

## 量子回路シミュレータの性能分析と考察

青木望美†

山崎雅文‡

平井聡‡

山岡茉莉‡

福本尚人‡

小口正人†

†お茶の水女子大学

‡富士通株式会社

## 1 はじめに

近年、量子化学計算など一部の計算について非常に高速な計算を実現できる可能性があるとして量子コンピュータが注目されている。量子コンピュータの計算の高速化を可能とする最大の特徴に、量子ビットがある。従来型コンピュータでは最小単位のビットを0または1のどちらかで表すが、量子ビットでは0と1の重ね合わせ状態を保持することが可能であり、量子コンピュータではこの量子ビットの重ね合わせ状態を利用することで解が出力させる確率を高め、結果を出力する。しかし、現在も複数の量子コンピュータが存在するものの計算にノイズが含まれ、これにより結果にエラーが含まれるなど、理想的な挙動の実現には至っていない。

ここで、従来型コンピュータ上で量子コンピュータの挙動を表現する量子回路シミュレータが存在する。量子回路シミュレータを用いた場合計算結果はノイズの影響を受けないため、量子アプリ開発などを促進することができると考えられ、量子コンピュータの研究・開発と並行して量子回路シミュレータの研究を行うことは有用である。

本稿では量子回路シミュレータのひとつであるQiskit Aerについて、複数の性能分析ツールを用いてシミュレータ実行中のシステム稼働状況に関するデータを収集し、収集したデータからボトルネックの推測や性能改善方法について考察を行う。

## 2 実験

## 2.1 実験概要

分析対象の量子回路シミュレータにはQiskit Aerを使用する[1]。また、量子回路シミュレータにはいくつかの方式が存在するが、今回はすべての量子状態をメモリに保持し、行列ベクトル積を計算することで量子コンピュータの挙動を表現するState Vector方式を採用した。作成した量子回路には、量子体積モデル回路を用いた[2]。また、量子ビットの個数は31個、量子回路の深さは10とした。この場合、メモリ上に配置されるState Vectorの容量は32GBである。

Performance analysis and consideration of a quantum circuit simulator

†Nozomi Aoki

‡Masafumi Yamazaki

‡Akira Hirai

‡Mari Yamaoka

‡Naoto Fukumoto

†Masato Oguchi

†Ochanomizu University

‡FUJITSU LIMITED

プログラム実行中のシステム稼働状況のデータ収集にはCPU使用率や割り込みに関する情報を取得可能なmpstat、プロセスやメモリ、CPUなどに関する情報を広く取得可能なvmstat、Linux Kernelのイベントデータ収集などが可能なperf stat[3]、CPUのリソース利用などの情報をリアルタイムで取得可能なPCMTools[4]を使用した。

また、実験環境を表1に示す。

表1: 実験環境

|        |  |
|--------|--|
| 実験用サーバ | Primergy RX2540 M1                     |
| OS     | Rocky Linux 8.6                        |
| CPU    | Intel Xeon プロセッサ<br>E5-2697v3(2.60GHz) |
| メモリ    | 128GB                                  |

各性能分析ツールとPythonプログラムの実行について、性能分析ツールの実行開始から約10秒後にPythonプログラムの実行を開始する。また、性能分析ツールはそれぞれ1秒間隔でデータ収集を行う。各性能分析ツールのバージョンは表2に示す通りである。なおmpstatはsysstatパッケージに含まれるツールであるため、sysstatのバージョンを記述する。

表2: 性能分析ツールのバージョン

|         |                           |
|---------|---------------------------|
| sysstat | 11.7.3                    |
| vmstat  | procps-ng 3.3.15          |
| perf    | 4.18.0-425.3.1.el8.x86_64 |
| pcm     | 202107-5.el8              |

## 2.2 結果

Qiskit Aerライブラリを使用したPythonプログラムの実行時間は約261秒であった。

mpstatで収集したデータでは、実験サーバに搭載された計14個の全CPUのユーザ時間の割合(%usr)について、Pythonプログラム実行中に約100%の値となった。よって分析対象のPythonプログラムはCPUレベルでの並列処理は行われていることがわかる。全CPUのユーザ時間の割合の平均値であるCPU=allの%usr値の変化は図1の通りである。

また、vmstatで取得した空きメモリ量のデータから、約33.7GB相当のメモリが確保され続けていることがわかった。

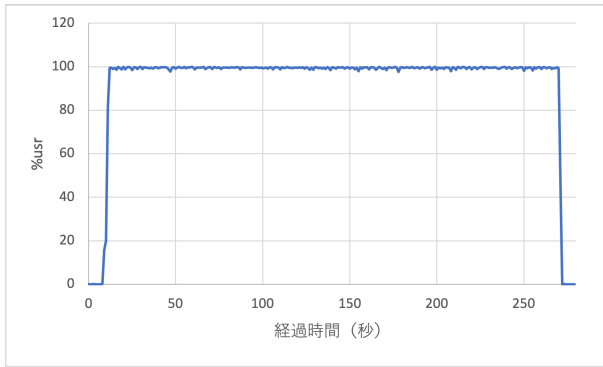


図 1: 全 CPU の %usr 平均値の変化

続いて perf stat で収集したデータについて記述する。perf stat を用いた計測は複数回行い、そのいずれも IPC の値が 3 前後とかなり高い値で推移した。図 2 にコア ID=2 のコアの IPC 値の変化に関するグラフを示す。

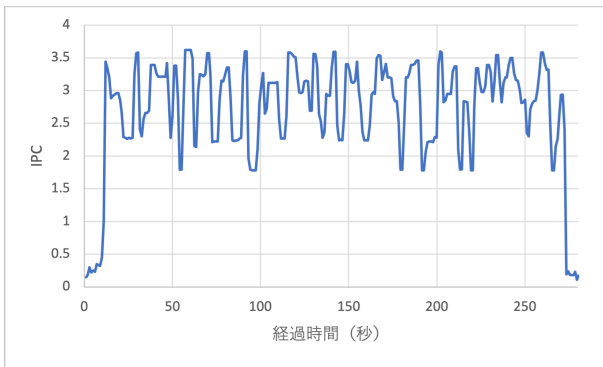


図 2: IPC の変化

また、プログラム実行中の L1 キャッシュ、L2 キャッシュ、L3 キャッシュのキャッシュミス率の全コア平均値はそれぞれ 1.7%、45.2%、65.4%程度であった。

図 3 は、PCM Tools で測定したプログラム実行中のメモリバンド幅の変化である。メモリバンド幅は約 20 GB/s から 40 GB/s の間で変化していた。この値は、使用サーバの理論的なメモリバンド幅である 68 GB/s に対して小さい。また、実効的な最大メモリバンド幅を引き出すことができるとして知られている STREAM ベンチマーク [5] を実行したときのメモリバンド幅は 57 GB/s であり、この数値より小さい。

以上の結果から、Qiskit Aer ライブラリを使用したシミュレータの処理効率は非常に良く、メモリバンド幅は実効的な最大のメモリバンド幅より小さいことがわかった。State vector のシミュレータは行列ベクトル積を繰り返すもので、連続したアドレスに対してメモリアクセスを繰り返すため、メモリバンド幅は更に引き出せる可能性が高い。よって Qiskit Aer はメモリバンド幅の改善を行うことが性能の向上に繋がると考えている。

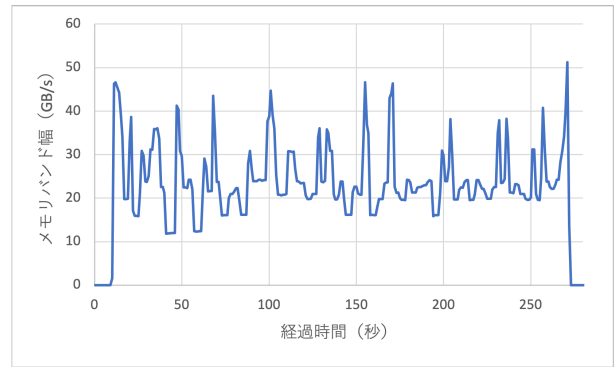


図 3: メモリバンド幅の変化

### 3 まとめと今後の予定

量子回路シミュレータの性能分析を行うにあたり、量子コンピュータ用フレームワークの Qiskit に含まれる Qiskit Aer のライブラリを使用した Python プログラムを用意し、Linux Perf などの性能分析ツールを使用してシステム稼働状況に関するデータを収集した。その結果、Qiskit Aer のライブラリを使用したシミュレータは処理効率が非常に良く、また連続したアドレスにメモリアクセスを繰り返す挙動をするにも関わらずメモリバンド幅の値は小さいため、この点を改善することでプログラムの性能を改善することができると考えられる。

今後は今回分析を行ったシミュレータについてさらなる分析と性能改善手法の模索を行うとともに、他の量子回路シミュレータについても同様の分析を行うことで性能の比較などを行う。

### 4 謝辞

本研究の一部はお茶の水女子大学と富士通株式会社との共同研究契約に基づくものである。

### 参考文献

- [1] Qiskit aer simulator. [https://qiskit.org/documentation/tutorials/simulators/1\\_aer\\_provider.html](https://qiskit.org/documentation/tutorials/simulators/1_aer_provider.html).
- [2] A. W. Cross, L. S. Bishop, S. Sheldon, P. D. Nation, and J. M. Gambetta. Validating quantum computers using randomized model circuits. *Phys. Rev. A*, Vol. 100:p.32328(online), 2019. DOI: 10.1103/PhysRevA.100.032328.
- [3] Perf wiki. [https://perf.wiki.kernel.org/index.php/Main\\_Page](https://perf.wiki.kernel.org/index.php/Main_Page).
- [4] Intel® performance counter monitor - a better way to measure cpu utilization. <https://www.intel.com/content/www/us/en/developer/articles/technical/performance-counter-monitor.html>.
- [5] Stream: Sustainable memory bandwidth in high performance computers. <https://www.cs.virginia.edu/stream/>.