

High Performance Computing in Quantum Computing Simulators

Parallel implementation of quantum simulators on classical computers

答辩人：陈科

Faculty of Artificial Intelligence
School of Information Engineering
Nanchang University

高性能并行计算期末答辩
2023 年 5 月 12 日

个人介绍-我的量子经历

最早从 2021 年 1 月（大一寒假）接触量子计算，之后也打算继续从事量子计算与量子信息领域的研究。

- (2023 年 5 月-至今) 与 Linke 实验室中科大老师交流量子网络科研任务。
- (2023 年 4 月-至今) 本源量子计算编程挑战赛，目前完成初赛 (8/400+)，正在进行决赛。
- (2022 年 12 月-至今) 量子模拟器源码阅读，筹备从零构建量子模拟器项目。
- (2022 年 6 月-至今) 参与中科大暑期量子培训班，进行量子计算方面知识培训。
- (2022 年 3 月-2022 年 5 月) 华为黑客松量子算法竞赛，优化经典混合量子神经网络，解决手写图像识别任务。
- (2021 年 1 月-2022 年 5 月) 复现 QuCloud 论文的量子比特映射。

我还参与一些量子会议讲座；阅读论文一些论文；通过 inoreader 关注量子最新消息。写了一些博客与报告。

Quantum is Pure Science and not Magic

目录

- 1 Background (背景介绍)
- 2 HPC in Quantum Computing Simulator (量子模拟器中的高性能计算)
- 3 Evaluation (实验评估)
- 4 Conclusions and future work (结论与未来工作)
- 5 References (参考文献)

背景介绍-Qubit(量子比特)

量子计算表达信息的方式不同，一个量子比特能够蕴含更多的信息。

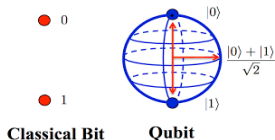


图: Bit and Qubit



图: “不死不活，即死又活”

比特与量子比特表示示例

一个经典比特表示为:

$$0|1 \quad (1)$$

一个量子比特表示为:

$$|\psi\rangle = \alpha_0|0\rangle + \alpha_1|1\rangle \quad (2)$$

背景介绍-Quantum Gate(量子门)

类似于经典的逻辑门，量子中对量子比特进行操作称为量子算符（也叫量子门）。











Gate	Symbol	Unitary	Gate	Symbol	Unitary
Pauli X		$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$	U1		$\begin{bmatrix} 1 & 0 \\ 0 & e^{i\lambda} \end{bmatrix}$
Pauli Y		$\begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}$	U2		$\frac{1}{\sqrt{2}} \begin{bmatrix} 1 & -e^{i\lambda} \\ e^{i\phi} & e^{i(\phi+\lambda)} \end{bmatrix}$
Pauli Z		$\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$	U3		$U(\theta, \phi, \lambda)$
Hadamard		$\frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$	Z Rotation		$\begin{bmatrix} e^{-\frac{i\phi}{2}} & 0 \\ 0 & e^{\frac{i\phi}{2}} \end{bmatrix}$
CNOT		$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$	SWAP		$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$

图: 一些量子门

背景介绍-Quantum Circuits(量子电路)

量子门按时间序列作用到量子比特上，即称为一个量子电路。

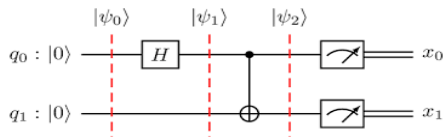


图: 一些量子门

Bell 态制备电路的运算过程

暂时不考虑最后的测量算符，测量可以理解为概率采样：

$$|\psi_2\rangle = CNOT|\psi_1\rangle = CNOT(H \otimes I|\psi_0\rangle) \quad (3)$$

通过 线性代数的矩阵运算，即可得到最终的量子态。

背景介绍-目前的经典量子模拟器

目前专用量子计算机存在问题：(1) 成本很高、资源有限；(2) 存在较大噪声，纠错操作较少；(3) 无法做到随时、有效使用。因此大多考虑在经典计算机上用量子模拟器来研究量子算法与应用。

Year	Product	Institution/Company	Number of qubits/system	Technology	Public research
2016	Quantum Experience	IBM	1-60+	Superconducting	✓
2016	Quantum Cloud Service (QCS)	rigetti	35/11	Superconducting	✓
2016	Leap (Quantum Cloud Platform)	D-WAVE	5000+	Quantum Anneal	✓
2016	Qip (Open-Source Quantum Computing Framework)	Google	53	Superconducting	✓
2016	Quantum Inspire	Qutech	Superconducting 5	Superconducting	✓
2016	Azure Quantum	Microsoft	Including 10-50 devices from IonQ and Honeywell	Ion trap	✓
2016	AWS Braket	aws	Includes Rigetti's 31-qbit device	Superconducting Ion trap	✓
2020	Tensorflow Quantum (Quantum Machine Learning Open-Source Library)	Google	53	Superconducting	✓
2020	Optical quantum cloud platform	Q-CTRL	53	Photonic	✓
2021	IonQware Quantum QC	IONQ	Includes IBM's 65-qbit device	Contains almost all technologies	✓
2017	Quantum Computing Cloud Platform	QCCloud	11	Superconducting	✓
2018	HiQ	HUAWEI	-	Simulation /Simulation	✓
2019	Borging quantum computing (simulation) cloud platform	QUBUS	-	Simulation /Simulation	✓
2020	Origin Quantum computing cloud platform	本源量子	5	Superconducting	✓
2020	Quantum leaf	中科院	Access to the hardware of the Institute of Physics, Chinese Academy of Sciences	Superconducting	✓
2020	Tianqi (quantum cloud platform)	SPINQ	5	Nuclear magnetic resonance (NMR)	✓
2020	Second-generation quantum computing cloud platform	本源量子	12	Superconducting	✓
2021	HiQ Cloud	本源量子	18	Superconducting	✓
2021	First-generation superconducting quantum computing cloud platform	本源量子	8	Superconducting	✓

本源量子
ORIGIN QUANTUM



Origin Quantum

本源量子具备经典操作系统的基础功能，更带来了高效利用量子计算机资源的解决方案。支持多量子任务的并行计算与调度和对量子计算机持续不间断的校准优化。

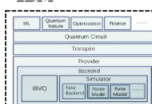
HUAWEI



MindQuantum

华为HiQ量子计算云平台提供多种在线开发环境，包括基于开源软件交互式开发环境Jupyter Notebook、基于云原生开发环境CloudIDE和量子线路图形编程环境HiQ Composer。

IBM



IBM qiskit

Qiskit是一款开源软件，用于在电路、脉冲和算法层面上处理量子计算机。此外，在这个核心模块之上还存在一些特定于领域的应用程序API。

图：现有量子计算平台

背景介绍-经典计算机上的量子模拟器实现方法

常见的量子模拟器模拟方法 [1]:

- 状态向量模拟;
- 密度矩阵模拟;
- 稳定器 (Clifford stabilizer) ; 扩展稳定器 (Clifford + T);
- 矩阵乘积态 (matrix_product_state) ;
- 张量网络模拟;[2] (ASC22 决赛赛题)
- 脉冲模拟 [3];
- ZX-calculus[4]

常见的量子模拟器硬件资源:

- CPU, 优秀实现有 IBM[1]、牛津大学量子研究中心开发的 QUEST[5] (ASC20-21 赛题) 等;
- GPU, 优秀实现有 NVIDIA[6]、清华翟季东老师 (清华超算队指导老师) 组;
- FPGA (22 年 CCF, 讨论 FPGA 在量子模拟器上的加速);
- 其他超算集群, 比如神威量子模拟器 (2021 年戈登贝尔奖)、Summit 等。

背景介绍-StateVector(状态向量) 的量子模拟器

StateVector 将量子态存储为向量的形式。

$$|\psi\rangle = \alpha_0|0\rangle + \alpha_1|1\rangle = \begin{bmatrix} \alpha_0 \\ \alpha_1 \end{bmatrix} \quad (4)$$

在计算机上模拟量子电路的运行，我们只需要进行矩阵的乘积即可：

基于状态向量方法的 Bell 态制备电路运算过程

假设初始态为 $|00\rangle$ ：

$$|\psi_2\rangle = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \left(\frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \otimes \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \right) = \begin{bmatrix} \frac{1}{\sqrt{2}} \\ 0 \\ 0 \\ \frac{1}{\sqrt{2}} \end{bmatrix} \quad (5)$$

量子模拟器中的高性能计算-Qiskit

下面我们目前非常优秀的量子模拟器 qiskit-aer[1] 来研究量子模拟器中的高性能并行计算部分。

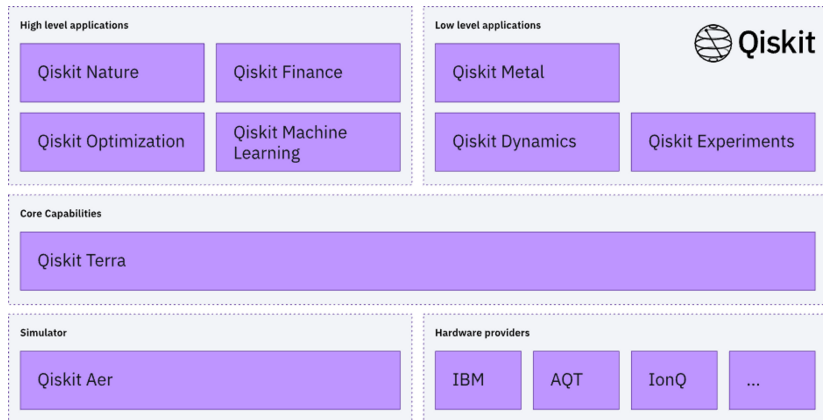
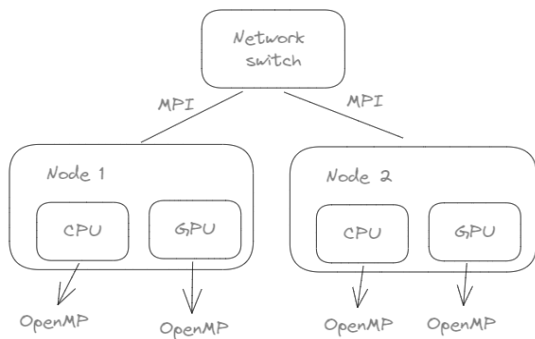


图: IBM Qiskit 量子系统与量子平台

量子模拟器中的高性能计算-基本的集群架构

高性能计算中，通过MPI实现不同节点之间的通信并行，通过OpenMP实现CPU或GPU的线程级并行。



图：简单的集群架构示意图

注意，上述没有涉及内存，CPU与GPU之间的通信，GPU之间的通信。其中GPU之间的通信可能涉及，需要使用NVLink或PCIe。

量子模拟器中的高性能计算-qiskit-aer 中的并行计算

CPU 架构下的并行计算：

- 通过 OpenMP，针对单一量子电路下，单个多核 CPU 实现**矩阵乘积**以及**数据加载**的线程级并行。
- 通过 OpenMP，针对多个不同量子电路，单个多核 CPU 实现**多个量子电路任务**的线程级并行。
- 通过 OpenMP，针对多个相同量子电路 (又称 shots)，单个多核 CPU 实现**不同的 shot**的线程级并行。
- 通过 MPI，针对单一量子电路，多个 CPU 之间实现量子电路的**分块多节点并行**处理。

GPU 架构下的并行计算：

- 通过 MPI 或**GPU 通信协议**，针对单一量子电路，多个 GPU 之间实现量子电路的**分块多节点并行**处理。
- 通过 OpenMP，利用**批处理**优化多个 shot 并行。

量子模拟器中的高性能计算-OpenMP 实现矩阵相乘的线程级并行

假设原来需要的时间为 T ，通过 OpenMP 并行执行 for 循环，理论上只需要 $T/omp_threads_$ 加线程开销的时间 T' 。

OpenMP 实现并行矩阵相乘求解概率

```
#pragma omp parallel for if (num_qubits_ > omp_threshold_
    && omp_threads_ > 1) num_threads(omp_threads_)
//判断量子比特数是否大于阈值，大于则并行。
for (int_t j=0; j < END; j++) {
    //probability函数内部是矩阵乘积，实现矩阵相乘的线程级并
    行。
    probs[j] = probability(j);
}
```

当量子比特数较少时，线程开销的时间相对较高，将存在 $T < T'$ 。当量子比特数较多时，矩阵计算的开销更大，即 $T > T'$ 。(默认 14 个 qubits)

量子模拟器中的高性能计算-OpenMP 实现 shot 并行

同一个量子电路，重复执行多次实验，每个实验都将测量得到一个结果(可能不同)，这个过程称为一个 shot。最终需要统计每次 shot 的结果，即为我们需要的最终结果。

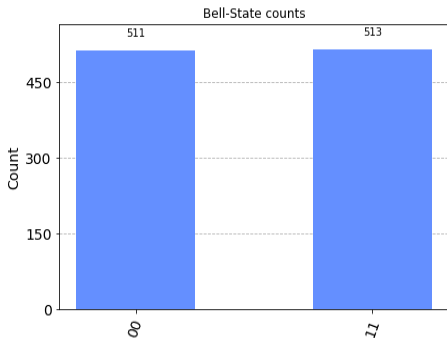


图: 1024 次 shot 的 Bell 电路统计结果

不同 shot 之间不存在数据依赖，可直接用 OpenMP 实现线程级并行。

量子模拟器中的高性能计算-MPI 量子电路分块模拟

MPI 主要在不同的节点之间，主要的区别是不同节点之前不共享内存，需要进行信息的传递。

States are divided into chunk and data exchange is done per chunk to save memory space

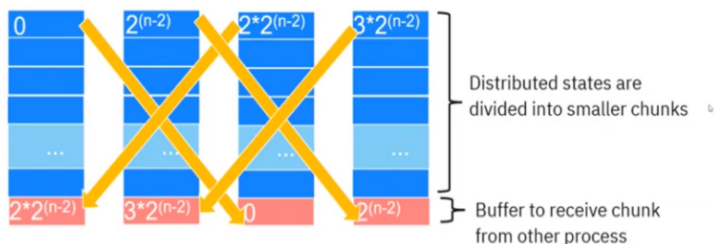


图: MPI 量子电路并行模拟 [7]

上述例子中，将状态向量分成 4 块，分别在 4 个节点上存储与计算，只有涉及块之间存在双量子门作用，才需进行数据交换。

实验评估-CPU 并行

后面我还尝试一些实验，在服务器上实现了一些测试代码，通过测试结果来评估并行算法的效果。

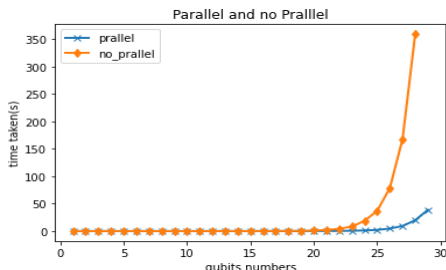


图: 并行与非并行的性能差别 (shots is 1024, Circuit is QuantumVolume)

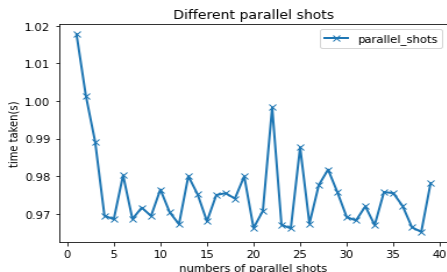


图: 不同并行 shot 量的性能差别 (shots is 1000, Circuit is QuantumVolume)

Intel®Xeon® CPU ES-2698 v4 @ 2.20GHz, where each CPU has 40 cores

实验评估-GPU 并行

针对 GPU，在服务器上实现了一些测试代码，通过测试结果来评估并行算法的效果。

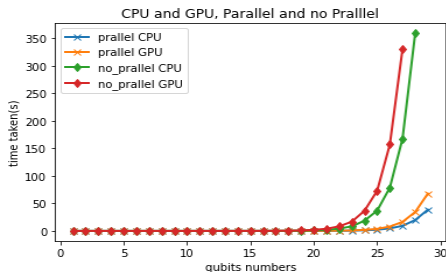


图: CPU 与 GPU 的并行性能差别 (shots is 1024, Circuit is QuantumVolume)

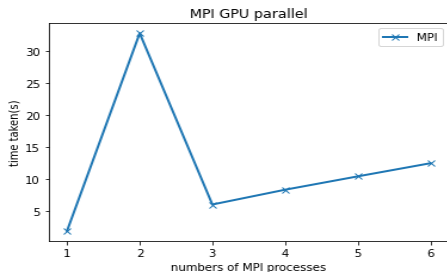


图: 不同 MPI 进程下, GPU 运行的性能差别 (shots is 1024, Circuit is QuantumVolume, qubits is 24)

4× NVIDIA® Tesla® V100

Conclusions and future work (结论与未来工作)

结论:

- 在量子比特数量较大的情况下, OpenMP 并行矩阵相乘以及其他操作能够很有效的提高性能。
- MPI 并行 GPU 运算, 总共具有 4 块, 在进程为 3 时, 达到多块 GPU 的最优解, 可能是合理的。
- GPU 的计算速度要比 CPU 慢一点, 不太合理。
- 改变 shot 的并行量, 性能差异不太, 不太合理。

未来工作:

- 在 github 提交 issue, 在 slack 讨论存在的问题。
- 上述使用多节点的集群, 后续考虑搭建一个多节点的服务器集群。

个人认为, 量子模拟器目前很难做出很大创新, 但是学习和研究量子模拟器, 能够很好的理解量子计算的工作原理、提高自己的高性能计算水平。对于学习量子计算和量子信息有很大的帮助。

参考文献 I

- [1] Qiskit contributors. *Qiskit Aer*. 2023. DOI: [10.5281/zenodo.2573505](https://doi.org/10.5281/zenodo.2573505).
- [2] Román Orús. “A practical introduction to tensor networks: Matrix product states and projected entangled pair states”. In: *Annals of Physics* 349 (Oct. 2014), pp. 117–158. DOI: [10.1016/j.aop.2014.06.013](https://doi.org/10.1016/j.aop.2014.06.013). URL: <https://doi.org/10.1016%5C%2Fj.aop.2014.06.013>.
- [3] Thomas Alexander et al. “Qiskit Pulse: Programming Quantum Computers through the Cloud with Pulses”. In: *Quantum Science and Technology* 5.4 (Aug. 2020), p. 044006. ISSN: 2058-9565. DOI: [10.1088/2058-9565/aba404](https://doi.org/10.1088/2058-9565/aba404). (Visited on 04/23/2023).
- [4] The ZX-calculus contributors. *The ZX-calculus*. <https://zxcalculus.com/>. 2023.

- [5] QuEST contributors. *QuEST: a high performance simulator of quantum circuits, state-vectors and density matrices*.
<https://github.com/QuEST-Kit/QuEST>. 2023.
- [6] NVIDIA CUDA Quantum contributors. *NVIDIA CUDA Quantum: The platform for hybrid quantum-classical computing*.
<https://developer.nvidia.com/cuda-quantum>. 2023.
- [7] Jun Doi. “Parallel GPU Quantum Circuit Simulations on Qiskit Aer”.
In: ().