

Qiskit noise simulator vs reality

Category: Teach Quantum

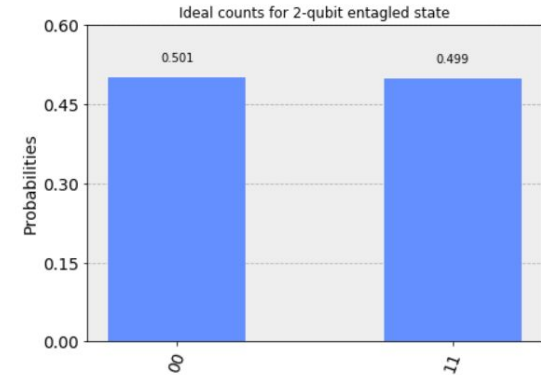
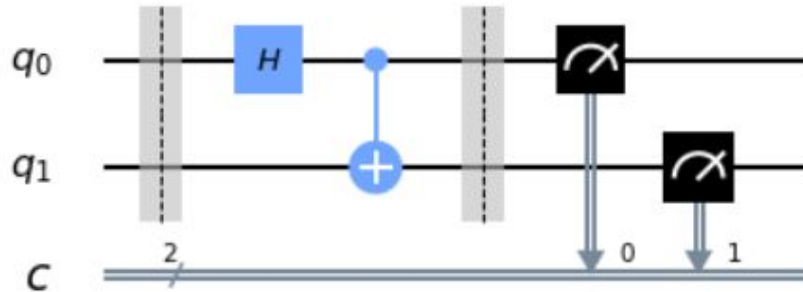
Introducción

- De la misma forma que se puede enviar los circuitos a ser ejecutados en máquinas reales, Qiskit permite realizar simulaciones locales
- La simulaciones locales son más “limpias” que las ejecuciones reales
- ¿Porque hay diferencia entre la ejecución real y la simulada?
 - La diferencia es que **la ejecución real está sujeta al ruido**
 - Para lidiar con este problema Qiskit dispone de herramientas para modelar el ruido



Experimento - Circuito de prueba

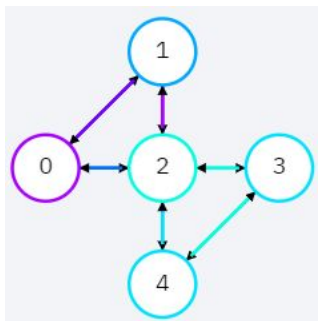
- Se ha seleccionado un circuito de dos Qubit entrelazados (estado de bell). Es un circuito simple con resultados conocidos.



Experimento - Chip de ejecución

- Los circuitos definidos en Qiskit se transpilan y se ejecutan en chips concretos.

Ej: ibmq_5_yorktown



```
# -----  
# Enparejamiento de Qubits del chip  
# -----  
device = provider.get_backend('ibmq_5_yorktown')  
properties = device.properties()  
coupling_map = device.configuration().coupling_map  
print(coupling_map)  
  
[[0, 1], [0, 2], [1, 0], [1, 2], [2, 0], [2, 1], [2, 3], [2, 4], [3, 2], [3, 4], [4, 2], [4, 3]]
```

Experimento - Modelo de ruido

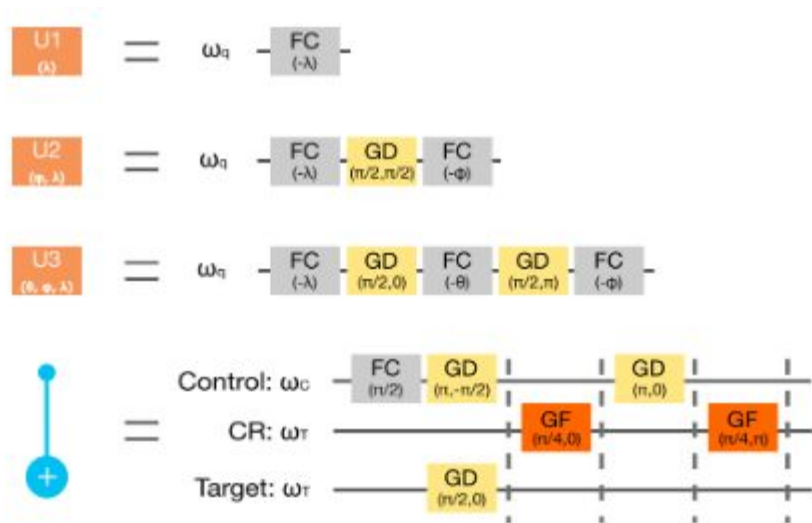
- Se puede modelar el ruido como una lista en la que los elementos son tripletas de valores (*<tipo puerta>*, *<qubit enlazados>*, *<tiempo>*)

```
gate_times = [  
    ('u1', None, 0), ('u2', None, 100), ('u3', None, 200)  
    , ('cx', [0, 1], 800)  
    , ('cx', [], 0)  
]  
  
# Construct the noise model from backend properties  
# and custom gate times  
noise_model = noise.device.basic_device_noise_model(properties, gate_times=gate_times)  
print(noise_model)
```

NoiseModel:

```
Basis gates: ['cx', 'id', 'u2', 'u3']  
Instructions with noise: ['measure', 'cx', 'id', 'u2', 'u3']  
Qubits with noise: [0, 1, 2, 3, 4]  
Specific qubit errors: [('id', [0]), ('id', [1]), ('id', [2]), ('id', [3]), ('id', [4]), ('u2', [0]), ('u2', [1]), ('u2', [2]), ('u2', [3]), ('u2', [4]), ('u3', [0]), ('u3', [1]), ('u3', [2]), ('u3', [3]), ('u3', [4]), ('cx', [0, 1]), ('cx', [0, 2]), ('cx', [1, 0]), ('cx', [1, 2]), ('cx', [2, 0]), ('cx', [2, 1]), ('cx', [2, 3]), ('cx', [2, 4]), ('cx', [3, 2]), ('cx', [3, 4]), ('cx', [4, 2]), ('cx', [4, 3]), ('measure', [0]), ('measure', [1]), ('measure', [2]), ('measure', [3]), ('measure', [4])]
```

Experimento - Tiempos ibmq_5_yorktown



- La puerta CNOT(cx) contiene
 - Una puerta FC \Rightarrow Que no cuenta
 - 3 puertas GD $\Rightarrow 3 \times 150\text{ns}$
 - 2 puertas GF $\Rightarrow 2 \times 190\text{ns}$
 - Buffer adicional de 5 puertas $\Rightarrow 5 \times 8\text{ns}$
 - Total $\Rightarrow 870\text{ns}$**
- Valores tomados de la información del chip de Yorktown

YORKTOWN DEVICE INFO

https://github.com/Qiskit/ibmq-device-information/blob/master/backends/yorktown/V1/version_log.md

GATES GRAPH

<https://github.com/Qiskit/ibmq-device-information/tree/master/backends/melbourne/V1>

Experimento - Modelo de ruido

```
gate_times = [  
    ('u1', None, 0), ('u2', None, 100), ('u3', None, 200)  
    , ('cx', [0, 1], 800)  
    , ('cx', [], 0)  
]
```

```
# Construct the noise model from backend properties  
# and custom gate times
```

```
noise_model = noise.device.basic_device_noise_model(properties, gate_times=gate_times)  
print(noise_model)
```

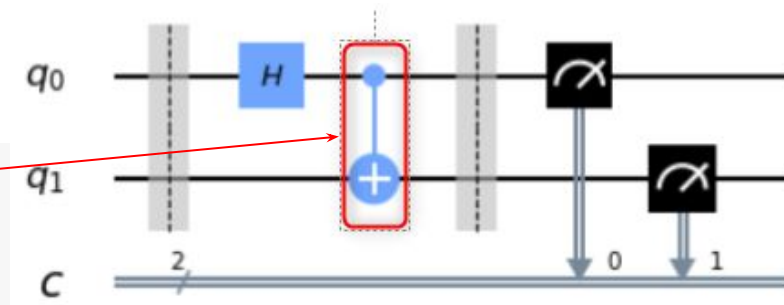
NoiseModel:

Basis gates: ['cx', 'id', 'u2', 'u3']

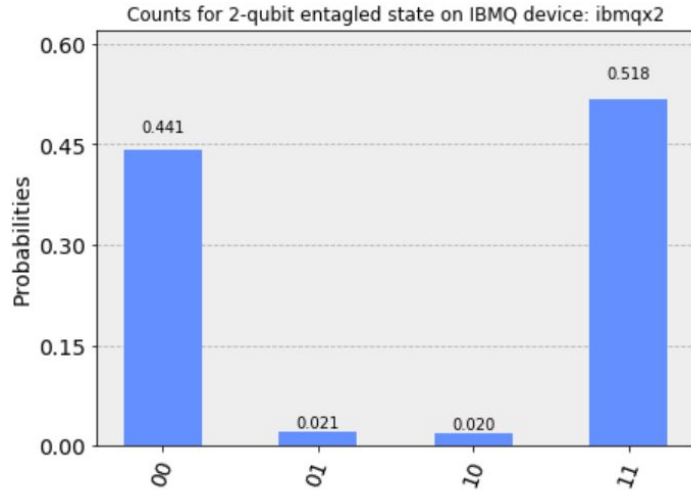
Instructions with noise: ['measure', 'cx', 'id', 'u2', 'u3']

Qubits with noise: [0, 1, 2, 3, 4]

Specific qubit errors: [('id', [0]), ('id', [1]), ('id', [2]), ('id', [3]), ('id', [4]), ('u2', [0]), ('u2', [1]), ('u2', [2]), ('u2', [3]), ('u2', [4]), ('u3', [0]), ('u3', [1]), ('u3', [2]), ('u3', [3]), ('u3', [4]), ('cx', [0, 1]), ('cx', [0, 2]), ('cx', [1, 0]), ('cx', [1, 2]), ('cx', [2, 0]), ('cx', [2, 1]), ('cx', [2, 3]), ('cx', [2, 4]), ('cx', [3, 2]), ('cx', [3, 4]), ('cx', [4, 2]), ('cx', [4, 3]), ('measure', [0]), ('measure', [1]), ('measure', [2]), ('measure', [3]), ('measure', [4])]

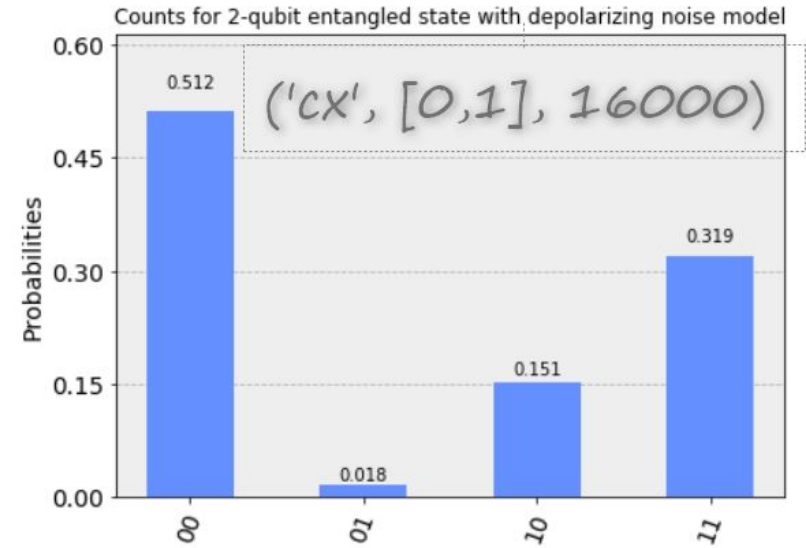
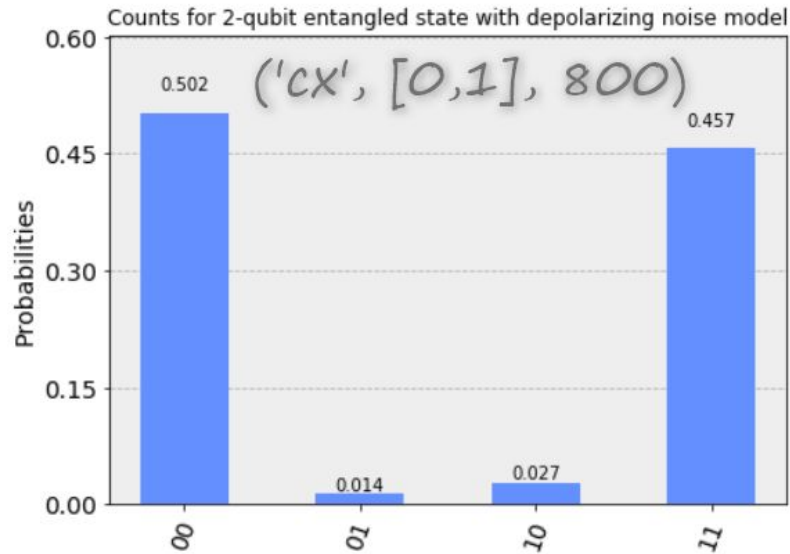


Experimento - Ejecución sobre chip



- Ejecución sobre el chip: ibmq_5_yorktown
- Ejecución que muestra mayoritariamente el entrelazado entre los Qubit 0 y 1

Experimento - Ejemplos de modelo



Conclusiones y líneas futuras

- Ninguna ejecución ni simulación devuelven los mismo resultados exactos
 - La ejecución real es (en general) menos precisa que el modelo simulado
 - Qiskit permite modelar el ruido modificando los tiempos de respuestas de los chips concretos
-
- Los circuitos se transpilan para los chips \Rightarrow Creemos que afecta al modelo
 - Sabemos que el tiempo afecta al resultado \Rightarrow No sabemos en qué medida

