ICOT Technical Report: TR-264

TR-264

Parallel Inference System Researches in the FGCS Project

by

T. Chikayama

June, 1987

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

Authors must enclose a signed copy of this form when submitting manuscripts

IEEE COPYRIGHT FORM

Signed form, appropriately completed, MUST ACCOMPANY any paper to be published by IEEE.
PLEASE READ REVERSE OF THIS FORM FOR FURTHER DETAILS.

TITLE OF PAPER: Takashi Chikayama 1176
Ins New Gen Comp Tech LIC87
Mita Kokusai 21F
AUTHOR(S): 1-4-28 Mita Minato-ku
108 Tokyo, JAPAN
PUBLICATION TIT...
MONTH AND YEAR:

RETURN FORM TO:

IEEE COMPUTER SOCIETY
1730 Massachusetts Avenue N.W.
Washington, D.C. 20036-1903

PART A — COPYRIGHT TRANSFER FORM

(NOTE: Company or other forms may not be substituted for this form. U.S. Government employees whose work is not subject to copyright may so certify by signing Part B below.)

The undersigned, desiring to publish the above paper in a publication of the IEEE or cosponsored by the IEEE, hereby transfer their copyrights in the above paper to The Institute of Electrical and Electronics Engineers, Inc., known as the IEEE.

Returned Rights: In return for these rights, the IEEE hereby grants to the above authors, and the employers for whom the work was performed, royalty-free permission to:

1. Retain all proprietary rights other than copyright, such as patent rights.
2. Reuse all or portions of the above paper in other works.
3. Reproduce, or have reproduced, the above paper for the author's personal use or for company use provided that (a) the source and IEEE copyright are indicated, (b) the copies are not used in a way that implies IEEE endorsement of a product or service of an employer, and (c) the copies per se are not offered for sale.
4. Make limited distribution of all or portions of the above paper prior to publication.
5. In the case of work performed under U.S. Government contract, the IEEE grants the U.S. Government royalty-free permission to reproduce all or portions of the above paper, and to authorize others to do so, for U.S. Government purposes.

IEEE Obligations: In exercising its rights under copyright, the IEEE will make all reasonable efforts to act in the interests of the authors and employers as well as in its own interest. In particular, the IEEE requires that:

1. The consent of the first-named author be sought as a condition in granting republication permission to others.
2. The consent of the undersigned employer be obtained as a condition in granting permission to others to reuse all or portions of the paper for promotion or marketing purposes.

In the event the above paper is not accepted and published by the IEEE or is withdrawn by the author(s) before acceptance by the IEEE, this agreement becomes null and void.

T. Chikayama
AUTHORIZED SIGNATURE

TITLE IF NOT AUTHOR

EMPLOYER FOR WHOM WORK WAS PERFORMED

June 2, '87
DATE FORM SIGNED

PART B — U.S. GOVERNMENT EMPLOYEE CERTIFICATION

(NOTE: If your work was performed under Government contract but you are not a Government employee, sign transfer form above and see item 5 under Returned Rights.)

This will certify that all authors of the above paper are employees of the U.S. Government and performed this work as part of their employment and that the paper is therefore not subject to U.S. copyright protection.

AUTHORIZED SIGNATURE

TITLE IF NOT AUTHOR

NAME OF GOVERNMENT ORGANIZATION

DATE FORM SIGNED

IEEE COPYRIGHT FORM
(continued)

Information to Authors

IEEE POLICY

Thank you for your interest in the IEEE. As you know, the IEEE distributes its technical publications throughout the world, and it is necessary to translate and abstract its publications, and articles contained therein, for inclusion in various compendiums and similar publications, etc. When an article is submitted for publication by the IEEE, the IEEE understands that its acceptance of the article implies that the IEEE has the rights to do all of the things it normally does with such an article.

In connection with its publishing activities, it is the policy of the IEEE to own the copyrights in its technical publications, and to the contributions contained therein, in order to protect the interests of the IEEE, its authors and their employers, and at the same time to facilitate the appropriate re-use of this material by others.

The new United States copyright law requires that the transfer of copyrights in each contribution from the author to the IEEE be confirmed in writing. It is therefore necessary that you execute either Part A–Copyright Transfer Form or Part B–U.S. Government Employee Certification on the reverse side of this sheet and return it to the Editor (or person who supplied this sheet) as promptly as possible.

AUTHOR/COMPANY RIGHTS

If you are employed and you prepared your paper as a part of your job, the rights to your paper initially rest with your employer. In that case, when you sign the copyright transfer form, we assume you are authorized to do so by your employer and that your employer has consented to all of the terms and conditions of this form. If not, it should be signed by someone so authorized.

> NOTE RE RETURNED RIGHTS: Just as the IEEE now requires a signed copyright transfer form in order to do "business as usual," it is the intent of this form to return rights to the author and employer so that they, too, may do "business as usual." If further clarification is required, please contact: Ms. Patricia Penick, Manager, Publication Administrative Services, IEEE, 345 E. 47th St., New York, NY 10017 (212) 644-7550.

JOINT AUTHORSHIP

For jointly authored papers, only one signature is required, but we assume all authors have been advised and have consented to the terms of this form.

U.S. GOVERNMENT EMPLOYEES

Authors who are U.S. Government employees are not required to sign the Copyright Transfer form (Part A), but any co-authors outside the Government are.

Part B of the form is to be used instead of Part A only if all authors are U.S. Government employees and prepared the paper as part of their job.

> NOTE RE GOVERNMENT CONTRACT WORK: Authors whose work was performed under a U.S. Government contract but who are not Government employees are required to sign Part A–Copyright Transfer Form. However, item 5 of the form returns reproduction rights to the U.S. Government when required even though the IEEE copyright policy is in effect with respect to the reuse of material by the general public.

Thank you for your prompt cooperation.

Parallel Inference System Researches in the FGCS Project

Takashi Chikayama

ICOT

Abstract

In the fifth generation computer systems (FGCS) project of Japan, *logic programming* and *parallel processing* are adopted as the principle of both software and hardware system development. Their amalgamation, *parallel inference systems* are being investigated in ICOT.

The target of the hardware development effort is a parallel inference machine (PIM) with about 10^3 element processors. A smaller scale PIM with about 10^2 processors is planned in the intermediate stage of the project. Another system called multi-PSI is planned, which connects up to 64 personal sequential inference machines (PSI) relatively loosely for providing parallel software development environment. Both of these inference machines have the same kernel language (KL1) based on flat GHC as the machine language.

As a software project, an operating system for parallel inference machines named PIMOS (parallel inference machine operating system) is under development. Both of the parallel inference machines are planned to have the same operating system.

1 Introduction

The parallel inference system research in ICOT is based on the 10-year plan of the Japanese fifth generation computer systems (FGCS) project. The project started from April 1982 and consists of the initial stage (1982–84), the intermediate stage (1985–88) and the final stage (1989–91), ten years in total.

The final target of the project is to build a prototype of a system called the knowledge information processing system (KIPS). One of the principal functions of KIPS is a highly parallel inference feature[1]. The target of the parallel inference system research is to develop a highly parallel inference machine with about 10^3 processing elements and with more than 10^5 LIPS inference speed.

Besides the development effort for the inference hardware itself, systems for supporting parallel software development are also developed. The personal sequential inference machine PSI is one of such systems. In the intermediate stage, the multi-PSI system which connects up to 64 PSI relatively loosely is planned.

[1] Another principal function is the knowledge-base feature.

Both PIM and multi-PSI will have basically the same machine language called KL1. KL1 is based on flat GHC[11] with additional features to make it self-contained, allowing *everything* including the whole operating system to be written in the language.

For operating these parallel inference machines, a system called PIMOS (parallel inference machine operating system) is being investigated. It will be written completely in the KL1 language without using extra-logical features.

2 PIM

PIM developed in the final stage is planned to have about 10^3 processors to attain 10^8 to 10^9 LIPS. In the intermediate stage, a PIM with about 10^2 processing elements is planned[4]. Its processing elements have 200 to 500 KLIPS of inference speed; the total system is expected to attain more than 10^7 LIPS including the hardware communication overhead and the software operating system overhead.

The PIM planned in the intermediate stage consists of *clusters* connected with the inter-cluster packet switching network. Each cluster consists of around 8 processing elements, a communication controller and a shared memory, all connected on a common bus. Each processing element has a cache of the shared memory. The coherency of the cache will be kept by a rather complicated special protocol[8].

From the preliminary investigation by simulation, the common bus will be the performance bottleneck if a naive implementation scheme is used. A specialized garbage collection scheme for KL1[2] is expected to lower the cache miss hit ratio, lowering the buss traffic.

3 Multi-PSI

Multi-PSI machines consist of *sequential* inference machines connected loosely by specialized network hardware. As the processing elements are originally designed to be stand-alone processors, no hardware level optimization is made for parallel processing. The main objectives of the

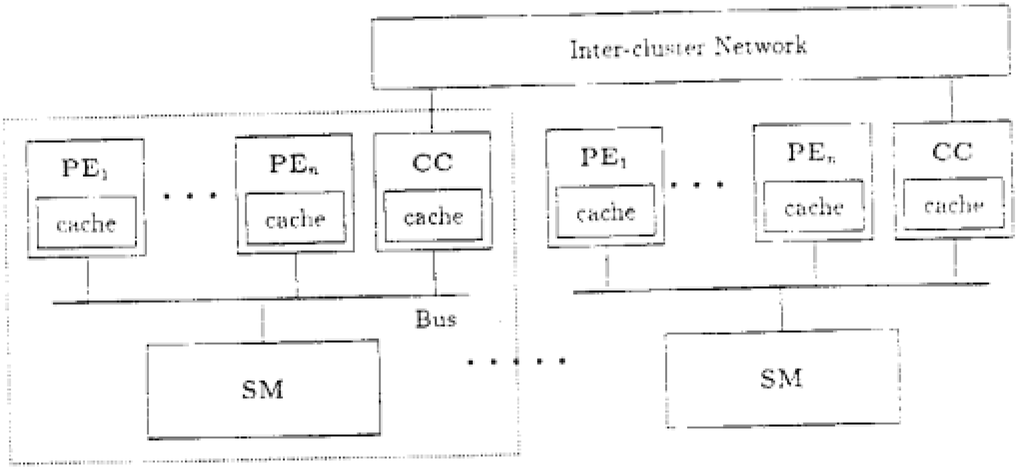

Figure 1: Overview of PIM

Multi-PSI systems is to establish the parallel execution mechanism for KL1 and to support parallel software development.

There are two versions of Multi-PSI machines.

3.1 Multi-PSI V1

Multi-PSI V1[10] is based on the older version of the PSI machine. 6 PSI machines are connected. The objective of this system is mainly to investigate parallel execution algorithm for the KL1 language[5]. KL1 programs are compiled into ESP[1], which is based on sequential Prolog, and is executed with run-time support system written also in ESP.

In this system, lower level support for GHC execution is only for speeding up network communication. As the ESP language is not most suitable for implementing a virtual machine, the performance of the system is rather limited. However, numerous statistics could be collected using the system, that influenced the design of its successors very much. The system started working April 1986 and numerous data are collected on it.

3.2 Multi-PSI V2

Multi-PSI V2 consists of up to 64 processing elements of the newer, more compact version of PSI, called PSI-II[7]. Unlike Multi-PSI V1, KL1 programs are translated to machine code (KL1B[6]) and executed by the firmware. The whole set of firmware for parallel programming language execution will be written. The performance of the system is expected to be much higher than the V1 system: More than 100 KRPS ave. and 200 KRPS max. by a single processor. The hardware of the system will be ready in 1987, and the total system is expected to begin functioning in 1988.

4 Languages

GHC[11] was chosen as the base of the KL1 language for parallel inference systems. The GHC language inherits many of its features from Concurrent Prolog[9] and Parlog[3], and belongs to the so-called committed choice parallel logic programming language family. Unlike Prolog or Prolog-based parallel logic programming languages, committed choice languages are powerful enough to express *everything*, and GHC is the simplest of such languages. Because of implementation feasibility and efficiency, a *flat* version of GHC was chosen, which does not require complicated management of the AND-tree structure.

The KL1 language has several extensions to the original GHC. One of the most essential extensions is the notion of *sho-en*. *Sho-en*, or *manor* in English, is similar to the meta-call mechanism seen in other committed choice languages. Like the meta-call, the *sho-en* mechanism can be used to protect the outside of *sho-en* from failure inside a *sho-en*. In addition, limits of computational resources (execution time, memory, etc.) consumed in a *sho-en* can be controlled from outside. This feature is indispensable to write an operating system in KL1.

5 Operating System

An operating system for parallel inference machines called PIMOS is being developed. Its target machine is temporarily the Multi-PSI V2, but it will also be used on PIM's.

PIMOS will have the following characteristics.

Logic-Based: PIMOS will be described completely in KL1, without using extra-logical features at all. Even I/O devices connected to the inference machine have *logical* interface[2].

[2]Logical I/O devices will be simulated by I/O front processors in

Integrated: Although PIMOS is an operating system for parallel machines, it is an integrated operating system working as one unit, rather than consisting of many operating systems distributed on processing elements. Basically, PIMOS itself is not aware of the processor boundaries. This principle is considered to be more efficient than the distributed approach for controlling large scale programs that require frequent inter-processor communication.

Practical: Although PIMOS *is* an experimental system, its purpose is not limited to show the feasibility of logic based operating systems. It is for providing a practical programming environment for parallel algorithm development.

6 Concluding Remarks

In the past, researches in parallel computer hardware have been relatively independent from parallel software researches. Basically, the hardware or system implementation researches were for implementing more efficient environment for executing *already existing* software.

The principle of the parallel inference systems development in ICOT is rather different in this point: Software and hardware researches should be combined more closely. Software or even algorithms optimized for sequential machines may not be optimal for parallel machines. Thus, software should change when hardware changes.

However, there is a chicken-and-egg problem: Without parallel hardware, practical parallel software cannot be developed; without parallel software, it is hard to know what kind of parallel hardware is appropriate. ICOT's approach to solve this problem is a stepwise bootstrapping. The first step is to settle a (temporary) software-hardware interface, namely the KL1 language, and implement it (as the multi-PSI system). The next step is to develop various software systems on it (including PIMOS). By running the resultant software, many unknown parameters of the behavior of parallel software will be revealed. The next generation of the hardware (PIM) will be based on these experiences, and software is developed on this machine. Optimal software on the new machine might have still different parameters, which is reflected on the next generation hardware, and so on.

Acknowledgement

This paper is based on various documents and personal communication provided by many researchers of ICOT and researchers working on various projects related to ICOT, too numerous to be listed here.

the first version. If KL1 should be going to be one of the dominant languages for describing operating systems of parallel machines, it may become *standard* device interface protocol.

References

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

[2] T. Chikayama and Y. Kimura. Multiple reference management in flat GHC. In *Proceedings of 4th International Conference on Logic Programming*, 1987.

[3] K. L. Clark and S. Gregory. *Parlog: A Parallel Logic Programming Language*. Research Report TR-83-5, Imperial College, 1983.

[4] A. Goto and S. Uchida. *Toward a High Performance Parallel Inference Machine — The Intermediate Stage Plan of PIM —*. ICOT Technical Report TR-201, ICOT, 1986.

[5] N. Ichiyoshi, T. Miyazaki, and K. Taki. A distributed implementation of flat GHC on the multi-PSI. In *Proceedings of 4th International Conference on Logic Programming*, 1987.

[6] Y. Kimura and T. Chikayama. An abstract KL1 machine and its instruction set. In *Proceedings of 4th International Conference on Logic Programming*, 1987.

[7] H. Nakashima and K. Nakajima. Hardware architecture of the sequential inference machine PSI-II. In *Proceedings of 4th Symposium on Logic Programming*, 1987.

[8] M. Sato et al. KL1 execution model for PIM cluster with shared memory. In *Proceedings of 4th International Conference on Logic Programming*, 1987.

[9] E. Y. Shapiro. *A Subset of Concurrent Prolog and Its Interpreter*. ICOT Technical Report TR-003, ICOT, 1983.

[10] K. Taki. *The Parallel Software Research and Development Tool: Multi-PSI System*. ICOT Technical Report TR-237, ICOT, 1986.

[11] K. Ueda. *Guarded Horn Clauses: A Parallel Logic Programming Language with the Concept of a Guard*. ICOT Technical Report TR-208, ICOT, 1986.