2004－DSM－35　(13)
2004／9／24

組込み機器を用いたコンパクトクラスタ計算機の開発と暗号処理への応用

佐々木　慶文[1]，川村　暁[1]，青木　孝文[2]，伊藤　貴康[1]

石巻専修大学理工学部[1]，東北大学大学院情報科学研究科[2]

概要

次世代ユビキタスコンピューティング環境においては，ネットワーク結合された組込みCPUによる並列・協調処理が重要であることが予想される．筆者らは，ユビキタスコンピューティングの各種応用のプロトタイピング環境を低コストで実現することを目的とし，“ユビキタス・コンピューティング・クラスタ (Ubiquitous Computing Cluster: UCC)”と呼ぶクラスタ計算機を開発した．UCCは，市販の組込み機器を計算ノードとして用いることにより，世界最小クラスのサイズ，消費電力およびコストを実現している．本稿では，UCCの構成および性能評価について述べるとともに，一例として，UCCをカオス・ニューラルネットワークに基づく暗号化処理に応用した結果について報告する．

Development of a Compact Cluster Computer with Embedded Devices and Its Application to Stream Cipher

Yoshifumi Sasaki[1], Akira Kawamura[1], Takafumi Aoki[2] and Takayasu Ito[1]

Faculty of Science and Engineering, Ishinomaki Senshu University[1],
Graduate School of Information Sciences, Tohoku University[2]

Abstract

This paper presents a compact cluster computer with embedded CPUs, called “Ubiquitous Cluster Computer (UCC)”, in order to provide 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 products. As an application example, we also discuss implementation of a stream cipher based on Chaos Neural Network (CNN) technique.

1. はじめに

近年，組込みプロセッサ（CPU）は車載ナビゲーション，携帯電話，ディジタルカメラ，家電製品など様々な機器に搭載されている．また，これらの組込み機器の中には，先進的なOSやネットワーク機能を備え，論理的にハイエンドCPUを搭載したPCと同等の機能を有するものも少なくない．将来的には，このような高性能組込み機器が相互にネットワーク結合され，並列協調処理を行うようなユビキタスコンピューティング環境が実現されることが予想される[1]．今後，組込み機器を用いた多様な分散協調アプリケーションの開発が重要となり，このためのプロトタイピング環境の整備が必要になると考えられる．しかしながら，従来，並列協調アプリケーションのプロトタイピング

においては, 巨大なラックにマウントされた高価な PC クラスタなどを用いる必要があり，コストや設置スペースなどの制約から開発環境を整備・利用することが困難であった.

これに対して筆者らは, 様々なユビキタスコンピューティング応用のプロトタイピングを極めて低コストで実現することを目的とし, ユビキタスコンピューティングクラスタ (Ubiquitous Computing Cluster: UCC)と呼ぶコンパクトクラスタ計算機を開発した[2].

UCC は 4 台の計算ノードとネットワーク HUB（100Mbps Fast Ethernet）がコンパクトなスケルトンラックにマウントされたクラスタ計算機である．UCC の最大の特徴は，コストを抑えるために，構成要素として組込み CPU を搭載した市販の製品（Commercial Off-The-Shelf: COTS）を用いている点である.

計算ノードとしては組込み CPU，メモリ，ハードディスクドライブ, ネットワークアダプタおよび USB インタフェースを搭載した市販の Network Attached Storage (NAS)を採用した. 並列処理のための安定したプロセッサ間通信を実現するため，各計算ノードには Linux OS とリモート通信サービスなどを搭載している. また，並列処理インタフェースとして，最も典型的な Message Passing Interface (MPI) [3]および Parallel Virtual Machine (PVM) [4]の 2 種類を用意している.

本クラスタ計算機は, 組込み型 COTS 製品を用いることにより，世界最小クラスのサイズ（390mm×280mm×150mm），消費電力（60W）およびコストを実現している．このため，ユビキタスコンピューティング応用のプロトタイピングのみならず, 教育機関や企業における並列処理技術の教育プログラムへ容易に導入することが可能である.

また，UCC では，各計算ノードに 2 ポートの USB インタフェースを搭載しており， Web カメラや音声入力装置などの様々な USB 入出力機器を接続することが可能である. このため, 比較的容易に各種アプリケーションシステムへの拡張が可能である.

本稿の構成を示す．次章で，開発した UCC の構成を示す．第 3 章では，並列処理における基本的なメッセージパッシング性能の実機による検証結果を示す．第 4 章では，ネットワークセキュリティを目的とした暗号化･復号処理への応用を示す. 暗号化･復号処理は演算量が比較的多く，また，処理の性質上，並列処理が有効な場合が多い. 一例として，カオスを用いた秘密鍵ストリーム暗号の並列処理応用を試みた結果について示す. 第 5 章はまとめである.

2. UCC の構成

図 1 に開発した UCC のアーキテクチャを示す. 開発したクラスタ計算機は, ノード#0, #1, #2 および#3 の 4 台の計算ノードで構成される. これらの計算ノードは，汎用のネットワーク HUB を介して, 100Mbps Fast Ethernet で相互に接続されている.

ユビキタスコンピューティング応用のプロトタイピング環境を低コストで実現するため, UCC を構成する機器には，容易に入手可能な COTS 製品を用いた．計算ノードとしては, 266MHz 動作の SH4 CPU， 64MB SDRAM, 120GB ハードディスクドライブ, 100Mbps Fast Ethernet および 2 ポートの USB 2.0 インタフェースを搭載した Network Attached Storage (NAS)を採用している．計算ノードの詳細仕様を表 1 に示す.

各計算ノードには，フリーウェアであり，かつ, 安定したプロセッサ間通信が供給できるという観点から，Linux OS（Debian Linux 2.4.21 for SH4）[6]を搭載している．また，プロセッサ間通信を行うリモートコマンド(rsh,など)をプリインストールしている．計算ノード#0 は管理サーバを兼ねており，ログイン ID の管理（NIS），ファイル共有（NFS），リモートログイン（telnet）およびファイル転送(FTP)を行うための各種サービスを提供する.

UCC のアプリケーション開発環境としては, GNU C（gcc-3.0.4，g++-3.0.4）および Fortran77 コンパイラ，vi および GNU Emacs エディタ，並列処理インタフェースがプリインストールされており，導入後，すぐにアプリケーション開発が可能である. 並列処理インタフェースとしては，最も典型的な MPI である mpich-1.25 と PVM-3.4.2 の 2 種類を用意している.

図 2 に UCC のプロトタイプを示す．また，UCC の詳細仕様を表 2 に示す．UCC は，すべての構成機器に組込み型 COTS 製品を用いた

こととOSなどのソフトウェアとしてフリーウェアを用いたことにより，サイズ（390mm×280mm×150mm），消費電力（60W TYP），コストともに世界最小クラスを達成している．このため UCC は特別な電源や空調などの設備を必要とせず，また，卓上に乗るほどコンパクトであることから，プロトタイピングのみならず，教育機関や企業における並列処理技術の教育プログラムなどに比較的容易に導入することが可能である．

UCC の特長の 1 つとして，すべての計算ノードが USB 2.0 インタフェースを搭載している点が挙げられる．すなわち，Web カメラ，音声入力装置など USB ベースの様々な入出力機器やセンサを接続することにより，容易に様々なアプリケーションシステムに拡張することが可能である．図 3 は UCC に Web カメラを接続した画像処理システムへの拡張例である．このほかにも，音声入出力機器などの接続が確認されており，マイクロフォンアレイシステムなどへの拡張が可能である．

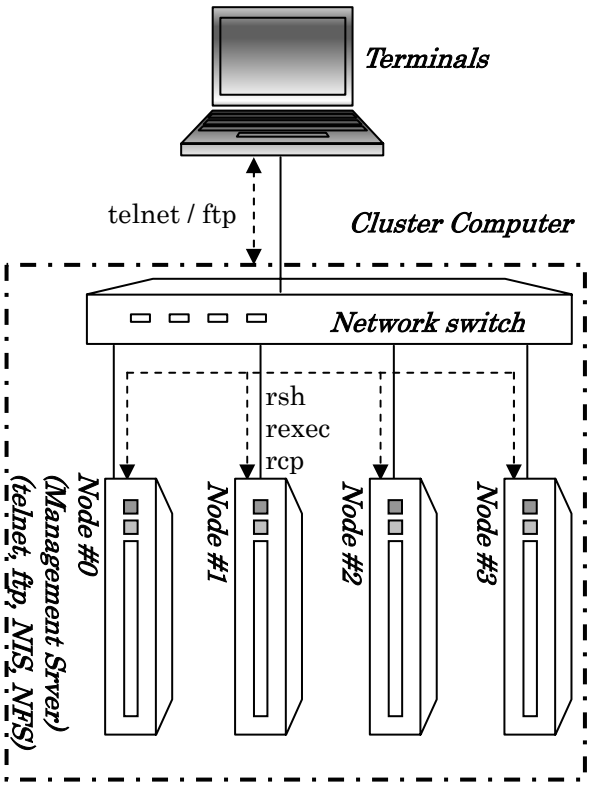

図 1: UCC のアーキテクチャ

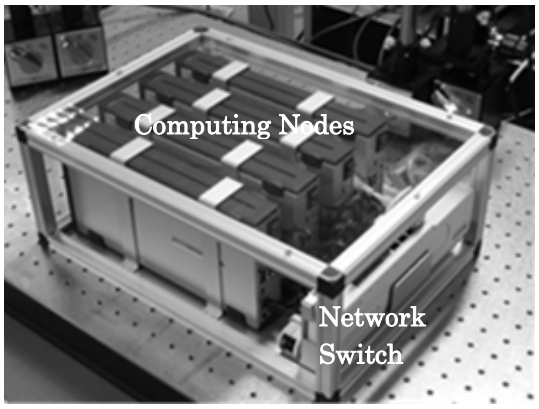

図 2: UCC のプロトタイプ

図 3: 簡易画像処理システムへの拡張例

表 1: 計算ノード（NAS）の諸元

CPU	SH4 (SH7751R, 266MHz)
Memory	64MB SDRAM
HDD	120GB, ATA133, 5400rpm
NIC	10/100 BASE-T (RTL-8139C+)
I/F	USB 2.0×2port
OS	Debian Linux 2.4.21

表 2: UCC の諸元

Computing nodes		Embedded NAS
Number of computing node		4
Network interface		10/100 BASE-T
Power consumption		60W (TYP)
Size [mm]		W390 x D280 x H150
Software	OS	Debian Linux 2.4.21
	Server functions(*)	NIS, NFS
	Communication functinos	telnet,ftp,rsh,rexec,rcp
	Development environment	GNU C, C++, F77
		vi, GNU emacs
		MPI(mpich), PVM
(*) only for the management server		

3. UCC の通信性能評価

本節では UCC のメッセージパッシング通信性能について基本的な評価を行った結果を示す．評価には MPI 関数による通信性能を評価するための典型的なベンチマークプログラムである Pallas MPI benchmark (PMB) [6]を用いた．本稿では，特に基本的な通信である ping-pong 通信とブロードキャスト通信の通信遅延およびバンド幅を測定した．

図 4 に ping-pong テストの結果を示す．ping-pong テストは，2 つのプロセッサが交互にメッセージを送受信する際の通信遅延およびバンド幅を測定する．データサイズが小さい場合には，通信オーバーヘッドのためにスループットが低くなっているが，データサイズが大きくなるにつれて帯域幅は大きくなり，最大で 70Mbps 程度の性能を示している．UCC では一般的な 100Mbps Fast Ethernet を用いている点と市販の安価なスイッチング HUB を用いている点を考慮すると，十分な性能が出ていると考えられる．

図 5 はブロードキャスト通信テストの結果である．ブロードキャスト通信テストでは，計算ノード#0 から計算ノード#1，#2 および#3 に対してメッセージを送信した場合の，最も遅い通信時間を測定している．結果としては ping-pong 通信の場合とほぼ同様な傾向が見られる．ピーク性能は 36Mpbs 程度と，ping-pong 通信の場合ほどは性能が出ないが，許容できる範囲の結果が得られている．

これらの性能評価の結果から，UCC は並列処理応用に対して十分な通信性能を有すると考えられる．

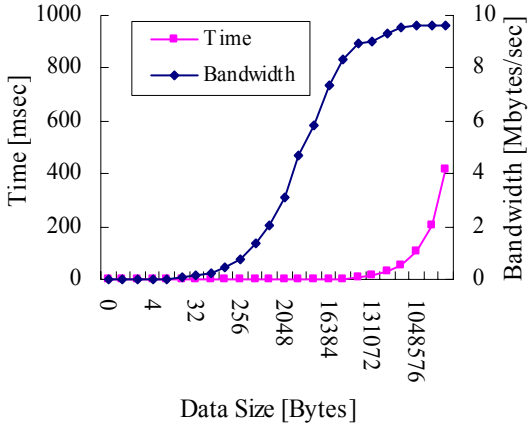

図 4: ping-pong 通信のベンチマーク結果

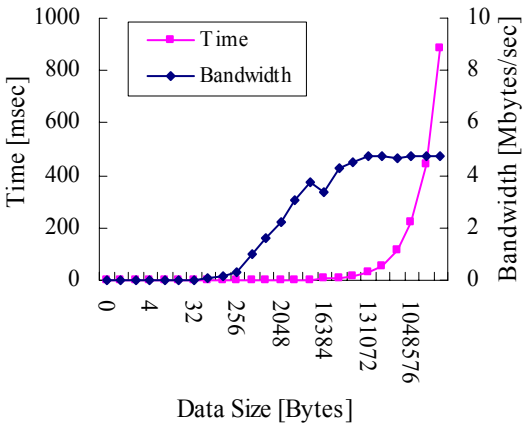

図 5: ブロードキャスト通信のベンチマーク結果

4. UCC の暗号処理への応用

筆者らは，UCC によるユビキタスコンピューティング応用として，既に指紋照合システムの開発に着手している[7]．本稿では，もう一つの事例として，カオスを用いた秘密鍵ストリーム暗号の並列処理への応用を試みた．ここでは，暗号化エンジンとしてカオス・ニューラルネットワーク（CNN）を用いた暗号系（CNN Cipher）[8-10]を UCC へ実装した．

通常のニューロンモデルから構成されたニューラルネットワークにおいて，カオス応答する構成を CNN と定義する．CNN では，カオス出力から一様性に優れた乱数が生成でき，CNN Cipher では，この乱数を暗号化・復号に用いる．擬似乱数を用いた暗号系の安全性は，暗号化に用いる乱数系列の予測不可能性に依存する．CNN Cipher は初期値鋭敏性および長期予測不可能性を有するカオス時系列から生成した一様性に優れた擬似乱数系列を用いる．

CNN Cipher の概要を示す．図 6 に，4 個のニューロンを用いた CNN の構成を示す．ネットワーク構成はカオス応答するものを用いている．カオス信号は，そのままではカオス特有の分布を示す．このカオス出力から一様性に優れた乱数を生成するために，カオス出力の一部を取り出す手法を採用した（図 7，図 8）．

まず，任意のニューロンのカオス出力を取り出す（図 7）．取り出した出力は double 型（IEEE 754 形式）の浮動小数点表現されており，この仮数部の下位ビット側を取り出し乱数を得る（図 8）．このようにして得られた乱数を，暗号

化および復号に用いた（図 9）．暗号化および復号の方法から明らかなように，本暗号系は CNN から生成された乱数成分を逐次的に用いるストリーム型の秘密鍵暗号である．ここで暗号鍵（秘密鍵）は，乱数生成に用いる CNN の構成および乱数の取り出し手法から生成する．

本手法に基づく暗号化処理について，実機上での実験を行った．図 10 にシステムの構成を示す．任意の平文データ PT がノード#0 に入力されるものとする．高速処理のため，PT をノード数に応じて均等に PT0 - PT3 の部分データに分割し，各計算ノードで並列処理を行う．各計算ノードには同じ構成の CNN 暗号化エンジンを実装し，計算ノード#0 から転送された部分データを並列に暗号化する．暗号化された部分データ ET0 - ET3 はノード#0 に戻され，ノード#0 はこれらを結合して暗号文 ET を生成し，出力する．ただし，本実験では，平文／暗号文の入出力部分は実装しておらず，平文データ PT は，初期化の際にノード#0 において生成している．

本実験では，任意のノード数について，データサイズを 64 バイトから 16M バイトまで変化させて処理時間を測定した．ノード数は 1，2 および 4 について測定を行っている．

図 11 に測定結果を示す．ノード数が 2 の場合の処理時間はノード数が 1 の場合と比較して 2/3 程度，ノード数が 4 の場合は 1/2 程度となっている．ノード数に完全に比例した性能向上が見られない理由としては，通信のオーバーヘッドが考えられる．また，図 12 は測定時間から UCC による暗号化処理の性能をバンド幅の形式で表示したグラフである．多少ばらつきはあるものの，4 ノードで処理を行った場合，最大で 30Mbps の暗号化性能を有することから，UCC では，実用的なレベルでの暗号化処理が可能であると考えられる．

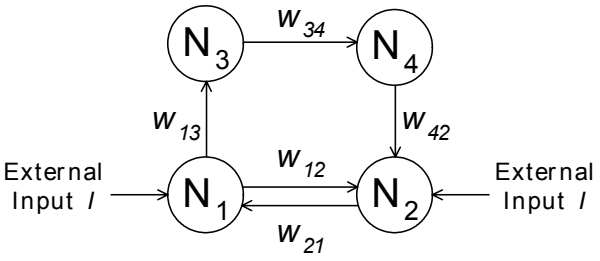

図 6: 4 ニューロンのカオス・ニューラルネットワークの構成（*I* は外部入力）

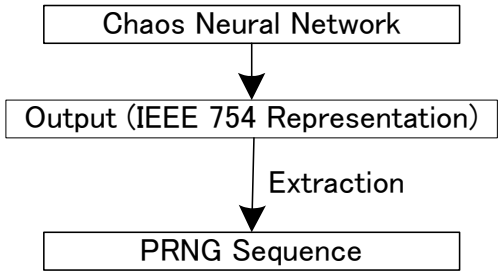

図 7: 擬似乱数生成法

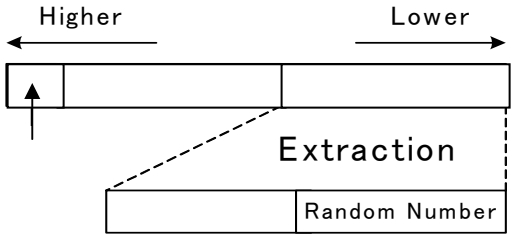

図 8: カオス出力からの擬似乱数の抽出

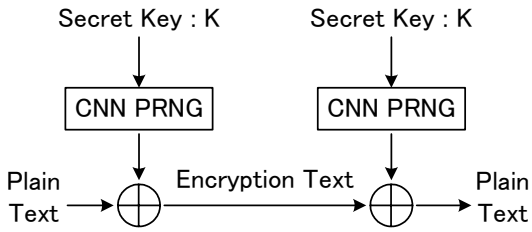

図 9: カオス暗号系の構成

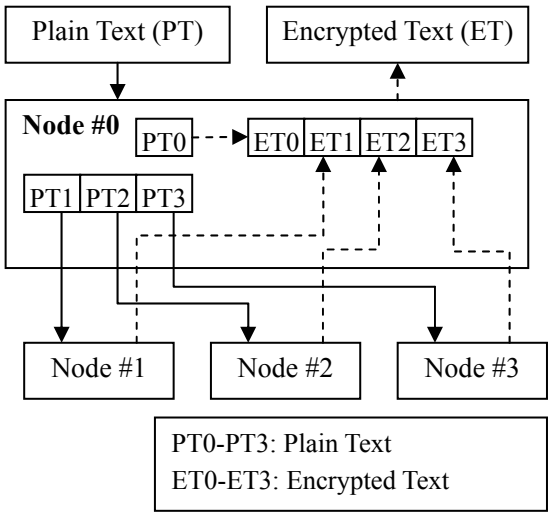

図 10: UCC による暗号化処理システム

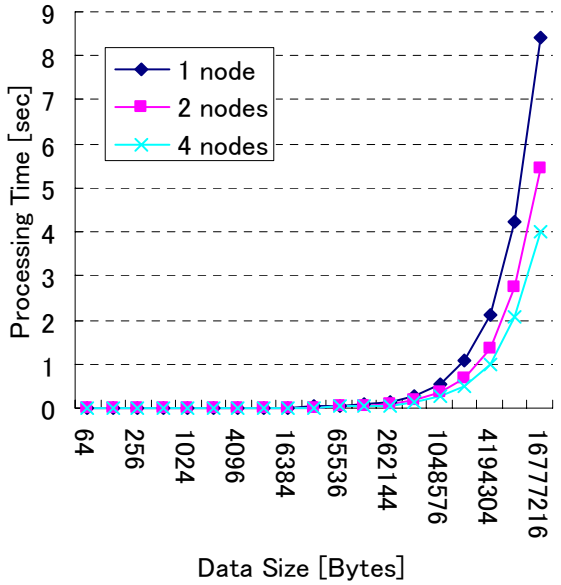

図 11: 暗号化処理時間の測定結果

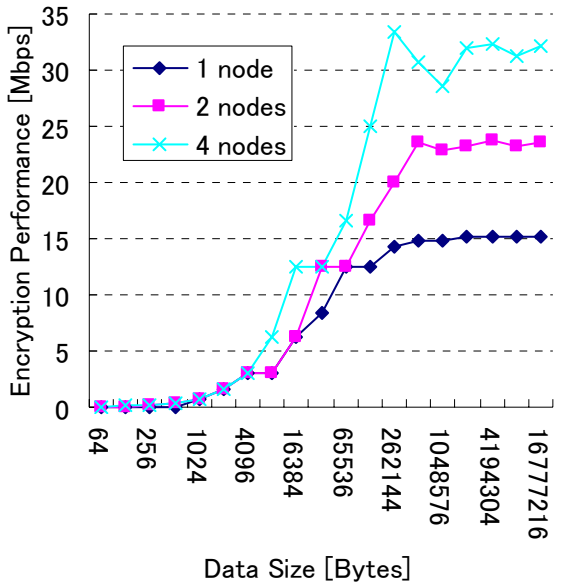

図 12: UCC による暗号化処理性能

5. まとめ

本稿では, ユビキタスコンピューティング応用のプロトタイピング環境を低コストで実現するユビキタスコンピューティングクラスタ（Ubiquitous Computing Cluster: UCC）の開発について報告した. 開発したクラスタは, すべての構成要素として市販の製品を採用することにより, 世界最小クラスのサイズ, 消費電力およびコストを実現している. このため, ユビキタスコンピューティング応用へのプロトタイピングのみならず, 教育機関や企業における並列処理技術の教育プログラムへの容易な導入が可能である.

また, ユビキタスコンピューティング応用の一例として, カオスを用いた暗号系の並列化および UCC への実装について述べた. 計算機実験による評価結果から, 10/100 BASE-T 等の一般的に用いられるネットワークに対して十分な速度が得られ, 本クラスタの有用性が示された.

References

[1] M. Weiser, "Some Computer Science Issues in Ubiquitous Computing," Communications of the ACM, vol. 36, No. 7, pp. 75-84, 1993.

[2] Ubiquitous Computing Cluster http://www.aoki.ecei.tohoku.ac.jp/ucc/

[3] MPICH-A Portable Implementation of MPI http://www-unix.mcs.anl.gov/mpi/mpich/

[4] PVM: Parallel Virtual Machine http://www.csm.ornl.gov/pvm/pvm_home.html

[5] Debian Distribution for SH3 and SH4 http://debian.dodes.org/index.en.html

[6] Pallas MPI Benchmark http://www.pallas.com/e/products/pmb/

[7] S. Sukaridhoto, Y. Sasaki, K. Ito and T. Aoki, “Development of a Compact Cluster Computer with Embedded CPUs” (submitted to IES2004).

[8] H. Yoshida, K. Yoneki, Y. Tsunekawa and M. Miura, ”Chaos Neural Network,“ Proc. of ISPACS'96, vol.1of3, pp.16.1.1-16.1.5, 1996.

[9] 川村 暁, 吉田等明, 三浦 守, 通常のニューロンより成るカオス・ニューラルネットワークの最小構成, 電子情報通信学会論文誌(A), Vol.J84-A, No.5, pp.586-594, 2001.

[10] S. Kawamura, H. Yoshida, M. Miura and M. Abe, “Implementation of Uniform Pseudo Random Number Generator and Application to Stream Cipher based on Chaos Neural Network,” Proc. of the International Conference on Fundamentals of Electronics, Communications and Computer Sciences, R-18, pp.4-9, 2002.