Rogues Gallery VIP - Quantum Computing - Spring 2025

Noah Dillard, Bishop Glover, Stephen Katz, Melody Lee, Krish Prasad, Tanush Prathi, Erica Wang, Francis Wu

Key Research Questions

- How can we develop a circuit generator using machine learning (ML) that minimizes noise?
- How can we leverage simulations data for this purpose?
- What are the limits of quantum simulations?
- How may we explore noise mitigation and error correction techniques?

Semester Goals

Following up with simulation work, the team seeks to continue delving deeper and achieve outcomes in four key areas. Namely,

1. **Data pipeline:** method to generate randomized circuits.
2. **Data collection and analysis:** Further investigation of the types of data collected.
3. **Quantum simulations:** Create more quantum simulation algorithms
4. **Benchmarking:** Identify bottlenecks in quantum simulations

Limitations in Quantum Computing

Quantum computing bottlenecks in [1]:

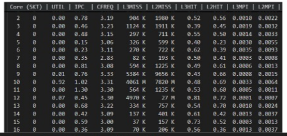
1. Read-in and read-out cost
2. ***Resource constraints and error ← subteam focus***

Motivations

- Understanding GPU and CPU performance on simulations gives upper bound for classical performance.
- Understanding error types and mitigation techniques increases coherence of information obtained from quantum computer.
- Underlies goal to make realization of quantum advantage possible.

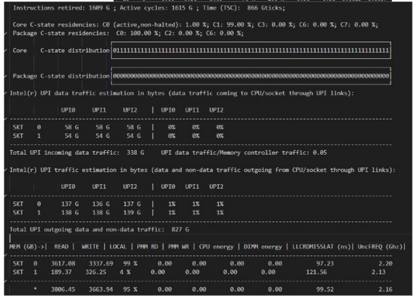
Benchmarking/Simulations

- Finalized implementations of Qiskit benchmarking algorithms
- Created scripts for scaling problem size
- Refactored algorithm implementations for ease of use



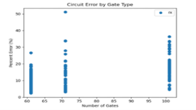
Core	(SKT)	UTIL	IPC	CFREQ	L3MISS	L2MISS	L3HIT	L2HIT	L3MPI	L2MPI
2	0	0.00	0.78	3.19	904 K	1980 K	0.52	0.56	0.0010	0.0022
3	0	0.00	0.46	3.23	1124 K	1911 K	0.39	0.45	0.0019	0.0032
4	0	0.00	0.48	3.15	297 K	711 K	0.55	0.50	0.0014	0.0033
5	0	0.00	0.15	3.06	326 K	599 K	0.40	0.23	0.0030	0.0055
6	0	0.00	0.23	3.11	270 K	722 K	0.62	0.39	0.0035	0.0093
7	0	0.00	0.35	2.83	82 K	193 K	0.50	0.41	0.0003	0.0008
8	0	0.00	0.81	3.08	594 K	1225 K	0.49	0.61	0.0006	0.0013
9	0	0.01	0.76	3.33	5384 K	9656 K	0.43	0.66	0.0008	0.0015
10	0	0.92	1.02	3.31	4061 M	7820 M	0.48	0.69	0.0033	0.0064
11	0	0.00	1.30	3.30	564 K	1235 K	0.53	0.60	0.0005	0.0011
12	0	0.07	0.45	3.30	4970 K	27 M	0.81	0.72	0.0001	0.0007
13	0	0.00	0.68	3.22	334 K	757 K	0.54	0.70	0.0010	0.0024
14	0	0.00	0.42	3.09	137 K	401 K	0.61	0.42	0.0013	0.0037
15	0	0.00	0.59	3.00	37 K	157 K	0.73	0.52	0.0003	0.0013
16	0	0.00	0.36	3.09	70 K	206 K	0.56	0.36	0.0013	0.0037

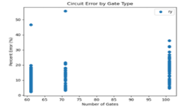
- Completed environment set-up for usage measurement libraries
- DCGMI and NVIDIA-SMI support
- Set up Intel PCM resource monitoring
- Collected initial CPU utilization measurements for Bernstein-Vazirani and Simon's Algorithms

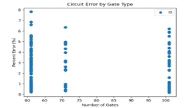


```
Instructions retired: 1609 G ; Active cycles: 1615 G ; Time (TSC):  866 Gticks;

Core C-state residencies: C0 (active,non-halted): 1.00 %; C1: 99.00 %; C3: 0.00 %; C6: 0.00 %; C7: 0.00 %;
Package C-state residencies:  C0: 100.00 %; C2: 0.00 %; C6: 0.00 %;

Core    C-state distribution 0111111111111111111111111111111111111111111111111111111111111111111111111111

Package C-state distribution 0000000000000000000000000000000000000000000000000000000000000000000000000000

Intel(r) UPI data traffic estimation in bytes (data traffic coming to CPU/socket through UPI links):

              UPI0     UPI1     UPI2    |  UPI0   UPI1   UPI2
--------------------------------------------------------------------------------
SKT    0       58 G     58 G     58 G   |    0%     0%     0%
SKT    1       54 G     54 G     54 G   |    0%     0%     0%
--------------------------------------------------------------------------------
Total UPI incoming data traffic:  338 G     UPI data traffic/Memory controller traffic: 0.05

Intel(r) UPI traffic estimation in bytes (data and non-data traffic outgoing from CPU/socket through UPI links):

              UPI0     UPI1     UPI2    |  UPI0   UPI1   UPI2
--------------------------------------------------------------------------------
SKT    0      137 G    136 G    137 G   |    1%     1%     1%
SKT    1      138 G    138 G    139 G   |    1%     1%     1%
--------------------------------------------------------------------------------
Total UPI outgoing data and non-data traffic:  827 G

MEM (GB)->|  READ |  WRITE | LOCAL | PMM RD | PMM WR | CPU energy | DIMM energy | LLCRDMISSLAT (ns)| UncFREQ (Ghz)|
--------------------------------------------------------------------------------
SKT   0    3617.08    3337.69   99 %      0.00      0.00      0.00      0.00        97.23          2.20
SKT   1    189.37    326.25    4 %      0.00      0.00      0.00      0.00       121.56          2.13
--------------------------------------------------------------------------------
      *    3806.45    3663.94   95 %      0.00      0.00      0.00      0.00        99.52          2.16
```

Exploring Error Mitigation with Machine Learning







Updates to Data Pipeline

- **Parquet unpacking script**: parquet file → (unpacking) circuit results from IBM hardware
- **Random circuit generator**: test distribution of noise over random circuits

Exploratory Project:

Objective: Given $|\psi\rangle$, devise U such that

- Ideal circuit: $U|\psi\rangle = b$
- Noisy circuit: $U^*|\psi\rangle = b^*$

We select $U = X, Y, Z$ such that $U^{2k} = I$. We tested the error rates for 2k = 16, ..., 100 to see if it would be possible to calculate a matrix M so that $MU^{2k*}|\psi\rangle = Ib$.

Conclusion: Not viable for hardware error with randomness.

Response: Pivot to slightly different project focused on error for future semesters.

Challenges

- Directory permission issues
- Heterogeneous hardware, OS and software compatibility
- Resolving CUDA and cuQuantum environment issues for simulations
  - Unable to perform simulations using GPU
  - Scattered work environments
- Limited number of qubits with simulation of Simon's Algorithm due to memory limitations
- Variation in error in data obtained limited viability

Future Work

- ***New project directions***
  - Exploration of IBM error models
  - **Add new research question:** impact of different types of error on circuit performance?
  - Example error models on Qiskit: coherent unitary error, phase damping error models
- ***Leveraging resources***
  - Q: Access to QCUPS?
  - Explore different hardware types
  - Investigate error rates on different hardware
- Using a myriad of different benchmark algorithms

References

[1] Aaronson, S. (2015). Read the fine print. *Nature Physics*, *11*(4), 291-293.