ISSN 2278-3083

Volume 2, No.2, March – April 2013

International Journal of Science and Applied Information Technology

Available Online at http://warse.org/pdfs/ijsait01222013.pdf

GOOGLE APP ENGINE AND PERFORMANCE OF THE WEB APPLICATION

Dr. U.S Pandey

Associate Professor, School of Open Learning Delhi University, Delhi, India

Email:uspandey1@gmail.com,

Anjali Jain

Mewar University, Gangrar, Chittorgarh(Rajasthan),India

Email:anjalijain81@gmail.com

ABSTRACT

Cloud computing is an internet-based model of computing, where the shared information, software and resources are provided to computers and other devices upon demand. Cloud computing provides enormous business opportunity. Cloud computing could dramatically lower the need for upfront investments in it and ongoing maintenance. User pay only for the time, resources, and capacities he uses while scaling up to accommodate the changing business needs. There are various cloud computing platforms available. One very interesting cloud computing platform is Google App Engine (GAE). Cloud computing has emerged recently that focuses on reduction of expenses on resources and thus the application can be developed in a pay as you go manner. Then the web applications can be uploaded to the cloud and maintained without any issues on the enterprise side. Cloud computing on the other hand has emerged as a solution to cut down the enterprises expenditures but there is a limited literature about how to use it.

This article focuses on how to move social applications into the cloud and on the evaluation of their performance. The study shows how to implement a social networking application using GAE cloud with limited code and evaluates the scalability of the applications.

Key words: Apache JMeter, Cloud computing, Goggle App Engine, Social networking services.

1. INTRODUCTION

According to NIST “Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction .”

Investopedia explains 'cloud computing'-cloud computing is so named because the information being accessed is found in the "clouds", and does not require a user to be in a specific place to gain access to it. Companies may find that cloud computing allows them to reduce the cost of information management, since they are not required to own their own servers and can use capacity leased from third parties. Additionally, the cloud-like structure allows companies to upgrade software more quickly. [1]

Cloud computing refers to applications and services offered over the internet. These services are offered from data centres all over the world, which collectively are referred to as the "cloud." this metaphor represents the intangible, yet universal nature of the internet. [2]

Cloud computing is a model for enabling on-demand network access in order to share computing resources such as network bandwidth, storage, applications, etc , that is able to be rapidly scalable with minimal service provider management [3].

Gartner defines cloud computing as a style of computing in which scalable and elastic IT-enabled capabilities are delivered as a service using Internet technologies. In cloud computing, the word cloud is used as a metaphor for "the Internet," so the phrase cloud computing means "a type of Internet-based computing," where different services -- such as servers, storage and applications -- are delivered to an organization's computers and devices through the Internet.[17]

Cloud computing offers the services on demand and are delivered economically. Any user with an internet connection can access the cloud and the services it provides. Since these services are often connected, users can share information between multiple systems and with other users. Traditionally data storage devices and computer set up need to be at the same location, but cloud allows users to access to services any time, on any device, anywhere in the world. Cloud Computing allows the users to upload and maintain their data, application software on the remote servers which can be easily accessible anywhere, anytime with the help of internet. With the help of Cloud Computing user doesn’t have to install applications. This helps small and middle size entrepreneurs to set up their businesses when they are not able to indulge much amount of hardware and storage space. It also provides more efficient computing by centralizing data storage, processing and bandwidth. As consumers, we now expect that the information should be available us in real time, and business applications in the cloud are heading in that direction as well. [4]

According to analyst outfit IDC, "cloud services" are the consumer and business products, services and solutions that people access and consume in real-time over the Internet. Cloud services are typically easy to set up, with the customer not having to worry about the complications of implementation. The services are also based on a "pay-as-you-go" or subscription model to make them more affordable for various business types and sizes.

In cloud computing there is no direct connection to a server, instead this is a model which delivers the information technology services and resources are retrieved from the internet through web-based tools and applications. Cloud computing allows to work remotely. Here data and applications are stored on the servers.

According to Roger Jennings 2013 will bring continued growth to all primary cloud services segments; SaaS, PaaS and IaaS combined will grow at a compounded annual growth rate (CAGR) of about 25%. PaaS and IaaS alone will grow from $964 million in 2010 to $3.9 billion in 2013, a CAGR of 60%, according to the 451 group. Platform-agnostic IaaS will continue to amass the lion's share of revenues, but provider-specific PaaS will gain a larger split of the pie as developers become more comfortable making the transition from on-premises applications.

Gartner inc. Predicts the market for public cloud computing services will swell to $207 billion by 2016, up from $109 billion today. It also predicts that in 2017, the chief marketing officer will spend more on it.

The economist-IBM report [5] provides some shining examples of companies that are employing cloud computing to not only cut it costs, but also play a role in disruptive innovation*:*

1. Through cloud cost flexibility, online marketplace gains access to more powerful analytics online.
2. Greater business scalability enables online video retailer to meet spikes in demand
3. Greater market adaptability provides online entertainment platform the ability to reach any type of customer device.
4. Masked complexity enables access to services, no matter how intricate the technology they're built on.

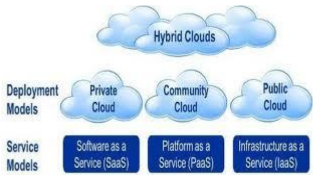
According to the Cloud Computing Manifesto, "The key characteristics of the cloud are the ability to scale and provision computing power dynamically in a cost efficient way and the ability of the consumer to make the most of that power without having to manage the underlying complexity of the technology" [18].

2. CHARACTERISTICS OF CLOUD COMPUTING [19]

- It is infinitely scalable.
- It provides one or more of an infrastructure for platforms, a platform for applications or applications themselves.
- Clouds can be used for every purpose from disaster recovery/business continuity through to a fully outsourced ICT service for an organisation.
- Currently the major cloud providers had already invested in large scale infrastructure and now offer a cloud service to exploit it.
- As a consequence the cloud offerings are heterogeneous and without agreed interfaces cloud providers essentially provide datacentres for outsourcing.
- There are concerns over security if a business places its valuable knowledge, information and data on an external service.

3. DEPLOYMENT MODELS OF CLOUD COMPUTING

NIST specified the following deployment models of cloud computing:-


Hybrid Clouds
Deployment Models
Private Cloud
Community Cloud
Public Cloud
Service Models
Software as a Service (SaaS)
Platform as a Service (PaaS)
Infrastructure as a Service (IaaS)

Private cloud-The cloud infrastructure is provisioned for exclusive use by a single organization comprising multiple consumers (e.g., business units). It may be owned, managed, and operated by the organization, a third party, or some combination of them, and it may exist on or off premises.

Community cloud-The cloud infrastructure is provisioned for exclusive use by a specific community of consumers from organizations that have shared concerns (e.g., mission, security requirements, policy, and compliance considerations). It may be owned, managed, and operated by one or more of the organizations in the community, a third party, or some combination of them, and it may exist on or off premises.

Public cloud-The cloud infrastructure is provisioned for open use by the general public .It may be owned, managed, and operated by a business, academic, or government organization, or some combination of them. It exists on the premises of the cloud provider.

Hybrid cloud- The cloud infrastructure is a composition of two or more distinct cloud infrastructures (private, community, or public) that remain unique entities, but are bound together by standardized or proprietary technology that enables data and application portability(e.g., cloud bursting for load balancing between clouds).

4. BENEFITS OF CLOUD COMPUTING [6]

- **Proven web-services integration.** Cloud computing technology is far easier to integrate with your existing applications - whether traditional or virtual.
- **100% virtual infrastructure** - no software, no onsite installation. Cloud computing slashes the capital costs of getting up and running.
- **Faster and lower-risk deployment.** With cloud server technology you can save your business years and millions of dollars.
- **Support for deep customisations.** Cloud computing in India allows deep customisation and application configuration. Cloud computing technology is well suited to application development, ideal for organisation's constant evolution.
- **Empowered business users.** Cloud computing technology allows instant, point-and-click customisation and report generation for business users. So it doesn't have to spend half its time making minor amendments and running reports.
- **Automatic upgrades with no downside.** Cloud computing infrastructures allow upgrading to the latest-and-greatest version of an application without spending precious time and resources rebuilding customisations and integrations. You no longer have to choose between upgrading and preserving all your hard work, because all customisations and integrations are automatically preserved - during and after an upgrade.

@ 2012, IJSAIT All Rights Reserved

5. DISADVANTAGES OF CLOUD COMPUTING [7]

- **Cost**
  - Despite the appearance of cheaper software within the cloud, customers should be aware of what they are actually getting before they buy. Will the software provide you with all of the features needed? How important are these features to the company?
- **Security**
  - Cloud computing is Internet computing. This means that the information is passing through an unsecure network (the Internet) and the information be transmitted should be monitored and regulated accordingly. However, established and reputable cloud service providers are aware of this concern and offer sophisticated data security systems to combat this potential problem.
- **Internet Connectivity**
  - Cloud services do require internet connectivity. No internet connectivity = No cloud services. This issue can be partially remedied by a stable and reputable service provider. However, even the best internet providers will experience down time.
- **Technical Issues**
  - Even the best cloud service provider, regardless of how stringent their maintenance standards, will occasionally experience service outages and technical issues.

6. SOCIAL NETWORKING SERVICES

A **social networking service** is an online service, platform, or site that focuses on facilitating the building of social networks or social relations among people who, for example, share interests, activities, backgrounds, or real-life connections. A social network service consists of a representation of each user (often a profile), his/her social links, and a variety of additional services. Most social network services are web-based and provide means for users to interact over the internet, such as E-mail and instant messaging. Online community services are sometimes considered as a social network service, though in a broader sense, social network service usually means an individual-centred service whereas online community services are group-centred. Social networking sites allow users to share ideas, activities, events, and interests within their individual networks. [8]

Google App Engine was released in April 2008, is a platform for building, hosting and scaling web applications using Google's infrastructure. It allows a developer to build and test web applications without the worry of maintaining servers, and so forth. It's pretty much a plug and plays type of solution, just upload your application, and it's ready to serve your users. A lot of these applications are very raw and are just test beds for the developers. There are quite a few that are useful and have some potential. This post highlights 15 of them. [16]

7. GOOGLE APP ENGINE

App engine is Google's cloud-based platform for hosting web applications written using Java, Python or an experimental implementation of the Go programming language-stated by Mikael Ricknäs[9].

The ability to host applications in Europe was first announced in June, but was then only offered to users with a so-called premier account, which cost $500 per month. Google is now changing that to include all paying users, the company said in a blog in 2012.

Google App Engine allows running web applications on Google's infrastructure. App Engine applications are easy to build, easy to maintain, and easy to scale as your traffic and data storage needs grow. With App Engine, there are no servers to maintain: you just upload your application, and it's ready to serve your users. You can serve your app from your own domain name using Google apps. Or, you can serve your app using a free name on the appspot.com domain. You can share your application with the world, or limit access to members of your organization. [10]

Applications can be developed in several programming languages like Java, Python in Google App Engine. Google App Engine provides Java, Python and Go runtime environments and their corresponding libraries to compile and run the respective codes.

These runtime environments are built to ensure that your application runs quickly, securely, and without interference from other apps on the system. With App Engine, you only pay for what you use. There are no set-up costs and no recurring fees. The resources your application uses, such as storage and bandwidth, are measured by the gigabyte, and billed at competitive rates. You control the maximum amounts of resources your app can consume, so it always stays within your budget. App Engine costs nothing to get started. All applications can use up to 1 GB of storage and enough CPU and bandwidth to support an efficient app serving around 5 million page views a month, absolutely free. When you enable billing for your application, your free limits are raised, and you only pay for resources you use above the free levels. [11]

Google App Engine makes it easy to build an application that runs reliably, even under heavy load and with large amounts of data. App Engine includes the following features [12]:

- Dynamic web serving, with full support for common web technologies.
- Persistent storage with queries, sorting and transactions.
- Automatic scaling and load balancing.
- APIs for authenticating users and sending email using Google accounts.
- A fully featured local development environment that simulates Google App Engine on your computer
- Task queues for performing work outside of the scope of a web request.
- Scheduled tasks for triggering events at specified times and regular intervals.

8. APACHE JMETER

I developed a social networking website using Python on Google App Engine. After developing this web application is tested under different workloads.

I used Apache JMeter to run the tests .The **Apache JMeter™** desktop application is open source software, a 100% pure Java application designed to load test functional behaviour and measure performance. It was originally designed for testing web applications but has since expanded to other test functions. Apache JMeter may be used to test performance both on static and dynamic resources (files, Servlets, Perl scripts, Java objects, data bases and queries, ftp servers and more). It can be used to simulate a heavy load on a server, network or object to test its strength or to analyze overall performance under different load types. You can use it to make a graphical analysis of performance or to test your server/script/object behaviour under heavy concurrent load. [13]

It is used to test web applications for many server types including web requests (http, https). The experiments are conducted on a machine with the following configuration 2.40 GHZ Processor, 4.00 GB RAM, 100MBPS network card running over a 400MBPS Ethernet hub. The operating system is Microsoft Windows 7. The language used is Python with the GAE runtime.

The evaluation for testing the Python services is to test the performance of the applications under different workloads. The attributes like verification, validation that define the performance of the application are scalability, reliability, and resource usage which demonstrate whether a system meets performance criteria. The load tests are modelled to simulate the expected number of users accessing the website concurrently. The stress testing is done to test the applications performance beyond normal users to determine the stability of the application, and break the application by overwhelming its resources [14].

For evaluating the performance of the applications developed with Python language different test beds are designed .Each test plan has to answer the following questions.

a. What is the anticipated normal workload?
b. What is the anticipated peak number of users?
c. What is the good time to load test the application? This may sometimes crash the servers.
d. What is testing intended to achieve?
e. What is the sequence for the test?
    - Functional (low –volume of users)?
    - Benchmark (average number of users)?
    - Load test (maximum number of users)?
    - Test destructively (the hard limit)?

The test bed for testing the performance is based on load tests with varying workloads. The tests are performed for low, normal, and high volume of users. The low and normal users are for 10, and 50 users. The tests handle a peak load of 100 users concurrently for high volume of users. The tests were usually performed everyday in the morning for n days (where n=5). The sequence of the tests is used to handle 3 types of user's functional users that handle less number of requests. The bench-mark for the users is to test for an average of 50 users, and high flow of user requests of about 100.

The reasons to carry out the tests with a limit of 10, 50 and 100 users are based on the limited community of users who concurrently access the application. So the tests are conducted with those limited values to mimic the real time user scenario. The applications are tested during Monday to Friday, morning 9am till 12 noon. The n value is chosen as 5 because the tests are carried during the week days to see the performance of the GAE servers.

The tests are conducted using Apache JMeter as a preliminary evaluation mechanism even though it is believed that results may not be 100% accurate due to minor Java timing errors. However, to test the reliability of the services and JMeter the tests are repeated for 5 days. The reason for testing on weekdays is to check the performance of the GAE services when there is traffic on the network to simulate the requests on the web. The traffic is shaped due to requests on the network where there may be people be watching videos, connecting to heavy audio and video files during the weekdays. This argument is true but the other networks which may be used for commercial or networking purposes may also be shaped by the network providers. Thus I conclude that these results are preliminary tests to check the scalability of the applications using Python language.

The Apache JMeter can be used to test applications using the http or file transfer protocol (ftp) that can create the test plan based on the requirements. The JMeter has a web test plan that has two important components the test plan and a work bench. The test plan is a container to perform tests and the work bench is a container for any test to be performed or a portion of the test to be moved in to the test plan. The test plan has many sub components that can be added to it. When the test plan is right clicked a context menu appears to add the items to a test plan.

The test bed for this experiment creates a load on servers and tests the performance of the services accordingly. A JMeter test creates a loop and a thread group. The loop simulates sequential requests to the server with a delay and a thread group is designed to simulate concurrent load. A load test using the JMeter test plan is to execute a sequence of operations.

The important components of a test plan are thread group, controllers, assertions, listeners, timers, and configuration elements.

The **"thread group"** tells the users number of users to simulate, how often the user requests need to be sent, and how many requests they need to send. There are two types of **"controllers"** samplers and logical controllers. The samplers tell the JMeter to send a request and wait for the result. There are many samplers like http request, ftp request, and JDBC request etc. Logic controller enables to customize the logic of that JMeter follows. The **"assertion"** allows assert whether the results returned from the server are as according the results that we expected. The **"listeners"** provide the information that JMeter gathers when a test is run. The **"timers"** are used to pause between each web request that JMeter send to the server. By default the timer is off. The **"configuration element"** is used to add or modify the requests and works with the samplers [15].

9. GOALS OF EVALUATION

- To evaluate the services by varying number of requests (10, 50, and 100).
- To perform repeated requests for n days and calculate the difference between them.
- To calculate the time taken for each request individually for n days.
- To calculate the time taken by the workflows.
- To test the performance of the services and workflows.
- To study the scalability of services developed in Python and Java.

10. PHASE1: TEST BED FOR THE EXPERIMENTS IN PYTHON

The test bed for the services in Python is to evaluate the performance of the services over a certain time period. Initially each service from both the languages is evaluated for a fixed period of n days where (n=5). The services are evaluated for varying number of client requests with a fixed time difference between each request.

@ 2012, IJSAIT All Rights Reserved

The phase 1 experimental test bed services developed in Python is evaluated with increasing loads for 5 days. Each of the measurements is recorded everyday in the morning 9 am till 12.

11. TEST BED

The test bed is a simple http web request using Apache JMeter and defined in 4 steps. In step1 a thread group is created. The thread group tells JMeter the number of users, how often the requests have to be sent, and how many requests have to be sent. These properties are explained by the fields number of threads (users), ramp-up period (in seconds), and loop count. The field number of threads tells the JMeter the number of user simulations to be created, the ramp-up period in seconds indicates the time delay between each thread. For example if the number of threads is 6 and ramp-up period is 12 seconds then JMeter would send each request with a delay of 2 seconds. The number of loops indicates number of times the requests have to be sent.

In step2, the tasks to be executed by the JMeter are defined. The thread group is selected and mouse right click option is chosen to add a config element the “http request defaults”.

In step3 -a http request is defined by selecting the thread group and choosing a sampler with the name http request. In this page the name, path, port, and method filled. The name, path and port numbers are the same as default “http request defaults” but the method is changed accordingly based on the type of request get or a post. The lists of parameters are sent along with the request depending on the type of request.

In step4, the results of the test are viewed by adding a listener to the test plan. This element is used to store the results of the http request. The listener is added by selecting the test plan an adding a listener and adding an element to view results in a table. These results show the number of the requests, the thread group they belong to, time taken in milliseconds, the result of the request either success or failure.

12. RESULTS OF THE PHASE1 EXPERIMENTS

This section discusses the results for the Python based services that are evaluated using the test bed in section. The services are evaluated for varying number of client requests 10, 50, 100 for 5 days. The tables in the 1 show the services evaluated with varying client requests for Python language.

Table 1: TABLE SHOWING THE LIST OF SERVICES EVALUATED IN PYTHON

Service name	**Request type**
Login	Post
Main	Get
Account	Get
Poststories	Post
Comments	Post

Table 1 shows the services in Python and the graphs for the Python services in the same order as tabulated.

The individual services are evaluated for 5 days for the best and worst performances with the workloads. The graphs for each of services are shown as in the sequence of the tabulated service names. Also, for the tests on the services in Python, maximum, minimum, and average values for the time taken in milliseconds are calculated. Based on these values the delta is calculated.
The delta is the difference between maximum and minimum values. The graphs display variance values for each service in Python.
Delta (a) =maximum (a)-minimum (a)

Where ‘a’ is the column in the table.

Results of the Python services based on the test plan all the services in Python are evaluated for different number loads of client requests (10, 50, 100) for n days where (n=5). The services in Python are evaluated for checking the best and worst performance among the five days with workloads and also for calculating the overall variance.

The first part of the evaluation shows the performance of the services for the workloads 10, 50, and 100 over a period of 5 days the graphs are recorded to calculate the best, worst, and average time taken. The Figure 1, 2, and 3 show the time taken for login service for 10, 50, and 100 workloads. In each graph the best and worst time taken is observed as the best time defines the least time taken to process the workload. The worst time indicates the maximum time taken to process the workloads.

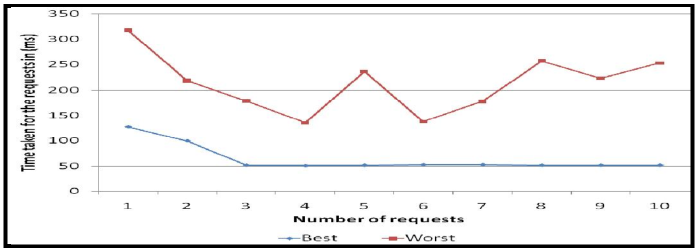


Figure1:The performance of Python login service for workload (10).

The Figure 1 shows the best performance as the least time taken to process the workload of 10 requests.

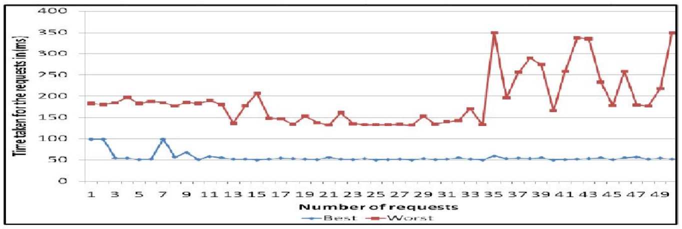


Figure 2-the performance of Python login service for workload (50)

Figure 2 shows the huge difference between the best performance and the worst performance. It is understood that the average performance for the services should be between the best and the worst performance graphs.

@ 2012, IJSAIT All Rights Reserved

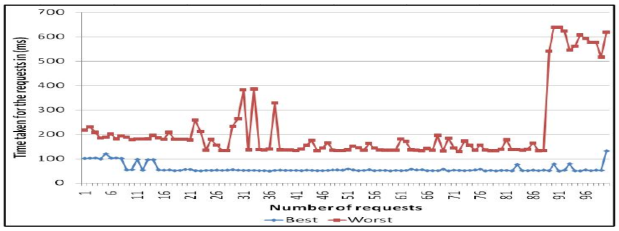


Figure 3-the performance of Python login service for workload (100).

The Figure 3 shows the performance graph for login service for a workload of 100. In this graph the worst performance is indicated by the highest time taken line. It is indicated as worst performance because of increase in workload, the time taken is also increased after 85 requests.

The second part of the evaluation shows the variance of services. For each day the average time taken and its delta is calculated. As discussed earlier delta is calculated as the difference between maximum and minimum time recorded for a day. Similarly the averages and variances for the five days are taken and graph is drawn with the variances recorded from day 1 till day 5(from Monday till Friday).the Figure 4 shows the time taken for login services in Python based on the variances. The 3 different lines indicate the time taken for varying loads with the variance as indicated.

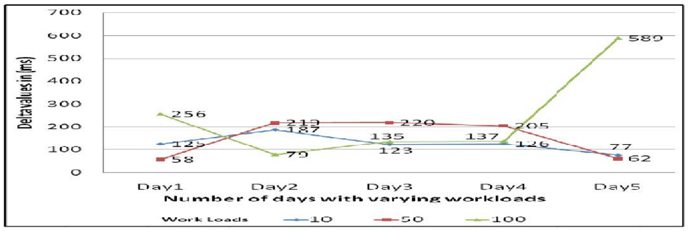


Figure 4-difference of the times taken (ms) using login service in Python for workloads.

The Figure 4 shows the time taken for login service with different workloads. This service performs a post operation. With increased workload the value of delta also increases. For example for the workload 100 the delta value of time taken each day starting with day 1 is high and gradually decreases with the GAE balancing the load by running the application on many servers. When the load on the servers comes down, time taken for the requests also decreases as a result the graph shows rise and fall at certain locations.

Similarly the performance graphs for the main service are shown in Figure 5 till Figure 8. The Figure 5 shows the performance graph for 10 requests.

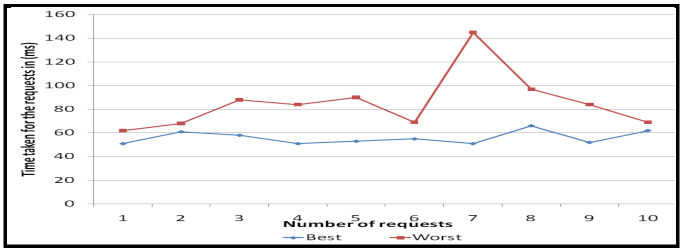


Figure 5-the performance of Python main service for workload (10).

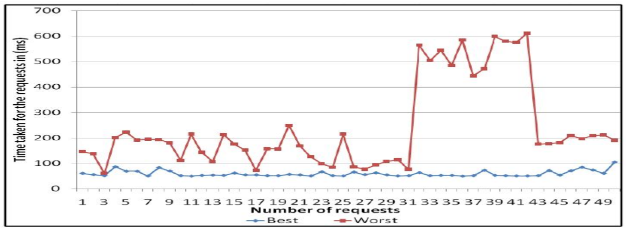


Figure 6-the performance of Python main service for workload (50).

The Figure 6 shows the performance graph for workload of 50 requests.

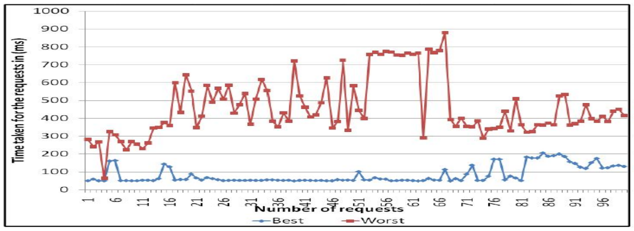


Figure 7-the performance of Python main service for workload (100).

The Figure 7 the performance graph for workload of 100 requests. It performs a get operation.

In the Figure 5, the best and the worst performance doesn't overlap where as in Figure 6 and Figure 7 shows the overlap with a difference in time taken in each of the graphs .but for workload 50, best and worst graph shows a minimum difference between them. Figure 7 shows the time taken for worst performance line shows many fluctuations in a graph indicated with high volume of requests and maximum time taken. But the increased loads in both the Figures are balanced by the GAE servers that run the applications. The servers running the applications have balanced the load by sharing the load with the server next to it thus a lowering line indicating the decreasing load as shown.

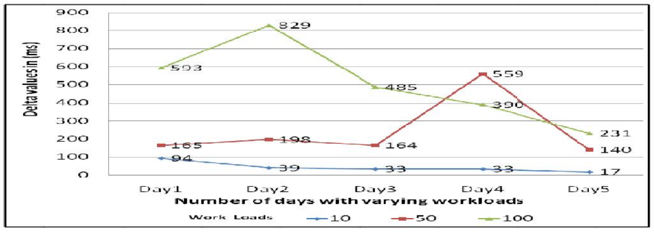


Figure 8-difference of the times taken (ms) for main service in Python for workloads.

The Figure 8 shows the delta value for main service in Python for the workloads (10, 50, 100) in this figure it is observed that the delta value for processing 50 requests on a day 4 is more than the delta value for 100 requests. For 100 requests workload starting at day 3the delta value drops each day. This is due to GAE servers automatically balancing the load after a certain threshold where they distribute the load to the corresponding servers automatically. During this time the overall load on the available servers reduced and the time taken for processing the requests reduces. Using this approach there is a low failure of the requests and fewer the burdens on the server and ultimately time is reduced.

Similarly Figure 9, 10, and 11 show the performances of the accounts service using workloads.

@ 2012, IJSAIT All Rights Reserved

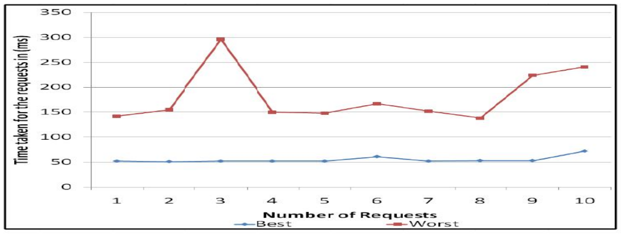


Figure 9-the performance of Python accounts service for workload (10).

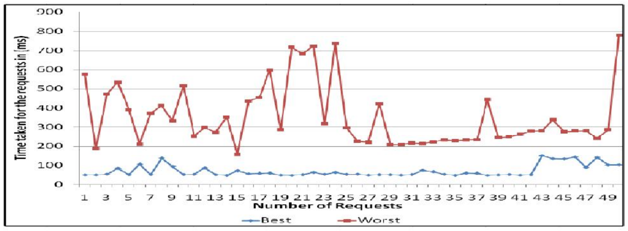


Figure 10-the performance of Python accounts service for workload.

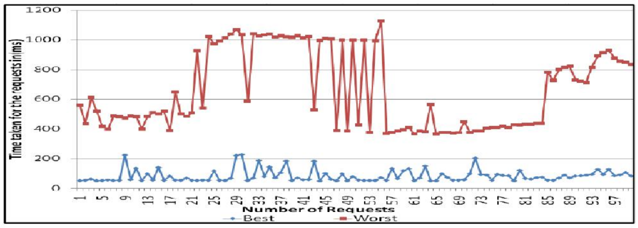


Figure 11-the performance of Python accounts service for workload (100).

The Figure 12 shows the delta values for accounts service for all the workloads. For the account service, the peaks in the graphs are observed similar to main service for the workloads 100 shown in Figure 11.

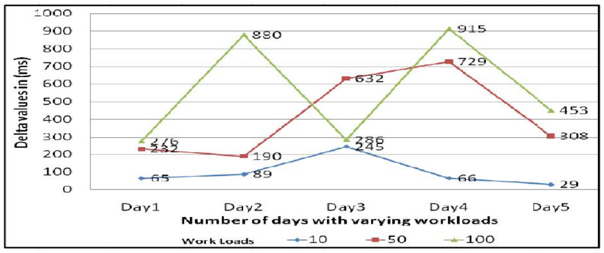


Figure 12- difference of the times taken (ms) for accounts service in Python for workloads.

The Figure 12 shows the delta values for the accounts service. The accounts service performs a get request operation. In this graph the delta values show increase in numbers with increase in workloads but gradually decrease as new servers are fired up to balance the loads. Similarly the graphs are recorded for poststories service. The poststories service performs post operation. The Figures 13 till 16 are the graphs for poststories service.

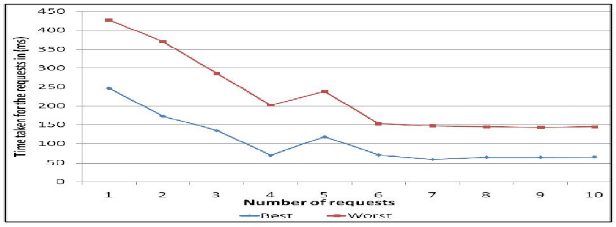


Figure 13-the performance of Python poststories service for workload (10).

The Figure 13 shows the time taken for the requests in worst line is in the same shape as the time taken for the best. This indicates that they are taking constant time with the worst graph taking twice the time of best.

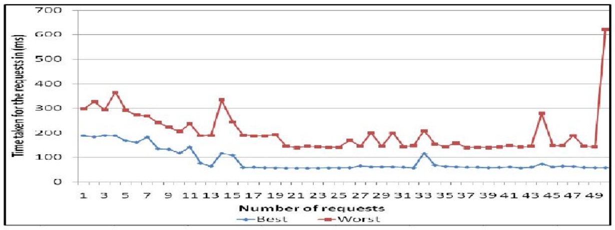


Figure 14-the performance of Python poststories service for workload (50).

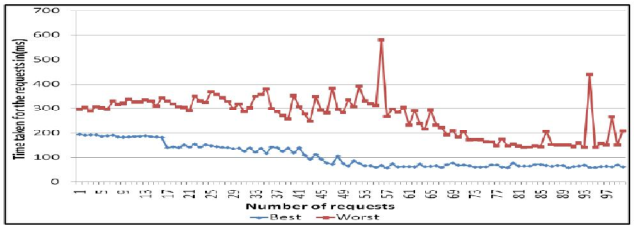


Figure 15-the performance of Python poststories service for workload (100).

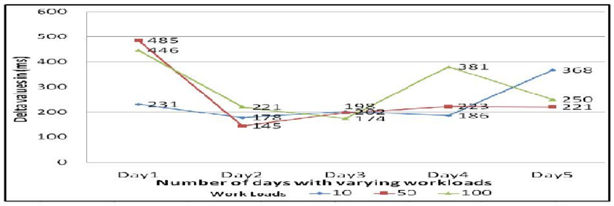


Figure 16- difference of the times taken (ms) for poststories service in Python for workloads.

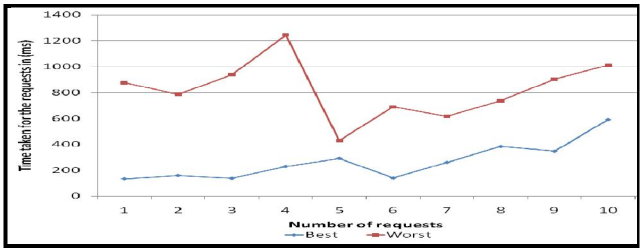


Figure 17-the performance of Python comments service for workload (10)

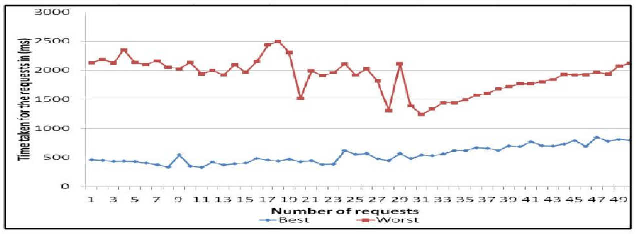


Figure 18-the performance of Python comments service for workload (50).

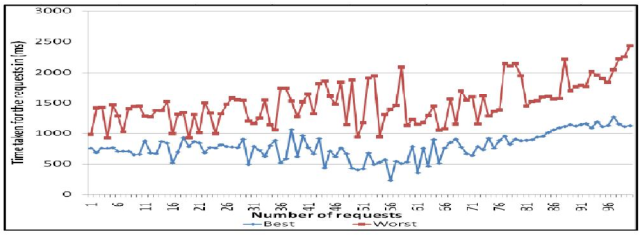


Figure 19-the performance of Python comments service for workload (100).

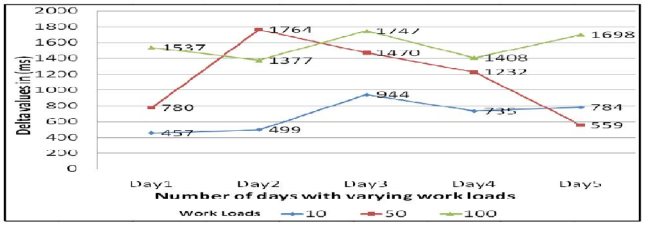


Figure 20- difference of the times taken (ms) for comments service in Python for workloads.

13. PHASE2: WORKFLOWS FOR THE EXPERIMENTS IN PYTHON

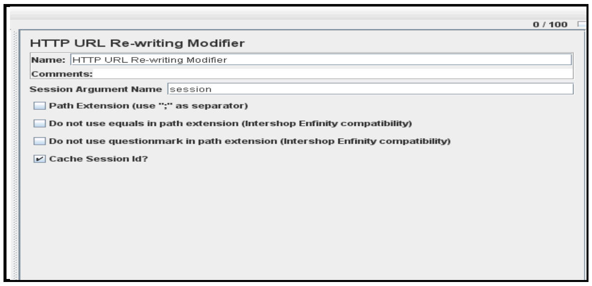
The Python experiments aim at the performance of the services as a workflow. In these experiments a set of services are arranged as a workflow in such a way that users execute the requests as the path mentioned in the workflow. Each time specified number of requests are sent to the workflow for n days where n=5. The workflows are run every day in the morning for n days and graphs are recorded accordingly.

14. TEST BED

The test bed for the experiments using Apache JMeter is explained in this section for the workflows. The workflows are designed separately for the experiments in Python.

The test bed has a sequence of 5 important steps. The test bed is same as the test bed for individual services but with additional simple controllers for session information. The test bed consists of the thread group which is used to define number of users, ramp-up time in seconds and the number of loops. The workflows are tested with workloads 10, 50, and 100. These workloads are tested on everyday for n=5 days. Usually a client logs into the experiment using “login” service which leads into the main service and the client may post a story using the “poststories”. Later the client may post a comment and change the account settings. These operations are defined in the workflow with the requests to different pages. Each page request is defined by the http request defaults and is associated with a http requests. The http requests defaults are having the default values based on which http request is processed.

The simple logic controllers are used in the workflow to organize the samplers and other logical controllers. The simple logic controllers can be added by right clicking on the thread group. In the simple logic controllers http URL re-writing modifier is added where the session information is managed using the variable mentioned. The session id is cached if the option for the cache is checked and can be used for the other services. The results are recorded using the view results in a table which is added by right click option on the thread group. The Figure 21 shows the http URL re-writing modifier.



0 / 100

HTTP URL Re-writing Modifier

Name: HTTP URL Re-writing Modifier

Comments:

Session Argument Name session

☐ Path Extension (use ";" as separator)

☐ Do not use equals in path extension (Intershop Enfinity compatibility)

☐ Do not use questionmark in path extension (Intershop Enfinity compatibility)

☑ Cache Session Id?

Figure 21- the http URL re-writing modifier and its properties.

15. RESULTS OF THE PHASE2 EXPERIMENTS

This section discusses the results of the workflows using Python that are evaluated using JMeter. The workflows are evaluated for workloads 10, 50, and 100. The workloads are recorded for n=5 days as to measure the performance of the GAE services during the days of a week. The workflow is defined as a sequence of service navigation through an application. For the Python experiment the workflow is in the order of the services mentioned in table 1. The first set of graphs shown in Figure 22 till Figure 44 show the best, and worst performances of the workflow in Python for the workloads (10, 50, and 100).

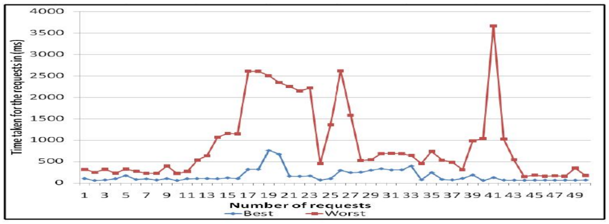


Figure 22-the performance of the Python workflow for workload (10).

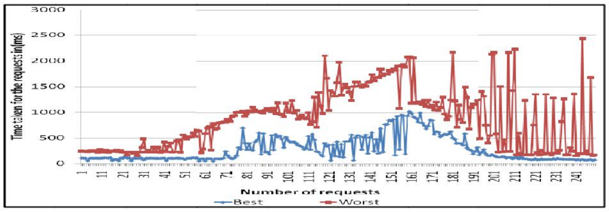


Figure 23-the performance of the Python workflow for workload (50).

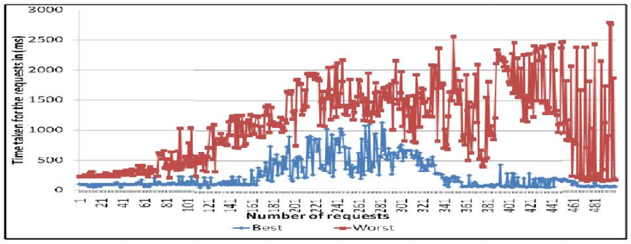


Figure 24-the performance of the Python workflow for workload (100).

For the workflows everyday minimum and maximum values are calculated. The workflows delta values are calculates as the difference between maximum and minimum.

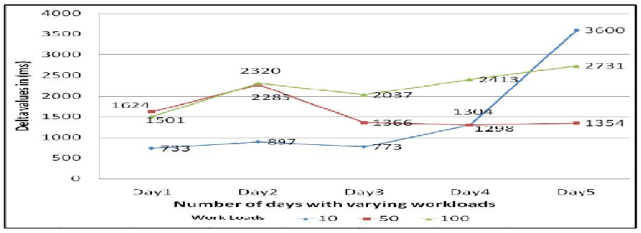


Figure 25- difference of the times taken (ms) for the workflow in Python for workloads.

16. CONCLUSION

A cloud computing platform enables applications to be hosted in an internet-accessible virtual environment that supplies the necessary hardware, software, network, and storage capacities and provides for security and reliability, removing much of the burden of purchasing and maintaining hardware and software in-house. In the cloud, you can develop, deploy, and manage applications as you have in the past and integrate these services to your on-premise applications. You pay only for the time, resources, and capacities you use while scaling up to accommodate the changing business needs.Typically, cloud environments provide low level abstractions of computation and storage. Computation and storage clouds act as building blocks where high level service clouds and mash-ups can be created. Storage clouds are often used to prolong the capabilities of storage-limited devices and provide transparent access to data from anywhere. In this article a web application was developed. The goals of the experiments are to investigate the GAE cloud architecture and its design principles and to develop the application with the basic services. Investigate accessing the applications using different clients. In experiments the MVC architecture guidelines are used.

Experiments are conducted to evaluate the performance of the applications in order to test it with varying workloads for Python. For the purpose of testing a tool is used called 'Apache JMeter'. Although the argument "Apache JMeter is not a best tool for testing the scalability of the applications for concurrent client requests", it is one of the most popular one. The repeated experiments prove that the time taken for the services does not change rapidly.

REFERENCES

[1] http://www.investopedia.com/terms/c/cloud-computing.asp#ixzz2lgpzxqtt
[2] http://www.techterms.com/definition/cloud_computing
[3] A. Mell,t. Grance," **The NIST definition of cloud computing**", available at:Csrc.nist.gov/groups/sns/-cloudcomputing/ cloud-def-v15.doc
[4] http://www.salesforce.com/in/cloudcomputing/
[5] Joe Mckendrick," **Shining examples of cloud computing in action**", Feb 22, 2012, available at -http://www.forbes.com/sites/-joemckendrick/2012/02/22/6-shining-examples-of-cloud-computing-in-action/
[6] http://www.salesforce.com/in/cloudcomputing/
[7] http://www.opticfusion.net/blog/2013/02/01/advantages-and-disadvantages-of-cloud-computing/
[8] http://en.wikipedia.org/wiki/social_networking_service
[9] Mikael Ricknäs," **Google lets more users host App Engine applications in Europe**" December 14, 2012, available at-http://www.infoworld.com/d/cloud-computing/google-lets-more-users-host-app-engine-applications-in-europe-209211
[10] https://developers.google.com/appengine/docs/-whatisgoogleappengine
[11] https://developers.google.com/appengine/docs/-whatisgoogleappengine
[12] https://developers.google.com/appengine/docs/-whatisgoogleappengine
[13] http://JMeter.Apache.org/
[14] Solanki," **Jmeter: a tool for performance testing your webap**", july 3rd 2010, available at-http://www.slideshare.net/amitkssolanki/JMeter-performance-testing-your-webapp
[15]" **The Apache jakarta project**", available at-http://jakarta.Apache.org/JMeter/usermanual/build-web-test-plan.html
[16] Mike Fruchter , "**Useful Google App Engine Applications**", available at - http://blog.louisgray.com/2008/12/15-useful-google-app-engine.html
[17] http://www.webopedia.com/TERM/C/cloud_computing.html
[18] "**The open cloud manifesto: A call to action for the worldwide cloud community**", available at-http://opencloudmanifesto.org/opencloudmanifesto1.
[19] http://cordis.europa.eu/fp7/ict/ssai/docs/cloud-report-final.pdf