THE INTERNATIONAL JOURNAL OF SCIENCE & TECHNOLEDGE

SSM: Secure Storage Migration among Cloud Providers

Rakesh Sachdeva
Department of Computer Science, G.N.D.U, India
Prabhpreet Kaur
Department of Computer Science, G.N.D.U, India

Abstract:
As Cloud Computing industry is advancing toward an era where resources are delivered as a service rather than a product. It provides infrastructure, software, application, platform, data and resources as a service over the internet with on-demand and pay-per-use model. Clouds are used for hosting a large range of services. These services between different Cloud Service Providers, has different pricing model and the cost of individual resources are very different. Heterogeneity among different Cloud providers causes many problems like vendor lock-in. In order to decrease the dependency and minimize the cost of running a service, it becomes mandatory to move the data between different Clouds. The focus of this paper is to solve portability conflict in IaaS(Storage) offerings. Results demonstrate in the paper shows how portability can be achieved using virtual appliances. Our efforts also focus on the security of storage data migration between different clouds. The solution to secure storage migration between clouds mainly involves in Secure Socket Layer Negotiations, exchanging temporary keys, tokens and blocks encryption in distributed file systems.

***Key words**: Cloud data; Storage Migration; Security; OVF; Cloud Sim; Data Centers; SSL*

1. Introduction

With the development of parallel computing distributed computing and grid computing a new computing evolved named as Cloud Computing. The main aim of Cloud Computing is to provide all IT resources as a service to end user whether it is storage, CPU power, software or anything. Cloud Computing virtually provides an infinite pool of resources which can be allocated to different users for short span with on-demand and pay-as-use basis. Also due to growing number of Cloud Service Providers has created a diverse Cloud market offers different pricing models within the same service schema but on different instance types, or different agreement conditions. In a scenario where consumer wants to shift data locally or migrate to another Cloud provider and stop using Cloud services. This situation is known as Portability issue in Cloud Computing. Pressman[1] defines portability as the ease with which the software can be transported from one environment to another without any data or configuration lost. There may be any reason for which a user wants to change their Cloud Service Provider. Some of them are require better alternative options, change in business and technical strategy or dissatisfaction from services of current Cloud service Provider. Also due to growing amount of data made users to consider an alternative storage plan transferring their data to a public or anther cloud[2], because their own private storage system cannot afford so large amount of data. It will be cost effective to migrate some storage data to another cloud rather than to invest too much money on hardware equipment. As the current services do not meet their needs, so users has to choose another Cloud Service Provider. Most of the time data migrations between cloud providers are conducted without considering any potential security threats such as regular network attack, middle attack, reply attack, and so on. To address these issues, we have propose temporary key negotiation service that manages all the expense and expertise required to maintain a large number of keys. Our solution is much secure and available for live migration of data storage from one cloud to another. It involves creating virtual cloud environment using hypervisors and creating appliances to be ported to another cloud provider. Among different standards, Cloud Environment in this paper has been virtualized through VMware Studio. VMware Studio[3] is an integrated development tool which takes existing applications and software then packages these applications into virtual machines or vApps also called virtual appliances that are ready to start, run or work. VMware Studio can build many kinds of VMs like Linux based VMs or Windows-based virtual machines. DMTF Open Virtualization Format (OVF)[4] which introduced in 2007 is a first step towards hypervisor independence thus achieving Cloud portability. OVF provides a way to move the virtual machine in form of virtual appliance from one hosted environment to another. OVF standardizes the use of a container that stores metadata of virtual machine and enables the migration of virtual machine. Live migration of cloud storage data i.e. of OVF has been achieved through CloudSim[5] using temporary key management system which ensures that the general stealth of unique keys.

The rest of the paper is structured as follows. In Section II, will demonstrate the related work and problems in existing systems, In Section III, this paper will describe major design considerations that are important in developing a virtual for an cloud provider

environment and storing the users data running on cloud and creating an virtual appliance to be migrated between the different cloud service providers. In the Section IV, the detail process of migration of cloud storage data with the current security threats like middle attack and solution through the key management service are described which also includes creation and distribution or temporary keys between SSL, aiding encryption, and deletion of these keys. Section V describes the whole prototype implementation of the model with the key functionalities. The last Section VI represents the concluding remarks and future work.

2. Problem Statement

Portability is ability to move data or application from one provider to another provider without any loss, large cost or security issues. Data is trapped with single provider so it forces cloud user to stay with one service provider. Achieving data portability is difficult because cloud providers uses different models, programming paradigms and market their own version of same technology. These models are difficult to change or adapt because they are transparent to cloud user. Big companies like Microsoft, Google, and Amazon are reluctant to agree on widely accepted standards promoting their own standards making it more difficult and complicated. Dominance of big companies increases the lock-in effect and it affects small scale and middle scale companies to enter into the cloud market. There is no portability factor maintained in policy management, security management and data management of Cloud Computing.

Lack of much common standards between different Cloud Service Provider also is a one of the major hurdle in portability i.e. different packaging standards and framework can possibly lead to different portability solutions which are not compatible with each other. Among different standards, there exist an common DMTF Open Virtualization Format (OVF), is a first step towards hypervisor independence thus achieving Cloud portability. OVF standardizes the use of a container that stores metadata of virtual machine and enables the migration of virtual machine. But the problem with this standard is that it is an offline method of migrating the stored data from one cloud Service Provider to another. Aim of this paper is to use existing standards and create an architecture using semantics so that cloud user can easily switch to other Cloud service Providers via Live Migration of data without any obstructions and thus solve portability issues in IaaS for Storage. Next problem discussed in this paper is how data can be migrated from one cloud storage Provider to another storage Provider more securely? If the provided services do not meet the user's needs, they have to choose another service provider. Most of current data migrations between are conducted without considering such potential security threats as middle attack, Distributed Denial of services, regular network attack, reply attack, and so on. As shown in Figure 1, there is no any special care or burden has been taken about the security issue while importing or exporting the confidential user's data at the time of migrating into or from Provider A to Provider B.

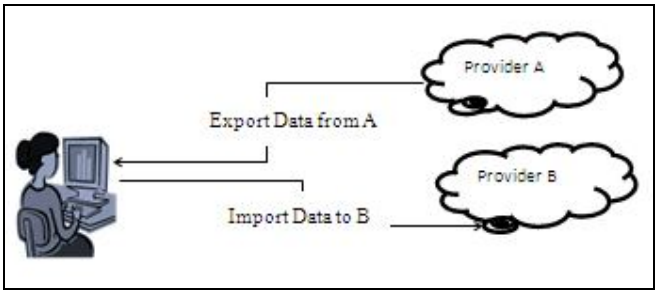

Figure 1: Data Migration From Cloud A to Cloud B

This paper also focuses on the security of public cloud, especially, the security issue involved in data migration between two clouds or between a Cloud Provider to another Cloud Provider.

3. State-of-the-Art in Portability Solutions

Govindarajan and Lakshmanan[6] report that besides brokers and APIs, interoperability should be investigated through control, data and other additional issues, such as security management , policy management and deployment provisioning aspects. Moreover, they also propose to build relevant layers of abstraction to help the issues on portability and interoperability. Mahdi et al.[7] presented a process to achieve data portability among Cloud providers. Design patterns and graphs are used to migrate the databases. However there are many limitations as they don't store foreign keys and are limited to family of design patterns. Fermín et al.[8] focuses on how a complex enterprise-class transactional applications can be made deployment-agnostic by means of the parameterization mechanisms offered by the Open Virtualization Format (OVF) standardized by the Distributed Management Task Force (DMTF), and then customized at deployment time (either automatically or by requesting user input) by means of the OVF activation mechanism. However they tested their proposed solution in Vmware Hypervisors. Portability issues across different hypervisors are still to discuss. Qingni Shen, Lizhe, Zhang, Xin Yang[9] discussed regarding threats and security issues in their paper ,their research motivation focuses on the security of data migration between different clouds. They describe some threats when doing data migration form one Provider to another and also proposes a security model to deal with the security issues on storage migration between cloud Provider. The solutions to secure data migration between cloud Providers mainly involve in Secure Socket Layer(SSL) communication or negotiation, block encryption and exchanging temporary tickets in distributed file system. However, their model for block encryption costs appx. 9 times as time cost without Migration Decision Module. Julian Jang-Jaccard, Avnish Manraj, Surya Nepal[10] also presents an proposed model to work with portable key management service that is highly secure and available for data migration among the Storage Cloud Providers. In their approach, the data is as secure as its corresponding keys. The main issue was where a user needs to manage a large number of (secret) keys

The solution was to all keys are stored in a tamper-proof hardware within a portable USB device that users can carry with them all the time in order to provide high security and availability while migration. But their approach for Trusted Platform Module is used to store the keys as well as to implement those keys was in the dedicated sealed environment so imitating towards individual hardware dependency while retrieving cloud data.

4. Creating Storage Intendancy among Cloud Providers

Infrastructure is needed for any PaaS offering. In the proposed solution in the paper infrastructure is separated form Platform provider. Third party infrastructure is proposed to use. PaaS provider has no control over user data. User data resides at separate IaaS and it is accessed by application at PaaS provider using internet. So, whenever user wants to change IaaS provider user doesn't have to worry about any data loss. This gives total freedom for user to select any infrastructure provider. There in need some kind of requirements to achieve this kind of independency among all three layers i.e IaaS, PaaS and SaaS.

Some of requirements of this solution are:

- There are basic 4 actors in this scenario. User, Platform Provider, Infrastructure Provider and SLA manager. Account of all these actor should be managed from a web portal where these actors login or signup. So, first requirement is Login and SignUp of actors.
- Secondly, provider should able to register their services and user should able to search most capable services. Semantic matching should be used for making matching process more effective.
- After Infrastructure Provider is selected then platform provider should be search. Virtual appliance should be created using OVF packaging format so that it can be hypervisor neutral.
- A SLA manager should be assigned to create and maintain quality of service.

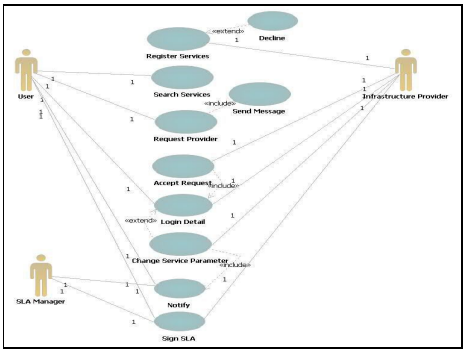

Figure 2: Infrastructure Deployment

Above written are basic requirements of proposed solution. As shown in Figure 2, the use case depicts main activities with infrastructure provider in the propose architecture. User can search Infrastructure provider, request the provider for using its services. Infrastructure provider accepts or rejects the request. If accepted SLA manager create an SLA between user and provider based on quality of service and store it.

4.1. Appliance Creation Service (ACS)

After selecting all requirements and signing SLA next step is to create an appliance of desired requirements to deploy it to any infrastructure provider. Appliances are prebuilt, pre-configured, ready to run, hypervisor neutral virtual machines. There are mainly two solutions for creating virtual appliances. First solution there exist different hypervisors that are used to create Virtual Machine like VMware, VirtualBox, VmSphere etc. New appliance can be created according to user requirements using tools like Vmware Studio. Once created appliance with user requirements are stored in database for any future use. These databases are known as Appliance Market Place. All these hypervisors has its own market place for its appliances. Second solution for creating virtual appliances is the use of Simulators. Simulators are often used as a technical validation mean, and most of the time, a custom tool is developed for each paper. The problem with the simulators is that most of the simulators have not gone through a proper validation. Depends upon each accuracy and the reproducibility of their results have not been verified. Cloud simulators are essential to test both VM migration techniques as well as Cloud Broker Algorithms. There are many different simulation tools that can be used for the experimental development of Cloud infrastructures, such as SGCB, CloudSim,, or iCanCloud. CloudSim[11] is an extensible toolkit used to model and simulate Cloud infrastructures, including data centres, users, user workloads, and application provisioning. In this Paper we have used both kinds both the above mentioned kinds to achieve and test our results regarding independency, interoperability and portability of our VM's among different platforms.

4.2. OVF Packaging

After gaining required disk image or space, user need to pack these image along with other user requirements such as operating system and hardware requirements into a standard format which can run at any hypervisor. The OVF is chosen for this purpose which is adopted by the industry (Vmware, VirtualBox, Citrix). An OVF is a packaging format for software appliances. Once installed, an OVF adds them to the user's resources list and infrastructure as a self-consistent, self-contained, software solution for achieving a desired goal. An OVF might contain a fully-functional and tested web-server, OS and database combination. It is mainly used for transport mechanism for virtual machine templates. A Single OVF may contain a single VM, or many VMs

(dependents upon the need of software appliance developer for the arrangement best suits their application). Before they can be run OVFs must be installed in the user's System. OVF packaging is done by using ovftool. Hypervisor like VirtualBox or VMware provides OVF Tool through both command-line utility and User Interface that supports importing and exporting of OVF packages from ESX hosts and other hypervisor products.
Some file systems on clouds may have restriction such as on maximum file size. For example, FAT32 file system allows files maximum up to 2GB.So user can split the OVF files from a generated OVF package into pieces of any restricted maximum size.
-To create an OVF package optimized for a any file system which supports maximum of 3 GB, use the following command on OVF Tool:> ovftool --chunkSize= 3gb <source> Package.ovf
where <Source> is the path of the source OVF package file and Package.ovf is the OVF package to be spit into chunks. Each file chunk has a sequentially numbered suffix. For example, A 12GB disk, after split on the chunks have these names:
dsk1.vmdsk.x0000000, dsk1.vmdsk.x0000001, dsk1.vmdsk.x0000002, dsk1.vmdsk.x0000004
So these files can be uploaded on the Another Hypervisor like VMware, VirtualBox or simulator CloudSim to be uploaded for the migration of Storage data form one Cloud Storage provider to Another Cloud Storage Provider.

5. Data Migration of Current Storage System
While current cloud storage systems like Hadoop Distributed File System(HDFS), Amazon's S3, provide data migration service to cloud users, however they all do not take into account the potential threats and worries brought by user's Confidential data migration [12].

5.1. Current Migration Models with Threats
The whole procedure of data migration between Cloud Storage Provider 1 and Cloud Storage Provider 2. As shown in Figure 3. There are three kinds of entities which involve in the process[13].

- User: User which indicates who wants to migrate their Storage data and sends the migration request to its storage service provider.
- Central node: The central node, at the current Storage Provider i.e. source cluster's task is responsible for accepting authenticated user's command to start the migration task and to accept the read request from the 'data node' where the actual data in the source cluster is stored and return location information about the data. The central node of the target system where the data to be migrate, is responsible for processing write requests from Data Nodes of source Provider.
- Data Node: Data node stores the user's data and process the any kind of data request from its Central Node. The request running at the Data Node of the source cluster is started by the Central Node and performs a partial task for whole migration job. The request gets location information of source files along with current location and outputting files with target addresses or locations. As the User sends the Migration Request to current storage service provider (System 1), the request includes source data, target place (where the data to be migrate) and subject. The Central Node of System 1 checks the user's authentication to verify if that user has enough right to migrate the data specified in the Migration Request. The Central Node of System 1 sends the Write Request to the Central Node of target System 2 which contains subject, destination path.
- The Target Central Node of System 2 checks the permission of that User to verify if User A has enough right to write that data to the path specified in the Write Request. If yes, permission accepted, The Central Node of System 2 will synchronize with its Data Nodes to Generate Write Tokens that will be used by Data Nodes of System 1 for writing data to Data Nodes of System 2. After receiving the Token from the Central node of System 2, The Central Node of System 1 distribute token to its Data Nodes which store the data. The Central Node also sends the address of target Data Node to the source Data Node. Then, the source Data Nodes sends blocks to the target Data Nodes.

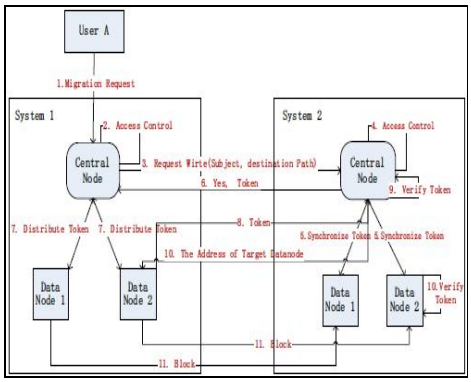

Figure 3: A General Cloud Storage System Architecture

5.2. Threats in Current Techniques
From the above mentioned procedure, there may exist three potential security threats during the data migration. Therefore, the aim of this paper to take into account how to migrate data securely concurrently to decrease the time cost to migrate. In the next section, the paper presents the solution for migrating the data more securely through the negotiation of temporary key management system.

- The communication takes between the Central Nodes (data centers) in the two Cloud Storage Systems. The tokens are the principle part for the whole the migration strategy. During the initial communication process, the security arguments i.e. token might be intercepted or tampered. The attacker will act as a legal party to get user's confidential data.
- The source Data Node sends request for data access with the token generated in the migration process. It is possible that attacker will intercept the token and get the information about the addresses which is supposed to be only known by the source Data Node.
- The actual data transfers takes between Data Nodes in the two Cloud Storage Systems. Except the confidentiality and integrity of the Data transferred, attacker might also intercept the Data and get the Hash Value of the Data after the temperament. At last, the attacker will reply the tempered data to the source Data Node.

Therefore, the aim of this paper to take into account how to migrate data securely concurrently to decrease the time cost to migrate. In the next section, the paper presents the solution for migrating the data more securely through the negotiation of temporary key management system.

6. Securing the Data Migration

Both Data Nodes and Central Node of any System are possible to become the target or disguised object of attackers. There are some methods needs to enhance the security of the migration process through the authentication of entities and protection of migrated data.

First, before the beginning the migration process, the Central Nodes of both the Systems should do some kind of security negotiation to prepare for the sequential migration procedure and the negotiation done through the authentication of both central nodes. The Central Node of receiving end of the migration process, must have right to authenticate the Data Nodes of source cluster. No one can write data to the target cluster, without the permission of the Central Node of destination cluster. Second, both Systems (System1 and System 2) have to handle data migration by some security method. Without doing so, attackers can intercept the keys, monitor or temper the data. The user's data may be highly confidential and important, so the migration of this kind of data must achieve confidentiality.

6.1. SSL Phase Negotiation

It is well known that data migration process happens between two different in two different cloud service providers. It is necessary to establish a security connection between two different clusters i.e, cloud system before the migration process. In the SSL protocol-based negotiation process, the two Central Nodes of both systems will be responsible for negotiating some important security parameters required for the migration process to take process. These parameters include: the temporary session key for message authentication code (MAC) computing, ran key (randomly generated key) for symmetric encryption and the temporary tickets with minimum migration privilege.

6.2. Migration Tickets with Minimum Privilege

As Tickets are generally used to verify the identity of any system and grant appropriate permission to the verified system. For Cloud storage migration process, the main holders of the Tickets are the Data Nodes of source cluster where the data is currently placed. These tickets are requested by the source Central Node and accepted by the target Central Node using SSL connection there are still security risks. An attacker can intercept the tickets, and get read permissions. As the level of security of software can not completely prevent the occurrence of this viral behavior, but it can restrict an attacker to minimize the impact of stolen tickets. So firstly, minimize the permissions that one ticket can grant without affecting the normal migration. So that even if an attacker may able to obtains tickets, the attacker should have limited permissions. The permission that is associated with one ticket can grant is called 'the smallest cluster migration privilege'. This ticket contains the information about two IDs (the source Data Node ID and output file ID). The ticket is temporary and should be used once only. In case if there are duplicate tickets waiting for verification, indicating the existence of the attacker. This situation should be promptly reflected to the migration administer. Moreover, because in cloud environment during migration there always many source 'data nodes' running in parallel, the use of two IDs to define this kind of ticket is appropriate. This solution also makes the tickets with the characteristics of one-time usage, so once a ticket is verified, this ticket can be destroyed.

6.3. Data Encryption Process

Encryption using the secret key discussed during the SSL negotiation phrase makes sure the confidence of migration data. For each data block using the hash algorithm and append generated hash value. The solution includes use random key determined using SSL negotiation to compute a message authentication code with the hash value. So that even if an attacker might be able to intercept the hash value to determine data. It should be limited to only one block The final network transmission format of data is: (data block + message authentication code + block hash value). This format can achieve integrity, confidence and prevent tempering.

7. Implementing Protocol

As depicted in Figure 4, our technical prototype is based on subproject CloudSim[14]: a toolkit for modeling and simulation of cloud. The CloudSim toolkit simulates a distributed file system comprised of clusters of cheap machines. It supports behavior and system modeling of Cloud components such as data centers, Brokers, virtual machines, Cloudlets and resource provisioning components such as common cloud storage architecture. DATACENTERS act as Central Node in the system and their STORAGE NODES act as Data Nodes in the CloudSim. The data which has to be migrated packed in the OVF standard format in Hypervisor such as VMware or VirtualBox using OVF Tool. The packed OVF file is splits according to the target Cloud file system that it

supports as explained in Sec IV OVF packaging. Then the splined files are uploaded in CloudSim and migrated through one cloud's DataCenter's nodes to another DataCenter's nodes. The security in migration process of whole storage data is achieved in the through the following procedure.

The secure migration in the entire process works as follows :

- After verifying the request of Storage migration permission of User, the SLA will request a SSL connection between the DataCenter's of both the Cloud Systems. The two DataCenters will negotiate related parameters mentioned in Section VI.
- After receiving Migration Request from System 1's DataCenter, System 2's DataCenter generates a temporary session key (T.Kdn) that will be used for communication between the source Nodes and the target DataCenter, and generates a random number (R.hash) that will be used for double hash computation. Subsequently, the target DataCenter sends T.Kdn and R.hash to the source DataCenter.
- After distributing migration task to Nodes, the System 1's DataCenter sends a request for tickets with the list of IP of Nodes.
- System 2's DataCenter generates a series of tickets and encrypts the tickets by T.Kdst, a key only known by System 2, and then returns the encrypted tickets **T** to System 1's DataCenter. **T=(tickets{IP,T.Kdst {ticket(s,ip,filepath)}})**
- After receiving the encrypted tickets, the System 1's DataCenter distributes the ticket, T.Kdn, and R.Dhash to its every Nodes one by one.
- The SSL connection terminates after the distribution of keys. Every Node encrypts the encypted tickets with a timestamp by T.Kdn, and send the Double encypted tickets to System 2's DataCenter.
- System 2's DataCenter decrypts the tickets and
- updates the timestamp to every ticket, and eventually, it will return the address of System 2's Node one by one which the System 1's Node sends block to.
- Every System 1's Node receives the address of System 2's Node. Before the transmission, Node will encrypt the block using session key, make a hash value (Hash1) to block and another hash value (Hash 2)by using R.hash. Then, System 2's Node sends the three parts to System 2's Node. Figure 4, explains the SSL negotiation explains the procedure for exchanging temporary keys, Session keys and secure data migration process between different Cloud Service Providers. The CloudSim tool used for large inter-cluster migration and the work of migrating of data is done by the nodes that run in parallel across the cluster. The secure inter-cloud migration control was implemented in the DataCenters. CloudSim is a secure communication protocols between two Datacenters and between Storage Nodes uses block transmission protection.

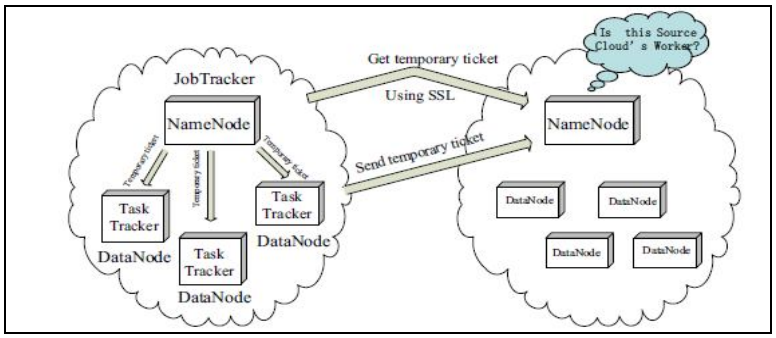

Figure 4: SSL Prototype in Data Migration

8. Conclusion and Future Work

This paper provides an insight to the essential aspect of standardization in Cloud computing, portability using virtualization and independence in cloud IaaS environment. OVF tool in Hypervisors provides an independent approach to implement appliances to any Provider. Converting user requirements to OVF to be a standard package format for cloud deployment. OVF helps user to change IaaS provider without changing PaaS provider and vice-versa. SLA manager monitors QoS provided to the user by the Cloud Service Provider. While migrating user's confidential data there may exist several security threats. The SSL embedded in the Cloud Node of cloud storage system succeeds in solving the threats that may brought up during the data migration process. SSL protocol negotiation helps nodes to share parameters more safely. This mentioned protocol for protecting confidential and private data service can be adopted by almost all of the cloud storage system, as it is purely based on the client end and has no much interaction or burden with the server or cloud provider.

9. References

1. Roger S. Pressman,.: McGraw Hill Internat., ch.14, pp. 398-415.
2. S.Kmara, K.Lauter, "Cryptographic Cloud Storage," Proceedings of Financial Cryptography: Workshop on Real-Life Cryptographic Protocols and Standardization 2010, January 2010, pp. 111-116.
3. Vmware Inc., "Developer"s Guide to Building vApps and Virtual Appliances", Palo Alto, CA, User Manual EN-000831-00, 2012.
4. DMTF, "Open Virtualization Format v.1.0," DMTF, White Paper DSP 2017 v1.0.0, 2007.

5. Rodrigo N. Calheiros, Rajiv Ranjan, Anton Beloglazov, Cesar A. F. De Rose, and Rajkumar Buyya. Cloudsim: a toolkit for modelling and simulation of cloud computing environments and evaluation of resource provisioning algorithms. Softw. Pract. Exper., , January 2011. Govindarajan and Lakshmanan, "Overview of Cloud Standards," Cloud Computing, Springer London, vol. 1, no. 1, pp. 77-89, 2010.
6. Mahdi Negahi Shirazi, Ho Chin kauan, and Hossein Dolatabadi, "Design Patterns to Enable Data Portability between Clouds‟ Databases," in 12th International Conference on Computational Science and Its Applications, 2012, pp. 117-120.
7. Fermín Galán, Miguel Gómez, Fernando de la Iglesia, Ignacio Blasco, and Daniel Morán, "Autoconfiguration of Enterprise-class Application Deployment in Virtualized Infrastructure Using OVF Activation Mechanisms," in 6th International DMTF workshop on Systems and Virtualization Management (SVM 2012), 2012, pp. 412-421.
8. Shen Q., Zhang L., Yang X., Yang Y.,Wu Z.,"SecDM: Securing Data Migration Between Cloud Storage Systems**",** in 9th IEEE International Conference on Dependable, Autonomic and Secure Computing, 2011,pp. 636-641.
9. Jang-Jaccard J., Manraj A., Nepal S., "Portable Key Management Service for Cloud Storage", 8th International Conference Conference on Collaborative Computing: Networking, Applications and Worksharing , October 14-17, 2012, pp. 147-156.
10. S.K. Garg and R. Buyya. Networkcloudsim: Modelling parallel applications in cloud simulations. In Utility and Cloud Computing (UCC), 2011 Fourth IEEE International Conference on, pages 105{113, 2011.
11. W. Cong, Q. Wang, K. Ren, W. Lou, "Ensuring data storage security in cloud computing," In: Proc. of IWQoS 2009, 2009, pp. 1-9.
12. Konstantin Shvachko, Hairong Kuang, Sanjay Radia, Robert Chansler. The Hadoop Distributed File System. In Proceedings of the 26th IEEE Symposium on Mass Storage Systems and Technologies, pp: 1-10, 3-7 May 2010, Incline Village, NV.
13. Buyya R, Ranjan R, Calheiros RN. InterCloud: Utility-oriented federation of cloud computing environments for scaling of application services. Proceedings of the 10th International Conference on Algorithms and Architectures for Parallel Processing (ICA3PP 2010), Busan, South Korea. Springer: Germany, pp. 328–336,21–23 May 2010