Exploring the Performance of Authentication Mechanism in Integrated Cloud Services Environment with CSP

Dong Wang[1], Mingquan Zhou[1,2], Sajid Ali[3], Pengbo Zhou[2], and Guohua Geng[1]
[1] School of Information Science and Technology, Northwest University, Xi'an 710127, P. R. China
[2] College of Information Science and Technology, Beijing Normal University, Beijing 100875, P. R. China
[3] Department of Computer Science, University of Education, Lahore 60000, Pakistan
Email: wangdongxidian@gmail.com; {mqzhou_nwu, snfa_bnu}@yahoo.com; houpengbo@bnu.edu.cn; ghgeng@nwu.edu.cn

Abstract —Most of the current cloud computing platforms are integrated massive cloud services, which aim to provision abundant computing and storing services to end users. Web portal works as a manager of a large volume of cloud services and end users in cloud computing environment. Nevertheless, a major hurdle of formal adoption of web portal for cloud services is performance that is not always investigated in the past, and we believe there is still a lot of room to improve performance in web portal without affecting the integrated cloud services environment. This paper proposes a cloud service integration architecture with Cloud Service Portal (CSP) and Cloud Service Bus (CSB) in hybrid cloud. We present a four-layer CSP architecture and define a run-time data model with some operators in CSP. Three run-time data storage patterns including file pattern, database pattern and memory pattern are given by CSP to explore its performance. We improve an authentication mechanism from OAuth 2.0 and use to our integration architecture. Finally, we conclude that memory storage pattern is more cost effective and has fast response time as compare to others two patterns in CSP login and authority services. It means, the proposed architecture has good performance to be used in cloud service authentication.

Index Terms—Cloud service, cloud service portal, authentication mechanism, portal performance

I. Introduction

Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to share computing resources that can be rapidly provisioned and released with minimal management effort [1]. Enterprises are planning to, or have already started to integrate applications into their enterprises' information systems due to the extremely compelling cost-saving potential for cloud based deployments [2]. It is an absolutely necessary technology in today's competitive market. Many enterprises' information systems make use of cloud services. Meanwhile, various cloud computing service providers are available with their supporting services in the cloud environment. Cloud services are becoming more and more popular, both for enterprises aiming to outsource parts of their IT activities to third-party data centers or Cloud platforms, as well as for the end users [3]. To provide computation, storage and network capacity for cloud services, multiple geographically separated servers or server clusters interconnected by a physical network constitute the cloud infrastructure (data centers) [4]. Data centers providing the cloud computing services are increasing economically and variably. Because of these trends, the efficient operation of data centers is becoming more and more important.

According to the flexibility and efficiency of cloud computing in network provisioning, it has attracted great attention from the industry and academics. Cloud computing services and applications are commercialized and delivered in a manner similar to traditional resources such as water, electricity, gas, and telephony [5]. An increasing number of cloud services like Google Drive (a file storage and synchronization service), Dropbox (a file hosting service), Evernote (a closed source freemium suite of software and services), etc. are used by millions of users on the desktops and mobile devices. Although cloud computing services and data center are growing and gaining popularity, the authentication mechanism for the integrated cloud services is still a hot issue.

OAuth 2.0 [6] is an open authorization protocol specification defined by IETF OAuth Working Group which enables applications to access each other's data. It is a protocol enabling a client application, often a web application, to act on behalf of a user, but with the user's permission [7]. Based on OAuth 2.0, some authentication methods were presented in cloud service authorization. Rabea Kurdi and Martin Randles [8] proposed a security authentication with access permissions in an E-government Web Portal Application. The portal is designed to allow different members to access the portal by separating the roles of members to improve the flexibility of the system for administration. End users' access time permissions are hold by portal server. Sawesi al. [9] introduced a secured authentication framework and XML-based authentication method to be employed by the web-based ECommerce portals. They designed a security framework for authentication using XML digital

Manuscript received November 11, 2015; revised April 14, 2016.
This work was supported by research on key technology of virtual restoration mosaic in terracotta army 20136101110019, and Research on the method of virtual restoration of damaged Terracotta Army based on global optimization 61373117".
Corresponding author email: mqzhou_nwu@yahoo.com.
doi: 10.12720/jcm.11.4.388-395

signature in conjunction with PKI standard through elliptic curve in B2C cloud based on E-Commerce portal. But the performance of XML-based authentication method in E-Commerce portal is not discussed.

Traditionally, most of the cloud services integration solutions build a portal for services management and end users access control [10]-[12]. Leslie Liu [13] has presented a cloud service portal to provide remote management access to virtualized device management servers hosted in a service cloud. It is acting as an integrated point to bring together functions and services on the cloud, and presents a device-and-role-aware view to the incoming user. ChienMing Tu al. [14] have proposed a Cloud Management Portal (CMP) in their cloud system's prototype that is called CHT Cloud Orchestration (CHTCO). It is a single-entry web portal that provides customers and system administrators with complete lifecycle management of virtual machine and integration of resources monitoring.

The above mentioned cloud portals have been recently introduced. However, they don't care about their performance. While performance is a key issue for each software architecture, previous cloud portals are not absolutely suitable for the actual complex network environment without performance.

Previous works have not been specially address the need of authentication mechanism when massive users concurrent access a same portal. Our design is targeted to build a lightweight, modular and extensible cloud service integration architecture in hybrid environment, in which a potent authentication mechanism can be deployed. And we proposed a Cloud Service Portal (CSP) that manages a large number of cloud services and end users in the integration architecture. We have defined three run-time data storage patterns to investigate CSP performance in different storage patterns. They are as (1) File pattern (CSP run-time data is stored in disk), (2) Database pattern (CSP run-time data is stored in database) and (3) Memory pattern (CSP run-time data is stored in memory). In this study, we improve OAuth 2.0 for high performance authentication mechanism by CSP in cloud service environment. And we have explored the different performance of three CSP storage patterns in our framework.

The remainder of this paper is organized as follows. Section II presents the technology and methods of our goal. In Section III, we introduce our experiments and results, which show login and authority services response time in the proposed authentication mechanism. Then a conclusion and future work is presented in Section IV.

II. Material and Methods

To design, implement and evaluate the cloud service integration architecture to a Cloud Service Portal (CSP) and a Cloud Service Bus (CSB) (extended from enterprise service bus [13], [14]), this section gives the methodological approach.

A. Cloud Service Integration Architecture

To provide a comprehensive management and high performance authentication mechanism to the end users in hybrid cloud environment, we have designed a cloud service integration architecture which evolved from integrated system architecture introduced in [15]-[17]. Fig. 1 illustrates the overall system architecture. CSP is presented to be a popular solution for cloud services integration [18]. In this paper, we will employ CSP as a cloud manager of cloud services and end users. It provides two key features to our architecture: cloud services authentication and end user management. We adopt CSB to compatibly support the cloud services and easily combine the primitive mechanisms of cloud software as user authentication, user management, data delivery and etc [19]. And a data center is built to support increasing computational and data storage demand of growing cloud services. It stores all information for cloud services and end users to be used in the architecture [8]. It collects data from every cloud service and shares them to CSP.

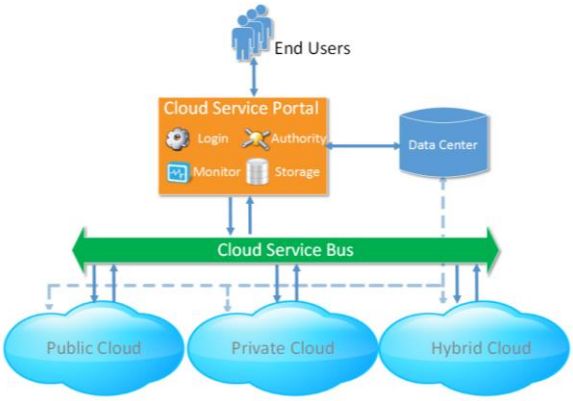


Fig. 1. Cloud service integration framework

CSP is a system module which provides access control, end user data management, and authentication mechanism. It includes four main functions: login, authority, monitor and storage. Users can use an interface that is provided by CSP to access cloud services of multiple cloud service providers. We will give the clearly description to CSP in next subsection.

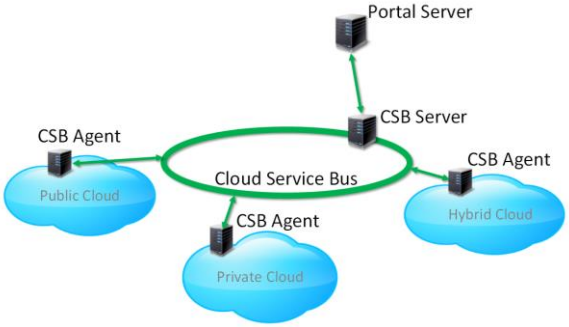


Fig. 2. Cloud service bus moudle

We use CSB module that serves the architecture as a bridge to communicate with CSP and multiply cloud

services. It works for data center which is connected to CSP. CSB is consisted by CSB server and CSB agents. CSB server is corresponding to CSB agent deployed within local or remote cloud platform [19]. The CSB agent provides the system architecture with the capacity to integrate a new cloud platform without modification. An example of the CSB module is shown in Fig. 2.

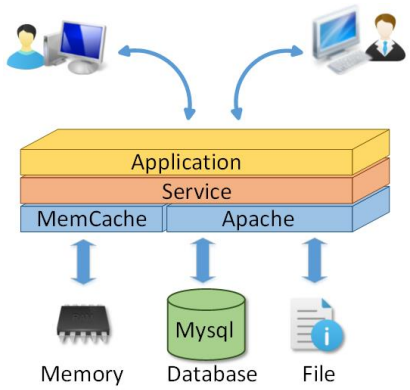


Fig. 3. Four-Layer cloud service portal architecture

B. Cloud Service Portal

CSP is a core component in the architecture (see Fig. 1) as it is connected to all cloud services through CSB and served as a gateway for users. End users have permission to use their subscribed cloud services after entering CSP. However, our goal is to analyze the contribution and performance of CSP. As a consequence we present a four-layer CSP architecture, see Fig. 3. The four layers are described as follow:

- The first layer is represented by the web application which is used for user accesses the portal through a web browser;
- The second layer consists of some services communicating with local web applications and remote cloud services;
- The third layer is represented by system threads where the portal services are deployed. The threads includes Apache and Memcache;
- The fourth layer is a data storage layer contains internal and external storage. Internal storage is memory, and the external storage is database and file. After user login, run-time data is stored in this layer.

TABLE I: CSP SERVICES DESCRIPTION

Service name	Service description
L=Login	Login service is used for end user logins CSP.
A=Authority	Authority service provides authentication mechanism for cloud services.
S=Storage	Storage service gives CSP three run-time data storage patterns.
M=Monitor	Monitor service manages end user run-time data.

In the second layer, CSP services include login, authority, storage and monitor. The description of CSP services is shown in Table I. The services are developed by PHP and deployed in Apache.

In order to storage and operate data in CSP, we provide a data storage layer in the portal architecture and define run-time data model and operators. They are elaborated as following.

Definition 1 (Run-time data model). End user run-time data model is represented as {Token, A, T} where Token is the unique identification used to present end user login CSP and mark the data model. A is the set of end user attributes and T is the timestamp that the end user last time has action in CSP.

The operators are defined and they can manage run-time data. We represent an end user run-time data as RD.

Definition 2 (Generate). CSP generates a RD when an end user enters it. In RD, Token is a random value, A is collected from cloud data center, and T is current time.

Definition 3 (Search). CSP searches RD by access Token.

Definition 4 (Modify). CSP modifies T in RD.

Definition 5 (Check). CSP checks the validity of RD.

Definition 6 (Remove). CSP removes RD by Token.

Definition 7 (fetch). CSP verifies end user validity and gets user information in data center.

The detail description of services in the second layer is shown in Fig. 4. And the figure also shows the services access data center and run-time data by the six data operators.

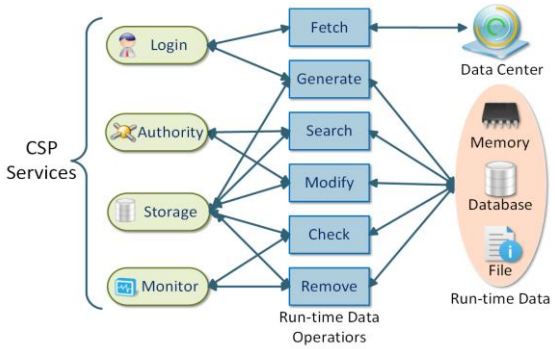


Fig. 4. CSP services

The CSP is deployed in 'Apache + Php + Mysql' development framework, which is similar to the Quran portal in [11]. Apache is the world's most popular server software providing a secure, efficient, extensible server and HTTP services in sync with the current HTTP standards [20]. To extend the capacity of Apache to access memory, we add Memcache in the third layer of the architecture. Memcache is used to improve the poor performance of conventional storage mechanism [21].

C. Authentication Mechanism

A key technical underpinning of the Cloud is authentication mechanism. Authentication provides consistent run-time data management and protection for outside end users to access services in cloud. Our work is based on OAuth 2.0, and the security authentication with access permissions in [16], [22], [23]. We focus on the internal run-time data management and authentication interface of CSP to improve its efficiency. According to

the proposed cloud service integration architecture, we present a forceful two-step authentication mechanism. The abstract flow illustrated in Fig. 5 describes an overview involved in proposed authentication mechanism considering a generic implicit grant scenario. Login step and authority step are the two responsible sub steps of authentication mechanism. Login step includes step 1 to 8. Authority step is the rest steps.

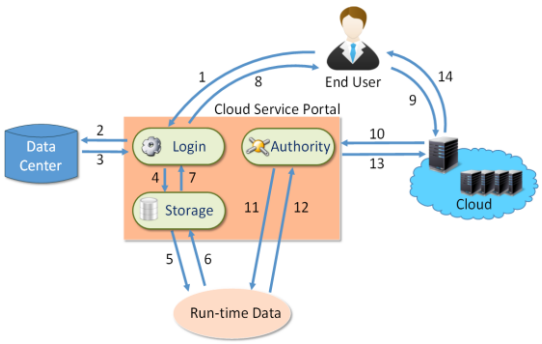


Fig. 5. Representation of step wise authentication mechanism

The login step follows the sub steps as:

1) End user requests to enter CSP with name and password. The communication channel is secured with SSL.

2) Login service authenticates the end user with his name and password in data center.

3) Data center gives a set of end user attributes back to login service.

4) Login service generates a token and sends it with user attributes to storage service.

5) Storage service checks CSP data storage patterns and saves user attributes in run-time data identified with the unique access token.

6) User data is stored in run-time data successfully.

7) Storage service sends operation successful message to login service.

8) Login service generates a personalized interactive page by access token and gives it back to end user.

The authority step follows the sub steps as:

9) End user requests cloud service with his access token.

10) Cloud service extracts the token and sends it to CSP authority service.

11) Authority service gets the token, calls search operator to get user's run-time data. After that it calls modify operator to update the user's run-time data time to current time.

12) Authority service receives user's run-time data.

13) Authority service responds cloud service with user's run-time data.

14) Cloud service extracts user's run-time data, checks user permission, and generates a personalized interactive page for end user.

Finite State Machine (FSM) is a well-known mathematical model for designing a method with a finite number of states. The role of the action status as a state machine in the various states, transfer conditions defined for the state machine used to change the status value of the input and output conditions, and ultimately by the user to define rule-based state machine system [24], [25]. According to the user state model in SOAE [16], our authentication mechanism using a user state transition automaton to illustrate the steps in detail. Fig. 6 shows the diagram of the state machine for proposed authentication mechanism.

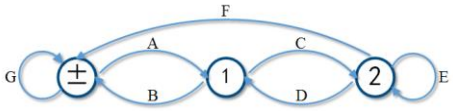


Fig. 6. User state transition automaton

We assume that EIP is connected with a cloud service, and the end user has the permission to enter the cloud service from CSP. So we define the start state by "-" and the end state by "+". The state of user login CSP is "1", the state "2" means user enters cloud service. The actions of end user in Fig. 6 are described in Table II.

TABLE II: ACTIONS IN USER STATE TRANSITION AUTOMATON

No.	Label	Description
1	A	End user enters the cloud platform in CSP
2	B	End user exits the cloud platform from CSP
3	C	End user accesses cloud service from CSP
4	D	End user accesses CSP from cloud service
5	E	End user does operation in cloud service
6	F	End user exits the cloud platform from cloud service
7	G	End user enters the cloud platform unsuccessfully

Some previous work used RESTful APIs to develop authentication function of portal without performance analysis [6], [7], [10]. We also tried RESTful interface in our research, but the performance is low. So we built the authority service to accept http request of cloud services in our CSP architecture. In next section, our experimental results will show the proposed CSP architecture has good performance, especially when massive cloud services concurrent request for authentication.

D. CSP Algorithms

According to the entire workflow of CSP services, CSP algorithms include login, authority, monitor and storage. Fig. 7 illustrates the relationship of CSP algorithms. They are designed as the implementation of CSP services to improve CSP performance.

In CSP algorithms, we define some symbols such as DC is cloud data center. R_File is file in run-time data. R_Db is database in run-time data. R_Mem is memory in run-time data. INFO_USER is a set of end user information queried from data center. Some functions are elaborated here. Is_Empty is used to check whether a data set is empty or not. Format is a function used to make a data set of user to fixed run-time data structure. Is_Register is a function used to check whether the cloud service is registered by an identification string. Search and remove are the operators of run-time data.

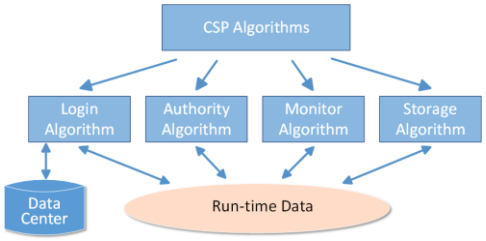


Fig. 7. CSP algorithms

Algorithm 1 describes the login service workflow.

Algorithm 1: login algorithm.

Input: End user name N_{USER} and password P_{USER}.

Output: Run-time data RD, personalized interactive page of the end user $PAGE_{USER}$.

It has the following steps:

1) N_{USER} → Login, P_{USER} → Login
2) INFO_USER = query N_{USER} and P_{USER} in DC
3) If Is_Empty(INFO_USER) != NULL
4) Begin
5) Token = rand()
6) RD = { Token, Format(INFO_USER), current time }
7) RD → R_File or R_Db or R_Mem
8) Generate $PAGE_{USER}$ by RD
9) End

Algorithm 2 is used to get the run-time data of an end user from CSP authority service.

Algorithm 2: authority algorithm.

Input: End user's Token is used for access cloud service, IDENTIFYservice is an identification of cloud service.

Output: Run-time data RD.

This algorithm has the following steps

1) Token→ Authority, IDENTIFYservice → Authority
2) If Is_Register(IDENTIFYservice) == TRUE
3) Begin
4) data = Search Token in RD
5) If Is_Empty(data) != NULL
6) Inner loop begin
7) data→ Cloud service
8) Update data.T = Now in RD
9) End inner loop
10) End

Algorithm 3 describes the monitor service workflow.

Algorithm 3: monitor algorithm.

Input: Limit time of run-time data ΔT.

Output: Run-time data RD.

This algorithm also has the following steps:

1) For each data in RD
2) Begin
3) If (Now - data.T) < ΔT
4) Inner loop begin
5) remove(data)
6) End inner loop
7) End

III. Experimental Results

In this section we present testing activities and resulting measures used to estimate authentication mechanism performance with CSP and integration architecture. Since our aim is to build a high-performance authentication mechanism in cloud service environment. These tests are therefore to be seen as a qualitative validation of the authentication mechanism as an enabling technology. We explored the different performances of CSP by using three run-time data storage patterns by numbers of end users concurrent use a cloud service. As the authentication mechanism has two sub steps, we separate the tests to login part and authority part.

Tests are executed on six different machines: AMD FX-4300 CPU (3.8GHz) desktop computer with 4GB of DDR3 RAM. One computer is used as CSP server and two are used as cloud service providers. Others are installed HP LoadRunner 11 to generate virtual users in tests [26], [27]. LoadRunner computers run test scripts and create numbers of virtual users for authentication. The experiment virtual network environment is shown in Fig. 8.

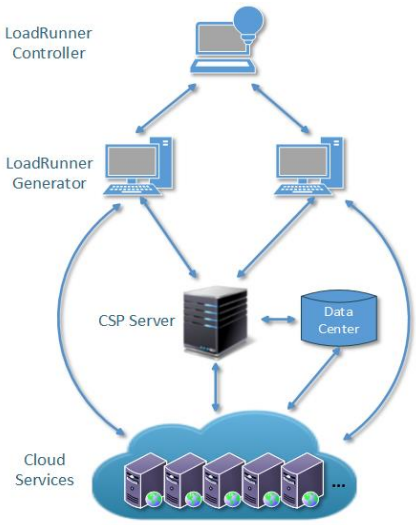


Fig. 8. An example of virtual network environment for Experiment

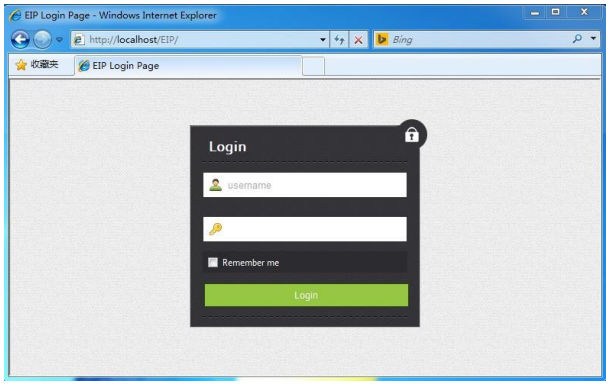


Fig. 9. User login page in CSP.

We used Apache2.2.21, Php5.3.10, and Mysql5.5.20 to build a simplest cloud service integration architecture which has a CSP and five cloud services. The CSP is

based on the architecture proposed in subsection II-B and supported the three run-time data storage patterns. The cloud services support proposed authentication mechanism to verify end user's access token and get his run-time data from CSP. User login page in CSP is shown in Fig. 9. User portal page in CSP is shown in Fig. 10.

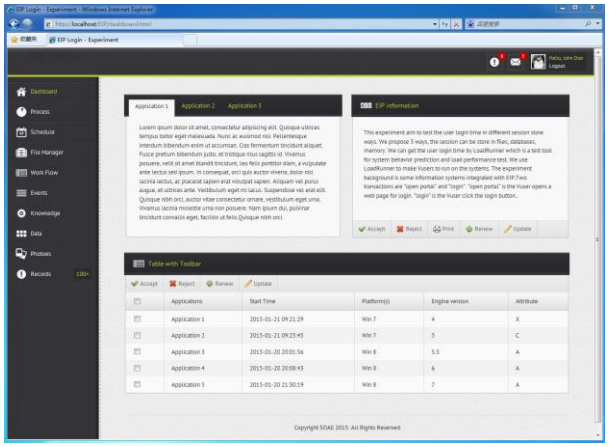


Fig. 10. User portal page in CSP

We met some difficulties during our experiments. In the first instance, we developed CSP application by CodeIgniter2.1.3, which is a popular PHP framework. But it cannot support the concurrent access for more than 800 users. Therefore, we made the CSP application by PHP code without framework.

Moreover, we tried to make an API of CSP authority service as web service by using nusoap-0.9.5, which is a PHP package for building web service. The web service cannot support the concurrent access user number for more than 700. Authors in [6], [7], [10] used RESTful APIs to develop authentication function. We also tried a rest-server package of CodeIgniter to make RESTful authority web service, unfortunately the max concurrent access user number is 500. So we used HTTP message to transmit the run-time data of user from CSP server to cloud service.

Finally, the Apache server, which runs by the default configuration file, cannot support more than 500 users for concurrent access. We reset two configuration files of Apache in Windows 7. One is httpd.conf, we deleted the "#" of the line "Include conf/extra/httpd-mpm.conf". Another is httpd-mpm.conf, we changed the configuration value in WinNT MPM, set the parameter "ThreadsPerChild" with 700 and "MaxRequestsPerChild" with 10000.

We edited two LoadRunner scripts for tests. (1) End user concurrent enters CSP (Login service is triggered). (2) End user concurrent accesses cloud service by access token (Authority service is triggered). In each part of the experiments, we tested the three run-time data storage patterns at least five times. Average time cost in login service and authority service is shown in Table III, IV.

Comparing our authentication mechanism with a user permission verification method based on EIP [28], the maximal number of user concurrent accessing is 500, however, we increase the maximal number to 4000 in our tests. The performance of our authentication mechanism is better than the verification method [28], as our authentication mechanism supports more end users for concurrently login and access system with the same time cost. The following experimental results show the performance of the proposed authentication mechanism with its three run-time data storage patterns.

Table III shows CSP login service test results by concurrent access of 1000, 2000, 3000, 4000 virtual users. The CSP login service response time is calculated by LoadRunner for the whole users. Fig. 11 shows how the CSP response time curve varies with the increase of the concurrent access users in each run-time data storage patterns.

TABLE III: CONCURRENT ACCESS TIME COST IN CSP LOGIN SERVICE

User	Memory	Database	File
1000	4.415s	5.537s	6.776s
2000	10.221s	13.853s	15.478s
3000	19.245s	20.356s	23.894s
4000	24.697s	27.74s	35.831s

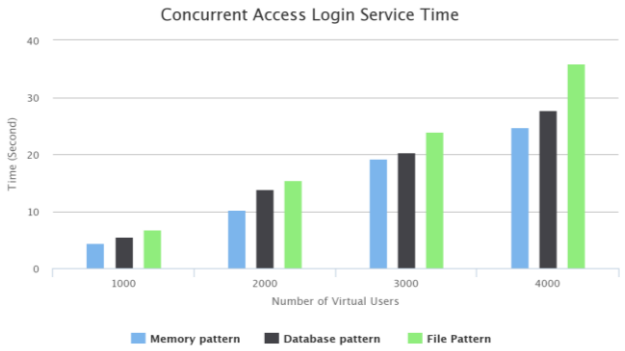


Fig. 11. Concurrent login time

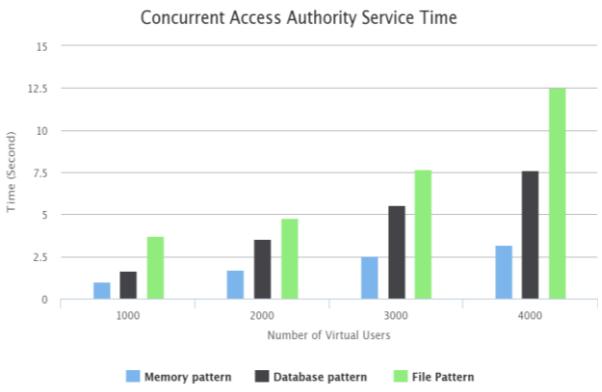


Fig. 12. Concurrent access time

In the authority service test, we used 1000, 2000, 3000, 4000 users for concurrent access cloud services. It would trigger the proposed authentication mechanism. Table. IV shows the test results. As shown in Fig. 12, CSP three storage patterns' performance is linear increased with the growing number of users. As expected, memory storage pattern leads to better performance, as less reading and writing data time is required. Moreover, by comparing the three curves, it comes evident that the proposed authentication mechanism with CSP run-time data memory storage pattern has better performance than other

two patterns. This result strongly supports the feasibility of proposed integration architecture and authentication mechanism.

TABLE IV: CONCURRENT ACCESS TIME COST IN CSP AUTHORITY SERVICE

User	Memory	Database	File
1000	0.993s	1.63s	3.692s
2000	1.686s	3.522s	4.798s
3000	2.52s	5.571s	7.667s
4000	3.187s	7.614s	12.502s

IV. CONCLUSIONS AND FUTURE WORK

In this paper, we addressed the challenge of reducing heavy processing pressure of cloud service portal to improve performance of authentication mechanism. The proposed integration architecture with CSP is simple, extensible and most importantly available to use. Under this assumption, we show an improved high-performance authentication mechanism. The experiments show that proposed approach is able to support 4000 users concurrent access login and authority service.

The proposed architecture is used in a cloud environment which has 40,000 users. According to the statistical data of a month in this platform, the max user number in concurrent access login and authority service is 2,000. Therefore the number of the concurrent access user is 5% of the total number of users. As a result, the proposed approach possesses sustainability and feasibility of high-performance to support 80,000 users in real time cloud service environment.

The architecture that we proposed reduces the pressure of the CSP. However, it is undeniable that there are still some limitations in our work. Such as, concurrent access of 10000 users or more will reduce the performance of CSP. The bottleneck of the system performance is still on the CSP server. For this reason we need to optimize it with multiple CSP in the future work.

ACKNOWLEDGMENT

This work was supported by the "Research on key technology of virtual restoration mosaic in Terracotta Army (20136101110019)", "Research on the method of virtual restoration of damaged Terracotta Army based on global optimization (61373117)" and "Web services monitoring technology in distributed network environment based on CEP (YZZ14119)".

REFERENCES

[1] Y. Choi, S. Lee, J. Kim, *et al.*, "The method to secure scalability and high density in cloud data-center," *Information Systems*, vol. 48, pp. 274–278, March 2015.

[2] C. Tang and J. Liu, "Selecting a trusted cloud service provider for your SaaS program," *Computers and Security*, vol. 50, pp. 60-73, May 2015.

[3] P. Casas and R. Schatz, "Quality of experience in cloud services: Survey and measurements," *Computer Networks*, vol. 68, no. 11, pp. 149–165, 2014.

[4] D. Liao, G. Sun, V. Anand, and H. Yu, "Survivable provisioning for multicast service oriented virtual network requests in cloud-based data centers," *Optical Switching and Networking*, vol. 14, no. 8, pp. 260–273, 2014.

[5] R. Buyya, C. S. Yeo, S. Venugopal, J. Broberg, and I. Brandic, "Cloud computing and emerging IT platforms: vision, hype, and reality for delivering computing as the 5th utility," *Future Generation Computer Systems*, vol. 25, no. 6, pp. 599–616, 2009.

[6] Securing RESTful Web Services Using Spring and OAuth 2.0 [Online]. Available: http://www.hsc.com/Portals/0/Uploads/Articles/WP_Securing_RESTful_WebServices_OAuth2635406646412464000.pdf

[7] Securing RESTful Web Services with OAuth2. [Online]. Available: http://blog.cloudfoundry.org/2012/10/09/securing-restful-web-services-with-oauth2/

[8] R. Kurdi and M. Randles, "Proposed design for an e-government web portal application using cloud computing," in *Proc. Sixth International Conference on Developments in eSystems Engineering (DeSE)*, 2013, pp. 317-322.

[9] K. G. A. Sawesi, M. M. Saudi, and M. Z. Jali, "Designing a new e-commerce authentication framework for a cloud-based environment," in *Proc. IEEE 4th Control and System Graduate Research Colloquium*, 2013, pp. 53-58.

[10] J. Yang, R Anand, S. Hobson, J. Lee, Y. Wang, and J. M. Xu, "Data service portal for application integration in cloud computing," in *Proc. 8th International Conference & Expo on Emerging Technologies for a Smarter World*, 2011, pp. 1-4.

[11] Z. A. Adhoni and H. Al Hamad, "An API for quran portal using drupal technology," in *Proc. Fifth Applications of Digital Information and Web Technologies*, 2014, pp. 160-164.

[12] Y. Wu and X. Cheng, "The design and implementation of special market orientated integrated e-business portal system," in *Proc. International Conference on E-Business and E-Government*, 2010, pp. 145-148.

[13] J. Maa, H. Yub, and J. Guo, "Research and implement on application integration based on the apache synapse ESB platform," in *Proc. AASRI Conference on Computational Intelligence and Bioinformatics*, 2012, pp. 82–86.

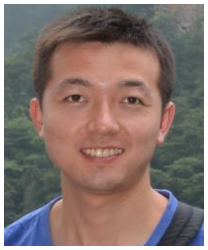
[14] J. Ryan, "Rethinking the ESB: Building a secure bus with an SOA gateway," *Network Security*, vol. 2012, no. 12, pp. 14–17, 2012.

[15] L. Liu, R. Moulic, and D. Shea, "Cloud service portal for mobile device management," presented at IEEE 7th International Conference on e-Business Engineering (ICEBE), 474-478, 2010.

[16] D. Wang, M. Zhou, and S. Ali al, "Exploring the user response time of login application based on SOAE," presented at The 12th IEEE International Conference on Advanced and Trusted Computing (ATC 2015), Beijing, 2015, pp. 378-385.

[17] S. Yan, B. S. Lee, G. Zhao, D. Ma, and P. Mohamed, "Infrastructure management of hybrid cloud for enterprise

users," in *Proc. 5th International DMTF Academic Alliance Workshop on Systems and Virtualization Management*, 2011, pp. 1-6.
[18] C. Tu, S. Ku, J. Tseng, H. Kao, F. Lu, and F. Lai, "CHT cloud orchestration: An integrated cloud system of virtualization platform," in *Proc. 16th Asia-Pacific Network Operations and Management Symposium*, 2014, pp. 1-6.
[19] A. Sun, J. Zhou, T. Ji, and Q. Yue, "CSB: Cloud service bus based public saas platform for small and median enterprises," in *Proc. International Conference on Cloud and Service Computing*, 2011, pp. 309-314.
[20] Apache. [Online]. Available: http://httpd.apache.org/
[21] J. Yang, W. Ping, L. Liu, and Q. Hu, "Memcache and MongoDB Based GIS Web Service," in *Proc. Second International Conference on Cloud and Green Computing*, 2012, pp. 126-129.
[22] M. Noureddine and R. Bashroush, "A provisioning model towards oauth 2.0 performance optimization," in *Proc. 10th IEEE International Conference on Cybernetic Intelligent Systems*, 2011, pp. 76-80.
[23] F. Yang and S. Manoharan, "A security analysis of the OAuth protocol," in *Proc. IEEE Pacific Rim Conference on Communications, Computers and Signal Processing*, 2013, pp. 271-276.
[24] S. Wu, "The implement of animation state transitions in interactive scenes based on graphic finite state machine," in *Proc. 4th International Conference on Intelligent Human-Machine Systems and Cybernetics*, 2012, pp. 242-245.
[25] D. H. Kim, J. Kwon, K. Chen, and K. Choi, "Finite state machine for vehicle detection in highway surveillance systems", in *Proc. 19th Korea-Japan Joint Workshop on Frontiers of Computer Vision*, 2013, pp. 84-87.
[26] P. Li, D. Shi, and J. Li, "Performance test and bottle analysis based on scientific research management platform," in *Proc. 10th International Computer Conference on Wavelet Active Media Technology and Information Processing*, 2013, pp. 218-221.
[27] X. Yan, F. Wen, C. Fan, and X. Wang, "Performance testing of open laboratory management system based on load runner," in *Proc. International Conference on Instrumentation, Measurement, Computer, Communication and Control*, 2011, pp. 164-167.
[28] D. Wang, M. Zhou, and S. Ali al., "Analyze login and certification time based on SOA with EIP architecture," presented at The 12th IEEE International Conference on Advanced and Trusted Computing (ATC 2015), Beijing, 2015, pp. 626-629.



Dong Wang was born in Shaanxi Province, China, in 1987. He is a Ph.D. student at the School of Information and Technology, Northwest University, China. As a part of his Ph.D. research, he is currently exploring solutions for information systems integration and web portals. His main research interests are in distributed systems, and more specifically in the area of complex event processing. He is the leader of an innovative talent training project in Northwest University.

Mingquan Zhou is a professor and doctoral supervisor at the College of Information Science and Technology, Beijing Normal University and director of Key Laboratory Engineering Research Center of Virtual Reality and Application, Ministry of Education, China. His research interests are information processing, computer graphics and 3D visualization.

Sajid Ali received the Postdoctoral & Ph.D. degrees from the College of Information Science and Technology, Key Laboratory Engineering Research Center of Virtual Reality and Application, Ministry of Education, Beijing, China. Beijing Normal University, China in 2013 and 2015 respectively and the MS (CS) and MSc (CS) degrees from Department of Computer Science, the University of Agriculture, Faisalabad, Pakistan in 2003 and 2005, respectively. He is a faculty member of University of Education, Lahore. His current research interests include sensor motion, Biometrics Technology, 3D-human motion, animation, and computer network.

Pengbo Zhou, is a Ph.D. student at the College of Information Science and Technology, Beijing Normal University and Beijing Key Laboratory of Digital Preservation and Virtual Reality for Cultural Heritage. His interests are in intelligent information processing, cultural heritage protection, and 3-D model processing.

Guohua Geng, is a professor and doctoral supervisor at the School of Information and Technology, Northwest University, China. Her research interests include intelligent information processing, database and knowledge base; she is the author and co-author of about 80 papers. She was the leader and main investigator of several polish and national projects addressing picture processing, three-dimensional model analysis and processing, cultural heritage protection and restoration, and intelligent information system.