Contribution ID: **106** Type: **Poster contribution**

An efficient method for fine-grained access authorization in distributed (Grid) storage systems

Wednesday 1 March 2006 18:30 (20 minutes)

The ARDA group has developed an efficient method for fine-grained access authorization in distributed (Grid) storage systems. Client applications obtain “access tokens” from an organization’s file catalogue upon execution of a file name resolution request. Whenever a client application tries to access the requested files, the token is transparently passed to the target storage system. Thus the storage service can decide on the authorization of a request without itself having to contact the authorization service.
The token is protected from access and modification by external parties using public key infrastructure. We use GSI authentication for identification to the catalogue service and to storage I/O daemons. The authorization system is as secure as GSI authentication and public key infrastructure can be. To improve the performance for the catalogue interaction, we use GSI authenticated sessions between client and server: after an initial full GSI authentication we encrypt every interaction between client and server with a dynamic symmetric key and achieve a 20 times faster performance.

The main information inside an authorization envelope are the TURL to be used by I/O daemons, the permissions on that TURL, which are ‘read’,’write’,’write-once’ and ‘delete’, the lifetime of that token, the certificate subject and the storage system name for which this token was issued. One token can contain the authorization for a group of files.

Traditional approaches use proxy->uid mapping services to apply local filesystem permissions. In a direct comparison an access token is equivalent to a VOMS proxy certificate who’s proxy extensions authorize access to only one file or a group of files. However VOMS is not the appropriate system to perform authorization on file level since the issue time for such an envelope is very critical (in our implementation only few ms per access) and the VOMS integration, a VOMS server would need to be directly connected to the used file catalogues.

Our method is well applicable in situations, where every GRID user needs to have the possibility to declare a file as private to him.
The same would require in the traditional approach already one worldwide configured UID per VO member, which is very difficult to maintain if not impossible. In our implementation user roles and groups are completely virtualized through definitions in a file catalogue and do not need the one to one correspondence of roles and groups in storage systems.
In the future virtual machines might be the solution for a virtual user concept, but they are still far from deployment in the present Grid infrastructure.
Permissions in the catalogue must be attached to file GUIDs and the catalogue must make sure, that every GUID can be registered only once!

A well performing prototype using the AliEn Grid file catalogue and xrootd as a data server has been implemented. The integration of other catalogue or I/O daemons would be simple. The catalogue service itself can run different file

catalogue plug-ins. The token is moved as part of a file URL, i.e. no I/O protocol changes are needed. I/O daemons need one modification in the ‘open’ command to decrypt the authorization envelope, reject access or replace the initial TURL passed to the open command with the TURL quoted in the envelope. This functionality is encapsulated in a C++ shared library, which allows to define additional authorization rules for certain VOs, certificates or TURL paths.

Author: PETERS, Andreas (CERN)

Presenter: PETERS, Andreas (CERN)

Session Classification: Poster and Demo session + cocktail

Track Classification: Poster session