

Novel Distributed High Performance Computing Solution

Philipp Hickisch¹, Dennis Hohlfeld¹

¹University of Rostock

Abstract

When working with large data sets or optimization problems, substantial computational power is required. Computing hardware capacity often limits model size and simulation speed. Although commercial and open-source cluster software solutions exist, they are typically either costly to license or cumbersome to deploy and maintain, and they offer limited flexibility. To address these limitations, we propose a fully automated, distributed architecture that employs a control server and an arbitrary number of heterogeneous client compute nodes.

Conventional cluster systems commonly rely on job schedulers (e.g., SLURM) and message-passing interfaces (e.g., MPI). These frameworks demand high-bandwidth interconnects such as InfiniBand, which increases cost and operational complexity. While they enable a single model to be distributed across multiple nodes, the networking and synchronization overhead can be prohibitive. In contrast, our method adopts a lightweight client-server model. Using the COMSOL Application Builder, each client node autonomously communicates with the control server and solves assigned models. After initialization, nodes request new jobs from the server automatically. Once a job is completed, the results are sent back to the server. This workflow allows for dynamic scaling of compute resources and ensures that failed jobs can be requeued and rerun at a later time. The control server software is written in Python and is open-source. It manages node connections, maintains a job queue, and stores results in a database. Users submit jobs by specifying parameter sets for a given model, which are enqueued for distribution to the cluster.

To demonstrate the framework, we apply it to the optimization of a water-cooled heat sink for power electronics. The design domain is given by an array of square cells which are randomly assigned fluidic ($p = 0.7$) or solid ($p = 0.3$) properties. A subsequent geometry postprocessing converts this geometry into smooth and continuous flow channels. A temperature boundary condition is imposed on all walls. The optimization objective is to maximize the net heat flux at the outlet. Meshing and solving each candidate design take several minutes (160 000 DOF), but distributing thousands of independent evaluations across a cluster reduces the overall time required for the optimization, from days on a single workstation to mere hours, underscoring the scalability of the approach.

The principal advantage of this approach is its flexibility and ease of use. The number of compute nodes can also be adjusted during cluster operation. The system can utilize CPU-Locked, Named Single-User, and Floating Network licenses. Multiple operating systems and COMSOL versions can run concurrently. The only requirement for client nodes is HTTP-connectivity to the control server, enabling effortless scaling of hardware resources and improving simulation throughput and efficiency.

Figures used in the abstract

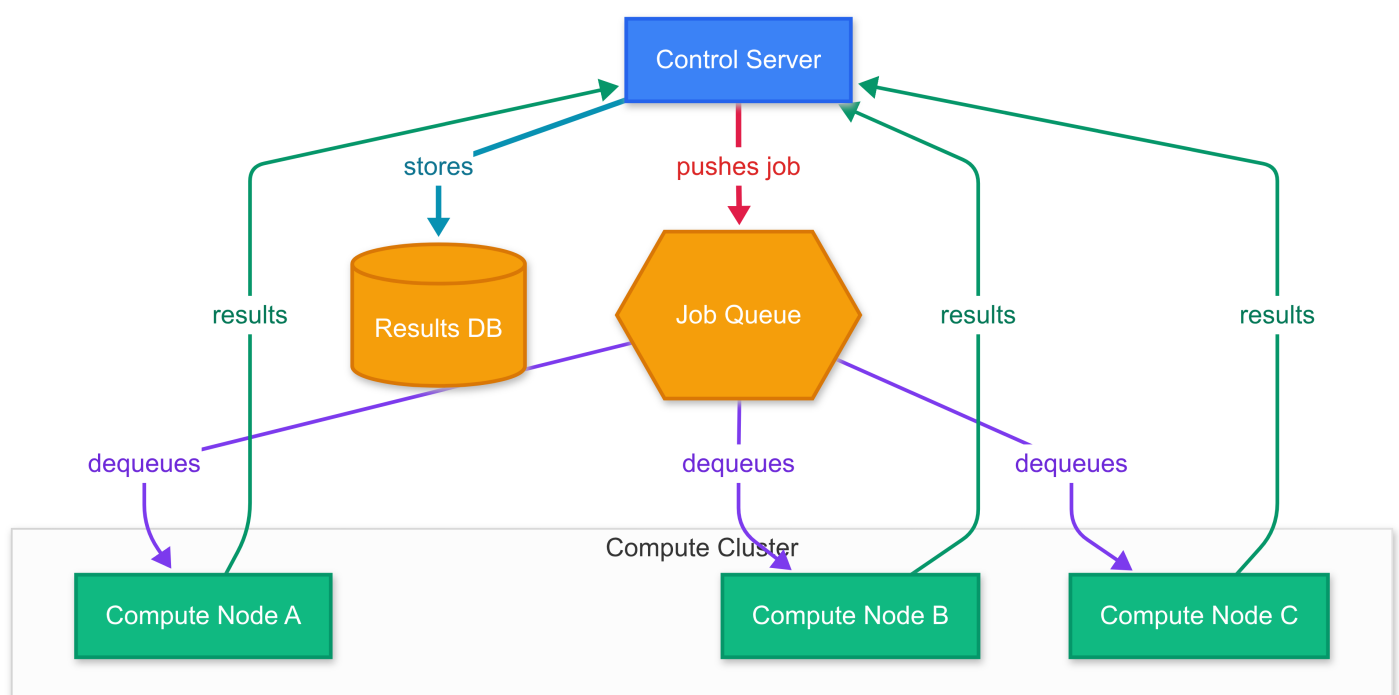


Figure 1 : Distributed pipeline: the Control Server enqueues jobs to a central Job Queue; Compute Nodes dequeue and process tasks; results are stored in the Results DB and reported to the Control Server.

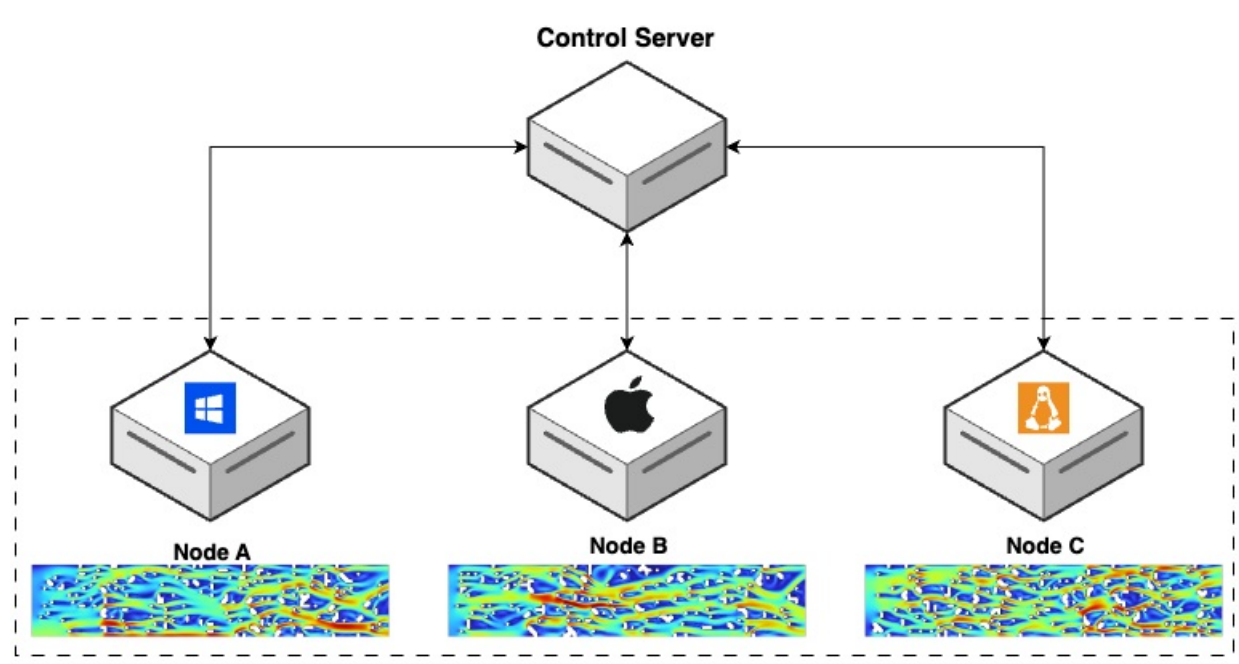


Figure 2 : Figure 2. A three-node cluster with heterogeneous operating systems solves an optimization problem of a computational fluid dynamics (CFD) model.

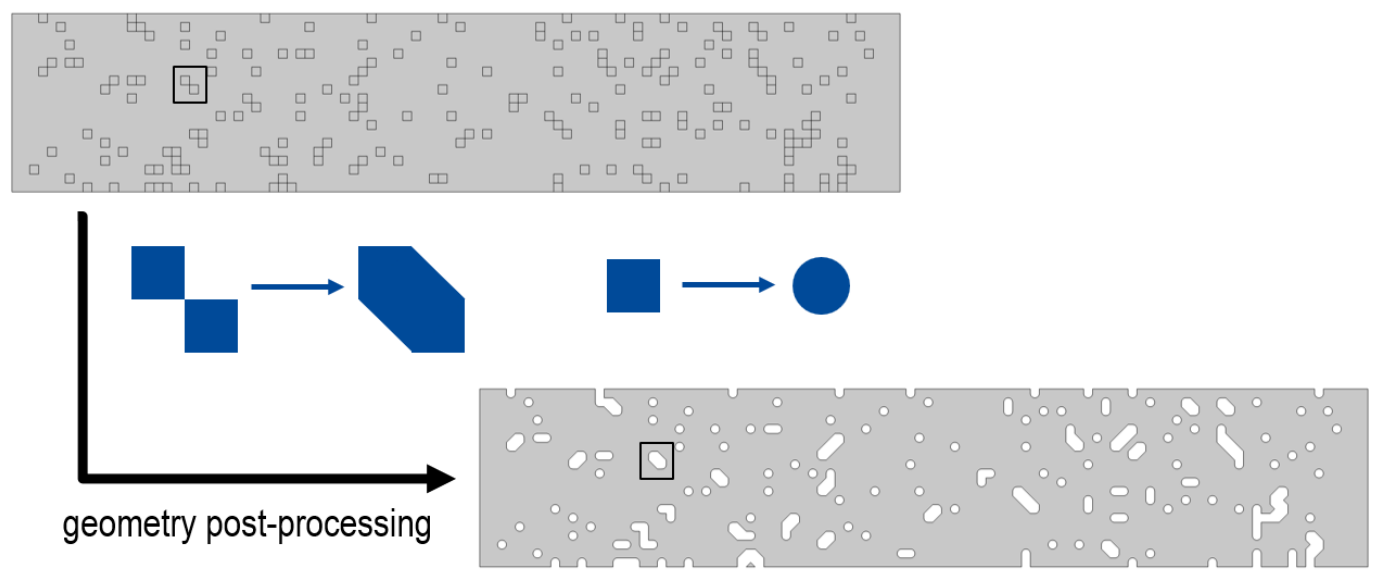


Figure 3 : Figure 3. Arbitrary grid pf solid and fluid 'pixels' is prepared for fluidic simulation, Top: Original geometry, Bottottom: optimized geometry