Energy management on PlaFRIM

Supervisor : Brice Goglin (TADaaM)
Intern : Corentin Mercier
Summary

01. Overview of PlaFRIM's consumption
02. Shut down idle nodes with SLURM
03. Reactivity margin and custom timeouts
04. Energy saved thanks to the new system
05. User manual
06. How you can help us
Overview of PlaFRIM's consumption
PlaFRIM's share in the building consumption

Overview of Inria BSO electricity consumption

- Plafrim (nœuds)
- Plafrim (clim)
- Autres serveurs (racks)
- Autres serveurs (clim)
- Pompe à chaleur (chauffage)
- Pompe à chaleur (clim)
- Local hydraulique
- Autre (part jours travaillés)
- Autre (part constante)

500 MWh / year (50 x 75 m² apartment)

Source: cldd-bso@inria.fr
# PlaFRIM usage overview

## Machine utilization per node group

<table>
<thead>
<tr>
<th></th>
<th>arm</th>
<th>bora</th>
<th>brise</th>
<th>diablo</th>
<th>kona</th>
<th>miriel</th>
<th>mistral</th>
<th>sirocco</th>
<th>souris</th>
<th>visu</th>
<th>zonda</th>
</tr>
</thead>
<tbody>
<tr>
<td>%</td>
<td>2%</td>
<td>49%</td>
<td>17%</td>
<td>21%</td>
<td>3%</td>
<td>28%</td>
<td>26%</td>
<td>39%</td>
<td>19%</td>
<td>4%</td>
<td>36%</td>
</tr>
</tbody>
</table>

- idle nodes consumption = 128 344 kWh
- In 2021, 1 kWh = 0.11 €
  > money used to power idle nodes = 14 118 €
## Power saving strategies

### Non-exhaustive list of strategies and their impact

<table>
<thead>
<tr>
<th>Strategy</th>
<th>Impact</th>
</tr>
</thead>
<tbody>
<tr>
<td>Shut down idle nodes</td>
<td>High</td>
</tr>
<tr>
<td>Reduce CPU frequency during jobs</td>
<td>Moderate</td>
</tr>
<tr>
<td>Use the &quot;powersave&quot; governor on idle nodes</td>
<td>Low</td>
</tr>
<tr>
<td>Overprovisioning</td>
<td>Low</td>
</tr>
</tbody>
</table>
# Power saving strategies

## Non-exhaustive list of strategies and their impact

<table>
<thead>
<tr>
<th>Strategy</th>
<th>Impact</th>
</tr>
</thead>
<tbody>
<tr>
<td>Shut down idle nodes</td>
<td>High</td>
</tr>
<tr>
<td>Reduce CPU frequency during jobs</td>
<td>Moderate</td>
</tr>
<tr>
<td>Use the &quot;powersave&quot; governor on idle nodes</td>
<td>Low</td>
</tr>
<tr>
<td>Overprovisioning</td>
<td>Low</td>
</tr>
</tbody>
</table>
Shut down idle nodes with SLURM
SLURM power saving mechanism

1. Identify nodes which have been idle for at least **SuspendTime** seconds.
2. Execute **SuspendProgram** with an argument of the idle node names.
3. Identify the nodes which are in power save mode, but have been allocated to jobs.
4. Execute **ResumeProgram** with an argument of the allocated node names.
5. If the node fails to respond within **SlurmdTimeout**, the node will be marked DOWN and the job **requeued** if possible.

*NB: Every name in bold is a variable in slurm.conf*
SLURM's mechanism limits

- New interactive jobs get to wait for nodes to power up
  - some nodes should remain idle to serve small jobs (reactivity margin)
- **SuspendTime** is the same for every node
  - special nodes are used for an extended time even if not allocated
  - arm01, souris, etc
Reactivity margin and custom timeouts
Reactivity margin : main difficulty

- Only read access to each node idle counter
  > `scontrol show node (LastBusyTime)`
- One way to write to it
  > make an allocation via `salloc` / `srun`
Reactivity margin

[Business days]

date_range = NOT Holidays AND NOT Saturdays AND NOT Sundays
hour_range = 8:00 to 17:30
bora_margin = 4 # will keep 4 bora idle
miriel_margin = 1

[Holidays]
date_range = 2022/07/15 to 2022/08/20 OR 2022/12/15 to 2023/01/04
#hour_range = 4:00 to 5:00 # no hour_range defined -> section valid all day
bora_margin = 1
keep_nodes = sirocco[04-25],miriel087
Reactivity margin

[Business days]
```plaintext
date_range = NOT Holidays AND NOT Saturdays AND NOT Sundays
hour_range = 8:00 to 17:30
bora_margin = 4 # will keep 4 bora idle
miriel_margin = 1
```

[Holidays]
```plaintext
date_range = 2022/07/15 to 2022/08/20 OR 2022/12/15 to 2023/01/04
#hour_range = 4:00 to 5:00 # no hour_range defined -> section valid all day
bora_margin = 1
keep_nodes = sirocco[04-25],miriel087
```

- srun -N4 -C bora -job-name=keepIdle true
- srun -N1 -C miriel -job-name=keepIdle true
Custom timeouts

● Based on a "registration" system
  > begins when a wanted node is up
  > sends small jobs until the end of the "registration"
  > delete the "registration" after its end + SuspendTime

● Each section is a node list
  > define the number of hours and/or days that you want
  > each node of the list will stay up for the time wanted
Custom timeouts: example

- SuspendTime = 30 minutes, custom timeout = 1 hour
- Without the system
Custom timeouts: example

- SuspendTime = 30 minutes, custom timeout = 1 hour
- The script is called every 15 minutes
Custom timeouts : example

- The script is called every **15 minutes**
- Node busy **15 minutes** after end of registration -> new registration will occur
Limits of the system

- Based on job allocation
  > quicker increase in job IDs
  > create useless entries in SLURM database
  > may cause a denial of service on very large clusters
Energy saved thanks to the new system
Overview of the energy saved

<table>
<thead>
<tr>
<th>Suspend Time</th>
<th>arm</th>
<th>bora</th>
<th>brise</th>
<th>diablo</th>
<th>kona</th>
<th>miriel</th>
<th>mistral</th>
<th>sirocco</th>
<th>souris</th>
<th>visu</th>
<th>zonda</th>
<th>saved kWh</th>
<th>saved €</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>2%</td>
<td>49%</td>
<td>17%</td>
<td>21%</td>
<td>3%</td>
<td>28%</td>
<td>26%</td>
<td>39%</td>
<td>19%</td>
<td>4%</td>
<td>36%</td>
<td>128 344</td>
<td>14 118</td>
</tr>
<tr>
<td>1 hour</td>
<td>3%</td>
<td>55%</td>
<td>18%</td>
<td>22%</td>
<td>3%</td>
<td>30%</td>
<td>28%</td>
<td>42%</td>
<td>19%</td>
<td>5%</td>
<td>38%</td>
<td>122 675</td>
<td>13 494</td>
</tr>
<tr>
<td>2 hours</td>
<td>4%</td>
<td>59%</td>
<td>19%</td>
<td>24%</td>
<td>4%</td>
<td>32%</td>
<td>30%</td>
<td>43%</td>
<td>20%</td>
<td>5%</td>
<td>40%</td>
<td>118 123</td>
<td>12 994</td>
</tr>
<tr>
<td>4 hours</td>
<td>5%</td>
<td>66%</td>
<td>21%</td>
<td>26%</td>
<td>5%</td>
<td>35%</td>
<td>33%</td>
<td>47%</td>
<td>22%</td>
<td>7%</td>
<td>43%</td>
<td>110 513</td>
<td>12 156</td>
</tr>
</tbody>
</table>

Proportion of powered nodes in their group according to SuspendTime
User manual
New node states

- New symbols will be present when running `sinfo`
  - `#`: the node is **powering up**
  - `%`: the node is **powering down**
  - `~`: the node is **down**

- Example

```
cmercie2@miriel045:~$ sinfo
PARTITION AVAIL TIMELIMIT NODES STATE NODELIST
routage*  up 3-00:00:00:00 1 idle% sirocco25
routage*  up 3-00:00:00:00 2 idle# diablo04,miriel087
routage*  up 3-00:00:00:00 2 idle~ bora[040,044]
routage*  up 3-00:00:00:00 2 idle miriel088,zonda21
```
salloc / srun

- What happens if a node that you requested is down?
  > srun blocks and wait for all your nodes then your job begins
  > salloc returns immediately
    - you have your allocation!
    - you can't connect until all your nodes are ready

```bash
smercine2@miriel045:~$ salloc -N2 -C miriel
salloc: Granted job allocation 498108
[498108] > smercine2@miriel045:~$ ssh miriel087
Access denied by pam_slurm_adopt: you have no active jobs on this node
Authentication failed.
[498108] > smercine2@miriel045:~$ ssh miriel087
Last login: Fri Aug 12 13:38:31 2022 from miriel045.formation.cluster
```
squeue and node failure

● Jobs sent by the system are called "keepIdle"
  > shouldn't last in the queue
  > if you see too many of them, there's a problem
● Node can fail to boot
  > put in down~ state

● For more information, check the new section 3.10 on the PlaFRIM documentation!
  > https://plafrim-users.gitlabpages.inria.fr/doc/#energy
How you can help us
Beta-test on Formation

- The system is under beta-testing!
  - on the "Formation" cluster
  - send us an email to get registered
  - we would like any feedback to improve the system
CPU frequency and consumption
CPU frequency and consumption

- Reducing a little the frequency leads to high savings
- You can easily reduce the CPU frequency with SLURM
  - `salloc -N1 -C bora -cpu-freq=HighM1`
  - `salloc -N1 -C bora -cpu-freq=2400000` (2.4 GHz)
- The highest the frequency, the highest the savings
  - No real benefit if the max. frequency is low
- !\ Some machines only accepts specific frequencies
  - `/sys/devices/system/cpu/cpuX/cpufreq/scaling_available_frequencies`
Thank you!

Feel free to ask any question!