

RESEARCH

Open Access

# Cloud-based active content collaboration platform using multimedia processing

Byong John Han<sup>1</sup>, In-Yong Jung<sup>1</sup>, Ki-Hyun Kim<sup>1</sup>, Do-kwang Lee<sup>1</sup>, Seungmin Rho<sup>2</sup> and Chang-sung Jeong<sup>1\*</sup>

## Abstract

In this article, a new content-centric collaboration platform called active content collaboration platform which supports automated content-centric collaboration on cloud system is presented. It supports event-driven automatic collaboration by specifying each active work based on active work description language and automating the execution of collaborative task flow composed of active works. It supports a modularized and extensible architecture by making its components in separate modules. Also, it provides a scalable high-performance architecture by supporting multi-level active work processing in active work execution engine, and allocating virtual machines for the computation intensive high-level action such as volume rendering through auto-scalable allocation on cloud system. For the experiment and evaluation, we shall show the results of implementing the collaborative medical application on our system where computation intensive application like a volume rendering is used for MRI analysis.

**Keywords:** Cloud collaboration, Active content, Content-centered, Multimedia processing

## 1. Introduction

Nowadays, collaboration organization becomes more complicated and grows bigger. Therefore, it requires huge collaboration platform through networks. Modern softwares for content-centric collaborations support web applications for business tasks based on traditional software systems such as enterprise content management (ECM) [1] or repository softwares. Those content-centric collaboration platforms reduce network traffic for collaboration content transfer. Previous collaboration platforms were developed to provide private, isolated, and unique methods used for specific organization [2]. It means that those softwares should be customized for each organization for providing collaboration services, and hence it is hard to retain them flexibly. Furthermore, collaboration between organizations with different systems requires additional cost, and makes security problems. This limitation of isolated collaboration system makes collaboration between separated organizations harder.

In this article, we propose a new active content collaboration platform (ACCP) which supports automated

event-driven service called “active work” on cloud system. It provides a modularized architecture and cloud management system to provide system extendibility so we can attach new functional module into ACCP easily. Also, ACCP provides separated content-centric virtual collaborative work space called “active content repository (ACR)”. All contents stored in ACR are called “active content” which can be managed automatically by our collaboration platform.

Our platform has three key features. The first feature is that it provides virtual collaborative content workspace to collaboration users. With this content-centric collaboration system, collaboration users can interact with each other, and share their contents via network. The second feature is that it provides functions to execute a unique task flow with automated management engine based on event-driven scheme for separated content spaces. Each content space is created for their unique purpose. If there arises any action by user in content space, collaboration platform creates events and collects them to carry out its corresponding. Those collected events are validated by task flow management engine which execute a task according to the predefined task flow. The task flow can be monitored, and managed by collaboration users. The third feature is that it

\* Correspondence: csjeong@korea.ac.kr

<sup>1</sup>Department of Electronic Engineering, Korea University, Seoul, Republic of Korea

Full list of author information is available at the end of the article

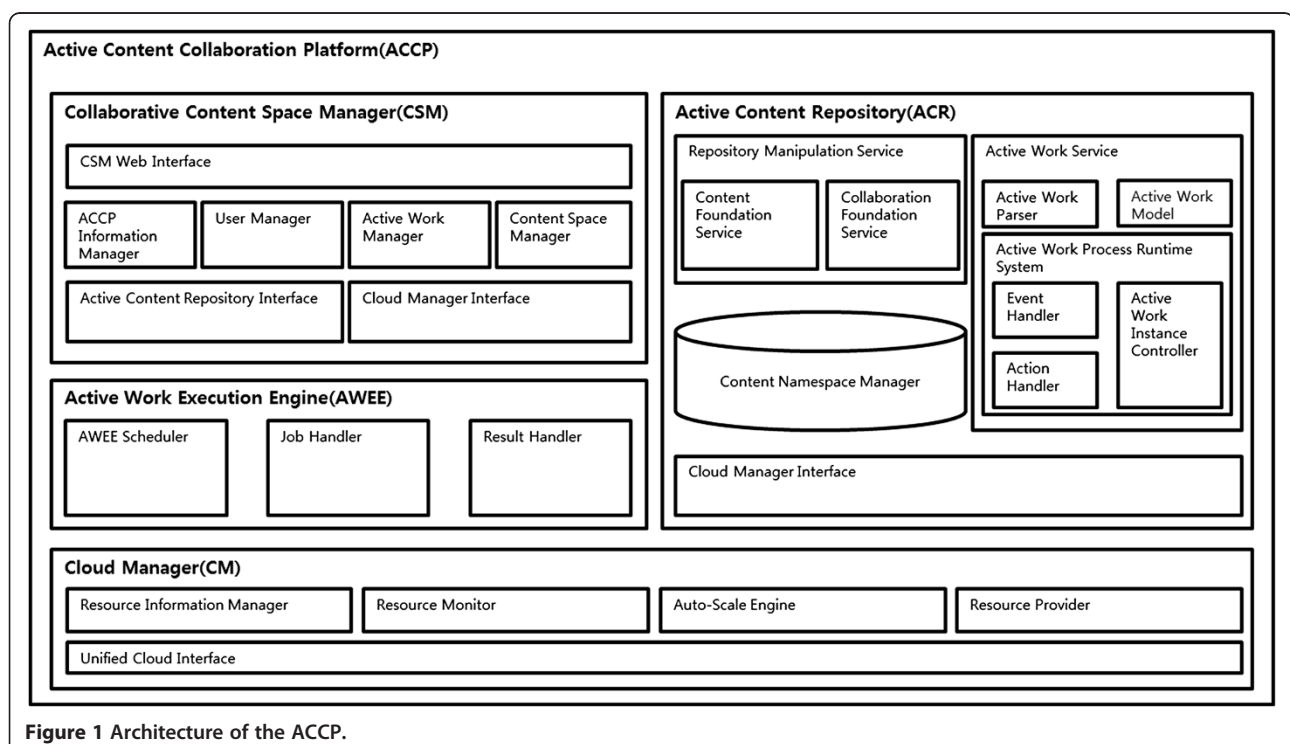
provides modularized and scalable architecture to support extensibility, flexibility, and quality of service by exploiting cloud infrastructure system. With modularized architecture, the manager of collaboration platform can attach various multimedia processing functions for each organization easily. For example, in case of music collaboration, we can attach music producing application instantly. Also, the huge collaboration platform needs to balance its work load by controlling repository size and computational load.

The outline of this article is as follows. Section 2 describes related works and Section 3 describes the main architecture of ACCP. Section 4 describes the implementation of our platform and the performance evaluation. Section 5 describes conclusion of this article.

## 2. Related works

Various commercial ECM [1], open-sourced content management system (CMS) [3], and repository soft wares for content-centered collaboration are in the market such as Microsoft Sharepoint [4], EMC Documentum [5], Alfresco [6], Drupal [7], Joomla [8], DSpace [9], and so on. Alfresco opened their layered 3-tier architecture (Content Repository, Application Server, and Collaborative Web Apps) and clustered repository for enterprise scale massive content management. It provides standardized interfaces based on JSR-170 [10], CMIS v1.0 [11],

and unique web service API. Furthermore, it supports simple workflow for automation via embedded jBPM [12] workflow module. HP Labs Fractal Project [13] suggested cloud collaboration with event-driven automation named ‘active behavior’, and introduced key features for content-centric cloud collaboration. There are many web-based SaaS (Software as a Service) [14] built on RIA (Rich Internet Application) [15] technology such as Google Wave [16], Zoho.com [17], Zimbra [18], and so on. These services show examples as customizable workspaces and applications for clients on cloud ecosystems. Also, there are many collaborative systems such as SonART [19], ConnectBoard [20], and Virtual Campfire [21] which support multimedia processing such as image display and signal processing. SonART is a flexible multimedia environment which allows for networked collaborative interaction with applications for art, science, and industry. It provides an open-ended framework for integration of image and audio processing methods with flexible network features. ConnectBoard is capable of capturing and delivering realistic, genuine eye contact as well as accurate gaze awareness with respect to shared media. Virtual Campfire enables the flexible realization of community information systems with divers and complex multimedia content such as videos, images, and 3D data on smart mobile device. These services show examples as dynamic multimedia collaboration on network.



**Figure 1** Architecture of the ACCP.

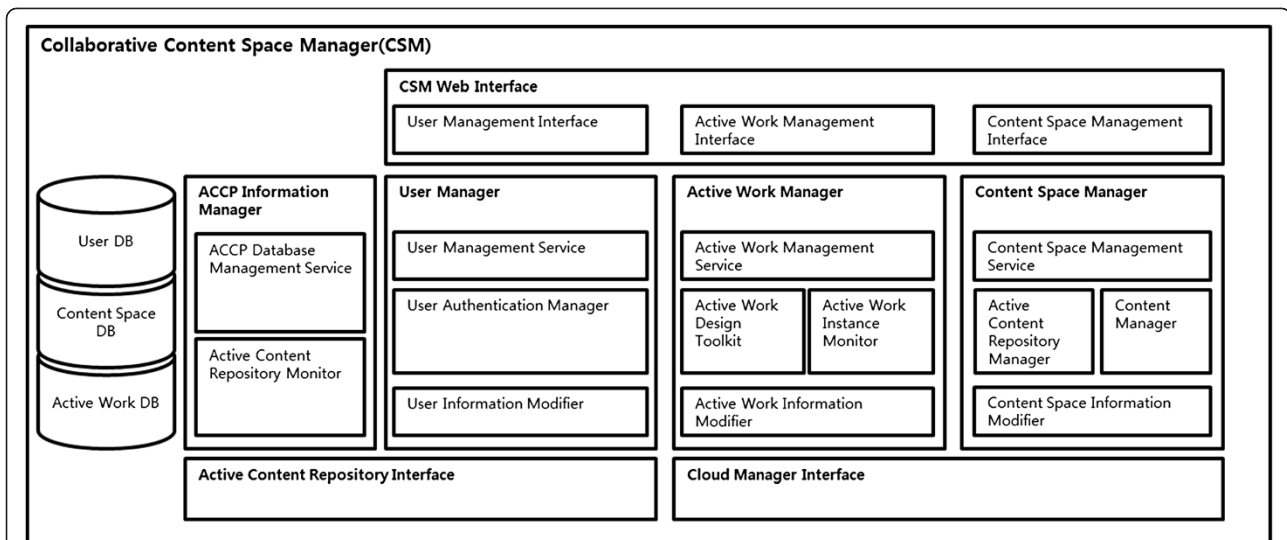


Figure 2 Architecture of CSM.

### 3. Active content collaboration platform

#### 3.1. Key features

In order to support scalable event-driven content collaboration system on cloud, ACCP has the following key features.

1. *Content-centric collaboration support:* Previous collaboration paradigm was user-to-user collaboration by using communication tools. This kind of collaboration causes huge network traffic and inconvenience in collaboration, because all contents

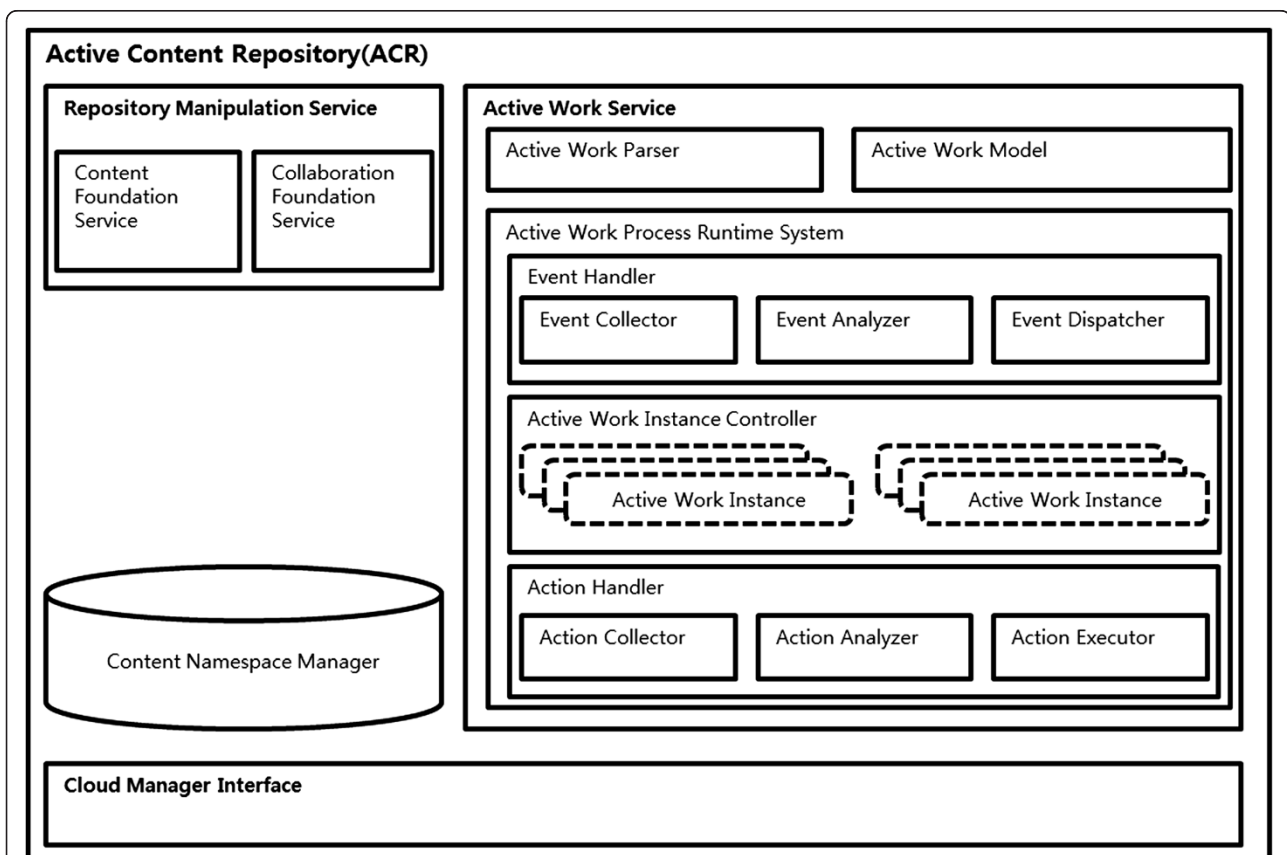


Figure 3 Architecture of ACR.

used in collaboration should be transferred by network, and collaboration users should work for all steps of task. To solve those problems, content-centric collaboration is developed. In content-centric collaboration, there is a centralized CMS which can be used as a collaborative workspace by users. With centralized CMS, we can reduce network traffic and manages collaboration platform easily.

2. *Event-driven automatic collaboration support*: ACCP supports collaboration with automated event-driven collaboration. This event-driven automation is very useful for collaboration support, because generated events can be used as a trigger of advancing collaboration task flow. If there is any change in content space in ACCP, it generates events due to the information of action which causes changes. ACCP also supports users to describe a collaboration task flow. This service enables users to create their own collaboration task flow. Collaboration task flow is composed of event, action, and rule. When an event is generated, system catches it, and executes some actions due to the rule defined.
3. *Scalable high performance cloud-based architecture*: ACCP supports a modularized and extensible architecture based on cloud technology in order to manage massive user groups and contents. ACCP offers a scalable high-performance architecture through auto-scalable allocation for its components on cloud. High computational tasks are allocated in different virtual resources by cloud management component. Furthermore, ACCP supports multi-level active work processing: low-level action such as simple content manipulation is processed in content repository, while computation intensive high-level

action such as content transformation, rendering, encoding, and compression are out-sourced, and managed in active work component.

### 3.2. Architecture

ACCP is composed of four components: Collaborative Content Space Manager (CSM), ACR, Active Work Execution Engine (AWEE), and Cloud Manager (CM). Figure 1 shows an overall architecture of ACCP.

CSM provides a user interface as well as various management functions for collaboration. ACR is a work space for a collaboration group which provides its own Repository Manipulation Service and Active Work Service for collaboration. AWEE connects external application with ACR, and executes various distributed high-performance application for collaboration in professional task area. Finally, CM is a management component for allocating and managing computational resource used to run ACCP.

#### 3.2.1. CSM

CSM provides a web user interface for collaborative system management service. Figure 2 shows the detailed architecture of CSM.

CSM hosts a group of content spaces for collaboration between massive user groups. The web user interface of CSM provides three interfaces for user management, active work management, and content space management. All information of users is managed by accessing User Management Interface. Collaboration users can modify their information and group information. Users can create and modify their unique active work through using active work Management Interface. Content Space Management Interface provides functions for creating a new

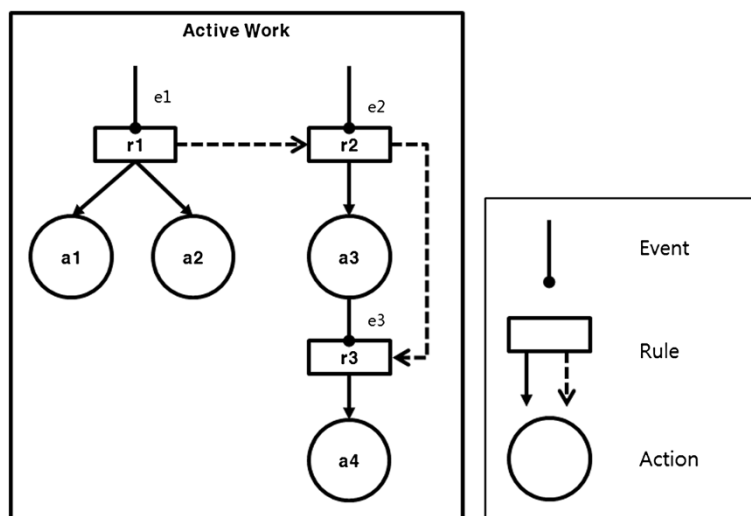
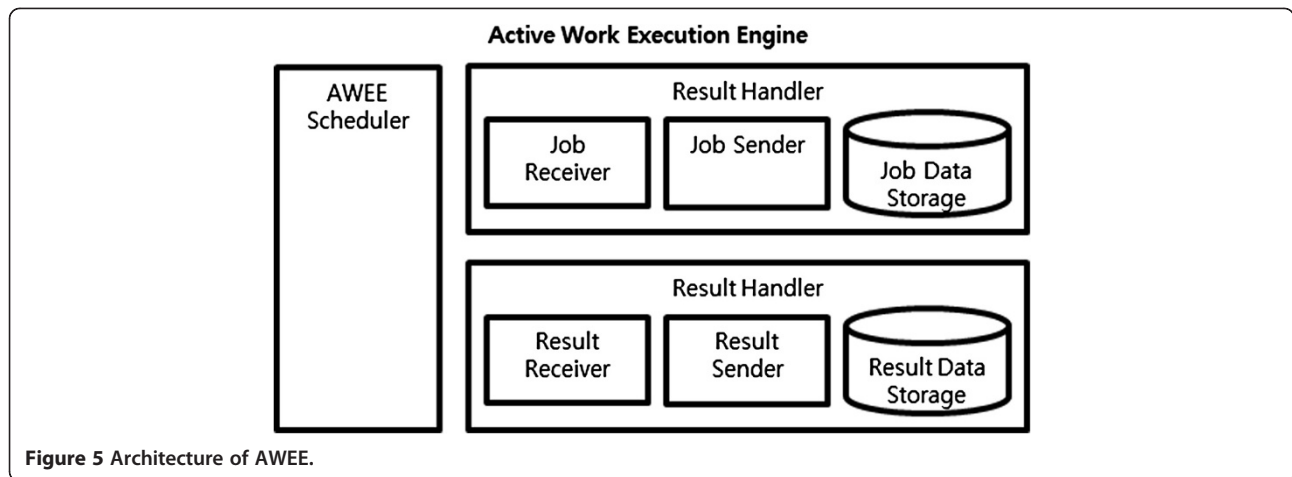


Figure 4 Example of active work.



ACR, accessing contents stored in ACR, and modifying information of ACR.

CSM has ACCP Information Manager to collect and manage the information of the whole users, content spaces, and active works in collaboration domain. Collaboration users can access and modify collaborative information easily with the centralized information manager.

User Manager provides User Information Modifier to edit user or group information, and User Authentication Manager to certify user who accesses ACCP.

Active Work Manager provides Active Work Information Modifier, Active Work design toolkit, and Active Work Instance Monitor. User can edit active work information by using Active Work Information Modifier. Also, user can create his unique active work with Active Work design toolkit, and monitor the current running active work instance by using Active Work Instance Monitor.

CSM provides ACR management function, content management function, and content space information modifier function. A collaboration group manager can create or modify a work space with ACR manager, and

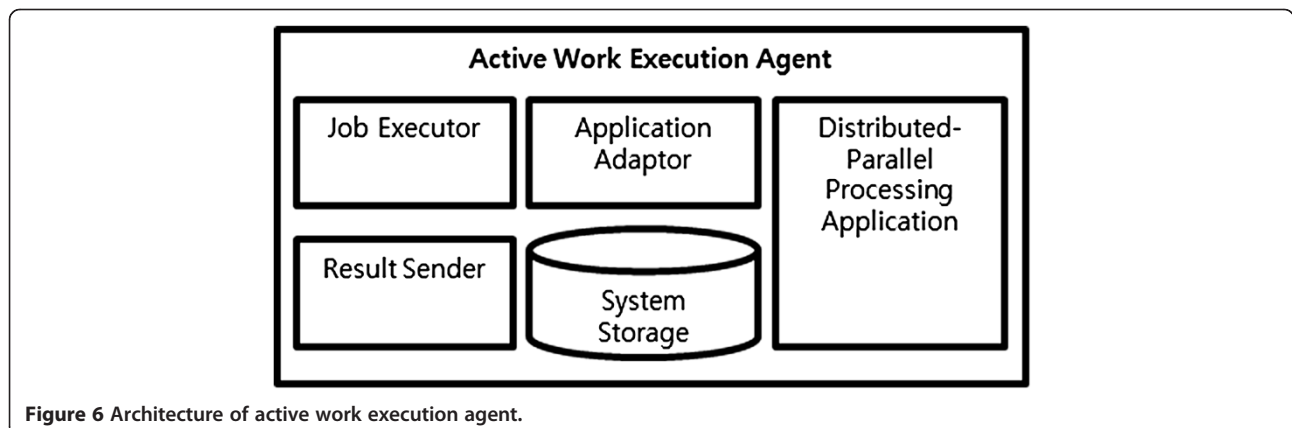
collaboration members can access contents in work space by using content manager.

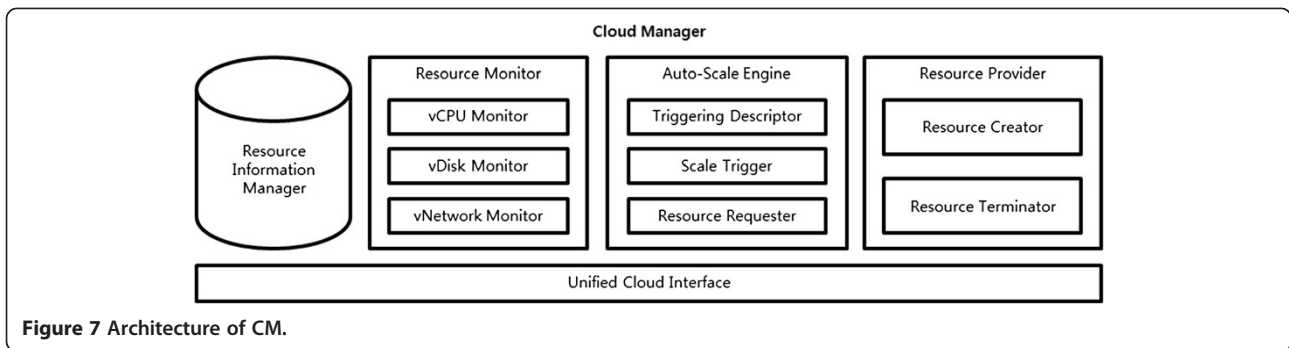
User, Active Work, and CSMs are provided as their corresponding web service through the CSM web user interface: User Management Interface, Active Work Management Interface, and Content Space Management Interface.

### 3.2.2. ACR

ACR is a content repository which provides automated collaboration service called active work with two main services: Repository Manipulation Service and Active Work Service. Figure 3 shows the detailed architecture of ACR.

Repository Manipulation Service stores and manages the content produced during collaboration between users by providing a web service function to access collaborative information and content data. It is composed of two services: Content Foundation Service and Collaboration Foundation Service. The former supports CSM to access content space, while the latter supports CSM to access user and active work information of ACR. ACR has





Content Namespace Manager which provides a scalable distributed file system (DFS).  
 In huge collaboration, static storage may cause lack of available space to store contents. To solve this problem, Content Namespace Manager requests additional virtual machine resources or terminating useless ones to CM Interface, thus modifying the whole size of content space storage.

Active Work Service is composed of three components: Active Work Description Parser, Active Model, and Active Work Runtime System. Active Work Service supports collaboration by using user information and active work

which stored in Content Namespace Manager. Active work is a description of collaboration task flow. It is composed of three kinds of elements: actions, events, and rules. The example of active work is presented in Figure 4.

Active work is described in our own description language called Active Work Description Language (AWDL). When collaborating, user requests to activate stored active work in Content Namespace Manager, Active Work Parser in Active Work Service parses AWDL file, and generates Active Work Instance which is composed of event, rule and action instances. Generated Active Work Instances are managed by Active Work Instance Controller,

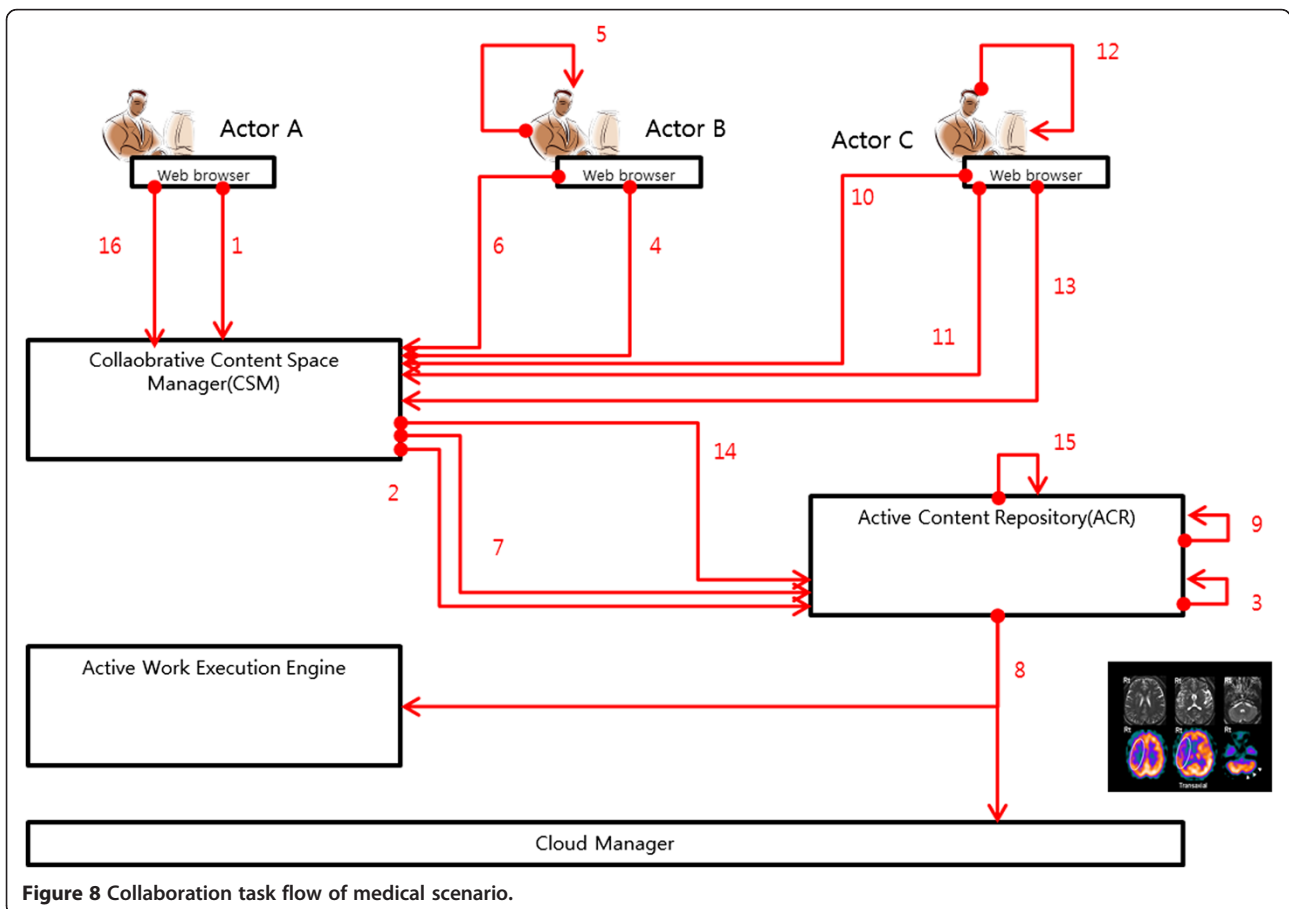


Figure 8 Collaboration task flow of medical scenario.

**Table 1 Each arrow in medical scenario**

1	Deploy active work through CSM (brain analysis with MRI)
2	Deploy active work in ACR (brain analysis with MRI)
3	Low-level action (create message content: "request MRI volume data" to B)
4	View content (message content: "request MRI volume data")
5	Create brain MRI data
6	Upload content through CSM (MRI volume data)
7	Upload content in ACR (MRI volume data)
8	High-level action (execute volume rendering with AWEE)
9	Low-level action (create message content: "request 2D MRI analysis data" to C)
10	View content (message content: "request MRI analysis data")
11	View content (multimedia content: "MRI 2D data")
12	Analyze 2D MRI data and create analysis report
13	Upload content through CSM ("MRI analysis report")
14	Upload content in ACR ("MRI analysis report")
15	Low-level action (create message content: "MRI analysis is done" to A)
16	View content (multimedia content: "MRI analysis report")

and waiting valid events. When any change occurs in Content Namespace Manager, ACR generates event due to the occurred change, and collects them into event collector. After events are collected, event analyzer generates event instance, and event dispatcher sends it to Active Work Instance Controller to advance active work. Event instances work as a trigger of advancing active work. In the example of active work showed in Figure 4, there are three rules: r1, r2, and r3, three events: e1, e2, and e3, and five actions: a1, a2, a3, a4, and a5. r1 is the entry point of active work. If any event stored in event collector is equal to e1, r1 executes a1 and a2 by using Action Handler, and changes entry point to r2. Action Collector in Action Handler collects action instances which are delivered from Active Work Instance Controller. When new action instance is collected, Action Analyzer analyzes it and generates batch process of action

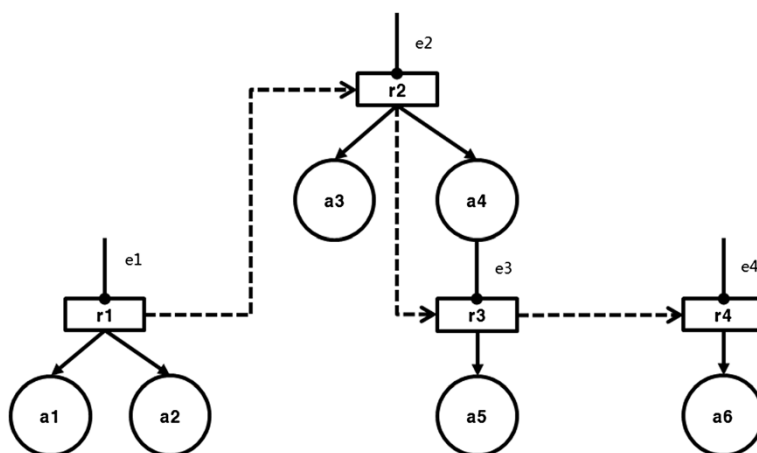
execution. Action Executor runs action due to generated batch process by using Repository Manipulation Service and AWEE. There are two types of action: low-level action and high-level action. Low-level action is a simple task such as messaging to other collaboration user and content modification by using Repository Manipulation Service. High-level action is a task which may not be executed by own functions of ACR. To execute high-level action, Action Executor requests CM to create new virtual machine which executes additional collaboration function for jobs in professional area by using CM Interface.

**3.2.3. AWEE**

AWEE is a management component for providing external application service. Figure 5 shows the architecture of AWEE.

**Table 2 Active work description in medical collaboration scenario**

Active work	Action	a1	Low-level action (message content: "notify startup of active work" to "A", "B", "C")
		a2	Low-level action (message content: "request MRI volume data" to "B")
		a3	Low-level action (message content : "volume data is stored" to "C")
		a4	High-level action (execute volume rendering)
		a5	Low-level action (message content : "volume rendering is done" to "C")
		a6	Low-level action (message content : "MRI analysis done" to "A")
Event		e1	Deploy active work (medical collaboration)
		e2	Upload content (MRI volume data)
		e3	High-level action (execute volume rendering)
		e4	Upload content (MRI analysis data)
Rule		r1	Run a1 and a2 when e1 comes
		r2	Run a3 and a4 when e2 comes
		r3	Run a5 when e3 comes
		r4	Run a6 when e4 comes



**Figure 9** Diagrams of active work in medical scenario collaboration.

AWEE is composed of three components: AWEE Scheduler, Job Handler, and Result Handler. When a new high-level action is sent, Job Receiver receives its corresponding job data from ACR, and stores them into Job Data Storage. AWEE Scheduler handles the stored job data, and request Job Sender to run high-level action in the Active Work Execution Agent which is installed in virtual machine on cloud. Active Work Execution Agent has several components: Job Executor, Application Adaptor, and Result Sender as shown in Figure 6.

Job Executor receives job data from AWEE, and executes them. After job is done, Result Sender returns the results to AWEE. However, there are many kind of application used in collaboration. So, we need application adaptor which enables each application to be executed on cloud system. Result Receiver in Result Handler stores the result data which is sent from virtual machine. Finally, Result Sender returns the result data to ACR with first in first out algorithm.

### 3.2.4. CM

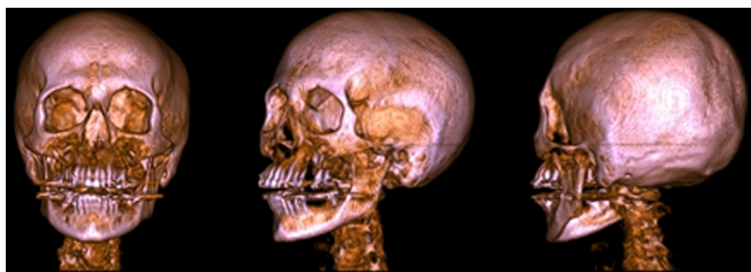
CM is a resource management component which provides all computational resources used in ACCP by using various cloud infrastructure system such as Eucalyptus and

Openstack. It is composed of five components: Resource Information Manager, Resource Monitor, Auto-Scale Engine, Resource Provider, and Unified Cloud Interface. The architecture of CM is shown in Figure 7.

Unified Cloud Interface provides an integrated interface for various infrastructures. It supports the following functions: creating, terminating, modifying, and monitoring virtual machine instance. Resource Monitor collects all the information of virtual machine instances for ACCP through Unified Cloud Interface, and Resource Information Manager stores the collected information. Auto-Scale Engine analyzes resource information stored in Resource Information Manager, decides the creation or termination of virtual machines according to the load of ACCP, and asks Resource Provider to create or terminate virtual machine instances.

## 4. Experiment and evaluation for multimedia processing

ACCP is implemented using Java on eclipse-indigo of Microsoft Windows 7. The web interface of ACCP is designed with full JSP which can be used on several web browsers including mobile web browsers. When initiating ACCP service, CM creates three virtual machines



**Figure 10** 2D image data of volume rendering used in medical scenario.



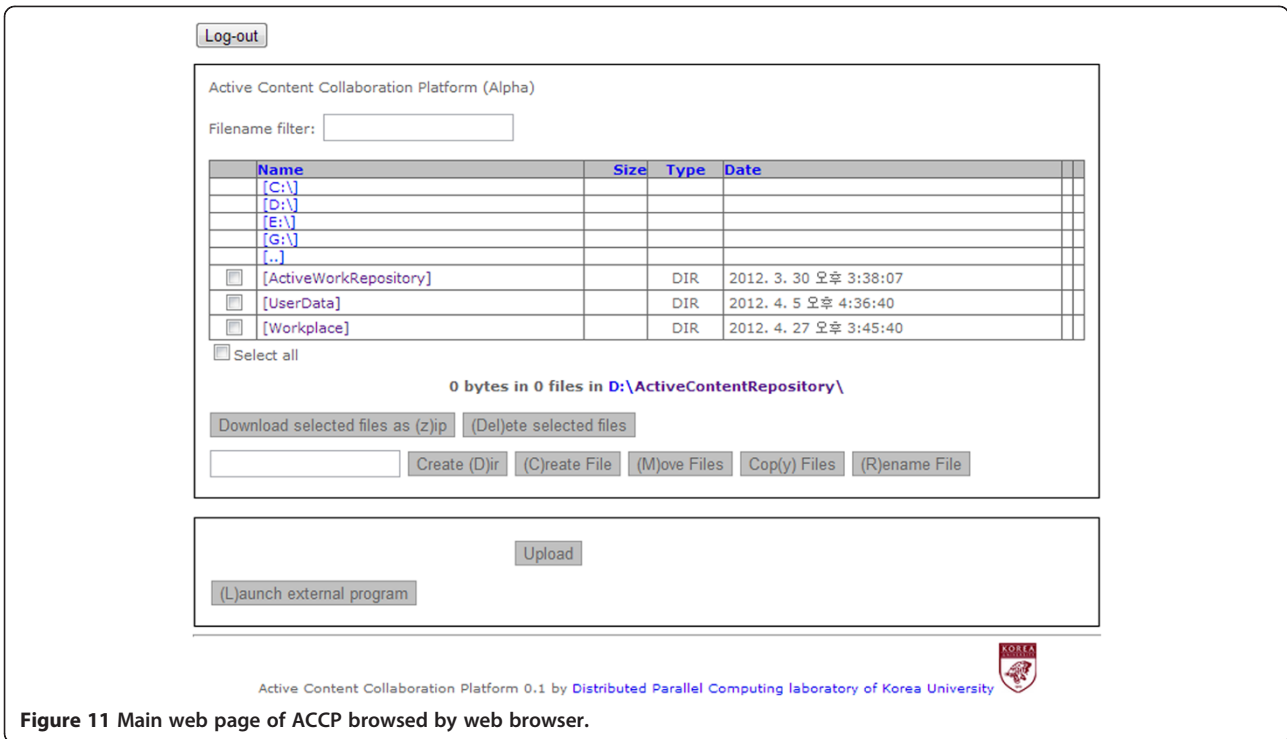


Figure 11 Main web page of ACCP browsed by web browser.

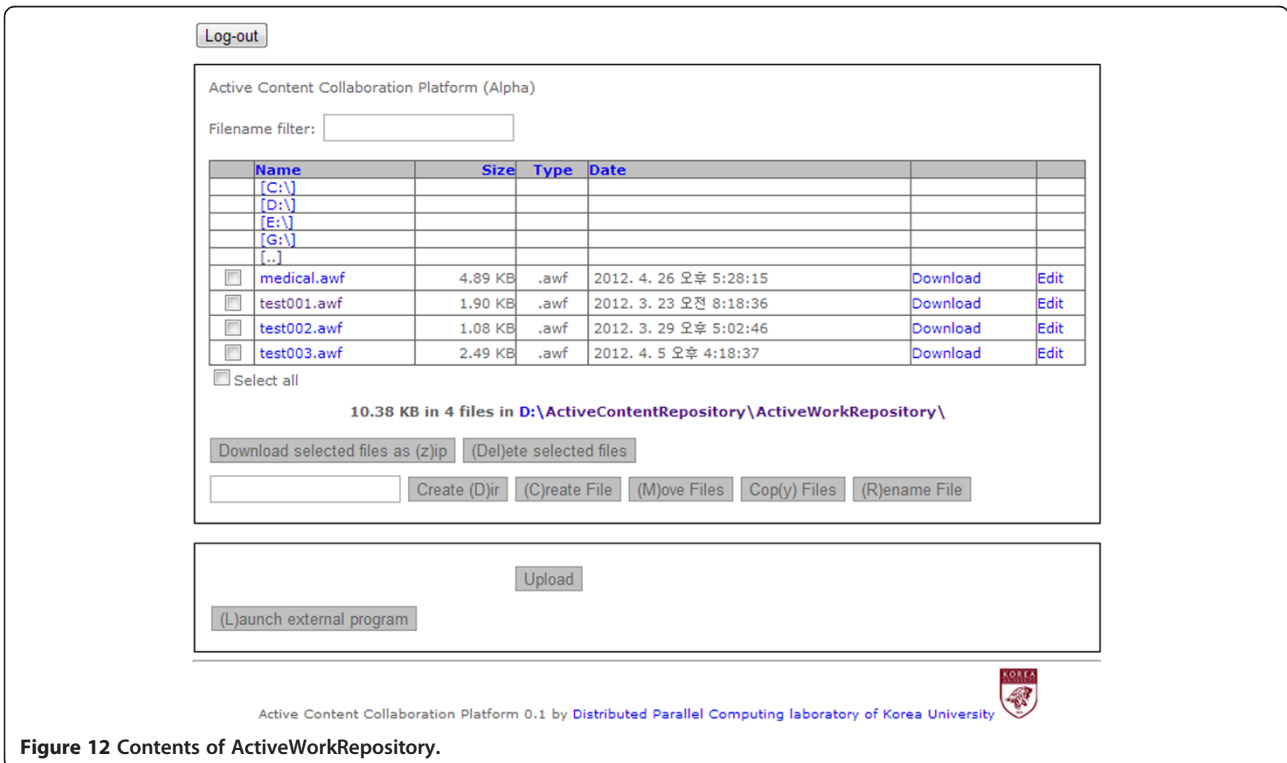
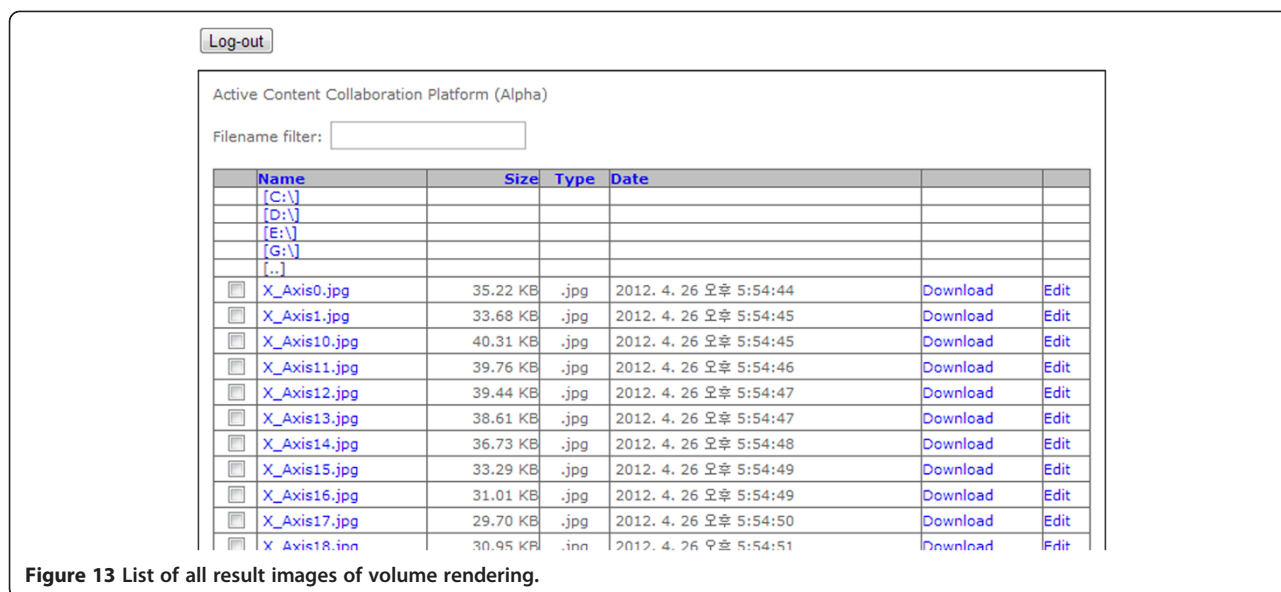


Figure 12 Contents of ActiveWorkRepository.



**Figure 13** List of all result images of volume rendering.

each for CSM, ACR, and AWEE through Unified Cloud Interface. Resource Information Manager stores information of each created virtual machine instances. CSM receives the information about ACR, and ACR about AWEE for high-level action, respectively, from Resource Information Manager. After startup, collaboration users can access ACCP through CSM web user interface. As a multimedia processing application, we shall show a collaborative medical scenario where computation intensive application like a volume rendering is used for MRI analysis.

#### 4.1. Experiment

In medical scenario, three actors collaborate with each other for the analysis of MRI image produced by using volume rendering. Actor A is a family doctor, actor B an MRI doctor, and actor C a brain doctor. As shown in Figure 8, a collaboration flow of medical scenario consists of 16 steps. Each step is described in Table 1 in detail.

In step 1, A deploys an active work for medical collaboration through CSM web user interface. In step 2, CSM stores the information for the active work into ACR as in Table 2.

The active work is composed of six actions (a1, a2, a3, a4, a5, and a6), four events (e1, e2, e3, and e4), and four rules (r1, r2, r3, and r4). Each rule defines the relationship between events and actions which should be executed

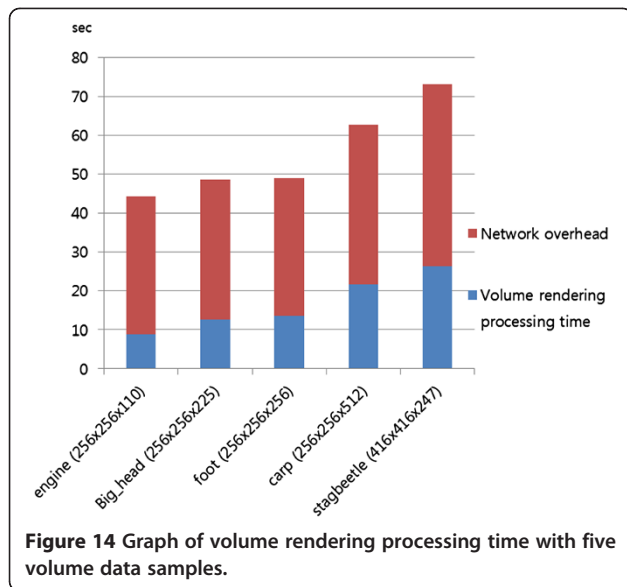
when those events occurs. It can be represented by three diagrams each interconnecting actions, events, and rules as shown in Figure 9.

When active work is deployed correctly, e1 is generated in content space, and a1 and a2 are executed due to r1 in step 3, and each doctor is notified about the startup of active work. In step 4, B accesses CSM to view message from collaboration system. After receiving MRI volume data request message, B creates MRI volume data with their own MRI machine, and uploads it into ACCP through CSM web user interface in steps 5 and 6. In step 7, the volume data are stored in ACR, and e2 is generated. Due to r2, a3 is executed to notify C that the volume data are stored in ACR, and a4 is executed to transform MRI volume data into 2D image in step 8, respectively. In step 8, ACR requests AWEE to execute volume rendering by creating virtual machines through CM. After volume rendering, ACR stores 2D image of volume data, and e3 is generated. Due to r3, a5 is executed to send the message to C that the volume rendering is done in step 9. An example of 2D volume rendering image is shown in Figure 10.

After reading message from a5 in step 10, C views the MRI image in step 11, analyzes it in step 12, and uploads the analysis report into ACR through CSM in step 13 and 14. When the analysis report is stored in ACR, e4 is

**Table 3** System specification of cloud infrastructure system (Eucalyptus)

	CPU	Memory	HDD	Number of machines	Role
Machine type 1	Xeon E5620 2.40 GHz (2EA)	12 GB RAM	1 TB	1	Cloud controller, cluster controller, storage controller, Walrus
Machine type 2	Xeon E5606 2.13 GHz (2EA)	12 GB RAM	1 TB	4	Node controller



generated, and due to r4, a6 is executed to notify A that the MRI analysis is done in step 15. In step 16, A view the MRI analysis report stored in ACR.

After running medical collaboration, we can access contents which are produced in collaboration process. ACCP provides JSP-based web browser which can be accessed by several devices including mobile device. The main web page of ACCP is expressed in Figure 11.

In Figure 11, there are three directories: ActiveWork-Repository stores active work description files. Collaborative users can upload or modify Active Work Description files with this menu. The contents of ActiveWork-Repository are shown in Figure 12. UserData stores all user data such as ID, password, message history, etc. Collaborative users can manage their information by modifying these data. Workplace is a content repository which stores the contents for collaboration. In case of medical collaboration scenario, MRI volume data, 2D

MRI images, and brain analysis report are stored in Workplace directory.

After executing high-level active work using volume rendering, brain doctor can view 2D MRI image data by using web interface in CSM. The list of all result images of volume rendering is expressed in Figure 13.

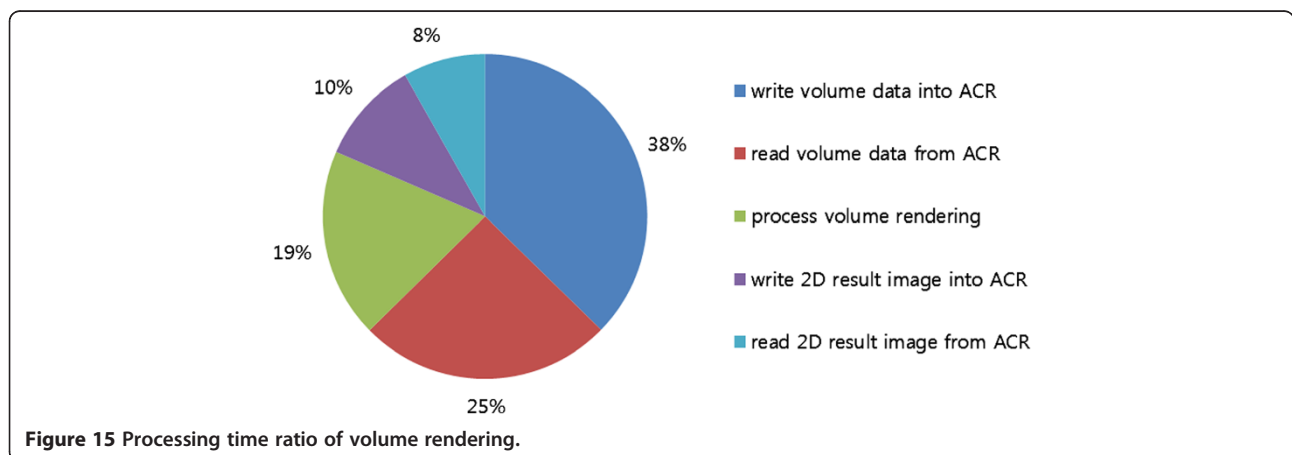
Volume rendering application generates 2D images of skull for all  $x$ - and  $y$ -axis. Brain doctor can view 2D images like Figure 9, and analyze them to edit analysis report.

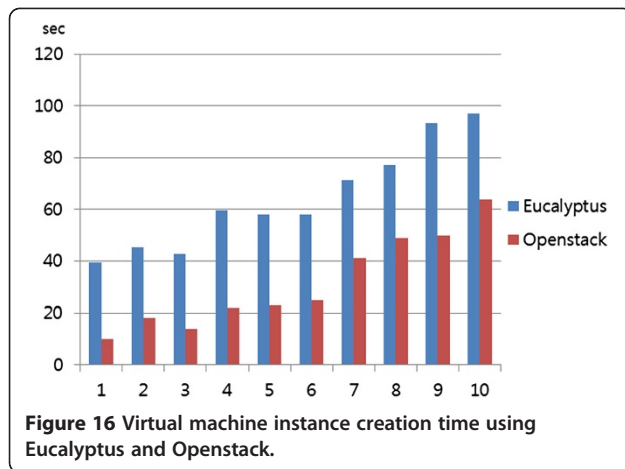
#### 4.2. Evaluation

ACCP can provide computation intensive multimedia applications which support collaboration in professional area by using cloud infrastructure through AWEE. With this external application adaptor, people can collaborate in various areas efficiently. In our scenario, we make use of volume rendering application for 3D multimedia data which is known computation intensive in medical application.

We use a volume rendering engine called “Fast Shear Skew Warp Volume Visualization using Poisson Disk Sample” to support medical collaboration. It provides very fast volume rendering service compared with other previous services. To evaluate performance of volume rendering in our system, we use Eucalyptus version 1.6.2 as a cloud infrastructure system for running ACCP. The system specification is expressed in Table 3.

Each machine uses Ubuntu-10.04-LTS OS and KVM as a hypervisor to run multiple virtual operating systems. With this infrastructure system, we created three virtual machines for CSM, ACR, and AWEE, and five virtual machines in cloud infrastructures. Each virtual machine has 1 CPU, 1536 MB RAM, and 100 GB HDD storage. Content Namespace Manager uses Hadoop DFS to combine distributed storage with one name node, four data nodes. We execute volume rendering for five volume data samples stored by Content Namespace Manager. The processing time of volume rendering for each volume data samples is shown in Figure 14.





The blue bar shows the actual volume rendering processing time when using the volume data stored in ACR directly, and the red bar shows the network overhead taken to transfer data between ACR and virtual machines through AWEE. The blue bar consists of five steps: In step 1, volume data are stored in ACR. In step 2, volume rendering application reads volume data which stored in ACR. After reading volume data, volume rendering application generates 2D images of volume data in step 3. Generated 2D images are stored in ACR in step 4, and finally, 2D images are read for view to collaboration users in step 5. In case of medical scenario, the processing time ratio of each step is shown in Figure 15.

The portion of the volume rendering process is 19%, while the portion of storage I/O overhead using DFS is 81% as shown in Figure 15.

Also, ACCP provides scalable storage in content repository by using cloud infrastructure system. When ACR needs to extend its content repository, it requests new virtual machine instance to CM through CM interface. We make use of two different cloud infrastructures: Eucalyptus and Openstack. Figure 16 shows the time taken for virtual machine creation.

When creating new virtual machine, Eucalyptus uses round-robin scheduler, and Openstack simple filter scheduler, respectively. As shown in Figure 16, Openstack is better than Eucalyptus. For single virtual machine instance, Openstack takes only 10 s while Eucalyptus takes 40 s. For ten virtual machine instances, both of Eucalyptus and Openstack take longer than 1 min. This delay may cause unexpected error in collaboration, since ACR needs to wait for the successful completion of storage extension. To handle this problem, CSM blocks user access while storage extension is on progressing.

## 5. Conclusion

In this article, we have presented a new ACCP which supports automated content-centric collaboration on cloud

system. It supports a modularized and extensible architecture by making CSM, ACR, and AWEE in separate modules. CSM provides a user interface as well as various management functions for collaboration. ACR is a work space for a collaboration group which provides its own Repository Manipulation Service and Active Work Service for collaboration. AWEE connects external application with ACR, and executes various distributed high-performance application for collaboration in professional task area. Finally, CM is a management component for allocating and managing computational resource used to run ACCP. Also, it offers scalable high-performance architecture by supporting multi-level active work processing in AWEE, and allocating VMs for the computation intensive high-level action such as volume rendering through auto-scalable allocation on cloud system. Moreover, our system supports event-driven automatic collaboration by specifying each active work based on AWDL and automating the execution of collaborative task flow composed of active works. For the experiment and evaluation, we have shown the results of implementing the collaborative medical application on our system where computation intensive application like a volume rendering is used for MRI analysis.

### Abbreviations

ACCP: Active Content Collaboration Platform; ACR: Active Content Repository; AWEE: Active Work Execution Engine; CM: Cloud Manager; CMIS: Content Management Interoperability Services; CMS: Content Management System; CPU: Central Processing Unit; CSM: Collaborative Content Space Manager; DFS: Distributed File System; ECM: Enterprise Content Management; HDD: Hard Disk Drive; IaaS: Infrastructure As A Service; jBPM: Java Business Process Management; JSP: Java Servlet Page; JSR: Java Specification Request; KVM: Kernel-based Virtual Machine; LTS: Long Term Support; MRI: Magnetic Resonance Imaging; RIA: Rich Internet Application; SaaS: Software As A Service.

### Competing interests

The authors declare that they have no competing interests.

### Acknowledgement

This research was supported by the MKE (Ministry of Knowledge Economy), Korea, under the ITRC (Information Technology Research Center) support program supervised by the NIPA (National IT Industry Promotion Agency) (NIPA-2012-H0301-12-3006), Next-Generation Information Computing Development Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (2012-0006425), and Ministry of Culture, Sports and Tourism (MCST) and Korea Creative Content Agency (KOCCA) in the Culture Technology (CT) Research & Development Program (R2012030096).

### Author details

<sup>1</sup>Department of Electronic Engineering, Korea University, Seoul, Republic of Korea. <sup>2</sup>Department of Multimedia, Sungkyul University, Anyang, Republic of Korea.

Received: 31 October 2012 Accepted: 31 January 2013

Published: 6 March 2013

### References

1. D. Sturdy, Enterprise content management. *Legal Information Management*. 7(160-164) (2007). doi:10.1017/S1472669607001521
2. B. David, J.S. Ericson, R. Michael, *Toward cloud based collaboration services*, in *Proceedings of the 2009 Conference on Hot Topics in Cloud Computing*

3. *CMS*. [http://en.wikipedia.org/wiki/Content\\_management\\_system](http://en.wikipedia.org/wiki/Content_management_system)
4. *Microsoft Sharepoint*. <http://sharepoint.microsoft.com/>
5. *EMC Documentum*. <http://www.emc.com/domains/documentum/index.htm>
6. *Alfresco*. <http://www.alfresco.com/>
7. C. Stephane, D. Renaud, C. Tim, P. Axel, D. Stefan, Produce and consume linked data with Drupal. The Semantic Web - ISWC 2009. Lecture Notes in Computer Science **5823**, 763–778 (2009)
8. *Joomla*. <http://www.joomla.org/>
9. M.K. Smith, M. Barton, M. Bass, M. Branschofsky, G. McClellan, D. Stuve, R. Tansley, H. Walker Julie, D. Space, An Open Source Dynamic Digital Repository (Corporation for National Research Initiatives. D-Lib Magazine January 2003 **9**(1). ISSN 1082–9873
10. T. Roy, *Felding, JSR-170 Overview: Standardizing the content Repository Interface, March 13, 2005* (Day Management AG white paper)
11. *CMS*. [http://en.wikipedia.org/wiki/Content\\_Management\\_Interoperability\\_Services](http://en.wikipedia.org/wiki/Content_Management_Interoperability_Services)
12. *jBPM*. <http://www.jboss.org/jbpm>
13. S. Edwin, B. Guillaume, R. James, *Fractal Conceptual Prototype: Active Behaviors (HP Laboratories, HPL)* (Hewlett-Packard Development Company white paper, 2009), p. 66
14. *SaaS*. [http://en.wikipedia.org/wiki/Software\\_as\\_a\\_service](http://en.wikipedia.org/wiki/Software_as_a_service)
15. *RIA*. [http://en.wikipedia.org/wiki/Rich\\_Internet\\_application](http://en.wikipedia.org/wiki/Rich_Internet_application)
16. *Google wave*. <http://mashable.com/2009/05/28/google-wave-guide/>
17. *Zoho*. <http://www.zoho.com/>
18. *Zimbra*. <http://www.zimbra.com/>
19. B.-T. Oded, B. Jonathan, C. Bryan, D. Michelle, S. Gary, SONART, *The Sonification Application Research Toolbox, Proceedings of the 2002 International Conference on Auditory Display* (, Kyoto, Japan, 2003)
20. T. Kar-Han, I.N. Robinson, C. Bruce, A. John, ConnectBoard, *Enabling Genuine Eye Contact and Accurate Gaze in Remote Collaboration (HP Laboratories, HPL-2011 -83). Multimedia and Expo (ICME)* (2010 IEEE International Conference, Suntec City, 1923 –July 2010), pp. 430–435p. ISSN 19457871,– ISBN 978-1-4244-7491-2
21. R. Klamma, G. Toubekis, Y. Cao, D. Renzel, M. Jarke, M. Jansen, *Virtual Campfire—Cultural Heritage Management and Presentation on Mobile Devices based on Interoperable Cross Platform MPEG-7 Multimedia Web Services*

doi:10.1186/1687-1499-2013-63

**Cite this article as:** Han et al.: Cloud-based active content collaboration platform using multimedia processing. *EURASIP Journal on Wireless Communications and Networking* 2013 **2013**:63.

**Submit your manuscript to a SpringerOpen<sup>®</sup> journal and benefit from:**

- Convenient online submission
- Rigorous peer review
- Immediate publication on acceptance
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

---

Submit your next manuscript at ► [springeropen.com](http://springeropen.com)

---