JUST DO IT
HÖHERE ABTEILUNG FÜR INFORMATIONSTECHNOLOGIE

tgm
Die Schule der Technik

Laborprotokoll

iSCSI Initiator & OCFS2 Cluster Dateisystem

Systemtechnik Labor
5BHIT 2016/17, Gruppe B

Kocsis, Limbeck, May, Ucel, Weber

Version 0.1
Begonnen am 14. Oktober 2016
Beendet am 18. Oktober 2016

Note:
Betreuer: Markus Schabel

Inhaltsverzeichnis

1 Einführung

Diese Übung zeigt die Anwendung von iSCSI und OCFS2.

1.1 Ziele

Das Ziel dieser Übung ist einen iSCSI-Initiator zu konfigurieren. Ebenfalls soll ein OCFS2 Cluster konfiguriert werden.

1.2 Voraussetzungen

- Grundlagen iSCSI
- Grundlagen der Netzwerktechnik
- Grundlagen Linux

1.3 Aufgabenstellung

1.3.1 Basics

Installieren und Konfigurieren Sie eine NAS Appliance, welche Block Devices (Festplatten) über iSCSI im Netzwerk freigeben kann (z.B. openfiler, FreeNAS), falls diese nicht bereits für die Übung zur Verfügung gestellt wurde (je nach Gruppe/Zeitplan).

Installieren Sie auf einem Linux-Client einen iSCSI Initiator (bei Debian/Ubuntu im Paket open-iscsi) und konfigurieren Sie diesen entsprechend: zumindest die Authentifizierung muss in der Datei /etc/iscsi/iscsi.conf konfiguriert werden (danach muss open-iscsi neu gestartet werden):

```
node.session.auth.authmethod = CHAP
```

```
node.session.auth.username = test
```

```
node.session.auth.password = test
```

Im Anschluss können Sie den iSCSI Client (Initiator) mit dem iSCSI Server (Target) verbinden. Im ersten Schritt soll der Server (die IP-Adresse muss logischerweise angepasst werden) nach den von ihm angebotenen Targets (spezifischen Block-Devices) befragt werden:

```
iscsiadm -m discovery -t sendtargets -p 10.5.104.33
```

Dieser Befehl sollte eine Liste an Targets ausgeben, wie z.B.:

```
iqn.2006-01.com.openfiler:tsn.target01
```

```
iqn.2006-01.com.openfiler:tsn.target02
```

Danach kann man ein spezifisches Target im Client als "lokale"Festplatte registrieren (bzw. mit `-logout` wieder entfernen):

```
iscsiadm -m node -T iqn.2006-01.com.openfiler:tsn.target01 -p 15.104.33 --login
```

Bei Erfolg sieht man in der Kernel-Console (`dmesg`) das neue Block-Device wie etwa `sdb`.

1.3.2 Selbstständige zu lösende Aufgaben

- Das iSCSI Target soll automatisch beim booten des Clients/Initiators verbunden werden.
- Installieren Sie `ocfs2` auf mindestens 3 Computern (die auf das Block-Devices per iSCSI verbunden sind).
- Konfigurieren Sie einen OCFS2 Cluster für diese 3 Computer (Nodes), z.B. über die Datei `/etc/ocfs2/cluster.conf`.
- Formatieren Sie das gemeinsame iSCSI Block Device im OCFS2 Dateisystem (`mkfs.ocfs2`) - dies muss nur auf einem Computer durchgeführt werden, da es sich ja um ein Cluster-Dateisystem handelt.
- Überprüfen Sie, wie sich das Dateisystem bei gemeinsamen Zugriff auf eine Datei verhält.
- Dokumentieren Sie Ihre Durchführung und Ihre Ergebnisse.

Die Arbeit ist in Gruppen zu 3-4 Teilnehmern möglich. Jeder Teilnehmer muss den iSCSI-Initiator und OCFS2 konfiguriert haben.

Abgabe des Protokolls (1x/Gruppe) als pdf.

2 Ergebnisse

2.1 Theorie

2.1.1 iSCSI

Überblick:
Internet Small Computer Systems Interface ist eine Erweiterung des *SCSI* Standards. Es erlaubt Clients Daten über TCP in ein *Storage Area Network* (NAS) zu speichern, als wäre es direkt mit der Festplatte verbunden. Damit ist es möglich, mehrere Festplatten von unterschiedlichen Standorten zu einer virtuellen Festplatte zusammenzufassen, welche für mehreren Clients erreichbar ist. Es lassen sich große Distanzen überwinden, da sich iSCSI zwischen Router, host-to-switch, storage-array-to-storage-array implementieren lässt.

Funktionsweise:
iSCSI überträgt block-level Data zwischen iSCSI initiator und den Festplatten, auch iSCSI targets genannt. Der iSCSI initiator kapselt die SCSI Kommandos und Daten in Pakete für den TCP/IP Layer. Diese Pakete werden dann über das Netzwerk mittels point-to-point Verbindung übertragen. Das iSCSI Protokoll trennt dann die SCSI Kommandos von den Daten, so dass es für den Client aussieht, als wäre er direkt lokal mit der Festplatte verbunden.

Vorteile:
Der größte Vorteil von iSCSI gegenüber anderen Möglichkeiten, wie FCIP(Fiber Channel over IP), ist, dass keine teuren und komplexen Switches bzw. Router wie für FCIP benötigt werden. Das macht es billiger und einfacher zu verwalten. Weiteres ist iSCSI Variante auch weit flexibler als andere Möglichkeiten.

2.1.2 OCFS2

Oracle Cluster File System ist ein verteiltes Dateisystem entwickelt von Oracle Corporation veröffentlicht unter der GNU General Public License. Das Dateisystem ist also eine open source Variante, welche leistungsstarke Performance und hohe Verfügbarkeit bietet. Es verwendet die gleiche Semantik wie lokale Dateisysteme und lässt sich mit jeder beliebigen Anwendung verwenden.

2.2 iSCSI Initiator

2.2.1 Basics

Da openfiler bereits zur Verfügung gestellt wurde, musste keine NAS Appliance installiert und konfiguriert werden.

Die restliche Vorgehensweise entspricht genau der Aufgabenstellung. Zuerst wurde das Ubuntu Package `open-iscsi` installiert. Nachdem in der Datei `/etc/iscsi/iscsi.conf` die Authentifizierungsmethode auf CHAP (Challenge Handshake Authentication Protocol) gesetzt und die Anmeldedaten konfiguriert wurden, konnte auch schon die Liste der verfügbaren Targets ausgegeben werden.

```
daniel@daniubuntu:~$ sudo iscsiadm -m discovery -t sendtargets -p 10.0.106.81
10.0.106.81:3260,1 iqn.2006-01.com.openfiler:tsn.96b8938ac933
10.0.106.81:3260,1 iqn.2016-09.syt:tsn.data1
10.0.106.81:3260,1 iqn.2016-09.syt.tgm.data0
10.0.106.81:3260,1 iqn.2006-01.com.openfiler:tsn.465272b2cbcc
10.0.106.81:3260,1 iqn.2016-09.syt:tsn.data0
```

Listing 1: Verfügbare Targets

Die Option `-m` legt den Operation-Mode als `discovery` fest. Im `discovery` Modus wird mit `-t` der Discovery-Typ auf `sendtargets` gesetzt. Mit `-p` wird die Zieladresse festgelegt. Danach wurde das `data0` Target als "lokale" Festplatte registriert:

```
daniel@daniubuntu:~$ sudo iscsiadm -m node -T iqn.2016-09.syt:tsn.data0 -p 10.0.106.81 --login
Logging in to [iface: default, target: iqn.2016-09.syt:tsn.data0, portal: 10.0.106.81,3260] (multiple)
Login to [iface: default, target: iqn.2016-09.syt:tsn.data0, portal: 10.0.106.81,3260] successful.
```

Listing 2: Target "lokal" registrieren

Hier wurde der Modus auf `node` geändert und mit `-T` wird der Targetname angegeben. Mit `--login` wird spezifiziert, dass man sich im `node` Modus beim angegebenen Target einloggen möchte.

In der Kernel Console sieht man nun das neue Block Device:

```
[ 1008.354886] sd 33:0:0:0: Attached scsi generic sg2 type 0
[ 1008.369151] sd 33:0:0:0: [sdb] 1048576 512-byte logical blocks: (537 MB/512 MiB)
[ 1008.418559] sd 33:0:0:0: [sdb] Write Protect is off
[ 1008.418605] sd 33:0:0:0: [sdb] Mode Sense: 77 00 00 08
[ 1008.430186] sd 33:0:0:0: [sdb] Write cache: disabled, read cache: disabled, doesn't support DPO or FUA
```

Listing 3: neues Block Device

Auch mit `fdisk -l` kann man sehen, dass das neue Block Device eingebunden wurde:

```
Disk /dev/sdb: 512 MiB, 536870912 bytes, 1048576 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
```

Listing 4: `fdisk -l`

2.2.2 iSCSI Verbindung beim Booten

Diese Aufgabe wurde erst erledigt, nachdem OCFS2 fertig konfiguriert wurde.

Zuerst müssen die Dienste `o2cb` und `OCFS2` in den Autostart hinzugefügt werden. Dies wurde mit

`sudo update-rc.d <service> defaults`

erreicht. [1] Danach wurde mittels

`sudo dpkg-reconfigure ocfs2-tools`

eingestellt, dass ein OCFS2-Cluster zur Startzeit gestartet wird.

Letztendlich muss nur noch der Eintrag

`/dev/sdb /tmp_mount ocfs2 _netdev 0 0`

in die Datei `/etc/fstab`, die für das automatisierte Einhängen von Partitionen verantwortlich ist, geschrieben werden. [2]

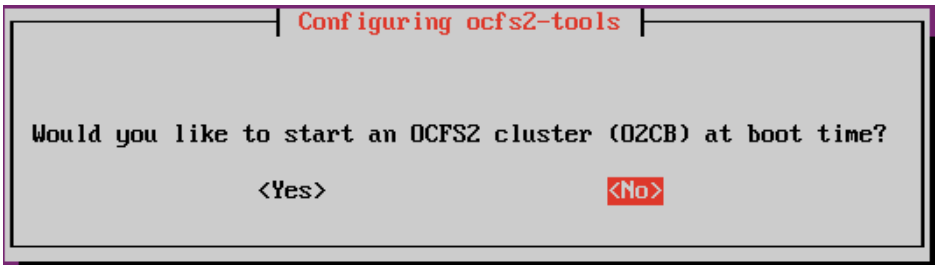

Abbildung 1: Autostart der OCFS2-Tools

Die erste Spalte beschreibt das einzuhängende Dateisystem. Die zweite Spalte definiert den Mountpoint, dass heißt wo das Dateisystem eingehängt wird. Die dritte Spalte spezifiziert die Art des Dateisystems. Danach folgt die Option `_netdev`, welche spezifiziert, dass es sich um einen Netzwerkspeicher handelt und deswegen so lange mit dem Einhängen gewartet wird, bis eine Netzwerkverbindung zur Verfügung steht. Die erste Zahl steht für Sicherungseinstellungen mittels `dump`. In diesem Fall bedeutet 0 keine Sicherung. Die zweite Zahl gibt an, ob und in welcher Reihenfolge die Partition beim Systemstart in die regelmäßigen Dateisystemprüfungen einbezogen wird. 0 bedeutet hier wieder keine Prüfung. [3]

2.3 OCFS2

2.3.1 Installation

Zuerst wurden verschiedenste `OCFS2` Tools mittels

```
sudo apt-get install ocfs2*
```

installiert.

2.3.2 Konfiguration

Nun wurde mit der Datei `/etc/ocfs2/cluster.conf` ein Cluster für 4 Nodes konfiguriert. Diese Konfiguration muss bei jeder Node (Client) vorhanden sein.

```
node:
name = ubuntu
cluster = ocfs2
number = 1
ip_address = 10.0.104.10
ip_port = 7777

node:
name = ubuntu2
cluster = ocfs2
number = 2
ip_address = 10.0.106.93
ip_port = 7777

node:
name = ubuntuWeber
cluster = ocfs2
number = 3
ip_address = 10.0.106.90
ip_port = 7777

node:
name = daniubuntu
cluster = ocfs2
number = 4
ip_address = 10.0.106.100
ip_port = 7777

cluster:
name = ocfs2
heartbeat_mode = local
node_count = 4
```

Listing 5: `/etc/ocfs2/cluster.conf`

Für jede Node werden Hostname, zugehöriger Cluster, Nummerierung und notwendigen Adressdaten spezifiziert.

Zusätzlich ist der `heartbeat_mode` beim Cluster interessant. Dieser Modus definiert, wie die Nodes den gemeinsamen Zugriff auf geteilte Ressourcen koordinieren und die jeweiligen Status überwachen. Im `local` Modus starte jede Node einen Heartbeat Thread, wenn ein OCFS2 Device eingehängt wird. Werden mehrere Devices auf einer Node eingehängt bedeutet dies einen großen CPU-Overhead. Weiters kann auch ein Netzwerk, sowie I/O Overhead enstehen. Dies passiert wenn viele Nodes oder sogar mehrere Cluster vorhanden sind. Im `global` Modus müssen sich nicht die Nodes um die Heartbeat-Funktion kümmern. Der Hearbeat startet automatisch mit dem Cluster. [4]

2.3.3 Starten des Clusters

Danach werden die Module des Diensts `o2cb` geladen:

```
sudo service o2cb load
```

Um diese dann auch Online zu schalten wird

```
sudo service o2cb online
```

ausgeführt. [2]

2.3.4 Formatieren des Volumes

Bevor die anderen Clients auf das Volume zugreifen können muss ein Client dieses formatieren. mit dem `mkfs.ocfs2` Befehl kann ein Volume formatiert werden. mit `-N` wird die Anzahl der Nodes definiert, die sich mit dem Filesystem verbinden können. Diese kann im Nachhinein erhöht, jedoch nicht verringert werden. mit -L gibt man dem Filesystem einen Namen.

```
mkfs.ocfs2 /dev/sdb -N 4 -L "test123"
```

Falls ein schon vorhandenes Filesystem überschrieben werden muss kann dies mit dem Zusatz `-F` erwirkt werden.

2.3.5 Einhängen des Volumes

Nachdem das Dateisystem initialisiert wurde, kann das OCFS2 Volumen eingehängt werden. Dies geschieht mit dem `mount` Befehl, der mit `-t` die Art des Dateisystems beschreibt. Danach wird spezifiziert was wo eingehängt wird. [2]

```
sudo mount -t ocfs2 /dev/sdb /tmp_mount
```

2.4 Testing

Zuerst wurde mit Client 1 eine Testdatei erstellt. Danach hat Client 2 die Datei geöffnet und bearbeitet.

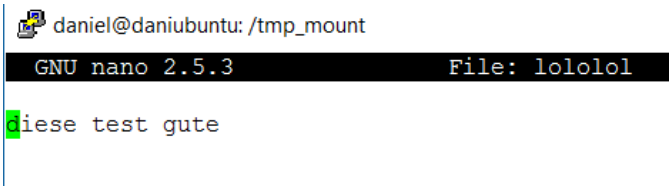

Abbildung 2: Editieren der gemeinsamen Datei

Versucht nun Client 1 ebenfalls die Datei zu editieren erhält man folgende Fehlermeldung:

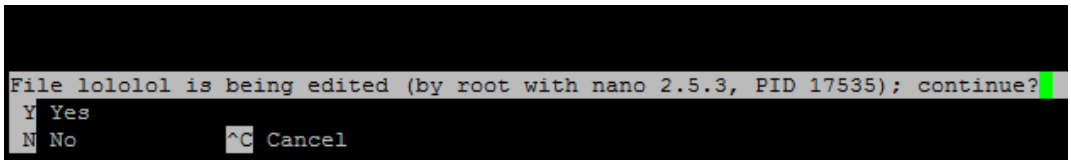

Abbildung 3: Gleichzeitiges Editieren

Im Endeffekt ist immer die letzte Aktion gültig und überschreibt die vorherigen Änderungen. Dies ist jedoch nur möglich, da nano ein Journal führt und selbst die Datei lockt. Würde eine

andere Applikation, welche diese Funktionalitäten nicht implementiert, eine Datei gleichzeitig von 2 verschiedenen Clients bearbeiten, würde diese korrupt werden.

Literatur

[1] chantra. How-to: Managing services with update-rc.d. `https://www.debuntu.org/how-to-managing-services-with-update-rc-d/`, Juli 2007. [Online; Zugriff am: 18. Oktober 2016].

[2] Oracle. Ocfs2 1.2 - frequently asked questions. `https://oss.oracle.com/projects/ocfs2/dist/documentation/v1.2/ocfs2_faq.html`, 2009. [Online; Zugriff am: 18. Oktober 2016].

[3] noisefloor. Wiki/fstab. `https://wiki.ubuntuusers.de/fstab/`, Januar 2016. [Online; Zugriff am: 18. Oktober 2016].

[4] heartbeat. Preparing a cluster for ocfs2. `https://docs.oracle.com/cd/E37670_01/E37355/html/ol_prepare_ocfs2.html`, 2012. [Online; Zugriff am: 18. Oktober 2016].

Tabellenverzeichnis

Listings

Abbildungsverzeichnis