Integrating multiple QPUs in a pre-existing HPC environment

Applicative insights from the integration of Ruby and Lucy

louis.beaurepaire@asplus.fr – HPC/QC Engineer @ AS+

Alliance Services Plus: HPC experts

Overview of our activity

- Specialized in services around HPC and AI for 15+ years
- On-site activity on France's 3 national Tier-1 computing centers (TGCC, IDRIS, CINES)
- Operating on major industrial supercomputers (Safran, Total Energies)

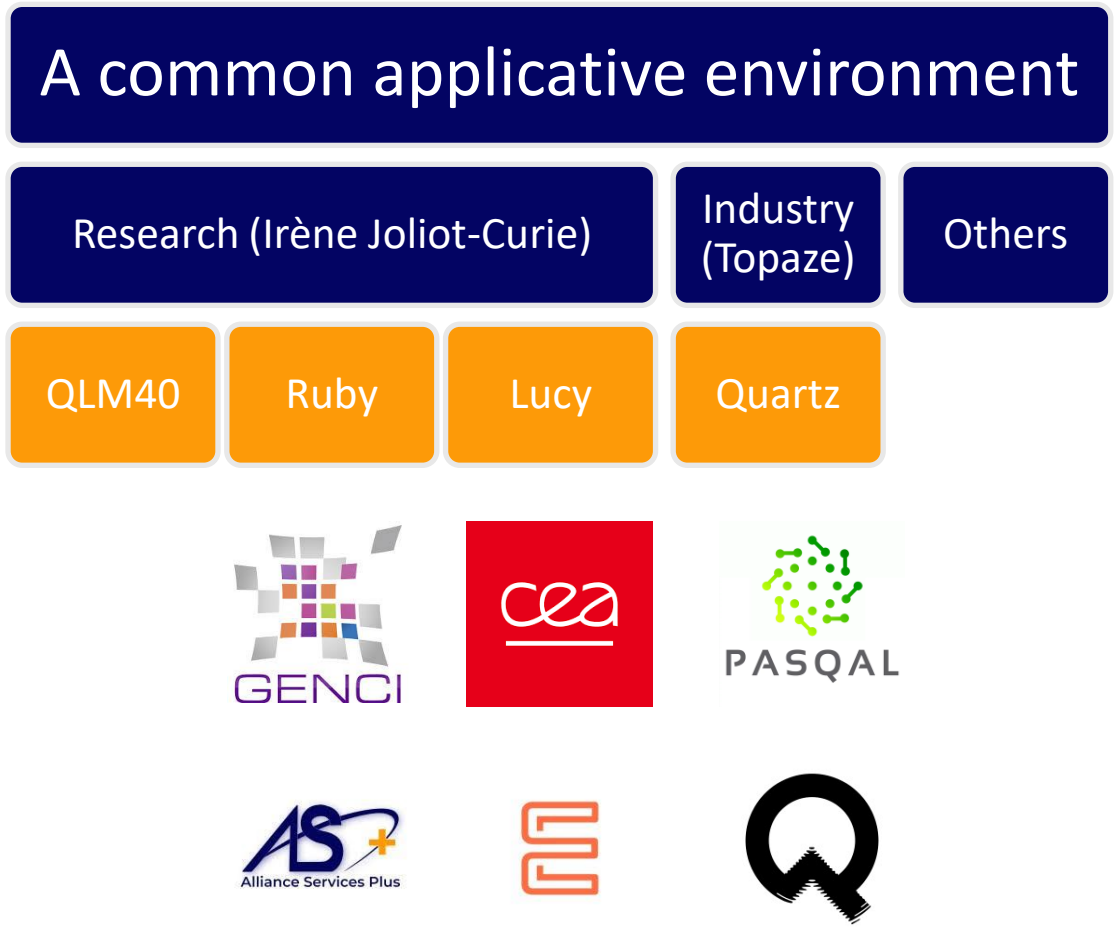
Our role at the TGCC

- Setting up the applicative environment
- Integrating software in our computing environment
- Running performance benchmarks
- Providing user support
- Technical lead of the quantum stack

The context: a complex environment

- Multiple air-gapped supercomputers in constant evolution
- Wide panel of users and needs
- Ever-evolving environment with shifting priorities
- Multiple actors working together behind the scene

→ We need a tailor-made solution



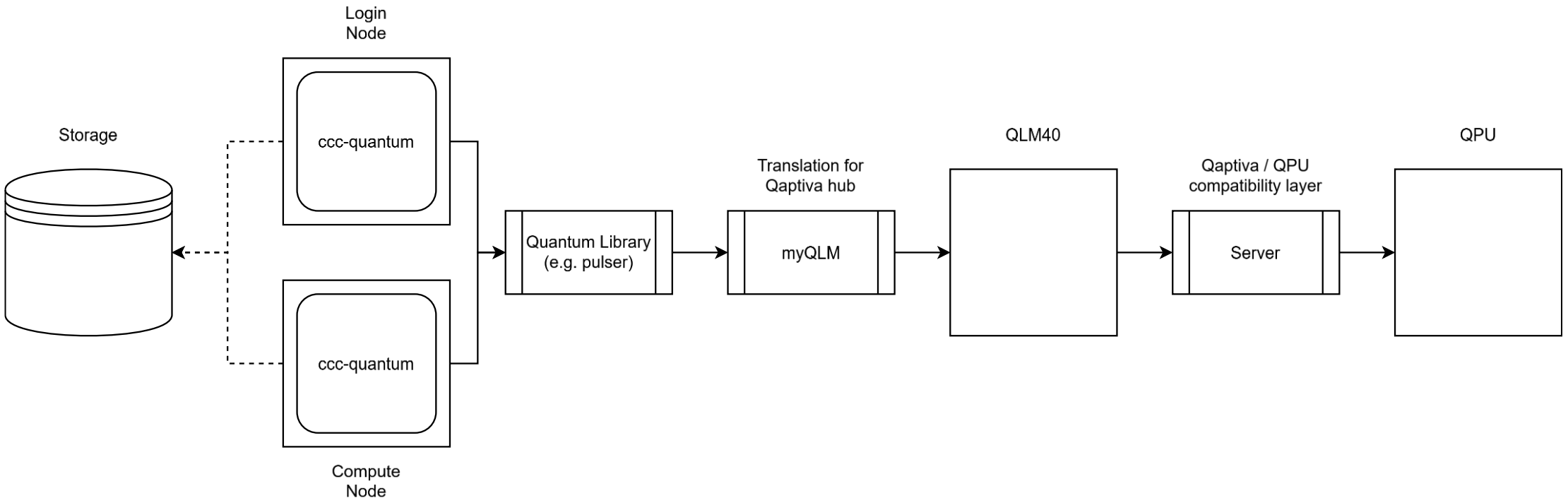
Our main challenge:

How to make quantum stacks available and user-friendly on an HPC infrastructure?

Our solution: a container dedicated to Quantum Computing

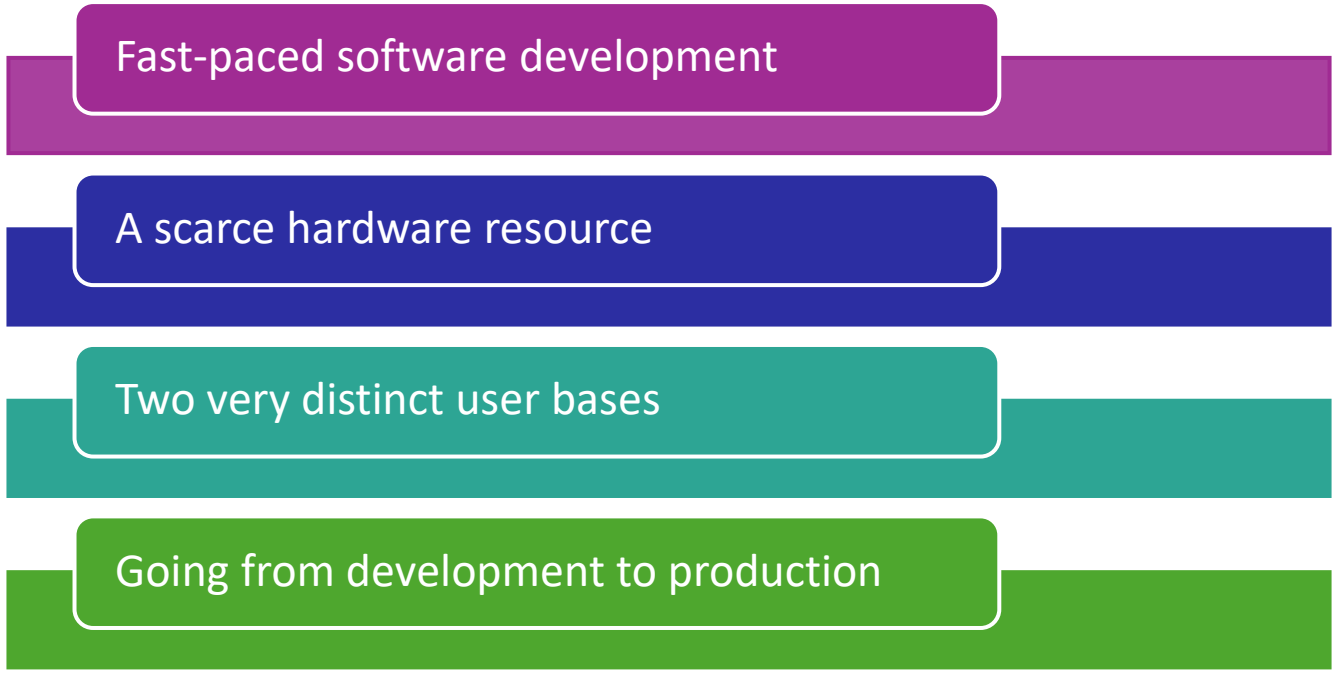
- The ccc-quantum container provides a unified environment for quantum computing at the TGCC
  - Integrated in the existing HPC context (BRIDGE, pcocc-rs, remote desktop)
  - All available hardware is made addressable
  - Provision of QC libraries (myqlm, qiskit, pulser, perceval, cirq)
  - Provision of common tools (jupyter notebooks, matplotlib, etc)
  - Provision of tutorials and learning resources
- This approach allows for complete control over the environment

A typical submission workflow



- Container is usable on login or compute node
- Filesystems are accessible from the container
- Usual libraries are usable directly, translation to the Qaptiva hub model is handled automatically
- Admins need to make sure the ccc-quantum client side is compatible with the server side

Challenges that differ from our usual HPC needs



The provided stack must be up-to-date

Fast-paced software development

- Packages are provided by the manufacturer and are closely related to the hardware capabilities → updating is crucial to offer the most of a machine
- The global QC ecosystem fosters novelty → release-cycle of packages is shortened
- The provided stack must remain coherent and functional

A scarce hardware resource

Two very distinct user bases

Going from development to production

Resource scheduling: a tale as old as HPC

Fast-paced software development

A scarce hardware resource

- A single QPU can be requested by multiple concurrent jobs (the issue is not new but has often been solved by multiplicity)
- For now, QPU usage is serialized and handled by a dedicated scheduler
- The Qaptiva acts as an access hub

Two very distinct user bases

Going from development to production

Two communities with different needs

- Fast-paced software development
- A scarce hardware resource
- Two very distinct user bases
  - QC-abled users not versed in HPC and its customs
  - HPC users that want to explore the possibilities offered by QC
  - Both communities require support and assistance on different topics
- Going from development to production

Integrating for production

- Fast-paced software development
- A scarce hardware resource
- Two very distinct user bases
- Going from development to production
  - Functional tests are easy, compliance benchmarking is not
  - Written codes can be made non-functional with a software update

Main takeaways

Fast-paced software development	Deployment of the applicative stack must be robust and reliable
A scarce hardware resource	Provision to the user of the hardware stack must be further improved upon
Two very distinct user bases	Support and learning material must cater to the two audiences
Going from development to production	The operational side of quantum computing is still overlooked

Our internal tool: `quantum-easydeploy`

- A tool to deploy ready-to-use hybrid computing environments
- Scalable, customizable, built for cloud but usable anywhere
- Perfect for prototyping and iterating on ideas
- An Ansible playbook and three commands to start computing:

```
$ docker run hpc-base
$ ansible-playbook quantum.yaml
$ conda activate qenv
```

- Developed with open-source as a mid-goal

Closing thoughts

- While some issues have been addressed, we are preparing for new challenges
  - Incompatibilities between two vendor software stack
  - The need for support will soon morph in need for counsel
  - Availability of QPUs will become critical
- The operational side of quantum computing must be production-ready before any real computational chain can reach the production stage
- Some ideas only look good on paper, prototyping is crucial