

Of memory and batch systems

Alessandra Forti
ATLAS Jamboree
27 January 2016



Introduction

- This talk
 - Some reminders on memory meaning, submission parameters, and batch systems
 - Some memory plots
 - Brokering
 - Proposal
 - Batch systems/CEs consolidation
 - Shares at sites
 - Mcore/Score and prod/analysis shares
- Cloud computing (see Ryan slides)
- If time
 - Squid & Perfsonar



Memory according to the OS (reminder)

- 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



From ATLAS to kernel

Parameters table

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	?	?	?	?	?	?

(*) with cgroups support enabled

(**) ARC-CE has a HTCondor backend with *Limit* parameters which make it simpler

What really happens with the memory? i.e. what can we really limit? So far it seems we can limit only the address space if

Batch system	rss	rss+swap	vmem	needs cgroups to do sensible things
Torque/maui	-	-	RLIMIT_AS	N/A
*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	in 9.1.1	in 9.1.1	RLIMIT_AS	yes

Experiments	corecount	rss	rss+swap	vmem	cputime	walltime	comment
ALICE	-	-	-	-	-	-	-
ATLAS current	corecount	maxmemory	maxmemory	-	maxtime*ncores	maxtime	-
ATLAS future	corecount	maxrss	maxrss+maxswap	-	maxtime*ncores	maxtime	maxrss+maxswap really usable only by cgroups enabled sites
CMS	-	-	-	-	-	-	-
LHCb	-	-	-	-	-	-	-

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

Experiments

Batch System Parameters

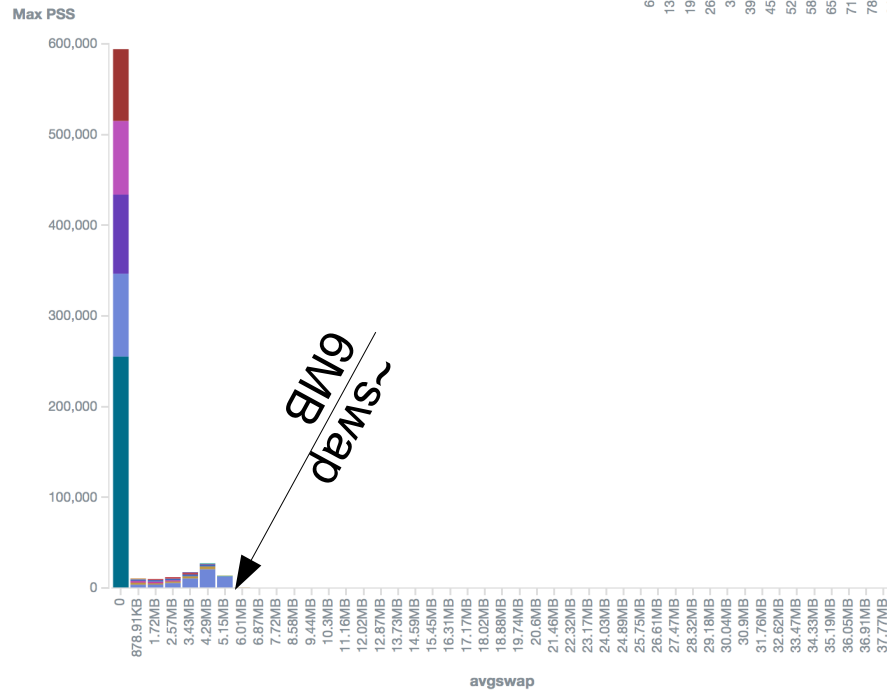
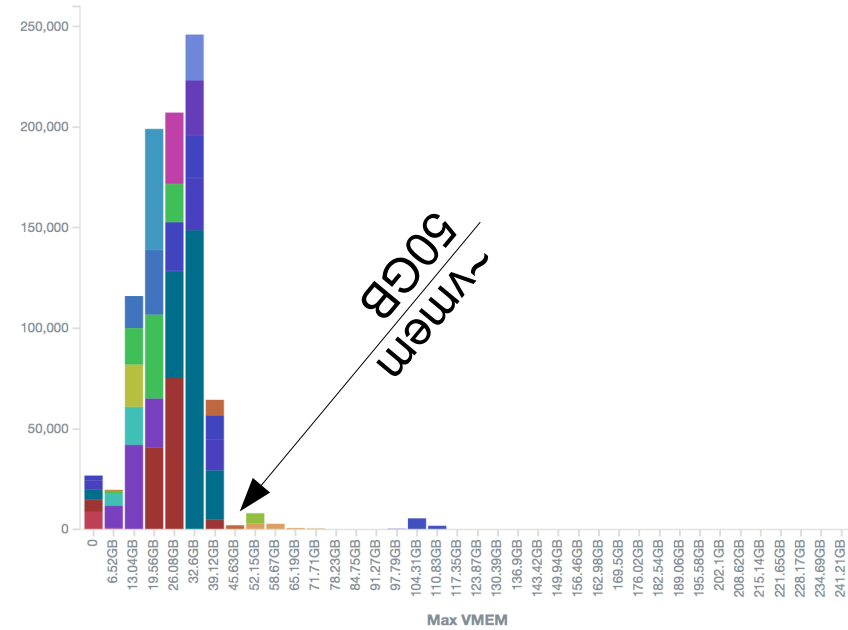
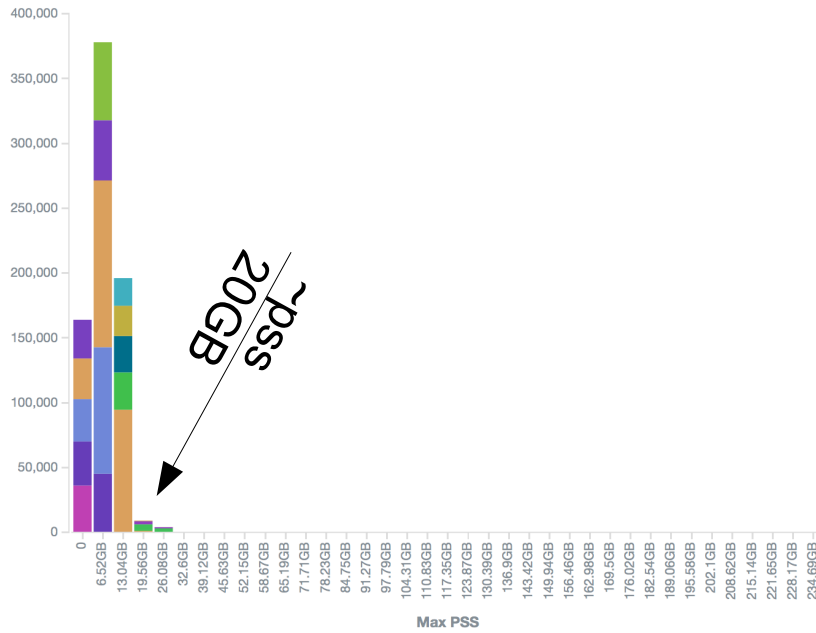


What batch systems do?

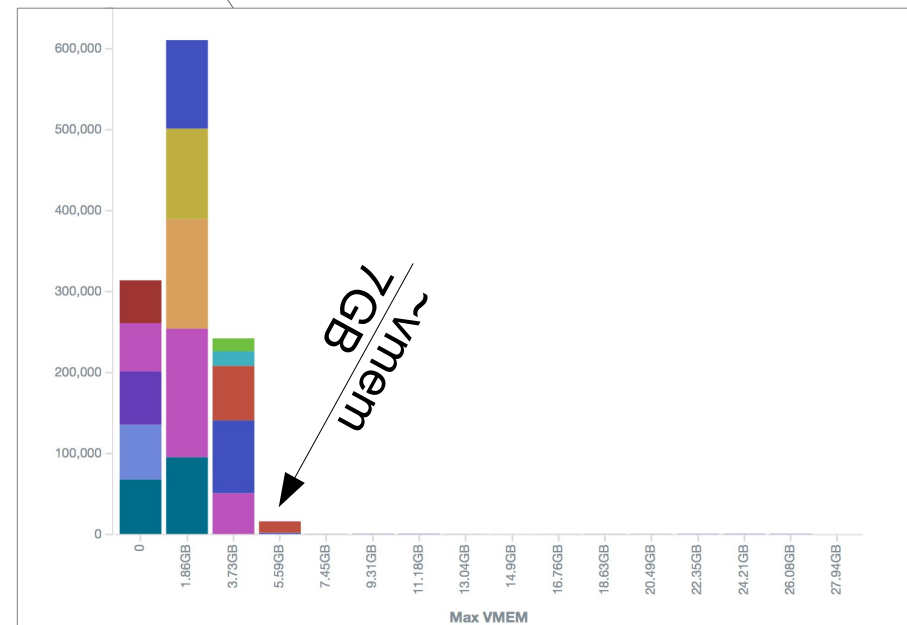
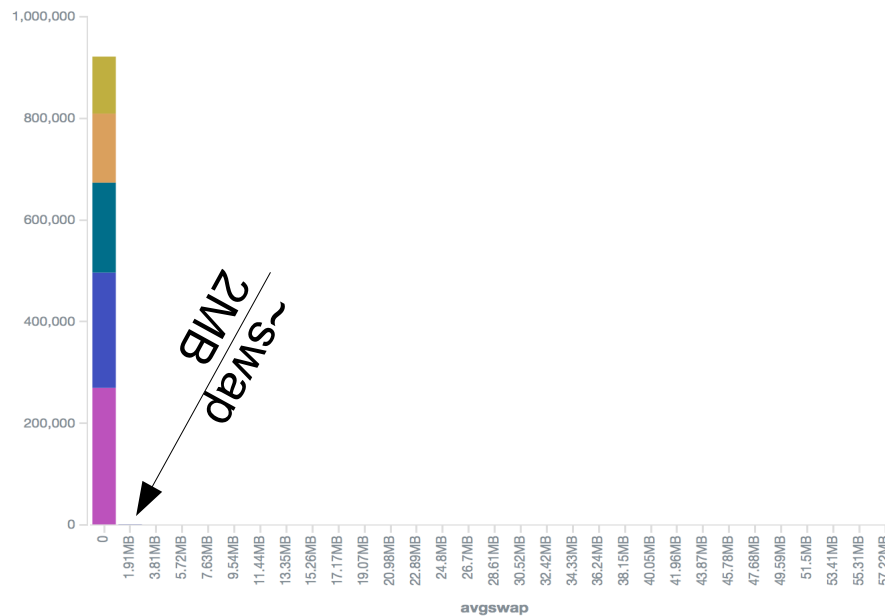
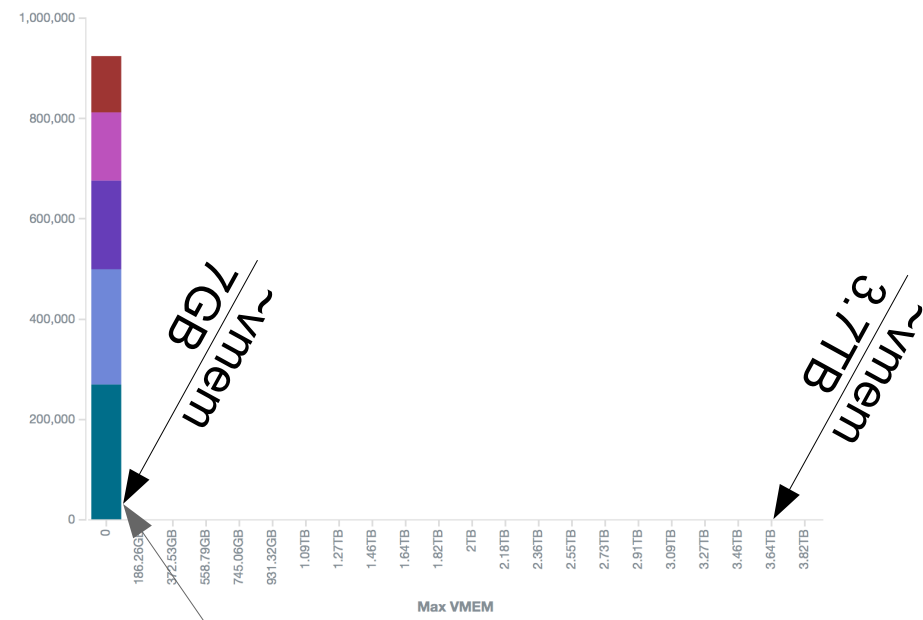
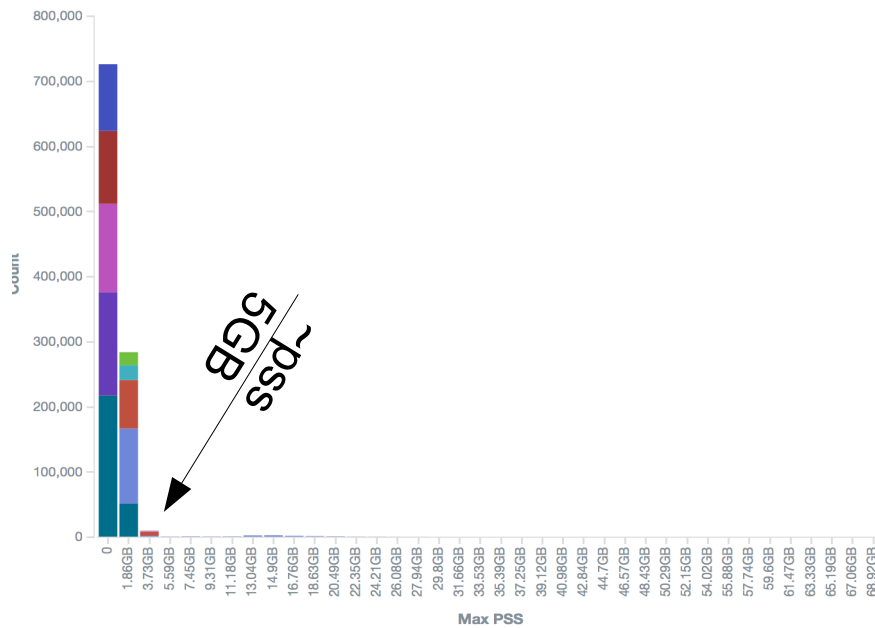
- Batch systems **without** cgroups
 - See the same RSS as reported in smaps
 - Kill on vmem which is **NOT** a physical memory measure
 - If you insist on this you need to set it at least 3 times the RAM requested by the job
- Sites with cgroups
 - Can setup soft and hard limits on the values the job 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



MCORE plots



SCORE plots



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
 - Compared to PandaQueues maxrss value to broker the jobs
 - PandaQueue value used for those jobs as memory parameter to pass to the batch system
 - Some jobs can use in excess of the scouts value and maybe brokered to queues that kills them



Proposal

- To use maxrss with correct values it would be useful to setup lo/hi memory queues PandaQueues.
 - They can be mapped to 1 batch queue with large values but at least we can broker correctly to sites that can handle it
- To support sites that don't kill we should resume the plan to get the pilot kill above a certain threshold
 - From the plots twice the memory requested could be a a good compromise.
 - Jobs exit gracefully
 - Can be resubmitted to higher memory queues
 - Likely sooner than when the batch system kills resulting in lost heart beat




Batch system transition

- 13/04/2014

Functionality	Torque/Maui	SLURM	HTCondor	USGE/SoGE	LSF
Number of sites 	101	10	10	14	7

- 26/01/2016

Functionality	Torque/Maui	SLURM	HTCondor	USGE/SoGE	LSF
Number of sites 	92	14	17	14	6
Developer Support	maui not supported	yes	yes	yes	yes

- Slight shift towards HTCondor and SLURM
 - Choice up to the site
 - HTCondor workshop
- Important to move to batch system that support cgroups
 - LSF and UGE can do it to but not for free



Shares at sites

- Fair shares have changed in august
 - Analysis: 25%
 - Production: 75%
 - SCORE: 20%
 - MSCORE: 80%
- Total: 40% SCORE and 60% MSCORE
 - 50% reported usage not that far off
 - Needs to understand why even without SCORE we cannot go beyond this.
- If you haven't looked into it yet a reminder that some recipes can be found
 - WLCG Multicore TF pages: Torque, HTCondor, SGE

