International Journal of Recent Technology and Engineering (IJRTE)
ISSN: 2277-3878, Volume-8 Issue-3, September 2019

IRMM: An Integrated Resource Management Model for Grid Computing

Makhan Singh

Abstract: ***Grid computing is a collection of heterogeneous systems or heterogeneous objects that are geographically distributed over a network. Resource management is a process in which various activities like allocation of resources and scheduling are performed for handling issues like load balancing, reliability, scalability, maximum, throughput, minimum expectation time and security. There are several factors that make resource management difficult as different system may have different requirements, properties, conditions and different access and cost models. Resource management in Grid is the method of identifying requirements, finding corresponding resources to the applications, allocating those matching resources, scheduling and monitoring. In Grid resource management resource broker plays the very important role. Users communicate with a resource broker to access the grid information. Resource broker discover the resource that are available and negotiates with their owners or their agents to get the reservation of resources. Number of approaches exists through which one can develop grid resource management systems. In this paper a new architectural model has been implemented for grid resource management which is based on the characteristics of both the Economical model and Hierarchical model.***

Keywords: Grid Computing, Resource Management, Scheduler, Economical model and Hierarchical model.

I. INTRODUCTION

Grid computing is a technology that combines a number of resources distributed across wide area networks (WAN) that can support distributed applications on a large scale. Grid computing is the process of sharing processes over multi-computers. Grid computing applies several computer resources in a network simultaneously to a single problem. Grid computing allows the virtualization of distributed computing and data resources such as processing, network bandwidth and storage capacity to create a single system image, providing seamless access to vast IT capabilities for users and applications. As Grid logically combines various resources owned by distinct people or organizations, choosing the correct resource management architecture model plays a significant part in its commercially success [1]. In creating grid resource management systems, there are a number of methods that can be followed. The structure of the scheduler influences the architectural model of resource management schemes. Scheduler structure relies on how many resources are scheduled for jobs and computations, and the domain where resources are placed. There are numerous proposed methods through which Grid Resource Management Systems can be developed.

Manuscript published on 30 September 2019
* Correspondence Author
Makhan Singh*, Computer Science & Engineering, UIET, Panjab University, Chandigarh, India. Email: singhmakhan@pu.ac.in

II. Related Work

In literature, there are many Architectural Models of Resource Management which are discussed in detail as follows.

A. The Hierarchical Model

This Model for Grid Resource Management Architecture is an outcome of the Grid Forum. It enables owners of remote resources to implement their own external users policy. This model combines a centralized model with a decentralized model, but it looks more like a centralized model. A job when submitted by user to job control agent, which further calls an admission agent. The admission agent checks the resource demands of the job (maybe consulting with a grid information system) and determines that adding the job to the system's present work pool is secure. The job is passed by admission agent to the scheduler, which uses the Grid Information System to perform resource discovery and then consult with domain control agents to determine the current resource state and availability. The set of mapping are computed by the scheduler and passed to deployment agent. The deployment agent negotiates for the resources specified in the schedule with the domain control agents and obtains reservations for the resources. These reservations are passed to the job control agent. At the proper time, the job control agent works with a different deployment agent, and the deployment agent coordinates with the appropriate domain control agents to start the tasks running. The progress of the job is tracked by the monitor. The decision of rescheduling may be done later if performance is less than expected [2].

B. Economy Model

The Grid Resource Management Model for Economy/Market Resource Management captures both Hierarchical and Abstract Owner Model's essential elements. Many of the modern grid systems fit the hierarchical model, and Abstract Owner seems to be futuristic, but implicitly points to the computational need for economy [3]. The problems mentioned in the hierarchical model are relevant to the market model, but they emphasize the use of resource management and scheduling based on the economy.

C. Abstract Owner Model

The Abstract Owner Model removes some of the limitations of the hierarchical model as in this model users directly communicate with the owner for resource reservation. It has two windows, which are, Order Window and Pickup Window. The user negotiates with owner to place the order for resources through the order window. These negotiations may include asking about the earliest availability of the resource with its cost etc. If the potential client is not satisfied with the outcome of the negotiations, it may simply conclude negotiations, or it may actually place an order. The resources are delivered through the pickup window from the owner to the client [2]. To understand this model take an example of restaurant in which different windows are working for placing the order like for Chinese food a different window, for Punjabi food a different window and they all have individual resources to complete their order. When order gets ready signal is given to the client to take their order. Same procedure is done in this model to get the resource. This model may remove some of the limitations of the hierarchical model but it also has some limitations.

D. Agent Based Model

To effectively schedule applications that require grid resources, the agent system bridges the gap between Grid users and resources. PACE is used [4] to provide predictive data on the performance of applications running on high-performance local resources, with the grid users being provided with relevant application tools. An application request is submitted with a corresponding application model developed using the tools. At the same time, PACE resource tools are embedded in each grid resource to provide an appropriate resource model that is an important aspect of the resource's service information. Multiprocessor scheduling system uses the evaluation engine, which acts as a local grid resource manager. In the implementation of the system, the A4 methodology [4] is used at the meta-level. Each agent acts as a representative at the meta-level for a particular scheduler and sees the local grid resource as its ability to deliver service. A communication layer, a coordination layer and a local management layer are included in the agent structure. Agents cooperate with each other and perform advertising, discovery and delivery functions to schedule applications requiring the use of available resources.

E. Trust Aware Model

The Grid system as a whole is split into Grid Domains (GDs) in Trust Aware Model. The GDs are autonomous administrative entities that consist of a collection of resources and clients that are managed by a single administrative authority. It is easy to address problems such as scalability, site autonomy, and heterogeneity by organizing a Grid as a set of GDs. It combines two virtual domains with each GD:

- A Resource Domain (RD) to indicate resources within the GD and
- A Client Domain (CD) to indicate clients within the GD.

Since RDs and CDs are mapped to GDs by virtual domains, some instances of RDs and CDs can map to the same GD. An RD has the following characteristics appropriate to the TRMS (Trust aware Resource Management System):

- Ownership
- Set of Type of Activity (ToA) it supports
- Trust Level (TL) for each ToA

The ToAs set identifies the functionalities supplied by the resources that are component of the RD [5]. Some examples of operations that a job may participate in on an RD include printing, data storage, and display services. Associating a TL with each ToA gives clients the flexibility to selectively open services.

F. Agreement based Model

A resource consumer requires comprehending and controlling resource behavior in order to influence resource use, often requiring assurances or guarantees about the level and type of service the resource provides. The owner, on the other hand, intends to keep local control over how the resource can be used and how much service information is exposed to the resource consumer [6]. A popular way to reconcile these two conflicting requirements is to negotiate an agreement referred to as a Service Level Agreement (SLA) whereby a resource provider "contracts" with a client to provide some measurable capability. Agreements explicitly state the conditions between a resource user and resource provider that enable customers to know what to expect from resources. An agreement can be seen as more than just a declaration of performance conditions. Rather, an agreement can be seen as a declaration of common terms of policy to be respected by the agreement's provider and consumer. As such, an arrangement offers a strong virtualization mechanism or resource abstraction mechanism.

G. Novel Agent Based Model

It is constructed logically as a two-layered hierarchical Heap Sort Tree (HST) to make a balanced and effective grid computing system for job scheduling. Each node of the grid computing system has two types of agents deployed on it in this model. The two types of agents are: Autonomy Representation Agent (ARA) and Node State Monitoring Agent (NSMA), both commonly known as GRMA (Globus Resource Allocation Manager) [7]. The ARA agent is triggered only on the very node of an autonomy whose present computational capacity is the largest of all autonomy nodes. All ARA agents should construct themselves collaboratively into one top layer of HST, namely Top-HST, and the root node of the built Top-HST is the very ARA whose available computational capability is the largest in the entire grid computing system.

H. CCRS (Centrally Coordinated Resource Sharing) Model

In a virtual organization, if there is a third party acting as coordinator, just like the double auctioneer [8], this sharing relationship is called Centrally Coordinated Resource Sharing (CCRS), it is the responsibility of the coordinator agent to determine the resource consumption allocation policy under this sharing mode. Coordinator agent must retain certain significant information within the organization, such as the function and status of participating agents and sharing policies and regulations, etc. The coordinator's function is not unchangeable, so other officials can compete in winning this part.

I. ICRS (Individually Coordinated Resource Sharing) Model

If there is no center coordinator in a virtual organization, such as an economic market negotiating model, resource providers and consumers can negotiate contracts without third party interference, this sharing relationship is called Individually Coordinated Resource Sharing (ICRS). Under this sharing mode, the resource consumption allocation policy is implemented through distributed decision making [8].

J. FSRS (Fair Share Resource Sharing) Model

Another sharing relationship needs the autonomous agent to have double roles in virtualization, i.e. the agent can be both resource provider and resource consumer, similar to the sharing relationship between P2P. Under this mode of sharing, each agent can use resources only when they contribute resources and the greater the contribution they make [1].

III. PROPOSED INTEGRATED MODEL

In Grid computing resources are geographically distributed in large areas and every resource has its own policies, rules, terms and conditions according to which they charge from users in terms of cost or they can share their resource by fair share policy in other words give and take. The owner's choice matters a lot more than that of clients. There is a need of more number of choices for clients so that they can actually get what they want. In this model clients get full rights to choose according to their requirements and budget. The architecture and working of the proposed Hierarchical and Economy based Resource Management 'Integrated Model' is discussed in the following sections.

A. Architecture of the Proposed Model

The basic architecture of Economical and Hierarchical based Resource Management Model ("Integrated Model") is shown in Figure 1.

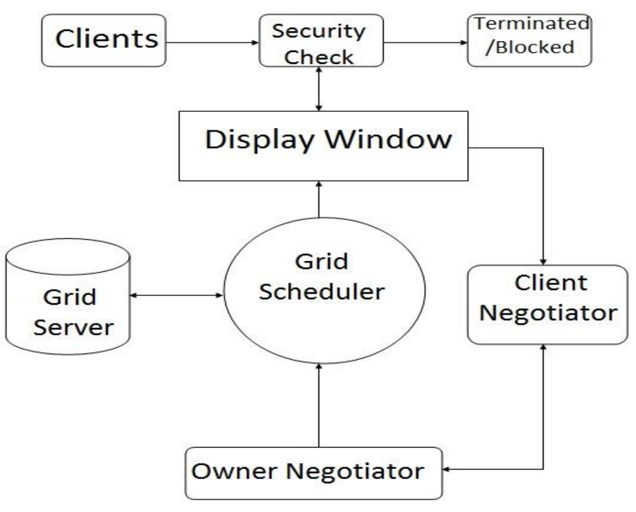

Figure 1: An Integrated Resource Management Model

The description of the components used in this model is as follow:

1) Clients/ User & Security Check

User can be an organization or a client of particular organization which is a part of virtual organization. Users submit their jobs for execution to the Resource Management System. A job may be web based, network based, local machine, remote machine, or any specific task to be performed. For example, downloading a movie from shared downloading server or converting and splitting a video can be a submitted job. Security Check provides the facility to authenticate the users in the model before they can access the resources. Non authentic users are not allowed any access and they are blocked and their request is dropped/ terminated. Authentication is done with help of CA certificates. We also have a component 'Queue'. A queue is a specific type of collection where the jobs are held in order and the main activities on the collection are adding jobs to the position of the rear terminal and removing jobs from the position of the front terminal.

2) Display Window

The display window component includes features that are type of resources, machine name, current status, and duration, resources available and cost for resources per hour. 'Type of Resources' tells what kind of resources is available on grid in other words name of resource that is available. 'Machine Name' shows at which machine a particular resource is available on Grid and whether that resource is free or reserved shown by current status. If the resources are reserved then duration feature give the time for which the resources are reserved and the time at which resources become available is given by resource available feature. 'Cost' is also shown at display window according to unit per hour. With the help of display window, user can easily get required information.

3) Grid Scheduler

This component is responsible for resource scheduling by interacting with grid information server and identifying the list of approved machines, tracking resource status information and scheduling resources according to the requirements of clients. Open source resource manager Torque PBS is the job/resource scheduler used for the management and proper distribution.

4) Grid Server

It stores information of the resources that are shared by the distributed systems of the grid environment or in other words resources that are shared within a virtual organization. In grid environment resources from multiple organizations are dynamically pooled into virtual organization to solve specific problems. So grid server store information of all the pooled resources for example in this model resource storage is used, grid information server have all details that where the data is stored, who is using that data and what is its current status. The PostGreSQL relational database is deployed for storages.

5) Owner & Client Negotiator

The Owner negotiator is negotiating with resource users and sending resource access. It is aimed at maximizing its owner's resource utility and profit in order to gain as much profit as possible. A constant check is maintained for the number of incoming requests for the allocation of the resources and simultaneously it maintains the record. Whereas Client negotiator works under the supervision of schedulers advice. It negotiates for access to resources at a low cost with different machine owners. Its aims are to give better quality of services and more resources at low cost. Various parameters are used for negotiation between the owner and clients.

B. Working of the Model

Working flow of the Hierarchal and Economical based Resource Management Model is discussed in this section.

First & Second components of the above said model are Client/User & Security Check respectively. The client submits a job to the virtual organization. User can be an organization or a client of particular organization which is a part of Virtual Organization/ Resource Management System for execution. This job may be web based, network based, local machine, remote machine, or any specific task to be performed. For example, downloading a movie from shared downloading server or converting and splitting a video can be a submitted job. Security Checker provides the facility to authenticate the users in the model before they can access the resources. Non authentic users are not allowed any access and they are blocked and their request is dropped/ terminated. Authentication is done with help of CA certificates. For scheduling of job we used TORQUE (PBS). TORQUE is an open source resource manager providing control over batch jobs and distributed computer nodes. TORQUE manages jobs that users submit to various queues on the system, each queue representing a group of resources with attributes necessary for the queue's jobs. Commonly used TORQUE tools include qsub, for job submission; qstat, for monitoring the status of jobs; and qdel, for terminating jobs prior to completion. We also have a component 'Queue'. A queue is a specific type of collection where the jobs are held in order and the main activities on the collection are adding jobs to the position of the rear terminal and removing jobs from the position of the front terminal. Queue used the FIFO for serving the jobs. MD5 encryption based algorithm is also used by grid for authentication purpose.

1) Flow of Job Submission

Node A: This is at client side. This node is running with the configuration PBS server as job management tool having CA client certificate generated by the CA server.

Node B: This is the server node. It has PBS server as the job management tool. This node has all required services in the running state. This node manages all the jobs and schedules them. Here Node A is requesting server Node B for resources reservation. For this purpose we are using Java Code. Node A requests server for the resource management.

2) Steps followed for Job Submission

- Node A requests Node B: Node B checks for its authentication. As it is mentioned above that grid allows authorized users to use all the grid services, Node B will have two certificates one for user grid of client machine, second for the root user of the client machine.
- After the authentication client checks for resources. Here when the client requests, it checks for the current grid state.

Current grid state means:

- What type of resources available on grid?
- Resources are free or reserved?
- How long resources will be freed or reserved?
- At what cost resources are available?

If it is in state of accepting and executing a job, Node B sends a message "Grid is Free" to the node A. After that the client is free to submit job. If grid is not free it sends after approximately how many minutes the grid will be free. The Figure 2 describes the flow of job submission.

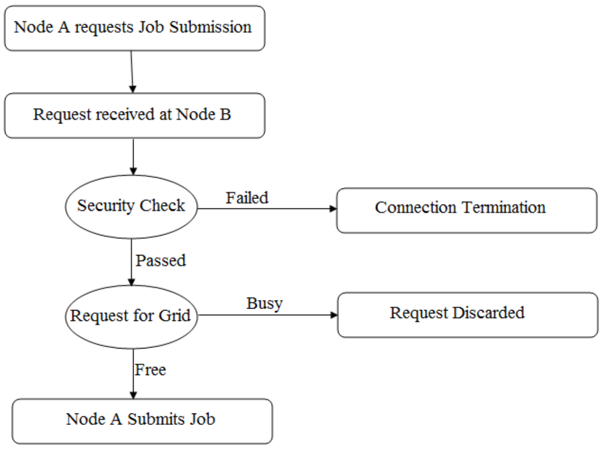

Figure 2: Flow of Job Submission

As described the job submission flow begins with authenticity and even if authenticity test is passed, the grid will only be allocated in Free State.

Third component is Display Window. After the successful security check the user makes access to display window. It includes features that are type of resources, machine name, current status, and duration, resources available and cost for resources per hour. 'Type of Resources' tells what kind of resources is available on grid in other words name of resource that is available. 'Machine Name' shows at which machine a particular resource is available on Grid and whether that resource is free or reserved shown by current status. If the resources are reserved than duration feature give the time for which the resources are reserved and the time at which resources become available is given by resource available feature. 'Cost' is also shown at display window according to unit per hour. With the help of display window, user can easily get required readings. The client check resource status whether resource is free or reserved, if reserved then job is either send to queue or job can be terminated by the client. Whenever resources become free, a signal is send to queue so that clients can claim for resources. Display window is updated by the grid scheduler time to time and all information is stored in the grid information server.

Fourth and fifth components of discussion are Grid Scheduler and Grid Server, respectively. Grid Scheduler is responsible for resource scheduling by interacting with grid information server and identifying the list of approved machines, tracking resource status information and scheduling resources according to the requirements of clients. In this model PBS is the job scheduler used for the job management. PBS, the Portable Batch System, is a networked subsystem for submitting, monitoring, and controlling a workload of batch jobs on one or more systems. With PBS, jobs can be scheduled for execution on one system according to scheduling policies that attempt to fully utilize system resources without over committing those resources, while being fair to all users. Open source resource manager Torque PBS is the job/resource scheduler used for the management and proper distribution. Whereas, Grid Server stores information of the resources that are shared by the distributed systems of the grid environment or in other words resources that are shared within a virtual organization.

In grid environment resources from several different organizations are dynamically pooled into virtual organization to solve specific problems. So grid server store information of all the pooled resources for example in this model resource storage is used, grid information server have all details that where the data is stored, who is using that data and what is its current status. The PostGreSQL relational database is deployed for storages.

Next are Onwer and Client Negotiators. After collecting data from the display window client sends request for resources to owners. Then client's negotiator and owner's negotiator communicate with each other.

3) Sequence of the interactions which take place between the two parties is:

- **Connecting:** Client sends request to different machine owners regarding the resources that client wants to communicate.
- **Connected:** If owner is interested to communicate with client then the owner sends confirmation to the client.
- **Negotiation (resource requirement, QoS, budget):** After confirmation there are certain calls that take place for negotiation between client & owner, these are:
  (a) Client sends detail of its requirements
  (b) Owner sends detail of its policies
  As shown in Figure 3, if owner and clients both agree with each other for established terms and conditions, then deal is confirmed and resources are reserved. But if they do not agree, then deal is rejected or terminated.
- **Disconnect:** Both parties terminate the connection after negotiation.

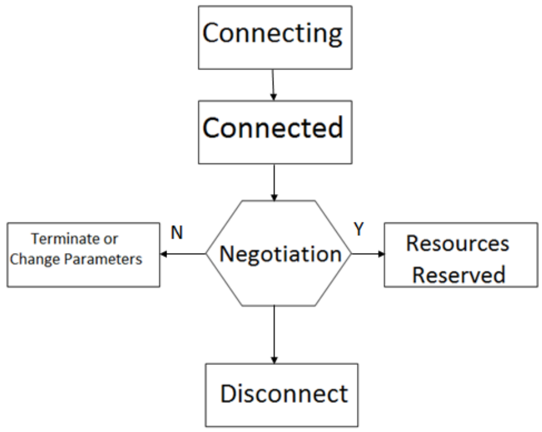

Figure 3: Negotiation Communication

The Negotiation can be done with respect to parameters

- Cost
- Job Priority
- Corporate Priority
- Task Value
- Task Time
- Feasibility

If both parties agree with established conditions, rules and regulations, then deal is final and resources are reserved for that particular time on fixed cost. When resources got reserved then owner's negotiator sends this information to the grid scheduler. Grid scheduler updates display window and stores the whole information to grid server, after that execution of job is started. If both parties do not agree, then negotiation is terminated and client is free to either negotiate with other owners or change the parameters of deal and restart negotiation with previous owner. This is the one way through which these components interact with each other.

Implementation and Results

IV. PREPARING SYSTEMS

In the Grid Environment, there are two systems (one client and one server). The configurations of the systems are mentioned below:

- **Grid client (Node A):** Connecting to a grid that appears as a "cluster" of Torque (PBS) job managed machines represented by Node B.
- **CA Server (Node B):** To authenticate the grid hosts and user to use the grid resources.

Following steps have taken to prepare systems for the installation of the grid:

1) Two machines running CentOS i386 32 bit are required. After the installing Linux configure the yum so that different required packages can be installed.
2) After the installation of Linux, update the systems to get all latest update packages installed. After this install gcc compiler over all the nodes by running the following command.
   [root@nodeA]# yum install gcc
   Configuring NTP Server: To synchronize the Grid node to a uniform time standard, a NTP (Network Time Protocol) server is required. Each machine should be running NTP client service to synchronization Grid machine.
3) Configuring Firewall & SELinux: Here project Firewall & SELinux are disabled.
4) Configuring /etc/hosts file: Each machine must have following /etc/hosts file as below:
   [root@nodeA~]#cat/etc/hosts
   # do not remove the following line, as various programs # that require network functionality will fail. 127.0.0.1 localhost.localdomain localhost
   192.168.9.40nodeA.linux.com node A
   192.168.9.39nodeB. linux.com node B
   Root access to all the machines is necessary in order to complete some of the installation and for other administrative tasks.
5) Make sure the proper user accounts are set up:
   - Root Account: Administration, installation, updating and removing to software packages
   - Globus account: Administration, installation, updating and removing to Globus toolkit 4.0.1 source package with CA updates.
   - Grid user account :User of Grid system.

It can be done by running following command:-
[root@nodeA]# adduser globus
[root@nodeA]# adduser grid
Next assign the passwords to them by using passwd command.

V. RESULTS

Node A: This is client side node. This node is running with the configuration PBS server as job management tool having CA client certificate generated by the CA server.
Node B: This is the server node. It has PBS server as the job management tool. This node has all required services in the running state. This node manages all the jobs and schedules them.
Here Node A is requesting server Node B for resources reservation. For this purpose using Java Code Node A request server for the resource management. Node A when requests the Node B, Node B checks for its authentication. As Grid is the authorized user for the client to use all the Grid services. Means Grid is the user who has CA client side certificate at the client machine. Node B has two certificates one for user grid of client machine, second for the root user of the client machine.
After the authentication client checks for resources. Here when the client requests, it checks for the current grid state. If it is in state of accepting and executing a job. Node B send a message "Grid is Free" to the node A. After that the client is free to submit job. If grid is not free it sends after approximately how many minutes the Grid will be free.
With respect to the flow diagram in Figure 2, if a user is authenticated by CA server only then user can made a request for the resources allocation. Parameters related to results are explained as:

- Response by Server: It is the status sent by the CA server after the process of authentication. If it is free it will show status free else busy.
- Finishing time of running job: It is the time to be taken by the Grid to finish the current job when Grid is showing busy status.
- Cost: it is the per unit price of Grid multiplied by the request completion time.
- Time for request completion: it is the time taken in authentication process and time taken to execute a job.

Table- I: Allocation of resources in IRMM for Grid computing.

Request Time of Client	Response by Server	Finishing Time of Running Job	Per Unit Price of Grid	Time for Request Completion	Cost
11.30 am	Busy	45 Minutes have to wait			
2.00 pm	Free	None	20	32 Min	640
2.20 pm	Busy	12 minutes			
3.00 pm	Free		20	44 Min	880
3.15 pm	Busy				
3.30 pm	Busy				
6.00 pm	Free		20	35 Min	700

It is clear from the Table I that Only CA server is doing allocation of the resources. If any node gets free while others are busy, CA will still show the status busy. Grid can be reserved only if all the nodes are free.

VI. CONCLUSION

The paper presented a Resource Management Model for Grid Computing. The Proposed Model is titled as "Integrated Model", which is combination of Hierarchical and Economic based Resource Management Model for Grid Computing. After studying the different models, it is very clear that mostly administrative have all the rights and user has nothing in its side, which means owners enforce their policies on the users. Only economical model gives rights to the users but it also has some limitations. So there is a need of a new architectural model for resource management in grid computing which is based on characteristics of both the Hierarchical and Economical Model.

REFERENCES

1. R. Buyya, D. Abramson, and J. Giddy, "Nimrod-G: Architecture for a Resource Management and Scheduling Systems in a Global Computational Grid", The 4th International Conference on High Performance Computing in Asia-Pacific Region, 2000.
2. Rajkumar Buyya, Steve Chapin, and David DiNucci, "Architectural Models for Resource Management in the Grid", Proceedings of the First IEEE/ACM International Workshop on Grid Computing, 2000.
3. Rajkumar Buyya, David Abramson and Jon Giddy, Economy Driven Resource Management Architecture for Computational Power Grids, Intl. Conf. on Parallel and Distributed Processing Techniques and Applications (PDPTA 2000), USA.
4. J.Cao, D.J. Kerbyson, and G.R. Nudd, "Performance Evaluation of an Agent-Based Resource ManagementInfrastructure for Grid Computing", in Proc. of 1st IEEE Int. Symp. On Cluster Computing and the Grid, 311-318, 2001.
5. Farag Azzedin, Muthucumaru Maheswaran, "Towards Trust-Aware Resource Management in grid computing systems" Proceedings of the 2nd IEEE/ACM international symposium on cluster computing and the grid, 2002.
6. Czajkowski, K. Foster, I. Kesselman, Agreement-Based Resource Management Proceedings of the IEEE Volume 93, Issue 3, March 2005.
7. Mark Baker, Rajkumar Buyya, Domenico Laforenza, The Grid: International Efforts in Global Computing, Intl. Conference on Advances in Infrastructure for Electronic Business, Science, and Education on the Internet, Italy, 2000.
8. Yang Liu; De-xian Zhang; Jian-ping Fan; Multi-workloads sharing oriented resource management model for grids Computing, Communication, Control, and Management, 2009. CCCM 2009. ISECS International Colloquium.

AUTHORS PROFILE

Makhan Singh graduated in 2000 with a degree in computer science and engineering from Punjab Technical University, Jalandhar, India and a Postgraduate degree in computer science from Indore University in 2002.His research interests include Privacy and Storage Management in Distributed Systems and Information dispersal in Cloud Computing.