Probabilistic Admission Control for Elastic Cloud Computing

Kleopatra Konstanteli and Theodora Varvarigou
School of Electrical and Computers Engineering
National Technical University of Athens, Greece
Email: {kkonst,dora}@mail.ntua.gr

Tommaso Cucinotta
Real-Time Systems Laboratory
Scuola Superiore Sant'Anna, Pisa, Italy
Email: cucinotta@sssup.it

***Abstract*—This paper tackles the problem of optimum allocation of elastic services on virtualized physical resources by incorporating a probabilistic approach in terms of availability guarantees, which allows for reducing the physical computational resources that are required for elasticity reasons. The resulting probabilistic optimization problem also allows for proper trade-offs among business level objectives. Its output is the set of the admitted services, as well as the allocated computing capacity for each service component that comprise the services on the selected physical hosts. The problem was modeled on the General Algebraic Modeling System (GAMS) and solved under realistic provider's settings that demonstrate the efficiency of the proposed method.**

***Keywords*-admission control; elasticity; cloud computing; optimum allocation.**

I. Introduction

Admission control has the goal of deciding whether a set of services can be admitted in a given infrastructure, and deciding what the optimum allocation of the services to the available underlying resources is, in case of acceptance. This task is far from trivial, since in many cases, admitting a new service increases the risk of already deployed ones failing. On the other hand, a strict acceptance policy results in an increased number of rejections, and consequently reduced revenue.

In the Cloud, each service is considered to be a set of virtualized components, i.e. Virtual Machines (VMs), that are activated according to their workflow pattern each time a user request arrives. At admission control time, the Infrastructure Provider (IP) must consider not only the basic requirements but also the extra requirements that may be added at runtime, defined as elastic requirements. In many cases, the elastic requirements may be quite large compared to the basic ones. For example, given a service with a high variation in the number of users, the VMs that may be required to be added at runtime may be many times multiple of the basic ones. Thus, the elastic requirements play a significant role in the cost of hosting the service, and the IP has a strong interest in investigating the possibility of narrowing the resources that need to be booked for elasticity reasons. At the same time, such an approach may increase the possibility of deviating from the agreed quality of service level, and the imposed penalties may as well outgain the advantages of this approach.

From the IP's perspective, a proper metric expressing the goodness of the found allocation is significant of the cost. However, nowadays an IP's overall acceptance policy may include other factors such as the risk of collaborating with a given Service Provider (SP), and the level of trust between them, as well as other factors such as how eco-efficient a given host is [3].

The problem of allocation of real-time distributed tasks on heterogeneous hosts has been investigated in [5]. In that paper the focus was on deterministic guarantees in hard real-time systems, whereas the approach in this paper focuses on soft real-time services with elastic requirements. Many works exist that address stochastic real-time systems [9], [2], [11]. In prior works of ours [4], [8], the problem of optimum allocation of soft real-time services with probabilistic guarantees was tackledby achieving overbooking of the resources , whereas the present paper focuses on elastic services whose requirements may dynamically change, depending on the dynamically varying patterns of the requests and users.

In more detail, the proposed method incorporates the actual probabilities of requiring extra computational capacity for the services into the admission control test, thus allowing for reducing the physical resources that need to be booked for elasticity reasons. The resulting probabilistic admission control test also allows for trade-offs among business level objectives. The output of the optimization problem are the hosts and the overall computational capacity (basic and elastic) that is allocated to each component of the services under examination.

II. Problem Description

In the context of the problem under study, largely inspired by the OPTIMIS [6] and S(o)OS[1] European Projects, an IP owns physical hosts with potentially heterogeneous characteristics, and establishes Service-Level Agreements (SLAs) with SPs for hosting distributed services over a period of time. Each service is composed of components that are horizontally scalable, i.e. they are capable of distributing their own work over a number of VMs which can be deployed on different cores, processors, and hosts.

[1]More information is available at: http://www.soos-project.eu/.

The IP books a set of physical resources for hosting the VMs that encapsulate the components. Each component is characterized by specific computing requirements in terms of an abstract single-valued performance metric, as explained in detail in Section II-A. The SP can specify a lower and an upper limit to the computing requirements of each component that correspond to the basic and elastic requirements that may be needed during runtime, further translated into basic and elastic VMs. At the first activation, only the basic VMs participate in the service execution. During runtime and according to the workload and the policies in place, elastic VMs may be added, the number of which cannot exceed the defined limit.

A. Performance Model

In this work, it is assumed that the computing capabilities of the hosts and the computing requirements of the services may be expressed in terms of a single performance metric. For example, across a set of hosts with similar capabilities in terms of the Instruction-Set Architecture (ISA) of the CPUs, the computing capabilities may be approximated in terms of instructions per second that each host can process, accounting for the different clock speeds and number of CPUs and available cores. Although, a more precise performance model could consider a vector of metrics, we assume an ideal model of scalability in which each service can be arbitrarily decomposed in a number of possibly imbalanced replicas running over possibly heterogeneous hosts, and that the whole performance of a service is given by the native sum of the performance of the decomposed replicas (thus neglecting the additional workload distribution and synchronization overheads).

Additionally, each service isassociated with an availability of overall computing capability of u and an overall QoS value ψ per unit of allocated computing capacity. Whenever the service will find as available an overall computing capability of $x \leq u$, its overall QoS will actually become $\psi \cdot x$. Therefore, despite the ideal operational level, with a resource requirement of u bringing an overall QoS of $\psi \cdot u$, and given an actual allocation of $x \leq u$, the service will exhibit an overall QoS of $\psi \cdot x \leq \psi \cdot u$.

B. Resources & Services Notation

The IP's resources are modelled as an interconnection of potentially heterogeneous networks and computing nodes:

- A set of computing nodes, or hosts: $\mathcal{H} = \{1, \ldots, N_H\}$.
- Each host $j \in \mathcal{H}$ is characterized by an available computing capacity $U_j \in \mathbb{P}$, which expresses the value of a given system-wide reference performance metric (see Section II-A).

The following notation is used to refer to services:

- A set of service instances: $\mathcal{S} = \{1, \ldots, N_S\}$.
- Each service $s \in \mathcal{S}$ is a workflow of $n^{(s)}$ components (encapsulated inside VMs): $\mathcal{S}^{(s)} \triangleq \left\{\xi_1^{(s)}, \ldots, \xi_{n^{(s)}}^{(s)}\right\}$.

Each component $\xi_i^{(s)} \in \mathcal{S}^{(s)}$ is characterized by the following parameters:

- Minimum basic allocation $\theta_i^{(s)}$ that $\xi_i^{(s)}$ needs to perform its basic functionality;
- Maximum extra allocation $\Xi_i^{(s)}$, also called elastic requirements, that $\xi_i^{(s)} \in \mathcal{S}^{(s)}$ may properly exploit;
- Apportioned overall QoS if the service is admitted $\Psi^{(s)}$, and the apportioned extra QoS per unit of allocated (elastic) computing capacity $\psi_i^{(s)}$.

Note that, with an actual allocation of a service across the hosts, the resulting overall service QoS $Q^{(s)}$ amounts to:

$$Q^{(s)} = \Psi^{(s)} + \sum_{i=1}^{n^{(s)}} \psi_i^{(s)} \cdot \left(\sum_{j \in \mathcal{H}} x_{i,j}^{(s)} - \theta_i^{(s)} \right). \tag{1}$$

C. Service Level Agreement Model

The SLA that is established between the IP and the SP for a given service $s \in \mathcal{S}$ carries the following parameters:

- The computing requirements of each component $\xi_i^{(s)}$: $\psi_i^{(s)}$, $\theta_i^{(s)}$ and $\Xi_i^{(s)}$.
- A minimum probability $\phi^{(s)}$ that there are sufficient resources for the activation of the VMs when needed.
- A gain $G^{(s)}$ for the IP in case the service is accepted.
- A penalty $P^{(s)}$ for the IP if the QoS restrictions are not met.

III. Problem formulation

A. Unknown Variables

First of all, let us introduce the variables (unknown) to be computed. These are the allocated (both basic and elastic) computing capacity for the components on the hosts: $\forall s \in \mathcal{S}$, $\forall i \in S^{(s)}$, $\forall j \in \mathcal{H}$, $x_{i,j}^{(s)} \in \mathbb{P} \subset \mathbb{R}^+$. If a component $\xi_i^{(s)}$ is not given any computing capacity on a given host j, then $x_{i,j}^{(s)} = 0$. A component that is rejected is characterized by: $x_{i,j}^{(s)} = 0$, $\forall j \in \mathcal{H}$. Note that the actual decomposition of each service into VMs is a lower-level detail that is not needed to be addressed in the formulated allocation problem.

In order to allow the possibility of rejecting one or more services that are being examined at the same time, we introduce into the problem the derivative Boolean variables $\left\{x^{(s)}\right\}$ with a value of 1 if the whole service $\mathcal{S}^{(s)}$ is admitted and 0 otherwise. These can be put in relationship with the $\left\{x_{i,j}^{(s)}\right\}$ allocation variables through a pair of inequality constraints that force them to give enough computing capacity for the basic requirements of each component $\left\{\theta_i^{(s)}\right\}$ with $x^{(s)} = 1$, or alternatively be null with $x^{(s)} = 0$ (the entire service is rejected).

B. Allocation Constraints

The problem allocation constraints are the following:

- The maximum allocated computing capacity for each component should not exceed the limit defined by the basic plus the elastic computing capacity for $\xi_i^{(s)}$:

$$\forall s \in \mathcal{S}, \forall i \in S^{(s)}, \sum_{j \in \mathcal{H}} x_{i,j}^{(s)} \leq \theta_i^{(s)} + \Xi_i^{(s)}. \quad (2)$$

- The additional load imposed on each host cannot overcome its residual available computing capacity:

$$\forall j \in \mathcal{H}, \sum_{s \in \mathcal{S}} \sum_{i \in S^{(s)}} x_{i,j}^{(s)} \leq U_j. \quad (3)$$

C. Probabilistic Horizontal Elasticity

If statistical knowledge about the actual resource requirements experienced at run-time by each component is known to the IP (e.g. from historical monitoring data of the service), then this information can be used to tune the allocation in such a way that the service runs flawlessly with at least a minimum probability $\phi^{(s)}$. To this purpose, let $x_i^{(s)} = \sum_{j \in \mathcal{H}} x_{i,j}^{(s)}$ denote the overall computing capacity for a given component. Assuming that the IP has knowledge about the cumulative probability function $F_i^{(s)}(\cdot)$ of the real-valued random variable $X_i^{(s)}$ representing the computational capacity that the component may require at runtime, then the probability that it may use a computing capacity up to x is $F_i^{(s)}(x) = P\left[X_i^{(s)} \leq x\right]$.

In order to deal simultaneously with all the components that comprise the service, we use the joint cumulative distribution function of all random variables $X_i^{(s)}, \forall i \in \left\{1, \ldots, n^{(s)}\right\}, F^{(s)}(x_1, .., x_{n^{(s)}})$. Then, instead of reserving resources for the maximum elasticity requirements deterministically, it is sufficient to guarantee that the probability for $\mathcal{S}^{(s)}$ to find enough available computing power when actually required to be higher than a minimum $\phi^{(s)}$:

$$F^{(s)}(x_1^{(s)}, \ldots, x_n^{(s)}) \geq \phi^{(s)}. \quad (4)$$

Note that, if $\phi^{(s)} = 1$, then the deterministic case is obtained as a particular case of the probabilistic one.

D. Objective Function

From the IP's perspective, a metric for the goodness of an allocation solution may be significant of the additional costs possibly needed to admit the services. In order to formalize this, we introduce an extra cost of ζ_j associated with turning on a host j that is unused when the problem is formulated (j $\in \mathcal{H}_{off} \subset \mathcal{H}$), and we also introduce the h_j Boolean variables stating whether or not the host $j \in \mathcal{H}$ is used in the allocation, which can be encoded using the $x_{i,j}^{(s)}$ variables. Then, a simple term to consider in the objective function is the total additional cost associated to the admitted reservations: $C = \sum_{j \in \mathcal{H}_{off}} h_j \cdot \zeta_j$. The same reasoning may be applied to other host or SP-level information that is available and of importance to the IP, such as the eco-efficiency of the hosts using mechanisms as in [7].

Additionally, the probabilistic framework, as introduced in Section III-C implies that with a maximum probability of $\overline{\phi^{(s)}} \triangleq 1 - \phi^{(s)}$ an admitted service is not expected to find the resources needed for the extra capacity available, leading to the necessity to pay the penalty $P^{(s)}$ back to the customer. Therefore, for each service $s \in S$ that is admitted into the system ($x^{(s)} = 1$), the expected penalty due to SLA violations $\overline{\phi^{(s)}} P^{(s)}$, should be subtracted from the immediate gain $G^{(s)}$.

Taking into account the contributions due to the overall QoS brought to the system by the admitted services, to the expected revenue, and to the additional costs due to the need of new hosts, we finally obtain the following objective function:

$$\max \sum_{s \in \mathcal{S}} x^{(s)} (w_G (G^{(s)} - \overline{\phi^{(s)}} P^{(s)}) - w_C C + w_Q Q^{(s)}) \quad (5)$$

where w_G, w_C, and w_Q are proper coefficients useful for adapting the heterogeneous quantities in the sum, and configuring the relative weights of the different factors in the overall IP's acceptance policy.

Table I
HOSTS AND SERVICES CHARACTERISTICS

$j \in \mathcal{H}$	ζ_j	U_j
j_1	-	20
j_2	-	18
j_3	5	1.5
j_4	8	1.0

S	s_1	s_2
$G^{(s)}$	100	150
$P^{(s)}$	3	3
$\phi^{(s)}$	0.7	0.8
$\Psi^{(s)}$	10	10
$\psi_i^{(s)}$	1	1

IV. SIMULATION RESULTS

The formalized Mixed-Integer Non-Linear Programming (MINLP) optimization problemwas modeled on the General Algebraic Modeling System (GAMS) [1], and solved using the Branch and Reduce Optimization Navigator (BARON) [10]. The presented results have been obtained using GAMS v23.3 on Intel (R) dual-core 2.99 GHz processor with 2 GB of RAM.

We consider an indicative case study of four hosts $\mathcal{H} = \{j_1, j_2, j_3, j_4\}$, with the characteristics shown in Table I, and with two of them being unoccupied $\mathcal{H}_{off} = \{j_3, j_4\}$. For simplicity reasons, the CPU cores are considered to be homogeneous, i.e. $U_j = 1.0$ refers to a single-core host, $U_j = 2.0$ to a dual-core host, etc. The hosts are connected to the same subnet, and it is further assumed that the requirements of the services are negligible compared to the capacity of the underlying network.

Under these settings, we consider two services $S = \{s_1, s_2\}$ requesting admission with the parameters shown in Table II. Both services consist of four components $S^{(s)} = \{\xi_1, \xi_2, \xi_3, \xi_4\}$. Each component has the same capacity requirements both for the basic and the elastic requirements: $\theta_i^{(s)} = 1$, and $\Xi_i^{(s)} = 4$. For simplicity and without loss of generality, we consider that the probability distributions of the components capacity requirements x_i are independent and uniformly distributed in the interval $[\theta_i^{(s)}, \theta_i^{(s)} + \Xi_i^{(s)}]$.

Table II
INDICATIVE CASES UNDER EXAMINATION

Case	w_G	w_C	w_Q
I	1	1	0
II	1	1	1
III	1	1	3

Table III
COMPARISON OF DIFFERENT CASES

Case	I	II	III
Unoccupied hosts	j_3, j_4	j_4	-
Accepted elastic req.	30	31.5	32
Gain	246.5	245.5	244.5
Solution time (secs)	0.469	0.343	0.344

The values of the parameter $\phi^{(s)}$ are less than 1 (see Table I), and are kept fixed for all cases. This helps in highlighting the way this flexibility in terms of availability is regulated by the weights of the different factors in the objective function. The Ψ^s and $\psi_i^{(s)}$ QoS parameters are also kept fixed, so that they have no influence on discriminating which services to admit.

Three indicative cases are examined as presented in Table II. In Case I, the weights w_G and w_C are non-zero, whereas the QoS weight w_Q is set to 0, denoting a profit-driven IP. According to the BARON output, the optimal solution allocates all components on the already occupied hosts $\{j_1, j_2\}$, whereas the accepted elastic requirements are compressed to the minimum allowed by Equation 4, with service s_2 being granted a larger amount since $\phi^{(s_2)} > \phi^{(s_1)}$.

For Case II in which the QoS weight w_Q is increased to 1 meaning that the IP becomes more sensitive in terms of the QoS offered to the clients, the optimal allocation pattern now includes the previously unoccupied host j_3 (the unoccupied host of the lowest cost), which is now turned on for hosting more elastic requirements as compared to the previous case. Further increasing w_Q as in Case III, leads to the acceptance of the maximum elastic requirements and the second unoccupied host j_4 being turned on as well.

V. CONCLUSIONS

This paper presented a probabilistic admission control test, which incorporates statistical knowledge about needing extra elastic requirements for the services, and thus allowing for reducing the physical computational resources that need to be booked for elasticity reasons. The proposed model can be extended to allow for proper trade-offs among high-level business objectives, depending on their relative importance. The presented method was modeled on GAMS and solved under realistic provider's settings that demonstrate its effectiveness.

ACKNOWLEDGEMENTS

The research leading to these results has received funding from the European Community's Seventh Framework Programme FP7 under grant agreement n.257115 OPTIMIS – Optimized Infrastructure Services and n.248465 S(o)OS – Service-oriented Operating Systems.

REFERENCES

[1] General Algebraic Modeling system (GAMS). GAMS Development Corporation. Available at http://www.gams.com/.

[2] L. Abeni and G. Buttazzo. QoS guarantee using probabilistic deadlines. *Euromicro Conference on Real-Time Systems*, 0:0242, 1999.

[3] A. Beloglazov, J. Abawajy, and R. Buyya. Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing. *Future Generation Computer Systems*, May 2011.

[4] T. Cucinotta, K. Konstanteli, and T. Varvarigou. Advance reservations for distributed real-time workflows with probabilistic service guarantees. In *Proceedings of the IEEE International Conference on Service-Oriented Computing and Applications (SOCA)*, Taipei, Taiwan, December 2009.

[5] A. Davare, Q. Zhu, M. Di Natale, C. Pinello, S. Kanajan, and A. Sangiovanni-Vincentelli. Period optimization for hard real-time diustributed automotive systems. In *Proc. of DAC'07*, San Diego, California, USA, June 2007.

[6] Ferrer et al. Optimis: A holistic approach to cloud service provisioning. *Future Generation Computer Systems*, In Press, Corrected Proof:–, 2011.

[7] Í. Goiri, F. Julia, R. Nou, J. L. Berral, J. Guitart, and J. Torres. Energy-aware scheduling in virtualized datacenters. In *Proceedings of the IEEE International Conference on Cluster Computing*, pages 58–67, 2010.

[8] K. Konstanteli, T. Cucinotta, and T. Varvarigou. Optimum allocation of distributed service workflows with probabilistic real-time guarantees. *Springer Service Oriented Computing and Applications*, 4(4):229–243, 10 2010.

[9] A. F. Mills and J. H. Anderson. A stochastic framework for multiprocessor soft real-time scheduling. In *Proceedings of the 16th IEEE Real-Time and Embedded Technology and Applications Symposium*, RTAS '10, pages 311–320, Washington, DC, USA, 2010.

[10] N. V. Sahinidis. Global optimization and constraint satisfaction: The branch-and-reduce approach. *C. Bliek, C. Jermann, and A. Neumaier (eds.), Lecture Notes in Computer Science, Springer, Berlin*, 2861:1–16, 2003.

[11] H. Zheng, J. Yang, and W. Zhao. QoS probability distribution estimation for web services and service compositions. In *IEEE International Conference on Service-Oriented Computing and Applications (SOCA)*, pages 1 –8, December 2010.