



Quantum  
Computing

Robson,  
Matthew

# High Speed Computing with Clustered and Quantum Computers

M. A. Robson<sup>1</sup>

LASACS Department  
Liberal Arts and Science Academy

Period 1  
Independent Study Computer Science

Fall Final, December 2025



# What Is Quantum Computing?

Quantum  
Computing

Robson,  
Matthew

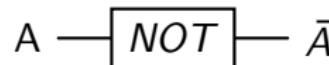
Quantum computing covers many concepts, but it is important to first start with classical computing.

## Classical Information

In a classical computer it stores information as a 1 or a 0 in what is called a bit.

## Some Common Gates

NAND, NOR, AND, OR, XOR → Complex structures such as ALUs and data storage.





# Circuit Complexity

Quantum  
Computing

Robson,  
Matthew

In circuits it is important to cover the complexity required to achieve certain goals. Complexity is classified in Big O Notation.

Additionally, we can classify these further into P, NP, NP-complete, etc.

## Time Complexity Example

A loop of N elements has a time complexity of  $\mathcal{O}(N)$ .

## Theorem 1.0

Any NP complete problem can be solved in polynomial time on a classical computer.



# Turing Completeness

Quantum  
Computing

Robson,  
Matthew

Simply, a Turing-complete computer is one that is able to compute any and Turing-computable function.

## Definition 1.0

Some computational device is taken as Turing-complete or Turing-equivalent if it is probable that it can compute the values for a function for every function of its argument. ie. A common place computer.

While Turing-complete computers are powerful, they begin to break down at the meta level in cases such as the infamous Halting Problem. (see the LASACS lunch time lecture)



# A Single Qubit

Quantum  
Computing

Robson,  
Matthew

A single qubit is given by  $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ .

$\alpha$  is given to be a number in the reals and  $\beta$  is given to be a number in the complex space. Because of this we use a global phase to ensure  $\alpha$  stays real.

## Definition 2.0

A qubit is best described by  $|\psi\rangle = e^{i\gamma}(\alpha|0\rangle + \beta|1\rangle) | \alpha \in \mathbb{R}, \beta \in \mathbb{C}$

This is important to understand in the context of Euler's equation:  $e^{ix} = \cos x + i \sin x$

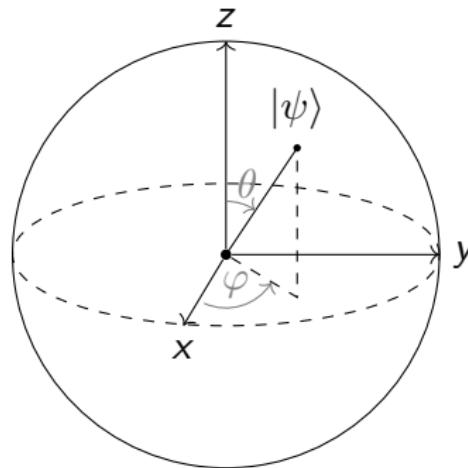


# Quantum Representation

Quantum  
Computing

Robson,  
Matthew

$$|\psi\rangle = e^{i\gamma}(\alpha|0\rangle + \beta|1\rangle) | \alpha \in \mathbb{R}, \beta \in \mathbb{C}$$



$$\alpha = \cos \frac{\theta}{2}$$

$$\beta = e^{i\varphi} \sin \frac{\theta}{2}$$



# Linear Algebra Crash Course

Quantum  
Computing

Robson,  
Matthew

In terms of our qubit we can define some vectors to represent our previous kets for 0 and 1:

$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

Plugging this into our previous formulas we can see:

$$|\psi\rangle = \alpha \begin{pmatrix} 1 \\ 0 \end{pmatrix} + \beta \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} \alpha \\ \beta \end{pmatrix}$$

We rewrite all of this work we have done in terms of vectors because it allows for our later use of matrices to act as the gates and for massively parallelized work.



# More Linear Algebra

Quantum  
Computing

Robson,  
Matthew

Now with the use of these vectors we can perform some algebraic operations such as the tensor product allowing for us to combine gates and qubits.

$$\begin{pmatrix} \alpha & \beta \\ \gamma & \delta \end{pmatrix} \otimes \begin{pmatrix} A & B \\ \Gamma & \Delta \end{pmatrix} = \begin{pmatrix} \alpha \begin{pmatrix} A & B \\ \Gamma & \Delta \end{pmatrix} & \beta \begin{pmatrix} A & B \\ \Gamma & \Delta \end{pmatrix} \\ \gamma \begin{pmatrix} A & B \\ \Gamma & \Delta \end{pmatrix} & \delta \begin{pmatrix} A & B \\ \Gamma & \Delta \end{pmatrix} \end{pmatrix} =$$

$$\begin{pmatrix} \alpha A & \alpha B & \beta A & \beta B \\ \alpha \Gamma & \alpha \Delta & \beta \Gamma & \beta \Delta \\ \gamma A & \gamma B & \delta A & \delta B \\ \gamma \Gamma & \gamma \Delta & \delta \Gamma & \delta \Delta \end{pmatrix}$$



# Quantum Gates

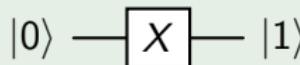
Quantum  
Computing

Robson,  
Matthew

## What are Quantum Gates?

A quantum gate like a classical gate changes the state of a piece of information. Unlike many classical gates, a quantum gate is always reversible.

### Pauli-X Gate



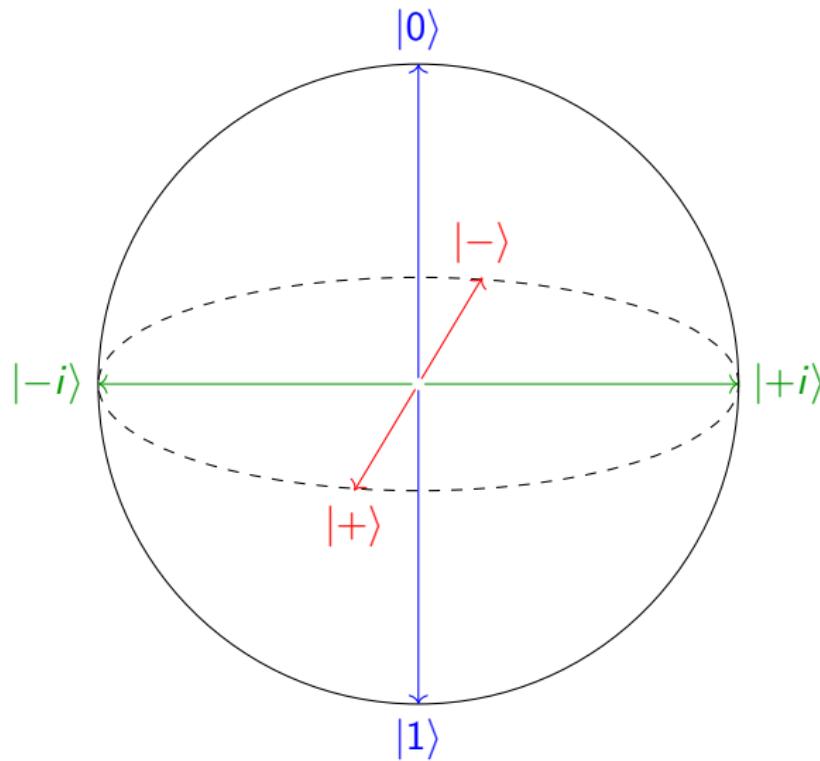
The Pauli-X gate acts in a similar fashion to that of a classical not gate.



# Quantum Measurement

Quantum  
Computing

Robson,  
Matthew





# Change of Basis

Quantum  
Computing

Robson,  
Matthew

It can be seen that the  $|+i\rangle, |-i\rangle, |-\rangle, |+\rangle$  all fall along the equator of the bloch sphere. We can represent these points on the sphere using the following formulas:

## Basis

$$|+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$$

$$|-\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$$

$$|+i\rangle = \frac{1}{\sqrt{2}}(|0\rangle + i|1\rangle)$$

$$|-i\rangle = \frac{1}{\sqrt{2}}(|0\rangle - i|1\rangle)$$

It is trivial to see how this concept can be extended to further to any arbitrary basis.



# More Quantum Gates

Quantum  
Computing

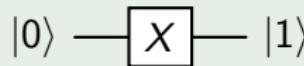
Robson,  
Matthew

As we saw before, we can turn a  $|0\rangle$  into a  $|1\rangle$  using the Pauli X gate.



Following this idea we should be able to perform the same action on both other axis that we saw in the last slide:

## Inversion Gates





# Issue With Multi-Qubit Quantum Gates

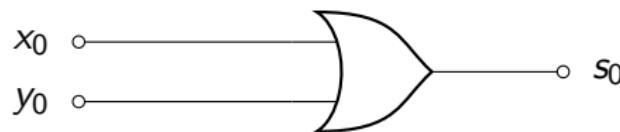
Quantum  
Computing

Robson,  
Matthew

In order to perform actual logic it is important to have qubits that care about multiple qubits in input. The issue is the fact that we must follow the reversibility law.

## Fundamental Law of Quantum Gates

Any gate in a quantum computer must be reversible with some other gate. i.e. Given some output and what the gate does you must always be able to perfectly predict what the input is.





# Multi-Qubit Gates Solved

Quantum  
Computing

Robson,  
Matthew

In order to have a reversible gate we could construct a gate that follows this type of format.

$$\begin{pmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \end{pmatrix} \rightarrow \begin{pmatrix} c_0 \\ c_1 \\ c_3 \\ c_2 \end{pmatrix}$$

We can simply see that this can be constructed as a matrix below.

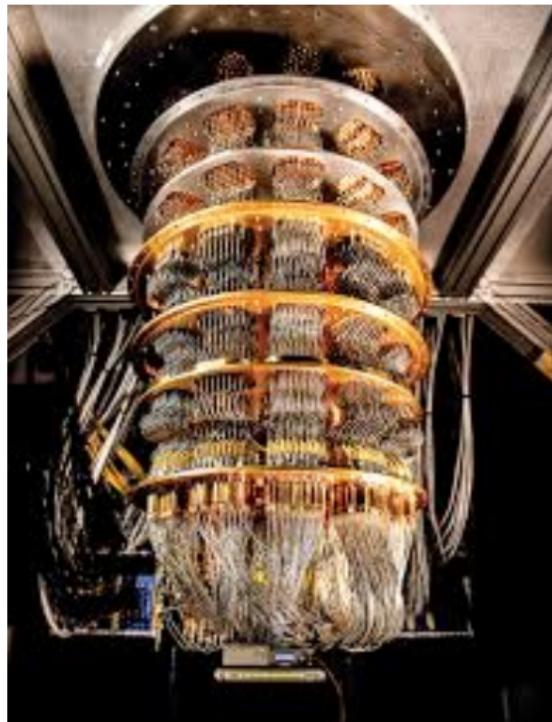
$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \end{pmatrix} = \begin{pmatrix} c_0 \\ 0 \\ 0 \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ c_1 \\ 0 \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ 0 \\ c_2 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ c_3 \\ 0 \end{pmatrix}$$



# The Importance of Quantum Computers

Quantum  
Computing

Robson,  
Matthew



Impacts over \$5 trillion in  
Industry!

- Encryption Cracking Speed-Up
- Molecular Chemical Interaction Simulations
- Materials Logistics



# Grover's Algorithm

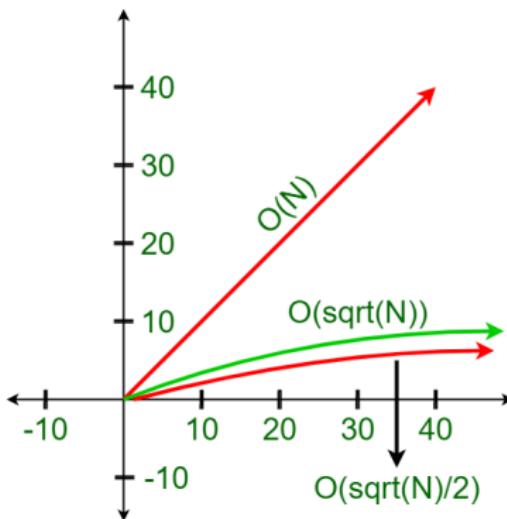
Quantum  
Computing

Robson,  
Matthew

Remember that NP problems are those that can't be solved easily (if you can please tell me) but can be checked to be true or false very quickly.

Grover's Algorithm claims that it can solve one of these problems in  $\mathcal{O}\sqrt{N}$  time using a quantum computer which is much faster than the  $\mathcal{O}(N)$  time using a classical computer.

Running Time Comparison





# Grover's Solution

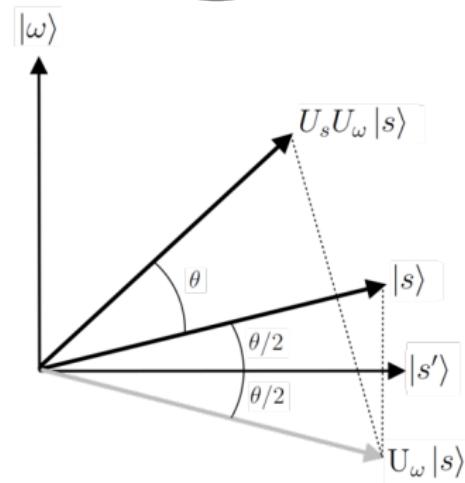
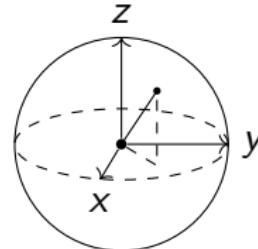
Quantum  
Computing

Robson,  
Matthew

$$|s\rangle = |+\rangle^{\otimes n} = \frac{1}{\sqrt{N}} \sum_{x \in \{0,1\}^n} |x\rangle$$

$$|s\rangle = \frac{1}{\sqrt{N}} |w\rangle + \frac{1}{\sqrt{N}} \sum_{i \neq w} |i\rangle$$

$$|s\rangle = \frac{1}{\sqrt{N}} |w\rangle + \sqrt{\frac{N-1}{N}} |r\rangle$$





# Approximating the Solution

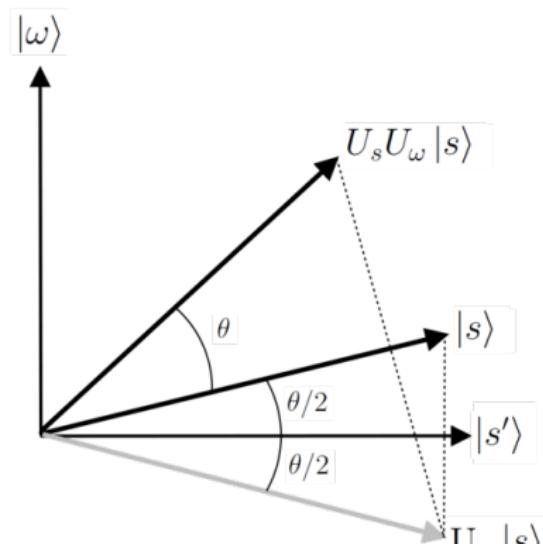
Quantum  
Computing

Robson,  
Matthew

$$(-1)^1 \sin \theta |w\rangle + (-1)^0 \cos \theta |r\rangle$$

$$= -\sin \theta |w\rangle + \cos \theta |r\rangle$$

Repeatedly applying this transformation followed by a flip across the vector  $|s\rangle$  will follow the process shown in diagram to the right where we are able to flip  $\theta$  rads each process.

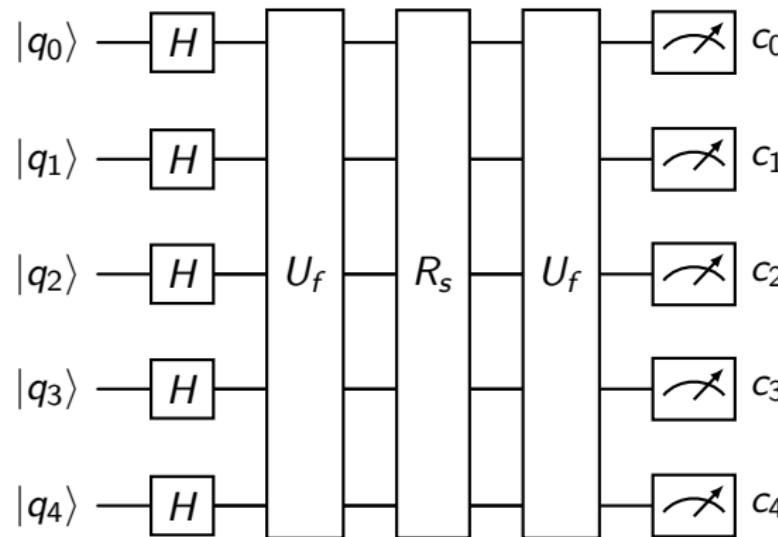




# A Circuit Diagram

Quantum  
Computing

Robson,  
Matthew

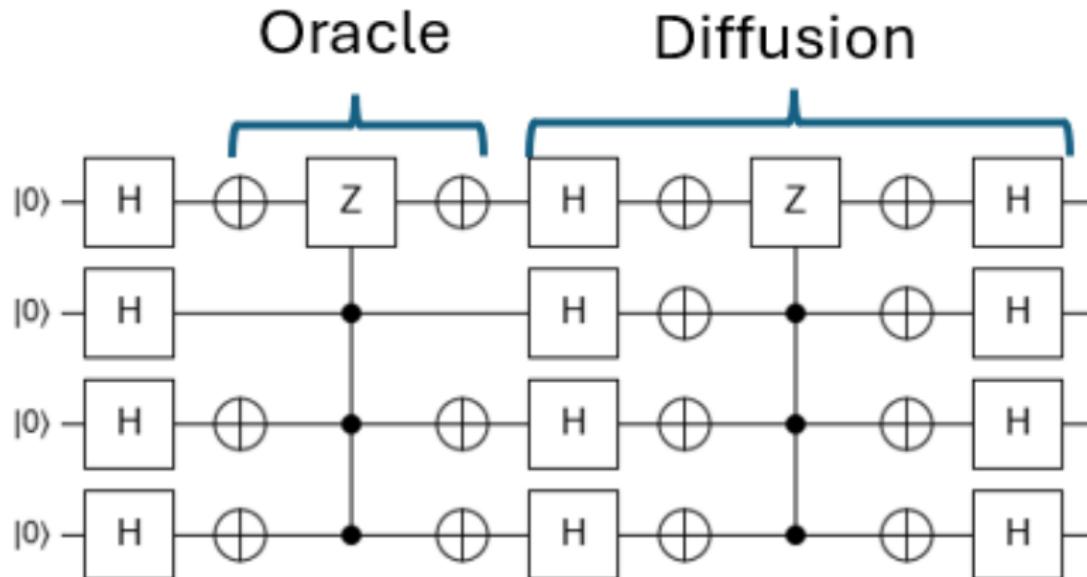




# Expanding The Diagram

Quantum  
Computing

Robson,  
Matthew

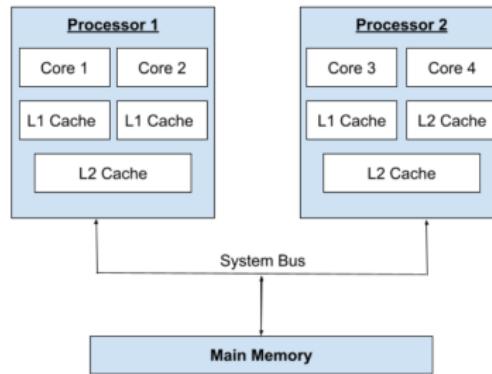




# A Different Approach

Quantum  
Computing

Robson,  
Matthew



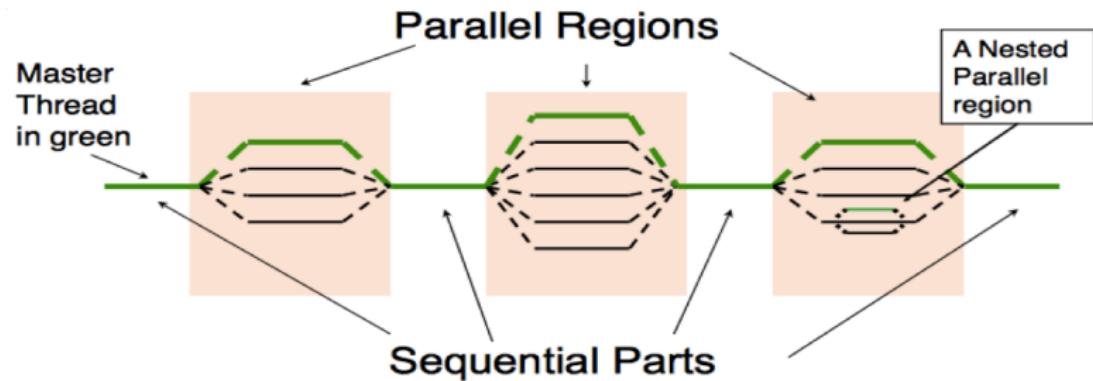
While quantum computing has many benefits and will lead to major speed ups in the future it is important to recognize the technology today that can speed up processes.



# OpenMP

Quantum  
Computing

Robson,  
Matthew



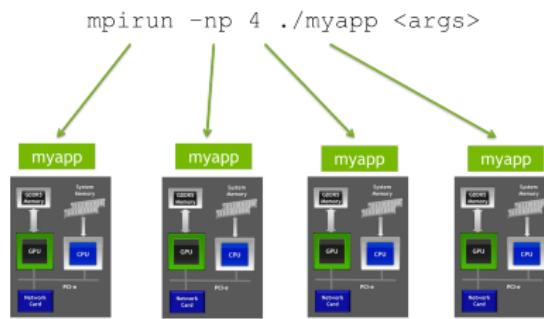
OpenMP is an API that deals with multi-processing code. This is often leveraged in programs where you have to complete a similar action thousands or millions of times (vector multiplication for example).



# OpenMPI

Quantum  
Computing

Robson,  
Matthew



OpenMPI opens the world of multi-processing computing to cover multiple computers. OpenMPI allows for applications to scale to hundreds or even thousands of compute nodes.



# Example OpenMP Code

Quantum  
Computing

Robson,  
Matthew

```
#include <stdio.h>
#include <omp.h>

#define N 100000

int main() {
    int A[N], B[N], C[N];
    #pragma omp parallel for
    for (int i = 0; i < N; i++) {
        C[i] = A[i] + B[i];
    }
    printf("Vector addition complete.\n");

    return 0;
}
```



# Example OpenMPI Code

Quantum  
Computing

Robson,  
Matthew

```
#include <stdio.h>
#include <mpi.h>

int main(int argc, char *argv[]) {
    int rank, size;
    MPI_Init(&argc, &argv);

    MPI_Comm_rank(MPI_COMM_WORLD, &rank);
    MPI_Comm_size(MPI_COMM_WORLD, &size);

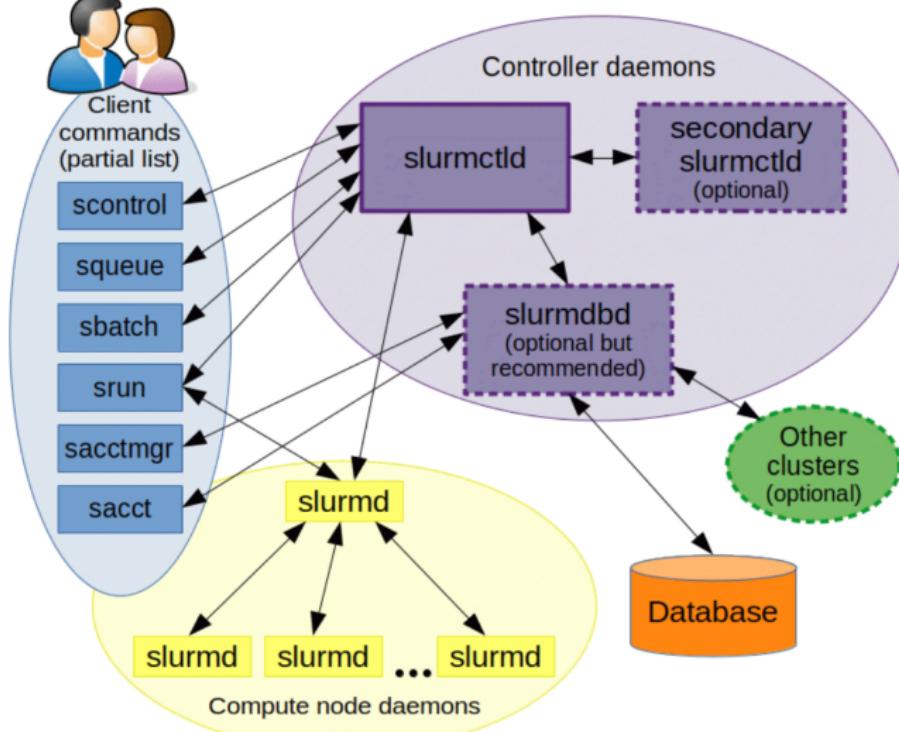
    printf("Hello from process rank %d
           of total %d processes.\n", rank, size);
    MPI_Finalize();
    return 0;
}
```



# Implementing a Clustered Computer With Slurm

Quantum  
Computing

Robson,  
Matthew





# Quantum Computing

Robson,  
Matthew