

In [16]:

```
import qiskit as qk
from numpy import pi
from qiskit.tools.monitor import job_monitor
```

In [17]:

```
qk.IBMQ.enable_account('e5eecd2c7f67e3d48ab76560eb52fbe576486d2d8301ff4a304315bf9b7ea9fc6d091bb6d056cde37c3953c70f291dad5ccf1bb54b9567acaec32756e72224b')
```

```
C:\ProgramData\Anaconda3\lib\site-packages\qiskit\providers\ibmq\ibmqprovider.py:251: UserWarning: Credentials are already in use.
  warnings.warn('Credentials are already in use.')
```

In [18]:

```
q_1 = qk.QuantumRegister(2, 'd')
q_2 = qk.QuantumRegister(1, 'm')
c = qk.ClassicalRegister(2, 'c')
circ = qk.QuantumCircuit(q_1, q_2, c)
```

In [19]:

```
circ.cx(q_1[0], q_1[1])
circ.cx(q_1[0], q_2)
circ.cx(q_1[1], q_2)
circ.measure(q_2[0], c[0])
circ.cx(q_1[0], q_1[1])
circ.measure(q_1[0], c[1])
```

Out[19]:

```
<qiskit.circuit.measure.Measure at 0x1d48948a470>
```

In [20]:

```
qk.execute(circ, qk.BasicAer.get_backend('qasm_simulator'), shots=1024).result().get_counts()
```

Out[20]:

```
{'00': 1024}
```

In [21]:

```
real_backend = qk.providers.ibmq.least_busy(qk.IBMQ.backends(simulator=False, operational=True))
```

In [22]:

```
real_backend.name()
```

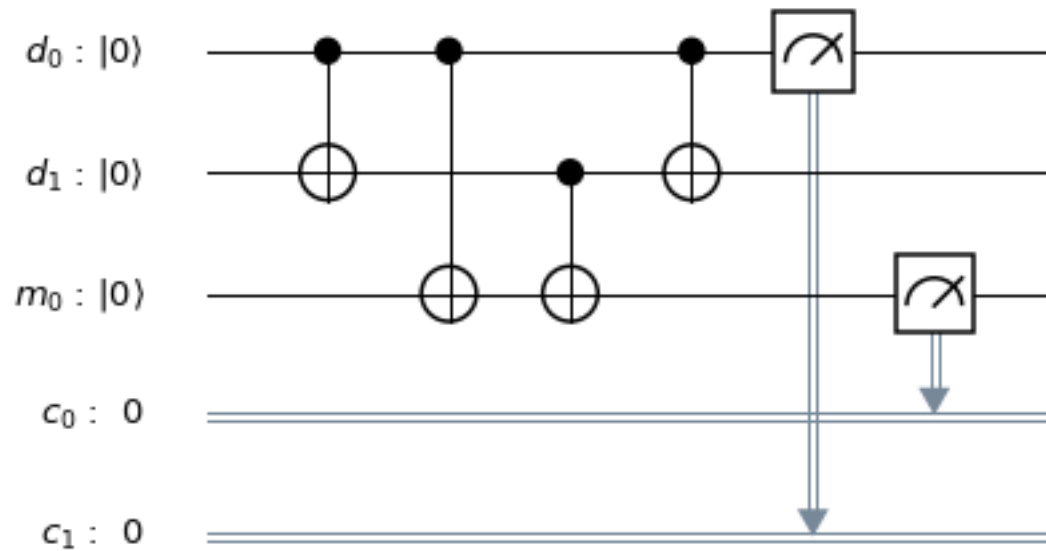
Out[22]:

'ibmqx4'

In [23]:

```
circ.draw(output='mpl')
```

Out[23]:



In [24]:

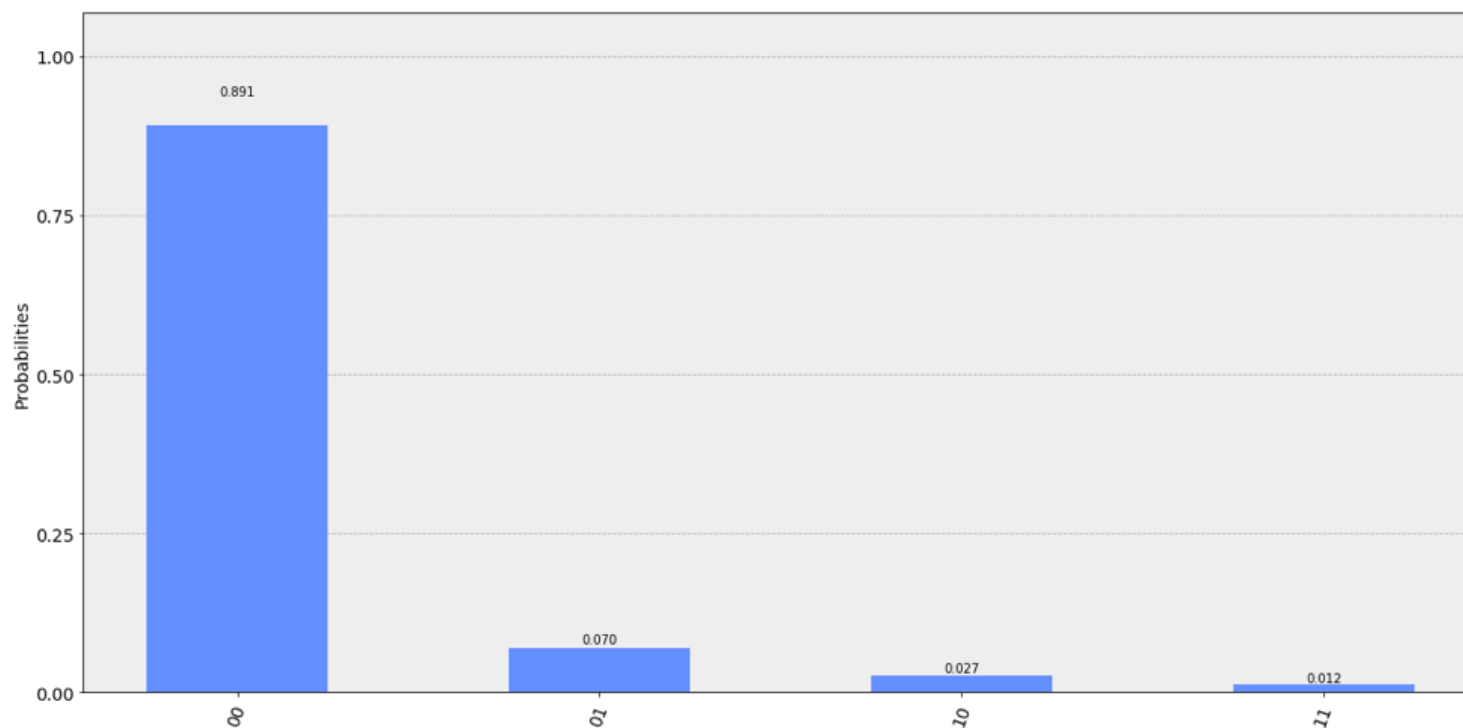
```
job = qk.execute(circ, backend=real_backend, shots=8192)  
job_monitor(job)
```

Got a 500 code response to /api/Jobs/5cc013ec49679f007a76f7b4/status: 500 Error: Failed to establish a backside connection

In [25]:

```
qk.tools.visualization.plot_histogram(job.result().get_counts(), figsize=(20, 10))
```

Out[25]:



In [26]:

```
q_control = qk.QuantumRegister(1, 'q')  
c_control = qk.ClassicalRegister(1, 'c')  
circ_control = qk.QuantumCircuit(q_control, c_control)
```

In [27]:

```
circ_control.measure(q_control, c_control)
```

Out[27]:

```
<qiskit.circuit.instructionset.InstructionSet at 0x1d48a3a1828>
```

In [30]:

```
job = qk.execute(circ_control, backend=real_backend, shots=8192)  
job_monitor(job)
```

In [31]:

```
qk.tools.visualization.plot_histogram(job.result().get_counts(), figsize=(20, 10))
```

Out[31]:

