Barcelona
Supercomputing
Center
Centro Nacional de Supercomputación

EXCELENCIA
SEVERO
OCHOA

SLURM Simulator improvements and evaluation

Marco D’Amico
Ana Jokanovic
Julita Corbalan

SLUG’18

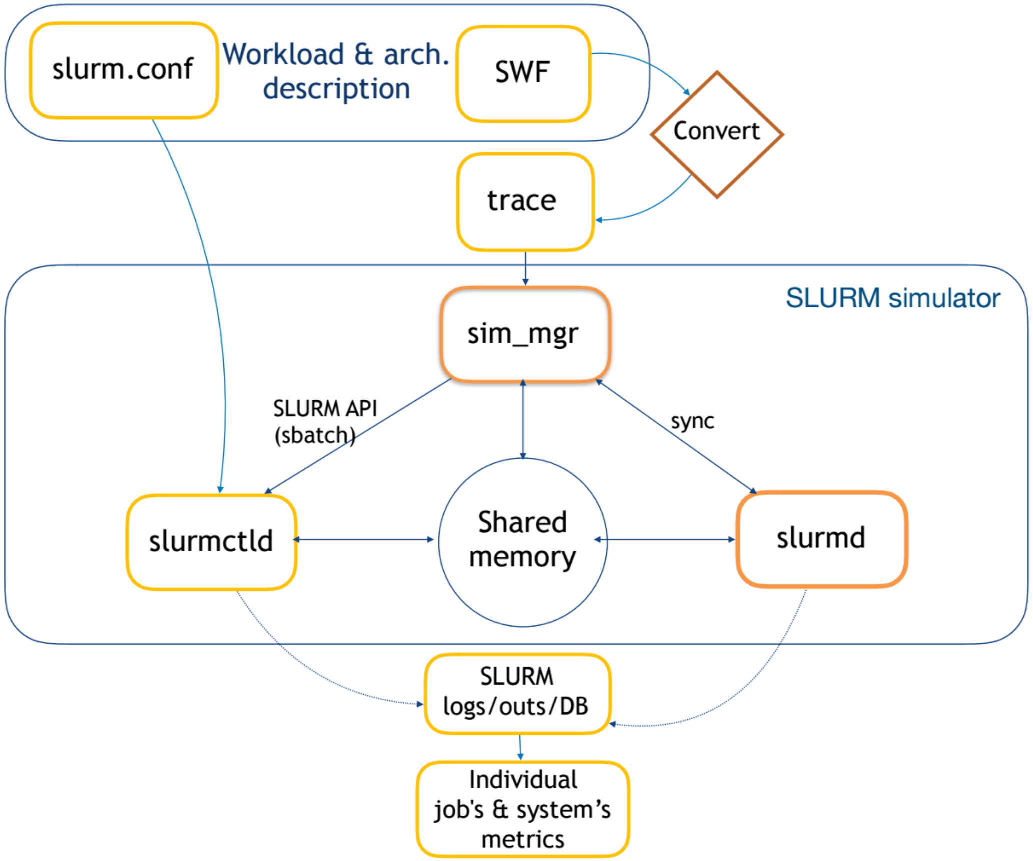
Introduction

- SLURM Simulator is able to simulate workloads execution
- Why not just a simulator?
  - It keeps code structure, features, parameters of SLURM
  - In production:
    - Improve cluster performance
  - In research:
    - Test behavior of scheduling policies
    - Test bigger systems not yet in production
- Other implementation-specific simulators:
  - Qsim: based on Cobalt, specific for Blue Gene systems
  - Moab simulator: proprietary software

A bit of history

- SLURM Simulator was born in 2011:
    - *Slurm Simulator*, Alejandro Lucero, BSC (SLUG'11)
        - Based on SLURM v2.2.6
- It grew up:
    - *Using and Modifying the BSC Slurm Workload Simulator*, Stephen Trofinoff and Massimo Benini, CSCS (SLUG'15)
        - Ported to v14.03.8
        - Improved code and usability
    - *Simunix, a large scale platform simulator*, David Glesser and Adrien Faure, Bull AtoS (SLUG'16) → **code missing!**
        - Integrated with Simgrid
    - *ScSF: A Scheduling Simulation Framework*, Gonzalo P. Rodrigo at al. (JSSP'17) → **our starting point!**
        - Faster
        - Partially addressed problems affecting the simulator accuracy

SLURM Simulator structure



SLURM Simulator

- Simulator uses front-end mode
- One simulated second per step
- A new component, *sim_mgr*, manages:
  - Simulation start/end
  - Simulation time
  - job submissions
- *slurmd* was modified to fake job execution
  - batch job lunch is simulated (no steps, no tasks created)
- *slurmctld* synchronize with a new RPC: MESSAGE_SIM_HELPER_CYCLE
  - Allows to process all the messages and operations happening in a specific second

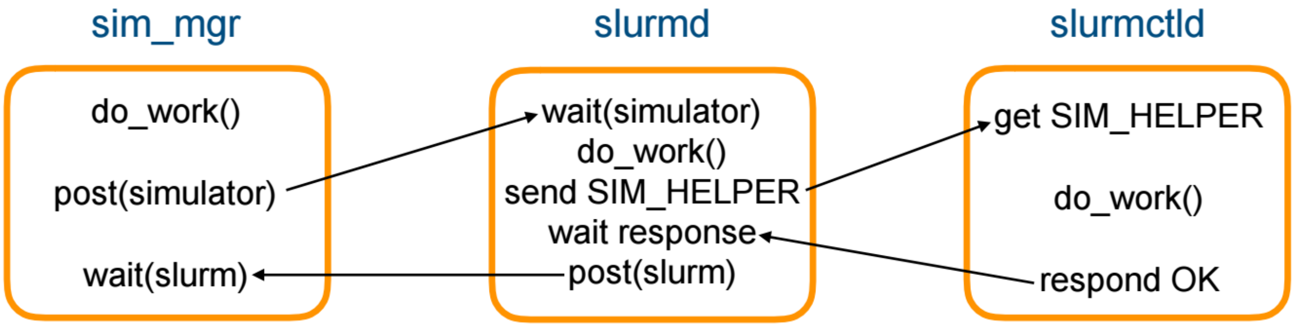
Problems in the SLURM Simulator

We encountered different bugs, producing delays and deadlocks:

- Wrong synchronization between simulator components
  - Caused by sleeps, concurrent operations on shared variables, semaphores
- Delays in RPC exchange
  - Caused by uncontrolled epilog messages
- Delays in scheduler calls
  - Caused by oversimplification of scheduler calls and time dependent events: periodic call of scheduler and background operations
- Other small bugs

Fixes in synchronization: the implemented flow

- 1 second step simulation, for each second:
  - *sim_mgr* sends REQUEST_SIM_JOB to *slurmd* communicating jobid and duration, then submits the job to *slurmctld* via SLURM API
  - *slurmd* sends job termination msg to *slurmctld* for ending jobs
  - *slurmd* sends MESSAGE_SIM_HELPER_CYCLE to *slurmctld* to synch and waits for a response
  - *slurmctld* process new jobs, ending jobs, time triggered calls (e.g. backfill), then responds OK to *slurmd*
  - *slurmd* unlocks *sim_mgr*
  - *sim_mgr* go to the next second



Fixes in RPC exchange

- For each terminated job:
  - slurmd sends a *termination message*
  - slurmctld respond with a request of *epilog*:
    - This message was not processed in the current second, especially when the number of ending/starting jobs increases
    - Job end is marked after slurmd execute epilog and respond to slurmctld
  - Both slurmd and slurmctld were not waiting for finished epilog!
  - **We made slurmctld and slurmd wait all epilogs**
  - shared counters between threads managing RPCs
- All jobs now complete in one simulated second!

Fixes in slurmctld

- All time dependent events affected
- Fixed backfill **periodic** calls, using interval parameter from slurm.conf
  - Checking last backfill time and interval during MESSAGE_SIM_HELPER_CYCLE processing
- We moved check of priority scheduler in MESSAGE_SIM_HELPER_CYCLE
  - Fixed scheduler call **at job end**
  - Fixed **periodic** scheduler call
- Fixed new arriving RPC that could interrupt backfill
  - Backfill now consume only one simulated second!

Other improvements

- Ported to version 17
- Implemented reading from SWF traces
- Implemented multiple simulation in the same machine (no VM are necessary)
- Stop simulation at end of a simulated trace (option)
- Scripts for lunching simulations, collecting results, output extraction, analysis and graphs generation

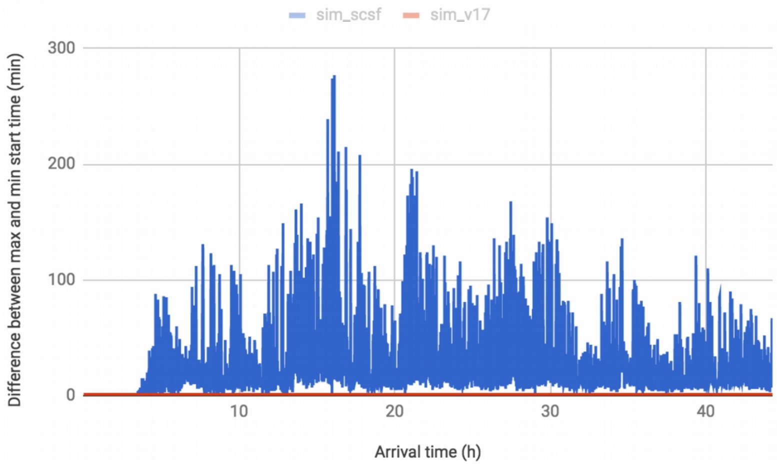
Evaluation: Testbed

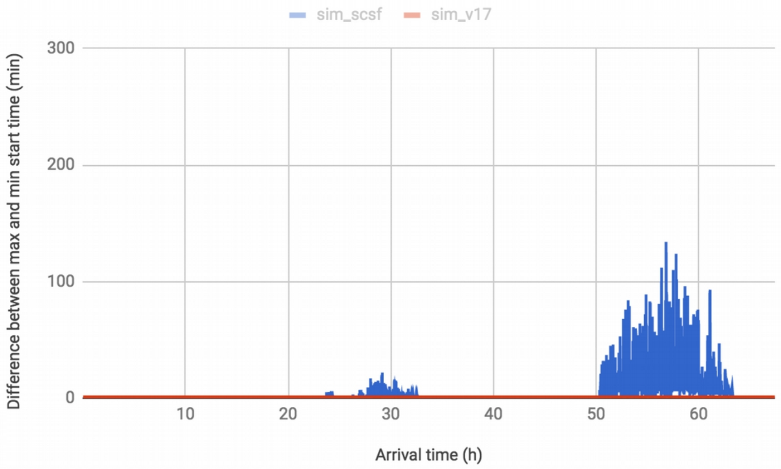
- **Consistency** evaluation: 4 logs generated with Cirne model, 5000 jobs, 3456 nodes:
  - ANL, CTC, KTH, SDSC arrival patterns
  - 10 runs
  - About 5 days of simulated time
- **Accuracy** evaluation: 4 logs generated with Cirne model and converted to real jobs submissions
  - Comparing simulator and real SLURM
  - 10 nodes, 200 jobs, about 2 hours makespan
- **Performance** evaluation:
  - ANL Intrepid complete log: 68936 jobs, 40960 nodes
  - CEA Curie complete log: 198509 jobs, 5040 nodes

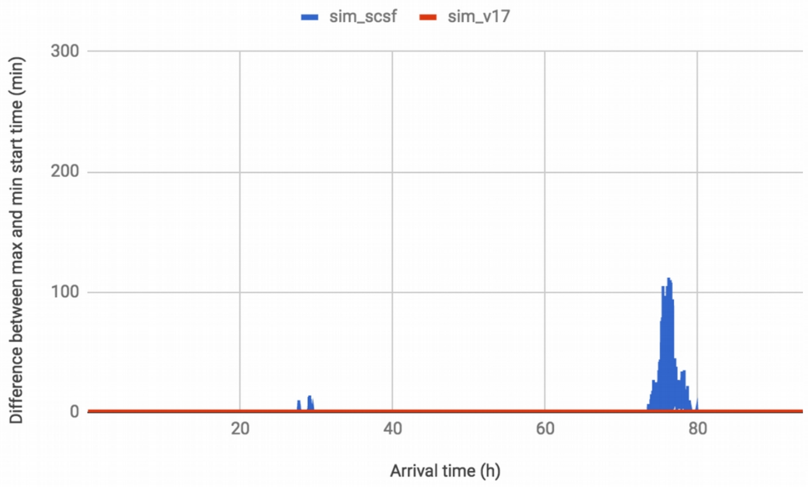
We compared *ScSF* simulator with our improved version

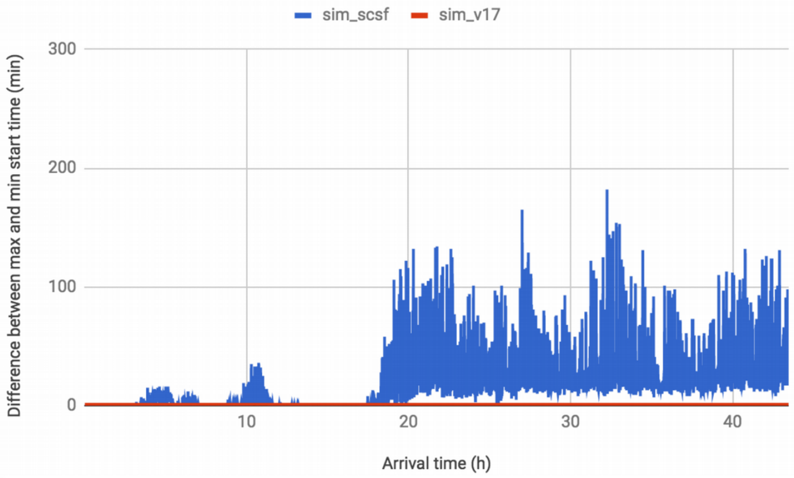
Evaluation: Consistency

- Our simulator is deterministic, we removed all the sources of error!

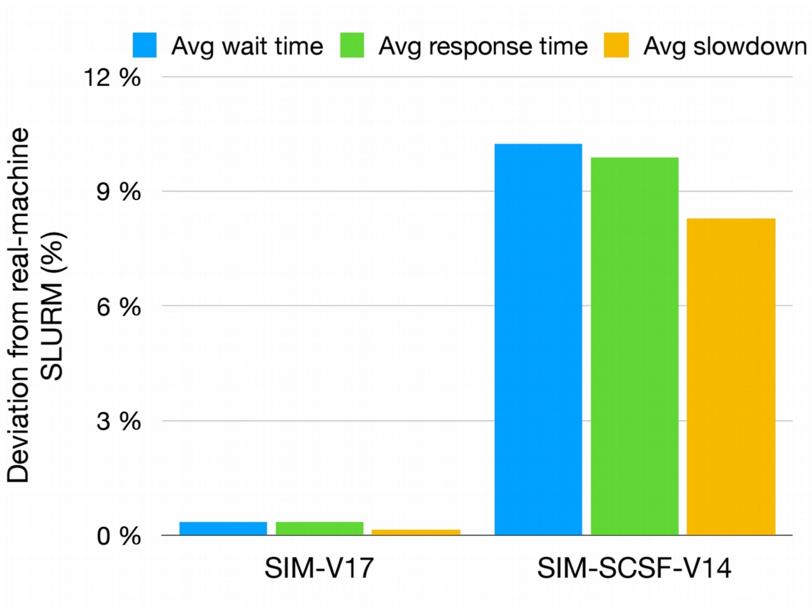


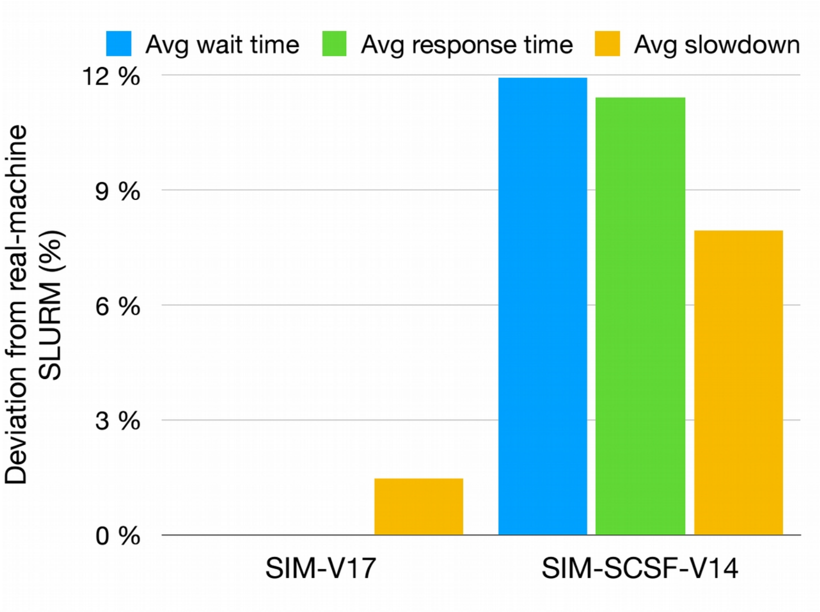


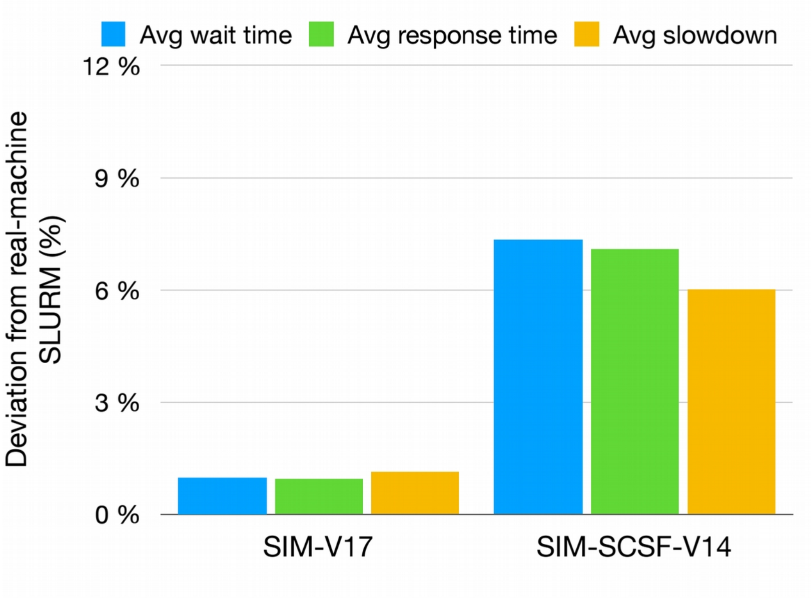


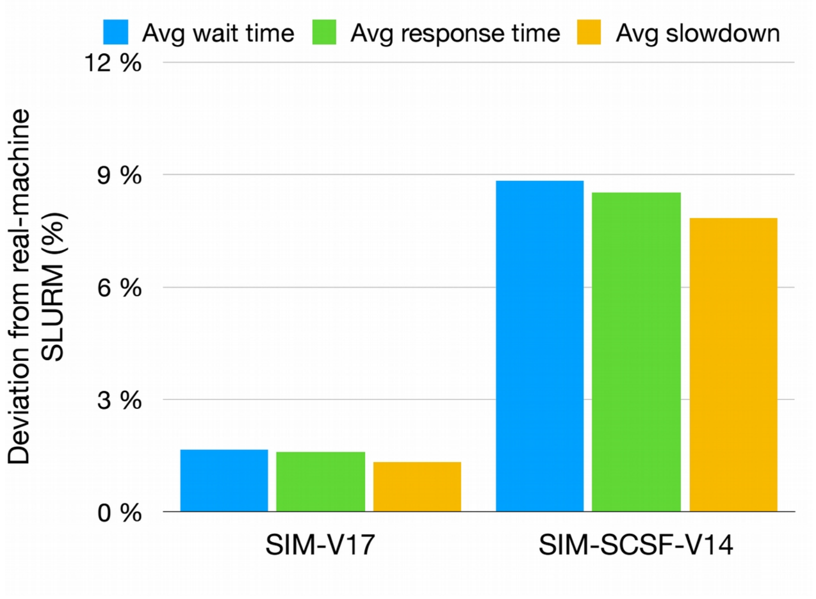


Evaluation: Accuracy

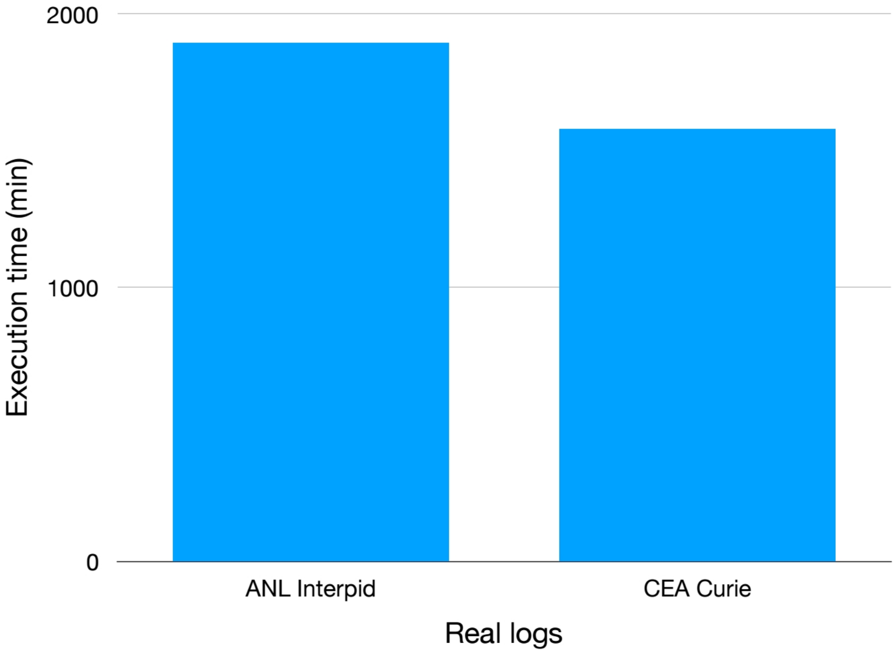


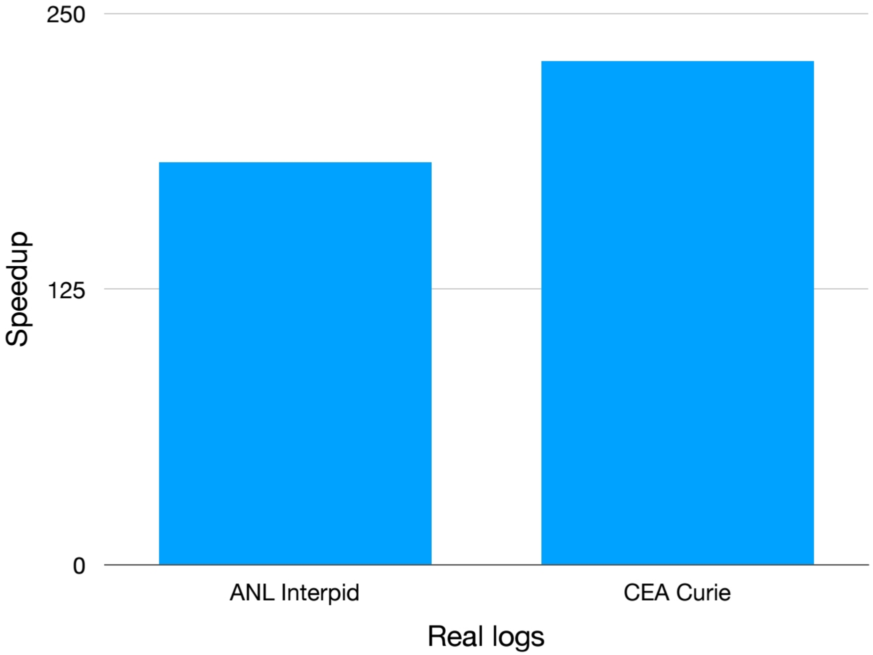






Evaluation: Performance





Barcelona Supercomputing Center
Centro Nacional de Supercomputación

Conclusion and future work

- SLURM Simulator is a powerful tool for research and system administration
- We improved the Simulator and we made it deterministic
- We evaluated its consistency, accuracy and performance
- A paper is in the evaluation process
- SLURM Simulator is used in European Projects (DEEP-EST)
- Future work
  - Further improve performance and accuracy
  - Event driven simulator, not updating time second by second
  - Git repository available soon
  - Improve support for failed or canceled jobs
  - Implement support for heterogeneous jobs in input

Open problems

- Simulator code needs to be improved and cleaned
  - Many #ifdefs around the code
  - Main source code modified
- Simulator code needs a maintained repository and documentation
  - Very difficult to understand how the simulator works
  - Many changes over time not documented

BSC Barcelona Supercomputing Center
Centro Nacional de Supercomputación

EXCELENCIA
SEVERO
OCHOA

Thank you

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