Lightweight Distributed Computing System Oriented to LHAASO Data Processing

Jingyan Shi, Xiaowei Jiang, Chaoqi Guo, Ran Du， **Yaodong Cheng**

shijy@ihep.ac.cn

IHEP – CC

Norfolk, Virginia, USA • May 8-12, 2023
CHEP
2023
Computing in High Energy & Nuclear Physics

Outline

1 Introduction and Motivation

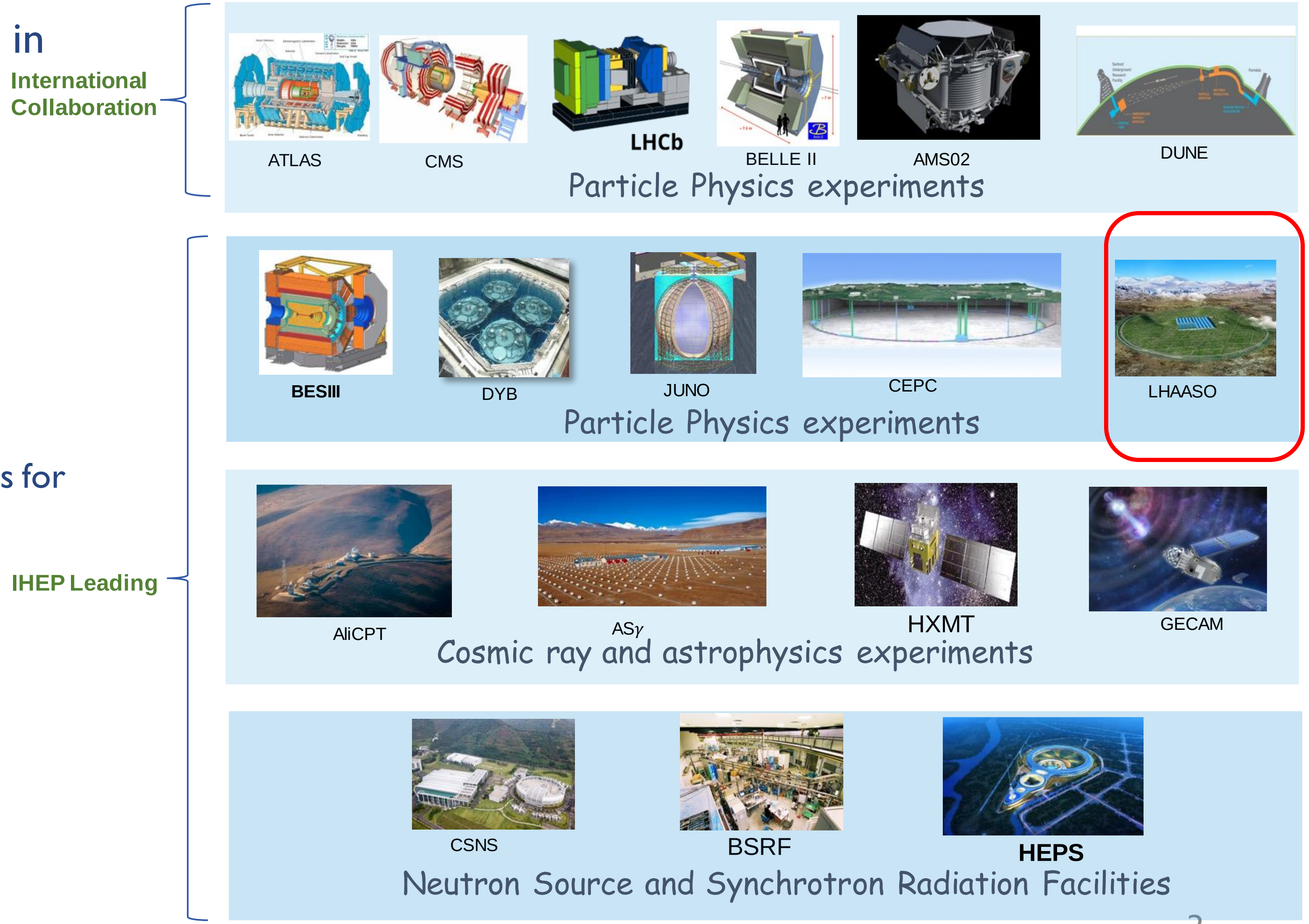
2 Local Cluster Expansion

3 Next Plan

4 Summary

Introduction to the Institute of High Energy Physics (IHEP)

- The largest fundamental research center in China with research fields:
  - Experimental Particle Physics
  - Theoretical Particle Physics
  - Astrophysics and cosmic-rays
  - Accelerator Technology and applications
  - Synchrotron radiation and applications
  - Nuclear analysis technique
  - Computing and Network application
- Computing Center of IHEP
  - Provides computing, storage, network services for HEP experiment offline data processing
    - Computing:
      - HTCondor Cluster, SLURM Cluster, Grid site
    - Storage:
      - Lustre file system, EOS file system
      - EOSCTA is used as the Tape management
    - Network:
      - computing center backbone: 160Gbs,
      - WAN bandwidth: 40Gbs



Particle Physics experiments

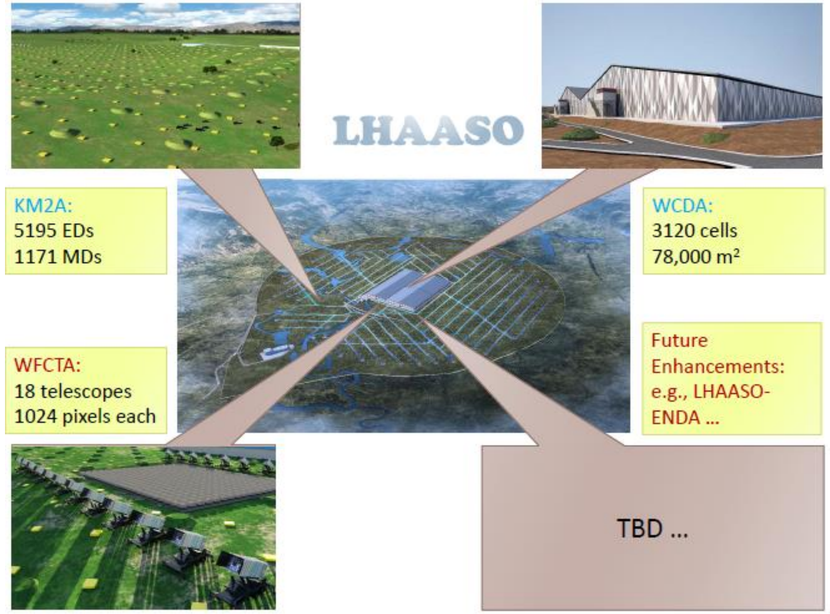
Particle Physics experiments

Cosmic ray and astrophysics experiments

Neutron Source and Synchrotron Radiation Facilities

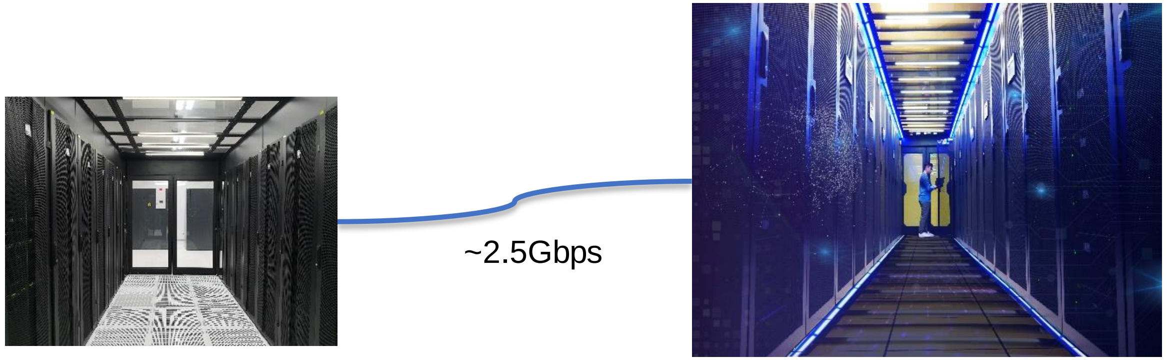
A brief introduction to LHAASO

- Large High Altitude Air Shower Observatory (LHAASO)
  - A new generation all-sky facility
    - Combined study of cosmic rays and gamma rays
    - Wide energy range of $10^{11} – 10^{17}$ ev
  - Located in Daocheng, Sichuan Province
    - Altitude: 4410 m
    - Coverage area: 1.3 km^2
  - Fully completed in Jun. 2021
    - Raw data per year: 13PB (7PB more than the plan)
    - Storage capacity: > 40 PB (20PB more than the plan)



LHAASO Data Processing

- Computing issues
  - No mature data management system developed
  - Most users are not sophisticated
- Computing environment
  - LHAASO software is stored at /CVMFS
  - LHAASO data is stored at local **EOS**
  - Most tasks are HTC job and running at HTCondor cluster of IHEP
  - User auth is based on Kerberos (krb5)
  - A simplified job management tool developed for users
    - For example: hep_sub -g lhaaso job.sh



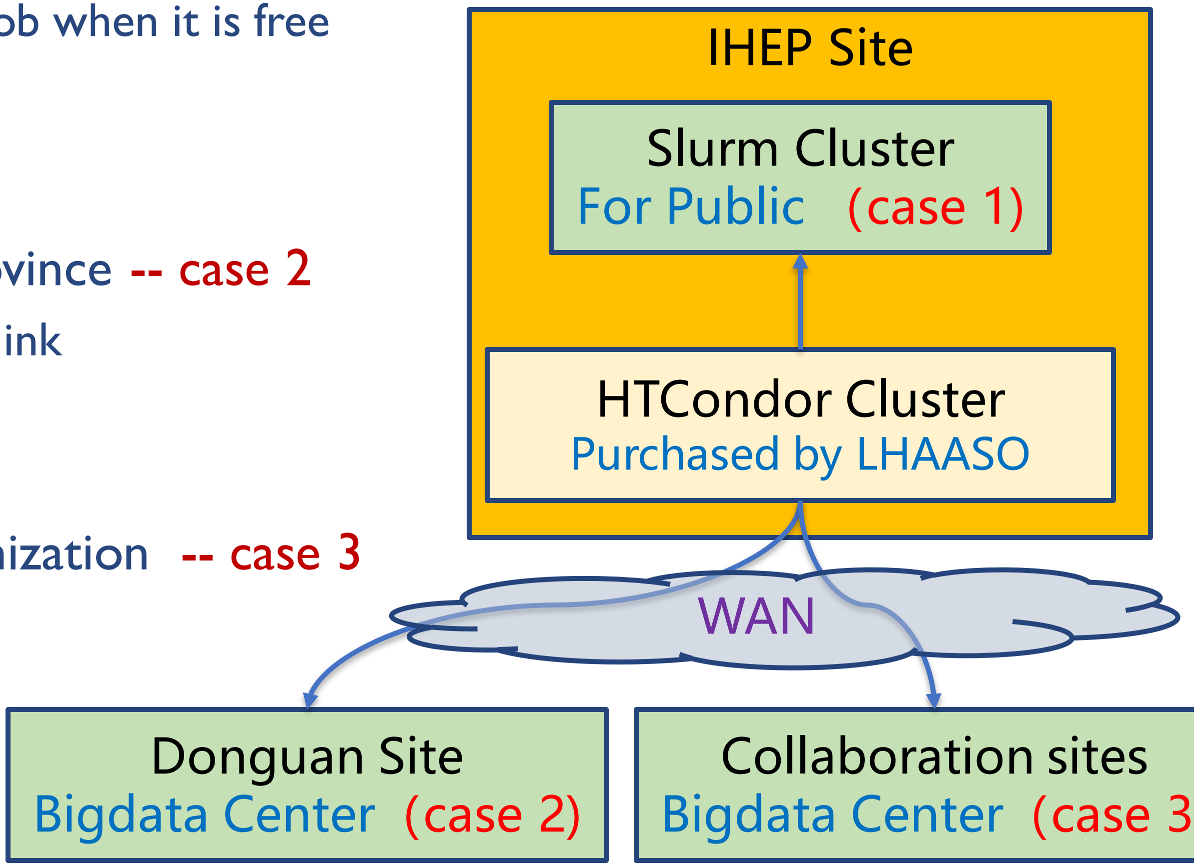
The small on-site Data Center at Daocheng (altitude 4500m)

The Computing Center at IHEP, Beijing

- Big gap between the requirement and reality
  - Estimation: ~20k CPU cores and 40 PB disk storage are required
  - Reality: <11k CPU cores

Find More Resources for LHAASO

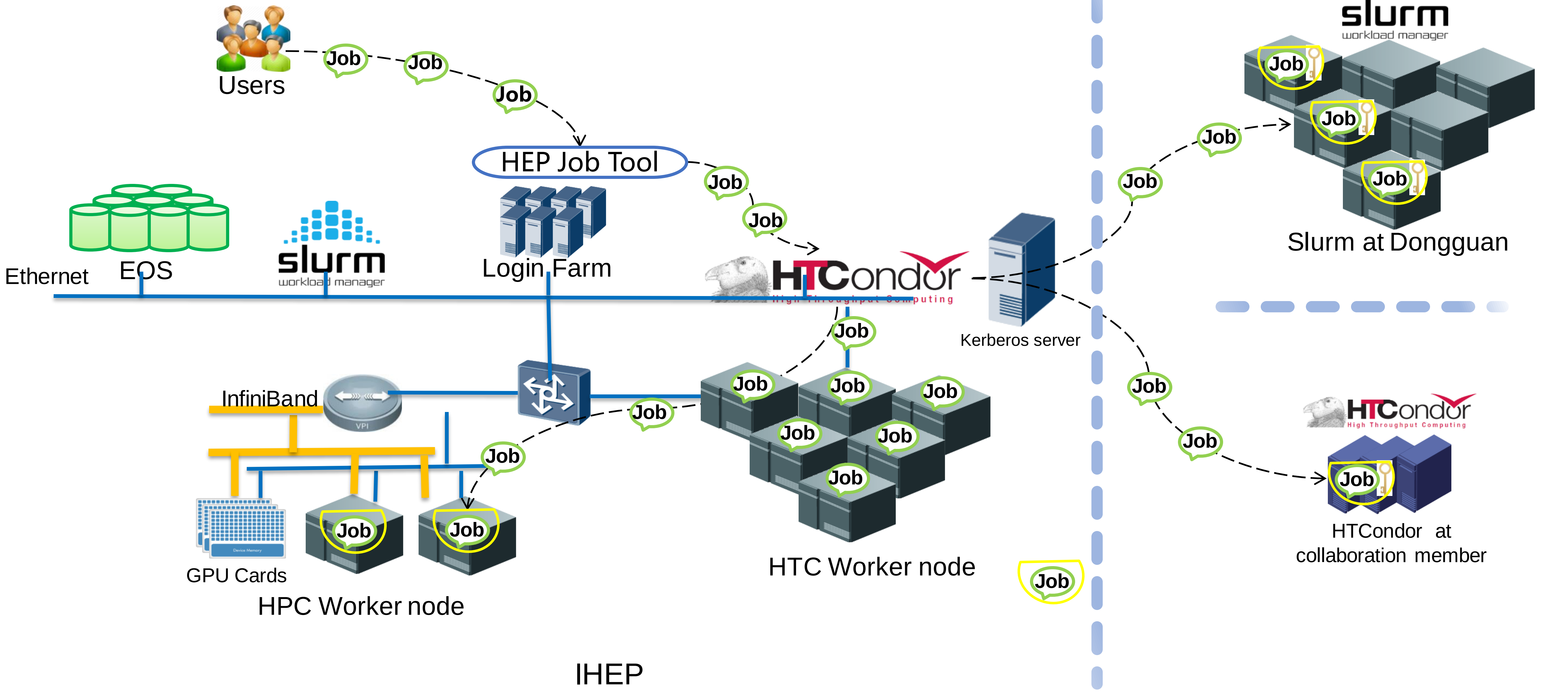
- IHEP local HTCondor cluster (~15k cpu cores) is the main place for LHAASO data processing
- IHEP local Slurm cluster -- case 1
  - One partition (~1k CPU cores) can accept LHAASO job when it is free
    - Known idle time period
    - Same user name space as IHEP HTCondor cluster
    - IHEP EOS is accessible from the slurm worker node
- Big Data center located at Dongguan, Guangdong province -- case 2
  - ~4k X86 CPU and 10k ARM CPU with 10G network link
  - No permanent storage provided
  - Different user name space from the IHEP cluster
- Small sites at domestic collaboration member organization -- case 3
  - Small resources with limited network connection
  - No mature technical support



Light Weight Distributed Computing for LHAASO

- Keep IHEP cluster as the main cluster
- Expand IHEP cluster to the remote resource
  - Add remote worker nodes into LHAASO CPU pool of the IHEP HTCondor cluster
    - Submit glidein batch job to the remote site
    - Run IHEP HTCondor startd inside the glidein job
- Keep the same usage pattern for LHAASO data processing
  - Jobs are submitted to IHEP HTCondor cluster
- Suitable jobs are scheduled to the remote job slots
- User kerberos token is transferred with the user job to the remote worker node
  - Result is copied back to IHEP EOS via xrootd with token
- No direct data access to IHEP EOS during job running

Design of the LHAASO Cluster Extension



Schedule Job to the Suitable Job Slots

- LHAASO job classification
- 3 LHAASO detectors have their own simulation, reconstruction and analysis jobs
  - Classify the jobs based on the CPU time and IO access
  - Take one of the detector, WFCTA, as the example
  - "jobtype" attribute is set by "hep job tool" when user submits the job

Job Type	Input data	Output data	CPU time
corsika	Little	a lot	too much
geant4	a lot	mid	too much
corsika+geant4	little	mid	too much
reconstruction	mid	a little	a little
analysis	mid	a little	a little

Suitable to run at Dongguan

Suitable to run at remote site

Suitable to run at IHEP slurm

Suitable to run at IHEP htcondor

User Authentication

- After User login to the IHEP cluster successfully, his Kerberos token is generated
- The token is transferred to the worker node with the user job
  - Prolong token lifetime
    - Job is in the queue
      - User token is copied to the token dir by hep_job tool and a root deamon is responsible to prolong and clean the tokens
    - Job is running
      - The wrapper inside the glidein exports token path as the environment variable
        - Job access IHEP EOS from the remote site by the token
      - The wrapper starts a process to prolong the token during the job lifetime
      - The token would be cleaned by the "startd" of worker node after the job is finished

No Direct Data Access to IHEP EOS

- Provide WFCTA job script (saved at cvmfs)to the user.
- Both IHEP cluster and remote site use the same WFCTA job script
  - Transfer the input data file to the local disk of the worker node based on the authentication of job token
  - Job result is written to the local disk of worker node firstly
  - The result will be transferred back to the IHEP EOS via XRootd (xrdcp) with the job token authentication
  - Clean the data in the job directory at worker node

Case 1: Running at IHEP Slurm Cluster

- User name space and EOS file system are same as that of IHEP HTCondor cluster
  - Submit glidein jobs to the Slurm worker nodes during the idle period as the root privilege
    - Glidein jobs run as user "condor" which is same as the owner of "startd" daemon running at the local HTCondor cluster
    - LHAASO jobs run inside startd
    - All the types of LHAASO job can run at IHEP SLURM cluster

Case 2 and Case 3: Running at Remote Resource

- Submit glidein slurm/htcondor jobs from login node of the remote cluster
  - Glidein jobs then run a 'startd' daemon on remote nodes which connects HTCondor at IHEP
  - A job slot is added to the IHEP HTCondor cluster
  - Glidein job slot is set only accept dedicated job type job (corsika, geant4 etc.)
- Corsika jobs and geant4 jobs are submitted to IHEP cluster by user
- The job will be scheduled to the glidein job slots at remote site
  - The last step of the job is to transfer result file back to IHEP EOS with the token auth.

Others

- ARM machine support – testing
  - We have about 10k ARM CPU cores
  - Compile LHAASO software on ARM architecture
    - Physical Result evaluation is under going
  - Compile HTCondor on ARM architecture
    - ARM HTCondor worker node is ready

Next Plan

- Lightweight Distributed Computing System Oriented to LHAASO Data Processing provided 2.4M CPU hours and generated 80TB simulation data for LHAASO
- Next Plan
  - ARM machine will be in production next month
  - Glidein factory is under going
  - More efficient scheduling algorithms need to be developed

Summary

- LHAASO needs more computing resources
- A lightweight dHTC designed and deployed for LHAASO
  - expand IHEP local cluster to the remote site
  - Keep the user cluster usage pattern
  - Have integrated remote resource from several sites
- More works need to be done

Thank you for your attention!

Any questions?