

# 量子计算模拟及优化方法综述

喻志超<sup>1</sup>, 李扬中<sup>1</sup>, 刘磊<sup>2,1</sup>, 冯圣中<sup>1</sup>

(1. 国家超级计算深圳中心(深圳云计算中心), 广东 深圳 518055;

2. 中国科学院计算技术研究所 计算机体系结构国家重点实验室, 北京 100190)

**摘要:** 在处理某些大规模并行问题时, 量子计算因量子位独特的叠加态、纠缠态特性, 相比经典计算机在并行处理方面具有更明显的优势。现阶段, 物理量子比特计算机受限于可扩展性、相干时间和量子门操作精度, 在经典计算机上开展量子计算模拟是研究量子优越性和量子算法的有效途径。然而, 随着量子比特数量的增加, 模拟所需的计算机资源呈指数增长。因此, 研究大规模量子计算模拟在保证计算准确度、精度及效率的情况下减少模拟所需资源具有重要的意义。针对目前的量子计算模拟的相关工作, 本文梳理总结了基于经典计算机的量子计算模拟的基本方法, 并讨论了量子线路模拟的优化方法。本文的主要内容包括: (1) 基于量子比特、量子门和量子线路阐述量子计算的基本原理; (2) 总结和分析不同量子计算机模拟方法的思路和优缺点; (3) 列举目前常见的量子计算模拟器; (4) 针对量子计算模拟的开销问题, 讨论基于超级计算机集群的量子计算模拟优化方法。

**关键词:** 量子计算; 量子线路; 概率幅模拟; 量子计算模拟器; 超级计算机 开放科学(资源服务)标志码(OSID):



## Quantum computing simulation and optimization methods: A survey

Yu Zhichao<sup>1</sup>, Li Yangzhong<sup>1</sup>, Liu Lei<sup>2,1</sup>, and Feng Shengzhong<sup>1</sup>

(1. National Supercomputing Center in Shenzhen (Shenzhen Cloud Computing Center), Guangdong, Shenzhen 518055, China;

2. State Key Laboratory of Computer Architecture, Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China)

**【Abstract】** Based on the unique superposition states and/or entangled states of qubits, quantum computing has obvious advantages in dealing with some problems requiring large-scale parallel processing than classical computers. At present, the physical quantum computer is faced with the limitation of scalability, coherence time and precision of quantum gate operation, and it is an effective way to carry out quantum computational simulation on classical computers. However, as the number of qubit increases, the computer resources required for quantum simulation grow exponentially. Therefore, it is of great significance to study the time and space resources required for the simulation while ensuring the computational accuracy. In view of the current research work of quantum computational simulation, this paper summarizes the basic methods of quantum computational simulation based on classical computers and discusses the optimization methods of quantum circuit simulation. The main content is as following: (1) the basic principles of quantum computing are expounded from quantum bits, quantum gates and quantum circuits. (2) the ideas and advantages and disadvantages of different methods of amplitude simulation are summarized and analyzed. (3) the common quantum computing simulators are listed. (4) the simulation methods of quantum computing based on supercomputer are discussed.

**【Key words】** quantum computation; quantum circuit; amplitude simulation; quantum simulator; supercomputer

DOI:10.19678/j.issn.1000-3428.0061787

**基金项目:** 国家重点研发计划(2017YFB1001604, 2018YFB0204403); 国家自然科学基金项目(62072432)

**作者简介:** 喻志超(1992—), 男, 博士, 博士后研究员, 主要研究方向为高性能计算与量子计算; 李扬中, 博士, 高级工程师; 刘磊, 博士, 副研究员; 冯圣中, 博士, 研究员。

## 0 概述

量子计算 (Quantum Computing) 是一种遵循量子力学规律调控量子信息单元进行计算的新型计算方式。由于量子力学叠加态的存在, 量子计算机与经典计算机相比具有更强大的并行信息处理能力, 并有望解决经典计算机无法解决的问题。量子计算被认可的潜在应用领域包括密码学<sup>[1]</sup>、量子化学<sup>[2]</sup>、材料科学<sup>[3]</sup>和机器学习<sup>[4]</sup>等。

研究量子计算技术, 需要开展精确的量子门操作和纠错、容错等实验, 以及模拟量子叠加、纠缠等状态, 构造足够复杂的相干量子计算实验体系, 并开展适用的量子计算模型和算法研究。量子计算的主要研究方向包括建造含有更多量子比特的通用量子计算机、设计更有效的量子算法以及优化量子程序编译器<sup>[5-6]</sup>, 从而使得量子计算机能被广泛使用。现阶段的量子计算机属于 Noisy Intermediate Scale Quantum (NISQ) 计算机<sup>[7-8]</sup>, 量子位的数量尚不过百, 量子位状态脆弱且易受干扰, 无法保证量子程序的可靠执行, 也无法适用于求解大规模的科学计算问题。自从谷歌提出“量子优越性” (Quantum Supremacy) 以来, 如何展示现阶段量子对于经典计算机的优势一直是研究热点<sup>[9-11]</sup>。由于量子比特受到噪声的影响易失去量子特性, 基于量子纠错码的容错架构可能是利用量子计算机执行大规模计算的重要基础, 因此, 设计一种使用适度资源即能有效对抗实际噪声的实用量子纠错也是量子计算核心问题之一<sup>[12],[13]</sup>。此外, 探索新的效率明显优于经典计算机算法的量子算法, 也已经成为量子计算理论研究的一个重要分支<sup>[14]</sup>。

由于已知量子运行规律, 可使用经典计算机模拟量子计算。量子计算模拟器相比于物理量子比特计算机的突出优势是其功能齐全, 可执行的量子程序规模不会受到物理量子计算机保真度的制约, 是理想的量子算法测试平台<sup>[15]</sup>。量子计算模拟包括计算并存储所有量子态、部分或者单个量子态概率幅和含噪声模拟多种形式, 不同的模拟方法适应于不同的量子计算目标。许多研究机构开发了量子线路模拟器用来支撑量子计算领域复杂问题的模拟, 如“太章”模拟器<sup>[10-11]</sup>、ProjectQ<sup>[16]</sup>、qHiPSTER<sup>[17]</sup>和 QuEST<sup>[18]</sup>等。量子模拟器都可被部署于分布式超级计算机集群, 利用超算集群的运算性能实现量子计算的高效模拟, 模拟过程中需要考虑子任务划分和通讯优化问题, 通过分布式节点上任务的合理分配和减少通信成本的设计策略, 以达到均衡分布式节点间负载的目标。

为了能够模拟具有更多量子比特数、深度更深的量子线路, 如何减少模拟所需存储和提高模拟计算效率是关键问题。目前对算法的优化改进, 使得模拟器在实际运用中已经取得不小的突破。针对目前的量子计算模拟的研究工作, 本文综述了实现量子计算模拟的基本方法, 总结了量子线路模拟的优化方法和目前已有的模拟器, 并阐述了在超算上模拟量子计算的核心问题。

## 1 量子计算原理

本节我们将主要介绍量子计算理论中的背景知识, 包括量子比特、量子门和量子线路等。

### 1.1 量子比特 (Qubit)

经典计算机的比特位只有 0 和 1 两种状态, 而

量子比特中这样确定的状态则被称为基本态，除此之外还有可能是两种状态的叠加态。一个量子比特（量子位）的状态（状态向量 $|\psi\rangle$ 表示）可以看作是由两个基本态（计算基态向量 $|0\rangle$ 和 $|1\rangle$ ）组成的二维复空间向量。根据量子规律，单量子比特可以是两种基本态叠加成的任一种状态。可以用状态向量公式表示为：

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle. \quad (1)$$

其中系数 $\alpha$ ， $\beta$ 为复数值的概率幅， $|\alpha|^2$ 和 $|\beta|^2$ 分别为对应状态的概率且满足 $|\alpha|^2 + |\beta|^2 = 1$ 。

$n$  量子比特的状态向量可以表示为 $2^n$ 个基态的复合态：

$$|\psi\rangle = \alpha_{0\dots 00}|0\dots 00\rangle + \alpha_{0\dots 01}|0\dots 01\rangle + \dots + \alpha_{1\dots 11}|1\dots 11\rangle \quad (2)$$

也可以表示为 $|\psi\rangle = \sum_{i=0}^{2^n-1} \alpha_i |i\rangle$ ，其中 $\sum_{i=0}^{2^n-1} |\alpha_i|^2 = 1$ 。

## 1.2 量子门 (Quantum Gate)

经典计算使用逻辑门操作经典比特的状态，量子计算则使用量子门操作量子位状态。量子态 $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ 满足归一约束 $|\alpha|^2 + |\beta|^2 = 1$ ，经过量子门作用后得到量子态 $|\psi\rangle = \alpha'|0\rangle + \beta'|1\rangle$ 依旧满足归一约束，所以量子门相应矩阵 $U$ 须为酉性<sup>[19]</sup>，即 $U^*U = I$ ， $U^*$ 为 $U$ 的共轭转置。因此，量子态的制备和任何对量子态的操作都是酉操作，量子门是对所选量子位进行酉操作的算子。

按照操作涉及的量子比特的个数可以将量子门分为单量子比特门和多量子比特量子门。单量子比特操作中最常用的主要是 Hadamard 门、Pauli-X 门、

表 1 量子门操作  
Table 1 quantum gates

量子门名称	符号	矩阵形式
Hadamard 门	H	$\frac{\sqrt{2}}{2} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$
Pauli-X 门	X	$\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$
Pauli-Y 门	Y	$\begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}$
Pauli-Z 门	Z	$\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$
相位门	S	$\begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix}$
受控非门	CNOT	$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$

Pauli-Y 门、相位门 (S 门) 等。多量子比特操作有 CNOT 门、受控 Z 门 (C-Z 门)、Toffoli 门等。表 1 为常用的量子门操作及其矩阵形式。

以 Hadamard 门为例，假设一个目标量子比特为  $k$  的 Hadamard 门，则其概率幅计算和操作更新规则为：

$$\begin{cases} \alpha'(*\dots*0_k*\dots*) = \frac{\sqrt{2}}{2} [\alpha(*\dots*0_k*\dots*) + \alpha(*\dots*1_k*\dots*)] \\ \alpha'(*\dots*1_k*\dots*) = \frac{\sqrt{2}}{2} [\alpha(*\dots*0_k*\dots*) - \alpha(*\dots*1_k*\dots*)] \end{cases} \quad (3)$$

## 1.3 量子线路 (Quantum Circuit)

量子算法是一系列基础的量子门组合而成，实现的形式称为量子线路。量子线路以量子比特为基本的存储单元，将量子逻辑门连接在一起来实现特定的计算功能，可以直观地表示出量子算法的结合和复杂度。本质上，量子计算可以描述为对包含量子比特的量子系统实施一系列酉变换，然后对变换之后的状态进行测量，最后得到计算结果的过程。因此，量子计算的过程包括初态设置、状态变换和测量三个步骤。图 1 为 2 个量子比特的 Grover 搜索算法线路图<sup>[1]</sup>，其中每一条线代表一个量子比特，而量子门操作使用不同的方块表示。

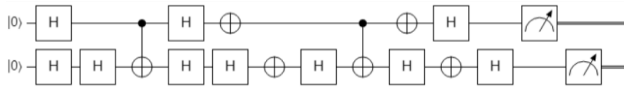
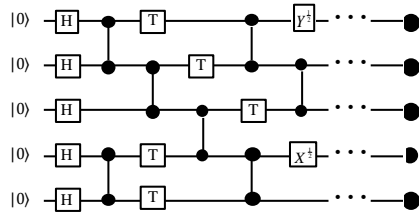


图 1 Grover 量子搜索算法示意图

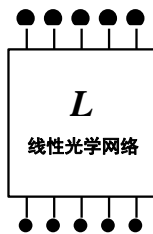
Fig. 1 Schematic diagram for the Grover's search quantum algorithm

“量子优越性”是量子计算研究路线中的一个重要里程碑，即现阶段实现针对特定问题的计算能力超越经典超级计算机。为了研究量子优越性，研究人员提出了一些非常适合于展现出量子设备计算能力的特定问题，包括随机量子线路采样 (Random Circuit Sampling)<sup>[9,18]</sup>、IQP 线路 (Instantaneous Quantum Polynomial)<sup>[21]</sup>和高斯玻色采样 (Boson Sampling)<sup>[20-21]</sup>等。谷歌量子研究团队所针对的问题是随机量子线路采样问题，我国中科大团队致力于高斯玻色采样问题的研究。

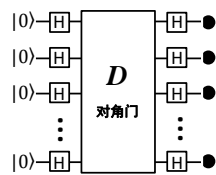
谷歌的“Sycamore”随机量子线路是在特定图结构上，周期地作用单量子比特门和两量子比特门，且最终以单量子比特门结束<sup>[9,22,23]</sup>。其中，单量子比



(a)



(b)



(c)

图 2 实现量子计算优越性的三种量子线路，

(a)随机量子线路；(b)玻色采样；(c)IQP 线路

Fig. 2 Three different circuits for achieving quantum supremacy, (a) random quantum circuits, (b) boson sampling, (c) IQP circuits.

特门均匀随机取自于一个单量子门的集合，两量子比特门为

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\theta) & -i \sin(\theta) & 0 \\ 0 & -i \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 0 & e^{-i\phi} \end{bmatrix}$$

其中， $\theta = 90^\circ$ ， $\phi = 30^\circ$ 。随机量子线路采样问题是一个足够复杂且又适合当前带噪声量子计算机求解的计算问题，能够以较少的门操作数量来实现较大纠缠的量子态。随机量子线路采样是最近展现量子优越性的主流候选问题，主要有三个原因：(1) 随机线路采样问题非常适合于在二维结构的超导量子计算芯片上实验实现；(2) 已有很多理论工作证明了随机线路采样问题的困难性；(3) 随机量子线路采样满足反集中 (anti-concentration) 性质，表现为噪声对输出结果的概率造成的影响更小。

玻色采样过程的模拟是计算给定输入光子分布时得到某种输出光子分布的概率，其最重要的部分在于求解矩阵积和式，而计算复矩阵的积和式对经典计算机来说是很困难的。IQP 线路代表一类简化的量子线路模型，线路的初始量子态都为零态，之后对每个量子比特作用一个 Hadamard 门。接着，在这些量子比特上作用对角的门，最后再次在每个量子比特上作用 Hadamard 门。

### 1.4 量子操作系统 (Quantum OS)

量子操作系统是量子计算机资源的管理者，负责管理、监控、配置量子位硬件资源，调度量子程序运行的先后顺序，控制量子程序单独或并发映射，并提供具体的量子线路映射策略，保证量子程序的可靠度。为了充分利用量子计算机强大的计算能力，

文献<sup>[26]</sup>倡导建立量子操作系统来管理量子计算机硬件资源。文献<sup>[5-6]</sup>在国内率先开展了量子操作系统的原型研制,提出 QuOS 原型,用于并发多量子程序在量子芯片上的映射、调度和优化;该系列工作设计了优化健壮量子位资源利用率的程序映射策略、跨程序量子位交换的映射方法和多量子程序调度机制,为量子操作系统的软硬件栈的设计与优化提供了原型系统和参考经验。

### 1.5 量子模拟 (Quantum Simulation)

以上内容是量子计算相关的简要背景知识。目前,由于量子计算机的真机还处在发展阶段,通常单机可用的量子位数量较少(在 100 以内)且量子位状态保持不稳定,不能完成有实际意义的、较大规模的量子运算。在量子硬件实现方案没有确定下来之前,很多量子算法所展现出的巨大计算潜力并未得到真机的实验验证,目前主要是基于数学理论或者基于经典计算机的形式进行模拟验证。

量子计算机除了和经典计算机有相似模型之外,还存在一些特有的计算模型。比较常用的量子模型主要包括量子线路模型、绝热量子计算模型、One-way 量子计算模型和拓扑量子计算模型等。虽然这些模型设计的出发点不同,但其计算结果的输出是一致的,且相互之间可以进行转化。在处理某类特定问题的计算中,采用不同的算法计算复杂度可能不一样,选择一款好的计算模型能够简化算法的实现流程。

采用超级计算机(超算)模拟量子计算的过程,是目前研究量子计算,辅助量子计算机执行任务的一种方式。国内外科研机构、公司、高校积极开展

相关工作。量子模拟内涵丰富,本文将从量子计算模拟方法、国内外量子模拟器、超算集群模拟等几个方面深入介绍,后文将详述相关内容。

## 2 量子计算模拟方法

量子计算模拟中,主要是利用输入不同的矩阵来改变要计算的量子态的状态,从而引起量子线路状态的变化。但在量子线路中,由一个个的量子逻辑门级联而成对于逻辑门的操控精度需要有极高的要求,所以若量子比特门的保真度不高,逐级增大,则会严重影响最终计算结果。受制于量子比特自身的物理特性,计算错误的发生常常无法避免,但可以避免错误对最终计算结果的影响。此外,评估量子计算中的错误严重性,对于改进器件的设计、优化控制参数以及使用缓解协议最小化错误至关重要。

### 2.1 量子计算概率幅模拟

量子计算概率幅模拟方法是通过对概率幅的计算在经典计算机上实现对量子计算的模拟。在量子线路模拟过程中,通过计算一个量子线路后终态在各个基态上的概率幅获取量子计算的模拟结果。该方法可分为全概率幅模拟和单概率幅模拟,其主要区别在于,全概率幅模拟一次模拟计算能够算出量子态所有概率幅,单概率幅模拟只计算出所有概率幅中的一个。除此之外,还有部分概率幅模拟,目的是在更低的硬件条件下实现更高的模拟效率。

全概率幅模拟方法直接计算并存储全部量子态,计算的空间复杂度随着模拟的量子比特数指数增加。以  $n$  个量子比特为例,需要计算  $2^n$  个复数形式的量子态来描述该量子系统,如果采用双精度浮

点数表达概率幅,则需要存储空间的量子态为  $2^{4+n}$  字节。目前即使在大型超级计算机上,也很难模拟一个超过 50 个量子比特的量子线路。一方面,一个包含 50 个量子比特的线路,完整的状态有  $2^{50}$  个,即便采用单精度浮点来存储,也需要约 4PB 的内存空间,这几乎达到了当今超级计算机可使用内存的最大值。另一方面,即使超级计算机的内存够用,处理这些海量数据也比较困难,原因在于数据在超算系统的网络传输中将产生巨大的通信负荷<sup>[27]</sup>。

经典的方法包括薛定谔方法<sup>[28]</sup>和费曼方法<sup>[29]</sup>。薛定谔方法存储全部量子态,按顺序计算作用不同量子门后的全部振幅。对于  $n$  个量子比特所需  $2^{4+n}$  字节的内存,每个量子门需要  $O(2^n)$  的计算时间,对于深度为  $m$  的量子线路,所需模拟时间约  $O(mn2^n)$ 。费曼方法为对于终态的每个分量,枚举通过量子线路中量子门能从初态到这一态的变化路径,将所有这些路径的值累加即可得到对应的概率幅,计算过程中存储一条路径最多需要  $O(n^2m)$  的空间,对于深度为  $m$  的量子线路,计算一个状态的振幅需要  $O(2^{mn})$ 。文献<sup>[8]</sup>将上述两种方法相结合提出薛定谔-费曼方法,以达到平衡算法的时间和空间复杂度的目标。此外,研究人员提出了数据压缩方法<sup>[29]</sup>、辅助变量法<sup>[30]</sup>、MPS 方法<sup>[31]</sup>、PEPS 方法<sup>[32]</sup>和部分振幅模拟<sup>[33]</sup>等,这些方法使得 45 个量子位以上全概率幅的大规模量子计算模拟得以实现。

在量子计算模拟中可将量子线路转换为具有酉正性的特殊张量网络表示,其单个概率幅的计算可以转换成张量网络缩并问题。张量网络缩并算法不直接存储实时的量子态,而是将初始量子态和测量

基矢当成很多一维张量,这些张量和量子门操作组成一个时间和空间维度上的三维张量网络。如何对一个数百节点的高维张量网络,选取一条接近最优的张量收缩路径,是张量网络收缩算法的关键问题。

文献<sup>[18,34]</sup>采用张量网络方法计算特定量子基态的概率幅,在模拟过程中利用对角矩阵缩减计算项数,并且采用无向图模型,优化张量网络缩并顺序,从而缩减张量网络缩并开销。文献<sup>[35]</sup>改进张量网络缩并顺序优化算法,使其适用于分布式集群,并利用分布式计算的特性,拆分张量网络计算任务为子任务,使每个子任务的执行时间尽可能短,最终在阿里云集群上用 1PB 内存实现了 81 量子位、40 深度的量子线路模拟。

谷歌提出的随机量子线路结构是一个高度正则的张量网络,张量网络进行缩并时,正则性将会表现为占用大多数计算时间的结点,构成了一条“主干”。对几乎所有的运算时间都花在节点的个数相较“分支”节点较少的“主干”上。因此,文献<sup>[8]</sup>提出了一种基于张量网络收缩的模拟方法,该方法主要分为三个步骤:图划分、局部优化和动态切片。首先该方法将张量网络的节点划分为若干块,包含“主干”和“分支”部分,持续重复对“主干”部分划分直到缩并成点后,对剩下点再进行计算缩并,通过识别和优化这样的“主干”,该方法可以比 Google 的模拟算法快 200000 倍。

此外,文献<sup>[30]</sup>基于张量网络态来求解随机量子线路采样问题。张量网络态算法,可以认为是对张量网络收缩算法选取了一条特定的收缩路径,即预先收缩时间维度,然后收缩剩下的空间方向的二维

表 2 常见几类量子计算模拟方法特点及优缺点

Table 2 Features, advantages and disadvantages of common quantum computing simulation methods

方法	方法特点	优缺点
薛定谔方法	按顺序计算作用不同量子门后的全部振幅	测量多个振幅与测量单个振幅的复杂度基本一致, 算法的空间复杂度随量子比特数指数增加
费曼方法	分别计算每条路径对应张量并求和	对 $m$ 层的线路计算一个概率幅需要 $O(2^{mn})$
薛定谔-费曼算法	将量子态存储为两个张量	牺牲时间复杂度来降低一部分空间复杂度
张量网络收缩算法	将初始量子态和测量基矢当成一维张量, 这些张量和量子门操作组成一个时间和空间维度上的 3 维张量网络, 将整个张量网络直接收缩即可得到单个振幅	测量多个振幅的复杂度随振幅数量线性增长
张量网络态算法	对节点上的张量利用奇异值分解或 QR 分解进行自动压缩, 以降低节点张量的维度	仅需要寻找一个数十个节点的平面图上的最佳收缩路径, 很少或者完全不需要经验性参数

的张量网络, 通过对节点上的张量利用奇异值分解或 QR 分解进行自动压缩, 以降低节点张量的维度。

文献<sup>[36]</sup>提出“Big-Head”方法, 并使用 60 个英伟达 GPU 组成的小型计算集群, 5 天的时间内完成了谷歌量子随机线路的模拟。其所采用的“Big-Head”方法将终态的  $n$  个量子比特分为两组, 大小分别为  $n_1$  和  $n_2$ 。任意位串  $s$  可视为两条位串  $s_1$  和  $s_2$  的拼接, 其概率可计作  $P_U(s) = P_U(s_1; s_2)$ 。当  $n_2=0$  时, 该方法和张量网络的单概率幅估算一样。方法最重要的环节就是找到一个满足上述条件, 时间与空间复杂度都比较合适的缩并阶数。使用分割算法将整个张量网络切割成两份  $G_{\text{head}}$  与  $G_{\text{tail}}$ , 切割边  $n_c$  条。该分割算法会在对  $n_1$  和  $n_2$  大小限制的基础上最小化。要找到分割点需要保证同时缩并两个部分的计算复杂度不会太大, 为了达到这个目的, 方法中用一种聚类算法将每个子图都分割成更小的子图, 直到每个子图都足够小。

常见几类量子计算模拟方法的特点及优缺点如表 2 所示。全概率幅模拟方法的计算复杂度随模拟

量子线路的深度线性增长, 因此该类方法适合模拟具有较少量子比特数, 任意深度的量子线路, 适用于需要获取全部模拟结果等场景。部分概率幅模拟方法的效率高, 相比于全概率幅模拟方法能够模拟具有更多量子位的浅深度的量子线路。张量网络态算法可以用几乎相同的模型来研究不同硬件架构下且不同深度的随机量子线路, 原因在于其通常存在一个简单的路径搜索算法, 仅需寻找一个数十个节点的平面图上的最佳收缩路径, 因而实现起来只有很少或者完全不需要经验性参数。而张量网络收缩算法则往往需要考虑具体线路来确定其经验参数。相比而言, 张量网络收缩算法在更大空间搜索最优路径, 原则上来说具有更大的优化空间。

## 2.2 针对 Clifford 线路的量子计算模拟

量子比特通常是不稳定的, 为了维持逻辑量子比特的准确性, 需要进行量子纠错。和传统的纠错方法不同, 由于量子不可克隆定理、量子叠加态塌缩 (或称波函数塌缩) 的限制, 对量子比特进行纠错必须加入辅助量子比特。相对于经典比特, 量子

比特还存在相位上的错误，这也为量子纠错带来了更大的复杂性。通常需要上千个物理量子比特来实现一个高保真的逻辑量子比特，而这也会为系统带来大量的噪声。针对容错、抗退相干、保护目标数据的准确存储以及正确的编码、运行等问题，目前科研人员提出了一些解决方案，包括无消相干子空间 (Decoherence Free Subspace, DFS)、拓扑量子计算以及基于量子纠错码的容错量子计算等<sup>[37]</sup>。

因为噪声而导致的计算错误问题，可以使用量子纠错码来解决在错误的分布满足某些条件的情况下，可以把最终计算结果出错的概率降得任意低，这被称作容错量子计算。和经典纠错相比，量子纠错不仅需要处理比特错误，还需要处理相位错误。容错量子计算在量子计算码的基础上，通过级联的方式实现量子逻辑门，以至于当物理操作的出错率低于容错阈值时，量子逻辑门的错误率急剧降低，从而实现可靠的量子计算。

量子纠错码与级联结合，实现容错量子计算要求，横向实现量子逻辑门，使得错误的位数和传播控制在量子纠错码的纠错范围内。量子纠错码上可以横向实现的量子逻辑门基本来自于 Clifford 群。Clifford 群指把 Pauli 群共轭映射到 Pauli 群的操作形成的群，包括 Hadamard 门，相位门 S 和受控非门 CNOT 等。

Stabilizer codes 是一种基于“稳定子(stabilizer)”的定义通用量子纠错码的框架，来表述和操作需要的量子态。它使用 Clifford group 中的量子门对稳定子态进行编码和解码。拓扑码 (topological code) 是一种比较容易实现的编码方式，可以构造局部的

稳定子测量来避免远程的量子比特的交互，这可以降低对设备的要求。拓扑码的一个显著的优点在于，人们可以通过增加单位晶胞 (unit cell) 的大小，来提升检错和纠错的容量。然而，随着编码容量的增大，“解码”的任务必将会对量子态造成干扰 (non-trivial measurement)，因而设计有效的解码算法成为拓扑码最重要的任务。表面码 (Surface code) 的关键问题在于，要使用它来实现通用量子门集合是比较困难的，它缺少 Toffoli 门以及  $\pi/8$  门。在容错量子计算的标准模型中，通常使用魔法态注入 (magic state injection) 过程来构造一个完整的量子门集合。

随机基准测试 (Randomized benchmarking) <sup>[38]</sup> 和量子过程层析 (Quantum Process Tomography, QPT) <sup>[39]</sup> 可以分别测量噪声量子信道的平均门保真度和完整信息，但这两种方法在几个量子位系统中是有效的。交叉熵基准测试可以用于验证多量子比特系统，但其无法直接应用于经典计算机上，不可模拟的量子电路。量子计算机的并行计算能力随量子比特数目增加而指数上升，同时，表征量子线路误差的难度也会指数上升<sup>[9]</sup>。文献<sup>[40]</sup>提出一种基于 Clifford 采样的方法来描述量子线路的误差，克服了已有的误差表征方法，但都只适用于少数几个比特的量子线路，很难被用在未来成百上千个量子比特的量子线路表征中。Clifford 采样将一个指数复杂度的问题转换成了多项式的问题，具有很高可扩展性，适用于大规模量子线路参数的优化以及筛选等多个领域。

Gottesman-Knill 定理指出一个仅由 CNOT、



Hadamard 门和相位门组成的量子线路可以在经典计算机上有效地模拟,文献<sup>[41]</sup>改进了定理中的算法,去除对高斯消元法的依赖,并给出两种稳定态内积计算的有效算法。针对 Clifford + T 门线路的模拟,研究人员开展了一系列的工作,包括模拟包含少量 T 门的量子线路和考虑 nonstabilizer 的输入态+ Clifford 线路的模拟等等<sup>[42-43]</sup>。为了解决量子线路的容错和可靠性问题,研究人员提出了量子线路的一系列结构等效规则和优化操作策略,以尽量减少 T 门的数量,增加 T 门的深度,最小化线路级,减少容错实现代价并提高线路可靠性<sup>[44]</sup>。

### 2.3 针对含噪声线路的量子计算模拟

嘈杂的中尺度量子 (NISQ) 设备的计算能力对于量子信息科学具有基础和实际重要性。如果没有减少或绕过物理噪声的策略,在苛刻的条件下任何量子程序的执行可能都会失败。降噪技术的策略是设计更稳健的量子位来延长时间,执行更精确的门操作以提高它们的可靠性等。随机编译(Randomized compiling)是一种常用的策略,即在量子线路中插入随机门。所有相干误差和非马尔可夫噪声都可以转换为随机泡利误差,在保留一个量子线路的逻辑操作的同时误差容易被检测和纠正。虽然噪声对单个随机线路的影响可能有所不同,但在多个随机线路上的预期噪声被打乱,并调整成一种简单的随机形式。

文献<sup>[45]</sup>中考虑了在对称或非对称去极化噪声下一类大的随机通用量子线路,并证明了如果每个量子门的噪声是恒定的,这些量子线路的输出可以用系统大小的时间代价多项式进行经典有效的模拟。

文献<sup>[46]</sup>结合了纯态分解和低维基投影的思想来有效地模拟噪声量子线路。提出将混合密度矩阵分解为一个类似于纯态集合的低秩矩阵,将门和 Kraus 运算子应用于该低秩矩阵,并计算了输出密度矩阵和概率分布。在该过程中包括对密度矩阵的迭代压缩,以保持一个具有最小误差的数值紧凑形式。

## 3 量子计算模拟器

量子模拟器是在经典计算机上实现对量子计算机的模拟。近年来,基于量子计算原理,国内外许多公司机构设计并开发了大量的量子模拟器,例如, QisKit 量子模拟器 (IBM)<sup>[47]</sup>、QDK 量子模拟器 (Microsoft)<sup>[49]</sup>、“太章”量子模拟器 (阿里巴巴)<sup>[10-11]</sup>等。现有的量子模拟器规模仍属于嘈杂的中型量子 (NISQ),模拟量子比特数在 100 以内,主要受限于经典计算机存储技术和处理能力,本文系统梳理了国内外常见的量子模拟器,如表 3 所示。

Qiskit<sup>[47]</sup>是由 IBM 推出的 Python 量子开发工具集,共有 Terra (量子计算基础库)、Aer (带有噪声的模拟器)、Aqua (量子算法)、Ignis (检测和去除实际量子计算机噪声)四大组件,该量子模拟器可用于机器学习、自然科学、经济金融以及解决

表3 量子模拟器

Table 3 Quantum simulators

量子模拟器	研究机构	时间	开发语言	最大量子比特
QisKit	IBM	2017	Python	63
QDK	Microsoft	2017	Q#	30
太章	阿里巴巴	2018	Python	81
HiQ	华为	2018	Python	42
ProjectQ	苏黎世联邦理工学院	2018	Python	22
Cirq	Google	2018	Python	20
Qpanda	本源量子	2019	C++	49
QuEST	QTechTheory	2019	C++	38
IQS	Intel	2020	C++	42
量浆	百度	2020	Python	25

最优化问题。

QDK<sup>[49]</sup>是 Microsoft 推出的 Q#量子开发工具集, Q#是一款面向量子计算机开发的程序语言, 结合了 Python、C#等多种语言的设计元素, 对量子计算机的逻辑进行了高度的封装。QDK 包括 Q#编译器、量子库和量子模拟器, 该量子模拟器可应用于量子机器学习、量子化学等领域。

太章<sup>[10-11]</sup>是阿里巴巴推出的 Python 量子开发工具集, 目前存储技术最高存储不到 60 量子比特的量子态, 太章采用张量网络收缩的动态拆分办法, 减少了量子线路模拟的资源占用, 支持量子算法和纠错。

HiQ<sup>[50]</sup>是华为推出的 Python 量子开发工具集, 包括华为 HiQ 量子模拟器、HiQ Pulse、HiQ Fermion 等功能模块, 目前主要应用于量子化学领域。

ProjectQ<sup>[16]</sup>是苏黎世联邦理工学院推出的 Python 量子开发工具集, 该工具提供了量子模拟器、编译器框架、编译器插件和资源估算等模块, 可实现量子线路的模拟。

Cirq<sup>[51]</sup>是 Google 推出的 Python 量子模拟器, 用户可通过 Cirq 实现对量子线路的精确控制, 同时开源的 OpenFermion-Cirq, 可应用于量子化学领域。在 Cirq 的基础上, Google 应用量子机器学习原理开源 tensorflow-quantum (TFQ), 该框架可有效应用于量子分类、量子线路优化等问题。

Qpanda<sup>[52]</sup>是本源量子推出 C++量子工具集, 该工具集量子模拟器基础上, 实现了大量量子算法, 例如, 量子近似优化算法 (QAOA)、Deutsch-Jozsa 算法、Grover 算法等。

QuEST<sup>[18]</sup>是牛津大学 QTechTheory 团队推出的 C++量子模拟器, 在模拟基本量子计算的基础上, 支持 CPU、GPU 多平台运行。

Intel Quantum Simulator (IQS)<sup>[53]</sup>是 Intel 推出的 C++量子模拟器, 前身为 qHiPSTER, 其提供高性能计算功能, 可支持用户通过超级计算机进行模拟。

量浆<sup>[54]</sup>是百度推出的量子模拟器, 支持量子神经网络的搭建和训练, 提供了多个量子机器学习案例, 例如量子近似优化算法 (QAOA)、变分量子特征求解器 (VQE)、量子神经网络的贫瘠高原效应 (Barren Plateaus)、量子分类器 (Quantum Classifier) 等。

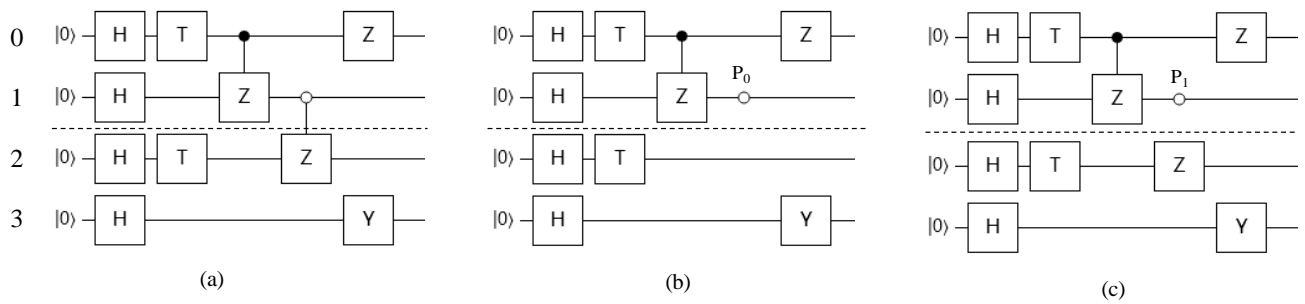
目前, 量子模拟器属于嘈杂的中型量子 (NISQ) 模拟器, 在未来几年, 提升量子比特数、优化量子模拟器、构建模拟器上层软件将成为量子模拟器研究和挑战的热点。

#### 4 基于超算集群的量子计算模拟

超级计算机集群为量子计算的模拟提供了硬件平台, 基于超算集群的量子计算模拟主要涉及以下两项影响性能的关键问题:

(1) 任务拆分。任务拆分即拆分量子线路为多个子线路, 并将其分配至不同的节点进行计算。任务拆分的目的是利用超算集群的运算性能, 来应对量子计算模拟技术带来的海量存储和计算资源开销。合适的任务拆分策略可协助提升模拟性能, 缩减节点间通信开销, 允许更多任务被拆分为子任务并行计算, 充分利用超算集群的并行处理能力。

(2) 通信优化。通信优化涉及两项优化内容:



(a)

(b)

(c)

图 3 分割纠缠量子线路的示意图

当分割线切断了一个 CZ 门时 (a)，线路便会分成 (b) 和 (c) 两种状态分枝。

Fig. 3 Example of a partitioning scheme.

When cutting through a CZ gate (a), the circuit generates two branches of states in (b) and (c).

节点间通信次数，以及单次通信传输数据量（单次通信开销）。两项优化内容之间亦可进行权衡。不合理的通信策略可能造成节点间带宽拥堵、通信延时增加、节点数据饥饿等问题。因此进行通信优化是提升超算集群运算性能的关键。

下文分别阐述了任务拆分和通信优化的内容及代表性工作。

#### 4.1 任务拆分

量子线路拆分技术已在工作中被证明了是缩小问题规模和利用多设备并行计算的可行和有效手段<sup>[10,49,50,51]</sup>。该技术可把量子线路拆分为多个子线路，缩小任务规模，使计算机能在有限的计算和存储资源下完成某个子任务，利用超算集群并行计算能力，能够比不拆分时完成同等线路大小的模拟任务<sup>[56]</sup>。

现有的量子线路拆分方法通常采用分割多线路量子门操作的方式来拆分。拆分一个多线路门操作可生成多个小规模子线路。将子线路分别执行后的结果进行合并，可得到原线路的执行结果。

文献<sup>[33]</sup>为了能够在经典计算机上模拟更多量子比特，提出了一种“slice”的方法，该方法将原来可能存在纠缠的整个量子线路切割成两个或多个互不纠缠的子线路，并按照这种方式在 IBM Blue

Gene/Q 超级计算机上完成了对 49 个量子比特、线路深度为 27 的量子线路以及 56 个量子比特、线路深度为 23 的量子线路的模拟，提升了经典计算机所能够模拟的量子程序的规模。

如果在分割时把线路分割成不同的子部分而分割线遇到了 CZ 门 (图 3a)，那么则必须考虑控制线路对目标线路所有可能的影响。当控制线路 1 在当前状态是 $|0\rangle$ 时，CZ 门对目标线路 2 不产生影响 (图 3b)；当控制线路处于 $|1\rangle$ 状态时目标线路则要经过一个 Z 门操作。这样，为了剥离一个 CZ 门的纠缠，系统状态在此处便产生了分枝，而且使得子线路的个数翻倍。

假如一个系统在分割前有  $N$  条线路，全概率幅模拟则需要的存储空间为  $2^N$ 。应用分割法后产生了 2 个各有  $N/2$  条线路的系统，这个系统的储存空间大概等同于  $2 \times 2^{N/2}$  个量子比特。如果在分割时分割了  $n$  个 CZ 门，则最终的空间需求为  $2 \times 2^{N/2} \times 2^n = 2^{(N/2+n+1)}$ 。对于有大量线路的系统，层数较少的模拟的确可以大大减少内存的需求。但随着层数的增加，所切割的 CZ 门数量也会相应地增加，从而降低使用此方法的优势。最终当  $n > N/2 - 1$  时，该方法对内存的需求比不分割还要大，从而丧失了

应用价值。

根据上述方法，文献<sup>[35]</sup>把 6×7、7×8、8×8 的量子线路分割成尺寸大致相等的 2、3 和 4 部分，并对这些线路进行了多达 36 层的模拟。为了衡量线路模拟的困难程度，其定义了一个叫 Complexity in qubit 的物理量；该量实际上等同于需要同样内存大小的未拆分线路的量子比特数，如图 4。通过理论分析可发现，随着深度的增加，线路复杂度呈单调增长，因为线路会遇到越来越多的 CZ 门。当所有深度的 CZ 门总量接近线路大小的一半时（见上段），该方法就达到了一个临界深度。使用该方法进行超过临界深度的模拟会比不使用更加困难。当线路大小增加时，临界深度也相应地增加。因此该方法适用于大尺寸少深度的线路。

文献<sup>[8]</sup>对分割法进行了改进，提出了一种名为“implicit decomposition”的分割算法<sup>[57]</sup>。此算法把线路分成不固定但有类似大小的三部分，分割线会随着线路深度逐渐变化。图 5 对有 4 个量子比特的线路展示了 implicit decomposition 的分割情况。

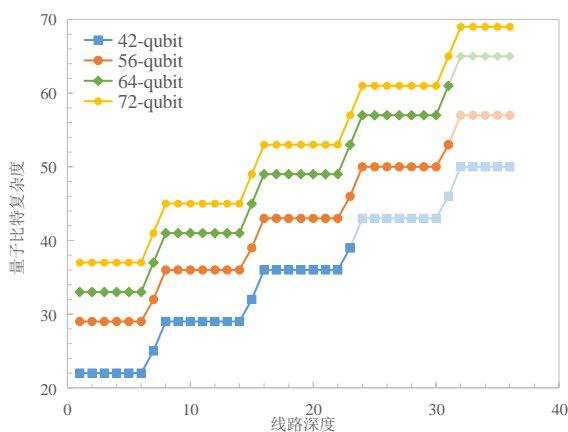


图 4 CZ 门被分割后产生的状态分支<sup>[35]</sup>。

每分割一个 CZ 门，系统的子线路个数则翻倍

Fig. 4 Simulation of random circuits with 42 to 72 qubits by the slicing method. The lines with light colors represent the depth at which the slicing method use more space than direct simulation

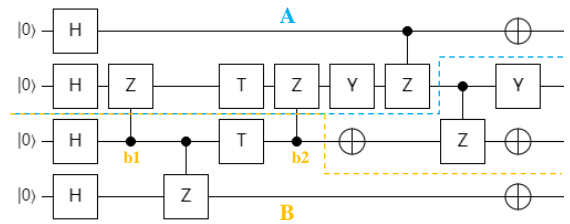


图 5 一个含有 4 个量子比特的线路分割成三部分的示意图<sup>[57]</sup>  
Fig. 5 Illustration of a 4-qubit circuit sliced into three sections

A 部分有两条线路，同时也分割了两个 CZ 门（门的状态由线路 2 上的 b1 和 b2 决定），因此 A 部分完整的状态需要  $2^2 \times 2^2 = 16$  个概率幅来表示。B 部分本来也同样需要 16 个概率幅来表示，但由于线路 2 在经过位置 b2 后结束，因此原作者认为 l2 的状态必须和线路 2 的最终状态相一致，所以 B 部分的状态向量可以减少一半的概率幅数。

根据此算法，该团对在“太湖之光”上模拟了 7×7 个量子比特的 39 层随机线路，此规模已超过了在 Blue Gene/Q 机器上同等量子比特数下的 27 层模拟，达到了国际领先水平。

线路拆分的方式侧重点在于模拟更大的、拥有更多的相互纠缠量子比特的线路，其可以达到的深度受多量子门总数量的影响。如果一个线路含有较低密度的多量子门，那么该系统中的各线路的纠缠程度就会较低，因此也可以模拟更大深度的量子线路。

#### 4.2 节点数据传输优化

在超算集群上进行量子计算模拟过程中，节点间和存储之间存在数据依赖，跨节点和设备的通信无法避免，因此需要研究计算过程中的数据分布和数据传输方法以降低模拟过程的通信开销。文献<sup>[28]</sup>研究了量子算法中节点间的数据关联性，针对量子

计算中, 每个量子门仿真过程中访问的概率幅数据进行分析, 提出了一种 MPI 进程间每个量子执行前进行的数据交互策略, 从而降低节点间数据的关联性, 实现数据本地化, 降低通信频率到多项式级。文献<sup>[46]</sup>中研究了异构众核集群环境下的大规模量子计算模拟, 提出了通过最大化本地执行的量子门数量来降低通信频率的方法, 在异构众核架构超级计算机 Cori II 上实现了 45 量子比特的量子计算机模拟实验。文献<sup>[58]</sup>为了解决不同 GPU 之间的高数据依赖性, 利用 GPU Direct 技术实现了使用 GPU 直接 P2P 传输的方案, 使用 4 块 GPU 进行了多 GPU 的量子计算模拟, 相比较 libquantum 获得了 358 倍的加速比, 并行效率达到了 0.92。

量子比特可以分为三类, 本地量子比特、节点内非本地量子比特和节点外非本地量子比特。当量子门作用在非本地量子比特上时, 会产生指数级的通信频率问题。假设每个节点的内存空间可容纳  $L$  位量子比特的所有概率幅, 则模拟  $N$  位的量子比特则需  $2^{N-L}$  个节点。若把这些节点进行二进制编码, 则量子态  $(a_{N-1} \dots a_2 a_1 a_0)$  的概率幅将存储在  $(a_{N-1} a_{N-2} \dots a_{L+1} a_L)$  号节点上的第  $(a_{L-1} K a_2 a_1 a_0)$  号位置, 其中  $a_i \in \{0,1\}$  为第  $i$  位的量子态。例如, 当  $N=5, L=2$  时, 系统使用了 8 个节点来存储所有概率幅, 节点编号由 (000) 至 (111)。每个节点存储 4 个概率幅, 编号由 (00) 至 (11)。那么, 量子态 (11010) 的概率幅就存储在第 (110) 号节点中的第 3 个地址上。

由于所有概率幅分布储存在各节点上, 一些门操作可能需要进行节点间通信。以单门 H 为例, 对

第  $i$  条线路的单门操作需要更改量子态概率幅  $(K a_i K)$ , 此操作需要读取和更新量子态系数  $(K 0_i K)$  和  $(K 1_i K)$ 。当  $i < L$  时, 这两组概率幅存放在同一节点的内存中, 门操作可在节点内完成而无需进行跨节点通信。所有  $2^{N-L}$  个节点可同时并行地完成  $2^{L-1}$  个本地读写和计算操作 ( $2^{L-1}$  个本地操作还可分配给节点内各核间并行)。因此, 这前  $L$  位量子比特称为本地 (local) 量子比特。当对第  $j$  条线路进行单门操作且  $j > L$  时, 第  $(a_{N-1} K 0_j K a_L)$  和  $(a_{N-1} K 1_j K a_L)$  号节点上的概率幅需要互相交换信息, 此时系统需要  $2^{N-1}$  次 MPI 通信。第  $L$  至  $N-L$  位量子比特称为全局 (global, 或“非本地”) 量子比特。跨节点通信的延时要比节点内通信大很多, 而且当通信过多时容易产生网络阻塞, 所以本地量子位上的门操作要比全局量子位操作有更小的时间开销。

假如现在要对第 0 和 1 位量子做一个 CZ 门操作  $CZ^{0,1}$ , 即把所有的  $A=(***01)$  的量子态与  $B=(***11)$  相交换。一个\*号代表一位量子态, 为表达简洁把  $a_4 a_3 a_2$  化简为\*\*\*。由于第 0 和 1 位是本地量子比特, 这个  $CZ^{0,1}$  操作可以在各节点内快速交换  $(***01)$  和  $(***11)$  的数据来完成 (如图 6)。

另一方面, 假如要对第 3 和 4 位的全局量子比特进行  $CZ^{3,4}$  操作, 那么概率幅  $C=(01***)$  和  $D=(11***)$  互换的操作就需要跨节点进行 (此处\*\*\*为  $a_2 a_1 a_0$ ), 并要靠 MPI 实行 8 次数据交换 (如图 7 所示), 运算时间要比  $CZ^{0,1}$  长得多。图 7 中当进行  $CZ^{0,1}$  操作时, 节点内进行 1 次信息交换, 交换位置由黑曲线标出。当进行  $CZ^{3,4}$  操作时, 节点间进行 8 次信息交换, 交换位置由红直线标出。

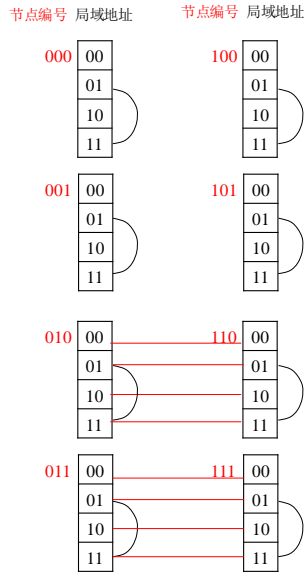


图 6 一个使用 8 个节点储存 5 个量子比特的系统示意图  
Fig. 6 Illustration of a system with eight nodes storing five qubits

$CZ^{i,j}$  是一种比较简单的操作，因为其数学表达矩阵只有 4 个非零系数。对于一些比较复杂的门操作，例如 XX Ising 门，所有存储 (00\*\*\* ) 量子态概率幅的节点需要和存储 (11\*\*\* ) 的节点通信，存储 (01\*\*\* ) 和 (10\*\*\* ) 的节点也互相通信，共 16 次。如果不对算法进行优化，全局量子位的计算将会变得很缓慢。

利用数据本地化的原理，可以把需要模拟的量子位从全局位置移至本地位置，移动后在节点内的内存中进行运算，可大大节省时间开销<sup>[58]</sup>。假设节点内存可容纳的量子位数  $L$  不小于需要模拟的位数，以 3, 4 位的 XX Ising 门为例，把第 3、4 位量子比特的数据与第 0、1 位互换，即：

- (00\*01) 与 (01\*00) 互换
- (00\*10) 与 (10\*00) 互换
- (00\*11) 与 (11\*00) 互换
- (01\*10) 与 (10\*01) 互换
- (01\*11) 与 (11\*01) 互换

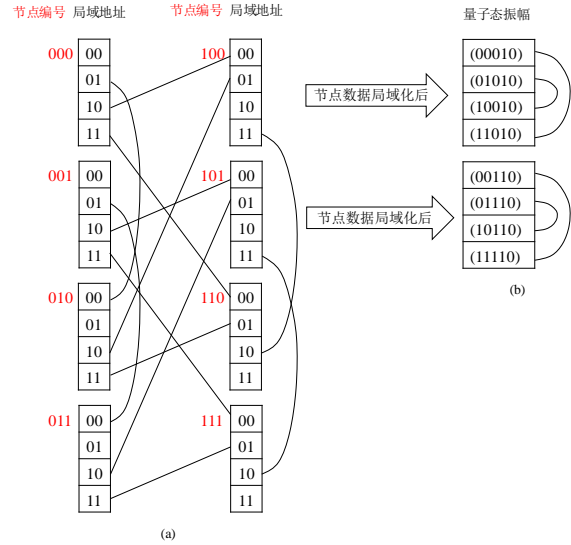


图 7 数据本地化示意图  
Fig. 7 Illustration of data localization

(10\*11) 与 (11\*10) 互换

以上 6 组互换中的每一组实际包含了 2 次互换，共 12 次，少于优化前所需的 16 次数据通信量。交换后，所有 (\*\* $a_2a_1a_0$ ) 数据都存储在编号为 (\*\* $a_1a_0a_2$ ) 的节点内，各节点可以同时并行地执行门操作，如图 7。图 7 (a) 中经过 12 次数据交换后，第 3、4 位量子比特的概率幅 (\*\* $a_2a_1a_0$ ) 将会存储在同一个节点 ( $a_1a_0a_2$ ) 内，图 7 (b) 为经交换后节点 (100) 和 (101) 所存数据对应的量子态概率幅。操作完毕后，再根据下一个门所模拟的线路编号进行下一次的数据本地化，以此类推。

以此系统 ( $N=5, L=2$ ) 为例，该方法能够把任意复杂的双量子门操作降低至 12 次数据互换。对于一些非常简单的门操作，直接操作可能不需要 12 次数据互换。这需要算法在模拟中作具体判断。该方法还可类推用来优化多量子门操作，详情不在此文展开。

### 5 总结

本文总结了量子计算机的模拟方法，模拟全部

量子态的概率幅的方式所需内存，随模拟的量子位数指数级增长，可模拟的量子位数受到内存大小的制约。部分概率幅模拟方法的效率高，相比于全概率幅模拟方法能够模拟具有更多量子位的浅深度的量子线路。其中，张量网络收缩方法可以将全局问题转化为局部的张量处理问题，从而有效地降低了时间复杂度和空间复杂度。对于量子计算模拟器在内存使用和运算性能方面的优化算法，分割法通过把线路分成若干个更小的区域，可以把占用的空间降低一半量子比特数的数量级。数据本地化则通过把全局量子位上的数据转移到本地量子位上，让模拟器能够在节点内用更少的时间并行完成门操作计算。特别是对于比较复杂的门操作，本地化的数据转移量也比直接模拟的数据转移量更少。为了从量子计算优越性发展到大规模的量子计算，量子纠错的技术也同样需要发展。

量子计算模拟在算法和工程上仍有很大的优化空间。在算法方面，张量网络缩并的顺序极大地影响计算效率，未来可探索寻找更优缩并顺序算法；在工程方面，需要应用张量运算来充分发挥硬件性能，进一步降低线路模拟所需的时间。这些方面都可以作为未来量子模拟器计算的主要研究内容。这些方面都可以作为未来大规模量子计算的主要研究内容，此外可扩展性和如何容错同样是量子计算研究的关键。

#### 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] BABBUSH R, MCCLEAN J, WECKER D, et al.

- Chemical basis of Trotter-Suzuki errors in quantum chemistry simulation[J]. *Physical Review A*, 2015, 91(2): 022311.
- [3] BAUER B, WECKER D, MILLIS A J, et al. Hybrid quantum-classical approach to correlated materials[J]. *Physical Review X*, 2016, 6(3): 031045.
- [4] BIAMONTE J, WITTEK P, PANCOTTI N, et al. Quantum machine learning[J]. *Nature*, 2017, 549(7671): 195-202.
- [5] DOU Xinglei, LIU Lei. A New Qubits Mapping Mechanism for Multi-programming Quantum Computing[C]// International Conference on Parallel Architectures and Compilation Techniques, ACM, 2020: 349-350.
- [6] LIU Lei, DOU Xinglei. QuCloud: A New Qubit Mapping Mechanism for Multi-programming Quantum Computing in Cloud Environment[C]// International Symposium on High Performance Computer Architecture, IEEE, 2021: 167-178.
- [7] PRESKILL J. Quantum Computing in the NISQ era and beyond[J]. *Quantum*, 2018, 2: 79.
- [8] LI Gushu, DING Y, XIE Yuan. Tackling the qubit mapping problem for NISQ-era quantum devices[C]//Proceedings of the Twenty-Fourth International Conference on Architectural Support for Programming Languages and Operating Systems. 2019: 1001-1014.
- [9] ARUTE F, ARYA K, BABBUSH R, et al. Quantum supremacy using a programmable superconducting processor[J]. *Nature*, 2019, 574(7779): 505-510.
- [10] HUANG Cupjin, ZHANG Fang, NEWMAN M, et al. Classical simulation of quantum supremacy circuits[J]. arXiv preprint arXiv:2005.06787, 2020.
- [11] CHEN Zhaoyun, ZHOU Qi, XUE Cheng, et al. 64-qubit quantum circuit simulation[J]. *Science Bulletin*, 2018, 63(15): 964-971.
- [12] ATAIDES J P B, TUCKETT D K, BARTLETT S D, et al. The XZZX surface code[J]. *Nature communications*, 2021, 12(1): 1-12.
- [13] ALMUDEVER C G, LAO L, FU X, et al. The engineering challenges in quantum computing[C]//Design, Automation & Test in Europe Conference & Exhibition (DATE), IEEE, 2017: 836-845.

- [14] SUN Xiaoming. A survey on quantum computing[J]. SCIENTIA SINICA Informationis, 2016, 46(8): 982-1002. (孙晓明. 量子计算若干前沿问题综述[J]. 中国科学: 信息科学, 2016 (008): 982-1002.)
- [15] FAN Hang. Quantum computation and quantum simulation[J]. Acta Physica Sinica, 2019, 67(12): 120301-120301. (范桁. 量子计算与量子模拟[J]. 物理学报, 2019, 67(12): 120301-120301.)
- [16] STEIGER D S, HANER T, TROYER M. ProjectQ: an open source software framework for quantum computing[J]. Quantum, 2018, 2: 49.
- [17] SMELYANSKIY M, SAWAYA N P D, ASPURU G A. qHiPSTER: The quantum high performance software testing environment[J]. arXiv preprint arXiv:1601.07195, 2016.
- [18] JONES T, BROWN A, BUSH I, et al. QuEST and high performance simulation of quantum computers[J]. Scientific reports, 2019, 9(1): 1-11.
- [19] NIELSEN M A, CHUANG I. Quantum computation and quantum information[M]// Cambridge, UK: Cambridge University Press, 2002.
- [20] BOIXO S, ISAKOV S V, SMELYANSKIY V N, et al. Characterizing quantum supremacy in near-term devices[J]. Nature Physics, 2018, 14(6): 595-600.
- [21] BREMNER M J, JOZSA R, SHEPHERD D J. Classical simulation of commuting quantum computations implies collapse of the polynomial hierarchy[J]. Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences, 2011, 467(2126): 459-472.
- [22] AARONSON S, ARKHIPOV A. The computational complexity of linear optics[C]//Proceedings of the forty-third annual ACM symposium on Theory of computing. 2011: 333-342.
- [23] WU Bujiao, CHENG Bin, JIA Fei, et al. Speedup in classical simulation of Gaussian boson sampling[J]. Science Bulletin, 2020, 65(10): 832-841.
- [24] BOULAND A, FEFFERMAN B, NIRKHE C, et al. On the complexity and verification of quantum random circuit sampling[J]. Nature Physics, 2019, 15(2): 159-163.
- [25] ZLOKAPA A, BOIXO S, LIDAR D. Boundaries of quantum supremacy via random circuit sampling[J]. arXiv preprint arXiv:2005.02464, 2020.
- [26] CORRIGAN G H, WU D J, BONEH D. Quantum Operating Systems[C]// Proceedings of the 16th Workshop on Hot Topics in Operating Systems, 2017: 76-81.
- [27] AARONSON S, CHEN Lijie. Complexity-theoretic foundations of quantum supremacy experiments[J]. arXiv preprint arXiv:1612.05903, 2016.
- [28] FEYNMAN R P. Simulating physics with computers[J]. Int. J. Theor. Phys, 1982, 21(6/7).
- [29] WU Xinchuan, DI Sheng, DASGUPTA E M, et al. Full-state quantum circuit simulation by using data compression[C]//Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis. 2019: 1-24.
- [30] DE RAEDT H, JIN F, WILLSCH D, et al. Massively parallel quantum computer simulator, eleven years later[J]. Computer Physics Communications, 2019, 237: 47-61.
- [31] MCCASKEY A, DUMITRESCU E, CHEN M, et al. Validating quantum-classical programming models with tensor network simulations[J]. PloS one, 2018, 13(12): e0206704.
- [32] GUO Chu, ZHAO Youwei, HUANG Heliang. Verifying Random Quantum Circuits with Arbitrary Geometry Using Tensor Network States Algorithm[J]. Physical Review Letters, 2021, 126(7): 070502
- [33] PEDNAULT E, GUNNELS J A, NANNICINI G, et al. Pareto-Efficient Quantum Circuit Simulation Using Tensor Contraction Deferral[J]. arXiv preprint arXiv:1710.05867, 2017.
- [34] BOIXO S, ISAKOV S V, SMELYANSKIY V N, et al. Simulation of low-depth quantum circuits as complex undirected graphical models[J]. arXiv preprint arXiv:1712.05384, 2017.
- [35] CHEN Jianxin, ZHANG Fang, HUANG Cupjin, et al. Classical simulation of intermediate-size quantum circuits[J]. arXiv preprint arXiv:1805.01450, 2018.
- [36] PAN Feng, ZHANG Pan. Simulating the Sycamore quantum supremacy circuits[J]. arXiv preprint arXiv:2103.03074, 2021.
- [37] WU Xiangyan, XU Yanling, YU Yafei, et al. Fault-tolerantly implementing dense rotation operations based on non-stabilizer states[J]. Acta Physica Sinica,



- 2014, 63(22).
- [38] PROCTOR T J, CARIGNAN D A, RUDINGER K, et al. Direct randomized benchmarking for multiqubit devices[J]. Physical review letters, 2019, 123(3): 030503.
- [39] BLUME K R, GAMBLE J K, NIELSEN E, et al. Demonstration of qubit operations below a rigorous fault tolerance threshold with gate set tomography[J]. Nature communications, 2017, 8(1): 1-13.
- [40] WANG Z, CHEN Y, SONG Z, et al. Scalable evaluation of quantum-circuit error loss using Clifford sampling[J]. Physical Review Letters, 2021, 126(8): 080501.
- [41] AARONSON S, GOTTESMAN D. Improved simulation of stabilizer circuits[J]. Physical Review A, 2004, 70(5): 052328.
- [42] BRAVYI S, GOSSET D. Improved classical simulation of quantum circuits dominated by Clifford gates[J]. Physical review letters, 2016, 116(25): 250501.
- [43] BU K, KOH D E. Efficient classical simulation of Clifford circuits with nonstabilizer input states[J]. Physical review letters, 2019, 123(17): 170502.
- [44] HE Xianya, Guan Zhijin, Ding Fei. The Mapping and Optimization Method of Quantum Circuits for Clifford+ T Gate[J]. Journal of Applied Mathematics and Physics, 2019, 7(11): 2796-2810.
- [45] GAO Xun, Duan Luming. Efficient classical simulation of noisy quantum computation[J]. arXiv preprint arXiv:1810.03176, 2018.
- [46] CHEN YiTing, COLLIN Farquhar, ROBERT M P. Low-rank density-matrix evolution for noisy quantum circuits[J]. npj Quantum Information, 2021, 7(1): 1-12.
- [47] CROSS A. The IBM Q experience and QISKit open-source quantum computing software[C]//APS March Meeting Abstracts, 2018: L58. 003.
- [48] PEDNAULT E, GUNNELS J A, NANNICINI G, et al. Leveraging secondary storage to simulate deep 54-qubit sycamore circuits[J]. arXiv preprint arXiv:1910.09534, 2019.
- [49] SICILIA M A, SANCHEZ A S, MORA C M, et al. On the Source Code Structure of Quantum Code: Insights from Q# and QDK[C]// International Conference on the Quality of Information and Communications Technology. Springer, Cham, 2020: 292-299.
- [50] HUAWEI. <https://hiq.huaweicloud.com/en/>
- [51] GOOGLE <https://quantumai.google/cirq>
- [52] ORIGIN QUANTUM. <http://www.originqc.com.cn/QPanda/download.html>
- [53] GUERRESCHI G G, HOGABOAM J, BARUFFA F, et al. Intel Quantum Simulator: A cloud-ready high-performance simulator of quantum circuits[J]. Quantum Science and Technology, 2020, 5(3): 034007.
- [54] BAIDU. <https://qml.baidu.com>
- [55] MARKOV I L, SHI Yaoyun. Simulating quantum computation by contracting tensor networks[J]. SIAM Journal on Computing, 2008, 38(3): 963-981.
- [56] TABAKIN F, BRUNO J D. Qcmpi: A Parallel Environment for Quantum Computing[J]. Computer Physics Communications, 2009, 180(6):948-964.
- [57] Li Riling, Wu Bujiao, Ying Mingsheng, et al. Quantum supremacy circuit simulation on Sunway TaihuLight[J]. IEEE Transactions on Parallel and Distributed Systems, 2019, 31(4): 805-816.
- [58] ZHANG Pei, YUAN Jiabin, LU Xianwen. Quantum computer simulation on multi-GPU incorporating data locality[C]//International Conference on Algorithms and Architectures for Parallel Processing. Springer, Cham, 2015: 241-256.

请在文章最后给出前 3 位作者的联系方式(长期有效的手机、电话及 E-mail)。

喻志超 13051915050 yuzc@nscsz.cn

李扬中 13544510045 liyz@nscsz.cn

通讯作者: 刘磊 15810455459 lei.liu@zoho.com