Forum TERATEC 23

Unlock the future

31 MAI & 1er JUIN 2023 • Au Parc Floral, Paris

Un événement organisé par infoprodigital Teratec

Joseph MIKAEL
Forum Teratec 31/05/2023

Is it too early for a quantum algorithms engineer to ignore integration issues ?

Plan

1. Quantum @ EDF

2. The landscape of integration levels

3. The impact of integration and hardware on the algorithms

1.

Quantum @ EDF

EDF R&D : THE WORLD'S MOST ADVANCED ENERGY GROUP R&D DIVISION

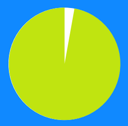
€487 million
budget in 2021 for
EDF R&D division (EDF SA)

1/3 is used for forecasting and paving the way for the Group's future

2/3 is used to support the performance of the Group's various business lines

Serving the EDF Group's
Raison d'Etre

"**To build a net zero energy future** with electricity and innovative solutions and services, to help save the planet and drive wellbeing and economic development."



99% of EDF R&D division operating budgets in France dedicated to decarbonising and enabling the transition of energy systems

EDF R&D IN FIGURES

R&D OUR EXPERTISE

Consumer knowledge | Data Science | Economy | Electrotechnology

Telecommunications / Industrial data processing | Polymer / Corrosion chemistry

Energy efficiency | High-power electricity | Environment / Waste

Data processing - virtual reality / Augmented reality | Civil engineering | Scientific computing

Applied mathematics | IT / Telecom | Fluid mechanics | Mechanics

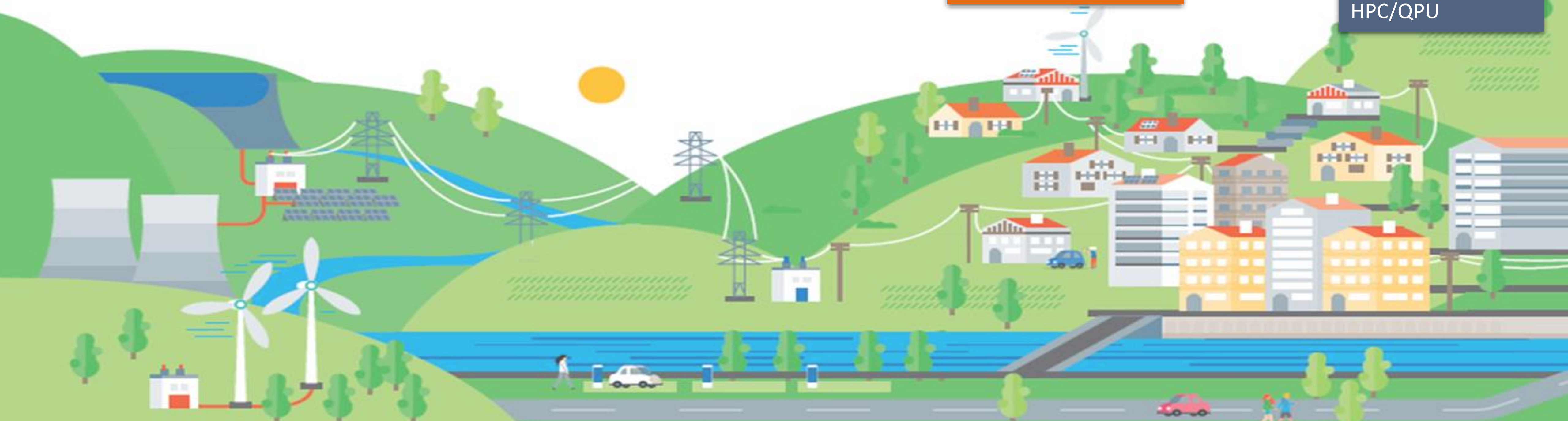
Mathematics / Physics / Computing | Neutronics technology | Cycle / Fuel physics

Intellectual property | Materials science metallurgy | Thermal-hydraulics and simulation

Quantum @each point of the value chain

Quantum Team
Teams of experts cumulating more than 20 000 citations
20 papers published on Quantum

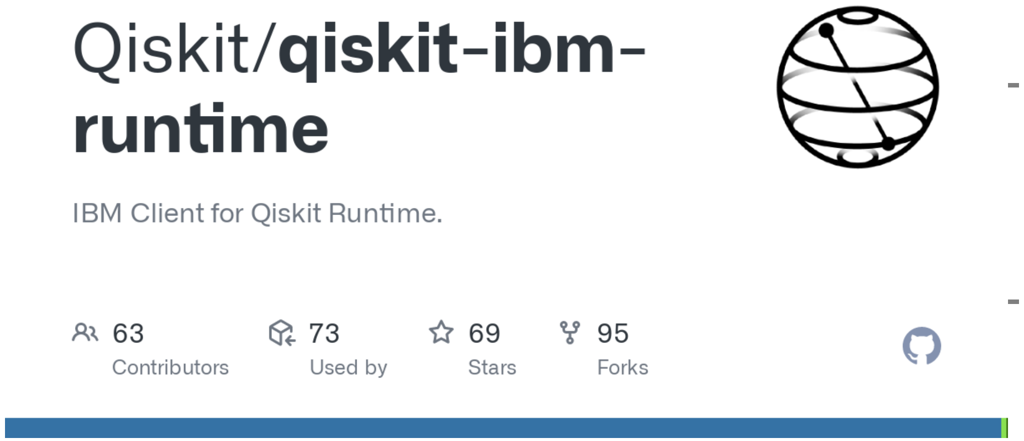
Production				Networks	Energy management		IT
Material Simulation of internals	Safety Risk Assesment	Partial Differential Equations	Metrology	Optimal Power Flow	Optimisation Scheduling Smart Charging V2G	Machine Learning Forecasting	Post-quantum Cryptography Integration HPC/QPU



2.

The landscape of integration level

QPU – CPU : Four level of integration

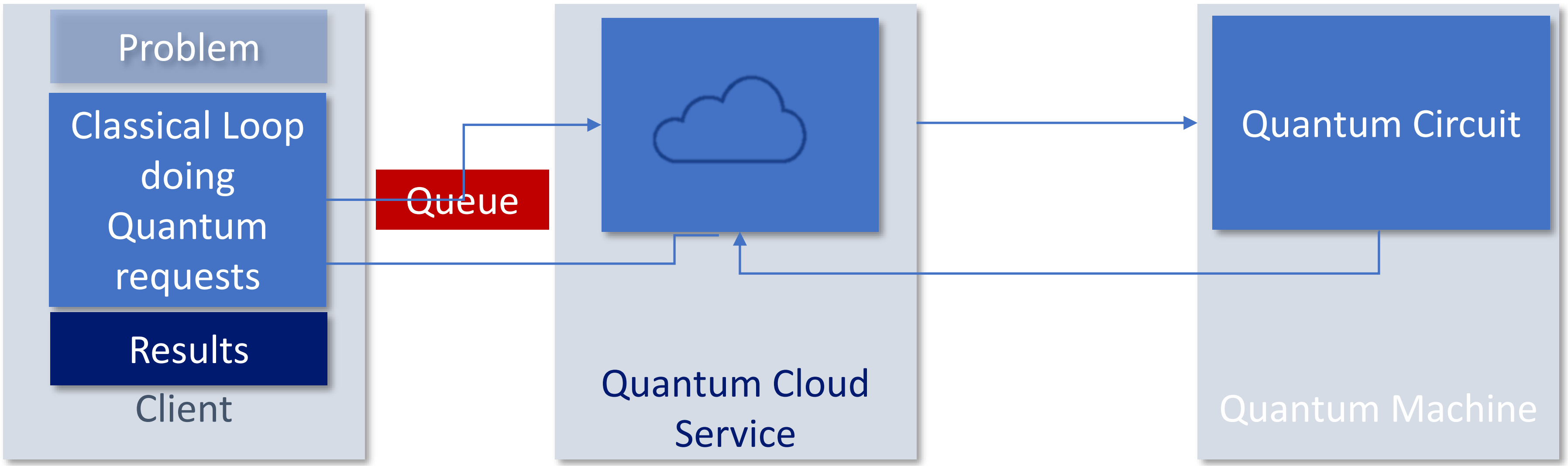


- On the Quantum Side the present and the future is hybrid.
- IBM & Microsoft defines 4 level of integration.

QPU – CPU : Four level of integration

Level 1 : Basic integration

- A QPU is somewhere and the CPU computations are done somewhere else.
- A *QAOA like* algorithm including a loop between the CPUs and the QPUs makes you ask for a new job and you queue at each iteration !

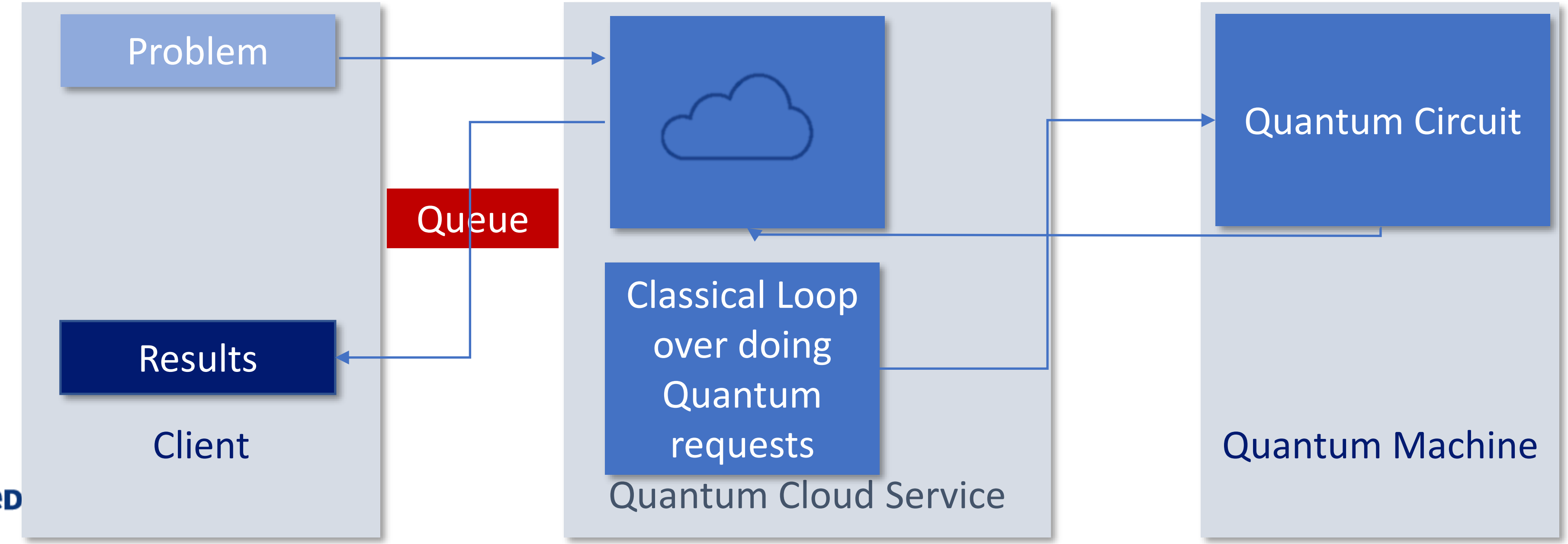


QPU – CPU : Four levels of integration

Level 2 : Integrated Batch

- The loop is executed in the cloud, with lower latency to the quantum machine and preferential queuing to the quantum machine.
- The quantum state is discarded at the end of each iteration of the loop and only a small amount of information is sent back to the pilot loop for the next iteration.
- This level does not execute anything that cannot be executed at level 1

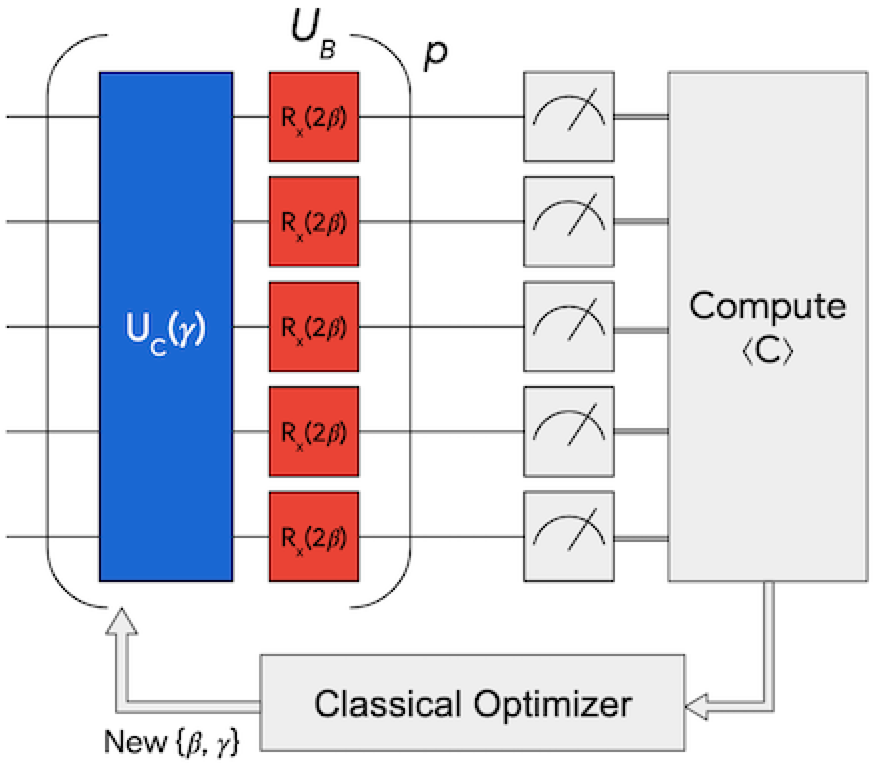
➔ For some algorithms, it offers a significant acceleration (up to 120 times faster with this mode according to IBM).
➔ You may lose some control over your algorithm.



QPU – CPU : Four level of integration

Level 3 : Integrated Quantum Computing

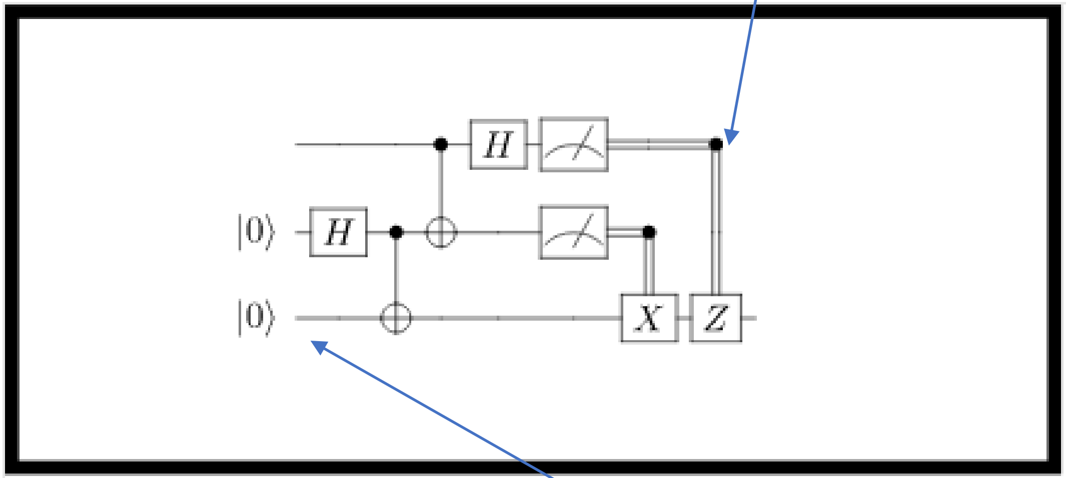
- Quantum static circuits may contain measurement operations only at the end of the circuit.



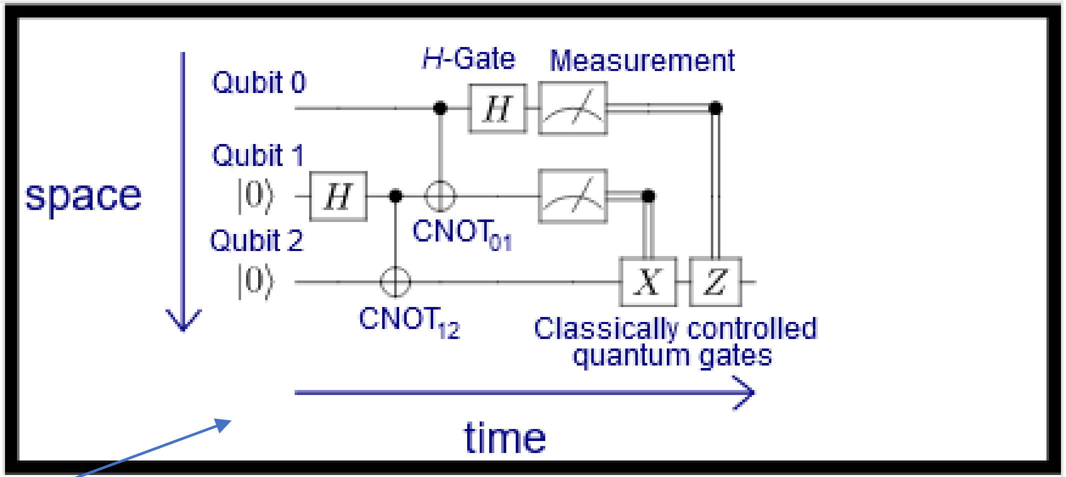
QPU – CPU : Four level of integration
Level 3 : Integrated Quantum Computing

For some algorithms, it may be much more efficient to do some partial measurement in order to change the parameters of the remaining parts of the circuits.

The gate is applied conditionally to the output measured classically



Teleportation Circuit



Annotated Teleportation Circuit

Courtesy of Microsoft

The partial measure and associated classical treatment has to be done during the coherence time of the whole circuit.

QPU – CPU : Four levels of integration

Level 4 : High Performance Quantum Computing

The restrictions of Level 3 are that

- the classical code and the quantum code can only interact between the iterations of the loop
- the classical code can only modify the parameter of the circuit, not changing it completely.

A tighter integration between the classical code and the quantum code is the object of level 4 in which one can modify the circuit itself and not only the parameters.

2.

Integration, Hardware & Algorithms

The impact of integration on the classical side.

- Today
  - a clever integration between say CPUs and GPUs might « only » accelerate the computation.
  - Apart from expert developers, algorithms developers don't care that much on the integration level nor on the machines their algorithms are applied on.

➔ *Modern Machine Learning scientists even just add a Flag « import tensorflow_gpu » to use a GPU.*

The integration became so high that it became transparent to the end user.

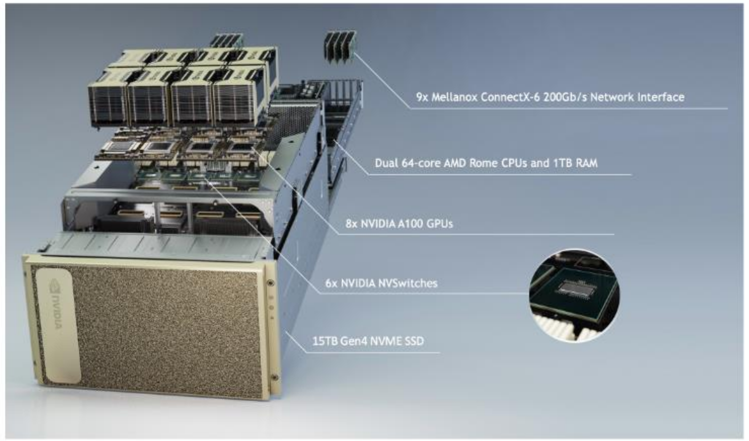


Figure 1. Major components inside NVIDIA DGX A100 system

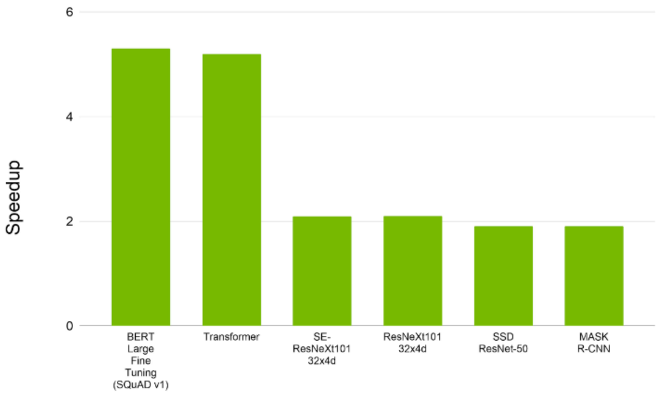


Figure 4. TF32 can provide over 5X speedup compared to FP32, PyTorch 1.6 in NGC pytorch:20.06-py3 container, training on BERT-Large model. Results on DGX A100 (8x A100 GPUs). All model scripts can be found in the Deep Learning Examples repository

The level of integration & Hardware on
Quantum algorithms

- In the Quantum World, things are still different from a hardware to another.
- **It might be a desirable feature for now no converging that fast.**
- In the following, we show some examples showing how the nature of the hardware and of the integration might lead to very different natures of algorithms.

Hardware impact on the Algorithms
Combinatorial problems 1/2

Let's consider a Combinatorial problem having some constraints.
To deal with these problems, one may

- **Transform the problem into a QUBO** and add additional terms to handle the constraints.
  - On **Dwave** the implementation will be conditioned to the required connectivity [1].
  - On **Gate based models**, one can run a QAOA algorithms but the more constraints you'll have the more pain you'll have to make the algorithm converges [2].

- **Transform the problem into a Graph** (e.g Max Independent Set) and use the Rydberg Blockade Mechanism to integrate some constraints [3]

➔ At the end depending on your problem and the machine, you won't get even the same kind of results.

[1] Deleplanque, Samuel. "Quantum Computing for solving the 3SAT problem by reduction to the MIS combinatorial optimization problem." (2023).
[2] Akshay, V., et al. "Reachability deficits in quantum approximate optimization of graph problems." *Quantum* 5 (2021): 532.
[3] Nguyen, Minh-Thi, et al. "Quantum optimization with arbitrary connectivity using Rydberg atom arrays." *PRX Quantum* 4.1 (2023): 010316.

Integration impact on the Algorithms

Combinatorial Problems (Maximum Independent Set (2))

- One can imagine that in the Future, QPUs will be installed in parallel.
- The way this parallelisation is done may change in a deep way both the algorithms and the machine requirements.
  - With a naive parallelisation between QPUs that only share the results, you won’t gain anything
  - If you add the possibility to do a qubit teleportation, both the requirements in terms of qubits and connectivity might be much more lower.

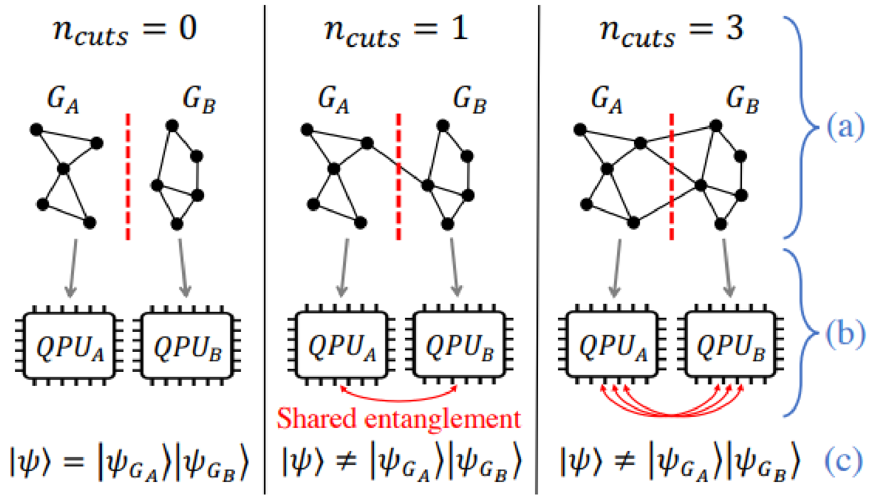


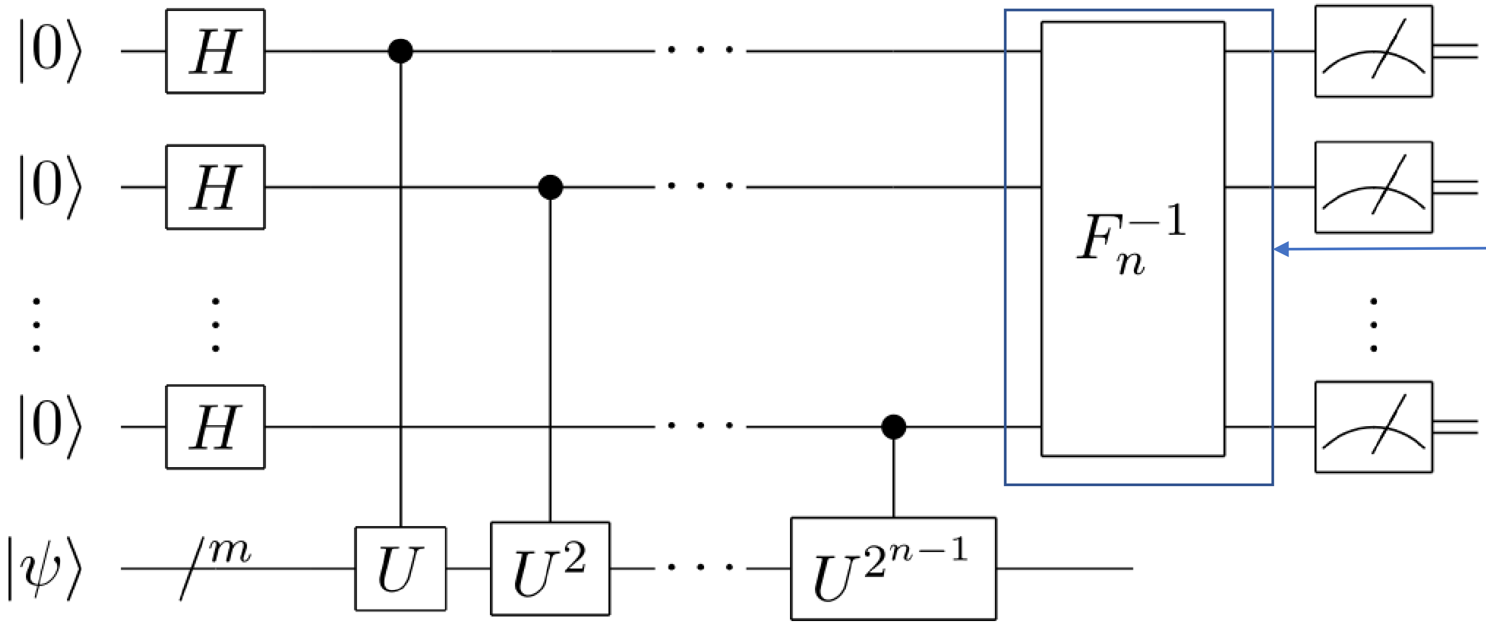
Figure 2: In-depth view of three possible compilations within the ansatz construction and quantum circuit cutting steps of Figure 1. After the graph has been partitioned, (a) the ansatz is constructed and the hyperparameter n_{cuts} indicates how many edges should be allowed to cross the partition. That ansatz is then (b) mapped to distributed quantum computers using quantum circuit cutting techniques. When (c) the ansatz is executed, the final output state $|\psi\rangle$ may be entangled if $n_{cuts} > 0$. As more edges are allowed to cross the partition, more information from the original input graph is preserved at the cost of increasing quantum communication overheads.

Saleem, Zain H., et al. "Divide and Conquer for Combinatorial Optimization and Distributed Quantum Computation." *arXiv preprint arXiv:2107.07532* (2021).

Integration impact on the Algorithms

Quantum Phase estimation

- The vanilla version of Quantum phase estimation is the most costly in terms of quantum hardware: it requires a large number of qubits, low error rate and good enough connectivity.



You won't get such entangled state in the NISQ era

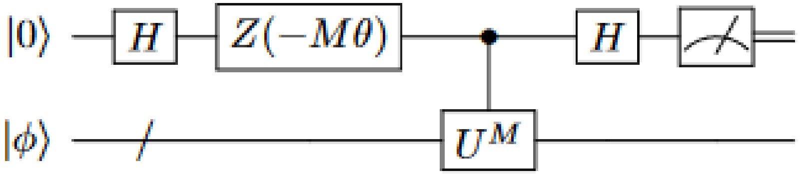
Inspired from https://vivienlonde.github.io/blog/estimation-de-phase

Integration impact on the Algorithms Quantum Phase estimation

- **Bayesian Phase estimation** [1] is an improvement of Phase Estimation and is based on the Iterative Phase estimation family.
- It is the most demanding in terms of classical hardware as
  - it needs an exponentially long classical calculation in terms of the number of bits of precision required at each iteration.

  Moreover we need to be able to keep the eigenstate |ψ⟩ in quantum memory during this classical computation time.

PE algorithms use this circuit to infer the bits in a binary expansion of φ in order from least significant to most significant.



[1] K. M. Svore, M. B. Hastings, and M. Freedman, “Faster phase estimation”

Inspired from https://vivienlonde.github.io/blog/estimation-de-phase

Integration impact on the Algorithms Quantum Phase estimation

- **Random walk** estimation is still based on Iterative Phase Estimation but has the disadvantage of being adaptive:
  - the parameters of the new iteration depend on the measurement results of the previous iteration requiring very fast communication between the quantum processor and classical one.

Algorithm	Pro	Cons
Phase Estimation (PE)	Low Integration with HPC Required	Far from being attainable in the NISQ Era. High entanglement required
Bayesian PE	Closer to NISQ	HPC requirement too high
Random Walk PE	Closer To NISQ	Very high integration between HPC & QPU

Inspired from https://vivienlonde.github.io/blog/estimation-de-phase


Merci