SlashGrid, HTTPS and fileGridSite

30 October 2002

Andrew McNab, University of Manchester

mcnab@hep.man.ac.uk

Overview

- SlashGrid framework
- certfs filesystem
- GACL Access Control Lists
- HTTPS
- fileGridSite HTTPS server
- fileGridSite examples with curl
- curlfs for SlashGrid
- "G-HTTPS"
- HTTP(S) as data protocol too?
- Summary

SlashGrid ("/grid") framework

- SlashGrid is a framework for making "Grid-aware" filesystems
  - in particular, filesystems where Grid credentials not Unix UID or GID determine access to files
- Individual filesystems are provided by dynamically loaded plugins
  - configuration in /etc/fstab
- Filesystems so far: certfs, gmapfs, httpfs and curlfs
- SlashGrid verifies and manages the Grid credentials associated with a UID and makes this available to plugins in an efficient way.
- Currently uses GSI proxies, full X509 certificates or gridmap (grid-mapfile or Pool account gridmapdir)
- Will also pull information from LCMAPS in future versions.

certfs filesystem

- In TB 1, pool accounts create files as the Pool UID
  - If Pool UID is recycled, files are now owned by new user of that UID!
- With certfs, file access is controlled by per-directory or per-file ACL's in terms of certificate subjects.
- This means that if my UID is reused, the new user can't access my files because their credentials don't match the ACL.
- If I come back with a different UID, I do match the ACL since my Grid credentials are the same, and I can read my old files.
- certfs uses GACL ACL library, so ACL's can also include VOMS or LDAP VO groups, CAS objects or other credentials supported by GACL in the future.
- certfs has been stress tested: eg you can build a bootable Linux kernel in a directory hierarchy on a certfs filesystem (~100,000 operations?)

GACL Access Control Lists

- XML format for controlling read, list, write and admin access
- Can specify DN, LDAP VO group, VOMS group or CAS objects
  - Can easily be extended to other credential types
- C libgacl is provided to manipulate ACL objects in memory
  - permissions (eg write), credentials (eg a DN), entries and whole ACL's.
  - access functions allow you to construct and test ACL components
  - aim for efficiency since ACL may need to be evaluated repeatedly
  - working on Java implementation of the same API
- Aim to standardise some form of GACL ACL's through GGF Authorisation WG.
- GACL API intended to insulate applications from changes to XML representation.

HTTPS

- HTTPS is an interesting and important protocol for several reasons:
  - it is by far the most widely deployed secure protocol
  - has a large amount of high quality software that we could leverage
  - has excellent interaction with Firewalls, Network Address Translation and Application Proxies
    - has the potential to solve some of the problems sites have with private IP farms
  - along with HTTP, is the basis for Web and Grid Services
- HTTPS consists of HTTP/1.1 over an SSL connection
  - security done by SSL layer, using X509 certificates (including GSI)
- HTTP/1.1 (rfc2616) and extensions like WebDAV (rfc2518) have a rich set of methods (GET, PUT, DELETE, COPY etc) headers ("Expires:" etc) and Errors ("413 Request Entity Too Large")
  - so a standard way exists for many of the transfer operations we need

fileGridSite

- Read (GET) well supported by HTTPS servers.
- However, write (PUT and DELETE) usually left to CGI programs, servlets etc.
- Access control also limited to client IP or HTTP passwords.
- fileGridSite adds Grid authorisation and write operation support to Apache
  - a cut-down version of GridSite (used for https://marianne.in2p3.fr)
  - file rather than webpage orientated (no fancy headers on HTML etc)
  - uses GACL to handle the Access Control Lists
  - can work with mod_ssl-GSI so clients can authenticate with a GSI proxy
- **Turns an Apache webserver into a Grid HTTPS fileserver with the key functionality of a GridFTP server.**

fileGridSite examples with curl

- Curl is a standard HTTP/HTTPS command line client (cf wget)
- Get a file using GSI proxy in /tmp/x509up_u100
  - `curl --capath /etc/grid-security/certificates/ --cert /tmp/x509up_u100 https://a.b.com/example1.txt`
- Copy a file to the fileGridSite server with HTTP PUT:
  - `curl --capath /etc/grid-security/certificates/ --cert /tmp/x509up_u100 --upload-file /tmp/example2.txt https://a.b.com/example2.txt`
- Delete a file with HTTP DELETE:
  - `curl --capath /etc/grid-security/certificates/ --cert /tmp/x509up_u100 --request DELETE https://a.b.com/example2.txt`
- Create a directory with PUT to …/
  - `curl --capath /etc/grid-security/certificates/ --cert /tmp/x509up_u100 --request PUT https://a.b.com/newdir/`

curlfs for SlashGrid

- curl is built on top of a general library, libcurl
  - handles persistent HTTP and HTTPS connections, SSL setup etc
- To add HTTP and HTTPS filesystems to SlashGrid, have made a libcurl filesystem plugin: curlfs
- This maps parts of the URL space into the local filesystem:
  - https://a.b.com/newdir/ ---> /grid/https/a.b.com/newdir/
- Works with any standard HTTP or HTTPS server
  - rpm -i /grid/http/datagrid.in2p3.fr/distribution/globus/beta-21/RPMS/*
- SlashGrid framework provides GSI proxy or full cert/key to curlfs so it can make authenticated requests.
- Write with HTTP/1.1 PUT and DELETE being added to curlfs
  - Will complement fileGridSite support for these on server side

"G-HTTPS"

- A proposal for backwards compatible extensions to HTTPS
  - being discussed on wp2-sec and wp7-security lists
- Adds GSI proxy delegation to HTTPS using additional methods (eg PUT-PROXY) and headers (eg Delegation-ID)
- Allows services to return generalised metadata in headers or by URL
  - initially this allows services to return the GACL ACL of a response for more efficient caching (ie sharing cached copies with other users.)
- Aim is to avoid breaking existing HTTPS systems and to achieve "pass through" compatibility:
  - even if HTTPS client or server software doesn't understand extensions, they can make them available to the application which does
- Agree common extensions for use by Grid Services people (WP2/WP3) and file access people (WP5/WP6) Then --> GGF.

HTTP(S) as data protocol too?

- HTTP(S) has a large amount of high quality software we can leverage
- It is supported by existing web browsers (email my job output URL?)
- It works well with Firewalls (just one port), NAT (one way connection) and Proxy Servers (Squid etc already available.)
- It can be extended to do "Grid things" like delegation: G-HTTPS
- Should be straightforward to add to Storage Element
  - Much of SE prototyping was done with HTTPS
- It can support application-level multiple streams and striping by using the standard Range: header to set up many partial fetches.
  - A lot of websites run replicated/clustered server farms for HTTP(S)
- Kernel-based HTTP servers like tux and khttpd are very efficient
  - negotiate in HTTPS, then get Redirect: to HTTP transfer URL for data??

Summary

- SlashGrid provides framework for Grid-aware filesystems
- certfs filesystem provides local disk storage controlled by Grid credentials
  - resolves Pool Account recycling problem
- GACL Access Control Lists support DN, VOMS, CAS etc
- fileGridSite HTTP(S) server has been written
  - supports read/write with standard utilities like curl
- curlfs written for SlashGrid: maps URL's into filesystem
- "G-HTTPS" proposal for Grid extensions to HTTPS
- HTTPS may be a viable alternative to GridFTP, even for data
- Source code for SlashGrid, GACL, fileGridSite, curlfs is in EDG CVS
- See http://www.gridpp.ac.uk/authz/ for more details