A compact cluster computer with embedded CPUs and its application to rapid prototyping of fingerprint verification system

Yoshifumi Sasaki[1a)], **Koichi Ito**[2], **Takafumi Aoki**[2], **and Tatsuo Higuchi**[3]

[1] *Faculty of Science and Engineering, Ishinomaki Senshu University*
1 Shinmito, Minamisakai, Ishinomaki, Miyagi 986–8580, Japan
[2] *Graduate School of Information Sciences, Tohoku University*
05 Aoba, Aramaki-aza, Aoba-ku, Sendai, Miyagi 980–8579, Japan
[3] *Faculty of Engineering, Tohoku Institute of Technology*
35–1 Kasumi-cho, Yagiyama, Taihaku-ku, Sendai, Miyagi 982–8577, Japan
a) *sasakiy@isenshu-u.ac.jp*

Abstract: This paper presents a compact cluster computer, called "UCC (Ubiquitous Computing Cluster)", which provides a cost-effective prototyping environment for design and test of ubiquitous computing applications. We achieve extremely small size, low power consumption and low cost by employing COTS (Commercial Off-The-Shelf) embedded CPUs. We also present an application of UCC to fingerprint verification using phase-based image matching and its performance analysis. Our experimental observation shows that the cluster is useful for prototyping practical application programs targeted at network-connected embedded CPUs.

Keywords: ubiquitous computing, parallel processing, parallel computers, cluster computers, fingerprint verification
Classification: Science and engineering for electronics

References

[1] M. Weiser, "Some Computer Science Issues in Ubiquitous Computing," *Comm. ACM*, vol. 36, no. 7, pp. 75–84, 1993.
[2] Inrevium Web Site. http://www.inrevium.jp/board/ubqcc.html
[3] Ubiquitous Computing Cluster Support Page.
http://www.aoki.ecei.tohoku.ac.jp/UCC/index.htm
[4] K. Ito, H. Nakajima, K. Kobayashi, T. Aoki, and T. Higuchi, "A fingerprint matching algorithm using phase-only correlation," *IEICE Trans. Fundamentals*, vol. E87-A, no. 3, pp. 682–691, 2004.
[5] K. Ito, A. Morita, T. Aoki, T. Higuchi, H. Nakajima, and K. Kobayashi, "A Fingerprint Verification Algorithm Using Phase-Based Image Matching for Low-Quality Fingerprints," *Int. Conf. on Image Processing (to be published).*

© IEICE 2005
DOI: 10.1587/elex.2.465
Received July 23, 2005
Accepted August 22, 2005
Published September 10, 2005

[6] debian.dodes.org. http://debian.dodes.org/index.en.html
[7] LAM/MPI Parallel Computing. http://www.lam-mpi.org/
[8] Pallas MPI Benchmark.
http://www.pallas.com/e/products/pmb/
[9] FFTW (Fastest Fourier Transform in the West). http://www.fftw.org/

1 Introduction

Today, embedded CPUs can be found in a vast variety of products ranging from cellular phones and automobile navigation systems up to network-connected household appliances. Some of these embedded devices can run advanced operating systems, such as Linux, to support flexible network connectivity. This trend accelerates the technology toward the age of "ubiquitous computing", that is to integrate computation into the environment enabling people to interact with computers more naturally [1]. In such situation, cooperative parallel processing with embedded CPUs will become one of the most important technologies to realize a variety of pervasive applications. One of the problems in managing research and development projects for ubiquitous/pervasive computing is the lack of cost-effective standardized platform for prototyping application programs on network-connected embedded CPUs.

Addressing this problem, we present a compact cluster computer with embedded CPUs, called a "Ubiquitous Computing Cluster (UCC)" (see the URL [2, 3] for detail), which provides a rapid prototyping environment for ubiquitous computing applications at very low cost. We utilize a Commercial Off-The-Shelf (COTS) product of Network Attached Storage (NAS) as a computing node for UCC to realize cost-effective prototyping environment.

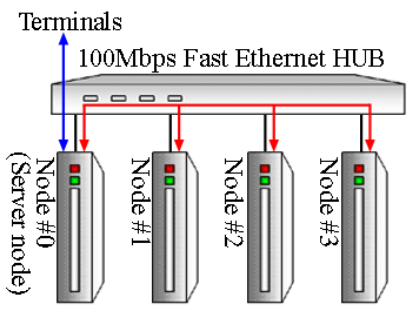
In this paper, we describe the system overview of UCC, and its application to the rapid prototyping of a parallel processing algorithm for fingerprint verification, where the algorithm uses phase-based fingerprint image matching [4, 5]. Although the actual performance of the fingerprint matching depends on the target CPUs and peripheral components, the use of UCC as a typical hardware platform makes possible to verify functionality of the algorithm itself and to obtain a rough estimate of its performance depending on the degree of parallelism. UCC is also useful for software engineering education as an easy-to-use platform for teaching fundamental concepts on embedded computers and parallel processing.

2 System overview

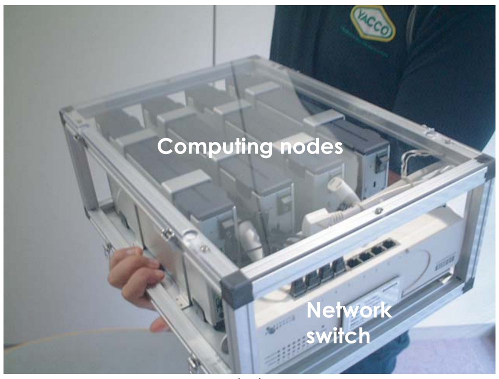
Fig. 1 (a) shows the overall architecture of UCC. It consists of four computing nodes #0, #1, #2 and #3, where the node #0 works as a server node and is directly accessible from terminals outside. These computing nodes are connected over 100 Mbps Fast Ethernet. We decided to use COTS products in constructing UCC to achieve the cost-effective prototyping environment. A Network Attached Storage (NAS), whose specification is given in Fig. 1 (c), is employed as a computing node. Every computing node runs

© IEICE 2005
DOI: 10.1587/elex.2.465
Received July 23, 2005
Accepted August 22, 2005
Published September 10, 2005

Debian/GNU Linux 2.4.21 configured for SH4 [6]. The server node #0 provides NIS and NFS services for managing login IDs and sharing user files. Telnet and FTP services are also supported to allow login and file transfer from outside. Fig. 1 (b) shows a product picture of UCC. The four computing nodes and a network HUB are mounted together in a small skeleton rack. We could achieve extremely compact size of 390 mm × 280 mm × 150 mm, low power consumption of 60 W and low cost by employing COTS products.



(a)



(b)

Computing node (Embedded NAS)	CPU	SH7751R (SH4), 266 MHz
	Main memory	SDRAM, 64 MB
	HDD	ATA133, 5400 rpm, 120 GB
	Network I/F	10/100 Base-T Fast Ethernet
	USB I/F	USB 2.0, 2 ports
Number of nodes	4	
Network switch	10/100 Base-T Fast Ethernet switching HUB	
Size [mm]	W390 x D280 x H150	
Power consumption	60W	

(c)

Fig. 1. System overview of UCC: (a) architecture, (b) product picture, and (c) overall specification.

UCC has the following features as a reference hardware platform for prototyping application programs for network-connected embedded CPUs.

- UCC supports a variety of development tools and libraries for parallel programming including GNU C, C++ compilers, vi, GNU Emacs editors, PVM (Parallel Virtual Machine), and message passing interfaces LAM/MPI [7] and MPICH.
- UCC provides a large capacity of storage (about 100 GB per node) for system libraries, user programs and data.
- UCC has enough point-to-point communication bandwidth for practical applications. The bandwidth is evaluated as 50 Mbps by Pallas MPI Benchmark (PMB) [8].
- Multiple UCCs can be easily stacked to extend the number of computing nodes.
- Each computing node has a dual-port USB interface, which enables us to extend UCC to various applications with real-world interface devices.

© IEICE 2005
DOI: 10.1587/elex.2.465
Received July 23, 2005
Accepted August 22, 2005
Published September 10, 2005

3 Rapid prototyping of fingerprint verification system

This section describes an application example of UCC – the prototyping of fingerprint verification software targeted at embedded CPUs. We employ two UCCs with a pressure sensitive fingerprint sensor connected to a USB port. The purpose of this case study is to find an efficient implementation of the verification algorithm (proposed by the authors in [4, 5]) for typical embedded applications.

The fingerprint verification algorithm considered here employs phase-based image matching using the Band-Limited Phase-Only Correlation (BLPOC) function [4, 5]. The use of phase components in 2D DFTs (two-dimensional discrete Fourier transforms) of fingerprint images makes possible to achieve highly robust fingerprint verification even for low-quality images. Experimental evaluation reported in [4, 5] demonstrates an efficient verification performance of the proposed algorithm compared with a typical minutiae-based algorithm.

A major drawback of this algorithm is its computation cost due to the use of 2D DFTs for phase-based image matching. In order to realize real-time verification on typical embedded CPUs, we need to reduce computation time significantly, while maintaining a sufficient level of verification performance. In this case study, we use UCCs to analyze the possibility of creating an efficient fingerprint verification system, which combines the phase-based matching algorithm with embedded parallel processing.

We consider here the following two approaches to reduce the computation time of phase-based fingerprint matching:

Reducing the image size

The original algorithm assumes the use of 256×256-pixel fingerprint images. In this case, computation of 2D DFT on the embedded CPU (SH4 266 MHz used in UCC) takes almost one second. This is not allowed with respect to real-time processing since the original algorithm requires more than 100 times computation of 2D DFTs per a single verification task. Our experimental observation shows that we can reduce the image size to 128×128 pixels without considerable degradation of verification performance.

Simplifying the algorithm

Let f be the input fingerprint image to be verified and g be the registered fingerprint image. We generate a set of rotated replicas g_θ of g over the angular range $-40^\circ \leq \theta \leq 40^\circ$ with an angle spacing 1° in advance. We modify the original algorithm to reduce the verification time and to enhance the efficiency in parallel processing on UCCs as follows:

[Step 1] Capture an input image f.

[Step 2] Calculate the 2D DFT of f.

[Step 3] Calculate the phase spectrum of f.

[Step 4] Calculate the cross-phase spectrum between f and g_θ for all θ.

[Step 5] Calculate the BLPOC function between f and g_θ for all θ.

[Step 6] Calculate the matching score between f and g_θ for all θ.

© IEICE 2005
DOI: 10.1587/elex.2.465
Received July 23, 2005
Accepted August 22, 2005
Published September 10, 2005

[Step 7] Find the highest matching score for verification.

We analyze the verification performance of the simplified algorithm. The fingerprint image database used in our experiment consists of impressions obtained from 30 subjects having low quality fingerprints [4]. We capture 11 impressions of the right index finger for every subject. Thus, the total number of fingerprint images used in this experiment is 330. We compare three matching algorithms: (A) the original algorithm [4, 5], (B) the simplified algorithm, and (C) a typical minutiae-based algorithm. The performance is evaluated by the Equal Error Rate (EER) in fingerprint recognition. Note that EER is defined as the error rate where the False Non-Match Rate (FNMR) and the False Match Rate (FMR) are equal.

Table I (a) shows the EER for the three algorithms. The EER of the simplified algorithm (B) is 2.80%, while the EER of the original algorithm (A) is 1.95%. Thus, the performance degradation due to the algorithm simplification is not so significant. Even after simplification, the verification performance is much better than typical minutiae-based algorithm (4.81%).

Table I. Verification performance and computation time evaluated on a single node of UCC.

(a) Comparison of EER and overall computation time for the three algorithms.

Algorithm	**EER [%]**	**Overall CPU time on UCC [sec]**
Original algorithm (A)	1.95	300
Simplified algorithm (B)	2.80	14
Minutiae-based algorithm (C)	4.81	–

(b) CPU time profiling for the simplified algorithm (B).

	Algorithm steps	**Actual CPU time on UCC [sec]**
S	Step 1: Capture an input image	0.124000
S	Step 2: Calculate the 2D DFT of the input image	0.175000
S	Step 3: Calculate the phase of the input image	0.050000
P	Step 4: Calculate the cross-phase spectrum for all θ	2.464000
P	Step 5: Calculate the BLPOC function for all θ	11.146000
P	Step 6: Calculate the matching scores for all θ	0.076000
S	Step 7: Find the highest matching score	0.000001
Overall CPU time		14.035000

We implemented both the original algorithm (A) and the simplified algorithm (B) on a single node (SH4 266 MHz) of UCC, where FFTW library [9] – a C subroutine library for computing DFT – are used to accelerate computation. On a single node of UCC, the original algorithm (A) consumes about 5 minutes while the simplified algorithm (B) takes only 14 seconds as listed in Table I (a). Thus, the simplified algorithm (B) has a potential to support real-time verification through parallel processing on multiple CPUs. Table I (b) shows the actual CPU time for each algorithm step. The symbol ’P’ denotes the steps that can be effectively parallelized since they are

© IEICE 2005
DOI: 10.1587/elex.2.465
Received July 23, 2005
Accepted August 22, 2005
Published September 10, 2005

repeated for 81 independent rotated replicas of the registered image.

We implemented the parallelized version of the simplified algorithm (B) on the 8-node cluster system using two UCCs. The parallel program can be summarized as follows: (i) the server node #0 computes the phase component of the input image (Step 1–3), while the other nodes wait for the completion of the computation, (ii) the node #0 broadcasts the phase component by *MPI_Bcast* function of LAM/MPI library, (iii) each node receives this and computes the cross-phase spectrum, the BLPOC function and the matching score between the input image f and the rotated replica g_θ of the registered image (Step 4–6), and (iv) the node #0 collects the matching scores to find the highest score (Step 7) using *MPI_Reduce* function with *MPI_MAX* operation. Fig. 2 plots actual computation time and speed-up factor with different number of computing nodes. The dashed line denotes estimated computation time given by the formula: $t = t_S + \frac{t_P}{N}$, where the sequential segment t_S and the parallel segment t_P are evaluated as $t_S = 0.350$ [sec] and $t_P = 13.686$ [sec], respectively. The actual computation time is almost consistent with the estimated computation time considering the overhead of MPI functions. When the number of node is 8, the actual computation time is 2.49 seconds, which clearly suggests the possibility of real-time verification with network-connected embedded CPUs.

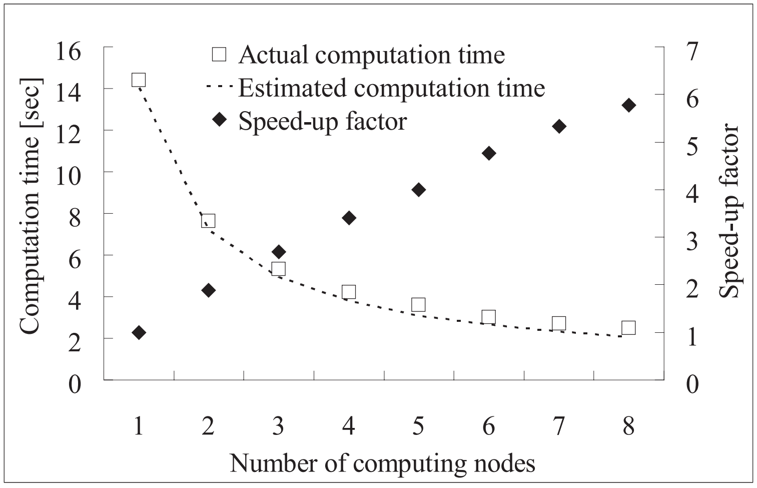


Fig. 2. Computation time and speed-up factor of parallelized algorithm.

4 Conclusions

This paper presented a compact cluster computer with embedded CPUs and its application to rapid prototyping of a high-performance fingerprint verification system. Although real-time performance depends on the details of hardware implementation, an easy-to-use platform for embedded parallel processing allows us to capture the fundamental characteristics of software designs for embedded applications.

© IEICE 2005
DOI: 10.1587/elex.2.465
Received July 23, 2005
Accepted August 22, 2005
Published September 10, 2005