Batch systems parameter table

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 cgroups is not enabled.

Batch system	rss	rss+swap	vmem	needs cgroups todo 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

Batch systems parameters description

Torque/Maui

*GE

UGE 8.2.0 with cgroups

Matt Raso-Barnett, Sussex

When cgroups memory support is enabled it introduces new parameters to control it and augments existing parameters:

- m_mem_free replaces h_rss.

This is set as either the 'memory.limit_in_bytes' parameter, or the 'memory.soft_limit_in_bytes'.

The difference is in how the job is treated if it goes over it's limit: the first case is a hard limit, so if the job exceeds it's m_mem_free value it will be terminated immediately. The second case is a soft limit, so if the process exceeds the limit but the system as whole is not under memory pressure, then the process will be allowed to exceed the limit. When the system comes under pressure the limit is then applied and the process is forced down to the limit set.

I haven't done a huge amount of testing of how the soft limit works at the moment, but it's easy to switch between the two, so I would like to understand this better in the coming weeks, as it's something we are interested in using.

- h_vmem can be managed by cgroups, instead of being an rlimit.

Specifically, the limit becomes the 'memory.memsw.limit_in_bytes' parameter under the memory cgroup.

If h_vmem is set but no m_mem_free, then automatically a hard memory.limit_in_bytes is also set to the same size. If they are both set and, say, m_mem_free is higher than h_vmem, then m_mem_free will be reduced to the h_vmem limit.

Anyway, basically the story here is, yes, UGE can do the things you want to do.

But it might warrant a new line in the table, and I could potentially make a modified version of the sge_local_submit_attributes.sh script to use m_mem_free instead and do some testing with the soft memory limits.

Htcondor

Andrew Lahiff, RAL

- CPU time: there is no equivalent parameter, but you can restrict CPU time by including something like "RemoteSysCpu + RemoteUserCpu > 259200" in SYSTEM_PERIODIC_REMOVE or in PeriodicRemove in the job ClassAd or a number of other places. When a job submitted to an ARC CE requests a certain amount of CPU time the ARC CE adds it into PeriodicRemove.
- wall time: there is no equivalent parameter, but you can restrict wall time by including something like "CurrentTime - EnteredCurrentStatus > 259200" in SYSTEM_PERIODIC_REMOVE or in PeriodicRemove in the job ClassAd or a number of other places. When a job submitted to an ARC CE requests a certain amount of wall time the ARC CE adds it into PeriodicRemove.
- core count: RequestCpus
- memory (RSS): RequestMemory
    - Note on RAL setup: if a job specifies RequestMemory, **condor won't care at all if your job exceeds this memory if you're not using cgroups**. The job would need to have something like this defined: **PeriodicRemove = ResidentSetSize > RequestMemory*1000** in order to get condor to kill jobs which have exceeded their requested memory. The ARC CE adds this to the jobs it submits to condor. Alternatively, the site can have this in the condor config on the CEs: **SYSTEM_PERIODIC_REMOVE = ResidentSetSize > RequestMemory*1000**. There are a variety of other ways it could be done as well as you'd expect with condor. Once we've enabled cgroup memory limits on all our worker nodes we'll stop our ARC CEs from adding anything to do with memory into PeriodicRemove and just let cgroups handle everything.
- memory (Vmem): there isn't one by default, but you could make up your own way of doing this easily.
- swap: In condor 8.3.1 & above swap can be limited for jobs via cgroups: https://htcondor-wiki.cs.wisc.edu/index.cgi/tktview?tn=4417
    - Note on RAL setup: I haven't looked into this yet since it's in the dev series (8.3.x) while we're using the stable series (8.2.x) in production. Currently for our worker nodes with memory cgroup limits enabled we restrict the amount of swap available to the htcondor cgroup, so this places a limit of the total sway useable by all jobs on a node (but not jobs individually).

SLURM

Andrej Filipcic, Ljubiana

- corecount: --ntasks --nodes 1 (--nodes to force 1 node)
- memory: --mem-per-cpu (or --mem, it's mem per node, ARC uses mem-per-cpu)
  - with cgroups, corecount*mem-per-cpu will be the job limit or RSS
  - without cgroups, the memory estimate is not accurate, and it depends on which process tracker is enabled in slurm config.
- vmem: no per job setting, but VSizeFactor in slurm config can be set. if not set, there is no vmem limit
- cputime: no setting (cputime is automatically limited to corecount*walltime) wall time: --time

LSF

Computing Elements parameters

Computing Element	corecount	rss	rss+swap
CREAM-CE Glue1	JDL: CpuNumber= corecount; WholeNodes=false; SMPGranularity= corecount	GlueHostMainMemoryRAMSize	GlueHostMainMemoryVirtualS
CREAM-CE Glue2	JDL: CpuNumber= corecount; WholeNodes=false; SMPGranularity= corecount	GLUE2ComputingShareMaxMainMemory	GLUE2ComputingShareMaxVi
ARC-CE	(count = corecount)(countpernode = corecount)	memory(*)	-
HTCondor-CE	xcount	maxMemory	N/A

Experiments

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

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

Docs

- Glue Monitoring twiki
- Glue2.0 schema
- JDL Guide
- CREAM Information System providers
- ARC-CE xRSL manual

-- AlessandraForti - 2014-11-20

This topic: LCG > BSPassingParameters
Topic revision: r20 - 2017-01-17 - AlessandraForti