Memory handling in the ATLAS submission system

Alessandra Forti

CHEP2016

October 2016

Layout

- Memory Problem
- Memory changes in ATLAS and elsewhere
- Memory in the OS
- What Batch systems do
- PandaQueues setup
- From ATLAS to Kernel
- Memory monitoring
- Inside ATLAS
- Brokering
- Lost heartbeat
- Conclusions

Problem

- Batch systems work better if jobs pass parameters about the resource requirements
  - Used for internal scheduling and for limiting excessive usage
- Pilot system is a late binding system
  - Whatever the payload requirements it passes uniform requirements to the batch systems on the grid
  - Working when payloads were more uniform
    - Single core below 2GB memory
  - Now single core, multicore, himem, lowmem
    - Required a re-think

Memory evolution

- 4 major changes have affected memory handling
  - Increased size of the events
    - more memory consumed
  - 32bit → 64bit
    - Further increased the memory footprint
  - Introduction of multicore
    - it reined the total memory per core but it creates a new category of jobs with larger memory requirements
  - Redefinition of vmem in the kernel and shared mem reporting
    - vmem ≠ ram + swap becomes irrelevant as a quantity
      - Even traditional OS tools (ulimit) don't report correct values
      - Older batch systems still use old definition/tools
    - Multi core jobs shared memory not correctly reported by traditional OS tools

Memory according to the OS

- Memory definition is changing
  - Vmem: memory mapping in 64bit can be several times the actual memory used.
  - Smaps RSS: physical memory used by a job double counting the memory shared with other jobs
    - Different from cgroups RSS
  - **Smaps PSS: physical memory used by a job without double counting**
  - **cgroups RSS: physical memory used by the jobs without double counting**
    - Related to smaps PSS

What batch systems do?

- Batch systems **without** cgroups
  - See the same RSS as reported in smaps
  - Kill on vmem which is **NOT** a physical memory measure
    - If you insist on this you need to set it at least 3 times the RAM requested by the job
- Sites with cgroups
  - Can setup soft and hard limits on the values the job reports
  - Soft limit allows the kernel to decide if the job can keep on using the extra RAM or has to swap
  - Hard limit will kill the job based on RAM
    - Often set to 2 or 3 times the RAM requested by the job

PandaQueues Setup

- The parameters the pilot passes to the batch system are taken from the site configuration of the brokering system called PandaQueues
  - Each site has a number of PandaQueues with different parameters
    - Once 1 queue for analysis and 1 queue for production
      - Now more are needed
    - A new combination of params means a new queue
  - Seems a lot but eventually the matrix of possible values is not large
    - Estimates of the parameters values are not exact values

From ATLAS to kernel

Experiment

Experiments	corecount	rss	rss+swap	vmem	cputime	walltime
ATLAS old	corecount	maxmemory	maxmemory	-	maxtime*ncores	maxtime
ATLAS current	corecount	maxrss	maxrss+maxswap	-	maxtime*ncores	maxtime

Had to track the whole chain before any changes

Computing Elements

Computing Element	corecount	rss	rss+swap	vmem	cputime	walltime
CREAM-CE Glue1	JDL: CpuNumber= corecount; WholeNodes=false; SMPGranularity= corecount	GlueHostMainMemoryRAMSize	GlueHostMainMemoryVirtualSize	GlueHostMainMemoryVirtualSize(*)	GlueCEPolicyMaxCPUTime	GlueCEPolicyMaxWallClockTime
CREAM-CE Glue2	JDL: CpuNumber= corecount; WholeNodes=false; SMPGranularity= corecount	GLUE2ComputingShareMaxMainMemory	GLUE2ComputingShareMaxVirtualMemory(*)	GLUE2ComputingShareMaxVirtualMemory(*)	GLUE2ComputingShareMaxCPUTime	GLUE2ComputingShareMaxWallTime
ARC-CE	(count = corecount) (countpernode = corecount)	memory(*)	-	memory(*)	cputime	walltime
HTCondor-CE	xcount	maxMemory	N/A	N/A	N/A	maxWallTime

Batch systems

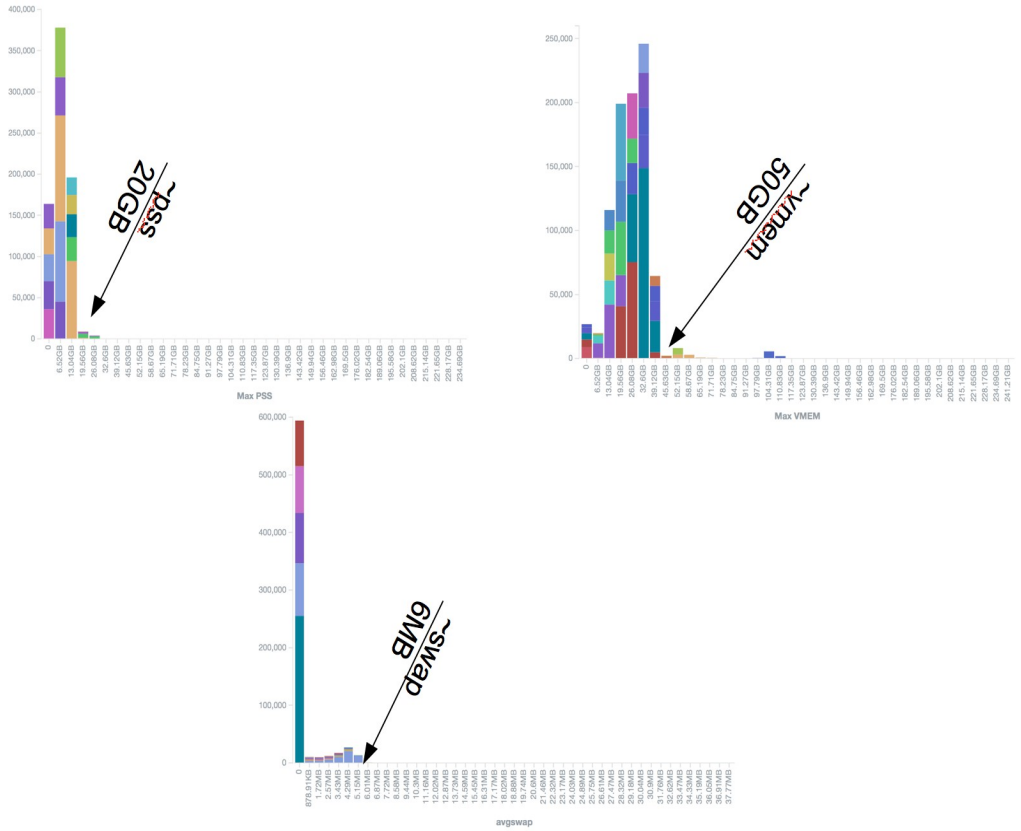
Batch system	corecount	rss	rss+swap	vmem (address space)	cputime	walltime
Torque/maui	ppn	mem	-	vmem	cput	walltime
*GE	-pe	s_rss	-	s_vmem	s_cpu	s_rt
UGE 8.2.0(*)	-pe	m_mem_free	h_vmem	s_vmem	s_cpu	s_rt
HTCondor(**)	RequestCpus	RequestMemory	No default (Recipe)	No default (Recipe)	Recipe	Recipe
SLURM	ntasks,nodes	mem-per-cpu	-	No option	No option	time
LSF	?	?	?	?	?	?

Batch systems to OS

Batch system	rss	rss+swap	vmem	needs cgroups to do sensible things
Torque/maui	-	-	RLIMIT_AS	N/A
Torque/MOAB or PBSPro >=6.0.0	yes	yes	RLIMIT_AS	yes
*GE	-	-	RLIMIT_AS	N/A
UGE >=8.2.0	yes	yes	RLIMIT_AS	yes
HTCondor	yes	in 8.3.1	-	yes
SLURM	yes	-	-	yes
LSF >=9.1.1	yes	yes	RLIMIT_AS	yes

Memory monitoring

- Memory monitoring has been added to all the pilots
  - Extracts values from /proc/<PID>/smaps of all job processes
  - Value used for
    - Brokering
    - Killing of rogue jobs at sites which don't impose limits
    - Systematic studies of jobs



Inside ATLAS

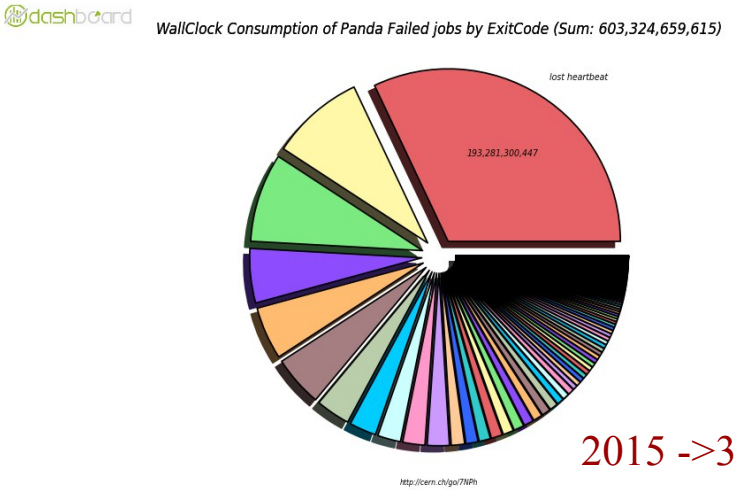
- To use maxrss with correct values sites can setup lo/hi memory PandaQueues.
  - They can be mapped to 1 batch queue with large values but the jobs will be brokered correctly to sites that can handle it
- To support sites that don't kill
  - The pilot kills above a certain threshold
    - Using twice the memory requested
      - Decision taken based on the plots from memory monitoring
      - Applied to both analysis and production jobs
    - Jobs exit gracefully
    - Production jobs are resubmitted to higher memory queues
      - Sooner than when the batch system kills resulting in lost heart beat

Brokering

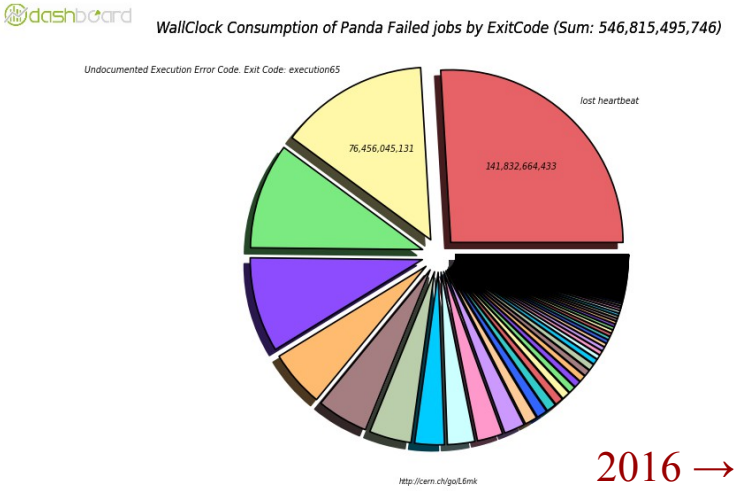
- Scout jobs sent to T1s to find how much memory the task needs
  - Memory value used to broker is that reported by the scouts as measured by the memory monitor
  - Scouts value compared to PandaQueues memory value to broker the jobs
  - PandaQueue value used for those jobs as memory parameter to pass to the batch system.
    - It's the max the jobs can use
  - Some jobs can use in excess of the scouts value and maybe brokered to queues that kills them
    - The system will re-broker the production ones

Lost heartbeat

- Lost Heartbeat is a catchall error message for when the panda server loses contact with a job for 6 hours
  - Many causes but a major one is the batch system killing on memory.
  - Largest component in wasted walltime
- Since introduction of memory handling in ATLAS and at sites progressive reduction of wasted walltime due to LH.



2015 ->32.0%



2016 → 25.9%

Conclusions

- One of the longest standing requests from sites is now satisfied
- ATLAS can better distribute the workload
  - The system was designed to support older batch systems that cannot support cgroups and don't handle memory correctly anymore
- The introduction of memory handling has reduced the weight of the “lost heartbeat” error
  - Partly because less jobs die
  - Partly because the errors are now better reported