

# Cluster Dateisysteme

Ausarbeitung zum Vortrag im Rahmen des Proseminars

„Speicher- und Dateisysteme“

an der Universität Hamburg im Sommersemester 2012

Betreuer:

Michael Kuhn

Michaela Zimmer

Ausgearbeitet von:

Marcus Struwe

Matr. Nr. 4932603

Hamburg, 29.9.2012

# INHALT

1.)	Einleitung.....	2
2.)	Motivation.....	2
a.	Konsistenz der Daten .....	2
b.	Performanz.....	2
3.)	Begriffe und Besonderheiten bei Rechnerclustern .....	3
a.	Grundlegende Architektur .....	3
i.	Hochverfügbarkeitscluster .....	4
ii.	Load – balancing Cluster .....	4
iii.	High Performance Computing Cluster .....	4
b.	Probleme bei Rechnerclustern.....	5
4.)	Funktionen von Cluster Dateisystemen .....	7
a.	Locking .....	7
b.	Fencing.....	8
5.)	Ein Beispiel zu Cluster Dateisystemen - GPFS.....	9
a.	Striping.....	9
b.	GPFS – Pro und Contra.....	9
6.)	Bildnachweis.....	11
7.)	Literaturverzeichnis.....	12

## 1.) EINLEITUNG

In der vorliegenden Arbeit sollen Cluster Dateisysteme anhand verschiedener Aspekte betrachtet werden. Zu Beginn wird begründet, weshalb Cluster Dateisysteme überhaupt benötigt werden. In dem darauffolgenden Abschnitt sollen einige Begriffe und Sachverhalte, die im Zusammenhang mit Rechnerclustern für die weiteren Betrachtungen von Bedeutung sind, erläutert werden. Im Anschluss werden einige Funktionen von Cluster Dateisystemen näher betrachtet. Abschließend wird ein Cluster Dateisystem (GPFS) als Beispiel für die Umsetzung der vorgenannten Anforderungen und Problemstellungen vorgestellt.

## 2.) MOTIVATION

In diesem Abschnitt soll die Notwendigkeit von Cluster Dateisystemen begründet werden. Es gibt einige wesentliche Unterschiede zwischen Cluster Dateisystemen und anderen Dateisystemen, die unter den besonderen Bedingungen von Rechnerclustern eine entscheidende Rolle spielen.

Auf der einen Seite sind Cluster Dateisysteme insbesondere für den gleichzeitigen Zugriff auf große Dateien von Bedeutung. Der parallele Zugriff auf Dateien stellt vor dem Hintergrund von Datenintegrität und Konsistenz der Daten ein besonderes Problem dar.

### a. KONSISTENZ DER DATEN

Als Beispiel für dieses Problem könnte man sich das folgende Szenario überlegen:

Vorausgesetzt es gibt einen gleichzeitigen Zugriff auf eine Datei, so stellt sich die Frage, wie man verhindert, dass dieselben Datenbereiche eines Dateielements mit zwei verschiedenen Werten beschrieben werden können. Angenommen ein Client A öffnet die Datei x und schreibt den Wert 5 hinein und weiter angenommen, Client B öffnet die Datei x gleichzeitig und schreibt den Wert 7 hinein, so ergeben sich zwei Probleme. (N.N., nach : Cluster Dateisystem - Wikipedia, 2012)

Zum einen stellt sich nun die Frage, welche Datei letztlich erhalten werden soll und zum anderen, wie ein solcher Zugriff auf gleiche Datenbereiche und ein gleichzeitiges Verändern verhindert werden kann.

Eine Sperrung einer gesamten Datei durch den Zugriff eines Clients, wie im Falle von Einzelplatzrechnern bei Zugriff einer Anwendung auf die Datei, scheidet somit aus Gründen der Performanz aus.

### b. PERFORMANZ

Andererseits spielt im Bereich von Rechnerclustern auch die Bandbreite, mit denen Datenmengen transportiert werden können, eine große Rolle. Rechnercluster verfügen über eine große Menge an Rechenkapazität. Um diese ausschöpfen zu können, muss der lesende und der schreibende Datenfluss in ausreichender Größe gegeben sein. Bei verteilten

Dateisystemen, also solchen die nicht lokal beim Client gespeichert sind, sondern auf Netzwerkressourcen vorliegen, ist die Bandbreite des Netzwerks als mögliche Begrenzung der Leistungsfähigkeit des Gesamtsystems anzusehen.

Die sich aus den dargestellten Anforderungen ergebende Notwendigkeit einer Koordination der Zugriffe auf eine Datei macht ein Cluster Dateisystem notwendig, welche unter den gegebenen Bedingungen den Zugriff regelt und dabei die spezifischen Probleme und Anforderungen berücksichtigt.

### 3.) BEGRIFFE UND BESONDERHEITEN BEI RECHNERCLUSTERN

Rechnercluster besitzen eine eigene Terminologie, welche für die weitere Darstellung benötigt wird. Darüber hinaus gibt es zahlreiche Aspekte, die in ihrer Form nur bei Rechnerclustern vorkommen und sollen daher für die weiteren Betrachtungen an dieser Stelle erläutert werden.

#### a. Grundlegende Architektur

Der Begriff Rechnercluster an sich beschreibt zunächst eine Menge an Rechnern, welche über ein Netzwerk miteinander verbunden sind. Jeder dieser Rechner besitzt bestimmte Ressourcen, z.B. Rechenkapazität. Ein Zusammenschluss von solchen Rechnern, die Knoten oder „nodes“ genannt werden, stellt ein Cluster dar. Die Dienstleistungen, die die Summe dieser Knoten zur Verfügung stellen, können von anderen Rechnern in Anspruch genommen werden. Diese Rechner werden Clients genannt. Über die Modalitäten der Zusammenarbeit oder der Aufgabenverteilung, sowie über den Zweck dieses Verbundes ist damit noch keine Aussage getroffen.

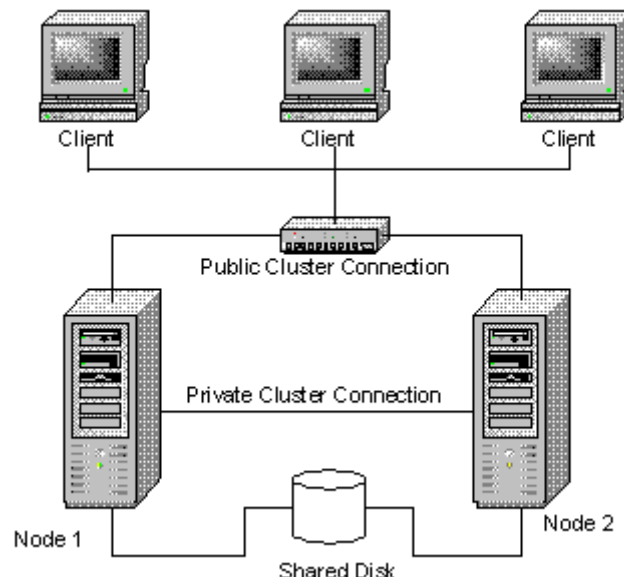


ABBILDUNG 1: PRINZIPELLER AUFBAU EINES EINFACHEN CLUSTERS MIT ZWEI KNOTEN

In der Praxis unterscheidet man einige charakteristische Typen von Clustern, welche im Folgenden dargestellt werden sollen. (N.N., Clustertypen : HA Cluster, HPC Cluster, ... )

## **i. HOCHVERFÜGBARKEITSCUSTER**

Unter dem Begriff Hochverfügbarkeitscluster versteht man solche Rechnerverbünde, welche in ihrer Funktion einen besonderen Anspruch an die Verfügbarkeit der bereitgestellten Funktionen oder Daten stellen. Solche Cluster sollen auch bei dem Ausfall einzelner Komponenten oder Verbindungen ihre Aufgaben erfüllen. Denkbare Beispiele sind Rechnerverbünde zur Steuerung des Straßenverkehrs oder Bankrechner.

Die Komponenten werden in einem Cluster allgemein als Knoten bzw. „nodes“ bezeichnet. Von besonderer Bedeutung ist, wie im Abschnitt 4.) deutlich werden wird, der Fall einer verlorenen Verbindung zwischen den Knoten. Eine Absprache bezüglich der Zugriffe auf Dateien fiele damit aus und Dateninkonsistenz wäre die mögliche Folge. Bei Hochverfügbarkeitsclustern sollte zudem der Aufbau des Clusters so vorgenommen werden, dass es möglichst viele Redundanzen gibt. Dadurch kann man verhindern, dass ein einziger Ausfall einer Komponente den gesamten Cluster lahm legen kann. Als ein Beispiel kann die räumliche Trennung der Knoten zur Sicherstellung der Versorgung mit Strom angesehen werden. Selbst wenn in einer Region die Stromversorgung ausfällt, kann damit der Cluster weiterhin erreichbar sein und seine Aufgaben erledigen. Man versucht also solche Fehlerquellen zu eliminieren, welche zum Ausfall des gesamten Clusters führen könnten. Solche Fehlerquellen, welche den gesamten Cluster außer Funktion setzen würden, werden als „single point of failure“ bezeichnet. Vor allem in Bereichen in denen Hochverfügbarkeit eine Rolle spielt, sind solche „single points of failure“ zu vermeiden. (N.N., Single Point Of Failure - Wikipedia, 2010)

## **ii. LOAD – BALANCING CLUSTER**

Bei den „load – balancing“ Clustern spielt die Aufteilung der zu bearbeitenden Aufgaben eine große Rolle. Entfällt auf einen Cluster eine große Zahl von Zugriffen oder Anfragen, so ist es entscheidend die Belastung der einzelnen Knoten so zu verteilen, dass im Idealfall auf jeden Knoten die näherungsweise gleiche Last entfällt. Bei einem möglichen Ausfall eines Knotens muss die Verteilung der Aufgaben auf die anderen Knoten diesen Umstand berücksichtigen. Um dieses Ziel zu erreichen, ist eine Kommunikation zwischen den Knoten unabdingbar. Ein wesentlicher Aspekt bei dieser Art der Cluster ist die Möglichkeit der Verwendung von einfacher Hardware, welche allein durch den Verbund eine große Leistung erzielen kann. Statt einen sehr leistungsfähigen Einzelrechner zu verwenden, können also günstigere z.T. handelsübliche Rechner verwendet werden.

## **iii. HIGH PERFORMANCE COMPUTING CLUSTER**

Ein High Performance Computing Cluster dient der schnellen Verarbeitung großer Datenmengen. Ein Beispiel für solche Rechnercluster sind Klimarechner oder solche Rechnercluster, die große Datenmengen aus wissenschaftlichen Experimenten verarbeiten sollen. Hierbei spielt die Zeit, welche für die zumeist komplexe und damit rechenintensive Verarbeitung der Daten benötigt wird, eine große Rolle. Aus dieser Notwendigkeit ergibt sich eine Reihe von Anforderungen an die zum Einsatz kommende Hardware. Neben der Anforderung an eine große Zahl von Berechnungen, welche durch die Zahl der Fleißkommaberechnungen (FLOPS) angegeben werden kann, durchführen zu können muss der Cluster auch die zu verarbeitende Datenmenge auf die einzelnen Knoten verteilen und

von ihnen zurücknehmen können. Die Möglichkeit auf jeden Client den kompletten Datenbestand zu übertragen und seine Verarbeitung auszulösen ist einerseits wegen der bereits angesprochenen Problematik im Zusammenhang mit der Konsistenz des Datenbestands ungünstig. Andererseits würde dies im Falle von großen Datenmengen auch die Anforderungen an die Ressourcen in Form von Speicherplatz bei den Knoten unnötig in die Höhe treiben.

## b. PROBLEME BEI RECHNERCLUSTERN

Bei Clustern im Allgemeinen kommen noch einige weitere Begrifflichkeiten zur Anwendung. Mit dem Begriff „interconnect“ bezeichnet man die Verbindung zwischen den Knoten eines Clusters. Diese Verbindung wird für eine Reihe von Zwecken verwendet. Die Kommunikation umfasst dabei Absprachen bezüglich der Lastverteilungsaufgaben, die Verteilung von Daten, oder auch den sog. „heartbeat“. Der „heartbeat“ stellt dabei eine Form der Kommunikation dar, durch welche sich die Knoten untereinander Signale geben, die signalisieren, dass der sendende Knoten noch aktiv und erreichbar ist. Bei dem Ausbleiben des Signals für eine bestimmte Dauer, wird davon ausgegangen, dass der betreffende Knoten nicht mehr aktiv ist. (N.N., Cluster Interconnect - Wikipedia, 2009)

Diese „heartbeat“ Signale sind entscheidend für die Funktion eines Clusters. Bei sogenannten „shared – nothing“ Konfigurationen eines Clusters verfügt jeder Knoten über eigene Ressourcen an Daten und technischer Ausstattung. Jeder Knoten verfügt darüber hinaus über den gleichen Stand der Daten. In einer aktiv/passiv Konfiguration übernimmt der Knoten A alle Aufgaben und sendet zudem über den „interconnect“ den „heartbeat“ aus. Der Knoten B bleibt dabei passiv, solange der „heartbeat“ vom Knoten A empfangen wird. Erst wenn dieser ausbleibt, übernimmt der Knoten B sämtliche Aufgaben, da davon ausgegangen wird, dass Knoten A seine Aufgaben nicht mehr wahrnehmen kann („Failover“). Wird beim Knoten B wieder ein „heartbeat“ des Knoten A empfangen, werden alle Aufgaben wieder von Knoten A übernommen und Knoten B verhält sich wieder passiv („Failback“). Eine solche Konfiguration erfüllt die Bedingungen der Hochverfügbarkeitscluster ohne Einschränkungen in der Performanz.

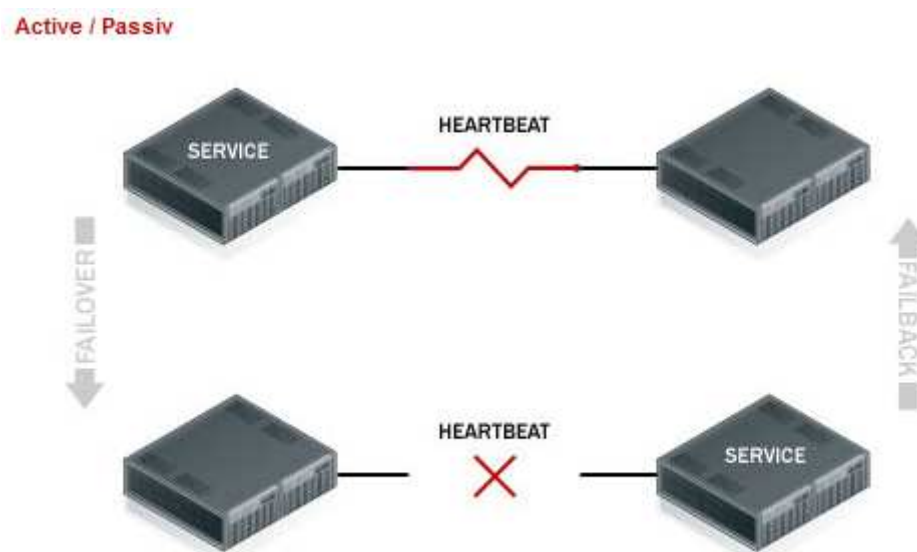


ABBILDUNG 2 DARSTELLUNG ZUR VERANSCHAULICHUNG DES FAILOVER BZW. DES FAILBACK

Im Gegensatz dazu verhält sich ein Cluster mit der aktiv/aktiv Konfiguration derart, dass beide Knoten sich die Aufgaben teilen. Sie verfügen dabei zum Beispiel über einen gemeinsamen Datenbestand („shared-all“ Konfiguration) und regeln den Zugriff von Clients darauf. Entfällt nun der „heartbeat“ eines Knotens, so übernimmt der verbleibende Knoten die gesamten Aufgaben allein. Ein Ausfall wirkt sich hierbei lediglich auf die Performanz aus, die Verfügbarkeit ist weiterhin gewährleistet. (N.N., HA Cluster : HA Clusterlösungen, HA Cluster...)

Im Falle von zwei Knoten ist der Fall noch recht übersichtlich. In der Regel sind aber bei Cluster Architekturen deutlich mehr Knoten beteiligt. Fällt nun die Kommunikation zwischen einzelnen Knoten aus, so stellt sich unmittelbar die Frage, welches der verbliebenen Teile des Clusters die Aufgaben übernehmen soll. Im Falle eines Ausfalls eines Knotens selbst, wäre dieser Umstand kein großes Problem. Die verbliebenen Knoten würden die Aufgaben übernehmen und der ausgefallene Knoten kann keine Veränderungen mehr vornehmen, da er ausgefallen ist. Schwieriger wird es, wenn lediglich die Verbindung zwischen Knoten gestört ist. Ein solcher Fall liegt zum Beispiel dann vor, wenn von drei Knoten die Kommunikation zwischen einem Knoten und den anderen Knoten ausfällt. Aus der Sicht des Knotens A sind Knoten B und C ausgefallen. Es wäre denkbar, dass nun Knoten A alle Aufgaben des Clusters, zum Beispiel die Zugriffsregelung auf Dateien durch Clients, übernimmt, da er davon ausgeht, dass Knoten B und C ausgefallen sind. Aus der Sicht der Knoten B und C ist allerdings Knoten A ausgefallen, wodurch sich diese zur Übernahme der Kontrolle der clientseitigen Dateizugriffe entscheiden werden. In einem solchen Fall wäre der Zugriff durch Clients auf den Datenbestand auf zwei Teilcluster aufgeteilt, welche von den jeweiligen Zugriffen der anderen Clients des anderen Teilclusters nichts wissen. Ein solcher Fall würde unweigerlich in der Inkonsistenz des Datenbestands enden. Man nennt ein solches Szenario „split-brain“ Szenario, weil hierbei zwei Clusterteile untereinander nicht kommunizieren und jeweils die gleichen Aufgaben übernehmen. (N.N., Networks and Servers : Failover Clustering (III))

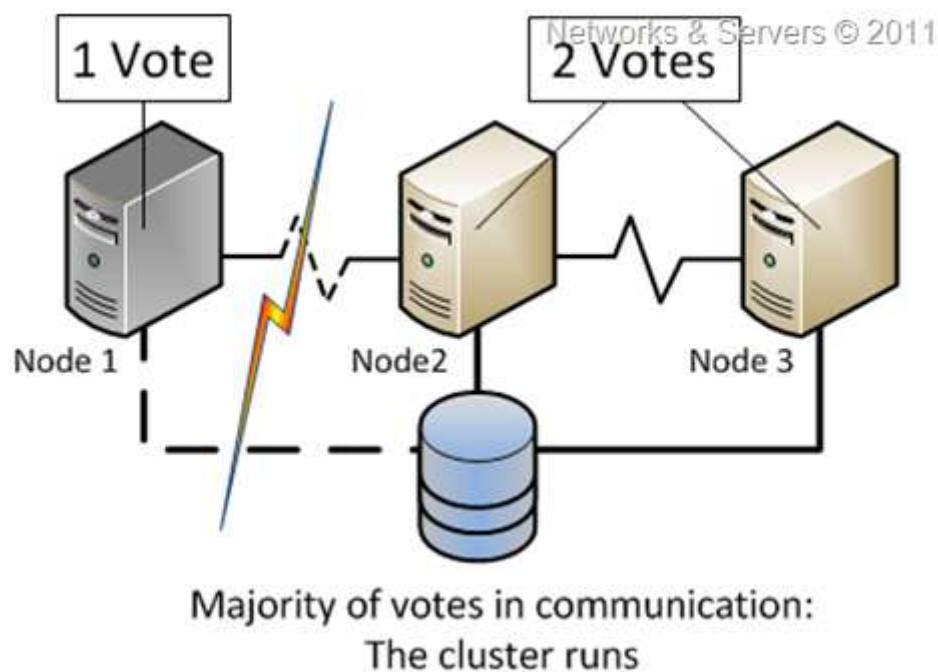


ABBILDUNG 3 DARSTELLUNG DES FENCING BEI EINER UNGERADEN ANZAHL VON NODES

Um dies zu verhindern wird in der Praxis eine Regel eingesetzt, welche dem verbliebenen größeren Cluster, also dem, welcher mehr als die Hälfte der ursprünglich vorhandenen Knoten durch deren „heartbeat“ als aktiv und erreichbar annehmen kann, sämtliche Aufgaben überträgt („fencing“). Der kleinere Teilcluster erkennt diesen Zustand also auch ohne Kommunikation und verhält sich passiv. Für das geschilderte Beispiel ergibt sich, aus einer ursprünglichen Anzahl von drei Knoten aus der Sicht des Knoten A, das Fehlen der Erreichbarkeit von mehr als der Hälfte der ursprünglichen Knoten. Er würde sich also abschalten. Für die Rechner B und C, zwischen denen weiterhin Kommunikation besteht, stellen sie die größere Hälfte des Ursprungsclusters dar, und beantworten die Anfragen ihrer Clients weiterhin.

Dadurch sind die Zugriffe der Clients auf den Datenbestand eindeutig geregelt und der Datenbestand bleibt konsistent. Allerdings sind bei solchen Konfigurationen geradzahlige Anzahlen von Knoten wieder problematisch und werden möglichst vermieden.

## 4.) FUNKTIONEN VON CLUSTER DATEISYSTEMEN

Aus den bereits geschilderten Problemen bei Rechnerclustern ergibt sich die Notwendigkeit eines Dateisystems, welches diese Aspekte hinsichtlich der Erfordernisse von Rechnerclustern berücksichtigt.

### a. LOCKING

Der Aspekt, welcher bisher am häufigsten angesprochen wurde und der auch ein, wenn nicht das entscheidende Kriterium darstellt, ist die Konsistenz von Daten. Entscheidend hierbei ist, den schreibenden Zugriff auf Daten zu kontrollieren.

An Einzelplatzsystemen erfolgt diese Zugriffskontrolle in der Regel durch das Sperren („lock“) der gesamten Datei zugunsten einer Anwendung oder eines Prozesses. Wird also ein schreibender Zugriff auf eine Datei angefordert, so wird diese Datei für alle anderen Anwendungen gesperrt. Der Zugriff auf diese Datei in schreibender, also verändernder, Art kann durch verschiedene Anwendungen damit nur noch sequenziell erfolgen. Auf solchen Systemen erfolgt die Zugriffskontrolle, also das gezielte Sperren von Dateien mittels des jeweiligen Betriebssystems. (N.N., nach : Cluster Dateisystem - Wikipedia, 2012)

In Rechnerclustern, in denen ein Zugriff auf eine Datei gleichzeitig und durch mehrere Clients möglich sein soll, ist daher eine andere Art der Kontrolle nötig, da es in diesem Falle kein übergeordnetes Betriebssystem gibt. Daher ist es erforderlich den Zugriff durch eine Instanz erfolgen zu lassen. Diese Instanz ist in dem Fall des Cluster Dateisystems das Dateisystem selbst. Verwaltet werden die Informationen über das Sperren der Zugriffe in den Metadaten. Diese Metadaten werden auf sogenannten Metadatenservern vorgehalten.

Denkbar wäre das Sperren der gesamten Datei, sobald ein Zugriff darauf erfolgt. Aus Gründen der Performanz wäre dies im Falle von großen Dateien, die parallel verarbeitet werden sollen, aber zu unflexibel, da eine Bearbeitung in diesem Fall nur sequenziell erfolgen könnte. Bei Cluster Dateisystemen gibt es zahlreiche Konfigurationsmöglichkeiten die Größe der zu sperrenden Dateiateile zu konfigurieren. Die „locks“ können auf „byterange“ Ebene, also auf der Basis einer bestimmten Bytelänge, auf „block“ Ebene, also auf einzelne Blocks oder auf Dateiebene („file – level“) erfolgen. (Landhäußer, 2010)



Letztere ist aus den bereits ausgeführten Gründen nicht günstig. Die Festlegung, für welche Bereiche auf welcher Ebene eine Sperrung erfolgt, hängt also von der Nutzung der jeweiligen Daten ab und ist für die effektive Umsetzung einer Architektur eines Clusters jeweils festzulegen. Bei den Überlegungen muss darauf geachtet werden, dass die Bereiche nicht zu groß gewählt werden. Wenn die Bereiche zu groß gewählt werden, so wird man bei den Zugriffen häufig auf den Fall stoßen, dass zwei Clients auf den gleichen Teilbereich zugreifen wollen, deren Zugriff dann sequenziell, und damit langsam erfolgen muss.

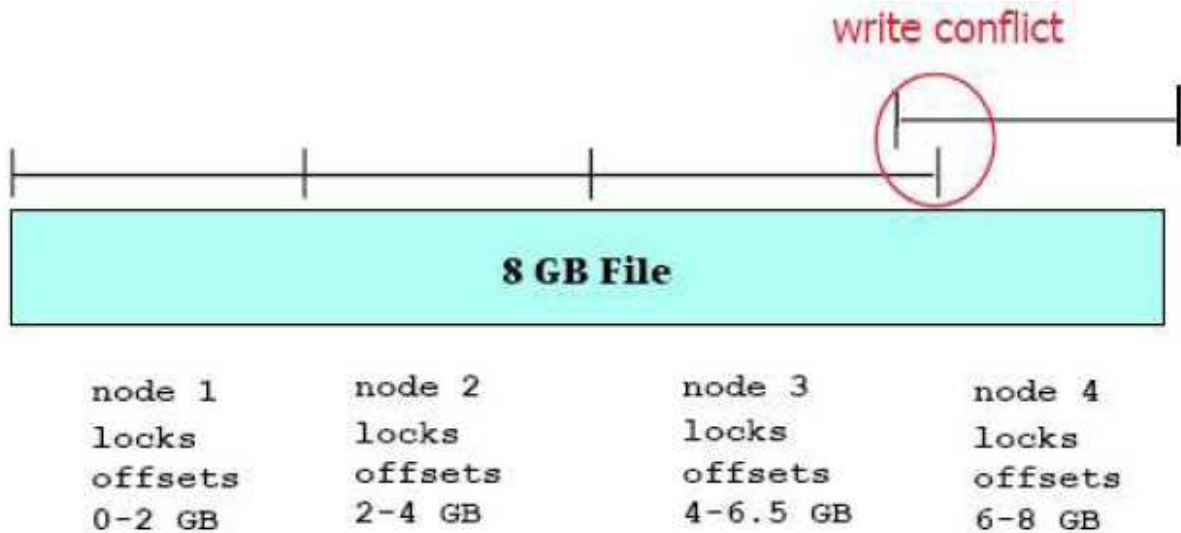


ABBILDUNG 4 DARSTELLUNG DER ÜBERLAPPUNG BEI ZU GROSSER WAHL DER DATEIBEREICHE

Werden die Bereiche hingegen zu klein gewählt, so nimmt die Menge an zu übertragenden Informationen für das Sperren bzw. Entsperrten von Bereichen erheblich zu und belastet das Netzwerk sowie die Metadatenserver. Eine günstige Wahl der Größen der jeweils gesperrten Bereiche einer Datei oder eines Datenbestandes hat dabei eine erhebliche Auswirkung auf die Performanz der Datenzugriffe.

Durch die Verwaltung dieser Sperren sind dabei zwei wesentliche Ziele, die für die Cluster Dateisysteme relevant sind, bereits erfüllt. Die Konsistenz der Daten ist gewährleistet und die Performanz ist durch den gleichzeitigen Zugriff mehrerer Clients auf eine Datei gesteigert, wenn keine Sperren auf Dateiebene erfolgen.

## b. FENCING

Ein weiteres Problem stellt die Verwaltung der Metadaten dar. Wie bereits im Abschnitt 3.b) skizziert, müssen insbesondere für den Fall von Defekten innerhalb des Clusters eine Reihe von Faktoren berücksichtigt werden. Dem bereits grob dargestellten Aspekt des Abschaltens bestimmter Clusterteile, dem sog. „fencing“ kommt eine umfassende Bedeutung zu. Das Abschalten bestimmter Teile eines Clusters zur Gewährleistung der Datenkonsistenz kann dabei auf eine ganze Reihe von Arten erfolgen. Der bereits geschilderte Fall, bei dem über die „heartbeats“ jeder Knoten selbst darüber entscheiden kann, ob er aktiv bleiben kann, oder ob er die Anfragen ignorieren muss, ist dabei nur eine Möglichkeit. Im Falle einer Aufteilung des Clusters in zwei gleich große Teile entscheidet die „Voting – Disk“ über den Aufrecht zu erhaltenden Teil des Clusters. Der „Voting – Disk“ kommt hierbei die Aufgabe zu, über den zu erhaltenden Teil des Clusters zu entscheiden. (N.N., Quorum (Informatik) - Wikipedia, 2011) In diesem Fall ist es ebenfalls möglich einen bestimmten Teil eines Clusters zu erhalten. Wenn durch einen Ausfall zwei Clusterteile entstehen kann auch ein als besonders wichtig geltender

Teil erhalten werden, selbst wenn es sich um den zahlenmäßig geringeren Teil der beiden Clusterteile handelt. Solche Fälle verwendet man insbesondere dann, wenn die Relevanz der Daten auf den erreichbaren Datenträgern nicht gleich wichtig ist.

## 5.) EIN BEISPIEL ZU CLUSTER DATEISYSTEMEN - GPFS

Als Beispiel für ein Cluster Dateisystem wird im Folgenden das „General Parallel File System“ als typischer Vertreter beschrieben. Entwickelt wurde dieses Dateisystem von IBM unter dem Namen „tiger shark multimedia file system“. Einige der Rechnercluster, die zu den Top500 der leistungsfähigsten Rechnern der Welt zählen, verwenden dieses Dateisystem. Die historische Entwicklung dieses Dateisystems kann man noch an den Kommandozeilenbefehlen erkennen, welche mit den beiden Buchstaben „mm“ für „multimedia“ beginnen. Dieses Dateisystem war also zunächst für die großen Datenmengen, welche bei Multimediadateien auftreten, entwickelt worden. Unter dem Namen GPFS ist es seit 1998 auf dem Markt verfügbar und ist aktuell in der Version 3.5 (April 2012).

### a. STRIPING

Eines der Merkmale von Clusterdateisystemen ist das sog. "striping", welches auch bei GPFS eingesetzt wird. Darunter versteht man das Aufteilen einer Datei auf mehrere Festplatten in Streifen (sog. „stripes“). Eine Datei befindet sich damit auf mehreren Festplatten und würde bei dem Verlust einer einzigen Festplatte dazu führen, dass alle Dateien, von denen sich „stripes“ auf der ausgefallenen Festplatte befanden, unbrauchbar geworden wären. Um dieses Problem zu umgehen wird ein Raid – System eingesetzt, durch welches die gleichen Daten möglichst verteilt auf mehreren Festplatten redundant vorgehalten werden. Bei einem Ausfall einer Festplatte lässt sich der Datenbestand aus den redundanten Kopien wiederherstellen. Der Grund für das „striping“ liegt bei dem Gewinn an Performanz im Bereich des Datendurchsatzes. Festplatten haben von sich aus einen relativ geringen Datendurchsatz. Eine Datei, bzw. bestimmte Bereiche einer Datei, wie im Falle von Cluster Dateisystemen, würden für den Fall, dass die gesamte Datei auf einer Festplatte läge, ein sequenzielles Lesen der Datei erfordern, wenn mehrere Clients auf diese eine Datei zugreifen. Durch das Aufteilen in „stripes“, welche idealerweise an die Größe der in Abschnitt 4) erwähnten Größe der zu sperrenden Bereiche orientiert ist, wird aber ein schnelleres Bereitstellen der Daten für die Clients ermöglicht. Da bei der Aufteilung die Anfragen mit einiger Wahrscheinlichkeit in der Mehrheit der Fälle an verschiedene Festplatten gerichtet sein werden, können Anfragen parallel erfolgen, wodurch sich der Datendurchsatz erhöht.

Durch das Prinzip des „striping“ entstehen allerdings auch große Anforderungen an die Verwaltung der Metadaten. Ein Verlust oder eine Korruption dieser Daten würde das gesamte Dateisystem unbrauchbar machen. Auch bei den Metadaten wird daher das Prinzip des „striping“ in Verbindung mit redundanter Verteilung (Raid) angewandt. (Breier, 2007)

### b. GPFS – PRO UND CONTRA

Abschließend möchte ich noch einige Aspekte zu GPFS ausführen, welche für den Einsatz oder gegen ihn sprechen.

Durch den Hersteller und Entwickler IBM ist eine große Verlässlichkeit im Hinblick auf Wartung und Weiterentwicklung des Dateisystems zu erwarten. Das GPFS Cluster Dateisystem ist zudem leicht

skalierbar auf fast alle Anwendungsfälle. Der Einsatz in Clustern der Top500 - Liste spricht für die gute Anpassung an eine Reihe von Forschungsgebieten. Ein weiterer wesentlicher Faktor ist die umfangreiche Dokumentation, die zu diesem Dateisystem vorhanden ist. Es gibt aber auch einige Aspekte, die unter Umständen gegen den Einsatz von GPFS sprechen. Einerseits muss GPFS lizenziert werden und ist nicht frei einsetzbar, andererseits wird für den Einsatz zum Teil spezifische Hardware benötigt. (Breier, 2007)

## 6.) BILDNACHWEIS

Abbildung 1: Prinzipieller Aufbau eines einfachen Clusters mit zwei Knoten.....	3
<a href="http://technet.microsoft.com/en-us/library/bb727114.aspx">http://technet.microsoft.com/en-us/library/bb727114.aspx</a>	
Abbildung 2 Darstellung zur Veranschaulichung des Failover bzw. des Failback .....	5
<a href="http://www.happyware.de/_serverstorage/IT-Loesungen/Clusterloesungen/Clustertypen/HA-Cluster/">http://www.happyware.de/_serverstorage/IT-Loesungen/Clusterloesungen/Clustertypen/HA-Cluster/</a>	
Abbildung 3 Darstellung des Fencing bei einer ungeraden Anzahl von Nodes.....	6
<a href="http://networksandservers.blogspot.de/2011/09/failover-clustering-iii.html">http://networksandservers.blogspot.de/2011/09/failover-clustering-iii.html</a>	
Abbildung 4 Darstellung der Überlappung bei zu grosser Wahl der Dateibereiche .....	8
<a href="http://berrendorf.inf.fh-bonn-rhein-sieg.de/lehre/ss05/parsys/s_GPFS.pdf">http://berrendorf.inf.fh-bonn-rhein-sieg.de/lehre/ss05/parsys/s_GPFS.pdf</a>	

## 7.) LITERATURVERZEICHNIS

- Breier, S. (22. 05 2007). *Verteilte Dateisysteme I: GPFS CXFS*. Abgerufen am 29. 09 2012 von [http://www.ipvs.uni-stuttgart.de/abteilungen/sgs/lehre/lehrveranstaltungen/hauptseminare/SS07/Cluster%20Computing\\_terminen/dateien/Verteilte\\_Dateisysteme\\_GPFS\\_CXFS.pdf](http://www.ipvs.uni-stuttgart.de/abteilungen/sgs/lehre/lehrveranstaltungen/hauptseminare/SS07/Cluster%20Computing_terminen/dateien/Verteilte_Dateisysteme_GPFS_CXFS.pdf)
- Landhäußer, A. (01. 09 2010). *Source Talk Tage 2010*. Abgerufen am 29. 09 2012 von Hochperformante parallele Filesysteme für HPC: <http://www.sourcetalk.de/2010/downloads/presentations/landhaeusser.pdf>
- N.N. (02. 08 2009). *Cluster Interconnect - Wikipedia*. Abgerufen am 29. 09 2012 von [http://de.wikipedia.org/wiki/Cluster\\_Interconnect](http://de.wikipedia.org/wiki/Cluster_Interconnect)
- N.N. (28. 12 2010). *Single Point Of Failure - Wikipedia*. Abgerufen am 29. 09 2012 von [http://de.wikipedia.org/wiki/Single\\_Point\\_of\\_Failure](http://de.wikipedia.org/wiki/Single_Point_of_Failure)
- N.N. (19. 03 2011). *Quorum (Informatik) - Wikipedia*. Abgerufen am 29. 09 2012 von [http://de.wikipedia.org/wiki/Quorum\\_\(Informatik\)](http://de.wikipedia.org/wiki/Quorum_(Informatik))
- N.N. (10. 04 2012). *nach : Cluster Dateisystem - Wikipedia*. Abgerufen am 29. 09 2012 von <http://de.wikipedia.org/wiki/Cluster-Dateisystem>
- N.N. (kein Datum). *Clustertypen : HA Cluster, HPC Cluster, ...* . Abgerufen am 29. 09 2012 von [http://www.happyware.de/\\_serverstorage/IT-Loesungen/Clusterloesungen/Clustertypen/](http://www.happyware.de/_serverstorage/IT-Loesungen/Clusterloesungen/Clustertypen/)
- N.N. (kein Datum). *HA Cluster : HA Clusterlösungen, HA Cluster...* Abgerufen am 29. 09 2012 von [http://www.happyware.de/\\_serverstorage/IT-Loesungen/Clusterloesungen/Clustertypen/HA-Cluster/](http://www.happyware.de/_serverstorage/IT-Loesungen/Clusterloesungen/Clustertypen/HA-Cluster/)
- N.N. (kein Datum). *Networks and Servers : Failover Clustering (III)*. Abgerufen am 29. 09 2012 von <http://networksandservers.blogspot.de/2011/09/failover-clustering-iii.html>