Using NERSC supercomputers for high energy nuclear physics applications with ALICE

Markus Fasel
Lawrence Berkeley National Laboratory

for the ALICE Collaboration

ACAT 2015, Valparaiso

Outline

- Introduction of NERSC HPC systems
- Cori, the next generation NERSC supercomputer
- Our tool for running jobs on NERSC HPC systems
- Our first performance comparison of NERSC HPC systems running ALICE applications

NERSC systems

Computing systems

Edison 2.58 PF

5576 Nodes
134k CPUs
Intel Xeon Ivy Bridge
24 cores / node
64 GB RAM / node
7.6 PB SCRATCH

Hopper 1.3 [illegible]

retired

[illegible] Opteron
24 cores / node
32 GB RAM / node
2.2 PB SCRATCH

PDSF

Batch farm
Mix of AMD and Intel CPUs
120 Nodes
~3k Cores
2-4 GB RAM / core

Common file systems

/project: 9.1 PB quota-based GPFS, for long term storage (not optimized for I/O)

/home: 275 TB, user home dirs

HPSS: 240 PB (max) tape file system for data archiving

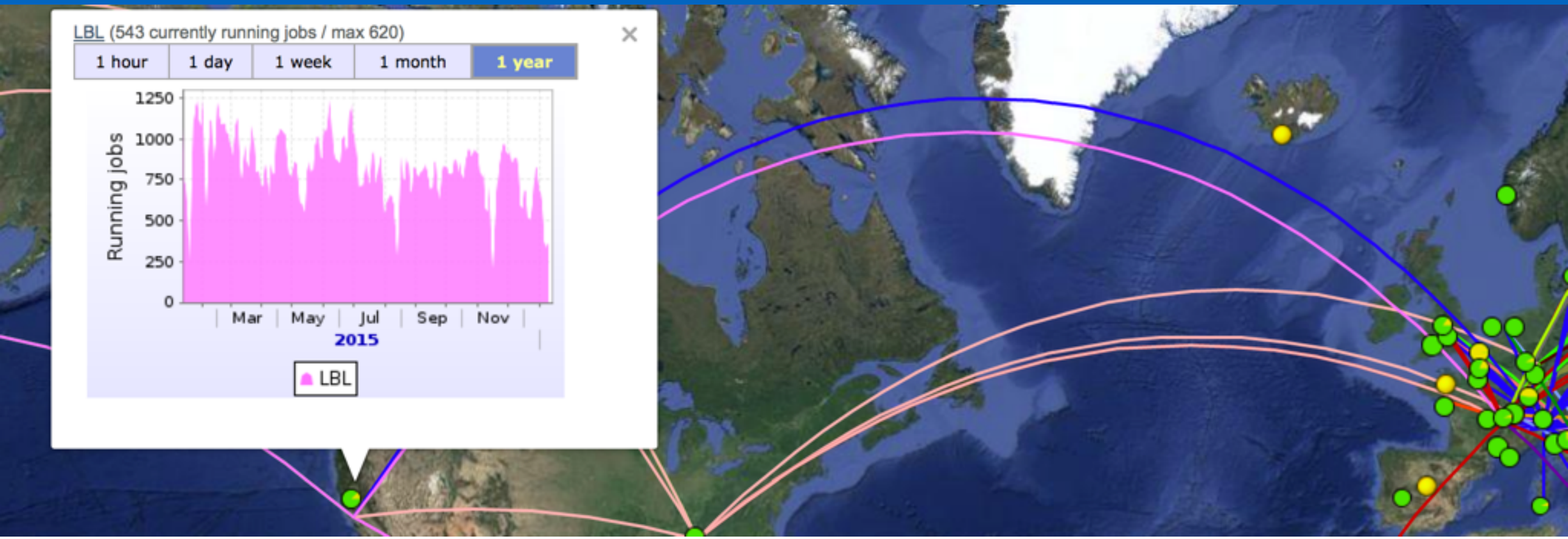
Central services

Data transfer nodes

4 nodes, 10 Gigabit wan connection per node

Science gateway for web apps

PDSF, the local batch farm

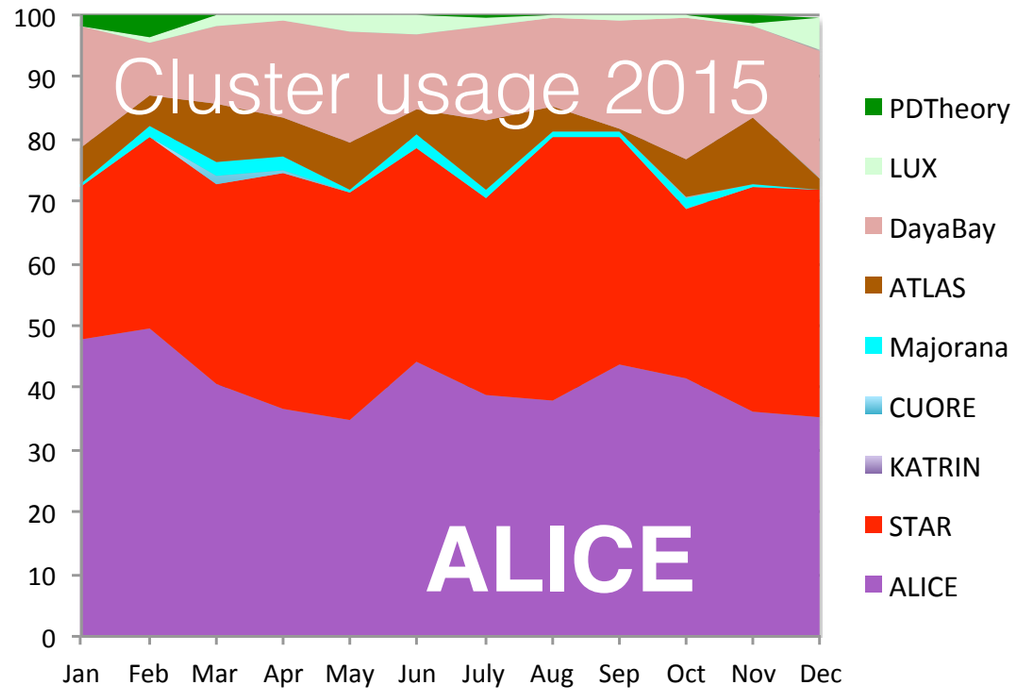


ALICE Tier2 Grid Site

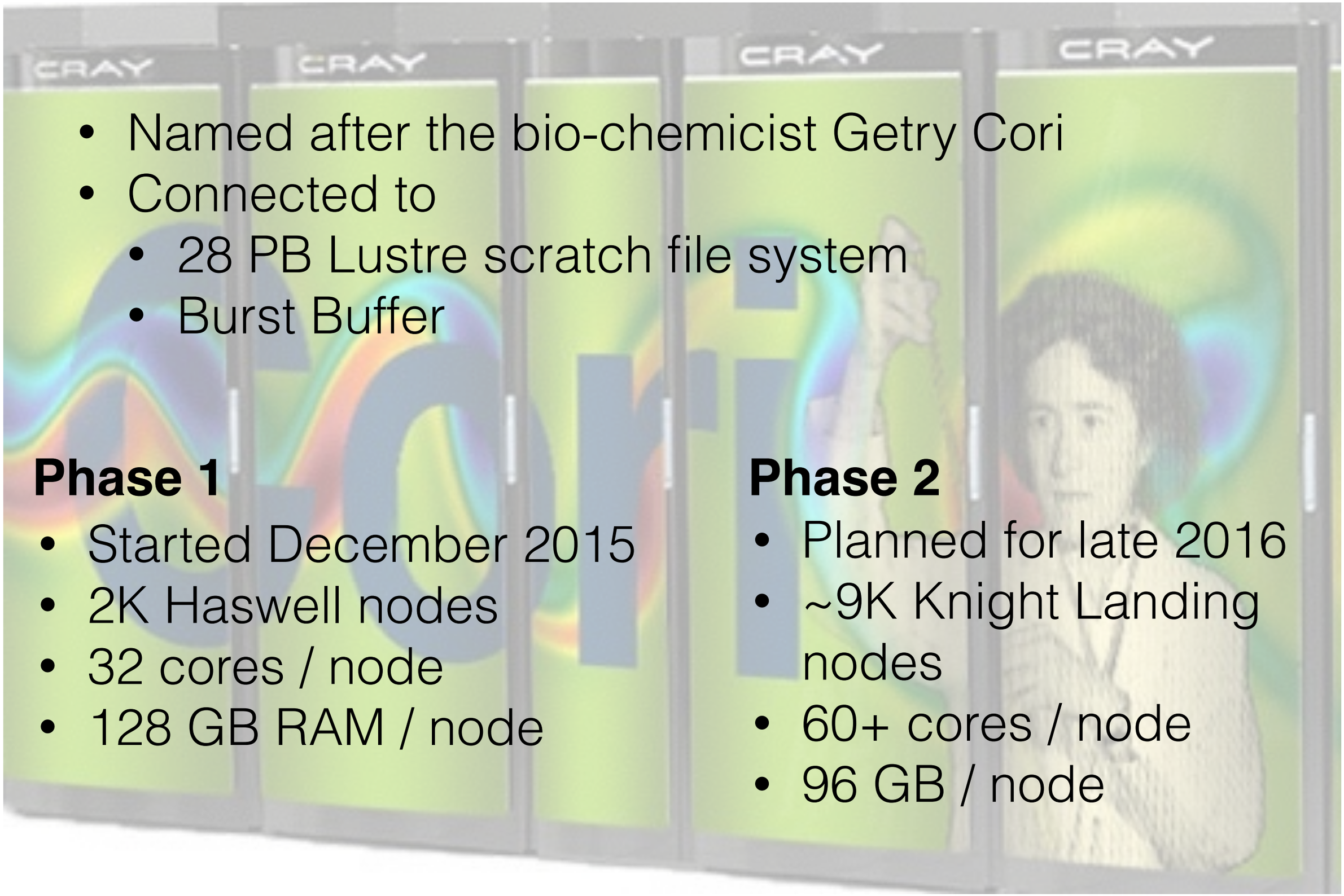
~800 Slots → ~40% of pdsf

In addition serving many other groups with US contribution

STAR, ATLAS, DayaBay …



Cori, the next generation NERSC supercomputer



- Named after the bio-chemicist Getry Cori
- Connected to
  - 28 PB Lustre scratch file system
  - Burst Buffer

Phase 1

- Started December 2015
- 2K Haswell nodes
- 32 cores / node
- 128 GB RAM / node

Phase 2

- Planned for late 2016
- ~9K Knight Landing nodes
- 60+ cores / node
- 96 GB / node

High-performance data processing with the burst buffer

File system for I/O intensive jobs

- Cray Data Warp technology
- SSD based
- 800 GB/s peak I/0
- Size
  - At Phase 1: 750 TB
  - At Phase 2: ~1.5 PB

Possible use cases:

- Cache storage for data files providing fast access during analysis jobs (I/O intensive jobs)
- Temporary storage for job output used in consecutive jobs
- …

→ **R&D project at NERSC**

Limitations in using supercomputers for HPC applications

- Bandwidth-limited outgoing network connection
  - However special high-performance scratch file systems
- Scheduling challenges
  - Job execution time
    - Whole-node scheduling necessary (accounting)
- Limited memory per CPU core
  - Historically very low memory / core
  - No swap

Adaption of the workflow

Motivations for the usage of HPC systems

Need for additional resources

- Local R&D
- sparse resources to to share with the grid

Opportunistic use of resources we can access

Explore how these type of resources fit into the ALICE grid model

- R&D
- AliceO^2 upgrade

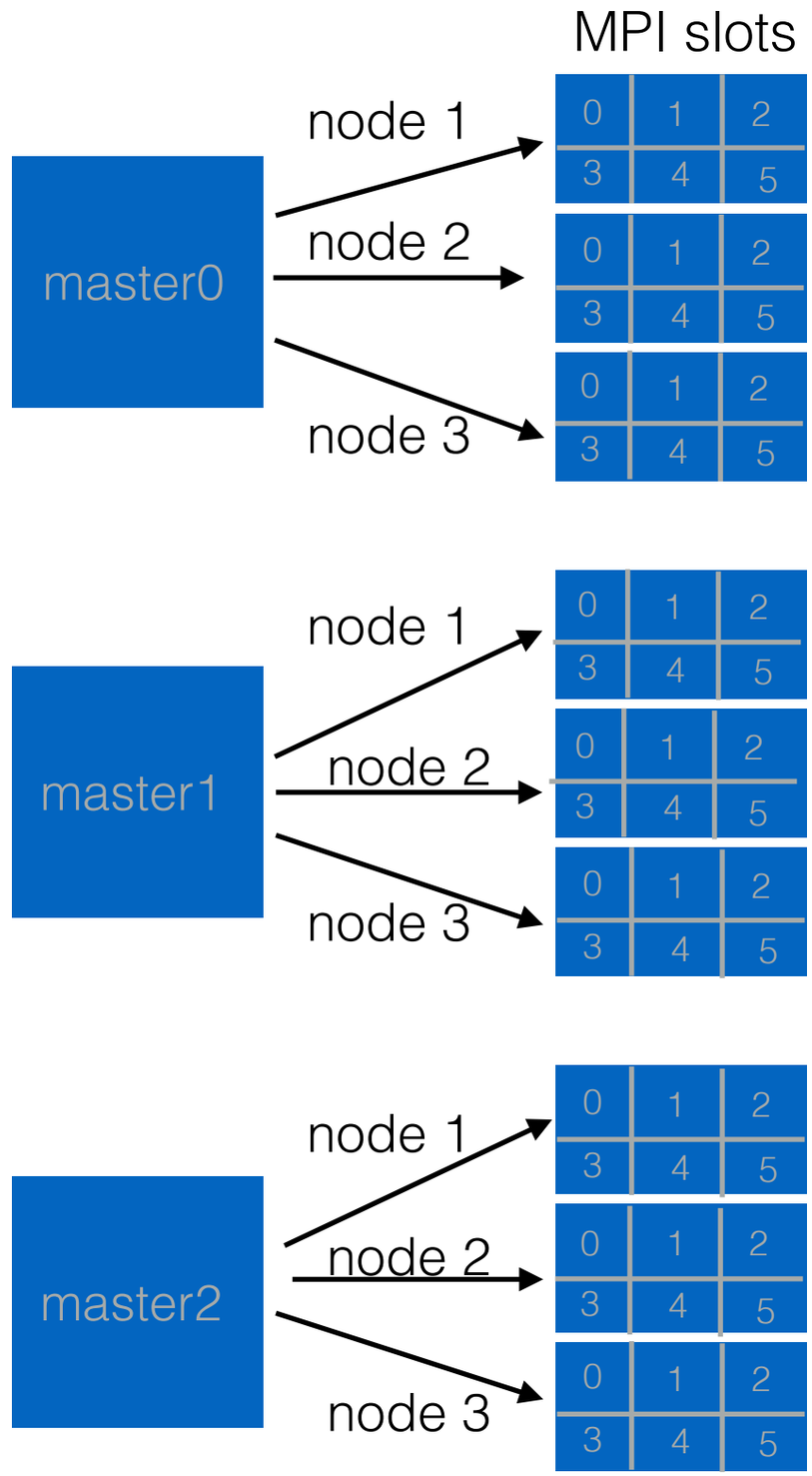
Our tool: ANALISA

Tool which runs multiple serial jobs as a MPI job

- Submitter:
  - Splits a master into n sub jobs
- Worker (MPI):
  - Runs the subjobs (payload)
- Job description: config, json, xml

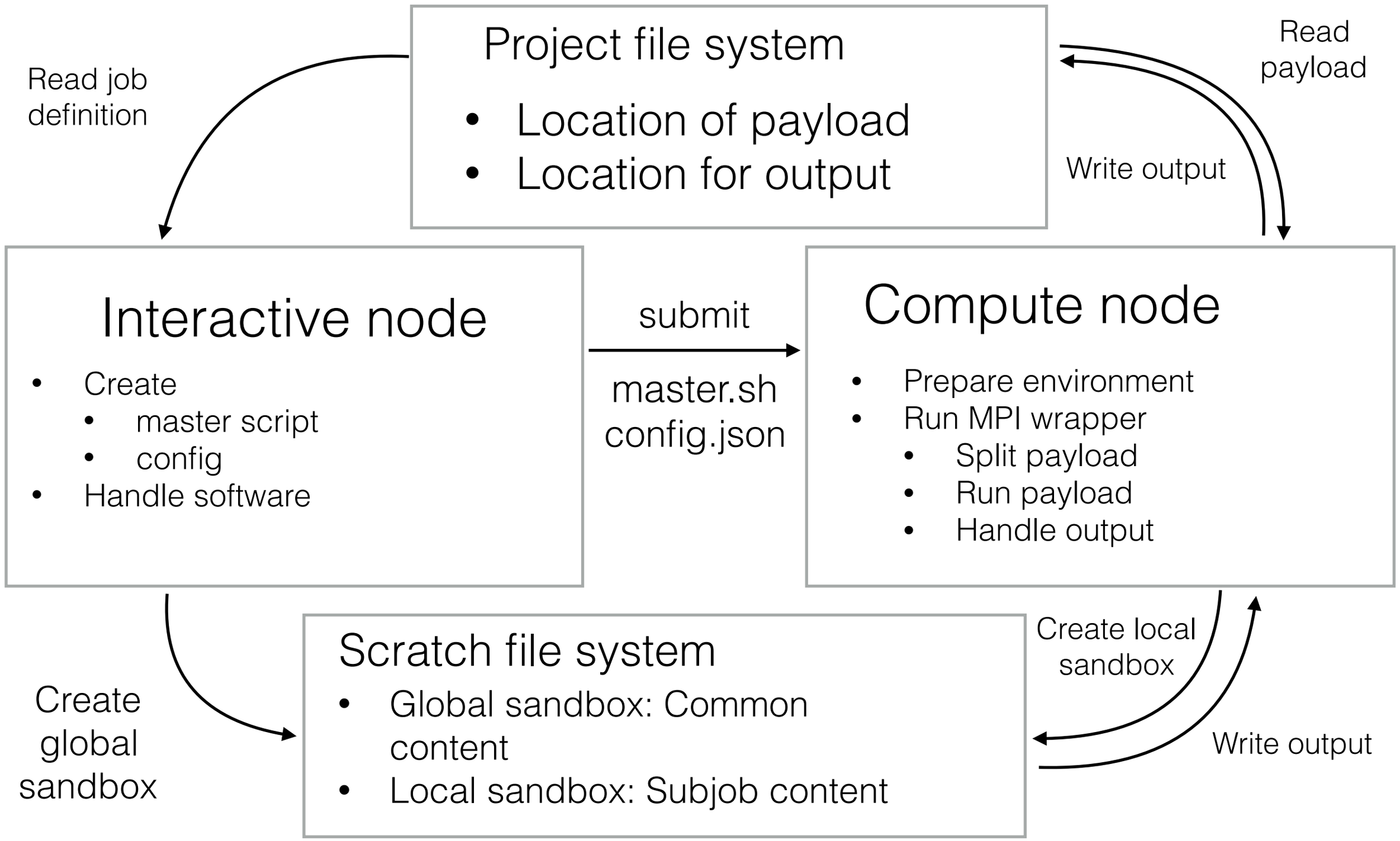
Key facts:

- PYTHON, mpi4py
- BSD-type license
- https://bitbucket.org/berkeleylab/analisa



Hiding complexity of resource management for the user

ANALISA Workflow



Job configuration

ANALISA needs to take care about:

- Job splitting
- Handling of input and output
- Payload

Need for a mechanism to configure job

- config.ali (Grid-JDL inspired)
- JSON
- XML

Example config.ali

```
outputbase:  /project/projectdirs/alice/mfasel/ppfast/PYTHIA8PtHard/result_pilot_test
executable:  /project/projectdirs/alice/mfasel/ppfast/PYTHIA8PtHard/source/runsim.sh
validation:  /project/projectdirs/alice/mfasel/ppfast/PYTHIA8PtHard/source/validation.sh
arguments:   "--event #jobid --pthardbin 1 --nevents 100"
packages:    ROOT/v5-34-08, GEANT3/v1-15a_rootv5-34-08, AliRoot/master
inputfiles:  /project/projectdirs/alice/mfasel/ppfast/PYTHIA8PtHard/source/simana.C
outputfiles: histos.root@root_archive.zip;sim.log,worker.log@log_archive.zip
njobs:       24
timelimit:   2.00.00
queue:       regular
jobname:     PYTHIA8PtHardPilot_test
```

Software handling

Packman

- Buildbot compiling software for different platforms
  - xml build recipe
  - sh build script
- Archived as tar files in software repository on file system
- Extracted to scratch file system before job execution

Pros:
- Under user control
- For user / unreleased software

Cons:
- Needs manual intervention
- Requires independent result validation

cvmfs via parrot / shifter

- Shifter:
  - Docker container with full copy of cvmfs content running on compute node
- Parrot:
  - Tool mounting a copy of the cvmfs file catalogue located on persistent file system under original path

Pros:
- For released software
- User not involved

Cons:
- Under control of the sys admins

Parrot: http://ccl.cse.nd.edu/software/parrot/
Shifter: http://www.nersc.gov/research-and-development/user-defined-images/

Performance tests on NERSC systems

Collision system

pp, pPb, PbPb at different centre-of-mass energies

Event type

min. Bias, jet-jet, force particle, force decay …

Type of ALICE simulation jobs

Generator

Pythia6/8, HIJING, DPMJET …

Transport

Geant3/4

ALICE Software version

ROOT5, GEANT, AliRoot

Mix of jobs representing reference use case

Test cocktail

4 Scenarios

- pp, $\sqrt{s}$ = 7 TeV:
  - PYTHIA6
  - Min. Bias
  - Tune Perugia 2011
- pp, $\sqrt{s}$ = 8 TeV:
  - PYTHIA8
  - Min. Bias
  - Tune Monash2013
- p-Pb, $\sqrt{s_{NN}}$ = 5.02 TeV:
  - DPMJET
  - Min. Bias
- Pb-Pb, $\sqrt{s_{NN}}$ = 5.02 TeV:
  - HIJING
  - Min. Bias

All except Pb-Pb: 100 events / job
Pb-Pb: 5 events / Job

Software version:

- Identical in all cases, based on ROOT v5-34-30 and GEANT3 v2-0

Condition database

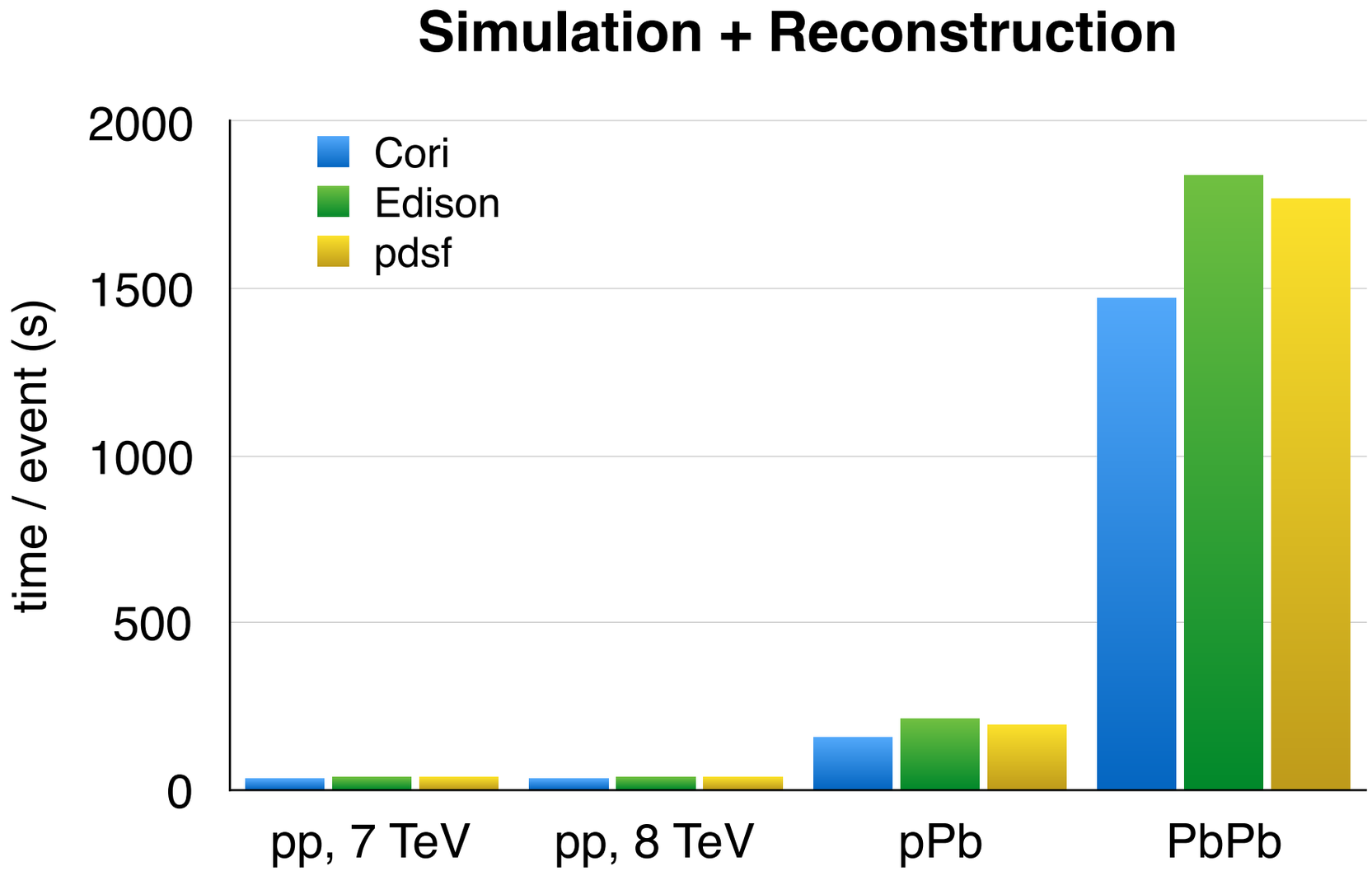
- Full copy on scratch storage for HPC clusters
- Full copy on project file system for PDSF

Job Parameters:

- Cori:
  - 20 Nodes, 32 jobs / Node
- Edison:
  - 26 Nodes, 24 jobs / Node
- PDSF:
  - 400 jobs / use case

Payload exactly as it runs on the grid!

Job execution time



High performance cluster are competitive compared to standard batch farms

PDSF has a mixture of different CPU types

- Same performance to Cori for jobs on same CPU type

Outlook: Integration into the ALICE grid

Proof of concept successfully done

Alien job

- Payload and job configuration is supplied to the worker remotely
- Software distribution centralized (CVMFS)

Test jobs

- Payload and job configuration on local file system
- Software build and distribution local

Integration development

- Pilot job (a precursor to the payload) requires access to the outside world
- Preparation of sandbox and software distribution
- Correct payload execution time limits estimates

Startup: Intended for simulation only

Software distribution with shifter and parrot

Shifter:

- Minimal SLC6 docker container
- 2 Images:
  - Only Software
  - Software + condition database

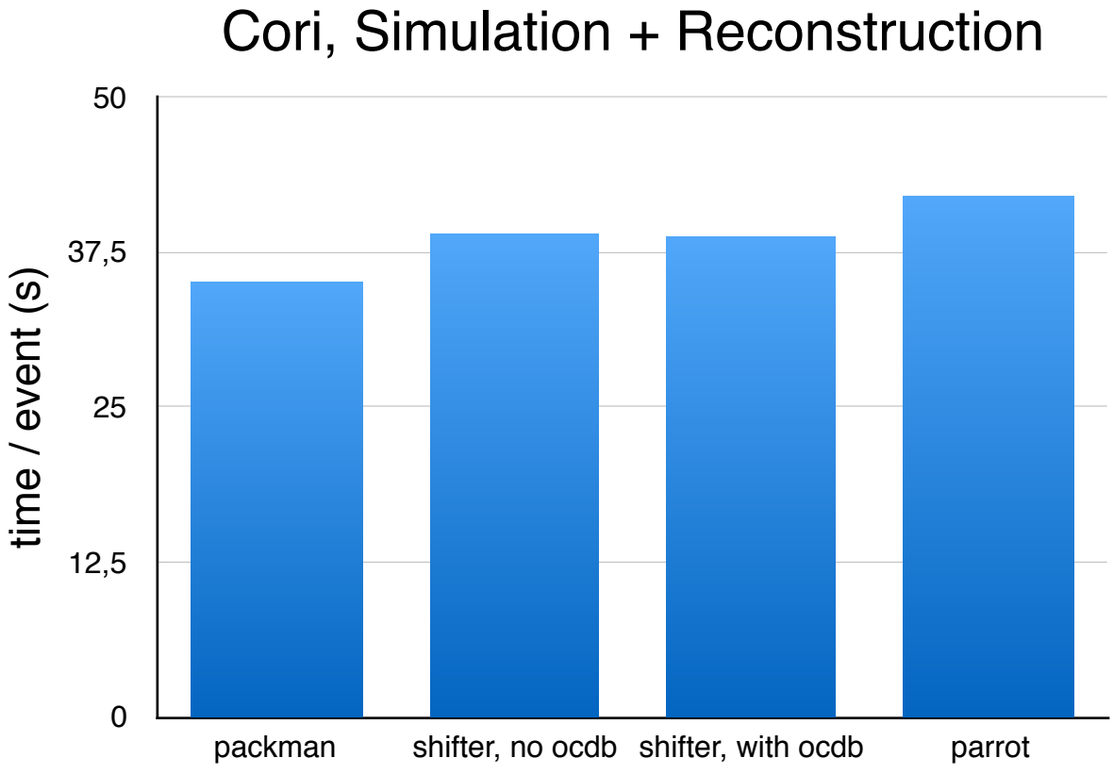
Data (software, condition database) part of the image!

Parrot:

Shifter used to provide a native SLC6 from which parrot is run

Data (software, condition database) external!

pp, $\sqrt{s}$ = 7 TeV Perugia2011 in all cases



First tests show that cvmfs be provided on Cori - optimizations ongoing

Cori Phase 2

- Different CPU (Intel Xeon Knight Landing compared to Intel Xeon Haswell)
- 60 cores / node, 96 GB memory
- Optimized for massively parallel processing

Code developed for Cori Phase 1 provides seamless access to Cori Phase 2

Ideal testbed for AliceO^2

Conclusions

- Tool for running serial payload on MPI clusters
- Successful test of ALICE jobs on NERSC supercomputer, including the newest Cori system
- Thanks to shifter, a native operating system environment is made available to the compute nodes
- Integration into the ALICE grid ongoing