Title
DBaaS-Expert: A Recommender for the Selection of the Right Cloud Database

ISBN
9783319083254

Authors
Sahri, Soror
Moussa, Rim
Long, Darrell DE
et al.

Publication Date
2014

DOI
10.1007/978-3-319-08326-1_32

DBaaS-Expert: A Recommender for the Selection of the Right Cloud Database

Conference Paper · June 2014

DOI: 10.1007/978-3-319-08326-1_32

4 authors:

Sahri Soror

Paris Descartes, CPSC

Rim Moussa

University of Carthage

Darrell D. E. Long

University of California, Santa Cruz

Salima Benbernou

Paris Descartes, CPSC

DBaaS-Expert: A Recommender for the Selection of the Right Cloud Database

Soror Sahri[1], Rim Moussa[2], Darrell D.E. Long[3], and Salima Benbernou[1]

[1] Université Paris Sorbonnes Cité, Université Paris Descartes, France
{soror.sahri,salima.benbernou}@parisdescartes.fr
[2] LaTICE Laboratory, University of Tunis, Tunisia
rim.moussa@esti.rnu.tn
[3] Storage Systems Research Center, University of California, Santa Cruz, USA
darrell@cs.ucsc.edu

Abstract. The most important benefit of Cloud Computing is that organizations no longer need to expend capital up-front for hardware and software purchases. Indeed, all services are provided on a pay-per-use basis. The cloud services market is forecast to grow, and numerous providers offer database as a service (DBaaS). Nevertheless, as the number of DBaaS' offerings increases, it becomes difficult to compare various offerings through checking of a documentation ads-oriented. In this paper, we propose and describe *DBaaS-Expert* – a framework which helps a user to choose the right DBaaS Cloud Provider among DBaaS' offerings. The core components of *DBaaS-Expert* is first an ontology which captures cloud data management systems services concepts, and second a ranking core which scores each DBaaS offer in terms of criteria.

1 Introduction

Cloud computing has emerged as a new paradigm, which allows *enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources that can be rapidly provisioned and released with minimal management effort or service provider interaction* [1]. Cloud providers typically publish their service description, pricing policies and Service-Level-Agreement (SLA) rules on their websites in various formats. A data management system is one of the applications that are deployed in the cloud, the service is denoted as database-as-a-service (DBaaS). Nevertheless, as the number of DBaaS offerings increases, with different cost plans and different services, it becomes necessary to be able to automate the ranking of DBaaS offerings along a company needs. Therefore, a company should have detailed knowledge of the offerings of cloud providers that can meet its operational needs. This is not obvious, since expertise in cloud computing is required but is lacking.

We propose *DBaaS-Expert*-a framework addressing the selection of the most suitable cloud-based data management system. In order to develop this framework, first we conducted a thorough DBaaS offerings review. Second, we propose a list of dimensions, which describe DBaaS offerings and an ontology for

T. Andreasen et al. (Eds.): ISMIS 2014, LNAI 8502, pp. 315–324, 2014.

DBaaS. Third, we compute the ranking values of database service candidates based on user requirements. In our work, we perform the ranking following a known method of multi-criteria decision-making (MCDM): *Analytic Hierarchy Process (AHP)*. To the best of our knowledge, there is no existing work which addressed the problem of selection of the right DBaaS offer. The remainder of the paper is organized as follows. A review of related work is provided in Section 2. In Section 3, *DBaaS framework* is presented. Section 4 presents *DBaaS ontology*. Section 5 presents the ranking core based on *AHP*. In Section 6, a use case is devised for validating our framework. Finally, we conclude the paper.

2 Related Work

The existence of multiple options and features of cloud services, makes the selection of the appropriate cloud provider very difficult and challenging. Several works have proposed to automate cloud service selection for IaaS and PaaS models. Next, we overview related work and highlight our contribution.

Some reviewed papers are based on benchmarking [2,3]. Most of them propose revolving well known TPC benchmarks into benchmarks for data management systems' assessment in the cloud. Curino et al. [4] propose *OLTP-Bench*, an open-source framework for benchmarking on-line transaction processing (OLTP) and web workloads. Other reviewed papers [5–7] adopted a different approach based on the proposal of a meta-data model for the description of Cloud Service Providers (CSP) offerings. Among ontology-based papers one cite Zhang et al. [7]. They implemented *CloudRecommender*-a system for infrastructure services (IaaS) selection. They propose a Cloud Computing Ontology to facilitate the discovery of IaaS services categorized into functional services and QoS data. Variability modeling is used to understand and define commonalities and variabilities in software product lines and to support product derivation. For instance, Wittern et al. [6] adopt feature modeling to capture aspects and configurations of cloud services. Quinton et al. [5] address services selection from multiple CSPs using Hybrid Modeling. Indeed, within a multi-cloud configuration, a $CSP\ A$ is selected for hosting the database, while a $CSP\ B$ is selected for hosting the application. They use feature models to describe cloud systems configurations. For specific description format, Arkaitz et al. [8], have defined an XML schema that guides the description of the different capabilities of cloud storage systems such Amazon, Azure.

Multi-Criteria Decision Making methods take into consideration multiple conflicting criteria (p.e., cost vs. quality) that need to be evaluated in making decisions. Menzel et al. [9] propose *CloudGenius framework* that provides a multi-criteria approach in decision support, namely AHP technique, to automate the selection process focusing on IaaS models. Garg et al. [10] provide a *SMICLOUD framework* measuring the quality of CSPs based on QoS attributes proposed by Cloud Service Measurement Index Consortium (CSMIC) [11] and ranking the cloud services according to these attributes. *SMICloud* considers only quantitative attributes (such that response time, cost) in the context of IaaS Clouds.

Previous overviewed research work addressed only IaaS and PaaS cloud models. In this paper, we propose *DBaaS-Expert* a framework for scoring and ranking DBaaS offers in terms of criteria. For this purpose, we first propose an ontology allowing a full description of any available DBaaS offer made by a Cloud Service Provider. Second, we propose a ranking core based on a well admitted mathematical method for multi-criteria decision-making, which is AHP.

3 DBaaS-Expert Framework

In this section, we first formulate DBaaSs' offerings ranking as a Multi-Criteria Decision Making (MCDM) problem. Then, we propose a framework and a system architecture for solutioning the problem.

3.1 DBaaSs' Offerings Ranking Problem Statement

The typical MCDM problem deals with the evaluation of a *set of alternatives* in terms of a *set of decision criteria*; where *alternatives* represent the different choices available to the decision maker and *criteria* (or attributes) represent the different dimensions from which the *alternatives* can be viewed. *Criteria* are rarely of equal importance, therefore *criteria* will be weighted in terms of their importance to the decision maker. When a suitable process is applied to the problem, a rating of the alternatives can be formed into a rank. The *DBaaSs' Offerings Ranking Problem* is an MCDM problem. Indeed, *first* the set of M DBaaSs' offerings, denoted as $DBaaS_1$, $DBaaS_2$, $\ldots DBaaS_M$ (e.g., Amazon RDS, Google BigQuery) map to alternatives. *Second*, DBaaSs' offerings are characterized by a set of N decision criteria C_1, C_2, $\ldots$, C_N (e.g., Performance, high-availability capacity, elasticity, security, and so on). The objective of any solution addressing the problem is how to evaluate the set of offerings in terms of the set of criteria with two objectives (O_1) *maximize Quality and Capacity of Service* and (O_2) *minimize cost* under fully or partially satisfying a *set of user-requirements.*

3.2 Proposed Framework

DBaaS-Expert is a framework for cloud-based data management system selection. It helps to find relevant database criteria that meet users' requirements. The main component of *DBaaS-Expert* is the *DBaaS ontology.* The latter is detailed in Section 4. The main functionalities of *DBaaS-Expert* are ensured by the following modules: *application* module, *mapping* module and *ranking* module. These modules are part of the logical architecture of *DBaaS-Expert framework* depicted in Figure 1:

- *Application Module.* Users communicate with *DBaaS-Expert* via the application module. This module allows users to enter their business requirements. In order to enable the matching between user requirements and the ontology, the application module presents the DBaaS dimensions (as defined in

the ontology). Consequently, for each dimension, users can choose among its possible concepts or individuals with respect to the ontology.
- *Selection Module*. This module first maps the entered user requirements to concepts present in the DBaaS ontology, and then according to the mapping, it selects DBaaS offers which satisfy the user business requirements.
- *Ranking Module*. The obtained DBaaSs offers from the selection module are ranked. We perform the ranking following a well known mathematical method of multi-criteria decision-making: *Analytic Hierarchy Process (AHP)*.

Notice that the input of the *ranking module* is already filtered in the *selection module*. That is the application of AHP method is adapted to the selection from DBaaS ontology. Consequently, the ranking is done dynamically according to the user query. At the best of our knowledge, our work is the first to combine the use of ontology and an MCDM method for selection purpose.

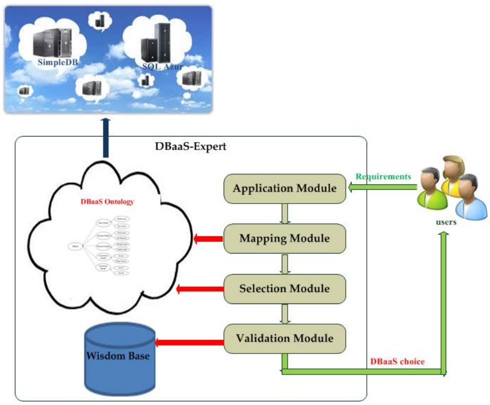

Fig. 1. Logical Architecture of *DBaaS-Expert* Framework

4 DBaaS Ontology

We propose an ontology derived from our review of most known DBaaS providers and their offerings. To the best of our knowledge, this work is the first to report a *DBaaS ontology*. Due to the lack of space, we only describe the relevant concepts of *DBaaS ontology*. Concepts are divided into four categories which relate to (i) basic concepts, (ii) quality of service concepts, (iii) capacity of service concepts and (iv) cost of service concepts. More details about the ontology concepts and its description using *Protégé ontology editor* are in [12].

4.1 Basic Concepts

- *DBaaS Offer*: Examples of DBaaS offer individuals are: Herokupostgres [13], Oracle DBaaS [14], Amazon SimpleDB [15], Google BigQuery [16].
- *Cloud Service Provider*: Each *DBaaS offer* admits a unique CSP. Examples of CSPs individuals are: Amazon, Google, Micorsoft.

4.2 General Concepts

General concepts describe each DBaaS offer.

- *Workload Type*: Workload type is*On-Line Transaction Processing* (OLTP), or *On-Line Analytical Processing* (OLAP).
- *Storage Model*: It is either traditional Hard Disk Drives (HDDs) systems, In-memory data systems or Solid State Devices (SSDs) systems.
- *Data Model*: Numerous systems exist such as Relational stores, Key-value stores, Document-oriented stores, Graph DBMS, etc.
- *Consistency Model*: Consistency guarantees affect latency and system response to concurrent read and write requests. A non-exhaustive list of consistency models are causal, eventual, strict and weak consistency models.
- *System Constraints*: Constraints are related to data volume handled, cluster size, ease of manageability as scripts running tools, etc.
- *Resource*: Resources are specific hardware configurations that the customer chooses for running its workload, such as virtual machine instance with CPU and RAM characteristics or a network with a bandwidth characteristic.
- *Trial Version*: some DBaaSs are available for free trial during a period of time.

4.3 DBaaS -Quality of Service Dimensions

- *Service Level Agreement*: SLAs capture the agreed upon guarantees between a service provider and its customer. They define the characteristics of the provided service including service level objectives, as maximum response times, minimum throughput rates and data consistency, and define penalties if these objectives are not met by the service provider.
- *Client Support*: Ideally, every customer receives $24 \times 7 \times 365$ support from the Cloud Service Provider.

4.4 DBaaS -Capacity of Service Dimensions

- *High-Availability*: In order to overcome hardware failures, data storage systems implement redundancy through replication, erasure-resilient codes or both. Redundant data is refreshed either synchronously or asynchronously. Also, in order to overcome a whole data center outage, some cloud service providers afford data distribution across different geographical zones.
- *Security*: To ensure confidentiality of sensitive data in the cloud, it is important that the data be encrypted.

- *Elasticity*: Elasticity is the ability to scale-up (provision new nodes) and scale-down (release nodes) a data storage system when the underlying application demands it.
- *Scalability*: Scalability is the ability of a system, to increase total throughput under an increased load when hardware resources are added.
- *Interoperability and Portability*: For long viability, the company should be able to easily migrate to another CSP, and gets its data back in a standard format. Hence, cloud providers and customers must consider all legal issues.

4.5 DBaaS -Cost of Service Dimensions

- *Cost Model*: Even though, many services look similar from the outside, CSPs have different pricing models for storage, CPU, bandwidth, and services for DBaaSs. Indeed, (i) *service cost* is either usage-based or subscription-fee based; (ii) for *bandwidth cost*, most CSPs provide data transfer to their data centers at no cost, while data download is priced; (iii) for *CPU cost*, there are two types of providers charging. The first is *instance-based* for which the CSP charges the customer for the number of allocated instances and how long each instance is used. This is regardless of whether the instances are fully utilized or under utilized. The second is *CPU cycles-based*, for which the CSP charges the customer for the number of CPU cycles a customers application consumes; and finally (iv) the *storage cost* is either *block-rate pricing* or *bundling pricing*. CSPs adopt *storage block-rate pricing* where the range of consumption is subdivided into subranges and the unit price is held constant over each subrange. Other providers adopt instead a *bundling pricing* mode, also called quantity discount.

5 DBaaS-Expert Ranking Core

In order to solve the *DBaaSs' Offerings Ranking Problem*, our approach uses *Analytic Hierarchy Process* (AHP) [17] for DBaaS rating and scoring. In this Section, we describe the application of AHP for solutioning *DBaaSs' Offerings Ranking Problem*.

5.1 AHP-Based Solution for DBaaSs' Offerings Ranking Problem

The main stages of the ranking process are (i) devise the AHP tree, (ii) depict criteria weights, (iii) assess selected offers of DBaaSs satisfying user requirements, and finally (iv) compute the score of each offer.

AHP Tree for DBaaS Offerings' Scoring Problem. Figure 2 illustrates the AHP tree proposed for DBaaSs' scoring and ranking along a set of criteria. The set of proposed criteria derives directly from the *DBaaS Ontology* proposed in section 4. The apex of the DBaaS hierarchy consists of three main criteria:

best quality service, best capacity of service and most affordable service. Each criterion parents criteria from the second layer. For instance, in Figure 2, the *quality of service (QoS) criterion* parents *client support*, *SLA fullfilment* and *dispute resolution* sub-criteria.

Depiction of Relative Importance of Criteria for DBaaS Selection. First of all, we assume that criteria weights are specified by the user of *DBaaS-Expert*. The weights of importance of the criteria are determined using *pairwise comparisons*. Weights are chosen in a scale of 1 to 9, as recommended by Saaty [17]. For instance, *Criterion* C_j is between to be classified as *equally important* (corresponding value is 1) and *moderate more important* (corresponding value is 3) than *Criterion* C_i. Thus, the corresponding comparison assumes the value of 2. A similar interpretation is true for the rest of the entries of Table 1-(a). The size of the matrix is N^2 (9 for $N = 3$ where N is the number of criteria). The diagonal values are unity (i.e., $criteria\_weights[i, i] = 1$) and the matrix is such that cells in the upper diagonal and cells in the lower diagonal are in inverse relationship (i.e., $criteria\_weights[i, j] = \frac{1}{criteria\_weights[i,j]}$). Notice that, given N criteria, $\frac{N\times(N-1)}{2}$ comparisons are performed. The next step is to compute the *eigenvector* of the *squared criteria' weights matrix* in order to extract the relative importances implied by the previous comparisons. Given a judgement matrix with pairwise comparisons, the corresponding *eigenvector* is obtained by *first* raising the pairwise matrix to powers that are successively squared each time, *second* calculating sums over rows and normalizing values by dividing sum over rows by the sum over column (i.e., geometric mean of each row). This process is iterated until the *eigenvector* does not change from previous iteration (i.e., Δ is negligible). Since, criteria are organized in a hierarchical way, there are two types of weights, namely *local weights* and *global weights*. *Local weights* correspond to weights of criteria of same level, and are calculated as demonstrated in Table 1. The Global weight of a sub-criterion $sub - C_j$ is equal to the product of its local weight and the global weight of C_j.

Table 1 demonstrates pairwise-comparisons of criteria. First, the user enters $weight\_of\_C_i$ compared to $weight\_of\_C_j$ (i.e., half important), $weight\_of\_C_i$ compared to $weight\_of\_C_k$ (i.e., three times more important), and $weight\_of\_C_j$ compared to $weight\_of\_C_k$ (i.e., four times more important). In the example, two iterations are performed. At each iteration, the *squared matrix*, *eigen vector*, and Δ*eigenvectors* are calculated. The example shows that *Criterion* C_i is the second most important criterion ($W_i = 0.3196$), *Criterion* C_j is the most important criterion ($W_j = 0.5584$) and *Criterion* C_k is the least important criterion.

Assessment of DBaaS' Offerings. First of all, the assessment of DBaaSs' offerings in terms of criteria is performed by experts. Assessment of DBaaSs' offerings in terms of criteria may be done by two ways, (i) *pairwise comparisons* as for criteria: assessments are chosen in a scale of 1 to 9; or (ii) direct assignment through metering of each criterion. The latter is challenging. Indeed, criteria have different types of data (cost: \$, transaction throughput: Tps, feature: bool) and

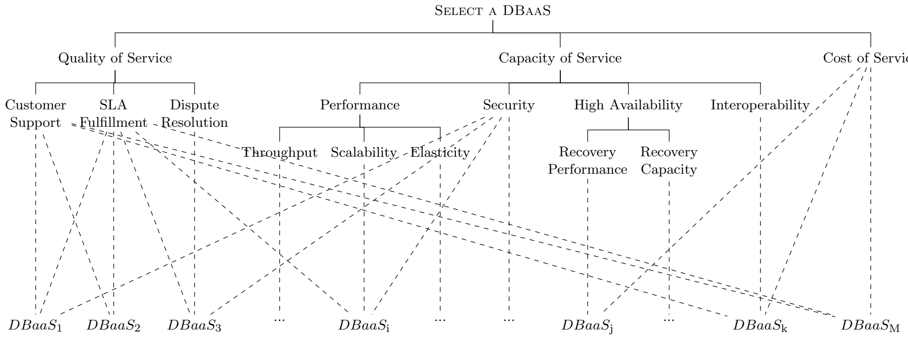

Fig. 2. AHP Tree for DBaaS Offerings' Ranking Problem

Table 1. Example of Transformation of the Matrix *Weights of Criteria* into a *Priority Vector*

	C_i	C_j	C_k
C_i	1	$\frac{1}{2}$	3
C_j	2	1	4
C_k	$\frac{1}{3}$	$\frac{1}{4}$	1

(a) 1^{st} Matrix.

	C_i	C_j	C_k
C_i	3.0000	1.7500	8.0000
C_j	5.3332	3.0000	14.0000
C_k	1.1666	0.6667	3.0000

(b) 1^{st} Squared Matrix.

	C_i	C_j	C_k	$\sum$	*eigenvector*
C_i	3.0000	1.7500	8.0000	12.7500	0.3194
C_j	5.3332	3.0000	14.0000	22.3132	0.5595
C_k	1.1666	0.6667	3.0000	4.8333	0.1211
			$\sum$	*39.9155*	*1.0000*

(c) 1^{st} eigenvector.

	C_i	C_j	C_k
C_i	27.6653	15.8330	72.4984
C_j	48.3311	27.6662	126.6642
C_k	10.5547	6.0414	27.6653

(d) 2^{nd} Squared Matrix.

	C_i	C_j	C_k	$\sum$	*eigenvector*	Δ
C_i	27.6653	15.8330	72.4984	115.9967	0.3196	-0.0002
C_j	48.3311	27.6662	126.6642	202.6615	0.5584	0.0011
C_k	10.5547	6.0414	27.6653	44.2614	0.1220	-0.0009
			$\sum$	*362.9196*	*1.0000*	

(e) 2^{nd} eigenvector.

Table 2. Example of Assessment of 3 DBaaSs along 2 criteria in order to obtain columns i and j of the *Decision Matrix*

	$DBaaS_x$	$DBaaS_y$	$DBaaS_z$	*eigenvector*
$DBaaS_x$	1	6	6	0.75
$DBaaS_y$	$\frac{1}{6}$	1	1	0.13
$DBaaS_z$	$\frac{1}{6}$	$\frac{1}{1}$	1	0.13

(a) DBaaSs' assessment along $Criterion_i$.

	$DBaaS_x$	$DBaaS_y$	$DBaaS_z$	*eigenvector*
$DBaaS_x$	1	1	9	0.48
$DBaaS_y$	$\frac{1}{1}$	1	8	0.46
$DBaaS_z$	$\frac{1}{9}$	$\frac{1}{8}$	1	0.06

(b) DBaaSs' assessment along $Criterion_j$.

different considerations (cost: lower is better, throughput: greater is better, ...). For the sake of simplicity, DBaaS offers are assessed in terms of criteria along pairwise comparisons. Table 2 illustrates DBaaS offerings assessment along two criteria through pairwise comparisons.

Table 3. Decision Matrix

		Criteria					
		C_1	C_2	...	C_i	...	C_M
		W_1	W_2	...	W_i	...	W_M
	$DBaaS_1$	$a_{1,1}$	$a_{1,2}$	...	$a_{1,i}$	...	$a_{1,M}$
	$DBaaS_2$	$a_{2,1}$	$a_{2,2}$	...	$a_{2,i}$	...	$a_{2,M}$
Alternatives	$DBaaS_3$	$a_{3,1}$	$a_{3,2}$	...	$a_{3,i}$	...	$a_{3,M}$
	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
	$DBaaS_N$	$a_{N,1}$	$a_{N,2}$	...	$a_{N,i}$	...	$a_{N,M}$

DBaaS Score Calculus. After the alternatives are compared with each other in terms of each one of the decision criteria and the individual *priority vectors* are derived; the priority vectors become the columns of the decision matrix as shown in Table 3, and the score of each $DBaaS_i$ is calculated as follows, $Score(DBaaS_i) = \sum_{j=1}^{M} a_{i,j} W_j$

6 Use Case

Hereafter, we describe the outline of typical use case of *DBaaS-Expert*.

- *(Step 1)* The user enters business requirements via the application module of *DBaaS-Expert*. For instance, he may choose *OLAP* as workload type and *Relational store* as a data model.
- *(Step 2) DBaaS-Expert* maps the user requirements to concepts from the DBaaS ontology and filters all relevant DBaaS offers. The outcome of this step is a list of offers of DBaaSs satisfying the user business requirements.
- *(Step 3)* corresponds to AHP run,
  - *(Step 3-a) DBaaS-Expert* builds a tree, with a subset of DBaaSs selected from *Step 2*. Notice that, all other alternatives are discarded from selection,
  - *(Step 3-b) DBaaS-Expert* allows the user to perform paiwise comparisons of the different criteria, in order to obtain a *priority vector* for criteria.
  - *(Step 3-c) DBaaS-Expert* uses its expert knowledge materialized in the wisdom base for the assessment of the selected DBaaS offerings among criteria, and builds the *Decision Matrix*,
  - *(Step 3-d) DBaaS Expert* rates and ranks the offers and returns the result to the user,

7 Conclusion

In this paper, we propose DBaaS-Expert framework, which allows a user to choose the most suitable DBaaS. Our contribution is two fold: First, we propose a *DBaaS ontology*. Second, ranking DBaaS offers using a well known AHP - mathematical method for multi-criteria decision-making. In the future, we plan to continue the current work research to include DBaaS assessment using user feedbacks and past experiences.

References

1. Mell, P., Grance, T.: The NIST definition of cloud computing (2011), http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf
2. Binnig, C., Kossmann, D., Kraska, T., Loesing, S.: How is the weather tomorrow?: Towards a benchmark for the cloud. In: DBTest (2009)
3. Kossmann, D., Kraska, T., Loesing, S.: An evaluation of alternative architectures for transaction processing in the cloud. In: SIGMOD Conference, pp. 579–590 (2010)
4. Carlo, C., Djellel, D., Andrew, P., Philippe, C.M.: Benchmarking oltp/web databases in the cloud: The oltp-bench framework. In: Proc. of 4th CloudDB, pp. 17–20 (2012)
5. Quinton, C., Haderer, N., Rouvoy, R., Duchien, L.: Towards multi-cloud configurations using feature models and ontologies. In: Proc. of the Intl. Workshop on Multi-Cloud Apps and Federated Clouds, pp. 21–26 (2013)
6. Wittern, E., Kuhlenkamp, J., Menzel, M.: Cloud service selection based on variability modeling. In: Liu, C., Ludwig, H., Toumani, F., Yu, Q. (eds.) Service Oriented Computing. LNCS, vol. 7636, pp. 127–141. Springer, Heidelberg (2012)
7. Zhang, M., Ranjan, R., Nepal, S., Menzel, M., Haller, A.: A declarative recommender system for cloud infrastructure services selection. In: Vanmechelen, K., Altmann, J., Rana, O.F. (eds.) GECON 2012. LNCS, vol. 7714, pp. 102–113. Springer, Heidelberg (2012)
8. Ruiz-Alvarez, A., Humphrey, M.: An automated approach to cloud storage service selection. In: Proc. of 2nd ScienceCloud Workshop, pp. 39–48 (2011)
9. Menzel, M., Ranjan, R.: Cloudgenius: Decision support for web server cloud migration. In: Proc. of 21st WWW, pp. 979–988 (2012)
10. Garg, S.K., Versteeg, S., Buyya, R.: Smicloud: A framework for comparing and ranking cloud services. In: Proc. of 4th IEEE UCC, pp. 210–218 (2011)
11. CSMIC: Cloud service measurement index consortium, http://www.cloudcommons.com/fr/about-smi
12. DBaaS-expert: Technical report (2014), https://sites.google.com/site/rimmoussa/investig_dbaas
13. Heroku: Herokupostgres: Sql database-as-a-service (2013), https://postgres.heroku.com/
14. Oracle: Database as a service: Reference architecture: An overview (2011), http://www.oracle.com/technetwork/topics/entarch/oes-refarch-dbaas-508111.pdf
15. Amazon: SimpleDB (2013), http://aws.amazon.com/documentation/simpledb/
16. Sato, K.: An inside look at google bigquery (2013), https://cloud.google.com/files/BigQueryTechnicalWP.pdf
17. Saaty, T.: The Analytic Hierarchy Process: Planning, Priority Setting, Resource Allocation. McGraw-Hill (1980)