Data Mining on PC Cluster connected with Storage Area Network: Its Preliminary Experimental Results

Masato Oguchi [1,2] and Masaru Kitsuregawa [1]
[1] Institute of Industrial Science, The University of Tokyo
7-22-1 Roppongi, Minato-ku, Tokyo 106-8558, Japan
[2] Research and Development Initiative, Chuo University
42-8 Ichigaya Honmura-cho, Shinjuku-ku, Tokyo 162-8473, Japan
E-mail: oguchi@computer.org

Abstract— **Personal computer/Workstation (PC/WS) clusters have become a hot research topic recently in the field of parallel and distributed computing. They are considered to play an important role as a large scale computer system, such as large server sites and/or high performance parallel computers, because of their good scalability and cost performance ratio. In the viewpoint of applications, data intensive applications such as data mining and ad-hoc query processing in databases are considered very important for massively parallel processors, in addition to the conventional scientific calculation. Thus, investigating the feasibility of such applications on a PC cluster is meaningful.**

In this paper, a PC cluster connected with Storage Area Network (SAN) is built and evaluated. For disk-to-disk copy operation, SAN clusters are much better than LAN clusters. A data mining application is implemented on the cluster. This application requires iterative scans of shared disks, which degrade execution performance due to I/O-bottleneck. In order to resolve the problem, a dynamic data copy method is proposed and evaluated. This method prevents the performance degradation caused by shared disk bottleneck in SAN clusters.

I. Introduction

Latest high performance computer systems are using commodity parts as their components, including CPUs, disks, and memories, rather than proprietary parts. This is because technologies for such commodity parts have matured enough to be used for high-end computer systems. While an interconnection network between nodes has not yet been commoditized until now, some common-purpose networks, e.g. Fast/Gigabit Ethernet, are the strong candidates as a de facto standard of high speed communication networks. With the progress of technologies for such commodity high speed local area networks(LANs), future high performance computer systems will undoubtedly employ commodity networks as well.

Thus, PC/WS clusters using high speed commodity LANs have become an exciting research topic in the field of parallel and distributed computing. They are considered promising platform for future high performance parallel computers, because of their good scalability and cost performance ratio. In the viewpoint of application, we believe data intensive applications including data mining and data warehousing are extremely important for high performance computing in the near future. We previously developed a large scale PC cluster connected with ATM-LAN, and implemented several database applications to evaluate their performance and the feasibility of such applications using PC clusters[1][2].

These LAN-connected PC clusters are used as a system of large server site and/or a high performance parallel computer. In both cases, huge volume of data might be transferred frequently from one node's disk to another, for the execution of parallel computing, load distribution, maintenance of the system, and so on. Because LAN clusters are shared-nothing systems, that is to say, all nodes of the cluster are connected only with a LAN, data is always transferred through the LAN. However, the bandwidth of a LAN in clusters should not be flooded with these kinds of data transfer, because LANs have to be reserved for other purposes as well, such as client-server request communication and parallel/distributed computing among nodes.

In order to reduce LAN traffic and raise availability of nodes in the cluster, Storage Area Networks(SANs), e.g. Fibre Channel, has come to be adopted[3]. SANs can link storage devices directly to all nodes of the cluster, therefore, SANs prevent the congestion of LAN traffic. In the case of SAN clusters, different from LAN clusters, each node does not have to communicate with each other through a LAN for reading data from other nodes' disks, because a pool of storage is shared among all nodes and can be accessed directly through a SAN with no burden to the other nodes nor LANs.

In this paper, we have built a PC cluster which has a SAN-connection as well as a LAN-connection, and examined its performance features. Characteristics of basic data transfer on the cluster are evaluated. Performance of parallel data mining application on the SAN cluster is examined. The method of a dynamic data copy through a SAN during application execution is proposed and discussed.

The rest of paper is organized as follows. In Section II, an overview of our SAN-connected PC cluster is presented, and the characteristics of the cluster is examined. In Section III, the data mining application and its parallelization are explained. The data mining application is implemented and evaluated in Section IV. A dynamic data copy method, which is expected to prevent I/O-bottleneck situation in use of shared disks, is proposed and evaluated also in this section. Final remarks are made in Section V.

II. SAN CLUSTER PILOT SYSTEM AND ITS PERFORMANCE

A. Related works on PC/WS clusters

So many discussions investigating PC/WS clusters can be found in the literature. Initially, the processing nodes and/or networks were built from customized designs, since it was difficult to achieve good performance using only off-the-shelf products[4][5]. Such systems are interesting as a research prototypes, but most of them failed to be accepted as a common platform. However, because of advances in workstation and network technologies, we can build reasonably high performance WS clusters using off-the-shelf workstations and high speed LANs[6][7].

Until recently, workstations were overwhelmingly superior to personal computers, in terms of performance as well as sophisticated software environments. Recent PC technology, however, has dramatically increased its CPU, main memory, and cache memory performance. While RISC processors used in todays WSs provide higher floating point performance than microprocessors used in PCs, some applications such as database processing primarily require good integer performance. Since todays PCs and WSs have almost comparable integer performance, PCs have better cost performance ratio than do WSs for database operations. High speed bus architecture such as the PCI bus has also improved I/O performance of PCs. Moreover, sophisticated UNIX-based operating systems have been implemented on PCs, which should be a great help for realization of PC clusters. Since the size of PC market is much larger than the WS market, further increase in the cost performance ratio is expected for PC clusters.

Several projects on PC clusters were reported[8][9], in which some scientific calculation benchmarks were executed on the cluster. Because performance of PCs and networks used in those projects was not good enough, absolute performance of such clusters was not attractive compared with high-end massively parallel processors. However, preferably good cost/performance has been achieved in these PC clusters[9].

As the performance of PCs has increased dramatically afterward, variety of research projects on PC clusters have been reported until now[10][11][12][13][14]. We believe that data intensive applications such as data mining and ad hoc query processing in databases are quite important for future high performance computers, in addition to the conventional scientific applications[15].

TABLE I

EACH NODE OF THE PC CLUSTER

CPU	Intel 800MHz Pentium III
Main memory	128Mbytes
IDE hard disk	Quantum Fireball 20Gbytes
FC SCSI hard disk	Seagate Cheetah 18.2Gbytes
Operating System	Solaris 7 for x86
Fast Ethernet NIC	3Com 3C905B-TX
Fibre Channel NIC	Emulex LP8000 Host Bus Adapter

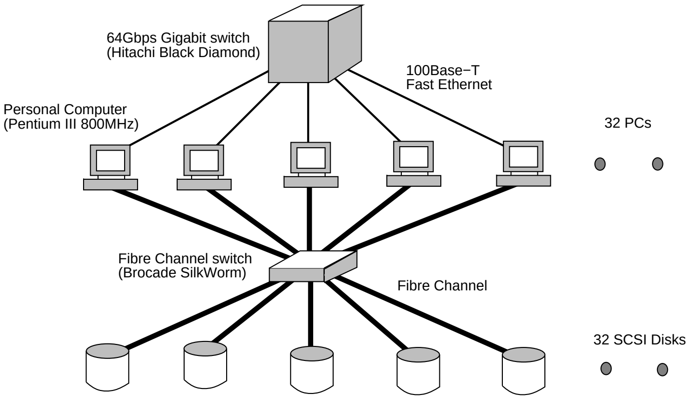

Fig. 1. An overview of SAN-connected PC cluster pilot system

B. An overview of our SAN-connected PC cluster pilot system

Recently we have built the following SAN-connected PC cluster pilot system, and evaluated its data handling performance features. 32 nodes of 800MHz Pentium III PCs are connected with Fast Ethernet as well as Fibre Channel. Each node consists of the components shown in Table I.

All 32 PCs of the cluster and 32 FC SCSI hard disks are connected with a Fibre Channel. Seagate Cheetah 18.2Gbytes is used as SCSI hard disks, and Brocade SilkWorm 2800 is used as a Fibre Channel switch. Switching ability of this device is 200MB/sec per port. Hitachi Black Diamond 6800, which has 64Gbps switching ability, is used as a Fast Ethernet Switch. This switch has more than enough capacity to connect 32 nodes with Fast Ethernet. An overview of the PC cluster is shown in Figure 1.

C. Disk-to-disk copy performance of the system

As a basic characteristic of SAN cluster, disk-to-disk copy performance is measured on the PC cluster pilot system described in the previous subsection. The following two cases of disk copies are compared in this experiment:

In the first case, data is copied from one disk to the other just like LAN cluster. That is to say, the source node reads data from a hard disk, then sends it through Fast Ethernet LAN to the destination node, which receives the data and writes it to its own hard disk. In this case, although the hard disks are accessed

TABLE II

Disk-to-disk copy performance of the PC cluster

Case1: Copy through Fast Ethernet LAN

Node	Source	Destination
CPU(Sys)	20%	40%
LAN(Send)	90Mbps	10Mbps
LAN(Receive)	10Mbps	90Mbps
I/O(Read)	10MB/sec	0
I/O(Write)	0	10MB/sec

Case2: Copy through Fibre Channel

Node	Source	Destination
CPU(Sys)	20%	–
LAN(Send)	0	–
LAN(Receive)	0	–
I/O(Read)	10MB/sec	–
I/O(Write)	10MB/sec	–

through Fibre Channel, each node uses one of shared disks just like its local disk.

In the second case, one node accesses both the source disk and the destination disk through Fibre Channel, and copies the data directly by oneself.

The volume of copied data is 100Mbytes, and the block size of LAN transfer is 8Kbytes. The result of performance at the source and the destination nodes are shown in Table II. The copy times of these two cases are almost the same, because data transfer rate is saturated at a hard disk read/write speed.

As shown in Table II, Fast Ethernet LAN is occupied by the transferred data in the first case. Moreover, CPU usages are relatively high in this case, especially at the destination node. These features are not preferable for PC clusters used as a large server site system. In the second case, on the other hand, the data is copied only by the source node. Therefore, neither other nodes nor the LAN are occupied by this copy. Apparently this mechanism is suitable for the PC clusters, which desire to save the bandwidth of LANs and the CPU power for other purposes.

III. Parallelization of data mining application and its implementation

A. An overview of data mining

Data mining has become an important application in the field of high performance computing. Data mining is a method for the efficient discovery of useful information, such as rules and previously unknown patterns existing among data items in large databases, thus allowing for more effective utilization of existing data. One of the best known problems in data mining is mining of association rules from a database, so called “basket analysis”[16][17]. Basket type transactions typically consist of a transaction identification and items bought per transaction. An example of an association rule is “if customers buy A and B, then 90% of them also buy C”.

The best known algorithm for association rule mining is the Apriori algorithm proposed by R. Agrawal of IBM Almaden Research[18]. Apriori first generates so-called candidate itemsets (groups consisting of one or more items), then scans the transaction database to determine whether the candidates have the user-specified minimum support. In the first pass (pass 1), support for each item is counted by scanning the transaction database, and all items that achieve the minimum support are picked out. These items are called large 1-itemsets. In the second pass (pass 2), 2-itemsets (pairs of two items) are generated using the large 1-itemsets. These 2-itemsets are called the candidate 2-itemsets. Support for the candidate 2-itemsets is then counted by scanning the transaction database. The large 2-itemsets that achieve the minimum support are determined. The algorithm goes on to find the large 3-itemsets, the large 4-itemsets, and so on. This iterative procedure terminates when a large itemset or a candidate itemset becomes empty. Association rules that satisfy user-specified minimum confidence can be derived from these large itemsets.

B. Parallelized association rule mining

To improve the quality of the rule, very large amounts of transaction data must be analyzed and this requires considerable computation time. We have previously studied several parallel algorithms for mining association rules[19], based on Apriori. One of these algorithms, called Hash Partitioned Apriori (HPA), is implemented and evaluated on the PC cluster.

HPA partitions the candidate itemsets among processors using a hash function, like the hash join in relational databases. HPA effectively utilizes the whole memory space of all the processors, and therefore it works well for large scale data mining. The steps of the algorithm are as follows.

1. Generate candidate k-itemsets:

All processors have all the large $(k-1)$-itemsets in memory when pass k starts. Each processor generates candidate k-itemsets using large $(k-1)$-itemsets, applies a hash function, and determines a destination processor ID. If the ID is the processor’s own, the itemset is inserted into the hash table, otherwise it is discarded.

2. Scan the transaction database and count the support value:

Each processor reads the transaction database from its local disk. It generates k-itemsets from those transactions and applies the same hash function used in phase 1. The processor then determines the destination processor ID and sends the k-itemsets to it.

When a processor receives these itemsets, it searches the hash table for a match, and increments the match count.

3. Determine large k-itemsets:

Each processor checks all the itemsets it has and determines large itemsets locally, then broadcasts them to the other processors. When this phase is finished at all processors, large itemsets are determined globally. The algorithm terminates if no large

TABLE III

THE NUMBER OF ITEMSETS AND THE EXECUTION TIME AT EACH PASS

C	Number of candidate itemsets
L	Number of large itemsets
T	Execution time of each pass [sec]

pass	C	L	T
pass 1	–	1219	41.8
pass 2	742371	126	589.1
pass 3	92	52	32.6
pass 4	27	26	31.7
pass 5	8	8	30.9
pass 6	1	0	30.0

itemset is obtained.

IV. EXECUTION OF DATA MINING APPLICATION ON SAN CLUSTER PILOT SYSTEM

A. Implementation and execution of HPA program

The HPA program explained in Section III is implemented on our SAN-connected PC cluster pilot system. Solaris socket library is used for the inter-process communication. As a type of socket connection, SOCK_STREAM is used, which is two-way connection based byte stream. All processes are connected with each other by the socket connections, thus forming mesh topology.

Transaction data is produced using data generation program developed by Agrawal, designating some parameters, such as number of transaction, number of different items, and so on[18]. In this experiment, the number of transaction is 10,000,000, the number of different items is 5,000, and the minimum support is 0.6%. The size of the transaction data is about 800Mbytes in total. The message block size is 8Kbytes, and the disk I/O block size is 64Kbytes in this experiment.

The produced data is stored at one of SCSI hard disks, which is shared by all PCs through Fibre Channel. During execution of the application, this data is accessed by all processes concurrently. The contents of the data is divided by the number of nodes almost equally, and each node of the cluster reads its own portion during the execution. The number of nodes employed in this application is eight. The result of HPA is shown in Table III.

Under the above parameters, the execution of HPA program iterates until pass 6. It is known that the number of candidate itemsets in pass 2 is very much larger than in other passes, as shown in the table. This often happens in association rule mining.

In the next experiment, the transaction data is divided and distributed to many disks before the execution of the program, in order to avoid a conflict of disk reads by multiple nodes. The number of disks used is two, four, and eight. The contents of data are divided by two – eight almost equally, and copied to shared SCSI FC disks beforehand. During the execution, each node reads its own portion from a designated disk.

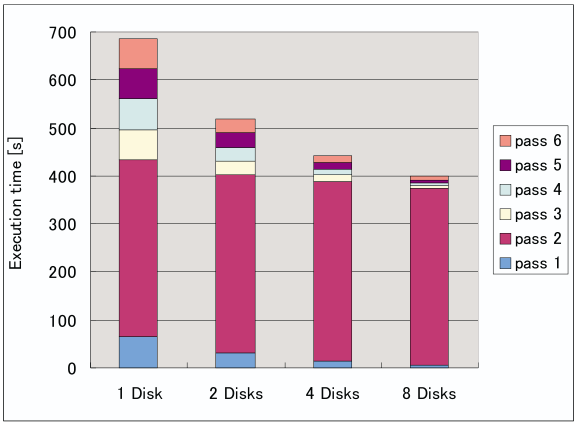

Fig. 2. Execution time of HPA program using 1 – 8 Disks

In Figure 2, execution time of HPA program is shown, when one – eight disks are used. According to the result, the execution time becomes shorter when the number of disks changes from one to eight. When only one shared disk is used, all nodes read data from the same disk, but as the number of disks increases, disk accesses are distributed so that less disk read conflicts happen. As shown in the figure, the execution time becomes shorter in pass 1 and passes 3 – 6. Thus, the disk access bottleneck is resolved in these passes, as the number of disks becomes larger.

On the other hand, the execution time of pass 2 is almost equal in all cases. As shown previously, the number of candidate itemsets in pass 2 is enormous, so that it takes long time to process the data in pass 2. Therefore, the execution time is almost equal in all cases, because pass 2 is a CPU-bound condition rather than a I/O-bound condition.

B. Dynamic data copy method

By monitoring the execution of the program, we have found pass 2 of HPA as a CPU-bound condition. On the other hand, the CPU load is not high in other passes. In those passes, the access to the shared disk becomes bottleneck.

Because each node can access all disks in the storage pool, it is possible to copy portion of required data to other disks which can be accessed exclusively. Hence in the HPA program, portion of the data is copied to their own disks when the data is read during pass 1, then the copied data, instead of the original one, is used afterward. In Figure 3, the execution time of the proposed dynamic data copy method is shown. The proposed method is compared with the original way in which data is read only from the shared disk repeatedly.

As shown in the figure, the execution times in pass 1 and pass 2 are almost equal in both methods. In pass 1, this is because

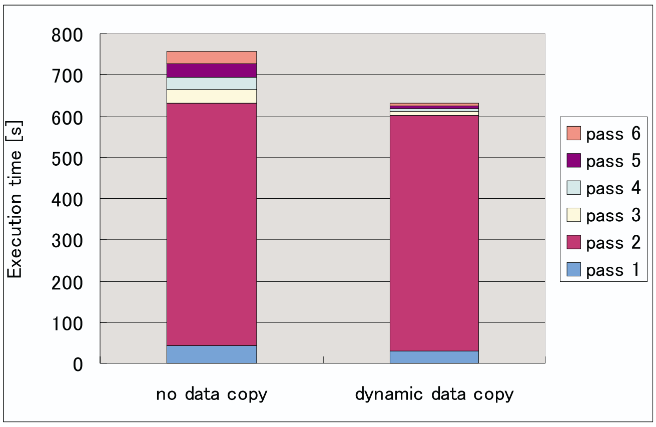

Fig. 3. Execution time of HPA program with dynamic data copy method

data must be read from a shared disk in both cases. In pass 1, the data is copied to the other disks which can be accessed exclusively. However, because pass 2 is a CPU-bound condition, the execution times are almost the same in both methods. In pass 3 − pass 6, on the other hand, the execution time in the dynamic data copy method is shorter than that of no data copy method. This is because I/O-bottleneck situation is resolved by dynamic data copy. While shared disks of a SAN cluster are quite useful for the parallel/distributed computing, sometimes hard disks should be scanned repeatedly in data-intensive applications, which degrades execution performance. The dynamic data copy method achieves better performance in such a case.

V. Conclusions

In this paper, a PC cluster connected with Storage Area Network is built and evaluated. SAN-connected PC clusters are suitable for a large server site because data transfer between disks does not have to be sent through a LAN, thus the bandwidth of a network as well as the load of CPUs can be saved.

We have implemented and evaluated a data mining application on our SAN-connected PC cluster pilot system. In this application, transaction data is scanned repeatedly in iterative passes. Therefore, a dynamic data copy method, in which data is copied from a shared disk to other disks during the execution, is considered to be effective. As a result of the experiment implemented on the SAN cluster, the execution time of each pass becomes shorter, after the copies are completed in the first pass. Thus, the proposed dynamic data copy method is expected to achieve better performance is these cases.

Acknowledgments

This project is partly supported by the Japan Society for the Promotion of Science (JSPS) RFTF Program and New Energy and Industrial Technology Development Organization (NEDO). We would like to thank Tokyo Electron Ltd. for technical help with Fibre Channel-related issues. Hitachi, Ltd. gave us extensive technical help with Gigabit switch.

References

[1] T. Tamura, M. Oguchi, and M. Kitsuregawa: "Parallel Database Processing on a 100 Node PC Cluster: Cases for Decision Support Query Processing and Data Mining", *Proceedings of SC97: High Performance Networking and Computing (SuperComputing '97)*, November 1997.

[2] M. Oguchi, T. Shintani, T. Tamura, and M. Kitsuregawa: "Optimizing Protocol Parameters to Large Scale PC Cluster and Evaluation of its Effectiveness with Parallel Data Mining", *Proceedings of the Seventh IEEE International Symposium on High Performance Distributed Computing*, pp.34-41, July 1998.

[3] B. Phillips: "Have Storage Area Networks Come of Age?", *IEEE Computer*, Vol.31, No.7, pp.10-12, July 1998.

[4] R. S. Nikhil, G. M. Papadopoulos, and Arvind: "*T: A Multithreaded Massively Parallel Architecture", *Nineteenth International Symposium on Computer Architecture*, pp.156-167, May 1992.

[5] M. Blumrich, K. Li, R. Alpert, C. Dubnicki, E. Felten, and J. Sandberg: "Virtual Memory Mapped Network Interface for the SHRIMP Multicomputer", *Proceedings of the Twenty-First International Symposium on Computer Architecture*, pp.142-153, April 1994.

[6] C. Huang and P. K. McKinley: "Communication Issues in Parallel Computing Across ATM Networks", *IEEE Parallel and Distributed Technology*, Vol.2, No.4, pp.73-86, Winter 1994.

[7] D. E. Culler, A. A. Dusseau, R. A. Dusseau, B. Chun, S. Lumetta, A. Mainwaring, R. Martin, C. Yoshikawa, and F. Wong: "Parallel Computing on the Berkeley NOW", *Proceedings of the 1997 Joint Symposium on Parallel Processing(JSPP '97)*, pp.237-247, May 1997.

[8] T. Sterling, D. Saverese, D. J. Becker, B. Fryxell, and K. Olson: "Communication Overhead for Space Science Applications on the Beowulf Parallel Workstation", *Proceedings of the Fourth IEEE International Symposium on High Performance Distributed Computing*, pp.23-30, August 1995.

[9] R. Carter and J. Laroco: "Commodity Clusters: Performance Comparison Between PC's and Workstations", *Proceedings of the Fifth IEEE International Symposium on High Performance Distributed Computing*, pp.292-304, August 1996.

[10] A. Barak and O. La'adan: "Performance of the MOSIX Parallel System for a Cluster of PC's", *Proceedings of the HPCN Europe 1997*, pp.624-635, April 1997.

[11] H. Tezuka, A. Hori, Y. Ishikawa, and M. Sato: "PM: An Operating System Coordinated High Performance Communication Library", *Proceedings of the HPCN Europe 1997*, pp.708-717, April 1997.

[12] M. Oguchi, T. Shintani, T. Tamura, and Masaru Kitsuregawa: "Characteristics of a Parallel Data Mining Application Implemented on an ATM Connected PC Cluster", *Proceedings of the HPCN Europe 1997*, pp.303-317, April 1997.

[13] Y. Ishikawa, A. Hori, H. Tezuka, S. Sumimoto, T. Takahashi, F. O'Carroll, and H. Harada: "RWC PC Cluster II and SCore Cluster System Software – High Performance Linux Cluster", *Proceedings of the Fifth Annual Linux Expo*, pp.55-62, 1999.

[14] M. Banikazemi, V. Moorthy, L. Herger, D. K. Panda, and B. Abali: "Efficient Virtual Interface Architecture (VIA) Support for the IBM SP Switch-Connected NT Clusters", *Proceedings of the International Parallel and Distributed Processing Symposium*, pp.33-42, May 2000.

[15] M. Oguchi, T. Tamura, T. Shintani, and M. Kitsuregawa: "Implementation of Parallel Data Mining on an ATM Connected PC Cluster and Performance Analysis of TCP Retransmission Mechanisms", *The Transactions of the Institute of Electronics, Information and Communication Engineers*, Vol.J81-B-I, No.8, pp.461-472, August 1998.

[16] U. M. Fayyad, G. P. Shapiro, P. Smyth, and R. Uthurusamy: "Advances in Knowledge Discovery and Data Mining", *The MIT Press*, 1996.

[17] V. Ganti, J. Gehrke, and R. Ramakrishnan: "Mining Very Large Databases", *IEEE Computer*, Vol.32, No.8, pp.38-45, August 1999.

[18] R. Agrawal and R. Srikant: "Fast Algorithms for Mining Association Rules", *Proceedings of the Twentieth International Conference on Very Large Data Bases*, pp.487-499, September 1994.

[19] T. Shintani and M. Kitsuregawa: "Hash Based Parallel Algorithms for Mining Association Rules", *Proceedings of the Fourth IEEE International Conference on Parallel and Distributed Information Systems*, pp.19-30, December 1996.