2010 Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises

Application Development: Fly to the Clouds or Stay in-House?

Stamatia Bibi
Department of Informatics
Aristotle University of Thessaloniki
Thessaloniki, Greece
sbibi@csd.auth.gr

Dimitrios Katsaros Panayiotis Bozanis
Department of Computer & Communications Engineering
University of Thessaly
Volos, Greece
{dkatsar, pbozanis}@inf.uth.gr

***Abstract*— Cloud computing is a recent trend in IT that moves computing and data away from desktop and portable PCs into large data centers, and outsources the "applications" (hardware and software) as services over the Internet. Cloud computing promises to increase the velocity with which applications are deployed, increase innovation, and lower costs, all while increasing business agility. But, is the migration to the Cloud the most profitable option for every business? This article presents a study of the basic parameters for estimating the potential costs deriving from building and deploying applications on cloud and on-premise assets.**

Keywords- Cloud computing, software development, software cost.

I. INTRODUCTION

IT managers are recently faced with the problem of making a selection between cloud computing and on-premise development and deployment. Cloud computing option is attractive especially if the quality delivered and the total cost is satisfying and the risks are reasonable. The real question for many IT departments is whether the cost of transition to an external compute cloud will be low enough to benefit from any medium-term savings [1], [2], [17]. In order to be able to provide answers to the above question a formal cost analysis of cloud and on-premise deployment should be performed in order to compare thoroughly the two alternatives. This paper presents basic parameters for estimating the potential costs deriving from building and deploying applications on cloud and on-premise assets.

Moving from traditional IT infrastructure and adopting cloud computing technologies raises two important questions. The first one is about the cost required for such a change and the second one refers to the quality of services provided. A thorough analysis of the estimated costs and quality associated with the two alternatives will help an IT manager define the pros and cons of each solution. Such an analysis will point out which is the right combination of cloud and premise based assets and can indeed provide the optimal solution. As mentioned in [10] the key is not choosing between the two solutions but being strategic about where to deploy various hardware and software components of a total solution.

The assessment of cloud computing costs is more evident compared to the assessment of on premises development and deployment. The cost of cloud computing services initially depends on the usage of three types of delivery models namely software-as-a-service, platform-as-a-service and infrastructure-as-a-service. The usage is counted and billed based on the committed resources per hour or the number of users per hour.

Estimating the cost of software development and deployment based on on-premise assets is a more complex procedure. On–premises application development includes a variety of different costs associated with IT infrastructure and software development. Estimating in-house development and deployment of software is a difficult task as there is a variety of different cost drivers related to personnel, product, process, hardware and operation expenses.

In [8] a framework for estimating the value of cloud computing is presented that is based on the business scenario and the quantitative analysis. In this paper we focus on the quantitative analysis and suggest specific attributes that should be taken into consideration before making a decision regarding cloud adoption of assets.

In this paper we introduce a framework for estimating the value of IT deployment when using a combination of cloud and on-premise assets. For this reason we analyze costs that need to be assessed in both deployment options. In Section II we describe cloud computing utilities their associated costs and corresponding on-premise costs, Section III discusses infrastructure and software development costs related to on-premise application deployment. In section IV we provide a three step procedure that will assist IT managers to understand the benefits of combined solutions. Section V concludes the paper and discusses future work.

II. CLOUD COMPUTING UTILITIES COSTS

In this section our goal is to clarify which services are offered by cloud computing and how they are related to on-premise software and system costs. The main purpose of Cloud Computing is to provide a platform to develop, test, deploy and maintain Web-scale applications and services [7]. A formal, and widely accepted, definition of cloud computing is not found in the literature [9], but most resources refer to this term for anything that involves the delivery of hosted services (applications, platforms, raw computing power and storage, and managed services) over the Internet which are billed by consumption. These services are broadly divided into three categories [3], [4]: Software-as-a-Service (SaaS), Platform-as-a-Service (PaaS) and Infrastructure-as-a-Service (IaaS).

978-0-7695-4063-4/10 $26.00 © 2010 IEEE
DOI 10.1109/WETICE.2010.16

Software as a Service is a software distribution and usage model that is available via a network to the customers. Both horizontal and vertical market software are leased through Service Level Agreements. Typical examples of horizontal SaaS are subscription management software, mail servers, search engines and office suites. Examples of vertical SaaS are more specialized software such as Accounting software, Management Information systems and Customer Relationship Management systems.

SaaS investment is typically limited to the subscription fee. This pricing model provides a predictable investment that follows a pay per usage billing scheme. Usually costs are calculated considering user licenses, customizations costs and end user support and training costs [11]. The last three types of cost refer mostly to software for vertical needs. These costs are determined in a Service Level Agreement (SLA) that defines the pay-on-demand rates.

Costs related to adopting horizontal software on-premise involves licensing fees, maintenance and upgrade costs. Expenses related to software for vertical needs developed on premise are increased, in comparison with horizontal software, because the adoption of such software requires all software development lifecycle costs. These costs involve analysis, design, development, testing, delivery, training and maintenance costs.

Platform as a service is a development platform hosted to the cloud and accessed via a network. The functionality that PaaS offers involves at least the following: Developer studios that include all necessary tools to build a web application, seamless deployment to hosted runtime environment and management and monitoring tools. PaaS offers the potential for general developers to build web applications without having any tools installed in their own space.

Infrastructure as a Service involves the physical storage space and processing capabilities that enable the use of SaaS and PaaS if wanted otherwise these services are used autonomously by the customer. Hardware, servers and networking components are typical examples of IaaS.

PaaS and IaaS costs are usually interrelated. PaaS costs involve usage of operating systems, database management systems, web hosting server software, batch processing software and application development environments. Usually costs of PaaS are calculated as on-demand instances per hour. On-Demand Instances refer to the number of servers used, the operating systems and middleware applications loaded to them. PaaA payment is based on on-demand instances that allow payment for compute capacity by the hour measured. IaaS costs are also based on CPU, memory and hard disk usage. Annual agreements can be made based on the reserved instances and the level of cpu and memory usage [12].

The associated costs regarding platform customization in in-house development depends on selected middleware products. Middleware products can be divided into two groups open source software which is freely distributed and commercial software. The costs in these two cases are significantly different. Open source software although it is not purchased it presents significant installation and maintenance costs. Commercial products expenses apart from the software licenses usually are accompanied by an additional service agreement that includes training, upgrading and maintenance costs. The costs related to in – premise infrastructure involve the usual costs of hardware such as equipment cost, electricity and pooling costs, maintenance, upgrading and administration costs.

III. ON-PREMISE SOFWARE DEVELOPMENT AND DEPLOYMENT

This Section discusses the costs related to IT infrastructure and software development for an application based on on–premise assets.

A. IT Infrastructure Costs

When estimating software development and maintenance costs, IT infrastructure costs should also accounted. IT costs are non-negligible as usually they stand up to 60% of Total Ownership costs [5], [6], [18]. Unlike software development estimation IT estimation is a simpler process as infrastructure and services are more tangible. The cost drivers that influence IT costs as mentioned in [18] and [7], [19] can be operational attributes and business premises.

Operational attributes refer to hardware costs, software and system license fees. Hardware costs include new resources acquisition, replacement and maintenance of existing resources. Hardware acquisition costs depend on the infrastructure hardware list (servers) and the end user hardware list (laptops, CPU, printers). Hardware maintenance costs usually are estimated using measures that compute the Mean Time To Failure (MTTF) or Mean Time Between Failures (MTBF). Software, system and database license fees refer to operational software that will be installed in computer systems necessary for the operation of the new application. License costs are defined by the number of inbound and outbound workstations in which the new application will be installed. The number of users usually affects cost mainly through the number of software licenses needed and recruitment and training costs.

Several performance factors are associated with the non-functional requirements of an application that apart from the need to incorporate them in the software also rise the need for business premises. The average transaction rate, the storage needs, security issues and reliability factors require computational power and capacities. Computational power in low level is related to electricity costs. Other business premises that are necessary for IT development and are associated to total costs involve labor rates, outsourcing agreements and operational locations. Labor rates are related to the personnel expenses and training procedures.

Outsourcing agreements may include hardware/software leasing or development. Different physical locations of the organization and different access points to the application are associated to rental or leasing expenses.

TABLE I. OPERATIONAL DRIVERS

	Drivers
	Operational Drivers
New resources	Servers, Laptops, PCs
	Peripheral devices, CPU, memory
	WAN/LAN equipment
Maintenance and replacement costs	CPU
	Hard Disk
	Power supply
	CPU Cooler
License fees	Application Software (office applications, mail)
	System Software (Operating system)
	Database (Licences for end users)

TABLE II. BUSINESS PREMISES

	Drivers
	Business Premises
Personnel Expenses	Labor Rates
	Training expenses
Electricity costs	Electricity consumption
Physical Locations	Rental expenses

B. *Software Development Costs*

Software development costs are divided into four groups. Product, platform, process and personnel drivers are pointed out in the literature [20] as the most important aspects that determine software costs.

Product attributes related to a software project include descriptive variables and size indicators. The aggregation of variables of both categories is indicative of the complexity of the new projects and the expected difficulties that might rise. Descriptive variables provide information regarding the development type of the project, the application type (IT project type ERP, MIS, CRM or Web applications, e.t.c) and the user type of the application (professional, amateur, concurrent, casual. In order to estimate size attributes an initial assessment of functional requirements is necessary. From functional requirements we can provide a size estimate measured in function points [21] or in Lines of Code [20]. Accurate size estimation is a very important task as it is considered to directly affect the amount of effort required to complete a software project.

TABLE III. PRODUCT DRIVERS

Table Head	Drivers	
	Product Drivers	*Metric*
Type of project	Application Type	ERP, Telecommunications, Logistics, e.t.c
	Business Type	Medical, Public Sector, Transports, Media e.t.c
	Development Type	New Development, Re-development, Enhancement
User type	Level of usage	Amateur, Professional, Casual
	Number of Users	1-50, 50-200, 200-1000, >1000
Size	Source Code Lines	Lines of Code (LOC)
	Function Points	Number of Function Points

TABLE IV. PROCESS DRIVERS

Table Head	Drivers	
	Product Drivers	*Metric*
Technical attributes	Distributed Databases	1-5 Scale that depicts the necessity of the attribute.
	On-line Processing	1-5 Scale
	Data communications	1-5 Scale
	Back-ups	1-5 Scale
	Memory constraints	1-5 Scale
	Use of new or immature technologies	1-5 Scale
Non-functional requirements	Reliability	1-5 Scale
	Performance	1-5 Scale
	Installation Ease	1-5 Scale
	Usability	1-5 Scale

Non-functional requirements affect the values of platform drivers and can oppose certain constraints or lead to conflicting interests. Examples of non-functional requirements are software reliability, database size, security issues, performance standards, usability issues and transaction rates. Other drivers that directly affect platform costs are incremented memory needs, increased storage facilities and maintenance of back up files. All the above parameters capture platform complexity of the software under development.

Process attributes refer to all project supplements that may be used and enable the development and delivery of quality software within cost and time limitations. Among these characteristics the use of CASE (Computer Aided Software Engineering), the utilization of methods, techniques and standards are the main aspects that define the level of support and observation of the development procedure. Productive development teams usually follow a well-defined and guided process. Proven best practices, methodologies and the selection of the appropriate lifecycle processes are aspects that a development team should rely on to complete a project. The success of a project, the time and cost required for its completion depends on the existence of a well-managed process.

TABLE V. SOFTWARE PROCESS AND PERSONNEL REQUIREMENTS

Table Head	Drivers	
	Product Drivers	***Metric***
Use of Case Tools	Versioning tools	% of usage
	Analysis & Design Tools	% of usage
	Testing Tools	% of usage
Management Process	Use of lifecycle models	Yes or No
	Managed development Schedule	1-5 Scale
Methodologies	Existance of best practices	1-5 Scale
	Software Reuse	% of the total LOC

Software costs are also dependant on personnel team attributes. Typical examples of this group of cost drivers are the experience of the team, the analysts' capabilities, the familiarity with the programming language and the application. Recent studies also point out that cultural characteristics also determine software costs. Well structured teams that encourage communication, allow knowledge exchange and support reward mechanisms are more productive compared to impersonal teams. The capabilities of the personnel and the motivation of the environment affect directly the productivity of a development team thus the total developments costs.

TABLE VI. SOFTWARE PROCESS AND PERSONNEL REQUIREMNTS

Table Head	Drivers	
	Product Drivers	***Metric***
Experience	Analysts cababilities	1-5 Scale
	Programmers experience	1-5 Scale
	Familiarity with the problem domain	1-5 Scale
Cultural issues	Reward mechanism	1-5 Scale
	Collaboration	1-5 Scale
	Cabable leadership	1-5 Scale

IV. DESICION MODEL

IT managers are faced with the problem of selecting how and where to develop and deploy their applications. The requirements of an application will determine the choice between cloud computing and development on premises or even a combination of both [1]. Each of the two different options presents advantages and disadvantages on various fields. The business goals and priorities of the application will determine the level of usage of cloud or premise assets. IT decision making often requires trading between innovation and time-to-value advantages of cloud computing against performance and compliance advantages of development on-premise. For this reason we propose a three step procedure that will assist in decision making:

1) Assess software and infrastructure development costs.

2) Define quality characteristics.

3) Estimate user demand.

The issue of deciding whether to develop and deploy the applications in the cloud was also addressed in [8], but our three-step process is somewhat more generic as it includes detailed recording of relevant parameters.

A. Assess Sofware and Infrastructure Development Costs

This procedure involves costs assessment of the two alternative solutions. Sections II and III will be useful to keep in mind all the relevant aspects of the problem. A five year total cost of ownership projection will be useful to determine long-term benefits of each solution.

We will discuss the deployment of Customer Relationship Management Systems; a common business application that is becoming popular on the cloud. We will focus on software development costs of such an application.

Customer Relationship Management (CRM) is an information industry term for methodologies, software, and Internet capabilities that help an enterprise manage customer relationships in an organized and efficient manner [22]. CRM functionality may include product plans and offerings, customer notifications, design of special offers, e.t.c.

Based on data coming from the International Standards and Benchmarking Group (ISBSG) CRM systems on the average require 1867 total effort hours for completion with average size measured in IFPUG function points 181.5 fp (the average is calculated using the median point of the distribution of values in order to minimize the effect of outliers). Keeping in mind average US salaries (4141 US$), 1867 effort hours correspond to 233 workdays, 11,65 months and 48242$. Analyzing the projects that include relevant data we can see that 56% of the projects require development teams larger than 9 people. All CRM projects developed in-house followed a particular methodology while only 33% of projects that presented values for that field were supported by the use of CASE tools. Programming Languages used are C, C#, Cobol, Visual basic and Oracle at 65%. The hardware used for the application was 39% of the applications PCs and 15% clients and servers. The dominant database was Oracle 41,1%.

On the other hand CRM cloud applications with Zoho [14] and Salesforce [13] leading providers charge based on the number of users and the number of applications accessed. The prices range from 12$ per month to 75$ per month, per user. Considering in that case 5 potential users that will use a sublist of the product features charged 50$ per month the annual costs are calculated to be 3000$.

B. *Define Quality Characteristics*

Quality characteristics are closely associated to business goals and most of the times are defined as non-functional requirements. An initial assessment involves the definition of non functional requirements and their priority.

Among the quality characteristics that are incorporated in cloud computing is improved performance. Computers in a cloud computing system boot and run faster because they have fewer programs and processes loaded into memory [15]. Compatibility is another attribute that is supported by cloud computing. Documents created in a Web application can be read and processed without any special installation on the users PCs. Increased data reliability is also ensured as cloud is considered the ultimate back-up. Interoperability and availability are two other quality characteristics of cloud computing. Interoperability and availability allow user to have access to the system any time, anywhere by any computer or network.

In premise software advantages involve data accessibility, ownership and safety. The biggest advantage of on-premise software is that businesses have complete control over their critical business data [16]. This is also a main benefit for data intensive applications that should support high volumes of transactions. Other advantage of the on-premise software is that it allows integration with existing software/hardware resources. Customization is one more quality characteristic of in–premise software.

CRM systems usually store, handle and process sensitive private data of customers that should not in any case leak to competitors. Therefore safety is an important non functional requirement. Other important features involve the back-up file storage, and online any-time, any where immediate access to the system. Usability is another important feature for such an application. A customer should be able to navigate through different functionalities and access the information he needs easily and quickly. Prioritizing non functional requirements is an indicator that will help managers take a decision regarding the development and deployment of a system.

C. *Estimate User Demand*

Estimating the expected demand of the application is also very important in order to assess costs. Expected demand is associated with the number of users. The number of users affects licensing costs and hardware costs. Licensing costs are considered for users that access the applications and make changes of any kind. On the other hand, for hardware as the number of users increases, the hardware must also be improved or performance becomes unacceptable. Centralized database models present reasonable costs for 5-10 users, but present exponential growth of costs as the number of users increases. Distributed models are a solution to such problems shifting costs to PCs. Administration fees are also affected by the number of users as normally one administrator is considered every 5-10 users.

While estimating the number of users according to [8], one should keep in mind four types of demand. Expected Demand: Seasonal demand, Expected demand: Temporary effect, Expected Demand: Batch processing, Unexpected Demand: Temporal Effect.

For the CRM system seasonal demand refers to sales and retails periods that usually present increased demand volume. Temporary effect may refer to clearance period or possible relocation that are seldom events that may cause extra demands. Expected demands: Batch processing may involve for the CRM a period that massive advertisements are shifted. Finally, unexpected demands for the CRM may occur when a new product of the company becomes very popular unexpectedly.

V. Conclusions and Future Work

In this paper we have taken a first step towards identifying all relevant costs of cloud computing and on-premises infrastructure and software. We proposed a three step decision model for evaluating the two alternatives. Software development and infrastructure costs, desired quality characteristics of the application and expected number of users are the main aspects that a software manager has to consider. The final choice may be the deployment of an application on the cloud, on business premises or by adopting a combination of the two aforementioned alternatives.

In our future work we aim to evaluate the proposed model on real world applications deployment and compare the three alternatives (cloud, on–premise, an amalgamation of the two) based on data coming from both in-house development and cloud hosting. In particular for the hybrid of the two worlds, we plan to elaborate on the cases where it is more profitable and derive appropriate "rules-of-thumb", since we argue that this model will be the one that will finally dominate the market.

Additionally, we plan to work toward revisiting the software development cost models, which should be reconsidered under the prism of new programming paradigms, such as Map-Reduce [23], [24]. In the future data centers, which will be the heart of cloud computations, such new programming paradigms are expected to dominate the development market.

References

[1] M. Armbrust, A Fox, R. Griffith, A.D. Joseph, R.H. Katz, A. Konwinski, G. Lee, D.A. Patterson, A. Rabkin, I. Stoica, M. Zaharia, "*Above the clouds: A Berkeley view of cloud computing*", Technical Report EECS-2009-28, University of California at Berkeley, February, 2008.

[2] M. Armbrust, A. Fox, R. Griffith, A.D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, M. Zaharia, "*A view of cloud computing*", Communications of the ACM, vol. 53, no. 4, 2010, pp. 50-58.

[3] M.D. Dikaiakos, D. Katsaros, P. Mehra, G. Pallis, A. Vakali, "*Cloud computing: Distributed Internet computing for IT and scientific research*", IEEE Internet Computing, vol., 13, no. 5, 2009, pp. 10-13.

[4] A. Lenk, M. Klems, J. Nimis, S. Tai, T. Sandholm, "*What's inside the cloud? An architectural map of the cloud landscape*", Proceedings of the ICSE Workshop on Software Engineering Challenges of Cloud Computing (CLOUD), 2009, pp. 23-31.

[5] J. Gray, "*Distributed computing economics*", Technical Report MSR-TR-2003-24, Microsoft Research, March 2003.

[6] K. McRitchie, S. Accelar, "*A structured framework for estimating IT projects and IT support*", Joint Annual Conference ISPA/SCEA Society of Cost Estimating and Analysis, May 2008.

[7] A. Optitz, H. Konig and S. Szamlewska, "*What does grid computing cost*", Journal of Grid Computing, vol. 6, no. 6, 2008, pp. 385-397.

[8] M. Klems, J. Nemis and S. Tai, "*Do clouds compute? A framework for estimating the value of cloud computing*", Lecture Notes in Business Information Processing, Springer-Verlang, vol. 4, 2009, pp. 110-123.

[9] L.M. Vaquero, L Rodero-Merino, J. Caceres, M. Lindner, "*A break in the clouds: Towards a cloud definition*", ACM SIGCOMM Computer Communications Review, vol. 39, no. 1, 2009, pp. 50-55.

[10] Combination http://blogs.cisco.com/collaboration/comments/why_cloud_vs._premise_is_the_wrong_question/

[11] SaaS costs http://www.crmlandmark.com/saasTCO.htm

[12] Amazon Platform as a service http://aws.amazon.com/ec2/

[13] CRM SaaS: computing http://www.salesforce.com/platform/platform-edition/

[14] CRM SaaS: http://www.zoho.com/

[15] M. Miller, Cloud computing pros and cons for end users, http://www.informit.com/articles/article.aspx?p=1324280

[16] G. MacGowan, Helping small businesses choose between on-demand and on-premise software, http://www.computerworld.com/s/article/9002362/Helping_small_businesses_choose_between_On_demand_and_On_premise_software

[17] Cloud Computing Congress, http://www.cloudcomputingchina.org/

[18] http://www.galorath.com/index.php/library/userconference2008/

[19] Chapter 12, *Sofwtare Cost Estimating*. http://www.techamerica.org/

[20] B. Boehm, *Software Engineering Economics*. Englewood Cliffs, NJ, Prentice-Hall, 1981.

[21] A. J. Albrecht, "*Measuring application development productivity*", Proceedings of the Joint SHARE, GUIDE, and IBM Application Development Symposium, October 1979, pp. 83–92.

[22] Laudon & Laudon, *Management Information Systems*, Pearson, 2009.

[23] J. Dean and S. Ghemawat, "MapReduce: Simplified Data Processing on Large Clusters", Proceedings of the ACM/USENIX Symposium on Operating System Design and Implementation (OSDI), 2004, pp. 137-150.

[24] Hadoop, http://hadoop.apache.org/