

# The Cumulus Project: Build a Scientific Cloud for a Data Center

Lizhe Wang, Jie Tao, Marcel Kunze, Dharminder Rattu  
Steinbuch Centre for Computing (SCC), Research Center Karlsruhe (FZK)  
Hermann-von-Helmholtz-Platz 1, 76344 Eggenstein-Leopoldshafen, Germany

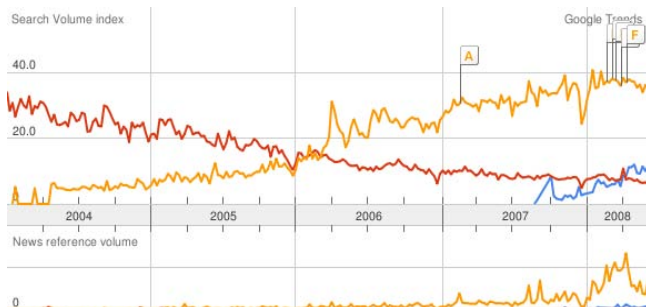
Alvaro Canales Castellanos  
Department of Computer Science, University Karlsruhe (TH)  
76128 Karlsruhe, Germany

## Abstract

*Cloud computing emerges as a new computing paradigm which aims to provide reliable, customized and QoS guaranteed dynamic computing environments for end-users. This paper reviews our early experience of Cloud computing based on the Cumulus project for data centers. In this paper, we introduce the Cumulus project with its various aspects, such as testbed, infrastructure, middleware and application models.*

## 1 Introduction

Cloud computing currently emerges as a hot topic due to its abilities to offer flexible dynamic IT infrastructures, QoS guaranteed computing environments and configurable software services. As reported in Google trends (Figure 1), Cloud computing (blue line), which is enabled by Virtualization technology (yellow line), has already outpaced Grid computing by far [8] (red line).



**Figure 1. Cloud computing in Google trends**

Numerous projects in industry and academia have already started, for example, the RESERVOIR project [31] - an IBM and European Union joint research initiative for

Cloud computing, Amazon Elastic Compute Cloud [21], IBM's Blue Cloud [17], scientific Cloud projects such as Nimbus [26] and Stratus [34], and OpenNEBula [29]. HP, Intel Corporation and Yahoo! Inc. recently announced the creation of a global, multi-data center, open source Cloud computing test bed for industry, research and education [12, 23].

In this paper we present the Cumulus Cloud computing project. The Cumulus project is intended to build a scientific cloud for data centers by merging existing Grid infrastructures with new Cloud technologies. This paper discusses the infrastructure, middleware and application models for the Cumulus project and is organized as follows. Section 2 discusses the definition of Cloud computing in terms of concepts, characterization and enabling technologies. In Section 3 the Cumulus project is presented. Section 4 concludes the paper.

## 2 Cloud computing: Definition, characterization and enabling technologies

### 2.1 Cloud computing, a recurrence of the history?

Cloud Computing is becoming one of the next IT industry buzz words: users move out their data and applications to 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: time-sharing computing server served for multiple users. Until 20 years ago when personal computers came to us, data and programs were mostly again located in local resources. Certainly currently Cloud computing paradigm is not a recurrence of the history. 50 years ago we had to adopt the time-sharing server due to limited computing resource. Nowadays Cloud computing comes into fashion due to the complex IT infrastructure. Users have to manage various

software installation, configuration and update. Computing resource and other hardware are prone to be outdated. Therefore outsourcing computing platforms is a natural solution for users to handle complex IT infrastructures.

At the current stage, 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:

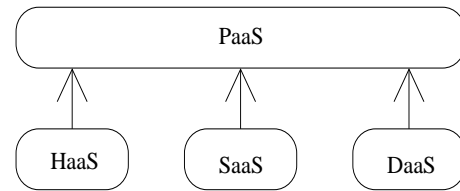
*A computing Cloud is a set of network enabled services, providing scalable, QoS guaranteed, normally personalized, inexpensive computing platforms on demand, which could be accessed in a simple and pervasive way.*

## 2.2 Functionalities

Computing clouds render users with services to access hardware, software and data resources in transparent ways:

- **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 usage metering and pricing, users could buy IT hardware - or even an entire data center - as a pay-as-you-go subscription service. The HaaS is flexible, scalable and manageable to meet your needs [2].
- **SaaS: Software as a Service**  
Software or an application is hosted as a service and provided to customers across the Internet. This mode eliminates the need to install and run the application on the customer's local computer. SaaS therefore alleviates the customer's burden of software maintenance, and reduces the expense of software purchases by on-demand pricing.
- **DaaS: Data as a Service**  
Data in various formats and from various sources could be accessed via services by users on the network, in a transparent, logical or semantic way. Users could, for example, manipulate remote data just like operate on a local disk or access data in a semantic way in the Internet.

Based on the support of HaaS, SaaS, and DaaS, Cloud computing in addition can deliver Platform as a Service (PaaS) for users. Users thus can on-demand subscribe to their favorite computing platforms with requirements of hardware configuration, software installation and data access demands. Figure 2 shows the relationship between the services.



**Figure 2. Cloud functionalities**

## 2.3 Key features

Cloud computing distinguishes itself from other computing paradigms, like Grid computing [8], Global computing [7], Internet Computing [14] in the following aspects:

- **User-centric interfaces**  
Cloud services could be accessed with user-centric interfaces:
  - The Cloud interfaces do not force users to change their working habits, e.g., programming language, compiler, or operating system.
  - The Cloud client which is required to be installed locally is lightweight. For example, the Nimbus Cloudkit client size is around 15 MB.
  - Cloud interfaces are location independent and can be accessed by some well established interfaces like Web services framework or Internet browser.
- **On-demand service provision**  
Computing Clouds provide resources and services for users on demand. Users can customize their computing environments later on, for example, software installation, network configuration, as users usually own administrative privileges.
- **QoS guaranteed offer**  
The computing environments provided by computing Clouds can guarantee QoS for users, e.g., hardware performance like CPU speed, I/O bandwidth and memory size.
- **Autonomous System**  
The computing Cloud is an autonomous system and managed transparently to Cloud users. Hardware, software and data inside Clouds can be automatically re-configured, orchestrated and consolidated to a single platform image, finally rendered to Cloud users.

## 2.4 Enabling technologies

A lot of enabling technologies contribute to the Cloud computing, here we identify several state-of-the-art techniques:

- **Virtualization**  
Virtualization technologies partition hardware and thus providing flexible and scalable computing platforms. Virtual machine techniques, such as VMware [37] and Xen [1], offer virtualized IT-infrastructures on demand. Virtual network advances, such as VPN [6], support users with a customized network environment to access cloud resources.
- **Serviceflow and workflow orchestration**  
Computing Clouds offer a complete set of service templates on demand, which could be composed by services inside the Cloud. Cloud should be able to automatically orchestrate services from different sources and of different types to form a serviceflow or workflow for users.
- **Web service and SOA**  
Cloud services are normally exposed as Web services, which follow the industry standards like WSDL [36], SOAP [32] and UDDI [27]. The service organization and orchestration inside clouds could be managed in a Service Oriented Architecture (SOA). A set of Cloud services furthermore could be organized in a SOA, make themselves available on various distributed platforms and could thus be accessed across networks.
- **Web 2.0**  
The Web 2.0 describes the trend in the use of World Wide Web technology and Web design to enhance creativity, information sharing, and, most notably, collaboration among users. These concepts have led to the development and evolution of Web-based communities and hosted services [5].  
The essential idea behind Web 2.0 is to improve the interconnectivity and interactivity of Web applications. The new paradigm to develop and access Web applications enables users access the Web more easily and efficiently. Cloud computing services in nature are Web applications. It is thus a natural technical merging that Cloud computing adopts the Web 2.0 techniques.
- **World-wide distributed storage system**  
A Cloud storage model should foresee:
  - A network storage system, which is backed by distributed storage providers (e.g., data centers), offers storage capacity for users to lease. The data storage could be migrated, merged, and managed transparently to end users for whatever data formats. Examples are Google File System [11] and Amazon S3 [22]. A Mashup is a Web application that combines data from more than one source into a single integrated storage tool [20]. SmugMug [33] is an example of

Mashup, which is a digital photo sharing Web site, allowing the upload of an unlimited number of photos for all account types, providing a published API which allows programmers to create new functionality, and supporting XML-based RSS and Atom feeds.

- A distributed data system which provides data sources accessed in a semantic way. Users could locate data sources in a large distributed environment by the logical name instead of physical locations. Virtual Data System (VDS) [35] could be good reference.

- **Programming model**  
Users drive into the computing Cloud with data and applications. Some Cloud programming models should be proposed for users to adapt to the Cloud infrastructure. To the simplicity and easy access of Cloud services, the Cloud programming model, however, should not be too complex or too innovative for end users.

MapReduce [3, 4] is a programming model and an associated implementation for processing and generating large data sets across the Google worldwide infrastructures. The MapReduce model firstly involves applying a “map” operation to some data records – a set of key/value pairs, and then processes a “reduce” operation to all the values that shared the same key. The Map-Reduce-Merge [38] method evolves the MapReduce paradigm by adding a “merge” operation. Hadoop [25] is a framework for running applications on large clusters built of commodity hardware. It implements the MapReduce paradigm and provides a distributed file system - Hadoop Distributed File System. The MapReduce and Hadoop are adopted by recently created international Cloud computing project of Yahoo!, Intel and HP [12, 23].

### 3 Cumulus: A scientific Cloud

#### 3.1 Overview

The Cumulus project is an on-going Cloud computing project at the recently established Steinbuch Centre for Computing (SCC) at the Karlsruhe Institute of Technology (KIT). It intends to provide virtual machines, virtual applications and virtual computing platforms for scientific computing applications.

The Cumulus project currently is running on high performance HP and IBM blade servers with Linux and the Xen hypervisor. Cumulus aims to build a testbed and infrastructure in the context mentioned above and integrates mainly already existing technology.

We design the Cumulus in a layered architecture (see also Figure 3):

- The Cumulus frontend service resides on the access point of Cumulus and accepts users' requirements of virtual machine operation.
- The OpenNEBula works as Local Virtualization Management System (LVMS). The frontend of OpenNEBula communicates with the Globus virtual workspace service via SSH.
- The OpenNEBula frontend communicates to its backends and Xen hypervisors on the hosts via SSH for virtual machine manipulation.
- OS Farm  
OS Farm [18] is a service for generating and storing Xen VM images and Virtual Appliances. We use OS Farm as a tool for virtual machine template management.

This design philosophy could bring scalability on the one hand and keep the autonomy of data centers. The Cumulus frontend does not depend on a specific Local Virtualization Management System. The data center inside the Cloud could define its own resource management policy such as IP address lease or virtual machine resource allocation.

### 3.2 Cumulus frontend: re-enforcement of the Globus virtual workspace service

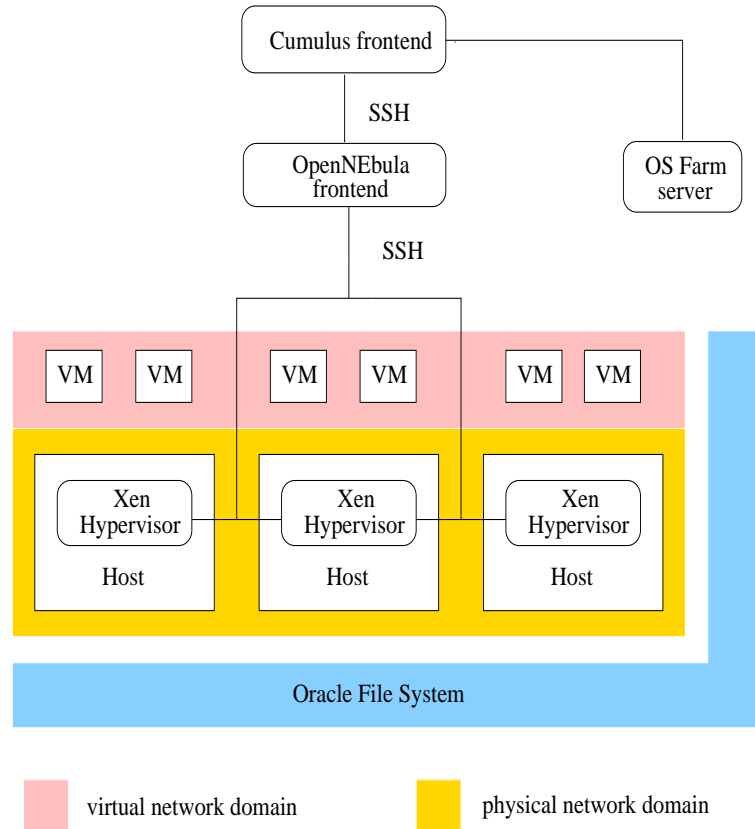
The Globus virtual workspace service contains two pieces of software:

- the workspace service frontend, and
- the workspace control agent.

The workspace service frontend receives the virtual machine requirements and distributes them to various backend servers. The workspace control agent is installed on each backend server and talks with the Xen server in order to deploy a workspace.

In principal the Globus virtual workspace service could be employed as a Cloud service frontend. However we identified some limitations of the Globus virtual workspace service:

- Computer Data centers in general run their own specific Local Virtualization Management Systems (LVMS), like OpenNEBula or VMware Virtual Infrastructure, to manage their local infrastructures. The Globus virtual workspace service however demands to install the workspace control agents directly on the backend servers. This use scenario provided by Globus workspace service lacks the generality.



**Figure 3. Cumulus architecture**

- Globus virtual workspace service provides three network settings: the *AcceptAndConfigure* mode, the *AllocateAndConfigure* mode and the *Advisory* mode. In general users do not care about network configuration for virtual machines. Users only demand the virtual machine IP address in order to access virtual machines. We suggest that a network IP address lease is associated with virtual machine resource allocation within clouds. The network setting could be provided by the backends of a Cloud service and thus be transparent to end users.

We re-engineered the Globus virtual workspace service to adapt it as a Cumulus frontend service. This comprised the following steps:

- Remove the control agent such that the Globus virtual workspace service could talk directly to the virtual machine hypervisors installed on the backend servers.
- Extend the Globus frontend service and make it work with various virtual machine hypervisors and LVMS, such as OpenNEBula, VMware server and VMware Virtual Infrastructure.

- Support a new networking solution – the *forward* mode: users don't need to input network configuration information, the backend servers allocate the IP address for the virtual machine and return it to the users.

### 3.3 OpenNEbula as Local Virtualization Management System

The OpenNEbula is used to manage our distributed blade servers. It provides the resources for virtual machine deployment. Currently the OpenNEbula employs NIS (Network Information System) to manage a common user system and NFS (Network File System) for OpenNEbula shared directory management. However it has been widely recognized that NIS has a major security flaw: it leaves the users' password file accessible by anyone on the entire network. To employ OpenNEbula in a more professional way, we merged OpenNEbula with some modern secure infrastructure solutions like LDAP [13] and Oracle Cluster File System [30].

### 3.4 OS Farm for virtual machine image management

We based our virtual machine image repository on the OS Farm service. OS Farm renders two interfaces for Cloud users:

- A Web interface where users can input parameters for virtual machine image construction, and
- A HTTP service which could be accessed via *wget*, for example:

```
wget
http://www.fzk.de/osfarm/create?name=
&transfer=http&class=slc_old&arch=i386
&filetype=.tar&group=core&group=base
```

We have built an OS Farm client and embedded it into the Cumulus frontend. When users require virtual machine images, the Cumulus frontend service invokes a *wget* command to generate virtual machine images for users. This implementation frees users from the pain to generate and submit virtual machine images manually.

### 3.5 Networking solution

We provide a new networking solution: the *“forward”* mode. Users do not have to specify anything for the network requirements. The OpenNEbula starts the Xen virtual machine images, allocates dynamic IP addresses for the virtual machines and then returns them to the users. In addition the backend servers could implement some complex

network management policies and return IP addresses to the users. The new networking solution is transparent to users, thus it is named as the *“forward”* mode.

### 3.6 Access to the Cumulus service

The Cumulus service provides the following access methods for users:

- via Globus virtual workspace service client or Nimbus cloudkit client,
- with Grid computing workbench or existing Grid portals, like g-Eclipse [10], Open Grid Computing Environment [15] and GridShell [16].

The g-Eclipse project [9] is an international research effort, which aims to build an integrated workbench framework to leverage the power of existing Grid infrastructures. The g-Eclipse project plans to extend itself to access our Cumulus Cloud service by the following steps:

1. create a VO for the Cumulus user,
2. provide a Grid credential,
3. browse the available virtual machine images,
4. launch the desired virtual machines,
5. access the virtual machine.

We also expect to integrate the OS Farm into the g-Eclipse platform using its generic Grid connection concept. In the g-Eclipse project, a connection allows the user to link to a local or remote file system. To establish a connection, users need only to provide the necessary information to describe the connection.

### 3.7 Typical use cases

In the following we present several typical use cases and patterns for Cumulus usage:

- GridKa school training support  
Research Center Karlsruhe yearly holds the GridKa school, which is a training session covering topics related to Scientific Grid Computing. In GridKa school, more than 100 participants take courses and practice various Grid middlewares. To manage such a scenario, the server blades inside Cumulus are required to provide several hundreds identical virtual machines with some pre-installed software. Each virtual machine is allocated a dynamic IP address and the trainees are able to access their own virtual machine directly.

- Virtual e-Science infrastructure  
Users could dynamically organize their customized e-Science infrastructure on the distributed virtual machines provided by Cumulus. The OS Farm server provides prototype virtual machines which are then enriched with the requested software configuration, for example, an e-Science infrastructure could contain GridSAM [24] as Grid middleware, ActiveBPEL engine [28] as workflow engine to build a workflow system. With these pre-configured virtual machine images, users could build an e-Science infrastructure at run-time by multiple service orchestration and invocation.
- Virtual computing center  
Users could lease a large set of virtual machines from the Cumulus, deploy a middleware on the distributed virtual machines to build a virtual computing center, for example, a complete UNICORE [19] installation. A virtual computing center functions in the same way as a real computing center. It provides portals or Web services via which users can access computing resources in the virtual computing center. The builder of the virtual computing center, who normally works as the administrator of the virtual center, provides application level software and libraries on the computing resources and defines the management policies and access control rules.

## 4 Conclusion

This paper reviews the recent advances of Cloud computing and presents our early results from the Cumulus Cloud computing project. The Cumulus project aims to build a scalable scientific cloud: Cumulus could adaptive to various virtual machine hypervisors as backend, furthermore delegates users' requirements to other Clouds. This objective defines the resource re-organization and service orchestration for Cloud computing.

## References

- [1] P. Barham, B. Dragovic, K. Fraser, S. Hand, T. L. Harris, A. Ho, R. Neugebauer, I. Pratt, and A. Warfield. Xen and the art of virtualization. In *Proceedings of the 19th ACM Symposium on Operating Systems Principles*, pages 164–177, New York, U. S. A., Oct. 2003.
- [2] Here comes HaaS [URL]. [http://www.roughtype.com/archives/2006/03/here\\_comes\\_haas.php](http://www.roughtype.com/archives/2006/03/here_comes_haas.php), access on June 2008.
- [3] J. Dean. Mapreduce and other building blocks for large-scale distributed systems at google. In *Proceedings of the USENIX Annual Technical Conference*, 2007.
- [4] J. Dean and S. Ghemawat. Mapreduce: simplified data processing on large clusters. *Commun. ACM*, 51(1):107–113, 2008.
- [5] Web 2.0 definition [URL]. [http://en.wikipedia.org/wiki/web\\_2/](http://en.wikipedia.org/wiki/web_2/), access on June 2008.
- [6] B. Gleeson etc. A framework for ip based virtual private networks. Rfc2764, The Internet Engineering Task Force, Feb. 2000.
- [7] G. Fedak, C. Germain, V. Néri, and F. Cappello. Xtremweb: A generic global computing system. In *Proceedings of the 1st IEEE International Symposium on Cluster Computing and the Grid*, pages 582–587, 2001.
- [8] I. Foster and C. Kesselman. *The grid: blueprint for a new computing infrastructure*. Morgan Kaufmann, 1998.
- [9] g-Eclipse project [URL]. <http://www.geclipse.org/>, access on Oct. 2008.
- [10] The g-Eclipse project [URL]. <http://www.geclipse.org/>, access on July 2008.
- [11] S. Ghemawat, H. Gobioff, and S. Leung. The google file system. In *Proceedings of the 19th ACM Symposium on Operating Systems Principles*, pages 29–43, 2003.
- [12] Intel HP and Yahoo! Create Global Cloud Computing Research Test Bed [URL]. <http://www.hp.com/hpinfo/newsroom/press/2008/080729xa.html>, access on Sep. 2008.
- [13] V. A. Koutsonikola and A. Vakali. LDAP: framework, practices, and trends. *IEEE Internet Computing*, 8(5):66–72, 2004.
- [14] M. Milenkovic, S. H. Robinson, R. C. Knauerhase, D. Barkai, S. Garg, V. Tewari, T. A. Anderson, and M. Bowman. Toward internet distributed computing. *IEEE Computer*, 36(5):38–46, 2003.
- [15] The Open Grid Computing Environments Portal and Gateway Toolkit [URL]. <http://www.collab-ogce.org>, access on July 2008.
- [16] Grid Shell project [URL]. <http://grid.rit.edu/wiki/doku.php/grid:shell/>, access on July 2008.

- [17] IBM Blue Cloud project [URL]. <http://www-03.ibm.com/press/us/en/pressrelease/22613.wss/>, access on June 2008.
- [18] OA Farm project [URL]. <http://cern.ch/osfarm/>, access on July 2008.
- [19] Unicore project [URL]. <http://www.unicore.org/>, access on Nov. 2007.
- [20] Mashup project [URL]. [http://en.wikipedia.org/wiki/mashup\\_web\\_application\\_hybrid/](http://en.wikipedia.org/wiki/mashup_web_application_hybrid/), access on June 2008.
- [21] Amazon Elastic Compute Cloud [URL]. <http://aws.amazon.com/ec2/>, access on Nov. 2007.
- [22] Amazon Simple Storage Service [URL]. <http://aws.amazon.com/s3/>, access on Sep. 2008.
- [23] Global Cloud Computing Research Test Bed Wiki [URL]. <http://cloudtestbed.org/>, access on Oct. 2008.
- [24] GridSAM [URL]. <http://gridsam.sourceforge.net/>, access on Nov. 2007.
- [25] Hadoop [URL]. <http://hadoop.apache.org/>, access on Sep. 2008.
- [26] Nimbus Project [URL]. <http://workspace.globus.org/clouds/nimbus.html/>, access on June 2008.
- [27] OASIS UDDI Specification [URL]. <http://www.oasis-open.org/committees/uddi-spec/doc/tcspecs.htm>, access on June 2008.
- [28] OMII-BPEL [URL]. <http://sse.cs.ucl.ac.uk/omii-bpel/>, access on Nov. 2007.
- [29] OpenNebula Project [URL]. <http://www.opennebula.org/>, access on Apr. 2008.
- [30] Oracle Cluster File System [URL]. <http://oss.oracle.com/projects/ocfs/>, access on June 2008.
- [31] Reservoir Project [URL]. <http://www-03.ibm.com/press/us/en/pressrelease/23448.wss/>, access on June 2008.
- [32] Simple Object Access Protocol (SOAP) [URL]. <http://www.w3.org/tr/soap/>, access on Nov. 2007.
- [33] SmugMug [URL]. <http://www.smugmug.com/>, access on June 2008.
- [34] Status Project [URL]. <http://www.acis.ufl.edu/vws/>, access on June 2008.
- [35] Virtual Data System [URL]. <http://vds.uchicago.edu/>, access on Nov. 2007.
- [36] Web Service Description Language (WSDL) [URL]. <http://www.w3.org/tr/wsdl/>, access on Nov. 2007.
- [37] VMware virtualization technology [URL]. <http://www.vmware.com/>, access on Nov. 2007.
- [38] H. Yang, A. Dasdan, R. Hsiao, and D. S. Parker Jr. Map-reduce-merge: simplified relational data processing on large clusters. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pages 1029–1040, 2007.