Vol.2014-IOT-24 No.17
2014/2/28

安価なコンピュータを用いた実験・教育用並列計算機環境の構築

信田 圭哉[1]　長谷川 明生[1]

概要：一般的な PC よも安価な Raspberry Pi を使用し，コンピュータクラスタを構成する．並列計算機環境の構築の一部を自動化するシステムを開発し，コストの面と扱いやすさの点から初学者向きの環境を目指す．

Construction of parallel computer for experiments and education using inexpensive computer

Nobuta Keiya[1]　Hasegawa Akiumi[1]

Abstract: We construct a computer cluster using Raspberry Pi which is lower cost than common PC. We developped a system to automate a part of construction of parallel computing environment, and are aimed at environment for beginner in cost and in term of ease of use.

1. はじめに

ハイパフォーマンスコンピューティングの分野において，MPI を用いた分散メモリ型の並列計算機に対する注目が高まっている．教育や簡単な実験目的での並列計算機の利用には，金銭的コスト，管理コストの問題がある．

Cox らは，一般的なパーソナルコンピュータ（以下，PC）よりもはるかに安価な Raspberry Pi を使用し，低コストなコンピュータクラスタが構築できることを示した [1]．Cox らは，64 台の Raspberry Pi をネットワーク接続し，その上で MPI を使用した並列計算機を実現している．

本研究では，Cox らの手法に基づき 24 台の Raspberry Pi を用いて MPI の並列計算機を構築した．並列計算環境の構築の一部を自動化して管理コストを減らし，コストの面と扱いやすさの点から初学者向きの環境を目指す．

2. Raspberry Pi を使用したクラスタの構築

構築したクラスタの写真を図 1 に示す．図に示すように，24 台の Raspberry Pi を，アクリル板を加工して作成した箱に並べた．4 台の Raspberry Pi を乗せた板が箱の中に 6 枚入っている．

2.1 Raspberry Pi

Raspberry Pi はクレジットカード程度の大きさのコンピュータであり，Raspberry Pi Model B は 700MHz の ARM1176 プロセッサをコアに，512MB の RAM を搭載している．Raspberry Pi 上では Linux ベースの OS が動作し，市販の PC と同様に使える．そのため，Beowulf 型と呼ばれるコンピュータクラスタの実験に利用できる．

一般的な PC は専用の並列計算機に比べ安価であるが，1 台数万から数十万円はかかる PC を複数台揃えるには，金銭的コストが問題になる．教育や簡単な実験を行う目的であれば，金銭的コストはできるだけ低くしたい．Raspberry Pi は日本で購入する場合 1 台 3000 円〜4500 円程で，一般的な PC と比較し安価であるため，金銭的コストを低くできる．

ストレージには SD カードを使用し，Raspberry Pi 用にビルドされた OS のイメージを SD カードに書込むことで使用できる．OS には Raspbian wheezy（バージョン 2013-02-09）を使用した．最新版は Raspberry Pi Foundation の

[1] 中京大学
Chukyo University

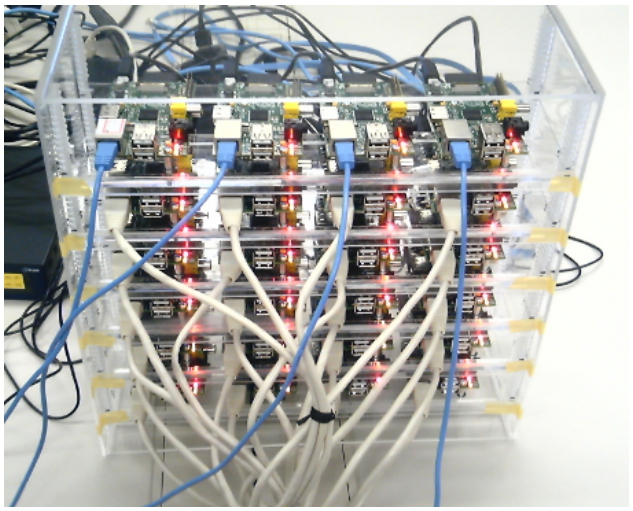

図 1 Raspberry Pi 24 台を使用したクラスタ

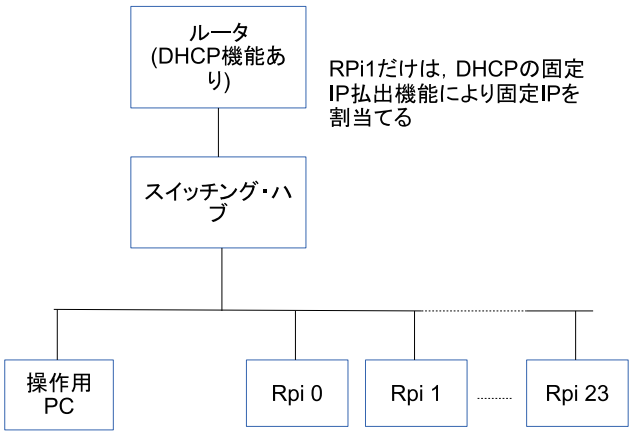

図 2 ネットワーク構成

Web ページ [2] からダウンロードできる．電力供給は microUSB ケーブルを使用し，消費電力は $5V * 700mA = 3.5W$ 程であり，電力の面でも低コストであることが期待できる．10/100 Base-T の Ethernet ソケットがあり，ネットワーク接続に使用する．

2.2 ネットワーク構成

ネットワーク構成の概要を図 2 に示す． クラスタを構成するのは Raspberry Pi のみで，図には Rpi n（n=0〜23）で示している．これらを全て同じサブネットに接続する．操作用 PC をクラスタを構成するマシンと同じネットワークに接続する．

MPI プログラムにおいて，プロセスのランクに応じてマスタとスレーブという呼び方がある．マスタプロセスを実行するマシンのことを，ここではマスタマシン，それ以外をスレーブマシンと呼ぶ．マスタマシンの操作は，操作用 PC から SSH でログインして行う．

ネットワーク接続には 24 ポート以上接続するだけのスイッチングハブが必要で，ルータには DHCP サーバ機能が備わっているものを使用した．Raspberry Pi 側では固定 IP の設定はせず，DHCP により自動的に IP を取得するようにした．DHCP サーバ側で，マスタマシンとなる Raspberry Pi と操作用 PC に対し固定の IP アドレスを割当てるようにした．

2.3 MPICH の構築

本研究では MPICH2（バージョン 1.4.1p1）のソースコードを，Web ページ [3] からダウンロードし使用した．ソースコードのビルドは Raspberry Pi 上で行った．MPICH の configure オプションは，インストールディレクトリのみを指定した．MPICH を特に指定せずにビルドすると，通信を SSH で行うよう設定される．

クラスタを構成する全てのマシンに同じライブラリをインストールする必要があるが，1 台のマシンにインストールした後，残りのマシンはその OS イメージをコピーして使用すればよい．

全てのマシンにイメージをコピーした後，MPICH に付属している円周率を求めるプログラムを使用し，PC クラスタとして動作していることを確認した．

3. 並列計算環境の自動構築

並列計算機を手軽に扱うためには管理コストをできるだけ減らしたいという要求がある．管理コストには，ハードウェアやソフトウェアプラットフォームのセットアップ，クラスタの構成等が挙げられる．

並列計算機環境の管理コストのうち，ハードウェアのセットアップは手作業で行う必要がある．本研究の場合，Raspberry Pi を搭載する箱の作成や，ケーブルの配線等に時間を費やした．

OS や MPI ライブラリ等のソフトウェアプラットフォームについては，マシン 1 台分を手作業で行い，他のマシンについてはディスクイメージをコピーし利用することができるため，台数を増やしてもコストが大きく増加することはない．

クラスタを構成するには，クラスタを構成するマシンのネットワーク上のアドレスをマスタマシンが全て知っている必要がある．MPICH を使用する場合，使用する全てのマシンの IP アドレスのリストファイルを，管理者がマスタマシンに用意する．リストは 1 度作れば同じ環境を利用し続けられるが，例えばクラスタにマシンを追加したときや，マシンの台数を適宜変化させて実験を行いたい時などにはリストを再度作成する必要がある．そこで，リストを自動作成するシステムを作成した．

3.1 IP アドレスリストの自動作成

作成したシステムは，マスタマシン用のプログラムとスレーブマシン用のプログラムで構成される．マスタマシンとスレーブマシン上のプログラムの動作について説明する．

マスタマシンのプログラムは，起動後に自身の IP アドレスを IP アドレスリストの先頭に書込む．スレーブマシンのプログラムは，マスタマシンに対し TCP で所定のメッセージを送る．これは，このシステムで使用しているプログラムであるということを示すもので，”ADD Node”というメッセージを送るようにした．マスタマシンのプログラムは，メッセージを受け取ったら接続してきたスレーブマシンの IP アドレスを記録する．その IP アドレスが既にリストに存在しなければリストに追加し，”OK”というメッセージを返信する．スレーブマシン側は，メッセージを受け取ったらプログラムを終了する．

これらプログラムは，OS 起動時に自動起動するように設定した．全ての Raspberry Pi の電源を入れた後，マスタマシン上に，マスタマシンと全てのスレーブマシンの IP アドレスリストが作成されていることを確認した．

3.2 並列計算以外でのリストの利用

IP アドレスリストは，並列計算以外にも，MPI プログラムやその他のプログラムのインストール，再起動やシャットダウン等のコマンドの送信等に利用できる．

MPICH を使った MPI プログラムは，クラスタを構成する全てのマシンの同じパスに実行ファイルを置く必要がある．そこで，リストを利用し scp コマンド等を使用することで，全てのマシンの同じパスにプログラムをコピーできる．

ただし，マシン毎に別のプログラムを配置するようなチューニングを行う際には単純なリスト利用法が使えない．その場合，プログラム毎にどのマシンにインストールするか設定し，その設定に基づいてコピーを行う等の方法が考えられる．

4. 今後の展望

4.1 クラスタを構成するマシンの追加

今回の実験では 24 台の Raspberry Pi を用いてクラスタを構築した．今後は，クラスタを構成するマシンの台数を増やし性能評価を行っていきたい．また，クラスタを構成するマシンに異なるベンダーのマシンを追加し，ヘテロジニアスな並列計算機環境を構築していくことを検討している．その際，Raspberry Pi よりも高価ではあるが高性能である Pandaboard[4] を使用することを考えている．

Pandaboard は ARM アーキテクチャのコア OMAP4460 を搭載したボードである．価格は 25000 円〜27000 円程度で，コアの動作周波数は 1.2GHz である．Raspberry Pi と同様に，ARM 向けにビルドされた linux 系の OS で動作させられる．価格が Raspberry Pi と比較すると 5 倍程度になるが，それでも一般的な PC と比較すれば安価であると言える．

Raspberry Pi で行った方法と同等の手間でセットアップ可能であり，PC よりも安価であるという利点を生かしたままクラスタの性能の向上を期待できる．

4.2 クラスタ構成の動的な変更

今回作成したシステムは簡易的なもので，IP アドレスをリストに追加した後にその IP アドレスを持つマシンをネットワークから切断した場合，リストの再構成は行われない．クラスタを構成するマシンを動的に変化させるには，マスタマシンはリストの中の IP アドレスが使用可能であるか定期的に検査する必要がある．

また，リストに対する操作を，操作用 PC において GUI で行うことも考えられる．各マシンをネットワークから切断せずにリストの内容を変更することによって，クラスタを構成するマシンの台数を動的に変化させることが可能になる．これを，以下で紹介するシステムに組込むことも視野に入れている．

5. 関連研究：通信及び CPU 使用率のモニタシステム

本研究に関連するシステムとして，中京大学の林が，クラスタを構成する各マシン間の通信と，各マシンの CPU 使用率を可視化するシステムを開発した [5]．プログラマは，このシステムを使うことで，自身の書いた並列プログラムの通信と処理を視覚的に解析できる．

並列計算プログラムは，各マシンに均等に処理を分散し，それらマシンが均等に処理を行うことで，処理能力を高くすることが期待できる．つまり，並列プログラムは，クラスタ内のマシン間での処理に偏りがあると本来の性能を発揮できない．例えば，あるマシンのプロセッサが高負荷の処理を行い，他のプロセッサが休んでいるというような状態や，通信が 1 対 1 でしか発生しないという状態であると偏りがあると言える．また，通信量をできるだけ少なくし通信オーバーヘッドを減らすことも必要である．

そこで，このシステムでは，マシン間の通信量と，それぞれのマシンのもつプロセッサの使用率をグラフィカルに表示する．これにより，並列プログラムの偏りを，プログラムの動作中に一見して判断することが可能になる．

5.1 システム構成

システムの構成を図 3 に示す．このシステムは，操作用 PC 上のプログラムと，クラスタを構成するマシン上のプログラムから成る．

このシステムは，操作用 PC 上のプログラムから操作する．操作用プログラムは，ユーザインタフェースを持ち，パケットキャプチャと各マシンの CPU 使用率を算出する機能を持っている．

クラスタを構成するマシン上のプログラムは，操作用 PC 上のプログラムから発行されたメッセージに応じて，自身

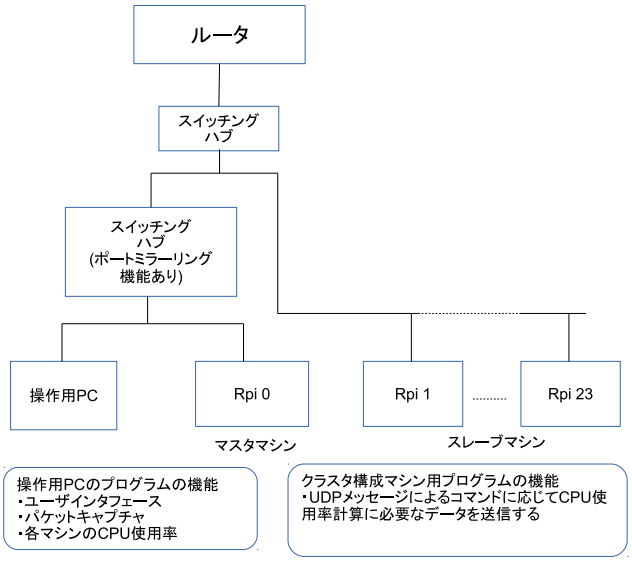

図 3 並列計算モニタシステムの概要

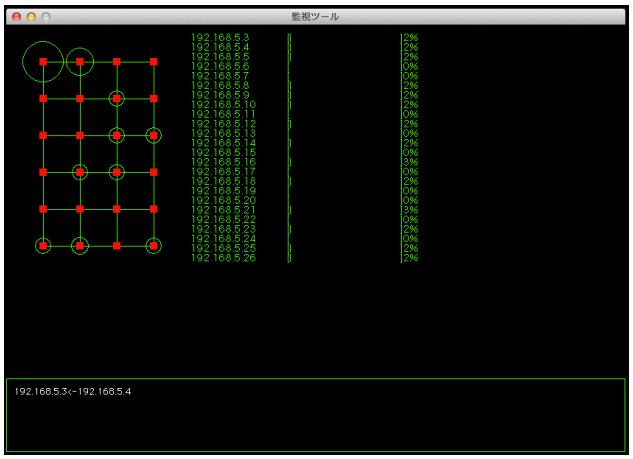

図 4 並列計算モニタシステムのユーザインタフェース

の CPU 使用率の算出に必要な情報を返信する．操作用 PC 上のプログラムは，全てのクラスタ構成マシンからの返信を受けとり，それぞれの CPU 使用率を算出してディスプレイに表示する．

5.2 ネットワーク構成

ネットワーク構成は図 2 とほぼ同様だが，操作用 PC と Rpi 0 はポートミラーリング機能を持つスイッチングハブに接続する．クラスタ内のネットワーク内の通信を監視するために，このプログラムは TCP パケットをキャプチャする．通常のスイッチングハブを用いると，操作用 PC を接続したポートへのパケットだけしかキャプチャできない．そのため，ポートミラーリング可能なスイッチングハブを用意するなど，パケットを取得する方法を用意する必要がある．

ネットワーク内の全ての通信を監視する必要はなく，マスタマシンの通信を監視すればよい．理由としては，偏りが無いプログラムであれば，マスタマシンはジョブを生成し，次に全てのスレーブマシンに対して均等に分配するはずであるからだ．

5.3 ユーザインタフェース

ユーザインタフェースを図 4 に示す．図の上部左側にある 24 個の四角形は，クラスタを構成している 24 台のマシンを表している．

この四角形をマウスでクリックすると下側の枠内に，そのマシンがどこと通信しているかを，それらの IP アドレスと通信方向を表す矢印で表示する．

クラスタ内のあるマシンが別のマシンと通信を行うと，それぞれに対応する四角形を中心に線で円を描く．この円の大きさは通信量を表し，一定時間あたりの通信量が多ければ円が大きく，通信量が少なければ円が小さく描かれる．

右側には CPU 使用率を表示している．使用率は，バーの長さと数値で表示し，全てのマシンの CPU 使用率を確認することができる．

6. まとめ

Cox らの手法に基づき，24 台の Raspberry Pi を使用することで，安価なコンピュータクラスタを構築した．クラスタを構成するマシンの IP アドレスを自動でリスト化することで，クラスタ構築の手間を軽減した．

現状では，クラスタを構成するマシンを動的に変化させるまでは至っていない．そこで，IP アドレスリストに対する操作を操作用 PC において GUI で行い，クラスタの構成を動的に変化させることを提案した．関連研究として，並列プログラムの動作を監視し，並列プログラムの開発を補助するシステムについて紹介した．

参考文献

[1] Simon J. Cox, James T. Cox, Richard P. Boardman, Steven J. Johnston, Mark Scott, Neil S. O'Brien:Iridis-pi: a low-cost, compact demonstration cluster, Cluster Computing, DOI, 10.1007/s10586-013-0282-7(2013)
[2] Raspberry Pi: http://www.raspberrypi.org/downloads
[3] MPICH — High-Performance Portable MPI: http://www.mpich.org/static/tarballs/1.4.1p1/
[4] Pandaboard: http://pandaboard.org/
[5] 林瑶太，並列計算効率化のための通信モニターシステムの開発，中京大学情報理工学部情報システム工学科卒業論文，2014