

Backup in SOHO

oder

Backup in meinem Heim-Netzwerk mit dirvish und rsync

Jörg Brüche

Senior Production Engineer, MySQL / Sun / Oracle
(joerg.bruehe@web.de)

2011-Nov-5

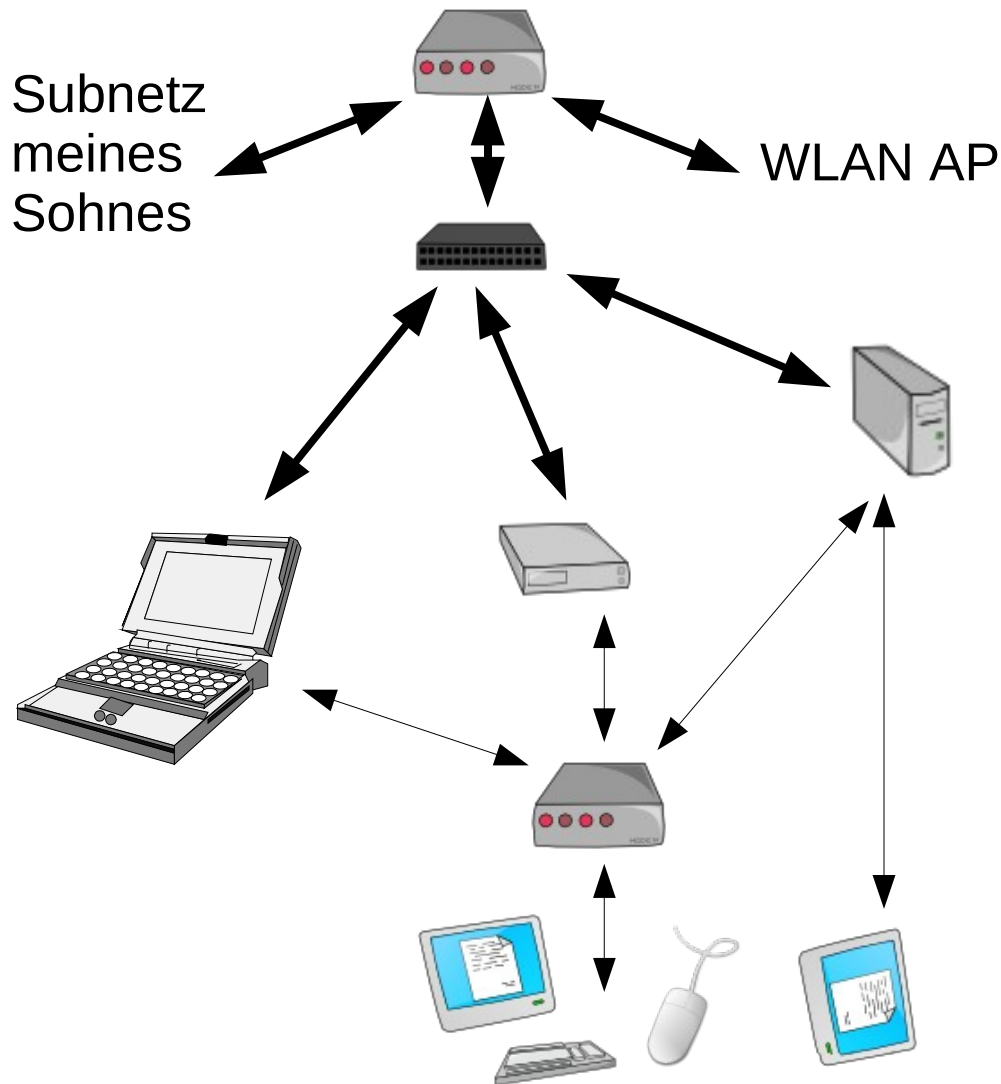
Vorbemerkungen

- Oracle (ex Sun, ex MySQL AB), Telearbeit

„SOHO“ = „small office, home office“

- Vortrag beruht auf persönlichen Vorlieben und Bedarf
- Kein vollständiger Markt-Überblick
- Gibt persönliche Erfahrungen wieder
- YMMV (= „Your mileage may vary“):
nicht unbedingt übertragbar

Mein Heimnetz (Auszug)



DSL-Modem, Firewall,
DHCP, 4-Port Router

8-Port Switch

Tower PC (dual-head)
(Haupt-Rechner)

Desktop PC „Pegasos“
(PowerPC CPU)

Laptop

4-Wege KVM-Switch
(je 1* VGA + 2* PS/2)

2* 19" TFT, davon 1 rotiert

Mein Heimnetz: Benutzung

- Alle Maschinen benutzen Linux (SuSE oder Debian)
- Plattengröße jeweils zwischen 60 und 500 GB
- Mehrere Quell- und Testbäume auf jeder Maschine
- Dual- oder Multi-Boot; Virtuelle Maschinen

- Tower: Haupt-Arbeitsmaschine:
Mail + IRC, Entwicklung, Tests; Privates
- Pegasos: Entwicklung, Tests (andere CPU !)
- Laptop: Ersatz für Tower, auch für Reisen (Mail + IRC)

- Maschinen laufen nur bei Benutzung
- Mail mit Thunderbird (direkt IMAP + POP3),
Tower und Laptop vor jedem Wechsel synchronisiert

Warum sichern ?

- Inhalte können wichtig sein (Verträge, Online-Banking, ...)
- ... oder unwiederbringlich (Fotos, ...)
- ... oder aufwändig zu erzeugen (Konfiguration, ...)

- Platten können ausfallen
- Benutzer können Fehler machen
- Hard- und Software kann defekt sein

=> Datenverlust

„Niemand will Backup, aber alle wollen **Recovery**“

„Wenn die Datei nicht gesichert wurde,
dann war der Inhalt wohl nicht wichtig“

Was will ich sichern ?

- Alle Maschinen
- Alle Dateien: System, Software, Konfiguration, User
- Derzeit: 4 – 40 GB Snapshot-Größe je Maschine

- Lokale Entwicklungs-Bäume

- Keine Quell-Bäume (Inhalt wechselt zu häufig, Kopien vom Firmen-Server – Wiederherstellung ist einfach)
- Keine Test-Bäume

- (ToDo) Maschinen-Konfiguration (Partitionierung, ...)
- (ToDo) Alternative Installationen (multi-boot !)

Eigenschaften eines perfekten Backup (IMO !)

- Vollständig auf einem Datenträger, ungeteilt
- Mehrere Generationen parallel, direkter Vergleich möglich („diff -r“)
- Schnell, leise

- Für jede Maschine in meinem Heimnetz
- Keine spezielle Hardware auf den Clients
- Erweiterbar, wenn Maschinen dazukommen

- Generationen von Backups möglich
- Inklusive System-Sicherung (ideal: boot-fähig)
- Schreibschutz der Medien und Auslagerung möglich
- Auch bei Ausfall der Backup-Maschine zugreifbar

Festlegung einer Backup-Architektur

- Band ist langsam, geringe Kapazität, laut
- und erlaubt keinen direkten Zugriff für Vergleiche
- DVD hat Größen-Begrenzung
- und sollte lokal angeschlossen sein (= zuviel HW)
- Platte ist groß, schnell und preiswert
- Netzwerk-Zugriff vermeidet Client-HW Anschaffung

**=> Separate kleine Backup-Maschine
und externe Platten beschaffen**

Warum keine NAS-Box ?

Typisches NAS (fertig verfügbar)

- bietet SMB und NFS, vielleicht FTP, mit Web-Interface, aber ich will rsync und Unix-artigen lokalen Zugriff („diff“)
- hat gemeinsame interne Platte für Betrieb + Archiv
- wenn zwei Platten, dann ohne Kontrolle (JBOD)
- c't 5/2009, S. 104 - 120

Besser ein eigenes Äquivalent zur NAS-Box

- auf Linux/Unix-Basis, für lokale Tools und rsync-Zugriff
- mit externen Archiv-Platten (Redundanz!)
- und interner System-Platte (bevorzugt Flash/SSD)
- vielleicht erweiterbar für weitere Dienste (CUPS, ...)

Prinzip der rsync-Backups

- (Init:) Vollständige Kopie des gewünschten Baums auf der Archiv-Platte erzeugen unter eindeutigem Namen
- (Loop:) Mittels „rsync“ prüfen, ob der aktuelle Baum mit dem letzten Archiv-Stand übereinstimmt
 - Neues Verzeichnis anlegen
 - Identische Dateien per Hard-Link vom letzten Archiv übernehmen (aber nicht gelöschte)
 - Neue Dateien kopieren
- Ergebnis: Unveränderte Dateien nur einmal auf der Platte, mit Links aus jedem Sicherungs-Lauf
- Jeder Sicherungs-Lauf ein vollständiger Baum

Beispiel

```

$ cd /blue-snapshot/DIRVISH/trift2-HOME
$ sudo du -sk 20080303-1727 20080310-2055
864008 20080303-1727
301248 20080310-2055
$ sudo du -sk 20080310-2055
882380 20080310-2055
$ ls -l 20080310-2055
total 200
-rw-r--r-- 1 root root 175049 Mar 10 20:58 index.gz
-rw-r--r-- 1 root root 7098 Mar 10 20:58 log.gz
-rw-r--r-- 1 root root 764 Mar 10 20:58 summary
drwxr-xr-x 3 root root 4096 Mar 10 10:08 tree
$ ls -l 20080310-2055/tree/home/
total 44
...
drwxr-xr-x 75 joerg users 4096 Mar 10 17:15 joerg
drwxr-xr-x 37 ute users 4096 Mar 9 23:05 ute

```

Backup-Tools auf Basis 'rsync' (unvollständig)

- Selbstgeschriebene Skripte (aber warum?)
- rsnapshot <http://www.rsnapshot.org/>
Meine ersten Versuche
- dirvish <http://www.dirvish.org/>
Derzeitige Lösung wegen Artikel in c't 2006/07 (S. 212)
<http://www.heise.de/kiosk/archiv/ct/2006/7/212>
- backuppc <http://backuppc.sourceforge.net/>
Zu spät entdeckt, scheint gute Alterungs-Mechanismen zu haben, aber wohl kein Äquivalent zum „vault“
- Weitere auf der Homepage von rsync:
<http://samba.anu.edu.au/rsync/>

Mein Wechsel von „rsnapshot“ zu „dirvish“ (1)

- rsnapshot hat (nur) eine zentrale Konfigurations-Datei
- mit der Liste aller (zu sichernden) Maschinen und Filesysteme
- und erlaubt damit keine (nur) teilweise Verfügbarkeit der Backup-Clients

- Mehrere Konfigurationen mit „include“-Dateien für gemeinsame Teile scheitern an Parser-Restriktionen
- Mehrere separate Konfigurations-Dateien möchte ich nicht warten
- „Austricksen“ mittels Datei-Manipulationen möchte ich vermeiden

Mein Wechsel von „rsnapshot“ zu „dirvish“ (2)

- dirvish erlaubt mehrere „vaults“ in mehreren „banks“
- „**bank**“: Filesystem zur Ablage von Backups
- „**vault**“: Baum innerhalb einer „bank“, legt die Daten-Quelle (Maschine, Verzeichnis) fest
- Vaults sind **eigenständig** (Konfiguration, Log, Index)
- Platten mit Vaults können ohne Änderung an anderen Maschinen benutzt werden (anderer Backup-Server !)
- Getrennte Vaults sind perfekt für teilweise Verfügbarkeit und unterschiedliche Backup-Jobs („Mail“ vs „System“)
- Vaults in verschiedenen Banks verteilen Daten über Platten (**Ausfallsicherheit**)



Sprachregelung: „Client“ und „Server“

- „Client“ benutzt eine Dienstleistung („Service“), „Server“ erbringt sie
Alltagsvergleich: Kunde und Lieferant
- Backup-Dienstleistung ist das Speichern von Dateien: Server hat die Speichermedien, Client liefert die Daten
- Client kann leistungsmäßig die stärkere Maschine sein
- Dirvish-Backup wird vom Server angestoßen („pull“)
Alltagsvergleich: Lieferant macht Angebot
- Client- und Server-Rolle für jeweils eine Dienstleistung, nicht allgemein für eine Maschine
- Backup-Server kann (gleichzeitig!) für andere Dienste ein Client sein, z.B. für ssh-Aufrufe

Backup-Server: Igel-Compact

- Thin Client, 600 MHz VIA Eden CPU, 1 GB RAM, 4 GB Compact Flash als System-Platte (Debian 5.0.3), Schnittstellen: VGA + DVI, Tastatur + Maus (PS/2), seriell, parallel, LAN, 3* USB 2.0 (Combär, Januar 2009: 120 €)
- Externe Platten: 2* je 400 GB (SATA über USB)

Früherer Versuch mit Igel-J, nicht geeignet:

- USB 1.1 ist unbrauchbar langsam
- Riser-Karte macht Datenfehler (=> USB 2.0 nicht möglich)
- 256 MB ist zu wenig RAM für fsck auf großen Filesystemen und (auch mit Swap) für große rsync-Jobs

Ersatzweise Pegasos, aber separater Backup-Server flexibler



Konfiguration: Server

- LVM nutzen, für Wachstum der Backup-Filesysteme
- Backup-Platte über „Volume ID“ identifizieren (mount)
(Device-Namen sind nicht konstant)
- Externe Platten eindeutig identifizieren (z.B. Gehäuse)
- Auch Backup-Server können ausfallen:
Auf Wechsel zu einer anderen Maschine vorbereiten
(Hardware-Anschluss, Typ des Filesystems, LVM,
neutrale Konfiguration)
- Backups archivieren ?
- „Alterung“ / automatisches Löschen

Server: /etc/fstab

```
# /etc/fstab: static file system information.
#
# <filesystem> <mount pt>    <type>    <options>                <dump> <pass>
proc          /proc          proc      defaults                    0      0
/dev/hdb1    /              ext3     defaults,errors=remount-ro 0      1
/dev/scd0    /media/cdrom0 udf,iso9660 user,noauto                 0      0

#
# File systems in "vg_blue"
#     =BLUE_LV_HOME /blue-home      ext3 defaults,noauto    0      0
LABEL=BLUE_LV_SNAP /blue-snapshot ext3 defaults,noauto    0      0
```

man mke2fs:

-L new-volume-label

Set the volume label for the filesystem to new-volume-label.

The maximum length of the volume label is 16 bytes.

Konfiguration: Client

- Gute (logische) Backup-Einheiten definieren, unterscheiden zwischen „komplett“ (lang, selten) und kleineren (schnell, häufig) Teilen (Mail !)
- Konsistentes Schema für alle Maschinen
- Alle Details lokal auf dem Backup-Client halten (= bei den Daten) für leichtere Pflege
- „Moduln“ für den rsync-Dämon nutzen:
man rsyncd.conf
man rsync
usr@serv> rsync client::modul
usr@serv> rsync -r client::modul

Konfiguration: Sicherheit (1)

- System-Backup benötigt „root“-Rechte zum Lesen:
rsync-Dämon kann das auch ohne password-loses ssh
 - + kein Login-Risiko, einstellbar auf **read-only**, IP-Adressen
 - ohne Verschlüsselung im Netz
- Rsync-Dämon kann als „root“ auf Backup-Client laufen und an Nicht-„root“ auf Backup-Server liefern
- SOHO hinter Router/Firewall kann ohne Verschlüsselung auskommen (IMHO, YMMV !)

...

Konfiguration: Sicherheit (2)

...

- Alternative: rsync über ssh, mit oder ohne Passphrase
 - + Verschlüsselung
 - System-Sicherung nur als „root“
 - ohne Passphrase: wer will das erlauben?
 - mit Passphrase: schlecht zu automatisieren
 - evtl. leere Passphrase und IP-Adress-Bindung
- Admin-Anforderungen auf allen Maschinen prüfen
- Ich (persönlich !) bevorzuge rsync-Dämon + Moduln

Beispiel rsync-Dämon und Modul

```
# Auszug aus /etc/rsyncd.conf (client)
[hometest]
...
include = /home/ /home/joerg/ /home/ute/ /home/lost+found/ \
- /* - /home/* - /home/*/.thumbnails/ - /home/**/Cache/
# "- /home/*" is needed for "- /home/*/.thumbnails/" to work!
# missing: exclusion of several other caches
...
```

- Modul prüfen: rsync aufrufen ohne Ziel-Pfad
usr@serv> rsync -r client::modul
- **usr@serv**> rsync -r client::hometest | \
egrep -v 'WANT1|WANT2|...'
- **usr@serv**> rsync -r client::hometest | \
grep 'NOTWANT'

Einige Backup-Zeiten

Inkrementeller remote-Backup

- „HOME“ tree (785 MB): 2 min 28 sec
- „MAIL“ (345 MB, 245 MB neu): 1 min 20 sec
- „ALL“ tree (3.15 GB): 31 min 50 sec
- „ALL“ (8 GB, 900 MB neu): 34 min 5 sec
- „ALL“ (42 GB, 3 GB neu): 66 min 22 sec
- „WORK“ (500 MB aus 14,7 GB, init): 2 min 17 sec

- fsck (100 GB ext3, 75% used) 15 min
- fsck (196 GB ext3, 93% used) 34 min 52 sec

Inkrementell lokal (rsyncd):

- „ALL“ (2 GB von Flash) 10 min

Windows sichern

Allgemein:

- „rsync“ ist auch auf Windows verfügbar

=> Dirvish kann auch Windows (übers Netz) sichern

Bei mir:

- Laptop hat auch Windows (vorinstalliert, geschrumpft)
- Windows läuft nur ohne Netz, nur für lokale Arbeiten
- Linux hat Mount-Eintrag für die Windows-Partition

=> Ich sichere die Windows-Partition als Teil der Linux-Sicherung

Stärken und Schwächen von Dirvish

- + Dirvish beruht auf rsync:
 - Separate Maschinen, kein direkter Programm-Zugriff
- + Vaults sind unabhängig voneinander:
 - Häufigkeiten, Alterung, ... sind individuell einstellbar
 - Hinzufügung / Änderung / Wegfall sind einfach
- Dirvish beruht auf rsync, Vaults sind unabhängig:
 - Hardlinks nur bei Namensgleichheit in aufeinander folgenden Sicherungen desselben Vaults

Alternativen: anderes Tool (z.B. „storebackup“) oder eigene Nachbearbeitung

Work In Progress: Client-Skripte (1)

- Dirvish unterstützt Skripte vor und nach der Sicherung: pre-server, pre-client, rsync, post-client, post-server
- Festlegung im Vault
- Sicherung nur bei Erfolg der pre-Skripte (Exit-Code 0)
- Typische Aufgaben:
Client: Service stoppen oder Schnappschuss erzeugen
Server: Archivierung, Cleanup, Status, Info-Mail, ...
- Skripte erhalten Environment.-Variable:
DIRVISH_SERVER, DIRVISH_CLIENT, DIRVISH_SRC,
DIRVISH_DEST, DIRVISH_IMAGE
- Skript-Ausgabe wird im Log (Default: komprimiert)
(im Sicherungs-Baum) abgelegt

Work In Progress: Client-Skripte (2)

- Skripte laufen unter Kennung wie Sicherung, client-Skripte brauchen ssh-Login im Batch!

=> Eigener Backup-User

- mit batch-fähigem ssh-Login
- Rechte ausreichend für Skript-Aufgaben
- ssh-Login absichern, Rechte beschränken

Alternative: Server-Skript kann beliebige Aktionen ausführen, also auch ssh-Sitzungen auf Client

Meine vorgesehene Skript-Nutzung:

- 1) Plausibilitäts-Kontrolle auf Client (Multi-Boot!)
- 2) Vor/Nachbereitung, z.B. System-Status schreiben

Mein TODO (Backup)

- Automatisierung !
- Skripte auf den Clients, die lokal Konfigurations-Info erstellen + aktualisieren, um sie als Files zu sichern
- Plausibilitäts-Kontrolle über Hostname und IP-Adresse, als Schutz gegen Netz-Änderungen
- Bei System-Sicherungen die Betriebssystem-Version prüfen (großer Upgrade, Multi-boot-Maschinen)
- Automatisches Löschen an unregelmäßigen Backup-Rhythmus anpassen
- Stromversorgung der Backup-Platten per Software schalten (geschaltete Steckdosenleiste, Eigenbau oder z.B. www.gembird.de <http://sispmctl.sourceforge.net/>)

Links / Dokumentation

- <http://www.dirvish.org>
- <http://wiki.edseek.com/howto:dirvish>
- `man rsync`
- `man rsyncd.conf`
- [http://www.mikerubel.org/computers/
rsync_snapshots/](http://www.mikerubel.org/computers/rsync_snapshots/)
Konzepte, Selbstbau
- <http://storebackup.org/>
Hardlink-Backup über NFS ohne rsync

Danke!

Fragen?

Jörg Brüche

joerg.bruehe@web.de

Backup in SOHO

Anhang 1: Auszüge aus Konfigurations-Dateien

Jörg Brühe

Senior Production Engineer, MySQL / Sun / Oracle
(joerg.bruehe@web.de)

2011-Nov-5

Konfiguration Server: /etc/dirvish/master.conf (1)

```
#  
# /etc/dirvish/master.conf      Central configuration file  
#  
# To be (partially) overridden by "default.conf" in the vaults  
#  
  
bank:  
    /blue-snapshot/DIRVISH  
    /noir-snapshot/DIRVISH  
  
checksum:      1  
  
expire-default: +9 weeks  
  
log:      gzip  
  
...
```

Konfiguration Server: /etc/dirvish/master.conf (2)

...

exclude:

```
    /etc/mtab
```

```
# no:  /var/lib/nfs/*tab
```

```
# no:  /var/cache/apt/archives/*.deb
```

```
    .kde/share/cache/*
```

```
    .firefox/default/*/Cache/*
```

```
# no:  /usr/src/**/*.*
```

```
    lost+found/
```

```
# For example, the exclusion rules above exclude /etc/mtab
```

```
# and any files with a specific ending in /var/lib/nfs.
```

```
# It also excludes any Debian packages in /var/cache/apt/archives,
```

```
# all files in the K Desktop Environment's cache directory,
```

```
# and all files in Firefox's cache directory, with a wildcard to deal
```

```
# with the random filename it is given.
```

```
# Finally it uses a wildcard to match any directories under /usr/src
```

```
# that have object files
```

```
# and the specific directory lost+found/, where ever it may be.
```

```
# You can define additional excludes in vault specific configuration
```

```
files for even more control.
```

Konfiguration Server: /blue-snapshot/DIRVISH/trift2-MAIL/dirvish/default.conf

```
#
# /blue-snapshot/DIRVISH/trift2-MAIL/dirvish/default.conf
#
# 'trift2' is the main work machine (Linux, Athlon 32 bit)
#
# Backup is done via a rsync daemon !
# IP address will never match host name, so it will never be dropped.

client:          192.168.2.34
tree:            :mail
xdev:            0
index:           gzip
image-default:  %Y%m%d-%H%M

expire-rule:
#MIN  HR    DOM    MON    DOW    EXPIRE
# *    *    *      *      1      +3 months
# *    *    1-7    *      su     +1 year
# *    *    1-7    1,4,7,10  1     never
# no:   *    10-20  *      *      +10 days
# See documentation, note "master.conf": "expire-default"
# Note: *Last* matching rule applies !
```

Konfiguration Client: /etc/rsyncd.conf (1)

```
gid = users  
read only = true
```

```
transfer logging = true  
log format = %h %o %f %l %b  
log file = /var/log/rsyncd.log  
pid file = /var/run/rsyncd.pid
```

```
hosts allow = 192.168.2.36 192.168.2.39  
slp refresh = 300
```

```
[all]
```

```
    path = /  
    comment = The whole local file system  
    uid = 0  
    gid = 0  
    exclude from = /etc/rsyncd.exclude.ALL  
    transfer logging = false
```

```
...
```

Konfiguration Client: /etc/rsyncd.conf (2)

...

```
[mail]
path = /home
comment = Mail directories of the local users
uid = 0
gid = 0
include = /*/ /**thunderbird/ /**thunderbird/** - /* - /*/**
transfer logging = false
```

```
[home]
path = /
comment = Home directories of the local users
uid = 0
gid = 0
include = /home/ /home/joerg/ /home/ute/ /home/lost+found/ \
  - /* - /home/* - /home/*/.thumbnails/ - /home/**/Cache/
# "- /home/*" is needed for "- /home/*/.thumbnails/" to work!
# missing: exclusion of several other caches
transfer logging = false
```

Konfiguration Client: /etc/rsyncd.exclude.ALL

```
/MySQL/  
/dev/  
/proc/  
/sys/  
/media/  
/mnt/  
/other_root/  
/spare/  
/var/spool/  
home/joerg/.java/deployment/cache  
home/joerg/.java/deployment/log  
home/joerg/.kde/share/config/session  
home/joerg/.ooo-2.0-pre/user/config/imagecache  
home/joerg/.opera/cache4  
home/joerg/.opera/cacheOp  
home/joerg/.opera/images  
home/joerg/.mozilla/default/xbmx0471.slt/Cache  
home/joerg/.mozilla/firefox/pwt89vfq.default/Cache  
home/joerg/.mozilla/firefox/a3v4fcks.default/Cache  
home/joerg/.thumbnails  
...
```

Backup in SOHO

Anhang 2: Ergänzungen

Jörg Brühe

Senior Production Engineer, MySQL / Sun / Oracle
(joerg.bruehe@web.de)

2011-Nov-5

Heimnetz ist anders als Firmennetz

- Maschinen laufen nicht durchgehend (7*24)
- Keine Band-Roboter mit Automatisierung
- Keine schnellen Bandlaufwerke (DLT und ähnlich)
Exabyte / DAT ist langsam
- Geräuschprobleme (kein separates RZ)
- Kein Operator / Backup-Personal
- Backup ist „eine Aufgabe unter vielen“
und kann Zeitdruck zum Opfer fallen

Mein früherer Backup

- (Alter) Tower hatte SCSI-Adapter
- Vorhandene Bandlaufwerke + Kassetten:
QIC („6150“: 150 MB), Exabyte (? GB), DAT
- DAT-Laufwerke (HP) haben (bei mir !) nie funktioniert
- 150 MB ist zu klein (ohne Komprimierung)
- Exabyte: Kassetten noch zu klein für kompletten Schnappschuss, braucht ca. 3 h, ist (zu) laut
- Band-Backup nur für den Tower, nicht regelmäßig, über mehrere Kassetten verteilt
- Kein Backup für die anderen Rechner (außer Mail)

Ein guter Backup: Seagate „Backup Exec“

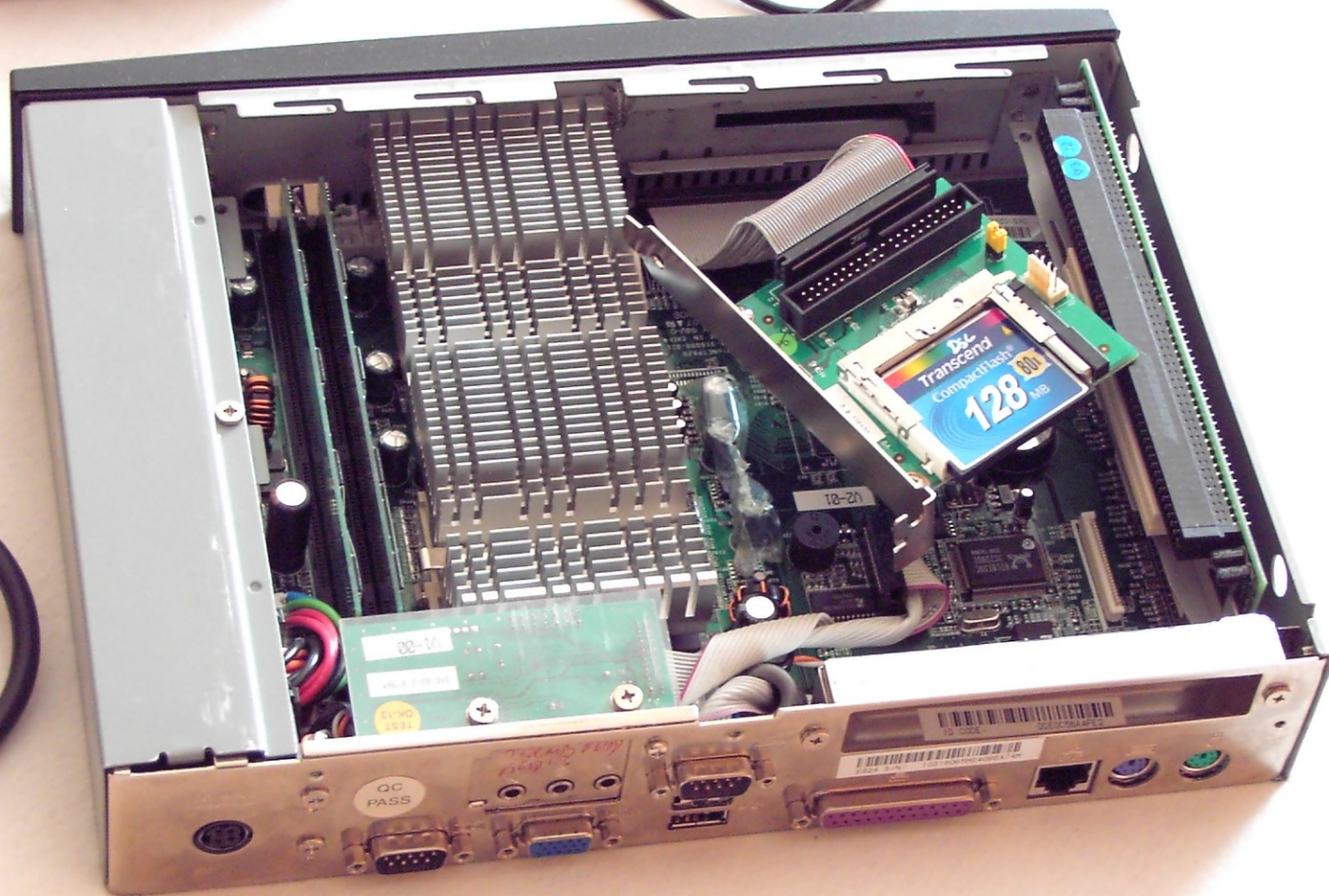
- Produkt für Windows NT und andere (ca. 1997)
- Erzeugt vollständige Sicherung von „C:“ auf Band
- ... und passende Boot-Floppy mit „fdisk“ sowie Restore-Programm
- Erfolgreich benutzt zum Schrumpfen eines vorinstallierten Windows 95

Vielleicht ähnlich ?

HP „One-button recovery“, andere ?

Ein perfekter Backup: IBM AIX „mksysb“

- Bestandteil von AIX (Basis) seit Mitte der 90er (mind.)
- AIX benutzt (immer) „logical volume management“ und kann Filesysteme („jfs“) im laufenden Betrieb vergrößern (einschließlich root-FS)
- Sicherung auf Band (QIC, Exabyte, ...)
- „mksysb“ erzeugt ein **bootfähiges** (Firmware !) Band mit der gesamten „root volume group“
- Separates Tool „backupvg“ für Daten-VGs
- Band-Boot + Restore reorganisiert die „rootvg“ (zusammenhängende Filesysteme)
- ... und kann dabei LVs / FSe kleiner anlegen



Hardware-Beschaffung

„Igel-J“ (www.igel.com) als gebrauchtes Gerät:

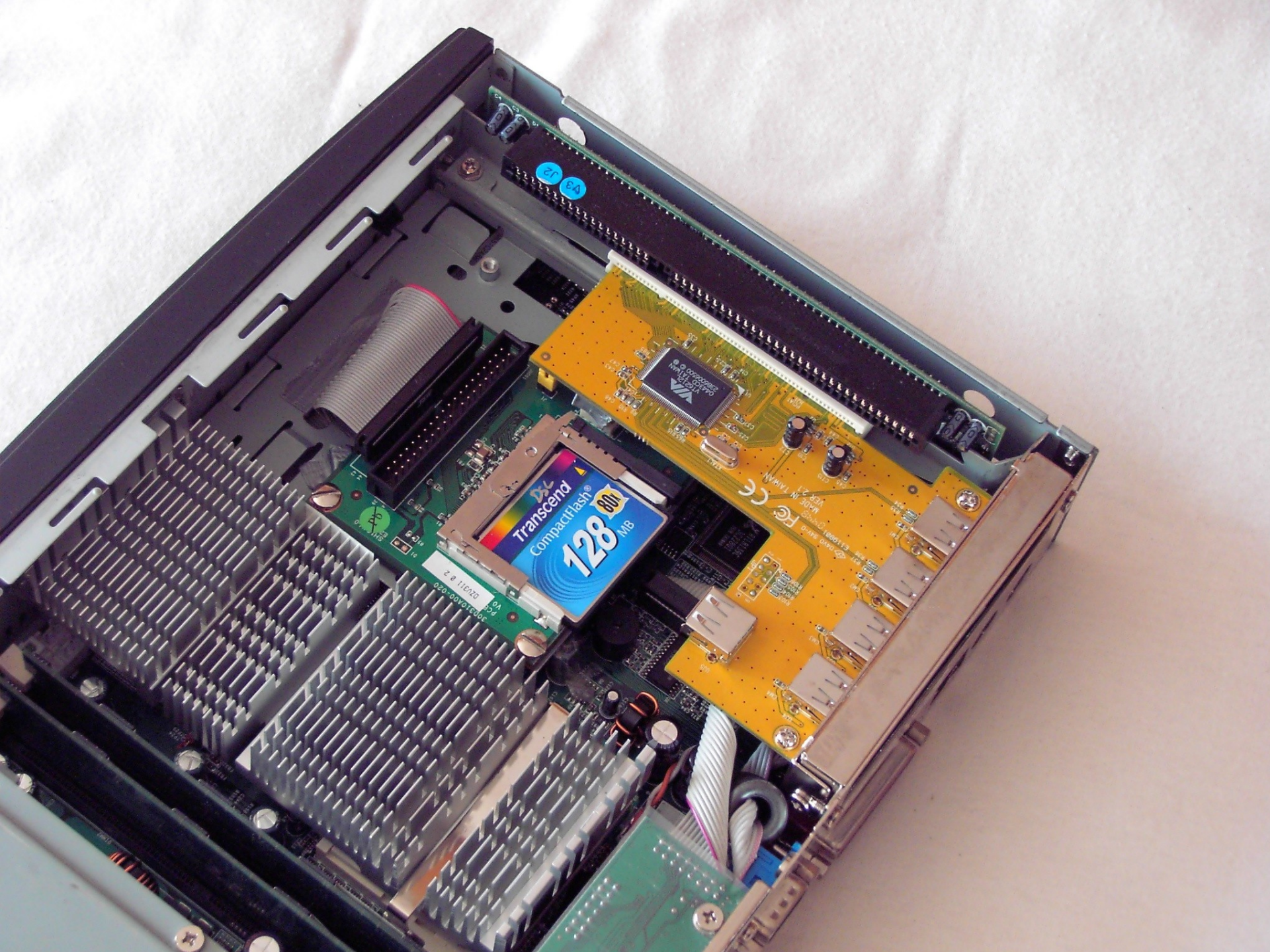
- Thin Client, 800 MHz VIA CPU, 256 MB RAM (max!), Compact Flash als System-Platte
Schnittstellen: VGA, Tastatur + Maus (PS/2),
2* seriell, parallel, RJ45, 2* USB 1.1
- Riser-Karte mit kombiniertem EISA/PCI-Steckplatz
- kein Lüfter, externes Netzteil (12 V, 5 A)

einschließlich 128 MB Flash (DSL „Damn Small Linux“)
und freier 1 GB Flash:

105 € (September 2007, „Combär“ Berlin)

Erfahrungen mit Igel-J: Start

- Debian „stable“ (4.0) installiert (Textmodus ohne X):
~ 500 MB der 1 GB Flash-Karte
(von externer CD via USB, komplettiert über Internet)
- Kein Swap-Device auf Flash konfigurieren, „noatime“!
- „rsnapshot“ installiert und konfiguriert
- Externe Platten beschafft: 400 GB Samsung (SATA)
in Revoltec eSATA/USB-Gehäuse
- Anschluss über on-board USB:
Arbeitet, aber langsam (USB 1.1)



Erfahrungen mit Igel-J: Beschleunigung

- PCI-USB-Karte in Riser-Karte, Platte angeschlossen: „reset“ während des Schreibens
- Händler: „VIA-Chips sind dafür bekannt“ :-)
- Neue Karte mit NEC-Chip: Arbeitet, schnell, aber (reproduzierbar !) Defekte im File-System
- Andere Karte (MGA ?): Ebenso
- Tekram SCSI-Karte und Exabyte Bandlaufwerk: Datenfehler beim Kontroll-Lesen

**=> Riser-Karte nicht zu benutzen,
Maschine ist auf on-board USB beschränkt**

Erfahrungen mit Igel-J: Vergrößerung

- Backup (lokal + remote) arbeiten mit on-board USB
- Backup Filesystem (LVM, ext3) auf 30 GB vergrößert für weitere Datenbereiche (dirvish „vaults“):
Maschinen-Crashes während „fsck -f“ bei Speicher-Anforderung (reproduzierbar)
- Swap-Space (auf Daten-Platte): „fsck“ erfolgreich
- Vergrößert auf 50 GB, und größere Backup-Jobs:
Maschinen-Crashes während „rsync“ (reproduzierbar),
wieder bei Speicher-Anforderung

=> 256 MB RAM sind zu wenig für meine Jobs

Wechsel auf andere Maschine

- Backup nicht auf der Haupt-Maschine (besserer Schutz)
- Pegasos (512 MB RAM) auf Debian 4.0 umgestellt, LVM installiert
- PCI-USB-Karte (USB 2.0, NEC-Chip) installiert
- Backups (lokal + remote) laufen erfolgreich
- Problem: Das Gehäuse erlaubt keine „full size“ Karten, „slim line“ nicht verfügbar (weder USB noch eSATA)