**STAR Cloud Migration - SECURE DATA TRANSFER CLOUD MIGRATION ARCHITECTURE**

Shanthosh Priyanka S1, Jothilakshmi R2

Assistant Professor, GGR College of Engineering, Vellore1

Assistant Professor (Senior), SAS, VIT University Vellore2

[shanthoshpriyankavit@gmail.com](mailto:shanthoshpriyankavit@gmail.com)

**ABSTRACT**

The main principle of the cloud computing is to give provision for the stake holders to compute resources via network. One of the principle deterrents blocking more extensive selection of capacity cloud benefits in seller lock-in, a circumstance cloud benefits in which a lot of information that are put in one stockpiling framework cannot be moved to an alternate merchant. To keep this circumstance we introduce a progressed secure **Data Transfer Layer (DTL),** empowering a cloud to add an exceptional organizational cloud layer to effectively import information from another cloud layer with an included layer of security confirmation for secure transmission of information and to guarantee the customers getting to the information are solid clients. Proposed architecture gives a secure architecture and show execution as a feature of a virtual cloud server, which is a vast storage cloud intended for substance driven information. Proposed architecture framework is fit to coordinate storage data from different clouds, with an extra layer of security to evade the outsider assaulting towards the Data Transfer Layer. The clients can access the information through the DTL by beating a security level. At last we examine the exchange of information from old cloud provider to new cloud provider without any attack to the information and leakage of data.

**1 INTRODUCTION**

Cloud platform should fulfill the requirements specified by the stake holders. Many enterprises are opting cloud platform for their IT operations since it has virtualization technologies. Cloud computing takes the enterprise level and allows them in reduction of cost through utilization of vast resources. Reduced administrations through reduction of manual human works. Higher infrastructures and faster deployment of the cloud cycle. Cloud computing can be defined as the combination of platform and applications. Cloud computing has the added advantage of very large data centers and powerful servers. Though the cloud computing technology has many advantages, there are many issues to be sorted to overwhelm the development of the cloud computing technology in a wider path. One among the major ten issues of cloud computing technology is **Vendor lock-in.** The vendor lock-in is defined as the struck into the single cloud provider. The proposed work delivers an enhanced architecture to migrate data from one cloud provider to another cloud provider with an additional security layer to avoid the data loss and the third party intrusion.

When migrating huge amount of data from one cloud provider to another there are many challenging issues such as:-

* Data loss
* Data leakage
* Third party intrusion
* Stealing of data

Distributed computing (Cloud) has ended up overwhelmingly prominent. It gives a for all intents and purposes boundless measure of computational, framework and information assets to its clients at an open expense, as no innovation has ever done in the recent past. Its fame has additionally developed because of the considerable potential outcomes that it accommodates out sourcing support errands, consequently permitting associations to focus on their centre skills.

The principal of them is the nonappearance of institutionalization. Cloud suppliers supply distributed computing administrations which have distinctive determinations starting with one supplier then onto the next. As a result of this, cloud clients get to be reliant on a merchant's administrations and are not able to change to an alternate seller without undertaking significant expenses. This circumstance is known as 'seller lock-in'. Current cloud administrations are given as an one-size-fits-all arrangement, as a rule with preconfigured and solid IaaS/PaaS/SaaS blends. The seller lock-in turns into a ton more apparent when certain SaaS applications are assembled to particular stages or bases (e.g. .Net applications must be facilitated by Windows/Azure based situations). More remote than this, the SaaS arrangements gave by cloud merchants frequently don't conform to the client's prerequisites (e.g. a few SaaS arrangements don't satisfy the QoS levels or information form prerequisites requested by clients because of incompatibilities with the fundamental foundation/stage).

**1.2 PROBLEM DEFINITION**

Huge amount of data becomes the main concern for the data migration. Though there is a solution for the data migration, security becomes a major hurdle to migrate the data securely over cloud environment. The data’s without any encryption strategies cannot be sent as it is over the cloud network. There are many security threats by the third party intruders while migration of the data, meanwhile either migration or working of the business environment cannot be stopped. Hence the problem definition states the following issues:

* Data migration is one of the biggest hurdles in cloud computing environment
* Time and cost makes the cloud migration a bigger issue
* Working of the business environment cannot be stopped due to the data migration
* If there is a solution for the cloud migration, security becomes a major hurdle again
* Data’s cannot be passed over migrating environment without any encryption process
* There is in need of security layers in the data migration process
* Ensuring the migration without any data loss is the biggest problem in the cloud business environment
* Due to these reasons cloud migration is impossible

**2. LITERATURE SURVEY**

**2.1 RECENT APPROACHES**

In recent years many researchers have come up with many different forms ideas to overcome the issue of vendor lock in, some of which are discussed below:

**“Data On-boarding in Federated Storage Clouds by Gil Vernik, Alexandra Shulman-Peleg, Sebastian Dippl, Ciro Formisano, Michael C. Jaeger, Elliot K. Kolodner, Massimo Villari (IEEE – 2013)”**

The system is capable of integrating storage data from various clouds. Once the setup is done the user can access the data immediately from the new cloud provider. VISION cloud is used. There is no need of any modifications in the old cloud. Linked containers are presented one in old cloud and another in new cloud. Federation triggering, direct execution and unified view. Federation background execution. Interoperability with other cloud.

**“Towards Portability and Interoperability Support in Middleware for Hybrid Clouds by Ansar Rafique, Stefan Walraven, Bert Lagaisse, Tom Desair, and Wouter Joosen (IEEE – 2013)”**

Vendor lock-in can be avoided when multiple external clouds are supported and effectively exploited. Provides middleware platform for the hybrid cloud. Abstraction layer is created for the core component, the portability component, the interoperability component. Created API provides the portability and interoperability. BLOB storage is created with No SQL

**“Exploring Models and Mechanisms for Exchanging Resources in a Federated Cloud Uniform API for common PaaS Services by Ioan Petri, Tom Beach, Mengsong Zou, Javier Diaz-Montes, Omer Rana and Manish Parashar (IEEE – 2013)”**

Federation cloud system can overcome the cloud migration. CometCloud provides an overlay that enables multiple type of cloud systems to be federated through special gateway. Local infrastructure to be connected to the global Market. Workload is reduced. Federation management space is created to function the overall transmission.

**“Winds of Change:From Vendor Lock-In to the Meta Cloud by Benjamin Satzger, Waldemar Hummer, Christian Inzinger, Philipp Leitner, and Schahram Dustdar (IEEE – 2013)”**

A new concept namely metacloud is introduced as a solution for cloud migration. This provides user to select the right cloud services for a particular use case support an applications initial deployment and runtime migration. Inside the meta cloud, metacloud API is formed which provides an unified programming interface to abstract from the differences among provider API implementations. Resource templates is generated to describe the cloud service necessary to run an application using necessary templates. Meta cloud proxy provides proxy objects, which are deployed with the application and run on provisioned cloud resources. Transparent use of cloud computing. Integration problem is sorted out.

**“An Analysis of Vendor Lock-in Problem in Cloud Storage by Seyed Majid Razavian, Hadi Khani, Nasser Yazdani, Fatemeh Ghassemi (IEEE – 2013)”**

Cloud migration is done using erasure coding. Data migration is shown logarithmically. Load balancing is done. Maintaining the load balance in big data is the main concern.

**“A federated CometCloud infrastructure to support resource sharing by Ioan Petri, Tom Beach, Mengsong Zou, Javier Diaz-Montes, Omer Rana and Manish Parashar (IEEE – 2013)”**

Federation cloud system can overcome the issue of vendor lock-in. Proposed work is comet cloud which enables multiple type of cloud system to be federated through the use of gateways. A new proposed framework with comet cloud is generated with specialized gateways. Cloud to be cooperated for possible computing.

**“A Systematic Review of Cloud Lock-in Solutions by Gabriel Costa Silva, Louis M. Rose, Radu Calinescu (IEEE – 2012)”**

This paper is a survey paper which identifies, analyze and classify the existing solutions to the vendor lock-in. There is in need of: Exploiting established solutions from areas that are closely related to cloud computing. Increasing empirical evidence to raise confidence in existing solutions.Addressing the socio technical and business challenges related to cloud lock-in. Clear cut survey on the existing work.

**“Towards model-driven provisioning, deployment, monitoring, and adaptation of multi-cloud systems by Gabriel Costa Silva (PhD Candidate), Louis M. Rose and Radu Calinescu (Supervisors) (IEEE -2011)”**

This paper outlines the plan to address vendor lock-in by applying techniques from the model-driven engineering (MDE). Literature review and future work is proposed. Supporting technical and socio-technical aspects of cloud portability using MDE modeling and model management. Decreasing the cost of reengineering. Providing models, tools and theories to enable the medium-term evaluation of switching cloud. Exploring related search areas to identify candidate technique. Finding a use case to work on. Developing tools and artifacts. Accessing proposed hypothesis.

**“SuperCloud: Economical Cloud Service on Multiple Vendors by Qin Jia, Robbert Van Renesse, and Hakim Weatherspoon (ACM -2013)”**

Proposed work is defined as the “Super Cloud”. This supports applications launched across multiple public and private cloud. Virtual network topologies can be migrated across many heterogeneous environments. The enabling technology of this method is to provide a uniform virtualization layer provided by nested virtualization, Xen-Blanket. The architecture is comprised of five layers with the topmost layer as the super cloud. Xen VM is used to monitor the usage allowance and Virtual monitor.

**“CMotion: A Framework for Migration of Applications into and between Clouds by Tobias Binz, Frank Leymann, David Schumm (IEEE – 2010)”**

A new proposed method namely “Cloud Motion Framework” is coined to migrate composite applications into and between clouds. CMotion assumes that the dependencies of components are modeled explicitly and the components are self-contained. CMotion uses adapters to make previously incompatible technologies able to work together. It processes the complete application stack and generates alternatives to deploy the application, while ensuring it still provides the same functionality. CMotion framework describes the following steps:Prerequisites for applications processed by CMotion Generation of Alternatives. Evaluation and Selection of alternatives. Deployment.

**“Identifying Adaption Needs to Avoid the Vendor Lock-in Effect in the Deployment of Cloud SBAs by Javier Miranda Juan Manuel Murillo, Joaquín and Guillén, Carlos Canal (ACM – 2012)”**

Proposed an alternative solution for the cloud migration – “Software Adoption Technique. Other than two options of cloud migration such as Intermediate layer and Middleware Seeking, a new work namely SA is defined. Different providers have different API and software service platforms. Specific computational entities should be defined to make a right way interaction between the components. Range of service adoptions are analyzed. Desired scenario with adaptors such as: Adaption between the components, Adaption between components and cloud specific services, Adaption using third party components (SaaS).

**“Interfaces for Placement, Migration and Monitoring of Virtual Machines in Federated Clouds by Erik Elmroth and Lars Larsson (IEEE – 2009)”**

Technological neutral interfaces and architectural additions for migrating and monitoring Virtual Machines in cloud migration. This paper provides a basic structure to migrate the data from one cloud provider to another. Federated Cloud migration is introduced. Migration is done with the help of Virtual Machine. Provision for cloud migration using inter and intra site VM migration.

**2.2 EXISTING METHOD**

Cloud migration is done using the data on boarding in federated storage clouds. In the existing method there is a layer that connects between the old cloud service provider and the new cloud service provider, which the business environments are using. The layer is termed as federation layer. This federation layer uses the VISION cloud, where the new data storage is stored. VISION cloud is capable of connecting up to 10 different geographically connected data centers. Thus without affecting the work of Business environment the cloud data migration is done. Existing method gives a clear cut view to transfer the data but there is lack of security in the given methods.

**2.3 PROPOSED METHOD**

Proposed system describes an efficient architecture for the data migration with security as the main concern. Since the data’s passed are highly confidential, security plays a vital role. Proposed framework has a layer between the old cloud and the new cloud provider namely **Data Transfer Layer,** through the **Virtual Cloud Server.** Virtual Cloud Server has the capability to load the balance between the client and the server when huge amount of data are transferred. While the data is transferred from old cloud to new cloud the users can access the data from the new cloud once the DTL is properly connected. Proposed work provides a clear phase architecture, where the intrusion of third party is minimized.

**3 PROPOSED SYSTEM**

**3.1 BASIC DATA MIGRATION ARCHITECTURE**

Services

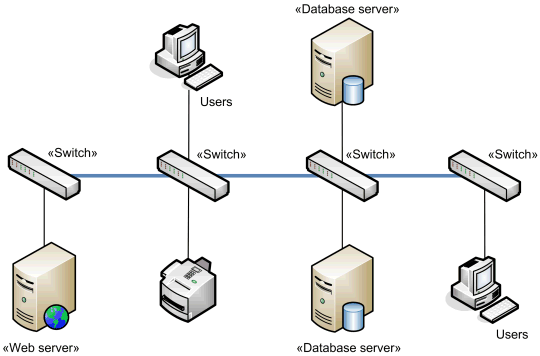
Virtual Desktop

Applications

Storage data

Software Platform

Router



Switch

1. Vendor lock-in, 2. Third Party attack, 3. Data Steal

Cloud Migration

Paas, IaaS, Saas

Vendor Service

Cloud Provider

Appln Server

DB Server

Fig 1: Data Migration Architecture

**3.2 ARCHITECTURE DIAGRAM**

**OLD CLOUD**

**NEW CLOUD**

DATA MIGRATION

**VIRTUAL CLOUD SERVER**

**CLIENT**

**SERVER**

**DANGER MODE**

**NORMAL MODE**

**THIRD PARTY ATTACK**

DATA LINK LAYER

2

6

3

1 7

4

5(B)

Fig.2. Secure Transfer of Data in Cloud Migration 5(A)

Environment

**4 DETAILED DESIGN**

**4.1 FLOW DIAGRAM**

**CLIENT REQUEST**

**INPUT CREDENTIALS**

**SELECT BANDWIDTH**

**ENCRYPT DATA AND SEND**

**AUTHORISED CLIENT**

**UN AUTHORISED REQUEST**

**CHECK LOAD BALANCE AND SEND TO SERVER**

**CHECK MODE AND SEND TO DANGER MODE**

**TESTING IN DANGER MODE**

**UNAUTHORISED REQUEST**

**DETECTION IN NORMAL MODE AND SEND BACK TO CLIENT**

**CLIENT RECEIVES DATA**

**NO**

**NO**

**4.2 TYPES OF DATA**

**Detect table:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **NAME** | **TYPE** | **NULL** | **KEY** | **EXTRA** |
| SourceIP | varchar(50) | No | Primary | Auto\_increment |
| Rtime | varchar(50) | No |  |  |
| Rno | int(10) | No |  |  |
| Ctime | varchar(50) | No |  |  |
| Art | varchar(50) | No |  |  |

**Legitimate profile:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **NAME** | **TYPE** | **NULL** | **DEFAULT** | **EXTRA** |
| ClientName | varchar(50) | No |  |  |
| CIPAddress | varchar(50) | No |  |  |
| Cport | varchar(50) | No |  |  |
| Bandwidth | varchar(50) | No |  |  |
| Sendtime | varchar(50) | No |  |  |
| EST | varchar(50) | No |  |  |
| SecretKey | varchar(50) | No |  |  |

**Pool values :**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **NAME** | **TYPE** | **NULL** | **DEFAULT** | **EXTRA** |
| MessageID | varchar(50) | No |  |  |
| Message | varchar(500) | No |  |  |
| Key | varchar(50) | No |  |  |

**Virtual Server :**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **NAME** | **TYPE** | **NULL** | **DEFAULT** | **EXTRA** |
| Pname | varchar(50) | No |  |  |
| Pip | varchar(50) | No |  |  |
| Pport | varchar(50) | No |  |  |

**Received info :**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **NAME** | **TYPE** | **NULL** | **DEFAULT** | **EXTRA** |
| Rno | Int(10) | Yes |  |  |

**Request info :**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **NAME** | **TYPE** | **NULL** | **DEFAULT** | **EXTRA** |
| ServerName | varchar(50) | Yes |  |  |
| SeverIP | varchar(50) | Yes |  |  |
| Sport | varchar(50) | Yes |  |  |
| Noofrequest | varchar(50) | Yes |  |  |

**Response info :**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **NAME** | **TYPE** | **NULL** | **DEFAULT** | **EXTRA** |
| EID | varchar(50) | Yes |  |  |
| Ename | varchar(50) | Yes |  |  |
| Eaddress | varchar(50) | Yes |  |  |
| EDOB | varchar(50) | Yes |  |  |
| EmailID | varchar(50) | Yes |  |  |
| PhoneNo | varchar(50) | Yes |  |  |
| Position | varchar(50) | Yes |  |  |
| Salary | varchar(50) | Yes |  |  |

**Router info :**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **NAME** | **TYPE** | **NULL** | **DEFAULT** | **EXTRA** |
| Rname | varchar(50) | Yes |  |  |
| Rip | varchar(50) | Yes |  |  |
| Rport | varchar(50) | Yes |  |  |

**Save record :**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **NAME** | **TYPE** | **NULL** | **DEFAULT** | **EXTRA** |
| RNo | Int(11) | No |  | Auto\_increment |
| Name | varchar(50) | Yes |  |  |
| RMessage | varchar(50) | Yes |  |  |
| Rsize | varchar(50) | Yes |  |  |
| Rkey | varchar(50) | Yes |  |  |
| Bandwidth | varchar(50) | Yes |  |  |
| ART | varchar(50) | Yes |  |  |
| SendTime | varchar(50) | Yes |  |  |
| ReceivedTime | varchar(50) | Yes |  |  |
| SourceAddress | varchar(50) | Yes |  |  |
| UserType | varchar(50) | Yes |  |  |

**Server info:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **NAME** | **TYPE** | **NULL** | **DEFAULT** | **EXTRA** |
| Servername | varchar(50) | Yes |  |  |
| ServerIP | varchar(50) | Yes |  |  |
| Port | varchar(50) | Yes |  |  |
| Normalport | varchar(50) | Yes |  |  |
| Dangerport | varchar(50) | Yes |  |  |

**5 IMPLEMENTATION**

**5.1 MODULES**

Proposed framework consist of seven modules such as Client, Server, Normal mode (Server), Danger mode Server (Server), Router, Virtual Cloud Server, Database. This proposed

**MODULE – 1 CLIENT**

The client

**Name:** Client

**Run at:** Client request from Working Place

**Input:**  User Credentials

**Result:**  Request the VCS to provide access to work in the data’s (Cloud migration takes place)

**ALGORITHM**

Step 1: Let ‘A’ be the request sent to the server to access the data.

Step 2: The request ‘A” consists user credentials (Uname/Uid)

Step 3: Let ‘x’ be the input credentials to be selected such as Bandwidth

Step 4: Security plays an vital role hence the given input is encrypted

A(x,k) ----------- (1)

Step 4: Thus the data is transferred to the server with the following input credentials

A (x, k)

Step 5: The input is passing to the server with the parameters such as size and time

A(S,T) ---------(2)

A - Data

X – Bandwidth

K – Key

T- Time

**MODULE – 2 VIRTUAL CLOUD SERVER**

**Name:** Virtual Cloud Server

**Run at:** Intermediate cloud server between old and new cloud provider through DTL

**Input:**  Credentials obtained from Client

**Result:** Manages the load balance and forwards the data to the server

**ALGORITHM**

Step 1: Let ‘M’ be the message received from the client

M(A, E) --------(3)

Step 2: The received data is sent toward load balance

L(A,E) ------------(4)

Step 3: After he load balance process the data is sent to the server

**MODULE 3 - SERVER**

**Name:** Server

**Run at:** Server from new cloud provider

**Input:**  Credentials from VCS

**Result:**  Security Authentication

**ALGORITHM**

Step 1: Receives the input from client towards VCS

M(A,E) -------------(5)

Step 2: Every request to the server is assumed as danger request and forwarded to Danger mode

**MODULE 4 – SERVER DANGER MODE**

**Name:** Server – Danger Mode

**Run at:** Server from new cloud provider

**Input:** Danger request from Server

**Result:** Security Authentication

**ALGORITHM**

Step 1: Danger mode receives the input

M(A,E)

Step 2: Authentication process is done with

M(S,IP,X,T) ----------(6)

Step 3: Testing process takes place with

T ∑ (Sid, IP, X, T) ------(7)

Step 4: If above condition is not satisfied it indicates there is an attack by intruder

**MODULE 5 – SERVER NORMAL MODE**

**Name:** Server – Normal Mode

**Run at:** Server from new cloud provider

**Input:** Normal mode request from Danger mode

**Result:** Security Authentication

**ALGORITHM**

Step 1: Normal mode receives the input

M(A,E)

Step 2: The message is detected with the source IP address and again decrypted

M(S,IP,X,T) ------(8)

Step 4: Once the entire process is done the response message is generated

The proposed work consists of 5 modules. Once client sends request to the Server, the request reaches to VCS where the load balancing is done. Further the request is sent to the Server.

Each and every request is considered as danger request and sent to the danger mode. In danger mode verification process is done and sent to the normal mode.

Normal mode encrypts data with the key and sends back to the VCS, whereas VCS again forwards the acceptance response to the client.

This proposed work enhances security to the system and when intruder is attacking from the same client details access is denied.

**6 CONCLUSIONS**

Proposed STAR cloud migration provides efficient and secure data migration architecture. Attack from the third party users is prevented. Accessing data from the Data Link Layer is protected. Data migration with security is maintained and implemented. This architecture provides a eminent pathway for security and fast migration of data. This design proposes an efficient solution for vendor lock in with security as a main concern.

**REFERENCES**

1. Erik Elmroth and Lars Larsson, “Interfaces for Placement, Migration and Monitoring of Virtual Machines in Federated Clouds”, IEEE 2009
2. Javier Miranda Juan Manuel Murillo, Joaquín and Guillén, Carlos Canal, “Identifying Adaption Needs to Avoid the Vendor Lock-in Effect in the Deployment of Cloud SBAs”, ACM 2012
3. Tobias Binz, Frank Leymann, David Schumm, “CMotion: A Framework for Migration of Applications into and between Clouds”, IEEE 2010
4. Qin Jia, Robbert Van Renesse, and Hakim Weatherspoon, SuperCloud: Economical Cloud Service on Multiple Vendors, ACM 2013
5. Gabriel Costa Silva (PhD Candidate), Louis M. Rose and Radu Calinescu (Supervisors), “Towards model-driven provisioning, deployment,monitoring, and adaptation of multi-cloud systems”, IEEE 2011
6. Gabriel Costa Silva, Louis M. Rose, Radu Calinescu, “A Systematic Review of Cloud Lock-in Solutions”, IEEE 2012
7. Ioan Petri, Tom Beach, Mengsong Zou, Javier Diaz-Montes, Omer Rana and Manish Parashar, “A federated CometCloud infrastructure to support resource sharing”, IEEE 2013
8. Seyed Majid Razavian, Hadi Khani, Nasser Yazdani, Fatemeh Ghassemi, “An Analysis of Vendor Lock-in Problem in Cloud Storage”, IEEE 2013
9. Benjamin Satzger, Waldemar Hummer, Christian Inzinger, Philipp Leitner, and Schahram DustdaR, “Winds of Change:From Vendor Lock-Into the Meta Cloud”, IEEE 2013
10. Gil Vernik, Alexandra Shulman-Peleg, Sebastian Dippl, Ciro Formisano, Michael C. Jaeger,

Elliot K. Kolodner, Massimo Villari, IEEE 2013