EFFICIENT HPC-QC RESOURCES ALLOCATION

Introduction

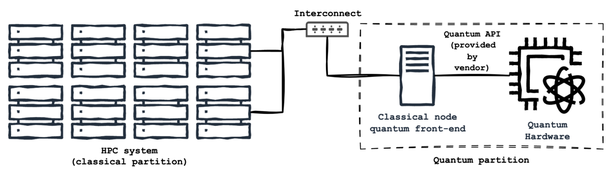
The drive for HPC-Quantum integration

Quantum Computers are considered an excellent candidate for accelerating computationally expensive tasks in chemistry and materials research.
These fields of study also represent typical examples of research domains for High Performance Computing (HPC); therefore, hybrid classical-quantum approaches are desirable and actively researched.

To enable the integration of real HPC workloads (i.e., large-scale MPI jobs) with quantum jobs beyond the well-known challenges related to compilers, software stack, programming languages, etc., we claim that addressing the fair and efficient allocation of quantum and classical resources is crucial. Without such strategies, both quantum and classical resources risk being underutilised or wasted.

In this work, we evaluated the approaches presented by Viviani et al. [1]:

- **Time-based multiplexing (virtual QPUs)**
- **MPI malleability**
- **Workflows**



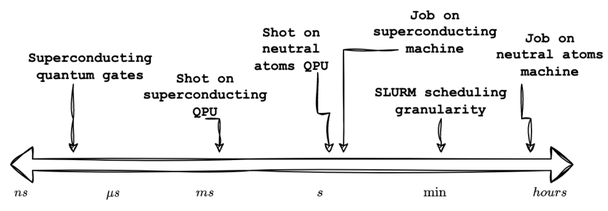
Background and Challenges

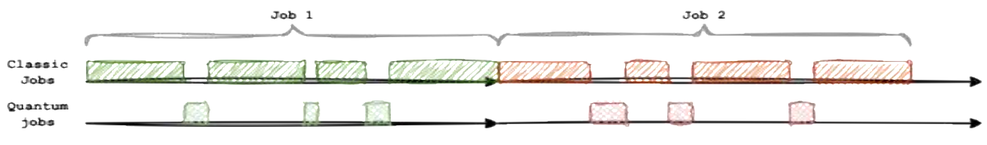
Quantum computers are a scarce resource

Due to technological limitation, complexity and cost, Quantum Computers are far from being an industrialised commodity. Conversely, they are a scarce and costly resource, often shared among many users through cloud-based access models. To unlock the potential of hybrid HPC-QC systems, we need efficient scheduling strategies that maximize utilization, minimize waiting times, and ensure fair access to these limited quantum resources.

Focusing on workload imbalance

A key factor in defining an effective scheduling strategy lies in the relative time scale of quantum and classical workloads within hybrid jobs. Given the current characteristics of the available superconducting hardware, and of typical quantum-accelerated applications, it is particularly relevant to consider scenarios in which the quantum contribution occupies only a marginal portion of the total runtime.
Without a proper resource allocation strategy, heterogeneous jobs are submitted under a simple co-scheduling model to a computing system with only one available QPU. In this setting, each user obtains exclusive access to the QPU during the quantum phase of their job. Due to the pronounced workload imbalance between classical and quantum computations, either the QPU or the classical nodes remain idle for most of the execution time, leading to suboptimal utilization of the system.

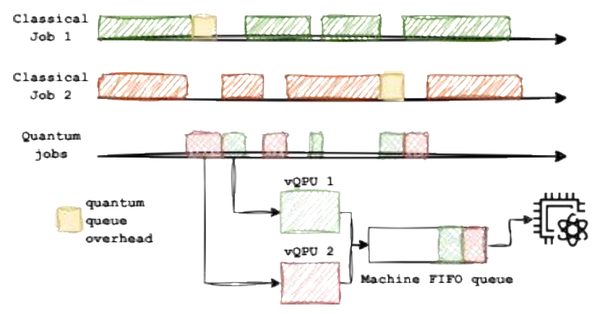




Methods - Virtual QPUs

Enabling concurrency with time-multiplexing

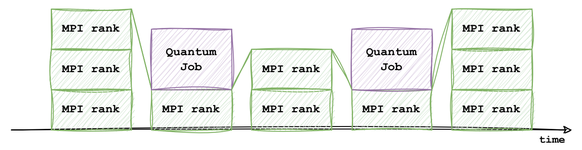
A promising strategy relies on time-based multiplexing, which enables multiple applications to share the same physical QPU by alternating their quantum executions. In this model, the QPU is shared over time, allowing each application to perform its quantum operations within distinct temporal windows. This concept can be effectively realized through QPU virtualization, where a fixed or dynamic number of virtual QPUs (vQPUs) are defined, each corresponding to a fraction of the QPU's total runtime



Methods - Malleability

Dynamic scaling of classical resources

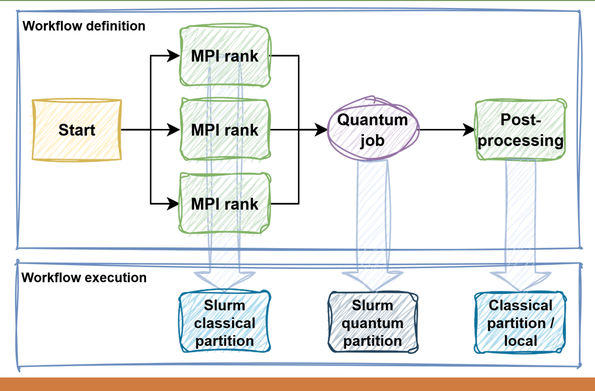
Malleability refers to the ability of parallel applications to adjust their resources at runtime, such as the number of nodes. Malleable applications can adapt to changing system conditions, improving overall resource utilization and reducing wait times on the target cluster. Malleable applications can release most of their resources when appropriate, for example, upon a quantum job submission. When the result becomes available, the classical part can immediately continue computation on an active node and/or request new nodes for a parallel workload.



Methods - Workflows

Orchestration of tasks across resources

An established strategy relies on defining multiple separate steps whose execution is managed by a workflow engine. Users can specify the desired series of operations and the possible execution environments using workflow-specific languages. Each workflow part can run on the optimal number of computing nodes, reducing budget consumption compared to statically allocating all nodes required throughout the application's lifespan.



Results - vQPUs

BBQ-MIS: Solving the Graph Coloring Problem

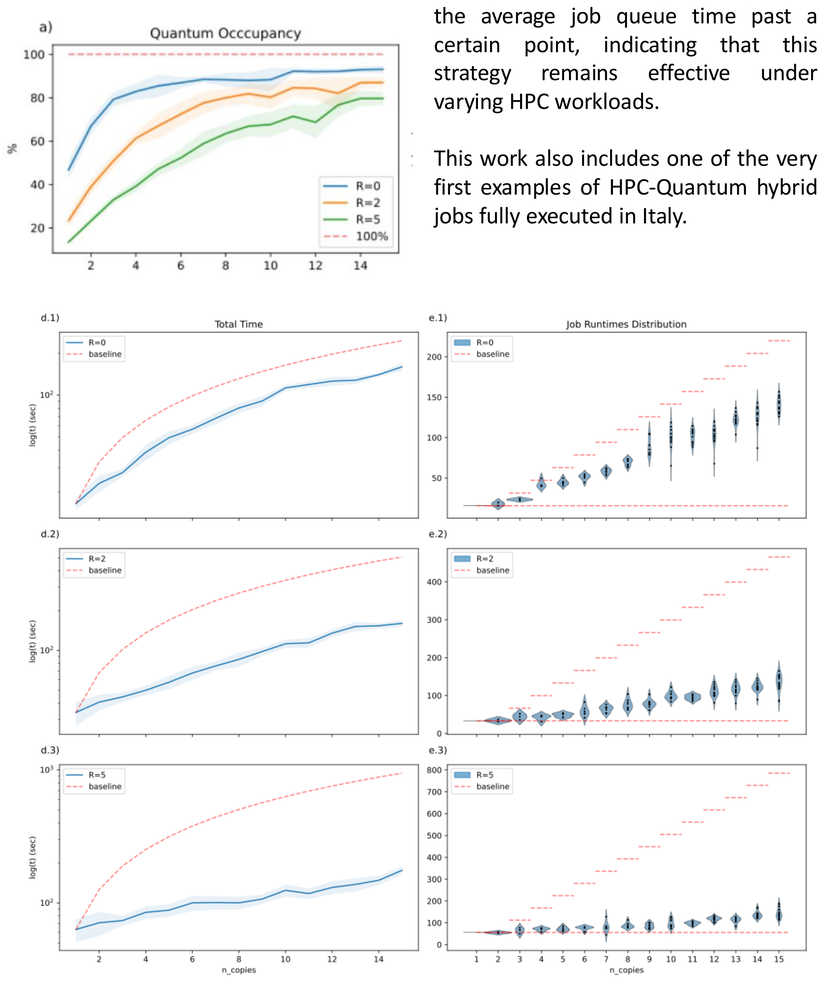
The selected benchmark problem for testing the effectiveness of the vQPUs strategy is the Graph Coloring (GC) problem, which consists of finding the minimum number of labels required to assign to the vertices of an undirected graph such that no two adjacent vertices share the same label.
Each GC instance is solved using the hybrid branch-and-bound BBQ-MIS algorithm, originally introduced in [2]. Owing to the algorithm's inherent structural parallelism, multiple MPI workers can be allocated to accelerate the search for the optimal solution, making this method suitable to be executed in an HPC environment.
The search for the optimal solution proceeds by decomposing the original problem into several Maximal Independent Set (mIS) subproblems. Each MIS problem is then solved using the Quantum Approximate Optimization Algorithm (QAOA).

Setup

The experiment required the execution of a batch job in an HPC environment with access to both classical and quantum resources. The classical workload was carried out on the Leonardo supercomputer, hosted at the CINECA facilities, while the quantum workload was dispatched to the "Lagrange" quantum computer, hosted at Politecnico di Torino and managed by Fondazione LINKS. The experimental workflow is organized around a main loop in which, at each iteration, an increasing number of GC instances is submitted simultaneously. This setup was designed to emulate concurrent access to the QPU by multiple users. In addition, workflow runs were executed under different classical-quantum workload ratios (i.e., relative duration), denoted with parameter R in the plots.

Metrics

All metrics are evaluated as the number of concurrent jobs, i.e., *"n_copies"* , increases. The considered metrics are:
a) **quantum occupancy**: the percentage of the total execution time during which the QPU is actively used.
b) **total time**: the elapsed time between the start of the first GC job and the completion of all jobs on the cluster



Discussion

The experimental results demonstrate that time-based multiplexing optimizes QPU usage, while significantly reducing overall execution time at cluster level. In addition, this performance gain becomes increasingly pronounced as the workload imbalance grows, making this approach particularly effective under the considered superconducting-hardware scenario. The observed trends also suggest that increasing the number of vQPUs instantiated in the HPC system does not significantly affect the average job queue time past a certain point, indicating that this strategy remains effective under varying HPC workloads.

This work also includes one of the very first examples of HPC-Quantum hybrid jobs fully executed in Italy.

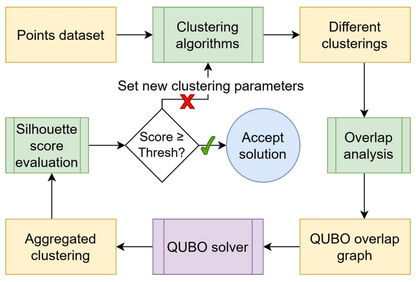
Results – Malleability and Workflows

Reducing clustering aggregation to a QUBO problem

The selected benchmark application for testing the effectiveness of the malleability and workflow strategy is the clustering aggregation problem. Our formulation builds upon the work in [3], focusing more on HPC resources.
We perform a selection of clustering algorithms over a dataset of points. The results of every algorithm are combined into a graph, where vertices are groups of points and edges represent overlaps between them. To determine a valid exclusive clustering, we reduce our problem to a Maximum Independent Set problem and write it in a QUBO formulation, which can be solved by a QUBO solver, either classical or quantum. Our application computes the Silhouette score for the candidate solution and continues looping until it reaches a custom-defined threshold.

Setup

These experiments were run on a custom-built cluster in the E4 datacenter. The classical parts of the application were scheduled on a compute partition comprising three nodes. In contrast, the quantum parts were dispatched to a node in another partition, which acted as a quantum emulator. We ran each approach both singularly and concurrently. In the latter approach, we scheduled the execution of two identical applications to run simultaneously. We ran each experiment using two variants of the quantum emulator: the first took a fraction of a second, akin to superconducting QC, while the other took about two minutes, mimicking a neutral-atoms QC. We used the DMR framework to implement malleability support and UniTo's StreamFlow to orchestrate workflow executions.



Metrics

We compare our approaches to a baseline in which every node in the compute partition is allocated for the entire application timespan.
The considered metrics are:
a) **wall time**: the elapsed time between the start and the end of an application.
b) **resource usage**: the sum of the product between the number of classical nodes used in any application part and the duration of that part.

TABLE I
EXECUTIONS WITH 2 MINUTES LONG QUANTUM JOBS.

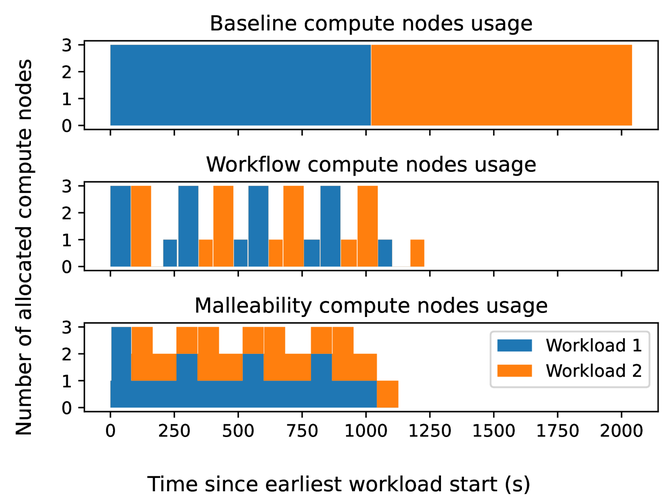
Execution Type	Mode	Wall time [seconds]	Resource usage [node-seconds]
Baseline	Single	1019.58 ± 0.85	3058.74 ± 2.56
Workflow	Single	1057.80 ± 6.02	1161.20 ± 6.94
Malleability	Single	1029.06 ± 1.54	1647.75 ± 1.54
Baseline	Double	2038.43 ± 0.96	6115.30 ± 2.89
Workflow	Double	1226.00 ± 1.58	2324.00 ± 3.39
Malleability	Double	1127.65 ± 1.18	2817.73 ± 1.27

TABLE II
EXECUTIONS WITH SHORT (< 1 SECOND) QUANTUM JOBS.

Execution Type	Mode	Wall time [seconds]	Resource usage [node-seconds]
Baseline	Single	539.44 ± 0.53	1618.33 ± 1.60
Workflow	Single	569.00 ± 3.94	1148.00 ± 1.87
Malleability	Single	549.60 ± 1.86	1168.29 ± 1.81
Baseline	Double	1076.98 ± 1.79	3230.95 ± 5.37
Workflow	Double	1089.00 ± 1.00	2324.00 ± 4.24
Malleability	Double	648.61 ± 2.08	1622.63 ± 1.05

Discussion

The experimental results demonstrate various advantages over the baseline, depending on the situation. In the case of single executions, both malleability and workflows achieved lower resource usage, while their wall times increased slightly. In the case of resource contention, our alternative approaches finished earlier, with the malleable approach up to 1.8x faster than the baseline. The workflow approach was the best one regarding resource usage, while the malleability one represented a good compromise between the baseline and the workflow approach. Our data suggest that our methods were particularly efficient in experiments with longer quantum parts, while their advantage in short quantum job experiments was less pronounced.



REFERENCES

1) Viviani, Paolo, et al. "Assessing the Elephant in the Room in Scheduling for Current Hybrid HPC-QC Clusters." *2025 International Conference on Dependable Systems and Networks Workshops (DSN-W)*. pp. 184-187, 2025.
2) Vercellino, Chiara, et al. "Bbq-mis: a parallel quantum algorithm for graph coloring problems." *2023 IEEE International Conference on Quantum Computing and Engineering (QCE)*. Vol. 2. IEEE, 2023.
3) Scotti, R., et al. "A clustering aggregation algorithm on neutral-atoms and annealing quantum processors". 2024 arXiv. https://doi.org/10.48550/ARXIV.2412.07558

AUTHORS:
Giacomo Vitali, Chiara Vercellino – Fondazione LINKS, Torino, Italy – Politecnico di Torino, Torino, Italy
Marco Cipollini – Politecnico di Torino, Italy
Paolo Viviani, Alberto Scionti, Olivier Terzo – Fondazione LINKS, Torino, Italy
Daniele Gregori, Matteo Barbieri, Gabriella Bettonte, Elisabetta Boella, Fulvio Ganz, Simone Rizzo, Roberto Rocco – E4 Computer Engineering, Scandiano, Italy
Sergio Iserte, Antonio J. Peña, Petter Sandås – Barcelona Supercomputing Center (BSC-CNS), Barcelona, Spain
Jonathan Frassineti, Sara Marzella, Andrea Muratori, Daniele Ottaviani – CINECA, Casalecchio di Reno, Italy
Iacopo Colonnelli – Università di Torino, Torino, Italy

Project funded by the European Union NextGenerationEU, ICSC National Centre, CN00000013, MUR Act n. 1031 - 17/06/2022

SCA/HPCAsia 2026, January 26 - 29, Osaka International Convention Center, 5-3-51 Nakanoshima, Kita-ku, Osaka 530-0005 (JP)