

Science Clouds: Early Experiences in Cloud Computing for Scientific Applications

K. Keahey¹, R. Figueiredo², J. Fortes², T. Freeman¹, M. Tsugawa²

¹University of Chicago, ²University of Florida

Abstract

The Science Clouds provide EC2-style cycles to scientific projects. This document contains a description of technologies enabling this project and an early summary of its experiences.

1 Introduction

The Science Clouds project was initiated by the University of Chicago (UC) and the University of Florida (UFL) as a by-product of the NSF SDCI “Missing Links” project with two objectives:

- Make it easy for scientific and educational projects to experiment with EC2-style cloud computing, and
- Better understand the potential and challenges that cloud computing poses for these communities and what can be done to overcome them

The Science Clouds project allows members of the scientific community to lease resources for short amounts of time, in a manner similar to Amazon’s EC2 service [1]: a client requests a resource lease for a few hours and, if the request is authorized, a virtual machine (VM) is deployed. The client can then use the VM as needed (e.g., ssh to it, move data to it, or run computations) for the requested time. The power of this model lies in the fact that the client is allowed to bring a VM configured to his/her exact specifications and is given an exclusive ownership of the leased resource (the VM) to be shared with others only at the client’s discretion. Unlike the EC2 service, the Science Clouds do not require users to directly pay for usage. Instead, we loosely verify that the person asking for an allocation is indeed a member of the scientific community (through verifying an email account with the sponsoring institution, web pages, pointers to papers, etc.) and ask for a short writeup of the scientific project. Based on the project the individual is allocated a small (testing), middle (development), or large (science) hour credit on the Science Clouds.

The first cloud, at the University of Chicago, became available on March 3, 2008, and was named “nimbus” [2] (the name was eventually adopted by our software project). It was deployed on a partition of 16 nodes of the TeraPort cluster [3] (each node composed of two 2.2 GHz AMD64 processors, 4 GB RAM, and 80 GB local disk). The Chicago cloud allocated 16 public IPs to implement the VM leases and originally provided 100 GB of storage space (we recently purchased a 500 GB disk to accommodate the raising traffic). The University of Florida cloud [4], made available on May 13, 2008, offers 16 nodes, each of which has two Intel Xeon/Prestonia 2.40 GHz processors, 3GB of RAM and 18GB of local disk. VM storage is backed by a 3.4TB Fibre-channel storage server. The UFL cloud configuration contains an innovation: private IP addresses are used in deployed VMs, and network virtualization is used to connect VMs to the client/owner machines. The use of virtual networks opens the possibility of deploying cross-domain, wide-area virtual clusters, seamlessly overcoming connectivity limitations imposed by wide-area links (e.g., firewalls and NATs). Both Science Clouds were configured with the nimbus toolkit [5] to enable remote leasing of resources via VMs, and both were configured to support lease semantics that corresponded to EC2 “immediate leases”: a request either results in

immediate VM deployment or is rejected.

The following presentation describes the configuration and summarizes our early experiences with the Science Cloud testbed.

2 The CloudKit: Deploying and Configuring the Clouds

In this section we describe tools that were used to configure the Science Clouds and plan to explore in the future.

2.1 The Nimbus CloudKit

Nimbus toolkit (formerly known as the virtual workspace service) was developed with the goal of providing an *open source* implementation of a service that allows a client to lease remote resources by deploying VMs onto those resources. The primary objectives of nimbus are to provide infrastructure semantics addressing the needs of the *scientific community*, in particular, through resource leases. The first version of the workspace service was released in September 2005 after roughly two years of R&D. As its functionality grew, we decided to make the service available as a set of components (since version 1.3); and acknowledging that it was no longer just one service, we changed the name to nimbus toolkit (since version 2.0).

Today, the nimbus toolkit consists of the following components (Figure 3 shows their dependency graph):

- *Workspace service*, which allows a remote client to deploy and manage flexibly defined groups of VMs. The service is composed of a WS front-end and a VM-based resource manager deployed on a site. The workspace service supports two front-ends: one based on the Web Service Resource Framework (WSRF) [6], and one based on Amazon's EC2 WSDL.
- *Workspace resource manager*, which implements deployment of VM leases with "immediate" semantics on a site infrastructure.
- *Workspace pilot*, which extends existing local resource managers (LRMs) such as Torque [7] or SGE [8] to deploy virtual machines to allows RPs to use virtualization without significantly altering the site configuration.
- The *workspace control* tools, which are used to start, stop, and pause VMs; implement VM image reconstruction and management; connect the VMs to the network; and deliver contextualization information (currently work with Xen [9] and KVM [10]).
- *Infrastructure-as-a-Service (IaaS) gateway*, which allows a client presenting a PKI credential to use another IaaS infrastructure (with different credentials). The IaaS gateway is currently used to map between X509 PKI credentials and EC2 accounts for specific projects and to enable scientific projects to run on Amazon EC2.
- *Context broker*, which allows a client to deploy a "one-click" functioning virtual cluster as opposed to a set of "unconnected" virtual machines as well as "personalize" VMs (i.e.,

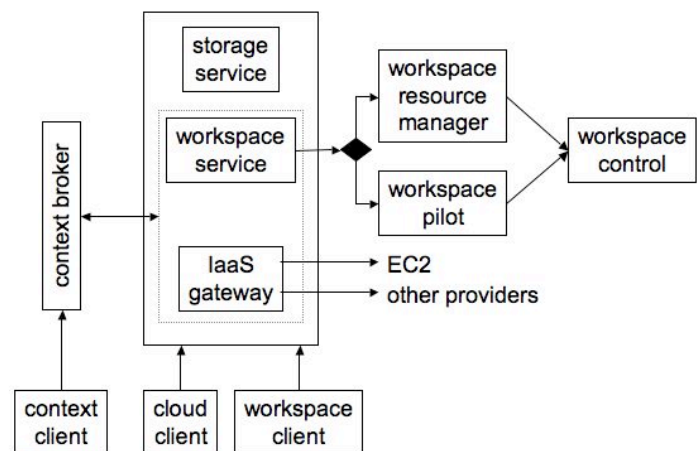


Figure 3: Dependency graph of the nimbus toolkit

seed them with secrets).

- *Workspace client*, which provides full access to workspace service functionality (in particular, a rich set of networking options) but is relatively complex to use and thus typically wrapped by community-specific scripts.
- *Cloud client*, which provides access to only a select set of functions but is very easy to use and popular as an end-user tool.
- *Nimbus storage service*, which provides secure management of cloud disk space giving each user a “repository” view of VM images they own and images they can launch. It works in conjunction with globus GridFTP [11], which allows us to support any network file system, SAN, and so forth that GridFTP can interface to (this includes the possibility of drawing from parallel file sources).

2.2 Virtualizing the Network

Communication between resources is of key importance in cloud deployments; however the presence of firewalls and NATs in the current IPv4 Internet limits the connectivity among VM host servers and deployed VMs, especially when crossing LAN boundaries. This problem is exacerbated by the fact that not all sites are willing or able to assign blocks of public IPs for use by VMs. Network virtualization technologies can overcome such limitations, and allow:

- on-demand creation/deployment/tearing-down of multiple isolated virtual networks, enabling the creation of virtual private clusters on a per-user basis.
- the deployment of cross-domain virtual clusters offering full connectivity to the VMs connected in different private networks;

Using the workspace service in conjunction with the VPN technology allowed us to use the workspace service to deploy VMs on a private network. The UFL cloud is deployed with a VPN server, which accepts the same X.509 PKI credentials that end-users present to the workspace service front-end: thus, the operation of creating a grid proxy was simply combined with the operation of joining the private network. End-users do not need to keep independent credentials for cloud services and virtual networks. Likewise, RPs do not need to manage different identifications.

More sophisticated network virtualization is represented by the ViNe network overlay project [20]. ViNe has been used to deploy a 32-node virtual cluster, spanning resources in both UC and UFL resources, such that all VMs participating in the clusters were in the same address space. To do this, we deployed a VM containing ViNe router in each location: these routers were used to orchestrate packet forwarding between sites. Two distinct characteristics of ViNe make it appealing for clouds: its low network virtualization overhead and its management oriented architecture that can offer virtual network services to nimbus.

There are several possible VPN-based approaches to provide user accessibility to Science Clouds. The choice of an appropriate approach depends on the policies that cloud providers and users are comfortable with, and these may vary across different science domains. One of the techniques we have developed targets lowering the complexity in management and use, with the goals of making it very simple for cloud providers to manage a VPN infrastructure and also simple for users to access the service.

To address accessibility, we build upon user-friendly interfaces exposed of Web-based social networking infrastructures (such as FaceBook, LinkedIn) to facilitate the creation of user and group relationships, and the management of host certificates. With this technique, a Science Cloud provider can deploy a VPN simply by creating and managing a moderated user group in a

social network infrastructure. Users can download a "Social VPN" virtual router and request to join the Science Cloud group; when they are authorized by the group moderator, the process of key generation, certificate signing, and VPN tunneling happens automatically, without user intervention, even if the front-end service to the Cloud is behind a NAT or firewall. We have developed an open-source SocialVPN application for Windows and Linux, which is based on the Facebook APIs and the IPOP virtual network [21,22]; the application is currently being extended to handle management tasks including certificate revocation, as well as being ported to MacOS.

3 Cloud Utilization

The Science Clouds have been in operation for five months. In the following, we present information about how the clouds were used, what applications they attracted, and the usage patterns we observed. The data discussed is based on utilization numbers from the University of Chicago cloud observed from March 3 to August 4, 2008.

Both the number of users and the time they have been spending on the cloud have risen significantly over the past five months. As of the time of this writing, we have 60+ users authorized to use the cloud, and new requests from scientific application projects worldwide come every week. The overall utilization of the clouds was around 20% (see Figure 1), with the peak per-week utilization of 86% reached in the second half of July (week of 07/14). This is remarkable utilization considering that immediate leases on a small resource do not lend themselves to a very efficient use of resources, an issue that we are working to resolve in our research [12]. One interesting measure of utilization is the number of lease requests that were rejected with the "cluster full" message: virtually no requests were rejected before 07/14. In the period after 07/14, a total of 65 requests were rejected.

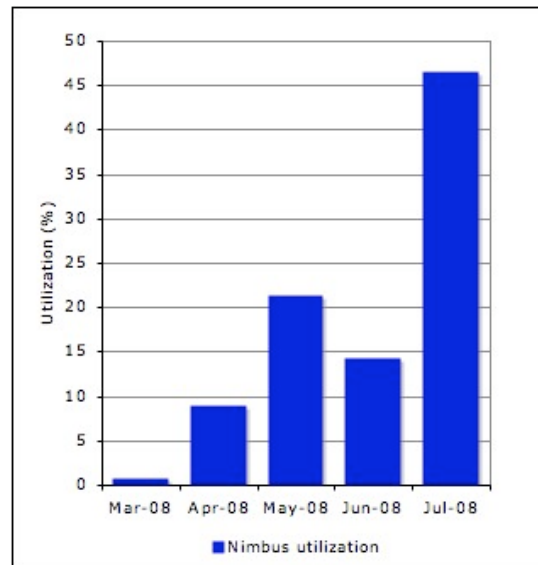


Figure 1: Utilization of the Chicago cloud per month: the graph shows utilization from 3rd to 3rd of every month scaled to the number of days with the month.

It is no coincidence that utilization increased significantly in mid-July. On 07/09 the nimbus team released and

integrated into the cloud deployment the context broker that allows users to create one-click virtual clusters. This enabled new applications (such as the Alice high-energy physics experiment [13] and Montage workflow testing [14]) to run and old ones to run in new configurations.

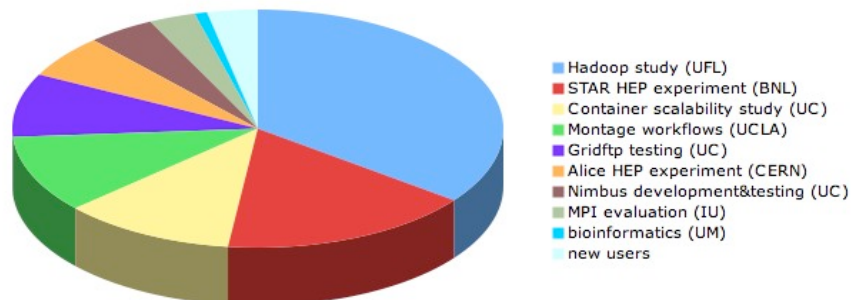


Figure 2: Per-project utilization of the nimbus science cloud at UC (only projects that spent more than 5000 minutes are called out)

The cloud proved popular among projects as diverse as high-energy physics, computer science, bioinformatics, and more recently economics. It is also being explored for use in educational projects [15]. This diversity is remarkable considering it still is relatively difficult for scientific applications to use cloud computing. Moreover, we are seeing many diverse applications co-existing, for example, interactive sessions with scientific runs. To date, two papers have been written about work using the cloud [14, 16].

Figure 2 shows a per-project breakdown of the overall cloud utilization by various projects in the defined period: the most time has been used by the projects that have been using the cloud the longest: computer science project studying the behavior of Hadoop [17] over distributed environments and the STAR high-energy physics runs [18]. Many new projects came onboard in July, but not all of them have resulted yet in significant usage.

One significant obstacle that prevents projects from considering cloud computing is scarcity of resources: while 16 nodes are sufficient to build proof-of-concept solutions, it is not enough for a typical scientific production run where hundreds of nodes are required. We circumvented this barrier by developing the IaaS gateway that (since June 2007) allows us to run scientific codes on EC2. The gateway enabled the first production run of applications on a virtual STAR cluster on 100 nodes of EC2 in September 2007 [19] (STAR was an alpha tester of the context broker technology for nearly a year now). This led to the development of a pattern where we use the UC cloud for small runs and move sponsored large runs to EC2.

Another approach to engaging more resources is the study [16] which shows how distributed resources can be combined to create a seamless platform for bioinformatics applications (and also server as an underlying platform for data-intensive cloud computing). To this end, the UC and UFL clouds connected by ViNe were used to execute a popular bioinformatics tool called BLAST [23]. A leading MPI version and the newly developed Hadoop version of BLAST were evaluated, in a real non-simulated wide-area cluster environment. Both MPI and Hadoop were able to deliver performance comparable to a local cluster, encouraging the use of the cloud when a large number of VMs is required, even if it is necessary to use resources across multiple domains. As stated in Section 2.2, virtual networks are of key importance in such scenarios.

The “nimbus cloud model” has proved popular among resource providers. Several sites either have already installed nimbus (UFL, Clemson University, University of Victoria (Canada), Vrije University (Amsterdam), ForschungsZentrum Karlsruhe and Masaryk University (Brno)) or have expressed intention of doing so (Indiana University, Purdue, ORNL, and others). Many of those installations were inspired by the nimbus cloud. In fact, the GridFTP and container scalability tests at UC proved so popular that two new private clouds were configured on newly purchased infrastructure to support this mode of usage for internal UC projects.

References

1. *Amazon Elastic Compute Cloud (Amazon EC2)*: <http://www.amazon.com/ec2>.
2. *The Nimbus Cloud*: <http://workspace.globus.org/clouds/nimbus.html>.
3. *The TeraPort Cluster*: http://www.ci.uchicago.edu/research/detail_teraport.php.
4. *The Florida Cloud*: <http://www.acis.ufl.edu/vws/>.
5. *The Nimbus Toolkit*: <http://workspace.globus.org/>.
6. Czajkowski, K., D. Ferguson, I. Foster, J. Frey, S. Graham, I. Sedukhin, D. Snelling, S. Tuecke, and W. Vambenepe, *The WS-Resource Framework*. 2004: www.globus.org/wsrp.
7. *Torque*: <http://www.clusterresources.com/pages/products/torque-resource-manager.php>.

8. *Sun Microsystems Grid Engine.*
9. Barham, P., B. Dragovic, K. Fraser, S. Hand, T. Harris, A. Ho, R. Neugebar, I. Pratt, and A. Warfield. *Xen and the Art of Virtualization.* in *ACM Symposium on Operating Systems Principles (SOSP).*
10. *KVM: Kernel-based Virtual Machine.*
11. Allcock, W., *GridFTP: Protocol Extensions to FTP for the Grid.* 2003, Global Grid Forum.
12. Sotomayor, B., K. Keahey, and I. Foster. *Combining Batch Execution and Leasing Using Virtual Machines.* in *HPDC 2008.* 2008. Boston, MA.
13. *ALICE: A Large Ion Collider Experiment:*
<http://aliceinfo.cern.ch/Public/Welcome.html>.
14. Hoffa, C., T. Freeman, G. Metha, E. Deelman, and K. Keahey, *Exploration of the Applicability of Cloud Computing to Large-Scale Scientific Workflows.* to be submitted to SWBES08: Challenging Issues in Workflow Applications, 2008.
15. Medaris, K., *New specialization will focus on supercomputing:*
<http://www.purdue.edu/uns/x/2007b/071217HackerHPC.html>.
16. Matsunaga, A., M. Tsugawa, and J. Fortes, *CloudBLAST: Combining MapReduce and Virtualization on Distributed Resources for Bioinformatics Applications.* submitted to eScience 2008, 2008.
17. *Hadoop:* <http://hadoop.apache.org/>.
18. *The STAR Experiment.* 2007: www.star.bnl.gov.
19. *The Nimbus RSS News Feed:* <http://workspace.globus.org/news.html>.
20. Tsugawa, M., J. Fortes. *A Virtual Network (ViNe) Architecture for Grid Computing.* in IPDPS 2006.
21. Figueiredo, R., P. Oscar Boykin, Pierre St. Juste, David Wolinsky. *Social VPNs: Integrating Overlay and Social Networks for Seamless P2P Networking.* In IEEE WETICE/COPS 2008.
22. <http://www.socialvpn.org>
23. Altschul, S.F. , et. al. Basic Local Alignment Search Tool. *Journal of Molecular Biology*, 1990