Dynamic Accounts in TB1.3

What we could do with what we've got now...

Andrew McNab, University of Manchester

mcnab@hep.man.ac.uk

Outline

- What we do in TB 1.2
- Dynamic Accounts details
- SlashGrid
- UID to DN map in SlashGrid
- DN-based home directories
- What we could do in TB1.3
- Benefits of using ACL's
- Grid ACL vs VoMS/CAS
- Get rid of "UID domains"?
- Certfs as native "container" hosting environment

What we do in TB1.2

- VO authorisation provided by LDAP VO services
- mkgridmap used to periodically constructs grid-mapfile
  - entries have "." as local user to invoke Dynamic Accounts patch
- When a job or gsiftp session comes in, patched gridmap.c associates user's DN with one of the pool of Dynamic Accounts.
- Dynamic Accounts have Unix group membership, quotas etc set by local admin in normal way.
- For jobs, Globus passes job to PBS etc. and the job is executed on a worker node as the pool user, using normal Unix permissions.
- Home directories and grid-map files are NFS-shared
  - the delegated user proxy is accessible through this.

Dynamic Accounts details

- Auditing possible since all DN=>UID mappings recorded in log files.
- Same pool mappings are shared across a farm by sharing gridmapdir of lock files with NFS.
- Existing system works ok for CPU+tmpfile only jobs.
- But, all files owned by Unix UID of the Dynamic Account.
- This means we cannot trivially recycle Dynamic Accounts
  - need to make sure all files they own are deleted when recycled
  - current Testbed sites only recycle accounts when they reinstall
- Lack of recycling ok while Testbed is small, but want to be able to scale this all up, and leave it running for months/years on end
  - so will need recycling at some point

SlashGrid ("/grid")

- A framework for creating "Grid-aware" filesystems
  - different types of filesystem provided by dynamically loaded (and potentially third-party) plugins.
  - Source, binaries and API notes: http://www.gridpp.ac.uk/slashgrid/
  - RPM's in datagrid.in2p3.fr repository work on RedHat 6.2, 7.x and can cope with you doing a Linux kernel build on a SlashGrid filesystem.
- certfs.so plugin provides local storage governed by Access Control Lists based on DN's, stored in a cache: /var/spool/slashgrid/grid
- Since most ACL's would have just one entry, this is equivalent to file ownership by DN rather than UID.
  - solves admin worries about long lived files owned by dynamic accounts.
  - if dynamic accounts are prevented from writing to normal disks, then no chance they will write something unpleasant somewhere unexpected.

UID to DN map in SlashGrid

- This is done using gridmap mechanism if possible.
  - Uses standard Globus call / shared library, so local choice of Dynamic Accounts is used.
  - Since Testbed sites share grid-mapfile/gridmapdir via NFS, can access this on worker nodes.
  - Doesn't rely on a credential a user process could fake (eg nominating someone else's proxy in their home directory.)
- If this fails, then a proxy /tmp/x509up_uNNN created by grid-proxy-init is looked for.
  - This allows interactive use of SlashGrid filesystems.

DN-based home directories

- gmapfs plugin provides a virtual directory apparently populated with symbolic links.
- gridmap mechanism used to derive mappings.
- The links map usernames to url-encoded directory names
  - /grid/gmap/gpool023 -> /grid/home/O=Grid/O=UKHEP/OU=hep.man.ac.uk/CN=Andrew%20McNab
- This means Unix passwd file can list fixed home directories for Dynamic Accounts.
- But the symbolic links will change in step with DN to UID mapping.
- SlashGrid can work on top of NFS, so can use NFS-shared root-owned /var/spool/slashgrid/grid
- SlashGrid daemon on the node maps this onto /grid, subject to ACL's.

What we could do in TB1.3

- LDAP VO services / mkgridmap still used to construct grid-mapfile
- When a job or gsiftp session comes in, patched gridmap.c still associates user's DN with one of the pool of Dynamic Accounts.
  - gridmap.c creates home directory for user's DN if doesn't already exist.
- Dynamic Accounts have no particular Unix group membership etc.
- Job files etc created in home directory like /grid/home/O=Grid/O=UKHEP/OU=hep.man.ac.uk/CN=Andrew%20McNab
- Globus passes job to PBS etc. and the job is executed on a worker node as the pool user.
- User can read home directory, since SlashGrid can see the mapping in gridmap via NFS, and can access the underlying cache via NFS.
- **All files created are owned by the DN not the UID.**

Benefits of using ACL's

- SlashGrid/certfs means we can replace UID ownership with DN ownership, but get other benefits too.
- GACL library used to manipulate XML ACL's.
- This ACL format let's you specify permissions individually: read, list, write, admin (= modify ACL)
- … and different credentials such as VO groups or VoMS attributes in addition to user DN's.
- This means you can give read or write permission to groups of people, at the directory level.
    - (Intend to support file specific ACL's in the future.)

Grid ACL vs VoMS/CAS

- CAS provides ACL-like feature of specifying what action (eg write) is permissible on an object (eg tau-wg-montecarlo).
- (If using lots of subgroups within a VO, could achieve much the same thing: eg define a group of people in tau-wg-montecarlo-write)
- In some cases, this could be used to provide ACL functionality.
- However, it is too coarse grained and too heavyweight for all contexts
  - eg if my job creates a temporary, working directory in /grid/tmp, I don't want to setup a new entry on the central CAS machine to control this.
- The two systems should be seen as complementary
  - when you create some tau Monte Carlo, put it somewhere the ACL gives write access for people with "tau-wg-montecarlo write.")
  - when you just create a temporary directory, the ACL defaults to just the creator having admin access.

Future: get rid of "UID domains"?

- Each testbed site currently constitutes a "UID domain" in which DN=>UID mappings must be consistent on all machines.
  - Currently achieved by sharing grid-mapfile or gridmapdir by NFS (or replicating with LCFG)
- This arises from two major components:
  - NFS sharing of disks.
  - Local batch (usually PBS) by default assumes same UID on front and backend machines.
- Would simplify recycling of pool accounts on gatekeeper if didn't need to maintain this consistency:
  - gatekeeper would just allocate a pool UID which had no processes already running
  - if use "gsiftpfs" instead of NFS, then DN=>UID mappings done dynamically on SE etc too - getting rid of NFS is a worthy goal in its own right.
  - but, would need to configure / modify PBS etc to dynamically allocate a UID on backend node and copy proxy?

Certfs as container hosting environment

- Some of the OGSA discussions make distinction between simple (eg native Linux) and container (eg Java or .NET) hosting environments.
- The original motivation for “in a box” environments is security.
- OGSA interest is in creating new services dynamically: this is easier if services are “in a box” to start with.
- Certfs is motivated by desire to keep users from making long lived UID-owned files.
- However, it is also a step towards the kind of dynamic environments OGSA talks about.
- Is the answer to our concerns about security and our desire for flexible, dynamic services, to make Unix UID’s as transitory as Process Group ID’s?

Summary

- Most of the concerns of admins are being addressed to some extent.
- Current VO system is probably sufficient, but CAS would be more flexible.
- Pool accounts are useful but limited by UID file ownership issues.
- SlashGrid / certfs intended to provide solution to this.
- Defining a Grid ACL format deals with other issues too.
- Do this in XML: what format?
- GACL library provides API for handling whatever is decided.
- How far can we go towards make UID's purely transitory?