ZFS

„Populating 128-bit file systems would exceed the quantum limits of earth-based storage. You couldn't fill a 128-bit storage pool without boiling the oceans.“

- Jeff Bonwick

GLIEDERUNG

- Geschichtlicher Überblick
- Technische Einführung
- Main Features
- Minor Features
- Nachteile und Kritik
- Zusammenfassung, Quellen

GESCHICHTLICHER ÜBERBLICK

ANFÄNGE UND MOTIVATION

- Ankündigung im September 2004, erste Version im Oktober 2005
- Früher „Zettabyte File System“, heute nur noch ZFS
- Motivation: Zukunftssicheres (für alle Zeiten) und modernes (neue Technologien) Dateisystem

ZFS

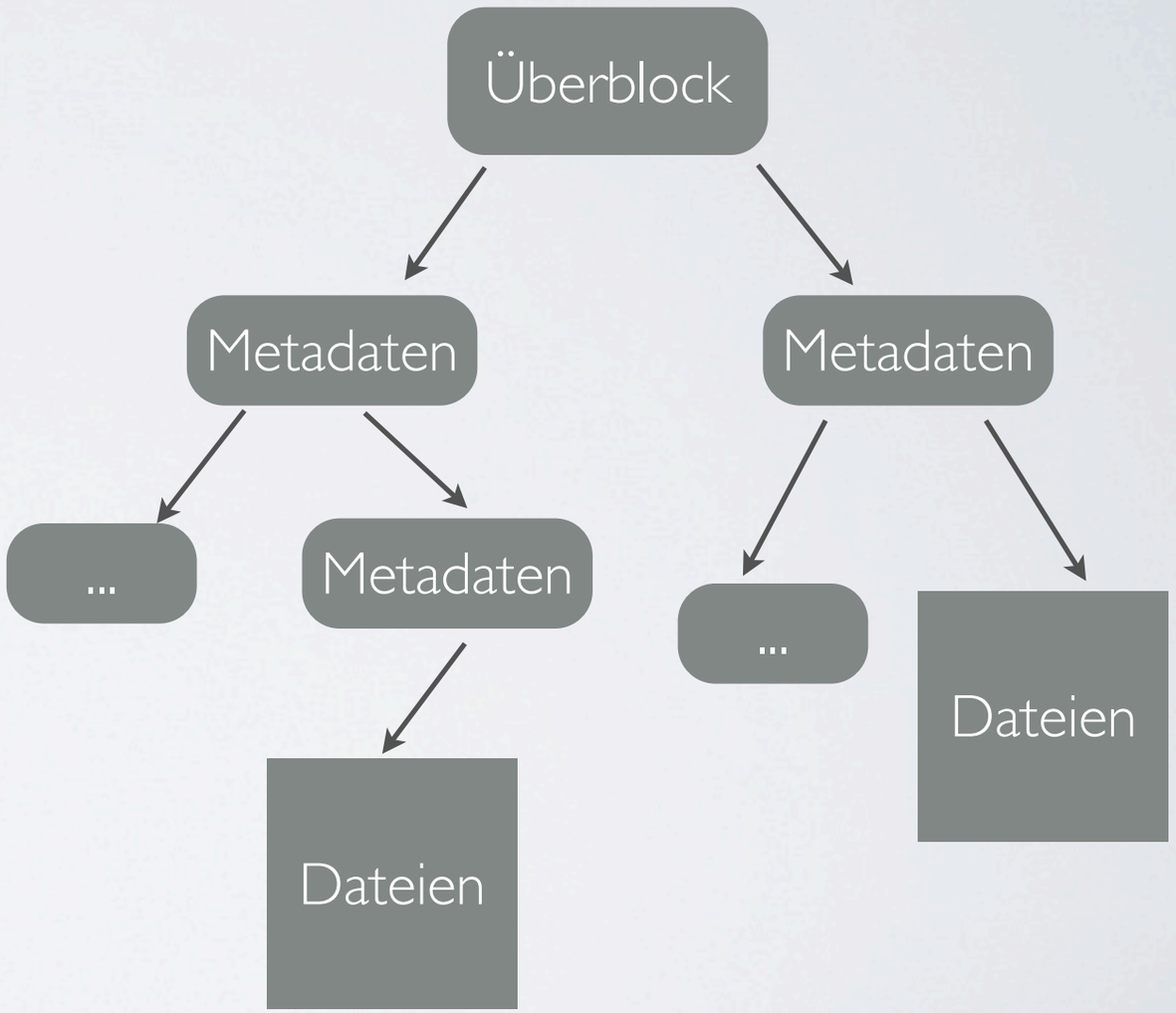
„Populating 128-bit file systems would exceed the quantum limits of earth-based storage. You couldn't fill a 128-bit storage pool without boiling the oceans.“

- Jeff Bonwick

TECHNISCHE EINFÜHRUNG

GRUNDLEGENDER AUFBAU

- Aufbau in einer Baumstruktur
- Metadaten helfen, sich zurecht zu finden
- Kein Formatieren notwendig, variable Blockgrösse

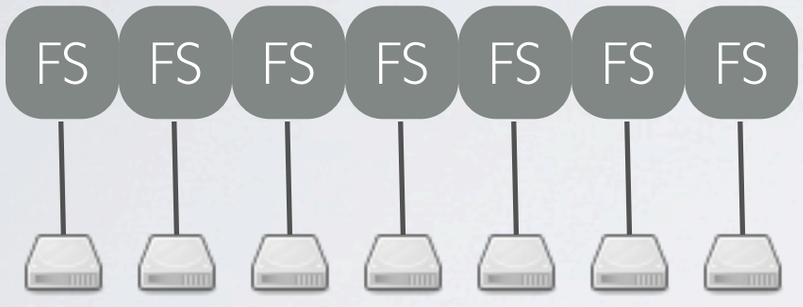


PLATTFORMEN

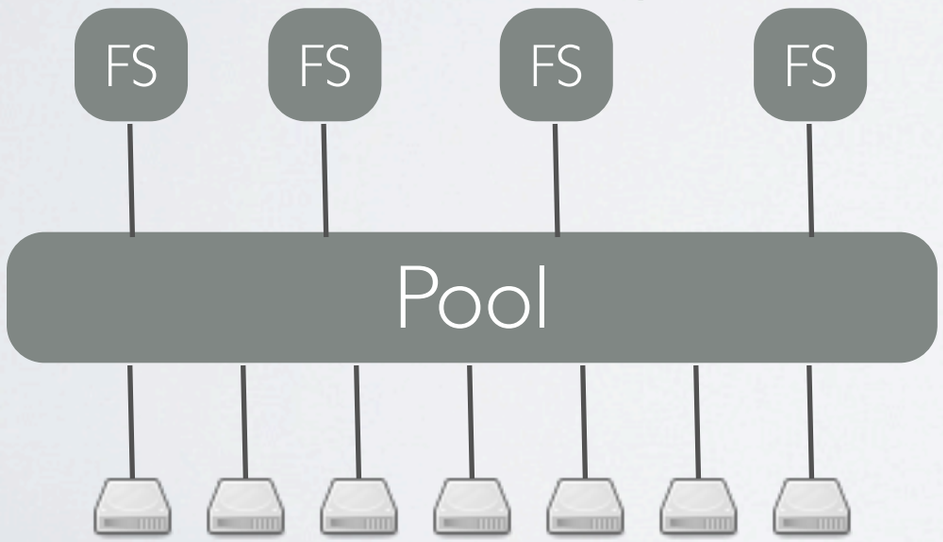
- ZFS ist Teil von OpenSolaris
- MacOS hatte Unterstützung, diese wurde wieder vollständig entfernt
- Linux anfangs nur über FUSE, inzwischen auch nativ
- FreeBSD hat bereits vollständige Unterstützung

POOLS

Herkömmliches Konzept:



Poolkonzept:



- Alle Festplatten werden zu einem Pool zusammengefasst, auf dem dann die Dateisysteme aufsetzen
- Vorteile:
  - Schneller
  - Flexibler

MAIN FEATURES

TRANSAKTIONSKONZEPT

- „Copy on Write“: Neue Daten überschreiben keine Alten
- Der alte Überblock bleibt solange erhalten, bis die Transaktion komplett abgeschlossen ist
- Ein Teil des Speichers muss dafür immer unbelegt sein

SNAPSHOTS (CLONES)

- Read-Only Kopie des Dateisystems zu einem bestimmten Zeitpunkt
- Platzsparend implementiert
- Ein Clone ist ein schreibbarer Snapshot

DATENINTEGRITÄT

- Der „Weg“ der Daten zur Festplatte ist fehleranfällig
- Herkömmliche Dateisysteme prüfen unzulänglich auf die Integrität der Daten
- ZFS prüft geschriebene Daten mit „starken“ Hashes

SELBSTREPARATUR

- ZFS erkennt kaputte Daten (wenn gelesen wird)
- Liegt ein Spiegel der Platte vor, können kaputte Daten wieder durch richtige Ersetzt werden
- „Scrub“: Alle Daten werden überprüft und gegebenenfalls repariert

MINOR FEATURES

ATTRIBUTE

- Attribute legen bestimmte Regeln und Verhaltensweisen für das System fest
- „Quota“ und „Reservierung“: Maximal- /Minimalgrösse für ein Dateisystem
- Kompression: Verschiedene Kompressionsverfahren können angewendet werden

EINFACHES BACKUP

- ZFS kann Snapshots direkt in Bitströme umwandeln
  - Daraus resultieren sehr einfache Backups
- Sogar nur mit veränderten Daten
  - Daraus resultieren sehr schnelle Backups

HYBRIDSYSTEME MIT FLASH

- Vorteile von Flashspeichern sind bekannt, allerdings sind Systeme, die nur Flashspeichern verwenden sehr teuer
- ZFS kann Hybridsysteme aus Festplatten und Flashspeichern bauen
- Vorteile: Schneller und kostengünstiger

DEDUPLICATION

- Daten, die mehrfach vorhanden sind, können erkannt werden
- Diese müssen dann, sehr einfach durch die Baumstruktur, nur ein Mal geschrieben werden

NACHTEILE UND KRITIK

- 128-Bit Adressierung ist heutzutage noch überdimensioniert
- Zusammenfassung von sonst getrennten Schichten (Dateisystem, RAID, Volume Manager)

ZUSAMMENFASSUNG UND QUELLEN

ZUSAMMENFASSUNG

- ZFS soll ein modernes Dateisystem sein, maßgeblich dafür entwickelt „auf alle Zeit“ zu reichen
- Aufbau in einer Baumstruktur: Vom Überblock über Metablöcke zu den Daten
- Wichtige Features: Poolkonzept, Transaktionskonzept, Datenintegrität, Snapshots

QUELLEN

- http://hub.opensolaris.org/bin/view/Community+Group+zfs/faq
- http://en.wikipedia.org/wiki/ZFS, vom 25.01.2011, 21:00
- http://de.wikipedia.org/wiki/ZFS, vom 25.01.2011, 21:00
- http://www.fh-wedel.de/~si/seminare/ws08/Ausarbeitung/02.zfs/funktionen.html
- http://vimeo.com/8666489
- http://zfsonlinux.org/