# Parallel File Transfer for Grid Economic

Chia-Wei Chu[1], Ching-Hsien Hsu[1], His-Ya Chang[2], Shuen-Tai Wang[2], and Kuan-Ching Li[3]

[1] Department of Computer Science and Information Engineering, Chung Hua University, Hsinchu, Taiwan
{cwc,robert}@grid.chu.edu.tw
[2] National Center for High-Performance Computing, Hsinchu, Taiwan
{jerry,stwang}@nchc.org.tw
[3] Department of Computer Science and Information Engineering, Providence University, Taichung, Taiwan
kuancli@pu.edu.tw

**Abstract.** In data grid environments, datasets are usually replicated to many servers when taking into consideration its efficiency. Since these files are usually huge in size, how to efficiently transmit and access between servers and grid users is an important issue. In this paper, we present an economy-based parallel file transfer technique using P2P co-allocation scheme, aiming to service grid applications efficiently and economically in data grids. Taking into consideration the cost factor, we present a novel mechanism for selection of appropriate server, by combining with an adaptive file decomposition scheme and the dynamic adjustment method. In order to evaluate the performance of the proposed method, we have implemented the proposed algorithm and compared with several other previous published techniques. Simulation results demonstrate that the proposed technique is economically effective.

**Keywords:** Data Grid, Parallel File Transfer, Dynamic Adjustment, Co-allocation, P2P.

## 1 Introduction

Grid technology has the great ability to interconnect computational and storage resources scattered at various locations, mutually communicating via networks to share computations and data stores of parallel applications. A number of scientific experimentations with large amount of data analysis such as high-energy physics, gene and protein in biology, simulation of earth science, and other macro issues like origin of cosmos are able to be solved efficiently.

Due to its high efficiency, research for this technology has been broadly performed recently. With this perception, grid computing is viewed as one of important applications for the next-generation network that resource exchanges and knowledge sharing are materialized through the grid technology employed by people. By effectively integrating various specialized tools like computation equipment, databases, software via broadband networks, the grid technology is also a safe, stable, yet simple platform.

P. Mueller, J.-N. Cao, and C.-L. Wang (Eds.): Infoscale 2009, LNICST 18, pp. 76–89, 2009.

Based on the fact that a single computer's performance cannot deal with requirements in scientific applications with enormous amount of data derived from scientific experiments, the grid technology is possibly favorable to these applications by distributing available computations and resources around the world, based on its geometric locations. Presently, the grid technology has been extensively applied in computation and data storage.

In this paper, an efficient scheme for parallel file transfer is proposed, where datasets are replicated to several server stations in data grid environments as copies containing large-size files usually whose efficient distribution and transmission have become a critical issue. Facing this topic [2, 12, 14], one scholar argued the co-allocation architecture [15] and multiple co-allocation manners to fulfill data or files parallel-downloaded in multiple server stations.

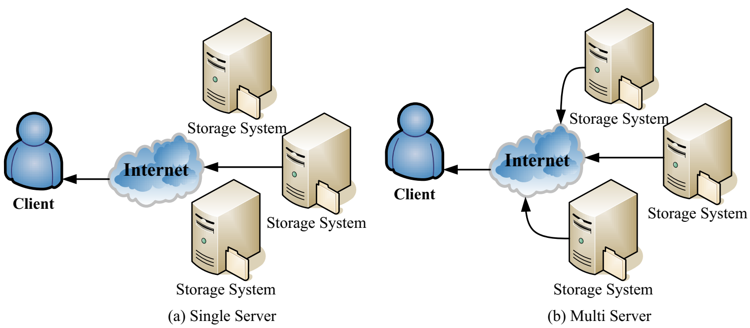


**Fig. 1.** Single Server vs. Multi-Server

Owing to the possibility of worse efficiency in transmission of data downloaded from one server by a client within popular Internet with jam occurring, one method of adopting multiple servers for improvement in efficiency of downloading data (see Figure 1) is available to the data grid environment for solving this issue. In this way, the transmission manners extensively employed for multiple servers have been recommended. As a result of data stored in different locations [2, 15], the concept of agent servers argued by some scholars is able to stabilize the efficiency of parallel transfer [1, 6, 7].

There exist several related works ongoing. Identical-size file blocks distributed in each transmission server make a faster server to spend more time than a slower one while transferring the last file block [15], another approach is to use different servers for transmission of identical file blocks but consuming more network resources [9], ways to minimize differences in time consumed by all servers completing transmission and manage unnecessary waste of network resources for transmission of identical file blocks has become one important issue.

In this paper, the proposed research is to take into account a server's efficiency in transmission and to propose a novel mechanism that effectively and dynamically distribute transmission in which is accessed by cost effects to avoid excessive waste. Knowing that the Grid technology is being employed to various enterprises and even daily lives in the future, enterprisers shall comprehend the importance of effectively improving service quality and increasing customers' satisfaction.

The rest of this paper is organized as follows. The co-allocation architecture and related work is introduced in Section 2, while in Section 3 the mechanism of improving co-allocation scheme is presented. Section 4 describes our approach (*EEPT*), and experimental result performance analysis is done in Section 5. Finally, Section 6 provides the conclusions and future work of this research.

## 2 Related Work

### 2.1 Co-Allocation Architecture

As one of most common technology used in the development of data grids, the Co-Allocation Architecture [15] is also taken as the main scheme in this investigation and partially improved at its architecture. As shown in Figure 2 for operations of the Co-Allocation architecture, a client applying for one request will hand over necessary data and messages to one broker that acquires the available target files' resource contents via an information server such as Grid Information Service (*GIS*) and Monitoring & Discovery System (*MDS*). Servers are selected by the broker, the co-allocation is able to download data and files requested by the client from system-storage servers via GridFTPs. Files are then forwarded to the client from the broker for completion of downloading.

Other transmission mechanisms such as Brute-Force, History-Based and Dynamic Load Balancing are also provided [15]:

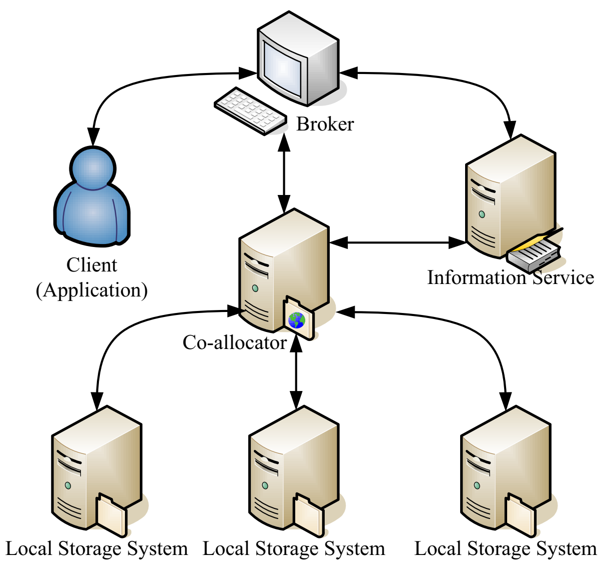


**Fig. 2.** Co-Allocation Architecture

(1) Brute-Force: According to the number of servers, data files are divided into n blocks with an identical size apiece.
(2) History-Based: As a method derived and improved from Brute-Force, History-Based with "more" identical-size blocks are separated from data files and able to distribute those blocks according to the status of server networks.
(3) Dynamic Load Balancing: With more and smaller segmented blocks from data files, the transmission of one block is completed by one server only it is assigned to conduct the next transmission after the previous one is finished.

However, these transmission methods as presented are available to networks with complete quality but possibly change effect of transferring files in case of some problems observed in network system. Furthermore, the contingent efficiency for transmission on realistic networks prompts some scholars to offer other improved transmission mechanisms matching realistic situations.

## 2.2 The Recursive-Adjustment

The Recursive-Adjustment Co-Allocation [3, 4, 5] is an option to reduce the waiting time consumed in transmission of the last block by the slowest in speed server. At the initial stage, this operation is to divide data files into several blocks as a server's references for measurement of bandwidths and then decide the quantity of blocks needed to be delivered by each server, according to computed bandwidths. With the fast-speed server completing its transmission, the blocks for each server's next transmission can be recomputed until the remained block is less than a threshold set to end this recursive-adjustment mechanism for blocks to be transmitted. The Recursive-Adjustment Co-Allocation is presented in Figure 3, where E(Ti) is the expected time for completion of transmission and Ti the realistic time for completion.

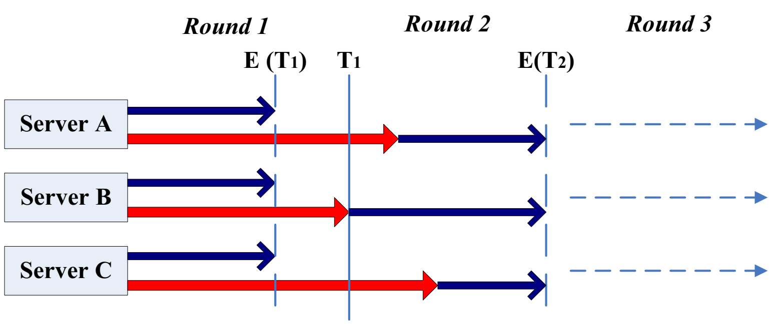


**Fig. 3.** Recursive-Adjustment Co-Allocation

## 2.3 Dynamic Co-Allocation Scheme with Duplicate Assignments

Without the presence of a forecast mechanism or adjustment in a server's variable bandwidths, the Dynamic Co-allocation Scheme with Duplicate Assignments (*DCDA*) [9, 10] has excellent performance. Using a ring-sorting algorithm, *DCDA* divides D datasets into k identical-size blocks which can be delivered by each server without data intersection. Once one assigned block is completely transmitted, another undelivered block will be assigned to this server in accordance to the algorithm which is employed recursively until all k blocks of D dataset are completely transmitted. With this method, possible idle time such as time consumed in communication between a server and a broker will be neglected without exception.

According to the author's algorithm for transmission, there is one problem in *DCDA*. Multiple servers repeatedly transferring identical blocks and therefore occupy large amount of network resources. As result, network jam between other clients and servers is generated. In contrast to above, when not taking into consideration a server's status or selecting servers within above mentioned studies, other studies incorporating selection of servers are described next.

### 2.4 Co-Allocation with Server Selection

The main objective of the Abort and Retransfer mechanism [8] is to improve the worst-efficient server's transmission for the last block. Despite distributed data files, this mechanism allows terminating the worst-efficient server's unfinished operations halfway and transfers undelivered blocks with inspections completed to higher-efficient server makes consumption in time improved for a successful transmission.

Having a P2P environment as the background architecture, the Adaptive Dynamic Parallel Download (*adPD*) [16] distributes each identical-size block segmented from data files, which need to be transferred to individual servers whose transmission tasks can be assisted by the fast transmission-completed server by replacing the slower one. This mechanism's advantage is that no excessive time is consumed in the adjustment of data when transferred in a P2P environment.

Based on algorithms for Co-Allocation mechanism, the Efficient and Adaptive (*EA*) [11] simultaneously access a server's performance as a reference of selecting a server for data transmission and able to reduce consumption in time for selecting servers and fulfills requirements for efficient parallel downloading.

## 3 Research Architecture

Despite quick and temporary communication time consumed in links and communications among all units under the Co-allocation mechanism previously presented, the time accumulated also affects the consumption of time for the completion of transmission. Due to this, a new architecture with the P2P concept has been added for reduction of time consumed in data transfer and links and communications among different units, as shown in Figure 4.

To receive a request for a file with data formats and features to be downloaded from a client, a broker is able to acquire a detailed list of servers owning the requested file by linking to an information server. Once the request for this detailed list is accepted, the client can actively link the file server based to P2P concept. In this fashion, the time consumed in links and communications is drastically reduced, since no data downloaded from a broker but a file allocation conducted by a client.

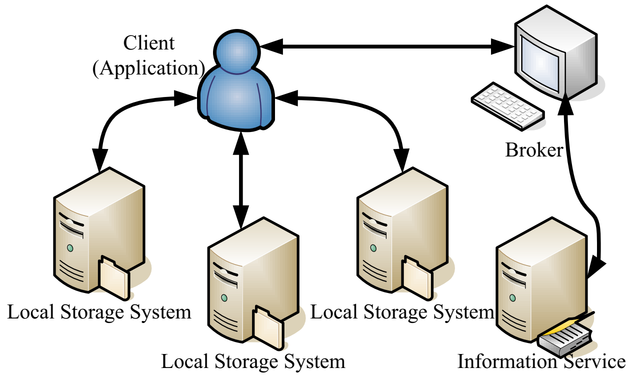


**Fig. 4.** P2P Co-Allocation Architecture

Under considerations in economic costs consumed in server selection of a client for the purpose of downloading files from a server, a mechanism matching given requirements in costs is developed in this proposed research, for the selection of servers which may permit efficient download of requested data files. Maintained at normal running status, all servers mentioned in the paper are guaranteed to be compliant with information servers.

## 4 Efficient and Economic Parallel File Transfer (EEPT)

One of main factors that affect a client's costs with transmission costs included, a client's bandwidths beyond considerations of a transmission mechanism previously is taken into account in this investigation. A server selected in accordance with a client's bandwidths will reduce costs without transmission efficiency being influenced.

### 4.1 Server Selection and File Decomposition

For the case of a server's costs under consideration, the way to select a server becomes the first priority as a result of consumption in costs correlated with a mechanism for the selection of servers. Restricted conditions for this selection of the most beneficial information server are shown as follows: $\beta^{avg}$ (a server's average network bandwidth), $\beta^{max}$ (a server's maximum network bandwidth), $\beta^{min}$ (a server's minimum network bandwidth), $C$ (a server's consumed cost), and $\beta_{client}$ (a client's bandwidth).

As for the first condition for selection, we employ the most common algorithm, Greedy Knapsack Problem, for the determination of a *CP (Capacity Price),* which will be selected first while having the maximum value. Because of an identical value acquired for *CPs*, the second condition with available information about servers such as $\beta^{max}$ and $\beta^{min}$ for a stable value guaranteed is indispensible. In this way, a stable value is expected to 0 as closed as possible. In order to simplify the presentation in the following section, we have some definitions:

**Definition 1**

*$\beta_i^{avg}$ is the average bandwidth of server i.*

**Definition 2**

*$\beta_i^{max}$ is the maximum bandwidth of server i.*

**Definition 3**

*$\beta_i^{min}$ is the minimum bandwidth of server i.*

**Definition 4**

*$C_i$ is the price to use server i to download file per time unit.*

**Definition 5**

*$\beta_{client}$ is the limitation of client bandwidth.*

By deducing a server's *CP* which is acquired from Equation (1), a $\beta_{client}$ equal to or close to zero signifies that the Greedy Knapsack Problem fulfills and serve as an efficient server for transmission. However, considering possible identical *CP*s, we may obtain a server's stable value by means of Equation (2), since one server with a stable value is instrumental to provide a better transmission and efficient use costs. Higher performance server with lower price means higher *CP* value. To simplify

$$CP_i^{avg} = \beta_i^{avg} / C_i \tag{1}$$

In addition, an influence factor $\alpha$ is used for evaluating reliability of a data server. A smaller $\alpha$ is equivalent to less undulation at network's bandwidth which is ideal for one server, especially selected to stabilize transmission efficiency due to slight undulation at bandwidths, and corresponds to economic benefits in costs. The influence factor is defined as follows.

$$\alpha_i = \left(\beta_i^{max} - \beta_i^{min}\right) / \beta_i^{avg} \tag{2}$$

The usage of $\alpha$ in Equation (2) also reflects the definition of average deviation ($\mu$), in probability and statistics, as show in Equation (3). A smaller $\mu$ is equivalent to a more stable network's bandwidth.

$$\mu = \frac{\Sigma \left| \beta_i^{nt} - \beta_i^{avg} \right|}{n} \tag{3}$$

Where $\beta_i^{nt}$ denotes the network bandwidth at the $i^{th}$ time interval; and $n$ is the total amount of time intervals.

Depending on our mechanism on selecting of servers, we are able to distribute blocks from data files directly proportional by using servers' average bandwidth. The example is shown as follows.

For data (1000MB) downloaded by one client, $\beta_{client}$ is 100MB/s and information of servers is listed in Table 1.

**Table 1.** Information of storage serve

| | $\beta_i^{avg}$ | $\beta_i^{\min} \sim \beta_i^{\max}$ | $C$ | $CP$ | $\alpha$ |
|---|---|---|---|---|---|
| Server 1 | 20 MB/s | 15~25 | 2 | 10 | 0.5 |
| Server 2 | 30 MB/s | 20~40 | 2 | 15 | 0.67 |
| Server 3 | 50 MB/s | 45~55 | 4 | 12.5 | 0.2 |
| Server 4 | 50 MB/s | 30~70 | 4 | 12.5 | 0.8 |

As a result of *CP*=15, Server 2 is taken as one of transmission servers ($\beta_{client}$ – $\beta_2^{avg}$ = 70). According to this architecture, $\beta_1^{avg}$ = 20Mbps and $\beta_3^{avg}$ = 50Mbps can be chosen as servers for transmission. With these servers considered for transmission, the bandwidths (average) designated to all servers are respectively 20MB/s (server 1), 30MB/s (server 2) and 50MB/s (server 3), and the sizes divided for each block are 200MB, 300MB and 500MB

### 4.2 Dynamic File Transfer

Despite the size of each server's block to be transmitted and target for transmission simultaneously completed by all servers are known, dynamic adjustments are still required due to a network's dynamic bandwidths. To avoid excessive economic costs wasted in idle time, an easily-understood dynamic adjusted mechanism can reduce idle time and make all servers complete transmission almost at the same time, as shown in Fig. 5.

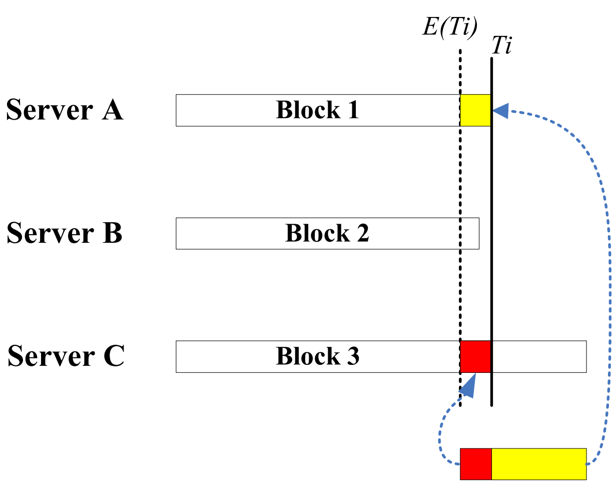


**Fig. 5.** *EEPT* Dynamic Adjustment

With the transmission of one block first completed by one better-efficient server which will be informed for the ongoing progress in block transmission conducted by other servers, this server is able to assist the server completing the least block transmission by sharing the remained block to be transmitted. According to the bandwidth traffic last observed, the remained block to be transmitted shall be redistributed directly proportional. When transmission is simultaneously completed by all servers with minimized consumption in economic costs caused by idle time for the situation of an identical expected target, the whole transmission will be the most efficient. However, it is necessary to set threshold conditions for the sake of avoiding endless development in the adjustment mechanism. To this end, two threshold conditions under this dynamic adjustment mechanism are set. The former one is Remained block to be transmitted, and the latter one is the transmission progress.

While assisting another server's transmission, a server has to check whether the remained traffic is less than or equal to the remained block to be transmitted. For that matter, since the minimum designated to the lowest one of all servers $\beta^{max}$ , an assistant transmission will not be conducted for a remained block less than $\beta^{min}$ and all transmission tasks will be ongoing until completion. As threshold, the minimum $\beta^{min}$ is used for selecting one server matching economic costs also being efficient. On the other hand, to complete transmission before an expected schedule, the minimum, $\beta^{min}$ as one threshold condition, is used for excluding endless dynamic adjustments.

In order to avoid termination of networks out of the first threshold condition with a transmission task failed before an expected schedule, a client has to monitor the transmission progress to exclude terminated networks affecting the whole efficiency. Against this requirement, the progress for transmission of other blocks shall be monitored with one transmission completed by the fastest server and an ongoing dynamic adjustment is conducted. In case of no expected transmission progress within a preset period, the remained blocks will be designated to other servers for transmission forthwith and distributed directly proportional according to servers' bandwidth traffic observed last.

Continue the example in Section 4.1. Segmentation of blocks' sizes is based on the selected average bandwidth for servers: Block1 (200 MB), Block2 (300 MB) and Block3 (500 MB) are designated to Server1, Server2 and Server3 for transmission, respectively. For the case of Server3 completing transmission of block3 before the expected schedule, the client is able to comprehend the transmission progress of Server1 and Server2 simultaneously. Next, for the sake of materializing transmission simultaneously, a block shall be adjusted according to threshold conditions.

## 5 Experiments and Performance Analysis

Experimental results on the efficiency for completion time, idle time, cost, and overhead are obtained and analyzed in this section. The study is based on discussions of our proposed method with other mechanisms argued by relevant studies, as: (1) Single, (2) Brute Force, (3) History, (4) Recursive and (5) Dynamic Co-allocation Scheme with Duplicate Assignments (*DCDA*).

The given conditions in this simulation include $\beta_{client}$ =100 Mbps, the information of servers (shown in Table 2) and the network status of servers (shown in Figure 6).

**Table 2.** Experiment setup

| | $\beta_i^{avg}$ | $\beta_i^{\min} \sim \beta_i^{\max}$ | $C$ | $CP$ | $\alpha$ |
|---|---|---|---|---|---|
| Server 1 | 30 MB/s | 10~ 50 | 2 | 15 | 1.34 |
| Server 2 | 60 MB/s | 40 ~ 80 | 4 | 15 | 0.67 |
| Server 3 | 20 MB/s | 10 ~ 30 | 1 | 20 | 1 |
| Server 4 | 20 MB/s | 15 ~ 25 | 1 | 20 | 0.5 |
| Server 5 | 50 MB/s | 40 ~ 60 | 4 | 12.5 | 0.4 |

According to the server selection policy of *EEPT*, server 4 is firstly chosen as one server for transmission, owing to server 4 has the highest *CP* value and its α value is lower than server 3 when both server 3 and server 4 have *CPs* equal to 20. As the transmission flow of selected servers ( $\beta_4^{avg}$ =20 Mbps) does not exceed the bandwidth of client ( $\beta_{client}$ 100 Mbps), more servers should be selected for parallel file transfer. As a result, server 3 will be chosen in next, resulting a 40 Mbps transmission flow ( $\beta_4^{avg} + \beta_3^{avg}$ =40 Mbps); and finally, server 2 is chosen, making a 100 Mbps transmission flow ( $\beta_4^{avg} + \beta_3^{avg} + \beta_2^{avg}$ = 100 Mbps).

Experimental simulation results for detailed comparison between our proposed mechanism and other mechanisms listed early this section are presented next. Experimental simulation results are obtained utilizing the proposed mechanism for the

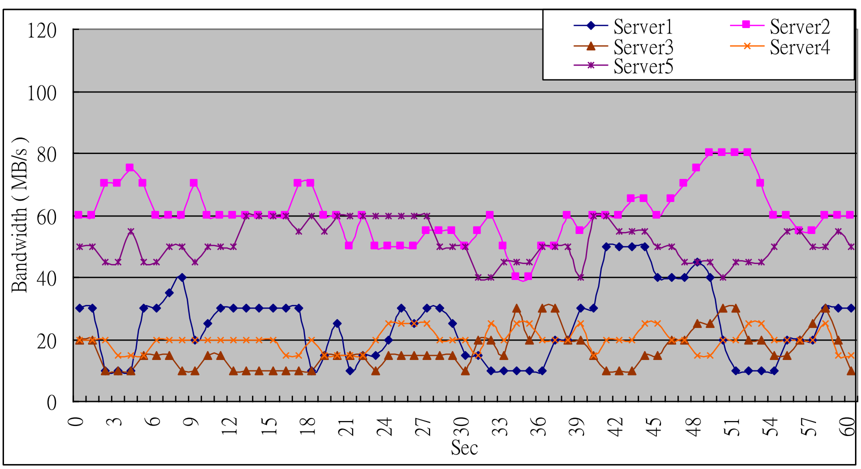


**Fig. 6.** Varying network bandwidth variations

selection of servers, employed for the purpose of corresponding to requirements in efficient servers and avoiding payments at the expense of extra economic costs. In experimental simulations, the sizes of files to be transmitted are 512MB, 1GB, 1.5GB and 2GB.

As shown in Figure 7 for experimental simulations, our mechanism has been verified that the consumption in time for completion of transmission approaches to our expected target and is significantly reduced when compared with other methods. As shown in Figure 8 for comparisons of idle time, the consumption in time with the Dynamic Adjustment technology employed is decreased significantly. In this regard, depending on the degree of simultaneous completion for files transmitted by servers, the performance in idle time will be excellent with minimum time difference.

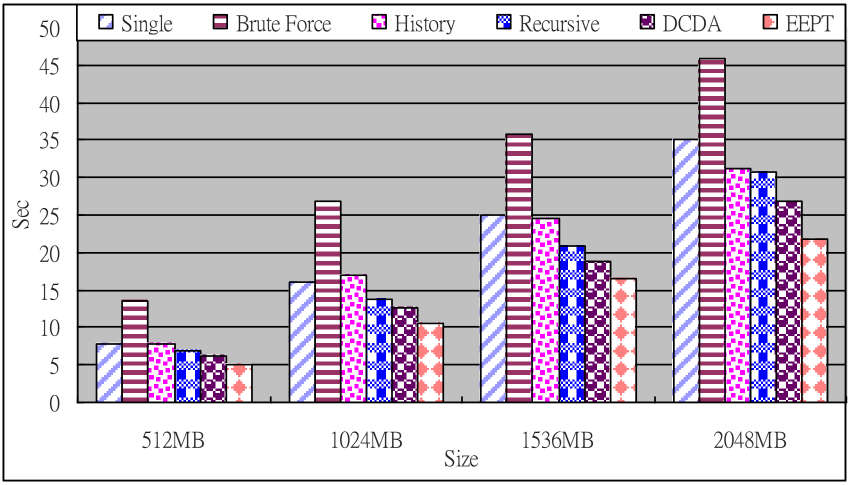


**Fig. 7.** Performance comparisons - Completion time

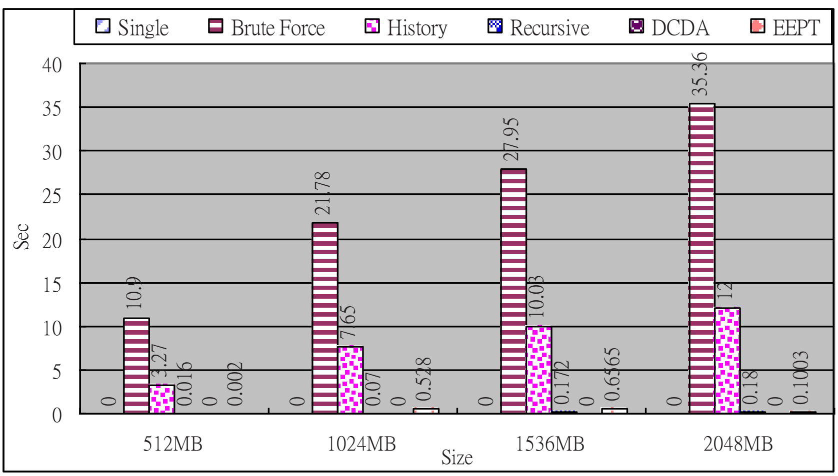


**Fig. 8.** Performance comparisons - Idle time

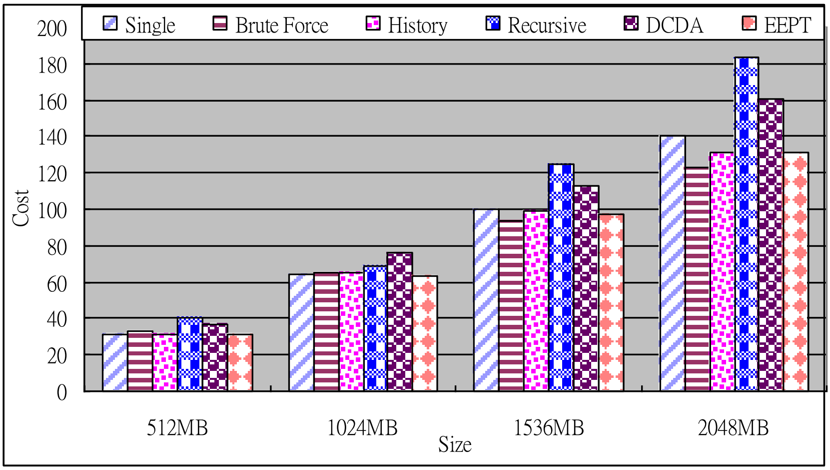


**Fig. 9.** Performance comparisons - Transmission cost

Figure 9 indicates costs consumed by all methods, according to conditions listed in Table 1 wherein the cost from the Dynamic Adjustment technology is higher than others due to larger number of high-efficient servers used in transmission. As set data (costs per second) closing the realistic situations, costs of one high-efficient transmission server indicated in Table 1 are higher than others. Due to possible unfair evaluation to the Dynamic Adjustment technology based on data shown in Figure 9 only, the overhead is designed for fair comparisons:

$$Overhead = Completion\ time \times Cost \qquad (4)$$

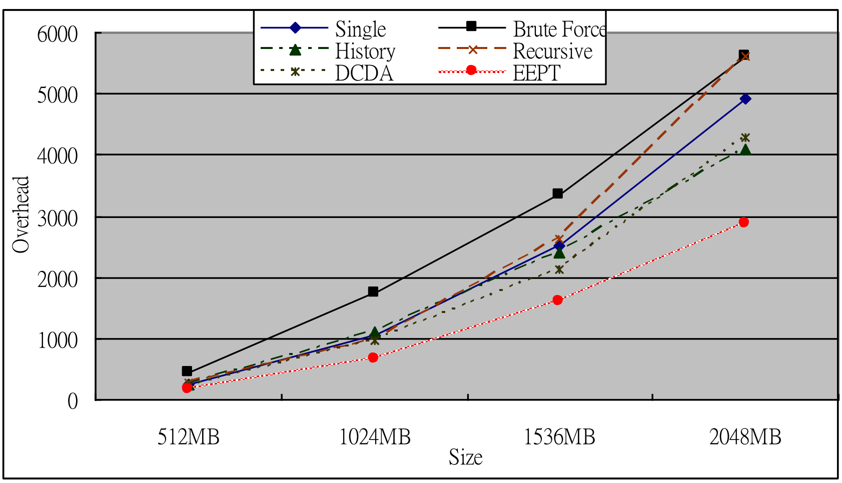


**Fig. 10.** Performance comparisons - Transmission overhead

As shown in Figure 10, a smaller value for overhead is equivalent to a transmission efficiency matching completion time and costs. Based on simulation data with costs incorporated taking into consideration, we can find more servers selected for transmission which do not correspond to better transmission efficiency though higher costs. Therefore, it is required to choose servers matching realistic requirements in transmission. From data indicated, our mechanism is one choice to fit requirement in costs.

## 6 Conclusions and Future Work

With the consideration of economic issues for parallel file transfer in data grids, the proposed mechanism which is the dynamic adjustment scheme combined with server selection method was verified that is able to significantly curtail costs without affecting transmission quality and can maintaining transmission efficiency. As one easily-understood concept and matching realistic requirements in transmission, the dynamic adjustment technology mentioned here along with our P2P Co-Allocation architecture is able to reduce time spent in communication during distribution of blocks conducted by servers.

As future work, a forecasting mechanism executed with exact algorithms will be favorable to the dynamic adjustment technology when precisely segmenting blocks and distributed to other servers. Thus, to reach the final target, researchers shall focus on detailed designs in addition to successful transmission completed synchronously by servers.

## References

1. Kawano, A., Funasaka, J., Ishida, K.: Parallel Downloading Using Variable Length Blocks for Proxy Servers. In: 27th International Conference on Distributed Computing Systems Workshops (ICDCSW 2007), pp. 59–66 (2007)
2. Gkantsidis, C., Ammar, M., Zegura, E.: On the Effect of Large -Scale Deployment of Parallel Downloading. In: IEEE Workshop on Internet Applications (WIAPP 2003), pp. 79–89 (2003)
3. Yang, C.-T., Yang, I.-H., Wang, S.-Y., Chen, C.-H., Li, K.-C.: Improvements on Dynamic Adjustment Mechanism in Co-Allocation Data Grid Environments. Journal of Supercomputing 40(3), 269–280 (2007)
4. Yang, C.-T., Chi, Y.-C., Yang, M.-F., Hsu, C.-H.: A Recursively-Adjusting Co-Allocation Scheme with a Cyber-Transformer in Data Grid. Future Generation Computer Systems 25(7) (2009)
5. Yang, C.-T., Chi, Y.-C., Fu, C.-P., Hsu, C.-H.: An Anticipative Recursively-Adjustment Mechanism for Redundant Parallel File Transfer in Data Grids. In: Proceedings of the Thirteenth IEEE Asia-Pacific Computer Systems Architecture Conference (ACSAC 2008), Hsinchu, Taiwan, August 4-6, pp. 1–8 (2008)
6. Funasaka, J., Nakawaki, N., Ishida, K., Amano, K.: A parallel downloading method of coping with variable bandwidth. In: Proceedings of the 23rd International Conference on Distributed Computing Systems Workshops (ICDCSW 2003), pp. 14–19 (2003)

7. Funasaka, J., Kawano, A., Ishida, K.: Implementation issues of parallel downloading methods for a proxy system. In: Proceedings of the 25th IEEE International Conference on Distributed Computing Systems Workshops, ICDCSW 2005 (2005)
8. Chang, R.-S., Wang, C.-M., Chen, P.-H.: Replica Selection on Co-allocation Data Grids. In: Cao, J., Yang, L.T., Guo, M., Lau, F. (eds.) ISPA 2004. LNCS, vol. 3358, pp. 584–593. Springer, Heidelberg (2004)
9. Bhuvaneswaran, R.S., Katayama, Y., Takahashi, N.: Dynamic Co-allocation Scheme for Parallel Data Transfer in Grid Environment. In: Proceedings of the First International Conference on Semantics, Knowledge, and Grid (SKG 2005), pp. 19–24 (2005)
10. Bhuvaneswaran, R.S., Katayama, Y., Takahashi, N.: Coordinated Co-allocator Model for Data Grid in Multi-sender Environment. In: Dan, A., Lamersdorf, W. (eds.) ICSOC 2006. LNCS, vol. 4294, pp. 66–77. Springer, Heidelberg (2006)
11. Chang, R.-S., Lin, C.-F., Ruey, J.-H., His, S.-C.: An efficient and bandwidth sensitive parallel download scheme in data grids. In: 3rd International Conference on Communication Systems Software and Middleware and Workshops, 2008 (COMSWARE 2008), pp. 296–301 (2008)
12. Vazhkudai, S., Schopf, J.: Predicting Sporadic Grid Data Transfers. In: Proceedings of 11th IEEE International Symposium on High Performance Distributed Computing (HPDC 2002), July 2002, pp. 188–196 (2002)
13. Vazhkudai, S., Schopf, J.: Using Regression Techniques to Predict Large Data Transfers. International Journal of High Performance Computing Applications (IJHPCA 2003), 249–268 (August 2003)
14. Vazhkudai, S.: Enabling the Co-Allocation of Grid Data Transfers. In: Proceedings of the Fourth International Workshop on Grid Computing (GRID 2003), November 17, pp. 44–51 (2003)
15. Xu, Z., Xianliang, L., Mengshu, H., Chuan, Z.: A Speed-based Adaptive Dynamic Parallel Downloading Technique. In: Proceedings of ACM Special Interest Group Operating, pp. 63–69 (2005)