Quantum computing simulations with Intel-QS

Dr. Fabio Baruffa
Senior HPC Application Engineer, Intel IAGS
Quantum computing: key concepts

**Superposition**

Classical Physics

Heads or Tails

Quantum Physics

Heads and Tails

**Entanglement**

N Quantum Bits or **Qubits** = $2^N$ States

1)  
2)  
3)  
4)

**Fragility**

Observation or noise causes loss of information

- 50 Entangled Qubits = more states than any possible supercomputer
- 300 Entangled Qubits = more states than atoms in the universe
- Fragility will require error correction and likely millions of qubits
THE PROMISE OF QUANTUM COMPUTING

Exponential speedup $\leftrightarrow$ surpassing the limits of scaling
How does one create a quantum computing system that takes a quantum algorithm as input and automatically performs a computation on qubits?

A systems perspective of quantum computing: https://doi.org/10.1063/PT.3.4163
Augmenting the Traditional HPC Systems

~50+ Qubits: Proof of concept
  • Computational power exceeds supercomputers
  • Learning test bed for quantum “system”

~1000+ Qubits: Small problems
  • Limited error correction
  • Chemistry, material design
  • Optimization

~1M+ Qubits: Commercial scale
  • Fault tolerant operation
  • Cryptography
  • Machine Learning
**BUILDING QUBITS**

Superconducting Qubits

\[ E_z = \varepsilon_1 - \varepsilon_0 \]

\[ \hbar \omega_0 = \delta \varepsilon_{01} \]

\[ \omega \sim \text{GHz} \]

\[ \mu \text{wave} \]

Spin Qubits in Silicon

Single electron transistor
WHAT CAN WE DO NOW?
SIMULATING QUANTUM ALGORITHMS ON CURRENT SYSTEMS
THE POWER OF QUANTUM COMPUTING: EXPONENTIAL COMPLEXITY

\[ \begin{align*}
1 & \quad |0\rangle, |1\rangle \\
2 & \quad |00\rangle, |01\rangle, |10\rangle, |11\rangle \\
3 & \quad |000\rangle, |001\rangle, |010\rangle, \ldots, |111\rangle \\
10 & \quad |00\ldots\rangle, |00\ldots1\rangle, \ldots, |11\ldots1\rangle \\
20 & \quad |00\ldots\rangle, |00\ldots1\rangle, \ldots, |11\ldots1\rangle \\
30 & \quad |00\ldots\rangle, |00\ldots1\rangle, \ldots, |11\ldots1\rangle \\
40 & \quad |00\ldots\rangle, |00\ldots1\rangle, \ldots, |11\ldots1\rangle \\
\end{align*} \]

\[ \begin{align*}
2 &= 2^1 \\
4 &= 2^2 \\
8 &= 2^3 \\
1k &= 2^{10} \\
1M &= 2^{20} \\
1G &= 2^{30} \\
1T &= 2^{40} \\
\end{align*} \]
Quantum Algorithm

Physically

N input qubit

Sequence physical signals and pulses to manipulate the quantum register

Practically

Probability distribution over the classical states

Sequence of Quantum Gates

Maximizing the probability of the good solution
Quantum computing – speed up

The number of gates times the number of repetitions determines the execution time, not necessarily the size of the problem!
The Intel® Quantum Simulator is a single node or distributed high-performance implementation of a quantum simulator that can simulate general single-qubit gates and two-qubit controlled gates.

It is based on a complete representation of the qubit register state in terms of a distributed vectors.
Distributional Compositional Semantics (DisCo) model \([1,2,3]\)

- NLP algorithms to compute meanings of two sentences and decide if their meanings match
- Incorporates grammatical structure of sentences in a language into the analysis algorithms

Quantum advantage in memory requirement during analysis

- **Example**: word-meaning space of a corpus, if based on 2000 most common words
  - One transitive verb: \(~1\) GB in classical, 33 qubits in quantum
  - 10K transitive verbs: \(~10\) TB in classical, 47 qubits in quantum


**DisCo Model**

- Sentence meaning determined by combining word adjacency in text corpus + known meanings of component (basis) words
- Sentences are represented as tensor products of each individual word (nouns, verbs, etc)
- Convert graphical notation directly to quantum mechanical representation

**Example**

- Sentence structure: noun-verb-noun. "John eats cake", "John likes dogs".

\[
\begin{align*}
\text{John} & \in \mathcal{N} \\
\text{eats} & \in \mathcal{N} \otimes \mathcal{S} \otimes \mathcal{N}
\end{align*}
\]

- Entire meaning space given by the tensor product:

\[
\mathcal{N} \otimes (\mathcal{N} \otimes \mathcal{S} \otimes \mathcal{N}) \otimes \mathcal{N}
\]
“John eats cake, Mary swims in water”

\[ C^{-1} |\text{subject}\rangle \otimes |\text{verb}\rangle \otimes |\text{object}\rangle \]

\[ C^{-1} \left( \sum_i |s_i\rangle \right) \otimes \left( \sum_j |v_j\rangle \right) \otimes \left( \sum_k |o_k\rangle \right) \]

\[
\begin{align*}
|00\rangle & \rightarrow \text{John eats water} \\
|01\rangle & \rightarrow \text{John swims water} \\
|10\rangle & \rightarrow \text{Mary eats water} \\
|11\rangle & \rightarrow \text{Mary swims water}
\end{align*}
\]

\[
\begin{align*}
(\text{John}) & \otimes (\text{eats}) & \otimes (\text{water}) \\
(\text{Mary}) & \otimes (\text{swims}) & \otimes (\text{cake})
\end{align*}
\]

\[ q_0 = \{\text{John, Mary}\} \]

\[ q_1 = \{\text{eats, swims}\} \]

\[ q_2 = \{\text{water, cake}\} \]
RESULTS

![Graph showing results for different measurements.](image)
CONCLUSIONS

• Having a commercial system is not enough, knowledge of new quantum algorithms and applications is required

• Prototyping and testing new Quantum Algorithms on HPC system is beneficial for learning the new field -> Intel Quantum Simulator

• The Natural Language Processing method is a good candidate prototype application to be run on a quantum computer

• We collaborate with different research partners to explore the potential of new algorithms for quantum computing

https://github.com/intel/intel-QS
THANK YOU!