Quantum/HPC Hybridization with Qaptiva – Recent Achievements and Perspectives

Cyril ALLOUCHE
VP, Quantum R&D

an atos business

EVIDEN

Integrating QPUs in the HPC center

The need for (large scale) emulation

Perspectives : large scale heterogeneous computing

1 Integrating QPUs in the HPC center

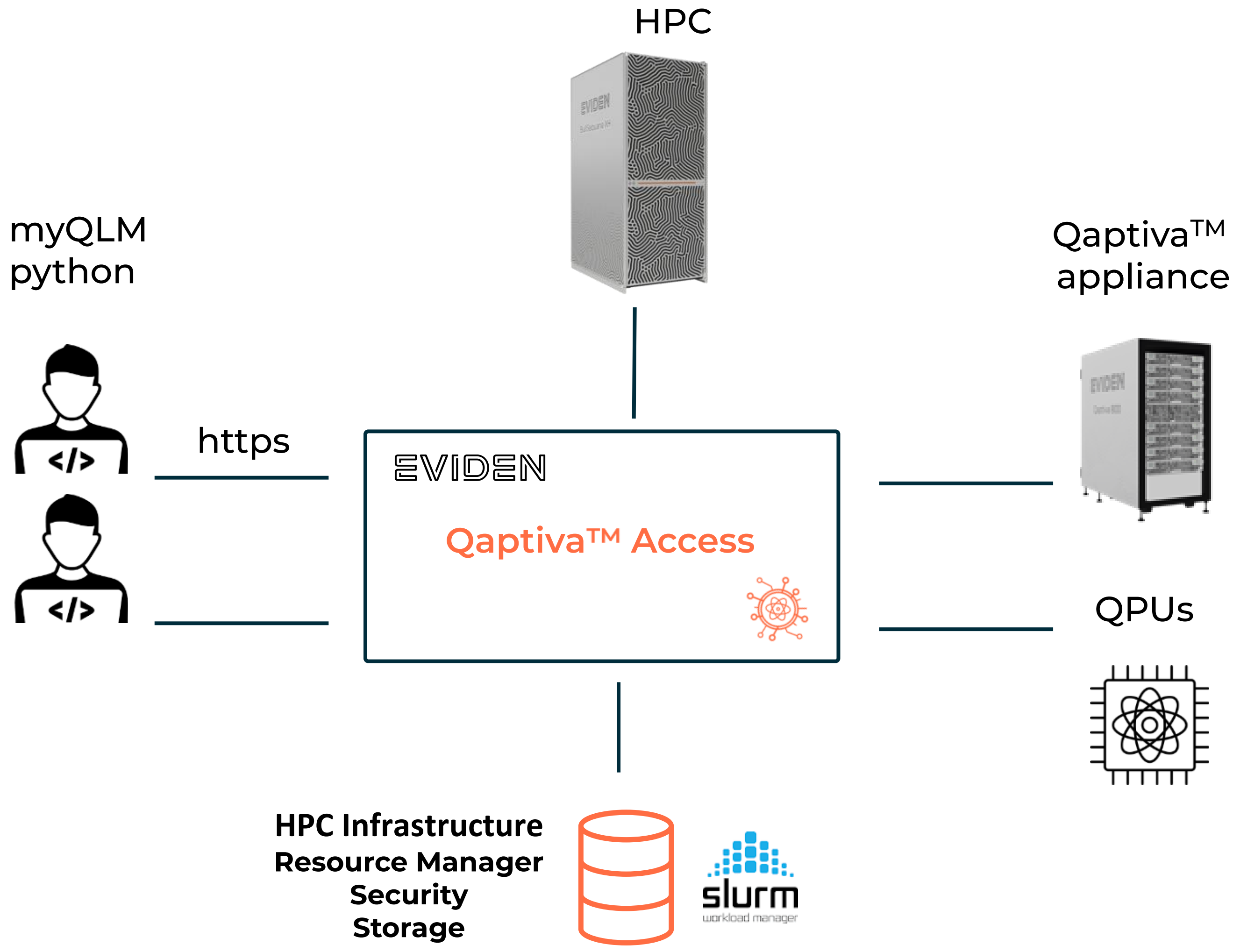
Challenges to integrate (early) QPUs into a HPC datacenter

1. Availability
2. Resource Management
3. Multi-User

Our solution: hybridization node – Qaptiva Access

Integrating Quantum Technologies in the HPC - Qaptiva™ Access

HPC & Quantum hybridization



Integration of any quantum processing unit (QPU) into the HPC infrastructure

- Qaptiva Access : HPC/QC hybridization node
- myQLM library provides quantum programming capabilities to the cluster
- Enables scheduling of QPUs with SLURM
- Enables quantum emulators on HPC

Used in major HPC-QC integrations:

- **TGCC, HQI platform**, French national supercomputing infrastructure
- **Julich Supercomputing Center**, HPCQS and Qsolid projects

Qaptiva Empowering HQI at TGCC

myQLM library deployed on compute nodes

Large quantum simulations Can be run on cluster

requests for quantum resources go thru Qaptiva Access

Qaptiva Access

Qaptiva appliance handles fat node simulations, compilations, etc..

Pasqal QPU receives quantum instructions from Access, with respect with Slurm scheduling

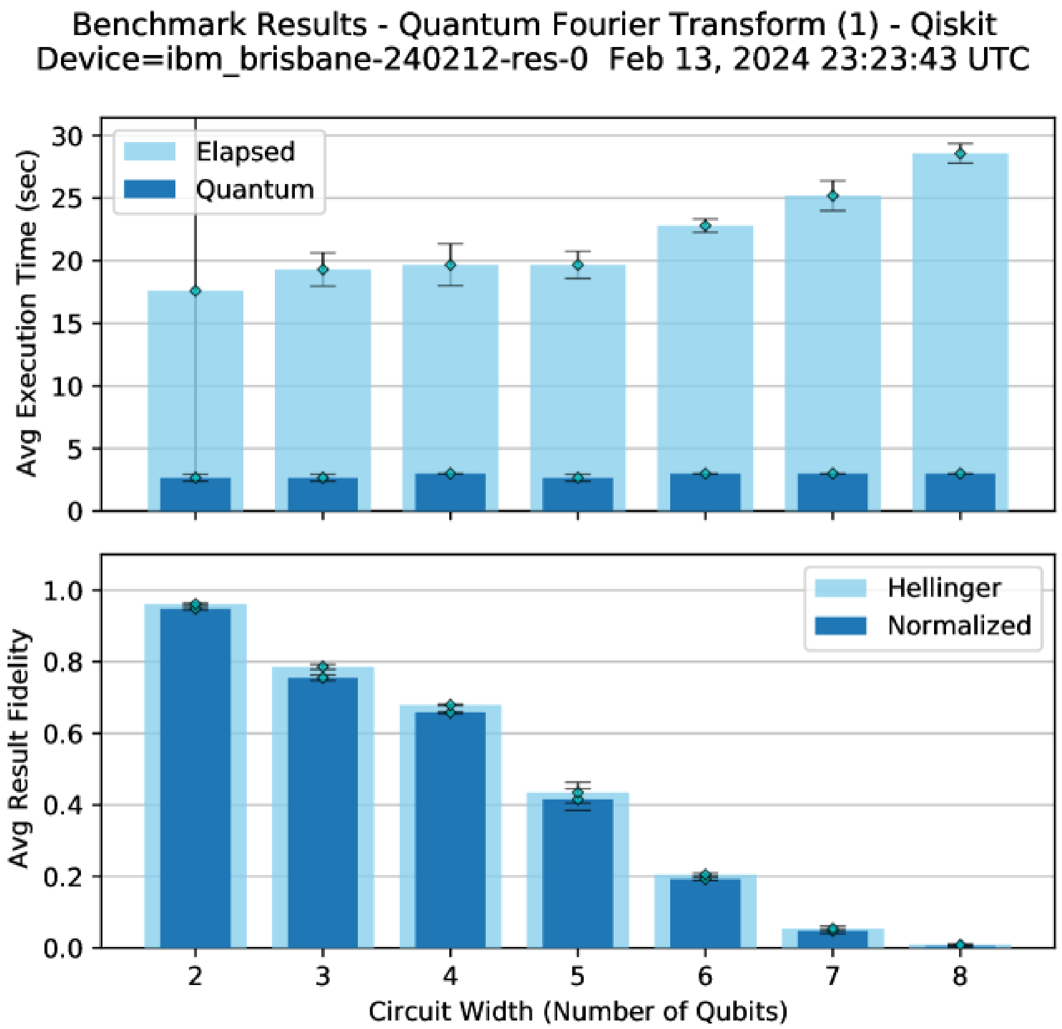
slurm
workload manager

Qaptiva delegates scheduling of quantum resources to Slurm

EVIDEN

2 The Need for (Large Scale) Emulation

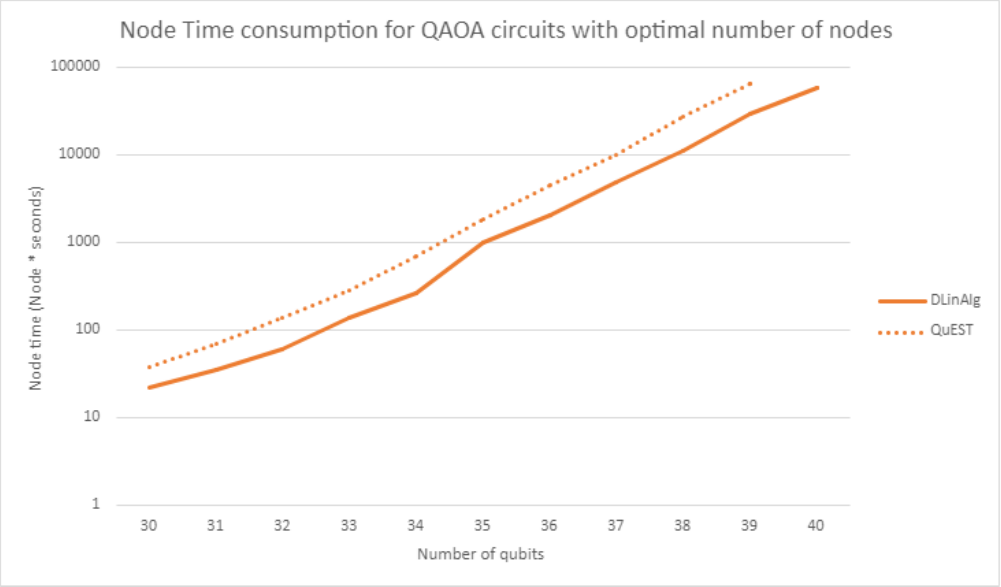
Today's QPUs are still Noisy and Slow

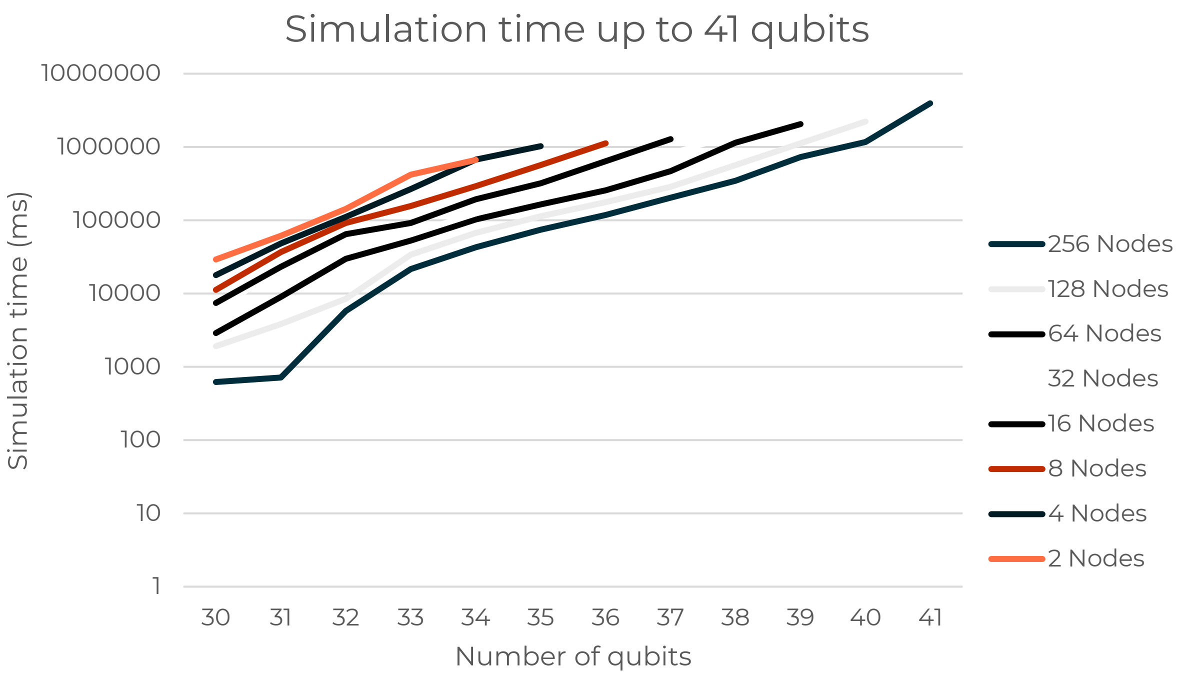


https://arxiv.org/html/2402.08985v1

Emulation on the HPC with Qaptiva

- MPI distributed emulators, highly optimized for scaling
  - **reaching 43 qubits with 1024 nodes**
- Launched either as HPC jobs (sbatch) or from python interface with myQLM





EVIDEN

3 Perspectives : Large Scale Heterogeneous Computing

NISQ Programming Paradigm Won’t Scale

- NISQ model :
  - Control flow managed by CPU
  - Quantum circuits created by CPU
  - Repeated evaluation of circuit by QPU

  ⇒ **QPU online slave of CPU, interpreted programming**

- This cannot scale to multi CPU * multi QPU
- This cannot scale to FTQC
- Interpreted programming is not compatible with existing HPC applications

- **Need for a heterogeneous HPC – Quantum paradigm**

Q-Pragma – A C++ Framework for Large Scale Quantum Computing

Q-Pragma C++ framework:

Extending C++ with Quantum Computing only with pragmas directives

Enables hybridization in existing C++ codes without rewriting

Compatible with standard heterogeneous computing techniques

https://arxiv.org/abs/2309.02605

Shor algorithm in 10 lines

```
uint64_t to_divide = …, random_base = …, measurement
= 0UL;

#pragma quantum scope with (to_divide, random_base,
measurement)
{
    qint_t<QSIZE> first_register;
    wall::H<QSIZE>(first_register);

    qint_t<QSIZE> second_register =
        qpragma::pow(random_base, first_register) %
to_divide;
    reset(second_register);

    qft<QSIZE>(first_register);
    measurement = measure_and_reset(first_register);
}
```

EVIDEN

Questions

Thanks!

For more information, please contact:
Cyril ALLOUCHE