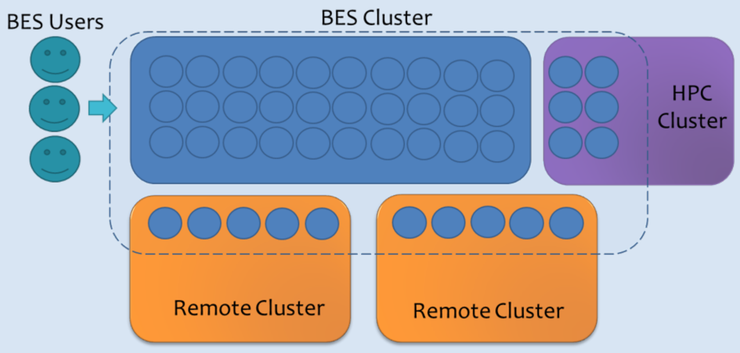
Using Kerberos Token in Distributed Computing System at IHEP

Xiaowei JIANG, Chaoqi GUO, Qingbao HU, Ran Du, Jingyan SHI, Gongxing SUN
jiangxw@ihep.ac.cn, guocq@ihep.ac.cn, huqb@ihep.ac.cn, duran@ihep.ac.cn, shijy@ihep.ac.cn, sungx@ihep.ac.cn

Background

At IHEP, we are building a distributed computing platform to integrate several Chinese sites. However, several long-history experiments are tightly bounded to the local cluster environment. For example, BES experiment, one of the biggest experiments in IHEP,is bounded to the local resources, including DB, storage service, computing service, etc. So we are using a way called 'Cluster Expansion' to expand the local cluster to the distributed computing cluster that makes BES jobs migrated to remote sites nearly transparently.



Considering crossing the different sites, a secure certification method is needed. Because the kerberos token has been used as the main certification method in local computing cluster at IHEP, it's naturally expanded to the distributed computing system.

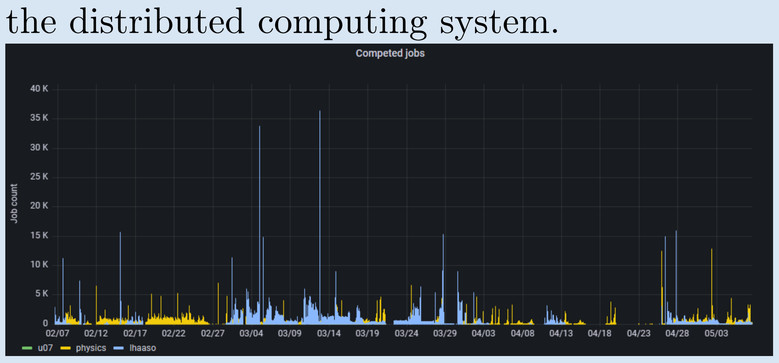
Problems

A kerboros token with a specific validity time is generated when a user logins the submitter node. Before the validity time expires, the token can be used to access other services, for example the AFS and EOS storage service. For the certification solution based on kerboros token at IHEP, the biggest problem is how to extend the validity time. We need to guarantee the kerberos token is renewed in there time points or periods:

1. when a job is submitted
2. when the job is idle in job queue
3. when the job is running.

Conclusion

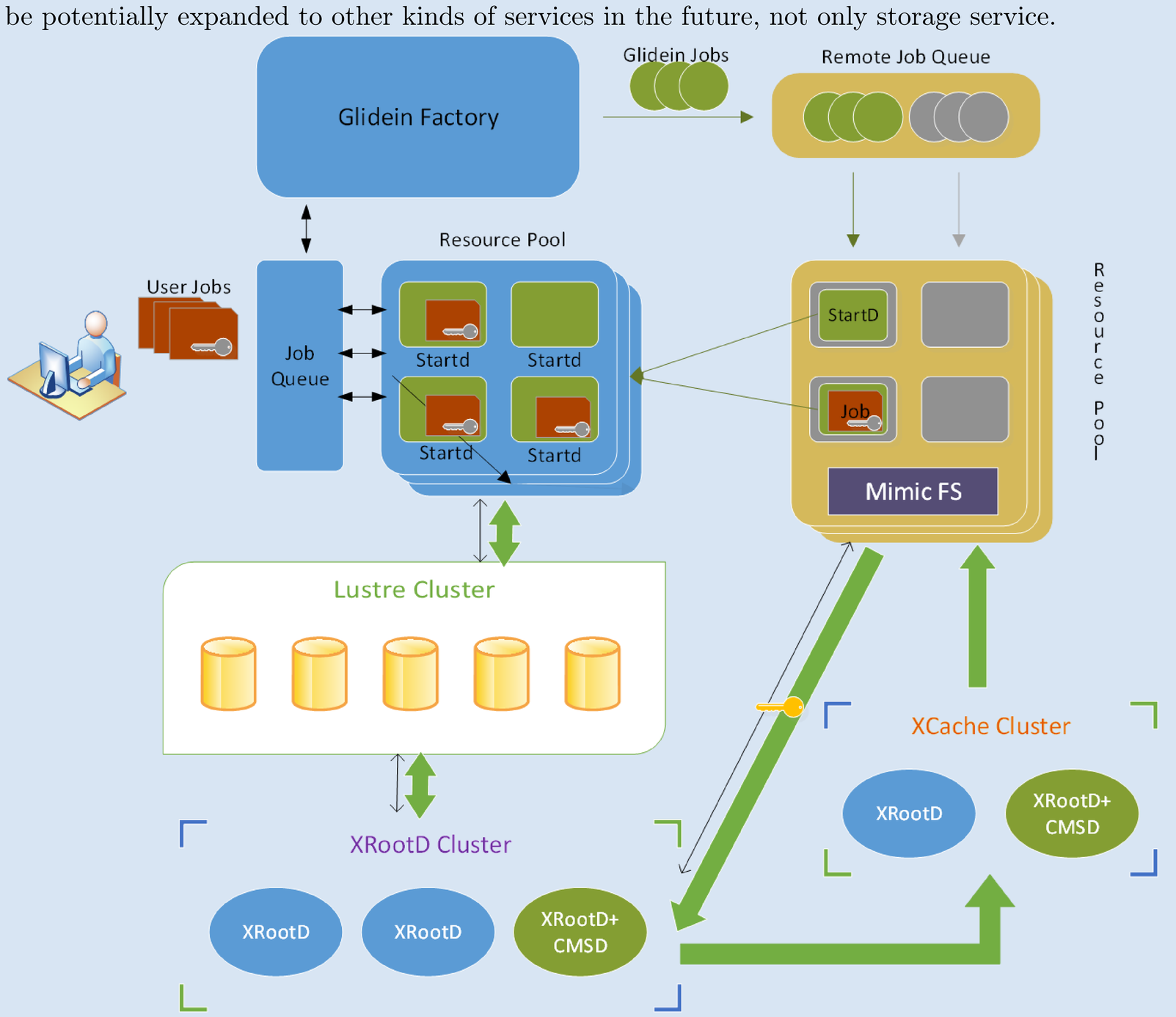
The kerboros token has been used in the distributed computing system at IHEP. With kerboros token, a job can safely access the necessary services from remote side, especially the storage services including Lustre, XCache and EOS. The key problem is how to guarantee the kerboros token valid during the whole lifetime of the job. This paper introduces the solution of extending the validity time of kerboros token in IHEP case. Currently, all the jobs running in the remote sites are using kerboros token to access storage services. In last three months, there are totally 2 million jobs using kerberos token in the distributed computing system.



The solution is also tested and deployed in the ARM worker nodes, and some of the above jobs used kerboros token in ARM environment.

Kerboros Token in Distributed Computing System

As mentioned before, a kerboros token is generated when a user logins the submitter node. At IHEP, in default, 2 days of validity time and 7 days of renew time limit will be given when the kerboros token created. The token is valid before validity time expires and the validity time can be renewed before reaching renew time limit. In terms of computing system, the kerboros token is always used by jobs. On submitter node, the user submits a job and the token is attached as one of job input files. The scheduling system will send the job and the token to a worker node. The worker node can be a local worker node or a remote worker node. This paper is focusing on the remote site and more caring about the operation mechanism on the remote worker node. The kerboros token is stored in the job executing directory and initialized into the job environment when the job arrived the remote worker node. With the kerboros token, the job can access storage services, including Lustre, XCache and EOS in IHEP case (the below figure shows using kerboros token to access Lustre and XCache). As the certification of kerboros token covers the local computing cluster and the remote sites, it can be potentially expanded to other kinds of services in the future, not only storage service.



Kerboros Token Validity Extension

The key problem of using kerboros token in the distributed computing system is how to extend the validity time. The solution in this paper consists of three parts.

1. Extend the validity time when a job submitted. With HepJob, the job submission tool in IHEP, the token can be transferred to the worker node with the job. When the job is submitted, the token validity time will be automatically extended to 2 days again. This function is implemented as a plugin and added into the experiment job workflow over the HepJob. This function can guarantee all jobs have a newly validity time (2 days in IHEP case) when the jobs submitted. Besides, the function can also copy the token to the token repository, which is only a common directory in this case. The token repository will be referred in the following step.

2. Periodically extend the validity time when a job sits in job queue. A job would be sit in idle stat for a long time when the resources are busy. In an extreme scenario, the idle time can be more than 2 days. That means the job will take a expired token to the worker node. The expired token can not be fixed and used by the job any more and the job will exit with an error. So the token should be continuously renewed when the job is idle in job queue. We implemented a simple service to watch on the token repository and periodically renew all the tokens in the repository. This can make all tokens are newly before the jobs are sent to worker node.

3. Periodically extend the validity time when a job is running on the worker node. When a job starts running on a worker node, it would run for a time more than 2 days. That means the token will be invalid when a job runs for more than 2 days and the job will fail. A simple tool called 'token engine' is implemented and launched on the worker node when a job starts running. The token engine starts a service and keeps watching on the kerboros token which is stored in a specific directory and initialized in the job environment. Periodically, 1 hour of time interval in this case, the token engine enforces to renew the token until the renew time limit reached or the job completed. That can guarantee the token is valid during the job runs on the worker node.