

# Performance Analysis based on two Leading Cloud Computing Platforms: Google App Engine and Amazon Web Service

Chao He [he.chao@wustl.edu](mailto:he.chao@wustl.edu) (A paper written under the guidance of [Prof. Raj Jain](#))

[Download](#)



## Abstract

Cloud computing is recognized as a revolution in the computing area, meanwhile, it also brings the question on the necessity and applicability of this new industry standard. This paper aims at the analysis of the performance comparison of cloud computing platforms and traditional web servers. Two significant cloud computing platforms, Google App Engine and Amazon Web Service, are chosen for study. The analysis indicates that cloud computing platforms can get reasonable performance compared to traditional web servers.

**Keywords:** cloud computing, Google App Engine, Amazon Web Service, traditional web servers, round-trip time, network throughput, network bandwidth, measurement, performance analysis.

## Table of Contents

- [1 Introduction](#)
  - [1.1 Background](#)
  - [1.2 Key Characteristics](#)
- [2 Two Leading Cloud Computing Platforms](#)
  - [2.1 Google App Engine](#)
  - [2.2 Amazon Web Service](#)
  - [2.3 Comparison of Google App Engine and Amazon Web Service](#)
- [3. Performance Analysis](#)
  - [3.1 Analysis for Google App Engine](#)
    - [3.1.1 Measuring Tools and Implementations](#)
    - [3.1.2 Metrics Selection](#)
    - [3.1.3. Parameters Selection](#)
    - [3.1.4 Performance Analysis](#)
  - [3.2 Analysis for Amazon Web Service](#)
    - [3.2.1 Machine Specifications and Instanced for Amazon Web Service and Camillus](#)
    - [3.2.2 Metrics and Parameters Selection](#)
    - [3.2.3 Performance Analysis](#)
  - [3.3 Future Work](#)
- [4 Summary](#)
- [5 List of Acronyms](#)
- [6 References](#)

## 1 Introduction

Cloud computing is a general concept in which services (computing, storage, data access and etc.) do not depend on the end users' physical locations or configuration of their systems that deliver the services. It is an innovative product based on existing techniques such as grid computing, distributed computing, parallel computing, utility computing, network storage technologies, virtualization, load balance, etc. The purpose of cloud computing is to merge several low cost computing units to one higher level system with strong computing ability and deliver some specified techniques (Infrastructure as a Service (IaaS), Platform as a Service (PaaS), Location as a Service (LaaS)) to end users. The key idea of cloud computing is to form a computing pool which can distribute its resources based on the user's needs. In this chapter, the background will be introduced in section 1.1 and the key characteristics will be presented in section 1.2.

## 1.1 Background

The 'cloud' was used as a metaphor for the Internet especially for telephone networks and later used to describe the Internet in computer network diagrams and the infrastructure it conceals. Nowadays, cloud computing has grown up to a mature industry standard supported by many companies. There are many cloud computing platforms existing in use such as Google App Engine (GAE), Amazon Web Service (AWS), HP Cloud-enabled computing, IBM Cloud Computing and etc. More and more companies invest in lots of money on the research and development in cloud computing area.

## 1.2 Key Characteristics

- a. Agility: rapidly and inexpensively re-provisions resources to end users.
- b. Application Programming Interface (API): provides the same way of accessing the cloud software as well as the interactions between local machines and end users.
- c. Cost: dramatically reduces the cost, capital expenditures are converted to operational expenditures in a cloud model which means the resource consumption is based on the users' needs.
- d. Device and location independence: allows users to access the cloud systems anywhere through the Internet regardless of their locations.
- e. Multi-tenancy: resources-sharing allows centralization, increases peak-load capacity and improves utilization and efficiency.
- f. Reliability: improved if multiple redundant sites are used.
- g. Scalability: dynamically delivers the resources to end users. Users don't need to worry about the peak load in the system.
- h. Security: could be enhanced by the centralization, at least as good as the traditional working systems of the users' own.
- i. Maintenance: easy to manage and maintain since there is no software installed in the end users' computers.
- j. Metering: the resources usage should be measured per client on a day-to-day basis.

The next chapter will give a brief introduction of two leading cloud computing software platforms: Google App Engine and Amazon Web Service.

## 2 Two Leading Cloud Computing Platforms

This chapter will show the histories and features of two leading cloud computing infrastructure, the Google App Engine and Amazon Web Service, in section 2.1 and 2.2. The general differences between these two software platforms are also presented in section 2.3.

## 2.1 Google App Engine

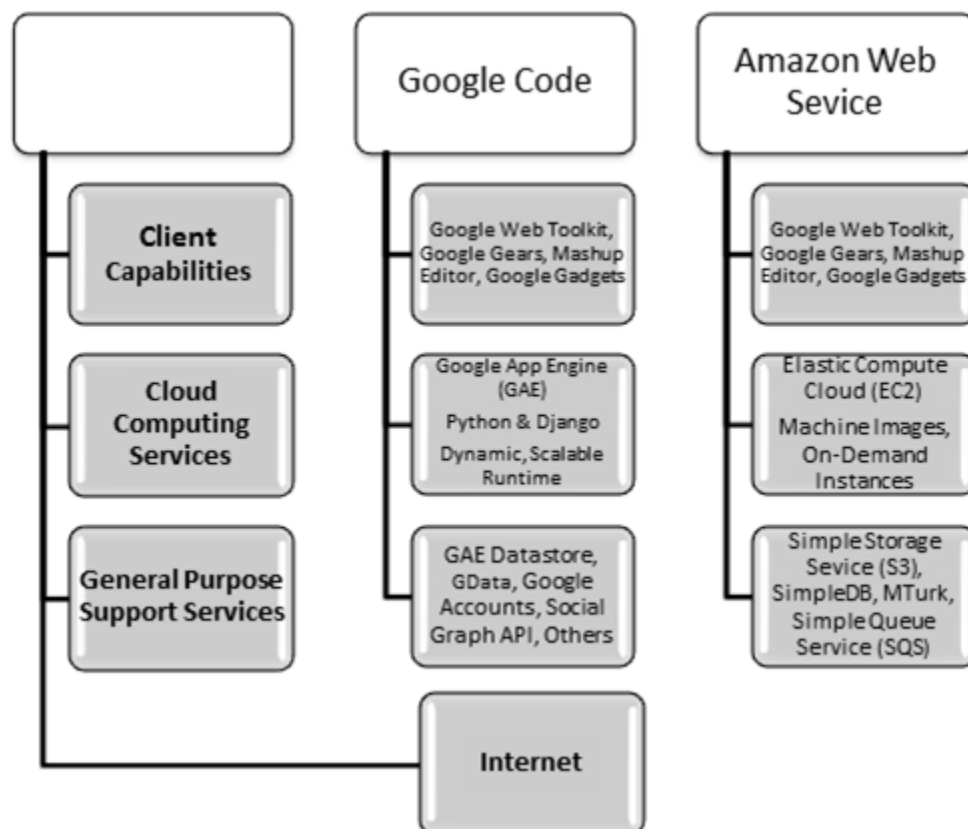
On April 2008, Google released the beta version of the Google App Engine which allows the developers to develop the applications based on Python. The developers can also use Google's infrastructures to manage their developing process (maximum 500MB storage space). For the excessive part, Google will charge 10-12 cents/GB on per CPU per hour basis. The key idea of GAE is to virtualize the apps across multiple data centers and servers.

## 2.2 Amazon Web Service

Amazon's 'cloud' was initialized in 2002 and named Amazon Web Service. It is a web based remote computing collection. It is constructed based on four key services, Simple Storage Service (S3), Elastic Compute Cloud (EC2), Simple Queuing Service and SimpleDB. In other words, Amazon now provide the storage service, computing service, queuing service and data base access service through the Internet. Other Services include Amazon Associates Web Services (A2S), Amazon AWS Authentication, Amazon Virtual Private Cloud (VPC) and etc.

## 2.3 Comparison of Google App Engine and Amazon Web Service

The comparison of Amazon Web Service and Google App Engine is shown in Figure 1. The main difference between Amazon Web Service and Google App Engine is that Amazon Web Service is IaaS while Google App Engine is PaaS. The next chapter will analyze the performance of Google App Engine and Amazon Web Service in two different aspects respectively.



**Figure 1: Comparison of Amazon Web Service and Google App Engine**

## **3 Performance Analysis based on Google App Engine and Amazon Web Service**

This chapter will begin with the analysis of Google App Engine in section 3.1, measuring the difference of performances between the Google App Engine and traditional web servers. Then the performance analysis for Amazon Web Service will be presented in section 3.2. The future work is discussed in section 3.3.

### **3.1 Analysis for Google App Engine**

The subsections below from 3.1.1 to 3.1.4 will give a brief performance analysis on the Google App Engine by comparing the Google App Engine with the traditional web server. The analysis on how to find the importances of the effects is also presented in these four subsections.

#### **3.1.1 Measuring Tools and Implementations**

The httpperf measurement tool [6] and Planetlab testbed [7] are used in this case. The httpperf which was developed by David Mosberger and other staff at Hewlett-Packard Research Laboratories is a test tool used to measure the performance of the web servers. The Planetlab testbed is a virtual lab network established on March 2002, which consists of a bunch of lab machines distributed around the globe most of which are hosted by research institutions. Same tests are running on the Google App Engine and the traditional web servers.

#### **3.1.2 Metrics Selection**

For the study of Google App Engine, a list of two most important performance metrics is given below:

- a. Round-trip time (RTT)
- b. Network throughput

Round-trip time is the time it takes from the data being processed to reach the host and returns back to the user. It is an important metric for cloud computing area since it can give a better insight into comparison of the latencies of the Google App Engine with the traditional web servers. RTT is measured in seconds.

The network throughput metric measures the data transferred through the network connection for a period of time. It also responds to the bandwidth of the system. In this case, it can show the difference in bandwidth between the Google App Engine and the traditional web servers. Network throughput is measured in kB/sec.

#### **3.1.3 Parameters Selection**

Two major parameters are selected in the experiment, the data size and the number of requests per Planetlab node.

Three data size types are chosen, small image (12kB), medium image (350kB) and large image (1MB). The number of requests per Planetlab node is selected to be 1, 10 and 100.

#### **3.1.4 Performance Analysis**

The experiment results for the RTT are shown in Table 1. The data in Table 1 are collected from *Network-based Measurements on Cloud Computing Services*. [1]

**Table 1 RTT results for GAE and traditional web host, data are collected from Network-based Measurements on Cloud Computing Services [1].**

Image Size	12kB			350kB			1MB		
# of req/Planetlab node	1	10	100	1	10	100	1	10	10
RTT for GAE(sec)	1	5	47	1	10	40	1	15	43
RTT for TWS(sec)	1	13	62	1	50	510	1	120	1380

Using the analysis methods introduced in *The Art of Computer Systems Performance Analysis* [4], this experiment design can be considered as a 2\*3\*3 design. The factors are platform, image size and number of request per Planetlab node. The factors and their levels are listed in Table 2.

**Table 2: Factors and Levels for Google App Engine Study.**

Symbol	Factor	Level 1	Level 2	Level 3
P	Platform	Google App Engine(GAE)	Traditional Web Server	
I	Image Size	12kB	350kB	1MB
N	Number of Request per Planetlab Node	1	10	100

Since the max/min in Table 1 is 1380, a log transformation should be performed for this case. The Analysis of Variance (ANOVA) is shown in Table 3.

From the results we get from Table 3, the RTT is highly affected by two factors, the platform and the number of requests per Planetlab node. Since the dominant effect, the number of requests per Planetlab node, explains 79.428% of variations while the second most significant effect, the platform, only explains 7.448% of variations, as a matter of fact, we can say that the cloud computing can get reasonable performance if the resources in the infrastructures are appropriately distributed.

**Table 3: ANOVA table of RTT for Google App Engine Study.**

Component	Sum of Squares	Percentage of Variation	Degree of Freedom	Mean Square
y	39.754		18	
y_bar	22.925		1	
y-y_bar	16.829	100	17	
Main effects	15.265	90.708	5	3.053
P	1.253	7.448	1	
I	0.645	3.831	2	
N	13.367	79.428	2	
First-order interaction	1.299	7.722	8	0.162
PI	0.305	1.814	2	
PN	0.669	3.978	2	

<b>IN</b>	<b>0.325</b>	<b>1.930</b>	<b>4</b>	
<b>Second-order interaction</b>	<b>0.264</b>	<b>1.570</b>	<b>4</b>	<b>0.066</b>
<b>PIN</b>	<b>0.264</b>	<b>1.570</b>	<b>4</b>	

The experiment results for network throughput are shown in Table 4. These results are collected from *Network-based Measurements on Cloud Computing Services*. [1]

**Table 4: Throughput results for GAE and traditional web host, data are collected from Network-based Measurements on Cloud Computing Services [1].**

<b>Image Size</b>	<b>12kB</b>			<b>350kB</b>			<b>1MB</b>		
<b># of req/Planetlab node</b>	<b>1</b>	<b>10</b>	<b>100</b>	<b>1</b>	<b>10</b>	<b>100</b>	<b>1</b>	<b>10</b>	<b>10</b>
<b>RTT for GAE(sec)</b>	<b>50</b>	<b>36</b>	<b>32</b>	<b>275</b>	<b>220</b>	<b>175</b>	<b>250</b>	<b>260</b>	<b>135</b>
<b>RTT for TWS(sec)</b>	<b>42</b>	<b>22</b>	<b>26</b>	<b>120</b>	<b>70</b>	<b>65</b>	<b>75</b>	<b>65</b>	<b>60</b>

Using the same method applied in the previous analysis, we can also get the ANOVA results in Table 5. Since the ratio of max/min is 11.9, we should try the log transformation as well.

**Table 5: ANOVA table of throughput for Google App Engine Study.**

<b>Component</b>	<b>Sum of Squares</b>	<b>Percentage of Variation</b>	<b>Degree of Freedom</b>	<b>Mean Square</b>
<b>y</b>	<b>67.581</b>		<b>18</b>	
<b>y_bar</b>	<b>65.408</b>		<b>1</b>	
<b>y-y_bar</b>	<b>2.173</b>	<b>100</b>	<b>17</b>	
<b>Main effects</b>	<b>2.008</b>	<b>92.408</b>	<b>5</b>	<b>0.402</b>
<b>P</b>	<b>0.549</b>	<b>25.283</b>	<b>1</b>	
<b>I</b>	<b>1.329</b>	<b>61.155</b>	<b>2</b>	
<b>N</b>	<b>0.130</b>	<b>5.970</b>	<b>2</b>	
<b>First-order interaction</b>	<b>0.156</b>	<b>7.197</b>	<b>8</b>	<b>0.020</b>
<b>PI</b>	<b>0.115</b>	<b>5.278</b>	<b>2</b>	
<b>PN</b>	<b>0.018</b>	<b>0.837</b>	<b>2</b>	
<b>IN</b>	<b>0.023</b>	<b>1.081</b>	<b>4</b>	
<b>Second-order interaction</b>	<b>0.009</b>	<b>0.395</b>	<b>4</b>	<b>0.002</b>
<b>PIN</b>	<b>0.009</b>	<b>0.395</b>	<b>4</b>	

Notice that the major effects for the network throughput study are the platform (explains 25.283% of the variations) and image size (explains 61.155% of the variations). This will lead to the conclusion that the throughput for Google App Engine and traditional web servers strongly depend upon the data size transmitted through the network. Compared to the effect of image size, the effect of the platform can be neglected. Therefore, it is possible that Google App Engine can get reasonable performance by wisely choosing the data size.

## 3.2 Analysis for Amazon Web Service

The following subsections from 3.2.1 to 3.2.3 will introduce the analysis based on Amazon Web Service. The

Amazon Web Service and Camillus are used for comparison. The study on one thread bandwidth is presented in these subsections.

### 3.2.1 Machine Specifications and Instances for Amazon Web Service and Camillus

The Machine specifications and instances for Amazon Web Service and Camillus are given in the Table 6. The data are collected from *A Quantitative Analysis of High Performance Computing with Amazon's EC2 Infrastructure: The Death of the Local Cluster?* [2]. Notice that Amazon Web Service has five instance types, each one requires different CPU speed, memory storage, disk storage and I/O performance. One ECU (EC2 Compute Unit) is equivalent to 1.0 – 1.2 GHz 2007 Opteron or Xeon processor performance. The Camillus is used for performance comparison.

**Table 6: Table of machine specifications and instances for Amazon Web Service and Camillus, data are collected from A Quantitative Analysis of High Performance Computing with Amazon's EC2 Infrastructure: The Death of the Local Cluster? [2].**

Specification and Instance Type	CPU	Memory(GB)	Disk(GB)	I/O Performance
M1.Small 32bit, 1 core	1*ECU	1.7	160	Moderate
M1.Large 64bit, 2 cores	2*ECU	7.5	850	High
M1.XLarge 64bit, 4 cores	2*ECU	15	1690	High
C1.Medium 32bit, 2 cores	2.5*ECU	1.7	350	High
C1.Xlarge 364bit, 8 cores	2.5*ECU	7	1690	High
Camillus	64-bit dual-CPU Intel Xeon E5345 Quad-Core	16	--	High

### 3.2.2 Metrics and Parameters Selection

The metric for studying Amazon Web Service is chosen to be the memory bandwidth since nowadays applications consume a lot of memory bandwidth based on the data stored in the memory.

The parameter is chosen to be the actions of the application which include four levels: copy, scale, add and triad.

### 3.2.3 Performance Analysis

The experiment result is shown in Table 7. All the experiment data are collected from *A Quantitative Analysis of High Performance Computing with Amazon's EC2 Infrastructure: The Death of the Local Cluster?* [2].

**Table 7: Experiment results for Amazon Web Service study, data are collected from A Quantitative Analysis of High Performance Computing with Amazon's EC2 Infrastructure: The Death of the Local Cluster? [2].**

	1 Thread Bandwidth in GB/s			
Machine Specifications	Copy	Scale	Add	Triad
M1.Large	2.058	1.777	1.868	1.725

<b>M1.XLarge</b>	<b>2.551</b>	<b>2.394</b>	<b>2.434</b>	<b>2.178</b>
<b>C1.Medium</b>	<b>2.865</b>	<b>2.852</b>	<b>3.114</b>	<b>3.097</b>
<b>C1.Xlarge</b>	<b>2.849</b>	<b>2.840</b>	<b>3.126</b>	<b>3.120</b>
<b>Camillus</b>	<b>2.834</b>	<b>2.830</b>	<b>3.171</b>	<b>3.160</b>

Notice that the observations in Table 7 can be considered as the paired observations for each level of the factor. Then using the method introduced in *The Art of Computer Systems Performance Analysis* [4], we can compare the performance differences between each Amazon Web Service instance and the Camillus. The analysis results are listed in Table 8.

**Table 8: Performance differences analysis for Amazon Web Service study.**

	<b>Sample Mean</b>	<b>Sample Variance</b>	<b>Sample Standard Deviation</b>	<b>90% Confidence Interval for Mean</b>
<b>M1.Large &amp; Camillus</b>	<b>0.285</b>	<b>1.062</b>	<b>1.031</b>	<b>(-1.115,1.686)</b>
<b>M1.XLarge &amp; Camillus</b>	<b>0.152</b>	<b>0.376</b>	<b>0.613</b>	<b>(-0.681,0.985)</b>
<b>C1.Medium &amp; Camillus</b>	<b>0.004</b>	<b>0.003</b>	<b>0.052</b>	<b>(-0.067,0.075)</b>
<b>C1.Xlarge &amp; Camillus</b>	<b>0.003</b>	<b>0.001</b>	<b>0.034</b>	<b>(-0.043,0.051)</b>

From the results we get from Table 8, we can say that the performances are not different for each of these four observation pairs since the 90% confidence intervals all include zero. In other words, the performances on the 1 thread bandwidth study for the Amazon Web Service and the traditional machine within 90% confidence interval are the same. If we properly construct the architecture of the cloud computing infrastructure, we can get reasonable performance compared with the traditional machines in the memory bandwidth aspect.

### 3.3 Future Work

The experiment designs introduced in this chapter only include a small amount of metrics that should be measured for the cloud computing infrastructure. Other important metrics such as the data loss rate should be studied in future tests.

In addition to the conclusion in this chapter, we also need to analyze the reasons why these factors will affect performances. It will give us a better insight into the cloud computing.

## 4 Summary

Cloud computing, as one of the most innovative techniques today, redefines the way of communication. It allows the end users to store and load their data or do the complex computing tasks anytime anywhere with a single device which can access the Internet no matter if it is a cellphone or a laptop. We don't need to worry about the data loss in our own laptops because all the data are stored in the remote drivers instead of the local drivers.

In this paper, performance analysis on two popular cloud computing platforms by comparing them to



traditional web servers is introduced. The analysis methods in *The Art of Computer Systems Performance Analysis* [4] are conducted. As we can see from the analysis above, for the study on the RTT of Google App Engine, the dominant effect is the number of request per Planetlab node which explains 79.428% of the variation while the second main effect is the platforms which only explains 7.448% of the variation. In the second experiment on Google App Engine, the network throughput is affected by the image size (explains 61.155% of variations) and the platform (explains 25.283% of variations). This tells us that the platform is not a significant bottleneck in the RTT and throughput aspects of cloud computing. For the test on Amazon Web Service, by comparing one tread bandwidth of Amazon Web Service and Camillus, the results indicate that neither of these two planforms is superior than the other. In a conclusion, the cloud computing infrastructure can get quite reasonable performance compared to the traditional web servers depending upon the service delivered. This may provide a better insight on how to construct the cloud computing infrastructure and platforms.

## 5 List of Acronyms

A2S	Amazon Associates Web Services
ANOVA	Analysis Of Variance
API	Application Programming Interface
AWS	Amazon Web Service
EC2	Elastic Compute Cloud
GAE	Google App Engine
IaaS	Infrastructure as a Service
LaaS	Location as a Service
PaaS	Platform as a Service
RTT	Round-Trip Time
S3	Simple Storage Service
TWS	Traditional Web Server
VPC	Amazon Virtual Private Cloud

## 6 References

1. Vinod Venkataraman, Ankit Shah, Yin Zhang, "Network-based Measurements on Cloud Computing Services", <http://www.cs.utexas.edu/~vinodv/files/cc-measure.pdf>.
2. Zach Hill, Marty Humphrey, "A Quantitative Analysis of High Performance Computing with Amazon's EC2 Infrastructure: The Death of the Local Cluster?", the 10th IEEE/ ACM International Conference on Grid Computing (Grid 2009). Oct 13-15 2009.
3. Dominique A. Heger, "A Performance Centric Introduction to Cloud Computing", <http://www.dhtusa.com/media/CldPerf.pdf>.
4. Raj Jain, "The Art of Computer Systems Performance Analysis: Techniques for Experimental Design, Measurement, Simulation, and Modeling", Wiley, 1991, pp 203-220, pp 381-389.
5. "Cloud computing – Wikipedia, the free encyclopedia", [http://en.wikipedia.org/wiki/Cloud\\_computing](http://en.wikipedia.org/wiki/Cloud_computing).
6. "httpperf – Wikipedia, the free encyclopedia", <http://en.wikipedia.org/wiki/Httpperf>.
7. "Planetlab testbed", <http://www.planet-lab.org>.
8. Kaiqi Xiong, Harry Perro, "Service Performance and Analysis in Cloud Computing", <http://www4.ncsu.edu/~hp/Kaiqi10.pdf>.
9. Daniel A. Menasce, Paul Ngo, "Understanding Cloud Computing: Experimentation and Capacity Planning", Proc. 2009 Computer Measurement Group Conf, Dec.7-11, 2009.

10. Thomas B Winans and John Seely Brown, "Cloud Computing, a collection of working paper", 2009, <http://www.johnseelybrown.com/cloudcomputingpapers.pdf>.
- 

Last modified on April 24, 2011

This and other papers on latest advances in performance analysis are available on line at

<http://www1.cse.wustl.edu/~jain/cse567-11/index.html>

[Back to Raj Jain's Home Page](#)