Themis: A Spot-Market Based Automatic Resource Scaling Framework

Stefania Victoria Costache, Nikos Parlavantzas, Christine Morin, Samuel Kortas

Themis: A Spot-Market Based Automatic Resource Scaling Framework

Stefania Costache *
EDF R&D, Clamart, France
INRIA Rennes - Bretagne Atlantique Centre, France

Nikos Parlavantzas INRIA
Rennes - Bretagne Atlantique Centre, France INSA Rennes, France

Christine Morin INRIA
Rennes - Bretagne Atlantique Centre, France

Samuel Kortas
EDF R&D, Clamart, France

ABSTRACT

Cloud computing brings new provisioning models that offer applications more flexibility and better control over their resource allocations. However these models suffer from the following problem: either they provide limited support for applications demanding quality of service, or they lead to a limited infrastructure utilization. In this paper we propose Themis, a novel resource management system for virtualized infrastructures based on a virtual economy. By limiting the coupling between the applications and the infrastructure through the use of a dynamic resource pricing mechanism, Themis can support diverse types of applications and performance goals while ensuring an efficient resource usage.

1. MOTIVATION

High Performance Computing (HPC) infrastructures are used to execute increasingly complex and dynamic applications. Given the workload dynamicity and heterogeneity, a general concern in managing these infrastructures is ensuring the best application performance and resource utilization. Cloud providers partially address this concern by offering users "instant" access to virtualized resources [1], and thus, allowing them to meet their SLOs more effectively. However, to maintain this "instant provisioning" illusion a cloud provider needs to keep a large amount of idle resources.

An approach that overcomes this limitation is to automatically allocate resources to applications through a spot market. In this approach users submit bids for resources and their allocations and the resource price vary with the total user demand. This mechanism allows users with valuable requests to receive resources instantly while users with less valuable requests can benefit more from resources in low utilization periods. When applying this approach in managing virtualized HPC infrastructures, two challenges need to be addressed: (i) how to implement a spot market mechanism in an efficient way to provide maximum resource utilization and fairness to users; (ii) how to provide support to applications to meet their SLOs.

*The author is currently a PhD student at INRIA and EDF R &D. This work is supported by ANRT through the CIFRE sponsorship No. 0332/2010

2. THEMIS

We have developed Themis, a spot-market based resource allocation system. The contribution that Themis brings is a decentralized resource control system that uses dynamic pricing as a user demand regulating mechanism. Our system relies on the use of a virtual economy and is composed of: (i) independent application controllers that use feedback-loops to dynamically provision resources for applications and meet their SLOs; (ii) a market-based proportional-share scheduler to control the resource allocation; (iii) a virtual bank to manage the system's total currency. Figure 1 describes our system's components and the communication between them. We detail these components next.

Application Controllers: Themis' application controllers use feedback loops based on generic adaptation policies to change the resource allocation given the current spot price, users valuation for applications and their specified SLOs. An application controller uses plugins that monitor, parse application performance information and adapt incrementally the number of virtual machines and submitted bid. Users can replace these plugins with their own, to build controllers specific to their application types. To provide support for building specific feedback loops, we provide two generic adaptation policies that change the virtual machine bid and the number of virtual machines. To execute their applications on the infrastructure, users start an application controller with an XML application configuration file. The application configuration contains: (i) deployment information: a template used to specify any cloud-manager required virtual machine configuration; files to be transferred to virtual machines; required scripts to be executed to start and stop application components; (iii) budgeting information (i.e., an initial credit amount and the rate at which it is renewed); (iv) adaptation policy information (i.e., policy type to be used, specific thresholds). Based on this information the application controller provisions an initial number of virtual machines and starts the application.

Market-based Scheduler: The market-based proportional-share scheduler ensures maximum resource utilization through

its work-conserving nature and fine-grained resource control. The scheduler allocates fractional amounts of CPU resource to virtual machine instances proportional to the virtual machine bid and inversely proportional to the current resource price. Previous work has shown that this mechanism can provide a good efficiency even when users anticipate the effect of changing their bids on the resource price [5]. To ensure the work-conserving property of the proportional-share policy (i.e., any available resource amount is distributed between the virtual machines), and that each virtual machine receives a share as close as possible to the one that it deserves, the scheduler also performs load-balancing between hosts. The load-balancing algorithm uses a tabu-search heuristic that keeps the number of virtual machine migrations below an administrator-defined limit.

Cloud Manager: We built our system on top of the OpenNebula cloud manager [7]. The components of Themis interact with OpenNebula through its provided XML-RPC API to : (i) perform virtual machine operations (e.g., deploy, migrate, terminate); (ii) retreive monitoring information about physical nodes and virtual machines; (iii) retreive user accounts information and authenticate the users in the system to allow them to submit their applications. As OpenNebula does not support virtual machine dynamic resizing, the scheduler enforces its computed CPU allocations by communicating with the node hypervisors (i.e., Xen).

Virtual Bank: A virtual bank manages the system's total currency. Users are funded with an amount of virtual currency periodically recharged, based on the amount that was spent. Currently, to avoid hoarding of credits and budget depletion, a simple policy is implemented: when an application is submitted, the virtual bank creates an account for it and transfers from the user's account an initial amount of credits, replenished periodically at a user-specified rate.

3. VALIDATION

We have previously simulated the architecture of Themis using CloudSim toolkit [3] and tested simple scaling policies for two types of applications, namely rigid and elastic [4]. Currently we have implemented Themis in Python. The communication between its components is done through XML-RPC. All the information regarding application and user accounts is kept in a MySql database storage.

To show a real use-case of Themis, we have deployed it on 10 nodes from Grid'5000 [2] and developed an application controller that automatically scales a Condor [6] managed virtual cluster using feedback regarding the resource price and the number of pending jobs. To change the number of virtual machines we have used a simple adaptation policy: (i) the number increases when the number of jobs in Condor's queue is higher than a given threshold; (ii) the number decreases when there are fewer jobs in queue or the resource price is too high. We submitted two batches of jobs. During the first submission the infrastructure was underutilized and the Themis controller increased the number of virtual machines, leading to a maximum resource utilization and job-processing throughput. During the second submission the infrastructure was highly used and the Themis controller could not scale the number of virtual machines due to its budget limitation and the high resource price.

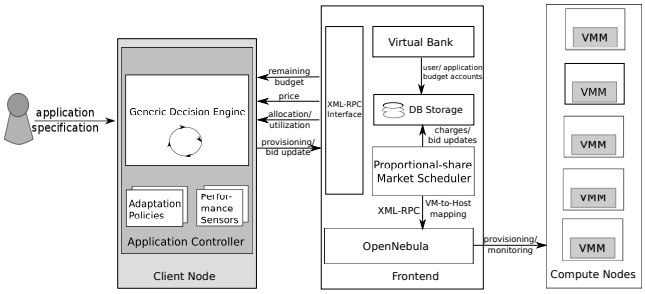

Figure 1: System overview.

4. CONCLUSIONS

We have presented an automatic resource management framework based on a spot market. With Themis the resource control is decentralized, allowing each application to apply its own adaptation policies while the resource management is kept generic and simple.

We currently plan to measure the impact of the resource control decentralization and virtual machine operations. We will extend our system in two directions: (i) define controller policies for different application types; (ii) extend the scheduling algorithm to support allocation of multiple types of resources. Regarding the controller policies, we plan to take scaling decisions based on an application reconfiguration cost and future price predictions. Regarding the scheduling algorithm we plan to investigate how application preferences for multiple resources can be considered in the design of a fair efficient allocation policy.

5. REFERENCES

[1] AmazonEC2. http://aws.amazon.com/ec2/.

[2] R. Bolze, F. Cappello, E. Caron, M. Daydé, F. Desprez, E. Jeannot, Y. Jégou, S. Lanteri, J. Leduc, N. Melab, G. Mornet, R. Namyst, P. Primet, B. Quetier, O. Richard, E.-G. Talbi, and I. Touche. Grid'5000: A large scale and highly reconfigurable experimental grid testbed. *International Journal of High Performance Computing Applications*, 2006.

[3] R. N. Calheiros, R. Ranjan, A. Beloglazov, C. A. F. De Rose, and R. Buyya. Cloudsim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms. *Software: Practice and Experience*, 2011.

[4] S. Costache, N. Parlavantzas, C. Morin, and S. Kortas. An economic approach for application qos management in clouds. In *Euro-Par 2011: Parallel Processing Workshops*, Lecture Notes in Computer Science. 2012.

[5] M. Feldman, K. Lai, and L. Zhang. The proportional-share allocation market for computational resources. *IEEE Transactions on Parallel and Distributed Systems*, 20(8), 2009.

[6] M. Litzkow, M. Livny, and M. Mutka. Condor-a hunter of idle workstations. In *8th International Conference on Distributed Computing Systems*, 1988.

[7] B. Sotomayor, R. Montero, I. Llorente, and I. Foster. An Open Source Solution for Virtual Infrastructure Management in Private and Hybrid Clouds. *IEEE Internet Computing*, 2009.