Meeting Application Constraints Efficiently by Deploying Cloud Resources On-demand

Mohsen Amini Salehi, Alexandre di Costanzo, Rajkumar Buyya
Cloud Computing and Distributed Systems Lab (CLOUDS Lab)
Computer Science and Software Engineering Department
University of Melbourne
Melbourne, Australia
{mohsena,adc,raj}@csse.unimelb.edu.au

Abstract— **Cloud computing and Infrastructure-as-a-Service providers, such as Amazon Elastic Computing Cloud (EC2), have paved the way towards computing as a utility. The integration of the cloud providers with in-house infrastructure can be used to support on-demand resource provisioning, providing elasticity in modern applications, and making dynamic adaptation of the service capacity to variable user demands. Cluster and grid computing environments are two examples of services which can obtain a great benefit from these technologies. In this work we aim to propose resource provisioning policies for applications which cannot finish within their deadline using current resources.**

Keywords-Cloud Computing; Market-oriented Scheduling;QoS Constrains.

I. INTRODUCTION

As more complicated and resource consuming applications are emerging, the need for deploying more resources is rising. Technologies that can provide and manage resources for such applications are becoming more essential. Although recently Cloud computing [1] sounds promising in providing illusion of "infinite resource" for user applications, the challenge ahead will be making efficient provisioning policies for applications to take the advantage of public Cloud providers, organizational resources, along with resources from other organizations. Intelligent decisions, made by provisioning policies in taking resources from Cloud providers, can result in large benefits in overall turnaround time.

The advances in virtual machine and network technologies has resulted in appearing commercial providers, which offer numbers of resources to users, charging in a "pay-as-you-go" fashion. As the provided resources are on a "Cloud" whose physical infrastructure is unknown to the users; it is called "Cloud Computing". This new paradigm for the provision of computing infrastructure on the network (IaaS) has reduced the capital expenditure for infrastructure and management. Amazon Elastic Compute Cloud [2], or as it is known "Amazon EC2", is the most famous IaaS provider which is considered the pioneer in the Cloud computing era.

High performance applications, such as Bag of Tasks (BoT), usually have QoS constraints in terms of deadline, and budget which available resources alone may not be able satisfy them. One solution for this problem is taking the advantage of Cloud resources. However, Clouds resources are not free and users are charged based on their usage amount (usually in an hourly basis). A reasonable approach is getting resources from Cloud providers in an on-demand manner. Therefore, in these circumstances the challenges are:

a) When is it necessary to ask public Clouds to aid local resources to be able to satisfy the application deadline?

b) How many virtual machines should be created on the Cloud based on the user budget?

Market-oriented computing [3] considers computing resources in economic terms in a way that deploying resources implies paying to resource providers for utilizing the computing resources. Market-oriented scheduling policies enable both resource consumers and suppliers to maximize their utility and profit.

Gridbus Broker [4] is a user-level scheduler that mediates access to distributed resources running diverse middleware. It is able to interface with various middleware services, such as PBS, Condor, and SGE. The Broker has equipped by market-oriented scheduling strategies which consider differing objectives namely, time and cost optimization. It maps jobs to the appropriate resources to meet these objectives.

In this contribution we try to propose policies for the aforementioned challenges. More specifically, the policies lease virtual machines from cloud provider when it predicts that it cannot meet the deadline. The number of leased virtual machines depends on the budget provided by the user. These policies have been implemented in the Gridbus Broker.

II. PROPOSED SOLUTION

In this section we propose our solution for satisfying user deadline when the deadline could not be met by relying just on local resources. As mentioned before, our solution tries to reinforcing local infrastructure processing power by deploying resources from commercial cloud providers in exchange of a fee. Fig. 1 depicts a brief and high level view of the proposed policy.

The proposed solution has been implemented in Gridbus Broker. There are several reasons that motivated us to use Gridbus broker for the integration of Cloud resources with other accessible resources such as local resources, or resources from other organizations.

First of all, Gridbus broker has a proven performance for running different types of applications such as Bag-of-Tasks (BOT) and Parameter Sweep Application (PSA), Workflows, and data-intensive applications in real environments [4]. Secondly, Gridbus broker is highly extensible for supporting any kind of middleware. Third feature of Gridbus broker is being able to schedule tasks based on user QoS needs, such as deadline, cost or both. The last but not the least, is that Gridbus broker has implemented based on Market-oriented resource management idea, which its time has finally come with the emergence of commercial Cloud providers. More specifically, Cloud providers, such as Amazon, charges user for computational power as well as storage and bandwidth. Gridbus broker, however, consider costs of all of these items.

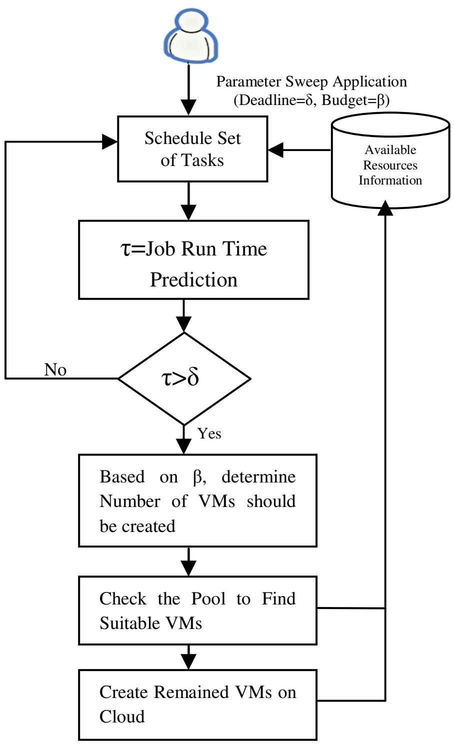

Figure 1. Flowchart of the proposed policy.

Changes were required to allow the broker to create virtual machines on Amazon EC2 on demand. This has been done by developing a new actuator for Gridbus broker and changing the way users are charged in Gridbus broker.

Connecting to Amazon EC2 has been using Amazon Web Services (AWS). Moreover, implementing the proposed policy would need changes in the scheduler part of the broker as well as making another part for managing pool of hired virtual machines.

III. EXPECTED OUTCOME

The expected outcome of the proposed system is showing how using public Clouds computational capabilities helps to meet QoS constrains needed by a high performance application. Reinforcing local resources power is done through hiring optimal number of virtual machines from the Cloud provider and considers the budget available for the user.

In my demonstration I will use real parameter sweep application (such as PovRay [5] which is an image rendering application). The demonstration would be in two different scenarios.

In the first scenario, I will show that if the application could be finished without extra resources in its deadline, the broker would not hire any virtual machine from Cloud providers.

In the second scenario, I will show how PovRay can be finished in its deadline by hiring proper amount of resources from Amazon, while the same application could not be finished in the deadline without cloud resources.

Fig. 2 and Fig. 3, show the way cloud resources are declared in the context of Gridbus Broker; and a PovRay application running on both local and Cloud resources respectively.

```
<service type="compute" cost="5.0 " mappingID="remote">
    <compute firewall="false" middleware="fork" >
    <fork hostname="snowball.cs.gsu.edu" />
    </compute>
    </service>
<service type="compute" cost="10.0" mappingID="amazon">
    <compute middleware="EC2" >
    <EC2>
    <hostname>ami-dfbb5ab6</hostname>
    <minInstance>1</minInstance>
    <maxInstance>2</maxInstance>
    <instanceType>m1.small</instanceType>
    <availabilityZone>us-east-1a</availabilityZone>
    </EC2>
    </compute>
    </service>
```

Figure 2. Sample Service Description File provided by the user to Request Cloud Resources.

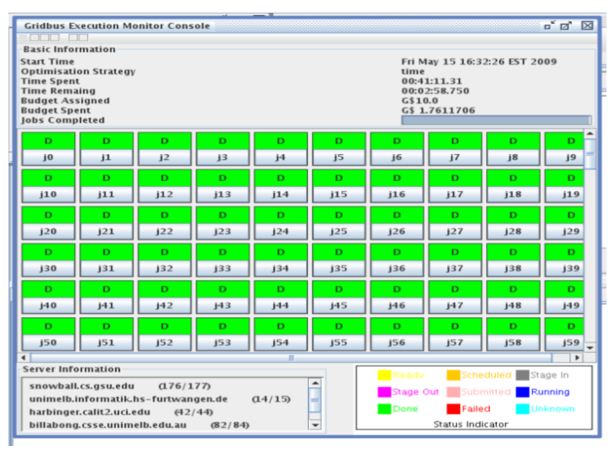

Figure 3. A screenshot of the Gridbus broker running parameter sweep application on Cloud resources along with the usual resources from other providers.

REFERENCES

[1] Michael Armbrust, Armando Fox, Rean Griffith, Anthony D. Joseph, Randy H. Katz, Andrew Konwinski, Gunho Lee, David A. Patterson, Ariel Rabkin, Ion Stoica, Matei Zaharia, Above the Cloud, A Berkely view of Cloud Computing.

[2] Amazon compute cloud: http://aws.amazon.com/ec2/

[3] Rajkumar Buyya, Chee Shin Yeo, and Srikumar Venugopal, Market-oriented Cloud computing: Vision, hype, and reality for delivering it services as computing utilities, 2008.

[4] Srikumar Venugopal, Krishna Nadiminti, Hussein Gibbins and Rajkumar Buyya, Designing a Resource Broker for Heterogeneous Grids, Software: Practice and Experience, 38(8):793-825, 2008.

[5] Pov-Ray- the Persistence of Vision Raytracer: http://www.povray.org