

A Multi-Commodity Grid Logistics Model for Data Transportation in Federated Datagrid

Muralikrishnan Ramane
Department of Computer Science and
Engineering, Pondicherry Engineering College,
Puducherry, India

F Sagayaraj Francis
Department of Computer Science and
Engineering, Pondicherry Engineering College,
Puducherry, India

ABSTRACT

Data Sharing is the process of making data resources available across geographically distributed institutions working on a common purpose. When there is need or request, these data resources are transported so as to co-allocate them at desired destinations. Distributed computing, particularly Grid computing environment involves transporting huge volume of data across geographically spread sources and destinations managed by an efficient data management architecture called the Datagrid. This affinity towards movement of data is to have the data and task collocated to eliminate network traffic and to improve performance by reducing latency. Participants of the datagrid are autonomous organizations, organized in a federated fashion that can request for resources from the federation as long as they are authentic and authorized and can fall into or fall out of the grid at their own will. Data Transportation mechanism in grid can be compared to the Multi-index transportation problem i.e. an n-dimensional problem (here $n = 3$) and usually referred to as multi-commodity transportation problem with an additional index for handling multiple resource types. This work proposes a novel method to transport resources taking various challenges posed by the dynamic grid environment into account.

General Terms

Distributed Computer Systems

Keywords

Data Sharing, Data Resource Allocation, Grid Data Management, Grid Uncertainty.

1. INTRODUCTION

A Grid [1, 2, 3] can be defined as a large-scale geographically distributed hardware and software infrastructure composed of heterogeneous networked resources owned and shared by multiple administrative organizations which are coordinated to provide transparent, dependable, pervasive and consistent computing support to a wide range of applications. It enables distributed supercomputing, on-demand availability of data and collaborative computing.

Grid computing involves large number of computers, arranged as clusters and connected via networks controlled using efficient task management techniques ranging from trivial to complex problems too large for a single server or even super computer to handle. Grid computing [3] uses a process known as cycle stealing (i.e.) grabbing spare compute cycles on machines across

networks, when available, to get a task done. Grids assume a virtual pool of resources rather than computational nodes.

The Grid concept [3] has a basic underlying problem of coordinated resource sharing in dynamic, multi-institutional virtual organizations. Access to computers, software, data, and other resources, requires collaborative problem-solving and resource-brokering strategies with resource providers and consumers defining clearly and carefully just what is shared, who is allowed to share, and the conditions under which sharing occurs. A set of individuals and/or institutions defined by such sharing rules form a virtual organization (VO). VOs enable disparate groups of organizations and/or individuals to share resources in a controlled fashion, so that members may collaborate to achieve a shared goal.

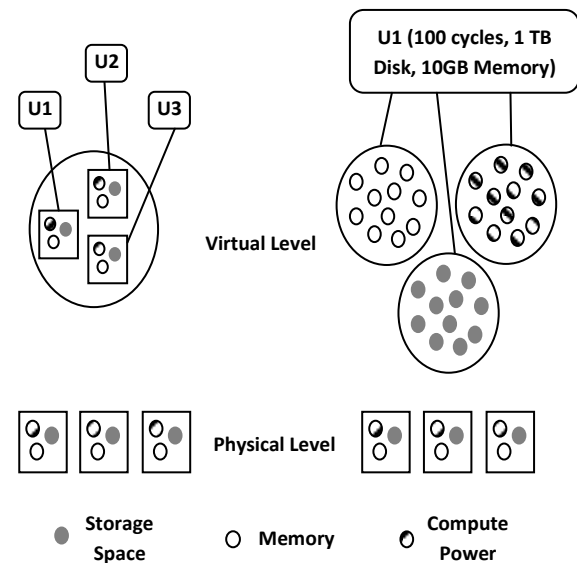


Figure 1. Conventional distributed environments vs. Grid

Conventional distributed systems [4] are based on resource ownership while grids are aimed at resource sharing. As shown in Fig.1 the grid at virtual level forms pool of resources contrary to conventional systems that try to first find an appropriate node to map the process to, and then satisfy the resource needs locally, grids are based on the assumption of an abundant and common pool of resources. Thus, first the resources are selected and then the mapping is done according to the resource selection.

In grids, the virtual pool of resources is dynamic and uncertain [5]. These resources can be added and withdrawn at any time at their owner's discretion, and their performance or load can change frequently over time. Due to all these reasons the user has very little or no apriori knowledge about the actual type, state and features of the resources constituting the pool.

The communities of researchers that need to access and analyze the data are often large and almost always geographically distributed. This combination of large dataset size, geographic distribution of users & resources and computationally intensive analysis results in stringent performance demands. The compute grid is merely half of any grid environment. To view grid in its entirety, it is essential to consider data management within the grid provided by the so called data grid.

Datagrid provides a platform [6] through which users can access aggregated computational, storage and networking resources to execute their data-intensive applications on remote data. It promotes a rich environment for users to analyze data, share the results with their collaborators and maintain state information about the data seamlessly across institutional and geographical boundaries. Datagrids are manifested in different ways in different systems based upon the source of data, whether single or distributed, the size of data and the mode of sharing. Data Grids are being adopted widely for sharing data and collaboratively managing and executing large-scale scientific applications that process large datasets distributed around the world.

The Federation model [7, 8] is prevalent in Data Grids created by organizations that wish to share data in already existing storage resources as shown in Fig. 2.

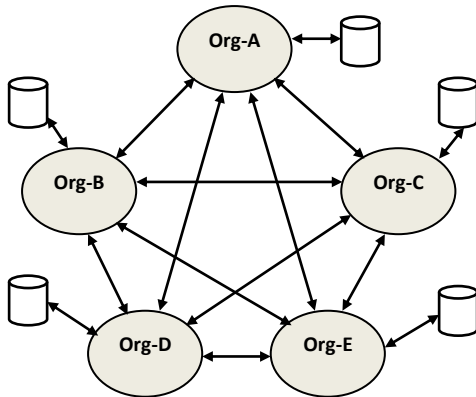


Figure 2. Resource Sharing in Federated Environment

Researchers at participating organization can request data from the federation as long as they have the proper authentication. Each institution retains control over its storage resource. Federation mechanisms are also required for data sharing between logical name spaces, for scalability of logical name spaces and for performance optimization in distributed environments.

Data movement within the grid [9] over and over again becomes too expensive than to spare some compute cycles by co-allocating data and running the task locally. The affinity towards data can be achieved by both compute grid (tasks are routed to the data) and datagrid (migrating data to the nodes). Data

migration will yield higher data affinity levels, thus minimizing network traffic and increasing the performance of the grid. The data grid can be operated in both proactive and reactive ways to achieve data affinity. This work clearly focuses on the reactive strategy by making real-time adjustments to data migration by routing the data to nodes where the data does not exist.

This paper proposes a cost model for virtual resource allocation in grid environment. The rest of this paper is structured as follows. Section 2 summarizes the related work. Section 3 discusses multi-commodity logistics model. Section 4 provides experimental results for uncertainty. Section 5 concludes this paper and suggests future work.

2. RELATED WORK

In a world-wide computational Grid, users typically want their jobs to be executed as fast as possible, whereas the goal of a Grid infrastructure is to assure specific quality of service for all participants. In order to reconcile these apparently contrasting goals, methods have been proposed based on economy strategies for efficient access and distribution of data replicas in a datagrid. Some of them are briefly discussed.

2.1 Economy-Based Optimization in a Datagrid

In this strategy [10] data-seeking agents negotiate with data-storing agents for optimum prices. Data-storing agents store the most useful files at their respective Grid sites. In another strategy, Optimization agents [11] such as Access Mediator, Peer-Peer Mediator and Storage Broker [12] are used. Optimization agents perform Replica Initiation, Selection and Replacement to make optimal decisions. Using economic models helps take calculated decisions on a dynamic grid where the availability of resources can change without forewarning.

2.2 Agent Based Optimization Using Auction Protocol

The goal of the auction protocol [11] is to select the cheapest replica of a file needed by a job running on a computing element. In other words, the protocol is used by optimization agents to perform the replica selection task. A procurement Vickrey auction is used, where agents are interested in purchasing the lowest-cost available resource.

2.3 Cost Minimized Virtual Resource Allocation in Next Gen Networks

In this method [13] Transportation problem is adapted to manage distributed resources in the next generation network infrastructure. This scheme automatically adjusts the virtual resource allocation to minimize costs and uses Inventory Control to predict future virtual resource needs in order to maintain a stock of the required resources. Simulated results show significant improvement in virtual resource management.

3. MULTI - COMMODITY GRID LOGISTICS MODEL

To perform any computational task such as analysis of remote data, the data resources must be co-allocated to the participants by aggregating required datasets from multiple sources and

transporting it to the destination or requesting participant. Efficient data management architecture, namely the Datagrid, manages these distributed resources in federated fashion. Participants can query the federation with proper credentials using an Access Mediator.

Access Mediator is responsible for querying the federation and finding the physical location of the resource using a Metadata Catalog. The Metadata catalog [14, 15] maintains global namespaces to enable transparency. Access Mediators are present at every participating node.

This work focuses on the federated model of the data grid. When participants request the federation, the mediator interprets the request and identifies the requirement. Then, it gathers information on data resources from the metadata catalog and queries the appropriate nodes. Finally, the mediator forms a sharing strategy with the volunteering participants and the data transportation process begins by selecting the resource with minimum cost. The resource is then transported to the destination. The focus of this work is to transport resources of different sizes from multiple sources to multiple destinations.

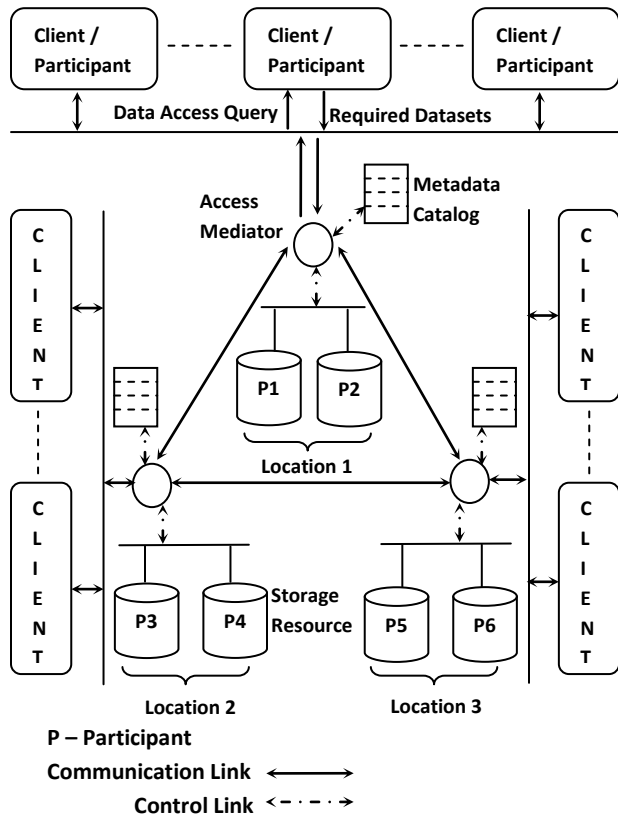


Figure 3. Data Sharing Model

The challenge in this setup is to transport data resources from multiple sources to destinations at minimum cost. This work adapts a special case of the Transportation Problem i.e. the Multi-Index Transportation problem to find a solution.

The data sharing model is designed as shown below in Figure 3. The participants form a federated model and Access mediators are available at every participating node. The participants requiring a resource act temporarily as clients and make the

requests. In this multi-source, multi-destination model where the requestor acts as the destination and the donor acts as the source.

The Transportation problem [16] is a linear programming problem to minimize total shipping cost of shipping a commodity from sources (e.g., factories) to destinations (e.g., warehouses) satisfying both the supply limits and the demand requirements. It assumes that the shipping cost on a given route is directly proportional to the number of units shipped on that route. It is therefore concerned only about finding an optimal distribution plan for a single commodity and does not consider multiple data items and their properties.

On the contrary a DataGrid contains multiple types of data resources each having its own set of properties. Hence transportation cost varies with each data item. This work tries to address this concern by adapting a multi-commodity [17, 18, 19] based approach. In this work, the virtual resources are the commodities to be transported.

3.1 Transportation Cost Model

The three dimensional cost model as shown in Fig 4. is designed with three indices each representing a dimension. The model consists of m number of Sources, n number of Destinations and r number of Resources

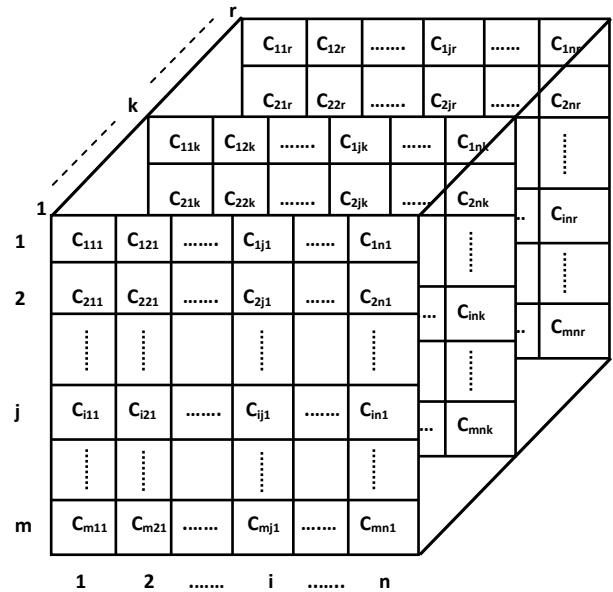


Figure 4. Transportation Cost Model

The cost model is constructed based on unit cost of transporting a resource of type k from source i to destination j denoted by C_{ijk} . An allocation model is designed similar to the above model which represents the number of resource units of type k transported from source i to destination j denoted by X_{ijk} .

The total cost of the transportation can be calculated by using the formula:

$$\sum_{i=1}^{PS} \sum_{j=1}^{PD} \sum_{k=1}^{PR} C_{ijk} \cdot X_{ijk} \dots\dots\dots(1)$$

An allocation strategy is formed based on the cost model for cost optimized distribution. Every participant shares a set of resource and can request the same to the grid. The cost of transporting a resource is determined as the product of transporting unit resource of type k and number of units of type k between i and j .

The fundamental model was devised for transporting hardware commodities whereas data resources in grid are of software kind. The cost model works based on the following assumptions:

Let

P be the total number of participants available in the grid

R be the total number of resources available in the grid

T be the total number of resource types available in the grid

PS be the set of source participants sharing the resource

PD be the set of destination participants requesting the resource

PR be the set of resource types participating in the grid

Each participant P_x has a set of resources RS_x

RS_x is the subset of R

RS_i be the supply at PS_i

RS_j be the demand at PD_j

Each set RS_x contains resource of type k

RS_{xk} be the k^{th} type resource available at P_x

RS_{xk} is the subset of RS_x

RS_{ki} be the k^{th} type resource shared by PS_i

RS_{jk} be the k^{th} type resource requested by PD_j

RS_{ijk} be the total number of resource types transported between PS_i and PD_j .

C_{ijk} be the cost of transporting unit k^{th} type resource from PS_i to PD_j

X_{ijk} be the allocation type for k^{th} type resource transported from PS_i to PD_j

E_{ij} be the total number of resources transported from PS_i to PD_j

In the process of adaptation this work has brought in significant modifications suitable for grid environment.

a. Allocation Units

Hardware commodities being shipped must be produced in multiple amounts based on the demand at warehouses i.e. if a product p has demand of 100 units then it must be produced physically at the source and then transported, whereas data resources in grid being software in nature need to transport only a copy of the required resource. In this work the allocation model manipulates only 0's or 1's representing the availability of the resource.

b. Source can be a Destination

In the hardware model a source (factory) will be the place of production and cannot intake any product. The destination being a warehouse cannot produce a product, but in a grid the source

(participant) can be the donor and also be the requestor of a different resource. So at any instant a participant can request for a set of resource as well as sharing its own.

The following constraints must be satisfied for the model to form an optimal distribution plan.

1. Availability Constraint

$$X_{ijk} = \begin{cases} 0 & \text{not available} \\ 1 & \text{available} \end{cases} \dots\dots\dots (2)$$

2. The total number of resource types transported from the source participant should be less than or equal to its supply.

$$\sum_{k=1}^{PR} RS_{ki} \leq RS_i \dots\dots\dots (3)$$

Where, $i=1,2,..PS$

3. The total number of resource types transported to the destination participant should be greater than or equal to its demand.

$$\sum_{k=1}^{PR} RS_{jk} \geq RS_j \dots\dots\dots (4)$$

Where, $j=1,2,..PD$

4. The total number of resource types transported from source i to destination j should be less than or equal to total number of resources transported.

$$\sum_{k=1}^{PR} RS_{ijk} \leq E_{ij} \dots\dots\dots (5)$$

Where, $i=1,2,..PS; j=1,2,..PD$

The algorithm for finding minimum cost resource is given below in figure 5: Nearest Supplier Algorithm

Step1: Select a resource of type k from cost matrix.

Step2: Select a destination participant requesting the resource of type k .

Step3: Find the nearest or least cost source participant sharing the resource.

Step4: Allocate the resource i.e. set value 1 to the cell and assign value 0(demand satisfied) to corresponding destination participant.

Step5: Iterate steps 3 and 4 for all available destinations.

Step6: Iterate steps 2 to 5 for all available resource types.

Step7: Total transportation cost is calculated using Eq(1).

Figure 5. Algorithm for finding minimum cost resource (nearest supplier)

4. EXPERIMENTAL RESULTS

To understand variations in the model behavior due to uncertainty in grid a set of 4 experiments were conducted for each of the two cases. In the first case the model was assumed to be static i.e. a participant leaving the grid does not inform the co-volunteers. In the second case the model was assumed to be dynamic where the participant informs its failure or load to its co-volunteers before leaving the grid. The following experiments were conducted and outcome of each experiment is shown in individual graphs. The experiments were done on an Intel Core 2 Duo Processor T6600 machine with 3GB of RAM on Windows 7 operating system. A database of 2 million records was created which represents the resource transfer between sources and destinations and their cost.

The experiment for uncertainty was conducted by removing the sources individually and in sets which leads to reduction in supply levels. The model was analyzed to find how far it is functional on the event of random failure based on various replication levels (number of participants a resource is replicated).

Table 1. Experiment Setup

No. of Participants	50
No. of Resources	5000
No. of Participants a Resource is Replicated (Replication Level)	Exp1: 0-10% Exp 2: 0-20% Exp 3: 0-30% Exp 4: 0-40%
Cost Range	1-1000 (Geographic Proximity of the node)

Case 1: Test for cost variations with increase in uncertainty (participant failure) – Static approach

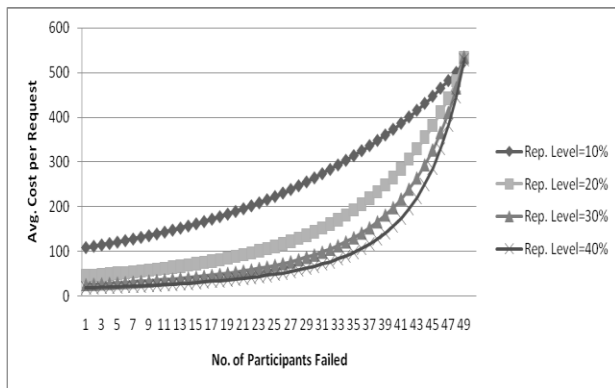


Figure 6. No. of Participants Failed Vs. ACPR (Average Cost per Request)

Case 2: Test for cost variations with increase in uncertainty (participant failure) – Dynamic approach

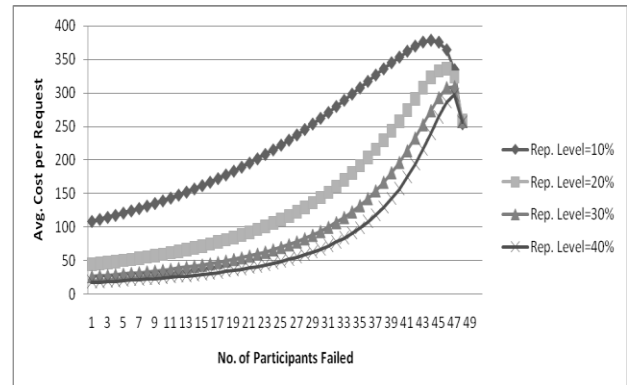


Figure 7. No. of Participants Failed Vs. ACPR (Average Cost per Request)

4.1 Inference

Case 1:

Figure 6 shows the fall in average cost for increase in replication levels. In this case the average cost per request increases with increase in participant failure but at the same time, if the number of sources failed exceeds a threshold the model becomes dysfunctional since it cannot operate under such minuscule supply of resources. The model is operational until $p-1$ participants fail. The cost of operation increases rapidly since participant failure is not notified thereby satisfying unnecessary demands of non-existing participants. On analyzing the model from the charted values the average cost per request increases with increase in participant failure and with increase in replication levels the difference in average cost per request is minimized.

Case 2:

Figure 7 shows the fall in average cost for increase in replication levels which is similar to the previous case but here the average cost per request increases gradually with participant failure which shows the better performance of the dynamic approach. In the dynamic approach since the participants inform the failure, both the request (demand) and response (supply) of the particular participant is made invalid. Here the model stops its operation when $p-1$ participants fail because no resource transfers occur with one participant available in the federation. When there is a decrease in replication level, the number of supply units is reduced thereby forcing the model to cater to the same level of requirements with reduced availability. Hence the model underperforms on such cases.

At the same time higher replication levels does not mean the model perform efficiently because with increase in replication levels the overhead of managing the data leads to inefficiency. So a balance in cost and replication level is what this experiment tries to achieve. It is evident from the charted values there is no major cost differences at level 30% and 40% and further increase in replication will even minimize the difference. So instead of analyzing higher replication levels this work

concludes that with 40% replication the model would perform efficiently.

On analyzing both the cases the performance of the model improves in the later (dynamic) approach, i.e. the average cost per request and maximum cost of operation for the dynamic approach is optimum and far lesser than the static model.

5. CONCLUSION AND FUTURE WORK

This paper provides an allocation model for optimal distribution of data resources in the grid. The behavior of the model is studied using suitable experiments when resource availability is uneven. The results have been illustrated using graphs showing the average cost per request for various scenarios. It can be inferred from the results that uncertainty is the key challenge to a grid setup. Addressing the issue of uncertainty can be either proactive or reactive and it necessitates us to study about the sensitivity of the model which is currently under our experimentation. This model can also be extended by considering type of transport as the fourth dimension.

6. ACKNOWLEDGEMENTS

The authors would like to thank all those who have shared their valuable inputs, especially **Gowri Shankar Ramaswamy**, Department of Computer Science and Engineering, Pondicherry Engineering College, Puducherry, India for his insights, suggestions and time throughout the course of this work.

7. REFERENCES

- [1] Miguel L. Bote-Lorenzo, Yannis A. Dimitriadis. Eduardo Gómez-Sánchez. Grid Characteristics and Uses: A Grid Definition. *First European Across Grids Conference*, Santiago de Compostela, Spain, February 13-14, 2004.
- [2] I. Foster, "What is the Grid? A Three Point Checklist", *Grid Today*, Vol. 1, No. 6, 22 July 2002. <http://www.gridtoday.com/02/0722/100136.html>.
- [3] I. Foster, C. Kesselman, and S. Tuecke. "The anatomy of the Grid." *The International Journal of Supercomputer Applications*, pp.1-25, 2001.
- [4] Z. Nemeth and V. Sunderam. "A comparison of conventional distributed computing environments and computational grids." In *International Conference on Computational Science (ICCS)*, Amsterdam, 2002, pp.729-738.
- [5] Erwin Laure, Heinz Stockinger, and Kurt Stockinger Performance Engineering in Data Grids." *Concurrency and Computation: Practice & Experience*, pp.1-20, 2005.
- [6] Chervenak .A, Foster .I, Kesselman .C. "The Data Grid: Towards architecture for the distributed management an analysis of large scientific datasets." *The Journal of Network and Computer applications*, pp.1-10, 2000.
- [7] S. Venugopal, R. Buyya, and K. Ramamohanarao, "A Taxonomy of Data Grids for Distributed Data Sharing Management, and Processing." *ACM Computing Surveys*, Vol.38, ACM Press, NY, USA, pp.1-60, 2006.
- [8] Rajasekar. A, Moore. R and Schroeder. W. "Data Grid Federation." In *Proceedings of the 11th International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA 2004)* CSREA Press, Las Vegas, USA, 2004, pp.1-6.
- [9] Michael Di Stefano. "Distributed Data Management for Grid Computing." Wiley Interscience, July 2005, pp.1-275.
- [10] M. Carman, F. Zini, L. Serafini, and K. Stockinger. Toward an economy-based optimization of file access and replication on a data grid. In *Workshop on Agent based Cluster and Grid Computing at International Symposium on Cluster Computing and the Grid (CCGrid 2002)* (Berlin, Germany, May 2002), IEEE-CS Press, pp. 340–345.
- [11] D.G. Cameron, C. Nicholson, K. Stockinger. "Formal analysis of an agent-based optimization strategy for Data Grids." *The Multiagent and Grid Systems International Journal*, pp.1-18, 2004.
- [12] Baru. C, Moore. R. Rajasekar. A, and Wan. M. "The SDSC Storage Resource Broker." In *Proceedings of CASCON'98*. IBM Press, Toronto, Canada, 1998, pp.1-11.
- [13] R Farha, A Leon-Garcia. "Operations Research methods for autonomic resource management." *The Journal of Network and Systems Management- Springer*, pp.158-182, 2009.
- [14] Lichun Zhu. "Metadata Management in Grid Database Federation." *ACM Transactions on Database Systems*, pp.3- 21, 2007.
- [15] DCMI Initiative. "Dublin Core Metadata Element Set, <http://dublincore.org/documents/2010/10/11/dces/>, Oct. 11, 2010 [Nov. 30, 2010].
- [16] R. Panneerselvam, "Operations Research." 2nd Edition, Prentice Hall, 2002, pp.63-119.
- [17] JK Sharma, "Extensions and special cases of transportation problem: A survey." *Digital library of India*, pp. 928-940, 1977.
- [18] Haley, K. B. "The multi-index problem, "Operations Research, *JSTOR*, pp. 368-379, 1963.
- [19] Z. Lelkes, T. Kovacs, I. Jones, "Multi commodity transportation and supply problem with stepwise constant cost function," *European Symposium on Computer Aided Process Engineering*, Elsevier, 2005, pp. 1069-1074.