Resources Sharing Based on HTCondor for Multiple Experiments

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

On behalf of scheduling group of

Computing Center, IHEP

Outline

1 Experiments & Resources at IHEP

2 Motivation of Resource Sharing

3 Works designed and developed

4 Future work

5 Summary

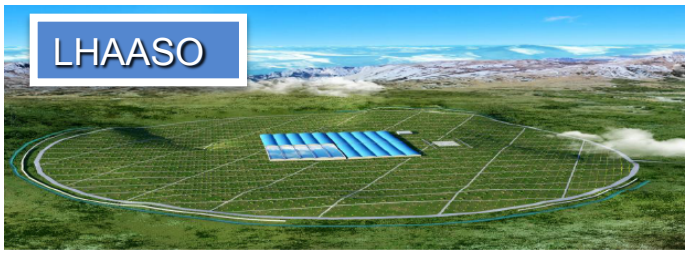
HEP Experiments at IHEP

BESIII (Beijing Spectrometer III at BEPCII)
100TB raw data/year *19

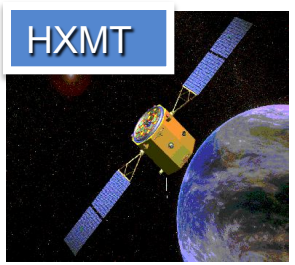
DYB (Daya Bay Reactor Neutrino Experiment)
200TB/year* 9 years

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

YBJ (Tibet-ASgamma ARGO-YBJ Experiments)



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



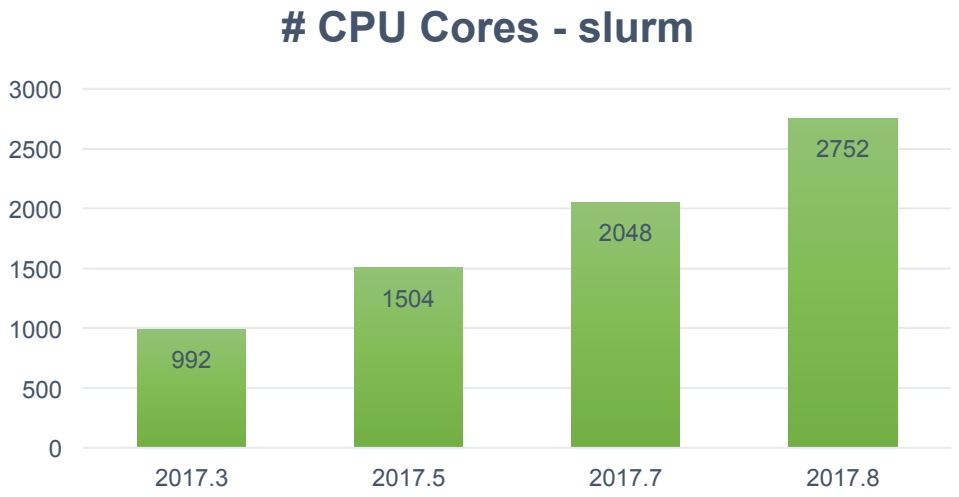
Hard X-Ray Moderate Telescope

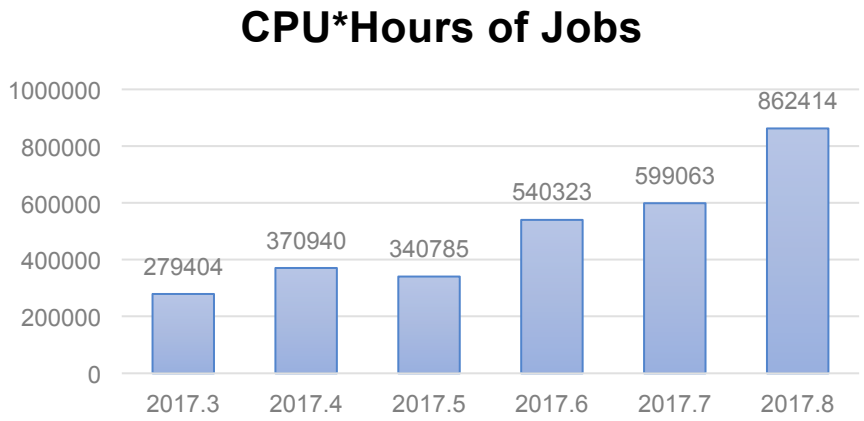
HTCondor Cluster Status

- Resources
  - 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;
  - 100,000 jobs in queue at peak time
  - Serial single-core jobs

Slurm Cluster

- Aim to HPC
- Created this year
- Resources
  - 1 master node
  - 1 accounting & monitoring node
  - 16 login nodes
    - shared with HTCondor Cluster
  - 131 work nodes: 2,752 CPU cores, 8 GPU cards
  - will be scaled to 4000 CPU cores next year
- Jobs (2017.3~2017.8)
  - Parallel jobs
  - # Jobs : ~7,700
  - CPU hours : ~3 million CPU hours

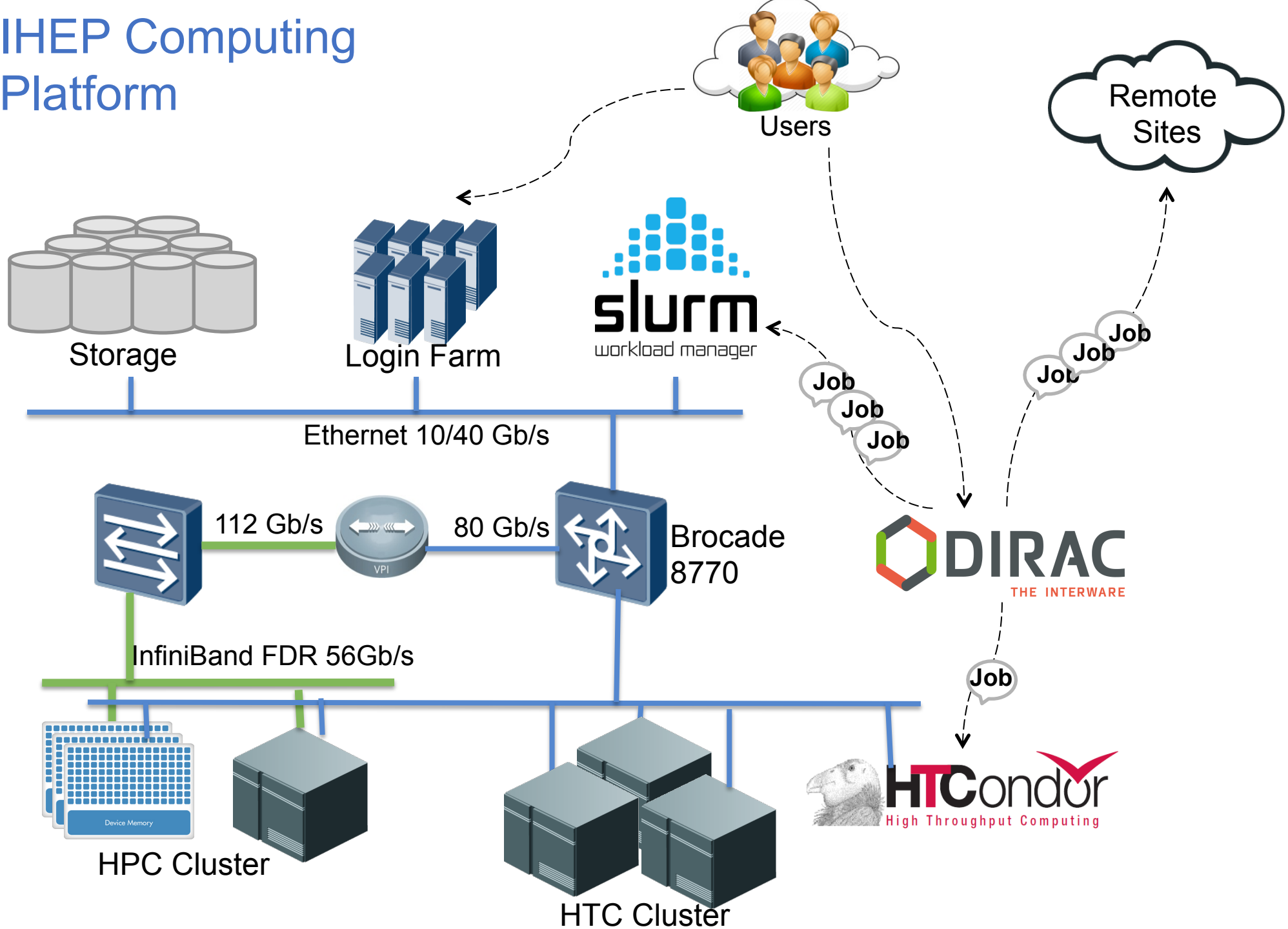




Cloud Computing at IHEP

- Based on Openstack Kilo
- Virtual Computing Cluster
  - **1041** cpu cores—1 cpu core vs 1 virtual core
  - Provide virtual machine on demand of real computing requirement
  - Transparent to users

IHEP Computing Platform



Outline

1 Experiments & Resources at IHEP

2 **Motivation of Resource Sharing**

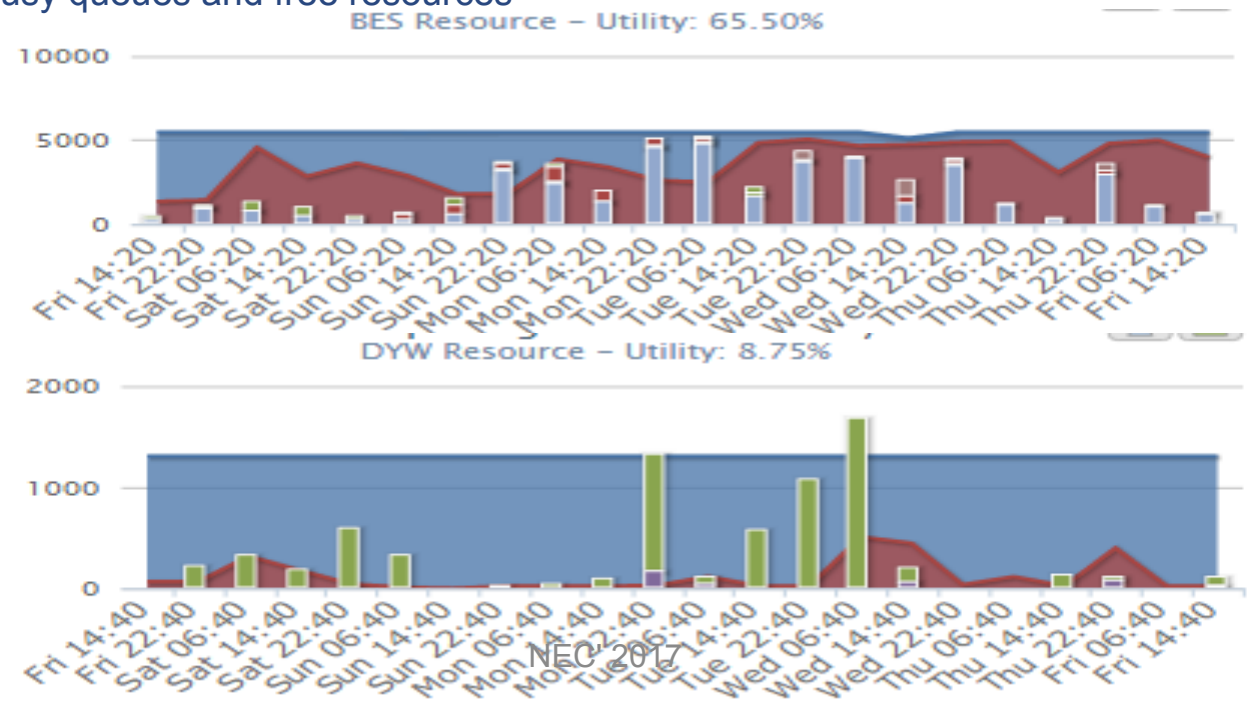
3 Works designed and developed

4 Future work

5 Summary

Motivation: Busy Queue and Free Resource

- Before: Only one PBS Cluster
  - No resource sharing between experiments
  - 55 jobs queues with group permission limits configured
  - Separated resource partition: Each work node running jobs for dedicated experiment
- Resource is limited
- Low resource utility
  - Coexistence busy queues and free resources



Basic Thought: Resource Sharing

- Resource sharing at HTCondor Cluster
  - Break the resource separation
  - Busy exp. can take more resources from the resources of free exp.
  - Fairness guarantee
    - Peak computing requirements from different experiments usually happened at different time period
    - Jobs from free experiment have high priority
    - The more resource the experiment shares, the more its jobs can ben scheduled
  - Virtual machine provides dynamic job slots to meet peak requirement
- Backfill to HPC job slot : under development
  - HTCondor job can be scheduled to and run at Slurm cluster
    - Backfill policy

Outline

1 Experiments & Resources at IHEP

2 Motivation of Resource Sharing

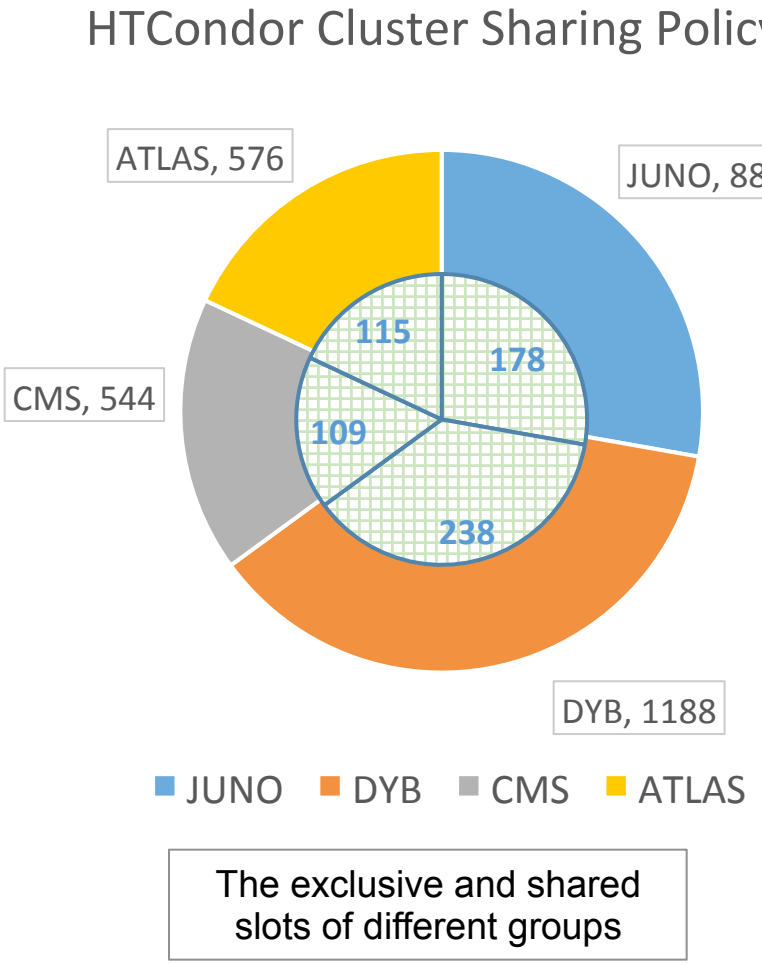
3 **Works designed and developed**

4 Future work

5 Summary

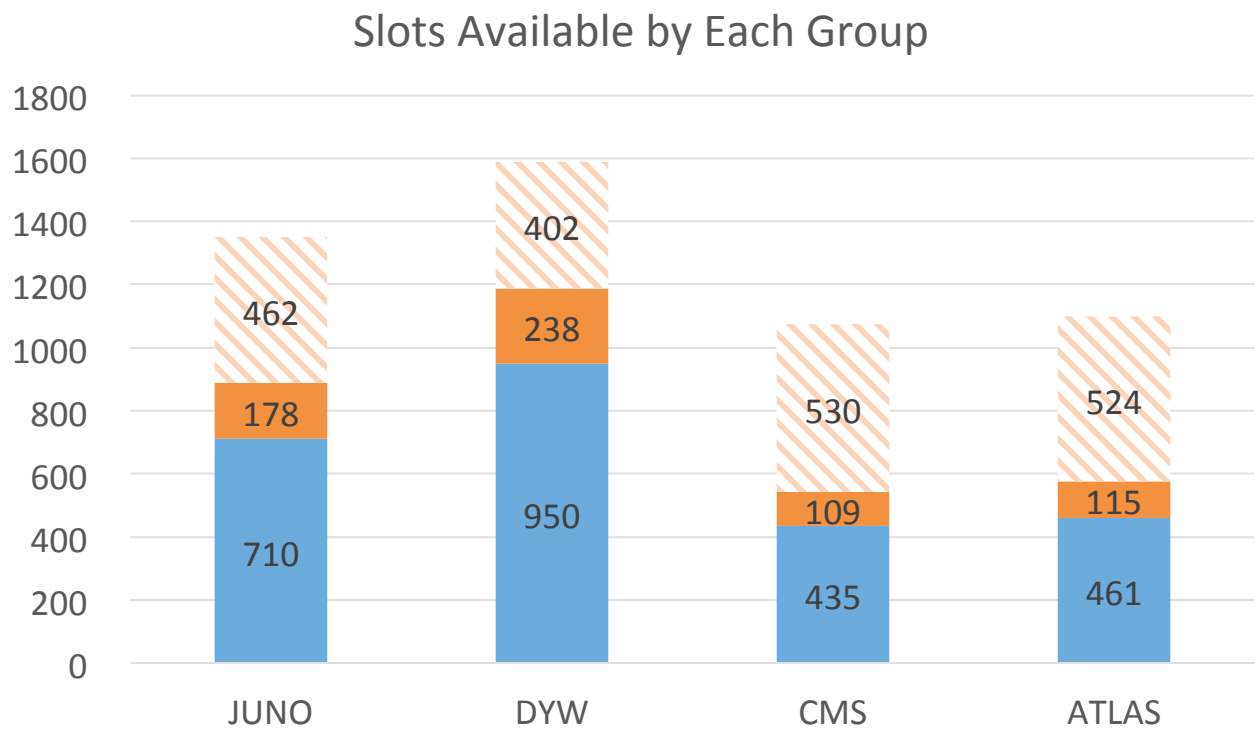
Resource Sharing at HTCondor

- 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 owner
- Shared resource pool
  - Resource contributed by all experiments
  - Slots can be dispatched to all jobs
  - At least 20% slots are shared by each exp.
  - encourage experiments to share more resources



The exclusive and shared slots of different groups

Resource Sharing with HTCondor



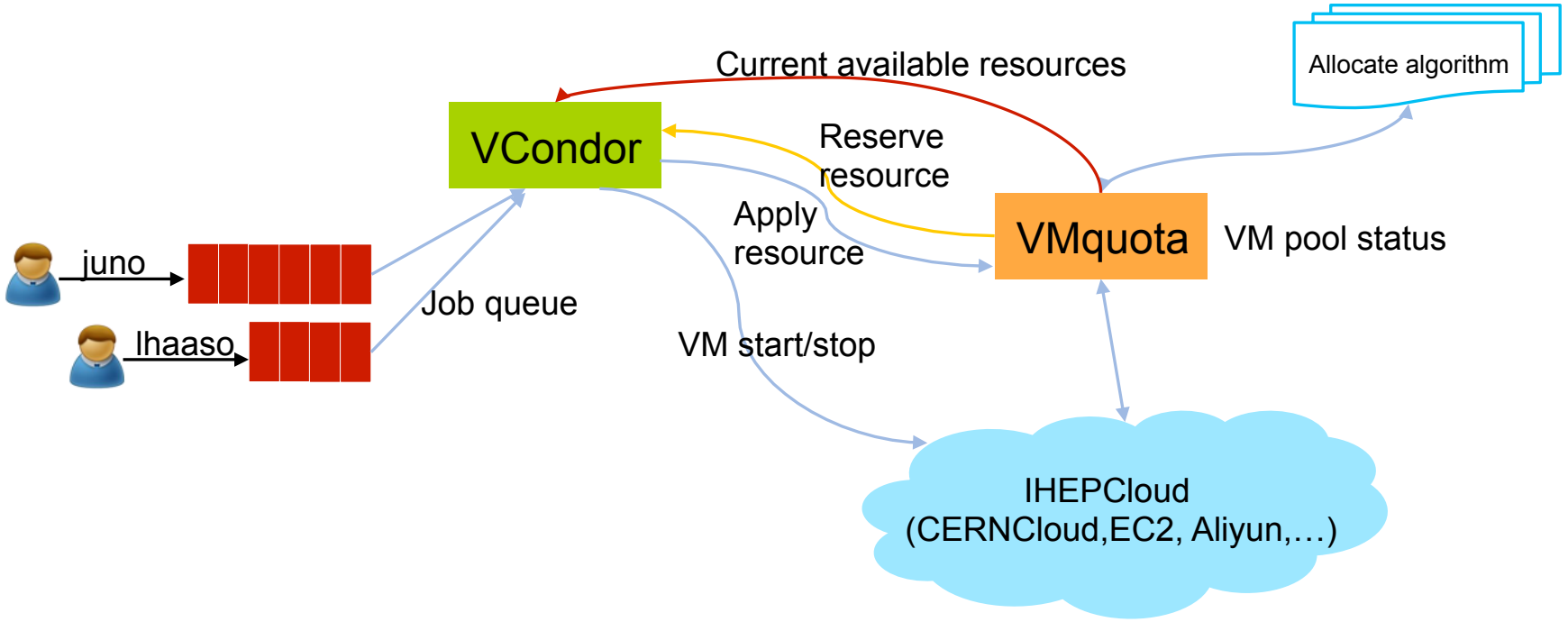
The exclusive, shared and max allocable slots for each exp.

Fairness and priority

- Scheduling preference
  - Jobs are preferred to run on exclusive slots
  - The shared slots are kept for busy experiments
- Group quota
  - The more resources contributed, the more job exp. can run
  - Define linux group for each exp.
  - The initial group quota is set to the amount of real resources from experiments
  - The 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 inside same group users

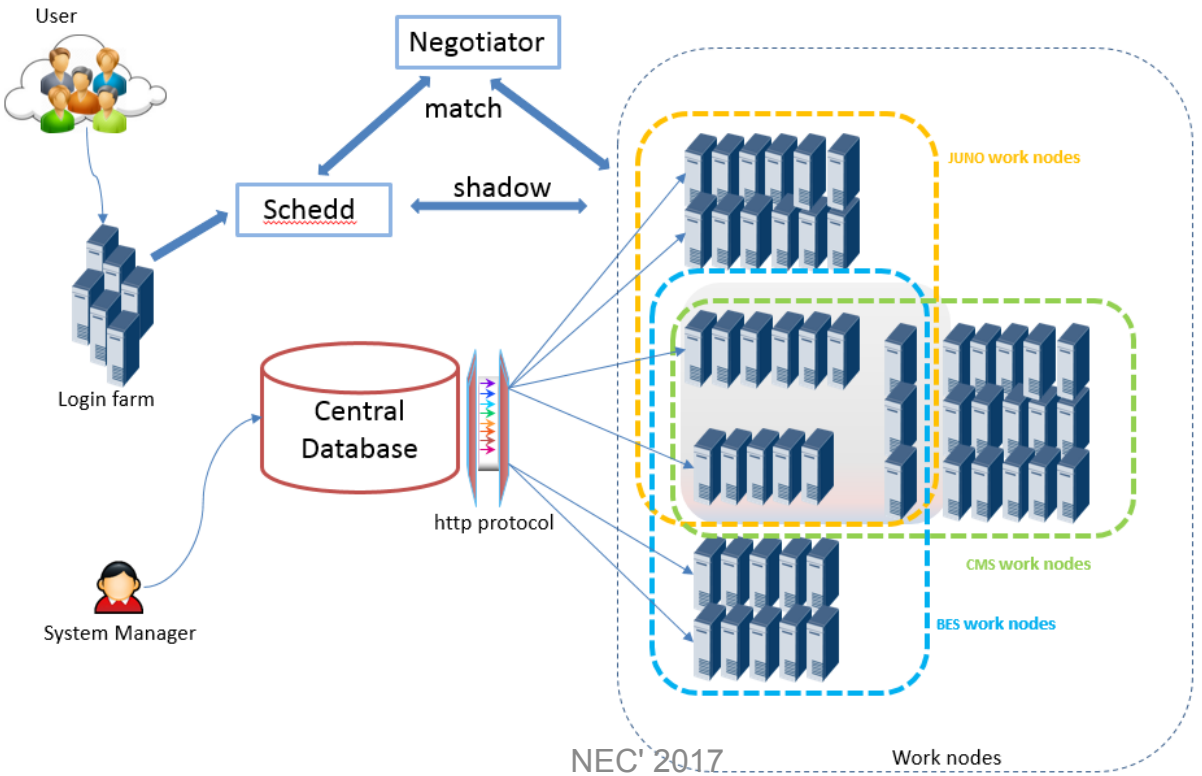
Virtual Computing Cluster Meets Peak Demand

- Allocate resources on demand
  - Busy exp. computing
- Implement resource integration
- Meet peak computing requirement



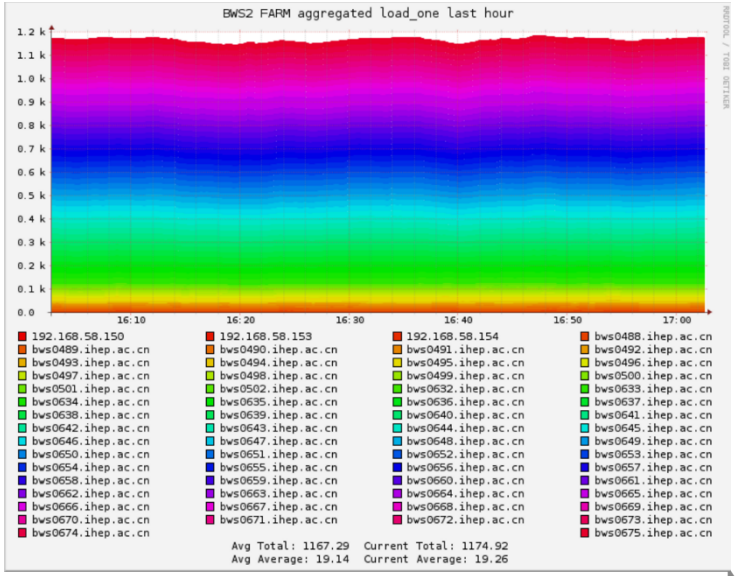
Central Controller

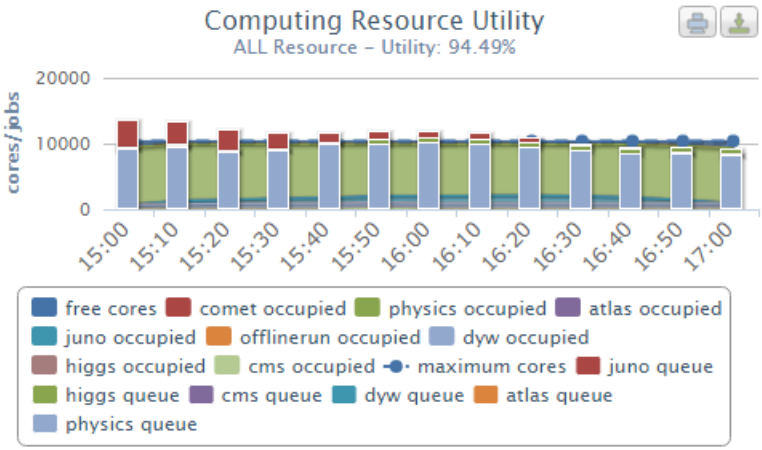
- The central control of groups, users and resources
  - All information is collected into Central Database
  - Necessary information is published to relative services

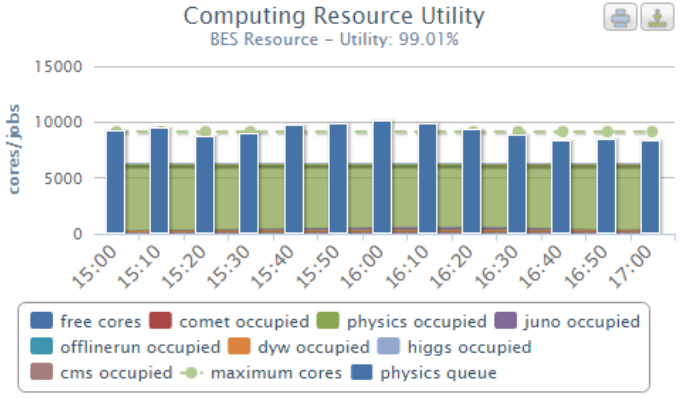


Job Monitoring

- Queuing and running statistics
  - The overall clusters
  - Each group/experiment
- The exclusive and shared resource statistics
- Nagios and Ganglia





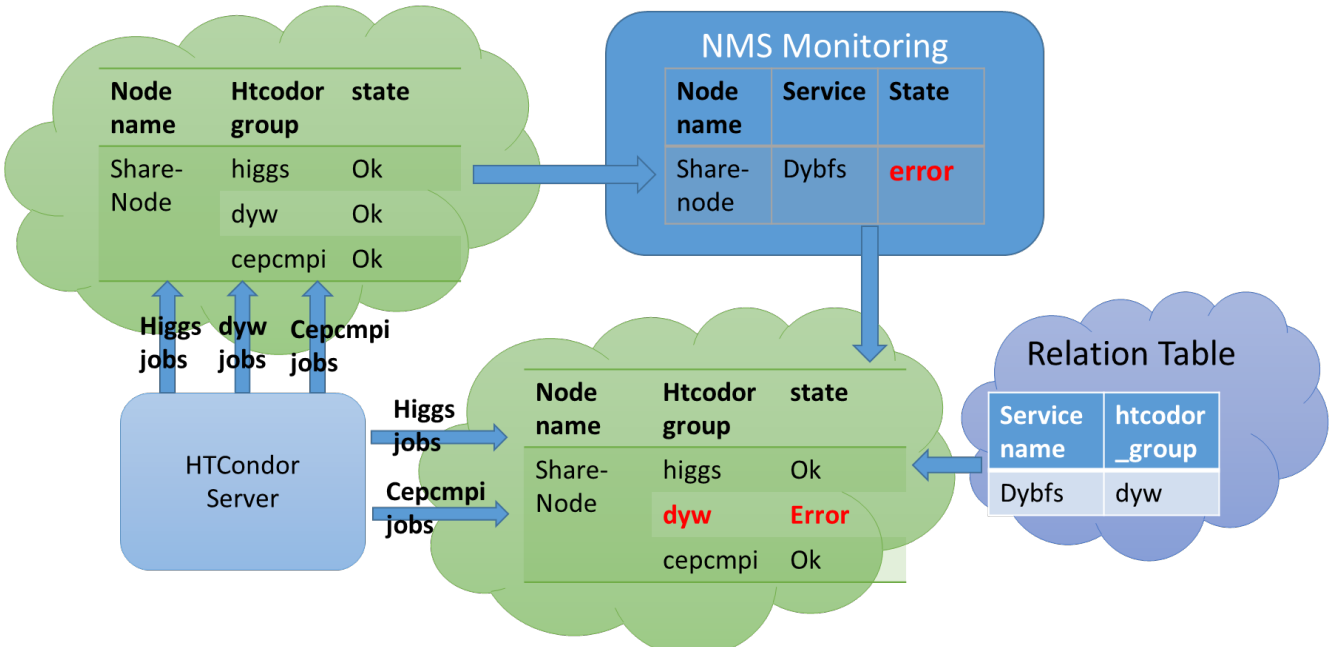


Status Summary For All Host Groups

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-崔涛 (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 杨朝 (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
辽门中微子计算节点 (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 杨朝 (Login-Servers)	70 UP 1 DOWN : 1 Unhandled	866 OK 23 CRITICAL : 11 Unhandled 12 on Problem Hosts

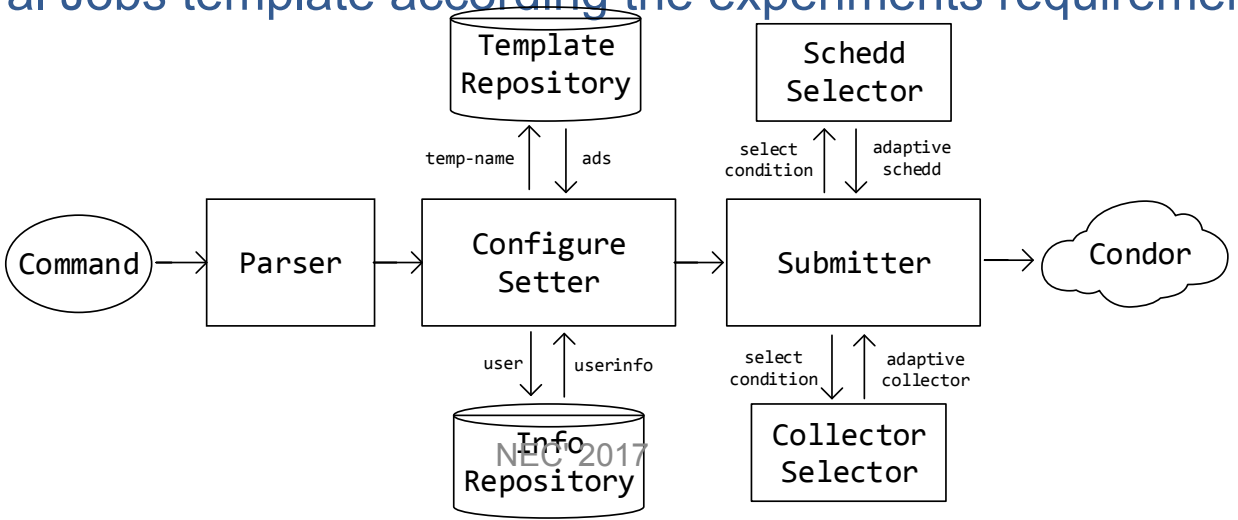
Error Detection and Recovery

- Health status of all workers are collected into Central Database
- Workers' attributes automatically modified through the information published by Central Database



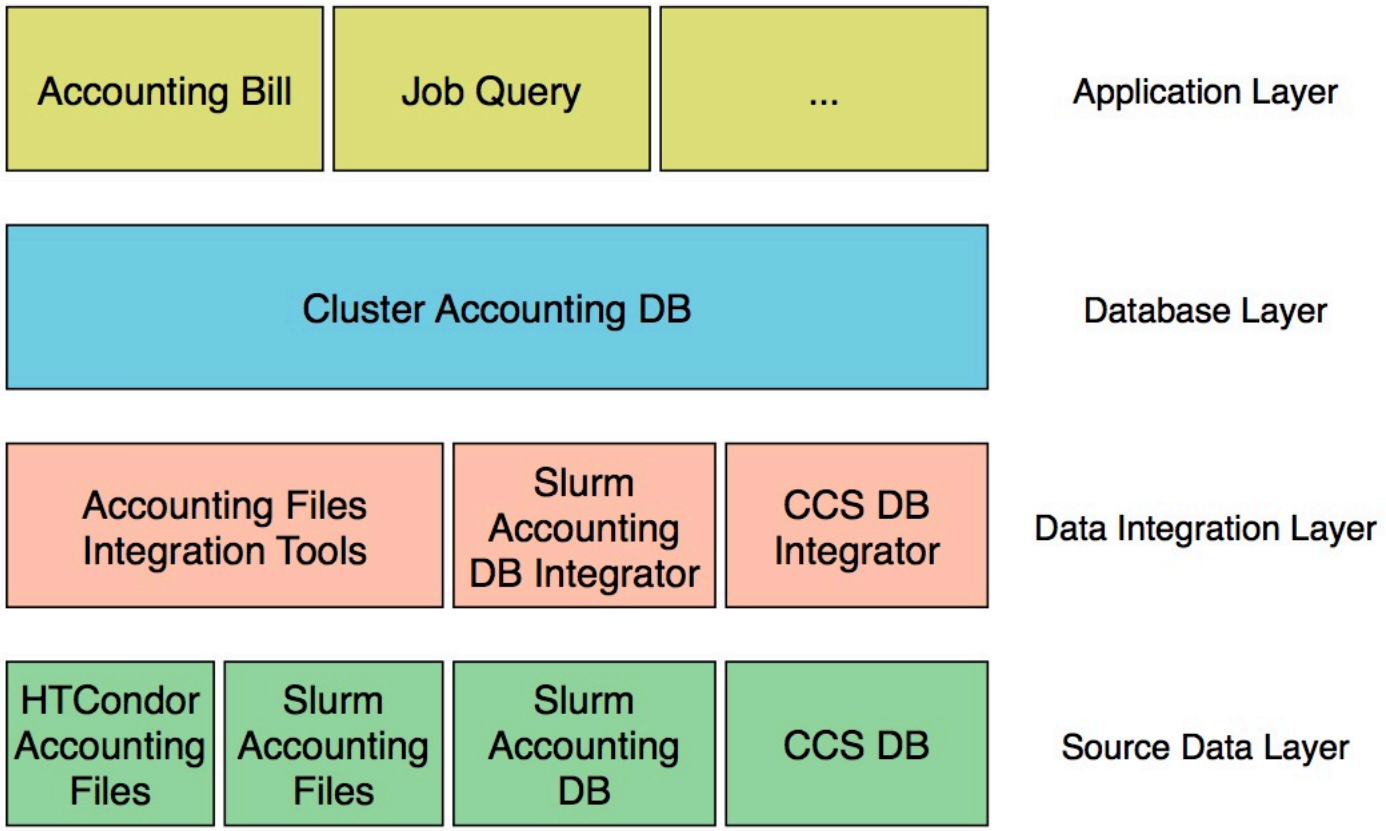
The Toolkit: hep_job

- Motivation
  - 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

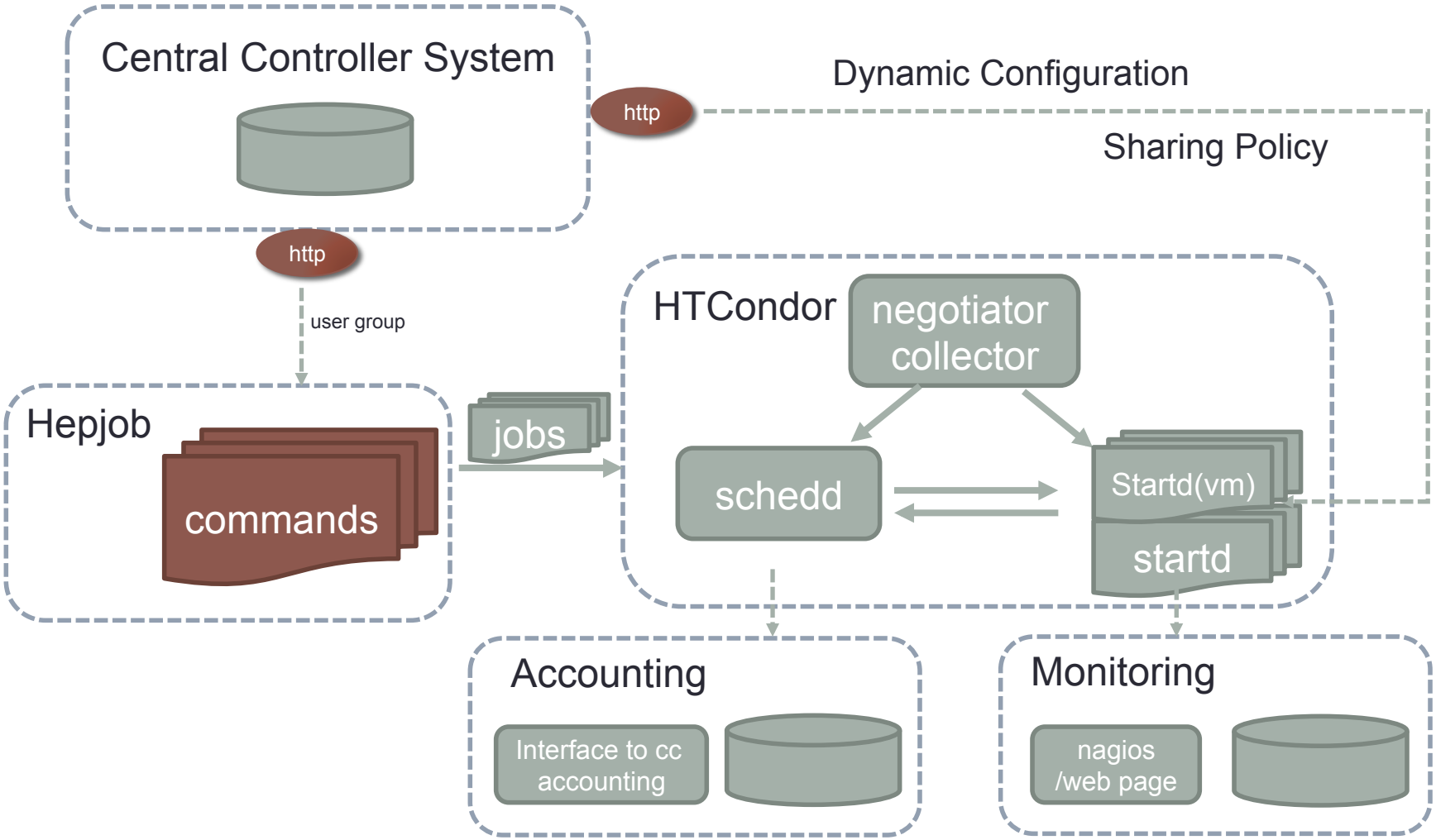


Global Accounting

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

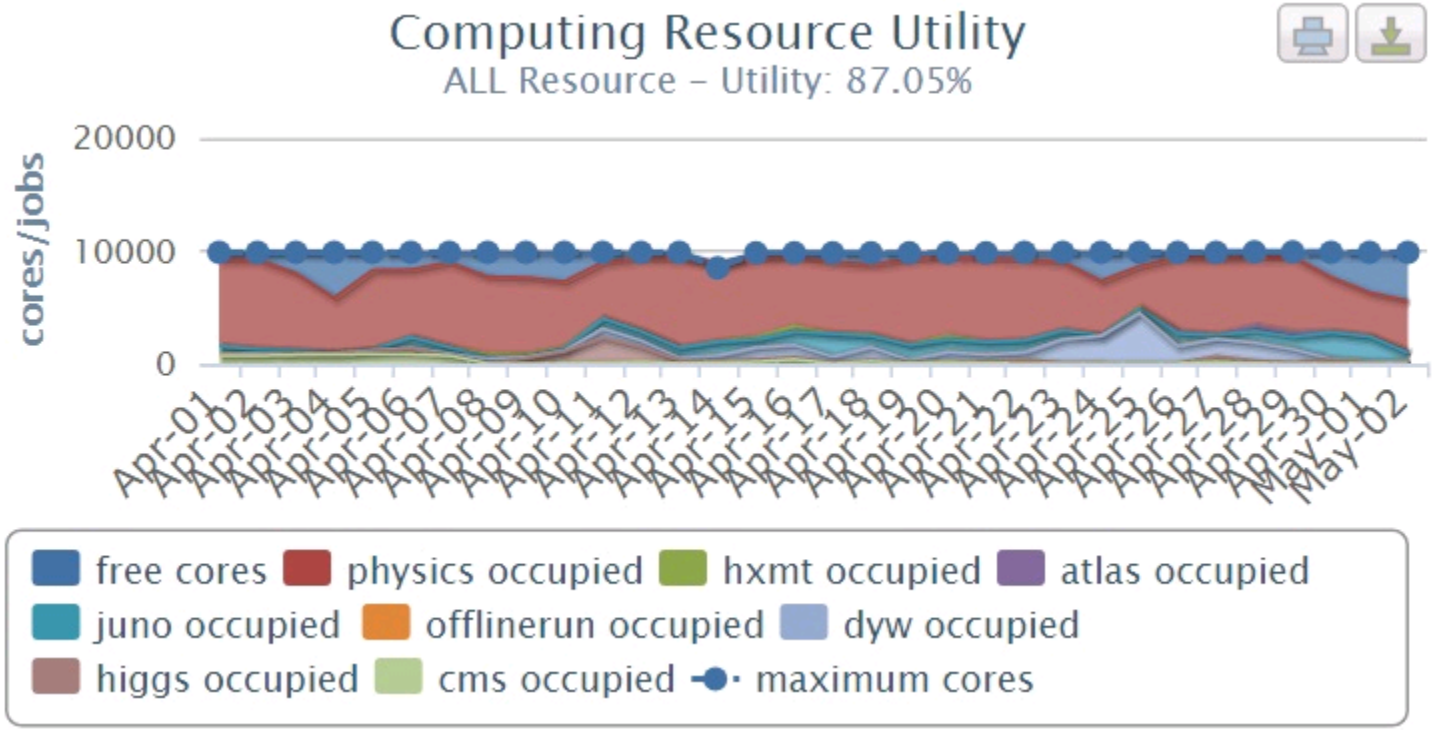


Put all together



Resource Utility Improvement

- The overall resource utility with HTCondor : ~85%



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

Outline

1 Experiments & Resources at IHEP

2 Motivation of Resource Sharing

3 Works designed and developed

4 **Future work**

5 Summary

Future Work

- Automatically tuning the resource sharing ratio according to the overloads of each group
  - The integration of Job Monitoring and Central Controller
- Running HTCondor job at Slurm cluster based on backfill policy
- Supports for remote HTCondor sites

Outline

1 Experiments & Resources at IHEP

2 Motivation of Resource Sharing

3 Works designed and developed

4 Future work

5 **Summary**

Summary

- The throughput (job slot utility) is significantly improved with the resource sharing policy
- We implemented a number of tools to enhance the system interaction and robustness
- More work need to be done

- Thank you !
- Question?