

フィジカルコンピューティングを活用した、 並列プログラミング教育環境の構築 *

堀内泰輔*¹・宮寄敬*²

Construction of Parallel Programming Education Environment using Physical Computing

HORIUCHI Taisuke and MIYAZAKI Takashi

キーワード：フィジカル・コンピューティング，並列プログラミング，スーパーコンピュータ，Raspberry Pi，クラスタ

1. まえがき

並列プログラミングはこれまでに科学技術計算を中心に発展してきた。最近では、CPU 技術の限界やネットワークの高速化、さらには機械学習に代表される AI 技術からの要請などにより、欠かせないものとなっている。

この教育を行うためには、その実習環境が必須となる。しかし、スーパーコンピュータの利用がコスト的に無理であることはもちろん、PCで行うことも運用上やそのスケールの大きさから問題が多く、これまではあまり試みられて来なかった。

しかし、昨今は Raspberry Pi (以下、ラズパイと略す) に代表される安価で小型の高性能マイコンが登場しており、これを用いた教育用のクラスタは非常に実現しやすいものになって来ている。

本論では、16 台のラズパイを用いて効果的な実習ができる環境の構築を開始したので、ここに報告する。

2. 本研究の背景

著者らはこれまでに、高専にふさわしいフィジカルコンピューティング教育のための多くの教育システムを開発し、授業実践を行ってきた。最近では、科目「フィジカルコンピューティング」(4 年生選択科目、後期 1 単位)において、開発したラズパイ実

験システムを基本に、学生の自主的な活動に重点を置いた総合実習を行い、成果を上げて来た¹⁾。

この実習では学生の自由な発想を基にしてラズパイを用いたシステムの設計・製作を行い、成果物を発表させる形式を採っている。最近では AI を用いた作品テーマも多く見られるようになり、機械学習のためのラズパイの実行速度が問題になる場面も生じてきている。

これには、並列プログラミングの知識が前提となるが、高専のカリキュラムには専門学科を除いてこれを行う科目は存在しない。よって、並列プログラミング教育の必要性を痛感して、このためのデモンストラシステムの構築ならびに並列プログラミング実験のカリキュラム作成を行うこととした。

3. 先行事例と構築用システムの現状

3-1 Raspberry Pi クラスタの先行事例

ラズパイの登場以来、多くのクラスタの事例が存在するが、本格的なクラスタとしては Cox らが[2]で発表した 64 台構成のものがある。

これを母体として、信田³⁾らは並列計算環境構築の一部を自動化し管理コストを減らした初学者向けの教育用環境を目指した。

さらに、米国オークリッジ国立研究所では一般市民をターゲットに TinyTitan というビジュアル性を持たせたシステムを構築して展示している[4]。これを図 1 に示すが、9 台のラズパイを用い、実行状態を各台の大型カラーLED と、マスターPC に接続されたディスプレイを使って実行状態の理解が深まるように工夫されている。さらに、一般の人々が同じものが自作できるように、GitHub 上でハードウェアと

* 2019 年 3 月 17 日 日本産業技術教育学会第 35 回情報分科会研究発表会で報告

*1 一般科特命教授

*2 電気電子工学科教授

原稿受付 2019 年 5 月 20 日

ソフトウェアの両方を公開するオープンハード/ソフトの形態を採っているところも興味深い。

3-2 クラスタ構築システムの現状

並列プログラミングのための規格としては、MPI が有名であり、現在のクラスタのほとんどは、これを実装した MPICH と Open MPI を用いている。これらは C 言語でのプログラミングを前提としており、Open MPI を用いた先行事例としては、[2]³⁾[4]の他に[5]がある。これは出版されていることもあり、今回開発したデモシステムはこれを基にしている。

MPI のもう一つの実装例は mpi4py であり、これはラズパイのデフォルト言語である Python を用いたものである。最近の Web 上の事例では、これを用いたものが多い。

前述の総合実習では言語として Python を用いることを原則としているため、実際のカリキュラムには、これを取り込む必要がある。

4. デモシステムの構築

4-1 ハードウェア構成

ラズパイには多くの種類があるが、今回はクラスタの目的でもある高速性の必要から、最新版の Raspberry Pi 3 Model B+を採用した。これはクアッドコア（4コア）で 1.4GHz で動作する。台数は多いほど好ましいが、デモ用であることを考慮して、16台とした。

先行事例を見ると、LAN ケーブルや USB ケーブル（電源用）の取り回しに苦労している様子が見られたため、できるだけ細く短く弾力性のあるものを選定した。デモシステムとしての考慮事項の一つに携帯性がある。よって、駆動用 PC を除いて、一体化する必要から、小型のスチールラック（3段）を用いた。1段目にはスイッチングハブを、2段目にはラズパイ 16台を積層型ケース（4台用）を4つ用いて設置した。また、構築時に必要となるインターネット接続用ルータも奥に格納した。3段目（最上段）には電源（USB 用）を載せることになるが、駆動用ノート PC を載せられるように、電源は3段目の下側にぶら下げる方式とした。

さらにデモ効果を高めるために、TinyTitan からヒントを得て、カラーLED マトリックスパネルを全面に取り付けることとした。これにより、実行状態をノード（各ラズパイ）ごとにビジュアルに理解できる。

以上、購入物品一覧を表1に、デモシステムの全景を図2に示す。

4-2 ソフト構成

16台のラズパイのOSには多くの選択肢があるが、



図1 TinyTitan 全景

表1 デモシステム 購入物品一覧

品名	規格	購入先	単価	数量	金額
マイコン	Raspberry Pi 3 Model B+	KSY	4,860	16	77,760
積層型ケース	Raspberry Pi 専用保護用クリア・アクリルケース	Amazon	1,889	4	7,556
Lan ケーブル	エレコム 0.3m 爪折れ防止コネクタ やわらか CAT6準拠	Amazon	220	8	1,760
Lan ケーブル	エレコム 0.5m ツメが折れない やわらか CAT6準拠	Amazon	368	8	2,944
USB ケーブル	A オス-マイクロB オス 0.3m	秋月電子	110	16	1,760
USB急速充電器	Anker PowerPort 10 (60W 10ポート USB 急速充電器)	Amazon	3,799	2	7,598
SDカード	東芝 microSDHC 16GB 100MB/s THN-M203K0160 UHS-I	Amazon	675	16	10,800
スイッチングハブ	TP-Link 24ポート TL-SG1024D 10/100/1000Mbps	Amazon	11,900	1	11,900
スチールラック	Doshisha ルミナスメタルラック 3段	Amazon	2,975	1	2,975
RGB LED マトリックスパネル	16x32 RGB LED マトリックスパネル	スイッチサイエンス	3,499	1	3,499



図2 デモシステム全景

今回は最新版の Raspbian (Stretch, 2018/11/13) を採用した。

なお、駆動用 PC 側は Linux であれば任意のものを選択できるが、今回は、本機のラズパイとの相性を考慮し、通常の PC 上に搭載できる Raspbian Stretch with desktop を選定した。これは、USB メモリ上に構築できるため、Windows ノート PC に USB メモリを挿入した状態で Raspbian を起動でき、デモとしても理解しやすいものとなった。

次に、MPI の実装に関しては、[5]の事例をそのまま採用することとした。これにより、C 言語による Open MPI のための各ライブラリをインストールが必要となる。

4-3 デモシステムのネットワーク構成

図3に、デモシステム全体のネットワーク構成を示す。

16 台のラズパイはそれぞれに入出力機器を接続することは無意味であるから、駆動用 PC から SSH 接続することになる。

16 台のラズパイは、1 台をマスター、それ以外をスレーブ 15 台に割り当てる。並列プログラムの起動はマスターが受け持ち、自身の 4 ノードと 15 台の総計 60 ノード、つまり最大 64 ノードに対しての実行が可能となる。

4-4 デモシステムの構築手順

最初に駆動用ノート PC 用に Raspbian Stretch with desktop システムを USB メモリ上に構築した。

次に、マスター用 SD カードに Raspbian をインストール後、駆動用ノート PC にて SSH でマスターを制御できるようにした。そして、マスター上に Open MPI をインストールした後に、円周率計算のための並列計算プログラムを書き込み、ノードを 1 ~ 4 個に変化させながら、ループ回数も変えながら、実行所要時間を計測し、並列処理の効果を確認した。

以上でマスター内のシステムが完成したので、SD カード全体を 15 枚のカードにそれぞれコピーした。これらの内容はほとんど同じであるが、IP アドレスやホスト名は変更する必要があるため、各スレーブに SD カードを装着して起動後にその設定をそれぞれ行う必要があった。さらに、構築や運用を容易にするために、SSH 実行時の認証鍵の作成や、データ共有のための NFS の設定も行った。

以上により、16 台すべてに実行環境が整ったことになるので、マスターより、スレーブの台数（ノード数）を増やしながらか実行時間の計測を行った。

5. デモシステムの評価

図4に、円周率のループ計算を、30,000 回、300,000 回のそれぞれの反復回数にて、1 ~ 64 ノードで並列実行させたときの、所要時間の変化をグラフ化したものを示す。

まず、反復回数が少ないうちは、ノード数が増えても必ずしも実行時間が短縮されない箇所が見受けられる。これは、メッセージ通信レベルで何らかの遅延が生じた結果と思われる。したがって、回数を 30 万回にした結果、きれいな減少曲線となることを確認した。

ノード増加に伴う実行速度短縮の効果は、約 10 ノード以降でグラフが平坦になってしまう結果となった。これは、計算環境ではなく MPI の通信環境が原因であることは言うまでもない。これを改善するには、高性能なスイッチングハブを利用することが必要となる。

なお原著[5]では古いモデルである Raspberry Pi の 2 と 3 (16 台総計 64 コア, 76.8GHz) を用いているが、本システムでは最新モデルであるため、64 コア, 89.6GHz を達成できたことになる。

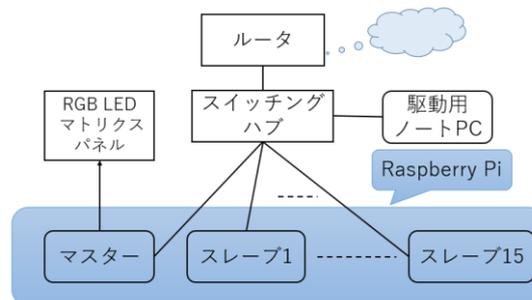


図3 ネットワーク構成

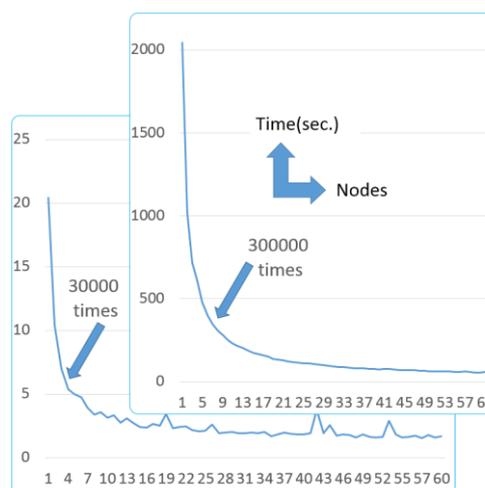


図4 ノード数増加に伴う実行時間の変化

6. 教材製作の構想と課題

今後は 2019 年度の授業での実験カリキュラムを構築する必要がある。

前述のように、総合実習では Python 言語の利用を原則としているため、デモシステムのような C 言語での実験ができない学生が学科によっては存在する。よって、Python 実装である mpi4py を用いる必要がある。

実験順序としては、①各学生に貸与されたラズパイ 1 台内での並列処理実験 (1～4 ノード)、②隣の席の学生と共同して、5～8 ノードでの並列処理実験、③次第に学生数を増加させていき、最後に全学生 20 名による 80 ノードでの並列処理実験、が考えられる。

この実験を有線 LAN 環境で行うのは非現実的であるので、無線 LAN 環境での実験が可能になるように設定を加える必要がある。もちろん、有線 LAN と比較して通信速度が低下するのは止むを得ないが、実験レベルであるために問題はないと思われる。

デモシステムの駆動用 PC の OS として USB メモリ駆動の Raspbian を採用した。これを複数個用意しそれぞれに今回開発したシステムをコピーするだけで、複数台の PC を用いたクラスターを容易に構築できることになる。この先行事例に⁶⁾があるが、この構築と運用を工夫すれば、通常のクラスでの科目でも並列プログラミングの実習が行えるため、今後の課題としたい。

謝辞 本研究は、科研費 (17K01168) の助成による。

参 考 文 献

- 1) 堀内泰輔, 宮寄敬, 西正明, 山本博章:「高専における, IoT 社会に求められる技術力と創造性を育むフィジカル・コンピューティング教育の実践」, 日産技教 第 61 回全国大会 (信州) 講演要旨集, 情報 2 (1B12) (2018.8)
- 2) Simon J. Cox et al.:“Iridis-pi : “A low-cost,compact demonstration cluster”,Cluster Computing, June (2013)
- 3) 信田圭哉, 長谷川 明生:「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」, 情報処理学会研究報告, インターネットと運用技術 (IOT), vol.2014-IOT-24, no.17 pp.1-4,(2014.2)
- 4) Oak Ridg National Laboratory : “TinyTitan/SPH: Parallel realtime SPH” , <http://tinytitan.github.io/> (2015.4)
- 5) Carlos R Morrison : “Build Supercomputers with Raspberry Pi 3” , Packt Publishing,(2017.3)
- 6) 広島県立広島国泰寺高等学校 科学部物理班: 「KNOPPIX for MPI v1.0 について」, <https://www29.atwiki.jp/kokutaiji-science/pages/11.html/> (2005.10)