Swappen über Netzwerk

Autor: Matthias Kleine (*kleine_matthias@gmx.de*)
Layout: Matthias Kleine (*kleine_matthias@gmx.de*)

Dieses Kapitel beschreibt die Einrichtung einer Auslagerungsdatei über das Netzwerk-Blockdevice. Dies ist nur unter speziellen Umständen sinnvoll oder notwendig, wie zum Beispiel auf Geräten, die nur Readonly-Filesysteme anbieten und über relativ wenig Speicher verfügen. Will man einem solchen Gerät mehr virtuellen Speicher gönnen, ist das Auslagern auf eine andere Maschine eine interessante Alternative.

Inhaltsverzeichnis

1 Einleitung

In manchen Situationen kann es sinnvoll oder notwendig sein, eine Swap-Partition auf einem entfernten Rechner einzurichten. Leider ist es nicht einfach möglich, zu diesem Zweck NFS zu verwenden. Zwar gibt es bereits seit den Kernelversionen 2.0.x Patches, welche dies ermöglichen sollen, doch auch mit einem Standard-2.4er Kernel scheitert der Versuch in den Tiefen von TCP/IP.

Glücklicherweise (und verblüffenderweise zugleich) gibt es eine Alternative. Diese befindet sich zwar laut Entwickler noch in einem experimentellen Stadium, funktionierte jedoch in den Versuchen des Autors auf Anhieb. Das gute Stück heißt "Network Block Device" und muß entweder in den Kernel einkompiliert sein oder als Modul vorliegen. Erstaunlicherweise fand sich das Modul bei mehreren Distributionen bereits unter /lib/modules, so daß (mit einem 2.2.x'er Kernel) ein einfaches

```
root@linux /root/ # modprobe nbd
```

genügte, um das Modul einzubinden. Wer dieses Glück nicht hat, wird nicht um eine neue Kernelkonfiguration herumkommen. Die richtige Option findet sich unter

Block Devices -> Network Block Device support

Natürlich müssen sowohl der Rechner, der das neue Swapdevice erhalten soll, als auch der entfernte Rechner die entsprechende Kernelunterstützung verwenden.

2 Konfiguration des entfernten Rechners

2.1 Systemseitige Konfiguration

Richten wir zunächst den Rechner ein, der den Swapspace zur Verfügung stellen soll. Durch das obige modprobe-Kommando haben wir NBD-Support eingebunden. Jetzt legen wir das neue Device an. Hierzu verwenden wir mknod:

```
root@linux /root/ # mknod /dev/nd0 b 43 0
```

/dev/nd0 ist der neue Devicename. b zeigt an, daß es sich um ein Blockdevice handelt. 43 ist die sogenannte "major number". Alle Devices sind unter Linux in Gruppen eingeteilt, denen jeweils besondere major numbers zugeordnet sind. Die einzelnen Geräte werden dann durch eine "minor number" unterschieden. Diese wird hier als das letzte Argument, also "0" an mknod übergeben. Das neue Device können wir uns nun anschauen:

```
root@linux /root/ # ls -l /dev/nd0

brw-rw-r--   1 root     root      43,   0 Feb 27 15:49 /dev/nd0
```

Major und minor number listet ls dort, wo normalerweise die Dateigröße zu finden ist. Das Device ist jetzt angelegt und kann verwendet werden. Für weitere Devices würden wir einfach eine fortlaufende minor number vergeben:

```
root@linux /root/ # mknod /dev/nd1 b 43 1
root@linux /root/ # mknod /dev/nd2 b 43 2
root@linux /root/ # mknod /dev/nd3 b 43 3
```

2.2 Installation und Konfiguration des NBD-Servers

Um eine Datei zu exportieren, kann der NBD-Server von Pavel Machek verwendet werden. Pavel ist auch der Autor des Network Block Devices. Das Programm kann über http://atrey.karlin.mff.cuni.cz/~pavel/nbd/nbd.html bezogen werden. Hier (und unter /usr/src/linux/Documentation/nbd.txt) finden sich übrigens auch weitere Informationen zum Thema.

Nach dem Download kompilieren wir den Server einfach mittels

```
root@linux /root/ # configure && make
```

Dabei sollten zwei Binaries ndb-server und nbd-client erzeugt werden. Auf dem Rechner, der den Swapspace zur Verfügung stellt, brauchen wir nur den nbd-server. Dieselbe Prozedur werden wir später auf dem Rechner wiederholen, der das neue Swapdevice erhalten soll. Auf diesem werden wir nur den nbd-client benötigen.

Nun erzeugen wir eine Datei, die wir später exportieren können. Hierzu verwenden wir die Spezialdatei /dev/zero, die uns beliebig viele Nullbytes liefert. Quick and dirty geht es so:

```
root@linux /root/ # cat /dev/zero > /tmp/swapfile
```

Um die Größe der Datei festzulegen, verwendet man allerdings besser dd für diesen Zweck, zum Beispiel so:

```
root@linux /root/ # dd if=/dev/zero of=/tmp/swapfile bs=1024 count=65536
```

Nun müssen wir dem nbd-server noch mitteilen, daß er diese Datei exportieren soll. Ein einfacher Aufruf ohne Parameter liefert uns folgendes Usage:

```
root@linux /root/ # nbd-server

Usage: port file_to_export [size] [-r]
       -r read only
```

Wir suchen also einen freien Port und geben diesen mit der zugehörigen Swapdatei an:

```
root@linux /root/ # nbd-server 1024 /tmp/delme

Export size recognized as 34254848.
```

Die Datei ist exportiert.

3 Konfiguration des lokalen Rechners

Die Einrichtung des Network Block Device geschieht analog zu derjenigen auf dem Entfernten Rechner: Einbinden der Kernelunterstützung, Anlegen der Devices mittels mknod und Kompilation des nbd-clients wurden bereits oben beschrieben. Fehlen noch die Verbindung zum nbd-server und das Aufsetzen des Swapdevice.

Die Verbindung zum entfernten Rechner stellen wir mittels des nbd-client her. Betrachten wir zunächst dessen Usage-Informationen:

```
root@linux /root/ # nbd-client

Usage: host port nbd_device -swap
```

Im wesentlichen werden also der Hostname des exportierenden Hosts, die Portnummer und ein vorhandenes Network Block Device verlangt. Die Option -swap hat der Autor bis zum jetzigen Zeitpunkt noch nicht verwendet, so daß wir diese einmal außer Acht lassen.

```
root@linux /root/ # nbd-client helios 1024 /dev/nd0
```

Damit haben wir eine TCP-Verbindung zu der zuvor exportierten Datei. Richten wir diese also nun als unser neues Swapfile ein:

```
root@linux /root/ # mkswap /dev/nd0

Setting up swapspace version 0, size = 34250752 bytes

root@linux /root/ # swapon /dev/nd0
```

Das war's. Schau'n wir mal nach:

```
root@linux /root/ # cat /proc/meminfo

        total:    used:    free:  shared: buffers:  cached:
Mem:  31657984 14843904 16814080        0    24576  9510912
Swap: 34250752  5124096 29126656
MemTotal:     30916 kB
MemFree:      16420 kB
MemShared:        0 kB
Buffers:         24 kB
Cached:        9288 kB
HighTotal:        0 kB
HighFree:         0 kB
LowTotal:     30916 kB
LowFree:      16420 kB
SwapTotal:    33448 kB
SwapFree:     33448 kB
```

Viel Spass beim Swappen.