
NIS (alias YP)

Overview

- History
- Short description of each map
- Map building
- Map propagation
- Problems
- Future
- Questions / discussion

History - early Unix

- Each machine standalone
- Each machine had own textual copy of user databases (eg /etc/passwd etc)
- Textual form inefficient for large number of users => DBM format

History - Sun and YP

- Diskless workstations
 - » common environment
- Led to birth of Yellow Pages
 - » one YP master
 - » several YP slaves
 - » many YP clients

Master / slaves / clients

YP files and their maps I

- passwd
 - » passwd.byname
 - » passwd.byuid
 - » passwdext
- group
 - » group.byname
 - » group.bygid
- hosts
 - » hosts.byname
 - » hosts.byaddr
 - » dns
- ethers
 - » ethers.byname
 - » ethers.byaddr

YP files and their maps II

- locale
 - » locale.byname
- timezone
 - » timezone.byname
- networks
 - » networks.byaddr
 - » networks.byname
- services
 - » services.byname
- rpc
 - » rpc.bynumber
- protocols
 - » protocols.byname
 - » protocols.bynumber

YP files and their maps III

- netgroup
- netgroup.byuser
- netgroup.byhost
- publickey
- publickey.byname
- netmasks.byaddr
- passwd_dialup
- quotas
- pcap
- master.hesiod
- lcfg

YP (really Hesiod) files

- homes.remote
- usr.remote
- export.remote
- obj.remote
- usr.ubin
- homes.local
- usr.local
- export.local
- obj.local

Map building

- `/usr/local/bin/yped <filename>`
- Master file structure
 - » `/etc/nis/src/` `/etc/nis/src/RCS`
 - » `/etc/nis/src/lcfg` `/etc/nis/src/lcfg/RCS`
 - » `/etc/nis/src/amd` `/etc/nis/src/amd/RCS`

How to load a previous revision of a YP file

- `cd /etc/nis/src`
 - `rm passwd; rm passwdext`
 - `co passwd; co passwdext`

 - `yped passwd`
- OR**
- `/var/nis/master/dcs.ed.ac.uk/Makefile passwd`

Map Building (continued)

- `yed` calls `/var/nis/master/dcs.ed.ac.uk/Makefile`
- This Makefile calls one of
 - » `Makefile.standard`
 - » `Makefile.local`
 - » `Makefile.amd`
 - » `Makefile.pcap`
- One file can generate several YP maps

/etc/group and its maps

/etc/group

```
cs_co:*:28017:  
cs_lec:*:28018:  
cs_sec:*:28019:  
cs_tech:*:28020 :
```

group.byname

```
cs_co cs_co:*:28017:  
cs_lec cs_lec:*:28018:  
cs_sec cs_sec:*:28019:  
cs_tech cs_tech:*:28020:
```

group.bygid

```
28017 cs_co:*:28017:  
28018 cs_lec:*:28018:  
28019 cs_sec:*:28019:  
28020 cs_tech:*:28020
```

ypmatch

- To find out GID given group name

```
[cull]ajs: ypmatch cs_co group.byname  
cs_co:*:28017:
```

- To find out group name given GID

```
[cull]ajs: ypmatch 28018 group.bygid  
cs_lec:*:28018:
```

Makefile rule for group

```
group.time: $(SRCDIR)/group
  -@if [ -f $(SRCDIR)/group ]; then
    awk 'BEGIN { FS=":"; OFS="\t"; } { print $$1,$$0 }' \
      $(SRCDIR)/group | $(RELEASE) $(NISDIR)/group.byname; \
    awk 'BEGIN { FS=":"; OFS="\t"; } { print $$3, $$0}' \
      $(SRCDIR)/group | $(RELEASE) $(NISDIR)/group.bygid; \
    touch group.time; \
    echo "updated group"; \
    $(MERGEGROUP) ; \
    if [ ! $(NOPUSH) ] ; then \
      $(NISPUSH) group.byname; \
      $(NISPUSH) group.bygid; \
    fi
  fi
```

Special build cases

- passwd
 - » special treatment by yped
 - » passwdext
 - » homes.remote

- group
 - » distribution by template

Special build cases II

- lcfg
 - » /usr/local/etc/nis/makelcfgmap
- netgroup
 - » much of map auto-generated
- hosts

Map propagation - Sun's way

- Makefiles generate DBM file for each map
- Makefiles call `yppush` for each map
- For each bunch of 4 slaves, `yppush`
 - » enumerates the DBM
 - » sends the textual map to the 4 slaves
- Problems
 - » inefficient
 - » hangs if a slave is down or confused

Map propagation - our way

- Makefiles generate text file for each map
- Makefiles call `nispush` for each map
- `nispush`
 - » broadcasts the map in chunks
 - » asks each slave in turn what it missed
 - » iterates
- More efficient and doesn't hang on down or hung slaves

Map reception

- Each slave runs a version of nispull
 - » nispull_udp
 - » nispull_tcp
- nispull
 - » listens for broadcasts of maps
 - » plonks file in `/var/nis/slave/dcs../incoming`
 - » calls makemap