Unit-2: Migrating into cloud

Here starts the lesson!

Contents

- Broad Approaches to Migrating into the Cloud
- The Seven-Step Model of Migration into a Cloud
  VM Migration
- Cloud Middleware and Best Practices
- Concept and Need of Cloud Middleware
- QoS Issues in Cloud
- Data Migration and Streaming in Cloud
- Interoperability

Approaches to migrating

- It refers to the movement or transfer between different physical machines without any discontinuity.
- What challenges and obstacles clients might have to overcome to tap into the cloud?

When you migrate from a client to the cloud, the issues you will face fall into the following overall categories.

- Data loss
- Wasted costs
- Added latency
- Lack of visibility & control
- No clear cloud migration strategy in place

Security

Security

Security is an obvious threshold question, if the cloud is not secure, enterprises will not consider migrating to it fearing their sensitive data will be tampered. For eg Users must ensure that they understand the underlying infrastructure of the cloud to which they migrate from their clients and must also advise clients to include security in their cloud SLAs and terms of service.

Vendor Management

When the user is going to migrate with the outsource providers, then the service level agreements and its terms are thoroughly checked.

Challenges in cloud migration

1. Financial cost:-

Rewriting application,
Training users on new system,
And performance issues.

2. Adoption resistance

People tend to resist change

3. Skill Shortage

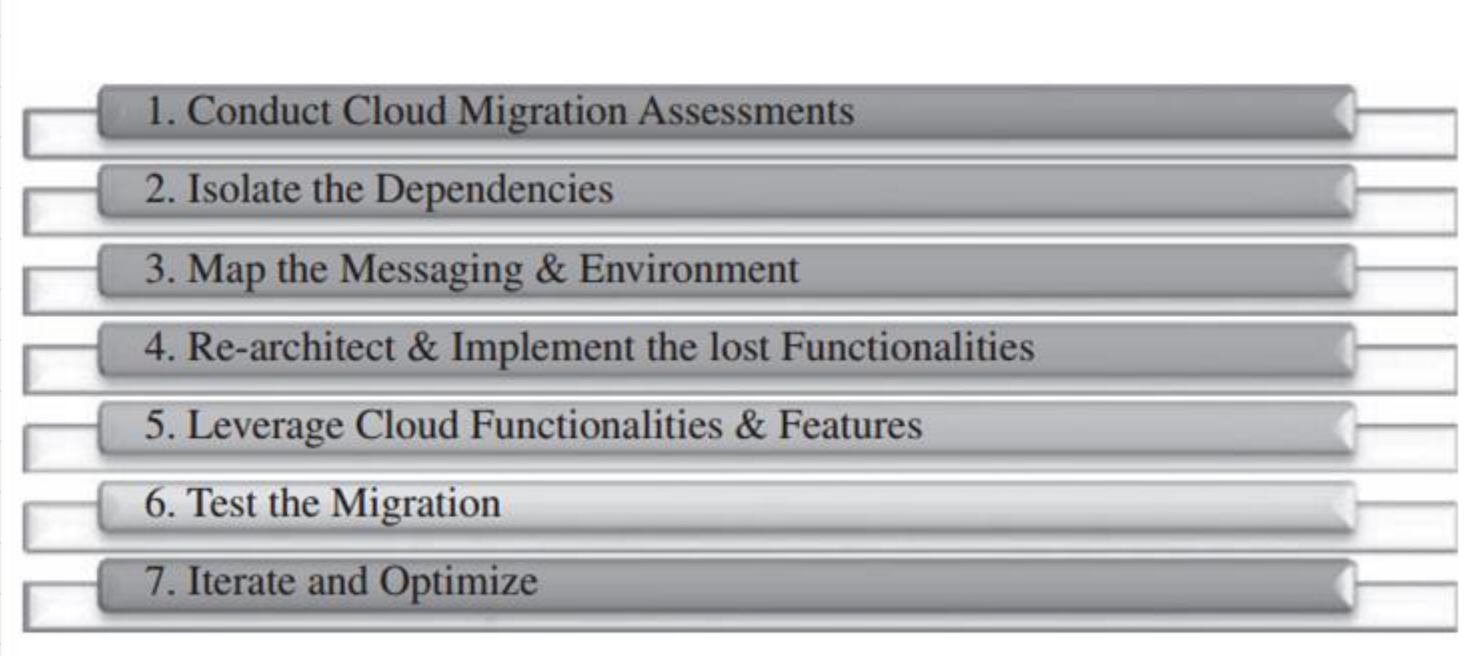
Finding people who have skills

IT and business stakeholders must work together and have to:

- Clearly state business objectives for the cloud migration.
- Define project scope of the cloud migration.
- Provide a set of guiding principles for all to follow.

Seven step migration model

Steps to perform suitable migration



1. Conduct Cloud Migration Assessments
2. Isolate the Dependencies
3. Map the Messaging & Environment
4. Re-architect & Implement the lost Functionalities
5. Leverage Cloud Functionalities & Features
6. Test the Migration
7. Iterate and Optimize

Step 1

Cloud migration assessments comprise assessments to understand the issues involved in the specific case of migration at the application level or the code, the design, the architecture, or usage levels.

These assessments are about the cost of migration as well as about the ROI that can be achieved in the case of production version.

Step 2

Isolating all systemic and environmental dependencies of the enterprise application components within the captive data center

Step 3

Generating the mapping constructs between what shall possibly remain in the local captive data center and what goes onto the cloud.

Step 4

substantial part of the enterprise application needs to be rearchitected, redesigned, and reimplemented on the cloud.

Step 5

We leverage the intrinsic features of the cloud computing service to augment our enterprise application in its own small ways.

Step 6

we validate and test the new form of the enterprise application with an extensive test suite that comprises testing the components of the enterprise application on the cloud as well

Step 7

Test results could be positive or mixed.

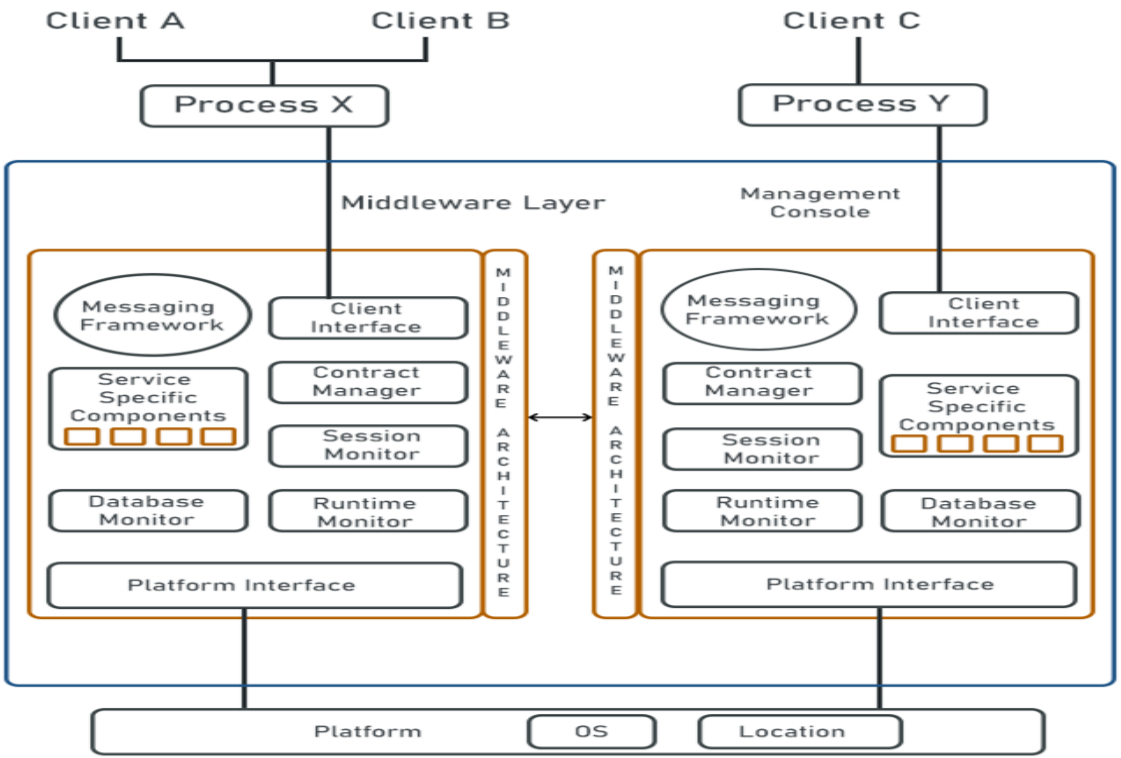
In the latter case, we iterate and optimize as appropriate. After several such optimizing iterations, the migration is deemed successful

Cloud middleware

- It provides services to software applications beyond those available from the operating system.
- Middleware essentially functioning as hidden translation layer. It is also called "glue" because it connect different software platforms and devices together.

- It bridges the gap between applications tools & databases in order to provide unified services to the users.
- Middleware is sometimes called "plumbing" because it connects two application and passes data between them. Allows data contained in one database to be accessed through another.

MIDDLEWARE ARCHITECTURE



Definition of Concepts

Most middleware follows the service-oriented architecture (SOA) design or is designed as a platform-as-a-service (PaaS) solution. SOA is an architectural style that tries to achieve loosely coupled software applications that interact among themselves to run as a whole.

It is adopted by organizations trying to decouple all their business units, depending on integration and reusability for daily operations. SOA allows organizations to use existing application and system investments.

Each of these components must be able to interact with one another and other parts of the system. Apart from some basic components, each type of middleware needs a specific component. For example, a database middleware needs a database manager component.

Components in middleware

- **Middleware management console**

This console provides an transactions, configuration management, and contract rules.

- **Platform interface**

Middleware needs to work across multiple platforms, irrespective of where it resides. This is the interface that is in direct contact with the backend servers.

- **Common messaging framework**

Middleware requires messaging services to communicate with services, applications, and platforms. Most of these frameworks rely on existing standards such as simple object access protocol (SOAP), representational state transfer (REST), or Javascript object notation (JSON).

Need of cloud middleware

Configure and control connections and integrations

Based on information in a client or front-end application request, middleware can customize the response from the back-end application or service. In a retailer's ecommerce application, middleware application logic can sort product search results from a back-end inventory database by nearest store location, based on the IP address or location information in the HTTP request header.

Secure connections and data transfer

Middleware typically establishes a secure connection from the front-end application to back-end data sources using Transport Layer Security (TSL) or another network security protocol. And it can provide authentication capabilities, challenging front-end application requests for credentials (username and password) or digital certificates.

Manage traffic dynamically across distributed systems

When application traffic spikes, enterprise middleware can scale to distribute client requests across multiple servers, on premises or in the cloud. And concurrent processing capabilities can prevent problems when multiple clients try to access the same back-end data source simultaneously.

Quality of service-issues in Cloud

Quality of Service refers to the ability of networks to attain maximum bandwidth and handle other network elements like latency, error rate and uptime. Quality of Service include the management of other networks resource by allocating priorities to specific type of data (audio, video and file).

It is a challenge to implement QoS in cloud computing applications. There are many techniques to provide quality of service to the cloud applications. Scheduling, admission control and dynamic resource provisioning are some techniques used to achieve that goal.

Challenges

Scheduling:

Cloud service scheduling categorized into two categories: user level and system level. At user level scheduling deals with problems raised by service providing between both service provider and customer. Market based and auction based schedulers are fit for ruling the supply and demand of cloud resources. Market based resource allocation is powerful in cloud computing environment where resources are handed over to user as a service. The system level scheduling handles with resource management in datacenter. Datacenter contain many physical machines, Million request sent from user’s side, scheduling these requests to the physical machines done in datacenter. This scheduling affect the performance of datacenter. Service provisioning in cloud systems based on Service Level Agreement (SLA). SLA is the contract between service provider and customer mentioning the terms of agreement including the nonfunctional requirement represented as QoS.

Challenges

Admission Control: The main purpose of admission control is to provide strong performance. At admission control time, the Infrastructure Provider (IP) must consider the extra requirement along with the fundamental computational and networking necessities that may be required to be added to runtime so it become flexible. In many cases, these flexible requirements may be very large comparing it to the normal requirements. For example, if there are many users are working on cloud application with high divergence, the number of virtual machines are required more and that may be added at runtime many times multiple of the number of the basic ones. So that, the number of flexible requirements plays important role in the total requirements and therefore the cost of hosting the service

Challenges

Resource provisioning:

Dynamic resource provisioning is the process of assigning available resources to the cloud application. Resource allocation will make services suffer if the allocation not managed in the right way. Resource provisioning will solve this problem by allowing the service providers to manage the resources of modules individually. Resource Allocation Strategy (RAS) is all about integrating service provider services activities to allocate insufficient resources within the limit of cloud environment so that it meets the needs of the cloud application. It need the demand and type of resources for each application to complete the user task.

Cloud Interoperability

Cloud interoperability. This term refers to the ability of two or more systems or applications to exchange information and to use the information that has been exchanged together.

There is a strong need for the development of integrated interoperability authentication among all provider

Standards

When consumer wishes to migrate from one cloud Provider to another, interoperability falls into these categories:

- Data and Application Portability: It means by running applications and data, consumers should be able to migrate easily from one cloud provider to another without any lock-in issue.
- Platform Portability: It means application development environment or IDE (Integrated Development Environment) should be capable enough to run over any type of cloud infrastructure.
- Infrastructure Portability: It means virtual server or machine images should have the freedom of portability. They should be able to migrate from one cloud provider to another.