Economy-Based Data Replication Broker

Henry Lin[1], J. H. Abawajy[2], Rajkumar Buyya[1]
[1]Grid Computing and Distributed Systems Lab
Dept. of Computer Science and Software Eng.
The University of Melbourne
Parkville, Melbourne, VIC 3010, Australia
[2]Grid and Pervasive Computing Lab
School of Eng. and Information Technology
Deakin University
Geelong, VIC, 3072, Australia

Abstract

Data replication is one of the key components in data grid architecture as it enhances data access and reliability and minimises the cost of data transmission. In this paper, we address the problem of reducing the overheads of the replication mechanisms that drive the data management components of a data grid. We propose an approach that extends the resource broker with policies that factor in user quality of service as well as service costs when replicating and transferring data. A realistic model of the data grid was created to simulate and explore the performance of the proposed policy. The policy displayed an effective means of improving the performance of the grid network traffic and is indicated by the improvement of speed and cost of transfers by brokers.

1. Introduction

E-Science refers to large scale scientific endeavors that are increasingly carried out via collaborations of global scale [1]. Application domains that are increasingly using this paradigm to solve big scientific problems are high energy physics, molecular chemistry and astrophysics. Typically, such scientific collaborations [3][4][5] generate massive quantities of data and require access to very large data collections and computing resources. This means, e-Science requires efficient and fair access to data, which in turn requires improved solutions in dealing with large volume transfers, storage management, data consistency and wide-area cyber-infrastructure [20]. To address this requirements, a distributed data management [16] infrastructure called data grids have emerged [2] [6].

One of the essential optimization services of a data grid is the data replication service. This service seeks to improve the network traffic by copying heavily accessed data to appropriate locations and managing their disk usage [8]. Moreover, a storage failure at the original node could be recovered by the replicas and transcontinental transfers can be reduced by creating localized sets of replicas. However, in order to realize the potential of the replication and the data grid itself, there is a requirement for a decentralized replication management systems. One of the methods proposed to manage grid resources have been to employ market forces analogous to the real world [9]. In economy resource management grids, resources are sold to users as a service. When individual players in the node make decisions based on economic rationality, economic theory indicates that the market economy will iterate towards a balanced and sustainable system of service demand and supply [10].

This paper will detail improvements to data replication schemes for a data grid that utilizes a commodity market economy for the purposes of resource management. When economic resource management models are applied to replication management, it translates the question of where (and when to some degree) to replicate data into the sub-problems of costing and valuation of services. The medium for valuation and purchasing services can be real currency or it can be abstract grid tokens or credits as long as they can be redeemed for future services. The act of purchasing of services or transactions can be mediated through a market [19]. The interaction between the mediator and players involved in the transaction are determined by a particular economic model with its associated transaction protocol. These models will specify the process in determining the price for the use of grid resource(s), and transaction protocols, methods in accounting, billing and other analogous fiscal services [17].

Our contribution is the design and development of an economic-based data replication broker that allows a broker or a user using the broker to transfer data with a view to reduce both the transfer time of a replica and/or the cost of the transfer. The policy allows flexibility in that users may specify the degree in which to trade between transfer time optimization versus the minimizing the cost of the transfer service. This flexibility will allow low priority applications to operate in cost saving modes while high priority applications can achieve close to maximal transfer speeds relative to network conditions. In terms of replication, this means that users can decide the high level long term strategy in replication transfers. A realistic model of the data grid was created to simulate and explore the performance of the proposed policy.

The policy displayed an effective means of improving the performance of the grid network traffic and is indicated by the improvement of speed and cost of transfers by brokers.

The rest of the paper is organized as follows. In Section 2, the background and related work are discussed. The system model and the proposed data replication broker are discussed in detail in Section 3. Also, the baseline policy used to compare the proposed policy is discussed in this section. In order to test the effectiveness of the proposed data replication broker, a model of a data grid was implemented and simulated in Section 4. Simulation results are discussed in Section 4. The conclusion and future direction are discussed in Section 5.

2. Background and Related Work

The replication mechanism attempts to determine when, where and what data to replicate across the data grid. In the context of data grid computing there are two main key decision making tools in a data replication scheme: the *replication scheduler*; that determines when and where to perform a replication over a network in order to fulfill its goals; and the *resource broker* that determines how and when to acquire grid services and resources for higher level components. Existing data replication schemes focus on one of these key decision making tools while we combine both to obtain the best results.

The problem of finding an optimal replica allocation at a given state of the data grid (i.e., an allocation that has minimum transfer cost for a given read-write pattern), has been shown to be NP-complete [22]. Hence, many heuristic data replication schemes have been proposed in the literature [7][11][12][13][14][18]. A dynamic replication scheme was proposed based on a variety Vickery auction for an economy based replication solution is discussed in [11]. By not assessing the quality of service one receives from a provider, the consumer cannot be considered completely economically rational. From an economic perspective this blind side to quality of service allows for lower quality services to be offered at lower costs – slowly forcing a global reduction in grid service quality over time as higher quality services become ignored and unprofitable. Factors of transfer quality such as node-to-node link reliability, actual transfer times, and server reliability have been ignored so that future broker decisions will make the same choices.

In [21], several simple data replication strategies were suggested and simulated for their data saving properties on a grid model that resembles the hierarchical network structure of LHC. The methods proposed are an interesting variant of replication scheduling where replicas are pushed by remote nodes instead of being pulled by local policy. This may present some problems especially in LHC where many heterogeneous organizational and national interests may not allow for their resources to be utilized without being able to exercise local control. Furthermore, a Fast Spread technique may become too taxing on network bandwidth as each file, ranging between 2 to 10 gigabytes, will be cyclically replaced all over the grid. As storage runs out of space, it will remove a replica, only to have to replica it again at a later date.

In the rest of the paper, we assume that the grid network uses a fair share policy for bandwidth. Also in addition to standard data grid middleware services, we assume there are network services that return basic network statistics such as: the number of transfer connections currently active on a server for grid traffic; the maximum available bandwidth between two grid nodes; and the currently available bandwidth between two grid nodes. Actual network sensors such as NWS [23], which is a distributed network traffic forecasting system, would be able to collect these statistics.

3. Data Replication Broker and Policy

This section describes the proposed data replication broker and policies that utilizes a *commodity* market economy for the purposes of resource management. We will detail the broker algorithms that determine the "best" server when given a file replication request from the user (or replica selection service) and a list of hosting servers from a service directory query.

Figure 1 describes the services that are most commonly used in the proposed data replication scheme and shows where the resource broker is situated. The file transfer functionality of the data replication broker is accessed in three scenarios:

1. A user application that requires data.
2. An automated replication scheme through the replica selection service has initiated replication.
3. Site owner that has foreknowledge of expected data demand.

In all situations, the resource broker is given the identity of the file to be replicated and provided there is enough credit to complete the transfer service, the broker will select the "best" resource provider to initiate the transfer.

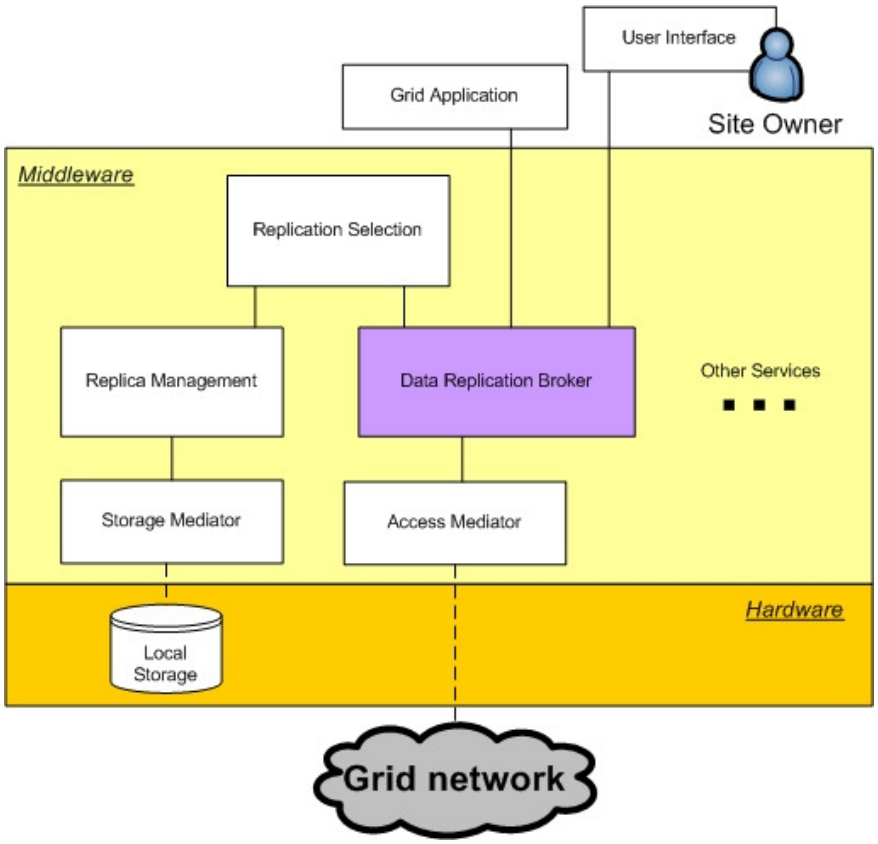

Figure 1: Top level data replication services

3.1 File Transfer Protocol

The general protocol for file transfer is as follows:

1. A replication request of size $f > 0$ MB is received by the broker.
2. The broker communicates with the nearest service directory and retrieves the *service list* (list of service providers) that is able to serve the file. This list also contains the cost of transferring data per megabyte p_j associated with some server j. This cost is the agreed service charge by the service provider.
3. The broker will input the service list, the file request and other statistical information required into the transfer policy. The policy algorithm will output the "best" service provider j^*.
4. The broker will confirm the transfer and the cost per MB of transfer p_{j^*} directly with j^*.
5. The broker will place the transfer request into the *transfer queue*.
6. Transfer will begin immediately and is monitored throughout the transfer.
7. When the transfer is complete, the service provider will bill the broker for the total service cost T and inform the data grid bank, where:

$$T = f\, p_{j^*} \qquad (1)$$

8. The broker pays the total service cost through the data grid ban
9. Transfer details are registered and stored for use in future server selection decisions in the broker.

3.2 Service Pricing Policy

The service provider's ultimate goal is to try to maximize its revenue. Therefore, in order to achieve the most amount of revenue over time from file transfers, the service provider will attempt to increase the chances of return customers by maintaining a standard of service while setting a competitive price.

A service provider will periodically upload the price of its transfer service along with a list of served files to the nearest service directory. By posting a price on the service directory, the service provider is agreeing to a contract to charge a service consumer this price until the price has timed out at the service directory. Furthermore, the per megabyte price of the service is confirmed between the broker and the service provider at the beginning of transfer. Therefore, even if the service provider updates its service rate during a transfer, the consumer will still be charged at the initial confirmed rate.

We use a dynamic costing policy and the pricing function for a price per megabyte p on a service provider j is defined as shown in Eq. (2). The main motivation for implementing a dynamic costing policy was to simulate the resource balancing effects of an economy market and to this end, a simple function should be sufficient.

In Eq. (2), the constant base price of the service is essentially multiplied by the level of the demand; the function above is left intentionally non-continuous to reduce price fluctuation and to reflect the manner in which pricing decisions are made after a period of observation and analysis.

$$p(t_j) = \begin{cases} b_j, q_1 \le t_j \le q_2 \\ \frac{3b_j}{2}, q_2 < t_j \le q_3 \\ 2b_j, q_3 < t_j \le q_4 \\ 3b_j, t_j > q_4 \end{cases} \quad (2)$$

where b_j is the base price of node j, t_j is the of transfers for node j and $0 \le q_1 < q_2 < q_3 < q_4$ are constants that define the boundary of the pricing classes. A node will periodically upload the price of its transfer service along with a list of served files to the nearest service directory.

3.3 Data Replication Broker

3.3.1 Minimum Cost and Delay Policy

We refer to the data replication broker proposed in this paper as the minimize cost and delay (MCD) policy. Figure 2 show the MCD algorithm in pseudo-code form. In the algorithm, the parameters "serverList" and "localNode" denote the set of potential servers and a local server respectively.

MCD uses a scoring function (i.e., SDP in the Fig. 2) to determine the "best" server which is the server with the highest score. The function is given the cost per MB of a server and the expected delay time for a file to finish transferring. The delay time begins from the start of a transfer for a file until its completion. The minimum delay time is a QoS estimation based on the current available bandwidth between a potential service provider and the broker. The fitness function is given below:

$$s_{ij}^{Dp} = -w_D D_{ij} - w_p p_j \times f \quad (3)$$

The parameters in Eq. 3 are defined as follows: s_{ij}^{Dp} is the score based on delay and cost between nodes i and j , $w_D > 0$ is the weight for D_{ij} (which is the estimated transfer delay between nodes i and j), $w_p > 0$ is the weight for p_j (which is the cost per MB for service provider j), and f is the file size of the requested file.

The function has placed negatives in front of the weights since it makes intuitive sense that the best server has the best score. The weights w_D and w_p are parameters that determine the negative significance of increased cost and increased transfer delay. If a consumer values the time to transfer a file over increased costs the value of w_D should increase and similarly if the consumer wishes to save credits and does not hold the delay time to be as important then w_p should be increased. If two servers are found to be equally as good, then the algorithm will resolve the tie by choosing server that is closest to the broker (i.e., distance in Fig. 2).

```
Algorithm MCD
INPUT  : serverList, localNode;
OUTPUT: bestServer← Ø
BEGIN
    FOR (a = 0; a < |serverList|; a++) DO
        IF (bestServer = Ø) THEN
            bestServer ← serverList [a];
        ELSEIF (SDP(a) < SDP(bestServer)) THEN
            bestServer ← serverList [a];
        ELSE
            IF (SDP(a) = SDP(bestServer)) THEN
                A = distance(localNode, a);
                B = distance(localNode, bestServer);
                IF (A < B) THEN
                    bestServer ← serverList [a];
                ENDIF
            ENDIF
        ENDIF
    ENDFOR
    return bestServer
END MCD
```

Fig. 2: General protocol for file transfer.

The algorithm will output the "best" service provider. The best server is decided as a compromise between choosing the servers with the shortest time required to fully transfer the file and the ones with the lowest total estimated service cost.

3.3.2 Least Cost Policy

The least cost policy will be used as a baseline to compare with the performance of the proposed policy. In a market economy, this policy is the most basic of

decision making heuristics. The least cost algorithm finds the best server by finding the server with the least cost in the server list. If during the search, the current server has the same cost as the best server found so far, then we consider the server that is closest to the service consumer (the local node) as the new best server.

4. Performance Evaluation

In this section, we describe the implementation and the performance evaluation of the proposed replica management approach.

4.1 Experimental Setup

We used MONARC [15], a data grid simulator, to simulate the effects of the policies discussed in this paper. In order to realize the policies discussed in this paper, however, several main components were built from the ground up. The data grid configuration that forms the basis of testing is shown in Figure 4 and the associated per regional center configurations are summarized in Table 2. The default policy parameters are summarized in Table 3, parameter values that have * denote values that will be varied from one test run to another. The purpose in the basic configuration above was to simulate the scenario where a popular set of LHC experimental data files have been initially been released into the grid. It includes tier 0 (CERN), tier 1 (t1a, t1b, t1c) and tier 2 (t2xx) nodes where tier 0 and 1 nodes typically have higher bandwidth available to them (≥450Mbps). The tier 2 nodes then rush to access or replicate the data files. The demand patterns for the files are randomly distributed amongst 20 files and the service requests are normally distributed to occur between 100 and 300 seconds. The close distribution of the service demands helps simulate busy data traffic in a global data grid as well as to produce a variable global data demand behavior which will be explained later.

Due to the large file sizes and limited bandwidth for every server and WAN that it is connected to, there will exist all three types of transfer bottlenecks as discussed. For example if all the consumers are using services from server t1a, then the maximum bandwidth demanded by all t1a's consumers (1800Mbps) will exceed the 1000Mbps maximum limit on the t1a router. In fact demands from all the other consumers except t2a1 will overwhelm CERN's router first. By setting the distribution of request occurrences on consumers close so that the next replication request will often occur before the previous file finishes will cause the maximum bandwidth available per transfer to vary over time. Early in the simulation, transfers will still finish relatively quickly. As time passes, there will be instances of concurrent sessions with more and more transfer on all nodes until the requests run out and a drop in the number of transfers running concurrently on a node occurs.

The interactions of the nodes in the grid are complex especially due to the existence of the variable priced services, file request (binomially distributed via time of request arrival) and file sizes. The smallest conceptual granularity of data sent through the network for file transfers are *fragments*. The size of the fragment is configurable with a default maximum size is set at of 10MB. To minimize waste, smaller sized fragments are allowed for the last fragment of a file transfer. The grid fabric uses a fair share policy for bandwidth. That is if there are x number of transfers on a link between site a and site b, and if both sites have the same bandwidth y both up and downstream respectively then the maximum effective transfer speed (ETS) of every ongoing transfer is:

$$\mathrm{ETS} = \frac{y}{x} \quad (Mbit/s) \qquad (4)$$

4.2 Performance Metrics

The cost of a service is measured in how much data is transferred on a per megabyte basis. The smallest conceptual granularity of data sent through the network for file transfers are *fragments*. The sizes of the fragments are a parameter of simulation set at a default maximum size of 10MB, to minimize waste, smaller sized fragments are allowed for the last fragment of a file transfer. The larger than normal maximum size of the fragment is used to improve the efficiency of the simulation.

We used a number of metrics to compare the proposed policy with the baseline policy. The average time required to transfer a good MB is a per transfer measurement of:

$$\mu' = \frac{\sum Time_{Transferring}}{FileSize - \sum Fragment_{BAD}} \qquad (5)$$

where FileSize denotes the total size of the file, $\mathrm{Time}_{\mathrm{Transfering}}$ is the total time for transferring the file and $\mathrm{Time}_{\mathrm{BAD}}$ denotes the total number of bad fragments. This μ' is averaged over all transfers for an individual node, and then averaged over all nodes in the 4 simulations that used a specific policy. The transfer time to transfer a good MB is a performance measurement that factors in the delay time experienced from bad connections and bad fragments.

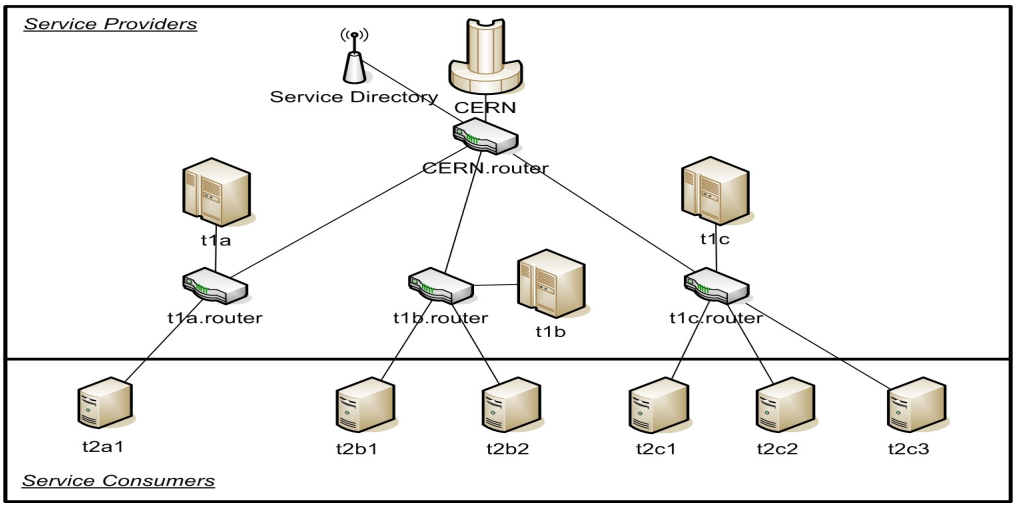

Figure 4: Data grid configuration in the LHC/MONARC structure.

Regional Center Name	WAN speed (Mbps)	LAN speed (Mbps)	Has Transfer Service	Requires Service	Files Demanded	Probability of error during service e_t	Base cost per MB b_j
CERN	1000	600	Yes	No	-	0.1%	2
t1a	1000	550	Yes	No	-	2%	1
t1b	1000	500	Yes	No	-	1%	1.5
t1c	1000	450	Yes	No	-	0.5%	1.7
t2a1	1000	300	No	Yes	15	-	-
t2b1	1000	300	No	Yes	15	-	-
t2b2	1000	300	No	Yes	15	-	-
t2c1	1000	300	No	Yes	15	-	-
t2c2	1000	300	No	Yes	15	-	-
t2c3	1000	300	No	Yes	15	-	-

Table 1: Basic data grid node parameter configuration.

Parameter	***Description***	***Value***	***Comment***
Q_1	Service pricing policy class boundaries	0	Small boundaries due to limited amount of consumer nodes.
Q_2		2	
Q_3		3	
Q_4		4	
f	File size ranges	2K -10K MB	From suggestions by [13]
w_D	MCD constant weight for delay D_{ij}	1*	
w_p	MCD constant weight for cost p_j	0.5*	
Δ	The maximum difference between the highest MCD scoring server and the lowest for MCD-2R candidate selection	200*	
w_g	Weight for the good fragment ratio in the *Rs'* function	30	$w_g + w_e$ must = 100
w_e	Weight for the reverse bad connection intensity ratio λ in the *Rs'* function	70	
w_{sen}	Sensitivity constant for the *Rs* function	20	

Table 2: Default policy parameters.

5. Results and Discussions

Characteristics of sample transfer sessions from this experiment of the MCD algorithm will first be presented below to validate the difference during transfers with and without transfer service migrations. The economic data resource management system were shown to perform with marked improvements when using a variable pricing scheme that is based on the

expected actions of a rational agent compared to a fixed pricing scheme.

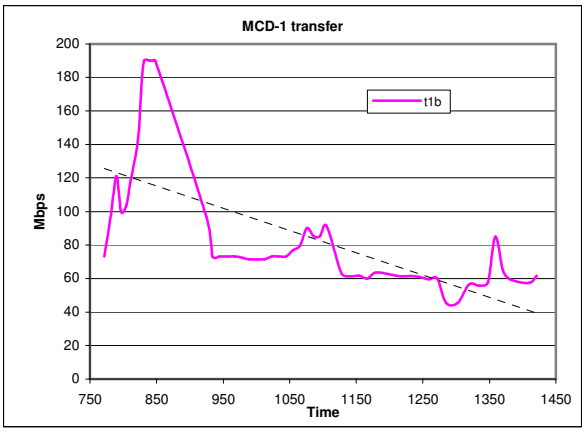

Figure 5: Chart for a typical MCD transfer session with linear trend lines.

5.1 Analysis of Mixed Policy

This experiment compares the performances of the 2 policies on an individual node basis in a data grid with variable pricing scheme. Only the performance of the individual nodes are analysed, not the global grid. Analysis ranges from the quality of the service experienced by the node to the behaviour of individual file transfers. Nodes on the grid have different policies and several different node configurations are run to isolate differences in performance between one policy and another. For example the broker of regional center t2a uses a cheapest cost policy. All policies are installed and simulated once on each consumer node. This to factor in the performance effects that the relative network position of the node in a data grid may have, if any.

This particular transfer service was taken from early in the simulation where average transfer speeds are expected to decrease (average simulations ran for approximately 100+ simulated minutes). The transfer began with relatively high speeds then drops and maintains a low speed for the duration of the transfer. The speed reduction maybe due to various factors such as an increase of transfer requests, a reduction in service quality at the server end or the increase in traffic on the link. From the trend line, the gradient of the decrease in transfer speed over time is more distinct.

5.2 Average Transfer Speed & Cost per MB

Figure 6 and Figure 7 show that the average transfer speed and average cost of the transfer per MB respectively.

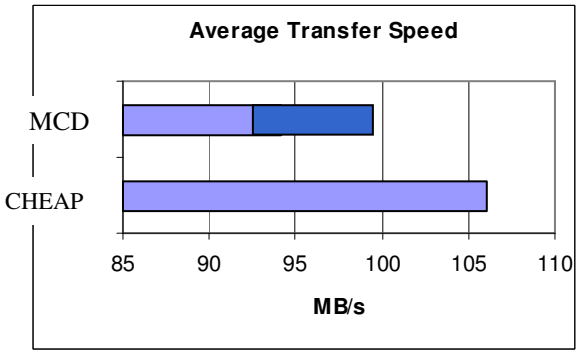

Figure 6: Average transfer speed

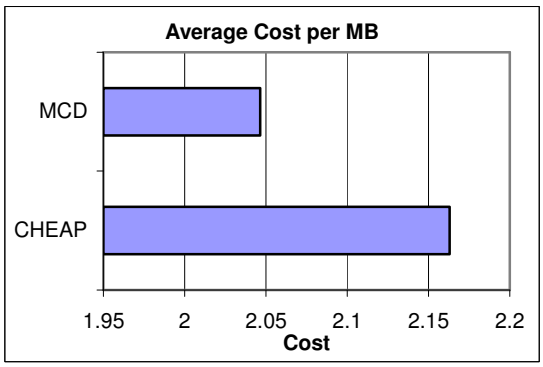

Figure 7: Average cost per MB

The cheapest algorithm used more credits per MB on average than MCD policy. This is due to the tendency for cheapest policy servers to respond slowly to pricing changes in the servers, compared with MCD that factors in speed of transfer.

6. Conclusion and Future Work

This paper explored the effectiveness of economy-based resource management in data grids and proposed a policy for a data replication broker that improved replication. The economic data resource management system were shown to perform with marked improvements when using a variable pricing scheme that is based on the expected actions of a rational agent compared to a fixed pricing scheme. This finding can be generalized to all services on a data grid that use a common currency for transactions in a commodity market. Our future work plan is to use network traces and live data demand patterns on actual data grids. A better pricing policy for the servers should also be explored. An interesting avenue maybe found from

employing game playing algorithms for strategic pricing. A related avenue for research would be to employ different pricing schemes for different classes of users in order to entice desirable customers. The network model was simplified by using a fair share policy. In future works, simulations of policies should explore networks that allow for dedicated classes of network usage.

7. References

[1] T. Hey and A. E. Trefethen, *The UK e-Science Core Programme and the Grid*, Future Generation Computer Systems, Vol. 18, Issue 8, 1017-1031pp, Elsevier Science, Amsterdam, Netherlands, 2002.

[2] A. Chervenak, I. Foster, C. Kesselman, C. Salisbury and S. Tuecke, *The Data Grid: Towards an Architecture for the Distributed Management and Analysis of Large Scientific Datasets*, http://www.globus.org/, 1999. 132.

[3] Belle Collaboration, http://belle.kek.jp/

[4] ATLAS at the University of Chicago, http://hep.uchicago.edu/atlas/

[5] Large Hadron Collider (LHC) at CERN, http://lhc-new-homepage.web.cern.ch/lhc-new-homepage/

[6] W. Hoschek, J. Jaen-Martinez, A. Samar, H. Stockinger, K. Stockinger, *Data Management in an International Grid Project*, 2000 Intl. Workshop on Grid Computing (GRID 2000), Bangalore, India, December 2000.

[7] K. Holtman, *Object Level Replication for Physics*, Proceedings of 4th Annual Globus Retreat, Pittsburgh, July 2000.

[8] W. Allcock et. al., S. *Data Management and Transfer in High-Performance Computational Grid Environments*. Parallel Computing. 2001.

[9] R. Buyya, J. Giddy, and D. Abramson, *A Case for Economy Grid Architecture for Service-Oriented Grid Computing*, 10th IEEE International Heterogeneous Computing Workshop, In conjunction with IPDPS 2001, San Francisco, California, USA, April 2001.

[10] L. McKnight, J. Boroumand, *Pricing Internet Services: Approaches and Challenges*. IEEE Computer 2000; 33(2): 128–129.

[11] W. H. Bell, D. G. Cameron, R. Carvajal-Schiaffino, A. P. Millar, K. Stockinger, and F. Zini. *Evaluation of an Economy-Based File Replication Strategy for a Data Grid*. In Proceedings of 3rd IEEE Int. Symposium on Cluster Computing and the Grid (CCGrid 2003), Tokyo, Japan, May 2003.

[12] M.Carman, F.Zini, L.Serafini, and K.Stockinger. *Towards an Economy – Based Optimisation of File Access and Replication on a Data Grid.* In Workshop on Agent based Cluster and Grid Computing at 2nd International Symposium on Cluster Computing and the Grid (CCGrid2002), Berlin, Germany, May 2002.

[13] K. Ranganathan and I. Foster. *Identifying Dynamic Replication Strategies for a High Performance Data Grid.* In Proc. of the International Grid Computing Workshop, Denver, CO, November 2001.

[14] T. Loukopoulos and I. Ahmad. *Static and Adaptive Data Replication Algorithms for Fast Information Access in Large Distributed Systems*. ICDCS '00: Proceedings of the 20th International Conference on Distributed Computing Systems (ICDCS 2000), IEEE Computer Society Press, USA, 2000.

[15] I. Legrand, C. Dobre, C. Stratan, MONARC Collaboation, *MONARC 2 – Distributed Systems Simulation*, Technical Report, 2003. http://monarc.cacr.caltech.edu:8081/www_monarc/Papers/

[16] H. Stockinger. *Distributed Database Management Systems and the Data Grid.* 18th IEEE Symposium on Mass Storage Systems and 9th NASA Goddard Conference on Mass Storage Systems and Technologies, San Diego, April 17-20, 2001.

[17] R. Buyya, D. Abramson, and S. Venugopal, *The Grid Economy*, Proceedings of the IEEE, Volume 93, Issue 3, 698-714pp, ISSN: 0018-9219, IEEE Press, New York, USA, March 2005.

[18] J. Abawajy, *Placement of File Replicas in Data Grid Environments*, Proceedings of the International Conference on Computational Science 2004: 66-73.

[19] J. Yu, S. Venugopal, and R. Buyya, *A Market-Oriented Grid Directory Service for Publication and Discovery of Grid Service Providers and their Services*, The Journal of Supercomputing, Volume 36, No. 1, 17-31pp, Springer, Science+Business Media, Berlin, Germany, April 2006.

[20] S. Venugopal, R. Buyya, and K. Ramamohanarao, *A Taxonomy of Data Grids for Distributed Data Sharing, Management and Processing*, ACM Computing Surveys, Volume 38, No. 1, -53pp, ACM Press, New York, USA, March 2006.

[21] K. Ranganathan and I. Foster. *Identifying Dynamic Replication Strategies for a High Performance Data Grid.* In Proc. of the 2nd International Grid Computing Workshop, Denver, USA, Nov. 2001.

[22] O. Wolfson and A. Milo, *The multicast policy and its relationship to replicated data placement.* ACM Trans. Database Systems, Vol. 16, No. 1, 181-205pp, Mar. 1991.

[23] R. Wolski. *Dynamically Forecasting Network Performance to Support Dynamic Scheduling Using the Network Weather Service*. In 6th High-Performance Distributed Computing, Aug. 1997.