Migration from Grid Engine to HTCondor

The talk provides details about the goals and general setup of this migration. It further focuses on Kerberos support, registry integration and node operating automation.

DESY/IT-Systems: Thomas Finnern Martin Flemming Christoph Beyer Yves Kemp
Outline of Talk

• Plan and Status
• Approach and Policies
• Kerberos and AFS Integration
• User Registry Integration
• Node Automation and Control
• Outlook and Conclusions

• Main Focus: BIRD
We have a Plan!

<table>
<thead>
<tr>
<th>Past</th>
<th>Calendar Reservation Tool</th>
<th>Son Of Grid Engine SoGE</th>
<th>Torque + MySched</th>
<th>Cream CE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Work in Progress</td>
<td></td>
<td>SoGE</td>
<td>HTC Pilot</td>
<td>HTC Grid</td>
</tr>
<tr>
<td>Running</td>
<td></td>
<td></td>
<td></td>
<td>ARC CE</td>
</tr>
<tr>
<td>Future</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

[Diagram showing various components and connections like HPC Maxwell, BIRD NAF, GRID, etc.]
Approach and Policies for HTCondor

- User Friendly Migration
  - Kerberos and AFS Integration
  - Growing BIRD Pilot with 500+ Cores
- 1 “Master” Server supporting BIRD
  - Collector and Negotiator
  - Quota/Fairshare configuration
- 2 BIRD-Scheduler
  - For job-friendly service restart
  - With secure token access
  - Policy settings on scheduler
  - Check projects against registry
  - Forward submitter project to worker
- 10 Remote Submit Hosts
  - User Login for job preparation
  - Project specific default settings
  - No dependencies from/to running jobs
- 500 Worker Nodes
  - Common node setup
  - Currently 30+ in Pilot
- Optional GRID Integration
  - Add GRID share to negotiator
  - Add GRID scheduler
  - Add GRID worker resource
Kerberos and AFS Integration

- Kerberos authentication
- AFS as shared filesystem
- Valid tokens during job run time
- 1 week maximal job run time
- Secure token generator on protected servers
- Consistently prolong current token
- Generation of AFS tokens out of Kerberos tokens
Kerberos and AFS Integration

1. Get_Token.sh
2. Token_Shepherd_Sched.sh
3. Token_Shepherd_Worker.sh
4. (Job_Wrapper.sh)

kinit, aklog
arc(Take), kinit, condor_aklog
kinit-prolong, condor_aklog
Token_Update.htc
User Registry Integration

- Set adequate project on group submit host
- User may switch to another project
- Project defaults to primary registry group
- Resulting project will be checked against registry
- Resulting project will define fairshare/quota group
- Resulting project will be set on worker as primary group
User Registry Integration

1. Generate_UserMap.sh
2. Condor_submit
   cron, ldap, Transforms.htc
3. Job_Wrapper.sh
4. Quota.htc
Node Automation and Control

- Automated Operation of Nodes
  - For Problems (e.g. node failures)
  - For Service (e.g. cluster kernel update)
  - Manually/CLI or by scripting
- Disable, drain, reboot and reset Nodes
  - No preemption or job killing
  - No specific operator knowledge needed
- Authentication
  - Based on Authenticated Remote Command (arc)
  - User based (Operators, Admins) Batchnode.sh
  - Server based (Scripts, Cluster Reboot, Kernel Upgrades, …)
  - Node based (local Monitoring, …)
- Transparent
  - All states in one view
  - Hourly status update
  - Sets/resets exact icinga downtimes
  - Works for all pools
    - SoGE, GRID, PILOT, TEST, …
Node Automation and Control

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>Batchnode.sh</td>
<td>waiting</td>
<td>Node.cron</td>
<td>Server and Node Actions</td>
</tr>
<tr>
<td></td>
<td>Time</td>
<td>Cron.hourly</td>
<td>Scripts and Tools</td>
</tr>
</tbody>
</table>
Outlook and Conclusions

• BIRD/NAF
  • „Proof of Concept“ for planned feature done
  • Waited for HTCondor 8.7.3 providing
    • Full Kerberos and AFS support
    • Transforms for Scheduler Policy Settings
  • Pilot running with first users
  • Smooth Transition appears to be possible
• GRID and BIRD/NAF
  • Some common operating tools running
  • Pilot HTC Server are prepared
• Next Steps may be …
  • More BIRD and GRID Integration
  • Docker for different operating system flavours
  • Backfill of HPC resources with HTCondor