Seminar course Quantum Software Systems

(aka "qc-systems-seminar")
Kick-off meeting
https://dse.in.tum.de/

Francisco Romão Emmanouil (Manos) Giortamis Prof. Pramod Bhatotia



Course instructors



Chair of Distributed Systems & Operating Systems

https://dse.in.tum.de/team/







Manos Giortamis

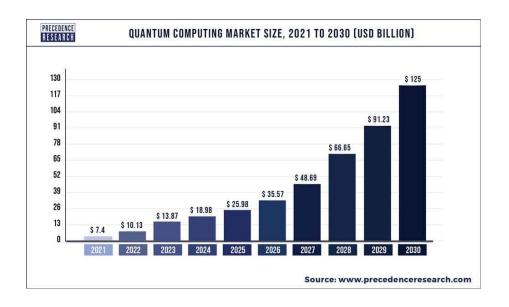


Prof. Pramod Bhatotia

Motivation for quantum computing (QC)



- Quantum computers will be the world's fastest computing devices
 - They can solve problems intractable for classical computers
- QC is still at an infant stage
 - Many open problems and opportunities for research exist
 - Exciting field for exploration and discovery

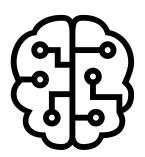


Applications of QC





Chemistry



ΑI



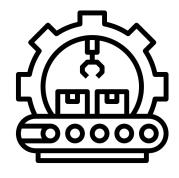
Pharmaceuticals



Cybersecurity



Finance



Manufacturing

Tech giants + startups adopt QC





















Quantum vs classical computing

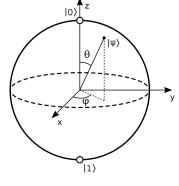


	Classical	Quantum
Bit	0 or 1	Superposition of o and 1
Hardware	>10 ⁹ of bits, "perfect"	10 ² -10 ³ of qubits, noisy
Programming	High-level	Qubit/Gate level
Determinism	Yes	Inherently probabilistic

Qubits



- Quantum devices perform computations on qubits
- Qubit: A two-state quantum-mechanical system
- State of a qubit: Represented as a two dimensional complex vector



$$|\psi\rangle = a_0|0\rangle + a_1|1\rangle$$

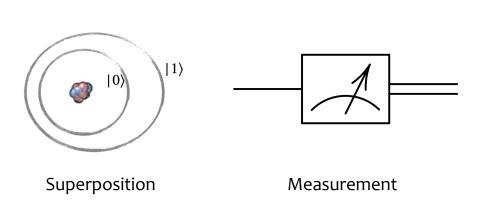
$$a_0, a_1 \in \mathbb{C}$$

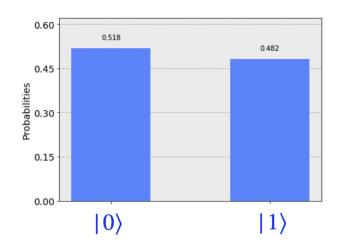
$$\sum_{k=0}^{n-1} |a_k|^2 = 1$$

Superposition and measurement



- Superposition: Qubit in state 'between' $|0\rangle$ and $|1\rangle$
- State of a qubit in superposition cannot be observed/copied
- After measuring a qubit, it decoheres to either $|0\rangle$ or $|1\rangle$





Quantum operations



- Quantum Operator = Gate
- Gates are reversible

Unary Gates

(e.g. Hadamard Gate)

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

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

Binary Gates (e.g. CNOT Gate)

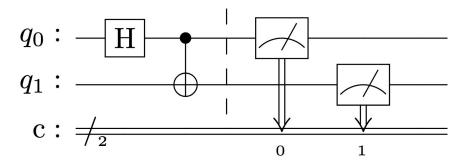
$$|+\rangle$$

$$\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$$
 $|0\rangle$

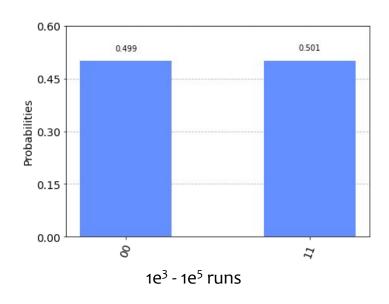
$$CNOT = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

Quantum circuits





Quantum circuit with 2 qubits, 2 gates and 2 measurements



Quantum Computers

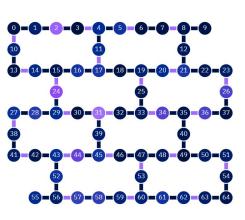




Quantum computer



Quantum chip

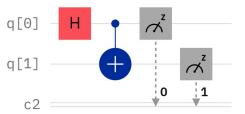


Qubit topology

Programming quantum computers



```
# Importing standard Qiskit libraries
from qiskit import QuantumCircuit, transpile, IBMQ
from qiskit.visualization import plot_histogram
provider = IBMQ.load account()
backend = provider.backend.ibm oslo
gc = QuantumCircuit(2)
qc.h(0)
qc.cx(0,1)
qc.measure all()
qc = transpile(qc, backend)
job = backend.run(qc)
counts = job.result().get counts()
plot histogram(counts)
```



Visualised circuit

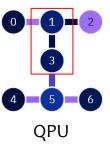
```
1   OPENQASM 2.0;
2   include "qelib1.inc";
3
4   qreg q[2];
5   creg c[2];
6
7   h q[0];
8   cx q[0], q[1];
9   measure q[0] -> c[0];
10  measure q[1] -> c[1];
```

Assembly

Programming quantum computers



```
# Importing standard Qiskit libraries
from qiskit import QuantumCircuit, transpile, IBMQ
from qiskit.visualization import plot_histogram
provider = IBMQ.load account()
backend = provider.backend.ibm oslo
gc = QuantumCircuit(2)
qc.h(0)
qc.cx(0,1)
qc.measure all()
qc = transpile(qc, backend)
job = backend.run(qc)
counts = job.result().get counts()
plot histogram(counts)
```



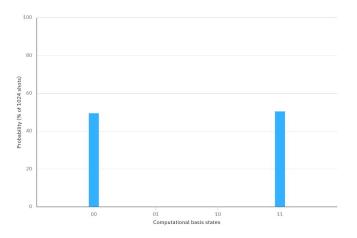
```
OPENQASM 2.0;
include "qelib1.inc";
qreg q[7];
creg meas[2];
rz(pi/2) q[0];
sx q[0];
rz(pi/2) q[0];
cx q[0],q[1];
barrier q[0],q[1];
measure q[0] -> meas[0];
measure q[1] -> meas[1];
```

New assembly

Programming quantum computers



```
# Importing standard Qiskit libraries
from qiskit import QuantumCircuit, transpile, IBMQ
from qiskit.visualization import plot_histogram
provider = IBMQ.load account()
backend = provider.backend.ibm oslo
gc = QuantumCircuit(2)
qc.h(0)
qc.cx(0,1)
qc.measure_all()
qc = transpile(qc, backend)
job = backend.run(qc)
counts = job.result().get_counts()
plot histogram(counts)
```



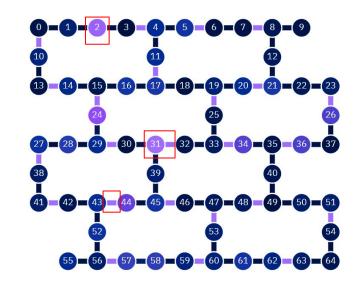
Results

Current state: NISQ era



Noisy Intermediate-Scale Quantum (NISQ) era

- Noisy hardware:
 - Prone to environmental noise
 - Prone to decoherence errors and cross-talk noise
 - Limited error mitigation/correction

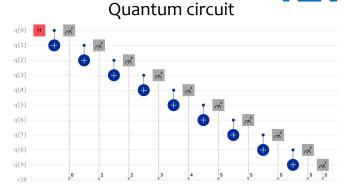


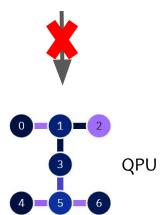
Purple nodes and links have higher noise!

Current state: NISQ era



- Intermediate-Scale:
 - Currently up to a few 100s of qubits
 - 10.000s needed for quantum advantage
 - Low quantum-volume
 - Limited qubit connectivity





Research questions



Existing QC hardware is limited in terms of quantity and quality

Which are the research challenges for software systems in quantum computing?

Tentative topics



Papers from top conferences (e.g., ASPLOS, HPCA, MICRO, PLDI)

Tentative topics

#1: Transpilation (qubit mapping)

#2: Quantum resource management

#3: Circuit cutting & knitting

#4: Circuit multiprogramming

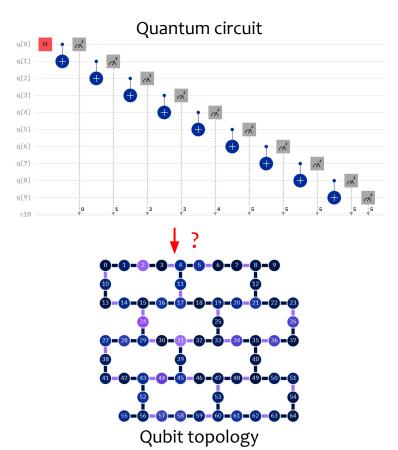
#5: Circuit transformations

Challenge #1: Transpilation (qubit mapping)



Transpilation modifies a given circuit to match the topology of a specific quantum backend

- How can we optimally map logical qubits to physical qubits?
 - Avoid noisy qubits
 - Avoid noisy qubit links
 - Minimise SWAP operations
- How can we do it fast?
 - NP-hard problem
 - Greedy approaches
 - Heuristics

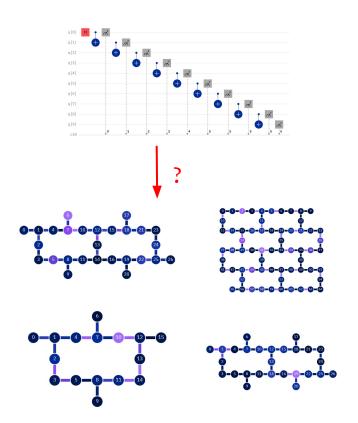


Challenge #2: Quantum resource management



Selecting the best machine for a given quantum circuit is challenging

- Which machines does the circuit fit into?
- Which machines support the circuit's operations?
- Which topology best fits our circuit?
- Which machine has the best noise properties?

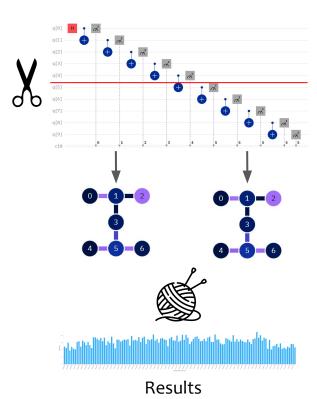


Challenge #3: Circuit cutting and knitting



Circuit cutting and knitting is a method of dividing a quantum circuit into smaller fragments, executing them and merging the results back

- Which are the optimal cut locations?
- What is the additional quantum cost?
- What is the additional classical cost?
- How can we mitigate the (exponential) costs?

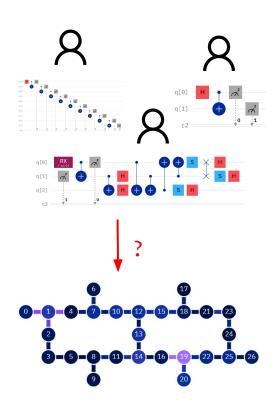


Challenge #4: Circuit multiprogramming



Multiprogramming enables multiple circuits to be executed on the same QPU in parallel

- How can we optimally map multiple circuits on a QPU?
 - Equally good partitions
- How can we minimise circuit interference?
 - Crosstalk noise
 - Measurement interference
 - Unequal depths

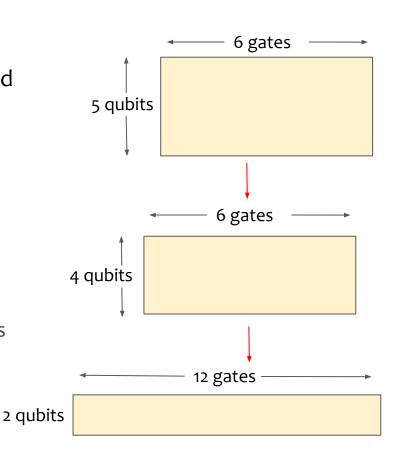


Challenge #5: Circuit transformations



There are circuit transformations that aim to change a circuit in a way that its fidelity is improved

- Circuit compaction
 - Width (# of qubits) reduced
 - Depth (# of consecutive gates) increased
- How to use it optimally?
 - Tradeoff between allocating less qubits but adding intermediate measurement operations
 - Possible decoherence errors



Format

Bird's eyes view





Team (2 students per team)



Research papers
(Top systems conferences)



Understand



Research ideas



1 presentation



1 short report



Peer-reviewing

Overview



Phase I

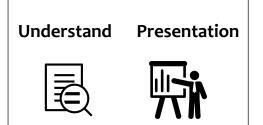
Phase II: Understand & explore

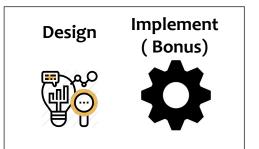
Phase III: Research

Phase IV: Report & review

Kick-off









Phase I: Kick-off meeting





Format and motivation (all participants meeting)



Team formation (2 students per team)



Paper selection (Top systems conferences)

The first week

NOTE

- 1. A list of papers will be provided for FCFS bidding
- 2. Paper presentation guidelines will be provided for the next phase

Phase II: Understand & explore





Understand the paper(s)

Focus

- Understand the paper and related work
- 2. Also **explore** a "laundry list" of research ideas/directions



Paper presentation

Focus

- Explain the work/related work ("why?" and "how?")
- 2. Explain and discuss all possible research directions
- 3. Pick a research direction

Phase III: Research





Research work

Focus:

Indepth research work to nail-down the problem and detailed approach to solve it!



Research prototype

Bonus:

(Optional)

"Build the system to solve it!" and show us the working idea and associated results

Phase IV: Report & review









Prepare a single "short & sweet" report summarizing

- (a) Paper
- (b) Research work



Peer-review

Focus

Give constructive (positive and critical) feedback for

- (a) Paper summary
- (b) Research work



Overall timeline



Phase I Phase II: Understand & explore Phase III: Research

Phase IV: Report & review

Kick-off



Understand Presentation



Design

Implement (optional)



Report

Peer-review





Milestone #1: Team formation & paper selection

Milestone #2: Paper

presentations

Milestone #3: Research work complete

Milestone #4: Report submission

Milestone #5: Peer-reviewing

1 week

3 weeks

2 weeks

3 weeks

1 week

2 weeks

Meeting

Meeting

Organization



- Format
 - Team-based seminar course (2 students per team)
- Communication
 - Slack for announcements and information sharing
 - Hotcrp for report submission and peer-reviewing
- Meetings (in-person, attendance is compulsory)
 - **Meeting #1:** Kick-off
 - **Meeting #2:** Paper presentation

Learning goals



- Learn about the cutting-edge research in quantum computing systems
- Promote critical thinking
- Cultivate an environment for innovation
 - To push the boundaries by advancing the state-of-the-art
- Improve scientific skills
 - Presentation
 - Writing
 - Communication: discussion and arguing
 - Mentorship: giving feedback and moderating discussion
- Encourage system building and evaluation
 - Learn by building, breaking, and benchmarking systems
- Importantly, to have fun!

Code of conduct



University plagiarism policy

https://www.in.tum.de/en/current-students/administrative-matters/student-code-of-conduct/

Decorum

- Promote freedom of thoughts and open exchange of ideas
- Cultivate dignity, understanding and mutual respect, and embrace diversity
- Racism and bullying will not be tolerated

Contact



- Francisco Romão
 - <u>francisco.romao@tum.de</u>
- Manos Giortamis
 - <u>emmanouil.giortamis@in.tum.de</u>
- Prof. Pramod Bhatotia
 - <u>pramod.bhatotia@in.tum.de</u>
- All seminar-related info: https://github.com/TUM-DSE/seminars



Communication:

Join us with TUM email address (@tum.de) <u>ls1-courses-tum.slack.com</u> #ws-24-qc-systems