Outline

- Introduction
- PBS 13 Design, Complexities and Challenges
- PBS 18 Design, Differences and Simplifications
- PBS 18 Specifics
- ML Software Stack
- Docker Challenges and Solutions
Introduction

GE HPC Focus:

• Scientist Focus on Science not HPC
• User Experience drives HPC design
• Simulations for Multiple GE Businesses
• Altair – GE relationship and product enhancements
• PBS Introduction and evolution
System Overview – PBS 13
PBS 13 Complexities/Challenges

• Multiple CPU/Core/Memory Architectures in Cray and Linux
• Single JobID for many PBS Complexes (ex. qstat 123456.server1 @server2)
• Job Dependencies*
• Peering Delay and Challenges
• Multiple Complexes for Multiple Policies, Multiple PBS Branches
• **Hardware Choice to occur at Run Time not Submit Time**
• Custom Hook to Re-write –l select after job peered to Execution Cluster

* Provided by Altair Development
System Overview – Consolidation/Simplification

- PBS 18.2 Server
- No Custom Scheduling Logic
- Visualization
- Single Server
- Platform Specific Queue
- ML - GPU
PBS 18 Improvements

- 4 PBS Servers/Complexes to 1 with one code branch.
- All HPC now in single pane of glass and single PBS complex
- Dispatch speed with no Peering
- Network Topology Placement Sets by Scheduler
- **Multiple scheduling polices applied to different compute types**
- **Multiple Hardware Choices via Altair provided customizations**
Software Overview

- Cray CLE 5
- RHEL 7
- CFD 3rd Party Applications
- FEA 3rd Party Applications
- Docker/Nvidia-docker
- Docker Trusted Registry

- Cuda 9&10
- Python 2&3
- Ubuntu 16.0.4
- Tensor/Keras/Caffe
- TensorBoard/Jupyter Notebook
PBS 18 MultiSched Partitions

<table>
<thead>
<tr>
<th>Cray</th>
<th>Linux</th>
<th>Visualization</th>
<th>ML-GPU</th>
</tr>
</thead>
</table>
| • Backfill  
• Fairshare  
• Group Limits  
• Topology Placement Sets  
• Multiple CPU, Memory Architectures | • Backfill  
• Fairshare  
• Topology Placement Sets  
• Multiple CPU, Memory and System Architectures | • GPU Consumption  
• GLX Visualization  
• FIFO  
• Job Limit per User | • GPU Consumption  
• Container Hook  
• Cgroup Hook |

© 2019 General Electric Company - All rights reserved
### Scheduler, Queue, and Node Connection

<table>
<thead>
<tr>
<th>Scheduler</th>
<th>Queue and Node</th>
</tr>
</thead>
<tbody>
<tr>
<td>create sched linux</td>
<td>set queue linux partition = p_linux</td>
</tr>
<tr>
<td>set sched linux partition = p_linux</td>
<td>set queue cray partition = p_cray</td>
</tr>
<tr>
<td></td>
<td>set queue visual partition = p_visual</td>
</tr>
<tr>
<td></td>
<td>set queue mlaas partition = p_mlaas</td>
</tr>
<tr>
<td>create sched cray</td>
<td>set node linux_node partition = p_linux</td>
</tr>
<tr>
<td>set sched cray partition = p_cray</td>
<td>set node clogin78_2111 partition = p_linux</td>
</tr>
<tr>
<td></td>
<td>set node clogin78_1856 partition = p_cray</td>
</tr>
<tr>
<td>create sched visual</td>
<td>set node visual_node partition = p_visual</td>
</tr>
<tr>
<td>set sched visual partition = p_visual</td>
<td>set node mlaas_node partition = p_mlaas</td>
</tr>
<tr>
<td>create sched mlaas</td>
<td></td>
</tr>
<tr>
<td>set sched mlaas partition = p_mlaas</td>
<td></td>
</tr>
</tbody>
</table>
Docker Security Problems

- Root in container
- Root access to storage volumes
- Root can mount other volumes
- Export Controlled exposure
- Can use Docker User Namespace. `docker run` can disable it
PBS Pro 18 Docker Features

- PBS Hook places user in container as UID of job owner
- User **NOT** in docker group
- Configurable storage mounts to pass through
- Environmental variable passing
- Environmental variable stripping
- TCP Port Forwarding into Container*
Hook/Node Setup

[root@hooks]# more PBS_hpc_container.CF
{
   "docker_cmd": "/usr/bin/docker",
   "nvidia_docker_cmd": "/usr/bin/nvidia-docker",
   "remove_env_keys": [],
   "mount_paths": ["/data","/home","/model","/projects","/scratch"],
   "port_ranges": ["8000-8500"]
}

Node Configuration:

set node mlaas_node
resources_available.allows_container = True

Not in Group

schmid@mlaas_node:$ docker run
dtr.server.com/schmid/ubuntu1604:keras-tensorflow-1.12-py3
docker: Got permission denied while trying to connect to the Docker daemon socket at unix:///var/run/docker.sock:
bash-4.2$ more container.sh

#PBS -q mlaas
#PBS -l select=1:ncpus=1:ngpus=1:ectag=us
#PBS -l place=pack
#PBS -e std_err.txt
#PBS -o std_out.txt

#PBS -I

#PBS -v
CONTAINER_IMAGE=dtr.server.com/schmid/ubuntu1604:keras-tensorflow-1.12-py3

-bash-4.2$ qsub container.sh
qsub: waiting for job 123456 to start
qsub: job 123456 ready

schmid@1cf418c89a4b:/workspace$ id
uid=6907 gid=99999999 groups=99999999

schmid@fe0ab2a21fe7:/workspace$
### Cgroup Limits

<table>
<thead>
<tr>
<th>Physical Host</th>
<th>Container</th>
</tr>
</thead>
<tbody>
<tr>
<td>root@mlaas_node:~# nvidia-smi</td>
<td>schmid@fe0ab2a21fe7:/workspace$ nvidia-smi</td>
</tr>
<tr>
<td>Mon Apr 8 20:38:57 2019</td>
<td>Tue Apr 9 00:38:24 2019</td>
</tr>
<tr>
<td></td>
<td>+-----------------------------------+</td>
</tr>
<tr>
<td></td>
<td>NVIDIA-SMI 410.48 Driver Version: 410.48</td>
</tr>
<tr>
<td></td>
<td>+-----------------------------------+</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>GPU Name Persistence-M</td>
</tr>
<tr>
<td></td>
<td>Fan Temp Perf Pwr:Usage/Cap</td>
</tr>
<tr>
<td></td>
<td>0 Tesla V100-SXM2... Off</td>
</tr>
<tr>
<td></td>
<td>1 Tesla V100-SXM2... Off</td>
</tr>
<tr>
<td></td>
<td>+-----------------------------------+ +-----------------------------------+</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Conclusions

• PBS 18 simplifies complex environment for users
• PBS 18 makes multi CPU architecture cray look the same
• PBS 18 single code base provides single feature set
• PBS 18 handles large security problems with Docker
• PBS 18 Hooks and Ubuntu support make it a fit for ML workloads/containers
• GPU restrictions via PBS cgroup integration