Quantum Operating Systems

Henry Corrigan-Gibbs et al (HotOS workshop 2017)
Presented by Hsuan-Chi Kuo

Background

Problem 1 (Unstructured Search). Given a function $f : \{1, \ldots, N\} \to \{0, 1\}$ find an $x^* \in \{1, \ldots, N\}$ such that $f(x^*) = 1$.

Informal Theorem 2 (Grover [35]). *There is a quantum algorithm for the Unstructured Search Problem that invokes f at most $O(\sqrt{N})$ times and that succeeds with probability at least $2/3$.*

Applications

- **Code fuzzing: Run a piece of code with edge-cases to check if the program misbehaves**
    - **Quantum programs takes only √N inputs to complete N possible inputs with classic programs**
    - **If the program fails on k inputs it only take √(N/k) time to determine what they are**
- Password cracking
    - Quantum systems might kill many cryptographic algorithms.
    - Traditionally, in a brute-force way, attackers spend time linear to the size of the input space, N, cracking the password. On the other hand, it takes time linear to the size of √N on a quantum system.

Applications

- Code fuzzing: Run a piece of code with edge-cases to check if the program misbehaves
  - Quantum programs takes only √N inputs to complete N possible inputs with classic programs
  - If the program fails on k inputs it only take √(N/k) time to determine what they are
- **Password cracking**
  - **Quantum systems might kill many cryptographic algorithms.**
  - **Traditionally, in a brute-force way, attackers spend time linear to the size of the input space, N, cracking the password. On the other hand, it takes time linear to the size of √N on a quantum system.**

3 Questions

- What new abstractions could a quantum OS expose to the programmer?

- How could the power of quantum computers improve the performance of classical software systems?

- What would a distributed system of quantum computers look like?

What new abstractions could a quantum OS expose to the programer?

Qthreads API

- Allows programmers to code as usual just like pthread
- Programmers can create a qthread which does classical compuatation.
- It is faster to answer questions like which qthread returns non-zero value by qthread_join* functions.

How could the power of quantum computers improve the performance of classical software systems?

- **Most Interesting Job First (MIJF) Scheduler**
  - **The classic scheduler checks N tasks in the pool to determine the most interesting task while the quantum one checks only $\sqrt{N}$ tasks**
- Edit distance problem
  - Useful in biology when looking for the gene matches
  - Gather and compare the result with qthread_join_max
- Map Reduce
  - Faster reduce by qthread_join_sum
- Unit Testing
  - Less cases ($\sqrt{N}$) to run to achieve the same coverage of N cases on a classical computer.

Edit distance problem: given two strings, find the minimum number of one-character edits to transform one string to another.

How could the power of quantum computers improve the performance of classical software systems?

- Most Interesting Job First (MIJF) Scheduler
  - The classic scheduler checks N tasks in the pool to determine the most interesting task while the quantum one checks only $\sqrt{N}$ tasks
- **Edit distance problem**
  - **Useful in biology when looking for the gene matches**
  - **Gather and compare the result with qthread_join_max**
- Map Reduce
  - Faster reduce by qthread_join_sum
- Unit Testing
  - Less cases ($\sqrt{N}$) to run to achieve the same coverage of N cases on a classical computer.

Edit distance problem: given two strings, find the minimum number of one-character edits to transform one string to another.

How could the power of quantum computers improve the performance of classical software systems?

- Most Interesting Job First (MIJF) Scheduler
    - The classic scheduler checks N tasks in the pool to determine the most interesting task while the quantum one checks only $\sqrt{N}$ tasks
- Edit distance problem
    - Useful in biology when looking for the gene matches
    - Gather and compare the result with qthread_join_max
- **Map Reduce**
    - **Faster reduce by qthread_join_sum**
- Unit Testing
    - Less cases ($\sqrt{N}$) to run to achieve the same coverage of N cases on a classical computer.

Edit distance problem: given two strings, find the minimum number of one-character edits to transform one string to another.

How could the power of quantum computers improve the performance of classical software systems?

- Most Interesting Job First (MIJF) Scheduler
    - The classic scheduler checks N tasks in the pool to determine the most interesting task while the quantum one checks only $\sqrt{N}$ tasks
- Edit distance problem
    - Useful in biology when looking for the gene matches
    - Gather and compare the result with qthread_join_max
- Map Reduce
    - Faster reduce by qthread_join_sum
- **Unit Testing**
    - **Less cases ($\sqrt{N}$) to run to achieve the same coverage of N cases on a classical computer.**

Edit distance problem: given two strings, find the minimum number of one-character edits to transform one string to another.

What would a distributed system of quantum computers look like?

- Very speculative
- “A quantum phenomenon known as superdense coding allows a quantum client to prefetch data from a quantum server, even if the client has no idea what data it will need in the future.” quoted from the paper.
- “If a client and server share a single entangled qubit (i.e., a preshared qubit), the server can transmit two classical bits of information to the client by sending a single qubit to the client” quoted from the paper.
- Pre-shared(entangled qubit) information allows clients and servers transmit at a double rate.

Takeaway

- It’s speculative.
- Sequential algorithms can reduce its time complexity to $\sqrt{N}$
- Entangled bits can double the transmit rates