Distributed Quantum Computing across Heterogeneous Hardware with Hybrid Dependency Hypergraphs

Maria Gragera Garces, Chris Heunen, and Mahesh K. Marina

Overview

Distributing quantum computations across heterogeneous devices introduces communication and coordination costs that depend on both quantum and classical operations.
We present a novel abstraction called Hybrid Dependency Hypergraphs (HDHs) to model space, time, and type dependencies in distributed quantum execution.
This poster explores how HDHs support network-level reasoning about communication patterns, and how they can expose cost trade-offs across various quantum computational models.

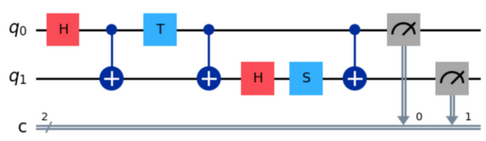
1 The Distributed Quantum Computing Mapping problem

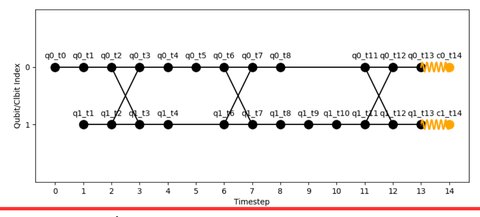
Quantum workload
Circuit/ MBQC pattern/ Quantum walk/ Hybrid computation /...

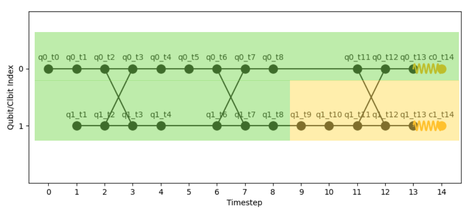
Intermediate abstraction
A representation that captures the dependencies of data through the computation.

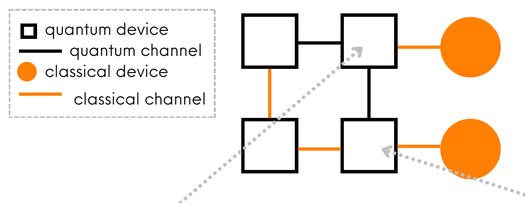
Partitioning
The representation is cut into sub-workloads that may be computed sequentially or in parallel in a network of quantum and classical devices.

Mapping to a hybrid network
Sub-workloads are mapped to a network of collaborating quantum and classical devices. Depending on the type of available channel (classical or quantum), communication primitives are used to connect the partitions..

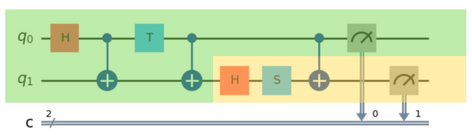




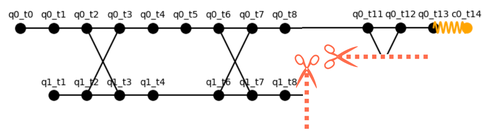




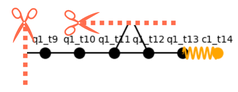
Can be mapped based on:
- device capacity
- device type
- interconnectivity between devices



Partition 1



Partition 2

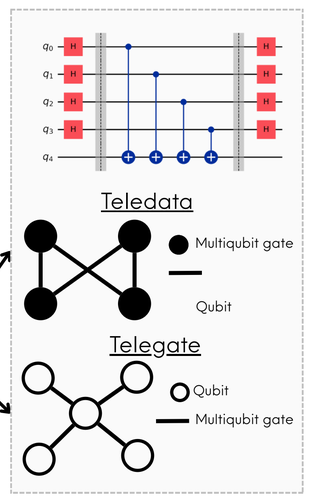


2 What is a HDH?

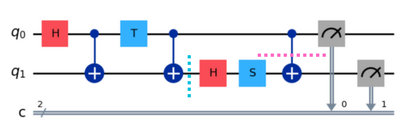
Directed hypergraph that encodes the temporal and spatial dependencies created between quantum states during a quantum or hybrid classical-quantum computation. HDH encodes this information through the following objects:

Symbol	Meaning
●	Classical node
●	Quantum node
〰	Classical hyperedge
—	Quantum hyperedge
◌	Predicted node
- -	Predicted hyperedge

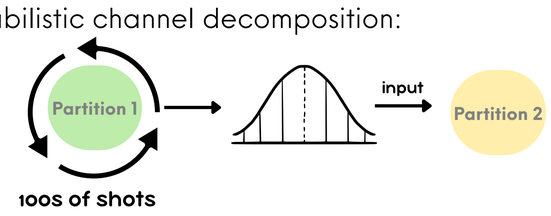
Other abstractions



Under the presence of a quantum channel:
- Gate cuts = non local gates
- Wire cuts = teleportation protocol



If no quantum channel is available, then we can simulate its presence through probabilistic channel decomposition:



While this method is useful for today's technology, it is exponentially costly.

3 Why do we need a unifying abstraction?

1) Existing abstractions (e.g., teledata, telegate) are not designed to be general-purpose constructs
- Due to their limited scope, they do not consider all partitioning options and so do not allow assessing different partitioning technique.

2) Existing abstractions only support the quantum circuit model of computation.

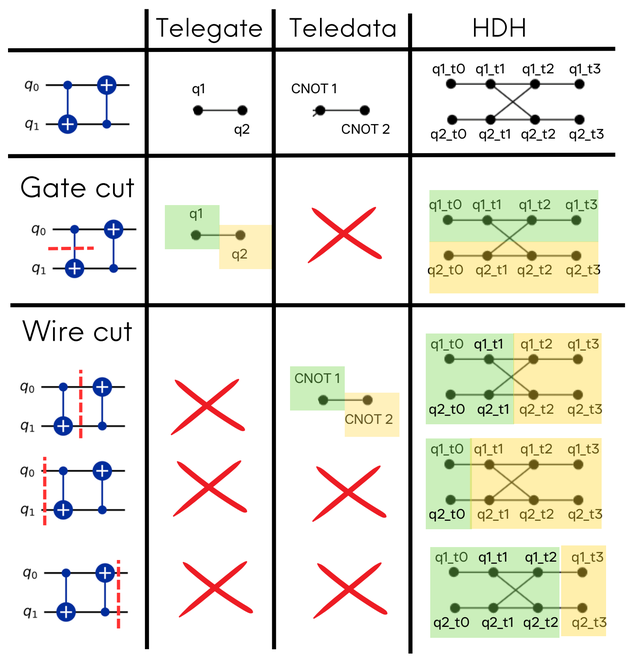
3) No consideration of classical data, which has two undesirable consequences:
1) Hybrid computations cannot be represented
2) Natural cutting points arising from classical operations such as mid-circuit measurement are not considered

4 Representing different quantum computation models with HDHs

One can think of HDHs as a hypergraph representing the state transformations through a quantum computation. This perspective enables the creation of mappings that translate operations from different computational models to HDH motifs.

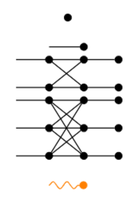
Here are two examples of popular model mappings:

HDHs superseed previous abstractions by representing all possible available partitions:



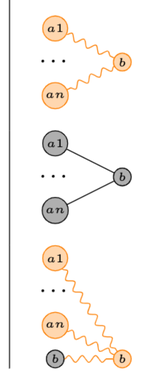
Quantum circuits

Operation	Motif
Qubit initialization	
Single qubit gate	
Two qubit gate	
Three qubit gate	
Measurement	

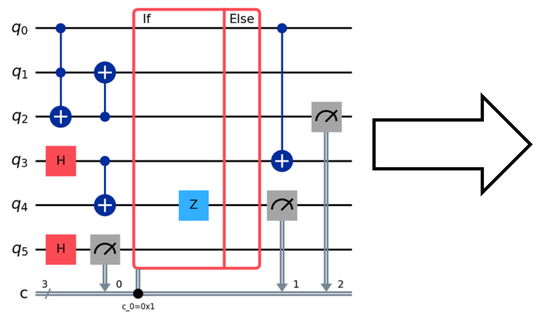


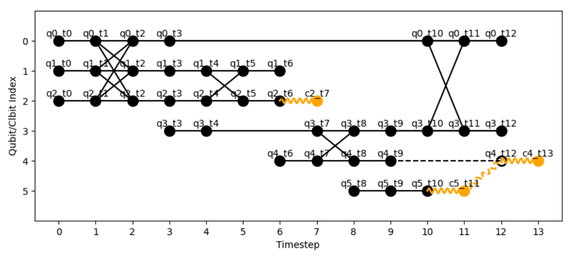
Measurement Based Quantum computing

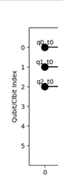
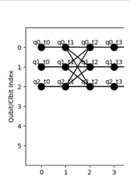
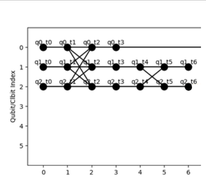
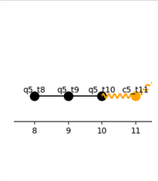
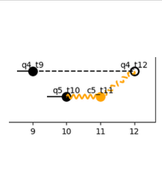
Operation	Motif
$N_b^{a_1,\ldots,a_n}$ or $C_b^{a_1,\ldots,a_n}$	
$E_{a_1,\ldots,a_n,b}$	
$M_b^{a_1,\ldots,a_n}$	



5 An example of a HDH construction





Subsection of circuit					
Resulting HDH					

6 Evaluation of HDH partitions

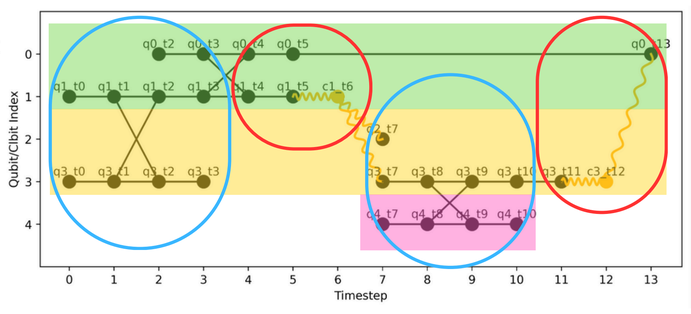
Partitions of HDHs can be evaluated through 3 main metrics:
- **Partition size**: the number of qubits or bits/cbits required to perform the sub-computation
- **Communication cost**: the number of hyperedges (classical or quantum) across the cut(s)
- **Parallelism,** defined as $\sum_{t \in T} \min(|L \cap \rho_t|, |R \cap \rho_t|)$ given a set of L and R partitions and p operations across t timesteps . This means that the parallelism created by a cut is the minimum amount of work performed by any two concurrently partitions.

In the setting above for instance, parallelism contributions would follow:

Parallelism = 0 monopartition

Parallelism = 1 at each of these timesteps as each partition is doing 1 unit of work (state trasnformation)

Total partition parallelism = 1 + 1 + 1 + 1 + 0 + 0 + 1 + 1 + 1 + 1 + 0 + 0 + 0 = 8



7 Open source library

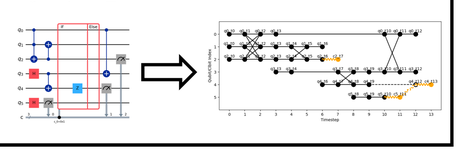
```
pip install hdh
```

```
import hdh
from hdh.models.circuit import Circuit
from hdh.visualize import plot_hdh

circuit = Circuit()

# Set of instructions
circuit.add_instruction("ccx", [0, 1, 2])
circuit.add_instruction("h", [3])
circuit.add_instruction("h", [5])
circuit.add_instruction("cx", [3, 4])
circuit.add_instruction("cx", [2, 1])
circuit.add_conditional_gate(5, 4, "z")
circuit.add_instruction("cx", [0, 3])
circuit.add_instruction("measure", [2])
circuit.add_instruction("measure", [4])

hdh = circuit.build_hdh() # Generate HDH
fig = plot_hdh(hdh) # Visualize HDH
```



SUPPORTED BY UNITARY FOUNDATION