The Scheduling Strategy and Experience of IHEP HTCondor Cluster

Shi, Jingyan (shijy@ihep.ac.cn)

On behalf of scheduling group of

Computing Center, IHEP

Outline

1. Migration to HTCondor
2. Scheduling Policy to HTCondor
3. Works Designed and Developed
4. Problems We Met
5. Summary and Future Work

The Migration to HTCondor

- Motivation
  - PBS had been used at IHEP for more than 10 years
  - Limited Scalability and growing resources and users
    - ~10,000+ job slots and 20,000+ jobs: Performance bottleneck
- Migration to HTCondor: Better performance and active community
- Migration step by step with risk control
  - Jan, 2015 : ~ 1,100 CPU cores
  - May, 2016: ~ 3,500 CPU cores
  - Dec, 2016: ~ 11,000 CPU cores

Current Status

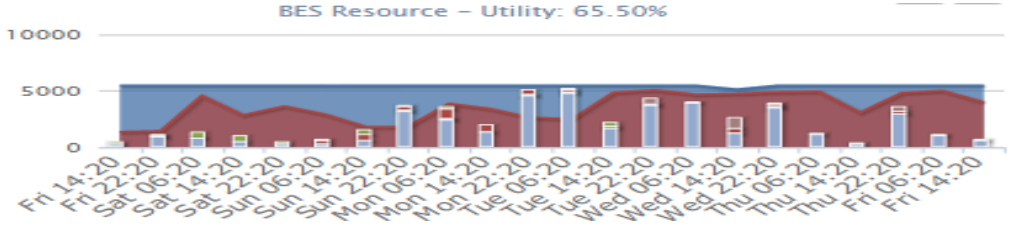
- Architecture
  - 28 submitting nodes
  - 2 scheduler machine (local cluster, virtual cluster)
  - 2 central manager (local cluster, virtual cluster)
  - ~ 10,000 physical CPU cores + an elastic number of virtual slots
- Jobs
  - Avg 100,000 jobs/day;
  - 60,000 jobs in queue at peak time
  - Serial and single-core jobs

Outline

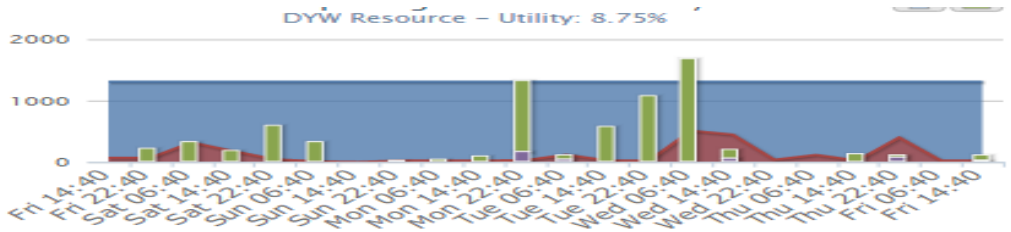
1. Migration to HTCondor
2. **Scheduling Policy to HTCondor**
3. Works Designed and Developed
4. Problems We Met
5. Summary and Future Work

Resource Divided at PBS Cluster

- Several HEP experiments supported
  - BES, Daya Bay, Juno, Lhaaso, HXMT etc.
  - Resources are funded and dedicated for different experiments
  - No resource sharing among experiments
  - 55 jobs queues with group permission limits set at PBS
- Low resource utility
  - Coexistence busy queues and free resources



Busy Group



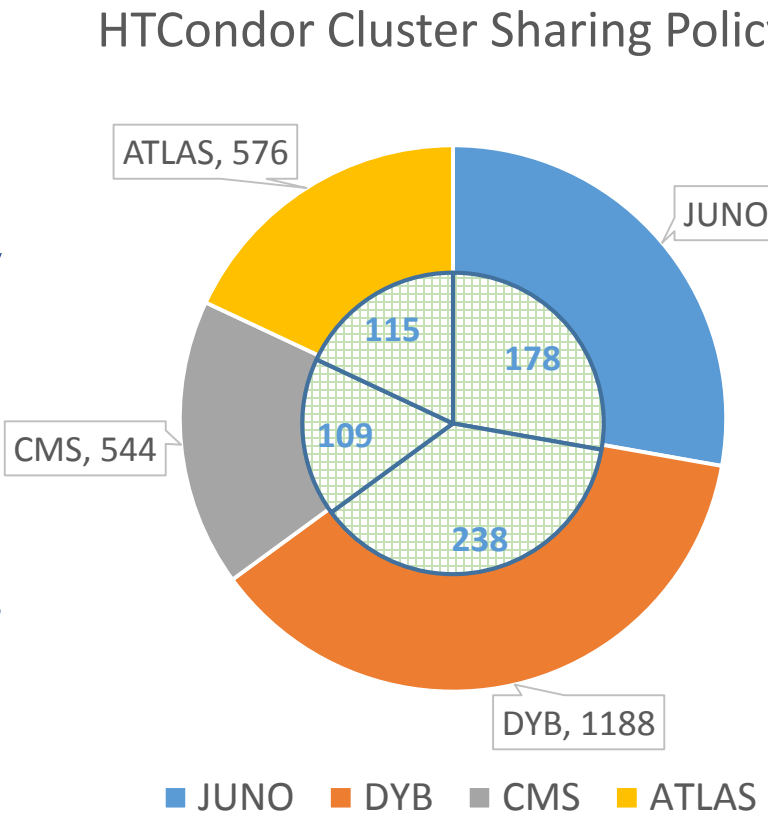
Idle Group

Scheduling Strategy at HTCondor Cluster

- Resource sharing
  - Break the resource separation
  - Busy groups can occupy more resource from the resource of idle groups
- Fairness guarantee
  - Peak computing requirements from different experiment usually happened at different time period
  - Jobs from idle group have high priority
  - The more resource the experiment contribute to share, the more its jobs can be scheduled to run

Resource Sharing at HTCondor

- Based on job slots (mainly CPU cores)
- As a first step, resources are partially shared
- Some exclusive resources are kept by experiments own
  - Only run jobs from the resource owner
- Sharing resource pool
  - Resource contributed by all experiments
  - Slot can accept for jobs from all experiments
  - At least 20% slots are shared by each experiment
  - encourage experiments to share more resources



The exclusive and shared slots of different groups

Fairness and Priority

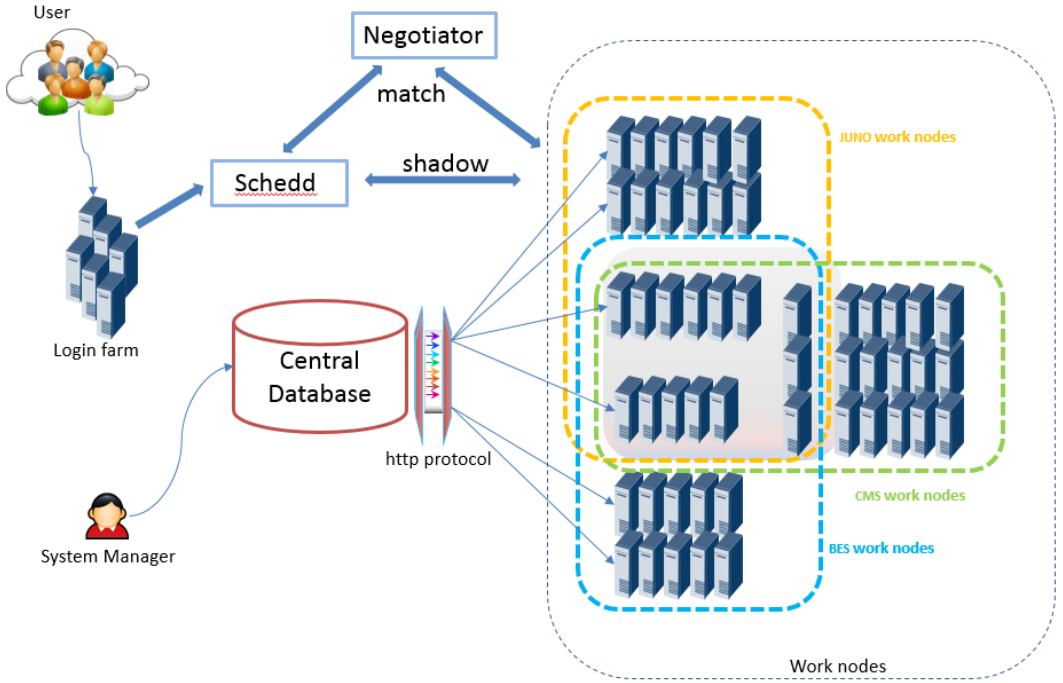
- Scheduling preference
  - Jobs prefers to run on exclusive slots of its own experiment
  - The shared slots are kept for busy experiments
- Experiemnt quota
  - Users from the experiment are in the same linux group
  - The initial group quota is set to the amount of real resources from experiments
  - The quota can be exceeded if there are idle 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 inside same group users

Outline

1. Migration to HTCondor
2. Scheduling Policy to HTCondor
3. **Works Designed and Developed**
4. Problems We Met
5. Summary and Future Work

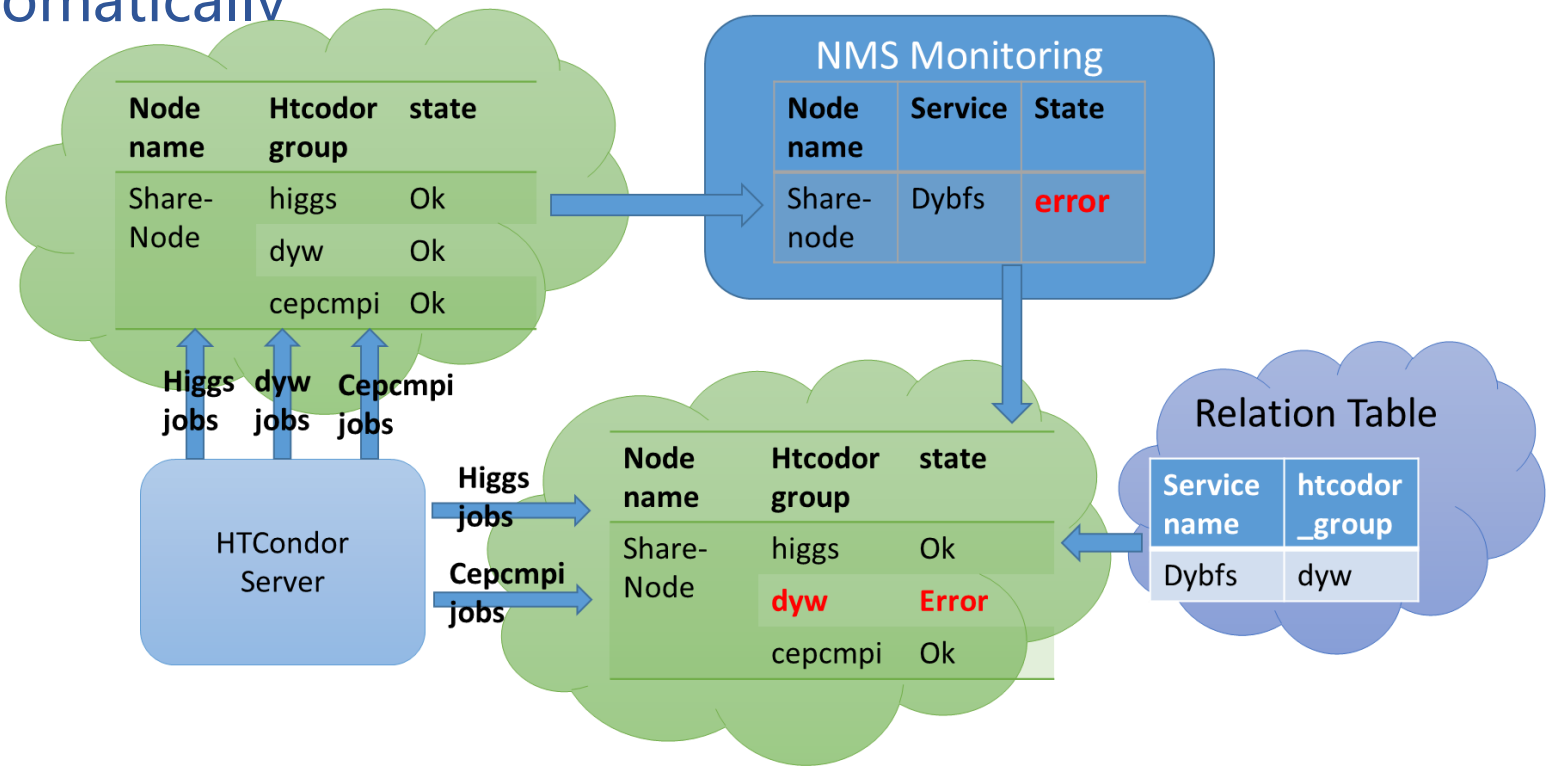
Central Controller

- The central control of groups, users and work nodes
  - All information is collected and saved into Central Database
  - Necessary information is updated and published to relative services
  - Work nodes update its configuration via httpd periodically



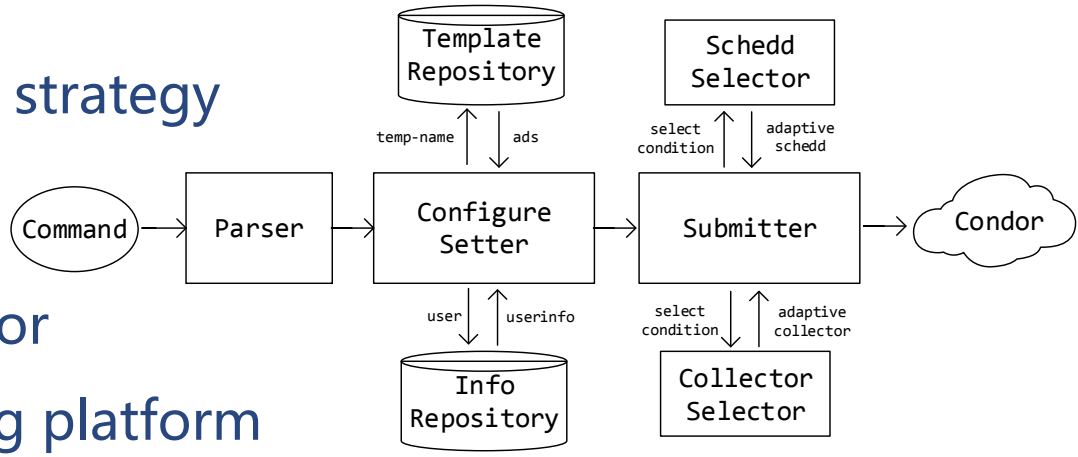
Error Detection and Recovery

- Health status of all workers are collected from monitoring system and saved into Central Database
- Central controller updates work noworkers’ attributes automatically



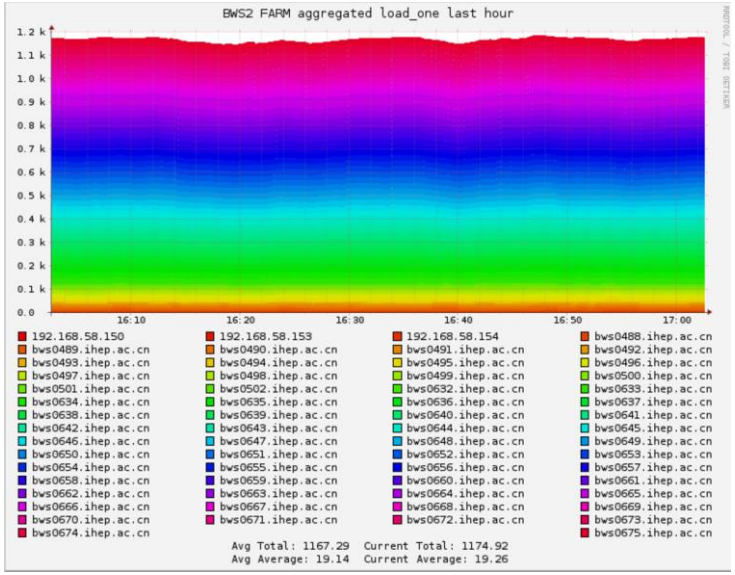
The Toolkit: hep_job

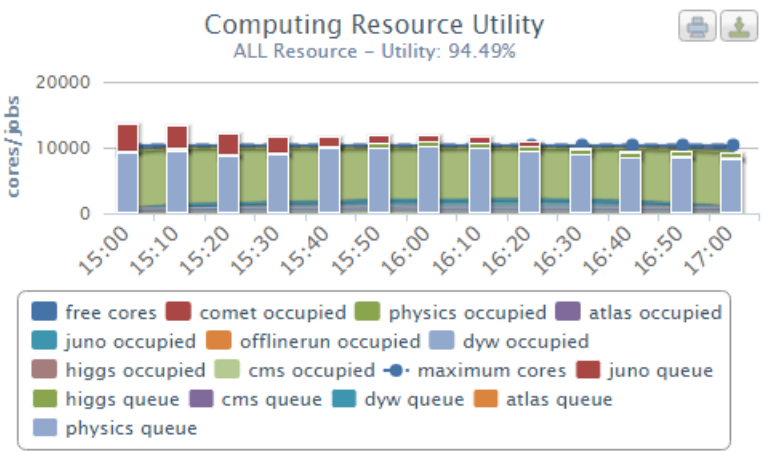
- Motivation
  - Smooth migration from PBS to HTCondor for users
  - Simplify users' work
  - Help to achieve our scheduling strategy
- Implementation
  - Base on python API of HTCondor
  - Integrated with IHEP computing platform
  - Server name, group name
  - Several Jobs template according the experiments requirements

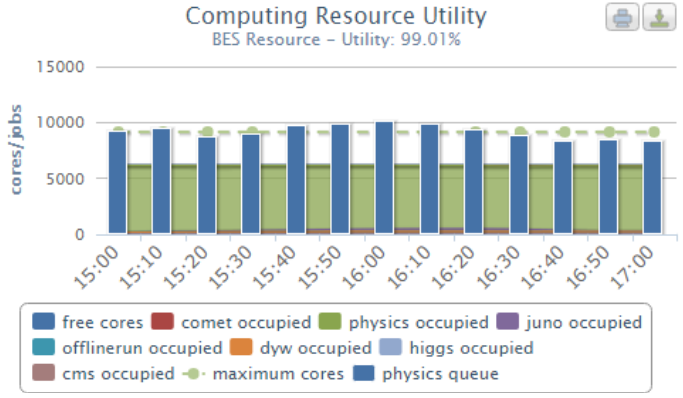


Job Monitoring

- Queueing and running statistics
  - The overall clusters
  - Each group/experiment
- The exclusive and sharing resource statistics
- Nagios and Ganglia





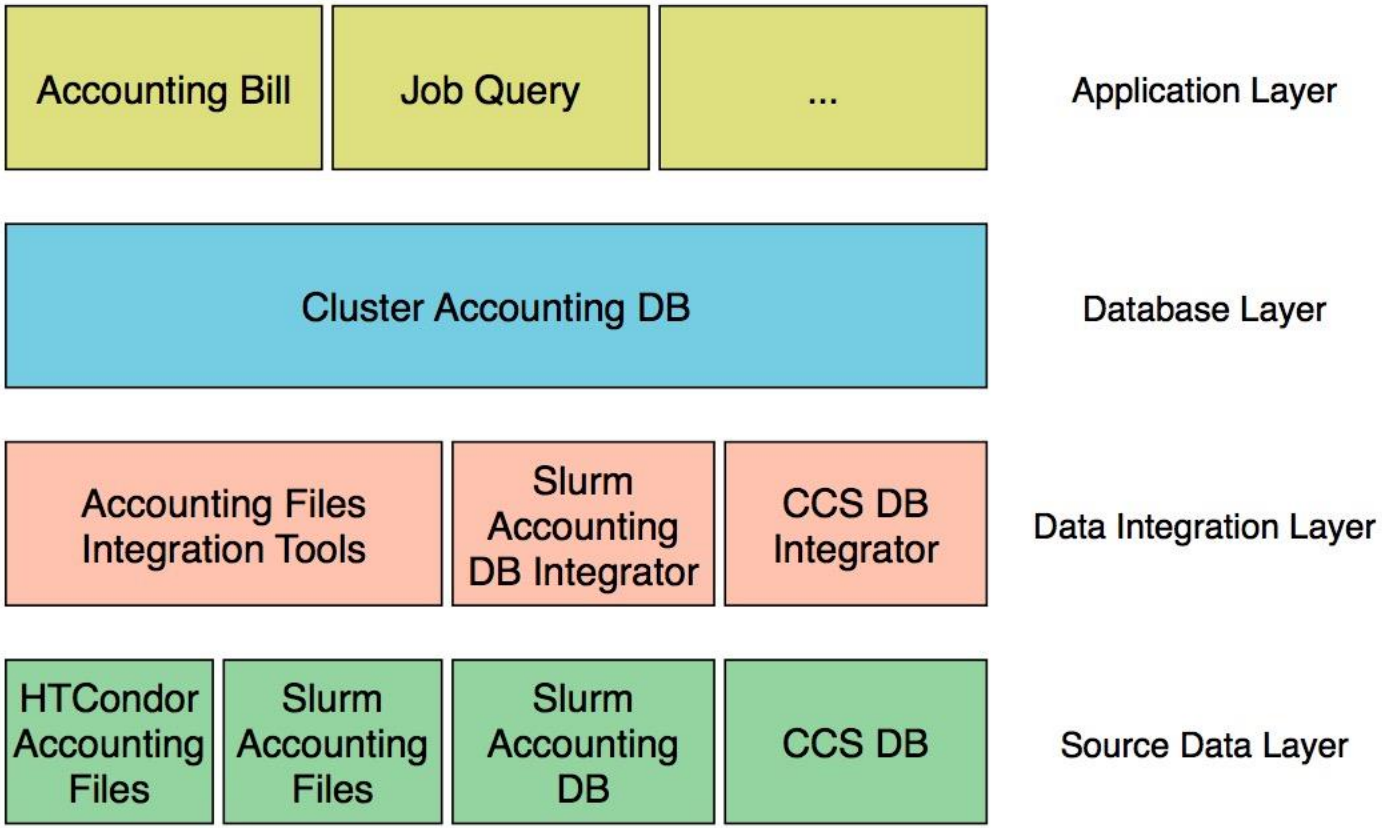


Status Summary For All Host Groups

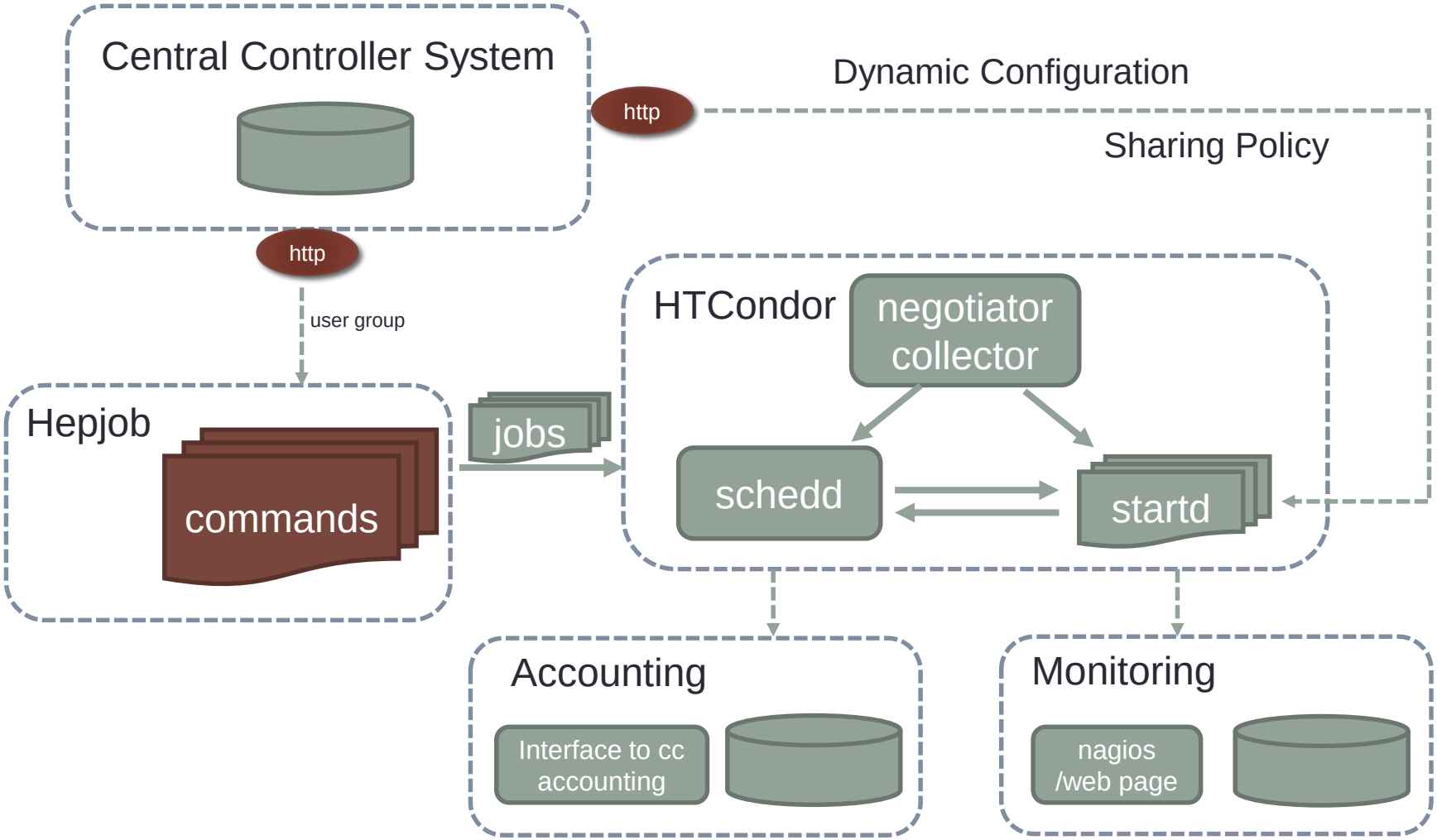
Host Group	Host Status Summary	Service Status Summary
AMS CWS HXMT节点 负责人，闫晓飞 (AMS-Servers)	150 UP	1975 OK 2 WARNING : 2 Unhandled 2 CRITICAL : 2 Unhandled
AWS计算节点 负责人，系统组值班人员 (AWS-servers)	43 UP	551 OK 1 WARNING : 1 Unhandled
bws dbws计算节点 负责人，系统组值班人员 (BWS-Servers)	470 UP	7033 OK 5 UNKNOWN : 5 Unhandled 1 CRITICAL : 1 Unhandled
备份服务器-魏秋玲 (Bak-Servers)	9 UP	25 OK
BIO计算结点负责人系统组值班人员 (Bio-servers)	29 UP	347 OK
计算中心节点cac ccib map nano 负责人，系统组值班人员 (CC-Servers)	46 UP	460 OK
云计算服务器-崔勇 (Cloud-Servers)	9 UP	10 OK
数据库服务器 (DB_SERVER)	9 UP	9 OK
DWS计算节点 负责人，系统组值班人员 (DWS-Servers)	106 UP	1483 OK
数据服务器 负责人 杜国红 杨敏 (Data-Servers)	8 UP	12 OK
GPU 负责人 文硕频6067 (GPU-Servers)	122 UP	1503 OK
存储服务器 (GRASS-Servers)	15 UP	45 OK
负责人 系统组值班人员 (Gluster-Servers)	13 UP	78 OK
高能所网格节点 lwn cac (IHEP-Grid)	67 UP	444 OK 2 CRITICAL : 2 Unhandled
江门中微子计算节点 (JNWS-Servers)	48 UP	506 OK
作业管理服务器pbs condor slurm (Job-servers)	5 UP	10 OK
登录节点 负责人 杜国红 杨敏 (Login-Servers)	70 UP 1 DOWN : 1 Unhandled	866 OK 23 CRITICAL : 11 Unhandled 12 on Problem Hosts

Global Accounting

- Detailed accounting to each group and each user
- Weighting slots with slow/fast CPU, Memory, Disk, etc.



Put All Together



Outline

1. Migration to HTCondor
2. Scheduling Policy to HTCondor
3. Works Designed and Developed
4. **Problems We Met**
5. Summary and Future Work

Problems We Met – dishonest user

- Claimed with other group member to obtain more job slots
- Running sshd daemon at work nods
  - ssh to work nodes without password
  - run big MPI task from login node secretly
  - occupied more cpu cores than the slots declaimed in job
- How to deal with
  - Add group priority check at wrapper of work nodes
  - Zombie process check deployed at work nodes to kill the process which does not belong to jobs running on the work node

Problem We Met – job hung

- Sched daemon can not be connected in a short time suddenly
- Jobs are hung and re-queued unexpectedly
- Reason:
  - Default open file limit: 1024
- How to deal with
  - Increase the system limit
  - Restart sched process

```
03/10/17 17:56:47 (pid:1105883) Started shadow for job
7339826.0 on slot1@bws0472.ihep.ac.cn
<192.168.57.232:6795?addrs=192.168.57.232-6795> for
physics.mahl, (shadow pid = 3809619)
03/11/17 01:50:56 (pid:1105883) ERROR: Child pid 3809619
appears hung! Killing it hard.
03/11/17 01:50:56 (pid:1105883) Shadow pid 3809619
successfully killed because it was hung.
03/11/17 01:50:56 (pid:1105883) Shadow pid 3809619 for job
7339826.0 exited with status 4
03/11/17 01:50:56 (pid:1105883) ERROR: Shadow exited with
job exception code!
```

Problems We Met – sched owner changed

- The owner of “condor_sched” changed from condor to normal user
- Reason:
  - Disk mounted at Sched server inaccessible
- How to deal with:
  - Disk check added and report to monitoring
  - Version upgrade consideration

Outline

1. Migration to HTCondor
2. Scheduling Policy to HTCondor
3. Works Designed and Developed
4. Problems We Met
5. **Summary and Future Work**

Summary and Future Work

- Summary
  - The resource utility has been significantly improved with the resource sharing policy
  - We implemented a number of tools to enhance the system interaction and robustness
- Future work
  - Automatically tuning the resource sharing ratio according to the overloads of each group
    - The integration of Job Monitoring and Central Controller
  - HTCondor sites union

Thank you !

Question?