International Journal of Information & Computation Technology.
ISSN 0974-2239 Volume 4, Number 17 (2014), pp. 1815-1823

http://www. irphouse.com

Performance Gathering and Implementing Portability on Cloud Storage Data

Rakesh Sachdeva[1], Prabhpreet Kaur[2]

[1, 2]Dept. of CSE, Guru Nanak Dev University
Amritsar, India.

Abstract

As information and communication technology develop rapidly. Researchers, Journalists, experts and IT analysts are referring towards the cloud as a technology revolution. Cloud computing is not just a fad anymore. Even in short, cloud computing is completely is affecting almost everyone in real. It is a distributed computing model for enabling on-demand access to rapidly scalable resources, the resources include infrastructure as a service (IaaS), platforms as a service (PaaS), and software as a service (SaaS). Cloud Computing can be very beneficial for both small scale and middle scale organizations as these organizations can lease IaaS at economical rates which reduces capital cost. Quality of Service (QoS) is monitor and maintained by signing a Service Level Agreement (SLA) between cloud providers and users. The focus of this research paper is to solve the issue of portability conflict in IaaS offerings and is a roadmap toward more competitive market for cloud providers and users. The focus of this research is to solve portability conflict in IaaS offering. An algorithm is proposed for providing compensation in case of violation of stated SLA. A live migration technique also has been explained to migrate storage data from one cloud provider to another more securely.

.

Keywords- IaaS, Virtualization, parameters, SLA, Migration, OVF, Cloudsim, Temporary keys.

Introduction

In today's world, Technology plays the major role, as it enables creation of new platforms for the business, with the development of new technology devices. With the development in new networking technologies and the increase in the demand for computing resources, many organizations have been prompted to outsource their IaaS resources such as storage, networks or other computing needs from the cheaper

infrastructure resources provided by cloud providers. The IaaS provisioning deals with the physical allocation, configuration and implementation of different resources, which involve different physical hosts, storage services, different virtualization technologies, as well as and networks in the cloud along with the datacenters, virtual machines on different locations**.** The main reason why most of the organizations are moving towards cloud computing paradigm is due to the scalability, on-demand Services and pay as you use policies. Scalable means infrastructure automatically scales up and down according to requirement. Cloud computing also provides infinite number of resources virtually for users on demand over the network so users can rent IT resources, they can compose their software systems for dynamic and flexible need. Pay as you use model works on the policy of money is paid until the provisioned resources are used for example Processor for two hours and storage for five days.

Despite the advancement in the technology of Cloud Computing from industry as well as academics contributions, it faces some challenges for its worldwide adoption especially in the case of small organizations. The main reason for not adopting this technology is that most of the current cloud solutions do not meet the considerations of portability and interoperability. In Portable environment users can compare the services and choose from different providers and can easily switch between different Cloud providers without effecting user data and configuration. Cloud Computing is pay-as-you-use model which totally depends upon quality of services provided by the provider. If the services provided by the providers are taking more time, money than conventional methods then no one will adopt these services. If quality of service is discussed earlier and signed by both parties then provider is bonded to provide that level of service which increases adoption and trust of consumers in Cloud Computing. A Service Level Agreement (SLA) plays very valuable role for all parties in terms of understanding cost, schedule and performance because their relationship is stated explicitly. SLA's creation is very complex and difficult task. As both parties should be benefitted from SLA so, Service provider should know all types risks that may evolve in SLA before signing it. Mastroeni et al. [1] examine the risk of violating the SLA obligations. Many agreements are signed before using Cloud services and compensation is provided for violations as per discussed in agreement but when a violation of privacy and illegal access to sensitive information is detected, it could become difficult to identify who is liable for such violations in virtualized environment [2]. Cloud resources are based on SLA, which states usage terms and conditions and proper compensation for violations. QoS information provided by cloud provider can't be trusted because data source is in control of resource provider rather than Cloud user. How Cloud user will know its SLA is achieved or not? Process is required to specify and manage SLA so that information can't be change or false. Most of the algorithms are focused on violation towards the users only even if the user wants to migrate its storage before its time completion as the users does not getting the desired performance from its current provider. So an proper compensation algorithm is needed when there is any violation.

Cloud can be commonly classified as either private or public. As a user of public storage cloud, requirements like price, security level, and storage amounts varies among different users. If the service provided by the current cloud provider does not

meet user's needs, users have to choose another service provider. The reason for which a user wants to change their current Cloud Service Provider may be due to change in business and technical strategy or dissatisfaction from services of current Cloud service Provider. It will be more convenient to the user to move their storage data to another cloud provider.

Performance gathering

As SLA is signed by all providers and users and it is monitored by a third party SLA Manager. SLA manager evaluates the performance of the Cloud Providers. To evaluate the performance of me need an efficient simulator which can check the performance of current cloud provider on the basis of different parameters such as speed, accuracy or storage or instructions execution per sec. Scilab is an good option to check the performance of check the performance of distributed environment. Based upon some factors and parameters, some graphs has been generated, which analyses the performance of both current Cloud Provider and the cloud provider to where the data have to migrate. It observers all values of most of the parameters which are signed during the Service Level Agreement by the Cloud Service providers as well as the offered services during the running period of time to the users based upon that performance the user decide whether there is any need to migrate from current storage service provider if the offered services scale does not matches with desired or signed service's scale during SLA.

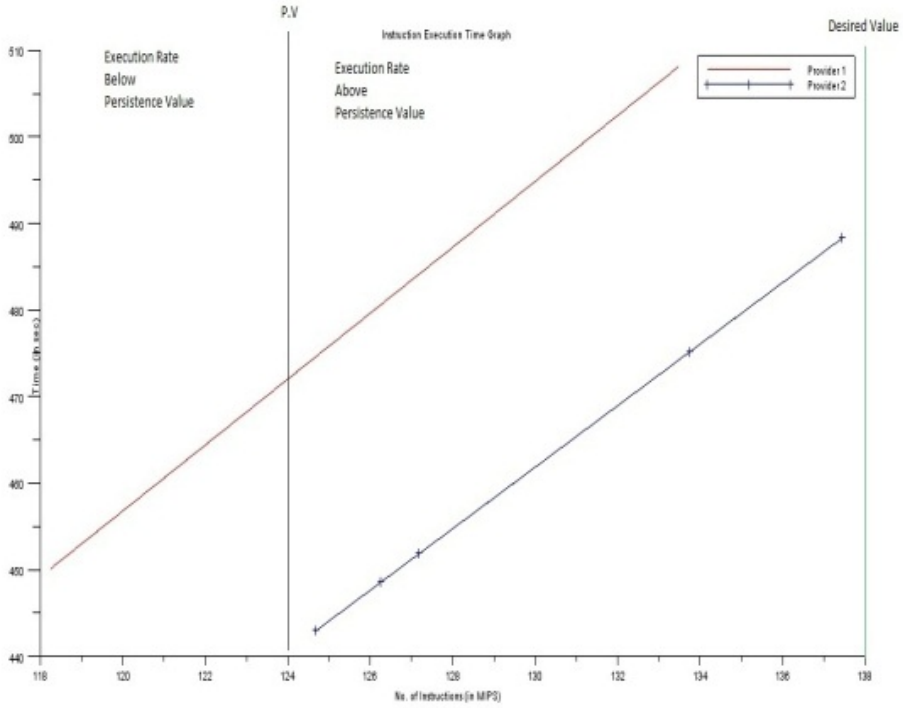
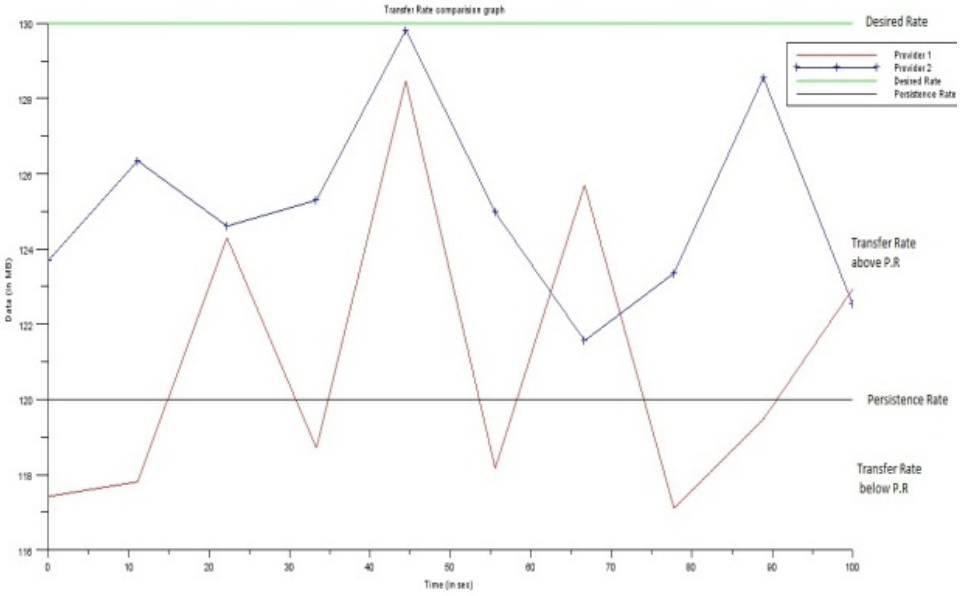


Fig 1(a) Instruction execution Time Graph



. Fig.1(b)Transfer rate comparision graph

Figure 1(a) shows the variability in execution of the instructions rate of different cloud providers. Desired value is the value which is signed during SLA or for which the user is paying. P.V is the Persistence value provides the probability to which the value can be persist. Every Desired value can't be achieved in real time application because of many reasons. For example, There are total 3, 00, 000 of instructions. Desired rate for executions of the instructions is 138 instructions per second (MIPS) so 138 concurrent requests can be managed by provider and here PV value is 124 means the service provider has to execute at least 124 instructions per second but not less than 124.As shown in fig1(a), the current service provider (Provider 1) is enable to provide that scale of the services for which the user has signed for. There are some points at which provider 1 is offering services of instruction execution per second is below P.V which leads to loss of quality of services and offering for which the user is paying. On the other hand the Provider 2, has the services and offering of instruction execution per second is always above the P.V.

Similar is the case of Transfer rate Comparison graph in figure 1(b), where the transmission rate offered by both cloud Service providers i.e current cloud provider and service provider where the data has to be migrate. Here is the variability observed while performance gathering in both cloud providers for which the user is paying or going to pay. Again, there are some points in which transfer rate is below persistence value of current service provider which may effects the clients of the user for which the user is paying to cloud service provider. On the other hand, the provider 2's transfer rate is still above the Persistence value. So, if the services or parameters offered by the current cloud provider are most of the time below P.V the user has to

take decision of migration of their whole storage data to another which will provide better services to the users on the same cost on same amount of data.

SLA COMPENSATION generation ALGORITHM

This paper presents a solution for portability conflicts raised during deployment and migration of data in a IaaS environment by separating three fundamental IaaS entities: Application, User Data and Infrastructure. Quality of Service (QoS) is maintained by signing a Service Level Agreement (SLA) between all parties and it is monitored by third party SLA manager. With the introduction of Service Level Agreement (SLA) and SLA violation compensation algorithms quality of service delivered is maintained. Many agreements are signed before using Cloud services and proper compensation is provided for violations. But when a violation of privacy and illegal access to sensitive information is detected, it could become difficult to identify who is liable for such violations in virtualized environment. An algorithm is proposed for providing compensation in case of violation of stated SLA. In which, the user doesn't have to worry about Quality because proper compensation will be provided if SLA is violated. In Automated IaaS management Architecture SLA manager finds any violation from signed SLA and calculates its compensation. Here, an algorithm is proposed to calculate compensation.

SLA COMPENSATION GENERATION ALGORITHM

```
ENTER THE VALUE OF EACH PARAMETERS DV, PV, AV, TIME AND COMP/UNIT IN MATRICES NAMED SLA[N][5]
      // n is the number of parameter, here n=4.
1.  For i= 1 to n
   //if  Achieved value is greater than Persistence value, no compensation generated
2.  if(PV[i]<AV[i]) then
3.  Print "No Compensation of ith Parameter"
   // If Achieved value of parameter is lower than Persistence value, compensation generated depends upon time
and compensation per unit cost of individual parameter.
4.  else (PV[i]>=AV[i])
5.  comp[i]=(DV[i]-AV[i])*Tim[i]*C_PU[i];
6.  Print "Compensation of ith Parameter is:" comp[i]
7.  total_comp=total_comp+comp[i];
8.  Print total_comp;
9.   End else if;
10. End;
```

Desire Value

DV is value that provider promise to provide to Cloud user. It is recorded when provider and user signed SLA.

Persistence Value:

PV provides the probability to which desired value will be achieved or the least value for which the cloud user can persist or tolerate.

Achieved Value
AV is value of metrics that is actually achieved or provided by provider. Its value is continuously by SLA monitoring organization.

Time:
It is amount of time after which each metrics will be checked. By default this value is in months.

Compensation per Unit
It is amount of cost to be paid in compensation for each unit of PV. This value is in Dollars.

Here in Figure 2, an example has been explained in which cloud service provider has to be pay an compensation amount of 2294$ to the cloud provider for the violation of their services as mentioned in the Service Level Agreement.

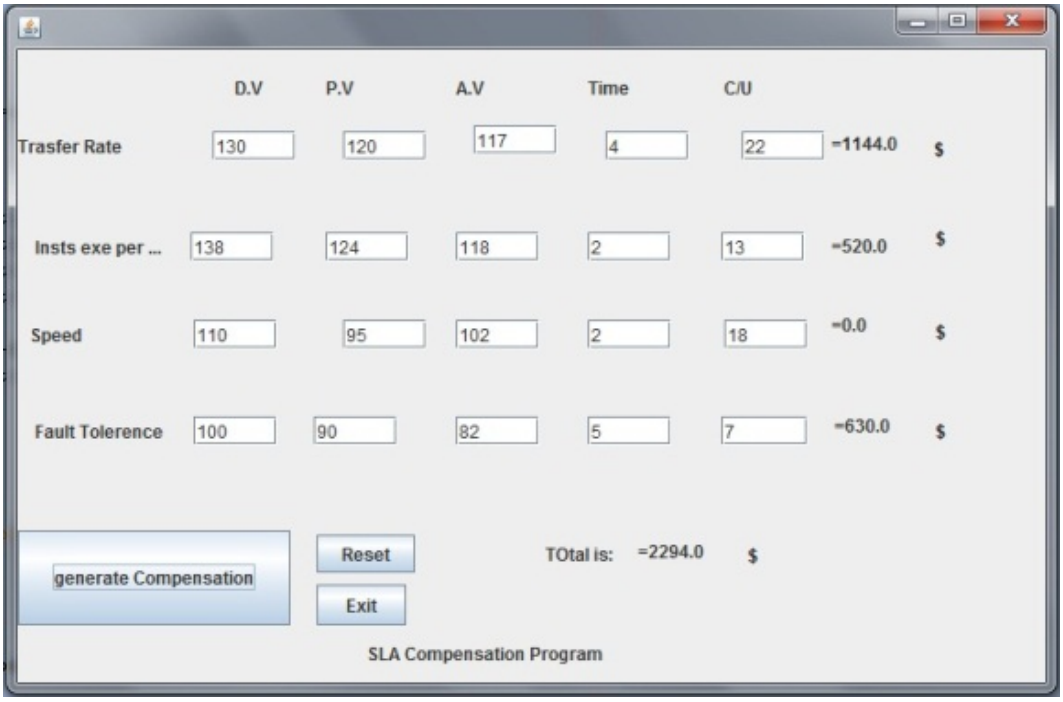


Fig.2 Compensation Generation Program

Portability of storage data
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 different cloud providers use different models, programming languages and market paradigms and their own version of same technology. These

models are difficult to change or adapt because they are transparent to cloud user. Big organizations like Amazon, Google, Facebook and Microsoft are reluctant to agree on widely accepted standards promoting their own standards making it more difficult and complicated. Dominance of big organizations increases the lock-in factor and it affects small scale and middle scale companies to enter into the cloud market. 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 [3] Open Virtualization Format (OVF), is a first step towards hypervisor independence thus achieving Cloud portability. OVF format standardizes the use of storage container that stores metadata of virtual machine and enables the migration of virtual machine. But the lack of property factor with this standard is that it is an offline method of migrating the stored data from one cloud Service Provider to another. As the user is not satisfied with performance of the current cloud provider, he/she wants to port or migrate their storage data to another cloud provider.

The whole procedure of data migration between Cloud Storage Provider 1 and Cloud Storage Provider 2. There are three kinds of entities which involve in the process.

User:
User is the authenticated person who wants to migrate their Storage data and sends the migration request to its storage service provider.

Central node:
The central node of the current Storage Provider i.e. source cluster checks the authentication of user's command to start the migration task if valid and accept the read request from the 'data node' where the actual data of that user is stored in the source cluster and returns the address of the data. The central node of the target system or where the data is to be migrate, is responsible for processing the write requests from Data Nodes of source Provider.

Data Node:
Data node stores the user's data and process the any kind of data rad and write request from its Central Node.

implementing secure migration process
Our technical prototype is based on subproject CloudSim [4]: 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, Cloudlets, virtual machines, Brokers and resource provisioning components such as common cloud storage architecture. DATACENTERS act as Central Node in the whole procedure and their STORAGE NODES act as Data Nodes in the CloudSim. The data which is to be migrated packed

in the OVF standard format in Hypervisor such as VMware or VirtualBox using OVF Tool. Then these OVF file is uploaded in CloudSim and migrated through one cloud's DataCenter's nodes to another DataCenter's nodes. The results of the current and target cloud provider has been simulated through CloudSim. The security in migration process[5][6] 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 for Storage migration permission of User, the SLA will request a SSL(Secure Socket Layer) connection between both DataCenters of Cloud Systems. Then these two DataCenters will negotiate for related parameters like temporary session key for message authentication code (MAC) computing, rndm key (randomly generated key) for symmetric encryption and the temporary tickets with minimum migration privilege.
- After getting migration request from cloud provider 1's DataCenter, cloud provider 2's DataCenter will generate a temporary session key (Tmp key) that will be used for communication between the both DataCenter nodes, and generates a random number (Rndm hash) that will be used for double hash computation. Then, the target DataCenter sends
- Temp Key and Rndm hash to the source DataCenter.
- After distributing migration tasks to Nodes, the cloud provider 1's DataCenter sends a request for tickets with the list of IP addresses of Nodes.
- System 2's DataCenter generates a series of tickets and encrypts the tickets by Temp Key, a key only known by cloud provider 2, vand then returns the encrypted tickets **T** to cloud provider 1's DataCenter.
- Where **T=(tickets{IP, Temp Key {ticket(s, ip, attual_filepath)}})**
- After receiving the encrypted tickets, the cloud provider 1's DataCenter distributes the ticket, Temp Key, and Rndm hash to its every Nodes one by one.
- The SSL connection terminates after the distribution of keys. Every Node of cloud provider1's Datacenter encrypts the encypted tickets with a timestamp by Temp Key, and send the Double encypted tickets to cloud provider 2's DataCenter.
- Cloud Provider 2's DataCenter decrypts the tickets and the timestamp is updated to every ticket, and eventually, it will return the address of Cloud Provider 2's Node one by one which the cloud Provider 1's Node sends block to.
- Every node of cloud provider 1 receives the address of every node of Cloud provider 2. 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 Rndm hash. Then, Cloud provider 2's Node sends the three parts to Cloud Provider 2's Node.

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 migration between inter-cloud control is implemented in the DataCenters. CloudSim is a secure communication protocols between two Datacenters and between Storage Nodes uses block transmission protection.

Conclusion

This paper measures the essential parameters required to measure the performance of a cloud system. This paper also discuses an insight to the essential aspect of accepting standardization in Cloud computing, independence in cloud IaaS environment and portability using virtualization. Performance of cloud provider is measured by a third party called SLA manager, which checks the performance and values of parameters that has been signed during agreement between both parties users and cloud provider. If the desired or upto its persistent value not achieved then the cloud provider has to pay a proper compensation to the user for their loss of services parameters values. After that the user is independent to move to another cloud provider.

References

[1] Loretta Mastroeni and Maurizio Naldi, "Violation of Service Availability Targets in Service Level Agreements, " in *Proceedings of the Federated Conference on Computer Science and Information Systems*, 2011, pp. 537-540.

[2] Vmware Inc., "Developer‟s Guide to Building vApps and Virtual Appliances", Palo Alto, CA, User Manual EN-000831-00, 2012.

[3] DMTF, "Open Virtualization Format v.1.0, " DMTF, White Paper DSP 2017 v1.0.0, 2007.

[4] Rodrigo N. Calheiros, Rajiv Ranjan, Anton Beloglazov, Cesar A. F. De Rose, and Rajkumar Buyya. Cloudsim: a toolkit for modelling and

[5] 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] 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.

[7] 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.