2010 11th ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing

CloudVO: Building a Secure Virtual Organization for Multiple Clouds Collaboration

Jianxin Li, Bo Li, Zongxia Du, Linlin Meng
School of Computer Science & Engineering
Beihang University, Beijing China
{lijx, libo, duzx, mengll}@act.buaa.edu.cn

***Abstract*—Cloud computing has become a popular computing paradigm in which virtualized and scalable resources are consolidated to provide services over Internet. However, the resource capability of a single cloud is generally limited, and some applications often require various cloud centers over Internet to deliver services together. Therefore, a Virtual Organization (VO) will be a promising approach to integrate services and users across multiple autonomous clouds. However, how to build a secure virtual organization to achieve the collaboration goals is a critical problem, and some issues such as membership agreement, policy conflict and trust management should be adequately addressed. In this paper, we present a framework CloudVO which based on security policies and trust management techniques to provide some flexible and dynamic VO management protocols for clouds. Therefore, CloudVO can achieve inter-cloud collaboration without destroying a cloud's local policies. Based on previous VO security management experiences, we have conducted some preliminary simulations to verify the effectiveness our approaches for cloud computing environments.**

Keywords: Cloud Computing, Virtual Organization, Trust Management, Access Control, Security Policy

I. INTRODUCTION

Nowadays, cloud computing [1-3] has become a popular computing paradigm in which virtualized and scalable resources are provided as services over the Internet. In a cloud, software is provided continuously as services with a simple and transparent manner, and the resources can be allocated dynamically, and users have a sense there are unlimited computing and storage service capabilities. Various cloud computing products and projects have been tremendously beneficial to network applications, such as Amazon EC2 [4], IBM Blue Cloud [5] etc. Cloud computing infrastructures have combined the virtualization technologies and the service oriented architecture (SOA) technologies [6], where a software need not been downloaded or maintained by users, and instead the cloud center will provide a remote access mechanism.

Based on the virtual machine system, a cloud platform can be effectively built. However, as application demands for scalable computing power, a single cloud center generally could not provide large scale of resources for Internet users. Therefore, multiple cloud centers sometimes need to collaborate to achieve some business goals. For example, an enterprise cloud has limited resources, and it can use some virtual resources provided by Amazon when some peak resource requests arrive. Besides, some researchers may want to build a virtual lab environment across geographical distribution of physical hosts, and it need to build an organization across multiple clouds.

It is known that a virtual organization has enabled resource sharing and problem solving across multiple organizations (i.e., autonomous domains) in a grid environments [7, 8]. The virtual organization (VO) is composed of a set of entities (e.g., resources, services, and users) from different autonomous domains collaborating in order to complete some cross-organization cooperative tasks. As the example shown in Figure 1, there are two clouds form a virtual organization, and their virtual machine pool can be shared each other. The virtual organization has gained advantages to server for industrial and commercial applications like dynamic enterprises, on demand computing, on demand services providers, outsourcing business processes and business-to-business collaborations.

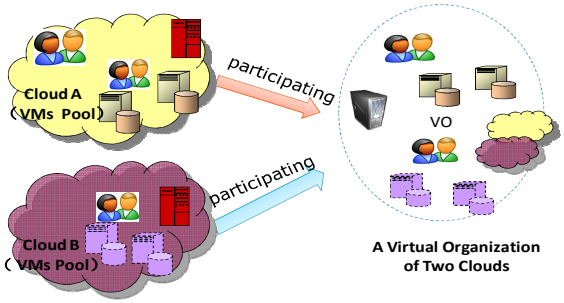

Figure 1. **A Virtual Organization composed of two clouds**

However, the virtual organization approach adopted in Grids aims to build a new organization without regard to the original security policy of a domain. In a cloud computing environment, every cloud has its own virtual resource pool and users, and makes its corresponding security policy. In a virtual organization, several clouds will be united, and thereby there are some typical characteristics as follows:

- *dynamic*: The formation of a cloud VO is often dynamic, and the original cloud security policy need be considered. Moreover, a user also joins in or leaves out a virtual organization dynamically.
- *autonomous*: The participating clouds of a VO have requirements to enforce the access control policy for their services independently, this is because every cloud has its own business goal. Moreover, no cloud is willing to join a virtual organization in the threat of violating its local cloud security policy.
- *distributed*: The clouds and large number of users are highly distributed, so some scalable trust

978-0-7695-4088-7/10 $26.00 © 2010 IEEE
DOI 10.1109/SNPD.2010.35

establishment mechanisms, e.g. authorization delegation, role mapping across domains, are needed.

These characteristics bring some new challenges to the virtual organization security management for multiple clouds. It is necessary to rely on the existing cloud security infrastructure and policies rather than set up fully new policies when a virtual organization needs to be created. However, current employed management infrastructures have some limitations respecting these challenges.

First, cloud data centers are dynamic and autonomous organizations, and it is hard to select a single administrator for a virtual organization. Therefore, how to make admission decision from multiple clouds for the user joining or leaving action is a first issue should be addressed.

Second, the existing Grid virtual organization mainly aims to build a new organization, and assign new memberships and permissions in a centralized VO server. But a cloud generally may server for users over Internet, e.g., Amazon, so the local cloud security policy should be considered with the VO policies together. Therefore, the collaboration policies may have some conflicts with the local cloud policy.

Third, in a virtual organization, not only the clouds are distributed, but also the virtual resources are distributed even in a cloud. Therefore, the trust management of virtual resources is another important issue. In CloudVO, we adopt role mapping policy and delegation policy to define the trust relationship among cloud virtual resources.

In this paper, we propose a scalable framework for the dynamic formation of VO in clouds, and some trust management and secure interoperation concepts are adopted to accomplish the collaborating goals. In particular, we use the delegation and role mapping policies to specify the collaboration relationship between clouds within the VO, and we still relay on the existing cloud security mechanism to make authorization decision. We believe it is an interesting direction for future cloud collaboration.

The remainder of this paper is organized as follows. Section II introduces the related work. We then present the overview of our framework, CloudVO, and the protocols our solution bases on in Section III. In Section IV we describe the system designing. In Section V, we conduct performance evaluation n Section VI. Finally, we conclude the whole paper in Section VII.

II. RELATED WORK

In a cloud computing environment, the “computing and storage capacity" that users required will be transferred to the Internet (the so-called "cloud"), and been provided as services through virtualization, SOA etc. technologies. Many leading and well-known international IT companies such as Amazon, Google, Microsoft etc. have started the relevant plans including EC2 (Elastic Compute Cloud), S3 (Simple Data Storage Service), Windows Azure and so on. In particular, the software-as-service mode has become a new application software delivery and sales mode, and the software licensing mode form traditional software distribution to leasing subscription, software deployment mode also shifts from local enterprise to the cloud centre.

A. Cloud Management System

In the cloud computing platform [2], the representative systems include Amazon computing, storage and other services. In the Amazon storage service S3, the targeted objects (similar documents), or memory segment (similar to directories), owners can assign access control policy to specify who can read and write. Computing service EC2 in Amazon uses the firewall policy configuration to create multiple virtual networks, based on IP and routing to achieve a different virtual machine security, prevent unauthorized access; in Amazon's SQS Messaging service, you can use it to easily create, store and access text to hear, is to build AWS (Amazon Web Service) applications based on operation commands can access privileges for different queue management, and adoption of policy rules in the underlying language, to achieve identity-based, the source IP address, and environmental context of the queue time for access authorization. However, these management methods are still limited to a single management domain virtual resource management, multi-domain Pool unable to meet the joint-time policy distribution management needs.

Recently, Google released Chrome OS [9] which is based on a solid foundation of Ubuntu Linux, it uses the Chrome Web browser as its interface to any and all applications. In Chrome OS, security is one of three key features. When users boot Chrome OS it checks the integrity of the OS and if it finds that your OS has somehow been corrupted or compromised by malware, it simply re-downloads a fresh copy of the OS from the cloud center. All user information is stored in the cloud, and the OS can be restored at any point in time without any detriment to user data.

VMware also released a cloud operating system vSphere™ 4 [10] which is an industry’s operating system for building an internal cloud, and enabling the delivery of efficient, flexible and reliable IT as a service. It is specifically designed to holistically manage large collections of infrastructure - CPUs, storage, networking-as a seamless, flexible and dynamic operating environment. VMware believes that virtualization is the key underpinning technology to enable cloud computing.

However, all of these cloud systems are used to build a cloud, and not take consideration of the collaboration among multiple clouds.

B. VO Systems in Grids

There are many research work in Grids, and the most two representative systems for virtual organization in grids are CAS and VOMS.

CAS (Community Authorization Server) [7] is developed by Globus project to enforce access control policies within the VO using X.509 extensions. It is a centralized approach for authorization in virtual organizations, and allows resource providers to delegate their access control policies to the community server. The major character is the fact that CAS does not involve groups or roles, but only permissions. This means that the ultimate authorization decision about the service access is made by the server.

VOMS (Virtual Organization Membership Service), which is similar with CAS, is a virtual organization

authorization service. It uses assertions that bind user attributes to make authorization decision rather than user identity. That is, to access shared resource, a user provides an attribute certificate issued from the VO to identify the role of an entity, and the policy regarding exactly what rights those membership grants is decided by ultimate resource provider. It is now used into gLite and Enabling Grids for E-Science in European (EGEE) projects.

Existing work about virtual organization lacks flexibility to support ad-hoc and pervasive collaboration, which is not appropriate for most dynamic short-lived cooperation among them and frequently happens. We argued that the new roles or permission assignment for the virtual organization is a limitation for the dynamic collaboration, and demolish the autonomy of every domain.

In addition, MyProxy [11] used in GSI allows users to delegate credentials to services acting on their behalf, but it is a simple single-sign on method. Currently, there are some advanced investigations in the area of trust management [12], e.g., dRBAC [13], RT [14], and secure interoperation [15] for the cross-domains collaboration. In these approaches, the delegation rules are used extensively and have made multi-domain collaborations a reality. However, these systems assumed the authorization delegation rules are fully distributed, and the users always needs find a trust chain dynamically to establish trust with the target service providers. Therefore it is not impractical to replace the existing systems in virtual organization for grids.

III. OVERVIEW OF CLOUDVO

A. Design of CloudVO

CloudVO is a virtual organization framework that based on the decentralized trust management, and provides a distributed, dynamic and autonomous collaborating environment.

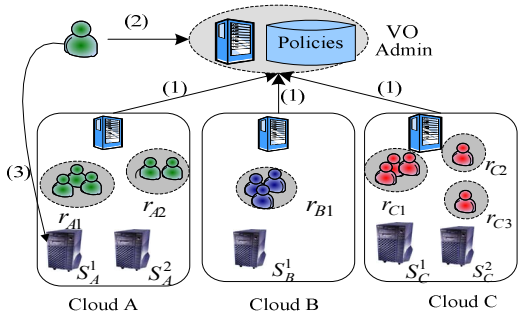

Figure 2. **The framework of CloudVO**

As Figure 2 shows, a virtual organization is composed of special services and roles from some clouds, e.g., Cloud *A*, *B* and *C*. It is noted that a centralized VO role membership server (called VO_{admin}) is agreed by all participating clouds in advance, and the ultimate role assignment still depends on the original cloud's policies. The VO polices are submitted by every clouds which describes the clouds collaboration. This means that a cloud user should firstly obtain the role membership from the VO server before he accesses the target services resided in other clouds.

Definition 1 (**Virtual Organization**) *A VO is denoted by a tuple* $VO = (u_{admin}, U, P_{VO}, \sum_U P)$, *where* u_{admin} *is the administrator of the centralized server in a CloudVO, and U is a set of participating clouds, e.g., Cloud A, Cloud B and Cloud C, and* P_{VO} *is a agreed policy for the CloudVO management by all VO members, and* $\sum_U P$ *is the collaboration policies made by all participating members from the set U.*

There are two key phases which play an important role during the virtual organization management. One phase is the virtual organization formation through the collaboration policies, and the u_{admin} will define the roles and role relations within the CloudVO. In general, a CloudVO server should be chosen and agreed by the decision-making domain groups in advance. The other phase is the service authorization within a target cloud when a request occurred from another cloud.

Unlike CAS and VOMS, as Figure 3 shows, CloudVO separates the phases of *User Policies* and *VO Policies*, and the user polices are reused based on the original policy in a local cloud. The VO policies are only for new roles and relations for collaborating tasks.

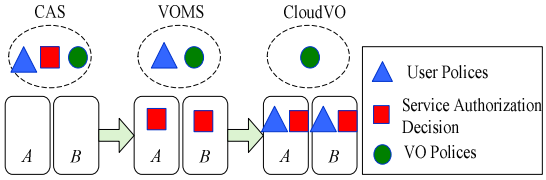

Figure 3. **A comparison of the three VO systems**

B. Policy Statement

To support the flexible and dynamic virtual organization management, we adopt the idea of trust management and security interoperation in decentralized internet. The basic concept in trust management is delegation, and the intention behind delegation is that some entities in a system delegates authority to another active entity to carry out some functions on behalf of the former. The role-based access control is much suitable for specifying the security requirements for a wide range of commercial, medical, government applications and moreover gets much standardization supporting.

In CloudVO, we adopt two types of policy statement.

Role Delegation Statement: $A \Rightarrow B[r,d]$, where *A* and *B* are the administrators of two clouds, *r* is the same role name used in both of two clouds, and *d* is called the *delegation depth* of this delegation, and it is either a positive integer or symbol ∞ (∞ means unlimited depth). *A* is called the *issuer* of this statement, and *B* is called the *delegatee* of this delegation.

Role Mapping Statement: *m*: (r_{A1}, r_{B1}), where r_{A1} and r_{B1} are roles of clouds *A* and *B* respectively, *cons* is the constrains on role parameters. For instance, the role *student* has a parameter of *age*, we can set a constraint as *cons*=(*age*>*18*)

In many cases, we can regard a delegation statement without delegation depth constraint as a special role mapping

statement with the same role name. But they are much different on semantic and thus we distinguish them in this paper.

At the service side, we adopt the XACML (with RBAC profile) to make fine-grained access control.

C. The Protocols in CloudVO

In CloudVO, we mainly address the problems of membership agreement, policy conflict and trust management. As demonstrated in Figure 2, there are some sub protocols in CloudVO, the first is virtual organization membership agreement protocol, the second is policy conflict checking protocol, and the third is trust management protocol.

Membership Agreement Protocol:

Firstly, we will show how to achieve membership agreement based on a decision-making group, and we define a threshold approach to process a new cloud's joining request.

Definition 2 (Decision-making Group): *Let $U_{SUPER} \in U$ be a set of decision-making clouds group. The admission to new member joining action is based on the rule*: threshold(k, $|U_{SUPER}|$), $k \leq |U_{SUPER}|$. *It indicates that admission should be permitted by at least k of participants from the set U_{SUPER}.*

The virtual organization management protocol is as follows:

(1) $A \rightarrow VO_{admin}$: $\{A_{cert}, join\}_{PriK\text{-}A}$

(2) $VO_{admin} \rightarrow U_{SUPER}$: $\{A_{cert}\}_{PriK\text{-}VOadmin}$

(3) $VO_{admin} \rightarrow A$: $\{resp\}_{PriK\text{-}VOadmin}$

During the first step of this protocol, a cloud A ($A \notin U$) wants to join into a CloudVO, it will send a *join request* message to the VO_{admin}. During the second step, after VO_{admin} verified the validation of the cloud owner, it will broadcast the *join* request message to all the members of decision-making group U_{SUPER}. If k out of members of U_{SUPER} permit this request, then the A will become a member of U. During the last step, VO_{admin} will send a *resp* message including the joining results to the cloud A.

***Policy Conflict Checking Protocol*:**

Secondly, we show how to check the policy conflict in the virtual organization collaboration policy. Based on the role mapping policy and delegation policy, there may be some policy conflict in this collaboration. As shown in Figure 4, there is a virtual organization composed of two clouds, and every cloud has its own following local security policy:

- ***Cloud A***: $ROLES$={ r_{A1}, r_{A2}}, and two policies (r_{A1}, r_{A2}) and m_2=(r_{VO1}, r_{A2}) ;
- ***Cloud B:*** $ROLES$={r_{B1}, r_{B2}, r_{B3}}, and three policies (r_{B1}, r_{B2}) , (r_{B1}, r_{B3}) and m_3=(r_{VO1},r_{B2}) .

The CloudVO Server has the following collaboration policy:

- ***CloudVO*** Server: ROLES={r_{VO2}, r_{VO1}}, and two policies (r_{VO2}, r_{VO1}) and m_1=(r_{A1},r_{VO2}).

In this example, because the collaboration policies are incorrect, and there will be a policy conflict to the local policy of Cloud A. From m_1=(r_{A1},r_{VO2}), (r_{VO2}, r_{VO1}) and m_2=(r_{VO1},r_{A2}), we can get (r_{A2}, r_{A1}), thereby there will be a policy conflict with (r_{A1}, r_{A2}) of Cloud A.

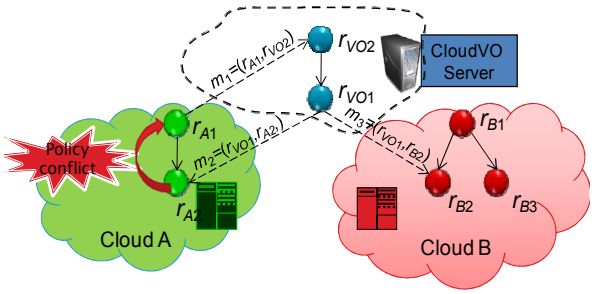

Figure 4. **A policy conflict example for CloudVO**

Based on the following Definition 3, the virtual organization collaboration policy should be checked after a cloud joining request is agreed by the decision-making group.

Definition 3 (Policy Conflict Checking**):** *Let VO=(u_{admin}, U, P_{VO}, $\sum_U P$) be a virtual organization, and if every* (r_1, r_2)$\in$ $P_{VO} \cup \sum_U P$, there is no (r_2, r_1), *then we say there is no policy conflict in this virtual organization.*

As shown in Figure 4, the virtual organization policy can be organized as a graph. In this graph, every node is the roles of clouds or CloudVO Server, and every edge is the original role hierarchy or role mapping relations. Based on this graph, it is very easy to verify whether there are some policy conflicts.

Definition 4 (Role Graph): *A* Role *Graph G is a directed graph* ⟨*R, E*⟩, *where R is a set of nodes of Role, and E is a set of edges adjacent to Node, and the following properties hold*:

- *The role set R is non-empty and finite*;
- *The edge in E has two types, RoleHierarchyEdge, and RoleMappingEdge, and special constrains are associated with the edge*.

In a role graph G, a policy conflict is there is a path (r_0, r_1, …, r_n) but (r_n, r_0) is also an edge of this graph. We use the following classic depth first searching (DFS) algorithm to detect the possible cycle in this graph. If there is a cycle, the virtual organization collaboration polices will not be secure and have policy conflicts.

Algorithm 1: Cycle Detection for a Role Graph

```
Input: A role graph G= ⟨R, E⟩
Output: whether exists policy conflict
public boolean rolegraph_cycledetection (RoleGraph G){
 for (each r∈ R) { // set the default value
  color[r] = WHITE;
  predecessor[r]= NIL;}
 for (each r∈ R) { //check the policy conflict in the graph
  if (color[r]== WHITE)
   graph_check(r);
  }
 }
private boolean graph_check (Node r){
   color[r] = BLACK;
   for (each p∈ getAdjacencyRoles(r) ){ // get the adjacent roles for the
role r
       if (color[p] == BLACK and predecessor[r] ≠p) {
          return false;}// there exists cycle, i.e. policy conflict
       if (color[p] = WHITE ) {
```

```
        predecessor[p] = r;
        graphcheck(p);} // recursively invoking this function
    }
    color[r] = BLACK;
   return true;
  }
```

Trust Management Protocol:

Thirdly, we show the trust establishment procedure in the CloudVO. When a user wants to login in the CloudVO, it should first send the required role membership list associated with a set of cloud *U*, the VO_{admin} will generate the mapping roles according the collaboration policies and return to user a *ticket*. The detail of this protocol is as follows:

(1) $u \rightarrow VO_{admin}$: $\{u_{cert}, ROLES\}_{PriK\text{-}u}$

(2) $VO_{admin} \rightarrow u$: $\{ticket_{roles}\}_{PriK\text{-}VO}$

(3) $u \rightarrow service$: $\{ticket_{roles}, service\}_{PriK\text{-}u}$

During the first step, the requester *u* will send its identity and belonged role set (signed by its cloud root credential) of its cloud to the VO_{admin}. During the second step, the VO_{admin} will verify the validation of the user certification and the role set of its local cloud, and then assign the qualified virtual organization roles (in a ticket with short validated time) to this user. During the last step, the user will send the service request to the target cloud, and VO membership of user *u*.

To improve the performance of policy conflict checking, we cached a partial of *Role Graph* to avoid parses the policy frequently. This approach will make some performance improvement when the collaboration policy is not often change. Otherwise, all policies will be parsed again.

IV. DESIGNING OF CLOUDVO SYSTEM

As shown in Figure 5, it is an example of CloudVO composed of two clouds. In a CloudVO, the major four types of workflow are as follows:

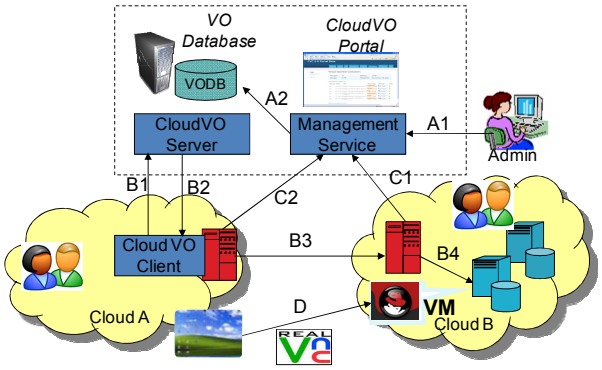

Figure 5. **The Workflow of CloudVO System.** The steps are: **A1:** the *Admin* logins into the CloudVO portal; **A2:** Authenticating the identity of admin and operating the VO database via *Management Service*; **B1:** The *Cloud A* sends VO membership credential requests **to** the *CloudVO Server*; **B2:** The *CloudVO Server* generates a credential; **B3:** The *Cloud A* portal sends a job scheduling request to another *Cloud B* portal; **B4**: The *Cloud B* makes authorization decision for the job scheduling across clouds; **C1, C2**: Virtual Resource Information Reporting or Querying; **D**: Accessing the VM Desktop via RealVNC.

- ***CloudVO Management:*** It is used to create, update, or delete a CloudVO, and the administrator can login in the CloudVO portal (step A1), and then configure the policy of a VO (step A2).
- ***Cross-Cloud Scheduling:*** It is used to retrieve a VO credential (steps B1&B2). The requester firstly executes the command cloud_vo_init (like cas-proxy-init) to get the initial assigned role membership. The security handlers are configured for the encryption and signature of message level security. Finally, the job is scheduled to another cloud through a secure tunnel (steps B3&B4).
- ***Virtual Resource Information Service***: It is used to collect the meta-information of virtual resources (e.g., shared virtual machines), and return the updated information to a cloud (steps C1&C2).
- ***VNC Viewer Accessing***: It is used to access a remote VM Desktop through VNC Viewer (step D), and the communication can be secured by SSH.

The architecture of the CloudVO Server service is illustrated in Figure 6.

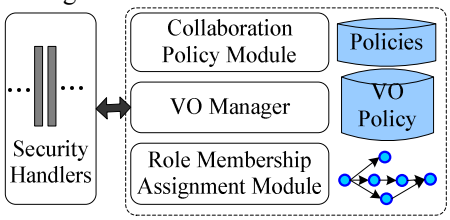

Figure 6. **he VO Management Service**

The basic components are as follows:

- ***VO Manager*:** The VO policy is enforced, and the requester will be authenticated when needs role membership assignment, which will invoke the other two modules according to the request message.
- ***Collaboration Policy Module***: It is used to check the possible conflict in the collaboration policy. Its functions are to ensure there is no violated policy for the whole virtual organization policy.
- ***Role Membership Assignment Module***: When a requester wants to access the services of other clouds, it will ask the server to assign the qualified VO roles.

V. SYSTEM SIMULATION

To get a feel of the performance of our approach, we have implemented some key components of CloudVO, and service is deployed on a cluster node with two Intel Xeon 2.8GHz CPU, 2G RAM, Linux operating system and 100Mbps Internet connection. Based on this simulation environment, we evaluated the approaches of CloudVO service by some comprehensive experiments and the results indicate it is feasible.

First of all, authentication is a core procedure for CloudVO protocols, and we use the following metric to measure the efficiency of different mechanisms:

- *Authentication Time.* In CloudVO, it is the total message round trip time period in user side from

sending the authentication request message to receiving the authentication response message.

In CloudVO, we test two types of security mechanism. One is the SOAP security mechanism of Globus GSI toolkit which provides a proxy credential. Another is the OpenSSH RSA security mechanism. We have executed the two mechanism ten times, and calculate their authentication time. As shown in Figure 7, the average authentication time is about 300-400 ms for two mechanisms, and there are no obvious differences between these two mechanisms.

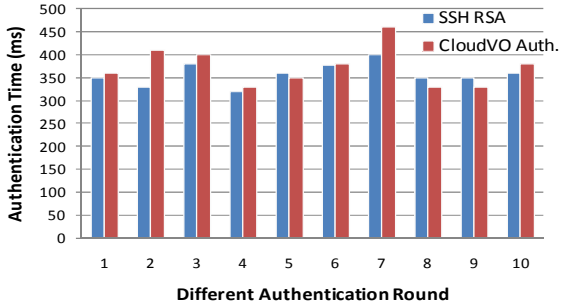

Figure 7. **Authentication Time of SSH and CloudVO**

Then, we evaluated the performance of detecting policy conflict algorithm. We simulate different types of virtual organizations with different number of roles and clouds. In this simulation, the number of roles is varied from 10 to 150, and the number of clouds is varied from 2 to 15, and we use the following metric to measure the effects of our approach.

- *Overall Policy Checking Time*. We measure the time to detect the possible conflicts in the virtual organization collaboration policies where every virtual organization has different number of clouds, and every cloud has different number of roles.

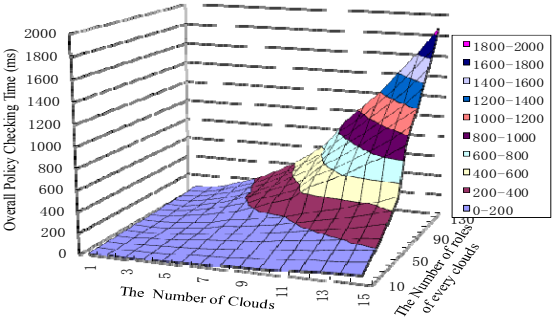

Figure 8. **The Overall Policy Conflict Detecting Time**

The results are shown in Figure 8, we can see that the time of policy conflict detecting for a virtual organization increases roughly linearly with the increasing number of clouds and roles. These two parameters are the major factors that impact the performance of policy conflict detection.

VI. Conclusion and Future Work

With the increasing prevalence of virtualization and cloud technologies as a platform for hosted service provision, virtual organization provides a new mode for multiple clouds to be rapidly united to provide services.

In this paper, we proposed a new framework, CloudVO, for secure virtual organization management across multiple clouds. CloudVO adopts the delegation and role mapping polices to bridge the collaboration inter-organization to address the challenges of distributed, dynamic and autonomous cloud environments. Our major contributions of CloudVO include membership agreement, policy conflict and trust management. Based on CloudVO, a virtual organization service are easily designed and deployed. We are also integrating our approaches into an ongoing virtual computing environment.

Acknowledgment

This work is partially supported by grants from the China National Science Foundation (No. 60903149, 60731160632), China 863 High-tech Program (No. 2009AA01Z419), China 973 Fundamental R&D Program (No. 2005CB321803).

References

[1] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, "Above the Clouds: A Berkeley View of Cloud Computing," 2009.

[2] A. d. Costanzo, M. D. d. Assunção, and R. Buyya, "Harnessing Cloud Technologies for a Virtualized Distributed Computing Infrastructure," IEEE Internet Computing, vol. 13, pp. 24-33, 2009.

[3] K. Keahey, M. Tsugawa, A. Matsunaga, and J. Fortes, "Sky Computing," IEEE Internet Computing, vol. 13, pp. 43-51, 2009.

[4] "The Amazon Elastic Compute Cloud (Amazon EC2)," pp. http://aws.amazon.com/ec2/.

[5] "IBM Blue Cloud," pp. http://www.ibm.com/ibm/cloud/.

[6] M. Papazoglou, "Service-oriented computing: concepts, characteristics and directions," presented at Proceedings of the Fourth International Conference on Web Information Systems Engineering(WISE 2003), 2003.

[7] L. Pearlman, V. Welch, I. Foster, C. Kesselman, and S. Tuecke, "A Community Authorization Service for Group Collaboration," presented at the IEEE 3rd International Workshop on Policies for Distributed Systems and Networks, Monteray, California, U.S.A. , 2001.

[8] I. Foster, C. Kesselman, and S. Tuecke, "The Anatomy of the Grid: Enabling Scalable Virtual Organizations," International Journal of High Performance Computing Applications, vol. 15, pp. 200-222, 2001.

[9] "Google Chrome OS," http://www.chromesys.com.

[10] "VMWare vSphere 4: Prviate Cloud Operating System," in http://www.vmware.com/products/vsphere/.

[11] J. Basney, M. Humphrey, and V. Welch, "The MyProxy Online Credential Repository," Software: Practice and Experience, vol. 35, pp. 801-816, 2005.

[12] M. Blaze, J. Feigenbaum, and J. Lacy, "Decentralized trust management," presented at IEEE Symposium on Security and Privacy, Oakland, CA,USA, 1996.

[13] E. Freudenthal, T. Pesin, L. Port, and E. Keenan, "dRBAC: Distributed Role-based Access Control for Dynamic Coalition Environments," presented at The 22 nd International Conference on Distributed Computing Systems (ICDCS'02), Vienna, Austria, 2002.

[14] N. Li, J. C. Mitchell, and W. H. Winsborough, "Design of a Role-based Trust Management Framework," presented at Proceedings of 2002 IEEE Symposium on Security and Privacy, Berkeley, California, 2002.

[15] M. Shehab, E. Bertino, and A. Ghafoor, "Secure collaboration in mediator-free environments," presented at Proceedings of the 12th ACM conference on Computer and communications security(CCS05), Alexandria, VA, USA, 2005.