

분산형 양자컴퓨팅 소프트웨어 연구 개발 동향

Trends in Research and Software Technologies for Distributed Quantum Computing Software

황용수 (Y. Hwang, yhwang@etri.re.kr)

김태완 (T. Kim, TaewanKim@etri.re.kr)

배은옥 (E. Bae, eobae@etri.re.kr)

양자컴퓨팅연구실 선임연구원/실장

양자컴퓨팅연구실 선임연구원

양자컴퓨팅연구실 선임연구원

ABSTRACT

Quantum computing, which leverages the fundamental principles of qubit superposition and entanglement, is a transformative technology for addressing computationally intractable problems across sectors such as artificial intelligence, finance, and pharmaceuticals. However, one key limitation in current quantum computing technologies is the scaling problem – as the number of qubits increases, error rates rise significantly, limiting computational reliability and scalability. Distributed (or modular) quantum computing has emerged as a promising solution by connecting multiple small-scale quantum processor units via quantum and classical communication networks. This paradigm has been proposed as a practical solution to the scalability limitations of monolithic architectures. Leading technology companies like IBM and IonQ, along with academic research institutions, are actively developing modular quantum systems, signaling a shift toward networked quantum architectures. This paper provides a brief overview of distributed quantum computing principles, examines the software frameworks enabling distributed operations, and explores emerging distributed quantum algorithms designed to harness the full potential of these systems.

KEYWORDS 분산형 양자컴퓨팅, 시스템 소프트웨어, 응용 알고리즘

I. 서론

최근 글로벌 산업 및 과학기술계에서 양자컴퓨팅에 관한 관심이 급격히 고조되고 있다. 이는 기존의 고전컴퓨터로는 해결하기 어려웠던 난제들을 극복

하는데 양자컴퓨팅이 효과적인 역할을 할 것이라는 기대감 때문이다. 특히 인공지능, 금융, 제약, 소재 등 고차원 복잡성을 가진 문제 영역에 양자컴퓨팅을 도입하는 것이 산업 구조 자체를 바꿀 수 있는 게임 체인저로 인식되고 있다. 예를 들어, 고전컴퓨터

* DOI: <https://doi.org/10.22648/ETRI.2025.J.400507>

* 이 논문은 2025년도 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원(2019-0-00003) 및 국가과학기술연구회 글로벌 TOP 전략연구단 지원사업(No. GTL25011-000)의 지원을 받아 수행된 연구임.



본 저작물은 공공누리 제4유형

출처표시+상업적이용금지+변경금지 조건에 따라 이용할 수 있습니다.

©2025 한국전자통신연구원

는 분자 시뮬레이션 시에 원자 단위 상호작용을 정확히 표현하는 데 있어 지수적 계산 복잡도가 필요하지만, 양자컴퓨터는 분자의 양자상태를 자연스럽게 모사할 수 있기 때문에 제약 및 재료과학에서 획기적인 계산 능력을 기대할 수 있다.

양자컴퓨팅은 정보의 기본 단위로 큐비트(Qubit: Quantum Bit)를 사용한다. 0과 1의 상태를 동시에 가질 수 있는 중첩과 다중 큐비트 간의 얽힘이라는 양자역학 고유의 특성을 활용함으로써 병렬적이고 상호 연결된 계산을 수행할 수 있다. 이 덕분에 많은 전문가가 고전컴퓨터로는 수백 년이 걸릴 연산을 수 분 이내에 해결할 수 있을 것이라는 기대감을 가지고 있다.

하지만, 이러한 거대한 잠재력 앞에 넘어야 할 현실적인 난관이 있는데, 바로 스케일(Scale) 문제이다. 현재의 양자컴퓨터는 큐비트 개수를 늘리면서 안정성을 유지하는 데 어려움을 겪고 있다. 중첩과 얽힘 등의 양자역학적 특성을 유지해야 하는 큐비트는 외부 환경에 매우 민감한데, 양자 시스템 내에 큐비트의 개수를 늘리면 상호간섭의 정도가 심해져 오류율이 증가한다. 또한, 제어 인프라의 복잡성이 급격히 증가하는 등 시스템 전체의 안정성을 해치게 된다. 이에, 2023년 IBM이 공개한 1,121큐비트 양자 프로세서 Condor[1]는 단일 칩 구조에서 구현할 수 있는 물리적 큐비트 수의 최대 한계에 근접했다는 평가를 받기도 한다.

이러한 양자컴퓨팅 규모 확장의 한계를 극복하기 위한 방안으로 분산형 양자컴퓨팅(Distributed Quantum Computing) 또는 모듈형 양자컴퓨팅(Modular Quantum Computing)이 최근 큰 주목을 받고 있다. 이 방식은 여러 개의 소형 양자 프로세서를 네트워크로 연결하여, 단일 양자 프로세서의 한계를 넘어선 대규모 양자연산을 가능하게 한다. 또한, 양자오류를 개별 양자프로세서에 국소화하여 전체 시스템

의 안정성을 확보하는 전략도 함께 취할 수 있다.

이미 IBM의 Quantum System Two와 같은 최신 양자컴퓨팅 시스템은 모듈형, 분산형 접근 방식을 채택하고 있고[2], 이온트랩 방식의 양자컴퓨터를 개발하고 있는 IonQ 또한 차세대 양자컴퓨팅 전략으로 모듈형 방식을 추진하고 있다[3]. 대학 연구실에서도 관련 실험연구가 진행되고 있는데, 영국의 옥스퍼드 대학은 2025년 2월, 두 개의 분리된 양자 프로세서를 광자 네트워크 인터페이스로 연결한 시스템을 활용하여 분산형 양자 알고리즘을 시연하였다[4].

분산형 양자컴퓨팅은 기존 단일칩 기반 양자컴퓨팅이 갖는 확장성 문제 같은 근본적인 기술 한계를 극복하고, 양자 인터넷과 같은 차세대 통신 인프라와의 연계를 통해 단순한 성능 개선을 넘어선 새로운 컴퓨팅 패러다임을 제시하고 있다. 즉, 미래의 양자컴퓨터는 본질적으로 네트워크화되고 모듈화된 형태로 발전할 것이라는 암묵적 합의가 형성되고 있다.

본고에서는 이러한 분산형 양자컴퓨팅을 구현하기 위한 시스템 아키텍처에 대해서 간략하게 살펴보고, 이러한 시스템을 운용하는 시스템 소프트웨어와 분산형 양자컴퓨팅 기반의 응용 알고리즘에 대해서 살펴보고자 한다. 분산형 양자컴퓨팅에 대해서 살펴보기에 앞서, 고전 분산형 시스템은 무엇인지부터 살펴보도록 하자.

II. 분산형 컴퓨팅 개요

분산형 컴퓨팅(Distributed Computing)은 여러 개의 독립적인 컴퓨터 시스템이 네트워크를 통해 협력하여 하나의 계산 작업을 수행하는 컴퓨팅 모델의 미한다(그림 1 참조).

각 노드는 고유의 자원을 가지고 있으며, 작업을

병렬로 처리하거나 분할하여 동시에 수행함으로써 전체 계산 효율성과 신뢰성을 높일 수 있다. 이와 같은 방식은 단일 고성능 컴퓨팅 시스템 대비 확장성(Scalability)과 내결함성(Fault Tolerance) 측면에서 높은 장점을 갖는데, 최근 들어 빅데이터 분석, 클라우드 기반 서비스, 복잡한 과학 연산, 대규모 인터넷 서비스 등에서 필수적인 구조로 자리 잡고 있다.

분산형 컴퓨팅은 다음의 요소들로 구성된다.

- **노드(Node):** 분산 시스템에 참여하는 독립적인 연산능력을 갖는 개별 컴퓨터
- **네트워크(Network):** 노드들을 서로 연결하여 데이터와 명령을 주고받는 인프라
- **미들웨어(Middleware):** 노드 간의 통신, 자원 관리 및 데이터 동기화 등을 담당하는 소프트웨어
- **분산운영체제(Distributed OS):** 여러 컴퓨터를 단일 시스템처럼 운용하는 소프트웨어

이러한 분산시스템이 효과적으로 적용된 사례를 살펴보면, 2004년 구글이 대규모 데이터 처리를 위해 MapReduce라는 분산처리모델을 제안했다[5]. 데이터를 “Map” 단계에서 분산 처리한 후, “Reduce” 단계에서 결과를 집계하는 방식으로 수천 대의 서버에서 병렬연산을 효율적으로 처리할 수 있게 해

주었는데, 분산형 컴퓨팅 개념이 성공적으로 구현된 대표적인 사례로 손꼽히고 있다. 이외에도, SETI@Home 프로젝트[6]는 UC Berkeley가 전 세계 사용자의 개인컴퓨터를 연결해 외계 지적 생명체의 신호를 분석한 프로젝트로 가정용 PC 수백만 대를 묶어 가상의 슈퍼컴퓨터를 만든 사례로 주목받았다.

이러한 분산형 컴퓨팅은 대규모 데이터 처리 등에 효과적임이 입증되었지만, 여전히 기술적인 과제를 안고 있다. 병렬연산에서의 시간 차이 및 지연이 발생하는 노드 간 동기화 문제, 데이터 교환 시 과도한 네트워크 사용으로 인한 네트워크 병목, 노드 간 원활하지 못한 데이터 동기화 문제로 인한 데이터 일관성 유지 등이 분산형 컴퓨팅 성능 고도화를 위해 해결해야 할 사항들이다.

III. 분산형 양자컴퓨팅 개요

분산형 양자컴퓨팅(DQC: Distributed Quantum Computing)은 여러 대의 양자 프로세서를 양자 네트워크로 연결하여, 단일 양자컴퓨팅 시스템처럼 동작하게 하는 기술로, 현재 개발 중인 양자컴퓨터가 가진 확장성의 문제를 해결하기 위해 제안된 개념이다(그림 2 참조).

분산형 고전컴퓨팅의 경우, 대형 연산 작업을 개

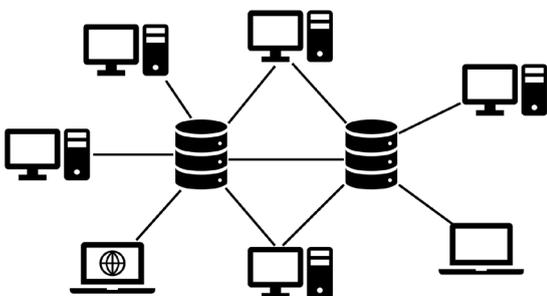


그림 1 분산형 컴퓨팅 개념

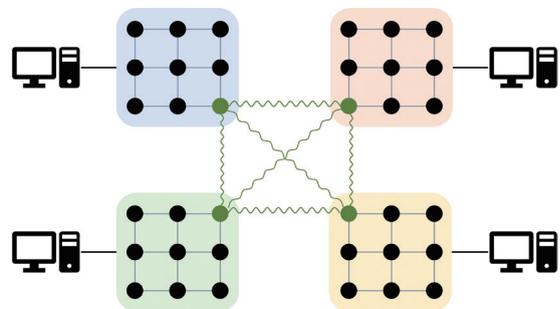


그림 2 분산형 양자컴퓨팅 개념도

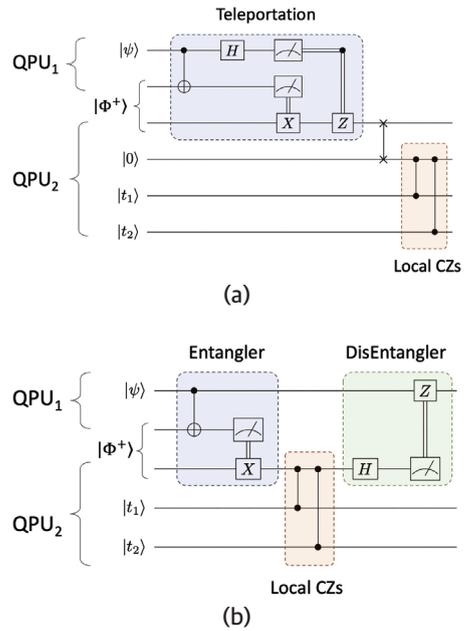
별 노드들에 배분하여, 각 노드들의 연산능력을 병렬적으로 활용하는 것이 목적이라면, 분산형 양자 컴퓨팅은 단순히 연산능력을 합치는 것을 넘어, 양자 얽힘을 통해 여러 양자 프로세서가 마치 하나의 거대한 양자 프로세서로 동작하도록 구성하는 것을 목표로 한다는 차이가 존재한다.

분산형 양자컴퓨팅은 분산 대상을 기준으로 다음과 같이 두 가지 유형으로 구분이 가능하다[7]. 다수의 소규모 계산 자원을 연결하여 대규모 연산을 수행하는 것을 자원 분산 양자컴퓨팅(Resource DQC)이라고 하고, 데이터가 여러 당사자에게 분산되어 있는 경우 데이터 분산 양자컴퓨팅(Data DQC)이라고 한다. 특히, 데이터 분산 양자컴퓨팅에서는 각 주체들이 국소적인 데이터만을 사용함으로써 외부에 데이터를 노출하지 않고 보안을 유지하면서 공동으로 문제 해결이 가능하다.

분산형 양자컴퓨팅 시스템의 핵심은 양자상태를 물리적으로 전송하고, 얽힘을 공유하며, 연산을 병렬처리하는 데 있으며, 다음의 요소들로 구성된다.

- **양자노드:** 국소적 양자연산과 얽힘 기반으로 타 양자노드들과 양자 정보를 주고받을 수 있는 양자 프로세서
- **양자 통신 링크:** 양자얽힘을 활용한 양자전송(Quantum Teleportation) 기술을 기반으로 하며, 양자 중계기(Repeater), 양자 메모리(Memory) 등으로 구성됨
- **양자 네트워킹 및 라우팅:** 서로 다른 양자노드 간의 양자 정보 교환을 조율하기 위한 정교한 양자 네트워킹 프로토콜 필요
- **제어 시스템:** 전체 시스템을 관리하는 고전컴퓨터. 각각의 양자노드 및 양자 통신 기기에 연결되어 있음

분산형 양자컴퓨팅에서 양자노드 간 비국소적인



출처 Reproduced from D. Barral et al., "Review of Distributed Quantum Computing: From single QPU to High Performance Quantum Computing," Comput. Sci. Rev., vol. 57, 2025, pp. 100747.

그림 3 (a) 양자 데이터 전송(TeleData)
(b) 양자 게이트 전송(TeleGate)

양자연산을 실행하기 위해서는, 양자 데이터를 원격지에 전송하여 원격 노드내에서 국소적으로 연산을 실행하거나, 양자 데이터는 현재 위치에 두고, 양자 게이트를 원격으로 전송하는 방식을 사용한다. 전자를 양자 데이터 전송 또는 TeleData라고 하고, 후자를 양자 게이트 전송 또는 TeleGate라고 부른다(그림 3 참조).

IV. 분산형 양자컴퓨팅 시스템 소프트웨어

복수의 소형 양자 프로세서들을 연결하는 분산형 양자컴퓨팅 시스템을 성공적으로 구축 및 운영하기 위해서는 단순히 하드웨어의 연결만으로는 충분하지 않고, 하드웨어 위에서 정교하게 동작하는 시스

템 소프트웨어가 필수적이다. 이는 기존의 고전 분산 컴퓨팅에서 사용되던 운영체제, 작업 스케줄러, 자원 관리자 등의 개념과 유사하지만, 양자 정보 특유의 제약 조건(예: 얽힘상태의 유지, 큐비트 복제 금지, 측정 시 상태 붕괴 등)을 반영해야 한다는 점에서 고전 분산형 소프트웨어와는 전혀 다른 소프트웨어 설계가 요구된다.

분산형 양자컴퓨팅 시스템을 운영하기 위한 시스템 소프트웨어가 수행해야 할 기능은 다음과 같다. 첫째, 사용자가 요구하는 거대 양자계산을 다중 양자노드에 배분하고, 작업 실행 순서 등을 스케줄링해야 한다. 둘째, 다중노드 사이에 양자얽힘을 생성/유지해야 한다. 셋째, 분산 양자연산 도중 발생한 오류를 국소적으로 억제하여, 타 양자노드로 전파되지 않도록 한다. 마지막으로, 고전컴퓨팅 시스템과의 인터페이스를 통해 전체 연산의 흐름을 조율해야 한다.

이러한 기능을 구현하기 위해 현재까지 제안되었거나 개발 진행 중인 소프트웨어들은 대체로 다음의 수직적인 구조를 따른다. 최상단은 응용(Application) 계층으로 사용자가 양자 프로그래밍언어를 사용하여 양자 알고리즘을 작성하는 단계이다. 두 번째 계층은 컴파일러(Compiler) 또는 미들웨어(Middleware) 계층으로, 응용 계층에서 작성된 고수준(High-Level)의 추상적인 양자 프로그램 코드를 하드웨어가 이해할 수 있는 저수준(Low-Level)의 명령어로 변환작업이 이루어진다. 사용자의 대규모 연산을 다중 양자노드에 분산시키고, 노드 간의 통신을 최적화하는 등, 분산형 시스템 특유의 요구사항을 처리한다. 세 번째 계층은 제어/관리 계층으로 소프트웨어와 하드웨어의 중간 역할을 수행하는데, 양자노드들의 상태를 모니터링하고, 작업 스케줄링, 양자-고전 하이브리드 워크로드 관리 등을 수행한다. 양자오류정정 또한 이 계층에서 이루어진다.

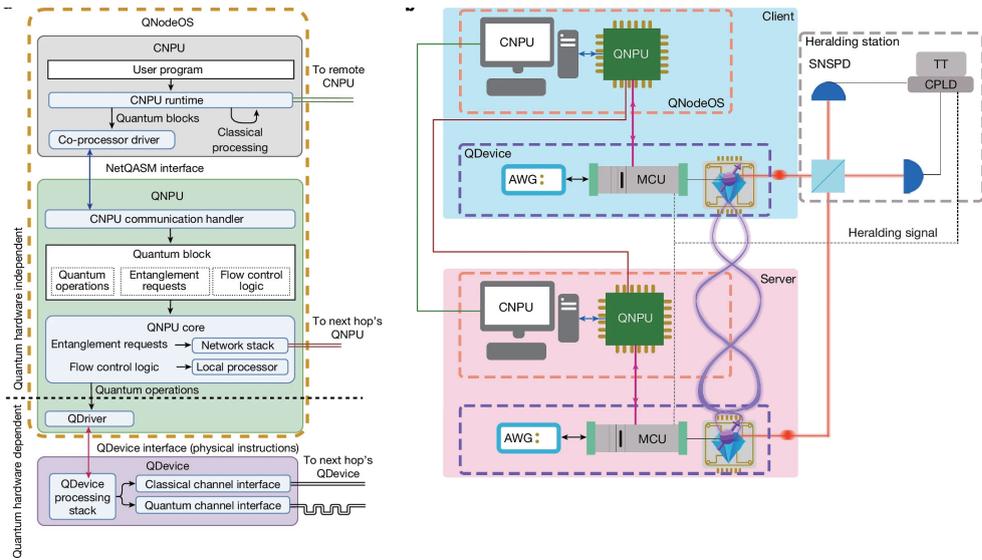
네 번째 계층은 하드웨어 추상화/인터페이스 계층으로 특정 물리 큐비트 플랫폼 특유의 물리적 특성을 추상화하여, 상위 계층의 소프트웨어가 특정 하드웨어에 종속되지 않고 작동할 수 있도록 지원하는 역할을 한다. 마지막으로 최하단 계층은 물리적 하드웨어 계층으로 실제 물리 큐비트를 대상으로 양자연산이 이루어지는 곳이고, 양자노드들 간의 연결도 이루어진다.

1. 분산형 양자컴퓨팅 운영체제

양자컴퓨팅 운영체제는 양자 하드웨어를 관리하고, 양자 자원을 조율하며, 양자 알고리즘의 실행을 용이하게 하는 소프트웨어 계층이다. 고전적인 운영체제가 메모리, CPU 프로세스 및 I/O 작업을 관리하는 것과 유사하게, 양자컴퓨팅 운영체제는 양자얽힘 관리, 양자오류정정 등과 같은 양자컴퓨팅의 고유한 문제를 처리해야 한다.

양자컴퓨팅 운영체제의 주요 기능은 큐비트 자원 관리(물리/논리 큐비트 할당 및 해제, 얽힘 자원 관리), 양자오류정정 실행 그리고 양자연산 스케줄링 및 최적화 등이 있다.

현재까지 공개된 대표적인 분산형 양자컴퓨팅 운영체제로는 네덜란드 델프트 공대(TU Delft)의 QNodeOS[8]가 있다. QNodeOS는 물리 큐비트 플랫폼 독립적인 고수준 소프트웨어로 양자네트워크 어플리케이션을 실행할 수 있는 아키텍처를 제공한다(그림 4 참조). 다이아몬드 NV 센터와 이온트랩 플랫폼을 지원하기 위해서 QDriver를 포함한다. QDriver는 QNodeOS에 포함된 유일한 하드웨어 특성 종속 요소로 NetQASM(Net Quantum ASSEMBly code)으로 표현된 양자연산 명령어를 물리 플랫폼 맞춤형의 기계어로 변환하는 기능을 가지고 있다. 즉, 이



출처 Reprinted from C.D. Donne et al., "An operating system for executing applications on quantum network nodes," Nature, vol. 639, 2025, pp. 321-328.

그림 4 QNodeOS 아키텍처

연구에서는 이온트랩 양자노드와 NV 센터 양자노드를 위한 QDriver를 구현하고 시험검증하였다.

양자노드 간의 작업 스케줄링은 분산형 양자컴퓨팅 운영체제의 중요한 기능이다. 각 양자노드(QPU: Quantum Processing Unit)의 상태(큐비트 사용량, 얽힘 가용성 등)를 고려하여 작업을 분산 배치하고, 양자 얽힘의 생성과 유지시간을 고려해 연산 순서를 최적화해야 한다. 참고로, IBM의 Qiskit(Quantum Information Software KIT) Runtime은 이러한 스케줄러 기능을 도입해 QPU 자원을 동적으로 할당하고 있으며, AWS(Amazon Web Services)의 Braket 플랫폼에서는 PennyLane과 연동해 멀티 디바이스 하이브리드 작업 스케줄링 기능을 지원하고 있다[9]. 이외에도 Harvard-MIT 공동연구 프로젝트인 QuNetStack은 양자얽힘 네트워크를 통해 다수의 QPU를 연결한 환경에서 동기화 스케줄링을 구현하고 있으며, 얽힘 생성 실패에 대비한 재시동 기능까지 포함되어 있다[10]. MIT-IBM Watson AI Lab에서는 분산

QPU 환경에서 Reinforcement Learning 기반 스케줄링 최적화 알고리즘을 실험하고 있으며, 이를 통해 실행시간 단축, 에러율 감소, 얽힘 경로 최적화를 동시에 달성한 사례를 발표하였다[11].

2. 분산형 양자컴퓨팅 컴파일러

양자컴파일러는 고수준 양자 알고리즘을 물리 큐비트에 매핑하는 소프트웨어로, 추상적인 양자 프로그래밍 명령어를 양자 프로세서에서 실행 가능한 기계어로 번역해주는 역할을 수행한다. 양자하드웨어 플랫폼에 특화된 게이트로 분해하고, 플랫폼 구조에 맞게 게이트 시퀀스를 최적화하는 역할까지 수행한다.

분산형 양자컴퓨팅 컴파일러의 주요한 과제는 서로 다른 양자노드 간의 비국소적 연산 오버헤드를 최소화하는 회로분할이다. 양자얽힘 기반의 양자전송은 여러 양자노드 사이에서 양자연산을 가능하게

하는 핵심요소이지만, 구현을 위한 시공간 비용이 높아 양자연산 실행 시 주요 병목 현상이 된다. 따라서, 분산형 양자컴파일러는 비국소적 양자연산의 사용을 최소화하는 것을 목표로 한다.

분산형 양자컴파일러는 일반적으로 분석단계와 합성단계로 구성된다. 분석단계는 프로그램 코드의 어휘적, 구문적, 의미론적 정확성을 확인하고, 중간 표현(IR: Intermediate Representation)을 생성한다. 분산형 환경에 특화된 프로그래밍 언어를 개발함으로써 분석단계에서 중간표현을 효과적으로 생성할 수 있다. QMPI(Quantum Message Passing Interface)[12]는 고전 프로세서 노드와 양자 프로세서 노드를 구분하고, 양자업힘쌍과 같은 양자자원을 포함함으로써 고전통신의 메시지 전달 방식을 분산형 양자 시스템으로 확장한 대표적 사례이다.

합성단계에서는 중간표현을 실제 분산형 하드웨어에서 실행 가능한 저수준 언어로 변환하며, 대상 노드에 대한 최적화 등을 수행한다. 합성단계는 다음과 같이 세분화할 수 있는데, 다중 큐비트 연산 수를 최소화하는 최적화, 논리 큐비트를 물리 큐비트에 매핑하는 큐비트 매핑, 그리고 마지막으로 사용자 프로그램이 양자역학적 특성을 만족하는지 확인하는 검증단계가 그것이다.

분산형 양자컴파일러는 컴파일 과정에서 양자노드 간의 작업배분 및 결과 취합 방식의 유형에 따라 다음과 같은 양자회로의 분산 모델이 존재한다.

- **회로 분산(Circuit Distribution):** 단일 회로를 최적으로 분할하여 여러 양자노드에 분산시키고, 양자통신채널을 이용하여 양자노드 간 비국소적 연산을 통해 상호작용함
- **회로 절단(Circuit Cutting):** 단일 양자회로를 서로 얽히지 않은 작은 크기의 회로로 분할하여 실행하고, 각 회로의 실행결과를 고전통신을 통해서 결합하고 전체 결과 도출

- **단순 병렬(Embarrassingly Parallel):** 통신이 거의 필요 없는 작업을 여러 양자노드에 배분하는 방식

프랑스 소재 Welinq는 2025년 1월, 분산형 양자 컴퓨팅 전용 컴파일러인 araQnc를 출시하였다[13]. 복잡한 양자 알고리즘을 여러 개의 소형 양자노드에 분할하여 양자 네트워크상에서 실행할 수 있도록 설계되었으며, 특히 양자 네트워크 및 양자 장치 환경에 따라 최적의 분산 실행구조로 컴파일 가능하다. Well Fargo 연구진은 분산형 양자컴퓨팅 시뮬레이터상에서 양자 알고리즘의 실행 및 벤치마크를 지원하는 컴파일러 프레임워크를 제시하였다[14]. 이외에도, 유럽 “Quantum Internet Alliance” 프로젝트를 주도하는 파르마 대학의 Michele Amoretti 교수팀 또한 네트워크와 QPU의 제약사항 등을 고려하여 컴파일 및 실행 매니지먼트를 지원하는 모듈형 분산 양자컴퓨팅 컴파일러를 발표하였다[15]. 이들은 향후 컴파일의 대상 아키텍처를 분산형 데이터센터 환경으로 확장할 계획이다.

이외에도, 분산환경에서 최적의 컴파일러를 개발하기 위한 다양한 시도가 이어지고 있고, 양자컴퓨팅 글로벌 기업들도 자사의 소프트웨어를 분산형 환경으로 확장하기 위한 시도들을 진행하고 있다.

3. 분산형 양자컴퓨팅 미들웨어

분산형 양자 미들웨어는 분산된 양자 자원과 고전 컴퓨팅 자원 사이의 통신 및 상호 운용성을 관리하고, 개발자에게 추상화된 환경을 제공하는 것을 목적으로 한다.

즉, 분산형 양자컴파일러는 사용자의 추상적인 양자 프로그램을 분산된 양자 하드웨어에 맞게 변환하고 분할하는 역할을 수행하는 반면, 미들웨어

표 1 분산형 컴파일러와 분산형 미들웨어 비교

구분	분산형 컴파일러	분산형 미들웨어
주요 기능	대규모 양자회로를 분산된 양자노드에 맞게 변환 및 최적화	분산된 양자 및 고전 자원 간의 통신, 관리 및 상호 운용성 보장
작업 대상	양자 알고리즘	하드웨어 자원 (QPU, 고전컴퓨터, 양자 네트워크)
역할	양자 프로그램을 여러 양자노드가 이해할 수 있는 언어로 번역	분산형 시스템의 각 부분을 연결하고 제어하는 운영체제 또는 관리자
목적	주어진 양자 프로그램을 가장 효율적으로 실행하는 데 필요한 회로 생성	개발자들에게 분산형 양자 하드웨어 환경의 복잡성을 숨기고, 원활한 시스템 환경 제공

는 변환된 프로그램이 실제 분산 시스템에서 효과적으로 실행되도록 자원관리 및 통신을 지원하는 인프라적 역할을 수행한다(표 1 참조).

Galician Supercomputing Center(CESGA)와 Qoro Quantum은 “Qoro Divi”라는 이름의 미들웨어 플랫폼을 기반으로 HPC(High Performance Computing) 환경에서 하이브리드 알고리즘(VQE, QAOA 등)의 분산 양자회로 시뮬레이션을 수행하였다[16]. Cisco에서는 광학스위치를 기반으로 여러 양자노드를 상호연결하여 대규모 양자컴퓨팅이 확장 가능한 양자 데이터 센터 아키텍처를 제시하였다[17]. 이를 위해서 네트워크 인식 기반의 양자 오케스트레이터(Orchestrator)를 제시하였는데, 이는 분산형 양자 네트워크 시스템에서 고수준의 양자 응용 프로그램과 물리적 계층을 연결하는 역할을 수행한다.

4. 분산형 양자컴퓨팅 시뮬레이터

분산형 양자컴퓨팅 시뮬레이터는 실제 분산 양자 알고리즘 연구, 하이브리드 HPC 환경 실증, 양자 네트워크 모델링 등을 중심으로 연구개발이 활발히 진행되고 있다.

다음은 주요 분산형 양자컴퓨팅 시뮬레이터 프레임워크에 대한 연구개발 내용이다.

4.1 DQCS

DQCS(Distributed Quantum Computing Simulator)는 분산형 양자 알고리즘의 성능을 평가하고 벤치마킹하기 위해 개발된 시뮬레이터로, 특히 NISQ(Noisy Intermediate-Scale Quantum) 시스템을 적용한 분산형 양자컴퓨팅 구현을 고려하여, 노드 간의 비국소적 양자 게이트에서의 노이즈 영향을 시뮬레이션할 수 있다[14].

DQCS는 사용자 알고리즘에 대해서 분산형 양자 회로를 생성하고, 시뮬레이션하는 기능을 제공한다. 그리고, 알고리즘의 실행 충실도와 얽힘 생성 단계 수와 같은 분산형 양자컴퓨팅 관련 성능을 측정하고, 벤치마킹할 수 있도록 한다. 이외에도, 동적 양자회로(Dynamic Quantum Circuits)를 사용하여 양자자원 오버헤드를 줄이고 충실도를 높이는 방법을 연구하는 데 활용될 수 있다.

해당 논문에서는, 양자위상추정(Quantum Phase Estimation)과 양자진폭추정(Quantum Amplitude Estimation) 알고리즘에 대한 분산형 실행 모사 결과를 보여준다. 양자위상추정 알고리즘에 대한 모사 시에 동적 양자회로 기법을 적용함으로써 2-큐비트 양자게이트의 오류 영향을 최소화할 수 있음을 보인다.

4.2 Interlin-q

Interlin-q는 분산형 양자 알고리즘을 설계하고, 검증하는 과정을 단순화하기 위해 만들어진 소프트웨어 프레임워크로, 단일 양자회로를 분산 양자컴퓨팅 시스템에 적합하도록 자동으로 재구성할 수 있다[18]. 양자 네트워크 시뮬레이터인 QuNet-Sim[19]을 기반으로 구축되었으며, 병렬 및 분산 양자 알고리즘을 시뮬레이션하는 데 중점을 두고 있

다. 이를 위해서 고수준의 스케줄링 알고리즘을 포함하여, 수평적 속도향상으로 알고리즘의 실행시간을 단축하는 것이 가능하다.

Interlin-q는 대규모 양자연산을 고성능으로 실행하는 것보다는 분산형 양자컴퓨팅을 실행하는 데 필요한 구성요소들을 시험검증하는 것을 목적으로 하며, 이를 실현하기 위해 사용자 노드, 제어노드, 계산노드로 구성된 분산형 양자컴퓨팅 시뮬레이션 아키텍처를 제공한다. 이를 활용하여, 분산형 위상 추정 알고리즘과 수소분자(H₂)의 바닥상태 에너지 추정 알고리즘의 분산 시스템 실행을 모사하였다.

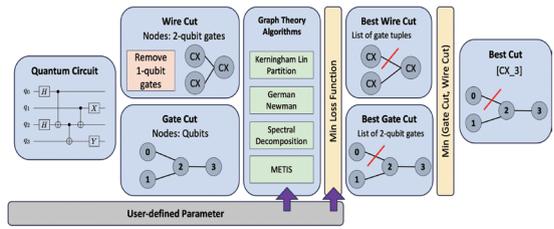
4.3 Qdislib

Qdislib은 스페인의 바르셀로나 슈퍼컴퓨팅센터에서 개발한 분산형 양자컴퓨팅 시뮬레이션 프레임워크 라이브러리이다[27,28]. 대규모 양자회로를 더 작고 관리하기 쉬운 서브회로로 분해하여 독립적으로 실행하고, 나중에 재구성하는 기능을 제공하는데, 특히 양자회로를 그래프로 표현하여 효율적인 분할, 조작 및 실행을 가능하게 하며, 회로의 큐비트 분할과 게이트 분할 기술 모두를 지원한다.

Qiskit[29] 및 Qibo[30] 같은 잘 알려진 양자컴퓨팅 라이브러리와 호환 가능하며, PyCOMPSs[31,32]와 같은 분산 컴퓨팅 프레임워크를 통해 이종 컴퓨팅 시스템에서 양자서브회로를 완전히 병렬 실행할 수 있도록 설계되어, 고전-양자 하이브리드 양자컴퓨팅 시스템과의 원활한 통합이 가능하다.

Qdislib의 핵심 기능 중 하나는 FindCut 알고리즘으로, 사용자 정의 제약조건(회로의 최소분할 등)에 따라 양자회로를 최적으로 분할할 수 있다(그림 5 참조).

분할된 양자회로들은 PyCOMPSs 런타임을 통해 이종컴퓨팅 시스템에서 병렬로 실행 가능하고, 이후 각 회로의 측정결과는 원래 회로의 기댓값을 재구성하는 데 사용 가능하다. 이러한 접근 방식은 유



출처 Reproduced from M. Tejedor et al., "Distributed Quantum Circuit Cutting for Hybrid Quantum-Classical High-Performance Computing," arXiv preprint, 2025. doi: 10.48550/arXiv.2505.01184

그림 5 FindCut Flow[27]

연성, 높은 병렬화 그리고 최소 큐비트 사용량과 최대 병렬 처리 간의 균형을 제공하여, 향후 대규모 양자회로의 확장성과 효율적인 분산형 양자컴퓨팅 실행을 가능하게 할 것이다.

4.4 Diskit

Diskit은 IBM의 Qiskit 프레임워크를 기반으로 하는 연구 프로젝트로, 분산 양자컴퓨팅 환경에서 양자 회로를 효율적으로 실행하기 위해 설계되었다[33]. 기존 Qiskit이 단일 양자 프로세서에서 양자 회로를 효율적으로 실행하기 위해 설계되었다면, Diskit은 여러 개의 물리적으로 분리된 양자노드를 활용해 더 큰 규모의 양자컴퓨팅 작업을 수행한다.

다른 분산형 양자컴퓨팅 소프트웨어와 동일하게 대규모 분산형, 고전-양자 하이브리드 양자컴퓨팅 시스템에서 분산형 양자 알고리즘 개발을 지원한다.

5. 분산형 양자컴퓨팅 소프트웨어 개발 키트

양자 소프트웨어 개발 키트(SDK: Software Development Kit)는 양자 알고리즘을 개발, 테스트 및 배포하기 위한 포괄적인 프레임워크를 제공한다. 이러한 SDK에는 Q#, Qiskit과 같은 양자 프로그래밍 언어와 양자회로 합성, 시뮬레이션 및 양자 하드웨어 실

행을 위한 도구가 포함된다. 이들은 알고리즘 개발을 위한 고수준 추상화를 제공하여, 개발자가 양자 역학의 복잡한 세부 사항을 깊이 이해하지 않아도 양자 프로그램을 작성할 수 있도록 한다.

분산형 양자컴퓨팅 개념이 부상하면서, 이를 위한 소프트웨어 도구의 필요성도 점차 커지고 있다. 하지만, 현재 분산형 양자컴퓨팅 SDK는 초기단계에 있으며, 대부분 고전-양자 하이브리드 컴퓨팅에 초점을 맞추고 있다.

V. 분산형 양자컴퓨팅 응용 알고리즘

분산형 양자컴퓨팅은 임의의 양자 알고리즘을 다중 양자노드에서 효과적으로 실행할 수 있도록 지원한다. 따라서, 분산형 환경을 대상으로 특별하게 설계된 알고리즘 만을 실행하도록 제한하지 않는다. 다만, 현재까지 일부 알고리즘들이 분산형 환경에 맞게 개발되어 있는데, 모듈분할, 양자얽힘 연동 그리고 결과통합 과정이 포함되어 있다.

- **모듈분할(Modular Decomposition):** 전체 연산을 분할 가능한 하위 연산으로 나누고, 각 QPU에 배정
- **양자얽힘 연동:** 분산된 QPU 간의 상호작용을 위해 얽힘 큐비트를 미리 생성하여 공유. 원격 CNOT 게이트 또는 양자전송 활용
- **결과통합:** 각 QPU에서의 결과를 고전적/양자적 방식으로 통합

1. 분산 Grover 알고리즘

Grover 알고리즘은 고전적으로 선형적인 복잡도를 갖는 비구조적 데이터 탐색 문제를 $O(\sqrt{N})$ 으로 단축하는 양자 데이터 검색 알고리즘으로, 최근 들어 분산형 환경을 대상으로 변형된 데이터 검색

알고리즘이 개발되고 있다. 분산형 환경에서는 탐색 공간을 여러 양자노드에 분할하고, 각 양자노드가 자신의 영역 내에서 Grover 연산을 수행한 다음, 결과를 얽힘 기반 방식으로 통합하는 구조를 사용하는 방식이다.

앞서 기술한 바와 같이 최근 옥스퍼드 대학의 연구진은 두 대의 분리된 이온트랩 모듈을 광학방식으로 연결하여, 분산형 구조상에서 Grover 알고리즘을 성공적으로 실행하여 보고하였다[4]. 이외에도, 중국 연구진은 여러 멀티-타겟노드를 대상으로 분산형 데이터 검색 알고리즘을 개발하였는데, 기존 제안된 알고리즘보다 더 적은 큐비트를 사용하거나 정확성을 높였다[20,21].

2. 분산 양자시뮬레이션

양자시뮬레이션은 분자 구조, 화학반응, 고체 물리 등의 다양한 현상을 모사하는 데 중요하다. 분산형 시뮬레이션에서는 전체 해밀토니안을 부분 연산으로 나누어 각 QPU에서 병렬적으로 계산하고, 얽힘 교환 등을 통해서 시뮬레이션 전체를 구성한다.

미국 연구진은 모듈형 양자컴퓨팅 환경에서 양자 화학 시뮬레이션을 효율적으로 실행할 수 있는 분산 알고리즘(dUSCC)을 제안하고, 실제분자 (H_4) 시스템에 적용해 양자모듈 간 통신지연에도 안정적으로 동작함을 입증하였다[22].

3. 분산 QAOA

분산형 QAOA(Distributed Quantum Approximate Optimization Algorithm)는 기존 QAOA가 대규모, 고차원 최적화 문제에 적용하기 어렵다는 한계를 극복하기 위해 등장한 연구이다. 대형 조합 최적화 문제를 여러 개의 작은 서브문제로 분할하고, 서브문

제를 다중 양자노드에서 병렬로 실행하는 전략을 채택한다. 각 서브문제 솔루션을 반복적으로 통합해 전역 해답에 수렴시키는 방식으로 최종 해답을 찾는다. 참고문헌 [23]에서는 분산형 QAOA를 적용하여, 기존 QAOA보다 훨씬 높은 근사비율과 짧은 시간 이내에 재료과학 분야의 최적화 문제를 성공적으로 해결할 수 있음을 보였다.

4. 분산 VQE

분산형 VQE(Variational Quantum Eigensolver)는 복잡한 양자 시스템의 바닥상태 에너지(Ground-State Energy)와 같은 고윳값을 찾는 대표적인 양자 알고리즘인 VQE를 분산형 양자컴퓨터에서 수행하는 기술이다. 거대 계산을 여러 양자노드로 분할 병렬화함으로써 단일 양자 프로세서에서 해결하기 어려웠던 대규모 문제를 보다 효율적으로 풀 수 있도록 설계되었다.

캐나다 연구진은 양자노드 간 제한된 정보교환 환경에서도 분산형 양자컴퓨팅이 양자계산문제를 정확하게 해결할 수 있음을 보였다[24]. 특히 분산형 안사츠(Ansatz)를 사용하여, 모듈 간 비국소 연산을 세 번 수행하는 것만으로도 모듈 간 연산이 없는 경우에 비해 유의미한 이점이 있음을 확인하였다.

루이지애나 주립대 연구진은 매개변수화된 양자 회로(PQC: Parameterized Quantum Circuit)를 여러 양자노드에서 분산 실행할 수 있게 하는 소프트웨어 플랫폼을 개발하여 오픈소스 소프트웨어(DVQE)로 공개하였다[25].

5. 분산 양자머신러닝

분산형 양자머신러닝이란 단일 양자프로세서의 한계를 극복하고 복잡한 작업을 여러 양자노드에

걸쳐 관리 및 처리함으로써 양자머신러닝의 기능과 성능을 확장하는 것을 목표로 한다.

미국의 William & Mary 대학 연구진은 분산형 양자머신러닝을 분산 유형에 따라 연합학습(Federated Learning)과 모델병렬학습(Model-Parallel Learning)으로 구분하고, 각각에 대한 기본 개념을 제시하였다. 그리고, 이를 IBM Quantum 플랫폼에서 테스트한 결과 최적화된 모델의 경우 96% 수준의 높은 정확도를 얻었다.

- **양자연합학습** 여러 양자노드가 각자의 국소적인 데이터를 사용하여 전체 양자머신러닝 모델을 학습하는 기술
- **양자모델병렬학습** 전체 양자머신러닝 모델을 복수의 서브모델로 분할되어 분산 양자노드에 분산되며, 각 노드들이 할당된 데이터를 병렬로 처리

네덜란드의 TNO에서는 여러 당사자가 자신의 데이터를 노출하지 않고도 안전하게 양자 알고리즘을 수행할 수 있도록 하는 분산형 양자머신러닝(데이터 분류기)을 제안하였다[26]. 물리적으로 데이터를 한군데 모으지 않고, 공유된 얽힘상태를 이용해 계산서버와 여러 당사자가 분산형으로 데이터를 인코딩하고 분류한다. IBM Qiskit 기반의 시뮬레이션을 통해서, 분산형으로 수행한 계산 결과와 단일 시스템상에서 수행한 계산 결과가 일치함도 보였다.

VI. 분산형 양자컴퓨팅 소프트웨어의 도전 과제

분산형 양자 알고리즘은 이론적으로는 다양한 확장 가능성을 지니지만, 실제 구현을 위해서는 다음과 같은 기술적 과제가 존재한다. 먼저, 고품질의 양자노드 간 얽힘이 필요하다. 그리고 양자 전송 기반 양자 상태 이동의 높은 성공률이 요구된다. 이 외에도 양자노드 간 실시간 동기화와 정교한 스케줄링이 요구된다.

또한, 양자 시스템은 매우 높은 오류율을 가지고 있기 때문에, 오류정정(QEC: Quantum Error Correction)은 필수적인 요소이다. 분산 환경에서는 각 양자노드가 개별적으로 오류정정을 수행할 뿐만 아니라 분산된 결과를 기반으로 전체 시스템의 오류를 통합적으로 관리해야 한다. 이를 위해 운영체제에서는 분산 양자오류정정(Distributed QEC) 기능을 포함해야 한다. 각 양자노드 간의 오류 상관성을 추적하고, 피드백 경로를 구성하며, 논리 큐비트 상태의 재구성에 필요한 정보를 교환할 수 있어야 한다.

기존의 고전적인 통신 방식은 양자 상태의 민감성 때문에 적용하기 어려우므로 양자 통신 네트워크와 양자 메모리를 통합하는 소프트웨어 아키텍처 연구가 매우 중요한 도전 과제로 부상하고 있다.

기술적인 부분 외에도, 양자 하드웨어 간의 호환성과 표준화 부재도 큰 난관으로 손꼽힌다. 초전도체, 이온트랩, 광자 기반 등 다양한 QPU들이 존재하지만, 이들을 하나의 네트워크에서 원활히 연동하기 위한 공통 인터페이스와 실행환경이 매우 부족하다.

미래의 분산형 양자컴퓨팅 소프트웨어는 이러한 한계를 넘어 표준화, 추상화, 자동화라는 세 가지 축을 중심으로 발전할 것으로 예상된다. 첫째, NetQASM과 같은 공통 명령어 집합 및 QNodeOS 같은 운영체제가 발전하면서, 하드웨어에 독립적인 플랫폼 표준화가 추진될 것이다. 둘째, 응용 개발자가 하드웨어 세부사항을 신경 쓰지 않아도 되도록 하는 고수준 프로그래밍 추상화 계층이 강화될 것이다. 마지막으로, 대규모 네트워크에서의 얽힘 분배, 자

원 할당, 오류 정정은 사람이 일일이 최적화할 수 없으므로, AI 기반 자동화 소프트웨어 스택과 자율적 네트워크 제어 알고리즘이 중요한 연구 방향이 될 전망이다. 이를 통해 궁극적으로 전 세계적으로 연결된 양자인터넷(Quantum Internet) 기반의 분산 양자컴퓨팅 실현을 가능케 할 것이다.

VII. 결론 및 제언

양자컴퓨팅 커뮤니티에서는 대규모 결합허용 양자컴퓨팅이 단일 양자칩 기반 양자컴퓨팅 시스템을 통해 달성되기보다는 분산형 시스템을 통해서 이루어질 것이라는 공감대가 형성되고 있다. 이는 견고한 양자 통신 및 분산 시스템 소프트웨어 개발이 양자컴퓨팅의 잠재력을 완전히 실현하기 위한 선택적 개선 사항이 아니라 근본적인 요구사항임을 의미한다.

양자컴퓨팅 연구개발의 어려움을 논할 때, 하드웨어의 한계가 자주 강조되고 있지만, 소프트웨어 및 프로그래밍 언어에서의 표준화 부족, 양자 프로그래밍의 디버깅 복잡성, 기존 HPC 인프라와의 통합 문제 등도 양자컴퓨팅 소프트웨어 개발의 중요한 과제로 지적된다. 향후, 양자컴퓨팅 채택속도와 잠재력 실현은 소프트웨어 생태계의 성숙도에 크게 좌우될 것으로 예상된다. 따라서, 이러한 과제를 해결하기 위해서 표준화, 직관적인 프로그래밍 도구, 그리고 견고한 미들웨어 개발과 함께, 양자 교육 및 인력 개발에 대한 지속적인 투자가 필요하다.

참고문헌

- [1] D. Castelvecchi, "IBM releases first-ever 1,000-qubit quantum chip," Nature, vol. 624, no. 238, 2023.
- [2] R. Mandelbaum et al., "How IBM will build the world's first large-scale, fault-tolerant quantum computer," IBM blog, 2025. 6. 10. <https://www.ibm.com/quantum/blog>

- [3] HPCwire Website. <https://www.hpcwire.com>
- [4] D. Main et al., "MapReduce: Simplified Data Processing on Large Clusters," *Nature*, vol. 638, 2025, pp. 383-388.
- [5] J. Dean and S. Ghemawat, "The Optimization and Improvement of MapReduce in Web Data Mining," in *Proc. USENIX Symp. Oper. Syst. Des. Implementation*, (San Francisco, CA, USA), Dec. 2004, pp. 137-150.
- [6] David P. Anderson et al., "SETI@home: an experiment in public-resource computing," *Commun. ACM*, vol. 45, no. 11, pp. 56-61.
- [7] J.C. Broschero et al., "Distributed Quantum Computing: Applications and Challenges," arXiv preprint, 2024. doi: 10.48550/arXiv.2410.00609
- [8] C.D. Donne et al., "An operating system for executing applications on quantum network nodes," *Nature*, vol. 639, 2025, pp. 321-328.
- [9] <https://aws.amazon.com/blogs/aws>
- [10] S. DiAdamo et al., "Distributed Quantum Computing and Network Control for Accelerated VQE," *IEEE Trans. Quantum Eng.*, vol. 2, no. 3100921, 2021, pp. 1-12.
- [11] W. Luo et al., "Adaptive Job Scheduling in Quantum Clouds Using Reinforcement Learning," arXiv preprint, 2025. doi: 10.48550/arXiv.2506.10889
- [12] T. Haner et al., "Distributed Quantum Computing with QMPI," arXiv preprint, 2021. doi: 10.48550/arXiv.2105.01109
- [13] "araQne: Welinq's Compiler for Distributed Quantum Computing," Welinq, 2025. 1. 15. <https://www.welinq.fr/blog/araqne-welinqs-compiler-for-distributed-quantum-computing>
- [14] S. Muralidharan, "The simulation of distributed quantum algorithms," arXiv preprint, 2025. doi: 10.48550/arXiv.2402.10745
- [15] D. Ferrari et al., "Compiler Design for Distributed Quantum Computing," *IEEE Trans. Quantum Eng.*, vol. 2, no. 4100720, 2021, pp. 1-20.
- [16] Qoro Website. <https://qoroquantum.net>
- [17] H. Shapourian et al., "Quantum Data Center Infrastructures: A Scalable Architectural Design Perspective," arXiv preprint, 2025. doi: 10.48550/arXiv.2501.05598
- [18] R. Parekh et al., "Quantum Algorithms and Simulation for Parallel and Distributed Quantum Computing," arXiv preprint, 2022. doi: 10.48550/arXiv.2106.06841
- [19] S. DiAdamo et al., "QuNetSim: A Software Framework for Quantum Networks," arXiv preprint, 2021. doi: 10.48550/arXiv.2003.06397
- [20] D. Qiu et al., "Distributed Grover's algorithm," *Theor. Comput. Sci.*, vol. 993, no. C, 2024.
- [21] H. Li et al., "Distributed exact multi-objective quantum search algorithm," arXiv preprint, 2024. doi: 10.48550/arXiv.2409.04039
- [22] T. Xue et al., "Efficient algorithms for quantum chemistry on modular quantum processors," arXiv preprint, 2025. doi: 10.48550/arXiv.2506.13332
- [23] S.M. Kim et al., "Distributed Quantum Approximate Optimization Algorithm on a Quantum-Centric Supercomputing Architecture," arXiv preprint, 2025. doi: 10.48550/arXiv.2407.20212
- [24] I. Khait et al., "Variational quantum eigensolvers in the era of distributed quantum computers," *Phys. Rev. A*, vol. 108, no. 5, 2023.
- [25] M. Hasanzadeh and A. Kargarian, "Distributed Implementation of Variational Quantum Eigensolver to Solve QUBO Problems," arXiv preprint, 2025. doi: 10.48550/arXiv.2508.17471
- [26] N.M.P. Neumann and R.S. Wezeman, "Distributed Quantum Machine Learning," arXiv preprint, 2022. doi: 10.48550/arXiv.2208.10316
- [27] M. Tejedor et al., "Distributed Quantum Circuit Cutting for Hybrid Quantum-Classical High-Performance Computing," arXiv preprint, 2025. doi: 10.48550/arXiv.2505.01184
- [28] Qdislib Website. <https://qdislib.readthedocs.io/en/latest/>
- [29] IBM Webstie. <https://www.ibm.com/quantum/qiskit>
- [30] Qibo Website. <https://qibo.science>
- [31] E. Tejedor et al., "PyCOMPSSs," *Int. J. High Perform. Comput. Appl.*, vol. 31, no. 1, pp. 66-82.
- [32] COMPSSs Website. <https://compss-doc.readthedocs.io>
- [33] <https://github.com/Interlin-q/diskit>