

(研究ノート)

## Live USB を用いた並列計算環境の有用性の確認

### Confirmation of the usefulness of the parallel computing environment using Live USB

小宮 全<sup>1</sup>

Zen Komiya

倉山 太一<sup>2</sup>

Taichi Kurayama

小宮 浩美<sup>3</sup>

Hiromi Komiya

#### 要旨

小規模の大学・短期大学では、計算機センターが設置されていないことが多い。本研究では、東京交通短期大学において、コンピューター室を使い並列計算環境を構築し、通常の授業に支障を与えることなく数値計算を実行することが可能であることを示した。また、これらは本学にとどまらず他の同様の大学においても成り立つ可能性があることも、併せて示した。

キーワード： 数値計算、パラメータサーチ、並列計算、Live USB、MPI

#### 1. はじめに

近年、個人で所有するパソコンのスペックが向上し文章・動画の作成等の一般的な作業を実施することに関しては、パソコンの能力不足を感じる機会は大幅に減少してきた。しかし、数値計算を実施する際は、多くの計算機パワーが必要となり使用可能な計算機資源を入手できるかが重要な要件となっている。例えば、ある事象に対して多くの計算機資源を必要とするパラメータサーチの実施を希望しても、数値計算用の計算機資源を所有していない小規模大学・短大においては実施することができない。しかし一方、これらの大学・短大では、コンピューター室や貸し出し用 PC の稼働率はそれほど高くない。そこでそれらの教育用の計算機資源を活用して、数値計算を実施することを考えた。

これらに近いアイデアは古くからあった。比較的成功した例としては、Live CD/DVD や Live USB の教育利用である (園田・八巻 [2010]、内藤・大淵 [2012]、濱田 [2011])。さらに、Live USB を用いて並列計算システムを構築する研究も存在する (八巻 [2008])。

<sup>1</sup> 〒170-0011 東京都豊島区池袋本町2丁目5-15 東京交通短期大学 教授 zen.komiya@toko.hosho.ac.jp

<sup>2</sup> 〒264-0007 千葉県千葉市若葉区小倉町1639-3 植草学園大学 准教授 t-kurayama@uekusa.ac.jp

<sup>3</sup> 〒261-0014 千葉県千葉市美浜区若葉2丁目10-1 千葉県立保健医療大学 教授 hiromi.komiya\_12@cpuhs.ac.jp

## 2. 本研究の着想に至った経緯と研究の目的

東京交通短期大学において、個人所有のパソコンでは実行できない程度の数値計算を実行しようと計画をしても、現実的には短大としてそれらを実行するだけの数値計算用の計算機資源を有していない。また、医療系大学の研究グループメンバーも、同様の問題を抱えていた。大学・短大として、数値計算用の計算機資源を所有していない一方で、コンピューター室や貸し出し用ノートパソコン等の教育用の計算機資源の稼働率は高くなく、計算機資源を遊ばせている状況になっている(表1)。表1の利用可能時間は連続での時間ではない。稼働率の算出の際の分母は授業が実施されている総時間としている。よって、週末や授業終了後から次の日の授業開始までの時間は考慮していない。すなわち、各大学の運用ルールによるが、利用可能時間は申請等により大きくなる。

表1 コンピューターの利用可能時間

	利用可能台数	スペック	発売日	利用可能時間 (時)	稼働率 (%)
A 大学*1	32	Intel Core i3-7100	2017年1月	28.5	20
B 大学*2	64	Intel Core i5-8400	2017年11月	30.0	21
C 大学	16*3	Intel Core i3-8145U	2018年7月	週によって異なる	11
	32 (LL)	Intel Core i5-8600	2018年4月	前期/後期 25.5/24.0	32/36
	32	Intel Core i3-8145U	2018年7月	37.5	非公開

\*1 授業時間後及び日曜日は申請により使用可能

\*2 土日は申請により使用可能

\*3 土日及び授業時間後は申請により使用可能

そこで、本研究ではコンピューター室や貸し出し用ノートパソコンを利用して、通常の授業に支障を与えることなく数値計算を実施することが可能か検証することを目的とした。また、いくつかの先行研究に関して、先行研究で構築された数値計算環境の再現を試みたが、インターネットを經由してLiveUSBのOSイメージを入手することが出来なかったため、筆者らの環境で先行研究に基づいた数値実験を実施することが出来なかった。そのような経緯からも制作物を残すことも目的の一つとする。

## 3. 歩行運動方程式

今回の数値計算の実施可能性の検証の題材として、人間の受動歩行モデルの数値シミュレーションを採用した。具体的な歩行モデルは倉山・小宮 [2022] で再導出し、小宮・倉山 [2022] でヤコビアン形式の導出をした以下の連立微分方程式で表される。方程式中の変数や定数に関しては小宮・倉山 [2022] を参照していただきたい。これらの連立微分方程式の中にある定数 $\gamma$ 、 $M$ 、 $m$ 、 $l$ が数値がパラメータとなる。

$$Ml^2\ddot{\theta} + 2ml^2(1 - \cos\phi)\ddot{\theta} - ml^2(1 - \cos\phi)\ddot{\phi} - ml^2\sin\phi(\dot{\phi}^2 - 2\dot{\theta}\dot{\phi}) - Mgl\sin(\theta - \gamma) - mgl\sin(\theta - \gamma) + mgl\sin(\theta - \phi - \gamma) = 0 \quad (1)$$

$$ml^2(1 - \cos\phi)\ddot{\theta} - ml^2\ddot{\phi} + ml^2\sin\phi\dot{\theta}^2 + mgl\sin(\theta - \phi - \gamma) = 0 \quad (2)$$

### 3.1. 数値パラメータ

前述した定数 $\gamma$ 、 $M$ 、 $m$ 、 $l$ は、数値の制限はあるものの、値を原理的に決定することはできず、数値パラメータとなる。例えば、 $m$ を1から10まで1刻み、 $M$ を10から1000まで10刻み、 $l$ を0.01から1.0まで0.01刻み、 $\gamma$ を0.01から1.0まで0.01刻みとすると $100 \times 100 \times 100 \times 100 = 100000000$ 通りとなる。1回の計算に0.04秒かかるとしても、 $4000000 \text{秒} \div 1111 \text{時間} (46 \text{日})$ 程度となる。このパラメータセットを本番用パラメータセットと呼ぶ。

今回は現実的な時間内での比較を実施するために、パラメータセットは以下のステップとし、総計算パターン数は $256 \times 10 \times 5 \times 5$ より64000通りとした。

変数	ステップ数
$\gamma$	256
$M$	10
$m$	5
$l$	5

### 4. 計算手法

今回は Live USB (Debian) で並列計算環境を構築した (以降このシステムのことを LiveUSB クラスタと呼ぶ)。今回は LiveUSB クラスタとの速度比較のため、LiveUSB クラスタと異なる CPU を搭載した2つの PC (PC-A および PC-B) での計算も実施した。計算に利用したクラスタ及び PC の CPU スペックは表2の通りである。

表2 各システムで使用している CPU 一覧

システム名	CPU 名	コア / スレッド数
LiveUSB クラスタ	Intel Core i3-7100	2コア / 4スレッド
PC-A	Intel Core i7-1280P	6+8コア / 20スレッド
PC-B	AMD Ryzen 9 5900HX	8コア / 16スレッド

LiveUSB クラスタは32ノードであるため最大で128スレッドの並列計算が可能である。Live USB に利用した OS イメージは github で公開している(<https://github.com/zen-komiya/LiveUSB-MPI>)。計算に使用した Python プログラムは小宮・倉山 [2022] で作成したものを MPI を使用して並列化したもので、並列化部分は以下に示した通りである。今回使用したプログラムにはファイルの入出力はないものを採用した。

```

gamma_steps= 256;
M_steps = 10;
m_steps = 5;
l_steps = 5;

gamma_max = 1.0;

```

## Live USB を用いた並列計算環境の有用性の確認

```
M_max = 1000.0;
m_max = 10.0;
l_max = 1.0;

if rank == 0:
    tantou_size, amari = divmod(gamma_steps, size)
    counts = [tantou_size+1 if p < amari else tantou_size for p in
range(size)]
    starts = [sum(counts[:p]) for p in range(size)]
    ends = [sum(counts[:p+1]) for p in range(size)]
    data = [(starts[p], ends[p]) for p in range(size)]
else:
    data = None

data = MPI.COMM_WORLD.scatter(data, root=0)

for igamma in range(data[0], data[1]):
    gamma = (igamma+1)*(gamma_max/gamma_steps)
    for iM in range(M_steps):
        M = (iM+1)*(M_max/M_steps)
        for im in range(m_steps):
            m = (im + 1)*(m_max/m_steps);
            for il in range(l_steps):
                l = (il+1)*(l_max/l_steps);
```

並列化した Python プログラムを見てもわかるように、パラメータサーチを実施するのみであるため並列化にかかるプログラミング作業のコストはほぼない。MPI でのプログラミングはネットワークの通信料が増えることが多いが、パラメータサーチはネットワークが十分ではないコンピューター室等に適している。

### 5. 結果

LiveUSB クラスタ、PC-A および PC-B を使って数値計算をした結果を表3に示す。表3の並列数とは、MPI の実装である MPICH を実行する際のプロセス数を表している。また、計算時間は秒数を示しており、すべての並列数に対して数値計算を10回ずつ実施し、その平均値となっている。参考値として同じプログラムを別の PC (PC-A および PC-B) で実行した結果を示した。PC-A および PC-B に関しては、表2に示してあるようにスレッド数がそれぞれ20スレッド、16スレッドとなっているため、並列数を16までの計算にとどめた。これらの数値も10回実行した平均値を示している。

表3 各システムでの MPICH の並列数と計算時間 (秒)

並列数	LiveUSB クラスター	PC-A	PC-B
1	2721.9	1414.7	2155.6
2	1360.6	743.0	1463.2
4	681.6	436.7	724.0
8	341.7	317.4	387.1
16	170.5	259.9	244.0
32	85.3	---	---
64	42.9	---	---
128	39.6	---	---

## 6. 考察

表3の結果より、並列数が1、すなわち並列計算をせず単体の計算としての計算能力は LiveUSB クラスターより PC-A および PC-B の方が高いことが分かる。次に LiveUSB クラスターの並列数に対する計算時間の変化について見てみる。表3より並列数と反比例して計算時間が減少していることが見て取れる。これは今回の計算がパラメータサーチであるため並列度を高めればそのまま直接的に計算時間が減少するため当然の結果ではあるが、ほぼ理論値を出すことができたことが分かる。ここでの理論値とは、プログラムの全実行時間を並列化できたと仮定した値のことである。しかし、並列度が128の場合は、理論値通りとなっていない。並列度が128というのは、各ノードの4スレッドをすべて計算に使用することを意味しており、当然オーバーヘッド分だけ計算が遅くなるため、こちらも当然の結果である。次に PC-A についてだが、表3には6+8コアと表示してあるが6コア（2スレッド）は P コア（高性能コア）であり8コア（1スレッド）は E コア（高効率コア）のハイブリッド・アーキテクチャとなっている。そのため、表3中では P コアを使用している並列度4まで理論値に近い値となっている。PC-B については、8コア16スレッドであるため並列度8までが理論値に近い値となっている。LiveUSB クラスターは並列度が64の時に、計算速度が63.4倍になっており本番用パラメータセットでは4000000秒 $\div$ 1111時間（46日）の計算時間がかかっていたものが、4000000秒 $\div$ 63.4 $\div$ 63091秒 $\div$ 17.5時間となる。これは46日間かかっていた計算時間が1日未満で終了することを意味している。並列計算を実施するという観点からは、十分に実用に耐えられる計算時間内で結果を得ることをできるといえるであろう。これに加え通常利用している大学・短大でのコンピューター室や貸し出し用 PC を使用してこれらの計算を実施可能か表1に基づき比較した。表1の利用可能台数とは同時に利用できるコンピューターの台数を表しており、利用可能時間は連続ではないが利用可能な総時間を意味している（ただし連続の場合もある）。また、スペックを見るとわかるように今回構築した LiveUSB クラスターとほぼ同程度であり、今回ターゲットとした大学・短大での計算機のリプレイス頻度も償却年数から考えると同程度である可能性が高いため、本論文での結果はある程度一般性を持つと考えられる。これらの利用可能台数や利用可能時間を考慮すると、本論文で検討した計算量程度であるならば、通常は授業で利用しているコンピューター室や貸し出し用 PC を利用して数値計算を実行することが可能である。通常授業を実施する中で、これらの LiveUSB クラスターシステムを運用するうえで考

## Live USB を用いた並列計算環境の有用性の確認

慮すべき点ある。それは、システムのセットアップと撤収にかかる時間である。本研究では親機と予備機を含めて40台でのクラスターを構築し、そのうちの32台を使用した。また、LiveUSB クラスターが起動している状態から全台の電源が落ちるまでの時間は、USB への書き込み時間にもよるが約10分であった。両時間共に十分に実用に耐えられる時間である。

今回はパラメータサーチを実施することを目的としたので、OpenMP でも同様のパフォーマンスが出たであろう。本研究で MPI を採用した理由は2つあり、1つ目としては今回はパラメータサーチを実施するだけであったので並列化プログラミングのコストがほとんどかからないこと、2つ目は今後の拡張性とチューンナップの効率性を考えてのことである。

### 7. まとめ

本研究により東京交通短期大学での通常の授業に支障を与えることなく、コンピューター室を使い数値計算を実行することが可能であることがわかった。まず最初に Live USB を利用して、コンピューター室に設置してある PC を使って並列計算環境 (LiveUSB クラスター、32ノード) の構築をし、それにより大規模な数値計算を時間短縮することが可能であることを確認した。それに加え実際に学内に設置されているコンピューターで数値計算を実施する際には数値計算環境への切り替えが必要だが、切り替え時間は5分程度であり切り替え時間と計算時間を合計しても利用可能時間に納まる可能性が高いことも確認できた。さらに、中小規模大学のコンピューター室の設置計算機や利用可能時間に関して、東京交通短期大学と同じような条件であることから、ある程度一般性を持たせることも可能であることが示唆された。今回の数値計算を実施したタイミングは、土曜日の授業後から月曜日の授業開始前、夏季休暇中、春季休暇中に実施であったが、多くの大学・短大のコンピューター室の最低台数が40台程度であることより32台で1セットの並列環境を構築するのが都合よく、4世代程度前のハードウェアになりがちなコンピューター室のスペックであれば、最新のパソコンと比較して十分なパフォーマンスを出すことができ、十分使用に耐えられることがわかった。

また、Live USB クラスターはコンピューター室のハードウェアに依存しないため、ハードウェアのリプレイスに伴い、自動的に Live USB クラスターの並列環境のアップデートになるため、非常に都合がよい。しかし、最近の中小規模大学の傾向として、学生がノートパソコンを持参し、コンピューター室はパソコンラボのような活用方法も検討されているため、そのような形態になった場合は、Live USB を使用した本システムを利用することができなくなる。

次の課題としては、本番パラメータセットでの計算を通常授業期間に実施し、稼働率の低いコンピューター室の利活用の実践することである。

本研究の一部は科研費21K10596の助成を受けたものです。

### 8. 参考文献

八巻 宏太 [2008]、「USB メモリを用いた PC クラスタ自動構築システムによる教育用 PC 80台を利用したクラスタ構築実験」、『平成20年度電気関係学会北陸支部連合大会講演論文集』、電気学会  
園田 潤・八巻 宏太 [2010]、「理工学系教育に柔軟に対応できる Live Linux 自動再構成システムの開発と情報処理教育への適用」、『電気学会論文誌A』、130巻1号、電気学会、pp.74-80

内藤幸一郎・大淵慶史 [2012]、「USB-KNOPPIX MATH を利用した計算機支援環境の実践的構築能力と活用能力促進教育」、『工学教育研究講演会講演論文集』、pp.120-121

濱田龍義[2011]、「USB 起動 KNOPPIX/Math/2010について USBbootableKNOPPIX/Math/2010」、『数理解析研究所講究録』、第1759巻、数理解析研究所、pp.74-80

小宮全 [2023]、「LiveUSB-MPI」、(<https://github.com/zen-komiya/LiveUSB-MPI>)

倉山太一・小宮全 [2022]、「歩行運動方程式の導出過程-Simplest walking model-」、『支援工学理学療法学会誌』、1巻1号、日本理学療法学会連合、pp.64-72

小宮全・倉山太一 [2022]、「歩行運動方程式の数値計算方法-Simplest walking model-」、『支援工学理学療法学会誌』、1巻1号、日本理学療法学会連合、pp.73-78