ATLAS SCALE-UP TEST ON PIZ DAINT

Gianfranco Sciacca

AEC - Laboratory for High Energy Physics, University of Bern, Switzerland

LHConCray WG - 7 November 2017

TEST SETUP

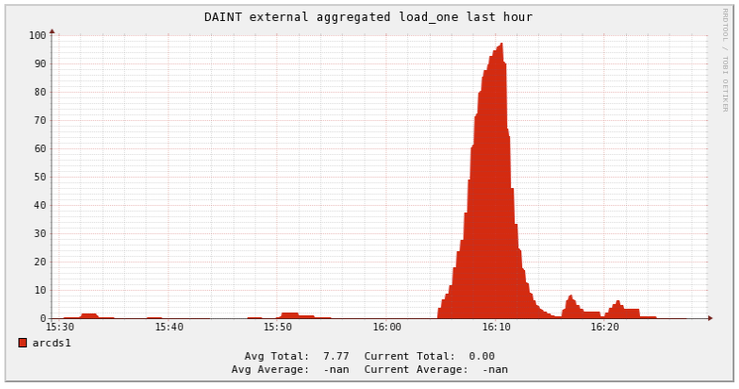
- **ARC setup:** 1 ARC CE + 1data stager (both doing staging) - `maxdelivery="100"`
  - No ARC caching
  - 2 LRMS queues: wlcg (production queue), atltest (added for this test)

- **Preliminary setup:** SLURM reservation with 11 Piz Daint nodes: 72 HT slots, 64GB RAM
  - originally decided to use 64 out of the 72 slots
  - 16-core jobs: 44 jobs to fill the system (704 slots)
- **Scale-up setup:** SLURM reservation with 384 Piz Daint nodes: 72 HT slots, 64GB RAM
  - decided to use all of the 72 slots
  - 18-core jobs: 1536 jobs to fill the system (27648 slots)

- **Job setup:** Validation task: https://bigpanda.cern.ch/task/12491843/
  - 4M events, **40 k jobs, 40k input files**, up to 148MB/file (mostly 115MB)
  - jobs tuned to ~1h duration (maxEvents=100)
  - ramCount=900 MBPerCore
  - Output expected: ~70MB/job

PREPARATION

Started 02 Nov 4 PM



- **Started submitting jobs, 2 Nov at 4PM**
  - **Load spike on the data stager, breaks GPFS**
  - **set `maxdelivery="30"`**
  - **we also had:**

```
# 300 at the end means that it wont cancel/submit more than 300 jobs at the same time
maxjobs="40000 20000 8000 80000 300"
```

  - **Jobs started running**
  - **Settled eventually on:**

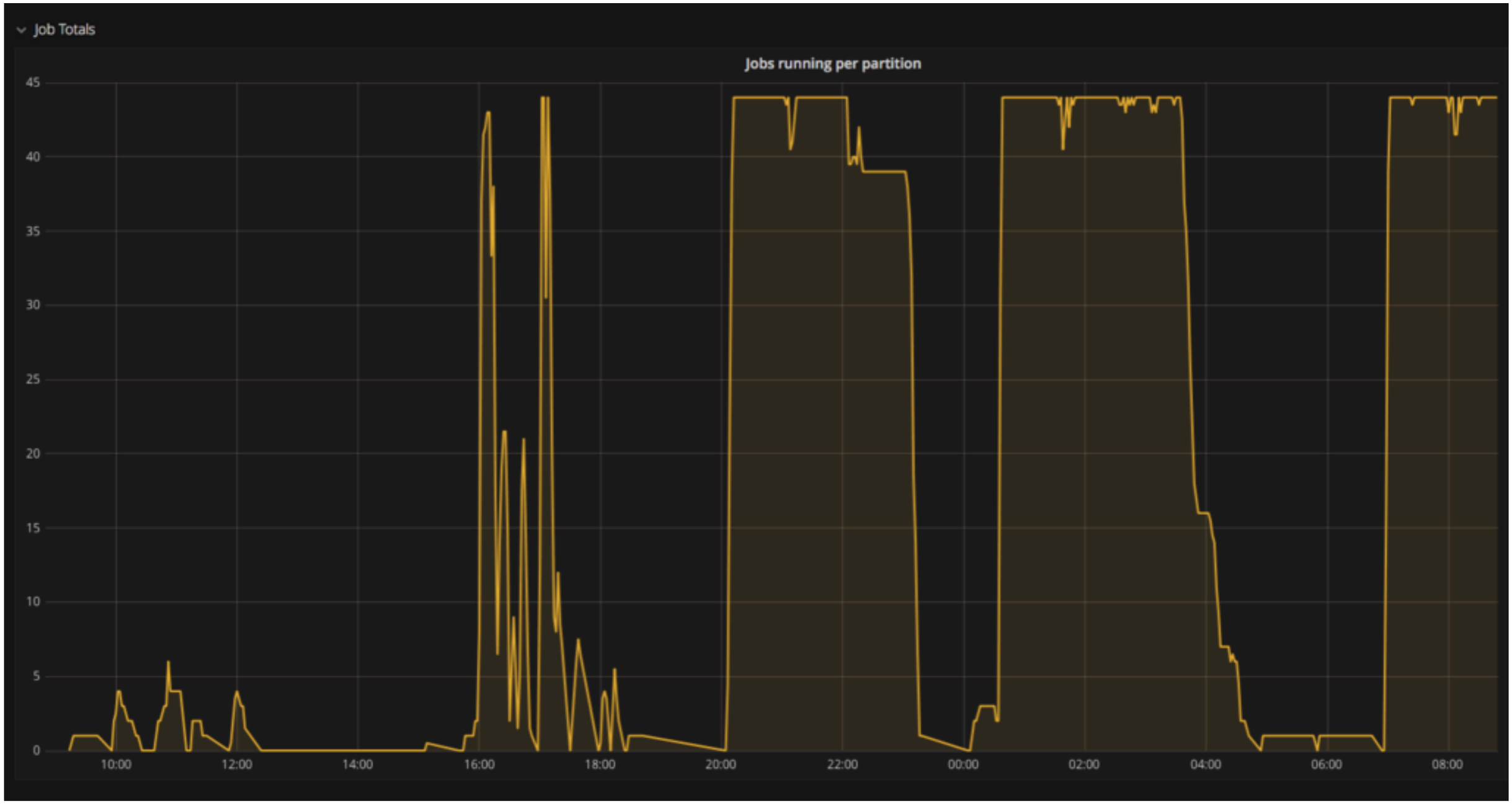
```
[grid-manager]
maxjobs="40000 20000 8000 80000 800"
[data-staging]
maxdelivery="30"
```

- **The ARC CE reports 0 running, only for the "atltest" partition. The "wlcg" partition seems to be reported correctly**
- **This prevented the aCT from submitting continuously**

```
[root@arc04 arc]# tail /var/spool/nordugrid/jobstatus/job.helper.errors
/usr/share/arc/scan-SLURM-job: line 226: [: -ne: unary operator expected
/usr/share/arc/scan-SLURM-job: line 228: [: ExitCode: integer expression expected
date: invalid date 'Start'
date: invalid date ‘End'
/usr/share/arc/scan-SLURM-job: line 287: - : syntax error: operand expected (error token is "- ")
/usr/share/arc/scan-SLURM-job: line 226: [: -ne: unary operator expected
/usr/share/arc/scan-SLURM-job: line 228: [: ExitCode: integer expression expected
date: invalid date 'Start'
date: invalid date 'End'
/usr/share/arc/scan-SLURM-job: line 287: - : syntax error: operand expected (error token is "- ")
```

PREPARATION

Job pattern with bad infosys



PREPARATION

Bad output from scan-SLURM-job

- **What is the issue?**
  - **At times `sacct` does not return anything, but `scontrol` does for a specific jobid. In such cases the script seems to die miserably**
  - **ARC seems capable of producing the correct value of `nordugrid-cluster-usedcpus` for one queue only**
  - **It seems to query SLURM for the first queue that is defined?**
- **We decided to move to a dedicated ARC CE (a 10GbE VM now, no staging) and do all the staging over the data stager only**
  - **Jobs started flow from aCT and run in stable condition**
  - **Unfortunately, we did NOT realise this one was running `nordugrid-arc-arex-5.3.0-1.el7.centos.x86_64`**
  - **Only realised it after the scale-up run had started**
  - **We considered upgrading on the fly vs. babysit**
  - **Decided it was too risky to upgrade (the admin was not comfortable doing that)**

SCALING UP

Started 06 Nov 8 AM

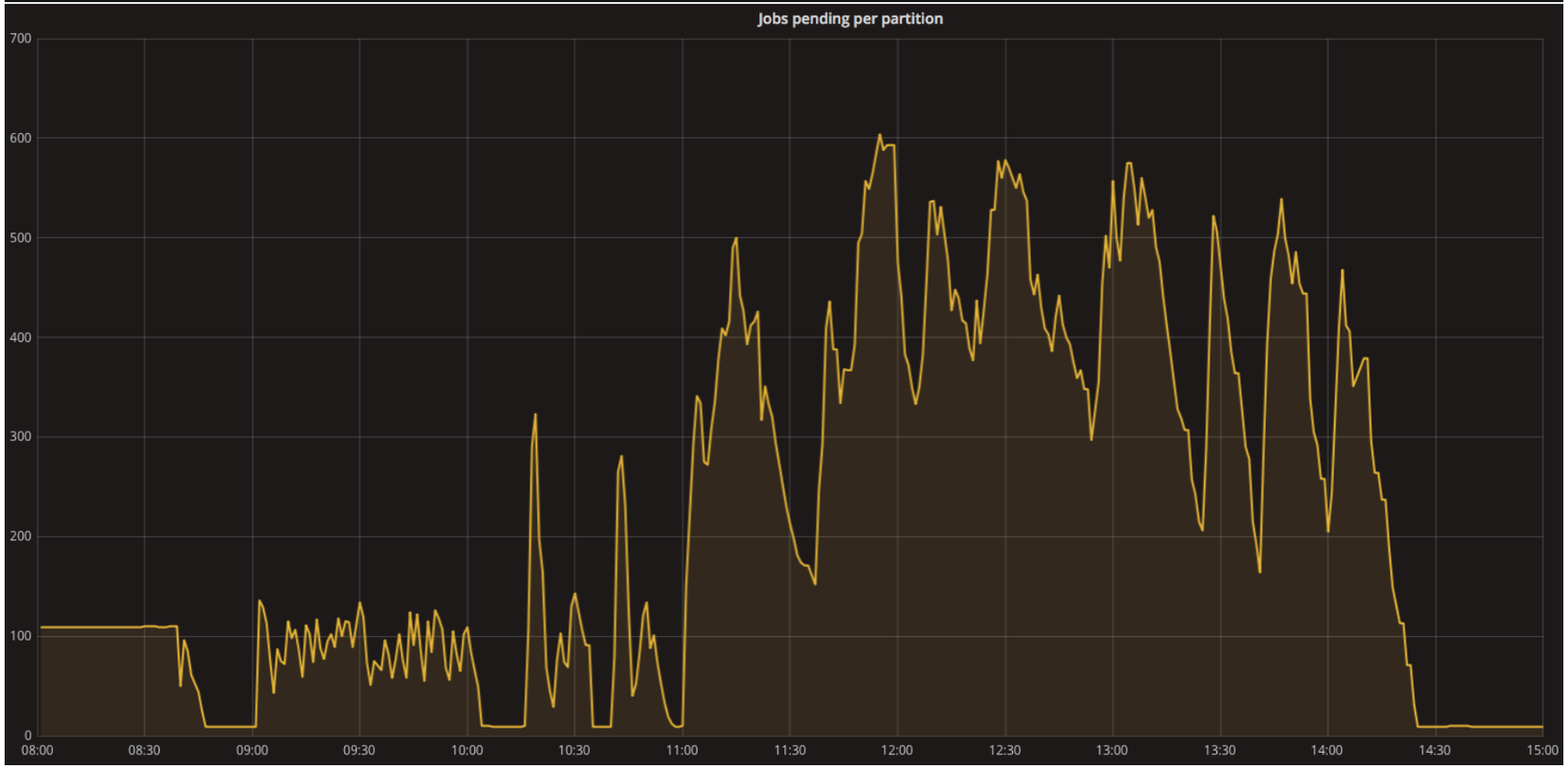
- **Reached 1420 jobs (25560 cores) in ~1h**
  - **a-rex died straight away, needed restarting by hand**
    `[2017-11-06 08:53:13] [Arc.Daemon] [ERROR] [78862/28075008] Watchdog detected application timeout - killing process`
  - `[2017-11-06 08:53:13] [Arc.A-REX] [INFO] [78864/28075008] Shutting down job processing`
  - `[2017-11-06 08:53:13] [Arc.A-REX] [INFO] [78864/28075008] Shutting down data staging threads`
  - **fairly linear otherwise, 27 jobs/min (486 slots)**
  - **seemingly dominated by SLURM, not aCT/ARC or GPFS**
  - **gazillion of messages like**

    `(arched:61671): GLib-WARNING **: GChildWatchSource: Exit status of a child process was requested but ECHILD was received by waitpid(). Most likely the process is ignoring SIGCHLD, or some other thread is invoking waitpid() with a nonpositive first argument; either behavior can break applications that use g_child_watch_add()/g_spawn_sync() either directly or indirectly.`

    These seen to be harmless. Then why?

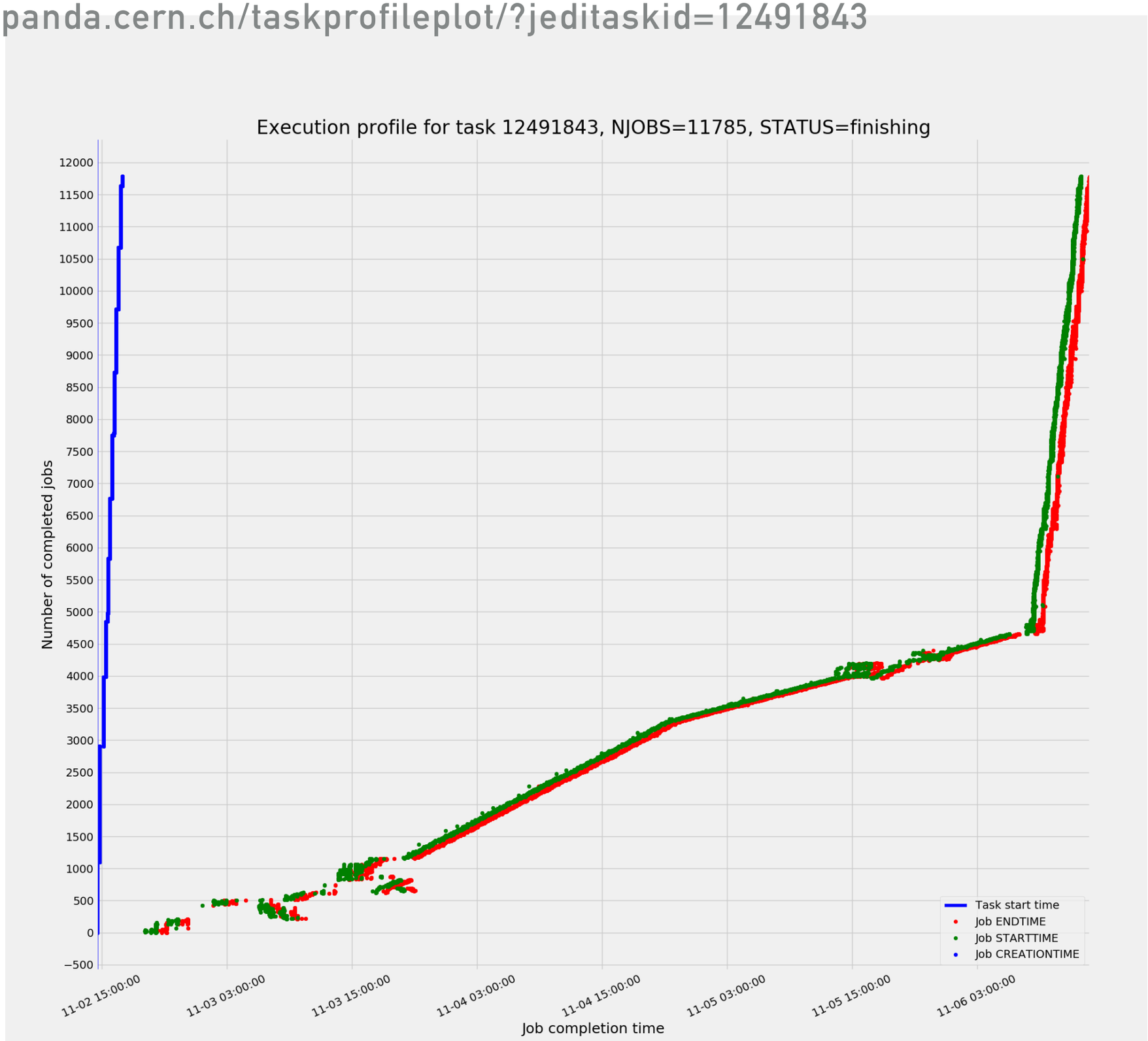
- **Increased the maxqueued on the aCT to have a large enough buffer and avoid draining between restarts**
- **Stable running for 3h from 11 AM**
- **We disabled the watchdog, still some/several manual a-rex restarts needed**
- **Stopped submission at 2 PM**
- **Killed all running from the aCT at 2:45 PM**
- **System clean at 3 PM**

TEST SUMMARY



TEST SUMMARY

https://bigpanda.cern.ch/taskprofileplot/?jeditaskid=12491843



TEST SUMMARY

http://dashb-atlas-job.cern.ch/dashboard/request.py/dailysummary#button=resourceutil&sites%5B%5D=CSCS-LCG2&sitesCat%5B%5D=CH-CHIPP-CSCS&resourcetype=All&sitesSort=2&sitesCatSort=2&start=null&end=null&timerange=last48&granularity=Hourly&generic=0&sortby=16&series=30&activities%5B%5D=all

- **1M events processed (25% of total): 10162 jobs (out of 11785)**
- **Total input size: 1TB (no ARC caching), output size: 0.7TB (staged to a SE in Spain)**
- **Max running jobs reached 1432 (25774/27648 cores - 93.22% , some nodes were down)**
- **Unfortunately we could not test the latest stable ARC version**
- **My feel is that ARC can easily become a bottleneck (if unstable, etc...)**