



[admire-eurohpc.eu](http://admire-eurohpc.eu)

# Adaptive multi-tier intelligent data manager for Exascale



ADMIRE Users Day

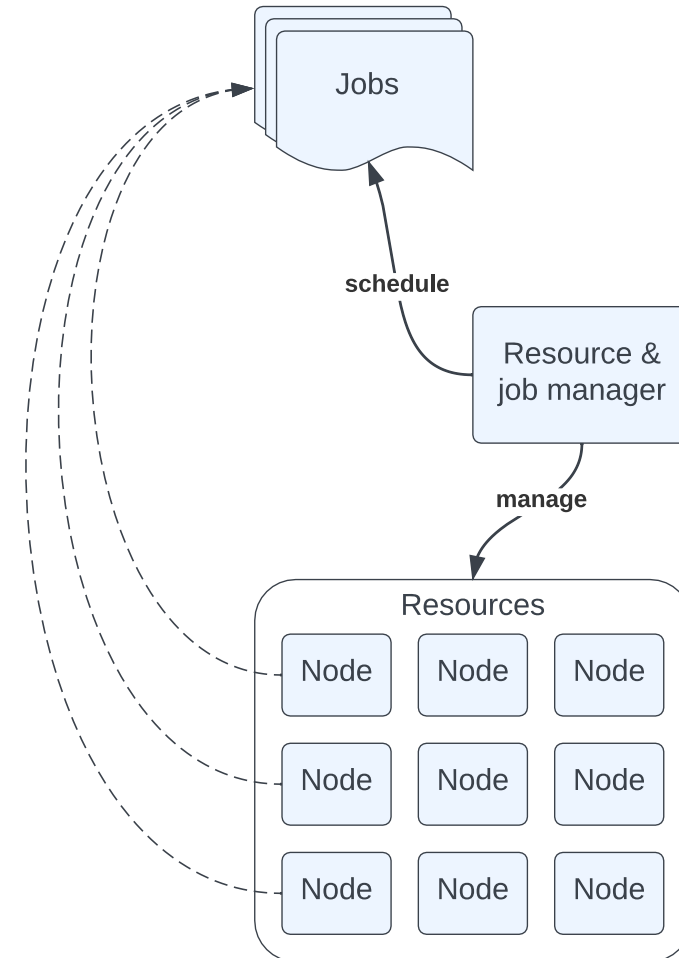
## ElastiSim: A Batch-System Simulator for Malleable & Evolving Workloads

**Taylan Özden**  
**Technical University of Darmstadt**

**December 12<sup>th</sup>, 2023**  
**Barcelona Supercomputing Center**

Grant Agreement number: 956748 — ADMIRE — H2020-JTI-EuroHPC-2019-1

- Resource and job management systems (also often called batch systems) schedule jobs and provide resources in large-scale computing environments
- Depending on the objective, batch systems aim to maximize system efficiency and decrease job completion times
- Scheduling algorithms are key components to improve system performance

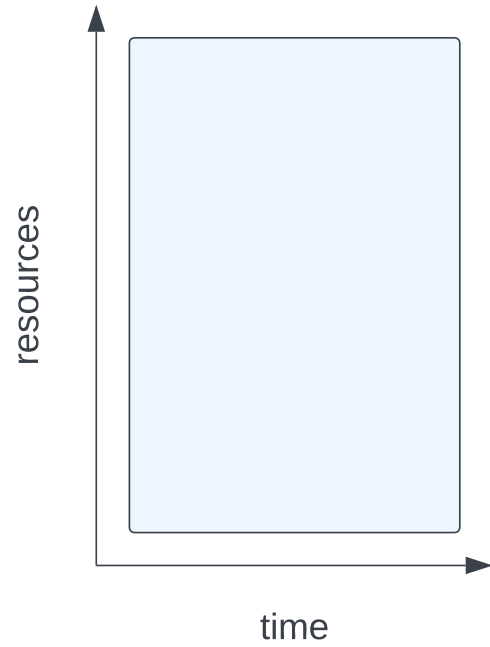


- Feitelson and Rudolph proposed four job categories
- Distinguished by *who* decides the number of assigned resources *at which time*:

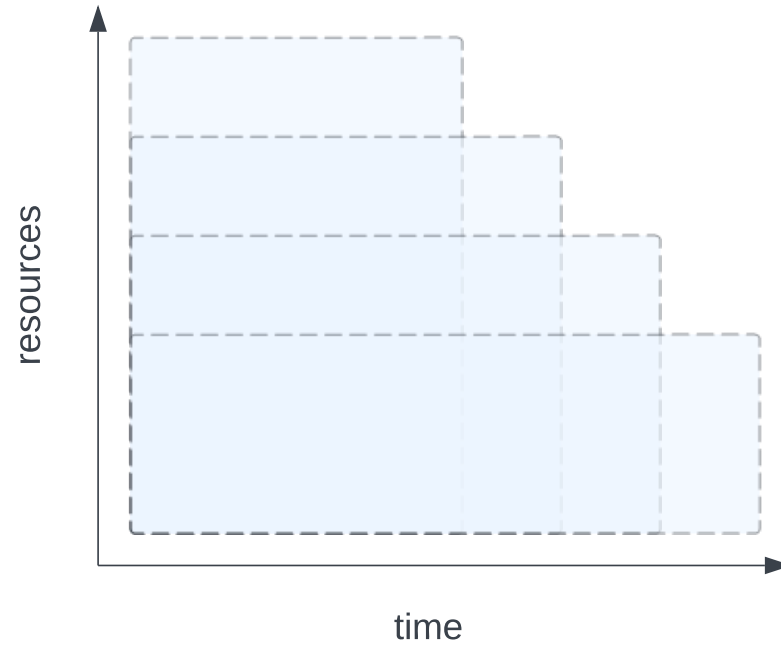
Decision	by user	by system
at submission	rigid	moldable
at runtime	evolving	malleable

- Evolving and malleable jobs are often classified as adaptive jobs

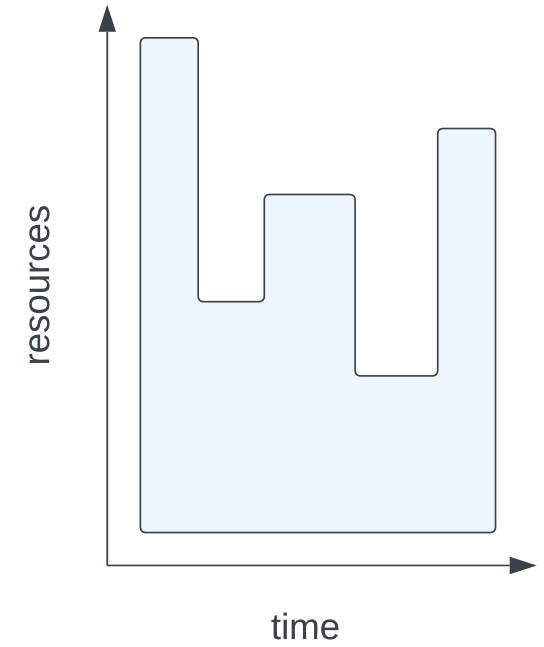
Rigid job



Moldable job



Adaptive job

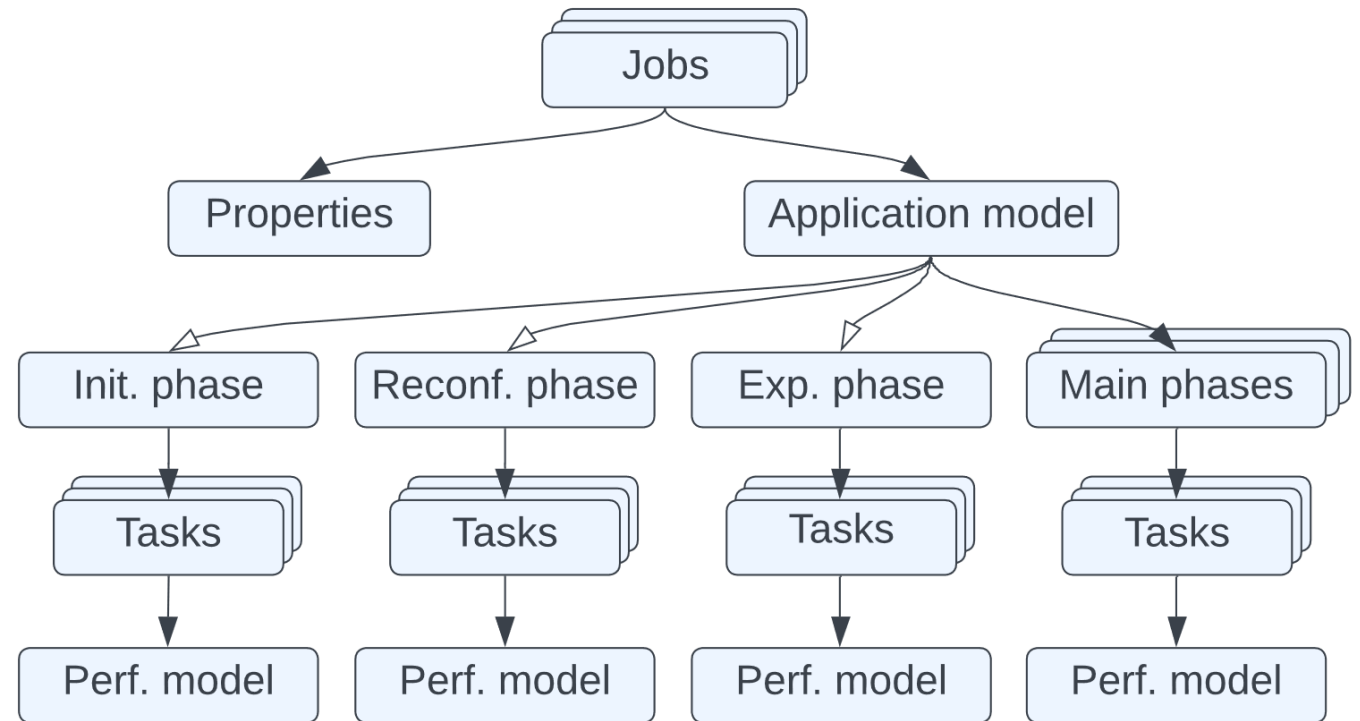


- Experiments on real systems?
  - Expensive
  - Time-consuming
  - Resource-intensive
  - Possible system threats
  - Not available to everyone
- Simulations
  - Fast
  - Reproducible
  - Independent
  - Variable constraints
  - Resource-efficient

- ElastiSim is a simulator that simulates
  - jobs and applications,
  - the batch system supporting rigid, moldable, malleable, and evolving workloads,
  - the scheduling algorithm (as part of the batch system),
  - the platform (powered by SimGrid).
- Typical use case: evaluating algorithms for the combined scheduling of rigid, moldable, malleable, and evolving jobs

- Why workload modeling?
  - Reliability of platform simulations depends highly on the executed workload in the simulated environment
  - Non-representative workloads lead to inconclusive results
- Workload characterizations indicate that workloads tend to alternate between phases
- We propose a workload model comprising jobs and application models

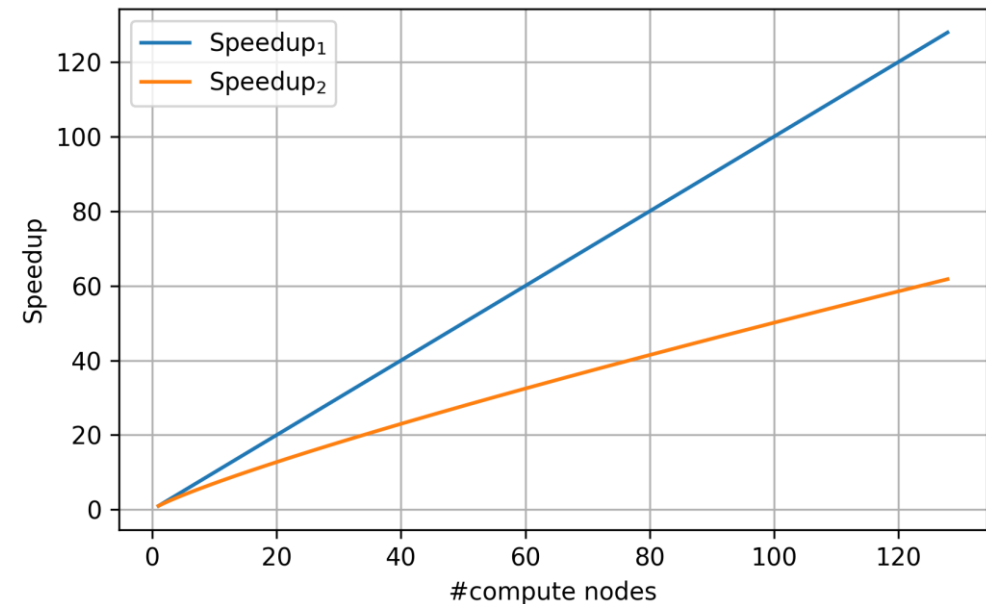
- A workload comprises jobs, with each job holding
  - properties (e.g., requested number of resources, submission time, etc.)
  - an application model
  
- Application models represent the simulated application
  - Divided into phases and tasks
  - Tasks support performance models to describe the load simulated on the platform



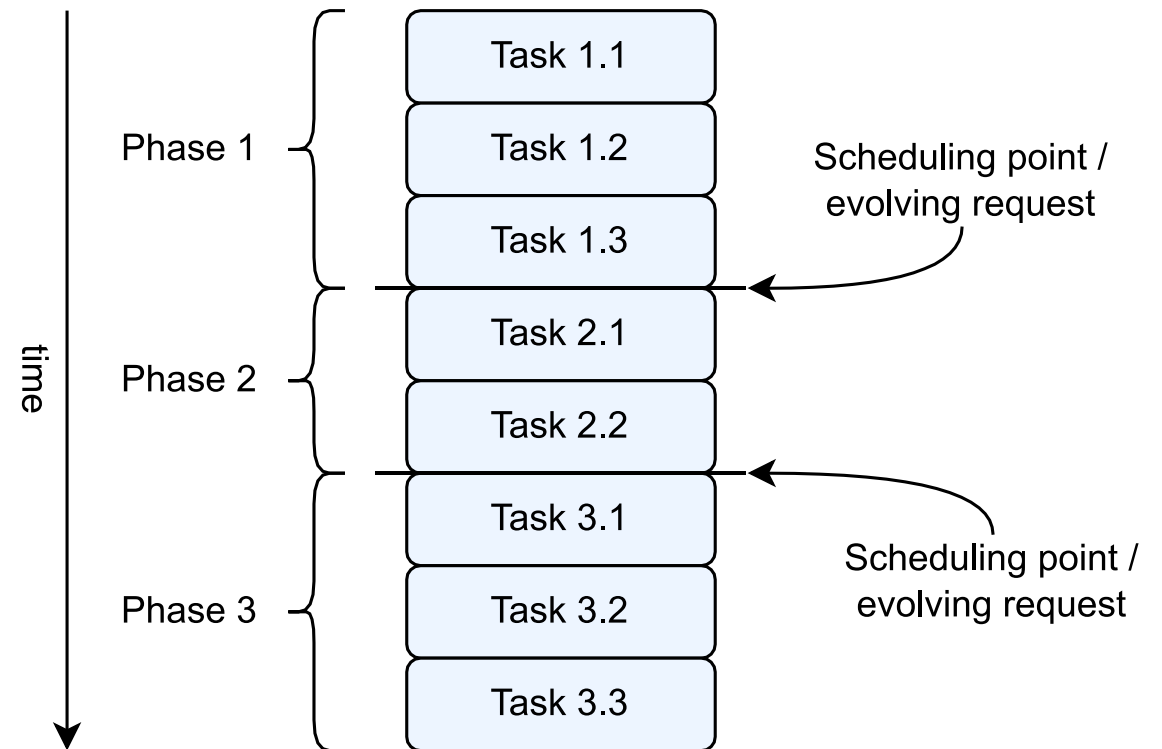


- Human-readable, mathematical functions
- Allows application models to adapt to resource (re)configurations
- Can be obtained by
  - inspecting applications
  - using tools (e.g., Extra-P)

- $\frac{FLOPs}{\#compute\ nodes}$
- $\frac{FLOPs}{(\#compute\ nodes)^{0.85}}$

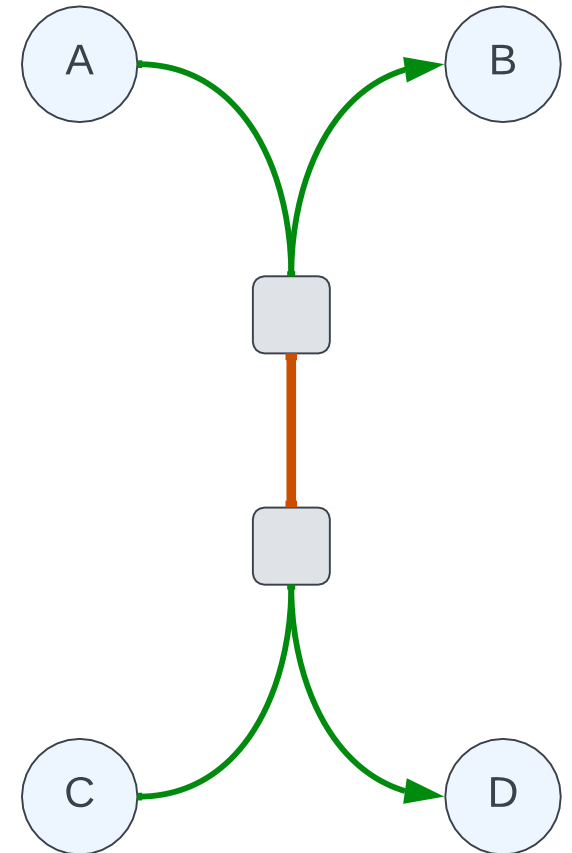


- Applications follow a building-block approach using phases and tasks
- ElastiSim provides various task types
  - Computation, Communication, I/O...
- Each task defines a *payload* introducing the load on the simulated platform
  - (e.g., bytes to communicate)
  - Supports performance models
  - Payload distribution patterns define *how* the payload is distributed over the resources

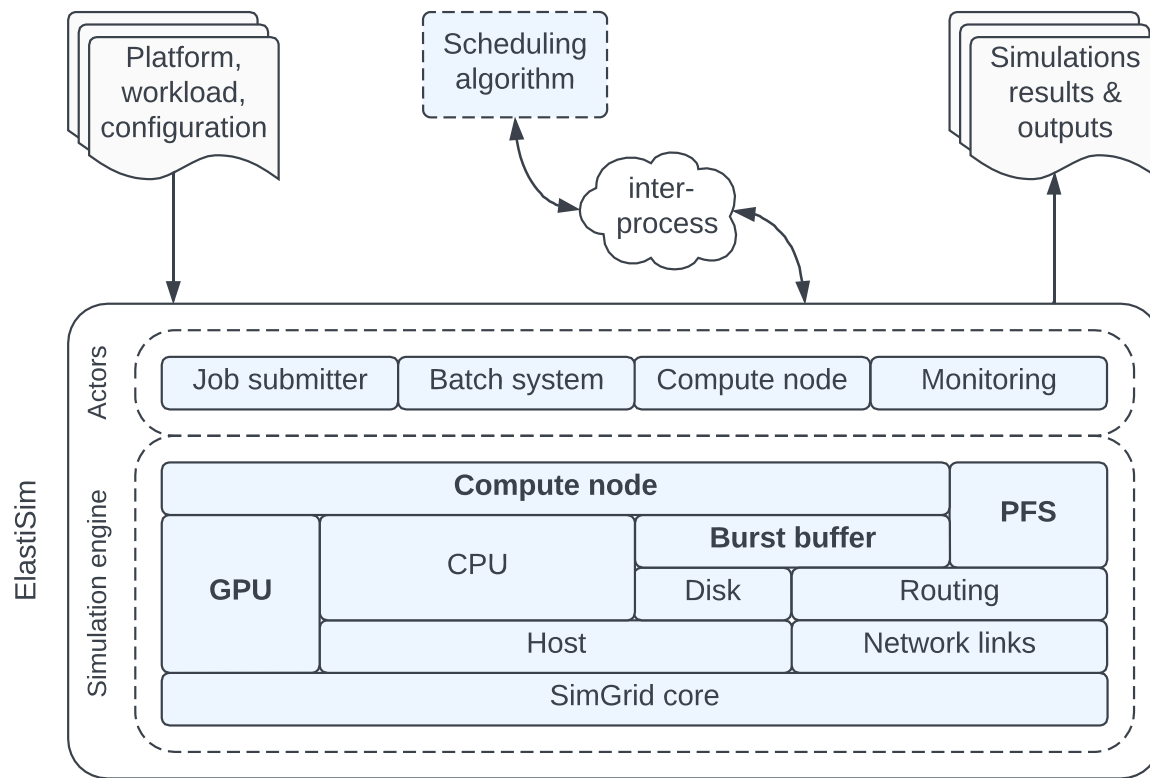


- Evolving jobs can request resources from the scheduler during runtime
  - Specified separately for each phase
  - Implicit scheduling point invoking the scheduler
  - The scheduler can reject or alter evolving requests
- Users define evolving requests either by specifying a
  - number, or
  - performance model evaluated by the runtime at the evolving request
    - (e.g., `num_nodes * 4`)
- ElastiSim introduces a fifth type, *adaptive* jobs, combining the features of malleable and evolving jobs
  - Allows application to contain phases that define either scheduling points or evolving requests

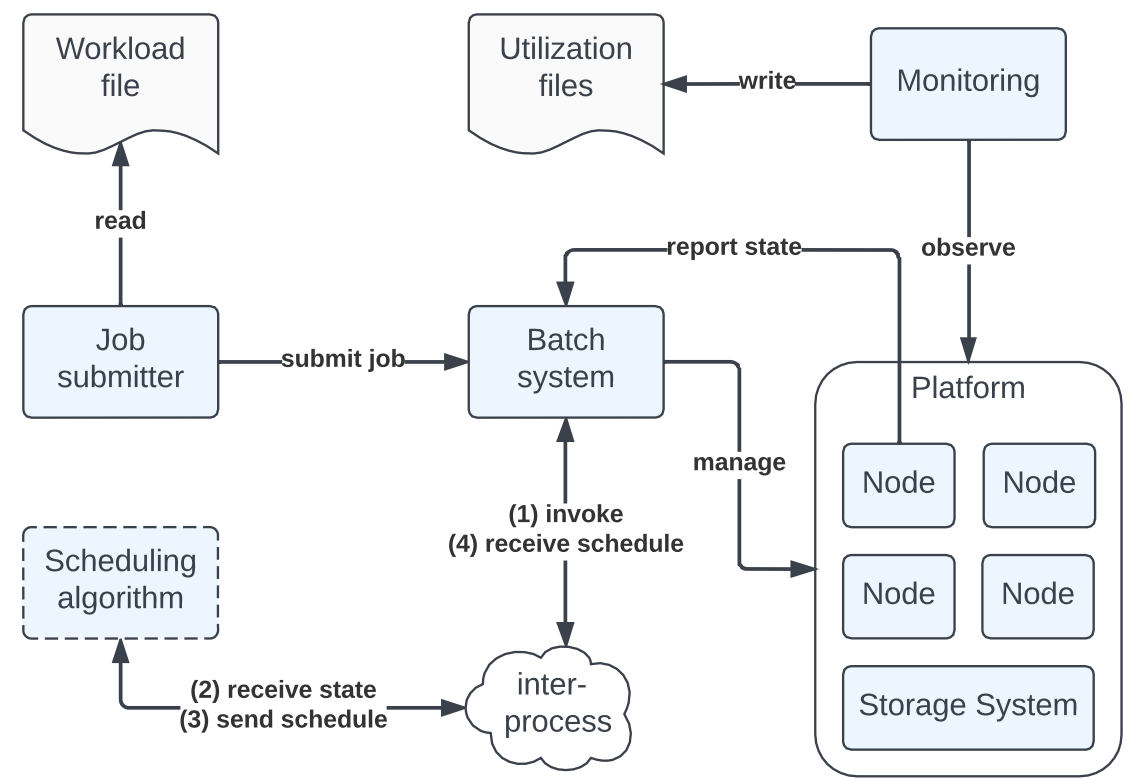
- State-of-the-art frameworks are discrete-event simulators modeling networks using a
  - packet-level, or
  - flow-level approach
  
- Packet-level simulators model every network packet as an event
  
- Flow-level simulators define network communication as data flows consuming available bandwidth
  
- We employ the flow-level simulator SimGrid below ElastiSim



## Architecture

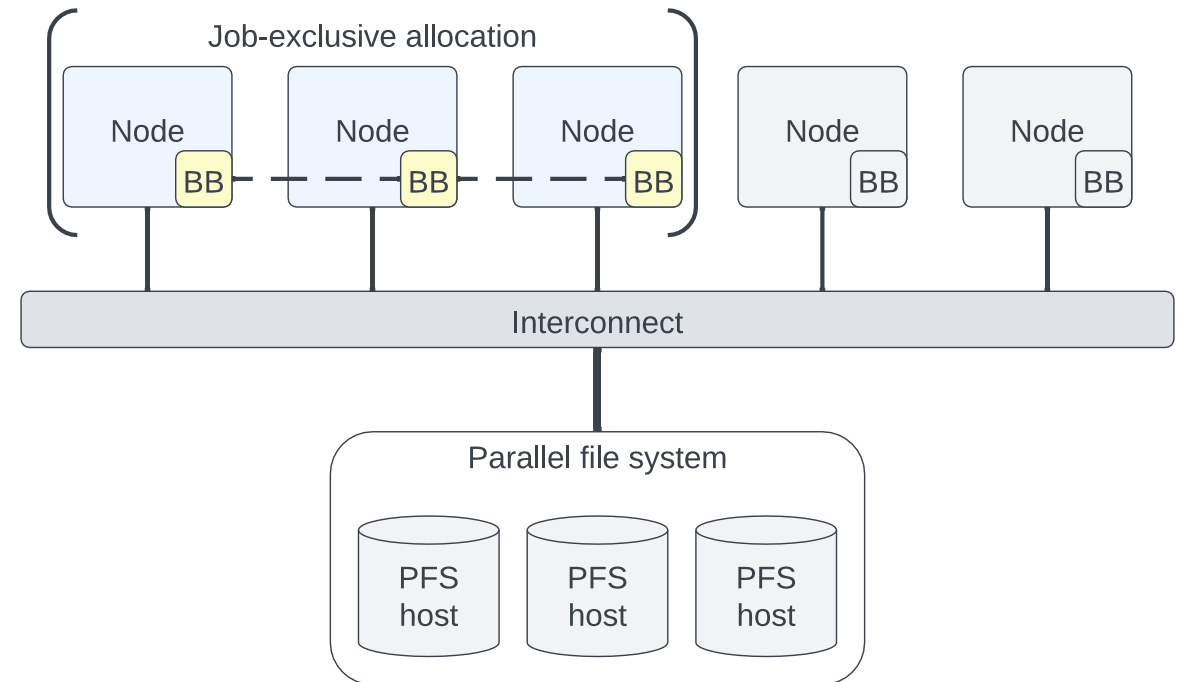


## Actors

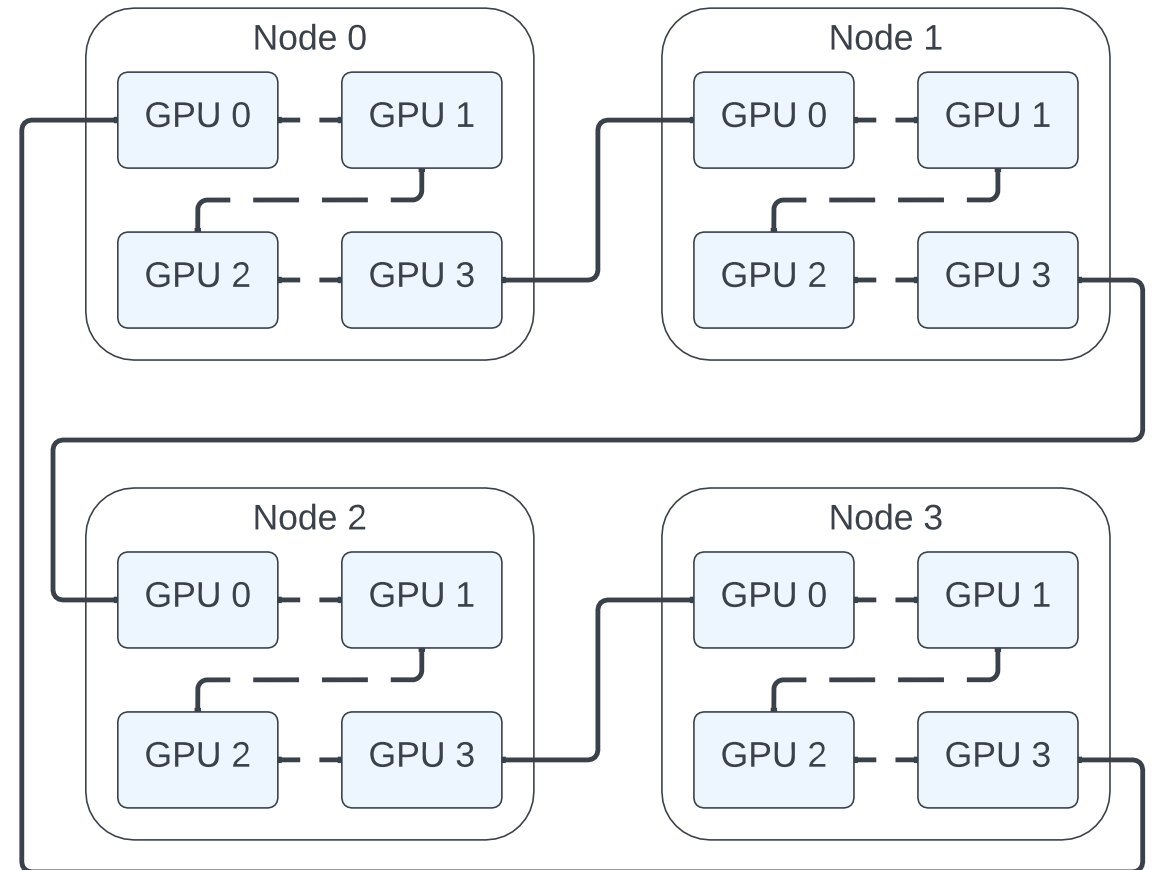


- **ElastiSim supports periodic and event-based algorithm invocation**
  - User-specified interval, if periodic
  - Invocation on job submission, finalization, scheduling points, evolving requests, or post reconfiguration, if specified
- **Each invocation contains the following information:**
  - job queue
  - state of each compute node
  - system metrics
- **Each job reports its progress**
  - Defined by the number of completed and total number of phases

- ElastiSim provides semantics for two types of storage systems
  - Parallel file systems (PFSs)
  - Node-local burst buffers (BBs)
- PFSs are modeled as dedicated I/O nodes behind a single namespace
- Node-local burst buffers come in two variants
  - Exclusive access
  - Wide-striped access



- Multiple GPUs per compute node
  - User-specified performance
  
- GPUs are fully connected
  - User-specified bandwidth
  
- Automatic detection of intra- and inter-node communication





- ElastiSim is a unique tool to simulate malleable and evolving jobs
- Website: <https://elastisim.github.io>  
(includes Slack invitation)
- GitHub: <https://github.com/elastisim>
- Contact me:  
[taylan.oezden@tu-darmstadt.de](mailto:taylan.oezden@tu-darmstadt.de)



*ElastiSim*



[admire-eurohpc.eu](http://admire-eurohpc.eu)

# Adaptive multi-tier intelligent data manager for Exascale



Thank you!

**Taylan Özden**  
**Technical University of Darmstadt**

**December 12<sup>th</sup>, 2023**  
**Barcelona Supercomputing Center**

Grant Agreement number: 956748 — ADMIRE — H2020-JTI-EuroHPC-2019-1