Volume 4, Issue 12, December 2014 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

A Comparative Analysis of Resource Management in Cloud and Grid Computing

Poonam Pandey*
CS Dept, UP Tecnical University
UP., India

Anjali Jain
IT Dept, UP Tecnical University
UP., India

Harsh Khatter
CS Dept, UP Tecnical University
UP., India

Abstract*— Cloud computing and Grid computing are two intricately connected concepts which are very popular among researchers as well as users because of its efficient resource management capability. Numbers of definitions have been given till now for Cloud and Grid. Both of these computing paradigms are milestones for the revolutionary change that has come into the field of distributed computing but when we talk about resource management then how these two paradigms differ from each other?, which one is better and why? What are the considerations which one has to think upon while using any of these two? In this paper we have tried to answer all these questions by doing a 360 degree comparative analysis in context of resource management. In the first section we have given an introduction of Grid and Cloud Computing paradigm and have described the Grids and Clouds resource management. In the next section we have explained the concepts whose understanding is must in order to understand the challenges faced by these two computing paradigms. In the next section we have presented a table which gives a comparative overview of these two computing paradigms. We have tried to bring out the challenges being faced today by these two computing paradigms. Finally we have drawn some conclusions based on our analysis and have suggested some future work in this area.

***Keywords*— TeraGrid, GridFTP, WSDL, SOAP, Lrm**

I. INTRODUCTION OF GRID AND CLOUD COMPUTING

In Cloud Computing we don't need to compute on local computers, and everything is handled by the centralized facilities which are operated by third-party that has the computation and storage utilities. It has made the tasks much easier than earlier. Ofcourse the idea of cloud computing was not totally new. In 1961 itself, John McCarthy had said that "computation may someday be organized as a public utility". The concept of Grid computing was introduced nearly in the mid of 1990s and because of this it became possible to provide on demand computing power to consumers. The idea of a Computing Grid is analogous to the electric power grid in form and utility. A number of large-scale computational grid systems like Open Science Grid, Tera Grid, Earth System Grid, EGEE etc have been developed by researchers and these are able to provide computing power, data and software, on demand. The question that comes to the mind of most of the people –Is "Cloud Computing" just an alias for Grid Computing? Which can be answered as ***Yes*** if we think of the vision of these two computation technologies? Since both of these aims to reduction of computing cost, improved flexibility and reliability because now not only a consumer can operate his/her computer but also a third party can always be there to help him out whenever he needs it. But the same question can also be answered as no because at present dealing with massive data and hence demand for computing has increased at a very large rate and therefore use of Grids and Clusters is making the task expensive. Companies like Google, Microsoft, amazon are containing more than hundreds of thousands of computers and what people need is just a credit card to on demand access to all these data centers.

a) DEFINITION of Cloud Computing

A number of discussions have been done in the literature on how Cloud can be defined [1].We can say that in Cloud Computing, *Clouds can be seen as a pool of storage, platforms and computation power which is abstract, dynamically scalable ,reliable, flexible and virtualized. And all these utilities are available on demand to the consumers.*
Some key points which are to be noted about Cloud Computing are:

- distributed computing paradigm
- massively scalable,
- an abstract entity that delivers different levels of services to customers outside the Cloud
- driven by economies of scale [2]
- the services can be dynamically configured (via virtualization or other approaches)
- Services can be delivered on demand.

In this era of Internet where enormous computing power and storage requirements are the need of hour, every organization like research institutes Governments are adopting Cloud Computing.
The main contributing factors for making everyone move towards Cloud Computing:

- rapid decrease in hardware cost
- increase in computing power and storage capacity,
- advent of multi-core architecture and modern supercomputers consisting of hundreds of thousands of cores;
- the exponentially growing data size in scientific instrumentation/simulation and Internet publishing
- the wide-spread adoption of Services Computing and Web 2.0 applications.

b) Understanding Clouds, Grids, and Clusters, Supercomputers, Web 2.0, Distributed Systems.

Figure 1 illustrates the relation between Clouds, Grids, Clusters, Supercomputers, Web 2.0 and Distributed Systems.
From the figure what can be observed is that Cloud Computing lies at the large-scale side. Supercomputing and Cluster Computing have been more focused on traditional non-service applications. Grid Computing overlaps with all these fields where it is generally considered of lesser scale than supercomputers and Clouds.

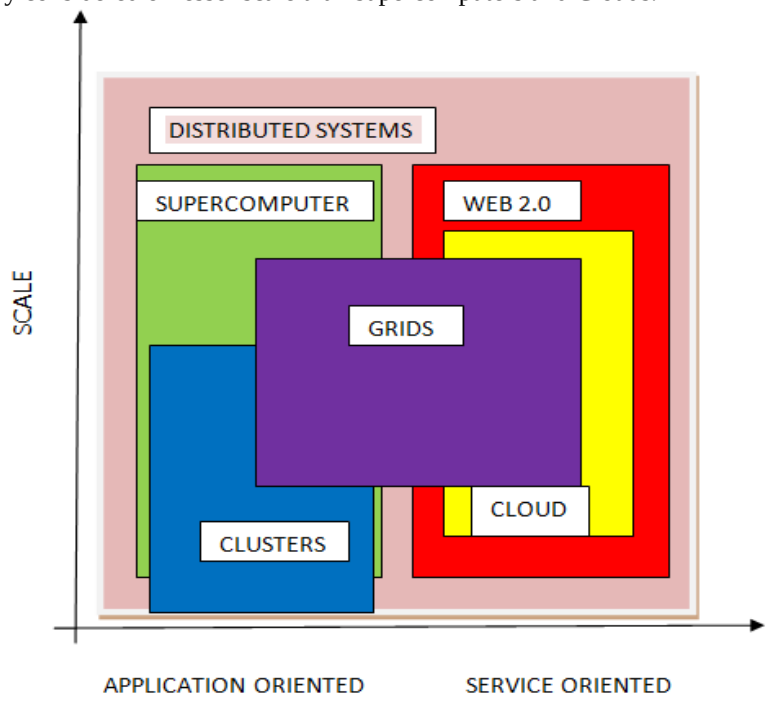

Figure 1: Grids and Clouds Overview

Aim of Grid Computing is to "enable resource sharing and coordinated problem solving in dynamic, multi-institutional virtual organizations" [18][20]. The goal of such a paradigm is to enable federated resource sharing in dynamic, distributed environments. The vision for Clouds and Grids are similar, details and technologies used may differ, but the two communities are struggling with many of the same issues. In this paper we are comparing and contrasting Cloud Computing with Grid Computing from different angles and have tried to cover all the essential characteristics of both, with the hope to impart a clear knowledge of what Clouds are, various kinds of applications which they are expected to support, and challenges which Clouds may have to face in the future in order to gain momentum and adoption. We hope this will help Cloud as well as Grid community to gain deeper and clear understanding of the goals, assumptions, status, and directions, and will provide a more elaborated view of both technologies.

II. SIDE-BY-SIDE COMPARISONS OF CLOUD AND GRID

In this section we have compared Grids and Clouds from all the perspectives like- architecture model, security model, business model, virtualization, programming model, data model, compute model, to provenance and applications. Number of opportunities and challenges that Grid Computing and Cloud Computing have brought in front of researchers and the IT industry are also outlined.

a) Business Model

In one line we can say that business model states: *Pay one time, Use Unlimited*. In business model, the provider is paid by the customer on a consumption basis, very similar to basic utilities charges such as gas, electricity, and water. The model relies on economies of scale as it drives prices down for users and profits up for providers.
Example: Amazon provides a centralized Cloud consisting of EC2 and S3 which are Compute Cloud and Data Cloud respectively. The former is charged based on per instance-hour consumed for each instance type and the later is charged by per GB-Month of storage used. In addition, data transfer is charged by TB /month data transfer, depending on the source and target of such transfer.

The business model for Grids is basically project-oriented in which the users or community represented by that proposal have certain number of service units (i.e. CPU hours) they can spend. For example, the TeraGrid operates in this fashion, and requires increasingly complex proposals be written for increasing number of computational power. The TeraGrid has more than a dozen Grid sites, all hosted at various institutions around the country. What happens is that when an institution joins the TeraGrid with some resources, it knows well that others in the community can now use these resources. It also acknowledges the fact that it has also gained now access to a dozen other Grid sites. Various other economic models have been proposed and applied in practice [5].

Architecture

Grids came to the scene in the mid-90s for addressing large-scale computation problems using a network of resource-sharing computers and thus it was able to deliver the computation power affordable only by supercomputers and large dedicated clusters at that time. The main motive was that these high performance computing resources were expensive and it was not easy to get access to those resources, so the initial point was to use heterogeneous and dynamic resources from number of geographically distributed institutions.

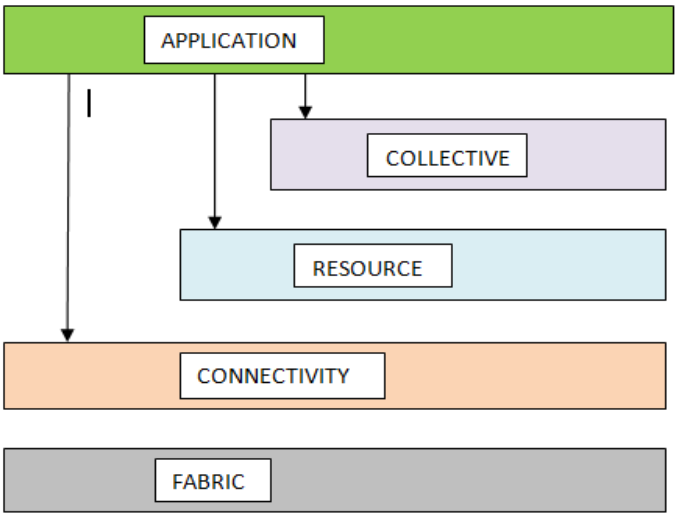

Figure 2: Grid Protocol Architecture

There are five different layers in Grid Architecture which provide protocols and services (see Figure 2).

1) Fabric layer: Provides access to different resource types such as compute, storage and network resource, code repository, etc. Grids have fabric components like local resource managers (i.e. PBS[5], Condor [48], etc). General-purpose components such as GARA (general architecture for advanced reservation) [9], and specialized resource management services such as Falkon[24].

2) Connectivity layer defines authentication and core communication protocols for easy and secure network transactions.

3) Resource layer defines protocols for the publication, discovery, negotiation, monitoring, accounting and payment of sharing operations on individual resources. The GridFTP [2] for data access and high-speed data transfer and the GRAM (Grid Resource Access and Management) [8] protocol is used for allocation of computational resources and for monitoring and control of computation on those resources.

4) Collective layer captures interactions across collections of resources, directory services such as MDS (Monitoring and Discovery Service) [26] allows for the monitoring and discovery of VO resources, Condor-G [13] and Nimrod-G [4] are examples of co-allocating, scheduling and brokering services, and MPICH [18] for Grid enabled programming systems, and CAS (community authorization service) [11] for global resource policies.

5) Application layer comprises whatever user applications built on top of the above protocols and APIs and operate in VO environments. Clouds are developed to address Internet-scale computing problems where some assumptions are different from those of the Grids. Clouds are usually referred to as a large pool of computing and/or storage resources, which can be accessed via standard protocols via an abstract interface. Clouds can be built on top of many existing protocols such as Web Services (WSDL, SOAP), and some advanced Web 2.0 technologies such as REST, RSS, AJAX, etc. We define a four-layer architecture for Cloud Computing in comparison to the Grid architecture, composed
Of

1) Fabric,
2) Unified resource,
3) Platform, and
4) Application Layers.

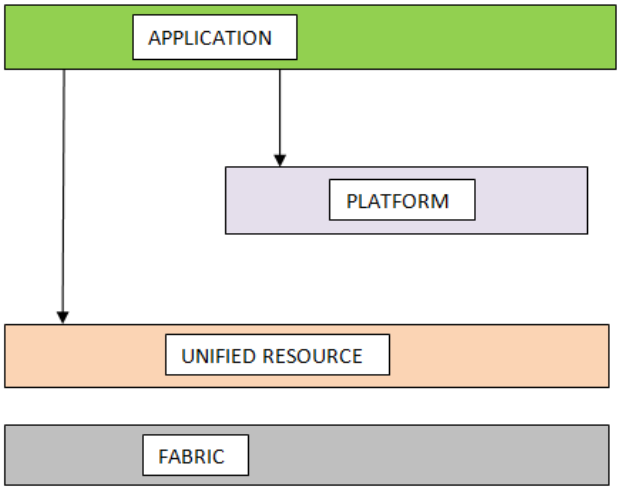

Figure 3: Cloud Architecture

1) The ***fabric layer*** contains the raw hardware level resources, such as computer resources, storage resources, and network resources.

2) The ***unified resource layer*** contains resources that have been abstracted/encapsulated (usually by virtualization) so that they can be exposed to upper layer and end users as integrated resources, for instance, a virtual computer/cluster, a logical file system, a database system, etc.

3) The ***platform layer*** adds on a collection of specialized tools, middleware and services on top of the unified resources to provide a development and/or deployment platform. For instance, a Web hosting environment, a scheduling service, etc.

4) Application layer contains the applications that would run in the Clouds.

Clouds in general provide services at three different levels (*IaaS*, *PaaS*, and *Saas* [29]) as follows, although some providers can choose to expose services at more than one level.
Infrastructure as a Service (IaaS) [29] provisions hardware, software, and equipments (mostly at the unified resource layer, but can also include part of the fabric layer) to deliver software application environments with a resource usage-based pricing model. Infrastructure can scale up and down dynamically based on application resource needs. Typical examples are Amazon EC2 (Elastic Cloud Computing) Service [3] and S3
(Simple Storage Service) [3] ,Eucalyptus [7] is an open source Cloud implementation that provides a compatible interface to Amazon's EC2, and allows people to set up a Cloud infrastructure at premise and experiment.
Platform as a Service (PaaS) [29] offers a high-level integrated environment to build, test, and deploy custom applications. An example is Google's App Engine [16], which enables users to build Web applications on the same scalable systems that power Google applications.
Software as a Service (SaaS) [29] delivers special-purpose software that is remotely accessible by consumers through the Internet with a usage-based pricing model. Live Mesh from Microsoft allows files and folders to be shared and synchronized across multiple devices.

Compute Model: Most Grids use a batch-scheduled compute model, in which a local resource manager (LRM), such as PBS, Condor, SGE manages the computer resources for a Grid site, and users submit batch jobs (via GRAM) to request some resources for some time. Many Grids have policies in place that enforce these batch jobs to identify the user and credentials under which the job will run for accounting and security purposes, the number of processors needed, and the duration of the allocation. Due to the expensive scheduling decisions, data staging in and out, and potentially long queue times, many Grids don't natively support interactive applications; although there are efforts in the Grid community to enable lower latencies to resources via multi-level scheduling, to allow applications with many short-running tasks to execute efficiently on Grids [24].
Cloud Computing compute model will likely look very different, with resources in the Cloud being shared by all users at the same time (in contrast to dedicated resources governed by a queuing system). This allows latency sensitive applications to operate natively on Clouds.

Data Model: The critical role of Cloud Computing goes without saying, but the importance of Client Computing cannot be overlooked either for several reasons:
1) For security reasons, people might not be willing to run mission-critical applications on the Cloud and send sensitive data to the Cloud for processing and storage;
2) Users want to get their things done even when the Internet and Cloud are down or the network communication is slow;

3) With the advances of multi-core technology, the coming decade will bring the possibilities of having a desktop supercomputer with 100s to 1000s of hardware threads/cores.
Furthermore, many end-users will have various hardware driven end-functionalities, such as visualization and multimedia playback, which will typically run locally.
Data Grids [10] have been specifically designed to tackle data intensive applications in Grid environments, with the concept of virtual data [12] playing a crucial role.

Data Locality: One approach is to improve schedulers to be data-aware, and to be able to leverage data locality information when scheduling computational tasks; this approach has shown to improve job turn-around time significantly [25].

Virtualization: Virtualization provides the necessary abstraction such that the underlying fabric (raw compute, storage, network resources) can be unified as a pool of resources and resource overlays (e.g. data storage services, Web hosting environments) can be built on top of them. Virtualization also enables each application to be encapsulated such that they can be configured, deployed, started, migrated, suspended, resumed, stopped, etc., and thus provides better security, manageability, and isolation.
There are also many other reasons that Clouds tend to adopt virtualization:
1) Server and application consolidation, as multiple applications can be run on the same server; resources can be utilized more efficiently;
2) Configurability, as the resource requirements for various applications could differ significantly, some require large storage, some compute, in order to dynamically configure and bundle (aggregate) resources for various needs, virtualization is necessary as this is not achievable at the hardware level;
3) Increased application availability, virtualization allows quick recovery from unplanned outages, as virtual environments can be backed up and migrated with no interruption in service;
4) Improved responsiveness: resource provisioning, monitoring and maintenance can be automated, and common resources can be cached and reused.
Grids do not rely on virtualization as much as Clouds do, but that might be more due to policy and having each individual organization maintain full control of their resources (i.e. by not virtualizing them). However, there are efforts in Grids to use virtualization as well, such as Nimbus [32] (previously known as the Virtual Workspace Service [15]), which provide the same abstraction and dynamic deployment capabilities. A virtual workspace is an execution environment that can be deployed dynamically and securely in the Grid. \

Monitoring: Cloud monitoring is not as straightforward as in Grids, because Grids in general have a different trust model in which users via their identity delegation can access and browse resources at different Grid sites, and Grid resources are not highly abstracted and virtualized as in Clouds; for example, the Ganglia [14] distributed (and hierarchical) monitoring system can monitor a federation of clusters and Grids and has seen wide adoption in the Grid community. Essentially monitoring in Clouds requires a fine balance of business application monitoring, enterprise server management, virtual machine monitoring, and hardware maintenance, and will be a significant challenge for Cloud Computing as it sees wider adoption and deployments.
On the other hand, monitoring can be argued to be less important in Clouds, as users are interacting with a more abstract layer that is potentially more sophisticated; this abstract layer could respond to failures and quality of service (QoS) requirements automatically in a general-purpose way irrespective of application logic. In the near future, user-end monitoring might be a significant challenge for Clouds, but it will become less important as Clouds become more sophisticated and more self-maintained and self-healing.

Provenance: Provenance refers to the derivation history of a data product, including all the data sources, intermediate data products, and the procedures that were applied to produce the data product. Provenance information is vital in understanding, discovering, validating, and sharing a certain data product as well as the applications and programs used to derive it. In some disciplines such as finance and medicine, it is also mandatory to provide what is called an "audit trail" for audition purpose. In Grids, provenance management has been in general built into a workflow system, from early pioneers such as Chimera [12], to modern scientific workflow systems, such as Swift [30], Kepler [20], and VIEW [19] to support the discovery and reproducibility of scientific results. It has also been built as a standalone service, such as PreServ [17], to facilitate the integration of provenance component in more general computing models, and deal with trust issues in provenance assertion. Using provenance information, scientists can debug workflow execution, validate or invalidate scientific results, and guide future workflow design and data exploration. While provenance has first shown its promise in scientific workflow systems [12] and database systems [28], a long-term vision is that provenance will be useful in other systems as well, necessitating the development of a standard, open, and universal representation and query model. Currently, the provenance challenge series [23] and the open provenance model initiative [22] provide the active forums for these standardization effort and interaction. On the other hand, Clouds are becoming the future playground for e-science research, and provenance management is extremely important in order to track the processes and support the reproducibility of scientific results [27]. Provenance is still an unexplored area in Cloud environments, in which we need to deal with even more challenging issues such as tracking data production across different service providers and across different software and hardware abstraction layers. In other words, capturing and managing provenance in Cloud

environments may prove to be more difficult than in Grids, since in the latter there are already a few provenance systems and initiatives, however scalable provenance querying [31] and secure access of provenance information are still open problems for both Grids and Clouds environments.

b) Different models of Cloud Computing and Grid Computing

In the following table we have compared the different models of Cloud and Grid computing.

A.	*B.* ***Cloud Computing***	*C.* ***Grid Computing***
D. ***Architecture Model***	*E. Cloud Computing lies at the large-scale side.*	*F. Grid Computing overlaps with many other fields where it is generally considered of lesser scale than supercomputers and Clouds.*
G. ***Resource Management***	*H. In Cloud resources are being shared by all users at the same time (opposite to the dedicated resources used by a queuing system). Hence it allows only latency sensitive* *I. applications to operate on Clouds. It ensures good enough level of Quality Of Service to the end users.*	*J. Grids focus on integrating existing resources with their hardware, operating systems, local resource management, and security infrastructure.* *K.*
L. ***Security Model***	*M. Currently, the security model for Clouds seems to be relatively* *N. simpler and less secure than the security model adopted by* *O. Grids.*	*P. The Grid approach to security might be more time consuming, but it adds an extra level of security to help* *Q. Prevent unauthorized access.* *R.*
S. ***Application Model***	*T. The applications that will run on Clouds are not well defined,* *U. but can certainly be characterized as loosely coupled, transaction oriented* *V. and likely to be interactive*	*W. Opposite to clouds, Grids support batch scheduled applications which are not interactive and support many other well defined applications.* *X.*
Y. ***Programming Model***	*Z. Mesh-up's and scripting (JavaScript, PHP, Python etc) have been taking the place of a workflow system in the Cloud world because there is no easy way to integrate services and* *AA. applications from various providers. They are essentially data integration approaches, because they take outputs from a service/application, transform them and feed into another. Example: Google App Engine uses a modified Python runtime and chooses Python scripting language for Web application development.*	*BB. Grids primarily target large-scale scientific computations, so it must scale to leverage large number/amount of resources, and we would also naturally want to make programs run fast and efficient in Grid environments, and programs also need to finish correctly, so reliability and fault tolerance must be considered. We briefly discuss here some general programming models in Grids. MPI (Message Passing Interface) [21], MapReduce [6]*

III. CONCLUSIONS AND LIGHTS TO THE FUTURE

In this paper, we have show n that a lot commonality is shared between Clouds and Grids in their vision, architecture and technology, but still there are many differences between them with respect to the points such as security, programming model, business model, compute model, data model, applications, and abstractions. We have also identified the

challenges and opportunities in both of these fields. We hope that such a close comparison can help the Cloud and Grid communities understand, share and evolve infrastructure and technology within and across the globe.

Cloud and Grid both will move towards a mix of micro production and large utilities, with enhanced numbers of small-scale producers. We need to support on-demand provisioning and configuration of integrated "virtual systems" providing the precise capabilities needed by an end-user. We also need to define protocols that allow users and service providers to discover and hand off demands to other providers, to monitor and manage their reservations, and arrange payment. Tools are also needed for managing both the underlying resources and the resulting distributed computations. The methods used to achieve these goals in today's commercial clouds are not open and general purpose, but mostly of them are proprietary and specialized for the specific uses. Interoperability between providers has not yet surfaced. Some of the required protocols and tools will come from the smart people from the industry .Some ideas will come from the smart people from academia and government labs. It will be interesting to see to what extent Clouds and Grids can proceed along parallel paths or can manage to find common cause.

REFERENCES

[1] "Twenty Experts Define Cloud Computing", SYS-CON Media Inc, http://cloudcomputing.sys con.com/read/612375_p.htm, 2008.

[2] J. Silvestre. "Economies and Diseconomies of Scale," The New Palgrave: A Dictionary of Economics, v. 2, pp. 80–84, 1987.

[3] Amazon Simple Storage Service (Amazon S3), http://aws.amazon.com/s3, 2008.

[4] R. Buyya, D. Abramson, J. Giddy. "Nimrod/G: An Architecture for a Resource Management and Scheduling System in a Global Computational Grid", IEEE Int. Conf. on High Performance Computing in Asia-Pacific Region (HPC ASIA) 2000.

[5] R. Buyya, K. Bubendorfer. "Market Oriented Grid and Utility Computing", Wiley Press, New York, USA, 2008.

[6] J. Dean, S. Ghemawat. "MapReduce: Simplified Data Processing on Large Clusters", Sixth Symposium on Operating System Design and Implementation (OSDI04), 2004.

[7] Eucalyptus, http://eucalyptus.cs.ucsb.edu/, 2008.

[8] I. Foster, C. Kesselman. "Globus: A Metacomputing Infrastructure Toolkit", Intl J. Supercomputer Applications, 11(2):115-128, 1997.

[9] I. Foster, C. Kesselman, C. Lee, R. Lindell, K. Nahrstedt, A. Roy. "A Distributed Resource Management Architecture that Supports Advance Reservations and Co-Allocation", Intl Workshop on Quality of Service, 1999.

[10] A. Chervenak, I. Foster, C. Kesselman, C. Salisbury, S. Tuecke. "The Data Grid: Towards an Architecture for the Distributed Management and Analysis of Large Scientific Datasets", Jrnl. of Network and Computer Applications, 23:187-200, 2001.

[11] I. Foster, C. Kesselman, L. Pearlman, S. Tuecke, and V. Welch. "The Community Authorization Service: Status and Future," In Proc. of Computing in High Energy Physics (CHEP), 2003.

[12] I. Foster, J. Vöckler, M. Wilde, Y. Zhao, "Chimera: A Virtual Data System for Representing, Querying, and Automating Data Derivation", SSDBM 2002: 37-46.

[13] J. Frey, T. Tannenbaum, I. Foster, M. Livny, and S. Tuecke. "Condor-G: A Computation Management Agent for Multi- Institutional Grids," Cluster Computing, 5 (3). 237-246. 2002.

[14] Ganglia, http://sourceforge.net/projects/ganglia/, 2008.

[15] K. Keahey, I. Foster, T. Freeman, and X. Zhang. "Virtual Workspaces: Achieving Quality of Service and Quality of Life in the Grid", The Scientific Programming Journal, 2006.

[16] Google App Engine, http://code.google.com/appengine/, 2008.

[17] P. Groth, S. Miles, W. Fang, S. Wong, K. Zauner, L. Moreau. "Recording and using provenance in a protein compressibility experiment", in Proceedings of the 14th IEEE Int. Symposium on High Performance Distributed Computing (HPDC), 2005.

[18] N. Karonis, B. Toonen, and I. Foster. MPICH-G2: A Grid- Enabled Implementation of the Message Passing Interface. Journal of Parallel and Distributed Computing, 2003.

[19] C. Lin, S. Lu, Z. Lai, A. Chebotko, X. Fei, J. Hua, and F. Fotouhi, "Service-Oriented Architecture for VIEW: a Visual Scientific Workflow Management System", IEEE International Conference on Services Computing (SCC), pp.335-342, 2008.

[20] B. Ludäscher, I. Altintas, C. Berkley, D. Higgins, E. Jaeger, M. Jones, E. Lee, J. Tao, Y. Zhao, "Scientific workflow management and the Kepler system. Concurrency and Computation: Practice and Experience", 18(10).

[21] M.P.I. Forum. "MPI: A Message-Passing Interface Standard", 1994.

[22] Open provenance model initiative, http://twiki.ipaw.info/bin/view/Challenge/OpenProvenanceModelReview, 2008.

[23] The provenance challenge series: http://twiki.ipaw.info/bin/view/Challenge/, 2008.

[24] I. Raicu, Y. Zhao, C. Dumitrescu, I. Foster, M. Wilde. "Falkon: a Fast and Light-weight tasK executiON framework", IEEE/ACM SuperComputing 2007.

[25] I. Raicu, Y. Zhao, I. Foster, A. Szalay. "Accelerating Largescale Data Exploration through Data Diffusion", International Workshop on Data-Aware Distributed Computing 2008.

[26] J. M. Schopf, I. Raicu, L. Pearlman, N. Miller, C. Kesselman, I. Foster, M. D'Arcy. "Monitoring and Discovery in a Web Services Framework: Functionality and Performance of Globus Toolkit MDS4", Technical Report, Argonne National Laboratory, MCS Preprint #ANL/MCS-P1315-0106, 2006.

[27] Y. Simmhan, Beth Plale, Dennis Gannon. "A survey of data provenance in e-science", SIGMOD Record 34(3): 31-36, 2005.

[28] W. C. Tan. "Provenance in databases: Past, current, and future", IEEE Data Eng. Bull., 30(4):3–12, 2007.

[29] "What is Cloud Computing?", Whatis.com. http://searchsoa.techtarget.com/sDefinition/0,,sid26_gci1287881,00.html, 2008.

[30] Y. Zhao, M. Hategan, B. Clifford, I. Foster, G. von von- Laszewski, I. Raicu, T. Stef-Praun, and M. Wilde. "Swift: Fast, reliable, loosely coupled parallel computation", IEEE Int. Workshop on Scientific Workflows, pages 199–206, 2007.

[31] Y. Zhao and S. Lu, "A Logic Programming Approach to Scientific Workflow Provenance Querying", IPAW, LNCS Volume 5272, pp.31-44, 2008.

[32] K. Keahey, T. Freeman. "Contextualization: Providing One- Click Virtual Clusters", to appear, eScience 2008, 2008.

