計算機アーキテクチャ　128－7
ハイパフォーマンス
コンピューティング　70－7
（１９９８．３．５）

ATM結合型大規模PCクラスタにおける並列データマイニングとTCP再送機構の解析

小口正人、新谷隆彦、田村孝之、喜連川優

東京大学 生産技術研究所

〒106-8558 東京都港区六本木7-22-1

{oguchi,shintani,tamura,kitsure}@tkl.iis.u-tokyo.ac.jp

PCクラスタは近い将来の大規模並列計算機システムとして有望であると考えらている。またATMの技術は次世代の標準的な高速通信ネットワークとして有力な候補である。

本論文では100台のパーソナルコンピュータからなるATM結合型PCクラスタを構築し、PCクラスタに適したトランスポート層プロトコルの性質を議論する。ATMスイッチのセル廃棄により引き起こされるTCP再送の解析を行い、大規模PCクラスタ上の並列処理に適した再送機構のパラメータ設定を明らかにした。

またデータベース処理を中心とするアプリケーションは、科学技術計算と並び大規模並列処理の重要なアプリケーションであると考えられる。そこで本研究では並列データマイニングアプリケーションをPCクラスタに実装して評価を行った。提案したTCPパラメータの設定方式を用いることにより、100ノードのPCクラスタ上で並列データマイニングを実行した場合の性能向上が達成された。

Parallel Data Mining on a Large Scale PC Cluster Connected with an ATM Switch and Analysis of TCP Retransmission Mechanism

Masato OGUCHI, Takahiko SHINTANI, Takayuki TAMURA,
and Masaru KITSUREGAWA
Institute of Industrial Science, University of Tokyo

7-22-1 Roppongi Minato-ku, Tokyo 106-8558, Japan

{oguchi,shintani,tamura,kitsure}@tkl.iis.u-tokyo.ac.jp

PC clusters have come to be studied intensively, for a large scale parallel computer in the next generation. ATM technology is a strong candidate as a de facto standard of high speed communication networks.

In this paper, ATM connected PC cluster consists of 100 PCs is reported, and characteristics of a transport layer protocol for the PC cluster are evaluated. Retransmission caused by cell loss at the ATM switch is analyzed, and parameters of retransmission mechanism suitable for parallel processing on the large scale PC cluster are clarified.

In the viewpoint of applications, we believe that data intensive applications such as ad-hoc query processing in databases and data mining is very important for massively parallel processors, in addition to the conventional scientific calculation. Thus parallel data mining application is implemented and evaluated on the cluster. Using a TCP parameters according to the proposed optimization, sufficient performance improvement is achieved for parallel data mining on 100 PCs.

1 Introduction

Recently, PC/WS(Personal computer/Workstation) clusters have become a hot research topic in the field of parallel and distributed computing. They are considered to play an important role as large scale parallel computers in the next generation, from the viewpoint of good scalability and cost performance ratio. The reasons are as follows:

Composition of today's high performance parallel computers are evolving from proprietary components, e.g. CPUs, disks, and memories, into commodity parts. This is because technologies for such commodity parts have matured enough to be used for high-end computer systems. While an interconnection network has not yet been commoditized thus far, ATM technology is one of strong candidates as a de facto standard of high speed communication networks. With a high performance network such as ATM, future parallel computer systems will undoubtedly employ commodity networks as well. ATM switches and NICs (Network Interface Cards) are already becoming cheaper, increasing their cost performance ratio as a result.

Looking over these technological trends, ATM connected PC cluster is considered quite promising platform for future high performance parallel computers. In the viewpoint of applications, we believe that data intensive applications such as ad-hoc query processing in databases and data mining are extremely important for massively parallel processors in the near future.

In this paper, several features of ATM connected PC cluster consists of 100 PCs are examined. Characteristics of a transport layer protocol for the interconnection network are discussed, and parameters of retransmission mechanism suitable for parallel processing on the large scale PC cluster are clarified. The proposed method is evaluated using a parallel data mining application, and considerably good performance scale-up is achieved up to 100 PCs.

2 Background of this research work

2.1 Studies of PC/WS clusters

Several investigations concerning 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[1][2]. 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 WS and network technologies, we can build reasonably high performance WS clusters using off-the-shelf workstations and high speed LANs[3].

Until now, several projects on PC clusters have been reported[4][5], in which some scientific calculation benchmarks were executed on the cluster. Because performance of PCs and networks used in the projects was not good enough, absolute performance of such clusters was not attractive compared with high-end massively parallel processors. Preferably good cost/performance has been achieved however, in these PC clusters[5].

2.2 Features of our project

Our studies on PC cluster have several features, different from other research works, as follows:

First, we have constructed a large scale PC cluster. While other reported PC clusters have several or several tens nodes at most, we realized a cluster consists of 100 Pentium Pro PCs. As far as the authors know, there have not been done researches on large scale PC clusters, having over 100 nodes. The only reported cluster consists of over 100 nodes is UCB's NOW project in which 105 SPARC WSs are connected with Myrinet[6]. Since amount of data which is processed and transferred simultaneously is quite different between the cases of large scale and small scale clusters, it is difficult to discuss the behavior of a large scale system unless we construct over 100 nodes clusters actually.

Second feature, ATM is used for the communication network in our cluster. Since other high speed networks such as Fast Ethernet are also widely used, some cluster experiments employ those media. Moreover, a cluster-oriented network like Myrinet has come to be commercially available[6][7][8], which provide better network performance, although several restrictions exist, such as a limited distance between nodes. Compared with these networks, ATM is extensively used from local area to widely distributed environments. This seamless structure and its quality control mechanisms are among the merits of ATM technology, compared with other high speed networks. Although it has been said ATM may not be suitable for pure data transmission purposes and/or does not fit with traditional computer communication protocols such as TCP/IP, recent dramatical improvements of computer and NIC technology are solving these problems.

Because ATM is developed as a general network rather than a dedicated network, we must investigate if any problem exists when it is used as a connection network of the large scale PC cluster. Especially in this paper, we are focusing on a transport layer protocol on ATM networks, that is, how to make TCP/IP over ATM work well on the cluster. TCP is not only a very popular reliable protocol for computer communication, but also having quite general function as a transport layer. Thus the results of our experiments must be valid even if other transport protocols are used, for investigating connection-oriented communication protocols on large scale clusters.

As a third feature of our project, we used data intensive applications for the evaluation of the PC cluster. Various research projects to develop PC/WS clusters have been reported until now. Most of them however, only measured basic characteristics of PCs and networks, and/or some small benchmark programs were examined. In NOW project also, the WS cluster is evaluated using scientific applications and simple sorting programs[6]. Data intensive applications such as ad-hoc query processing in databases and data mining are considered very important applications for parallel processors, in addition to the con-

ventional scientific calculations. In this paper, we employ data mining as an example of data intensive applications.

3 An overview of our PC cluster

In our pilot system, 100 PCs are connected with an ATM switch. 200MHz Pentium Pro PCs are used as the node of the cluster. Each node consists of components shown in Table 1.

Table 1: Each node of PC cluster

CPU	Intel 200MHz Pentium Pro
Chipset	Intel 440FX
Main memory	64Mbytes
Disk drive	2.5Gbytes IDE hard disk
OS	Solaris2.5.1 for x86
ATM NIC	Interphase 5515 PCI ATM Adapter

All nodes of the cluster are connected by a 155Mbps ATM LAN as well as a Ethernet. We use RFC-1483 PVC driver, which support LLC/SNAP encapsulation for IP over ATM[9][10]. Only UBR traffic class is supported in this driver. TCP/IP over ATM is used as communication protocols.

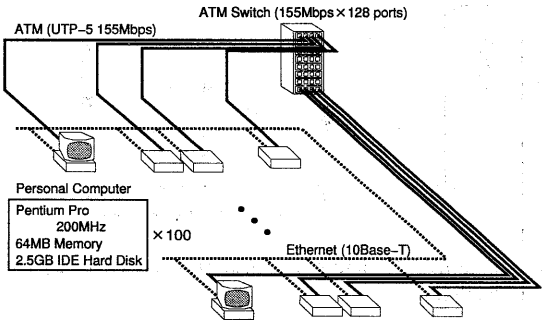

Figure 1: An overview of the PC cluster

HITACHI's AN1000-20 is used as an ATM switch. Since this switch has 128 port 155Mbps UTP-5, all nodes can be connected directly with each other, hence no need to be a cascade configuration. An overview of the PC cluster is shown in Figure 1. On this PC cluster, we achieved about 120Mbps throughput in the case of point-to-point communication, even with so-called "heavy" TCP/IP protocol.

4 Optimization of transport layer protocol parameters

4.1 Broadcasting on the cluster and TCP retransmission

In parallel and distributed applications, barrier synchronization and exchanges of data are executed frequently. In such a case, all-to-all broadcasting takes place. Even if the amount of broadcasting data is not large, a lot of collisions happen in a large scale ATM connected PC cluster, if timing of the broadcasting is the same at all nodes. This is a serious problem. When broadcasting is performed almost simultaneously at all nodes, and a network becomes heavily congested, cells will be discarded at the ATM switch and TCP retransmission should happen as a result. Thus retransmission by a transport layer protocol must have significant meaning on ATM connected PC clusters.

Several experiments are executed on 100 nodes of the PC cluster, in order to investigate retransmission characteristics. Two parameters changed here are 'maximum interval of TCP retransmission' and 'minimum interval of TCP retransmission'. We call them 'MAX' and 'MIN' respectively in the rest of the paper. The default setting is MAX = 60000[msec] and MIN = 200[msec] in the current version of Solaris. The interval of retransmission is dynamically changed according to the mechanism of TCP, within the limits between MAX and MIN.

4.2 Changing maximum interval of TCP retransmission

First, a simple all-to-all broadcasting program is executed on the cluster using 100 nodes. In this program, each node performs barrier synchronization at first, then send 50Kbytes data to all the other nodes, and execute barrier synchronization again.

The broadcasting program is executed when MAX is changed while MIN is fixed at the default value (200[msec]). The execution time of the program is shown in Figure 2. In each case, the program is executed ten times respectively, and all results are indicated by different marks on the Figure.

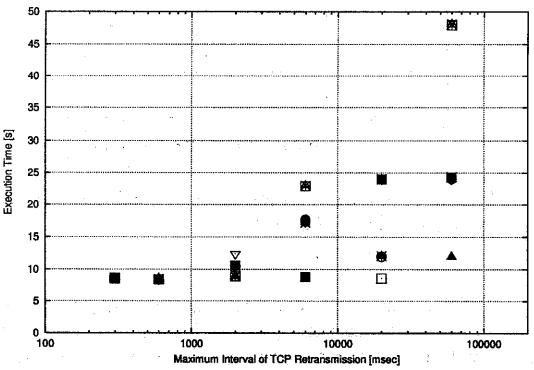

Figure 2: Execution time of the broadcasting program (MIN = 200[msec])

As shown in the figure, the marks are dispersed but not completely at random, because the execution time in this evaluation is decided mainly by TCP retransmission interval, which takes an exponential back-off value such as

6sec, 12sec, 24sec, and so on. The execution time tends to be smaller when MAX is short. Since the application itself is not changed, these differences must come from TCP retransmission waiting only.

The most right points of the figure is MAX = 60000[msec], that is, the default value. Obviously the default value of MAX is not favorable as the execution time becomes longer due to the unnecessarily long retransmission interval. Since general communication protocols assume to be used in a wide area distributed environment, maximum interval of retransmission is set to be quite long. Such long retransmission interval is meaningless in the case of local cluster, thus we should prevent it from becoming longer.

4.3 Optimization of retransmission interval for a large scale PC cluster

Next, the broadcasting program is executed when MIN is changed. MAX is also changed from the default value, such as MAX = MIN + 100 [msec]. The execution time is shown in Figure 3, and the amount of TCP retransmission during the execution is shown in Figure 4. The amount of TCP retransmission is represented per each node, which is the average value in the cluster. In this experiment also, the program is executed ten times respectively, and all results are indicated by different marks on the Figures.

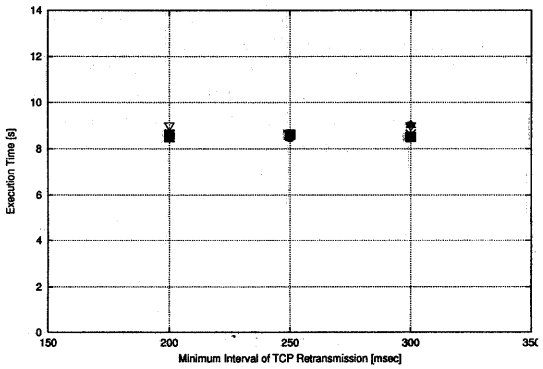

Figure 3: Execution time of the broadcasting program (MAX = MIN + 100 [msec])

The execution time of the program is reasonably short, and not so much changed when MIN varies. On the other hand, the amount of retransmission is dispersed randomly as MIN is changed.

The mechanism of TCP is dynamically changing the interval of retransmission, to set the most suitable value at each moment. Different from the case of communication among several number of nodes, however, this method is not sufficient for a large scale PC cluster, because great number of nodes may use the same value for the interval of retransmission, which causes a collision and heavy traffic congestion.

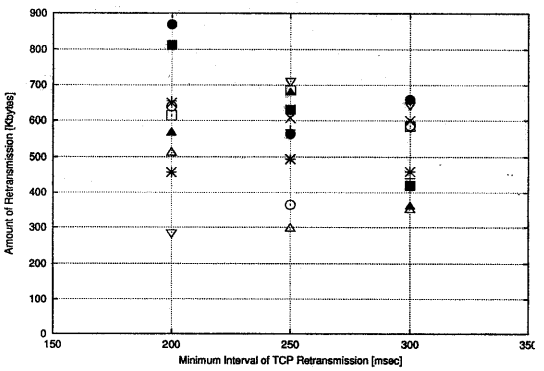

Figure 4: Amount of retransmission (MAX = MIN + 100 [msec])

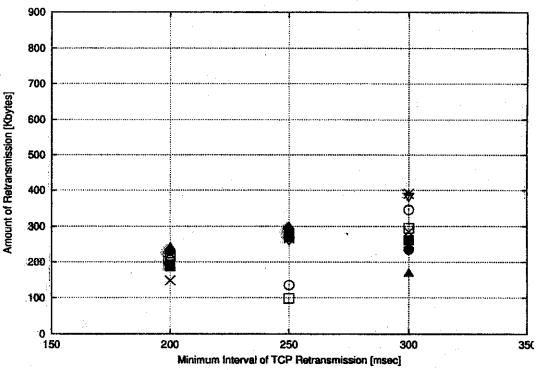

Figure 5: Amount of retransmission (MAX = MIN + 100 [msec])

Thus we used randomly different value from node to node, as the interval of TCP retransmission. In this experiment, random value between X ... X + 100[msec] is used as MIN, and MAX = MIN + 100[msec]. Note that MAX and MIN is randomly different from node to node, but they are not changed during the execution dynamically. The amount of retransmission during the execution is shown in Figure 5. The execution time of the program has become almost the same with Figure 3.

The amount of retransmission shown in Figure 5 is totally lower than that shown in Figure 4. According to these results, changing the interval of TCP retransmission dynamically may not be enough for an application including simultaneous multicasting, in the case of a large scale PC cluster. Using randomly distributed value among nodes provides better performance in such a case.

5 Parallelized data mining application

Data mining is the method of the efficient discovery of useful information such as rules and previously unknown patterns existing among data items embedded in large databases, which allows more effective utilization of existing data. One of the most well known problems in data mining is mining of the association rules from a database, so called "basket analysis"[11]. Basket type transactions typically consist of transaction id and items bought per-transaction. An example of an association rule is "if a customer buys A and B then 90% of them buy also C". The most well known algorithm for association rule mining is Apriori algorithm proposed by R. Agrawal of IBM Almaden Research[12][13].

In order to improve the quality of the rule, we have to handle very large amounts of transaction data, which requires considerably long computation time. We have studied several parallel algorithms for mining association rules until now[14], based on Apriori. One of these algorithms, called HPA(Hash Partitioned Apriori), is implemented and evaluated.

Apriori first generates candidate itemsets, then scans the transaction database to determine whether the candidates satisfy the user specified minimum support. At first pass (pass 1), support for each item is counted by scanning the transaction database, and all items which satisfy the minimum support are picked out. These items are called large 1-itemsets. In the second pass (pass 2), the 2-itemsets (length 2) are generated using the large 1-itemsets. These 2-itemsets are called the candidate 2-itemsets. Then support for the candidate 2-itemsets is counted by scanning the transaction database, the large 2-itemsets which satisfy minimum support are determined. This repeating procedure terminates when large itemset or candidate itemset becomes empty. Association rules which satisfy user specified minimum confidence can be derived from these large itemsets.

HPA partitions the candidate itemsets among processors using a hash function as in the hash join. HPA effectively utilizes the whole memory space of all the processors. Hence, HPA works well for large scale data mining.

6 Execution of the parallel data mining application on PC cluster

6.1 Implementation of HPA program

HPA program has been implemented on the PC cluster pilot system. Each node of the cluster has a transaction data file on its own hard disk. At each node, two processes are created and executed: One process makes candidate itemsets from previous large itemsets, and sends it to the other process, which puts the data into a hash table. Also in the data counting phase, one process generates itemsets by scanning the transaction data file, and send it to the other process on the node decided by hash function, which checks and increments its hash table value appropriately.

Solaris socket library is used for the inter-process communication. All processes are connected with each other by socket connections, thus forming mesh topology. As a type of socket connection, SOCK_STREAM is used, which is two-way connection based byte stream. In the ATM level, PVC (Permanent Virtual Channel) switching is used since the data is transferred continuously among all the processes.

Transaction data is produced using data generation program developed by Agrawal, designating some parameters such as the number of transaction, the number of different items, and so on. The produced data is divided by the number of nodes, and copied to each node's hard disk.

The parameters used in the evaluation is as follows: The number of transaction is 10,000,000, the number of different items is 5000, and minimum support is 0.7%. The size of the transaction data is about 800Mbytes in total. The message block size is set to be 8Kbytes, and the disk I/O block size is 64Kbytes in this experiment.

6.2 Execution of HPA program and evaluating effectiveness of the proposed optimization

The execution time is measured when the number of PCs is changed. The Speedup ratio calculate from the execution time is shown in Figure 6. A solid line indicates the case using default TCP retransmission parameters, i.e. MAX = 60000[msec] and MIN = 200[msec], and a dotted line indicates the case using optimized parameters proposed in Section 4 (MIN = 250 ... 350[ms], MAX = MIN + 100[ms]).

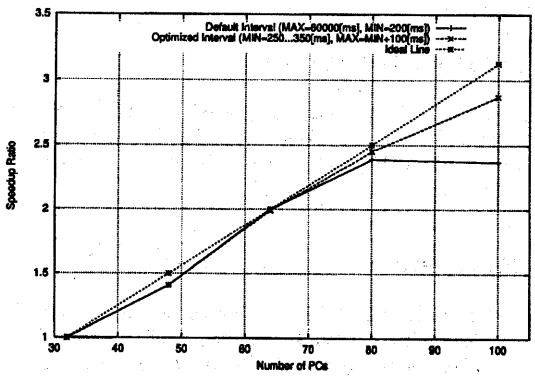

Figure 6: Speedup ratio of HPA program

Reasonably good speedup is achieved as the number of PCs is increased. Using default parameters, however, speedup is saturated and even become worse when the number of PCs is 100. As we traced the number of data transmission during the execution of HPA program, TCP

retransmission was observed in last several passes. Because little data is processed in these last passes, the program execution time itself must be quite short. At the end of each pass, barrier synchronization and exchange of data is needed among all nodes, that is, all-to-all broadcasting takes place. Thus broadcasting is performed almost simultaneously in all nodes at these passes, which causes network congestion and TCP retransmission when the number of PCs is large.

Different from the default case, good speedup is achieved up to 100 PCs using optimized parameters. Since the application itself is not changed, this difference comes from TCP retransmission, occurred along with barrier synchronization and all-to-all data broadcasting, as we saw in Section 4. This kind of barrier synchronization and data broadcasting is frequently used in parallel and/or distributed applications. According to the result of this experiment, the proposed method to optimize retransmission parameters must be quite effective, especially in the case of numbers of nodes being quite large, e.g. 100, in PC clusters.

7 Conclusion

In this paper, optimization a transport layer protocol parameters for the large scale PC cluster was discussed. Retransmission caused by cell loss at the ATM switch was analyzed, and parameters of retransmission mechanism suitable for parallel processing on the large scale PC cluster were clarified. This method was evaluated using data mining application. Default TCP protocol could not provide good performance, since a lot of collisions occur in all-to-all multicasting executed on the large scale PC cluster. Using a TCP parameters according to the proposed optimization, sufficient performance improvement has been achieved for parallel data mining on 100 PCs.

Acknowledgment

This project is partially supported by NEDO (New Energy and Industrial Technology Development Organization). HITACHI, Ltd. technically helped us extensively ATM related issues.

References

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

[2] 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.

[3] 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.

[4] 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.

[5] 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.

[6] 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.

[7] 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.

[8] 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.

[9] J. Heinanen: "Multiprotocol Encapsulation over ATM Adaptation Layer 5", *RFC1483*, July 1993.

[10] M. Laubach: "Classical IP and ARP over ATM", *RFC1577*, January 1994.

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

[12] R. Agrawal, T. Imielinski, and A. Swami: "Mining Association Rules between Sets of Items in Large Databases", *Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data*, pp207-216, May 1993.

[13] R. Agrawal and R. Srikant: "Fast Algorithms for Mining Association Rules", *Proceedings of the 20th International Conference on Very Large Data Bases*, September 1994.

[14] 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.