Machine Learning for HPC

Théo SAILLANT[1,2] — CEA, DAM, DIF, F-91297 Arpajon, France
Jean-Christophe WEILL[1] — Université Paris-Saclay, ENS Paris-Saclay, CNRS, Centre Borelli, F-94235 Cachan, France
Nicolas VAYATIS[2]
Mathilde MOUGEOT[2,3] — ENSIIE, 91000 Evry, France

To have the appropriate tools for the operation and maintenance of computing centers by 2020, the analysis of massive data produced by the supercomputer seems essential.
We propose here to analyze two types of data: the metadata of the calculations submitted to predict the power consumed and the text logs emitted by the computing nodes to check their state.

Power prediction

Inputs that can be used for prediction

The inputs are the informations given by the user about his needs :

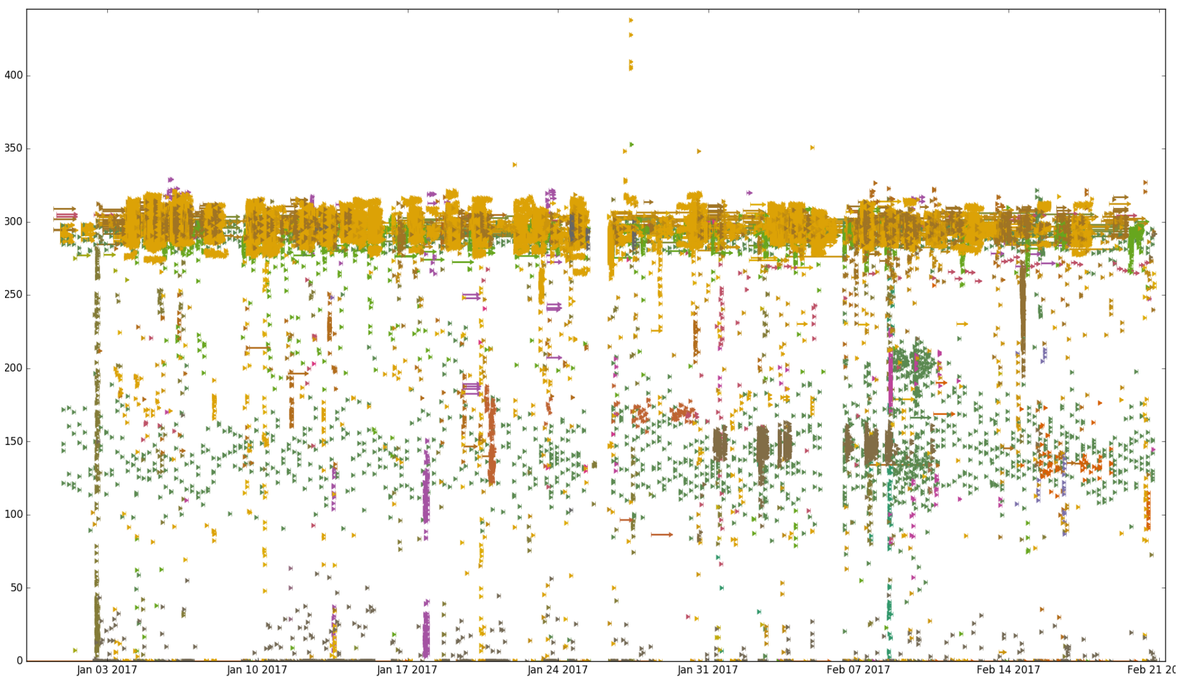
- UID (User ID), GID (Group ID of the user)
- submit date
- number of nodes (asked, allocated) or number de CPU's (asked, allocated)
- timelimit (duration before killing the job), QoS (Quality of Service, test or production job)...

Outputs, collected after computation

- time elapsed (duration of computation)
- consumed energy (in Joule)

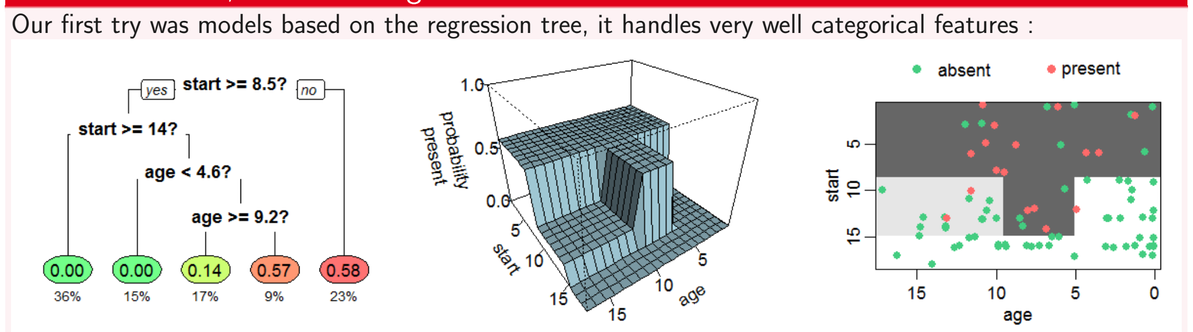
Predict $\mathrm{Power/nodes} = \frac{\mathrm{consumedenergy}}{\mathrm{nodes} \times \mathrm{elapsed}}$ as a function of inputs

Notice : one core consumes about 10 Watts when active, so 300 Watts/nodes.
COBALT's data : the color is given by the GID, the vertical axis is the power by node



Prediction model, Rote Learning

Our first try was models based on the regression tree, it handles very well categorical features :

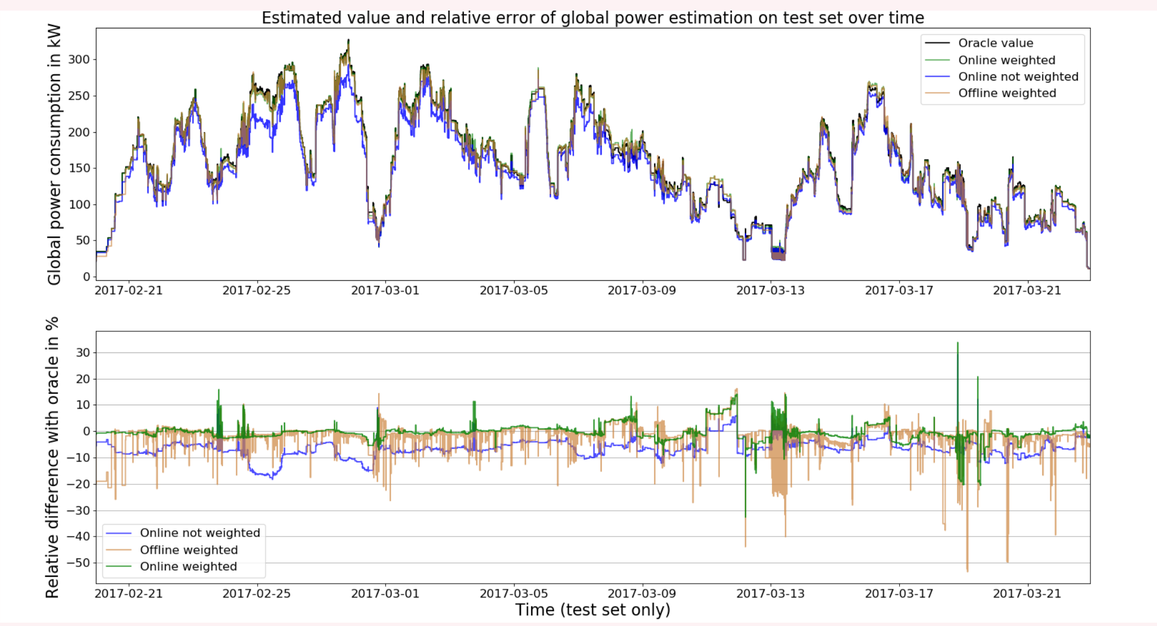


Because this is a too simple model, we use also bagging and boosting which gives us respectively gradient boosting regression and random forest.
After that we decide to implement a very simple model easy to interprete (1) :
For the job i, let us note (X_i, Y_i) the input data and the power by nodes respectively. We compute the classical estimators of statistics on the empirical probability distribution conditionned on the possible values of the inputs :

$$\mathbb{E}[Y_i | X = X_i] \quad \text{or} \quad q^\alpha \text{ such that } \mathbb{P}[Y_i < q^\alpha | X = X_i] = \alpha$$

Reweighting and online computations for global power consumption monitoring

The GID and the number of tasks per nodes are the most characteristic inputs for estimation. We compute the estimation online and remove the bias favoring too much short jobs for global power prediction using the formula of the Exponential Weighted Moving Average for irregular time-series (1).



In general, the informations about the user are almost sufficient to get a good prediction. The jobs on COBALT are so recurrent that this simple model is good enough.

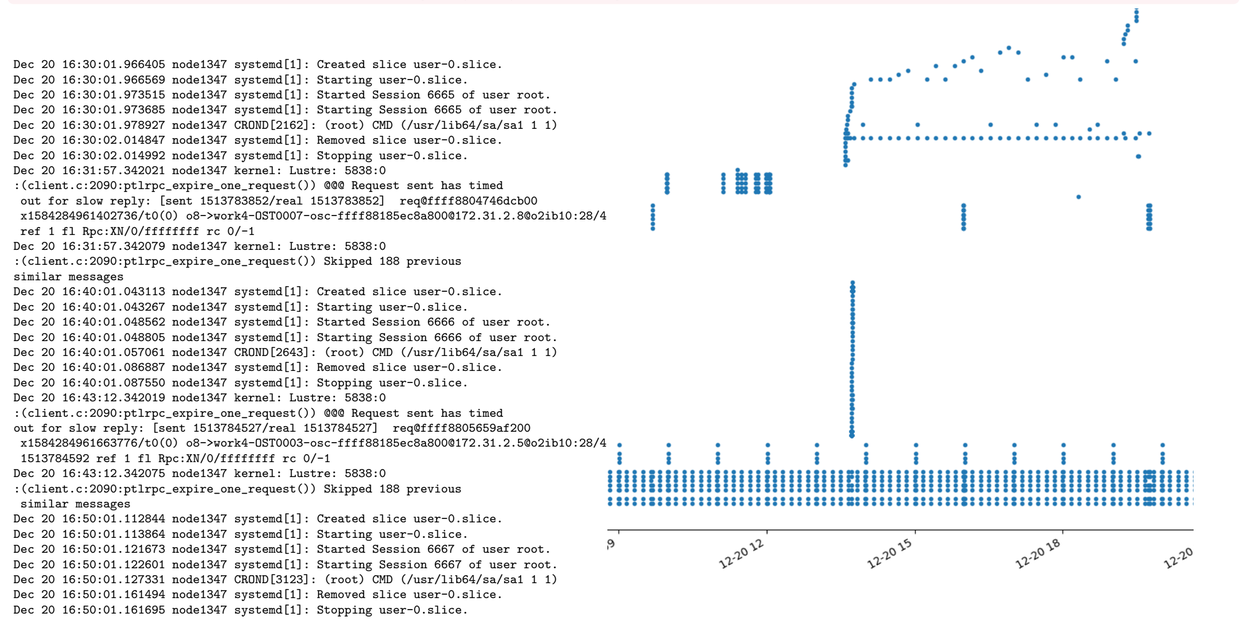
[1] Saillant, T., Weill, J.C., Mougeot, M.: Predicting job power consumption based on rjms submission data in hpc systems. In: International Conference on High Performance Computing. pp. 63–82. Springer (2020)

Textual logs analysis

What is "log" ?

Logs : event dated as a slightly formatted string, displayed by a console program and saved in order to ensure its proper execution.

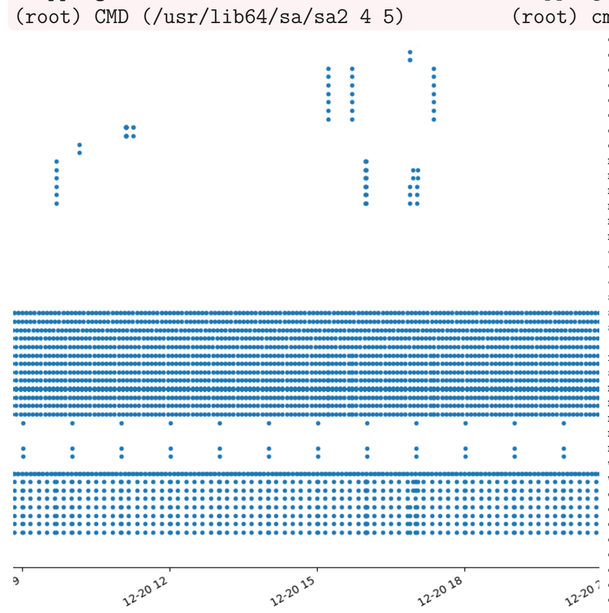
- Difficulty reconciling textual and temporal informations
- "bursty behavior" : the intensity of event is spiked
- message not very structured with more or less formal formats (syslog)
- important messages are hidden in redondancy (+90% inutiles)
- the logs are spread in many files (cf \var\log folder)



Logs structure and templates

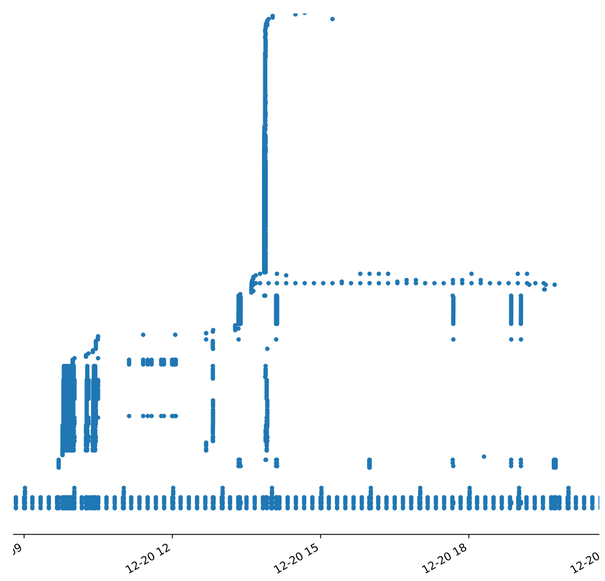
We need to gather the logs and one way to think that two logs are equal is to infer if they are emitted by the same line in source code. So they should share a template. To extract those templates, anonymizing the logs is a first step. Each template is then hashed to a number so that it is uniquely defined.
This gives use the steps bellow : brut data, anonymized and finally hashed

`Created slice user-0.slice.`	`created slice user-_#NMBZ#_.slice.`	`#000`
`Started Session 6666 of user root.`	`started session _#NMBY#_ of _#USER#_.`	`#001`
`(root) CMD (/usr/lib64/sa/sa1 1 1)`	`(root) cmd (_#PATH#_ _#HADY#_)`	`#002`
`Removed slice user-0.slice.`	`removed slice user-_#NMBZ#_.slice.`	`#003`
`Stopping user-0.slice.`	`stopping user-_#NMBZ#_.slice.`	`#004`
`(root) CMD (/usr/lib64/sa/sa2 4 5)`	`(root) cmd (_#PATH#_ _#HADY#_)`	`#002`




```
error: specifying on_fail=fence and stonith-enabled=false makes no sense
error: specifying on_fail=fence and stonith-enabled=false makes no sense
error: specifying on_fail=fence and stonith-enabled=false makes no sense
error: specifying on_fail=fence and stonith-enabled=false makes no sense
error: specifying on_fail=fence and stonith-enabled=false makes no sense
error: specifying on_fail=fence and stonith-enabled=false makes no sense
error: specifying on_fail=fence and stonith-enabled=false makes no sense
error: specifying on_fail=fence and stonith-enabled=false makes no sensenoti_#HWAD#_ calculated
noti_#HWAD#_ configuration errors found during pe processing.  please run "crm_verify -l" to id
noti_#HWAD#_ transition _#NMBY#_ (complete=_#NMBY#_, pending=_#NMBY#_, fired=_#NMBY#_, skipped=
noti_#HWAD#_ state transition s_transition_engine -> s_idle
nrpe : tty=unknown ; pwd=_#PATH#_ ; user=root ; command=_#PATH#_ vg1a vg1b vg2a vg2b
nrpe : tty=unknown ; pwd=_#PATH#_ ; user=root ; command=_#PATH#_
nrpe : tty=unknown ; pwd=_#PATH#_ ; user=root ; command=_#PATH#_ pacemaker
configuration file _#PATH#_ is marked executable. please remove executable permission bits. pro
configuration file _#PATH#_ is marked executable. please remove executable permission bits. pro
created slice user-_#NMBZ#_.slice.
starting user-_#NMBZ#_.slice.
started session _#NMBY#_ of _#USER#_.
starting session _#NMBY#_ of _#USER#_.
(root) cmd (_#PATH#_ _#HADY#_)
removed slice user-_#NMBZ#_.slice.
stopping user-_#NMBZ#_.slice.
nrpe : tty=unknown ; pwd=_#PATH#_ ; user=root ; command=_#PATH#_
nrpe : tty=unknown ; pwd=_#PATH#_ ; user=root ; command=_#PATH#_ -u-w
nrpe : tty=unknown ; pwd=_#PATH#_ ; user=root ; command=_#PATH#_ _#NMBY#_ -r _#NMBY#_
noti_#HWAD#_ state transition s_idle -> s_policy_engine
noti_#HWAD#_ on loss of ccm quorum: ignore
warning: blind faith: not fencing unseen nodes
warning: blind faith: not fencing unseen nodes
error: specifying on_fail=fence and stonith-enabled=false makes no sense
error: specifying on_fail=fence and stonith-enabled=false makes no sense
error: specifying on_fail=fence and stonith-enabled=false makes no sense
error: specifying on_fail=fence and stonith-enabled=false makes no sense
error: specifying on_fail=fence and stonith-enabled=false makes no sense
error: specifying on_fail=fence and stonith-enabled=false makes no sense
error: specifying on_fail=fence and stonith-enabled=false makes no sense
error: specifying on_fail=fence and stonith-enabled=false makes no sense
error: specifying on_fail=fence and stonith-enabled=false makes no sense
error: specifying on_fail=fence and stonith-enabled=false makes no sense
error: specifying on_fail=fence and stonith-enabled=false makes no sense
error: specifying on_fail=fence and stonith-enabled=false makes no sense
error: specifying on_fail=fence and stonith-enabled=false makes no sense
error: specifying on_fail=fence and stonith-enabled=false makes no sense
error: specifying on_fail=fence and stonith-enabled=false makes no sense
error: specifying on_fail=fence and stonith-enabled=false makes no sense
error: specifying on_fail=fence and stonith-enabled=false makes no sense
error: specifying on_fail=fence and stonith-enabled=false makes no sense
error: specifying on_fail=fence and stonith-enabled=false makes no sense
error: specifying on_fail=fence and stonith-enabled=false makes no sense
error: specifying on_fail=fence and stonith-enabled=false makes no sense
error: specifying on_fail=fence and stonith-enabled=false makes no sense
error: specifying on_fail=fence and stonith-enabled=false makes no sense
error: specifying on_fail=fence and stonith-enabled=false makes no sense
error: specifying on_fail=fence and stonith-enabled=false makes no sense
error: specifying on_fail=fence and stonith-enabled=false makes no sense
error: specifying on_fail=fence and stonith-enabled=false makes no sense
error: specifying on_fail=fence and stonith-enabled=false makes no sense
error: specifying on_fail=fence and stonith-enabled=false makes no sense
error: specifying on_fail=fence and stonith-enabled=false makes no sense
error: specifying on_fail=fence and stonith-enabled=false makes no sense
error: specifying on_fail=fence and stonith-enabled=false makes no sense
error: specifying on_fail=fence and stonith-enabled=false makes no sense
error: specifying on_fail=fence and stonith-enabled=false makes no sense
error: specifying on_fail=fence and stonith-enabled=false makes no sense
error: specifying on_fail=fence and stonith-enabled=false makes no sense
error: specifying on_fail=fence and stonith-enabled=false makes no sense
error: specifying on_fail=fence and stonith-enabled=false makes no sense
error: specifying on_fail=fence and stonith-enabled=false makes no sense

ridft invoked oom-killer: gfp_mask=_#HEXX#_, order=_#NMBY#_, oom_score_adj=_#NMBY#_
ridft cpuset=_#PATH#_ mems_allowed=_#HADY#_
cpu: _#NMBY#_ pid: _#NMBY#_ comm: ridft taint_#HWAD#_ g oe ------------ _#VERX#_ _#CNTR#_
hardware name: bull bullx blade/che-u , bios biosx_#IPVF#_ _#DATY#_
call tr_#HWAD#_
[<_#HEXY#_>] dump_stack+_#HEXX#_/_#HEXX#_
[<_#HEXY#_>] dump_header+_#HEXX#_/_#HEXX#_
[<_#HEXY#_>] ? cred_has_capability+_#HEXX#_/_#HEXX#_
...
[<_#HEXY#_>] do_page_fault+_#HEXX#_/_#HEXX#_
[<_#HEXY#_>] page_fault+_#HEXX#_/_#HEXX#_
mem-info:
active_anon:_#NMBY#_ inactive_anon:_#NMBY#_ isolated_anon:_#NMBY#_
node _#NMBY#_ dma fr_#HWAD#_kb min:_#BITS#_ low:_#BITS#_ high:_#BITS#_ active_anon:_#BITS#_ ina
lowmem_reserve[]: _#HADY#_ _#HADY#_
node _#NMBY#_ dma: _#NMBZ#_*_#BITS#_ (u) _#NMBZ#_*_#BITS#_ _#NMBZ#_*_#BITS#_ _#NMBZ#_*_#BITS#_
node _#NMBY#_ dm_#HWAD#_ _#NMBZ#_*_#BITS#_ (uem) _#NMBZ#_*_#BITS#_ (uem) _#NMBZ#_*_#BITS#_ (uem
node _#NMBY#_ hugepages_total=_#NMBY#_ hugepages_free=_#NMBY#_ hugepages_surp=_#NMBY#_ hugepage
...
node _#NMBY#_ hugepages_total=_#NMBY#_ hugepages_free=_#NMBY#_ hugepages_surp=_#NMBY#_ hugepage
_#NMBZ#_ total pagecache pages
_#NMBZ#_ pages in swap cache
swap cache stats: _#NMBY#_ _#NMBY#_, delete _#NMBY#_, find _#NMBZ#_/_#NMBZ#_
free swap = _#BITS#_
total swap = _#BITS#_
_#NMBZ#_ pages ram
_#NMBZ#_ pages highmem/movableonly
_#NMBZ#_ pages reserved
[ pid ] uid tgid total_vm rss nr_ptes swapents oom_score_adj name
[ _#NMBY#_] _#HADY#_ _#HADY#_ _#HADY#_ _#NMBY#_ systemd-journal
...
[ _#NMBY#_] _#HADY#_ _#HADY#_ _#HADY#_ _#NMBY#_ smartd
[_#NMBY#_] _#HADY#_ _#HADY#_ _#HADY#_ _#NMBY#_ sleep
out of memory: kill process _#NMBY#_ (ssh) score _#NMBY#_ or sacrifice child
killed process _#NMBY#_ (ssh) total-vm:_#BITS#_, anon-rss:_#BITS#_, file-rss:_#BITS#_, shmem-rs
ridft invoked oom-killer: gfp_mask=_#HEXX#_, order=_#NMBY#_, oom_score_adj=_#NMBY#_
ridft cpuset=_#PATH#_ mems_allowed=_#HADY#_
cpu: _#NMBY#_ pid: _#NMBY#_ comm: ridft taint_#HWAD#_ g oe ------------ _#VERX#_ _#CNTR#_
hardware name: bull bullx blade/che-u , bios biosx_#IPVF#_ _#DATY#_
_#HADY#_ _#HADY#_
_#HADY#_ _#HADY#_
_#HADY#_ _#HAD[width=\textwidth, trim={20cm, 0cm, 10cm, 0cm}Y#_
call tr_#HWAD#_
[<_#HEXY#_>] dump_stack+_#HEXX#_/_#HEXX#_
[<_#HEXY#_>] dump_header+_#HEXX#_/_#HEXX#_
[<_#HEXY#_>] ? cred_has_capability+_#HEXX#_/_#HEXX#_
```



Outlooks

We will focus on the events marked as shown on the graphs. This representation is the starting point for methods to create alerts or extract the underlying structure in order to detect anomalies or to perform predictive maintenance.