Distributed Quantum Computing workflows with ColonyOS

Quantum computing is evolving rapidly across the globe, and Europe is no exception, with significant investments powering new research frontiers. However, a key challenge lies in efficiently orchestrating complex computations that often require integrating classical high-performance computing (HPC) resources with diverse quantum hardware. Here is an overview on ColonyOS, an open-source meta-operating system designed to address this need by orchestrating distributed computing workflows across heterogeneous environments (including cloud, HPC, standalone workstations, and can be extended towards quantum systems).

The following sections discuss the benefits of such orchestration, highlight key ColonyOS features, and showcase example applications in managing quantum computations. ColonyOS has the potential to streamline hybrid calculations and accelerate the path towards quantum-accelerated supercomputing.

The need for distributed quantum computing

Quantum computing scientists and engineers across different disciplines need to leverage classical and quantum resources in their workflows. They need to develop and test algorithms locally on their personal computers before scaling them up to more complex environments. This iterative process involves expanding the algorithm in terms of parameters, noise models, system complexity, and other details. The next step is to run the workflow on a quantum computer with a few qubits. Of course, due to the limited accessibility to quantum hardware or the limited qubits available, a common target of that workflow is to run it on a powerful computer, which can be an HPC resource capable of simulating hundreds of qubits with the help of some packages and software development kits. Such solutions have the potential to use extensive resources, making it suitable to distribute the workflow over multiple compute resources. For reusability, users require the ability to access those resources seamlessly, saving time by storing profiles of the hardware that this specific workflow can run on across different European compute infrastructures.

Leveraging ColonyOS for distributed quantum computing

ColonyOS is an open-source meta-operating system designed to streamline workload execution across diverse and distributed computing environments, including cloud, edge, HPC, and IoT. More details are provided in the arXiv article. This capability makes it well-suited for managing complex, resource-intensive quantum computing tasks. The software is available under the MIT Licence and can be accessed via GitHub. Comprehensive tutorial notebooks are also available to facilitate onboarding.

Key features of ColonyOS that would help drive quantum-accelerated supercomputing

Distributed microservice architectures

ColonyOS employs a microservices architecture, where independent executors handle specific tasks. This design supports distributed quantum computing by allowing quantum tasks to be executed across geographically dispersed quantum and classical computing resources in a hybrid fashion. Executors can be deployed independently and scaled horizontally, ensuring efficient parallel processing.

Workflow orchestration

The platform enables users to define complex, multi-step workflows across distributed executors. This is particularly beneficial for quantum computing applications, which often require iterative execution of quantum circuits, optimisation steps (e.g., variational quantum eigensolver (VQE) algorithms), and hybrid quantum-classical computations. ColonyOS manages dependencies and execution sequencing, ensuring seamless operation across diverse computational systems.

Scalability

Given the potential for node failures in distributed infrastructures, ColonyOS is designed to reassign tasks dynamically if an executor fails. This approach minimises computation disruptions and enhances overall system reliability.

Platform-Agnostic integration

ColonyOS can operate across multiple platforms, including cloud services and HPC environments. This flexibility aligns with the hybrid quantum-classical infrastructures often required for quantum computing workflows, allowing for efficient orchestration of tasks on both classical supercomputers and quantum processors.

The distributed architecture, task orchestration capabilities, and scalability of ColonyOS make it a powerful solution for managing quantum computing workflows. By leveraging ColonyOS, users can efficiently coordinate tasks across quantum and classical computing environments, accelerating the development and deployment of quantum algorithms that are advancing toward further use in quantum-accelerated supercomputing.

Example: orchestrating and visualising quantum computing workflows with ColonyOS

The following snippets are related to an example where ColonyOS orchestrated Qiskit variational calculations with different noise models. ColonyOS generated

multiple simulations accounting for different noise models. For more implementation details and related code, please visit the blog post here.

ColonyOS serialises Qiskit objects, metrics, and metadata from each part of the workflow into an SQLite database. This database is then exposed to localhost via a simple Flask API, which connects to a React frontend that presents two key views of the results data. Both views display the same data and allow ranking across a set of metrics but do so in different ways:

The first way is through the metrics table—a simple (in-development) table that displays each noise simulation computation along with data from its related variational simulation.

Noise Models Metrics

Select Metrics to Rank By:

☑ Run Time ☐ JS Divergence ☐ Entropy Difference ☑ Optimal Value   [Compute Ranking]

Rank	ID	Type	Run Time	JS Divergence	Optimal Value	Entropy Difference	Noisy Entropy	Noiseless Entropy
1	b57cda33-396f-4bb5-bc22-64eff667e06d	noise	2.96	0.49	N/A	0.19	9.70	9.51
2	58e02057-4937-4cca-b0c7-17bb7a19aba3	noise	3.06	0.57	N/A	0.13	9.82	9.69
3	0205470c-d530-496e-9b38-34bd7b41973a	noise	3.07	0.50	N/A	0.21	9.72	9.51
4	99d03e58-949e-4591-969f-8007f66530ff	noise	3.11	0.48	N/A	0.26	9.57	9.32
5	6a0b230b-84f3-430e-a4e7-4371a645a9a3	noise	3.12	0.55	N/A	0.14	9.77	9.63
6	df76dbbf-1e14-4e0c-95c0-60e079453a21	noise	3.12	0.57	N/A	0.09	9.85	9.76
7	7f34b329-beb6-4d51-87fe-a0acf0b67bb5	noise	3.16	0.51	N/A	0.32	9.64	9.32
8	fc879d70-1f68-487a-8e2d-d06b6e0d61a7	noise	3.21	0.56	N/A	0.03	9.80	9.76
9	776c0981-5f93-4675-9e3d-3437f6bb32f3	noise	3.38	0.53	N/A	0.16	9.78	9.63
10	65c60a75-6c50-482e-b369-aafd48bb2518	noise	3.57	0.59	N/A	0.06	9.84	9.78

Figure 1: The ColonyOS metrics table displaying results from variational simulations with different noise models.

The second way is through a workflow graph showing how each step in the workflow is connected and which steps depend on its information.
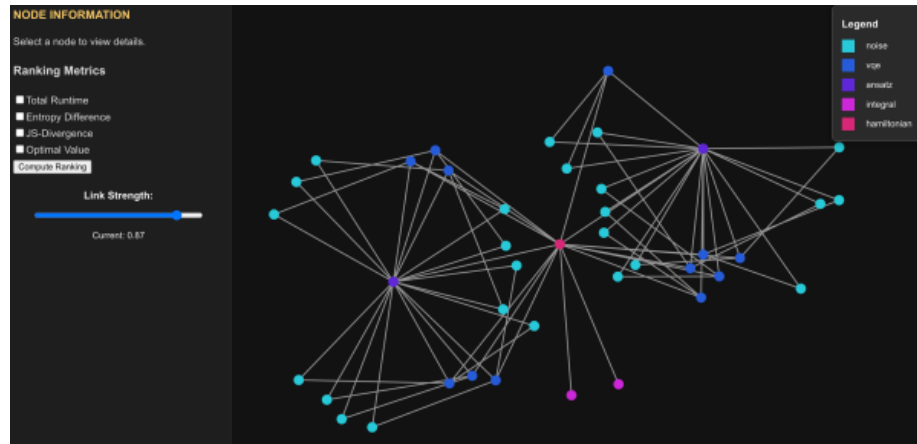

Figure 2: The ColonyOS workflow graph visualising the connections between

different computation steps.

Here, the legend explains which part of the calculation workflow the nodes correspond to. A node information panel displays metrics of the selected node. It allows one to compute rankings across nodes (similar to the metrics table) while rescaling and labelling nodes as a function of rank, as seen here:
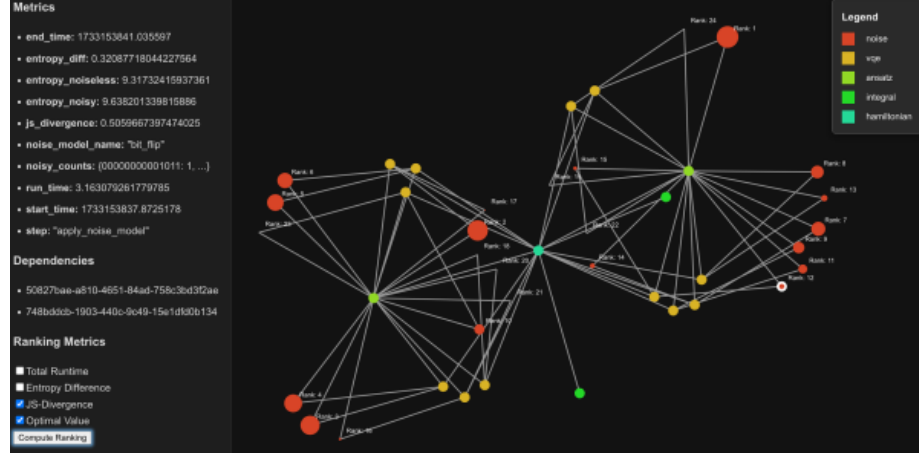

Figure 3: The workflow graph ranked based on specific metrics, highlighting performance differences.

With more complicated systems and calculations, the database could present a denser graph providing easily searchable sets of data.

─────────────────

These examples highlight ongoing efforts to integrate ColonyOS into quantum computing workflows—a promising step toward distributed computing orchestration. This work also leveraged graph data analytics to view and analyse quantum computation outcomes. ColonyOS, as part of the European Compute Continuum Initiative, could become a vital part at the orchestration layer for the EuroHPC-JU hybrid quantum-classical infrastructure, enabling seamless utilisation of resources like LUMI-Q and MareNostrum Q with the existing HPC classical resources. To explore ColonyOS further, check out the GitHub repository and the available tutorials.

─────────────────

Acknowledgement

This blog post is based on work by Erik Källman, first presented at the Nordic Quantum Autumn School 2024 and further elaborated in an accompanying blog post here. The original content can be found in Erik's blog. The concepts and implementations have been adapted and expanded with permission to showcase the potential of ColonyOS in distributed quantum computing workflows. We

---