

スマホでグリッドプロジェクト 2022

—1万台で大規模分散処理—

石川 ナディーム[†] 野村 勇登[†] 依田 侑也[†] 植月 修[‡] 福永 圭佑[‡]

名古屋 清次[‡] 佐原 潤哉[‡] 石畑 宏明[†] 瀬之口 潤輔[†]

東京工科大学 コンピュータサイエンス学部[†] 伊藤忠テクノソリューションズ株式会社[‡]

1. はじめに

近年、AI やシミュレーション技術の発展により、インターネット上でやり取りされるデータが増大している。それに伴い、データセンターでは、膨大なデータを処理するために、機器の小型化が進み、機器の高密度化が進んだ。高密度な機器を効率的に利用するためには機器を冷却する必要があり、電力消費量が増大している。電力消費量の問題は、コストのみではなく、環境問題にも発展する。この問題に対してスマートフォンなど冷却の必要のないデバイスを用いての分散処理は有用である。

本プロジェクトでは前年度までに、スマホの計算リソースとしての価値を確認している [1]。

本研究では前年度までの分散処理システムを改良した本システムにて、大規模に分散処理を行った。スーパーコンピュータと処理性能を比較する事で、本システムを評価した。

2. 提案手法

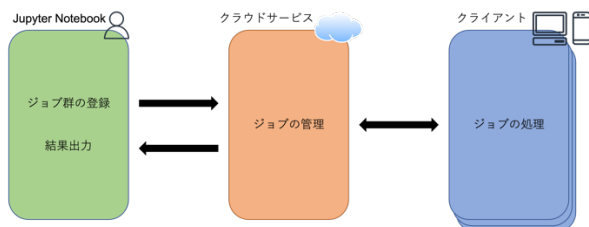


図 1：分散処理アーキテクチャ

図 1 のように本システムでは、Jupyter Notebook 上で独自制作の並列化ライブラリを利用することで、並列可能な演算処理をジョブ（小処理）に分け、システムに登録する事ができる。このライブラリは、既存の並列化ライブラリである Joblib に沿った形で作成されており、

導入を容易にする事が期待される。クライアントは、クラウドサービスを介して登録されたジョブを取得し、処理を実行する。

3. 評価実験

3.1. 実験手法

本研究では株価予測プログラム [2] を本システム上と、東京大学情報基盤センターの Oakbridge-CX スーパーコンピュータシステム上にて動かし、処理速度を比較する。

株価予測プログラムは、独自に用意したデータセットを用いて、決定木の学習と翌日の TOPIX の先物価格の対数変化率の予測を 1 万日分を行う。予測は日毎に独立しており、並列化する事ができるため、ジョブの分割単位は 1 日分の予測とする。株価予測プログラムのパラメータは、それぞれ予測日数を 1 万日、GA（遺伝アルゴリズム）の世代数を 10 世代、CMA-ES（共分散行列適応進化戦略）の世代数を 20 世代、学習件数を 2500 件と固定して実験を実施した。このパラメータは、1 ジョブあたりの処理時間に線形に関与する。

表 1：コンテナとスマホの処理時間の比較

ワーカー名	処理時間 (s)
コンテナ	20.5
iPhone11	18.8
iPhone12mini	17.2
Google Pixel 7	19.8
iPad Air第3世代	20.3

本研究では、AWS のコンテナサービスである、Fargate にて提供されるコンテナ 1 台をスマホ 1 台相当として実験を行う。そのため、株価予測シミュレータでの実験をする前に、コンテナとスマホの処理速度の比較実験を行い、コンテナの性能を評価する。本システム上で、1 台のワーカーが 100 万番目のフィボナッチ数列を見つけるプログラムの処理を 3 回を行い、平均した速度を比較した。スマホは、iPhone11, iPhone12mini, Google Pixel 7, iPad Air 第 3 世代をそれぞれ用いて、実験を行った。実験結果は表 1 のように、全ての

Project for Grid Computing with Smartphones 2022 - distributed computing with 10,000 units -

Nadeem Ishikawa[†] Hayato Nomura[†] Yuya Yoda[†]
Osamu Uetsuki[‡] Keisuke Fukunaga[‡] Seiji Nagoya[‡]
Junya Sawara[‡]

Hiroaki Ishihata[†] Junsuke Senoguchi[†]

[†] School of Computer Science, Tokyo University of Technology

[‡] ITOCHU Techno-Solutions Corporation

スマートフォンがコンテナの処理速度を上回っており、差も少ないため、コンテナ 1 台をスマホ 1 台相当として本研究では扱う。

Oakbridge-CX は、各ノードに 56 コアの CPU があるため、1 ノード約 50 プロセス（ワーカー）として実行した。コンテナ側のワーカー数はコンテナ数に一对一に対応する。

3.2. 実験結果

表 2：コンテナによる処理時間の比較

ワーカー数	処理時間(s)	理論値(s)
1000	5929	5330
2000	3219	2665
5000	1662	1066

表 3：Oakbridge-CX による処理時間の比較

ノード数	ワーカー数	処理時間(s)
2	100	26235
10	500	5349
20	1000	2789
40	2000	1417
100	5000	627
200	10000	420

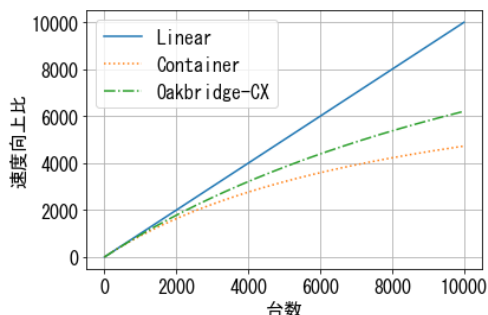


図 2:アムダールの法則による性能予測

まず、コンテナ数増加に伴う処理時間の変化を評価する。表 2 の理論値は、コンテナ 1 台が 1 ジョブを処理した処理時間が 533 秒だったことから、1 ジョブあたり 533 秒で処理されると仮定をして算出している。コンテナの処理における、全てのコンテナ数のケースにおいて、実際の処理時間は理論値にコンテナ 1 台が 1 ジョブを処理した処理時間と 50 秒程を加算した値となっている。これはシステム上でジョブの割り当てがうまくいかず、コンテナ間で処理するジョブが被ったことで発生する余りのジョブの処理時間や通信時間、分散処理に関わるジョブ管理等のオーバーヘッドが存在している事が原因だと考えられる。

次に本システムを使ったコンテナによる処理と Oakbridge-CX 上での処理の処理時間を比較する。本システムを用いてコンテナ数 5000 台で処理をした処理時間は 1662 秒であった。これは

Oakbridge-CX 上で 20 ノード、1000 ワーカーにて処理を行った処理時間（2789 秒）よりも少ない時間である事がわかる。本システムではスーパーコンピュータのノードの一部を使い処理を行うような大規模演算処理を処理するポテンシャルがあるといえる。本システムを使ったコンテナによる処理にて、1 万ジョブに対してコンテナ 1 万台で処理をしたときの処理時間をアムダールの法則に従うものとして導出し、1129 秒であった（図 2）。この値は、Oakbridge-CX 上で 40 ノード、2000 ワーカーにて処理を行った処理時間（1417 秒）よりも処理が早く終わることを示している。実際に本システムがワーカー 1 万台を捌く事ができる事は、負荷テストツールである Locust を用いて確認した。

ジョブ数と同数以上のワーカーで処理をするとき、処理時間はワーカー自体の性能に大きく依存する。本システムでは、ワーカーとして主にスマートフォンやサーバー等を想定している。特にスマートフォンの性能は年々かなり上昇しており、オーバーヘッドの効率化等システムの改善をする事で、今後現在のスーパーコンピュータで行うような大規模な演算処理をより効率的に行うことが期待される。

4. おわりに

前年度までのシステムを改良して、実際に大規模に分散処理を行い、大規模な分散処理に耐えるスケーラビリティを確認した。スーパーコンピュータと性能を比較した結果、ワーカー数を増やす事で、スーパーコンピュータのノード数によっては、近い性能での処理が期待される事がわかった。

本システムを利用することで、限りある資源を有効活用でき、低コスト、低電力で演算処理を実行できる。

今後はシステムのオーバーヘッド等のさらなる効率化を行い、AI やシミュレーションを用いた実際の問題解決に活用する。

参考文献

- [1] 瀬之口潤輔. “多進木 Bi-Level GA によるノイズを含む空間の抽出と複雑系データの予測”, 人工知能学会論文誌 Vol. 36, No. 5, pp. F-L52_1-7, 2021.
- [2] 小谷晏経, 島田雄気, 柳川陸, 植月修, 福永圭佑, 名古屋清次, 佐原潤哉, 瀬之口潤輔, 石畑 宏明. “スマホでグリッドプロジェクトーみんなのスマホで大規模計算ー”, 第 84 回全国大会講演論文集, Vol. 2022, No. 1, pp. 405-406, 2022.