Considerations on CPU and Memory limits in the WN

Alessandra Forti

ATLAS Jamboree

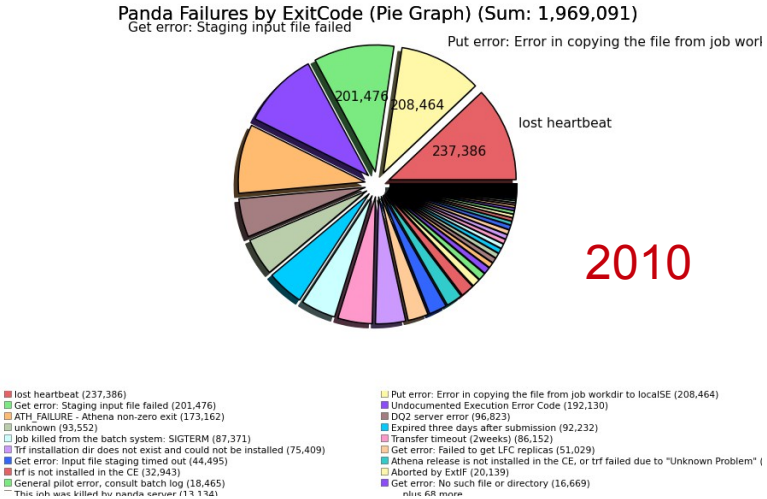
CERN

10th December 2012

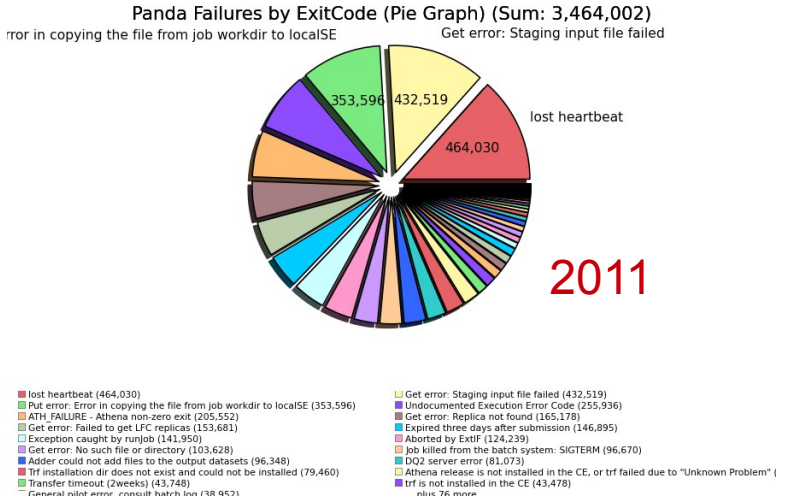
This talk

- Lost heartbeat: follow up of talks given in July
  - Review of CPU hours and walltime (Alessandra)
  - CPU and Wall time follow-up (Rod)
- Memory: how to limit memory leaks without killing all the jobs
- A proposal for a new dashboard

Lost Heartbeat: Quantified



2010

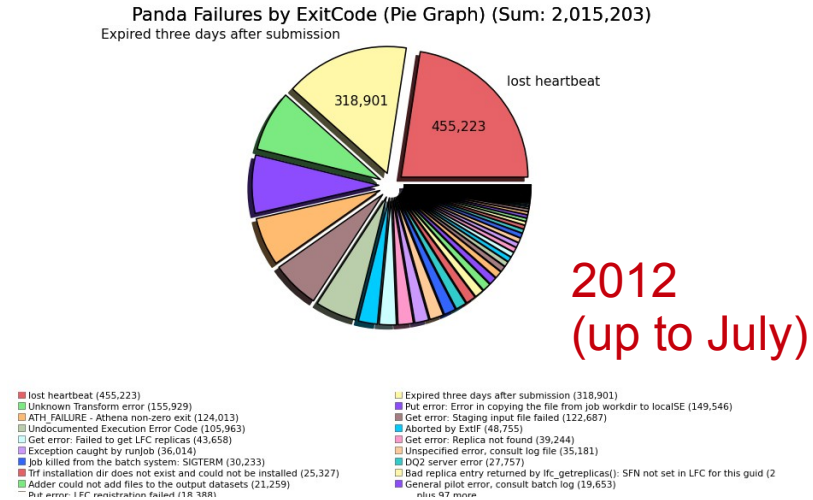


2011

- All Atlas T2 errors per year.

  Lost heartbeat is cumulatively the most dominant error in atlas production at T2s.

  It is important also in analysis but in the monitoring it's squashed by more common Athena problems.



2012
(up to July)

Lost Heartbeat: what is it?

- Lost heartbeat happens when the jobs loose connection with the panda server.

  Causes can be many: WN crashes, CE crashes, power cuts, network problems, panda crashes, WMSglidein problems....

  But the most common is: jobs exceed batch system limits on resources. In particular CPU or Wall time.

  - It's not because sites don't give enough resources
  - VOid card: 3120 mins (52h) on a 1 kSI2k machine

    ~2.2 kSI2k should require only 23.6h

    Mcr with 48h queue still failed

    Other sites have even longer queues even up to 96h

Superficial dig on tasks

- There are some tasks that have bugs and they simply loop until the batch system kills them.

  https://savannah.cern.ch/support/?129431

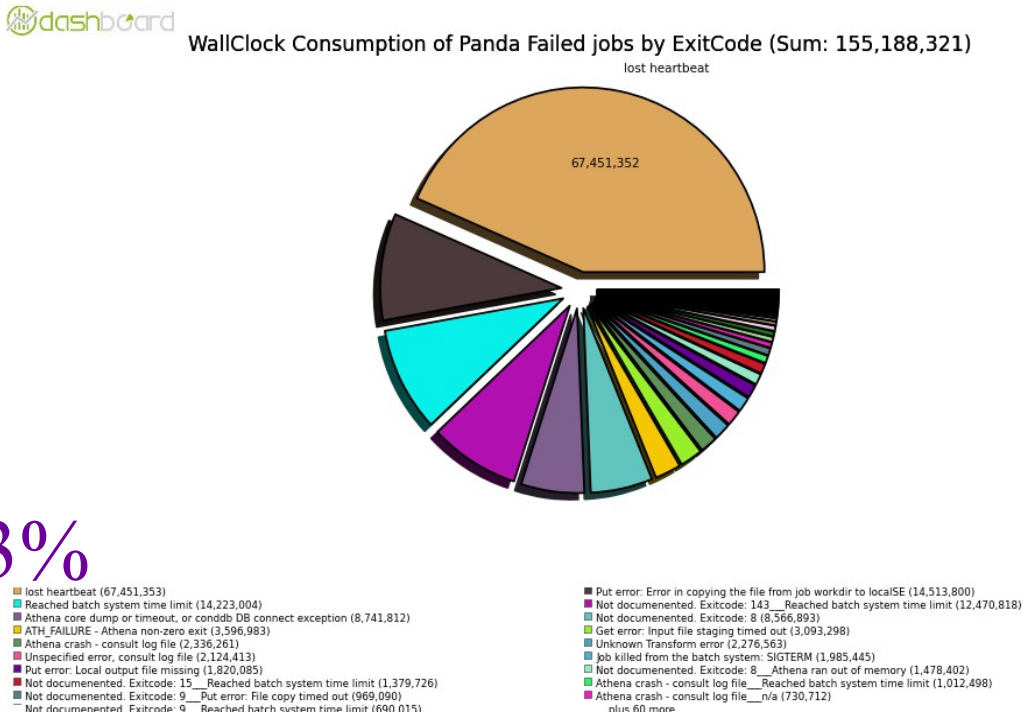
- Some other tasks seem just to require a very long time and are not brokered correctly.

Wasted resources

- Problem is not the number of jobs is the amount of CPU/Wallclock hours wasted

  8,535*48=409,680 hours or 17,070 days

- Prototype dashboard now reports CPU time/Wall time per error.



- Just as indication

  UK last week

  Lost heartbeat 43%

GridPP
UK Computing for Particle Physics

Two problems

- Why these jobs do not terminated with a different error?

  Pilot problem

- Why jobs that require long time aren't scheduled on longer queues?

  Brokering problem

Pilot Problem

- Most batch systems send a SIGTERM before a SIGKILL. Pilot doesn't seem to catch this consistently

  Sometimes it does and we get “Job killed by signal 15: Signal handler has set job result to FAILED, ec = 1201” or “Reached maximum batch system time limit “

  Sometimes it doesn't and we get “lost heartbeat”

- Catching the signal consistently would help

  Report a correct error code rather than lost heartbeat

  - And identify which jobs are using more than they should

  Terminate the job cleanly

  - This requires to configure a time delay between at SIGTERM and SIGKILL and the pilot using it.

- Fixing this doesn't avoid the waste of resources from jobs that really take long time.

Brokering Problem

- To avoid wasting resources we need to better broker the jobs.
- A new algorithm since last August

  Use scout jobs measures to predict how much time they require and use maxtime parameter in schedconfig to broker the jobs.

  This helped reducing dramatically the number of "lost heartbeat" due to exhausted CPU time/Wall time.

- Didn't work at all sites

  RAL was complaining still few weeks ago

  - Looping jobs might be the culprit though

maxtime

- There were setting maxtime automatically for all queues
  - Schedconfig/bdii inconsistencies
  - Obsolete queues
  - It looks fixed now
- The algorithm is an approximation it doesn't guarantee all jobs will be schedule correctly
  - From this week a higher cpucount is assigned to jobs that fail
    - Automatically set at 96h if jobs fail with “Reached maximum batch system time limit”.

Memory errors

- Another subset of “Lost heartbeat”
- Requested 2GB pmem 4GB vmem

  Most sites at Atlas request do not set limits in batch system

- Multicore machines have enough memory to allow jobs exceed the requirement without problem

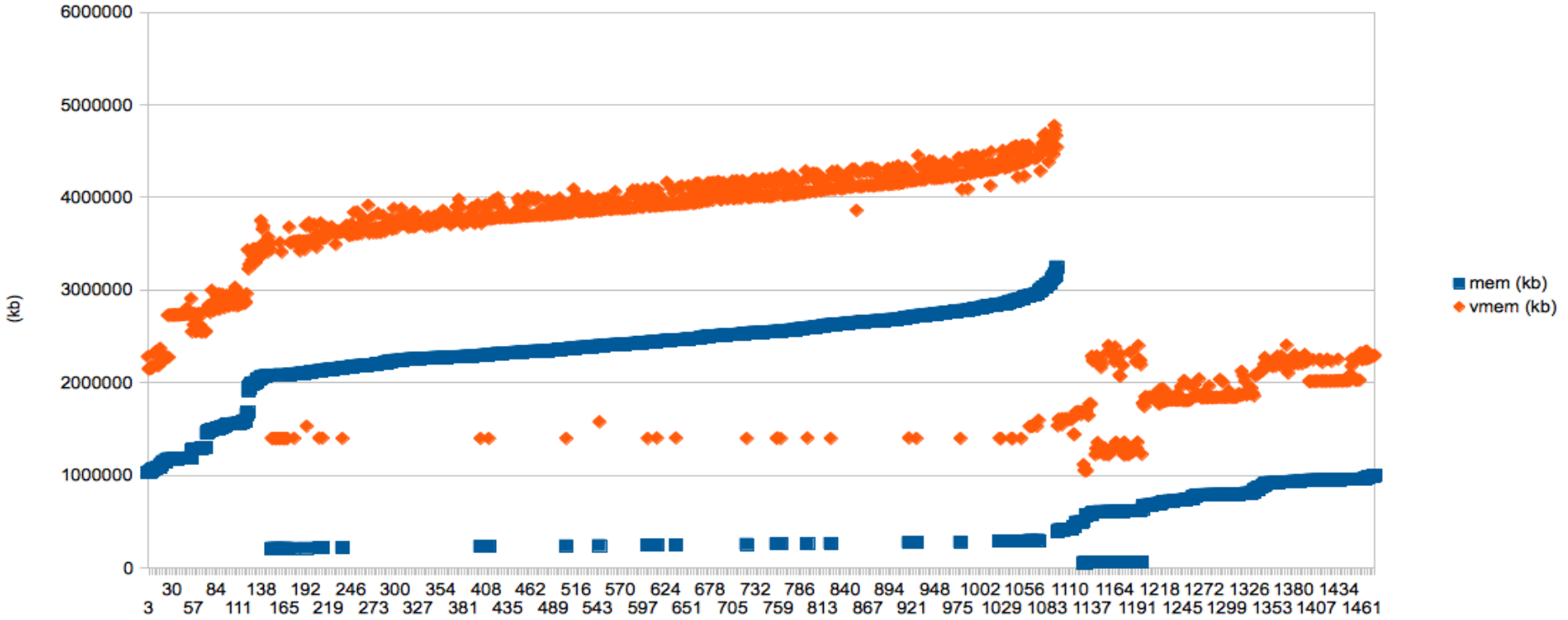
  Still the physical memory is not inexhaustible

  No way to contain memory leaks

  WNs crashing without much reason

  - Other (VOs) jobs affected
  - Wasted resources

Problem



- Want to limit memory leaks without affecting most jobs

A way to limit

- Batch System
  - Recipes are batch system dependent
  - To limit in the batch system without killing most jobs you need to over allocate resources.
  - Torque/Maui http://tinyurl.com/d692cro
  - Simplest recipes affects all the users
    - Often not as demanding but...
- SchedConfig/AGIS queue config
  - 1 memory parameter to set
  - Batch system independent
  - Uses ulimit to set vmem per process
  - Leaves a trace in the log files
    - Setting VMEM limit to 3000000kB
  - No worry about pilot/batch system communication
  - Atlas self contained
  - Haven't had a crashing node since.

New dashboard proposal

- In the past 1.5 year I've looked at a number of failure modes

  Cvmfs files corruption, cvmfs timeout, lost heartbeat, replica not found, memory problems, missing shared libraries.....

- Used the historical dashboard, panda monitor matching batch system logs

- Historical dashboard helped a lot giving aggregate information

  But it has some limitations

  - No task information
  - No links to the jobs
  - No possibility to select errors
  - Difficult to understand if errors spread in multiple sites

New dashboard proposal

- DDM2 can do all this for transfers
- Proposal to have a “DDM2” style dashboard for jobs
  - Task vs sites matrix
  - Success and errors rates
    - Errors grouped by type
  - Links back to job errors
  - Capability to plot single or multiple errors
    - Per task
    - Per site

Conclusions

- Lost heartbeat has been tamed by better brokering production jobs (requires maxtime).

  Analysis and group jobs might still cause problems

  - There is neither "user" nor group scout scheme

  Pilot catching SIGTERM **consistently** would help identify the jobs that still are not brokered correctly.

- Memory errors can be put under control by limiting the memory parameter in schedconfig on all the queues.
- A DDM2 style dashboard for job errors would drastically cut down the debugging effort.