J. Basic. Appl. Sci. Res., 5(5)33-43, 2015
© 2015, TextRoad Publication

ISSN 2090-4304
Journal of Basic and Applied Scientific Research
www.textroad.com

Grid Computing in Light of Resource Management Systems: A Review

Saddaf Rubab[1], Mohd. Fadzil Hassan[2], Ahmad Kamil Mahmood[3], Nasir Mehmood Shah[4]

[1,2,3,4] Department of Computer and Information Sciences, University Teknologi PETRONAS, Bandar Seri Iskandar, 31750, Tronoh, Perak, Malaysia

Received: February 28, 2015
Accepted: April 30, 2015

ABSTRACT

The grid computing gives a new view of distributed computing environment. It is a network of collaborating resources to complete a task or process. Resource management is considered as one of the primary module of a grid system. The paper provides a detailed study of existing resource management systems to explore future research directions. It also explains the types of grids and common terms associated with resource management systems to analyze their performance. The study established on grid resource management systems is on basis of systematic review principles.

KEYWORDS: Grid Computing, GRM, protocol, Scheduling, resource management, RMS, Interoperability

1. INTRODUCTION

Computer systems are going through a revolution from mainframe computers to multi core designed computers. The spread of high speed network connections and availability of high computing power convinced people to change to the way of information handling and computing services. The concept of distributed computing emerged by sharing resources within a network and make execution of user tasks efficient. Distributed computing provides a network of independent computers appear to provide user as a single unified system. In 1990s, many of research projects were in progress focusing on distributed computing. The main theme was to develop a distributed system that appears as one computer node to users. I-Way demonstration was connecting 11 networks, conducted at IEEE/ACM 1995 supercomputing conference by Ian Foster [1]. For this, many programs were developed to make communication possible between different resources and applications running on each network/computer node. These experiments were then later on named as Grid Computing. Remarkable computational power is available to users at any time at many computer nodes installed in a grid system just like the power grids make electric power accessible from any number of electric outlets [2].Grid computing, is arrangement of distributed computing which integrates computer networks by linking distributed heterogeneous resources and computer nodes.

Grid computing is a virtual organization of distributed computing in an environment to share geographically dispersed resources as they assume absence of central location and administration. Virtual Organization (VO) can be comprised of small networks in same physical location or large group of resources installed in different networks distributed around the world[3]. All the work done in high performance, cluster, peer-to-peer, and internet computing areas can be covered collectively under one umbrella of Grid computing. The main theme is, each environment works for information and resources sharing and management but with the difference of resource type or location. In grids, the resources are present at different locations and belong to different owners. This makes the resources heterogeneous in nature which demands more management when demanded for solving a different segment of task and the solution will be comprised of results from dispersed resources.

The paper objective is to present the literature review on grid architectures in light of resource management and also to discuss several options for making any grid architecture secure to motivate the future works of developing any grid resource management system secure.

Section 2 describes the processes involved in collecting the related published manuscripts, which are restrictive and unrestrictive search criteria methods. Section 3 elaborates the comparison criteria for grid architectures followed by Section 4, which gives a detailed comparison. Finally, Section 5 settles the paper by concluding the outcomes of the performed literature review and charting the future works.

2. MATERIALS AND METHODS

The published literatures reviewed in this paper were collected through two different ways, namely restrictive and unrestrictive research criteria. The subsections that follow contained both types of reviews.

***Corresponding Author:** Saddaf Rubab, Department of Computer and Information Sciences, University Teknologi PETRONAS, Bandar Seri Iskandar, 31750, Tronoh, Perak, Malaysia. saddaf_g02754@utp.edu.my

In unrestrictive search, the information was collected from journal articles, conference proceedings and websites as the grid infrastructures publish their reports online on their corresponding websites. This covers only about the grid architectures and their salient features to compare the characteristics, presented in Section III and IV. The keywords “grid system”, “grid infrastructure” and “grid architecture” were used when the search was conducted using websites and online reports. The same keywords with some additional module names of a grid yielded search results from journal articles and conference proceedings. In authors’ case, the keywords “grid resource management system”, “grid infrastructure and resource management” and “grid architecture and resource management” were used. The information collected from the searches was then refined to ensure only the relevant ones were selected for review of grid systems in light of resource management features.

The search criteria to select published articles for review were then tough in order to gain related references related to resource management of grid systems.

3. Resource Management Systems Comparison Metrics

3.1. Classification/Type of Grid

There are many approaches to classify grids like some do it on the basis of structure variation and some on the resources used in grid[4, 5].

Today we observe service and desktop grids [4] as two different inclinations in the development of grid systems. A grid service is firstly created and allows access by a large number of users in service grids. Middleware or a set of software is required to be installed when a resource wants to become a part of grid system like EGEE [6] and Nordu Grid [7]. In service grids, to maintain a middleware requires services of experts because it is very complex and therefore services grids are restricted to organizations and institutions where professional experts are dedicated for this purpose.

In desktop grids, the architecture is simple as compared to service grids. It integrates the heterogeneous distributed resources like CPU cycles offered by public users and so are also referred as public or volunteer grid computing systems. Desktop grids invite donors to share or contribute their spare resources. SETI@HOME project[8] is an example of desktop grids, which involved four million personal computers approximately.

The method of computations in these two is different. Service grids follows job push mechanism in which a job, task or service is being pushed on a resource to execute. Pull model is being practiced in desktop grids in which a resource attracts or ask fetches task or job requested by users. In the section IV we will use the mentioned discrimination of computation methods to categorize existing resource management systems.

The grid resources are more powerful than commonly used network resources. The storage system, database, any scientific instrument, or execution cycles can be considered as resource in a grid[9]. The tangible hardware components used to build up the usual computing system are considered as physical resources in computing grid systems as well. These are also helpful in finding the logical resources on a grid. These include storage resources (e.g., RAM, ROM, hard disks, disk drives), computational resources (e.g., CPU cycles, clock time), network resources (e.g., LANs and its devices installed) and peripheral resources (e.g., input and output devices). Logical resources support the operation of grid physical resources. Logical resources include data, knowledge, application resources and services to access other resources. In the comparison of existing resource management systems we will take note of computational resources only.

A Resource Management System (RMS) is responsible for managing the resources available in any grid system. The purpose of grid actually defines which services an RMS will make available. But the elementary function of any RMS is to entertain the resource requests and allocate resources available in an installed grid system. There are many RMS proposed each with some different service availability.

To compare some of the existing RMS systems, we need to understand some basic properties on the basis of which we shall differentiate between them. Following properties will help to make the comparison matrices for analysis and review of the characteristics of existing RMSs.

3.2. Machine organizations

In a Grid how the machines are organized defines communication pattern of RMS which further defines its adoptability to different other platforms or environment. The machine organization also helps in determining the level of machine involvement because it provides ample information for designing the scheduling and communication schemes between the resources and resource requestors. We deal with three machine organizations in computational grid systems i.e., flat, cell and hierarchical organizations.

The flat machine organization enables all machines to communicate directly without the involvement of any intermediate machine or pathway as all machines lay on same level. In hierarchical machine organization, the intermediary machine is required to communicate between for the reason of level wise structure of machines placement. But the machines on the same level can communicate directly and also the machines one level up or down that level. It also provides easy installations if more machines or levels are to be added with time. This machine organization is being explored by many current grid systems. The cell machine organization make use of flat machine organization for organizing machines within a cell and if a cell contain many other cells then it will use hierarchical machine organization within a cell. In each cell there is one dedicated machine responsible for making communication feasible between other cells and it is the only machine visible outside its respective cell.

Hierarchical and cell machine organization have many similarities but they differ on the basis of boundary and visibility of internal structure. In cell organization, the boundary is well defined but the structure of cell has no visibility. In hierarchical machine organization, the structure of any hierarchy is visible outside its boundary.

3.3. Scheduling organizations

It describes the way to allocate resources. It could be central, decentralized, or hierarchical[10, 11]. In centralized scheduling, the authority rests at center. Data is hosted at a single point which make centralized scheduling simple and easy to implement but at the same time the single data point is the only point of failure and vulnerable. Multiple schedulers are used since there is no dedicated resource scheduler in decentralized scheduling, the scheduling and allocation of resources is regulated by resource provider and requestor. In hierarchical organization, the main scheduler interacts with local schedulers [12, 13]. In hierarchical scheduling, the resource providers are organized in hierarchies defining the level of units to be organized by each resource provider. The resource providers in higher hierarchy levels deal with scheduling of big units and as the hierarchy level lowers down the units' size decreases. On each level there is one job queue and one resource queue as well.

3.4. Resource Model

It provides information about how a RMS will describe and manage the resources in a grid. In the object based model, resource model provides a mechanism but in schema based model, the resource data is described in a description language [14].

Going into more detail of schema based model explains that description language is used with integrity constraints and sometimes assimilated with a query language to describe the resources in an RMS. The schema elements which cannot be extended are recognized as fixed schema model and the elements providing ability to be extended fall under the extensible schema model. By the extensions, a new schema can be added to describe resources.

The resource operations are defined as part of RMS and are well-defined at the time of RMS creation in object based models. There is a possibility to extend the object models, although it is difficult to implement the extended object models.

3.5. Quality of Service (QoS)

It defines the level of service users' request from RMS or a resource and RMS is required to provide. QoS may deal with the attributes of time, efficiency, reliability, security, throughput etc. To estimate the QoS of any RMS considering multiple attributes is a difficult task and is yet posed a critical problem in distributed computing for scientists.

If a RMS does not specify the QoS for resource requests then such a RMS does not support QoS. There are RMSs which provide QoS at time of job submission but not for resource requests, they provide partial support for QoS [15]. If any RMS supports QoS, a negotiation on Service Level Agreement (SLA) with RMS is possible but not necessary.

3.6. Resource Discovery

A mechanism for resource discovery, allocation and utilization is provided by a RMS in a computational grid. The efficiency of any RMS is calculated on the basis of overhead occurred in matching resources with the requests arrived. It also helps in controlling the resource utilization rate. The queries are sent across the grid network to find the best suited resource in resource repository via some intermediary query processor or engine. The query if get into contact with a distributed resource repository, the resource discovery is said to have distributed query system else centralized query system. Agent based approaches are also explored for resource discovery purposes [16-23]. In agent based approach, a code fragment is sent to machines in grid and each machine will interpret it locally. Agents mimic the query based approach by monitoring and distributing resource information either in a periodic manner or in response to any action by other agent. But in agent based approach, agents possess decision control and makes resource discovery inherently distributed [21, 23].

3.7. Resource Distribution Methods

There are two methods i.e., push and pull [24]. In push distribution, jobs are assigned to resources by system. The other one, pull resource distribution, resources are made aware of waiting jobs and resources initiate the allocation process.

3.8. Breadth of scope

It defines the degree to which a grid system can be expanded by adding, removing or replacing the resources in grid system[25]. It also explains the adaptability and flexibility of services provided by a grid system. If the services of grid are defined for a specific platform, the breadth of scope is low and the grid system performance is poor. The systems which allow scaling the grid system and have adaptable services can be regarded as having medium breadth of scope or high breadth of scope. We can consider a scheduler[26], which only accepts independent tasks and if there is any task dependency then the scheduler will not accept those tasks for scheduling.

4. Survey of Existing Grid Resource Management Systems

The RMSs discussed here are selected considering all the available and famous in grid research. Grid RMSs are compared and discussed briefly according to the classification that we have developed in section 3. A summarized version of comparison is presented in TABLE 1 (a and b). The following RMSs are organized in ascending order of their evolution or advancement.

1. AppLeS (Application Level Scheduling)

AppLeS, project at University of California, San Diego, an application level grid scheduler assisting the scheduling activity by operating like an agent in active environment. It is a computational grid. AppLeS select resources for each application and so is designed separately for each application[27]. AppLeS templates can be used for the applications with similar structure. It can also make interaction with other resource management systems like, Legion, NetSolve and Globus. AppLeS uses hierarchical or decentralized scheduling and do not provide support for QoS. It has low breadth of scope.

2. Globus

It is a toolkit for grid computing and makes available a view of all available resources as a single virtual machine[28]. Globus has a hierarchical organization with higher level services residing on top can be constructed using the low level services at core [29]. All the available services are assembled with the resource information and push protocol is used to allocate the available resources. It offers QoS in terms of reserving resources.

3. NetSolve (A Network Enabled Computational Kernel)

It is an agent application server integrating resources lying on network to desktop application[30]. It has dedicated agent which works as resource broker and is responsible for resource scheduling and distribution. It is used for computational purposes along with decentralized scheduler. It uses push resource distribution and has hierarchical-cells machine organization. It provides QoS with high breadth of scope.

4. Darwin [31]

Darwin is a grid resource management system [31]which is customizable to provide value added network services electronically. It supports push based resource management for networked resources only but does not provide any for non-networked resources. Darwin operates in routers and provides QoS for network bandwidth. It uses hierarchical scheduling and utilizes Xena[31] for globally scheduling resources. It uses Hierarchical Fair Service Curve Scheduling (H-FSC) for higher level resource scheduling.

5. Condor (Cycle Stealing Technology for High Throughput Computing)

It was technologically advanced at University of Wisconsin at Madison, USA, aiming for high throughput computing environment. Condor allocates resources according to the resource usage policies of resource owner through unique remote system calls. Condor can have sub-condor pools organized in flats. Condor collector is held responsible for advertising and listening to the resource requests defined in Class Ad Language[32]. When a match is made between resource requested and available only then the resource requestor machine is notified about the compatibility.

6. Ninf (Network Infrastructure)

Ninf has client-server architecture[33], consisting of client interfaces, remote libraries and meta-server as primary components. Applications running on Ninfcall libraries to forward the resource requests to meta-server[33]. It doesn't offer QoS and breadth of scope is medium. Centralized resource discovery is performed in Ninf.

7. NWS (Network Weather Service)

It forecasts network and its resources in a distributed system fashion. It utilizes previous system states for forecasting mechanism [34]primarily considering the network bandwidth and CPU utilization. It runs extensible object model for gathering resource information. NWS serves fine for short-time processes, so its breadth of scope is low.

8. PUNCH (Purdue University Network Computing Hubs)

Itprovides operating services in a distributed computing environment as it works on principle of multi-user and multi-process environment based middleware[35, 36]. The basic purpose for its development was to provide an environment independent platform for applications. PUNCH lies on hierarchical machine organization with decentralized scheduling scheme. The breadth of scope in PUNCH is at middle level. It uses push resource distribution and QoS.

9. Legion

It is an operating system for grid, deployed at University of Virginia[37], fulfilling purpose to connect hosts varying from personal computers to parallel computers. Legion has objects being managed by their respective classes or meta-classes. These objects can create, destroy, schedule and activate the new instances. Legion follows hierarchical machine organization with the ad-hoc user oriented scheduling policy. The breadth of scope is high.

10. Lancaster DMRG (Distributed Multimedia Research Group)

The University of Lancaster Distributed Multimedia Research Group (DMRG) aimed to provide high performance distributed multimedia applications and conducted many projects including the Generic Object Platform Infrastructure (GOPI)[38, 39]. GOPI is based on CORBA and so it laid a flat machine organization with an extensible object model for gathering resource information.

11. MSHN (Management System for Heterogeneous Networks)

MSHN is collaboration between United States Department of Defense, some universities including Universities of Purdue and NOEMIX industry[40]. This project aimed at providing resource management system for distributed computing environment with a target of adaptable applications corresponding to the grid resources' condition[41, 42]. The main components of MSHN are client library, resource server, resource database and scheduling advisor. In MSHN, there is no clear specification of QoS, although in scheduler design it does support QoS. In MSHN, resource reservation is practicable.

12. Ninja

It is an extension of Java based infrastructure for supporting fault-tolerance, scalability, and internet based applications availability [43, 44]. Components of Ninja are described using XML and ISL (Interface Specification Language). It has hierarchical machine organization with on demand service grid feature. Its' resource model is an extensible object model[45]. The resource dissemination is push based with distributed query based resource discovery.

13. Bond

It is a service grid like Ninja but based on agents based resource discovery rather than query based working on Java based infrastructure using Knowledge Querying and Manipulation Language (KQML) [18, 19]. There are no autonomous domains in Bond and require extensible resource object model. Agent based resource discovery is followed in Bond with periodic push resource distribution.

14. CERN Data Grid

CERN, the European Organization for Nuclear Research, and the High-Energy Physics (HEP) community undertaken a project called ‘‘Research and Technological Development for an International Data Grid’’ [46]. CERN has multi-hierarchical machine organization. It has extensible schema model for resource model and distributed query based model for resource discovery.

15. Javelin

It is a Java based infrastructure to use as internet-wide parallel computing [37] constructed with clients, hosts and brokers as system components. If a client is seeking for resources, it need to be connected with broker to get access to the hosts attached with Javelin broker namely Broker Name Server (BNS). It offers QoS in load balancing for hosts. Javelin follows decentralized scheduling with push resource dissemination.

16. 2K

2K provides distributed operating system for a wide range of large scale distributed systems [16, 17]. It provides flexibility to operating systems for distributes services deployment using mobile agents. The broker in 2K is CORBA based called dynamic TAO[17]. 2K is considered as flat service grid using a decentralized machine organization. It uses extensible object model. Breadth of scope is high since multiple platforms are supported and QoS provision.

17. MOL (Meta Computing Online Kernel)

The core component of MOL is the kernel providing three-tier architecture for resource abstraction, access and management with high degree of reliability. MOL kernel is flexible since it can add or remove resources by dynamically reconfiguring itself [47]. MOL is considered as a service grid with hierarchical cells organization of machines. Push protocol is used for distributing resource information with a distributed query resource discovery.

18. Nimrod/G

It is a resource broker for managing and navigating application simulations on computational grids [48, 49] with the use of declarative language. It uses services of GRACE [50] to provide a low cost resource access. Nimrod/G provides support for managing application simulations, discovering resources, trading between resources, scheduling resources, navigating application simulations, and collecting results for user presentation. It follows a hierarchical machine organization and uses services of Globus and Leigon for resources discovery. It runs computational economy service of GRACE, thus supporting QoS.

19. GRACE (Grid Architecture for Computational Economy)

It works on principle of generic market based approach with co-existence of other grid systems like Globus [50]. The grid resource broker enable interactions of users with grid system. The resource broker allocates resource within the budget of users and jobs. GRACE uses Nimrod/G scheduler [48]not only for job scheduling but also for discovery, selection, and allocation of resources. The machine organization is decentralized and can be ad-hoc.

20. Cactus Worm

It is a demand-based grid computing system supporting adaptive application structure and dynamic resource features [20] through migration. Migration can only be performed when the performance is not as it is required thus provides QoS. The scheduling organization is ad-hoc based on requirements. Cactus Worm has high breadth of scope.

21. G-QoSM (Grid Quality of Service Management)

G-QoSM[51, 52] has Open Grid Service Architecture (OGSA) [61]. It support resource discovery with an agreement of providing QoS at application and network level. SLA between resource users and provider is established for providing resources since high breadth of scope is delivered. It can follow either centralized or decentralized scheduling organization.

22. Stanford peer initiatives

It makes use of peer-to-peer framework for trading and seeking the bids from distant web services with a reply of bid in response[53]. The lowest bid is selected by local web services to maximize the benefits. System preserves data by replicating the data for a possible period of time using a decentralized management system.

23. Wren [54]

It is a navigating approach for network measurement based on topologies[54] of clusters to Wide Area Networks (WANs) and figure out the possible holdups in network. The provided information is also useful to calculate the channel length using passive and active measurements. Scheduling organization can either be centralized or decentralized. QoS attributes are not being considered by Wren and breadth of scope is medium.

24. GHS (Grid Harvest Service)

GHS focuses on high scalability of network and its precision [55]. GHS has sub-systems for allocating tasks, managing execution, scheduling tasks, measuring and evaluating performance of system tasks. The performance can be improved by re-scheduling and making use of multiple scheduling algorithms such that first scheduling algorithm is dedicated for minimizing time for task execution whereas second will monitor the assignment of tasks to resources. The breadth of scope is medium in GHS.

TABLE 1 (a) Comparison of grid resource management systems - (a)

Name of System	Type of Grid	Machine Organization	Scheduling Organization	Resource information	Quality of Service (QoS)	Resource Discovery	Resource Distribution	Breadth of Scope
AppLeS[27] (Application Level Scheduling)	Computational	-	Hierarchical / Decentralized	Use model of with Globus, Legion, or NetSolve	No	Distributed query based system	Push	Low
Globus [28]	Grid Toolkit/ Middleware	Hierarchical Cells	Hierarchical	Extensible schema model	Yes	Distributed query based system	Push	High
NetSolve[30] (A Network Enabled Computational Kernel)	Computational & Service Grid	Hierarchical Cells	Decentralized	Extensible schema model	Yes	Centralized query-based system	Push	Medium
Darwin [31]	Service Grid	Hierarchical	Hierarchical	Fixed schema model	Yes	Distributed query based system	Push	Low
Condor [32] (Cycle Stealing Technology for High Throughput Computing)	Computational	Flat	Centralized	Extensible schema based model	No	Central query system	Push	Medium
Ninf[33] (Network Infrastructure)	Computational & Service Grid	Hierarchical	Decentralized	Extensible schema model	No	Centralized query-based discovery	Push	Medium
NWS [34] (Network Weather Service)	Service utilizing Grid	Hierarchical	Hierarchical	Extensible object model	Yes	Agent based system	Push	Low
PUNCH [35, 36] (Purdue University Network Computing Hubs)	Computational & Service Grid	Hierarchical	Decentralized	Extensible schema model	Yes	Distributed query based system	Push	Medium
Legion [37]	Computational	Flat Hierarchical	Hierarchical (ad-hoc)	Extensible object model	Yes	Distributed query based system	Pull	High
Lancaster DMRG [38, 39] (Distributed Multimedia Research Group)	Service Grid	Flat	Decentralized	Extensible object model	Yes	Distributed query based system	-	-
MSHN [40-42] (Management System for Heterogeneous Networks)	Computational & Service Grid	Flat	Centralized	Schema model	Yes	Centralized query based system	Push	-
Ninja [43-45]	Service Grid	Hierarchical	Decentralized	Extensible object model	No	Distributed query based system	Push	Medium
Bond [18, 19]	Service Grid	Flat	Decentralized	Extensible object model	Yes	Agent based system	Push	Low
CERN Data Grid [46]	Computational Data Grid	Hierarchical	Decentralized	Extensible schema model	No	Distributed query based system	Push	-

TABLE 1 (b) Comparison of grid resource management systems - (b)

Name of System	Type of Grid	Machine Organization	Scheduling Organization	Resource information	Quality of Service (QoS)	Resource Discovery	Resource Distribution	Breadth of Scope
Javelin [37]	Computational	Hierarchical	Decentralized	Fixed object model	Yes	Distributed query based system	Push	Low
2K [16, 17]	Service Grid	Flat	Decentralized	Extensible object model	Yes	Agent based system	Push	High
MOL [47] (Meta Computing Online Kernel)	Computational	Hierarchical Cells	Decentralized (ad-hoc)	Extensible schema model	No	Distributed query based system	Push	Low
Nimrod/G [48, 49]	Computational & Service Grid	Hierarchical Cells	Hierarchical	Use model of Globus or Legion middleware services with extension of computation al economy approach	Yes	Distributed query based system	Push	Medium
GRACE [50] (Grid Architecture for Computational Economy)	Computational	Hierarchical Cells	Decentralized (ad-hoc)	Schema based model extends with computation al economy approach	Yes	Distributed query based system	Push	High
Cactus Worm [20]	On-demand	Flat	Centralized (ad-hoc)	Schema based resource model	Yes	Central query system	Push	High
G-QoSM[51, 52] (Grid Quality of Service Management)	On-demand	Hierarchical	Centralized or Decentralized	Fixed object model	Yes	Distributed query based system	-	High
Stanford peer initiatives [53]	Computational	Hierarchical	Decentralized	Schema based model extends with computation al economy approach	Yes	Distributed query based system	Push	High
Wren [54]	Computational	Hierarchical	Centralized or Decentralized	Fixed object model	No	Distributed query based system	Push	Low
GHS [55] (Grid Harvest Service)	Computational	Hierarchical	Ad-hoc	Extensible object model	Yes	Distributed query based system	Push	Medium

5. Conclusion

The review of grid classification and anatomy provides a brief overview to enable a layman to understand the grid resource management systems. The management of resources is difficult for the reason of following limitations or characteristics of resources:

1. Heterogeneity of resources
2. Heterogeneity of platform
3. Geographically dispersed
4. Changing status of resources
5. Portability issues
6. Different policies administering resources

7. Autonomous nature
8. Huge number of resource requesters
9. Part of large distributed networks

To address the challenges posed by resource characteristics, grid systems must be consistent, efficient, time-saving and cost-effective in managing the resources available. Many techniques and systems have been proposed in past researches for resource management but there is no generic or complete solution suggested to the problem of resource management.

The systems are analyzed and compared on the basis of some common features. The study enables us to find that the systems are mostly computational or service grid. The Data grid, project by CERN is the only data grid. The grid schedulers like Nimrod/G and GRACE use resource models of other systems i.e., Globus or Legion. It is also investigated that a resource management system which provides QoS, makes available computational and service grid. But a resource management system without the support of QoS, makes computational grid available only. In the comparison, the resource management system with QoS provide variety of scheduling options like Cactus worm uses centralized and Darwin uses hierarchical. The resource distribution is correlated with the scheduling organization whereas the resource discovery is correlated with resource model. It was studied that query based system is used in schema models and objects models use the agent based system.

This study helped in identifying the resource management approaches and problems in grid systems. Further research can be carried out to examine the scheduling policies in grid systems in relation with grid sizes. The degree of extensibility of resource model either schema or based provides a room of future study. The need of time is to suggest a powerful, platform-independent and multi-user handling resource infrastructure to support resource management in grid environments.

REFERENCES

[1] I. Foster, 2000, "Internet computing and the emerging grid," nature web matters, vol. 7.

[2] M. M. Waldrop, 2002, "Grid computing," Technology Review (May 2002), vol. 31, p: 37.

[3] I. Foster, C. Kesselman, and S. Tuecke, 2003, "The anatomy of the grid," Berman et al.[2], pp: 171-197.

[4] Z. Balaton, Z. Farkas, G. Gombas, P. Kacsuk, R. Lovas, A. C. Marosi, et al., 2008, "EDGeS: the common boundary between service and desktop grids," Parallel Processing Letters, vol. 18, pp: 433-445.

[5] S. Choi, R. Buyya, H. Kim, E. Byun, and J. Gil, 2008, "A taxonomy of desktop grids and its mapping to state of the art systems," Grid Computing and Distributed Systems Laboratory, The University of Melbourne, Tech. Rep.

[6] A. Duarte, P. Nyczyk, A. Retico, and D. Vicinanza, 2008, "Monitoring the EGEE/WLCG grid services," in Journal of Physics: Conference Series, 119:052014.

[7] O. Smirnova, P. Eerola, T. Ekelöf, M. Ellert, J. R. Hansen, A. Konstantinov, et al., 2003, "The NorduGrid architecture and middleware for scientific applications," in Computational Science—ICCS 2003, ed: Springer, pp: 264-273.

[8] D. P. Anderson, J. Cobb, E. Korpela, M. Lebofsky, and D. Werthimer, 2002, "SETI@ home: an experiment in public-resource computing," Communications of the ACM, vol. 45, pp: 56-61.

[9] I. Foster and A. Iamnitchi, 2003, "On death, taxes, and the convergence of peer-to-peer and grid computing," in Peer-to-Peer Systems II, ed: Springer, pp: 118-128.

[10] K. Krauter, R. Buyya, and M. Maheswaran, 2002, "A taxonomy and survey of grid resource management systems for distributed computing," Software: Practice and Experience, vol. 32, pp: 135-164.

[11] T. F. C. Anglano, F.Giacomini, F. Prelz, M. Sgaravatto, 2001, "EDG WP01 Report on current technology".

[12] F. Pascual, K. Rzadca, and D. Trystram, 2009, "Cooperation in multi-organization scheduling," Concurrency and Computation: Practice and Experience, vol. 21, pp: 905-921.

[13] S. Meraji and M. R. Salehnamadi, 2013, "A Batch Mode Scheduling Algorithm for Grid Computing," Journal of Basic and Applied Scientific Research, vol. 3, pp: 173-181.

[14] B. Jacob, M. Brown, K. Fukui, and N. Trivedi, 2005, Introduction to grid computing: IBM, International Technical Support Organization.

[15] M. Maheswaran, 1999, "Quality of Service Driven Resource Management Algorithms for Network Computing," in PDPTA, pp: 1090-1096.

[16] F. Kon, R. H. Campbell, M. D. Mickunas, K. Nahrstedt, and F. J. Ballesteros, 2000, "2K: A distributed operating system for dynamic heterogeneous environments," in High-Performance Distributed Computing, 2000. Proceedings. The Ninth International Symposium on, pp: 201-208.

[17] D. Carvalho, F. Kon, F. Ballesteros, M. Roman, R. Campbell, and D. Mickunas, 2000, "Management of Environments in 2K," in Parallel and Distributed Systems, 2000. Proceedings. Seventh International Conference on, pp: 479-485.

[18] L. Boloni and D. C. Marinescu, 2000, "An object-oriented framework for building collaborative network agents," in Intelligent systems and interfaces, ed: Springer, pp: 31-64.

[19] K. Jun, L. Boloni, K. Palacz, and D. C. Marinescu, 2000, "Agent-based resource discovery," in Heterogeneous Computing Workshop, 2000.(HCW 2000) Proceedings. 9th, pp: 43-52.

[20] G. Allen, D. Angulo, I. Foster, G. Lanfermann, C. Liu, T. Radke, et al., 2001, "The Cactus Worm: Experiments with dynamic resource discovery and allocation in a grid environment," International Journal of High Performance Computing Applications, vol. 15, pp: 345-358.

[21] G. Kakarontzas and I. K. Savvas, 2006, "Agent-based resource discovery and selection for dynamic grids," in Enabling Technologies: Infrastructure for Collaborative Enterprises, 2006. WETICE'06. 15th IEEE International Workshops on, pp: 195-200.

[22] A. L. Lopes and L. M. Botelho, 2012, "Efficient algorithms for agent-based semantic resource discovery," in Agents and Peer-to-Peer Computing, ed: Springer, pp: 71-82.

[23] L. Han and D. Berry, 2008, "Semantic-supported and agent-based decentralized grid resource discovery," Future Generation Computer Systems, vol. 24, pp: 806-812.

[24] N. Russell, A. H. Ter Hofstede, D. Edmond, and W. M. van der Aalst, 2005, "Workflow resource patterns".

[25] H. Hussain, S. U. R. Malik, A. Hameed, S. U. Khan, G. Bickler, N. Min-Allah, et al., 2013, "A survey on resource allocation in high performance distributed computing systems," Parallel Computing, vol. 39, pp: 709-736.

[26] 2011, Continuous Availability for Enterprise Messaging: Reducing Operational Risk and Administration Complexity. Available: http://www.progress.com/docs/whitepapers/public/sonic/sonic_caa.pdf

[27] F. Berman and R. Wolski, 1997, "The AppLeS project: A status report," in Proceedings of the 8th NEC Research Symposium.

[28] I. Foster and C. Kesselman, , 1997, "Globus: A metacomputing infrastructure toolkit," International Journal of High Performance Computing Applications, vol. 11, pp: 115-128.

[29] K. Czajkowski, I. Foster, N. Karonis, C. Kesselman, S. Martin, W. Smith, et al., 1998, "A resource management architecture for metacomputing systems," in Job Scheduling Strategies for Parallel Processing, pp: 62-82.

[30] H. Casanova and J. Dongarra, 1997, "NetSolve: A network-enabled server for solving computational science problems," International Journal of High Performance Computing Applications, vol. 11, pp: 212-223.

[31] P. Chandra, Y.-H. Chu, A. Fisher, J. Gao, C. Kosak, T. E. Ng, et al. , 2001, "Darwin: Customizable resource management for value-added network services," Network, IEEE, vol. 15, pp: 22-35.

[32] J. Basney, M. Livny, and T. Tannenbaum, 1999, "Deploying a high throughput computing cluster," High performance cluster computing, vol. 1, pp: 356-361.

[33] H. Nakada, M. Sato, and S. Sekiguchi, 1999, "Design and implementations of ninf: towards a global computing infrastructure," Future Generation Computer Systems, vol. 15, pp: 649-658.

[34] R. Wolski, N. T. Spring, and J. Hayes, 1999, "The network weather service: a distributed resource performance forecasting service for metacomputing," Future Generation Computer Systems, vol. 15, pp: 757-768.

[35] N. H. Kapadia and J. A. Fortes, 1999, "PUNCH: An architecture for Web-enabled wide-area network-computing," Cluster Computing, vol. 2, pp: 153-164.

[36] N. H. Kapadia, R. J. Figueiredo, and J. A. Fortes, 2000, "PUNCH: Web portal for running tools," IEEE Micro, vol. 20, pp: 38-47.

[37] S. J. Chapin, D. Katramatos, J. Karpovich, and A. S. Grimshaw, 1999, "The legion resource management system," in Job Scheduling Strategies for Parallel Processing, pp: 162-178.

[38] G. Coulson, 1999, "A configurable multimedia middleware platform," IEEE Multimedia, vol. 6, pp: 62-76.

[39] G. Coulson and M. Clarke, 1998, "A distributed object platform infrastructure for multimedia applications," Computer Communications, vol. 21, pp: 802-818.

[40] D. A. Hensgen, T. Kidd, D. St John, M. C. Schnaidt, H. J. Siegel, T. D. Braun, et al. , 1999, "An overview of MSHN: The management system for heterogeneous networks," in Heterogeneous Computing Workshop, 1999.(HCW'99) Proceedings. Eighth, pp: 184-198.

[41] G. G. Xie, D. Hensgen, T. Kidd, and J. Yarger, 1998, "SAAM: An integrated network architecture for integrated services," in Quality of Service, 1998.(IWQoS 98) 1998 Sixth International Workshop on, pp: 117-126.

[42] M. Schnaidt, D. Hensgen, J. Falby, T. Kidd, and D. St John, 1999, "Passive, domain-independent, end-to-end message passing performance monitoring to support adaptive applications in MSHN," in High Performance Distributed Computing, 1999. Proceedings. The Eighth International Symposium on, pp: 337-338.
[43] S. D. Gribble, M. Welsh, E. A. Brewer, and D. E. Culler, 1999, "The MultiSpace: An Evolutionary Platform for Infrastructural Services," in USENIX Annual Technical Conference, General Track, pp: 157-170.
[44] S. E. Czerwinski, B. Y. Zhao, T. D. Hodes, A. D. Joseph, and R. H. Katz, 1999, "An architecture for a secure service discovery service," in Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking, pp: 24-35.
[45] T. D. Hodes and R. H. Katz, 1999, "ADOCUMENT-BASEDFRAMEWORKFOR INTERNET AP PLIC AT IONCON TR OL".
[46] W. Hoschek, J. Jaen-Martinez, A. Samar, H. Stockinger, and K. Stockinger, 2000, "Data management in an international data grid project," in Grid Computing—GRID 2000, ed: Springer, pp: 77-90.
[47] J. Gehring and A. Streit, 2000, "Robust resource management for metacomputers," in High-Performance Distributed Computing, 2000. Proceedings. The Ninth International Symposium on, pp: 105-111.
[48] R. Buyya, D. Abramson, and J. Giddy, 2000,"Nimrod/G: An architecture for a resource management and scheduling system in a global computational grid," in High Performance Computing in the Asia-Pacific Region, 2000. Proceedings. The Fourth International Conference/Exhibition on, pp: 283-289.
[49] R. Buyya, J. Giddy, and D. Abramson, 2000, "An evaluation of economy-based resource trading and scheduling on computational power grids for parameter sweep applications," in Active Middleware Services, ed: Springer, pp: 221-230.
[50] R. Buyya, D. Abramson, and J. Giddy, 2001, "A case for economy grid architecture for service oriented grid computing," in Parallel and Distributed Processing Symposium, International, pp. 20083a-20083a.
[51] R. J. Al-Ali, A. Hafid, O. F. Rana, and D. W. Walker, 2003, "QoS Adaptation in Service-Oriented Grids," in Middleware Workshops, pp: 200-210.
[52] R. J. A. Ali, O. F. Rana, D. W. Walker, S. Jha, and S. Sohail, 2012, "G-QoSM: Grid service discovery using QoS properties," Computing and Informatics, vol. 21, pp: 363–382.
[53] B. F. Cooper and H. Garcia-Molina, 2002, "Bidding for storage space in a peer-to-peer data preservation system," in Distributed Computing Systems, 2002. Proceedings. 22nd International Conference on, pp: 372-381.
[54] B. B. Lowekamp, 2003, "Combining active and passive network measurements to build scalable monitoring systems on the grid," ACM SIGMETRICS Performance Evaluation Review, vol. 30, pp: 19-26.
[55] X.-H. Sun and M. Wu, 2005, "GHS: a performance system of grid computing," in Parallel and Distributed Processing Symposium, 2005. Proceedings. 19th IEEE International, pp: 228a–233.