HTCondor at Syracuse University – Building a Resource Utilization Strategy

Eric Sedore
Associate CIO
HTCondor Week 2017

Syracuse University
Information Technology Services (ITS)
Research Computing Philosophy @ Syracuse

- Good to advance research, best to transform research (though transformation is not always related to scale)
- Entrepreneurial approach to collaboration and ideas
- Computing resources are only one part of supporting research
- **Strive to use computational resources at 100% utilization, 100% of the time**
- Computational resources must support multiple academic areas
Computational Resources @ Syracuse

- Academic Virtual Hosting Environment (AVHE) – private cloud
  - 1000 cores, 25TB of memory
  - Individual VMs (students, faculty, staff), small clusters
  - 2 PB of storage (NFS, SMB, DAS per VM), multiple performance tiers
- OrangeGrid – high throughput computing pool
  - scavenged desktop grid, 13,000 cores, 17TB of memory
- Crush – compute focused cloud
  - Coupled with the AVHE to provide HPC and HTC environments
  - Made up of heterogeneous hardware, different areas within Crush are focused on different needs (high IO, latency/bandwidth, high memory requirements…)
  - 12,000 cores (24,000 slots with HT), 50 TB of memory
- SUrge – GPU focused compute cloud
  - 240 commodity NVidia GPUs
  - Individual VMs / nodes scheduled via HTCondor
**Resource deployment**

<table>
<thead>
<tr>
<th>Researchers can utilize existing “standard” environments or build a unique environment</th>
</tr>
</thead>
<tbody>
<tr>
<td>“Virtual Clusters” network, data, scheduling</td>
</tr>
<tr>
<td>Tools for deploying and managing 10,000+ VM’s in 4 virtual environments (KVM, Hyper-V, vSphere, VirtualBox)</td>
</tr>
<tr>
<td>Virtualize everything – systems for building nodes, no affiliation, everything loosely coupled (i.e. researchers never touch bare metal)</td>
</tr>
</tbody>
</table>

*Syracuse University*

*Information Technology Services (ITS)*
Allocation of resources

Syracuse Researchers

Public Science (E@H...)

Open Science Grid (OSG)
Hybrid and Opportunistic
# What resources should Syracuse provide?

<table>
<thead>
<tr>
<th>Resource Type</th>
<th>Description</th>
<th>Provided By</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>“Large scale / Specialized”</td>
<td>accomplished in national infrastructure</td>
<td>National Resources</td>
<td>Not enough need (today) to invest at this level</td>
</tr>
<tr>
<td></td>
<td>10,000+ cores, 100’s of TB’s of memory, PB’s of data</td>
<td></td>
<td></td>
</tr>
<tr>
<td>“Medium scale”</td>
<td>accomplished in clusters</td>
<td>Syracuse</td>
<td>Utilization at 85+% (from an IT Perspective)</td>
</tr>
<tr>
<td></td>
<td>1000’s of cores, 10’s of TB’s of memory, TB’s of data</td>
<td></td>
<td></td>
</tr>
<tr>
<td>“Small / Medium scale”</td>
<td>accomplished in the cloud</td>
<td>Syracuse</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1-200 cores, 1GB-2TB of memory, TB’s of data</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Individual virtual machines to small clusters</td>
<td></td>
<td></td>
</tr>
<tr>
<td>“Small scale”</td>
<td>accomplished on desktops/laptops</td>
<td>Syracuse</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1-4 cores, 1-16GB of memory, GB’s of data</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Core Elements

• HTCondor
  • Primary tool for resource scheduling – everything (almost) else is a pain!
  • Node advertising capabilities
  • Simplicity of addition/removal of nodes (part its scavenging roots)
  • Flexibility – small simple environments to larger more complicated environments

• Virtualization (KVM, Hyper-V, vSphere, VirtualBox)
  • Abstraction – shim allows us to easily reallocation resources, including networking and storage
  • Flexibility – easy to run multiple kinds of workload (Windows/Linux)

• In-house coding / scripting – primarily in management / deployment – interacting with hypervisors
Pain Points

- VM Management – we have ~20 VM environments within Crush alone
  - Versioning, automation, best of breed VM / monolith VM
  - What do we need? Singularity / Docker When do we need it? Now!
- Staff Expertise
  - Complexity, staff resources, single person dependencies - systems focused on being operated by a fraction of a staff member
  - Nuance/elegance is lost, often the “right way” is set aside in the necessity to move on to the next
Musings on Our HTCondor Experience

• Law of unintended consequences is alive and well – changes always have impact
• There is a knob for everything…
• Logging is spectacular, deep, voluminous - “a blessing and a curse”
• You can have multiple versions of HTCondor components in your environment, but anecdotally you will occasionally find “odd” interactions