Evaluating SLURM Simulator with Real-Machine SLURM and Vice Versa

Ana Jokanovic
Marco D’Amico
Julita Corbalan

PMBS18
Outline

• Introduction and motivation
• BSC SLURM Simulator structure
• Contributions in BSC SLURM Simulator
• Evaluation of the simulator
• Evaluation of real SLURM: use cases
• Conclusions & future work
Introduction and motivation

• Why simulator and not a generic simulator?
  - It keeps code structure, features, parameters of SLURM
  - It allows reusing code developed for SLURM, i.e. plugins

• Used in research:
  - Evaluate new scheduling policies
  - Evaluate new systems not yet in production

• Used in production systems:
  - Improve cluster performance
A bit of history

• BSC SLURM Simulator was born in 2011:
  - Slurm Simulator, Alejandro Lucero, BSC (SLUG’11)
    • Based on SLURM v2.2.6
• Latest version:
  - ScSF: A Scheduling Simulation Framework, Gonzalo P. Rodrigo at al. (JSSP’17) → our starting point!
    • Based on SLURM v14.03
    • Faster
    • Partially addressed problems affecting the simulator accuracy
SLURM Simulator structure

**Input preparation**

- `slurm.conf`
  - Workload & arch. description
- SWF
  - Convert
- trace

**Simulation**

- `sim_mgr`
- `slurmctld`
  - SLURM API (sbatch)
- `shared memory`
- `slurmd`
  - sync

**Output analysis**

- SLURM logs/outs/DB
- Individual job’s & system’s metrics
SLURM Simulator

• A new component, *sim_mgr*, manages:
  - Simulation start/end
  - Simulation time – simulating one second per iteration
  - Job submissions
• *slurmd* was modified to fake job execution
  - Multiple nodes are represented by the same *slurmd*
  - Batch jobs are simulated (no steps, no tasks created)
• *slurmctld* synchronizes with a new RPC: MESSAGE_SIM_HELPER_CYCLE
  - Allows to process all the messages and operations happening in a specific second
Contributions in the SLURM Simulator

We encountered different bugs, producing delays and deadlocks:

- Wrong synchronization between simulator components
  - Caused by sleeps, concurrent operations on shared variables, semaphores
  - Solved by implementing a two semaphores synchronization

- Delays in RPC exchange and jobs duration
  - Caused by uncontrolled epilog messages
  - Solved by managing the number of running epilogs

- Delays in scheduler calls
  - Caused by oversimplification of scheduler calls and time dependent events: periodic call of scheduler and background operations
  - Solved by removing sleeps and implementing periodic calls into SIM_HELPER window
Contributions in the SLURM Simulator

We encountered different bugs, producing delays and deadlocks:

- **Wrong synchronization between simulator components**
  - Caused by sleeps, concurrent operations on shared variables, semaphores
  - Solved by implementing a two semaphores synchronization

- **Delays in RPC exchange and jobs duration**
  - Caused by uncontrolled epilog messages
  - Solved by managing the number of running epilogs

- **Delays in scheduler calls**
  - Caused by oversimplification of scheduler calls and time dependent events: periodic call of scheduler and background operations
  - Solved by removing sleeps and implementing periodic calls into SIM_HELPER window
Other improvements

- Ported to version 17
- Implemented reading from SWF
- Implemented multiple simulation in the same machine (no VM are necessary)
- Scripts for lunching simulations, collecting results, output extraction, analysis and graphs generation
- Demo at BSC booth on: Tuesday 13, at 3pm
Evaluation

- **Consistency** evaluation: we compare multiple runs of the same simulation, and we try to understand causes of variation between runs
- **Accuracy** evaluation: we run the same workload in real SLURM and in the simulator
- **Performance** evaluation: we run big workloads in terms of system size and number of jobs and we evaluate Simulator speedup

- We compared ScSF Simulator with BSC version
Evaluation: Consistency

- **Consistency** evaluation: 4 logs generated with Cirne model, 5000 jobs, 3456 nodes:
  - ANL, CTC, KTH, SDSC arrival patterns
  - About 5 days of simulated time
- In sim_scsf variance depends on the system load

![Graphs showing time differences for KTH and SDSC](image-url)
Evaluation: Consistency

- **Consistency** evaluation: 4 logs generated with Cirne model, 5000 jobs, 3456 nodes:
  - ANL, CTC, KTH, SDSC arrival patterns
  - About 5 days of simulated time
- BSC Simulator is deterministic, variance was caused by errors!

![ANL vs. CTC graphs](image)
Evaluation: Accuracy

- **Accuracy** evaluation: 4 logs generated with Cirne model and converted to real jobs submissions
  - Comparing SLURM simulator and real SLURM in Marenostrum 4
  - 10 nodes, 200 jobs, about 2 hours makespan
**Evaluation: Accuracy**

- **Accuracy** evaluation: 4 logs generated with Cirne model and converted to real jobs submissions
  - Comparing SLURM simulator and real SLURM in Marenostrum 4
  - 10 nodes, 200 jobs, about 2 hours makespan
- Real SLURM is not deterministic!
Evaluation: Performance

- **Performance** evaluation:
  - ANL Intrepid complete log: 68936 jobs, 40960 nodes, Jan 2009 to Sept 2009, 9 months
  - CEA Curie complete log: 198509 jobs, 5040 nodes, Feb 2012 to Oct 2012, 9 months
- Up to 240x speedup
Use cases

- **Use cases** evaluation: evaluate a system by using SLURM Simulator
  - Running Cirne with ANL arrival pattern, 5000 jobs, 3456 nodes
    1) Analyze backfill interval
    2) Analyze number of tested jobs by the scheduler
    3) Analyze scaled up/down system performance
Use case 1: backfill interval
Use case 2: number of tested jobs
Use case 3: system size
Conclusion and future work

- SLURM Simulator is a powerful tool for research and system administration
- We did the first ever accuracy evaluation with a real scheduler implementation
- SLURM Simulator is used in European Projects (DEEP-EST)
- We published BSC Simulator’s code at:
  - https://github.com/BSC-RM/slurm_simulator
  - https://github.com/BSC-RM/slurm_simulator_tools

- Future work
  - Evaluate the accuracy comparing bigger runs
  - Event driven simulator, not updating time second by second
  - Model execution time based on hardware
  - Implement support for heterogeneous jobs
Thank you

ana.jokanovic@bsc.es
marco.damico@bsc.es
julita.corbalan@bsc.es