# Parallel Information Retrieval on an SCI-Based PC-NOW

Sang-Hwa Chung, Hyuk-Chul Kwon, Kwang Ryel Ryu, Han-Kook Jang, Jin-Hyuk Kim, and Cham-Ah Choi

Division of Computer Science and Engineering, Pusan National University,
Pusan, 609-735, Korea
{shchung, hckwon, krryu, hkjang, variant, cca}@hyowon.pusan.ac.kr

**Abstract.** This paper presents an efficient parallel information retrieval (IR) system which provides fast information service for the Internet users on low-cost high-performance PC-NOW environment. The IR system is implemented on a PC cluster based on the Scalable Coherent Interface (SCI), a powerful interconnecting mechanism for both shared memory models and message passing models. In the IR system, the inverted-index file (IIF) is partitioned into pieces using a greedy declustering algorithm and distributed to the cluster nodes to be stored on each node's hard disk. For each incoming user's query with multiple terms, terms are sent to the corresponding nodes which contain the relevant pieces of the IIF to be evaluated in parallel. According to the experiments, the IR system outperforms an MPI-based IR system using Fast Ethernet as an interconnect. Speed- up of up to 4.0 was obtained with an 8-node cluster in processing each query on a 500,000-document IIF.

## 1. Introduction

As more and more people are accessing the Internet and acquiring a vast amount of information easily, more people consider that the problem of information retrieval (IR) resides no longer in the lack of information, but in how we can choose from a vast amount the right information with speed. Many of us have already experienced that some IR systems provide information service much faster than others. How fast an IR system can respond to users' queries mostly depends on the performance of the underlying hardware platform. Therefore, most of the major IR service providers have been urged to spend several hundred thousand dollars to purchase their hardware systems. However, for many small businesses on the Internet, that cost is too high.

In this paper, as a cost-effective solution for this problem, a PC cluster interconnected by a high-speed network card is suggested as a platform for fast IR service. With the PC cluster, a massive digital library can be efficiently distributed to PC nodes by utilizing local hard disks. Besides, every PC node can act as an entry to process multiple users' queries simultaneously.

It is extremely important to select a network adapter to construct a high-speed system area network (SAN). For a message passing system, the Fast Ethernet card or the Myrinet card can be used. For a distributed shared memory (DSM) system, the SCI card can be considered. Fast Ethernet developed for LAN is based on complicated protocol software such as TCP/IP, and its bandwidth is not high. The Myrinet[1] card is a high-speed message passing card with a maximum bandwidth of 160Mbyte/sec. However, the network cost is relatively high because Myrinet

requires crossbar switches for the network connection. Besides, its message-passing mechanism is based on time consuming operating system calls. For applications with frequent message-passing, this can lead to performance degradation. To overcome the system call overhead, systems based on user-level interface for message-passing without intervention of operating system have been developed. Representative systems include AM[2], FM[3], and U-Net[4]. Recently, Myrinet is also provided with a new message-passing system called GM[5], which supports user-level OS-bypass network interface access.

The SCI (Scalable Coherent Interface: ANSI/IEEE standard 1596-1992) is designed to provide a low-latency (less than 1μs) and high bandwidth (up to 1Gbyte/sec) point-to-point interconnect. The SCI interconnect can assume any topology including ring and crossbar. Once fully developed, the SCI can connect up to 64K nodes. Since the SCI supports DSM models that can feature both of NUMA and CC-NUMA variants, it is possible to make transparent remote memory access with memory read/write transactions without using explicit message-passing. The performance of the SCI-based systems has been proven by the commercial CC-NUMA servers such as Sequent NUMAQ 2000[6] and Data General's Aviion[7].

In this research, the SCI is chosen as an underlying interconnecting mechanism for clustering. The Parallel IR system is implemented on an SCI-based PC cluster using a DSM programming technique. In the IR system, the inverted-index file(IIF) is partitioned into pieces using a greedy declustering algorithm and distributed to the cluster nodes to be stored on each node's hard disk. An IIF is the sorted list of terms (or keywords), with each term having links to the documents containing that term. For each incoming user's query with multiple terms, terms are sent to the corresponding nodes which contain the relevant pieces of IIF to be evaluated in parallel. An MPI-based IR system using Fast Ethernet as an interconnect is also constructed for comparison purpose.

## 2. PC Cluster-based IR System

### 2.1 Typical IR System on Uniprocessor

Figure 1 shows the structure of a typical IR system implemented on a uniprocessor. As shown in the figure, once a user's query with multiple terms is presented to the system, for each query term in turn the IR engine retrieves relevant information from the IIF in the hard disk. When all the information is collected, the IR engine performs necessary IR operations, scores the retrieved documents, ranks them, and sends the IR result back to the user. For the efficient parallelization of the system, it is important to find out the most time consuming part in executing the IR system. Using the sequential IR system developed previously[8], the system's execution time is analyzed as shown in Figure 2. In the sequential system, the most time consuming part is disk access. Thus, it is necessary to parallelize disk access. This can be done by partitioning the IIF into pieces and distributing the pieces to the processing nodes in a PC cluster.

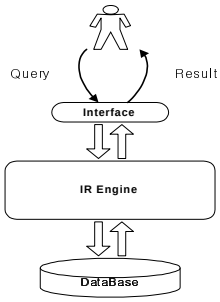


**Fig. 1.** A typical IR system

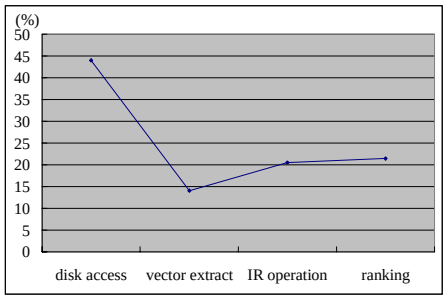


**Fig. 2.** Execution time analysis in the sequential IR system

### 2.2 Declustering IIF

Most current IR systems use a very large lookup table called an inverted index file (IIF) to index relevant documents for given query terms. Each entry of the IIF consists of a term and a list of ids of documents containing the term. Each of the document ids is tagged with a weight of the term for that document. Given a query, all the query terms are looked up from the IIF to retrieve relevant document ids and the corresponding term weights. Next, the documents are scored based on the term weight values and then ranked before they are reported back to the user.

Since our IR system processes user's query in parallel on a PC cluster, it is desirable to have the IIF appropriately declustered to the local hard disks of the processing nodes. We can achieve maximum parallelism if the declustering is done in such a way that the disk I/O and the subsequent scoring job are distributed as evenly as possible to all the processing nodes. An easy random declustering method would be just to assign each of the terms (together with its list of documents) in the IIF lexicographically to each of the processing nodes in turn, repeatedly until all the terms are assigned. In this paper, we present a simple greedy declustering method which performs better than the random method.

Our greedy declustering method tries to put together in the same node those terms which have low probability of simultaneous occurrence in the same query. If the terms in a query all happen to be stored in the same node, the disk I/O cannot be done in parallel and also the scoring job cannot readily be processed in parallel. For an arbitrary pair of terms in the IIF, how can we predict the probability of their co-occurring in the same query? We conjecture that this probability has a strong correlation with the probability of their co-occurrence in the same documents. Given a pair of terms, the probability of their co-occurrence in the same documents can be obtained by the number of documents in which the two terms co-occur divided by the number of all the documents in a given document collection. We calculate this probability for each of all the pairs of terms by preprocessing the whole document collection.

When the size of the document collection is very large, we can limit the calculation of the co-occurrence probabilities only to those terms which are significant. The reason is that about 80% of the terms in a document collection usually exhibits only a single or double occurrences in the whole document collection and they are unlikely to appear in the user queries. Also, since the number of terms in a document collection is known to increase in log scale as the number of documents increases, our

method will not have much difficulty in scaling up. As more documents are added to the collection, however, re-calculation of the co-occurrence probabilities would be needed for maintenance. But, this would not happen frequently because the statistical characteristics of a document collection does not change abruptly.

In the first step of our greedy declustering algorithm, all the terms in the IIF are sorted in the decreasing order of the number of documents each term appears. The higher this number the more important the term is in the sense that it is quite likely to be included in many queries. This is especially true when the queries are modified by relevance feedback[9]. This type of terms also have a longer list of documents in the IIF and thus causes heavier disk I/O. Therefore, it is advantageous to store these terms in different nodes whenever possible for the enhancement of I/O parallelism. Suppose there are *n* processing nodes. We assign the first *n* of the sorted terms to each of the *n* nodes in turn. For the next *n* terms, each term is assigned to the node which contains a term with the lowest probability of co-occurrence. From the third pass of the term assignment, a term is assigned to such a node that the summation of the probabilities of co-occurrence of the term with the terms already assigned to the node is the lowest. This process repeats until all the terms in the IIF are assigned.

### 2.3 Parallel IR System Model

The PC cluster-based parallel IR system model is shown in Figure 3. The IR system consists of an entry node and multiple processing nodes. The participating nodes are PCs with local hard disks and connected by an SCI-based high-speed network. The working mechanism of the parallel IR system model can be explained as follows. The entry node accepts a user' query and distributes query terms to processing nodes (including itself) based on the declustering information described in the previous subsection. Each processing node consults the partitioned IIF using the list of query terms delivered from the entry node, and collects the necessary document list for each term from the local hard disk. Once all the necessary document lists are collected, they are transmitted to the entry node. The entry node collects the document lists from the participating processing nodes (including itself), performs required IR operations such as AND/OR and ranks the selected documents according to their scores. Finally the sorted document list is sent back to the user as an IR result.

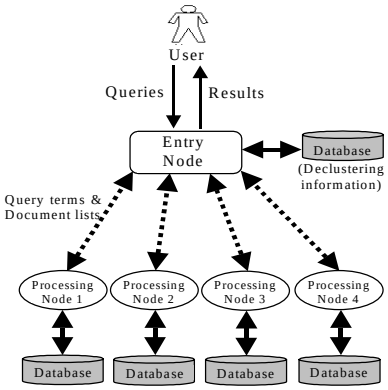


**Fig. 3.** Parallel IR system model

## 2.4 Experimental PC Cluster System

In this research, an 8-node SCI-based PC cluster system is constructed as shown in Figure 4. Each node is a 350MHz Pentium II PC with 128Mbyte main memory and 4.3Gbyte SCSI hard disk, and operated by Linux kernel 2.0.36. In the cluster, any PC node can be configured as an entry node. As shown in the figure, each PC node is connected to the SCI network through the Dolphin Interconnect Solution (DIS)'s PCI-SCI bridge card. There are 4 rings in the network, and 2 nodes in each ring. The rings are interconnected by the DIS's 4×4 SCI switch. For DSM programming, the DIS's SISCI (Software Infrastructure for SCI) API[10] is used. With this configuration, the maximum point-to-point bulk transfer rate obtained is 80 Mbyte/sec approximately.

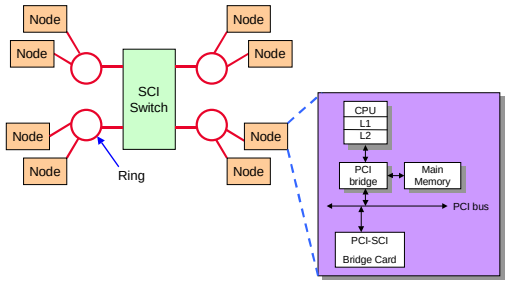


**Fig. 4.** SCI-based 8 node PC cluster system

For comparison purpose, an 8-node Fast Ethernet-based PC cluster system is also constructed. Each PC node has the same configuration as the SCI network's node except that a PCI Fast Ethernet Adapter is used for networking. A switching hub is used to interconnect PC nodes in the cluster. For message-passing programming, MPICH 1.1.1[11] is used. In this case, the maximum point-to-point bulk transfer rate obtained is 10 Mbyte/sec approximately.

## 2.5 SCI-based DSM Programming

The SCI interconnect mechanism supports DSM programming. By using SISCI, a node in the SCI-based PC cluster can establish a mapping between it's local memory address space and a remote node's memory address space. Once the mapping is established, the local node can access the remote node's memory directly. In DSM programming, the communication between PC nodes in the cluster is done using remote read and remote write transactions instead of message-passing. These remote read/write transactions are actually carried out using the remote read/write functions provided by SISCI. When the IR program is actually coded, most of the remote memory transactions are implemented using the remote write function. This is because the remote write function performs about 10 times faster than the remote read function in the DIS's PSI-SCI bridge card.

## 3. Performance of PC Cluster-based IR System

### 3.1 Performance Comparison between SCI-based System and MPI-based System

In this experiment, average query processing times are measured for the 8-node SCI-based system, the 8-node MPI-based system and a single node system. The IIF is constructed from 100,000 documents collected from articles in a newspaper. A user's query consists of 24 terms. Each query is made to contain a rather large number of terms because the queries modified by relevance feedback usually have that many terms. The IIF is randomly declustered to be stored on each processing node's local disk.

As shown in Table 1, the disk access time is reduced for both the SCI-based system and the MPI-based system when compared with the single node system. However, the MPI-based system is worse than the single node system in total query processing time because of the communication overhead. The SCI-based system has much less communication overhead than the MPI-based system, and performs better than the single node system. The speed-up improves with further optimizations presented in the following subsections.

**Table 1.** Query processing times of 8-node SCI-based system and 8-node MPI-based system (unit : sec)

| | SCI-based system | MPI-based system | Single-node System |
|---|---|---|---|
| Send query term | 0.0100 | 0.0251 | 0 |
| Receive document list | 0.0839 | 0.2097 | 0 |
| Disk access | 0.0683 | 0.0683 | 0.2730 |
| IR operation | 0.0468 | 0.0468 | 0.0468 |
| Total | 0.2091 | 0.3500 | 0.3198 |

### 3.2 Effect of Declustering IIF

The greedy declustering method is compared with the random method on a test set consisting of 500 queries each containing 24 terms. To generate the test queries we randomly sampled 500 documents from a document collection containing 500,000 newspaper articles. From each document, the most important 24 terms are selected to make a query. The importance of a term in a document is judged by the value $tf \times idf$, where $tf$ is the term's frequency in that document and $idf$ is the so called inverse document frequency. The inverse document frequency is given by $\log_2(N/n) + 1$, where $N$ is the total number of documents in the collection and $n$ is the number of documents containing the term. Therefore, a term in a document is considered important if its frequency in that document is high enough but at the same time it does not appear in too many other documents. Table 2 shows the experimental results comparing the random clustering and the greedy declustering methods using those 500 queries on our 500,000 document collection.

**Table 2.** Comparison of random declustering and greedy declustering (unit: sec)

| | Random declustering | Greedy declustering |
|---|---|---|
| Average query processing time | 0.5725 | 0.5384 |
| Accumulated query processing time for 500 queries | 286.2534 | 269.1919 |

### 3.3 Performance with Various-sized IIF

In this subsection, the performance of the SCI-based parallel IR system is analyzed with the number of documents increased up to 500,000. These documents are collected from a daily newspaper, and 500,000 documents amount to the collection of the daily newspaper articles for 7 years. The size of IIF proportionally increases as the number of documents increases. For example, the size of IIF is 300 Mbytes for 100,000 documents, and 1.5 Gbytes for 500,000 documents. The 8-node PC cluster and the greedy declustering method are used for the experiment.

The experimental result is presented in Figure 5. It takes 0.1805 seconds to process a single query with the 100,000 document IIF, while it takes 0.2536 seconds with the 200,000 document IIF and 0.5398 seconds with 500,000 document IIF. As the IIF size increases, the document list for each query term becomes longer, and the time spent for IR operations (AND/OR operations) increases considerably. As a result, the IR operation eventually takes more time than the disk access, and becomes the major source of bottleneck.

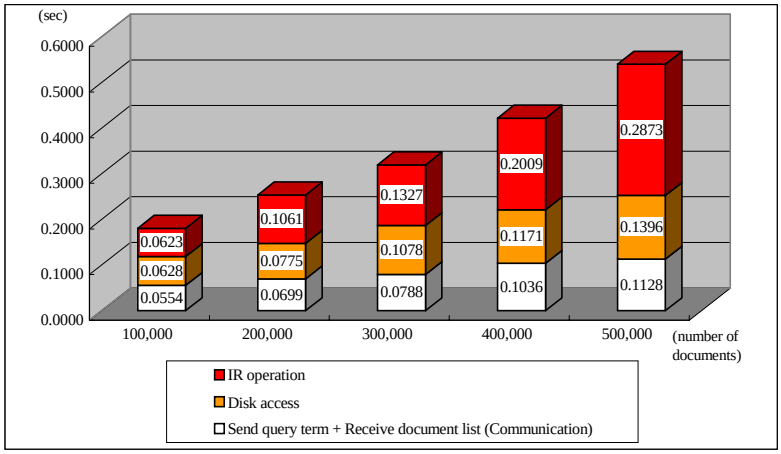


**Fig. 5.** IIF size vs. query processing time

### 3.4 Reducing IR Operation Time

As presented in the previous subsection, the IR operation time turns out to be a new overhead as the IIF size increases. In the IR system, AND/OR operations are performed by the entry node after all the necessary document lists are collected from the processing nodes. However, it is possible to perform AND/OR operations partially to the document lists collected in each processing node. So, each processing node can transmit only the result to the entry node. This helps in reducing not only the IR operation time but also the communication time.

The performance of the improved system in comparison with the original system is shown in Figure 6. In the experiment, the 8-node PC cluster, the greedy declustering method and 500,000 document IIF are used. In the original system, the IR operation takes 0.2873 seconds which is more than 53% of the total query processing time. However in the improved system, the IR operation takes only 0.1035 seconds which is about 35% of the total time. Thus, the IR operation takes less time than the disk access again. The communication time is also reduced from 0.1128 seconds to 0.0500 seconds, and the total time is reduced to almost half when compared with the original system.

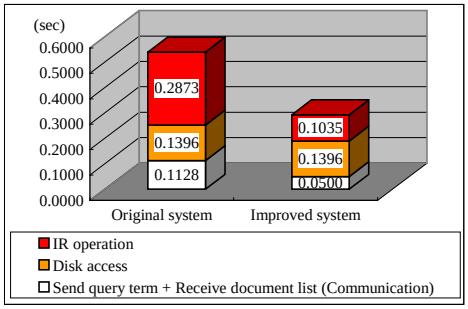


**Fig. 6**. Query processing time with reduced IR operation time

Figure 7 shows the speed-up of the parallel IR system. The maximum speed-up obtained from the 8-node system when compared with the single node system is 4.0. As shown in the figure, the speed-up of the parallel IR system is saturated rapidly from the 4-node system. As the number of the processing nodes in the system increases, the disk access time[1] is reduced because the average number of query terms assigned to each node decreases. However, the IR operation time and the communication time rather increase as the number of document lists transmitted to the entry node increases, and attenuate the overall speed-up. The problem may be alleviated by applying the following idea. Instead of sending all the document lists to the entry nodes, intermediate nodes can be utilized to merge the document lists by performing AND/OR operations in advance as shown in Figure 8. Thus the entry node finally handles only two document lists. This will help in reducing both the IR

[1] The disk access time includes the time spent for partial AND/OR operations in the processing nodes.

operation time and the communication time. Experiments need to be performed to verify the above idea .

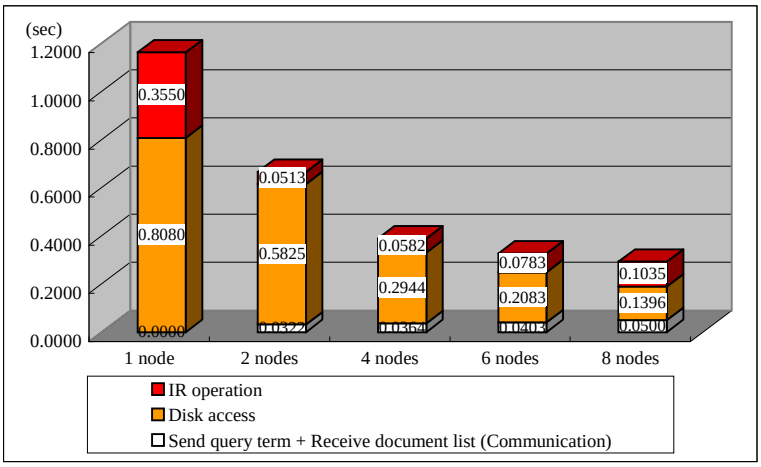


**Fig. 7**. Number of processing nodes vs. query processing time

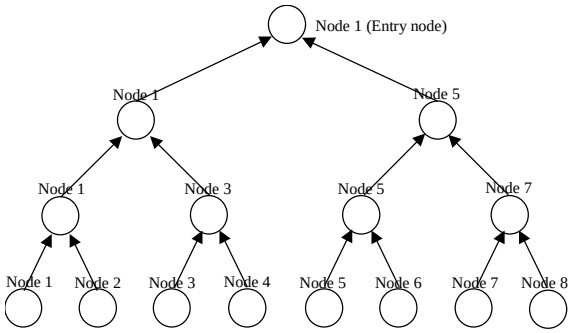


**Fig. 8**. Merging document lists in intermediate nodes

## 4. Conclusions

In this paper, as a cost-effective solution for fast IR service, an SCI-based PC cluster system is proposed. In the parallel IR system developed on the PC cluster, the IIF is partitioned into pieces using a greedy declustering algorithm and distributed to the cluster nodes to be stored on each node's hard disk. For each incoming user's query with multiple terms, terms are sent to the corresponding nodes which contain the relevant pieces of IIF to be evaluated in parallel. The IR system is developed using a DSM programming technique based on SCI. According to the experiments, the IR system outperforms an MPI-based IR system using Fast Ethernet as an interconnect. Speed-up of 4.0 was obtained with the 8-node cluster in processing each query on a

500,000-document IIF.

Currently, the parallel IR system has a single entry node. In the future research, a PC cluster based IR system with multiple entry nodes will be developed. Each processing node in the cluster system can act as an entry node to process multiple users's queries simultaneously. This will help in improving both the IR system's utilization and throughput. With more research effort, we hope this model to be evolved as a practical solution for low-cost high-performance IR service on the Internet.

## References


1. IEEE, "MYRINET: A GIGABIT PER SECOND LOCAL AREA NETWORK", IEEE-Micro, Vol.15, No.1, February 1995, pp.29-36.

2. "Active Messages: a Mechanism for Integrated Communication and Computation", Thorsten von Eicken and David Culler, et. al., 1992.

3. "Fast Messages (FM): Efficient, Portable Communication for Workstation Clusters and Massively-Parallel Processors", IEEE Concurrency, vol. 5, No. 2, April-June 1997, pp. 60-73. (Pakin, Karamcheti & Chien)

4. "U-Net: A User-Level Network Interface for Parallel and Distributed Computing", Anindya Basu, Vineet Buch, Werner Vogels, Thorsten von Eicken, Proceedings of the 15th ACM Symposium on Operating Systems Principles (SOSP), Copper Mountain, Colorado, December 3-6, 1995.

5. http://www.myri.com/GM/doc/gm_toc.html

6. "NUMA-Q: An SCI based Enterprise Server", http://www.sequent.com/products/ highend_srv/sci_wp1.html

7. "SCI Interconnect Chipset and Adapter: Building Large Scale Enterprise Servers with Pentium Pro SHV Nodes", http://www.dg.com/about/html/sci_interconnect_ chipset_and_a.html

8. S.H.Park, H.C.Kwon, "An Improved Relevance Feedback for Korean Information Retrieval System", Proc. of the 16th IASTED International Conf. Applied Informatics, IASTED/ACTA Press, pp.65-68, Garmisch-Partenkirchen, Germany, February 23-25, 1998

9. Salton, G. and Buckley, C., "Improving retrieval performance by relevance feedback", American Society for Information Science, 41, 4, pp. 288-297, 1990.

10. http://www.dolphinics.no/customer/software/linux/index.html

11. "A High-Performance, Portable Implementation of the MPI Message Passing Interface Standard", http://www-unix.mcs.anl.gov/mpi/ mpich/docs.html