Einrichtung eines SAN-Clients unter Ubuntu 14.04

Einleitung

Innerhalb des Speichernetzes (Storage Area Network, SAN) wird Speicherplatz in Form von virtuellen Platten durch mehrere Server zur Verfügung gestellt. Der Zugriff erfolgt über das Fibre-Channel-SAN. Dieses besteht aus zwei getrennten Netzwerken ("Fabric"), so dass jeder Client mit zwei Fibre-Channel-Ports (je ein Port pro Fabric) mit dem SAN verbunden ist. Der Disk-Virtualisierungsserver besitzt pro Fabric jeweils zwei Anschlüsse, daher erscheint jede SAN-Platte auf den Klienten vierfach. Diese vier Fibre-Channel-Geräte enthalten jedoch jeweils identische Daten (in Wirklichkeit handelt es sich um vier verschiedene Pfade zur selben Platte). Daher sollten diese Geräte niemals direkt verwendet werden. Stattdessen werden die vier Pfade per Software zu einem gemeinsamen Device verbunden ("multipathing"). Diese Lösung hat den Vorteil, dass sie fehlertolerant ist und sowohl den Ausfall eines Adapters (in Client oder Server), als auch den Ausfall einer Fabric verkraftet. Zusätzlich werden die Server, die die virtuellen Platten an die Clients verteilen, als Paare betrieben. Bei Ausfall eines Servers übernimmt der andere Server des Paares für die Clients transparent dessen Aufgaben ("Failover"). Hierbei kommt es zu einer kurzen Unterbrechung und es ist notwendig die Clients so zu konfigurieren, dass während der Unterbrechung keine Fehler an das Betriebssystem gemeldet werden. Die Konfiguration unter Ubuntu 14.04 wird im Folgenden beschrieben.

Konfiguration

Die Konfiguration muss mit Root-Rechten ausgeführt werden.

Zunächst sollte mit lsscsi überprüft werden, ob die virtuelle Platte sichtbar ist. Unter Ubuntu ist lsscsi standardmäßig nicht installiert, daher wird es im ersten Schritt installiert:

```
> apt-get install lsscsi
.
.
.
> lsscsi |grep FALCON
[1:0:0:0]    disk    FALCON   IPSTOR DISK      v1.0  /dev/sdd
[1:0:1:0]    disk    FALCON   IPSTOR DISK      v1.0  /dev/sde
[2:0:0:0]    disk    FALCON   IPSTOR DISK      v1.0  /dev/sdf
[2:0:1:0]    disk    FALCON   IPSTOR DISK      v1.0  /dev/sdg
```

Im oben gezeigten Fall ist eine Platte vorhanden, die über vier Pfade zu erreichen ist.

Sollte die virtuelle Platte nicht sichtbar sein, ist es notwendig, nach neuen Geräten zu suchen. Hierzu wird die ID der FiberChannel-Adapter benötigt. Diese kann mit `lsscsi -H |grep qla` ausgegeben werden, sofern FiberChannel-Adapter von QLogic verwendet werden.

```
> lsscsi -H |grep qla
[1]    qla2xxx
[2]    qla2xxx
```

Bei neueren QLogic-Adaptern (8 GBit-Karten von Typ 25xx) wird als Gerätetyp nur `(null)` ausgegeben.

Im obigen Beispiel sind die IDs 1 und 2. Diese Adapter werden nun zu einem Rescan aufgefordert.

```
> echo "- - -" > /sys/class/scsi_host/host1/scan
> echo "- - -" > /sys/class/scsi_host/host2/scan
```

Die Bezeichnungen `host1` und `host2` müssen an die jeweiligen IDs angepasst werden.

Wenn die Devices sichtbar sind wird das "Multipathing" konfiguriert. Das notwendige Paket ist nicht standardmäßig installiert und sollte nachinstalliert werden:

```
apt-get install multipath-tools
```

Anschließend legen Sie die Datei multipath.conf in /etc/multipath.conf ab.

```
defaults {
        polling_interval 10
        path_grouping_policy    multibus
        failback        immediate
        no_path_retry  queue
        user_friendly_names yes
        dev_loss_tmo 300
        fast_io_fail_tmo 5
}
blacklist {
        device {
                vendor .*
                product .*
        }
}
blacklist_exceptions {
        device {
                vendor "FALCON.*"
                product ".*"
        }
}
```

Jetzt kann das Multipath-System neu gestartet und das das Multipathing eingerichtet werden:

```
> /etc/init.d/multipath-tools restart
> multipathd -k"reconfigure"
> multipath
create: mpath0 (36000d77900004e44523c55589f2511d4)  FALCON,IPSTOR DISK
[size=40G][features=0][hwhandler=0]
\_ round-robin 0 [prio=4][undef]
 \_ 1:0:0:0 sdd 8:48  [undef][ready]
 \_ 1:0:1:0 sde 8:64  [undef][ready]
 \_ 2:0:0:0 sdf 8:80  [undef][ready]
 \_ 2:0:1:0 sdg 8:96  [undef][ready]
```

In der Zeile `multipathd -k"reconfigure"` darf kein Leerzeichen zwischen `-k` und `"reconfigure"` stehen.

Der Status des Multipath-Systems lässt sich mit `multipath -ll` ausgeben.

```
> multipath -ll
mpath0 (36000d77900004e44523c55589f2511d4) dm-4 FALCON,IPSTOR DISK
[size=40G][features=1 queue_if_no_path][hwhandler=0]
\_ round-robin 0 [prio=4][active]
 \_ 1:0:0:0 sdd 8:48  [active][ready]
 \_ 1:0:1:0 sde 8:64  [active][ready]
 \_ 2:0:0:0 sdf 8:80  [active][ready]
 \_ 2:0:1:0 sdg 8:96  [active][ready]
```

Im oben gezeigten Fall ist ein Multipath-Device mit der ID 36000d77900004e44523c55589f2511d4 vorhanden, wobei die Zugriffe über die vier Pfade (sdd, sde, sdf, sdg) gleichmäßig verteilt werden ("round-robin"). Alle Pfade sind in Ordnung ("active", "ready") und die Eigenschaft "queue_if_no_path" ist gesetzt, so dass das System alle Zugriffe in eine Warteschlange stellt, falls alle Pfade ausfallen (z. B. weil der Virtualisierungsserver ausgefallen ist und es einen Moment dauert bis der Ersatzserver die Dienste übernimmt). Die virtuelle Platte kann im Beispiel als /dev/mapper/mpath0 verwendet werden (der Name unterhalb von /dev/mapper wird in der ersten Zeile von `multipath -ll` nach der ID ausgegeben).

Verwendung der virtuellen Platte

Benutzung von Partitionen

Das Multipath-Device kann mit `fdisk` partioniert werden. Hierzu wird das Gerät verwendet, das von `multipath -ll` in der ersten Zeile ausgegeben (Das Multipath-Gerät ist sowohl über `/dev/mapper/mpath0` als auch über `/dev/dm-4` verfügbar, die Benutzung von `/dev/mapper/mpath0` ist jedoch vorzuziehen, da dieser Devicename mit der ID der virtuellen Platte verknüpft ist. Das dm-Device `/dev/dm-4` wird vom Device-Mapper dynamisch erzeugt und es nicht sicher, dass dieses Gerät immer `dm-4` zugeordnet wird).

Ausgehend vom obigen Beispiel mit /dev/mpath0 würde dies mit

```
> fdisk /dev/mapper/mpath0
```

geschehen. Die Partitionen sind direkt verwendbar, es werden entsprechende Geräte unter `/dev/mapper/mpath0-partx` (wobei x die Partitionsnummer bezeichnet) angelegt. Zu beachten ist, dass auf diese Weise maximal 2 TB verwendet werden können. Für größere Platten existieren drei Möglichkeiten: GPT-Partionen (diese können mit gparted erzeugt werden), partitionslose Verwendung (Dateisystem wird direkt auf dem Multipath-Device - im Beispiel `/dev/mapper/mpath0` - erzeugt) oder logical Volume Management (siehe nächster Abschnitt).

Benutzung des LVM

Alternativ zur klassischen Partitionierung der virtuellen Platte kann das "Logical-Volume-Management" LVM eingesetzt werden. Dies hat gegenüber der Partitionierung den Vorteil einer erhöhten Flexibilität, da Volumes im laufenden Betrieb vergrößert werden können. Ein Nachteil ist die größere Komplexität, da zusätzlich zum Multipathing auch das Volume-Management konfiguriert werden muss.

Im LVM gibt es drei Ebenen. Die oberste Ebene besteht aus logischen Volumes, die vergleichbar mit Partitionen im klassischen Schema sind. Logische Volumes befinden sich in Volume Groups und diese wiederum bestehen aus einem oder mehreren physischen Volumes. Physische Volumes entsprechen normalerweise Festplatten, in diesem Fall also der per Multipath verwalteten virtuellen Platte.

Der erste Schritt bei der Verwendung des LVM ist es, ein "Physical Volume" auf der virtuellen Platte anzulegen. Wie im Beispiel oben ist die virtuelle Platte `/dev/mapper/mpath0`

```
> pvcreate /dev/mapper/mpath0
  Physical volume "/dev/mapper/mpath0" successfully created
> pvdisplay
  --- NEW Physical volume ---
  PV Name               /dev/mapper/mpath0
  VG Name
  PV Size               40.00 GB
  Allocatable           NO
  PE Size (KByte)       0
  Total PE              0
  Free PE               0
  Allocated PE          0
  PV UUID               02i9AJ-m3ka-1n5Z-VDpg-3VH7-LnxN-pf9VIE
```

`pvcreate` erzeugt ein physisches Volume und das anschließende `pvdisplay` zeigt die physischen Volumes an. Nach der Erzeugung gehört das physische Volume noch keiner Volume-Group an, daher ist der Eintrag `VG Name` leer und "Allocatable" ist "NO".

Der nächste Schritt ist das Anlegen einer Volume-Group, die das physische Volume enthält. Der Name kann frei gewählt werden, im Beispiel heißt die Gruppe "vg_data".

```
> vgcreate vg_data /dev/mapper/mpath0
  Volume group "vg_data" successfully created
```

Die logischen Volumes können schließlich in der Volume-Group erzeugt werden. Im folgenden werden zwei logische Volumes mit den Bezeichnungen "lv_data1" und "lv_data2", jeweils mit einer Größe von 10 GB erzeugt. Die Option `-n` gibt den Namen des logischen Volumes an und `-L` die Größe (Angaben können in Megabyte `M`, Gigabyte `G` oder Terabyte `T` erfolgen) oder `–l100%FREE` (wenn alles benutzt werden soll). Ferner ist die Angabe der Volume-Group (in diesem Beispiel `vg_data`) notwendig.

```
> lvcreate -n lv_data1 -L 10G vg_data
  Logical volume "lv_data1" created
> lvcreate -n lv_data2 -L 10G vg_data
  Logical volume "lv_data2" created
> lvdisplay
  --- Logical volume ---
  LV Name                /dev/vg_data/lv_data1
  VG Name                vg_data
  LV UUID                maPLte-dW7f-CKEO-IAe0-GnAf-gv0M-5e9Pk1
  LV Write Access        read/write
  LV Status              available
  # open                 0
  LV Size                10.00 GB
  Current LE             2560
  Segments               1
  Allocation             inherit
  Read ahead sectors     0
  Block device           254:5

  --- Logical volume ---
  LV Name                /dev/vg_data/lv_data2
  VG Name                vg_data
  LV UUID                gYgNrX-HBGO-4SSz-VsGO-gr8Z-9yRs-OM9omD
  LV Write Access        read/write
  LV Status              available
  # open                 0
  LV Size                10.00 GB
  Current LE             2560
  Segments               1
  Allocation             inherit
  Read ahead sectors     0
  Block device           254:6
> vgdisplay
  --- Volume group ---
  VG Name               vg_data
  System ID
  Format                lvm2
  Metadata Areas        1
  Metadata Sequence No  3
  VG Access             read/write
  VG Status             resizable
  MAX LV                0
  Cur LV                2
  Open LV               0
  Max PV                0
  Cur PV                1
  Act PV                1
  VG Size               40.00 GB
  PE Size               4.00 MB
  Total PE              10239
  Alloc PE / Size       5120 / 20.00 GB
  Free  PE / Size       5119 / 20.00 GB
  VG UUID               S8Wjpb-xmE5-Ieud-aoB0-qeFA-QQc1-yCFjnd
```

Im Beispiel ist die Hälfte das Speicherplatzes zugeordnet (in `vgdisplay` unter `Alloc PE / Size`) und weitere 20 GB sind frei. Diese lassen sich später den logischen Volumes zuordnen, um den Speicherplatz zu erweitern. Die logischen Volumes können nun unter dem in `LV Name` angegebenen Devicepfaden (in der `lvdisplay` Ausgabe) wie gewöhnliche Partitionen verwendet werden (Filesystem anlegen, mounten, etc.).

Vergrößerung von logischen Volumes

Logische Volumes lassen sich vergrößern, wenn noch freier Speicherplatz in der Volumegroup vorhanden ist. Dies lässt sich mit `vgdisplay` überprüfen.

```
> vgdisplay
  --- Volume group ---
  VG Name               vg_data
  System ID
  Format                lvm2
  Metadata Areas        1
  Metadata Sequence No  3
  VG Access             read/write
  VG Status             resizable
  MAX LV                0
  Cur LV                2
  Open LV               1
  Max PV                0
  Cur PV                1
  Act PV                1
  VG Size               40.00 GB
  PE Size               4.00 MB
  Total PE              10239
  Alloc PE / Size       5120 / 20.00 GB
  Free  PE / Size       5119 / 20.00 GB
  VG UUID               S8Wjpb-xmE5-Ieud-aoB0-qeFA-QQc1-yCFjnd
```

Im Beispiel sind 20 GB (5119 "Physical Extents" (PE) zu je 4 MB) in der Volume-Group frei. Im Beispiel wird jetzt dem logischen Volume lv_data1 mit `lvextend` zusätzlicher Speicher zugeordnet.

```
> lvextend -L +10G /dev/vg_data/lv_data1
  Extending logical volume lv_data1 to 20.00 GB
  Logical volume lv_data1 successfully resized
```

Mit der Option `-L +10 GB` wird die Größe des Volume um 10 GB erweitert (ohne "+" würde die neue Größe des Volume angegeben) oder mit `-l+100%FREE` wird der gesamte verfügbare Speicher hinzugefügt.. Das Volume ist jetzt vergrößert, der zusätzliche Speicherplatz aber noch nicht verfügbar, da die ursprüngliche Größe noch im Filesystem gespeichert ist. Daher muss noch die Größe des Filesystems angepasst werden. Wenn ext3 (bzw. ext2) als Filesystem verwendet wird, erfolgt dies mit `resize2fs` oder bei Nutzung von xfs das Kommando `xfs_growfs`. Die Vergrößerung des Volumes und des Filesystems können Online, also im laufenden Betrieb bei gemountetem Filesystem erfolgen.

Hinweise zum Failover

Die Virtualisierungsserver werden im Rahmen einer Hochverfügbarkeitslösung als Failover-Paare betrieben. Die zwei Partner des Paares überwachen sich hierbei gegenseitig. Sollte ein Partner feststellen, dass der Andere ausgefallen ist, übernimmt er die Funktionen des ausgefallenen Servers. Dieser Vorgang dauert bis zu zwei Minuten. Aktuelle Linux-Systeme kennen zwei Parameter für Fibre Channel Geräte, die in diesem Zusammenhang eine Rolle spielen: `fast_io_fail_tmo` und `dev_loss_tmo`. Der Parameter `dev_loss_tmo` beschreibt einen Zeitraum ("Time-Out"), den das System abwartet, bevor es ein nicht reagierendes Fibre-Channel-Geräte als fehlerhaft erkennt und weitere IO-Operationen zu diesem Gerät blockiert. Er ist standardmäßig für Q-Logic-Adapter auf 45 Sekunden gesetzt. Der zweite Parameter `fast_io_fail_tmo` ist standardmäßig deaktiviert ("off"); mit dieser Einstellung wartet das System bis der `dev_loss_tmo` Zeitraum abgelaufen ist, bevor IO-Fehler weitergegeben werden. Dies ist in einer Multipath-Umgebung nicht sinnvoll, da dann das Multipath-System keine Mitteilung über einen fehlerhaften Pfad erhält, bis der `dev_loss_tmo` Zeitraum abgelaufen ist. Um dies zu Ändern kann der Parameter `fast_io_fail_tmo` auf eine Zeit in Sekunden eingestellt werden, nach der IO-Fehler weitergegeben werden. Bewährt hat sich eine Einstellung von 5 für `fast_io_fail_tmo` und 300 für `dev_loss_tmo`. Letzteres ist ausreichend, damit das System die Geräte auch beim Failover nicht als fehlerhaft abmeldet.

Ab Ubuntu 12.04 werden diese Parameter über den Multipath-Daemon verwaltet. Hierzu werden die Werte in `/etc/multipath.conf` gesetzt. In der obigen `multipath.conf` sind die Einträge bereits vorhanden.

Bei der Verwendung von LVM ist ferner ein Problem in Ubuntu zu beachten, da die LVM-Geräte im Boot-Prozess aktiviert werden, bevor das Multipathing-System gestartet wird. Hierdurch wird das LVM-Logical-Device an einen Pfad gebunden anstatt an das Multipath-Gerät. Dies lässt sich mit `dmsetup deps` anzeigen:

```
> dmsetup deps
  mpath0: 4 dependencies  : (8, 96) (8, 80) (8, 64) (8, 32)
  vg_test-lv_test: 1 dependencies : (8, 32)
```

`vg_test-lv_test` ist an einen der Pfade gebunden, der auch darüber bei mpath0 aufgeführt ist. Korrekt wäre eine Bindung an mpath0 (`(252, x)` wobei x das dm-Gerät für den mapth bezeichnet). Das Problem tritt bei einem neuen logischen Volume zunächst nicht auf, da das Gerät zu einem Zeitpunkt aktiviert wird,

Eine Lösung des Problems ist es, das Multipathing bereits in der initrd zu starten (die LVM-Geräte werden in der initrd gestartet, vermutlich, um einen Start von einem LVM-Root-Filesystem zu erlauben). Hierzu gibt es das Paket multipath-tools-boot. Dieses behebt für sich das Problem jedoch nicht. Der Start der von Multipathing und LVM erfolgt über udev-Regeln, wobei die LVM-Regel die höhere Priorität hat. Daher ist es notwendig, die Reihenfolge der udev-Regeln zu verändern:

```
> apt-get install multipath-tools-boot
> mv /lib/udev/rules.d/95-multipath.rules /lib/udev/rules.d/80-
multipath.rules
```

In der Datei `/usr/share/initramfs-tools/hooks/multipath` muss die Zeile

```
for rules in 95-multipath.rules; do
```

in

```
for rules in 80-multipath.rules; do
```

geändert werden.

```
> update-initramfs -u
```

Nach Änderungen an `/etc/multipath.conf` ist ein Aufruf von `update-initramfs -u` notwendig, um die neue multipath.conf in der initrd unterzubringen.

Wenn die Änderungen angeschlossen sind, sollte überprüft werden, ob die Zuordnung der Devicemapper-Geräte nach einem Reboot korrekt sind:

```
> dmsetup deps
  mpath0: 4 dependencies  : (8, 64) (8, 32) (8, 48) (8, 80)
  vg_test-lv_test: 1 dependencies : (252, 0)
```

Im Beispiel oben ist das logische Volume von (252, 0) abhängig (252 ist die Major-ID des Devicemapper, (252, 0) entspricht somit dm-0 und dies wiederum mpath0).

Hinzufügen von virtuellen Platten

Im folgenden Abschnitt wird beschrieben, wie eine zusätzliche virtuelle Platte hinzugefügt wird.

Nachdem die neue virtuelle Platte dem System von den SAN-Operatoren zugeordnet wurde, muss sie vom System erkannt werden. Hierzu wird die ID der FiberChannel-Adapter benötigt. Diese kann mit `lsscsi -H |grep qla` ausgegeben werden, sofern FiberChannel-Adapter von QLogic verwendet werden.

```
> lsscsi -H |grep qla
[1]    qla2xxx
[2]    qla2xxx
```

Im obigen Beispiel sind die IDs 1 und 2. Diese Adapter werden nun zu einem Rescan aufgefordert.

```
> echo "- - -" > /sys/class/scsi_host/host1/scan
> echo "- - -" > /sys/class/scsi_host/host2/scan
```

Die Bezeichnungen `host1` und `host2` müssen an die jeweiligen IDs angepasst werden.

Jetzt sind im `multipath -ll` zwei virtuelle Platten zu sehen (unter Ubuntu wird die neue Platte vom Multipath-System konfiguriert, sobald sie erkannt wurde):

```
> multipath -ll
mpath1 (36000d77a000045b8523c555ef1f8c817) dm-7 FALCON  ,IPSTOR DISK
[size=40G][features=1 queue_if_no_path][hwhandler=0]
\_ round-robin 0 [prio=4][active]
 \_ 2:0:1:1 sdk 8:160 [active][ready]
 \_ 1:0:0:1 sdh 8:112 [active][ready]
 \_ 1:0:1:1 sdi 8:128 [active][ready]
 \_ 2:0:0:1 sdj 8:144 [active][ready]
mpath0 (36000d77900004e44523c55589f2511d4) dm-4 FALCON  ,IPSTOR DISK
[size=40G][features=1 queue_if_no_path][hwhandler=0]
\_ round-robin 0 [prio=4][active]
 \_ 1:0:0:0 sdc 8:32  [active][ready]
 \_ 1:0:1:0 sdd 8:48  [active][ready]
 \_ 2:0:0:0 sde 8:64  [active][ready]
 \_ 2:0:1:0 sdf 8:80  [active][ready]
```

Die neue Platte ist jetzt unter `/dev/mapper/mpath1` verfügbar.

Das neue Device kann jetzt entweder wie oben beschrieben partitioniert werden, oder es kann ein physisches Volume angelegt werden. Bei Verwendung des LVM-Systems kann das neue Device nach Erzeugung des physischen Volumes mit `vgextend` einer bestehenden Volumegroup zugeordnet werden, um den Speicherplatz der Volumegroup zu erweitern.

Zum Abschluss sollte der Multipath-Daemon zu einer Rekonfiguration aufgefordert werden, damit alle Einstellungen konsistent sind. Dies erfolgt durch (kein Leerzeichen zwischen `-k` und `"reconfigure"`!):

```
> multipathd -k"reconfigure"
ok
```

-- JensDoebler - 13 Aug 2012