University of Huddersfield Repository

Wei, Guo and Joan, Lu

Investigation into Mobile Learning Framework in Cloud Computing Platform

Original Citation

This version is available at http://eprints.hud.ac.uk/21809/

http://eprints.hud.ac.uk/

Investigation into Mobile Learning Framework in Cloud Computing Platform

Wei Guo
School of Computing and Engineering
University of Huddersfield
Huddersfield, West Yorkshire, HD1 3DH, UK
wei.guo@hud.ac.uk

Joan Lu
School of Computing and Engineering
University of Huddersfield
Huddersfield, West Yorkshire, HD1 3DH, UK
j.lu@hud.ac.uk

Abstract—**Cloud computing infrastructure is increasingly used for distributed applications. Mobile learning applications deployed in the cloud are a new research direction. The applications require specific development approaches for effective and reliable communication. This paper proposes an interdisciplinary approach for design and development of mobile applications in the cloud. The approach includes front service toolkit and backend service toolkit. The front service toolkit packages data and sends it to a backend deployed in a cloud computing platform. The backend service toolkit manages rules and workflow, and then transmits required results to the front service toolkit. To further show feasibility of the approach, the paper introduces a case study and shows its performance.**

Keywords: Cloud computing, mobile devices, service set

I. INTRODUCTION

The increasing demands for efficient resources utilisation result in the use of cloud computing. M. Armbrust et al have proposed that the adoption of virtualised resources brings high scalability and availability to applications deployed in clouds [1]. Cloud computing can offer scalable storage and resources expansibility [2, 3].

A great number of relative mobile application techniques integrated in clouds have been proposed by [4, 5, 6, 7, 16]. D. Minifie and Y. Coady have proposed that mobile devices can access cloud computing platforms using the Web to improve transit accessibility [4]. W. Rochadel et al have described education applications of remote experimentation for mobile devices [5]. P. Pocatilu et al have depicted communication between mobile applications and cloud storage services [6]. D. Popa. et al have proposed mobile cloud applications and traceability [7]. A. Butoi et al have described cloud-based mobile learning applications [16]. However, they only focus on certain aspects of design and implementation of mobile applications in cloud computing platforms. Currently few publications have taken cloud-based system-level framework and generic service framework into consideration. Therefore, this paper proposes a cloud-based system-level framework including generic front service and backend service frameworks.

In this paper, a new approach for mobile devices deployed in the cloud is advocated, which consists of two ends. 1) One is the front service toolkit, used mainly for receiving messages from end users and sending processing messages to the backend service toolkit. If the backend service toolkit returns the proposed results according to the business mode, it will show corresponding information on the user interface. 2) The other one is the backend service toolkit, which is responsible for executing business

data flow and rules; and it is also to store the necessary information into databases.

This paper is organised as follows. Section 2 describes related researches. Section 3 proposes frameworks of the front and backend service toolkits. Section 4 outlines a case study implementing the above frameworks. Section 5 presents experiments of the case in Section 4 and discusses the results. Section 6 sketches the conclusions.

II. BACKGROUND

Mobile application techniques in the cloud are proposed by [4, 5, 6, 7]. Designing and development of cloud based M-Learning tools are introduced by [16]. P. Pocatilu proposes a framework for syncing mobile applications with cloud storage services [6]. However, this has not considered security between mobile devices and the cloud computing platform. D. Popa et al have proposed how data security, data privacy and trust are described [7]. Y. Gu et al have designed services and components for transmitting files from mobile devices to the cloud to trade-off between performance and battery life [10]. J. Dong Lee and J. Park have introduced system layer, application layer and user layer for a mobile cloud learning system [11]. A rendering adaptation technique is proposed to enable multimedia applications on rich mobile devices [12]. The utilisation of a component-based approach for the framework results in shading the implementation details of sophisticated functionalities when it is running [13, 14, 15].

Z. Meng and J. Lu have proposed the implementation of the emerging mobile technologies in facilitating a mobile exam system [17]. They have proposed opportunities for interactive learning systems with evolutions in mobile devices [18]. Integration of smartphone's intelligent techniques for authentication in a mobile exam login process is introduced [19]. A. Butoi et al have introduced cloud-based mobile learning applications including software design and quality metrics [16].

The relative researches mainly focus on usage of computing and storage ability of cloud or mobile communication. In this paper, architecture of front and backend toolkits are system-level perspective and built based modules, such that availability and extension can be improved.

Reasons for design:
- A system-level framework is proposed for mobile applications with the cloud so as to improve software availability and expansibility.
- Generic front service and backend service frameworks are to reduce application development costs.
- The generic front and backend service framework make applications in cloud easy to maintain.
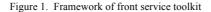
III. ARCHITECTURE FRAMEWORK

In this section, the frameworks of the front service toolkit and backend service toolkit are proposed. The front service toolkit and backend service toolkit can work individually and only performs the cooperation with each other when the user requires.

A. The Framework of Backend Service Toolkit

In this section, architecture of the front service toolkit is proposed. The architecture consists of variable packager, array packager, XML packager, variable analyser, array analyser and XML analyser as demonstrated in Figure 1, which will be discussed in this section.
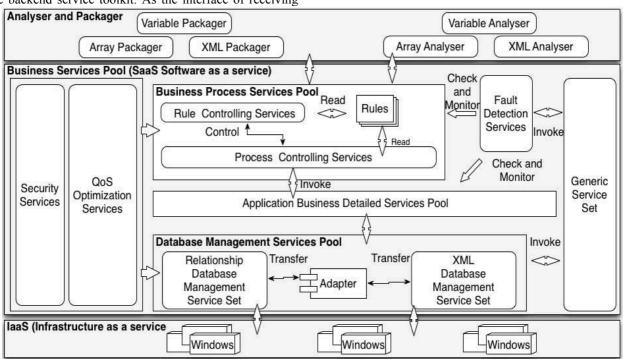

Figure 1. Framework of front service toolkit

Packager: All of the packagers are utilised for package information and data from users and then the packaged messages are transmitted to backend services in the cloud. In particular, the XML packager has the responsibility for organizing data according to XML schema defined in advance.

Analyser: Analysers consists of variable analyser, array analyser and XML analyser. They are corresponding to the above packagers separately, which are applied to extract information and data from packages routed from the backend service toolkit. As the interface of receiving messages, they are required to be identical to the packagers.

B. The Framework of Backend Service Toolkit

In this section, architecture of the backend service toolkit is proposed. The architecture is mainly composed of analyser and packager, business services pool, and IaaS (Infrastructure as a Service) as demonstrated in Figure 2, which will be discussed in this section.

Figure 2. Framework of backend service toolkit

Analyser and packager: Analysers and packagers of the backend service toolkit have the same main functions as the front service toolkit. The front service toolkit uses these functions to trigger the transmission of data and messages to backend.

Application business detailed service pool: It is used for containing and deploying components and services of applications. Note that mobile applications are divided into functions as web services in the cloud. The cloud computing platform publishes web services through a Web server. Web services implementing application business have hierarchical relationships. Similar to Java SDK, web services in this layer can invoke each other and, due to specific user requirements, can work together to complete application functions.

Business process services pool: This layer is a web service container which is integrated with the rule engine and workflow engine. They offer unique standard web service interfaces to external systems and mobile devices. Workflow engine organise the relationships among web

services and defines main work flow and message flow. And hence, the rule engine can deal with decisions of work flow in terms of application requirements. In particular, when there is more than one work flow in applications, the rule engine is considered to handle the flow of information and messages.

Database management services pool: This applies mapping techniques to map objects from the above layers into records in database. Here, two categories, namely, relationship database and XML database, are utilised. The backend program makes use of database management service sets to store data into tables and to retrieve record from tables. The backend service toolkit employs multiply databases to store data so as to restore storage if the main database crashes.

Fault detection services: This has the responsibility for detection and fault tolerance of faults of the front and backend service toolkits. The backend service toolkit includes service backup functions, which can back up central services and failure-prone services.

Generic service set: Common and generic services are abstracted and placed in this layer. This layer provides services to developers and other service sets. Developers can exploit libraries of services in this set to develop other applications.

Security services and QoS optimisation services: Developers and end users can define and put their security services into this module. The security module and QoS optimsation module work together to confirm that applications run and meet user requirements.

C. Fault Tolerance for Backend Service Toolkit

In this section, a fault tolerance scheme for backend service toolkit is proposed. The fault tolerance level of services of backend toolkit is defined and specified by developers to perform the fault-tolerant execution.

1) Fault Tolerance Parameters Packaging

The front service toolkit is not only responsible for packaging data from users and sending the message to backend service toolkit, but adds a fault-tolerant identifier into the metadata of request messages.

Fault tolerance parameters definition: The front service sends the request sequence, $Request = \langle service\_id, R, \text{P} \rangle$, to the backend service toolkit. Here, $R = \{c, n, s, l\}$ is a fault-tolerant parameters set, where c, n, s and l denote computing, network, storage resources and fault tolerance level respectively. For example, $Request = \langle service\_i, \{c_i, n_i, s_i, l_i\} \rangle$ represents front service $service\_i$ which needs c_i computing, n_i network and s_i storage resources respectively. P is a request parameters set, which is a business request from specified functions.

For a mobile application, whose backend is based on cloud platform, the whole created fault tolerance metadata is $Request = \langle app\_services, \{\sum c, \sum n, \sum s, I\} \rangle$. To save resources and transmission cost, the fault tolerance parameters list is sent when the front service toolkit finds change of parameters of services.

2) Replica Virtual Machine Selection

The fault detection service of the backend service toolkit has responsibility for selecting a replica virtual machine according to the received fault tolerance parameters from the front service toolkit. At first, the fault tolerant service needs to collect the domain knowledge of cloud infrastructure and to generate its network topology. Then, the fault detection service selects the replica placement nodes to back up the original services. In case, failures of services occur, the fault selection redirects requests from users to replicas of services.

a) Cloud Network Topology Construction

Figure 3 lists cloud fault tolerance components, which can collect computing, network and storage resources information from virtual machines.
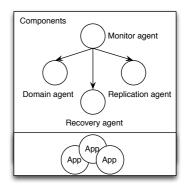

Figure 3. Cloud fault tolerance components

Domain agent: The domain agent aims to execute as a domain knowledge collection management. Resources information collection management may handle constraints on selecting duplication placement nodes to create network topology.

Monitor agent: The monitor agent is responsible for monitoring all its own agents in the cloud computing infrastructure (e.g., domain agent, replication agent and recovery agent) by means of a time-out mechanism. The responsibility of the main monitor agent is divided into three categories: the status validation of other agents in the identical component; coordination with other monitor agents; replication and recovery invocation for replicating and recovering agents and components.

Recovery agent: The recovery agent completes the goal of recovering the last checkpoint state from failures. When failures are detected, the monitor agent collects the abnormal failure node information from stable network storage and triggers the recovery mechanism.

Replication agent: The replication agent fulfills the target of replicating applications and files in virtual machines to the available calculated node with fault tolerance degree, in the presence of the client's requirements.

b) Matching Operation with the Requirements from Front Services

As mentioned earlier, the fault-tolerant and recovery component can receive the requirements from front services through the analyser and packager. Hence, the component allows front services to specify fault tolerance

properties. For the fault tolerance level, it is based on the virtual machine level in this paper. It mainly depends on the computing, network and storage resources. Here the virtual machine with high computing, network and storage resources is considered as a node having higher fault-tolerant capabilities.

IV. A CASE STUDY

The section shows an application, called iPlayCode, which is deployed in a cloud computing platform. iPlayCode is a mobile-based application and its backend applies web service techniques. Its game rules and workflow are stored in the backend in the cloud. All the operations of iPlayCode to database are packaged as web services.

TABLE I. CENTRAL SERVICES OF IPLAYCODE IN CLOUD

Index	Service Name	Function Descriptions
1	GameShuffleService	The web service is to disrupt the order of the question list. In this case students cannot remember answers to questions.
2	GameMarkService	The service is used to mark students' responses and store it into the database.
3	GetQuestionsService	The service gets question contents and sends them to mobile devices.
4	GameCounterService	The service is to record information from user accessing mobile applications.
5	GameMathService	The service is generic to make developers invoke many times only if they use the backend service toolkit.
6	SetAnswersService	The service stores users' answers into database.
7	SetCommentsService	Users can submit comments on iPlayCode to help improve it.

Table 1 lists the central services of the backend service toolkit of iPlayCode and their function descriptions. In particular, the service, GameShuffleService, is used to disrupt the order of questions. The reason for its usage is that the order of questions in database is fixed. To make students unable to remembering answers, iPlayCode needs to change the sequences of retrieved questions. All the operations of changing order of questions run on the backed, which can hide detailed performance to avoid students' deducing the details. In addition, the service, GameMathService, is a generic maths package. Note that, although user interfaces and functionalities of applications differ, their basic functionalities are identical. The generic service set of the backend service toolkit includes a basic development service pool. These services are packaged to offer the standard interfaces to developers.
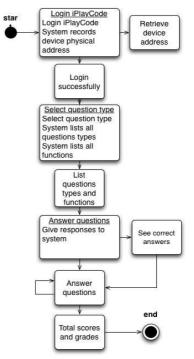

Figure 4. Business flow of iPlayCode

Figure 4 lists the business flow of iPlayCode. Specifically, end users need to login to iPlayCode typing their user name. The system would receive the physical address of physical devices and send both address and user name to the backend service toolkit. After backend services store this identity information into the database, they would send back the success message to the front applications on mobile devices. Then end users can choose a question level to select questions. After that, the system gives a response of question lists to mobile phones. After users finish questions, they can submit answers to backend services and services can store submitted answers.

Figure 5 introduces the network topology of the cloud computing environment which iPlayCode uses. From the graph, cloud controller 1 manages 6 virtual machines. These virtual machines deploy all the services of the backend service toolkit. Cloud controller 1 can forward the invocation from users to services and messages are returned to mobile devices through it.
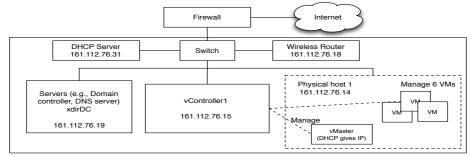

Figure 5. Network topology of cloud computing environment

V. Experiment

The section shows actual deployment experiment of iPlayCode. First, conditions and system setup are proposed to describe the basic cloud infrastructure. Second, applications are introduced that are used in the experiment. Finally, data analysis is demonstrated.

A. Conditions and System Setup

Citrix XenServer 6.0.2 is employed as the cloud infrastructure. The virtual machines are listed in Table 2. To deploy services of iPlayCode, hybrid operation systems and physical hosts are employed. All the implementation of iPlayCode is based on modular and component approaches.

TABLE II PARAMETERS OF CLOUD ENVIRONMENTS

	Host A	Host B	Host C	Host D	Host E	Host F
IP	161.112.1.22	161.112.1.23	161.112.1.24	161.112.1.25	161.112.1.26	161.112.1.27
OS	Windows 2008 R2	Windows XP	Windows 7	Windows 7	Windows XP	Windows 7
CPU	Intel P-IV 2.0G	Intel P-IV 2.0G	Intel P-IV 2.0G	Intel P-IV 2.0G	Intel P-IV 2.0G	Intel P-IV 2.0G
RAM	2048 MB	1024 MB	1024 MB	1024 MB	1024 MB	1024 MB
HDD	350G	500G	500G	500G	500G	500G

B. Testing of the System

1) Testing Strategy

The following list describes test mobile equipment, test cloud environment and test items. Experiment includes two parts. One is the mobile application. Here iPhone, iPad and Android phones are used to test. The other one is the cloud environment test. Citrix XenServer 6.02 as the cloud infrastructure is utilised to offer virtual computing, network and storage resources to backend services.

- Test mobile equipment: Apple iPhone, iPad and Android phones.
- Test cloud environment: Wi-Fi and Citrix XenServer 6.0.2.
- Test items: interface, functionality and communication time.

2) Test Results

TABLE III TEST RESULTS

Test Items	Mobile Platforms	Apple iOS 7.0 (iPhone 5s)	Android 4.2 (SAMSUNG)
Interface	Information display	√	√
	Element position	√	√
	Layout of page	√	√
Functionality	Logining System	√	√
	Selecting question level	√	√
	Answering questions	√	√
	Automatic marking	√	√
	Storing marks	√	√
	Storing answers	√	√
Communication time	Logining couner	√	√
	Getting question speed	√	√
	Storing marks	√	√
	Storing answers	√	√

(Note: √ denotes correct or acceptable)

C. Data of Experiment

Figure 6 and Figure 7 illustrate execution time of services of iPlaycode deployed on the web server and in the cloud separately. As shown from these figures, services, GameMarkService, GetQuestionsService and SetAsnwersService occupy the maximum running time. Execution time of services in the cloud is longer than on the web server. Although services need more time to respond to mobile devices, they can offer application availability, reliability and expansibility.
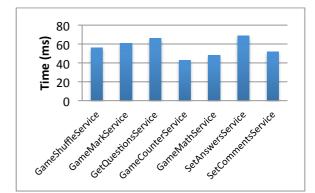

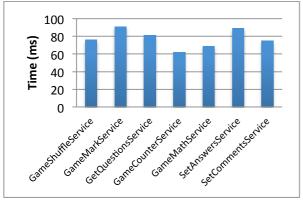Figure 6. Execution time of services deployed on traditional Web server

Figure 7. Execution time of services deployed in cloud

VI. CONCLUSION AND FUTURE WORK

A. Conclusion

This paper proposes design and development of mobile applications in the cloud. It includes front service toolkit and backend service toolkit. At the end of the paper, a case study, iPlayCode is proposed, which is deployed on Citrix XenServer 6.0.2. Execution time of services in the cloud is compared with that on a traditional web server. This investigation contributes to the following points:

- The system-level framework can improve availability and expansibility of mobile learning applications.
- The framework makes software easy to maintain.
- The toolkits reduce application development costs at the expense of some communication time.

B. Future Work

The future studies will include issues listed as follows:

- Enhance the functionality of the system according to user preference.
- Enhance analysis of the responses from students for learning performance evaluations.
- Shorten the response time between the front and the backend service toolkit.

REFERENCE

[1] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. H. Katz, A. Konwinski, G. Lee, D. A. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, "Above the clouds: A Berkeley view of cloud computing," EECS Dept., Univ. California, Berkeley, UCB/EECS-2009-28, Feb. 2009.

[2] Amazon Elastic Compute Cloud [Online]. Available: http://aws.amazon.com/ec2/

[3] Eucalyptus Systems [Online]. Available: http://www.eucalyptus.com/

[4] D. Minifie and Y. Coady, "Getting mobile with mobile devices: using the web to improve transit accessibility," W4A '09 Proceedings of the 2009 International Cross-Disciplinary Conference on Web Accessibililty (W4A), Sep 2009, pp. 123-126

[5] R., W, S. Simão, J.P. , B. Silva, J. , D. Alves, G.R. "Educational application of remote experimentation for mobile devices," Remote Engineering and Virtual Instrumentation (REV), 2013 10th International Conference on, 6-8 Feb. 2013, pp. 1 – 6

[6] P. Pocatilu, C. Boja, C. Ciurea, "Syncing Mobile Applications with Cloud Storage Services," Informatica Economică vol. 17, 2/ 2013, pp. 96-108

[7] D. Popa, K. Boudaoud, M. Borda, M. Cremene, "Mobile cloud applications and traceability," Networking in Education and Research, 2013 RoEduNet International Conference 12th Edition, 26-28 Sept. 2013, pp. 1-4

[8] M. Armbrust, A. Fox, R. Griffith, et al. A view of cloud computing. Communications of the ACM. 2010, 53(4): 50-58.

[9] R. Guerraoui and M. Yabandeh, "Independent faults in the cloud," in Proc. 4th Int. Workshop Large Scale Distributed Syst, Middleware, no. 6. 2010, pp. 12-17.

[10] Y. Gu, V. March and B. Lee, "GMoCA: Green Mobile Cloud Applications," GREENS, page 15-20. IEEE, (2012)

[11] J. Lee, J. Park*, "Application for Mobile Cloud Learning," 16th International Conference on Network-Based Information Systems, 4/2013, pp. 296-299

[12] S. Wang and S. Dey, "Adaptive Mobile Cloud Computing to Enable Rich Mobile Multimedia Applications," IEEE TRANSACTIONS ON MULTIMEDIA, VOL. 15, NO. 4, JUNE 2013,pp. 870-883

[13] M. Hiltunen and R. Schlichting, "An approach to constructing modular fault-tolerant protocols," in Proc. 12th Symp. Reliable Distributed Syst., 1993, pp. 105-114.

[14] G. Jung, K. M. Sim. Agent-based adaptive resource allocation on the cloud computing environment. The 40th International Conference on Parallel Processing Workshops(ICPPW 2011). IEEE Computer Society, 2011: 345 - 351.

[15] R. Jhawar and M. Santambrogio, "Fault tolerance management in cloud computing: a system-level perspective," IEEE. Systems Journal. July. 2013, pp. 288-297.

[16] A. Butoi, N. Tomai, L. Mocean, "Cloud-Based Mobile Learning," Informatica Economică vol. 17, no, 2/2013, pp. 27-41.

[17] Z. Meng, J. Lu, "Implementing the Emerging Mobile Technologies in Facilitating Mobile Exam System," 2nd International Conference on Networking and Information Technology, IPCSIT 25th-26th November 2011, 17 . IACSIT Press, Hong Kong, China, pp. 80-88. ISBN 978-981-07-0680-7.

[18] Z. Meng and J. Lu, "Opportunities of Interactive Learning Systems with Evolutions in Mobile Devices: A Case Study," In: Proceedings of the 2011 International Conference on Internet Computing ICOMP 2011. CSREA Press, pp. 238-244. ISBN 1601321864

[19] Z. Meng and J. Lu, and A. Sawsaa, "Integrating Smartphone's Intelligent Techniques on Authentication in Mobile Exam Login Process," In: Computational Collective Intelligence. Technologies and Applications: 5th International Conference, ICCCI 2013, Craiova, Romania, September 11-13, 2013, Proceedings. Lecture Notes in Computer Science (8083). Springer, London, pp. 130-142. ISBN 978-3-642-40494-8