

Remote-CNOT 논리 큐빗 구조에서 기울기 벡터에 따른 초기 배치 이득 분석

강유진, 허준*

고려대학교

yujin20@korea.ac.kr, *junheo@korea.ac.kr

Analysis of the benefits of initial placement based on gradient vectors in a remote-CNOT logical qubit architecture

Kang Yu Jin, Heo Jun*

Korea Univ.

요약

본 논문은 Remote-CNOT 을 사용하는 Row-type 논리 큐빗 구조에서 논리 연산 종류별 개수를 반영하여 초기 배치를 결정하고, 반영 비율을 결정하는 기울기 벡터에 따른 이득을 분석한다. 기존 기법에서는 직선의 기울기 벡터를 (1,1)만 활용하였으므로 본 연구는 (1,2), (2,1)을 추가로 사용하여 기울기에 따른 이득을 추가 분석한다. 임의로 생성한 양자 회로와 기존 벤치마크 회로 대해 시뮬레이션을 수행하여 시간 비용을 측정한 결과, (1,1)이 대부분의 경우에서는 이득을 보였으나, CNOT 의 비율이 높은 회로의 경우 (1,2)가 더 적은 시간 비용을 사용하는 것을 확인할 수 있었다.

I. 서론

논리 큐빗을 배치하는 방법인 논리 큐빗 구조에서 하나의 논리 큐빗을 폐치라고 할 때, 정보가 저장되는 논리 큐빗을 data 폐치, 논리 연산에서 소모되는 논리 큐빗을 ancilla 폐치라고 한다. 논리 큐빗의 초기 배치에 관여되는 논리 연산은 CNOT, S(\sqrt{Z} , $\pi/4$ gate), T($\pi/8$ gate) 연산으로 ancilla 폐치를 사용한다는 공통점이 있다[1-3]. Checkerboard, tile-based 논리 큐빗 구조의 경우, 멀리 떨어진 논리 큐빗 간 논리 CNOT 을 수행하기 위해 논리 SWAP 을 추가한다[4]. 그러나, row-type 구조에서는 remote-CNOT 을 통해 둘 사이에 있는 ancilla 폐치를 소모하므로 논리 SWAP 없이 수행할 수 있다. 따라서 Remote-CNOT 을 수행하는 data 폐치 사이에 있는 data 폐치들은 CNOT, S, T 연산을 수행할 수 없다. 따라서 논리 큐빗 초기 배치 방법에 동시 수행 가능한 연산이 결정되므로 전체 회로의 수행 시간 비용에 영향을 준다.

Row-type 구조를 처음 제안한 연구에서는 CNOT 의 개수만 고려하여 많은 순서대로 가운데에 배치하였다[5]. 이후, 각 논리 연산의 개수를 2 차원 평면에 표현하고 직선 위로 사영하여 초기 배치를 결정하는 방식을 제안하였으나 (1,1)의 기울기 벡터를 갖는 직선만 사용하였다[6]. 본 논문에서는 (1,2)와 (2,1)의 기울기 벡터를 갖는 직선 위로 사영하여 초기 배치를 결정하고 각 방식에서 측정된 시간 비용을 비교하였다. 이때 (1,2)와 (2,1)는 정규화 하여 사용하였으나 본 논문에서 편의상 정규화 전의 값으로 표기하였다.

II. 본론

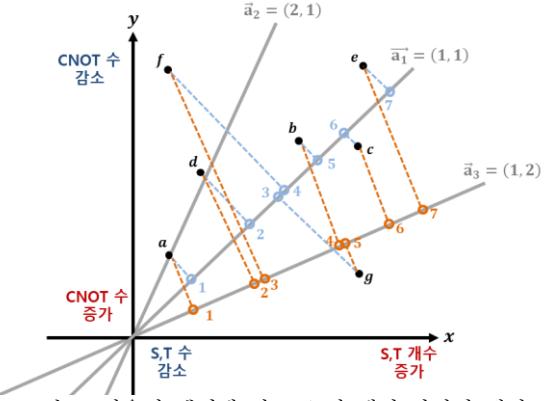


그림 1. 기울기 벡터에 따른 초기 배치 방법의 예시

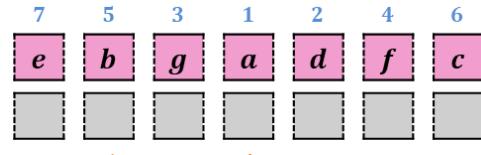


그림 2. 기울기 벡터가 (1,1)인 경우(위)와, (1,2)인 경우(아래)에 초기 배치 적용 예시

Circuit	논리 큐빗 수	논리 연산 수	R_{CNOT} [%]	$R_{S,T}$ [%]	시간 비용 [d cycle]				(0,1) 대비 Ratio [%]		
					(0,1) ^[5]	(1,1) ^[6]	(2,1)	(1,2)	(1,1)	(2,1)	(1,2)
Random1	10	60	16.67	66.67	26.4	26.4	26.2	25.4	0.0	-0.8	-3.8
Shor15	6	225	40.00	46.67	282.3	280.3	284.6	284.0	-0.7	0.8	0.6
3AQFT	4	36	38.89	38.89	48.3	48.0	48.1	48.0	-0.6	-0.4	-0.6
Random2	10	60	41.67	41.67	54.5	50.6	54.0	52.6	-7.2	-0.9	-3.6
Adder0-5	16	290	44.14	43.45	291.8	296.8	290.4	289.2	1.7	-0.5	-0.9
Adder1-8	18	273	47.25	41.03	370.2	365.7	373.4	373.4	-1.2	0.9	0.9
Y-dist(7)	8	23	52.17	30.43	24.5	24.4	24.1	23.8	-0.4	-1.6	-2.9
A-dist(15)	16	55	63.64	27.27	67.2	62.6	64.0	65.1	-6.8	-4.8	-3.1
A-dist(20)	25	89	69.66	22.47	109.7	103.4	111.0	108.2	-5.7	1.2	-1.4
Random3	7	25	72.00	16.00	39.5	39.8	38.4	39.0	0.7	-2.7	-1.3

표 1. 시뮬레이션을 수행한 회로의 특성 및 Projection 방법에 따른 시간 비용

CNOT 연산 수를 N_i , S 와 T 연산 수를 M_i 라 할 때, 그림 1 의 2 차원 평면에서 각 점은 $(x_i, y_i) = (1 - N_i / \sum_i N_i, M_i / \sum_i M_i)$ 로 정의되며, 기울기 벡터가 각각 $\vec{a}_1 = (1,1)$, $\vec{a}_2 = (2,1)$, $\vec{a}_3 = (1,2)$ 인 직선 위에 사영되어 원점에 가까운 순으로 Row-type 구조 가운데에 할당된다. 1 번에서 7 번까지의 순서는 동일한 점이더라도 직선의 기울기에 따라 다르게 결정된다.

그림 2에서 data 패치는 분홍색 박스이며 ancilla 패치는 회색 박스이다. 그림 1에서 정해진 순서에 따라 논리 큐빗 구조의 가운데부터 이와 가까운 지점 순으로 번갈아 할당된다. 제시된 예시에서 큐빗 b, f, g 는 기울기 벡터에 따라 다른 순서를 갖는다.

기준 연구에 사용된 벤치마크와 임의의 지표로 생성된 회로(Random 1,2,3)를 사용하여 시간 비용을 계산하는 시뮬레이션을 수행하였다. MATLAB 환경에서 window size가 30인 스케줄링 및 라우팅을 사용하였으며 동일한 회로에 대해 10회 실시하고 평균값을 사용하였다.

표 1은 회로의 특성과 시뮬레이션 결과를 제시한 것이다. CNOT 및 S, T 연산 수에서 논리 연산 수를 나눈 값은 R_{CNOT} , $R_{S,T}$ 이며, 더 높은 값은 과란색으로 표기하였다. 시간 비용의 단위는 d cycle이고, d 는 code distance 이므로 시간 비용은 d 의 배수로 증가한다.

(0,1)로 표기된 경우는 [5]의 방식을 나타낸 것으로 CNOT의 개수만 고려한 방법이며 (1,1)은 [6]의 방식이다. 시간 비용에서 가장 적은 값은 주황색 명암으로 표기하였다. (0,1) 대비 ratio는 (0,1) 일 때를 기준으로 기울기 벡터를 바꾼 방법마다 차이를 비율로 계산하였다. 기울기 벡터를 사용한 모든 방식이 [5]에 제시된 초기 방식보다 더 적은 시간 비용이 소요되었다. 최소 시간 비용은 (1,1)의 기울기 벡터를 사용한 경우에서 -7.2%까지 가장 많이 발견되었으나, (1,2)의 기울기 벡터를 사용할 때 더 많은 회로에서 최소 -0.6%에서 최대 -3.8%의 이득을 보였다.

III. 결론

Remote-CNOT 을 사용하는 Row-type 구조는 논리 CNOT, S, T 모두 ancilla 패치를 활용하기 때문에, 세 연산의 개수를 모두 고려한 초기 배치 방법이 요구된다. 기존에는 CNOT 만 고려하거나, 모두 고려하더라도 하나의 기울기 벡터만 활용하여 사영하였다. 본 연구는 다양한 기울기 벡터를 추가로 활용하여 시간 비용이 얼마나 감소하는지 확인하고 성능을 비교하였다.

주어진 모든 회로에서 사영을 사용하지 않는 (0,1) 기법 보다 (1,1), (1,2), (2,1)의 기울기 벡터를 사용한 기법의 이득을 확인할 수 있었다. (1,1) 방식이 가장 적은 비용을 가지는 회로는 6 개로 가장 많았으나 이득이 없는

4 개의 회로도 존재하였다. (1,2) 방식은 가장 적은 비용을 가지는 회로가 4 개였지만, 2 개의 회로를 제외한 모든 회로에서 시간 비용 이득을 보였다. 따라서 최대의 이득을 원하는지 또는 다수의 회로에서 이득을 원하는지의 기준에 따라 기울기 벡터를 선정할 수 있다.

추후에는 더 다양한 벤치마크 회로와 기울기 벡터로 분석하거나, 수행 회로의 CNOT, S, T 연산 비율에 따라 최적의 기울기 벡터를 유동적으로 찾아서 적용하는 연구를 고려할 수 있을 것이다[7].

ACKNOWLEDGMENT

본 연구 논문은 정부(과학기술정보통신부)의 지원으로 정보통신 기획평가원의 지원을 받아 수행된 연구임 (No. 2020-0-00014, 결합허용 논리양자큐빗 환경을 제공하는 양자운영체제 원천기술 개발). 이 논문은 2025년도 정부(과학기술정보통신부)의 지원으로 정보통신기획평가원의 지원을 받아 수행된 연구임 (No. 2022-0-00463, 양자인터넷 구현을 위한 유선 양자중계기 개발).

참 고 문 현

- [1] Kang, Yujin, et al. "Fault-tolerant quantum computation using low-cost joint measurements." *Quantum Information Processing* 23.5 (2024): 190.
- [2] Horsman, Dominic, et al. "Surface code quantum computing by lattice surgery." *New Journal of Physics* 14(12), 2012.
- [3] Litinski, Daniel. "Magic state distillation: Not as costly as you think." *Quantum* 3 (2019): 205.
- [4] Lao, Lingling, et al. "Mapping of lattice surgery-based quantum circuits on surface code architectures." *Quantum Science and Technology* 4.1 (2018): 015005.
- [5] Lee, Jonghyun, et al. "Lattice surgery-based Surface Code architecture using remote logical CNOT operation." *Quantum Information Processing* 21.6 (2022): 217.
- [6] 강유진, 허준. "Row-type 논리 큐빗 구조의 초기 배치 방법." *한국통신학회 학술대회논문집* (2025): 1596-1597.
- [7] Vanhatalo, Erik, Murat Kulahci, and Bjarne Bergquist. "On the structure of dynamic principal component analysis used in statistical process monitoring." *Chemometrics and intelligent laboratory systems* 167 (2017): 1-11.