Condor Scalability and Management at Brookhaven National Laboratory

Alexander Withers
alexw@bnl.gov
CondorWeek 2007
Overview of Condor at BNL

- RHIC/USATLAS Computing facility
  - Condor is the primary batchsystem in use
  - LSF still used by some users (provides global license counters)
- Not the only BNL group to use Condor
  - PHENIX experiment uses Condor to help power their 600MB/s DAQ/production facility
- USATLAS Physics Applications Software group
- 4800+ processors running Condor
  - 5 pools, 3 central managers, 1 quill server
  - 4 grid gatekeepers, 100+ submit nodes
Policies in Use

- In general pools use either suspension or machine Rank with MaxJobRetirementTime to define a notion of priority.
- Users add custom flags to their jobs to define the type of job.
- Other flags are added by Condor upon submission.
- The startd enforces restrictions by also looking at Owner and other job attributes.
- Preempt for out of control jobs.
- Preemption_Requirements and MaxJobRetirementTime for fairness between users.
Example START Expression

Start = (((RealExperiment == "atlas") && (VirtualMachineID >= 7) && ((TARGET.RACF_Group != "short" || TARGET.RACF_Group != "dial" || Owner != "usatlas2" || (stringListMember("acas0201", "acas0200,acas0201,acas0202,acas0203,acas0204") && TARGET.RACF_Group == "lcg-ops") || TARGET.RACF_Group == "lcg-dteam") && (RemoteWallClockTime < 5400))) || ((RealExperiment == "atlasev") && ((VirtualMachineID >= 7) && (VirtualMachineID <= 5)) && ((TARGET.RACF_Group == "usatlas") || (TARGET.RACF_Group == "usatlas-grid" || (stringListMember("acas0201", "acas0200,acas0201,acas0202,acas0203,acas0204") != FALSE && TARGET.RACF_Group == "lcg-atlas") || TARGET.RACF_Group == "bnl-local") && (((vm7_Activity == "Busy") + (vm7_Activity == "Retiring") + (vm8_Activity == "Retiring") + (vm8_Activity == "Busy") + (vm5_Activity == "Busy") + (vm5_Activity == "Retiring") + (vm6_Activity == "Retiring") + (vm6_Activity == "Busy")) < 2))) || (((RealExperiment == "atlasev") && (VirtualMachineID <= 5)) && ((TARGET.RACF_Group == "grid" || (stringListMember("acas0201", "acas0200,acas0201,acas0202,acas0203,acas0204") != FALSE && TARGET.RACF_Group == "lcg") && (((vm7_Activity == "Busy") + (vm7_Activity == "Retiring") + (vm8_Activity == "Retiring") + (vm8_Activity == "Busy") + (vm5_Activity == "Busy") + (vm5_Activity == "Retiring") + (vm6_Activity == "Retiring") + (vm6_Activity == "Busy")) < 2))) || (((RealExperiment == "atlasev") && (VirtualMachineID <= 3) && (VirtualMachineID >= 1) && (VirtualMachineID < 3)) && (((TARGET.RACF_Group == "gridgr01") || TARGET.RACF_Group == "gridgr02") || TARGET.RACF_Group == "gridgr03") || TARGET.RACF_Group == "gridgr04") || TARGET.RACF_Group == "gridgr05") || TARGET.RACF_Group == "gridgr06") || TARGET.RACF_Group == "gridgrXX") || TARGET.RACF_Group == "gridgr08") || TARGET.RACF_Group == "gridgr09") || TARGET.RACF_Group == "gridgr10") || TARGET.RealExperiment == "atlas") && (((vm7_Activity == "Busy") + (vm7_Activity == "Retiring") + (vm8_Activity == "Busy") + (vm5_Activity == "Busy") + (vm5_Activity == "Retiring") + (vm6_Activity == "Retiring") + (vm6_Activity == "Busy") + (vm3_Activity == "Busy") + (vm3_Activity == "Retiring") + (vm4_Activity == "Retiring") + (vm4_Activity == "Busy") + (vm4_Activity == "Retiring") + (vm4_Activity == "Busy") < 2)))) && (Owner != "jalex" && Owner != "grau" && Owner != "smithj4") && FALSE == FALSE)
Increase in Usage and Resources

• >400 users actively using Condor
• >10000 job slots
• Past 3 months: 2.8m jobs, 6.2m wallclock hours
• Computing resources added every year
  • New machines and Xen: even more job slots
  • Growth has been nonlinear, can we handle next year?
• Farm needs to be occupied with jobs
• Users need access to resources in a fair manner without significant delays
• Problem: one central manager may not be able to handle the load, how do we plan for the future?
Divide and Conquer

- Solution: divide the work load between three machines and divide the resources between five pools
- Use flocking to create one virtual pool
- Foreign jobs are immediately evicted if the resource is wanted by a local job
- A user’s job will run on the other pools unless they prevent it from doing so
- Response time has been very good, thus allowing growth
- Other measures to increase response time from negotiator:
  - SIGNIFICANT_ATTRS (now automatic)
  - Increased negotiation cycle
### Condor pool stats for total

<table>
<thead>
<tr>
<th>Metric</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maximum Waiting</td>
<td>150469.33</td>
</tr>
<tr>
<td>Average Waiting</td>
<td>48171.53</td>
</tr>
<tr>
<td>Maximum Running</td>
<td>4753.83</td>
</tr>
<tr>
<td>Average Running</td>
<td>3826.62</td>
</tr>
<tr>
<td>Maximum Suspended</td>
<td>1685.41</td>
</tr>
<tr>
<td>Average Suspended</td>
<td>392.22</td>
</tr>
<tr>
<td>Maximum CPUs</td>
<td>4332.01</td>
</tr>
<tr>
<td>Average CPUs</td>
<td>4194.93</td>
</tr>
</tbody>
</table>

**Total Running Hours**: 10347815.48  
**Total Suspended Hours**: 1060631.76

- **No. of CPUs**  
- **No. of Waiting Jobs**  
- **No. of Condor Running Jobs**  
- **No. of Suspended Jobs**

Generated Tue Apr 24 10:37:16 EDT 2007
Quill

- One quill server to handle all five pools
- First server (dual Xeon 3GHz, 4GB RAM, and SCSI drives with SW raid1) could not handle the load
  - `condor_q` would sometimes take 10 minutes
  - 100+ submit nodes being activity used
  - Optimizing postgresql didn’t seem to help
- Investigated a variety of small scale storage hardware and configurations
  - Found it difficult to quantitatively measure Quill’s performance
  - Used benchmarks to model the behavior of Quill that we were seeing
Quill, cont.

• Our tests involved a variety of factors: SATA vs. SAS, HW raid vs SW raid, etc.
• Baseline: SATA systems with SW raid10 and raid5 with minimum number of drives
• Results: SAS, HW raid, raid10 (no surprise), more spindles helps too
• New server with 8GB of RAM, 6 drives for data, system and postgresql logs on other disks
• Other benefits: shared memory set to use half the system RAM and increased working RAM (postgresql specific parameters)
• Please contact me for specifics: alexw@bnl.gov
Monitoring and Maintenance

• 5 pools each with its own complex policy
  • Important to monitor and record usage
• We use several features in Condor to make monitoring easy
• Historical data stored in MySQL and RRDs
  • Quill is used as well to collect historical data
  • Quill’s schema is sometimes difficult to deal with
Making Queries Easy

- Many submit machines: not easy to query the schedds
- User uses custom job attributes to target job
- Insert job’s attributes into machine’s classad: STARTD_JOBEXPRS
- Make queries using these inserted attributes to show how many jobs are running where
- Insert other attributes to get an idea of who is using the resources, how much memory, disk usage, etc.
```
[root@acas0010 ~]# condor_status -constraint 'RACF_Group == "short"'

<table>
<thead>
<tr>
<th>Name</th>
<th>OpSys</th>
<th>Arch</th>
<th>State</th>
<th>Activity</th>
<th>LoadAv</th>
<th>Mem</th>
<th>ActvtyTime</th>
</tr>
</thead>
<tbody>
<tr>
<td>vm13@acas0015</td>
<td>LINUX</td>
<td>INTEL</td>
<td>Claimed</td>
<td>Busy</td>
<td>0.970</td>
<td>1024</td>
<td>0+00:07:43</td>
</tr>
<tr>
<td>vm16@acas0015</td>
<td>LINUX</td>
<td>INTEL</td>
<td>Claimed</td>
<td>Busy</td>
<td>0.990</td>
<td>1024</td>
<td>0+01:13:16</td>
</tr>
<tr>
<td>vm13@acas0016</td>
<td>LINUX</td>
<td>INTEL</td>
<td>Claimed</td>
<td>Busy</td>
<td>0.950</td>
<td>1024</td>
<td>0+00:09:55</td>
</tr>
<tr>
<td>vm15@acas0110</td>
<td>LINUX</td>
<td>INTEL</td>
<td>Preempting</td>
<td>Vacating</td>
<td>0.930</td>
<td>1024</td>
<td>0+00:00:10</td>
</tr>
<tr>
<td>vm16@acas0110</td>
<td>LINUX</td>
<td>INTEL</td>
<td>Claimed</td>
<td>Busy</td>
<td>0.960</td>
<td>1024</td>
<td>0+00:52:49</td>
</tr>
<tr>
<td>vm7@acas0188.</td>
<td>LINUX</td>
<td>INTEL</td>
<td>Claimed</td>
<td>Busy</td>
<td>1.020</td>
<td>1024</td>
<td>0+01:01:05</td>
</tr>
<tr>
<td>vm8@acas0188.</td>
<td>LINUX</td>
<td>INTEL</td>
<td>Claimed</td>
<td>Busy</td>
<td>0.980</td>
<td>1024</td>
<td>0+00:52:30</td>
</tr>
<tr>
<td>vm7@acas0189.</td>
<td>LINUX</td>
<td>INTEL</td>
<td>Claimed</td>
<td>Busy</td>
<td>0.330</td>
<td>1024</td>
<td>0+01:01:44</td>
</tr>
<tr>
<td>vm8@acas0190.</td>
<td>LINUX</td>
<td>INTEL</td>
<td>Claimed</td>
<td>Busy</td>
<td>1.010</td>
<td>1024</td>
<td>0+01:01:58</td>
</tr>
</tbody>
</table>

Total Owner Claimed Unclaimed Matched Preempting Backfill

<table>
<thead>
<tr>
<th></th>
<th>INTEL/LINUX</th>
<th>200</th>
<th>0</th>
<th>190</th>
<th>0</th>
<th>0</th>
<th>10</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total</td>
<td>200</td>
<td>0</td>
<td>190</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>10</td>
<td>0</td>
</tr>
</tbody>
</table>
```
Condor pool queue usage stats for usatlas

No. of CPUs
short lcg grid nonatlas_grid general_q
prod distr_analysis usatlas usatlas-grid bnl-local

Generated Tue Apr 24 14:29:32 EDT 2007
Usage for anatrain

```
condor_status -pool condor02.rcf.bnl.gov:9662 -constraint 'CPU_Type == "crs" && Turn_Off == False'
```

Machines: 492
Owner: 0
Claimed: 490
Unclaimed: 2
Matched: 0
Preempting: 0

- claudius@bnl.gov: 15 (r: 150, i: 5, h: 0)
- dask@bnl.gov: 3 (r: 3, i: 0, h: 0)
- phnxreco@bnl.gov: 287 (r: 548, i: 4, h: 0)
- anatrain@bnl.gov: 53 (r: 96, i: 2844, h: 0)
- manuyen@bnl.gov: 132 (r: 196, i: 0, h: 0)

- vml@rcas2043.rcf.bnl.gov 1.01 Claimed Retiring 04/26-14:59:23 phnxreco@bnl.gov rcrsuser
- vm2@rcas2043.rcf.bnl.gov 1.00 Claimed Retiring 04/26-14:59:23 phnxreco@bnl.gov rcrsuser
- vm3@rcas2043.rcf.bnl.gov 0.96 Claimed Busy 04/26-11:54:01 phnxreco@bnl.gov rcrsuser4.rcf
Dynamic Policy Changes

• Complex policy on each pool that allows a wide variety of job types to run

• Convenient to restrict certain jobs from running on certain nodes

• Solution: special machine attributes that can be set remotely
  • SETTABLE_ATTRS_CONFIG, HOSTALLOW_CONFIG, ENABLE_*_CONFIG

• Machine attribute is placed in START, RANK, etc. expression
  • condor_config_val -name rcas6006 -startd -set "CRS_Turn_Off = True"
Dynamic Policy Example

Prevent “crs” jobs from running but allow the current ones to finish

# local job start expression
CRS_Turn_Off = False
LOCAL_JOB = (RealExperiment == $(CPU_Experiment) && \
  ((Job_Type == "cas" && (VirtualMachineID == 1 || VirtualMachineID == 2)) || \
  (Job_Type == "osg" && (VirtualMachineID == 1 || VirtualMachineID == 2)) || \
  (Job_Type == "crs" && Owner == $(CPU_User) && $(CRS_Turn_Off) == False))

[root@condor01 CONFIG]# condor_status -constraint 'CRS_Turn_Off == True'

<table>
<thead>
<tr>
<th>Name</th>
<th>OpSys</th>
<th>Arch</th>
<th>State</th>
<th>Activity</th>
<th>LoadAv</th>
<th>Mem</th>
<th>ActvtyTime</th>
</tr>
</thead>
<tbody>
<tr>
<td>rcas6004.rcf. LINUX</td>
<td>LINUX</td>
<td>INTEL</td>
<td>Claimed</td>
<td>Busy</td>
<td>2.720</td>
<td>8192</td>
<td>2+04:40:42</td>
</tr>
<tr>
<td>rcas6006.rcf. LINUX</td>
<td>LINUX</td>
<td>INTEL</td>
<td>Owner</td>
<td>Idle</td>
<td>3.160</td>
<td>8192</td>
<td>0+00:20:04</td>
</tr>
<tr>
<td>rcas6115.rcf. LINUX</td>
<td>LINUX</td>
<td>INTEL</td>
<td>Unclaimed</td>
<td>Idle</td>
<td>0.000</td>
<td>8192</td>
<td>0+03:00:04</td>
</tr>
<tr>
<td>rcas6156.rcf. LINUX</td>
<td>LINUX</td>
<td>INTEL</td>
<td>Unclaimed</td>
<td>Idle</td>
<td>0.140</td>
<td>8192</td>
<td>0+01:50:04</td>
</tr>
</tbody>
</table>
Extending Condor

- Make heavy use of Condor’s cron facility
- Insert useful machine attributes such as 5 min. and 15 min. load
  - Can’t use these attributes in any startd expressions
- Usually rely on `NEGOTIATOR_REQUIREMENTS`
- Other attributes are used by jobs
  - One example: projected disk usage
    - User transfer text file predicting how much disk space they will use (based on file placed in `_CONDOR_SCRATCH_DIR`)
    - Other jobs avoid machines where disk space
First Gold Beam-Beam Collision Events at RHIC at 30+30 GeV/c per beam recorded by STAR

Questions, Comments?
alexw@bnl.gov