

# Grid Architecture and Implementation for ALICE Experiment

Jin HUANG\*, Pablo SAIZ\*\*, Latchezar BETEV\*\*, Federico CARMINATI\*\*, Costin GRIGORAS\*\*,  
Steffen SCHREINER\*\*\*, Jianlin ZHU\*\*\*\*

\*Research Institute 709, China Shipbuilding Industry Corporation, Wuhan, 430070, China

\*\* CERN, 1211 Geneva, Switzerland

\*\*\* Technische Universitaet Darmstadt, Germany

\*\*\*\* South-Central University for Nationalities, Wuhan City, 430074, China

derick0320@gmail.com, pablo.saiz@cern.ch, latchezar.betev@cern.ch, federico.carminati@cern.ch, costin.grigoras@cern.ch,  
steffen.schreiner@cern.ch, dczhou.iopp@gmail.com, zhujianlin1120@gmail.com

**Abstract**—According to the requirement of the Physics Data Challenge and many years of experiences from the actual operations, we propose the Grid Architecture for the ALICE Experiment. This architecture builds on AliEn2 framework and AliEn2 is the grid framework for the whole system's management and also provides the client interface for accessing the ALICE Grid platform. On the base of AliEn2, we adopt some middlewares to make the platform more flexible, availability and efficiency. At the end, the whole system has been worldwide deployed at around 88 sites for distributed production of raw data and Monte Carlo data, detector simulation, reconstruction and real data analysis. We observe the activities of the platform from different point of view and report the performance of the whole ALICE grid platform. The whole Grid Architecture for ALICE experiment plays an important role in High Energy Physics. It successfully manages the resources distributed all over the world and has a great efficiency system to make all peoples from and in different countries working together. The experiments show the whole system works fine and efficient.

**Keyword**— Grid, AliEn, Job Management, Transfer Management, Middleware

Manuscript received May 15, 2013. This work was sponsored by the National Natural Science Foundation of China with Key Grant 11020101060, and the "973" Grant of MOST of China 2013CB837803.

Jin HUANG is with Research Institute 709, China Shipbuilding Industry Corporation, Wuhan, China(Tel:86-13476165997; email:derick0320@gmail.com)

Pablo SAIZ is with CERN, Geneva, Switzerland(email: pablo.saiz@cern.ch)

Latchezar BETEV is with CERN, Geneva, Switzerland(email: latchezar.betev@cern.ch)

Federico CARMINATI is with CERN, Geneva, Switzerland(email: federico.carminati@cern.ch)

Costin GRIGORAS is with CERN, Geneva, Switzerland(email: costin.grigoras@cern.ch)

Steffen SCHREINER is with Technische Universitaet Darmstadt, Germany(email: steffen.schreiner@cern.ch)

Jianlin ZHU is with South-Central University for Nationalities, Wuhan, China(email:zhujianlin1120@gmail.com)

## I. INTRODUCTION

Geneva, 30 March 2010, Beams collided at 7 TeV in the LHC at 13:06 CEST, marks the start of the Large Hadron Collider(LHC) research programme. The A Large Ion Collider Experiment(ALICE) is one of the main experiments in LHC, which is dedicated to the study of nucleus-nucleus collisions.

The ALICE experiment produces large amount of data (peta byte data each year), which requires huge computing power to compute and analysis the data, and Peta Byte order of storage to store raw and analysed data. ALICE uses the grid platform[1] to distribute and analyze all raw data and Monte Carlo(MC) data among the 88 sites that participate in the ALICE Collaboration[2].

In this paper, we give the architecture for the whole grid platform and describe the main components in the system for a better understanding the system. At the final, we give the performance analysis of the platform in different views related with services, jobs, transfers and network since the first LHC run.

## II. GRID ARCHITECTURE

Grid middleware distributions are often large software artifacts, which include a set of components providing a basic functionality. Such capabilities include (but are not limited to) job management, storage management, authentication and authorization, and resource monitoring. Considering the ALICE experiment:

- 1) The resources are complex which are distributed all over the world and the reources are increasing every year .
- 2) The large amount of data (PB) produced by the ALICE experiment every year which is related with Data Management, Transfer Management,etc;
- 3) Each data analysis job has different requirements such as different softwares and datasets which requires Package Management;

4) The security framework has to be considered to ensure Information Security, Monitoring, Authentication and Authorization;

5) Supporting other Virtual Organizations and middlewares.

With the considerations mentioned above, the Grid Architecture designed for ALICE experiment is described in Figure 1. In Figure 1, there are three main layers for the Grid Architecture:

1) **External Software:** These are the basic softwares which are used to build the grid platform. The main framework is based on AliEn2[3] which uses the perl language. AliEn2 is a framework that has been developed by the Alice Collaboration with the aim of offering to the end user a transparent access to distributed computing and storage resources all over the world. LDAP is used to keep the static information about each site, user, service, etc. The relational database is adopted such as mysql, postgresql, oracle, etc. The softwares required by the Jobs are also supported such as AliRoot, Root, etc.

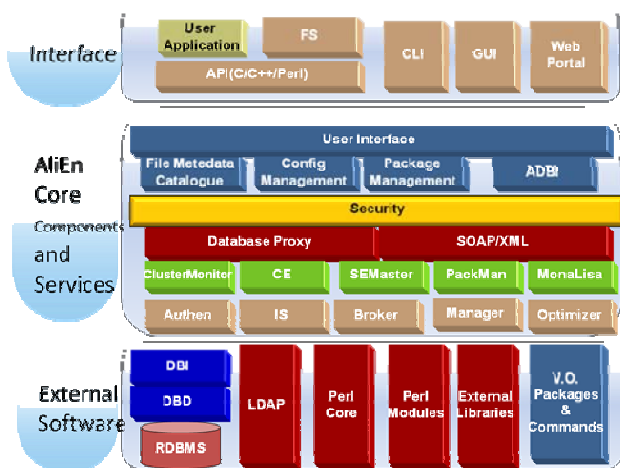


Figure 1. The Grid Architecture for ALICE Experiment

2) **AliEn Core Components and Services:** The core components of AliEn infrastructure[4] are Web Services, which communicate with each other via XML-based messages using SOAP (Simple Object Access Protocol) protocol. The services at AliEn2 are categorized into two aspects:

- Central services- Authen, IS, Job/Transfer/JobInfo Manager, Job/Transfer Broker, Job/Transfer/Catalogue Optimizer, SE/ PackMan/Messages Master, Logger, API, FTD.
- Site services- Cluster Monitor, PackMan, CE, SE, MonaLisa. JobAgent service is running on each worker node in the site.

All the public information about the Central Services and Site Services are published through the LDAP.

3) **Interface:** The Grid Architecture provides different interfaces for the users to use the resources with different purpose. Such as through the client of AliEn2, users could submit/cancel/resubmit the jobs, add/delete/backup the files.

Though the web portal of monitoring system, users could check the system running information in any place any time. With the plugin mechanism, the middlewares could be supported easily.

The required Grid functionality is provided via a combination of the open source softwares, the common middlewares offered by many Grid projects, and the ALICE-specific services from AliEn2.

**A. Middlewares**

There are some important middlewares that the ALICE Grid Architecture adopts and have already been deployed in the sites all over the world.

1) **VOBOX[5]:** It focus on the managing the services deployed inside it and softwares for the data analysis, providing renew proxy service for providing long-life proxies. The mechanism in VOBOX is that a proxy renewal service runs for automatic renewal or registering proxies via the MyProxy mechanism to provide long-lived agents and services. the proxy renewal service on the VOBOX uses a setuid root cron job to generate a proxy for the machine, then uses that proxy to authenticate to the MyProxy server in order to renew the registered proxies.

2) **CREAM(Computing Resource Execution And Management service)[6,7]:** It is a service for job management at the Computing Element(CE) level. to which the CE has interface for supporting and managing different batch system. Moreover, a CE implements additional features with respect to the ones provided by the underlying batch system, such as Grid-enabled user authentication and authorization, accounting, fault tolerance and improved performance and reliability.

3) **Xrootd[8,9]:** The xrootd system is a high-performance data server. It is useful to build scalable, fault tolerant and high performance storage systems for POSIX-like data access. Unlike NFS, the xrootd protocol includes a number of features like:

- Communication optimizations, which allow clients to accelerate data transfers.
- An extensive fault recovery protocol that allows data transfers to be restarted at an alternate server.
- A comprehensive authentication framework.
- Peer-to-peer elements that allow xrootd servers to cluster together while still providing a uniform name space.

4) **CERN Advanced STORAGE manager(CASTOR)[10,11]:**

It is a Hierarchical Storage Management system developed at CERN used to store physics production files and user files. Files can be stored, listed, retrieved and accessed in CASTOR using command line tools or applications built on top of the different data transfer protocols like RFIO (Remote File IO), ROOT libraries, GridFTP and XROOTD. CASTOR manages disk cache(s) and the data on tertiary storage or tapes. Xrootd is one of the protocols supported for data transfer and access.

**B. Central Services and Site Services**

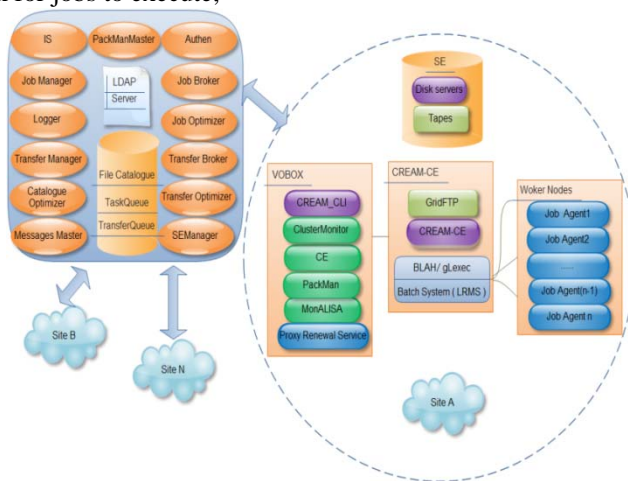
The LHC experiment has four experiments (ALICE, ATLAS, CMS and LHCb) and each experiment is one Virtual Organization (VO). One VO only has one set of Central services and could have many site services.

A typical deployment example in one VO is given in Figure 2 to show how the platform works.

The Central services along with LDAP server, File Catalogue, TaskQueue and TransferQueue take responsible for the management of the whole system and they are deployed at CERN. In the section, we will discuss detailed relationship between these services.

In Figure 2, to manage the local resources in one site, the following components are needed:

1) Site Services (Cluster Monitor, PackMan, CE, MonALISA) for communicating with the central services to ask for jobs to execute;



**Figure 2.** Central services and Site services in a VO

2) VO-BOX for providing a reliable mechanism to the site services to accomplish various tasks (Start/Stop services deployed at each site, renew proxies, installed software management, etc).

3) CREAM-CE for getting the JobAgents from CE and distributing them to worker nodes.

4) JobAgent service running on each worker node for doing some tasks to make sure that the analysis jobs could be executed automatically and successfully, monitoring the status of the analysis jobs, reporting the information about the running jobs and asking more jobs to execute when the job is finished.

5) The Storage Element is used for managing the file-based data. Each site provides certain storage space for the experiment and storage could be tape and disk. In Figure 2, the xrootd protocol is chosen as the transfer protocol for data transfer and data access.

**III. IMPLEMENTATION**

**A. Job Management**

In the system, Job Management is mainly related with Job Manager, Job Broker, Job Optimizer, JobInfo Manger, Task Queue and Authen from Central Services, CE, Cluster Monitor, JobAgents in one site.

The whole workload management system is based on a central job queue which holds all tasks that have to be executed. The Job Broker model is based on pilot jobs: the system submits generic pilots to the computing centres batch gateways, and the assignment of a real job is done only when the pilot wakes up on the worker node. The model facilitates a flexible fair share user job distribution.

A Computing Element (CE) is some set of computing resources localized at a site. A CE includes a Grid Gate (GG), which acts as a generic interface to the cluster; a Local Resource Management System (LRMS), and the cluster itself, a collection of Worker Nodes (WNs) where the nodes the jobs run. The Grid Gate in ALICE is CREAM-CE, which is responsible for accepting jobs and dispatching them for execution on the WNs via the LRMS. The LRMS types are OpenPBS/PBSPro, LSF, Maui/Torque, BQS and Condor.

The jobs in ALICE experiment are described by the Job Description Language (JDL). The users submit the jobs to the Central services, the related services processes the JDL, inserted into the Task Queue and distributed the jobs to the proper sites. Figure 3 shows the *Job Flow Model* of the platform. The main components in the *Job Flow Model* is as follows:

1) TaskQueue is the database with all the Jobs that have to be done. The heart of the job management system is the central TaskQueue which calculates the priorities of the jobs. SITE\_CE is the database with all the jobs detail information running in each site.

2) Job Manager service is responsible for accepting jobs submitted from the users and inserting jobs into TaskQueue, checking the job quota of the user, managing each status of JobAgent and each file generated by the jobs.

3) Job Broker service is in charge of distributing the jobs to each site matching the requirement of the jobs. If the TaskQueue is the heart of the job model, the Job Broker is the mind. The Job Broker is responsible for deviding which job will be executed on site. The decision will be made depending on the data that the job has to analyze, the software packages, time needed by the job and any specific user requirements (name of the site, partition, disk space, etc). The coversation between Job Broker and CE use pull model, and it means that CEs periodically ask the Job Broker whether it has jobs to be executed. If the JDL of the CE matches the JDL of Jobs, Job Broker will send the JDLs of the jobs to that CE. Otherwise, the CE will sleep for a while and then ask the Job Broker again.

4) Job Optimizer service does JDL parameter optimization for all the jobs waiting to be executed. According to the information gathered from the whole system, the Job Optimizer could change the content of JDL and make the jobs easier be matched with the JDL of CE. Along the same lines,

one can also implement policy monitors to enforce VO policies by altering job priorities. the Broker can modify the job's JDL entry by adding or elaborating requirements based on the detailed information it can get from the system like the exact location of the dataset and replicas, client and service capabilities. The job execution model AliEN uses a pull model. In a pull implementation, the Broker does not need to maintain the status of the system.

From the Figure 3, we could see that after the CE in one site get JDL of jobs from the Central services, it will cooperate with CREAM-CE to send the jobs to the worker nodes for execution.

The execution of jobs is based on central services, and the site services act as the interface to the local resources. The site services announce their capabilities, and based on that, the central services will decide which job can be assigned there. According to Figure 4, the whole status of job execution could be described in four stages:

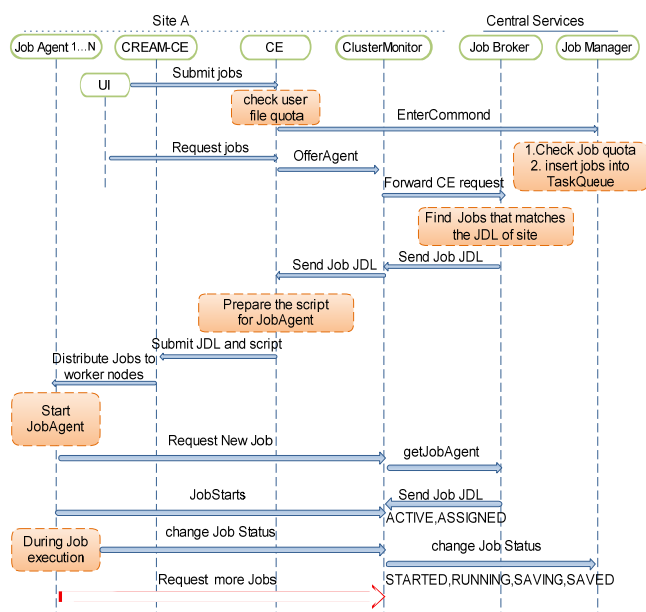


Figure 3. Job Flow Model

1) **Submission:** The user submits jobs to the CE service. The CE service gets the JDL for this job, then reports this job to the Job Manager Service. The Job Manager Service firstly checks whether the user has enough job quota to execute the job, then does some extra preparation work for this job, and update the job status to "INSERTING".

2) **Optimizer:** The Job Optimizer Service is used to optimize the jobs with the "INSERTING" status. The Job Optimizer Service firstly inits the execution environment in AliEn2 client shell for the job, prepares the job token, inserts the job into TaskQueue and updates the job status to "WAITING".

3) **Broker/Agent:** The JobAgent Service asks the ClusterMonitor Service for jobs to execute. The ClusterMonitor sends messages to the Job Broker Services

whether it has the suitable jobs for this site. The Job Broker Service checks the queue status of this site whether it is open or not, finds the suitable jobs which match this site and sends them to the ClusterMonitor Services, and updates the job status to "ASSIGNED". After receiving the jobs from ClusterMonitor, The JobAgent starts to run the jobs. According the jobs' JDL, it creates directories, starts the monitoring process to report the status to ClusterMontior which will send the message to the Job Manager Service. During the execution of the job, the job status could be "STARTED", "RUNNING", "SAVING", "SAVED". At the same time, the JobAgent also includes starting monitor, sending JobAgent status, executing the job, checking process, executing ValidationScript, registering logs, etc. The JobAgent status could be "JOB\_STARTED", "RUNNING\_JOB", "SAVED".

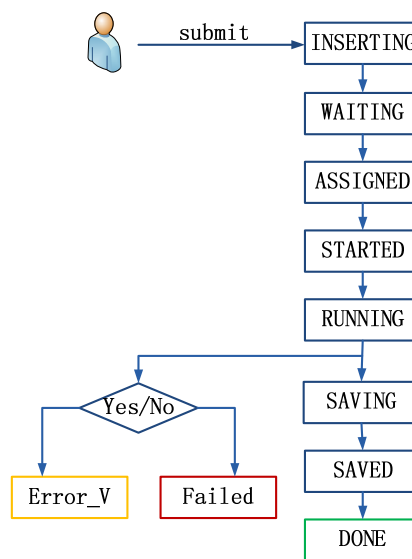


Figure 4. General status job execution diagram

4) **Output registration:** The Job Optimizer Service finds the jobs marked with "SAVED" and "SAVED\_WARN" from TaskQueue, then creates the output information in the user's AliEn client, registers the files in the catalogue, and updates the job stauts to "DONE".

**B. Transfer Management**

With the consideration of safety of the data, resource usage, ALICE group uses different strategies to deal with data taken from the detectors, data from the users, and data from the jobs. In ALICE experiment, one copy of the raw data from the experiment will be stored at CERN(Tier 0) and a second copy will be distributed among the external Tier 1 centres outside of CERN, thus providing a natural backup. Reconstruction to the Event Summary Data(ESD) level will be shared by the Tier1 centres, with the CERN Tier 0 responsible for the first reconstruction pass. Subsequent data reduction to the Analysis Object Data(AOD) level, all Tier 1 and 2 centres will



participate analysis and Monte Carlo production. The Tier 1s will perform reconstruction and scheduled analysis, while the Tier 2s will perform Monte Carlo production and end-user analysis. For all procedures and data analysis mentioned above, the system faces a large amount of data transfer and needs an efficiency Transfer Management to handle all the situations.

The Transfer Management in ALICE experiment includes file catalogue, Transfer Manager, Transfer Broker, Transfer Optimizer, Transfer Queue, FTD and other Storage Management middlewares( Xrootd, CASTOR, etc). The grid platform provides the basic command to store, copy and delete files. Further more, ALICE also provides functions dealing with the multi-copies of the data, the data transfer between different data server and the data access to the files.

The files are identified by a Logical File Name(LFN) in the grid platform. The files also are identified by a Globally Unique Identifier(GUID),which is a fixed-format string generated by the middleware and guaranteed to be absolutely unique. When the user requests a transfer, the LFN, the destination Storage Element(SE) and the type of transfer have to be specified. The JDL syntax for transfer jobs has the following fields: LFN of the file to be replicated; SE where the file has to be replicated; size of the file; Type of replication (Cache,mirror or masterCopy); Requirements of the transfer. The meaning is as follows:

- 1) Cache: the transferred file will not be registered in the catalogue.
- 2) Mirror: the new PFN will be inserted in the catalogue,and any user from the same site will get this new PFN instead of the original.
- 3) masterCopy: the PFN will be registered as the master PFN, and the previous PFN will be mirrors.

In Figure 5, The main components in the **Transfer Flow Model** is as follows:

1) A database( TransferQueue ) with all the transfers that have to be done; the file catalogue stores the connections between the LFN and pointers to any replicas.

2) Transfer Broker service keeps a description in JDL for each FTD. The description includes name of the FTD; SE close to the FTD; free space in the cache. Each FTD sends its JDL to the Transfer Broker, and the Broker tries to match it against the list of transfers that have to be done. If there is an action that the FTD could do, the Broker sends a JDL with the description of what the FTD is supposed to do. If there was no action, the FTD will sleep for a while and request another transfer when it wakes up.

3) Transfer Manager service is responsible for inserting transfer in the system, changing their status and measuring the time each transfer spent in each state.

4) Transfer Optimizer service is responsible for taking all the new requested transfer, and specifying the PFN and SE that currently has that LFN.

FTD supports many protocols,such as xrd3cp,xrdcp,bbftp,cp,GridFTP,GSIFTP,etc. Most of the sites

support xrdcp and xrd3cp. Xrd3cp and xrdcp are able to transfer data files at high speed by using a large number of parallel TCP connections.

According to Figure 6, the transfer status could be categories into four stages.

1) **Mirror**: When the system receives the “mirror” command, firstly checks the permission of the user, checks the path whether the user has permission to use, gets the information about where the source data is kept, reports the transfer to the Transfer Management Service. The Transfer Management Service registers this transfer information into the Transfer Database.

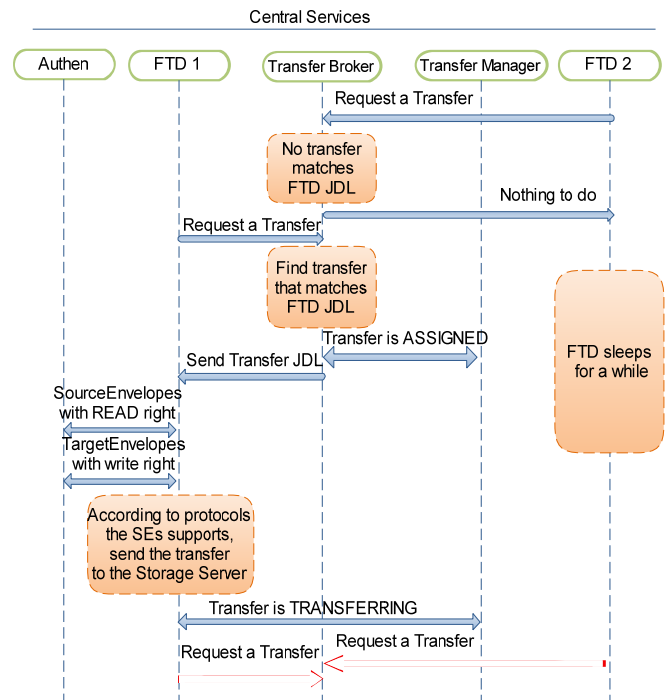


Figure 5. Transfer Flow Model

2) Optimizer: The Transfer Optimizer Service changes the status of this transfer into “INSERTING”. The Transfer Optimizer Service got the information such as SE, guid, LFN,etc, then finds the common protocols for this transfer, and updates the status of transfer into “WAITING”.

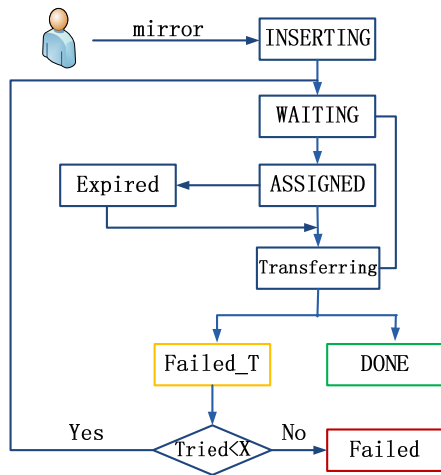


Figure 6. General Status Transfer diagram

3) **Broker:** The Service FTD requests the transfer from the Broker Manager Service, and updates the status of transfer into “ASSIGNED”.

4) **FTD:** The Service FTD got the transfer JDL , asks the Transfer Manager Service changes the status of transfer to “TRANSFERRING”. The Service FTD starts to prepare the envelope information for transferring the files. First the FTD gets user and access information in both source SE and target SE from the Authen Service. The FTD starts the transfer with the setting protocol and asks the Transfer Manager Service to change the status transfer to “DONE” when the transfer is finished.

**C. Packages Management**

AliEn provides a service for automatic installation of the software packages required by the job. The installation, upgrade, configuration and removal of application software packages are done by the AliEn Package Manager (PackMan) service. This service is a helper service that automates the process of installing, upgrading, configuring, and remove software packages from a shared area (software cache ) on a Grid site. The Package Manager Service does not manage the installation of middleware software; it manage the installation of middleware software; it manages the packages that are common for all users of a VO and possibly packages provides by individual users.

The packages are installed on demand, when requested by the Job Agent running on a worker node or upon an explicit request by the VO Software Manager. The Package Manager checks if the requested package is already installed in the cache and if that is not the case, it proceeds with the installation. The right version of the package is downloaded from the File Catalogue and installed in the directory specified as a service configuration parameter. It will also install all the dependencies required by the package. The Package Manager returns a string with a command (a shell script) that the client

has to execute to configure the package and all its dependencies. This script performs actions like setting environment variables. Figure 7 shows which commands use could use to install the packages in the client interface of AliEn2.

The package installation can be triggered by a process running on the worker node or by a person with appropriate privileges (identified by the certificate subject). In both cases the requestor obtains the lease for the package for a specified time. The Package Manager manages the local disk cache and will clear the disk space if it needs the disk space to install newer packages but it will not remove the packages for which someone holds the lease. The maximum lease time for the packages is a configurable parameter. While any user or process can list already installed packages, only the VO administrator can remove a package from the local cache regardless of its current lease status and in that case the currently running jobs requiring that package might fail. Removing a package does not remove the packages that depend on it. If any of those packages are used, the removed package dependency will be automatically installed again.

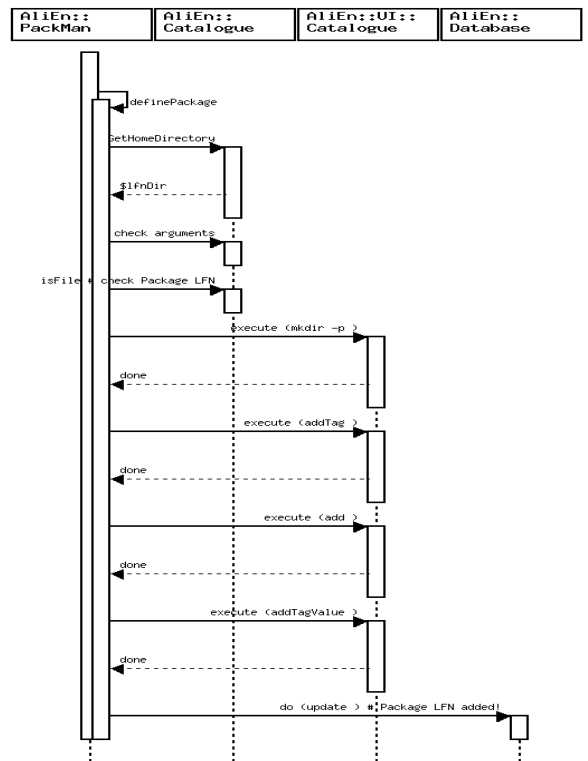


Figure 7. The process of PackMan

**D. Security Framework**

Considering that the resources for the grid platform in ALICE experiment distribute all over the world, the Security Infrastructure for AliEn2 can be divided into five layers. General overview of these levels is described in Figure 8.

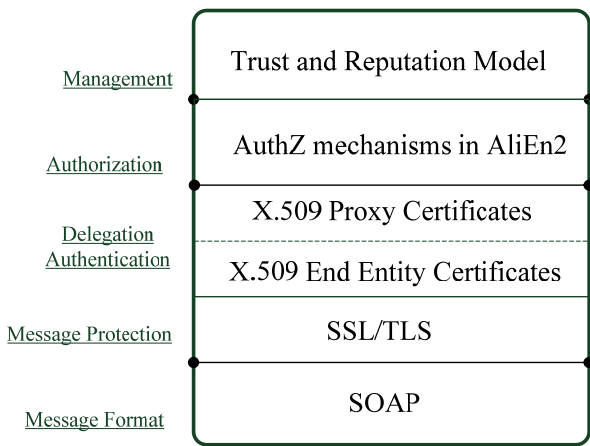


Figure 8. The Security Framework for AliEn2

1) **Layer 1:** Message Format: the message exchanged between different services is based on Simple Object Access Protocol;

2) **Layer 2:** Message protection: the protection for the communication channel for AliEn2 adopts SSL/TLS protocol. SSL/TLS protocol ensures the transport level security. It provides for both integrity protection and privacy (via encryption);

3) **Layer 3:** Delegation/Authentication: The credentials in AliEn2 follows X.509 specification. X.509 EECs are for global authentication. They provide each entity with a unique identifier and a method to assert that identifier to another party through the use of an asymmetric key pair bound to the identifier by the certificate; X.509 proxy certificates are extension to X.509 certificates. GSI proxy certificates and VOMS proxy certificates as a means of authenticating user and delegating authority from users to remote servers. They could bridge disparate security mechanisms, basic support for delegation, policy distribution, single sign-on;

4) **Layer 4:** The Authorization uses AuthZ mechanisms in AliEn2. AuthZ mechanisms are based on the File Catalogue and Quota to authorize the uses to access a certain set of resources.

5) **Layer 5:** Management: there could be many different and effective managements integrated into this layer, such as Trust and Reputation management could be designed for improving the efficiency of the AliEn2. Monitor system is to monitor the resource usage. Trust management system to create, negotiate, and manage trust between other systems or "strangers".

The detail implementation of the information security and Authentication could see the paper[12]. The authorization for file catalogues, files transfer could check paper[13].

IV. EXPERIMENTAL RESULTS

For ALICE experiment, we use MonALISA to monitor the whole grid platform. MonALISA[14] is a globally scalable framework of services to monitor and help manage and optimize the operational performance of Grids, networks and running applications in real-time. The information starts from March 20,2010. The information provided by MonALISA could be seen from the statistics listed as follows.

Figure 9 is the entrance of MonALISA monitoring system for ALICE experiment and it shows the MonALISA Repository for ALICE experiment and the MonALISA collect more than 1700 metrics.

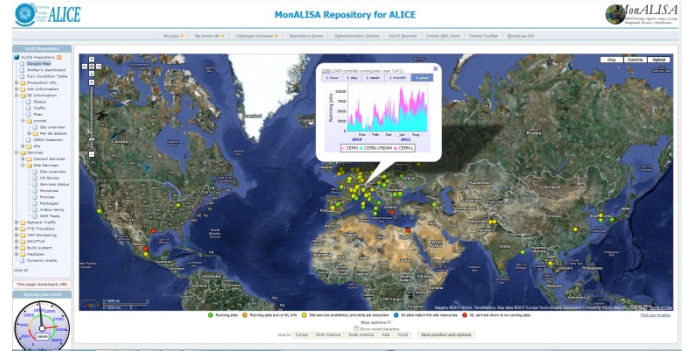


Figure 9. MonALISA Repository for ALICE

Figure 10 is the number of running jobs every day in each site. The peak number of jobs which are running at the same time in the grid system is 32538.

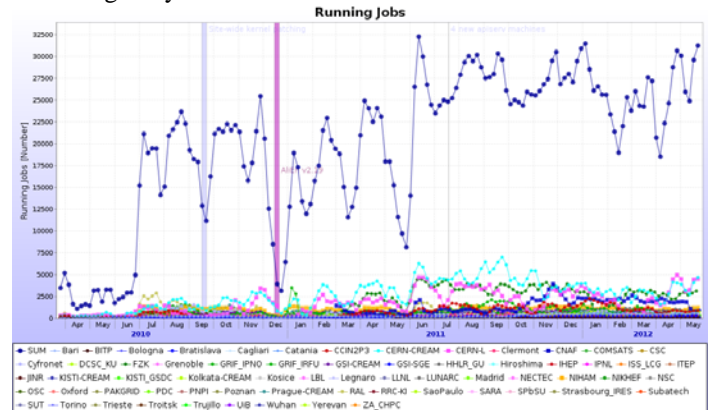


Figure 10. The total number of running jobs per day

Figure 11 describes the RAW data collected from the detectors of ALICE experiment. This is the real data which was recorded during the beam-beam collision and the maximum size of the RAW data is close to 6PB. From the Figure 11, the update of the softwares has been done during the closure of detectors.

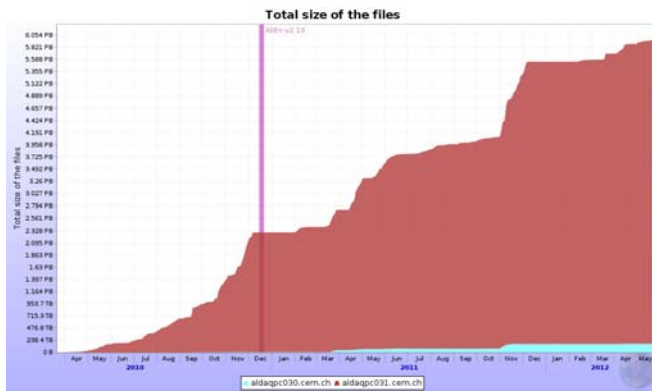


Figure 11. Raw data taken from detectors of ALICE

Figure 12 shows the network traffic of some SEs. The maximum incoming traffic is 87.51 MB/s and the maximum outgoing traffic is 2.341 MB/s. The central services will arrange the transfer jobs according to the traffic size of each SE which could avoid the traffic bottleneck.

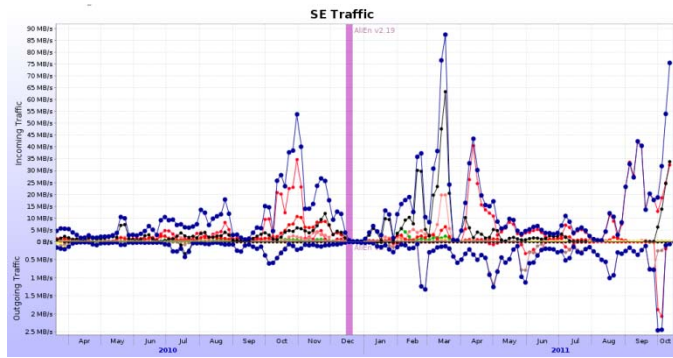


Figure 12. Incoming Traffic and Outgoing Traffic of SEs

There are 8 tape storage elements(ALICE::CCIN2P3::TAPE, ALICE::CERN::CASTOR2, ALICE::CERN::TOALICE, ALICE::CNAF::TAPE,ALICE::FZK::TAPE, ALICE::NDGF::DCACHE\_TAPE,ALICE::RAL::TAPE,ALICE::SARA::DCACHE\_TAPE) which are used for RAW replication. Figure 13 shows the transfer rate for each tape storage element and the maximum rate is 220.5 MB/s.

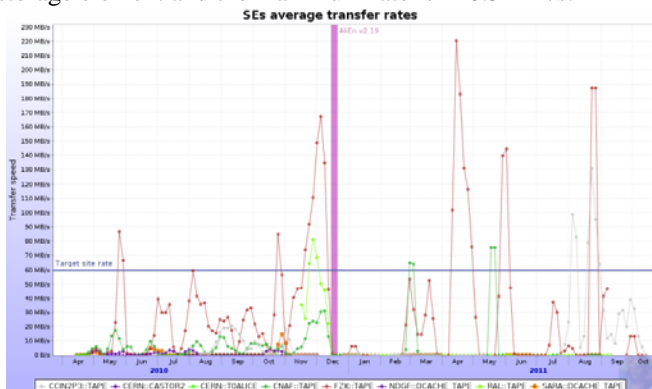


Figure 13. The transfer rates of tape storage elements

There are 54 disk storage elements and 8 tape storages in grid platform of the ALICE experiment. In Table 1, total size of all the storage elements are 43.67PB and 18.344 PB is used. There are 187,720,276 files saved in the SE.

TABLE 1. DIFFERENT TYPE OF STORAGE ELEMENTS

| Storage Type | Total size | Used size | Number of files |
|--------------|------------|-----------|-----------------|
| Disk storage | 12.87 PB   | 6.454 PB  | 165,434,307     |
| Tape storage | 30.8 PB    | 11.89 PB  | 22,285,969      |

There are many other functions to provide information to site administrators to understand the system and tuning their system to get a better performance. Such as there is a Bandwidth test function in the network traffic, choosing one site, you could get the related information (i.e. site name, speed, hops, RTT, etc) in traffic in and out views with other sites ranked by speed.

Until now, about 88 sites join the ALICE grid platform and contributes the resource to the experiment. Around 43 high performance servers provide central services. We have about 54 disk storage elements and 8 tape storages elements the size of which are around 43.67 PB.

V. CONCLUSION

The whole Grid Architecture for ALICE experiment plays an important role in High Energy Physics. It successfully manages the resources distributed all over the world and has a great efficiency system to make all peoples from and in different countries working together.

A lot of experiences are got from operating the grid system. As the increasing number of users and resources, there are many works on-going:

- Developing interface to more middlewares.
- Developing the Java version AliEn2 platform[15].
- Developing Trust Model for reliability and efficiency.
- Optimizing the whole grid platform.

ACKNOWLEDGMENT

I would like to thank the entire members of ALICE experiments and CERN who do the contribution to the development and deployment of the Whole system. Thanks to the ALICE offline group for giving me the chance to join the development of AliEn2 project and helping me in many areas.

The work is supported by the NSFC ( Key Grant 11020101060, IRG11221504 and 11005044), the CCNU11Z01001(Key Grant), the Program of Introducing Talents of Discipline to Universities of China (B08033) .

REFERENCES

[1] The ALICE Collaboration, ALICE: Technical Design Report of the Computing, Printed at CERN June 2005, ISBN 92-9083-247-9.



[2] (2011)The ALICE Collaboration website. [Online]. Available: <http://aliweb.cern.ch/>

[3] (2011)The AliEn2 website. [Online]. Available: <http://alien2.cern.ch/>

[4] S Bagnasco et al, "AliEn: ALICE environment on the GRID", Journal of Physics: Conference Series, Volume 119, Part 6, 062012, 2008

[5] P.M. Lorenzo, IT/GS, CERN, "VOBOXes and LCG Services" in Alice T1/T2 Tutorial for Site Admins, CERN, Geneva, 26-27 May 2009.

[6] Cristina Aiftimier, etc, "Design and implementation of the gLite CREAM job management service", Future Generation Computer Systems, Volume 26, Issue 4, April 2010, Pages 654-667.

[7] (2011)The CREAM website. [Online]. Available: <http://grid.pd.infn.it/cream/>

[8] Dorigo A, Elmer P, Furano F and Hanushevsky A, "XRootD/TXNetfile: a highly scalable architecture for data access in the ROOT environment", Proceedings of the 4th WSEAS International Conference on Telecommunications and Informatics (World Scientific and Engineering Academy and Society (WSEAS)), Wisconsin, USA, 2005.

[9] (2011) The Xrootd website. [Online]. Available: <http://xrootd.slac.stanford.edu/>

[10] J. P. Baud, B. Couturier, C. Curran, J. D. Durand, E. Knezo, S. Occhetti, O. Barring, "CASTOR status and evolution", Computing in High Energy and Nuclear Physics, 24-28 March 2003, La Jolla, California.

[11] (2011) The CASTOR website. [Online]. Available: <http://castor.web.cern.ch/>

[12] J. Zhu, P. Saiz, F. Carminati, L. Betev, D. Zhou, P. M. Lorenzo, A. G. Grigoras, C. Grigoras, F. Furano, S. Schreiner, O. V. Datskova, S. S. Banerjee and G. Zhang, Enhancing the AliEn Web Service Authentication, 2011 J. Phys.: Conf. Ser. 331 062048.

[13] S. Schreiner, S. Bagnasco, S. S. Banerjee, L. Betev, F. Carminati, O. V. Datskova, F. Furano, A. G. Grigoras, C. Grigoras, P. M. Lorenzo, A. J. Peters, P. Saiz and J. Zhu, Securing the AliEn File Catalogue – Enforcing authorization with accountable file operations, 2011 J. Phys.: Conf. Ser. 331 062044.

[14] (2011)The MonALISA website. [Online]. Available: <http://alimonitor.cern.ch>

[15] (2011)The JAliEn website. [Online]. Available: <http://jalien.cern.ch/>

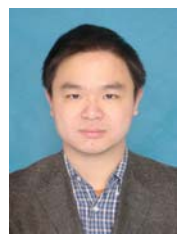
"Monitoring and control of large systems with MonALISA", "Monitoring, accounting and automated decision support for the alice experiment based on the MonALISA framework".



**Dr. Federico Carminati.** He obtained the Italian doctors degree in High Energy Physics at the University of Pavia in 1981. After working as an experimental physicist at CERN, Los Alamos and CalTech, he was hired at CERN where he has been responsible for the development and support of the CERN Program Library and the GEANT3 detector simulation MonteCarlo. From 1994 to 1998 he has participated in the design of the Energy Amplifier under the guidance of Prof. C.Rubbia (1984 Nobel Physics Laureate) in the development of innovative MonteCarlo techniques for the simulation of accelerator driven fission machines, and of the related fuel cycle. In January 1998 he has joined the ALICE collaboration at LHC assuming the leadership of the ALICE software and computing project. Since January 2001 he is holding the position of Work Package Manager in the European DataGRID project. He is responsible for the High Energy Physics Application Work Package whose aim is to deploy large scale distributed HEP applications using the GRID technology.



**Dr. Costin Grigoras.** He obtained his bachelor and master degree in computer science from Politehnica University of Bucharest, Romania in 2004. He obtained his PhD degree in computer science from Politehnica University of Bucharest, Romania in 2009. He is a fellow at CERN from March, 2006 to present. He is currently focusing on distributed systems monitoring, automated decision taking and optimizing databases and code to handle high volumes of data. He is also interested in parallel processing, multicore-aware applications and new technologies in general.



**Dr. Jin Huang,** obtained his Ph.D degree in Software Engineering from Huazhong University of Science and Technology, China. He obtained his BSc from Huazhong University of Science and Technology, MSc from Huazhong University of Science and Technology, China. He got a University Scholarship for one year study at CERN from 2009 – 2011. He was a visiting student of HKUST from September, 2008 to March, 2009.



**Mr. Steffen Schreiner.** He is a PhD candidate in Technische Universitaet Darmstadt, Germany. His research area is security of the distributed system. The published papers include "Enhancing the AliEn Web Service Authentication", "Securing the AliEn File Catalogue", "A tool for optimization of the production and user analysis on the Grid" at CHEP2011, "A security architecture for the ALICE grid services" at ISGC2012, "A Mediated Definite Delegation Model allowing for Certified Grid Job Submission", etc.



**Mr. Pablo Saiz.** He is a fellow of CERN. His research area is grid systems. The published papers include "AliEn: ALICE Environment on the Grid", "AliEn resources brokers", "Alienfs-a linux file system for the alien grid services", "Experiment Dashboard: the monitoring system for the LHC experiments".



**Jianlin Zhu,** Yancheng City, Jiangsu Province, China. November 20, 1981. She obtained her Bachelor degree in Computer Science from School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan City, China in 2004. She obtained her Master degree in Computer Architecture from School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan City, China in 2007.



**Dr. Latchezar Betev.** He is working in the offline team of the ALICE collaboration at CERN and is responsible for the operation of the grid infrastructure of the experiment. His main interests include large-scale distributed computing and monitoring and control of remote systems. The published papers include

She is currently a Ph.D student in College of Physical Science and Technology, Central China Normal University, Wuhan City, China. Her research work focuses on grid platform, security of distributed system, program analysis.