Quarry

Providing fast circuit fidelity estimation.

Hayden Coffey, Kanghee Park, Saurabh Kulkarni

Motivation

- Hardware noise impacts circuit performance.
  - Noisy backend -> poor fidelity.
  - Each platform has its own pitfalls.
  - **How do we know which one is best for our circuit?**
    - Trying each one is impractical (long queue times, etc.).
    - More platforms -> larger search space.
    - Simulating platform specific noise is expensive/slow.

- Goal: Provide a tool to help researchers make this decision.
  - A rough estimate of platform performance gives researchers a starting point.

Existing Work

- Estimating Fidelity:
    - Simulation [2.1, 2.2]
    - Variational Hybrid Quantum-Classical Algorithms (VHQCA) [4.2]
    - Execution and measurement of select qubits. [5.3]
    - Estimated Success Probability (ESP) [5.6]
    - Machine Learning Methods [5.7]
- Previous approaches are either too slow, or do not take into account platform specific attributes.
- Our goal is a tool that can provide real time updates to estimates as researchers design a circuit.

Quarry

- A tool to assist with quantum circuit design.
- Strategy:
  - Query the search space of available platforms/configurations with a circuit.
  - Provide researchers with an ordered list of recommended compatible machines.

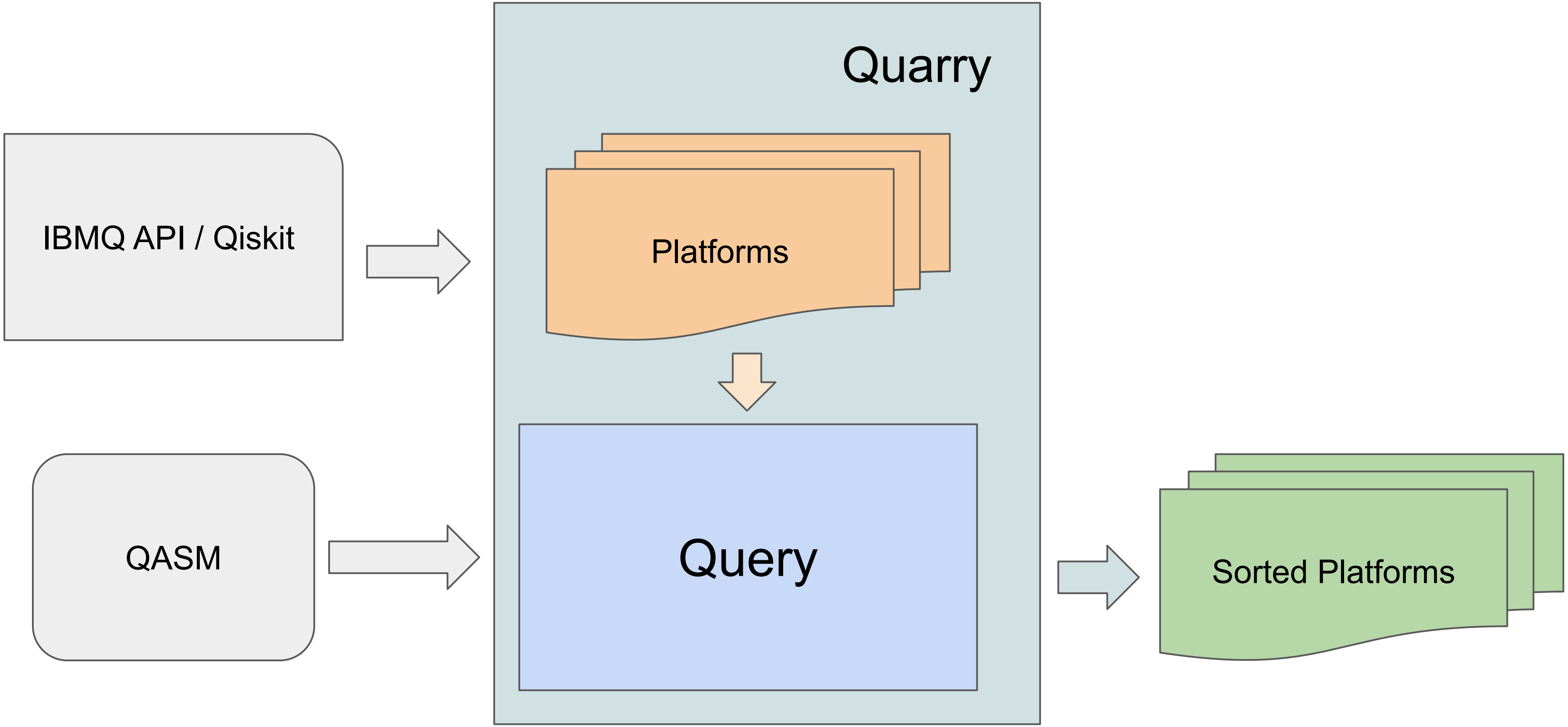
- Challenges:
  - What is a meaningful order? (What metrics?)
  - Recommendations can be rough estimates, but still need to be valid.
  - Needs to be fast. (Low latency)
  - Answers should be presented in a meaningful way. (Reach Goal: VSCode Plugin)

Quarry: Output Metrics

- Given a quantum circuit and a platform, provide a subset of the following:
  - **Fidelity**: PST, TVD, ESP, L2 Distance, Hellinger distance, and Entropy.
  - **Mapping Cost**: Inserted swap count.

- Challenges:
  - Computational cost cannot explode.
  - Metrics need to be distinct / meaningful.
  - Qubits drift, calculated values are subject to change.
  - Provided machine characterization data can be faulty.

Quarry: Design



Quarry: Design

```
else:
    n = 10
    backends = QUtil.getFakeBackends(qc, n)

#Simulation
query(qc, backends, evalCircuitSim, printResults)

#ESP Estimate
query(qc, backends, evalCircuitESP, printResultsESP)

#ML Models
PredictorV1.load_models()
query(qc, backends, evalCircuitPredictV1, printResults)

PredictorV2.load_models()
query(qc, backends, evalCircuitPredictV2, printResults)
```

Estimating via Simulation

- Naive approach.
- Simulate a circuit on all compatible backends.
- Calculate “fitness” of platforms and provide recommendation.
- Fitness ~= PST / (TVD + Entropy + Swaps + L2 + Hellinger)
  - Only ordering WRT other machines matters.
  - Formula could use improvement.

qasm/SupermarQ/qaoa_fermionic_10.qasm 53.742684(s) ++++++++++++++

Backend Name	PST	TVD	Entropy	Swaps	L2	Hellinger	Fitness
fake_rueschlikon	0.812	0.357	5.557	0	0.044	0.659	0.503
fake_melbourne	0.431	0.638	6.328	25	0.068	0.992	0.089
fake_johannesburg	0.440	0.651	6.305	31	0.076	1.002	0.081
fake_poughkeepsie	0.391	0.687	6.328	31	0.074	1.048	0.070
fake_montreal	0.460	0.645	6.247	85	0.072	0.988	0.042
fake_guadalupe	0.477	0.631	6.250	96	0.067	0.962	0.040
fake_boeblingen	0.407	0.682	6.326	78	0.078	1.037	0.040
fake_tokyo	0.418	0.673	6.286	82	0.070	1.046	0.039
fake_singapore	0.411	0.666	6.348	128	0.070	1.030	0.027
fake_almaden	0.366	0.706	6.353	119	0.080	1.085	0.025

Estimating via ESP

- Exchange detail for speed.
- Estimate probability of success from noise model.
- Does not require simulation.

$$ESP = \prod_{i=1}^{N_{gates}} g_i^s * \prod_{j=0}^{N_{meas}} m_i^s$$

$$g_i^s = (1 - g_i^e) \quad m_i^s = (1 - m_i^e)$$

```
qasm/SupermarQ/qaoa_fermionic_10.qasm 5.922470(s)
Backend Name        ESP
fake_rueschlikon    1.000
fake_guadalupe      0.211
fake_melbourne      0.040
fake_montreal       0.018
fake_poughkeepsie   0.014
fake_boeblingen     0.002
fake_singapore      0.001
fake_almaden        0.000
fake_johannesburg   0.000
fake_tokyo          0.000
```

https://dl.acm.org/doi/abs/10.1145/3352460.3358257

Predictive Model V1

- Neural Network: Aim to estimate metrics obtained from simulation
- Design and train a model for each target metric.
- Does not require compilation of circuit beyond unrolling of gates to platform basis gates.
- Inputs: Gate counts, machine name/size, topology average node degree.

qasm/SupermarQ/qaoa_fermionic_10.qasm 0.923696(s) ++++++++++++++

Backend Name	PST	TVD	Entropy	Swaps	L2	Hellinger	Fitness
fake_melbourne	0.917	0.160	0.917	0	N/A	N/A	0.407
fake_rueschlikon	0.917	0.160	0.917	0	N/A	N/A	0.407
fake_poughkeepsie	0.917	0.160	0.917	0	N/A	N/A	0.407
fake_tokyo	0.917	0.160	0.917	0	N/A	N/A	0.407
fake_montreal	0.107	0.430	1.416	1	N/A	N/A	0.039
fake_guadalupe	0.107	0.430	1.452	1	N/A	N/A	0.039
fake_almaden	0.064	0.409	1.386	0	N/A	N/A	0.025
fake_boeblingen	0.064	0.409	1.396	0	N/A	N/A	0.025
fake_johannesburg	0.057	0.409	1.244	0	N/A	N/A	0.022
fake_singapore	0.057	0.411	0.991	6	N/A	N/A	0.018

Predictive Model V2

- Inputs can be improved.
- New parameters:
    - **Fidelity**: Average success rate per gate type / measurement.
    - **Topology**: Graph density. Average connectivity, clustering, and shortest path.
    - **Circuit**: *QASMBench* (Pacific Northwest National Laboratory)
        - Circuit width/depth, retention lifespan, gate/measurement density, size factor, entanglement variance.

qasm/SupermarQ/qaoa_fermionic_10.qasm 6.609076(s) ++++++++++++++

Backend Name	PST	TVD	Entropy	Swaps	L2	Hellinger	Fitness
fake_guadalupe	0.987	0.228	3.962	0	0.038	0.000	0.376
fake_montreal	0.992	0.239	4.139	0	0.039	0.000	0.374
fake_singapore	0.837	0.597	3.390	0	0.067	0.000	0.285
fake_melbourne	0.988	0.734	1.837	16	0.364	0.000	0.215
fake_poughkeepsie	0.988	0.734	2.078	16	0.417	0.000	0.214
fake_tokyo	0.988	0.734	2.306	16	0.428	0.000	0.213
fake_boeblingen	0.588	0.470	3.657	0	0.113	0.000	0.207
fake_rueschlikon	0.931	0.712	1.842	16	0.364	0.000	0.204
fake_almaden	0.526	0.482	3.736	0	0.112	0.000	0.184
fake_johannesburg	0.511	0.744	4.043	0	0.108	0.000	0.162

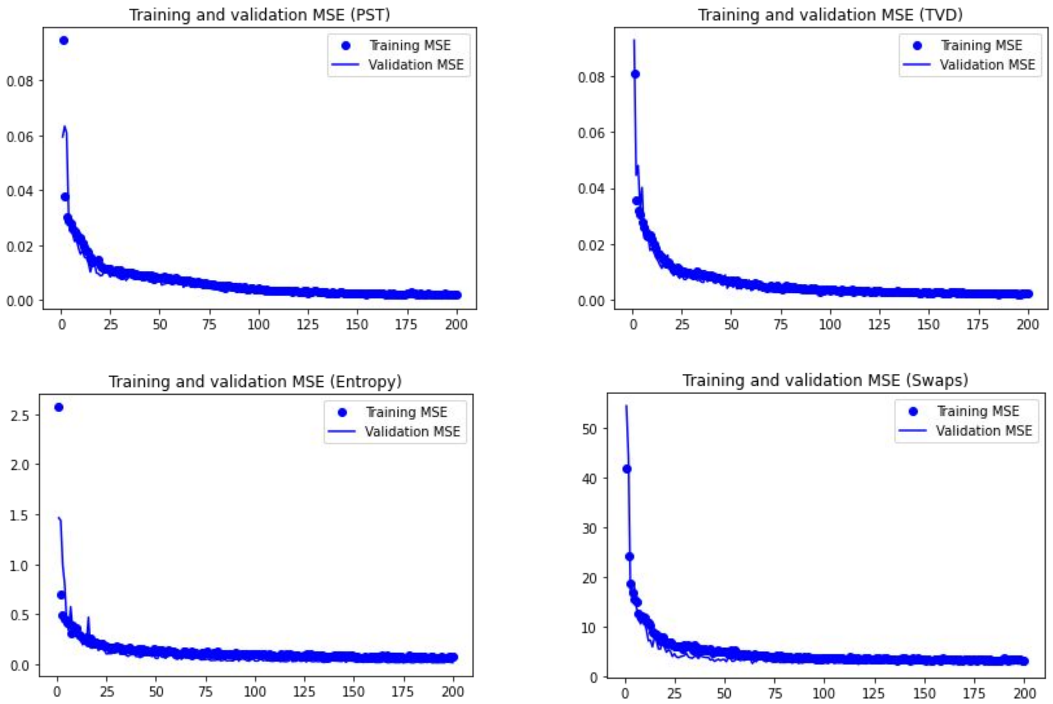
https://github.com/pnnl/QASMBench

Evaluation

- Per circuit performance evaluation in progress.
- Scaling in time and memory consumption:

Circuit Name	Method	Wall Clock Time (s)	User Time (s)	System Time (s)	Peak RSS (KB)
qasm/SupermarQ/qaoa_fermionic_10.qasm	Simulation	18.025355	113.93	5.98	927344
qasm/SupermarQ/qaoa_fermionic_10.qasm	ESP	8.300045	13	2.5	717196
qasm/SupermarQ/qaoa_fermionic_10.qasm	PredV1	0.925668	5.44	2.37	1623340
qasm/SupermarQ/qaoa_fermionic_10.qasm	PredV2	6.169059	11.3	2.51	1722360
qasm/QASMBench/medium/qf21_n15/qf21_n15.qasm	Simulation	790.637025	14758.46	9.43	1410568
qasm/QASMBench/medium/qf21_n15/qf21_n15.qasm	ESP	9.192249	14.17	2.61	717368
qasm/QASMBench/medium/qf21_n15/qf21_n15.qasm	PredV1	1.080091	5.49	2.45	1619380
qasm/QASMBench/medium/qf21_n15/qf21_n15.qasm	PredV2	7.045953	12.39	2.57	1721648

Results (V1)



Training and validation MSE (PST)
Training MSE
Validation MSE
0.08
0.06
0.04
0.02
0.00
0
25
50
75
100
125
150
175
200
Training and validation MSE (TVD)
Training MSE
Validation MSE
0.08
0.06
0.04
0.02
0.00
0
25
50
75
100
125
150
175
200
Training and validation MSE (Entropy)
Training MSE
Validation MSE
2.5
2.0
1.5
1.0
0.5
0.0
0
25
50
75
100
125
150
175
200
Training and validation MSE (Swaps)
Training MSE
Validation MSE
50
40
30
20
10
0
0
25
50
75
100
125
150
175
200

Results (V1)

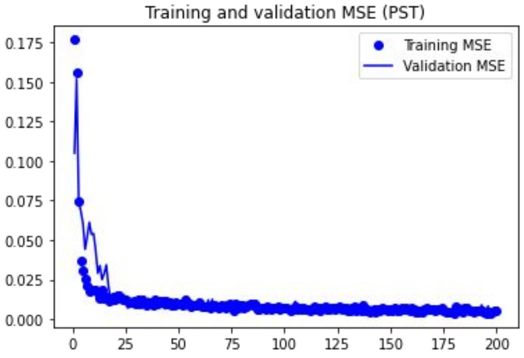
Prediction	Actual
.510997	.526367
.041219	.078125
.060580	.058594
.643715	.649414
.132602	.142578
.208487	.186523
.758020	.781250

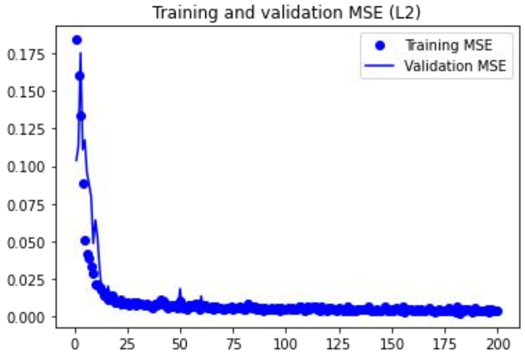
Similar circuit

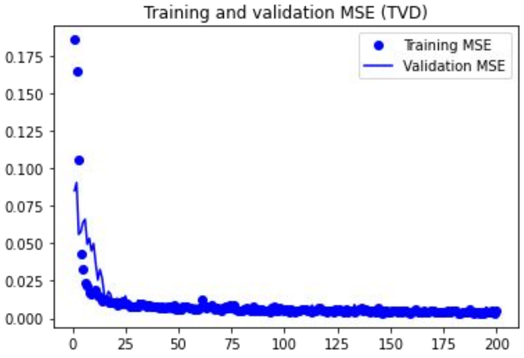
Prediction	Actual
.134381	.612305
.083689	.344727
.144355	.000977
.191454	.465820
.175614	.560547
.142928	.721680
.134386	.617187

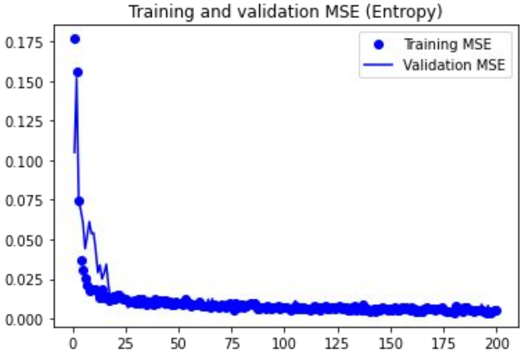
New circuit

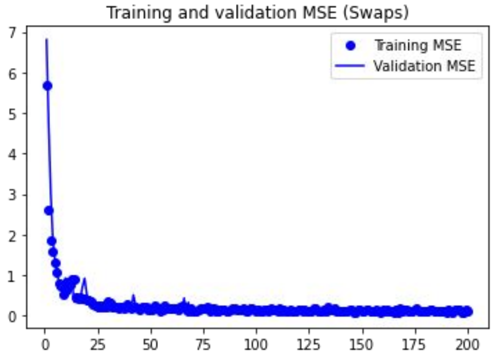
Results (V2)

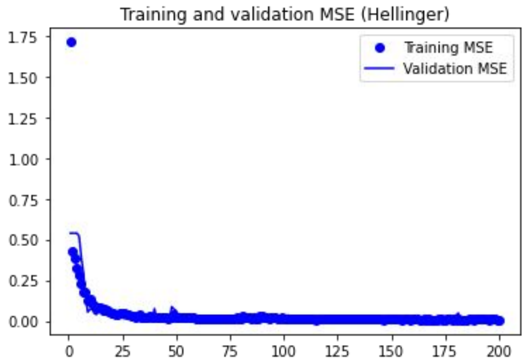

Training and validation MSE (PST)
Training MSE
Validation MSE
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
0 25 50 75 100 125 150 175 200


Training and validation MSE (L2)
Training MSE
Validation MSE
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
0 25 50 75 100 125 150 175 200


Training and validation MSE (TVD)
Training MSE
Validation MSE
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
0 25 50 75 100 125 150 175 200


Training and validation MSE (Entropy)
Training MSE
Validation MSE
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
0 25 50 75 100 125 150 175 200


Training and validation MSE (Swaps)
Training MSE
Validation MSE
7
6
5
4
3
2
1
0
0 25 50 75 100 125 150 175 200


Training and validation MSE (Hellinger)
Training MSE
Validation MSE
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00
0 25 50 75 100 125 150 175 200

Results (V2)

Prediction	Actual
.741174	.765625
.696171	.649414
.862628	.898438
.865645	.875000
.663417	.703125
.125723	.152344
.740014	.721680

Similar circuit

Prediction	Actual
.536968	.137695
.625335	.306641
.407491	.086914
.358605	.133789
.439015	.068359
.458579	.078125
.726478	.294922

New circuit

Conclusions

- Quarry: Tool for providing rapid quantum circuit fidelity estimation.
- Simulation
  - Detailed results, too slow for usage in tool.
  - Scales poorly in time/memory usage.
- ESP
  - Relatively cheap, provides limited details.
- Predictor V1/V2
  - Fast, currently inaccurate.
  - Memory intensive (WRT ESP).

Image Source: https://www.youtube.com/watch?v=ljz5uQQ_HkM

Future Works

- Optimize selection of machines
- Fitness function redesign
  - Better input/output metrics
- Neural Network Design
  - Look for more sophisticated model
  - Data sanitization
  - Find more efficient data generation methods
- Optimize query and remove redundant calculations
- Add compiler optimization options to search space
- Plugin/Frontend to visualize output

References

- [2.1] Konstantinos Georgopoulos, Clive Emary, and Paolo Zuliani. Modeling and simulating the noisy behavior of near-term quantum computers
- [2.2] Easwar Magesan, Daniel Puzzuoli, Christopher E. Granade, and David G. Cory. Modeling quantum noise for efficient testing of fault-tolerant circuits
- [4.2] Marco Cerezo, Alexander Poremba, Lukasz Cincio, and Patrick J. Coles. Variational Quantum Fidelity Estimation
- [5.3] Steven T. Flammia and Yi-Kai Liu. Direct Fidelity Estimation from Few Pauli Measurements.
- [5.6] Nishio, Shin and Pan, Yulu and Satoh, Takahiko and Amano, Hideharu and Meter, Rodney Van. Extracting Success from IBM's 20-Qubit Machines Using Error-Aware Compilation
- [5.7] Xiaoqian Zhang, Maolin Luo, Zhaodi Wen, Qin Feng, Shengshi Pang, Weiqi Luo, and Xiaoqi Zhou. Direct Fidelity Estimation of Quantum States Using Machine Learning