TR-393

Preliminary Evaluation of the connection network for the Multi-PSI System

by
K. Masuda, H. Ishizuka, H. Iwayama(Mitsubishi),
K. Taki and E. Sugino

June, 1988

ICOT

Mita Kokusai Bldg. 21F
4-28 Mita 1-Chome
Minato-ku Tokyo 108 Japan

(03) 456-3191～5
Telex ICOT J32964

Institute for New Generation Computer Technology

Preliminary Evaluation of the connection network for the Multi-PSI system

Kanae MASUDA, Hirokazu ISHIZUKA, Hiroaki IWAYAMA
Mitsubishi Electric Corporation,
325 Kamimachiya, Kamakura-shi, Kanagawa 247 JAPAN,

Kazuo TAKI, Eiji SUGINO
Institute for New Generation Computer Technology(ICOT),
4-28, Mita 1-chome, Minato-ku, Tokyo 108 JAPAN,

Abstract

The Multi-PSI system has been developed in the Fifth Generation Computer Systems(FGCS) project in Japan. It is an experimental parallel machine mainly used for the parallel software research.

Network-connected multiprocessor systems like the Multi-PSI have such a nature that interprocessor communication costs much higher than intra-processor processing, which affects the system performance. However, measurements of the communication cost for a realisitic multiprocessor system have rarely been seen.

This paper describes the network system of the Multi-PSI-V1 system and measurement of the interprocessor communication cost for the network system.

1 Introduction

The Parallel Inference Machine(PIM) is one of the most important research themes of Japan's FGCS project. In the first three-year stage of the project, PIM research sought to develop basic technologies of machine architecture and parallel execution mechanisms, and accumulated several methods for constructing machine hardware and parallel execution methods of logic programs [2]. But several serious problems of parallel software were revealed by the research, and we recognized that parallel software research has to be done along with parallel architecture research.

We think it is important to prepare a research and development environment for parallel programs first in which programs can be executed in parallel and developed and evaluated efficiently. So we plan two systems ; the first is Multi-PSI-V1(version1) focusing on ease of realization, and the second is Multi-PSI-V2 improving machine scale, functions, and speed.

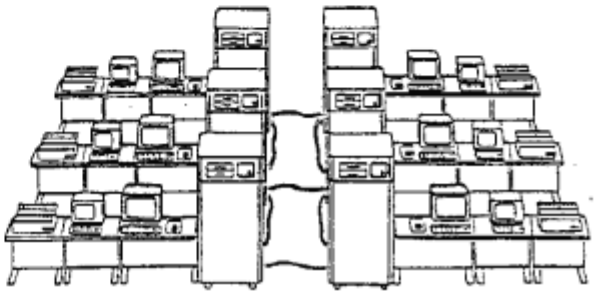


Figure 1: Multi-PSI-V1 system

(1)Multi-PSI-V1

Multi-PSI-V1 contains 6 to 8 PSI machines connected by a dedicated lattice network ; each PSI has its own I/O devices (see figure 1) [6]. The purpose is to implement a prototype of KL1(Kernel Language version 1) language processing system experimentally in a short period [7]. The parallel logic programming language KL1 is based on GHC and implemented in ESP, the sequential system description language of PSI [1] [8].

(2)Multi-PSI-V2

A smaller and faster version of the PSI machine(PSI-II) without I/O is used for the PEs, and 16 to 64 PEs are connected with a new version of the network, which is improved in speed and functions [5]. One PSI machine is used as a front-end processor. The KL1 language execution system is written in firmware to realize high execution speed. The PIMOS(Parallel Inference Machine Operating System) is implemented, and the dynamic load balancing method, parallel algorithms, and

large-scale parallel application programs are studied.

Network-connected multiprocessor systems like the Multi-PSI have such a nature that inter-PE communication costs much higher than intra-PE processing. This fact strongly affects the system performance when the inter-PE communication rate increases. It is important to know the inter-PE communication cost for the researches of load balancing and parallel algorithms. However, measurements of the communication cost for a realistic multiprocessor system have rarely been seen.

This paper describes the network system of the Multi-PSI-V1 system and measurement of the inter-PE communication cost for the network system coupled with KL1 language processing system, a realistic parallel language processor.

2 Connection Network of Multi-PSI-V1 System

2.1 Overview of the System

The PEs are PSI machines, 6 to 8 PEs being connected on a dedicated lattice network. There is no shared memory. Inter-PE communications are performed by exchanging message packets. A user does not handle special message primitives, but simply writes goals with pragmas and unifications. The pragma is an annotation which allows the programmer to specify explicitly how the goals should be assigned to the processors. Inter-PE communication messages are automatically generated by the language execution system. There are two major message types; one concerns KL1 goal management, such as goal sending and termination reporting; and the other concerns unification across the PEs. Each PE has a different address space, and when a reference pointer is passed from one PE to another PE, the internal address of the source PE is converted to a global identifier(ID) and then sent [3]. The address translation table between internal address and global ID is maintained in each PE.

Figure 2 shows the configuration of Multi-PSI-V1. The underlined sections have been newly developed, and the other sections represent the PSI machine system itself.

2.2 Connection Network Hardware

We think that the network-connected structure is essential for large scale multiprocessor systems. We also think that a network, in which logical inter-PE distance is not uniform like mesh, hyper cube, etc., is important for very large scale systems. From

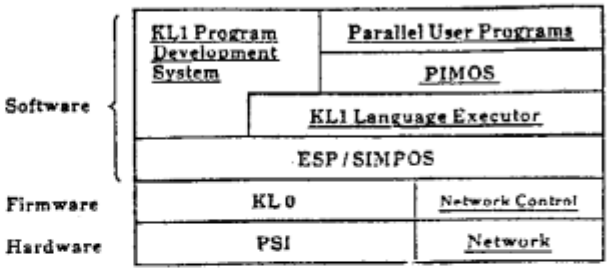


Figure 2: Configuration of the Multi-PSI-V1

this reason, we chose mesh network for the Multi-PSI to study load balancing method and parallel algorithms which can be applied to very large scale multiprocessor systems.

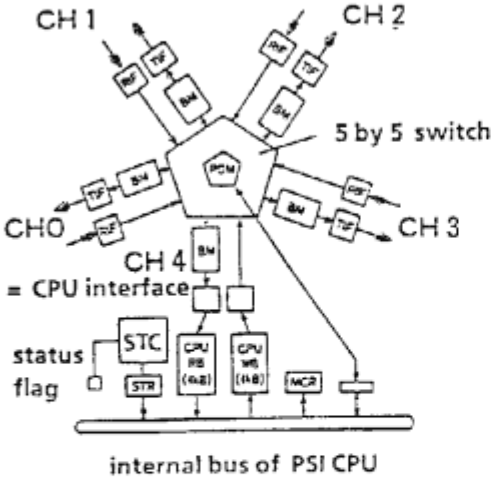
Figure 3 shows the block diagram of the network hardware. It is installed in the option slots of CPU internal bus, and four cables are used to connect adjacent four PEs. Each connection of PE to PE is called a channel. A channel contains two signal sets, one is for transmission and another is for receiving. Data is transferred in 10-bit parallel containing a parity bit. The transmission rate is approximately 500k bytes/sec for each direction.

Message packets are constructed by software. Each packet has a destination PE number in the packet header. The network recognizes the destination PE number, and if equal to its own, the network takes the packet into its receiving buffer. If the PE number is different, the network looks up the path table to get the channel number to which the packet should be re-transmitted. There are simple routing methods which avoid this network deadlock. One involves passing the packet to the horizontal direction prior to the vertical direction when the packet should be passed to the PE on the diagonal direction.

3 Evaluation of the Connection Network System

3.1 Objectives

The inter-PE communication cost contains network transfer cost and cost of pre- and post-processing for the network transfer. The former is affected by the inter-PE distance, the latter is not. These pre- and post-processing contain inter-PE goal management, address translation, and data format conversion for the network transfer. These processing cannot be



PDM :path table
TIF :transmitting interface
RIF :receiving interface
BM :buffer memory
CPUWB:CPU write buffer
CPURB :CPU read buffer
MCR :mode control register
STR :status register

Figure 3: Block diagram of the Network Hardware

removed from the execution system of an network-connected multiprocessor system. It is very important to know the inter-PE communication cost and its components not only for parallel software research like load balancing and parallel algorithms but also for network design and tuning of parallel language processing system.

The objective of the evaluation is to show the inter-PE communication cost and its components for the Multi-PSI-V1 system, a realistic parallel processing system of the logic programming language KL1. The other objective is to make an estimation of those costs for the Multi-PSI-V2, more powerful and practical system than Multi-PSI-V1. We expect that the measuments will give an order of the cost value which can be used practically in the study of load balancing, parallel algorithms, etc.

Multi-PSI-V1 consists of the KL1 language processing system and the connection network system. The hierarchical structure of the connection network system is as shown in figure 4.

(1)The KL1 Processing System

This is a language processing system written in ESP that executes and controls KL1 programs.

(2)Network Handler

This takes the messages from the language processing system, formats them into byte sequences, and passes them to the handler kernel, also performing the corresponding reverse conversion processes. It also handles the KL1-related goal management for transmission and reception, and address translation. It is written in ESP.

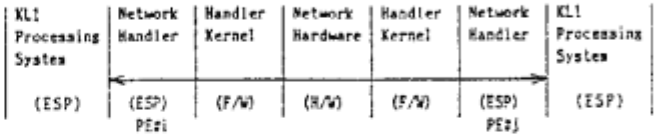


Figure 4: Layered structure of the Network System

(3)Handler Kernel (Network Support Firmware)

This takes the byte sequences from the network handler, and performs conversion in both directions between byte sequences and network packets. The handler kernel is implemented as firmware in KL0 evaluable predicates.

3.2 Measurement Methods

3.2.1 Basic concept about the Measurements

We measured the various data and assessed them in terms of the following two main items to evaluate the connection network system [4]:

(1) Comparison of intra-PE and inter-PE processing performances (in terms of the processing time for a single reduction). We found the time taken to perform one reduction on a single PE, and the time to perform it on two PEs, and discussed the macro performance of the connection network system.

(2) Analysis of dynamic characteristics of each layer of the connection network. More detailed analysis of the data from (1) above, yields a quantitative estimate of the comparative cost (in time) of using the different layers of the connection network system.

Here, we chose to take the time to perform one reduction on a two-PE multi-PSI. In other words, we defined the comparative intra-PE and inter-PE performance as the ratio of the time taken to perform the reduction for a given goal on a single PE to the time taken to perform the reduction for the same goal on the other PE after sending it to the other PE and receiving the reply back. This is an

```
test3_1a(0) :- true | true.
test3_1a(N) :- N1 := N-1 | wait3_1a(R,N1),
               call(bench_mark@t3_1a,R,_).
wait3_1a(success,N) :- true | test3_1a(N).
t3_1a :- true | alloc(5)@@tt3a(x).
tt3a(X) :- true | X=atom.
```

Figure 5: An example of the bench-mark program(unify test)

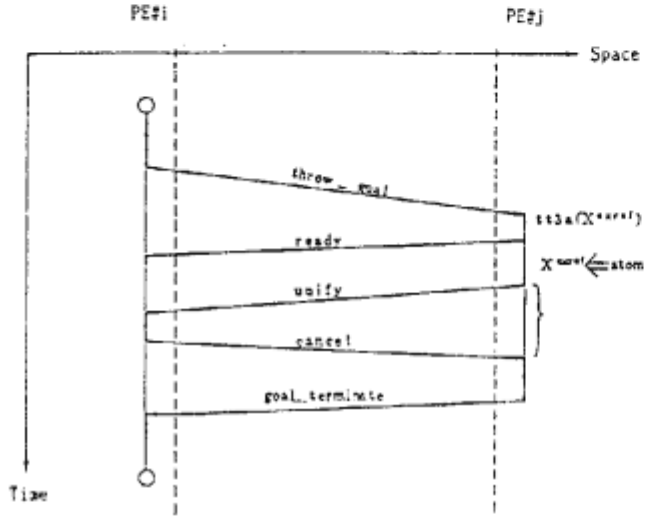


Figure 6: An example of the message exchange

extremely important parameter when writing KL1 programs, in deciding how much the rate of inter-PE communication is allowed against the intra-PE processing. A number of benchmark programs of the type shown in figure 5 were prepared, and execution time measured (see figure 6). We took the total of thirteen orders for inter-PE communications ("throw_goal","unify","read",etc.) and measured the cost for varying types of arguments and numbers of executions of them. We timed them by looping each program 100 times, taking the average time to reduce the variance of the timer supported by the operating system.

3.2.2 Communication cost for Different Layers

(1) Handler communication cost (Ha) can be derived by taking the time of execution (L) found in accordance with the principles given in 3.2.1, and deducting from this the cost of the handler kernel (F) and of the network hardware (H):

```
Ha = L - F - H
```

where F and H are as described in items (2) and (3) below.

(2) Handler kernel cost (F) is measured by the GEVC counter, which counts the execution time for evaluable predicates such as "mpsi_write_buffer" and "mpsi_read_buffer",etc., that either read data from or write it to the interface registers one byte at a time.

(3) Network hardware communication cost (H) is derived from analysis of hardware operation.

3.3 Measurement Results and Consideration

Consider the first item, the comparison between the intra-PE and inter-PE performances. First selecting particularly important items like "throw_goal","unify" and "read" from among the inter-PE communication orders, we investigated the communication costs.

Figure 7 shows the results of running different benchmark programs with different argument types (atom,integer,undefined variable,structure,list) and different argument numbers(1,2,3,4,5).

The vertical axis of the graph is the average time taken for the one execution of 100 loops, and shows both the time taken when the goal is assigned to a different PE (with pragma) and when the goal is handled within the PE (without pragma). The difference between the time (A) with pragma and (B) without pragma can be taken as an approximation to the communication cost. It is important to note that reductions are of two types: those that have been optimized, and those that have not. In our benchmark programs, those without pragmas have been optimized, but those with pragmas have not (see figure 8 for Rop and Rn). When calculating the communication cost, it is not enough to know the difference in execution time; account must also be taken of the difference in the reduction time. In fact, therefore, the communication cost (L) is given by the following equation:

```
L = execution time A (with pragma)
    - execution time B (without pragma)
    - (reduction time Rn
          - reduction time Rop)
```

Note that the optimized reduction time Rop ($1.12 msec$) and the non-optimized time Rn

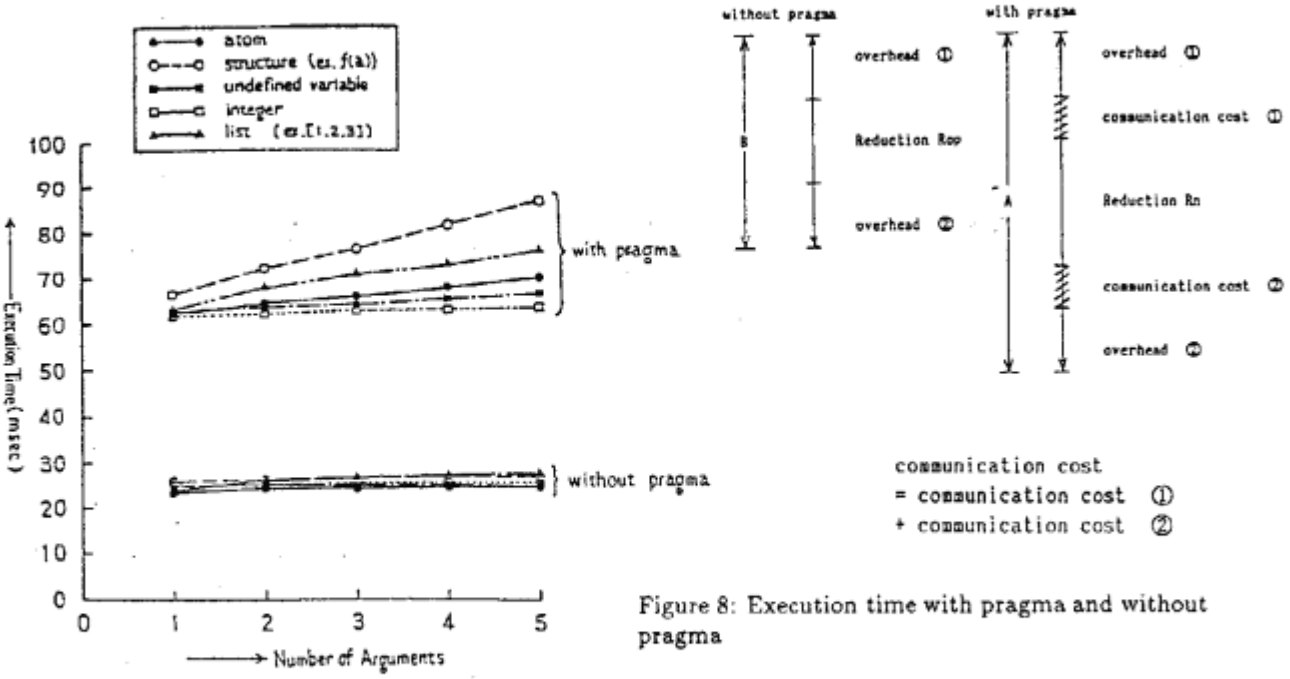


Figure 7: Execution time for predicate argument numbers(throw_goal)

Figure 8: Execution time with pragma and without pragma

(2.25*msec*) were measured using a different benchmark program.

Again, the ratio of the inter-PE communication processing time to the intra-PE processing time is given by the following equation:

```
Processing time ratio = ( L + Rn ) / (Rn)
```

Table 1 gives the ratios, R, derived from the processing times (A,B) with and without pragma for different types and numbers of arguments.

It is clear from this table that, although there is some scatter in the processing time ratio, R, it is generally about 20:1. This indicates that it takes about 20 times longer to perform the reduction for the same goal by addressing another PE and waiting for the reply than by performing it within the same PE. It means that the inter-PE communication consumes much more processing time than a goal reduction within a PE, and the communication degrades the system performance. The volume of inter-PE communition processing should be reduced by these corresponding amount.

Now let us consider the second item, of determining the communication cost for each level of the network. In this respect, the size of the packet for orders ("throw_goal","read","unify",etc.) is fixed for each type of argument, enabling calculation of communication cost for firmware (the handler kernel) and hardware, and this in turn enables the handler communication cost to be calculated (see table 1).

Table 1 shows that the ratios of the hardware cost and firmware cost are much lower than the handler communication cost, approximately in the ratio of 1:3:500. It means that the communication bottleneck is not the hardware message transmission but the software message processing in this system.

Finally, let us extrapolate these results to Multi-PSI-V2. The speed of the handler is expected to increase by about 100-fold, and the network hardware by about 5-fold. The processing time ratio for the Multi-PSI-V1,

```
Ratio = (communication cost
       + cost of one reduction)
         / (cost of one reduction)
```

will change into the following for the Multi-PSI-V2:

```
Ratio = (handler cost/100
   + firmware cost + hardware cost/5
   + one reduction cost/100)
      / (one reduction cost/100)
```

Take the case at the head of table 1. This calculation yields a value that increases from 17.8 to 28-fold.

This implies that the inter-PE communication cost for the V2 system will be higher than the V1 system, or rather that the volume of inter-PE processing in the V2 system should be reduced by the corresponding amount.

Table 1: Results of execution time with pragma and without pragma

(msec)

Measurement Items (Communication Orders)	Arguments No	Arguments Types	Results A	Results B	Communication Cost (Ha+F+H) L	F/W Cost F	H/W Cost H	Handler cost Ha	Processing Time Ratio R
throw_goal (including ready_goal _terminates)	1	atom	62.72	23.90	37.69	0.23	0.07	37.39	17.8
		undef	62.61	24.36	37.12	0.23	0.07	36.82	17.5
		integer	61.96	24.76	36.07	0.22	0.07	35.78	17.0
		structure	66.62	25.41	40.08	0.24	0.07	39.77	18.8
		list(1)	63.10	23.40	38.57	0.24	0.07	38.26	18.1
		list(2)	64.33	24.11	39.09	0.24	0.07	38.78	18.4
		list(5)	66.80	25.39	40.28	0.25	0.08	39.94	18.9
	2	atom	64.90	24.17	39.60	0.24	0.08	39.28	18.6
		undef	63.61	24.60	37.88	0.25	0.08	37.55	17.8
		integer	62.51	25.11	36.27	0.23	0.07	35.97	17.1
		structure	72.42	26.06	45.23	0.26	0.08	44.89	21.1
		list(1)	68.49	26.44	40.92	0.25	0.08	40.58	19.2
	3	atom	66.62	24.11	41.38	0.26	0.08	41.04	19.4
		undef	64.78	24.57	39.08	0.27	0.08	38.73	18.4
		integer	63.27	25.09	37.05	0.23	0.07	36.75	17.5
		structure	76.92	26.24	49.55	0.28	0.09	49.18	23.0
		list(1)	71.53	26.66	43.74	0.28	0.09	43.37	20.4
	4	atom	68.64	24.19	43.32	0.27	0.09	42.96	20.3
		undef	66.12	24.68	40.31	0.28	0.09	39.94	18.9
		integer	63.42	25.16	37.13	0.24	0.07	36.82	17.5
		structure	82.01	26.54	54.34	0.30	0.10	53.94	25.2
		list(1)	73.83	27.01	45.69	0.30	0.10	45.29	21.3
	5	atom	70.62	24.13	45.36	0.29	0.09	44.98	21.2
		undef	67.11	24.69	41.29	0.30	0.09	40.90	19.4
		integer	64.05	25.13	37.79	0.24	0.08	37.47	17.8
		structure	87.13	26.73	59.27	0.33	0.10	58.84	27.3
		list(1)	76.70	27.18	48.39	0.33	0.10	47.96	22.5
unify (including cancel)	1	atom	87.99	28.34	22.89	0.14	0.04	22.71	—
		undef	88.48	28.78	22.94	0.14	0.04	22.76	
		integer	88.99	29.21	23.02	0.13	0.03	22.86	
		structure	89.46	29.66	23.22	0.15	0.04	23.03	
		list(1)	87.18	27.53	22.89	0.15	0.04	22.70	
read (including read_answer)	1	atom	101.44	70.45	30.99	0.12	0.04	30.83	—
		undef	—	—	—			—	
		integer	100.31	71.10	29.21	0.11	0.03	29.07	
		structure	106.13	72.03	34.10	0.13	0.04	33.93	
		list(1)	101.11	69.15	31.96	0.13	0.04	31.79	

*) A : with pragma
B : without pragma
L : A-B+1.12-2.25
F : Firmware Cost
H : Hardware Cost
Ha : Ha = L+F+H
R : (L+2.25)/2.25

4 Conclusion

It was shown that the inter-PE processing costs around twenty times more expensive than the intra-PE processing and the communication bottleneck is not the hardware message transmission but the software message processing in this system. Network-connected multiproccessor systems like the Multi-PSI may have these characteristics in general. In such case, these measurements can be applied to give a guideline of the maximum inter-PE communicaion rate which does not reduce the system performance much.

It was also shown that the communication frequency on the Multi-PSI-V2(high performance and more practical model) would have to be maintained lower than the V1 system.

The Multi-PSI-V2 system is under development, on which the dynamic load balancing method, parallel algorithms, and large-scale parallel application programs will be studied.

Acknowledgments

The authors would like to thank Dr.Shunichi Uchida and other members of Fourth Laboratory of ICOT for their valuable suggestions.

References

[1] T. Chikayama. Unique features of ESP. In *Proceedings of FGCS '84*, 1984.

[2] S. Goto and S. Uchida. *Toward a High Performance Parallel Inference Machine - The Intermediate Stage Plan of PIM-*. ICOTTR TR-201, ICOT, 1986.

[3] N. Ichiyoshi, et al. A Distributed Implementation of Flat GHC on the Multi-PSI. In *Proceedings of the 4th International Conference on Logic Progrmming*, 1987.

[4] K. Masuda, et al. Network Evaluation of the Multi-PSI system. In *Proceedings of the 34th Conference of IPSJ(in Japanese)*, 1987.

[5] H. Nakashima, et al. Hardware Architecture of the Sequential Inference Machine:PSI-II. In *Proceedings of 4th Symposium on Logic Programming*, 1987.

[6] K. Taki, et al. Hardware design and implementation of the personal sequential inference machine(PSI). In *Proceedings of FGCS '84*, 1984.

[7] K. Taki. The parallel software research and development tool : Multi-PSI system. In *Proceedings of France-Japan Artificial Intelligence and Computer Science Symposium 86*, October 1986.

[8] K. Ueda. *Guarded Horn Clauses*. ICOTTR TR-103, ICOT, 1985.