Browser Integrated Vendor Neutral Cloud QoS Monitoring System

Frankline Makokha
School of Computing and Informatics
University of Nairobi, Kenya
Email: goldmedalist321 [AT] gmail.com

Elisha, E. Opiyo; Christopher K. Chepken
School of Computing and Informatics
University of Nairobi, Kenya

Abstract—**The desire to optimize both capital and operational costs has driven most corporates to adopt cloud computing as both a cost cutting and efficient way of providing services to their clients. Consequently the number of cloud providers has increased posing a decision challenge to the cloud users. This paper presents a chrome browser integrated QoS monitoring that is not tied to the architecture of any cloud provider and hence capable of being used to compare the performance of various cloud providers to aid in deciding which cloud to procure as well as aiding in Service Level Agreement confirmation. The prototype was implemented using JavaScript and SQL lite database and tested on Google docs, Salesforce Hubspot and shopify. The results could be used to compare against those measured by the cloud provider's QoS tools which are normally integrated in the cloud solutions.**

Keywords-Vendor Neutral ; QoS, Service Level Agrement ; cloud computing.

I. INTRODUCTION

Due to convergence of cloud services, cloud providers propose same services differentiated solely by prices and performance levels [1]. Whereas tools exist on the provider's platform to monitor the QoS of the platform, it does not offer a comparison option for the cloud users to compare two or more cloud platforms over the same services. Further, it is not possible to validate the QoS as measured by the cloud provider's cloud integrated tool.

The various QoS monitoring models from which the current tools are designed from are the Quality of Service MONitoring as a Service Model (QoSMONaaS), CloudQual, Agent based model and Adaptive QoS-driven Monitoring Model [2].

The Agent based model uses software agents installed in virtual machines [3], the QoSMONaaS model is a modifiable model enabling portability from one cloud platform to another which relies on an adaptation layer between it and the cloud platform [4], CloudQual relies on the cloud provider's API to interface with the cloud platform [5], while the Adaptive QoS-driven Monitoring Model uses various modules each with a driver to monitor a specific QoS parameter, further it also has a certifier module to confirm that the QoS agreed can be met by the platform [6].

An analysis done by [7] shows that all the cloud QoS monitoring tools developed by the above models are tied to the architecture of the cloud provider and thus can not be used for cross vendor comparison.

To eliminate the problem of vendor platform dependent tools, a vendor neutral model is proposed by [7] which could be used for cross vendor QoS comparison.

II. VENDOR NEUTRAL CLOUD QOS MONITORING MODEL

A. Current Framework for QoS Monitoring

From an analysis done by [2], the current high level cloud QoS monitoring models are developed from the framework shown in figure 1.

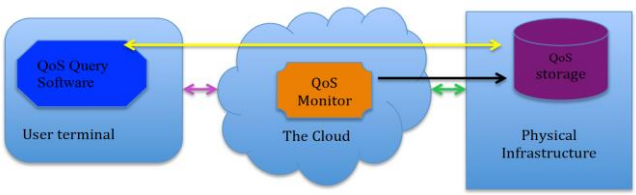

Figure 1: Current High Level Framework for QoS Monitoring

The above framework explains why the tool has to be tied to the architecture of the cloud platform as it resides in the cloud and this makes it vendor specific.

B. *Proposed Framework for QoS Monitoring*

To overcome the vendor specificity of the current framework [7] proposed a new framework as depicted in figure 2.

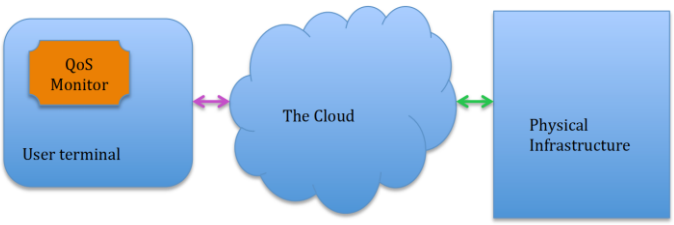

Figure 2: Proposed Framework for Cloud QoS Monitoring

From figure 2, delinking the tool for QoS monitoring from the cloud platform provides a vector for creating a vendor neutral QoS Monitoring model [7].

To realize this [7] proposed a model based on browser based cloud access method for software as a service solution, which would be implemented as a browser extension. The proposed model is depicted in figure 3

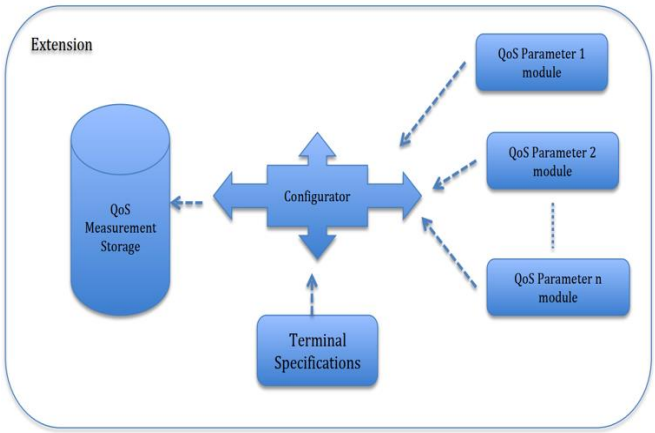

Figure 3: Proposed Vendor Neutral Cloud QoS Monitoring Model

The decision to develop a bowser based model was informed by the fact that Software as a Service solution is accessed via web portals [8].

III. IMPLEMENTATION OF THE PROPOSED VENDOR NEUTRAL MODEL

The proposed vendor neutral model was implemented as an extension on chrome web browsers. The development tools used were a combination of standard web development technologies, namely HTML, CSS, JavaScript, node.js and SQLite database.

Google chrome was chosen because according to [9] it is the most widely used browser with the highest number of extensions developed for it.

A. *QoS Metrics to Monitor*

This experiment was designed to measure Quantified cloud QoS metrics namely Service response time, which is the time taken for the cloud service to initialized and ready for use, availability which is the total time the cloud service is usable out of the entire period the user may want to use it and stability which is the variations in the service response times.

B. *Development Methodology*

The development of the model was done iteratively using a prototype approach where the developed module of the model was coded, tested to confirm whether it measures the desired parameters accurately. This was done for each of the modules in the new vendor neutral cloud QoS Model.

C. *Algorithm Development*

After the prototype was developed and confirmed to be measuring the set parameters, the following algorithms were developed from the prototype.

i) Terminal Specifications Module Algorithm

```
On Extension Installation
    Create and Assign Client_ID          // Auto generate unique
                                         number to be used to identify
                                         user /user terminal throughout
                                         the monitoring process

    Get Client_ID details as:            // get the number of processors
        cpu_numberOfCore,                the terminal has, the
        cpu_archName,                    manufacturer of the processor,
        cpu_modelName,                   model , memory size and the
        ram_size,                        date of installation
        date_joined

    Connect to the Database identify table   // connect to the database and
        Log Client_ID                        store the data
        Log Client_ID  details
End
```

ii) Internet connection monitoring algorithm

```
While the monitoring status is on
    Create operation ID                          // Auto generate unique
                                                 number to be used to map to
                                                 client ID created earlier
    Check supplied url:                          // check if the url is valid for
                                                 cloud computing service
    Check internet connection status             // confirm the terminal is
        If internet available                    connected to the internet
        Get and log connection parameters as:
            Round trip time;                     Effective type whether is 3G
            Downlink:                            or 4G speeds
            EffectiveType:
        While Monitoring status is on
        and
        Monitor the connection parameters
End
```

iii) QoS Parameter Monitoring Algorithm

```
While url is valid and internet connection is up
    While url is loading                         // check the time it takes for
        Log the start of loading time and end of loading time   the main page to initialize
                                                 ready for use
    On complete of url loading :                 // listen to any actions the
        Listen to user mouse and button events   user performs on the page
    On user event executed:                      // log the start and end time
        Log the start of user event and time to completion of user event   of listened user event
End
```

D. Tool Integeration into Chrome Browser

To install the vendor neutral cloud QoS monitoring into chrome the following steps were followed:

i) On the chrome browser type: chrome://extensions

This provides an option to either package an extension or load-unpackaged extension. Given this is a trial extension chose load unpacked and browse to the location where the package to be loaded is stored.

A snapshot of the interface displayed during integration of the developed QoS tool into chrome is shown in figure in as in figure 4.

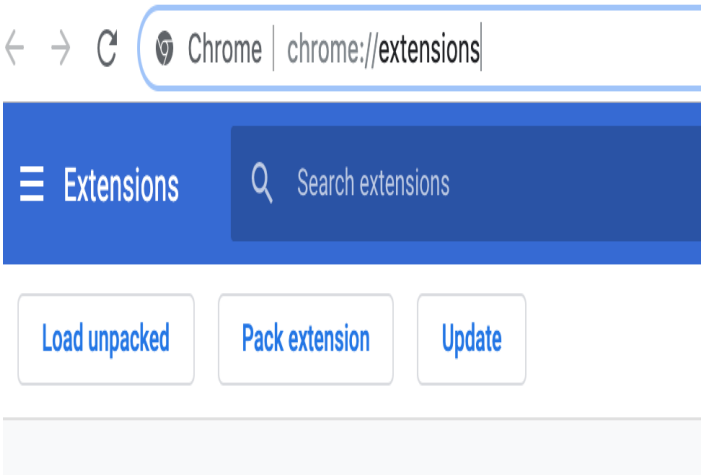

Figure 4: Interface to Integrate QoS Tool in Chrome Browser

After loading the tool, it will appear among the list of extensions already integrated in chrome as shown in figure 5.

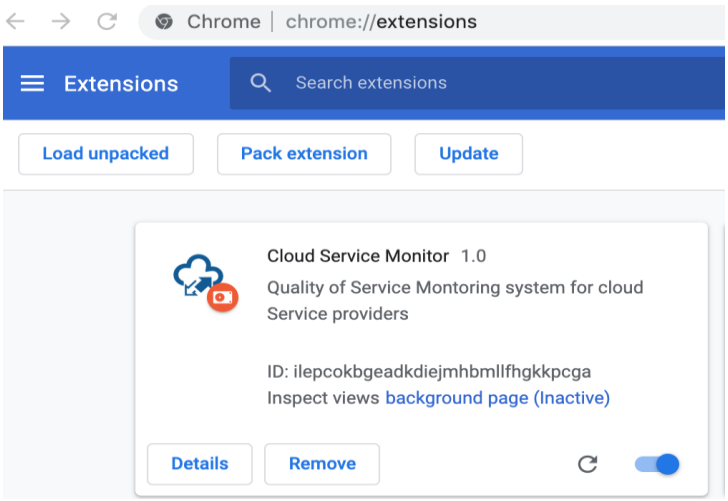

Figure 5: Cloud Service Monitor Integrated in Chrome

After installation, the tool was run from the computer terminal using the command: *npm run dev* as shown in figure 6.

```
Abus-MacBook-Pro:qos-webapplication abuhamza$ npm run dev

> qosmonitor@1.0.0 dev /Users/abuhamza/Desktop/QoSaPP/qos-webapplication
> nodemon bin/www

[nodemon] 1.19.0
[nodemon] to restart at any time, enter `rs`
[nodemon] watching: *.*
[nodemon] starting `node bin/www`
QoS Application running on port 8484
```

Figure 6: Starting the QoS Monitoring Tool

E. *Experimentation*

From figure 6, we learn that the application is running as local host on port 8484 accessible on the browser. The application is opened as shown in figure 7.

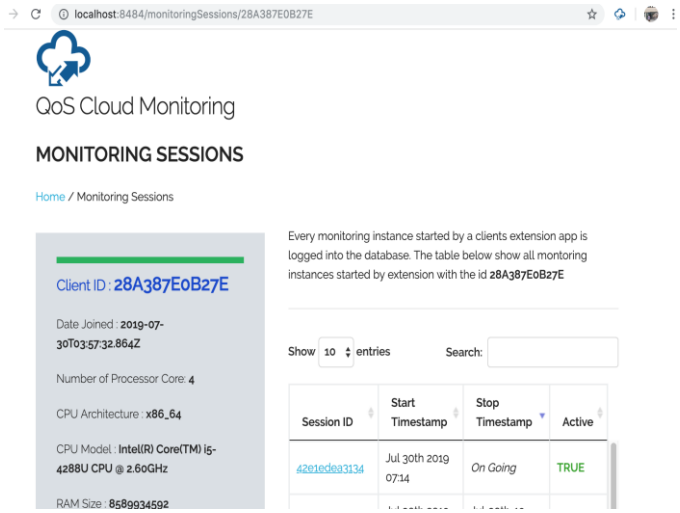

Figure 7 : Active QoS Monitoring Platform on the Browser

Once the platform is turned on, the configuration was done for the cloud service sites to be monitored. The tool will record the configured QoS parameters and store the results in the database created on the fly during tool installation.

F. *Results*

For the experiments conducted on salesforce, google docs and hubspot free cloud services, the results are as shown in table 1.

No	Cloud Provider	Service Response Time	Availability	Stability
1.	Sales Force	2.93 sec	100%	0.252 sec (stable)
2.	Google (docs)	4.83 sec	100%	1.654 sec (stable)
3.	Hub Spot	2.45 sec	100%	1.574 sec (stable)
4.	Shopify	2.59 sec	100%	1.3 sec (stable)

Table 1: Cloud Providers QoS Monitoring Results

From table 1, the service response time is the average time it took for the user specified service to be initialized and ready for use.

Availability was measured as the number of instances the user request for a service and gets the service against the number of instances the requested service is not available.

Service stability was computed using standard deviation. a standard deviation value greater than the mean means the system is not stable while a standard deviation value of less than the mean implies the system as stable.

The results were measured under the same system conditions and Internet conditions namely a mac book pro laptop with Intel(R) Core(TM) i5-4288U CPU @ 2.60GHz and an average internet effective type 3G.

IV. CONCLUSION

Due to the successful implementation and experimentation with the proposed vendor neutral model, it is recommended that a case study be conducted to compare results of the new vendor neutral model with those from the vendor specific model under the same test environment and QoS Parameters.

REFERENCES

[1] Mamoun Hussein Mamoun et Eslam Mohamed Ibrahim (2014). A Proposed Framework for Ranking and Reservation of Cloud Services. *International Journal of Engineering and Technology, Volume* 4, No. *9.*

[2] Makokha, F., Okello-Odongo and Opiyo, E.(2017). Challenges of Quality of Service Monitoring in Cloud Computing Solutions. *International Journal of Computer and Information Technology* .Volume 06– Issue 06.

[3] Dhingra, M., Lakshmi, J., and Nandy, S. K. (2012). Resource Usage Monitoring in Clouds in 12th Proceedings of the 2012 ACM/IEEE 13th International Conference on Grid Computing. IEEE Computer Society: Washington. Pages 184-191 .

[4] Adinolf, O., Cristaldi, R., Coppolino, L. and Romano, L. (2012). QoS-MONaaS: A Portable Architecture for QoS Monitoring in the Cloud. Signal Image Technology and Internet Based Systems (SITIS), 2012 Eighth International Conference on, Napes, Italy. IEEE.

[5] Zheng, X., Martin, P., Brohman, K. and Xu, L. D. (2014). CLOUDQUAL: A Quality Model for Cloud Services. IEEE Transactions on Industrial Informatics, Vol. 10, No. 2.

[6] Serhani, M.A., Atif, Y. and Benharref, A. (2014). Towards an adaptive QoS-driven monitoring of cloud SaaS. International Journal on Grid and Utility Computing, Vol. 5, No. 4

[7] Makokha, F and Opiyo, E.(2018). A Vendor Neutral QoS Monitoring Model for SaaS Cloud Computing Solutions. International Journal of Computer and Information Technology (ISSN: 2279 – 0764) Volume 7– Issue 01, January 2018.

[8] Buyya, R., Broberg, J. and Goscinski, A. (Eds., 2011). *Cloud Computing: Principles and Paradigms*. Hoboken, New Jersey: John Wiley & Sons, Inc.

[9] Sanchez-Rola, I., Santos, I., and Balzarotti, D. (2017). Extension Breakdown: Security Analysis of Browsers Extension Resources Control Policies: Proceedings of the 26th USENIX Security symposium (2017), Vancouver, BC, Canada