Optimizing Resources in Cloud, a SOA Governance View

Marin Litoiu*, Milena Litoiu**

* York University, Canada
mlitoiu@yorku.ca
**CGI Toronto, 150 Commerce Valley Drive West, Markham, Ontario, L3T 7Z3
milena.litoiu@cgi.com

Abstract. This position paper proposes a governance model for efficient power and computing resources usage in cloud computing environments. We show how optimization of resource allocation can achieve substantial cost reduction. Then we show the governance that needs to be in place and the steps that ensure an efficient optimization.

Keywords: Web services, Service Oriented Architecture (SOA), SOA maturity, SOA Governance, organizational change, standardization, methodology, software development life cycle (SDLC).

1 Introduction

In a changing economic climate IT is confronted with new challenges which require aligning the its priorities to new business needs: reducing cost, becoming more environmentally-friendly are just examples that require companies to have a second look at how their IT departments and IT infrastructure operate. At the same time, the heavy reliance on IT increases the importance of proper compliance to policies, guidelines, limits, etc; adjustments in governance are likely needed to realize these desiderates. In order to react to environmental challenges and constrains, increased agility is expected from organizations and therefore from their IT. For example, responsiveness to changing regulations in the financial sector becomes more important than ever. Banks need to quickly assess their exposure to current financial events. They need to quickly answer questions related to the impact of the current economic climate on their organization, and this is not always trivial. This is because the information is usually scattered on various systems and formats; at times it is hard to even determine what information is salient to the task at hand. Finding the relevant pieces of information in a timely fashion, putting them together and making sense of it all takes time and money. This is just an example of the kind of agility that needs to be achieved virtually overnight in order to respond effectively to current environmental challenges. Similar examples exist for other industries; agility, responsiveness and compliance to regulations are by no means exclusive characteristics of the financial industry.

Governance has also to account for new computing models, among which cloud computing is the most prominent one. In cloud computing, the computations happen remotely, in the “cloud”, that is an Internet hosted data centre (see Fig. 1). To achieve economies of scale, a data center hosts a large number of applications. Consolidating computing, storage and network resources and sharing them across many applications are ways to reduce cost and power consumption. That is possible through two emerging technologies: Virtualization and Software as a Service (SaaS).

- Virtualization is a process of substitution in which a physical resource is substituted by many logical (virtual) resources. The virtual resources retain the properties of the physical resource, but will have less capacity.

- Software as a Service (SaaS) is the delivery of software functionality online, similar to the one installed on a local machine. Depending on the content of the service, a cloud can offer Infrastructure as a Service (raw computing services such as CPU and storage), Platform as a Service (COTS, tools, middleware for developing and deploying applications) and Application as a Service (end user services). Pricing in a cloud is not uniform; there are multiple pricing models as the cloud user can be charged *per usage, per subscription, per transaction.*

This paper shows what is needed in order to achieve resource utilization optimization within a cloud.

The original contributions of this paper are:

- we define the cloud cost saving as an optimization over a receding horizon
- we propose the cloud architecture that enables the optimization
- we define the governance aspects of the cloud deployment and operations
- we show how the instantiation of SOA governance makes cloud optimization possible.

Related work that motivated this paper includes: the first author and his collaborators showed ([2] and [3]) that in the presence of the steady workloads there is a potential for high cost reduction. This reduction is achieved by optimizing the resource allocation to applications in the cloud. Also, [1] makes the case that by consolidating resources, the cost of running a centre is reduced considerably as is the cost of the service experienced by the end user. The present paper takes into account variable workloads and proposes a new optimization technique as well as a SOA governance approach to cloud optimization. There is a concerted industrial and academic effort to devise a general SOA governance model [5]. Governance models are supposed to lay a foundation of understanding and provide generic approaches to governance. As there is no single model for good governance, no matter what governance framework we use, we still need to tailor it for the specifics of the organization and/or the business problem at hand. In this paper we look at how to apply it to cloud computing and resource optimization in cloud.

The remainder of the paper is organized as follows: Section 2 defines the cloud cost savings as an optimization problem; Section 3 shows how cloud optimization is an instance of the SOA governance and Section 4 presents the conclusions.

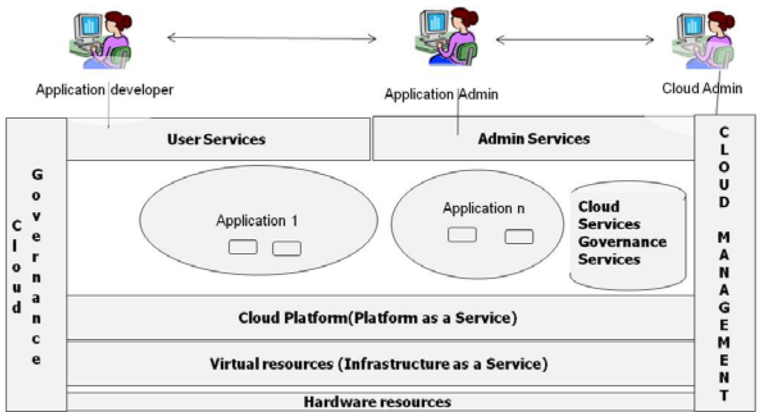

Fig. 1. A cloud overview

2 Cloud Costs and Penalties

In general, the profit of a cloud owner can be expressed as

Profit_cloud = Revenue (Price_application, #Applications) – COST_cloud **(1)**

The cost of cloud (COST_cloud) is proportional with the number of resources run; it will be detailed later. Revenue depends on the pricing schema and, in general, it is directly proportional with the number of applications(#Applications) in the cloud and the price charged per application. The price charged per application (Price_application) is a function of the number of resources used by the application (#Resources), the price per resource (PricePerTimeUnitPerResource) and the total time the resource is used (Time) as shown in equation (2) below.

Price_application=#Resources * Time * PricePerTimeUnitPerResource **(2)**

The equations above reveal that the cloud owner and the customer of the cloud have conflicting interests. The revenue of the cloud owner and hence his profit depend on the price charged per application (Price_application) so the cloud owner will like that factor as high as possible. On the other hand, the customer, who owns the application, would like to minimize the price per application (Price_application). Since the price per application depends on the number of resources (#Resources), the customer and the cloud owner can use the number of resources as a reference term. The only problem is that the number of resources per application varies with the application's variable workload, and the promise is that the cloud will automatically detect the need (or lack of) for resources. Therefore a rigorous mechanism should be in place that accounts for resource consumption based on workload and time. At the same time, an application running in a cloud has to meet the Service Level Agreements (SLAs), the contractual obligations between the cloud owner and the application owner. SLAs define quality metrics thresholds, such as maximum response time and minimum throughput that the application should achieve under variable workloads. If SLAs are breached, the cloud owner pays penalties.

The section below explains in more detail a proposed cost minimization solution implemented by the cloud owner. This solution can be shared with the application owner and be subject of a governance plan between the two parties.

2.1 Model Predictive Control (MPC) and Optimization

We look at how we can simultaneously minimize two components of the cost:

(a) the penalties P for breaching the SLA agreements

and

(b) the cost C of virtual resources allocated to the applications running in the cloud

$$J=P+C \tag{3}$$

The goal is to minimize the cost J over a time window T. We express that as a Model Predictive Control (MPC) multivariable optimization algorithm that uses:

- an internal dynamic model of the cloud and its application
- a history of past control optimization actions
- a cost function J over the receding prediction horizon, to calculate the optimum resource allocation in the cloud subject to technological and operational constraints.

The optimization can be detailed as:

$$\text{Minimize } J = \int_{t}^{t+T} \left(\sum_{i=1}^{N} w_{xi}(r_i - x_i) + \sum_{i=1}^{N} w_{ui}u_i \right) dt \tag{4}$$

subject to:

- technological constraints (e.g. low/high limits on number of resources, of their utilization limits, power and performance characteristics, etc.)
- policies
- compliance.

The variables in (4) are:

- T: the time horizon for optimization

- N: the number of applications hosted in the cloud
- t: the current time
- x_i = *i* -th SLA variable (e.g. measured response time)
- r_i = *i* -th reference variable (e.g. required response time)
- u_i = *i* -th controlled variable (e.g. virtual resources) provisioned or unprovisioned for application i
- w_{xi} = weighting coefficient reflecting the relative importance of x_i. Large w_{xi} means that it is very important to meet the SLAs for application *i*.
- w_{ui} = weighting coefficient penalizing relative large values for u_i. Large w_{ui} means that the cost of resources is more important than meeting SLAs for application *i*

The first term of equation (4) measures the breach of the SLAs. When r_i is smaller that x_i, there is a violation of the SLAs. The second term of equation (4) measures the cost of resources u_i, for each application 1 to N.

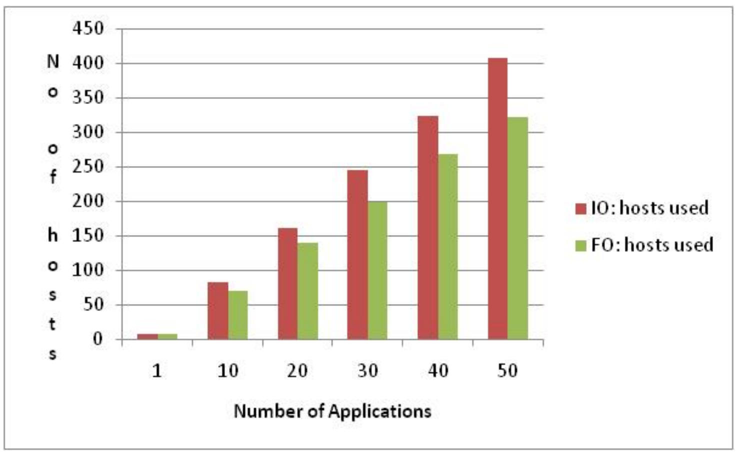

Fig. 2. Optimization results (as reported in [2]), for 2 types of optimization: IO incremental optimization, FO-full optimization

Theoretical results, based on simulation of the cloud infrastructure and of application workloads have shown that there is a huge potential for optimization. Fig. 2 shows two types of optimization as reported in [3][2]. Optimization is done for a time unit (it is not over a horizon T as this paper proposes) and minimizes the number of hosts (not the overall cost J as this paper proposes). In Fig. 2, incremental optimization deploys the new applications incrementally, on resources left available by the already deployed applications. Full Optimization performs a global optimization across all applications even if that means the redeployment of running applications.

Next section will introduce the necessary cloud governance that needs to be in place for the equation (4) to be properly implemented and accounted for. We will follow the SOA governance model.

3 Governing the optimization process

According to The Open Group's SOA Source book [5], any SOA Governance model (Fig. 3) is made of:

- Technology (including infrastructure and tools that enable governance)
- People (organizational structure), including roles and responsibilities
- Processes (including governed and governing processes)

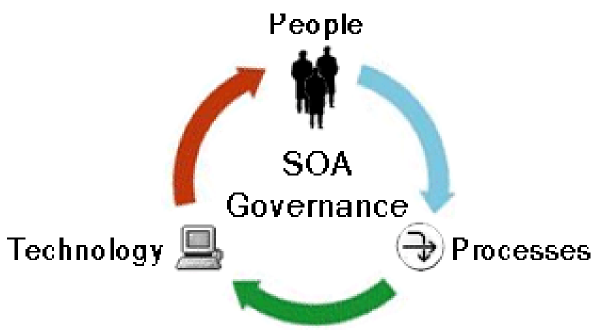

Fig. 3. SOA Governance Model

Once the key governance is put in place (including roles and responsibilities, policies, standards, guidelines, etc), it needs to be actively managed (e.g. compliance measurement, change management, exception and appeals mechanisms, etc.).

As the remainder of the paper refers uniquely to the governance processes put in place to ensure resource optimization in the cloud, these processes are assumed to be in sync with the general governance framework in place, aligning with the business governance principles generally adopted by the cloud.

3.1 The technology

Fig. 4 shows the building blocks of the cloud that can implement the optimization problem expressed in (4). First, the cloud has a monitoring service that traces quality measures x_i, workload per application f_i (number of users or arrival rates) as well as the resource usage U_i.

The measured values are the input in the optimization block together with the policies, compliance regulations, the weights **w** as well as the Service Level Objectives r_i. The Optimization Engine will compute a solution of the optimization problem, that is a set of values u_i over a finite horizon and communicate the decision to the resource pool. The latter will assign the resources u_i to the application App i. This cycle is repeated continuously, at a rate depending on the applications' dynamics.

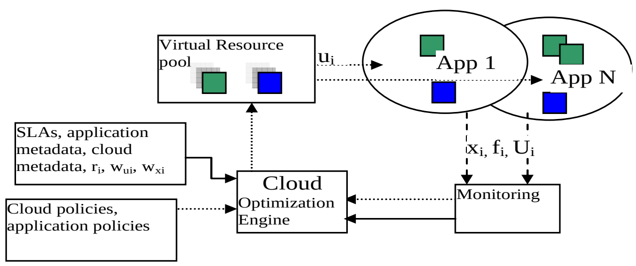

Fig. 4. Building blocks of the cloud optimization

While in theory the optimization and cost reduction is possible, in practice can only be accomplished if processes are in place and if people involved in cloud development and operations have well defined roles and communicate effectively.

3.2 The people

Many roles and responsibilities are involved with the set up and management of governance. They range from Executive and SOA Steering Boards, to Solution and Service Development teams. In this paper we only focus on several roles and responsibilities that are relevant to cloud optimization:

- A steering committee. Its role is to approve the strategic direction as well as make decisions about the cloud governance. Cost optimization should be a criterion for the committee to address, as guidelines and policies are put in place. This steering committee should include the executives, cloud strategists as well as business sponsors from the cloud and its cloud users /customers / SaaS providers. The steering committee makes sure that the cloud as well as the applications running in the cloud are enabled for optimization and have the required capabilities for monitoring and execution of the optimization plans.

- A Cloud Center of Excellence needs to determine the best governing and governed processes (based on best practices), determine compliance checkpoints and metrics as they relate to the optimization.
- A Governance board is responsible for the definition of standards, policies, guidelines as well as the compliance processes. It also defines the exceptions policies, for processes triggered by sub-optimal situations, breached SLAs or unsynchronized applications that preclude cloud optimization.
- Application Developers assemble available services and/or develop new ones if necessary. They provide the monitoring and execution end points needed to measure key performance indicators and to fine-tune the application at runtime. The application metadata, a metamodel of the application allowing the optimization engine to infer relationships among different components of the application and determine possible causes of performance degradation, needs to be managed as well. In most cases, the metamodel takes the shape of a performance model.
- Application Administrator uploads and configures the application in the cloud, sets up the application Service Level Objectives and management policies for the application; an application administrator might update the application metamodel with cloud specific parameters and policies.
- Cloud administrator is in charge with the overall cloud management, according to the cloud policies applicable to all customer applications that share resources on the cloud. A cloud administrator makes sure the applications are sharing the resource pools (Fig. 4) and that the weights **w** reflect company priorities.

3.3 The processes

Processes can be governing (those put in place to ensure a proper level of governance) and governed. The optimization process is a governed process. Other necessary processes refer to the development, deployment, monitoring and change management of the optimization process. Fig. 5 highlights the relationships between these processes.

The optimization process is a governed process; it performs the following activities:

- monitors the resources in the cloud as well as the SLAs,
- analyzes current and future workloads
- determines re- configuration points,
- devises a change plan and
- executes that plan.

This managed process is likely an autonomic process, which rarely requires user intervention.

Development and deployment correspond to the SOA Solution Life-Cycle (SDLC) process [5]. We also have a monitoring process and change management of the optimization process. Some activities occur at the application provider site, others occur at the cloud owner site, yet others are cross-site.

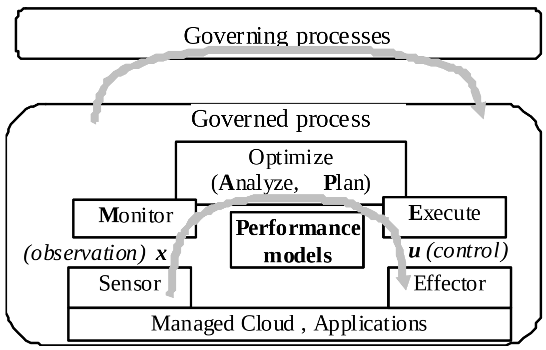

Fig. 5. Governed and governing processes for cloud optimization

The SDLC activities executed at the application owner site are normal SDLC activities with additional constrains imposed by the cloud programming model and optimization activities. They include:

- Requirements gathering that takes into account the cloud environment
- Solution design and development that enables SLAs monitoring , autonomic tuning and provisioning of the application
- Testing in cloud environment
- Deployment in the cloud

Activities at the cloud owner site include:

- Service and Solution Portfolio Management
- Resource optimization
- Resource upgrades
- Change management of cloud services all application depend on
- Communication among cloud roles

Communication processes between cloud and application providers:

- Communication between cloud and SaaS providers
- Communication between SaaS providers and end users
- Communication with the steering board
- Communication for change management

Done properly, this combination of processes, people responsibilities and technology choices help minimize resources costs, as well as increase agility/responsiveness to changes in the environment.

4 Conclusions

This position paper presented a cloud computing optimization problem and gave examples of governance mechanisms that needed to be put in place to minimize an objective function. We considered a cloud environment that automatically provision hardware and software resources to customer applications; further, customers are charged per usage, per number and type of resources used. Since the number of resources is dynamically computed by the cloud, governance processes have to be put in place to make sure that dynamic allocation of resources is optimized. This means that the cloud owner should NOT overprovision the applications; optimization ultimately is meant to lead to cost reductions that can be transferred to customers.

The proposed optimization and the surrounding governance fit into exiting governance models like the SOA governance technical standard from The Open Group, customized and enhanced to handle the cloud optimization. We discussed the technology, people and processes specifics that make the cost reduction possible.

We can envision several barriers to the adoption of the proposed models:

Firstly, although there are a number of SOA Maturity Models and a few methodologies to determine the maturity of the SOA adoption for an organization, determining the right level of governance is still not trivial. This is because there is no "one size fits all" recipe of SOA governance success. Also, this is because every organization is different. The governance levers that need to be in place in every case largely depend on the organization's specifics.

Secondly, other factors like the organizational structure, culture and education, as well as the level of resistance to change are to be considered when adopting this approach. Poor alignment between business needs and IT priorities is a clear indication that something needs to change in order to improve the organization's efficiency.

To make the proposed model successful, several enablers need to be put in place:

Firstly, a robust and meaningful set of metrics that are linked to business priorities.

Secondly, a periodic tracking of the key performance indicators like cost, agility, reuse, time to market, customer satisfaction and retention. For example, changes in governance can improve key agility indicators (KAI) like time to market, etc. Periodic measurements are used to compare actuals to baselines in order to make sure that processes stay on target and business goals are consistently achieved.

REFERENCES

[1] Michael Armbrust et al., Above the Clouds: A Berkeley View of Cloud Computing, *Technical Report*, http://www.eecs.berkeley.edu/Pubs/TechRpts/2009/EECS-2009-28.pdf, May 2009.

[2] Li J., Chinneck J., Woodside M, Litoiu M., "Fast Scalable Optimization to Configure Service Systems having Cost and Quality of Service Constraints,"

Proceedings of IEEE International Conference on Autonomic Systems, Barcelona, June 2009.

[3] Li J., Chinneck J., Woodside M, Litoiu M., Gabriel Iszlai, Performance Model Driven QoS Guarantees and Optimization in Clouds, *Proceedings of ACM/IEEE Symposium on Cloud Computing Challenges*, Vancouver, May 2009.

[4] Dirk Krafzig,Karl Banke,Dirk Slama, Enterprise SOA: Service Oriented Architecture Best Practices, Prentice Hall, 2008.

[5] The Open Group's Service-Oriented Architecture (SOA) Book, http://www.opengroup.org/projects/soa-book/, July 2009.

[6] IBM, SOA Governance, White paper, http://www-304.ibm.com/jct03001c/software/solutions/soa/gov/lifecycle/, July 2009.

[7] Software AG White Paper, http://www.softwareagusa.com/media/pdfs/Getting_Started_Guide_SOA_Success.pdf, July 2009.

[8] SOA Governance- IBM's approach ftp://ftp.software.ibm.com/software/soa/pdf/SOA_Gov_Process_Overview.pdf, July 2009.

[9] Kontoginannis K., Lewis G., Smith D., Litoiu M., Muller H., Schuster S., Stroulia E., "The Landscape of Service-Oriented Systems, A Research Perspective," *Proceedings* of *Development for Service Oriented Architectures (SD-SOA 2007); 29th IEEE/ACM Software Engineering Conference on Software Engineering (ICSE 2007)*, pp. 84-91, Minneapolis, MN, USA; May 2007.