A Cloud-based Collaborative Video Story Authoring and Sharing Platform

Chen Wang[1], Rajiv Ranjan[1], Xiangmin Zhou[1], Karan Mitra[1], Suhit Saha[1], Meng Meng[1], Dimitrios Georgakopoulos[1], Lizhe Wang[2] and Peter Thew[1]

[1] Information Engineering Laboratory, CSIRO ICT Centre. Email: firstname.lastname@csiro.au
Corresponding Author: Rajiv Ranjan (rajiv.ranjan@csiro.au)

[2] Center for Earth Observation & Digital Earth, Chinese Academy of Sciences
Email: LZWang@ceode.ac.cn

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. Rich web pages, software downloads, interactive communications, and ever-expanding universe of digital media require a new approach to multimedia (e.g., audio, video, images, etc.) content sharing and management. Size and volume of multimedia content is growing exponentially. Though there are many online videohosting services (YouTube, flickr, dailymotion, etc.) for storing and viewing multimedia content, there is clear lack of software platforms that can help end-users in content authoring and sharing activities in a social setting. This paper introduces the principle concepts of multimedia content sharing in a social setting and presents a novel platform that supports cloud-based collaborative video story authoring and telling. The platform leverages social networks for video search and recommendation for the purpose of story telling. The Social Content Authoring and Sharing Platform (SCASP) leverages MediaWise Cloud Content Orchestrator (MCCO) that supports do-it-yourself creation, search, management, and consumption of multimedia content. MCCO supports dynamic content delivery using CPU and storage services available in the public cloud. The platform exploits flexible content distribution capabilities of MCCO for a group of possibly geographically distributed end-users who collaborate on a specific task such as video story telling.

Index Terms - Content Delivery Network, Cloud Computing, Media Management, Media Delivery, Media Consumption, Personalization, Quality of Service

I. Introduction

There is a large number of videos produced and stored in public (e.g., YouTube) and private (e.g., Smartphones) data repositories on a daily basis. Story-telling is a common use-case for using these videos where an end-user composes a set of videos together to tell a story, either for learning purposes or sharing experiences. For example, a news editor may search and author story based on video collected from multiple private and public repositories; an instructor may produce teaching materials using a set of publicly available video clips. Existing search engines have limitation on identifying useful video contents for users with different needs [7]. For a user who attempts to author a story using a variety of videos, there are number of a few challenges with existing technologies.

A search engine often returns a long list of videos that are relevant to the keywords the end-user enters into the search engine. How the videos in the list are suitable for the story line requires the end-user's further investigation and the amount of work involved often overwhelms the end-user considering the number of videos returned by a search engine. It is often the case that the highly ranked videos are not the most appropriate ones for a story line under authoring. In addition, videos are not organized in a structured manner based on the content, which makes identifying videos that match the story topic difficult and time consuming. Many existing work tackle this problem by integrating textual and visual concepts to group videos.

Our method treats story authoring and sharing as a collaborative process. It is different to other methods mainly in the sense that we integrate collaborators' contributions via social networking services to help story-telling. These contributions include adding comments, recommending relevant videos etc. With certain automated information processing, the method can effectively reduce the workload of an individual when authoring a story consisting of a large number of videos from various sources.

We assume that the story author has a number of friends or collaborators who have knowledge on the story topic and are willing to contribute to the story. It is common in the real world that these people participate in a same event or have mutual interests on the topics of the story. As stories are diverse and the group of people who may be interested in contributing them are dynamic, managing these users with different interests is a challenging problem. We address this problem by leveraging the power of social networks to organize a dynamic group of people who are contributing to a story.

The main technology that helps in managing and distributing audio and video content over the Internet is the content distribution networks (CDNs). Current approaches to provide content delivery networks warrant a provider to either setup his own computer, storage, broadband network, and other IT infrastructures or lease them from established 3rd party ISPs or telecommunication network providers (e.g., Akamai and Telstra). Both of the aforementioned approaches have been proven to be economically non-viable, hence out of reach for all (e.g., public libraries, universities, schools, SMEs, social network-based users etc.) but large enterprises. For example, a group of people (e.g., tourists or a school class on a field trip) intending to author and share stories of videos describing the points of interest would need acess to affordable video management and distribution technology. These videos may have been captured via their smartphones during their trip. Such typical use-cases of video story telling often require significant amount of effort from story authors to identify relevant videos, organize them based on topics or timelines and edit them to fit into story lines. Existing CDN search and distribution technologies are not suitable cope with the needs of these use-cases. The two aspects are intertwined. Data locality plays an important role in traditional CDNs. Data stored in a surrogate server in a CDN is likely to be accessed frequently by end-users nearby geographically in short time period. However, in the collaborative video authoring scenario, the data locality does not necessarily hold, e.g., a story author in Australia may have collaborators around the world.

The MediaWise Cloud project in CSIRO has developed the MCCO platform [5],[10],[11] to tackle the problem of traditional CDN technologies, with a focus on supporting such *dynamic content sharing and distribution* using cheap public cloud services (CPU, storage, and network). In particular, MCCO supports the following novel capabilities: (1) Exclusively utilize public Internet and public cloud infrastructure. The MCCO can utilize the cloud storage and CPU services from virtually any public cloud provider, while not requiring the ownership of private networks and servers. (2) Supply content management services that can be used by content managers (a class of end-users (e.g., school going kids, school teachers, etc.) that is currently not supported by CDNs) to manage content authoring and delivery workflows for their end users. (3) Support dynamic content delivery to enable collaborative activities. (4) Provide seamless and personalized end-user experience. The proposed Social Content Authoring and Sharing Platform (SCASP) leverages MCCO and demonstrates MCCO's capabilities as regards to content management and distribution.

The MCCO and SCASP platforms therefore treats the cloud as an intermediary between video sources and end-users. It makes use of the ubiquitousness of public cloud services to help video distribution and video story authoring. In our platform, the cloud is both an infrastructure provider and a social portal on which users interact with each other. The main objective of developing the proposed SCASP platform on top of MCCO is to transparently and dynamically place video data among geographically distributed cloud services to speed up video data access and leverages a user's social connections to collaboratively author video stories. The MCCO plays the role of an orchestrator of cloud services related to video production, delivery and consumption.

This paper builds on our previously published papers [5] [12]. As compared to our previous papers, we make following novel contributions here:

1) It makes story-telling a collaborative process and supports the main author of a story to interactively integrate collaborators' input including comments and recommendations;
2) The collaboration is dynamically organized by leveraging the services provided by social networks, in our case, Facebook services;
3) The platform aggregates collaborators' recommendations and comments. It is capable of processing certain information in an automatic manner, e.g., it can organize information based on topics and timeline to further help story authoring and sharing.
4) It supports collaborative video editing; It exploits flexible content distribution of MCCO for a group of possibly geographically distributed end-users who collaborate on a specific task such as video story authoring;

The rest of the paper is organized as follows: Section 2 summerises the related work; Section 3 introduces the architecture of our MCCO; Section 4 focuses on the SCASP and video editing in the MediaWise Cloud; Section 5 presents the details related to integration of SCASP, MCCO, and social network (Facebook); Section 6 articulates our experiments; and Section 7 concludes the paper.

2. Related Work

This section surveys the existing work for content authoring, including the video similarity search and the video authoring techniques.

Many search techniques have been proposed to identify the relevant videos of a given end-user query. Typical video search techniques can be classified into two categories: text-based and content-based [14]. In text-based approaches, tags or keywords are usually extracted from video titles or surrounding descriptions for searching the relevant videos. In content-based approaches, visual features, such as color histograms, grey intensity or some local interest descriptors, are extracted and used in video similarity matching to obtain a more objective video list or re-rank the results obtained from text-based search. In [15], question answering techniques are exploited to support personalized news video retrieval. In [9], Neo et. al proposed to leverage extractable video semantics with relevant external information sources for event-based analysis, which discovers the topic hierarchy and supports question answering. In [18], a hierarchical video content summarization technique was proposed to describe the categories, events and detailed information about the video in different levels for efficient video skimming and management. In [13], the textual and visual concepts are integrated for video browsing and autodocumentary. In [17], Zhou et.al proposed to convert each video into a compact word sequence that can be further used for video matching effectively and efficiently by extending string matching techniques. In [16], Zhou et.al proposed to exploit tensor series model for the search over video sharing community containing videos without large variations. Existing techniques have greatly improved the performance of video retrieval from different aspects. However, while the collaboration of users are not considered in [15], [9], other video retrieval techniques [18], [13], [17], [16] can not process the interaction between end-users and system at all.

Video authoring techniques have been proposed to edit videos for different applications such as e-Learning and entertainment. In [8], Meng et.al proposed a two-step compressed domain video editing approach which includes the production stage and the post-production stage. The original videos from cameras are compressed in the production stage and further compressed in postproduction stage. The editing tools allow end-users to perform different video authoring functions and video transformations for different visual effects. In [2], techniques have been proposed to compose a piece of video along a storyline with smooth flow of information and to customize the authoring under temporal playout constraints. In [3], Bianchi presented the AutoAuditorium System technology and operational characteristics, which can create a multi-camera video program of a lecture in real time automatically. In [4], authors presented a system for video capture and live transmission on mobile phones. These techniques were proposed for effectively video production in a single user operation environment, while the interaction of multiple users is not considered in process of video editing.

Our method treats story-telling as a collaborative process. It is different to other methods mainly in that we integrate collaborators' contributions via social networking services to help story-telling. These contributions include adding comments, recommending relevant videos etc. With certain automated information processing, the method can effectively reduce the workload of an individual when composing a story out of a large number of videos from various sources.

3. Mediawise Cloud Content Orchestrator

Recall that in MediaWise Cloud project we have developed a cloud-based generic and scalable software framework called MCCO for supporting the end-to-end lifecycle operations required for managing video content via clouds and the Internet. The MCCO exploits public clouds for offloading computing, storage, network, and content distribution functionalities in a cost effective manner. Detailed information on MCCO can be found in publications [10] and [11]. However, for the completeness of this paper we briefly describe the technical details of MCCO here.

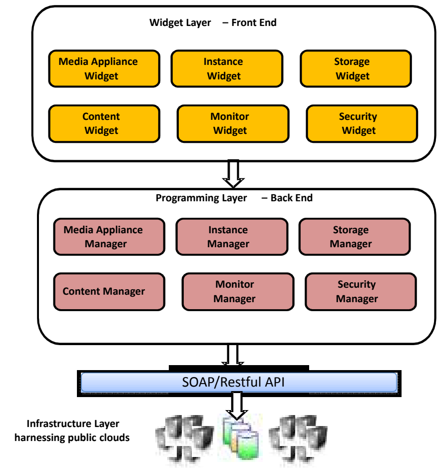

Fig. 1 : MediaWise Cloud Content Orchestrator Architecture

MCCO offers enhanced flexibility and elasticity as it inherits pay-as-you-go model from public cloud services. 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 services to monitoring their runtime QoS statistics (e.g., latency, utilization, and throughput). MCCO orchestrate public cloud services via open-source RESTFul APIs. It supports deployment, configuration and monitoring of content and cloud services using Web-based widgets. These widgets hide the underlying complexity related to cloud services and provide an easy do-it-yourself interface for content management. The high level architecture of MCCO is shown in Fig. 1. 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 inhouse 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 (CPU service).

3.1 Infrastructure Layer

This layer provides cloud-based hardware resources as virtualization enabled services such as CPU, storage, routers and switches that hosts the content management appliances such as streaming server, indexing server, and editing server. Hardware resources expose certain configuration that can be allocated to content management appliances. For example, a streaming server appliance available from Wowza[1] 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. In general, cloud providers manage resources at Infrastructure layer through hardware virtualization technologies 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 services to run at the same time. Each virtual service believes it has its own hardware. Virtualization isolates the hardware resources from each other, thereby making fault tolerant and isolated security behaviour possible.

3.2 Programming Layer

This layer implements the logic for interfaces exposed by Widget layer. For example, the Media Appliance Manager implements Cloud service API that allows Appliance Widget to list the set of content management 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 services, 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.

3.3 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.

1) **Media (Content Management) Appliance Widget:** It lists the set of media appliances associated with content owner's account. In general, an appliance is pre-configured, selfcontained, virtualization-enabled, and pre-built software resources or appliances (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 (Content Delivery Network).
2) **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.
3) **Storage Widget:** It allows content owners to upload content and media appliances to storage service (e.g., Amazon S3). Cloud storage services provide a highly durable and available storage for a variety of content types, including web applications and multi-media files.
4) **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.
5) **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.
6) **Security Widget:** It manages all the authentication and authorization credentials related to orchestrating content (e.g. content access secret key) and cloud services (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.

[1]. http://www.wowza.com/

4. Social Content Authoring and Sharing Platform (SCASP)

There are the following roles in the SCASP:

1) **Content provider:** A content provider owns data objects, such as video/audio and data records that can be shared among a group of end-users. A content provider has a data repository but limited capability to effectively distribute its data objects to end-users. It therefore delegates the data distribution functionality to a third party content distributor. The content provider may request information flow control when sharing certain data objects to different groups of end-users. The content distributor needs to take care of this aspect as well. The data objects in the data repository are indexed by titles, authors, publishing dates and tags.
2) **Story authors:** A content author makes use of the content from various content provider to compose stories for learning and curating purpose. There is a large number of videos in the data repositories. It is a time consuming process for the author to compose a story. The system enables the author and a group of contributors to collaborate on the story authoring. A content end-user has a list of social links maintained by the social information aggregator.
3) **Story contributors:** A story contributor either comments on the story under composition or recommends videos related to the story.

4) **Social information aggregator:** A social information aggregator is an application running on a Facebook like system that aggregates the social links of subscribers and maintains their content access information. The aggregator is able to access public information of an end-user as well as some personal information granted by the end-user. In addition, it is capable of aggregating information about a group of users.
5) **Content distributor:** a content distributor (MCCO in our case) is a service provider that makes use of end-user information from a social information aggregator to optimize the content access experience of end-users of a content provider. It replicates the content from the content provider by using virtualized services from a cloud infrastructure provider to optimize the access experience of content end-users.
6) **Cloud infrastructure provider:** A cloud infrastructure provider offers geographically distributed physical resources as virtualization enabled services, including CPU and storage services for the MCCO to orchestrate.

The content distributor (MCCO) is in the center and connects the content provider, the social information aggregator and the cloud infrastructure provider. The design is to make use of cloud infrastructure and social information for the following purposes:

1) better content placement based on end-users' locations.
2) better content customization for a group of end-users of a particular content provider to curate data objects of certain topics.

In this paper, we focus on the collaboration between the story author and the story contributors.

4.1 System Overview

The overview of the system is illustrated in Fig. 2.

We assume that the story author has a number of friends or collaborators who have knowledge on the story and are willing to contribute to the story. It is common in the real world that these people participate in a same event or have mutual interests on the topics of the story. As stories are diverse and the group of people who may be interested in contributing them are dynamic, managing these end-users with different interests is a challenging problem. We address this problem by leveraging the power of social networks to organize a dynamic group of people for contributing to a story.

The SCASP mainly consists of the following two components:

1) A Web user interface as a working bench for a story author to compose stories. The user interface requires the story author to login using her Facebook account. The authentication is done using OAuth 2.0 protocol [6]. The access of user information is through Facebook Open Graph API.
2) A social information aggregator that does the following tasks:
   a) publishes the story author's activities to a social networking site. The activities include the metadata of the story under authoring as well as ongoing editing activities;

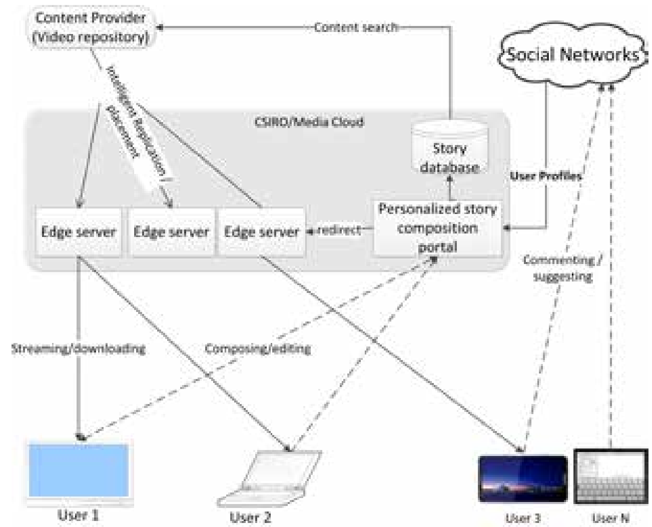

Fig. 2 : Overview of SCASP.

   b) retrieves relevant information contributed by collaborators, including a collaborator's comments to the story and recommended relevant videos by the collaborator;
   c) processes the information and discovers certain topics as well as timelines to present to the story author

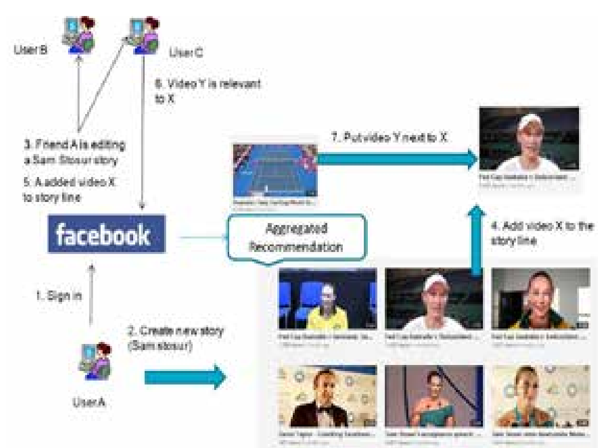

Fig. 3 : An example of the basic information flow in the platform.

Fig. 3 gives an example that shows the basic information flow in the platform. In this scenario, an author creates a story about Sam Stosur, an Australian tennis player. Firstly, the author, e.g., a sports news editor, signs in the platform using her Facebook account. Secondly, she submits a query with keyword "Sam Stosur" to search for videos to start a story. The platform passes the query to the search services of video repositories. Each video repository returns a list of relevant videos. The author may select one and add it to the storyboard. Thirdly, the platform automatically publish the action to Facebook. Fourthly, a collaborator who is in the author's friend list in Facebook may add comments or recommend videos that are related to the story. In the next step, the recommender subsystem processes these comments and recommendations. It then presents recommendations to the story author who may take the recommendation and make changes on the storyboard.

Apparently, an active collaborator is likely to recommend videos that are more relevant to the story line or add comments that can help the author to improve the story line. However, when there are a number of collaborators, reviewing recommendations and comments may become a time consuming process. In our platform, we provide a recommendation service for easing the task.

4.2 Recommender

The recommendation service fits into the system as shown in Fig. 4.

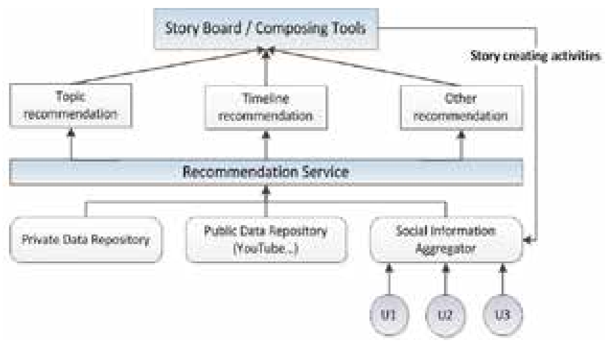

Fig. 4 : The recommender architecture in SCASP.

The recommender refines both video and text data from video sources and the social information aggregator. A video data source can be a private data repository or a public data repository. The social information aggregator maintains information such as the author's story creating activities, the collaborator list and manages the input from collaborators. The recommendation service produces the following three types of recommendations to the story author:

1) **Relevant topics:** a list of videos categorized by topics that can be derived from the story line under authoring. A collaborator is capable of identifying topics among a set of videos and recommending these videos to the author.
2) **Timeline recommendation:** a group of collaborators may annotate videos based on the event time in the videos. The service can present related videos along the time line to help story authoring.
3) **Other recommendation:** mainly contains comments collected from the social information aggregator that have no clear structures.

4.3 Video Editing or Authoring

Video editing service is the core component in SCASP that interacts with content end-users and enables multiple end-users to perform collaborative video authoring simultaneously. General video authoring operations include video merging and splitting. We only focus on video authoring that creates videos based on existing short video segments in story authoring system.

Video authoring is an important process in video production, which delivers to end-users a coherent presentation of an event or narrative based on a candidate set of available video segments. This service has been driven by its wide applications in e-learning and entertainment domains. For example, in e-learning, teachers may compose a programming lecture using several existing related video lectures. In entertainment, it is common that movie producers compose a whole movie based on a number of short segment video scenes. In social community, end-users may produce new clips based on existing videos for recreation. However, traditional techniques for video authoring are conducted by a single user over a single machine. Although recent system has been developed to permit multiple users to compose a single video together [1], it is still challenging to create a video that reflects an inherent story in a collaborative way.

Motivated by the weak points of existing video authoring techniques, we perform a collaborative video production in our story authoring and sharing platform (SCASP), which takes the following characteristics into consideration. i.e. the system supports multi-user collaboration in the video authoring in a social community. Interactive video production is performed by receiving a number of video clips that are recommended by the contributors and selected by the video author of the story telling platform, conducting editing operations using the API functions provided by the Xuggler video editing framework, and returning the final produced new video of the composite story. We choose Xuggler framework for our video editing implementation because of its advantages over other tools. For example, it allows Java programs to decode, encode, and experience almost any video format without codecs problems. It can simplify programming from Java, and perform video editing in reasonable fast speed.

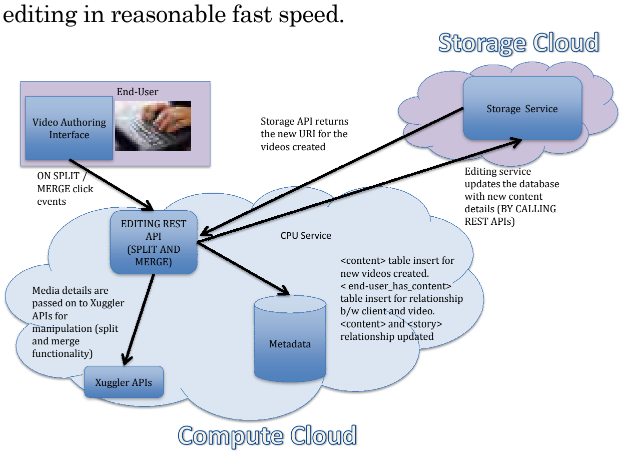

Fig. 5 : Architecture of Video Authoring Service.

The flow of the application starts, when an end-user interacts with SCASP via its from the Web Application Interface (see Fig. 5) (which we also call as MediaWise Web Portal) and thereby makes a call to the MCCO that controls the orchestration operations related cloud-based CPU services that hosts the editing appliance. The editing appliance implements the video authoring service by using Xuggler editing framework. as a web application. The editing appliance supports the split and merge functionality, as two different video authoring operations. The details of the url location of content and other request data are passed on to the Xuggler API block, where the actual operation happens, and the content created is kept in the instance, which is passed on to storage cloud, via MCCO storage manager.

4.4 Usage

A typical scenario for using the system is shown as below:

1) The story author signs into the system using her Facebook account;
2) The author uses the search functionality to find videos. This will show how the search results are presented and how a story is initiated;
3) The collaborators sign into their Facebook account and see the actions of the story author;
4) A collaborator recommends related videos and comments on the story;
5) The storyboard organizes collaborators' input based on different topics. We will show how the story author uses the recommendations to further develop the story.

The whole video story authoring is performed by several steps. First, a story author signs into the system by inputting her Facebook account and password. Then, she can obtain a collection video candidates for story authoring task by performing of the search function in video sharing social communities, such as Youtube and Metacafe etc. as shown in Fig. 6. Following that, collaborators sign into the system, where the operations of the story author are visible to the collaborators, and collaborators participate in the activities such as recommending video components, and commenting the story as shown in Fig. 7. Recommendations are organized in a story board based on their topics. Finally, the story author composes the videos and produces a complete story based on the recommendations.

Fig. 6 : User interface: searching for videos for composing a story.

Fig. 7 : User interface: commenting and recommending related videos to the story author.

5. System Implementation

The MediaWise Cloud [5] consists of portal, MCCO, and the SCASP. The SCASP includes the collaborative video story authoring component, the recommender and the video editing components, which are implemented as Web services at the SaaS-level. Each of these components can be hosted efficiently by the MCCO [10][11] on Amazon EC2 and Amazon S3 cloud infrastructure. The content providers using MCCO can host the media-related content such as the audio-visual files at different geographical locations. The story authors and story contributors use the MediaWise portal to access, search, manipulate and discuss the content.

The MediaWise Portal uses a standard Model-View-Controller (MVC) pattern to respond to requests from the web application using the Spring framework to provide the support to MVC interactions. The controller manages the processing from the Web application including the home page, the video player page, the story creation page, the content mash up page and the local database. All the operations supported by the MediaWise Portal and the MCCO are implemented using the RESTful Web APIs. For example, user login and story comments are implemented using the Spring- Social Framework APIs, MCCO operations are supported by the Amazon EC, Amazon S3 and SmartGWT APIs. We also developed a number of APIs to manage the backend system, i.e., to perform complex operations such as performing content search and story-related operations. Our backend system comprises of the MediaWise Database.

5.1 MediaWise Cloud database

The MediaWise database is a MySql database that has its own REST service to query, fetch and update the data related to the user, content and stories. The REST service provides services to fetch the metadata in the form of keywords for videos using the queries such as: select all videos or select all videos with a specific keyword. Fig. 8 show three table out of a total of eight from the MediaWise database used to manipulate the content and its related stories. Using these tables, information related to content such as where it is stored, its related keywords and owner is retrieved. Further, content related to particular stories can also be easily manipulated.

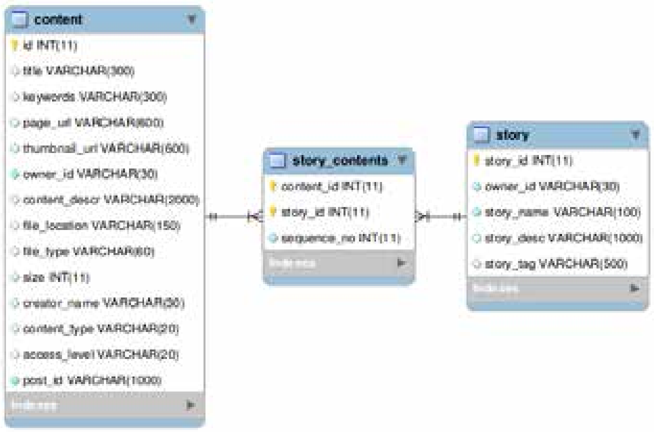

Fig. 8 : The database schema for video content and stories.

As mentioned previously, there are several other tables in the MediaWise database to include data related to the cloud providers, quality of service statistics and end-users. Due to sake of brevity, we do not discuss them in the paper.

5.1.1 Example: searching by keyword

In this example, consider a case whether an end-user using the MediaWise portal, enters the keywords related to particular videos he/she is looking for. The user submits the query to the application. The controller manages the processing of the query: first sending the query to the database server via a REST call. The database server, using a utility class, queries the MySql data and converts the SQL result set into an XML package that is returned back to the controller. Lastly, the MVC model is populated with video objects derived from the XML and then displayed on the web page with the results. Fig. 10 shows the returned XML response after making a REST call.

5.2 Facebook integration

As mentioned previously, *story authors* and *story contributors* logs in to the MediaWise Portal and comments on the story using the social media, in our case Facebook. Our web application interacts with Facebook to update the user's "timeline" and retrieve comments added to the timeline post. When a video is viewed via the web application, an entry is created

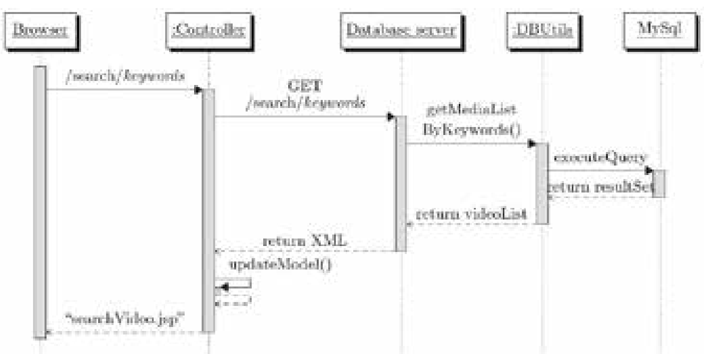

Fig. 9 : Sequence diagram showing the process of content search using the MediaWise portal.

Fig. 10 : Returned XML response regarding content using the REST call.

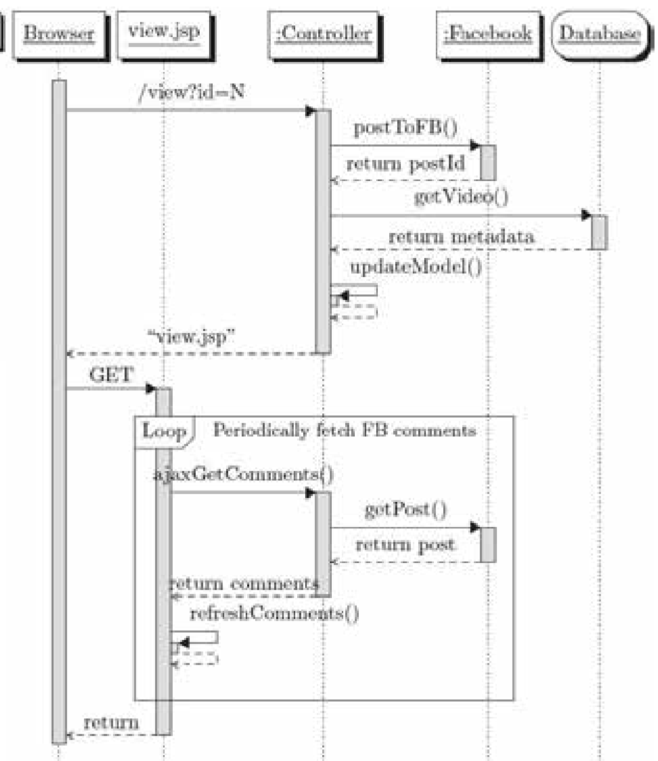

Fig. 11 : Sequence diagram showing the process of MediaWise Portal interaction via the web browser with Facebook.

5.3 Content mashup and editing

Another novel feature of the MediaWise Cloud is the ability to aggregate content from local and remote sources such as YouTube. The metadata related to the new videos is then stored in the database such that the end users can easily search the new content, associate the content to stories and comment on them. In particular, the metadata from the selected video, such as title, description and keywords, is extracted and used to populate a "editing" web page. The end-user can then modify this metadata as they require and save the information into the local database. The video then becomes the part of the system. The story author and the story contributers can then manipulate the new content. Fig. 12 shows the sequence diagram for metadata retrieval for YouTube videos and its storage in the MediaWise database.

6. Experimental Analysis

We considered a scenario where an end user at CSIRO, Canberra, Australia, performed merge and split operations

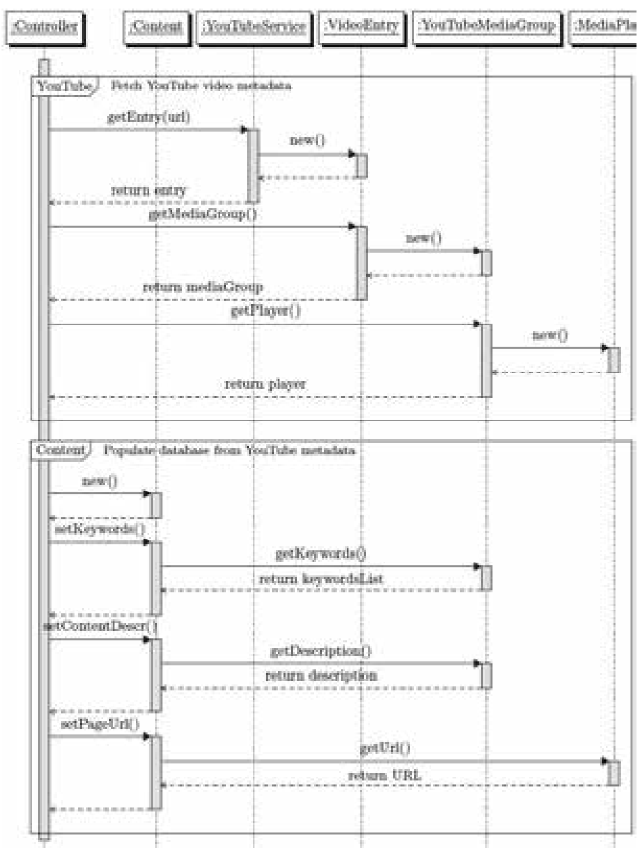

Fig. 12 : Sequence diagram showing the process of fetching and storage of YouTube metadata for video search.

using the MediaWise Portal. We considered different file sizes: 2 MB, 20 MB, 40 MB, 351 MB, 478 MB. We also considered different combinations of the merge and split operations. For example, we performed experiments by merging 2 to 5 files together. For results evaluation, we merged files of different sizes randomly and considered two metrics i.e., file merge time and file upload time. The merge and split operations were performed using a MacBook Pro running OSX 10.8.2 on Intel i7 2.2 GHz processor with 4GB RAM.

Table 1 shows the results related to the merge and upload operation's performed using the MediaWise portal. We conclude from the results that both merge and upload time depends mainly on file sizes and not on the number of files considered for the operations. After each merge operation, the file was automatically uploaded to the Amazon S3 datacenter in USA i.e.,us-east-1. We note that as the location of the datacenter is far from Australia the upload time is quite high ranging from 124 seconds approx. to 2435 seconds approx. depending on file sizes.

Table 2 shows the results related to split operations.We considered three files of varying sizes and video duration. We performed split operations at different time durations. For example, for a file with 2624 seconds duration, we performed a split operation to create a new file of length 262 seconds, i.e., 10% of the file was considered. We concluded from our results that the time to perform split (split time) depends on the length of the final video required by the end user. It is also depended on the quality of the video. For example, lower quality video (file 1)will take lesser time compared to the higher quality video (file 3). Finally, after the split operation, the new files were uploaded to Amazon S3. The upload time is proportional to the file size.

7. Conclusion

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 social content authoring and sharing platform and its integration with MediaWise cloud content orchestrator platform. The proposed platform enables collaborative video story authoring system over social networks, and demonstrated our capability of handling video authoring issues in cloud environment. We also conducted some experiments, which further proves the feasibility of our approach.

In this article, we presented some initial thoughts on collaborative multimedia cloud computing and our preliminary research in this area. The research and development efforts in multimedia cloud computing is at early stage and many problems still need investigation. For example, ensuring QoS for different multimedia operations (e.g., production, storage, digitalization, and consumption) needs further research. Other open research problems in this area includes media security, location aware media placement, media overlay, and peer-topeer architecture for engineering media services.

TABLE 1
Performance analysis of merge and upload operations pertaing to 5 different video files. Y signifies that the file is used for merge operation.

Video File1 (2MB)	Video File2 (20MB)	Video File3 (40MB)	Video File4 (351MB)	Video File5 (478MB)	No of Videos	File Merge Time (sec)	File Upload Time (sec)
Y	Y	N	N	N	2	124.47	212.45
Y	N	Y	N	N	2	245.40	374.09
Y	N	N	Y	N	2	797.12	13750.64
Y	Y	Y	N	N	3	521.78	612.56
Y	Y	Y	Y	N	4	1179.45	21896.79
Y	Y	Y	Y	Y	5	2435.90	43672.89

TABLE 2
Performance analysis of split and upload operations.

Video	% of video split (duration)	Split Time(sec)	Upload Time(sec)
Video 1, 40 MB, 60 sec	10 (60 sec)	34.69	49.64
Video 1, 40 MB, 300 sec	50(300 sec)	116.38	189.67
Video 1, 40 MB, 540 sec	90(540 sec)	216.35	339.72
Video 2, 367 MB,2624 sec	10(262 sec)	163.67	2407.65
Video 2, 367 MB,2624 sec	50(1312 sec)	759.76	12659.54
Video 3,713 MB,4957 sec	10(496 sec)	353.28	478.90
Video 3,713 MB,4957 sec	50(2479 sec)	1289.80	23589.61

References

[1] http://www.youtube.com/editor.

[2] Gulrukh Ahanger and Thomas D. C. Little. Automatic composition techniques for video production. *IEEE Trans. on Knowl. and Data Eng.*, 10(6):967–987, November 1998.

[3] Michael Bianchi. Automatic video production of lectures using an intelligent and aware environment. In *Proceedings of the 3rd international conference on Mobile and ubiquitous multimedia*, MUM '04, pages 117–123, New York, NY, USA, 2004. ACM.

[4] A. Engstr¨om, M. Esbj¨ornsson, and O. Juhlin. Mobile collaborative live video mixing. In *Proceedings of the 10th international conference on Human computer interaction with mobile devices and services*, MobileHCI '08, pages 157–166, New York, NY, USA, 2008. ACM.

[5] D. Georgakopoulos, R. Ranjan, K. Mitra, and X. Zhou. Mediawise designing a smart media cloud. In *International Conference on Advances in Cloud Computing*, 2012.

[6] E. Hammer-Lahav, D. Recordon, and D. Hardt. The oauth 2.0 authorization protocol. Network Working Group Internet-Drafts, http://tools.ietf.org/pdf/draft-ietf-oauth-v2-12.pdf.

[7] Richang Hong, Jinhui Tang, Hung-Khoon Tan, Chong-Wah Ngo, Shuicheng Yan, and Tat-Seng Chua. Beyond search: Event-driven summarization for web videos. *ACM Trans. Multimedia Comput. Commun. Appl.*, 7(4):35:1–35:18, December 2011.

[8] Jianhao Meng and Shih-Fu Chang. Cveps - a compressed video editing and parsing system. In *Proceedings of the fourth ACM international conference on Multimedia*, MULTIMEDIA '96, pages 43-53, New York, NY, USA, 1996. ACM.

[9] Shi-Yong Neo, Yuanyuan Ran, Hai-Kiat Goh, Yantao Zheng, Tat-Seng Chua, and Jintao Li. The use of topic evolution to help users browse and find answers in news video corpus. In *ACM Multimedia*, pages 198–207, 2007.

[10] R. Ranjan, K. Mitra, and D. Georgakopoulos. Mediawise cloud content orchestrator. *Journal of Internet Services and Applications* (Accepted October 2012).

[11] R. Ranjan, K. Mitra, S. Saha, D. Georgakopoulos, and A. Zaslavsky. Do-it-yourself content delivery network orchestrator. *Web Information Systems Engineering-WISE 2012*, pages 789–791, 2012.

[12] C. Wang, M. Meng, X. Zhou, and R. Ranjan. A social network based collaborative story composition platform. *International Conference on Service Oriented Computing-ICSOC 2012.*

[13] Xiao Wu, Chong-Wah Ngo, and Qing Li. Threading and autodocumenting in news videos. *IEEE Signal Processing Magazine*, 23(2):59-68, 2006.

[14] Bo Yang, Tao Mei, Xian-Sheng Hua, Linjun Yang, Shi- Qiang Yang, and Mingjing Li. Online video recommendation based on multimodal fusion and relevance feedback. In *Proceedings of the 6th ACM international conference on Image and video retrieval*, CIVR '07, pages 73-80, New York, NY, USA, 2007. ACM.

[15] Hui Yang, Lekha Chaisorn, Yunlong Zhao, Shi-Yong Neo, and Tat-Seng Chua. Videoqa: question answering on news video. In ACM Multimedia, pages 632–641, 2003.

[16] Xiangmin Zhou, Lei Chen, and Xiaofang Zhou. Structure tensor series-based large scale near-duplicate video retrieval. *IEEE Transactions on Multimedia*, 14(4):1220-1233, 2012.

[17] Xiangmin Zhou, Xiaofang Zhou, Lei Chen, Yanfeng Shu, Athman Bouguettaya, and John A. Taylor. Adaptive subspace symbolization for content-based video detection. *IEEE Trans. Knowl. Data Eng.*, 22(10):1372–1387, 2010.

[18] Xingquan Zhu, Jianping Fan, Ahmed K. Elmagarmid, and Xindong Wu. Hierarchical video content description and summarization using unified semantic and visual similarity. Multimedia Syst., 9(1):31-53, 2003.

About the Authors

Chen Wang received his Ph.D. from Nanjing University. He is a senior research scientist at CSIRO ICT Centre, Australia. His research interests are primarily in distributed, parallel and trustworthy systems. His current work focuses on accountable distributed systems, resource management in cloud computing and demand response algorithms in the smart grid.

Rajiv Ranjan received his Ph.D. from the University of Melbourne (2009). He is a scientist at CSIRO ICT Centre, Australia. His research interests are primarily in cloud, grid, peer-to-peer, and service computing systems. His current work focuses on QoS optimizations in multimedia cloud computing systems.

Xiangmin Zhou received the Ph.D. degree in computer science from the University of Queensland, Brisbane, Australia, in 2008. Currently, she is a Research Scientist at CSIRO ICT Centre, Australia. Her research interests include multimedia databases and streams, query processing and query optimization, social network analysis and mining, and cloud computing.

Karan Mitra is a Research Intern in CSIRO Canberra ICT Center, Australia. His research interests include quality of experience modelling, measurement and prediction, next generation wireless networks, mobility management protocols, cognitive wireless networks and context-aware computing.

Suhit Saha is a Research Intern in CSIRO Canberra ICT Center, Australia. He is a Masters student at College of Engineering and Computer Science at Australian National University, and has been in software development industry for past 3 years, in which he has made contributions to Software Systems made for Insurance firms, after which he had been a part of a team developing PC and cross-platform mobile games.

Meng Meng is a Research Intern in CSIRO Canberra ICT Center, Australia. She has completed a Masters degree in information technology from the University of New South Wales (2012).

Dimitrios Georgakopoulos is currently a Research Director at the CSIRO ICT Centre, Australia, where he leads the Information Engineering Laboratory. Prior to CSIRO, Dimitrios held research and management positions in several industrial laboratories in the US, including Telcordia, where he helped found Telcordia's Research Centers in Austin, Texas, and Poznan, Poland, Microelectronics and Computer Corporation, GTE (Verizon) Laboratories, and Bellcore. He is Adjunct Professor at the Australian National University. Dimitrios has led major (multi-year/multi-million) projects in four continents and he has extensively published in the areas of business process management, complex event processing, collaboration management, and data management. He served as the General or Program Chair of a dozen major conferences as well as many other smaller conferences or workshops. Dimitrios twice received outstanding paper awards from the IEEE Computer Society. He was nominated for the Computerworld Smithsonian Award in Science, and received a Verizon Excellence Award.

Lizhe Wang is a professor at Center for Earth Observation and Digital Earth of Chinese Academy of Sciences (CAS) and a ”ChuTian Scholar“ Chair Professor of China University of Geoscience (CUG). Dr. Wang currently leads a group in CAS with a focus of HPC and data-intensive computing for earth observation applications. He also co-leads the scientific computing lab in CUG with a focus on scientific cloud computing and GPGPU computing. Dr. Lizhe Wang is an IEEE senior member and ACM professional member.

Peter Thew has been in the computing industry for over 30 years, and has worked in Canberra, interstate and overseas. During his career he has worked on a large number of diverse projects including satellite tracking fishing vessels in the South Pacific, saving historical documents from World War Two and working on several personnel systems. He has spent the last three years as the project manager for the ”Flood Extent Monitoring“ which satellite imagery to identify surface water to produce daily flood maps for Australia. The flood maps are combined with the CSIRO's Digital Elevation Model of Australia to estimate the volume of water within a flood. He is currently working on a number of projects including Mediawise and further flood extent modelling work along the Murray River.