



# 计算机科学

COMPUTER SCIENCE

## 基于矩阵乘积态的有限纠缠量子傅里叶变换模拟

刘晓楠, 廉德萌, 杜帅岐, 刘正煜

引用本文

刘晓楠, 廉德萌, 杜帅岐, 刘正煜. 基于矩阵乘积态的有限纠缠量子傅里叶变换模拟[J]. 计算机科学, 2024, 51(9): 80-86.

LIU Xiaonan, LIAN Demeng, DU Shuaiqi, LIU Zhengyu. [Simulation of Limited Entangled Quantum Fourier Transform Based on Matrix Product State](#) [J]. Computer Science, 2024, 51(9): 80-86.

---

## 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)

### [Grover量子搜索算法在“嵩山”超级计算机系统模拟](#)

Simulation of Grover's Quantum Search Algorithm in "Songsshan" Supercomputer System  
计算机科学, 2024, 51(9): 96-102. <https://doi.org/10.11896/jsjcx.230600219>

### [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/jsjcx.230800193>

### [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/jsjcx.220400051>

### [基于“嵩山”超级计算机系统下HHL算法的模拟实现](#)

Simulation Implementation of HHL Algorithm Based on Songsshan Supercomputer System  
计算机科学, 2023, 50(6): 74-80. <https://doi.org/10.11896/jsjcx.220500108>

### [混沌自适应量子萤火虫算法](#)

Chaotic Adaptive Quantum Firefly Algorithm  
计算机科学, 2023, 50(4): 204-211. <https://doi.org/10.11896/jsjcx.220100242>

# 基于矩阵乘积态的有限纠缠量子傅里叶变换模拟

刘晓楠<sup>1,2</sup> 廉德萌<sup>2,3</sup> 杜帅岐<sup>2,3</sup> 刘正煜<sup>2,3</sup>

1 数学工程与先进计算国家重点实验室(信息工程大学) 郑州 450000

2 国家超级计算郑州中心 郑州 450000

3 郑州大学计算机与人工智能学院 郑州 450000

(prof.liu.xn@foxmail.com)

**摘要** 与经典计算不同,在量子计算中量子比特可以处于叠加态,多个量子比特之间还可以形成纠缠态。表示  $n$  个量子比特组成的量子态需要存储  $2^n$  个振幅,这种指数级的存储开销使得大规模的量子模拟难以进行。然而当量子态的纠缠程度有限时,使用矩阵乘积态表示量子态仅需要线性的空间复杂度,可以扩大模拟的规模。使用 HIP-Clang 语言,基于 CPU+DCU 的异构编程模型,使用矩阵乘积态表示量子态,对量子傅里叶变换进行模拟。结合矩阵乘积态的特点,对量子傅里叶变换线路进行分析,减少模拟实现时不必要的张量缩并运算与正交化构建。对模拟过程中的张量缩并进行分析,使用 TTGT 算法完成张量缩并运算,同时利用 DCU 的并行处理能力来提高效率。对模拟结果进行分析,分别通过振幅误差与半经典 Draper 量子加法器的结果验证了模拟的正确性。对模拟规模进行分析,当量子态的纠缠熵最大时,使用 16 GB 的内存空间最多只能模拟 24 位的量子态,而当量子态内部纠缠程度较低时,可以对上百位的量子态进行量子傅里叶变换模拟。

**关键词** 量子模拟;量子傅里叶变换;矩阵乘积态;异构计算;DCU;HIP-Clang

**中图分类号** TP385

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

LIU Xiaonan<sup>1,2</sup>, LIAN Demeng<sup>2,3</sup>, DU Shuaiqi<sup>2,3</sup> and LIU Zhengyu<sup>2,3</sup>

1 State Key Laboratory of Mathematical Engineering and Advanced Computing(Information Engineering University), Zhengzhou 450000, China

2 National Supercomputing Center in Zhengzhou, Zhengzhou 450000, China

3 School of Computer and Artificial Intelligence, Zhengzhou University, Zhengzhou 450000, China

**Abstract** Unlike classical computing, qubits in quantum computing can be in the superposition state and entangled state can be formed between multiple qubits. Representing a quantum state composed of  $n$  qubits requires storing  $2^n$  power amplitudes. The exponential memory cost makes large-scale quantum simulation difficult. Using the HIP-Clang language, based on the heterogeneous programming model of CPU+DCU and representing the quantum state with the matrix product state, quantum Fourier transform is simulated. By combining the characteristics of the matrix product state and analyzing the quantum Fourier transform circuit, unnecessary tensor contraction operations and orthogonalization construction are reduced during simulation implementation. Tensor contraction during simulation is analyzed and the TTGT algorithm is used to complete tensor contraction operations while utilizing DCU's parallel processing capabilities to improve efficiency. Simulation results are analyzed and the correctness of the simulation is verified through amplitude error and semi-classical Draper quantum adder results. Analyzing simulation scale, when the entanglement entropy of the quantum state is maximum, using 16 GB of memory can simulate up to 24 bit quantum states at most, while when the entanglement of the quantum state is limited, it can simulate hundreds of qubits of quantum Fourier transform.

**Keywords** Quantum simulation, Quantum Fourier transform, Matrix product state, Heterogeneous computing, DCU, HIP-Clang

## 1 引言

自 1982 年 Feynman 等提出量子计算的概念以来,量子算法的发展取得了长足的进步,出现了许多可以用来解决实际

问题的量子算法。如 Shor<sup>[1]</sup> 提出的用于求解质数乘积分解以及离散对数的 Shor 算法, Harrow 等<sup>[2]</sup> 提出的用于求解线性方程组的 HHL 算法。这些算法相较于经典算法,时间效率提升显著,甚至可以达到指数级。这使得解决在有限时间

到稿日期:2023-03-29 返修日期:2023-07-28

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:廉德萌(dm.lian@outlook.com)

内破解 RSA 加密这类经典计算机难以完成的任务成为了一种可能。

量子傅里叶变换是一种有效的对量子振幅进行傅里叶变换的量子算法<sup>[3]</sup>,是上述乃至更多量子算法的重要组成部分。此外,量子傅里叶变换相较于经典的快速傅里叶变换,也有着指数级的加速。对量子傅里叶变换进行研究,有助于加深对量子计算的理解,扩展量子傅里叶变换的应用范围,同时也方便了对其他量子算法进行研究。

由于量子退相干等因素的影响,量子计算机的研制建造工作十分困难。当前量子计算机仍处于嘈杂的中尺度量子计算机阶段<sup>[4]</sup>,距离通用量子计算机仍有不小距离。因此,当前使用经典计算机对量子计算进行模拟仍是有必要的。

传统的量子模拟常采用全振幅模拟方法,需要存储全部量子态的振幅。模拟  $n$  个量子比特构成的叠加态时,需要存储  $2^n$  个振幅。指数级的存储开销限制了对量子傅里叶变换进行模拟的最大规模。使用矩阵乘积态表示量子态,可以有效地对纠缠程度有限的量子态进行量子傅里叶变换模拟<sup>[5]</sup>。本文使用矩阵乘积态表示量子态,利用量子态的内部纠缠结构降低存储开销,从而实现大规模的量子傅里叶变换模拟。

量子傅里叶变换是许多量子算法的关键部分,国内外许多学者曾使用不同的模拟方法对其进行模拟实验。Xie 等<sup>[6]</sup>、Liu 等<sup>[7]</sup>基于全振幅模拟方法分别在“嵩山”超级计算机系统和神威·太湖之光上完成了 44 和 46 个量子比特的量子傅里叶变换模拟。Woolfe 等<sup>[8]</sup>对量子傅里叶变换算符构造矩阵乘积算符,完成了有限纠缠下对近似量子傅里叶变换的模拟。Saitoh<sup>[9]</sup>基于 TEMPS 方法,开发了量子模拟库 ZKCM\_QC,并对一些量子算法进行了模拟。

本文基于 CPU+DCU 的异构编程模型,在“嵩山”超级计算机系统的单个计算节点上完成了基于矩阵乘积态的量子傅里叶变换的模拟以及验证。结合矩阵乘积态的特点,对量子傅里叶变换线路进行分析,减少了张量缩并以及矩阵正交化的次数。对模拟过程中的张量缩并运算进行分析,使用 TTGT (Transpose-Transpose-GEMM-Transpose) 算法,按照最优缩并顺序进行张量的缩并运算,并利用 DCU 的并行处理能力,提高了模拟的效率。

本文第 2 章和第 3 章分别介绍了矩阵乘积态以及量子傅里叶变换的理论基础;第 4 章对基于矩阵乘积态的量子傅里叶变换模拟在 DCU 上的实现进行介绍;第 5 章对模拟结果进行了分析;最后总结全文并展望未来。

## 2 矩阵乘积态

矩阵乘积态是对量子态系数张量进行分解得到的一系列阶数较低的张量<sup>[10]</sup>,如式(1)、式(2)所示:

$$|\phi\rangle = \sum_{s_1 s_2 \dots s_N} \phi_{s_1 s_2 \dots s_N} \prod_{n=1}^N |s_n\rangle \quad (1)$$

$$\phi_{s_1 s_2 \dots s_N} = \sum_{s_1 a_1} \mathbf{A}_{s_1 a_1}^{(1)} \mathbf{A}_{s_2 a_1 a_2}^{(2)} \dots \mathbf{A}_{s_{N-1} a_{N-2} a_{N-1}}^{(N-1)} \mathbf{A}_{s_N a_{N-1}}^{(N)} \quad (2)$$

其中,指标  $(s_n)$  为物理指标,代表希尔伯特空间;相邻两个张量共有的指标  $(a_n)$  为辅助指标,在张量缩并的过程中可以被求和掉。物理指标的维数  $d$  由其表示的量子系统决定,对单个量子比特进行表示时维数为 2,对应量子比特的两个基态

$|0\rangle, |1\rangle$ 。辅助指标的维数上限  $\chi$  则根据实际情况由人为给定。

对量子态进行 Schmidt 分解<sup>[11]</sup>,对应了对其振幅矩阵进行奇异值分解,如式(3)所示:

$$|\phi\rangle = \sum_{a=0}^{D-1} \Lambda_a |U^a\rangle |V^a\rangle \quad (3)$$

其中,  $|U^a\rangle, |V^a\rangle$  对应左右两边的量子体系,  $\Lambda_a$  为左右两个量子系统的纠缠谱。如图 1 所示,经过一次奇异值分解,左奇异矩阵为矩阵乘积态中的第一个张量,对应第一个量子比特。合并奇异值矩阵和右奇异矩阵,得到剩下的  $n-1$  阶张量对应剩下的  $n-1$  个量子比特。经过  $n-1$  次奇异值分解,分解  $n$  阶张量为  $n$  个小张量,从而构造出矩阵乘积态。

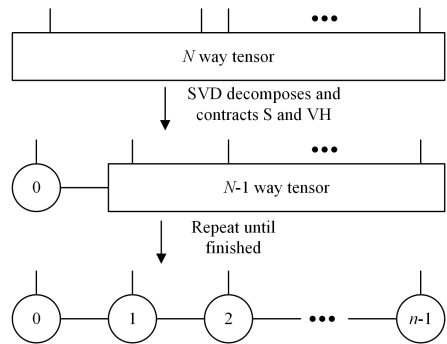


图 1 构建矩阵乘积态

Fig. 1 Construction of matrix product state

使用全振幅模拟时,量子态的参数随着量子比特的个数呈指数上升,空间复杂度为  $O(2^n)$ 。使用矩阵乘积态模拟,给定辅助指标的维数上限  $\chi$ ,此时矩阵乘积态中各个张量的最高维度为  $\chi^2 d$ ,参数个数仅随量子比特的个数线性增长,空间复杂度为  $O(nd\chi^2)$ 。

辅助指标的维数上限决定了矩阵乘积态表示的量子态的纠缠熵的上限<sup>[12]</sup>。当辅助指标的维数上限设置过低时,会引入误差。可通过构造矩阵乘积态的中心正交形式,对正交中心的张量进行分解时,根据辅助指标的维数上限  $\chi$  进行裁剪,使整体误差最小。

## 3 量子傅里叶变换

量子傅里叶变换是对一组正交基  $|0\rangle, |1\rangle, \dots, |2^n-1\rangle$  进行如式(4)操作的量子算符。

$$|j\rangle \rightarrow \frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} x_j e^{2\pi i j k / 2^n} |k\rangle \quad (4)$$

作用在任意量子态上的量子傅里叶变换表示如式(5)所示:

$$\begin{aligned} \sum_{j=0}^{2^n-1} x_j |j\rangle &\rightarrow \frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} e^{2\pi i j k / 2^n} |k\rangle \\ &= \sum_{k=0}^{2^n-1} y_k |k\rangle \end{aligned} \quad (5)$$

其中,振幅  $y_k$  由振幅  $x_k$  通过离散傅里叶变换得到。量子傅里叶变换的量子线路如图 2 所示,首先是一些按顺序排列的 Hadamard 门与受控相移门,矩阵形式如式(6)所示,最后通过 SWAP 门,矩阵形式如式(7)所示,调换量子比特的输出次序。

$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}, R_k = \begin{bmatrix} 1 & 0 \\ 0 & e^{2\pi i/2^k} \end{bmatrix} \quad (6)$$

$$SWAP = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (7)$$

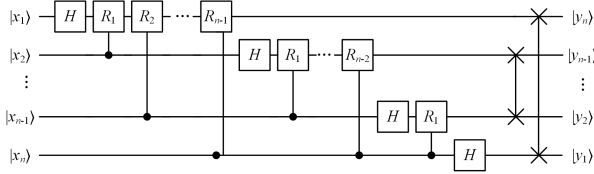


图2 量子傅里叶变换线路图

Fig. 2 Quantum Fourier transform circuit diagram

## 4 模拟实现

使用矩阵乘积态表示量子态时,矩阵乘积态中的每个张量对应一个量子比特,量子门作用在量子比特这一过程通过张量缩并来完成。对量子傅里叶变换进行模拟,需要构建量子态的矩阵乘积态形式,然后根据量子线路完成相应的张量缩并即可。

通过矩阵乘积态,可以有效地对纠缠程度不高的量子态进行模拟,但在模拟过程中也存在一些约束,如二比特门必须作用在两个相邻的量子比特上。此外,为使总体误差最小,在与二比特量子门对应的张量缩并,再分解成两个小张量时,需要以该张量为中心,构造矩阵乘积态的中心正交形式,再根据辅助指标的维数上限对矩阵进行裁剪。本文结合量子傅里叶变换的量子线路的特点,减少不必要的张量缩并以及正交化运算,同时利用 CPU+DCU 的异构编程优势,在 DCU 端实现张量与矩阵的运算,提高模拟的效率。

### 4.1 矩阵乘积态的初始化

在对量子傅里叶变换进行模拟前,首先要构建量子态的矩阵乘积态表示。对于已知振幅的量子态,可以通过对振幅进行若干次奇异值分解来构造其矩阵乘积态表示。对于矩阵乘积态中的各个张量  $A_{s_j a_{j-1} a_j}^{(j)}$ ,合并第一个辅助指标  $a_{j-1}$  与物理指标  $s_j$ ,将其视为一个矩阵  $A_{a_{j-1} s_j, a_j}$ 。奇异值分解得到的左奇异矩阵  $U$  对应当前量子比特,合并右奇异矩阵  $V^\dagger$  和奇异值矩阵  $S$  对应剩下的量子比特。每轮完成奇异值分解后,需要按照设定的辅助指标的维数上限  $\chi$  对  $U$  按列裁剪,对  $V^\dagger$  按行裁剪,使维数始终处于预设的范围内。实现如算法 1 所示。

#### 算法 1 密度矩阵生成矩阵乘积态

Input: (dense, n)

Output: mps

1. // 传输振幅数组到 DCU 端
2. dense, host  $\leftarrow$  amplitudes
3. dense, device  $\leftarrow$  dense, host
4. initialize(mps)
5. // 奇异值分解构造矩阵乘积态
6. for  $i \leftarrow 0$  to length-1 do
7.  $U, S, V^\dagger \leftarrow$  SVD(dense)
8. mps[i]  $\leftarrow$  trim(U)

9. dense  $\leftarrow$  trim(S) @ trim( $V^\dagger$ )

10. reshape(dense, 2m, n/2)

11. mps[length-1]  $\leftarrow$  dense

12. return mps

在初态量子态未知时,根据量子线路的惯例,输入量子态  $|0\rangle^{\otimes n}$ 。量子比特的数目  $n$  和辅助指标的维数上限  $\chi$  决定矩阵乘积态中各个矩阵的形状,输入量子态  $|0\rangle^{\otimes n}$  决定各个矩阵的值。构造矩阵乘积态时,生成  $n$  个相应维度的矩阵,并将其主对角元素的值全部设为 1,对应量子态  $|0\rangle^{\otimes n}$  的一种矩阵乘积态表示。

### 4.2 量子傅里叶变换的线路图分析

使用全振幅模拟时,不存在额外限制,可以直接根据图 2 中量子傅里叶变换的量子线路图进行模拟。但基于矩阵乘积态模拟时,二比特门必须作用在两个相邻的量子比特上,此时使用图 2 中的量子线路。在遇到受控相移门时,需要使用 SWAP 门,将目标比特与控制比特移动到相邻位置。通过计算,使用图 2 的量子线路会额外插入  $\sum_{i=2}^n (i-1)(i-2)$  个 SWAP 门,因此计算开销较大。

本文使用 Fowler 等<sup>[13]</sup>提出的量子傅里叶变换线路,在其基础上合并受控相移门与交换门,构造受控 SR 门。受控 SR 门的量子线路实现如图 3 所示。受控 SR 门的矩阵形式是一个酉矩阵,如式(8)所示,因此可以直接作为量子门使用。修改后的量子傅里叶变换线路图如图 4 所示。

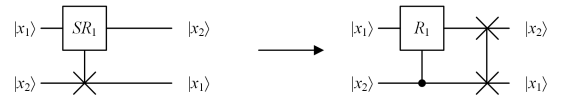


图3 受控 SR 门的量子线路图

Fig. 3 Quantum circuit diagram of controlled-SR gate

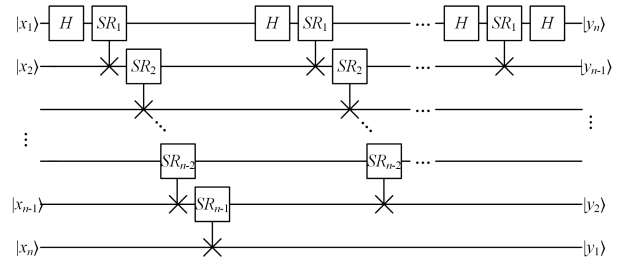


图4 修改的量子傅里叶变换线路图

Fig. 4 Modified quantum Fourier transform circuit diagram

$$cSR = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & e^{2\pi i/2^k} \end{bmatrix} \quad (8)$$

给定初态  $|x_1 x_2 \dots x_n\rangle$ ,按照图 4 所示的量子傅里叶变换线路进行演化,演化过程如式(9)所示。Hadamard 门作用在量子比特  $|x_1\rangle$  后,  $|x_1\rangle$  演化为  $(|0\rangle + e^{2\pi i 0 \cdot x_1} |1\rangle)/\sqrt{2}$  ( $0 \cdot x_1 x_2 \dots x_n$  是二进制小数)。然后通过  $n-1$  个受控 SR 门,量子比特  $|x_1\rangle$  移动到量子线路的最下方,对应的量子态演化为  $(|0\rangle + e^{2\pi i 0 \cdot x_1 x_2 \dots x_n} |1\rangle)$ 。对其余量子比特重复执行这一过程,即可得到按图 4 线路演化后得到的末态。

$$\begin{aligned}
|x_1 x_2 \cdots x_n\rangle &\xrightarrow{H} \frac{1}{\sqrt{2}}(|0\rangle + e^{2\pi i 0 \cdot x_1} |1\rangle) |x_2 \cdots x_n\rangle \\
&\xrightarrow{SR(1)} \frac{1}{\sqrt{2}} |x_2\rangle (|0\rangle + e^{2\pi i 0 \cdot x_1 x_2} |1\rangle) |x_3 \cdots x_n\rangle \\
&\quad \dots \\
&\xrightarrow{SR(n-1)} \frac{1}{\sqrt{2}} |x_2 x_3 \cdots x_n\rangle (|0\rangle + e^{2\pi i 0 \cdot x_1 x_2 \cdots x_n} |1\rangle) \\
&\quad \dots \\
&\xrightarrow{SR(1)} \frac{1}{\sqrt{2^n}} (|0\rangle + e^{2\pi i 0 \cdot x_n} |1\rangle) \otimes (|0\rangle + e^{2\pi i 0 \cdot x_{n-1} x_n} |1\rangle) \otimes \cdots \otimes (|0\rangle + e^{2\pi i 0 \cdot x_2 x_n} |1\rangle) \otimes (|0\rangle + e^{2\pi i 0 \cdot x_1 x_2 \cdots x_n} |1\rangle) \quad (9)
\end{aligned}$$

给定相同初态  $|x_1 x_2 \cdots x_n\rangle$ , 根据图 2 所示的量子傅里叶变换线路进行演化, 演化过程如式(10)所示。对于相同的初态, 通过经典的量子傅里叶变换线路以及修改后的量子傅里叶变换线路得到的末态相同, 因此这两种线路实现等价, 即对应的酉矩阵等价。

$$\begin{aligned}
|x_1 x_2 \cdots x_n\rangle &\xrightarrow{H} \frac{1}{\sqrt{2}}(|0\rangle + e^{2\pi i 0 \cdot x_1} |1\rangle) |x_2 \cdots x_n\rangle \\
&\xrightarrow{R} \frac{1}{\sqrt{2}}(|0\rangle + e^{2\pi i 0 \cdot x_1 x_2} |1\rangle) |x_2 \cdots x_n\rangle \\
&\quad \dots \\
&\xrightarrow{R} \frac{1}{\sqrt{2}}(|0\rangle + e^{2\pi i 0 \cdot x_1 x_2 \cdots x_n} |1\rangle) |x_2 \cdots x_n\rangle \\
&\quad \dots \\
&\xrightarrow{R} \frac{1}{\sqrt{2^n}} (|0\rangle + e^{2\pi i 0 \cdot x_1 x_2 \cdots x_n} |1\rangle) \otimes (|0\rangle + e^{2\pi i 0 \cdot x_2 \cdots x_n} |1\rangle) \otimes \cdots \otimes (|0\rangle + e^{2\pi i 0 \cdot x_{n-1} x_n} |1\rangle) \otimes (|0\rangle + e^{2\pi i 0 \cdot x_n} |1\rangle) \\
&\xrightarrow{SWAP} \frac{1}{\sqrt{2^n}} (|0\rangle + e^{2\pi i 0 \cdot x_n} |1\rangle) \otimes (|0\rangle + e^{2\pi i 0 \cdot x_{n-1} x_n} |1\rangle) \otimes \cdots \otimes (|0\rangle + e^{2\pi i 0 \cdot x_2 x_n} |1\rangle) \otimes (|0\rangle + e^{2\pi i 0 \cdot x_1 x_2 \cdots x_n} |1\rangle) \quad (10)
\end{aligned}$$

修改线路后, 无需额外的 SWAP 门即可满足使用矩阵乘积态模拟时二比特量子门的两个目标比特需要处于相邻位置的要求, 减少了量子线路中二比特量子门的数量。同时量子线路中只有两种量子门, 在模拟初始阶段只需初始化一个 Hadamard 门张量与一个受控 SR 门张量。通过在模拟过程中对受控 SR 张量进行更新, 并对其进行复用, 可以避免反复创建量子门张量带来的性能开销。

### 4.3 量子门的实现

基于全振幅模拟时, 量子门作用在量子态的所有基态上, 对振幅进行变换, 最后汇总振幅生成新的量子态。而使用矩阵乘积态模拟时, 量子门作用在量子比特这一过程则是由缩并量子门和量子比特对应的张量来实现, 如图 5 所示。

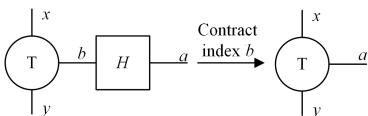


图 5 量子门张量与量子比特张量缩并

Fig. 5 Gate tensor and qubit tensor contraction

本文使用 TTGT 算法, 通过转置将两个张量变为相应的矩阵形式, 其中两个矩阵的列指标和行指标分别对应待缩并的张量指标。然后通过矩阵乘法消去相应指标, 最后通过转置即可得到缩并后的张量。虽然转置操作引入了额外开销, 但进行矩阵乘运算时有更好的局部性, 总体性能会优于直接进行对应下标求和的方法。在对量子傅里叶变换进行模拟的过程中, 主要有两种张量缩并: 1) 目标比特对应的张量与 Hadamard 门对应的张量的缩并; 2) 目标比特与受控比特以及受控 SR 门对应张量的缩并。

在进行与 Hadamard 门相关的张量缩并时, 因为只存在两个张量, 所以缩并顺序唯一。Hadamard 门对应的张量为 2 阶的, 且只作用在第一个量子比特上, 因此无需转置, 可以直接使用 `rocblas_zgemm` 函数来完成缩并。使用爱因斯坦求和约定, 缩并过程如式(11)所示:

$$H_{ab} A_{by} \rightarrow A_{ay} \quad (11)$$

其中, 下标  $y$  为辅助指标。对物理指标  $b$  进行缩并, 从而得到新的张量  $A_{ay}$ , 对应 Hadamard 门作用后的量子比特。

在进行与受控 SR 门相关的张量缩并时, 存在 3 个张量, 对应两种缩并顺序。取辅助指标的维数上限为  $\chi$ 。当优先缩并两个量子比特对应的张量, 再与受控 SR 门对应的张量进行缩并时, 计算总次数为  $x^3 d^2 + x^2 d^4$ ; 而当优先缩并量子比特对应张量与受控 SR 门张量, 再与另一个量子比特对应张量进行缩并时, 计算总次数为  $\chi^4 d^3 + \chi^2 d^4$ 。在实际模拟中  $\chi$  一般大于  $d$ , 因此首先缩并两个量子比特对应张量, 再与受控 SR 门张量进行缩并, 缩并过程如式(12)所示:

$$SR_{abcd} A_{xzy} A_{ydz} \rightarrow SR_{abcd} A_{xcdz} \rightarrow SR_{abcd} A_{cdzx} \rightarrow A_{abzx} \rightarrow A_{abxz} \quad (12)$$

其中, 下标  $x, y, z$  为辅助指标, 对于矩阵乘积态中第一个张量和最后一个张量, 可以将对应的辅助指标  $x$  或  $z$  的维数视为 1。对物理指标  $c, d$  进行缩并, 从而得到新的张量  $A_{abxz}$ , 对应受控 SR 门作用后的量子比特。

### 4.4 中心正交形式构建的优化

二比特门作用后, 两个量子比特对应的低阶张量  $A_{j_j a_{j-1} a_j}^{(j)}$  和  $A_{j_{j+1} a_j a_{j+1}}^{(j+1)}$  缩并成为一个高阶张量  $A_{j_j j_{j+1} a_{j-1} a_{j+1}}$ , 维数升高至  $\chi^2 d^2$ , 需要再分解为两个三阶张量以满足矩阵乘积态形式。分解过程中需要根据预设的辅助指标上限  $\chi$  进行裁剪。为保证裁剪最优, 需要以该张量为中心, 构造矩阵乘积态的中心正交形式。过程如算法 2 所示。

#### 算法 2 构造矩阵乘积态的中心正交形式

Input: (MPS, n, j)

Output: mps

1. // 左正交化
2. for  $i \leftarrow 0$  to  $j-1$  do
3.  $Q, R \leftarrow QR(\text{mps}[i])$
4.  $\text{mps}[i] \leftarrow Q$
5.  $\text{reshape}(\text{mps}[i+1], m/2, 2n)$
6.  $\text{mps}[i+1] \leftarrow R @ \text{mps}[i+1]$
7.  $\text{reshape}(\text{mps}[i+1], 2m, n/2)$
8. // 右正交化
9. for  $i \leftarrow n-1$  to  $j+2$  do

10.  $Q, R \leftarrow QR(\text{mps}[i], T)$
11.  $\text{mps}[i] \leftarrow Q \cdot T$
12.  $\text{mps}[i-1] \leftarrow \text{mps}[i-1] @ R \cdot T$
13. return mps

通过模拟测试,构造矩阵乘积态的中心正交形式的时间开销占整个量子傅里叶变换的时间开销的 90% 左右。为提高算法效率,应对这一步进行优化。

如图 6 所示,在完成量子比特与受控 SR 门的缩并,并进行奇异值分解得到两个三阶张量后,未参与缩并的张量的正交形式得以保留,且奇异值分解得到的第一个张量满足左正交形式。此时再构造矩阵乘积态的中心正交形式时,无需进行额外的 QR 分解。因此,我们在模拟过程中,对各个张量的正交形式进行记录。构造中心正交形式时,满足正交形式的张量直接跳过,直到出现第一个不满足正交形式的张量为止。通过分析优化后,模拟过程中出现的 QR 分解的次数由  $n(n-1)(n-2)/2$  减少为  $(n-1)(n-2)/2$ ,有效提高了模拟的效率。

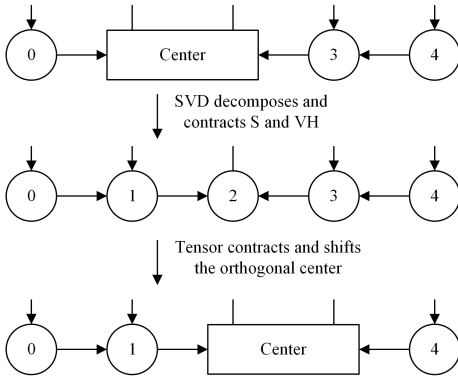


图 6 矩阵乘积态的正交中心转移

Fig. 6 Orthogonal center shifts of matrix product state

## 5 实验验证

在“嵩山”超级计算机系统的单个计算节点上完成基于矩阵乘积态的量子傅里叶变换的模拟。计算核心为海光一号 CPU 与海光一号 DCU。海光一号 DCU 采用类 GPU 架构,精简了逻辑判断、分支跳转等一些控制单元以及拥有数目众多的算术运算单元,适用于如本文中有大量矩阵运算的高度计算密集型的作业。此外,其 Device Memory 为 16GB,足够用来进行模拟与结果验证。

程序使用 HIP-Clang 实现,运行环境为 ROCm 4.1.1。使用 rocBLAS 与 rocSOLVER 库在 DCU 端进行矩阵相乘、矩阵分解等运算。模拟开始时首先在 CPU 端进行一些初始化工作,其余计算在 DCU 端完成,最后将结果传输到 CPU 端,得到模拟后的结果。

### 5.1 正确性验证

分别对 20 位的直积态、GHZ 纠缠态、W 纠缠态、W 纠缠态的扩展,以及随机生成的量子态进行模拟。将通过量子傅里叶变换后的矩阵缩并成振幅向量,保留 20 位有效数字,保存为 CSV 格式。与 CIRQ 量子模拟器进行量子傅里叶变换的结果进行比较,使用二范数验证误差。不同的量子态通过量子傅里叶变换后的误差如表 1 所列。

表 1 模拟误差

Table 1 Simulation error

STATE	CHI	Norm
$\sum_{i=0}^{20}  i\rangle$	2	$2.2308 \times 10^{-7}$
$\sum_{i=0}^{2^{20}-1}  i\rangle$	2	$5.6536 \times 10^{-7}$
GHZ State	2	$6.9016 \times 10^{-7}$
$ \tau\rangle \otimes \sum_{i=0}^{17}  0\rangle$	8	$3.1700 \times 10^{-7}$
20 bits W State	1024	$2.5081 \times 10^{-7}$
Random State	1024	$5.3889 \times 10^{-7}$

矩阵分解使用 rocsolver\_zgesvd 和 rocsolver\_zgeqrf,此外还要使用 rocsolver\_zungqr 来提取 QR 分解中的 Q 矩阵,算法本身会产生误差,此外还会有浮点数误差。使用二范数验证误差,计算该方法经过量子傅里叶变换得到的末态振幅与 CIRQ 模拟器得到的末态振幅的欧氏距离,对于 20 位量子比特的  $2^{20}$  个振幅,总体误差处于  $10^{-7}$  量级。

在对振幅误差进行验证的基础上,继续验证中心正交形式构造的有效性。输入 20 个量子比特的随机量子态,模拟过程中使用不同的辅助指标位数上限  $\chi$ 。对比在模拟过程中构造矩阵乘积态的中心正交形式与不构造中心正交形式输出振幅,使用二范数来验证误差。其结果如图 7 所示。

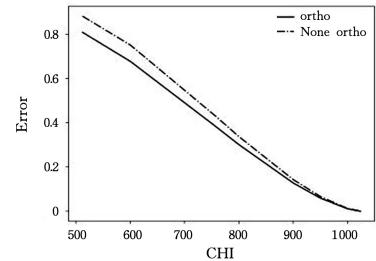


图 7 中心正交形式的有效性验证

Fig. 7 Validation of central orthogonal form

对于内部纠缠结构复杂的 20 位量子态,当辅助指标的维数上限  $\chi$  不足以表示量子态的内部纠缠时,通过构造矩阵乘积态的中心正交形式可以减小模拟的误差。随着  $\chi$  的不断升高,差距不断缩小,直到可以有效地对量子态进行表示。

在对振幅进行验证的基础上,进一步使用半经典 Draper 量子加法器<sup>[14]</sup>对模拟的有效性进行验证。该算法将输入量子态映射到傅里叶空间,根据待加数插入若干相移门,最后使用逆量子傅里叶变换得到结果。输入态概率幅如图 8 所示。

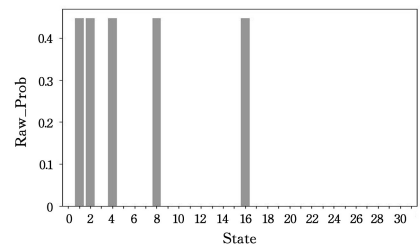


图 8 初态概率幅

Fig. 8 Initial state probability amplitudes

使用半经典 Draper 量子加法器,待加数设置为 3,模拟完成后,末态概率幅如图 9 所示。

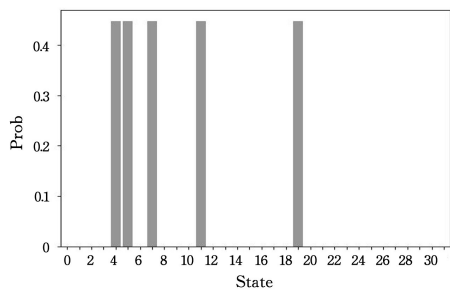


图9 末态概率幅

Fig. 9 Final state probability amplitudes

通过模拟实验,给定初态经过半经典 Draper 加法器后,各个基态的变换与预期一致,增加了3。表明了该模拟的正确性。

## 5.2 功能性测试

### 5.2.1 模拟时间开销

基于矩阵乘积态的量子傅里叶变换模拟的时间开销受限于量子比特的数目与辅助指标的维数上限。量子比特的数目增多,模拟过程中的矩阵运算和张量运算的次数增多。辅助指标的维数增多,模拟过程中的矩阵运算与张量运算的时间开销增加。

固定量子比特的数目为64,选取不同的辅助指标维数上限,模拟时间开销如图10所示,横坐标代表辅助指标维数上限 $\chi$ ,纵坐标代表模拟的时间开销,单位为s。

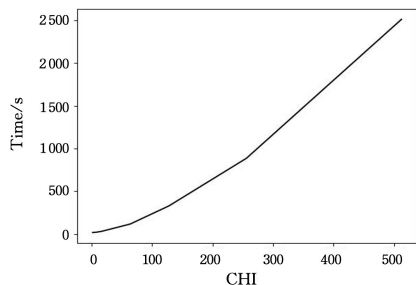


图10 64位量子态模拟时间开销

Fig. 10 Time cost of 64 bit quantum state simulation

固定辅助指标的维数上限 $\chi$ 为2,模拟不同规模的量子态,模拟时间开销如图11所示。横坐标代表模拟的量子比特的数目,纵坐标代表模拟耗时,单位为s。

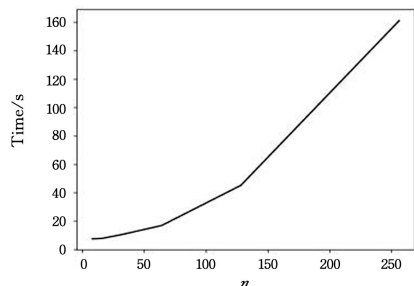


图11 模拟时间开销

Fig. 11 Time cost of simulation

初始阶段,时间开销低且上升趋势较缓。这部分的时间开销主要是因为 rocblas\_handle 句柄的初始化等产生了一些

固定开销。对图10和图11进行分析可以发现,模拟的时间开销主要受限于辅助指标的维数上限,维度上升导致矩阵与张量运算耗时增加。

当量子态内部纠缠程度有限时,与全振幅模拟方法相比,该模拟方法无论是在模拟规模还是模拟时间开销上都有提升。基于全振幅模拟,当模拟的规模达到44个量子比特时,需要2048个计算节点同时运算,且需要花费8542.651s。而当量子态内部纠缠程度较低时,使用矩阵乘积态只需单个计算节点便可进行更大规模的量子傅里叶变换模拟,且时间开销更低。

### 5.2.2 模拟规模分析

当辅助指标 $\chi$ 的值为 $2^{n/2}$ 时,可以准确表示任意的 $n$ 位量子态。此时空间复杂度为 $O(nd2^n)$ ,远大于全振幅模拟方法的空间复杂度 $O(2^n)$ 。当量子态纠缠熵最大时,在“嵩山”超级计算机系统的单个计算节点上进行量子傅里叶变换模拟,使用矩阵乘积态最多只能精确模拟24位的量子态,且模拟时间为1041.45s;而使用全振幅模拟,足以模拟32位量子态。

当量子态内部纠缠有限时,模拟的规模可达数百位量子比特。该模拟方法相较于全振幅模拟有一定局限性,更适合模拟量子纠缠程度较低的量子态。

**结束语** 本文使用矩阵乘积态表示量子态,基于CPU+DCU的异构编程模型对量子傅里叶变换算法进行了模拟。对模拟结果进行分析并证明了模拟的有效性。模拟的最大规模受限于量子比特内部的纠缠结构,对纠缠程度较低的初态进行模拟,可以有效扩大模拟的规模,然而在面对一些纠缠熵较高的量子态时,表现不佳。因此在结合其他量子算法时,需要对量子态的纠缠程度以及允许的误差范围进行分析,为辅助指标的维数上限 $\chi$ 设置合适的值,对这一问题仍需继续研究。

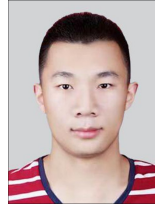
## 参考文献

- [1] SHOR P W. Algorithms for quantum computation: discrete logarithms and factoring[C]// Proceedings 35th Annual Symposium on Foundations of Computer Science. IEEE, 1994: 124-134.
- [2] HARROW A W, HASSIDIM A, LLOYD S. Quantum algorithm for linear systems of equations[J]. Physical Review Letters, 2009, 103(15): 150502.
- [3] NIELSEN M A, CHUANG I L. Quantum computation and quantum information[M]. Cambridge: Cambridge University Press, 2010: 217-221.
- [4] LI X W, FU X, YAN F, et al. Current status and future development of quantum computing research[J]. China Engineering Science, 2022, 24(4): 133-144.
- [5] BROWNE D E. Efficient classical simulation of the quantum Fourier transform[J]. New Journal of Physics, 2007, 9(5): 146.
- [6] XIE J M, HU W F, HAN I, et al. Quantum Fourier Transform Simulation Based on “Songshan” Supercomputer System[J]. Computer Science, 2021, 48(12): 36-42.

- [7] LIU X N, JING L N, WANG L X, et al. Large-scale Quantum Fourier Transform Simulation Based on SW26010[J]. Computer Science, 2020, 47(8): 93-97.
- [8] WOOLFE K J, HILL C D, HOLLENBERG L C L. Scale invariance and efficient classical simulation of the quantum Fourier transform[J]. arXiv:1406.0931, 2014.
- [9] SAITOH A. Fast and Accurate Simulation of Quantum Computing by Multi-Precision Mps: Recent Development[M]// Physics, Mathematics, and All that Quantum Jazz. 2014: 49-69.
- [10] GAWATZ R, BALRAM A. Matrix Product State Based Algorithms[J/OL]. [https://qdev.nbi.ku.dk/student\\_theses/RGawatz\\_Msc.pdf](https://qdev.nbi.ku.dk/student_theses/RGawatz_Msc.pdf).
- [11] EKERT A, KNIGHT P L. Entangled quantum systems and the Schmidt decomposition[J]. American Journal of Physics, 1995, 63(5): 415-423.
- [12] SCHOLLWÖCK U. The density-matrix renormalization group in the age of matrix product states[J]. Annals of Physics, 2011, 326(1): 96-192.
- [13] FOWLER A G, DEVITT S J, HOLLENBERG L C L. Implementation of Shor's algorithm on a linear nearest neighbour qubit array[J]. Quantum Information & Computation, 2004, 4(4): 237-251.
- [14] BEAUREGARD S. Circuit for Shor's algorithm using  $2n+3$  qubits[J]. Quantum Information & Computation, 2003, 3(2): 175-185.



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



**LIAN Demeng**, born in 1999, postgraduate. His main research interests include quantum algorithm and high-performance computing.

(责任编辑:何杨)

## 冯登国院士将出席 CNCC2024 并发表特邀报告

CNCC2024 将于 10 月 24—26 日于浙江省东阳市横店镇举办。CCF 会士, 中国科学院院士, 网络与信息安全领域的杰出专家, 中国科学院软件研究所冯登国研究员受邀出席并将发表特邀报告。

冯登国院士是中国科学院软件研究所研究员, 博士生导师。他在 Theor. Comput. Sci., J. Cryptology, IEEE IT, CCS 等国际重要期刊和会议上发表了 200 多篇论文。

此外, 冯登国院士还是多部重要著作的作者, 包括《网络空间安全概论》、《大数据安全与隐私保护》、《可信计算理论与实践》等。他还荣获了国家科技进步一等奖、国家技术发明二等奖等多项奖励。

冯登国院士还担任过国家 863 计划信息安全技术主题专家组组长、国家信息化专家咨询委员会委员以及国家 973 计划项目首席科学家等, 为我国的科技发展和信息安全事业倾注了大量心血。

此次冯登国院士在 CNCC2024 大会上的特邀报告, 无疑将为与会者带来一场学术盛宴, 我们期待他的精彩分享, 共同探讨网络与信息安全领域的最新进展和未来趋势。

据 CCF 微信公众号

中国计算机学会  
China Computer Federation

**CNCC 2024**  
特邀报告讲者

**冯登国**  
CCF会士, 中国科学院院士  
中国科学院软件研究所研究员

网络与信息安全专家, 中国科学院软件研究所研究员、博士生导师。在 THEOR. COMPUT. SCI., J. CRYPTOLOGY, IEEE IT, CCS 等国际重要期刊和会议上发表学术论文 200 多篇, 主持研制国际和国家标准 20 多项, 出版有《网络空间安全概论》、《大数据安全与隐私保护》、《可信计算理论与实践》等多部著作, 荣获国家科技进步一等奖、国家技术发明二等奖等多项奖励。担任过国家 863 计划信息安全技术主题专家组组长、国家信息化专家咨询委员会委员、国家 973 计划项目首席科学家等。

时间: 2024年10月24日-26日  
地点: 浙江省东阳市横店镇圆明新园