社団法人 情報処理学会 研究報告
IPSJ SIG Technical Report

2003－MPS－45 (8)
2003／6／24

Performance Evaluation of 3-Dimensional MIN with Cache Consistency Maintenance Mechanism

YASUKI TANABE,† TAKASHI MIDORIKAWA,† DAISUKE SHIRAISHI,†
MASAYOSHI SHIGENO,† TOSHIHIRO HANAWA††
and HIDEHARU AMANO†

In this paper we evaluate the two component architectures for the MIN-connected multiprocessors: the Piled Banyan Switching Fabrics (PBSF) and MIN with Cache consistency mechanism (MINC). The PBSF is a high bandwidth MIN with three dimensional structure. The MINC is a mechanism for controlling the consistency of private cache modules located between processors and the MIN. The simulation result shows that (1) the MINC improves performance and (2) the PBSF with cache provides the sufficient throughput.

Keywords: parallel architectures, multistage interconnection networks, instruction level simulation

1. Introduction

Multistage Interconnection Networks (MINs) have been well researched as an interconnection mechanism between processors and memory modules especially for the middle-scale multiprocessors. However, in the conventional MINs, their complicated structure and pin-limitation problem have been a stumbling block to implement.

To address such problem, we proposed a novel architecture of MINs, which called SSS(*Simple Serial Synchronized*)-MIN in 1992[1]. The SSS-MIN simplifies MIN with the synchronized bit-serial communication: all packets are transferred serially and synchronously. Synchronized bit-serial communication can simplify the structure/control, and also solves the pin-limitation problem. With the simple structure, a highly integrated chip which works with a high frequency clock rate can be utilized.

In the previous works, we proposed two key mechanisms for SSS-MIN: Piled Banyan Switching Fabrics (PBSF)[2] and MIN with Cache Consistency mechanism (MINC)[3]. The PBSF is the network topology, which provides high communication throughput with a three dimensional structure and multiple outlets. The MINC is the cache system, which is installed in the network between the MIN and processors.

For the broader study of SSS-MIN, we developed an instruction level simulator for the MIN connected multiprocessor system. The simulator is built using our parallel simulator development library ISIS[4]. In this paper, we present the design and implementation of the simulator and evaluation results.

In Section 2.1, the concept, structure and control of the SSS-MIN are introduced. In Section 2.2 and 2.3, the architecture of PBSF and MINC is described. In Section 3, the design and the implementation of the simulator are explained. In Section 4, the evaluation results are presented.

2. Component architectures

2.1 SSS-MIN

The basic operation of the SSS-MIN is illustrated in Figure 1. Like the ATM (Asynchronous Transfer Mode)-based packet switching systems for telecommunication, all packets are inserted into the SSS-MIN serially synchronized with a common frame clock. Since each switching element stores only one bit (or a few bits) of the packet, the SSS-MIN behaves like a set of shift registers with a switching capability. After cycles for passing through all stages, the packet headers come out at the output of the MIN.

When a conflict occurs, either one of the packets has to be routed to an incorrect direction because the SSS-MIN provides no packet buffers at each switching element. A conflict bit is set in the routing tag of the misrouted packet so that it is treated as a *dead packet* and never interferes with other packets. Packets treated as a dead packet are reinserted from input buffer again in the next frame.

In order to compensate for the loss of synchronization, techniques called pipelined circuit switching and stage hopping are introduced. The details are shown in[1].

2.2 Piled Banyan Switching Fabrics

The packet collision creates a large perfor-

† Keio University
†† Tokyo University of Technology

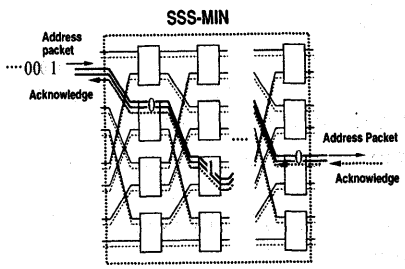


Fig. 1 Structure of the SSS-MIN

mance overhead, because the packet which routed to incorrect destination has to be reinserted again in the next frame. To address this problem, we proposed the Piled Banyan Switching Fabrics(PBSF)[2].

In the PBSF, banyan networks are connected in the three dimensional direction (Figure 2). Each switching element provides four inputs/outputs(two for horizontal, and two for vertical direction). Packets are inserted into the highest layer of banyan network, and transferred to the horizontal direction. When two packets collide, one of them descends to the corresponding switching element in the next lower layer with one clock delay. The descending packet may collide again with packets that are already in the next layer. In this case, one of them is sent further down to the next lower layer network.

If the collision were happened at the bottom layer, the descending packet is simply discarded. As already explained, the dead/discarded packets are re-inserted to the network in the next frame.

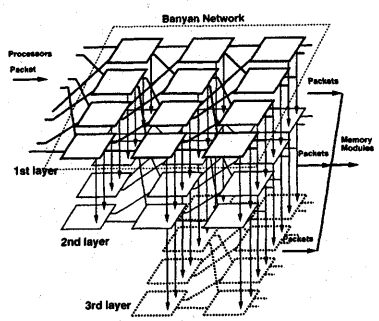


Fig. 2 Piled Banyan Switching Fabrics

2.3 MINC

2.3.1 The directory management method

To reduce latency and congestion in the MIN, a cache system is indispensable between the processors and MIN.

We proposed a cache consistency maintenance hardware for MIN called MINC (MIN with Cache Consistency mechanism)[3].

Unlike several networks which combine MINs and caches[5][6], the key idea of the MINC is a cache directory scheme called the RHBD[7], which was proposed for a massively parallel processor JUMP-1[8]. This technique can be easily applied to the MIN because of its embedded tree structure.

2.3.1.1 RHBD scheme

In this scheme, the bit-map of the hierarchical directory is equipped only in the main memory module and reduced using two techniques.

- Using the common bit-map for all nodes of the same level of hierarchy (tree)
- Sending a message to all children of the node (thus, broadcasting) when the corresponding bit in the map is set.

The reduced directory is not stored in each hierarchy but stored only in the root. Message is multicasted according to the reduced bit-map attached to the message header. Using this method, a message is quickly transfered since multicast does not require to access the directory in each hierarchy level.

3. Instruction level simulator and its model

3.1 An instruction level simulator

Although various performance evaluation results are presented for MIN, most of them are based on probabilistic assumptions with some exceptions[5][6]. However, in order to evaluate performance of cache consistency mechanism MINC, instruction level simulation is required.

We developed an instruction level simulator for MINs by using a simulator development library called "ISIS"[4]. ISIS is an architecture independent simulation kit for multiprocessors. It includes various small simulators called "Units" such as processors, buses, memories, and I/O devices. The simulator supports execution of MIPS executable binaries.

We developed new ISIS models for the PBSF and MINC. Also for a comparison, we developed a traditional wormhole MIN model.

3.2 Simulation model: SNAIL-2

Our instruction level simulator is based on the architecture of SNAIL-2. SNAIL-2[9] is the second prototype of an SSS-MIN multiprocessor. It was developed mainly for evaluating the PBSF and the MINC with practical applications.

As shown in Figure 3, the PBSF connects the processing units with the shared memory modules.

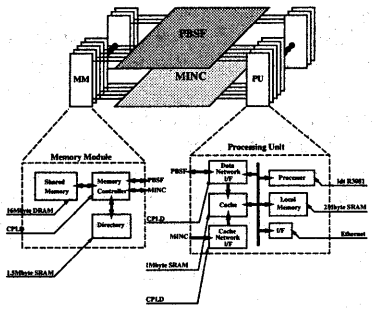


Fig. 3 The structure of the SNAIL–2

4. Evaluation with instruction level simulation

Table 1 shows the default parameters used in a simulation.

Table 1 Environment

Number of PUs	1 - 64
Cache	
size	256KB / PU
Number of ways	2-way
line size	32 byte
Data Trans. Network	
Number of Layers	2-layer
Frame Clock	40 clock
Link Width(PU->MM)	16 bit
Link Width(MM->PU)	8 bit
C.Coherent Network	
Switching Element I/O	4×4
Pruning cache	512 entry × 2 way / Sw

For evaluation, we implemented four applications: Radix, FFT, LU and Ocean from SPLASH-II benchmark suits[10].

- Radix is a parallel radix sorting program. Since the shared data is small, it does not require frequent data exchange or synchronization. 131072 items are sorted.
- FFT is a parallel fast $\sqrt{n}$ basis Fourier's transform with 6-step algorithm. The data exchange is minimized among the processors. The size is set to be 2^{16}.
- LU is a parallel LU decomposition program for a 192 × 192 matrix.
- Ocean is an ocean tide simulation program based eddy and boundary currents. Since a large data is shared among the processors, it requires the frequent data exchange. The target grid size is set to 130 × 130.

4.1 Performance of SNAIL-2

Figure 4 shows the performance versus size of SNAIL-2. The result is normalized to that with 1 PU for each application. As seen in the figure, the performance of the SNAIL-2 is improved even with 64 PUs in FFT, LU and Ocean, whereas it is degraded in Radix.

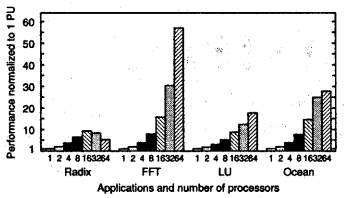


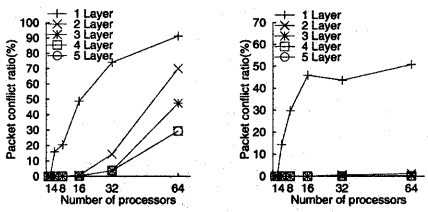
Fig. 4 Speed up ratio

4.2 Performance evaluation of PBSF

4.2.1 Number of layers of PBSF

Figure 5 (a) and (b) compare the ratio of conflicting packets when LU is executed with or without the cache.

The performance is much improved for adding the first extra layer. It is saturated, however, after the second additional layer. Considering the required hardware and output pins, the PBSF with 2 layers is the optimal in this situation.



(a)LU w/o cache (b)LU w cache

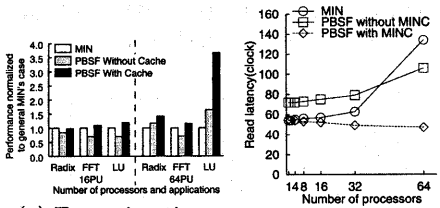
Fig. 5 PBSF packet conflict ratio

4.2.2 Comparison with traditional MINs

To investigate the advantages and disadvantages of the PBSF, we also simulated a traditional wormhole MIN. Figure 6 (a) compares the results of the both networks. The performance is normalized to a system with 1 PU. Note that, for the traditional MIN, the MINC cache coherent control could not be applied. So the PBSF SSS-MIN is evaluated with/without cache, while only the traditional MIN without cache is evaluated.

In the system with 16 PUs, the performance of the PBSF SSS-MIN without cache is less than that of the traditional MIN. This is mainly for the overhead of packets synchronization and serialization. Meantime, the performance of PBSF SSS-MIN with cache is almost the same as those of traditional MINs. In the system with 64 PUs, the PBSF SSS-MIN out performs the traditional MIN in performance even without cache.

Figure 6 (b) compares the read latency of each



(a) Execution time (b) Read Latency (Radix)

Fig. 6 PBSF compared with general MIN

network in Radix. This shows the read latency is directly influences to the performance. In the SSS-MIN, extra wait cycles are needed for all packets to synchronize with a common frame clock. This extra cycle makes the latency of the network longer.

4.3 Performance improvement by using cache

In this discussion, we focus on the cache performance of the systems with 16 PUs and 64 PUs. Figure 7 shows the performance gain ratio of the system with cache. As seen in the figure, the cache improves the execution time at most 130% and at least 10%.

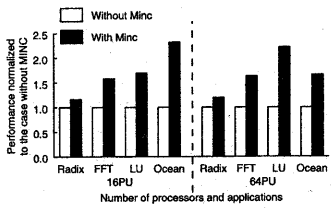


Fig. 7 Cache effect on performance

The cache reduces the number of accesses to shared memory and lightens the network load. As a result, the packet conflict rate is decreased. This effect greatly contributes to the network performance because the packet conflict adds a large extra latency to the access time – as already explained, the SSS-MIN re-inserts the conflicting packets in the next frame.

5. Conclusion

We evaluated the PBSF and MINC with an instruction level simulator; the PBSF and MINC are the two important component architectures for the MIN-connected multiprocessors.

2-Layer PBSF topology achieves a high through-put even with the large scale system as 64PUs. In the low congestion network however, the SSS-MIN has poorer performance than the traditional wormhole MIN because of the larger network latency.

The MINC technique allows a system to use a coherent shared data cache in the network between MIN and processor. The shared data cache decreases the congestion of data transfer network thereby much improves the network performance. The MINC network achieves a satisfactory performance on the cache coherent packet transaction when the system size is small.

References

1) K.Gaye H.Amano, L.Zhou. Sss(simple serial synchronized) - min: a novel multi stage interconnection architecture for multiprocessors. In *Proc, of the International Conference on Parallel Processing*, Vol. I, pp. 571–577, 1992.
2) Y.Fujikawa T.Hanawa, H.Amano. Multistage interconnection networks with multiple outlets. In *Proc, of the International Conference on Parallel Processing*, Vol. I, pp. 1–8, 1994.
3) T.Hanawa, T.Kamei, H.Yasukawa, K.Nishimura, and H.Amano. Minc: Multipstage interconnection network with cache control mechanism. *IEICE Trans. on Information and Systems*, Vol. E80-D, pp. 863–870, 1997.
4) H.Amano M.Wakabayashi. Environment of multiprocessor simulator developemnt. In *Proc. of International Synposium on Parallel Architectures Algorithms and Networks*, pp. 64–71, 2000.
5) R.Iyer and L.Bhuyan. Design and evaluation of a switch cache architecture for cc-numa multiprocessors. *IEEE Trans. on Comput.*, Vol. 49, No. 8, pp. 779–797, 2000.
6) Ravi R. Iyer and Laxmi N. Bhuyan. Using switch directories to speed up cache-to-cache transfers in cc-numa multiprocessors. In *Proc. of Int'l Parallel and Distributed Processing Symposium*, pp. 721–728, 2000.
7) T.Kudoh, H.Amano, T.Matsumoto, K.Hiraki, Y.Yang, K.Nishimura, K.Yoshimura, and Y.Fukushima. Hierarchical bit-map directory schemes on the rdt interconnection network for a massively parallel processor jump-1. In *Proc. of International Conference on Parallel Processing*, Vol. I, pp. 186–193, 1995.
8) H.Tanaka. The massively parallel processing system jump-1, 1996.
9) T.Midorikawa, D.Shiraishi, M.Shigeno, Y.Tanabe, T.Hanawa, and H.Amano. Snail-2: a sss-min connected multiprocessor with cache coherent mechanism. In *Proc. of Parallel and Distributed Computing, Applications and Technologies*, pp. 17–24, 2002.
10) S.C.Woo, M.Ohara, E.Torrie, J.P.Singh, and A.Gupta. The splash-2 programs: Characterization and methodological considerations. *Proceedings of the 22nd International Symposium on Computer Architecture*, pp. 24–36, 1995.