

Resource requirements on nodes

Andrej Filipčič
Jožef Stefan Institute, Ljubljana

ATLAS VO card

- No (hardware) changes foreseen for Run-2
 - ➔ Max used physical non-swap i386 memory size (MB) 2048
 - ➔ Max used physical non-swap x86_64 memory size (MB) 2048
 - ➔ Max size of scratch space used by jobs (MB) 20000
 - ➔ Max time of job execution (minutes) 3120
 - ➔ Job wall clock time limit (minutes) 5760
- For details, see
 - ➔ <http://operations-portal.egi.eu/vo/view/voname/atlas>
- But – configuration changes are needed
 - ➔ e.g. support for multicore, large-memory, long jobs

Complex ATLAS job requirements

- In Run-1:
 - ➔ Fixed resources per job
 - ➔ Well, mostly, we needed:
 - High-memory (4GB RSS) queues for special tasks
 - Long queues (4-day) to hospitalize jobs using excessive cputime
- Run-2: things will change drastically
 - ➔ Mixture of single/multicore jobs
 - ➔ Significant fraction of high-memory jobs
 - ➔ Mixture of long (simulation) and short (analysis) jobs

Benefits of ProdSys-2, Rucio

- Dynamic jobs, dynamic placement (distributed datasets)
- Each job will be sampled for resource usage:
 - ➔ Physical memory (RSS) profile
 - ➔ Swap profile
 - ➔ cpu usage profile
- Finished job reports will be registered in PanDA
- Scout job reports will be used to estimate the job resource limits for the rest of the jobs in a task
 - ➔ maxrss – maximum physical memory usage
 - ➔ maxcputime – maximum total cputime usage (multicore agnostic)
 - ➔ ... + any other metrics which might be useful (e.g. network, total I/O...)
- Tasks will have job resource requirements much better defined

Dynamic Scheduling and Resource Reservations

- Passing the job resource requirements to the batch system
 - ➔ Part of WLCG Multicore Task Force effort
 - ➔ ARC-CE supports it out of the box
 - ➔ CREAM-CE supports it through add-on local “blah” extension scripts – Task Force will standardize and prepare default scripts for each batch system in use
 - ➔ OSG/Condor needs some more investigation
- GOAL:
 - ➔ Optimize the resource usage
 - e.g. place 1x 6GB RSS reconstruction + 7x 1.4GB RSS simulation on a 16GB 8-core worker node
 - ➔ Limit the cputime/walltime for short jobs
 - Give higher priority
 - Use more opportunistic cores reserved other VOs (drains faster)
 - Backfilling the nodes which are draining for large (multicore) jobs

Dynamic scheduling cntd.

- Classic pilot model not suitable:
 - ➔ Relies on uniform job requirements (2 GB, 2 days)
- Static ATLAS PanDA queue model not scalable (high maintenance cost)
 - ➔ 5-10 different PanDA queues per site
 - ➔ 5-10 corresponding batch queues
- Dynamic queue model:
 - ➔ AGIS: single, base queue with site properties
 - e.g. maxcores 64, maxmemory 256GB, maxwalltime 6 days
 - ➔ PanDA/AGIS creates task queues dynamically, as needed
 - e.g. ncores=8, memory=16GB, walltime=6 hours
 - ➔ Auto pilot-factory sends the pilots with corresponding resource limits to CE/batch
 - ➔ Number of different queues should be kept small

Direct payload submission model

- arcControlTower:
 - ➔ Picks payload from PanDA
 - ➔ Submits predefined jobs to ARC-CE sites
- A) “network-less” mode
 - ➔ ARC-CE handles downloads/uploads
 - ➔ aCT does all communication to PanDA (heartbeats)
 - ➔ Jobs on WNs only do local processing
 - ➔ Used on Nordugrid, HPC sites (no-grid, no external network)
- B) Prefetched pilot mode (in development)
 - ➔ aCT picks a job payload, sends it to ARC-CE
 - ➔ ARC-CE submits a batch job with corresponding resource limits
 - ➔ Pilot job on the WN picks the prefetched payload, and starts in full-pilot mode (download or direct I/O, communication with PanDA, upload, completion steps)

-
- Neither dynamic queues or aCT prefetched mode are ready yet, but they will be before the Run-2

Memory Requirements

- Standard WN deployment: 2GB/core
- Limits are needed
 - 64-bit code not auto-protected by 3.3GB RSS, 4GB VMEM limit
 - Too many high-memory jobs can kill a node
 - Shared sites usually impose strict limits
- Recommendation:
 - Limit total RSS job usage, limit SWAP
 - RSS limit is only possible with cgroups enabled batch systems (SLURM, condor, UGE?)
 - Sites are encouraged to deploy it
 - VMEM only sites:
 - Jobs will request RSS, not VMEM → VMEM should be scaled by a factor of ~2
 - “ATLAS only” sites:
 - No limits on memory – the job-resource monitor will make sure the resources are not overused
- Switching to RSS for jobs/tasks will happen early next year

WN Disk space requirements

- In Run-2, same as in Run-1
- But the jobs will change:
 - ➔ All the output (and thus input) files will be optimized to fit in a 2-5GB range (possible with JEDI dynamic jobs), to reduce (or get rid of) the number of merge jobs
 - ➔ Most of the cpu-intensive jobs will be multi-core
 - Effective reduction of local disk space usage on the node
 - One 8-core job is I/O friendlier than 8 single-core jobs
- Direct I/O:
 - ➔ Recommended for analysis (xrootd)
 - ➔ SE infrastructure must scale to a high number of connections and bandwidth

Local network requirements

- **AtlasDerivation Framework:**
 - Most of the common analysis processing will be driven centrally through the train production
 - Less need for CPU intensive user analysis
- **Most of the user analysis is expected to:**
 - Process big amount of data
 - Use less cputime
 - A single analysis job on derived datasets can use 40MB/s
- **Local network:**
 - 1Gb/s on fat nodes (32-core, 64-core) will be a bottleneck → consider 10Gb/s or 40Gb/s infrastructure
 - SE – WN bandwidth might need to be improved

Batch System Recommendations

- Move away from unsupported torque/maui
 - ➔ New features like cgroups support likely not to be available
- Recommended, from ATLAS perspective, i.e. supporting RSS limits
 - ➔ HTCondor
 - ➔ SLURM
 - ➔ SGE
 - ➔ Other batch systems might do the job as well
- Move to generic batch queues (one)
 - ➔ Most batch systems support universal queues with dynamic job resource reservations
 - ➔ Hard partitioning between analysis/production, single/multicore is discouraged → sites will likely suffer inefficient use of resources (empty nodes)
 - ➔ Universal queue will be needed for dynamic PanDA queues
 - ➔ Deployment needs working parameter passing from CE to the batch

Summary

- We want to implement much better resource allocation and usage in the future
- We want to simplify site settings and batch queue configurations
- The central infrastructure will be developed in the next few months to be ready for Run-2