Adaptive and Business-driven Service Placement in Federated Cloud Computing Environments

Luca Foschini
Dipartimento di Informatica - Scienza e Ingegneria
University of Bologna
Bologna, Italy
luca.foschini@unibo.it

Mauro Tortonesi
Department of Engineering
University of Ferrara
Ferrara, Italy
mauro.tortonesi@unife.it

***Abstract*—The emergence of large-scale federated Cloud computing environments and of dynamic resource pricing schemes presents interesting saving opportunities for service providers, that could dynamically change the placement of IT service components in order to reduce their bills. However, that calls for smart management solutions able to respond to pricing changes by dynamically reconfiguring IT service component placement in federated Cloud environments so to enforce high-level business objectives defined by the service providers. This paper proposes a novel adaptive and business-driven IT service component reconfiguration solution based on what-if scenario analysis and on genetic-algorithm optimization. Our solution is able to model complex Cloud computing IT services and to evaluate their performance in a wide range of alternative configurations, by also detecting the optimal placement for their components. The paper presents the experimental evaluation of our framework in a realistic scenario that consists of a 2-tier service architecture with real-world pricing schemes. The results demonstrate the effectiveness of our solution and the suitability of business-driven IT management techniques for the optimal placement of service components in federated Clouds.**

Keywords—Cloud computing; Federations; Business Driven IT Management.

I. INTRODUCTION

The emergence of Cloud computing infrastructures of fully interconnected data centers offering their computational resources on a pay-per-use basis is a great challenge and opportunity to develop new services and IT management systems on a global scale; that is especially true for federated environments that include multiple data centers and different Cloud administration entities. These novel federated Cloud systems are typically characterized by ever-changing service loads and require highly flexible and self-adaptive management solutions to dynamically and continuously supervise and handle the current status of executing services and support components by initially placing, and then possibly moving them depending on currently monitored working conditions and possible economic advantages.

Many Cloud providers are already introducing dynamic resource pricing schemes, such as Amazon EC2 Spot Instance Service [1], where resource prices are not only different between different Clouds, but also fluctuate over time. Those new dynamic pricing schemes pave the way to significant operation cost reductions. At the same time, to take full advantage of those new possibilities it is necessary to solve several open management issues spanning from virtualization issues, such as Virtual Machine (VM), storage, and network migration, to optimal resource (VM, storage, networking, etc.) placement computation, from large-scale federated Cloud monitoring to standardization and interoperability of the different Application Programming Interfaces (APIs) adopted by various Cloud providers, and so forth.

Among those technical challenges, this paper will focus on the specific problem of enabling adaptive and intelligent service placement computation in large-scale federated environments. This specific area has already been explored by several works along different directions that all share the common goal of balancing the Cloud provider internal objectives, namely minimizing infrastructure/hardware resource usage and granting Service Level Agreement (SLA) agreed with final service providers, and the external objectives of service providers using the Cloud, typically cost minimization and fulfillment of SLAs. However, most of the works available in the literature tend to focus more on internal objectives, such as, minimizing energy consumption either in a single data center [2, 3, 4] or in federated ones [5, 6, 7], and balancing incoming load to prevent resource shortages [8, 9], to reorganize service schemes and to deliver agreed SLAs by typically considering traditional IT performance metrics. The optimization of external objectives, instead, apart a few specific seminal studies [10, 11], is still a widely neglected practice. I this context we claim that especially business-driven approaches would be highly beneficial for service providers to obtain the best quality-cost service tradeoffs.

To bridge that gap this paper proposes a novel adaptive and business-driven service reconfiguration solution for federated Cloud computing environments. Our solution presents several core original elements. First, it considers business objectives for the service component placement optimization, thus enabling IT managers to identify the configuration with the lowest business impact. In addition, our solution leverages on a simulative approach and on *what-if scenario analysis* to reenact Cloud computing IT services under different configurations, thus providing a much better capability to accurately capture peculiar behavior of real-life IT services than analytic methods. Finally, our solution leverages on genetic algorithm-based optimization for the robust and resilient exploration of the large and dynamically-changing

978-3-901882-50-0

space of possible IT service component placement configurations. The seminal experimental results presented in this paper, based on a relatively realistic 2-tier service scenario and on real costs for a federated Cloud computing environment implemented on top of 3 different Amazon EC2 data centers, demonstrate the effectiveness of our solution.

The remainder of this paper is organized as follows. Section II introduces needed background material and related work in the literature. Section III presents our framework and outlines its main components; Section IV details the business-driven service model; and Section V provides additional details about the adopted cost functions and our genetic algorithm approach. Finally, Section VI shows collected experimental results, and then conclusions and directions of future work end the paper.

II. Service Placement in Cloud Computing Environments

This section provides some needed definitions that we use in the remainder of the paper and introduces our reference scenario to clarify the main goals of our management solution. The section also presents an overview of the literature on related research topics.

A. Business-driven Service Placement

Adopting a widely used agreed definition, we distinguish three main types of actors in Cloud systems: *service users*, *service providers*, and *Cloud providers*. Service users are the final clients that require access to particular online services. Service providers seize the opportunity to build new services, in order to increase their economical revenue, and tend to externalize the execution of their own services to avoid the deployment of costly private IT infrastructures. Finally, Cloud providers are usually big players, such as Amazon, Google, and IBM, that offer service providers all the system resources needed to execute their services on a pay-per-use basis.

In this paper, we address federated Cloud scenarios where we assume different Cloud providers join their efforts and data centers to mutually benefit of underutilized resources and to support service movement from one Cloud to another, by also letting service providers choose, place, and migrate their services to the most convenient Cloud. Let us briefly note that this scenario is still unavailable and several technical issues, from reciprocal authentication and security to interoperability still need to be solved. At the same time, these novel federated platforms will bring several advantages not only to Cloud providers, but also to service providers and to final service users.

Here, we are particularly interested to focus on the external perspective of service providers to study both cost, (e.g., price minimization with obvious cost savings also for the final service users) and business gains (e.g., fault-tolerance and reliability, vendor lock-in, etc.) in these highly dynamic and open federated ecosystems. As a result, we focus on the realization of a management solution that continuously monitors the pricing schemes proposed by Cloud providers and dynamically reconfigures the IT service component placement in order to align it with Cloud pricing and the business objectives defined by service providers.

However, evaluating the performance of an IT service configuration is a very difficult task because IT-level metrics are plenty and the use of automated performance optimization process would force to consider multi-criteria decision making methods. Large IT systems, such as federated Cloud systems further increase the complexity of these management tasks. Therefore, we claim the relevance of *business impact analysis* techniques that represent a significantly better criterion to adopt for the performance optimization of IT support organizations. In fact, business impact-driven optimization aims at minimizing the adverse impact of service disruptions on the business, by considering all the costs attached to critical incident occurrences.

In addition, business impact represents a more convenient way to evaluate IT services from the optimization perspective because evaluating IT services through their business impact allows considering single metrics in the optimization process, by significantly simplifying it compared to more complex multi-objective optimization methods. In addition, business impact analysis does not require to consider explicit constraints on allowed IT service configurations (e.g., a maximum threshold for service response times), because undesired service configurations will have a high business impact and therefore will be automatically ruled out by the optimization process.

B. Related Work

Our work is focused in business-driven service placement in federated Clouds; currently, there are many works in literature addressing several related aspects. Here, we consider two very close areas mainly focusing on internal Cloud provider objectives, and a third one including seminal business-driven management solutions that focus more on service provider external objectives: i) *power-efficient service placement* that adopts greedy consolidations to minimize power consumption due to hosts, network, and cooling systems; ii) *service placement for load-balancing* with the opposite goal to prevent resource shortages and frequent VM relocations; and iii) *business-driven service placement* that considers not only technical parameters and SLAs, but also higher-level business drivers. In the following, without any pretence of being exhaustive, we briefly survey a selection of most significant state-of-the-art efforts along these directions briefly presenting, for each considered system, its main pros and cons.

Power-efficient service placement solutions deal with both single (intra) and federated (inter) data center deployment scales. Starting with the first ones, Mistral is a recent proposal that employs A*-search techniques to optimize both VMs performance and power consumption and to find complex reconfiguration actions; shown experimental results, collected in a real testbed, make this proposal extremely solid [2]. With a similar practical perspective, authors in [4] thoroughly assessed how traffic demands between VMs deployed on the same physical influence host CPU and memory overhead, to avoid excessive overhead due to local communications. Finally, [3]

focuses on reducing the total energy consumption associated with network elements, namely powered-on switches and patches. Focusing solutions for large-scale federated Cloud systems, service placement in Cloud computing has his roots in the Grid computing. Authors in [5] address optimal locations of data centers in the transport network and automatic demand provisioning to reconfigure the network by virtual topology mapping. [6] proposes a new Mixed Integer Linear Programming (MILP) formulation for energy-efficient Cloud network design; while [7] extends that model to take into account new additional constraints, such as the nearest data center(s) and intra-and-inter energy-efficiency objective in VM-placement.

With regards to service placement for load-balancing, different works addressed such topic by considering SLAs and technical requirements (e.g., local CPU and memory at each physical host); [8] surveys selection of solutions in the field. For instance, in [12] authors consider Cloud data centers with server and storage virtualization facilities, and strive to increase load balancing at multiple layers, including servers, switches, and storage, by solving this problem as a multi-dimensional knapsack problem. More recently, some seminal works have started to concentrate also on Cloud networking because network represents a significant bottleneck, both in single and federated data centers. [13] focuses on the problem of network-aware VM placement with the goal of reducing the aggregate traffic into the data center (e.g., by co-locating VMs that highly communicate); they introduce a new placement problem, called Traffic-aware VM Placement Problem (TVMPP), that belongs to NP-hard problems, and propose a heuristic approach to solve it in a reasonable time. With a different perspective, authors in [9] define a VM placement problem, called Min Cut Ratio-aware VM Placement (MCRVMP), aimed not only to satisfy predicted communication demands between VMs, but also to be resilient to dynamic traffic time-variations, so to minimize the number of VM relocations.

Let us finish this very brief survey focusing on very recent efforts on business-driven service placement for large-scale federated Cloud environments that are the closest to our proposal. A seminal work in this area is [11] that addresses the management of changes to IT infrastructure and services to satisfy business goals and to minimize costly disruptions on the business, by focusing on an interesting real case study about a 2-tier service. Several other works have recently started offering virtualized resources and services in the form of Virtual Data Centers (VDCs) consisting of VMs connected through virtual switches, virtual routers, and virtual links with guaranteed bandwidth. VDC Planner is a recent proposal that provides a migration framework to improve the success rate of VDC mapping requests while minimizing total VM migration costs [10], thus increasing Cloud provider revenues. With a more external service provider perspective, [14] presents a comparative analysis of the economic models for Cloud computing and traditional in-house IT service delivery; the adopted Net Present Value (NPV) metric allows to evaluate two main alternatives, namely, one for emergent countries and one for established countries, by considering 4 different service types. This proposal is much related to our work, but it still lacks risk-related aspects and the modeling of migration-to-the-Cloud processes; in addition, it does not consider federated Cloud deployments.

Differently from previous approaches, our solution leverages on *what-if scenario analysis* and on *genetic algorithm-based optimization* techniques in order to identify the optimal VM placement configuration within a federated Cloud. The optimization criterion is the lowest business impact according to the current customer set profile and management policies.

III. THE FRAMEWORK

We propose an adaptive and business-driven service reconfiguration solution for IT services in federated Cloud computing environments. The framework continuously analyzes input data in order to respond to changes such as service request load spikes and different business objectives. The framework analyzes the data to build accurate and up-to-date models of both service requests and service processing times. These models are used to reenact the IT service behavior in order to evaluate different service configurations in what-if scenario simulations.

For the sake of simplicity, and without hindering the generality of the proposed approach, our solution currently focuses on VMs as the basic building blocks for the realization of more complex IT services. In other words, the solution proposed in this paper conceptually operates at the Infrastracture-as-as-Service (IaaS) level with the main goal of finding the best placement configuration of the VMs in federated Cloud environments. In addition, our framework leverages on *business impact analysis* in order to evaluate the alignment of service configurations with the business objectives set by the management.

Fig. 1 shows the architecture of our framework. The Demand Monitoring and Demand Model components are, respectively, in charge of continuously monitoring customer requests and of analyzing them in order to build a model of the service request process that could be exploited for the what-if scenario analysis.

The Service Monitoring and Service Model components have a similar function, as they analyze service logs and build a model of the service execution (e.g., service time distribution, current service component placement, etc.), that is essential for the accurate reenactment of the IT service. In fact, the identification of best VM placement for an IT service through what-if scenario analysis needs accurate models used to evaluate how services will behave under different configuration and working conditions.

The Optimization component is the core part of our framework: it is in charge of reenacting the Cloud computing IT service and of evaluating possible alternative service placement configurations. The what-if scenario analysis and the business impact analysis functions are realized by two dedicated sub-components with the same names.

The Configuration Management and Policy Management components represent the interface that the framework

provides to final users, namely, service providers. These components are, respectively, in charge of enabling the user to provide a configuration of the Cloud computing environment (e.g., number of data centers, service model, etc.) and of the optimization policies to apply (e.g., business objectives, parameters for the optimization algorithm, etc).

Finally, the Decision Making component is in charge of selecting the best IT service placement configuration, according to the user preferences and the output data provided by the Optimization component. The Decision Making component would be ideally connected to an actuator capable of automatically putting the service configuration in place.

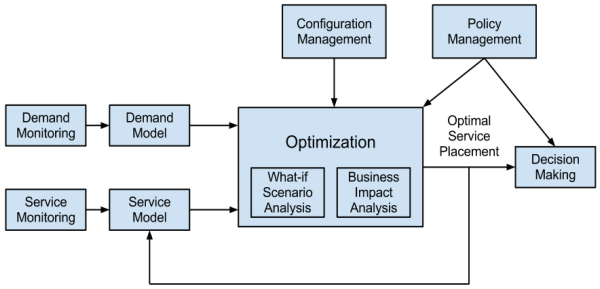

Fig. 1. The architecture of our framework.

IV. Modeling, Reenactment, and Business-impact Assessment of IT Services

Modeling IT services is a very challenging task, and requires to consider tradeoffs in model complexity. In fact, modeling customer requests involves several parameters. First of all, in order to correctly reenact the workload on the Cloud computing IT service, demand models should accurately capture the inter-arrival time patterns in service requests. As we demonstrated in [15], customer service requests typically have non-trivial patterns difficult to accurately capture and call for the adoption of sophisticated techniques based on non-parametric statistics.

In addition, the accurate reenactment of Cloud computing services requires to account for metrics that have an impact on performance. For instance, for SLA evaluation we need to calculate the latency involved in customer requests that depends from the relative position of customer location and service VM instance location. As a result, demand models should generate service requests with attributes such as the request's (the customer's) originating locations.

Finally, as the customer base might change, service request behavior might significantly change over time. That calls for the periodic re-evaluation of the demand model, suggesting the adoption of a continuous process for the analysis of the service.

To solve the above modeling issues, a possible approach is the one followed by [10] that considers Cloud computing services as built on top of fully independent components. This approximation permits to significantly limit the complexity of the service placement framework, enabling the adoption of analytic methods to identify the optimal service placement.

However, this is not well suited for the modeling of complex services, usually based on 2- or 3-tier architectures, often adopted in Cloud computing services. In order to consider complex services, that are likely to benefit the most from optimal component placement solutions, we need to capture the relationships between service components and to measure the impact of a component reallocation to a different data center on the whole service performance. More specifically, we need to consider different types of components, each one modeled as a queue with an incoming service request rate and service time distribution.

On top of these components, we define services as the result of interactions between the basic building blocks (namely, VMs). This is very similar to business process modeling in Service Oriented Architectures (SOA). As a result, we can leverage on service composition concepts such as the ones from Business Process Execution Language (BPEL) standard and similar solutions to define complex service workflows.

In addition, we need to consider the latencies involved in the interactions between different components. As a result, in our model we consider: *data centers* that can host service components; and a *mapping function*, defined as *L(dc1,dc2)*, that provides the latency of communications between components running in data center *dc1* and those running in data center *dc2*. Since the reenactment of this service model cannot be performed with analytic methods due to its intrinsic complexity, it calls for simulative approaches [15, 16].

V. Optimal IT Service Placement

This section delves into the realization details of our framework. First, it introduces the main cost estimation phases, then it focuses on the proposed cost function, and finally it reports some details about the adopted decision making approach.

A. Cost Estimation Phases

The cost estimation of an IT service involves 3 main phases: *new IT service configuration and deployment cost estimation*, *Service Level Objective (SLO) penalty estimation*, and (where applicable) *IT service reorganization costs*.

In the first phase, our framework calculates the costs according to an input table reporting the fees for federated Cloud providers taken into account. These costs are based on real data taken from all major Cloud providers (Amazon, Microsoft, IBM, etc.) and may change over time and for different geographic locations, depending on data center location, by being influenced by several factors (energy cost, local security conditions, etc.).

The second phase, instead, evaluates whether switching to the new IT service configuration would cause the service provider to incur in SLO penalty violations. To this end, our tool requires the user to provide information about the SLO penalties that the service provider stipulated with his customers.

Finally, the third phase is triggered only when necessary and it is needed in order to calculate costs related to IT service

reorganization. For instance, in case of Cloud computing services, migrating VMs between different data centers might result in traffic costs as well as costs related to performance loss; moreover, it might be required to stop the service during the migration, and that would also contributed to increase the costs related to temporary service unavailability, and so forth.

B. Cost Function

From a theoretical perspective, the service placement problem can be formalized as the following optimization problem:

$$\min BI(x) \quad subject\ to\ x \in S_{PC} \tag{1}$$

where the variable x represents the IT service configuration; the set S_{PC} represents the space of possible IT service configurations to explore; and BI is a cost function that evaluates the business impact of the IT service configuration x.

As the BI function is very complex, we cannot leverage on gradient-descent-based techniques for its optimization. Instead, we need to consider the adoption of meta-heuristics designed for large-scale optimization, since the space to explore is typically very wide. In particular, we believe that the complex nature of this class of optimization problems calls for the adoption of *genetic algorithms*.

In fact, genetic algorithms have the very desirable property of being resilient to changes in the optimization function as it may occur in our problem. In addition, genetic algorithms can be modified to control the number of times that the optimization function is evaluated for every optimization cycle. Finally, genetic algorithms can be easily modified to take advantage of parallel execution, such as map/reduce-based solutions in Cloud computing environments.

C. Decision Making

The evaluation of different configurations and the detection of the optimal one needs to be followed by a decision making phase with the purpose of choosing whether the new configuration should be put in practice or not. In fact, service reconfigurations are time- and resource-consuming, and should be performed as seldom as possible.

As a result, there is the need to put in place solutions that limit the frequency of service placement reconfigurations. For instance, in order to avoid frequent reconfiguration, we enforce a minimum time interval between reconfigurations. Similarly, to avoid ping-pong effects, we allow service reconfigurations only when the difference between the (expected) business impact of the current and the new configuration exceeds a pre-configured threshold, thereby effectively implementing a hysteresis-based reconfiguration process.

VI. Experimental Evaluation

We realized a prototype implementation of the service component placement framework described in the previous sections in the Ruby (http://www.ruby-lang.org/) programming language. We used our prototype to evaluate the behavior of our service placement framework in a limited but significant test scenario that attempts to capture the most critical aspects of service placement in federated Cloud computing environments.

More specifically, we focused on the problem of optimally placing the components of a Cloud computing IT service. We considered a simple service workflow that we believe captures the behavior of a 2-tier architecture Cloud computing IT service with a reasonable approximation. Our model implements service components as G/G/1 queues and considers 2 different component types. Service requests are first directed to a Level 1 component that deals with the first part of the service process, then forwarded to a Level 2 component that finishes servicing the request, and finally a response is returned to the customer. Service times are modeled as normally distributed random variables: Level 1 service times have a mean of 9 milliseconds and a standard deviation of 1 millisecond, while Level 2 service times have a mean of 12 milliseconds and a standard deviation of 2 milliseconds.

We assume that each service component requires a dedicated VM instance in order to run. In addition, we assume that Level 1 components can fit into a "medium" size VM, while Level 2 component require a "large" size VM, by considering Amazon EC2 pricing scheme (see Table 1 below). For simplicity, we consider architectures with an automatic data replication function between the VMs running in different data centers already in place.

We modeled request inter-arrival times as a Pareto-distributed random variable, with a mean of 1.5E-4 seconds (which corresponds to 6666.67 requests per second) and a location of 1.2E-4 seconds. In fact, Pareto distributions are widely adopted in research literature to model inter-arrival times of service requests [15, 16]. We also assumed that requests were uniformly distributed among 3 data centers (namely, Amazon US, Japan, and Brazil). Finally, we modeled latencies in the communication between customers and the Cloud data centers by assigning to each request a random communication latency, sampled from a (truncated) Gaussian distribution with a mean of 10 milliseconds and standard deviation of 2.5 milliseconds.

For the cost evaluation, we considered the realistic pricing model shown in Table I: these figures represent real market prices for the Amazon EC2 Cloud [17]. We considered neither data transfer costs nor any additional per-request cost because these metrics are independent of service component placement.

TABLE I. Pricing for Medium and Large size VMs in Amazon EC2 Data Centers. Figures are represented in USD / hour.

	Amazon US	Amazon Japan	Amazon Brazil
Medium	0.160	0.184	0.230
Large	0.320	0.368	0.460

We also adopted a rather simple business impact model that, besides VM pricing, considers only SLO violation penalties. More specifically, we consider a 500$ penalty in case the average time to service a request increases above 50 milliseconds.

Finally, for the optimization process we considered a traditional genetic algorithm that implements a binary tournament selection phase and a reproduction phase based on point mutation and one-point crossover. We used conservative values for crossover probability (0.98) and point mutation probability (1 / bitstring length) parameters, and selected a population size of 128. We adopted a rather straightforward bitstring representation of the component placement state, as depicted in Fig. 2. More specifically, our representation adopts bitstrings which are divided in as many sections as the number of data centers to consider. Each of these sections is further divided in other subsections, one for each service type to consider, whose length is the number of VMs of the given type that can be allocated in the corresponding data center. A bit set to 1 in section x and subsection y therefore represents the allocation of a VM to implement service type y in data center x. While this scheme leads to relatively large bitstring sizes – and, as a result, longer convergence times – it allows for an easily understandable representation of the service placement configuration.

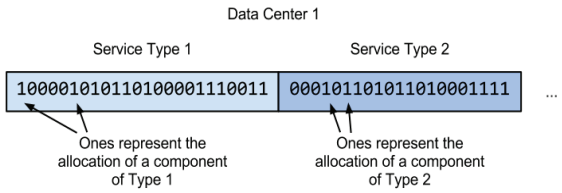

Fig. 2. Bitstring representation of service component placement state.

The configuration with the lowest business impact that we obtained (1746.72 $/day) from this experiment is depicted in Fig. 3, which shows the number of VMs allocated for each data center and service type.

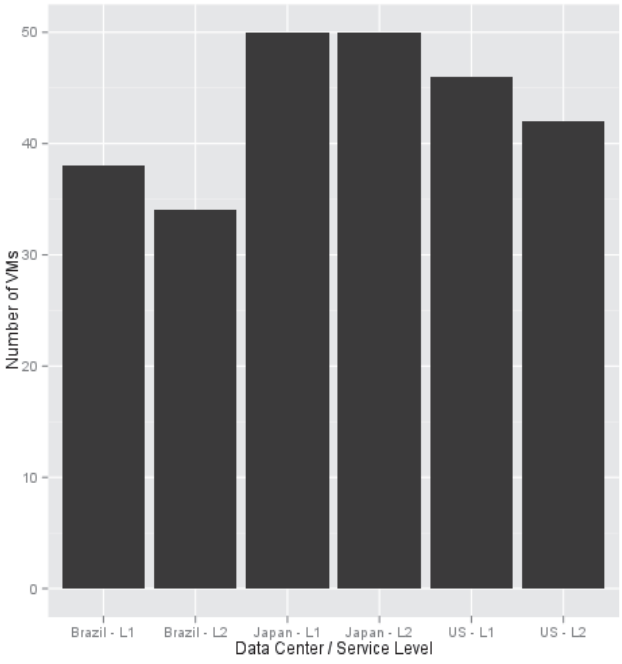

Fig. 3. VM allocation in the first experiment.

To evaluate the response of our placement framework to a pricing variation, we then changed the pricing scheme to simulate a price drop in one of the data center and ran a second experiment. More specifically, we considered a 5% decrease in the prices of the Amazon Brazil data center.

The new configuration selected by our prototype presented a lower business impact (1641.12 $/day) than the one from the previous experiment. This demonstrates that the prototype was capable of dynamically adapting and finding a better configuration after the pricing scheme change. Fig. 4 presents the resulting VM allocation for the second experiment.

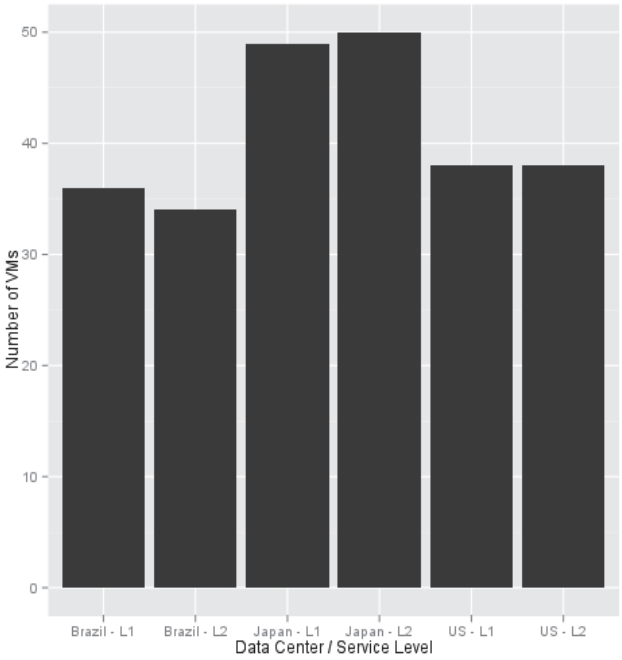

Fig. 4. VM allocation in the second experiment.

VII. Conclusions and Future Work

The emergence of dynamic resource pricing schemes, such as Amazon EC2 Spot Instance Service, calls for adaptive service placement schemes able to detect changes in operating conditions and to dynamically reconfigure IT services accordingly. Business-driven IT management approaches represent a perfect match for the problem of service component placement in federated Cloud computing environments. In fact, the pay-per-use pricing schemes of Cloud computing providers suggest and significantly facilitate the adoption of business-driven optimization techniques for IT services.

In this context, the results shown in this paper demonstrate that business-driven IT management techniques based on what-if scenario analysis represent a very promising avenue of research. We realized a prototype of the proposed framework and we employed it to draw seminal experimental results about a simulated 2-tier service in a federated Cloud computing environment. The prototype was capable of evaluating many alternative component placement configurations for the simulated IT service, and to dynamically respond to changes in pricing.

Based on these promising results, we are already working to further expand our framework by focusing on several directions. First of all, we are investigating possible refinements of the service model to implement more complex service workflows and to express the effects produced by VM migrations (and perhaps dependencies on data shared by different components). In addition, we are running extensive tests to confirm the effectiveness of genetic algorithm-based optimization and to evaluate possible alternative meta-heuristics. Finally, we are studying how to integrate our automated framework within a real IaaS support based on the OpenStack Cloud with the final goal to perform suggested service placement reconfigurations without requiring any human intervention.

REFERENCES

[1] Home page of Amazon EC2 Spot Instances: aws.amazon.com/ec2/spot-instances/

[2] G. Jung, et al., "Mistral: Dynamically Managing Power, Performance, and Adaptation Cost in Cloud Infrastructures", in *Proc. of the IEEE 30th International Conference on Distributed Computing Systems (ICDCS'10)*, pp.62-73, 2010.

[3] V. Mann, et al., "VMFlow: leveraging VM mobility to reduce network power costs in data centers", in *Proc. of the 10th international IFIP conference on Networking (NETWORKING'11)*, pp. 198-211, 2011.

[4] A. Corradi, M. Fanelli, L. Foschini, "VM consolidation: A real case based on OpenStack Cloud", *Elsevier Future Generation Computer Systems*, available online 4 June 2012, DOI: http://dx.doi.org/10.1016/j.future.2012.05.012, 2012.

[5] X. Dong, T. El-Gorashi, and J. M. H. Elmirghani, "Green IP over WDM networks with data centers", *IEEE/OSA J. of Lightwave Technology*, vol. 29, no. 12, pp. 1861-1880, 2011.

[6] B. Kantarci and H. T. Mouftah, "Energy-Efficient Demand Provisioning in the Cloud", in *Proc. of Optical Fiber Communication Conference (OFC)*, pp. 1-3, 2012.

[7] B. Kantarci, L. Foschini, A. Corradi, H. T. Mouftah, "Inter-and-Intra Data Center VM-Placement for Energy-Efficient Large-Scale Cloud Systems", in *Proc. of IEEE GLOBECOM Workshop on Management and Security technologies for Cloud Computing (ManSecCC),* pp. 1-6, 2012.

[8] M. Mishra, A. Das, P. Kulkarni, A. Sahoo, "Dynamic resource management using virtual machine migrations", *IEEE Communications Magazine*, vol.50, no.9, pp.34-40, 2012.

[9] O. Biran, A. Corradi, M. Fanelli, L. Foschini, A. Nus, D. Raz, E. Silvera, "A Stable Network-Aware VM Placement for Cloud Systems", in *Proc. of IEEE/ACM Int.'l Conf. on Cloud, Cluster and Grid Computing (CCGrid)*, pp. 498-506, May 2012.

[10] Q. Zhang, Q. Zhu, M. Zhani, R. Boutaba, "Dynamic Service Placement in Geographically Distributed Clouds", in *Proc. of the 32nd IEEE International Conference on Distributed Computing Systems (ICDCS 2012)*.

[11] S. Hagen, A. Kemper, "Facing the unpredictable: Automated adaption of IT change plans for unpredictable management domains", in *Proc. of IEEE International Conference on Network and Service Management (CNSM)*, pp.33-40, 2010.

[12] A.Singh, M. Korupolu, and D. Mohapatra, "Server-storage virtualization: integration and load balancing in data centers." in *Proc. of the 2008 ACM/IEEE conference on Supercomputing* (SC'08), pp. 1-12, 2008.

[13] X. Meng, V. Pappas, and L. Zhang, "Improving the scalability of data center networks with traffic-aware virtual machine placement," in *Proc. of the 29th IEEE Conference on Information Communications (INFOCOM'10)*, pp. 1154-1162, 2010.

[14] K. Sripanidkulchai and S. Sujichantararat, "A business-driven framework for evaluating cloud computing", in *Proc. of 7th IEEE/IFIP International Workshop on Business-Driven IT Management (BDIM 2012)*, pp. 1335 - 1342, 2012.

[15] C. Bartolini, C. Stefanelli, M. Tortonesi, "Synthetic Incident Generation in the Reenactment of IT Support Organization Behavior", accepted for publication in *Proc. of the 13th IFIP/IEEE Integrated Network Management Symposium (IM 2013) - Technical (main) track*.

[16] H. Qian, C.S. Surapaneni, S. Dispensa, and D. Medhi, "Service Management Architecture and System Capacity Design for PhoneFactor--A Two-Factor Authentication Service", in *Proc. of 11th IFIP/IEEE International Symposium on Integrated Network Management (IM 2009)*, pp. 73-80, New York, June 2009.

[17] Amazon EC2 pricing web page: http://aws.amazon.com/ec2/#pricing