



Verteiltes Persistenz-System

Mykhaylo Kabalkin

01.12.2006

Übersicht



- Motivation und Problematik
- Ziel
- Anforderungen
- Systemarchitektur – erster Entwurf
- Architekturkomponenten
- Risiken

- Wo und wie werden meine Daten gespeichert?
- Sind die Daten immer verfügbar?
- Wie kann man auf alte Daten zugreifen?
- Wie sicher sind die Daten?
- Wie können unterschiedliche Dienste auf die Daten zugreifen?

Persistenz-Service System

Anforderungen



- Verfügbarkeit und Ausfallsicherheit
- Transparentes Handling von Daten
- Skalierbarkeit
- Unterschiedliche Versionen der Daten behalten
- Sicherheit

Wie beantworte ich die Anforderungen?

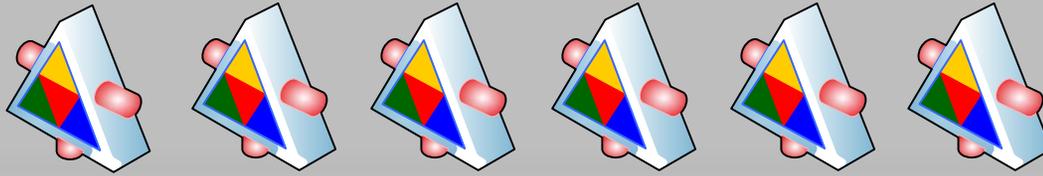
- **Verfügbarkeit und Ausfallsicherheit**
 - Redundanz von Daten
- **Skalierbarkeit**
 - Verteilung von Daten über mehrere Rechner
- **Unterschiedliche Versionen der Daten behalten**
 - Versionskontrolle
- **Sicherheit**
 - Zugriffskontrolle

Systemarchitektur



Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Persistenz-Service Schicht



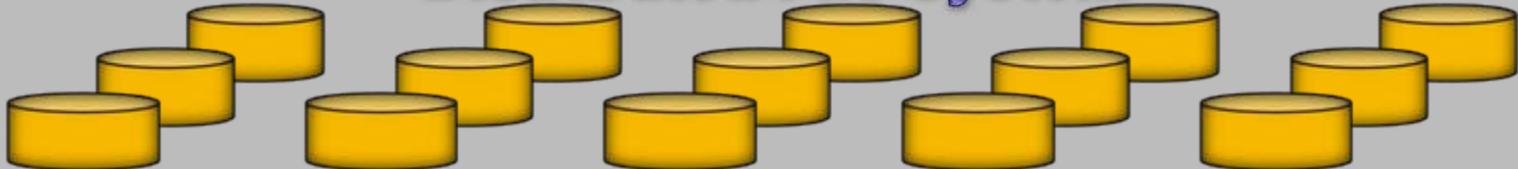
Administration

Zugriffskontrolle

Sichtbarkeit

Versionskontrolle

Distributed File System



Systemarchitektur - erster Entwurf

- RAID-Systeme dienen zur Organisation mehrerer Festplatten zu einem logischen Laufwerk
 - sind nur für lokale Speicherung gedacht
- Gewünscht ist die Verteilung über viele Rechner
 -  Über RAID sprechen wir hier nicht
 -  Es geht um Distributed File System

Distributed File System



Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

- Verteilung über viele Rechner
- Immer verfügbar
 - Sollte ein Rechner ausfallen, ist dies für Endbenutzer transparent
- Redundanz durch Replikation
- Ausfallsicher
- Trennung von Metadaten und echten Daten
- Beispiele:
 - GoogleFS
 - Andrew File System (AFS)
 - Lustre FS

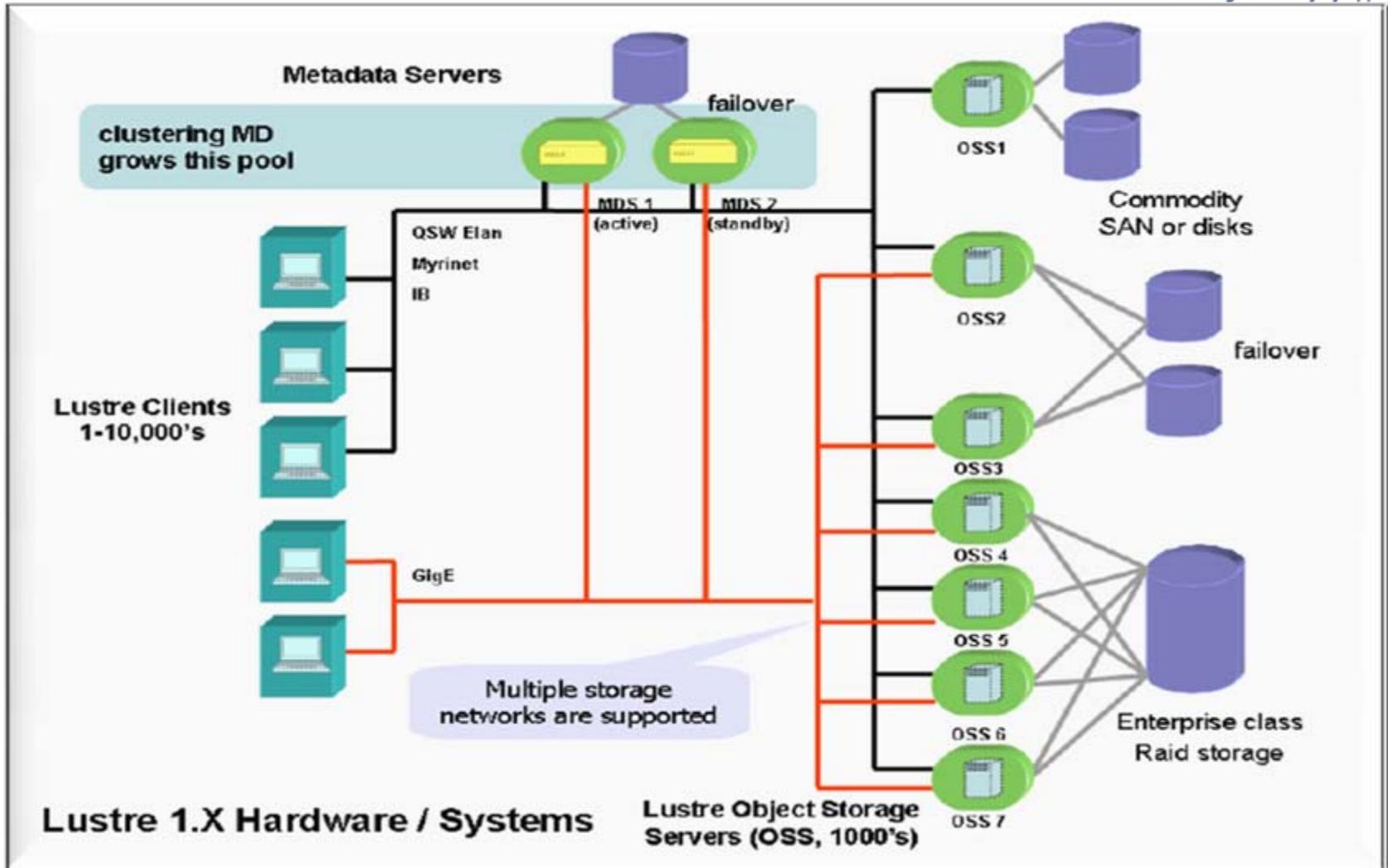
Lustre File System



- Skalierbares, sicheres, robustes, ausfallsicheres Cluster Datei System von Cluster File System Inc.¹
- Trennung von Metadaten und echten Daten
 - Object Storage Server (OSS)
 - Meta Data Server (MDS)
 - Lustre Clients
 - Interaktion mit OSSs für Daten I/O
 - Interaktion mit MDS für die Metadaten

¹ <http://www.clusterfs.com/>

Lustre File System

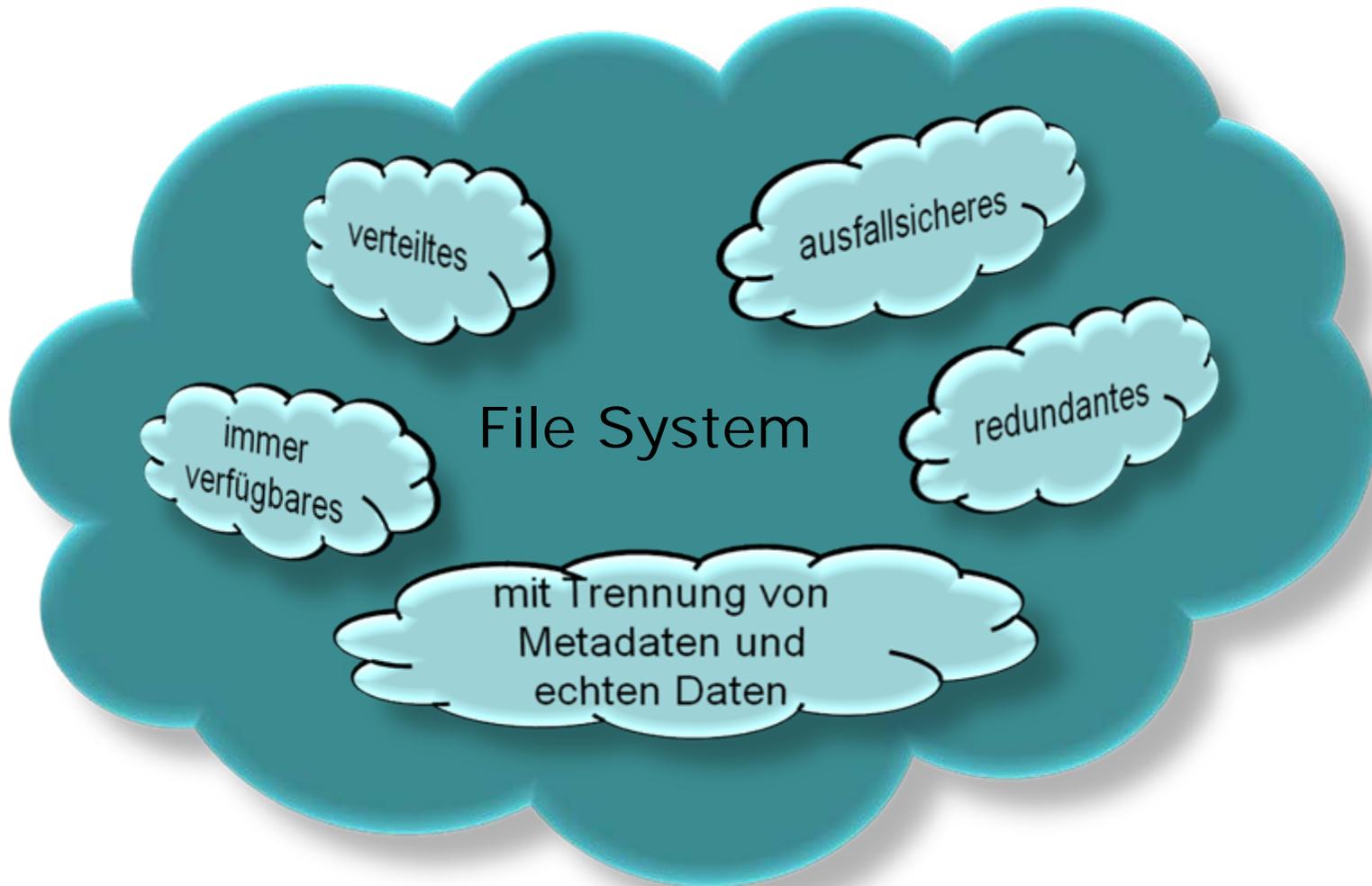


A Lustre Cluster [1]

Lustre File System



Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences



- Unterschiedliche Versionen behalten
- Zeitschlitz zurücksetzen
 - z.B. man möchte den Stand der Daten vom 01.01.2000 sehen
 - Time Machine von Mac OS X Leopard (verfügbar ab Anfang 2007) [2]

- Concurrent Versions System (CVS)
 - Ein Software-System zur Versionsverwaltung von Dateien
 - Die Benutzer müssen immer online sein
- Distributed Concurrent Versions System (DCVS)
 - Lokales Repository
 - Automatische Synchronisation gegen Main-Repository
 - Beispiele:
 - Distributed Concurrent Versions System <http://dcvs.elegosoft.com/>
 - Bazaar <http://bazaar-vcs.org/>
 - GNU Arch <http://www.gnu.org/software/gnu-arch/>
 - Git - Fast Version Control System <http://git.or.cz/>

- Kerberos als verteilter Authentifizierungsdienst [3]
 - UbiComp Projekt PDirectory – Java-Implementierung des Kerberos-Protokolls [4]
- Role Based Access Control
 - Rechtevergabe
 - Leserechte
 - Suchrechte
 - Schreibrechte
 - Neue Instanz erzeugen
 - Neue Version der Daten erzeugen
 - Löschen von Daten
- Berechtigungsstrukturen in kollaborativen Umgebungen [5]

- Für wen sind welche Daten sichtbar?
- Wie löscht man die Daten?
 - Viele Distributed File Systemen können nicht löschen
 - als gelöscht markiert
 - oder
 - keine Leserechte



- XML-basierte Schnittstelle
- Simple API
- Beliebige Dienste können angebunden werden
(z.B. Suchdienst, Abodienst, Maus Pointer)

- Versions System auf einem verteilten File System
- Komplexität der Konfiguration
- Keine Standard-Hardware (IBM Blade-Server)

IBM Blade Server



Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences



[1] Cluster File System Inc., Lustre Manual, Version 1.4.7.1-man-v35

<https://mail.clusterfs.com/wikis/lustre/LustreDocumentation?action=AttachFile&do=get&target=LustreManual35.pdf>

[2] Time Machine, Mac OS X Leopard

<http://www.apple.com/de/macosex/leopard/timemachine.html>

[3] RFC4120, The Kerberos Network Authentication Service (V5) <http://rfc.net/rfc4120.html>

[4] Projekt „PDirectory“ <http://users.informatik.haw-hamburg.de/~ubicomp/projekte/Kerberos>

[5] Horst Mund: Berechtigungsstrukturen in kollaborativen Umgebungen

<https://users.informatik.haw-hamburg.de/~ubicomp/arbeiten/diplom/mund.pdf>

[6] Cluster File System Inc., Lustre: A Scalable, High-Performance File System

<http://www.lustre.org/docs/whitepaper.pdf>

[7] Sanjay Ghemawat, Howard Gobioff, and Shun-Tak Leung: The Google File System

<http://labs.google.com/papers/gfs-sosp2003.pdf>



Vielen Dank!