Future Generation Computer Systems 28 (2012) 58–65

Contents lists available at SciVerse ScienceDirect

Future Generation Computer Systems

journal homepage: www.elsevier.com/locate/fgcs

Deadline-driven provisioning of resources for scientific applications in hybrid clouds with Aneka

Christian Vecchiola [a], Rodrigo N. Calheiros [a], Dileban Karunamoorthy [a], Rajkumar Buyya [a,b,*]

[a] Cloud Computing and Distributed Systems (CLOUDS) Laboratory, Department of Computer Science and Software Engineering, The University of Melbourne, Australia
[b] Manjrasoft Private Limited, Melbourne, Australia

ARTICLE INFO

Article history:
Received 23 December 2010
Received in revised form
2 May 2011
Accepted 7 May 2011
Available online 14 May 2011

Keywords:
Resource provisioning
Hybrid cloud
QoS
E-Science
Aneka

ABSTRACT

Scientific applications require large computing power, traditionally exceeding the amount that is available within the premises of a single institution. Therefore, clouds can be used to provide extra resources whenever required. For this vision to be achieved, however, requires both policies defining when and how cloud resources are allocated to applications and a platform implementing not only these policies but also the whole software stack supporting management of applications and resources. Aneka is a cloud application platform capable of provisioning resources obtained from a variety of sources, including private and public clouds, clusters, grids, and desktops grids. In this paper, we present Aneka's deadline-driven provisioning mechanism, which is responsible for supporting quality of service (QoS)-aware execution of scientific applications in hybrid clouds composed of resources obtained from a variety of sources. Experimental results evaluating such a mechanism show that Aneka is able to efficiently allocate resources from different sources in order to reduce application execution times.

1. Introduction

Cloud computing [1] platforms are rapidly emerging as the preferred option for hosting applications in many business contexts. Start-up companies are relying on public cloud infrastructures to deploy their applications, which helps in reducing their initial costs. Larger companies are also adopting clouds, either public clouds for expanding their existing infrastructures or rapid deployment of test environments, or private clouds for dynamic on-demand provisioning of virtual resources among their internal divisions.

The same widespread adoption of cloud platforms for application deployment is not yet observed in the case of scientific computing applications. Scientific computing, or computational science, is the field of study concerned with devising mathematical models and numerical techniques aiming to address problems in science and engineering. These problems typically involve long-term computer simulations, huge dataset processing, and other types of large-scale computations which, in many cases, require the availability of large IT infrastructures. Traditionally, these needs have been addressed on the internal computing infrastructure of research institutions, which are typically composed of clusters or resources from local area networks (desktop grids).

Later, with the advent of grid computing [2], larger infrastructures were made available to scientists. The availability of both software and infrastructure for grid computing significantly influenced the growth of scientific computing research. As a result, scientific applications became the major user of grid facilities, and many public grids have been deployed for supporting these applications. However, high utilization rates observed in grids, along with technical and bureaucratic issues, limit their utilization.

Moreover, these resources may be insufficient in certain periods of time. For example, peak demand for scientific resources may be seen in some parts of the year, which can lead to long waiting times for utilization of these resources, or the available resources for one application may be insufficient to complete the application before its deadline. In these cases, scientific resources may be complemented by cloud resources. Moreover, by leasing cloud computing services on a pay-per-use basis, even minor institutions can easily access a large number of resources, which are utilized and paid for only for the time they are actually utilized.

For this vision to be achieved, however, middleware supporting provisioning of resources from both local infrastructures and public clouds (known as hybrid clouds) is required, so that applications can transparently migrate to public virtual infrastructures [3].

Aneka [4] is a software platform for building and managing a wide range of distributed systems, allowing applications to receive

* Corresponding author at: Cloud Computing and Distributed Systems (CLOUDS) Laboratory, Department of Computer Science and Software Engineering, The University of Melbourne, Australia.
E-mail address: rbuyya@unimelb.edu.au (R. Buyya).

0167-739X/$ – see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.future.2011.05.008

resources provisioned from different sources, such as desktop grids, scientific grids, clusters, private clouds, and public clouds. Such a hybrid cloud built with resources from a variety of sources is managed transparently by Aneka. Therefore, this platform enables the execution of scientific applications in hybrid clouds.

The contribution of this paper is twofold. First, it describes Aneka's features that are responsible for supporting the quality of service (QoS)-aware execution of scientific applications in hybrid clouds composed of resources obtained from a variety of sources. Second, it presents experimental results of the utilization of these features in an actual hybrid cloud. Results show that Aneka is able to efficiently allocate resources from different sources in order to reduce application execution times.

The rest of this paper is organized as follows. Section 2 presents an overview of scientific applications and current support for them in the cloud. Section 3 presents a general overview of the Aneka framework. Section 4 describes the provisioning mechanism in Aneka and how it supports different resources. Section 5 presents the algorithms for deadline-driven resource provisioning in Aneka. Section 6 presents the results of using Aneka's capabilities for the execution of a scientific application in a hybrid cloud. Section 7 presents related works, and Section 8 concludes the paper.

2. Scientific computing in the cloud

With the advent of grid computing, large computing infrastructures became available to academic and research institutions for deploying and executing scientific applications. These infrastructures are either contained within the boundaries of an institution, such as a University or a high-performance computing (HPC) facility, or span across several organizations and spread worldwide as in the case of the Enabling Grid for E-sciencE (EGEE) [5], TeraGrid [6], and Open Science Grid (OSG) [7].

Because grid resources are shared worldwide, it is necessary to provide users with controlled access to these resources, which leads to competitive use of these facilities that favors large research institutions and more expensive projects. In addition, technical issues related to the specific nature of the runtime environment provided by grids limit the widespread use of these facilities: in many cases, applications have to be written either according to a specific programming model (e.g., Message Passing Interface [8], Bag of Tasks, or workflows) or for a specific operating system, mostly Unix-like.

Cloud computing provides an alternative approach that solves some of these problems. Such technology delivers IT infrastructure and services on demand on pay-per-use basis. The use of such a billing scheme makes clouds accessible to everyone, from large academic institutions to minor research groups. Moreover, clouds are able to provide distributed systems that grow and shrink dynamically according to the requirements of users, which can identify convenient trade-offs between cost and performance for experiments.

Moreover, virtualization technologies remove many of the technical issues previously mentioned. For example, Infrastructure as a Service (IaaS) solutions allow scientists to prepackage and configure the basic building blocks required for carrying out their experiments. This allows a higher degree of customization that helps cover a wider range of scenarios for scientific computing applications. Finally, provisioning on demand of cloud resources simplifies their integration into existing infrastructures. Academic institutions already have their computing facilities, and these infrastructures can be extended by adding virtual resources or services leased from the cloud, which creates a hybrid infrastructure that serves the institution for the time needed to perform huge computations or large-scale experiments.

Although the use of cloud computing for scientific applications is still limited, the first steps towards this goal have been already made. One of the first cloud-based infrastructures for computational science, Science Cloud [9], has been deployed by the joint efforts of the University of Chicago, the University of Illinois, Purdue University, and Masaryk University.

From a research point of view, studies have been conducted on the feasibility of using clouds for scientific computing. A study by Evangelinos and Hill [10] shows that HPC scientific applications deployed in regular Amazon instances have performance similar to the performance achieved with low-cost clusters, whereas a study by Deelman et al. [11] shows that the cost of cloud for HPC applications is acceptable. From a public cloud provider's perspective, there are initiatives such as Amazon EC2's cluster computing instances,[1] which offer HPC resources connected via a high-throughput network. Such an initiative from Amazon provides a platform with better performance for HPC applications than the regular platform evaluated by Evangelinos and Hill, making the adoption of clouds more compelling to researchers.

The aim of using hybrid public/private clouds is a core feature of the Aneka platform, which enables not only utilization of such clouds, but also utilization of virtually any kind of computational resource available for applications, including idle desktops from local networks, clusters, and grids. The Aneka platform is presented in the next section, and the deadline-driven provisioning of resources for scientific applications is presented in Section 4.

3. Aneka: an extensible cloud application platform

Aneka [4] is a software platform and a framework for the development of distributed applications in the cloud. It harnesses the computing resources of a heterogeneous network of workstations, clusters, grids, servers, and data centers, on demand. It implements a Platform as a Service model, providing developers with Application Programming Interfaces (APIs) for transparently exploiting physical and virtual resources. In Aneka, application logic is expressed with a variety of programming abstractions and a runtime environment on top of which applications are deployed and executed. System administrators leverage a collection of tools to monitor and control the cloud, which can be a public virtual infrastructure available through the Internet, a network of computing nodes in the premises of an enterprise, or a combination of them.

The core feature of the framework is its service-oriented architecture that allows customization of each cloud according to the requirements of users and applications. Services are also the extension point of the infrastructure: by means of services, it is possible to integrate new functionalities and to replace existing ones with different implementations. In this section, we briefly describe the architecture and categorize the fundamental services that compose the infrastructure.

3.1. Framework overview

Fig. 1 provides a layered view of the framework. Aneka provides a runtime environment for executing applications by leveraging the underlying infrastructure of the cloud. Developers express distributed applications by using the API contained in the Software Development Kit (SDK) or by porting existing legacy applications to the cloud. Such applications are executed on the Aneka cloud, represented by a collection of internetworked nodes hosting the Aneka Container. One of the nodes runs the *Aneka master*, which provides resource management and application scheduling capabilities, and the other nodes run *Aneka workers* that process tasks that compose the application.

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