M3: Scalable measurement mitigation demo

Hwajung Kang, Ph.D.
Education Delivery Program Manager, IBM Quantum

IBM **Quantum**

Problem

- Measurement errors currently important source of errors for quantum computing systems

System	Avg. assignment error (%)
Alibaba 11Q [31]	7.4
Google 53Q Sycamore [32]	3.2
IBM 27Q Falcon_R5.11 [33]	1.1
IONQ 11Q [34]	0.4
Quantum Inspire 5Q Starmon [35]	4.0
Rigetti 32Q Aspen-9 [34]	6.1

TABLE I. Representative error rates for cloud-accessible quantum computing systems.

Measurements are good, but not perfect.

- All the measurement mitigation methods that had been used are limited to few qubits due to operating on the full probability space that **grows exponentially with the number of qubits.**

- M3 (Matrix-free Measurement Mitigation) method is not restricted by number of qubits.

Background

- Measurement error mitigation is well described by a simple linear system:

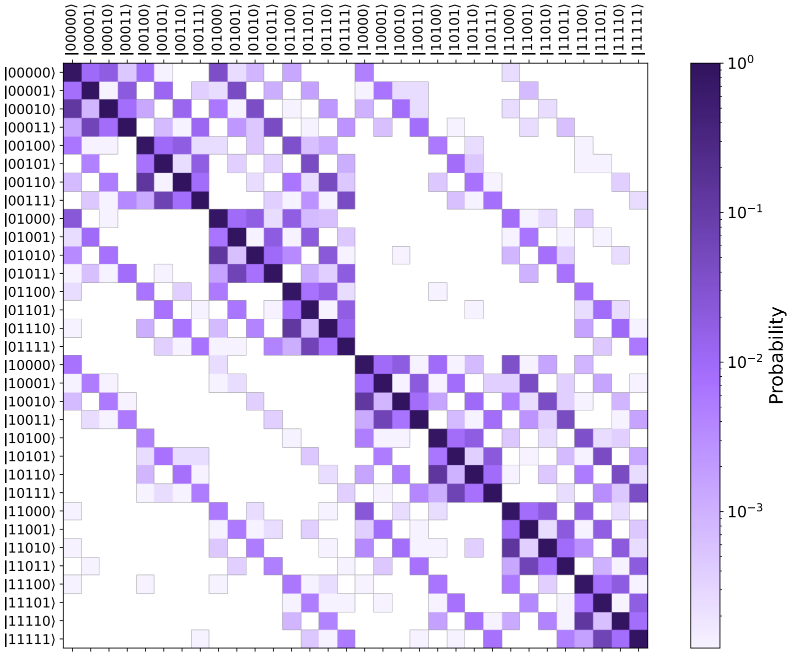
$$\vec{p}_{\text{noisy}} = A\vec{p}_{\text{ideal}},$$

what you got   what you wanted

- A-matrix ($2^n * 2^n$, n : number of qubits) probabilistically maps a given ideal measurement outcome (bit-string) to one that is observed in the noisy output (columns sum to one).

- The dimensionality of the A-matrix grows exponentially with the number of measured qubits.

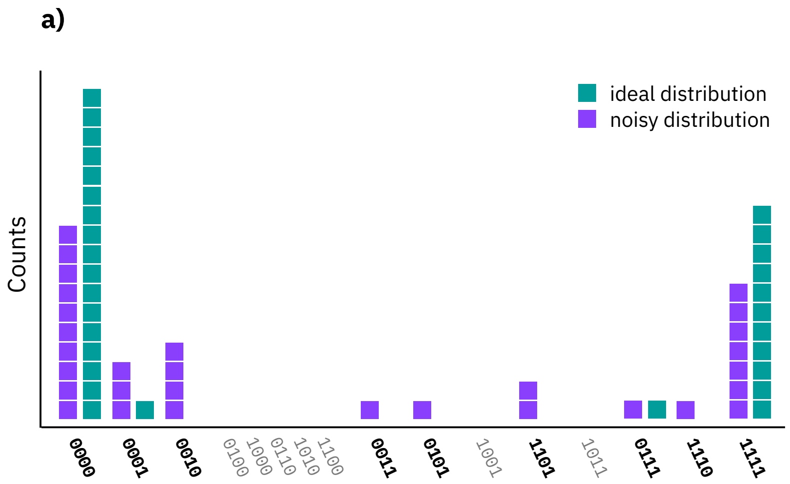
- Note that A has nice properties.

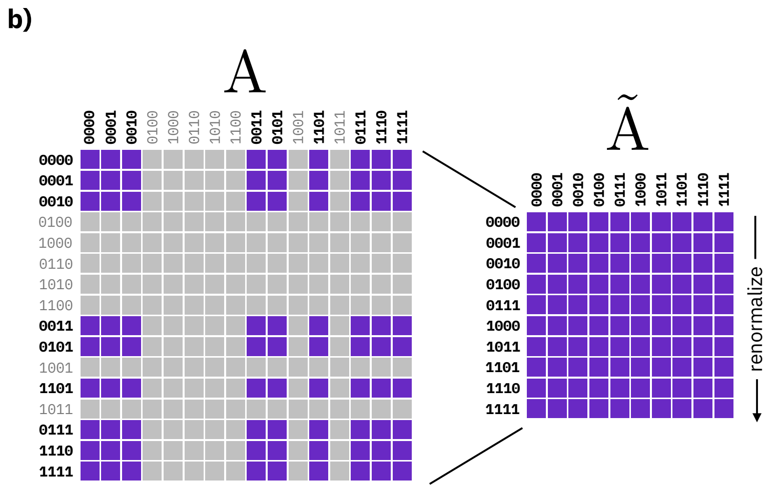


Assignment (A-matrix) for Kolkata 0 -> 4

Approach

- Can look at measurement mitigation in a sub-space of the full exponential space defined by only those bit-strings observed in the noisy output.





Approach

b: noisy data $\quad$ x: ideal data

A : stochastic mtx for q_0 $\quad$ B : stochastic mtx for q_1

$$\vec{p}_{\text{noisy}} = A\vec{p}_{\text{ideal}},$$

$$\begin{aligned}
b_{00} &= B_{00}A_{00}x_{00} + B_{00}A_{01}x_{01} + B_{01}A_{00}x_{10} + B_{01}A_{01}x_{11} \\
b_{01} &= B_{00}A_{01}x_{00} + B_{00}A_{11}x_{01} + B_{01}A_{10}x_{10} + B_{01}A_{11}x_{11} \\
b_{10} &= B_{10}A_{00}x_{00} + B_{10}A_{01}x_{01} + B_{11}A_{00}x_{10} + B_{11}A_{01}x_{11} \\
b_{11} &= B_{10}A_{10}x_{00} + B_{10}A_{11}x_{01} + B_{11}A_{10}x_{10} + B_{11}A_{11}x_{11}
\end{aligned}$$

col_norm_00 $\quad$ col_norm_10 $\quad$ col_norm_11

Approach

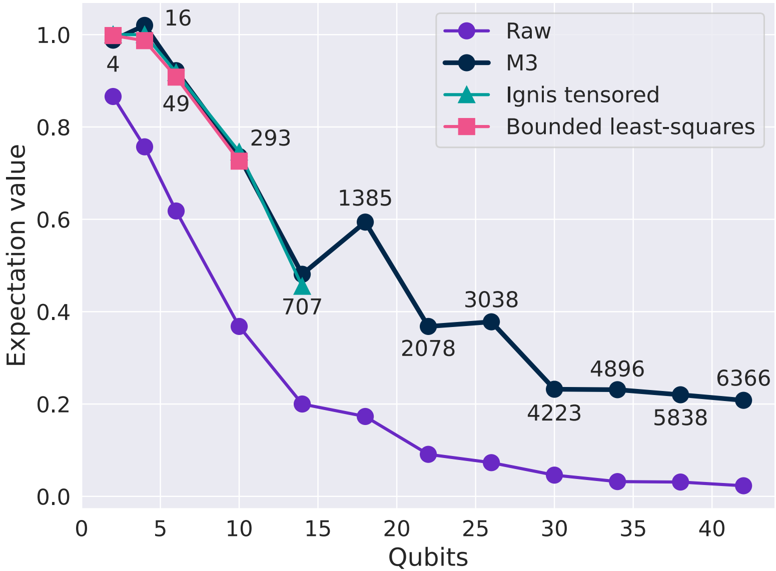
- It is possible to iteratively solve the linear system using solvers that use only matrix-vector products, but do not construct the matrix.

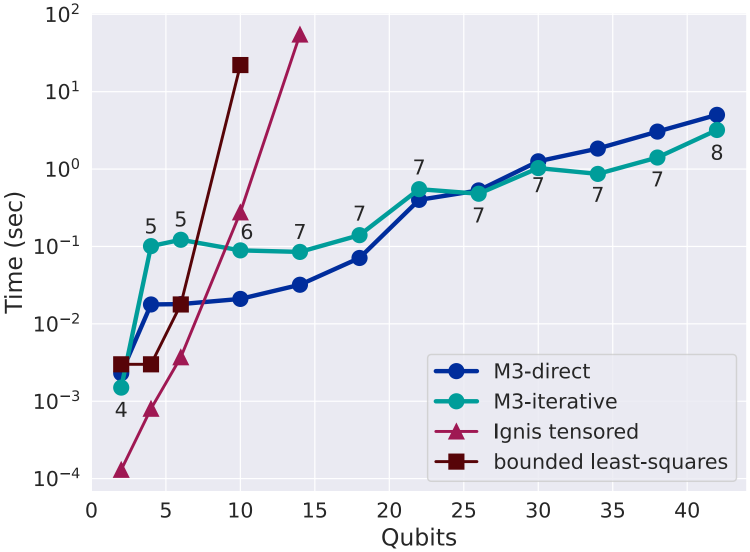
    - Examples: **GMRES**, BICGSTAB, Jacobi, etc...

- Usually much slower than direct decomposition, but rapid convergence in just a few iterations (<10), performance competitive to LU or better, is possible exploiting A-matrix diagonal dominance.

- Uses orders of magnitude less memory.

Example (42-qubit GHZ mitigation, Manhattan)





- at 42Q solution, M3-iterative takes 3sec and uses ~ 1MiB of memory

- Storing sparse full-dim A-matrix (Hamming distance 3) requires ~ 580 PiB, 120x more than Fugaku supercomputer memory

References

Paper : https://scirate.com/arxiv/2108.12518 **(accepted to PRX Quantum)**

Source Code : https://github.ibm.com/IBM-Q-Software/mthree

Thank you!