

# 面向分布式超导量子计算架构的量子线路映射\*

朱鹏程<sup>1,2</sup>, 卫丽华<sup>1,2</sup>, 冯世光<sup>2</sup>, 周祥臻<sup>3</sup>, 郑盛根<sup>4</sup>, 管致锦<sup>2</sup>



<sup>1</sup>(宿迁学院 信息工程学院, 江苏 宿迁 223800)

<sup>2</sup>(南通大学 信息科学技术学院, 江苏 南通 226019)

<sup>3</sup>(清华大学 计算机科学与技术系, 北京 100084)

<sup>4</sup>(鹏城实验室, 广东 深圳 518055)

通信作者: 朱鹏程, E-mail: zhupc@squ.edu.cn

**摘要:** 近年来, 超导量子互连技术的研究取得了重要进展, 这为构建分布式超导量子计算架构提供了有效途径. 分布式超导架构在网络拓扑、量子比特连通性、以及量子态传输协议等方面对量子线路的执行施加了严格约束. 为在分布式架构上调度和执行量子线路, 需要通过专门的映射工序对量子线路进行适配底层架构的变换, 并将变换后的线路交由网络中多个 QPU (quantum processing unit) 协同运行. 分布式量子线路映射需向原始线路插入辅助的量子态移动操作, 这些操作 (尤其是 QPU 间量子态移动操作) 具有较高的错误率. 因此, 减少映射所需的量子态移动操作数对于保证分布式计算的成功率至关重要. 基于超导量子互连技术和超导 QPU 的技术特征构建一种抽象的分布式量子计算模型, 并基于该抽象模型提出一种分布式量子线路映射方法, 该方法由量子比特分布式映射和量子态路由两个核心模块组成, 前者以量子态路由开销为代价函数, 通过局部寻优和模拟退火相结合的策略生成近最优的初始映射; 后者根据量子门执行的不同情形构建多个启发式量子态路由策略, 并通过灵活应用这些策略最小化插入的量子态移动操作数. 所构建的分布式抽象模型屏蔽了底层架构中和量子线路映射无关的物理细节, 这使得基于该模型的映射方法可适用于一类分布式超导架构而非某个特定架构. 另外, 所提方法可作为辅助工具参与分布式网络拓扑结构的设计和评价. 实验结果表明, 所提算法可以有效降低映射所需的 QPU 内量子态移动操作 (即 SWAP 门) 数和 QPU 间量子态移动操作 (即 ST 门) 数. 相较已有算法, 在所有基准线路上平均减少 69.69% 的 SWAP 门和 85.88% 的 ST 门, 且时间开销和已有算法接近.

**关键词:** 超导量子计算; 量子网络; 分布式计算; 量子处理器; 量子线路映射

**中图法分类号:** TP303

中文引用格式: 朱鹏程, 卫丽华, 冯世光, 周祥臻, 郑盛根, 管致锦. 面向分布式超导量子计算架构的量子线路映射. 软件学报. <http://www.jos.org.cn/1000-9825/7167.htm>

英文引用格式: Zhu PC, Wei LH, Feng SG, Zhou XZ, Zheng SG, Guan ZJ. Quantum Circuit Mapping for Distributed Superconducting Quantum Computing Architecture. Ruan Jian Xue Bao/Journal of Software (in Chinese). <http://www.jos.org.cn/1000-9825/7167.htm>

## Quantum Circuit Mapping for Distributed Superconducting Quantum Computing Architecture

ZHU Peng-Cheng<sup>1,2</sup>, WEI Li-Hua<sup>1,2</sup>, FENG Shi-Guang<sup>2</sup>, ZHOU Xiang-Zhen<sup>3</sup>, ZHENG Sheng-Gen<sup>4</sup>, GUAN Zhi-Jin<sup>2</sup>

<sup>1</sup>(College of Information Engineering, Suqian University, Suqian 223800, China)

<sup>2</sup>(School of Information Science and Technology, Nantong University, Nantong 226019, China)

<sup>3</sup>(Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China)

<sup>4</sup>(Peng Cheng Laboratory, Shenzhen 518055, China)

\* 基金项目: 国家自然科学基金 (62072259); 江苏省自然科学基金 (BK20221411); 宿迁市科技计划面上项目 (H202117)

收稿时间: 2022-08-29; 修改时间: 2023-09-01, 2023-12-14; 采用时间: 2024-02-07; jos 在线出版时间: 2024-09-04

**Abstract:** In recent years, research on the interconnect technology of superconducting qubits has made important progress, providing an effective way to build a distributed computing architecture for superconducting quantum computers. The distributed superconducting architecture imposes strict constraints on the execution of quantum circuits in terms of network topology, qubit connectivity, and quantum state transfer protocols. To execute and schedule quantum circuits on a distributed architecture, the circuit mapping process is required to transform the quantum circuits to adapt to the underlying architecture and then to distribute the transformed circuits to multiple QPUs. The distributed circuit mapping process necessitates the insertion of additional quantum operations into the original circuit. Such operations, especially the inter-QPU state transfer operations, are susceptible to noise, leading to high error rates. Therefore, minimizing the number of such additional operations inserted by the mapping process is critical to improving the overall computation success rate. This study constructs an abstract model of distributed quantum computing based on the technical features of the interconnect technology of superconducting qubits and today's superconducting QPUs. Moreover, this study proposes a distributed quantum circuit mapping approach based on this abstract model. The proposed approach consists of two main components: the distributed qubit mapping algorithm and the qubit state routing algorithm. The former formulates the problem of distributing qubits to different QPUs as a combinatorial optimization problem and employs simulated annealing enhanced with local search to find the initial mapping that brings the optimal total routing cost. The latter constructs several heuristic qubit routing rules for different scenarios and integrates them systematically to minimize the additional operations inserted by the mapping process. The abstract model shields any technical details of the underlying architecture that are irrelevant to circuit mapping, which makes the mapping method applicable to a class of such networks rather than a specific one. Moreover, the approach proposed in this study can be used as an ancillary tool to design and evaluate the network topology of distributed systems. The experimental results show that, compared to the baseline approach, the proposed approach reduces the number of intra-chip operations (SWAP gates) and inter-chip operations (ST gates) by 69.69% and 85.88% on average, respectively, with a time overhead similar to existing algorithms.

**Key words:** superconducting quantum computing; quantum network; distributed computing; quantum processing unit (QPU); quantum circuit mapping

超导电路<sup>[1]</sup>是目前实现量子计算技术所用的主流技术路线之一。近年来,超导量子技术发展势头迅猛,超导量子处理器 (quantum processing unit, QPU) 包含的量子比特数迅速增加,以 IBM 为例,其最新超导 QPU 已达到 127 个量子比特的规模,并预计在 2023 年达到 1 000 的规模。现有的超导 QPU 已在量子随机线路采样等实验上初步展示了量子计算优势<sup>[2-5]</sup>,但仍难以实现 QEC (quantum error correction) 技术。容错量子计算需要更多的量子比特资源,但由于制冷、控制、布线、噪声抑制、以及芯片制造等多方面的限制,仅通过扩展单个超导芯片内可容纳的量子比特数难以满足容错计算的资源需求<sup>[6]</sup>。在此背景下,分布式计算模式为持续扩展超导量子计算规模提供了另外一种可行途径<sup>[7,8]</sup>。近年来,超导量子互连技术的研究<sup>[9-16]</sup>取得了重要进展,这些研究为超导量子比特间的交互提供了一种短程的量子通信信道。在这类互连技术的协助下,多个处于不同稀释制冷机中的超导 QPU 可以连接在一起,从而形成分布式量子计算架构。分布式架构可以提供超过单个 QPU 容量的量子计算服务,IBM、Google、以及 Rigetti 等公司均在其量子计算发展路线里对分布式架构做了规划。

量子计算任务一般以量子线路的形式表达,在分布式量子计算架构上执行量子线路时,需要将量子线路中包含的量子比特映射至架构中特定 QPU 的物理量子比特上,并以遵循底层物理约束的方式按需在 QPU 间以及 QPU 内移动量子态和执行量子门。该处理过程被称为分布式量子线路映射。单个 QPU 上的量子线路映射问题已被证明是 NP 完全的<sup>[17,18]</sup>,而分布式架构上的量子线路映射不仅要考虑每个 QPU 施加的物理约束,还要同时考虑底层通信机制所施加的物理约束,因此其复杂性更高。虽然目前存在一些分布式量子线路映射的研究<sup>[19-24]</sup>,但是这些研究均面向一种基于量子因特网 (quantum Internet)<sup>[25]</sup>的分布式计算模型。和分布式超导架构所用的互连技术不同,量子因特网使用量子隐形传态 (quantum teleportation) 技术<sup>[26]</sup>在 QPU 间传输数据,该技术主要面向远程量子通信场景,而非量子计算场景,其在传输效率、保真度、以及时延等方面难以达到协同量子计算的要求。另外,在这种模型中,任意两个 QPU 均是直接连通的,且 QPU 内的任意一对量子比特也是直接连通的,这种全连通性在超导量子计算技术下是难以实现的。因此,这种分布式模型过于理想化,既不适用于量子计算场景,也不符合超导量子计算设备和超导量子互连技术的发展现状。相应的,基于该分布式模型的量子线路映射方法无法直接应用于分布式超导量子计算架构。

为在分布式环境下执行量子线路, 映射方法需要引入辅助的量子态移动操作. 在分布式超导量子计算架构中, 一般通过 SWAP 门在 QPU 内移动量子态, 而通过基于量子信道的传输协议<sup>[9-16]</sup>在 QPU 间移动量子态. 现有超导量子计算机也被称为中等规模带噪声量子 (noisy intermediate-scale quantum, NISQ) 计算设备, 这类设备具备以下特征: 易受噪声影响、量子比特连通性受限、量子态相干时间短、以及量子操作错误高. 分布式映射所需的量子态移动操作, 尤其是 QPU 间量子态移动操作, 是错误率较高的量子操作, 在分布式映射过程中过多引入这两种操作会大幅降低量子线路的执行成功率. 本文面向分布式超导量子计算架构, 以降低插入的量子操作数为目标, 提出了一种分布式量子线路映射方法, 该方法包含两个关键组件: 量子比特分布式映射算法和量子态路由算法. 本文的主要贡献包括以下 3 个方面.

(1) 基于超导 QPU 和超导量子互连的特征, 构建了分布式超导量子计算架构的抽象模型. 该模型屏蔽了底层量子计算和通信技术细节, 并考虑了 QPU 间和 QPU 内量子态移动的代价差异, 从而为分布式量子线路映射研究和评价提供了一个硬件无关的平台.

(2) 基于分布式架构抽象模型, 对分布式量子比特线路映射问题作了形式化描述.

(3) 以减少总体路由代价为目标, 基于模拟退火和局部寻优策略提出了一种分布式量子比特映射算法.

(4) 以减少 SWAP 门数为目标, 构建了 QPU 内量子态路由策略; 以减少 ST (state transfer) 操作数为目标, 构建了 QPU 间量子态路由策略; 基于上述两种路由策略, 提出了一种启发式量子态路由算法.

## 1 相关工作

本节将简单介绍超导量子互连技术的相关进展, 并简要回顾分布式量子线路映射的相关研究工作.

### 1.1 超导量子网络研究进展

近年来, 超导量子比特的短距互连技术取得了重要的实验进展. 文献 [9,10] 提出了基于飞行光子的超导量子比特互连协议; 文献 [11-13] 提出了一种基于驻波模式的超导量子互连协议; 文献 [14-16] 提出了一种基于倒装芯片 (flip-chip) 的互连架构. 这些研究均为连接不同超导 QPU 提供了短距量子信道, 可在 QPU 间实现 70%-90% 的量子态传输保真度. 与以远距离通信为目的的量子隐形传态技术<sup>[26]</sup>不同, 这些超导互连技术在进行量子态传输时无需消耗额外的量子纠缠对, 也无需量子测量和经典通信信道的辅助, 在节约软硬件资源的同时更提供了较高的传输效率和保真度. 另外, 基于超导互连技术的量子态传输在时延上和 QPU 内双比特量子门的执行时长处于同一量级, 且传输保真度正随着技术迭代正逐渐接近双比特量子门. 这些研究为超导量子计算规模的进一步扩展指明了方向, 文献 [27] 通过仿真实验初步验证了这种分布式量子计算架构是构建大规模超导量子计算机的可行方案, IBM 和 Google 已将这种分布式模型作为进一步扩展量子计算平台的候选方案<sup>[28,29]</sup>.

### 1.2 量子线路映射及分布式映射

在超导 QPU 中, 量子比特间的连通性通常是受限的, 在特定排列模式 (如二维网格) 下, 每个量子比特仅允许和近邻量子比特直接交互, 即双比特量子门操作仅允许作用在一对近邻量子比特上, 这种约束被称为近邻交互约束. 然而, 量子线路在设计时并未考虑该约束, 双比特量子门允许作用在任意一对量子比特上, 这使得量子线路通常无法和超导 QPU 直接兼容. 量子线路映射通过插入辅助量子门的形式将原始线路转换为功能等价的物理可执行线路. 近年来, 单个超导 QPU 上的量子线路映射方法已得到广泛研究<sup>[30-41]</sup>. 目前, 量子线路映射方法也已成为量子计算工业软件 (如 IBM 的 Qiskit<sup>[42]</sup>、本源量子的 QPanda 以及百度的量易伏) 的必备模块之一.

分布式量子线路映射是指在分布式量子计算架构上的线路映射问题, 相较于单 QPU 上的映射问题而言, 其在执行量子门时要同时考虑 QPU 以及底层通信机制所施加的物理约束, 因此较单 QPU 上的映射问题更复杂. 虽然近期出现了一些分布式量子线路映射方法<sup>[19-24]</sup>, 但这些方法均面向一种过于理想化的分布式计算模型. 该模型以量子隐形传态技术作为通信机制, 形成了一个 QPU 间全连通的量子计算网络, 并假定任意时刻可用的量子纠缠对数量不受限. 另外, 该模型没有考虑 QPU 内的物理约束, 比如可用的量子比特数量和具体分布情况以及超导 QPU 上普遍存在的近邻交互约束. 这种理想化模型在通信机制、网络拓扑以及 QPU 特性等方面和超导量子计算的发展

现状存在巨大差异,因此现有方法无法直接应用于分布式超导量子计算架构上的分布式量子线路映射。

## 2 系统模型及问题定义

在本节中,我们基于低温超导量子局域网和超导 QPU 的特点抽象出一种可扩展的分布式计算模型,并基于该模型对量子线路映射问题作形式化描述。

### 2.1 分布式超导量子计算模型

量子比特之间的受限连通性催生了对量子线路映射的需求。在分布式架构上,这种受限连通性包括两个方面:一是 QPU 内量子比特的受限连通性,即双比特量子门仅允许作用在通过耦合总线直接相连的两个量子比特上;二是 QPU 间的受限连通性,即跨 QPU 的量子态传输仅允许发生在一对通过量子信道互连的 QPU 之间。虽然目前超导量子比特的互连技术存在多种实现方案<sup>[9-16]</sup>,这些方案在技术细节上各有区别,但它们在逻辑层面功能相似,即通过特定的量子信道将量子态从信源比特传输至另外的信宿比特。另外,在这些方案下量子态传输的保真度和时延均较为接近。上述特点为基于这些互连技术构建分布式超导量子计算的抽象模型奠定了基础。该抽象模型的形式化描述如公式(1)所示,其包括3个要素:网络拓扑  $TPG$ 、QPU 的耦合图  $CG$  以及 QPU 内/间量子态移动操作集合  $MV$ ,其中,  $TPG$  是由各 QPU 作为顶点、量子信道作为连边的无向图,  $CG$  是 QPU 的量子比特耦合图。

$$DM = (TPG = (M, E_{\text{channel}}, W_{\text{channel}}), CG = (V, E_{\text{couple}}, W_{\text{couple}}), MV) \quad (1)$$

分布式模型的网络拓扑和 QPU 耦合图可以合并为一个总的量子比特连通图,一个示例如图1所示。该示例包含4个 QPU:  $M_0$ 、 $M_1$ 、 $M_2$  和  $M_3$ ,它们通过量子信道(图中的折线边)相连,构成了一个二维晶格型网络拓扑结构。虚线方框内的子图是各 QPU 的量子比特耦合图, QPU 内的量子比特同样按二维晶格结构排列,相互之间的连线表示量子比特间的耦合总线,双比特量子门仅允许作用在 QPU 内直接相连的一对量子比特上。将和量子信道相连的量子比特称为通信量子比特(虚线边框节点),而其余量子比特称为数据量子比特(实线边框节点)。和基于隐形传态的量子因特网技术不同,在分布式超导计算技术下通信量子比特和数据量子比特在物理实现方式上类似,因此通信量子比特也可像数据量子比特一样用于执行量子门。此外,规定每个通信量子比特仅与一个数据量子比特直接相连,这种做法有利于屏蔽量子信道中的噪声对于 QPU 内部的干扰<sup>[27]</sup>。需特别说明的是,图1仅给出了分布式模型的一个示例,实际上网络拓扑结构以及 QPU 内部的耦合结构可采用任意图模型。本文所提分布式量子线路映射方法对于类似的分布式模型具有普适性,而非仅局限于图1所示模型。

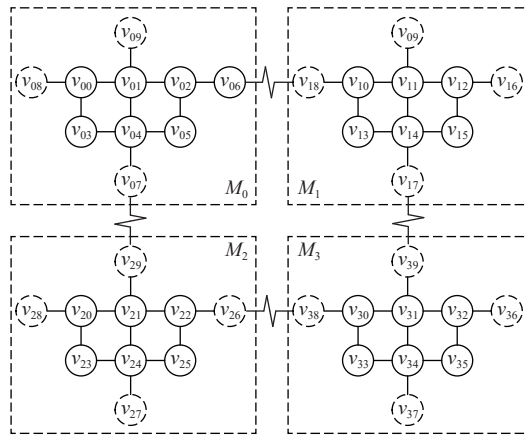


图1 分布式超导量子计算架构的量子比特连通图

不失一般性,假定分布式模型中所有 QPU 均支持 Clifford+T 门库<sup>[43]</sup>,该门库是一个常用的通用量子门库,包括若干单比特量子门和唯一的双比特量子门——CNOT 门(其电路符号及功能如图2(a)所示)。在 Clifford+T 门库的支持下, QPU 内量子态传输可通过 SWAP 门实现,该门的电路符号如图2(c)所示,其用于交换两个量子比特的

量子态, 可通过 3 个级联的 CNOT 门实现, 如图 2(d) 所示. 另外, 使用 ST 门作为基于各种不同超导互连方案的量子态传输操作的抽象表示, 其电路符号和逻辑功能如图 2(b) 所示, 图中的箭头表示量子态的传输方向, 即箭头从信源量子比特指向宿量子比特. ST 门仅允许作用在由量子信道直接相连的一对通信量子比特上, 并且由于现有的超导互连技术<sup>[9-16]</sup>普遍要求在传输前将信宿量子比特初始化为  $|0\rangle$  态, 因此 ST 门同样应遵循该约束, 即要求信宿量子比特的输入态须为  $|0\rangle$  态. 综上, 公式 (1) 中的量子态传输操作集合  $MV = \{\text{SWAP 门}, \text{ST 门}\}$ .

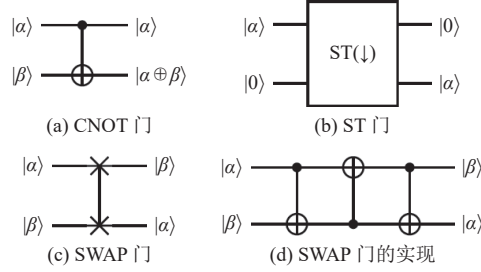


图 2 实现 QPU 内/间量子态移动的量子门

ST 门相较 CNOT 门具有更严重的错误倾向, 以 IBM 的 QPU 为例, 其 CNOT 门的平均错误率在  $10^{-2}$ – $10^{-3}$  量级之间, 而 ST 门的平均错误率在  $10^{-1}$  和  $10^{-2}$  的量级之间<sup>[9-16]</sup>, 约为 CNOT 门错误率的 10 倍. 因此, 就错误率而言, QPU 间/内的量子态移动具有不同的代价. 为保证量子线路的执行成功率, 分布式量子线路映射方法应能感知 ST 门和 CNOT 门在错误率上的差异. 为此, 通过为分布式量子比特连通图中的每条边赋予相应的权值来体现这种错误率差异. 具体而言, 将 QPU 内连边 (即  $E_{\text{couple}}$ ) 的权值设置为 1, 而 QPU 间连边 (即  $E_{\text{channel}}$ ) 的设置 10. 需说明的是, 1 和 10 的权重设置仅是根据目前工艺现状给出的一个粗略估计, 随着 CNOT 门和 ST 门实现工艺的改进, 权重可做相应的调整, 本文所提分布式映射方法并不局限于特定的边权设置.

在为分布式量子比特连通图的边添加权重信息后, 该图可通过公式 (2) 描述, 其中顶点集是各 QPU 中量子比特集合的并集, 而边集/边权集合是  $TPG$  和  $CG$  中边集/边权集合的并集. 该描述中包含了网络拓扑、量子比特耦合结构、以及路由代价等重要信息, 将作为分布式量子线路映射方法的输入. 本文后续部分将分布式量子比特连通图简称为分布式连通图.

$$DQC = \left( V_D = \bigcup_{i=0}^N V_i, E_D = E_{\text{channel}} \cup E_{\text{couple}}, W_D = W_{\text{channel}} \cup W_{\text{couple}} \right) \quad (2)$$

## 2.2 分布式量子线路映射问题

在本文所提分布式模型中, 双比特量子门仅允许作用在同一 QPU 内的一对近邻物理量子比特上, 即满足近邻交互约束. 虽然存在协议 (gate teleportation)<sup>[44]</sup> 允许双比特量子门作用在位于不同 QPU 的两个量子比特上, 但是这种协议和量子隐形传态协议类似, 需要量子纠缠对、测量操作、以及经典信道的辅助, 因此本文暂不考虑这种双比特量子门的远程执行协议.

分布式量子线路映射方法主要包含两个处理步骤, 分别是量子比特分布式映射和量子态路由. 在分别介绍这个步骤之前, 首先给出几个相关概念. 为便于说明, 本文将待执行的量子线路称为逻辑线路, 使用  $LC = (Q, G)$  表示, 其中  $Q$  表示由逻辑线路中所有量子比特构成的集合, 将这些量子比特称为逻辑量子比特,  $G$  表示逻辑线路中的量子门集合; 另外, 将分布式架构中的量子比特称为物理量子比特, 并将经映射过程转换的结果线路称为物理线路, 并使用  $PC = (V, G')$  表示, 其中  $V$  为线路所用的物理量子比特集合,  $G'$  为量子门集合, 包括逻辑线路中的量子门以及在映射过程中插入的 SWAP 门和 ST 门. 本文使用了两种模型表示量子线路, 分别是线路模型和 DAG (directed acyclic graph, 有向无环图) 模型. 图 3 给出了一个量子线路的线路模型及其对应的 DAG 模型, DAG 模型中每个结点代表线路中的一个量子门, 结点之间的有向边定义了量子门的执行优先级, 某个量子门只有在其所有前驱门都已被执行的情况下才可被调度. 根据 DAG 模型, 可以将量子线路分成若干层, 在原始 DAG 模型中没有前驱结点

的门构成第 1 层子线路  $L_0$ , 将第 1 层中所有量子门从 DAG 模型中删除得到新的 DAG 模型, 新模型中无前驱的量子门构成第 2 层子线路  $L_1$ , 类似的可进一步得到  $L_2$ 、 $L_3$  等子线路层, 逻辑线路的最大层数也称为线路深度. 需说明的是, 本文给出的示例线路主要用于说明分布式映射问题, 由于单比特量子门不受近邻交互约束制约, 因此我们在示例线路中忽略了单比特量子门.

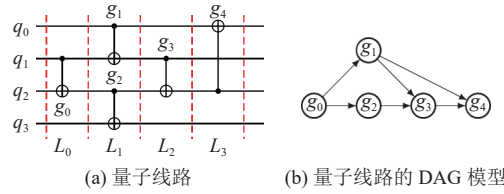


图 3 逻辑线路及其 DAG 表示

分布式量子线路映射的第 1 步, 即量子比特分布式映射, 用于建立逻辑量子比特到物理量子比特的映射关系. 逻辑量子比特数应小于或等于可用的物理量子比特数, 但通过向逻辑线路中增加额外的空量子比特, 总可以使得逻辑量子比特数等于物理量子比特数. 因此, 为便于说明, 假定两者数目相等. 定义 1 中给出了规模为  $n$  的量子比特映射关系的数学描述, 其中的物理量子比特集合包括了来自分布式模型中各 QPU 的全部可用量子比特.

**定义 1.** 量子比特映射关系. 给定逻辑量子比特集合  $Q = \{q_0, q_1, \dots, q_{n-1}\}$  和物理量子位集合  $V_D = \{v_0, v_1, \dots, v_{n-1}\}$ , 量子比特映射是一个双射函数  $\pi: Q \rightarrow V_D$ , 满足  $\pi(q_i) = \pi(q_j)$  当且仅当  $q_i = q_j$ .

量子比特映射  $\pi$  可表示为如公式 (3) 所示的置换形式, 其中第 1 行是逻辑量子比特序号, 而第 2 行是与逻辑量子比特对应的物理量子比特序号.

$$\pi = \begin{pmatrix} 0 & 1 & \dots & n-1 \\ j_0 & j_1 & \dots & j_{n-1} \end{pmatrix}, \text{ 其中, } 0 \leq j_0, j_1, \dots, j_{n-1} < n \quad (3)$$

在公式 (3) 中,  $\pi(i) = j_i$  表示在映射  $\pi$  下逻辑量子比特  $q_i$  将被分配给物理量子比特  $v_{j_i}$ ,  $0 \leq i < n$ . 公式 (3) 还可简写为如下的单行形式:

$$\pi = \{v_{j_0}, v_{j_1}, \dots, v_{j_{n-1}}\} \quad (4)$$

将最初的量子比特映射关系称为初始映射. 一旦给定初始映射, 便可以开始分布式量子线路映射的第 2 步, 即量子态路由. 量子态路由按照 DAG 定义的依赖关系依次读取量子门, 若遇到不满足近邻交互约束的双比特量子门, 则通过插入 SWAP 门或 ST 门对该门相关的两个量子态 (即逻辑量子比特) 进行路由, 直至这两个量子态移动至同一 QPU 上的两个近邻物理量子比特上. 在进行量子态路由时, 本文将逻辑量子比特和量子态视为两个可相互替换的等价概念. 为便于说明, 根据是否满足近邻交互约束以及是否可立即调度, 对量子门的不同类型做了如下定义.

**定义 2.** 活跃门. 若逻辑线路中的一个量子门在该线路所对应 DAG 模型中无前驱节点 (即入度为 0), 则称该门为活跃门.

一个量子门是否是活跃门和量子比特映射关系无关, 只取决于在其所属 DAG 模型中的位置, 图 3 中的  $g_0$  是唯一的活跃门.

**定义 3.** 局域门和非局域门. 给定分布式连通图和量子比特映射关系, 若在该映射关系下, 一个量子门所作用的物理量子比特均位于同一个 QPU 内, 则称该门为局域门, 否则称之为非局域门.

根据定义 3, 单比特量子门总是局域门, 而双比特量子门则根据其对应的物理量子比特分布情况可为局域门或非局域门.

**定义 4.** 可执行门. 给定逻辑线路、分布式连通图以及量子比特映射关系, 若一个量子门既是活跃门又是局域门, 并同时满足近邻交互约束, 则称该门为可执行门.

给定配置下局域门、非局域门以及可执行门的一个示例如例 1 所示.

例 1: 给定如图 1 所示的分布式连通图和如图 3 所示的逻辑线路, 若当前量子比特映射关系为  $\pi: \{v_{02}, v_{10}, v_{11}, v_{12}\}$  下, 即  $\pi(q_0) = v_{02}$ ,  $\pi(q_1) = v_{10}$ ,  $\pi(q_2) = v_{11}$ ,  $\pi(q_3) = v_{12}$ , 则逻辑线路中的局域门有  $g_0, g_2$  和  $g_3$ , 而非局域门有  $g_1$  和  $g_4$ . 另外  $g_0$  满足近邻交互约束, 因是当前配置下线路中唯一的可执行门.

量子态路由按照依赖关系依次读取逻辑线路中的量子门, 并通过插入 SWAP 门/ST 门依次将不满足近邻交互关系的活跃门转换为可执行门, 当且仅当一个量子门是可执行门时, 才可以对其进行调度. 最终, 逻辑量子比特映射至的物理量子比特、量子态路由涉及的物理量子比特、逻辑线路中的各量子门以及插入的 SWAP 门和 ST 门共同构成满足架构约束的物理线路.

例 1 续: 在初始映射关系  $\pi: \{v_{02}, v_{10}, v_{11}, v_{12}\}$  下, 量子态路由向原始线路中插入了 2 个 SWAP 门和 1 个 ST 门, 得到如图 4 所示的物理线路.

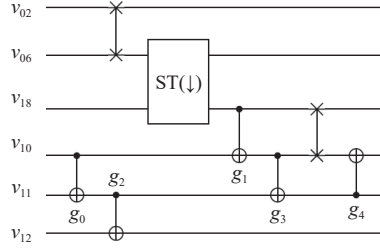


图 4 满足交互约束的物理线路

图 4 仅是给出了当前配置下物理线路的一种可能结果, 实际上不同的初始映射或量子态路由策略将导致不同 SWAP 门和 ST 门插入方式. 为保量子证线路调度后的执行成功率, 分布式量子线路映射以最小化插入的 ST 门数和 SWAP 门数为优化目标, 在当前 ST 门错误率偏高的情况下, 减少 ST 门数尤为重要. 综上, 我们将分布式量子线路映射问题的作如下形式化定义.

**定义 5.** 分布式量子线路映射. 给定逻辑线路  $LC=(Q, G)$ 、分布式连通图  $DQC=(V_D, E_D, W_D)$  以及正整数  $k$  和  $i$ , 求在初始映射  $\pi$  下, 能否通过插入最多  $k$  个 SWAP 门和  $i$  个 ST 门使得  $LC$  中的任意双比特量子门均满足近邻交互约束.

### 3 分布式量子比特映射算法

本节将重点介绍分布式量子比特映射算法, 该方法以最小化总体路由代价为目标, 基于模拟退火和局部搜索生成初始量子比特映射关系.

#### 3.1 总体路由代价函数

在分布式连通图  $DQC=(V_D, E_D, W_D)$  上, 两个物理量子比特间的最短加权路径直接反映了将两者量子态近邻化所需的 ST 门数和 SWAP 门数, 其中, 最短路径中权重为 10 的边数等于所需的 ST 门数, 权重为 1 的边数等于所需的 SWAP 门数减 1. 因此, 我们将两个物理量子比特间的加权路径长度定义为近邻化一个双比特量子门的所需量子态路由代价.

**定义 6.** 量子态路由代价. 给定分布式连通图  $DQC=(V_D, E_D, W_D)$ , 设  $v_i$  和  $v_j$  是两个不同的物理量子比特, 即,  $v_i, v_j \in V_D$  且  $i \neq j$ , 则  $v_i$  和  $v_j$  之间的量子态路由代价等于两者在  $DQC$  上的最短加权路径长度.

设  $rcost[][]$  为  $DQC$  的路由代价矩阵, 其中  $rcost[v_i][v_j]$  表示  $v_i$  和  $v_j$  之间的量子态路由代价. 则在映射关系  $\pi: Q \rightarrow V_D$  下, 给定一个双比特量子门  $g(q_i, q_j)$ , 近邻化该门所需的量子态路由代价  $rcost[\pi(q_i)][\pi(q_j)]$ . 据此可进一步给出逻辑线路在特定映射关系下的总体量子态路由代价.

**定义 7.** 逻辑线路的总体量子态路由代价. 给定逻辑线路  $LC(Q, G)$  和分布式连通图  $DQC=(V_D, E_D, W_D)$ , 在映射关系  $\pi: Q \rightarrow V_D$  下,  $LC$  的总体量子态路由代价等于近邻化全部双比特量子门所需的路由代价之和.

根据定义 7, 量子线路  $LC$  的总体路由代价如公式 (5) 所示, 其中,  $count(g(q_i, q_j))$  用于统计  $LC$  中作用在  $q_i$  和  $q_j$  上的双比特量子门总数.

$$tcost(LC, DQC, \pi) = \sum_{(q_i, q_j) \in Q^2} count(g(q_i, q_j)) \cdot rcost[\pi(q_i)][\pi(q_j)] \quad (5)$$

公式 (5) 的值越小, 表明  $\pi$  越有利于将逻辑线路中频繁交互的逻辑量子比特对聚集在相同 QPU 上.

### 3.2 量子比特分布式映射算法

量子比特分布式映射方法的主要任务便是在所有可能的量子比特映射关系中找到使公式 (5) 取最小值的  $\pi^*$ , 如公式 (6) 所示. 其中  $n$  表示量子比特数目,  $S_n$  表示由所有可能量子比特映射关系构成的集合, 由于每个映射关系可等价的视为一个  $n$  元素的全排列, 则  $S_n$  可视为一个  $n$  阶置换群.

$$\pi^* = \operatorname{arg\,mint}_{\pi \in S_n} tcost(LC, DQC, \pi) \quad (6)$$

公式 (6) 定义的优化问题具有  $O(n!)$  规模的可行解空间, 因此很难通过暴力枚举法获得其中的最优初始映射. 在这种情况下, 可以借助元启发式算法在求解速度和精度上取得较好的平衡. 因此, 我们以模拟退火<sup>[45]</sup>为算法框架, 并通过最速下降法对退火过程进行强化, 从而提出了一种量子比特分布式映射方法.

模拟退火<sup>[45]</sup>是一种常用的元启发式算法, 通过模拟固体退火过程, 在问题状态空间内进行寻优搜索, 其可以概率的跳出局部最优并最终趋近于全局最优. 然而, 模拟退火算法探索问题状态空间的方式是完全随机的, 其在得到某个局部最优解之前通常会遍历大量较差的解, 在迭代次数不够的情况下, 算法最终获得的解会存在质量不佳或不稳定的情况. 为加速模拟退火趋近局域最优解的过程, 我们引入了一种基于最速下降的启发式算法. 该启发式算法的基本思想是从一个给定的映射关系  $\pi$  出发, 以公式 (5) 为代价函数, 通过最速下降的方式不断交换  $\pi$  中的两个不同元素, 使得频繁交互的逻辑量子比特逐渐“聚拢”至同一 QPU, 并最终生成量子比特映射的一个局部最优解. 算法 1 给出了该算法的详细过程, 其以对换运算  $p_{ij}$  (交换  $\pi$  中的第  $i$  和第  $j$  个元素) 作为邻域算子对当前解  $\pi$  的邻域结构进行探索, 在邻域结构中搜索最小化公式 (5) 的对换操作  $p_{uv}$ , 并用  $p_{uv}$  更新当前解  $\pi$ . 算法 1 将重复该过程直至  $\pi$  为可达邻域内的局部最优解.

---

#### 算法 1. 量子比特映射局部最优解生成算法.

---

输入: 映射关系  $\pi$ , 逻辑线路  $LC$ , 分布式模型  $DQC$ ;

输出: 可达邻域内的局部最优解  $\pi^*$ .

---

1.  $loc\_opt = \text{FALSE}$ ; //初始化循环控制条件
  2. **WHILE** not  $loc\_opt$  **DO** //循环至找到局部最优解
  3.  $\Delta max = 0$ ; //  $\Delta max$  记录代价函数值的最大降幅
  4. **FOR** each possible  $p_{ij}$  **DO** //  $p_{ij}$  表示交换  $\pi$  的第  $i$  和第  $j$  个元素
  5.  $\Delta = tcost(\pi) - tcost(\pi \oplus p_{ij})$ ; //计算应  $p_{ij}$  后公式 (5) 的降幅
  6. **IF**  $\Delta > \Delta max$  **THEN** //判断是否更新  $\Delta max$
  7.  $\Delta max = \Delta, u = i, v = j$ ;
  8. **END IF**
  9. **END FOR**
  10. **IF**  $\Delta max > 0$  **THEN** //如果  $\Delta max$  发生更新
  11.  $\pi = \pi \oplus p_{uv}$ ; //更新当前  $\pi$
  12. **ELSE** //否则将  $loc\_opt$  设置为 TRUE
  13.  $loc\_opt = \text{TRUE}$ ;
  14. **END IF**
  15. **END WHILE**
-



算法 1 从一个任意给定的可行解出发, 可迅速找到其可达邻域空间内的局部最优解. 在算法 1 的每次循环中, 共需为  $O(n^2)$  个对换操作计算公式 (5) 的值, 而计算公式 (5) 的时间复杂度为  $O(n)$ , 且算法 1 最多循环  $O(n^2)$  次, 因此算法 1 的时间复杂度为  $O(n^5)$ .

算法 2 使用模拟退火过程不断探索和跳出由算法 1 产生的量子比特映射局部最优解, 从而逐步逼近全局最优. 算法 2 采用了一种非均匀退火控制机制<sup>[40]</sup>, 相关控制参数如表 1 所示, 其中  $\Delta_{min}$  和  $\Delta_{avg}$  分别表示退火过程中目标函数差值  $\Delta$  的最小值和平均值, 在实际设定时我们通过将算法 2 的总迭代次数  $L$  设定为 100 对这两个参数取值进行估计.

表 1 算法 2 模拟退火相关参数设定

参数	参数设定
总迭代次数	$L=1000$
初始温度	$t_0=0.3\Delta_{min}+0.7\Delta_{avg}$
终止温度	$t_1=0.7\Delta_{min}+0.3\Delta_{avg}$
温度衰减系数	$\beta=(t_0-t_f)/Lt_0t_f$
降温函数	$t=t/(1+\beta t)$

### 算法 2. 量子比特分布式映射算法.

输入: 逻辑线路  $LC$ , 分布式连通图  $DQC$ ;

输出: 量子比特映射关系  $\pi^*$ , 即物理量子比特的排列.

```

1. set_parameters(); //根据表 1 设置参数  $L, t_0, t_1, \beta$ 
2.  $t=t_0$ ; //将温度初始化为  $t_0$ 
3.  $\pi=\pi^*=randomize()$ ; //随机初始化  $\pi$  和  $\pi^*$ 
4. FOR  $i=1$  to  $L$  DO //最多迭代  $L$  次
5.    $\pi_1=perturb(\pi)$ ; //对  $\pi$  作扰动得到  $\pi_1$ , 扰动由连续多次随机对换组成
6.    $\pi_1=local\_search(\pi_1)$ ; //调用算法 1, 从  $\pi_1$  出发寻找局部最优解
7.    $\Delta=tcost(\pi_1)-tcost(\pi_0)$ ; //计算  $\pi_1$  和  $\pi_0$  在代价函数上的差值
8.   IF  $\Delta < 0$  THEN
9.     accept=TRUE; //接受  $\pi_1$ 
10.  ELSE IF  $rand() < e^{-\Delta/t}$  THEN
11.    accept=TRUE; //否则, 以  $e^{-\Delta/t}$  的概率接受  $\pi_1$ 
12.  ELSE
13.    accept=FALSE; //拒绝  $\pi_1$ 
14.  END IF
15.  IF accept THEN
16.     $\pi_0=\pi_1$ ; //将  $\pi_0$  更新为  $\pi_1$ 
17.     $\pi^*=(tcost(\pi_0) < tcost(\pi^*) ? \pi_0, \pi^*)$ ; //更新目前为止的最优解  $\pi^*$ 
18.  END IF
19.    $t=t/(1+\beta t)$ ; //降低温度
20. END FOR

```

算法 2 重点考虑了量子线路中量子比特的整体交互结构, 但一般而言量子线路的不同子部分会表现出不同的交互特性. 由于量子线路的执行具有严格的时序性, 在构建初始映射时重点关注量子线路中最先执行的部分子线路要比考虑整体线路更具实际意义. 因此, 在实验中我们以包含前  $10n$  个双比特量子门的子线路作为算法 2 的输

入. 算法 2 共需  $L$  次循环, 在每次循环内, 对算法 1 的调用是最耗时的操作, 因此算法 2 的时间复杂度为  $O(L \cdot n^5)$ .

#### 4 分布式量子态路由算法

本节将重点介绍构成分布式量子态路由算法, 包括 QPU 内量子态路由策略、QPU 间量子态路由策略以及量子态路由整体策略.

##### 4.1 QPU 内量子态路由策略

QPU 内量子态路由策略以最小化线路映射所需的 SWAP 门数为主要目标, 在分布式量子线路映射场景中, 存在 3 种情形需要在 QPU 内部移动逻辑量子比特 (量子态). 第 1 种情形是, 在将某 QPU 上的一个逻辑量子比特/量子态传输至另外一个 QPU 时, 该逻辑量子比特尚未处于通信量子比特上, 此时需要将该逻辑量子比特交换至同一 QPU 上的通信量子比特, 将这种情形对应的量子态路由策略简称为  $IQR_1$ ; 第 2 种情形是, 某 QPU 上的一个通信量子比特即将作为 ST 门的目标量子比特, 即接受从其他 QPU 传输而来的量子态, 但是该量子比特已被某个逻辑量子比特占用, 此时为执行 ST 门需将某个空闲量子比特的量子态 ( $|0\rangle$  态) 交换至该通信量子比特, 将这种情形对应的量子态路由策略简称为  $IQR_2$ ; 第 3 种情形是, 量子线路中当前活跃门是不满足近邻交互约束的局域门, 此时为了执行这些局域门, 需要将其相关的两个逻辑量子比特移动至 QPU 内的近邻位置上, 将该情形对应的量子态路由策略简称为  $IQR_3$ .

$IQR_1$  的实现相对简单, 首先找到在当前映射关系下该逻辑量子比特对应的物理量子比特, 然后沿着该物理量子比特和相应通信量子比特之间的最短路径插入 SWAP 门, 即可将该逻辑量子比特交换至特定的通信量子比特, 如例 2 所示.

例 2: 给定如图 1 所示的分布式连通图和量子比特映射关系  $\pi$ , 并假定  $\pi(q) = v_{18}$ , 即逻辑量子比特  $q$  位于计算节点  $M_1$  的物理量子比特  $v_{11}$  上, 此时若要将  $q$  从  $M_1$  传输至  $M_0$ , 则首先要将  $q$  从  $v_{11}$  移至通信量子比特  $v_{18}$ , 由于  $v_{11}$  到  $v_{18}$  间的最短路径为:  $v_{11} \rightarrow v_{10} \rightarrow v_{18}$ , 则插入 SWAP 门序列:  $\{\text{SWAP}(v_{11}, v_{10}), \text{SWAP}(v_{10}, v_{18})\}$  便可实现  $q$  的移动.

$IQR_2$  要求当前 QPU 内至少存在一个空闲物理量子比特, 即至少存在一个尚未关联逻辑量子比特的物理量子比特, 我们首先在这些空闲量子比特中找到距离通信量子比特最近的那个, 然后沿着该空闲量子比特和通信量子比特之间的最短路径插入 SWAP 门, 从而将空闲量子比特上的  $|0\rangle$  态移动到通信量子比特, 如例 3 所示.

例 3: 给定如图 1 所示的分布式连通图, 假设即将执行  $\text{ST}(v_{18}, v_{06})$ , 但是  $v_{06}$  被某个逻辑量子比特占用, 若此时  $M_0$  上未被逻辑量子占用的空闲量子比特有  $\{v_{05}, v_{07}, v_{08}\}$ , 其中  $v_{05}$  距离  $v_{06}$  最近, 它们之间的最短路径为:  $v_{05} \rightarrow v_{02} \rightarrow v_{06}$ , 则通过插入以下 SWAP 门序列:  $\{\text{SWAP}(v_{05}, v_{02}), \text{SWAP}(v_{02}, v_{06})\}$ , 可以将  $|0\rangle$  态从  $v_{05}$  交换至  $v_{06}$ , 从而解除  $v_{06}$  的被占用状态, 为执行 ST 门做好准备.

$IQR_3$  用于将待调度的局域门转换成满足近邻交互约束的可执行门, 与  $IQR_1$  和  $IQR_2$  不同,  $IQR_3$  不是一次性插入多个 SWAP 门, 而是每次决策仅插入一个 SWAP 门, 通过多次决策使得所有考虑的局域门转换成可执行门, 其单次执行步骤如下所示: (1) 根据所考虑的局域门集合, 生成候选 SWAP 门集合  $SW = \{\text{SWAP}(v_1, v_2) | v_1 \text{ 或 } v_2 \text{ 至少和一个局域门相关}\}$ ; (2) 为集合  $SW$  中的每个 SWAP 门计算效应函数值 (如公式 (7) 所示), 并选择效应函数值最大的 SWAP 门, 在出现平局情况时随机选取一个效应值最大的 SWAP 门; (3) 返回选中的 SWAP 门, 并相应的更新映射关系. 公式 (7) 用于量化在映射关系  $\pi$  下  $\text{SWAP}(v_1, v_2)$  对映射后续  $K+1$  层子线路所需路由代价的影响. 其中,  $\Delta rcost(g, \pi)$  用于计算在插入 SWAP 门后量子门  $g$  的路由代价减少值, 如公式 (8) 所示,  $\tau$  表示应用 SWAP 门后得到新的映射关系,  $\Delta rcost$  的最大取值为 1, 最小取值为 -1.

$$sw\_eff(v_1, v_2, \pi) = \sum_{g \in L_0} \Delta rcost(g, \pi) + \omega \cdot \frac{1}{K} \sum_{k=1}^K \sum_{g \in L_k} \Delta rcost(g, \pi) \quad (7)$$

$$\Delta rcost(g(q_i, q_j), \pi) = rcost[\pi(q_i)][\pi(q_j)] - rcost[\tau(q_i)][\tau(q_j)] \quad (8)$$

公式 (7) 不仅考虑了即将调度的  $L_0$  层子线路, 还同时考虑了后续  $K$  层子线路, 其等式右边第 1 部分量化了

SWAP 门对于  $L_0$  层子线路中所有量子门的路由代价影响效应, 而第 2 部分量化了对后续  $K$  层线路平均每层的路由代价影响效应, 其中是  $\omega$  一个小于 1 的权重系数, 该权重系数体现了量子线路调度的优先级, 即  $L_0$  层优先. 公式 (7) 的值越大, 则表示该 SWAP 门越有利于将后续线路层中的局域门转换为可执行门. 公式 (7) 中的路由代价矩阵  $rcost[][]$  可以预先计算, 对于包含  $n$  量子比特的量子线路, 每一层最多有  $n/2$  个量子门, 因此计算公式 (7) 需耗时  $O(n)$ .

例 4 给出了根据  $IQR_3$  选择和插入一个 SWAP 门的示例.

例 4: 给定一个包含 4 个量子比特和 3 个量子门的逻辑线路:  $\{g(q_0, q_1), g(q_2, q_3), g(q_1, q_2)\}$ 、如图 1 所示的分布式模型以及量子比特映射关系  $\pi: \{v_{00}, v_{02}, v_{01}, v_{05}\}$ , 设  $K=1, \omega=0.2$ , 则在当前配置下, 待调度的活跃门有两个:  $g(q_0, q_1)$  和  $g(q_2, q_3)$ , 它们均是不满足近邻交互约束的局域门; 构造的 SWAP 门候选集合共包括 9 个 SWAP 门, 如  $SWAP(v_{00}, v_{01})$ 、 $SWAP(v_{00}, v_{03})$  等, 其中每个 SWAP 门都和活跃门至少共用一个物理量子比特; 根据公式 (7) 计算每个 SWAP 门的效应值, 如  $SWAP(v_{00}, v_{01})$  对应的值为  $-0.2$ ,  $SWAP(v_{01}, v_{02})$  对应的值为 2; 其中  $SWAP(v_{01}, v_{02})$  的效应值最大, 因此将该 SWAP 门插入物理线路, 并更新  $\pi$  得到  $\tau$ , 即  $\{v_{00}, v_{01}, v_{02}, v_{05}\}$ , 在映射关系  $\tau$  下, 两个活跃门均为可执行门.

## 4.2 QPU 间量子态路由策略

非局域门相关的两个量子态 (即逻辑量子比特) 分别位于分布式网络中两个不同 QPU 之上, 为了执行非局域门, 需要首先将该门相关的量子态移动至同一 QPU 上, 即将该非局域门转换成局域门. ST 门用于在相邻 QPU 之间传输量子态, 所插入的 ST 门数是评价分布式量子线路映射的一个重要指标, 而 QPU 间量子态路由策略以最小化线路映射所需 ST 门数为主要目标. 为专注于量子态在 QPU 间的流动, 在分布式架构的网络拓扑图上分析和求解量子态路由问题. 图 5 给出了如图 1 所示分布式连通图对应的网络拓扑, 相较分布式连通图, 网络拓扑图忽略了各 QPU 内部的连接结构, 而仅保留 QPU 间的连接结构, 图 5 中的每个结点均代表一个 QPU, 每条边代表连接两个不同 QPU 的量子信道, 权重均为 1. 此外, 我们还定义了一个用于在相邻 QPU 间传输量子态的宏操作  $MT(q, M_s, M_t)$ , 其中  $q$  表示待传输的量子态,  $M_s$  表示信源 QPU, 而  $M_t$  表示信宿 QPU. 宏操作  $MT$  可分解成由多个 SWAP 门和唯一 ST 门构成的量子操作序列, 在具体实现  $MT$  时, 首先根据  $IQR_1$  将  $q$  移动至相应通信量子比特, 然后按需根据  $IQR_2$  清空  $M_t$  上的待用通信量子比特, 最后通过 ST 门将  $q$  移至  $M_t$  上的通信量子比特, 如例 5 所示.

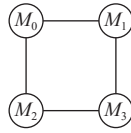


图 5 分布式网络拓扑模型

例 5: 给定量子比特映射关系  $\pi$ , 并假定  $\pi(q)=v_{11}$ , 即逻辑量子比特  $q$  位于计算节点  $M_1$  的物理量子比特  $v_{11}$  上, 且  $M_0$  上的  $v_{06}$  已被某个逻辑量子比特占用, 此时  $M_0$  上未被逻辑量子占用的空闲量子比特仅有  $v_{05}$ , 则  $MT(q, M_1, M_0)$  可分解为:  $\{SWAP(v_{11}, v_{10}), SWAP(v_{10}, v_{18}), SWAP(v_{05}, v_{02}), SWAP(v_{02}, v_{06}), ST(v_{18}, v_{06})\}$ , 其中前 3 个 SWAP 门根据  $IQR_1$  插入, 第 4 和第 5 个 SWAP 门根据  $IQR_2$  插入.

为评价  $MT$  宏操作的作用效应, 构建了如公式 (9) 所示的代价函数. 公式 (9) 中的  $\Delta n_{lcoal}$  表示插入该  $MT$  操作前后  $K+1$  层子线路中新增加的局域门数量; 而  $\Delta n_{lcoal}$  后的子公式在形式上和公式 (7) 类似, 仅是将分布式连通图的代价矩阵  $rcost[][]$  替换为网络拓扑图上的代价矩阵  $ncost[][]$ , 这分子公式用于计算  $MT$  操作带来的在网络拓扑图上的路由代价变化, 其在  $\Delta n_{lcoal}$  相等的情况下, 为选择  $MT$  操作提供进一步的依据. 其中的  $\Delta ncost(g, \pi)$  表示应用  $MT$  操作后量子门  $g$  在网络拓扑图上的路由代价降幅, 如公式 (10) 所示,  $\tau$  表示应用  $MT$  门更新当前映射后  $\pi$  得到新的映射. 公式 (9) 的值越大, 则表示该  $MT$  操作越有利于将后续线路层中的非局域门转换为局域门. 和公式 (7) 类似, 计算一次公式 (9) 需耗时  $O(n)$ .

$$mt\_eff(q, M_1, M_2, \pi) = \Delta n_{local} + \sum_{g \in L_0} \Delta n_{cost}(g, \pi) + \omega \cdot \frac{1}{K} \sum_{k=1}^K \sum_{g \in L_k} \Delta n_{cost}(g, \pi) \quad (9)$$

$$\Delta n_{cost}(g(q_i, q_j), \pi) = n_{cost}[\pi(q_i)][\pi(q_j)] - n_{cost}[\tau(q_i)][\tau(q_j)] \quad (10)$$

对于 QPU 间量子态路由而言, QPU 的实际容量, 即 QPU 上的可用物理量子比特总数, 同样是须重点考虑的约束之一. 当一个 QPU 上的所有物理量子比特均被逻辑量子比特占用, 即该 QPU 的容量已满时, 为向该 QPU 传入一个逻辑量子比特, 首先要从其中移出一个逻辑量子比特.

QPU 间量子态路由策略 CQR 如图 6 所示, 第 1 步中的候选 MT 操作集合  $MTs = \{(q, A, B) | q \text{ 为和非局域门相关的逻辑量子比特, } A \text{ 为 } q \text{ 所在的 QPU, } B \text{ 为网络中和 } A \text{ 相邻的 QPU}\}$ ; 在第 2 步中, 根据公式 (9) 在  $MTs$  中选择效应值最大的  $MT(q, A, B)$  操作, 若出现平局情况, 则从效应值最大的  $MT$  操作中随机选取一个; 第 3 和 4 步在  $B$  容量已满的情况下确定网络中距  $B$  最近的 QPU 以及之间的最短路径  $path$ ; 第 6-9 步构成的循环沿着  $path$  每次将一个逻辑量子比特从当前节点  $S$  移动到  $path$  上的下一个节点  $T$ , 其中, 第 7 步中的候选  $MT$  操作集合  $MT_0 = \{(p, S, T) | p \text{ 为节点 } S \text{ 上的逻辑量子比特, } T \text{ 为 } S \text{ 在 } path \text{ 上的下一个节点}\}$ , 第 8 步根据公式 (9) 在  $MT_0$  中选择效应值最大的  $MT$  操作, 若出现平局情况, 则随机选择一个效应值最大的  $MT$  操作; 第 10 步将一次 CQR 调用过程中所选的  $MT$  操作序列逆序返回.

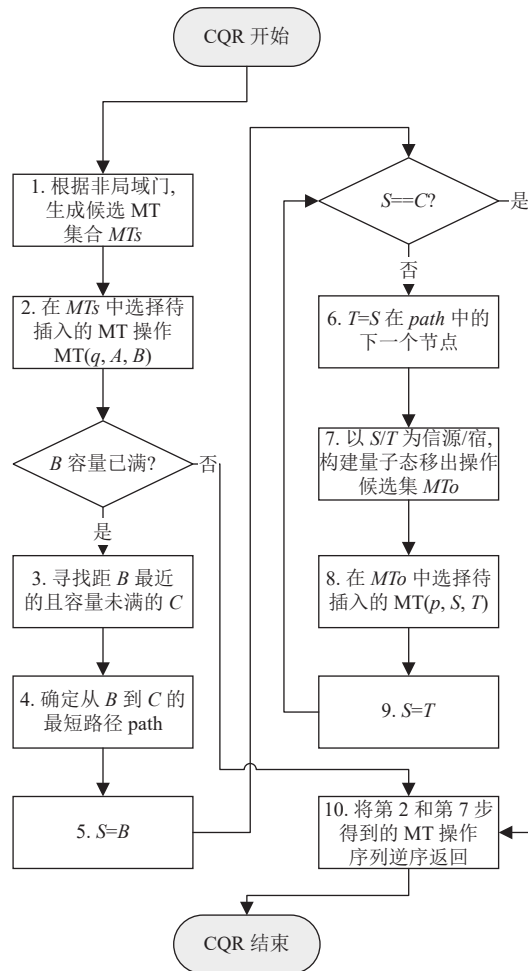


图 6 QPU 间量子态路由策略 CQR

设分布式模型共有  $n$  个量子比特总数, 其中每个 QPU 均包含  $s$  个量子比特, 则模型中共包含  $n/s$  个 QPU, 并构成一个  $\sqrt{n/s} \cdot \sqrt{n/s}$  的二维网格型网络拓扑. CQR 的第 1 步最多包含  $4n$  个候选 MT 操作, 因此第 2 步共需  $O(n^2)$  时间确定待插入的 MT 操作; 第 3 步需要  $O(\sqrt{n/s})$  时间; 由于网络拓扑中所有 QPU 间的最短路径可以预先确定, 因此第 4 步确定 *path* 所需时间为  $O(1)$ , 而 *path* 长度为  $O(\sqrt{n/s})$ , 即第 6–9 步的循环次数最多为  $O(\sqrt{n/s})$ ; 第 7 步 *MT<sub>o</sub>* 最多包含  $s$  个候选 MT 操作, 因此第 8 步需耗时  $O(s \cdot n)$ ; 则第 6–9 步的循环共需耗时  $O(s \cdot n \cdot \sqrt{n/s})$ ; 综上, 一次调用 CQR 的时间复杂度为  $O(n^2 + n^{1.5} \cdot s^{0.5})$ , 由于  $s < n$ , CQR 的渐进时间复杂度可记为  $O(n^2)$ .

#### 4.3 总体路由策略及方法

本文提出的量子态路由方法严格遵循 DAG 模型定义的优先级关系依次读取和处理量子线路中的活跃门集合, 直至量子线路中的所有量子门均成为满足物理约束的可执行门. 在当前映射关系下, 活跃门集合中的门可分为 3 类: 可执行门、不可执行的局域门以及非局域门. 对于处于活跃态的这 3 类门, 量子态路由方法优先处理可执行门, 即, 将可执行门从逻辑线路中删除, 并将其加入物理线路; 然后根据 QPU 间量子态路由策略依次插入 MT 宏操作尽可能地将非局域门转换成局域门, 进而根据 QPU 内量子态路由策略依次插入 SWAP 门直至将局域门转换成可执行门.

量子态总体路由策略及方法如算法 3 所示, 其中的第 6–10 步根据 CQR 策略对非局域门作转换, 需补充说明的是, 为防止 CQR 策略生成的 MT 操作将原先的局域门转换为非局域门, 在调用 CQR 时将冻结局域门相关的量子态, 即禁止移动这些量子态的 MT 操作成为 CQR 策略用到的候选操作. 但量子态的冻结可能会导致在某些极端情况找不到候选 MT 操作对非局域门作进一步转换, 当检测到这种情况时, 我们会提前退出第 6–10 步的循环, 继而进入处理局域门的环节.

---

#### 算法 3. 分布式量子态路由算法.

---

输入: 逻辑线路  $LC$ , 分布式模型  $DQC$ , 初始映射关系  $\pi$ ;

输出: 物理线路  $PC$ .

---

1.  $dag = gen\_DAG(LC)$ ; //为  $LC$  生成 DAG 模型
  2. **WHILE**  $dag \neq NULL$  **DO** //循环直至 DAG 为空
  3.  $acgs = get\_active(dag)$ ; //读取  $dag$  中的活动门集合
  4.  $(exg, loc, nloc) = partition\_act(acgs, \pi, DM)$ ; //将  $acgs$  中的门分为 3 类: 可执行门  $exg$ , 局域门  $loc$ , 和非局域门  $nloc$
  5.  $dag.remove(exg)$ ;  $PC.append(exg)$ ; //从  $dag$  中删除所有可执行门, 并将这些门加入物理线路
  6. **WHILE**  $nloc \neq NULL$  **DO** //使用 CQR 策略将非局域门转换成局域门
  7.  $mtops = cqr(nloc)$ ; //调用一次 CQR 策略
  8.  $PC.append(mtops.decompose())$ ; //将 CQR 策略返回的 MT 宏操作分解成 SWAP 门和 ST 门, 并加入物理线路
  9.  $\pi = \pi \oplus mtops$ ; //根据 CQR 返回的 MT 操作更新映射关系
  10. **END WHILE**
  11.  $swaps = iqr3(loc)$ ; //使用  $IQR_3$  策略处理局域门
  12.  $PC.append(swaps)$ ; //将  $IQR_3$  策略返回的 SWAP 门插入物理线路
  13.  $\pi = \pi \oplus swaps$ ; //根据  $IQR_3$  策略返回的 SWAP 门更新映射关系
  14. **END WHILE**
- 

给定量子线路  $LC(Q, G)$ , 设分布式模型共有  $n$  个量子比特总数, 且每个 QPU 均包含  $s$  个量子比特, 并构成一个  $\sqrt{n/s} \cdot \sqrt{n/s}$  的二维网络拓扑. 在最坏情况下, 算法 3 遍历的每个量子门均是非局域门, 网络拓扑中两个 QPU 间的最大距离为  $O(\sqrt{n/s})$  的量级, 则为将一个非局域门转换成局域门, 最多需要调用 CQR 策略  $O(\sqrt{n/s})$  次; 另外, 若假定 QPU 内量子比特同样按方形网格排列, 则两个物理量子比特的最大距离约为  $O(\sqrt{s})$  量级, 则将一个局域门转

换成可执行门最多需调用  $IQR_3$  策略  $O(\sqrt{s})$  次; 每次调用 CQR 或  $IQR_3$  的时间复杂度为  $O(n^2)$ , 且  $LC$  中共有  $|G|$  个门, 则算法 3 最坏时间复杂度为  $O((\sqrt{n/s} + \sqrt{s}) \cdot n^2 \cdot |G|)$ , 在本文中  $s=10$ , 则算法 3 时间复杂度约为  $O(|G| \cdot n^{2.5})$ .

另外, 算法 3 返回的物理线路还可作如下优化. 作为算法 3 输入之一的初始映射关系由算法 2 生成, 该算法在生成映射关系时综合考虑了位于量子线路左部的子线路, 而不仅是即将处理的第  $L_0$  层子线路. 因此, 在该初始映射下通常在  $L_0$  层子线路中存在不可执行的量子门, 为将这些门转换成可执行门, 需要插入若干 ST 门或 SWAP 门, 而这些插入的辅助门不依赖于原始线路中的任何量子门, 因此可直接将这些辅助门更新后的映射关系当作初始映射, 并将这些门从物理线路中删除, 如例 6 所示.

例 6: 给定由例 1 生成的物理线路, 如图 4 所示. 该物理线路最左边的  $SWAP(v_{02}, v_{06})$  和  $ST(v_{06}, v_{18})$  将映射关系由最初的  $\pi: \{v_{02}, v_{10}, v_{11}, v_{12}\}$  更新为  $\{v_{18}, v_{10}, v_{11}, v_{12}\}$ , 且这两个插入的辅助门并不依赖于原始量子线路中的任何量子门, 因此将初始映射直接设定为  $\pi: \{v_{18}, v_{10}, v_{11}, v_{12}\}$ , 并从物理线路中删除这两个辅助量子门.

## 5 实验与结果

本节通过基准量子线路库上的实验对本文所提分布式量子线路映射方法所需的路由代价 (即 ST 门数和 SWAP 门数) 以及时间性能作评价.

### 5.1 实验配置

本文实验采用了一种被广泛用于量子线路映射评价的基准线路库<sup>[30-38]</sup>, 该基准库中包含了百余个量子线路, 每个量子线路均由 Clifford+T 门组成, CNOT 门是其中唯一的双比特量子门, 量子线路的规模从几十量子门到数十万量子门不等, 线路包含的量子比特数目从 5 到 16 个不等, 为了便于测试分布式映射算法的性能, 我们从中选取了量子比特数目超过 10 的部分线路进行测试. 另外, 所有实验除特别说明外均以如图 1 所示的分布式连通图作为映射的目标量子平台.

为便于评价本文算法的总体性能, 通过对现有基于假想模型的分布式量子线路映射方法<sup>[21,22]</sup>的改造和适配构造了一种基线算法 (简称 DQP). 文献 [21,22] 所提分布式映射方法的基本工作过程如下: 首先, 以最小化非局域门数量为目标, 将逻辑量子比特划分成若干不相交的子集, 每个子集对应网络中的一个 QPU; 然后, 按序依次调度量子线路中的量子门, 当遇到一个非局域门时, 通过隐形传态技术将该门的一个量子态移至另一个量子态所在的 QPU 上, 并在执行完这个门后, 立刻将该量子态移回原来位置, 本文将这种量子态传输策略称为往返传输策略. 这些方法以隐形传态次数作为首要评价指标, 并假定网络中的 QPU 两两相互连通且每个 QPU 内的量子比特也是全连通的, 此外在跨 QPU 传送量子态时未考虑 QPU 的容量问题. 为了匹配本文的分布式模型, 我们对 DQP 算法作了如下适配: (1) 在划分逻辑量子比特时, 要求每个子集的元素个数最多不超过 QPU 内的数据量子比特个数, 并将每一个子集中的逻辑量子比特随机映射到对应 QPU 内的数据量子比特上, 如此不仅可以建立完善的量子比特映射关系, 还使得通信量子比特处于空闲态, 从而确保使用量子态往返传输策略时 QPU 总有剩余空间; (2) 在网络中两个物理量子比特间传输量子态时, 以 ST 门和 SWAP 门组成的序列取代隐形传态, 即沿着两个量子比特间的最短路径, 通过 ST 门和 SWAP 门实现量子态的往返传输. 关于 DQP 算法进一步的知识请查阅文献 [21,22].

在实验中, 本文提出的分布式量子线路映射方法 (简称 DQM) 在生成初始映射时 (算法 2), 仅将逻辑量子比特映射到 QPU 内的数据量子比特上, 如此不仅和 DQP 算法的做法保持一致, 还可以使得量子线路分布到更多 QPU 上, 更有利于评价分布式映射算法的路由代价. 另外, 在实验中, 公式 (7) 和公式 (9) 中的权重因子  $\omega$  均等于 0.2, 前瞻层数  $K$  除特别说明外取值为 5; 算法 2 中总迭代次数  $L$  设定为 1000 次.

本文所提算法使用 Python 语言实现, 并使用 Qiskit 作为读取基准量子线路的辅助工具. 所有实验运行的硬件环境配置如下: Intel i7-4710HQ CPU, 8 GB 内存.

### 5.2 算法总体性能评价

首先将 DQP 算法和 DQM 算法进行比较, 从而检验 DQM 算法在路由开销以及时间开销等方面的性能, 实验结果如表 2 所示. 表 2 的前 4 列分别给出了基准量子线路的相关信息, 包括序号、线路名称、量子比特数以及包

含的 CNOT 门数; 第 5–10 列分别给出了两种算法在各基准线路上的运行结果, 包括插入的 SWAP 门数 (#SWAP)、ST 门数 (#ST) 以及以 s 为单位的运行时间 (time); 最后两列分别给出了 DQM 算法在 SWAP 门和 ST 门上相较于 DQP 算法的减少幅度. 如表 2 所示, DQM 在 SWAP 门数和 ST 门数上远优于 DQP, 在所有基准线路上, DQM 所需的 SWAP 门较 DQP 最多减少 89.39%, 平均减少 69.69%; 类似的, DQM 所需的 ST 门最多减少 94.59%, 平均减少 85.88%.

表 2 DQP 算法和 DQM (本文) 算法的性能比较

No.	基准线路		DQP				DQM			优化幅度 (%)	
	name	#n	#g	#SWAP	#ST	time (s)	#SWAP	#ST	time (s)	#SWAP	#ST
1	qft_10	10	90	298	96	6.38	77	26	6.81	74.16	72.92
2	sys6-v0_111	10	98	184	58	5.71	64	4	5.88	65.22	93.10
3	rd73_140	10	104	158	74	5.63	73	4	5.64	53.80	94.59
4	sym6_316	14	123	208	78	11.93	112	14	14.98	46.15	82.05
5	rd53_311	13	124	267	90	10.88	108	18	11.30	59.55	80.00
6	rd84_142	15	154	329	106	13.70	130	12	14.36	60.49	88.68
7	ising_model_10	10	90	20	20	2.74	15	9	2.78	25.00	55.00
8	qft_16	16	240	1463	432	15.30	354	105	16.48	75.80	75.69
9	ising_model_16	16	150	57	40	5.86	50	19	5.03	12.28	52.50
10	wim_266	11	427	832	296	7.76	323	47	7.42	61.18	84.12
11	cm42a_207	14	771	1692	550	10.65	608	66	10.26	64.07	88.00
12	pm1_249	14	771	1951	550	9.78	671	115	8.68	65.61	79.09
13	dc1_220	11	833	2845	1144	6.89	671	88	6.99	76.41	92.31
14	sqrt8_260	12	1314	3728	1198	10.48	1101	130	11.17	70.47	89.15
15	radd_250	13	1405	4332	1482	13.00	1267	219	14.17	70.75	85.22
16	adr4_197	13	1498	4684	1644	15.13	1332	226	11.60	71.56	86.25
17	misex1_241	15	2100	7556	2744	20.75	1815	312	24.73	75.98	88.63
18	rd73_252	10	2319	6876	2370	7.29	1811	296	8.83	73.66	87.51
19	cycle10_2_110	12	2648	9700	2928	10.55	2302	395	14.37	76.27	86.51
20	square_root_7	15	3089	9335	2752	16.94	2753	370	20.10	70.51	86.56
21	sqn_258	10	4459	13829	4918	9.01	3369	398	9.45	75.64	91.91
22	inc_237	16	4636	14579	5486	37.82	3846	608	37.80	73.62	88.92
23	rd84_253	12	5960	22785	7336	14.58	4970	717	15.02	78.19	90.23
24	root_255	13	7493	29689	10244	16.76	5919	694	17.26	80.06	93.23
25	co14_215	15	7840	28291	8552	26.39	6640	857	24.63	76.53	89.98
26	mlp4_245	16	8232	28963	9264	24.00	7013	1020	28.83	75.79	88.99
27	sym9_148	10	9408	27995	10036	9.70	6765	904	11.99	75.83	90.99
28	life_238	11	9800	33688	11192	12.24	7916	1073	12.22	76.50	90.41
29	max46_240	10	11844	34640	10676	10.99	9580	1470	12.07	72.34	86.23
30	clip_206	14	14772	61854	21818	21.67	12198	1538	23.95	80.28	92.95
31	9symml_195	11	15232	54676	18304	12.92	12570	1708	16.44	77.01	90.67
32	sym9_193	11	15232	51570	15972	13.69	12959	1979	16.54	74.87	87.61
33	dist_223	13	16624	60240	19478	20.19	14073	1986	19.97	76.64	89.80
34	sym10_262	12	28084	93147	28756	19.07	24017	3630	23.13	74.22	87.38
35	urf3_279	10	60380	195298	66474	26.23	50700	8662	33.17	74.04	86.97
36	plus63mod4096_163	13	56329	222417	72346	30.97	48818	6875	40.40	78.05	90.50
37	urf6_160	15	75180	325841	119700	48.70	72316	13962	56.80	77.81	88.34
38	hwb9_119	10	90955	306500	103090	37.51	72358	9637	43.95	76.39	90.65
39	ground_state_estimation_10	13	154209	420548	130214	82.88	44609	23977	80.05	89.39	81.59
40	urf4_187	11	224028	715546	240168	87.22	174385	24469	111.86	75.63	89.81

DQM 方法在 SWAP 门和 ST 门上取得的优势主要源于以下几点原因: (1) DQM 在映射逻辑量子比特时同时考虑了量子线路结构、分布式系统的网络拓扑以及 QPU 内量子比特的排列方式, 而 DQP 仅考虑了量子线路本身

的结构,网络拓扑和物理量子比特排列信息的缺失,使得 DQP 生成的初始量子比特映射质量低于 DQM; (2) DQP 在移动量子态时所用的往返传输策略虽然简单易实现,但在优化 SWAP 门和 ST 门数上存在很大局限性,相反, DQM 所用的 IQP 和 CQP 策略能在多个候选操作中选择最有利于降低路由代价的 SWAP 门或 ST 门,从而减少所需的 SWAP 门和 ST 门总数.此外, DQM 和 DQP 在所有基准线路上所需的 SWAP 门数均普遍多于 ST 门数,这主要是由于无论是将非局域门转换成局域门还是将局域门转换成可执行门均需插入 SWAP 门,而 ST 门仅在转换非局域门时用到.在时间开销方面, DQM 和 DQP 均是多项式时间复杂度算法,且在进行分布式量子比特映射时均采用了多迭代的算法模型,因此两者的时间开销相近.

### 5.3 QPU 内量子态路由策略的性能评价

面向单个 QPU 芯片的量子线路映射在近几年得到了大量关注和研究,但由于底层架构的巨大差异,这些已有方法无法直接适用于分布式场景.然而,仅将这些方法用于实现分布式映射场景下 QPU 内的量子态路由是显然可行的.本文通过将 Qiskit<sup>[42]</sup>中的两种先进量子线路映射方法 (LookaheadSwap 和 SabreSwap) 分别集成至本文所提的分布式量子态路由算法 (见算法 3),实现了另外两种版本的分布式路由算法.所用集成方式如下:当算法 3 执行到 QPU 内量子态路由时 (算法 3 第 11 行),从当前量子线路为架构中的各 QPU 分别读取作用其上的由连续局域门组成的最大待调度子线路,并调用特定映射方法 (LookaheadSwap 或 SabreSwap) 完成各局域子线路向特定 QPU 的映射任务. LookaheadSwap 方法和文献 [36] 类似,通过对映射决策树的深度搜索选择最有利于线路执行的 SWAP 门,以下将集成 LookaheadSwap 的分布式量子态路由方法简称为 DQM\_L 算法; SabreSwap 方法<sup>[31]</sup>采用了一种前瞻性启发式搜索规则,通过计算候选 SWAP 门的映射代价函数值,选择待插入的 SWAP 门,以下将包含 SabreSwap 的分布式量子态路由方法简称为 DQM\_S 算法.

为评价 QPU 内量子态路由策略,在所有基准测试线路上对 DQM、DQM\_L 和 DQM\_S 分别进行测试.公平起见,3 种算法在每个基准线路上的实验均采用相同的初始映射:

$$\pi = \{v_{ij} \text{ for } i \text{ in } [0, 3] \text{ for } j \text{ in } [0, 5]\} \quad (11)$$

即,为量子线路中的各逻辑比特按以下顺序依次分配物理比特:  $\{v_{00}, v_{01}, v_{02}, v_{03}, v_{04}, v_{05}, v_{10}, v_{11}, v_{12}, \dots\}$ .

图 7 给出了 3 种算法在所有基准线路 (见表 2) 上所需的平均 SWAP 门数、ST 门数和平均用时,其中左边的纵轴用于统计门数,而右边的纵轴表示时间 (以 s 为单位).从图 7 可知,3 种算法所需的平均 ST 门数大致相当,这是由于三者采用了相同的 QPU 间量子态路由策略,至于 ST 门数上的少量差异主要源于在平局情况下 QPU 间量子态路由策略所作的随机选择.相反,由于三者采用了不同的 QPU 内量子态路由策略,其在平均 SWAP 门数存在较大差异.即便 LookaheadSwap 和 SabreSwap 等方法在单 QPU 映射问题上表现优异,但在分布式映射场景下显然表现欠佳,其分别对应的 DQM\_L 和 DQM\_S 算法在平均 SWAP 门数上均明显多于 DQM 算法.经分析原因如下:在分布式映射进行到 QPU 内量子态路由进程时,由于受非局域门的间隔,在各 QPU 上即将调度的由连续局域门构成的子线路通常是浅层子线路,这些局域子线路极小的线路深度使得现有单 QPU 映射方法无法充分发挥出所用深度搜索或前瞻性启发式搜索技术的优势;另外,由于这些单 QPU 映射方法并不区分通信量子比特和数据量子比特,因此在进行量子态路由时极易使一些待移出 QPU 的量子态远离通信量子比特,从而增加了后续 QPU 间量子态传输所需的 SWAP 门开销.相反, DQM 算法基于整个分布式连通图构建量子态路由策略,如公式 (7) 所示的效应函数中所考虑的多层子线路并不会被非局域门打断,从而保证了较强的前瞻能力.从图 7 还可知, DQM\_L 算法的平均耗时远高于 DQM 和 DQM\_S 算法,这主要由于其采用的深度搜索策略,在其默认配置中,每次选择 SWAP 门时均需在决策树上向下探索 5 层,而 DQM 和 DQM\_S 算法均是根据启发式代价函数直接做出决策.综上,相较 DQM\_L 算法和 DQM\_S 算法, DQM 算法在平均门数和时间性能上均表现更佳,这也表明了分布式场景下进行 QPU 内量子态路由时感知量子态在其他 QPU 分布情况的重要性.

### 5.4 网络拓扑对算法性能的影响

网络拓扑是设计分布式量子计算系统时需要考虑的重要因素之一,其对量子线路映射所需的路由代价同样有着重要影响.为了评价网络拓扑对本文 DQM 算法的性能影响,我们构建了一个具有线性拓扑的网络.除拓扑结构



不同以外, 该线性网络包含的 QPU 类型及数量均和图 1 所示的二维网络相同; 为使得网络中的每个 QPU 均参与量子线路映射, 我们还构建了由 1000 个随机线路组成的测试集, 测试集中的每个线路均包含 24 个量子比特. 根据 CNOT 门数的不同可将该随机线路测试集划分为 10 个子集, 其中每个子集均包含 100 个仅由 CNOT 门组成的随机线路, 各子集对应的 CNOT 门数分别为 {100, 200, 300, 400, 500, 600, 700, 800, 900, 1000}. 本文 DQM 算法在两种网络拓扑上的实验结果图 8 所示, 图 8 的横轴表示各测试子集线路中包含的 CNOT 门数, 纵轴表示在各测试子集上所需的平均 ST 门数和 SWAP 门数.

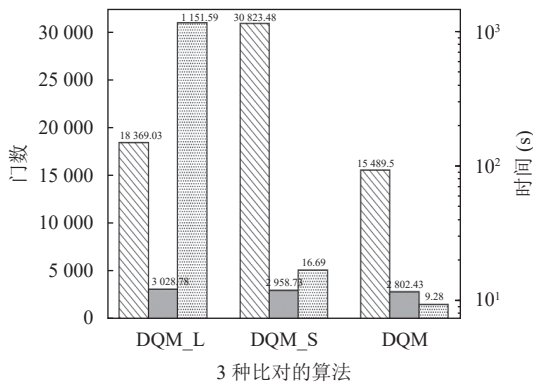


图 7 DQM\_L、DQM\_S 和 DQM 的性能比较

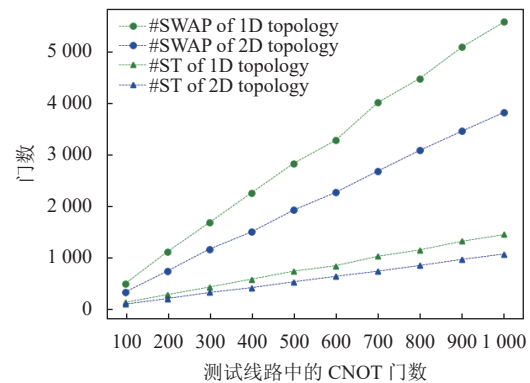


图 8 一维和二维网络拓扑上的分布式映射结果

由图 8 可知, 在两种网络拓扑上本文 DQM 算法所需的 ST 门数和 SWAP 门数均随着 CNOT 门数的增加呈近似线性增长关系, 且由于二维网络拓扑具备较好的连通性和较短的图直径, 其上所需的 ST 门数和 SWAP 门数普遍低于一维线性拓扑. 本文 DQM 算法可适应于任意网络拓扑结构, 因此其在特定网络拓扑上的映射代价可作为选择网络拓扑的参考因素之一.

## 6 总 结

本文面向超导量子互连技术和超导 QPU 的技术特征, 抽象出一种可扩展的分布式量子计算模型, 并基于该抽象模型提出了一种分布式量子线路映射方法. 本文构造的分布式抽象模型屏蔽了底层量子信道和 QPU 的物理细节差异, 从而使得本文映射方法可以适用于一类分布式量子计算系统. 本文分布式映射方法以最小化映射所需的辅助量子操作 (ST 门和 SWAP 门) 数为目标, 将线路中的量子比特分布到网络中的各 QPU 上, 并运用启发式量子态路由策略在网络中按需移动量子态, 从而使得量子线路中的所有量子门均满足 QPU 施加的物理约束. 实验结果表明, 本文所提方法较现有同类方法可以有效降低量子线路映射所需的辅助门数, 尤其是 ST 门数. 另外, 本文方法还可作为选择和评价分布式系统网络拓扑结构的辅助手段.

量子比特以及量子操作质量存在不容忽视的时空差异性, 是超导量子计算的显著特征之一, 在执行量子计算任务时优先选择更健壮的量子比特以及保真度更高的量子操作有助于提升量子计算的整体成功率. 类似的, 分布式量子线路映射方法在分配量子比特时优先选择整体质量更好的 QPU, 以及在移动量子态时优先选择保真度更好的 ST 门或 SWAP 门, 对于提升量子计算成功率同样有着极其重要的作用. 因此, 以提升量子线路保真度为目标设计和实现分布式量子线路映射方法是下一步的研究重点.

## References:

- [1] Krantz P, Kjaergaard M, Yan F, Orlando TP, Gustavsson S, Oliver WD. A quantum engineer's guide to superconducting qubits. *Applied Physics Reviews*, 2019, 6(2): 021318. [doi: 10.1063/1.5089550]
- [2] 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]
- [3] Wu YL, Bao WS, Cao SR, et al. Strong quantum computational advantage using a superconducting quantum processor. *Physical Review*

- Letters, 2021, 127(18): 180501. [doi: [10.1103/PhysRevLett.127.180501](https://doi.org/10.1103/PhysRevLett.127.180501)]
- [4] Zhu QL, Cao SR, Chen FS, *et al.* Quantum computational advantage via 60-qubit 24-cycle random circuit sampling. *Science Bulletin*, 2022, 67(3): 240–245. [doi: [10.1016/j.scib.2021.10.017](https://doi.org/10.1016/j.scib.2021.10.017)]
- [5] Huang HY, Broughton M, Cotler J, Chen ST, Li J, Mohseni M, Neven H, Babbush R, Kueng R, Preskill J, Mclean JR. Quantum advantage in learning from experiments. *Science*, 2022, 376(6598): 1182–1186. [doi: [10.1126/science.abn7293](https://doi.org/10.1126/science.abn7293)]
- [6] Brink M, Chow JM, Hertzberg J, Magesan E, Rosenblatt S. Device challenges for near term superconducting quantum processors: Frequency collisions. In: *Proc. of the 2018 IEEE Int'l Electron Devices Meeting*. San Francisco: IEEE, 2018. 611–613. [doi: [10.1109/IEDM.2018.8614500](https://doi.org/10.1109/IEDM.2018.8614500)]
- [7] Cuomo D, Caleffi M, Cacciapuoti AS. Towards a distributed quantum computing ecosystem. *IET Quantum Communication*, 2020, 1(1): 3–8. [doi: [10.1049/iet-qt.2020.0002](https://doi.org/10.1049/iet-qt.2020.0002)]
- [8] Rodrigo S, Abadal S, Alarcón E, Almudever CG. Will quantum computers scale without inter-chip comms? A structured design exploration to the monolithic vs distributed architectures quest. In: *Proc. of the 2020 Conf. on Design of Circuits and Integrated Systems*. Segovia: IEEE, 2020. 1–6. [doi: [10.1109/DCIS51330.2020.9268630](https://doi.org/10.1109/DCIS51330.2020.9268630)]
- [9] Kurpiers P, Magnard P, Walter T, Royer B, Pechal M, Heinsoo J, Salathé Y, Akin A, Storz S, Besse JC, Gasparinetti S, Blais A, Wallraff A. Deterministic quantum state transfer and remote entanglement using microwave photons. *Nature*, 2018, 558(7709): 264–267. [doi: [10.1038/s41586-018-0195-y](https://doi.org/10.1038/s41586-018-0195-y)]
- [10] Magnard P, Storz S, Kurpiers P, Schär J, Marxer F, Lütolf J, Walter T, Besse JC, Gabureac M, Reuer K, Akin A, Royer B, Blais A, Wallraff A. Microwave quantum link between superconducting circuits housed in spatially separated cryogenic systems. *Physical Review Letters*, 2020, 125(26): 260502. [doi: [10.1103/PhysRevLett.125.260502](https://doi.org/10.1103/PhysRevLett.125.260502)]
- [11] Leung N, Lu Y, Chakram S, Naik RK, Earnest N, Ma R, Jacobs K, Cleland AN, Schuster DI. Deterministic bidirectional communication and remote entanglement generation between superconducting qubits. *npj Quantum Information*, 2019, 5(1): 18. [doi: [10.1038/s41534-019-0128-0](https://doi.org/10.1038/s41534-019-0128-0)]
- [12] Zhong YP, Chang HS, Bienfait A, Dumur É, Chou MH, Conner CR, Grebel J, Povey RG, Yan HX, Schuster DI, Cleland AN. Deterministic multi-qubit entanglement in a quantum network. *Nature*, 2021, 590(7847): 571–575. [doi: [10.1038/s41586-021-03288-7](https://doi.org/10.1038/s41586-021-03288-7)]
- [13] Yan HX, Zhong YP, Chang HS, Bienfait A, Chou MH, Conner CR, Dumur É, Grebel J, Povey RG, Cleland AN. Entanglement purification and protection in a superconducting quantum network. *Physical Review Letters*, 2022, 128(8): 080504. [doi: [10.1103/PhysRevLett.128.080504](https://doi.org/10.1103/PhysRevLett.128.080504)]
- [14] Gold A, Paquette JP, Stockklauser A, Reagor MJ, Alam MS, Bestwick A, Didier N, Nersisyan A, Oruc F, Razavi A, Scharmann B, Sete EA, Sur B, Venturelli D, Winkleblack CJ, Wudarski F, Harburn M, Rigetti C. Entanglement across separate silicon dies in a modular superconducting qubit device. *npj Quantum Information*, 2021, 7(1): 142. [doi: [10.1038/s41534-021-00484-1](https://doi.org/10.1038/s41534-021-00484-1)]
- [15] Conner CR, Bienfait A, Chang HS, Chou MH, Dumur É, Grebel J, Peairs GA, Povey RG, Yan H, Zhong YP, Cleland AN. Superconducting qubits in a flip-chip architecture. *Applied Physics Letters*, 2021, 118(23): 232602. [doi: [10.1063/5.0050173](https://doi.org/10.1063/5.0050173)]
- [16] Kosen S, Li HX, Rommel M, *et al.* Building blocks of a flip-chip integrated superconducting quantum processor. *Quantum Science and Technology*, 2022, 7(3): 035018. [doi: [10.1088/2058-9565/ac734b](https://doi.org/10.1088/2058-9565/ac734b)]
- [17] Siraichi MY, Santos VFD, Collange C, Pereira FMQ. Qubit allocation. In: *Proc. of the 2018 Int'l Symp. on Code Generation and Optimization*. Vienna: ACM, 2018. 113–125. [doi: [10.1145/3168822](https://doi.org/10.1145/3168822)]
- [18] Botea A, Kishimoto A, Marinescu R. On the complexity of quantum circuit compilation. In: *Proc. of the 11th Int'l Symp. on Combinatorial Search*. Stockholm: AAAI, 2018. 138–142. [doi: [10.1609/socs.v9i1.18463](https://doi.org/10.1609/socs.v9i1.18463)]
- [19] Davarzani Z, Zomorodi-Moghadam M, Houshmand M, Nouri-Baygi M. A dynamic programming approach for distributing quantum circuits by bipartite graphs. *Quantum Information Processing*, 2020, 19(10): 360. [doi: [10.1007/s11128-020-02871-7](https://doi.org/10.1007/s11128-020-02871-7)]
- [20] Andrés-Martínez P, Heunen C. Automated distribution of quantum circuits via hypergraph partitioning. *Physical Review A*, 2019, 100(3): 032308. [doi: [10.1103/PhysRevA.100.032308](https://doi.org/10.1103/PhysRevA.100.032308)]
- [21] Houshmand M, Mohammadi Z, Zomorodi-Moghadam M, Houshmand M. An evolutionary approach to optimizing teleportation cost in distributed quantum computation. *Int'l Journal of Theoretical Physics*, 2020, 59(4): 1315–1329. [doi: [10.1007/s10773-020-04409-0](https://doi.org/10.1007/s10773-020-04409-0)]
- [22] Daei O, Navi K, Zomorodi M. Improving the teleportation cost in distributed quantum circuits based on commuting of gates. *Int'l Journal of Theoretical Physics*, 2021, 60(9): 3494–3513. [doi: [10.1007/s10773-021-04920-y](https://doi.org/10.1007/s10773-021-04920-y)]
- [23] Nikahd E, Mohammadzadeh N, Sedighi M, Zamani MS. Automated window-based partitioning of quantum circuits. *Physica Scripta*, 2021, 96(3): 035102. [doi: [10.1088/1402-4896/abd57c](https://doi.org/10.1088/1402-4896/abd57c)]
- [24] Ghodsollahee I, Davarzani Z, Zomorodi M, Plawiak P, Houshmand M, Houshmand M. Connectivity matrix model of quantum circuits and its application to distributed quantum circuit optimization. *Quantum Information Processing*, 2021, 20(7): 235. [doi: [10.1007/s11128-021-04920-y](https://doi.org/10.1007/s11128-021-04920-y)]

- 021-03170-5]
- [25] Wehner S, Elkouss D, Hanson R. Quantum Internet: A vision for the road ahead. *Science*, 2018, 362(6412): eaam9288. [doi: [10.1126/science.aam9288](https://doi.org/10.1126/science.aam9288)]
  - [26] Pirandola S, Eisert J, Weedbrook C, Furusawa A, Braunstein SL. Advances in quantum teleportation. *Nature Photonics*, 2015, 9(10): 641–652. [doi: [10.1038/nphoton.2015.154](https://doi.org/10.1038/nphoton.2015.154)]
  - [27] LaRacuent N, Smith KN, Imany P, Silverman KL, Chong FT. Modeling short-range microwave networks to scale superconducting quantum computation. arXiv:2201.08825, 2023.
  - [28] Chow JM. Quantum intranet. *IET Quantum Communication*, 2021, 2(1): 26–27. [doi: [10.1049/qt2.12002](https://doi.org/10.1049/qt2.12002)]
  - [29] Bardin J. Beyond-classical computing using superconducting quantum processors. In: Proc. of the 2022 IEEE Int'l Solid-state Circuits Conf. San Francisco: IEEE, 2022. 422–424. [doi: [10.1109/ISSCC42614.2022.9731635](https://doi.org/10.1109/ISSCC42614.2022.9731635)]
  - [30] Zulehner A, Paler A, Wille R. An efficient methodology for mapping quantum circuits to the IBM QX architectures. *IEEE Trans. on Computer-aided Design of Integrated Circuits and Systems*, 2019, 38(7): 1226–1236. [doi: [10.1109/TCAD.2018.2846658](https://doi.org/10.1109/TCAD.2018.2846658)]
  - [31] Li GS, Ding YF, Xie Y. Tackling the qubit mapping problem for NISQ-era quantum devices. In: Proc. of the 24th Int'l Conf. on Architectural Support for Programming Languages and Operating Systems. Providence: ACM, 2019. 1001–1014. [doi: [10.1145/3297858.3304023](https://doi.org/10.1145/3297858.3304023)]
  - [32] Nishio S, Pan YL, Satoh T, Amano H, van Meter R. Extracting success from IBM's 20-qubit machines using error-aware compilation. *ACM Journal on Emerging Technologies in Computing Systems*, 2020, 16(3): 32. [doi: [10.1145/3386162](https://doi.org/10.1145/3386162)]
  - [33] Niu SY, Suau A, Staffelbach G, Todri-Sanial A. A hardware-aware heuristic for the qubit mapping problem in the NISQ era. *IEEE Trans. on Quantum Engineering*, 2020, 1: 3101614. [doi: [10.1109/TQE.2020.3026544](https://doi.org/10.1109/TQE.2020.3026544)]
  - [34] Zhu PC, Cheng XY, Guan ZJ. An exact qubit allocation approach for NISQ architectures. *Quantum Information Processing*, 2020, 19(11): 391. [doi: [10.1007/s11128-020-02901-4](https://doi.org/10.1007/s11128-020-02901-4)]
  - [35] Zhou XZ, Feng Y, Li SJ. A Monte Carlo tree search framework for quantum circuit transformation. In: Proc. of the 39th Int'l Conf. on Computer-aided Design. San Diego: ACM, 2020. 138. [doi: [10.1145/3400302.3415621](https://doi.org/10.1145/3400302.3415621)]
  - [36] Li SJ, Zhou XZ, Feng Y. Qubit mapping based on subgraph isomorphism and filtered depth-limited search. *IEEE Trans. on Computers*, 2021, 70(11): 1777–1788. [doi: [10.1109/TC.2020.3023247](https://doi.org/10.1109/TC.2020.3023247)]
  - [37] Zhu PC, Guan ZJ, Cheng XY. A dynamic look-ahead heuristic for the qubit mapping problem of NISQ computers. *IEEE Trans. on Computer-aided Design of Integrated Circuits and Systems*, 2020, 39(12): 4721–4735. [doi: [10.1109/TCAD.2020.2970594](https://doi.org/10.1109/TCAD.2020.2970594)]
  - [38] Zhu PC, Feng SG, Guan ZJ. An iterated local search methodology for the qubit mapping problem. *IEEE Trans. on Computer-aided Design of Integrated Circuits and Systems*, 2022, 41(8): 2587–2597. [doi: [10.1109/TCAD.2021.3112143](https://doi.org/10.1109/TCAD.2021.3112143)]
  - [39] Dou XL, Liu L, Chen YT. An investigation into quantum program mapping on superconducting quantum computers. *Journal of Computer Research and Development*, 2021, 58(9): 1856–1874 (in Chinese with English abstract). [doi: [10.7544/issn1000-1239.2021.20210314](https://doi.org/10.7544/issn1000-1239.2021.20210314)]
  - [40] Zhou XZ, Feng Y, Li SJ. Supervised learning enhanced quantum circuit transformation. *IEEE Trans. on Computer-aided Design of Integrated Circuits and Systems*, 2023, 42(3): 437–447. [doi: [10.1109/TCAD.2022.3179223](https://doi.org/10.1109/TCAD.2022.3179223)]
  - [41] Pozzi MG, Herbert SJ, Sengupta A, Mullins RD. Using reinforcement learning to perform qubit routing in quantum compilers. *ACM Trans. on Quantum Computing*, 2022, 3(2): 10. [doi: [10.1145/3520434](https://doi.org/10.1145/3520434)]
  - [42] Wille R, van Meter R, Naveh Y. IBM's Qiskit tool chain: Working with and developing for real quantum computers. In: Proc. of the 2019 Design, Automation & Test in Europe Conf. & Exhibition. Florence: IEEE, 2019. 1234–1240. [doi: [10.23919/DATE.2019.8715261](https://doi.org/10.23919/DATE.2019.8715261)]
  - [43] Boykin PO, Mor T, Pulver M, Roychowdhury V, Vatan F. A new universal and fault-tolerant quantum basis. *Information Processing Letters*, 2000, 75(3): 101–107. [doi: [10.1016/S0020-0190\(00\)00084-3](https://doi.org/10.1016/S0020-0190(00)00084-3)]
  - [44] Chou KS, Blumoff JZ, Wang C, Reinhold PC, Axline CJ, Gao YY, Frunzio L, Devoret MH, Jiang L, Schoelkopf RJ. Deterministic teleportation of a quantum gate between two logical qubits. *Nature*, 2018, 561(7723): 368–373. [doi: [10.1038/s41586-018-0470-y](https://doi.org/10.1038/s41586-018-0470-y)]
  - [45] Delahaye D, Chaimatanan S, Mongeau M. Simulated annealing: From basics to applications. In: Gendreau M, Potvin JY, eds. *Handbook of Metaheuristics*. Cham: Springer, 2019. 1–35. [doi: [10.1007/978-3-319-91086-4\\_1](https://doi.org/10.1007/978-3-319-91086-4_1)]

#### 附中文参考文献:

- [39] 窦星磊, 刘磊, 陈岳涛. 面向超导量子计算机的程序映射技术研究. *计算机研究与发展*, 2021, 58(9): 1856–1874. [doi: [10.7544/issn1000-1239.2021.20210314](https://doi.org/10.7544/issn1000-1239.2021.20210314)]



朱鹏程(1982—), 男, 博士, 副教授, 主要研究领域为量子线路综合和优化, 可逆计算.



周祥臻(1991—), 男, 博士, 讲师, CCF 专业会员, 主要研究领域为量子计算, 量子编译器.



卫丽华(1984—), 女, 讲师, CCF 专业会员, 主要研究领域为量子线路综合, 智能调度算法.



郑盛根(1983—), 男, 博士, 副研究员, CCF 高级会员, 主要研究领域为量子算法, 光计算, EDA.



冯世光(1984—), 男, 博士, 讲师, CCF 专业会员, 主要研究领域为计算机复杂性理论, 量子计算.



管致锦(1962—), 男, 博士, 教授, 博士生导师, CCF 专业会员, 主要研究领域为可逆计算, 逻辑综合, 信息安全.