA, pSLA and cSLA. Figure 4 presents its cross-layer structure of SLAs metrics. It is worth noting that this cross-layer structure of SLA metrics still applies to other BDAAs

Table 2. Example of aSLA, pSLA, and cSLA for Google's inventory system

aSLA- SLA between customers and Google		
	SLO	90% of requests handled by Google smart inventory management BDAA will be processed in < 500ms
	Guarantee	Google smart inventory management BDAA must ensure availability at least 99%
	Violation	If the response time of Google's inventory management as agreed is not met during any period, all requests for that period will be free of charge. If availability is < 99%, then 10% of service credit will be applied
pSLA-SLA between BDSaaS and BDPaaS		
Date Ingestion	SLO	For a maximal capital cost of USD 0.056/vCPU per hour (batch type) or USD 0.069/vCPU per hour (streaming type) to the Data Ingestion framework in BDPaaS (e.g., Google App Engine), the maximum response time must be 1 second per ingestion
	Guarantee	The data ingestion service (i.e., Google App Engine) will be available of at minimum 99.95%
	Violation	If availability is < 99.95%, then 10% of service credit will be enforced
Date Storage	SLO	Requests to the Data Storage framework in BDPaaS (e.g., Google CloudSQL), the response time has to be lower than 1 second
	Guarantee	The data storage service (i.e., Google Cloud SQL)will be available of at least 99.95%
	Violation	If availability is < 99.95%, then 10% of service credit will be practised
Date Processing	SLO	Regarding a maximal financial expense of USD 0.056/vCPU per hour (batch type) or USD 0.069/vCPU per hour (streaming type) to the Data Processing framework in BDPaaS (e.g., Google Dataflow), response time has to be smaller than 1 second per dataflow job
	Guarantee	The data processing service (i.e., Google Dataflow) will be available of at least 99.5%
	Violation	If more than 2% of queries to the Data Processing framework(i.e., Google Dataflow) in BDPaaS violate SLOs, the customer will receive the financial credits by 0.02/violated dataflow jobs If availability is < 99.5%, then 10% service credit will be charged
Date Analysis	SLO	Requests to the Data Analysis framework in BDPaaS (e.g., Google BigQuery), response time has to be no more than 1 second
	Guarantee	The data analysis service (i.e., Google BigQuery) will be available of 99.9% at least
	Violation	If availability is < 99.5%, then 10% of service credit will be applied
cSLA-SLA between BDPaaS and CIaaS		
Compute	SLO	For a maximal capital cost of USD 0.0100/hour per VM instance of the cloud infrastructure computing service (i.e., Google Compute Engine), at least 10 VMs, each configured with one vCPU and 3.75GB memory must be provided
	Guarantee	Availability to customer of at least 99.99% should be provided
	Violation	If more than 1% of the time VMs at CIaaS layer has breached SLOs, USD 0.3/hour per the violated VM instance is penalized
Storage	SLO	For a maximal capital cost of USD 0.026 GB/month of the Cloud infrastructure storage service (i.e., Google Cloud Storage), at least 1TB multi-regional storage must be provided
	Guarantee	The data storage service (i.e., Google Cloud SQL)will be available of 99.95% at least
	Violation	10% of service credit will be imposed when availability is lower than 99.95%
Network	SLO	For a maximum financial cost of USD 0.050 per tunnel per hour of the cloud infrastructure network service (i.e., Google Cloud Network), at least 100 tunnels and 100GB bandwidth must be provided
	Guarantee	The cloud network service (i.e., Google Cloud network) will be available of 99.9% at minimum
	Violation	If availability is < 99.9%, then 10% of service credit will be enforced

although we take the Google smart inventory management system as an instance. Table 2 elaborately describes an illustrated example of this application's aSLA, pSLA and cSLA.

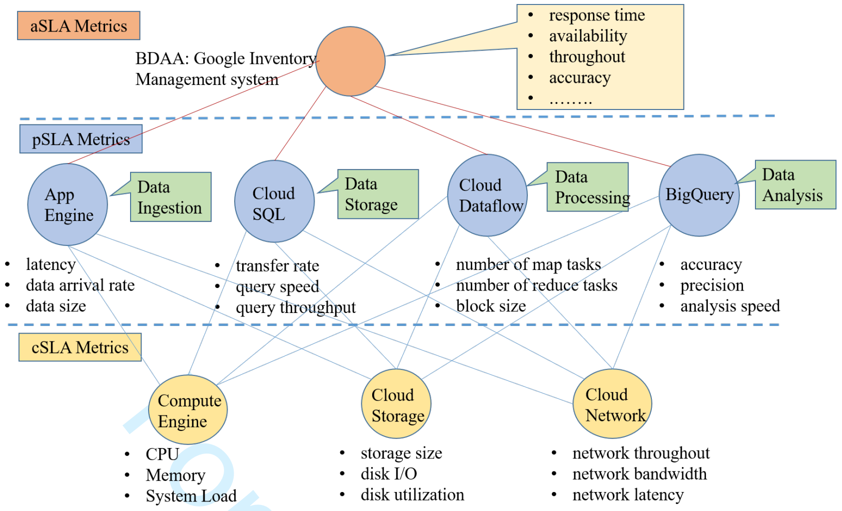

Fig. 4. Cross-layer SLA metrics for Google's inventory system

References

[1] 2014. GB917 SLA Management Handbook, Volume 2: Concepts and Principles Release 2.5. https://www.tmforum.org/resources/standard/gb917-sla-management-handbook-volume-2-concepts-and-principles-release-2-5-zip/

[2] 2017. Move Big Data To The Public Cloud With An Insight PaaS. https://go.forrester.com/blogs/insight-paas-accelerate-big-data-cloud

[3] 2019. Amazon Comprehend. https://aws.amazon.com/comprehend/

[4] 2019. Amazon Elastic Compute Cloud (EC2). https://aws.amazon.com/ec2/

[5] 2019. Amazon Redshift. https://aws.amazon.com/redshift/

[6] 2019. Amazon Simple Storage Service (S3). https://aws.amazon.com/s3/

[7] 2019. Apache Storm. http://storm.apache.org/

[8] 2019. Bandseye website. https://www.brandseye.com/

[9] 2019. Building Real-Time Inventory Systems for Retail. https://cloud.google.com/solutions/building-real-time-inventory-systems-retail

[10] 2019. Gartner IT Glossary: Service Level Agreement. https://www.gartner.com/it-glossary/sla-service-level-agreement/

[11] 2019. Google Query. https://cloud.google.com/bigquery/

[12] 2019. History of the Internet. https://en.wikipedia.org/wiki/History_of_the_Internet/

[13] 2019. Marketing Cloud Platform Overview. https://www.salesforce.com/au/products/marketing-cloud/platform

[14] 2019. Microsoft Data Lake Analytics. https://azure.microsoft.com/en-us/services/data-lake-analytics/

[15] 2019. Understanding service level agreements. https://www.ibm.com/support/knowledgecenter/en/SS9H2Y_7.6.0/com.ibm.dp.doc/sla_understanding_wsp.html

[16] 2019. The Website of Atscale. https://www.atscale.com/

[17] April 2017. Google Cloud SQL Service Level Agreement (SLA). https://cloud.google.com/sql/sla/

[18] March 2019. Google Compute Engine Service Level Agreement (SLA). https://cloud.google.com/compute/sla/

[19] November 2017. Google Prediction API and Google BigQuery SLA. https://cloud.google.com/bigquery/sla

[20] October 2016. Google App Engine Service Level Agreement (SLA). https://cloud.google.com/appengine/sla/

[21] October 2016. Google Cloud Dataflow Service Level Agreement (SLA). https://cloud.google.com/dataflow/sla/

[22] October 2016. Google VPN Service Level Agreement (SLA). https://cloud.google.com/vpn/sla/

[23] October 2018. Google Cloud Storage Service Level Agreement (SLA). https://cloud.google.com/storage/sla/

[24] September 2015. NIST Special Publication 1500-1. https://bigdatawg.nist.gov/_uploadfiles/NIST.SP.1500-1.pdf

[25] Divyakant Agrawal, Sudipto Das, and Amr El Abbadi. 2011. Big data and cloud computing: current state and future opportunities. In *Proceedings of the 14th International Conference on Extending Database Technology*. ACM, 530–533.

[26] Mohammed Alhamad, Tharam Dillon, and Elizabeth Chang. 2010. Conceptual SLA framework for cloud computing. In *Digital Ecosystems and Technologies (DEST), 2010 4th IEEE International Conference on*. IEEE, 606–610.

[27] Eman Aljoumah, Fajer Al-Mousawi, Imtiaz Ahmad, Maha Al-Shammri, and Zahraa Al-Jady. 2015. SLA in cloud computing architectures: A comprehensive study. *International Journal of Grid Distribution Computing* 8, 5, 7–32.

[28] Gregory R Andrews. 2000. Foundations of multithreaded, parallel, and distributed programming, Vol. 11. Addison-Wesley Reading.

[29] Karen Appleby, Sameh Fakhouri, Liana Fong, Germán Goldszmidt, Michael Kalantar, Srirama Krishnakumar, Donald P Pazel, John Pershing, and Benny Rochwerger. 2001. Oceano-SLA based management of a computing utility. In *2001 IEEE/IFIP International Symposium on Integrated Network Management Proceedings. Integrated Network Management VII. Integrated Management Strategies for the New Millennium (Cat. No. 01EX470)*. IEEE, 855–868.

[30] Claudio A Ardagna, Paolo Ceravolo, and Ernesto Damiani. 2016. Big data analytics as-a-service: Issues and challenges. In *2016 IEEE International Conference on Big Data (Big Data)*. IEEE, 3638–3644.

[31] Marcos D Assunção, Rodrigo N Calheiros, Silvia Bianchi, Marco AS Netto, and Rajkumar Buyya. 2015. Big Data computing and clouds: Trends and future directions. *J. Parallel and Distrib. Comput.* 79, 3–15.

[32] Feras A Batarseh and Eyad Abdel Latif. 2016. Assessing the quality of service using big data analytics: with application to healthcare. *Big Data Research* 4, 13–24.

[33] Mininath R Bendre and Vijaya R Thool. 2016. Analytics, challenges and applications in big data environment: a survey. *Journal of Management Analytics* 3, 3, 206–239.

[34] Fran Berman, Geoffrey Fox, Tony Hey, and Anthony JG Hey. 2003. Grid computing: making the global infrastructure a reality, Vol. 2. John Wiley and sons.

[35] Nico Brehm, Jorge Marx Gómez, and Claus Rautenstrauch. 2005. An ERP solution based on web services and peer-to-peer networks for small and medium enterprises. *International Journal of Information Systems and Change Management* 1, 1, 99–111.

[36] Rajkumar Buyya et al. 1999. High performance cluster computing: Architectures and systems (volume 1). *Prentice Hall, Upper SaddleRiver, NJ, USA* 1, 999.

[37] Hsinchun Chen, Roger HL Chiang, and Veda C Storey. 2012. Business intelligence and analytics: from big data to big impact. *MIS quarterly*, 1165–1188.

[38] Min Chen, Shiwen Mao, and Yunhao Liu. 2014. Big data: A survey. *Mobile networks and applications* 19, 2, 171–209.

[39] Adrian Ching, Lionel Sacks, and Paul McKee. 2003. Sla management and resource modelling for grid computing. In *London Communications Symposium (LCS 2003), London, UK*.

[40] Kristina Chodorow. 2013. *MongoDB: the definitive guide: powerful and scalable data storage.* " O'Reilly Media, Inc.".

[41] Christopher J Dawson, Roderick E Legg, and Erik Severinghaus. 2008. Management of grid computing resources based on service level requirements. US Patent App. 11/765,487.

[42] Andrea De Mauro, Marco Greco, and Michele Grimaldi. 2015. What is big data? A consensual definition and a review of key research topics. In *AIP conference proceedings*, Vol. 1644. AIP, 97–104.

[43] Jeffrey Dean and Sanjay Ghemawat. 2008. MapReduce: simplified data processing on large clusters. *Commun. ACM* 51, 1, 107–113.

[44] Jianguo Ding. 2016. *Advances in network management.* Auerbach Publications.

[45] K Sree Divya, P Bhargavi, and S Jyothi. 2018. Machine Learning Algorithms in Big data Analytics.

[46] Edward James Ellesson, Roch Andre Guerin, Sanjay Damodar Kamat, Arvind Krishna, Rajendran Rajan, and Dinesh Chandra Verma. 2002. Architecture for supporting service level agreements in an IP network. US Patent 6,459,682.

[47] Vincent C Emeakaroha, Marco AS Netto, Rodrigo N Calheiros, Ivona Brandic, Rajkumar Buyya, and César AF De Rose. 2012. Towards autonomic detection of SLA violations in Cloud infrastructures. *Future Generation Computer Systems* 28, 7, 1017–1029.

[48] Funmilade Faniyi and Rami Bahsoon. 2016. A systematic review of service level management in the cloud. *ACM Computing Surveys (CSUR)* 48, 3, 43.

[49] Victor AE Farias, Flavio RC Sousa, Jose Gilvan R Maia, Joao Paulo P Gomes, and Javam C Machado. 2018. Regression based performance modeling and provisioning for NoSQL cloud databases. *Future Generation Computer Systems* 79, 72–81.

[50] Nishant Garg. 2013. *Apache Kafka.* Packt Publishing Ltd.

[51] Jan Gerke and David Hausheer. 2005. 29. Peer-to-Peer Market Management. In *Peer-to-Peer Systems and Applications*. Springer, 491–507.

[52] Sanjay Ghemawat, Howard Gobioff, and Shun-Tak Leung. 2003. The Google file system.

[53] R Jo Gibbens, SK Sargood, FP Kelly, H Azmoodeh, R Macfadyen, and N Macfadyen. 2000. An approach to service level agreements for IP networks with differentiated services. *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences* 358, 1773, 2165–2182.

[54] Ankur Gupta and Lalit K Awasthi. 2010. Toward a quality-of-service framework for peer-to-peer applications. *International Journal of Distributed Systems and Technologies (IJDST)* 1, 3, 1–23.

[55] Majid Hajibaba and Saeid Gorgin. 2014. A review on modern distributed computing paradigms: Cloud computing, jungle computing and fog computing. *Journal of computing and information technology* 22, 2, 69–84.

[56] Irfan Ul Haq. 2010. A framework for SLA-centric service-based utility computing.

[57] Ibrahim Abaker Targio Hashem, Ibrar Yaqoob, Nor Badrul Anuar, Salimah Mokhtar, Abdullah Gani, and Samee Ullah Khan. 2015. The rise of "big data" on cloud computing: Review and open research issues. *Information Systems* 47, 98–115.