Monitoring Primer
HTCondor Week 2017
Todd Tannenbaum
Center for High Throughput Computing
University of Wisconsin-Madison
Ad types in the condor_collector

› \textit{startd} ads
  • An ad for each slot on each machine
  • \textit{State} = Unclaimed | Claimed | Drained \ldots
  • \textit{Activity} = Busy | Idle | Retiring | Vacating \ldots
  • CPUs, Memory, Disk, GPUs, \ldots

› \textit{submitter} ads
  • An ad for each unique user
  • RunningJobs, IdleJobs, HeldJobs, \ldots

› \textit{schedd}, \textit{master}, \textit{negotiator}, \textit{collector} ads
  • One ad from each daemon instance
Q: How many slots are running a job?

A: Count slots where State == Claimed (and Activity != Idle)

How?
Obvious solutions aren’t the best

% condor_status

slot7@ale-22.cs.wi LINUX  X86_64 Claimed  Busy  0.990  3002  0+00:28:24
slot8@ale-22.cs.wi LINUX  X86_64 Claimed  Busy  1.000  3002  0+00:14:13
slot1@ale-23.cs.wi LINUX  X86_64 Unclaimed Idle  0.920  3002  0+00:00:04

› condor_status | grep Claimed | grep –v Idle | wc –l
  • Output subject to change, wrong answers, slow

› condor_status –l | grep Claimed | wc –l
  • Wrong answers, really slow
Use constraints and projections

condor_status [-startd | -schedd | -master...]
- constraint <classad-expr>
- autoformat <attr1, attr2, ...>

condor_status -startd
  -cons 'State=="Claimed" && Activity!="Idle"
  -af name  | wc -l
Q: Which slots are running on machines where NFS is broken?

› Ask startd to run a script/program to test health of NFS

STARTD_CRON_JOB_LIST = tag
STARTD_CRON_tag_EXECUTABLE = detect.sh

• Script returns a ClassAd with attribute NFS_Broken = True | False

› condor_status -cons 'NFS_Broken==True'

› Could specify customized output (i.e. a report) for condor_status to display broken machines

Q: How many CPU cores are being utilized?

Sum the Cpus attribute for each slot that is Claimed and Busy:

```
% condor_status -startd \ 
  -cons 'State=="Claimed" && Activity!="Idle"' \ 
  -af Cpus | less
```

```
1
1
4
4
...
```

```
% condor_status -startd \ 
  -cons 'State=="Claimed" && Activity!="Idle"' \ 
  -af Cpus | st
```

Simple Statistics from command line
https://github.com/nferraz/st
Could have a cron job run every minute…

```
#!/bin/sh
echo `date`, ; condor_status \\
-con 'State=="Claimed" && Activity!="Idle"' \\
-af Cpus | st --sum
```

What if you have hundreds or thousands of metrics?

• `COLLECTOR_QUERY_WORKERS = 5000`?

How about query the collector just once per minute for all attributes needed to compute all metrics?
Ganglia and condor_gangliad

- condor_gangliad queries the condor_collector once per minute
  - DAEMON_LIST = MASTER, GANGLIAD,…
- condor_gangliad has config file to filter and aggregate attributes from the ads in the condor_collector in order to form metrics
- Forwards these metrics to Ganglia, which stores these values in a database and provides graphs over the web
Example metric definitions in condor_gangliad

[ 
  Name   = "CpusInUse";
  Aggregate = "SUM";
  Value   = Cpus;
  Requirements = State=="Claimed" && Activity!="Idle";
  TargetType = "Machine";
]

[ 
  Name   = "CpusNotInUse";
  Aggregate = "SUM";
  Value   = Cpus;
  Requirements = State!="Claimed" || Activity=="Idle";
  TargetType = "Machine";
]
Add a graph to a view dashboard: 
/var/lib/ganglia/view_Miron.json

```json
{
  "aggregate_graph":"true",
  "host_regex":[
    {
      "regex":"cm.chtc.wisc.edu"
    }
  ],
  "metric_regex":[
    {
      "regex":"(Cpus(InUse|NotInUse)"
    }
  ],
  "graph_type":"stack",
  "vertical_label":"cores",
  "title":"CPU Core Utilization"
}
```
Voila!
Why are cores not in use?

[ 
Name   = "CpusNotInUse_LowMemory";
Aggregate = "SUM";
Value   = Cpus;
Requirements = State=="Unclaimed" && Memory < 1024;
TargetType = "Machine";
]

[ 
Name   = "CpusNotInUse_Draining";
Aggregate = "SUM";
Value   = Cpus;
Requirements = State=="Drained";
TargetType = "Machine";
]
Unused Core Reasons

Unused Core Reasons last week

<table>
<thead>
<tr>
<th>Day</th>
<th>cm.chtc.wisc.edu CpusNotInUse_ClaimedIdle</th>
<th>cm.chtc.wisc.edu CpusNotInUse_Draining</th>
<th>cm.chtc.wisc.edu CpusNotInUse_LowMemory</th>
<th>cm.chtc.wisc.edu CpusNotInUse_NoJobsMatch</th>
<th>cm.chtc.wisc.edu CpusNotInUse_Owner</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mon</td>
<td>Now: 50.2</td>
<td>Now: 248.9</td>
<td>Now: 547.5</td>
<td>Now: 1.7k</td>
<td>Now: 35.9</td>
</tr>
<tr>
<td></td>
<td>Min: 1.0</td>
<td>Min: 74.5</td>
<td>Min: 111.0</td>
<td>Min: 197.5</td>
<td>Min: 15.0</td>
</tr>
<tr>
<td></td>
<td>Avg: 134.6</td>
<td>Avg: 215.3</td>
<td>Avg: 809.8</td>
<td>Avg: 1.2k</td>
<td>Avg: 34.3</td>
</tr>
<tr>
<td></td>
<td>Max: 1.1k</td>
<td>Max: 390.6</td>
<td>Max: 2.6k</td>
<td>Max: 3.1k</td>
<td>Max: 54.0</td>
</tr>
<tr>
<td></td>
<td>TotHr: 22.6</td>
<td>TotHr: 36.6</td>
<td>TotHr: 136.6</td>
<td>TotHr: 207.6</td>
<td>TotHr: 5.0</td>
</tr>
</tbody>
</table>
Memory Provisioned

Memory Utilization in CHTC last week

- cm.chtc.wisc.edu MemoryInUse
  - Now: 3.5M
  - Min: 1.2M
  - Avg: 3.1M
  - Max: 3.6M
  - TotHr: 523.3M

- cm.chtc.wisc.edu MemoryNotInUse
  - Now: 150.5k
  - Min: 426.0
  - Avg: 503.4k
  - Max: 2.4M
  - TotHr: 84.6M
Memory Used vs Provisioned

In `condor_config.local`:

```
STARTD_JOB_EXPRS = $(START_JOB_EXPRS) MemoryUsage
```

Then define `MemoryEfficiency` metric as:

```
[ 
  Name = "MemoryEfficiency";
  Aggregate = "AVG";
  Value = real(MemoryUsage)/Memory*100;
  Requirements = MemoryUsage > 0.0;
  TargetType = "Machine";
]
```
Example: Metrics Per User

[
    Name  = strcat(RemoteUser, "-UserMemoryEfficiency");
    Title = strcat(RemoteUser, " Memory Efficiency");
    Aggregate = "AVG";
    Value  = real(MemoryUsage)/Memory*100;
    Requirements = MemoryUsage > 0.0;
    TargetType = "Machine";
]

Dashboard(s) of useful charts
New Hotness: Grafana

Grafana

- Open Source
- Makes pretty and interactive dashboards from popular backends including Graphite's Carbon, Influxdb, and very recently ElasticSearch
- Easy for individual users to create their own custom persistent graphs and dashboards

```
condor_gangliad -> ganglia -> graphite
```

# gmetad.conf - Forward metrics to Carbon via UDP
carbon_server "mongodbtest.chtc.wisc.edu"
carbon_port 2003
carbon_protocol udp
graphite_prefix "ganglia"
Adding Grafana graph (Graphite)
Adding Grafana graph (Influxdb)
What sort of attributes are avail?

› Lots of good attributes in the collector by default; browse via
  • condor_status -schedd -l,
  • condor_status -submitter -l
  • condor_status -startd -l

› Lots more available via HTCondor "Statistics"
  • Especially in the schedd, collector
  • condor_status --direct --schedd --statistics all:2 <name>
  • Send to the collector via knobs
    STATISTICS_TO_PUBLISH and
    STATISTICS_TO_PUBLISH_LIST
  • All kinds of output, mostly aggregated
  • See TJ or Manual for details
RecentDaemonCoreDutyCycle

- Todd's favorite statistic for watching the health of submit points (schedds) and central manager (collector)

- Measures time not idle

- If goes 98%, your schedd or collector is saturated
Individual Job Monitoring

› Schedd Event Log (rotates)
  • Union of all job event logs for all jobs on a schedd
  • Config Knob: \texttt{EVENT\_LOG} = /some/file

› Audit Log (rotates)
  • Provenance of modifications to any job
  • Config Knob: \texttt{SCHEDD\_AUDIT\_LOG} = /file

› History File (rotates)
  • Schedd history: all job ads that left the queue
    • HISTORY = /file
  • Startd history: all job ads that used a slot
    • STARTD\_HISTORY = /file
  • View with condor\_history (local or remote)
CHTC per user usage for 02/28/16

<table>
<thead>
<tr>
<th>User</th>
<th>Completed Hours</th>
<th>Used Hours</th>
<th>Uniq Job Ids</th>
<th>Request Mem</th>
<th>Used Mem</th>
<th>Max Mem</th>
<th>Request Cpus</th>
<th>ShortJobStarts</th>
<th>All Starts</th>
<th>72 Hour</th>
<th>Min</th>
<th>25%</th>
<th>Median</th>
<th>75%</th>
<th>Max</th>
<th>Mean</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 Totals</td>
<td>257962</td>
<td>289261</td>
<td>348886</td>
<td>1024</td>
<td>51</td>
<td>nan</td>
<td>1</td>
<td>212309</td>
<td>419148</td>
<td>16</td>
<td>00:01</td>
<td>00:03</td>
<td>00:09</td>
<td>00:39</td>
<td>227:32</td>
<td>01:13</td>
</tr>
<tr>
<td><a href="mailto:jlang3@chtc.wisc.edu">jlang3@chtc.wisc.edu</a></td>
<td>44935</td>
<td>44935</td>
<td>1835</td>
<td>4884</td>
<td>3408</td>
<td>4268</td>
<td>1</td>
<td>31</td>
<td>3859</td>
<td>0</td>
<td>00:01</td>
<td>00:05</td>
<td>05:00</td>
<td>24:46</td>
<td>49:04</td>
<td>11:44</td>
</tr>
<tr>
<td><a href="mailto:psbennett@chtc.wisc.edu">psbennett@chtc.wisc.edu</a></td>
<td>34430</td>
<td>34703</td>
<td>4286</td>
<td>4000</td>
<td>660</td>
<td>2048</td>
<td>1</td>
<td>0</td>
<td>4464</td>
<td>0</td>
<td>00:02</td>
<td>01:29</td>
<td>04:21</td>
<td>12:37</td>
<td>42:00</td>
<td>07:46</td>
</tr>
<tr>
<td><a href="mailto:nu_davorka@chtc.wisc.edu">nu_davorka@chtc.wisc.edu</a></td>
<td>31543</td>
<td>31543</td>
<td>2836</td>
<td>2500</td>
<td>3</td>
<td>36</td>
<td>1</td>
<td>0</td>
<td>2836</td>
<td>0</td>
<td>03:23</td>
<td>08:13</td>
<td>11:01</td>
<td>14:06</td>
<td>25:28</td>
<td>11:07</td>
</tr>
<tr>
<td><a href="mailto:xmeng@cs.wisc.edu">xmeng@cs.wisc.edu</a></td>
<td>19106</td>
<td>29891</td>
<td>3107</td>
<td>6144</td>
<td>306</td>
<td>1082</td>
<td>1</td>
<td>1</td>
<td>12390</td>
<td>0</td>
<td>00:01</td>
<td>00:05</td>
<td>00:09</td>
<td>00:50</td>
<td>30:08</td>
<td>02:25</td>
</tr>
<tr>
<td><a href="mailto:nu_haas@chtc.wisc.edu">nu_haas@chtc.wisc.edu</a></td>
<td>16691</td>
<td>16776</td>
<td>27070</td>
<td>1024</td>
<td>10</td>
<td>42</td>
<td>1</td>
<td>0</td>
<td>27070</td>
<td>0</td>
<td>00:01</td>
<td>00:27</td>
<td>00:35</td>
<td>00:42</td>
<td>01:11</td>
<td>00:37</td>
</tr>
<tr>
<td><a href="mailto:gcherg8@chtc.wisc.edu">gcherg8@chtc.wisc.edu</a></td>
<td>15238</td>
<td>15238</td>
<td>23941</td>
<td>1024</td>
<td>28</td>
<td>36</td>
<td>1</td>
<td>4607</td>
<td>23941</td>
<td>0</td>
<td>00:01</td>
<td>00:02</td>
<td>00:23</td>
<td>01:03</td>
<td>05:39</td>
<td>00:47</td>
</tr>
<tr>
<td><a href="mailto:bchen_79@chtc.wisc.edu">bchen_79@chtc.wisc.edu</a></td>
<td>12604</td>
<td>12604</td>
<td>61</td>
<td>25000</td>
<td>18791</td>
<td>19423</td>
<td>8</td>
<td>0</td>
<td>68</td>
<td>0</td>
<td>00:15</td>
<td>04:04</td>
<td>07:13</td>
<td>48:47</td>
<td>72:00</td>
<td>23:10</td>
</tr>
<tr>
<td><a href="mailto:dschultz@icecube.wisc.edu">dschultz@icecube.wisc.edu</a></td>
<td>10034</td>
<td>11027</td>
<td>9235</td>
<td>2000</td>
<td>224</td>
<td>12621</td>
<td>1</td>
<td>1457</td>
<td>9878</td>
<td>0</td>
<td>00:01</td>
<td>00:35</td>
<td>01:06</td>
<td>01:30</td>
<td>19:31</td>
<td>01:18</td>
</tr>
<tr>
<td><a href="mailto:pbrendler@chtc.wisc.edu">pbrendler@chtc.wisc.edu</a></td>
<td>8634</td>
<td>8634</td>
<td>65</td>
<td>40960</td>
<td>19992</td>
<td>19999</td>
<td>15</td>
<td>0</td>
<td>65</td>
<td>0</td>
<td>04:12</td>
<td>07:22</td>
<td>07:56</td>
<td>09:29</td>
<td>31:15</td>
<td>08:51</td>
</tr>
<tr>
<td><a href="mailto:ice3simsr@icecube.wisc.edu">ice3simsr@icecube.wisc.edu</a></td>
<td>7215</td>
<td>7821</td>
<td>7836</td>
<td>2000</td>
<td>226</td>
<td>4209</td>
<td>1</td>
<td>810</td>
<td>7977</td>
<td>0</td>
<td>00:01</td>
<td>00:12</td>
<td>00:57</td>
<td>01:25</td>
<td>19:49</td>
<td>01:05</td>
</tr>
<tr>
<td><a href="mailto:pppmueller@icecube.wisc.edu">pppmueller@icecube.wisc.edu</a></td>
<td>5189</td>
<td>5189</td>
<td>690</td>
<td>8000</td>
<td>5</td>
<td>5043</td>
<td>1</td>
<td>6039</td>
<td>48816</td>
<td>0</td>
<td>00:01</td>
<td>00:03</td>
<td>00:05</td>
<td>00:08</td>
<td>27:09</td>
<td>00:07</td>
</tr>
<tr>
<td><a href="mailto:mleermann@icecube.wisc.edu">mleermann@icecube.wisc.edu</a></td>
<td>5125</td>
<td>7962</td>
<td>2524</td>
<td>2500</td>
<td>1767</td>
<td>1847</td>
<td>1</td>
<td>2</td>
<td>3217</td>
<td>0</td>
<td>00:01</td>
<td>00:07</td>
<td>02:15</td>
<td>04:27</td>
<td>11:27</td>
<td>02:28</td>
</tr>
<tr>
<td>osg <a href="mailto:osg@hep.wisc.edu">osg@hep.wisc.edu</a></td>
<td>4890</td>
<td>5647</td>
<td>887</td>
<td>2000</td>
<td>567</td>
<td>13405</td>
<td>1</td>
<td>4</td>
<td>943</td>
<td>0</td>
<td>00:02</td>
<td>00:18</td>
<td>03:53</td>
<td>08:09</td>
<td>22:57</td>
<td>05:32</td>
</tr>
<tr>
<td><a href="mailto:wguan@chtc.wisc.edu">wguan@chtc.wisc.edu</a></td>
<td>4182</td>
<td>4202</td>
<td>173858</td>
<td>1</td>
<td>51</td>
<td>16842</td>
<td>1</td>
<td>158393</td>
<td>173858</td>
<td>0</td>
<td>00:01</td>
<td>00:01</td>
<td>00:01</td>
<td>00:02</td>
<td>10:51</td>
<td>00:10</td>
</tr>
<tr>
<td><a href="mailto:elims@icecube.wisc.edu">elims@icecube.wisc.edu</a></td>
<td>3853</td>
<td>5614</td>
<td>1625</td>
<td>7000</td>
<td>1774</td>
<td>1810</td>
<td>1</td>
<td>122</td>
<td>4066</td>
<td>0</td>
<td>00:01</td>
<td>00:09</td>
<td>01:08</td>
<td>02:29</td>
<td>05:19</td>
<td>01:25</td>
</tr>
<tr>
<td><a href="mailto:mayeshiba@chtc.wisc.edu">mayeshiba@chtc.wisc.edu</a></td>
<td>3834</td>
<td>3834</td>
<td>4</td>
<td>20000</td>
<td>6662</td>
<td>10350</td>
<td>16</td>
<td>0</td>
<td>5</td>
<td>0</td>
<td>03:41</td>
<td>15:56</td>
<td>72:00</td>
<td>72:00</td>
<td>72:00</td>
<td>47:07</td>
</tr>
</tbody>
</table>
Upcoming

› condor_gangliad → condor_metricd
   • Send aggregated metrics to Ganglia
   • Write out aggregated metrics to rotating JSON files
   • Send aggregated metrics to Graphite / Influx

› A new "HTCondor View" tool
   • Some basic utilization graphs out-of-the-box
Check out Fifemon!

"Comprehensive grid monitoring with Fifemon has improved resource utilization, job throughput, and computing visibility at Fermilab"

- Probes, dashboards, and docs at: https://github.com/fifemon

- Fifemon Overview talk from HTCondor Week 2016:
  https://research.cs.wisc.edu/htcondor/HTCondorWeek2016/presentations/ThuRetzke_Fifemon.pdf
Thank you!