Quantum-classical computing at multiple time scales

Blake R Johnson, PhD
Distinguished Research Scientist, Quantum Engine Lead

IBM **Quantum**

May 2016:

IBM **Quantum**

First programmable quantum computer in the cloud with a simple composer for developing quantum circuits

May 3, 2016
Posted in: Quantum Computing

Quantum computing: It's time to build a quantum community

The New York Times

IBM Wants Everyone to Try a Quantum Computer

By John Markoff

May 4, 2016

WIRED

CADE METZ BUSINESS 05.04.16 12:01 AM

IBM IS NOW LETTING ANYONE PLAY WITH ITS QUANTUM COMPUTER

TechCrunch

IBM launches quantum computing as a cloud service

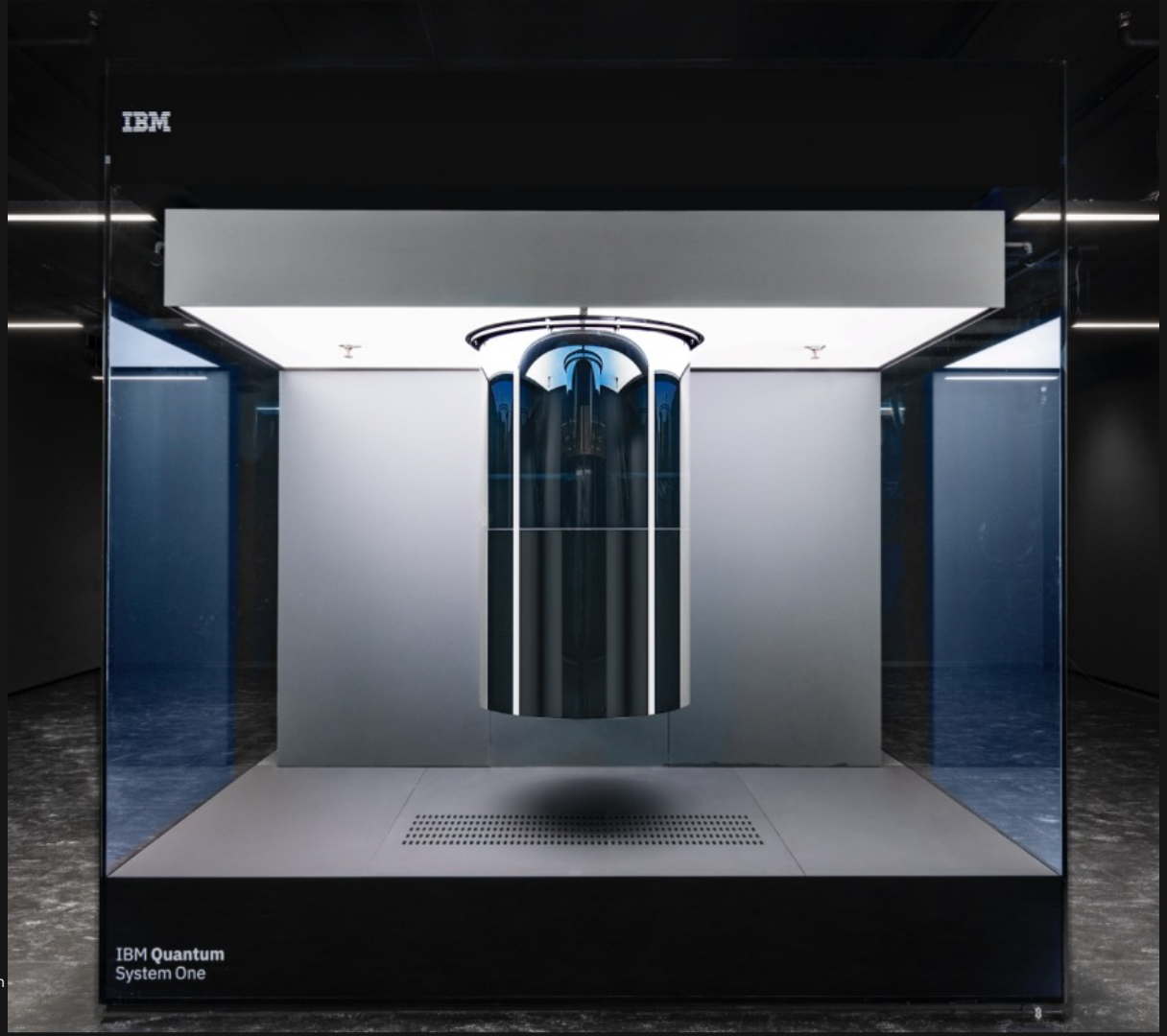
Ron Miller @ron_miller / May 4, 2016

Today: published 1500+ papers using IBM Quantum.

Today: 4 billion quantum circuits each day are run over the cloud

2019

IBM Quantum
System One



2022

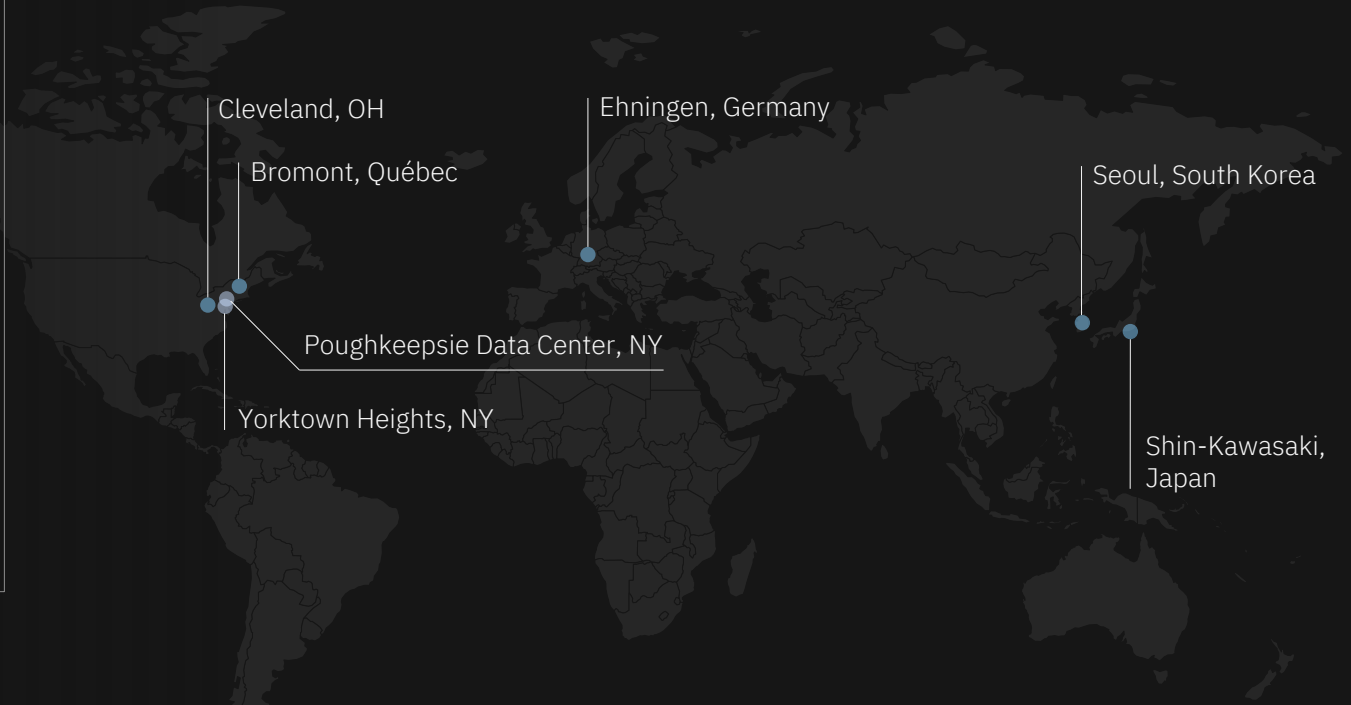
IBM Quantum data center

IBM Quantum Datacenters

IBM Poughkeepsie Data Center

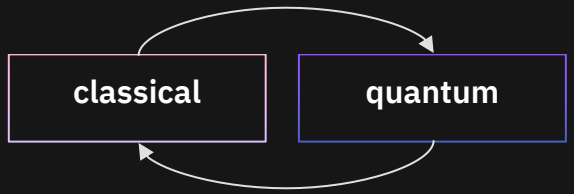
Global Deployment of IBM Quantum Computation Centers

Cleveland Clinic · PINQ² · Cornell University · 東京大学 THE UNIVERSITY OF TOKYO · Fraunhofer



60+ IBM Quantum systems deployed since 2016

Structure of a typical workload



Real workloads are not purely quantum, but rather require interaction between quantum and classical compute resources.

Time scales and resources for quantum computing

User programs

User programs
Seconds to days

Model developers
Enterprises

Quantum serverless

Elastic
100 miliseconds to hours

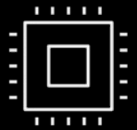
Algorithm Developers
Model developers

Qiskit Runtime

Near-time
Microseconds to seconds

Kernel Developers
Algorithm Developers

Dynamic circuits

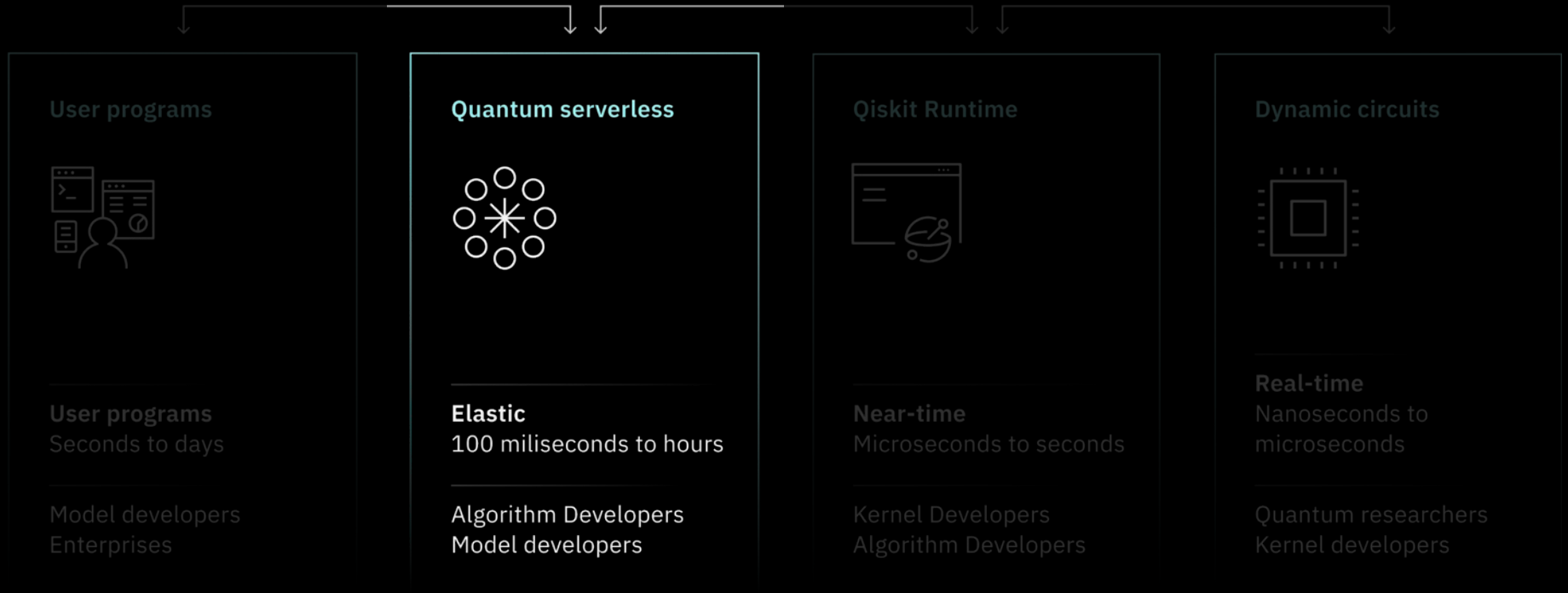


Real-time
Nanoseconds to microseconds

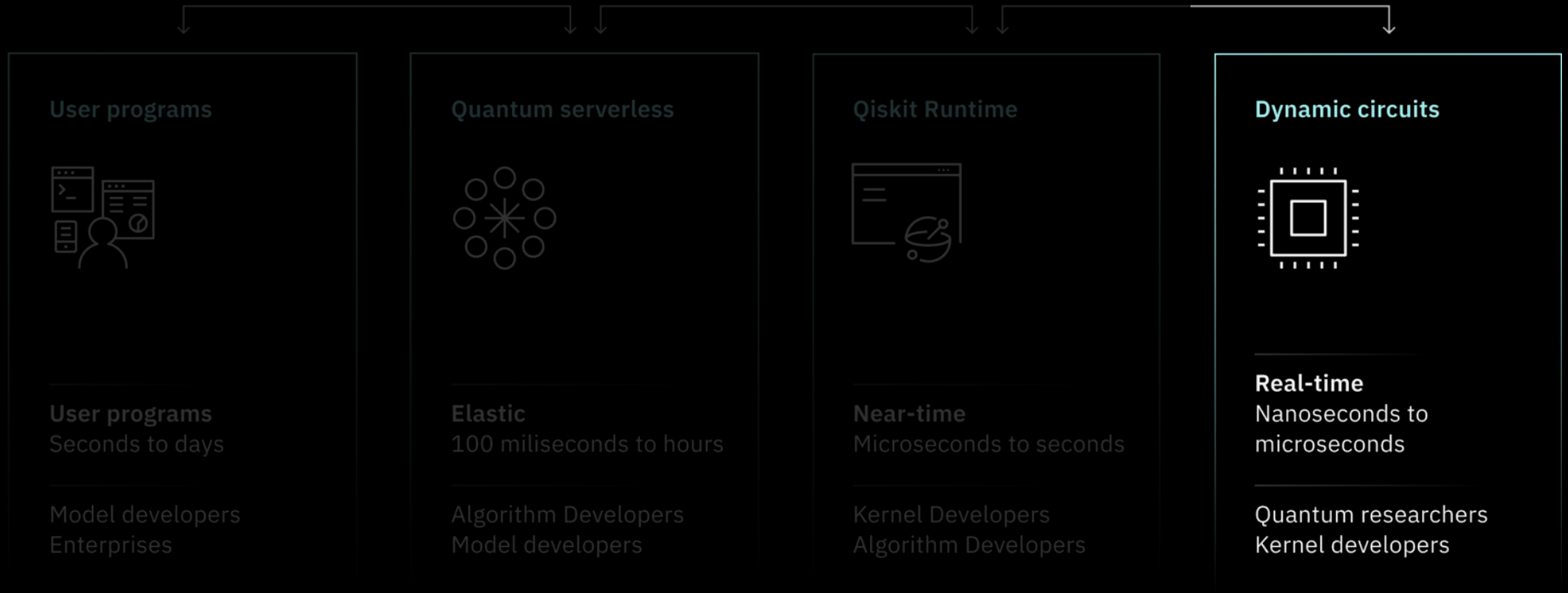
Quantum researchers
Kernel developers

Time scales and resources for quantum computing

IBM **Quantum**

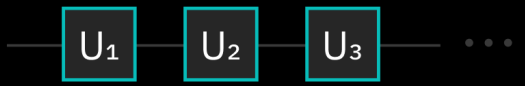


Time scales and resources for quantum computing



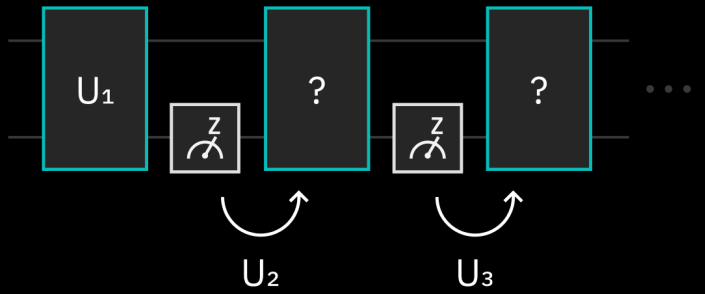
What are dynamic circuits ?

01 Regular quantum circuit



02 Dynamic circuit

New features:
- Mid-circuit measurements
- Real-time classical computing
- Feedforward



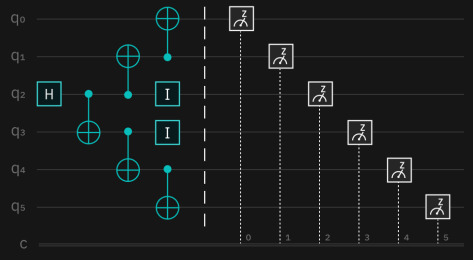
Power of Dynamic Circuits

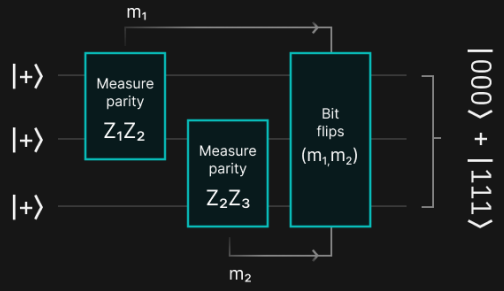
IBM **Quantum**

Reducing Circuit Depth $N^2 \rightarrow O(1)$

Cost N^2 auxiliary qubits

State preparation

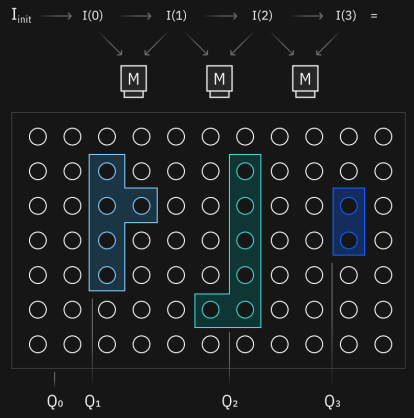




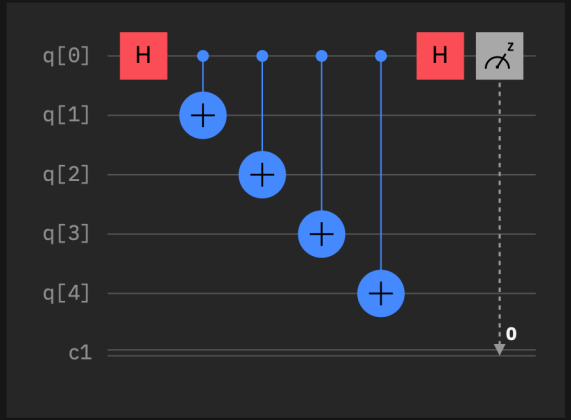
Alternative models for algorithms

Repeat until success[1]

Measurement model[2]



Parity checks for QEC



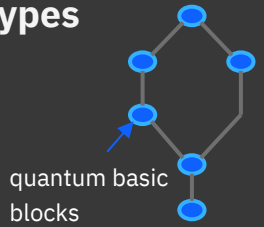
[1] Bocharov et al, PRL **114**, 080502 (2015)
[2] Raussendorf et al, PRA **68**, 022312 (2003)

OpenQASM3

A language for expressing dynamic circuits.

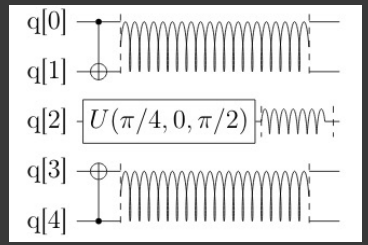
IBM **Quantum**

Classical control flow, instructions, and data types



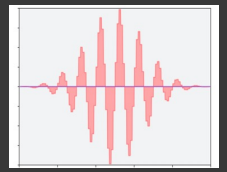
TeX-inspired timing relationships

Boxes and glues in TeX



Boxes and stretches in OpenQASM3

Connect pulse-level descriptions to gates

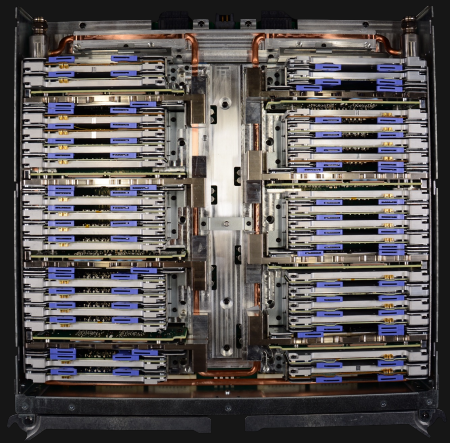


```
OPENQASM 3.0;
qubit q;    // phase estimation qubit
qubit r;    // target qubit for the controlled-unitary gate
angle[16] c;  // phase estimation bits


// prepare uniform superposition of eigenvectors of
phase
h r;

// iterative phase estimation loop
for i in [1:n] {
  reset q;
  h q;
  ctrl @ pow[2 ** i] @ U q, r;
  inv @ phase(c) q;
  h q;
  measure q -> c[0];
  // next iteration acts on the next bit
  c <<= 1;
}
```

A. Cross et al, arXiv:2104.14722
accepted to ACM Transactions on QC

Enabling Technology for Dynamic Circuits

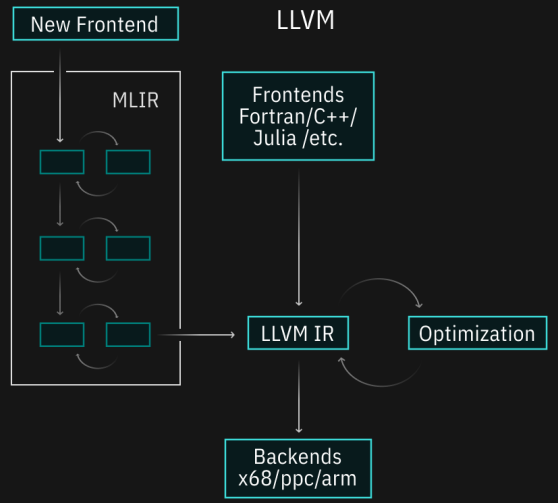
3rd Generation Control System



OpenQASM3 – *circuit language description*

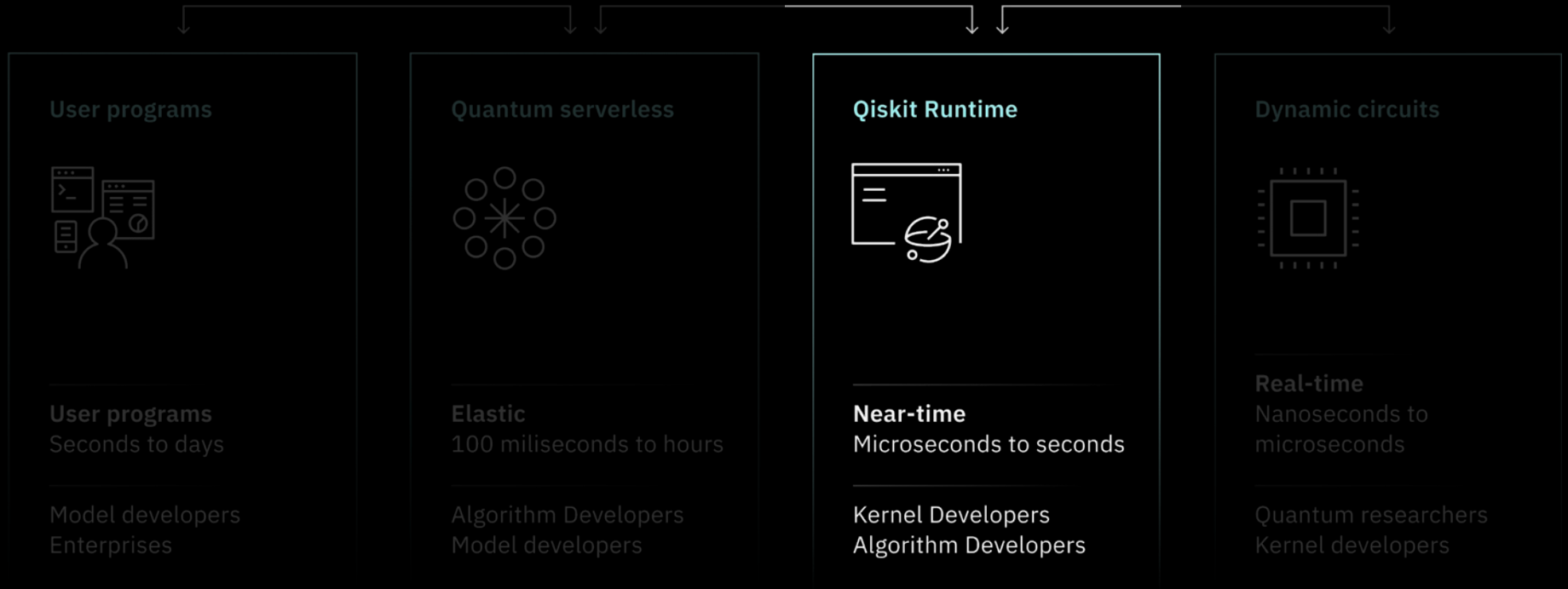
```
OPENQASM 3.0;
qubit q;
qubit r;
angle[16] c;

h r;
for i in [1:n] {
  reset q;
  h q;
  ctrl @ pow[2 ** i] @ U q, r;
  inv @ phase(c) q;
  h q;
  measure q -> c[0];
  c <<= 1;
}
```

OpenQASM3 – *native compiler*



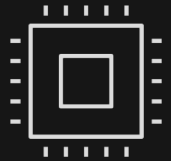
Time scales and resources for quantum computing

IBM **Quantum**

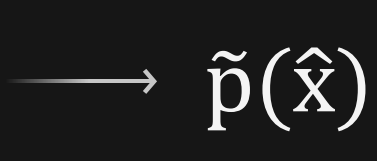


Core Primitives

Sampler

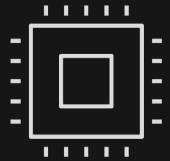


Circuit

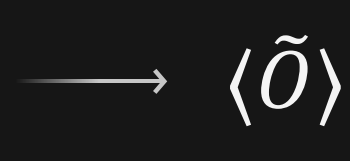


Quasi-probability distribution

Estimator

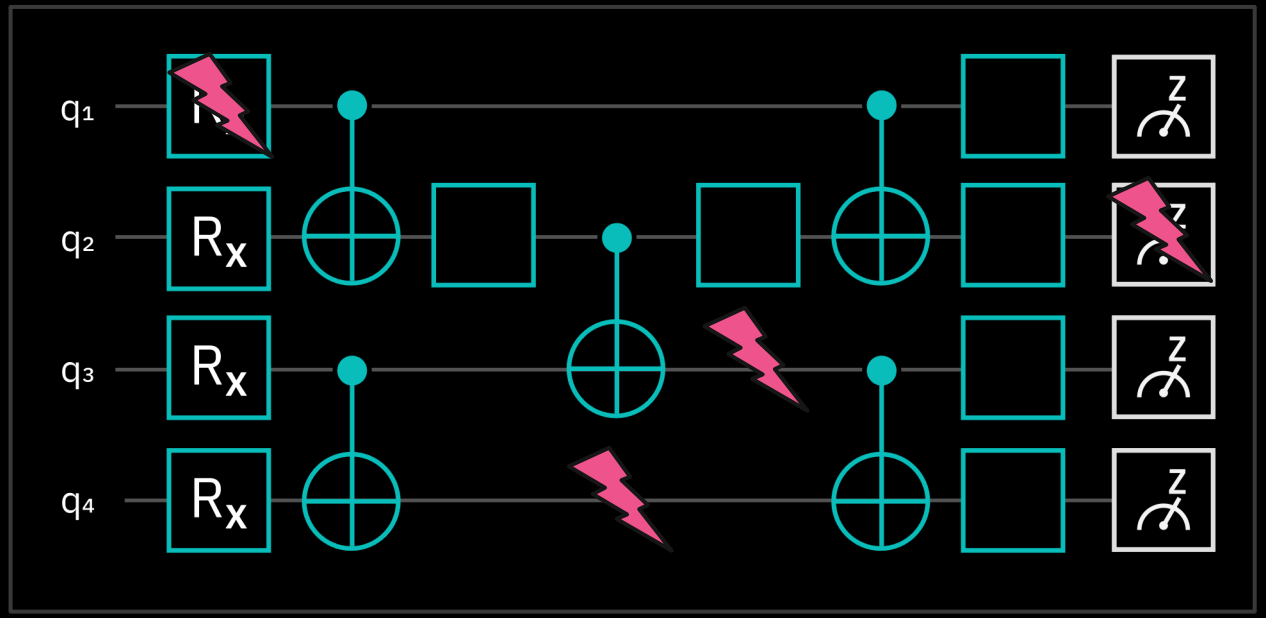


Circuit
+
Observable
$\tilde{O}$



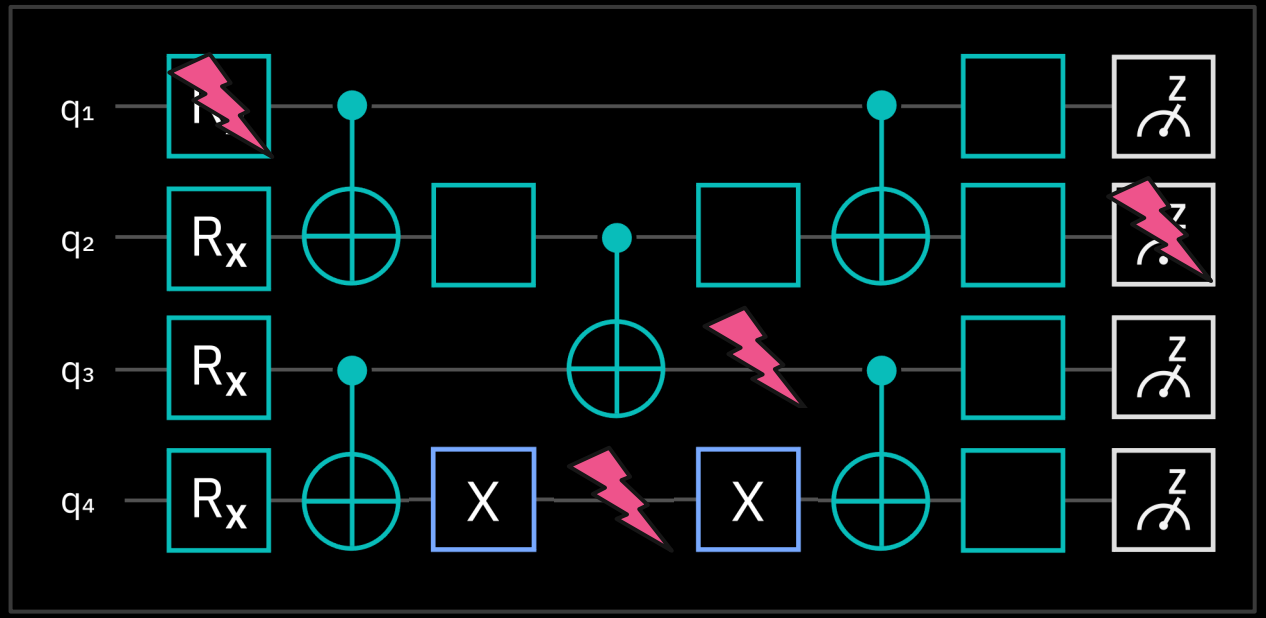
Expectation value

In quantum computation,
we must deal with errors.



In quantum computation, we must deal with errors.

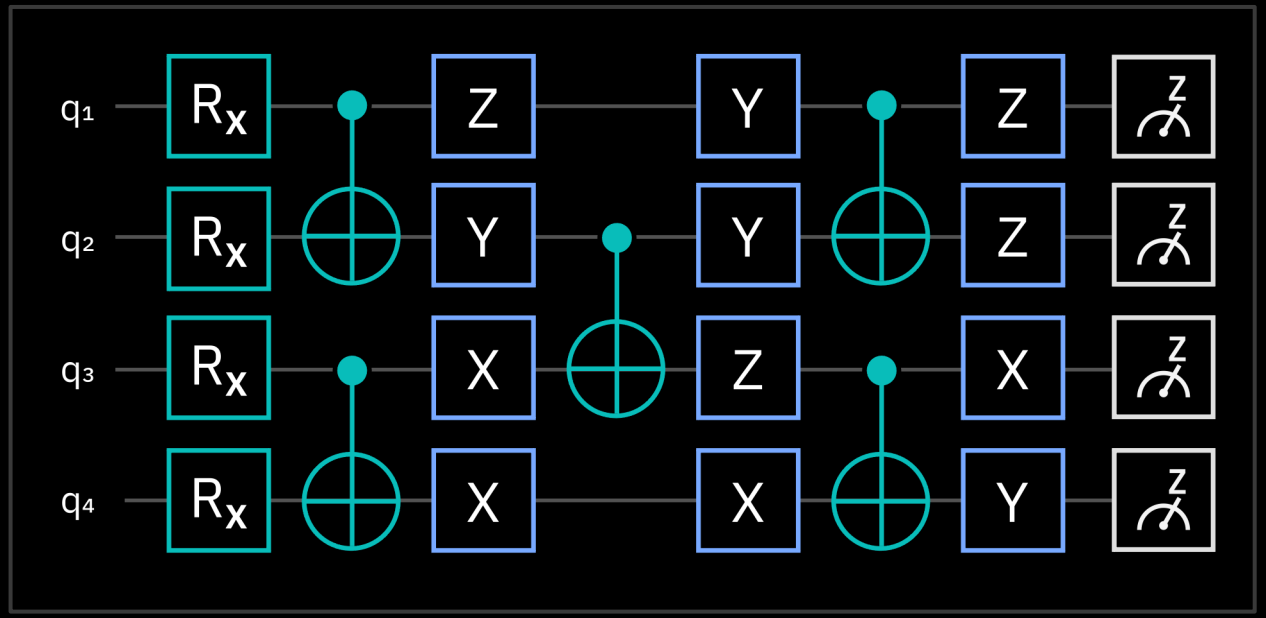
Error suppression reduces errors by modifying the circuit.



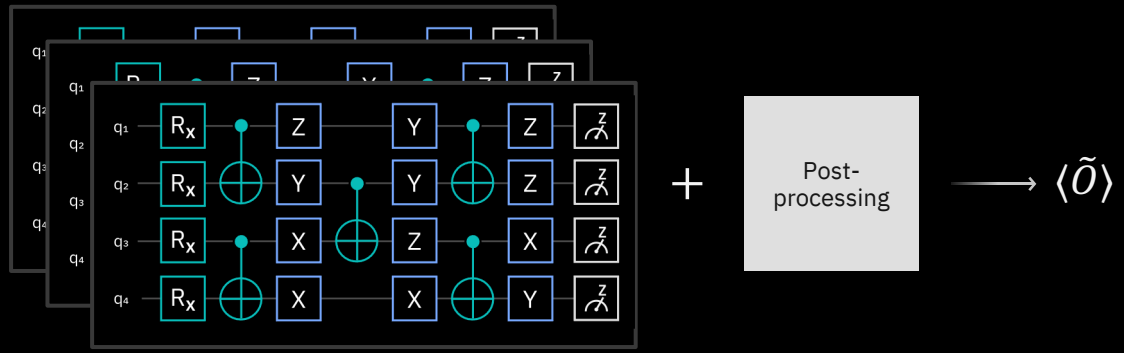
In quantum computation, we must deal with errors.

Error suppression reduces errors by modifying the circuit.

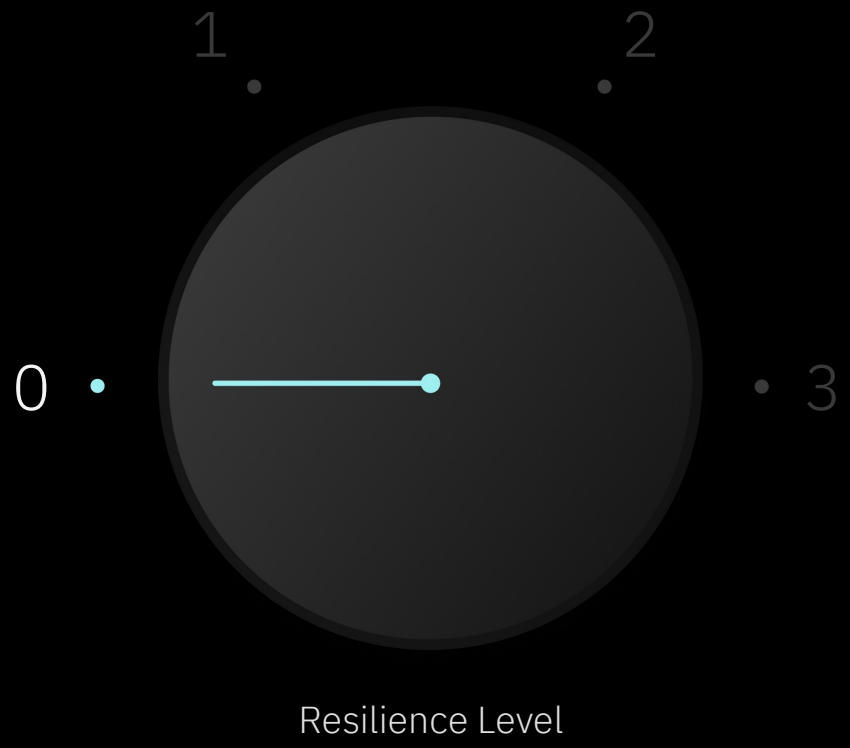
Error mitigation uses outputs from ensembles of circuits to increase accuracy in expectation values.



Error mitigation uses outputs from ensembles of circuits to increase accuracy in expectation values.

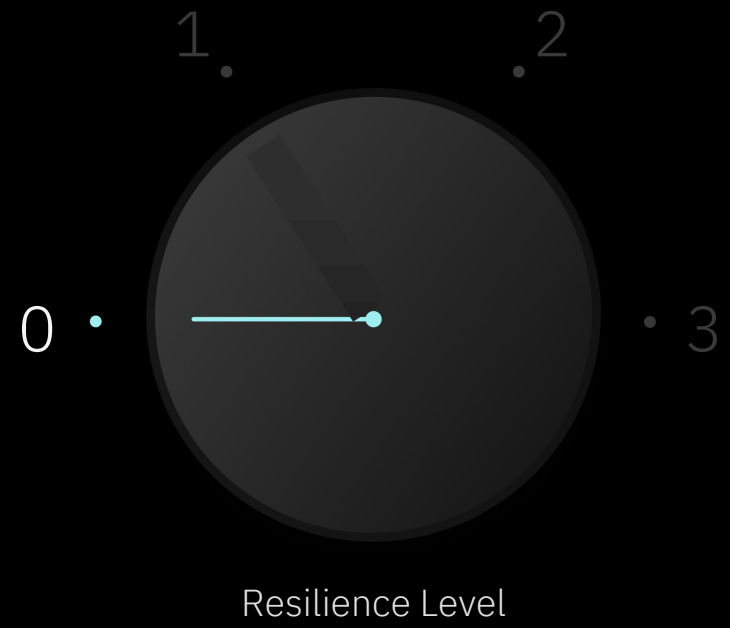


Simple interfaces for adjusting accuracy



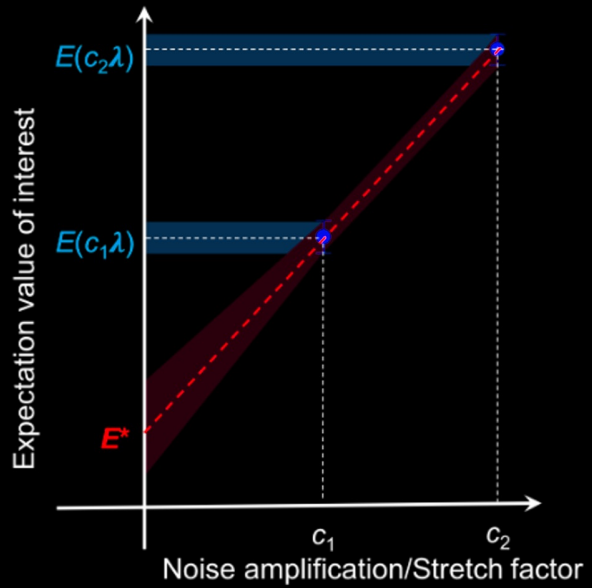
Measurement error mitigation is almost “free”

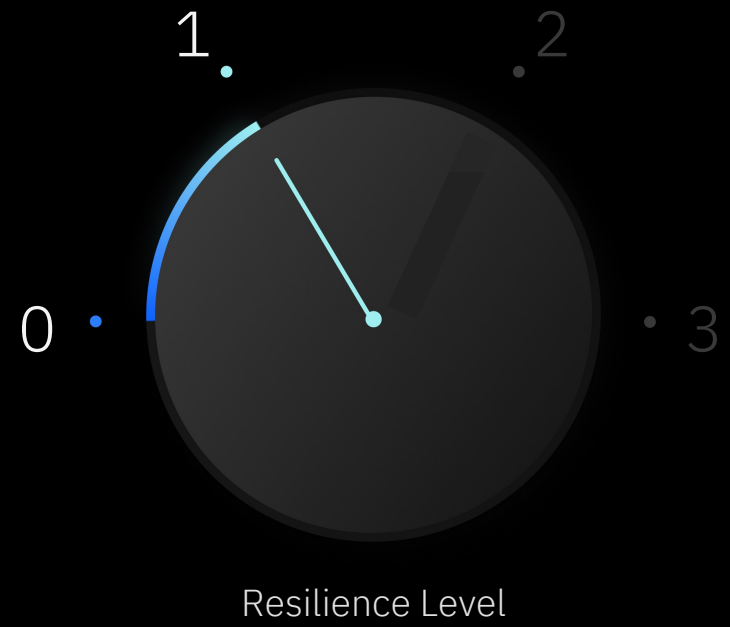
Primitive	Error Mitigation Method
Sampler	Model-based measurement error mitigation (M3)
Estimator	Model-free measurement error mitigation (TREX)



Error mitigation extends further at the cost of sampling overhead

Zero noise extrapolation

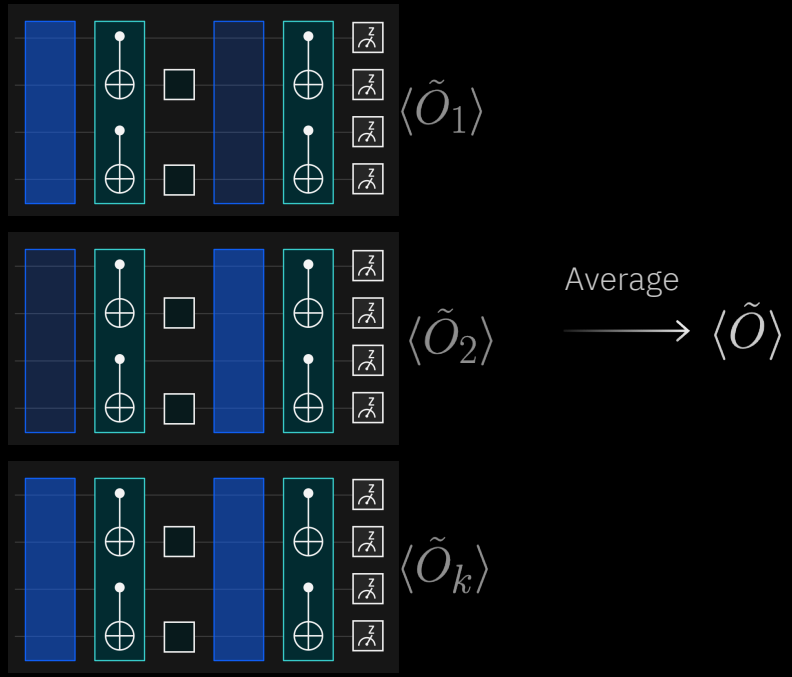


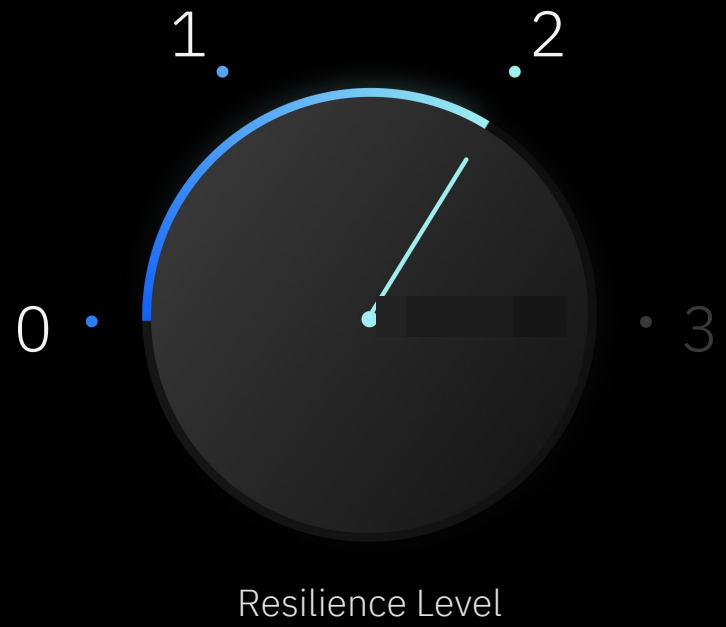


E.g. A. Kandala et al, Nature 567, 491 (2019)

More cost, more accuracy

Probabilistic Error Cancellation

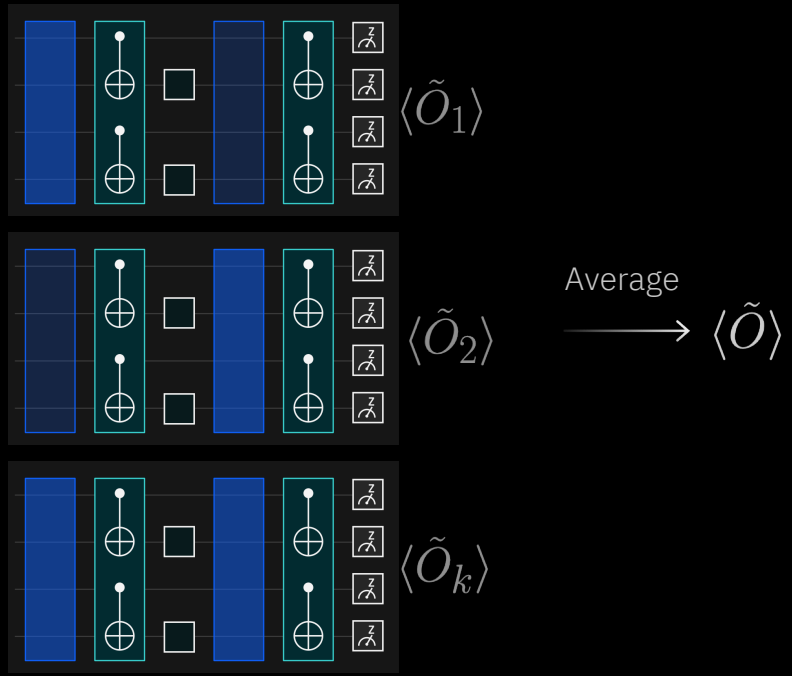




E. van de Berg, Z. Minev, K. Temme, A. Kandala, arXiv:2201.09866

Or in code...

Probabilistic Error Cancellation




```
options = Options(resilience_level=3)

with Session(service) as session:

    estimator = Estimator(session, options)

    job = estimator.run(circuits=[psi1],
                        observables=[H1],

parameter_values=[theta])
```

E. van de Berg, Z. Minev, K. Temme, A. Kandala, arXiv:2201.09866

Time scales and resources for quantum computing

User programs

User programs
Seconds to days

Model developers
Enterprises

Quantum serverless

Elastic
100 miliseconds to hours

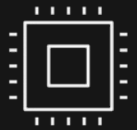
Algorithm Developers
Model developers

Qiskit Runtime

Near-time
Microseconds to seconds

Kernel Developers
Algorithm Developers

Dynamic circuits



Real-time
Nanoseconds to microseconds

Quantum researchers
Kernel developers


IBM Quantum
Hummingbird