PAAS Framework Implementation of Cloud Computing With Google Application Engine - A Review

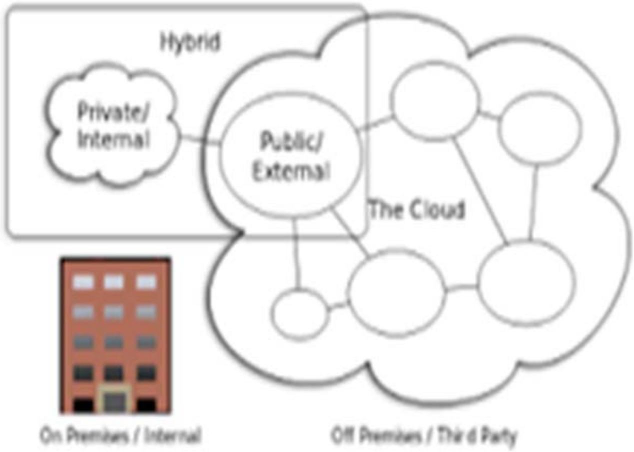
Amrita Sekhon
Assistant Professor and Head
Department of Computer Science
ASBASJS Memorial College, Bela (Ropar) Punjab INDIA-140001
Contact No 9463680940

Abstract- Cloud Computing is a paradigm shift that implies computing using the Internet by sharing computer resources in place of using storage or software of a local PC. It is the latest fab of the computer world. Amazon, Google, IBM and Microsoft are the big players in the game of cloud computing. Amazon has currently built the world's largest marketplace and is a major mover in this revolution of cloud computing. It is hosting a large number of other businesses on its cloud service infrastructure. Microsoft too entered the cloud computing scenario thus simplifying migration for all types of Windows. Cloud computing provides virtualized and dynamically scalable resources as a service across the Internet. According to the official definition given by NIST(National Institute of Standards and Technology) in 2011, "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." The NIST defines three service models of cloud computing namely Software as a Service (Saas), Infrastructure as a Service (Iaas) and Platform as a Service (Paas). This paper tries to evaluate the pros and cons of implementing cloud computing as Paas framework with GAE (Google Application Engine).

Keywords: Cloud Computing, Google Application Engine (GAE), Software as a Service (Saas), Infrastructure as a Service (Iaas), Platform as a Service (Paas), Google Computing Engine (GCE), Content Delivery Network (CDN)

INTRODUCTION

The concept of cloud computing indeed fulfills the need of the hour of IT (Information Technology) which is to add capabilities and increase capacity of a network on the fly without investing in any additional infrastructure or providing training to personnel or purchasing additional software. Cloud computing covers all pay-per-use services or subscription based services over the Internet in real time to extend its capabilities. Amazon's EC2 (Elastic Compute Cloud) and AWS (Amazon Web Services), Microsoft Azure and Apache Hadoop are few popular cloud services available. The Amazon uses an automated web based management system to provide an infrastructure as a service (Iaas) eg. storage service (disk or database) or virtual server or network end points of preferred capacity. Considering the point of deployment, cloud computing platform incorporates three types of clouds namely Private, Public and Hybrid cloud. Private cloud implies that its infrastructure is owned or leased and managed by a single organization. Public cloud (eg. Internet) implies that its infrastructure is owned by some cloud service sales organization which further sells its services in the industry or public arena. Hybrid cloud involves more than one type of independent clouds which are combined using some special techniques and standards such that the applications and data are transparent.



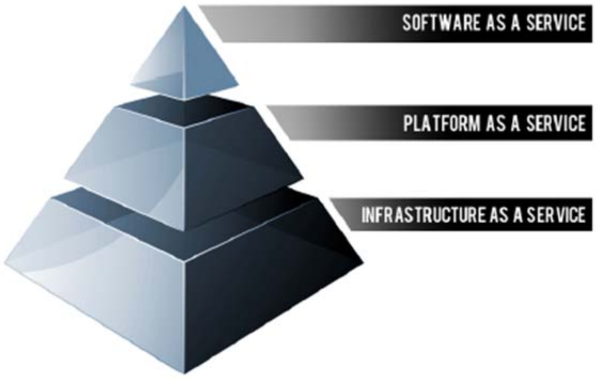
TYPES OF CLOUDS

The three favors of cloud computing are SAAS (Software as a Service), PAAS (Platform as a Service) and IAAS (Infrastructure as a Sevice).

Saas (cloud application service) is growing rapidly and is the largest cloud market. It uses Internet to deliver applications to clients who handle the interface while the applications themselves are managed by third party vendors. Generally Saas applications donot need any downloads while some need a few plugins and are run directly through web browsers. Healthcare related applications, customer relationship management and emails are provided as Saas. It provides users the software which is hosted by third party service providers. The application along with its data is stored on the third party's server. This is very convenient for most of the users while is unacceptable to those who seek more controlling power over the application and data. Examples of Saas include Microsoft 365 and Google Apps.

Paas (cloud platform service) is a computing platform that enables users to build fast and easy web applications without the need for buying or maintaining software and its entire infrastructure. Paas enables development, deployment and testing of applications cost-effectively too. Third party vendors manage storage, servers, networking and virtualization in addition to Paas software, while developers simply manage their applications. This middle level layer provides a framework for developing applications to users without the need to bother about the low level system management. The infrastructure management is automatically done for the users while they focus on their applications. Heroku, Google App Engine, Windows Azure, AWS Elastic Beanstalk, Force.com, Apache Stratos are examples of Paas.

Iaas (cloud infrastructure service) delivers cloud computing infrastructure such as storage, server, OS and networking facility as a service on-demand. These services are fully outsourced on demand to users who pay for services based on consumption. Iaas users manage data, applications, OS, middleware and runtime while third party providers manage storage, servers, networking and virtualization. Users are provided with the low level resource management controls (like distributed storage, VM provisioning, etc) and users are free to development their own application stack (database, HTTP server, OS, MVC framework, etc). although this provdes great flexibility to users, but it also implies that users have to manage fault tolerance and scaling features themselves. Microsoft Azure, AWS (Amazon Web Services), GCE (Google Computing Engine), Cisco Metapod, Rackspace and Joyent are examples of Iaas.



Cloud Computing Stack

GOOGLE APPLICATION ENGINE

Google Application Engine is a Paas cloud computing development environment for creating, managing and hosting web applications provided by Google. It is not just a framework but the complete platform as a service. Google App Engine virtualizes applications across multiple servers. It provides automatic scaling for web applications, ie when the number of requests for an application increases; GAE proportionately allocates more resources to the web application to handle the auxiliary demand. GAE provides a large number of functionalities like dynamic web serving with total support for the common web technologies, persistent storage along with queries, sorting and transaction facility, automatic load balancing, scheduled tasks to trigger events at regular intervals and specified times, task queues to perform work outside the scope of certain web requests, full featured local development environment to stimulate the App Engine on a PC and a range of APIs (Application Programming Interfaces) to authenticate users and send e-mails using Google accounts. The App Engine is a trustworthy way to host websites as the websites are served via Google's own data-centers.

The Google App Engine is free for some limited number of resources, while it charges a fee for each additional storage, bandwidth or CPU cycles as required by the application thereafter. Developers can freely use the preview release of GAE (Beta version) with few restrictions viz. they can serve applications with five million page views per month limit and 500 MB storage size consumable for free. Users pay for only the resources that they use. The resources used by the application are measured in terms of Gigabytes. Thus the App Engine is free for low usage.

GAE currently supports Java, Go, Python and PHP as programming languages to create a Google cloud. Applications can be built in Java using Maven, Go using Template/HTML package, Python using Ninja2 and WebApp2 and in PHP using Cloud SQL. Owing to its Java runtime environment, apps can be built with standard Java technologies like Java Servlets, JVM or languages such as JRuby, Clojure, Groovy, Scala and Rhino, which use JVM based compiler or interpreter. Users can host a company's website, host a live chat application using ChannelAPI (Comet for App Engine) , host a blog using Bloog or some other Python coded Blog software and do many other things using the App Engine.

GOOGLE APPLICATION ENGINE VS GOOGLE COMPUTING ENGINE

Google Cloud Platform offers cloud computing services to built websites and applications, store and analyse data on Google's infrastructure. Google Application Engine and Google Compute Engine are parts of the Google Cloud Platform.

Google Application Engine is a Paas. User simply needs to deploy his website and everything else is done by the platform eg. if that application (website) becomes successful ie traffic goes high, GAE automatically acquires more instances to handle the increased volume. User focuses on the application domain using the platform provided and the platform itself is completely managed by the cloud service. After the free quota available, users are charged for storage used, data transferred and instance-hours, etc. User need not manage the Operating system of the underlying software. GAE is ideal for inexperienced users. Two significant drawbacks of GAE over GCE are firstly it is comparatively expensive and secondly users have less control over certain things or have only one specific way to perform a certain thing.

Google Compute Engine is an Iaas. GCE provides a complete infrastructure in the form of a Virtual machine. User has to create and manage his own virtual application himself. User has complete control over the VM environment (eg. operating system, web server, database, etc). GCE actually uses Google Data Centers virtually. User has to manually configure his infrastructure by using Load Balancer to handle scalability. GCE provides more flexibility to apps and costs lesser than GAE. So GCE provides flexible and scalable virtual machine computing facilities in the cloud.

HOST WEBSITE ON GOOGLE APPLICATION ENGINE

Using GAE one can run web applications on Google servers. Users use CDN (Content Delivery Network) to host the webpage content which includes static content, downloadable files and image stylesheets, etc. Users can run their CDN freely (for limited content) on Google servers. Google allows users to easily scale and integrate their content with other Google applications. GAE was developed by Google after the grand success of Amazon's web services viz. EC2 (Elastic Compute Cloud) for computing and S3 (Simple Storage Service) for storage which are very powerful but require superior technical skills to use as compared by GAE. The following steps are required to use GAE as your personal CDN:

1. A programming language supported by the Google App Engine needs to be downloaded and installed eg. Python 2.5.2 from http://www.python.org/download/.
2. The Google App Engine SDK (Software Development Kit) may to be downloaded from http://code.google.com/appengine/downloads.html and then installed on the PC.
3. User needs a GMail account to sign up for the Google App Engine at appengine.google.com.
4. The users may create an application by simply clicking the 'Create an Application' button and give a unique name to the application (Application Identifier, which uses just lowercase alphabets and digits). User has to specific his/her cell phone number which is used to get an SMS with some code from Google, which confirms you as the owner of the Google App Engine account and is used to enter in the site.
5. 'Website.zip' file may be downloaded and unzipped, which contains sample webpages, images and some CSS (Cascading Style Sheets) to try to host on the GAE.
6. The 'app.yaml' file may be opened in a text editor (eg Notepad) and the word 'labnol' needs to be replaced with the Application Identifier specified earlier. This is used by the App Engine to correctly recognize the user application for uploading the files.
7. Modifications may be done by adding new files, images, and videos etc. which are to be uploaded to Google in this folder where the website was unzipped.
8. GAE Launcher program may be opened from Start menu to upload or deploy this website by choosing File -> Add Existing Application and browsing the folder to be uploaded. Clicking the 'Deploy' button and entering the Google account details (username and password) would make the website instantly available on < Application Identifier>.appspot.com.
9. To edit or modify the webpages of the website, 'Deploy' button needs to be clicked and after entering the Google details, changes may be made and finally uploaded to the App Engine again.

Your very own personal CDN is hosted as a sub domain of the appspot.com. The GAE allows you to register a web domain exclusively yours which may be mapped to the App Engine website. The statistics like disk space and bandwidth used by the user application may be analyzed from appengine.google.com.

FEATURES AND ADVANTAGES OF USING GOOGLE APPLICATION ENGINE

1. GAE provides SDK (Software Development Kit) to create applications locally.
2. GAE focuses on the facility to create apps quickly by providing its users a development server with high level of services and a simplified deployment mechanism. GAE is fully responsible for hosting the user application and running it for the user.
3. It provides a development platform for users who do not want to bother about the configuration and maintenance of the infrastructure.
4. No server maintenance is required. Users do not need to hire a database administrator or a system administrator, thus saving costs in the end.
5. GAE provides a fixed development model for users to adopt and accept as such. Everything is pre-confirmed and ready to use for web app development.
6. GAE provides automatic scaling to apps. To meet the needs of the programmer it increases the number of servers more users join in and also kills the unneeded servers when the traffic goes down.
7. GAE provides a free quota in its currently running beta version. Users can run web applications with low traffic and capacity for free during and after free trial period. As on May 2016, the free quota per day includes 5GB Cloud storage, 50k read/ write on NoSQL Cloud Datastore, 1GB network traffic outgoing, 1GB network traffic incoming, 1000 basic search operations, etc.
8. Users have to pay only for what they use. The App Engine provides a very fair billing system according to the traffic load and database usage above the free quota to use computing resources.
9. GAE application uses Content Delivery Network replication automatically.
10. GAE provides a number of Google services for apps to integrate with like Blobstore-to store large binary in database, Capabilities-to detect scheduled downtime and outages, Images- for image manipulation, Logs-for writing logs, OAuth-to grant third party permission, Prospective Search-a pre-build search query, Mail-for sending mails and many more.
11. Google is known to be the world's best network in terms of reliability and speed with data transfer rate of

300Mbits/sec and latency of 20ms. GAE uses Google's high speed network.

12. GAE uses secure Internet infrastructure of Google to store applications and data safely.
13. GAE provides a large option of programming languages to choose from like Java, Go, Python and PHP. Also other languages like JRuby, Clojure, Groovy, Scala and Rhino can run on GAE using the JVM (Java Virtual Machine).
14. The applications are 'sandboxed' (ie they donot obstruct one another) and are run by many servers by the Google App Engine.
15. Applications hosted on the App Engine are forbidden to perform certain actions like making OS calls, opening a socket connection or writing to a local file.
16. GAE provides compatibility feature as well. Users can switch the GAE application to other SDKs that are open source (like TyphoonAE, AppScale) or other platforms like Amazon EC2. So GAE can be integrated with multiple IDEs (Integrated Development Environments).
17. GAE uses NoSQL Datastore which is a schemaless object datastore having a SQL like query language, a vivid data modeling API and scalable storage.
18. GAE provides a Google Cloud SQL which is a fully managed web service allowing users to create and configure relational databases on the Google Cloud.
19. GAE provide Memcache which is an in-memory, distributed data cache used for improving the application performance greatly.
20. GAE provides Task Queues which are used by applications to work outside user requests by using discrete, small tasks to be executed later.

LOOPHOLES IN GOOGLE APPLICATION ENGINE

1. At present one cannot implement all kinds of application in Google App Engine as it is still in its beta version.
2. GAE is a Paas, so users have absolutely no control over the environment (syntax / stdlib, APIs, language modules, etc) in which their application runs. The OS cannot be accessed and no administrative tasks can be performed.
3. Despite powerful and simple API (Application Programming Interface) provided by GAE, it is indeed not flexible enough to satisfy all customer needs. There is, for instance, a 1MB restriction on the datastore and in case of Blobstore API, users cannot read, modify or update bits to blob after uploading.
4. Database service of GAE is not fully satisfactory from speedy data recovery point of view. It may take 250 msec time for reads and writes at times. Users face write- time out flaw as well.
5. Migrating from GAE to some other Paas service is quite cumbersome, especially for large applications as code to use task queues, datastore or other services needs to be modified for proper compatible functioning. More often users have to start from scratch.
6. Users have read-only access to filesystems on GAE.
7. Many languages like Pyrex and C are not supported, so their code modules cannot be uploaded and used.
8. GAE executes code called from HTTP Request only and executes some other scheduled background tasks.
9. Java applications can use only a subset of classes of JRE (Java Runtime Environment) Standard Edition (ie. JRE Class White List).
10. The Java applications used in GAE cannot create new threads.
11. Many of the existing web applications cannot run on GAE without modification as they require relational databases.
12. It is free for limited usage or small business owners but for more number of resource requirement and high spikes of traffic, it becomes quite expensive (ie. after the free quota expires)
13. It is difficult to make clean up in the datastore as it is reasonably slow.
14. GAE does not provide proper support for uploading Local database to Server Datastore.
15. GAE does not have a mature database as it has a complex primary key and complex java.awt.* support presently.
16. In case the Google server goes down, the applications will be affected accordingly.
17. A long list of policies provided by GAE must be read carefully eg. violation of the Google AdSense has resulted in bans commonly.
18. All pricing changes and company data information, etc are handled by the Google company itself and may make changes anytime.
19. GAE does not support geographic distribution of application servers, SD storage and low latency communication to virtual machines.
20. GAE doesnot support sockets so it cannot access e mails through POP3 (Post Office Protocol) or IMAP (Internet Message Access Protocol). It supports only standard URL (Uniform Resource Locator) fetch so Heroku, Rackspace need to be used.
21. Applications on Cloud run on shared infrastructure and the same works with GAE too. If some malicious software or Spam developed by some hacker is running beside your app in the same IP address, then this IP can be blocked for security reasons and you would suffer too.
22. It is expensive compared to other platforms. One instance on GAE is more expensive and performs less work on a comparative basis.
23. User gets 'locked-in' the Google infrastructure. Then user is in the hands of Google- user simply gets proprietary access to everything. Thus if a user has to port his application to some other platform, the entire data will be locked in the BigTable in GAE.
24. GAE takes 30 to 60 sec to complete a request which is a very annoying limitation.
25. GAE cannot be accessed through FTP (File Transfer Protocol). Only HTTP and HTTPS can be used to access GAE.
26. There is no straightforward way for performing massive updates to Google App Engine.

CONCLUSION

This paper tries to evaluate the implementation of cloud computing in Paas framework using Google Application Engine. GAE enables users to build web applications using Google's infrastructure. The apps are easy to build and maintain and are automatically scaled according to the traffic and data storage requirements. Users simply upload the app and it is ready to use leaving the platform handling to Google Cloud. GAE is a great tool and an ideal platform for startups and for small business owners but for the experienced ones who seek flexibility, this is certainly not the right thing. One way to tackle this situation is to use Google App Engine for some part of the application and traditional technology for the other part. There are risks involved in leaving everything in the hands of Google. A number of limitations persist in the beta version of Google App Engine which after overcoming would make GAE fit in the competitive world of today.

REFERENCES

Bedra, A. (2010). Getting started with Google app engine and Clojure. *IEEE Internet Computing*, *14*(4), 85.

Ciurana, E. (2009). *Developing with google app engine*. Apress.

De Jonge, A. (2011). *Essential app engine: building high-performance java apps with google app engine*. Addison-Wesley.

García-Gómez, S., Jiménez-Gañán, M., Taher, Y., Momm, C., Junker, F., Bíró, J., ... & Strauch, S. (2012). Challenges for the comprehensive management of Cloud Services in a PaaS framework. *Scalable Computing: Practice and Experience*, *13*(3), 201-214.

Google.com. Developer's Guide. http://code.google.com/appengine/docs/ whatisgoogleappengine. html, Accessed on 9 Feb, 2009.

Guermeur, D., & Unruh, A. (2010). *Google App Engine Java and GWT Application Development: Build Powerful, Scalable, and Interactive Web Apps in the Cloud*. Packt Publishing Ltd.

https://cloud.google.com/appengine/?csw=1, Accessed on 25 May, 2016.

https://cloud.google.com/free-trial/docs/free-trial-quotas, Accessed on 25 May, 2016.

http://cs.uwec.edu/~buipj/teaching/cs.491.s13/lecture_13_google_app_engine_windows_azure.html, Accessed on 25 May, 2016.

Kim, W. (2009). Cloud Computing: Today and Tomorrow. *Journal of object technology*, *8*(1), 65-72.

Lawton, G. (2008). Developing software online with platform-as-a-service technology. *Computer*, *41*(6), 13-15.

Malawski, M., Kuzniar, M., Wojcik, P., & Bubak, M. (2013). How to use Google App engine for free computing. *Internet Computing, IEEE*, *17*(1), 50-59.

Managing framework for PaaS in Cloud Computing. *IJCSI International Journal of Computer Science Issues*, *8*(5), 474-479.

Prodan, R., Sperk, M., & Ostermann, S. (2012). Evaluating high-performance computing on google app engine. *Software, IEEE*, *29*(2), 52-58.

Qayyum, J., Khan, F., LaL, M., Gul, F., Sohaib, M., & Masood, F. (2011). Implementing and Zahariev, A. (2009). Google app engine. *Helsinki University of Technology*.

Sanderson, D. (2009). *Programming google app engine: build and run scalable web apps on google's infrastructure*. " O'Reilly Media, Inc.".

Severance, C. (2009). *Using Google App Engine*. " O'Reilly Media, Inc.".

Wang, L., Von Laszewski, G., Younge, A., He, X., Kunze, M., Tao, J., & Fu, C. (2010). Cloud computing: a perspective study. *New Generation Computing*, *28*(2), 137-146.

Wikipedia.com. Cloud Computing. http://en.wikipedia.org/wiki/Cloud-computing, Accessed on 20 May, 2016.

Wikipedia.com. Google App Engine. http://en.wikipedia.org/wiki/Google-App-Engine, Accessed on 20 May, 2016.