Optimisation of the HPC cluster using SLURM simulator and AI

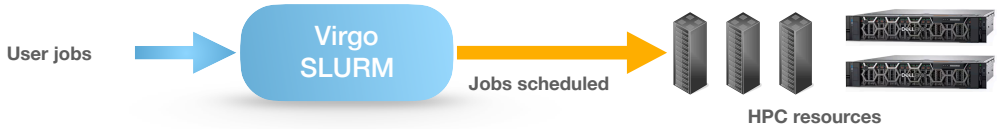
Saidev Polisetty

31.08.2026

2nd GSI/FAIR AI Workshop

Introduction

- SLURM is a cluster and scheduling management batch system.



  - The whole architecture is a well synchronised infrastructure depending on each other for a smoother workflow.
- The primary factor of determining this efficient scheduling and execution depends on how well SLURM is configured.
- Optimising/testing SLURM parameters on running cluster is not advised to avoid cluster breakdown.
  - Additionally, user jobs typically range from hours to 7 days (longer test periods).

SLURM Simulator and AI

- Alternate option: SLURM Simulator, as a tool to test setups for production cluster in shorter time and safer environments.
  - Entire workload of cluster can be tested without actually running the workload for real time.
  - Workload worth of 10 days on actual cluster takes approximately a day on simulator.

- SLURM simulator simulates workload and scheduling of SLURM.
  - Simulates job execution on defined cluster config and uses virtual time skipping.
    - Mimics actual job as sleep, instead of running and time is skipped.
  - Simulator can be used to test cluster setup and optimise performance by tuning the SLURM configuration.

- Additionally as separate tools, AI can be used for optimising parameter values.
  - LLM can spot potential areas for improvement in configuration and cluster.
  - OpenEvolve is planned for iterative simulation run to realise tuned parameters suitable for the HPC cluster.
  - It is an open version of Google’s AlphaEvolve.

Optimising using OpenEvolve

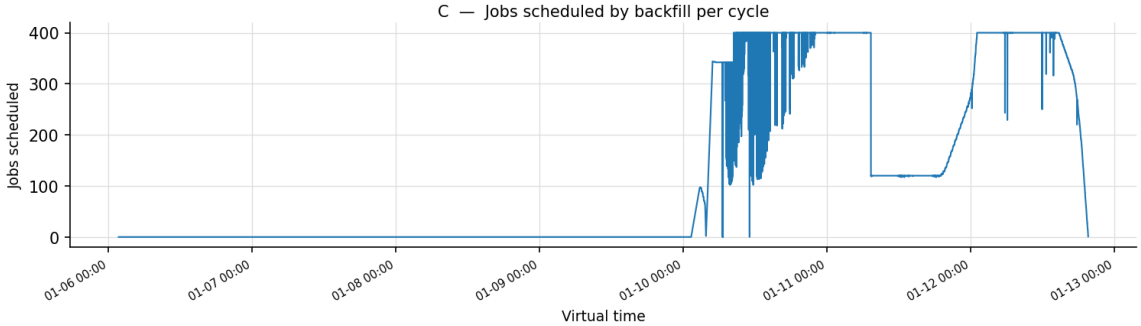
- OpenEvolve maintains a diverse population of candidate solutions and iteratively improves them based on an evaluator-provided fitness score.
- The configuration is run through the AI to find the optimal parameters.
  - Generate → Evaluate → Select → Evolve
  - ← (loop back)
  - Workflow as follows:
    - Generate and evaluate the SLURM configuration.
    - Run the workload on the simulator
    - Performance metrics such as throughput, wait time, utilisation, … measured
    - Fitness score given and next generation is run repeating the whole process.
- Current: It is aimed to use OpenEvolve to run iterative loops to optimise the configuration for the HPC cluster.

Optimising scheduler

- Slurm control (Slurmctld) is the main working daemon - schedules, monitors … jobs in the simulator.
- Daemon responsible for job execution is disabled in the simulator.
- Main scheduler schedules the jobs and backfill scheduler fills low priority jobs in the gaps.

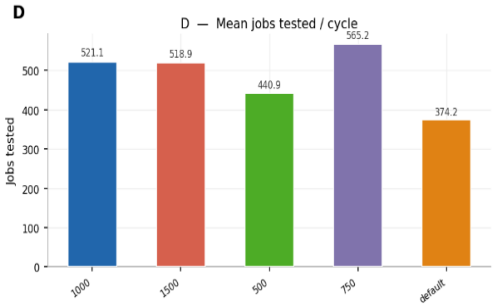
- Current parameters on Virgo 3/4 limits to a maximum jobs of 400 per cycle.
  - default_queue_depth: # of jobs to be considered for scheduling - 400
  - bf_max_job_start: max # of jobs to initiate per bf cycle - 400
  - bf_max_job_test: max # of jobs to be considered per bf cycle - 400
  - max_rpc_cont: for calls between requests - 100

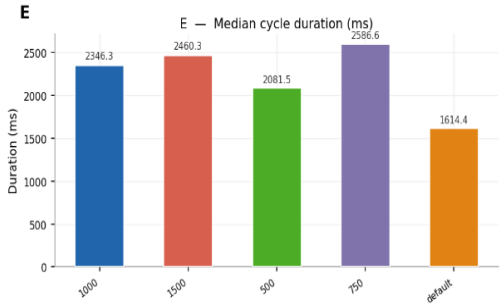
Current setup

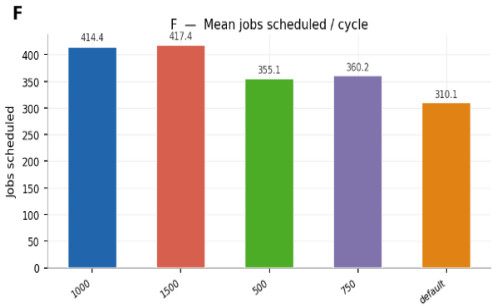


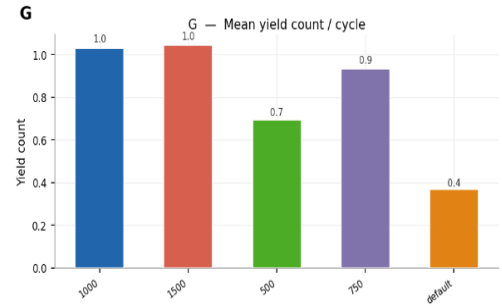
- With the current configuration, backfill scheduler hits limit under the regular operation of the 100k CPU HPC workload.
- The full extent of the HPC not completely attained.
- The backfill scheduler could test/schedule more jobs in parallel reducing the wait time.

Tuning of parameters









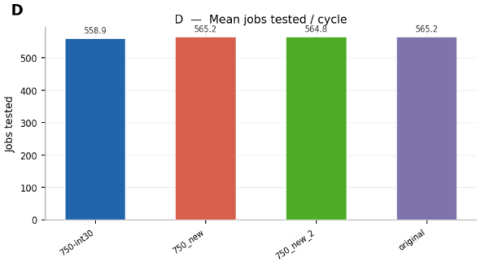
- With the default values, the scheduler is cut off to attain max. eff.
  - Allowing higher BF limits show more jobs being tested/scheduler as backfill.
  - Very large values can saturate the machine and increase latency.
  - Optimal values are needed.
- The ideal value of parameters provide low yield and test/schedule more jobs to reduce latency on the manager.

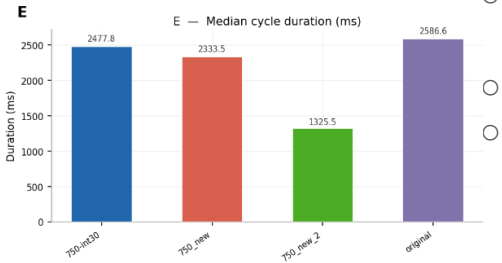
Optimising the configuration

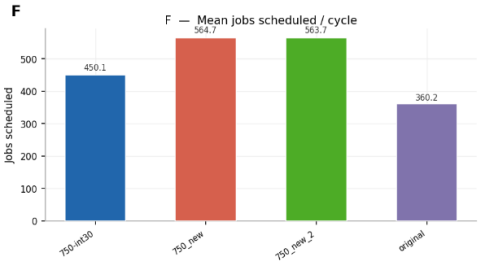
- Additional backfill parameters tested for increasing efficiency
  - To periodically releases locks
  - Number of mins into the future to look when considering jobs to schedule
    - Higher value -> less overhead and better responsiveness
    - Too high results in node starvation
  - Number of secs in the resolution of data maintained about when jobs begin and end
    - Higher values result in better responsiveness and quicker backfill cycles by using larger blocks of time to determine node eligibility
    - Higher value -> less efficient system planning
  - Number of secs between bf interactions
    - Higher value -> less overhead and better responsiveness
- The backfill scheduler could test/schedule more jobs efficiently and reduce the waiting time.
- In addition, Backfill scheduler has a large list of parameters for use.

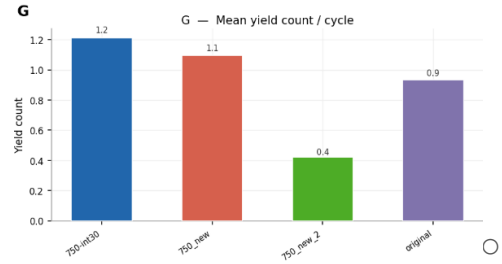
Tuning of parameters continued

- The limit with 750 showed promising results.
  - Further optimised with additional parameters.
  - More jobs are scheduled on using bf_window and bf_resolution parameters.
  - Yield count improved.
  - Further tuning is planned using OpenEvolve.

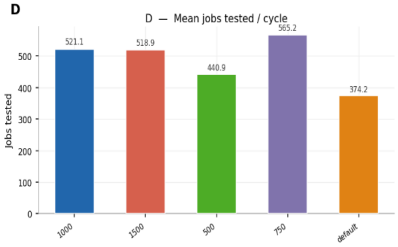


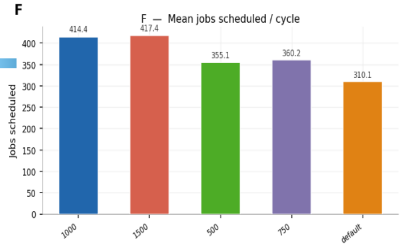






- Compared to old simulations better results.





Outlook and summary

- SLURM simulator is a handy tool for optimising HPC clusters.
- Various parameters can be tested on real workload in short periods.
  - Current tests already boost the performance.
- Traditional testing of various parameters via simulator can be further sped up and increase optimisation by usage of AI tools.
  - LLM models can speed up testing various parameters in iterative loops and provide optimal parameter values to use.
  - OpenEvolve is planned to use to test scheduling parameters for the HPC cluster.

Back up

Additional info

- Current available slurm version - 23.11 (2 years behind current slurm version).
- https://github.com/ubccr-slurm-simulator/slurm_sim_tools
- https://github.com/ubccr-slurm-simulator/slurm_simulator

Backfill cycle structure