Massively Parallel Processing Project of the Japanse Ministry of Education

Hidehiko TANAKA
Department of Electrical Engineering
The University of Tokyo*tanaka@mtl.t.u-tokyo.ac.jp

Abstract

Massively Parallel Processing Project started in 1992 as a Priority Area of Researh for the Ministry of Education in Japan. The objective of this research project is to establish the basic technology of massively parallel processing which is expected to be the fundamental tool to develop the high-level technologies of 21 century. The main goal of this project is to build up a system prototype of massively parallel processing system. This paper describes the organization of this project and discusses the research results up to this time.

1 Introduction

Information processing technology of which representative is computer is now undispensable tool for the research and development of science and technology. Computers are used as simulators to replace some experiments and as important tools to make the design work efficient. However, the requirments from the research and development of new technologies grows more and more in terms of processing power and memory capacity. For such requirements, single processor systems can not afford. Massively parallel processing is the key to solve this situation. Though parallel processing has a long history, massively parallel processing systems with reasonable high performance came out as commercial systems since only these 2 years. The technologies such as compiler, programming environment and operating systems have not developed yet in full scale, though preliminary software systems are available now.

Massively parallel processing project started in April 1992 as a priority area of research for the Ministry of Education of Japan to research and develop such fundamental technology of massively parallel processing. The member of this project are university researchers. We formed 5 groups of researchers and have been making research to build a sytem prototype of massively parallel processing, ie. system which can run massively parallel applications. This paper surveys this project, shows some research results, and discusses the future planning.

2 Objectives and Planning

The major components of massively parallel processing systems are applications, programming paradigm, programming languages, operating system and hardware architecture.

The objectives of our massively parallel processing project is to establish the basic technology for the creation of next generation supercomputers, which are expected to be general purpose simulators for the re-

[1] 7-3-1 Hongo, Bunkyo-ku, Tokyo 113, Japan

0-8186-7038-X/95 $4.00

search and development of high level science and technology. The characteristic point of our research is to make research of all necessary technologies from applications to hardware in an unified manner. That is, the requirement analysis of all elemental technologies is also included in our activity. Though this means that we need to do so many things at one time, we believe that we need to come across all kinds of experiences from the beginning in order to view reasonable far research target.

One way of basic research is to do freely all the elemental research topics in this field. However, we didn't take this approach, but took the way of building one system prototype under the cooperation of all members. Accordingly, our activity is concentrated around the design and implementation of a prototype massively parallel processing system called Jump-1.

3 Research Group Organization

We divided our activity into 4 groups, application development, language system development, operating system development and hardware development. We formed another group, management group which has the role of planning and integration of the 4 groups. Accordingly, the whole organization of this project is of 5 groups as showon in the table 1.

Table 1: **Group Organization**

Group Name	Number of Researchers
Management group	18
A group: Applications	7
B group: Description System	10
C group: Operating System	8
D group: Hardware System	11

Though the research activity is basically within the group which each member belongs to, several task forces are formed to discuss such items that span among a few groups. Examples are input-output system task force, security mechanism task force, language interface task force, memory management task force, simulator task force, etc. Working groups are formed corresponding with every specific research item in each group, ie. language design working group, OS design working group, system IO working group, connecting network working group, etc.

4 Applications

The objectives of application group is to develop a parallel computational modelling method which can be used to organize a simulator through mapping many kinds of problems directly to processors of a massively parallel computer. This group is developing a visualization technology of computed results and a set of benchmark applications for the evaluation of parallel processing.

The titles of research themes of this group are as follows.

1. Parallel Finite Element Method Based on Substructure Method

   This method is implemented on the Fujitsu AP1000 machine. For a problem of two-dimentional Poisson differential equation, the speed up gain of 27.9 was obtained on 64 processors machine.

2. Massively Parallel Cooperated Processing through Self-organizing Agents

   This is a research of self-organization of autonomous agents on massively parallel and dis-

tributed computational system.

3. Parallel Processing of Binary Decision Diagram on Distributed Computer Environment

   Data representaion by BDD is known to be efficient for processing. This research is a trial of parallelization of BDD. Parallel gain of 129 was attained by using AP1000 of 512 processor elements.

4. Design of Optimum Parallel Algorithm with the Communication Delay Taken into Account

   Fibo-Net which is an expansion of Fibonacci Cubes is introduced.

5. Design of Massively Parallel Machine for Computer Graphics

   In this theme, Massively Parallel Algorithms for Computer Graphics is designed for AP1000. The speed up of 200 is attained by using the AP1000 of 256 processor elements. The final target is to realize a real time processing system for computer graphics.

6. Evaluation of Parallel Processing

   Functional emulator DMSTEG is developed which can change the parameters of parallel machine such as connection topology, communication delay, shared memory scheme, synchronization method between adjacent processors and among all processors, etc.

7. Massively Parallel Processing by Irregular Field Model

   MGCG method is proposed for linear partial differential equations and evaluated on AP1000 for its parallel implementation. Speed up of 70-90 times is gained by using 100 processing elements machine.

5 Languages

The research objective of language group is to design a few languages for the description of programs which are executed on parallel machines, and to develop the language processors.

For the designing of parallel languages, we need to support 2 features. One is easy description of problems. The another is high speed execution of the programs. Moreover, the languges should be practical and have the descriptive power to express complicated process structure of MIMD execution. Accordingly, we divided our activities into 2 kinds of languages design. One is the design of language NCX. The other is the research of a set of more future oriented languages, of which representative is language V. NCX is a practical language derived from C and oriented toward data-parallel execution. Language V is a language for MIMD execution, which is derived from dataflow programming language.

Language NCX

1. Exection model: based on SIMD semantics. For MIMD machines, the compiler generates MIMD execution code which synchronizes among parallel activities through barrier at minimun number of points which are extracted by analysing part of the compiler.

2. Data structure: Field concept is introduced. Field is a set of virtual processors to which data set is mapped by one to one correspondence. Connecting topology can be given at the time of field definition. As the basic topology, we provided the mesh, binarytree and hypercube. Data processing is described for one virtual processor element.

```
/* Matrix Multiplication in NCX */
#include<stdio.h>  /* standard I/O */
#define N 128 /* size of matrix */
field matrix(N,N) on mesh;
float a,b,c on matrix;
main() in mono {
    float tmp on mono;
    in matrix(i,j) {
            a = 1.0 + N * i + j; /* initialze */
            b = 1.0 / a;         /* initialize */
            spawn(k:N)
                    c = (+= a@(i,k) * b@(k,j));
            tmp = (+= c);
     }
     printf("Check sum is %13.6e|n", tmp);
}
```

Figure 1: **Program Example in NCX**

3. Definition of variables and reference: Variables are defined by attaching the field name. Referencing of each variable is done by the pair of variable name and field index.

4. Functions: Function calls are done at the same time by all active virtual processors. Field information can be attached as one of the argument which can be used for the access to the field data in the body of function.

5. Input and Output: Beside the basic io, parallel io is supported which executes io directly from the field.

Figure 1 shows an example of program written in NCX.

Laguage V

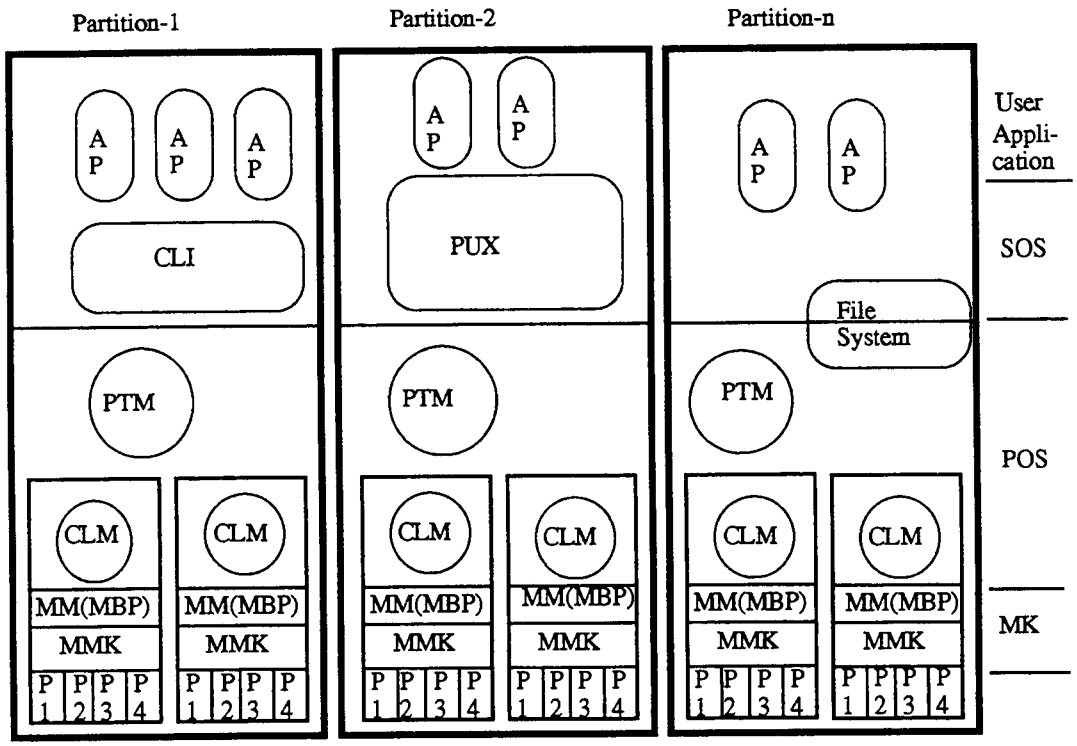
Language V is an experimental one for the research of massively parallel programming paradigm. The features are to pursue the natural MIMD semantics, description of object parallelism without consciousness of grain size, merging of data parallelism and object parallelism.

6 Operating Systems

The research objectives of operating system group is to develop a prototype of operating system COS for massively parallel computers, Jump-1 in particular.

Design Principles of COS are that it should be flexible, autonomous and fault tolerant OS with efficient security protection mechanism. The software architecture of COS is shown in figure 2.

Partition OS(POS) is the manager of hardware resource of which management unit is "partition". POS is made of 3 modules, Meta Micro Kernel(MMK), Cluster Manager(CLM) and Partition Manager(PTM). A MMK resides in each processing element and controls interruption, scheduling, inter-thread/ process communication and memory management. A CLM resides in each cluster which is the implementation unit of hardware and controls the cluster unique functions such as io control. A PTM resides in each partition and con-



P1:Processing Element 1, MMK: Meta Micro Kernel, MK:Micro Kernel
MM(MBP):MBP Based Memory Management, CLM:Cluster Manager,
PTM:Partition Manager, PUX:Parallel Unix, CLI:Common Language Interface,
POS:Partition OS, SOS:Service OS

Figure 2: **Architecture of COS**

trols communication between partitions, and dynamic expansion and shrinkage of partitions.

One of the unique features of this micor kernel is that it realizes a unified protection mechanism of memory access, communication and synchronization through utilizing the single address space of distributed shared memory as the capability. As the distributed shared memory of Jump-1 is enhanced by a set of Translation Lookaside Buffer, the single address space can be implemented with reasonable high speed. This solves the problems of conventional micro kernel in an elegant way. That is, though the access speed of conventional micro kernel such as Mach is tolerable for distributed processing, it is too slow to use for the access mechanism among parallel processors.

Service OS(SOS) realizes the service of system functions such as partition allocation, activation and stop of programs, file system, user interface, etc. COS has a few kinds of SOSes. At present, we are implementing 2 kinds of SOS as follows.

- PUX: PUX supports UNIX compatible interface and massively parallel oriented services to programmer. Former supports UNIX process management, UNIX file system and UNIX network

functions. Latter supports partition, massively parallel io and distributed shared memory management.

- Common Language Interfaces(CLI): CLI is made of runtime mechanism which helps the development of language processors of massively parallel machines and the abstraction of massively parallel hardware, and a set of libraries which are provided for the easy usage of hardware.

We started our implementation using Mach OS as the base. The elemental parts are being rewritten one by one according to the requirements of massively parallel processing. One example is the security mechanism of micro kernel.

7 Hardware Architecture

The research objectives of hardware architecture group is to design and implement a prototype of massively parallel computer. For the design of massively parallel computer, the implementation technology must play a crucial role. Accordingly, we made our decision that we would implement a real hardware of reasonable size. Through the experimental hardware implementation, we expect that we can get the insight of design point for the massively parallel computers.

The basic archtecture of our massively parallel computer which is called JUMP-1 is as follows.

1. processor: In distributed shared memory architecture, the handling of fine grain parallel processing is important to get the high performance. So, the basic architecture of Jump-1 is the parallel processing of mixed grain size: course grain process is processed by conventional RISC processor (Supersparc+) and fine grain is by a special processor called Memory Based Processor (MBP). The architecture of MBP is tuned to the non-local processing such as inter-processor communication and synchronization. MBP complements the characteristics of main processor that it is efficinet for local processing such as register operations but not for non-local processing.

2. Secondary cache memory system: It is required to realize an efficient distributed shared memory protocol, a direct word-level operation for synchronization in the shared memory space, and a message communications/thread operations through shared memory. The cache mechanism of Jump-1 have 3 features to support this requirement: the snoop mechanism is implemented within a cluster and directory system for inter-cluster, the prefetch and injection mechanism for high hit-rate of cache access, and the support mechanism of communication /synchronization by I-structure and FIFO for the high speed communication/synchronizaion between processors.

3. Interconnecting network: As the load of network traffic increases for the shared memory architecture, the design of topology is important. High efficient multicast capability is indispensable for the efficient shared memory. Moreover, the ability of network that the connecting network can be divided into a few subnetwork physically and dynamically is necessary to cope with the multi-user/job requirement. We designed a new topology called RDT(Rectangular Diagonal Torus) of which diameter is short enough, to fulfil these requirements.

4. IO subsystem: Our IO subsystem is a persistent

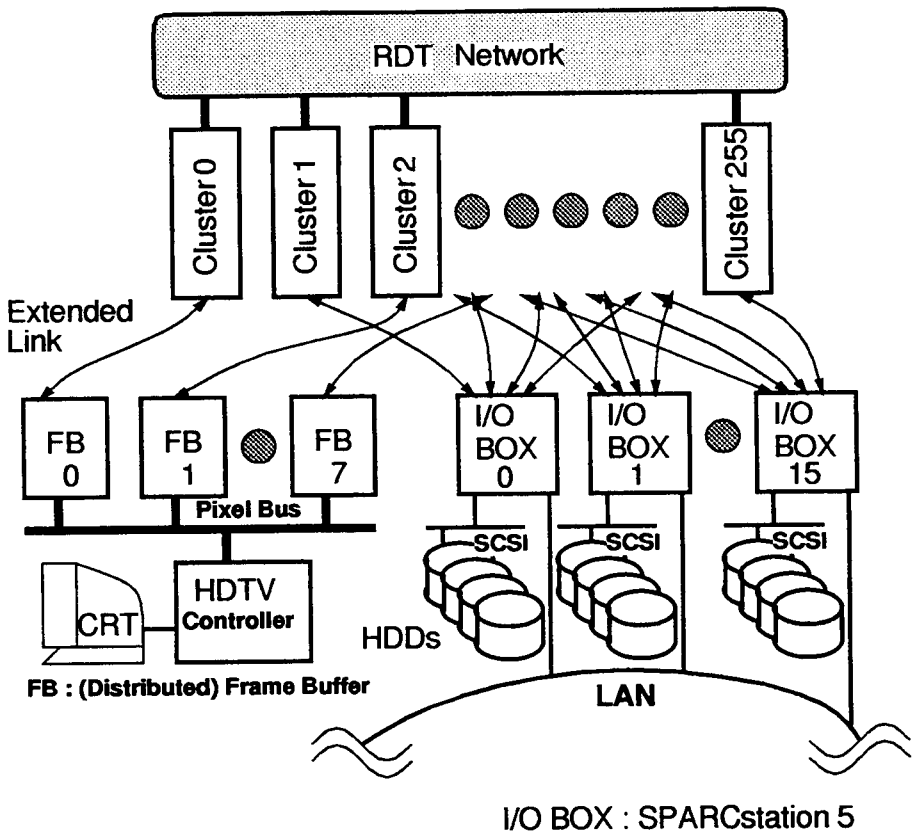


Figure 3: **Global Image of Jump-1 System**

object through shared memory. Main memory is expanded toward the internal memory of IO subsystem which is connected through a high speed serial link. All of these memories make up a distributed shared memory. IO peripherals are connected to the outside of this distributed shared memory.

Figure 3 shows the global image of Jump-1 system.

8 Present Status and Future Plan

The detail design of Jump-1 is finished. The special chips such as cache controller, memory based processor and network switch are now at the logic design stage. Some preliminary evaluation is done through simulation. So far as the COS is concerned, the implementation is going and tested on conventional Sun workstations of multi-processors. The third version of the specification of language NCX is issued and implemented on AP1000 machines. The compiler is now at the stage of improvement. Several application programs are gathered and analysed on AP1000.

We are expecting that we will be able to demonstrate the Jump-1 system within the next year. After that, we will refine and evaluate our system using a few more years and present the results to the public.

Beside that, we are planning to connect high speed communication line of ATM to Jump-1 and to try the integration of high speed communication line and high

speed massively parallel machine.

9 Conclusion

The massively parallel processing project as a priority area of research in Ministry of Education of Japan is presented. This project is of Universities and major research objective is to build a system prototype of massively parallel computer from applications to hardware. In this paper, some aspects of this prototype is presented such as applications, languages, operating system and hardware. The grant of this project is Priority Area of Research (1) Number 04235101. We are gratefull to the generous support of many manufacturers for the promotion of this project: Sun mcrosystems, Fujitsu, Toshiba and Hitachi.

References

[1] Proceedings of Symposium for Massively Parallel Processing, Priority Area of Research of Ministry of Education, First: Sept. 1992, Second: March 1993, Third: Sept. 1993, Fourth: March 1994 and Fifth: Sept.1994.