A case for EL8+

grigoras@cern.ch

A new experiment software stack

… with a new set of requirements

- One workflow instance (job) launching 80+ concurrent processes
- With one or more threads each
- 160+ shared libraries loaded by each *o2* process in the workflow

These jobs are thus multicore and require multiple cores (and the memory)

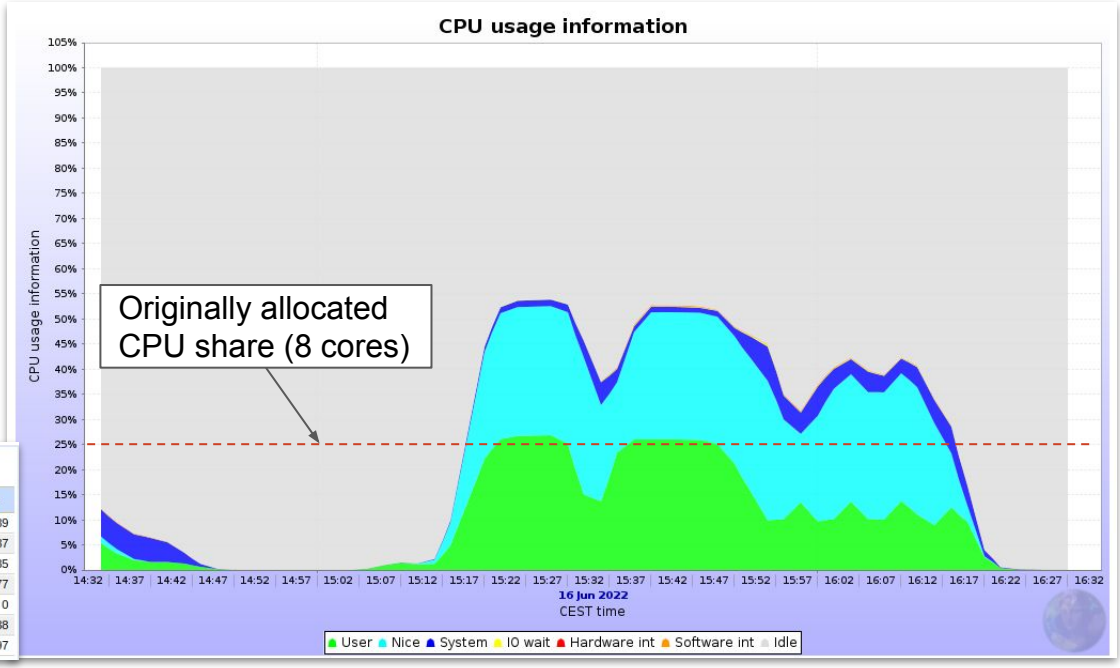
- we move to 8 core (or whole node) queues everywhere

They make efficient use of the 8*2GB memory that comes with them

and of the CPUs as well…

CPU over-usage by payloads

CPU usage of a 32-core idle machine running a simulation payload.



CPU usage information

	Series	Last value	Min	Avg	Max
1.	User	0.033	0.018	9.879	26.89
2.	Nice	0	0	11.86	33.87
3.	System	0.021	0.007	1.997	11.35
4.	IO wait	0.001	0.001	0.062	0.477
5.	Hardware int	0	0	0	0
6.	Software int	0.002	0.001	0.105	0.38
7.	Idle	99.94	45.89	76.09	99.97

Average job CPU efficiency →**160.25%**
(normalized to the 8 core allocation)

cgroups v2

Fine-grained resource control per job

- Num. CPUs per container
- CPU time utilisation / constraints
- Exposed memory / limits
- Available swap
- I/O limits

Until then we can only use *taskset* to control how much CPU they use

Due to the more efficient execution when CPU cores are correctly selected

- Mix `cgroups` v2 and `taskset` for a complete isolation of jobs

cgroups v2

Only available from EL8 onwards

Essentially we only need the host kernel to be new enough to support it

Then we run our payloads in singularity images unpacked in CVMFS:
`/cvmfs/alice.cern.ch/containers/fs/singularity/{centos7, rel8, rel9}`

So payloads see the expected `lsb_release` or other platform packages

GPUs are supported, if running on an EL8+ host

Container image configurable per CE

Monitoring enhancements

Payloads are continuously monitored / supervised while running (1 check @ 5s)

One of the parameters is the PSS, by which we also terminate jobs

as `sum(/proc/<PID>/smaps)`

The many *processes * libraries * segments* delay the evaluation, due to the very large size of the above proc entries (O(s) per process)

EL8+ kernels also offer `/proc/<PID>/smaps_rollup`

summary information, instantly produced

enough for our purposes

EL8+ ?

The requirements from the hosts are very light:

- recent kernel
- `cgroups` v2 enabled
- ability to start containers from CVMFS

Then we control the container version and the resources given to each payload

What are your plans to upgrade?