

量子模拟器优化综述^①

彭世昕¹, 张为华^{1,2}

¹(复旦大学 计算机科学技术学院, 上海 200438)

²(复旦大学 上海市数据科学重点实验室, 上海 200438)

通信作者: 张为华, E-mail: zhangweihua@fudan.edu.cn



摘要: 近年来, 不断发展的量子计算已成为众人关注的焦点. 然而, 量子硬件存在稀缺性和噪声等问题, 这使得研究量子算法、验证量子芯片等行为都依赖运行在经典计算机上的量子模拟器. 本文讨论了不同量子模拟器使用的主要模拟方法, 并讨论了主流的全振幅状态向量模拟器和基于张量网络的量子模拟器的各种优化. 最后, 我们总结了量子模拟器的现状和未来发展方向.

关键词: 量子计算; 量子电路模拟; 量子模拟器; 高性能计算

引用格式: 彭世昕, 张为华. 量子模拟器优化综述. 计算机系统应用, 2024, 33(6): 16-27. <http://www.c-s-a.org.cn/1003-3254/9559.html>

Overview on Quantum Simulator Optimization

PENG Shi-Xin¹, ZHANG Wei-Hua^{1,2}

¹(School of Computer Science, Fudan University, Shanghai 200438, China)

²(Shanghai Key Laboratory of Data Science, Fudan University, Shanghai 200438, China)

Abstract: In recent years, the rapidly evolving quantum computing has become the focus of attention. However, quantum hardware suffers from scarcity and noise, which makes the study of quantum algorithms and the verification of quantum chips rely on quantum simulators running on classical computers. In this study, the main simulation methods used by different quantum simulators are discussed, and various optimizations of mainstream full-amplitude state vector simulators and tensor network-based quantum simulators are explored. Finally, the current status and future directions of quantum simulators are summarized.

Key words: quantum computing; quantum circuit simulation; quantum simulator; high performance computing

1 引言

量子计算在因数分解^[1]和搜索问题^[2]已被证明相对于传统计算具有显著优势, 因此, 相关研究吸引了全球科学界的极大关注. 尽管如此, 量子硬件的稀缺性和构建实用量子计算机的技术挑战仍然是当前科学界面临的重大难题. 在这种背景下, 量子模拟器作为一种关键工具, 已经在开发和测试量子算法以及模拟量子处理器的实现方面发挥了重要作用. 特别是, 一个高效的量子电路模拟器对当前中尺度嘈杂量子时代 (noisy intermediate-scale quantum, NISQ)^[3,4]. 量子模拟器依赖

于经典计算资源, 能够提供更高的保真度, 从而为量子计算机的设计和发展提供关键的指导和参考. 然而, 随着量子电路中量子位数量的增加, 其对应的状态空间呈指数级增长, 这就要求量子模拟器作为一种运行在经典计算机上的软件, 必须不断进行优化以提升性能, 从而提高模拟的上限.

本文旨在深入探讨量子计算的基本原理, 并简要介绍其关键概念. 随后, 本文将重点分析量子模拟器中采用的主要模拟方法, 包括全振幅状态向量模拟和张量网络方法模拟, 并详细讨论这些方法在量子模拟器

① 收稿时间: 2023-12-14; 修改时间: 2024-01-17, 2024-02-23; 采用时间: 2024-03-04; csa 在线出版时间: 2024-04-30

CNKI 网络首发时间: 2024-05-07

中的算法级和系统级优化. 最终, 本文将对量子模拟器的未来发展进行预测和探讨. 深入研究量子模拟器中的模拟技术, 有助于进一步理解量子模拟的独特性, 为该领域的研究人员和从业者提供宝贵的见解和指导, 以便更有效地优化量子算法和提升系统性能, 进而为解决更复杂的问题创造可能性.

2 量子基本背景

本节主要介绍与量子模拟器相关的量子计算背景知识, 涵盖了量子比特、量子门以及量子电路的基础概念. 此外, 本节还展示了量子模拟器中常用于基准测试的重要算法和电路.

2.1 量子比特

经典比特的状态可以表征为 0 或 1, 而量子比特的状态 (即量子比特) 被视为 $|0\rangle$ 和 $|1\rangle$ 的叠加, 其状态可以表征为 $\alpha|0\rangle+\beta|1\rangle$, 其中 α 和 β 的模长平方和为 1. 量子比特与经典比特的区别在于后者存在 $|0\rangle$ 和 $|1\rangle$ 以外的状态, 是 $|0\rangle$ 和 $|1\rangle$ 的状态的线性叠加, 即叠加态.

2.2 量子门与量子电路

与经典计算机基于明确定义的布尔逻辑模型相比, 量子计算可以采用不同的实现方式, 包括绝热量子计算^[5]、簇态量子计算^[6]和拓扑量子计算^[7]. 目前, 最常用的量子计算模型是由 Deutsch 提出的量子电路模型^[8], 因为它与经典电路具有类似性.

量子电路是描述量子态演化的有效工具, 在量子电路中, 通过反复应用量子门来修改量子态. 量子门是量子电路的基本单元, 类似于传统逻辑门在数字电路中的作用. 与大多数传统逻辑门不同, 量子逻辑门必须是可逆的^[9], 它们可以用酉矩阵来表示.

量子电路的宽度指的是电路中包含的量子比特数 (又称量子位数), 通常情况下, 电路包含的信息与电路的宽度呈指数级关系. 量子电路的深度则定义为执行电路所需的最大连续时间步长^[10]. 其中时间步长是对状态向量上的最大门数目的单个应用, 以便可以并行和同时执行所有应用的门.

2.3 常见量子算法及电路

在量子计算的发展历程中提出过许多重要的量子算法和电路以充分的证明量子计算的优越性. 这些重要的量子算法及有关电路通常在评估量子模拟器的性能时被使用与讨论. 接下来, 我们将介绍比较重要的几个量子算法.

量子傅里叶变换^[11]: 傅里叶变换作为一种通用算法, 在信号处理、通信、图像压缩、密码学等领域都有重要的应用价值. 与传统的傅里叶变换算法相比, 量子傅里叶变换 (quantum Fourier transform, QFT) 具有更高的执行效率, 所以它是许多量子算法的核心组成部分之一. QFT 是应用最广泛的量子算法, 且是 Shor 算法中的重要组成部分^[1], 同样也是用于求解线性系统的 HHL 算法的重要单元^[12]. 其中常用的变体还有 QFT 的翻转电路 IQFT (inverse of QFT circuit, IQFT), 近似 QFT 电路 (approximate quantum Fourier transform circuit, AQFT)^[13].

量子搜索算法^[2]: 与经典算法相比, 量子 Grover 搜索算法在非结构化数据库搜索中提供了多项式的加速. 其中最重要的部分是幅度估计部分^[14]. Grover 算法本身是一个迭代的过程, 主要思路是从初始状态出发, 重复进行多次变换, 让系统越来越接近目标状态, 最后进行测量, 就能以很高的概率得到正确结果.

量子近似优化算法为代表的变分量子算法: 量子近似优化算法 (quantum approximate optimization algorithm, QAOA)^[15]是用于解决优化问题的量子算法. 对于组合优化问题, 量子近似优化算法短暂地具有比任何已知的多项式时间经典算法看起来具有更好的效果. QAOA 算法是变分量子算法 (variational quantum algorithm, VQA)^[16]的一种, 其使用经典优化器来训练参数化量子电路, 是 NISQ 时代有能力发挥量子优势的量子算法, VQA 广泛地应用在量子化学^[17], 组合优化^[15], 量子机器学习^[18-20]等各个领域.

量子优越性电路: Google 在 2019 年提出实现了量子优越性 (quantum supremacy, 又称量子霸权), 其中最重要的内容是随机电路采样 (random circuit sampling, RCS)^[3,21-23], 即从随机量子电路 (random quantum circuit, RQC) 生成的输出分布中近似采样量子位串, 这可以被认为是量子计算机的“hello world”程序. 这个任务也是量子复杂性理论中最强有力的理论支持^[21], 用于经典和量子计算之间的指数分离. 也正是这个“hello world”程序在 Google 的工作中被用来证明量子优越性^[22]. 而这个所谓的量子优越性电路即随机量子电路被作为目前事实上量子模拟器最重要的基准测试出现^[24]. 由于其相对较高的电路复杂性导致的对其进行经典模拟的困难性, 许多经典模拟的目标也正是挑战对随机量子电路模拟的上限, 同时意味着对量子优越

性任务本身进行挑战^[24-28]。

3 量子模拟器主要模拟方法

一般来说,要模拟 n 个量子位和深度 d 的量子电路,有两种主要的模拟策略.第1种策略在内存中保持完整的状态向量,也被称为状态向量全振幅方法^[21].第2种策略是将量子电路表示为张量网络,然后对张量网络进行收缩^[21]。

对于全振幅风格的模拟方式来说,经典计算机的内存限制了模拟的量子比特数上限,但这种方法能够模拟任意深度的电路,qHiPSTER被用来模拟多达42量子位的量子电路^[29].在Google提出量子霸权电路^[22]之后,使用0.5 PB (petabytes,千万亿字节)在Cori II超级计算机上模拟了45量子位的电路^[30],这都是使用全振幅方式的量子模拟器.一些研究提出使用张量网络来模拟低深度的量子霸权电路^[31],张量网络收缩方法也是量子电路模拟的另一重要方法。

当然,模拟量子电路也存在其他的思路.比如,Gottesman-Knill定理^[32]表明,只由Clifford门组成的电路可以在多项式时间内被有效地模拟,因此存在着针对Clifford门集的电路的模拟技术^[33-37].但是这种方法比较受限于门的种类,非Clifford门的增多会使得模拟难度呈指数级变大,所以其局限性相对较高.Clifford门集及功能描述见表1。

表1 Clifford门集及功能描述

量子门名称	功能描述
Hadamard	创建叠加态,实现量子比特的均匀叠加
Phase(S)	为量子位添加相位,特别是将 $ 1\rangle$ 乘以 i
CNOT	条件逻辑门,根据控制比特的状态翻转目标比特

基于状态向量的全振幅模拟方法和基于张量网络收缩的模拟方式是更加主流的模拟方法.接下来本节主要介绍基于状态向量的全振幅模拟方法和张量网络收缩方法的原理。

3.1 状态向量全振幅方法模拟

薛定谔方程是描述量子态演化的公式,薛定谔风格状态向量全振幅方法模拟量子电路的思想是存储所有的量子态信息,在整个计算过程中保存量子态的完整幅值,并根据每个量子门更新对应的幅值状态.在传统计算机中模拟薛定谔方程采用状态向量 (state vector, 又可称态矢) 表示的方式,由于波函数随时间演化整体改变,因此在内存中需要保存波函数的全部的状态向量,且每一步演化过程均需更新所有的状态向

量.由于具有 n 量子位的量子系统的状态向量大小为 2^n ,如果用双精度复数存储的话,所需要的内存大小为 2^{n+4} 字节.这种方法对量子电路中的门和拓扑结构方面没有限制,但存储完整量子态所需的内存会是指数级别的,而且还需要考虑到更新全量子态的通信所带来的开销.因此,全振幅方式模拟量子位较多的电路就需要依靠超级计算机的能力或者分布式计算等手段.一般来说,超过50个量子位的量子电路使用这种方式就很难进行模拟。

这种基于状态向量的全振幅方法存在一个基本的优化思想,即不需要完整的 2^n 级别的矩阵乘法,因为量子门的本身包含的信息量通常就是一个 2×2 的矩阵,所以并不需要在每次状态向量的更新过程中都进行指数级别的矩阵运算,只需要将状态向量信息中振幅按下标进行配对,进行逐对更新.双量子门的操作关注的状态向量的振幅也类似,将4个振幅分成一组,然后按组进行更新即可。

3.2 张量网络收缩方法模拟

张量网络收缩方法的主要思想是基于这样一个事实,即任何给定的量子电路都可以表示为张量网络,其中单量子门可以用二阶张量来表示,双量子门用四阶张量来表示^[38].通常,收缩这种网络的计算和内存开销至少与开放索引的数量(分别对应于输入状态和输出状态)成指数关系.因此,对于足够大的电路,张量网络的收缩也是不切实际的.张量网络收缩过程中计算和内存的开销是由收缩过程中的最大张量控制的。

通过将对应的量子电路表述为对应的张量网络,原有复杂的量子电路计算过程中就转换为一个张量网络收缩问题.值得注意的是,不同的收缩顺序所对应的乘法次数可能会有显著区别。

状态向量全振幅方法和张量网络收缩方法的主要特点如表2所示,它们基于的主要算法,适用的电路,性能特点以及实现的主要目标都有所区别。

表2 状态向量全振幅方法和张量网络收缩方法比较

对比项	状态向量全振幅方法	张量网络收缩方法
算法	薛定谔风格算法	张量网络收缩
适用电路	宽度低但深度高	宽度高但深度低
性能特点	内存密集型	计算密集型
目标	虚拟量子计算机	量子芯片验证

4 全振幅量子模拟器优化

使用薛定谔风格下的全振幅模拟能够模拟任意深

度的电路,并且产生精确的最终概率分布.因此,很容易使用全振幅模拟来模拟大型真实的量子计算机.其次,许多量子算法,如纠错码和量子隐形传态,需要状态向量的中间结果^[39-41].另外,对于电路深度较深的量子电路,全振幅模拟比张量网络收缩方法下的模拟更有效.并且有些量子电路如变量子本征求解器^[17]的电路会随着电路深度增加表现出更好的性能,所以没有拓扑和深度的限制的状态向量模拟方式很重要.

然而,同样值得注意的是,全振幅方式的量子模拟器的规模受到物理内存限制,因此,针对其进行计算和存储优化以提升性能是极其重要的.

基于薛定谔风格的全振幅量子模拟器种类很多,如IBM公司的Qiskit-aer^[42],Google公司的Cirq^[43],Q#^[44],TensorFlow Quantum^[45],QuEST^[46],IntelQS^[29,47],QCGPU^[48],Qibo^[49],Yao.jl^[50],Qulacs^[51],QX^[52],ProjectQ^[53],LIQUI^[54],Quantum++^[55],OpenQL^[56]等.

接下来将集中讨论应用于全振幅量子模拟器的算法级别和系统级别优化,其中包括一些通用优化策略,即这些优化不特定于任何特定类型的模拟器.

4.1 门融合方法

门融合方法可以通过合并量子门操作以充分利用硬件性能.具体来说,量子模拟器可以在应用相邻量子位之前,通过重新排列它们的操作顺序来缓存操作,这一技术被称为门融合.在某些情况下,特别是对于大的量子位,通过将它们各自的幺正矩阵相乘,并将得到的矩阵乘以状态向量来融合几个门,比一个接一个地应用原始门更有效.门融合技术通常与其他硬件特性一起使用,例如结合CPU的缓存,GPU的共享内存^[29,30,57]以确保融合出的新的量子门的大小能充分适应缓存大小以达到充分利用硬件性能的效果.

4.2 电路拆分方法

Pednault等人^[58]利用CZ量子门可以被拆分两组单量子门的张量积这一特性,将模拟的流程转化为:基于可分解门,将电路分解为 2^{m+1} 个子电路,其中 m 是要分解的门的数量;分别模拟每个子电路,每个子电路的模拟方法与经典的状态向量全振幅模拟一致;组合每个子电路的最终状态,得到原始电路的最终态.从而模拟了深度为27,量子比特数为 7×7 的随机量子电路.神威太湖之光上的一项工作^[27]同样利用CZ门的对角线特性,进而提出了一种隐式分解方案来解决模拟深度为39,量子比特数 7×7 的问题.

HyQuas^[57]将量子电路按深度划分为多个子电路,并对每个子电路使用不同的模拟技术来加速模拟过程,这种电路分区方法是按照电路深度进行拆分的.

4.3 基本运算优化

通过应用位运算优化和交换操作代替复杂乘法,可以显著提高量子电路模拟的计算效率,降低门运算时间.

计算机以二进制形式在程序中存储数字,按位操作是对存储在内存中的整数的二进制操作.因此,按位操作比其他操作更高效.在量子电路模拟的过程中,程序运行的很大一部分开销花费在乘除法上,PAS^[59]通过位运算优化乘除法以降低计算开销.

QX^[52]关注到有些矩阵与一个向量相乘时,它只是置换该向量的元素.如Pauli-X门是一种常见的单量子门,它对目标量子位执行位翻转,其对应的酉矩阵是一个置换矩阵,因此该矩阵与单位矩阵的张量积仍然是一个置换矩阵.因此,昂贵的复杂乘法可以被更快的交换操作取代.QX通过这种思路优化计算,从而减少了40%的门运算时间.

4.4 向量化优化

量子模拟器中的状态向量修改可以通过向量化并行执行来加速,从而实现数据级并行和更高的指令级并行性.

向量化是在CPU架构下实现指令级并行的一种通用方法.在量子模拟器中,修改状态向量的数据并行任务可以通过单指令多数据(single instruction multiple data, SIMD)^[60]执行来加速.例如, SIMD指令,如英特尔的高级矢量扩展(advanced vector extensions, AVX)^[61],可以同时向向量寄存器中的多个操作数进行操作,以便同时修改多个振幅的值.众多模拟器已支持向量化,利用Intel的AVX2,它们可以同时处理256位的数据.假设量子状态以双精度复数表示,这意味着可以同时处理2个复数振幅.许多模拟器根据编译器的内在特性开发了专用代码,以便使用SIMD指令高效地执行复杂计算^[29,42,43,51].Rollright^[62]通过对齐内存读取结合AVX2做到了更好的数据级并行.PAS^[59]通过混合向量化为不同的量子门的任务要求合理选择不同矢量长度的指令集,同时加入FMA(fused multiply add)指令集,以实现更高的指令级并行性.

4.5 内存访问优化

多数量子算法在模拟过程中产生大量冗余的内存

写操作,通过分支预测去除这些操作和采用缓存阻塞技术来减少昂贵的数据交换,提高性能.特定算法重构和门操作的简化也有助于减少复杂计算和内存访问,实现高效模拟.

例如,在一个任务中,因为它们的值在一系列计算之后没有改变,执行计算的两个不同下标的元素值是相同的,或者都是0. PAS 模拟器^[59]选择通过分支预测的方法去除这部分操作,这将大大减少程序的内存写操作,提高程序的整体运行性能.

通过合适的数据访问来推迟 CPU 最后一级缓存 (last level cache, LLC) 与主存储器中昂贵的数据交换的技术称为缓存阻塞^[63]. 因此,当状态向量的大小超过最后一级缓存的大小时,模拟器的性能受到内存带宽的限制. 现代 CPU 有很大的 LLC, LLC 具有比内存高得多的带宽,但利用高 LLC 带宽需要重构算法,使模拟器更适合 LLC. Rollright^[62]为了避免对更重要的量子位起作用的门遭受长时间的内存跨越和频繁的缓存丢失的问题,开发了一种递归类快速傅里叶变换算法,用于模拟非对角单量子门.

相较于 Hadamard 门,相移门或泡利门的矩阵是对角线或反对角线矩阵,它们可以允许更少的复杂乘法. 此外,量子门的应用主要是由连续的复杂的乘法和加法组成, QX^[52]通过融合复杂的乘法和加法进行进一步的优化,以实现高性能和更少的内存访问.

4.6 数据精度优化

在量子模拟器中,使用混合精度方案和动态数据标度策略可有效平衡精度与效率,减少因使用低精度数据结构导致的精度损失.

由于状态向量是由复数振幅组成的,两个基本的方案是 32 位浮点数和 64 位双精度浮点数. 更低或者更高精度的类型当然也可用,但会降低可信度或者显著增加计算负荷. 选择 32 位浮点类型的选择可以提高内存占用和吞吐量以优化模拟性能. 很多模拟器都提供了使用单精度浮点复数和双精度浮点复数的选择^[42,43,46,49],但是选用 32 位浮点数不可避免地要损失精度,在 Rollright 模拟器^[62]中,考虑到许多常见量子门都带有 $\sqrt{2}$ 的系数,如果应用到所有振幅上,精度损失将会很大,为了减少这种选用较低精度的数据结构带来的精度损失, Rollright 通过保存公共系数到振幅外从而实现了较好的性能.

Liu 等人^[26]提出了一种混合精度方案,它使用单精度和半精度浮点数来模拟随机量子电路. 首先,通过分析来检测计算的不同部分的精度灵敏度. 其次,设计的自适应标度方法对数据进行动态调整,特别是对精度敏感的部分,使误差保持在与单精度计算相近的水平. 通过对张量精度范围的分析,该工作提出了一种动态的数据标度策略,有效地防止了数据下溢,在精度和效率上实现了更好的平衡.

4.7 数据压缩

为了减轻全振幅量子模拟的内存压力,研究者们采用了数据压缩、基于决策图 (decision diagram, DD) 的模拟器优化和并行化技术,有效降低了内存需求,并提高了模拟性能.

Wu 等人^[64]提出了有损数据压缩,以降低模拟大规模量子电路的内存需求. Zulehner 等人^[65]利用了量子态和量子门操作中的冗余度,以压缩格式保存状态以节省内存,从而能够模拟更多的量子位.

Q-GPU^[66]中提供了一种混合方法,通过修剪和压缩来分别处理零振幅和非零振幅从而在不损失保真度的情况下显著减少了振幅交换.

基于决策图的模拟器可以通过减少状态向量和有关操作中的数据冗余来紧凑地表示量子状态和操作. 其中 QMDD^[67]是第一个实用的基于决策图的处理程序. Zulehner 等人^[65]而后也开发了一种利用量子状态和量子操作中冗余的模拟器, Burgholzer 等人^[68]借用张量网络的收缩顺序策略来优化基于决策图的模拟器中模拟量子电路的顺序. 而后的工作,他们根据切割受控门的思路^[58]来并行化基于决策图的模拟. Shen 等人^[69]通过将重排序技术和决策图方法结合也提升了决策图方式模拟的性能.

4.8 分布式思路

在全振幅量子模拟中,为应对随量子比特数增加而指数级增长的内存需求,引入分布式技术,有效利用多设备资源,提高模拟能力和效率.

随着量子比特数目的增加,所需要的内存空间呈指数级增长. 假设量子态振幅以双精度复数表示,30 个量子比特存储在内存空间中就需要 16 GB 的内存,而之后想继续扩展量子模拟器的能力,分布式技术的引入就至关重要^[70,71]. 在全振幅量子模拟中,分布式技术通常基于全局-局部量子比特的划分策略^[30]. 这里以消息传递接口 (message passing interface, MPI) 方法对应

的分布式方法^[71]为例来说明这种策略。

当有多个设备或机器来模拟量子电路时, MPI 可以有效地利用多个设备来模拟量子电路, 例如可以将 2^N 个幅度分配给 M 个 MPI 进程, 每个进程有 2^L 的内存, 其中 $M = 2^N/2^L$. 在这种情况下, 一般将 L 个量子比特视为局部量子比特, 将 $N-L$ 个量子比特视为全局量子比特. 此时, 当更新的幅度对仅在本地量子位时, 可以在每个过程中单独更新它. 如果更新的幅度对是对应于全局量子比特的量子比特, 则其全局量子比特需要全局-局部-交换策略. 全局-局部-交换策略是在每个过程中更新对应的量子状态信息, 使新目标位上的量子位成为本地量子位, 然后更新本地量子位.

全局局部量子位的划分方法是分布式量子电路模拟的重要思路, 无论是利用超级计算机集群, 还是利用多 GPU 的思路, 或者是用二级存储扩展模拟上限, 核心思路都基于这种全局-局部量子位的划分交换策略.

4.9 GPU 优化

在高性能计算平台上, 很多工作^[42,46,49-51,57,66,72-80]利用 GPU 加速量子电路模拟. 通过优化内存使用、数据交换策略和减少 CPU-GPU 通信开销等方法, 显著提高了量子模拟器的性能.

在 GPU 中, 存在 6 种内存类型: 寄存器、共享内存、局部内存、常量内存、纹理内存和全局内存. 与容量较大的全局内存相比, 共享内存的速度更快, 有效利用共享内存可以显著加速量子模拟器的性能. 例如, 可以将部分量子态复制到共享内存中, 以实现更低延迟的量子门操作^[76,77,80]. HyQuas^[57]优化了门融合技术, 使其适应共享内存的大小, 从而实现性能提升.

模拟器的一种有效方法是基于之前提及的分布式方法中的全局-局部-交换策略^[71]. HyQuas^[57]根据这种策略提出了以 GPU 为中心的数据交换策略, 以增强模拟性能. 而 dgQuEST^[81]则提出了一种流水线通信方案, 以减少分布式内存访问的开销, 进而提高在多 GPU 节点上的性能.

在使用 GPU 加速量子电路模拟时, CPU 与 GPU 之间的通信开销极为显著. Q-GPU^[66]主要专注于降低这种通信开销, 通过主动状态振幅传输、零振幅动态冗余消除和量子电路重排序等技术, 以最小化运行开销减少非零状态振幅引起的数据传输. dgQuEST^[81]有效地结合了 CPU 内存的大容量和 GPU 内存的高性能,

提出了一种 CPU-GPU 混合内存管理方案.

4.10 FPGA 优化

量子电路模拟通常涉及大量并行操作, 而现代的 FPGA 成为加速这类计算的理想选择. FPGA 提供了卓越的浮点计算能力, 并且具有极高的可扩展性. 作为一种可编程逻辑设备, FPGA 可以构建高度并行的架构, 这些架构能够紧密地模拟量子计算的特性. 在 FPGA 平台上, 已有许多工作开发了高效的并行能力模拟器^[82-86]. 例如, Qian 等人^[86]根据 FPGA 硬件实现量子的内在并行性, 提出 QFT-n 的递归方法, 以高效利用 FPGA 资源进行量子系统模拟.

4.11 存储设备优化

全振幅方式下的量子电路模拟是一个内存密集型任务, IBM 研究团队^[87]为了挑战 Google 的量子优越性任务, 提出可以在 Summit 超级计算机上通过使用硬盘克服量子指数级增长的复杂性, 只需几天而不是 Google 所提出的 1 万年就能完成量子优越性任务的模拟. 尽管 IBM 的研究团队未能具体实现其目标, 但值得关注的是, 采用存储设备扩展量子模拟器全振幅能力的一项工作, 即 SnuQS^[88]. SnuQS 设计了一个框架, 通过使用硬盘驱动器 (HDD) 和非易失性内存快闪存储器 (NVMe SSD) 等存储设备, 而非仅依赖于内存, 来扩展量子电路模拟的能力. 相对于传统模拟器, 它们依赖于虚拟内存来增强模拟能力, 但通常因 I/O 带宽使用不足和频繁的分页导致性能下降, SnuQS 引入了一种基于覆盖的内存管理技术, 以通过利用二级存储来增强量子模拟器的性能.

4.12 状态向量复用优化

针对含噪声量子模拟器的高计算开销, 通过状态向量复用优化, 如共享中间状态和重用子电路的中间结果, 可以有效减少噪声量子电路模拟的计算资源需求, 提高性能.

例如含有噪声的量子模拟器会在多次实验中反复执行同一个电路, 因此会带来较高开销. 之前讨论的量子模拟器优化主要集中在单次实验模拟的优化上, 很少考虑实验之间的优化. 通常情况下, 噪声的量子模拟实验复杂度通常比常规模拟过程高出数百倍^[89].

对于广泛应用于噪声模拟的蒙特卡罗模拟方法, 考虑到关注噪声的量子电路模拟需要多次评估, Li 等人^[90]提出了一种基于蒙特卡罗模拟的噪声实验, 其中多次注入误差, 并通过共享状态向量的中间状态在不

同实验之间临时存储和重用,以节省计算资源。而基于 Li 等人的思路, Wang 等人提出了 TQSim^[89],一种基于树的噪声模拟模型,动态地将一个电路划分为几个子电路,并在计算过程中重用这些子电路的中间结果,从而进一步提高了噪声模拟的性能。这些方法有助于有效减少噪声量子电路模拟的计算开销。

5 张量网络量子模拟器优化

张量网络收缩方法将量子电路的模拟转化为收缩相应张量网络的任务,这涉及按一定顺序收缩相应张量,直至只剩 1 个顶点。张量方法的复杂度随量子位的数量和电路的深度指数级增长,使得大规模电路的模拟在合理时间内难以实现。若仅对电路末端的 1 个或 1 小批状态幅值进行模拟,则张量网络的复杂度受到收缩过程中涉及的最大张量的约束,其复杂度会随着张量网络对应图的树宽呈指数增长^[91]。对于量子位数多但电路深度较浅的电路,这种方法特别有效,因为在许多情况下,复杂度也随电路深度指数增长。

例如, Google 提出的无向图模型方法^[31]以一种有效的方式采样量子比特数为 56,深度为 32 的电路,其中每个采样大约需要 600 s,但是仍然无法采样量子位 49 个,深度为 42 的量子电路。随后,阿里巴巴量子实验室^[25]根据无向图模型进一步实现了量子位 56,深度 42 的电路模拟。无向图模型方法本质上是一种解释量子态的位值与量子门之间关系的方法。位值会随着一系列量子门的操作而变化。对角矩阵不会改变位值,而非对角矩阵会改变位值。通过这种特点,可以合并同样位值的顶点,从而降低收缩的复杂度。该方法将原始量子电路映射到一个无向图模型,然后通过固定变量的值将无向图拆分为许多子图,然后通过变量消除算法对生成的图进行处理。然而,这种无向图方法的模拟只适用于相对简单的特定的电路类^[3]。

利用多体量子物理中量子态的投影纠缠对态 (projected entangled pair states, PEPS) 示量子态也是可行的方法。利用 PEPS 也可以构建通用量子电路模拟器, Chen 等人就通过这种方式模拟了量子比特数 64,深度 25 的量子电路^[92]。Google 和 NASA 合作提出了 qFlex^[93],重点研究了预期用于量子霸权实验的大小范围内的随机量子电路。后续的工作^[24]进一步将 qFlex 推广到超级计算机 Summit 上并实现了良好的性能。

在应用张量网络收缩方法的量子模拟器中,优化

不仅包括张量计算,还涵盖了收缩路径算法和张量切片算法的优化。接下来,我们将讨论这些优化方法。

5.1 张量计算优化

张量收缩的过程包括两个基本步骤:首先,张量指标的重新排列是必需的;其次,通过矩阵乘法来实现收缩。高阶张量的指标重排列要求数据项间的跨步移动,这对现有存储系统而言是一大挑战。因此,在高计算密度的多核处理器上,减少或隐藏重排列的开销成为实现有效张量收缩的关键设计问题。

SW_Qsim^[94]针对较窄的张量提供了一系列矩阵乘法和张量置换方法,并通过应用针对特定架构的优化来增强局部性和指令级并行性。Liu 等人^[26]在申威架构上实现了一种旨在减少内存访问的张量排列和矩阵乘法融合设计的通用矩阵乘法转置算法,以提高张量计算效率。

Gray 等人^[95]通过对张量网络进行预处理,通过吸收秩 1 和秩 2 张量可以极大地简化张量网络,大大减少张量网络整体的计算。

5.2 收缩路径算法优化

在基于张量的方法中,模拟特定样本或一批样本输出转化为相应张量网络的收缩问题。由于不同的收缩路径可能导致计算复杂性上的数量级差异,寻找最优收缩路径成为核心问题^[95]。目前最先进的张量网络收缩软件 Cotengra^[95]融合了多种方法,包括一系列有效的启发式算法来搜索压缩路径,如基于社区的方法和图划分方法,并提出了一种基于收缩树的方法来评估张量网络收缩过程中的相关开销。

阿里巴巴量子实验室^[25]采用了基于收缩树方法的特定策略,专注于识别和优化收缩树中的“茎”,即张量收缩的主要路径。一个“茎”是收缩树中的计算密集区域,通常包含大多数高秩张量。他们的研究表明,99% 的计算工作发生在这些“茎”中,因此,通过集中和优化“茎”,可以显著提高整个计算过程的效率。

另一方面,SW_Qsim^[94]针对特定场景提出了一种适用于矩形量子网格电路的最小内存压缩路径算法,以减少内存开销。Liu 等人^[26]考虑到将量子电路模拟任务映射到超大型超级计算机的背景,将寻找最佳收缩路径定义为一个多目标问题。虽然最小化计算复杂度非常重要,但生成更适合底层体系结构的张量对模拟速度同样至关重要。为了应对这种复杂情况,他们综合考虑了计算复杂度和计算密度,寻找最佳收缩路径,

因为计算密度在很大程度上决定了多核处理器上的性能。

5.3 张量切片算法优化

张量切片^[58]是一种常用的技术,用于平衡内存需求和我们可以执行的并发计算数量。切片的需求源于高阶张量的存储开销。在一些量子优越性电路对应的张量网络中,存在占用内存达到1 000 PB的张量^[95],超过大多数存储系统所能存储的极限。而通过切片有助于将内存需求降低到TB甚至GB级别,从而使模拟变得可行。

Cotengra^[95]构建了一个基于贪心算法的切片策略,通过反复选择最小化切片开销的维度,直到满足内存需求。阿里巴巴量子实验室^[25]采用了类似的贪心策略,在切片的两个步骤之间对收缩树进行局部调优,这种动态设计极大地降低了收缩树固有的切片开销。随后,阿里巴巴团队总结了相关工作,并提出了基于索引切片的量子模拟框架ACQDP^[96],将张量网络收缩任务分解为许多形状相同且易于并行的子任务。这些子任务的执行不依赖于相互之间的通信,使得这类算法易于在现代计算集群上部署。

与以往使用启发式算法的工作不同,Chen等人^[97]引入了“生命周期”概念,以更好地优化张量网络收缩中的切片过程。生命周期描述了张量网络中一条边如何通过分析其涉及的所有张量和收缩步骤来影响整个收缩过程。通过应用生命周期概念,可以显著减少原始计算的内存开销。在进程级别,利用张量切片实现分布式存储和并行化,并应用基于生命周期的切片查找方法来减少进程划分的开销。而在线程级别,切片有助于设计融合算法以减少内存访问,有时甚至可以将内存密集型内核转换为计算密集型内核,从而提高优化性能。这些方法有效地提升了张量网络收缩的效率。

6 结论与展望

量子模拟器在量子计算生态系统中扮演着至关重要的角色,对于模拟复杂的量子系统、解决优化问题、测试量子算法及错误校正研究等方面具有显著的重要性。

本文总结了两种主流的模拟方法,即状态向量全振幅模拟和张量网络收缩模拟,各有其独特特点。全振幅模拟在通用性方面表现更佳,但由于内存限制,其模拟的量子电路宽度受限。而张量网络收缩方法能够模

拟更广泛的量子比特电路,但通用性相对较低。量子模拟器在模拟量子系统时,由于量子叠加和纠缠特性,面临着指数级的存储和计算挑战。因此,优化量子模拟器以增强其模拟更大、更复杂量子系统的能力变得极为关键。本文深入探讨了量子模拟器中的关键优化技术和方法。特别是,本文重点分析了状态向量全振幅模拟方法和基于张量网络收缩的模拟策略及其优化。

未来量子模拟器的发展面临多项挑战。一方面,关键在于是否能充分利用量子态的特性,以充分激发传统计算的全部潜力,进而扩展量子模拟的能力上限;另一方面,开发适宜的算法来解决量子模拟中不可避免的指数级难题显得尤为重要。在展望未来量子模拟器的发展趋势时,有两个关键方面值得关注。首先,更有效地利用现有的高性能计算资源,实现模拟任务与计算资源的更有机结合,从而提高量子模拟器的效率和性能。其次,通过开发创新算法和优化模拟策略,不断增强量子模拟器的功能和计算能力。此外,未来的发展也需要针对具体应用场景和问题进行深入考虑,不同场景下的问题可能需要采用不同的模拟方法和策略。综上所述,量子模拟器的未来发展需在硬件、算法及应用需求间实现全面的综合考虑,以确保更高的效率和精确度。

参考文献

- 1 Shor PW. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM review*, 1999, 41(2): 303–332. [doi: [10.1137/S0036144598347011](https://doi.org/10.1137/S0036144598347011)]
- 2 Grover LK. A fast quantum mechanical algorithm for database search. *Proceedings of the 28th Annual ACM Symposium on Theory of Computing*. Philadelphia: ACM, 1996. 212–219.
- 3 Boixo S, Isakov SV, Smelyanskiy VN, *et al.* Characterizing quantum supremacy in near-term devices. *Nature Physics*, 2018, 14(6): 595–600. [doi: [10.1038/s41567-018-0124-x](https://doi.org/10.1038/s41567-018-0124-x)]
- 4 Preskill J. Quantum computing in the NISQ era and beyond. *Quantum*, 2018, 2: 79. [doi: [10.22331/q-2018-08-06-79](https://doi.org/10.22331/q-2018-08-06-79)]
- 5 Farhi E, Goldstone J, Gutmann S, *et al.* Quantum computation by adiabatic evolution. *arXiv:quant-ph/0001106*, 2000.
- 6 Raussendorf R, Briegel HJ. A one-way quantum computer. *Physical Review Letters*, 2001, 86(22): 5188–5191. [doi: [10.1103/PhysRevLett.86.5188](https://doi.org/10.1103/PhysRevLett.86.5188)]

- 7 Kitaev AY. Quantum error correction with imperfect gates. In: Hirota O, Holevo AS, Caves CM, eds. *Quantum Communication, Computing, and Measurement*. Boston: Springer, 1997. 181–188.
- 8 Deutsch DE. Quantum computational networks. *Proceedings of the 1989 Royal Society A: Mathematical and Physical Sciences*, 1989, 425(1868): 73–90.
- 9 Nielsen MA, Chuang IL. *Quantum Computation and Quantum Information*. Cambridge Press. 2002. http://almuhammadi.com/sultan/books_2020/Nielsen_Chuang.pdf.
- 10 Kaye P, Laflamme R, Mosca M. *An Introduction to Quantum Computing*. Oxford: Oxford University Press, 2007.
- 11 Jozsa R. Quantum algorithms and the Fourier transform. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 1998, 454(1969): 323–337.
- 12 Harrow AW, Hassidim A, Lloyd S. Quantum algorithm for linear systems of equations. *Physical Review Letters*, 2009, 103(15): 150502. [doi: [10.1103/PhysRevLett.103.150502](https://doi.org/10.1103/PhysRevLett.103.150502)]
- 13 Coppersmith D. An approximate Fourier transform useful in quantum factoring. arXiv:quant-ph/0201067, 2002.
- 14 Brassard G, Hoyer P, Mosca M, *et al.* Quantum amplitude amplification and estimation. *Contemporary Mathematics*, 2002, 305: 53–74.
- 15 Farhi E, Goldstone J, Gutmann S. A quantum approximate optimization algorithm. arXiv:1411.4028, 2014.
- 16 Cerezo M, Arrasmith A, Babbush R, *et al.* Variational quantum algorithms. *Nature Reviews Physics*, 2021, 3(9): 625–644. [doi: [10.1038/s42254-021-00348-9](https://doi.org/10.1038/s42254-021-00348-9)]
- 17 Tilly J, Chen HX, Cao SX, *et al.* The variational quantum eigensolver: A review of methods and best practices. *Physics Reports*, 2022, 986: 1–128. [doi: [10.1016/j.physrep.2022.08.003](https://doi.org/10.1016/j.physrep.2022.08.003)]
- 18 Biamonte J, Wittek P, Pancotti N, *et al.* Quantum machine learning. *Nature*, 2017, 549(7671): 195–202. [doi: [10.1038/nature23474](https://doi.org/10.1038/nature23474)]
- 19 Lloyd S, Mohseni M, Rebentrost P. Quantum algorithms for supervised and unsupervised machine learning. arXiv: 1307.0411, 2013.
- 20 Rebentrost P, Mohseni M, Lloyd S. Quantum support vector machine for big data classification. *Physical Review Letters*, 2014, 113(13): 130503. [doi: [10.1103/PhysRevLett.113.130503](https://doi.org/10.1103/PhysRevLett.113.130503)]
- 21 Aaronson S, Chen L J. Complexity-theoretic foundations of quantum supremacy experiments. *Proceedings of the 32nd Computational Complexity Conference*. Riga, 2017. 1–67.
- 22 Arute F, Arya K, Babbush R, *et al.* Quantum supremacy using a programmable superconducting processor. *Nature*, 2019, 574(7779): 505–510. [doi: [10.1038/s41586-019-1666-5](https://doi.org/10.1038/s41586-019-1666-5)]
- 23 Zlokapa A, Villalonga B, Boixo S, *et al.* Boundaries of quantum supremacy via random circuit sampling. *npj Quantum Information*, 2023, 9(1): 36. [doi: [10.1038/s41534-023-00703-x](https://doi.org/10.1038/s41534-023-00703-x)]
- 24 Villalonga B, Lyakh D, Boixo S, *et al.* Establishing the quantum supremacy frontier with a 281 Pflop/s simulation. *Quantum Science and Technology*, 2020, 5(3): 034003. [doi: [10.1088/2058-9565/ab7eeb](https://doi.org/10.1088/2058-9565/ab7eeb)]
- 25 Huang C, Zhang F, Newman M, *et al.* Classical simulation of quantum supremacy circuits. arXiv:2005.06787, 2020.
- 26 Liu Y, Liu X, Li F, *et al.* Closing the “quantum supremacy” gap: Achieving real-time simulation of a random quantum circuit using a new Sunway supercomputer. *Proceedings of the 2021 International Conference for High Performance Computing, Networking, Storage and Analysis*. St. Louis: IEEE, 2021. 1–12.
- 27 Li RL, Wu BJ, Ying MS, *et al.* Quantum supremacy circuit simulation on Sunway TaihuLight. *IEEE Transactions on Parallel and Distributed Systems*, 2020, 31(4): 805–816. [doi: [10.1109/TPDS.2019.2947511](https://doi.org/10.1109/TPDS.2019.2947511)]
- 28 Pan F, Chen KY, Zhang P. Solving the sampling problem of the sycamore quantum circuits. *Physical Review Letters*, 2022, 129(9): 090502. [doi: [10.1103/PhysRevLett.129.090502](https://doi.org/10.1103/PhysRevLett.129.090502)]
- 29 Smelyanskiy M, Sawaya NPD, Aspuru-Guzik A. qHiPSTER: The quantum high performance software testing environment. arXiv:1601.07195, 2016.
- 30 Häner T, Steiger DS. 0.5 petabyte simulation of a 45-qubit quantum circuit. *Proceedings of the 2017 International Conference for High Performance Computing, Networking, Storage and Analysis*. Denver: IEEE, 2017. 1–10.
- 31 Boixo S, Isakov SV, Smelyanskiy VN, *et al.* Simulation of low-depth quantum circuits as complex undirected graphical models. arXiv:1712.05384, 2017.
- 32 Gottesman D. The Heisenberg representation of quantum computers. *Proceedings of the 22nd International Colloquium on Group Theoretical Methods in Physics*. Hobart, 1999. 32–43.
- 33 Aaronson S, Gottesman D. Improved simulation of stabilizer circuits. *Physical Review A*, 2004, 70(5): 052328. [doi: [10.1103/PhysRevA.70.052328](https://doi.org/10.1103/PhysRevA.70.052328)]
- 34 Hillmich S, Markov IL, Wille R. Just like the real thing: Fast weak simulation of quantum computation. *Proceedings of the 57th ACM/IEEE Design Automation Conference (DAC)*.

- San Francisco: IEEE, 2020. 1–6.
- 35 Markov IL, Fatima A, Isakov SV, *et al.* Massively parallel approximate simulation of hard quantum circuits. Proceedings of the 57th ACM/IEEE Design Automation Conference (DAC). San Francisco: IEEE, 2020. 1–6.
- 36 Anders S, Briegel HJ. Fast simulation of stabilizer circuits using a graph-state representation. *Physical Review A*, 2006, 73(2): 022334. [doi: [10.1103/PhysRevA.73.022334](https://doi.org/10.1103/PhysRevA.73.022334)]
- 37 Bravyi S, Browne D, Calpin P, *et al.* Simulation of quantum circuits by low-rank stabilizer decompositions. *Quantum*, 2019, 3: 181. [doi: [10.22331/q-2019-09-02-181](https://doi.org/10.22331/q-2019-09-02-181)]
- 38 Biamonte J, Bergholm V. Tensor networks in a nutshell. arXiv:1708.00006, 2017.
- 39 Barends R, Kelly J, Megrant A, *et al.* Superconducting quantum circuits at the surface code threshold for fault tolerance. *Nature*, 2014, 508(7497): 500–503. [doi: [10.1038/nature13171](https://doi.org/10.1038/nature13171)]
- 40 Bennett CH, Divincenzo DP, Smolin JA, *et al.* Mixed-state entanglement and quantum error correction. *Physical Review A*, 1996, 54(5): 3824–3851. [doi: [10.1103/PhysRevA.54.3824](https://doi.org/10.1103/PhysRevA.54.3824)]
- 41 Brassard G, Braunstein SL, Cleve R. Teleportation as a quantum computation. *Physica D: Nonlinear Phenomena*, 1998, 120(1–2): 43–47. [doi: [10.1016/S0167-2789\(98\)00043-8](https://doi.org/10.1016/S0167-2789(98)00043-8)]
- 42 Cross A. The IBM Q experience and QISKit open-source quantum computing software. Proceedings of the 2018 APS March Meeting. 2018. Abstract ID: BAPS.2018.MAR.L58.3
- 43 Omole V, Tyagi A, Hanus A, *et al.* Cirq: A Python framework for creating, editing, and invoking quantum circuits. <https://sdmay20-08.sd.ece.iastate.edu/docs/Design-Document-v2.pdf>. [2023-11-17].
- 44 Svore K, Geller A, Troyer M, *et al.* Q#: Enabling scalable quantum computing and development with a high-level DSL. Proceedings of the 2018 Real World Domain Specific Languages Workshop. Vienna: ACM, 2018. 7.
- 45 Broughton M, Verdon G, McCourt T, *et al.* TensorFlow quantum: A software framework for quantum machine learning. arXiv:2003.02989, 2020.
- 46 Jones T, Brown A, Bush I, *et al.* QuEST and high performance simulation of quantum computers. *Scientific Reports*, 2019, 9(1): 10736. [doi: [10.1038/s41598-019-47174-9](https://doi.org/10.1038/s41598-019-47174-9)]
- 47 Guerreschi GG, Hogaboam J, Baruffa F, *et al.* Intel quantum simulator: A cloud-ready high-performance simulator of quantum circuits. *Quantum Science and Technology*, 2020, 5(3): 034007. [doi: [10.1088/2058-9565/ab8505](https://doi.org/10.1088/2058-9565/ab8505)]
- 48 Kelly A. Simulating quantum computers using OpenCL. arXiv:1805.00988, 2018.
- 49 Efthymiou S, Ramos-Calderer S, Bravo-Prieto C, *et al.* Qibo: A framework for quantum simulation with hardware acceleration. *Quantum Science and Technology*, 2022, 7(1): 015018. [doi: [10.1088/2058-9565/ac39f5](https://doi.org/10.1088/2058-9565/ac39f5)]
- 50 Luo XZ, Liu JG, Zhang P, *et al.* Yao.jl: Extensible, efficient framework for quantum algorithm design. *Quantum*, 2020, 4: 341.
- 51 Suzuki Y, Kawase Y, Masumura Y, *et al.* Qulacs: A fast and versatile quantum circuit simulator for research purpose. *Quantum*, 2021, 5: 559. [doi: [10.22331/q-2021-10-06-559](https://doi.org/10.22331/q-2021-10-06-559)]
- 52 Khammassi N, Ashraf I, Fu X, *et al.* QX: A high-performance quantum computer simulation platform. Proceedings of the 2017 Design, Automation & Test in Europe Conference & Exhibition (DATE). Lausanne: IEEE, 2017. 464–469.
- 53 Steiger DS, Häner T, Troyer M. ProjectQ: An open source software framework for quantum computing. *Quantum*, 2018, 2: 49. [doi: [10.22331/q-2018-01-31-49](https://doi.org/10.22331/q-2018-01-31-49)]
- 54 Wecker D, Svore KM. LIQUi>: A software design architecture and domain-specific language for quantum computing. arXiv:1402.4467, 2014.
- 55 Gheorghiu V. Quantum++: A modern C++ quantum computing library. *PLoS One*, 2018, 13(12): e0208073. [doi: [10.1371/journal.pone.0208073](https://doi.org/10.1371/journal.pone.0208073)]
- 56 Khammassi N, Ashraf I, Someren JV, *et al.* OpenQL: A portable quantum programming framework for quantum accelerators. *ACM Journal on Emerging Technologies in Computing Systems*, 2022, 18(1): 13.
- 57 Zhang C, Song ZY, Wang HJ, *et al.* HyQuas: Hybrid partitioner based quantum circuit simulation system on GPU. Proceedings of the 2021 ACM International Conference on Supercomputing. ACM, 2021. 443–454.
- 58 Pednault E, Gunnels JA, Nannicini G, *et al.* Breaking the 49-qubit barrier in the simulation of quantum circuits. arXiv: 1710.05867, 2017.
- 59 Bian HD, Huang JQ, Tang JH, *et al.* PAS: A new powerful and simple quantum computing simulator. *Software: Practice and Experience*, 2023, 53(1): 142–159. [doi: [10.1002/spe.3049](https://doi.org/10.1002/spe.3049)]
- 60 Cypher R, Sanz JLC. SIMD architectures and algorithms for image processing and computer vision. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 1989, 37(12): 2158–2174. [doi: [10.1109/29.45558](https://doi.org/10.1109/29.45558)]
- 61 Lomont C. Introduction to Intel® advanced vector extensions. Intel White Paper, 2011. 23. http://www.lomont.org/papers/2011/Intro_to_Intel_AVX-Final.pdf. [2023-11-18].

- 62 Fatima A, Markov IL. Faster Schrödinger-style simulation of quantum circuits. Proceedings of the 2021 IEEE International Symposium on High-performance Computer Architecture (HPCA). Seoul: IEEE, 2021. IEEE. 194–207.
- 63 Lam MD, Rothberg EE, Wolf ME. The cache performance and optimizations of blocked algorithms. ACM SIGPLAN Notices, 1991, 26(4): 63–74. [doi: [10.1145/106973.106981](https://doi.org/10.1145/106973.106981)]
- 64 Wu XC, Di S, Dasgupta EM, *et al.* Full-state quantum circuit simulation by using data compression. Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis. Denver: ACM, 2019. 80.
- 65 Zulehner A, Wille R. Advanced simulation of quantum computations. IEEE Transactions on Computer-aided Design of Integrated Circuits and Systems, 2019, 38(5): 848–859. [doi: [10.1109/TCAD.2018.2834427](https://doi.org/10.1109/TCAD.2018.2834427)]
- 66 Zhao YL, Guo YN, Yao Y, *et al.* Q-GPU: A recipe of optimizations for quantum circuit simulation using GPUs. Proceedings of the 2022 IEEE International Symposium on High-performance Computer Architecture (HPCA). Seoul: IEEE, 2022. 726–740.
- 67 Niemann P, Wille R, Miller DM, *et al.* QMDDs: Efficient quantum function representation and manipulation. IEEE Transactions on Computer-aided Design of Integrated Circuits and Systems, 2016, 35(1): 86–99. [doi: [10.1109/TCAD.2015.2459034](https://doi.org/10.1109/TCAD.2015.2459034)]
- 68 Burgholzer L, Ploier A, Wille R. Exploiting arbitrary paths for the simulation of quantum circuits with decision diagrams. Proceedings of the 2022 Design, Automation & Test in Europe Conference & Exhibition (DATE). Antwerp: IEEE, 2022. 64–67.
- 69 Shen JC, Long LB, Okita M, *et al.* A reorder trick for decision diagram based quantum circuit simulation. arXiv: 2211.07110, 2022.
- 70 De Raedt H, Jin FP, Willsch D, *et al.* Massively parallel quantum computer simulator, eleven years later. Computer Physics Communications, 2019, 237: 47–61. [doi: [10.1016/j.cpc.2018.11.005](https://doi.org/10.1016/j.cpc.2018.11.005)]
- 71 De Raedt K, Michielsen K, De Raedt H, *et al.* Massively parallel quantum computer simulator. Computer Physics Communications, 2007, 176(2): 121–136. [doi: [10.1016/j.cpc.2006.08.007](https://doi.org/10.1016/j.cpc.2006.08.007)]
- 72 Avila A, Maron A, Reiser R, *et al.* GPU-aware distributed quantum simulation. Proceedings of the 29th Annual ACM Symposium on Applied Computing. Gyeongju: ACM, 2014. 860–865.
- 73 Avila A, Reiser RHS, Pilla ML, *et al.* Optimizing D-GM quantum computing by exploring parallel and distributed quantum simulations under GPUs architecture. Proceedings of the 2016 IEEE Congress on Evolutionary Computation (CEC). Vancouver: IEEE, 2016. 5146–5153.
- 74 Amariutei A, Caraiman S. Parallel quantum computer simulation on the GPU. Proceedings of the 15th International Conference on System Theory, Control and Computing. Sinaia: IEEE, 2011. 1–6.
- 75 Li Z, Yuan JB. Quantum computer simulation on GPU cluster incorporating data locality. Proceedings of the 3rd International Conference on Cloud Computing and Security. Nanjing: Springer, 2017. 85–97.
- 76 Zhang P, Yuan JB, Lu XW. Quantum computer simulation on multi-GPU incorporating data locality. Proceedings of the 15th International Conference on Algorithms and Architectures for Parallel Processing. Zhangjiajie: Springer, 2015. 241–256.
- 77 Gutierrez E, Romero S, Trenas MA, *et al.* Simulation of quantum gates on a novel GPU architecture. Proceedings of the 7th WSEAS International Conference on Systems Theory and Scientific Computation. Vouliagmeni: WSEAS, 2007. 121–126.
- 78 Doi J, Takahashi H, Raymond R, *et al.* Quantum computing simulator on a heterogenous HPC system. Proceedings of the 16th ACM International Conference on Computing Frontiers. Alghero: ACM, 2019. 85–93.
- 79 Gutierrez E, Romero S, Trenas MA, *et al.* Parallel quantum computer simulation on the CUDA architecture. Proceedings of the 8th International Conference on Computational Science (ICCS 2008). Kraków: Springer. 700–709.
- 80 Gutiérrez E, Romero S, Trenas MA, *et al.* Quantum computer simulation using the CUDA programming model. Computer Physics Communications, 2010, 181(2): 283–300. [doi: [10.1016/j.cpc.2009.09.021](https://doi.org/10.1016/j.cpc.2009.09.021)]
- 81 Feng TY, Chen SY, You X, *et al.* dgQuEST: Accelerating large scale quantum circuit simulation through hybrid CPU-GPU memory hierarchies. Proceedings of the 18th Network and Parallel Computing (IFIP WG 10.3) International Conference. Paris: Springer, 2021. 16–27.
- 82 Khalid AU, Zilic Z, Radecka K. FPGA emulation of quantum circuits. Proceedings of the 2004 IEEE International Conference on Computer Design: VLSI in Computers and Processors. San Jose: IEEE, 2004. 310–315.
- 83 Fujishima M. FPGA-based high-speed emulator of quantum computing. Proceedings of the 2003 IEEE International Conference on Field-programmable Technology (FPT)(IEEE

- Cat. No. 03EX798). Tokyo: IEEE, 2003. 21–26.
- 84 Lee YH, Khalil-Hani M, Marsono MN. FPGA-based quantum circuit emulation: A case study on quantum fourier transform. Proceedings of the 2014 International Symposium on Integrated Circuits (ISIC). Singapore: IEEE, 2014. 512–515.
- 85 Silva A, Zabaleta OG. FPGA quantum computing emulator using high level design tools. Proceedings of the 2017 Argentine Symposium and Conference on Embedded Systems (CASE). Buenos Aires: IEEE, 2017. 1–6.
- 86 Qian Y, Wang MY, Chen JL, *et al.* Efficient FPGA emulation of quantum Fourier transform. Proceedings of the 2019 China Semiconductor Technology International Conference (CSTIC). Shanghai: IEEE, 2019. 1–3.
- 87 Pednault E, Gunnels JA, Nannicini G, *et al.* Leveraging secondary storage to simulate deep 54-qubit sycamore circuits. arXiv:1910.09534, 2019.
- 88 Park D, Kim H, Kim J, *et al.* SnuQS: Scaling quantum circuit simulation using storage devices. Proceedings of the 36th ACM International Conference on Supercomputing. ACM, 2022. 6.
- 89 Wang M, Huang R, Tannu S, *et al.* TQSim: A case for reuse-focused tree-based quantum circuit simulation. arXiv:2203.13892, 2022.
- 90 Li GS, Ding YF, Xie Y. Eliminating redundant computation in noisy quantum computing simulation. Proceedings of the 57th ACM/IEEE Design Automation Conference (DAC). San Francisco: IEEE, 2020. 1–6.
- 91 Markov IL, Shi YY. Simulating quantum computation by contracting tensor networks. SIAM Journal on Computing, 2008, 38(3): 963–981. [doi: [10.1137/050644756](https://doi.org/10.1137/050644756)]
- 92 Chen ZY, Zhou Q, Xue C, *et al.* 64-qubit quantum circuit simulation. Science Bulletin, 2018, 63(15): 964–971. [doi: [10.1016/j.scib.2018.06.007](https://doi.org/10.1016/j.scib.2018.06.007)]
- 93 Villalonga B, Boixo S, Nelson B, *et al.* A flexible high-performance simulator for verifying and benchmarking quantum circuits implemented on real hardware. npj Quantum Information, 2019, 5(1): 86. [doi: [10.1038/s41534-019-0196-1](https://doi.org/10.1038/s41534-019-0196-1)]
- 94 Li F, Liu X, Liu Y, *et al.* SW_Qsim: A minimize-memory quantum simulator with high-performance on a new Sunway supercomputer. Proceedings of the 2021 International Conference for High Performance Computing, Networking, Storage and Analysis. St. Louis: IEEE, 2021. 1–13.
- 95 Gray J, Kourtis S. Hyper-optimized tensor network contraction. Quantum, 2021, 5: 410. [doi: [10.22331/q-2021-03-15-410](https://doi.org/10.22331/q-2021-03-15-410)]
- 96 Huang C, Zhang F, Newman M, *et al.* Efficient parallelization of tensor network contraction for simulating quantum computation. Nature Computational Science, 2021, 1(9): 578–587. [doi: [10.1038/s43588-021-00119-7](https://doi.org/10.1038/s43588-021-00119-7)]
- 97 Chen YJ, Liu Y, Shi XM, *et al.* Lifetime-based optimization for simulating quantum circuits on a new Sunway supercomputer. Proceedings of the 28th ACM SIGPLAN Annual Symposium on Principles and Practice of Parallel Programming. Montreal: ACM, 2023. 148–159.

(校对责编: 张重毅)