System for Orchestrating and Monitoring Jobs in Personalized Medicine

Author: Oleksandr Musiichuk

Supervisor: Ing. Marta Jaroš, Ph.D.

Faculty of Information Technology, Brno University of Technology, CZ

1 Motivation

Personalized medicine requires treatment planning tailored to each patient in a specific time and context.

- Treatment is simulated as a workflow (task graph)
- Executed on HPC clusters using schedulers
- Schedulers allocate resources based on execution parameters
- These parameters affect time, cost and throughput

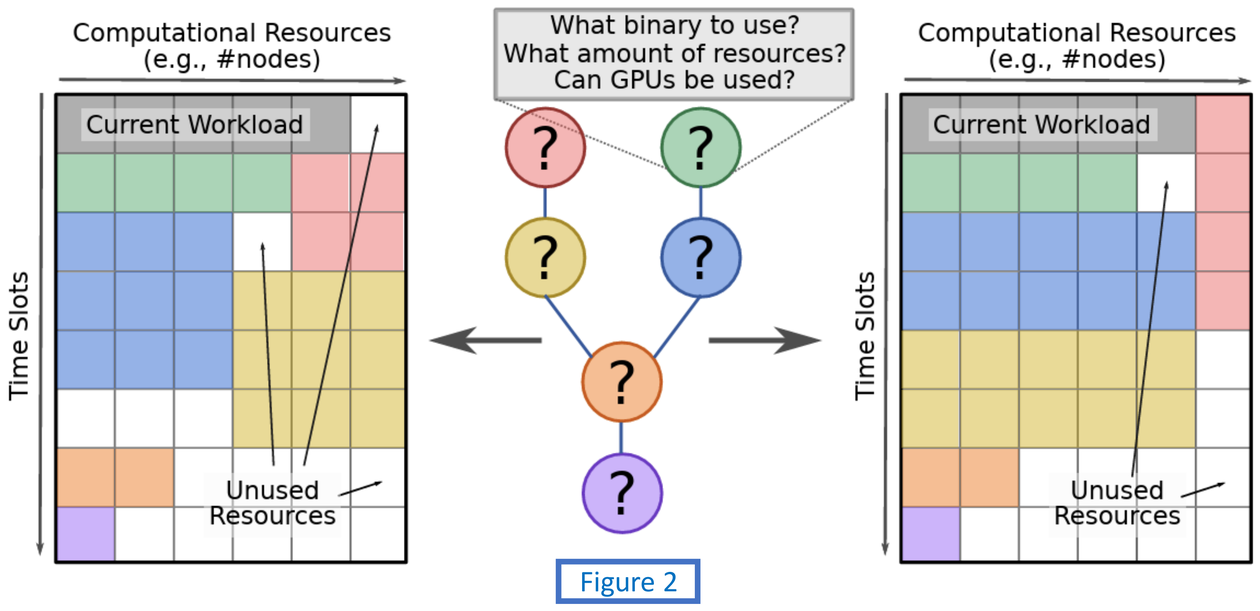


Figure 2

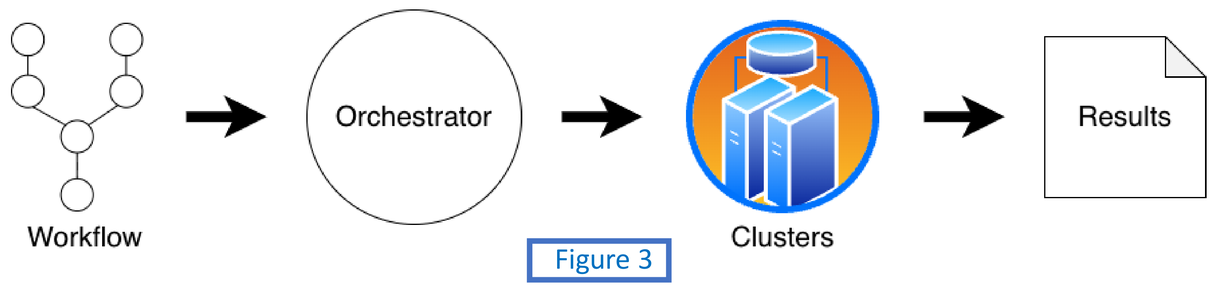


Figure 3

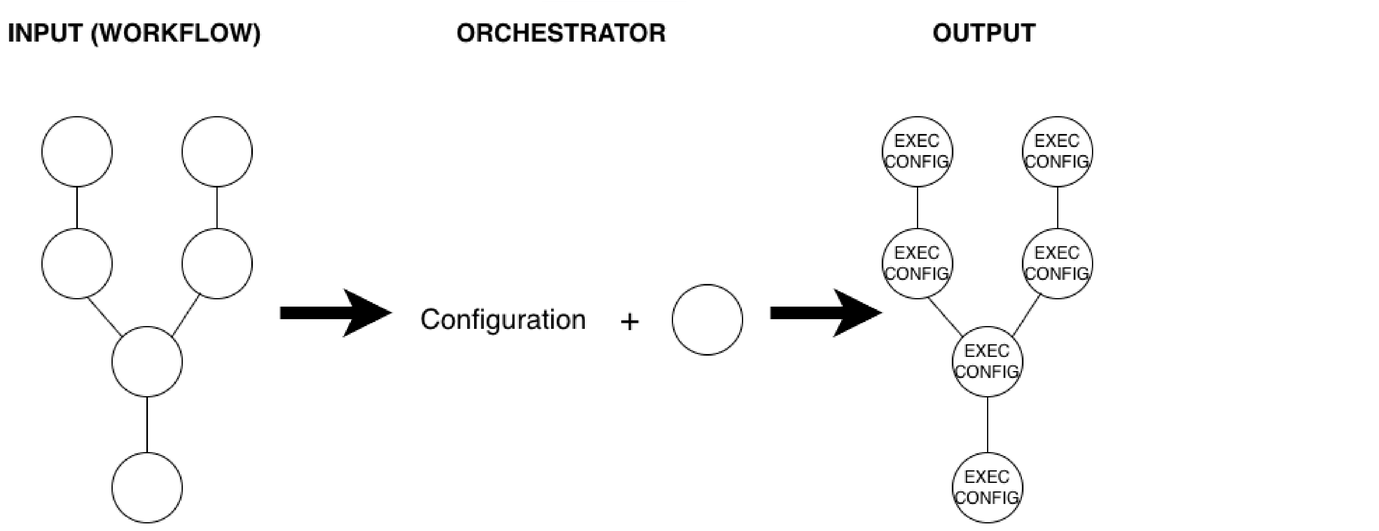
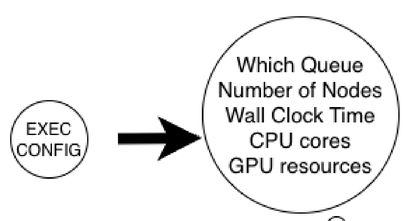


Figure 4 Different execution configurations (CPU, nodes, walltime)



4 Conclusion And Future Work

We designed a modular orchestrator for HPC workflow execution, bridging workflow definition and execution.

Benefits:
- flexible execution configuration
- modular and extensible design
- support for integrating optimization strategies

Future work:
- advanced optimization strategies
- priority-based scheduling
- evaluation on complex workflows

- More computing resources (nodes) → shorter execution time
- **Task scaling is never perfect → growing cost**
- cost = execution time * number of nodes

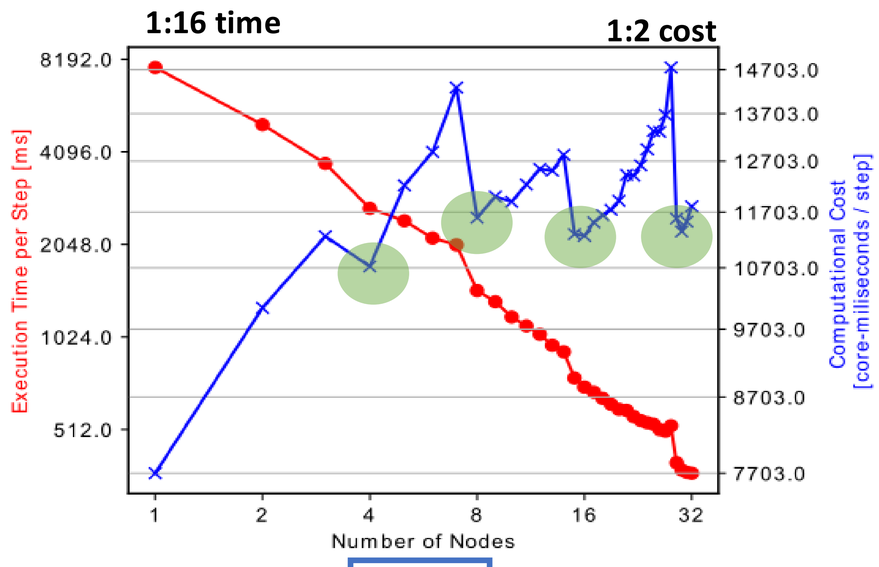


Figure 1

2 Problem

Treatment simulation workflows consist of dependent computational tasks that must be executed on HPC clusters.

Each task requires execution parameters:
(CPU, nodes, walltime)

These parameters affect:
- runtime
- queuing time
- resource usage
- scheduling efficiency and start-time predictability

Problem:
- hardcoded configurations
- no flexibility
- no integration of optimization strategies

3 Approach & Architecture

We model execution as a lifecycle of a workflow task:

1. A task is created as part of a workflow graph (Dependency Graph)
2. After dependencies are resolved, the task becomes ready
3. The orchestrator assigns:
   - executable binary
   - execution parameters (CPU, nodes, queue, walltime)
4. The task is transformed into a job script
5. The job is submitted via Remote Machine interface to HPC scheduler
6. Execution is monitored and results are collected

The orchestrator acts as a central decision-making module, connecting workflow definition with execution.

It integrates:
- Dependency Graph (structure)
- Remote Machine (execution)
- Optimizer (optional strategy selection)

The module is generic and extensible, allowing integration of different execution strategies.

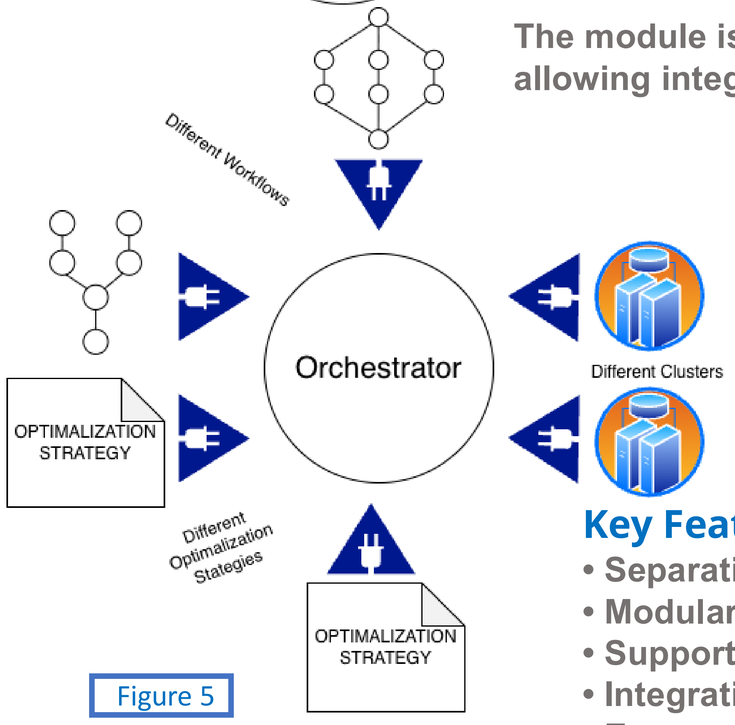


Figure 5

Key Features

- Separation of workflow logic and execution logic
- Modular design (multiple orchestrator implementations)
- Support for different execution configurations
- Integration point for optimization strategies
- Easy extensibility without modifying core system

Excel@FIT 2026

Acknowledgement

I would like to thank my supervisor Marta Jaroš for guidance and for providing access to the k-Dispatch codebase and HPC infrastructure at FIT BUT.