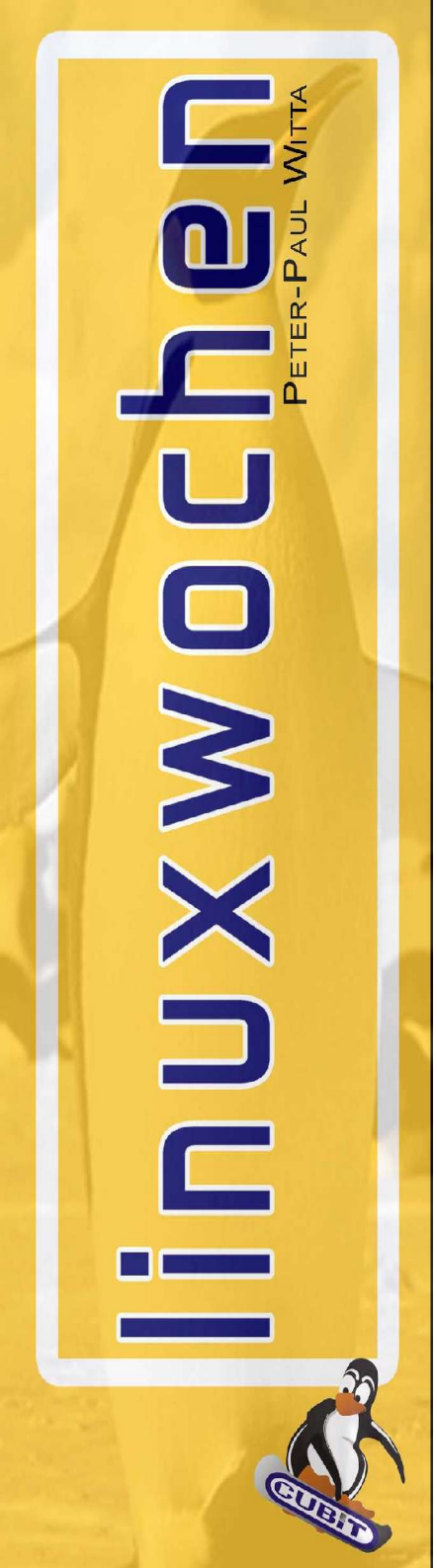


Unix Filesharing
NFS und NIS

Ing. Peter-Paul Witta
paul.witta@CUBiT.at

Unix Filesharing – NFS

- native Unix Filesharing
- Sun NFS ist Grundlage von NFS
- Unix Berechtigungen
- gute Integration in NIS
- Remote-Boot und Remote-Root Filesystem möglich
- Einfaches Prinzip (bis einschl. v3)
- basierend auf Sun RPC
- Schnelles NFS setzt Kernel Daemon voraus im Server
- Userspace NFS langsamer aber sicherer (alt!)

Storage Konsolidierung

- Statt SAN zur Plattenanbindung Ethernet
- warum nicht NFS statt iFCP und iSCSI
- weniger Overhead
- mit ausreichend Tuning entsprechend schnell
- Standard-Server oft langsam
- Tuning, tuning, tuning und richtige Hardware
  - viel RAM (>5 GB) und viel CPU (dual 2 Ghz)
  - richtiger IO-Bus (Fibre Channel, Escalade Storage Switch)
  - richtige Platten
  - ausreichend Spindeln für rapid random IO
  - ausreichend Bandbreite (mehrere 1000baseTX)
- oder spezielle Maschine (NetApp)

The Next Step: Desktop

- Diskless Linux Desktop
- Remote Boot Solution
- Semi Server-Based: Nutzung lokaler CPU und MEM Ressourcen
- DHCP kann NFS Rootserver mitteilen
- Performance Tuning wichtig
- kein Unterschied zu lokalem Arbeiten
- richtige mount Option, ggf. Failover-Systeme
- FAM für Desktop wichtig

NFS im Detail

- basiert auf Sun RPC
- udp und tcp möglich
- Performance erfordert individuelles Tuning
- Performance wichtig:
  - LAN: 100MBps => 8 MB/sec, Disk: 50 MB/sec
  - auch mit Gigabit LAN langsamer als Disk
  - aber: mit Tuning kann es verschmerzbar werden
- File (Echtdaten) und Attribut (Metadaten) Zugriff
- Caching von Metadaten zur Entlastung im RAM
- Write Coherency
- hard/soft Mount Option

Eigenschaften

- Namespace: server:/pfadname/dir/filename.txt
- Z.B. mount cube2:/mnt/shares /mnt/c2/shares
- Mount-Optionen:
  mount -orsize=8192,hard /mnt/c2/shares
- NFS Filesystemtreiber am Client notwendig
- am Server
- Server auch für kleinere Systeme (PDA)
- Auch im Embedded Bereich möglich
- Auch für Windows-Systeme erhältlich
- Standardprotokoll RFC 1094, RFC 3010 (v4), RFC 1813 (v3)
- Routebar, auch über Internet möglich

NFS im Einsatz

- Mount Options hard/soft
- tcp vs. udp: Wann welche Option?
- Paketgrösse bei UDP und TCP
- Symlink Support (lokale Auflösung!)

sys1:

/a-loc 2M
/exp
  /sub
  /lnk->/a-loc
  (2M)

sys2:

/a-loc 1M
/mnt
  /sub
  /lnk->/a-loc
  (1M)

mount sys1:/exp /mnt

sys1;/exp/sub/lnk/a
->sys1:/a-loc

sys2:/mnt/sub/lnk/a ->
sys2:/a-loc

NFS im Einsatz (2)

- Re-Export möglich: Storage-Hubbing
- Manchmal mounten Server Clients :-)
- TCP-IP reicht; Internet-tauglich; mit TCP sogar durch SSH-Tunnel
- zB: Zentral-Server mountet Filialserver via FRAD, erlaubt Zugriff auf alle Filialen und Rollouts per cp in Subdirectories
- volle Symlink, Hardlink, Attribut-Unterstützung
- “true Unix Technology” -> case sensitive, file-locking,...

Tuning

- TCP vs.UDP
  - TCP bei WAN und unreliable slow Links mit Packetloss
- Paketgrösse bei UDP: Fragmentierung vermeiden
- MTU=NFS-block-size+UDP-hdr+IP-hdr+Ethernet-hdr
- MTU 9000 bei Gigabit Ethernet:
  - richtigen Switch nehmen!!

Timeouts und Tuning im Detail

- acregmin/max: Attribute reguläre Datei
- acdirmin/max: Attribute Verzeichnis
- actimeo= alle 4 :-)
- rsize
- wsize
- timeo=Timeout für Algorithmus in 1/10 sec.
- noac bei gleichzeitigem Zugriff

Linux Details

- Kernel 2.4: nfsv3
- Kernel 2.6: experiementielle Patches für nfsv4
- nfsv3 ist aktuell und stabil nutzbar
- Kernel NFS v3 Server Produktiv
  - neue Features: BS>8k (bis 32K), TCP
- Userspace-NFSd
  - kein Locking Support
- Threadanzahl definierbar
  - Faustregel: 2*#cpu
  - bei langsamen Systemen mehr => weichere Verarbeitung
  - bei schlechten Links mal TCP probieren
  - IO muss mithalten können: sar, vmstat

NIS (yp)

- verteilt Unix Systemtabellen im Netz:
  - passwd, group, hosts, protocols, services, networks
  - optional shadow
- Einheitliche UserIDs und Passwörter Netzwerkweit
- NIS Domain
- NIS+ (Secure, kompliziert, wenig verbreitet)
  - “Muttersprachliches Domainkonzept” von Unix
  - kein Directory
  - nicht hierarchisch
  - Fehlertolerantes Fail-Over Konzept integriert

Struktur, Eigenschaften

- RPC basierend
- fügt sich gut in Unix ein
- relativ leicht handhabbar
- Einbindung via nsswitch und pam
  - + laufender Daemon

Einrichtung Client

- yp.conf:
  - ypserver 10.0.0.1
- oder
  - broadcast
- RPC: daher portmapper notwendig
- in nsswitch.conf
- in passwd:
+::::::
- bei pam integriert

```
nsswitch.conf

passwd:         nis compat
group:          nis compat
shadow:         nis compat

hosts:          files dns
networks:       files

protocols:      nis files
services:       nis files
ethers:         nis files
rpc:            nis files

netgroup:       nis
```

nsswitch, pam, passwd & co

- nsswitch.conf: sorgt ab glibc 2.x für Einbindung der Netzwerkuser
- ypbind: kommuniziert mit yp Server
- pam:
  - integriert Support in “standard” Modul
- passwd: mit +-Syntax können User selektiv ein- und ausgeschlossen werden
- neue Befehle: yppasswd statt passwd
- ypcat: zeitgt Maps an
- ypchsh, ypchfn,...

Einrichtung Server

- Installation Software
- ypinit -m
- danach immer “make” in /var/yp zum Updaten der Maps aus den Quellen
- Slave-Server: ypinit -s zum Einrichten
- Update der Slaves: yppush am Master oder ypxfer
- Rolle muss konfiguriert werden, damit ypbind sich auskennt
- Diskussion: shadow Passwords mitsenden?
  - -> wer ist Root auf den Clients?

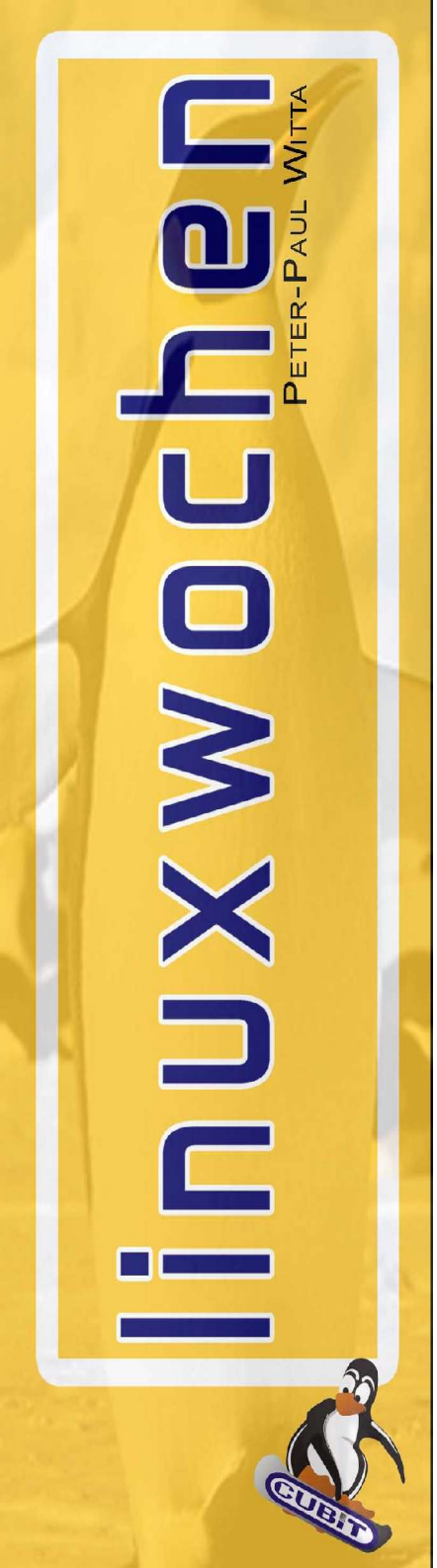
Failover Konzepte

- Master-Slave Konzept
- Ermittlung durch Client mittels Broadcast
- oder
- Konfiguration mehrerer Server
- Unix-Protokoll, daher in sich ausfallsicher
  - (vgl. PDC/BDC bei Windows)
- aber auch mit Service-Manager machbar

Sun RPC

- RPC Dienst
- für NFS

```
paul@spacewalker:~/linuxwochen04$ rpcinfo -p
   Program Vers Proto   Port
    100000    2   tcp    111  portmapper
    100000    2   udp    111  portmapper
    100003    2   udp   2049  nfs
    100021    1   udp  32768  nlockmgr
    100021    3   udp  32768  nlockmgr
    100005    1   udp    735  mountd
    100005    1   tcp    738  mountd
    100005    2   udp    735  mountd
    100005    2   tcp    738  mountd
    100024    1   udp    792  status
    100024    1   tcp    795  status
paul@spacewalker:~/linuxwochen04$
```



Sun RPC

- RPC Dienst
- rpcinfo -p ypserver
- für NIS
- für NFS
- lockmanager
- fam
- full-featured fileserver

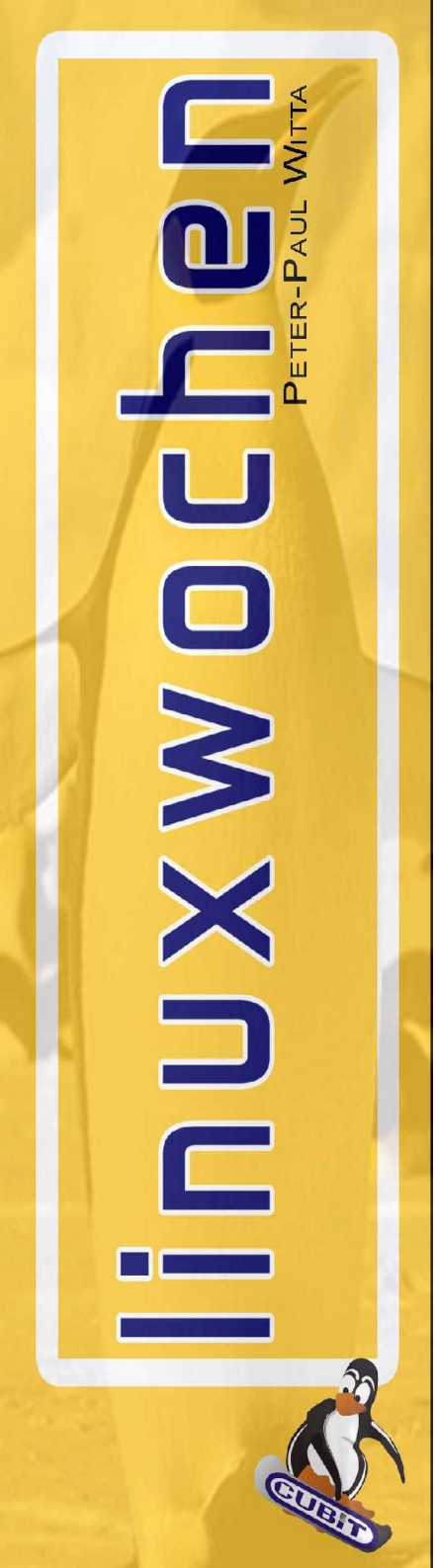
```
paul@wcube7:~$ rpcinfo -p cube2
   Program Vers Proto   Port
    100000    2   tcp    111  portmapper
    100000    2   udp    111  portmapper
    100004    2   udp    926  ypserv
    100004    1   udp    926  ypserv
    100004    2   tcp    929  ypserv
    100004    1   tcp    929  ypserv
    100009    1   udp    928  yppasswdd
    100007    2   udp    939  ypbind
    100007    1   udp    939  ypbind
    100007    2   tcp    942  ypbind
    100007    1   tcp    942  ypbind
    391002    2   tcp    620  sgi_fam
    100024    1   udp    892  status
    100024    1   tcp    895  status
    100003    2   udp   2049  nfs
    100003    3   udp   2049  nfs
    100003    2   tcp   2049  nfs
    100003    3   tcp   2049  nfs
    100021    1   udp  34179  nlockmgr
    100021    3   udp  34179  nlockmgr
    100021    4   udp  34179  nlockmgr
    100021    1   tcp  33111  nlockmgr
    100021    3   tcp  33111  nlockmgr
    100021    4   tcp  33111  nlockmgr
    100005    1   udp    930  mountd
    100005    1   tcp    934  mountd
    100005    2   udp    930  mountd
    100005    2   tcp    934  mountd
    100005    3   udp    930  mountd
    100005    3   tcp    934  mountd
```

Tipps, Tricks, Architektur

- schnelles Netzwerk
- schnelle Disks und IO-Subsystem
- FC/AL wenn möglich, sonst 3ware Escalade
- Attribute Caching wo immer möglich
- Server IO-Tunen
- oder gleich Netapp :-)
- NFS ist nicht alt
- NFS v3 und v4 mit NetApp entwickelt
- v4: Byte Range Locking, SMB-ähnliche Features, wirkt “strange”
- v3: TCP, UDP, Blocksize und Performance
- jedes System kann von NFS profitieren

Dedizierte Hardware

- NetApp Filer mit Cluster Option
- FC/AL Interfaces
- active/active load sharing
- Geheimnis: gute und schnelle Platten
- gute Softwarearchitektur
- viel Cache
- Demo auf den Linuxwochen Wien / CUBiT IT



NFS und NIS

DANKE!

Ing. Peter-Paul Witta
paul.witta@CUBiT.at