Google App Engine

Alexander Zahariev
Helsinki University of Technology
`a.zahariev@abv.bg`

Abstract

Cloud computing becomes more popular day by day. The "dream" of users to "to get rid" of administrators, huge investments on expensive hardware and human resources becomes true. The goal of this paper is to present a general overview of the new coming service offered by Google company - Google App Engine. "What is it for?", "What can I do with the system?" and "How does it work?" are some of the questions on which this paper will try to give an answer. In addition, brief overview and description of "cloud computing" and comparison of Google App Engine with other similar products will be provided to the reader as well.

KEYWORDS: Internet, Web Based Application, Web Development, Cloud Computing

Figure 1: The Cloud [9]

1 Introduction

Before we start, lets give a brief explanation of what is "Cloud Computing?" Computing model in which different tasks are assigned using combination of connection, software and services accessed over a network is called "Cloud Computing". And this combination of connections and servers is known as "the cloud"(Fig.1). With simple words using this technology, users are able to access supercomputer-level power using thin connection or other access point like iPhone, BlackBerry or laptop whenever they need them. Because of this "Cloud Computing" is also described as "On-Demand Computing"[8].

One of the major advantages is that users don't need to have any pervious understanding, knowledge, proficiency how to control technology infrastructure or "the cloud", which supports them. This concept join together several technology trends associated to internet which main goal is satisfying the computing needs of the customers. These technology trends are Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (Saas). We are going to focus on Saas technology and two major examples of Saas vendors are Salesforce.com and Google Apps. Both of them provide universal commerce solutions online that are accessed from a web browser, and at the same time software and data are stored on the company's servers. The name "cloud" is used as a symbol for the Internet, based on how the Internet is visualized in computer network diagrams [10].

This paper is divided into 6 sections. Sec.2 explains actually what is Google Application Engine. Sec.3 Reveals how the system w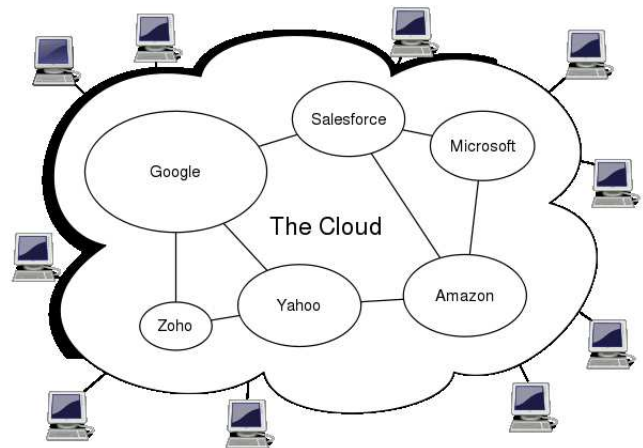orks and the requirements for it. Sec.4 shows what can be done with the system. Sec.5 presents brief comparison between the three leading products in this area - Google App Engine, Amazon Web Services and Microsoft Azure Service Platform. And finally in Sec.6 some discussions, conclusions and summarize of this paper will be provided to the reader.

2 What is it for?

So what is Google App Engine? According Kevin Gibbs which is App Engine Tech Lead, Google App Engine is "a system that exposes various pieces of Google's scalable infrastructure so that you can write server-side applications on top of them" [3]. Simply this is a platform which allows users to run and host their web applications on Google's infrastructure. These applications are easy to build, easy to maintain and easy to scale whenever traffic and data storage needed. By using Google's App Engine, there are no servers to maintain and no administrators needed. The idea is user just to upload his application and it is ready to serve its own customers. User has a choice either his product to be served by the free domain appspot.com or to allow Google Apps to serve it from domain chosen by the customer. Google also provide the user with the option to limit the access of the application within the members of his own organization or to share it with the rest of the world. The starting packet is free of charge and additional obligation. All the user have to do is to sign up for a free account, and then to develop and publish his own application. The starting package includes up to 500MB of storage and enough CPU power and bandwidth to

serve 5 million page views per month [1].

3   How does it work?

This section provides more detailed information about how the system actually works, and what kind of requirements it has to create an application.

3.1   The Application Environment

With this new service provided by Google it is really easy to create reliably applications which runs under heavy load and which use large amounts of data. Several key features are included in the environment[1]:

- 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 user's computer

Implementation of Google App Engine applications is done under Python programming language. Full Python language support along with most of the Python standard library comes with standard runtime environment. Currently Python is the only supported language by Google App Engine, but improvements to support other languages are in progress[1].

3.2   The Sandbox

All user applications operate in a secure environment. This environment has a limited access to the underlying operating system. Because of these limitations, App Engine is able to distribute application's web requests across various servers, which allows starting and stopping the servers to meet traffic demand. The sandbox separates the application in its own protected and reliable environment which is independent of the operating system, hardware or the physical location of the web server. Here are some of the restrictions which are included in the sandbox environment[1]:

- An application can only access other computers on the Internet through the provided URL fetch and email services and APIs. Other computers can only connect to the application by making HTTP (or HTTPS) requests on the standard ports.

- An application cannot write to the file system and can read files, but only files uploaded with the application code. The application must use the App Engine datastore for all data that persists between requests.

- Application code only runs in response to a web request, and must return response data within 30 seconds. A request handler cannot spawn a sub-process or execute code after the response has been sent.

3.3   The Python Runtime Environment

The Runtime environment provided by App Engine uses Python programming language, but other languages as well are taken into account to be supported in the near future. The current supported version of the Python runtime environment is 2.5.2 [1], which includes the standard Python library. All methods, except those ones which violates the sandbox restriction, like attempting to open a socket or write into a file, are included in the library. The modules which features are not supported by the standard are disabled, and the part of code which imports them will raise an error. All applications code must be written entirely only in Python language and code with extensions written in C is not supported.

Rich APIs for the Google Accounts, URL fetch, Datastore and email servers are provided by the Python environment. Simple web application framework called "webapp"[1] which makes easy to start building applications is provided as well by App Engine. In addition Django Web application framework, version 0.96.1 [1] is also included in App Engine, but it should be take into account that App Engine datastore is not a relational database, which is required by some Django components. The option to be included other third-party libraries is also supported. As long as they are implemented in Python programming language and if they not require any unsupported library modules, it should not be a problem to be used [1].

Now brief description of all APIs for the following services in Python Runtime Environment will be provided.

3.3.1   The Datastore

A powerful distributed data storage service is present by App Engine. A query engine and transactional storing accessible through a simple API, both running on Google's scalable infrastructure is provided with the App Engine datastore. This Python interface includes a data modeling API and similar to Structured Query Language (SQL) called GQL. Using these features developing data dependent applications should not be more difficult than creating it using normal web hosting service [12].

Similar like distributed web servers which grow with the amount of traffic, the distributed datastore grows as well when the amount of data raise. Unlike other traditional datastore, the App Engine datastore supports set of properties in data objects also known as "entities" [1]. Like any other data filters, data can be extracted from tables using queries filtered by property values.

Data modeling interface which can define structure for datastore entities is included in the Python API for the datastore. This data model specifies if property value must be within a given preset range or will assign a default value to it if not given. User application supports as much as needed little structure of data. The Integrity of data is very well guaranteed. Each application can execute multiple datastore operations in a single transaction which either succeed or fail. Concurrency control is very well ensured as well. Update to the any single data record occurs if another process is trying to access the same data record in the same time [12].

3.3.2   Google Accounts

A service API which integrates with Google Accounts is included in App Engine as well, and it provides better user control. Customer's applications can allow users to sign in their product by using Google accounts instead of implementing their own user account system. User can also distinguish whether the signed person is a simple user or a registered administrator for the application. Benefits of this are faster deployment of the product and faster and easier access to it [1]

3.3.3   URL Fetch

Resources on the internet, like web service or other data for example, can be accessed by any user application thanks to App Engine's URL fetch service. This service is fast as other Google services and can retrieve other web sources by using high-speed Google infrastructure [1]. It allows App Engine applications to exchange data with other hosts using HTTP requests [14].

3.3.4   Mail

Every user application can send message using App Engine's mail service, which also uses Google's company infrastructure [15]. This is very useful way for applications to communicate with users and notify other users for any activity or updates [15].

3.3.5   Memcache

Distributed in-memory data cache in front of or in place of reliable constant storage is often use by high performance scalable web applications for some tasks. Because of this reason Google App Engine supports memory cache service [13]. The Memcache service assures user applications with a high performance in-memory key-value cache that is available by numerous instances of the applications.

Data that does not need the persistence and transactional features of the datastore, like for example a temporary data or data copied from the datastore to the cache for high speed access, which makes most of the Memcache service usage [1].

3.3.6   Image Manipulation

Google provide as well and Image service which supports manipulating image functions, like resize, crop, rotate and flip images both in JPEG and PNG formats [1]. This is very useful service for customer's applications which are dealing with photographs or just provide users with a service to prepare their avatars.

3.4   Development Workflow

A web application which emulates all of the App Engine services on the local computer is included in the App Engine software development Kit (SDK). All of the APIs and the libraries available in App Engine are included in it. Simulations of the secure sandbox environment, which includes
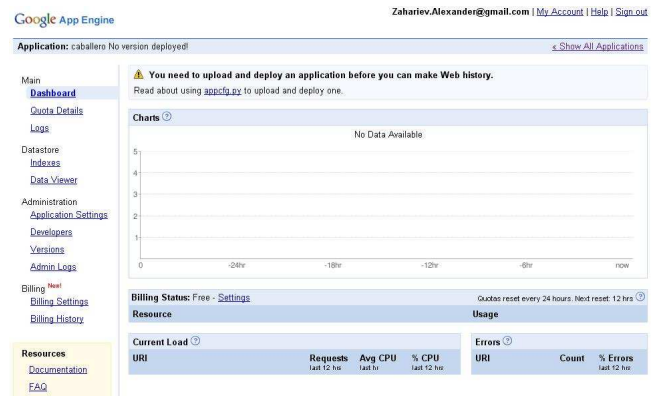

Figure 2: Admin Console view

check for imports of disabled modules or attempt to access not allowed system resources, are supported by the web server. Implementation of the Python SDK is done in pure Python programming language and it is operate on any operating system, including Windows, Mac OS and Linux which supports Python 2.5 release [2]. The Python software package is available on Python web page. The Google App Engine SDK can be obtained from the Google App Engine homepage, as a ZIP file, or as an installer available for Windows and Mac OS X. Tool for uploading the applications to App Engine infrastructure is included in the SDK. Each application consist two types of files, static files and configuration files. Once these files are ready, they can be uploaded by the tool, which also prompts the user about his Google account e-mail and password.

Very useful feature is that Google App Engine supports version of the applications. If customer develops new major release of his product, he can upload the new release as a new version, while the old version is still in use. Final tests of the newer release can be done in same time, before it can be switched on.

Management of the applications running on App Engine is done through an administration console (Fig.2). This a web-based interface which allows the customer to create new applications, configure domain names, change which version of the application is in use, examine access and error logs, and browse an application's datastore.

3.5   Quotas and Limits

Creating an App Engine application is not only easy but it is also free. The only thing user should do is to create an account and publish his application, all of this at no charge and without any obligations. Free accounts can support up to 10 applications, maximum of 500MB of storage for all of them and maximum 5 million page views per month again for all of them. If user decides to use more computing power, he can enable billing function, set a maximum daily budget, and assign this budget only to recourses he needs. Applications cannot use recourses from another account, and also according company's policy, operating applications on multiple free accounts that work in tandem is not allowed as well.

One good feature is that limitations, or "quotas", are con-

tinuously refreshing. At the beginning of each calendar day, a time-based quota will be refreshed if application reaches it. But fixed quotas for example like storage use are only reassured when user decreases usage.

There are other limitations which are not related to the quotas but which assures the stability of the system. As an example if the application requires serving a web request, it needs to bring back a respond within 30 seconds, otherwise the process will be terminated and it will raise an error code. If an application continuously reaches limitations of the timeouts, the time for timeouts will be reduced, to help keep system resources.

Another example of a service limit concerns the data store. Single query can return maximum 1000 data records as a result. If there is a possibility that the query will return more than 1000 records, it will return the first 1000 as a result. But it is very possible that such query will affect the 30 seconds timeout limitations which will raise an error. All of these limitations are done in respect to conserve resources on the datastore.

4   What can I do with the system?

In practice there are no limitations what can be done using the system. User can choose either to build a simple application or to create a powerful commercial product. On Google App Engine home page there is section which name is "App Gallery", in which many applications built with App Engine can be found. The visitors can search for applications in many sub categories like, Tools, Communications, Games, News, Finance, Sports, Lifestyle, Technology and Enterprise. There is also editor's choice section in which best ranked applications can be selected.

A good example of what can be done with this rising new technology is shown in the article: "How to: Create a Facebook App with Google App Engine and Best Buy Remix" by Omar Abdelwahed. Briefly, Best Buy's Remix API and iLike's Music API are combined to search for music products that users of this application can listen to and then purchase. A detailed tutorial how to create such application using Google App Engine and then how to uploaded in Facebook, can be found in the link provided in the reference section [11].

5   Comparison between Cloud Services

In this chapter brief comparison between Google App Engine and the other two major companies which offers similar services will be provided to the reader.

One of the biggest competitors in "The Cloud" of Google App Engine is Amazon's Web Service (AWS), so it is logical to compare it with it. But this next step in application development pushes also Microsoft with their Azure Service Platform to compete direct against Google and Amazon in the software as a service model.

According Phanfare's company blog, Google App Engine provides customers with vertically integrated development environment to create a web application in Python and it has pros and cons comparative to the Amazon Web Services approach which offers to the customer more industry standard pieces like Linux instances (EC2), key-value stores (SimpleDB) and web-service-based filesystems (S3)[16]. Amazon offer is a lower level offer, which is closer to the hardware, and it will take more time to build system with it, but the resulting system will be more extensible. In the other hand, if the customer wants to create new web application it will be faster and will bring him faster to the market[16].

Michael J. Miller, vice president at Ziff Brothers Investments compares the three major players in this field with their pros and cons. He concludes that all of them have their own strengths:

"So looking at these three players, you see each of them playing to their strengths. Amazon was early in the market, and has leveraged Internet standards and open source platforms to create a very flexible platform. Google is leveraging the work it has done with big databases and its internal development methods to create a powerful but more restrictive environment. And Microsoft is leveraging its traditional strength with developers and the breadth of its tools to create perhaps the largest array of services"[17].

Another opinion of Jessica Hodgson from Dow Jones Newswires shows that Microsoft's move is more strategically then truly to develop something new. Her opinion is:

"What I think Microsoft is trying to do is to freeze the market. They want people who may have been thinking about experimenting with on-demand products to hold off and wait to see what their products are like"[17].

6   Conclusion

It is too early to make general conclusions and statements about Google App Engine technology. This cloud computing solution is still too young, and recently appears on the market.

It looks like that the current situation with creating and developing even simple web applications is really challenging and demanding. Except writing the code there are a lot of other tasks to be done. First user need to write configuration file and startup scripts of the web server he want to use. Then he needs to set up the SQL database along with creating tables, lookups, user privileges, updating new versions and so on and so on. To find a proper physical machine, on which the product should be hosted, is not an easy thing as well. Even for the smallest applications which are used few times per week, needs to be paid significant amount of money for them, when traditional hosting services are in use. And after all when all things are set up, somebody needs to maintain the application. Sometimes machines crash, configuration files have errors, hard disks break, traffic starts to grow and etc. So to keep everything going on properly, takes money, time and efforts.

All of those problems were tried to be solved with App Engine. Several components while using App Engine makes this possible. First thing which should be take into account is the massive Google's company infrastructure. Second is the provided Python language runtime and support of more languages will appear in the near future. Third is the SDK

which completely emulates Google's server environment. This is useful because customers can create their applications locally, in real time, without uploading them. Then it comes the web based admin interface. This interface contains all the necessary tools to manage and debug user applications. And finally, the Datastore API, which is the Google's scalable persistence layer. All it needs to be done is to define URL which user wants to use, writing the code and to upload the application. Everything else will be done by Google [3].

According Kevin Gibbs which is App Engine Tech Lead, the design goals of Google are[3]:

- Make the system easy to use

- make it easy to scale

- make it free to get started

It looks so far that Google team achieved them.

References

[1] Google.com. Developer's Guide. http://code.google.com/appengine/docs/ whatis-googleappengine.html, Accessed on 9 Feb, 2009.

[2] Google.com. Google App Engine General Questions. http://code.google.com/appengine/kb/general.html, Accessed on 9 Feb, 2009.

[3] Google.com. Google App Engine Campfire One Transcript. http://code.google.com/appengine/articles/cf1-text.html, Accessed on 9 Feb, 2009.

[4] Google.com. Google App Engine Blog. http://googleappengine.blogspot.com/, Accessed on 9 Feb, 2009.

[5] Wikipedia.com. Google App Engine. http://en.wikipedia.org/wiki/Google-App-Engine, Accessed on 9 Feb, 2009.

[6] TechDirt Blog. Google Finally Realizes It Needs To Be The Web Platform. http://techdirt.com/articles/20080407/225749782.shtml, Accessed on 9 Feb, 2009.

[7] InfoWorld Blog. Cloud versus cloud: Amazon EC2, Google App Engine, GoGrid, and AppNexus. http://weblog.infoworld.com/tcdaily/archives/2008/07/video-tours-of.html?source=fssr, Accessed on 9 Feb, 2009.

[8] SearchEnterpriseDesktop.com. Definitions. http://searchenterprisedesktop.techtarget.com/sDefinition, Accessed on 9 Feb, 2009.

[9] Wikipedia.com. Cloud Computing. http://en.wikipedia.org/wiki/Cloud-computing, Accessed on 14 March, 2009.

[10] The industry standard. The Internet Cloud. http://www.thestandard.com/article/0,1902,5466,00.html, Accessed on 14 March, 2009.

[11] Google.com. Google App Engine Articles. http://code.google.com/appengine/articles/shelftalkers.html, Accessed on 9 Feb, 2009.

[12] Code.Google.com. Python Service APIs - Datastore. http://code.google.com/appengine/docs/python/datastore/. Accessed on 14 March, 2009.

[13] Code.Google.com. Python Service APIs - Memcache. http://code.google.com/appengine/docs/python/datastore/. Accessed on 14 March, 2009.

[14] Code.Google.com. Python Service APIs - Urlfetch. http://code.google.com/appengine/docs/python/datastore/. Accessed on 14 March, 2009.

[15] Code.Google.com. Python Service APIs - Mail. http://code.google.com/appengine/docs/python/datastore/. Accessed on 14 March, 2009.

[16] Blog.Phanfare.com. App Engine vs. Amazon Web Services. http://blog.phanfare.com/2008/04/google-app-engine-vs-amazon-web-services/. Accessed on 14 March, 2009.

[17] Infoq.com. Comparing Amazon's EC2, Google's App Engine and Microsoft's Azure. http://www.infoq.com/news/2008/11/Comparing-EC2-App-Engine-Azure. Accessed on 14 March, 2009.