Parallel GPU Quantum Circuit Simulations on Qiskit Aer

Jun Doi
IBM Quantum, IBM Research – Tokyo

doichan@jp.ibm.com

IBM **Quantum**

Qiskit Aer Overview

Qiskit is an opensource platform for quantum computing

Qiskit Aer is one of the components of Qiskit, an opensource quantum circuits simulator

Quantum circuits can be executed both on hardware and simulator

Qiskit Aer supports various simulation methods and noise models

- Statevector
- Unitary
- Density matrix
- Stabilizer
- MPS

- Pauli noise
- Kraus/superoperator noise
- etc..

Qiskit Aer GPU Support

Qiskit is an opensource platform for quantum computing

Qiskit Aer is one of the components of Qiskit, an opensource quantum circuits simulator

Quantum circuits can be executed both on hardware and simulator

Qiskit Aer supports various simulation methods and noise models

- Statevector
- Unitary
- Density matrix

GPU supported

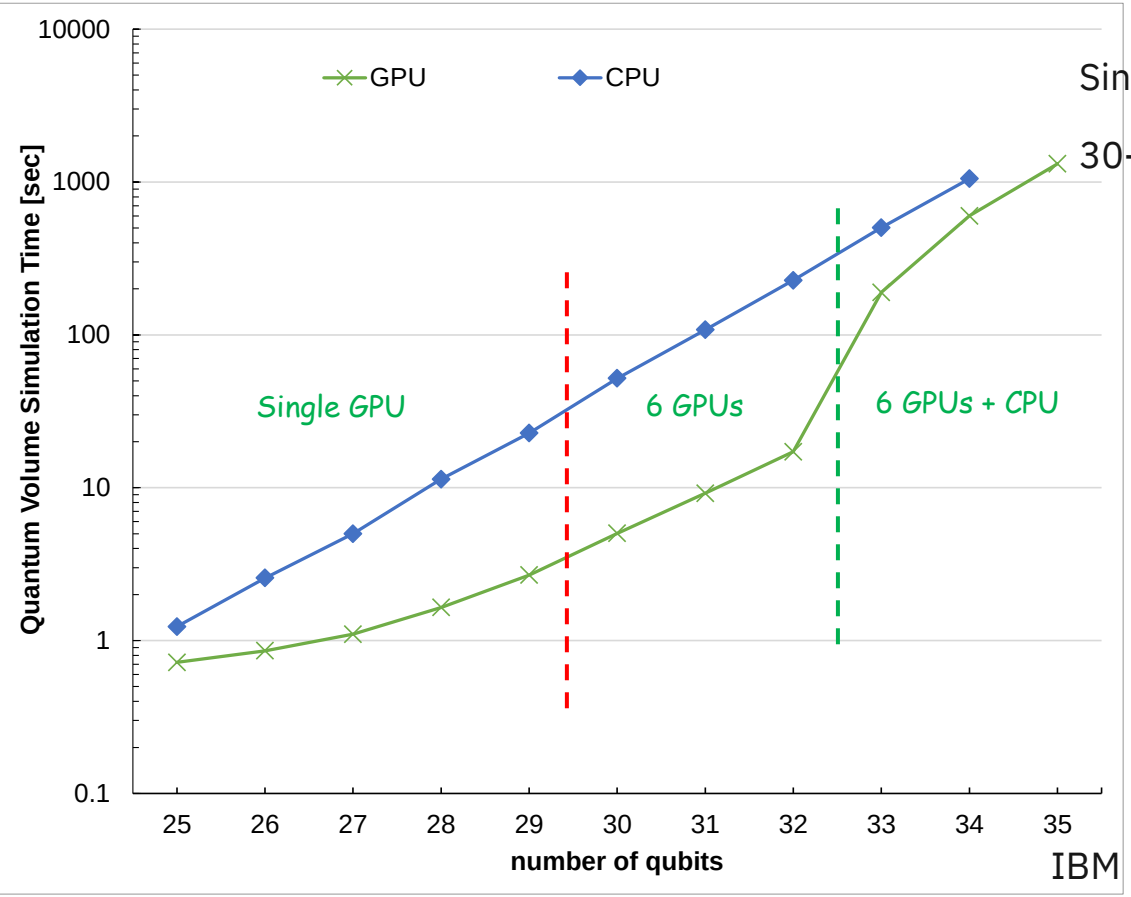
- Stabilizer
- MPS
- Tensor network

GPU support in plan

- Pauli noise
- Kraus/superoperator noise
- etc..

GPU Simulation Performance

Quantum Volume Circuit Simulation



Single GPU has 16 GB, <= 29 qubits statevector

30-32 qubits statevector can be stored on 6 GPUs

IBM Power System AC922 (6x NVIDIA Tesla V100)

Installing GPU Ready Qiskit Aer

1. Install Qiskit
   - pip install qiskit
2. Uninstall Qiskit Aer
   - pip uninstall qiskit-aer
3. Install GPU version of Qiskit Aer
   - pip install qiskit-aer-gpu

Simulating Quantum Circuits Using GPU

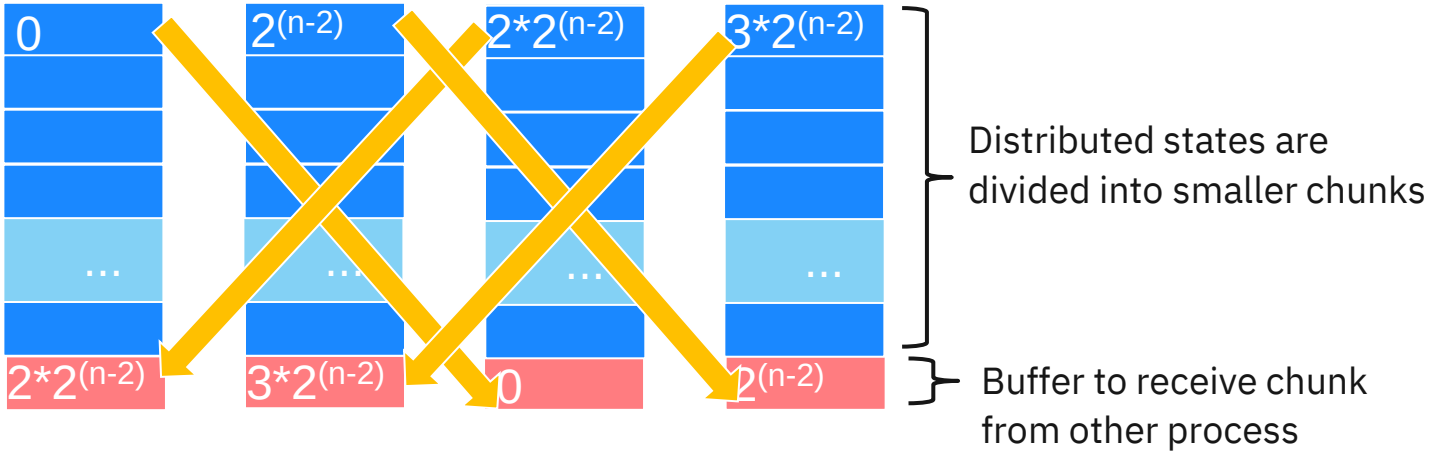

```
from qiskit import *
from qiskit.circuit.library import *
from qiskit.providers.aer import *

sim = AerSimulator(method='statevector', device='GPU')

shots = 100
depth=10
qubits = 25
circuit = transpile(QuantumVolume(qubits, depth, seed=0),
                    backend=sim,
                    optimization_level=0)
circuit.measure_all()

result = execute(circuit,sim,shots=shots,seed_simulator=12345).result()
```

Parallel Quantum Circuit Simulation

IBM **Quantum**

For large qubits circuits, quantum states are distributed into multiple GPUs or multiple processes (MPI)

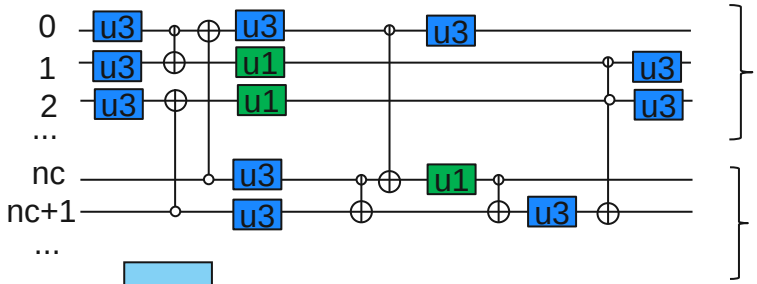
States are divided into chunk and data exchange is done per chunk to save memory space



Optimization of Parallel Quantum Circuit Simulation

IBM **Quantum**

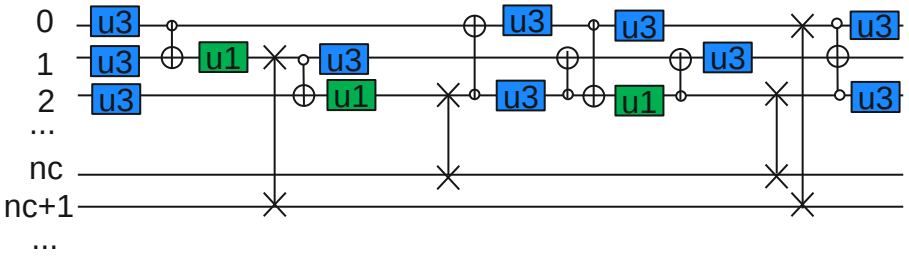
Transpiling quantum circuit before running simulation



< nc qubits gates can be applied inside chunk independently

>= nc qubits gates have to refer to other chunk that requires data exchange between chunks

Inserting swap gates to move all gates into chunk (< nc)

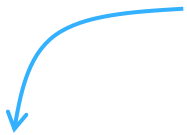


Now we can apply all gates without data exchange

There is no gate outside chunk, inserted swap gates only require data exchange

Simulating Quantum Circuits Using Multiple-GPUs


```
from qiskit import *
from qiskit.circuit.library import *
from qiskit.providers.aer import *

sim = AerSimulator(method='statevector', device='GPU', blocking_qubits=20)

shots = 100
depth=10
qubits = 25
circuit = transpile(QuantumVolume(qubits, depth, seed=0),
                    backend=sim,
                    optimization_level=0)
circuit.measure_all()

result = execute(circuit,sim,shots=shots,seed_simulator=12345).result()
```



Parallel Simulation on GPU Cluster (MPI)

There is no binary distribution of MPI supported Qiskit Aer, please build from source code

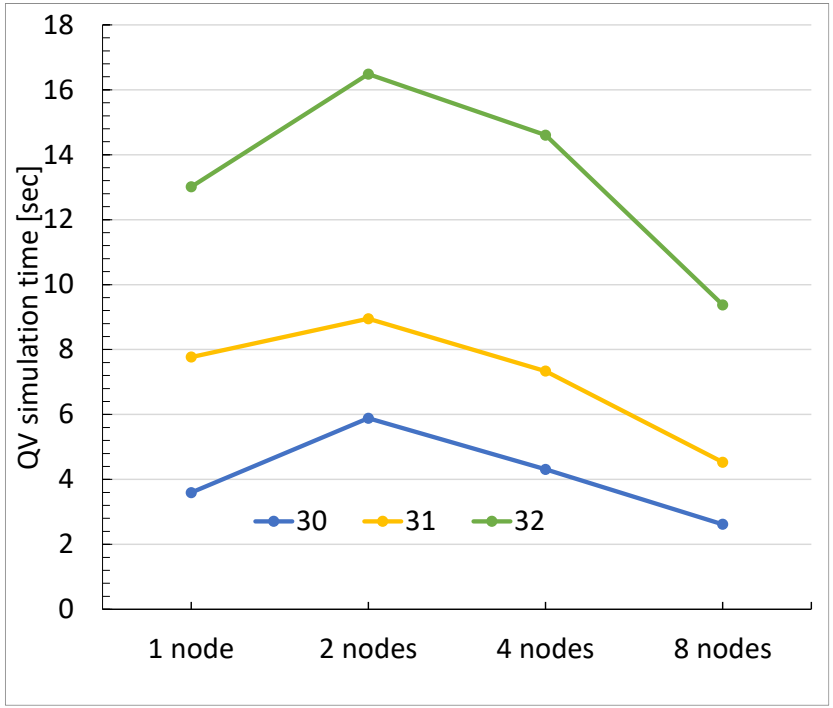
```
from qiskit import *
from qiskit.circuit.library import *
from qiskit.providers.aer import *
sim = AerSimulator(method='statevector', device='GPU', blocking_qubits=20)

shots = 100
depth=10
qubits = 35
circuit = transpile(QuantumVolume(qubits, depth, seed=0),
                    backend=sim,
                    optimization_level=0)
circuit.measure_all()
result = execute(circuit,sim,shots=shots,seed_simulator=12345).result()
if result.to_dict()['metadata']['mpi_rank'] == 0:
   print(sorted(result.to_dict()['results'][0]['data']['counts'].items(),key=lambda x:x[0]))
```

MPI Simulation Performance

IBM **Quantum**

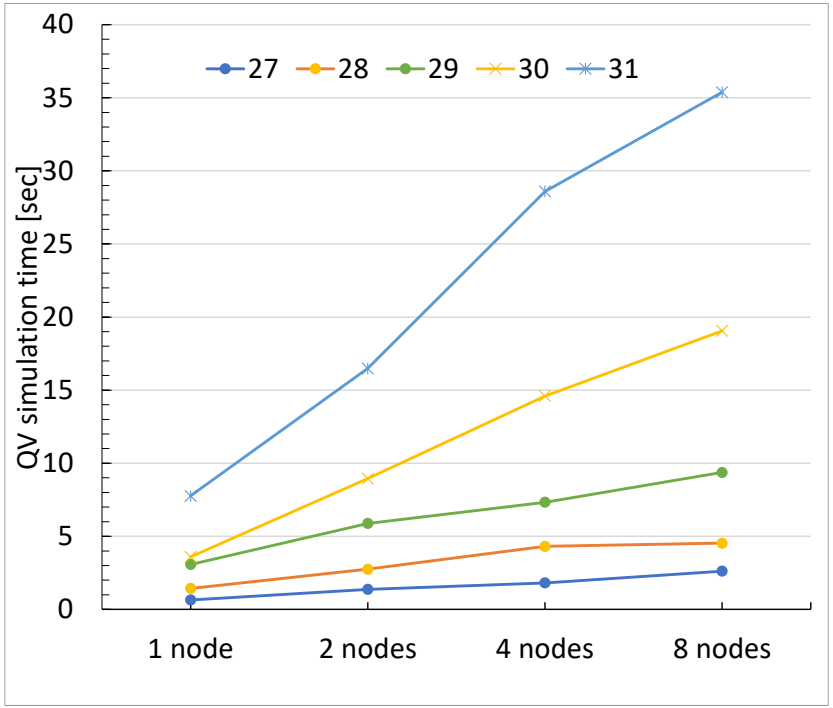
Strong scaling

(Fixed qubits of circuit)



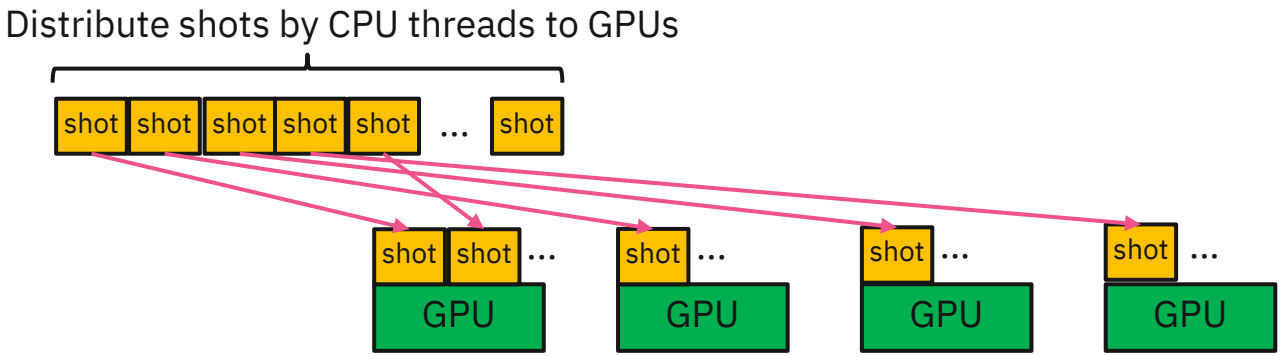
Weak scaling

(Fixed qubits / node)

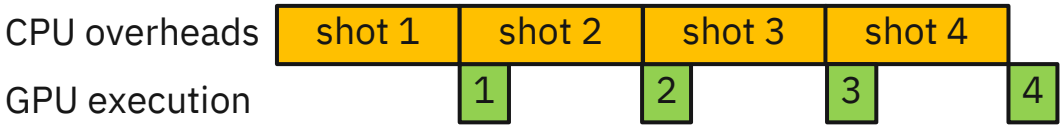


IBM Power System AC922 (6x NVIDIA Tesla V100)

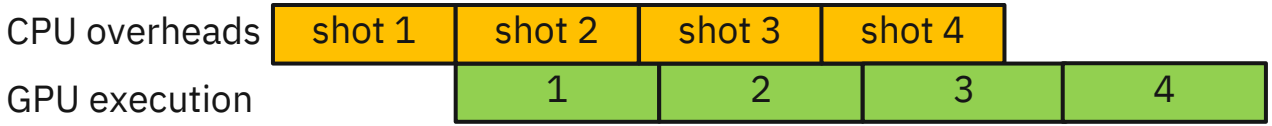
Shot Level Parallel Simulation



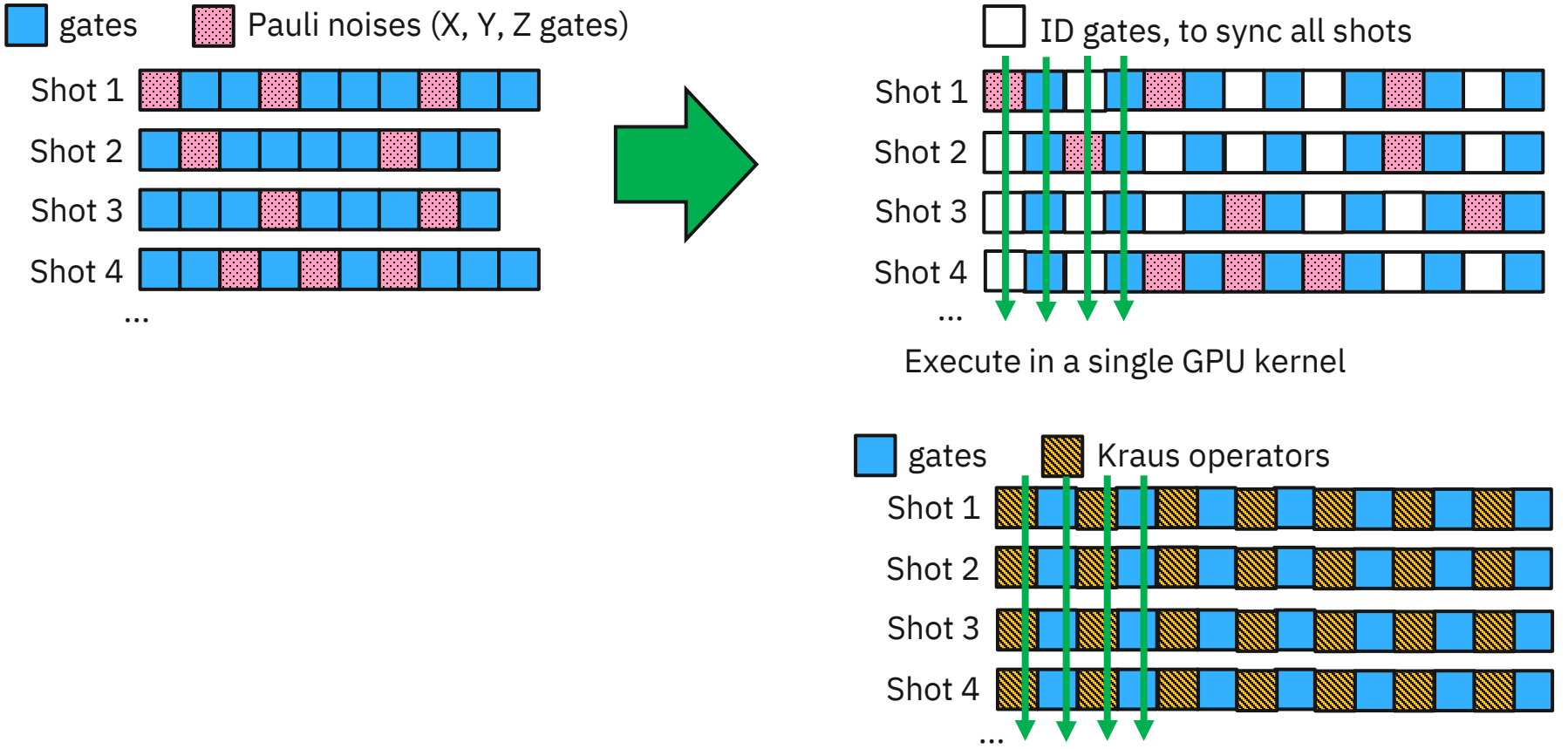
Problem for small qubits simulation



If qubits is larger enough, CPU overheads can be ignored ...

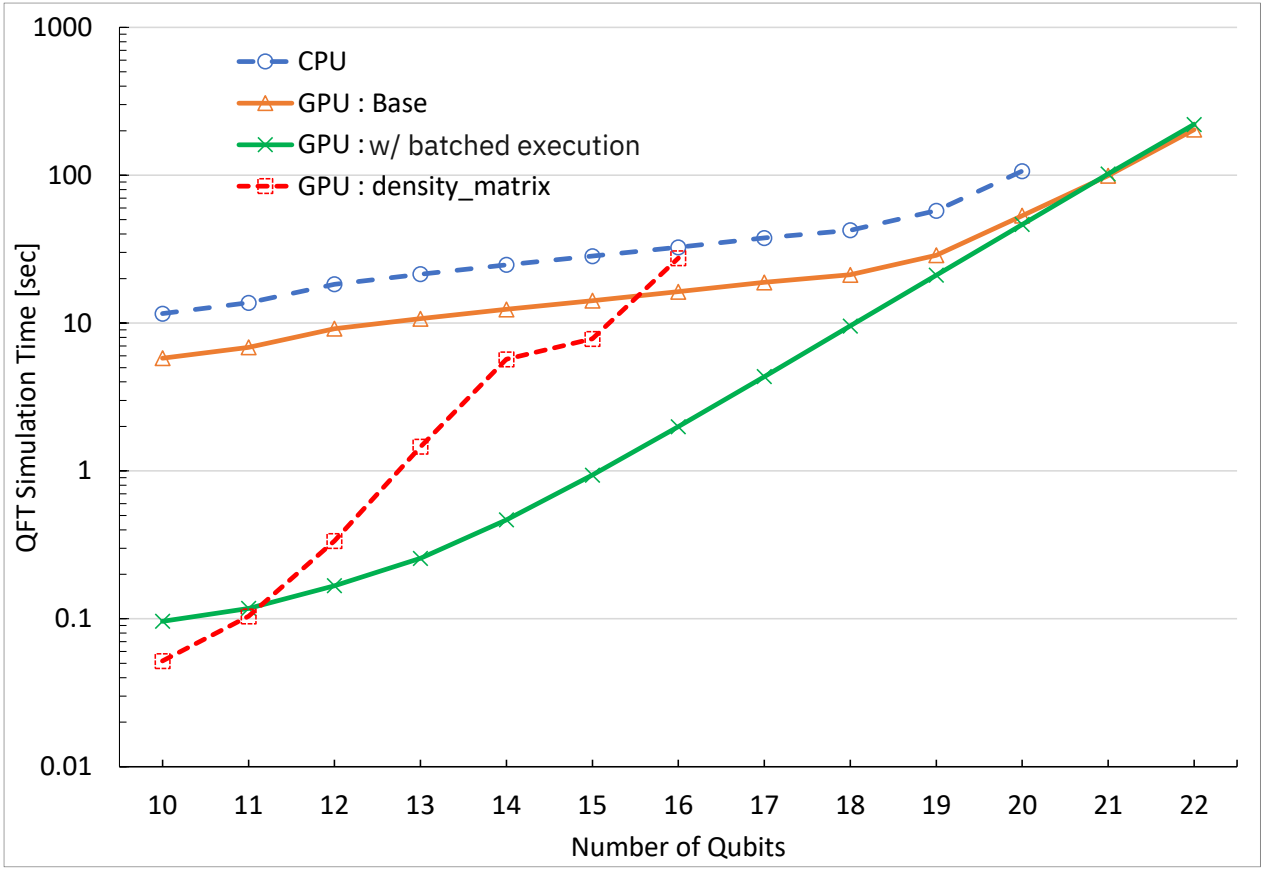


Multi-Shots Batched Execution on GPU



Multi-Shots Simulation Performance

IBM **Quantum**



Simulating QFT with 1% Kraus noise

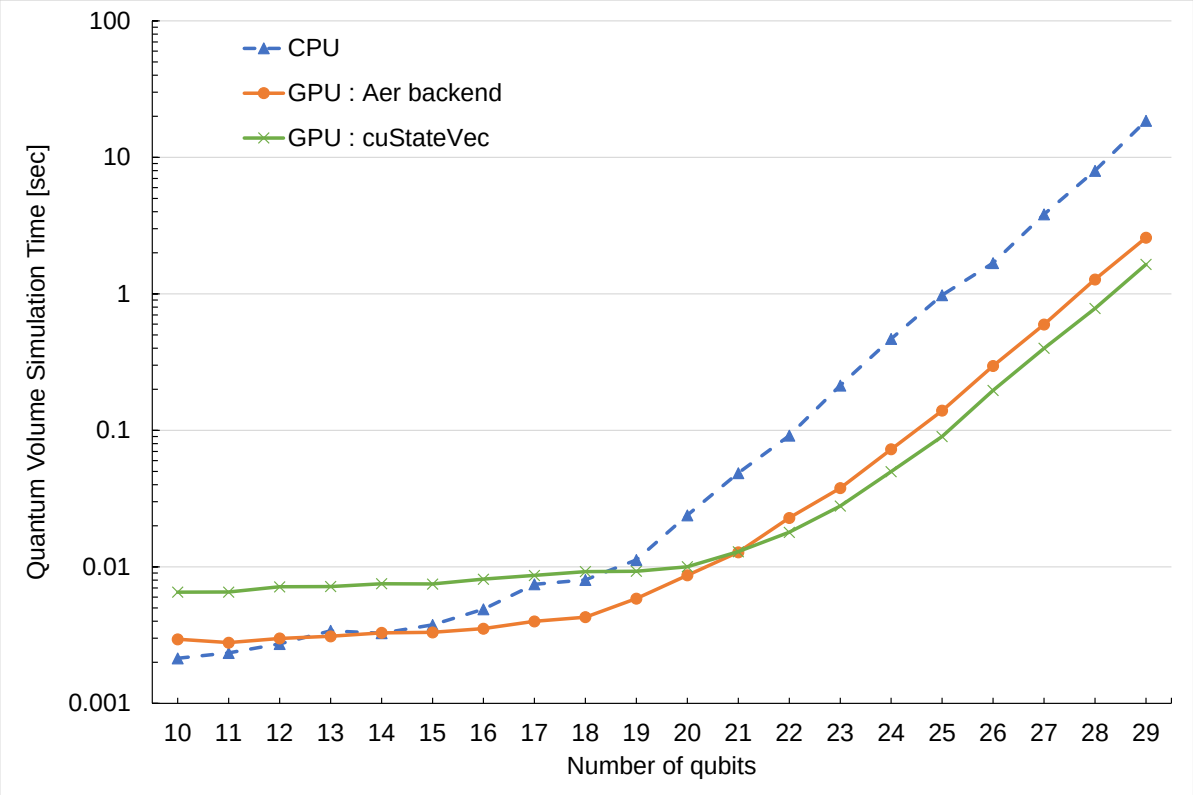
(4000 shots)

IBM Power System AC922 (6x NVIDIA Tesla V100)

Qiskit Aer cuQuantum Support

IBM **Quantum**

There is no binary distribution of cuQuantum supported Qiskit Aer, please build from source code



cuQuantum is enabled by setting ‘cuStateVec_enable=True’

Currently, statevctor, unitary and density matrix methods are supported

Summary

- Parallel Quantum Circuit Simulation on Qiskit Aer
  - Cache blocking transpiler to distribute larger qubits circuits
  - Batched execution to accelerate multi-shots simulations
- Plan
  - cuTensorNet supported tensor network simulator
  - Stabilizer simulator GPU support