Optimizing Biomedical Ultrasound Workflow Scheduling Using Cluster Simulation

Marta Jaros[1], Dalibor Klusacek[2], Jiri Jaros[1]

[1] Brno University of Technology, Faculty of Information Technology,
Centre of Excellence IT4Innovations, Brno, Czech Republic
{martajaros, jarosjir}@fit.vutbr.cz
[2] CESNET a.l.e., Brno, Czech Republic
klusacek@cesnet.cz

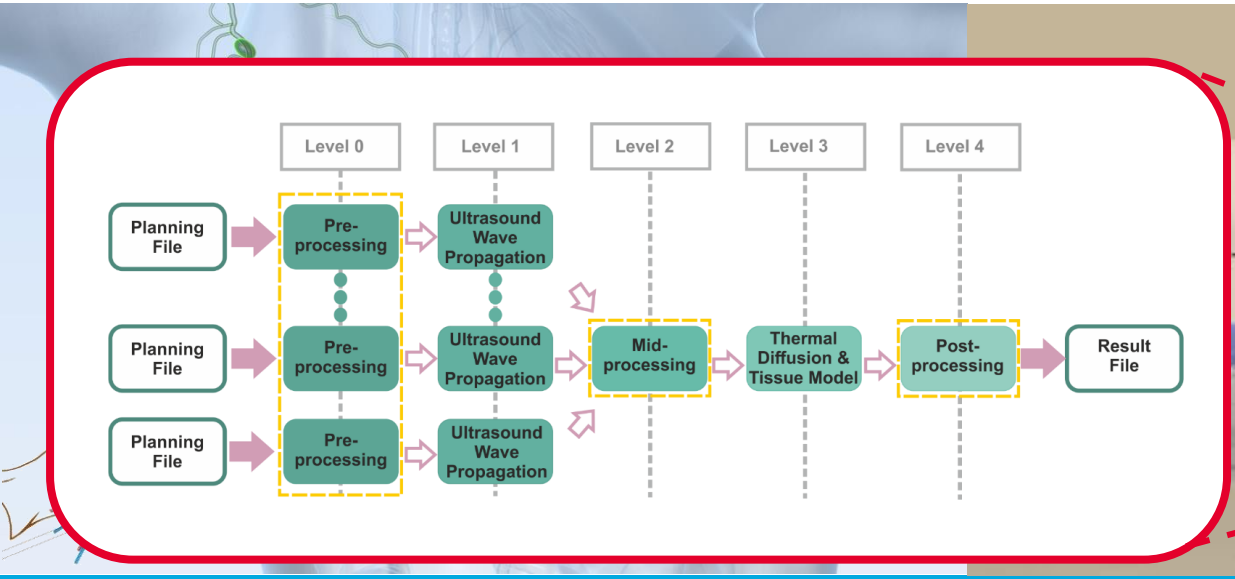
SUPERCOMPUTING TECHNOLOGIES GROUP

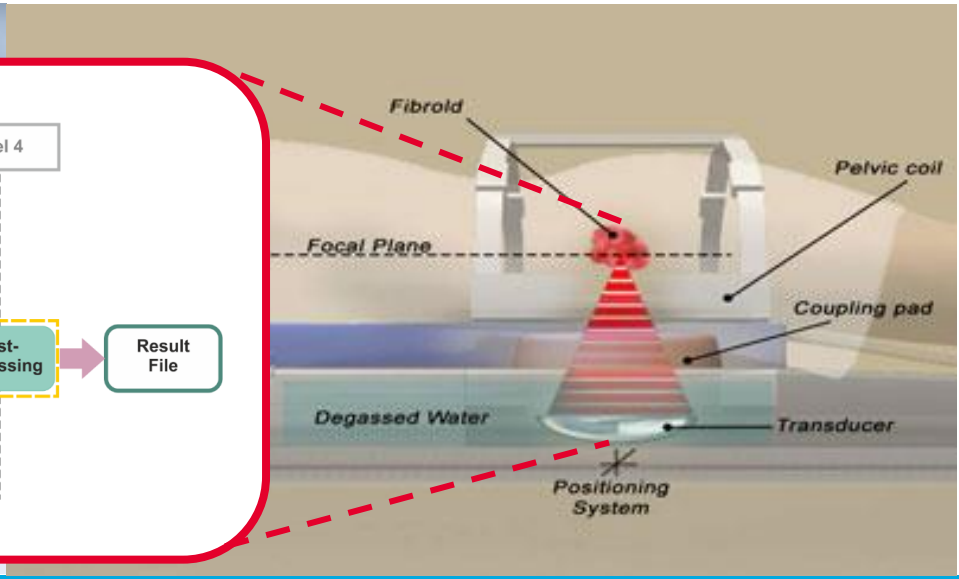
BRNO UNIVERSITY OF TECHNOLOGY FACULTY OF INFORMATION TECHNOLOGY

cesnet

Motivation / Case Study

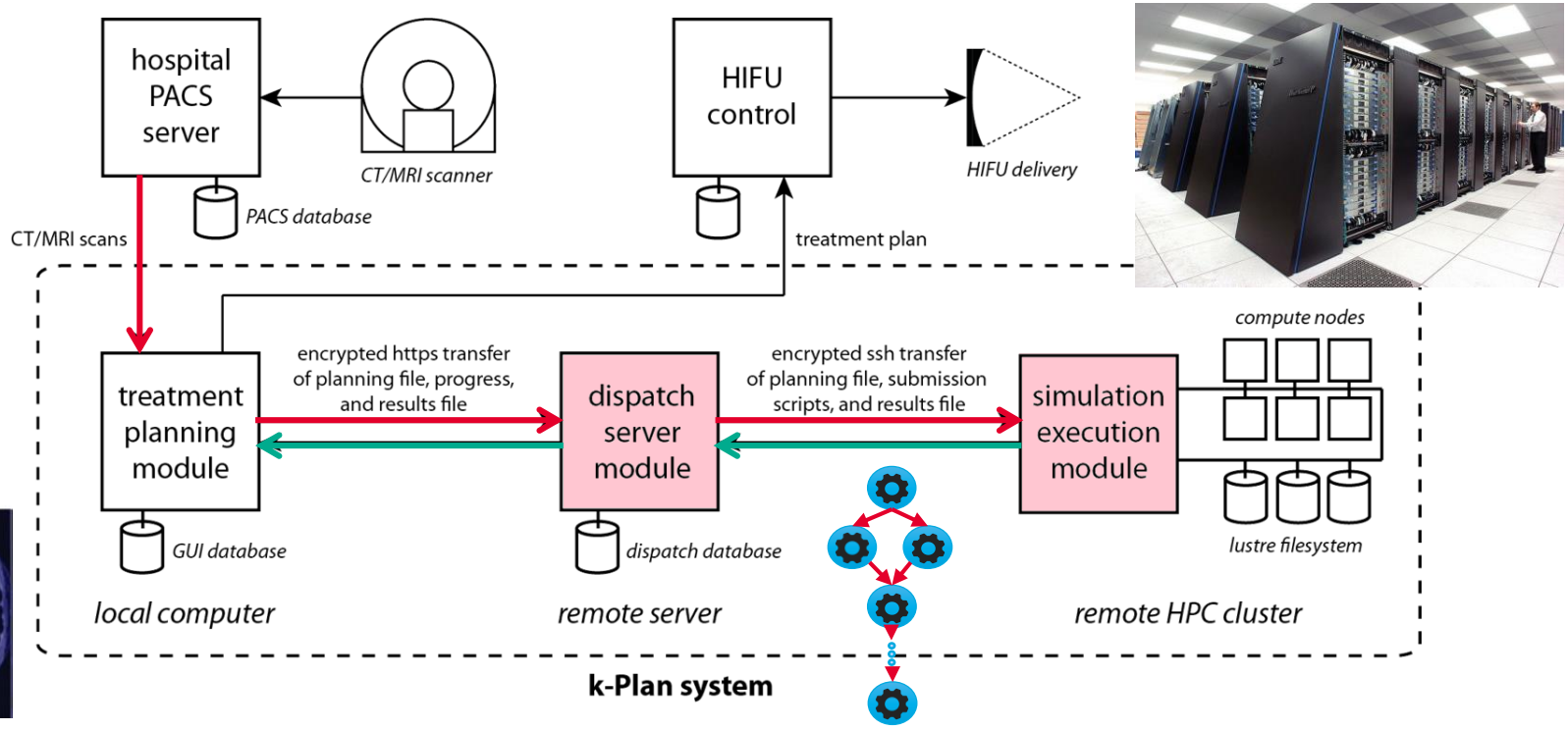
- Ultrasound treatment planning
- Examples of medical applications:
  - Surgery planning
  - Targeted drug delivery
  - Neurostimulation



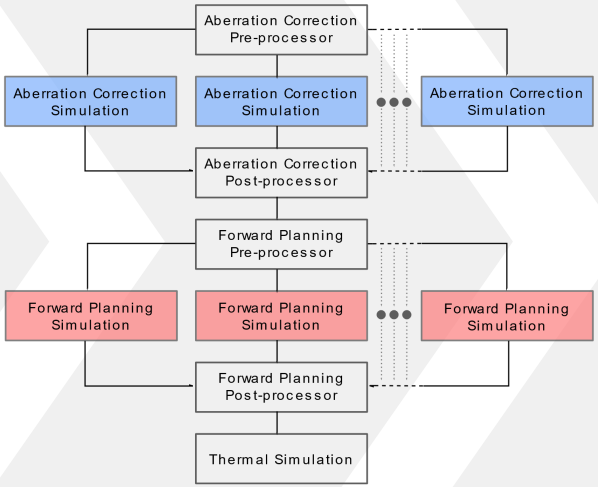


k-Dispatch Architecture

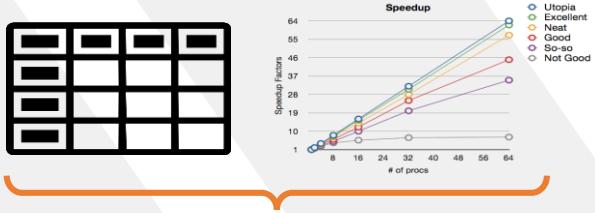
- Execution Planning
- Workflow Execution
- Data Exchange Between Tasks
- Cluster and Tasks Monitoring

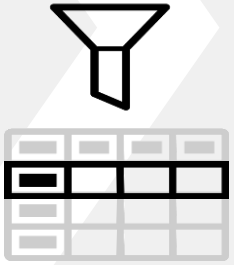


Provide an **automated**, **effective** and **failure-free workflow execution.**

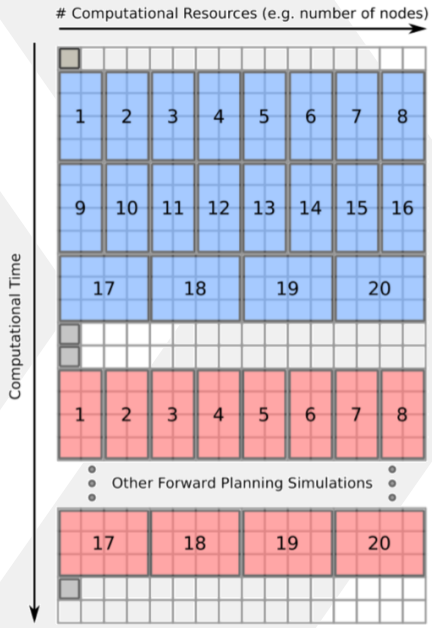


Performance Data

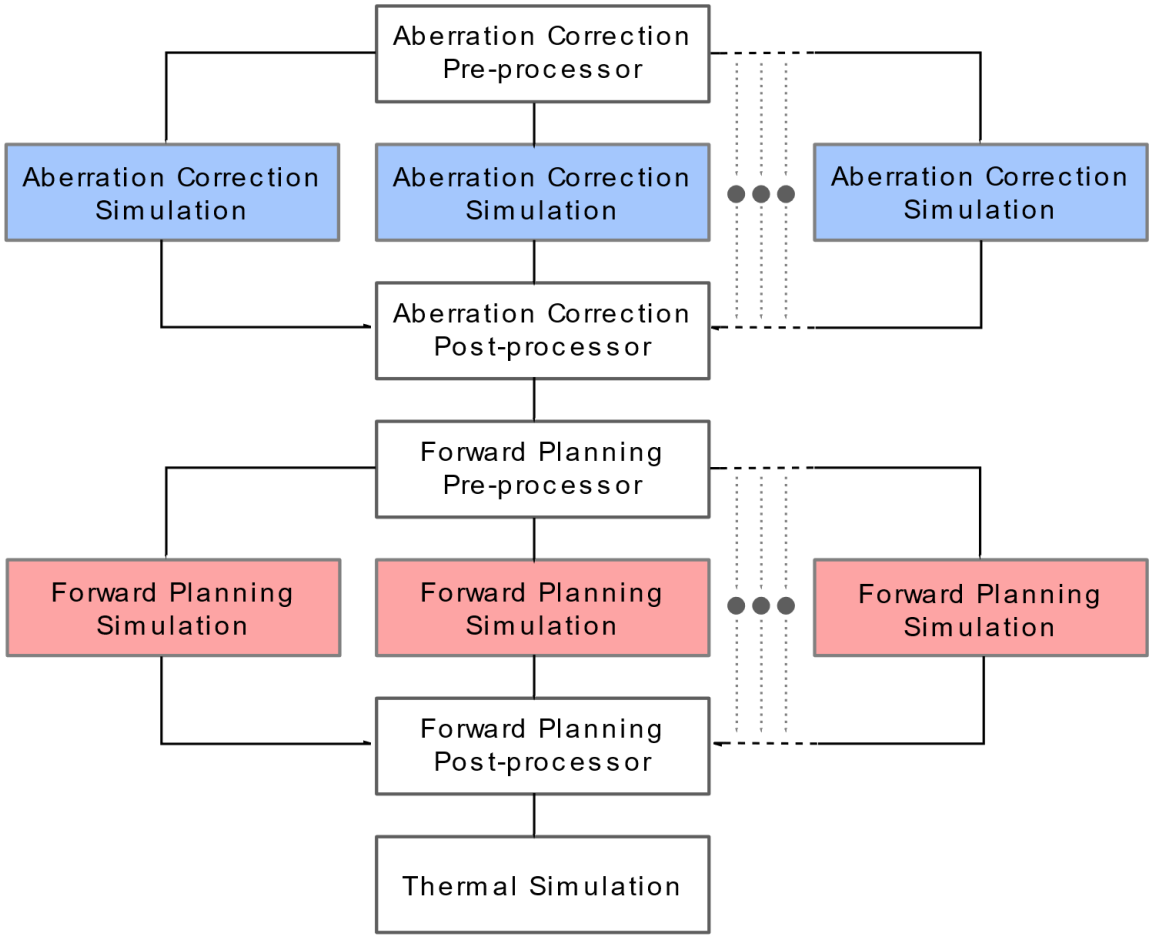




- Select the executable
- Select the execution parameters (HPC, # CPUs, HPC queue, ...)



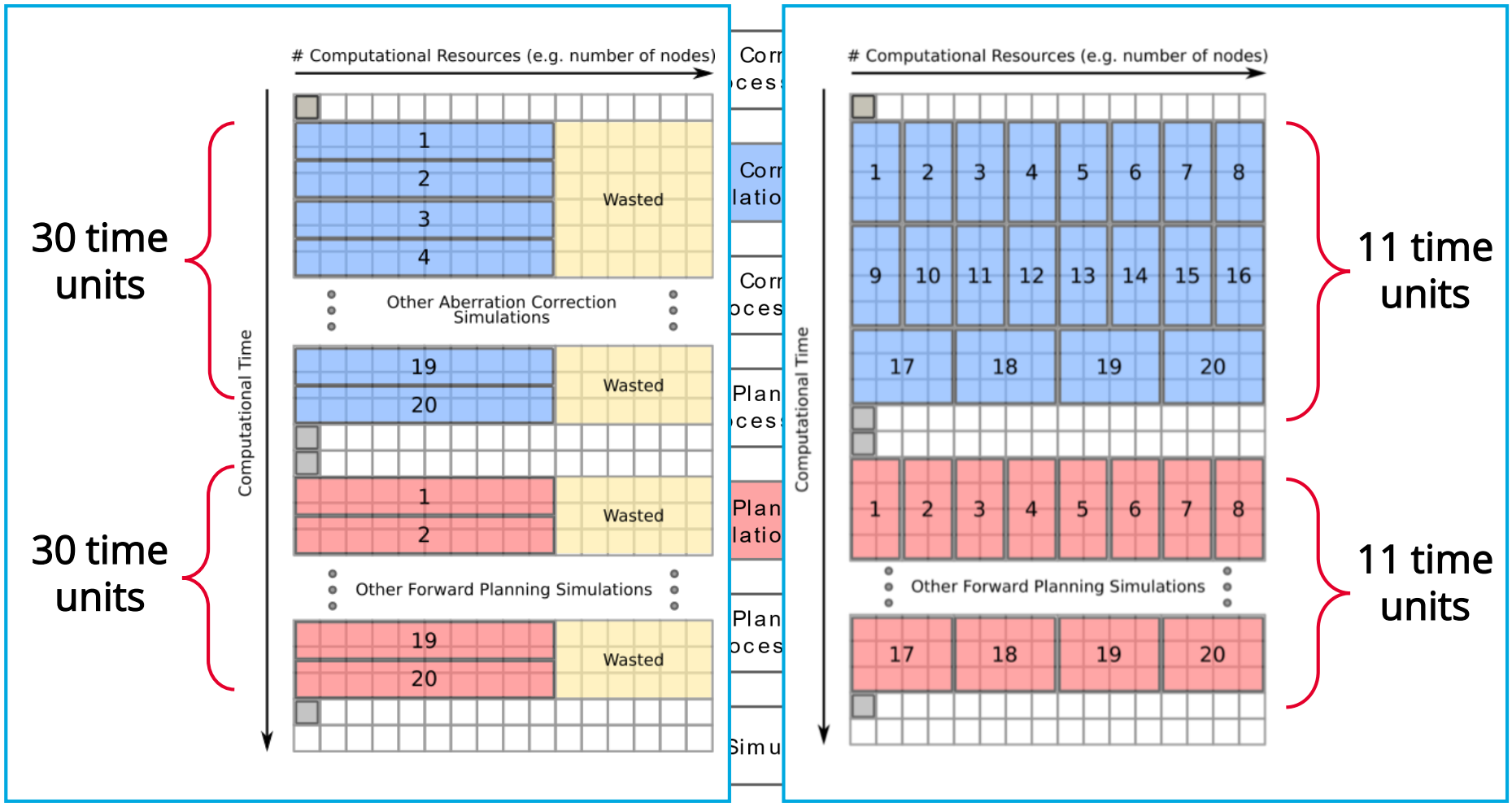
Planning Challenge: Effective Execution Plan



Planning Challenge: Effective Execution Plan

Example #1

Example #2



Task-level optimization

Workflow-level optimization

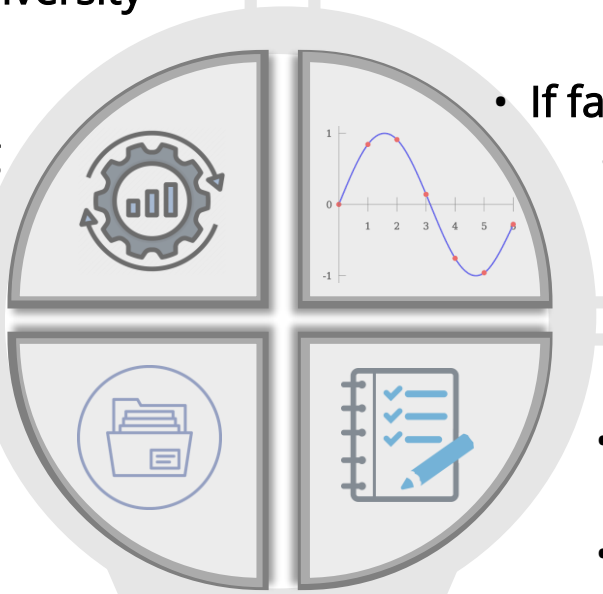
k-Dispatch Performance Modules

Optimizer

- Finds execution parameters based on task inputs
- Increases performance data diversity by small perturbations of the execution parameters
- Can be based on Hill Climbing or Simulated annealing

Interpolator

- Estimates the execution time and cost for a given amount of resources
  - Uses linear and cubic-spline interpolation
- If fails, maximum execution time and associated cost are used



Collector

- Updates performance data in the database after each successful run
- Provides feedback to the optimizer after successful run
- Adapts optimizing process to the cluster workload variations

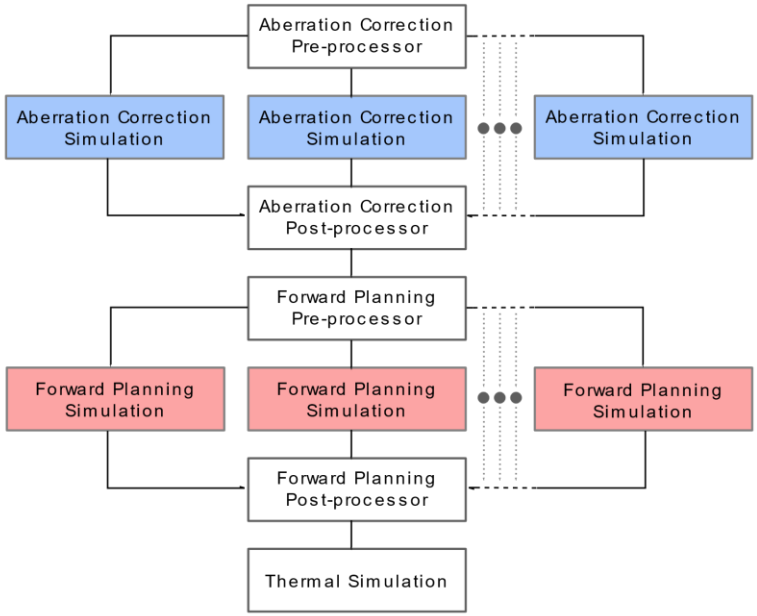
Evaluator

- Calculates the makespan (execution + queueing time)
- Runs the simulator (ALEA)

ALEA

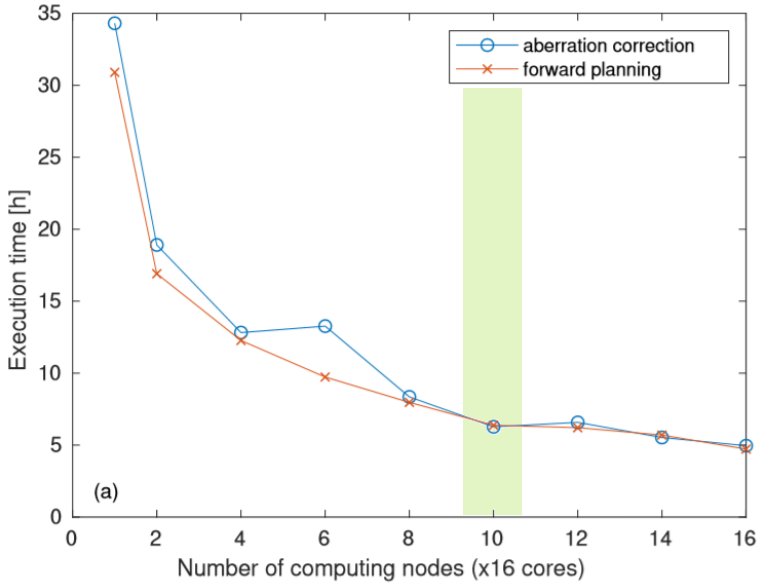
- Calculates the final quality criteria:
  $f = w * (t + q) + (1 - w) * c$

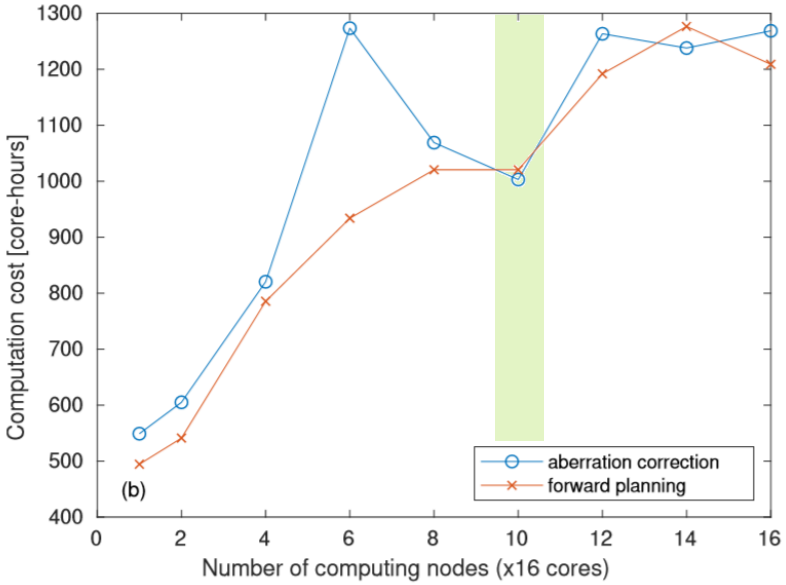
Experimental Setup



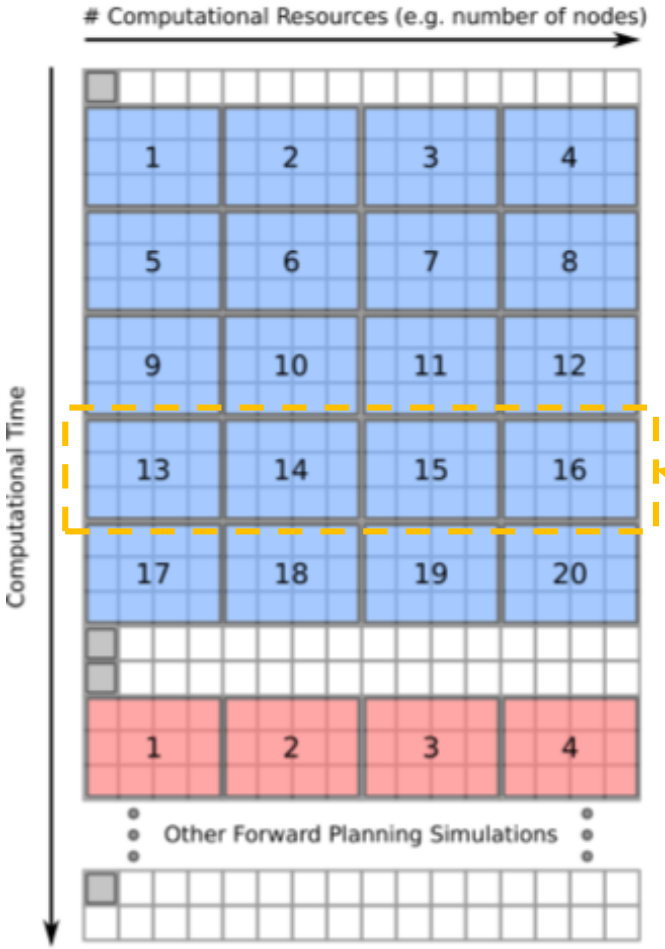
MiniAnselm:

- one computational queue
- 16 nodes of:
  - x86-64, 16 cores, two sockets (2x8 cores Intel Sandy Bridge E5-2665, 2.4 GHz), 64 GB RAM
- Infiniband QDR, full blocking fat tree
- Anselm at IT4Innovation (https://docs.it4i.cz/anselm/hardware-overview/)

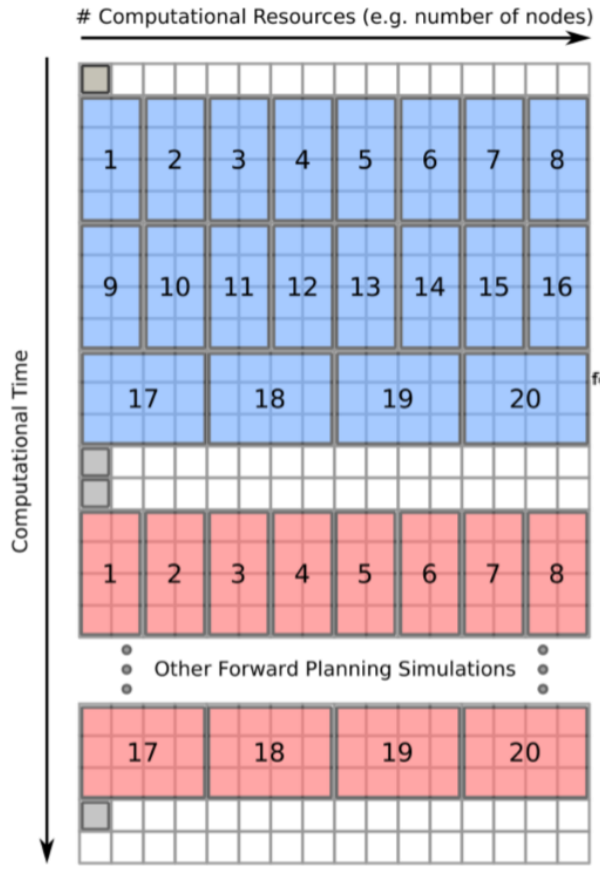


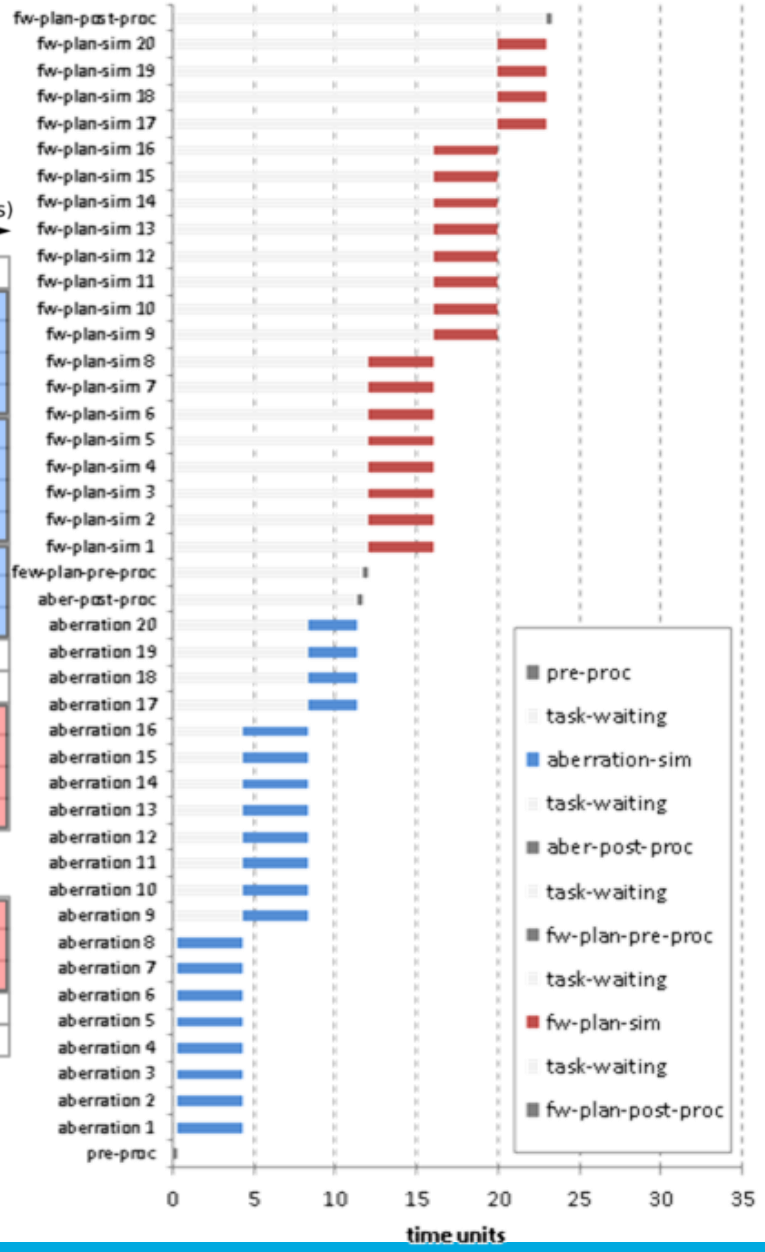


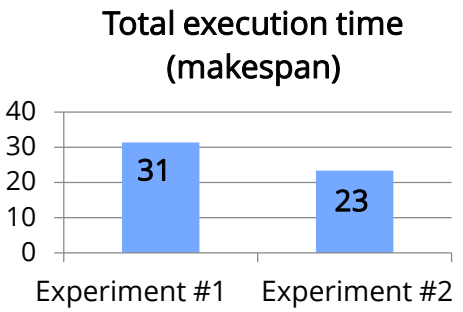
Experiment #1

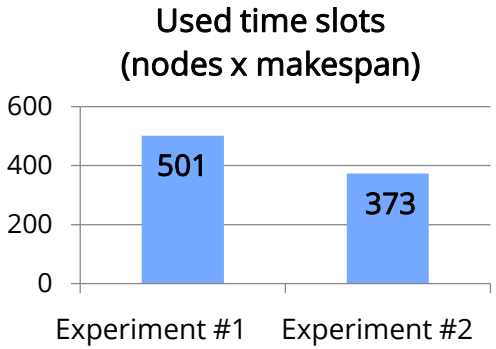


Experiment #2 + Results









Experiment #2 is of 26 % better (cheaper/faster).

Execution Planning Challenges

- **Optimizer:**
  - Selection of a fast algorithm (1 minute execution)
  - Adaptation on a changing workload
  - Multi-workflow optimization
- **Interpolator:**
  - Data filtering (anomalies, age, missing data)
  - Reliable interpolation in noisy data (oscillations)
- **Evaluator:**
  - Cluster workload snapshots
  - User priorities and backfilling parameters
- **Implementation challenges**:
  - Coupling Java based ALEA with Python based k-Dispatch

Conclusions

- Workflow planning based on performance data has proven the ability to save significant amount of computation time and cost (up to 26 % for a simple benchmark).
- New architecture composed of four modules has been introduced:
  - Optimizer - in a prove of concept stage.
  - Interpolator - partially implemented and being tested on large data sets.
  - Estimator - first attempt to integrate the ALEA simulator.
  - Collector - fully implemented and routinely used for a data collection.
- Future work:
  - Evaluation of different optimizer strategies.
  - Deployment on real systems with real workflows.

Questions?
Thank You for Your Attention