Towards a hybrid quantum operating system

Towards a hybrid quantum operating system

Andrea Pasquale on the behalf of the Qibo Team

9th May 2023, CHEP2023, Norfolk

Challenge

Is it possible to create from scratch a framework for all of this?

Introducing Qibo

Open-source full stack API for quantum simulation, hardware control and calibration

Qibojit

Quantum circuits are simulated through matrix multiplication

$$\psi'(\sigma_1, \ldots, \sigma_n) = \sum_{\boldsymbol{\tau'}} G(\boldsymbol{\tau}, \boldsymbol{\tau'}) \psi(\sigma_1, \ldots, \boldsymbol{\tau'}, \ldots, \sigma_n)$$

The number of operations scales exponentially with the number of qubits! We need more sophisticated tools to be able to simulate a quantum circuits with more qubits!

- NumpyBackend
- TensorFlowBackend
- QiboJITBackend:
  - CPU: Numpy tensor + custom operations with Numba JIT
  - GPU(S): Cupy tensors + custom operations using
    - Cupy JUST Raw kernels
    - NVIDIA cuQuantum API

Paper published on Quantum

Benchmark

Hardware Control using Qibo

For superconducting qubits gates are implemented by sending microwave pulses.

We need a framework to control all these devices at the same time.

Introducing Qibolab

Introducing Qibocal

A reporting tool for calibration using Qibo

Motivation

Suppose that we have assembled a quantum computer and we have a way to send pulses to the chip... are we done? NO

We need to characterize, validate and verificate our qubits.

Single Qubit Characterization

High level protocols

Applications

What can we achieve using Qibo + Qibolab + Qibocal?

Successfully performed a gradient descent on a QPU with a single using Parameter Shift Rule algorithm.

Outlook

Qibo is growing to accomodate different tasks:

- High performance quantum simulation: qibojit
- Hardware control: qibolab
- Hardware calibration: