



**Fachhochschule
Bonn-Rhein-Sieg**
University of Applied Sciences

Fachbereich Informatik
Department of Computer Science



Lustre – ein High-Performance- Dateisystem

von
Martin Pelzer
und
Hanno Tersteegen

Seminararbeit
zur Vorlesung
„Parellele Systeme“
gehalten von
Prof. Dr. Rudolf Berrendorf
im Sommersemester 2005



Inhaltsverzeichnis

1 Einleitung	3
1.1 Lizenzmodell.....	3
2 Anforderungen	4
3 Architektur von Lustre	6
3.1 Object Storage Targets (OST).....	7
3.2 Metadata Servers (MDS).....	8
3.3 Lustre Clients.....	10
3.4 Netzwerkunabhängigkeit.....	10
3.5 Beispiel.....	11
4 Verbreitung	13
5 Ausblick	14



1 Einleitung

Lustre ist ein verteiltes Dateisystem, das als besondere Vorteile seine Skalierbarkeit sowie seine hohe Performanz betont. Hergestellt wird Lustre von der Firma Cluster File Systems, Inc., von der es auch vertrieben wird. Lustre ist als Open Source unter der GPL erhältlich. Der Name Lustre setzt sich aus den Wörtern „Linux“ und „Clusters“ zusammen.

Diese Seminararbeit stellt Lustre vor und erläutert seine Architektur und Funktionsweise. Auch werden einige Konkurrenzprodukte kurz vorgestellt. Einen wichtigen Teil der Arbeit stellt auch das Kapitel „Ausblick“ dar, da die Entwicklung von Lustre längst nicht abgeschlossen ist und in der Roadmap noch einige interessante Features für die Zukunft geplant sind.

Lustre basiert in weiten Teilen auf offenen Standards. So setzt das Dateisystem beispielsweise auf Linux auf, benutzt XML zur Speicherung seiner Konfigurationsdaten, verwendet einen LDAP-Server und bietet eine SNMP-Schnittstelle. [white].

1.1 Lizenzmodell

Dieser Abschnitt soll kurz die Besonderheiten des Lizenzmodells von Lustre erläutern. Die Aussage, Lustre stehe unter der GPL ist nicht völlig korrekt. Lustre wird von der Firma Cluster File Systems [CFS] hergestellt und vertrieben. Dabei benutzt das Unternehmen zwei verschiedene Vertriebswege. Die aktuellste Version von Lustre ist Closed Source und kann direkt von Cluster File Systems bezogen werden. Etwas ältere Versionen von Lustre werden regelmäßig als Open-Source-Software herausgegeben und über eine eigene Webseite verbreitet [Lustre]. Diese Versionen stehen dann unter der GPL. Nach eigenen Angaben gibt der Hersteller neue Versionen ungefähr ein Jahr nach Verfügbarkeit als Open-Source-Variante heraus. Die Vergangenheit zeigt, dass dies auch schneller geschehen kann. So wurde die aktuell als Open Source verfügbare Version 1.2.4, die ab Juli 2004 vertrieben wurde, bereits Ende Februar 2005 unter der GPL veröffentlicht.



2 Anforderungen

In diesem Kapitel wird erläutert, welche Voraussetzungen erfüllt sein müssen, damit auf einem System Lustre eingesetzt werden kann. Die Ausführungen dieses Kapitels basieren sowohl auf [HowTo] als auch auf Informationen aus Readme-Dateien in der aktuell frei verfügbaren Lustre-Version 1.2.4.

Lustre läuft, wie die Erklärung des Namens bereits nahelegt, nur auf Linux-Systemen. Um Lustre verwenden zu können, muss der Kernel neu kompiliert werden. Die zur Zeit frei verfügbare Lustre-Version 1.2.4 bringt Kernel-Patches für

- Kernel 2.4.21 für Red Hat Enterprise sowie
- Kernel 2.6.5 für SuSE Enterprise Server

mit. Für beide Kernel stehen Patches sowohl für die 32- als auch die 64-Bit-Kernelversionen bereit – sowohl für die Intel-Itanium-64-Bit-Architektur „ia64“ wie auch für die Standard-64-Bit-Architektur „x86-64“. Zu allen diesen Patches gibt es auch noch eine Variante mit SMP-Unterstützung.

Neben der Möglichkeit, die Kernel-Patches zu verwenden, besteht natürlich die Möglichkeit, den Kernel komplett selbst anzupassen. Dadurch besteht theoretisch die Möglichkeit, Lustre auf jedem Linux-Kernel lauffähig zu machen. Eine Garantie gibt der Hersteller hierfür nicht. Auf der Lustre-Webseite wird von Tests auf Alpha- und PowerPC-Systemen berichtet. Nähere Angaben werden hierzu allerdings nicht geliefert.

Nach der Einrichtung des Kernels bestehen zur Installation von Lustre noch folgende Abhängigkeiten zu weiteren Softwarepaketen:

- readline,
- libxm12,
- Python und
- PyXML.

Diese Pakete sollten auf den meisten Systemen bereits standardmäßig installiert sein. Eine weitere Abhängigkeit besteht zu der Message Passing API „Portals“ der Sandia National Laboratories [Sandia]. Diese Software stellt eine Netzwerkabstraktionsschicht zur Verfügung, die Lustre verwendet, um mehrere Netzwerktypen miteinander kombinieren zu können.



„Portals“ wird allerdings mit Lustre mitgeliefert und muss daher nicht separat installiert werden. Einige tiefere Funktionalitäten in Lustre benötigen neben den hier angegebenen noch weitere Softwarepakete.

Die aktuelle Version 1.4 von Lustre, die noch nicht unter der GPL verfügbar ist, unterstützt laut Herstellerangaben nun auch PowerPC-Architekturen und läuft einhergehend damit auch unter Mac OS X.

3 Architektur von Lustre

In diesem Kapitel soll die Struktur eines Lustre-Systems aufgezeigt sowie die einzelnen Komponenten erklärt werden. Folgende Abbildung zeigt die verschiedenen Komponenten in einem solchen System:

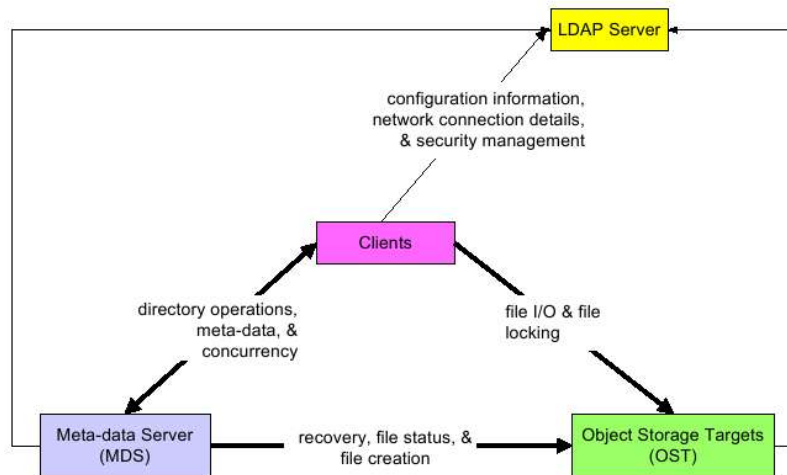


Abbildung 1: Aufbau eines Lustre-Systems [white]

In den in der Abbildung grün dargestellten Object Storage Targets (OST) werden die Dateien abgespeichert. Die blau dargestellten Metadata Server (MDS) verwalten die Daten, d.h. sie nehmen alle Anfragen wie das Anlegen neuer Dateien, das Auslesen einer Datei, das Verändern von Attributen einer Datei oder eines Verzeichnisses etc. entgegen und leiten sie an die entsprechenden OSTs weiter. Die Metadata Server halten die komplette Struktur der auf den OSTs gespeicherten Daten vor. Ein Client (in der Abb. lila dargestellt) ist ein Anwendungsrechner, auf dem ein Programm läuft, das die Daten in dem Lustre-Dateisystem verwendet. Ein Client hat also das Lustre-Dateisystem gemountet. Der LDAP-Server schließlich dient lediglich Namensdiensten. So wendet sich ein Client z.B. an den LDAP-Server um zu erfahren, wo ein MDS ist, an den er sich wenden kann.

Folgende Abbildung zeigt, aus welchen Bestandteilen die einzelnen Komponenten eines Lustre-Systems bestehen.

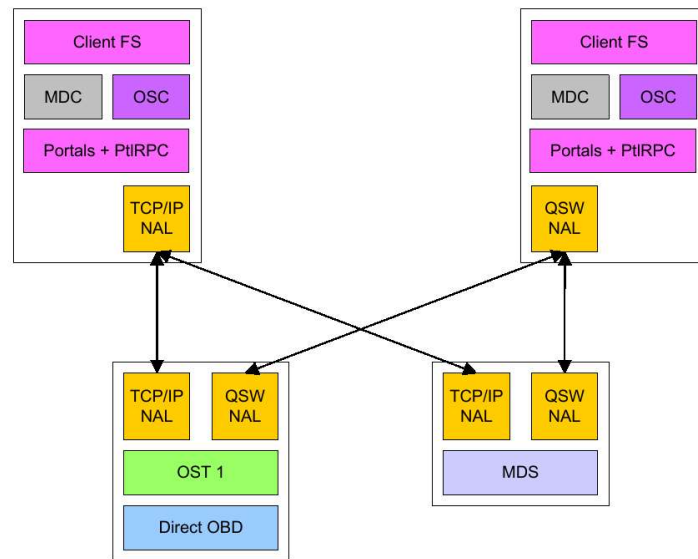


Abbildung 2: Bestandteile der einzelnen Komponenten eines Lustre-Systems [lustre-usr-1]

Die beiden oberen Komponenten sind Clients, unten links befindet sich ein OST und unten rechts ein MDS. Jede Komponente ist in verschiedene Module unterteilt. So besitzt z.B. jede Komponente ein Kommunikationsmodul (in der Abb. orange dargestellt). Die weiteren Bestandteile dieser Komponenten werden in den folgenden Unterkapiteln erläutert.

3.1 Object Storage Targets (OST)

Wie bereits erwähnt, werden in den OSTs alle Dateien eines Lustre-Systems gespeichert. Die OSTs selber stellen dabei eine Schnittstelle zwischen den anfragenden Instanzen (den Clients) und dem eigentlichen physikalischen Speicher dar. Dieser wird als Object Based Disks (OBD) bezeichnet. Die Kommunikation zwischen OST und OBD geschieht über einen Treiber. Daher kann eine OBD im Prinzip ein beliebiges Speichermedium mit beliebigem Dateisystem sein, solange ein entsprechender Treiber zur Verfügung steht. Dadurch ist es z.B. möglich, eine Festplatte, die als normale Linuxfestplatte mit Dateisystem ext3, ReiserFS, JFS oder XFS formatiert ist, als OBD in ein Lustre-System einzubinden. Die gerade erwähnten Dateisysteme werden standardmäßig von Lustre unterstützt, d.h. für sie werden Treiber mitgeliefert. Durch die Abstraktion über die OSTs sind die eigentlichen Dateisysteme und Speichermedien für die Lustre-Clients versteckt. Diese müssen nur mit den OSTs kommunizieren können. Die eigentliche Speicherung in einem konkreten Dateisystem ist für sie nicht von Belang (vgl. Abbildung 3).

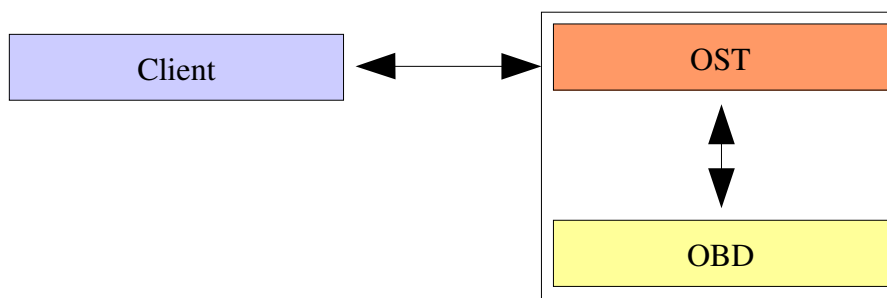


Abbildung 3: Clients kommunizieren immer mit OSTs, nie direkt mit OBDs

3.2 Metadata Servers (MDS)

Metadaten sind Daten über Daten, zum Beispiel die Dateistruktur, die Attribute einer Datei, ein Link auf eine Datei und einiges mehr. Alle journalisierenden Dateisysteme halten in ihrem Journal Metadaten über die gespeicherten Daten vor. Änderungen an den Daten werden erst an das Journal geleitet und dort abgespeichert. Die tatsächliche Änderung der Daten im Dateisystem kann asynchron zu einem späteren Zeitpunkt erfolgen. Durch die Speicherung aller Änderungen am Dateisystem kann die Konsistenz auch gewährleistet werden, wenn im Dateisystem Fehler auftreten. Dazu werden die im Journal gespeicherten Änderungen einfach noch einmal ausgeführt.

Das Journal liegt bei Lustre auf den sogenannten Metadata Servern, die ausschließlich diesem Zweck dienen (d.h. auf ihnen liegen keine Dateien des Dateisystems). Die eigentlichen Daten liegen also physikalisch an einem anderen Ort, was die Ausfallsicherheit durch den eben erwähnten Wiederherstellungsmechanismus abermals erhöht.

Wie bei vielen anderen Dateisystemen gibt es auch bei Lustre für jede Datei, jedes Verzeichnis etc. eine eindeutige Inode. Allerdings ist in dieser Inode nicht die Referenz direkt zu den Daten gespeichert, sondern eine Referenz auf ein Objekt, in dem die Daten gespeichert sind. Alle Inodes werden auf den MDSs verwaltet. Die Objekte, auf die die Inodes zeigen, liegen auf den OSTs (genauer gesagt: auf den OBDs; auf diese wird aber immer über die OSTs zugegriffen).

Wird eine Datei neu angelegt, so nimmt der Client Kontakt mit dem MDS auf, der die Inode für die Datei anlegt und seinerseits Kontakt zum OST aufnimmt, damit dieser ein Objekt anlegen kann, welches die Dateiinhalte speichert. Die Metadaten der Datei werden in der



Inode als erweiternde Attribute abgelegt. Die Daten selber werden durch das OST auf die darunter liegenden OBDs geschrieben (s.o.).

Ist eine Datei erst einmal angelegt, so können die Clients direkt mit den OSTs, welche die Lese- und Schreiboperationen an die darunter liegende OBD weiterleiten, kommunizieren.

Nur bei Veränderung der Dateiattribute oder Verschieben bzw. Löschen der Datei werden die MDSs angesprochen, die diese Informationen in ihrem Journal abspeichern und entsprechende Anpassungsbefehle an die OSTs versenden.

Neben der Abstraktion, die von den OSTs durchgeführt wird, unterstützen diese auch ein flexibles Modell, um neue Speicher (storage) zu einem existierenden Lustre-System hinzuzufügen. OSTs können problemlos in den laufenden Betrieb integriert werden. Analog dazu ist es auch möglich, weitere OBDs einfach dem Pool von Speichern (storage) hinzuzufügen und die neue OBD einem OST zuzuordnen.

In einem Lustre-System kann es mehrere MDSs geben, was mehrere Gründe haben kann. Zum Einen kann ein einzelner MDS schnell zu einem Flaschenhals im System werden, da alle Clients ihre Anfragen an einen einzelnen Rechner richten. Betrachtet man die Zielvorstellungen des Herstellers mit mehreren 10.000 Knoten, wird dieser Flaschenhals deutlich. Des Weiteren wird durch die mehreren MDSs die Ausfallsicherheit des Systems erhöht. Sollte ein MDS ausfallen, so kann ein Client durch eine Anfrage an den LDAP-Server die Adresse bzw. den Namen eines neuen MDS erfahren und seine Anfragen an diesen senden (Abbildung 4 verdeutlicht diesen Vorgang). [white]

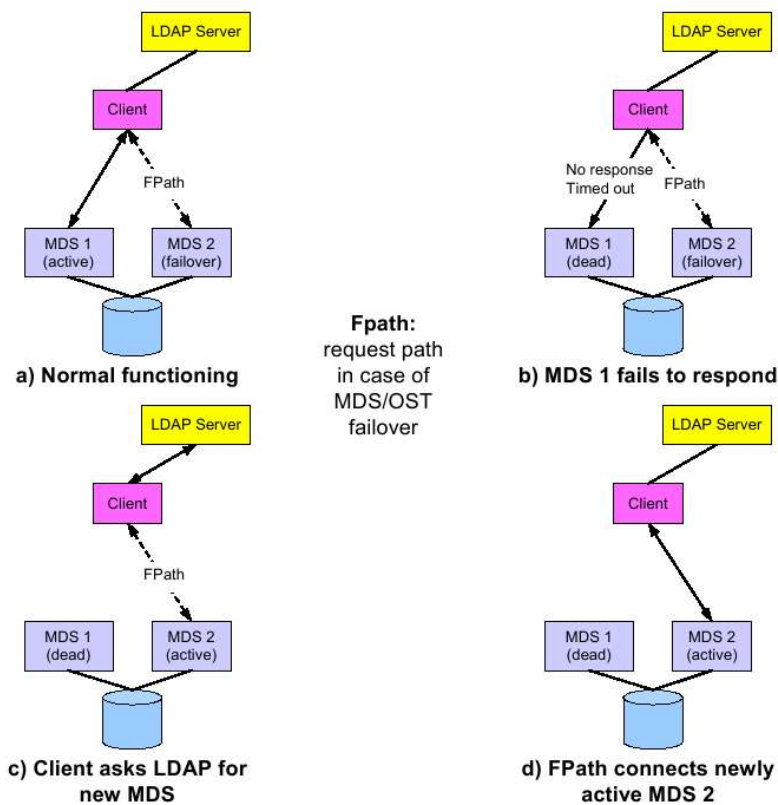


Abbildung 4: Fällt ein MDS aus, kann ein Client durch eine Anfrage an den LDAP-Server einen weiteren MDS in Erfahrung bringen und dann mit diesem kommunizieren [white]

3.3 Lustre Clients

Ein Client ist im Lustre-Kontext ein Computer, der die Dienste des Lustre-Systems in Anspruch nimmt, d.h. das Lustre-Dateisystem eingebunden hat und darauf zugreift, um Daten zu lesen oder zu schreiben. Lustre-Clients sind also die Rechner, die Anfragen an die MDSs stellen und Daten auf die OSTs schreiben bzw. von diesen lesen. Damit ein Rechner auf ein Lustre-System zugreifen kann, muss Lustre auf ihm installiert sein.

3.4 Netzwerkunabhängigkeit

Ebenso wie Lustre von den eigentlichen Dateisystemen durch die OSTs abstrahiert, abstrahiert es auch von den verwendeten Netzwerken. Dazu wird eine quelloffene Netzwerkabstraktionsschicht namens „Portals“ verwendet. Diese bietet neben der reinen Abstraktion auch noch eine Unterstützung für heterogene Netzwerke an. Dadurch wird es möglich, in einem Lustre-System verschiedene Netzwerktypen zu verwenden. Beispielsweise



kann man die MDSs über einen anderen Netzwerktyp an die Clients anbinden als die OSTs, was aufgrund der unterschiedlichen Aufgaben und damit zusammenhängenden Netzlast durchaus sinnvoll sein kann. [white]

Zur Zeit unterstützt Lustre neben Ethernet noch Quadrics-Netze. Hewlett Packard fügte für seine Lösung „HP Storage Works Scalable File Share“ [hp], welche Lustre verwendet, noch Unterstützung für Myrinet-Netze hinzu. Unterstützung von FibreChannel und InfiniBand ist geplant und noch für das Jahr 2005 angekündigt.

3.5 Beispiel

Dieses Kapitel zeigt den Aufbau eines wirklichen Lustre-Systems als exemplarische Darstellung, wie Lustre letztendlich verwendet wird. Folgende Abbildung zeigt einen beispielhaften Aufbau eines Lustre-Systems.

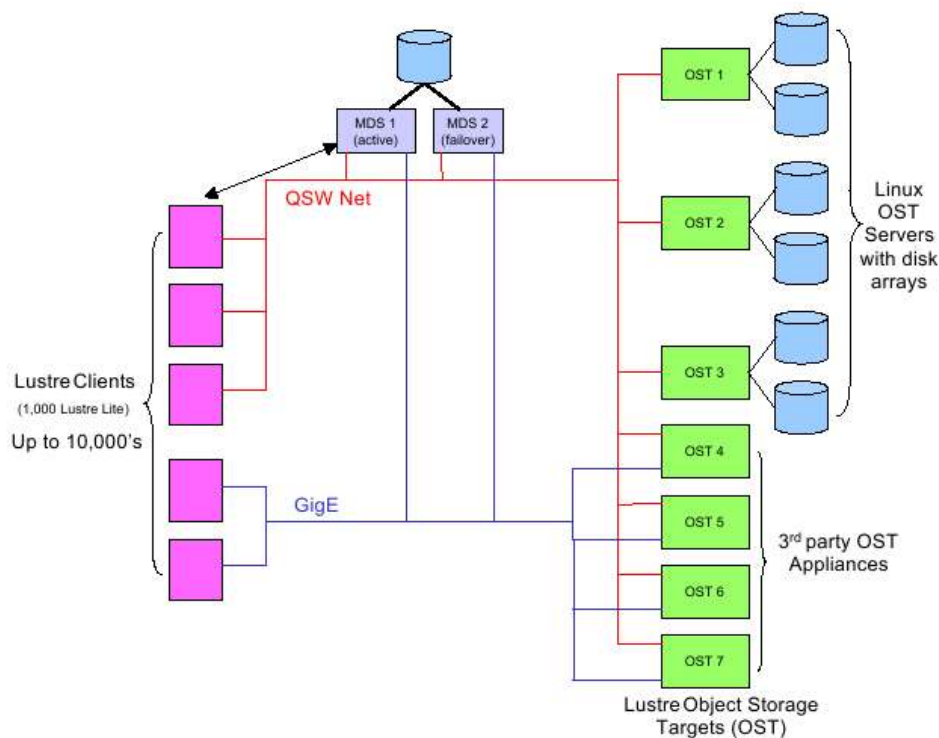


Abbildung 5: Beispiel eines Lustre-Systems [white]

Das Dateisystem erstreckt sich über die ganze rechte Seite (grün), auf der die OSTs, teilweise mit den dahinterliegenden OBDs (blau) dargestellt sind, und auch über den oberen Teil der Abbildung, der die MDSs, in diesem System zwei Stück, die auf eine gemeinsame Datenbasis zugreifen, enthält. Auf der linken Seite sind eine Reihe Clients dargestellt.



Man sieht, dass die Clients wie auch die MDSs nur mit den OSTs und niemals mit den OBDs direkt kommunizieren. Die Clients müssen aber nicht immer den Umweg über die MDSs gehen, sondern können, wie anhand der Netzwerkverbindungen ersichtlich ist, auch direkt mit den OSTs kommunizieren.

Das dargestellte System ist mit zwei verschiedenen Netzwerktypen vernetzt (Gigabit Ethernet - blau, und Quadrics - rot). Die oberen drei Clients kommunizieren über Quadrics mit den MDSs und allen OSTs, während die unteren beiden Clients, die über Gigabit Ethernet angebunden sind, neben den MDSs nur mit den unteren drei OSTs kommunizieren können. Die einzelnen Komponenten kommunizieren untereinander immer auf die gleiche Weise, d.h. die MDSs müssen nicht unterscheiden, ob sie gerade etwas über Ethernet oder Quadrics versenden. Grund hierfür ist die weiter oben beschriebene Netzwerkabstraktion, die auch heterogene Netzwerke unterstützt.



4 Verbreitung

Im Jahr 2003 liefen drei der acht schnellsten Supercomputer der Welt mit einem Linux-System. Alle diese Systeme verwendeten Lustre als Dateisystem [ols2003]. In der aktuellen Top-500-Liste der schnellsten Computer der Welt von November 2004 findet man beispielsweise das System „Thunder“, das Lustre als Dateisystem verwendet. „Thunder“ ist ein aus 4096 Itanium2-Prozessoren bestehendes Cluster, welches knapp 20 TeraFLOPS Performance (Rmax) liefert und in der aktuellen Liste auf Platz fünf steht. „Thunder“ steht in den USA im „Lawrence Livermore National Laboratory (LLNL)“, welches zum US-Energieministerium gehört [top500]. Nach Aussage von Hewlett Packard (s.u.) ist dieses Ministerium mit an der Entwicklung von Lustre beteiligt.

Neben Anwendungen in massiv parallelen Rechnern der Spitzenklasse wird Lustre aber auch in kleineren, etwas gängigeren System verwendet. So wird Lustre von Hewlett Packard in einem eigenen Produkt als Dateisystem verwendet. HP bietet eine Technologie namens „HP StorageWorks Scalable File Share“, kurz SFS, an. Dies ist ein fertiger Dateiserver, bestehend aus mehreren Festplatten, die sich in OSTs und MDSs aufteilen. Ein solcher Server kann bis zu 2 Milliarden Dateien verwalten (in den MDSs) und hat z.Zt. eine maximale Kapazität von 4,7 TB. Die Anbindung kann über Gigabit Ethernet, Quadrics oder Myrinet (vgl. Kapitel Anforderungen) erfolgen. Eine Unterstützung von InfiniBand ist in Zusammenarbeit mit Cluster File Systems noch im Jahr 2005 geplant.



5 Ausblick

Das Lustre Whitepaper [white] gibt einen ausführlichen Ausblick auf die geplante zukünftige Entwicklung des Lustre-Projekts. Die geplanten Erweiterungen sollen in diesem Kapitel vorgestellt werden.

– Lustre Global Namespace:

Aus Sicht des Endnutzers hat ein verteiltes Dateisystem den Vorteil, dass der Nutzer Zugang zu einer größeren Menge an Speicher hat, als es ihm normalerweise auf einem lokalen System möglich wäre. Um jedoch von verschiedenen Systemen auf die Daten zugreifen zu können, ist es sinnvoll eine einheitliche Zugangsmöglichkeit zu den Daten zu besitzen. Der klassische Ansatz, der auch von Lustre verfolgt wird, ist der globale Namensraum. Der globale Namensraum ist typischerweise ein einzelnes Verzeichnis durch das dem Nutzer der Zugriff auf das verteilte Dateisystem möglich gemacht wird (auch bekannt als „mounting“). Der Nutzer sieht so gar nicht mehr, dass das Dateisystem verteilt ist, da ihm nur die Daten als Ganzes gezeigt werden. Dementsprechend muss der Nutzer sich auch nicht darum kümmern, wo genau die Daten liegen, die er verwenden möchte. Die Lokalisierung übernimmt das Dateisystem.

– Performanzsteigerung:

Die Entwickler von Lustre arbeiten neben Erweiterungen durch neue Features auch immer an einer Steigerung der Performance von Lustre. Dabei geht es zum Einen um die Netzwerkperformance, aber auch um die maximale Anzahl von Knoten, die das Lustre-Dateisystem unterstützt. In den letzten Jahren wurde diese Zahl auf momentan mehrere tausend Knoten erhöht. Die Ziele der Entwickler sind allerdings wesentlich ehrgeiziger, so dass die Steigerung weiter gehen wird. Wirft man einen Blick auf momentane und zukünftige Supercomputerprojekte, so stellt man fest, dass die Anzahl der Prozessoren (und damit die Anzahl der Knoten in einem verteilten Dateisystem auf einem solchen Computer) in einem Supercomputer immer weiter wächst. Dies rechtfertigt die Bestrebungen der Lustre-Entwickler nach weiterer Steigerung in diesem Bereich.

– Mehr Sicherheit:

Dateisystemsicherheit ist ein sehr wichtiger Aspekt von verteilten Dateisystemen. Die Standard Sachziele sind Authentifizierung, Authentisierung und Verschlüsselung. Während Storage Area Networks (SAN) oft nicht geschützt sind, soll Lustre mit Secure



Network Attached Disk (NASD) derartige Funktionen ermöglichen.

Lustre integriert keinen spezifischen Authentifizierungsdienst, sondern kann zusammen mit bestehenden Mechanismen mittels des Generic Security Service Application Programming Interface (GSS-API) interagieren. GSS-API ist ein offener Standard, der sichere Sitzungskommunikation ermöglicht und dabei die Sachziele Authentication, Datenintegrität und Data confidentiality abdeckt. Lustre authentication wird Kerberos 5 sowie PKI Mechanismen als Back-End zur Authentisierung anbieten.

Autorisierung soll durch die Verwendung einer Zugangskontrollliste, die der POSIX ACL-Semantik folgt, erreicht werden. Die Flexibilität und zusätzlichen Möglichkeiten durch ACL sind besonders in Clustern von Wichtigkeit, die aus tausenden von Knoten und Nutzerkonten bestehen, wie es bei Lustre der Fall ist.

Der Schutz der Daten wird durch einen Verschlüsselungsmechanismus wie beim SFS (Secure File System) Dateisystem von StorageTek/University of Minnesota. Dort werden die Daten automatisch zur Laufzeit auf dem Client ver- bzw. entschlüsselt. Die Verschlüsselung basiert dabei auf einem Protokoll, welches gemeinsame Schlüssel verwendet.



Quellenverzeichnis

- [CFS] Cluster File Systems, Inc.: www.clusterfs.com, online 8.5.2005
- [HowTo] Cluster File Systems, Inc.: *LustreHowTo*,
<https://wiki.clusterfs.com/lustre/LustreHowto>, online 8.5.2005
- [hp] Hewlett Packard: *HP Storage Works Scalable File Share*,
<http://h71028.www7.hp.com/ERC/downloads/5982-9099EN.pdf>, online 8.5.2005
- [Lustre] Cluster Files Systems, Inc.: www.lustre.org, online 8.5.2005
- [lustre-usg-1] Cluster File Systems, Inc.: *Lustre File System*,
<http://www.lustre.org/docs/lustre-usg-1.pdf>, online 8.5.2005
- [ols2003] Cluster File Systems, Inc.: *Lustre: Building a cluster file system for 1,000 node clusters*, <http://www.lustre.org/docs/ols2003.pdf>, online 8.5.2005
- [Sandia] Sandia National Laboratories: www.sandia.gov, online 8.5.2005
- [sheet] Cluster Files Systems, Inc.: *Lustre datasheet*, www.lustre.org/docs/lustre-datasheet.pdf, online 8.5.2005
- [top500] Top500 Supercomputer Sites: www.top500.org, online 8.5.2005
- [white] Cluster Files Systems, Inc.: *Lustre whitepaper*,
www.lustre.org/doc/whitepaper.pdf, online 8.5.2005



Abbildungsverzeichnis

Abbildung 1: Aufbau eines Lustre-Systems [white]	6
Abbildung 2: Bestandteile der einzelnen Komponenten eines Lustre-Systems [lustre-usg-1]	7
Abbildung 3: Clients kommunizieren immer mit OSTs, nie direkt mit OBDs	8
Abbildung 4: Fällt ein MDS aus, kann ein Client durch eine Anfrage an den LDAP-Server einen weiteren MDS in Erfahrung bringen und dann mit diesem kommunizieren [white]	10
Abbildung 5: Beispiel eines Lustre-Systems [white]	11