QAOA 양자회로 절단 시 개별 회로 병렬 실행을 통한 시뮬레이션 시간 단축 연구

최지훈, 권영우
경북대학교

nob1ess@knu.ac.kr, ywkwon@knu.ac.kr

Reducing QAOA Simulation Time with Parallel Execution of Cut Circuits

Jihoon Choi, Young-Woo Kwon
Kyungpook National University

요 약

현대 양자 시뮬레이션에서 실행 시간은 중요한 성능 지표이다. 양자회로 절단은 실제 양자 디바이스의 제약을 완화하지만, 시뮬레이션에서는 다수의 하위 회로 처리로 인한 오버헤드를 발생시켜 실행 시간을 증가시킬 수 있다. 본 논문은 QAOA(Quantum Approximate Optimization Algorithm) 양자회로에 회로 절단 기법을 적용하고, 생성된 다수의 하위 회로들을 RunPod 의 서버리스 워커(serverless worker)에 분산하여 병렬 실행함으로써 시뮬레이션 시간을 단축하는 연구를 제시한다. 10-큐비트 QAOA 회로를 대상으로 실험한 결과, 회로 절단으로 인한 오버헤드를 효과적으로 극복하고 대규모 양자회로 시뮬레이션의 효율성을 크게 향상시킬 수 있음을 확인하였다.

Ⅰ. 서 론

양자컴퓨팅은 특정 문제에 대해 고전 컴퓨팅을 능가하는 잠재력을 지닌 기술로 주목받고 있으며, 특히 조합 최적화 문제 해결에 효과적인 QAOA[1]는 NISQ(Noisy Intermediate-Scale Quantum) 시대의 주요 알고리즘 중 하나이다. 그러나 현재 양자 컴퓨터는 큐비트 수와 연결성의 한계로 복잡한 QAOA 회로를 직접 실행하기 어렵다.

이러한 제약을 극복하기 위해 양자회로 절단(quantum circuit cutting)[2] 기술이 제안되었다. 이 기술은 큰 양자 회로를 여러 개의 작은 하위 회로로 분할하고, 각 하위 회로를 독립적으로 실행한 후 결과를 재구성하여 원래 회로의 결과를 얻는 방식이다[3]. 이는 제한된 양자 하드웨어에서 더 큰 규모의 문제를 다룰 수 있게 해주지만, 고전 시뮬레이터 환경에서는 다수의 하위 회로를 순차적으로 처리할 경우 오히려 전체 실행 시간이 증가하는 오버헤드가 발생한다.

본 연구에서는 이러한 시뮬레이션 오버헤드를 극복하기 위해, 절단된 하위 회로들을 다수의 독립적인 컴퓨팅 자원, 구체적으로 클라우드에 분배하여 병렬로 실행하는 방안을 탐구한다. PennyLane 프레임워크를 활용하여 QAOA 양자 회로를 구현하고 절단한 후, RunPod 환경에서 병렬 시뮬레이션을 수행하여 실행 시간 효과를 실험 및 분석한다.

Ⅱ. 본론

1. 연구방법

A. QAOA 알고리즘

본 연구에서는 조합 최적화 문제의 대표적인 예시인 MaxCut 문제[4] 해결을 위해 10 개의 큐비트를 사용하는 QAOA 를 적용한다. QAOA 회로는 초기 상태 준비, 문제 해밀토니안과 믹서 해밀토니안으로 구성된 QAOA 레이어 반복 적용, 그리고 결과 측정 단계로 구성된다. 본 연구에서는 QAOA 레이어 2 개(문제 해밀토니안 적용 1 회, 믹서 해밀토니안 적용 1 회)를 하나의 depth 로 간주한다.

B. 양자회로 절단 구현

양자회로 절단은 PennyLane 의 qcut 모듈을 활용하여 구현한다. QAOA 회로는 각 QAOA depth(즉, 2 개의 연산자 레이어) 적용 후 Wirecut 을 통해 절단된다. 예를 들어, QAOA depth 가 p 이면, 총 2p 개의 하위 회로가 생성될 수 있다 (단, 실제 분할 방식에 따라 달라질 수 있음. 본 연구에서는 depth p 일 때 2p 개로 분할). 생성된 다수의 하위 회로들은 RunPod 의 서버리스 워커에 분산되어 독립적으로 병렬 실행된다.

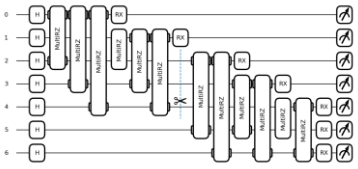

그림 1. QAOA 회로 절단 예시 시각화

C. 실행 시간 측정 방법

회로의 전체 시뮬레이션 시간은 모든 하위 회로들의 병렬 처리가 완료될 때까지 소요된 총 시간으로 측정한다. 이는 RunPod 환경에서 지정된 수의 워커를 사용했을 때의 실제 총 처리 시간(Total Processing Time)을 의미하며, 이 시간을 통해 병렬 처리의 효율성을 평가한다.

2. 평가

A. 실험 설정

실험은 10-큐비트 MaxCut 문제 해결용 QAOA 회로를 기반으로 하며, RunPod 서버리스 환경에서 PennyLane 을 사용하여 수행되었다.

실험 1: QAOA depth 4 (총 8 개의 하위 회로 생성)로 고정하고, 3 개의 RunPod 워커를 사용하여 측정 횟수(shots)를 1 회부터 10,000,000 회까지 변경하며 총 실행 시간을 측정했다 (그림 3).

실험 2: QAOA depth 10 (총 20 개의 하위 회로 생성)으로 설정하고, 측정 횟수를 100,000 회, 1,000,000 회, 10,000,000 회로 각각 고정한 상태에서 RunPod 워커 수를 1 개부터 10 개까지 늘려가며 총 실행 시간의 변화를 관찰했다 (그림 4).

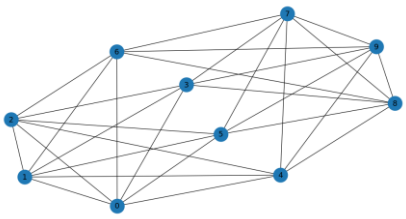

그림 2 MaxCut 문제(10nodes) 그래프 시각화

B. 실험 결과

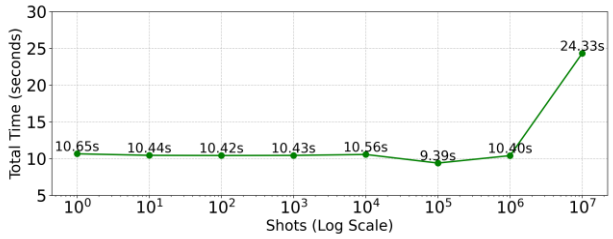

그림 3. Depth 길이 별 전체/개별 실행시간 차이

그림 3 는 QAOA depth 4, 고정된 3 개의 워커 환경에서 측정 횟수에 따른 총 실행 시간을 보여준다. 예상대로 측정 횟수가 증가함에 따라 총 실행 시간도 증가하는 경향을 나타낸다. 예를 들어, 10,000,000 shots 에서 약 24.33 초가 소요되었다. 이는 병렬 처리 환경에서도 기본적인 연산량 증가는 시간 증가로 이어짐을 보여준다.

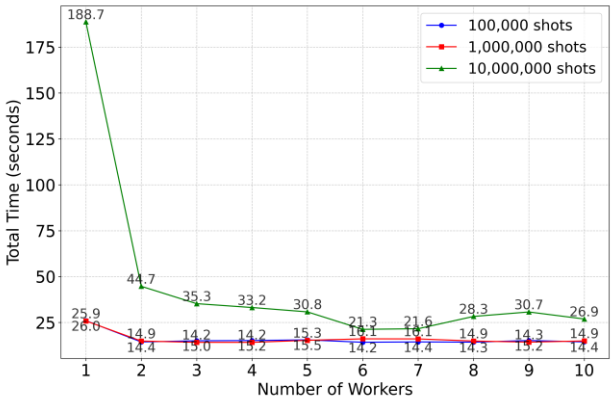

그림 4. QAOA 워커 수에 따른 총 실행 시간

그림 4 는 더 복잡한 QAOA depth 10 환경(20 개 하위 회로)에서 다양한 측정 횟수(10 만, 100 만, 1,000 만 shots)에 대해 RunPod 워커 수 증가에 따른 총 실행 시간 변화를 보여준다. 모든 측정 횟수 조건에서 워커 수가 증가함에 따라 총 실행 시간이 크게 단축되는 것을 확인할 수 있다.

특히 10,000,000 shots 의 경우, 워커 1 개를 사용했을 때 188.7 초가 소요되었으나, 워커 10 개를 사용했을 때는 26.9 초로 단축되어 약 7 배의 성능 향상을 보였다. 1,000,000 shots 에서는 워커 1 개일 때 25.9 초에서 워커 10 개일 때 14.9 초로, 100,000 shots 에서는 워커 1 개일 때 26.0 초에서 워커 10 개일 때 14.4 초로 단축되었다.

이는 분산 병렬 처리가 회로 절단 시뮬레이션의 효율성을 크게 향상시킬 수 있음을 명확히 보여준다. 다만, 워커 수가 일정 수준(예: 5-6 개 이상)으로 증가하면 성능 향상 폭이 다소 둔화되는 경향도 관찰되며, 이는 통신 오버헤드 또는 작업 분배의 한계 때문일 수 있다.

Ⅲ. 결론

본 연구는 QAOA 양자회로 절단 시뮬레이션에서 발생하는 오버헤드를 RunPod 서버리스 워커를 활용한 병렬 분산 처리를 통해 효과적으로 단축할 수 있음을 실험적으로 입증했다. 10-큐비트 QAOA 회로에 대해 depth 와 측정 횟수, 워커 수를 변경하며 실험한 결과, 특히 복잡한 회로와 많은 측정 횟수(최대 1,000 만 shots) 조건에서 워커 수를 늘림에 따라 전체 시뮬레이션 시간이 크게 감소함을 확인했다. 예를 들어, 1,000 만 shots, depth 10 조건에서 워커 10 개를 사용했을 때 워커 1 개 대비 약 7 배의 시간 단축 효과를 얻었다.

이러한 결과는 회로 절단 기법이 실제 양자 장치의 제약을 완화할 뿐만 아니라, 적절한 병렬 분산 처리 전략과 결합될 경우 고전 시뮬레이터 환경에서도 대규모 양자 알고리즘 탐색의 효율성을 높일 수 있음을 시사한다. 이는 분산 컴퓨팅을 활용한 양자 시뮬레이션 실행 시간 단축 연구[5][6]의 중요성을 뒷받침한다.

향후 연구로는 최적의 워커 수 및 자원 할당 전략, 다양한 양자 알고리즘에 대한 적용 가능성 확장, 그리고 실제 양자컴퓨터에서의 검증 등을 통해 본 연구의 실용성을 더욱 높일 수 있을 것이다.

참 고 문 헌

[1] J. Choi and J. Kim, "A tutorial on quantum approximate optimization algorithm (QAOA): Fundamentals and applications," in Proc. 2019 Int. Conf. Inf. Commun. Technol. Converg. (ICTC), Jeju, Korea (South), Oct. 2019, pp. 138-142. doi: 10.1109/ICTC46691.2019.8939749.

[2] A. Lowe et al., "Fast quantum circuit cutting with randomized measurements," Quantum, vol. 7, p. 934, Mar. 2023. doi: 10.22331/q-2023-03-02-934.

[3] M. Hart and J. McAllister, "Quantum Circuit Cutting Minimising Loss of Qubit Entanglement," in Proc. 21st ACM Int. Conf. Comput. Front. (CF '24), Ischia, Italy, May 2024, pp. 298-306. doi: 10.1145/3649153.3649189.

[4] M. Bechtold et al., "Investigating the effect of circuit cutting in QAOA for the MaxCut problem on NISQ devices," Quantum Sci. Technol., vol. 8, no. 4, Art. no. 045021, Oct. 2023. doi: 10.1088/2058-9565/acf59c.

[5] S. Kim, V. R. Pascuzzi, Z. Xu, T. Luo, E. Lee, and I.-S. Suh, "Distributed Quantum Approximate Optimization Algorithm on a Quantum-Centric Supercomputing Architecture," arXiv:2407.20212, Jul. 2024.

[6] D. Main, P. Drmota, D. P. Nadlinger et al., "Distributed quantum computing across an optical network link," Nature, vol. 638, pp. 383-388, Feb. 2025. (Published online Jun. 26, 2024). doi: 10.1038/s41586-024-08404-x.