IMPLEMENTATION AND EVALUATION OF RUNTIME DATA DECLUSTERING METHOD OVER SAN-CONNECTED PC CLUSTER

Masato Oguchi [1,2] and Masaru Kitsuregawa [2]

[1] Research and Development Initiative, Chuo University
42-8 Ichigaya Honmura-cho, Shinjuku-ku Tokyo 162-8473, Japan
Email: oguchi@computer.org
[2] Institute of Industrial Science, The University of Tokyo

Abstract. *In this paper, a PC cluster connected with Storage Area Network (SAN) is built and evaluated. In the case of SAN-connected cluster, each node can access all shared disks directly without LAN; thus, SAN-connected clusters achieve better performance than LAN-connected clusters for disk access operations. However, if a lot of nodes access the same-shared disk simultaneously, application performance degrades due to I/O-bottleneck. A runtime data declustering method, in which data is declustered to several other disks dynamically during the execution of application, is proposed to resolve this problem.*

Parallel data mining is implemented and evaluated on the SAN-connected PC cluster. This application requires iterative scans of a shared disk, which degrade execution performance severely due to I/O-bottleneck. The runtime data declustering method is applied to this case. According to the results of experiments, the proposed method prevents performance degradation caused by shared disk bottleneck in SAN-connected clusters.

Keywords. *Cluster Computing, Data Mining, Storage Area Network, Runtime Data Declustering*

1. INTRODUCTION

Recently, personal computer/workstation (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 to be a promising platform for future high performance parallel computers, because of their good scalability and cost performance ratio.

In terms of applications, data intensive applications including data mining and data warehousing are extremely important for high performance computing in the near future. We previously built a PC cluster connected 100 Pentium Pro PCs with ATM-LAN, and implemented several database applications to evaluate their performance and the feasibility of such applications using PC clusters [1].

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. In order to reduce LAN traffic and raise availability of nodes in the cluster, Storage Area Network (SAN), e.g. Fibre Channel, has come to be adopted [2]. SAN can link storage devices directly to all nodes of the cluster; therefore, SAN prevents the congestion of LAN traffic. In the case of SAN-connected clusters, different from LAN-connected clusters, each node does not have to communicate with each other through LAN for reading data from other nodes' disks, because a pool of storage is shared among all nodes and can be accessed directly through SAN with no burden to the other nodes nor LAN.

We have built a PC cluster pilot system, which has a SAN-connection as well as a LAN-connection, and examined its performance features. First, performance of parallel data mining application on the SAN-connected cluster is examined. In the case of SAN-connected cluster, each node can access all shared disks directly. However, if a lot of nodes access the shared disk simultaneously, performance of application must degrade due to I/O-bottleneck. A runtime data declustering method, in which data is declustered to several other disks through a SAN during the execution of application, is proposed and evaluated.

The rest of paper is organized as follows. In Section 2, one of data mining applications, association rule mining, is introduced. In Section 3, an overview of our SAN-connected PC cluster pilot system is shown, and data mining application is implemented and executed on it. A runtime data declustering method, which is expected to prevent I/O-bottleneck in use of shared disks, is proposed and evaluated in Section 4. Final remarks are made in Section 5.

2. ASSOCIATION RULE MINING

In terms of applications in parallel and distributed computing, data mining has attracted a lot of attention from both research and commercial community. 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. Large transaction processing system logs have been accumulated as a result of the progress of bar-code technology. Such data was just archived and not used efficiently until recently. The advance of microprocessor and secondary storage technologies allow us to analyze vast amount of transaction log data to extract interesting customer behaviors.

One of the best-known problems in data mining is mining of association rules from a database, so called "basket analysis"[3]. Basket type transactions typically consist of 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 Apriori proposed by R. Agrawal of IBM Almaden Research [4]. Apriori first generates so-called candidate itemsets (groups consisting of one or more items), and then scans the transaction database to determine whether the candidates satisfy 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 satisfy 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, which are called candidate 2-itemsets. Support for the candidate 2-itemsets is then counted by scanning the transaction database. The large 2-itemsets that satisfy the minimum support are determined. The algorithm goes on to find large 3-itemsets, 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.

In order to improve the quality of the rule, very large amounts of transaction data must be analyzed and this requires considerable computation time. Several parallel algorithms for mining association rules have been previously studied [5], based on Apriori. One of these algorithms, called Hash Partitioned Apriori (HPA), is implemented and evaluated on the PC cluster.

3. SAN-CONNECTED PC CLUSTER AND EXECUTION OF DATA MINING APPLICATION

3.1 OUR PC CLUSTER PILOT SYSTEM AND RELATED WORKS

Various research projects that develop and examine PC/WS clusters have been reported. 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 [6]. Such a system was interesting as a research prototype, 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 [7].

Several projects on PC clusters were reported [8], 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. As the performance of PCs has increased dramatically afterward, variety of research projects on PC clusters have been reported until now [9][10]. Previously, we built a large-scale ATM-connected PC cluster, and implemented and evaluated several database applications on it [1][11].

Recently, we have built a SAN-connected PC cluster pilot system. 32 nodes of 800MHz Pentium III PCs are connected with Fast Ethernet as well as Fibre Channel. Each node consists of components shown in Table 1.

All 32 PCs of the cluster and 32 SCSI hard disks are connected with a Fibre Channel. Seagate Cheetah 18Gbytes is used as SCSI FC hard disks, and Brocade SilkWorm 2800 is employed as a Fibre Channel switch. Switching ability of this device is 200MB/s 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 through Fast Ethernet with non-blocking. An overview of the PC cluster is shown in Figure 1.

Table 1. ***Each node of the PC cluster***

CPU	Intel 800MHz Pentium III
Main memory	128Mbytes
IDE hard disk	Quantum Fireball 20Gbytes
SCSI hard disk	Seagate Cheetah 18Gbytes
OS	Solaris 8 for x86
Fast Ethernet NIC	Intel PRO/100+
Fibre Channel NIC	Emulex LP8000 Host Bus Adapter

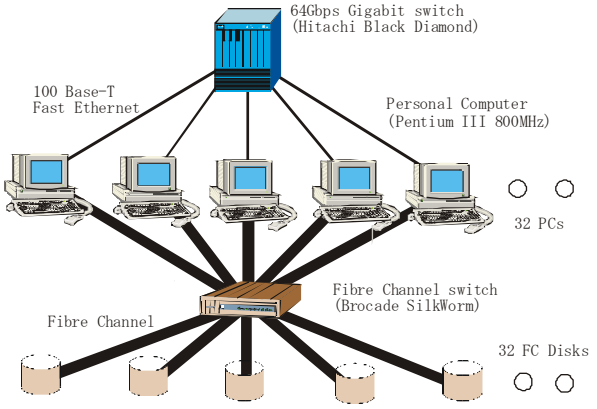

Figure 1. ***An overview of our PC cluster pilot system***

3.2 IMPLEMENTATION AND EXECUTION OF HPA PROGRAM

The HPA program explained in Section 2 is implemented on our SAN-connected PC cluster pilot system. Transaction data is produced using data generation program developed by Agrawal [4], designating some parameters, such as the number of transaction, the number of different items, and so on. The produced data is stored at one of SCSI FC hard disks, which is shared by all PCs through Fibre Channel. During execution of the application, all processes access this data concurrently. Each node of the cluster accesses its own portion of the data, which is assigned to each node almost equally.

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, thus forming mesh topology.

In this experiment, the number of transaction is 10,000,000, the number of different items is 5,000, and the minimum support is 0.7%. 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 number of nodes used in this application is eight. The result of HPA is shown in Table 2.

Using 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 that in other passes. This often happens in association rule mining.

The detail of system performance is measured and analyzed to investigate the behavior of HPA program executed on the SAN-connected PC cluster. CPU usage during the execution of HPA program is shown in Figure 2. This figure indicates that performance is bound by different factors from phase to phase. CPU binds performance of HPA program in pass 2. In pass 2, not only the user mode but also the system mode accounts a considerable part of CPU usage, which is supposed to be consumed for network operation. In other passes, on the other hand, performance is bound mostly by I/O operation instead of CPU. The characteristic of pass 3 is somewhat between CPU-bound and I/O-bound conditions.

Table 2. ***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
Pass1	-	1043	64.8
Pass2	543405	504	253.1
Pass3	395	315	74.6
Pass4	172	109	73.3
Pass5	30	29	66.1
Pass6	4	2	66.2

4. AN EVALUATION OF RUNTIME DATA DECLUSTERING OVER SAN-CONNECTED PC CLUSTER

4.1 RUNTIME DATA DECLUSTERING METHOD

In SAN-connected PC cluster, each node can access all shared disks through Storage Area Network. Thus, an application on each node does not have to care where data is actually located; in a local disk or remote. However, when a considerable number of nodes access to the same-shared disk simultaneously, application performance must degrade due to I/O-bottleneck.

If stored data is accessed repeatedly during the execution of application, the probability of access conflict on the shared disk increases. In such a case, it is preferable to decluster data from one disk to others during/after first access to the data. That is to say, each node copies portion of the data that will be accessed again afterward, from one shared disk to another, and it can be used exclusively. The copied portion of the data, instead of the original one, is accessed after the declustering is completed. We call this method runtime data declustering in the rest of this paper. In some applications, it is difficult to decluster the data before execution of program. Using this method, it is possible to decide dynamically which node needs which portion of the data.

Because SAN traffic does not interfere with each other when target disks are different, application performance should not degrade due to I/O-bottleneck after the data is declustered to several disks. Even in this method, first accesses to the shared

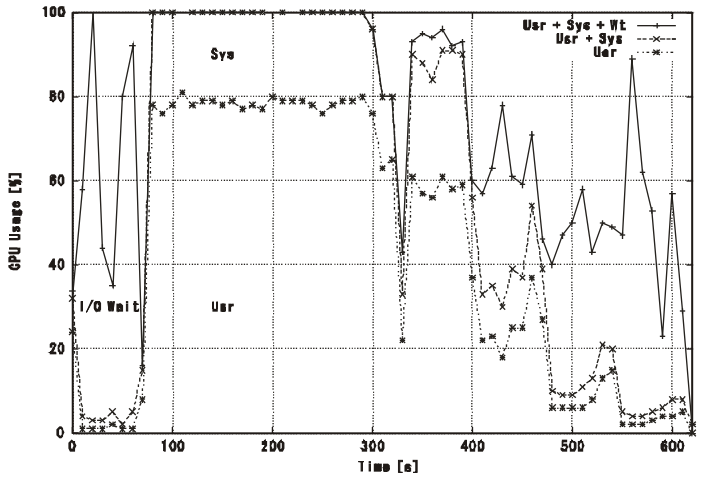

Figure 2. ***CPU usage during the execution of HPA program***

disk may still conflict. In addition, copy operation to another disk is required. However, this seems to degrade performance little, because the data is already read to each node during the first access, and write operation does not conflict on SAN.

4.2 AN EVALUATION OF RUNTIME DATA DECLUSTERING ON PC CLUSTER PILOT SYSTEM

According to Figure 2, we have found pass 2 of HPA program using one shared disk is a CPU-bound condition, but the CPU load is not high in other passes. In those passes, accesses to shared disks become bottleneck.

Runtime data declustering method proposed in previous subsection can be applied in this situation. Because each node can access all shared disks in the storage pool, it is possible to copy its own portion of the transaction data to another disk, which is used exclusively. In this method, portion of the data is copied to their own disks (eight disks in total) when the data is read in pass 1, then the copied data, instead of the original one, is accessed afterward.

In Figure 3, the execution times of the proposed runtime data declustering method are shown. In this experiment, the number of nodes is 16, and the number of transaction is 20,000,000. In the proposed method, the shared data is declustered to 16 FC disks. The proposed method is compared with the original one 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 data must be read from a shared disk in both cases. After pass 1, the data is declustered to other disks that are accessed exclusively. However, because pass 2 is not an I/O-bound condition, the execution times are almost the same in both methods. In pass 3 – 6; on the other hand, the execution time of runtime data declustering method becomes extremely shorter than that of original method. This is because I/O-bottleneck is resolved by runtime data declustering.

5. CONCLUSION

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

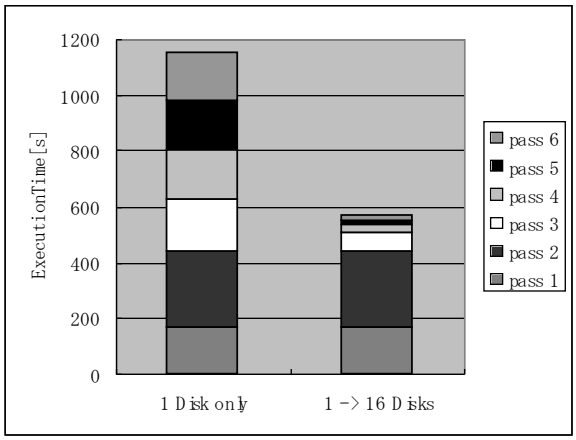

Figure 3. ***Execution time of HPA program (Runtime declustering from 1 to 16 disks)***

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 runtime data declustering method, in which data is declustered from a shared disk to other disks during the execution, is considered to be effective. As a result of the experiment evaluated on the SAN-connected cluster, the execution time of each pass becomes shorter after the declustering is completed in the first pass. Although the execution time of CPU-bound pass does not change, the execution times of I/O-bound passes become extremely shorter after the declustering because the proposed method resolves I/O-bottleneck problem.

While shared disks of a SAN-connected cluster are quite useful for the parallel/distributed computing, disks might be scanned repeatedly in some data-intensive applications, which degrades execution performance severely. The runtime data declustering method achieves better performance in such cases.

ACKNOWLEDGMENT

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

[3] M. J. Zaki: "Parallel and Distributed Association Mining: A Survey", IEEE Concurrency, Vol.7, No.4, pp.14-25, 1999.

[4] 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.

[5] 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.

[6] 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.

[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] 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.

[10] 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.

[11] M. Oguchi and M. Kitsuregawa: "Dynamic Remote Memory Acquisition for Parallel Data Mining on ATM-Connected PC Cluster", Proceedings of the Thirteenth ACM International Conference on Supercomputing, pp.246-252, June 1999.

Masato Oguchi received the B.E. degree in electrical engineering from Keio University in 1990, and the M.E. degree in electrical engineering and the D.E. degree in electronics engineering from the University of Tokyo in 1992 and 1995, respectively. From 1995 to 1996 he was a research fellow at National Center for Science Information Systems (NACSIS, Japan). Since 1996 he has been a research fellow at Institute of Industrial Science, the University of Tokyo. From 1998 to 2000 he was a visiting researcher at Aachen University of Technology in Germany. Since 2001 he has been a associate professor at Research and Development Initiative, Chuo University in Japan. His research interests include distributed computing systems and computer networks. He is a member of IEEE and ACM.

Masaru Kitsuregawa received the B.E. degree in electronics engineering in 1978, and the Ph.D degree in information engineering in 1983 from the University of Tokyo. In 1983 joined Institute of Industrial Science, the University of Tokyo as a lecturer. He is currently a professor at the University of Tokyo and is a director of center for multimedia information processing research. His research interests include parallel processing and database engineering. He currently serves a chair of ACM SIGMOD Japan Chapter and a member of steering committee of IEEE ICDE, PAKDD etc. He was a trustee member of VLDB endowment and the chairman of the technical group on data engineering in IEICE Japan. He is a director of Information Processing Society of Japan.