dHTC for LHAASO Exp.

Jingyan Shi, Xiaowei Jiang, Ran Du

Institute of High Energy Physics, Chinese Academy of Science
(IHEP, CC)

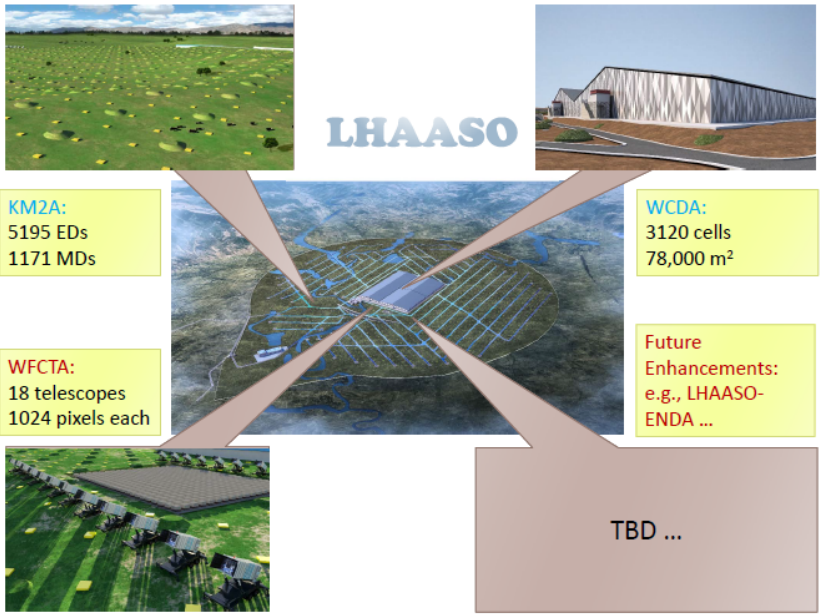
shijy@ihep.ac.cn
jiangxw@ihep.ac.cn
duran@ihep.ac.cn

Outline

1. Brief Introduction to LHAASO
2. Motivation
3. Design and Architecture
4. Current Status and Plan
5. Summary

Introduction to LHAASO Project

- 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 June, 2021
    - Data taken starts in 2018
    - Raw data per year : 6 PB
    - Storage Capacity > 20 PB



LHAASO Data Processing

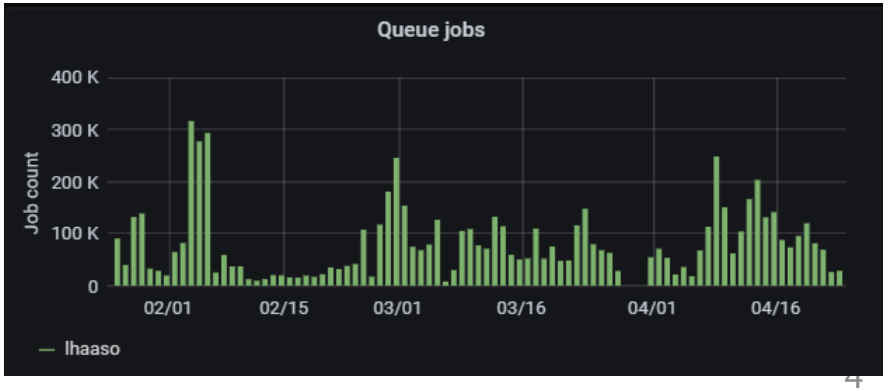
~2.2Gbps

- The small on-site Data Center at Haizi Mountain observatory (altitude ~4500m).
- ~2000 CPU cores and 1.64PB disk storage for calibration and rapid reconstruction.

- The large Offline Data Center at IHEP
- ~4000 CPU cores, 4PB disk storage and 20PB tape storag

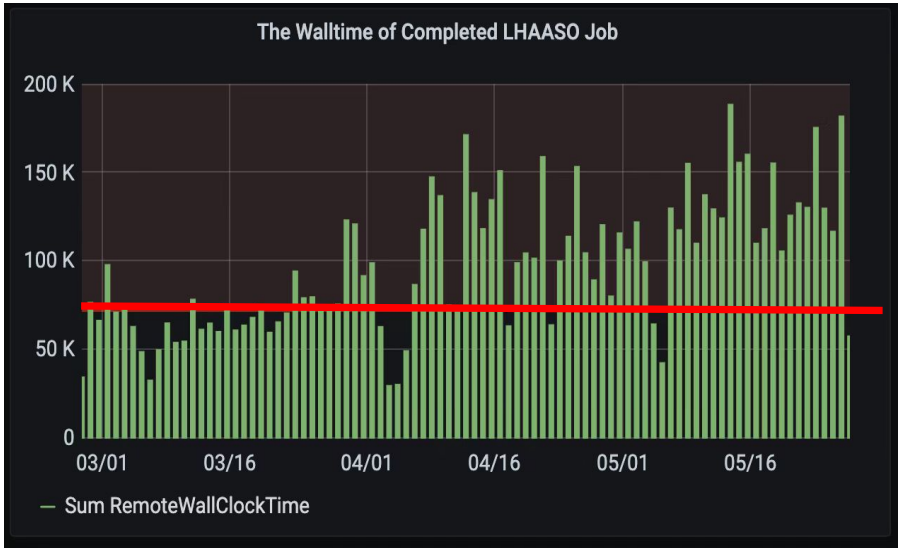
- Computing issues from software and users
  - LHAASO software is still under development
  - No mature data management system developed
  - Most users are not sophisticated
- LHAASO Computing requirement
  - Most data processing is serial with single CPU core running at HTCondor cluster at IHEP
  - Estimation: ~20K CPU cores and 20 PB disk storage are required
  - Reality: <5K CPU cores

- A simplified job management tool developed for users
  - For example: hep_sub -g lhaaso job.sh
- Big gap → jobs have to wait in queue for a long time



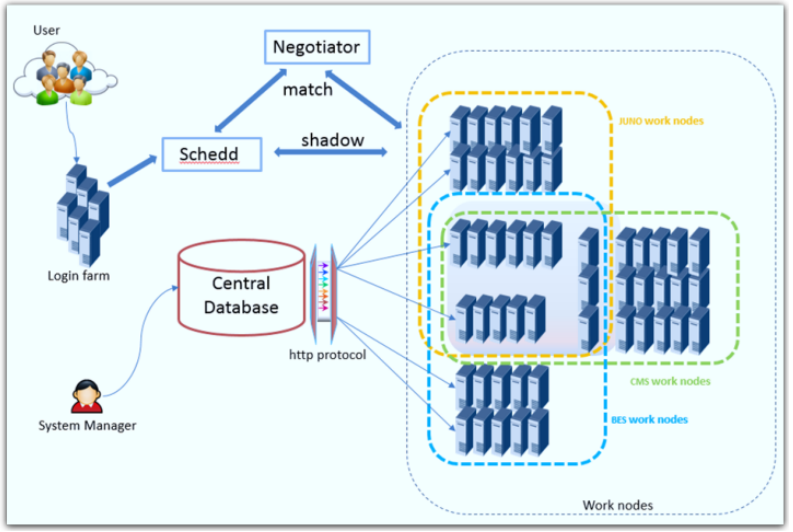
Find More Computing Resources for LHAASO(1)

- The HTCondor cluster at IHEP site -- ~25k cpu cores
  - Main persistent computing resources for LHAASO
  - All experiments share their CPU cores in a big pool
  - Fair share policy is adopted based on group quota for each Exp.
  - MAT[*] system is used to control the "match condition" on worker nodes
    - Maintenance automation tool for the dynamically adjust worker nodes and job matching policy
  - Average job slots efficiency of IHEP HTCondor cluster is over 90%
    - LHAASO got more CPU time from other exp. resources



Contributed by LHAASO

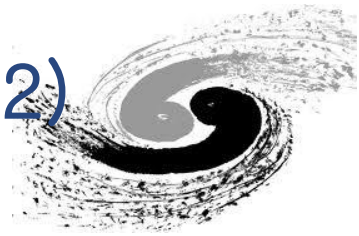
The walltime statistics of LHAASO job at HTCondor cluster



Architecture of MAT system

* Jingyan Shi;etc.; An automatic solution to make HTCondor more stable and easier, 24th International Conference on Computing in High Energy and Nuclear Physics

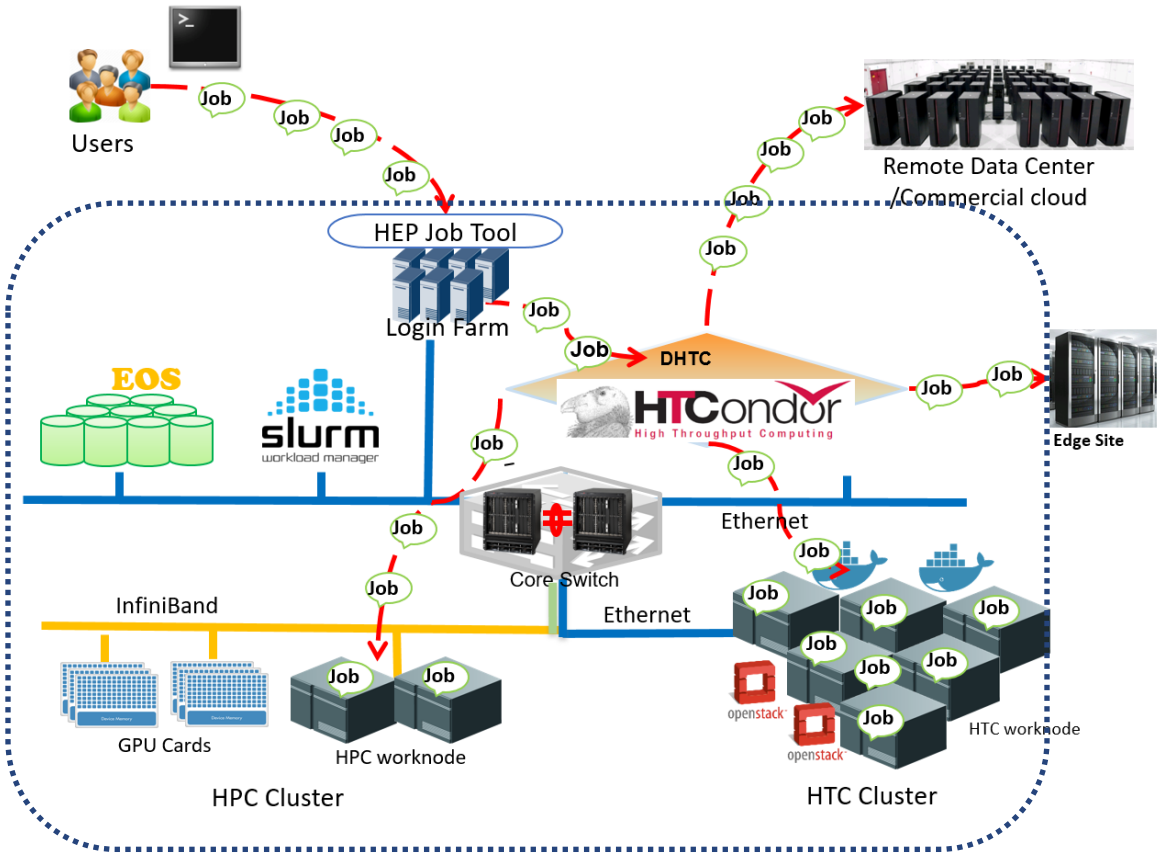
Find More Computing Resources for LHAASO(2)



- The Slurm cluster at IHEP -- case 1
  - Resource partitions with ~1000 CPU cores
    - Willing to accept LHAASO jobs when the partitions are idle
    - Known idle period
    - Same username space(uid/gid) as the HTCondor cluster
- More domestic computing sites would like to contribute resources
  - Large sites : Big Data Center -- case 2
    - A new Data Center supported by HUAWEI is built in Dongguan, Guangdong province
      - 30k intel x86 CPU cores, 10k ARM CPU cores and 3PB disk storage
      - 10 Gbps network connection to IHEP
  - Small sites from the collaboration organization -- case 3
    - Small scale and short of maintenance man power
  - Not a stable and persistent provider -- no MoU
    - The amount of the resources contributed to LHAASO would be changed based on site own requirement

Simplified dHTC for LHAASO

- Computing mode
  - Distributed High Throughput Computing:
    - A local cluster + glidein jobs
  - Keep the **same usage patterns**
    - Integrate remote computing resources into the HTCondor cluster at IHEP
- 3 issues need to be considered
  - Easy to integrate
  - Data access
    - LHAASO data processing features typical massive I/O mode
  - Authentication & Authorization



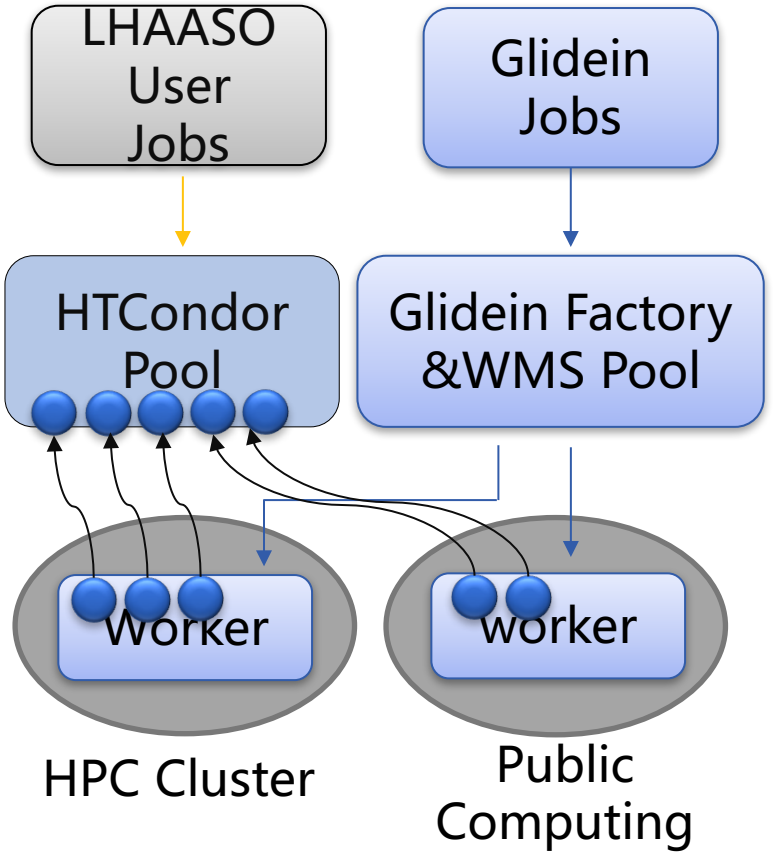
Simplified dHTC based on a local HTCondor cluster

Schedule LHAASO Jobs to Remote Worker Nodes

- Scheduled LHAASO jobs from local HTCondor cluster
- Issue 1 : easy to integrate
  - LHAASO jobs running in containers
    - Publish singularity image for LHAASO job to /cvmfs
    - Glidein jobs run in Singularity containers
    - Access LHAASO software from /cvmfs
  - Glidein job management
    - Submit glidein jobs manually during the know idle period
    - Next step: A simple glidein job tool is under development
      - Generate and submit glidein job automatically based on the real idle job slots
  - Keep the same usage pattern
    - All are wrapped in hep job tool
    - Example:
      - hep_sub **–site remote -jobtype lhaaso_wcda_simrec** job.sh
- Issue 2 : data access
  - Policy : try to reduce data transfer as much as possible
  - EOS replica mechanism for big data access
  - Xcache for temporary remote data access
- Issue 3 : authentication & authorizaiton
  - Now : the simplest way
    - Group "lhaaso" should be created on each remote site
    - LHAASO users are mapped to one or two "lhaaso group" users at remote sites
    - Glidein jobs are run with lhaaso uid/gid
  - Next step: Based on Tokens

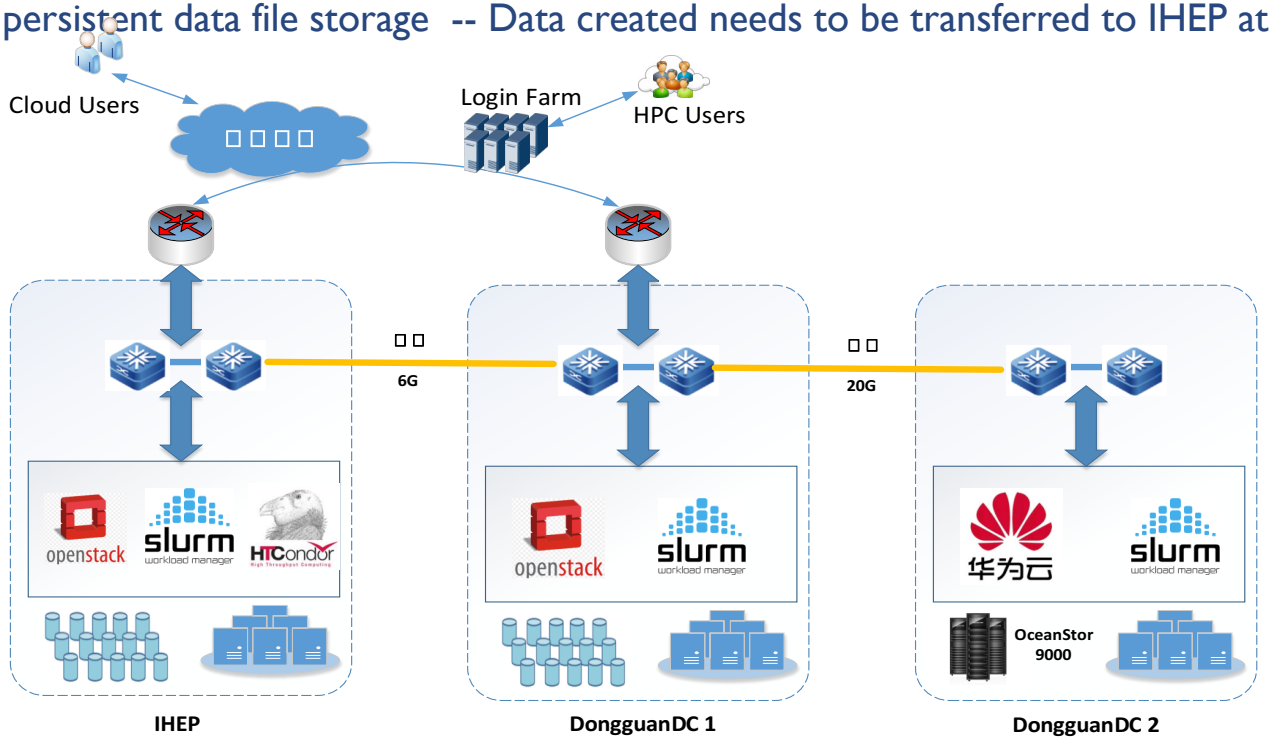
Case 1: the IHEP Slurm Cluster

- Environment is almost the same as the IHEP HTCondor cluster
  - User namespace: same uid/gid
  - Storage: same shared file system
  - Job running environment: almost same
  - Known job slots idle period
- Submit glidein jobs to Slurm worker nodes during the job slots idle period as root
  - Glidein jobs run as user "condor"
  - Same as the owner of "startd" daemon run in the local HTCondor cluster
  - Jobs access data files from a shared file system



Case 2: Dongguan Data Center (1)

- Two Slurm clusters are runing in the federation mode
  - serve for MPI jobs and corporate with Huawei
  - Heterogeneous resources: Intel x86 CPUs, Arm CPUs and GPU cards
    - No access to worker nodes directly from outside
    - Worker nodes can be access outside
  - 10 Gbps Network connection
  - Site Storage is available
    - 3 PB disk storage
    - Not for persistent data file storage -- Data created needs to be transferred to IHEP at last



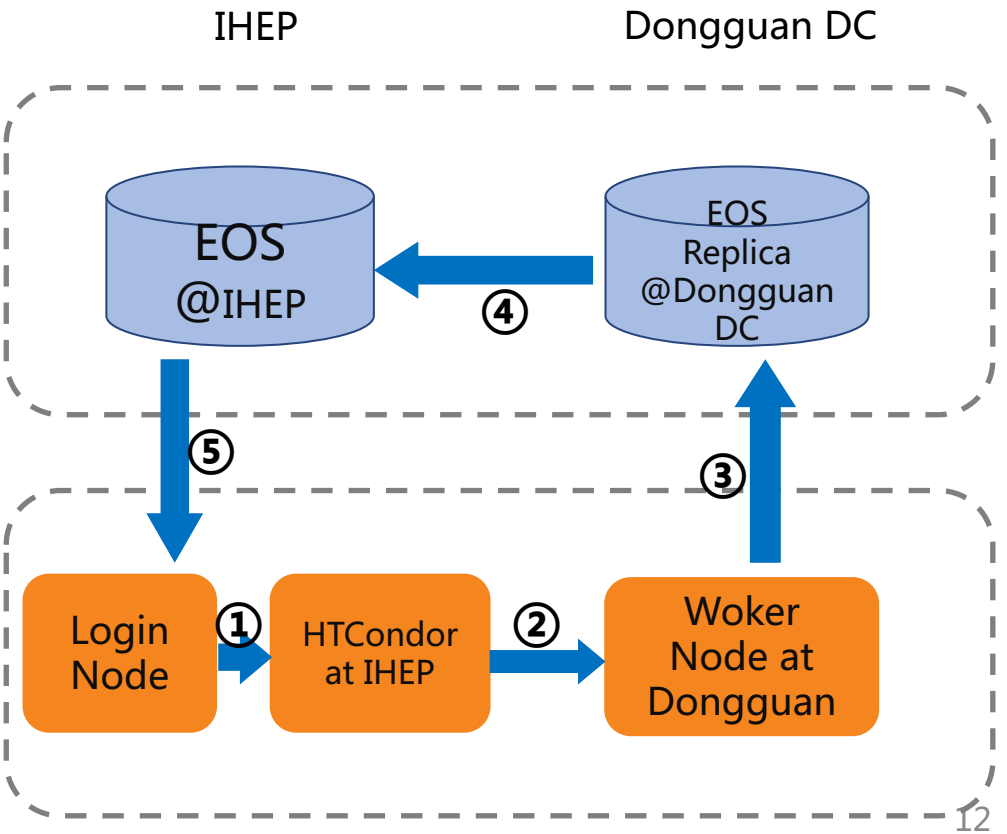
Run LHAASO Jobs at Dongguan Data Center

- Network would be a bottleneck
  - 10 Gbps Network connection but 10K job slots available
- Suitable jobs should be those with small input and output
- Choose WCDA, one detector of LHAASO, data processing
  - 5 types of jobs from WCDA: Corsika → Geant4-step1 →Geant4-step2 → Reconstruction → Analysis
  - The output from this step is the input of the next step
- Better choice:
  - Simulation + reconstruction in one job to reduced the transferred data
    - The output of "Geant4-step1" is replicated back with EOS replica mechanism
    - The output of reconstruction is transferred back on job exit by HTCondor
- Submit glidein jobs as the "lhaaso group" user

	Corsika	Geant4(step1)	Geant4(step2)	Reconstruction	Analysis
input	tiny	middle	large	small	various
output	middle	large	small	small	small
output persistent store?	No	**Yes**	No	**Yes**	No
CPU hours	long	Long	Short	Short	Various
users	Dedicate user	Dedicate user	Dedicate user	Dedicate user	Various user

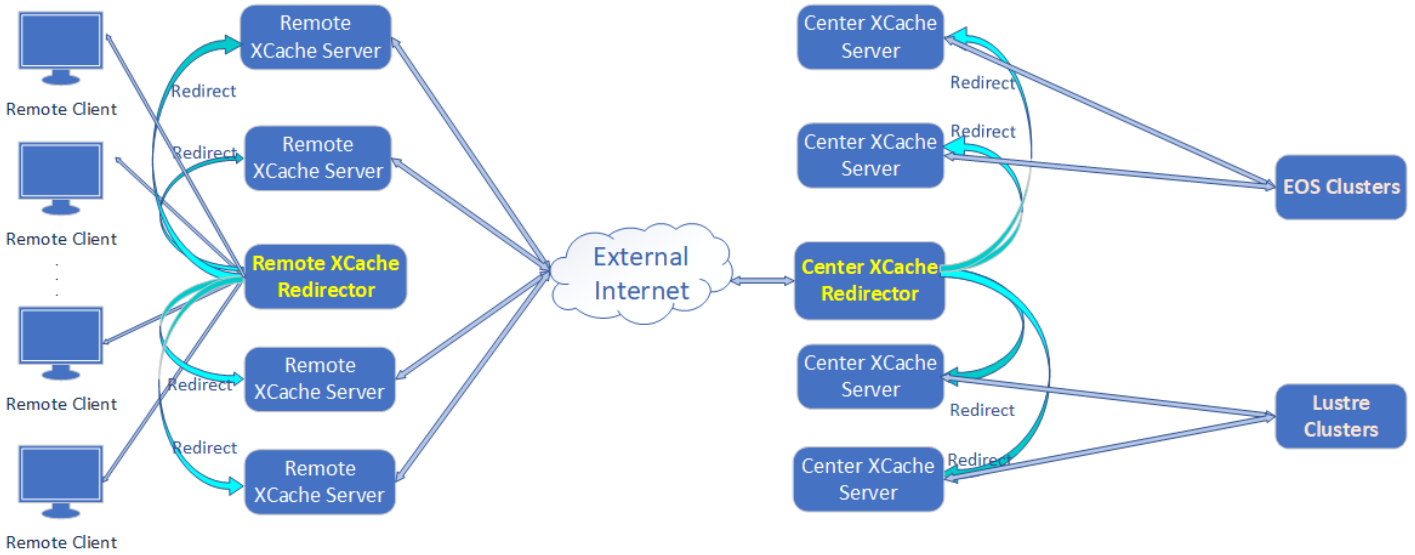
Run LHAASO Jobs at Dongguan Data Center (2)

- Data storage and access
  - EOS replica mode
    - The LHAASO data files created and saved to Dongguan EOS replica
  - Big data files are replicated by EOS asynchronously
  - Small data files are transferred back to IHEP by HTCondor on job exit
- Workflow
  - ① Users submit jobs to the IHEP HTCondor cluster
  - ② Jobs are scheduled to the glidein job slots at Dongguan DC
  - ③ Jobs write big data files into EOS replica at Dongguan DC, while small data files are written into local disks on worker nodes and later ransferred back to IHEP by HTCondor
  - ④ Big data are replicated to IHEP by EOS replica mechanism
  - ⑤ Users get output data files from the IHEP HTCondor cluster



Case 3: Edge Sites

- Edge sites
  - No dedicated storage for LHAASO
  - Small scale
- LHAASO jobs run in glidein job slots
  - Output data are written to local scratch directory of worker nodes
  - Job results are transferred back by HTCondor
- Set up xcache at edge sites
  - Jobs access IHEP storage via xcache
- Suitable Jobs
  - Short queue time
  - schedule to edge sites
    - Less stable worker nodes and less queue time



Current Status

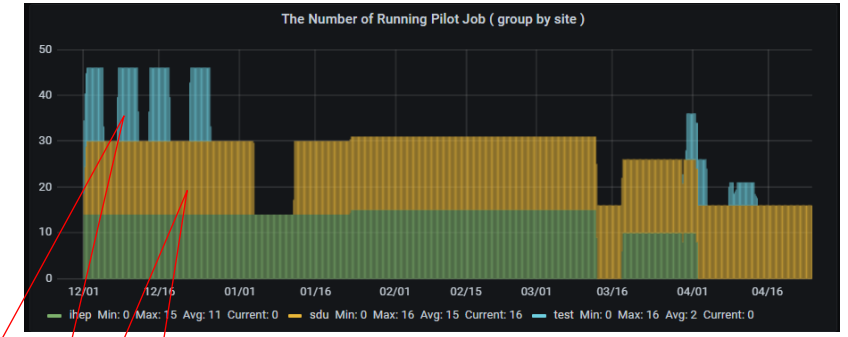
- IHEP Slurm Cluster runs LHAASO jobs at its idle slots
- Tests on Dongguan DC is almost finished
  - Will be in production next month
- 2 edge sites succeed to run LHAASO jobs

```
[shijy@schedd03 result]$ condor_q


-- Schedd: ihep_global_schedd@schedd03.ihep.ac.cn : <202.122.33.35:9618?... @ 05/27/21 17:21:45
OWNER BATCH_NAME    SUBMITTED   DONE   RUN    IDLE   HOLD  TOTAL JOB_IDS
shijy ID: 425      5/27 11:59    865   9091      _     43   9999 425.0-9998

Total for query: 9134 jobs; 0 completed, 0 removed, 0 idle, 9091 running, 43 held, 0 suspended
Total for shijy: 9134 jobs; 0 completed, 0 removed, 0 idle, 9091 running, 43 held, 0 suspended
Total for all users: 9134 jobs; 0 completed, 0 removed, 0 idle, 9091 running, 43 held, 0 suspended
```

Tests on jobs scheduled from IHEP and running Dongguan DC



LHAASO jobs running at edge site

Glidein job running at SDU site

Glidein job running at LZU site

Summary

- A big gap between the requirement and reality of LHAASO computing resources
  - No persistent resources provided from other sites
  - Depends on local HTCondor cluster
- Try to integrate more resources → Local HTCondor cluster + glidein job slots
  - Keep the same usage pattern
  - Three cases with handrolled glidein job slots
    - Local Slurm cluster, Dongguan Data Center and small edge sites
- Thank to the HTCondor team, especially to Greg
  - Got **tons** of help
- Lots of work need to be done in the coming days
  - Glidein job factory
  - Authentication and authorization based on tokens

Thank you!

Question?