ElastiSim: A Batch-System Simulator for Malleable Workloads
Taylan Özden, Tim Beringer, Hamid Fard, Arya Mazaheri, Felix Wolf
Technical University of Darmstadt
Resource & job management

- 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 utilization and decrease the time to completion of jobs.
- Scheduling algorithms are key components to improve system performance.
Job classification

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

<table>
<thead>
<tr>
<th>Decision</th>
<th>by user</th>
<th>by system</th>
</tr>
</thead>
<tbody>
<tr>
<td>at submission</td>
<td>rigid</td>
<td>moldable</td>
</tr>
<tr>
<td>at runtime</td>
<td>evolving</td>
<td>malleable</td>
</tr>
</tbody>
</table>

- Evolving and malleable jobs are often classified as *adaptive* jobs
Job classification

- **Rigid job**
  - Resources vs. Time
  - Constant resource requirement throughout time.

- **Moldable job**
  - Resources vs. Time
  - Resource requirement changes over time, allowing for adjustments.

- **Adaptive job**
  - Resources vs. Time
  - Dynamic resource requirement that can change based on real-time needs.
Evaluating scheduling policies

- Experiments on real systems?
  - Expensive
  - Time consuming
  - Resource intensive
  - Possible system threats
  - Not available to everyone

- Simulations
  - Fast
  - Reproducible
  - Independent
  - Variable constraints
  - Resource efficient
What is ElastiSim?

- ElastiSim is a simulator that simulates
  - jobs and applications,
  - the batch system supporting rigid, moldable, and malleable 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, and malleable jobs
Workload modeling

• 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
Workload structure
Performance models

- 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)

\[
\frac{FLOPs}{\text{#compute nodes}}
\]

\[
\left(\frac{FLOPs}{\text{#compute nodes}}\right)^{0.85}
\]
Application execution model

Tasks
- CPU
- GPU
- I/O
- Delay

Payload distribution patterns
- Uniform
- All-to-all
- Ring
- Total
- Master-worker

Execution
- Task 1.1
- Task 1.2
- Task 2.1
- Task 3.2
- Task 3.1

Phase 1
Phase 2
Phase 3
Platform simulations

- 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
SimGrid

- Flow-level based simulation framework
- Supports various distributed computing systems
  - High-performance computers
  - Clouds
  - Grids
- Highly scalable, validated, and used in hundreds of publications
ElastiSim architecture
ElastiSim actors
Scheduling protocol

- Scheduling algorithm is invoked periodically
  - User-specified interval
  - Invocation on job submission/completion, if specified
- Each invocation contains the following information:
  - job queue
  - state of each compute node
  - system metrics (e.g., I/O utilization)
- Each job reports its progress
  - Defined by number of completed and total number of phases
**Storage model**

- 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
GPU model

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

Intra- and inter-node communication (ring pattern)
Experimental results

- To evaluate ElastiSim, we established two experiments to
  - validate our workload and application model (1)
  - assess the applicability of ElastiSim in large-scale scenarios (2)

  (1) We trained various convolutional neural networks on a GPU cluster
      - Comparison of real runtime against simulated runtime of modeled application

  (2) We simulated 400 jobs from Microsoft’s DL cluster Philly
      - Applied various scheduling algorithms and evaluated results
Epoch runtime comparison

![Graphs comparing epoch runtime for different models and node counts.](image-url)
Scheduling policies (wait and turnaround times)
Scheduling policies (wait and running times)

Rigid FCFS, rigid SJF, malleable FCFS (top, left to right); malleable DRF, malleable SJF, malleable SRTF (bottom, left to right)
Conclusion

- ElastiSim is the first batch-system simulator supporting malleability


- Website: https://elastisim.github.io/
- GitHub: https://github.com/elastisim
- Contact: taylan.oezden@tu-darmstadt.de