ISSN:2231-0711

Minnu Moothedan et al | IJCSET(www.ijcset.net) | January 2016 | Vol 6, Issue 1, 57-61

Survey on SLA for PaaS Clouds

Minnu Moothedan[#1], Christina Joseph[#2]

Computer Science and Engineering Department,

Calicut University, Kerala, India

***Abstract*— Cloud Computing is one of the biggest buzzwords in the field of information technology today. Due to its wide popularity most of the organizations started shifting their business to cloud. Cloud provides flexible and reliable services to its users. However, this new technology has also created many challenges for service providers and customers, especially for those users who already own complicated legacy systems. Service Level Agreements (SLAs) are the means through which the provision of infrastructure, platform, and software services in Cloud Computing is regulated, along with functional and non-functional specifications of services. SLAs are intended to set a framework for the provision of services and for the cooperation between service providers and service consumers. Currently, Cloud SLAs are usually drafted by Cloud providers and do not allow much negotiation.**

***Keywords*— Cloud Computing, Service Level Agreements, PaaS**

I. INTRODUCTION

Cloud computing delivers infrastructure, platform, and software (application) as services, which are made available as subscription-based services in a pay-as-you-go model to consumers. These services in industry are respectively referred to as Infrastructure as a Service (Iaas), Platform as a Service (PaaS), and Software as a Service (SaaS). In a Feb 2009 Berkeley Report, Prof. Patterson et. al. stated Cloud computing, the long-helddream of computing as a utility, has the potential to transform a large part of the IT industry, making software even more at- tractive as a service. Clouds aim to power the next generation data centers by architecting
them as a network of virtual services (hardware, database, user-interface, application logic) so that users are able to access and deploy applications from anywhere in the world on demand at competitive costs depending on users QoS (Quality of Service) requirements . Developers with innovative ideas for new Internet services are no longer required to make large capital outlays in the hardware and software infrastructures to deploy their services or human expense to operate it . It offers significant benefit to IT companies by freeing them from the low level task of setting up basic hardware (servers) and software infrastructures and thus enabling more focus on innovation and creation of business values.

The cloud is just a metaphor for the internet. Cloud computing simply means the method of storing and accessing of data, resources and applications over the internet instead of using one's own computer hard drives. In other words, services are provided by cloud providers which can be used by customers as per their requirements and pay for what they use. Key characteristics of cloud which make it this much popular are: The purpose of the new phase in Web technology is to make the machines capable of understanding the semantics of the information presented on the Web. To be able to read and understand the Web as a human being does.

Cloud Computing comprises of three service models namely Infrastructure-as- a- Service (IaaS), Software-as-a-Service (SaaS) and Platform-as-a-Service (PaaS).

1. Infrastructure as a Service: IaaS offers storage and computing resources which can be used by developers to deliver their application as a service. Servers, storage systems, networking equipment, data centre space etc. are pooled and made available to handle workloads. The customer would typically deploy his own software on the infrastructure. Example: Windows Azure, Amazon EC2.
2. Software as a Service. SaaS providers offers access to applications hosted in cloud without the need to install or manage or buy any hardware for it.In this model, a complete application is offered to the customer, as a service on demand. A single instance of the service runs on the cloud multiple end users are serviced. On the customers side , there is no need for upfront investment in servers or so ftware licenses, while for the provider, the costs are lowered, since only a single application needs to be hosted maintained. Today SaaS is offered by companies such as Google, Salesforce, Microsoft, Zoho, etc.
3. Platform as a Service. PaaS offers computing platforms for the customers to develop their applications on cloud. Here, a layer of software, or development environment is encapsulated offered as a service, upon which other higher levels of service can be built. The customer has the freedom to build his own applications, which run on the provider's infrastructure. To meet manageability and scalability requirements of the applications, PaaS providers offer a predefined combination of OS and application servers, such as LAMP platform (Linux, Apache, MySql and PHP), restricted J2EE, Ruby etc. Google's App Engine, Force.com, etc are some of the popular PaaS examples.

Recently, virtualization has enabled the abstraction of computing resources such that a single physical machine is able to function as multiple logical VMs (Virtual Machines). A key benefit of VMs is the ability to host multiple operating system environments which are completely isolated from one another on the same physical machine. Another benefit is the capability to configure VMs to utilize different partitions of resources on the same physical machine. For example, on a physical machine, one VM can be allocated 10 percentage of the processing power, while another VM can be allocated 20 percentage of the processing power. Hence, VMs can be started and

stopped dynamically to meet the changing demand of resources by users as opposed to limited resources on a physical machine. In particular, VMs may be assigned various resource management policies catering to different user needs and demands to better support the implementation of SLA-oriented resource allocation.

To meet aforementioned requirements of SLA-based resource allocation of Cloud applications, future efforts should focus on design, development, and implementation of software systems and policies based on novel SLA-oriented resource allocation models exclusively designed for data centers.

A. PaaS Clouds

Today, the PaaS market is in the early stages of growth. However, as the technology matures, enterprises are turning to PaaS as a way to broaden general cloud adoption across the organization and to streamline the development process. Gartner forecasts that the global market for PaaS will grow form 1.2 billion in 2012 to more than 2.9 billion in 2016.

While those revenue numbers represent about 1 percent of the projected 131 billion cloud computing market, vendors continue to offer a range of services, from tools and application platforms for developers to services that can be used by business analysts. PaaS provides IT organizations with significant benefits:

1. Improved time to market with minimal capital costs : Developers can accelerate production of new cloud applications through access to a broad set of automated tools and technologies that enable them to design and deploy cloud-aware applications more quickly. Reduced development cycle time enables more new products to reach the market faster. With the ability to start, test, and deploy software projects dramatically reduced, IT also can support limited-duration projects such as marketing campaigns.
2. Access to services that are available only in the cloud : Developers can customize, extend, and integrate software as a service (SaaS) offerings from public cloud providers by selecting specific functionality to be delivered as a service through the PaaS platform. The solution is tailored to user needs so that line-of-business (LOB) managers are less likely to pursue shadow IT initiatives.
3. Ability to rehost or re-architect legacy applications to run in the cloud, often with minimal changes: Porting existing applications can cut IT operational costs, increase agility, broaden reach, and enable developers to focus on core competencies rather than the complexities of legacy infrastructure.
4. Address application integration issues by building cloud- aware applications specifically for dynamic environments: Designing applications that move across environments easily can increase adoption of private cloud technology internally. It also paves the way to a hybrid cloud that effectively integrates both internal and public-hosted resources.

PaaS is a group of services that abstracts application infrastructure, operating system, middleware, and configuration details, and provides developer teams with the ability to provision, develop, build, test, and stage applications. PaaS facilitates application deployment through
self-service, on-demand tools, resources, automation, and a hosted platform runtime container. This eliminates the need for an installation kit, and developers no longer have to configure and wait for physical servers or virtual machines (VMs) or to copy files from one environment to another as they move through the application life cycle. PaaS streamlines life-cycle management, from building the application to removing it at end of life, automating the many steps and functionality associated with each milestone. PaaS can also simplify version updates, patching, and other maintenance activities.

PaaS pushes an application to the cloud from a command-line interface or directly from an interactive development environment (IDE) using a plug-in. After analyzing the application, PaaS hosts it in the runtime container that matches its resource requirements. In addition to scaling capabilities, PaaS also provides high availability, automatic configuration, load balancing, and management tools. PaaS can instantiate multiple copies in the same or multiple clouds
for environments that might need to be isolated from others in the business. This is important for applications that must take into consideration compliance issues or internal-facing versus
external-facing sets of applications. In each of these usages, the developer can still use common tools and best practices, but has a separate, secure environment. With PaaS, companies can also combine local resources and data for personalized mashups for a variety of web services.

B. Service Level Agreements

The SLA management lifecycle aims to establish the SLA contract for two parties of service provider and service consumer involved in the business relationship around certain web service provisioning. For each party, there are different business objectives to be concerned; the consumer expects that they can get the necessary outsourcing function as well as stable service quality to fit the nonfunctional requirements of their application with minimized cost, on the other end, the provider want to get maximized profits through the service provisioning with minimized resource consumed, furthermore, service consumer and service provider may have different preferences or priorities when considering their own business objectives. In between of the two ends, the SLA contract expresses the consensus on different terms with which both two counter parties agreed for the establishment and enactment of the business relationship. To achieve the consensus about service quality, some kind of bilateral negotiation process usually need to be introduced in the SLA contracts establishment. In a bilateral negotiation process, some kind of time series function are needed to produce the offer and counteroffer upon every QoS term for each parties involved in the negotiation, considering the difference of business

objectives and preference, different function should be designed to create counteroffers independently for each party. Additionally, in an open market environment, there may be different user binding to certain service which is provided by different service provider. So the chance of comparing and selecting in multi candidates of consumers or providers should be also considered in negotiation mechanism. When the overlap is achieved on all terms after multi round negotiations, the negotiation process can be terminated and SLA will be established. Therefore, an appropriate autonomic negotiation mechanism plays a central role in SLA establishments.

Once the SLA was established and deployed, some kind of mechanism i.e. the runtime monitoring of SLA metrics in this stage and the SLA oriented dynamic resource management in next stage, is necessary to ensure that the SLA cannot be violated. In the stage of SLA monitoring, the values of QoS metrics defined in SLA will be monitored continually, and be checked whether the threshold of certain terms is violated, if the violation occurred, a notification will be sent to trigger the succeed actions. In the SLA monitoring mechanism, the relations between certain QoS level and the way of resource allocation need to be considered, especially in service composition scenario. For certain service compositions, their function is constructed under many other functions provided by external web services, therefor their nonfunctional properties also rely on the QoS value of those external services. On the other hand, from the perspective of those external service providers, because they do not deliver their service for only one consumer, the QoS properties will be fluctuating under different workload from different consumers, thus the QoS of service composition will also be dynamic changing at runtime.

The main goal of SLA management is to establish equilibrium between the consumer's requirements and the provider's capabilities of service delivering under the dynamic workload with a fluctuating runtime environment, where the SLA can be treated as some kind of mediation to reflect the dynamic changing of consumer's requirements and to constrain the resource allocation strategy of service provider. The reason of dynamic changing come from tow folds: one is the fluctuation of network environment itself; one is that, for service provider, because the workload from different user is changing continually, the performance of web service will
also be different in different workload under with fixed number of computing resources. For provisioning of stable QoS under the constraints of SLA, the self-adaptation control mechanism is necessary, for example, the admission control or dynamic resource allocation technologies. Especially under the background of cloud computing, the reallocation of computing resources can be complete in few seconds through virtualization technology, this bring up new opportunities
for more efficient and more flexible dynamic resource management technologies. The last stage of SLA auto-management, is in which the contract between service provider and consumer will be terminated, and the resources used will be released.

II. Literature Survey

PaaS (Platform as a Service) is an increasingly popular cloud model that delivers a complete development and hosting environment for cloud applications. This environment typically builds on virtualized resources owned by the PaaS provider or leased from public clouds on demand. PaaS customers deploy their applications in this environment while being completely shielded from managing the underlying resources, drastically reducing application management costs.
With the increasing use of PaaS, defining and maintaining SLAs (Service Level Agreements) between PaaS customers and providers becomes essential. In its most basic form, an SLA is a contract that specifies the service guarantees expected by the customers, the payment to the provider, and potential penalties when the guarantees are not met. A main limitation of current PaaS offerings is that the provided guarantees are based exclusively on resource availability
(e.g., number of virtual machines, memory size), rather than on application QoS properties (e.g., response time, throughput), which are more meaningful and useful to customers. As a result, it becomes the responsibility of PaaS customers to ensure QoS properties for their applications, which limits the value of PaaS systems.

D.Dib[1] proposes an PaaS system architecture, called Meryn, which is extensible with respect to programming frameworks. The architecture relies on a decentralized scheme to control the dynamic distribution of private resources among programming frameworks and to manage bursting to public clouds, when necessary. [1] also propose a profit optimization policy that aims at maximizing the overall provider profit while taking into account the payment of penalties when SLAs are unsatisfied. The policy optimizes the use of private resources, particularly in peak periods, before renting any public cloud resources. They evaluated it through a a series of simulations on the Grid'5000 experimental testbed, using the parapluie and the paradent clusters of the Rennes site and the results show that it provides up to 14.77 percent more profit for the provider and uses up to 80.99 percent less public cloud resources compared with a basic approach.

In [2] the authors focus on time and cost sensitive execution for data-intensive applications executed in a hybrid cloud. They consider two different modes of execution:
(1) Cost constraint-driven execution, where they minimize the time of execution while staying below a user-specified cost constraint, and (2) Time constraint-driven execution, where they minimize the cost while completing the execution within a user-specified deadline. They propose a model based on a feedback mechanism in which compute nodes regularly report their performance to a centralized resource allocation subsystem, and the resources are dynamically provisioned according to the user constraints.

In [3] the authors propose a decentralized economic approach for dynamically adapting the cloud resources of composite web service applications, so as to meet their SLA performance and availability goals in the presence of varying loads or failures. Their approach consists of checking and adapting the placement and the number of

replicas of application components for minimizing the operational cost of the application. The application components act as individual optimizers and autonomously replicate, migrate across VMs or terminate based on their economic fitness.

In [4] the authors consider the SLA-based resource allocation problem for multi-tier applications in cloud computing. Their objective is to optimize the total profit from the SLA contracts, reduced by the operational cost. A solution is proposed based on providing an upper bound on the total profit and applying an algorithm based on force-directed search.

In [5] the authors propose a resource allocation algorithm for SaaS (Software as a Service) providers to minimize infrastructure cost and SLA violations. To achieve this goal, they propose mapping and scheduling mechanisms to deal with the customer-side dynamic demands and resourcelevel heterogeneity. The mechanisms minimize the cost by optimizing the resource allocation within a VM, and thus allowing a cost-effective use of resources.

G.Li[6] proposes PSLA, a PaaS level SLA description language. PSLA is based on WSAgreement [7] which is an extendable SLA skeleton language. To the best of our knowledge, we propose for the first time a commonly usable and machine readable PaaS level SLA description language. PSLA is being adopted by Open Cloudware [8] project which aims at building a PaaS platform to manage applications running over multiple IaaSs during their lifecycle.

WS-Agreement [7] is widely used and has many existing implementations. WS-Agreement for Java framework (WSAG4J) is a tool to create and automatically manage SLAs. SLA managements, like offer validation, service level monitoring, persistence and accounting are supported. This makes the design and implementation of SLAs based on WS-Agreement much easier. Using WS-Agreement to create contract requires extensions for domain specific quality
expressions.

Wu et al. describe in [9] a resource allocation algorithm for SaaS providers that want to minimize infrastructure cost and SLA violations. They present cost-effective policies for mapping and scheduling in order to achieve profit maximization for the SaaS provider through the use of multiple IaaS providers. Li and Guo describe in [10] a stochastic ILP for optimal resource scheduling in cloud computing. They show how to select resources from public clouds to perform abstract services in business process instances while satisfying costs defined in SLAs. Reig et al. propose in [11] a strategy to minimize the computational resource consumption through a prediction system which determines the minimum cost resource for a job to be executed before its deadline to prevent SLA violations. Although these works present advances in scheduling tasks on clouds, none of them consider workflows.

Bittencourt and Madeira propose in [12] a strategy to schedule service workflows in a hybrid cloud. They show an algorithm called Hybrid Cloud Optimized Cost (HCOC) that decides which resources should be leased from the public cloud to increase the processing power of the private cloud. Although HCOC minimizes the monetary cost of the workflow execution within a deadline, Bittencourt and Madeira do not consider the notion of SLA, which is an important aspect in the business model proposed by the cloud paradigm. Pandey et al. describe in [13] a particle swarm optimization (PSO) heuristic to schedule application workflows. They show a model for task-resource mapping to minimize the overall execution cost in cloud computing environments. Nevertheless, they do not consider the notion of SLA on both user and provider sides.

In [14] R. Buyya proposes an SLA based resource provisioning for SaaS applications in cloud computing environment. It proposes customer driven heuristic algorithms to minimize the total cost (including infrastructure and penalty cost) by resource provisioning. These algorithms also take into account customer profiles (such as their credit level) and multiple Key Performance Indicator (KPI) criteria. A holistic way to quantify the customer experience is by considering KPIs from seven categories: Financial, Agility, Assurance, Accountability, Security and Privacy, Usability and Performance. To improve a SaaS application's performance
quality rating, we consider three KPIs, including one from provider's perspective: cost (part of the Financial category) and two from customers' perspective: service response time (part of the Performance category) and SLA violations (related to Assurance):Cost: the total cost including VM and penalty cost, Service response time: how long it takes for users to receive a response, SLA violations: the possibility of SLA violations creates a risk for SaaS providers. It proposes two algorithms 1)Minimizing the cost by minimizing the penalty cost through resource provisioning based on the customer's credit level 2)Minimizing the cost by rescheduling the existing requests.

III. Conclusion

In the next twenty years, service-oriented computing will play an important role in sharing the industry and the way business is conducted and services are delivered and managed. This paradigm is expected to have major impact on service economy; the service sector includes health services (e-health), financial services, government services, etc. This involves significant interaction between clients and service providers. In this thesis, we pointed out many challenges in addressing the problem of enabling SLA-oriented resource allocation in PaaS cloud computing to satisfy competing applications demand for computing services. The above survey discuss the deeper investigation in SLA-oriented resource allocation strategies that encompass computational risk management, and autonomic management of Clouds in order to improve the system efficiency, and improve profitability of service providers.

Acknowledgment

First and foremost I thank Almighty God for all the blessings endowed on me. I would also like to express my immense pleasure and gratitude towards management of my college for inspiring me to undertake this project. I deeply express my sincere thanks to to my project guide Ms. Christina Joseph for her whole hearted

support. I also would like to express my sincere gratitude to the HOD of Computer Science and Engineering department of Sahrdaya College Of Engineering And Technology. I would also like to extend my appreciation to all other faculty members for their help and advices.

REFERENCES

[1] D. Dib, N. Parlavantzas, and C. Morin, SLA-based Profit Optimization in Cloud Bursting PaaS, in Proceedings of 2014 14th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing, 2012.

[2] T. Bicer, D. Chiu, and G. Agrawal, Time and cost sensitive data-intensive computing on hybrid clouds, Cluster Computing and the Grid, IEEE International Symposium on, 2012.

[3] N. Bonvin, T. G. Papaioannou, and K. Aberer, Autonomic sla-driven provisioning for cloud applications, in CCGRID, 2011.

[4] H. Goudarzi and M. Pedram, Multi-dimensional sla-based resource allocation for multitier cloud computing systems, in Proceedings of the 2011 IEEE 4th International Conference on Cloud Computing, ser. CLOUD 11, 2011.

[5] L. Wu, S. Garg, and R. Buyya, Sla-based resource allocation for software as a service provider (saas) in cloud computing environments, in CCGrid, 2011.

[6] G. Li, F.Pourraz and P.Moreaux, âA.PSLA: a PaaS Level SLA Description Language, AI in Proceedings of 2014 IEEE International Conference on Cloud Engineering.

[7] A. Andrieux, K. Czajkowski, A. Dan, K. Keahey, H. Ludwig, T. Nakata, J. Pruyne, J. Rofrano, S. Tuecke, and M. Xu, Web services agreement specification (WS-Agreement), in Global Grid Forum, vol. 2, 2004.

[8] OpenCloudware, 2013. [Online]. Available: http://opencloudware.org/

[9] L. Wu, S. K. Garg, and R. Buyya, SLA-based admission control for a software-as-a service provider in cloud computing environments, Uni. of Melbourne, Australia, Tech. Rep. CLOUDS-TR-2010-7, sep. 2010.

[10] Q. Li and Y. Guo, Optimization of resource scheduling in cloud computing, in 12th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC), sep. 2010, pp. 315 -320.

[11] G. Reig, J. Alonso, and J. Guitart, Prediction of job resource requirements for deadline schedulers to manage high-level slas on the cloud, in 9th IEEE International Symposium on Network Computing and Applications (NCA), july 2010, pp. 162 -167.

[12] L. F. Bittencourt and E. R. M. Madeira, HCOC: a cost optimization algorithm for workflow scheduling in hybrid clouds, Journal of Internet Services and Applications, vol. 2, pp. 207-227, 2011.

[13] S. Pandey, L. Wu, S. M. Guru, and R. Buyya, A particle swarm optimization-based heuristic for scheduling workflow applications in cloud computing environments, in IEEE International Conference on Advanced Information Networking and Applications, 2010, pp. 400-407.