Journal of Internet Services and Applications
a SpringerOpen Journal

RESEARCH Open Access

MediaWise cloud content orchestrator

Rajiv Ranjan*, Karan Mitra and Dimitrios Georgakopoulos

Abstract

The growing ubiquity of Internet and cloud computing is having significant impact on media-related industries. These industries are using the Internet and cloud as a medium to enable creation, search, management and consumption of their content. Primarily, Content Delivery Networks (CDNs) are deployed for distributing multimedia content to the end-users. However, existing approaches to architecting CDNs have several limitations. Firstly, they do not harness multiple public cloud services for optimizing cost to performance ratio. Secondly, they lack support for dynamic and personalized content creation and distribution. Finally, they do not support end-to-end content lifecycle operations (production, deployment, consumption, personalization, and distribution).
To overcome these limitations, in this paper, we propose, develop and validate a novel system called MediaWise Cloud Content Orchestrator (MCCO). MCCO expands the scope of existing CDNs with novel multi-cloud deployment. It enables content personalization and collaboration capabilities. Further, it facilitates do-it-yourself creation, search, management, and consumption of multimedia content. It inherits the pay-as-you-go models and elasticity that are offered by commercially available cloud services.
In this paper, we discuss our vision, the challenges and the research objectives pertaining to MCCO for supporting next generation streamed, interactive, and collaborative high resolution multimedia content. We validated our system thorugh MCCO prototype implementation. Further, we conducted a set of experiments to demonstrate the functionality of MCCO. Finally, we compare the content orchestration features supported by MCCO to existing CDNs against the envisioned objectives of MCCO.

Keywords: Content delivery network, Cloud computing, Media management, Media delivery, Media consumption, Personalization, Quality of service

1. Introduction

Internet is having a significant impact on the media-related industries which are using it as a medium to enable delivery of their content to end-users. Rich web pages, software downloads, interactive communications, and ever-expanding universe of digital media require a new approach to content delivery. Size and volume of multimedia content is growing exponentially. For example, more than 30 billion pieces of content such as web links, news stories, blog posts, notes, and photo albums are shared each month on Facebook. On the other hand, Twitter users are tweeting an average 55 million tweets a day that includes web links and photo albums. Web pages and other multimedia content are being delivered through content delivery networks (CDN) [1] technologies. These technologies optimize network usage through dedicated network links, caching servers and by increasingly using peer-to-peer technologies. The concept of a CDN was conceived in the early days of Internet but it took until the end of 1990's before CDNs from Akamai and other commercial providers managed to deliver Web content (i.e., web pages, text, graphics, URLs and scripts) anywhere in the world and at the same time meet the high availability and quality expected by their end users. For example, Akamai [2] delivers between fifteen to thirty percent of all Web traffic, reaching more than 4 Terabits per second. Commercial CDNs achieved this by deploying a private collection of servers and by using distributed CDN software system in multiple data centres around the world.

A different variant of CDN technology appeared in the mid 2000's to support the streaming of hundreds of high definition channels to paid customers. These CDNs had to deal with more stringent Quality of Service (QoS) requirements to support users' experience pertaining to high definition video. This required active management

* Correspondence: rajiv.ranjan@csiro.au
CSIRO ICT Centre, Canberra, Australia

of the underlying network resources and the use of specialized set-top boxes that included video recorders (providing stop/resume and record/playback functionality) and hardware decoders (e.g., providing MPEG 4 video compression/decompression). Major video CDNs where developed by telecommunications companies that owned the required network and had Operation Support Systems (OSSs) to manage the network QoS as required by the CDN to preserve the integrity of high definition video content. Just like the original CDNs, video CDN also utilize a private collection of servers distributed around the network of video service provider. The first notable CDNs in this category include Verizon's FiOS and AT&T's U-verse. Some CDN providers such as Limelight Networks invested billions of dollars in building dedicated network links (media-grade fiber-optic backbone) for delivering and moving content from servers to end-users.

A more recent variant of video CDNs involves the caching video content in cloud storage and the distribution of such content using third-party network services that are designed to meet QoS requirements of caching and streaming high definition video. For example, Netflix's video CDN has been developed on top of Amazon AWS. CloudFront is Amazon's own CDN that uses Amazon AWS and provides streaming video services using Microsoft Xboxes. While Cloud-based CDNs [3,4] have made a remarkable progress in the past five years, they are still limited in the following aspects:

- CDN service providers either own all the services they use to run their CDN services or they outsource this to a single cloud provider. A specialized legal and technical relationship is required to make the CDN work in the latter case.
- Video CDNs are not designed to manage content (e.g., find and play high definition movies). This is typically done by CDN applications. For example, CDNs do not provide services that allow an individual to create a streaming music video service combining music videos from an existing content source on the Internet (e.g., YouTube), his/her personal collection, and from live performances he/she attends using his/her smart phone to capture such content. This can only be done by an application managing where and when the CDN will deliver the video component of his/her music program.
- CDNs are designed for streaming staged content but do not perform well in situations where content is produced dynamically. This is typically the case when content is produced, managed and consumed in collaborative activities. For example, an art teacher may find and discuss movies from different film archives, the selected movies may then be edited by students. Parts of them may be used in producing new movies that can be sent to the students' friends for comments and suggestions. Current CDNs do not support such collaborative activities that involve dynamic content creation.

In [5], we proposed the MediaWise cloud which facilitates the collaborative content production and deployment using public clouds. This paper builds on [5]. However, compared to [5], in this paper, we propose, develop and validate the MediaWise Cloud Content Orchestrator (MCCO). MCCO is an enabler for MediaWise cloud facilitating content orchestration operations (e.g., content production and deployment) across cloud service layers [3] including Software as a Service (SaaS), Platform as a Service (PaaS) and Infrastructure as a Service (IaaS) layers. MCCO aims to address the shortcomings of current CDN technologies. Compared to [5], we perform extensive experiments to validate MCCO. MCCO [a] aims to address the shortcomings of current CDN technologies. In particular, MCCO makes the following contributions:

- Unlike existing commercial CDN providers such as Limelight Networks and Akamai, MCCO eliminates the need to own and manage expensive infrastructure while facilitating content owner requirements pertaining to price, SLA, privacy and QoS. Instead, it can utilize the cloud storage and CPU resources from virtually any public cloud provider. This provides additional flexibility for meeting QoS requirements (e.g., by staging content in public cloud storage "closer" to its consumers and by choosing the most cost-effective combination of public cloud providers to deliver short term and long term content delivery services).
- We describe and discuss sample application domains (education, news and entertainment) which will benefit by using MCCO.
- We present design and prototype implementation of MCCO and conduct experiments to show its effectiveness.

To the best of our knowledge, we are first to clearly articulate the major research challenges involved with designing next generation media management CDNs. The rest of this paper is organised as follows: Section 2 articulates the sample media applications that will benefit from MCCO innovations. Section 3 presents the research vision, challenges, and objectives of designing MCCO. Section 4 describes the design, architecture and implementation of MCCO. Section 5 presents the early performance evaluation study of MCCO. Finaly, we discuss the conclusion and future work in section 6.

2. MediaWise innovations and sample applications

In the following sections, we discuss three interactive applications that will be demonstrated and used in the rest of this paper to help explain the planned innovations of our research regarding MCCO and illustrate its benefits. After discussing these applications, we will introduce the technical research areas of the MCCO Project that constitute its major functions. Finally, we outline the innovations of MCCO project, categorized by technical research areas, using examples from the application areas. Figure 1 shows the innovation areas of the MCCO project as intersections of the project's research and functional areas, the roles users play in producing, managing, and consuming multimedia content, and related application areas.

2.1 Applications and impact areas

In the following paragraphs, we describe the specific application areas we plan to use to demonstrate the impact of the MCCO project. Having specific applications and developing demonstrations that illustrate the benefits of this technology will provide focus in our research and facilitate adoption of MCCO.

2.1.1 Virtual classroom

A virtual classroom is a crucial component of an e-learning system. It requires two main capabilities:

- *synchronous communication and collaboration* for interactive teaching, questioning and answering, allowing class discussions, and supporting team work; and
- *asynchronous web-based knowledge management and dissemination* for making class-related material available to students, performing and submitting homework, and compiling with grading.

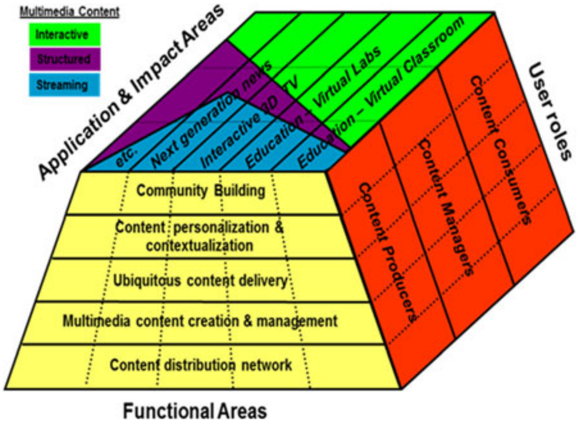

Figure 1 MCCO research areas organized by functional areas, applications, and user roles.

The MCCO project will develop and demonstrate a next generation virtual classroom. Synchronous communication and collaboration will be supported by a high resolution (100 M pixels) immersive videoconference environment that will be integrated with various state of the art tele-presence and shared workspace tools that are appropriate for remote teaching and class work. Provided workspace tools will include a virtual blackboard, screen sharing and co-browsing tools, as well as tools for scientific instrument sharing (e.g., virtual microscopy). Classroom tele-presense will be supported via low cost commercially available devices such as Microsoft's Kinect. It can achieve automatically tracking of the physical activities of all those in a virtual class room. For example, Kinect may be deployed at students' homes to track when two or more students raise their hands to ask or respond to a question. Automatic tracking of such activities can be used to make the virtual classroom more responsive (e.g., notify the remote teacher that a student has a question or requests permission to speak) and to improve classroom fairness (e.g., by automatically queuing and responding to student requests in the order they are manifested by a physical activity).

Next generation asynchronous web-based knowledge management and dissemination will be provided via scientific wikis. Surprisingly, by looking at current practices in many fields of asynchronous web-based, collaborative, distributed knowledge production and dissemination (ranging from community-managed web repositories to complex software development), it is easy to observe that *innovative forms of scientific publications are still lagging behind*, and that the world of scientific publications has been largely oblivious to the advent of the Web and to advances in ICT. Scientific knowledge dissemination is still based on the traditional notion of "paper" publication. In this application area, we want to explore the MCCO approach and how lessons learned from the social Web can be applied to provide a radical paradigm shift in the way scientific knowledge is created, disseminated, and maintained. We argue that novel technologies can enable a transition of "books" and "scientific publications" from its traditional "solid" form, (i.e. a crystallization in space and time of a scientific knowledge artifact) to a more fluid form (hereafter, what we call *Multimedia Scientific Knowledge Objects or MSKOs)*, that can take multiple shapes, evolves continuously in time, and is enriched by multiple sources. Conceptually, this application area is a multimedia counterpart of the traditional notion of scientific publications based on emerging and mature Web 2.0 services. We view the Multimedia Scientific Wiki service as a software application that manages MSKOs embodying a novel form of multimedia publications lifecycle. We consider MSKOs as being intrinsically:

- multi-media, i.e., they support different kinds of content, such as text, images, videos, slides, case studies, experimental datasets, and also include reviews and feedback by the community;
- multi-version, i.e., MSKOs and their constituents evolve over time as people contribute knowledge to them. They exist in multiple, incremental versions;
- multi-author, i.e., they enable the collaboration and contribution of a number of interested and expert authors on a specific MSKO, with different levels of “ownership” and control of the MSKO, and each able to claim credit and responsibility for the contribution; and
- multi-publication, i.e., they support the creation of new MSKOs by composing (and extending) existing ones.

Additional multimedia content in MSKOs includes video captures of demonstrations, experiments, and presentations, and other supplemental material that is essentially the complements of a publication, like experimental data, spatial and temporal information, etc., as well as the opinion and feedback of people in the community. All of these are part of the knowledge associated to an MSKO – and hence contribute to the creation of multimedia knowledge – as it facilitates the understanding of the authors’ original contribution. MSKOs evolve in time as scientific knowledge progresses, and have many actors contributing to their creation and evolution, according to various lifecycle processes. Authors progressively add knowledge (delta-increments) to multimedia content. The community – including other teachers and students, progressively validates the quality of multimedia publications and adds value to it in the form of comments and feedback. Evolution of a publication is not conceived as a novel publication to be created, evaluated, and published anew (with significant loss of time for the community), but rather as the evolution of an existing MSKO, possibly by different authors, with each able to claim credit and responsibility for their contribution.

The paradigm we want to explore in this application is similar to what started to happen twenty years ago in software engineering with the progressive adoption of more agile and iterative development processes, from the spiral model to extreme programming up to “*social*”, *open source development*. Furthermore, the open source and web communities teach us how to perform a “validation” and “credit attribution” of the work (which is key to people’s careers and goals, and their continued participation) that is fair, relatively accurate, allows for high-quality artefacts to be generated, but is also lightweight (relative to peer review today) in terms of reviewing time requirements.

Besides supporting MSKOs and their lifecycle, the Multimedia Scientific Wiki service will involve the development of a prototype of a publication centre, i.e., a set of tools that manage the entire lifecycle of MSKOs and its continuous evolution.

2.1.2 Virtual labs

Today’s Internet-based learning systems can be successfully applied to teach theoretical knowledge presented in the form of structured multimedia content. However, this form of presentation is often insufficient to teach practical skills. Serious games involving Virtual Labs provide users with interactive virtual reality environments, where they can collaboratively work on performing practical tasks and experiments. Serious games bring users exceptional freedom of experimentation. In a game environment, users can interact with virtual objects in a Virtual Lab similarly to the ways they would interact with real objects. For example, they can be confronted with interactive simulations of situations that they may not be able to experience in the real world. Another significant advantage is safety since unskilled learners are able to explore potentially dangerous situations without any risk of harm to themselves or damage to expensive equipment. Serious games may be particularly useful for presenting phenomena, which are:

- potentially dangerous (e.g., chemical or radioactivity experiments);
- macroscopic or microscopic (e.g., astronomical events and molecular movements);
- very fast or very slow (e.g., explosions and continental drift);
- normally hidden from view (e.g., inner workings of machines, human anatomy);
- normally inaccessible (e.g., a nuclear reactor, undersea life); or involve
- abstract concepts (e.g., magnetic fields, molecular forces).

There are many examples of subject domains where serious games have been used for educational purposes, e.g., geography, astronomy, chemistry, and physics. The level of success of applying serious games largely depends on the flexibility of learning environments to match particular user needs. The presented multimedia content can be tailored to the age, learning styles, and performance characteristics of users. Consequently, when users are provided with such highly-customized multimedia content, they become more interested and more engaged in the games and the degree of their satisfaction from the experience increases.

Serious games can serve as the focal point of vibrant eLearning communities. Multimedia content for learning may be created by content producers around the world that join together in communities working on a particular game. An example of a game community may consist of a user playing the role of an instructor, two advanced players, and several beginners. Within such a community two teams may be created. Each team is a sub-community, which must be composed of an advanced player, who is a team leader, and several beginners. The instructor may manage the way the game develops over time, and may judge the competitors and evaluate the achieved results.

In some serious games, player communities may be dynamic. Players may join or leave teams, and move from one team to another, if their profile fits requirements of a destination community. For example, a community may try to outbid another one for the best specialists by offering them better positions. The game provides additional interactive capabilities that help build gaming communities. When a player logs in, the game helps him/her find and re-join his/her team, or search for other gaming friends so he/she can create/join a new team.

There are several scenarios related to possible serious game environments. In one scenario the game may be immersed in a purely virtual environment, where multimedia content is presented in a 3D virtual space. In another scenario, the game may be located in an augmented reality environment, where multimedia content is displayed in real environments of the players. Augmented reality environments combine video streams presenting views of real environments with digital interactive multimedia content. Players using mobile devices could play an interesting variation of a serious game over physical terrain. In this case, the game environment includes real landmarks and real obstacles. A game may rely on searching for some real objects hidden in a real building or discovering some knowledge based on the information derived from the environment. This location-based variation of serious games fosters dynamic formation of location-based teams that are associated by physical proximity.

2.1.3 Ubiquitous news

There are several variations of this application. One variation revolves around a clearinghouse service for video and audio news. Creation of such multimedia news content would be performed by individuals that happen to be present in an event and use personal devices, such as video cameras and cell phones for multimedia capture. This multimedia news content is consumed by users that search the new clearinghouse service for the news they need. Another variation of this news service may allow consumers to request news, even in situations where the event of interest is current but no multimedia news content is being collected or where there is no content because the news request is for a future event. To satisfy such consumer needs, the news service must obtain news content by communicating and negotiating with other users that may be willing to serve as news content creators. To address these diverse requirements, the news clearinghouse service indexes and stores old content, streams new content from current events, and facilitates collaboration between content creators and consumers to obtain new multimedia content. In one version of the news service, multimedia content is not managed in any way (i.e., is raw audio and video that is not combined with other content or adapted for its consumers). In a more traditional multimedia news service, that content will be managed (e.g., enhanced with titles and narrative, and adapted to fit a specific spot in the news program).

Such content management may be performed by more specialized users that participate in another collaborative workflow that assembles a news program for consumers. When multimedia content is uploaded to the clearinghouse, contextual information (such as time and location) is automatically attached to the content. By using the clearinghouse the group is able to compose joint products, such as a “group trip report.” These products also may be posted to a structured multimedia service that permits easy viewing by the family back home for multimedia content performed by more specialized users that participate in another collaborative workflow that assembles a news program for consumers. Another variation of ubiquitous news may involve location-based indexing and search capabilities for multimedia content. Such a service may allow a group of people (e.g., tourists or a school class on a field trip) to make movies and record narratives describing the points of interest

3. Research vision, challenges and objectives of MCCO

This section summarizes the research vision, challenges, and objectives of designing MCCO. Table 1 presents a comparative study of the existing CDN services against the research objectives of MCCO.

3.1 Content creation and management

Users and organizations involved in the creation of multimedia content play the role of content producers. The content they produced can be requested by and delivered to users playing the role of content consumers. Existing models of multimedia content creation are usually one-to-many (i.e., they involve one producer and many consumers) [6,7]. Content producers perform digital capture of multimedia data, and use software tools to perform multimedia post-production (e.g., to add graphics, audio, and titles to raw video clips), and finally produce a

Table 1 A comparative study of the existing CDN services against the research objectives of MCCO

CDN Application Provider	Content creation and management	Ubiquitous content delivery	Content indexing	Content personalization and contexualization	Community building	Quality of service optimization
Limelight Networks [25]	Browser based interface to upload static content; supports multiple media types; do not support dynamic content creation	Dependent on private Limilight networks backbone for content delivery; supports bitrate streaming as configured by content user; only limelight audio/video player supported	Title and Keyword based	No	No	Handled behind the scenes; content providers have no control over QoS; best effort QoS at network layer
Oyala [26]	Browser-based interface to upload static content. Dynamic content creation is not supported.	Yes, to multiple devices using multiple formats	Title and Keyword based	Personalization is partially supported based on the type of devices to be used for media streaming	No	Yes, based on different CDN providers such as Akamai,
NetFlix [27]	N/A	Yes, to multiple devices using multiple formats	Title and Keyword based	Personalization is partially supported based on the type of devices to be used for media streaming	No	Yes, based on different CDN providers such as Akamai and now their own.
Akamai (Sola) and Ultraviolet [28]	Content can be managed partially, only for purchased content such as movies.	Yes, to multiple devices using multiple formats.	Title and Keyword based	Personalization is partially supported based on the type of devices to be used for media streaming	No	Yes, via Akamai. The content providers have no control over QoS provisioning.
MetaCDN [29]	Content cannot be created but can only be managed using pre-defined emplates	Yes, to multiple devices using multiple formats.	No	No	No	Yes but content providers have no control over QoS provisioning.
Rackspace [30]	Mainly content storage and provisioning	Yes, based on Akamai CDN	No	No	No	Yes, using Akamai. However, content providers have no control over QoS provisioning.

media product (e.g., a TV program, a podcast, or a set of structured multimedia web pages that link text, pictures, video and music content). The MCCO research will focus on many-to-many models for creating novel forms of expressive and interactive content. In particular, as a starting point, MCCO will utilize existing Web 2.0 technologies for creating structured multimedia content (i.e., tools allowing users to create web pages linking video, pictures, audio, text, and graphics), but broaden existing Web 2.0 strengths with new technologies for creating novel forms of interactive multimedia content (e.g., as discussed earlier, our research will consider interactive content involving virtual and augmented reality). In particular, novel research in the MCCO will focus on the following three content creation and management areas:

3.1.1 Novel types of multimedia content

Existing Web 2.0 technologies currently support the authoring of *structured multimedia content* (e.g., web pages linking images, sounds, videos, and animations). The MCCO will extend and broaden existing Web 2.0 strengths with a new environment aimed at supporting the creation and consumption of interactive multimedia content (e.g., interactive audio and video), as well as other novel forms of multimedia content (e.g., *virtual* and *augmented reality*) that are currently not supported by existing Web 2.0 technologies and tools.

3.1.2 Advanced content management services

Multimedia content management includes the creation, adaptation, and composition of *content management* (CM) services. Basic CM services package multimedia content together with functionality that manipulates it. Examples of basic content services include functions for storing, indexing, and searching multimedia content [8]. CM services may be adapted for a specific purpose, or personalized for individual users or communities. Creation, adaptation, and composition of CM services are accomplished by (possibly collaborating) users that play the role of *content managers*. Complex CM services may be created by adapting or composing simpler CM services. For example, a basic content management service may index movies, TV programs and news for future search. A complex content management service may provide an interactive TV guide and its functionality may use the basic indexing service.

The MCCO research will develop and demonstrate complex content management services for learning, news and entertainment applications. For example, we will build a software tool that allows users to collaboratively create non-linear multimedia presentations. For example, a teacher may create a new project. Teams of students can play the role of content creators and upload new content to the common project in the form of photos as well as audio and video recordings from mobile and desktop devices. Other students can become content managers and create a joint presentations by defining the relations (e.g., temporal, geographical, or topical) between these multimedia objects. Moreover we want to enhance Web 2.0 with novel technologies (for storing, indexing, searching, adapting, composing, and consuming streamed and interactive multimedia content) as well as innovative tools for collaborative editing, trends/hot topics/historical patterns identification and metadata feature management. The interactive video production will be further optimized by removing the redundant information that may introduced by multiple users. The redundant multimedia parts that are same in content are identified by incorporating the advanced near duplicate multimedia content detection techniques [9]. MCCO research in collaborative multimedia content management will provide next generation CM services for these functions; as well develop technologies for automating the adaptation and composition of CM services. Our research in MCCO will also develop a novel environment for automating the adaptation and composition of CM services.

3.1.3 Collaborative content management workflows

The research in MCCO will develop novel technology for supporting collaborative workflows that manage the lifecycle (i.e., the creation, management, and consumption) of complex multimedia content. In particular, we will research a wide range of user collaboration styles, develop novel technology for collaborative workflows managing the orchestration of services and human activities for multimedia content creation, management, consumption, and determine how to achieve greater collaboration scale, user participation, and efficiency. Collaborative workflows for multimedia content creation, management, and consumption may range from structured to unstructured. As an example of structured workflow that manages the lifecycle of multimedia content, consider the production of a web-based news program. The production of such a news program follows an established workflow process that involves the following steps: (1) capture and post-process new video clips of a newsworthy event, (2) search video archives for related video clips, (3) create/edit audio annotations for the new and possibly old video clips, (4) save the new video clips/audio annotations and enable future content-based search, (5) capture the video or audio presentation of a news script by a news anchor, (6) post all products on a website, and (7) enable both linear and non-linear viewing of the news products. At the other end of the spectrum, ad hoc multimedia content creation involves no explicit control flow or coordination between multimedia content producers or mangers. Consider an existing Web 2.0 service that allows users to post video clips of events they consider to be newsworthy. This involves unconstrained uploading and sharing of multimedia content. MCCO will develop new technologies that accommodate these and any style of work or user preferences ranging from ad hoc to highly-structured activities for collaborative content creation and management. We will focus particularly on identifying, modelling, and automating collaboration patterns that increase the scale, user participation, efficiency, and automation of collaborative media creation and management activities.

3.2 Ubiquitous content delivery

Multimedia content can be subscribed to or requested by *content consumers*. The goal of multimedia content delivery is to provide a seamless multimedia experience to users. In its simplest form, this involves rendering the multimedia content to deal with the constraints of the network that carries the multimedia content and the characteristics of the specific device a user utilizes to receive the content and interact with it. Advanced multimedia delivery involves the development of *content delivery* services (CD). CD services will interact with the network and appropriately adjust its QoS as needed to deliver specific multimedia content to a specific user. In particular, we will focus on research that will lead to the development of CD services. When given a specific content and a specific content consumer, these CD services will adjust QoS characteristics based on content requirements for maintaining its integrity, the device the user is using, his/her location, and the service contract. We will research content transformation to meet target device and network constraints. Another related research objective of MCCO is to provide ubiquitous content delivery between producers/managers that may be distributed across many states or nations. The research in MCCO will develop technology for ubiquitous content delivery accommodating desktop users as well as mobile users who may meet in a coffee shop.

3.3 Flexible content storage, compression, and indexing

Cloud storage resources allow content producers to store content on virtualized disks and access them anytime from any point on the Internet. These storage resources are different from the local storage (for example, the local hard drive) in each CPU resource (e.g., Amazon EC2 instance types), which is temporary or non-persistent and cannot be directly accessed by other instances of CPU resources. Multiple storage resource types are available for building content Orchestrator. Naturally, the choice of a particular storage resource type stems from the format (e.g., structured vs. unstructured) of the content. For instance, Azure Blob [10] and Amazon S3 [11] storage resources can hold video, audio, photos, archived email messages, or anything else, and allow applications to store and access content in a very flexible way. In contrast, NoSQL (Not Only SQL) storage resources have recently emerged to complement traditional database systems [12]. They do not support ACID transaction principles, rather offer weaker consistency properties, such as eventual consistency. Amazon SimpleDB [13], Microsoft Azure Table Storage, Google App Engine Datastore [14], MongoDB [15], and Cassandra are some of the popular offerings in this category. For example, in Amazon CloudFront [16], CDN contents are organized into distributions.

A distribution specifies the location of the original version of contents. The distribution can be hosted on cloud storage resources such as Amazon S3 or Amazon EC2 CPU resources. With an increase in the scale and the size of content distribution, efficient indexing and storage become a critical issue. The challenge is further aggravated in case of live and interactive content, where size of distribution (hence the indexing complexity) in not known in advance. Though cloud environments are decentralized by nature, existing application architecture tends to be designed based on centralized network models. To support efficient content production and consumption on scale of TeraBytes or PetaBytes, it is mandatory to design decentralized content indexing algorithm to enable access and search over large-scale database. It is worth noting that none of the existing cloud storage resources exposes content indexing APIs. It is up to the CDN application designer to come-up with efficient indexing structure that can scale to large content sizes. To help end-users find and retrieve relevant content effectively and to facilitate new and better ways of media delivery using cloud resources, advanced distributed algorithms need to be developed for indexing, browsing, filtering, searching and updating the vast amount of information available in multimedia content.

3.4 Content personalization and contextualization

We are just beginning to realize the power of location-aware services, especially for mobile devices, that aid users in their interaction with their immediate physical environment. Location-aware services allow users to become aware of physically proximate resources that they might not otherwise know about, provide convenience in finding and interacting with those resources, and enable interpersonal networking that takes physical location as well as ad-hoc community-building in account. Imagine young people planning their weekend meetings within a connected infrastructure which is location-aware: the group of young people meet ad-hoc, are aware of each other's locations and create media content using video cameras or chatting (through typing or directly through voice messages).

MCCO will also support additional contextual information, including users' resources and capabilities for networking and computing, their work and leisure activities, their preferences, and the communities to which they belong. For the user, the benefits of content awareness, convenience, and community-building will be enabled by each such context. For the content producers, context can be viewed as a mechanism for mass customization that better meshes the needs and interests of each user with the multimedia capabilities at his/her disposal. MCCO will support models of users profiles which will store contextual information, including users resources and capabilities for networking and computing, their work and leisure activities, their preferences, and the communities to which they belong. For a user, the benefits of content awareness, convenience, community searching and community building will be enabled by each such a user profile. For the content producers, this contextual information can be viewed as a mechanism for mass customization that better meshes the needs and interests of each user with the multimedia capabilities at his/her disposal. In addition, models of users' profiles will include the specification of competences, skills of users, and evaluation of users' former activities, *potentially in other communities*. Additionally the storage of profiles of content creators and managers is especially important to find users which fulfil the requirements of a given community.

In this part of the work, research objectives within the MCCO include the following: (1) *Semi-automatic contextualization* – the automatic construction and maintenance of context of MediaWise content users through mechanisms including user-modelling and data mining, but augmented with more interactive mechanisms such as knowledge elicitation. (2) *Personalization* – capturing the needs of users related to their evolving context so as to maximize the benefit to each user. (3) *Adaptation* – customizing content and services based on context to better meet the needs of users who ultimately pay for them. (4) C*ommunity building* – which extends personalization to groups of users with various commonalities. (5) Finally, *cross-community aspects of users' profile* – modelling aspects of users' profiles which are relevant for various communities to which they belong, such as skills and competences, as well as aspects of users' profiles related through their activities in other communities, such as level of involvement.

3.5 Community building

The concept of community is at the core of the Web 2.0 [17]. In most applications of the Web 2.0, the structures of communities in terms of members' competences are pre-defined, e.g. on flickr.com, users may publish their pictures to predefined groups: family members, friend or others. In current Web 2.0 applications, a user cannot tailor a community to his/her own needs, for instance defining an editor, a graphic designer and 5 journalists for a newspaper edition community. Nor can users easily find communities to which they may contribute, depending on their competencies, skills and centres of interest. Currently, users enter and leave communities in a chaotic way, depending mostly on recommendations of other users or links they may find while navigating.

Our research in MCCO intends to provide users with tools for community-building, based on a new model of communities and users' profiles. The proposed model of communities will integrate the concepts of competences and skills to specify the requirements of a community to be built. The model of users' profiles will support, on the one hand, the specification of competences and skills of users and, on the other hand, evaluation of users' former activities, potentially in other communities. The latter information will be obtained via: (1) real time monitoring of events (e.g., initiation and completion of user activities) that create, manage, and consume multimedia content, and (2) tracing such events involving one or more users to user communities. Based on the models of communities and users' profiles, new algorithms matching community specifications with users' profiles will be proposed for 1) identification of communities of interest, and 2) creation of new communities. Finally, as the requirements of a given community may evolve through time, the proposed model of communities will support adaptation of community specifications during the lifetime of a given community. Similarly, the profile of a given user will evolve in time along with the evaluation of his/her former activities. Therefore, tools supporting community management will take into account the high dynamics of communities and users' profiles.

3.6 Quality of service optimizer

It has been shown that one of the challenges in orchestrating cloud resources for managing CDN application is uncertainty. Resource uncertainty arises from a number of issues including user location, content type, malicious activities and heterogeneity. In some cases, media content delivery application may face with failure of resources or sometimes it may suffer from lack of sufficient resources. In this project, we will investigate the following issues for ensuring end-to-end QoS fulfilment in content production, consumption, and delivery.

3.6.1 Optimizing cloud resource selection

There are two layers in cloud computing as shown in Figure 2: a) service layer (e.g., Google App Engine, 3Tera Applogic, BitNami), where an engineer builds applications using APIs; and b) infrastructure layer (e.g., GoGrid, Amazon EC2), where an engineer runs applications inside CPU resources, using APIs provided by their chosen guest operating systems. Optimal CDN application performance demands bespoke resource configuration, yet no detailed, cost, performance or feature comparison of cloud providers exists. This complicates the choice of cloud providers.

The diversity of offering at this layer leads to a practical question: how well does a cloud provider perform compared to the other providers? For example, how does a CDN application engineer compare the cost/performance features of CPU, storage, and network resources offered by Amazon EC2, Microsoft Azure, GoGrid, FelxiScale, TerreMark, and RackSpace. For instance, a low-end CPU resource of Microsoft Azure is 30% more expensive than the comparable Amazon EC2 CPU resource, but it can process CDN application workload twice as quickly. Similarly, a CDN application engineer may choose one provider for storage intensive applications and another for computation intensive CDN applications. Hence, there is need to develop novel decision making framework that can analyse existing cloud providers to help CDN service engineers in making optimal selection decisions.

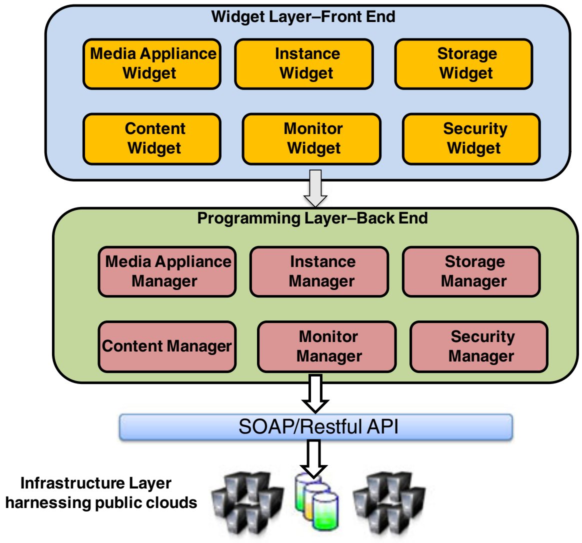

Figure 2 Software Architecture of MCCO.

3.6.2 Adapting to Dynamic CDN workload

Media delivery CDN applications must accommodate highly transient, unpredictable users behaviour (arrival patterns, service time distributions, I/O system behaviours, user profile, network usage, etc.) and activities (streaming, searching, editing, and downloading). Yet many cloud providers contract service-level agreements which stipulate specific QoS targets, such as how fast a web page is served. It is therefore important to all parties that highly variable spikes in demand, caused by large number of simultaneous requests for a shared CDN service, do not degrade QoS [18,19].

It is critical that MCCO is able to predict the demands and behaviours of hosted media applications, so that it can manage resources. Concrete prediction or forecasting models must be built before the demands and behaviours of a CDN application can be predicted accurately. The hard challenge is to accurately identify and continuously learn the most important behaviours and accurately compute statistical prediction functions based on the observed demands and behaviours such as request arrival pattern, service time distributions, I/O system behaviours, user profile, and network usage. The challenge is further aggravated by statistical correlation (such as stationary, short- and long-range dependence, and pseudo-periodicity) between different behaviours and activities of CDN application.

3.6.3 Adapting to uncertain cloud resource environment

The availability, load, and throughput of hardware resources (CPU, storage, and network) can vary in unpredictable ways Thus, ensuring that CDN applications achieve QoS targets can be difficult. Worse still, hardware resource status can be changed intentionally through malicious external interference. The recent high-profile crash of Amazon EC2 cloud which took down many applications is a salient example of unpredictability in cloud environments. Theoretically, the elasticity provided by cloud computing can accommodate unexpected changes in capacity, failure, adding hardware resources when need, and reducing them during periods of low demand, but the decision to adjust capacity must be made frequently, automatically, and accurately to be cost effective.

3.7 Other research objectives

The objective of our research in MCCO is to provide for *agility* in the creation, management, and consumption of content by small groups of users. For large groups of users and massive content, we will focus on providing maximum *efficiency*. Specifically, it should be noted that our research intends to specifically address scalability, reliability and flexibility issues. While scale may be defined in a variety of ways (e.g., by the number of end-to-end users involved in various content production, management and consumption roles, the number or sources and size of multimedia content, the group sizes of collaborating content producers and managers, the size of consumer communities, etc.), scale issues impact all functional areas and user roles as illustrated in Figure 1. Reliability is achieved through the fault-tolerance feature of a system in the face of sudden load spikes such as flash crowds. Flexibility of the system can be realized through the ease of use, integration to existing system and on-demand deployment. We intend to test selected scenarios within a mobile city environment so that local, as well as mobile, scenarios can actually be explored.

4. MediaWise cloud content orchestrator: design and architecture

As mentioned earlier, we developed a cloud-based generic and scalable software framework called MCCO [20] for supporting the end-to-end lifecycle operations required for managing content via clouds and the Internet. The MCCO exploit public clouds for offloading computing, storage, network, and content distribution functionalities in a cost effective manner. Cloud computing [3,4] assembles large networks of virtualized services: hardware resources (CPU, storage, and network) and software resources or appliances (e.g., databases, message queuing systems, monitoring systems, load-balancers). Cloud providers including Amazon Web Services (AWS), Microsoft Azure, Salesforce.com, Google App Engine, and others give users the option to deploy their application over a network of infinite resource pool with practically no capital investment and with modest operating cost proportional to the actual use.

MCCO offers enhanced flexibility and elasticity as it inherits pay-as-you-go model from public cloud resources. MCCO content orchestration operations include: (i) production: create and edit; (ii) storage: uploading and scaling of storage space; (iii) keyword-based content tagging and searching and (iv) distribution: streaming and downloading. At Cloud service level, MCCO capabilities span across a range of operations such as selection, assembly, deployment of cloud resources to monitoring their run-time QoS statistics (e.g., latency, utilization, and throughput). MCCO orchestrate public cloud resources via open-source RESTFul APIs. It supports deployment, configuration and monitoring of content and cloud resources using Web-based widgets. These widgets hide the underlying complexity related to cloud resources and provide *an easy do-it-yourself* interface for content management. The high level architecture of MCCO is shown in Figure 2. The Widget layer presents a unified front end for end users to perform aforementioned content orchestration operations. It hides the complexities related to all these operations by using a plethora of in-house and open source APIs. These APIs are implemented at the Programming layer and manage operations for Infrastructure layer. For example, starting and stopping a virtual machine.

4.1 Infrastructure layer

This layer provides cloud-based hardware resources such as CPU, storage, routers and switches that hosts the media appliances such as streaming server, indexing server, and editing server. Hardware resources expose certain configuration that can be allocated to media appliances. For example, a streaming server appliance available from Wowza [21] can be assigned following Amazon EC2 CPU configurations: 7.5 GB memory, 4 EC2 Compute Unit, 850 GB instance storage, 64-bit addressing and moderate I/O. More details on optimal hardware and appliance selection can be found in the following paper [22]. In general, cloud providers manage resources at infrastructure layer through hardware virtualization technologies [23] such as Xen, Citrix, KVM (open source), VMWare and Microsoft Hyer-V. Virtualization allows providers to get more out of hardware resources by allowing multiple instances of virtual resources to run at the same time. Each virtual resource believes it has its own hardware. Virtualization isolates the resources from each other, thereby making fault tolerant and isolated security behaviour possible.

4.2 Programming layer

This layer implements the logic for interfaces exposed by widget layer. For example, the Media Appliance Manager implements Cloud resource API that allows Appliance Widget to list the set of media appliances (e.g., streaming, indexing and editing servers) associated with owner's account. Programming Layer is also designed to allow engineers to plug-in different Cloud service APIs. Notably, each of the managers at this layer has to perform certain orchestration operation on infrastructure layer cloud resources, such as provisioning of a streaming server appliance over an Amazon EC2 or indexing of contents over Amazon S3. Currently, our implementation works with Amazon Web Service (AWS) and is being extended to support other Cloud providers.

4.3 Widget layer

Widget Layer encapsulates user interface components in the form of six principle widgets including Media Appliance, Instance, Storage, Monitor, Content, and Security. Next, we provide the brief details about each widget.

- Media Appliance Widget: It lists the set of media appliances associated with content owner's account. In general, an appliance [24] is pre-configured, self-contained, virtualization-enabled, and pre-built software resource unit (e.g., streaming, indexing and editing servers) that can be integrated with other compatible appliances for architecting complex applications such as video-on-demand CDN.
- Instance Widget: Content owners are required to describe the media appliances' deployment configurations that will affect and drive its instance's placement and performance. Configuration parameters include number of instances, their types, security setting, and monitoring preference. In context of Amazon's EC2, different instance types provide different minimum performance guarantees depending on their memory, storage, and processor configurations. Additionally, content owners or CDN administrators can also consider non-functional attributes related to deployments such as hosting cost, latency, throughput, scalability, and availability. The discussion of algorithms that consider optimization of these attributes is beyond the scope of this paper.
- Storage Widget: It allows content owners to upload content and media appliances to storage service (e.g., Amazon S3). Cloud storage resources provide a highly durable and available storage for a variety of content types, including web applications and multimedia files.
- Content Widget: It enables the functionality for tagging, indexing, and personalizing content with metadata. It exposes a drag and drop interface for mapping of an audio/video content from cloud storage to a media appliance. Content can be tagged with one or more keywords.
- Monitor Widget: It supports monitoring the status of media appliance instances, network and storage services. Monitored data such as media appliance throughput, utilization, disk I/O are made available in form of two-dimensional charts
- Security Widget: It manages all the authentication and authorization credentials related to orchestrating content (e.g. content access secret key) and cloud resources (access key and secret key). MCCO includes security credentials provided by Amazon EC2 which is read directly from a file stored in a web folder. In future, we will implement more security mechanisms to better manage the security credentials from different cloud providers.

4.4 Implementation

Although the design of MCCO is generic and extensible, our current implementation is specific to supporting orchestration of do-it-yourself CDN using public cloud resources. However, we believe the system is mature enough for validating the overall idea and vision. The system is entirely written in Java and makes use of number of open-source libraries:

- GWT and SmartGWT, to build basic user interfaces.
- Amazon's EC2 API, for implementing appliance life cycle operations (start, stop, suspend, and delete).
- Amazon's S3 API, for implementing data copy operation across S3 and media appliances.
- Amazon's CloudWatch API, for collecting performance data from CPU resource instances.
- JQuery, for interactive charting function.
- Apache Commons File Upload, for implementing upload function.
- JSch, for implementing secure communication channel to access the appliance instance environment.

MCCO supports adapter and factory design fattern to provide clear mechanisms for designing flexible software architecture that are highly adaptable and extendable.

5. Early experiments and preliminary results

Although we are actively working towards the implementation of MCCO, it is still a work-in-progress. Hence, in this section, we present our experiments and evaluation that we undertook for studying the feasibility of the proposed research vision.

5.1 Testbed setup

5.1.1 Media appliance configuration

The MediaWise testbed setup is shown in Figure 3. To stream audio and video content, we leverage the streaming media appliance provided by Wowza Media Systems [21]. In the current release, Wowza media appliance support following streaming protocols: Flash (RTMP and HTTP), Microsoft Silverlight, QuickTime/3GPP (RTSP/RTP). The Wowza appliance is deployed on Ubuntu operating system (ami-d7a273be).

For Wowza appliance to work properly, it has to be a properly integrated with content storage resource. As mentioned previously, cloud storage resources do not expose any content indexing API. Hence, we created our own appliance that supports keyword-based content indexing. This appliance had the following software configuration: a RESTFul indexing web service hosted on 32Bit Java Virtual Machine version 1.6 and Glassfish application server on a Windows 7. At the deployment phase, we created multiple instances of the streaming appliance and one instance of indexing appliance.

The indexing appliance stores the content URL in the MySQL in the format: http://[IP address]/vods3/_definst_/mp4:amazons3/[S3 Bucket name]/[File name]. At run time, the indexing appliance fetches the URL of end-user requested contents from storage resources to Wowza appliance.

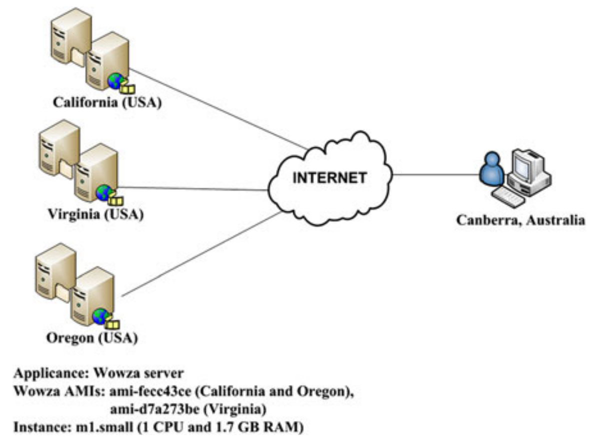

Figure 3 Experimental setup for MCCO project. Media server appliances are hosted at three locations across USA. A user in Australia then deploys appliances' and uploads and downloads the media content through MCCO.

5.1.2 Hardware resource (Infrastructure) configuration

We tested our implementation of MCCO in experiment where it was hosted on a 32Bit Java Virtual Machine version 1.6 and Glassfish application server on a Windows 7 with an Intel Core 2 Duo 2.5 Ghz CPU and 2 GB of RAM. We configured the MCCO with security credentials required for orchestrating Amazon EC2 and Amazon S3 resources hosted in US Oregon, US North California, and US Virginia availability zones. We hosted both media appliances using an Amazon EC2 large instance across these availability zones. By default, a large instance has the following hardware configuration: 7.5 GB of main memory, 4 EC2 Compute Unit (i.e. 2 virtual cores with 2 EC2 Compute Unit), 850 GB of local instance storage, and a 64-bit platform.

5.2 Results and discussions

For results validation, we considered two metrics, , average upload throughput and average upload delay. The choice of these metric were based on the fact that MCCO enables content creation and its timely sharing. This requires timely content availability to end users. In cloud systems, content can be geographically located around the globe. Usually, content is placed at the location closest to the user for QoS maximization. In this section, we estimate the aforementioned metrics by uploading the content to three different US-based Amazon S3 datacenters in Oregon, California and Virginia. These experiments were conducted from Canberra, Australia.

Figure 4 shows the graph for Wowza instance startup time based on three different availability zones across USA. It can be observed that the instance startup time can vary from around 1 minute to 3 minutes. We consider these times to be sufficient if there comes a need

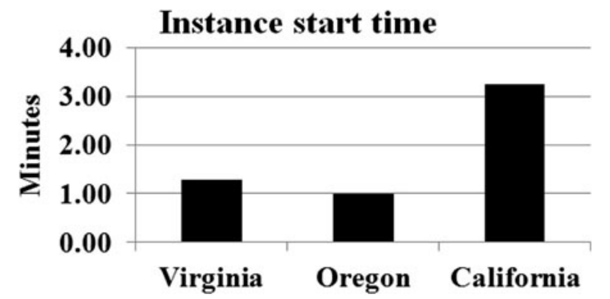

Figure 4 Amazon EC2 instance start time for datacenters in US.

to deploy more media appliances to manage varying access load.

Figures 5 and 6 shows the graph for average upload delay and average upload throughput estimated by MCCO by uploading video file content various file sizes to three different US-based Amazon S3 datacenters. We considered video file sizes of 51.70 MB, 103 MB, 201 MB and 500 MB. From these results, we concluded that average upload delay and throughout varies from one avaliabiity zone to another. In particular, we conclude that Oregon datacenter provided best network delay and throughput followed by Virginia and California data centers.

Figure 7 shows the content access delay which is the time difference between a button pressed by a user in application to the time video is rendered on the users screen. It can be concluded that content access delay, if high, a user has to wait for several seconds for video to start. This might not be optimal for users' viewing experience. Thus, the videos need to be hosted at and streamed from a location closest to the end user. From these experiments we concluded that MCCO can be used to deploy media appliances and host content across different geographical locations. It can be noted that content originating from Oregon datacenter started much earlier than California and Virginia.

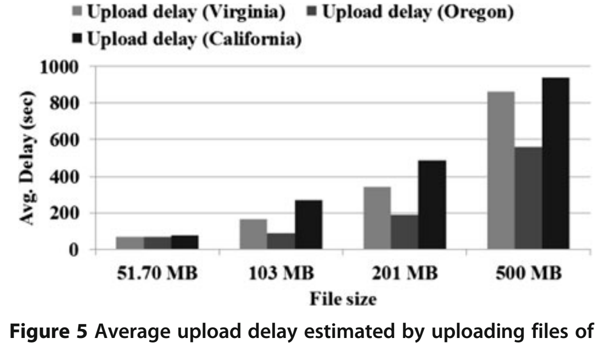

Figure 5 Average upload delay estimated by uploading files of various sizes to three Amazon S3 datacenters in US.

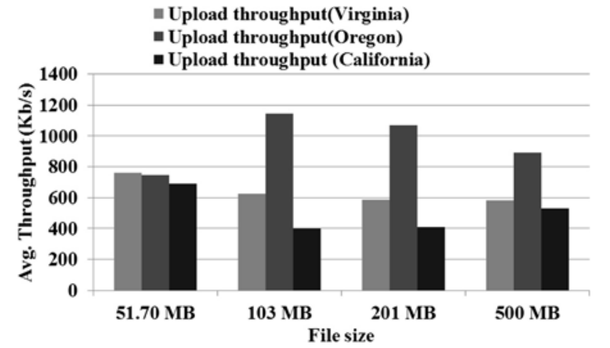

Figure 6 Average upload throughput estimated by uploading files of various sizes to three Amazon S3 datacenters in US.

6. Conclusions and future work

The growing ubiquity of the Internet and cloud computing is having significant impact on the media-related industries, which are using them as a medium to enable creation, search, management, and consumption of their contents online. We clearly articulate the architecture of MediaWise Cloud, its service components and associated research challenges, wherever applicable. Arguably, this paper is the first attempt at capturing the research and development challenges involved with engineering next-generation, *do-it-yourself CDN platform* using public cloud resources.

Our future work includes monitoring and learning of QoS-related performance of virtually all available public cloud services, and using this information for on demand prediction of expected QoS for media delivery requests. Other innovations will include providing seamless and personalized user experience,and allowing users to collaborate in creating, managing, and consuming multimedia content virtually from anywhere and whichever means available to them. To achieve seamless and personalized user experience, we will develop sophisticated context-aware, location-dependent media-related services. We will develop technologies that will present each user with media choices appropriate for his/her

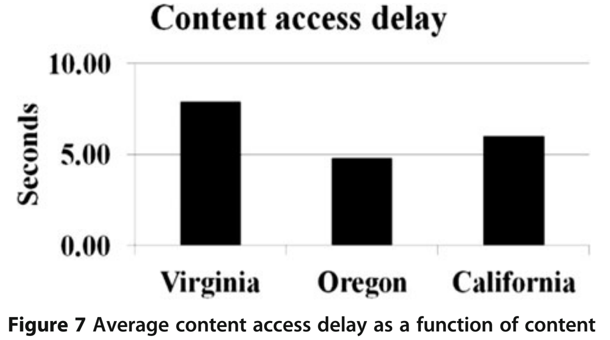

Figure 7 Average content access delay as a function of content location.

situation (i.e., location, task at hand, resources, skills, past activities, etc.). For seamless multimedia content experience, we will provide novel solutions for rendering such content to address network and user device constraints. To improve scale and efficiency of collaborative media creation and management activities, we will enhance user coordination by capturing and automating efficient collaboration patterns, activities, and processes. Further, MCCO will be used to develop and demonstrate three innovative applications in the areas of education, news and entertainment.

Interaction with cloud resources can be done through Application Programming Interface (API) in specific programming languages such as Java, C#, Python and Ruby on Rails. Unfortunately, most of the existing APIs supported by cloud providers (e.g., Amazon, Microsoft Azure, GoGrid and Ninefold) are not compatible with each other. These providers tend to have their own proprietary APIs which are not explicitly designed for cross-cloud interoperability. To tackle such heterogeneities, there is a design requirement to enforce standardization across service implementations. In future, we will take advantage of recent developments in the context of standardized cloud APIs including Simple Cloud, Delta Cloud, JCloud, and Dasein Cloud, respectively. These APIs simplify the cloud programming task by implementing single API that abstracts multiple heterogeneous APIs exposed by cloud providers.

Endnotes

[a]In rest of this paper, we address to the terms MCCO and MCCO project interchangeably.

Received: 30 November 2012 Accepted: 4 December 2012
Published: 30 January 2013

References

1. Buyya R, Pathan M, Vakali A (2008) “Content Delivery Networks”, Lecture Notes in Electrical Engineering (LNEE), vol 9. Springer, Germany. ISBN 978-3-540-77886-8
2. BigData & CDN, http://www.slideshare.net/pavlobaron/bigdata-cdn-oop2011-pavlo-baron, [ONLINE], Access date: 03/07/2012
3. Wang L, Ranjan R, Chen J, Benatallah B (eds) “Cloud Computing: Methodology, Systems, and Applications,”. CRC Press, Taylor and Francis Group, p 844, Publication Date: October 03, 2011. ISBN 9781439856413
4. Armbrust M et al (2010) “A view of Cloud Computing,”. Communications of the ACM Magazine 53(4):50–58. doi:10.1145/1721654.1721672, ACM Press
5. Georgakopoulos D, Ranjan R, Mitra K, Zhou X (2012) “MediaWise – Designing a Smart Media Cloud”, in Proceedings of the International Conference on Advances in Cloud Computing (ACC 2012), Banglore, India [Keynote paper]
6. Bianchi M (2004) “Automatic Video Production of Lectures Using an Intelligent and Aware Environment”. Proceedings of the third international conference on Mobile and ubiquitous multimedia: 117–123
7. Laibowitz M, Gong N-W, Paradiso JA (2010) “Multimedia Content Creation Using Societal-Scale Ubiquitous Camera Networks and Human-Centric Wearable Sensing”. in ACM Multimedia: 571–580
8. Prabhakaran B (1997) “Multimedia Database Management Systems”
9. Zhou X, Chen L (2010) “Monitoring near duplicates over video streams”. in ACM Multimedia: 521–530
10. AzureBlob, http://cloud-computing.learningtree.com/tag/azure-storage-blobs/ [ONLINE], Access date: 8/06/2012
11. AmazonS3, http://aws.amazon.com/s3/, [ONLINE], Access date: 8/06/2012
12. NoSQL, http://nosql-database.org/, [ONLINE], Access date: 8/06/2012
13. AmazonSimpleDB, http://aws.amazon.com/simpledb/, [ONLINE], Access date: 8/06/2012
14. GoogleAppEngineDatastore, https://developers.google.com/appengine/docs/python/datastore/, [ONLINE], Access date: 8/06/2012
15. MongoDB, http://www.mongodb.org/, [ONLINE], Access date: 8/06/2012
16. AmazonCloudfront, http://aws.amazon.com/cloudfront/, [ONLINE], Access date: 8/06/2012
17. Web2.0, http://en.wikipedia.org/wiki/Web_2.0, [ONLINE], Access date: 8/06/2012
18. Wada H, Suzuki J, Oba K (2009) “Queuing Theoretic and Evolutionary Deployment Optimization with Probabilistic SLAs for Service Oriented Clouds”. In: Proc. of IEEE ICWS International Workshop on Cloud Services. IEEE Computer Society, Los Angeles, CA
19. Calheiros RN, Ranjan R, Buyya R (2011) “Virtual Machine Provisioning Based on Analytical Performance and QoS in Cloud Computing Environments”, 40th International Conference on Parallel Processing (ICPP 2011). IEEE Computer Society, Taipei, Taiwan
20. Ranjan R, Mitra K, Saha S, Georgakopoulos D, Zaslavsky A (2012) “Do-it-Yourself Content Delivery Network Orchestrator”. Web Information System Engineering-WISE 2012, 789-791
21. Media Server & Video Streaming Server, Wowza Media Systems, http://www.wowza.com, [ONLINE], Access date: 5/07/2012
22. Menzel M, Ranjan R (2012) “CloudGenius: Decision Support for Web Server Cloud Migration”, In Proceedings of the 21st international conference on World Wide Web (WWW '12). ACM, New York, NY, USA, pp 979–988
23. Barham P et al (2003) Xen and the Art of Virtualization. Proceedings of the 19th ACM Symposi-um on Operating Systems Principles. ACM Press, New York
24. Ranjan R, Benatallah B “Programming Cloud Resource Orchestration Framework: Operations and Research Challenges”., p 19, Technical report, arXiv:1204.2204, Published Online on 10 April 2012
25. Limelight Networks, http://www.limelight.com/ [ONLINE], Access date: 07/06/2012
26. Ooyala, http://www.ooyala.com/, [ONLINE], Access date: 07/06/2012
27. Netflix, http://www.netflix.com, [ONLINE], Access date: 07/06/2012
28. Akamai, http://www.akamai.com/html/solutions/sola-solutions.html, [ONLINE], Access date: 07/06/2012
29. MetaCDN, http://www.metacdn.com/, [ONLINE], Access date: 07/06/2012
30. Rackspace, http://www.rackspace.com/cloud/, [ONLINE], Access date: 07/06/2012

doi:10.1186/1869-0238-4-2
Cite this article as: Ranjan *et al.*: MediaWise cloud content orchestrator. *Journal of Internet Services and Applications* 2013 **4**:2.