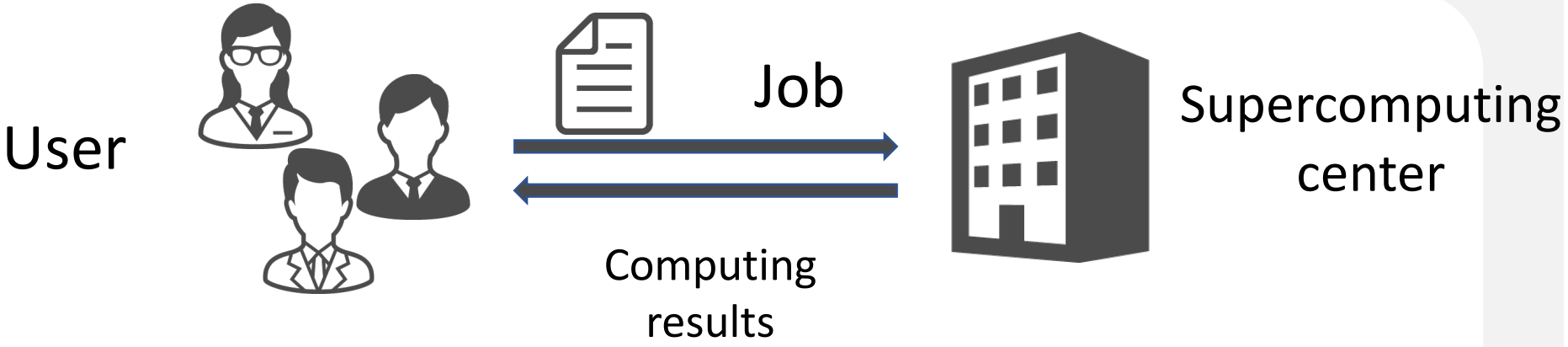
Job Scheduler Simulator Extension for Evaluating Queue Mapping to Computing Node

Susumu Date, Yuki Matsui, Yasuhiro Watashiba, Takashi Yoshikawa, Shinji Shimojo

Cybermedia Center, Osaka University

[Background] From supercomputing center viewpoint

- High Performance Computing Environment has been playing a role of greater importance.
- Efficient and higher job throughput is an important mission in supercomputing centers. Waiting time is also important..



Computing Cluster is a major architecture in HPC

[Background] From a system administration and management viewpoint

administrators

For High-throughput administration,

- Utilization restriction of computational resources on scheduler queue
- Per-user configuration in terms of job priority
- Selection of scheduling algorithm

Due to the scale-out and architectural heterogeneity of systems, a lot of configuration parameters should be considered.

➡ Administrators' workload is becoming heavier.

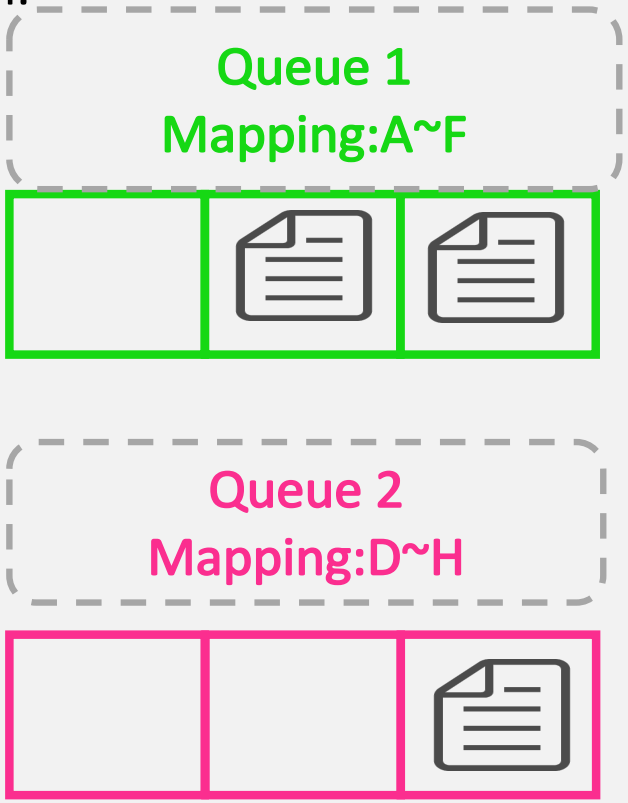
⬇

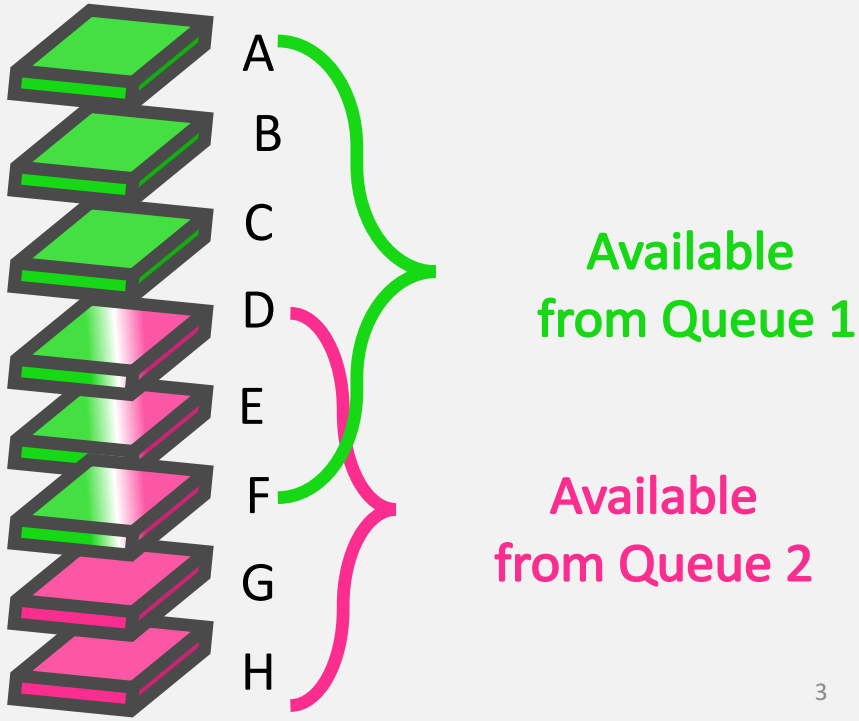
A mechanism that reduces administrators' workload is demanded.

[Configuration example] Queue mapping configuration

- Job scheduler has multiple queues configured in most cases.
  - A single queue might be good theoretically? but multiple queue are configured in reality.
    - Hybrid system of different architecture
    - Prioritization of jobs
    - etc

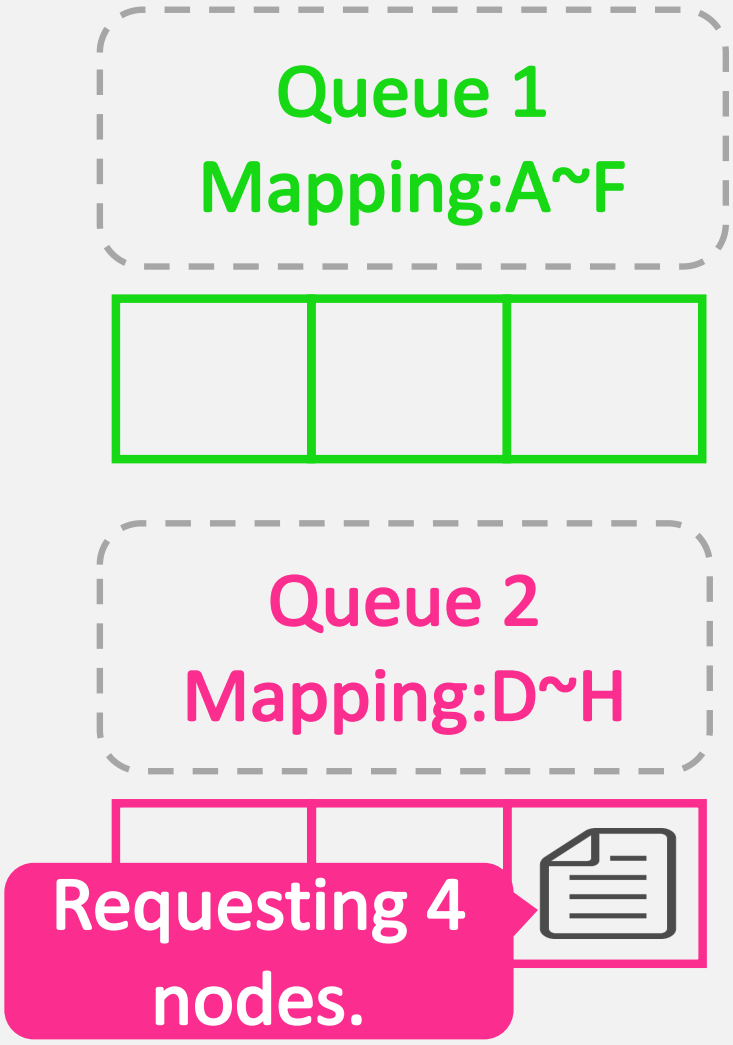
Mapping between queues and computing nodes becomes a hard configuration problem.

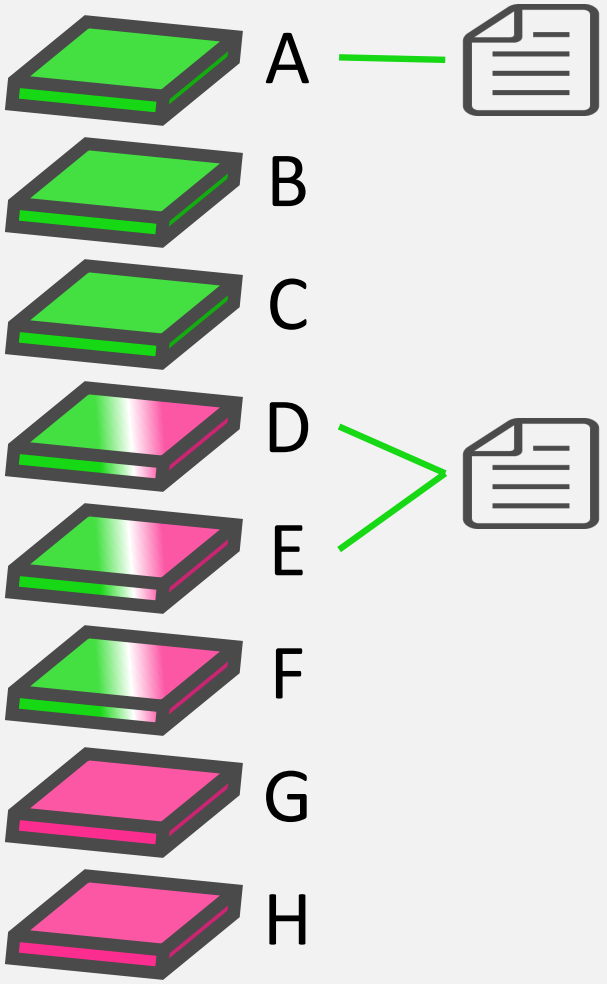




Example case (1/2)

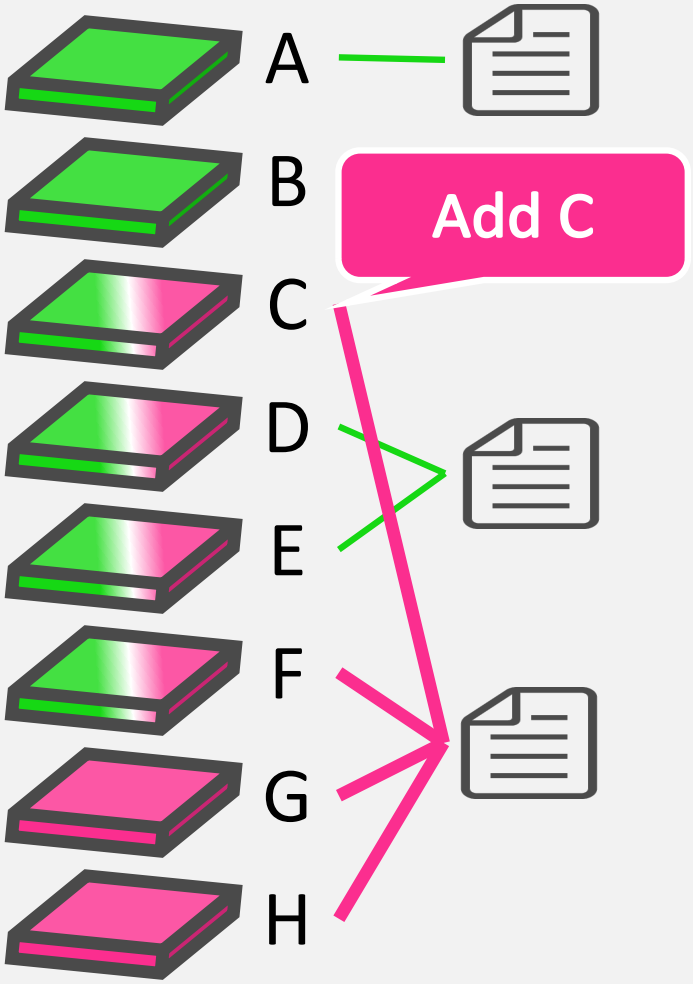
Initial state: job requesting 4 nodes on queue 2 cannot be assigned.



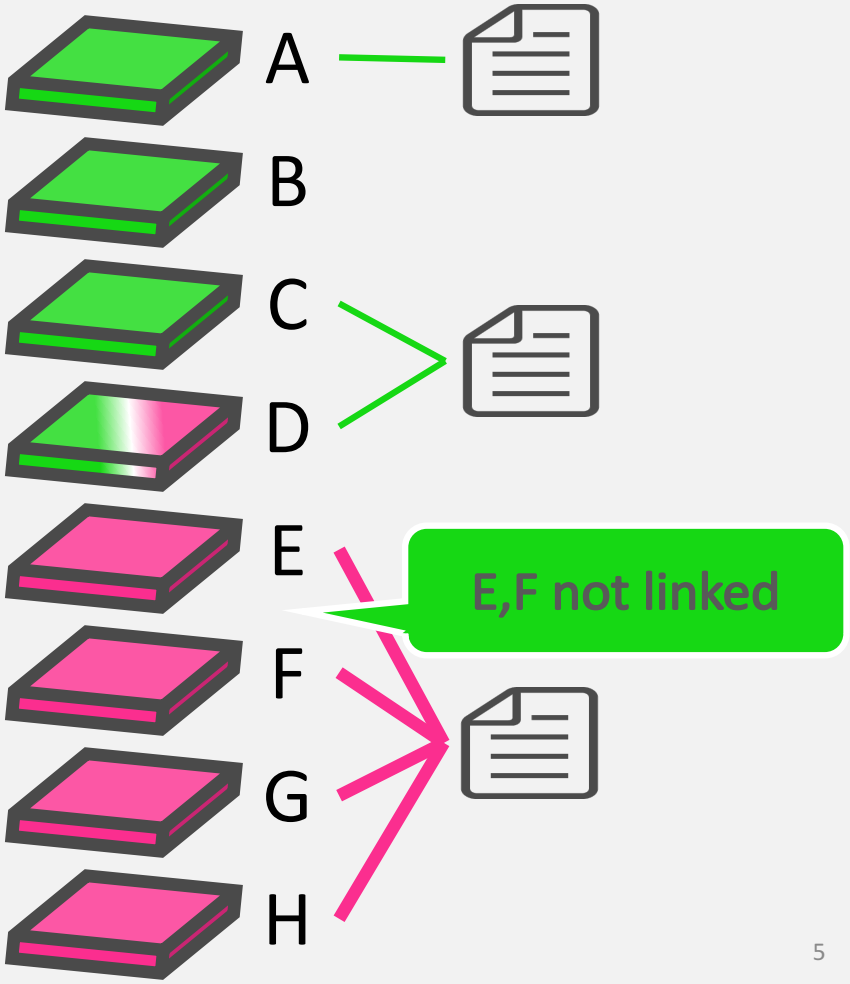


Example case (2/2)

If C is mapped to Queue 2,
the waiting job requesting 4 nodes can be executed on C, F, G and H.



If E, F are not linked to Queue 1, the running job on D and E could be executed and the waiting job requestinng 4 nodes can be run on E, F, G and H.



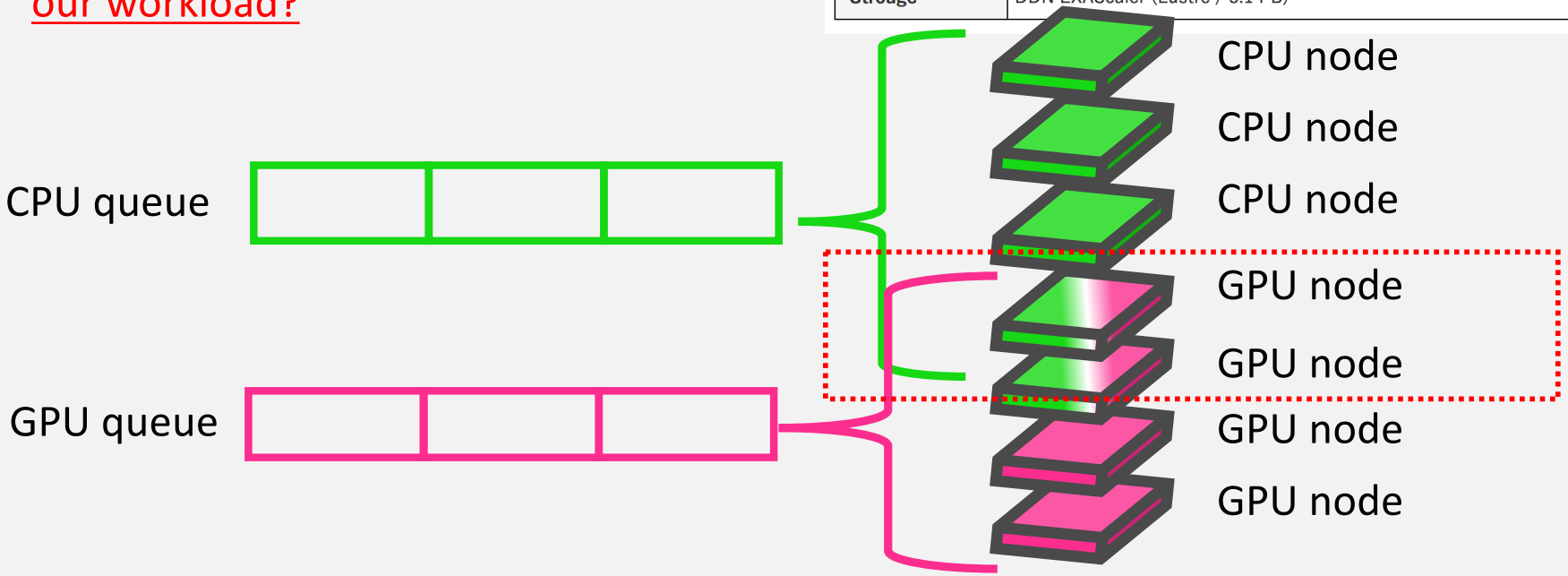
Actual example: queue mapping on OCTOPUS

Osaka university Cybermedia cenTer Over-Petascale Universal Supercomputer)

Theoretical Computing Speed	1.463 PFLOPS	
Compute Node	General purpose CPU nodes 236 nodes (471.24 TFLOPS)	CPU : Intel Xeon Gold 6126 (Skylake / 2.6 GHz 12 cores) 2 CPUs Memory : 192GB
	GPU nodes 37 nodes (858.28 TFLOPS)	CPU : Intel Xeon Gold 6126 (Skylake / 2.6 GHz 12 cors) 2 CPUs GPU : NVIDIA Tesla P100 (NV-Link) 4 units Memory : 192GB
	Xeon Phi nodes 44 nodes (117.14 TFLOPS)	CPU : Intel Xeon Phi 7210 (Knights Landing / 1.3 GHz 64 cores) 1 CPU Memory : 192GB
	Large-scale shared-memory nodes 2 nodes (16.38 TFLOPS)	CPU : Intel Xeon Platinum 8153 (Skylake / 2.0 GHz 16 cores) 8 CPUs Memory : 6TB
Interconnect	InfiniBand EDR (100 Gbps)	
Stroage	DDN EXAScaler (Lustre / 3.1 PB)	

A part of GPU nodes can be used for CPU nodes.

How many GPU nodes can be used against our workload?



[Motivation]Difficulties in configuring queue mapping

administrator

No objective criteria for configuration, relying on knowhow and experience

He/she cannot say whether his/her configuration gain high-throughput or not.

A lot of combination between configuration parameters have to be considered in a trial-and-error manner.

Proper mapping configuration have to be assisted.

[Approach] Use of Job scheduler simulation for analyzing the behavior of job assignments in system

- A number of job scheduler simulators are available.

  GridSim, Slurm simulator, gem5-gpu simulator, MERPSYS, ALEA…

  - However, most of job scheduler simulators do not help us in exploring the analysis space of queue mapping problem.
    - Many of job scheduler simulators focus on scheduling algorithm research development.

      ➡ Only single queue is supported.
    - Many of them does not allow us to configure the mapping relationship between computing node and queue.

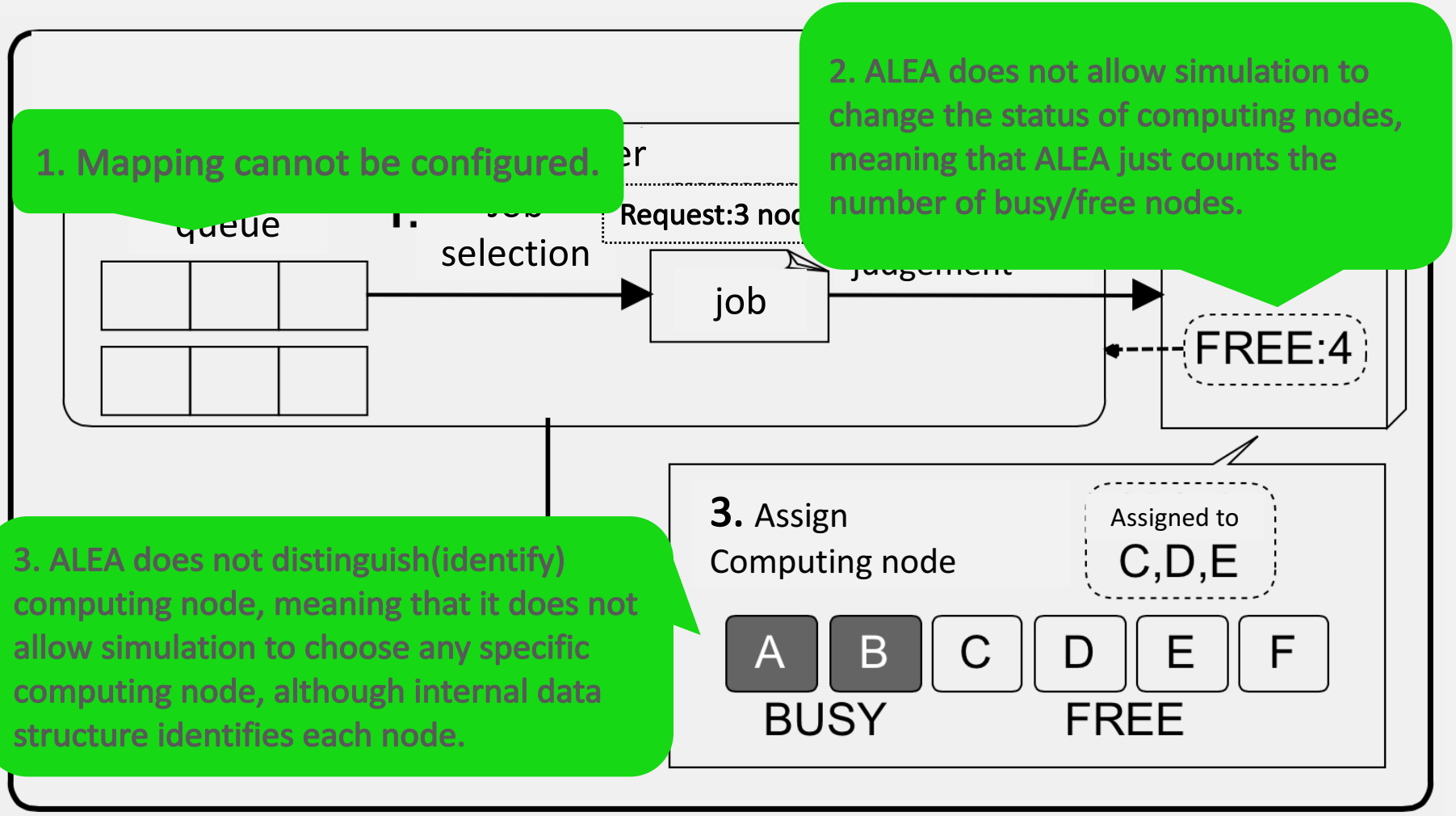
By modifying and extending an existent scheduler simulator,
we realize a tool that facilitates the system administrator to learn
which queue mapping is better for a certain series of job requests.

Our approach : to develop such a tool based on ALEA [Klusáček et.al. 2010].

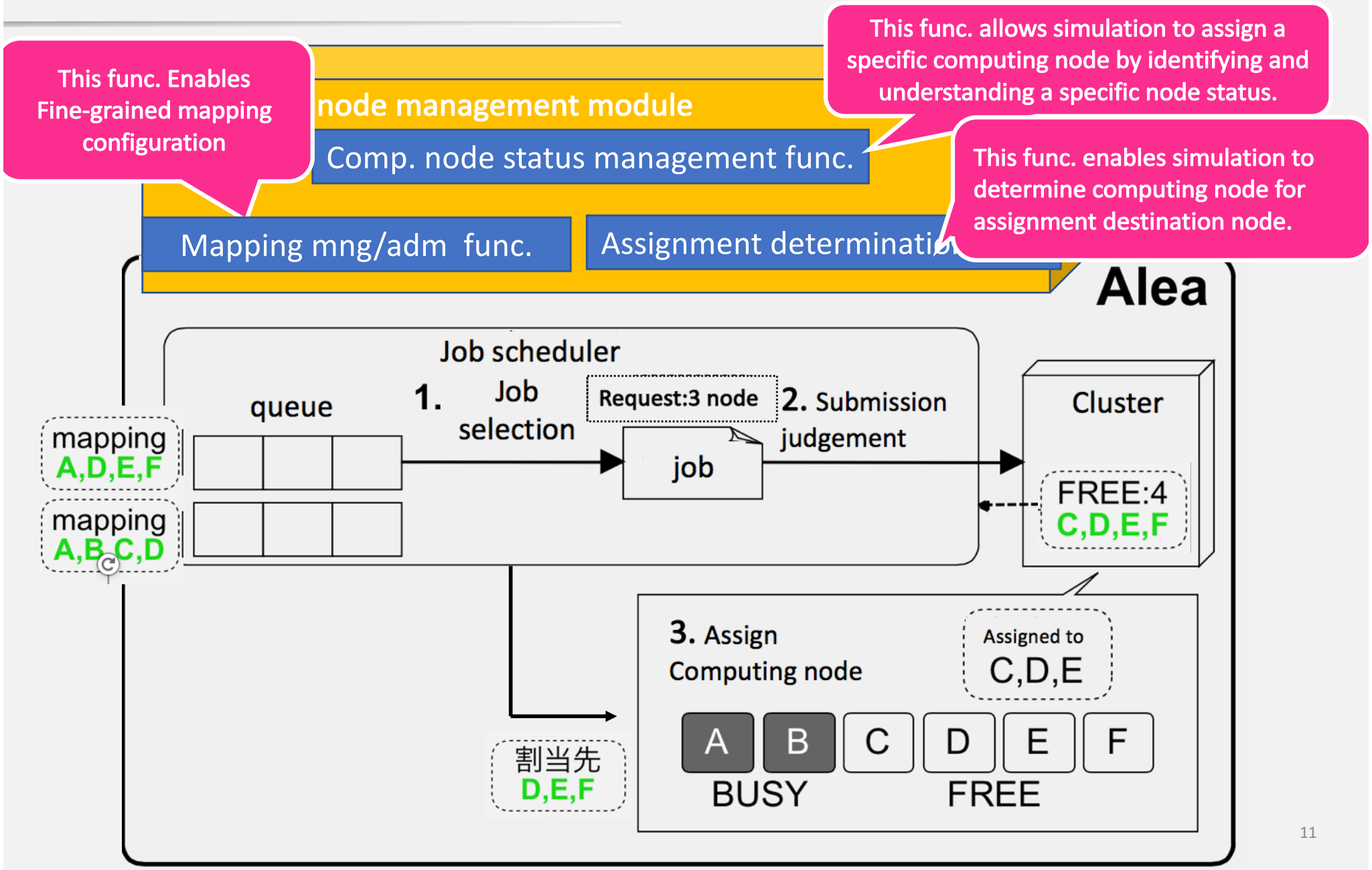
ALEA architecture and lacked functionalities.

ALEA: a grid-sim based simulator

However, three functionalities at least are lacked for achieving the goal.

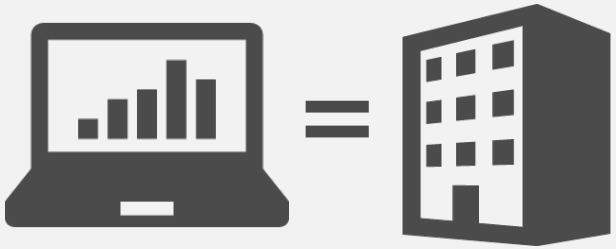


Solution: Computing node management module as Extension to ALEA



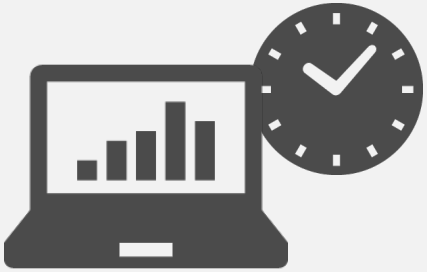
Evaluation

1. Comparison with real environment



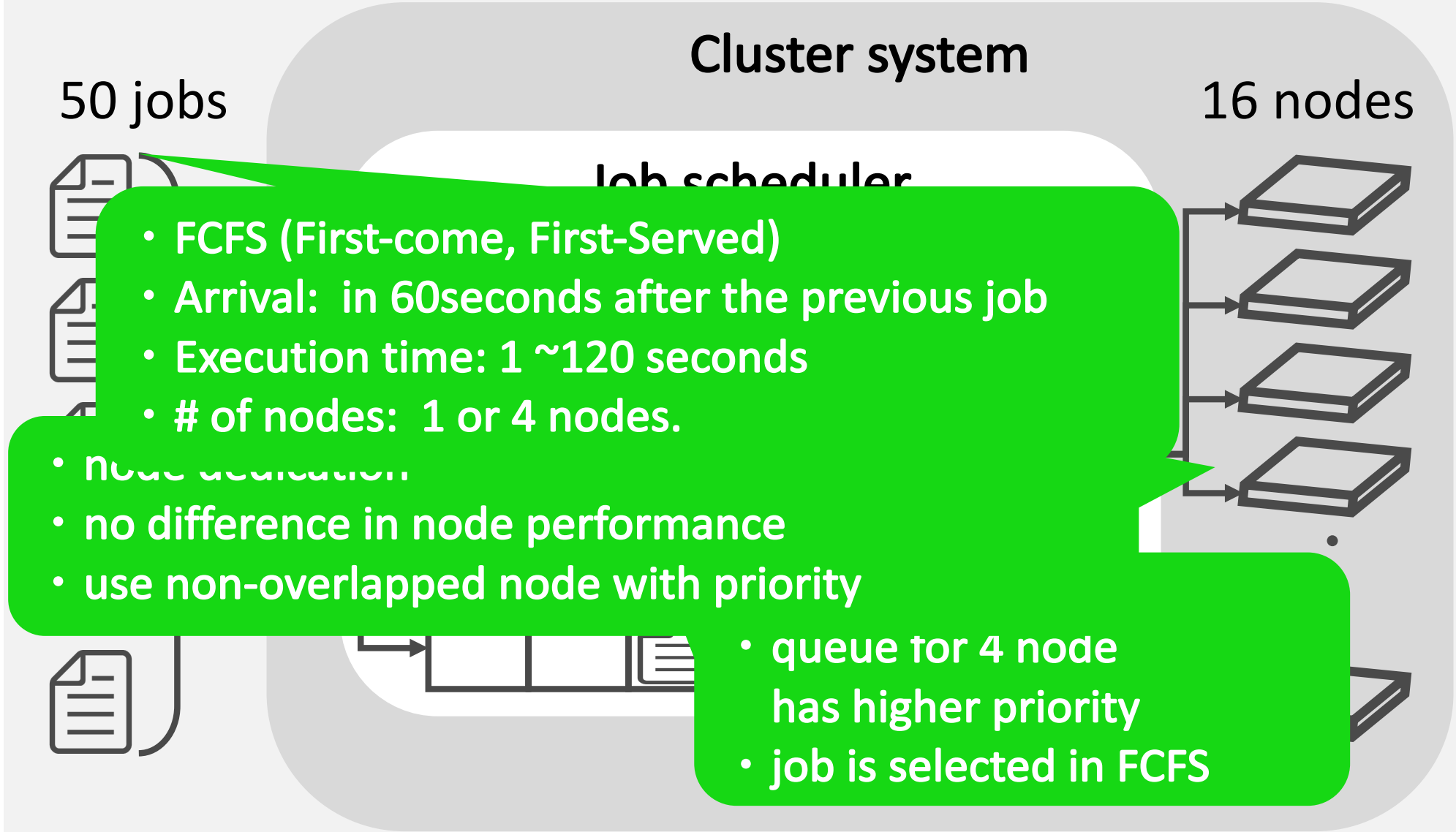
To verify whether to simulate the behavior of jobs on actual cluster system

2. Time to search all analysis space



To know how long it takes to derive to proper mapping configuration

Experiment conditions

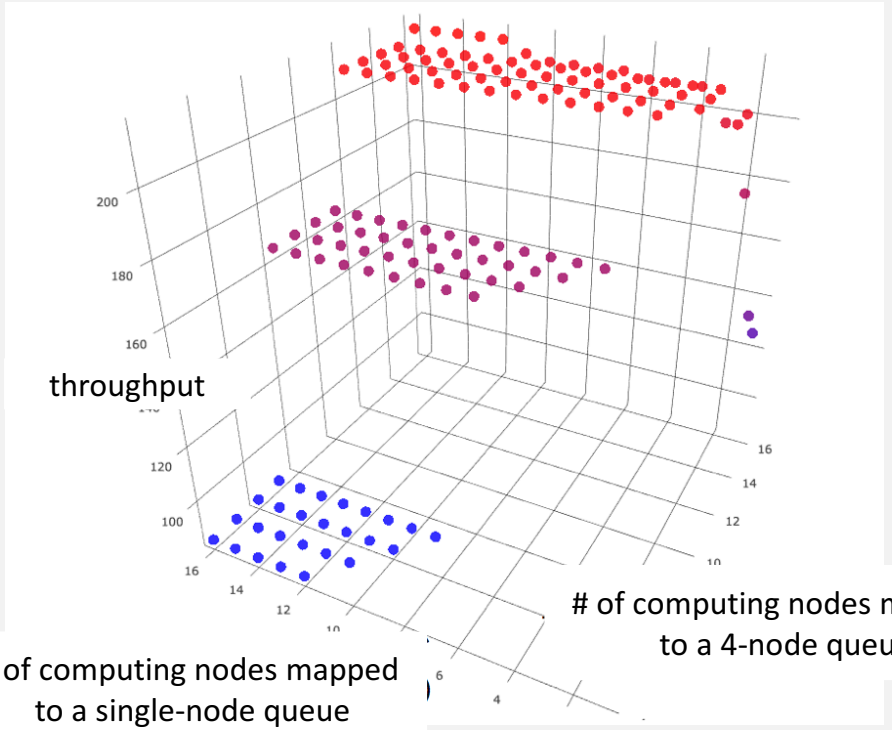


Experiment 1: **Comparison**

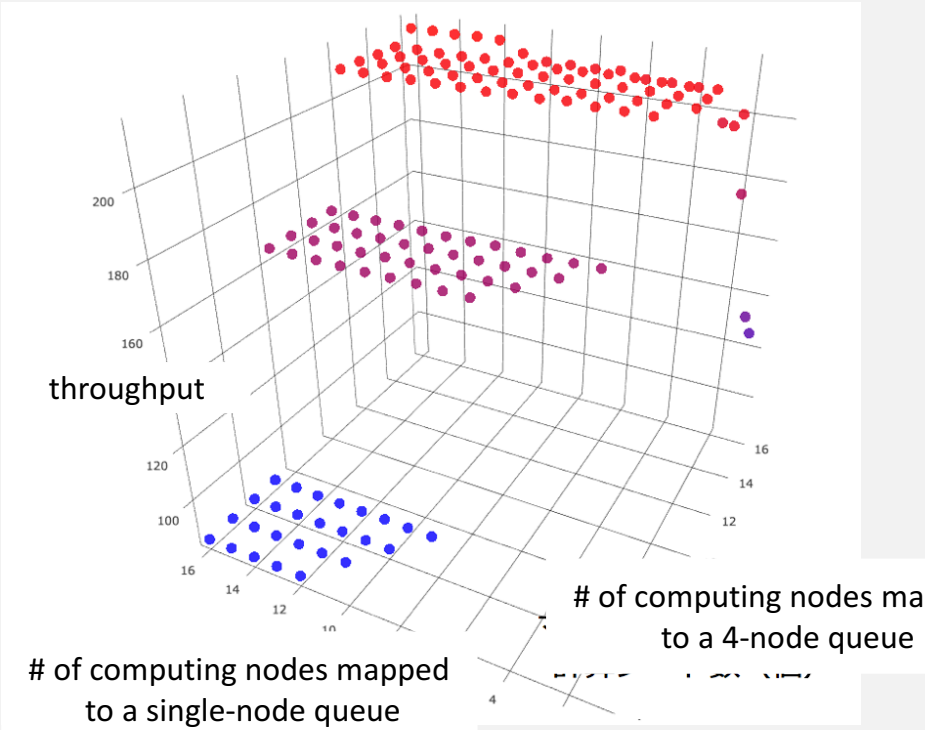
- Cluster system with Slurm deployed is simulated.

Simulator behaves in almost the same way as the real environment in terms of throughput

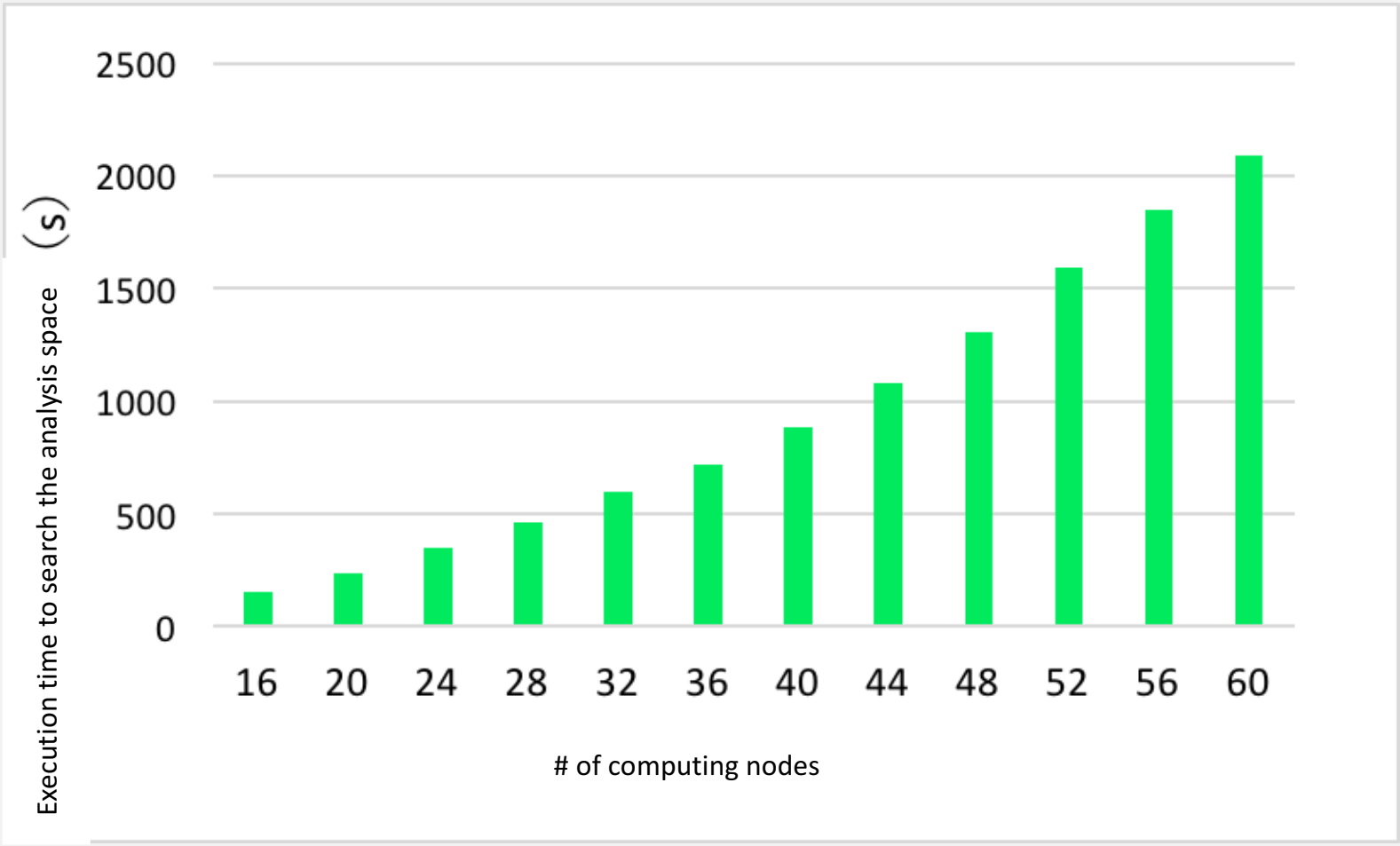
Simulator



Real environment

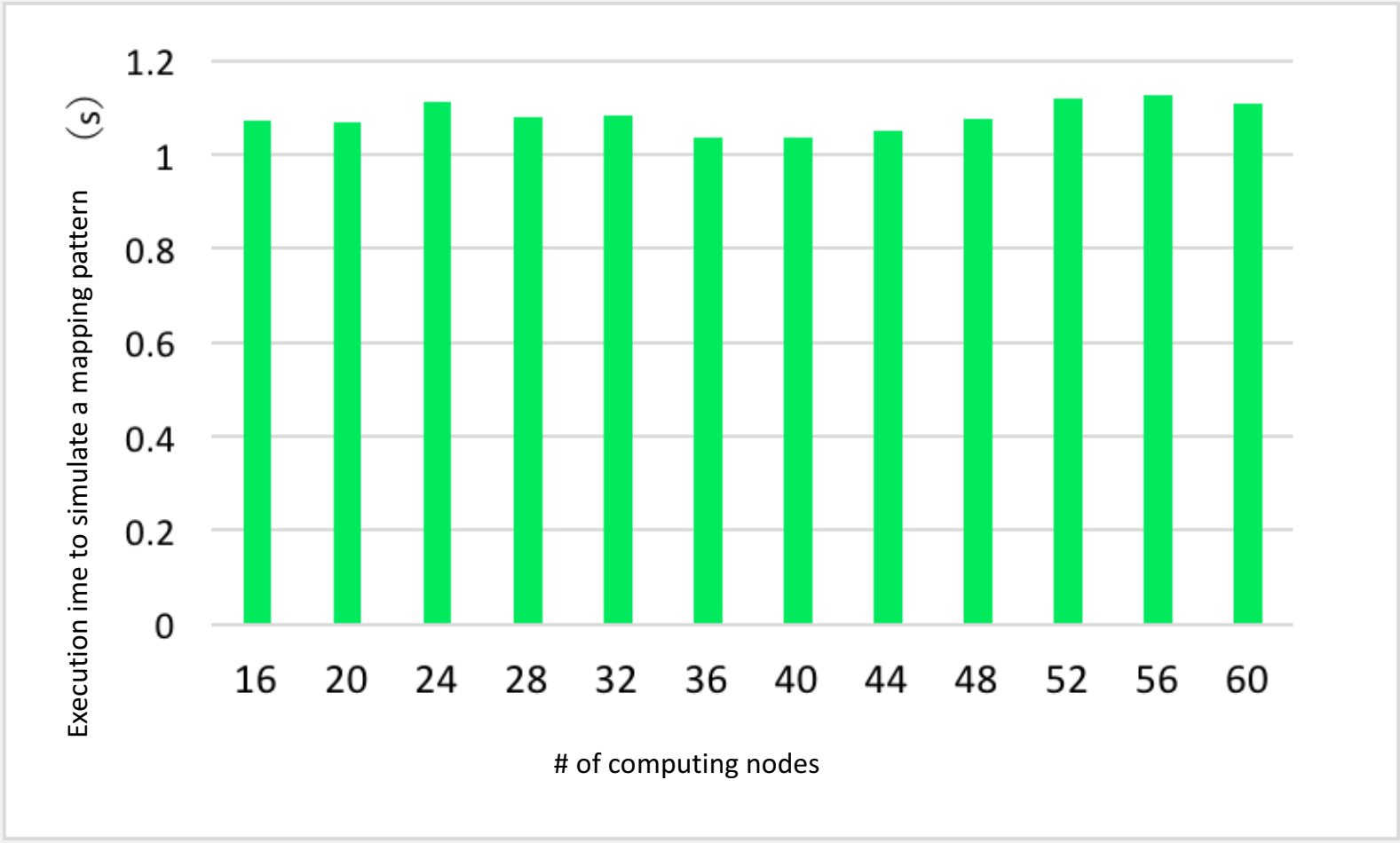


Experiment 2: Execution time to search the analysis space



Execution time dramatically increases..

Experiment 2: Execution time to simulate a mapping pattern



Execution time is not dependent on # of comp. nodes.

Summary

Problem：Currently, administrators are relying on their own experience and knowhow on scheduler’s queue mapping.

Goal: Build a job scheduling simulator that allows administrators to investigate the behavior of jobs on queue-computing node mappings.

Evaluation：So far, we have checked that extended simulator offers simulation results close to real environment.

Future Issues

- Reducing the execution time to search analysis space.