# OPEN STORAGE + ZFS

**Constantin Gonzalez**
Principal Field Technologist
Sun Microsystems GmbH

# Opensolaris Storage Software

# Eine wahre Geschichte

- Mail-Server für eine ganze Uni
- 2 Server, 2x RAID-Storage, FC-AL SAN
- Alles redundant ausgelegt
- UFS Logging aktiviert
- Regelmäßiges Backup
- Regelmäßige Restore-Tests
- Alles richtig gemacht?

# Doch...

- System Panic wg. “freeing free inode”
- fsck(1M) dauerte >10 Stunden
- Danach: Wieder ein Panic, gleicher Fehler!
- Backup einspielen:
  7 Stunden + rsync
- Ursache:
  Defekter FC-Adapter

# ZFS ganz kurz

## Der ZFS Spickzettel

- Storage Pools statt Volume Manager/Filesystem
- Skalierbare Baum-Struktur mit 128-Bit-Adressierung
- Transaktions-orientiert mit Copy-on-Write Semantik
- Einfache, schnelle, effiziente Snapshots/Clones
- Durchgängige Datenintegrität
- Selbst-reparierende Spiegel und RAIDs
- Ganz einfach zu administrieren

=> Ein frischer, moderner Ansatz für Dateisysteme!!

# Storage Pools

- Früher
  - > Volume Manager und Dateisystem getrennt
  - > Keiner weiß, was der andere tut
  - > Verschnitt durch mangelnde Flexibilität
  - > Fragmentierung von I/O-Kapazität

- Heute, mit ZFS Pools
  - > Volume Manager und Dateisystem in einem
  - > Durchgängige Optimierung
  - > Maximale Flexibilität und Speicher-Nutzung
  - > Beste, aggregierte I/O-Performance

# Skalierbare 128 Bit Baum-Struktur

- 128-Bit Adressierung
- Schnell und skalierbar: Neues FS = wenige Blöcke schreiben, fertig!
- Datenstruktur wächst mit der Nutzung mit
- Flache Hierachie:
  - 256 Pointer pro MD-Block
  - Wenig Ebenen
- Metadaten- und Über-Blöcke sind mehrfach redundant gespeichert
- Dynamische Block-Größe

# Copy-on-Write Transaktionen

# Snapshots

- Read-only, Zeitpunkt-Kopie eines ganzen Dateisystems
- Dank Baumstruktur und CoW einfach zu implementieren: Aktuellen Über-Block merken = Neuer Snapshot
- Platzsparend - nur Änderungen verbrauchen Platz
- Clone = Snapshot, der schreibbar ist.

# Durchgängige Datenintegrität

## Vertrauen ist gut – Kontrolle ist besser!

- **Bit-Fäule**
- **Phantom-Schreiboperationen**
- **Verirrte Lese/Schreib-Operat.**
- **DMA Paritäts-Fehler**
- **Treiber-Bugs**
- **Versehentliches Überschreib.**

- Prüfsummen beim Block
- Von der Platte erzeugt
- i.d.R.: Nur 8 Bit
- Fehler möglich!

- Prüfsummen bei den Pointern
- Verkettet: Validierender Merkle-Baum
- Algorithmus wählbar
  - > Aktuell: fletcher2, fletcher4, SHA256
- Fehler sehr unwahrscheinlich!

- **Bit-Fäule**
- **Phantom-Schreiboperationen**
- **Verirrte Lese/Schreib-Operat.**
- **DMA Paritäts-Fehler**
- **Treiber-Bugs**
- **Versehentliches Überschreib.**

# ZFS repariert sich selbst!

## Copy-on-write + verkettete Checksummen = Verläßlicher RAID-Durchblick

- Erkennung kaputter Datenblöcke durch Checksummen
- Kaputte Blöcke durch redundante Daten reparieren
- Im laufenden Betrieb, oder manuell per “Scrub”

Kaputte Daten erkannt

Holt gute Daten vom Spiegel

Repariert kaputte Seite

# RAID-Z und RAID-Z2

## Oder: Wie SW-RAID besser sein kann als HW-RAID

- Traditionelles RAID-5

- n+1 Platten, toleriert 1 Ausfall
- Read-Modify-Write = 3 Operationen
  - >Random-Writes = Hölle
- Vorsicht: Schreib-Loch!
- Daher: RAID-Controller mit Cache

- RAID-Z

- n+1 Platten, toleriert 1 Ausfall
- Nur Write, da Copy-on-Write = 1 Op.
  - >Random-Writes -> Sequential Writes
- Kein Schreib-Loch, da CoW
- Dynamische Stripe-Breiten

- RAID-Z2

- Wie RAID-Z, aber mit 2 Parity-Platten
- Toleriert 1 oder 2 defekte Platten
- Analog zu RAID-6, aber mit den Vorteilen von CoW

# Einfach zu administrieren

Terminal

File Edit View Terminal Tabs Help

```
# man zpool
# man zfs
#
# zpool create sesamstrasse mirror c0d0p0 c1d0p0
# ls /sesamstrasse
#
# zfs create sesamstrasse/home
# zfs set quota=1g sesamstrasse/home
# zfs create sesamstrasse/home/ernie
# zfs create sesamstrasse/home/bert
# zfs create sesamstrasse/home/kruemelmonster
#
# zpool add sesamstrasse mirror c0d1p0 c1d1p0
#
# zfs snapshot -r sesamstrasse/home@montag
# zfs rollback sesamstrasse/home/ernie@montag
```

# ... und noch viel mehr!

- Attribute
  - Kompression, NFS-Share, Quotas/Reservierung, etc.
- Hierarchische Dateisysteme mit Vererbung
  - Einmal Attribute für home setzen, tausende User verwalten
- Einfache Replikation/Backup
  - zfs send/receive wandelt Snapshots in Bit-Ströme
- NFS V4/NT ACLs
  - Erweiterte Schutzmassnahmen, z.B. Append-Only
- ZVOLs: Block Device Volumes mit ZFS
  - Für andere FS, Datenbanken, iSCSI-Targets

# Tipps für die Praxis

- Ausprobieren!
  - Zuhause, Laptop, Pilot-Projekt, etc.
- Mit Pools die Hardware-Seite administrieren
  - 1 Pool pro Storage-Qualität (Mirror, RAID-Z(2), etc.)
- Mit Filesystemen die User-Seite administrieren
  - 1 FS pro User, 1 FS pro Kunde, 1 FS pro Applikation usw.
- Snapshot early, snapshot often!
  - Per Cron-Job: Wöchentlich, täglich, stündlich, minütlich...
  - Pro Login, z.B. auf dem Fileserver
- Synergien mit NFS, iSCSI, SMF und anderen OS-Diensten nutzen

# ZFS und iSCSI = Das perfekte Team

- Einfach zu administrieren
  - > zfs create -V 1t pool/volume
  - > zfs set shareiscsi=on pool/volume
- Flexibel einsetzbar
  - > Solaris, Linux, Windows Initiators
  - > Erzeugen/Löschen von Volumes in Sekunden
  - > Sparse Volumes, Snapshots, Send/Receive, Quotas ...
- Performant
  - > 90-100 MB/s pro GBE beim Kunden gemessen
  - > Client-Tuning beachten

# ZFS Architektur

- ZPL: ZFS POSIX Layer
- ZVOL: ZFS Volumes
- ZIL: ZFS Intent Log
- ZAP: ZFS Attribute Processor
- DMU: Data Management Unit
- DSL: Dataset & Snapshot Layer
- ARC: Adaptive Replacement Cache
- ZIO: ZFS I/O Pipeline
- VDEV: Virtual Devices
- LDI: Layered Driver Interface

http://www.opensolaris.org/os/community/zfs/source/

# Mehr ZFS!

- man zpool(1M); man zfs(1M)
  - Da steht alles verständlich drin, sonst ist es ein Bug
- docs.sun.com
  - ZFS Administration Guide, man-Pages
- www.opensolaris.org
  - In der ZFS Community, zfs-discuss Mailingliste
- www.solarisinternals.com
  - ZFS Best Practices Guide Wiki
- blogs.sun.com/solarium
- “Das OpenSolaris ZFS Dateisystem”
  - Whitepaper zum LinuxTag 2007

# Festplatten...

- Können immer mehr speichern
  - 60% mehr GB pro Platte pro Jahr
  - Oder: 60% weniger Kosten pro GB
- Werden aber kaum schneller!
  - Ca. 125-250 IOPS pro Platte
  - Seit >10 Jahren: 7200-15K RPM
- Gehen dafür genauso häufig kaputt
- Server werden trotzdem immer schneller
  - Moore's Law: 50-60% schneller pro Jahr
- IOPS pro GB werden daher immer weniger!

5MB, 1956

1TB, 2008

# Das Problem in Kürze

# Flash Memory...

TOSHIBA
組み込み型NAND型フラッシュメモリ

- Wird immer billiger
  - >50% günstiger pro GB pro Jahr
- Ist zuverlässig
  - >5 Jahre Dauer-Schreib-Betrieb für SLC Flash
  - >Keine mechanischen Komponenten
- Sparsam im Strom- und Platz-Verbrauch
- Und vor allem schnell
  - >30.000 IOPS pro Controller beim Lesen
  - >10.000 IOPS pro Controller beim Schreiben
  - >100x schneller als Festplatten!
- “Enterprise Flash” ist auf dem Vormarsch

# Die Lösung: Flash Memory

# Die neue Speicher-Hierarchie

# ZFS kann das schon!

- ZFS is transaktional
  - > wie eine Datenbank
- ZFS Intent Log (ZIL)
  - > Zwischenspeicher für synchrones Schreiben
  - > ZIL im Flash:„Writezilla“
- Adaptive Read Cache (ARC)
  - > Cache für's Lesen
  - > Flash als Lese-Cache: L2ARC „Readzilla“
- Die Zukunft: Hybride Pools

# Beispiel für hybriden ZFS Pool

## Beim letzten IDF in Shanghai vorgestellt

- 4 Xeon 7350 CPUs (16 Cores)
- 32GB FB DDR2 ECC DRAM
- OpenSolaris mit ZFS

Konfiguration A:

7x146GB 10,000 RPM SAS Platten

Konfiguration B:

1x32G Flash ZIL “Platte”

1x80GB Flash L2ARC “Platte”

5x400GB 4200 RPM SATA Platten

# Ergebnisse: Hybrider ZFS Pool

- Gleiche Kosten, mehr Performance, weniger Verbrauch
- Bei großen Konfigurationen (48T–750T+) werden die Kosten komplett von der Ersparnis getragen

# ZFS und verschiedene OSe

- ZFS ist Teil von OpenSolaris
  - Open Source mit OSI-geprüfter CDDL-Lizenz
- Seit 6. April 2007 in FreeBSD integriert
- Mac OS X: Doch nicht :(
- ZFS auf Linux:
  - Ja, als Modul für FUSE
  - Aber nicht im Kernel
    - Bedenken von Kernel-Entw. gegenüber non-GPL-Code
    - Aber aus Sicht der CDDL: Kein Problem!
- ZFS unter GPL?
  - Jein: Die Basics zum Booten sind ja Teil von GRUB...

# Sun Storage 7000

- Open Storage NAS-System
  - > Preisgünstig wie ein x86 System
  - > Einfach zu bedienen wie ein NAS
  - > Alle Features von ACLs bis Windows
- Mit Innovativen Extras
  - > Datenintegrität mit ZFS
  - > Turbo-Lader auf SSD-Basis
  - > Allwissende DTrace Analytics
- Nur bei Sun
  - > Passend in S, M, L und XL-Größen

opensolaris

InfoWorld 2008 TECHNOLOGY OF THE YEAR AWARDS Best File System

# VIELEN DANK!

**Constantin Gonzalez**
constantin@sun.com
**blogs.sun.com/constantin**
**twitter.com/zalez**