Classical simulation of qubits on DaVinci-1 HPC in the MBQC frame

Sebastiano Corli
Polimi, CNR

Enrico Prati
Unimi, CNR

LEONARDO

In collaboration with:

C. Cavazzoni

F. Cerocchi

M. Dispenza

D. Dragoni

M. Proietti

Quantum Team in Milan (Italy)

CNR is the main public research body of Italy

8000 staff
@IFN: 80 staff

IBM Q
LEONARDO
VISTA TECHNOLOGY
ASI Agenzia Spaziale Italiana
NVIDIA
CINECA
aws
REPLY DATA
D-Wave

Quantum Team

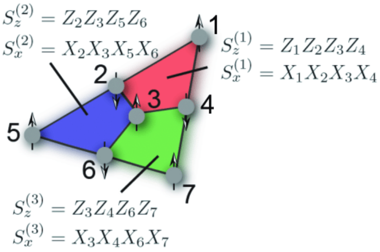
Head: Prof. Enrico Prati, PhD (Universita Statale Milano)

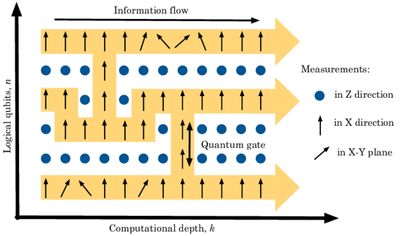
Lorenzo Moro (POLIMI)
Gabriele Agliardi (IBM POLIMI)
Sebastiano Corli (POLIMI)
Marco Maronese (UNIBO IIT)
Lorenzo Rocutto (UNIBO IIT)
Iris Paparelle (UNIMI)
Andrea Zanetti (POLIMI)
Davide Noè (UNIMIB)
Riccardo Molteni (UNIMIB)
Paolo Zentilini (UNIMI)
Andrea Rossoni (UNIMI)

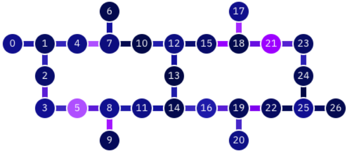
Graph Theory in Quantum Computing

Fields of interest

1. Error correction codes
2. Qubit connectivity
3. One-Way Quantum Computing

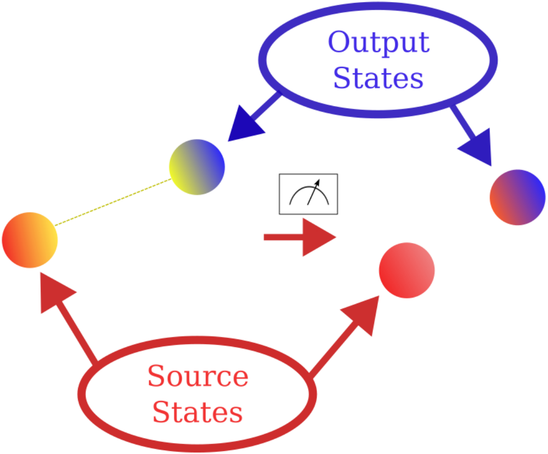






A. Bermudez et al. "Assessing the progress of trapped-ion processors towards fault-tolerant quantum computation." Physical Review X 7.4 (2017): 041061.

Measurement Based Quantum Computing (MBQC)



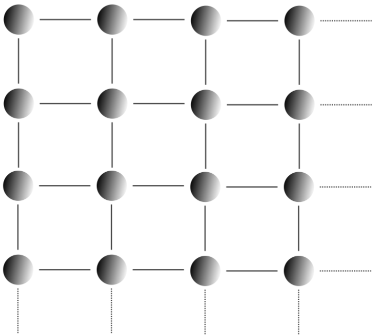
MBQC States

1. Source states
   - inputs
   - ancillae
2. Output states

MBQC Operators

- Entanglement
- Measurements
- Corrections

Graph States



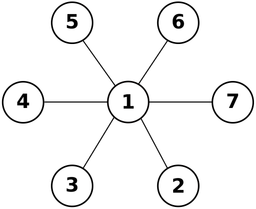
Graph

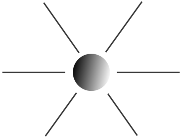
- Vertices:
  $V = \{1, 2, ..., n\}$
- Edges:
  $E = \{(1, 2), (2, 3), ..., (n-1, n)\}$
- Neighborhood of a:
  $N_a = \{b \in V | (a, b) \in E\}$

Cluster state

- Vertices: qubits
- Edges: entanglements

Stabilizers





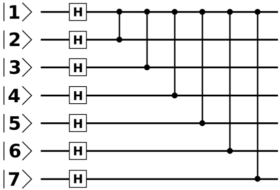
Graph Operators

- $\hat{X}$

| $\hat{Z}$

Stabilizer group

- $\mathcal{S}_n = \{\hat{K}_i, \hat{K}_j \in \mathcal{P}_n \,|\, [\hat{K}_i; \hat{K}_j] = 0\}$
- $\hat{K}_i \,|\psi\rangle = |\psi\rangle \;\forall i$



$$\begin{cases} \hat{K}_1 = \hat{X}_1 \hat{Z}_2 \hat{Z}_3 \hat{Z}_4 \hat{Z}_5 \hat{Z}_6 \hat{Z}_7 \\ \hat{K}_2 = \hat{Z}_1 \hat{X}_2 \end{cases}$$

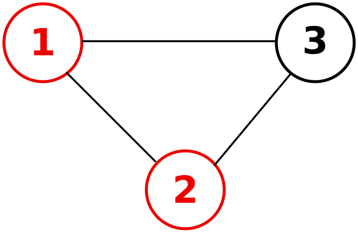
Hub code

Hubs set $\mathcal{B}$

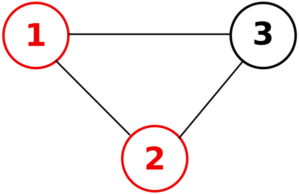
- Def: vertices covering all the edges $\cup_{a\in\mathcal{B}} N_a \supseteq E$

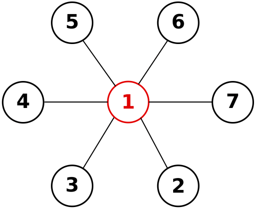
Algorithm Pseudo-code

$G = [V, E]$, $\mathcal{B}$
for $i \in V$ **do**
  if $i \in \mathcal{B}$ **then**
    $|i\rangle = |0\rangle$
  else
    $|i\rangle = |+\rangle$
for $i \in$ hubs **do**
  $|\Psi\rangle = \frac{1}{\sqrt{2}}[\hat{\mathbb{I}} + \hat{K}_i] |\Psi\rangle$



- $|\Psi\rangle = |00+\rangle$
- $\hat{G}_1 |\Psi\rangle = \frac{|00+\rangle + |10-\rangle}{\sqrt{2}}$
- $\hat{G}_2 |\Psi'\rangle = \frac{|00+\rangle + |10-\rangle + |01-\rangle - |11+\rangle}{2}$





Ring topology

- Number of hubs: $\lceil \frac{n}{2} \rceil$
- hub stabilizers: $\langle \hat{X}_1 \hat{Z}_2 \hat{Z}_3; \hat{Z}_1 \hat{X}_2 \hat{Z}_3 \rangle$
- state: $\frac{|00+\rangle + |10-\rangle + |01-\rangle - |11+\rangle}{2}$

Star topology

- Number of hubs: 1
- hub stabilizers: $\langle \hat{X}_1 \hat{Z}_2 \hat{Z}_3 \hat{Z}_4 \hat{Z}_5 \hat{Z}_6 \hat{Z}_7 \rangle$
- state: $\frac{|0{+}{+}{+}{+}{+}{+}\rangle + |1{-}{-}{-}{-}{-}{-}\rangle}{\sqrt{2}}$

Benchmarking Codes

Hub formalism

- Operations:
  - Preparing state: single qubit ($\hat{H}$ operator)
  - Entanglements: single qubit ($\hat{X}$, $\hat{Z}$ operators)
- Group generators: $\mathcal{S}'_n \subset \mathcal{S}_n$
- Scaling of operations:
  - Preparing state: $|\mathcal{B}|$ (number of hubs, $< n$)
  - Entanglements: $|\mathcal{B}| \cdot \sum_{i \in \mathcal{B}}(N_i + 1)$ (N_i number of neighboring vertices)

CZ formalism

- Operations:
  - Preparing state: single qubit ($\hat{H}$ operator)
  - Entanglements: two-qubits ($\hat{CZ}$ operator)
- Group generators: $\mathcal{S}_n$
- Scaling of operations:
  - Preparing state: n (number of qubits)
  - Entanglements: m (number of edges)

Classical simulation in MBQC frame

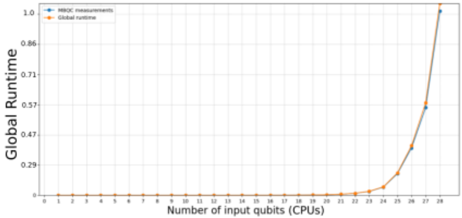
Implementation

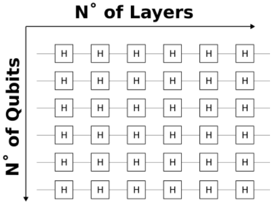
- Software: Paddle Quantum
- Hardware: DaVinci-1 HPC
- GPUs (high performances, low memory)
- CPUs (high memory, low performances)

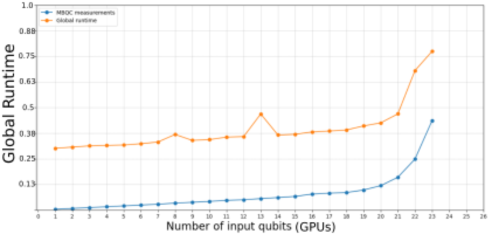
Our work

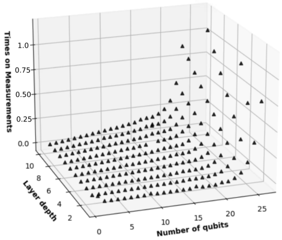
1. Stress Test: performances of Paddle Quantum on DaVinci-1
2.
3.

Stress Test: N-dimensional Hadamard Circuit









Classical simulation in MBQC frame

Implementation

- Software: Paddle Quantum
- Hardware: DaVinci-1 HPC
- GPUs (high performances, low memory)
- CPUs (high memory, low performances)

Our work

1. Stress Test: performances of Paddle Quantum on DaVinci-1
2. Wrapper for Paddle Quantum
3.

Wrapper for Paddle Quantum

Cons of Paddle Quantum

- Lacking API
- Tricky calls to functions

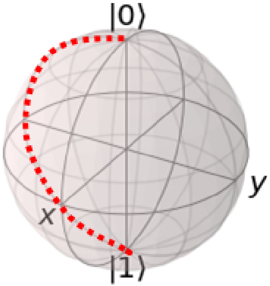
Wrapper

1. Better knowledge about the libraries
2. Better access to modules

```
##################################
## CLASS TEMPLATE FOR MCALCULUS ##
##################################
class Mbqc(Circuit):
    def __init__(self, angles, stabilizers, inputs, corrections, depth=1):
        super().__init__(len(inputs), depth)
        self.inputs = copy.copy(inputs)
        ## tuple are immutable type ##
        ## this turn to be useful   ##
        ## when defining __matmul__ ##
        self.save = tuple(inputs)
        self.angles = angles
        self.corrections = corrections
        ## Set the underlying graph ##
        self.mbqc = MBQC()
        self.mbqc.set_graph(stabilizers)
        ## Set the time intervals ##
        ## for entanglements
        self.dt_entangl = 0
        ## for measurements
        self.dt_meas = 0
        ## for corrections
        self.dt_corr = 0
```

Practical application

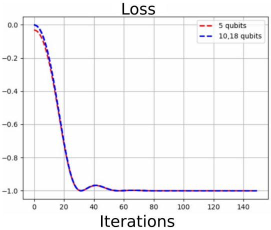
- Backpropagation for optimization
- Circuit: $[\hat{R}_x(\theta)\,|0\rangle]^{\otimes N}$
- Hyperparameter: θ

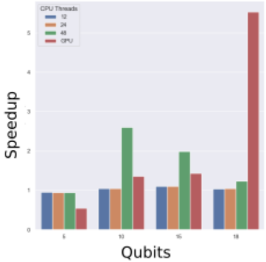


Numerical Results

Algorithm Pseudo-code

Set GPU/CPU threads
Set N
Set ITR
$|\Psi_0\rangle = |0\rangle^{\otimes N}$
$\hat{M} = \text{Diag}(-1, ..., -2^N)$
for $i \in ITR$ **do**
  $|\Psi_i(\theta)\rangle = \hat{R}_x(\theta)\,|\Psi_{i-1}\rangle$
  $Loss = -\langle\Psi_i(\theta)|\,\hat{M}\,|\Psi_i(\theta)\rangle$
  Backpropagation $\rightarrow \theta$

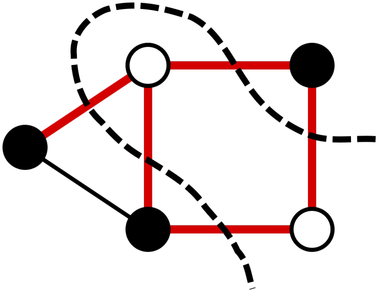




Highlights and Conclusions

Practical application

- New formalism in MBQC frame
- Numerical tests of Paddle Quantum libraries on DaVinci-1
- Wrapper development
- Backpropagation on Paddle Quantum libraries on DaVinci-1
- Future project: QAOA in MBQC frame for MaxCut problem



Pauli Matrices

$$\hat{X} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

$$\hat{Y} = i \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}$$

$$\hat{Z} = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

$$\hat{\sigma}_i \hat{\sigma}_j = \hat{\mathbb{I}} \delta_{ij} + i \epsilon_{ijk} \hat{\sigma}_k$$

Hadamard Gate

$$\hat{H} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

CZ gate

$$\hat{CZ} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix}$$