

Udev als Alternative zu ASMLib im Linux - Umfeld

Autor: Claus Cullmann , eXirius IT Dienstleistungen GmbH

Für Automatic Storage Management (ASM) auf Linux-Plattformen empfiehlt Oracle die Benutzung der ASMLib, um Partitionen oder LUNs als ASM-Disk der ASM-Instanz zur Verfügung zu stellen. In modernen Linux - Systemen gibt es die Alternative udev . Dieser Artikel beschreibt die Implementierung von udev und zeigt die Vor- und Nachteile gegenüber ASMLib.

Das Linux - Betriebssystem nutzt das `/dev` Verzeichnis, um verschiedene Geräte (Festplatten, Storage-Systeme) einzubinden. Jede Verbindung zielt auf ein bestimmtes Gerät. So werden z.B. SCSI-Festplatten mit a, b, c, durchnummeriert. Die Reihenfolge ist hier jedoch möglicherweise nicht immer gleich. Nach einem Neustart ist es möglich, dass die Festplatte, die vorher noch mit `/dev/sdb` ansprechbar war, nun unter `/dev/sde` zu finden ist. Analog funktioniert diese Einbindung bei Anbindungen von SAN bzw. NAS-Speichersystemen. Der Datenbank-Administrator muss sicherstellen, dass Daten der Datenbank immer auf dem gleichen Namen und mit entsprechenden Rechten abgelegt sind.

Eine Lösung : **ASMLIB**

Die Library ASMLib wird von Oracle zur Verfügung gestellt, um unter Linux Partitionen oder LUNs als ASM-Disk der Oracle ASM-Instanz bereitzustellen. Sie besteht aus den drei Paketen `oracleasm`, `oracleasm-support` sowie `oracleasm-lib`. Das Paket `oracleasm-support` ist vom entsprechenden Kernel abhängig.

ASMLib sorgt für

- Erstellen von persistenten Device-Namen
- Erstellen der korrekten Lese- und Schreibrechte auf dem Device
- Asynchroner I/O

Sowohl die Installation als auch die Implementierung ist unter

<http://www.oracle.com/technology/tech/linux/asmlib/install.html>

ausführlich beschrieben.

Die Alternative **udev**

Das auf den meisten Linux-Systemen inzwischen standardmäßig verwendete udev ist eine ausgereifte, mächtige und flexible Art der Einbindung von Geräten über einstellbare Regeln in den /dev-Baum. udev hat sich inzwischen als beste Alternative zu statischen Gerätedateien gegen das dev-Filesystem durchgesetzt. Der Administrator kann mithilfe des Filesystems fssys udev - Regeln erstellen, um Geräte unter einem persistenten Namen und entsprechenden Rechten in den Verzeichnisbaum einzubinden. Es handelt sich bei sysfs um ein Kernel- Filesystem, das Informationen bzgl. der angeschlossenen Geräte exportiert. udev kann diese Informationen nutzen, um entsprechende Namen für die an das System angeschlossenen Geräte zu erhalten.

udev Regeln sind im Verzeichnis */etc/udev/rules.d/ hinterlegt*. Default Regeln sind bei RHEL5 z.B. in der Datei */etc/udev/rules.d/50-udev.rules* einsehbar.

Regelsyntax:

Jede Regel wird angelegt mit einer Serie von Key-Value-Paaren, die durch Kommata getrennt sind. Vergleichende Regeln werden mit einem == aufgeführt, zuweisende Regeln werden mit einem = aufgeführt. Anbei ein Beispiel für eine solche Regel:

```
KERNEL=="hdb", NAME="meine_platte"
```

Der erste Eintrag ist der Vergleich (==). Treffen alle vergleichenden Key-Value-Paare zu, werden die Zuweisungsregeln aktiv. Im Beispiel wird die Festplatte hdb unter dem Namen */dev/meine_platte* eingebunden. Wichtig ist, dass eine Serie von Paaren zu einem Gerät in einer Zeile steht. In dem Vergleichsoperator == können folgende Wildcards benutzt werden:

* : Trifft auf jedes Zeichen beliebig oft zu oder

? : Trifft auf ein beliebiges Zeichen genau einmal

[] : Trifft auf ein einzelnes Zeichen in der Klammer (auch Wertebereiche sind erlaubt)

Der sysfs-Verzeichnisbaum

Neben dem oben aufgeführten Beispiel einer Basic-Regel wird man im Datenbankumfeld auf den sysfs-Verzeichnisbaum zugreifen. Dieser bietet detaillierte Informationen zu dem anzuschließenden Gerät wie Herstellercode, Produktnummern oder Seriennummern.

So werden in `/sys/block` die Festplatten eingebunden. Auf dem Laptop (Ubuntu 8) ist z. B. die Festplatte unter `/sys/block/sda` eingebunden. Man kann mit

```
$ cat /sys/block/sda/size
234441648
```

die Größe der Festplatte ermitteln. Da es jedoch sehr mühselig wäre, alle Informationen mit einzeln abzusetzenden Befehlen abzurufen, kann man die Attribute eines Gerätes auch mit dem Befehl `udevinfo` komplett abrufen.

Hier als Beispiel die gekürzte Ausgabe einer angebundenen 600 GB LUN eines HP-SAN, auf der eine Partition angelegt wurde (SLES10).

```
udevinfo -a -p /sys/block/sdc/sdc1

looking at device '/block/sdc/sdc1':
  KERNEL=="sdc1"
  SUBSYSTEM=="block"
  SYSFS{stat}=="65420704 3022599714 28338044 1533618492"
  SYSFS{size}=="1171797102"
  SYSFS{start}=="63"
  SYSFS{dev}=="8:33"

looking at device '/block/sdc':
  ID=="sdc"
  BUS=="block"
  DRIVER==" "
  SYSFS{stat}=="62116097 3305008 3022603730 588279328 28322564 10015
1533618492 56523756 0 78051680 645285372"
  SYSFS{size}=="1171808256"
  SYSFS{removable}=="0"
  SYSFS{range}=="16"
  SYSFS{dev}=="8:32"

looking at device
```

```

'/devices/pci0000:00/0000:00:10.0/host0/target0:0:2/0:0:2:0':
  ID=="0:0:2:0"
  BUS=="scsi"
  DRIVER=="sd"
  SYSFS{ioerr_cnt}=="0x2"
  SYSFS{iodone_cnt}=="0x563ff8e"
  SYSFS{iorequest_cnt}=="0x563ff8e"
  SYSFS{iocounterbits}=="32"
  SYSFS{retries}=="5"
  SYSFS{timeout}=="60"
  SYSFS{state}=="running"
  SYSFS{rev}=="J200"
  SYSFS{model}=="MSA2212fc      "
  SYSFS{vendor}=="HP          "
  SYSFS{scsi_level}=="6"
  SYSFS{type}=="0"
  SYSFS{queue_type}=="simple"
  SYSFS{queue_depth}=="32"
  SYSFS{device_blocked}=="0"

```

Wie zu erkennen, produziert udevinfo eine Liste von Key-Value-Paaren, um damit leicht ein Gerät eindeutig zuordnen zu können. Der erste Abschnitt beschreibt den Pfad (angegeben mit -p im udevinfo-Befehl.) Die Folgenden sind Parent-Beziehungen.

Man kann immer nur eine Parent-Beziehung nutzen. Würde man verschiedene Parent-Beziehungen mischen, so würde man die Regel nicht anwenden können, da udev hierarchisch die Eigenschaften des Baumes durchläuft.

In dem konkreten Beispiel nimmt man den Kernel und das Subsystem aus dem Pfad (der Partition). Die restlichen Informationen stammen aus dem 2. Parent-Block.

Damit kann das SAN eindeutig mit

```

KERNEL=="sd?1", SUBSYSTEM=="block", BUS=="scsi", DRIVER=="sd",
SYSFS{model}=="MSA2212fc      ",SYSFS{vendor}=="HP          "

```

identifiziert werden.

Durch das Fragezeichen im KERNEL ist es nun egal, ob das SAN mit sdc1, sde1 oder unter einem anderen Buchstaben eingebunden wird.

Anschließend weist man diesem Device die Werte

```
NAME="oracle_daten", GROUP="dba",OWNER="oracle",MODE="0660"
```

zu. Dadurch benennt man die Partition `oracle_daten`, die dem User `oracle` in der Gruppe `dba` gehört und die Rechte `0660` hat. Diese beiden Strings werden durch Kommata separiert in der Datei `10.local-rules` gespeichert. (`udev` liest das Verzeichnis `/etc/udev/rules.d/` aufsteigend mit den Zahlen vor der Regel durch) Die Regel kann man mit Hilfe des Befehls `udevtrigger` anwenden und anschließend ist die LUN unter `/dev` eingebunden.

```
$ ls -l dev/or*
brw-r----- 1 oracle disk 8, 33 Dec 10 12:03 dev/oracle_daten
```

In der ASM-Instanz kann dann die Datengruppe mit folgendem Befehl angeschlossen werden:

```
SQL> alter system set asm_diskstring = '/dev/oracle*'
scope=both;
SQL> create diskgroup daten external redundancy disk
'/dev/oracle_daten';
```

Hierbei kann man sicher sein, dass die LUN stets unter `/dev/oracle_daten` mit den korrekten Rechten angesprochen werden kann.

Vor - und Nachteile gegenüber ASMLib:

Mit ASMLib kann etwas einfacher mit einem einzigen Befehl eine LUN bzw. eine Festplatte gelabelt werden. Dadurch kann man auch, ohne sich in `udev` einlesen zu müssen, recht schnell eine Festplatte einbinden. Arbeitet man an einem RAC, so ist eine LUN, die einmal von einem Knoten gelabelt wurde, direkt über den ASMLib Diskstring `ORCL:*` von allen Knoten nach Absetzen des Befehls

```
/etc/init.d/oracleasm scandisk
```

als ASM-Kandidat in ASM ansprechbar.

Arbeitet man mit `udev`, so muss man die `udev`-Regel erstellen. Das Erstellen ist etwas aufwendiger als die Einbindung mit ASMLib. Im Fall eines RACs muss die

Regel auf alle Knoten kopiert werden. Anschließend ist nach Ausführen von udevtrigger die LUN in der ASM-Instanz ansprechbar.

Nutzt man udev, ist es nicht nötig zusätzliche Packages zu installieren und hat dadurch auch keine Kernelabhängigkeit. Führt man ein Upgrade des Kernel durch, so muss man sich nicht um die ASM-Integration kümmern, wodurch man eine kritische Fehlerquelle ausgeschaltet hat.

Im RAC-Umfeld wird nicht nur auf die ASM-Festplatte von verschiedenen Knoten zugegriffen, sondern persistent zusätzlich auf die Voting- und die OCR-Disks. Diese würde man dann mit udev einbinden. Die Architektur ist somit einheitlicher, wenn man dann die ASM-Disks auch über den gleichen Algorithmus anlegt.

FAZIT

Sowohl ASMLib als auch udev sind robuste und sichere Möglichkeiten zur Rechtevergabe und persistenten Geräteeinbindung. udev wird vom Standard-Linuxkernel direkt unterstützt, was ein nicht zu unterschätzender Vorteil gegenüber der Lösung ASMLib ist. Da man zusätzlich keine Kernel-Abhängigkeit hat, empfiehlt der Autor den Einsatz von udev.

Kontakt:

Claus Cullmann

claus.cullmann@eXirius.de