Internship topic

Optimization from gate-based quantum circuits to ordered hardware implementation

Context

As interest in quantum algorithms grows, along with the expansion of quantum hardware, implementing quantum circuits on actual quantum devices has become increasingly important. The topic of this internship focuses on an implementation known as *transpilation*, which involves developping classical optimization methods for both the expression of quantum circuits and their mapping to a specific quantum chip.

Qubits and circuits The fundamental unit of quantum information is called a *qubit*. Qubits can be implemented in quantum chips using various physical supports such as Josephson junction (used by superconductvity-based chips), trapped ions, or photons, to name a few. Computation operations are expressed using *gates* that operate on one or multiple qubits. Algorithms are called *circuits*.

While circuits can be designed by applying gates between any set of qubits, qubits need to be *physically* interconnected for a gate to have an action on them. For instance, a *controlled gate* acts on two qubits: the first one is used as a control, and the second one is modified depending on the operation and the value of the control qubit. Qubits are interconnected on the quantum chip by a given topology: two examples are given Figures 1 and 2. The circuit is modified by adding SWAP operations to swap qubits and produce a circuit whose interconnection graph is isomorphic to the topology of the chip.

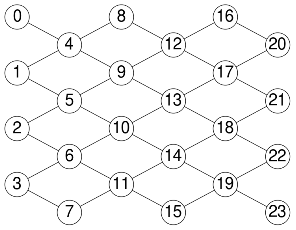

Figure 1: Topology of Yamaska quantum chip (Monarq).

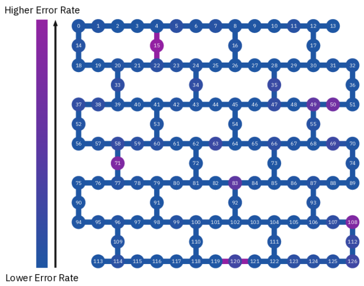

Figure 2: Topology of IBM Eagle quantum chip.

Sharing a quantum machine Nowaday's quantum chips feature a large number of qubits: IBM's Eagle chip has 127 qubits, Heron has 156 qubits, and Condor has 1 121 qubits. Anyon's Yamaska chip, on Calcul Québec's MonarQ machine, has 24 qubits. In December 2023, IBM anounced a 100 000-qubit system by 2033. Not all the circuits executed on such machines use all the qubits. It can be seen as a supercomputer: El Capitan, the fastest supercomputer in the current Top500 rankings[1], feature 43 808 CPUs. However, only a few applications use all the CPUs at the same time, and the machine is shared between parallel applications.

For a given machine featuring q qubits and a set of circuits $C_1, C_2, ..., C_n$, each C_i using varying numbers of qubits (less than q qubits) and with same execution time, the optimization problem of finding the order of execution that would minimze the total

[1] https://www.top500.org

makespan is a *scheduling problem*. Because the output produced by one circuit can be taken as an input of another circuit, we consider additional precedence constraints for this problem.

Optimization in two steps

We assume to be given a topology of a quantum chip, topology which is a subgraph of a 2D-grid. Given a set of circuits with precedence constraints, we aim to schedule their execution on the quantum chip. This can be done by a two-step optimization method:

First step. The circuits are described as an ordered sequence of one-qubit and two-qubit gates, gates supposed to be natively implementable on the hardware. The goal of the first step is to transform those circuits into a geometrical description with connectivity constraints of the topology hardware. The transformation will also take into account the depth of the circuit by minimizing the number of SWAP gates.

Second step. Given the set of geometrical description of the circuits, the second step aims at scheduling the execution of the circuits on the hardware. This scheduling must satisfy precedence constraints between circuits' execution in addition to tiling specific considerations. This scheduling problem, whithout the precedence constraints, has already been tackled in a previous internship.

Practical information

Employer:	Laboratory LIPN within the framework of an academic internship agreement
Internship location:	LIPN (Laboratoire d'Informatique de Paris Nord), UMR CNRS 7030 Université Sorbonne Paris Nord 99 avenue Jean-Baptiste Clément 93430 Villetaneuse
Duration:	4-6 months at spring 2026
Remuneration:	Internship compensation
Required level:	Second year of Research Master or third year of Engineering School
Profil:	Combinatorial optimization, Computer science, Applied mathematics, Programming (ex: Python or C)

Advisors

Camille Grange	(LIPN, Sorbonne Paris Nord)	grange@lipn.univ-paris13.fr
Roberto Wolfler Calvo	(LIPN, Sorbonne Paris Nord)	wolfler@lipn.univ-paris13.fr