Enhancement of Aer-based quantum_info

Shunsuke Sotobayashi
(Mentor: Hiroshi Horii (IBM Research – Tokyo))

Enhancement of Aer-based quantum_info

Improve performance of quantum_info with optimized runtime of Qiskit-Aer

Background:

- Qiskit-terra already has its quantum_info
- More performance by C++ is required
- Enhanced quantum_info should be implemented in Aer

At start point of this project:

- AerStatevector has been implemented in 0.11.0
- But other classes were not implemented yet:
  - AerDensityMatrix (target of this time!)

```
[1]: from qiskit import QuantumCircuit
     from qiskit_aer.quantum_info import AerStatevector

[2]: circuit = QuantumCircuit(2)
     circuit.h(0)
     circuit.cx(0, 1)
     sv = AerStatevector(circuit)
     sv.draw('latex')

[2]: $\frac{\sqrt{2}}{2}|00\rangle + \frac{\sqrt{2}}{2}|11\rangle$
```

Enhancement of Aer-based quantum_info

Improve performance of quantum_info with optimized runtime of Qiskit-Aer

Brief review for Checkpoints 1&2:

1. Study for preceding AerStatevector
   - Study its concept
   - Study its sequences of each procedure
   - etc.
2. Prototype implementation of AerDensityMatrix
   - Create and post related PRs
   - Discuss with my mentor each week
   - Also translate some notebooks of new Qiskit textbook (platypus)

Enhancement of Aer-based quantum_info

Improve performance of quantum_info with optimized runtime of Qiskit-Aer

By-products:

- Five PRs
  - Use AerStatevector in unittests #1621 (merged in qiskit-aer 0.11.1)
  - Support for initialization from Statevector #1644 (merged in qiskit-aer 0.11.2)
  - Add __init__.py to run tests for terra.states #1673
  - Fix a test #6 in mentor's branch
  - Accept matrix product state again #7 in mentor's branch
- Two translations of notebooks of Qiskit textbook (platypus)
  - https://gitlocalize.com/repo/7494/ja/notebooks/quantum-hardware/density-matrix.ipynb
  - https://gitlocalize.com/repo/7494/ja/notebooks/quantum-hardware/measuring-quantum-volume.ipynb

Enhancement of Aer-based quantum_info

Improve performance of quantum_info with optimized runtime of Qiskit-Aer

Finally:

- I created a draft PR
  - [Do not merge] Implement AerDensityMatrix #1676
- There are still several portions left to be discussed but it does work as expected!

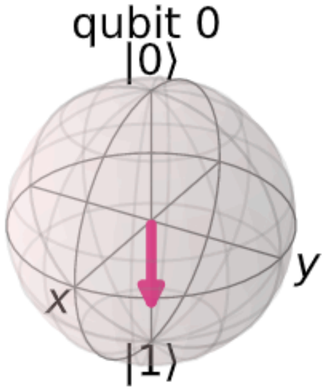
By the way:

- Its performance…?

[1]:
```
from qiskit import QuantumCircuit
from qiskit_aer.quantum_info import AerDensityMatrix
from qiskit.quantum_info import Kraus
from qiskit_aer.noise import pauli_error
```

[2]:
```
circuit = QuantumCircuit(1)
circuit.x(0)
error = pauli_error([('X', 0.1), ('I', 0.9)])
circuit.append(Kraus(error), [0])
dm = AerDensityMatrix(circuit)
fig = dm.draw('bloch')
fig.set_figwidth(fig.get_figwidth()*2//3)
fig.set_figheight(fig.get_figheight()*2//3)
fig
```

[2]:



Show time!!

[1]:
```
from qiskit import QuantumCircuit
from qiskit.quantum_info import DensityMatrix
from qiskit_aer.quantum_info import AerDensityMatrix
from qiskit.circuit.library import QuantumVolume
```

1 quibit circuits

[2]:
```
qc = QuantumVolume(1, seed=1111)
%timeit -o DensityMatrix(qc)
```

```
140 µs ± 1.46 µs per loop (mean ± std. dev. of 7 runs, 10,000 loops each)
```

[2]:
```
<TimeitResult : 140 µs ± 1.46 µs per loop (mean ± std. dev. of 7 runs, 10,000 loops each)>
```

[3]:
```
qc = QuantumVolume(1, seed=1111)
%timeit -o AerDensityMatrix(qc)
```

```
75.9 µs ± 658 ns per loop (mean ± std. dev. of 7 runs, 10,000 loops each)
```

[3]:
```
<TimeitResult : 75.9 µs ± 658 ns per loop (mean ± std. dev. of 7 runs, 10,000 loops each)>
```

- The result: AerDensityMatrix is about 2x faster than DensityMatrix at this point.
- Env: Google Cloud’s n1-highmem-4 instance (Ubuntu 18.04, 4 vCPU, RAM 26GB)

2 quibits circuits

[4]:
```
qc = QuantumVolume(2, seed=1111)
%timeit -o DensityMatrix(qc)
```

```
625 µs ± 6.9 µs per loop (mean ± std. dev. of 7 runs, 1,000 loops each)
```

[4]:
```
<TimeitResult : 625 µs ± 6.9 µs per loop (mean ± std. dev. of 7 runs, 1,000 loops each)>
```

[5]:
```
qc = QuantumVolume(2, seed=1111)
%timeit -o AerDensityMatrix(qc)
```

```
161 µs ± 2.75 µs per loop (mean ± std. dev. of 7 runs, 10,000 loops each)
```

[5]:
```
<TimeitResult : 161 µs ± 2.75 µs per loop (mean ± std. dev. of 7 runs, 10,000 loops each)>
```

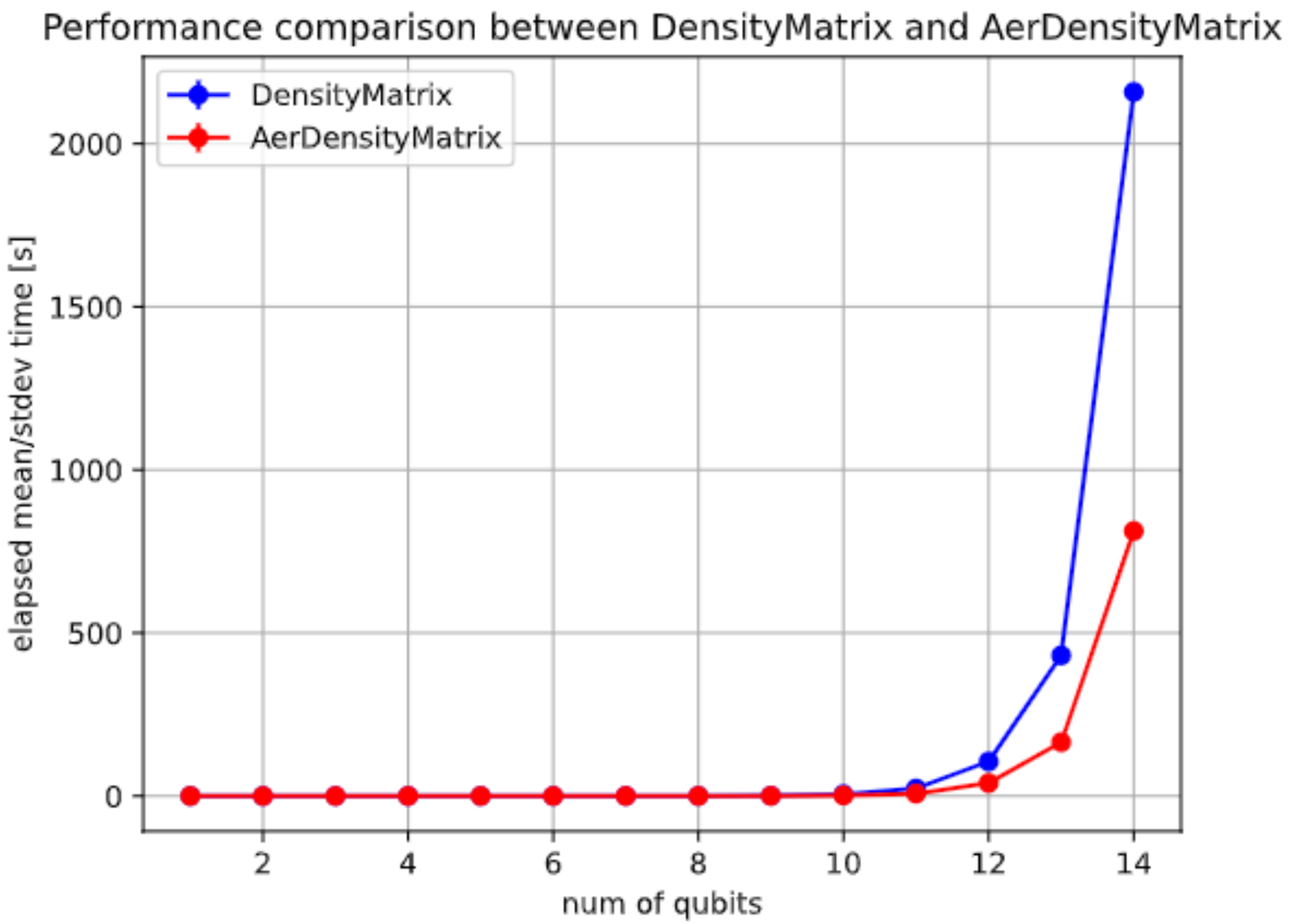
- The result: in this case, AerDensityMatrix is about 4x still faster.

[7]:
```
results_dm = []
results_adm = []

for n_qubits in range(1, 14+1):
    qc = QuantumVolume(n_qubits, seed=1111)
    if n_qubits <= 10:
        result = %timeit -o DensityMatrix(qc)
        results_dm.append([result.average, result.stdev])
        result = %timeit -o AerDensityMatrix(qc)
        results_adm.append([result.average, result.stdev])
    else:
        result = %timeit -n 1 -r 1 -o DensityMatrix(qc)
        results_dm.append([result.average, result.stdev])
        result = %timeit -n 1 -r 1 -o AerDensityMatrix(qc)
        results_adm.append([result.average, result.stdev])
```

- For more qubits…
- This experiments took very long time… (about 1 hour)

[13]:
```
plot_graph()
```



- The result: AerDensityMatrix is about 3x faster than DensityMatrix

Usage

Replace:

- from qiskit.quantum_info import DensityMatrix

With:

- from qiskit_aer.quantum_info import AerDensityMatrix

This is almost all you need to do! (and s/DensityMatrix/AerDensityMatrix/g)

Thank you very much!