AN ALTERNATE APPROACH TO RESOURCE ALLOCATION STRATEGY USING NETWORK METRICSIN GRID

C.Valliyammai, S.ThamaraiSelvi, R.Satheesh Kumar, E.Pradeep, and Naveen.K

Department of Information Technology, Anna University, Chennai

cva@annauniv.edu, stselvi@annauniv.edu, satheesh.ravindranath@gmail.com,
pradeepfree4u@gmail.com, naveen19892000@gmail.com

ABSTRACT

Monitoring in a grid environment involves the analysis of all resource metrics and network metrics. Monitoring the resource metrics helps the grid middleware to decide which job to be submitted to which resource. Decision for submitting a job will be better along with the consideration of the network metrics. Tuning of the network metrics is also made if the performance degrades.

KEYWORDS

Grid Monitoring; Network metrics; Resource metrics; Resource Selection, Network tuning, Mobile Agents.

1 INTRODUCTION

Grid environment is a large scale distributed network which consists of loosely coupled nodes acting in concert to perform intricate tasks. The nodes in the grid network may have different architecture in terms of operating systems or hardware. This results in a heterogeneous environment and is often in a decentralized network, rather than in a single location. Grid computing field involves large amount of research projects and has showed to be a integrated method of solving a given complex problem using highly configured computational computers.

2 NEED FOR GRID MONTIORING

The high-level computing jobs can be efficiently performed by analyzing various parameters that impact the process of computing. This process of analyzing the various parameters of the grid setup is known as monitoring [11]. Satoshi describes the need for maintaining the level of quality of the grid setup [26]. There are various factors affecting the quality such as network faults, component interdependencies etc. The quality of the grid setup can be maintained by regularly monitoring the activities within the grid setup, through the process of grid monitoring. This process of monitoring gives the details of the current execution scenario and can also help in predicting the future performance of the setup which will be useful to estimate the time required for completion of jobs as specified in [31]. After the submission of jobs to the grid

DOI : 10.5121/ijgca.2010.1203

setup, users often experience delay in job completion. This can be reduced to a great extent if the grid setup is monitored regularly and the performance problems [2] detected are rectified soon. The purpose of grid monitoring extends its concept leading to performance prediction and performance tuning of grid setup. In the grid setup there are various issues arising during the job execution. These issues namely the delayed job execution, blockage of the job may be due to the resource metrics or the network metrics. These issues should be identified by the process of monitoring, and should be corrected by the process of tuning. The process of identifying and tuning of grid performance is robust since the resources are distributed at different geological location and are connected together by network links [3]. Wu-Chun Chunga and Ruay-Shiung Changhave proposed an efficient protocol called the Grid Resource Information Retrieving (GRIR)protocol [10], which is based on the push data delivery model to obtain the accurate network status.

A monitoring and information system (MIS) is a key component of a distributed system or Grid, which provides information about the available resource metrics and their status. MIS can be used in a variety of ways: a resource broker may query the MIS to locate computing elements for the CPU and memory requirements according to a job submitted by the end-users; a program may collect a stream of data generated by MIS to direct an application or to a system administrator to send a notification when system load or disk space availability changes while identifying the possible performance anomalies [4].

There are two types of monitoring.

i. Active monitoring - Few test packets are injected in the original data channel and the performance is monitored. This measures the behavior of the packets on the network.

ii. Passive monitoring – Some observation posts are formed to monitor the flow of data packets without disturbing the actual flow of the data packets. This measures the behavior of the application while using the network.

3 NETWORK MONITORING

Albert describes the use of networked computational resources for the implementation of high sensor applications [28]. These high sensor applications required parallel computing in which the network performance is vital and needs to be monitored regularly. The process of network monitoring involves the evaluation of network performance of the links between the clients and head node of the grid setup. Some of the common metrics identified for network monitoring [6] viz., latency, jitter, packet loss, throughput, link utilization, availability and reliability. *Latency* in a network may vary because of the congestion in the channel, router, load of the end – end hosts and also the path followed by the packet during it's to and fro travel. *Jitter* generally means short-term variations. Jitter is a delay that varies over time. Jitter is also known as variation latency. *Packet loss* may take place due to hardware fault, congestion in the channel, corruption in the data packet sent. *Throughput* is constituted by several parameters namely, packet loss ratio, latency, jitter, delay, round trip time and available bandwidth. *Link utilization* can be calculated from the above throughput divided by the access rate and expressed in percentage. For some types of link, the service provider may give Committed Information Rate (CIR).*Availability* refers to the channel availability for a particular application to use at certain point of time. *Reliability* is related with the packet loss ratio and availability. This also involves the retransmission rate. The system administrators and application developers need variety of monitoring tools to analyze various network metrics such as round trip time, packet loss, bandwidth, jitter, latency, throughput etc. Various network-monitoring tools are available which helps in the efficient monitoring of the network [20]. In [32], various network metrics are considered for monitoring and are also tuned for the better performance of the grid setup. The network metrics and the cost function that are used in [32] shall be useful in estimating the overall performance of the grid setup.

Monitoring the network performance of the entire grid setup involves monitoring all the individual links between the nodes of the grid setup. Thus a Compound Cost Function (CCF) is required to monitor the overall network performance of the grid setup. The Compound CF is a function of CFs of all existing links in the grid setup. This Compound Cost Function (CFF) is designed to reflect the variation in each of the individual Cost Function (CF). Thus the overall network performance of the grid setup is determined using the Compound Cost Function (CFF) as shown in Equation (1).

$$\boldsymbol{CCF = f\ (CF_1, CF_2, CF_3, \dots, CF_n)} \qquad \text{----- (1)}$$

Where,

CF1, CF2, CF3, ..,CFn are the Cost Function values of individual links of the grid setup.

$$\mathbf{CF} = \frac{\mathbf{BW_{avg}}}{\mathbf{BW_{max}}} * \mathbf{e}^{-\frac{\tau^2}{2}} * \frac{\mathbf{lat_{min}}}{\mathbf{lat_{avg}}} * \boldsymbol{\alpha}^{\mathbf{p}} * \boldsymbol{\beta}^{\mathbf{jitter}} * \frac{\mathbf{Delay_{avg}}}{\mathbf{Delay_{max}}} \qquad \text{----- (2)}$$

Where,

$$BW_{avg} = \frac{1}{n}\sum_{i=1}^{n-1} BW_i$$

$$Delay_{avg} = \frac{1}{l}\sum_{j=1}^{l-1} Delay_j$$

$$\tau\ = 1 - \ [\mathrm{RTT_{avg}}/\mathrm{RTT_{max}}]$$

$$RTT_{avg} = \frac{1}{l}\sum_{k=1}^{l-1} RTT_k$$

p = Packet loss rate.
i = Number of values taken for calculating the mean bandwidth.
j = Number of values taken for calculating the mean one-way delay.
k = Number of values taken for calculating the mean Round Trip Time (RTT).
α = Varies in interval [0, 1] and tuned to balance the dependency of packet loss rate.
β = Varies in interval [0, 1] and tuned to balance the dependency of jitter.
latmin = Minimum latency between the corresponding pair of nodes.
latmin = Mean latency between the corresponding pair of nodes.
RTTmax = Maximum Round Trip Time between the corresponding pair of nodes.
BWmax = Maximum available bandwidth between the corresponding pair of nodes.
Delayavg = Mean one-way delay between the corresponding nodes.
Delaymax = Maximum one-way delay between the corresponding nodes.

3.1. Tuning

3.1.1. Bandwidth and One-way Delay

The network performance of the grid setup is improved by dynamically tuning the buffer size of the link, which has degraded. The buffer size is set to an optimal value based on the values of average available bandwidth and delay. The optimal socket buffer size is twice the size of the bandwidth * delay product of the link [6] as shown in Equation (3).

$\boldsymbol{Buffer\ capacity = 2 * BW * Delay}$ ----- (3)

Where,

BW = average available bandwidth

Delay = delay between the head node and the compute node.

The TCP throughput increases by altering the send and receive socket buffer size of the compute nodes [19].

3.1.2. Bandwidth and RTT

$$\boldsymbol{Buffer\ Size = RTT * BW} \qquad \text{---- (4)}$$

Where,

BW = average available bandwidth

RTT = Round Trip Time

Buffer size is used to alter the TCP socket buffer size (send and receive buffers) as shown in Equation (4). The kernel parameters namely 'tcp_rmem' and 'tcp_wmem' are changed. Throughput is considered as the variable bandwidth and hence the computation is performed.

3.1.3. Bandwidth and Latency

$$\boldsymbol{Buffer\ size = Throughput * Latency} \qquad \text{----- (5)}$$

Sending and receiving TCP socket buffer size is altered with the help of the value of the product of the throughput and the latency as shown in Equation (5). Every time while monitoring the threshold value is checked for and then decision of tuning is taken.

3.1.4. Jitter

The kernel parameter tcp_low_latency when set to 0, gives a high throughput. When it is set to 1, gives a low latency performance. Hence 'tcp_low_latency' is set to '1' when a high jitter value is identified and it is set to '0' when there is packet loss. The threshold values changes according to the environment.

3.1.5. Packet Loss

The kernel parameter 'tcp_fack' is available to select the forward acknowledgement protocol ON or OFF. When 'tcp_fack' is set to 0, congestion control algorithm is disabled. When it is set to 1, congestion control algorithm is enabled. Hence 'tcp_fack' is set to '1' when there is no packet loss and 'tcp_fack' is set to '0' when there is packet loss.

4 RESOURCE MONITORING

By maintaining the resource status constantly, the necessary information can be quickly provided as requested. However, the maintenance cost of resource status is heavily related to the total number of resources and number of times the status is updated. Therefore, trade-off between maintenance cost and data accuracy should be considered. Resource monitoring involves monitoring the available resources whenever a job is submitted to the grid middleware. A submitted job is often executed whenever sufficient amount of resources are freed by other jobs. Rajkumar enumerates various issues in Grid Resource Management [27]. The resources are geographically distributed and have their own scheduling mechanisms, prices, access permissions. All these factors need to be managed properly to provide better system performance and user satisfaction. Fufang describes the agent based resource management system [29] where the agents are designed to locate the largest available computation power within the grid setup and provide proper load balance. Junwei also describes the advantages of Agent-Based Resource Management Infrastructure [30], which reduces two major challenges of *adaptability* and *scalability* in grid environment.

In [10], Wu-Chun has introduced a Grid Resource Information Monitoring (GRIM) prototype. To take into account the dynamicity of the changing resources in grid, a push based data delivery protocol called Grid Resource Information Retrieving (GRIR) is used. Resource information is updated completely based on its availability and the requirement of sufficient resource metrics. One of the prominent techniques for resource monitoring is Grid Monitoring Architecture (GMA). Resources available in grid network are present in Producer or product Service. Monitoring of the resources in grid is done by the Consumer Service. Director Service is one, which makes the bridge between the consumers and the producers [22].

Resource monitoring will identify the bottlenecks among the available resources in the grid network. Globus Alliance developed the Globus toolkit, in which the *Monitoring and Discovery System (MDS)* is the most prominent monitoring software. The status of the resources is gathered by the Information provider [23].Swift Scheduler allocates jobs in Computational Grid by considering the length of the jobs, processing time, jobs' memory, and CPU requirements with respect to the priority of resources [23].

5 RESOURCE MONITORINGVS NETWORK MONITORING

In general, the node selection procedure for the job execution is done based on the resource parameters such as computational speed, CPU usage, memory etc. These parameters decide which node has the capability of performing the computation efficiently. In some cases the selection of nodes is done by also considering the network performance between the links to the nodes. Though the nodes may have high resource availability, it may delay the execution of the job due to network performance degradation. To reduce this kind of issues, the network parameters are also to be considered along with the resource parameters during node selection. Job submission is based on both CPU loads among the servers and latencies available in the network [24].

Though the resource parameters may be monitored well, the network parameters may have a role in effective transfer of data between the compute and head nodes. The need for network monitoring in grid is mainly when the parallel jobs are submitted to compute nodes with equal resource performance. In this case though the resource performances of the compute nodes are equal, the response from both the nodes may not be received in the same time. This effect is due to the variation in network performance between the links of the computational nodes. The purpose of network performance monitoring becomes crucial as the network size of the grid setup increases. This involves monitoring of more number of links on the network and estimates its network performance and support fault detection [6]. Collecting, relating and analyzing of network information are one of the important aspects of effective grid application and services. GMMPro, grid network and monitoring system provides the basic support for monitoring the grid network and has SNMP as its lower layer protocol [5].

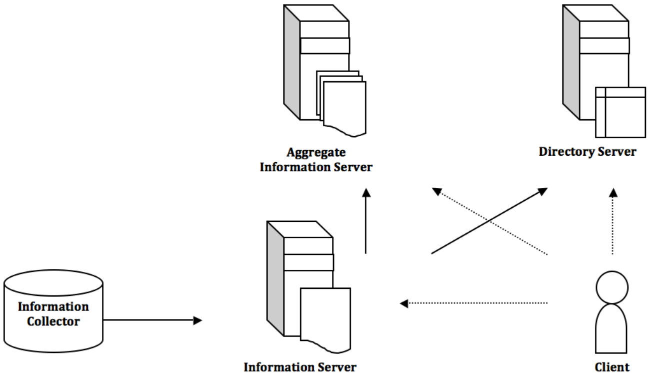

Figure 1 – General Model of monitoring system.

Figure 1 shows a general model of a monitoring system. The client queries and gets the information from all the servers.

The Global Grid Forum (GGF) has identified a Grid Monitoring Architecture (GMA) (Tierney et al 2002) with three major components namely producer, consumer and registry. The GMA model is shown in Figure 2. A producer registers a description of its event stream with the directory service. A consumer contacts the directory service to locate producers that have data relevant to its query. A communication link is then set up directly with each producer to acquire data, either by a publish/subscribe protocol, or by a query/response protocol. Consumers may also register with the directory service. These are then notified whenever new producers become available.

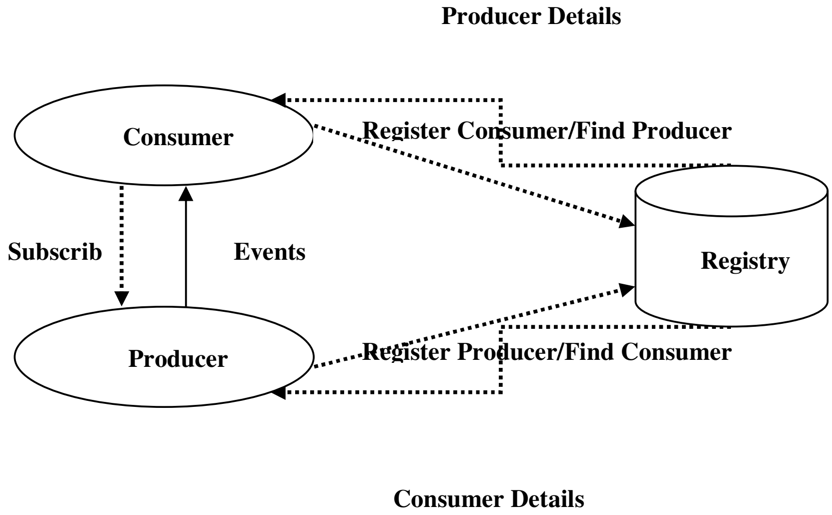

Figure 2. Components of GMA

Intermediarycomponents may be set up that consist of both a consumer and a producer. Intermediaries may be used to forward, broadcast, filter, and aggregate or archive data from other producers. The intermediary then makes this data available for other consumers from a single point in the grid.

By separating the tasks of information discovery, enquiry, and publication, the GMA is scalable. However, the GMA does not define a data model, query language, or a protocol for data transmission. It does not specify details about how information should be stored in the directory service.Different monitoring systems implement each block in their own way as compared in Table 1.

Table 1 – Comparison of Existing monitoring systems

	MDS-2	**R-GMA**	**Hawkeye**
Information Collector	Information Provider	Producer	Module
Information Server	GRIS	Producer Servlet	Agent
Aggregate Information Server	GIIS	Producer and Consumer	Manager
Directory Server	GIIS	Registry	Manager
Use Case	Single Query	Streaming Network Data	Single Query
Push/Pull model	Pull	Push and Pull	Triggers and Push

GRIS - Grid Resource Information Service
GIIS - Grid Index Information Service

6 GRID NETWORK MONITORING USING AGENTS

Grid environments have recently emerged as integrating infrastructure for distributed high-performance scientific applications (Foster and Kesselman 1999). Unexpected low throughput or high latency is common performance problems identified in high-performance distributed systems.Detailed end-to-end monitoring data from applications, networks, operating systems, and hardware must be correlated across time and space in order to determine the source of these problems. Detailed comparison of these data from a variety of angles is desired. To address this problem, a Mobile Agent based architecture that is designed to efficiently handle high-volume streams of monitoring data is proposed. This architecture is designed to be scalable and fault tolerant. In a complex system like grid, monitoring is important for understanding its operation, debugging, and failure detection and for performance optimization. A flexible monitoring architecture is introduced that provides advanced functions like actuators and guaranteed data delivery. Mobile Agent based network management equips agents with network management capabilities and allows them to issue requests to managed devices (or nodes) after migrating to these nodes.Mobile Agents give the flexibility of analyzing the managed node locally. Instead of querying the managed node for every fixed interval and analyzing the performance from management station, Mobile Agent can be dispatched to analyze the node locally. Due to its robustness, fault detection capabilities, less network usage, Mobile Agents are used for network management tasks.

7 MOBILE AGENT TECHNOLOGY

Mobile agents are considered one of the most powerful forms of code mobility (Giovanni 2004). They can exploit the high processing power available in the server machines by shifting the computations into the server side. A mobile agent is a software module able to migrate to among the hosts of a network and carry on a specific task. The state of the program that runs is saved, transported to the new host, and restored, allowing the program to resume where it left off.Mobile agent technology seems to be very adequate to cope with systems' heterogeneity and to deploy user customized procedures on remote sites (Kameshwari et al 2007). Mobile agents are autonomous and intelligent programs that moves through a network, searching for and interacting with services on the user's behalf and possess inherent navigational autonomy (Martin et al 2007). The mobile agents' role is vital in grid because to support the local monitoring, local correlation on management devices, deploying agents dynamically, good scalability, continuing execution on device when link-down or unreliable and return results when available (Puliafito and Tomarchio 2000).

Mobile Agents provide an effective way of migrating code and data together and return the results to the original user. Unlike RMI, which transfers data alone using stubs and skeletons, Mobile Agents uses the concept of migration of the code and the data. Network Monitoring in Grids using Mobile Agent help us to effectively utilize the idle resources available in the geographically separated areas in a more optimized way. The choice between mobile agents and client-server paradigm is discussed in (Antonio et al 1997) and their performance comparison is dealt in (Fuggetta et al 1998). Recently, comparison between mobile agent and RMI-based applications has attracted a growing attention. This is probably because mobile agents have a better fault tolerance when compared to the RMI (Aderounumu et al 2006).

Distributed information processing is a primary application of agent technology (Yousry et al 2007). Consider a mobile computing scenario with low bandwidth connection links. Agents suit them in a better way because the code to be executed migrates into the network leaving the portable device and performs necessary actions in the remote execution site. Only the result is returned back to the mobile device (Robert et al 2001).The techniques devised for protecting the agent platform are software-based fault isolation (Wahbe et al 1993), safe code interpretation, signed code, authorization and attribute certificates, state appraisal (William et al 1996), path histories (David et al 1995) and proof carrying code (Necula and Lee 1996). While those for protecting an agent are partial result encapsulation, mutual itinerary recording, itinerary recording with replication and voting, execution tracing (Giovanni 1997), environmental key generation (James and Bruce 1998), computing with encrypted functions (Thomas and Christian 1998) and obfuscated code.

Mobile Agent Paradigm has the following advantages in the field of Network Management:

- Reduction of the network load.
- Opportunity of performing operations of monitoring data analysis by means of algorithms that can be customized by the user and can be dynamically executed on-demand.
- Filtering of monitoring data at several abstraction levels, without high overheads for the system.
- Asynchronous and independent execution of tasks defined by the user.
- Integration of heterogeneous resources monitoring tools.
- On-demand enabling of the services.

The use of mobile agents has several potential benefits:

- Asynchronous task execution.
- More dynamics.
- Reduced communication.
- Improved real time abilities.
- Higher degree of robustness.
- Improved support of nomadic computing and intermittently connected devices.

8 RESEARCH ISSUES

There are various research issues in improving the efficiency of grid setup. Though the resource metrics contribute to the decision making of the resource selection, network metrics do play a significant role in deciding the hosts for job execution. The network metrics gives more detailed information regarding the quality and the performance of the hosts. Resource metrics gives information of the local system's efficiency only in terms of CPU utilization, concurrent processing, and memory utilization. There is a need of overlooking the degradation due to network metrics while considering the resource metrics too. Hence, analysis of the network metrics along with the resource metrics for the selection of the compute nodes during job submission is essential. The strategy for resource allocation can be designed in the form of an algorithm. The algorithm can be optimized to handle both the resource metrics as well as the network metrics.

One of the key issues is to consider the network metrics for monitoring and prediction of the grid setup apart from the regular resource metrics used. The efficiency of a grid setup can be estimated more accurately considering the network metrics. The existing resource selection and job scheduling algorithms can be altered by including the network metrics to improve the efficiency. Such improved version of algorithms will also consider the dynamic change in network load and other bottleneck situations. Another research issue is to optimize the dynamically varying network load by the process of tuning. The impact of network monitoring and tuning has to be optimized such that it doesn't source the bottleneck situation while monitoring and as well as in tuning.

REFERENCES

1. Tommaso Coviello, Tiziana Ferrari, Kostas Kavoussanakis, Loukik Kudarimoti, Mark Leese, Alistair Phipps, Martin Swany, Arthur S. Trew, "Bridging Network Monitoring and the Grid", CESNET 2006, Conference on Advanced Communications and Grids.
2. Dan Gunter, Brian Tierney, Keith Jackson, Jason Lee, Martin Stoufer, "Dynamic Monitoring of High-Performance Distributed Applications", 11th IEEE International Symposium on High Performance Distributed Computing 2002,Page: 163.
3. A.P. Millar, "Grid monitoring: a holistic approach", Grid PP UK Computing for Particle Physics (2006).
4. Xuehai Zhang, Jeffrey L. Freschl, Jennifer M. Schopf, "Scalability analysis of three monitoring and information systems: MDS2, R-GMA, and Hawkeye", Journal of Parallel and Distributed Computing, Volume 67, Issue 8, 2007.
5. WANG Junfeng, ZHOU Mingtian, ZHOU Hongxia, "Providing Network Monitoring Service for Grid Computing", Proceedings of the 10th IEEE International Workshop on Future Trends of Distributed Computing Systems (FTDCS'04).
6. Mark Leese, RikTyer and Robin Tasker, "Network Performance Monitoring for the Grid", (UK e-Science, 2005 All Hands Meeting) http://gridmon.dl.ac.uk/.
7. UDPmon webpage, http://www.hep.man.ac.uk/u/rich/net/index.html
8. TCPmon webpage, http://www.hep.man.ac.uk/u/rich/Tools_Software/tcpmon.html
9. R. E. Hughes-Jones,"Writeup for UDPmon: A Network Diagnostic Program", 2004.
10. Wu-Chun Chunga and Ruay-Shiung Chang, "A new mechanism for resource monitoring in Grid computing", 2008.
11. SerafeimZanikolas, RizosSakellariou, "A taxonomy of grid monitoring systems", Future Generation Computer Systems 21 (2005) 163–188.
12. A.C. Davenhall& M.J. Leese, "An Introduction to Computer Network Monitoring and Performance", 2005.
13. Pinger webpage, http://www-iepm.slac.stanford.edu/pinger/

14. David Medinets& David A. Cafaro, “Monitoring and scheduling”, IBM, 2007.
15. V. Jacobson, “Traceroute: A tool for printing the route packets take to a network host”, available from ftp.ee.lbl.gov/nrg.html.
16. V. Jacobson, C. Leres, S. McCanne, tcpdump, available at ftp://ftp.ee.lbl.gov/tcpdump.tar.Z.
17. B. Mah, "pchar: A tool for measuring Internet path characteristics," http://www.employees.org/bmah/Software/pchar/.
18. J. Goujun. “Methods for Network Analysis and Troubleshooting” http://www-didc.lbl.gov/~jin/network/net-tools.html.
19. IPerf: http://openmaniak. Com/iperf.php.
20. Thomas J. Hacker, Brian D. Athey, Jason Sommerfield, Pittsburgh, Deborah S. Walker, "Experiences Using Web100 for End-to-End Network Performance Tuning for Visible Human Testbeds".
21. List of network measurement tools: http://ncne.nlanr.net/software/tools/.
22. Brian Tierney, Ruth Aydt, Dan Gunter, Warren Smith, Martin Swany, Valerie Taylor, Rich Wolski, A grid monitoring architecture, The Global Grid Forum Draft Recommendation (GWD-Perf-16-3), August 2002.
23. Monitoring and discovery system, http://www.globus.org/toolkit/mds/.
24. K. Somasundaram, S. Radhakrishnan, “Task Resource Allocation in Grid using Swift Scheduler”, Int. J. of Computers, Communications & Control, ISSN 1841-9836, E-ISSN 1841-9844 Vol. IV (2009), No. 2, pp. 158-166 IEEE Transactions on Parallel and distributed Systems.
25. SupranamayaRanjan, Edward Knightly, “High Performance Resource Allocation and Request Redirection Algorithms for Web Clusters”.
26. Ken’ichiroShirose, Satoshi Matsuoka, "Autonomous Configuration of Grid Monitoring Systems", Proceedings of the 2004 International Symposium on Applications and the Internet Workshops (SAINTW’04).
27. RajkumarBuyya†, David Abramson†, and Jonathan Giddy, "Grid Resource Management, Scheduling and Computational Economy", In Proceedings of the 2nd International Workshop on Global and Cluster Computing (WGCC’2000).
28. Albert Reuther and Joel Goodman, "Dynamic Resource Management for a Sensor-Fusion Application via Distributed Parallel Grid Computing".
29. Fufang Li, Deyu Qi, Limin Zhang, Xianguang Zhang, and Zhili Zhang, "Research on Novel Dynamic Resource Management and Job Scheduling in Grid Computing", Proceedings of the First International Multi-Symposiums on Computer and Computational Sciences (IMSCCS'06).
30. Junwei Cao, Darren J. Kerbyson, and Graham R. Nudd, "Performance Evaluation of an Agent-Based Resource Management Infrastructure for Grid Computing".
31. C.Valliyammai, S.ThamaraiSelvi, M.Santhana Kumar, S.Sathish Kumar, R.Suresh Kumar, “Network Performance monitoring using mobile agents in grid”2009 IEEE International Advance Computing Conference.
32. C. Valliyammai, S. Thamarai Selvi, R. Satheesh Kumar, E. Pradeep, and K. Naveen, "Multi-agent Based Network Performance Tuning in Grid", In BAIP 2010, Springer LNCS, Information Processing and Management, Vol 70, pp. 297–304.
33. C. Valliyammai, S. ThamaraiSelvi, R. Satheesh Kumar, E. Pradeep and K. Naveen,”Novel Resource Allocation Strategy Using Network Metrics in GRID”, In NeCom 2010, Springer LNCS, “Recent Trends in Networks and communication”, Vol 90, pp. 107-113.

AUTHORS

S.ThamaraiSelvi received the B.E. Degree in mechanical engineering from AlagappaChettiar College of Engineering and Technology affiliated to Madurai kamarajUnniversity, and the M.E. degree in computer science and Engineering from GCT, Coimbatore and completed her Ph.D in computer science and engineering in ManonmaniamSundaranar University, Tirunelveli. She has 24 years of teaching experience. Her area of interest includes artificial neural networks, and Grid computing. She is handling five research projects funded by Central government Agencies.
E-mail: stselvi@annauniv.edu.

C.Valliyammaiis currently doing her Ph.D in computer science and engineering at the Anna University. She has eight years of teaching experience. Her area of interest includes network management, Grid computing and mobile agents. She published ten papers in journals and conferences.
E-mail: cva@annauniv.edu.

R.Satheesh Kumar has pursued his Bachelor of Technology in Information Technology atAnna University, Chennai, India.

He was an intern at Microsoft, IDC, Hyderabad, India as a PROGRAM MANAGER during May – July 2009. His research interests include networks, information security, cryptography and grid computing.

Mr.Satheesh is a Student member of IEEE.

Pradeep.Ehas pursued his Bachelor of Technology in Information Technology at Anna University, Chennai, India.

He has completed CCNA training at Cisco Labs. He is an alumnus of Cisco Networking Academy. His research interests include cryptography, network security, computer networks and grid computing.

Mr.Pradeep is a member of IEEE Communications Society and International Association of Engineers (IAENG).

Naveen.Khas pursued his Bachelor of Technology in Information Technology at Anna University, Chennai, India.

He has completed certified network courses conducted by Microsoft and Cisco. He holds MCP, MCSE, MCTS and CCNA. He is also pursuing CCNP and CWNA. His reach interests include networks, information security and grid computing.