Experience on HTCondor batch system for HEP and other research fields at KISTI-GSDC

Sang Un Ahn, Sangwook Bae, Amol Jaikar, Jin Kim, Byungyun Kong, Ilyeon Yeo

13 October 2016
@CHEP2016, San Francisco
Introduction

- GSDC Job Profiling for 2015 showed that resource utilization of global services varies from 65%(B) to 90%(A) while that of local services is below 25%(D) (worst 0.7%(C))

- Consistent job throughput from Grid consumed resources effectively (A, B)

- Chaotic job activities of local users led to low resource utilization (C, D)

Goal

- To improve resource utilization by allocating idle resources to where demanded (dynamic resource management)

- Constraints:
  - Resources are dedicated to each experiment based on MoU and being audited regularly on physical allocation
    - Dedicated resources are preferred by users since they can secure resources when crowded, esp. before conferences

- Batch systems are different: Torque for EMI (ALICE and Belle II), HTCondor for OSG (CMS) and LDG (LIGO)
  - Queues cannot be shared between Torque and HTCondor
Goal #2

- **To have one batch system for all: HTCondor**
  - Already have operation experience
  - Actively developed and widely used nowadays

- **Some issues with Torque**
  - Instability issue with Maui scheduler – required to re-start occasionally
    - Workaround: setting a cron job to check the health of maui service and re-start if required
  - Any change in “nodes” requires Torque to be re-started
    - Not applicable to a huge pool (more than thousands)

- **Constraints:**
  - HTCondor is incompatible with CREAM-CE
Procedure

- **Step 1:**
  - **Set-up a HTCondor pool for all Local farms**
    - In addition to CMS and LIGO, there are a few more experiments we support: Genome (HTCondor), RENO and TEM (Torque)

- **Step 2:**
  - **Set-up a bigger pool including Grid farms**
    - Replacing CREAM-CE by HTCondor-CE
    - Or making some modification on communicating part between CREAM-CE and Batch system
Obstacles

- **Step 1:**
  - We have some experience on HTCondor but, resource dedication does not require complicated scheduling policy i.e. we have few knowledges on HTCondor configuration

- **Step 2:**
  - CREAM-CE is not easily replaced by HTCondor-CE since EMI middleware does not support HTCondor-CE
  - Modification on CREAM-CE requires additional man-power
Requirements

- Dynamic Resource Allocation within a HTCondor Pool
  - Resources written in MoU should be guaranteed when users demand

- Separate User Interfaces
  - Sharing UI among different user groups may cause some issues:
    - Compilation on UI before job submission may affect overall performance of the machine badly
    - Exposure of mount points for experiment data or scratch may have potential security glitches even though the access by others is not allowed

- Remote Submission
  - Independent Schedd machines managing shared queues

- High Availability
  - Schedd
  - Central Manager: Collector, Negotiator
Test-bed Setup

UID_DOMAIN = example.com
CENTRAL_MANAGER1 = cm1.example.com
CENTRAL_MANAGER2 = cm2.example.com
CONDOR_HOST = $(CENTRAL_MANAGER1), $(CENTRAL_MANAGER2)
HA Daemons

SCHEDD_NAME = had_schedd@
MASTER_HA_LIST = SCHEDD
HA_LOCK_URL = file:/var/lib/condor/spool (NFS exported)
VALID_SPOOL_FILES = $(VALID_SPOOL_FILES) SCHEDD.lock

DAEMON_LIST = MASTER, COLLECTOR, NEGOTIATOR, HAD, REPLICATION
STATE_FILE = $(SPOOL)/Accountantnew.log
Accounting Group

- Define Accounting Group for sharing resources among several user groups
  - negotiator configuration
    - GROUP_NAMES = group_alice, group_cms, group_ligo, group_reno ...

- Place quota and allow to exceed the limit
  - negotiator configuration
    - GROUP_QUOTA_group_alice, GROUP_QUOTA_group_cms, ...
    - GROUP_ACCEPT_SURPLUS = True

- Preemption enabled to guarantee the quota
  - negotiator configuration
    - NEGOTIATOR_CONSIDER_PREEMPTION = True
    - PREEMPTION_REQUIREMENTS = $(PREEMPTION_REQUIREMENTS) && ((SubmitterGroupResourcesInUse < SubmitterGroupQuota) && (RemoteGroupResourcesInUse > RemoteGroupQuota)) || (SubmitterGroup =?= RemoteGroup)

DEMO

- No activity from group_alice, then everybody freely shares the resources as much as they want
- A group_alice user login and check the status and it shows that only 12 slots are available

```
[alice_user1@sched1 ~]$ condor_status -format "%s" AccountingGroup -format " | %s" State -format " | %sWn" Activity -constraint 'True' | sort | uniq -c | awk '{print $0; t += $1} END { printf("%7d total\n",t) }'
10 group_belle.belle_user1@example.com | Claimed | Busy
10 group_cms.cms_user1@example.com | Claimed | Busy
10 group_genome.genome_user1@example.com | Claimed | Busy
20 group_ligo.ligo_user1@example.com | Claimed | Busy
12 | Unclaimed | Idle
72 total
```

- Regardless the number of slots available, alice_user1 claims 40 slots

```
[alice_user1@sched1 ~]$ condor_submit job
Submitting job(s).................................
40 job(s) submitted to cluster 32.
```

Qouta

```
GROUP_QUOTA_group_alice = 40
GROUP_QUOTA_group_cms = 12
GROUP_QUOTA_group_ligo = 8
GROUP_QUOTA_group_belle = 4
GROUP_QUOTA_group_reno = 4
GROUP_QUOTA_group_genome = 4
```

Job Description File

```
... accounting_group = group_<exp>
accounting_group_user = <exp>_user1
...
```

- ligo user gets cms unused slots
- Preempted jobs goes to idle state
Remarks

- Delicate allocation policy is required when preemption enforced
  - Treatment on preempted jobs: Kill or Suspend?
  - Checkpoint would help preempted jobs resumed in other places
    - Checkpoint is known to only work with Standard Universe
    - But there is a way that it works with Vanilla Universe:
      http://www.uucam.ac.uk/scientific/camgrid/technical/blcr

- Fairshare based on user priority affects in a way that preemption does not work
  - Setting PRIORITY_HALFLIFE to high enough makes user priority (effectively) constant

- By default, remote job submission requires stronger security
  - Password, FS (or FS_remote) are not working
  - GSI, Kerberos methods has to be setup
To do

- Fine tuning on negotiator configuration is required to deploy the test-bed setup in production level
  - Should come with delicate resource policy

- Setup and test a Dedicated Scheduler for jobs submitted with Parallel Universe in order to allow a job to be run on two or more physical machines at the same time
  - 30% of Local LIGO user jobs run with MPI

- HTCondor-CE study for Grid
Conclusions

- We setup a test-bed with HTCondor to achieve the followings:
  - Dynamic resource allocation
  - One batch system for all

- Simple and quick setup showed that we could do what we want

- Just took one step towards the HTCondor world
  - Lots of things to study
References

- HTCondor Manual v8.4.X

- HTCondor: How To Admin Recipes
  - https://htcondor-wiki.cs.wisc.edu/index.cgi/wiki?p=HowToAdminRecipes

- Maintaining Accounting Group Quotas with Preemption Policy

- Checkpointing Vanilla Jobs with BLCR
  - http://www.ucp.cam.ac.uk/scientific/camgrid/technical/blcr