Supporting the Migration of Applications to the Cloud through a Decision Support System

Vasilios Andrikopoulos, Zhe Song, Frank Leymann

Institute of Architecture of Application Systems,
University of Stuttgart, Germany
{firstname.lastname}@iaas.uni-stuttgart.de

BibTeX:

```
@inproceedings{Andrikopoulos2013supporting,
  author    = {Vasilios Andrikopoulos, Zhe Song, and Frank Leymann},
  title     = {Supporting the Migration of Applications to the Cloud through a
Decision Support System},
  booktitle = {Proceedings of the 6th IEEE International Conference on Cloud
Computing (CLOUD 2013), June 27-July 2, 2013, Santa Clara Marriott, CA, USA},
  year      = {2013},
  pages     = {565-572},
  doi       = {10.1109/CLOUD.2013.128},
  publisher = {IEEE Computer Society}
}
```

2013 IEEE Sixth International Conference on Cloud Computing

Supporting the Migration of Applications to the Cloud through a Decision Support System

Vasilios Andrikopoulos, Zhe Song and Frank Leymann
Institute of Architecture of Application Systems (IAAS)
University of Stuttgart, Stuttgart, Germany
lastname@iaas.uni-stuttgart.de

***Abstract*—The motivation for this work is the necessity to be able to select an appropriate Cloud service provider offering for the migration of existing applications, based on cost minimization. While service providers offer pricing information publicly, and online tools allow for the calculation of cost for various Cloud offerings, the selection of which offering fits better the application requirements is left to application developers. For this purpose, this work proposes a migration decision support system that incorporates both offering matching and cost calculation, combining features from various approaches in the State of the Art. The proposed approach is then evaluated against existing tools.**

***Keywords*-Cloud migration; decision support; provider selection; cost calculation; usage patterns**

I. Introduction

The increasing domination of Cloud computing solutions in the software market means that existing applications may need to migrate to this environment, in order to reap the benefits of reduced infrastructural costs and dynamic access to computational resources. While applications have already started being developed specifically for the Cloud (forming Cloud-native applications), existing systems must be adapted to be suitable for the Cloud, requiring to make them Cloud-enabled [1]. Decision making on whether and how to migrate to the Cloud entails a series of parameters [2], and incorporates multiple dimensions with different analytical tasks that need to be supported [3]. Supporting this process in the literature mainly revolves around selecting an appropriate provider and calculating the cost of deploying and running the application, usually as part of an Infrastructure as a Service (IaaS) solution, see for example [2], [4], [5], and [6].

Calculating the cost of using Cloud offerings and comparing them with each other, despite the availability of information from the providers (e.g., [7]), is complicated by a number of factors. First of all, different Cloud providers use different pricing models. As discussed in [8], a Cloud service may be offered on a pay per use, subscription, prepaid per use (pay per use against a pre-paid credit), or combinations of subscription and pay per use, with dedicated computing resources rented for a period of time, and additional resources available on demand. More sophisticated options are also available. Amazon Web Services for example allow customers to bid for unused EC2 capacity by means of Spot Instances [9].

Furthermore, different providers define the parameters of their offerings in different ways. Windows Azure [7], for example, incorporates the cost of I/O bandwidth as part of the pricing for Cloud Service instances, offering Virtual Machine (VM) images for lease, while Amazon EC2 charges only for outbound data [9]. The same service is also typically offered in a variety of configurations on, e.g., the number of CPU cores, or the maximum storage size required, that further complicates the selection of an appropriate service provider. The related work in migration support of applications to the Cloud, e.g., [5], [6], [10], identify for this purpose a fixed set of parameters to be considered in calculating the cost of Cloud offerings. These parameters however do not necessarily cover all configuration options available by service providers.

In this work we focus on designing and developing a migration decision support system that addresses the issues of offering selection and cost calculation. In particular:

- we present a set of requirements for supporting the decision making in migrating applications to the Cloud;
- we propose a three-tiered architecture for a decision support system based on these requirements that incorporates both offering selection and cost calculation capabilities; and,
- we discuss a prototypical implementation of the proposed approach which we evaluate in practice.

The rest of this paper is structured as follows: Section II outlines our proposal for a Migration Decision Support System by describing the requirements to be satisfied and its main architectural components. Section III discusses the implementation of a prototype of our proposal and illustrates the main interaction points with the system. Section IV provides an evaluation of the prototype using publicly available tools as a point of reference. Section V summarizes related work, and Section VI concludes this paper.

978-0-7695-5028-2/13 $26.00 © 2013 IEEE
DOI 10.1109/CLOUD.2013.128

IEEE computer society

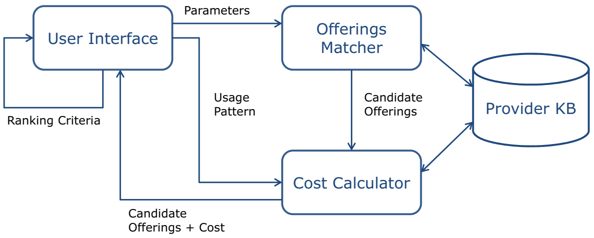


Figure 1: Overview of MDSS

II. Migration Decision Support System (MDSS)

A. Overview

Our investigation of the existing literature and available tools allowed us to extract the following requirements for a decision support system geared towards the migration of applications to the Cloud:

1) Ability to match user-provided requirements with available Cloud service provider offerings and calculate the cost of using each offering for a given period.
2) Ability to rank the proposed offerings based on different criteria beyond cost.
3) Ability to support variable requirements in terms of computational resources over periods of time to better match the varying demands of the users.
4) Existence of a knowledge base for Cloud service providers containing the information regarding their offerings and pricing models.
5) Availability of a user-friendly interface that allows user to navigate the system easily.

Figure 1 provides a high-level view of the proposed architecture for a *Migration Decision Support System (MDSS)* that addresses these requirements. In particular, a three-tiered architectural model is adopted with:

1) a *User Interface* as the front-end of the system to the users,
2) an *Offerings Matcher* and a *Cost Calculator* implementing the main functions of the system: selecting candidate solutions based on the user requirements, and calculating their costs for given periods and foreseen usage patterns, respectively, and
3) a *Provider Knowledge Base (KB)* which holds collected information regarding the offered solutions by Cloud service providers, and their pricing policies.

Each of these architectural components is discussed in detail in the following.

B. User Interface

The User Interface component allows users to interact with the system, express their requirements with respect to the desired characteristics of a Cloud service offering, retrieve available offerings and their calculated costs, and define criteria for dynamically ranking them. Initially, user requirements are expressed as a set of *Parameters* and a set of *Usage Patterns* that is foreseen by the user.

Table I: Variants of Usage Patterns

Variant	**Possible values**
Type	GB, Hours, Number of Transactions, ...
Trend	Increase, Decrease, Invariant
Period	Month
Rate	Percentage

Parameters can be either *numerical* (e.g., number of CPUs in a VM) or *non-numerical* (e.g., required licenses). The parameters to be defined by the user depend on the type of service required by the user: data storage services for example are usually defined in terms of space and incoming/outgoing traffic, while VM provisioning services require a longer list of parameters (number and frequency of CPUs, size of memory, size of disk, etc.).

Usage patterns allow for dynamically changing the computational needs during the foreseen period of consuming the Cloud service. For example, for a data storage service, a Usage Pattern can be defined to express the need for increasing or decreasing the required storage size at a given rate for a given period, e.g., 10% increase per month for the next 6 months. Usage Patterns can be chained together to create more complicated patterns. A periodic increase in required storage space around the holiday period, for example, can be expressed as an invariant trend over 10 months, with a rapidly increasing and decreasing period of 2 months. The system offers the option to the user to define usage patterns in terms of *Variants*, as summarized by Table I.

The User Interface relies on the Offerings Matcher to identify candidate Cloud service offerings that match the provided parameters, and on the Cost Calculator for providing a cost estimate for each of them given the defined usage pattern. The output from these two components is visualized and offered to the user for further input. In order to rank the candidate offerings, Analytical Hierarchy Process (AHP)-based approaches like CloudGenius [6] and SMICloud [10] require the user to provide a weighting scheme for the given parameters that they then use to order their solutions. MDSS focuses instead on allowing users to define the criteria to dynamically rank the results based on cost, or on any of the provided parameters. In this manner, the capacity for multi-criteria optimization as supported by AHP-based approaches is traded for less input required from the user, resulting in ease of use.

C. Provider Knowledge Base

A knowledge base (KB) is the basis for decision support systems. In the case of MDSS, the *Provider KB* con-

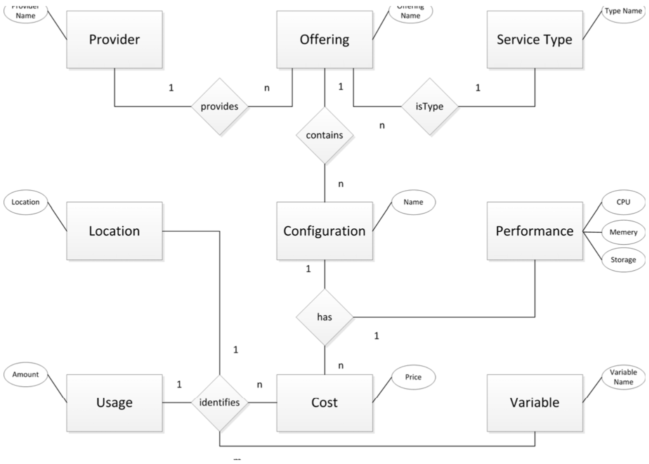
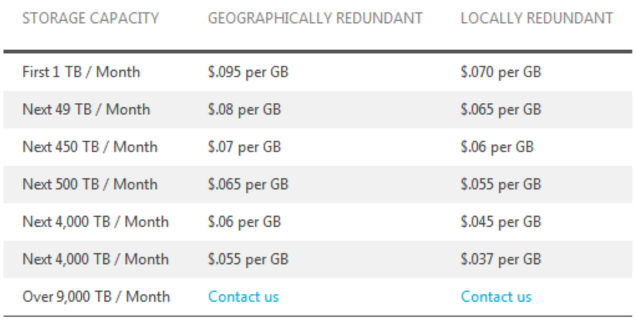


Figure 2: ER diagram of the Provider Knowledge Base Data Model



STORAGE CAPACITY	GEOGRAPHICALLY REDUNDANT	LOCALLY REDUNDANT
First 1 TB / Month	$.095 per GB	$.070 per GB
Next 49 TB / Month	$.08 per GB	$.065 per GB
Next 450 TB / Month	$.07 per GB	$.06 per GB
Next 500 TB / Month	$.065 per GB	$.055 per GB
Next 4,000 TB / Month	$.06 per GB	$.045 per GB
Next 4,000 TB / Month	$.055 per GB	$.037 per GB
Over 9,000 TB / Month	Contact us	Contact us

Figure 3: Pricing details for the Windows Azure Storage service [7]

tains information concerning the offerings of Cloud service providers and the associated pricing policies for each of them. Figure 2 summarizes the data model of the Provider KB using the ER notation. More specifically, each *Provider* like Google, Amazon or Microsoft provides a set of *Offering*s of different *Service Type*s. Amazon for example offers, among others, EC2 (an infrastructure, VM on demand service) and S3 (a data storage service), Google offers AppEngine (a PaaS solution) and so on. Each such Offering is offered with different *Configuration* options. The values for these options are reflected by the *Performance* entity as characteristics, e.g., 1, 2, 4 or 8 CPUs.

A different *Cost* function is associated with each Configuration over a number of *Variables* like the number of hours per day that the service is used. Periods of *Usage* with different pricing policies due to discounting for block usage are related to each function. The Cost functions also take into account the *Location* of the service, since different pricing policies usually apply for different regions. Defining the Cost function for each Configuration is a farely straightforward procedure given the fact that most Cloud service providers offer extensive information on how each of their offerings is charged. The Windows Azure Storage service pricing policy [7], for example, for the geographically redundant option, and with the defined cost of $0.01 per 100K transactions, can be formulated as follows:

$$f(x,y) = \begin{cases} (0.095 \times \frac{x}{1000}) + \frac{(0.01 \times y)}{100000} & x \in [0,1] \\ (95 + (0.8 \times \frac{x-1}{1000}) + \frac{(0.01 \times y)}{100000} & x \in (1,50] \\ (4015 + (0.7 \times \frac{x-50}{1000}) + \frac{(0.01 \times y)}{100000} & x \in (50,500] \\ \ldots \end{cases}$$

where x is the storage capacity in TB and y the number of storage transactions. For each case, the cost calculation includes the cost of the previous price brackets: for 3 TB of storage, for example, the cost for the first TB is $95 plus $0.8 dollars for every GB over that until 3 TB. The variables x and y are defined by the user through the provided Usage Pattern for the service as Variants. Each one of the $[0,1], (1,50], \ldots$ price brackets is stored as a different Configuration of the Windows Azure Storage Offering. Furthermore, for each variable a default minimum and maximum value is defined. If no values are provided for these variables by the user, i.e., they are not included in the defined parameters as part of the user requirements, then the minimum and maximum values are used during the cost calculation.

D. *Offerings Matcher*

The Offerings Matcher component is responsible for selecting from the Provider KB the Offerings whose Configurations satisfy the Parameters defined by the user (through the User Interface). Most parameters are measurable and numerical, e.g., number of CPU cores, or size of RAM. In this case, identifying which Performance characteristics match the provided Parameters is a matter of numerical comparison between variables of the same type. For example, a requirement for 4 CPU cores can be matched to Configurations that provide at least 4 cores. A number of critical parameters however, like the OS type, or the licenses required, are non-numerical, and matching them is reduced to string matching, e.g., if the user requires Linux instead of Windows to be offered by the service. An ontology of parameters can be added in the future to accommodate a semantics-based matching for such cases.

The algorithm traverses through the list of all Configurations for all Offerings that belong to the service type defined by the user and attempts to find Configurations with Performance Characteristics that match all the provided Parameters. Offerings with Configurations that satisfy the provided Parameters are considered as *Candidate Offerings*. In this sense, the Offerings Matcher adopts an optimistic selection policy: if no explicit constraint is expressed by the user w.r.t. some parameter then it is assumed that any offer by the provider is acceptable. Candidate Offerings are forwarded to the Cost Calculator before returned to the User Interface, in which users have the option to review and remove candidates that do not satisfy their needs.

E. Cost Calculator

The Cost Calculator component is responsible for providing an estimated cost for the usage of a Candidate Offering as defined by the Usage Pattern provided by the user. If no Usage Patterns are defined by the user then the assumption is that the desired consumption period is 1 month. Usage variants have also pre-defined minimum and maximum values stored in the KB in case none is provided by the user. In this case MDSS calculates the cost for both minimum and maximum values and returns only the lowest cost function.

The calculation for each Candidate Offering can be decomposed into the following steps:

1) Retrieve the cost functions for the Candidate Offering from the Provider KB. More than one cost functions may fit the defined period; in this case the rest of the steps are repeated for all of them, and only the lowest cost is returned at the end.
2) The variables of the cost function are identified. If no values are provided for them in one of the defined Usage Patterns, then the calculations below must be repeated both with their pre-defined minimum and maximum values.
3) Update the estimated usage based on the Variant defined in the Usage Pattern for each month; use the last known value if no further Variant is defined, or is defined as with an Invariant trend (Table I) over this period.
4) Evaluate the cost function for each month based on the estimated usage defined in the previous step and sum the results of the total consumption period. Return the estimated cost.

At the end of this procedure, each Candidate Offering is associated with an estimated Cost for the given Usage Patterns. The calculated cost is the minimum possible for this offering in the given period. This allows us to cover the cases where providers are giving discounts for larger periods of time without modifying the rest of the calculations.

The following section discusses the prototypical implementation of MDSS.

III. Implementation

The MDSS architecture discussed in the previous sections was implemented as a Web application on top of a relational database. In particular, the Provider KB was implemented in Microsoft SQL Server 2008 Express, containing 14 entities corresponding to the entities and relationships identified in Fig. 2. Data for all the Cloud offerings provided by Google and Microsoft were collected and encoded in the database, each one of which was associated with the cost functions defined in the respective pricing policy pages.

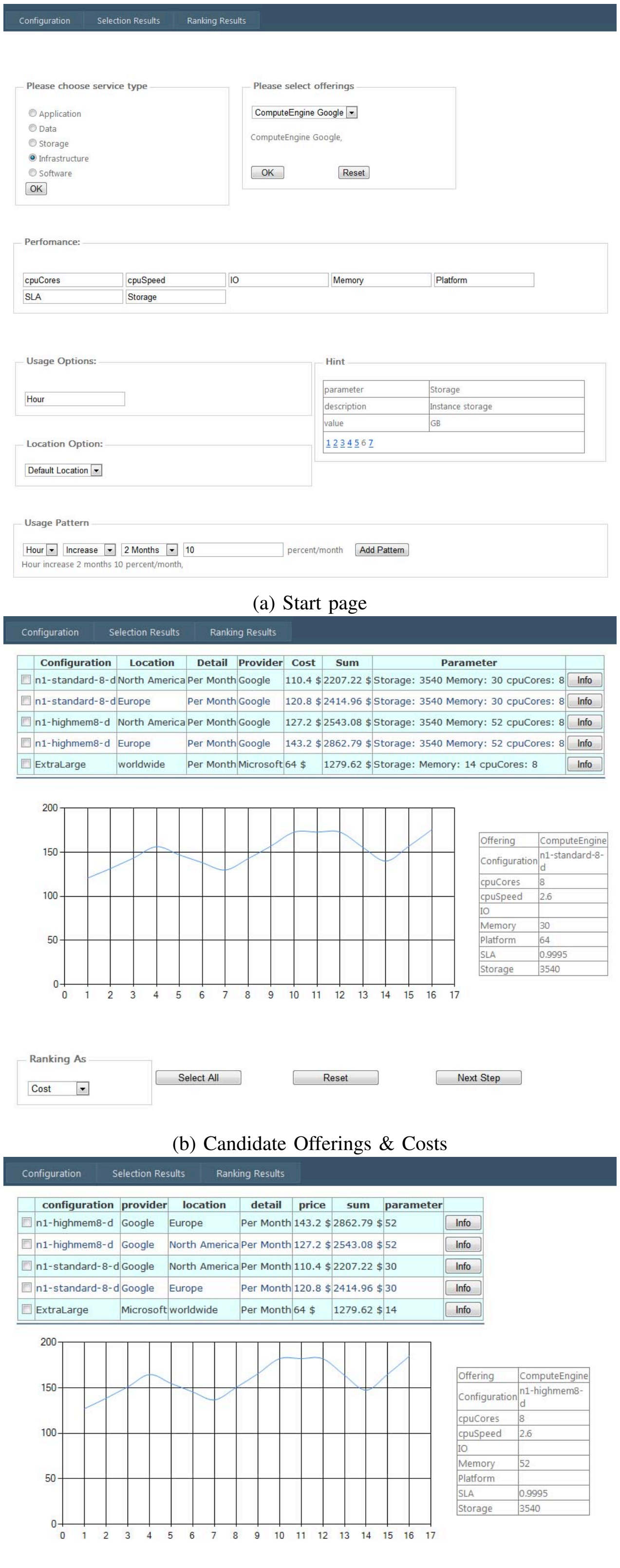
The User Interface was implemented as a set of ASP.NET Web pages, with the Offerings Matcher and Cost Calculator components implemented as the logic behind serving the pages. A combination of Microsoft Visual C# classes and SQL queries are used to realize the algorithmical steps described in the previous section. Microsoft Visual Studio 2010 based on Microsoft .NET Framework 4.0 was used for the development of the MDSS prototype.

Figure 4 provides screenshots from the three main interaction steps of the user with the MDSS prototype. Figure 4a is the main screen allowing the user to select which service type is required (e.g., VM as an infrastructure). The offerings available in the Provider KB are then dynamically loaded and the user is asked to select between one offering in particular (e.g., Google ComputeEngine) or all offerings available. Selecting one service type also loads dynamically the options for performance parameters that user can define as requirements. Help on each parameter is provided through the Hint frame on the right of the screen.

In Fig. 4a the user can define a minimum number of CPU cores for the desired VM, their minimum CPU speed, etc. Furthermore, the user can (optionally) provide values for the foreseen usage options, in this case how many hours per month would the VM be used. Usage Patterns can be defined at the bottom of the screen using the drop list and the Add Pattern button. In Fig. 4a, the user defined a 10% increase on the usage hours per month, for a period of 2 months. Further additions of patterns are interpreted as consequent changes in usage; adding for example another pattern with 20% increase for 3 months would result as total usage period of 5 months, with the first 2 months of usage (in terms of hours per month) increasing by 10%, and the next 3 by 20%. The desired location of the VM (as a region) can also be defined through the Location Options droplist.

Figure 4b shows the results of requesting a VM on Google Compute Engine or Windows Azure with minimum 8 CPU cores, at least 8 GBs RAM and 1000 GBs of disk storage for a fluctuating usage pattern across 15 months. A short description of each Candidate Offering is offered, together with the estimated total cost for this period and the configuration parameters that correspond to the requirements set by the user. Pressing the Info button for each Candidate Offering loads a table containing the complete information stored in the Provider KB for this offering, and provides an illustration of the cost calculation for this offering per month.

Candidate Offerings in this stage are returned in the order identified by the Offerings Matcher, i.e., not ranked. The next step is ranking based on a desired criterion, either calculated cost, or any of the parameters provided by the user. One or more of the Candidate Offerings can be selected for this purpose. Figure 4c, for example, shows the results



Please choose service type

○ Application
○ Data
○ Storage
● Infrastructure
○ Software

Please select offerings

ComputeEngine Google

ComputeEngine Google,

Perfomance:

cpuCores	cpuSpeed	IO	Memory	Platform
SLA	Storage			

Usage Options: Hour

Location Option: Default Location

Hint

parameter	Storage
description	Instance storage
value	GB
1 2 3 4 5 6 7	

Usage Pattern

Hour · Increase · 2 Months · 10 percent/month

Hour increase 2 months 10 percent/month,

(a) Start page

	Configuration	Location	Detail	Provider	Cost	Sum	Parameter	
□	n1-standard-8-d	North America	Per Month	Google	110.4 $	2207.22 $	Storage: 3540 Memory: 30 cpuCores: 8	Info
□	n1-standard-8-d	Europe	Per Month	Google	120.8 $	2414.96 $	Storage: 3540 Memory: 30 cpuCores: 8	Info
□	n1-highmem8-d	North America	Per Month	Google	127.2 $	2543.08 $	Storage: 3540 Memory: 52 cpuCores: 8	Info
□	n1-highmem8-d	Europe	Per Month	Google	143.2 $	2862.79 $	Storage: 3540 Memory: 52 cpuCores: 8	Info
□	ExtraLarge	worldwide	Per Month	Microsoft	64 $	1279.62 $	Storage: Memory: 14 cpuCores: 8	Info

Offering	ComputeEngine
Configuration	n1-standard-8-d
cpuCores	8
cpuSpeed	2.6
IO	
Memory	30
Platform	64
SLA	0.9995
Storage	3540

Ranking As: Cost

(b) Candidate Offerings & Costs

	configuration	provider	location	detail	price	sum	parameter	
□	n1-highmem8-d	Google	Europe	Per Month	143.2 $	2862.79 $	52	Info
□	n1-highmem8-d	Google	North America	Per Month	127.2 $	2543.08 $	52	Info
□	n1-standard-8-d	Google	North America	Per Month	110.4 $	2207.22 $	30	Info
□	n1-standard-8-d	Google	Europe	Per Month	120.8 $	2414.96 $	30	Info
□	ExtraLarge	Microsoft	worldwide	Per Month	64 $	1279.62 $	14	Info

Offering	ComputeEngine
Configuration	n1-highmem8-d
cpuCores	8
cpuSpeed	2.6
IO	
Memory	52
Platform	
SLA	0.9995
Storage	3540

(c) Ranking page

Figure 4: MDSS interface pages

of ranking the results of the selection in Fig. 4b by size of RAM. All the information from the previous screen is available, meaning that the user can access both the complete information for the offering, and the curve illustrating the cost calculation. Ranking by different criteria can be initiated by the same screen; the result of multiple rankings is currently however not cumulative. This functionality is currently under development.

We are currently in the process of collecting information from more Cloud providers to enrich the Provider KB and offer the MDSS as a service.

IV. Evaluation

For purposes of evaluating our proposal we use RightScale's PlanForCloud (PFC) solution [11] that offers similar functionalities in terms of cost calculation to the ones supported by MDSS. PFC is publicly available and allows dynamic usage definition and cost calculation across different providers, and in this sense it is ideal for comparison with our solution.

In the first stage of the evaluation we focus on the Offerings Matcher component. For this purpose we executed a series of queries for different offerings in both MDSS and PFC with the results coinciding in all cases. Table II shows the results of a request for an infrastructure solution of at least 8 CPU cores and 8 GB of RAM. The table omits offerings from providers other than Microsoft and Google, since they are currently not included in the database of MDSS. As shown in the table, both MDSS and PFC identify the same possible offerings, with MDSS differentiating between regions.

With respect to the Cost Calculator component, the evaluation is decomposed into two cases: one with a uniform, flat pricing policy on the usage of the offering, and another one with a dynamic policy in cost brackets like the one enforced by Windows Azure Storage in Fig. 3. Table III summarizes the output of MDSS, PFC and the Price Calculator offered by Microsoft (MPC) [12] for a fixed amount of 720 hours per month usage. All calculations took place during the second half of December 2012. As shown in the table, the lack of incorporating the location preference in PFC at the time of performing this evaluation results in differences between the calculated costs for Google Compute Engine (CE). With respect to Windows Azure, we can demonstrate that calculated costs by MDSS are indeed verified by the MPC calculation (for the WorldWide location option). PFC calculations differ from the output of both MDSS and MPC, probably due to differences in the basic unit-price in the knowledge base of PFC. In this respect, it can be demonstrated that contrary to PFC, the accuracy of the calculated costs by MDSS can be externally verified by MPC.

Table IV summarizes the prices calculated in a similar manner and period for using two storage solutions (Windows

Table II: Offerings Matcher Evaluation using PlanForCloud

MDSS			**PlanForCloud (PFC)**	
Google CE		Windows Azure	Google CE	Windows Azure
US	EU	Worldwide	N/A	N/A
n1-standard-8-d	n1-standard-8-d	ExtraLarge	n1-standard-8-d	ExtraLarge
n1-standard-8	n1-standard-8		n1-standard-8	
n1-highmem-8-d	n1-highmem-8-d		n1-highmem-8-d	
n1-highmem-8	n1-highmem-8		n1-highmem-8	

Table III: Cost Calculator Evaluation: Uniform Pricing

Provider	**Configuration**	**MDSS**		**PFC**	**MPC**	**Unit Price**	
		US	EU			US	EU
Google CE	n1-standard-1-d	$99.36	$108.72	$102.67		$0.138	$0.151
	n1-standard-4-d	$397.44	$434.88	$410.69		$0.552	$0.604
	n1-standard-8-d	$794.88	$869.76	$821.38		$1.104	$1.208
	n1-standard-1	$86.4	$95.04	$89.28		$0.12	$0.132
	n1-standard-4	$345.6	$380.16	$357.12		$0.48	$0.528
	n1-standard-8	$691.2	$760.32	$714.24		$0.96	$1.056
	n1-highmem-4-d	$457.92	$515.52	$473.18		$0.636	$0.716
	n1-highmem-8-d	$915.84	$1031.04	$946.37		$1.272	$1.432
	n1-highmem-4	$365.76	$411.84	$377.95		$0.508	$0.572
	n1-highmem-8	$731.52	$823.68	$755.9		$1.016	$1.144
	n1-highcpu-4-d	$244.8	$276.48	$252.96		$0.34	$0.384
	n1-highcpu-8-d	$489.6	$552.96	$505.92		$0.68	$0.768
	n1-highcpu-4	$195.84	$218.88	$202.37		$0.272	$0.304
	n1-highcpu-8	$391.68	$437.76	$404.74		$0.544	$0.608
Windows Azure	ExtraSmall	$9.36		$14.88	$9.36	$0.013	
	Small	$57.6		$63.24	$57.6	$0.08	
	Medium	$115.2		$126.48	$115.2	$0.16	
	Large	$230.4		$252.96	$230.4	$0.32	
	ExtraLarge	$460.8		$505.92	$460.8	$0.64	

Azure Storage in the Geographically Redundant configuration, and Google Cloud Storage) for different amounts of required storage space. The calculations in this case are very similar, and are again confirmed by MPC. Deviations on behalf of MDSS are due to the precision of the calculations when evaluating cost functions with many decimal points. Currently a simple formula evaluation library in JavaScript is used in MDSS for this purpose, but in the future a software library that allows better precision is to be used. Overall however, it can be seen that as far

V. Related Work

A Cloud service can be classified by many different factors that represent its overall performance and features. An example of a classification of such factors is provided by the Service Measurement Index (SMI) [10]. The SMI framework summarizes the most important QoS attributes for Cloud offerings on a high level, such like Accountability, Agility, and Assurance of service, Cost, Performance, Security, Privacy and Usability. SMI is used, among others, by the STRATOS framework [13], that focuses on provider selection based on these attributes. The actual decision process is complicated by a large number of factors and parameters defined on different levels, essentially resulting into a Multiple Criteria Decision Making (MCDM) problem [14]. Because of the structured relationship between factors and parameters an Analytic Hierarchy Process (AHP) is a proposed approach for facilitating the problem of MCDM.

A number of existing works use AHP to support the decision in selecting an appropriate Cloud service provider for a given set of requirements. In the SMICloud approach for example [10], AHP is used to compare parameters of different providers based on a value-based ranking method optimizing on cost for VM-oriented Cloud offerings. CloudGenius [6], building on the $(MC^2)^2$ framework [15], is also based on AHP. It considers VM offerings and other Cloud infrastructure services separately, and user input values are calculated by weighted parameters. The final ranking com-

Table IV: Cost Calculator Evaluation: Usage-bracket Pricing

Usage	Offering	MDSS	PFC	MPC	Unit Price
0.525 TB	Azure Storage Geographically Redundant	$49.88	$49.88	$49.88	$0.095/GB
	Google Cloud Storage	$44.62	$44.63		$0.085/GB
5.125 TB	Azure Storage Geographically Redundant	$425	$425.36	$425.36	$0.08/GB
	Google Cloud Storage	$398.5	$398.72		$0.076/GB
80.075 TB	Azure Storage Geographically Redundant	$6120.25	$6132.61	$6132.53	$0.07/GB
	Google Cloud Storage	$5464.2	$5464.4		$0.067/GB

prises of feasible combinations of VM and services. For the selection and combination of solutions, CloudGenius constructs a formal model to describe requirements, non-numerical and numerical attributes. In both cases however, applying AHP requires a signifcant amount of user input in order to prioritize the different requirements. In the case of MDSS we opted to offer the simpler but easier to use ranking of candidate offerings instead.

Furthermore, neither SMICloud, nor CloudGenius support dynamic usage definition. This functionality however is supported by the Cloud Adoption Toolkit [5], building on the tools in [2], which incorporates also a technology suitability analysis, and a stakeholder impact analysis to support the decision making. Calculating the costs however in this case relies on specifiying the application model in a UML deployment diagram with a custom profile, which requires additional effort on the user side. The toolkit also focuses on VM offerings. CloudCmp [4] extends beyond VM to include also different type of services and costs like storage and networking. The emphasis of this work however is on measuring the actual performance of the providers for a given set of applications. While not considered in this work, performance benchmarking would be an interesting addition to the migration decision support.

Finally, in terms of cost calculation tools, providers have started offering their own tools to potential and existing users of their solutions. In addition to the Windows Azure Pricing Calculator [12] discussed in the previous section, for example, Amazon offers the TCO calculator [16] that allows users to estimate the price difference between applications hosted on Amazon Web Services and on premises. All such tools are however limited to the offerings of one provider. Looking at cross-provider tools, and in addition to PlanForCloud [11] that we used for the evaluation, the Aotearoa [17] tool applies a migration support framework to achieve a multi-goal Cloud decision making. The tool however focuses on the MCDM aspect and requires users to define possible alternatives, goals and criteria themselves, without the support of a knowledge base. Ranking is depending on key weighting, resulting in big differences between alternatives, which in turn requires a good degree of experience on behalf of the users in providing weights.

VI. Conclusions and Future Work

Supporting the decision making process in migrating applications to the Cloud depends on a number of parameters. Given the emphasis of the Cloud computing paradigm in reducing expenses for application owners, decision making in the literature focuses on optimizing the selection of a Cloud service provider w.r.t. cost. This work proposes a migration decision support system that combines features from other approaches in the State of Art with those of publicly available cost calculation tools.

For this purpose we propose a three-tiered architecture incorporating a front-end for user interactions, including the ranking of returned results, a knowledge base with information collected from the Cloud service providers on their offerings, and a back-end that contains the provider selection and cost calculation logic. A prototype of this approach was implemented as a Web application using ASP.NET, Microsoft SQL Server, and Microsoft Visual C# on the .NET framework, respectively. The prototype was evaluated against a set of online tools, verifying the selection process and illustrating the precision of cost calculation.

The prototype is currently retooled to be offered as a service. The provider knowledge base is also being enriched with more providers and their offerings, and being kept up to date with the latest prices of existing providers. Furthermore, the capacity to combine requirements for multiple service types, e.g., data storage and networking, is envisioned to be provided, facilitating the decision making in migration types beyond the off-loading of the application stack on a VM [1]. Scalability of the application, both horizontal and vertical [18], must also be allowed as an option to system users, in addition to dynamic usage patterns currently offered, and considered for cost calculation purposes.

Finally, as discussed already in the introduction, the proposed work focuses only on provider selection and cost calculation. However as discussed in [3], decision support for application migration to the Cloud entails more dimensions. For example, while provider selection is a key decision, application stakeholders have also to decide how to distribute their application, which elasticity strategy to support, and how to deal with the effect of multi-tenancy to the QoS characteristics of their application. While cost calculation is an important analytical task towards supporting these decisions, a series of additional tasks like performance prediction and identification of security concerns is required [3]. In this sense the presented approach is only the first step towards a comprehensive migration support system.

Acknowledgment

The research leading to these results has received funding from the 4CaaSt project (http://www.4caast.eu) part of the European Union's Seventh Framework Programme (FP7/2007-2013) under grant agreement no. 258862.

References

[1] V. Andrikopoulos, T. Binz, F. Leymann, and S. Strauch, "How to adapt applications for the cloud environment - challenges and solutions in migrating applications to the cloud," *Computing (to appear)*, 2013. [Online]. Available: http://dx.doi.org/10.1007/s00607-012-0248-2

[2] A. Khajeh-Hosseini, I. Sommerville, J. Bogaerts, and P. Teregowda, "Decision support tools for cloud migration in the enterprise," in *Cloud Computing (CLOUD), 2011 IEEE International Conference on*. IEEE, 2011, pp. 541–548.

[3] V. Andrikopoulos, S. Strauch, and F. Leymann, "Decision support for application migration to the cloud: Challenges and vision," in *Proceedings of the 3rd International Conference on Cloud Computing and Service Science, CLOSER 2013, 8-10 May 2013, Aachen, Germany (to appear)*. SciTePress, 2013.

[4] A. Li, X. Yang, S. Kandula, and M. Zhang, "Cloudcmp: comparing public cloud providers," in *Proceedings of the 10th annual conference on Internet measurement*. ACM, 2010, pp. 1–14.

[5] A. Khajeh-Hosseini, D. Greenwood, J. Smith, and I. Sommerville, "The cloud adoption toolkit: supporting cloud adoption decisions in the enterprise," *Software: Practice and Experience*, vol. 42, no. 4, pp. 447–465, 2012.

[6] M. Menzel and R. Ranjan, "Cloudgenius: decision support for web server cloud migration," in *Proceedings of the 21st international conference on World Wide Web*. ACM, 2012, pp. 979–988.

[7] Microsoft, "Pricing details," http://www.windowsazure.com/en-us/pricing/details/?l=en-us.

[8] B. Suleiman, S. Sakr, R. Jeffery, and A. Liu, "On understanding the economics and elasticity challenges of deploying business applications on public cloud infrastructure," *Journal of Internet Services and Applications*, pp. 1–21, 2011.

[9] Amazon Web Services, "How AWS Pricing Works ," 2012. [Online]. Available: http://aws.amazon.com/whitepapers/

[10] S. Garg, S. Versteeg, and R. Buyya, "Smicloud: A framework for comparing and ranking cloud services," in *Utility and Cloud Computing (UCC), 2011 Fourth IEEE International Conference on*. IEEE, 2011, pp. 210–218.

[11] RightScale, "Cloud cost forecasting web site from right scale," http://www.planforcloud.com/.

[12] Microsoft, "Windows azure pricing calculator," http://www.windowsazure.com/en-us/pricing/calculator/.

[13] P. Pawluk, B. Simmons, M. Smit, M. Litoiu, and S. Mankovski, "Introducing STRATOS: A Cloud Broker Service," in *Cloud Computing (CLOUD), 2012 IEEE 5th International Conference on*, june 2012, pp. 891 –898.

[14] M. Zeleny, *Multiple criteria decision making*. McGraw-Hill New York, 1982, vol. 25.

[15] M. Menzel, M. Schönherr, and S. Tai, "(mc2)2: criteria, requirements and a software prototype for cloud infrastructure decisions," *Software: Practice and Experience*, 2011.

[16] Amazon, "Tco comparison calculator for web applications," http://tco.2ndwatch.com/.

[17] Aotearoa, "aotearoa," http://aotearoadecisions.appspot.com/.

[18] L. Vaquero, L. Rodero-Merino, and R. Buyya, "Dynamically Scaling Applications in the Cloud," *ACM SIGCOMM Computer Communication Review*, vol. 41, no. 1, pp. 45–52, 2011.