计算机科学
COMPUTER SCIENCE

# Grover量子搜索算法在“嵩山”超级计算机系统中的模拟

杜帅岐, 刘晓楠, 廉德萌, 刘正煜

引用本文

杜帅岐, 刘晓楠, 廉德萌, 刘正煜. Grover量子搜索算法在“嵩山”超级计算机系统中的模拟[J]. 计算机科学, 2024, 51(9): 96-102.

DU Shuaiqi, LIU Xiaonan, LIAN Demeng, LIU Zhengyu. Simulation of Grover's Quantum Search Algorithm in “Songshan” Supercomputer System [J]. Computer Science, 2024, 51(9): 96-102.

---

**相似文章推荐（请使用火狐或 IE 浏览器查看文章）**

**Similar articles recommended (Please use Firefox or IE to view the article)**

基于矩阵乘积态的有限纠缠量子傅里叶变换模拟

Simulation of Limited Entangled Quantum Fourier Transform Based on Matrix Product State

计算机科学, 2024, 51(9): 80-86. https://doi.org/10.11896/jsjkx.230300215

基于MPI+CUDA的DSMC/PIC耦合模拟异构并行及性能优化研究

Heterogeneous Parallel Computing and Performance Optimization for DSMC/PIC Coupled Simulation Based on MPI+CUDA

计算机科学, 2024, 51(9): 31-39. https://doi.org/10.11896/jsjkx.230300188

基于国产DCU异构平台的图匹配算法移植与优化

Transplantation and Optimization of Graph Matching Algorithm Based on Domestic DCUHeterogeneous Platform

计算机科学, 2024, 51(4): 67-77. https://doi.org/10.11896/jsjkx.230800193

DSMC/PIC耦合模拟的大规模高效混合并行计算研究

Large-scale Efficient Hybrid Parallel Computing for DSMC/PIC Coupled Simulation

计算机科学, 2023, 50(11A): 230300146-9. https://doi.org/10.11896/jsjkx.230300146

基于Grover算法的图着色问题求解

Solving Graph Coloring Problem Based on Grover Algorithm

计算机科学, 2023, 50(6): 351-357. https://doi.org/10.11896/jsjkx.220400051

计算机科学 COMPUTER SCIENCE
http://www.jsjkx.com
DOI:10.11896/jsjkx.230600219

# Grover 量子搜索算法在“嵩山”超级计算机系统中的模拟

杜帅岐[1,2]　刘晓楠[1]　廉德萌[1,2]　刘正煜[1,2]
1 国家超级计算郑州中心　郑州 450001
2 郑州大学计算机与人工智能学院　郑州 450001
(du_shuaiqi@163.com)

**摘　要**　量子计算凭借其叠加性和纠缠性,具有强大的并行计算能力。然而,目前的量子计算机不能在保证大规模量子比特处于稳定叠加态的同时,进行干涉、纠缠等量子操作。因此,当前研究和推动量子计算的有效途径是使用经典计算机模拟量子计算。Grover 量子搜索算法针对无序数据库搜索问题设计,将搜索的时间复杂度加速至开平方级,能加速机器学习中的主成分分析。因此,研究和模拟 Grover 算法,可以促进量子计算与机器学习结合领域的发展,为 Grover 量子搜索算法的应用以及量子机器学习在“嵩山”超级计算机系统中的模拟奠定基础。通过研究 Grover 量子搜索算法,模拟出了算法的量子线路。使用 Toffoli 量子门优化该量子线路,在减少了两个辅助量子比特的同时,提出了 Grover 算法的通用量子线路。实验基于“嵩山”超级计算机系统的 CPU+DCU 异构体系,使用了 MPI 多进程+HIP 多线程的两级并行策略。通过调整辅助比特在量子线路中的位置,减少了 MPI 进程间的通信;使用分片的方式传输数据依赖的量子态。对比串行版本,并行化的模拟算法取得了最高 560.33 倍的加速,首次实现了 31qubits 规模的 Grover 量子搜索算法。

**关键词**:Grover 量子搜索算法;异构体系;MPI;HIP;分片传输
**中图分类号**　TP385

# Simulation of Grover's Quantum Search Algorithm in "Songshan" Supercomputer System

DU Shuaiqi[1,2], LIU Xiaonan[1], LIAN Demeng[1,2] and LIU Zhengyu[1,2]
1 National Supercomputing Centerin Zhengzhou, Zhengzhou 450001, China
2 School of Computer and Artificial Intelligence, Zhengzhou University, Zhengzhou 450001, China

**Abstract**　With its superposition and entanglement properties, quantum computing has a powerful parallel computing capacity. However, current quantum computers cannot ensure stable superposition states of large-scale qubits while performing quantum operations such as interference and entanglement. Therefore, the current approach to promote quantum computing is to simulate quantum computing using classical computers. The Grover quantum search algorithm is designed for the problem of searching unsorted databases, reducing the time complexity to square root level, and accelerating principal component analysis in machine learning. Therefore, studying and simulating the Grover algorithm can promote the development of quantum computing combined with machine learning and lay the foundation for its application as well as the simulation of Grover quantum search algorithm in the "Songshan" supercomputer system. By studying the Grover quantum search algorithm, the quantum circuit of the algorithm is simulated. The Toffoli quantum gate is used to optimize the quantum circuit, proposing a universal quantum circuit for the Grover algorithm while reducing two auxiliary qubits. The experiment is based on the CPU+DCU heterogeneous system of the "Songshan" supercomputer system, using a two-level parallel strategy of MPI multiprocessing and HIP multithreading. By adjusting the position of the auxiliary qubits in the quantum circuit, the communication between MPI processes is reduced, and the data-dependent quantum states are transmitted using a fragmentation method. Compared to the serial version, the parallelized simulation algorithm achieves a maximum speedup of 560.33 times, realizing for the first time the Grover quantum search algorithm with a scale of 31 qubits.

**Keywords**　Grover's quantum search algorithm, Heterogeneous architecture, MPI, HIP, Fragment-based transmission

## 1　引言

与经典计算方式不同,量子计算是一种使用量子位代替经典位进行信息存储和处理的计算方法。经典计算机处理的经典位只存在两种状态即 0 或 1;而量子位具有叠加特性,可以处于 1 和 0 的叠加态。$n$ 个量子位可以表示 $2^n$ 个状态,且各自以一定的概率同时参与运算,这便是量子计算的并行优势。

到稿日期:2023-06-29　返修日期:2023-11-15
基金项目:国家自然科学基金(61972413,61701539)
This work was supported by the National Natural Science Foundation of China(61972413,61701539).
通信作者:刘晓楠(prof.liu.xn@foxmail.com)

Grover[1] 于 1996 年提出了量子搜索算法。该算法是一种概率搜索算法，能够成功地搜索到具有高概率的目标[2]。其基本思想是通过量子并行计算和量子干涉来加速搜索过程，从而在较短的时间内找到目标项，起到了二次加速的作用；同时在搜索目标时不需要考虑搜索问题的内部结构的性质，将注意力放在元素的索引上，因此具有很强的通用性[3]。对 Grover 算法的研究可以促进该算法量子线路深度和宽度的优化[4]，以及量子算法在机器学习[5]、图着色问题[6]及海量数据分析等方面的应用，同时也可以促进量子计算在药物发现、化学分析、电力能源等学科中的应用[7]。

然而，量子计算机的实现和运行需要极其复杂的技术和设备，目前只有少数实验室和企业能够拥有量子计算机，且量子算法的运行规模很小。因此，使用经典计算机模拟量子计算可以作为一种替代方案，但对计算机有着大规模内存和密集浮点型运算的要求，往往基于超级计算机平台实现[8]。目前已有学者分别在“嵩山”超级计算机系统和神威·太湖之光上完成了 44 和 46 比特的量子傅里叶变换算法[9-10]。

“嵩山”超级计算机系统是我国自主研发的新一代超级计算机，采用了海光一号 CPU＋海光一号 DCU 异构体系。该体系的优势在于，由 CPU 负责初始化数据和调度并行计算任务，擅长密集计算的 DCU 执行并行计算任务。

本文在串行 Grover 量子搜索算法的基础上，将其并行化，并提出改进后的任意比特位 Grover 量子搜索算法的通用量子线路；基于“嵩山”超级计算机系统，首次进行大规模 Grover 量子搜索算法的模拟，利用平台 CPU＋DCU 异构体系的特点，分摊存储量子态所需要的指数级内存开销，完成了 31 qubits（包含 15 个辅助量子比特）的 Grover 量子搜索算法的模拟。实验结果验证了 Grover 算法的通用量子线路，且与改进前相比，运行效率有显著的提高。

本文第 2 章分别从概率幅和向量的角度介绍了 Grover 量子搜索算法的理论基础；第 3 章对算法量子线路进行分析设计和优化，提出了通用量子线路；第 4 章基于“嵩山”超级计算机系统，对 Grover 算法的异构模拟以及优化进行详细介绍；第 5 章进行实验验证；最后总结全文并展望未来。

## 2　Grover 量子搜索算法

Grover 量子搜索算法主要是通过变换量子基态的概率幅，令所查询目标项对应的量子态的概率幅达到最大[11]。算法的搜索思想是将 $N$ 个元素进行索引编码，将索引与量子态对应，$n$ 个量子比特构造 $N=2^n$ 个搜索空间，通过算法的 Oracle 操作将要搜索的答案标记，平均反演算子操作放大标记态的概率幅，从而提升搜索到正确答案的概率。

图 1 给出了一个完整的 Grover 算法量子线路框架，涵盖初始化至等权叠加态、中间的 Oracle（Uw）、平均反演算子（Us）和最终的测量模块。Oracle 和平均反演算子组成一个完整的 G 迭代，可通过重复 G 迭代来改变所有量子态的概率[12]。

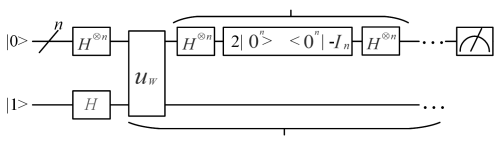



图 1　Grover 算法框架线路图

Fig. 1　Framework circuit diagram of Grover algorithm

G 迭代是 Grover 算法的核心操作，实现步骤如下。

（1）首先对于 $N$ 个搜索空间，并不知道目标态在哪，因此可以从任何地方开始，使用 Hadamard 量子门对基态变换，使每一个量子基态转换为均匀叠加态[13]，如式（1）所示。此时，所有量子态拥有相同的概率幅，被搜索到的概率也相同。如图 2 所示，虚线表示概率幅的平均值，所有量子态的条目高度相同，深色表示要搜索的量子态。

$$|\varphi\rangle=\frac{1}{\sqrt{N}}\sum_{i=0}^{N-1}|i\rangle \tag{1}$$

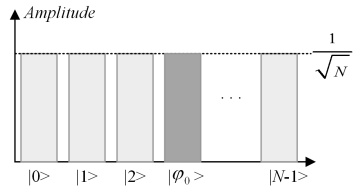



图 2　Grover 算法的量子初态

Fig. 2　Initial state of Grover algorithm

（2）构造 Oracle 算子 $U\omega$，对初始化后的量子叠加态进行判断，若当前量子态是目标态，则返回 $f(\varphi)=1$，否则返回 $f(\varphi)=0$，并通过式（2）对符合条件的量子态进行相位翻转（取反）。如图 3 所示，所有量子态的概率幅在数值上没有变化，而目标态的概率幅变为了负值。

$$|\varphi\rangle\rightarrow(-1)^{f(\varphi)}|\varphi\rangle,f(\varphi)=\begin{cases}1, & \varphi=\varphi_0\\ 0, & \varphi\neq\varphi_0\end{cases} \tag{2}$$

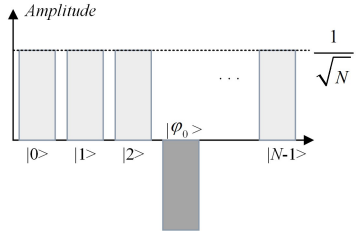



图 3　Oracle 操作后的相位翻转

Fig. 3　Flipping of phase after Oracle operation

（3）最后一步是构造平均反演算子 $Us$，经过 Oracle 操作后，目标态已被翻转，此时平均概率幅与初态平均概率幅相比有所减小，而 $Us$ 操作的目的是使目标态概率幅相较于平均概率幅放大，非目标的概率幅被缩小[14]。如图 4 所示，将目标态概率幅相对于平均概率幅 $C_\varphi$ 做相移，从而达到增大搜索到目标项概率的目的[15]。

$$U_S=2|\varphi\rangle\langle\varphi|-I \tag{3}$$

$$C_\varphi=\frac{1}{N}\sum_{\varphi}C_\varphi \tag{4}$$

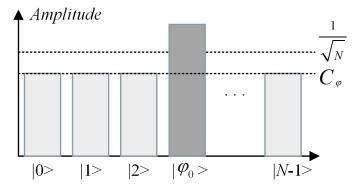



图 4　平均反演算子操作后的概率幅相移

Fig. 4　Amplitude of phase shift after average inversion operator operation

如图 5 所示，从平面向量的角度分析，假设 $|\alpha\rangle$ 是目标态，$|\beta\rangle$ 是非目标态，每一次 G 迭代目标态都会向 $|\alpha\rangle$ 旋转，在经过

约$\sqrt{N}$次$G$迭代后，目标态达到靠近$|\alpha>$的极限，此时目标态被勘测到的概率最大。

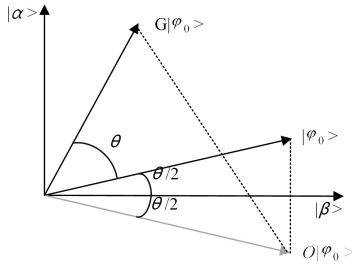



图 5 目标态相移的平面向量图

Fig. 5 Planar vector diagram of phase shift of target state

## 3 Grover 算法的通用量子线路

本文通过对开源量子仿真库 libquantum 中提供的 Grover 算法串行程序进行分析，基于 QuantuC 提出的 Toffoli 量子门实现 Grover 算法 Oracle 的思想，在 IBM Q 云平台实现并验证了 Grover 算法的通用量子线路。IBM Q 是一个云应用程序，用于对真实的量子硬件和高性能模拟器进行编程，可通过平台提供的 Circuit composer 方式对经典量子算法进行可视化创建。Circuit composer 是一种图形量子编程工具，用户可以通过拖放指令来构建量子电路，选择在真实量子计算机后端或量子模拟器中运行构建好的量子线路。

### 3.1 Grover 量子线路分析

Grover 算法的量子线路宽度和深度会随着量子比特位数的增加而增加，因此量子态的干涉、纠缠性更强；且 Oracle 操作的量子线路不固定，每增减一位量子比特或改变搜索条件，线路都会发生变化。目前已有学者在该算法及其应用方面给出了可参考的量子线路[16-19]，但存在规模小、辅助量子比特过多，以及使用了现有量子模拟平台不支持的量子门[18]等问题。

在对开源量子仿真库 libquantum 中串行版本的 Grover 算法并行移植的过程中，分析出算法的 Oracle($Uw$)和平均反演算子($Us$)都是由 Toffoli 量子门和 C-Not 量子门进行概率幅交换实现的。如图 6 所示，15 qubits 寻找 0101010 的 Grover 算法量子线路分为 7 个有效量子比特和 8 个辅助量子比特。

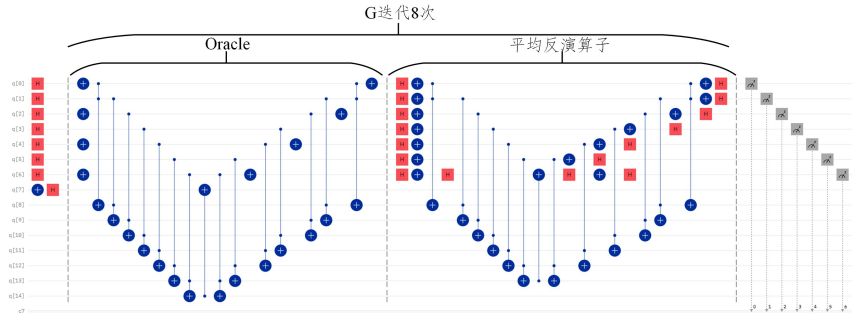



图 6 15 qubits 寻找 0101010 的量子线路

Fig. 6 Quantum circuit of 15 qubits search for 0101010

首先对全 0 基态的有效量子比特所处的量子位进行 Hadamard 变换，变换后有效量子位表示的量子态均为叠加态，辅助量子比特所处的量子位没有做任何操作，仍是全 0 的状态，而概率幅交换是由量子门的控制比特是否为 1 决定，因此 Oracle 操作的目的是根据搜索条件的不同，把辅助量子比特所处的高位表示的部分量子态与有效量子比特所处的低位表示的部分量子态进行概率幅交换。平均反演算子操作，是对 Oracle 操作后的量子态，不区分搜索条件，再次对部分量子态进行概率幅交换，一次迭代之后目标态的概率幅被放大。

### 3.2 优化后的通用量子线路

拉合尔管理科学大学的 Muhammad Faryad 教授于 2021 年提出了仅使用 Toffoli 量子门实现 Grover 算法 Oracle 操作的想法。实验基于该想法，对 Oracle 操作和平均反演算子操作均进行优化，量子态之间的概率幅交换均交给 Toffoli 量子门控制。优化后的 13 qubits 寻找 0101010 的 Grover 算法量子线路如图 7 所示。

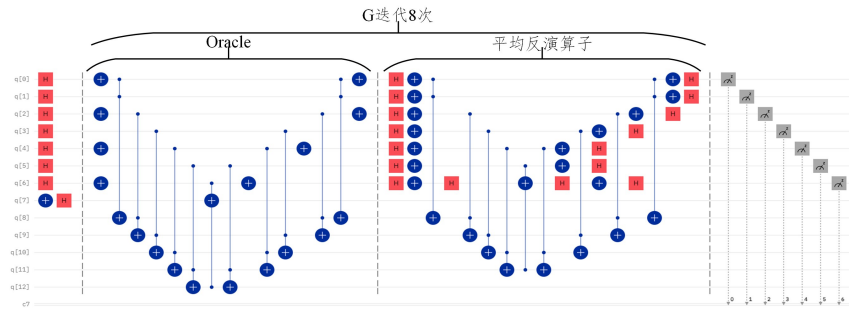



图 7 13 qubits 寻找 0101010 的量子线路

Fig. 7 Quantum circuit of 13 qubits search for 0101010

优化后的量子线路在寻找同一个序列时，同时降低了线路的深度和宽度。在线路宽度上，减少了两个辅助量子比特，宽度降低 2 个单位；在线路深度上，由于取消了 C-Not 量子门的使用，Oracle 操作和平均反演算子操作的深度都有所降低，$n$ 个 qubits 的量子线路共降低 $8\times(n/2+1)$ 个单位的深度。同时，对优化后的线路延伸到对任意位量子比特，搜索任意

序列都可构造相应的量子线路图，因此提出了 Grover 算法的通用量子线路。假设量子比特位数为 $n$，下面给出通用量子线路的伪代码，如算法 1 所示。

**算法 1** Grover 算法通用量子线路

```
输入：(target,n)
输出：probability of target
1. //Oracle 操作
2. // 对 target 量子位施加 sigma_x 门
3. for i←0 to n/2 do
4.    if target &(1≪ i)==0 do
5.       Sigma_X(Amplitudes,i)
6. //利用 toffoli 门实现相位翻转
7. Toffoli(Amplitudes,0,1,n/2+1)
8. for i←2 to n/2−1 do
9.    Toffoli(Amplitudes,i,n/2+i−1,n/2+i)
10. Toffoli(Amplitudes,i,n/2+i−1,n/2);
11. //翻转后再次施加量子门
12. for i←n/2−2 to 2 do
13. Toffoli(Amplitudes,i,n/2+i−1,n/2+i)
14. Toffoli(Amplitudes,0,1,n/2+1)
15. for i←0 to n/2 do
16.    if target &(1≪ i)==0 do
17.       Sigma_X(Amplitudes,i)
18. //平均反演算子操作
19. //相移阶段开始前的量子态叠加
20. for i←0 to n/2 do
21.    hadamard(Amplitudes,j)
22. for i←0 to n/2 do
23.    Sigma_X(Amplitudes,i)
24. hadamard(Amplitudes,n/2−1)
25. //开始相移
26. Toffoli(Amplitudes,0,1,n/2+1)
27. for i←2 to n/2−2 do
28. Toffoli(Amplitudes,i,n/2+i−1,n/2+i)
29. Toffoli(Amplitudes,i,n/2+i−1,n/2−1)
30. for i←n/2−3 to 2 do
31. Toffoli(Amplitudes,i,n/2+i−1,n/2+i)
32. Toffoli(Amplitudes,0,1,n/2+1)
33. //再次施加量子门
34. hadamard(Amplitudes,n/2−1)
35. for i←0 to n/2 do
36.    Sigma_X(Amplitudes,i)
37. for i←0 to n/2 do
38.    hadamard(Amplitudes,j)
```

对输入的基态使用 Hadamard 量子门变换为叠加态后，进入 Oracle 和平均反演算子操作，对相应的量子位施加量子门，达到量子态概率幅交换的效果。

## 4 模拟与优化

本文的目标是通过分析 Grover 算法的量子线路，实现对 Grover 量子搜索算法的大规模模拟。将量子态的初始化与线路的逻辑控制以及进程间的数据交换放在 CPU 端执行，将并行粒度高的概率幅密集计算与更新操作放在 DCU 加速器件中执行。使用高级编程语言（C 语言）模拟量子线路，HIP-C 编写核函数模拟量子门。

### 4.1 模拟实现

第 3 章中已经给出了 Grover 量子搜索算法的通用量子线路以及伪代码，模拟实现是对量子线路的仿真。

每一个量子态都有相应的复数系数（概率幅），使用浮点数分别存储概率幅的实部和虚部，实部和虚部共同组成一个结构体，用该结构体表示一个量子态[10]。在使用经典计算机模拟量子计算时，假定量子比特位数为 $n$，则经典计算机需要开辟足够的空间存储 $2^n$ 个单位的概率幅。存储压力对应的是计算压力，根据量子线路，当施加量子门时，会导致 $2^n$ 次概率幅的更新，数据量过于庞大，串行效率极差。为了体现量子计算的并行特性，在 CPU 端开启多个进程，将 $2^n$ 个单位的结构体数组均分到多个进程中，分摊指数级存储开销和计算需求，从而减轻节点内存压力，同时也可以通过进程号和数组下标定位量子态。假设在主机端开启 4 个进程存储概率幅，则每个进程存储的数据量为 $N=2^n/4$，此时每个概率幅在全局数据中的位置为 $N\times rank+index$，可以表示为：$|rank|index\rangle$，如图 8 所示。

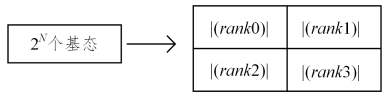



图 8 量子态存储分布图

Fig. 8 Diagram of quantum state storage distribution

量子态初始化后，CPU 调用 HIP 函数开辟 DCU 存储空间，并将划分后的数据传输至 DCU 端。在量子计算中，量子门作用在量子态上时，概率幅的计算和更新是并行过程，结合“嵩山”平台异构特性，开启多进程的同时，在设备端开启多个线程块，每个线程块的线程数为 256，线程的总规模根据主机端每个进程存储的数据量确定，线程的维度为一维，与存储单位一致。以有效量子比特位数为 13 为例，假设开启 4 个进程，则每个进程开启线程的规模为 16×256（16 个线程块），保证核函数一次性可以更新该进程所有的量子态。串行模拟时，算法一次迭代的时间开销是指数级；相比之下，并行模拟一次迭代的时间开销为常数级。但实际模拟时，由于数据放在多个进程中，每次迭代进程间需要进行数据交换，即进程通信。该过程是进程级别并行时最大的耗时模块，具体耗时对比见实验验证章节。

量子态的概率幅存储在不同的进程中，因此当计算概率幅时存在数据依赖。以 Hadamard 变换为例，数据依赖分为两种情况：（1）当目标比特落在同进程的数组区间时，各进程根据线程块的坐标与块内线程坐标定位数组下标，此时进程间不需要数据交换，直接计算即可。（2）当目标比特落在进程区间中时，该进程计算所需的信息存储在另一个进程中，两个进程间存在数据依赖，需要数据交换后再计算。两个进程通过 MPI 消息传递机制获取对方进程存储的数据，根据线程块的坐标与块内线程坐标计算出全局偏移，结合进程号定位数据。以 4 个进程为例，无数据依赖时的概率幅交换如图 9 所示，数据依赖时的概率幅交换如图 10 所示。

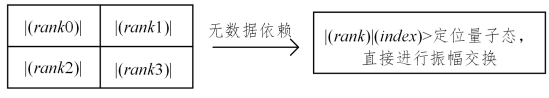



图 9 无数据依赖时的概率幅交换

Fig. 9 Probability amplitude exchange without data dependency

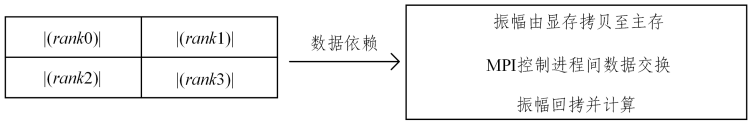



图 10 数据依赖时的概率幅交换

Fig. 10 Probability amplitude exchange with data dependency

当模拟量子门的核函数运算结束时，DCU 端存储的概率幅更新，按照线路逻辑继续施加量子门，直至线路模拟结束，概率幅由 DCU 端传回 CPU 端，完成算法的模拟。

### 4.2 模拟优化

虽然多个进程分担了存储压力，但每个进程处理的数据仍然是指数级别的。为了提高算法的模拟效率，本文针对 CPU 与 DCU 端以及量子线路分别做出了优化。

#### 4.2.1 异构优化

“嵩山”平台的单节点支持 32 个进程，在算法的模拟过程中，开启了 4 个进程用于验证。模拟完成后，增加进程数，以进一步提升模拟效率。表 1 分别针对 13 qubits，19 qubits 和 25 qubits 进行了开启 2～32 个进程的模拟耗时对比。

由表 1 中数据可知，当开启 4 进程时，模拟效率最高，增加进程数反而大幅降低了模拟效率。针对“嵩山”超级计算机系统的硬件架构分析后得知，平台节点使用的海光一号 CPU 将 32 核心划分为 4 个分区，每个分区通过硬件连接直接控制节点上的 1 个 DCU 加速器，因此当开启 4 个进程时，MPI 开启的 4 个进程分别来自不同的分区，避免了跨分区控制 DCU 加速器的额外开销[10]。因此在模拟时确定开启 4 个进程数，将进程编号与 DCU 编号绑定，使 4 个不同分区上的 4 个进程独立控制开辟 DCU 的显存空间，提高算法的模拟效率。

表 1 进程数与模拟耗时对比

Table 1 Comparison between processes and simulation time

| 进程数 | 2 | 4 | 8 | 16 | 32 |
|---|---|---|---|---|---|
| 15 qubits 耗时/s | 0.81 | 0.79 | 0.96 | 1.25 | 3.33 |
| 19 qubits 耗时/s | 3.06 | 2.19 | 2.22 | 2.83 | 12.31 |
| 25 qubits 耗时/s | 626.45 | 196.07 | 694.11 | 13 418.9 | 24 081.7 |

进程之间存在数据依赖，导致进程间通信时计算线程会处于闲置状态，随着模拟位数的提高，数据规模呈指数级增加，导致计算线程闲置状态越来越久，不利于提升程序效率。因此将数据分割，一次传输 64×256 个单位的数据，使 DCU 完成运算的时间开销分摊到 MPI 数据传输中，隐藏掉计算的时间开销，从而提高模拟效率。优化效果如图 11 所示。

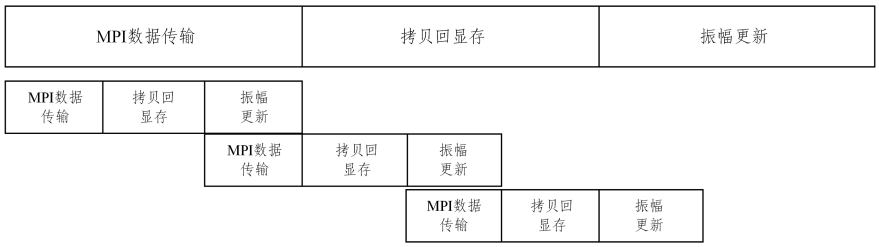



图 11 优化效果图(电子版为彩图)

Fig. 11 Diagram of optimization

#### 4.2.2 量子线路优化

概率幅存储在 4 个进程中，存储时将线路的低位连续存储在同一个进程，线路的高位分散存储，因此对最高位的两个量子比特施加量子门时，会产生进程间通信。为了减少进程间通信的次数，提高模拟效率，经过对量子线路的分析，发现初始化时被施加量子门的辅助量子比特是影响进程间通信次数的决定性因素。如图 12 所示，原线路红色为高位比特所在线路，每次迭代 Oracle 部分需要 4 次进程间通信去交换数据。

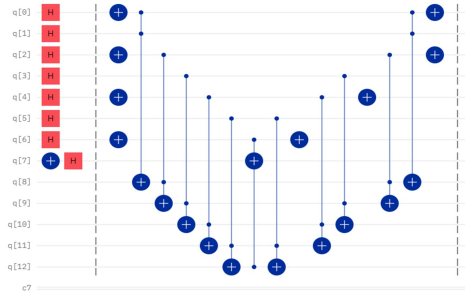


图 12 Oracle 优化前进程间通信次数(电子版为彩图)

Fig. 12 Communication times before Oracle optimization

通过调整上述辅助量子比特的位置到最高位，得到如图 13 所示的优化后 Oracle 线路，每次迭代进程间通信次数减少一次。

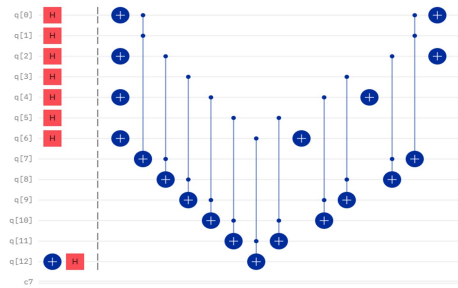


图 13 Oracle 优化后进程间通信次数

Fig. 13 Communication times between processes after Oracle optimization

同理，线路的平均反演算子操作的进程间通信次数也随之减少。优化后的 13 qubits 算法线路图如图 14 所示。由图可知，此时进程间通信主要在 Oracle 操作，平均反演算子操作不存在进程间通信，与优化前相比，每次 G 迭代都减少了 3 次进程间通信，整个模拟过程共减少了 $3\times(n/2+1)$ 次进程间通信，减少了近一半的通信次数。

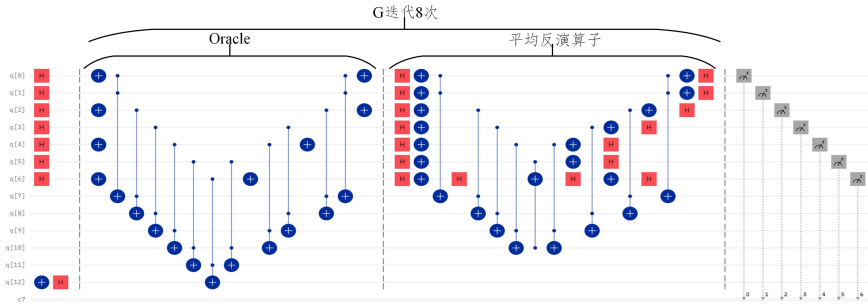



图 14 优化后的量子线路

Fig. 14 Quantum circuit after optimization

## 5 实验验证

### 5.1 量子线路验证

实验基于 IBM Q 平台验证，由于平台的量子计算机最高仅能提供 5 量子比特使用，因此使用 IBM Q 平台的量子模拟器 ibmq_qasm_simulator 验证，运行次数设定为 8192 次。

在验证量子线路时，首先选取 3 个不同的序列，从 11 qubits 到 31 qubits，针对每个不同序列都在 IBM Q 平台进行一次搜索验证，并统计目标序列的搜索成功次数，计算搜索成功率。由图 15 可知，搜索成功率均在 99.6%以上，验证了线路的正确性。

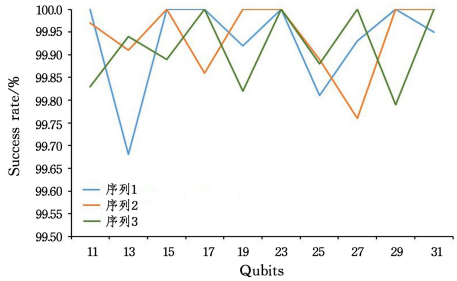



图 15 不同比特数的不同序列验证

Fig. 15 Verification of different sequences with different bits

随后针对 25 qubits 与 29qubits 各选取 9 个不同的序列，在 IBM Q 平台分别进行了搜索验证，并统计目标序列的搜索成功次数，计算搜索成功率。由图 16 可知，搜索成功率均在 99.7%以上。实验结果进一步验证了线路的正确性，同时也验证了线路的灵活性。由以上实验结果可知，量子线路验证成功。

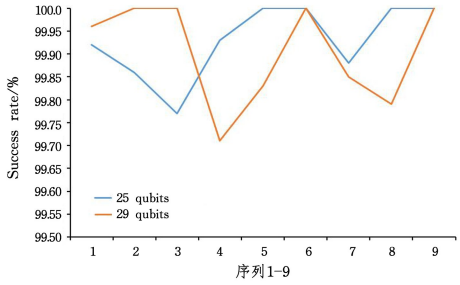



图 16 相同比特数的不同序列验证

Fig. 16 Verification of different sequences with the same bits

### 5.2 异构模拟验证

#### 5.2.1 实验环境

实验平台为嵩山超级计算机的单计算节点，搭载海光一号 CPU 与海光一号 DCU。海光一号 CPU 单片上拥有 32 个核心，最多可同时开启 64 个线程，稳定运行可达 3.0 GHz 以上；DCU 加速器采用类 GPU 架构，具有 16 GB 的全局内存。平台的单个计算节点上 CPU 和 DCU 的配比为 1∶4，拥有128 GB 内存空间，计算节点间配备 200 Gbps 的高速互通网络。程序由 HIP-Clang 实现，运行环境为 ROCm4.1.1。

#### 5.2.2 功能性测试

在 Grover 算法的串行版本中，使用 Hash 结构存储量子态概率幅，但 Hash 存储的优缺点十分明显。当模拟规模较小时，模拟效率很高，当算法模拟超过一定规模后，模拟耗时大幅度增加。表 2 列出了小规模模拟时针对同一个搜索解的串行与并行的耗时对比。可以看出，当模拟规模较小时，串行版本受益于 Hash 存储直接存取与键值对存储的优势，模拟效率高于并行版本。

表 2 运行耗时对比

Table 2 Running time consumption comparison

| Qubits | 5 | 7 | 9 | 11 |
|---|---|---|---|---|
| 串行耗时/s | 0.04 | 0.05 | 0.13 | 0.76 |
| 异构耗时/s | 0.71 | 0.71 | 0.71 | 0.74 |

但每增加一位量子比特，数据量都会翻倍，Hash 结构的存储面对大规模数据时，要考虑散列均匀性和存储冲突等问题，因此串行版本有很大的局限性。在“嵩山”超级计算机系统单节点上，串行版本的 Grover 算法的模拟上限为 26 量子比特，能给出正确结果的模拟上限只有 20 量子比特。本实验在异构模拟时采用了全概率幅模拟，使用浮点数结构体的数组存储量子态概率幅。此存储方式更适用于多进程并行时数据的连续分割，对分割后的数据使用多线程并行，通过二级并行优化线路中的密集计算，大幅提高了模拟效率。基于“嵩山”超级计算机系统的单计算节点，表 3 列出了大规模模拟时针对同一个搜索解串行与并行的耗时对比。

表 3 运行耗时对比

Table 3 Running time consumption comparison

| Qubits | 15 | 21 | 25 | 31 |
|---|---|---|---|---|
| 串行耗时/s | 5 | 2995 | 109863.58 | — |
| 异构耗时/s | 1.35 | 8.13 | 196.07 | 29679.5 |
| 加速比 | 3.7 | 368.39 | 560.33 | — |

随着模拟规模的增大，算法的迭代次数也会增加，因此模拟耗时会大幅度增加，这一点由表 3 数据可以直观看出。当模拟规模较小时，串行版本与并行版本计算效率都比较高，而并行版本的主要耗时是在 CPU 进程间的数据传输，因此加速

比很小；而当模拟规模增大时，串行版本的计算效率大幅度降低，并行版本的计算效率仍然保持较高的水平，因此加速比进一步得到提升。当模拟规模达到 25 量子比特时，加速比达到了 560.33。

本实验在“嵩山”超级计算机系统首次模拟了 31 qubits 规模的 Grover 量子搜索算法，不仅在“嵩山”平台单节点上提高了模拟规模的上限，且取得了优秀的加速比。本文使用的异构模拟思路与量子计算的天然并行性相契合，适用于具有清晰定义的量子线路的量子算法及应用。同时，Grover 算法能对机器学习的主成分分析阶段起到加速作用，因此模拟 Grover 算法可以促进量子机器学习与超算平台结合领域的发展。

**结束语** 本文针对 Grover 量子搜索算法的量子线路进行深入分析，相较于已有研究，通过优化 Oracle 操作和平均反演算子操作，降低了线路的宽度与深度；给出了任意规模任意解的通用量子线路伪代码，为后续 Grover 算法以及算法应用的研究提供了参考。但该研究仍存在辅助量子比特多的问题，量子线路的宽度有待进一步优化。

在量子线路的基础上，使用编程语言对其进行仿真，在“嵩山”超级计算机系统中进行了部署，完成了 Grover 量子搜索算法的 CPU+DCU 异构模拟。同时，进一步验证了量子计算在“嵩山”超级计算机系统上模拟的可行性，拓展了平台的研究领域，也为其他量子计算过程在该平台的模拟和验证提供了有力参考。本文仅在超算系统的单节点对 Grover 算法进行了模拟，节点利用率不够高，后续可以针对多节点展开工作。

## 参 考 文 献


[1] GROVER L K. A fast quantum mechanical algorithm for database search[C]// Proceedings of the Twenty-eighth Annual ACM Symposium on Theory of Computing. 1996:212-219.

[2] YIN A, HE K, FAN P. Quantum dialogue protocol based on Grover's search algorithms [J]. Modern Physics Letters A, 2019, 34(21):1950169.

[3] LIU X N, SONG H C, WANG H, et al. Overview of improvement and application of Grover algorithm [J]. Computer Science, 2021, 48(10):315-323.

[4] ZHANG K, KOREPIN V E. Depth optimization of quantum search algorithms beyond Grover's algorithm [J]. Physical Review A, 2020, 101(3):032346.

[5] CHEN H, GAO Y, ZHANG J. Quantum K-nearest neighbor algorithm[J]. Journal of Southeast University (Natural Science Edition), 2015, 45(4):647-651.

[6] SAHA A, SAHA D, CHAKRABARTI A. Circuit design for k-coloring problem and its implementation on near-term quantum devices[C]// 2020 IEEE International Symposium on Smart Electronic Systems(iSES)(Formerly iNiS). IEEE, 2020:17-22.

[7] HABIBI M R, GOLESTAN S, SOLTANMANESH A, et al. Power and Energy Applications Based on Quantum Computing: The Possible Potentials of Grover's Algorithm[J]. Electronics, 2022, 11(18):2919.

[8] LIU X N, JING L N, WANG L X, et al. Large-scale quantum Fourier transform simulation based on Shenwei 26010 processor [J]. Computer Science, 2020, 47(8):93-97.

[9] LIU X N, JING L N, WANG L X, et al. Quantum Fourier Transform Simulation on Sunway TaihuLight[C]// 2020 15th International Conference on Computer Science & Education (ICCSE). IEEE, 2020:833-837.

[10] XIE J M, HU W F, HAN L. Quantum Fourier Transform Simulation Based on" Songshan[J]. Supercomputer System, 2021, 48(12):36-42.

[11] SCHWABE P, WESTERBAAN B. Solving Binary with Grover's Algorithm[C]// Security, Privacy, and Applied Cryptography Engineering: 6th International Conference, SPACE 2016, Hyderabad, India, December 14-18, 2016, Proceedings. Cham: Springer International Publishing, 2016:303-322.

[12] SONG H C, LIU X N, WANG H, et al. Integer decomposition based on Grover search algorithm[J]. Computer Science, 2021, 48(4):93-97.

[13] FERNANDES D, DUTRA I. Using Grover's search quantum algorithm to solve Boolean satisfiability problems: Part I[J]. XRDS: Crossroads, The ACM Magazine for Students, 2019, 26(1):64-66.

[14] MAGNIEZ F, SANTHA M, SZEGEDY M. Quantum algorithms for the triangle problem[J]. SIAM Journal on Computing, 2007, 37(2):413-424.

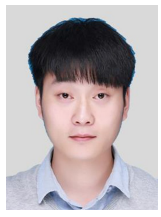
[15] KAIN B. Searching a quantum database with Grover's search algorithm[J]. American Journal of Physics, 2021, 89(6):618-626.

[16] MANDVIWALLA A, OHSHIRO K, JI B. Implementing Grover's algorithm on the IBM quantum computers[C]// 2018 IEEE International Conference on Big Data. IEEE, 2018:2531-2537.

[17] VEMULA D R, KONAR D, SATHEESAN S, et al. A Scalable 5, 6-Qubit Grover's Quantum Search Algorithm [J]. arXiv: 2205.00117, 2022.

[18] CHEN Y. Optimization of Grover's quantum search in multi-solution and hyperbit spaces[D]. Xi'an: Northwest University, 2021.

[19] PRONIN C B, OSTROUKH A V. Developing Mathematical Oracle Functions for Grover Quantum Search Algorithm[J]. arXiv: 2109.05921, 2021.

**DU Shuaiqi**, born in 1996, postgraduate. His main research interests include quantum algorithm and high-performance computation.

**LIU Xiaonan**, born in 1977, Ph.D, associate professor, master's supervisor. His main research interests include quantum algorithm and high-performance parallel computation.

（责任编辑：柯颖）