They called me mad!!!!!!!

The project blogs of Craig Strachan

- Home
- Projects

Automounter moves

November 6, 2014

For a long time, one of the projects on the Services Unit's list of things it would be nice to do if we ever have the time (and we never will) was moving from the AMD automounter to autofs. An even better plan would be to get rid of the need to use auto mounters completely and indeed the promise of this possibility was one of the reasons we adopted OpenAFS as our file system of choice but sadly we still have too many NFS exported file systems in use or this to be practical.Still AMD and our rather baroque set of scripts for generating the automounter maps and pushing them into LDAP seemed to work well so no need to panic. Until it transpires that am-utils isn't available on SL7. So now we have to look at getting autofs working with our setup ASAP. The following are some notes on the subject and some musing about the best course to take:

Firstly, what do we use the automounter for? We currently have 6 automounted file systems: /group, /legacy, /partition, /nethome, /public and /yesterday. Do we still need all of these? Almost certainly not.

/yesterday is a hangover from the days of NFS mounted home directories and has been supplanted by AFS. There are no live entries in this map so it can be safely decommissioned.

/public has two subdirectories, homepages which is the mountpoint for homepages web content and southbridge which was used as a mount for the legacy dai website but which is no longer used for that purpose. This latter subdir can safely be got rid of. Nethome provides the /home/ pathname which is doubtless embedded in many shell scripts and will need to be retained but as we only have AFS home directories now, it should be possible to replace the current complicated process for generating this map with a program map, a simple script which returns the appropriate AFS pathname when given a username. /legacy either points to machines which no longer exist or content which has not changed in at least the last 3 years. This would be an excellent opportunity to get rid of it.

This leaves us with /partition and /group. /partition, which is the centralised mount point for all the School's remaining NFS exports is useful for preventing multiple mounts of the same file system when two automount points exist on the same partition. By making the mounts symbolic links to the /partition mountpoint, we only need to mount the partition once. It's also a shadow of its former self, where once we had well over 100 exported partitions, now the group map, which is the main user of NFS mounts these days, only references 23 unique partitions. It's also easy to convert into a autofs map so we should probably keep it.

/group is the real issue. Although most of /group is in AFS these days, we still have 63 unique NFS mounted directories in the map spread over several submounts and liberally intermingled with the

AFS data. Tricky but we have no choice, we have to retain this map and data.

Things aren't quite as bad as they seem though. Because the AFS group data was created to mirror the structure of the automounted version as closely as possible, and because autofs supports wildcard keys, every AFS directory which has a final AFS pathname component which is the same as its key in the automounter map, and which does not share a submount with an NFS directory, can be dealt with by adding

```
* /afs/inf.ed.ac.uk/group/&
```

to the main group map.This included any submounts which do not include NFS mounts. In other words, we only need submount maps for submounts which hold NFS data. In the old group map, we had 24 submounts. A quick check suggests we will need 10 for autofs

Update 6/11/2014
Most of this was theoretical when I first wrote it but since then, I have a test autofs setup running on my desktop machine and using file maps configured using Kenny MacDonald's autofs component. This lead to the question, do we need to use LDAP to store the maps or should we just use the component? Neither option seemed to have a great advantage over the other and so I took the opportunity to pose this very question at the development meeting yesterday, as well as seeking agreement that we could dispense with the mounts mentioned above. There was not an issue with getting rid of the excess or ancient mountpoints but Stephen pointed out that if we used LCFG to manage the maps, every time a map was changed, a 1000+ profile rebuild would be triggered. I don't expect the maps to change very often but this still seemed like a compelling argument so LDAP it is!

Update 11/12/14
Well many a long furlong has passed beneath the wheels of this project since I last commented on it but there has been progress! The script to push rfe automounter maps into LDAP has been written, Toby has set up a test LDAP server to play with, the partitions map has been pushed into LDAP and autofs on my desktop machine has successfully used this map to mount the partitions. This feels like success!

There is still much to do however and my self-imposed deadline of having this all completed by the end of the year draws ever closer. The following still needs to be done:

- Convert the other amd maps (or at least those we are still going to use) into the new format
- Write the final version of the programatic map for /home, package it up and amend the LDAP script to create an entry for it in auto.master
- package up the LDAP script
- Create a header file to allow the use of autofs to be switched off and on.

Doesn't look too bad at first sight so long as nothing else gets in the way. More news as we have it.

Update 2/6/15
Well just as success seemed assured, a final problem reared its ugly head. To explain it, we need to look at how autofs maps are represented in LDAP. We have a separated branch of the LDAP tree, ou=AutofsMaps,dc=inf,dc=ed,dc=ac,dc=uk where we have two types of object class, automountmap which represents an indirect autofs map and automount which represents an entry within a map. A map entry looks like this:

```
dn: ou=partition,ou=AutofsMaps,dc=inf,dc=ed,dc=ac,dc=uk
ou: partition
objectClass: top
objectClass: automountMap
```

and a map entry like this:

```
dn: cn=ptn188,ou=partition,ou=AutofsMaps,dc=inf,dc=ed,dc=ac,dc=uk
cn: ptn188
objectClass: top
objectClass: automount
automountInformation: mullo.inf.ed.ac.uk:/disk/ptn188
```

It will be seen that before the entry dn can be created, the map dn has to already exist. If it doesn’t, we get an error. Naturally, we take care, when pushing the maps and entries into LDAP, to make sure that the map dns exist before creating the entries. The problem comes when we try to replicate the LDAP directory to a slave server. Our home brewed LDAP replication technology works by querying the ldap master for all changes that have occurred since a given timestamp and applying them one by one, in the order they’re returned. The LDAP docs make it clear that there are no guarantees about the order things are return in by a query meaning that we could potentially get one of the map entries returned before the map itself. As mentioned above, trying to push a map entry into LDAP when the map doesn’t exist would cause the replication to fail and probably require us to restart slapd on the slave server. That’s not so bad for SL7 (though it’s still pretty bad) as we only have a few LDAP servers. But every SL6 machine is a slave server so we’re faced with having to potentially restart slapd on hundreds of machines. Something to be avoided. To muddy the waters further, every time we have actually tried the replication, it has worked. But if the LDAP docs are to be believed, the next time we try it, it might not. So until the last SL6 machine is upgraded, it seems safer to continue to run a separate LDAP server to hold the automounter entries.

Leave a Comment » | | Permalink
Posted by Craig Strachan

- You have searched the They called me mad!!!!!!! blog archives for **‘automounter’**. If you are unable to find anything in these search results, you can try one of these links.
- Search automounter Search
- ## Recent Posts
  - Come in to the lab!
- ## Recent Comments
- ## Archives
  - August 2014
- ## Categories
  - Uncategorized
- ## Meta
  - Site Admin
  - Log out
  - Entries RSS
  - Comments RSS

- WordPress.org

Theme: Contempt by Vault9.