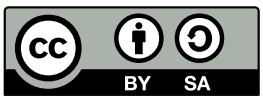
Lokales Storage Teil 1

Zinching Dang

12. August 2014



Lokales Storage im Allgemeinen

Lokales Storage im Allgemeinen

- Datenträger, die direkt am Host angeschlossen sind
- Anbindung über verschiedene Bus-Systeme möglich, z. B. SAS, SATA
- Verwaltung durch das Betriebssystem oder durch einen Hardware-Controller

Übliche Datenträger, die für lokales Storage verwendet werden, können Festplatten oder SSDs sein. Letztere werden aufgrund ihrer vergleichsweise kleinen Kapazitäten und hohen Preises meist als Caches eingesetzt, um Zugriffe auf oft benötigte Daten zu beschleunigen. Im Server-Umfeld sind vor allem SAS und SATA Bus-Systeme anzutreffen, bei manchen SSDs auch PCIe. Die verbauten Datenträger können dabei durch spezielle (Hardware-) RAID–Controller verwaltet werden oder über einen normalen Storage-Controller vom Betriebssystem selbst.

Partitionierung

Partitionierung

- Aufteilen des gesamten Speicherbereichs einer Festplatte in mehrere, kleinere Bereiche
- notwendig, um mehrere Betriebssysteme oder Dateisysteme zu verwenden
- Aufteilung steht in der sog. **Patitionstabelle**
- verschiedene Standards für Partitionstabellen, u. a. MBR und GPT
- befindet sich im Anfangs-/Endbereich des Datenträgers

Um den Speicherplatz auf Datenträgern effizient nutzen zu können, ist es in vielen Fällen sinnvoll, den Datenträger zu partitionieren. Hierbei wird der gesamte Speicherbereich in kleinere Bereiche aufgeteilt und kann z. B. für verschiedene Betriebssysteme oder Dateisysteme verwendet werden. Wie diese Aufteilung im Detail aussieht, wird in der Partitionstabelle festgelegt. Es gibt verschiedene Standards für diese Tabelle, zu den gebräuchlichsten gehören MBR und GPT. Die Partitionstabelle befindet sich im Anfangs- bzw. End-Bereich des Datenträgers.

Partitionstabellen

Partitionstabellen (1)

- **M**aster **B**oot **R**ecord
  - belegt die ersten 512 Bytes des Datenträgers
  - enthält den Code des Bootloaders und die Partitionstabelle
  - primäre und erweiterte Partitionen (max. 4 Partitionen insgesamt, max. 1 erweiterte Partition)
  - innerhalb einer erweiterten Partition: beliebig viele logische Partitionen
  - max. Datenträgergröße auf 2 TiB beschränkt (bei 512-Byte-Sektoren)

Der Master Boot Record (MBR) ist ein älterer Standard für die Beschreibung der Partitionstabellen. Der MBR befindet sich in den ersten 512 Bytes des Datenträgers und ist zum Großteil durch den Boot-Code für das Betriebssystem belegt. Es wird zwischen primären und erweiterten Partitionen unterschieden. Insgesamt können bis zu vier Partitionen angelegt werden, jedoch nur eine erweiterte. In der erweiterten Partition selbst können anschließend beliebig viele logische Partitionen angelegt werden. Eine weitere Beschränkung ist die maximale Datenträgergröße von 2 TiB (wenn 512-Byte-Sektoren verwendet werden). Dies ist durch die Verwendung von 32-Bit-Zahlen für die Adressierung historisch bedingt.

Partitionstabellen

Partitionstabellen (2)

- GUID (Globally Unique Identifier) Partition Table
  - MBR-Nachfolger, Teil des UEFI-Standards
  - beliebig viele Partitionen, max. Datenträgergröße 8 ZiB
  - GPT-Header im Anfangsbereich (Primary GPT-Header) und im Endbereich (Secondary GPT-Header) des Datenträgers

Der Nachfolger des MBR-Standards ist GUID (Globally Unique Identifier) Partition Table und ist Teil des UEFI-Standards (BIOS-Nachfolger). Mit GPT werden einige Beschränkungen des MBR-Standards gelöst, u. a. wird nicht mehr zwischen primären und erweiterten Partitionen unterschieden, es gibt auch keine maximal Anzahl an Partitionen mehr. Der größte Vorteil ist jedoch die Beschränkung der Datenträgergröße auf 8 ZiB (Präfix-Reihenfolge: Ki, Mi, Gi, Ti, Pi, Ei, Zi jeweils in 2^{10}-er Schritten). Neu ist auch, dass die Partitionstabelle redundant vorhanden ist, einmal im Anfangsbereich (Primary GPT Header) und einmal im Endbereich (Secondary GPT Header) des Datenträgers.

Partitionierungstools

Partitionierungstools

- `*fdisk` (nur MBR, kein GPT) und `parted`
- Partitionierung per Kommandozeilen-Befehl oder interaktiv möglich
- auch als GUI-Tools verfügbar: gparted, qtparted
- hier: nur `parted`

Das klassische Partitonierungstool `fdisk` und seine Derivate (`cfdisk` und `sfdisk`) unterstützen nur das MBR-Format. Soll die Festplatte mit GPT partitioniert werden, wird das modernere `parted` benötigt. Für `parted` sind auch grafische Frontends wie gparted und qtparted verfügbar.

parted

parted

- Aufruf: `parted /dev/sda`
  - bearbeitet die Partitionstabelle des Datenträgers `sda`
- wichtigsten Kommandos:
  - `help` und `help <CMD>`: allgemeine Hilfe und Hilfe zu einem bestimmten Kommando
  - `print`
  - `select <DEVICE>`
  - `mktable`
  - `mkpart`
  - `set`

Lab 6.1: Anlegen und Einbinden virtueller Festplatten

Lab: Anlegen und Einbinden virtueller Festplatten

- mit dem `virt-manager` zwei virtuelle Festplatten anlegen und an den eigenen Server anbinden
- überprüfen, ob die neuen Festplatten erkannt wurden
- Festplatten partitionieren

RAID

RAID

- **R**edundant **A**rray of **I**ndependent **D**isks
- meist werden mehrere Festplatten zusammen im Verbund betrieben
  - (höhere) Redundanz
  - größere Gesamtkapazitäten
  - höherer Datendurchsatz
- Hardware-Realisierung als spezieller RAID-Controller
- Software-Realisierung als Treiber
- bei sog. Fake-RAID ist nur ein Datenträger sichtbar, die RAID-Operationen werden jedoch in der CPU ausgeführt

Im Server-Umfeld spielt die Ausfallsicherheit eine wichtige Rolle, daher ist es erwünscht, dass der Ausfall einer Komponente nicht zu einem Ausfall des Gesamtsystems führt. Redundanz bei Datenträgern wird durch RAID-Verbunde geschaffen. Die frühe Bedeutung „Redundant Array of Inexpensive Disks“ wurde später durch „Redundant Array of Independent Disks” ersetzt, da heutige Datenträger für den Server-Bereich keineswegs köstengünstig sind. Ein RAID-Verbund kann mehrere Ziele haben, z. B. Redundanz schaffen, die Gesamtkapazität erhöhen, den Datandurchsatz erhöhen, etc. Diese können in Hardware mit einem RAID-Controller oder in Software mit entsprechenden Treibern realisiert werden. Hybride Ansätze, sog. Fake-RAID, verwenden spezielle Treiber in Kombination mit Controllern, an die zwar mehrere Datenträger angeschlossen werden, an das Betriebssystem jedoch nur ein Datenträger gemeldet wird.

Hardware- und Software-RAID

Hardware- und Software-RAID

- Hardware-RAID ist performanter, da der RAID-Controller viele Aufgaben übernimmt
- bei einem Controller-Defekt wird ein identischer RAID-Controller benötigt
- Software-RAID ist sehr flexibel und unabhängig von der verbauten Hardware
- langsamer und Resourcen-intensiver, da alle RAID-Operationen in der CPU ausgeführt werden

Hardware- und Software-RAID haben ihre Vor- und Nachteile. Der größte Unterschied ist, dass bei einem Hardware-RAID-Controller viele RAID-Operationen im Controller abgearbeitet werden und dadurch bei geringer CPU-Last ein höherer Durchsatz erreichbar ist. Fällt jedoch ein RAID-Controller durch einen Defekt aus, muss dieser durch ein identisches Modell, meist sogar mit gleicher Firmware-Version, ersetzt werden, andernfalls ist der Zugriff auf die Daten nicht mehr möglich. Darin liegen auch die Stärken von Software-RAID: durch die Verwaltung der Datenträger in Software ist es unabhängig von den Controllern. Der unmittelbare Nachteil ist, dass die CPU stärker belastet wird, da die RAID-Operationen dort ausgeführt werden.

RAID-Level

RAID-Level

- viele Möglichkeiten, Festplatten im RAID anzuordnen
- verschiedene RAID-Level bieten unterschiedlich hohe Daten-Sicherheit
- am Gebräuchlichsten:
  - (RAID 0)
  - RAID 1
  - RAID 5
- Kombinationen verschiedener RAID-Level auch möglich (z. B. RAID 10, RAID 50)

RAID-Level

RAID-Level

- RAID 0: Striping
  - bei `n` Festplatten werden die Daten auf alle `n` Festplatten verteilt
  - Gesamtkapazität entspricht Summe der Einzelkapazitäten
  - jedoch keine Redundanz und Defekt einer Festplatte führt meist zu Verlust **aller** Daten
- RAID 1: Mirroring
  - bei `n` Festplatten werden die Daten auf alle `n` Festplatten gespiegelt
  - sehr hohe Redundanz, Ausfall von `n-1` Festplatten ohne Datenverlust möglich
  - keine Erhöhung der Gesamtkapazität, Gesamtgröße hängt von der kleinsten Festplatte ab

RAID-Level

RAID-Level

- RAID 5
  - Kompromiss bzgl. Redundanz und Gesamtkapazität
  - benötigt mindestens 3 Festplatten
  - bei `n` Festplatten werden die Daten zusammen mit Paritäts-Informationen auf alle `n` Festplatten verteilt
  - `(n-1)`-fache Gesamtkapazität und Ausfall einer Festplatte möglich

MD-RAID

MD-RAID

- Software-RAID-Lösung unter Linux
- wird durch das Paket `mdadm` bereitgestellt
- u.a. werden RAID 0, RAID 1, RAID 5 und RAID 10 unterstützt
- aktuelle Informationen über das RAID befinden sich in `/proc/mdstat`

mdadm

mdadm

- Aufruf: `mdadm <MODE> <DEVICE> <OPTION> <COMPONENT>`
  - `MODE`: Bearbeitungs-Modus
  - `DEVICE`: MD-RAID-Device
  - `OPTION`: Optionen für den Modus
  - `COMPONENT`: Komponenten des MD-RAID
- wichtige Modi:
  - `--create`: neues MD-RAID erzeugen
  - `--grow`: ein vorhandenes MD-RAID ändern
- wichtige Optionen:
  - `--raid-devices=<N>`: Anzahl der Datenträger im MD-RAID festlegen
  - `--level=<X>`: RAID-Level festlegen
  - `--add`: neue Datenträger als Spare zu einem bestehenden MD-RAID hinzufügen

Lab 6.2: MD-RAID

Lab: MD-RAID

- RAID 1 über die beiden Partitionen erzeugen
- Status des RAID überprüfen