

DATEIDIFFERENZEN MIT XDELTA & CO. ERMITTELN

Binäres Puzzlespiel

Beim Backup oder bei der Verteilung von Archiven helfen Werkzeuge wie *rsync*, *jigdo* oder *xdelta*, nur Änderungen zu übertragen oder beim Empfänger bereits vorhandene Dateien auszunutzen. So bleibt das Transfervolumen klein und wertvoller Festplattenplatz lässt sich einsparen.

VON **MATTIAS SCHLENKER**

Backups sollen in der Regel nicht nur dazu dienen, einen bestimmten Systemzustand wiederherzustellen, sondern für Auswahl zwischen mehreren Zuständen in der Vergangenheit sorgen. Denn nicht immer wird ein Schaden sofort entdeckt – wenn beim gestrigen Backup des Webservers der Hacker schon drin war, nützt das schönste Vollbackup fast nichts, da das Zurückspielen den Schaden nicht repariert.

Die richtige Backup-Strategie plant derartige Zwischenfälle ein und erlaubt den Zugriff auf mehrere ältere Versionsstände. Damit das Volumen des Backups nicht proportional mit der Anzahl der vorhandenen Zwischenstände ansteigt, sind intelligente Werkzeuge sinnvoll, welche die Unterschiede zwischen zwei

Dateien erkennen und in eine separate Datei speichern.

Datei 1 + Änderungen = Datei 2

Ein Werkzeug, das zwei Dateien vergleicht und die Informationen, um aus der ersten Datei die zweite zu machen in einer Delta-Datei, speichert, ist *Xdelta*.

Es existiert in zwei Versionen: *Xdelta 1* wurde im Rahmen einer Diplomarbeit erstellt und versuchte, effiziente Strategien zum Vergleich von Dateien zu finden, bei denen gemeinsam vorhandene Fragmente auch in anderer Reihenfolge vorhanden sein können.

Eine Folge dieses Ansatzes war, dass bei *Xdelta 1* sowohl Ausgangsdatei als auch die Differenzen komplett in den virtuellen Spei-

cher – also Arbeitsspeicher und Auslagerungsspeicher passen müssen. *Xdelta 1* ist deshalb auf gegenwärtiger Hardware meist nur bis zu Dateigrößen von ein bis zwei Gigabyte praktikabel.

Besonders effizient arbeitet *Xdelta* auf Textdateien oder unkomprimiert aggregierten Textdateien. Schöne Beispiele hierfür sind die Dumps einer MySQL-Datenbank oder Tar-Archive mit Quellcodes oder Logdateien. Da *Xdelta* im Gegensatz zum meist für Patches eingesetzten *diff* oder zu Versionsverwaltungssystemen nicht zeilenweise arbeitet, sondern tatsächlich die größten gemeinsamen Abschnitte sucht, sind die resultierenden Delta-Dateien oft kleiner als komprimierte Patch-Dateien.

Sehr schön lässt sich die Anwendung von *Xdelta* auf MySQL-Datenbank-Dumps demonstrieren:

```
xdelta delta dump-montag.sql dump-
->dienstag.sql dump-montag-dienstag.
->xdelta
```

Gerade bei den typischen Anwendungsfällen von Content-Management-Systemen oder Onlineshops, bei denen die Anzahl der Änderungen moderat sind und eher Datensätze (neue Bestellungen, Kommentare) hinzukommen als modifiziert werden, lohnt sich *Xdelta*. Zudem komprimiert *Xdelta* die gespeicherten Änderungen.

Nach dem Anwenden von *Xdelta* können Sie die Datei *dump-dienstag.sql* entsorgen und die Datei *dump-montag.sql* komprimieren. Benötigen Sie den Dump vom Dienstag wieder, stellen Sie ihn mit

```
xdelta patch dump-montag-dienstag.
->xdelta
```

wieder her.

Xdelta sucht im aktuellen Arbeitsverzeichnis nach der Ausgangsdatei. Ihr Name sowie der Name der Zieldatei sind im Delta gespeichert und müssen nur dann als Parameter angegeben werden, wenn beispielsweise die Zieldatei in einem anderen Verzeichnis erzeugt werden soll oder die Ausgangsdatei anders be-

Weblinks

- [Homepage von Jigdo](http://www.atterer.net/jigdo)
www.atterer.net/jigdo
- [Homepage von rsync](http://samba.anu.edu.au/ftp/rsync)
<http://samba.anu.edu.au/ftp/rsync>
- [Homepage von Xdelta](http://www.xdelta.org)
www.xdelta.org



Platzsparend: Bei SQL-Dumps oder Logdateien erzeugt Xdelta schnell und effizient Differenzdateien.



Bei kleineren Dateien (bis zu mehreren Hundert MByte) arbeitet Xdelta1 effizienter, benötigt aber mehr Arbeitsspeicher.

nannt ist. Kritisch ist der Einsatz von Xdelta, wenn die zu vergleichenden Dateien teilweise komprimiert sind, wenn es sich beispielsweise um ISO-Images mit Software-Paketen handelt. Xdelta versucht in diesem Fall, die komprimierten Abschnitte zu entpacken, was häufig in riesengroßen Xdeltas resultiert. Mit der Option *--pristine* deaktivieren Sie dieses Verhalten, komprimierte Abschnitte bleiben unangetastet und werden „roh“ mit den entsprechenden Abschnitten der Zieldatei verglichen.

Ein großer Vorteil von Xdelta ist, dass das Anwenden des Deltas auf eine Datei weit weniger Speicher und Rechenleistung benötigt als dessen Erzeugung. Xdelta ist deshalb sehr gut geeignet, um kleine Korrekturen an großen Dateien, beispielsweise Änderungen an ISO-Images, Zeit und Traffic sparend zu verteilen.

Schon seit einer Weile ist Xdelta 3 erhältlich, der designierte Nachfolger mit deutlich erweitertem Funktionsumfang. Die größte Neuerung ist, dass nun nicht mehr die gesamte Ausgangsdatei im Arbeitsspeicher vorgehalten werden muss, wodurch sich Xdelta 3 auch für den Vergleich großer Dateien eignet.

Allerdings findet Xdelta 3 dadurch nicht immer die vergleichbaren Abschnitte, wenn sich die Reihenfolge der Fragmente verändert hat. Um das Verhalten des „alten“ Xdeltas zu er-

reichen, ist es deshalb oft nötig, mit den Fenstergrößen zu experimentieren. Auch die Syntax hat sich geändert. Die Erstellung unseres MySQL-Deltas erfolgt mit folgendem Befehl:

```
xdelta3 -s dump-montag.sql dump-
->dienstag.sql dump-montag-dienstag.xd3
```

Für eine spürbar höhere Kompression sorgen zwei zusätzliche Schalter:

```
xdelta3 -9 -S djw -s dump-montag.sql
->dump-dienstag.sql dump-montag-
->dienstag.xd3
```

Beim Anwenden eines Xdeltas auf einer Datei sind jetzt Name der Ausgangs-, sowie der Zieldatei anzugeben:

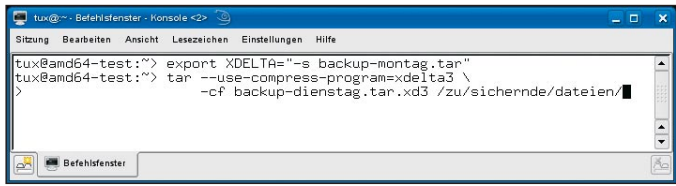
```
xdelta3 -d -s dump-montag.sql dump-
->montag-dienstag.xd3 dump-dienstag.sql
```

Bei einem Test, bei dem es darum ging, aus einem sidux-Lite-Image ein sidux-Full-Image zu erzeugen, schnitt Xdelta 3 zunächst miserabel ab. Grund war das Eingabefenster von 32 MByte. Erst als wir dieses mit dem Parameter *-B* auf 200 MByte erhöhten, lieferte Xdelta 3 ein ordentliches Ergebnis, das aller-

dings immer noch 71 MByte (371 MByte statt 300 MByte) über dem Delta von Xdelta 1 lag. Xdelta 3 kann seine Stärke also erst dann ausspielen, wenn die größere von Ausgangs- oder Zieldatei nicht mehr in den virtuellen Speicher passt.

rsync: Xdelta für das Netzwerk

Sollen zwei Dateien auf verschiedenen Rechnern synchronisiert werden und steht nur eine relativ schmalbandige Verbindung zur Verfügung, wäre eine angemessene Vorgehensweise, auf Rechner A aus einer alten und einer neuen Datei ein Xdelta zu erzeugen, dieses auf Rechner B zu kopieren und dort anzuwenden. Was bei einzelnen Dateien und relativ kleinen Änderungen funktionieren mag, resultiert in einem schier unglaublichen Aufwand, wenn ein ganzer Ast des Verzeichnisbaumes synchronisiert werden soll. Für diese Aufgabe existiert mit rsync eine Art „Xdelta fürs Netzwerk“.



Quelldateien können Xdelta3 per Umgebungsvariable übergeben werden. Das ermöglicht es, Xdelta3 bei tar als Kompressionsprogramm anzugeben und so Deltas ohne Umweg über temporäre Dateien zu generieren.

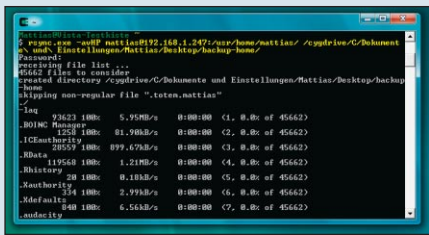
Und für Windows?

Alle vorgestellten Tools sind auch für Windows erhältlich, allerdings müssen unter dem Redmonder System kleinere Abstriche gemacht werden.

Jigdo für Windows bringt eine wenige MByte große Umgebung mit einigen Linux-Tools für Windows mit. Entpackt man Jigdo jedoch sauber beziehungsweise trennt *jigdo-lite.bat* nicht von den anderen Dateien, lässt sich Jigdo gut unter Windows einsetzen. Wir verwenden Jigdo auf der Heft-DVD zusammen mit einer weiteren Batch-Datei, um unter Windows originale ISO-Images von Distributionen auf Heft-

DVD wiederherzustellen. Rsync ist nur über die Unix-Umgebung Cygwin (www.cygwin.com) unter Windows verfügbar. Bei der Installation von Cygwin sollten Sie darauf achten, dass auch OpenSSH installiert wird, da Rsync diese Remote-Shell standardmäßig nutzt. Durch die Cygwin-Emulationsschicht sind geringe Leistungseinbußen gegenüber nativen Linux-Anwendungen möglich.

Xdelta und Xdelta 3 sind als native Windows-EXE-Dateien ohne weitere Abhängigkeiten verfügbar. Beide haben unter Windows ähnliche Performance-Werte wie un-



Auch unter Windows: Cygwin macht rsync nutzbar, allerdings muss man sich an seltsame Pfadnamen gewöhnen.

ter Linux, was Xdelta zu einem echten Plattform-übergreifenden Werkzeug macht.

```

mattias@curium: /tmp
*** 57 > cat wordpress-blog_rootserverexperiment.de-200706* > /dev/null
mattias@curium: /tmp
*** 58 > time xdelta delta wordpress-blog_rootserverexperiment.de-20070605-232300.sql \
20070605-20070612.xdelta
real    0m0.355s
user    0m0.295s
sys     0m0.068s
mattias@curium: /tmp
*** 59 > time xdelta3 -9 -S djw -s wordpress-blog_rootserverexperiment.de-20070605-232300.sql \
20070605-20070612.xd3
real    0m1.241s
user    0m0.760s
sys     0m0.088s
mattias@curium: /tmp
*** 60 >

```

Auch die Geschwindigkeitswertung geht bei kleinen Dateien an *Xdelta 1*.

Über eine SSH-Verbindung vergleicht *rsync* Dateien und Verzeichnisse und überträgt Änderungen häppchenweise. Dabei geht *rsync* sehr geschickt vor: Stimmen sowohl Dateigröße als auch Zeitstempel zwei gleichnamiger Dateien überein, gelten diese als identisch und werden ignoriert, ist dies nicht der Fall, werden Prüfsummen von Abschnitten gebildet und verglichen. Erst wenn diese nicht übereinstimmen, werden die Differenzen übertragen und die Datei auf der Empfängerseite wird auf den Stand der Senderseite gebracht. Erfolgt die Übertragung komprimiert, ist gerade bei kleineren Modifikationen die Lesegeschwindigkeit von Festplatte der Flaschenhals. Sehr praktisch ist die Verwendung des SSH-Tunnels: Auf beiden Seiten muss lediglich ein *rsync* vorhanden und im Pfad sein, Sie müssen jedoch keinen eigenen Serverdaemon einrichten, was auch aus

Ein *rsyncd* kann den FTP-Server ersetzen und Anbietern viel Traffic ersparen, wenn beispielsweise aktualisierte DVD-Images vertrieben werden. Hier handeln wir uns durch einen Debian-Mirror und laden anschließend den Inhalt des DVD-Verzeichnisses herunter.

sicherheitstechnischer Sicht sinnvoll ist. Der Standardaufruf für ein rekursives, komprimiertes Kopieren von einem Server –

hier die Sicherung von */var* – in ein lokales Backup-Verzeichnis lautet:

```
rsync -avzHP user@server:/var/
➔ /backup/var/
```

Beim Kopieren zum entfernten Rechner sind lediglich Quelle und Ziel zu vertauschen:

```
rsync -avzHP /backup/var/ user@server:/
➔ var/
```

Unser Beispiel verwendet auch die Parameter *-H* zur Identifikation „harder Links“ sowie *-P* (für *progress*, also Fortschritt, und *partial*, den partiellen Transfer von Dateien), die nicht immer notwendig sind. Damit auf Senderseite nicht mehr vorhandene Dateien gelöscht werden, können Sie den zusätzlichen Parameter *--delete* verwenden. Dieser ist aber gefährlich, wenn Sie beim Ziel einen Fehler machen: Befinden sich im angegebene-

```

mattias@curium: /usr/home/mattias
*** 17 > rsync ftp.hosteurope.de::debian-cd/
MOTD: Welcome to Ftp.hosteurope.de
MOTD: This is the public mirror server of Host Europe GmbH
MOTD:
MOTD: Have fun!
drwxr-xr-x 4096 2007/06/02 22:31:34 .
lrwxr-xr-x 0 2007/04/09 04:15:02 current
drwxr-xr-x 4096 2007/04/08 04:15:51 4.0_r0
drwxr-xr-x 16 2005/05/23 18:59:12 project
mattias@curium: /usr/home/mattias
*** 18 > rsync ftp.hosteurope.de::debian-cd/4.0_r0/
MOTD: Welcome to Ftp.hosteurope.de
MOTD: This is the public mirror server of Host Europe GmbH
MOTD:
MOTD: Have fun!
drwxr-xr-x 4096 2007/04/08 04:15:51 .
-rw-r--r-- 0 2007/04/08 21:24:11 Get-ISOs-with-HTTP!
drwxr-xr-x 88 2007/04/08 04:15:51 alpha
drwxr-xr-x 4096 2007/04/08 04:15:51 amd64
drwxr-xr-x 88 2007/04/08 04:15:51 arm
drwxr-xr-x 88 2007/04/08 04:15:51 hppa
drwxr-xr-x 4096 2007/04/08 04:15:51 i386
drwxr-xr-x 88 2007/04/08 04:15:51 ia64
drwxr-xr-x 88 2007/04/08 04:15:51 mips
drwxr-xr-x 88 2007/04/08 04:15:51 mipsel
drwxr-xr-x 48 2007/04/08 04:15:51 multi-arch
drwxr-xr-x 88 2007/04/08 04:15:51 powerpc
drwxr-xr-x 88 2007/04/08 04:15:51 s390
drwxr-xr-x 88 2007/04/08 04:15:51 source
drwxr-xr-x 48 2007/04/08 04:15:51 sparc
drwxr-xr-x 24 2007/04/07 15:13:05 trace
mattias@curium: /usr/home/mattias
*** 19 > rsync -xwP ftp.hosteurope.de::debian-cd/4.0_r0/1386/iso-dvd/ ./
MOTD: Welcome to Ftp.hosteurope.de
MOTD: This is the public mirror server of Host Europe GmbH
MOTD:
MOTD: Have fun!
receiving file list ...
3 files to consider
./
  debian-4.0-r0-1386-DVD-1.iso
  100938171 2% 1,40M/s 0:53:20

```

nen Pfad wichtige Dateien, werden diese ohne Rückfrage gelöscht:

```
rsync --delete -avzHP user@server:/var/
➔ /backup/var/
```

Mit dem Parameter *-x*, der das Überqueren von Mountpoints verhindert, kann *rsync* wunderbar dazu verwendet werden, vollständige Backups von Linux-Rechnern anzulegen. Das kann während des laufenden Betriebs passieren, allerdings sollten Sie dafür Daemons stoppen, die schreibend auf Dateien zugreifen: Insbesondere Datenbanken und Mail-systeme sind auf konsistente Dateien angewiesen, die sich während der Backup-Proze-

Differentielle Sicherung einer MySQL-Datenbank

⦿ Datenbankdumps, die unkomprimierte Textdateien sind, eignen sich besonders gut für die Bildung von Differenzen. Im folgenden Beispiel zeigen wir, wie mit den Delta-Dateien die Sicherungen vieler Tage vorgehalten werden können. Als Grundlage für die Deltas einer Woche dient ein Dump von Sonntag, dem 6. Mai. Die wochentags erstellten Dumps werden gleich nach der Erstellung der Deltas zum Sonntags-Dump gelöscht. Am Freitag, dem 11. Mai ergibt sich damit folgendes Bild:

```

dump-0506.sql
dump-0506-0507.xdelta
dump-0506-0508.xdelta
dump-0506-0509.xdelta
dump-0506-0510.xdelta
dump-0506-0511.xdelta

```

Am nächsten Sonntag soll wieder eine vollständige Sicherung als Basis für die Deltas

der nächsten Woche dienen. Der Trick besteht nun darin, ein Delta zu erstellen, das aus dem Dump vom 13. Mai den Dump vom 6. Mai erstellt, im Gegensatz zu den vorwärts gewandten täglichen Deltas also rückwärts gewandt ist. Am Montag dem 14. Mai ergibt sich folgendes Bild:

```

dump-0506-0507.xdelta
dump-0506-0508.xdelta
dump-0506-0509.xdelta
dump-0506-0510.xdelta
dump-0506-0511.xdelta
dump-0506-0512.xdelta
dump-0513.sql
dump-0513-0506.xdelta
dump-0513-0514.xdelta

```

Möchte man am Dienstag, dem 22. Mai, den Datenbankdump vom 7. Mai wiederherstellen, entsteht zunächst mit

```
xdelta patch dump-0520-0513.xdelta
```

der Sonntags-Dump vom 13. Mai, es folgt der Sonntags-Dump vom 6. Mai:

```
xdelta patch dump-0513-0506.xdelta
```

Aus diesem wiederum entsteht die gesuchte Datei:

```
xdelta patch dump-0506-0507.xdelta
```

Bei den meisten Content-Managementsystemen kommen eher Datensätze hinzu, als dass Veränderungen an bestehenden Daten vorgenommen werden. Das hält das rückwärts gewandte Xdelta kompakt, weil darin nur vermerkt sein muss, welche Abschnitte fehlen. Brennt man alle paar Monate den Ausgangsdump und die entstandenen Xdeltas auf CD oder DVD, lassen sich gerade beim Umfang typischer Content-Managementsysteme mit wenig Lagerplatz und Zeit tägliche Dumps von vor Monaten oder Jahren wiederherstellen.

dur nicht geändert haben. Beim Restore, das Sie beispielsweise mit einer Knoppix-CD vornehmen können, werden alle Zugriffsrechte und Zeitstempel wiederhergestellt. Problematisch ist in Einzelfällen lediglich der fehlende Inhalt von `/dev` beim Systemstart, der später von `udev` oder `devfs` überschattet wird. Sollten Sie auf dieses Problem stoßen, genügt es meistens, den Inhalt von `/dev` einer auf Platte installierten, aber gerade nicht gebooteten Linux-Installation ebenfalls mit `rsync` zum wiederherzustellenden Rechner zu kopieren. `rsync` ist übrigens nicht auf die Netzwerknutzung beschränkt. Lokal ist es ein besseres `cp`, das bei der Sicherung auf externe USB-Festplatte das erneute Kopieren vorhandener und nicht veränderter Dateien vermeidet. Hinsichtlich Transferrate und Prozesslast vieler USB-Treiber/Chipsätze bedeutet dies oft eine große Zeitersparnis:

```
rsync -avHP /home/mattias/ /media/sdb1/
↳ backup/mattias/
```

Dank der Cygwin-Umgebung für Windows (www.cygwin.com) ist `rsync` für das Redmon-



```
mattias@curium: /tmp/
*** 8 > rsync -avxHP --delete root@meinserver:/ /data/backup/server/hda2/
```

Ideal für die partitionsweise Sicherung: Der Parameter `-x` verhindert, dass Mountpoints übersprungen werden, `--delete` löscht auf Empfängerseite nicht mehr vorhandene Dateien. Als `root` ausgeführt, bleiben alle Berechtigungen erhalten.

der System verfügbar, wo es sich wunderbar mit Batch-Dateien automatisieren lässt. Ungewohnt ist auf dieser Plattform allenfalls, dass die Windows-Laufwerksbuchstaben auf `/cygdrive/C/` etc. abgebildet werden. Einen Haken hat die Sicherung mit `rsync` auf Windows-Zielrechnern jedoch: Windows kennt keine Hardlinks, keine Linux-Device-Files und handhabt Linux-Nutzer- und Gruppen nur unzureichend.

Damit ist `rsync` zwar ein ideales Werkzeug, um vom Windows-PC aus die Nutzdaten eines Webverzeichnis mit den Rechten eines unprivilegierten Nutzers zu sichern, aber nur bedingt dazu geeignet, eine Vollsicherung mit Root-Rechten vorzunehmen. Allerdings eignet sich `rsync` auch für den Datei-

austausch über das Internet, denn es lässt sich als Ersatz für einen FTP-Server nutzen.

rsync als Server

Dazu wird das Tool unter Angabe eines bereitzustellenden Verzeichnisses im Daemon-Mode gestartet. Möchte nun jemand zur Verfügung gestellte Dateien herunterladen, kann er dies in FTP-typischer Manier tun. Zum Tragen kommen `rsync`s Vorteile aber dann, wenn Teile des Verzeichnisbaums oder einzelne Dateien zumindest teilweise vorhanden sind. So kann `rsync` beispielsweise dazu benutzt werden, einen lokalen Paket-Mirror zu pflegen: Ganz automatisch werden nur neue Pakete heruntergeladen. Auch wenn Updates zwischen Dateiversionen eher klein ausfal-

KNOW-HOW IST BLAU.

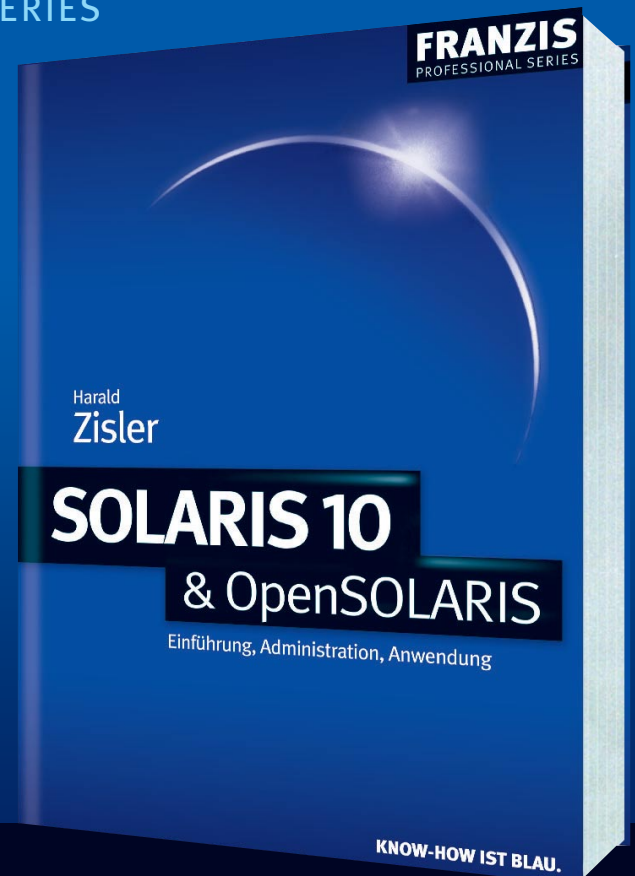
PROFESSIONAL SERIES

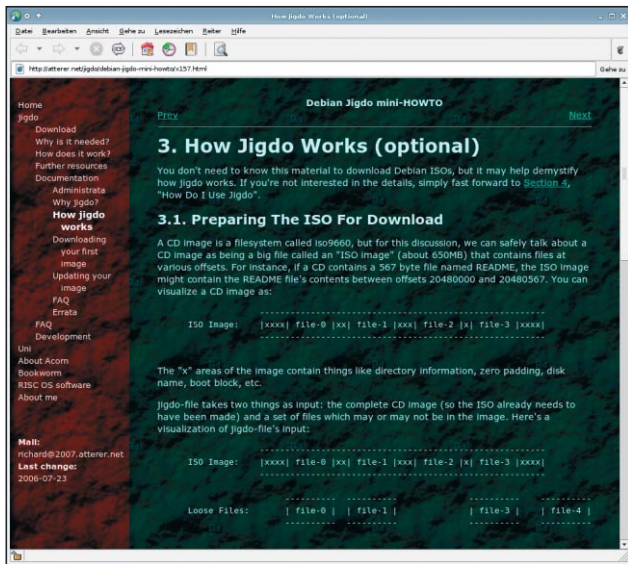
SOLARIS 10 & OpenSOLARIS

Einführung, Administration, Anwendung

Das legendäre Betriebssystem Solaris von Sun Microsystems steht seit 2005 unter freier Lizenz und damit jedem interessierten Anwender kostenlos zur Verfügung. Die Zahl der portierten Anwendungen für Desktop- und Servereinsatz wächst seitdem genauso rapide wie die Fangemeinde. Mit diesem Buch bietet Top-Autor Harald Zisler einen praxisorientierten Einstieg in die produktive Arbeit mit Solaris & OpenSolaris.

ISBN 978-3-7723-7297-1
EUR 39,95





Das Prinzip ist einfach: Ein DVD-Image kann als Konkatenation der enthaltenen Dateien gesehen werden. Jigdo ordnet Sektionen des Images die enthaltenen Dateien zu.

len, kann *rsync* helfen, den Traffic zu verringern. Allerdings muss der Herunterladende dazu gebracht werden, die alte Datei umzubenennen, bevor er den „Sync“ startet – und selbst dann ist bei vielen Webservern die durch den Dateivergleich anfallende Prozesslast ein größeres Problem als Trafficlimits.

Puzzlespiel

Auf einer anderen Ebene als *rsync* und *xdelta* arbeitet *jigdo* – auf Deutsch „Puzzle“. Dem Tool wird eine zu erstellende Datei und der Name eines Verzeichnisses mit Einzeldateien, welche sich in der zu erstellenden Datei wiederfinden lassen, übergeben. Als resultierende Datei eignen sich ein ISO-Image oder eine Tar-Datei am besten. Im Idealfall enthält

das erzeugte Jigdo-Temp-plate nur ein Gerüst, in das dann die benötigten Dateien eingepasst werden. Liegen die einzupassenden Dateien lokal vor, werden Sie anhand Ihrer MD5-Prüfsumme identifiziert. Ist das nicht der Fall, erfolgt der Download von einem HTTP- oder FTP-Server. Jigdo ist damit ideal geeignet, aktualisierte Installations-DVDs herzustellen, wenn eine ältere Version der DVD bereits vorliegt. Auch die Erstellung von Installations-CDs aus DVDs oder umgekehrt ist eine Domäne des im Debian-Umfeld entwickelten Tools. Um Jigdo anzu-

wenden, ist zumindest eine Jigdo-Datei notwendig, in der die Download-Pfade und Informationen über die zu erzeugende Datei gespeichert sind. Als Aufruf genügt dann:

```
jigdo-lite datei.jigdo
```

Jigdo wird zuerst überprüfen, ob ein Template vorhanden ist und dieses gegebenenfalls heruntergeladen. Es fragt dann nach einem lokalen Pfad, in dem lokal vorhandene „Puzzle-teile“ gesucht werden können und identifiziert diese anhand ihrer MD5-Summe. Ist die Identifizierung abgeschlossen, beginnt der Zusammenbau der Datei aus den Einzelteilen. Leider ist die Erstellung von Jigdo- und Template-Dateien etwas aufwändiger und die Dokumentation auf Hans Atterers Webseite

etwas spärlich. Um aus den Dateien im Verzeichnis */build* den Tarball *archiv.tar* (der natürlich viele in */build* enthaltene Dateien verwenden muss) zu erstellen, rufen Sie Jigdo wie folgt auf:

```
jigdo-file make-template --
image=archiv.
↳tar /build
```

Streben Sie an, Jigdo-Dateien weiterzugeben und zum automatischen Zusammenbau von Images oder Tar-Archiven zu verwenden, können Sie lokale Pfade auf Serverpfade mappen. Dazu muss ein eindeutiges Label (hier „Pub“) gewählt werden:

```
jigdo-file make-template --label Pub=/
↳build --uri Pub=
http://mein.server.
↳xyz/ --image=archiv.
↳tar /build
```

Der Autor dieses Artikels verwendet Jigdo, um auf Basis einer regelmäßig geernteten Debian-Musterinstallation mit *tar* gesicherte halb-fette Clients zu archivieren, bei denen zwar viel Anwendungs-Software installiert ist und sehr verschiedene Konfigurationen verwendet werden, aber eben keine Nutzdaten gespeichert werden - die werden per NFS zugemountet. So spart Jigdo bei jedem Tarball den Umfang der gemeinsam genutzten Dateien. Leider müssen diese ungepackt vorliegen, was sich elegant umgehen lässt, wenn man ein komprimiertes *Squashfs*-Image erstellt und dieses mit dem entsprechenden Kernelmodul via Loopback-Interface mountet. Der Zugriff auf die gemeinsamen Dateien kann damit wahlfrei erfolgen, und die Dekompression findet elegant im Hintergrund statt. jkn

Welche Dateitypen eignen sich für die Erstellung von Deltas?

🔍 **Tar-Archive:** Es kommt darauf an, was drinnen steckt. Zwischen zwei Archiven mit ähnlichem Inhalt und einigen leicht geänderten Dateien lassen sich meist sehr effizient mit Xdelta Differenzdateien erstellen. Sind komprimierte Dateien enthalten, ist gegebenenfalls der Schalter *--pristine* zu verwenden. Jigdo ist dann sinnvoll, wenn sich nur wenige Dateien von den Dateien im Verzeichnis mit den Puzzleteilen unterscheiden, sonst kann bereits bei kleinen Unterschieden eine vorhandene Einzeldatei nicht als Puzzlestück zum Einsatz kommen. Tar-Archive lassen sich meist sehr gut mit *rsync* synchronisieren.

🔍 **Komprimierte Dateien:** *zip*-Dateien oder mit *gzip* schwach komprimierte Dateien mit einem kurzen Index erlauben durchaus den Einsatz der vorgestellten Tools. Praktisch unmöglich ist ihr Einsatz auf Formaten wie *bzip2*, *rar* oder *7zip*, bei denen bereits ein verändertes Bit zwischen Ausgangs- und Zieldatei zur Folge hat, dass nach der Komprimierung praktisch keine Gemeinsamkeiten mehr feststellbar sind.

🔍 **Festplatten-Images:** Mit *ntfsclone* erstellte Images, die nur belegte Blöcke enthalten, lassen sich gut mit *rsync* synchronisieren, wenn der Fragmentierungsgrad nicht allzu hoch ist. In diesem Fall funktioniert

auch die Erstellung von Jigdo-Dateien aus einem Verzeichnis mit typischen Windows-Systemdateien. Der Einsatz von Xdelta ist angesichts der Dateigröße typischer Images oft nicht praktikabel.

🔍 **Textdateien:** Ideal für die Synchronisation mit *rsync* und die Erstellung von Xdeltas sind Verzeichnishierarchien, die z. B. Quellcodes oder Logfiles enthalten. Sollen Xdeltas erzeugt werden, ist es meist sinnvoll, diese auf unkomprimierten Archiven zu erstellen. Jigdo bietet sich seltener an, da bereits kleine Änderungen an einer Datei im Vorlagenverzeichnis bedeuten, dass sie nicht mehr verwendet werden kann.