

HPC-Centric Quantum Computing に向けたサービスに向けて

片桐孝洋¹⁾, 高橋一郎¹⁾, 森下誠²⁾, 星野哲也¹⁾, 河合直聡¹⁾, 永井亨¹⁾

1) 名古屋大学 情報基盤センター

2) 名古屋大学 大学院情報学研究科

katagiri@cc.nagoya-u.ac.jp

Toward Services for HPC-Centric Quantum Computing

Takahiro Katagiri¹⁾, Ichiro Takahashi¹⁾, Makoto Morishita²⁾, Tetsuya Hoshino⁴⁾,
Masatoshi Kawai¹⁾, Toru Nagai¹⁾

1) Information Technology Center, Nagoya Univ.

2) Graduate School of Informatics, Nagoya Univ.

概要

本報告では、将来実現すると期待されているエラー訂正可能な誤り耐性量子コンピュータ(FTQC)の実現まで量子関連研究をスーパーコンピュータと HPC 技術を用いて支援し、量子・古典ハイブリッド計算を実現する HPC-Centric Quantum Computing の概念を提案する。また主要技術として、自動チューニング (Auto-tuning) 技術が必要となることを述べる。ケーススタディとして、疑似量子アニーラ、量子回路シミュレータ、コヒーレントイジングマシンの性能パラメータチューニング問題を提起する。

1 はじめに

量子コンピューティングの実用化に期待が集まっている。特に近年、エラー訂正可能な誤り耐性量子コンピュータ(FTQC)の技術進展があり、FTQC が実現した際にシームレス移行できる計算基盤の構築が喫緊の課題となっている。

一方で、量子コンピュータ実機が開発されているが実機は少なく、実利用には多くの待ち時間が生じることが予想される。加えて、運用の観点では、まだまだ利用できる量子ビット数が少なく、量子アルゴリズム検証の面でも極めて不十分である。また FTQC が実用化するには、ハードウェア面でも多くの技術的な困難があるため、10 年以上の開発期間がかかると予想される。

そこで FTQC が実現されるまでの間、量子アルゴリズム開発など基礎研究推進が極めて重要になると予想する。そのため、誤差なく量子演算をシミュレートできる、量子回路シミュレータなどの利活用が不可欠となると著者は予想している。

量子回路シミュレータ技術は、現在、高性能計算 (High Performance Computing, HPC) 分野の主要技術である GPU (Graphics Processing Unit) コンピューティングを活用して処理の高速化が実現されて

いる。現在、現在 40 量子ビット程度までは無理なく、ノイズのない量子回路シミュレーションができるとされる。そのため、FTQC 実機が実用化されるまでの間は、量子回路シミュレータを用いて量子アルゴリズムの性能検証のニーズが増大すると予想する。しかしながら、量子回路シミュレータの実行性能の観点では、まだ研究の余地がある。そのため、HPC 分野において、量子回路シミュレータの高速化に関する研究進展が期待される。

一方で、量子コンピュータおよび量子関連計算で期待される処理の 1 つとして、組合せ最適化問題の求解のための量子アニーリングが期待されている。量子アニーリングは、量子コンピュータ実機に加えて、量子効果を利用しないが量子計算に発想を得て高速化した<疑似>量子アニーラがあり、我が国においては多数の研究開発がなされている。

量子および疑似量子アニーラの利活用において、QUBO (Quadratic Unconstrained Binary Optimization) 形式[1]により問題を定式化する必要がある。しかし、扱う問題とアニーリングアルゴリズムに起因する性能パラメータが内在している。この性能パラメータをチューニングしないと、高精度の解が得られないだけでなく、そもそも求解すらできない。

したがって、利用時の性能問題が生じている。

この一方で、現在運用中のスーパーコンピュータに、現在開発されている量子コンピュータを連結して実現する、量子・古典ハイブリッド計算環境の開発が注目されている。特に、量子・古典ハイブリッド計算においては、スーパーコンピュータと量子コンピュータの双方を簡便に活用できる計算基盤の開発がなされようとしている。

そこで本研究では、現在考えられる量子・古典ハイブリッド計算を効率よく実行するための要素技術について紹介するとともに、スーパーコンピュータ運用サービスへの展開について検討する。

2 HPC-Centric Quantum Computing

2.1 概要

我々は、FTQC 実用化まで量子コンピュータ研究を支援するためには、スーパーコンピュータを含む HPC 環境の積極的な活用をする必要があると考えている。現在、量子コンピュータを中心に置き、HPC 環境を活用する Quantum-Centric Supercomputing [2][3]の提案があるが、そのためには、HPC 環境を積極的に活用しないと実現できない。そこで我々は、**スーパーコンピュータを含む HPC 環境を中心に置き、量子コンピュータをアクセラレータとして活用する HPC-Centric Quantum Computing [4]**の提案を行っている。図 1 に、HPC-Centric Quantum Computing の概念を示す。

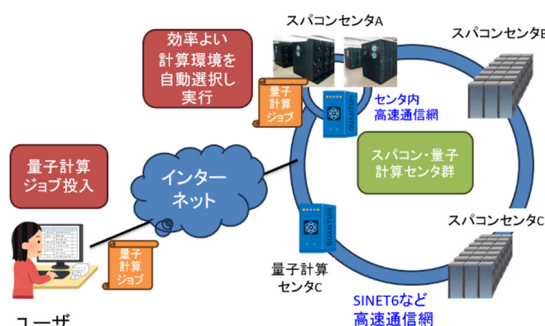


図 1 HPC-Centric Quantum Computing

図 1 では、近い将来的に、スーパーコンピュータと量子コンピュータを連結した計算機センター出現すると想定している。また、それらの計算機センターを SINET6 などの高速通信網で接続した、量子・古典ハイブリッド計算環境が構築されると予想している。

図 1 では、各計算機センターは、いくつかの構成が考えられる。

1. スーパーコンピュータのみ有しているが、外

部の量子計算センターとネットワーク連結したサービスを行う、量子・古典計算センター群

2. 計算機センター内に、スーパーコンピュータと量子コンピュータを有し、専用回線で結合された、量子・古典計算センター群。2024 年現在、理化学研究所がこのような計算機環境の構築を進めている [5]。

このセンターでは、量子と古典それぞれの処理に対して、処理の切り替えが重要になると想定される。そこで著者らが 20 年来研究してきた自動チューニング (AT) 技術を活用し、量子・古典の計算環境をユーザが意識することなく、適する計算機を自動的に切替える仕組みの開発を進める必要があると考えている。この仕組みにより、将来、FTQC の性能が古典環境を凌駕する時代が到来しても、ユーザはプログラム変更を意識せずに量子・古典計算を継続することができる。

3. スーパーコンピュータのみ有しており、他センターの量子コンピュータとの連結サービスをしていないスーパーコンピュータセンター群。

ここで、1～3すべての計算機センターは、何らかの高速ネットワークで連結されていると想定する。この時、ユーザが 1～3 の計算機センターのアカウントを複数有している場合、自分が持っている量子・古典計算プログラムは、1～3 の計算機センターを、量子・古典計算に活用できる点に注意する。このように、適する計算機センターの選択問題が生じると考えている。

この適する計算機センターの選択問題は、ログイン (SSH) 接続、データ移動 (クライアント→センター)、ジョブ投入、結果取得 (センター→クライアント) のジョブ管理の手間の問題であるといえる。つまり、クライアント側で、**ジョブフローの自動化を行うツールなどの仕組み**があれば、低い手間で各計算機センターの資源をできるようになり、計算機センター選択問題を解決できると予想される。

2.2 目的

我々は、図 1 の HPC-Centric Quantum Computing が将来できることを想定し、長期的ビジョンとして、以下の研究開発を行う必要があると考えてい

る。

1. 量子処理を記載することで簡便かつ高性能に実行できる量子・古典計算基盤開発

ユーザの観点では、量子・古典の計算環境を意識せず、高速かつ解の精度が高いようにジョブが自動実行され、計算結果が得られることが最重要である。そのためには、従来のジョブスケジューラが行っている、ユーザが直接どの計算機にジョブを投げるか指定した実行形態では、不十分であると考えている。つまり、ユーザが扱っている問題の大きさや問題の数理的な性質を自動判定し、実行速度や解の精度を予測した最適化により、自動的に適する計算実行環境（量子計算機実機や GPU による量子回路シミュレータの実行など）を切り替える必要がある。

2. スーパーコンピュータの高速化技術開発

量子・古典ハイブリッド計算環境では、古典計算環境の高速化が不可避である。スーパーコンピュータのハードウェアは日進月歩である。現在主流となっているマルチコア CPU や GPU のハードウェアを考慮した古典処理の高速化技術開発は必須となる。

これは、従来 HPC 分野で行っていたコード最適化と変わらないが、量子分野の利用形態を考慮すると、Python 環境での計算の高速化が必須になる。

現在開発されている、ノイズがある量子コンピュータ NISQ (Noisy Intermediate-Scale Quantum Computer) は、そのノイズにより、今後量子ビット数が増えていったとしても、アプリケーション上の制限から、精度の面で適用できる事例はきわめて制限されると予想される。その場合、量子アルゴリズムの性能検証には、量子回路シミュレータを使うしかない。また、NISQ が適用できる事例があっても、実際に利用できるかどうかは、解の精度に大きく依存する。つまり、NISQ 上で十分な精度が得られるかは、扱う問題の性質に依存することになる。このことから、入力データの特性に応じて NISQ 実機を利用するのか、量子回路シミュレータを利用するのか、の自動選択機能が必須機能になると考えている。

以上の 1~2 の開発面での課題や、上記の NISQ の利活用問題を解決するには、我々が 20 年来研究してきた自動チューニング(AT)技術[6]の活用が不可欠と考えている[4]。AT 技術の適用で、将来、

FTQC の性能が古典環境を凌駕する時代が到来しても、ユーザがプログラム変更を意識せずに量子・古典計算を継続することができる計算基盤の構築が可能である。

3 ケーススタディ

図 1 の HPC-Centric Quantum Computing 実現には、量子コンピュータ実機、GPU による量子回路シミュレータ、量子アニーラ実機、疑似量子アニーラなどを、問題特性や解の精度を考慮して、適切に資源を自動選択する機能の実現が必須となる。

ここではまず、量子アニーラを選択した場合の、性能パラメータチューニング問題に焦点を当て、HPC-Centric Quantum Computing で必須となる要素技術の 1 つという位置づけとして、ケーススタディを行う。

3.1 量子アニーラにおける性能パラメータチューニング問題

ここでは、量子および疑似量子アニーラで現れる性能パラメータチューニングの問題[7][8]を取りあげる。具体的には、組合せ最適化問題である最小頂点被覆問題における性能パラメータ問題[7]を紹介する。

量子および疑似量子アニーラでは、組合せ最適化問題を QUBO で定式化をする。最小頂点被覆問題の場合、QUBO 定式化には制約項の重み (W_a) とコスト項の重み (W_b) が含まれている。式 (1) で、これを説明する。

$$H = W_a \sum_{(u,v) \in E} (1 - x_u)(1 - x_v) + W_b \sum_{v \in V'} x_v \quad (1)$$

式(1)の性能パラメータは、得られる解の性能に大きな影響を及ぼす。報告[7]の事例を紹介する。ここでは、メッシュ形状のグラフを用意する。このとき最適解は、被覆ノードが並ばないように配置するのが最適解となる。したがって、最適解との差を厳密に評価できるベンチマークとなる。

このベンチマークを、疑似量子アニーラの 1 つである CMOS アニーリングマシン[9]で評価した。図 2 は $N=7$ における最適解回答率を、 W_a を変化させたものである。

図 2 より、最適解を 100%の確立で回答できる W_a の値は限定されており、それ以外は、最適解を回答できない。このように、量子アニーラにおい

ても、性能パラメタの調整は必須であるばかりか、解の性能に大きく影響を及ぼす。そのため、求解あたりの消費電力などを考慮した「エコな」HPC-Centric Quantum Computing の実現には性能パラメタチューニングが必須である。

3.2 量子回路シミュレータの高速化

次の事例は、GPU による量子回路シミュレーションである。ここでは、波動関数の時間発展量子シミュレーションのベンチマーク（量子シミュレーションベンチマーク）を用いて性能評価をした結果[7]を紹介する。

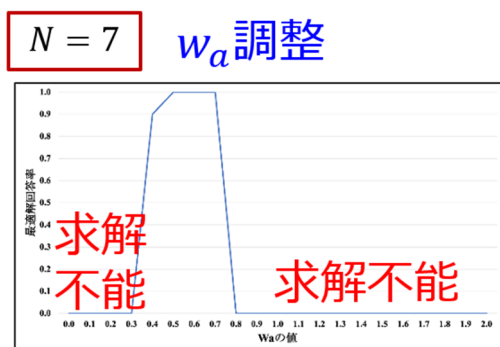


図2 CMOS アニーリングマシンでの

最適求解回答率[7] ($N=7$, W_a の値を調整)

量子回路シミュレータは、スーパーコンピュータ「不老」TypeII サブシステムの GPU で動作する、NVIDIA cuQuantum を利用している。

NVIDIA cuQuantum では、量子ゲートをくっつけるか、CPU 実行時のスレッド数、GPU 利用の有無、GPU の実装方式選択 (CUDA、cStateVec)、CUDA のブロックごとのスレッド数、GPU のデータブロック数などの、性能パラメタが存在する。ここでは、GPU の利用の有無、量子ゲートのくっつける数の調整を対象とした。

図3に、量子ビット数=8, 10, 16, 18 の波動関数の時間発展の量子シミュレーションにおいて、性能パラメタ `max_fused_gate_size` を 2, 3, 4, `gpu_mode` を 0, 1 の範囲で変化させた時の結果を示す。

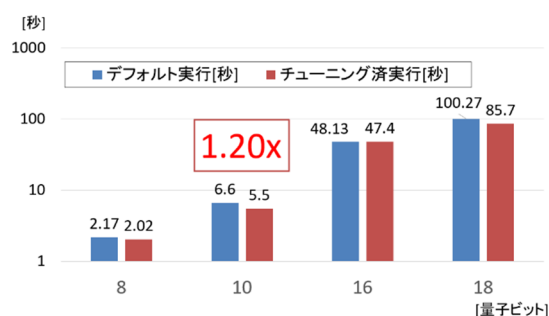


図3 量子回路シミュレータ cuQuantum の

性能パラメタチューニング結果[7]

図3から、一部の性能パラメタをチューニングでも、最大で1.2倍の速度向上が達成できることが分かる。チューニング対象の性能パラメタの範囲を広げ、より多き量子ビットでの実行時には、さらに性能チューニングの効果が増すと期待できる。したがって、量子回路シミュレータの性能パラメタチューニングも重要な技術要因になりうる。

3.3 量子関連計算を指向したパラメタ調査用ジョブ投入インタフェース開発

(1)概要

ここでは、量子アルゴリズム研究者のプログラム実行に関して、スーパーコンピュータ上での容易な実行を支援するインタフェース開発事例について解説する。

現在のスーパーコンピュータ環境は、Python や GPU での実行環境は、研究室のサーバ環境とほぼ同じ実行環境であるといえる。また、コンテナアプリケーションも整備もされており、ほぼローカル環境と相違なくなっている。しかし、ジョブの投入が、スーパーコンピュータ環境になじみのないユーザにとっては、利用の障壁と感じる要因の1つであると考えている。そこで本節では、ジョブ投入のための GUI (Graphical User Interface) の開発事例を紹介する。

(2)CIM と CACm

最適化問題の新しいソルバの1つとしてコヒーレントイジングマシン (Coherent Ising Machine, CIM) [10] が注目されている。一方、CAC (Chaotic Amplitude Control) を活用した CIM である CIM-CAC は、より難しい最適化問題で従来の量子アニーリング方式より効率が良いことが示されており [10]、研究の進展が期待される。

ここでは、CACm (Chaotic Amplitude Control with

momentum)[11]を対象にする。CACm ベンチマーク[12]が、フリーソフトとして公開されている。その性能を高めるため、性能パラメタ調整が必要になる。具体的には、CACm ベンチマークは、以下の表 1 の性能パラメタがある。

また、CACm ベンチマークは pyTorch で実装されており、スーパーコンピュータ上でも実行可能である。そこで、表 1 の性能パラメタチューニングを考える。

(3)GUI 利用例

CACm ベンチマーク[12]をスーパーコンピュータ「不老」で簡便に実行できる GUI 開発について説明する。しかし対象となる処理は、特定の実行可能ファイルを、特定の引数を指定して実行できるバッチジョブスクリプトの生成と、バッチジョブの投入であるため、汎用的な処理である。以下に、開発した GUI の利用例を説明する。

表 1 CACm の性能パラメタ[12]

| パラメタ | 説明 |
|-----------|-------------------------------|
| T | タイムステップ数 |
| $\beta 1$ | 初期 decay rate |
| $\beta 2$ | 最終 decay rate |
| α | Coupling strength |
| γ | Momentum term strength |
| ξ | auxiliary variables の 変化割合 |
| Δ | タイムステップサイズ |

- まず、スーパーコンピュータ「不老」が提供する様々なプログラムを利用するために、`~/.bashrc` に「/center/local/bin」のパスを通す必要がある。
- 設定後、`charge` と入力する。ログイン ID の予算情報とファイル使用量が表示される。
- 本 GUI を起動するには、X サーバ機能を搭載したターミナルソフトウェアを使って、スパコンのログインノードに公開鍵を使って SSH 接続して利用する。X サーバ機能を搭載したターミナルソフトを準備する必要がある。

ジョブスクリプト作成アプリは、汎用可視化ソフトの AVS/Express のテキストタイプのインタプ

リタ言語 (V 言語) で記述されている。GUI 表示が可能なログインノード上で、`vis` コマンドに V 言語で記述されたアプリのファイル名を指定して利用する。

例) `vis jobscr.v`

図 4 に、開発したジョブスクリプト生成 GUI を示す。

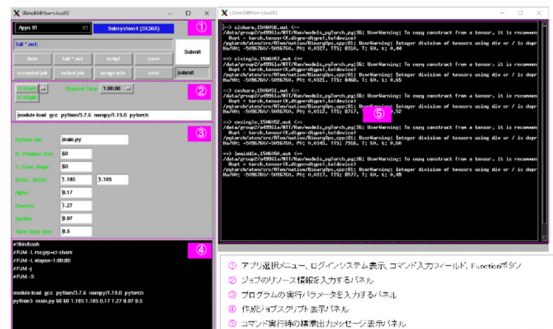


図 4 ジョブスクリプト生成 GUI

図 4 の各パネルの機能は以下である。

- ①アプリ選択メニュー、ログインシステム表示、コマンド入力フィールド、Function ボタン
- ②ジョブのリソース情報を入力するパネル
- ③プログラムの実行パラメタを入力するパネル
- ④作成ジョブスクリプト表示パネル
- ⑤コマンド実行時の標準出力メッセージ表示パネル

CACm ベンチマークのパラメタをチューニング (パラメタ調査委) するための専用 GUI を、以下の図 5 に示す。

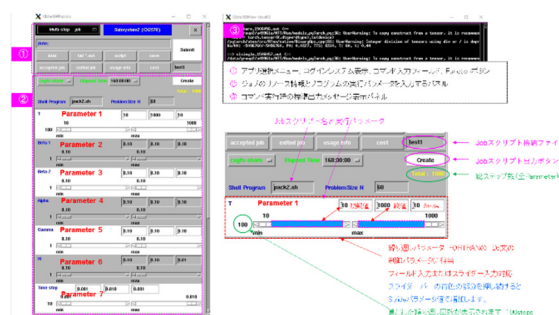


図 5 パラメタ調査用ジョブスクリプト生成 GUI

図 5 の、各パネルの説明を以下に示す。

- ①アプリ選択メニュー、ログインシステム表示、コマンド入力フィールド、Function ボタン
- ②ジョブのリソース情報とプログラムの実行パラメタを入力するパネル

③コマンド実行時の標準出力メッセージ表示パネル

図4と図5のGUIともに、スーパーコンピュータ「不老」専用のジョブ投入スクリプトが自動生成され、そのジョブ投入スクリプトを自動でジョブ投入して実行することができる。そのため、ユーザは、ジョブ投入スクリプトを気にする必要はない。

(4)考察

図5のパラメタ調査機能があるGUIについては、大量に生成される結果の可視化機能が欠かさない。この可視化機能の作成は、今後の課題である。

また、ユーザの与えたパラメタの全探索機能は一定の有効性はあるが、実行時間的に限界がある。また、利用の観点から考えると、課金量も増えて予算問題となる。そこで、膨大なパラメタ空間を効率よく探索する方法が必須となる。我々は、この機能をAT技術を用いて実現することを考えている。さらに、GUIと連結した形での実装を計画している。

この機能のGUI実装と性能評価は、重要な今後の課題である。また、GUIは量子関連技術のユーザサービスの観点でも必要であるため、サービスへの展開が期待される。

4 おわりに

本報告では、将来実現すると期待されているFTQCの実現までに、量子関連研究をスーパーコンピュータとHPC技術を用いて支援し、量子・古典ハイブリッド計算を実現する、HPC-Centric Quantum Computingの概念を提案した。また、その主要技術として、AT技術が必要となることを述べた。

将来のHPC-Centric Quantum Computingの実現に向けたケーススタディとして、疑似量子アニーラ、量子回路シミュレータ、コヒーレントイジングマシンの性能パラメタチューニング問題を紹介した。特に、ジョブ投入のための専用GUIの開発を進めており、今後、運用サービスへの展開が期待される。

今後、より多くのケーススタディを行う必要がある。また、HPC-Centric Quantum Computing実現のための実証研究(PoC)と、運用サービスへの展開

を行っていく予定である。

謝辞

本研究は、学際大規模情報基盤共同利用・共同研究拠点(JHPCN)、および、革新的ハイパフォーマンス・コンピューティング・インフラ(HPCI)の支援による(課題番号: jh240002)。

また本研究は、NTT Researchの資金援助と技術支援を受けております。著者一同、感謝いたします。

参考文献

- [1] Endre Boros et al., Local Search Heuristics for Quadratic Unconstrained Binary Optimization (QUBO), Journal of Heuristics, Vol. 13, pp. 99–132, 2007.
- [2] Yuri Alexeev, et al., Quantum-centric Supercomputing for Materials Science: A Perspective on Challenges and Future Directions, Future Generation Computer Systems, Vol. 160, pp. 666-710, 2024.
- [3] Javier Robledo-Moreno, et al, Chemistry Beyond Exact Solutions on a Quantum-Centric Supercomputer, arXiv, arXiv:2405.05068v1, pp. 1-38, 2024.
- [4] Takahiro Katagiri, HQCC-AT: An Application Programming Interface for Hybrid Quantum-Classical Computing with Auto-tuning Facility, September 2024. A preprint. DOI: 10.13140/RG.2.2.18404.39043
- [5] 理化学研究所、量子コンピュータとスパコンを連携利用するためのプラットフォーム研究開発プロジェクトを始動、2023年11月22日。 https://www.riken.jp/pr/news/2023/20231122_1/index.html
- [6] Takahiro Katagiri, Daisuke Takahashi, Japanese Autotuning Research: Autotuning Languages and FFT, Proceedings of the IEEE, Vol. 106, Issue 11, pp. 2056–2067, 2018.
- [7] 森下誠ほか、量子コンピューティングへの自動チューニングの適用と評価、研究報告ハイパフォーマンスコンピューティング (HPC)、Vol. 2023-HPC-188, No. 2, pp.1-7, 2023.
- [8] Ryoga Fukuhara et al., Performance Evaluation of CMOS Annealing with Support Vector Machine, arXiv, arXiv:2404.15752 [cs.PF], pp.1-8, 2024.
- [9] Annealing Cloud Web. <https://annealing-cloud.com/ja/index.html>
- [10] 山本喜久、最適化問題解決の未来：コヒーレントイジングマシン (CIM)、NTT 技術ジャーナル、2021. <https://journal.ntt.co.jp/article/16174>
- [11] Timothée Leleu, et al., Scaling advantage of chaotic amplitude control for high-performance combinatorial optimization, Communications

Physics, Vol. 4, No. 266, 2021.

- [12] CMCm code GitHub:
[https://github.com/NTTRI-PHI-
Algorithms/CACm](https://github.com/NTTRI-PHI-Algorithms/CACm)