Scientific Workflows and Clouds

By Gideon Juve and Ewa Deelman

In recent years, empirical science has been evolving from physical experimentation to computation-based research. In astronomy, researchers seldom spend time at a telescope, but instead access the large number of image databases that are created and curated by the community [42]. In bioinformatics, data repositories hosted by entities such as the National Institutes of Health [29] provide the data gathered by Genome-Wide Association Studies and enable researchers to link particular genotypes to a variety of diseases.

Besides public data repositories, scientific collaborations maintain community-wide data resources. For example, in gravitational-wave physics, the Laser Interferometer Gravitational-Wave Observatory [3] maintains geographically distributed repositories holding time-series data collected by the instruments and their associated metadata. Along with the large increase in online data, the need to process these data is growing.

In addition to traditional high performance computing (HPC) centers, a nation-wide cyberinfrastructure—a computational environment, usually distributed, that hosts a number of heterogeneous resources; cyberinfrastructure could refer to both grids and clouds or a mix of the two—is being provided to the scientific community, including the Open Science Grid (OSG) [36] and the TeraGrid [47]. These infrastructures, also known as grids [13], allow access to high-performance resources over wide area networks. For example, the TeraGrid is composed of computational and data resources at Indiana University, Louisiana University, University of Illinois, and others. These resources are accessible to users for storing data and performing parallel and sequential computations. They provide remote login access as well as remote data transfer and job scheduling capabilities.

Scientific workflows are used to bring together these various data and compute resources and answer complex research questions. Workflows describe the relationship of the individual computational components and their input and output data in a declarative way. In astronomy, scientists are using workflows to generate science-grade mosaics of the sky [26], to examine the structure of galaxies [46], and, in general, to understand the structure of the universe. In bioinformatics, researchers are using workflows to understand the underpinnings of complex diseases [34, 44]. In earthquake science, workflows are used to predict the magnitude of earthquakes within a geographic area over a period of time [10]. In physics, workflows are used to search for gravitational waves [5] and model the structure of atoms [40]. In ecology, scientists use workflows to explore the issues of biodiversity [21].

Today, workflow applications are running on national and international cyberinfrastructures such as OSG, TeraGrid, and EGEE [11]. The broad spectrum of distributed computing provides unique opportunities for large-scale, complex scientific applications in terms of resource selection, performance optimization, and reliability. In addition to the large-scale cyberinfrastructure, applications can target campus clusters, or utility computing platforms such as commercial [1, 17] and academic clouds [31].

However, these opportunities also bring with them many challenges. It's hard to decide which resources to use and how long they will be needed. It's hard to determine what the cost-benefit tradeoffs are when running in a particular environment. And it's difficult to achieve good performance and reliability for an application on a given system.

Clouds have recently appeared as an option for on-demand computing. Originating in the business sector, clouds can provide computational and storage capacity when needed, which can result in infrastructure savings for a business. One idea driving cloud computing is that businesses can plan only for a sustained level of capacity while reaching out to the cloud for resources in times of peak demand. When using the cloud, consumers pay only for what they use in terms of computational resources, storage, and data transfer in and out of the cloud.

Although clouds were built primarily with business computing needs in mind, they are also being considered in science. In this article we focus primarily on workflow-based scientific applications and describe how they can benefit from the new computing paradigm.

Workflow Applications

Scientific workflows are being used today in a number of disciplines. They stitch together computational tasks so that they can be executed automatically and reliably on behalf of the researcher. These workflows are composed of a number of image-processing applications that discover the geometry of the input images on the sky, calculate the geometry of the output mosaic on the sky, re-project the flux in the input images to conform to the geometry of the output mosaic, model the background radiation in the input images to achieve common flux scales and background levels across the mosaic, and rectify the background that makes all constituent images conform to a common background level. These normalized images are added together to form the final mosaic.

Figure 1 shows a mosaic of the Rho Oph dark cloud created using this workflow.

Montage mosaics can be constructed in different sizes, which dictate the number of images and computational tasks in the workflow. For example, a 4-degree square mosaic (the moon is 0.5 degrees square) corresponds to a workflow with approximately 5,000 tasks and 750 input images. Workflow management systems enable the efficient and reliable execution of these tasks and manage the data products they produce (both intermediate and final).

Figure 2 shows a graphical representation of a small Montage workflow containing 1,200 computational tasks. Workflow management systems such as Pegasus [4, 9, 39] orchestrate the execution of these tasks on desktops, grids, and clouds.

Another example is from the earthquake science domain, where researchers use workflows to generate earthquake hazard maps of

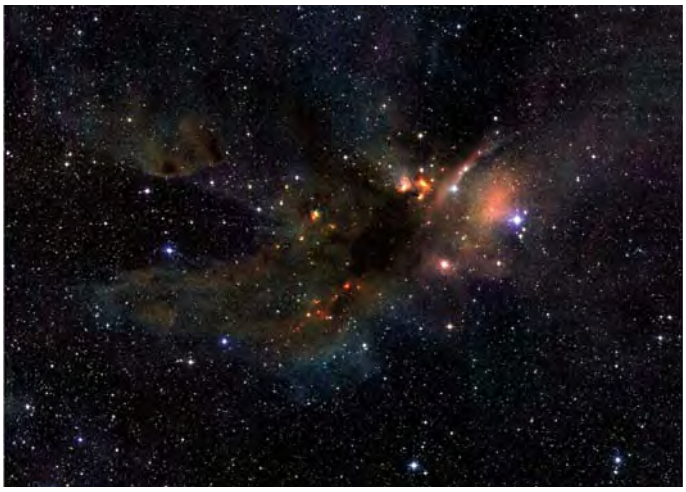

Figure 1: In this 75x90 arcmin view of the Rho Oph dark cloud as seen by 2MASS, the three-color composite is constructed using Montage. J band is shown as blue, H as green, and K as red. (Image courtesy of Bruce Berriman and J. Davy Kirkpatrick.)

Southern California [38]. These maps show the maximum seismic shaking that can be expected to happen in a given region over a period of time (typically 50 years).

Figure 3 shows a map constructed from individual computational points. Each point is obtained from a hazard curve (shown around the map) and each curve is generated by a workflow containing approximately 800,000 to 1,000,000 computational tasks [6]. This application requires large-scale computing capabilities such as those provided by the NSF TeraGrid [47].

In order to support such workflows, software systems need to

1) adapt the workflows to the execution environment (which, by necessity, is often heterogeneous and distributed),

2) optimize workflows for performance to provide a reasonable time to solution,

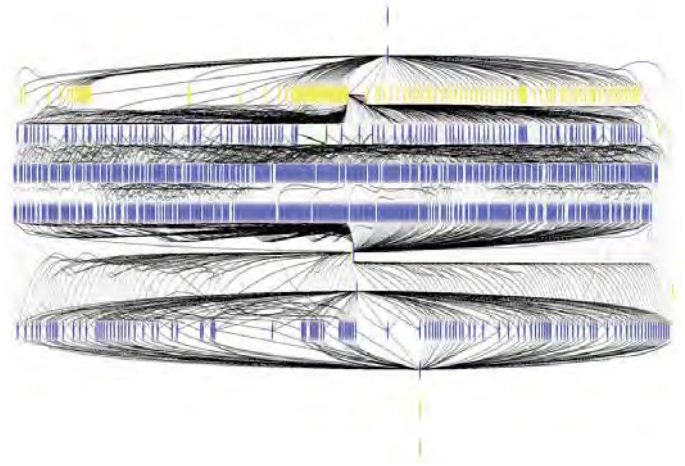

Figure 2: A graphical representation of the Montage workflow with 1,200 computational tasks represented as ovals. The lines connecting the tasks represent data dependencies.

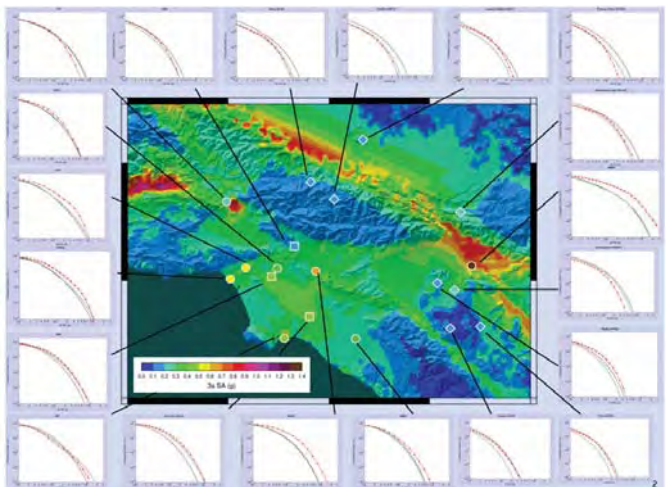

Figure 3: In this shake map of Southern California, points on the map indicate geographic sites where the CyberShake calculations were performed. The curves show the results of the calculations. (Image courtesy of CyberShake Working Group, Southern California Earthquake Center including Scott Callaghan, Kevin Milner, Patrick Small, and Tom Jordan.)

3) provide reliability so that scientists do not have to manage the potentially large numbers of failures, and

4) manage data so that it can be easily found and accessed at the end of the execution.

Science Clouds

Today, clouds are also emerging in academia, providing a limited number of computational platforms on demand: Cumulus [49], Eucalyptus [33], Nimbus [31], OpenNebula [43]. These science clouds provide a great opportunity for researchers to test out their ideas and harden codes before investing more significant resources and money into the potentially larger-scale commercial infrastructure.

To support the needs of a large number of different users with different demands in the software environment, clouds are primarily built using resource virtualization technologies [2, 7, 50] that enable the hosting of a number of different operating systems and associated software and configurations on a single hardware host.

Clouds that provide computational capacities (Amazon EC2 [1], Nimbus, Cumulus) are often referred to as an infrastructure as a service (IaaS) because they provide the basic computing resources needed to deploy applications and services. Platform as a service (PaaS) clouds such as Google App Engine [17] provide an entire application development environment including frameworks, libraries, and a deployment container. Finally, software as a service (SaaS) clouds provide complete end-user applications for tasks such as photo sharing, instant messaging [25], and many others.

Commercial clouds were built with business users in mind, but scientific applications can benefit from them as well. Scientists, however, often have different requirements than enterprise customers. In particular, scientific codes often have parallel components and use MPI [18] or shared memory to manage message-based communication between processors. More coarse-grained parallel applications such as workflows rely on a shared file system to pass data between processes.

Additionally, scientific applications are often composed of many inter-dependent tasks and consume and produce large amounts of data (often in the Terabyte range [5, 10]).

Clouds are similar to grids, in that they can be configured (with additional work and tools) to look like a remote cluster, presenting interfaces for remote job submission and data transfer. As such, scientists can use existing grid software and tools to get their work done.

Another interesting aspect of the cloud is that, by default, it includes resource provisioning as part of the usage mode. Unlike the grid, where jobs are often executed on a best-effort basis, when running on the cloud, a user requests a certain amount of resources and has them dedicated for a given duration of time. How many resources and how fast one can request them is an open question.

Resource provisioning is particularly useful for workflow-based applications, where overheads of scheduling individual, inter-dependent tasks in isolation (as it is done by grid clusters) can be very costly. For example, if there are two dependent jobs in the workflow, the second job will not be released to a local resource manager on the cluster until the first job successfully completes. Thus the second job will incur additional queuing delays. In the provisioned case, as soon as the first job finishes, the second job is released to the local resource manager and since the resource is dedicated, it can be scheduled right away. Thus the overall workflow can be executed much more efficiently.

Virtualization also opens up a greater number of resources to legacy applications. These applications are often very brittle and require a very specific software environment to execute successfully. Today, scientists struggle to make the codes that they rely on for weather prediction, ocean modeling, and many other computations work on different execution sites. No one wants to touch the codes that have been designed and validated many years ago in fear of breaking their scientific quality. Clouds and their use of virtualization technologies may make these legacy codes much easier to run. With virtualization, the environment can be customized with a given OS, libraries, software packages, and the like. The needed directory structure can be created to anchor the application in its preferred location without interfering with other users of the system. The downside is that the environment needs to be created and this may require more knowledge and effort on the part of the scientist than they are willing or able to spend.

Scientific Workflows

The canonical example of a cloud is Amazon's Elastic Compute Cloud (EC2), which is part of Amazon Web Services (AWS). AWS services provide computational, storage, and communication infrastructure on-demand via web-based APIs. AWS offers five major services.

1. *Elastic Compute Cloud* (*EC2*): a service for provisioning virtual machine instances from Amazon's compute cluster, which allows users to deploy virtual machine (VM) images with customized operating systems, libraries, and application code on a variety of predefined hardware configurations (CPU, memory, disk).

2. *Simple Storage Service* (*S3*): an object-based storage system for the reliable storage of binary objects (typically files), which provides operations to "put" and "get" objects from a global object store that is accessible both inside and outside Amazon's cloud.

3. *Elastic Block Store*: a block-based storage system that provides network attached storage volumes to EC2. Volumes can be attached to an EC2 instance as block device and formatted for use as reliable, unshared file system.

4. *Simple Queue Service*: a distributed queue service for sending messages between nodes in a distributed application, which allows messages queued by one node to be retrieved and processed by another.

5. *SimpleDB*: a structured key-value storage service, which enables database records to be stored, indexed and queried by key.

In addition, Amazon's cloud provides services for monitoring (CloudWatch), parallel computing (Elastic MapReduce), relational storage (RDS), and others.

There are many ways to deploy a scientific workflow on a cloud, depending on the services offered by the cloud and the requirements of the workflow management system. Many of the existing workflows were developed for HPC systems such as clusters, grids and supercomputers. Porting these workflows to the cloud involves either adapting the workflow to the cloud or adapting the cloud to the workflow.

Adapting the workflow to the cloud involves changing the workflow to take advantage of cloud-specific services. For example, rather than using a batch scheduler to distribute workflow tasks to cluster nodes, a workflow running on Amazon's cloud could make use of the Simple Queue Service. Adapting the cloud to the workflow involves configuring the cloud to resemble the environment for which the application was created. For example, an HPC cluster can be emulated in Amazon EC2 by provisioning one VM instance to act as a head node running a batch scheduler, and several others to act as worker nodes.

One of the great benefits of the cloud for workflow applications is that both adaptation approaches are possible.

Scientific workflows require large quantities of compute cycles to process tasks. In the cloud, these cycles are provided by virtual machines such as those provided by Amazon EC2. Many virtual machine instances must be used simultaneously to achieve the performance required for large scale workflows. These collections of VMs, called "virtual clusters" [12], can be managed using existing off-the-shelf batch schedulers such as PBS [34, 48] or Condor [8, 24]. Setting up a virtual cluster in the cloud involves complex configuration steps that can be tedious and error-prone. To automate this process, software such as Nimbus Context Broker [22] can be used. This software gathers information about the virtual cluster and uses it to generate configuration files and start services on cluster VMs.

In addition to compute cycles, scientific workflows rely on shared storage systems for communicating data between workflow tasks distributed across a group of nodes, and for storing input and output data. To achieve good performance, these storage systems must scale well to handle data from multiple workflow tasks running in parallel on separate nodes.

When running on HPC systems, workflows can usually make use of a high-performance, parallel file system such as Lustre [45], GPFS [41], or Panasas [37]. In the cloud, workflows can either make use of a cloud storage service, or they can deploy their own shared file system. To use a cloud storage service, the workflow management system would likely need to change the way it manages data. For example, to use Amazon S3, a workflow task needs to fetch input data from S3 to

a local disk, perform its computation, then transfer output data from the local disk back to S3. Making multiple copies in this way can reduce workflow performance.

Another alternative would be to deploy a file system in the cloud that could be used by the workflow. For example, in Amazon EC2, an extra VM can be started to host an NFS file system and worker VMs can mount that file system as a local partition. If better performance is needed then several VMs can be started to host a parallel file system such as PVFS [23, 52] or GlusterFS [16].

Although clouds like Amazon's already provide several good alternatives to HPC systems for workflow computation, communication and storage, there are still challenges to overcome.

Virtualization overhead. Although virtualization provides greater flexibility, it comes with a performance cost. This cost comes from intercepting and simulating certain low-level operating system calls while the VM is running. In addition, there is the overhead of deploying and unpacking VM images before the VM can start. These overheads are critical for scientific workflows because in many cases the entire point of using a workflow is to run a computation in parallel to improve performance. Current estimates put the overhead of existing virtualization software at around 10 percent [2, 15, 51] and VM startup time takes between 15 and 80 seconds depending on the size of the VM image [19, 32]. Fortunately, advances in virtualization technology, such as improved hardware-assisted virtualization, may reduce or eliminate runtime overheads in the future.

Lack of shared or parallel file systems. Although clouds provide many different types of shared storage systems, they are not typically designed for use as file systems. For example, Amazon EBS does not allow volumes to be mounted on multiple instances, and Amazon S3 does not provide a standard file system interface. To run on a cloud like Amazon's, a workflow application must either be modified to use these different storage systems, which takes time, or they must create their own file system using services available in the cloud, which is at least difficult and potentially impossible depending on the file system desired (for example, Lustre cannot be deployed on Amazon EC2 because it requires kernel modifications that EC2 does not allow).

Relatively slow networks. In addition to fast storage systems, scientific workflows rely on high-performance networks to transfer data quickly between tasks running on different hosts. The HPC systems typically used for scientific workflows are built using high-bandwidth, low-latency networks such as InfiniBand [20] and Myrinet [27]. In comparison, most existing commercial clouds are equipped with commodity gigabit Ethernet, which results in poor performance for demanding workflow applications. Fortunately, the use of commodity networking hardware is not a fundamental characteristic of clouds and it should be possible to build clouds with high-performance networks in the future.

Future Outlook

While many scientists can make use of existing clouds that were designed with business users in mind, in the future we are likely to see a great proliferation of clouds that have been designed specifically for science applications. We already see science clouds being deployed at traditional academic computing centers [14, 28, 30]. One can imagine that these science clouds will be similar to existing clouds, but will come equipped with features and services that are even more useful to computational scientists. Like existing clouds, they will potentially come in a variety of flavors depending on the level of abstraction desired by the user.

IaaS science clouds could provide access to the kinds of high-performance infrastructure found in HPC systems such as high-speed networks, and parallel storage systems. In addition they could come with science-oriented infrastructure services such as workflow services and batch scheduling services. PaaS science clouds could be similar to the science portals and gateways used today. They could provide tools for scientists to develop and deploy applications using domain-specific APIs and frameworks. Such systems could include access to collections of datasets used by the scientists, such as genome repositories and astronomical image archives. Finally, some commonly used science applications could be deployed using a SaaS model. These applications would allow scientists from around the world to upload their data for processing and analysis.

Additionally, HPC centers are looking at expanding their own infrastructure by relying on cloud technologies to virtualize local clusters, which would allow them to provide customized environments to a wide variety of users in order to meet their specific requirements. At the same time, HPC centers can also make use of commercial clouds to supplement their local resources when user demand is high.

Clearly, clouds can be directly beneficial to HPC centers where the staff is technically savvy. However, the adoption of clouds for domain scientists depends strongly on the availability of tools that would make it easy to leverage the cloud for scientific computations and data management.

Biographies

Gideon Juve is a PhD student in computer science at the University of Southern California. His research interests include distributed and high-performance computing, scientific workflows, and computational science.

Ewa Deelman is a research associate professor at the University of Southern California Computer Science Department and a project leader at the USC Information Sciences Institute, where she heads the Pegasus project, which designs and implements workflow mapping techniques for large-scale workflows running in distributed environments.

References

1. Amazon. Elastic compute cloud. **http://aws.amazon.com/ec2/**.
2. Barham, P., Dragovic, B., Fraser, K., Hand, S., Harris, T., Ho, A., Neugebauer, R., Pratt, I., and Warfield, A. 2003. Xen and the art of virtualization. In *Proceedings of the 19th ACM Symposium on Operating Systems Principles*. 164-177.
3. Barish, B. C. and Weiss, R. 1999. LIGO and the detection of gravitational Waves. *Physics Today 52*. 44.
4. Berriman, G. B., Deelman, E., Good, J., Jacob, J., Katz, D. S., Kesselman, C., Laity, A., Prince, T. A., Singh, G., and Su, M.-H. 2004. Montage: A grid enabled engine for delivering custom science-grade mosaics on demand. In *SPIE Conference 5487: Astronomical Telescopes*.
5. Brown, D. A., Brady, P. R., Dietz, A., Cao, J., Johnson, B., and McNabb, J. 2006. A case study on the use of workflow technologies for scientific analysis: Gravitational wave data analysis. In *Workflows for e-Science*, Taylor, I., Deelman, E., Gannon, D., and Shields, M., Eds., Springer.
6. Callaghan, S., Maechling, P., Deelman, E., Vahi, K., Mehta, G., Juve, G., Milner, K., Graves, R., Field, E., Okaya, D., Gunter, D., Beattie, K., and Jordan, T. 2008. Reducing time-to-solution using distributed high-throughput mega-workflows—Experiences from SCEC CyberShake.

In *Proceedings of the 4th IEEE International Conference on e-Science* (*e-SCIENCE'08*).

7. Clark, B., Deshane, T., Dow, E., Evanchik, S., Finlayson, M., Herne, J., and Matthews, J. N. 2004. Xen and the art of repeated research. In *Proceedings of the USENIX Annual Technical Conference, FREENIX Track*. 135–144.
8. Condor. **http://www.cs.wisc.edu/condor**.
9. Deelman, E., Singh, G., Su, M.-H., Blythe, J., Gil, Y., Kesselman, C., Mehta, G., Vahi, K., Berriman, G. B., Good, J., Laity, A., Jacob, J. C., and Katz, D. S. 2005. Pegasus: A framework for mapping complex scientific workflows onto distributed systems. *Scientific Program. J. 13*. 219-237.
10. Deelman, E., Callaghan, S., Field, E., Francoeur, H., Graves, R., Gupta, N., Gupta, V., Jordan, T. H., Kesselman, C., Maechling, P., Mehringer, J., Mehta, G., Okaya, D., Vahi, K., and Zhao, L. 2006. Managing large-scale workflow execution from resource provisioning to provenance tracking: The CyberShake example. In *Proceedings of the 2nd IEEE International Conference on e-Science and Grid Computing* (*e-SCIENCE'06*). 14.
11. EGEE Project. Enabling Grids for E-sciencE. **http://www.eu-egee.org/**.
12. Foster, I., Freeman, T., Keahey, K., Scheftner, D., Sotomayer, B., and Zhang, X. 2006. Virtual Clusters for Grid Communities. In *Proceedings of the 6th IEEE International Symposium on Cluster Computing and the Grid* (*CCGRID'06*). 513-520.
13. Foster, I., Kesselman, C., and Tuecke, S. 2001. The anatomy of the grid: Enabling scalable virtual organizations. *Int. J. High Perform. Comput. Appl. 15*. 200-222.
14. FutureGrid. **http://futuregrid.org/**.
15. Gilbert, L., Tseng, J., Newman, R., Iqbal, S., Pepper, R., Celebioglu, O., Hsieh, J., and Cobban, M. 2005. Performance implications of virtualization and hyper-threading on high energy physics applications in a grid environment. In *Proceedings of the 19th IEEE International Parallel and Distributed Processing Symposium* (*IPDPS'05*).
16. Gluster Inc. GlusterFS. **http://www.gluster.org**.
17. Google App Engine. **http://code.google.com/appengine/**.
18. Gropp, W., Lusk, E., and Skjellum, A. 1994. *Using MPI: Portable Parallel Programming with the Message Passing Interface*. MIT Press, Cambridge, MA.
19. Hyperic Inc. CloudStatus. **http://www.cloudstatus.com**.
20. InfiniBand Trade Association. InfiniBand. **http://www.infinibandta.org/**.
21. Jones, M., Ludascher, B., Pennington, D., and Rajasekar, A. 2005. Data integration and workflow solutions for ecology. In *Data Integration in Life Sciences*.
22. Keahey, K. and Freeman, T. 2008. Contextualization: Providing one-click virtual clusters. In *Proceedings of the 4th International Conference on eScience* (*e-SCIENCE'08*).
23. Ligon, W. B. and Ross, R. B. 1996. Implementation and performance of a parallel file system for high performance distributed applications. In *Proceedings of the 5th IEEE International Symposium on High Performance Distributed Computing*. 471–480.
24. Litzkow, M. J., Livny, M., and Mutka, M. W. 1988. Condor: A hunter of idle workstations. In *Proceedings of the 8th International Conference on Distributed Computing Systems*. 104-111.
25. Microsoft. Software as a service. **http://www.microsoft.com/serviceproviders/saas/default.mspx**.
26. Montage. **http://montage.ipac.caltech.edu**.
27. Myricom. Myrinet. **http://www.myri.com/myrinet/**.
28. NASA Ames Research Center. Nebula. **http://nebula.nasa.gov**.
29. NCBI. The database of genotypes and phenotypes (dbGaP). 2009. **http://www.ncbi.nlm.nih.gov/gap**.
30. NERSC. Magellan. **http://www.nersc.gov/nusers/systems/magellan**.
31. Nimbus Science Cloud. **http://workspace.globus.org/clouds/nimbus.html**.
32. Nurmi, D., Wolski, R., Grzegorczyk, C., Obertelli, G., Soman, S., Youseff, L., and Zagorodnov, D. 2008. Eucalyptus: A technical report on an elastic utility computing architecture linking your programs to useful systems. Computer Science Tech. rep. 2008-10. University of California, Santa Barbara.
33. Nurmi, D., Wolski, R., Grzegorczyk, C., Obertelli, G., Soman, S., Youseff, L., and Zagorodnov, D. 2008. The Eucalyptus open-source cloud-computing system. In *Cloud Computing and its Applications*.
34. Oinn, T., Li, P., Kell, D. B., Goble, C., Goderis, A., Greenwood, M., Hull, D., Stevens, R., Turi, D., and Zhao, J. 2006. Taverna/myGrid: Aligning a workflow system with the life sciences community. In *Workflows in e-Science*, Taylor, I., Deelman,E., Gannon, D., and Shields, M., Eds., Springer.
35. OpenPBS. **http://www.openpbs.org**.
36. Open Science Grid. **http://www.opensciencegrid.org**.
37. Panasas Inc. Panasas. **http://www.panasas.com**.
38. Paul, R. W. G., Somerville, G., Day, S. M., and Olsen, K. B. 2006. Ground motion environment of the Los Angeles region. *Structural Design Tall Special Buildings 15*. 483-494.
39. Pegasus. **http://pegasus.isi.edu**.
40. Piccoli, L. 2008. Lattice QCD workflows: A case study. In *Challenging Issues in Workflow Applications* (*SWBES'08*).
41. Schmuck, F. and Haskin, R. 2002. GPFS: A shared-disk file system for large computing clusters. In *Proceedings of the 1st USENIX Conference on File and Storage Technologies*.
42. Skrutskie, M. F., Schneider, S. E., Stiening, R., Strom, S. E., Weinberg, M. D., Beichman, C., Chester, T., Cutri, R., Lonsdale, C., and Elias, J. 1997. The Two Micron All Sky Survey (2MASS): Overview and status. In *The Impact of Large Scale Near-IR Sky Surveys*, F. Garzon, F., et al, Eds., Kluwer Academic Publishing Company, Dordrecht. 25.
43. Sotomayor, B., Montero, R., Llorente, I., and Foster, I. 2008. Capacity leasing in cloud systems using the opennebula engine. In *Cloud Computing and Applications*.
44. Stevens, R. D., Robinson, A. J., and Goble, C. A. 2003. myGrid: Personalised bioinformatics on the information grid. *Bioinformatics 19*.
45. Sun Microsystems. Lustre. **http://www.lustre.org**.
46. Taylor, I., Shields, M., Wang, I., and Philp, R. 2003. Distributed P2P computing within Triana: A galaxy visualization test case. In *Proceedings of the IEEE International Parallel and Distributed Processings Symposium* (*IPDPS'03*).
47. TeraGrid. **http://www.teragrid.org/**.
48. Torque. **http://supercluster.org/torque**.
49. Wang, L., Tao, J., Kunze, M., Rattu, D., and Castellanos, A. C. 2008. The Cumulus Project: Build a scientific cloud for a data center. In *Cloud Computing and its Applications*. Chicago.
50. Xenidis, J. 2005. rHype: IBM research hypervisor. IBM Research.
51. Youseff, L., Wolski, R., Gorda, B., and Krintz, C. 2006. Paravirtualization for HPC systems. In *Lecture Notes in Computer Science*, vol. 4331, 474.
52. Yu, W. and Vetter, J. S. 2008. Xen-based HPC: A parallel I/O perspective. In *Proceedings of the 8th IEEE International Symposium on Cluster Computing and the Grid* (*CCGrid'08*).