qibo: a full-stack framework for simulation, control and calibration of self-hosted qubit devices

Robbiati M. [1,2] Carrazza S. [1,2,3]

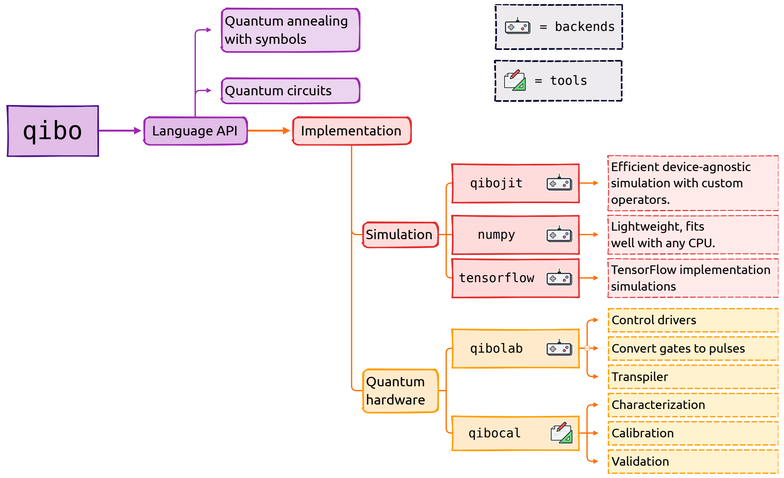
[1] TIF Lab, Dipartimento di Fisica, Università degli Studi di Milano, Milan, Italy. [2] CERN, Theoretical Physics Department, CH-1211 Geneva 23, Switzerland. [3] Quantum Research Center, Technology Innovation Institute, Abu Dhabi, UAE.

Abstract

We present qibo [1], a full-stack and open source framework which can be used for:

- efficiently simulate quantum computation routines, both in circuit notation and in the context of adiabatic computation;
- control self-hosted quantum devices with an hardware-agnostic approach;
- performing quantum characterization, calibration and verification routines on the hosted qubits.

OVERVIEW: the qibo ecosystem

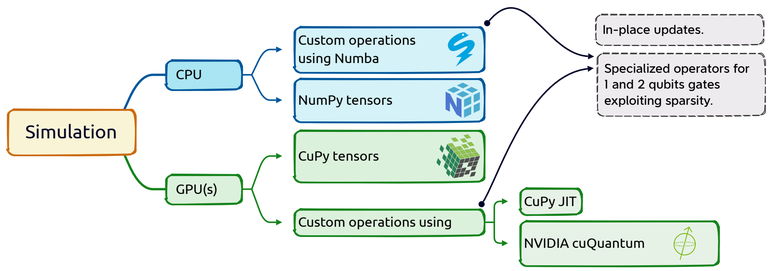


SIMULATION: qibojit

We do state vector simulation of a system of qubits $\{\sigma_j\}$, which solves:

$$\psi'(\sigma_1, ..., \sigma_n) = \sum_{\tau'} G(\tau, \tau')\psi(\sigma_1, ..., \tau', ..., \sigma_n), \tag{1}$$

The number of operations scales exponentially with N_{qubits}, thus we built the qibojit backend [2].

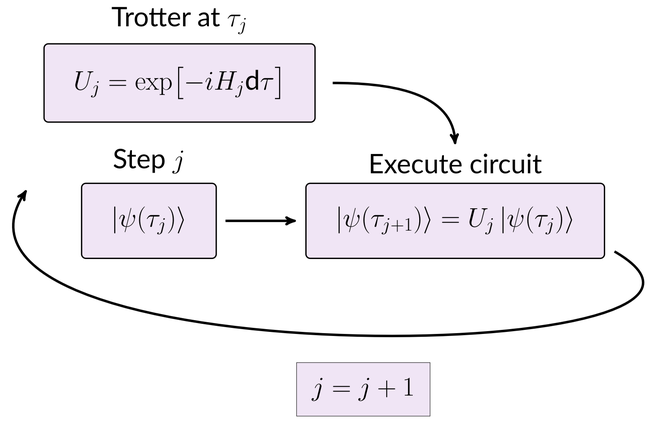


SIMULATION: adiabatic computation

In qibo symbolic hamiltonians can be defined and used to perform adiabatic computation:

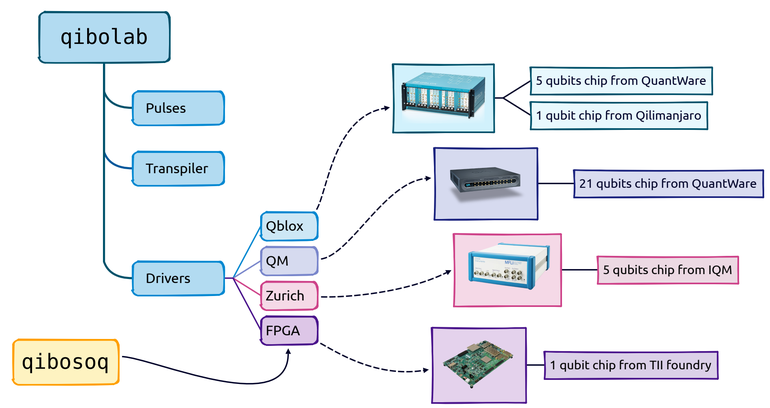
$$H_{\text{ad}}(\tau; \boldsymbol{\theta}) = \big[1 - s(\tau; \boldsymbol{\theta})\big] H_0 + s(\tau; \boldsymbol{\theta}) H_1, \tag{2}$$

even following a parametric scheduling $s(\tau, \boldsymbol{\theta})$.



CONTROL: qibolab

The full-stack framework is hardware agnostic!

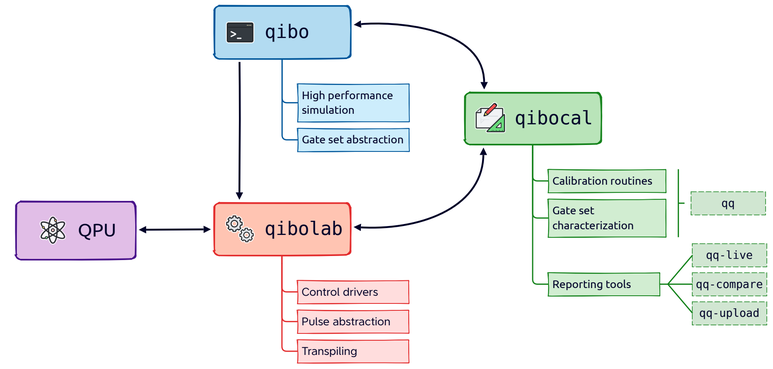


The qibo's high level language can be deploied on any quantum hardware technology by defining a platform object:

1. a quantum computation routine can be written with qibo;
2. define custom_platform object for a self-hosted device;
3. the hardware backend can be set via qibo.set_backend('qibolab', 'custom_platform').

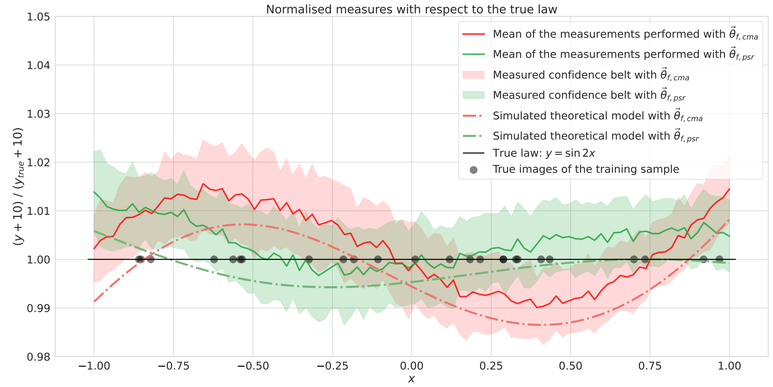
CALIBRATION: qibocal

Each qubit needs characterization, calibration, verification [3].



Full-stack gradient descent using qibo

We train a quantum circuit to fit $y = \sin(2x)$. The full gradient descent is performed on the device controlled via FPGA [4].



References

[1] S. Efthymiou, S. Ramos-Calderer, C. Bravo-Prieto, A. Pérez-Salinas, D. García-Martín, A. Garcia-Saez, J. I. Latorre, and S. Carrazza, "Qibo: a framework for quantum simulation with hardware acceleration," *Quantum Science and Technology*, vol. 7, p. 015018, dec 2021.

[2] S. Efthymiou, M. Lazzarin, A. Pasquale, and S. Carrazza, "Quantum simulation with just-in-time compilation," *Quantum*, vol. 6, p. 814, sep 2022.

[3] A. Pasquale, S. Efthymiou, S. Ramos-Calderer, J. Wilkens, I. Roth, and S. Carrazza, "Towards an open-source framework to perform quantum calibration and characterization," 2023.

[4] M. Robbiati, S. Efthymiou, A. Pasquale, and S. Carrazza, "A quantum analytical adam descent through parameter shift rule using qibo," 2022.