HTCondor Solutions for Several Scenarios at IHEP

Zou Jiaheng

On behalf of Scheduling Group at IHEP

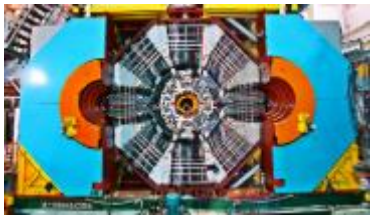
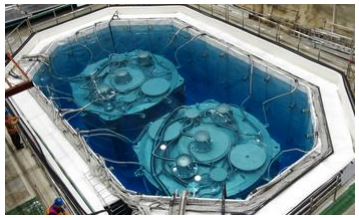
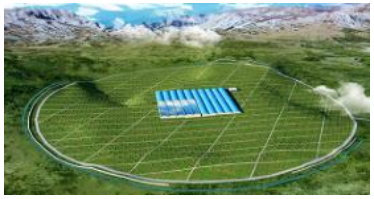
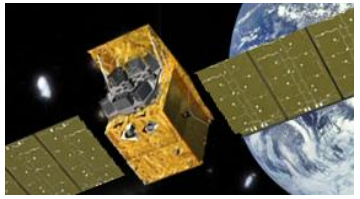
HTCondor Week 2019

Outline

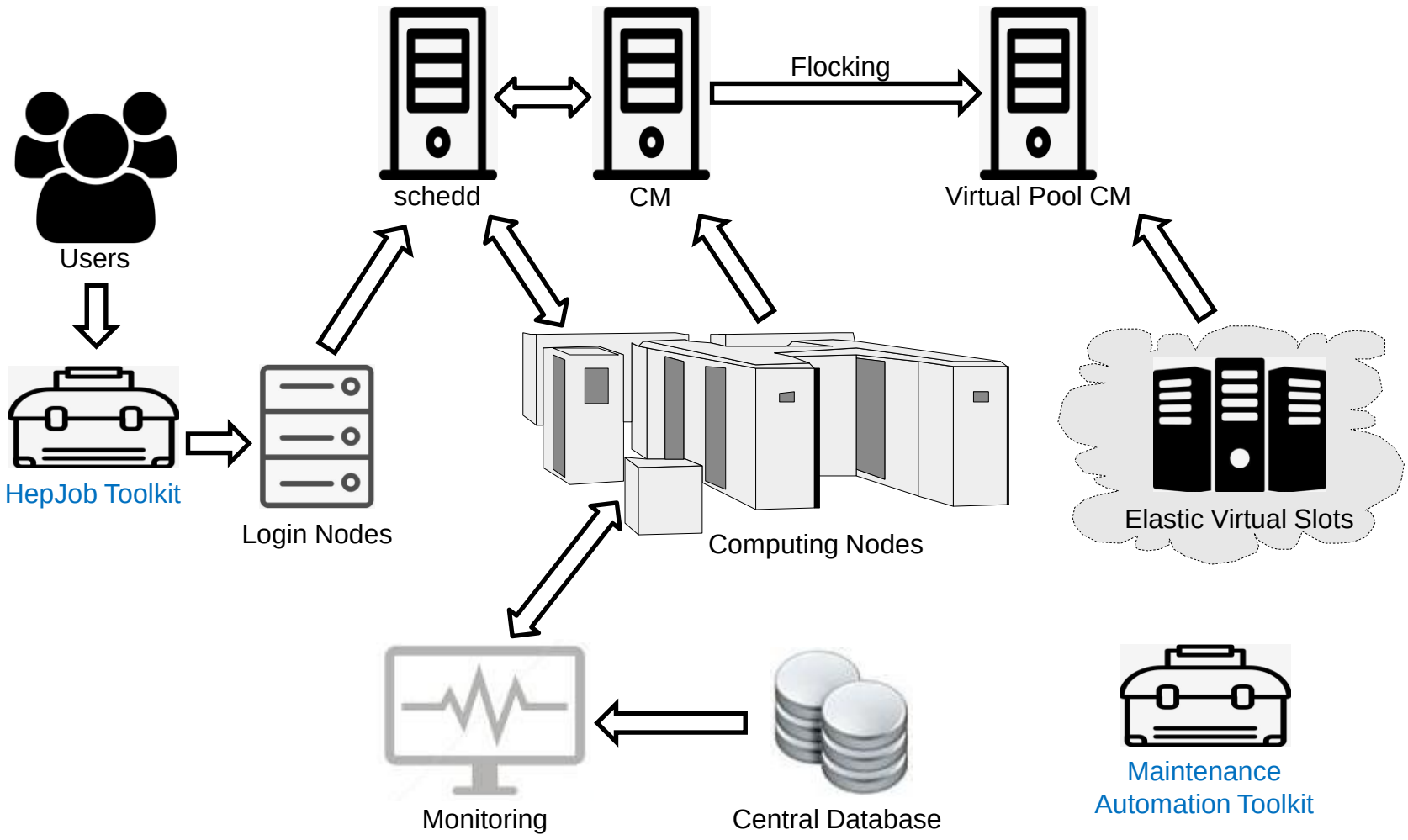
- Brief Introduction
- HTCondor Solutions at IHEP
  - Resource management
  - Job management
  - Abnormality management
- Summary

Computing at IHEP

- HEP Experiments run by IHEP
  - Collider: BESIII, CEPC …
  - Neutrino: Dayabay, JUNO
  - Cosmic ray: HXMT, LHAASO ...
- Local Computing Cluster
  - A HTC Cluster with HTCondor
    - Single core slots, serial jobs
    - >14,000 CPU cores, shared file system
    - 2000+ users from 10+ experiments
    - 300+ active users, 100,000+ jobs/day
  - A HPC Cluster with SLURM
    - MPI and GPU jobs

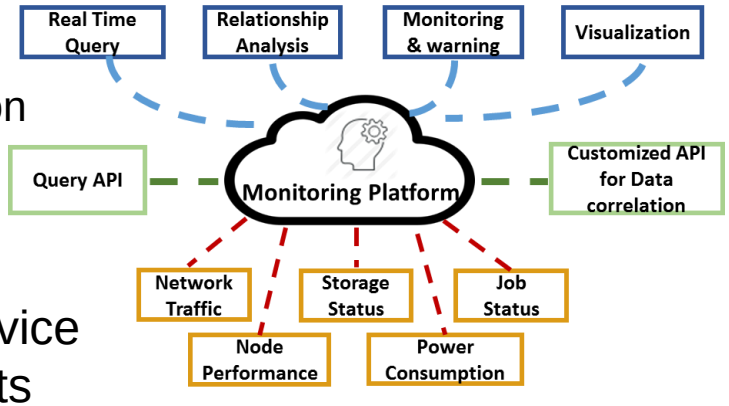






Overview of HTCondor Cluster at IHEP



Toolkits

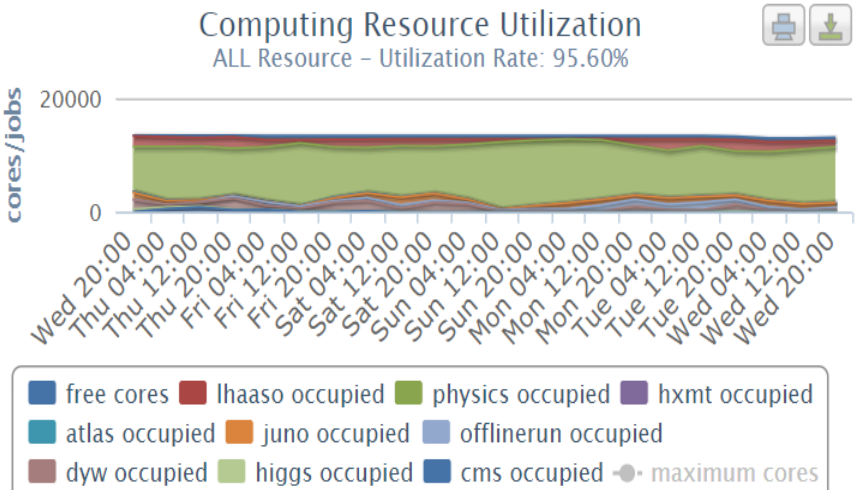
- HepJob Toolkit
  - Based on the Python API of HTCondor
  - A user tool for job submission, querying and deletion
  - Apply customized and mandatory job ClassAD attributes at the backend
- Maintenance Automation Toolkit (MAT)
  - Based on a new monitoring system at IHEP
    - Real-time acquisition, analysis and correlation of multidimensional information
    - Provide APIs for statistical analysis and automatic system alarms
  - Automatically update the HTCondor service configuration based on monitoring results



HTCondor Resource Sharing at IHEP

- 14 different experiments → 14 user groups
- Resources are funded and owned by different groups
- There are always some busy groups and some free groups
- Everyone can derive benefits from resource sharing
- The overall resource utilization rate keeps more than 95%

- Resource sharing policy at IHEP
  - All slots are shared to everyone
  - Group quota is set according to their contribution for fairness
  - Quota surplus is enabled to improve the overall resource utilization

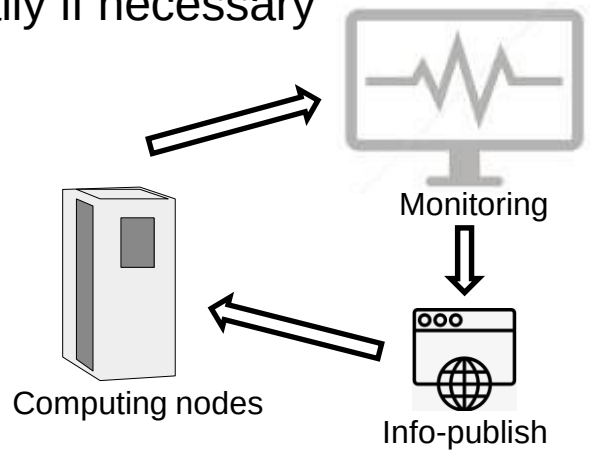


Outline

- Brief Introduction
- HTCondor Solutions at IHEP
  - Resource management
  - Job management
  - Abnormality management
- Summary

Resource Management (I)

- The central database, monitoring and info-publish system
  - All nodes with various attributes are recorded to the central database
  - Healthy detection to each node is collected by the monitoring system
  - Detection results are published via HTTP protocol
- A crontab task is running on each computing node
  - Retrieve its own health state from the info-publish system periodically
  - Update and reconfigure the startd automatically if necessary
- Automatic management of startd
  - Stop the service when there is critical error
  - Stop accepting jobs which are related to the error happened to the node
  - Recover the service when the error is fixed



Resource Management (II)

- Add a new node into the cluster
  - Add a record in the central database
  - Node state is collected and published
  - The node retrieves its own state and configures its HTCondor service automatically
- The evolution of the monitoring and info-publish system (in progress)
  - Lots of connections from the crontab tasks to the info-publish system
  - In general, error does not happen frequently
  - Server side: push error/recovery messages to work node
  - Work node: a daemon listens on the message and reconfigures the node
  - Less connections and less time delay than the crontab tasks

Special Requirements

- There are always special requirements from users
  - Some applications need bigger memory
  - Some users want exclusive nodes for software testing or something
  - And … etc.
- Manual configuration on a selected node
  - HTCondor configure files are loaded in alphabetical order
  - We can override any MAT configuration in a last loaded configure file
  - No need to stop the MAT features, and no side effect
  - But we might forget what we did to which node
- We are considering the integration with MAT in the future
  - Records in the central database will never be forgotten

Job Walltime Limitation (I)

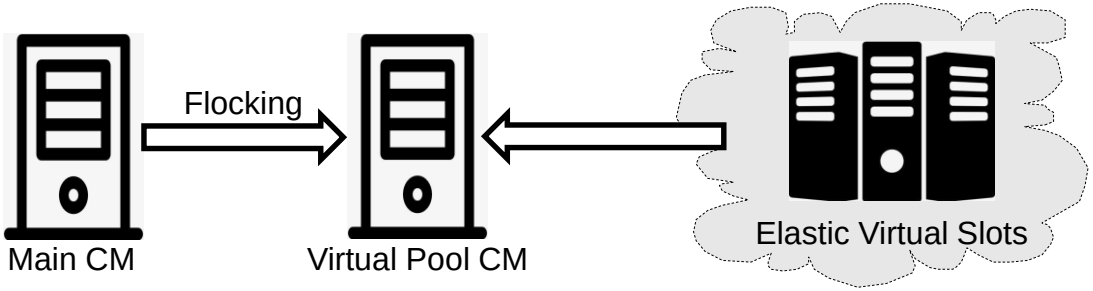
- Motivation
  - Job preempting is disabled, because most of our jobs can't be recovered
  - A large number of long jobs is harmful to fairness
  - We encourage users to set their jobs in proper grain size
- Configuration and effects
  - In the job wrapper, walltime is limited to 100 hours by default
  - In fact, the average job walltime is ~ 2 hours
  - ~ 100 jobs/slots are finished/freed in each minute
  - Higher priority users' job can be scheduled without long time delay

Job Walltime Limitation (II)

- In some cases, a long job can't be divided into shorter jobs
- IHEP HTCondor cluster
  - Users can submit long jobs with the HepJob toolkit
  - The number of long jobs for each user is limited to 10 in HepJob
  - Job walltime limitation is set in wrapper according to the walltime attribute set by the HepJob toolkit
- USTC HTCondor cluster – a remote site that managed by us
  - A small (~2000 CPU slots) cluster with fewer users
  - No complex group competing, no HepJob
  - A "long" group for long jobs
    - Set a quota to the "long" group without surplus
  - Normal jobs can occupy more slots when there are not so many long jobs

Job Flocking to the Virtual Pool

- The virtual pool consists of a elastic number of slots running on virtual machines
  - Can be used for exclusive computing tasks
  - Slots might be added or removed more dynamically
- Keep the architecture be simple
  - 1 schedd and 2 CM: no schedd is associated with the virtual pool
  - No jobs can be submitted to the virtual pool directly
  - Only selected jobs are flocked to the virtual pool



Abnormal User Behavior

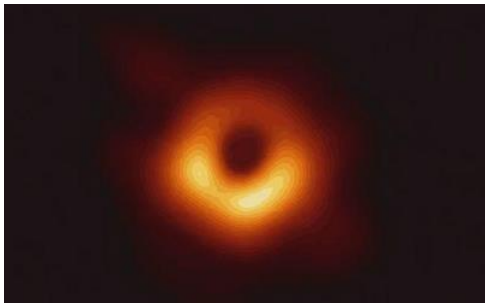
- HTCondor trusts users
- Once we found a user set the job owner to someone else
  - Steal job slots from others, the fairness is broken
  - We stopped the user's account as punishment ~ a sad story
- Then we try to prevent such behaviors in advance
  - In HepJob toolkit
    - Set right owner and accounting group when jobs are submitted
    - Warn the submitter and stop submission when there is any incorrect settings
  - In Job Wrapper
    - Double check to the job owner and accounting group
    - Stop illegal user jobs
    - This is necessary – some users prefer native commands rather than HepJob

Abnormal Jobs

- There are several cases of abnormal jobs, for example
  - Run MPI or multithreading jobs in single core slots
  - Write too much data to the local disk and crash the system
- Such information can be collected and analyzed by MAT
  - Kill abnormal jobs automatically
  - Warn the user and administrator in time
  - Keep computing nodes be robust
- More abnormal conditions will be collected and integrated with MAT

Black Hole Handling (I)

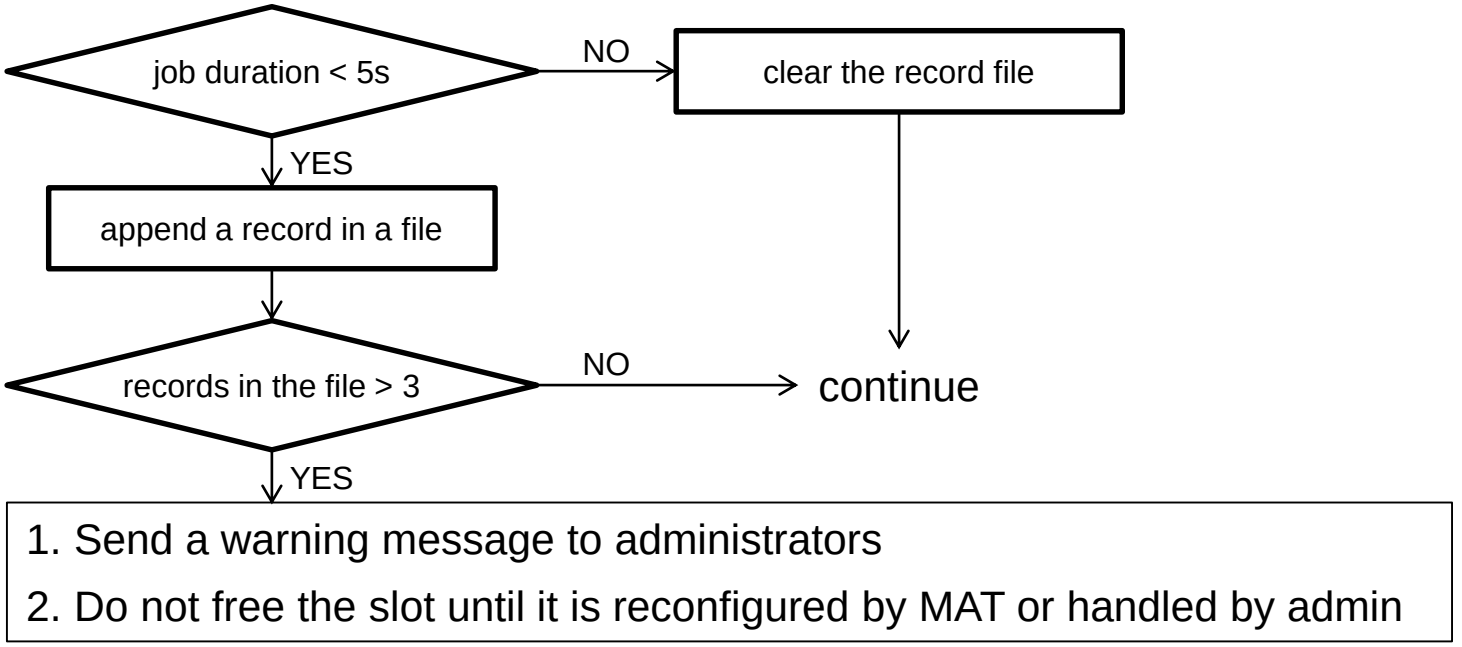
- A node can be a black hole in some conditions, such as a shared file system error
  - Jobs are terminated in a few seconds, and the slots are freed rapidly
  - A large number of jobs are scheduled to the error node and terminated in a very short period
  - A terrible problem to most sites ?



- Black hole detection and handling
  - The Maintenance Automation Toolkit (MAT)
    - Black hole can be detected by the monitoring system
    - The startd can be reconfigured according to the info-publish result
    - But, there is a delay of several minutes ~ thousands of jobs can be ruined
  - We always try to find a faster solution

Black Hole Handling (II)

- The Job Wrapper script of HTCondor
  - All the job information is known in the wrapper
  - Additional checks can be added after users' real jobs



  - This will be deployed at IHEP and the final effect is going to be verified

Summary

- We reach a very high resource utilization rate with HTCondor
- Many efforts are made to improve our computing service
  - Automatic maintenance
  - Detection and handling of abnormalities and system errors in time

Thanks for your attention!