

Inkrementelle Snapshots mit Bordmitteln

Im Kreis

Jan Bundesmann

Backups können durch die Vielzahl an Kopien eine Menge Platz benötigen. Wer mit wenig Aufwand inkrementelle Snapshots erstellen will, ist mit *rsnapshot* gut bedient.



Linux- oder besser Unix-Dateisysteme besitzen ein einfaches Mittel zur Datenduplizierung: den Hard Link. Dank seiner kann man etwa mit *cp* oder *rsync* inkrementelle Backups erzeugen und dadurch den Speicherverbrauch reduzieren. Möchte man beispielsweise den Ordner *source* im Verzeichnis *dest/backup1* sichern, kann man dafür *rsync* mit den Parametern *-a --delete --numeric-ids --relative --delete-excluded* verwenden. Durch die Wahl der Parameter löscht *rsync* im Quellverzeichnis nicht mehr vorhandene Dateien und aktualisiert veränderte, während es alles überspringt, was seit dem letzten Aufruf gleich geblieben ist.

Damit ältere Versionen der Ordnerstruktur erhalten bleiben, kann man das erste Backupverzeichnis in ein zweites kopieren. Übergibt man hierfür dem Befehl *cp* den Parameter *-al*, rekursiert dieser durch das Verzeichnis, kopiert die Dateien mit allen Attributen und erhält die interne Linkstruktur des Ordners. Die Dateien legt es nicht neu an, sondern erzeugt für sie einen Hard Link – gewissermaßen einen Snapshot. Im Laufe der Zeit verändern sich die Dateien und ein weiteres Backup steht an. Führt *rsync* die Übertragung veränderter Dateien durch, folgt es nicht dem Link der Zielfeile, sondern legt eine neue Datei an.

Bei regelmäßigen inkrementellen Snapshots wird speziell das Rotieren der Backupordner aufwendig. Auch helfen keine automatischen Regeln bei der Frage, welche Backups zu behalten sind. All diese Aufgaben übernimmt *rsnapshot*. Im Kern verwendet es die bisher vorgestellten Kommandos, so man sie nicht angepasst hat. Es ist bei allen gängigen Linux-Distribution in den Standardpaketquellen enthalten. Zudem lässt es sich unter den

meisten NAS-Betriebssystemen installieren. Binaries gibt es auch für die BSDs, dort funktioniert es genau wie unter Linux. Seine zentrale Konfiguration befindet sich in der Datei */etc/rsnapshot.conf*. Darin gibt man unter anderem das Zielverzeichnis des Backups an und wie viele Exemplare eines bestimmten Backuptyps aufgehoben werden sollen.

Die Rotation über die Hard Links erfordert einen lokalen Zugriff auf den Verzeichnisbaum. Deshalb muss *rsnapshot* die Sicherung vom Zielrechner aus durchführen. Zudem funktioniert die Speicherplatzoptimierung nur unter Dateisystemen, die Hard Links beherrschen. Externe Festplatten sind gegebenenfalls umzuformatieren. Holt sich *rsnapshot* die Dateien übers Netz, gibt man die Quelle im SSH-typischen Format *user@host:/path/* an. Für die Authentifizierung sollte man in diesem Fall einen Public Key verwenden.

Mehrstufig rotiert

Für die mehrstufige Rotation verwendet *rsnapshot* Retain Levels, die einer strengen Hierarchie folgen. Das tatsächliche Backup findet nur auf dem höchsten Level statt, die darunter nehmen lediglich an der Rotation teil. Zu jedem Level muss man angeben, wie viele Snapshots aufgehoben werden:

```
retain daily 7
retain weekly 5
retain monthly 6
```

Listing: /etc/crontab

```
# m h dom mon dow user command
00 02 * * * rsnapshot daily
30 01 * * 1 rsnapshot weekly
00 01 1 * * rsnapshot monthly
```

Den höchsten Level *daily* kann man direkt mit *rsnapshot daily* auf der Kommandozeile aufrufen. Dadurch löscht *rsnapshot* im Backupverzeichnis den Unterordner mit der höchsten Ordnung, hier *daily.5*. *daily.1* bis *daily.4* erhöht *rsnapshot* um jeweils eine Stufe und *daily.0* kopiert es per *cp -al* nach *daily.1*. Danach führt *rsnapshot* das Backup mit *rsync* auf *daily.0* aus.

Die anderen Backuplevels rotieren wie das erste, führen zum Schluss allerdings kein Backup durch, sondern kopieren lediglich das letzte Verzeichnis des vorherigen Levels. Der Befehl *rsnapshot weekly* würde *daily.6* nach *weekly.0* kopieren, *rsnapshot monthly* holt sich *weekly.4*.

Die Namen der Backupstufen liegen in der Hand der Administratoren, ebenso die Aufrufmethode und -häufigkeit. In vielen Tutorials finden sich die sprechenden Bezeichnungen *daily*, *weekly*, *monthly*. Die Manpage nennt sie *alpha*, *beta*, *gamma* – da Nutzer vom üblichen Rhythmus abweichen können. Wichtig ist ihre Reihenfolge in der Konfiguration. Auf Servern trägt man die Befehle direkt in */etc/crontab* ein. Ein Beispiel für tägliche, wöchentliche und monatliche Sicherungen zeigt das Listing. Für PCs bietet sich die spezielle *cron*-Option *@reboot* oder *anacron* an.

Trotz der einfachen Logik und schnellen Konfiguration von *rsnapshot* gibt es Gründe, weshalb sich verschiedene Backupwerkzeuge etabliert haben. Nicht in allen Fällen eignet sich *rsnapshot*, um ein einfaches, platzsparendes Backup anzulegen. Dienste wie Mailserver, Logging-Dienste oder Datenbanken, die ihre Daten in einzelnen großen Dateien ablegen, profitieren nicht davon, da bereits die Änderung eines einzigen Eintrags die gesamte Datei verändert. Für den entgegengesetzten Fall mit vielen kleinen Dateien bietet sich *rsnapshot* jedoch an. Bequem ist zudem, dass es kein proprietäres Speicherformat verwendet. In den Backupordnern liegen die Dateien in der ursprünglichen Struktur vor. Ein Restore besteht aus einem einfachen Zurückkopieren.

Auch wenn echte Sicherungen auf einen anderen Rechner gehören, lässt sich *rsnapshot* lokal verwenden, etwa wenn jemand mit Snapshots auf Verzeichnisebene einen Konfigurationszustand sichern will. So ließe sich etwas Ähnliches wie Copy on Write, das moderne Dateisysteme wie ZFS oder BTRFS implementieren, auch unter ext2/3/4 nutzen. (sun@ix.de)

Jan Bundesmann

ist der Systemadministrator von Mordor und beschäftigt sich mit Automatisierung im Rechenzentrum.

