2010 SourceTalk Tage

Hochperformante parallele Filesysteme für HPC

Andreas Landhäußer
T-Systems Solutions for Research GmbH
mailto:Andreas.Landhaeusser@t-systems-sfr.com

T Systems

Die Anfänge, Hardware

- IBM 350 RAMAC1956
  - Kapazität 5MB
  - Länge 1,52m (60“),
    Höhe1,73m(68“),
    Tiefe 0,74m (29“)
  - Kosten 185000 $US

Hardware heute

- Formfaktor 2,5“ (~~oder 3,5“~~) Kapazität 1-2 TB
- $\text{Accesstime} = \text{seektime} + \text{latency} + \text{controlleroverhead}$
- $\text{latency [ms]} = \frac{30000}{\text{rpms}}$
- Controller-overhead ca. 0,2 ms

Aussagekraft dieser Größen ist für die Performance eines Systems eher begrenzt, deshalb wird im professionellen Bereich die Kenngröße IOPS (Input/Output Operations per Second) verwendet.

IOPS

- Benchmarkangabe für Datenträgersysteme
- Je größer, desto schneller ist der Datenträger
- Leistung konventioneller Festplatten hängt von der Umdrehungszahl und der zugehörigen Elektronik ab.
  SATA-Festplatten erreichen nur 88 IOPS, während mit 15krpm FC-Disks 340 IOPS erreichbar sind.
  SSD (Halbleiterfestplatten) können bis zu 10000 IOPS und mehr erreichen.

Mehr IOPS?

- mehr IOPS, als die einer einzelnen Festplatte, lassen sich nur sehr schwer erreichen.
- Lösung: mehrere Festplatten/Controller/IO-Karten werden in einem Verbund benutzt.
  - höhere Leistung als mit einer einzelnen Festplatte
  - mehrere Festplatten/Controller/IO-Karten
  - >> teurer, viel ungenutzter Plattenplatz.

Redundant Array of Independent Disks (RAID)

- Erzeugung eines logischen Laufwerks aus mehreren Festplatten
  - RAID Controller
  - Software RAID

- Gebräuchliche RAID-Level
  - RAID 0 Striping = Beschleunigung ohne Redundanz
  - RAID 1 Mirroring = Spiegelung
  - RAID 5 Leistungssteigerung und Ausfallsicherheit

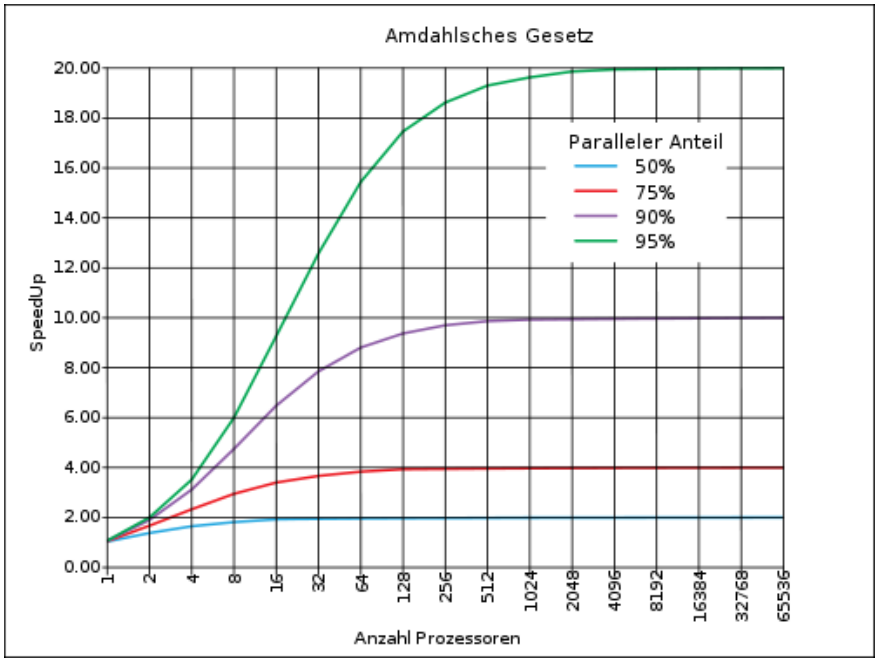
Hochleistungsrechner heute

- Cluster bestehen aus:
  - vielen voneinander unabhängigen Knoten (Nodes)
  - vielen voneinander unabhängigen Speichern
  - Hochleistungsnetzwerke zwischen den Nodes
- Codes, die Speicher und CPUs gemeinsam nutzen
- I/O soll gemeinsam auf wenige zentrale Festplatten gespeichert werden

Amdahlsches Gesetz: Speedup

- $$Speedup = \frac{1}{\sum_{k=0}^{n} \left(\frac{P_k}{S_k} \right)}$$
- P_k Prozentanteil des Codes, der beschleunigt oder verlangsamt wurde $\sum P_k = 100$
- S_k Geschwindigkeitsmultiplikator (1 = keine Veränderung)
- k laufende Nummer an der eine Veränderung durchgeführt wurde
- n Gesamtanzahl der Änderungen

Amdahlsches Gesetz



$$S = \frac{1}{(1-P) + o(N) + \frac{P}{N}} \leq \frac{1}{1-P}$$

Amdahlsches Gesetz

Beispiel:

- Ein Algorithmus, der 95% parallel auf einem 100 processing elements (PE) system läuft, wird eine aggregierte Leistung von ca. 17 PE liefern .

$$\left(\frac{1}{0.05 + \frac{1 - 0.05}{100}} \right) = 16{,}949$$

- 83 PE tragen nicht zur Gesamtleistung des Systems bei
  - Speicheralgorithmen haben selten einen Parallelisierungsgrad von mehr als 95%
  - Metadatenoperationen, die seriell sind, müssten dann z.B. auf 5% begrenzt werden
  - nur bei read-only Speichern möglich

Zugriffsmethoden auf Festplatten

- Network File System (nfs) Unix/Linux
- Common Internet File System (CIFS) Windows

Ermöglichen einen transparenten Zugriff von jedem Node auf das gemeinsame File System.

- Ohne Modifikation des Quellcodes nutzbar
- Aber: keine Abstimmung zwischen den Knoten, keine Cache-Coherence

Network attached Storage (NAS)

- Robust, da nur atomare Transaktionen

Nachteil:

- führt zu signifikanten Leistungseinbußen, da Protokoll Overhead mit steigender Netzleistung zunimmt
  - nfs Gbit Ethernet nur 60% max Leistung
  - CIFS unter 50%

Storage Area Network (SAN)

- Leistungssteigerung und Vereinfachung des Managements über zentrale Speicher-Dienste
- Hochgeschwindigkeits Netzwerk zwischen SAN und Node
- Alle Knoten können auf das SAN zugreifen, aber ein gemeinsamer Zugriff ist nicht vorgesehen.

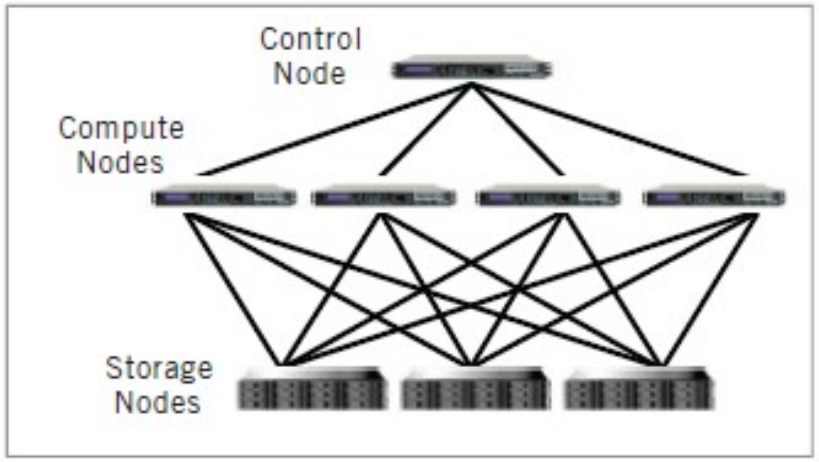
Erwartungen an Globale File Systeme

- Einheitlicher Zugriff: Global Name Space
- verschiedene Locking Schemata: File-, Block-, Byterange-level
- Sicherheit: global authentication/authorization
- Zuverlässigkeit: Eliminierung von Single point of Failure (SPOF)
- Verfügbarkeit: Routine Wartung kann ohne Unterbrechung des Nutzerbetriebes stattfinden
- Skalierbarkeit: bis zu >> Terabyte von Daten
- Standards: IEEE Posix konform
- Leistung: High Performance

Parallele Filesysteme

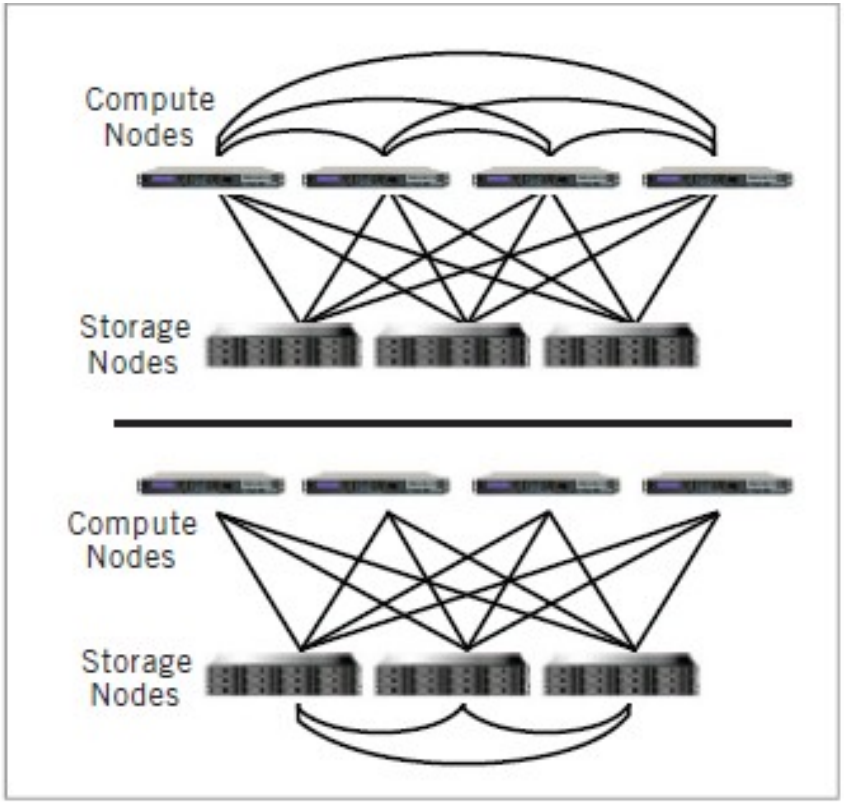
Funktionweise:

Zentraler Kontrollknoten



Parallele Filesysteme

Verteilte Kontrolle:



Parallele Filesysteme

- Lustre
- GPFS
- pNFS
- FhGFS
- Ceph
- hadoop/HDFS
- ...

Lustre

- massiv paralleles Filesystem,
- 1998 an der CMU von Peter Braam vorgeschlagen,
- von CFS entwickelt,
- 2007 von Sun gekauft,
- 2009/2010 wurde Sun von Oracle erworben

Lustre: Kenndaten

- Single Name Space
- Skalierbarkeit: mehr Storage Server
- Performance: high Performance

Lustre: Bewertung

Vorteile:

- Skalierbar für 10000de Klienten, große Nutzerbasis
- High Performance
- Posix-Schnittstelle
- Stabilität

Nachteile:

- zentraler Metadatenserver, HA-Lösung möglich
- schwierige Administration
- wenig Zugriffssicherheit
- ungewisse Zukunft

GPFS

- Entwickelt von IBM Anfang der '90er als Multi-Mediafilesystem.
- 1998 wurde dann daraus GPFS (General Parallel/Purpose Filesystem)
- Weiterentwickelt bis heute.

GPFS: Kenngrößen

- Single Name Space
- Posix-Schnittstelle + non-Posix
- Skalierbarkeit: mehr Server, mehr JBODS
- Performance: high Performance
- Locking Schemata: File-, Block-, Byterange-level
- Zuverlässigkeit: Filereplication, Snapshots
- Administration: einfach während des Betriebs, running updates

GPFS: Bewertung

Vorteile

- Stabilität
- High Performance
- Skalierbar für 10000de Klienten, große Kundenbasis
- Posix-Schnittstelle
- verteilte Metadaten
- Einfache Administration

Nachteile

- kommerzielle Software
- Nur auf Linux, Windows und AIX portiert

pNFS

- NFS V1 – V4 sind nicht für Cluster geeignet
- pNFS ist eine Erweiterung von NFS V4 aka NFS V4.1
- Standardisierung erfolgte im Januar 2010

pNFS: Kenngrößen

- Separation von Kontroll- und Datenverkehrs
- NFS Server muss nicht mehr auch den Festplattenspeicher bereitstellen
- NFS Filesystem kann auf mehrere Speichersysteme verteilt weden
- Files können im Client, ähnlich wie bei den Block-Volume-Managern, repliziert oder auf mehrere Speichersysteme verteilt werden

pNFS: Bewertung

Vorteile

- nahtlose Integration ohne Änderungen
- abwärts kompatibel

Nachteile

- keine cache Kohärenz,
- Mehrere Metadatenserver, aber nur Kopien des Metadaten Bestands

FhGFS

- Entwickelt 2006 vom Institut ITWM an der FhG in Kaiserslautern
- Paralleles Filesystem für die HPC-Community
- FhGFS wird von FhG weiterentwickelt

FhGFS: Kenngrößen

- Skalierbar
  - verteiltes Metadaten Management
  - interne Datensplitting und -Verteilung
- Flexibel
  - Metadaten, Storage und Clients können im laufenden Betrieb hinzugefügt werden
  - Server und Clients können auf demselben System laufen
  - Unterstützt, Ethernet aunf Infiniband, automatisches Failover
- Einfach zu nutzen
  - Installation ohne Kernel-Patches
  - Graphisches Administrationsinterface
    - Automatische Clusterinstallation
    - Überwachung aller lebenswichigen Funktionen und Performance Anzeige

FhGFS: Bewertung

- Vorteile
  - Freie Benutzung
  - Support durch Fraunhofer Gesellschaft
- Nachteile
  - kein HA
  - kein native Backup
  - zur Zeit nur Linux unterstützt

Ceph

- Ceph ist ein neuentwickeltes paralleles Filesystem:
  Originalzitat:

"Ceph is designed to handle workloads in which tens thousands of clients or more simultaneously access the samefile or write to the same directory – usage scenarios that bring typical enterprise storage systems to their knees."

Ceph: Kenngrößen

- Nahtlose Skalierung, Ceph verteilt automatisch die Daten auf neue Storageeinheiten
- Alle Daten werden automatisch auf verschiedene Speichereinheiten repliziert, schnelle Wiederherstellung nach einem Fehler
- Adaptiver Metadatenserver, dynamische Anpassung an die aktuelle Workload
- Pseudo-random data distribution function (CRUSH)
- Reliable object storage service (RADOS)
- Extent B-tree object File System (today btrfs)

Ceph: Bewertung

Vorteile:

- Wissenschaftliche Anwendungen, hoher Durchsatz
- extrem hohe Read/Write Leistungen
- fortschrittlichste verteilte Filesystem

Nachteile:

- jung (erst ab Kernel 2.6.34)
- nur Linux
- sehr komplex
- Funktionalitäten noch nicht ernsthaft getestet

HDFS: hadoop

- HDFS ist Teil des Apache hadoop Projekts zur Entwicklung von zuverlässiger, skalierbarer und verteilter Anwendungen.
- Hadoop wurde durch Google's MapReduce und das Google Filesystem inspiriert

hadoop: Kenndaten

- Single name space
- Zuverlässigkeit: rw replication, re-balancing, Kopie auf verschiedenen Servern
- Skalierbarkeit: mehr Server
- Performance: „grid“ computation, parallel transfer

hadoop/HDFS: Bewertung

Vorteile:

- Basis für GRID Infrastruktur
- verteilte Daten (hoher Durchsatz)
- Archivierung
- heterogener Einsatz

Nachteile:

- kein general purpose Filesystem
- keine Posix-Schnittstelle
- wenig Zugriffssicherheit
- Java

Literaturangaben

- Amdahl's Law 1967; Wikipedia
  http://en.wikipedia.org/wiki/Amdahl's_law
- Sebastian Breier; Uni Stuttgart: Verteilte Dateisysteme I: GPFS, CXFS
  http://www.ipvs.uni-stuttgart.de/abteilungen/sgs/lehre/lehrveranstaltungen/hauptseminar
- Kurt Gerecke; IBM Stuttgart: IBM System Storage-Kompendium 1952-2010
  http://www.ibm.com/systems/data/flash/de/resources/ibm_storage_compendium_de.pd
- Frank Kraemer; IBM Mainz: GPFS – The Cloud File System 23.2.2010
- Roland Laifer; Uni Karlsruhe: The Parallel File System HP SFS /Lustre on xc2
  http://cms.uni-karlsruhe.de/rz/docs/Lustre/rznews_sept2005.pdf
- Fabrizio Manfred; Zeropiu Furuholmen: Use Distributed File system as a Storage Tier
  http://www.netways.de/uploads/media/Fabrizio_Manfred_Use_Distributed_Filesystem_a
- Fraunhofer Parallel Filesystem FhGFS
  http://www.fhgfs.com/docs/FraunhoferFS_Flyer.pdf