

## Research on the current state of practical applications and limitations of quantum computing technology

Jaehyung Kim\*

\*Professor, Dept. of Computer Software Engineering, Dongyang Mirae University, Seoul, Korea

### [Abstract]

In this paper, we identify the factors that hinder the application of quantum computing technology to solve meaningful real-world problems, and suggest related research trends and directions. To this end, we summarize the basic knowledge of quantum mechanics from the perspective of computer science, which is necessary to understand the difficulties in applying quantum computing technology, and analyze the currently commercialized quantum computers and quantum programming layers from the literature. Through an analysis of the current status and utilization of cloud-based commercial quantum computing services, we identify four factors that currently hinder the practical application of quantum computing: high barriers to entry for quantum computer programming, constraints on noisy intermediate-scale quantum computers, a still-growing open source ecosystem, and difficulties in simulating realistic problem sizes, and suggest trends and directions for related research. In doing so, it is expected to contribute to laying the groundwork for practical applications of quantum computing technology.

▶ **Key words:** Cloud computing, Quantum computing, Software Application, IT Industry, Computer architecture

### [요 약]

본 논문에서는 양자컴퓨팅 기술을 활용해 유의미한 수준의 현실 문제를 해결하는 데 있어서 응용을 어렵게 하는 요인이 무엇인지 도출하고, 관련 연구 동향과 방향성을 제시한다. 이를 위해, 양자컴퓨팅 기술을 응용하는 일의 어려움을 이해하는 데 필요한 양자 역학의 기본 지식을 컴퓨터 공학의 관점에서 정리하고, 현재 상용화된 양자 컴퓨터와 양자 프로그래밍 계층을 문헌을 통해 분석한다. 또한 클라우드 기반 상용 양자컴퓨팅 서비스의 현황과 활용 방안에 대한 분석과정을 통해, 현시점에서 양자컴퓨팅의 실용적 응용을 어렵게 하는 요인으로 양자 컴퓨터 프로그래밍에 대한 진입장벽과 노이즈가 존재하는 중규모 양자 컴퓨터에 대한 제약사항, 아직 성장 중인 오픈 소스 생태계, 현실 문제 크기에 대한 시뮬레이션의 어려움의 네 가지 요인을 도출하고 관련 연구에 대한 동향과 방향성을 제시한다. 이를 통해, 양자컴퓨팅 기술의 실용적 응용을 위한 토대를 마련하는 데 기여할 것으로 기대된다.

▶ **주제어:** 클라우드 컴퓨팅, 양자컴퓨팅, 응용 소프트웨어, IT 산업, 컴퓨터 아키텍처

- 
- First Author: Jaehyung Kim, Corresponding Author: Jaehyung Kim
  - \*Jaehyung Kim (jaehyungkim@dongyang.ac.kr), Dept. of Computer Software Engineering, Dongyang Mirae University
  - Received: 2023. 02. 22, Revised: 2023. 03. 10, Accepted: 2023. 03. 13.

## I. Introduction

양자 역학(Quantum Mechanics)의 원리를 사용하여 새로운 방식으로 연산을 처리하는 양자 컴퓨터(Quantum Computer)는 최근 몇 년 동안 학계뿐만 아니라 산업계에도 많은 화제와 관심을 불러일으키고 있다. 특히, 양자 컴퓨터가 기존의 전통적인 컴퓨터(Classical Computers)가 해결하지 못한 많은 어려운 문제를 해결할 수 있을 것으로 기대받아왔다. 하지만 양자컴퓨팅 모델은 특정 유형의 문제에서만 전통적인 컴퓨터보다 나은 성능을 보여, 지금까지는 전통적인 컴퓨터 모델 수준의 범용성을 가지지 못한다. 이에 따라 어떤 문제에서 이러한 이점을 얻을 수 있는지 조사하고 이를 위한 양자 알고리즘을 개발하는 연구가 상용 양자 컴퓨터가 출시된 지금까지도 계속해서 이어지고 있다. 하지만 현재까지 클라우드 환경 등을 통해 실제로 사용할 수 있는 양자 컴퓨터가 가진 제약사항과 양자 알고리즘을 개발하는 과정에서 기존 고전 컴퓨터 계층에서의 양자 시뮬레이션의 본질적 어려움 등으로 인해 실제 현실 문제에 적용하기가 어려운 것이 사실이다.

본 논문에서는 먼저 양자 프로그래밍 언어를 이해하고, 실사용 환경에서의 난점을 파악하기 위해 개념 수준의 양자 역학을 간단한 수학적 표현에 기반해서 설명한다. 현재 클라우드 기반의 상용 양자컴퓨팅 환경에서 서비스 형태로 제공되는 양자 컴퓨터를 실용적으로 응용하기 위해 고수준 양자 프로그래밍 언어를 사용해 양자 프로그램을 개발하는 과정에서 발생하는 제약사항과 이후의 발전 방향에 대한 통찰을 제시한다. 이 과정에서 범용적으로 접근할 수 있는 상용 양자컴퓨팅 환경에서 사용할 수 있는 양자 프로그래밍 언어를 정리하고, 양자 컴퓨터 유형에 따라 달라지는 특성과 이를 다루기 위한 양자 프로그래밍 언어의 기능과 고려사항을 분석한다. 마지막으로 상용 양자 컴퓨터의 활용 방안과 실질적인 응용 관점에서의 어려움, 향후 발전 방향에 관해 설명한다.

## II. Quantum Computing Essentials

### 1. Quantum Physics and Quantum Computer

고전적인 컴퓨터(Classic Computer)란 프로그램이라 불리는 일련의 명령어의 집합에 따라 여러 가지 연산을 실행하는 장치를 의미한다. 비슷한 방식으로 정의하자면 양자 컴퓨터(Quantum Computer)란 양자 역학의 원리를 사용해 일련의 연산을 수행하는 장치를 의미한다[1]. 양자

컴퓨터에서 실행되는 프로그램을 양자 프로그램(Quantum Program) 또는 양자 소프트웨어(Quantum Software)라고 한다. 양자 프로그램을 작성하기 위해 반드시 양자 역학과 양자 컴퓨터에 대한 지식이 필요한 것은 아니지만 양자 프로그래밍 언어는 근본적으로 양자 역학의 특성을 활용해 인터페이스를 추상화한 것이므로, 양자 연산(Quantum Operations)을 이해하는 데 도움이 된다. 기존의 양자 프로그래밍 언어에서는 양자 알고리즘(Quantum Algorithm)을 작성하기 위해 양자 역학을 설명하는 수학적 지식이 요구되었으나, 최근 수년간 개발되고 발전한 오픈소스 양자 프로그래밍 프로젝트에서 더 높은 수준의 추상화를 통해 양자 컴퓨터의 응용 분야를 넓히기 위한 변화가 이어지고 있다[2].

### 2. Quantum Bit

양자(Quantum)란 불연속적인 에너지 준위(Energy Level)를 가진 개체(Entity)이며, 더 이상 나눌 수 없는 물리적 최소 단위를 의미한다. 양자는 불연속적인 에너지 준위를 가졌기 때문에 특정한 이산적인 값으로 에너지 상태를 표현하는 것이 가능하다. 이러한 양자 역학적 원리를 응용해 정보를 표현하고 연산을 실행하는 데 사용한 장치가 양자 컴퓨터라고 할 수 있다.

양자의 이산적 특성을 적용해 구현한 양자 정보의 물리적 매개체(Physical carrier)를 큐비트(Quantum Bit, Qubit)라고 한다. 이는 양자 컴퓨터에서 정보의 최소 단위를 의미하며, 고전 컴퓨터에서의 비트(Bit)의 개념과 유사하게 0과 1의 정보를 표현할 수 있다. 다음과 같이 디랙 표기법(Dirac Notation)을 사용해 큐비트의 상태를 나타낸다. 괄호 '<>'를 의미하는 'Bracket'에서 유래한 것으로 '<'를 'Bra'로, '>'를 'Ket'으로 읽는다. 이는 양자 역학에서 양자의 상태를 표현하는 표준 표기법이다. 이를 벡터 표기법으로도 표현할 수 있는데, 이것이 양자 프로그래밍 언어에서 큐비트를 주로 표현하는 방법이다[3].

Table 1. Representing information in classical bits and qubits

Classic Bit	Qubit - Dirac Notation	Qubit - Vector Notation
0	$ 0\rangle$	$\begin{bmatrix} 1 \\ 0 \end{bmatrix}$
1	$ 1\rangle$	$\begin{bmatrix} 0 \\ 1 \end{bmatrix}$

양자 역학적 원리를 따르는 물리계(Physical System)를 통해 실제 물리적 큐비트를 구축할 수 있는데, 현재까지 상용화된 양자 컴퓨터는 초전도체(Superconductor), 이온(Ion), 실리콘 기반 양자점(Quantum Dot), 중성원자(Neutral atom) 등을 사용해 큐비트를 구현한다. 즉, 어떤 방식으로 큐비트를 구현하느냐에 따라 양자 컴퓨터의 유형이 달라진다.

큐비트가 고전적인 비트와 다른 결정적인 부분은 0 또는 1의 정보를 나타내는 바닥 상태(Ground State)와 들뜬 상태(Excited State) 외에 확률적으로 0과 1을 동시에 표현할 수 있는 중첩상태(Superposition)를 갖는다는 점이다.

중첩상태를 이해하기 위해서는 먼저 양자 역학에 대한 기본적인 이해가 필요하다. 양자 수준의 미시세계에 속한 개체는 입자성과 파동성을 동시에 가진다는 이중성(Wave-particle duality)이 성립한다. 이때 양자의 모든 물리적 정보를 파동함수(Wave function)로 표현할 수 있다. 여기서 물리적인 정보란 ‘관측할 수 있는 물리량(Observables)’이라고 표현되며, 관측할 수 있는 물리량의 종류에 따라 각각 고유상태(Eigenstate)를 가진다. 이러한 고유상태들의 합으로 파동함수가 표현될 수 있으며, 각각의 물리량에 따라 대응되는 파동함수가 존재하게 된다. 파동함수를 구성하는 고유상태를 나타내는 각 항에는 해당 상태가 관측될 확률을 나타내는 계수(Coefficients)가 있다. 이 계수는 우리가 측정이라는 행위를 하는 순간 파동함수로 표현된 다양한 고유상태 중 한 가지 고유상태가 그 물리량의 상태로 결정된다는 것을 의미하며, 이를 파동함수 붕괴(Wave function collapse)라고 한다. 이를 통해 양자의 상태는 확률론적으로 나타낼 수 있고, 측정하는 순간 수많은 고유상태 중 한 가지가 그 확률에 따라 결정된다는 사실을 알 수 있다. 이렇게 측정 이전에 수많은 고유상태가 확률론적으로 동시에 존재하는 것을 중첩이라고 한다. 큐비트의 중첩상태는 관측 전에는 0과 1이 동시에 존재할 수 있는 상태로 정의되며, 측정 결과 0 또는 1을 확률적으로 반환하는 상태를 가리킨다. 실제로 중첩상태의 물리 큐비트를 양자 프로그래밍 언어를 사용해 측정할 때마다 일정 확률로 1 또는 0의 결과를 반환한다. 이를 다음과 같이 표기할 수 있다.

$$|+\rangle = \frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 \\ 0 \end{bmatrix} + \frac{1}{\sqrt{2}}\begin{bmatrix} 0 \\ 1 \end{bmatrix} \tag{1.1}$$

$$|-\rangle = \frac{1}{\sqrt{2}}|0\rangle - \frac{1}{\sqrt{2}}|1\rangle = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 \\ 0 \end{bmatrix} - \frac{1}{\sqrt{2}}\begin{bmatrix} 0 \\ 1 \end{bmatrix} \tag{1.2}$$

위 수식 1.1의 ‘|+>’는 ‘Ket plus’로, 수식 1.2의 ‘|->’는 ‘Ket minus’로 읽는다. 이들은 위상(Phase)이 서로 π만큼 다른 중첩상태의 큐비트를 의미한다. 벡터 형태로 표기하는 것이 중첩 연산의 확률론적 원리를 이해하는 데 도움이 된다. 물론, 중첩상태에 있는 큐비트에 대해서 임의의 위상을 만드는 것도 가능하다[4]. 다음 절에서 양자 컴퓨터의 측정 과정에서 큐비트의 중첩상태가 50%의 확률로 0과 1을 반환하는 원리를 막스 보른의 규칙(Born’s Rule)에 따라 간략히 설명한다.

### 3. Measurement

측정(Measurement)이란 큐비트에 저장된 정보를 확인하는 절차를 의미한다. 고전적인 컴퓨터에서 비트의 정보를 확인하기 위해 레지스터에 저장된 정보를 언제든지 읽을 수 있는 것과 달리 양자 컴퓨터에서는 관찰자 효과(Observer Effect)로 인해 정보의 확인 과정에서 물리적인 큐비트의 상태가 바뀔 수 있다. 따라서 양자 컴퓨터에서 데이터를 확인하기 위한 측정 과정은 특별한 연산으로 취급된다. 큐비트의 상태를 측정하는 양자 프로그래밍 인터페이스를 이해하기 위해서는 먼저 큐비트의 기저 상태인 ‘|0>’과 ‘|1>’을 각각 3차원 좌표계의 Z축(Z-axis)을 기준으로 위(Up)와 아래(Down)로 구분하는 것으로 Fig. 1과 같이 나타낸다. 이 축을 기준으로 위 상태인 ‘|0>’을 90° 회전한 상태를 ‘|+>’, -90° 회전한 상태를 ‘|->’로 나타낸다.

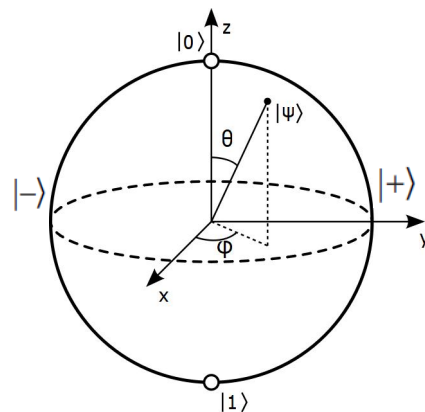


Fig. 1. Representing Qubit States on a Bloch Sphere

앞서 언급한 물리량 측정 과정에서 양자는 파동함수의 붕괴에 따라 특정 고유상태가 되어야 하므로 이때 특정 고유상태가 측정될 확률을 나타내는 파동함수의 계수 제곱의 합은 항상 1이 되도록 정규화된 확률 진폭(Probability Amplification)이 수식 1.3과 같이 성립한다. 이를 보른의 규칙이라 한다. 이때 계수는 실수 외에 복소수를 포함하므로 이를 3차원 좌표계에 시각화하면 Fig. 1과 같은 블로흐 구면으로 나타낼 수 있다.

$$|\alpha|^2 + |\beta|^2 = 1 \quad (1.3)$$

큐비트의 상태는 기본적으로 ‘|0>’과 ‘|1>’의 기저벡터(Basis vector)를 통해 표현되고, 측정이란 두 기저 중 어디에 더 가까운 위치에 큐비트가 위치하는지로 나타낼 수 있다. 이는 Z축의 기저벡터인 ‘|0>’ 또는 ‘|1>’ 대한 내적(Inner Product)을 통해 측정 대상인 큐비트의 Z축에 대한 정사영을 도출할 수 있으며, 수식 1.3에서처럼 내적 결과에 대한 절대값의 제곱이 확률이므로, 이를 통해 큐비트의 상태를 측정한 결과를 확인할 수 있다. 측정을 표기할 때는 디랙 표기법의 ‘Bra’를 통해 표현되며, 측정하려는 기저벡터에 대해 다음과 같이 나타낼 수 있다.

<기저 | 측정대상 큐비트의 상태>

예를 들어, ‘|+>’인 큐비트를 측정했을 때 관측값이 1일 확률은 ‘<1|+>’로 나타낼 수 있다. 여기서 ‘Bra’ 부분은 벡터의 전치(Transpose)를 의미한다. 다음은 측정을 통해 ‘|+>’인 큐비트로부터 1을 얻을 확률을 나타낸 수식이다.

$$\begin{aligned} |1\rangle &= \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \langle 1| = [0 \ 1], \\ \langle 1|+\rangle &= \langle 1| \frac{(|0\rangle + |1\rangle)}{\sqrt{2}} \quad (1.4) \\ &= \frac{1}{\sqrt{2}} \left( [0 \ 1] \begin{bmatrix} 1 \\ 1 \end{bmatrix} \right) = \frac{1}{2} \end{aligned}$$

여기서 중요한 것은 측정 결과는 항상 고전적인 비트값이라는 점이다. 실제로 양자 프로그래밍 언어를 사용해 중첩상태의 큐비트를 측정하면 0 또는 1중 하나의 값을 반환하지만, 반복해서 실행하면 반환되는 0과 1의 비율이 50:50에 수렴하게 된다. 이를 위해, 양자 컴퓨터에서 기저 상태나 중첩상태에 있는 큐비트를 인위적으로 만든다.

Table 2. Returns and probabilities measured on an underlying basis

Qubit state	Basis	Probability against basis	Measured Classic bit
0>	0>	100%	0
0>	1>	0%	
1>	0>	0%	1
1>	1>	100%	
+>	0>	50%	0 or 1
+>	1>	50%	
->	0>	50%	
->	1>	50%	

4. Operations on Multiple Qubits and Entanglement

고전적인 컴퓨터가 비트로 표현된 데이터에 대해 연산을 수행하기 위해서는 레지스터가 필요하다. 양자 컴퓨터를 이용해 실질적인 문제를 처리하기 위해서는 마찬가지로 양자 레지스터(Quantum Register)를 통해 다수의 큐비트에 대한 연산을 수행할 수 있어야 한다. 다수의 큐비트를 다루는 양자컴퓨팅의 방식은 양자 역학의 얽힘(Entanglement) 현상을 활용하는 것이다. 얽힘이란 특정 양자의 상태를 측정하는 행위가 또 다른 양자의 상태를 즉각적으로 결정짓는다는 개념으로, 이러한 상태에 있는 두 양자를 서로 얽혀있다고 표현한다. 단, 특정 양자의 상태가 미리 결정된 것이 아니라 측정하는 순간 결정되고, 이와 얽혀있는 다른 양자 역시 그 순간 확률적으로 상태가 결정된다. 서로 얽혀있는 양자의 상태는 벨 상태(Bell's State)에 따라 서로의 상태를 결정지을 수 있어야 하며, 이것이 양자 프로그래밍 언어에서 얽힘 현상을 응용할 때 그대로 적용된다. 결국, 양자 컴퓨터는 중첩과 얽힘을 활용해 연산을 수행해서 계산 복잡도를 낮추는 방식인 양자 병렬성(Quantum Parallelism)으로 고전적인 컴퓨터가 가진 연산 능력 이상의 성능을 보일 수 있다[5]. 큐비트에 대한 연산(Quantum Operation)이란 큐비트의 상태를 변화시키는 명령으로 벡터로 표현된 큐비트의 상태, 즉, 위상을 변화시키는 행렬 연산으로 설명할 수 있다. 양자 프로그래밍 언어에서 함수는 이러한 유형의 행렬 연산을 의미한다. 고전적인 컴퓨터에서 회로(Circuits)와 게이트(Gates)를 이용해 비트를 다루는 것처럼, 양자 컴퓨터 역시 큐비트를 다룰 때 양자 프로그래밍 언어를 사용해 회로와 게이트를 작성한다. 여기서 게이트가 행렬 연산에 해당하며, 결국 큐비트를 다루기 위해 큐비트와 게이트의 회로를 구성하고 측정하는 행위를 계속하는 일종의 네트워크를 구성하는 프로그램을 작성하여 현실 문제에 응용할 수 있게 된다.

### III. Commercial Quantum Computers

#### 1. Commercial Quantum Computers

실험실 속에서만 존재하던 양자 컴퓨터는 위에서 언급한 일정 개수 이상의 큐비트를 양자 역학의 원리에 따라 인위적으로 제어할 수 있게 되면서 본격적으로 상용화되기 시작했다. 현재까지 상용화된 양자 컴퓨터는 큐비트를 이용해 양자 연산을 수행하는 방법에 따라 크게 양자 게이트(Quantum Gate) 기반 범용 양자 컴퓨터와 양자 어닐링(Quantum Annealing) 기반의 특수목적용 양자 컴퓨터인 단열 양자 컴퓨터(Adiabatic Quantum Computer)로 나뉜다.

##### 1.1 Quantum Computer Architecture

여기서는 일반적인 개념의 양자 컴퓨터 아키텍처를 다루도록 한다.



Fig. 2. Quantum Computer Architecture

Fig. 2에서 양자 프로그래밍 언어를 사용해 개발한 양자 프로그램을 고전적인 CPU(Central Processing Unit)에서 명령어로 전환하고, 이를 양자 장치(Quantum Hardware)에서 실행할 수 있도록 신호(Signals)로 전환하는 DSP(Digital Signal Processor)를 거쳐 전달된다. 양자 장치에서 실행되는 명령어는 양자의 상태를 변환하고, 양자의 상태를 측정하는 일련의 작업으로 구성되어 있고, 최종적으로 사용자가 확인하길 원하는 내용을 결과로 전달받을 수 있다. 다음 절에서 양자 장치를 구성하는 방법에 따라 달라지는 두 가지 유형의 양자 컴퓨터를 다루도록 한다.

##### 1.2 Gate based General-Purpose Quantum Computers

범용 양자 컴퓨터(General purpose Quantum Computer)란 광범위한 유형의 양자 알고리즘을 프로그래밍 언어로 개발하고 이를 수행할 수 있는 컴퓨팅 시스템을 의미한다[6]. 범용 양자 컴퓨터는 앞서 언급한 물리적 큐비트를 실제로 구현하고, 중첩과 얽힘이 의도적으로 동작할 수 있도록 안정적인 상태를 구성할 수 있어야 한다. 큐비트의 상태를 게이트라는 함수를 통해 변환하고, 이로부터 결과를 측정하는 능력을 지닌 범용 양자 컴퓨터를 게이트 기반 양자 컴퓨터라고 한다.

미시세계 수준의 양자는 외부 자극에 민감하므로 큐비트의 상태를 지속해서 유지하는 것이 어렵다. 특히 동시에 중첩하거나 얽히게 만드는 큐비트의 수가 늘어나고 회로가 깊어질수록 노이즈(Noise)가 발생할 확률이 높아진다. 현재 상용화된 양자 컴퓨터는 이러한 어려움을 완벽하게 해결하지 못하고 큐비트를 다루는 과정에서 발생하는 오류를 수정(Error Correction)하는 추가적인 절차가 필수적이다. 이러한 유형의 범용 양자 컴퓨터를 노이즈 중간 규모 양자 컴퓨터(Noisy Intermediate-Scale Quantum computers, NISQ computers)라고 한다[7].

2019년 구글(Google)에서 발표한 범용 양자 컴퓨터인 시커모어(Sycamore)가 기존의 고전적인 컴퓨터 아키텍처를 가진 슈퍼컴퓨터의 성능으로도 단시간 내에 수행하기 어려운 문제를 실용적으로 사용할 수 있을 정도의 시간 내에 처리하는 양자 우위(Quantum Advantage)를 보였다고 발표했다[8]. 하지만 이어진 논쟁에서 양자 컴퓨터의 우월성에 대한 반박이 이어졌으며[9], 현재까지의 결론은 실제 실용적인 문제를 해결하는 관점에서는 여전히 양자 우위를 보이는 것에 대해서는 어려움이 있는 상태라고 할 수 있다. 현재 상용화된 범용 양자 컴퓨터에 대해서는 이후 클라우드 기반 양자컴퓨팅 서비스에서 함께 다루도록 한다.

##### 1.3 Adiabatic Quantum Computers

어닐링(Annealing)이란 재료의 특성을 변경하는 데 사용되는 열처리 공정을 의미하는 용어에서 유래한 것으로 그 과정에서 금속 온도가 서서히 낮아지는 방식과 유사하게 양자 시스템의 에너지가 최소 에너지 상태에 도달하기 위해서 서서히 낮아지는 과정을 은유적으로 표현한 용어다. 최초의 상용 양자 어닐링 기반 양자 컴퓨터인 D-Wave의 양자 컴퓨터는 초전도체를 이용해 만들어진 루프에서 최소 에너지를 갖는 상태와 방향을 통해 큐비트의 기저를 표현한다[10-11]. 이 유형의 양자 컴퓨터가 갖는 물리적 특성과 그에 대한 모델링의 결과로 국소적인 최소값을 찾는 최적화 문제에 활용될 수 있다[12]. 게이트 기반의 양자 컴퓨터가 갖는 범용성을 가지지 못하고 특정 유형의 연산만 수행할 수 있어 활용범위가 상대적으로 적은 대신에 앞서 언급한 큐비트의 중첩과 얽힘 상태를 많은 수의 큐비트들에 대해서도 안정적으로 유지할 수 있으므로 특정 유형의 현실 문제를 해결하는 데 더 용이할 수 있다. 앞서 II-4에서 언급한 큐비트의 연산은 게이트 기반 범용 양자 컴퓨터를 기준으로 설명한 것이며, 단열 양자 컴퓨터는 게이트 대신 큐비트의 에너지 상태, 즉, 해밀토니안(Hamiltonian)을 변화시키기 위한 연산을 수행해서 큐비트를 제어할 수 있다.

Table 3. Quantum devices with cloud quantum computing services

Manufacturer	QPU	Model	# of qubits	Architecture	Amazon	Azure
IonQ	Harmony	-	11	Trapped Ion	0	0
	Aria	-	23	Trapped Ion	X	0
QQC (Oxford Quantum Circuits)	Lucy	-	8	Superconducting	0	X
Quera	Aquila	-	256	Neutral Atom	0	X
Rigetti	Aspen-M	Aspen-M-2	80	Superconducting	0	0
		Aspen-M-3	80	Superconducting	0	0
Xanadu	Borealis	-	216	Photonics	0	X
D-Wave (Amazon Marketplace)	2000Q	-	2048	Superconducting	0	X
	Advantage QPU	Advantage 1.1	5760	Superconducting	0	X
		Advantage 4.1	5760	Superconducting	0	X
Quantinuum	H1	H1-1	20	Trapped Ion	X	0
		H1-2	12	Trapped Ion	X	0
Quantum Circuits, Inc.	N/A				X	Planned
PASQAL	N/A				X	Planned

## 2. Quantum Computing Services on Cloud

2019년 1월 최초의 게이트 기반 상용 양자 컴퓨터인 IBM의 Quantum System One이 출시된 이후로, 이온트랩(Ion Trap)을 이용한 IonQ, 초전도체 활용한 Rigetti, 포토닉스(Photonics)를 활용한 Xanadu 등의 상용 양자 컴퓨터가 출시되었다[13-16]. 이들은 모두 NISQ 유형의 양자 컴퓨터로 사용자들은 클라우드 서비스를 통해 일종의 공유자원으로 양자 컴퓨터를 사용할 수 있다. 서비스형 인프라(Infrastructure as a Service) 대신 서버리스(Serverless) 방식으로 사용한 만큼 비용을 지불하는 종량제 방식(Metered Payment)으로 제공된다. 양자 컴퓨터를 구축하는 데 드는 비용을 차치하고서라도, 인위적으로 물리적 큐비트의 상태를 유지하기 위해서는 극저온의 환경(Near-absolute zero)이 필요하다. 이러한 환경을 제공할 수 있는 시설이 데이터 센터가 될 수 있으며, 클라우드 서비스 제공자(Cloud Service Provider)가 양자컴퓨팅 서비스를 제공하는 것은 합리적 귀결이라고 할 수 있다. 물론, 대부분의 상용 양자 컴퓨터 제조사가 자체적으로 데이터 센터를 운영하며 클라우드 환경을 구축하여 서비스를 제공하기도 한다. Table 3은 대표적인 클라우드 서비스 제공자인 AWS(Amazon Web Services)와 Microsoft Azure에서 사용 가능한 상용 양자 컴퓨터를 나타낸다 [17-23]. 이 내용은 클라우드 서비스 제공자 중 직접 양자 컴퓨팅 서비스를 상용으로 제공하는 경우를 대상으로 선정하였다.

## 3. Quantum Software Stack

클라우드 서비스를 통해 일반 사용자들이 사용 가능한 범용 혹은 특수 목적용 양자 컴퓨터를 활용하기 위해서는 양자 알고리즘을 양자 프로그래밍 언어(Quantum Programming Language)를 통해 작성해야 한다. Fig. 3은 양자 컴퓨터를 다루기 위한 소프트웨어 스택을 나타낸다. 양자 프로그래밍 언어는 기존의 프로그래밍 언어와는 달리 양자 하드웨어를 직접적으로 다루기 위한 연산이므로 양자 연산에 대한 수학적 이해가 필수적이었다. 하지만 상용화 과정에서 양자 프로그래밍 언어를 추상화하고, 진입장벽을 낮추기 위한 시도가 이어져 오픈소스 언어인 Qiskit, Q# 등을 통해 양자 프로그램을 더욱 쉽게 작성할 수 있게 되었다. 하지만 클라우드 서비스는 모든 연산이 인터넷을 통한 API 호출로 구성되어 있고, 이에 대한 SDK(Software Development Kit)이 제공되고 있어 이를 통해 양자 연산을 실행할 필요가 있다. 이를 위해, 각 클라우드 서비스 제공자들은 자체적으로 프레임워크(Framework)를 제작하여 양자 프로그래밍 언어와 연결을 지원한다. Azure Quantum의 경우 QDK(Quantum Development Kit), AWS Braket의 경우에는 AWS Braket SDK와 양자 프로그래밍 언어별로 프로바이더(Provider)를 제공해 직접 양자컴퓨팅 서비스에 알고리즘을 실행할 수 있도록 제출할 수 있게 된다.

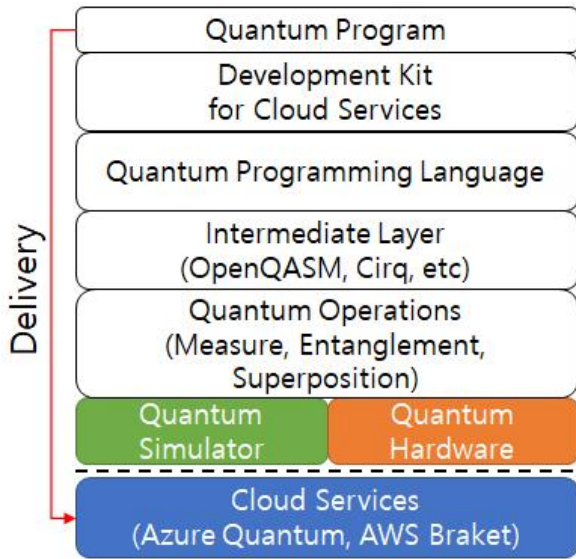


Fig. 3. Software Stack on Quantum Computer

## IV. Applications and Limitations

### 1. Industry Applications and Trends

앞서 언급한 실제 현실에서의 문제에 대한 양자 우월성에 대한 논쟁은 현시점을 기준으로 여전히 진행 중이다. 또한 현 수준의 상용 양자 컴퓨터를 유의미한 수준의 실제 문제를 해결하는 데 활용되기에는 많은 제약사항이 장벽으로 존재한다. 현재까지 방법적 측면에서는 주로 시뮬레이션을 통해 작은 수준의 문제를 해결하는 방식으로 검증을 수행하거나, 혹은 수학적 증명을 통한 잠재성을 강조하는 시도가 계속되고 있다. 응용 분야 측면에서는 보안(Security), 소재(Chemistry of Materials) 및 제약(Drugs), 은행(Banking) 및 금융(Finance), 첨단 제조업(Manufacturing) 등의 분야에서 양자 알고리즘을 통해 해결할 문제의 유형을 발견하려는 연구가 계속되고 있다 [24-28]. 다만, 이러한 문제 해결 유형은 기존에 쇼어(Shor)[29]와 그로버(Grover)[30]가 제안한 양자 알고리즘 유형에 기반한 일정한 분류 체계를 유지하고 있어 많은 연구에서 기존의 문제를 이러한 유형의 문제로 변환하는 방식으로 접근이 이뤄지고 있다.

최근에는 양자 프로그램에서 고전적인 컴퓨터에서 처리해야 할 부분과 양자 컴퓨터에서 처리해야 할 부분을 나눌 수 있는 하이브리드 프로그램을 지원하는 플랫폼이 늘고 있다. AWS Braket에서는 하이브리드 작업(Hybrid Jobs) 기능을 통해 하이브리드 양자-고전 알고리즘을 지원하고, 양자 컴퓨터의 성능을 최적화할 수 있다[31].

### 2. Challenges of utilizing quantum computing technology

이번 절에서는 현시점에서 양자 컴퓨터의 활용을 어렵게 하는 현존하는 요인들을 살펴본다.

#### 1.1 High Barrier to Entry

양자 프로그래밍 언어의 발전으로 비교적 높은 수준의 추상화를 통해 수학적 지식에 대한 요구가 크게 줄어들었으나, 여전히 양자 연산을 이해하기 위해서는 양자 역학과 수학에 대한 지식이 필요하다. 또한, 실제 문제에 양자 알고리즘을 적용하기 위해서는 기존의 문제를 양자 컴퓨터의 특성에 맞게 변환하는 작업을 거쳐야 하는데 이는 알고리즘에 대한 깊은 이해를 요구한다. 이러한 어려움을 해결하기 위해 새로운 추상화 계층을 통해 양자컴퓨팅 기술을 활용하기 위한 연구가 이어지고 있다[32-33].

#### 1.2 Quantum Hardware Limitations

클라우드 기반의 상용 양자컴퓨팅 서비스는 양자 컴퓨터를 공유자원(Shared Resource) 형태로 제공한다. 사용자가 전달한 양자 프로그램을 즉각적으로 실행하는 대신 큐(Queue)에 저장하고, 차례가 되면 프로그램을 실행한 후 그 결과를 사용자에게 반환한다. 따라서 현 상황에서는 자원을 점유해서 사용하기 어렵다는 제약이 있다. 또한, 많은 경우에 양자 컴퓨터의 노이즈를 줄이고 안정적인 상태를 유지하기 위한 정기적인 유지관리(Maintenance) 작업으로 인해 24시간 양자 컴퓨터 서비스를 운영하는 것이 어렵다는 제약도 존재한다. 향후 노이즈 중간규모 양자컴퓨터를 넘어서는 새로운 장치를 통해 이러한 제약들이 해결될 것으로 기대된다.

#### 1.3 Open Source Limitations

앞서 언급한 높은 진입장벽과 상용 양자 컴퓨터의 실제 현실 문제에 관한 응용 사례의 부재로 자연스럽게 오픈소스 생태계 규모가 상대적으로 작은 편이다. 학계와 산업계에서 큰 노력으로 상용 양자 컴퓨터의 하드웨어와 소프트웨어 스택을 표준화하고, 양자 프로그램의 개발과정을 체계화하려고 노력하고 있으나, 아직 많은 부분이 논의와 연구 수준에 머무르고 있다[34-36].

#### 1.4 Difficulties of simulation

상용 양자 컴퓨터가 도입되기 전부터 양자 알고리즘을 양자 컴퓨터의 방식으로 시뮬레이션하기 위한 시도가 양자 시뮬레이터(Quantum Simulator)를 발전시켰다. 다만,

양자 역학의 근본적인 원리에서 기인한 문제점으로 다중 큐비트를 기존의 고전적인 컴퓨터 환경에서 구현하기 위해서는 기하급수적인 메모리 공간을 요구한다는 제약이 있다. 예를 들어, 단순한 방식으로 50큐비트를 시뮬레이션하기 위해서는 약 16페타바이트(Petabytes)의 메모리 공간이 필요하다. 이런 문제를 해결하기 위해 다양한 방식으로 메모리 사용량을 줄이고, 시뮬레이션을 가속하려는 연구가 계속되고 있다. 특히, GPU(Graphics Processing Unit)를 활용해 CPU 병목을 해소하고, 대규모 클러스터를 통해 문제를 분할하고 필요한 메모리 공간을 확보하는 방식 등이 제시되고 있다[37-40].

## V. Conclusions

다양한 유형의 상용 양자 컴퓨터를 활용해 실제 의미 있는 현실 문제를 해결하려는 시도는 기존의 고전적인 컴퓨터 환경에서 하드웨어와 소프트웨어 분야가 발전한 방식과 유사한 흐름을 갖고 있다. 좋은 양자 프로그램을 개발하기 위한 소프트웨어 공학적인 측면에서부터 오픈소스 기반의 양자 프로그래밍 언어의 발전에 이르기까지 기존에 걸어왔던 길을 다시 한번 걸으려 하고 있다. 양자 컴퓨터의 특성에 기반한 프로그래밍 모델은 범용 양자 컴퓨터라 하더라도 구조적인 제약을 벗어날 수 없다. 특히 고전적인 컴퓨터가 가진 범용성으로 인해 양자 컴퓨터의 활용 목표는 기존에 해결하지 못한 문제에 입각하고 있으므로 주어진 목표가 높은 것이 사실이다. 양자 컴퓨터의 규모가 지금보다 커지고, 중첩과 얽힘 등의 상태를 안정적으로 유지할 수 있을 것을 가정한 다양한 연구와 시도는 그 시기가 왔을 때 딥러닝이 가져온 세상의 변화를 다시 한번 일으키는 시금석이 될 것으로 예상되므로, 연구 기회는 계속해서 생겨날 것으로 기대된다.

## ACKNOWLEDGEMENT

This work was supported by DMU(Dongyang Mirae Univ.) Research Grant.

## REFERENCES

- [1] Feynman, Richard P., "Feynman and computation," CRC Press, pp. 133-153, 2018. DOI: 10.1201/9780429500459-11
- [2] E.Y. Cho, et al., "State-of-the-art in Quantum Computing Software," Electronics and Telecommunications Trends, Vol. 36, No. 6, 2021. DOI: 10.22648/ETRI.2021.J.360607
- [3] Michael A. Nielsen and Isaac Chuang, "Quantum computation and quantum information," American Journal of Physics, Vol. 70, No. 5, pp. 558, 2002. DOI: 10.1119/1.1463744
- [4] Grover, Lov, and Terry Rudolph, "Creating superpositions that correspond to efficiently integrable probability distributions," arXiv preprint quant-ph/0208112, 2002. DOI: 10.48550/arXiv.quant-ph/0208112
- [5] Gill, Sukhpal Singh, et al., "Quantum computing: A taxonomy, systematic review and future directions," Software: Practice and Experience, Vol. 52, No. 1, pp. 66-114, 2022. DOI: 10.1002/spe.3039
- [6] DiVincenzo, David P., "The physical implementation of quantum computation," Fortschritte der Physik: Progress of Physics Vol. 48, No. 9-11, pp. 771-783, 2000. DOI: 10.48550/arXiv.quant-ph/0002077
- [7] Preskill, John., "Quantum computing in the NISQ era and beyond," Quantum, Vol. 2, pp. 79, 2018. DOI: 10.22331/q-2018-08-06-79
- [8] Arute, Frank, et al., "Quantum supremacy using a programmable superconducting processor," Nature, Vol. 574, No. 7779, pp. 505-510, 2019. DOI: 10.5061/dryad.k6t1rj8
- [9] Pan, Feng, Keyang Chen, and Pan Zhang., "Solving the sampling problem of the sycamore quantum circuits," Physical Review Letters, Vol. 129, No. 9, 2022. DOI: 10.1103/physrevlett.129.090502
- [10] Albash, Tameem, and Daniel A. Lidar., "Adiabatic quantum computation," Reviews of Modern Physics, Vol. 90, No. 1, 2018. DOI: 10.1103/revmodphys.90.015002
- [11] Johnson, Mark W., et al., "Quantum annealing with manufactured spins," Nature, Vol. 473, No. 7346, pp. 194-198, 2011. DOI: 10.1038/nature10012
- [12] Rajak, Atanu, Sei Suzuki, Amit Dutta, and Bikas K. Chakrabarti., "Quantum annealing: an overview," Philosophical Transactions of the Royal Society A, Vol. 381, No. 2241, 2023. DOI: 10.1098/rsta.2021.0417
- [13] IBM Research, IBM Quantum System One, <https://research.ibm.com/interactive/system-one/>
- [14] Egan, Laird, et al., "Fault-tolerant operation of a quantum error-correction code," arXiv preprint arXiv:2009.11482, 2020. DOI: 10.48550/arXiv.2009.11482
- [15] Rigetti, Rigetti official Site, <https://www.rigetti.com>
- [16] Xanadu, Xanadu official site, <https://www.xanadu.ai>



- [17] Amazon Web Services, AWS Braket, <https://docs.aws.amazon.com/braket/latest/developerguide/>
- [18] Microsoft, Azure Quantum, <https://azure.microsoft.com/en-us/products/quantum>
- [19] Catherine McGeoch and Pau Farre, "The D-Wave Advantage System: An Overview," D-Wave Technical Report Series 14-1049A-A, 2020.
- [20] Rigetti Systems, Aspen-M-2/Aspen-M-3 Processors, <https://qcs.rigetti.com/qpus>
- [21] Madsen, Lars S., et al., "Quantum computational advantage with a programmable photonic processor," *Nature*, Vol. 606, No. 7912, pp. 75-81, 2022. DOI: 10.1038/s41586-022-04725-x
- [22] IonQ, IonQ Aria, <https://ionq.com/quantum-systems/aria>
- [23] IonQ, IonQ Harmony, <https://ionq.com/quantum-systems/harmony>
- [24] Bova, Francesco, Avi Goldfarb, and Roger G. Melko, "Commercial applications of quantum computing," *EPJ quantum technology*, Vol. 8, No. 2, 2021. DOI: 10.1140/epjqt/s40507-021-00091-1
- [25] Bayerstadler, Andreas, et al., "Industry quantum computing applications," *EPJ Quantum Technology*, Vol. 8, No. 1, pp. 25, 2021. DOI: 10.1140/epjqt/s40507-021-00114-x
- [26] Egger, Daniel J., et al., "Quantum computing for finance: State-of-the-art and future prospects," *IEEE Transactions on Quantum Engineering*, Vol. 1, pp. 1-24, 2020. DOI: 10.1109/TQE.2020.3030314
- [27] Blunt, Nick S., et al., "Perspective on the Current State-of-the-Art of Quantum Computing for Drug Discovery Applications," *Journal of Chemical Theory and Computation*, Vol. 18, No. 12, pp. 7001-7023, 2022. DOI: 10.1021/acs.jctc.2c00574
- [28] Sales, Juan Francisco Ariño, and Raúl Andres Palacios Araos. "Adiabatic Quantum Computing for Logistic Transport Optimization," arXiv preprint arXiv:2301.07691, 2023. DOI: 10.48550/arXiv.2301.07691
- [29] Wong, Hiu Yung., "Shor's Algorithm," *Introduction to Quantum Computing: From a Layperson to a Programmer in 30 Steps*. Cham: Springer International Publishing, pp. 279-288, 2022. DOI: 10.1007/978-3-030-98339-0\_29
- [30] Baaquie, Belal Ehsan, and Leong-Chuan Kwek., "Grover's Algorithm," *Quantum Computers: Theory and Algorithms*. Singapore: Springer Nature Singapore, pp. 185-202, 2023. DOI: 10.1007/978-981-19-7517-2\_11
- [31] Danilo Poccia, *Introducing Amazon Braket Hybrid Jobs - Set Up, Monitor, and Efficiently Run Hybrid Quantum-Classical Workloads*, <https://aws.amazon.com/ko/blogs/korea/introducing-amazon-braket-hybrid-jobs-set-up-monitor-and-efficiently-run-hybrid-quantum-classical-workloads/>
- [32] Duckering, Casey. "New Abstractions for Quantum Computing," PhD diss., The University of Chicago, 2022. DOI: 10.48550/arXiv.2303.02578
- [33] Cobb, Adrian, Jean-Guy Schneider, and Kevin Lee., "Towards higher-level abstractions for quantum computing," In *Australasian Computer Science Week 2022*, pp. 115-124. 2022.
- [34] Bock, Sebastian, Raphael Seidel, and Colin Kai-Uwe Becker., "Towards a Standardised Quantum Software Stack," <https://ercim-news.ercim.eu/en128/special/towards-a-standardised-quantum-software-stack>
- [35] Openja, Moses, et al., "Technical debts and faults in open-source quantum software systems: An empirical study," *Journal of Systems and Software*, Vol. 193, 2022. DOI: 10.1016/j.jss.2022.111458
- [36] Zhao, Jianjun., "Quantum software engineering: Landscapes and horizons," arXiv preprint arXiv:2007.07047, 2020. DOI: 10.1016/j.jss.2022.111458
- [37] Chen, Zhao-Yun, et al., "64-qubit quantum circuit simulation," *Science Bulletin*, Vol. 63, No. 15, pp. 964-971, 2018. DOI: 10.1016/j.scib.2018.06.007
- [38] Gutierrez, Eladio, et al., "Parallel quantum computer simulation on the CUDA architecture," *Computational Science-ICCS 2008: 8th International Conference, Part I-8* pp. 700-709, Kraków, Poland, June 23-25, 2008. DOI: 10.1007/978-3-540-69384-0\_75
- [39] Gutiérrez, Eladio, et al., "Quantum computer simulation using the CUDA programming model," *Computer Physics Communications*, Vol. 181, No. 2, pp. 283-300, 2010. DOI: 10.1016/j.cpc.2009.09.021
- [40] Heng, Sengthai, Taekyung Kim, and Youngsun Han., "Exploiting GPU-based Parallelism for Quantum Computer Simulation: A Survey," *IEEE Transactions on Smart Processing & Computing*, Vol. 9, No. 6, pp. 468-476, 2020. DOI: 10.5573/IEIESPC.2020.9.6.468

## Authors



Jaehyung Kim received the B.S., M.S. in Computer Science and Engineering from Hanyang University in 2007 and 2009 respectively. He received Ph.D. degrees in Computer Science from Yonsei University Korea, in 2018.

Dr. Kim joined the faculty of the Department of Computer Software Engineering at Dongyang Mirae University, Seoul, Korea, in 2021. He is currently an Assistant Professor in the Department of Computer Software Engineering, Dongyang Mirae University. He is interested in cloud computing, database and storage technology, parallel computing.