Advances in Engineering Software 42 (2011) 714–722

Contents lists available at ScienceDirect

Advances in Engineering Software

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

Towards building a cloud for scientific applications

Lizhe Wang [a,*], Marcel Kunze [b], Jie Tao [b], Gregor von Laszewski [a]

[a] Pervasive Institute of Technology, Indiana University, USA
[b] Steinbuch Centre for Computing, Karlsruhe Institute of Technology, Germany

ARTICLE INFO

Article history:
Received 2 December 2009
Accepted 26 March 2011
Available online 14 June 2011

Keywords:
Cloud computing
Virtualization
Grid computing
Data center
Intrastructure as a Service
Parallel computing

ABSTRACT

The Cloud computing becomes an innovative computing paradigm, which aims to provide reliable, customized and QoS guaranteed computing infrastructures for users. This paper presents our early experience of Cloud computing based on the Cumulus project for compute centers. In this paper, we give the Cloud computing definition and Cloud computing functionalities. This paper also introduces the Cumulus project with its various aspects, such as design pattern, infrastructure, and middleware. This paper delivers the state-of-the-art for Cloud computing with theoretical definition and practical experience.

1. Introduction

The Cloud computing [19,1] was coined in late of 2007 and currently emerges as a hot topic. A computing Cloud is able to offer flexible dynamic IT infrastructures, QoS guaranteed computing environments and configurable software services. As reported in the Google trends shown in Fig. 1, the Cloud computing (the blue[1] line), which is enabled by virtualization technology (the yellow line), has already outpaced the Grid computing [2] (the red line).

This paper reports our experience on developing a scientific Cloud with some popular Cloud technologies. It is intended to demonstrate flexible design and implementation of a scientific Cloud, which does not bind to any specific underlying technologies. We declare that the Cumulus project enjoys the features such as flexibility, scalability and modularity. The rest of the paper is organized as follows. Section 2 presents related work of Cumulus project and Section 3 define the concept of Cloud computing. Section 5 overview the design of Cumulus project. Sections 6–9 show the various aspects of Cumulus project, such as the front-end service, backend service, virtual machine image preparation, and network solution. We explain the Cumulus usage via a use case – virtual computing center in Section 10 and a sample scientific application – parallel satellite image processing in Section 11. Finally Section 12 concludes the paper.

2. Related work

This section introduces some related work of our Cumulus project.

2.1. Globus virtual workspace and Nimbus

A Globus Virtual Workspace Service [4,3] is an abstraction of a computing environment which are dynamically available to authorized clients via invoking Grid services. Based on Globus virtual workspace services, a cloudkit named Nimbus [5] is developed to build scientific Clouds. The Nimbus contains a frontend Globus service and multiple workspace control agents on host resources for virtual machine deployment.

2.2. OpenNEbula

The OpenNEbula is a virtual infrastructure engine that enables the dynamic deployment and re-allocation of virtual machines in a pool of physical resources. The OpenNEbula system extends the benefits of virtualization platforms from a single physical resource to a pool of resources, decoupling the server, not only from the physical infrastructure but also from the physical location [17].

The OpenNEbula contains one frontend and multiple backends. The front-end provides users with access interfaces and management functions. The back-ends are installed on Xen servers, where Xen hypervisors are started and virtual machines could be backed. Communications between frontend and backends employ SSH. The

* Corresponding author.
E-mail address: lizhe.wang@gmail.com (L. Wang).

[1] For interpretation of color in Figs. 1, 4 and 9, the reader is referred to the web version of this article.

0965-9978/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.advengsoft.2011.05.007

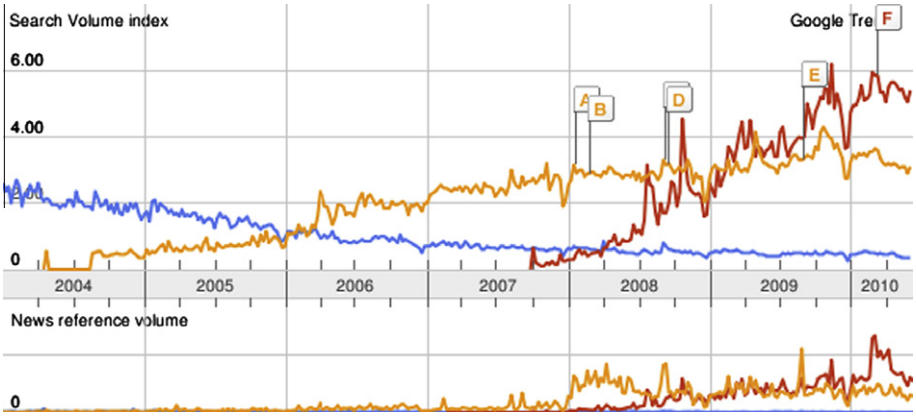

Fig. 1. Cloud computing in Google trends.

OpenNEbula gives users a single access point to deploy virtual machines on a locally distributed infrastructure.

2.3. OS Farm

The OS Farm [9] is an application written in Python to generate virtual machine images and virtual appliances used with the Xen VMM. Two methods are provided for users to connect to the OS Farm server:

- Web interface
  The Web interface provides various methods to enable users to request virtual machine images. The simplest method is simple request which allows users to select image class and architecture. The advance request gives the possibility to add *yum* packages to the virtual machine image. A drop down menu allows the user to choose packages from a list of predefined *yum* repositories. Using asynchronous Java Script and XML, a further list is returned which allows users to select the corresponding *yum* packages. Support for request via XML descriptions is also available. An image description can be uploaded through the Web interface of the OS Farm server as an XML file. A sample XML description is shown below:

```
<image>
   <name>myvm</name>
   <class>slc4_old</class>
   <architecture>i386</architecture>
   <package>emacs</package>
   <package>unzip</package>
   <group>Base</group>
   <group>Core</group>
   </image>
```

  HTTP interface

  The OS Farm also provides the HTTP interface. For example, users can use the *wget* to access the HTTP interface of the OS Farm server.

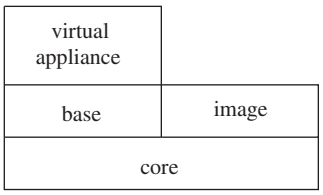

Fig. 2. Layers of the OS Farm image generation.

The images in the OS Farm are generated in layers which can be cached. The generation of a virtual machine image is divided into three layers or stages. Fig. 2 shows the layers of virtual machine image generation:

- The *core* layer is a small functional image with a minimal set of software which is either required to run the image on a VMM or required to satisfy higher level software dependencies.
- The *base* layer provides some additional software needed in order to satisfy requirements for *virtual appliances*.
- the *image* layer is on top of *core* layer and provides user-defined software in addition to the core software.
- A subclass of image is *virtual appliance*, which is an image with an extra set of rules aimed to allow the image to satisfy requirements of the deployment scenario.

3. Definition of Cloud computing

Cloud computing is becoming one of the next IT industry buzz words: users move out their data and applications to the remote "Cloud" and then access them in a simple and pervasive way. This is again a central processing use case. Similar scenario occurred around 50 years ago: a time-sharing computing server served multiple users. Until 20 years ago when personal computers came to us, data and programs were mostly located in local resources. Certainly currently the Cloud computing paradigm is not a recurrence of the history. A 50 years ago we had to adopt the time-sharing servers due to limited computing resources. Nowadays the Cloud computing comes into fashion due to the need to build complex IT infrastructures. Users have to manage various software installations, configuration and updates. Computing resources and other hardware are prone to be outdated very soon. Therefore outsourcing computing platforms is a smart solution for users to handle complex IT infrastructures.

At the current stage, the Cloud computing is still evolving and there exists no widely accepted definition. Based on our experience, we propose an early definition of Cloud computing as follows:

> *A computing Cloud is a set of network enabled services, providing scalable, QoS guaranteed, normally personalized, inexpensive computing infrastructures on demand, which could be accessed in a simple and pervasive way.*

3.1. HaaS: Hardware as a Service

Hardware as a Service was coined possibly in 2006. As the result of rapid advances in hardware virtualization, IT automation and