Techniques for the comparison of public cloud providers

Frans Simanjuntak, Marco Gunnink

Abstract—Cloud computing is the delivery of computing services over the Internet. It enables users to deploy applications on hardware provided by external providers, rather than building and maintaining their own servers. Cloud computing rose to popularity from 2006 with the introduction of Amazon Elastic Compute Cloud, Microsoft's Azure and the Google App Engine.

Cloud computing offers many benefits, such as cost, scalability, rapid deployment and accessibility. While most public cloud providers offer similar functionality, there are still many differences between the features they provide, their performance and cost. To help users select their ideal cloud provider, several techniques for the comparison of public cloud providers have been researched and developed. In this paper we compare three of these techniques: Analytic Hierarchy Process (AHP), CloudCmp and Technique for Order Preference by Similarity to Ideal Solution (TOPSIS).

Index Terms—Cloud computing, techniques, comparison, performance, cost, AHP, TOPSIS, CloudCmp.

1 Introduction

Over the past decade, cloud computing has become very popular [5]. And as the demand for cloud computing rises, so does the supply of public cloud providers. The main advantages of cloud computing are cost efficiency and usability [4]. Cloud providers operate on a pay-as-you-go model: users only pay for the resources they used. Users of public clouds can avoid large investments in hardware, upgrades and maintenance. This is especially beneficial for systems that only need to be used for a short time, such as those for research projects.

There are other benefits as well: cloud computing systems can be scaled up or down according to immediate demand. They provide virtually unlimited resources which can be used and paid for as needed. This also offers rapid deployment. New applications can be deployed quickly and upgraded as needed. The user can easily adapt the virtual hardware to the application's requirements when they change.

Cloud computing also offers benefits to the cloud providers. They can use their existing computing infrastructure more efficiently while gaining extra sources of revenue. Finally, use of cloud computing in large data centers helps reduce energy costs and pollution [1].

There are, however, also downsides to cloud computing. Access to the cloud computing systems depends on Internet connectivity: if there is an Internet outage at either the user or the cloud provider, the service cannot function. The user also loses some control and flexibility by using externally managed hardware. The available functionality depends almost entirely on what the cloud provider can offer and if it becomes inadequate the user may have to migrate their system to another provider. Finally, it may be difficult to ensure sufficient security for the user's system because it is always connected to the Internet and they often share the hardware with other users [10].

As we mentioned, there are many companies offering public cloud computing services. Choosing the most suitable one for a business' need is not easy. Different cloud providers offer different trade-offs in performance, cost and functionality. Comparing them is difficult, since a lot depends on what the business needs while all cloud providers attempt to make themselves appear attractive to all potential customers. Despite this, several techniques for comparing public cloud providers have been researched and developed, such as: Analytic Hierarchy Process (AHP), Technique for Order Preference by Similarity to Ideal Solution (TOPSIS), ELECTRE, CSP (Cloud Service Provider), CloudCmp framework, Fuzzy Inference System (FIS), Multi Attribute Group Decision Making (MAGDM). In this paper we compare AHP, CloudCmp and TOPSIS to determine the most suitable comparison technique.

• *Frans Simanjuntak, s3038971, E-mail: f.simanjuntak@student.rug.nl.*
• *Marco Gunnink, s2170248, E-mail: m.gunnink@student.rug.nl.*

We chose AHP and TOPSIS since they offer powerful techniques for decision making, which have already been proven and applied in other fields. CloudCmp, on the other hand is very new, but specific to the comparison of cloud providers. We are interested to see how these techniques can be combined for the selection of public cloud providers.

2 Description of Methods

In this section we will give the details of the three techniques used for the comparison of public cloud providers.

2.1 Analytic Hierarchy Process (AHP)

This section explains Analytic Hierarchy Process (AHP) method for decision making.

2.1.1 Basic Theory of AHP

Analytic Hierarchy Process (AHP) is an effective method for dealing with complex decision making. It is based on the experience gained by its developer, T.L. Saaty, while directing research projects in the US Arms Control and Disarmament Agency [2]. The theory behind this method is to derive the ratio scales from paired comparisons. The actual measurement such as price, weight, etc., or the subjective opinion such as satisfaction feeling and preference will be used as an input.

AHP generates a weight for each evaluation criterion according to the decision makers pairwise comparisons of the criteria. The higher the weight, the more important the corresponding criterion. Next, for a fixed criterion, AHP assigns a score to each option according to the decision makers pairwise comparisons of the options based on that criterion. The higher the score, the better the performance of the option with respect to the considered criterion. Finally, AHP combines the criteria weights and the option scores, thus determining a global score for each option, and a consequent ranking. The global score for a given option is a weighted sum of the scores it obtained with respect to all the criteria [11].

Even though AHP is known as a very flexible and powerful tool, it may require a large number of evaluations by the user, especially for problems with many criteria and options. In order to reduce the decision maker's workload, AHP can be completely or partially automated by specifying suitable thresholds for automatically deciding some pairwise comparisons. In order to obtain the final results, AHP performs sequential steps as follows [2]:

- *Step 1*: The problem is decomposed into a hierarchy of goal, criteria, sub-criteria and alternatives.
- *Step 2*: Data is collected from experts or decision-makers corresponding to the hierarchic structure, in the pairwise comparison of alternatives on a qualitative scale. Experts can rate the comparison as equal, marginally strong, strong, very strong, and

extremely strong. The opinion can be collected in a specially designed format as shown in Figure 1.

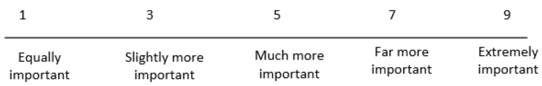

Fig. 1. Pairwise comparison [11]

- *Step 3*: The pairwise comparisons of various criteria generated at step 2 are organized into a square matrix. The diagonal elements of the matrix are 1. The criterion in the ith row is better than criterion in the jth column if the value of element (i, j) is greater than 1; otherwise the criterion in the jth column is better than that in the ith row.The (j,i) element of the matrix is the reciprocal of the (i, j) element.

$$A_n = (a_{ij})_{nxn} \tag{1}$$

- *Step 4*: The principal eigenvalue and the corresponding normalized right eigenvector of the comparison matrix give the relative importance of the various criteria being compared. The elements of the normalized eigenvector are termed weights with respect to the criteria or sub-criteria and ratings with respect to the alternatives.

- *Step 5*: The consistency of the matrix of order n is evaluated. Comparisons made by this method are subjective and AHP tolerates inconsistency through the amount of redundancy in the approach. If this consistency index fails to reach a required level then answers to comparisons may be re-examined. The consistency index, CI, is calculated as

$$CI = \frac{(\lambda_{max} - n)}{(n-1)} \tag{2}$$

where λ_{max} is the maximum eigenvalue of the judgement matrix. This CI can be compared with that of a random matrix, RI.

$$RI = \frac{(\lambda_{max}{}' - n)}{(n-1)} \tag{3}$$

The ratio derived is termed the consistency ratio, CR. Saaty suggests the value of CR should be less than 0.1 (see Table 1).

$$CR = \frac{CI}{RI} \tag{4}$$

- *Step 6*: The rating of each alternative is multiplied by the weights of the sub-criteria and aggregated to get local ratings with respect to each criterion. The local ratings are then multiplied by the weights of the criteria and aggregated to get global ratings.

Table 1. Random Consistency Index [11]

n	1	2	3	4	5	6	7	8	9	10
RI	0	0	0.58	0.90	1.12	1.24	1.32	1.41	1.45	1.49

Finally, AHP produces weight values for each alternative based on the judged importance of one alternative over another with respect to a common criterion.

2.1.2 AHP and Cloud Computing

Cloud computing is a paradigm for providing all sorts of cloud services, that can be rapidly deployed and delivered with minimal management cost and effort or reciprocity among service providers [8]. This paradigm has boosted many providers to release new cloud services in order to attract potential costumers. Unwittingly, the increasing growth of services among providers may be identical or even similar with different characteristic. This is, of course, difficult for the potential customers to judge and weight the one that balances their requirements best. As multiple criteria are involved during the selection of cloud services, selecting the best one can be difficult due to multiple consumer requirements and this process could be very time-consuming and error-prone.

Being the most widely used technique for complex decision making, AHP is one of the alternatives to overcome this problem. As mentioned earlier that AHP is intended for multi criteria decision making, therefore it is suitable to be applied. Since AHP computes the results based on criteria, first of all, we have to define the criteria for cloud computing.

There are two ways of defining criteria, one can be obtained from the user perspective and the other from the element of services offered by providers. If we decide to set the criteria based on the the user perspective, we must consider the things that really matter to them, e.g response time, throughput, availability, reliability, and cost. On the other hand, if we concern about the elements of cloud services, we can define the criteria based on the attributes they have in common, e.g price/hour, virtual core, memory, CPU performance, Disk I/O consistency, Disk Performance, and Memory Performance. After defining the criteria, the rest of AHP steps can be performed.

2.2 CloudCmp Framework

CloudCmp is a comparison framework focused primarily on providing a broad comparison tool that can be applied to many cloud providers while still providing useful data to cloud users [6]. To this end the developers of CloudCmp have identified a set of common services offered by cloud providers and used by cloud users. These services are: elastic compute cluster, persistent storage, intra-cloud network and wide-area network. For each service they implemented a set of performance metrics and tested them on four public cloud providers: Amazon AWS, Microsoft Azure, Google App Engine and Rackspace CloudServers. The next sections describe the performance metrics per cloud computing service.

2.2.1 Elastic compute cluster

An elastic compute cluster hosts and runs application code. The number of virtual instances can be scaled up or down, either by a request of the application or automatically by the cloud provider in response to the processor usage. For the elastic compute cluster, CloudCmp measures the following performance metrics:

- **Benchmark finishing time** To measure the CPU, memory and local disk I/O performance, CloudCmp uses a modified SPECjvm2008 [13] benchmark. Java was chosen because it is available on all measured providers. The benchmark is run on each cloud platform and the time to finish is used to compare the providers. For those that offer multi-threading capabilities a parallel version of the benchmark is also run.

- **Cost** CloudCmp uses the published prices or billing APIs to obtain the cost for running the benchmark.

- **Scaling latency** The scaling latency is the time it takes to request and launch a new virtual instance. CloudCmp divides this into two segments. The first segment is the time from when the new instance is requested and it is created and powered on. The second is the time between powering on and the instance being ready to use.

2.2.2 Persistent storage

Three types of persistent storage are offered by the cloud providers that CloudCmp looks at: table, blob and queue, though not all providers provide all three types. Tables provide storage for structural data, similar to the traditional relational database systems, but often with only limited support for complex queries, such as joins and grouping. Blob storage is designed for large unstructured chunks of data, like binary objects. Finally queues provide a way to send small amounts of data between different instances. CloudCmp uses the following metrics to compare storage performance:

- **Operation response time** The operation response time is measured from the time an operation is requested until the last byte reaches the client.

- **Time to consistency** The cloud providers measured by CloudCmp offer automatic replication for fault tolerance and high availability. However, this comes at the cost of consistency: a read operation that immediately follows a write may still return old or stale data. CloudCmp measures the time to consistency by continuously reading from the persistent storage until the latest data is returned.

- **Cost per operation** The cost per operation is also measured for persistent storage, using the published prices or the available billing APIs.

2.2.3 Intra-cloud & wide-area network

Since cloud computing services are always provided over the Internet, network connectivity and performance are of major importance. CloudCmp distinguishes between the intra-cloud and wide-area networks. The former connects instances within the same cloud to each other while the latter connects the cloud instances to the rest of the Internet. The following performance metrics are measured by CloudCmp for the intra-cloud and wide-area networks:

- **Throughput** CloudCmp uses TCP throughput to measure the path capacity: the maximum bandwidth between two endpoints. The throughput is measured both with the default TCP window size and a maximum TCP window size of 16MB to minimize the overhead.

- **Latency** The network latency is measured using the round-trip time between the instances within the cloud and to various locations on the wide-area network. CloudCmp uses the PlanetLab network [15] to measure latency from different physical locations.

2.3 Technique for Order Preference by Similarity to Ideal Solution (TOPSIS)

2.3.1 Basic Theory of TOPSIS

TOPSIS is a multi attribute decision making technique that sorts the alternatives for a decision problem which was introduced by Hwang and Yoon in 1981. The basic principle is that the selected alternative should have the shortest distance from the ideal solution and the farthest distance from the negative ideal solution in geometrical sense [12]. The TOPSIS method consists of the following steps [14]:

- *Step 1*: Determine the decision problem and identify the relevant evaluation criteria.

- *Step 2*: Develop the preference for the criteria by assigning them weight.

- *Step 3*: Construct the Decision Matrix for the alternatives based on the criteria.

$$DM = \begin{array}{c} \\ L_1 \\ L_2 \\ . \\ . \\ . \\ L_n \end{array} \begin{array}{c} \begin{array}{cccc} C_1 & C_2 & \ldots & C_n \end{array} \\ \begin{bmatrix} X_{11} & X_{12} & \ldots & X_{1n} \\ X_{21} & X_{22} & \ldots & X_{2n} \\ . & . & . & . \\ . & . & . & . \\ . & . & . & . \\ X_{m1} & X_{m2} & \ldots & X_{mn} \end{bmatrix} \end{array} \quad (5)$$

Where i is the criterion index ($i = 1..m$); m is the number of potential sites and j is the alternative index ($j = 1..n$). The elements $C_1, C_2..C_n$ refer to the criteria: while $L_1, L_2..L_n$ refer to the alternative locations. The elements of the matrix are related to the values of criteria i with respect to alternative j.

- *Step 4*: Calculate a normalized decision matrix for the above Decision Matrix.

$$NDM = R_{ij} = \frac{X_{ij}}{\sqrt{\sum_{i=1}^{m} X_{ij}^2}} \quad (6)$$

- *Step 5*: Compute the weighted normalized decision matrix (NDM) by multiplying weights of criteria with the corresponding alternatives value.

$$V = V_{ij} = W_j \times R_{ij} \quad (7)$$

- *Step 6*: Identify the Positive Ideal Solution

$$PIS = A^+ = \{V_1^+, ..., V_n^+\} = \{(max_j(V_{ij})|i \in I'), (min_j(V_{ij})|i \in I'')\} \quad (8)$$

where I′is associated with benefit criteria, and I″is associated with cost criteria.

- *Step 7*: Identify the Negative Ideal Solution

$$NIS = A^- = \{V_1^-, ..., V_n^-\} = \{(min_j(V_{ij})|i \in I'), (max_j(V_{ij})|i \in I'')\} \quad (9)$$

where I′is associated with benefit criteria, and I″is associated with cost criteria.

- *Step 8*: Compute separation of each criteria value for each alternative from both ideal and negative ideal solution.

$$S^+ = \sqrt{\sum_{j=1}^{n} (V_j^+ - V_{ij})^2} \quad i = 1, ..., m \quad (10)$$

$$S^- = \sqrt{\sum_{j=1}^{n} (V_j^- - V_{ij})^2} \quad i = 1, ..., m \quad (11)$$

- *Step 9*: Measure the relative closeness of each location to the ideal solution.

$$C_i = S_i^- / (S_i^+ + S_i^-), \quad 0 \le C_i \le 1 \quad (12)$$

- *Step 10*: Rank the preference order. According to the value of C_i the higher the value of the relative closeness, the higher the ranking order and hence the better the performance of the alternative. Ranking of the preference in descending order thus, allows relatively better performances to be compared.

2.3.2 TOPSIS and Cloud Computing

Since cloud service selection is a multiple criteria group decision-making (MCDM) problem, TOPSIS can be an alternative method that helps user to choose an ideal cloud provider. It can assist service consumers by analyzing available services using fuzzy opinions. Fuzzy TOPSIS is a method that can help in objective and systematic evaluation of alternatives on multiple criteria.

Similar to AHP, the first step to do is to define the criteria as input parameters. There are two ways of defining criteria either from the user perspective or from the attributes of public cloud providers as mentioned in section 2.1.2. After the criteria is defined, the rest of the TOPSIS steps can be performed.

3 Results

This sections lists the result of the comparison of public cloud providers using different techniques, namely AHP, CloudCmp Framework, and TOPSIS.

3.1 AHP

For the purpose of the analysis with AHP model, we studied and followed the experiment conducted by Mingrui.Sun et.al from the Harbin Institute of Technology, China [7]. The goal of the experiment was to select a cloud service for health medical rehabilitation system. In this experiment, the rehabilitation therapy system was deployed on three different cloud platforms. The users which were stroke patients can access these systems according to their personal preference and the demand of treatment. The experiment was expected to capture the paired comparison of criterion layer with respect to goal, the paired comparison of alternative layer with respect to criterion layer, and finally the comparison results of the alternatives.

In order to perform the AHP steps, first we have to compose the cloud computing selection problem into a hierarchy of goal, criteria and alternatives which subsequently used as the basis of performing the remaining steps of AHP. In this case, the hierarchy goal was to select a cloud service, the main criteria is the quality attributes which are assessed based on the user perspective such as the response time, throughput, availability, reliability, and cost and the alternative layers are the available cloud service providers[7].

After defining the goal, the main criteria, and the alternatives, then the pairwise comparison can be constructed followed by the organization of results into a square matrix. By using this square matrix, the rest of the steps can be performed. The results of the paired comparison matrix of criterion layer with respect to the goal is listed in Table 2 and the same pattern applies to criterion layer. The result of the paired comparison matrix of alternative layers with respect to each criterion layer is listed in Table 3. Finally, the weight of each cloud provider can be obtained by summing all weights of each alternative divided by the total number of criteria as listed in Table 4.

Table 2. Paired comparison matrix criterion layer with respect to the goal [7]

	CR=0.0944$<$0.1;		Global Weight=1.0000;		λ_{max}=5.4227	
Select Service Cloud	Response Time	Throughput	Availability	Reliability	Cost	w_i
Response Time	1	1	0.1667	0.125	0.3333	0.045
Throughput	1	1	0.1429	0.11111	0.1667	0.0371
Availability	1	7	1	1	7	0.03791
Reliability	1	9	1	1	9	0.444
Cost	1	3	0.1429	1	1	0.0947

From the results listed in Table 4, it can be seen that the value of CS3 is greater than CS1 and the value of CS1 is greater than CS2, therefore the cloud service 3 would be selected as an ideal cloud provider since the weight is higher than the others. The higher the weight, the more qualified the cloud service.

3.2 CloudCmp

Li *et al.* ran CloudCmp on AWS, Azure, App Engine and CloudServers during March, April and May of 2010. Rather than replicating all the results here, we give a short overview of the tests they could run for which providers. We then show the case studies they ran and the results obtained.

Table 3. Paired comparison matrix alternative layer with respect to the criterion layer [7]

CR=0.0944$<$0.1;		**Global Weight=0.045;**		**λ_{max}=5.4227**
Response Time	CS1	CS2	CS3	w_i
CS1	1	5	3	0.6483
CS2	0.2	1	0.5	0.122
CS3	0.3333	2	1	0.2297
CR=0.0904$<$0.1;		**Global Weight=0.0371;**		**λ_{max}=3.0940**
Throughput	CS1	CS2	CS3	w_i
CS1	1	0.2	4	0.1991
CS2	5	1	8	0.7334
CS3	0.25	0.125	1	0.0675
CR=0.0053$<$0.1;		**Global Weight=0.3791;**		**λ_{max}=3.0055**
Availability	CS1	CS2	CS3	w_i
CS1	1	5	4	0.6908
CS2	0.2	1	1	0.1488
CS3	0.25	1	1	0.1603
CR=0.0707$<$0.1;		**Global Weight=0.4440;**		**λ_{max}=3.0735**
Reliability	CS1	CS2	CS3	w_i
CS1	1	3	0.1667	0.1667
CS2	0.3333	1	0.125	0.0726
CS3	6	8	1	0.7612
CR=0.0000$<$0.1;		**Global Weight=0.0947;**		**λ_{max}=3.0000**
Cost	CS1	CS2	CS3	w_i
CS1	1	3	1	0.4286
CS2	0.3333	1	0.3333	0.1429
CS3	1	3	1	0.4286

Table 4. Cloud Service Evaluation using AHP [7]

Alternatives	Weight
CS1	0.4129
CS2	0.1349
CS3	0.4522

Due to legal concerns Li *et al.* anonymized the identities of the providers and indicate them as C_1 through C_4. Additionally, for the providers that offer them, they tested the framework on different tiers, indicated by $C_j.i$ for tier i of provider j, where lower numbers of i indicate slower and cheaper instances and higher values indicate faster and more expensive tiers. Provider C_3 only offers a single performance tier.

On the elastic compute cluster they tested both single- and multi-threaded programs, except on provider C_3, which does not support multi-threading. C_3 also does not support local disk I/O, so that was not tested on C_3 either. They measured the scaling latency for providers C_1, C_2 and C_4 as C_3 does not offer manual instance requests.

For the persistent storage the table services of providers C_1, C_3 and C_4 were compared by measuring the performance of `get`, `put` and `query` operations. Li *et al.* also measured the time to consistency, but found that only C_1 exhibited any inconsistency. They furthermore tested the blob storage on all providers, except C_3. And finally they tested the queue service on providers C_2 and C_4.

Finally, they measured the latencies of the intra- and inter-datacenter networks and the wide-area network. Since C_3 does not allow direct communication between instances, intra- and inter-datacenter latency was not measured for it. On the other hand, for the wide-area network, C_3 has the best performance, due to the fact that it offers automatic load-balancing. C_2 has the worst round-trip time, because they have fewer data-centers and all located relatively close to each other.

3.2.1 Case Studies

In addition to separate specific tests for the above-mentioned performance metrics, CloudCmp also measures the cloud performance with more integrated case studies. These case studies are intended to show the cloud performance in a more realistic scenario. The three case studies are an E-commerce website, a Parallel Scientific Computation application and a Latency Sensitive website.

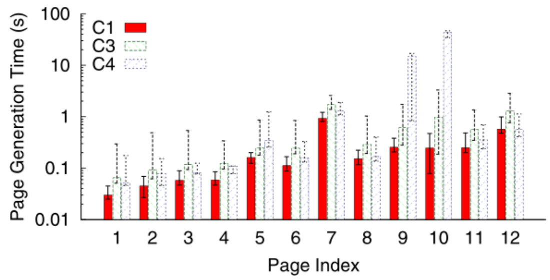

Figure 13: The page generation time of TPC-W when deployed on all three cloud providers that support it. The y-axis is in a logarithm scale.

Fig. 2. Page generation of TPC-W [6]

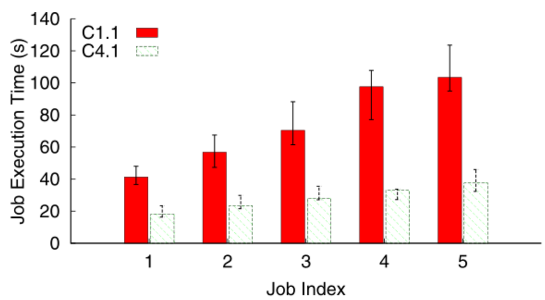

Figure 14: The job execution time of Blast when deployed on both C_1.1 and C_4.1. We show the results for five example jobs.

Fig. 3. Job execution time on Blast [6]

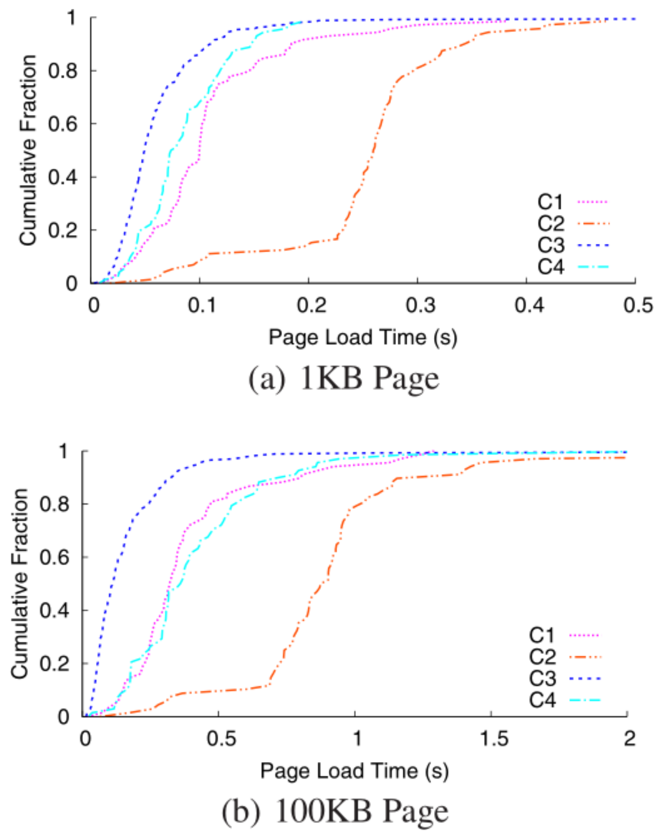

Fig. 4. Distribution of page download time [6]

The E-commerce website is tested with TPC-W [3], a standard benchmark for transactional web services. The E-commerce website mostly measures persistent storage speed as the goal is to minimize page generation time.

TPC-W was not deployed on C_3 because it does not offer a table service. Figure 2 shows the page generation time for the various pages it implements. Provider C_1 has the lowest generation time on all pages, consistent with the measurements from the table service. C_4 is much slower on pages 9 and 10 which contain many `query` operations. This too is consistent with earlier measurements, where C_4 is also shown to be slower on `querys`.

For the Parallel Scientific Computation, CloudCmp uses Blast [9], an application for DNA alignment. Blast is very CPU-intensive and therefore mainly assesses the performance of the Elastic Compute Cluster. It has one instance to accept jobs and uses a message queue to send the jobs to worker instances.

Since Blast requires a queue service, it was only tested on providers C_1.1 and C_4.1. The execution times are much lower on C_4.1, indicating it has much better computing performance than C_1.1. This was also the conclusion from the CloudCmp elastic compute cluster measurements.

Finally, the Latency Sensitive website is implemented with a simple web server that serves static pages. For CloudCmp, Li *et al.* downloaded two pages: a small one of 1KB and a large one of 100KB. The pages are downloaded from various locations, using PlanetLab. This case study evaluates the wide-area network latency.

Figure 4 shows the distribution of the page downloading times for all four providers. C_3 stands out as the fastest one, for both the small and large page. This comes as no surprise, since C_3 was also shown to have the lowest wide-area latency.

3.3 TOPSIS

In order to prove the capability of TOPSIS model as one of the alternative methods for selecting cloud providers, we reused the case study of AHP. Firstly we studied the experiment conducted by Mingrui.Sun et.al [7]. We notice that we can reuse some elements from the case study such as the criterion, the alternatives and also the weights which already listed in Table 3.

Reusing the AHP weights is indeed allowed because both AHP and TOPSIS have similar preliminary steps. However, the other input parameter such as the criterion parametric dataset cannot be obtained directly from the paired comparison matrix of alternative layer because TOPSIS works in different ways after the first step done. So, we analyzed the matrix of alternative layer and we made some adjustments towards its value. After that, we created a dataset 5 and constructed a decision matrix. This decision matrix was used as a base for executing the rest of the steps.

Table 5. Criterion Parametric Dataset

Criteria	**Alternatives**		
	CS1	CS2	CS3
Response Time	8	5	7
Throughput	7	9	5
Availability	8	5	6
Reliability	6	4	9
Cost	6	7	6

After applying the TOPSIS method using the above dataset combined with the criterion weight taken from AHP, it turns out the relative closeness value of CS3 is greater than the others, 0,957844269 to be exact (See Table6). This made the CS3 as the most preferable cloud provider followed by CS1 and CS2 respectively. Furthermore, the output of this experiment indicates that both TOPSIS and AHP produced similar results.

4 Discussion and Conclusion

In this paper, we have explained three techniques for the comparison of services offered by cloud providers: AHP, CloudCmp, and TOP-

Table 6. Cloud Service Evaluation using TOPSIS

Alternatives	Relative Closeness
CS1	0.525435122
CS2	0.070962879
CS3	0.957844269

SIS. We evaluated those techniques using case studies obtained from the literatures. Since AHP and TOPSIS are intended for multi-criteria decision analysis, we used similar cases when assessing them. However, it's not possible for us to use similar cases when it comes to evaluating CloudCmp because the evaluation properties are different compared to AHP and TOPSIS.

The results obtained from the studies show that both AHP and TOPSIS generated similar results. This phenomenon, of course, is easily explained since these techniques are meant to solve complex decision making. AHP uses a hierarchical structure and systematization in order to solve the cloud services selection problem and TOPSIS performs analysis of available cloud services using fuzzy opinions.

Although AHP is an effective tool for decision making, it has disadvantages, such as the decision maker's subjectivity, which can yield uncertainties when determining pairwise comparisons.

Then, we looked at CloudCmp. As mentioned, CloudCmp works significantly differently from the other two techniques. It sacrifices depth for broadness by only comparing common functionality, offered by all cloud providers. Despite their effort to be as broad as possible, it turned out that many features measured by CloudCmp were not available on all four providers that they tested. This means the results are not as comprehensive as they aimed to be and it remains difficult to give a complete comparison of all providers.

Fortunately, even though the tests executed by CloudCmp may seem very synthetic, their case studies show that they are still fairly representative of the performance of more realistic applications.

While CloudCmp cannot be used to select for specific functions, it is useful to get a detailed insight into the performance and cost tradeoffs. CloudCmp does this by computing the performance metrics for the compute cluster, persistent storage and network connections. Users may find CloudCmp useful to quickly determine which cloud services fit their superficial needs before delving into deeper comparisons with AHP or TOPSIS. On the other side, cloud providers can use CloudCmp to discover performance bottlenecks in their systems or find ways to improve their service and remain competitive.

TOPSIS was also introduced as another technique for cloud provider selection problems. As already mentioned before, TOPSIS performs analysis of available cloud services using fuzzy opinions. This technique makes use of the criterion's weights produced by AHP as an input parameter. TOPSIS is a popular technique because of its theoretical rigor, ability to represent the human rationale during selection, and prominence in solving traversal ranks [16].

As all three methods have their relative merits, there is no clear winner here. AHP and TOPSIS are both very comprehensive, but require a great deal of set-up and run-time on the cloud platforms. CloudCmp on the other hand is simple, but also only offers a shallow insight into the performance and does not take special functionality into account. It is therefore our advice to use a combination of the three techniques.

If the application has a special need for specific functionality, AHP and TOPSIS may be used to select the providers that offer this and then CloudCmp could be used to find the cloud provider with the optimal performance/cost ratio. On the other hand, if the application does not rely on special functionality, but instead needs high performance in processing, disk I/O or network, CloudCmp can be used to select one or more providers that offer the required performance. If the CloudCmp comparison delivers in many close results, AHP and TOPSIS can then be used to make a finer selection.

5 Future Work

Since AHP has some disadvantages such as the decision maker subjectivity can yield uncertainties when determining pairwise comparisons, we will incorporate AHP with another technique called fuzzy AHP in order to overcome this problem. We defer this for future work.

Acknowledgements

We would like to thank our expert reviewer and anonymous reviewer who gave feedbacks in order to improve our paper.

References

[1] J. Baliga, R. W. A. Ayre, K. Hinton, and R. S. Tucker. Green cloud computing: Balancing energy in processing, storage, and transport. *Proceedings of the IEEE*, 99(1):149–167, Jan 2011.

[2] Bushan.R and Rhai.K. Strategic decision making applying the analytic hierarchy process, 2004.

[3] D. F. García and J. García. Tpc-w e-commerce benchmark evaluation. *Computer*, 36(2):42–48, Feb. 2003.

[4] R. L. Grossman. The case for cloud computing. *IT Professional*, 11(2):23–27, March 2009.

[5] P. Gupta, A. Seetharaman, and J. R. Raj. The usage and adoption of cloud computing by small and medium businesses. *International Journal of Information Management*, 33(5):861 – 874, 2013.

[6] A. Li, X. Yang, S. Kandula, and M. Zhang. Cloudcmp: Comparing public cloud providers. In *Proceedings of the 10th ACM SIGCOMM Conference on Internet Measurement*, IMC '10, pages 1–14, New York, NY, USA, 2010. ACM.

[7] X. R. Mingrui.S, Tianyi.Z. Consumer-centered cloud services selection using ahp, 2013.

[8] G. Motta, N. Sfondrini, and D. Sacco. Cloud computing: An architectural and technological overview. In *Service Sciences (IJCSS), 2012 International Joint Conference on*, pages 23–27. IEEE, 2012.

[9] National Center for Biotechnology Information. BLAST: Basic local alignment search tool. `https://blast.ncbi.nlm.nih.gov/Blast.cgi`. Accessed: 2017-03-03.

[10] S. Ramgovind, M. M. Eloff, and E. Smith. The management of security in cloud computing. In *2010 Information Security for South Africa*, pages 1–7, Aug 2010.

[11] T. Saaty. The analytic hierarchy process, 1980.

[12] J. R. San Cristóbal. *Multi criteria analysis in the renewable energy industry*. Springer Science & Business Media, 2012.

[13] K. Shiv, K. Chow, Y. Wang, and D. Petrochenko. Specjvm2008 performance characterization. In *Proceedings of the 2009 SPEC Benchmark Workshop on Computer Performance Evaluation and Benchmarking*, pages 17–35, Berlin, Heidelberg, 2009. Springer-Verlag.

[14] V. Srikrishna.S, Sreenivasulu Reddy. A. A new car selection in the market using topsis technique, 2014.

[15] The Trustees of Princeton University. Planetlab. `http://planet-lab.org/`. Accessed: 2017-03-03.

[16] M. Whaiduzzaman, A. Gani, N. B. Anuar, M. Shiraz, M. N. Haque, and I. T. Haque. Cloud service selection using multicriteria decision analysis. *The Scientific World Journal*, 2014, 2014.