Jörg Schilling

Filesysteme

Ausblicke

Fokus Fraunhofer

Filesysteme jenseits vom Desktop

- Zuverlässigkeit
  - Viele Festplatten, RAID
- Geschwindigkeit (z.B. durch SSD)
- Snapshots (Einfrieren des FS Zustandes)
  - Blocklayer Cache, Copy On Write (COW)
- Verteilte Dateisysteme
  - Mehrere Server, mehrere Klienten
- Hierarchisches Speichermanagement
  - Schneller Dateisystem Cache und langsamer Hintergrundspeicher
- Copy On Write Filesysteme
  - WOFS (1991), WAFL (1994), ZFS (2001), BtrFS (2007)

Zuverlässigkeit

- **Große Filesysteme benötigen mehr als eine Platte**
- **Mit vielen Platten reduziert sich die „Lebensdauer“**
  - **Siehe MTBF**
- **Reduzierte Lebensdauer bedeutet höhere Ausfallrate**
- **Einfachste Lösung: Spiegeln der Inhalte**
  - **Nachteile: Teuer**
- **RAID (R**edundant **A**rray of **I**nexpensive/**I**ndependent **D**isks**)**
  - **z.B. 4 Platten: 3x Daten,1x EXOR der 3 Anderen**

RAID

- RAID 0: Vergrößerung des Speichers, mehrere Platten
- RAID 1: Spiegelung. z.B. 2 Platten
- RAID 5: Mehrere Platten + 1x Redundanz (Parität)
  - z.B.:
  - 3 Platten mit Daten (Striping)
  - 1 Platte mit Platte1 EXOR Platte2 EXOR Platte3
  - Nachteile: Nach Ausfall einer Platte nur noch Lesen
- RAID 6: Mehrere Platten + nx Redundanz (Parität)
  - z.B. 2x Redundanz, Weiterschreiben nach Ausfall

- Platten werden durch einen eigenen Rechner verwaltet
- Erhältlich seit mehr als 30 Jahren
- Nachteile:
  - Welche Daten sind richtig, wenn Bits kippen?
  - Alle Platten müssen vollständig initialisiert werden
  - Nach Austausch wieder Regeneration aller Blöcke
  - Viele Datenverbindungen ohne Sicherung
- FAZIT: Sichert nicht gegen BITROT

SSD (Solid State Drive)

- **Vorteile:**
  - **Deutlich verringerte „Seek-Zeiten“**
  - **Paralleler Zugriff auf mehrere Adressbereiche**
  - **Geringerer Ruhestromverbrauch**
  - **Stoßfestigkeit**
- **Nachteile:**
  - **Latenz durch notwendige Blockgröße**
  - **Überschreiben sehr langsam (vorher Löschen)**
  - **Geringere Anzahl von Schreibzyklen**
- **Qualitative Unterschiede (Heimsysteme ↔ Profisysteme)**

Snapshots

- Einfrieren eines Dateisystem-Zustandes
- Verfügbar seit ca. 25 Jahren
- UFS: Blocklevel Cache: alternativer Hintergrundspeicher
  - FreeBSD: Cache in freien Blöcken des FS
  - Solaris: Separates Cache- (Backstore-) Device
- Benötigt Platz für jeden modifizierten Block
- Nachteile: Anlegen kann mehrere Minuten dauern
- Häufig nur Read-Only (z.B. UFS)
- COW Cache: Status sichern, siehe auch ZFS und WAFL

- Echtes verteiltes System: z.B. Andrew Filesystem
  - Mehrere Server bilden ein großes Dateisystem
- Zur Ausfallsicherheit und Geschwindigkeit: QFS
  - Mehrere Server mounten ein Blockdevice (z.B. NAS)
  - Problem hier: Cachekohärenz zwischen den Servern
  - Lösung: Locks für Bereiche
- Zwischenlösung: NFSv4
  - Delegation von einzelnen Dateien an andere Server

Hierarchisches Speichermanagement (HSM)

- Sehr große Dateisysteme mit automatischem Backup
- Hierarchisch, weil mehrere Ebenen Speicher
  - z.B. Festplattencache, Hintergrundspeicher auf Band
  - Meta-Daten sind immer auf dem Plattencache
  - Selten genutzte Dateien auf Band ausgelagert
  - Automatische Arbeit mit Band-Robotersystem
- SAM-QFS (Solaris)
- GPFS (AIX) mit HSM Erweiterung erhältlich

Beispiel eines HSM in einem Berliner Institut

- 1 PB Plattencache für Metadaten und Dateiinhalte
  - z.B. 500 2 TB Platten
- Bandroboter
  - 70 Bandlaufwerke
  - 20000 Medienslots (Bänder), 15500 in Benutzung
  - 4,5 TB pro Band, 250 MB/s Schreibgeschwindigkeit
  - Archivierung im TAR Format
  - 70 PB Bandspeicher → 35 PB in zwei Kopien
- Nicht gecachte Daten werden in 90-540 Sek. Geladen
- Verwendete Software: SAM-QFS

Beispiel: SAM-QFS

- **S**torage **A**rchive **M**anager **– Q**uick **F**ile**S**ystem
- **Seit 2008 OpenSource**
- **Integriertes Platten Volume Management**
- **Minimalkonfiguration Plattencache + Band**
- **Metadaten können separat gecached werden**
- **Ausfallsicherheit durch Multiwriter Plattenanbindung**
- **Nach Crash wird nur der Metadatencache restauriert**
  - **Das Datencache füllt sich danach von Alleine**
  - **Das kann allerdings Wochen dauern**

ZFS, WAFL, …

- Moderne „COW“ Dateisysteme
  - Copy on Write → Blöcke nicht Überschreiben
- WOFS (Worm Filesystem) Entwickelt 1988-1991
  - Vermutlich erstes COW Dateisystem
  - Erstes Filesystem ohne fsck
- WAFL (Write Anywhere File Layout) Entwickelt ab 1993
- ZFS Entwickelt ab 2001

- Kein Block auf dem Speichermedium wird überschrieben
- Jeder Block bekommt beim Schreiben neue Adresse
- → Block mit der Liste der Blocknummern auch neu
- → „Inode“ für Datei auch neu
- → Blockliste der Directory ändert sich auch
- …
- Superblock ändert auch Speicherort
- Notwendig für WORM
- Hilfreich für SSD

Probleme mit COW

- Probleme den aktuellen „Superblock“ zu finden
- Lösung:
  - Bereich für Superblöcke festlegen
  - Suchalgorithmus für den darin neuesten Block
- Datei geändert → Blockliste für die Datei geändert
- „Inode“ für Datei neu schreiben
- Zugehörige Directory neu schreiben
- Das geht weiter bis zum Root-Dorectory
- Lösung (nur WOFS): Invertierte FS Struktur

Probleme mit COW

- Invertierte FS Struktur (WOFS):
  - Nicht Directory enthält Zeiger auf Dateien
  - Sondern Datei enthält Zeiger auf Directory
- Normale FS Struktur (ZFS):
  - Alle Directories von Datei bis Root neu schreiben
  - Fsync() einer Datei ist genauso teuer wie FS sync()
  - „Sichere“ Archiv Extraktion damit ca. 4x langamer
- Fazit: COW ist immer konsistent
  - Aber Konsistenz m. Bekanntem Zustand langsam

- Jeder Block hat eine Checksumme
- Integriertes Software RAID System
  - Separate Checksumme für jeden RAID Block
  - Superblock enthält eigene Checksumme
- ZFS-Pool → Blocklevel Verwaltung
- Darauf ZFS Dateisysteme, oder iSCSI NAS
- Alle Operationen Copy on Write
- Quotas für Dateisysteme, Nutzer, Gruppen
- Native NFSv4 (NTFS) ACLs
- Schreibbare Snapshots

Danke!

- **PDF dieser Vorlesung: http://cdrecord.org/Files/**

Es geht noch weiter mit ZFS...

- **Vortrag ZFS Datenstrukturen von Uli Gräf († 4.6.2013)**
  - **Gehalten u.a. am 30.10.2009 in Dresden**
  - **Verfügbar als Youtube Video (40min) mit Uli Gräf**