Cost models – pillars for efficient cloud computing: position paper

Catalin Negru* and Valentin Cristea

Faculty of Automatic Control and Computer Science, University POLITHEHNICA of Bucharest, Splaiul Independentei nr. 313, sector 6, Bucuresti RO-060042, Romania
E-mail: catalin.negru@cs.pub.ro
E-mail: valentin.cristea@gmail.com
*Corresponding author

Abstract: Cost models are fundamental building blocks for cloud computing. Cloud providers offer a very wide portfolio of services, while Cloud clients access them against some financial arrangement. There is a fundamental trade-off between what Cloud provider can offer in terms of resources (software or hardware), services (e.g., storage, e-mail) cost and what Cloud clients are willing to pay. The Cloud cost models today are distinct based on service models: Infrastructure as a Service (IaaS), Platform as a Service (PaaS) and Software as a Service (SaaS). The need to forecast the cost over a period of time imposes building of cost models, which have to be accurate and error free. In this paper, we make a survey and analyse existing cost models in cloud computing and discuss open issues related to the subject.

Keywords: cost models; cloud computing; efficiency; optimisation.


Biographical notes: Catalin Negru is currently a System Engineer and PhD student within the Computer Science Department, in Politehnica University of Bucharest, Romania. His general interests regard cloud computing. His current research is focused on optimising the cost for cloud storage systems.

Valentin Cristea is a Professor of the Computer Science and Engineering Department of University POLITEHNICA of Bucharest. His main fields of expertise are distributed systems, grid computing and e-services. In 2003 and 2011 he received the IBM faculty award. He published more than 120 specialist papers, 25 books (in Romanian, English, and French), and 60 technical reports. He is PhD supervisor, coordinator of the Master program on Advanced Software Services, and the Romanian coordinator of the Master program on Parallel and Distributed Computer Systems co-developed with VU Amsterdam.
1 Introduction

Cloud computing paradigm permits to the users to use computational and software resources in a pay-per-use manner. In this way the user has a great flexibility to adapt to changes. In cloud computing everything is provided as a service to cloud user, in a functional, usable and extremely powerful manner, meaning that cloud clients can access the services almost everywhere if they have internet connection and have access to a large number of resources. Companies like Google, Facebook, Yahoo! and Microsoft have invested in their data centres to support cloud services, the natural question that rises for that fact is the following one: where does the cost go in today’s cloud service data centres?

There are three principal types of services that can be offered in cloud computing: Infrastructure as a Service (IaaS), Platform as a Service (PaaS) and Software as a Service (SaaS). IaaS assume provisioning with fundamental computing resources such as processing, storage, networks and so on, to the consumer which can deploy and run arbitrary software, meaning middleware software, operating systems and others. The consumer has control over operating systems, storage and deployed applications but does not manage or control the underlying cloud infrastructure (Armbrust et al., 2010).

PaaS offer to the consumer the capability to deploy onto the cloud infrastructure applications created using programming languages, tools and libraries supported by the provider. The consumer has control over the deployed applications and possibly configuration settings for the application-hosting environment (Armbrust et al., 2010).

SaaS is the capability provided to the consumer to use the provider’s applications running on a cloud infrastructure. The applications are accessible from various client devices through either a thin client interface, such as a web browser (e.g., web-based e-mail), or a programme interface (Armbrust et al., 2010).

Cost structure represents the identification of how costs associated with the production of a good or service is distributed through the process (Martens et al., 2012). Is important to identify the cost factors that determine the overall cost for offering a certain service in cloud computing (Kondo et al., 2009).

Another important aspect that must be taken into consideration is represented by the metrics that cloud providers should use to quantify their investment, and to calculate the profitability for a certain service that is offered.

The remainder of this paper is organised as follows: Section 2 present the most important cost factors in cloud computing. In Section 3, we present metrics used in cost models to calculate the profitability of investment in a cloud computing platform. Section 4 review existing cost models in cloud computing. Section 5 presents different pricing strategies used in cloud computing. Section 6 presents different cost reduction strategies used today in cloud platforms to optimise the utilisation of a cloud platform. Finally, Section 7 presents a discussion related to cost models and draws a few conclusions.

2 Important cost factors in cloud computing

In this section, we present the most important factors that are included in the structure of the cost for a cloud platform. There are two sides of the same problem: the cloud provider perspective and cloud user perspective. The cloud provider must calculate his total cost of
ownership and adequately put price on his services to have profit and amortise his investment and on the other side cloud user has to calculate total cost of running his application in the cloud so there are cost factors that regard one side or another. Table 1 gives the costs for the service user, and also for the service provider.

<table>
<thead>
<tr>
<th>Table 1</th>
<th>TOC perspective</th>
</tr>
</thead>
<tbody>
<tr>
<td>Component</td>
<td>IaaS</td>
</tr>
<tr>
<td>Business process</td>
<td>Customer</td>
</tr>
<tr>
<td>Business logic</td>
<td>Customer</td>
</tr>
<tr>
<td>Middleware management</td>
<td>Customer</td>
</tr>
<tr>
<td>Application licensing/Support</td>
<td>Customer</td>
</tr>
<tr>
<td>OS management</td>
<td>Customer</td>
</tr>
<tr>
<td>OS licensing/Support</td>
<td>Service provider</td>
</tr>
<tr>
<td>Server/Storage/Networks</td>
<td>Service provider</td>
</tr>
<tr>
<td>HW/Maintenance</td>
<td>Service provider</td>
</tr>
<tr>
<td>Utilities</td>
<td>Service provider</td>
</tr>
<tr>
<td>ME equipment</td>
<td>Service provider</td>
</tr>
<tr>
<td>Real estate</td>
<td>Service provider</td>
</tr>
</tbody>
</table>

Source: Rutland (2012)

The authors of Kashef and Altmann (2011) identify five groups of cost factors that mainly regards cloud provider and partially to cloud users: electricity, hardware, software, labour and business premises.

The electricity cost factor refers to: electricity for cooling infrastructure, for powering computing and networking devices and other electronic devices. Also must be taken into consideration the two values of power consumption: the value when the system is idle and when the system is heavily used.

The hardware cost factor refers to the acquisition of hardware and networking resources, needed in-house. Because the all the resources have an optimal lifetime for exploitation, a depreciation parameter must be taken into consideration to evaluate the amortisation of the investment. Software cost refers to the price that cloud provider have to pay to purchase the software licenses for server operating systems, middleware software, application software and are applicable to a SaaS provider.

Labour cost factor include salaries for technicians who work for maintenance of the data centre and in the support area. This cost varies in function of the location of the data centre.

Business premises cost factor includes collateral cost such as: the price for data centre facility, price for all non-electric instruments, the price for cabling and so on.

A consumer of cloud services must take into consideration the following factors to calculate his total cost for running an application: the usage cost for servers (CPU hours), the cost for incoming data transfer in the cloud platform, cost for the amount of outgoing data transfer which are measured in Giga Bytes per seconds (GB/s), cloud storage factor measured in Giga Bytes (GB), the cost for executing input requests in cloud platform, the cost for executing an output requests from cloud platform.
3 Important metrics in cost models

To calculate the profitability of investment, the cloud provider must perform financial analysis with the aid of a few parameters that can be calculated along with Return of Investment (ROI) such as Net Present Value and Internal Rate of Return (Tak et al., 2011).

Net Present Value (NPV) is defined as the difference between the present value of cash inflows and the present value of cash outflows. NPV is used in capital budgeting to analyse the profitability of an investment or project. The formula for NPV (http://en.wikipedia.org/wiki/Net_present_value) is presented below:

$$ NPV(i, N) = \sum_{t=0}^{N} \frac{R_t}{(1+i)^t} $$

where \( t \) is the time of the cash flow; \( i \) is discount rate (the rate of return that could be earned on an investment in the financial markets with similar risk); the opportunity cost of capital; \( R_t \) is the net cash flow (the amount of cash, inflow minus outflow) at time \( t \).

The Internal Rate of Return (IRR) is a parameter for measuring the financial evaluation. It is the discount rate for which a project’s benefits exactly equal its costs, meaning that, it is the rate at which the project’s net present value is zero. The evolution of IRR parameter is shown in Figure 1. As can be seen a higher IRR parameter make an investment more desirable, and exceeding the IRR value, will produce business loss.

Figure 1  Usual internal rate of return (see online version for colours)

Equation (2) describes the IRR parameter:

$$ \sum_{t=0}^{\tau} \frac{B_t}{(1+i)^t} - C_0 = 0. $$

ROI represent an important indicator that helps in decision of moving or not in to the cloud, by measuring the efficiency of an investment. When calculating ROI a large number of factors involved in a business are taking into consideration. For calculating ROI, the following information’s are needed: the initial cost of project, the investment made, the cost savings done owing to the new investment (Misra and Mondal, 2011).
The ROI formula for cloud computing with time frame per month or per year is the following (Misra and Mondal, 2011):

\[
\text{ROI} = \frac{(\text{Initial cost} - \text{Final cost}) - \text{Investment}}{\text{Investment}} = \frac{\text{Costs saved} - \text{Investment}}{\text{Investment}}.
\]

4 Different cost models in cloud computing

The authors of Truong and Dustdar (2010) present basic cost models which are shown in Table 2. The first four models, \( M_{ds} \) (cost model for data storage); \( M_{cm} \) (cost model for computational machine); \( M_{dfi} \) (cost model for data transfer into the cloud) and \( M_{dfo} \) (cost model for data transfer out to the cloud) are provided by cloud providers in their pricing specifications. Utilising presented cost models and monitoring data the authors develop a cost estimation, monitoring and analysis service for the scientific application that run in cloud environments. The architecture of the solution is present in Figure 2. The shortcoming of this solution is represented by the fact that is applicable only to scientific applications and the model does not take into consideration the workload of the application (how many users utilise the service or application at one time) and therefore cannot be applicable to other types of applications or services for example a mail service or a website.

The authors of Kashef and Altmann (2011) proposed a cost model for hybrid cloud (i.e., the combinations of a private data centre (private Cloud) and the public Cloud). They present a conceptual model that assumes an organisation which comprises the execution of \( N \) applications and \( M \) services. Cloud users buy services to construct their applications. For the cost of data communication is used a directed weighted graph which is shown in Figure 3, where edges show data communication and vertices represent services. On the basis of the graph is constructed a distance matrix, which represent the data-transfer related cost factors \( a_{ij} \) between service \( i \) and \( j \). The authors propose a cost formula which is the sum of two types of cost: a fixed cost based on cost factors presented earlier and a variable cost based on the services used for cloud provider. The problem with this approach is that it takes into consideration direct usage of resource and do not take into account the Burstiness of the services, and unallocated resources. This problem in solved by the authors of Gmach et al. (2011) by taking into account the Burstiness of the applications and unallocated resources.

Another important aspect for cost models in cloud computing is represented by virtualisation technology, which impose supplementary challenges for IaaS providers to estimate cost and for SaaS providers to bill.

In Gmach et al. (2011), present three models for apportioning cost in a virtualised environment and indicate the one that provide the most robust and repeatable cost estimates. First, they consider a server-usage model that takes into account only the direct resource consumption by \( W \) workloads, with cost \( C_s \) of server \( s \), cost that is composed by CAPEX component (e.g., fraction of acquisition costs based on the length of the considered interval) and OPEX component (e.g., costs for power associated with the server). They define the following server workload:
Cost models – pillars for efficient cloud computing: position paper

\[
\prod_{s,w} = C_s \sum_{w'=} d_{s,w'} (3)
\]

To take into account Burstiness and unallocated resources the authors partition server cost \(C_s\) based on utilisation to get \(C^d_s\), \(C^b_s\), \(C^e_s\), respectively, where \(C^d_s\) corresponds to costs associated with the average utilisation of the server \(s\), and \(C^b_s\) and \(C^e_s\) correspond to the difference between peak and average utilisation of the resource, and difference between 100% and the peak utilisation of the resource, respectively.

Another model is server-burst model that divide the burst portion of cost for a server in a manner that is weighted by the Burstiness of each workload on the server. In a second step, unallocated resources of the server are apportioned based on the burst costs (Gmach et al., 2011). The server burst is defined as:

\[
\prod_{s,w} = C^d_s \sum_{w=0}^d d_{s,w} + C^b_s \sum_{w=0}^b (e + b_{s,w}) (4)
\]

\[
\prod_{s,w} = \prod_{s,w} + C^e_s \sum_{w=0}^e \prod_{s,w} (5)
\]

Table 2 Costs \(t_c\) and \(t_p(p)\) are the total elapsed time for executing computational task or data transfer and the execution time obtained with \(p\) parallel processes/threads, respectively. \(n\) is the number of machines used or to be used.

<table>
<thead>
<tr>
<th>Model</th>
<th>Activities</th>
<th>Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>(M_{ds})</td>
<td>Data storage</td>
<td>size(total) \times t_{sub} \times \text{cost (storage)}, where (t_{sub}) is subscription time</td>
</tr>
<tr>
<td>(M_{cm})</td>
<td>Computational machine</td>
<td>\text{cost(machine)}</td>
</tr>
<tr>
<td>(M_{dti})</td>
<td>Data transfer into the cloud</td>
<td>\text{cost(transfer}_{in})</td>
</tr>
<tr>
<td>(M_{dio})</td>
<td>Data transfer out to the cloud</td>
<td>\text{cost(transfer}_{out})</td>
</tr>
<tr>
<td>(M_{ds})</td>
<td>Single data transfer without the cost for machines performing the transfer</td>
<td>size(in) \times M_{dti} + size(out) \times M_{dio}</td>
</tr>
<tr>
<td>(M_{sm})</td>
<td>Sequential/multi-threaded program or single data transfer with the cost for machines performing the transfer (cost monitoring)</td>
<td>(t_c \times M_{cm} + \text{size(out)} \times M_{dio} + \text{size(in)} \times M_{dio} = f_{pi} \times M_{cm} + \text{size(out)} \times M_{dio} + \text{size(in)} \times M_{dio} )</td>
</tr>
<tr>
<td>(M_{se})</td>
<td>Sequential or multi-threaded program (cost estimation)</td>
<td>(f_{pi} \times M_{cm} + \text{size(out)} \times M_{dio} + \text{size(in)} \times M_{dio} = f_{pi} \times M_{cm} + \text{size(out)} \times M_{dio} + \text{size(in)} \times M_{dio} )</td>
</tr>
<tr>
<td>(M_{pm})</td>
<td>Parallel/MPI programs on multiple machines (cost monitoring)</td>
<td>(n \times M_{cm} \times t_c + \text{size(out)} \times M_{dio} + \text{size(in)} \times M_{dio} = f_{pi} \times M_{cm} + \text{size(out)} \times M_{dio} + \text{size(in)} \times M_{dio} )</td>
</tr>
<tr>
<td>(M_{pe})</td>
<td>Parallel/MPI programs on multiple machines (cost estimation)</td>
<td>(n \times M_{cm} \times f_{pi} + \text{size(out)} \times M_{dio} + \text{size(in)} \times M_{dio} = f_{pi} \times M_{cm} + \text{size(out)} \times M_{dio} + \text{size(in)} \times M_{dio} )</td>
</tr>
</tbody>
</table>
The third model proposed is pool-burst model. Burstiness cost and unallocated resources using measures for the $S$ servers in the resource pool instead of the individual servers:

$$\prod_{s,w} \text{pool-burst-temp} = C_s^d \frac{d_{s,w}}{\sum_{s'w'} d_{s',w'}} + \left( \sum_{j=1}^{S} C_j^b \right) \frac{\varepsilon + b_{s,w}}{\sum_{j=1}^{S} \sum_{w'=1}^{W} \varepsilon + b_{j,w'}}$$  \hspace{1cm} (6)$$

$$\prod_{s,w} \text{pool-burst} = \prod_{s,w} + \left( \sum_{j=1}^{S} C_j^d \right) \prod_{j=1}^{S} \frac{\prod_{w'=1}^{W} \text{pool-burst-temp}}{\prod_{j'=1}^{S} \text{pool-burst-temp}}.$$  \hspace{1cm} (7)$$

The cost models presented in Gmach et al. (2011) take into consideration only the server usage cost. The cost for data communication of the application is not taken into consideration so therefore is not a complete cost model.

---

C. Negru and V. Cristea

**Figure 2** Cloud cost estimation and monitoring service

![Cloud cost estimation and monitoring service](image)

*Source: Truong and Dustdar (2010)*

**Figure 3** Direct weighted graph

![Direct weighted graph](image)

*Source: Kashef and Altmann (2011)*
From the cost models presented above namely that one of them are tailored only to HPC applications, others take into consideration only the server cost and ignore the communication costs, raises clearly the necessity for a general cost model that can accurate predict the cost for running an application in cloud environment.

In Table 3, present the drawbacks of existing cost models.

<table>
<thead>
<tr>
<th>Cost model</th>
<th>Advantage</th>
<th>Disadvantages</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model for data storage</td>
<td>These cost models can be combined, in order to</td>
<td>Those models does not take into consideration the workload of the application</td>
</tr>
<tr>
<td>Model for computational machine</td>
<td>obtain a general cost model</td>
<td>(how many users utilise the service or application at one time)</td>
</tr>
<tr>
<td>Model for data transfer into the</td>
<td>Designed for applications that run in a public</td>
<td>Do not take into account the Burstiness of the services, and unallocated</td>
</tr>
<tr>
<td>cloud</td>
<td>and private cloud</td>
<td>resources</td>
</tr>
<tr>
<td>Model for data transfer out to the</td>
<td>Take into consideration the usage pattern of the</td>
<td>Cost for data communication is not considered</td>
</tr>
<tr>
<td>cloud</td>
<td>server</td>
<td></td>
</tr>
<tr>
<td>Cost model for hybrid cloud</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Server-usage model</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Server-burst model</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Pool-burst model</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

5 Pricing strategies in cloud computing

Another important aspect in cost models is represented by pricing mechanisms.

There are three categories of pricing mechanisms: fixed pricing, differential pricing and market pricing. In the fixed pricing category are the following billing options: pay-per-use, subscription and list price. For differential pricing are the following: service feature dependant, customer characteristic dependant, volume dependant and value based. In the market pricing category are three types of pricing mechanisms: bargaining, auction and dynamic market (https://developers.google.com/appengine/docs/billing).

For example Google use a combined approach, for different services, where the client can choose a pay-per-use for IaaS service model (the user pay CPU/hour), or in the case of SaaS Google offer a subscription model. The volume dependant pricing mechanism is used especially in cloud storage, where the client can have a better price if store a larger amount of storage (Niu et al., 2012).

The authors of Macías and Guitart (2011) propose a genetic algorithm. They start with a naive pricing function that evolves to a pricing function that offers suitable prices in function of the system status. The proposed model is evaluated by comparison with the following pricing strategies: fixed pricing, random pricing, utility maximisation price. The results obtained are shown in Figure 4.

As can be seen the cloud provider that use random-pricing (prices are offered randomly, in a uniform distribution, between buyer and seller) strategy is most inefficient. In case of fixed pricing (offered prices are the 5% between the minimum price that the provider can offer to not lose money and the maximum price that the client can pay to get benefit by buying the service) the revenue of the provider increase linearly
with the number of tasks. In conclusion Figure 2 shows that the provider using genetic algorithm gets the highest revenue in most of the scenarios. When the maximum number of tasks is high, both solutions are similar.

Figure 4  Comparison of revenues between four types of pricing. A provider with a flexible genome (200 chromosomes and 6% of mutations) is used (see online version for colours)

Source: Macías and Guitart (2011)

6 Discussion and conclusions

From the informations presented above we conclude that a major problem with cost models today are that there is not a general optimised model for estimating the costs, and is difficult to estimate cost factors, and their variation over time.

At data centre level cost are concentrated in servers, power infrastructure, networking and requirements. A low utilisation of this resources leads to very low efficiency and business loss. Also the power consumption has a very important role in the efficiency of the data centre and the reduction of costs. Geo-diversifying the location of data centres (Peterson et al., 2011) can improve performance and increase reliability in the event of site failures, and also reduce the costs, by buying cheaper labour for example (Greenberg et al., 2008).

A cost model that take into consideration the characteristics of applications or service such as: the data pattern, the data transfer, the average and peak utilisation (Yao et al., 2012), would be more realistic one and help to estimate more accurate the cost for running in the cloud. Also a good estimation of variable cost factors is needed; contrary the cost model will give poor results and estimations.

An important aspect is represented by the major cost factors and their proper estimation, since any error will have a major impact on the accuracy of the overall estimated cost. To be more precise, accurately focusing on major cost factors is suggested, since any error in their estimates has a large impact on the accuracy of the
overall cost estimation. For instance, in many cases, 31% of cost of data centres comes from labour, 30% from servers, and 25% from cooling (Greenberg et al., 2008). A low-quality estimate on those cost factors has a large impact on the overall cost.

References


Websites
http://en.wikipedia.org/wiki/Net_present_value
https://developers.google.com/appengine/docs/billing