Experience using Slurm on ARIS HPC System

Nikos Nikoloutsakos

GRNET
Greek Research and Technology Network, Greece

hpc.gnret.gr

27 September 2016
1. Introduction
   - Who we are
   - System Overview

2. Slurm
   - Configuration
   - Administration - Monitoring

3. Issues

4. Feature request
Greek Research Technology Network

GRNET enables researchers from Greece to obtain access to the powerful national High Performance Computing system ARIS.

Advanced Research Information System

ARIS Infrastructure provides state-of-the-art supercomputing capabilities for large-scale scientific applications.

GRNET provides services to:

- Greece - Greek Academic Community
  - Greek Universities
  - Technological institutions
  - Research centers
- Europe
  - PRACE (Tier 1 system)
  - DECI
  - other EU Projects (Vi-seem, Eudat, EGI,...)
Support Team HPC provides:

- Management of Infrastructure
- User Support
  - Comprehensive end-user support
  - User support in operational problems
  - Documentation
  - Educational and Training Events
- Application Support - Transfer and optimizing application
- Peer-Review support and coordination
Peer-Review Access
The criteria for the evaluation:

- Scientific Excellence
- Impact of the proposed research
- The need for HPC resources
- Maturity and experience of the principal investigator and his/her team
- Feasibility of the project based on a technical evaluation and the availability of resources
**Project Types**

**Preparatory-Development Projects**

Execution of scalability tests, performance tests, resolve issues. Code porting, development, optimization.

- **Review:** Technical only
- **Call:** Always Open
- **Access:** 2-4 months

**Production Projects**

Projects that have the technical expertise to take advantage of available resources and are selected by the procedure of peer review

- **Review:** Technical-Scientific
- **Periodic Call:** 2 per year
- **Access:** 1 year
First pilot operational phase in June 2015

- 150 projects
- 400 Users
- 24 Organizations
- 300 software modules
- 120,000 jobs submitted,
  46M core hours (1 year)
- 25 scientific publications (up to now)

https://hpc.grnet.gr/results-publications/
Compute Power: 180 TFlops (HPL) #465 Top500 - iteration June 2015
426 compute nodes: IBM NextScale n360 M4
8520 cores: 2x (Intel E5 2680v2@2.8Ghz - 10 core) per node
27TB total memory: 64GB memory per node (8 RDIMMS, 1866 MHz)
Half-width, 1U systems grouped in 6U enclosures (12 nodes per enclosure)
- 6 Racks, 6 enclosures per rack.
- Diskless
- IBM 1PB GPFS, Tape Library IBM TS3500 6PB
- Max nominal power consumption: 162 KW (154 KW on HPL). 183 KW with air-cooling.
- Mellanox SX6536 648-Port Infiniband Director Switch
- FDR 56 Gbits / sec
- Fat tree non-blocking mode
- 450 QSFP+ Optical cables
- 5 Km fabric cables
ARIS - Expansion

- 44 gpu nodes: “2 x NVIDIA Tesla k40m” accelerated nodes.
  - Dell Power Edge R730
  - 2 x Intel Xeon E5-2660v3@2.6GHz
  - 64 GB RAM
- 18 phi nodes: “2 x INTEL Xeon Phi 7120p” accelerated nodes.
  - Dell Power Edge R730
  - 2 x Intel Xeon E5-2660v3@2.6GHz
  - 64 GB RAM
- 44 fat nodes
  - Dell PowerEdge R820
  - 4x Intel Xeon E5-4650v2@2.4GHz
  - 512 GB RAM
- IBM 1PB GPFS,
- Tape Library IBM TS3500 6PB
14 support nodes, NextScale x3650 M4 2 x E5-2640v2
2x Management Nodes, 2x Login Nodes, 10x service nodes
Monitoring software xCAT, Nagios, Ganglia, BMS (Business Management System) Dell OpenManage, MRTG
Scheduler SLURM 14.11.8
XDMoD, UMGMT (User Management Tool) in house
ONE cluster "ARIS"

<table>
<thead>
<tr>
<th>Partition</th>
<th>Description</th>
<th>Nodes</th>
</tr>
</thead>
<tbody>
<tr>
<td>compute</td>
<td>Thin nodes</td>
<td>426</td>
</tr>
<tr>
<td>gpu</td>
<td>GPU nodes</td>
<td>44</td>
</tr>
<tr>
<td>phi</td>
<td>PHI nodes</td>
<td>18</td>
</tr>
<tr>
<td>fat</td>
<td>FAT nodes</td>
<td>24</td>
</tr>
<tr>
<td>taskp</td>
<td>Serial queue</td>
<td>20</td>
</tr>
</tbody>
</table>

Default timelimit 2 days
**Consumable Resources**

- \texttt{SelectTypeParameters= \texttt{CR\_CORE\_MEMORY}}
- Shared mode - unless user specifies \texttt{--exclusive}

**Resource Limits**

- \texttt{AccountingStorageEnforce = associations,limits,safe}

**Generic Resource (GRES) Scheduling**

- \texttt{GresTypes = gpu,mic}
- mic offload mode only
**MpiDefault = pmi2**

Supports MPI implementation being used on system: Intelmpi, OpenMPI, mvapich2

**The larger the job, the greater its job size priority.**

**PriorityFavorSmall = NO**

**Accounting Gather**

- `AcctGatherEnergyType=acct_gather_energy/ipmi`
- `AcctGatherInfinibandType=acct_gather_infiniband/ofed`
- `JobAcctGatherType = jobacct_gather/linux`
Multifactor Priority

- PriorityType = priority/multifactor
- PriorityWeightAge = 100
- PriorityWeightFairShare = 1000
- PriorityWeightJobSize = 1000
- PriorityWeightPartition = 0
- PriorityDecayHalfLife = 00:00:00
- PriorityUsageResetPeriod = WEEKLY
- PriorityMaxAge = 30-00:00:00
- PriorityWeightQOS = 0
### Fair Tree Fairshare

- **PriorityFlags** = FAIR_TREE
- **PriorityCalcPeriod** = 02:00:00

### Backfill Scheduling

- **SchedulerType** = sched/backfill
$mybudget

Core Hours Allocation Information for account : testproj

<table>
<thead>
<tr>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Allocated Core Hours</td>
<td>10000000.00</td>
</tr>
<tr>
<td>Project Consumed Core Hours</td>
<td>3410968.00</td>
</tr>
<tr>
<td>User Consumed Core Hours</td>
<td>523.00</td>
</tr>
<tr>
<td>Percentage of Project Consumed</td>
<td>34.11</td>
</tr>
<tr>
<td>Percentage of User Consumed</td>
<td>0.01</td>
</tr>
<tr>
<td>Account limits (Job,Node,Core)</td>
<td>0 0 0</td>
</tr>
</tbody>
</table>

$myreport

Time reported in CPU Hours

<table>
<thead>
<tr>
<th>Cluster</th>
<th>Account</th>
<th>Login</th>
<th>Proper Name</th>
<th>Used</th>
<th>Energy</th>
</tr>
</thead>
<tbody>
<tr>
<td>nimos</td>
<td>aris</td>
<td>begin</td>
<td>testproj nikolout+</td>
<td>Nikos Nikolout+</td>
<td>371</td>
</tr>
</tbody>
</table>
Users Management Tool

- Tool to manage project proposals and user access on the system.
- Associate project proposals to slurm accounting information
- Keep Track start end dates per project, Extensions: core hours-access period
- Project status, send alert emails to users
- Statistics consumed core-hours(%) per project

in development: Ruby on Rails
Introduction
Who we are
System Overview
Slurm
Configuration
Administration - Monitoring
Issues
Feature request

Allocated Nodes for ARIS compute

Allocated Nodes for ARIS taskp

Allocated Nodes for ARIS phi

Allocated Nodes for ARIS gpu
Helps users prepare batch job scripts for Slurm at ARIS.

Acknowledgment BYU Job Script Generator
https://github.com/BYUHPC/BYUJobScriptGenerator
Problem:
Reservation (daily) had 20 nodes, 15 where active, 5 where active by same user but for other job.
1 node (from 15) died, unable to reschedule.
Feature request

- More verbose error messages:
  Users could figure why a job is rejected.
  More information about which limit violated

- MPI Task 0: may need more memory
  Ability to specify less processes on first node.

- Allocation per GRES(gpu, mic) not only cpu ch

What’s Next

- upgrade to version 16
Thank you!