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](http://www.code-aster.org))
  - CFD: Code_Saturne ([www.code-saturne.org](http://www.code-saturne.org))
  - Pre and post-processing with SALOME ([http://www.salome-platform.org](http://www.salome-platform.org))

- **High Performance Visualization**
  - Parallel rendering
6 EXISTING CLUSTERS FOR 2 DIVISIONS

Divisions using clusters
- R&D Division
- Nuclear Engineering Division (SEPTEN)

ATHOS
- 18 k
- 800
- 400 TF
- 50 TB

ATHOS DEV
- 3000
- 12
- 5 TF
- 1,1 TB

ASTER 5
- 2300
- 96
- 50 TF
- 7 TB

ZUMBROTA
- 65 k
- 4096
- 800 TF
- 65 TB

PORTHOS
- 16 k
- 1600
- 600 TF
- 36 TB

EOLE
- 31 k
- 1157
- 1 PF
- 187 TB

CASANOVA
- 6400
- 400
- 100 TF
- 29 TB

MAN
- 2 x 10 Gb/s

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

<table>
<thead>
<tr>
<th>Year</th>
<th>Scibian 6</th>
<th>Scibian 7</th>
<th>Scibian 8</th>
</tr>
</thead>
<tbody>
<tr>
<td>2012</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2014</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2016</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2018</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2020</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- 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
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](https://github.com/edf-hpc/slurm-web)
- Documentation: [https://edf-hpc.github.io/slurm-web](https://edf-hpc.github.io/slurm-web)

- Information about jobs
SLURMWEB (2/4)

- Information about racks and nodes
Mapping between nodes and jobs

Rack View with nodes running a job

Information about the job running on the selected node
SLURMWEB (4/4)

- Software Architecture

Centralized Web Server

SlurmWeb dashboard

- HTML
- Javascript
  - jQuery
  - Bootstrap

One Cluster

Front-end Nodes

SlurmWeb

REST API

PySLURM

SLURM Controllers

slurmctld

Data in JSON format
HPC metrics: cluster porthos job 389734

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

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`
- `srin <prog>`
  - `slurm_spank_init()`
  - `slurm_spank_setenv()`
- `$SLURM_OS=scibianX`

**Compute Node**
- Get env. var.
  - `slurm_spank_job_prolog()`
- `slurmd`
  - `slurmstepd`
  - `systemd-nspawn -D /opt/scibianX`
  - `slurmstepd`
  - `<prog>`

**Diagram**
- Slurm workflow in containers.
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