Scheduling Virtual Machines and Grid Tasks on Clouds

Daniel M. Batista, Cesar G. Chaves, Nelson L. S. da Fonseca
Institute of Computing
State University of Campinas, Brazil
Email: {daniel,cchaves}@lrc.ic.unicamp.br, nfonseca@ic.unicamp.br

***Abstract*—Virtualization techniques are central to the provisioning of resources in cloud computing. This paper summarizes the results obtained with a new scheduler for grid applications in clouds. The scheduler takes into account not only the tasks dependencies and the resource demands, but also software requirements. Moreover, it accounts network link availability that has been widely neglected in the literature of grid scheduling.**

I. Introduction

The existence of a massive number of computers with multiple processing cores has motivated the search for solutions to leverage these computing power connected by the Internet in order to build a single virtual super computer. This has led to the creation of "grid computing" [1], a computing paradigm, which allowed a new class of applications and services [2].

As happened with the Internet, grids began to leave the academic environment to form part of the infrastructure of different organizations. The good experience obtained with grids prompted the industry to propose the paradigm of "cloud computing", which are environments that allow organizations to have transparent access to resources, in a pay-per-use basis.

A clear distinction between clouds and grids is that clouds provide a customized environment for each type of user. One way to ensure this environment customization at runtime, without a high cost, is through virtualization of hardware and software. As a matter of fact, virtualization is a key aspect for the functioning of clouds [3].

This paper summarizes the TVM (Tasks and Virtual Machines) scheduler, which schedules virtual machines (VMs) in clouds and tasks on those VMs. By scheduling both, tasks and VMs, it is possible to define virtual computing systems that go beyond the limitation imposed by the availability of resources. It is our best knowledge that there is no other proposal in the literature that accomplishes that. The scheduler is oriented to grid applications composed of dependent tasks.

II. The TVM scheduler

The objective of the TVM scheduler is to minimize the makespan of the application. It is implemented as an integer linear program. The produced schedule establishes the appropriate mapping of tasks and VMs onto hosts and communications, in a way that VMs satisfy the software requirements of the applications. Tasks are mapped in order to make the best possible virtualization of both computational and communication resources.

To assess the performance of the TVM scheduler, the same simulation experiments in [4] were conducted, with the addition of a VMs repository and the software requirements. The simulated grids were represented by graphs that describe the topology of the network composed by shared resources.

Several parameters of the simulated topologies were varied, like the the number of hosts and the availability of the links. Two metrics were evaluated: the makespan of the application when scheduled with the TVM scheduler and the execution time taken by the scheduler to return the schedule.

The results showed that the proposed scheduler reduced the makespan of the application when compared with the serial processing of the application (The lower the makespan, the better the scheduler). This reduction was about 45%. It was observed, too, that the longest execution time of the scheduler was 87.838s, which is an acceptable value for schedulers used in clouds. Moreover, the scheduler was evaluated in scenarios where the available bandwidth of the links was reduced. In these scenarios, the scheduler always proposed a new allocation that avoided the links with low availability.

III. Conclusion and future work

This paper summarized the TVM scheduler, a scheduler able to deal with VMs and task scheduling in clouds. The effectiveness and efficiency of this scheduler has been attested by simulation experiments using various network topologies. Our future work involves the relaxation of constraints of the linear program in order to reduce the time taken to produce schedules.

Acknowledgment

This work was supported by FINEP, FUNTTEL, CNPq and CAPES.

References

[1] I. Foster, "What is the Grid? A Three Point Checklist," *GRIDToday*, vol. 1, no. 6, Jul 2002, http://www-fp.mcs.anl.gov/~foster/Articles/WhatIsTheGrid.pdf Accessed at 11 Mar 2010.

[2] S. M. Larson, C. D. Snow, M. Shirts, and V. S. Pande, "Folding@Home and Genome@Home: Using Distributed Computing to Tackle Previously Intractable Problems in Computational Biology," in *Computational Genomics: Theory and Application*, Richard P. Grant, Ed. Horizon Bioscience, Sep 2004.

[3] R. Buyya, C. S. Yeo, S. Venugopal, J. Broberg, and I. Brandic, "Cloud Computing and Emerging IT Platforms: Vision, Hype, and Reality for Delivering Computing as the 5th Utility," *Future Generation Computer Systems*, vol. 25, no. 6, pp. 599–616, June 2009.

[4] D. M. Batista, N. L. S. da Fonseca, F. K. Miyazawa, and F. Granelli, "Self-Adjustment of Resource Allocation for Grid Applications," *Computer Networks*, vol. 52, no. 9, pp. 1762–1781, 2008.