SITE REPORT : ELECTRICITÉ DE FRANCE

Slurm User Group

Cécile Yoshikawa
September 27, 2016

HPC AT EDF

- About EDF
- How do we do HPC at EDF?
- Our HPC infrastructures
- Our in-house OS dedicated to scientific IT needs

ABOUT EDF

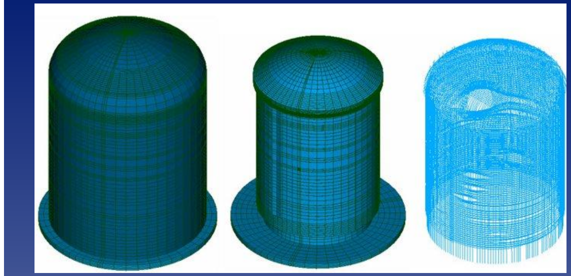
- **World's biggest electric utility**
  - 75B € in annual revenue, 37.6M clients worldwide
  - 160,000 employees worldwide

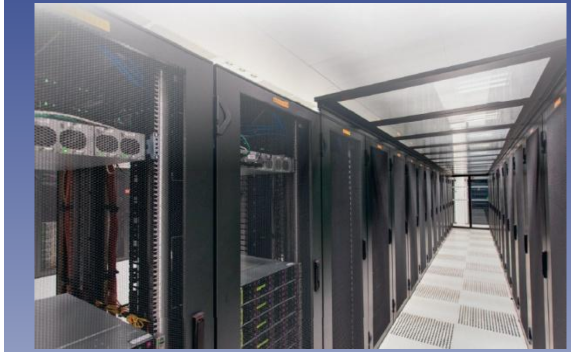
- **Main activities**
  - Electricity Generation & Engineering
  - Electricity Transmission & Distribution
  - Research & Development
  - Optimization & Trading
  - Products & Services

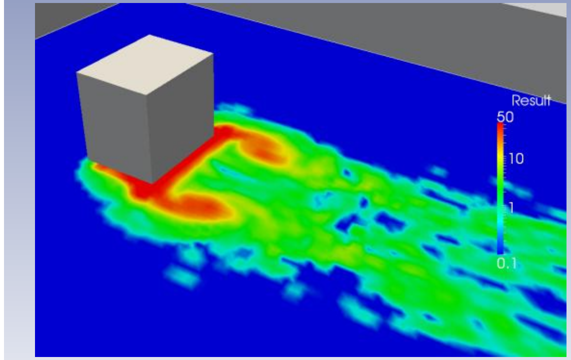
- **Importance of R&D and engineering divisions**
  - 650M € Net R&D budget in 2015
  - 541 patented & protected innovations

HPC AND SCIENTIFIC IT NEEDS

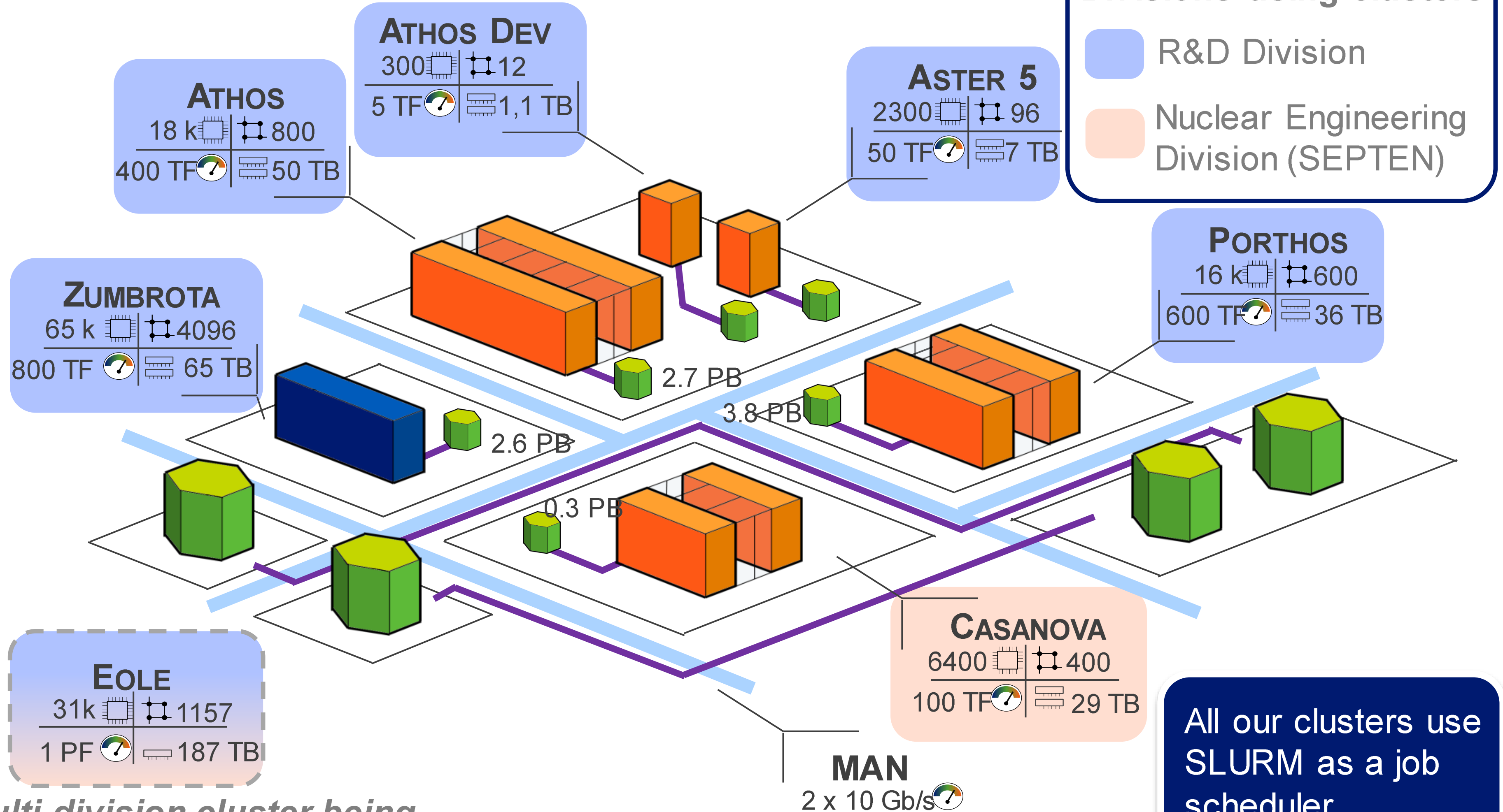
- **Modeling**
  - Approximate reality with a model
- **Simulations on a wide range of fields:**
  - Execution of numerical codes
  - Structural and fluid mechanics, neutronics for nuclear plant maintenance
  - Materials for renewable energies
- **In-house developed codes (Often Open Source):**
  - Structures and Thermomechanics Analysis: Code_Aster (www.code-aster.org)
  - CFD: Code_Saturne (www.code-saturne.org)
  - Pre and post-processing with SALOME (http://www.salome-platform.org)
- **High Performance Visualization**
  - Parallel rendering







6 EXISTING CLUSTERS FOR 2 DIVISIONS



ATHOS DEV
300 | 12
5 TF | 1,1 TB

ATHOS
18 k | 800
400 TF | 50 TB

ASTER 5
2300 | 96
50 TF | 7 TB

PORTHOS
16 k | 600
600 TF | 36 TB

ZUMBROTA
65 k | 4096
800 TF | 65 TB

CASANOVA
6400 | 400
100 TF | 29 TB

EOLE
31k | 1157
1 PF | 187 TB

2.7 PB

3.8 PB

2.6 PB

0.3 PB

MAN
2 x 10 Gb/s

Divisions using clusters

- 🟦 R&D Division
- 🟧 Nuclear Engineering Division (SEPTEN)

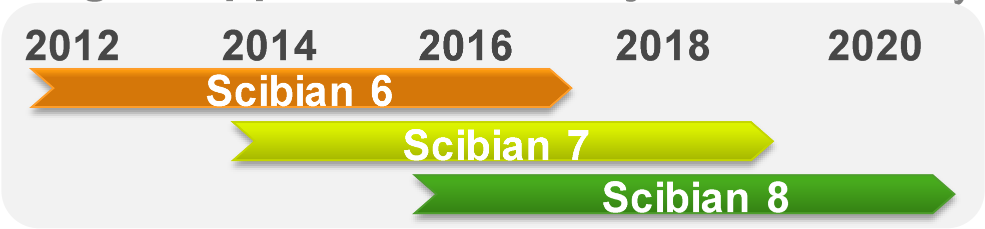
Multi-division cluster being installed for Jan. 2017

All our clusters use SLURM as a job scheduler

SCIBIAN

- **A Debian-based distribution: www.scibian.org**
  - Customizations to meet scientific IT needs
  - Initially an EDF custom distribution (Calibre)
    - Same distrib for workstations, servers & clusters
  - Being turned into an Open Source community project
    - Kick-off event on Sept. 30th at La Défense, Paris

- **Longer support for each major release:** beyond Oldstable



- **HPC with Scibian:**
  - Debian packaging of HPC dedicated SW:
    - GPFS, OFED, Mellanox IB stack, OPA to come
  - Custom Deployment System for diskless nodes
  - Tools on top of SLURM:
    - SLURM Dashboard, JobMetrics, NEOS

OUR SLURM USAGE AT EDF

- The functionalities we use
- A new challenge with our new cluster

DETAILS ON SLURM USAGES (1/2)

- **SLURM 15.08.8 on all clusters**

- **Separate partitions depending on node types**
  - Standard nodes
  - Large memory nodes
  - Graphical nodes with a GPU card

- **Several QOS on each cluster**
  - Selected partition
  - Required number of cores
  - Walltime

- **LUA Plugin for job submission**
  - Automatically route jobs into the proper QOS
  - https://github.com/edf-hpc/slurm-llnl-misc-plugins/tree/master/job_submit

DETAILS ON SLURM USAGES (2/2)

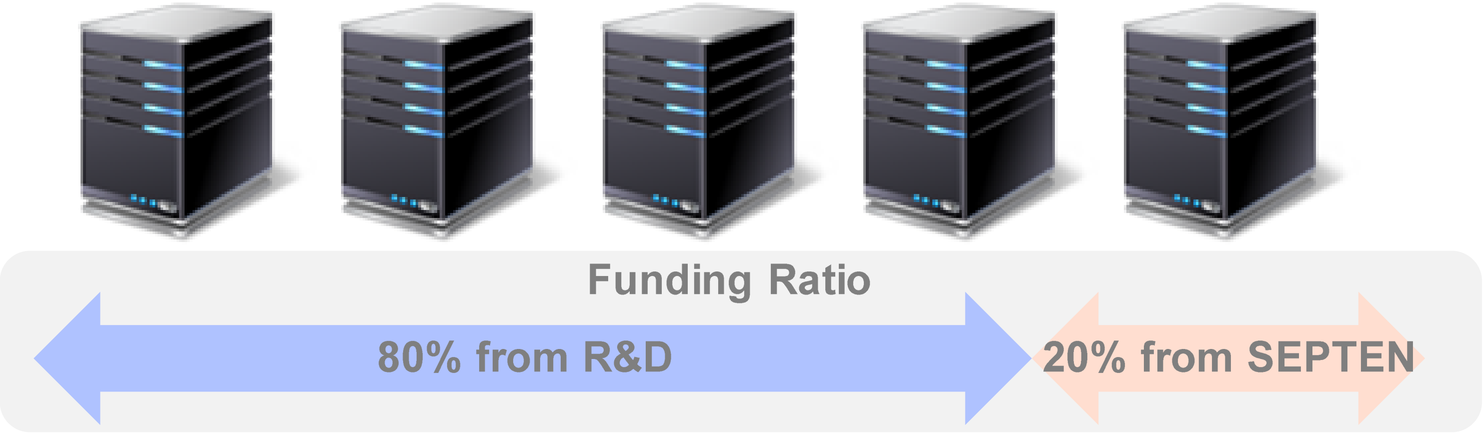
- **Accounting used on each cluster**
  - One dedicated database per cluster
    - Easy to maintain & to decommission
    - MariaDB in mode multi-master
  - One additional global PostgreSQL database collecting data from the per cluster databases, log files, LDAP information

- **Scheduling Policy**
  - Multi-factor Job Priority
  - Classic Fairshare Algorithm for existing clusters
  - Fair Tree Fairshare Algorithm for our new cluster

- **CPU and Memory as consumable resources**

- **Task Plugin: cgroup on the most recent clusters**
  - Memory controller (ConstrainRAMSpace=yes) to be used in our new cluster

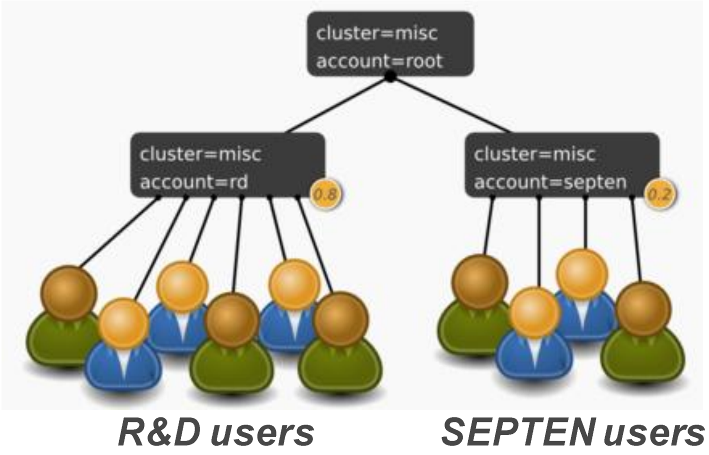
A NEW CLUSTER SHARED BY 2 DIVISIONS (1/2)

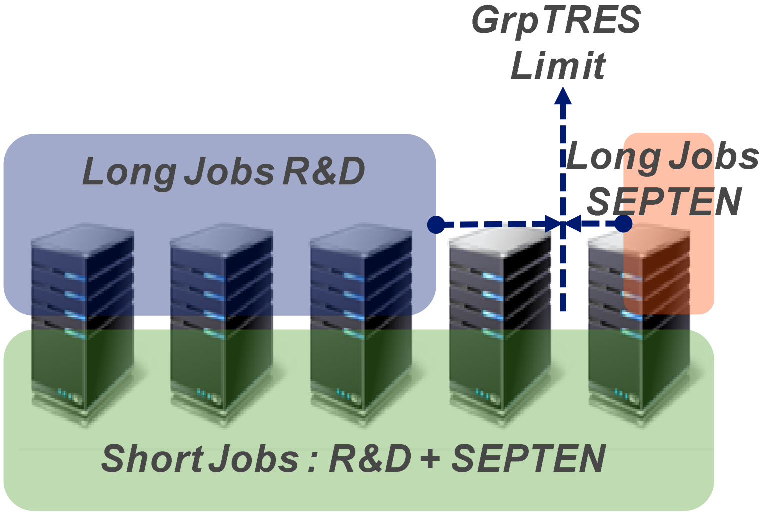


- **New constraints to share the resources**
  - Ideal resource ratio: 80% for R&D, 20% for SEPTEN
    - But no static sharing
  - If some resources of a division are unused, the other division should be able to use them
  - A division should be able to use all the resources it is entitled to within 8h

A NEW CLUSTER SHARED BY 2 DIVISIONS (2/2)

- **Solution to be implemented**
  - Fair Tree Fairshare Algorithm
    - PriorityFlag = FAIR_TREE
  - An account per division with fairshare factors according to the sharing ratio
  - Jobs are classified in 2 types :
    - Short jobs < 8h
    - Long jobs between 8h and 7 days
  - 1 QoS for short jobs shared between the 2 divisions
  - 2 QoS for long jobs, 1 for each division with a GrpTRES limit on the number of nodes
  - Higher priority for the short job QoS





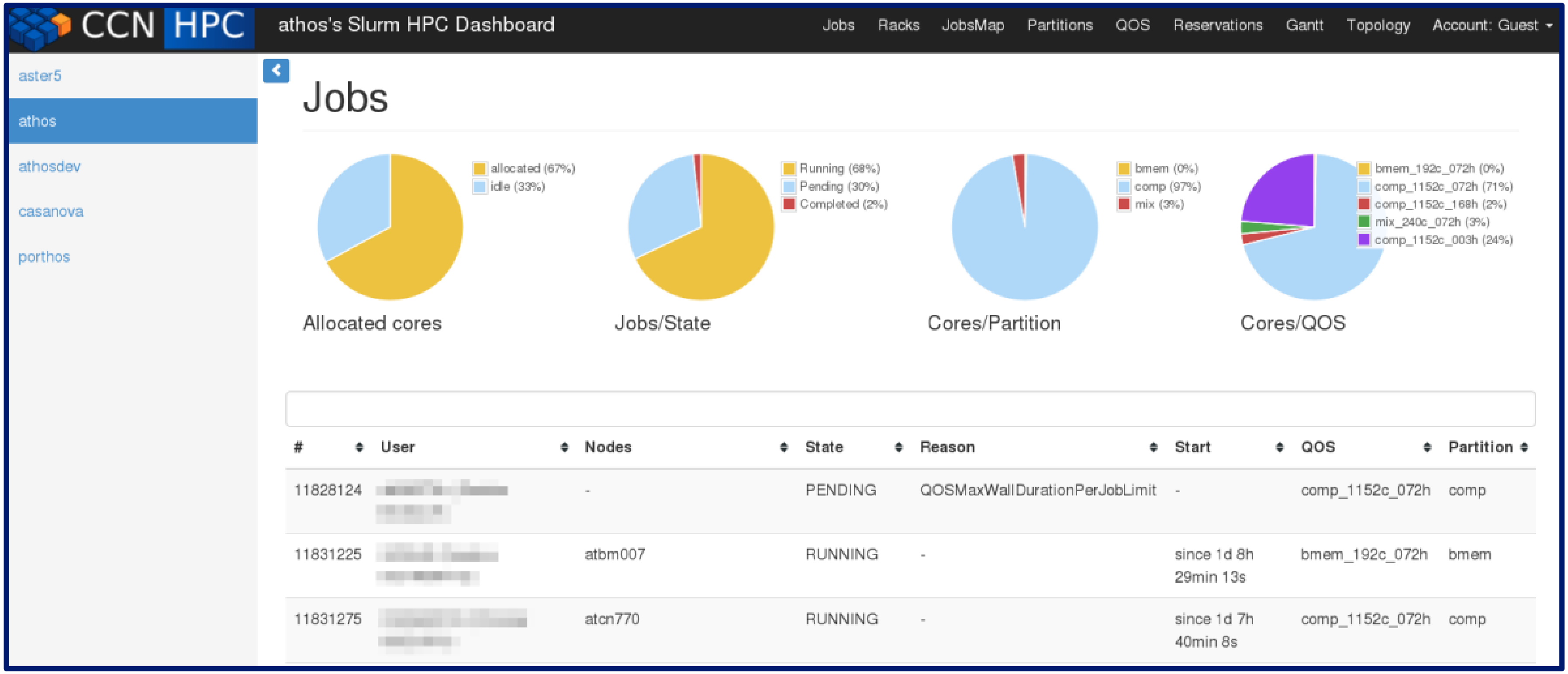
OUR IN-HOUSE DEVELOPED TOOLS TO WORK WITH SLURM

- **SlurmWeb**
- **JobMetrics**

SLURMWEB (1/4)

- A SLURM Dashboard for real time monitoring
- Sources: https://github.com/edf-hpc/slurm-web
- Documentation: https://edf-hpc.github.io/slurm-web

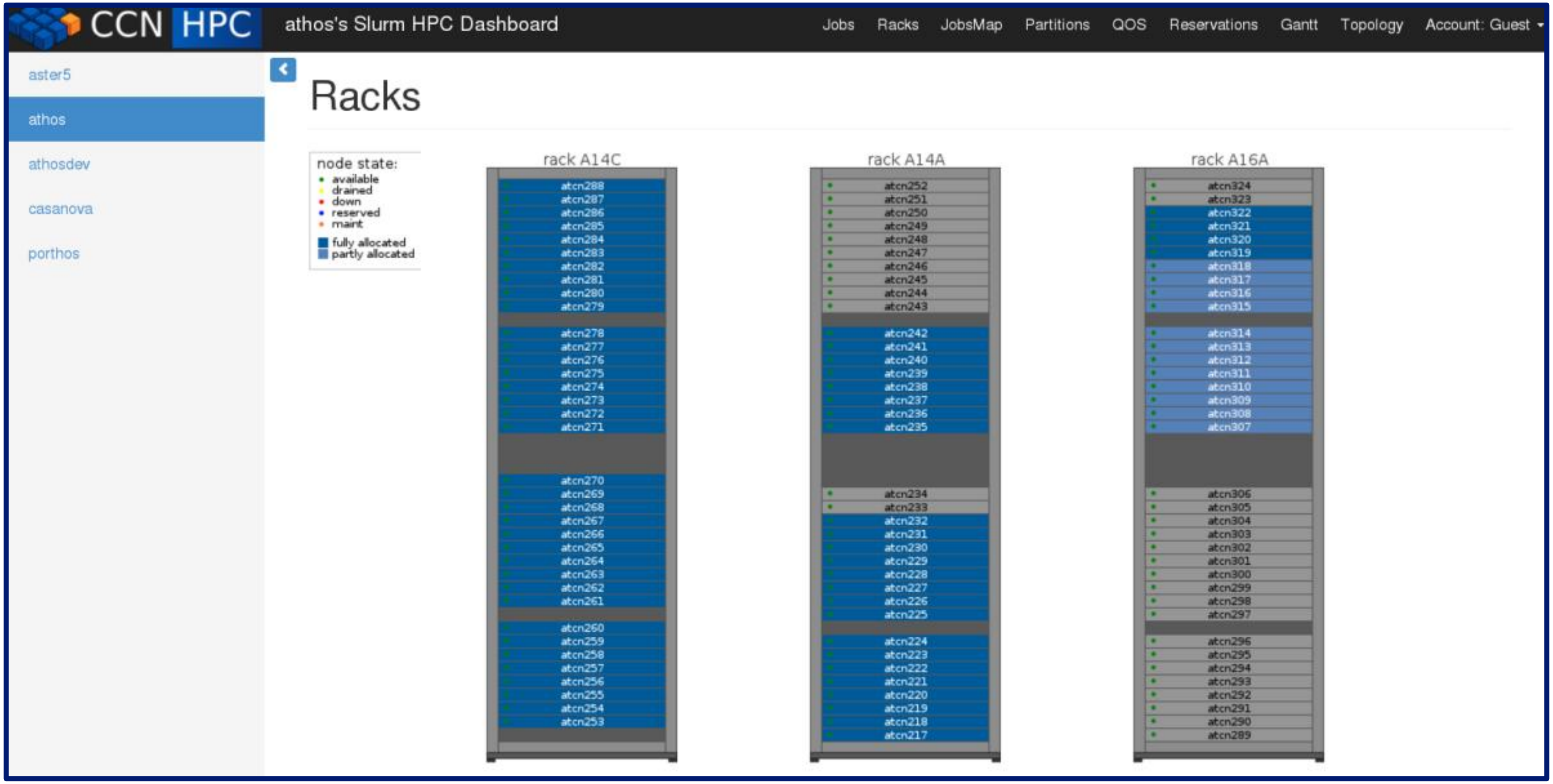
- Information about jobs



#	User	Nodes	State	Reason	Start	QOS	Partition
11828124	[redacted]	-	PENDING	QOSMaxWallDurationPerJobLimit	-	comp_1152c_072h	comp
11831225	[redacted]	atbm007	RUNNING	-	since 1d 8h 29min 13s	bmem_192c_072h	bmem
11831275	[redacted]	atcn770	RUNNING	-	since 1d 7h 40min 8s	comp_1152c_072h	comp

SLURMWEB (2/4)

- **Information about racks and nodes**



CCN HPC athos's Slurm HPC Dashboard

Jobs Racks JobsMap Partitions QOS Reservations Gantt Topology Account: Guest

aster5
athos
athosdev
casanova
porthos

Racks

node state:
● available
● drained
● down
● reserved
● maint
■ fully allocated
▨ partly allocated

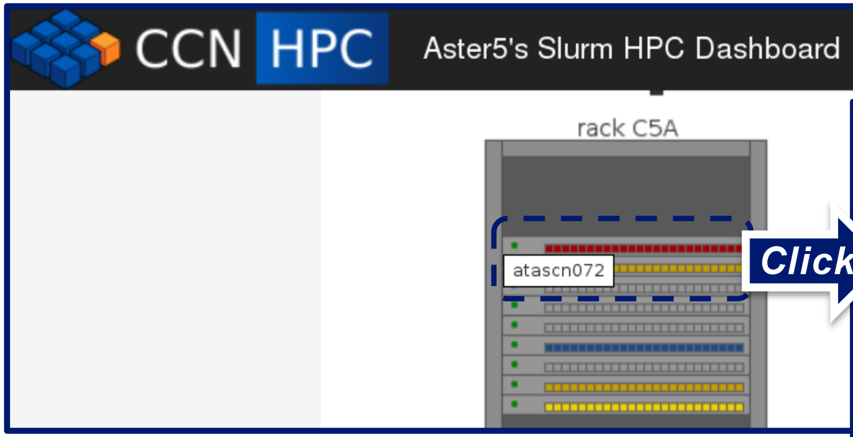
rack A14C
atcn288
atcn287
atcn286
atcn285
atcn284
atcn283
atcn282
atcn281
atcn280
atcn279
atcn278
atcn277
atcn276
atcn275
atcn274
atcn273
atcn272
atcn271
atcn270
atcn269
atcn268
atcn267
atcn266
atcn265
atcn264
atcn263
atcn262
atcn261
atcn260
atcn259
atcn258
atcn257
atcn256
atcn255
atcn254
atcn253

rack A14A
atcn252
atcn251
atcn250
atcn249
atcn248
atcn247
atcn246
atcn245
atcn244
atcn243
atcn242
atcn241
atcn240
atcn239
atcn238
atcn237
atcn236
atcn235
atcn234
atcn233
atcn232
atcn231
atcn230
atcn229
atcn228
atcn227
atcn226
atcn225
atcn224
atcn223
atcn222
atcn221
atcn220
atcn219
atcn218
atcn217

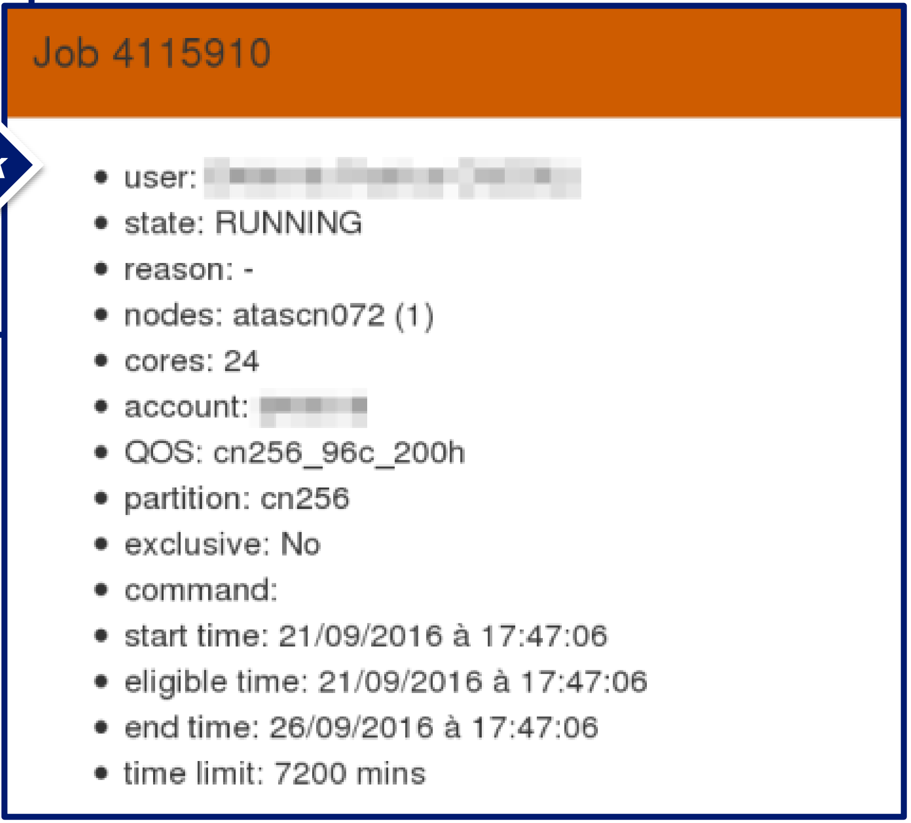
rack A16A
atcn324
atcn323
atcn322
atcn321
atcn320
atcn319
atcn318
atcn317
atcn316
atcn315
atcn314
atcn313
atcn312
atcn311
atcn310
atcn309
atcn308
atcn307
atcn306
atcn305
atcn304
atcn303
atcn302
atcn301
atcn300
atcn299
atcn298
atcn297
atcn296
atcn295
atcn294
atcn293
atcn292
atcn291
atcn290
atcn289

SLURMWEB (3/4)

- Mapping between nodes and jobs



Rack View with nodes running a job



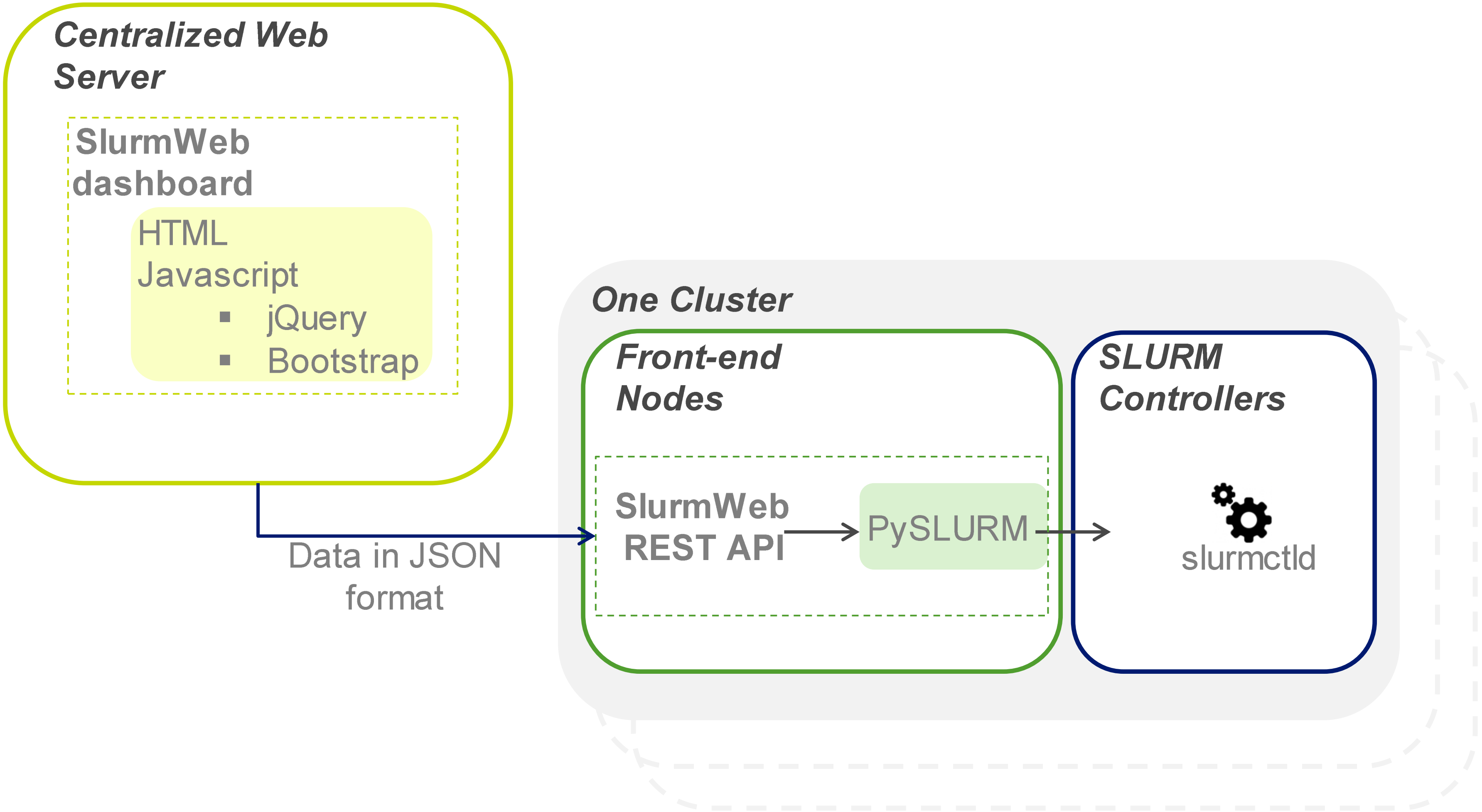
Job 4115910

- user: [redacted]
- state: RUNNING
- reason: -
- nodes: atascn072 (1)
- cores: 24
- account: [redacted]
- QOS: cn256_96c_200h
- partition: cn256
- exclusive: No
- command:
- start time: 21/09/2016 à 17:47:06
- eligible time: 21/09/2016 à 17:47:06
- end time: 26/09/2016 à 17:47:06
- time limit: 7200 mins

Information about the job running on the selected node

SLURMWEB (4/4)

- **Software Architecture**



Centralized Web Server

SlurmWeb dashboard

HTML

Javascript

- jQuery
- Bootstrap

Data in JSON format

One Cluster

Front-end Nodes

SlurmWeb REST API

PySLURM

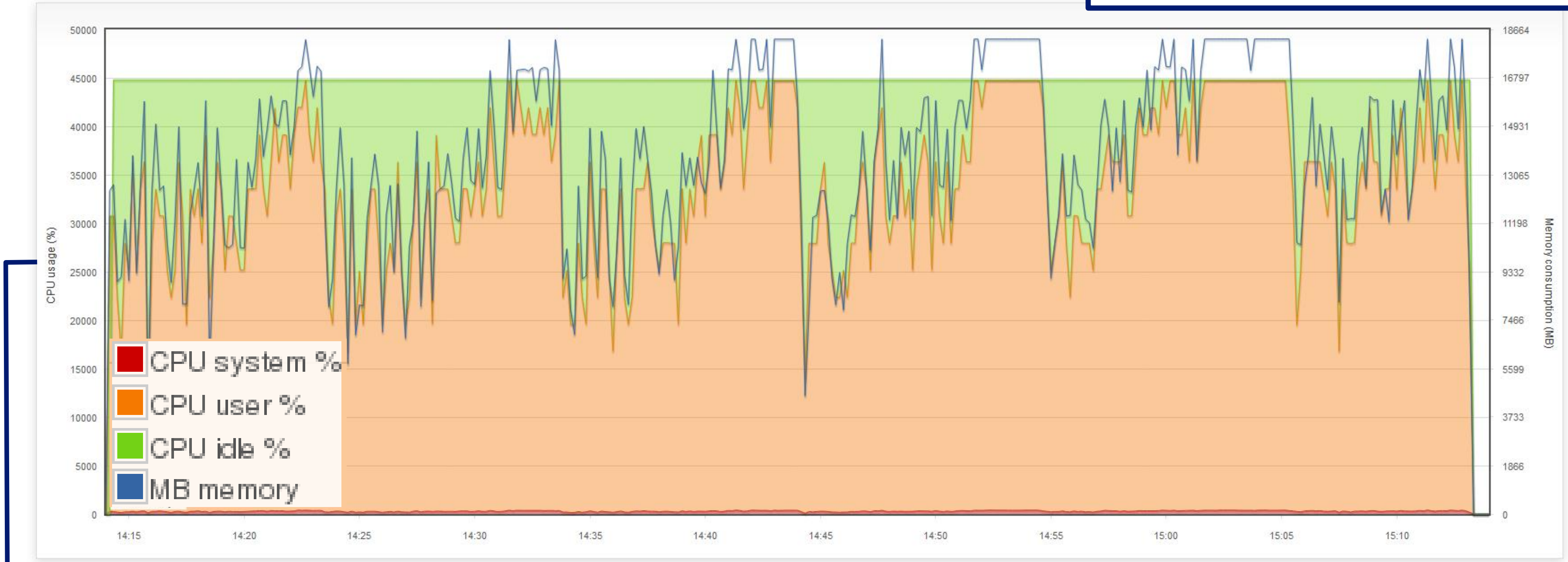
SLURM Controllers

slurmctld

JOBMETRICS (1/2)

- user: [redacted]
- state: RUNNING
- reason: -
- nodes: pocn[250,280,290-303] (16)
- cores: 448
- account: rdusers
- QOS: cn_0448c_024h
- partition: cn

HPC metrics: cluster porthos job 389734



etrics - Copyright © 2015 - CCN-HPC EDF SA

Real-time Total CPU Consumption of the job
100% : full consumption of <u>one core</u>

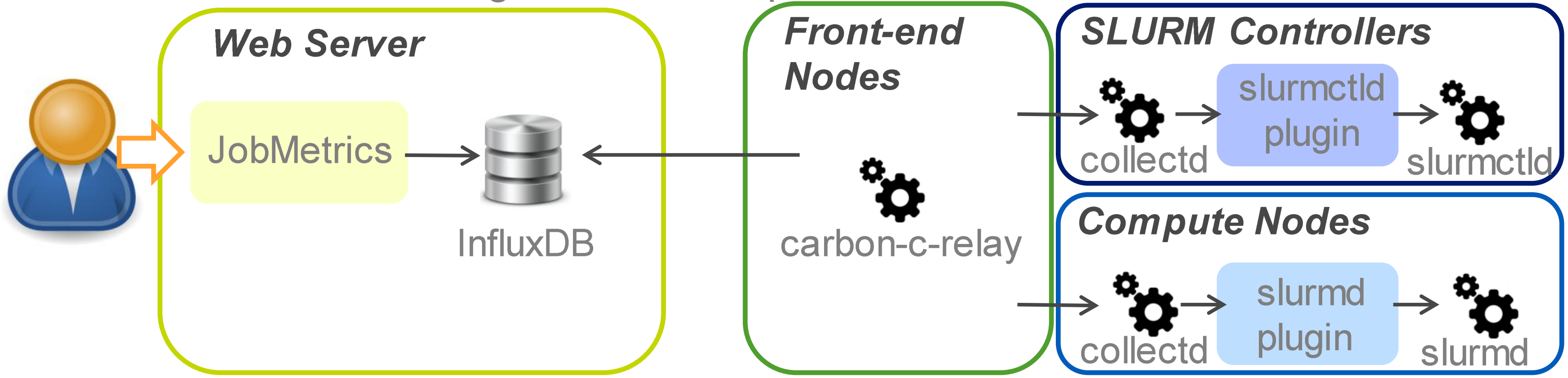
Real-time Total Memory Consumption of the job
Blue Graph

JOBMETRICS (2/2)

- **Web application to supply and display HPC job metrics such as**
  - Real-time CPU consumption for a job during its execution
  - Real-time memory consumption for a job during its execution

- **Prerequisites**
  - Use of *cgroups* task plugin to distinguish resource consumption in case of several jobs running simultaneously on one node

- **Implementation**
  - *Collectd* running on each computation node to collect metrics



- **Sources : https://github.com/edf-hpc/jobmetrics**

WHAT IS NEXT?

- SLURM jobs in containers

SLURM JOBS IN CONTAINERS

- **Initial Problem**
  - Natural OS life cycle
  - Some end-users want to use only qualified tools
    - Qualification sometimes takes a while
  - Developers want to test the newest tools available

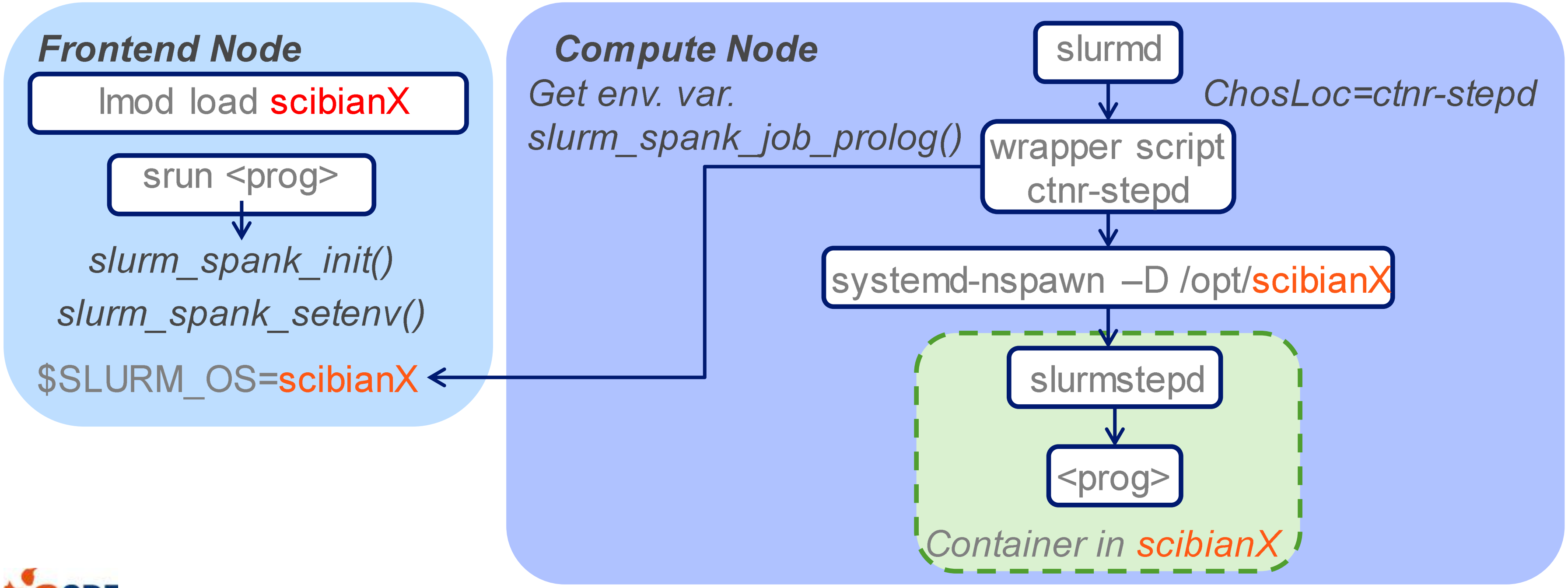
- **Goal**
  - Allow more flexibility at the end of life of one OS version
    - Be able to run jobs on an old OS version
  - Allow early code porting
    - Be able to run jobs on the upcoming OS version

=> Run jobs on several Scibian versions dynamically

- **Constraints**
  - Easy selection of the OS version
  - Serial and MPI jobs OK
  - No loss of performance

SLURM JOBS IN CONTAINERS

- **Technical approach**
  - **Usage**
    - Choice of OS version with an environment variable set up with *lmod*
  - **Containers**
    - systemd-nspawn to be launched by slurmd (*ChosLoc* parameter)



Frontend Node

lmod load scibianX

srun <prog>

slurm_spank_init()

slurm_spank_setenv()

$SLURM_OS=scibianX

Compute Node

Get env. var.

slurm_spank_job_prolog()

slurmd

ChosLoc=ctnr-stepd

wrapper script ctnr-stepd

systemd-nspawn –D /opt/scibianX

slurmstepd

<prog>

Container in scibianX

THANK YOU FOR LISTENING.
ANY QUESTIONS?

- All our tools are on Github:
  https://github.com/edf-hpc/

- Feel free to contact us:
  dsp-cspito-ccn-hpc@edf.fr
  cecile.yoshikawa@edf.fr