PROPOSAL OF SHARED MEMORY ACCESS METHODS FOR LAMBDA COMPUTING ENVIRONMENT

Hirohisa Nakamoto,[1] Ken-ichi Baba,[2] and Masayuki Murata[1]
[1]*Department of Information Networking,Graduate School of Information Science and Technology,Osaka University, Suita, Osaka 565-0871, Japan*
[2]*Cybermedia Center,Osaka University, Ibaraki, Osaka 567-0047, Japan*

Abstract: Optical transmission technology is studied actively in order to realize high-speed transmission and broadband networks. However, conventional packet-based switching technology cannot realize the true high quality communication for each connection. Then we propose new λ computing environment which has the virtual channels utilizing optical fibers connecting computing nodes. So we can offer the high-speed and reliable connection path/pipe which is necessary for SAN and Grid computing, to the users with such virtual channels. In this paper, we propose and evaluate an access method to the virtual ring network when we use it as a shared memory. As a result, we can show the performance of the proposed method to access the shared memory on photonic networks.

1. INTRODUCTION

In recent years, as users of networks such as the internet are increasing, the amount of traffic is increasing steadily. Various applications which utilize images become to be used, and the demand on the technology which enables the high speed and large scale transmission in a network is increasing. In order to satisfy these demands, the research for optical transmission technology has been developed actively. Research of WDM technologies which uses multiplexed light wavelengthes is main target of development and a new research of WDM techonology which can use 1000 wavelengthes is also advanced [1]. In recent years, IP over WDM network is studied and developed in order to provide high-speed transmission on the Internet based on WDM technology. Moreover, standardization of the routing technology of the Internet called GMPLS which is the communication technology using various optical technology for lower layer than WDM technology, is also advanced in IETF [2]. Furthermore, aiming at true IP communication of a photonic network, the research on the optical packet switch based on optical technology can also be begun [3].

However, many of such technologies presuppose the present Internet technology. That is, IP packet is treated as a degree of granule treating information, and it is making into the target of research and development how to carry it with high speed on a network. Therefore, as long as the architecture based on packet switching technology is taken, realization of the high quality communication to each connection is very difficult. New applied technologies such as SAN and Grid computing need to provide an end user with a high speed and reliable communication pipe, for that, a mass wavelength path should be set up between end users and provided for a user. That is, it is possible to provide an end user with a ultra high-speed and high quality communication pipe by building the photonic network which uses established fibers, or newly lays fibers if needed, and by utilizing the wavelength multiplexed in the fiber as the minimum particle size for information exchanges.

As middleware aiming at realization of the high-speed distributed computation environment using the optical network, OptIPuter is proposed [4]. It is studied and developed in order to build the Grid environment established on optical networks. It also provides virtual communication paths, but it bases on the present Internet technology and treats a packet as an informational particle size, so that the problem of packet processing which was mentioned previously arises.

Then we propose a new architedcture λ computing environment which has the virtual channels utilizing optical fibers connecting computing nodes. In the conventional Grid environment, data is exchanged with the message passing using TCP/IP. In λ computing environment, by realizing communication between nodes on Grid by not conventional TCP/IP but establisehed wavelength paths, we can achieve high-speed and hith riliable communication. Then, by making virtual channels on a mesh upon the photonic network which connected the network nodes and the computer nodes with the optical fibers, distributed computation on high speed channel is enabled. Moreover, it is possible to utilize wavelengths as a shared memory by constituting a virtual ring on λ computing environment. As a result, it is not necessary to distinguish the shared memory from a communication channel in a wide-area distributed system, and we expect that the high-speed data exchange between computing nodes is achieved (see Fig. 1).

In this paper, we propose and evaluate an access method to the virtual ring network which is utilized as a shared memory. Specifically we consider using the cache in the CPU and local memory of each computer group as cache of such a shared memory. When using the virtual ring as a shared memory, it is necessary unlike bus between CPU and the shared memory in the computer, to consider restrictions in the timing and the frequency of access, since the shared memory is spread out on a long-distance optical fiber, and to take into consideration the coherency between the shared memory in the virtual ring and the cache of each computer group more than the conventional shared memory system. However, it is not necessary to distinguish the shared memory in a wide-area distributed system from a communication channel, and it seems that the high-speed data exchange between computing nodes is achieved. As mentioned above, in consideration with such features, we propose a shared memory access method for λ computing environment and evaluate the method through simulations.

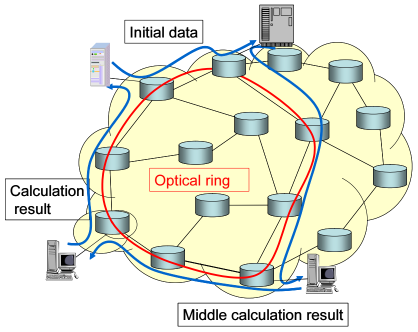

Figure 1. Virtual ring on photonic netowork

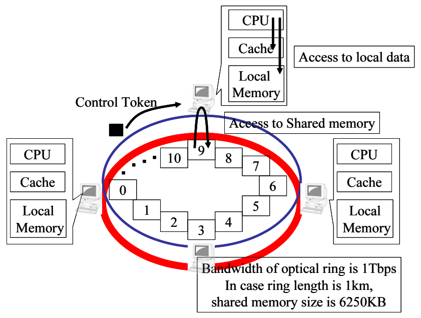

Figure 2. Network model

2. SHARED MEMORY AND ACCESS METHOD IN λ COMPUTING ENVIRONMENT

2.1 Network model on consideration

We show a network model in Fig.2. Computing nodes which compose λ computing environment are connected with optical fibers and those fibers make a ring network virtually. In this paper, I presuppose that each computing node has one CPU, a level 1 cache, and a local memory. A local memory is used for storage of programming codes, and a shared memory is used for storage of the shared data which all computing nodes use in computing.

A optical ring network has a wavelength path for a shared memory, and a wavelength path for control signals. The bandwidth of the wavelength path for the shared memory is set to 10Tbps, and propagation delay time is made into 5ns/m. The processing delay time in middle nodes, such as network devices which constitute a optical ring network, is not taken into consideration here, but it is assumed that it is included to propagation delay time. Therefore, when using a optical ring network as a shared memory, it is equivalent to the capacity of 6250KBytes per 1km of fiber length.

2.2 Shared memory access method

In conventional shared memory system, access to the shared memory from each processor is restricted by contention in a shared bus. Then in order to reduce access to the shared memory, cache is used generally to improve access speed. Usually, every processor has cache system and it is constituted by a level 1 cache, a level 2 cache, a level 3 cache, and many stages. When each processor has cache, it is necessary to fully take into consideration the consistency between the data on cache and on the shared memory.

In λ computing environment, the application program calculates intensively within a computing node, and after synchronous process it exchanges data, so it does not perform data exchange during processing within a computing node. Then we adopt the write back invalidation protocol because the number of write back access to the shared memory is the least of all snoop cache protocols. We condider in restrictions of memory access timing and cache coherency and propose the protocol in Sec. 2.2.1 In order to solve cache coherency problem when two or more computing nodes synchronously update the same data of local cache, the token for control messages is prepared. Moreover, parallel computers use synchronous operation in order to collaborate. As a kind of synchronous operation, there are atomic operation, caching of synchronous variable, waiting on a shared memory, a memory lock, the barrier synchronization method, and so on. Application programs to evaluate a proposed method firstly calculate locally and after local calculation perform synchronous operation. So in this paper, we adopted the barrier synchronization method. Explanation of the barrier synchronization method on a optical ring is shown in Sec. 2.2.2.

2.2.1 Write back invalidation protocol corresponding to a optical ring network. As mentioned above, we adopt write back invalidation protocol to solve cache coherency problem. In adapting write back invalidation protocol to our network model, we must take a care of control message, read miss process and write miss process.

In the conventional method, when a read miss occurs and the other computing node has relevant data, the computing node will send the data to the demanding computing node. However, in the case of the shared memory using the optical ring network, the demanding computing node directly can access to the shared memory. This is because the delay time of direct access to the shared memory is shorter than the time of waiting for transmission of relevant data from another computing node. In write back invalidation protocol, data of local cache has three states of Invalid (I) state, Clean (C) state, and Dirty (D) state. Then, when a line copy demand message comes and relevant data is in C state, each computing nodes does not return response message, but when relevant data is in D state, it returns response message. In this case, it is not necessary to access to a shared memory.

Next, we consider the case of processing for the write out from processor to cache. When data exists in cache in the C state and a processor writes out the data to local cache, the data will be in D state from C state. Then, the invalidation demand message to relevant data is added to the control token, and it is sent out to the wavelength path for control. At this time, other computing nodes with the data of a corresponding address know that the writing out to relevant data occurred, and change the data which they have in the self-cache into I state. Under the present circumstances, if two or more computing nodes write out the data of the same address of C state simultaneously, two or more computing nodes may hold the data of D state. To solve this problem, when a computing node needs to update cache in D state from C state, it firstly has to catch the control token. After catching the control token, it checks that the other computing node has not added the invalidation message to a relevant address. After cheching the control token, it adds the invalidation demand message to the control token, and sends the control token to wavelength for control. After above processing, it can change the cache of relevant data into D state from C state. Moreover, when the control token is caught and the other computing node has already added the invalidation message to a relevant address, relevant cache are changed into I state, and update of cache is yielded to other computing node.

2.2.2 Barrier synchronization. How to realize a barrier synchronization in the shared memory on the optical ring network is explained. First, a part of shared memory is allocated to synchronous memory area. When a synchronous memory is accessed, Fetch&Decrement operation like the conventional method is performed indivisibly. That is, it ensures that access to a synchronous memory indivisibly causes subtraction processing to the relevant data. Since only one computing node can access to the synchronous memory simultaneously, when using a optical ring network for a synchronous memory, execution of an atomic operation is easy. On an application program, in making syncronization among some computing nodes, each node accesses to the synchronous memory. The number of processors is set in that the synchronous memory. The value of the synchrounous memory will be set to 0 if all nodes access. If the value of a synchronous memory is set to 0, all nodes will finish synchronous process and begin to the next processing.

3. PERFORMANCE EVALUATION

In this section, we evaluate the performance of the shared memory access method proposed for the previous section through simulation. We show the results using not only the shared memory on λ computing environment, and but also the conventional TCP for the distributed computation in order to compare them readily. We referred to the ISIS library [5] currently developed at the Amano Laboratory at Keio University in coding simulation program.

3.1 Simulation model

We used the network model as follows. Each computing nodes in λ computing environment is interconnected with optical fibers and configures the ring network virtually. Each computing node has one CPU, a level 1 cache, and a local memory. Clock frequency of CPU is 3 GHz, capacity of a level 1 cache is 512KB, and capacity of local memory is 2GByte. It assumes that the computing node is put on the optical ring network with equal distance. A optical ring network has a wavelength path for shared memory, and a wavelength path for control signals. The bandwidth of the wavelength path for shared memory is set to 10Tbps, and propagation delay time is made into 5 ns/m. The processing delay time in the interface of each computing node and middle nodes, such as network devices which constitute a optical ring network, is not taken into account here, but it is assumed that it is included to propagation delay time. In order to compare, we assume the shared memory system exchanging data according to TCP. This system has one shared memory server and each computing node is connected to shared memory server by Ethernet. The performance of a computing node is the same as that of the case of a photonic shared memory model, and the distance from each computing nodes to a shared memory server is set to 1km. Transmission speed of Ethernet is set to 1Gbps.

In order to evaluate the performance, we use some of the Splash2 benchmark programs, such as the radix sort program which sorts the sequence of an integer value using radix sort algorithm, the product of matrix program which calculates product of $n \times n$ matrix, and the queen problem program which solves n-queen problem.

3.2 The result of radix sort program

At first, we show the number of execution clocks in CPU when we apply a radix sort program to the simulator of the shared memory model in λ computing environment in Fig. 3. The numbers of keys for sorting are 32768, 65536, and 131072. Even if it increases the number of computing nodes in the case of the problem size 32768, the advantage of parallel processing is not taken. This is because the rate of synchronous operation to total operation is large. However, when problem size becomes large like 65536 or 131072, it turns out that the effect of parallel process arises during the number of computing nodes is less than eight. However, the effect of parallel processing becomes weak as the number of nodes increases. We show the number of execution clocks in CPU when we apply a radix sort program on the simulator of TCP message passing model in Fig. 4. Although the same tendency as the shared memory simulator on λ computing environment is shown, the number of execution clocks is larger than the shared memory simulator on λ computing environment as a whole. From this results, when performing distributed processing to sufficient problem size, we found that the shared memory and access method for λ computing environment have advantages.

3.3 The result of product of matrix program

We show the number of execution clocks in CPU when we apply a product of matrix program to the simulator of the shared memory model in λ computing environment in Fig. 5. The matrix sizes are from 32×32 to 256×256. When problem size is small, the advantage of parallel processing is not taken. But when problem size becomes large, the effect of parallel processing is shown during the number of computing nodes is less than eight. Though we cannot show the graph of a TCP model, even if problem size becomes large, the advantage of parallel processing is not taken unlike radix sort program. From this results, when performing distributed processing to sufficient problem size, it is shown that the performance of shared memory method for λ computing environment is better than the method for TCP message passing.

3.4 The result of queen problem program

We show the number of execution clocks in CPU when we apply a queen problem program to the simulator of the shared memory model in λ computing environment in Fig. 6. The problem sizes are from 4×4 to 32×32. In the case of a queen problem, even if problem size becomes large, there is no advantage of parallel processing. This is because the number of synchronous access is larger than other application programs though we cannot show the graph because of lack of space.

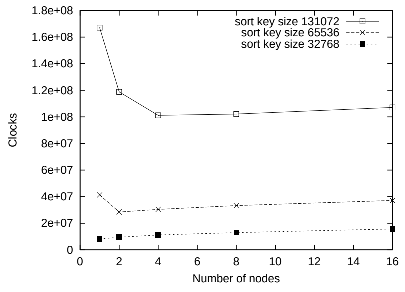

Figure 3. Processing time of radix sort program using photonic shared memory

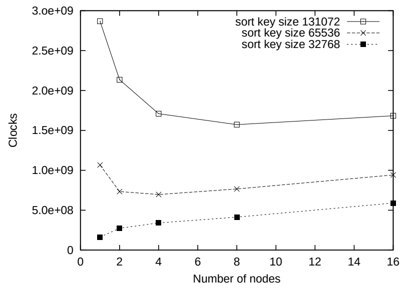

Figure 4. Processing time of radix sort program using TCP message passing

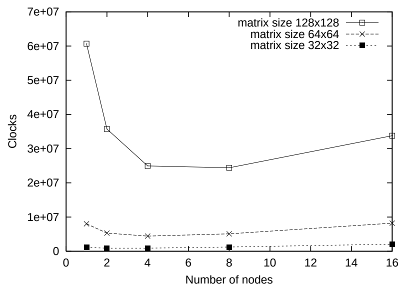

Figure 5. Processing time of product of matrix program using photonic shared memory

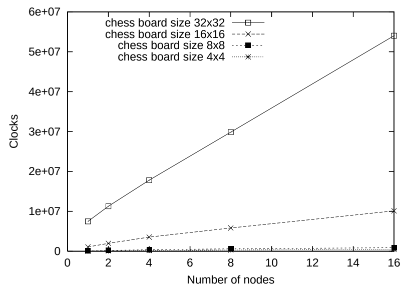

Figure 6. Processing time of queen problem program using photonic shared memory

4. CONCLUDION

In this paper,we proposed the shared memory access method in realizing the shared memory on photonic network. Moreover,we evaluated the performance of the proposed method using the benchmark program for parallel computing. As a result, we can show that the effectiveness of using optical ring as a shared memory and of parallel processing by the increase in the number of nodes when number of synchronous processing is small. An efficient shared memory access method and a practical use of a local memory is due to be considered from now on.

REFERENCES

[1] M. Murata and K. Kitayama, “Ultrafast photonic label switch for asynchronous packets of variable length,” in *IEEE INFOCOM 2002*, June 2002.

[2] E. L. Berger, “Generalized multi-protocol label switching (GMPLS) signaling functional description,” in *IETF RFC3471*, Jan. 2003.

[3] T. Yamaguchi, K. Baba, M. Murata, and K. Kitayama, “Scheduling algorithm with consideration to void space reduction in photonic packet switch,” *IEICE Transactions on Communications*, vol. E86-B, pp. 2310–2318, Aug. 2003.

[4] T. DeFanti, M. Brown, J. Leigh, O. Yu, E. He, J. Mambretti, D. Lillethun, and J. Weinberger, “Optical Switching Middleware for the OptIPuter,” *IEICE Transaction on Communication*, vol. E86-B, Aug. 2003.

[5] M. Wakabayashi and H. Amano, “Environment for multiprocessor simulator development,” *I-SPAN 2000*, pp. 64–71, Dec. 2000.