

Für Systemadministratoren ist der Befehl *rsync* eine Art Schweizer Messer beim Dateitransfer. Weil *rsync* dank der Delta-Kodierung nur Änderungen an bereits vorhandenen Dateien überträgt, hält sich der Datenverkehr auch beim regelmäßigen Abgleich großer Verzeichnisse in Grenzen. Viele Backup-Programme setzen deshalb darauf auf.

Die grundlegende Befehlssyntax dürfte den meisten Administratoren vertraut sein. Im einfachsten Fall spiegelt man das Verzeichnis *~/Dokumente* nach *~/Backup*:

```
rsync -avP ~/Dokumente ~/Backup
```

Soll *rsync* das Verzeichnis *Dokumente* als Ganzes in das Verzeichnis *Backup* synchronisieren, also ein Verzeichnis */Backup/Dokumente* erzeugen, darf man keinen Slash an die Quelle anhängen. Mit angehängtem Schrägstrich würde es die Dateien aus dem Ordner *Dokumente* nach *Backup* synchronisieren. Die Meta-Option *-a* oder *--archive* setzt die Optionen *-rlptgoD* für den Normalbetrieb. Durch *-v* oder *--verbose* gibt sich *rsync* gesprächiger und *-P* setzt die Optionen *--progress* und *--partial*. *rsync* zeigt dann den Fortschritt an und erlaubt die Fortsetzung bei einem Abbruch, da es die bisher übertragenen Daten nicht verwirft.

## Schnell gelöscht – auch große Mengen

Fügt man die Option *--delete* hinzu, löscht *rsync* bei dieser 1:1-Synchronisation alle auf dem Ausgangssystem nicht oder nicht mehr vorhandenen Dateien auch auf dem Zielsystem. Wer hier fatale Fehler vermeiden will, sollte den ersten *rsync*-Lauf zusätzlich mit dem Argument *--dry-run* versehen, der den Vorgang nur simuliert.

Löschen kann *rsync* übrigens ausgesprochen effektiv. Bei Verzeichnissen mit mehreren hunderttausend Dateien bricht *rm \** unter Umständen mit einem Speicherfehler ab. Das kann man mit

```
find ./ -type f -exec rm {} \;
```

umgehen, verlangsamt das Ganze aber erheblich. Ohne *rm*, nur mit *find ./ -type f -delete* wird es dreimal schneller. Nochmal doppelt so schnell ist *rsync*, wenn es das zu löschende Verzeichnis mit einem leeren Verzeichnis synchronisiert:

```
mkdir leer
rsync -a --delete leer/ <Verzeichnis>
rmdir leer
```

Will man das Verzeichnis auf ein anderes System spiegeln, ergänzt man das Ziel um den Benutzer- und den Hostnamen:

## Synchronisation von Dateien

# Gleichverteiler

## Michael Plura

*rsync* kann mehr als nur Verzeichnisse synchronisieren. Auch beim schnellen Löschen und dem Erstellen leerer Verzeichnisbäume leistet es gute Dienste.

```
rsync -avP ~/Dokumente 7
                        <Benutzer>@<Server>:Backup
```

Über unsichere Verbindungen sollte man eine SSH dazwischenschalten, da *rsync* den Datenstrom nicht verschlüsselt. Zusätzlich empfiehlt sich gerade bei langsamen Verbindungen das Einschalten der Kompression mit *-z*:

```
rsync -avPz -e "ssh -l <Benutzer> -p 22" 7
~/Dokumente <Benutzer>@Server:Backup
```

Bei der Schätzung der Übertragungszeit hilft *--dry-run* nicht, aber *--itemize-changes*. Wer regelmäßig auf ein anderes System im LAN synchronisiert, sollte keine per Samba/CIFS oder NFS eingebundenen Dateisysteme verwenden, sondern *rsync* als Service starten, der dann am Port 873 lauscht (siehe [ix.de/ix1808109](http://ix.de/ix1808109)).

## Aus- oder eingeschlossen

Bestimmte Dateien, Dateitypen oder Verzeichnisse lassen sich durch das Argument *--exclude=\*.\*bak --exclude=temp* ausschließen. Verzeichnisse wie in diesem Fall *temp* sind relativ zum *rsync*-Quellpfad anzugeben. Solche wiederkehrenden Parameter kann man wie bei *tar* in eine Datei eintragen und so leichter pflegen und übergeben:

```
rsync -avP --exclude-from=exclude.txt
~/Dokumente ~/Backup
```

Die Datei *exclude.txt* enthält in diesem Beispiel zwei Zeilen mit den Einträgen *\*.\*bak* und *temp*. Umgekehrt lassen sich durch *--include=\*.\*conf* auch bestimmte Dateien wieder einschließen. Kombiniert man beides und schließt etwa Verzeichnisse ein, Dateien aber aus, lässt sich über



die allgemeine *--filter=-* Regel aus einer bestehenden Verzeichnisstruktur der leere Verzeichnisbaum extrahieren:

```
rsync -avP --filter="+ */" --filter="- *" 7
~/Dokumente ~/VerzStruktur
```

Eine Sonderform der Auswahl bildet die Option *file-from=files.txt*, mit der sich explizit eine Dateiliste relativ zum Quellpfad angeben lässt.

Befindet sich im Zielpfad nur eine Teilmenge der Dateien des Quellpfades und will man nur genau diese Dateien synchronisieren, lässt sich das mit *--existing* erreichen. Es werden keine zusätzlichen Dateien zum Ziel kopiert oder dort gelöscht. Die Negation *--ignore-existing* würde Quelle und Ziel synchronisieren, dabei aber keine schon im Ziel vorhandenen Dateien überschreiben.

Filtern kann man auch nach Dateigröße. Die Synchronisation einer statischen Webseite etwa zusammen mit ISO-Images zum Download lässt sich optimieren, indem man zunächst mit *--max-size=250K* die kleinen und danach mit *--min-size=250K --bwlimit=100* die großen Dateien mit Bandbreitenbeschränkung überträgt, wobei der maximale Durchsatz in KByte/s anzugeben ist. Mit *--min-size=1* verhindert man, dass *rsync* leere Dateien kopiert. Die *..size*-Werte lassen sich bequem auch als *1.5m* oder *4g-1* angeben. (sun@ix.de)

## Michael Plura

lebt in Schweden und ist freier Autor mit den Schwerpunkten IT-Sicherheit, Virtualisierung und freie Betriebssysteme.

Alle Links: [ix.de/ix1808109](http://ix.de/ix1808109)