Analysis of the Mathematics and Implications of Bernstein-Vazirani Algorithm

Abu Bucker Siddik

Shamminuj Aktar

Presentation Outline

- ❑ Definition of Bernstein-Vazirani (BV) problem
- ❑ Classical Solution of BV Problem & Query Complexity
- ❑ Quantum Solution of BV problem
- ❑ Mathematical overview of BV algorithm
- ❑ Implementation of BV algorithm on IBM Quantum Machines
- ❑ Result Analysis & Discussion
- ❑ Conclusion

Bernstein-Vazirani Algorithm

- A quantum algorithm which solves the Bernstein - Vazirani problem
- BV algorithm invented by Ethan Bernstein and Umesh Vazirani in 1992
- BV algorithm allows Quantum Computers to outperform Classical Computers
- An extension of the Deutsch-Josza algorithm (DJ) algorithm

Bernstein-Vazirani Problem

- A black-box function ***f*** with ***n*** bit strings as input and one bit output

$$f : x \in \{0, 1\}^n \rightarrow \{0, 1\}$$

The function *f* is of the form:

$$f(x) = x \cdot a, \text{ where } a \text{ is secret string and } a \in \{0, 1\}^n$$

Problem: Find the *n* bit secret string *a* by querying *f* as few times as possible

The oracle returns f(x) = x · a [dot denotes the inner product modulo 2]

Example: Let, x = 100 and a = 101

So, $x \cdot$ a= (1) (1) + (0) (0) + (0) (1) (mod 2) = 1

Classical Implementation & Query Complexity

Classically we can solve the hidden string problem using n queries.

- Consider bit string $x = 100 \cdots 0$, where $x \in \{0,1\}^n$ and $a \in \{0,1\}^n$
- The oracle will send us back

$$f(x) = x \cdot a = a_1 \text{ , where where } a_1 \text{ is the first bit of } a$$

- Similarly, with bit string $x = 010 \cdots 0$, we will get second bit of a
- Classically for n bit hidden string, it requires n queries
- n classical queries is a lower bound for classical solution

Quantum Solution of BV problem

- BV algorithm can solve the BV problem with just one query for any n bit secret string - a linear speedup!

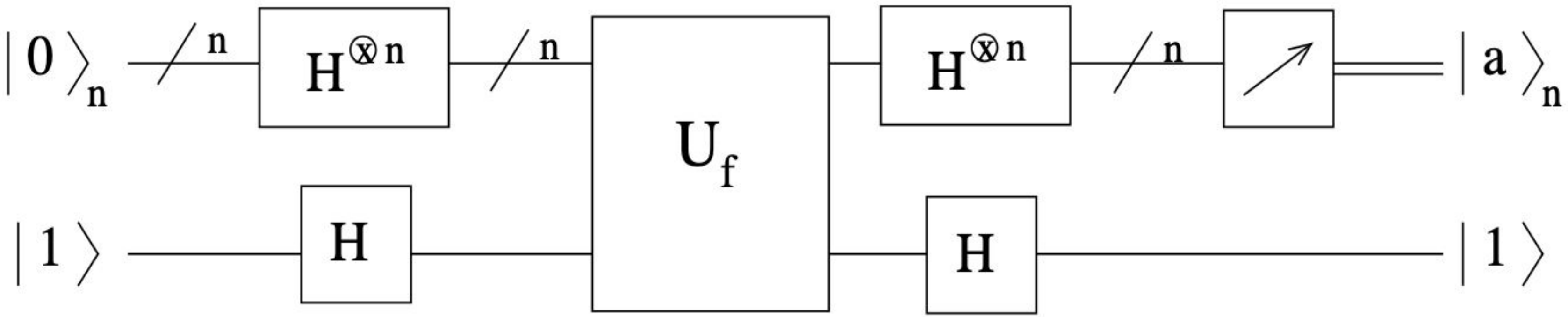

Fig: Quantum circuit for Bernstein-Vazirani algorithm

Source: https://young.physics.ucsc.edu/150/bv.pdf

Mathematical overview of BV algorithm

- First, we apply Walsh-Hadamard transformation to input qubits and Hadamard gate to target qubit :

$$H^{\otimes n}|0\rangle_n \otimes H|1\rangle = \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^{n-1}} |x\rangle_n \otimes \frac{|0\rangle - |1\rangle}{\sqrt{2}}.$$

- Then applying U_f gives us:

$$\frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} |x\rangle_n \otimes \frac{f(x) - f(x)'}{\sqrt{2}}$$

$$\Rightarrow \quad \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^{n-1}} (-1)^{f(x)} |x\rangle_n \otimes \frac{(|0\rangle - |1\rangle)}{\sqrt{2}}.$$

Source: https://young.physics.ucsc.edu/150/bv.pdf

Mathematical overview of BV algorithm

- We then again apply Walsh-Hadamard transformation to first register.
  - Applying Hadamard to one qubit gives us:

$$H|x\rangle_1 = \frac{1}{\sqrt{2}}\left(|0\rangle + (-1)^x|1\rangle\right) = \frac{1}{\sqrt{2}}\sum_{y=0}^{1}(-1)^{xy}|y\rangle\,.$$

  - So, $H^{\otimes n}$ will generate:

$$H^{\otimes n}|x\rangle_n = \sum_{y_{n-1}=0}^{1}\cdots\sum_{y_1=0}^{1}\sum_{y_0=0}^{1}(-1)^{\sum_{j=0}^{n-1} x_j y_j}|y_{n-1}\rangle\cdots|y_1\rangle|y_0\rangle$$

$$= \frac{1}{2^{n/2}}\sum_{y=0}^{2^n-1}(-1)^{x\cdot y}|y\rangle_n$$

Source: https://young.physics.ucsc.edu/150/bv.pdf

Mathematical overview of BV algorithm

- Finally we get,

$$\frac{1}{2^n}\sum_{x=0}^{2^n-1}\sum_{y=0}^{2^n-1}(-1)^{f(x)+x.y}\left|y\right\rangle_n$$

$$\Rightarrow \quad \frac{1}{2^n}\sum_{y=0}^{2^n-1}\left(\sum_{x=0}^{2^n-1}(-1)^{x.a+x.y}\right)\left|y\right\rangle_n \quad \Rightarrow \quad \left|a\right\rangle$$

We can write: $(-1)^{x.a+x.y} = (-1)^{(a\oplus y).x}$

We also know that $\sum_{x=0}^{2^n-1}(-1)^{z.x} = \begin{cases} 2^n, & \text{if } z=0 \\ 0, & \text{otherwise} \end{cases}$

Since we are also applying Hadamard transformation on target qubit, finally we will get,

$$\left|a\right\rangle_n \otimes \left|1\right\rangle$$

Source: https://young.physics.ucsc.edu/150/bv.pdf

Example: Quantum Solution of BV problem

- Let $n = 2$ qubits and a secret string $s = 11$
- The register of two qubits is initialized to zero $|\psi_0\rangle = |00\rangle$
- After applying Hadamard gate, we get: $|\psi_1\rangle = \frac{1}{2}(|00\rangle + |01\rangle + |10\rangle + |11\rangle)$
- For the string $s = 11$, the quantum oracle performs following operation:

$$|x\rangle \xrightarrow{f_s} (-1)^{x\cdot 11}|x\rangle.$$

$$|\psi_2\rangle = \frac{1}{2}\left((-1)^{00\cdot 11}|00\rangle + (-1)^{01\cdot 11}|01\rangle + (-1)^{10\cdot 11}|10\rangle + (-1)^{11\cdot 11}|11\rangle\right)$$

$$|\psi_2\rangle = \frac{1}{2}(|00\rangle - |01\rangle - |10\rangle + |11\rangle)$$

- Applying Hadamard gate to both qubits we get : $|\psi_3\rangle = |11\rangle$

Source: https://qiskit.org/textbook/ch-algorithms/bernstein-vazirani.html

Implementation on IBM Quantum Machines

- ❑ We implemented BV algorithm using Qiskit
- ❑ We used the following IBM Quantum Machines for our analysis:

IBM Machine	Qubit	T1 (µs)	T2 (µs)	Avg. CNOT Error	Avg. Readout Error
IBMQ Toronto	27	104.7	127.88	1.26E-02	4.31E-02
IBMQ Sydney	27	102.67	118.27	9.78E-03	3.63E-02
IBMQ Casablanca	7	86.39	80.94	1.60E-02	1.86E-02
IBMQ Santiago	5	125.07	142.27	6.58E-03	1.53E-02
IBMQ Rome	5	72.31	86.44	1.45E-02	2.45E-02

Source: https://quantum-computing.ibm.com/services?skip=0&systems=all

Implementation on IBM Quantum Machines

- For Hidden String **11011**, we get following circuit of Fig (a)
- Then we run the circuit on IBM QASM Simulator to verify.

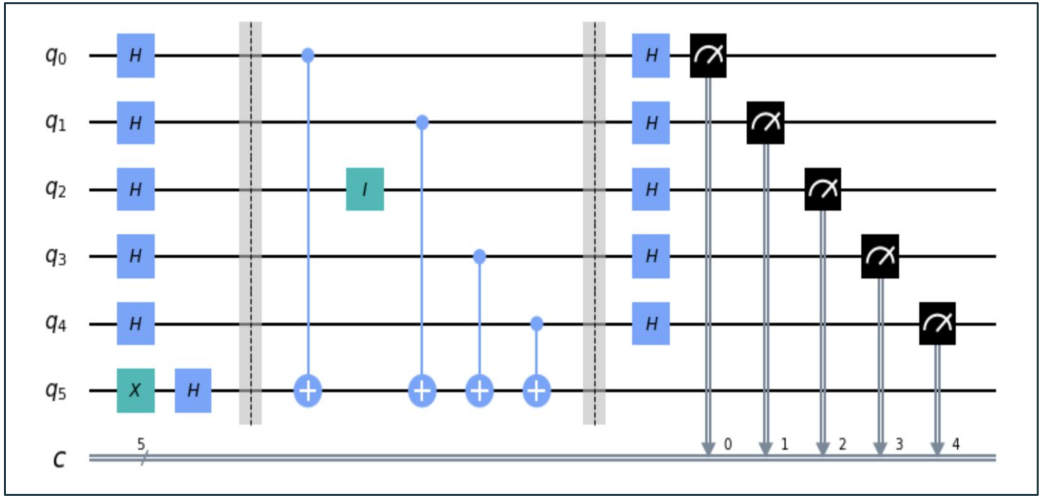

Fig (a) : Qiskit implementation of Bernstein-Vazirani algorithm

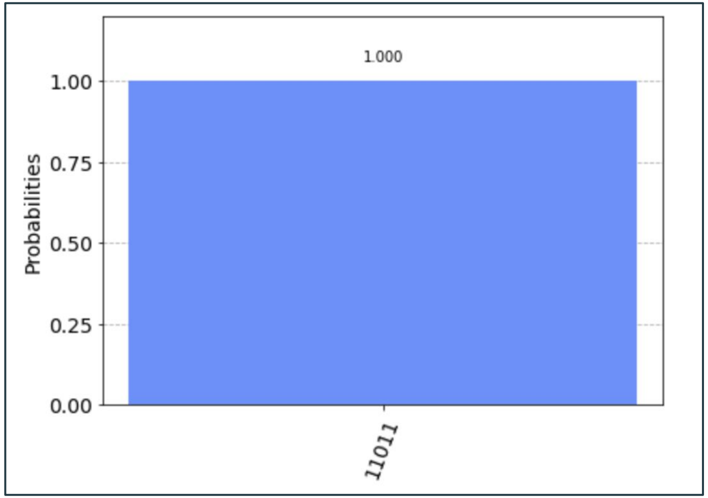

Fig (b) : Measured success rate for string 11011

Result Analysis & Discussion

- ❑ Here, we observed performance of measured success rate when secret string length $l = 3$ & $l = 4$

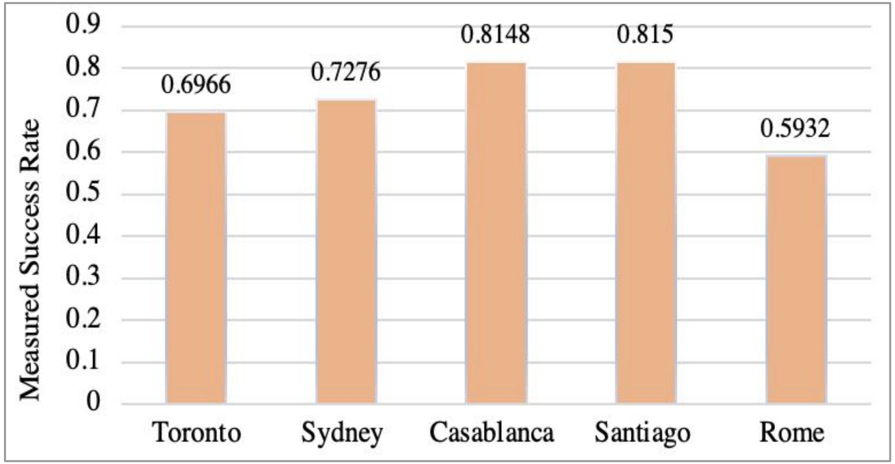

Fig (a) : Measured success rate of different quantum machine when string length l = 3

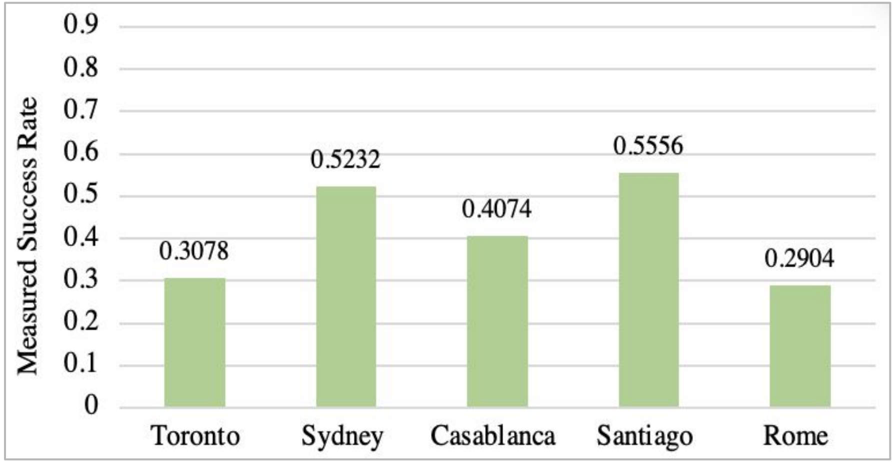

Fig (b) : Measured success rate of different quantum machine when string length l = 4

Result Analysis & Discussion

- ❑ Now, we want to see performance of **IBM Toronto** machine when we query for secret string with length *l = 2, 3, 4, 5, 6, 7*

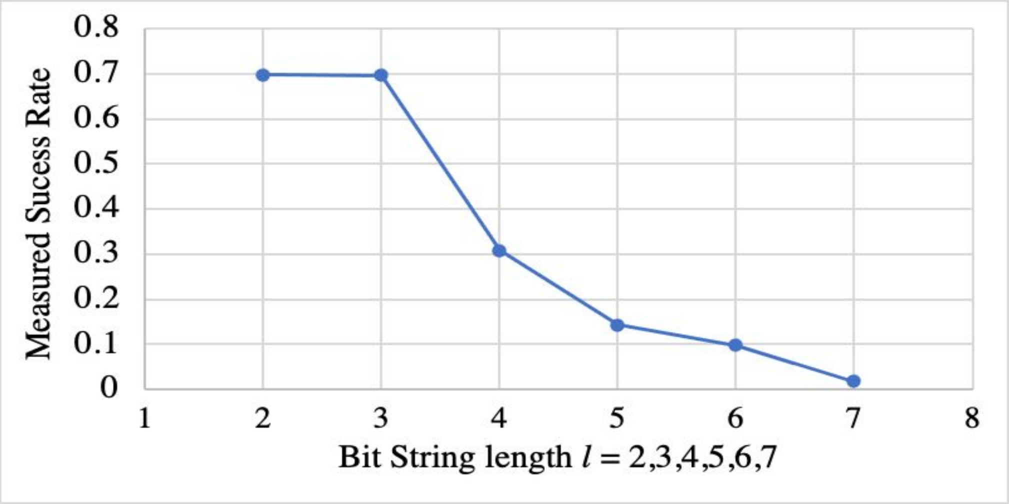

Fig: Measured success rate on IBM Toronto Quantum Processor for secret string with length l = 2, 3, 4, 5, 6, 7

Result Analysis & Discussion

- We ran the circuit on all the quantum machines varying secret string length *l = 2,3,4,5,6 & 7*

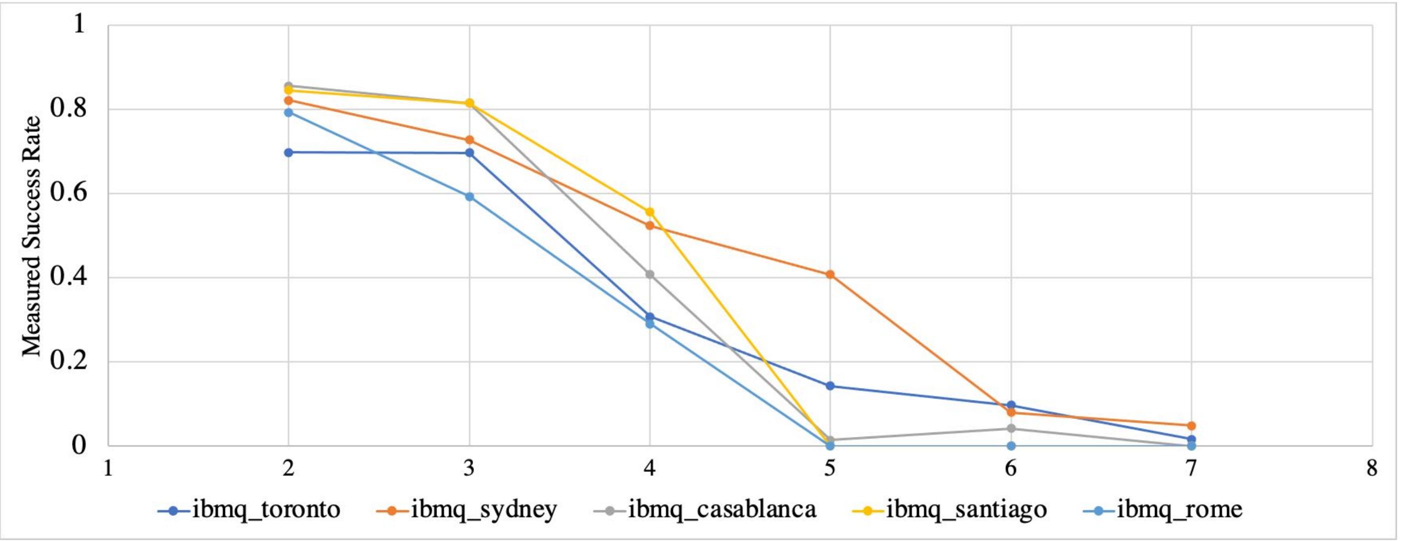

Fig: Measured success rate on different Quantum Processor for secret string with length l = 2,3,4,5,6,7

Conclusion

- Bernstein-Vazirani (BV) algorithm allows to solve hidden string problem with single query
- BV provides linear speedup over classical counterpart
- Implementation of IBM Quantum machine shows
  - Measured success rate decreases with increased bit string length
  - Machine's T1 & T2 time has significant importance on circuit execution
- Recent research showed application of BV algorithm in Quantum multiplication, cryptography etc.