Service-oriented cloud computing

A Survey on Cloud Computing

What is cloud computing?

“Cloud computing is the on-demand availability of computer system resources, such as data storage (cloud storage) and computing power, without direct active management by the user .”

Wikipedia

Cloud computing is a computing infrastructure and it is used for rapid delivery to computer resources as a utility in a dynamically scalable, virtualize manner.

The advantages of cloud computing are agility, lower entry cost, device independence, location independence, and scalability.

A Hierarchical View of Cloud Computing

There are four hierarchical of cloud computing which is Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS) as shown in figure 1.

Figure 1

Datacenter

This is the foundation of cloud computing which provides the hardware the clouds run on. Data centers are usually built in less populated areas with cheaper energy rates and a lower probability of natural disasters.

Infrastructure as a service

This layer is built on top of the data centers layer. Infrastructure as a service (IaaS) is online services that provide high-level APIs used to dereference various low-level details of underlying network infrastructure like physical computing resources, location, data partitioning, scaling, security, backup, etc. Users can scale up and down these computing resources on demand dynamically. DigitalOcean, Linode, Rackspace, Amazon Web Services (AWS), Cisco Metapod, Microsoft Azure, Google Compute Engine (GCE) are some popular examples of Iaas.

Platform as a Service

Platform as a Service is a category of cloud computing services that provides a platform allowing customers to assist application design, development, testing, deployment, monitoring, hosting on the cloud without the complexity of building and maintaining the infrastructure typically associated with developing and launching an app. It usually requires no software download or installation and supports geographically distributed teams to work on projects collaboratively.

PaaS Examples: AWS Elastic Beanstalk, Heroku, Windows Azure (mostly used as PaaS), Force.com, OpenShift, Apache Stratos, Magento Commerce Cloud.

Software as a Service

In SaaS, Software is presented to the end-users as services on demand, usually in a browser. It saves the users from the troubles of software deployment and maintenance. The software is often shared by multiple tenants, automatically updated from the clouds, and no additional license needs to be purchased. Features can be requested on-demand, and are rolled out more frequently. Because of its service characteristics, SaaS can often be easily integrated with other mashup applications.

SaaS examples: Dropbox.,Salesforce.,Cisco WebEx.,SAP Concur.,GoToMeeting.

Issues with Current Clouds

Users are often tied with one cloud provider

Even though the up-front cost for a cloud computing deployment is reduced and a long-term lease is eliminated, much effort and money are spent on developing the application for a specific cloud platform which makes it difficult to migrate the same application onto a different cloud. Often, migration simply may mean redevelopment

Computing components are tightly coupled

The current cloud implementations do not allow loose coupling of service components.

Lack of SLA supports

Currently, SLA is an obstacle that prevents wide adoption for cloud computing. Cloud computing infrastructure services such as EC2 are not yet able to sign the SLA needed by companies that want to use cloud computing for serious business deployment. Moreover, business is dynamic. Static SLA is not able to adapt to the changes in business needs as cloud computing promises to.

Lack of Flexibility for User Interface

UI is an important part of the application, and user experience can be a major evaluation factor for a business application. However, cloud/SaaS users are limited with UI choices because UI composition frameworks, such as the one proposed, have not been integrated with cloud computing.

Service-Oriented and Cloud Computing

SOA (Service Oriented Architecture) is built on computer engineering approaches that offer an architectural advancement towards enterprise systems. It describes a standard method for requesting services from distributed components and after that, the results or outcome is managed. The primary focus of this service-oriented approach is on the characteristics of service interface and predictable service behavior. Web Services means a set or combination of industry standards collectively labeled as one. SOA provides a translation and management layer within the cloud architecture that removes the barrier for cloud clients obtaining desired services. Multiple networking and messaging protocols can be written using SOA's client and components and can be used to communicate with each other. SOA provides access to reusable Web services over a TCP/IP network, which makes this an important topic to cloud computing going forward.

How SOA support for overcome issues of current cloud computing

When a business decides to migrate into cloud computing, some decisions must be considered. However, if that business has already employed Service-oriented Architecture, most of these significant issues are automatically solved. An advantage of incorporating Service-oriented architecture is that you can have a display of horizontal services that run through your entire organization. It is this horizontal display that comes to mind when businesses contemplate virtualizing their IT environment.

Specifically, When Service Oriented architecture comes into the picture, services that were already built can be recycled, repurposed, redistributed, and making them rapidly available throughout the company's network. How does this apply to cloud computing? SOA's ability to reuse services means that cloud costs can be decreased and improve more agility. so that, this allows increasing companies' scalability.

Otherwise, incorporating Service-oriented architecture lets companies the option of deploying a software-as-a-service in their cloud platform. Meantime, the cloud system pumps out processing power as demanded. This relationship between cloud computing and service-oriented architecture lets for greater flexibility and robustness, providing ample time to meet operational needs.

Service-Oriented Architecture lays down a framework that simplifies the management of information technology (IT) systems. Service-Oriented Architecture employs mechanisms that allow IT systems to work together cohesively within one enterprise cloud platform.

For the strongest cloud computing platform, a business needs interfaces and architectures that can reach cloud resources. This can not be achieved by applying links between cloud computing resources and core enterprise information systems. But, when we use Service-oriented architecture which provides the ability to look at business systems as an entire set of services. When we combine this with the cloud system, we can extend this to cloud resources.

Layered architecture of SOCCA

- *Individual cloud provider layer*

This layer resembles the current cloud implementation. Each cloud may have its own proprietary virtualization technology or utilize open source virtualization technology, such as Eucalyptus. Similar to Market-Oriented Cloud Architecture proposed in, within each individual cloud, there is a request dispatcher working with Virtual Machine Monitor and Service/App Governance Service to allocate the requests to the available resources. The distinction from current cloud implementations is that the cloud computing resources in SOCCA are componentized into independent services such as Storage Service, Computing Service and Communication Service, with open-standardized interfaces.

- *Cloud broker layer*

Cloud brokers serve as the agents between individual cloud providers and SOA layer. Each important cloud service hs an associated service broker type.

- *SOA layer*

This layer fully takes the advantages of the existing research and infrastructure from traditional SOA. Many existing SOA frameworks, such as CCSOA, UCSOA, GSE and UISOA can be integrated into this layer. The fundamental difference of the SOA layer of SOCCA from traditional SOA is that the service providers no longer host the published services anymore. Instead, they publish the services in deployable packages, which can be easily replicated and redeployed to different cloud hosting environments.

Application development on SOCCA

Application development on SOCCA is similar to the development on CCSOA.If there is a workflow template that matches the requirement. A workflow template is composed of service specifications which specify the functionalities and interfaces of services.

Service Package

Service providers of traditional SOA develop the logic of a service and provide its running environment. In SOCCA, services are published as re-deployable packages, namely service package. A service package contains the following information and files

Compiled code:

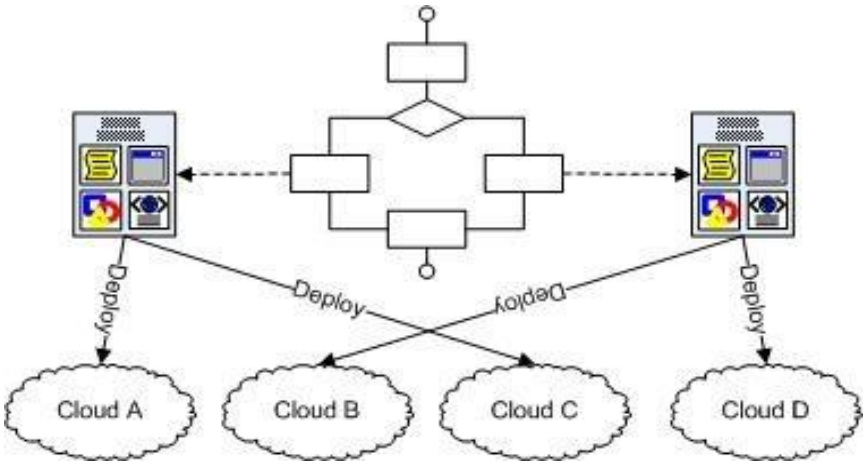
If service providers only use the standard APIs and protocols, a single version of complied code is enough; if service providers optimize the performance of their services by utilizing some platform unique APIs and features, complied code for each platform is needed.

Source code :

This is optional. It is useful to help its user to understand the service better, also gives the freedom to its users to tweak the services to accommodate their specific requirement.

Configuration file :

Services might use external basic services.



Application architecture on SOCCA

Advantages of SOCCA

- A new multi tenancy pattern single instance and multiple service instance becomes possible with SOCCA.
- SOCCA allows application to run on other cloud and interoperate with each other.
- SOCCA support easy migration of application from one cloud to another.
- SOCCA provides better support for Multi tenancy feature.

Disadvantages of SOCCA

- SOCCA does not provide any modeling language to map different language sets as different clouds support language.
- Instances for the same service cannot live on multiple clouds.
- SOCCA does not provide any modeling language to map different language sets as different clouds supports different language sets.
- Instance for the same service cannot live on multiple clouds.

References

- Tsai, Wei-Tek, Xin Sun, and Janaka Balasooriya. "Service-oriented cloud computing architecture." Information Technology: New Generations (ITNG), 2010 Seventh International Conference on. IEEE, 2010.
- Wikipedia - Cloud Computing. [Online]. http://en.wikipedia.org/wiki/Cloud_computing
- Carnalsoftware- The Role of Service-Oriented Architecture (SOA) in Cloud Computing [Online] https://www.carnalsoftware.com/cloud/service-oriented-architecture.html
- Cloud computing: G Boss, P Malladi, D Quan, L Legregni, H Hall - IBM white paper, 2007 - files.spogel.com.
- w3schools-service-oriented-architecture[Online] https://www.w3schools.in/service-oriented-architecture/
- Mark Chang, Jackson He, W.T. Tsai, Bingnan Xiao, and Yinong Chen, "UCSOA: User-Centric Service-Oriented Architecture," in *IEEE International Conference on e-Business Engineering*, 2006, pp. 248-255.