Dynamic Scheduling Strategy of HTCondor at IHEP

Shi, Jingyan (shijy@ihep.ac.cn)
On behalf of scheduling group of Computing Center, IHEP
Content

1. IHEP Cluster Introduction
2. Scheduling Strategy to HTCondor
3. Implementation and Deployment
4. Summary and Future Plan
HEP Experiments at IHEP

IHEP: Institute of High Energy Physics

BESIII (Beijing Spectrometer III at BEPCII)
100TB raw data/year *10 years

DYB (Daya Bay Reactor Neutrino Experiment)
200TB/year *9 years

JUNO (Jiangmen Underground Neutrino Observatory)
2PB/year *30 years

LHAASO
Large High Altitude Air Shower Observatory
1.2PB/year *10 years

HXMT
Hard X-Ray Moderate Telescope
HTCondor Cluster Resource

- Local cluster
  - ~10,500 CPU cores
  - Most are single core, series job slots
  - Managed by PBS for 10 years
    - Bottleneck: low scheduling performance
      - Large amount of jobs: 20,000 jobs in queues
      - Large scale: over 10,000 job slots
  - Migrated to HTCondor step by step with risk control
    - Jan, 2015: ~ 1,100 CPU cores
    - May, 2016: ~ 3,500 CPU cores
    - Dec, 2016: ~ 11,000 CPU cores
Content

1. IHEP Cluster Introduction
2. Scheduling Strategy to HTCondor
3. Implementation and Deployment
4. Summary and Future Plan
Resource Managed by PBS Cluster

- Several experiments supported
  - BES, Daya Bay, Juno, Lhaaso, HXMT etc.
  - Resources are funded and dedicated for different experiments
  - No resource sharing among the experiments
  - 55 jobs queues with group permission limits configured

- Low resource utility
  - Coexistence of busy queues and free resources
Busy Queue and Free Resource at PBS

BES Resource – Utility: 65.50%

DYW Resource – Utility: 8.75%
Basic Idea of the HTCondor Scheduling Strategy

- **Resource sharing**
  - Break the resource separation
  - Busy exp. can take more resources from that of the free exp.

- **Fairness guarantee**
  - Peak computing requirements from different exp. usually happened at different time periods
  - Jobs from free exp. have higher priority than the jobs from busy exp.
  - The more resources the exp. shares, the more its jobs can be scheduled
Resource Sharing at HTCondor

- Based on job slots (mainly CPU cores)
- As first step, part of resources are contributed to be shared
- Some dedicated resources are kept by experiments own
  - Only run jobs from owner exp.
- Sharing resource pool
  - Sharing resources contributed by all experiments
  - Sharing slots can be dispatched to all jobs
  - At least 20% slots of each exp. are shared
  - encourage exp. to share more resources

HTCondor Cluster Sharing Policy

<table>
<thead>
<tr>
<th>Experiment</th>
<th>Slots</th>
</tr>
</thead>
<tbody>
<tr>
<td>JUNO</td>
<td>888</td>
</tr>
<tr>
<td>DYB</td>
<td>1188</td>
</tr>
<tr>
<td>CMS</td>
<td>544</td>
</tr>
<tr>
<td>ATLAS</td>
<td>576</td>
</tr>
</tbody>
</table>

The dedicated and sharing slots of different groups
Resource Sharing with HTCondor

The dedicated, sharing and max allocable slots for each exp.

( In May, 2016 )
Fairness and priority

- Scheduling preference
  - Jobs are preferred to run at dedicated slots owned by exp.
  - The shared slots are kept for busy experiments

- Group quota
  - Define linux group for each exp.
  - The initial group quota is set to the amount of real resources from exp.
  - Group quota can be exceeded if there are free slots in the sharing pool

- Group priority and User priority
  - Group priority is correlated to the group quota and the group slots occupancy
  - User priority is effective within the same group users
Content

1. IHEP Cluster Introduction
2. Scheduling Strategy to HTCondor
3. Implementation and Deployment
4. Summary and Future Plan
Current HTCondor Status

- **Architecture**
  - 28 submitting nodes
  - 2 scheduler machine (local cluster, virtual cluster)
  - 2 central manager (local cluster, virtual cluster)
  - ~ 10,000 physical CPU cores + an elastic number of virtual slots

- **Jobs**
  - Avg 100,000 jobs/day;
  - 60,000 jobs in queue at peak time
  - Serial single-core jobs
Job Monitoring

- Queuing and running statistics
  - The overall clusters
  - Each group/experiment
- The dedicated and sharing resource statistics
- Nagios and Ganglia
Central Controller

- Control of groups, users and resources
  - All information is collected into the Central Database
  - Necessary information is published to the relative services
Error Detection and Recovery

- Workers’ health status are collected and report to Central Controller
- Workers’ attributes are self-updated automatically through the information published by Central Database
- No job will be scheduled to error worker
Global Accounting

- Detailed accounting to each group and each user
  - Accounting the contribution to other exp.
  - Accounting the extra resources occupied from other exp.
- Weighting slots with slow/fast CPU, Memory, Disk, etc.
The Toolkit for user: hep_job

- **Motivation**
  - Smooth migration from PBS to HTCondor for users
  - Simplify users’ work
  - Help to achieve our scheduling strategy

- **Implementation**
  - Base on python API of HTCondor
  - Integrated with IHEP computing platform
    - Server name, group name
  - Several Jobs template according the experiments requirements
Put all together

Central Controller System

Hepjob

commands

HTCondor

negotiator

collector

sCHEDD

startd

Accounting

Monitoring

nagios

/web page

http

interface to cc accounting

Sharing Policy

Dynamic Configuration

user group

jobs
Resource Utility Improvement

The HTCondor overall resource utility last month: ~87%

- The typical resource utility without resource sharing: 50% - 60%
- There is a significant improvement with the resource sharing policy
Content

1. IHEP Cluster Introduction
2. Scheduling Strategy to HTCondor
3. Implementation and Deployment
4. Summary and Future Plan
Summary and Future work

**Summary**
- The resource utility is significantly improved by the resource sharing strategy
- Central controller helps to provide stable computing service

**Future work**
- Resource sharing ratio will be tuned dynamically according to the overloads of each group
  - The integration of Job Monitoring and Central Controller
- Fine grain accounting system need to be developed
Thank you!

Question?