小型 PC クラスタ実験システム構築の検討

Construction of Experimental Cluster System with Card-sized PC

本田 隆司*　(*情報工学)

Takashi HONDA

Summary
This document describes a method of constructing a testbed system composed of the credit card sized PC boards. This system is used for the purpose of evaluating a Linux-based parallel processing functionarity. As the initial results it was possible to evaluate the performance degradation according with the dispersing treatment.

キーワード：カードサイズ PC クラスタ構成 並列処理 Linux
Keywords：Card-size PC Cluster Parallel Processing Linux

1. はじめに

近年の携帯端末に係る技術進展に伴い、主に組み込み用途に用いられる CPU 部品が高性能化し、かつ数多く安価に流通してきた。これらはデジタルカメラの記憶媒体として利用されている SD メモリカードを記憶媒体として利用し、超小型 PC として Linux などの汎用 OS のもとで単体で動作させることが可能である。これらの機器は処理能力的に前世代のノート PC に迫ることもあり、各種実験制御用などその利用範囲も拡大しつつある。

2. 並列処理システム

一方、社会的ニーズの高まりつつあるビッグデータ解析などにおいて、並列処理機構に関しての動向も近年その活用が着目されてきている。(1)
これまで並列処理機構は実用的手段としては高価な専用設備を利用することが求められたが、クラウドシステムなどの登場で計算資源リソースは仮想化され、その構成手法も多様化してきている。

我々はこの超小型 PC を多数用意してイーサネットワークで相互結合し、並列処理計算機構の評価用テストベッドシステムを構築することを計画した。

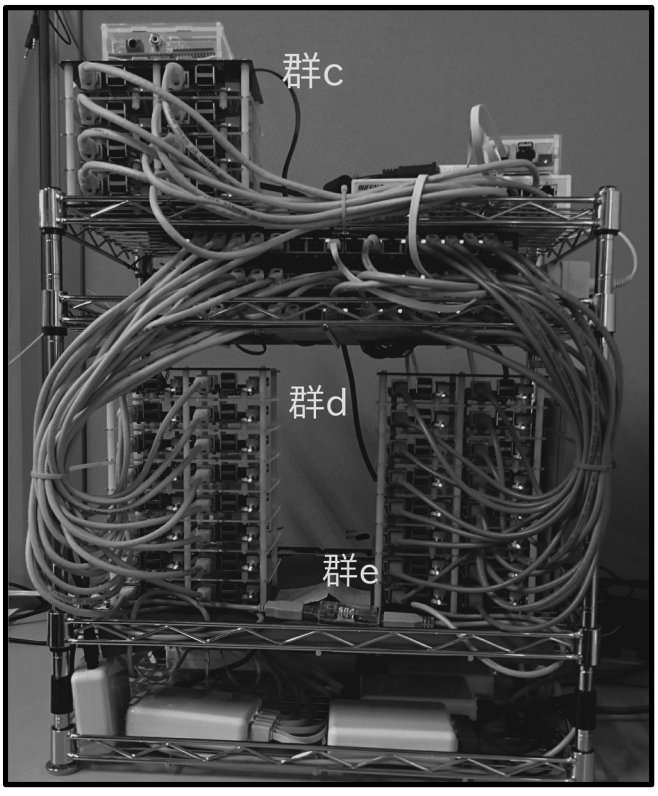


写真 1 今回構成したテストベット装置

3. テストベッドシステム

今回構成するシステムの目的として以下の項目を上げる。

(1) PC 台数と処理能力の関係(スケーラビリティ)
(2) 複数の並列処理手法の適用性
(3) 複数計算機の制御機構の検討

今回はその構成の容易さから、共有メモリを有しない疎結合計算機モデルを用いて、複数の独立した汎用 PC をイーサネット接続して分散処理装置を構成し、パフォーマンス評価実験を簡易に実施できることに主眼を置くこととした。

3.1 ノード構成

構築したシステムのハードウエア構成を下表に示す。電源ユニットに関しては容量の違う 2 種類を用意して長期連続運用での安定性を検証した。

CPU ボード (プロセッサ)	RaspberryPi type-B (700MHz/ARM11)
記憶媒体	SD カード 8GB
電源ユニット A	5V 4port 合計容量 2A
電源ユニット B	5V 6port 合計 5A(予備構成)
冷却ファン	随時設置

ソフトウェア構成(OS)としては Linux OS である GNU Debian 7.9 をベースとして Raspberry Pi 向けに移植したバージョン Raspbian wheezy (2)を使用した。

3.2 相互結合機構 (ネットワーク)

16 ポートギガビットスイッチ 3 台を採用した。

CPU ボード内蔵のネットワークインタフェースは 100Mbps Fast Ethernet であるが結合機構としてはより高速のギガビットスイッチを採用した。

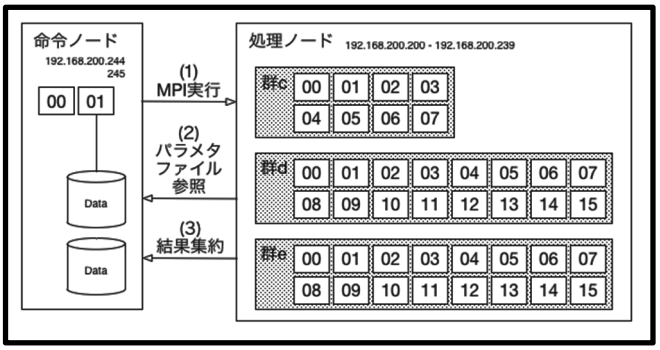


図 1 テストベットシステムの装置構成

構成した実システムを写真 1 に示す。そのシステム構成を図 1 に示す。

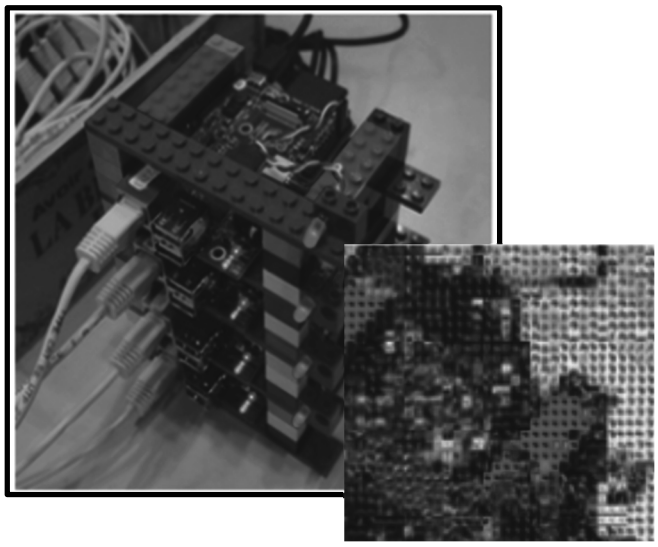
4. システム評価環境

評価手法として利用できるソフトウェア環境には以下に示すものが候補としてあげられる。

多くのものを初期段階で一次評価に用いたが、ここではその検討の一部を示す。

4.1 画像モザイク加工処理

今回試験した画像処理の例を図 2 に示す。



上:原画像 右:処理後(拡大)

図 2 分散モザイク処理結果画像(例)

4.2 MPI 試験

MPI(Message Passing Interface)は、並列計算処理を利用するために標準化された規格である(3)。この仕組を利用して複数計算ノードがメッセージを送受信して協調動作を行うことができる。

比較的構成が簡単なため今回はこの仕組で並列計算をさせることとした。詳細は次節で述べる。

4.3 その他の評価手法

以下の評価を実行中または準備中である。

・姫野ベンチ

並列処理を含むベンチマークテストとして多く活用される汎用ツールである(4)。今回も一次評価として利用した。

・動画分散エンコード処理

計算量の多い動画像圧縮を分散機構で処理する機構である(5)。適切なパラメタ選定中である。

5. MPI 評価結果

今回評価で用いた MPI 機構による並列円周率計算の結果を述べる。全体動作としては、図 1 に示す通り、MPI 機構により命令ノードより対象となる処理ノード群に対して処理開始を指示(1)し、共通メモリからパラメタをロード(2)し、処理完了後に計算結果を集積(3)するものである。

円周率計算は下に示す計算式(a)により面積計算にもとづき値を求めるものである。図 3 に示す区間 x=0～1 の面積を部分区間に N 分割し、それを K 台の複数 PC で分担して求積するものである。K=2/N=1,000,000 の場合、x=0.0～0.5 を 500,000 分割して PC1 が、x=0.5～1.0 を 500,000 分割して PC2 がそれぞれ求積することとなる。

$$\int_0^1 \frac{1}{(1+x^2)} \cdot dx = [\tan^{-1}(x)]_0^1 = \frac{\pi}{4} \quad \text{(a)}$$

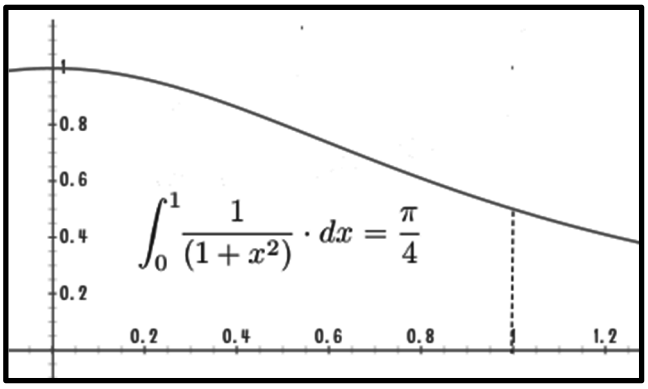


図 3 求積範囲の分割

領域分割数 N を変更することにより、計算処理量を変化させており、システム負荷パラメータとした。分割数 N は 1,000,000～30,000,000 まで変化させた。使用ノード数 K を 1/2/4/8/16/32/40 と変化させた。それらの組み合わせで処理完了までの時間の変化を観察した。

単純に言えば、使用ノード数 K を 2 倍にすれば全体の処理時間 t は 1/2 となり、その述べ処理時間は変化しないこととなるが、処理分割のオーバヘッドにより使用ノード数を増やせば述べ処理時間 Kt は増大することとなる。今回の評価はその増加を測定した。

5.1 評価結果

図 4 は使用ノード数 K をパラメータとして領域計算分割数 N を変化させた場合の処理時間の増大である。K=2 における分割数 N=30,000,000 の場合以外は所要時間は分割数 N に比例していることがわかる。

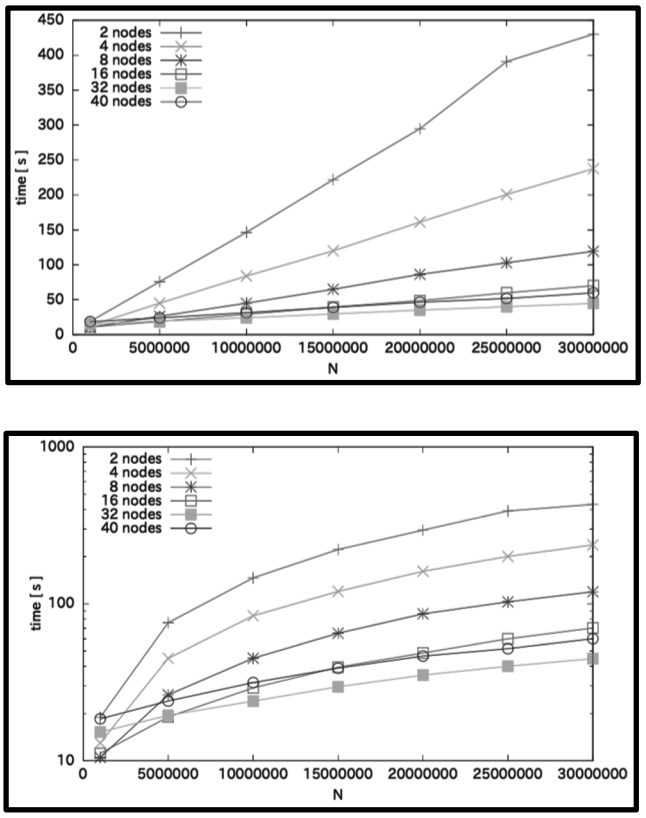
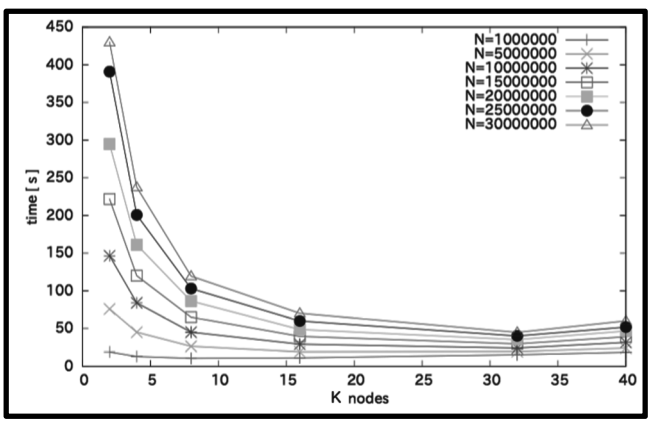


図 4 求積領域分割数 N と処理時間の関係

図 5 は分割数 N をパラメタとして使用ノード数 K を変化させた場合のノードあたりの所要処理時間である。分割数 N=1,000,000 を除いて処理ノード数の増加により処理時間は減少している。ただし K=40 の場合は K=32 に比べて増えている点が疑問点である。



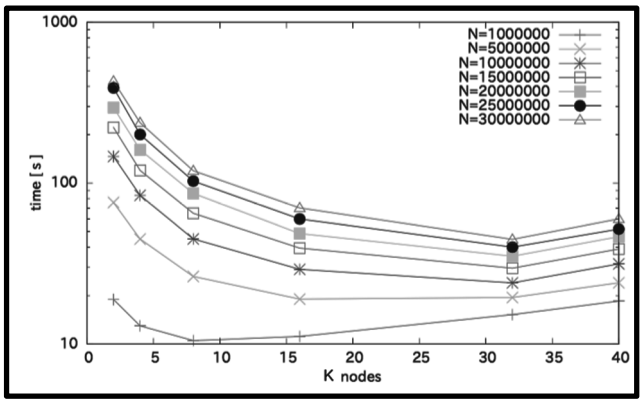


図 5 使用ノード数 K とノードあたりの所要処理時間

図 6 は述べ処理時間 Kt を比較したものであるが、使用ノード数 K の増加に伴い分割オーバヘッドが影響しており、特に領域分割数 N が小さい場合にその影響が大きいことが分かる。

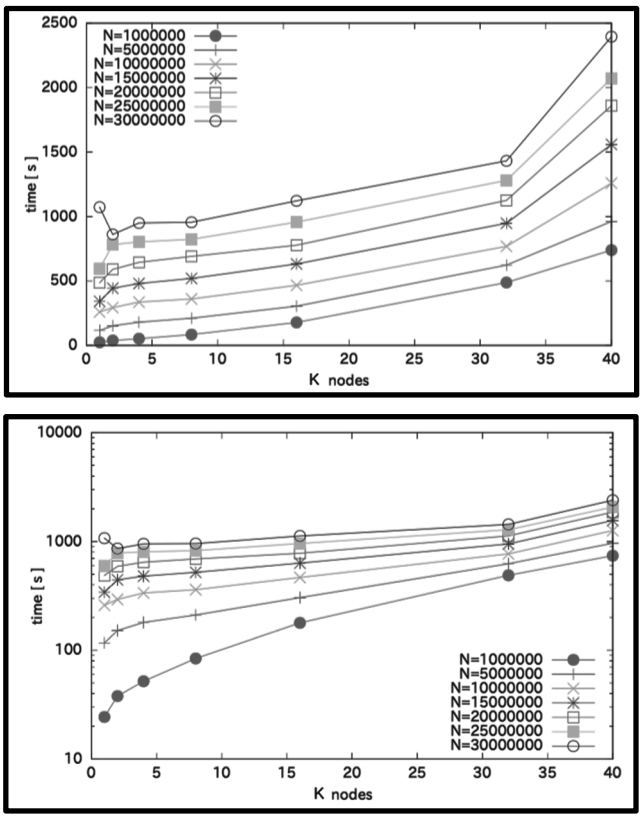


図 6 使用ノード数 K と延べ処理所要時間 Kt

6. まとめ

今回の企画においては複数台の Linux PC をネットワークでつなぎ分散処理系を構築でき、学生でも手軽に扱えるモデル実験システムを安価に構成することができた。本システムにより、Linux OS をベースとした各種分散処理系のパフォーマンス評価を実施できるようになった。

構築にあたっては携帯電話組み込み用途に供給されている ARM 系プロセッサを載せた SoC（System On A Chip）基板を用いた名刺サイズ Linux ミニ PC を 40 台用意し全体制御に 2 台追加し、100Mbps Fast Ethernet で相互接続して一体的に運用できるようにした。

構築の結果として以下の知見が得られた。

(1) 民生用の USB インターフェースの携帯電話用充電器を高密度に多数配置したが、長時間連続運転は想定外仕様のため特に電源系において放熱処理が必須であり、ファン空冷が必要であった。

(2) 複数台の小型機器の正常動作確認（死活監視）を含めシステム起動・停止には集中管理機構が必須であった。2 台の監視制御系を追加した。

(3) システム外ネットワーク（研究室ネットワーク）とは外乱遮断し独立ネットワークとして動作させて評価必要であることが判明した。

初期段階の評価は 2014 年度卒業研究をベースに行っており、今回は以下の項目を重点化して検討した。

(1) 簡易言語 Python と分散処理ソフトライブラリを用いで円周率計算における負荷分散処理機構の設計
(2) 集中型制御ノード（命令ノード）による分散ノード端末制御機構の構築
(3) ジョブ並列分割によるオーバヘッド評価
(4) 長時間安定性とシステム使用電力評価

次段階評価として、継続発展的に以下の点の解明に注力していく予定である。

(1) Python 以外の言語による並列処理ソフトウェア・ライブラリ仕様の評価
(2) 視覚化を含め学生レベルでもより手軽で扱いやすい評価インタフェースの構築
(3) 分散処理効果が明確に現れるアプリケーションおよび評価パラメタの検討

参考文献

1）ビッグデータ基盤
http://www.iij.ad.jp/biz/bigdatalab/bases.html
2）http://raspbian.org
3）Message Passing Interface (MPI)
https://ja.wikipedia.org/wiki/Message_Passing_Interface
4）姫野ベンチマーク http://accc.riken.jp/2145.html
5）Media Encoding Cluster
http://sf.net/projects/bripper