ISSN:0975-9646

Automatic Generation of Platforms in Cloud Computing

Mathews Abraham
Postgradution student
Department of Information Technology
Rajagiri college of Engineering and Technology

Kuttyamma A.J
Department of Information Technology
Rajagiri college of Engineering and Technology

Abstract— **Cloud computing that provides cheap and pay-as-you-go computing resources is rapidly gaining momentum as an alternative to traditional IT Infrastructure. As more and more consumers delegate their tasks to cloud providers, Service Level Agreements(SLA) between consumers and providers emerge as a key aspect. Due to the dynamic nature of the cloud, continuous monitoring on Quality of Service (QoS) attributes is necessary to enforce SLAs. Also numerous other factors such as trust (on the cloud provider) come into consideration, particularly for enterprise customers that may outsource its critical data. This complex nature of the cloud landscape warrants a sophisticated means of managing SLAs. This paper proposes a mechanism for managing SLAs The use of concepts of Service Level Agreements - SLA to control the utilization of computational resources of a provider is common in computing environments and can be used to ensure the quality of services that being offered.**

Index Terms— Cloud Computing; IaaS; PaaS; SaaS; SLA; Models.

I. Introduction

Cloud computing paradigm is gaining increasing attentions recently because it brings many economic benefits to users. Cloud computing can reduce capital expenditures, such as hardware costs and software license costs, and it also shrinksoperational expenditures such as costs of hiring IT personnel etc. Moreover, users will have universal data access and storage at multiple independent geographical locations

The objective of this work is to combine these paradigm. A major challenge in the environment of clouds is efficiently control the use of computational resources. So the use of SLA, can clearly define the responsibility of each part, and define the requirements that are being hired, to be able to check the contracts and resource efficiency Service Level Agreements(SLA) between consumers and providers emerge as a key aspect. Due to the dynamic nature of the cloud, continuous monitoring on Quality of Service (QoS) attributes is necessary to enforce SLAs.

Cloud computing services promise elastic computing and storage resources in a pay-as-you-go way. A SLA between a cloud service provider (CSP) and a user is a contract which specifies the resources and performances that the cloud should offer. However, the CSP has the incentive to cheat on SLA, e.g., providing users with less CPU and memory resources than that specified in the SLA, which allows the CSP to support more users and make more profits. A malicious CSP can disrupt existing SLA monitoring/verification techniques by interfering the monitoring/measurement process**.** This proposal differs from other existing services.

II. Cloud Computing

Cloud is defined as both the applications delivered as services over the Internet and the hardware and systems software in the data centres that provide those services. As consumers move towards adopting such a Service-Oriented Architecture, the quality and reliability of the services become important aspects. SLA to control the utilization of computational resources of a provider is common in computing environments and can be used to ensure the quality of services that being offered.

The cloud also focuses on maximizing the effectiveness of the shared resources. Cloud resources are usually not only shared by multiple users but are also dynamically reallocated per demand. This can work for allocating resources to users. The main enabling technology for cloud computing is virtualization.

Cloud computing adopts concepts from Service Oriented Architecture (SOA) that can help the user break these problems into services that can be integrated to provide a solution

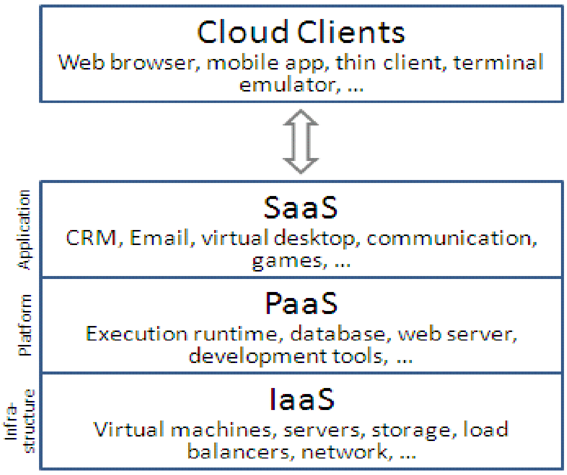
Cloud Clients
Web browser, mobile app, thin client, terminal emulator, ...
SaaS
CRM, Email, virtual desktop, communication, games, ...
PaaS
Execution runtime, database, web server, development tools, ...
IaaS
Virtual machines, servers, storage, load balancers, network, ...
Application
Platform
Infra-structure

Figure 1 – Cloud Architecture

In the PaaS models, cloud providers deliver a computing platform typically including operating system, programming language execution environment, database, and web server. Application developers can develop and run their software solutions on a cloud platform without the cost and complexity of buying and managing the underlying hardware and software layers.

In the SaaS model, cloud providers install and operate application software in the cloud and cloud users access the software from cloud clients. Cloud users do not manage the cloud infrastructure and platform where the application runs. This eliminates the need to install and run the application on the cloud user's own computers, which simplifies maintenance and support.

Cloud computing is client/mission-oriented, specified by Service Level Agreements (SLAs) between cloud service providers (CSPs) and users in a pay-as-you-go way. A SLA is a contract between the two parties, which states the details of the resources and performances that the cloud should offer. The typical SLA metrics include memory size, CPU speed, storage, size etc.

A TPA resolves the trust dilemma between a CSP and its users. There are several benefits of the TPA-based SLA verification framework. First, the TPA framework is flexible and scalable. It supports various types of tests targeting at different SLA metrics (e.g., memory or CPU).

Second, it supports testing from multiple (including a large number of) users, which could significantly enhance the capability of testing a cloud. Third, the TPA framework also relieves users from the verification burden.

III. Service Level Agreements - SLA

Cloud computing uses the concept of Service Level Agreements to control the use of computing resources arising from a provider. Normally, the SLA management strategies are based on an understanding of two distinct phases, where one relates to contract negotiation and the other is focused on real-time monitoring of its execution. Thus, one can say that an SLA management should include the definition of the contract, from a basic scheme with parameters of QoS(Quality of Service), SLA negotiation, monitoring of the SLA and also execution of the SLA, all this according to a policy defined by the parties. Implementing and monitoring SLA also have a supervisory role, with the intention of verifying whether the conditions of the contract for all running services are being met.

1) Resource Optimization

To meet requests, the provider needs a scheduling policy to set the priority of each one, considering how to optimize use of resources and how to preserve the QoS parameters that are guaranteed in the SLA.

2) Monitoring

The operationalization of resources must be monitored and this information will be used to verify that the QoS attributes defined in the SLA are being met.

3) Renegotiation

Since the premise of change will always exist, one of the Contracting Parts may wish to change the resource use policies, usually to conform to some external demand.

4) Accounting

What resources were used, to what extent and how long it should be counted, to serve as the basis for billing, including amounts relating to penalties in case of non-compliance with QoS.

IV. Properties Of SLA

The SLA is a legal format documenting the way that services will be delivered as well as providing a framework for service charges. Service providers use this foundation to optimize their use of infrastructure to meet signed terms of services. Service consumers use the SLA to ensure the level of quality of service they need and to maintain acceptable business models for long term provision of services.

The following are the main requirements of the SLA:

- SLA format should clearly describe a service so that the service consumer can easily understand the operation of the services
- Present the level of performance of service
- Define ways by which the service parameters can be monitored and the format of monitoring reports
- Penalties when service requirements are not met
- Present the business metrics such as billing and stipulate when this service can be terminated without any penalties being incurred

A. Architecture

Usually the architecture of an SLA system is divided into two main steps being considered the stages before the execution and in runtime as most relevant for management.

1) The services are registered by Provider, by setting its parameters to be used in SLA management system.
2) Mapping of QoS's parameters for SLA.
3) Storage of information related to services.
4) Negotiation of SLA contracts between clients and providers of service.

The second phase aims to monitor and control the states of services at runtime, and is responsible for capturing parameters of the application environment and to map them in the corresponding SLA parameters to identify whether there is any violation occurring in the contract. Obtaining service parameters such as time of request and response, time of the service if the service call was successful and whether the services were performed properly. Comparison of SLA parameters computed by means of service parameters at runtime with the SLA metrics from database to verify if there are problems but to also to define the operating parameters of the available resources.

The cloud is inherently dynamic and the resource usage changes dynamically. Hence any system that tries to enforce a SLA need to embrace this dynamic nature. All measuring tasks in a cloud context be performed through functions. Due to the mounting concerns of privacy and data security, consumers may be hesitant to disclose certain details to cloud providers. To identify a set of tasks that can be delegated to trusted third parties to cater for better security.

Cloud services are subjected to load fluctuations and SLA violations are more likely to happen during these transitions. The nature of these fluctuations are unpredictable and hence a static schedule for evaluating conditions The SLAs in the cloud context use a dynamic schedule for condition evaluations.

AUTOMATIC GENERATION OF PLATFORMS IN CLOUDS COMPUTING

In this architecture, assume that the cloud provider and the cloud consumer already participated in the negotiation process and have an agreed set of service parameters, i.e. the negotiation and SLA establishment steps are considered out of scope for this work. Once the SLA document is established, it needs to be deployed. In the term SLA deployment is defined as the process of validating and distributing the SLA, in part or full, to the involved parties. The provider and the consumer may not want to share the complete SLA document with supporting parties due to security considerations .There are three common WSLA services and some of their adaptations required in the cloud context.

The main purpose of this scenario is to provide an environment where, alike an uninitiated user as an experienced user can specify their needs, negotiate the terms of assurance services and their values, and then, automatically have available for use, in few minutes, a virtual environment. This environment will be monitored so that its features can be updated to ensure user satisfaction.

The process of using this scenario consists of four phases, where the customer interacts in phases 1 and 2 due to the need to specify the user's intention, while the phases 3 and 4 are under the responsibility of an automatic process.

The purpose of each phase is described below:

Phase 1

Initially will be available just the operating systems and applications more commons on the market, however systems and applications not originally contemplated may be included later, in accordance with the demand and the user can request any item that is of interest to him.

Phase 2

After the setting platform be held, will be available the terms and guarantees of use of service as well as limits (lower, middle and top) with the specification for the attributes and requirements (cpu, memory, disk, etc) for each item and their rates. This negotiation, with sending proposals of both, can happen indefinitely until to be found a common point and the agreement can be completed. At any time the user can go back to step 1 to change your choices and then have restarted the process of negotiating SLA.

At the end of this stage, considering all the specifications made by the user and their data, will be generated a model environment for this client. This model will be used to create the client platform and allows the verification of QoS parameters and can be used to update the environment when needed.

Phase 3

With the conclusion of the agreement, an automatic process is started to create a virtual machine with the requirements in accordance with the operating system and selected applications necessities to present an adequate performance considering the SLA's specifications.

Phase 4

After the virtual machine becomes available, starts the monitoring process, where quality attributes are monitored. If there is degradation of quality in the functioning of the environment, the model of the environment will be upgraded to meet customer expectations. Issues such as response time, efficiency of bandwidth, storage space, etc. will be considered.

LIFE CYCLE

There are three types of activities that are present in the life cycle of this environment, the Specification, the Negotiation and the Deploying will be utilized for automatic generation of the virtual machine that will be used by costumer.

Measurement Services: These services are responsible for measuring the runtime parameters of cloud provider’s resources. The service parameters like response time, throughput are constantly changed due to variability in service request from consumer side. In the context of the cloud however the usage and cost parameters are also dynamic.

Condition Evaluation Service: This service is responsible of getting the results from measurement services and evaluating the Service Level Objectives. If there are violations the Management service will be contacted.

Management Service: This service is responsible for taking corrective actions on violation of the Service Level Objectives. So that since the cloud represents utility type computing resources, the management service would be primarily handling financial penalties similar to the real world utility industry practices.

1) The CSP has complete controls of its own resources including physical machines,hypervisors, VMs.
2) The CSP is able to know any security material (such as an encryption key) used by a VM, because the material is stored in the physical memory and/or the hard drive that the hypervisor has access to. In addition, the CSP is able to modify any message sent by a VM without being detected. For example, if a VM runs a test and creates a timestamp after the test is completed, the cloud can change the timestamp stealthily, even if a message authentication code (MAC) is used. The hypervisor knows the key for the MAC and it can create a new valid MAC after changing the message.
3) The CSP will perform the cheating only if the cost is less than C, where C is a cost-related parameter. For example, if the CSP needs to reserve a GPU such that it is not detected by a GPU/CPU test, the cost of cheating may be too large for the CSP and it has no incentive to cheat.

PLATFORM AGENTS

1) Agent Negotiator

This agent acts during phases 1 and 2 of Scenario platform. Initially it loads the information about operating systems and applications available in our repository, moreover, the agent also stores information about the minimum configurations needed for the environment has a reasonable performance.

The processing power, memory, bandwidth and storage capacity for the proper functioning of the platform are calculated from the information in the environment's database and serve as input data for the SLA, setting the initial value the contract.

The Negotiator agent also controls the whole process of negotiation of the SLA and is responsible for extracting data from user's registration and keeps the trading environment, where each part send and receives proposals on a given contract. Its activity ends when the parties reach an agreement and the contract is established.

2) Agent Extractor

This agent starts his acting when the SLA is established. It is its responsibility to extract the information that are fundamentals to the Environment Model, therefore, from SLA is created a model that will be used to create the virtual machine. This model contains information about the user, the infrastructure of hardware and software, the guarantees of service and even the parameters of environment quality, too called QoS (Quality of Services) parameters. The extracting agent is also started when is needed to happen an update in environment. If for some reason you have to increase or decrease resources, or add or remove applications this agent reconstructs the environment model so that updates can be performed.

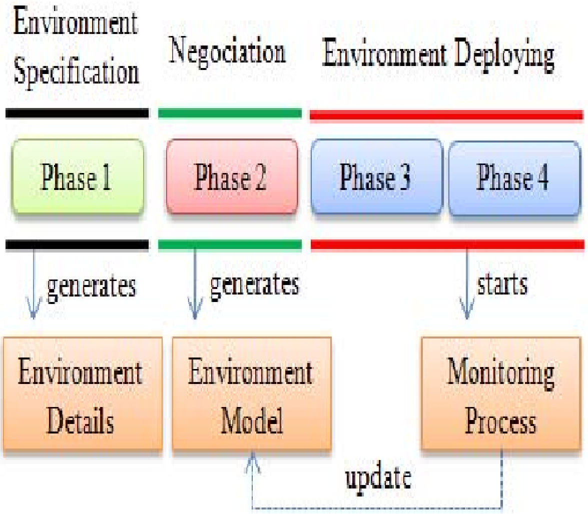

Figure 4 – Life Cycle

The first task of this agent is validating the model created by extracting agent, after model validation their work involves the creation of an appliance and a hardware template for the platform. The Creator Agent builds the appliance platform dynamically and at runtime, based on information contained in the environment model, and the template also is generated from data in the environment model. After completion of the appliance and the template, the Agent Creator uses a virtualizes to effectively create and deliver the virtual machine to the user.

3) Monitor Agent

The activities of the Monitor Agent start when the environment is released to the user to use. From this moment, are observed and stored information about the parameters of environmental quality, the QoS parameters, to be evaluated regularly. If the quality parameters are not being met or are experiencing overloads in the system, the Monitor Agent invokes the Negotiator Agent to offer a possible update of the environment to user, and the values adopted in this policy aims to benefit the customer, intending to reverse the dissatisfaction.

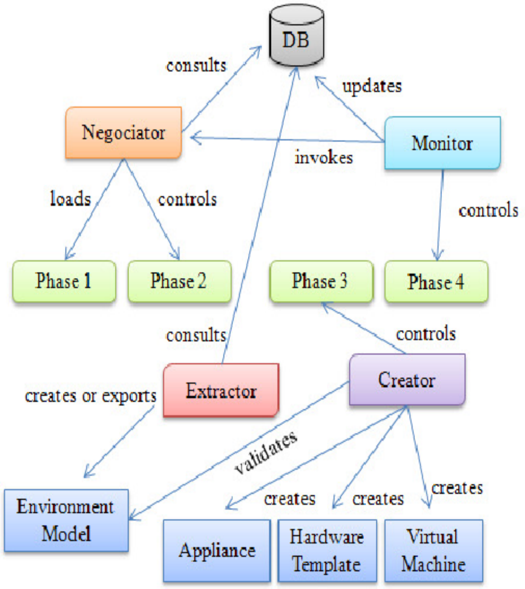

Figure 5 – Environments Agents

Medical treatments simplified: The future of cloud computing is not confined to entertainment and gaming options as it can contribute massively in the fields of medical sciences as well. As most of the contemporary treatment require computer assistance

CONCLUSION

As indicated in this proposed architecture, The need for a clear and formal methodology to handle SLAs in the context of cloud computing .WSLA, which suggests a very flexible architecture for managing SLAs between providers and consumers, seem to be the most suitable candidate. In applying WSLA however, the need for a host of support services arises. Some cloud computing providers may provide these support services but

WSLA does not strictly mandate such provisions and hence third parties can step into provide the necessary services. This as one of the strong points of WSLA where, true to the paradigm of SOA, every functionality is provided as a service that may not necessarily come from the same provider. One important observation we make in the context of clouds is the lack of standardization. This is especially important when attempt to apply monitoring across multiple clouds. Even though it is possible to cater for different cloud interfaces through a middleware, there is no universal set of metrics that can be monitored across cloud providers. With possibly lower costs, the services would be offered at lower rates, reaching new customers and bringing more profit.

Regarding management and updates: new operating systems and applications should be made available transparently to all users simultaneously.

REFERENCES

[1] CARR, N. Big Switch: Rewiring the World, from Edison to Google. Norton & Company, 2008.

[2] SELIC, B. The pragmatics of model-driven development. IEEE Software, v.20, n.5, p. 19-25. 2003.

[3] FOWLER, M. UML Distilled: A Brief Guide to the Standard Object Modelling Language. 3a. edição. Addison-Wesley, 2004.

[4] KLEPPE, A. WARMER, J. BAST, W. MDA Explained – The Model Driven Architecture: Practice an Promise. Addison Wesley.2003.

[5] DAS, C., MOHAN, G., ROY, R., BHATTACHARYA, S . Quovadis, SaaS a system dynamics model based enquiry into the SaaS industry. Information Management and Engineering (ICIME), 2010 The 2nd IEEE International Conference. pp.732737,16-18 April 2010.108,21-25 Sept.2009.

[6] CHONGF; CARRAROG. Architecture Strategies for Catching the Long Tail. Available: <http://www.cistratech.com>

[7] MELLOR, J. MDA Distilled – Principles of Model Driven Architecture. Addison-Wesley. 2004.

[8] MILLER, J., MUKERJI., J. Model Driven Architecture. Architecture Board ORMSC, 2001.

[9] CHI, Y., MOON, J., HACIGUMUS, H., TATEMURA, J. Slatree: a framework for efficiently supporting sla based decisions in cloud computing. In Proceedings of the 14th International Conference on Extending Database Technology,EDBT/ICDT'11,pages129–140, NewYork, NY, USA. ACM 2011.

[10] LIU, F., GUO, W., ZHI, Q., CHOU, W. SaaS Integration for Software Cloud. Cloud Computing (CLOUD), 2010 IEEE 3rd International Conference on vol., no., pp.402-409, 5-10 July2010.

[11] DAWOUD, W., TAKOUNA, I., MEINEL, C. Infrastructure

[12] YAQUB, E., WIEDER, P., KOTSOKALIS, C. Service Level Agreements for Cloud Computing: A Generic Platform for Conducting SLA Negotiations. p. 187-206. Springer New York2011.