A Multi-criteria Scoring Method based on Performance Indicators for Cloud Computing Provider Selection

Lucas Borges de Moraes[1], Adriano Fiorese[1] and Fernando Matos[2]
[1]Dept. of Computer Science (DCC), Santa Catarina State University (UDESC), Joinville, Brazil
[2]Dept. of Computer Systems (DSC), Federal University of Paraíba (UFPB), João Pessoa, Brazil

Keywords: Cloud Computing Providers, Performance Indicators, Scoring Method, Ranking, Selection.

Abstract: Cloud computing is a service model that allows hosting and on demand distribution of computing resources all around the world, via Internet. Thus, cloud computing has become a successful paradigm that has been adopted and incorporated into virtually all major known IT companies (e.g., Google, Amazon, Microsoft). Based on this success, a large number of new companies were competitively created as providers of cloud computing services. This fact hindered the clients' ability to choose among those several cloud computing providers the most appropriate one to attend their requirements and computing needs. This work aims to specify a logical/mathematical multi-criteria scoring method able to select the most appropriate(s) cloud computing provider(s) to the user (customer), based on the analysis of performance indicator values desired by the customer and associated with every cloud computing provider that supports the demanded requirements. The method is a three stages algorithm that evaluates, scores, sorts and selects different cloud providers based on the utility of their performance indicators for each specific user of the method. An example of the method's usage is given in order to illustrate its operation.

1 INTRODUCTION

The evolution of information society brought the need of efficient, affordable and on-demand computational resources. The evolution of telecommunications technology, especially computer networks, provided a perfect environment for the rise of cloud computing. Cloud computing has shown a new vision of service delivery to its customers. It became a differentiated paradigm of hosting and distribution of computer services all over the world via Internet.

Cloud computing abstracts to the user the complex infrastructure and internal architecture of the service provider. Thus, to use the service, the user don't need to perform installations, configurations, software updates or purchase specialized hardware (Hogan et al., 2013). On this way, the cloud computing model has brought the benefit of better use of computing resources (Zhang et al., 2010). In addition to being a convenient service, it is easily accessible via the network and it is only charged for the time that is used (Armbrust et al., 2009; Zhang et al., 2010). In this model all computing resources that the user needs, can be managed by the cloud provider (Zhang et al., 2010).

The success of cloud computing paradigm is currently noticeable and it has been adopted in major IT companies like Google, Amazon, Microsoft and Salesforce.com and has become a good source of development/investment both in the academy and industry (Zhou et al., 2010; Höfer and Karagiannis, 2011). This success led to the rising of a large number of new businesses such as cloud computing infrastructure providers. With the increasing amount of new cloud providers the task of choosing and selecting which cloud providers are the most suitable for each user's need has become a complex process. The process of measuring the quality of each provider and compare them is not trivial, as there are usually many factors involved, many criteria to be studied and checked out throughout the process.

Measuring the quality and performance of a cloud provider (called Quality of Service or simply QoS) can be made using various strategies. One well-known strategy is numerically and systematically to measure the quality of each provider's performance indicators (PIs), reaching a certain value or score. Thus, providers can be ranked and the provider that offer the higher score is theoretically the most appropriate provider to that

Moraes, L., Fiorese, A. and Matos, F.
A Multi-criteria Scoring Method based on Performance Indicators for Cloud Computing Provider Selection.
DOI: 10.5220/0006289305880599
In *Proceedings of the 19th International Conference on Enterprise Information Systems (ICEIS 2017) - Volume 2*, pages 588-599
ISBN: 978-989-758-248-6

user.

The research questions that this study aims to investigate and answer are:

- What are and what kind of PIs are used to describe cloud computing providers?
- How to utilize these PIs to systematically measure the quality of each provider for each user?

The answer to the second question is obtained through the method to be specified in this work, that is, how to use the different data types (numbers, classes, subclasses) collected from each cloud provider and stored in different PIs to score a finite list of different providers according to the needs and requirements demanded by every possible consumer of resources of these cloud service providers. Each cloud computing service consumer is an user of the proposed method. The consumer can have x different requirements, wishing the w best ranked cloud providers based on expected values for m PIs of interest.

The developed method is a logical/mathematical algorithm able to select the w more suited providers for each specific user, scoring and ranking each provider. This process is based on the utility of each user's interest PIs for each available provider. The utility of each PI is calculated based on their type (quantitative or qualitative), the nature of the behaviour of its utility function (Higher is Better or HB; Lower is Better or LB; Nominal is Best or NB (Jain, 1991)), the desired/expected value by the user (indicated through the input expression) and the value of its competitors (other providers to be analyzed by the method).

Therefore, this work aims to propose a simple, intuitive (logical) and agnostic method with high generality and high dimensionality, that is, flexible and applicable to any PIs that may exist, regardless of its type (quantitative or qualitative) for n generic providers with m generic PIs, where n and m can grow indefinitely.

This paper is organized as follows: Section 2 presents and discusses different PIs found in the studied literature to qualify cloud computing providers. Section 3 presents related works to the selection, scoring and ranking of cloud providers based on indicators. Section 4 presents and discusses the proposed method that scores and ranks the different cloud providers based on user's interest PIs. Section 5 illustrates an example, with hypothetical data, that represents an application of the proposed method, in order to validate it and to demonstrate its operation presenting the results. Finally, Section 6 presents the final considerations.

2 PERFORMANCE INDICATORS FOR CLOUD COMPUTING PROVIDERS

This section aims to expose and clarify some performance indicators (PIs) used to evaluate and qualify the different cloud computing providers. Indicators are tools that allow a synthesized gathering information for a particular aspect of the organization using metrics that are responsible for quantifying (assigning a value) the study of objects to be measured. In general, the indicators can be classified into two categories (Jain, 1991): Quantitative (discrete or continuous) and qualitative (ordered or unordered).

- **Quantitative:** They are those states, levels or categories that can be expressed numerically, and can be worked algebraically. The numerical values assigned can be discrete or continuous. Examples of discrete quantitative indicators are: number of processors, amount of RAM (Random Access Memory), disk block size, etc. Example of continuous quantitative indicators are: response time, weight, length of an object, area of a land, etc.
- **Qualitative:** Also called categorical indicators. These indicators have distinct states, levels or categories that are defined by an exhaustive and mutually exclusive set of subclasses, which may or may not be ordered. The ordered subclasses have a perceptible logical graduation among their subclasses, giving the idea of a progression between them. Examples of ordered qualitative PIs: security level (low, medium, high), frequency of use of a service (never, rarely, sometimes, often, always), etc. The unordered subclasses do not have the idea of progression, e.g.: type of computer service (processing, storage, connectivity), research purpose (scientific, engineering, education), etc.

We can also classify PIs according to the behavior of its utility function (Jain, 1991). This means how useful (effective benefit) the PI becomes when its numerical value increases or decreases. There are three possible classifications (Jain, 1991):

- **HB (*Higher is Better*):** Users and/or system managers prefer the highest possible values for that indicator. For instance: System throughput, amount of resources (money, memory, materials, etc.), availability of a service, etc;
- **LB (*Lower is Better*):** Users and/or system managers prefer the lowest possible values for this

indicator. For instance: Response time, delay, costs, etc.;

- **NB (*Nominal is Best*):** Users and/or system managers prefer specific values. Higher and lower values are undesirable. A particular value is considered the best. The system load is an example of this feature. A very high system utilization is considered bad to users because it generates high response times. On the other hand, a very low utilization is considered bad by system managers since the resources are not being used (idle).

For the cloud computing paradigm we have a especial set of PIs called key performance indicators (KPIs) defined at Service Measurement Index (SMI). The SMI was developed by the Cloud Service Measurement Index Consortium (CSMIC) (CSMIC, 2014) and represents a set of KPIs that provide a standardized method for measuring and comparing cloud computing services. It also provides metrics and guidelines to help organizations measuring cloud-hosted business services and it works as a framework that provides an holistic view of the quality of service required by cloud computing consumers. The SMI is a hierarchical structure whose upper level divides the measurement space into seven categories and each category is optimized by four or more attributes (subcategories). The seven major categories are (CSMIC, 2014): accountability, agility, service assurance, financial, performance, security and privacy, usability.

Figure 1 depicts a mental map that displays and classifies several PIs that can be used for evaluation and monitoring of cloud computing service providers according other technical literature (Garg et al., 2011; Garg et al., 2013; Sundareswaran et al., 2012; Shirur and Swamy, 2015; Baranwal and Vidyarthi, 2014). The PIs presented do not represent an exhaustive list of all existing PIs. They form a portion of the indicators most often found in scientific papers studied. These PIs can be quantitative (integer or real numbers), qualitative (represented by a category or a set of them - simple categorical or compound categorical) and/or may even fall into both types (can appear as quantitative or qualitative). It is important to note that PIs with boolean values were classified as qualitative (at the proposed method they will be treated as unordered qualitatives with only two categories). The selection method proposed in this work is this classification and it is agnostic, that is, its user can use any PI that he wishes (since its present for at least one provider registered in the database of the method), not limited to those listed in this Section.

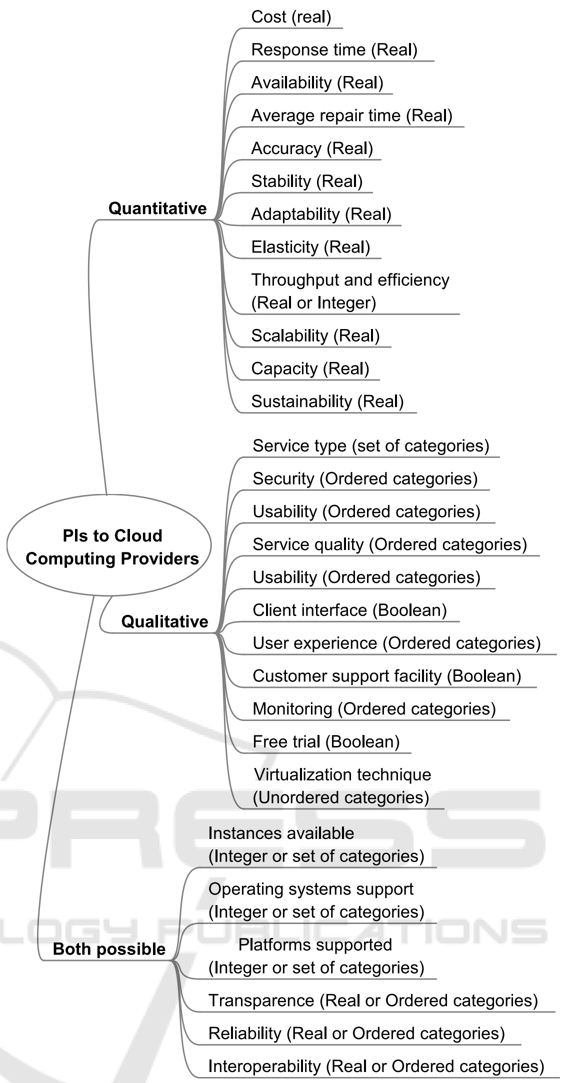

Figure 1: Classification of different PIs for cloud computing providers.

3 RELATED WORKS

This section presents related works already developed by other authors to rank and select cloud computing providers based on indicators.

Sundareswaran and others (Sundareswaran et al., 2012) proposed a new brokerage architecture in the cloud, where brokers are responsible for selecting the appropriate service for each user/customer. The broker has a contract with the providers, collecting their properties (performance indicators), and with consumers, collecting their service requirements. It analyzes and indexes the service providers according to the similarity of their properties. When the broker receives a cloud service selection request the broker will search the index to identify an ordered list of candidate providers based on how well they meet the needs of users.

The authors (Shirur and Swamy, 2015) specify a framework to quantify the efficiency of different cloud computing providers through the Quality of Service metrics (QoS). Based on that, the proposed framework ranks cloud computing providers. The framework divides the QoS metrics into two categories: application dependent metrics (reliability, availability, security, data center, cost, operating systems support, platforms supported, service response time, throughput and efficiency) and user dependent metrics (reputation, client interface, free trial, certification, sustainability, scalability, elasticity and user experience).

A framework called "SmiCloud" is presented in (Garg et al., 2013). It is responsible for measuring the quality of service (QoS) of cloud providers and ranking them based on that calculated quality. The quality is directly related to the values of each metric of the Service Measure Index (SMI) (CSMIC, 2014) classified into functional and non-functional. The work uses the Analytical Hierarchical Process (AHP) (Saaty, 2004) in the calculation of the quality and ranking of providers.

The framework developed in (Baranwal and Vidyarthi, 2014) presents an expectation of QoS metrics (also based on SMI) that the every cloud provider should have. This expectations are then used by a cloud broker that assists selecting the most appropriate ones. That framework uses a voting method that takes into account the user requirements for ranking cloud providers.

The work developed in (Wagle et al., 2015) proposes an evaluation model that verifies the quality and the status of service provided by cloud providers. The data is obtained by cloud auditors and is viewed via a heat map ordered by the performance of each provider, showing them in descending order of overall quality of service provided. This map represents a visual recommendation aid system for cloud consumers and cloud brokers. The main metrics are again based on the SMI: availability (divided in uptime, downtime and interrupt frequency), reliability (divided in load balancing, MTBF, recoverability), performance (latency, response time and throughput), cost (per storage unit and per VM instance) and security (authentication, encryption, and auditing).

The work developed in (Achar and Thilagam, 2014) present a broker based architecture for selecting the more suitable cloud provider based on the measurement of the quality of service provided. This approach prioritizes the selection of those providers who fits better the request sent to the broker. Selection involves three steps: To identify the proper and necessary PIs to request, evaluating the weight of each of these criteria using the AHP method, and ranking of each provider using the Technique for Order Preference by Similarity to Ideal Solution (TOPSIS), used to select the alternative which is closest to the ideal solution and farthest from negative ideal solution.

4 THE PROPOSED SELECTION METHOD

This section aims to present and discuss the proposed cloud service providers selection method. The following subsections expose how the method works, presenting a method overview, its inputs and outputs, steps and mechanisms for calculating the score and ranking of each cloud provider.

4.1 Method Overview

Figure 2 presents an overview of the selection method to be described in this Section. The database of cloud provider candidates and their performance indicators can be fed indirectly through websites such as "Cloud Harmony" (*https://cloudharmony.com*) or through cloud providers by their own (e.g: Amazon) or it can be consolidated by third parties.

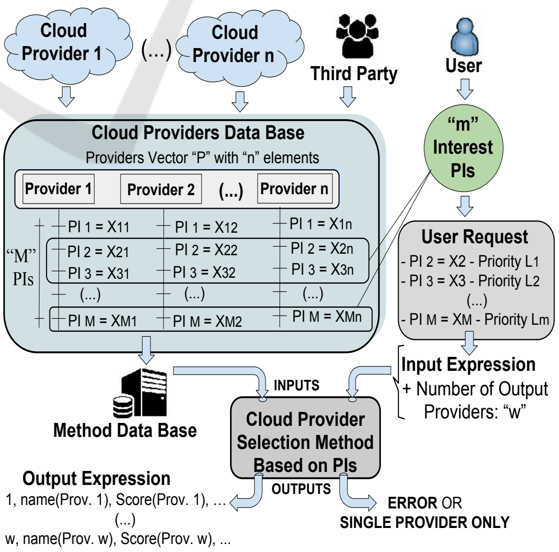

Figure 2: Multicriteria method for selecting cloud providers based on performance indicators.

Data input (Inputs) corresponds to a list P with n different candidates (cloud providers), each one with M different PIs (whose values are known), and an input expression (generated by the user of the method) containing m PIs of interest (subset of the known M PIs) and the priority level of each one. This priority level is set by the method's user according the classification adopted in the proposed method. The initial cloud providers list P will be filtered, at the first step of the method, based on the input expression and at the end it will have n' elements (with $n \geq n'$). If $n' = 0$, there is none available compatible provider to the user, so the method interrupts the process with an error message; if $n' = 1$, there is only a single compatible provider, which will be returned to the user; however, if $n' > 1$ the method proceeds along to another stage to rank providers. The expected output data (Outputs), except the special conditions mentioned, is a list with the cloud providers better scored by the method. The proposed method is divided into three main stages:

1. Elimination of incompatible cloud providers to the user;
2. Evaluation and scoring of interest PIs for each priority level;
   (a) Score quantitative PIs;
   (b) Score qualitative PIs.
3. Calculation of the final score for each cloud provider, ranking them and return results to the user.

4.2 Stage 1 – Elimination of Incompatible Cloud Providers

Figure 3 summarizes what occurs in the first stage of the method. In fact, the initial list of candidate providers (P) is cleaned from all incompatible cloud providers (concept discussed further), generating a new list P' with n' different providers.

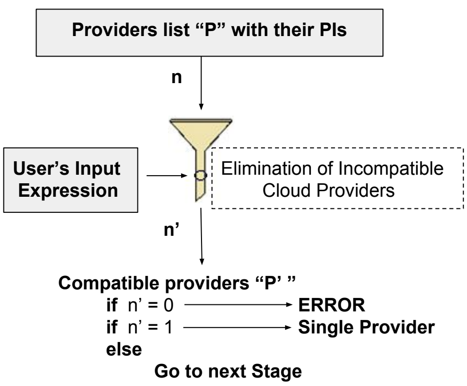

Figure 3: Stage 1 – Elimination of incompatible providers.

For the proposed method, we have created the classification of PIs' priority levels presented in Figure 4. Each PI can be classified as essential or non-essential by method's user. The non-essential PIs have different priority levels, which can vary between levels "High" and "Low". In order to simplify this work, it was adopted only one intermediate level of priority, named "Medium".

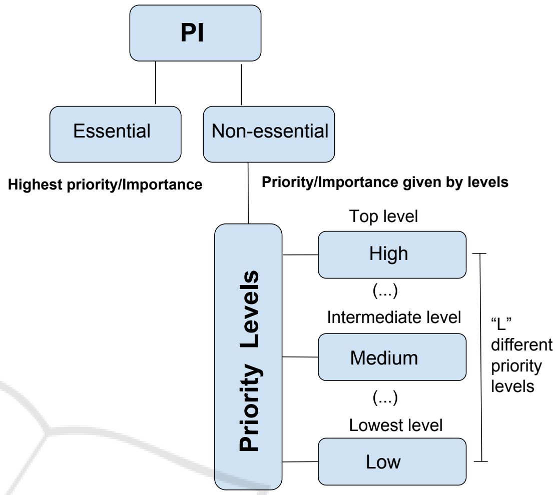

Figure 4: Classification of priorities of the PIs in the proposed method.

An essential PI has the highest priority/importance for the customer (user) and consequently to the proposed method. It indicates that if the specific value entered in the Input expression is not satisfied, the user cannot achieve their goals. This makes it a criterion of elimination. Thus, a provider that does not attend all essential PIs will be automatically deleted from the list P of valid candidates (compatible providers) because it is incompatible to that user in question.

On the other hand, if a non-essential PI is not attended, it does not incapacitate the user to achieve their goals. But, it can jeopardize them. How much it could be jeopardized (seriously or very little) is directly related to the priority level of the PI, set by the user. Do not attending a very high level PI means that the user will be seriously impaired in achieving their goals. On the other hand, do not attending a minimum priority level PI implies that the negative impact will be virtually imperceptible. The optional PIs can be automatically classified with minimum priority level.

In this work, it will be considered incompatible, the cloud provider that does not attend all the user's essential PIs. The concept of incompatibility and, in general, the scores of cloud providers, are based on the premise of attending/matching the desired PIs values (made available in the input expression by the user). However, what means a particular PI j attends a specific y value desired/required by the user? The answer to this question depends on the type of the PI and the behavior of the utility function of j. Thus, if j is quantitative, there are three possible classifications for the behavior of its utility function: HB (Higher is Better), LB (Lower is Better) or NB (Nominal is the best) (Jain, 1991).

Thus, given a quantitative PI j that stores the value x (number) and j belongs to the cloud provider i, if j attends the value y, specified by the user, then we can conclude that:

$$x \text{ attends } y \text{ then } \begin{cases} x \geq (y - t_j) \text{ if } j \in HB \\ x \leq (y + t_j) \text{ if } j \in LB \\ x = (y \pm t_j) \text{ if } j \in NB \end{cases} \quad (1)$$

Where t_j represents a certain tolerance regarding x, that is, a deviation from the desired value y tolerated by the user. Comprising the proposed method, the default tolerance value is zero, but it can be adjusted by the user via input expression.

Meanwhile, if the PI j is qualitative, it can be ordered or unordered (Jain, 1991). If it is unordered, the rule is simple: if x is the value that the user specify (y), then the PI j is attended, otherwise it is not. However, if PI j is ordered each value (category or class) has a certain relationship with the others, scaling from a lower level to a higher level. If the user specifies a low level value, a higher level value can also satisfy, it depends on the PI in question. An example of this are the qualitative PIs security and quality of service with values: “low”, “medium” and “high” (Sundareswaran et al., 2012). If the PI is quality of service and the user specify the value “medium”, the value “low” would not be appropriate, but the value “high” would be equally good, or even better. For the PI security level this isn’t always true, because a very high degree of security can be harder to work and this may impair the user’s work. Thus, higher and/or lower level values (categories) than the desired category y can also satisfy the user. Therefore, to solve this problem an ontology, similar to the one in (Jain, 1991), was created in order to indicate if an ordered qualitative PI has tolerances for categories below and/or above the desired category.

- **Higher is Tolerable (HT):** Categories above of the desired one are tolerable;
- **Lower is Tolerable (LT):** Categories below of the desired one are tolerable;
- **Higher and Lower are Tolerable (HLT):** Categories above and below of the desired one are both tolerable;

These tolerances can be set by the user via input expression. If nothing is informed, the default used is NB. Thus, given a qualitative PI j, which stores the value x (category), and j belongs to the provider i, if j attends the value y, specified by the user, then we can conclude that:

$$x \text{ attends } y \text{ then } \begin{cases} x = y \text{ if j is unordered OR } j \in NB \\ x \geq y \text{ if j is ordered AND } j \in HT \\ x \leq y \text{ if j is ordered AND } j \in LT \\ \text{j is ordered AND } j \in HLT \end{cases} \quad (2)$$

In any case of PI j, whether quantitative or qualitative, whose value x does not satisfy Equation 1 or Equation 2, respectively, it is said that j doesn’t attend the value of y, specified by the user. Therefore, taking into account the PI attending premise and the incompatibility concept, at the end of this initial stage it will remain a list of candidate cloud providers containing only those compatible ones with the requirements of essential PIs to be selected.

4.3 Stage 2 – Evaluation and Scoring Interest PIs for Each Priority Level

This second stage aims to score each provider individually, according to the utility (real benefit) of each one of its PIs. The higher the utility value associated with the PI, the higher the score. The utility is influenced by the value specified by the user (desired one) and also regarding the best value (bigger utility) among all candidate providers for that specific PI.

This stage will receive the list P', with the n' filtered providers from the previous stage. Each provider presents values for the m user’s interest PIs, which can be quantitative or qualitative. Each one of these PIs has a priority level associated with that is set by the user in the input expression. Thus, if L is the number of different available priority levels and m_l the amount of PIs with the lth level of priority, the score (Pts_l) for the ith provider is given by Equation 3.

$$Pts_l(i) = \frac{\sum_{k=1}^{m_l} (Pts(PI_k))}{m_l} \quad (3)$$

That is, the score of the lth priority level is the simple arithmetic average of the individual scores for each PI_k with the same priority level l, whether the PI is quantitative or qualitative. This stage ends when all the L levels are scored for each of the n' available providers. For example, in this work we considered four priority levels ($L = 4$): “Essential”(always maximum level), “High”, “Medium” and “Low”. Thus, each provider i will always have four scores at the end of this stage; one for each priority level. Regardless of the priority level, the individual score ($Pts(PI_k)$) for a quantitative and a qualitative PI is calculated in different ways.

4.3.1 Scoring Quantitative PIs

The score of a PI j of a provider i, will be 0 if its numerical value x doesn’t attend the numerical value y, specified by the user. If the value is attended, it will be scored in proportion to how useful (utility) this value is compared to all other compatible providers available in the candidates list (given by a constant). The evaluation function of a quantitative PI is shown in Equation 4. It always returns a normalized real number between 0 and 1 ($\forall\, value(j), y, X_{max}, X_{min} \geq 0$ and $C_1, C_2, t > 0$).

$$Pts(j) = \begin{cases} 0, \text{ if } value(j) \text{ doesn't attend } y \\ C_1 + C_2 * \dfrac{value(j) - y}{X_{max} - y}, & \text{if } j \in HB \\ C_1 + C_2 * \dfrac{y - value(j)}{y - X_{min}}, & \text{if } j \in LB \\ C_1 + C_2 * \dfrac{t - |y - value(j)|}{t}, & \text{if } j \in NB \end{cases} \tag{4}$$

Where the real constants (empirical parameters) C_1 and C_2 belong to the normalized open interval between $]0,1[$, and $C_1 + C_2 = 1$, mandatorily. The number X_{max} is the highest value among all other n providers in the list P' for that PI j; as well as X_{min} is the lowest value and t is the maximum tolerated distance (number) from the optimum point (y) for a NB PI (since that PI attends y, that is, belongs to the interval $[y-t; y+t]$). The value of t can be configured by the user.

The same happens with the coefficients C_1 and C_2. They can be tuned according the user understanding of how to weight the PI attending and how proportionally attended is the PI regarding the same PI on other providers. The constant C_1 weighs the score given the desired minimum match (including the tolerances associated with the PI, if any) between the value in the provider (x) and the value that the user wants (y), based on the PI type under analysis (HB, LB or NB). The constant C_2 weighs the score given to how much this PI value excels the desired minimum, that is, the value x is, in practice, better than the desired value y. It is noteworthy that the first coefficient (C_1) must be greater than the second one (C_2), because it is not interesting to weight more how better a PI is comprising its value in other cloud providers, in prejudice of the attending the user desired value. Also, it is essential that $C_1 + C_2 = 1$.

For this work, it was initially adopted $C_1 = 0.7$ and $C_2 = 0.3$. Thus, to the fact that a quantitative PI attends the value y (desired by the user), it is given a score of 0.7 (70% of the total score). The other 0.3 (30% of the total score) comes from how well ranked this PI is among all other competitors in the list of compatible candidate providers (P'). If the PI has the best value among all, the evaluation returns 1. If it has the lowest (but still attends the given value y) then $Pts(j)$ is 0.7. Summing up, if PI does not attend the value y, then evaluation returns 0, otherwise it returns a value between 0.7 and 1. Thus, when $j \in HB$, the higher its value, the closer to 0.3 will be the second term of the sum in Equation 4. When $j \in LB$, the lower its value, the closer to 0.3 will be the second term of the sum in Equation 4. Finally, when $j \in NB$, the closer to the y value, the closer to 0.3 the second term of the sum in in Equation 4.

4.3.2 Scoring Qualitative PIs

In spite of numerical values, qualitative PIs (or categorical) have categorical (string) ones. Therefore, qualitative PIs can be ordered or unordered. Comprising the unordered qualitative PIs, or the category is that one the user specified (receiving score 1) or not (receiving score 0). Regarding ordered qualitative PIs, the score depends on the tolerance supported and informed by the user (HT, LT, or HLT) from the PI’s value (category) offered by the provider. Categories of higher and/or lower levels to the desired category y can be tolerable to the user and may thus scoring. If nothing is mentioned about it in the input expression it is concluded that there is no tolerance and scoring proceeds in the same way as for unordered qualitative PIs.

For this work it was defined that the score of tolerable categories will be directly influenced by the disparity between the category specified by the user (y) and that offered by the provider (A). This means the greater the distance of the category in question (A) to the desired one (y) (both given by integers), the lower the score for that PI. Figure 5 presents how the score is influenced by the type of tolerance associated with a qualitative ordered PI, how to specify the categorical levels between the PI categories and how calculate the distance between these levels. The numbers identifying the desired category (y) and the tolerated ones are underlined. The score of a tolerated category is a constant multiplied by the normalized distance between the categories A and y.

The score of an ordered qualitative PI will always

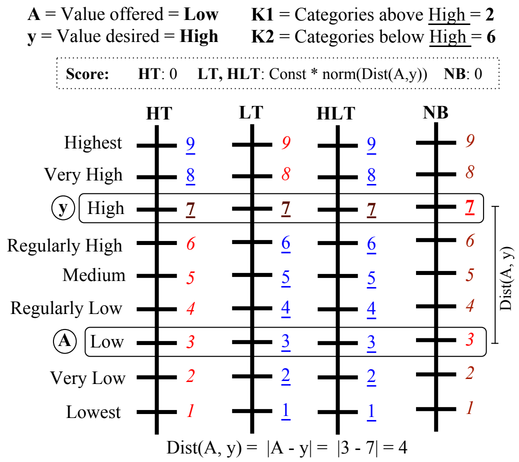

Figure 5: Relationship between the score, category-value and the type of tolerance associated with a hypothetical ordered qualitative PI with 9 distinct categories.

be a real value between 0 and 1. Therefore, in case of perfect match between the desired category (set by the user in the input expression) and the cloud provider offered category, then that category receives the maximum score 1. When do not occurs a perfect match, categories are scored according Equation 5. In this case, the desired neighboring categories (above and below) will score C_3 (with $0 < C_3 < 1$) and so on. In this sense, in Equation 5, $A = value(j)$ is the category under consideration (value of the PI j that is offered by the cloud provider), y the user's desired category and K_1, K_2 and $K_3 = K_1 + K_2$, the total number of tolerable categories, higher, lower to y or both depending on the PI is HT, LT or HLT, respectively. The distance between the categories $value(j)$ and y is the difference between their levels in module: $|level(value(j)) - level(y)|$. To each category is assigned a level value associated with a positive integer from 1 to the total of categories available (for that ordered qualitative PI), in increasing order of graduation (lower levels, lower numbers, higher levels, higher numbers, according to Figure 5). It is important to note that the bigger the distance between the category $value(j)$ and the desired category y, the smaller the score.

$$Pts(j) = \begin{cases} C_3 * \dfrac{K_1 - dist(A,y) + 1}{K_1}, & \text{if } j \in HT \\ C_3 * \dfrac{K_2 - dist(A,y) + 1}{K_2}, & \text{if } j \in LT \\ C_3 * \dfrac{K_3 - dist(A,y) + 1}{K_3}, & \text{if } j \in HLT \end{cases} \tag{5}$$

The real constant C_3 represents the maximum score that a tolerable category (another category different of the desirable y, but within the tolerances associated with that particular PI) can assume. Therefore, the smaller the value of C_3, more aggressive is the penalty (loss of score) applied to any and every PI j, whose category $value(j)$ diverges from the desired optimal value y. If $C_3 = 0$, then the method punctuates with zero any value different than y, whether the PI is ordered or not. This is an undesirable behavior, since it depreciates sub-optimal, and can penalize excessively providers that are also appropriate for the user. If $C_3 = 1$, the method depreciates the importance of reaching the optimal point for a qualitative PI, assigning too much punctuation to sub-optimal ones, encouraging the wrong choice of the best provider(s). For this work it was initially adopted as a starting point $C_3 = 0.7$, that is, a category next to y and within the tolerance will receive a score of 0.7 (70 % of the total score).

4.4 Stage 3 – Final Scoring for Each Cloud Provider

Previous two stages results in four scores for each cloud provider from the list P': one for each priority level: "Essential", "High", "Medium" and "Low", including quantitative and qualitative PIs. The consolidation of these scores in a single value will be the provider's final score. Therefore, a weighted arithmetic average will be used, where the coefficients (weights) are directly proportional to the priority levels. Equation 6 presents the score to a certain provider i. It is worthwhile to note that the sum of all weights must be 1 ($\alpha_1 + \ldots + \alpha_L = 1$).

$$P_{final}(i) = \sum_{l=1}^{L} (\alpha_l * Pts_l(i)) \tag{6}$$

An efficient technique for calculating each coefficient (α_l) is to use a matrix of judgements. A judgement matrix aims to model relationships (e.g.: importance, necessity, discrepancy, value, etc.) between the judged elements (Saaty, 2004). In this case, the elements to be judged (regarding the determination of weights) are the priority levels of each PI. Therefore, the judgement matrix is a matrix with dimension L, wherein each row and each column represents a different priority level, arranged in descending order of priority (from top to bottom – lines, from left to right – columns). This technique is used several times in the decision-making method called Analytic Hierarchy Process (AHP) (Sari et al., 2008; Ishizaka and Nemery, 2013; Fiorese et al., 2013).

Comprising this work, which has only four levels of priority, Table 1 presents a possible judgement matrix. The assigned values are based on the scale of Saaty (Saaty, 2004). In this case, the values in the judgement matrix indicate how important is the line element i with respect to the column element j. Thus, following this methodology to build the judgement matrix, we obtain all values in the diagonal equal to 1 and the observed inversions. On the last line, the elements of each column are summed up in order to advance the next step to find the weights, which is the normalization of this judgement matrix.

Table 1: Matrix of judgments: Relations of importance between each different priority levels.

Levels	Essential	High	Medium	Low
Essential	1	2	4	9
High	$1/2$	1	2	6
Medium	$1/4$	$1/2$	1	3
Low	$1/9$	$1/6$	$1/3$	1
Col. sum.	1,8611	3,6667	7,3333	19,00

Following the judgement matrix technique, the judgement matrix normalization takes place. This process takes each column element divided by its Col. sum. position, according Table 1. The results can be seen in Table 2, which represents Table 1 normalized.

Table 2: Normalized judgement matrix for each priority level.

Levels	Essential	High	Medium	Low
Essential	0,5373	0,5455	0,5455	0,4737
High	0,2687	0,2727	0,2727	0,3158
Medium	0,1343	0,1364	0,1364	0,1579
Low	0,0597	0,0455	0,0455	0,0526

Finally, the weights for each one of the priority levels are resolved summing the values on the priority level line of the normalized matrix and dividing the result by the number of priority levels (L), which is 4 in this case. Therefore, Table 3 shows the resolved priority level weights.

Table 3: Performance indicators' priority level weights.

Levels	**Weights**
Essential	0,5255
High	0,2825
Medium	0,1413
Low	0,0508

Thus, after the consistency checks on the judgement matrix (Saaty, 2004; Sari et al., 2008; Ishizaka and Nemery, 2013), which allowed its normalization and the weights get resolved, we have Equation 7, where the unknowns α_l of the Equation 6 are resolved. Therefore, Equation 7 represents the cloud provider score.

$$P_{final}(i) = 0,5255 * Pts_{ess.}(i) + 0,2825 * Pts_{high}(i) + 0,1413 * Pts_{med.}(i) + 0,0508 * Pts_{low}(i) \tag{7}$$

It is worth to note that the score of each provider i is normalized between 0 and 1. After scoring all providers, the list of compatible providers is ordered by score in descending order. Then, the proposed method returns the w first providers in that ordered list (highest scores) to the user.

5 USING THE PROPOSED METHOD

Once the cloud service provider selection method is specified, it is necessary to expose an example of its application on a set of real or hypothetical data in order to show its operation helping to answer questions about its procedure and results. Thus, this Section aims to apply the specified method on a possible data set. Table 4 shows an example of data that can be used for that. It shows 5 fictitious providers, each one with 7 interest PIs to the user. This data set represents the information regarding the candidate cloud providers that will be selected using the proposed method.

Table 4: Data set example of PIs to cloud providers.

PI/Provider	$Prov_1$	$Prov_2$	$Prov_3$	$Prov_4$	$Prov_5$
NI	5	3	9	12	10
NOS	1	2	4	8	6
Cost (U$/h)	0,30	0,50	1,00	1,50	1,20
RAM (Gb)	4	2	4	8	16
Storage (Gb)	10,0	50,0	80,0	15,0	5,0
Avail (%)	99,9	95,0	88,0	99,0	90,0
Sec	M	M	H	L	M
Key					
NI: Total types of available Virtual Machines. Integer ≥ 1.					
NOS: Total available operating systems. Integer ≥ 1.					
Cost: Average cost of the desired service. Given in U$/h.					
RAM: Average amount of RAM available. Given in GByte.					
Storage: Average amount of data storage. Given in GByte.					
Avail: Availability of the service per year, in average. Given in %.					
Sec: Estimated level of information security and privacy. It has 3 possible categories: High (H), Medium (M) and Low (L).					

Several steps compose the proposed method execution. The first step copes with the identification of the nature of each PI and with checking which one attend and which one do not attend the desired values specified by the user. To accomplish that, it is necessary that the proposed method recognizes that the PI "NI" is quantitative discrete with utility function HB; "NOS" is quantitative discrete NB; "Cost" is quantitative continuous LB; "RAM" is quantitative discrete HB; "Storage" is quantitative continuous HB; "Avail" is quantitative continuous HB; and "Sec" is qualitative ordered NB. This matching can be done since the cloud provider PIs database is kept updated by experts or by the user of the method, including this information about the PI's nature. Once the PI nature is acknowledged, the user needs to provide an input expression comprising which requirements (PIs), their values (including, eventually, tolerances and their values) as well as their priority levels. This input expression intends to be used by the proposed method to rank the PI attending providers, returning back to the user the w best ranked (when there are w). Table 5 shows an user input data used for this method working example. Values in brackets represent the tolerance for that PI.

Thus, taking into account the PI values provided by the user for this example, Table 6 shows the desirable values (based on the utility functions associated with them) and tolerable values (in accordance with tolerance values associated with utility functions provided by the user) for each PI.

Continuing the analysis of the PI values required by the user and those provided by cloud providers, Table 7 shows provider's max/min PI values needed for the final scoring of each provider.

Table 5: Example of data input entered by the user.

PIs and their values	Tolerance	Priority levels
$RAM \geq 4$	–	Essential
$Storage \geq 5$	0,5 GB	Essential
$Cost \leq 1,00$	0,10 U\$/h	High
$Avail \geq 90$	0,5 %	High
$NI \geq 8$	1	Medium
$Sec = Medium$	HT	Medium
$NOS = 3$	1	Low

Table 6: Analysis of the PIs presented in example.

PI	Desirable values	Tolerable values
RAM (GB)	$[4, +\infty)$	–
Storage (GB)	$[5,0; +\infty)$	$[4,5; 5,0)$
Cost (U\$/h)	$[0; 1,00]$	$(1,00; 1,10]$
Avail (%)	$[90,0; 100,0]$	$[89,5; 90,0)$
NI	$[8, +\infty)$	7
Sec	Medium	High
NOS	3	2 and 4

Thus comparing Tables 4 , 5 and 6 it is possible to determine if there is incompatible providers to eliminate from the list P (methods's Stage 1). It is

Table 7: Max/Min PIs values from providers.

PI	Max/Min values
RAM (HB)	$X_{max} = 16$ GB
Storage (HB)	$X_{max} = 80,0$ GB
Cost (LB)	$X_{min} = 0,30$ U\$/h
Avail (HB)	$X_{max} = 99,9$ %
NI (HB)	$X_{max} = 12$
Sec (NB)	–
NOS (NB)	$t = 1$

incompatible any provider that does not attend all the essential PIs (i.e., "RAM" and "Storage"). A PI attends certain desired value, if its value is in the range of desirable values or at least in the range of tolerable values, both identified in Table 6. On that basis, it was built Table 8, where (●) informs that the PI attends the user's value and (○) that it does not attend.

Table 8: Identification of provider's PIs that attend the user's desired values.

PI/Provider	$Prov_1$	$Prov_2$	$Prov_3$	$Prov_4$	$Prov_5$
RAM (E)	●	○	●	●	●
Storage (E)	●	●	●	●	●
Cost (H)	●	●	●	○	○
Avail (H)	●	●	○	●	●
NI (M)	○	○	●	●	●
Sec (M)	●	●	●	○	●
NOS (L)	○	●	●	○	○

Observation of Table 8 allows us to conclude that only cloud provider $Prov_2$ does not attend the essential PI "RAM". This observation is backed up to Table 4 that shows $Prov_2$ has only 2 GB of RAM, leaving user requirement of 4 GB or more, unattended. Thus, regarding the five PIs used, the only incomparable provider is $Prov_2$ and, therefore it should be removed from the list of suitable/compatible providers. Thus, the next stages only will consider the new generated list P' containing all providers (and their PIs), except $Prov_2$.

Next, on Stage 2, proposed method must score the quantitative and qualitative (Subsection 4.3.2) PIs, using Equation 4 and Equation 5, respectively, regarding $Prov_1, Prov_3, Prov_4$ e $Prov_5$. The score, by priority levels, as requested by user, is calculated according Equation 3. Table 9 presents the final scores by priority level of the PIs comprising each compatible provider. The constants used are: $C_1 = 0.7, C_2 = 0.3$ e $C_3 = 0.7$.

Next, on Stage 3, final score is calculated and consequently the ranking for each one of the four competing providers. This task is performed to each provider, according to Equation 7 (Subsection 4.4).

Table 9: Providers' scores by priority level.

Provider	Priority Levels			
	Essential	High	Medium	Low
$Prov_1$	0,71	1	0,5	0
$Prov_3$	0,85	0,35	0,7375	0,7
$Prov_4$	0,77	0,485	0,5	0
$Prov_5$	0,85	0,35	0,925	0

Coefficient values used in the weighted average are the values shown in Table 2. These weights are applied to the scores of each priority level already calculated. Table 10 presents the final score to the cloud providers 1, 3, 4 and 5.

Thus, according to Table 10, it is possible to rank the 4 competing providers in scoring descending order:

1. $Prov_1$ with 0,7263 points;
2. $Prov_3$ with 0,6853 points;
3. $Prov_5$ with 0,6763 points;
4. $Prov_4$ with 0,6123 points.

Therefore, $Prov_1$ is the most suitable to the user in this example. The proposed method can return to the user a list containing the w better ranked/ordered providers. Thus, given $w = 3$, the return would be: $\{1, Prov_1, 0.7263\}$, $\{2, Prov_3, 0.6853\}$,$\{3, Prov_5, 0.6763\}$.

6 FINAL CONSIDERATIONS

This work specified a multi-criteria scoring method to assist decision making that scores and ranks (orders) cloud computing providers in order to select the most suitable ones, based on the user's requirements (criteria) regarding their performance indicators. The requirement must present the performance indicators (PIs) of interest, the preference (desired) values for each PI and the priority of one over the other, that is, their importance to the fulfilment of user goals. The specified selection method comprises an intuitive and simple way to calculate whether the value of certain PI fits (attends) the user desired value and how to score it in a manner consistent with other competing available providers.

The proposed method is agnostic regarding which PIs to use in order to score, rank and select cloud providers. This means user can request, in his/her input expression, any PI and desired value. Notwithstanding, this work presented a method utilization example that has considered a set of indicators present in five works (Garg et al., 2011; Garg et al., 2013; Sundareswaran et al., 2012; Shirur and Swamy, 2015; Baranwal and Vidyarthi, 2014). In addition, we also used the Service Measure Index (SMI) framework (CSMIC, 2014), which provides a good set of PIs to measure and compare cloud computing services.

The proposed method is designed following three stages: 1) removing incompatible providers; 2) scoring quantitative and qualitative PIs by priority level and calculating final scores to providers; 3) ranking them and return results to user. The proposed method separates PIs into two types: essential and non-essential. The non-essentials have different degrees of importance, giving rise to distinct priority levels, thus, the higher the importance, the higher the priority. The final cloud provider score takes into account this priority. Thus, higher priority levels have larger weights and consequently they have higher influence on the final score. The final score is a real number between 0 and 1. The closer to 1, the most appropriate and preferable is that provider in relation to its competitors for that user in question.

The main benefits of the method are its high generality and high dimensionality, that is, the ability to work with any and all available PIs regardless of whether it is quantitative or qualitative, and its database can be easily and indefinitely expanded (number of total providers and the number of PIs of each provider to be considered for selection). The method is also simple and intuitive, since it does not require sophisticated mathematical and modelling skills to understand or use it.

The major limitation of the method is the need to have as pre-requisite a large database of cloud computing providers with the respective PIs registered for each provider. It is necessary to establish relationships of trust with the providers if it is decided that they will provide such data or, if it comes from third parties, they must somehow to ensure the data veracity. In addition to obtaining the data from the providers it is necessary to classify them: quantitative (HB, LB or NB) and qualitative (NB, HT, LT or HLT).

Other factor to consider is the need to adjust parameters C_1, C_2 and C_3 in the method. Although these parameters give more flexibility to the method, if they are poorly adjusted the method efficiency will be seriously compromised. The parameters are empirical constants that need several tests to draw more precise conclusions, mainly regarding the ratio of C_1 to C_2. It is mandatory to respect the domain presented (be a real between 0 and 1) and the restrictions: $C_1 + C_2 = 1, C_1 > C_2$ and $C_3 < 1$.

An example of the method was presented, demonstrating its use and the convenience of its adoption.

Table 10: Final score calculation.

General formula: $P_{final}(i) = 0,5255 * Pts_{essential}(i) + 0,2825 * Pts_{high}(i) + 0,1413 * Pts_{medium}(i) + 0,0508 * Pts_{low}(i)$	
Provider	**Final score**
$Prov_1$	$0,5255*(0,71)+0,2825*(1,00)+0,1413*(0,50)+0,0508*(0)=0,7263$
$Prov_3$	$0,5255*(0,85)+0,2825*(0,35)+0,1413*(0,7375)+0,0508*(0,70)=0,6853$
$Prov_4$	$0,5255*(0,77)+0,2825*(0,485)+0,1413*(0,50)+0,0508*(0)=0,6123$
$Prov_5$	$0,5255*(0,85)+0,2825*(0,35)+0,1413*(0,925)+0,0508*(0)=0,6763$

Future work includes testing the proposed method in realistic settings, as well as the creation of a cloud computing broker that incorporates the developed method.

ACKNOWLEDGEMENT

The authors would like to thank UDESC PROBIC scientific financial programme.

REFERENCES

Achar, R. and Thilagam, P. (2014). A broker based approach for cloud provider selection. In *2014 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, pages 1252–1257.

Armbrust, M., Fox, A., Griffith, R., Joseph, A. D., Katz, R., Konwinski, A., Lee, G., Patterson, D., Rabkin, A., Stoica, I., and Zaharia, M. (2009). Above the clouds: A berkeley view of cloud computing. Technical Report UCB/EECS-2009-28, University of California at Berkeley.

Baranwal, G. and Vidyarthi, D. P. (2014). A framework for selection of best cloud service provider using ranked voting method. In *Advance Computing Conference (IACC), 2014 IEEE International*, pages 831–837.

CSMIC (2014). Service measurement index framework. Technical report, Carnegie Mellon University, Silicon Valley, Moffett Field, California. Accessed in November 2016.

Fiorese, A., Matos, F., Alves Junior, O. C., and Rupeenthal, R. M. (2013). Multi-criteria approach to select service providers in collaborative/competitive multi-provider environments. *IJCSNS - International Journal of Computer Science and Network Security*, 13:15–22.

Garg, S. K., Versteeg, S., and Buyya, R. (2011). Smicloud: A framework for comparing and ranking cloud services. In *2011 Fourth IEEE International Conference on Utility and Cloud Computing (UCC)*, pages 210–218.

Garg, S. K., Versteeg, S., and Buyya, R. (2013). A framework for ranking of cloud computing services. *Future Generation Computer Systems*, 29:1012–1023.

Höfer, C. N. and Karagiannis, G. (2011). Cloud computing services: taxonomy and comparison. *Journal of Internet Services and Applications*, 2:81–94.

Hogan, M. D., Liu, F., Sokol, A. W., and Jin, T. (2013). *Nist Cloud Computing Standards Roadmap*. NIST Special Publication 500 Series. accessed in September 2015.

Ishizaka, A. and Nemery, P. (2013). *Multi-Criteria Decision Analysis: Methods and Software*. John Wiley & Sons, Ltd, United Kingdom.

Jain, R. (1991). *The Art of Computer Systems Performance Analysis: Techniques for Experimental Design, Measurement, Simulation, and Modeling*. John Wiley & Sons, Littleton, Massachusetts.

Saaty, T. L. (2004). Decision making – the analytic hierarchy and network processes (ahp/anp). *Journal of Systems Science and Systems Engineering*, 13:1–35.

Sari, B., Sen, T., and Kilic, S. E. (2008). Ahp model for the selection of partner companies in virtual enterprises. *The International Journal of Advanced Manufacturing Technology*, 38:367–376.

Shirur, S. and Swamy, A. (2015). A cloud service measure index framework to evaluate efficient candidate with ranked technology. *International Journal of Science and Research*, 4.

Sundareswaran, S., Squicciarin, A., and Lin, D. (2012). A brokerage-based approach for cloud service selection. In *2012 IEEE Fifth International Conference on Cloud Computing*, pages 558–565.

Wagle, S., Guzek, M., Bouvry, P., and Bisdorff, R. (2015). An evaluation model for selecting cloud services from commercially available cloud providers. In *7th International Conference on Cloud Computing Technology and Science*, pages 107–114.

Zhang, Q., Cheng, L., and Boutaba, R. (2010). Cloud computing: state-of-the-art and research challenges. *Journal of Internet Services and Applications*, 1:7–18.

Zhou, M., Zhang, R., Zeng, D., and Qian, W. (2010). Services in the cloud computing era: A survey. *4th International Universal Communication Symposium (IUCS 2010)*, pages 40–46.