Enhancing SLA using Push/Pull Services in Cloud Computing

N. Ani Brown Mary
Software Engineer

Abstract— **The word Cloud is an acronym for Internet. Service Level Agreement is used to negotiate for resources between a consumer and a service provider. SLA is also called as a resources or application contract between consumers and service providers. Since SLA is an important task it has to be considered and evaluated to improve the profit of service providers and consumers. Here we use pull and push services to improve the SLA contract in a efficient way. By using this push and pull services it improves better services and activation in SLA negotiation.**

Keywords: ***SLA, push services, pull services, Cloud computing, negotiation***

I. INTRODUCTION

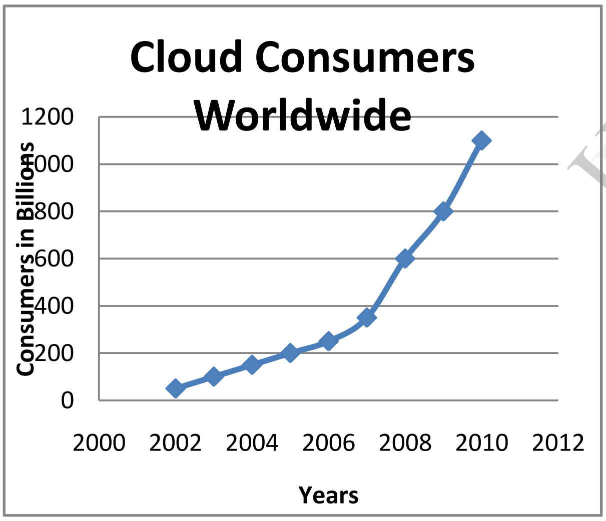

Figure:1 Cloud Consumers Worldwide in Billions

An SLA can be described as a contract between participants, outlined as a set of guarantees, QoS metrics and behaviours and should be treated as a legal document of the transaction[1]. More and more consumers are using Cloud services very interestingly.

Figure:1 shows the increasing consumers to Cloud services worldwide. This figure shows the tremendous growth rates up to 2008. That growth continues today, mainly due to China that has the largest number of consumers.

Cloud storage offerings are already available from providers located across the globe. The principle of Cloud computing and storage is that you shouldn't need to care where the processing occurs or where your data are stored. It is also possible for a consumer and the service provider to mutually agree upon a set of SLAs with different .performance and cost structure rather than a single SLA. The consumer has the flexibility to choose any of the agreed SLAs from the available offerings. At runtime, the consumer can switch between the different SLAs. Since SLA is a major task it is used for many purposes in cloud computing.

Each SLA goes through a sequence of steps starting from identification of consumers and cloud providers, then identifying their terms and conditions, then monitoring their contract and negotiation. The cloud provider manages and offers guarantees on availability of the infrastructure, namely, server machine, power, network connectivity, and so on. Enterprises manage themselves, their applications that are deployed on these server machines. The machines are leased to the consumers and are isolated from machines to other consumers. Cloud providers offers a service-level guarantees to consumers.

Service provider advertises base service offerings through standard publication media, and the consumers should be able to locate the service provider by searching the catalog. The consumers can search different competitive offerings and shortlist a few that fulfil their requirements for further negotiation. So SLA plays a vital role in Cloud computing. It act as a interface between consumer and service providers. Since we have to put special effects to improve SLAs. In our work push functions are used to structure the Virtual Machines to avoid the unnecessary consumer and service provider interactions.

II. RELATED WORK

Service Level Agreements are very useful to improve profit of service providers. Zhang et.al[5] designed a resource allocation scheduler to maximize the profits of web applications. Here the problem that occurs due to multi-tier SLA's has not been considered. Normally, traditional push services has been used only in mobile computing. But Brenna et.al[6] used push/pull services in information handling. The push functions are used to structure the Internet to avoid the unnecessary client-server interactions dominating the Internet. These functions are also used in banking[19] in an efficient way.

Morin et al[9] have identified SLA issues and their corresponding challenges, proposing to use risk and Service Level Agreement (SLA) management as the basis for a service level framework to improve governance, risk and compliance in cloud computing environments. Stefan et al[10] used an specified SLA to attract companies to outsource their services to clouds. Here providers need to offer consumers using key performance indicators only.

III. SEQUENCE DIAGRAM FOR PUSH SLA

This sequence diagram shows the relation between the consumers, SAAS, PAAS and IAAS. Service Level Agreements are used have the contract relationship and negotiation between each providers to provide the resources from virtual machines. Indicate new SLA shows the contract or offers that are provided by the service providers to consumers. That is pushed or shown to consumers using push SLA technique. Service Indication technique is very much useful to indicate services that are provided by various providers to consumers. Resources and Services are provided to various consumers depending on their SLA contract.

The Cloud providers are flexible in allocating and de-allocating computing resources among the co-located applications. Therefore, the cloud providers are also responsible for ensuring to meet their consumers application. It is also possible for a consumer and the cloud provider to mutually agree upon a set of SLAs with different performance and cost structure rather than a single SLA. The consumer has the flexibility to choose any of the agreed SLAs from the available offerings. At runtime, the consumer can switch between various SLAs. SLA displays the response from various consumers to the service providers. Both consumers and providers share their SLA constraints between them.

Then depending on the SLA constraints the providers provides offer to the consumers. Service provider advertises these base service offerings through standard publication media, and the consumers should be able to locate the service provider by searching the catalog. The consumers can search different competitive offerings and shortlist a few that fulfill their requirements for further negotiation. Individual SLAs for enterprises can be derived by customizing these base SLA templates. Thus this sequence diagram is very much useful to know the interactions between the consumers and the service providers with push SLA services.

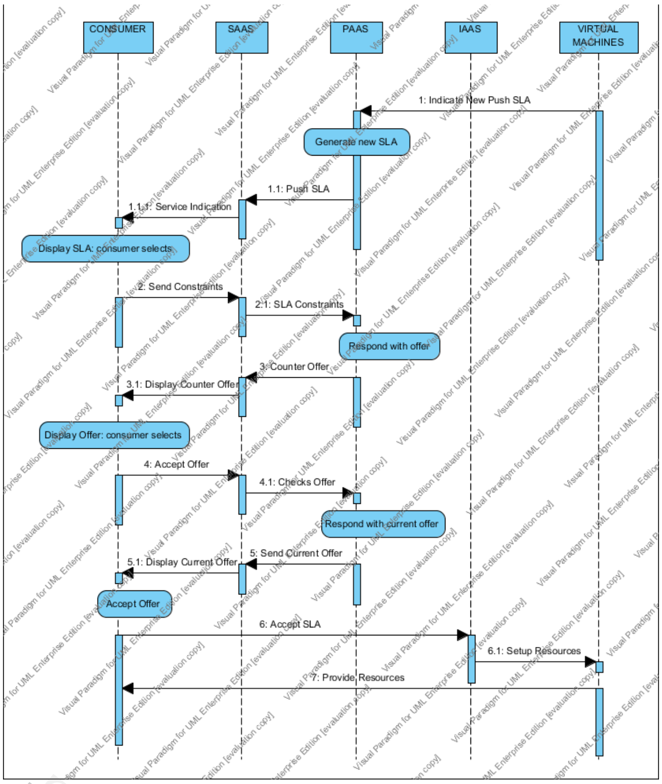

Figure:2 Sequence Diagram for SLA

IV. PUSH ARCHITECTURE

Consumers pulling resources from providers are typical for today's Cloud system. In a push context the service provider initiates the resource transfer, not the consumer. The Infrastructure as a Service provider transfers resource via the Platform as a Service provider to a consumer. The push access controls communication between Infrastructure as a Service provider and Platform as a Service provider. The push over the air is used between the Platform as a Service provider and the consumer. If the Infrastructure as a Service provider is able to use the push OTA, it can directly communicate with the consumer, too.

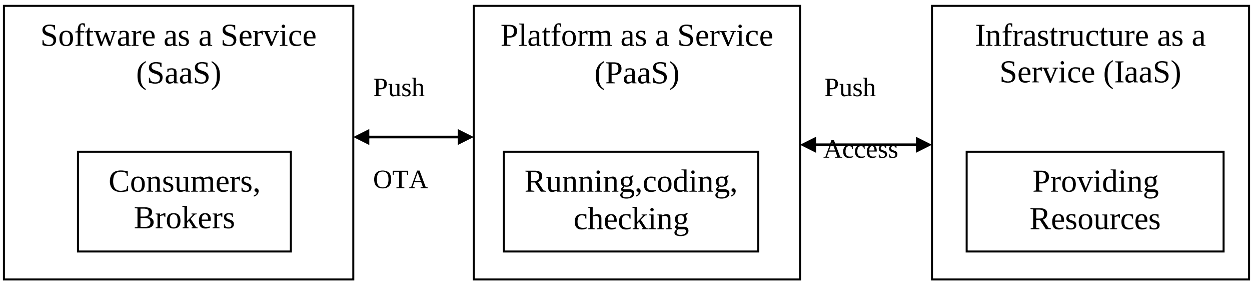
Software as a Service (SaaS)
Consumers, Brokers
Push
OTA
Platform as a Service (PaaS)
Running,coding, checking
Push
Access
Infrastructure as a Service (IaaS)
Providing Resources

Figure:3 Push Architecture

The Platform as a Service provider provides many functions to transform SLA agreements and applications exchanged between service provider and consumer. If required, the Platform as a Service provider send success or failure notifications to the Infrastructure as a Service provider. In Platform as a Service developers can design, build, and test applications that run on the cloud provider's infrastructure and then deliver those applications to consumers from the provider's servers.

Push access functions:

The push access procedure transfers applications from Infrastructure as a Service provider to the Platform as a Service provider. Although push access procedure was developed independently from the underlying delivery service, post procedures are used for transmission. Push access functions offers the following operations:

Push submission: Delivery of push applications from the Infrastructure as a Service provider to the Platform as a Service provider for forwarding to a consumer. The response indicates the initial acceptance or rejection of the forwarding request.

Result notification: The Platform as a Service notifies the Infrastructure as a Service so that the result of the push request can be known. With the help of this notification, the virtual server knows if the applications reached the consumer, if the consumer deleted the request, if any penalty occurred at the consumer or if the request arrived too late.

Push cancellation: The Infrastructure as a Service can try to cancel a push request before delivery to a consumer. The result of the Platform as a Service indicates if the cancellation was successful or not. Cancellation may be used, e.g., for periodic applications buffered at the Platform as a Service because the consumer is currently unavailable. In this case, only the most recent applications may be of interest.

Status query: With the help of push access procedure, the Infrastructure as a Service provider can query the status of a push resources. The result could indicate, if the Platform as a Service provider already delivered the resources or is still trying to reach the consumer.

Consumer capabilities query: The Infrastructure as a Service provider might be interested in the consumer's capabilities. Additionally, a Platform as a Service provider can inform the Infrastructure as a Service provider about its capabilities regarding resource transformation.

Push OTA functions:

The push OTA procedure is a very simple procedure used on Software as a Service. Software as a Service provides a shared state between a consumer and a virtual server to optimize resource transfer. Software as a Service offers general features needed for SLA contract exchange between cooperating consumers and virtual servers. Software as a Service provider introduces sessions that can be established from a consumer to a virtual server and may be long lived. Sessions can also be released in an orderly manner. The capabilities of suspending and resuming a session are important to cloud applications. Consumers and virtual servers can agree upon a common level of procedure functionality during session establishment. Example parameters to negotiate are maximum consumer file size, maximum outstanding requests, penalty options, and virtual server speed. Optionally, Software as a Service provider supports a consumer that can send multiple requests to a virtual server simultaneously. This improves efficiency for the requests and provides resources as fast as it possible. Latency is also improved, as each result can be sent to the consumer as soon as it is available.

The following service primitives have been defined:

po-push: unacknowledged push of request within a push session;

po-confirmedpush: acknowledged push of request within a push session;

po-pushabort: rejection of a push request by the consumer;

po-unit-push: unacknowledged push of request;

The following service primitives are also available for push session management:

pom-connect: creation of a push session by the consumer;

pom-suspend: suspension of a push session;

pom-resume: resume of a push session;

pom-disconnect: termination of a push session;

pom-sessionrequest: request for a push session by a virtual server;

A push SLA contract may comprise additional attributes indicating, e.g., the importance or the lifetime of the SLA contract. The following example shows a simple contract offer containing a special pizza, which is only valid for five minutes. If a consumer wants to use this offer, consumer has to request the indicated service via the Software as a Service provider. After five minutes, this push SLA contract will be automatically deleted.

```
<?xml version=”1.0”?>
<!DOCTYPE si PUBLIC “-//WAPFORUM//DTD SI
1.0//EN”
“http://www.wapforum.org/DTD/si.dtd”>
<si>
<indication
href=”http://www.piiiizza4u.de/offer/salad.wml”
created=”2002-10-30T17:45:32Z”
si-expires=”2002-10-30T17:50:31Z”>
salad special: The 5 minute offer
</indication>
</si>
```

Another variant of delayed service usage can be described using service loading. In this case, the Infrastructure as a Service also sends a short push contract to the consumer. In contrast to service loading, the consumer's broker decides when to submit the request, i.e., using the service. Although the service is accessed using a traditional pull scheme it looks for the consumer like a push service. The consumer does not notice the delay between the arrival of the push service and the pull. A disadvantage of service loading is the higher number of requests sent over the Cloud. The following example shows how a Infrastructure as a Service send a push contract to the consumer. No interaction with a consumer is intended.

```
<?xml version=”1.0”?>
<!DOCTYPE sl PUBLIC “-//WAPFORUM//DTD SL
1.0//EN”
```

```
“http://www.wapforum.org/DTD/s1.dtd”>
<sl>
href=”http://www.piiiizza4u.de/offer/salad.wml”>
</sl>
```

V. CONCLUSION

We think that although preliminary and ambitious, this work in progress addresses a critical issue in the growing domain of accountable and trustworthy cloud computing. In this context, push and pull services are an attempt to bridge a gap between the mobile computing and management levels of cloud computing.

The main contribution of this paper is a statement and discussion based on push/pull services in SLA. We propose as contributing elements towards the design of a service level cloud computing SLA push management framework. The planed framework will be organized around interrelated themes such as push function management coupled with SLA and push management.

REFERENCES

1. Viktor Yarmolenko, Rizos Sakellariou, Djamila Ouelhadj, Jonathan M Garibaldi,”SLA Based Job Scheduling: A Case Study on Policies for Negotiation with Resources”, published in the 45th Hawaii International Conference on System Sciences 2012.
2. Li Zhang, Danilo Ardagna,”SLA Based Profit Optimization in Autonomic Computing Systems”, published in the ICSOC' 04, 173 (15-19 Nov 2004).
3. Jon MacLaren, Rizos Sakellariou, Krish T. Krishnakumar, Jon Garibaldi, Djamila Ouelhadj,”Towards Service Level Agreement Based Scheduling on the Grid”, published in the Workshop on Planning and Scheduling for Web and Grid Services (in conjunction with ICAPS-04; Whistler, British Columbia, Canada, 100 (3-7 Jun 2004).
4. Yongjing A. Li, Jiang B. Wu,” An SLA based SaaS Security Level”, published in the TELKOMNIKA, Vol.11, No.7, July 2013, pp. 4111~4121 e-ISSN: 2087-278X.
5. Li Zhang, Danilo Ardagna,” SLA Based Profit Optimization in Web Systems”, published in the ACM Transactions WWW2004, May 17–22, 2004, New York, USA.
6.. N.Ani Brown Mary, "Profit Maximization for Service Providers using Hybrid Pricing in Cloud Computing" published in the Proceedings of the International Journal of Computer Applications Technology and Research Volume 2, Issue 3, 218 - 223, 2013.
7. Niehorster, Brinkmann, Kruger, Simon,” Enforcing SLAs in Scientific Clouds Cluster Computing”, published in the 2010 IEEE International Conference on, 2010:178-187.
8. Cheng Xu.”Research on Performance Management Mechanism Based on SLA in SaaS Application”, published in the ME Thesis, Shandong: Shandong University, 2010.
9. Jean-Henry Morin,Jocelyn Aubert, Benjamin Gateau,” Towards Cloud Computing SLA Risk Management: Issues and Challenges”, published in the 45th Hawaii International Conference on System Sciences 2012.
10. Stefan Frey, Claudia Luthje, Christoph Reich,” Key Performance Indicators for Cloud Computing SLAs”, published in the EMERGING 2013 : The Fifth International Conference on Emerging Network Intelligence.
11. Happe, Theilmann, Edmonds,Kearney,” Service Level Agreements for Cloud Computing”, published in the Springer-Verlag, 2011, ch. A Reference Architecture for Multi-Level SLA Management, pp. 13–26.
12. Ludwig, Keller, Dan, King, Franck, “Web Service Level Agreement (WSLA) Language Specification”, published in the v1.0, Jan. 2003. [Online]. Available: http://www.research.ibm.com/wsla/WSLASpecV1- 20030128.pdf [retrirved: may, 2013].
13. Foued Jrad, Jie Tao,Achim Streit,” SLA BASED SERVICE BROKERING IN INTERCLOUD ENVIRONMENTS”, published in the CLOSER 2012 - 2nd International Conference on Cloud Computing and Services Science.
14. Xuan Liu, Feng Xu,” The SLA Framework Based on Cloud Computing”, published in the International Workshop on Cloud Computing and Information Security (CCIS 2013).
15. Jian Zhang. “The Study of Cloud Computing Service Level Agreement”, published in theTelecommunications Network Technology, 2012. pp. 59-64.
16. N.Ani Brown Mary and K.Saravanan, "Performance factors of Cloud Computing Data Centers using [(M/G/1) : (∞/GD Model)] Queuing Systems" published in the Proceedings of the International Journal of Grid Computing & Applications (IJGCA) Volume 4, Issue 1, March 2013.
17.Selvarani, Sudha Sadhasivam. “A Novel SLA based Task Scheduling in Grid Environment”,published in the International Journal of Applied Information Systems (IJAIS) – ISSN : 2249-0868 Foundation of Computer Science FCS, New York, USA Volume 1– No.1, November 2012 – www.ijais.org.
18. Balakrishnan, Thamarai Selvi, Rajesh Britto “Service Level Agreement based Grid Scheduling”, published in the proceedings of the 2008 IEEE International Conference on Web Services.
19. Fouzia Mousumi, Subrun Jamil, “Push Pull Services Offering SMS Based m-Banking System in Context of Bangladesh”, published in the International Arab Journal of e-Technology, Vol. 1, No. 3, January 2010.
20. Aloussi, “SLA Business Management Based on Key Performance Indicators”, published in the Proceedings of the World Congress on Engineering 2012 Vol III WCE 2012, July 4 - 6, 2012, London, U.K.
21. N.Ani Brown Mary, "Profit Maximization For Saas Using Sla Based Spot Pricing in Cloud Computing", published in the Proceedings of the International Journal of Emerging Technology and Advanced Engineering (ISSN 2250-2459, An ISO 9001:2008 Certified Journal, Volume 3, Special Issue 1, January 2013).
22. Lars Brenna, Dag Johansen,” Configuring Push-Based Web Services”, published in the http://waif.cs.uit.no
23. Ruoying Zhang, Xuesong Que, Luoming Meng. “The Said Method and Application of SLA”,published in the Journal of Beijing University of Posts and Telecommunications, 2003. pp. 13-17.