Future Generation Computer Systems 29 (2013) 1012–1023

Contents lists available at SciVerse ScienceDirect

Future Generation Computer Systems

journal homepage: www.elsevier.com/locate/fgcs

A framework for ranking of cloud computing services

Saurabh Kumar Garg [a,*], Steve Versteeg [b], Rajkumar Buyya [a]

[a] Cloud Computing and Distributed Systems Laboratory, Department of Computing and Information Systems, University of Melbourne, 3053, Australia

[b] CA Technologies, Melbourne Victoria, 3004, Australia

ARTICLE INFO

Article history:
Received 1 March 2012
Received in revised form
6 June 2012
Accepted 11 June 2012
Available online 19 June 2012

Keywords:
Cloud computing
Service measurement
Quality of service
Service level agreement

ABSTRACT

Cloud computing is revolutionizing the IT industry by enabling them to offer access to their infrastructure and application services on a subscription basis. As a result, several enterprises including IBM, Microsoft, Google, and Amazon have started to offer different Cloud services to their customers. Due to the vast diversity in the available Cloud services, from the customer's point of view, it has become difficult to decide whose services they should use and what is the basis for their selection. Currently, there is no framework that can allow customers to evaluate Cloud offerings and rank them based on their ability to meet the user's Quality of Service (QoS) requirements. In this work, we propose a framework and a mechanism that measure the quality and prioritize Cloud services. Such a framework can make a significant impact and will create healthy competition among Cloud providers to satisfy their Service Level Agreement (SLA) and improve their QoS. We have shown the applicability of the ranking framework using a case study.

1. Introduction

Cloud computing has emerged as a paradigm to deliver on-demand resources (e.g., infrastructure, platform, software, etc.) to customers similar to other utilities (e.g., water, electricity and gas). The three main services are provided by the Cloud computing architecture according to the needs of IT customers [1]. Firstly, Software as a Service (SaaS) provides access to complete applications as a service, such as Customer Relationship Management (CRM) [2]. Secondly, Platform as a Service (PaaS) provides a platform for developing other applications on top of it, such as the Google App Engine (GAE) [3]. Finally, Infrastructure as a Service (IaaS) provides an environment for deploying, running and managing virtual machines and storage. Technically, IaaS offers incremental scalability (scale up and down) of computing resources and on-demand storage [1].

Traditionally, small and medium enterprises (SMEs) had to make high capital investment upfront for procuring IT infrastructure, skilled developers and system administrators, which results in a high cost of ownership. Cloud computing aims to deliver a network of virtual services so that users can access them from anywhere in the world on subscription at competitive costs depending on their Quality of Service (QoS) requirements [1]. Therefore, SMEs have no longer to invest large capital outlays in hardware to deploy their service or human expense to operate it. In other words, Cloud computing offers significant benefits to these businesses and communities by freeing them from the low-level task of setting up IT infrastructure and thus enabling more focus on innovation and creating business value for their services.

Due to such business benefits offered by Cloud computing, many organizations have started building applications on the Cloud infrastructure and making their businesses agile by using flexible and elastic Cloud services. But moving applications and/or data into the Cloud is not straightforward. Numerous challenges exist to leverage the full potential that Cloud computing promises. These challenges are often related to the fact that existing applications have specific requirements and characteristics that need to be met by Cloud providers.

Other than that, with the growth of public Cloud offerings, for Cloud customers it has become increasingly difficult to decide which provider can fulfill their QoS requirements. Each Cloud provider offers similar services at different prices and performance levels with different sets of features. While one provider might be cheap for storage services, they may be expensive for computation. For example, Amazon EC2 [4] offers IaaS services of the same computing capabilities at different pricing for different regions.

Therefore, given the diversity of Cloud service offerings, an important challenge for customers is to discover who are the "right" Cloud providers that can satisfy their requirements. Often, there may be trade-offs between different functional and non-functional requirements fulfilled by different Cloud providers. This makes it difficult to evaluate service levels of different Cloud

* Corresponding author.
E-mail addresses: saurabhkrgarg@gmail.com, sgarg@csse.unimelb.edu.au (S.K. Garg), Steve.Versteeg@ca.com (S. Versteeg), raj@csse.unimelb.edu.au (R. Buyya).

0167-739X/$ – see front matter © 2012 Elsevier B.V. All rights reserved.
doi:10.1016/j.future.2012.06.006

providers in an objective way such that the required quality, reliability and security of an application can be ensured. Therefore, it is not sufficient to just discover multiple Cloud services but it is also important to evaluate which is the most suitable Cloud service.

In this context, the Cloud Service Measurement Index Consortium (CSMIC) [5] has identified metrics that are combined in the form of the Service Measurement Index (SMI), offering comparative evaluation of Cloud services. These measurement indices can be used by customers to compare different Cloud services. In this paper, based on these identified characteristics of Cloud services, we are taking the state of the art one step further by proposing a framework (SMICloud) that can compare different Cloud providers based on user requirements. The SMICloud would let users compare different Cloud offerings, according to their priorities and along several dimensions, and select whatever is appropriate to their needs.

Several challenges are tackled in realizing a model for evaluating QoS and ranking Cloud providers. The *first is how to measure various SMI attributes of a Cloud service*. Many of these attributes vary over time. For example, Virtual Machine (VM) performance has been found to vastly vary from the promised values in the Service Level Agreement (SLA) by Amazon [4]. However, without having precise measurement models for each attribute, it is not possible to compare different Cloud services or even discover them. Therefore, SMICloud uses historical measurements and combines them with promised values to find out the actual value of an attribute. We also give precise metrics for each measurable attribute.

The second challenge is how to rank the Cloud services based on these attributes. There are two types of QoS requirements which a user can have: functional and non-functional. Some of them cannot be measured easily given the nature of the Cloud. Attributes like security and user experience are not easy to quantify. Moreover, deciding which service matches best with all functional and nonfunctional requirements is a decision problem. It is necessary to think critically before selection as it involves multiple criteria and an interdependent relationship between them. This is a problem of Multi-Criteria Decision-Making (MCDM) [6]. Each individual parameter affects the service selection process, and its impact on overall ranking depends on its priority in the overall selection process. To address this problem, we propose an Analytical Hierarchical Process (AHP) based ranking mechanism to solve the problem of assigning weights to features considering the interdependence between them, thus providing a much-needed quantitative basis for the ranking of Cloud services.

The rest of the paper is organized as follows. In the next section, we present an overview of SMI and its high level QoS attributes. Section 3 describes the SMICloud framework with its key components. Section 4 shows how metrics for various quality attributes can be modeled. Section 5 presents the Cloud ranking mechanism which is explained by a case study example in Section 6. Section 7 gives a brief overview of research work related to our work and finally we conclude this article in Section 8 with some future work.

2. Service measurement index (SMI)

SMI attributes are designed based on the International Organization for Standardization (ISO) standards by the CSMIC consortium [5]. It consists of a set of business-relevant Key Performance Indicators (KPIs) that provide a standardized method for measuring and comparing business services. The SMI framework provides a holistic view of QoS needed by the customers for selecting a Cloud service provider based on: Accountability, Agility, Assurance of Service, Cost, Performance, Security and Privacy, and Usability. There are currently no publicly available metrics or methods which define these KPIs and compare Cloud providers. SMI is the first effort in this direction. The following defines these high level attributes:

- Accountability—this group of QoS attributes is used to measure various Cloud provider specific characteristics. This is important to build the trust of a customer on any Cloud provider. No organization will want to deploy its applications and store their critical data in a place where there is no accountability of security exposures and compliance. Functions critical to accountability, which SMI considers when measuring and scoring services, include auditability, compliance, data ownership, provider ethicality, sustainability, etc.
- Agility—the most important advantage of Cloud computing is that it adds to the agility of an organization. The organization can expand and change quickly without much expenditure. Agility in SMI is measured as a rate of change metric, showing how quickly new capabilities are integrated into IT as needed by the business. When considering a Cloud service's agility, organizations want to understand whether the service is elastic, portable, adaptable, and flexible.
- Cost—the first question that arises in the mind of organizations before switching to Cloud computing is whether it is cost-effective or not. Therefore, cost is clearly one of the vital attributes for IT and the business. Cost tends to be the single most quantifiable metric today, but it is important to express cost in the characteristics which are relevant to a particular business organization.
- Performance—there are many different solutions offered by Cloud providers addressing the IT needs of different organizations. Each solution has different performance in terms of functionality, service response time and accuracy. Organizations need to understand how their applications will perform on the different Clouds and whether these deployments meet their expectations.
- Assurance—this characteristic indicates the likelihood of a Cloud service performing as expected or promised in the SLA. Every organization looks to expand their business and provide better services to their customers. Therefore, reliability, resiliency and service stability are important factors in selecting Cloud services.
- Security and Privacy—data protection and privacy are important concerns for nearly every organization. Hosting data under another organization's control is always a critical issue which requires stringent security policies employed by Cloud providers. For instance, financial organizations generally require compliance with regulations involving data integrity and privacy. Security and Privacy is multi-dimensional in nature and includes many attributes such as protecting confidentiality and privacy, data integrity and availability.
- Usability—for the rapid adoption of Cloud services, the usability plays an important role. The easier to use and learn a Cloud service is, the faster an organization can switch to it. The usability of a Cloud service can depend on multiple factors such as Accessibility, Installability, Learnability, and Operatibility.

3. SMICloud architecture

We propose the Service Measurement Index Cloud framework—SMICloud—which helps Cloud customers to find the most suitable Cloud provider and therefore can initiate SLAs. The SMICloud framework provides features such as service selection based on QoS requirements and ranking of services based on previous user experiences and performance of services. It is a decision making tool, designed to provide assessment of Cloud services in terms of KPIs and user requirements. Customers provide two categories of application requirements: essential and non-essential. Essential

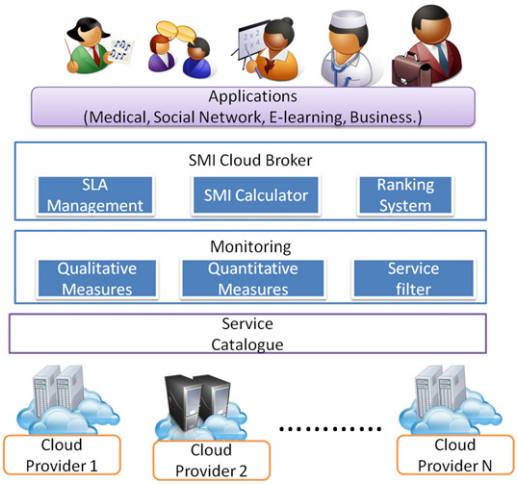


Fig. 1. SMICloud framework.

requirements allow the customer to specify 'deal-breakers', i.e. if a certain SMI attribute does not meet the required level, then the service is unacceptable, regardless of how all the other attributes are scored. The essential and non-essential requirements depend both on customers and their application needs. For example, for an academic user, the security level may not be an 'essential' requirement if their project is of no commercial significance. Based on these requirements, SMICloud gives as an output a sorted list of Cloud services which the customer can use to deploy their application. Fig. 1 shows the key elements of the framework:

1. SMICloud Broker: this component is responsible for interaction with customers and understanding their application needs. It collects all their requirements and performs discovery and ranking of suitable services using other components such as the SMICalculator and Ranking systems. SLA Management is the component that keeps track of customers' SLAs with Cloud providers and their fulfillment history. The Ranking System ranks the services selected by the Cloud Broker which are appropriate for user needs. The SMI Calculator computes the various KPIs which are used by the ranking system for prioritizing Cloud services.
2. Monitoring: this component first discovers Cloud services that can satisfy users' essential QoS requirements. Then, it monitors the performance of the Cloud services, for example for IaaS it monitors the speed of VMs, memory, scaling latency, storage performance, network latency and available bandwidth. It also keeps track of how SLA requirements of previous customers are being satisfied by the Cloud provider. For this layer, many tools are available, some of which we discuss in the related work section.
3. Service Catalogue: stores the services and their features advertised by various Cloud providers.

The two important issues in building the framework, as previously mentioned, are the measurement of various SMI KPIs and the ranking of Cloud services based on these measurements. In the next section, we present a QoS model for IaaS providers based on SMI KPIs. This model can be easily extended for SaaS and PaaS.

4. Quality model for IaaS provider

Cloud computing services can be evaluated based on qualitative and quantitative KPIs. Qualitative are those KPIs which cannot be quantified and are mostly inferred based on user experiences. Quantitative are those which can be measured using software and hardware monitoring tools. For example, providers' ethicality and security attributes are qualitative in nature. Since these KPIs represent generic Cloud services, only some of them are important for particular applications and Cloud services. For example, the installability attribute in usability is more relevant to IaaS providers than SaaS providers since in SaaS there is almost no installation on the customer end. In addition, the same KPI can have different definitions based on the service. Some of these parameters depend on customer applications and some are independent. For example, suitability is dependent on the customer while flexibility is determined by the provider. Therefore, it is complex to define precisely the SMI values for a provider, particularly when there are many parameters involved and parameter definitions also depend on many sub-attributes. Here we give some example definitions for the most important quantifiable KPIs, particularly in the context of IaaS. However, most of these proposed metrics are valid for other types of services. The modeling of qualitative attributes is beyond the scope of this paper.

4.1. Proposed metrics for cloud KPIs

4.1.1. Service response time

The efficiency of a service availability can be measured in terms of the response time, i.e. in the case of IaaS, how fast the service can be made available for usage. For example, if a user requests a virtual machine from a Cloud provider, then the service response time will represent the time taken by the Cloud provider to serve this request. This includes provisioning the VM, booting the VM, assigning an IP address and starting application deployment. The service response time depends on various sub-factors such as average response time, maximum response time promised by the service provider, and the percentage of time this response time level is missed.

- Average Response Time is given by $\sum_i T_i/n$ where T_i is time between when user i requested for an IaaS service and when it is actually available and n is the total number of IaaS service requests.
- Maximum Response Time is the maximum promised response time by the Cloud provider for the service.
- Response Time Failure is given by the percentage of occasions when the response time was higher than the promised maximum response time. Therefore, it is given by $100(n'/n)$, where n' is the number of occasions when the service provider was not able to fulfill their promise.

4.1.2. Sustainability

Sustainability is defined in terms of the environmental impact of the Cloud service used. It can be measured as the average carbon footprint or energy efficiency of the Cloud service. The property of Sustainability is classified as an Accountability attribute, which is used to measure properties related to the service provider organization itself independent of services being provided.

The carbon footprint metric is complex and depends on many factors. Therefore, SMICloud can get these values using Carbon calculators such as PUE Calculator [7]. Some of the well-known metrics that quantify various aspects of a Cloud datacenter can be used to calculate the energy efficiency and carbon footprint of a Cloud datacenter:

- DCiE and PUE: these are the most famous metrics used to measure energy efficiency of a Cloud computing service. DCiE (Data Center Infrastructure Efficiency) is defined as the percentage of the total facility power that goes to the IT equipment (primarily compute, storage, and network). PUE (Power Usage Efficiency) is the inverse of DCIE and is generally greater than one.

- DPPE (Data Center Performance per Energy) correlates the performance of the datacenter with carbon emissions. It is computed using the following formula:

$$DPPE = Data\ Center\ Work\ Carbon\ Energy = ITEU \times ITEE \times \frac{1}{PUE} \times \frac{1}{1 - GEC}$$

where

$$ITEU = \frac{Total\ Measured\ Energy\ of\ IT\,[KWh]}{Total\ Specification\ Energy\ IT\,(by\ manufacturer)\,[KWh]}$$

$$ITEE = \frac{a \sum Server\ Capacity + b \sum Storage\ Capacity + c \sum NW\ capacity}{Total\ Specification\ Energy\ IT\,(by\ manufacturer)\,[KWh]}$$

$$GEC = \frac{Green\ Energy}{DC\ Total\ Power\ Consumption}.$$

In the above formulation, ITEU represents the IT equipment utilization, ITEE represents the IT equipment energy efficiency, PUE represents the efficiency of the physical infrastructure and GEC represents the penetration of renewable (green) energy into the system. ITEU is the average utilization factor of all IT equipment included in the datacenter and can be considered as the degree of energy saving by virtual techniques and operational techniques that utilize the available IT equipment capacity without waste. ITEE aims to promote energy saving by encouraging the installation of equipment with high processing capacity per unit of electric power. Parameters a, b, c are weight coefficients.

4.1.3. Suitability

Suitability is defined as the degree to which a customer's requirements are met by a Cloud provider. There are two sub-cases before we can define suitability. First, if after filtering the Cloud providers, there is more than one Cloud provider which satisfies all the essential and non-essential requirements of the customer, then all are suitable. Otherwise, if filtering results in an empty Cloud provider list then those providers which satisfy the essential features are chosen. In this case, suitability will be the degree to which service features come closer to user requirements. The resultant metric is:

$$Suitability = \frac{number\ of\ non\text{-}essential\ features\ provided\ by\ service}{number\ of\ non\text{-}essential\ features\ required\ by\ the\ customer}$$

if only essential requirements are satisfied

$= 1$ *if all features are satisfied*

$= 0$ *otherwise*.

4.1.4. Accuracy

The accuracy of the service functionality measures the degree of proximity to the user's actual values when using a service compared to the expected values. For computational resources such as Virtual Machines, accuracy's first indicator is the number of times the Cloud provider deviated from a promised SLA. It is defined as the frequency of failure in fulfilling the promised SLA in terms of Compute units, network, and storage. If f_i is the number of times the Cloud provider fails to satisfy promised values for user i over the service time T, then accuracy frequency is defined as $\sum_i \frac{f_i}{n}$ where n is the number of previous users. Another indicator of accuracy is the accuracy value which is defined by $\sum_i \frac{(\alpha_t - \alpha_i)}{\alpha_i T_i}$, where α can be computational, network or storage unit of the service and T_i is service time T for user i.

4.1.5. Transparency

Transparency is an important feature of Cloud services due to the fast evolution of these services. According to CSMIC, transparency indicates the extent to which users' usability is affected by any changes in service. Therefore, it can be inferred as a time for which the performance of the user's application is affected during a change in the service. It can also be calculated in terms of the frequency of such effects. Therefore, it can be measured by $\sum \frac{1}{n} \sum \frac{time\ for\ service\ affect\ i}{number\ of\ such\ occurrences}$ where n is the number of customers using the service and i indicates the customer.

4.1.6. Interoperability

Interoperability is the ability of a service to interact with other services offered either by the same provider or other providers. It is more qualitative and can be defined by user experience. But since it is an important parameter for Cloud customers, we provide an approximation, which is defined as $\frac{number\ of\ platforms\ offered\ by\ the\ provider}{number\ of\ platforms\ required\ by\ users\ for\ interoperability}$.

4.1.7. Availability

The availability is the percentage of time a customer can access the service. It is given by: $\frac{(total\ service\ time) - (total\ time\ for\ which\ service\ was\ not\ available)}{total\ service\ time}$.

4.1.8. Reliability

Reliability reflects how a service operates without failure during a given time and condition. Therefore, it is defined based on the mean time to failure promised by the Cloud provider and previous failures experienced by the users. It is measured by:

$$Reliability = probability\ of\ violation \times p_{\text{mttf}} = \left(1 - \frac{numfailure}{n}\right) * p_{\text{mttf}}$$

where *numfailure* is the number of users who experienced a failure in a time interval less than promised by the Cloud provider, n is number of users, and p_{mttf} is the promised mean time to failure.

Reliability of storage can be defined in terms of durability, that is the chance of failure of a storage device.

4.1.9. Stability

Stability is defined as the variability in the performance of a service. For storage, it is the variance in the average read and write time. For computational resources, it is the deviation from the performance specified in SLAs, i.e., $\sum \frac{\frac{\alpha_{\text{avg},i} - \alpha_{\text{sla},i}}{T}}{n}$ where α can be computational unit, network unit or storage unit of the resource; $\alpha_{\text{avg},i}$ is the observed average performance of the user i who leased the Cloud service, $\alpha_{\text{sla},i}$ is the promised values in the SLA; T is the service time; and n is the total number of users.

4.1.10. Cost

Cost depends on two attributes: acquisition and on-going. It is not easy to compare different prices of services as they offer different features and thus have many dimensions. Even the same provider offers different VMs which may satisfy users' requirements. For instance, Amazon Cloud offers small VMs at a lower cost than Rackspace but the amount of data storage, bandwidth, and compute unit are quite different between two providers [4,8]. To tackle this challenge, we defined a volume based metric, i.e. the cost of one unit of CPU, storage, RAM, and network bandwidth. Therefore, if a VM is priced at p for *cpu* cpu units, *net* network units, *data* data unit and *RAM* memory units, then the cost of the VM is $\frac{p}{cpu^a * net^b * data^c * RAM^d}$ where a–c, and d are weights for each resource attribute and $a + b + c + d = 1$. The weight of each attribute can vary from application to application. For example, for some applications RAM is more important than CPU units,

therefore for this application $d > a$. We can use different weights for each attribute based on the user application. Generally users need to transfer data which also incurs cost. Therefore, the total on-going cost can be calculated as the sum of data communication, storage and compute usage for that particular Cloud provider and service.

4.1.11. Adaptability

Adaptability is the ability of the service provider to adjust changes in services based on customers' requests. It is defined as the time taken to adapt to changes or upgrading the service to a higher level (e.g. upgrading from a small Amazon VM to a medium size Amazon VM [4]).

4.1.12. Elasticity

Elasticity is defined in terms of how much a Cloud service can be scaled during peak times. This is defined by two attributes: mean time taken to expand or contract the service capacity, and maximum capacity of service. The capacity is the maximum number of compute units that can be provided at peak times.

4.1.13. Usability

The ease of using a Cloud service is defined by the attributes of Usability. The components such as operability, learnability, installability and understandability can be quantified as the average time experienced by the previous users of the Cloud service to operate, learn, install and understand it respectively.

4.1.14. Throughput and efficiency

Throughput and efficiency are important measures to evaluate the performance of infrastructure services provided by Clouds. Throughput is the number of tasks completed by the Cloud service per unit of time. It is slightly different from the Service Response Time metric, which measures how fast the service is provided. Throughput depends on several factors that can affect execution of a task. Let an user application have 'n' tasks and they are submitted to run on 'm' machines from the Cloud provider. Let $T_e(n, m)$ be the execution time of n tasks on m machines. Let T_o be the time overhead due to various factors such as infrastructure initiation delays and inter task communication delays. Therefore, the total throughput of a Cloud service is given by:

$$\alpha = \frac{n}{T_e(n, m) + T_o}.$$

The Cloud system efficiency indicates the effective utilization of leased services. Therefore, a higher value for efficiency indicate that the overhead will be smaller. System efficiency is given by: $\frac{T_e(n,m)}{T_e(n,m)+T_o}$.

4.1.15. Scalability

Scalability is important to evaluate in order to determine whether a system can handle a large number of application requests simultaneously. The ability to scale resources is an essential part of the elasticity provided by Cloud computing. However, this metric is more applicable from the performance perspective of user applications. The scalability has two dimensions: horizontal Cloud scalability (also known as 'scale out') and vertical Cloud scalability ('scale up'). Horizontal Cloud scalability means increasing Cloud resources of the same types such as initiating more virtual machines of the same type during peak load. Some of the aspects of horizontal scalability, we have already discussed during our discussion of measuring the elasticity of Cloud services. Therefore, to avoid overlap, here we consider only the vertical scalability that is defined as the ability to increase the capacity of a Cloud service such as a virtual machine by increasing resources such as the physical memory, CPU speed, or network bandwidth. Vertical scalability is an important quality measure for organizations who want to move to the Cloud. If the Cloud does not allow an application to scale well vertically, it can increase the costs of using Cloud services, particularly at peak times. The vertical scalability can be calculated as the maximum available increase in the resources of a Cloud service. Let r_{ij} be resource j that needs to be enhanced on Cloud service i. Let n and m be the number of resources assigned to a particular Cloud service and the number of Cloud services used by the user, respectively. The formulation of vertical scalability is: $\sum_i^m \sum_j^{n_i} (\textit{proportion of increase in } r_{ij})$.

4.2. Quality model assessment

In this section, we assess usefulness and practicability of the metrics proposed in this paper by using four criteria which are identified from IEEE Standard 1061 [9].

- *Correlation*. The metrics proposed in this paper are derived from quality attributes, i.e., KPIs required by the user's application. There is a strong linear association between quality attributes and their metrics. For example, Elasticity of a Cloud service depends on how fast the Cloud can grow and how much it can grow. Each of these values can affect the elasticity of an application. If a Cloud provider takes hours to increase the number of virtual machines, it will directly affect the QoS expected by the users.
- *Practical and computable*. According to this criterion, the proposed metrics should be computable practically with ordinate effort or time. Except for sustainability, the metrics proposed in this paper are easily computable by using various publicly available performance tools [10–12].
- *Consistency*. Similar to the criterion correlation, the values among quality attributes also have a strong linear association. If quality attribute values $A1$, $A2$, An, have the relationship $A1 > A2 > An$, then the corresponding metric values shall have the relationship $M1 > M2 > Mn$. It can be observed that each of the metrics is calculated based on numerical values of various performance characteristics of the Cloud service, therefore consistency is self-evident from the metrics.
- *Discriminative power*. The metric is capable of discriminating between high-quality Cloud services (e.g., short response time) and low-quality Cloud services (e.g., long response time). The set of metric values associated with the former should be significantly higher (or lower) than those associated with the latter. Let us assume there are three values of throughput for three Cloud services, i.e., Th_1, Th_2, Th_3,. Since each of these values are numerical in nature, we can have the relationship $Th_1 > Th_2 > Th_3$. Hence, in terms of the throughput, we can conclude that the first Cloud service can be ranked as the service which can handle the highest amount of workload.

5. Cloud service ranking

Ranking of Cloud services is one of the most important features of the SMICloud framework. The Ranking System computes the relative ranking values of various Cloud services based on the QoS requirements of the customer and features of the Cloud services. The ranking system takes into account two things before deciding from where to lease Cloud resources: (a) the service quality ranking based on AHP and (b) the final ranking based on the cost and quality ranking.

5.1. Service quality ranking using AHP

As discussed previously, Cloud services have many KPIs with many attributes and sub-attributes which makes the ranking

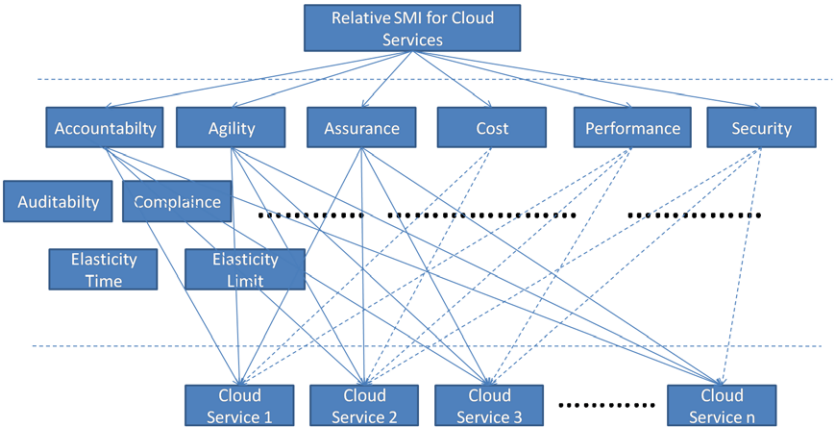


Fig. 2. AHP hierarchy for cloud computing.

process a complex task. This problem in the literature is defined as multiple criteria decision making (MCDM) [13]. The traditional weighted sum-based methods cannot be directly applied in such a hierarchical structure of attributes. In addition, some of the attributes do not have any numerical value, for example, security.

Without a structured technique, the evaluation of the overall quality of different Cloud services would be very difficult given the number of attributes involved. In addition, the challenge is to compare each Cloud service based on each attribute, how to quantify them and how to aggregate them in a meaningful metric. In general, such problems fall into the category of MCDM, where decision makers choose or rank alternatives on the basis of an evaluation of several criteria. Decision making involves managing trade-offs or compromises among a number of criteria that are in conflict with each other. There are three fundamental approaches to solving MCDM problems: Multiple Attribute Utility Theory (MAUT), outranking and Analytic Hierarchy Process (AHP). Most of the approaches proposed in the literature are variations of these three basic methods.

Multiple Attribute Utility Theory (MAUT) [14] is the simplest method that combines various preferences in the form of multiple attribute utility functions. In MAUT, utility functions for each criteria are combined with weighting functions of criteria. The primary advantage of using MAUT is that the problem is constructed as a single objective function after successful assessment of the utility function. Thus, it becomes easy to ensure the achievement of the best compromise solution based on the objective function.

The outranking approach is based on the principle of the degree of one alternative's dominance over another [15], rather than considering that a single best alternative can be identified. Outranking, thus, compares the performance of alternatives for each criterion and identifies the extent of a preference of one alternative over another without using a prescribed scale from the user. Outranking models are generally applied when aggregation of criteria metrics is not easy and measurement units are incommensurate or incomparable. The drawback of this approach is that many times it does not reach a decision and it is relatively complex to implement compared to other MCDM approaches.

Analytic Hierarchy Process (AHP) is one of the most widely used mechanisms for solving problems related to MCMD. AHP is a multi-criteria decision making approach that simplifies complex, ill-structured problems by arranging the decision factors in a hierarchical structure. Unlike MAUT, AHP is based on pairwise comparisons of decision criteria rather than utility and weighting functions. The pairwise comparison allows the decision maker to determine the trade-offs among criteria. The advantages of AHP over other multi-criteria methods are its flexibility, intuitive appeal to the decision makers and its ability to check inconsistencies [16]. In addition, AHP decomposes a decision problem into its constituent parts and builds hierarchies of criteria similar to KPIs in the SMI framework. AHP also helps to capture both subjective and objective evaluation measures. While providing a powerful mechanism for checking the consistency of the evaluation measures and alternatives, AHP reduces bias in decision making.

Therefore, to rank Cloud services based on multiple KPIs, we propose a ranking mechanism based on Analytic Hierarchy Process (AHP) [17]. There are three phases in this process: problem decomposition, judgment of priorities, and aggregation of these priorities. AHP gives a very flexible way for solving such problems and can be adapted to any number of attributes with any number of sub-attributes. In the first phase, the ranking of a complex problem is modeled in a hierarchy structure that specifies the interrelation among three kinds of elements, including the overall goal, QoS attributes and their sub-attributes, and alternative services. The second phase consists of two parts: a pairwise comparison of QoS attributes is done to specify their relative priorities; and a pairwise comparison of Cloud services based on their QoS attributes to compute their local ranks. In the final phase, for each alternative service, the relative local ranks of all criteria are aggregated to generate the global ranking values for all the services.

We describe the main steps to model the ranking problem in Cloud computing and then explain the overall calculation of ranks by a small case study example.

5.1.1. Phase 1: hierarchy structure for cloud services based on SMI KPIs

Fig. 2 presents the Cloud service hierarchy based on SMI KPIs. The first layer is the analysis goals, which aims to find the relative service management index of all the Cloud services which satisfy the essential requirements of the user. The second layer contains hierarchies of QoS attributes, both essential and non-essential. The bottommost layer contains the values of all the Cloud services for all the lowest QoS attributes in the hierarchy presented in the second layer.

5.1.2. Phase 2: computation of relative weights of each QoS and service

To compare two Cloud services, we need to assign weights to each attribute to take into account their relative importance. To address this issue we consider two types of weights:

- User assigned weights using AHP's standard method. The user of SMI Cloud can assign weights to each of the SMI attributes using values in some scale, for example [1..9] as suggested

Table 1
Relative importance value.

Equal importance/quality	1
Somewhat more important/better	3
Definitely more important/better	5
Much more important/better	7
Extremely more important/better	9

in the AHP method [17], to indicate the importance of one QoS attribute over another. The table of relative importance is given in Table 1. This methodology was proposed originally for calculating weights for each criteria in the AHP technique. This can be used to assign weights to all the QoS attributes. Users express their preferences for each attribute relative to other attributes.

- Arbitrary user assigned weights. A user can assign weights in their own scale rather than the one given by the AHP technique. In this case, the sum of all weights may not be 1 which is a requirement of AHP. For this case, we normalize all the weights.

5.1.3. Phase 3: relative value-based weights for ranking Cloud services

These weights define the relative performance of each Cloud service based on the values of the lowest level attributes. The process of assigning weights is not straightforward since the lowest level attributes can have various types of values. For example, the value of 'certifications' for a particular Cloud provider will be a list or a set. While the value of 'elasticity' will be a numerical value, values of some attributes may not be known. Therefore, the challenge is how to assign weights to each of the attributes when they are not quantifiable. To address this issue, we define the relative weights for ranking Cloud services based on a strategy proposed by Tran et al. [18]. In contrast to Tran et al.'s work, the relative weight metrics designed in our work also consider two types of QoS requirements of Cloud users, i.e. essential and non-essential.

Let w_q be the weight given by the user for SMI attribute q. Let v_i and v_j be the values of the attribute q for Cloud services i and j. Let s_i and s_j be the Cloud services and s_i/s_j indicate the relative rank of s_i over s_j. Let v_r be the required value specified by the user. To compare both values for each Cloud service, firstly, we first need to make sure that the dimensional units of both values are the same. For example, if we want to compare the price of two VM instances, then the price should be the price of 1 CPU unit, 1 memory unit, and 1 hard disk unit. If it is data communication, the cost should have the same dimension in terms of the price per 1 GB of data. Secondly, we have to compare the two values based on their types since the attribute values can vary from Boolean to a unordered set. For each type of attribute, a different comparison metric is proposed.

Thirdly, as discussed previously, users can specify essential and non-essential attributes. In the SMICloud framework, it is optional for users to specify their requirement values for non-essential attributes. Therefore, when comparing non-essential attributes for two services, there is a possibility that v_r is not specified by the user. Another consideration is that some of the attributes may not be possible to monitor with SMICloud due to the non-availability of such APIs by the Cloud provider. Our proposed relative ranking model is flexible enough to handle this issue; if any of the attribute values cannot be monitored then ranking weights assigned to Cloud services are based on weight w_q, or if v_r is not specified by the user, they are based on v_i and v_j.

The proposed relative ranking model for each type of attribute is given by:

Boolean:

$$\begin{aligned} s_i/s_j &= 1 \quad \text{if } v_i \equiv v_j \\ &= w_q \quad \text{if } v_j = 1 \text{ and } v_i = 0 \\ &= 1/w_q \quad \text{if } v_j = 0 \text{ and } v_i = 1. \end{aligned}$$

Numerical: it can be of two types, higher is better or lower is better. If higher is better then $\frac{v_i}{v_j}$ is the value of s_i/s_j. If the lower value is better, $\frac{v_j}{v_i}$ is the value of s_i/s_j. If q is a non-essential requirement, then it may be possible that one service may not have a value. In that case, s_i/s_j is equal to w_q if v_j is given, otherwise it will be $1/w_q$.

Unordered set: this can occur in attributes such as portability which may be defined by the number of platforms supported. To assign weights to Cloud services for such types of QoS attribute values, the size of the unordered set is considered. Let size(v_i) and size(v_j) be the number of elements in the sets for services i and j, respectively. Let size(v_r) be the size of set requested by the user for QoS attribute q. If q is an essential QoS attribute, then the Cloud service with the largest number of elements will be considered better and therefore higher weight will be assigned to it. The weights for such QoS attribute type values are calculated in the following way:

- if q is essential:

$$s_i/s_j = \frac{\text{size}(v_i)}{\text{size}(v_j)} \tag{1}$$

- if q is non-essential and if v_r is specified:

$$\begin{aligned} s_i/s_j &= \frac{\text{size}(v_i \cap v_r)}{\text{size}(v_j \cap v_r)} \quad \text{if } v_j \cap v_r \neq \phi \wedge v_i \cap v_r \neq \phi && (2)\\ &= 1 \quad \text{if } v_j \cap v_r \neq \phi \wedge v_i \cap v_r \equiv \phi && (3)\\ &= w_q \quad \text{if } v_j \cap v_r \neq \phi \vee v_i \cap v_r \equiv \phi && (4)\\ &= 1/w_q \quad \text{if } v_j \cap v_r \equiv \phi \vee v_i \cap v_r \neq \phi. && (5) \end{aligned}$$

Range type: many QoS attributes of Cloud services are given as a range of values. For example, the initiation time of a Virtual Machine can be represented as a range. In that case, if v_r is the value range required by the user, then the weights assigned to the services are:

- if q is essential:

$$s_i/s_j = \frac{\text{len}(v_i \cap v_r)}{\text{len}(v_j \cap v_r)} \tag{6}$$

- if q is non-essential and if v_r is specified:

$$\begin{aligned} s_i/s_j &= \frac{\text{len}(v_i \cap v_r)}{\text{len}(v_j \cap v_r)} \quad \text{if } v_j \cap v_r \neq \phi \wedge v_i \cap v_r \neq \phi && (7)\\ &= 1 \quad \text{if } v_j \cap v_r \equiv \phi \wedge v_i \cap v_r \equiv \phi && (8)\\ &= w_q \quad \text{if } v_j \cap v_r \neq \phi \wedge v_i \cap v_r \equiv \phi && (9)\\ &= \frac{1}{w_q} \quad \text{if } v_j \cap v_r \equiv \phi \wedge v_i \cap v_r \neq \phi. && (10) \end{aligned}$$

Using the above comparison metrics for each Cloud service, we obtain a one-to-one comparison of each Cloud service for a particular attribute. This will result in a one-to-one relative ranking matrix of size $N \times N$ if there are a total of N services. The relative ranking of all the Cloud services for the particular attribute is given by the eigenvector of the matrix. This eigenvector matrix is also called the Relative Service Ranking Vector (RSRV).

5.1.4. Phase 4: aggregation of relative ranking for each SMI attribute

In the final phase, the relative ranking vectors of each attribute are aggregated with their relative weights assigned in Phase 2. This aggregation process is repeated for all the attributes in the SMI hierarchy which results in the ranking of all the Cloud services based on KPIs.

5.2. Cost–quality based ranking

In the previous section, we proposed an approach to rank the Cloud services based on their KPIs. Although we can get from this approach the most appropriate Cloud service that can fulfill the user's requirements, the cost of service also plays a key role in the ranking process. In the literature [19], it is called the cost–value trade-off. A good ranking system should suggest to the user the Cloud service which gives the best QoS at the minimum cost. Therefore, in the final step, we compute the value/cost ratio for each service and finalize the ranking after analyzing this ratio. The complete process of ranking is illustrated in Section 6 with a case study.

5.3. Time complexity of AHP

In this section, we discuss the time complexity of the AHP based-ranking algorithm, which is used by SMICloud for each user request. The ranking mechanism consists of multiple phases. The first phase, to construct a hierarchy structure for cloud services, is a one time computation and will remain the same for all other requests. Thus, the time complexity of AHP depends mainly on the other three phases. Let there be m number of services to be compared, and L levels of attributes; each level has N_l number of attributes and n_{li} is the number of sub-attributes at level l of the ith attribute at level $l-1$.

For Phase 2, we need to compute relative weights for each QoS KPI. If the user assigns weights to KPIs between 0 and 1, then the time complexity of computing relative weights is linear. However, if the user assigns weights using AHP's standard method, then the time complexity of calculating the normalized weight vector for a group of sub-attributes is equivalent to the time taken for computing an eigenvector of size n_{li}, i.e.,$O((n_{li})^3)$ [17]. Therefore, the time complexity of computing relative weights for each level and each attribute is $O(\sum_{l=1}^{L}\sum_{i=1}^{N_{l-1}}((n_{li})^3))$.

For Phase 3, we calculate the relative weights of each Cloud service for the lowest level attributes. Since there are m services, the time complexity is $O(m^3N_L)$.

For Phase 4, aggregation of all relative ranking is done from bottom to top of the hierarchical structure constructed in the first phase. Each level has N_{l-1} groups of attributes and each group has n_{li} attributes that need to be aggregated for each service. The time complexity of aggregating all attributes at a level is the multiplication of two matrices, i.e. $O(N_l m)$. For all attributes at all levels, the time complexity is $O(m\sum_{l=1}^{L}N_l)$.

Therefore, the worst case total time complexity of the ranking mechanism is $O(\sum_{l=1}^{L}\sum_{i=1}^{N_{l-1}}((n_{li})^3)+m^3N_L+m\sum_{l=1}^{L}N_l)$.

6. Case study: ranking Cloud services based on QoS requirements

In this case study, the computation of the service index is done using the QoS data of three real Cloud providers. The QoS data is collected from various evaluation studies for three IaaS Cloud providers: Amazon EC2, Windows Azure, and Rackspace [12,20,21]. The unavailable data such as the security level is randomly assigned to each Cloud service. User weights are also randomly assigned to each QoS service attribute. The top level QoS groups are Accountability, Agility, Assurance, Cost, Performance and Security.

In the following, we show step by step the ranking computation process for Cloud services. The relative weighting method is used to calculate the relative ranking of Cloud services for each QoS attribute. For each attribute, a relative ranking matrix is constructed using the following method. Based on the data given in Fig. 3, the Relative Service Ranking Matrix (RSRM) for security will be:

$$RSRM_{\text{security}} = \begin{array}{c} \\ S1 \\ S2 \\ S3 \end{array} \begin{array}{c} \begin{array}{ccc} S1 & S2 & S3 \end{array} \\ \begin{pmatrix} 1 & 4/8 & 4/4 \\ 8/4 & 1 & 8/4 \\ 4/4 & 4/8 & 1. \end{pmatrix} \end{array}$$

Computing the Relative Service Ranking Vector (RSRV) for security from the matrix $RSRM_{\text{security}}$, we have

$$RSRV_{\text{security}} = [0.25\ 0.5\ 0.25].$$

Similarly, we have the relative service ranking vector of the Accountability: $RSRV_{\text{Accountability}} = [0.25\ 0.5\ 0.25]$.

For Agility, there are two QoS attributes which are further subdivided into sub-attributes. Elasticity of a Cloud service is inferred from the time it takes to scale up. Its RSRV is given by: $RSRV_{\text{Elasticity}} = [0.3470\ 0.1991\ 0.4538]$.

For each sub-attribute, i.e., CPU, memory and disk, RSRVs are given by:

$$RSRV_{\text{CPU}} = [0.3076\ 0.4102\ 0.2820]$$
$$RSRV_{\text{Memory}} = [0.3409\ 0.3181\ 0.3409]$$
$$RSRV_{\text{Disk}} = [0.3623\ 0.4373\ 0.2002].$$

Combining RSRV vectors of sub-attributes, i.e. CPU, memory and disk, we get the RSRM for 'Capacity':

$$RSRM_{capacity} = \begin{pmatrix} 0.30769 & 0.34090 & 0.36234 \\ 0.41025 & 0.31818 & 0.43738 \\ 0.28205 & 0.34090 & 0.20026 \end{pmatrix}.$$

Next, we compute the relative service ranking vector for the 'Capacity':

$$RSRV_{\text{capacity}} = \begin{pmatrix} 0.30769 & 0.34090 & 0.36234 \\ 0.41025 & 0.31818 & 0.43738 \\ 0.28205 & 0.34090 & 0.20026 \end{pmatrix} \begin{pmatrix} 0.5 \\ 0.3 \\ 0.2 \end{pmatrix}.$$

Therefore,

$$RSRV_{\text{capacity}} = \begin{pmatrix} 0.3286 & 0.3881 & 0.2834 \end{pmatrix}.$$

Similarly, the relative service ranking vector for Agility is given by:

$$RSRV_{\text{agility}} = \begin{pmatrix} 0.3286 & 0.34701 \\ 0.3881 & 0.19914 \\ 0.2834 & 0.45384 \end{pmatrix} \begin{pmatrix} 0.6 \\ 0.4 \end{pmatrix}$$
$$RSRV_{agility} = \begin{pmatrix} 0.336 & 0.3125 & 0.3516 \end{pmatrix}.$$

In a similar way we can compute the relative service ranking vector of all other top level QoS attributes, i.e., Assurance, Cost and Performance:

$$RSRV_{\text{assurance}} = \begin{pmatrix} 0.3812 & 0.2671 & 0.3517 \end{pmatrix}$$
$$RSRV_{\text{cost}} = \begin{pmatrix} 0.4073 & 0.3338 & 0.2589 \end{pmatrix}$$
$$RSRV_{\text{performance}} = \begin{pmatrix} 0.2846 & 0.1181 & 0.5973 \end{pmatrix}.$$

Finally, we aggregate all the RSRVs of all the attributes to get the relative service ranking matrix for three providers:

$$RSRM = \begin{pmatrix} 0.25 & 0.336 & 0.3812 & 0.1619 & 0.2846 & 0.25 \\ 0.5 & 0.3125 & 0.2671 & 0.1308 & 0.1181 & 0.5 \\ 0.25 & 0.3516 & 0.3517 & 0.7073 & 0.5973 & 0.25 \end{pmatrix}.$$

Top level QoS Groups (Weights)	First level Attributes (Weights)	Second Level Attributes (Weights)		Service 1 (S1)	Service 2 (S2)	Service 3 (S3)	Value Type	User Required Value
Accountability (.05)	level:0-10 (1)			4	8	4	Numeric	4
Agility (0.1)	Capacity (0.6)	CPU (0.5)	0.5	9.6	12.8	8.8	Numeric	4x1.6 GHZ
		Memory (0.3)	0.3	15	14	15	Numeric	10 GB
		Disk (0.2)	0.2	1690	2,040	630	Numeric	500 GB
	Elasticity (.4)	Time (1)	0.4	80-120	520-780	20-200	Range	60-120 sec
Assurance (0.2)	Availability (0.7)	0.7		99.95%	99.99%	100%	Numeric	99.9%
	Service Stability (0.2)	Upload Time (0.3)	0.3	13.6	15	21	Numeric	
		CPU (0.4)	0.4	17.9	16	23	Numeric	
		Memory (0.3)	0.3	7	12	5	Numeric	
	Serviceabilty (0.1)	Free Support (0.7)	0.7	0	1	1	Boolean	
		Type of Support (0.3)	0.3	24/7,Diagnostic Tools, Phone, Urgent Response	24/7,Diagnostic Tools, Phone, Urgent Response	24/7, Phone, Urgent Response	Unordered set	24/7, phone
Cost (0.3)	On-Going Cost (1)	VM Cost (0.6)	0.6	0.68	$0.96	0.96	Numeric	< 1 dollar/hour
		Data (0.2)	inbound	10	10	8	Numeric	100 GB/month
			outbound	11	15	18		200 GB/month
		Storage (0.2)	0.2	12	15	15	Numeric	1000 GB
Performance (0.3)	Service Response Time (1)	Range (0.5)	0.5	80-120	520-780	20-200	Range	60-120 sec
		Average Value (0.5)	0.5	100	600	30	Numeric	
Security (0.05)	level: 0-10 (1)			4	8	4	Numeric	4

Fig. 3. Case study example.

To get the final relative service ranking vector, we multiply the above RSRM with the weights of the top level QoS attributes:

$$RSRV = \begin{pmatrix} 0.25 & 0.336 & 0.3812 & 0.4073 & 0.2846 & 0.25 \\ 0.5 & 0.3125 & 0.2671 & 0.3338 & 0.1181 & 0.5 \\ 0.25 & 0.3516 & 0.3517 & 0.2589 & 0.5973 & 0.25 \end{pmatrix} \times \begin{pmatrix} 0.05 \\ 0.1 \\ 0.2 \\ 0.3 \\ 0.3 \\ 0.05 \end{pmatrix}.$$

Therefore, the relative ranking of all the Cloud services can be decided based on the resultant RSRV (0.3424, 0.2702, 0.3874). Based on the user requirements, the Cloud services are ranked as $S3 \succ S1 \succ S2$.

SMICloud allows users to visualize the differences between various Cloud services using Kivait graphs. For example, we can see from Kivait graphs (Fig. 4) how different Cloud services differ in providing best values for their KPIs. Cloud service S3 is best in terms of performance of the machine, however it is one of the lowest in terms of security. Therefore, S3 is a good alternative for the scientific community where security is a lower priority requirement and data is publicly available. On the other hand, Cloud service S2 is the best in terms of security which may be a key requirement for a user from a commercial organization.

In the final step, we will compute the ratio of quality (value) versus cost. To clearly see which Cloud service gives best value at minimum cost, we can visualize it using a graph as shown in Fig. 5. While Cloud service S3 has the best overall quality measure, it is also the most expensive. However, Cloud service S1 provides the best quality/cost ratio.

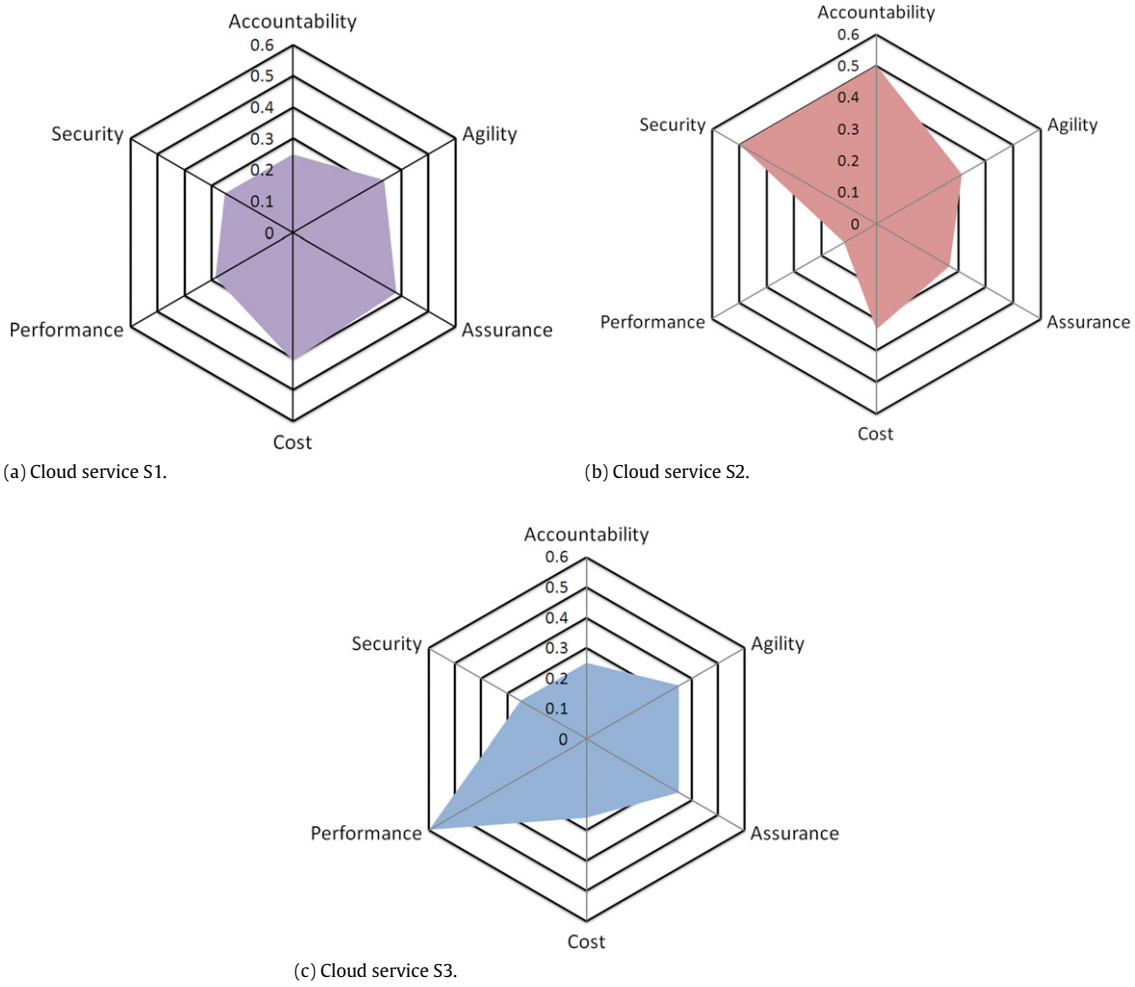
6.1. Overhead of ranking mechanism

Since execution time of the ranking mechanism highly depends on its implementation, analyzing the overhead of the ranking mechanism is not so direct. However, we can get an estimate by observing how much time ranking takes for different numbers of providers. We conducted an experiment to analyze the execution time of the ranking mechanism. We vary the number of providers while keeping the number of SMI attributes constant. As our main aim is to observe the overhead, we populated the data in regard to each attribute and their weights using a uniform distribution. In general, we can assume that for a particular type of application, the total number of SMI attributes will be bounded by a constant, then the worst case time complexity of our proposed mechanism depends on the number of providers. Fig. 6 shows that even up to 1000 service providers, the execution time is about 10 s which clearly indicates that our proposed methodology can be used for online selection during the acquisition of Cloud services.

7. Related work

In this section, we compare and contrast our work with previous research work for evaluating and comparing the performance of different Cloud services.

With the increasing popularity of Cloud computing, many researchers studied the performance of Clouds for different types of applications such as scientific computing, e-commerce and web applications. For instance, Iosup et al. [22] analyzed the performance of many-task applications on Clouds. Similarly, many performance monitoring and analysis tools are also proposed in the literature [22]. Our work complements these previous works by utilizing these tools and data to rank and measure the QoS of various Cloud services according to users' applications. Other works such as CloudCmp [12] proposed frameworks to compare the performance of different Cloud services such as Amazon EC2, Windows Azure and Rackspace. These works again focused on comparing the low level performance of Cloud services such as CPU and network throughput. In our work, we use performance data to measure various QoS attributes and evaluate the relative ranking of Cloud services.



(a) Cloud service S1.

(b) Cloud service S2.

(c) Cloud service S3.

Fig. 4. Comparison of Cloud services for different SMICloud KPIs.

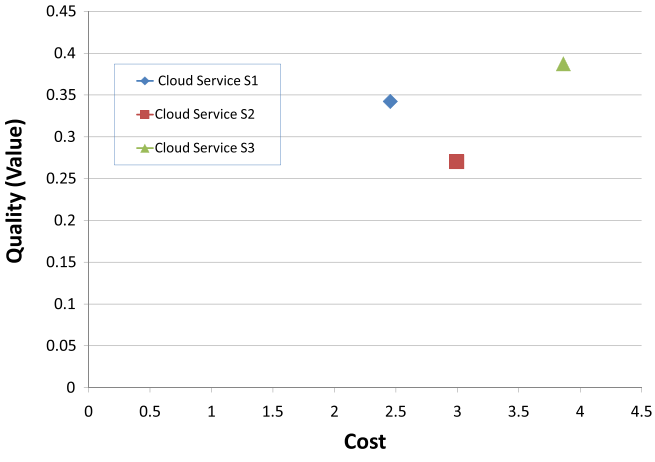


Fig. 5. Case study example.

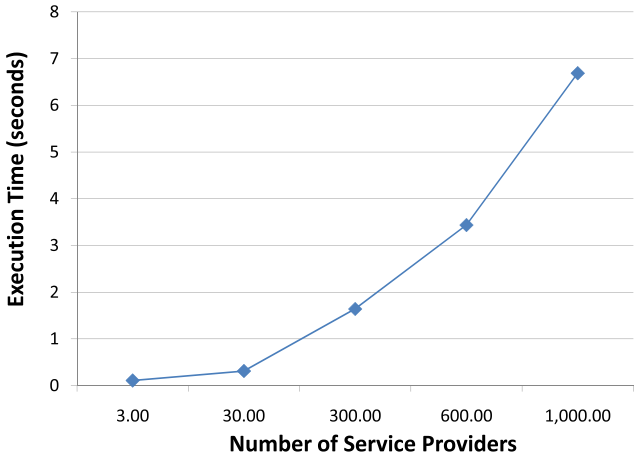


Fig. 6. Execution time of ranking mechanism.

In terms of performance metrics for Cloud computing, traditional High Performance Computing (HPC) benchmarks and metrics are not sufficient since they focus primarily on static system specific performance and cost [23]. Therefore, international consortiums such as CSMIC are working towards defining different measures for evaluating the performance of Clouds. Other than that, several performance benchmarks have been proposed. Binning et al. propose metrics for measuring scalability, cost, peak load handling, and the fault tolerance of Cloud environments [23]. The approach proposed by them is mainly focused on Web-based applications using standard TPC-W benchmarks [24]. Our goal is to develop a performance model for different Cloud computing systems rather than for a particular application. Yahoo! Cloud Serving Benchmark (YCSB) [25] aims to facilitate the performance

comparison between various Cloud data serving systems (e.g., Cassandra, HBase). Their work mainly focused on the analysis of performance implications on large database-intensive applications in the Cloud. Similar work is done by Kossmann et al. who proposed evaluation of end-to-end scalability aspects of Cloud database architectures [26]. They defined a set of performance and cost metrics for comparing the throughput, performance/cost ratio and cost predictability of Cloud database systems for increasingly large loads. Oh et al. proposed a framework and metrics to evaluate the re-usability of Cloud computing services [27]. Complementary to these works, our work focuses on performance analysis of IaaS providers, particularly in terms of their compute resource offerings, and we proposed a wider variety of measures which are specifically designed for Clouds. Several commercial providers have also started offering services to conduct detailed performance analysis of Cloud services such as CloudHarmony [11] which offers a service that can be used to view detailed benchmarking results of several Cloud providers. Such offerings can use our metrics to give more comprehensive results to their customers.

Even though the evaluation and comparative ranking of various Cloud services is quite new in the Cloud computing area, it is an old concept in other areas such as web services. The most related work in this area is done by Tran et al. [18]. This work also proposed a similar AHP based ranking technique. However, the algorithm was designed for web services and thus did not consider various performance parameters such as VM capacity which are specific to Cloud computing. In addition, we also define key performance and cost metrics based on the SMI [5] framework for Cloud computing services.

In summary, according to the authors' best knowledge, our work is the first to define all key performance metrics for QoS attributes in the SMI framework and apply AHP-based ranking in Cloud computing.

8. Conclusions and future work

Cloud computing has become an important paradigm for outsourcing various IT needs of organizations. Currently, there are many Cloud providers who offer different Cloud services with different price and performance attributes. With the growing number of Cloud offerings, even though it opens the chance to leverage the virtually infinite computing resources of the Cloud, it has also becomes challenging for Cloud customers to find the best Cloud services which can satisfy their QoS requirements in terms of parameters such as performance and security. To choose appropriately between different Cloud services, customers need to have a way to identify and measure key performance criteria that are important to their applications. Therefore, the Cloud Service Measurement Index Consortium (CSMIC) proposed a framework based on common characteristics of Cloud services. The aim of this consortium is to define each of the QoS attributes given in the framework and provide a methodology for computing a relative index for comparing different Cloud services.

In this context, this work presents the first framework, SMICloud, to systematically measure all the QoS attributes proposed by CSMIC and rank the Cloud services based on these attributes. We address some key challenges by designing metrics for each quantifiable QoS attribute for measuring precisely the service level of each Cloud provider. We proposed an Analytical Hierarchical Process (AHP) based ranking mechanism which can evaluate the Cloud services based on different applications depending on QoS requirements. Our proposed mechanism also addresses the challenge of different dimensional units of various QoS attributes by providing a uniform way to evaluate the relative ranking of Cloud services for each type of QoS attribute.

We believe the SMICloud framework represents a significant next step towards enabling accurate QoS measurement and Cloud service selection for Cloud customers. By using the techniques given in this work, Cloud providers can identify how they perform compared to their competitors and therefore they can improve their services.

In the future, we will extend our ranking algorithm to cope with variation in QoS attributes such as performance by adopting fuzzy sets. *We will also extend the quality model to non-quantifiable QoS attributes. We are also planning to implement the SMI framework and deploy on infrastructures provided by Amazon EC2 and Microsoft Azure. We will also extend a Cloud application platform such as Aneka* [28] *to utilise services of our framework while provisioning resources and scheduling execution of applications.*

Acknowledgments

This work is supported by the Australian Research Council (ARC) via Linkage Project grants with CA Technologies. We thank our colleagues especially Rodrigo N. Calheiros and Dileban Karunamoorthy for their comments on improving the paper. As a member of the Cloud Service Measurement Index Consortium (CSMIC), we benefited a lot from our participation. We would like thank all members of CSMIC for their constructive comments on this work.

References

[1] R. Buyya, C. Yeo, S. Venugopal, J. Broberg, I. Brandic, Cloud computing and emerging IT platforms: vision, hype, and reality for delivering computing as the 5th utility, Future Generation Computer Systems 25 (6) (2009) 599–616.
[2] M. Cusumano, Cloud computing and SaaS as new computing platforms, Communications of the ACM 53 (4) (2010) 27–29.
[3] E. Ciurana, Developing with Google App Engine, Apress, Berkeley, CA, USA, 2009.
[4] J. Varia, Best practices in architecting cloud applications in the AWS cloud, in: Cloud Computing: Principles and Paradigms, Wiley Press, 2011, pp. 459–490. (Chapter 18).
[5] Cloud Service Measurement Index Consortium (CSMIC), SMI framework. URL: http://beta-www.cloudcommons.com/servicemeasurementindex.
[6] J. Cochrane, M. Zeleny, Multiple Criteria Decision Making, Univ. of South Carolina Pr., 1973.
[7] H. Pan, Green Data Centers monthly newsletter February 2010, Information Gatekeepers Inc.
[8] Rackspace, Cloud servers, URL: http://www.rackspace.com.
[9] IEEE Standards Association and Others, IEEE STD 1061-1998, IEEE standard for a software quality metrics methodology, 1998.
[10] W. Sobel, S. Subramanyam, A. Sucharitakul, J. Nguyen, H. Wong, S. Patil, A. Fox, D. Patterson, Cloudstone: multi-platform, multi-language benchmark and measurement tools for web 2.0, in: Proceedings of Cloud Computing and its Application, Chicago, USA, 2008.
[11] C. Harmony, Cloudharmony.com, February 2012., http://cloudharmony.com/.
[12] A. Li, X. Yang, S. Kandula, M. Zhang, CloudCmp: comparing public cloud providers, in: Proceedings of the 10th Annual Conference on Internet Measurement, Melbourne, Australia, 2010.
[13] M. Zeleny, Multiple Criteria Decision Making, vol. 25, McGraw-Hill, New York, 1982.
[14] J. Dyer, Mautmultiattribute utility theory, Multiple Criteria Decision Analysis: State of the Art Surveys, 2005, pp. 265–292.
[15] J. Figueira, S. Greco, M. Ehrgott, Multiple Criteria Decision Analysis: State of the Art Surveys, vol. 78, Springer Verlag, 2005.
[16] R. Ramanathan, A note on the use of the analytic hierarchy process for environmental impact assessment, Journal of Environmental Management 63 (1) (2001) 27–35.
[17] T. Saaty, Theory and Applications of Analytic Network Process, vol. 4922, RWS Publications Pittsburgh, PA, 2005.
[18] V. Tran, H. Tsuji, R. Masuda, A new QoS ontology and its QoS-based ranking algorithm for web services, Simulation Modelling Practice and Theory 17 (8) (2009) 1378–1398.
[19] J. Karlsson, K. Ryan, A cost-value approach for prioritizing requirements, IEEE Software 14 (5) (1997) 67–74.
[20] J. Schad, J. Dittrich, J. Quiane-Ruiz, Runtime measurements in the cloud: observing, analyzing, and reducing variance, Proceedings of the VLDB Endowment 3 (1–2) (2010) 460–471.
[21] A. Iosup, N. Yigitbasi, D. Epema, On the performance variability of production cloud services, in: Proceedings of IEEE/ACM International Symposium on Cluster, Cloud, and Grid Computing, CA, USA.
[22] A. Iosup, S. Ostermann, N. Yigitbasi, R. Prodan, T. Fahringer, D. Epema, Performance analysis of cloud computing services for many-tasks scientific computing, IEEE Transactions on Parallel and Distributed Systems 22 (6) (2011) 931–945.

[23] C. Binnig, D. Kossmann, T. Kraska, S. Loesing, How is the weather tomorrow?: towards a benchmark for the cloud, in: Proceedings of the Second International Workshop on Testing Database Systems, RI, USA, 2009.
[24] D. Menascé, TPC-W: a benchmark for e-commerce, IEEE Internet Computing 6 (3) (2002) 83–87.
[25] B. Cooper, A. Silberstein, E. Tam, R. Ramakrishnan, R. Sears, Benchmarking cloud serving systems with YCSB, in: Proceedings of the 1st ACM Symposium on Cloud Computing, Indiana, USA, 2010.
[26] D. Kossmann, T. Kraska, S. Loesing, An evaluation of alternative architectures for transaction processing in the cloud, in: Proceedings of the 2010 International Conference on Management of Data, ACM, 2010, pp. 579–590.
[27] S. Oh, H. La, S. Kim, A reusability evaluation suite for cloud services, in: Proceeding of 2011 IEEE 8th International Conference on e-Business Engineering, ICEBE, Beijing, China, 2011.
[28] R. Calheiros, C. Vecchiola, D. Karunamoorthy, R. Buyya, The Aneka platform and QoS-driven resource provisioning for elastic applications on hybrid Clouds, Future Generation Computer Systems 28 (6) (2011) 861–870.

Saurabh Kumar Garg is a research fellow at the Cloud Computing and Distributed Systems (CLOUDS) Laboratory, University of Melbourne, Australia. He is one of the few Ph.D. students who completed in less than three years from the University of Melbourne. He has published more than 23 papers in highly cited journals and conferences. During his Ph.D., he has been awarded various special scholarships for his Ph.D. candidature. His research interests include resource management, scheduling, utility and grid computing, Cloud computing, green computing, wireless networks, and ad hoc networks.

Steve Versteeg is a Research Staff Member with CA Labs, based in Melbourne, Australia. His role is to coordinate collaborative research between universities and CA Technologies. His current projects are in the areas of cloud computing, software engineering, large scale endpoint emulation, role engineering and insider threat prediction. Steve's Ph.D. research was in the area of neural simulation. A well studied neural circuit was used as a case study for re-creating robust behaviour in computer systems. From 2004 until early 2008, Steve worked at WMind LLC as a senior developer and researcher on an experimental automated futures trading system. Steve holds a PhD in Computer Science from the University of Melbourne.

Rajkumar Buyya is Professor of Computer Science and Software Engineering; and Director of the Cloud Computing and Distributed Systems (CLOUDS) Laboratory at the University of Melbourne, Australia. He is also serving as the founding CEO of Manjrasoft., a spin-off company of the University, commercializing its innovations in Cloud Computing. He has authored over 400 publications and four text books. He also edited several books including "Cloud Computing: Principles and Paradigms" (Wiley Press, USA, Feb 2011). He is one of the highly cited authors in computer science and software engineering worldwide (h-index = 61 and 18500+ citations).