

# Red Hat Enterprise Linux 6

## Global File System 2

### Red Hat Global File System 2

#### Ausgabe 7



## Rechtlicher Hinweis

Copyright © 2011 Red Hat, Inc. and others.

The text of and illustrations in this document are licensed by Red Hat under a Creative Commons Attribution–Share Alike 3.0 Unported license ("CC-BY-SA"). An explanation of CC-BY-SA is available at <http://creativecommons.org/licenses/by-sa/3.0/>. In accordance with CC-BY-SA, if you distribute this document or an adaptation of it, you must provide the URL for the original version.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, JBoss, MetaMatrix, Fedora, the Infinity Logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux® is the registered trademark of Linus Torvalds in the United States and other countries.

Java® is a registered trademark of Oracle and/or its affiliates.

XFS® is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL® is a registered trademark of MySQL AB in the United States, the European Union and other countries.

All other trademarks are the property of their respective owners.

1801 Varsity Drive  
Raleigh, NC 27606-2072 USA  
Phone: +1 919 754 3700  
Phone: 888 733 4281  
Fax: +1 919 754 3701

## Zusammenfassung

Dieses Buch liefert Informationen über die Konfiguration und Wartung von Red Hat GFS2 (Red Hat Global File System 2) für Red Hat Enterprise Linux 6.

# Inhaltsverzeichnis

## Einführung

1. Zielgruppe
2. Ergänzende Dokumentation
3. Ihr Feedback ist gefragt!
4. Dokumentkonventionen
  - 4.1. Typografische Konventionen
  - 4.2. Konventionen für Seitenansprachen (engl.: pull-quotes)
  - 4.3. Anmerkungen und Warnungen

## 1. Überblick über GFS2

- 1.1. Neue und veränderte Features
  - 1.1.1. Neue und veränderte Features für Red Hat Enterprise Linux 6.0
  - 1.1.2. Neue und veränderte Features für Red Hat Enterprise Linux 6.1
- 1.2. Bevor Sie GFS2 einrichten
- 1.3. Unterschiede zwischen GFS und GFS2
  - 1.3.1. GFS2-Befehlsnamen
  - 1.3.2. Weitere Unterschiede zwischen GFS und GFS2
  - 1.3.3. Verbesserungen der GFS2-Leistung
- 1.4. GFS2-Knotensperrung
  - 1.4.1. Leistungsoptimierung mit GFS2
  - 1.4.2. Fehlersuche und -bereinigung der GFS2-Leistung mit GFS2 Lock Dump

## 2. Erste Schritte

- 2.1. Vorbereitung
- 2.2. Aufgaben zur erstmaligen Einrichtung
- 2.3. Implementierung eines GFS2-Clusters

## 3. Verwaltung von GFS2

- 3.1. Erstellung eines Dateisystems
- 3.2. Einhängen eines Dateisystems
- 3.3. Aushängen eines Dateisystems
- 3.4. Spezielle Überlegungen zum Einhängen von GFS2-Dateisystemen
- 3.5. Verwaltung von GFS2-Festplattenkontingenten
  - 3.5.1. Konfiguration von Festplattenkontingenten
  - 3.5.2. Verwaltung von Festplattenkontingenten
  - 3.5.3. Genauigkeit von Kontingenten pflegen
  - 3.5.4. Synchronisation von Kontingenten mit dem `quota sync`-Befehl
  - 3.5.5. Referenzen
- 3.6. Vergrößern eines Dateisystems
- 3.7. Hinzufügen von Journalen zu einem Dateisystem
- 3.8. Datenjournale
- 3.9. Konfiguration der `atime`-Aktualisierung
  - 3.9.1. Einhängen mit `relatime`
  - 3.9.2. Einhängen mit `noatime`
- 3.10. Unterbrechung der Aktivität eines Dateisystems
- 3.11. Reparieren eines Dateisystems
- 3.12. Bind Mounts und kontextabhängige Pfade
- 3.13. Einhängereihenfolge für Bind Mounts und Dateisysteme
- 3.14. Die GFS2-Rückzugfunktion

#### 4. Diagnose und Behebung von Problemen mit GFS2-Dateisystemen

- 4.1. GFS2-Dateisystem zeigt nur geringe Leistung
- 4.2. Einrichten von NFS auf GFS2
- 4.3. GFS2-Dateisystem hängt sich auf und erfordert Neustart eines Knotens
- 4.4. GFS2-Dateisystem hängt sich auf und erfordert Neustart aller Knoten
- 4.5. GFS2-Dateisystem hängt neu erstellten Cluster-Knoten nicht ein
- 4.6. Verbrauchter Platz in leerem Dateisystem

#### A. GFS2-Kontingentverwaltung mit dem `gfs2_quota`-Befehl

- A.1. Kontingente festlegen mit dem `gfs2_quota`-Befehl
- A.2. Kontingentgrenzen und -verbrauch anzeigen mit dem `gfs2_quota`-Befehl
- A.3. Kontingente synchronisieren mit dem `gfs2_quota`-Befehl
- A.4. Erzwingen von Kontingenten aktivieren/deaktivieren
- A.5. Aktivieren der Kontingentberechnung

#### B. Konvertieren eines Dateisystems von GFS auf GFS2

#### C. Versionsgeschichte

#### Stichwortverzeichnis

# Einführung

Dieses Buch liefert Informationen über die Konfiguration und Wartung von Red Hat GFS2 (Red Hat Global File System 2), das Bestandteil des Resilient-Storage-Add-Ons ist.

## 1. Zielgruppe

Dieses Buch richtet sich in erster Linie an Linux-Systemadministratoren, die bereits mit den folgenden Arbeitsvorgängen vertraut sind:

- Verfahren zur Linux-Systemadministration, einschließlich Kernel-Konfiguration
- Installation und Konfiguration von gemeinsam genutzten Speichernetzwerken, wie z.B. Fibre-Channel-SANs

## 2. Ergänzende Dokumentation

Werfen Sie einen Blick auf die folgenden Quellen für weitere Informationen zur Verwendung von Red Hat Enterprise Linux:

- *Installationshandbuch* — Liefert Informationen bezüglich der Installation von Red Hat Enterprise Linux 6.
- *Deployment-Handbuch* — Liefert Informationen zum Einsatz, der Konfiguration und der Administration von Red Hat Enterprise Linux 6.
- *Handbuch zur Speicheradministration* — Liefert Informationen über die effiziente Verwaltung von Speichergeräten und Dateisystemen auf Red Hat Enterprise Linux 6.

Werfen Sie für weitere Informationen über das Hochverfügbarkeits-Add-On und das Resilient-Storage-Add-On für Red Hat Enterprise Linux 6 einen Blick auf die folgenden Quellen:

- *Überblick über das Hochverfügbarkeits-Add-On* — Liefert einen umfassenden Überblick über das Red Hat Hochverfügbarkeits-Add-On.
- *Cluster-Administration* — Liefert Informationen zur Installation, Konfiguration und Verwaltung des Hochverfügbarkeits-Add-Ons.
- *Administration des Logical Volume Manager* — Liefert eine Beschreibung des Logical Volume Managers (LVM), inklusive Informationen zum Einsatz von LVM in einer Cluster-Umgebung.
- *DM Multipath* — Liefert Informationen über die Verwendung des Device-Mapper Multipath Features von Red Hat Enterprise Linux.
- *Lastverteilungs-Administration* — Liefert Informationen zur Konfiguration von Hochleistungssystemen und -diensten mit dem Red Hat Lastverteilungs-Add-On, einer Gruppe integrierter Software-Komponenten, die Linux Virtual Server (LVS) bereitstellen, um IP-Lasten über eine Gruppe realer Server zu verteilen.
- *Versionshinweise* — Liefert Informationen zu aktuellen Releases von Red Hat Produkten.

Dokumentation zum Hochverfügbarkeits-Add-On und andere Red Hat Dokumente stehen als HTML-, PDF-, und RPM-Versionen auf der Red Hat Enterprise Linux Dokumentations-CD sowie online unter <http://www.redhat.com/docs/> zur Verfügung.

## 3. Ihr Feedback ist gefragt!

Falls Sie einen Fehler in diesem Handbuch finden oder eine Idee haben, wie dieses verbessert werden könnte, freuen wir uns über Ihr Feedback! Bitte reichen Sie einen Fehlerbericht in Bugzilla (<http://bugzilla.redhat.com/>) für das Produkt **Red Hat Enterprise Linux 6** und die Komponente **doc-**

**Global\_File\_System\_2** ein. Vergewissern Sie sich beim Einreichen eines Fehlerberichts, dass Sie die Kennung des Handbuchs mit angeben: **rh-gfs2(EN)-6 (2011-05-19T15:15)**.

Falls Sie uns einen Vorschlag zur Verbesserung der Dokumentation senden möchten, sollten Sie hierzu möglichst genaue Angaben machen. Wenn Sie einen Fehler gefunden haben, geben Sie bitte die Nummer des Abschnitts und einen Ausschnitt des Textes an, damit wir diesen leicht finden können.

## 4. Dokumentkonventionen

Dieses Handbuch verwendet mehrere Konventionen, um bestimmte Wörter und Phrasen hervorzuheben und Aufmerksamkeit auf spezifische Informationen zu lenken.

In PDF- und Papierausgaben verwendet dieses Handbuch Schriftbilder des [Liberation-Fonts](#)-Sets. Das Liberation-Fonts-Set wird auch für HTML-Ausgaben verwendet, falls es auf Ihrem System installiert ist. Falls nicht, werden alternative, aber äquivalente Schriftbilder angezeigt. Beachten Sie: Red Hat Enterprise Linux 5 und die nachfolgende Versionen beinhalten das Liberation-Fonts-Set standardmäßig.

### 4.1. Typografische Konventionen

Es werden vier typografische Konventionen verwendet, um die Aufmerksamkeit auf spezifische Wörter und Phrasen zu lenken. Diese Konventionen und die Umstände, unter denen sie auftreten, sind folgende:

#### Nichtproportional Fett

Dies wird verwendet, um Systemeingaben hervorzuheben, einschließlich Shell-Befehle, Dateinamen und Pfade. Es wird ebenfalls zum Hervorheben von Tasten und Tastenkombinationen verwendet. Zum Beispiel:

Um den Inhalt der Datei **my\_next\_bestselling\_novel** in Ihrem aktuellen Arbeitsverzeichnis zu sehen, geben Sie den Befehl **cat my\_next\_bestselling\_novel** in den Shell-Prompt ein und drücken Sie **Enter**, um den Befehl auszuführen.

Das oben aufgeführte Beispiel beinhaltet einen Dateinamen, einen Shell-Befehl und eine Taste. Alle werden nichtproportional fett dargestellt und alle können, dank des Kontextes, leicht unterschieden werden.

Die Tastenkombination kann von einer Taste durch ein Pluszeichen, das alle Teile der Tastenkombination miteinander verbindet, unterschieden werden. Zum Beispiel:

Drücken Sie **Enter**, um den Befehl auszuführen.

Press **Ctrl+Alt+F2** to switch to a virtual terminal.

Der erste Absatz hebt die jeweilige Taste hervor, die gedrückt werden soll. Der zweite Absatz hebt zwei Tastenkombinationen hervor (jeweils ein Satz von drei Tasten, wobei jeder Satz gleichzeitig gedrückt wird).

Falls Quellcode diskutiert wird, werden Klassennamen, Methoden, Funktionen, Variablennamen und Rückgabewerte, die innerhalb eines Abschnitts erwähnt werden, wie oben gezeigt **nichtproportional fett** dargestellt. Zum Beispiel:

Zu dateiverwandten Klassen zählen **filesystem** für Dateisysteme, **file** für Dateien und **dir** für Verzeichnisse. Jede Klasse hat ihren eigenen Satz an Berechtigungen.

#### Proportional Fett

Dies kennzeichnet Wörter oder Phrasen, die auf einem System vorkommen, einschließlich Applikationsnamen, Text in Dialogboxen, beschriftete Schaltflächen, Bezeichnungen für Auswahlkästchen und Radio-Buttons, Überschriften von Menüs und Untermenüs. Zum Beispiel:

Wählen Sie **System** → **Einstellungen** → **Maus** in der Hauptmenüleiste aus, um die **Mauseinstellungen** zu öffnen. Klicken Sie im Reiter **Tasten** auf das Auswahlkästchen **Mit links bediente Maus** und anschließend auf **Schließen**, um die primäre Maustaste von der linken auf die rechte Seite zu ändern (d.h., um die Maus auf Linkshänder anzupassen).

To insert a special character into a **gedit** file, choose **Applications** → **Accessories** → **Character Map** from the main menu bar. Next, choose **Search** → **Find...** from the **Character Map** menu bar, type the name of the character in the **Search** field and click **Next**. The character you sought will be highlighted in the **Character Table**. Double-click this highlighted character to place it in the **Text to copy** field and then click the **Copy** button. Now switch back to your document and choose **Edit** → **Paste** from the **gedit** menu bar.

Der oben aufgeführte Text enthält Applikationsnamen, systemweite Menünamen und -elemente, applikationsspezifische Menünamen sowie Schaltflächen und Text innerhalb einer GUI-Oberfläche. Alle werden proportional fett dargestellt und sind anhand des Kontextes unterscheidbar.

#### ***Nichtproportional Fett Kursiv*** oder ***Proportional Fett Kursiv***

Egal ob nichtproportional fett oder proportional fett, ein zusätzlicher Kursivdruck weist auf einen ersetzbaren oder variablen Text hin. Kursivdruck kennzeichnet Text, der nicht wörtlich eingegeben wird, oder angezeigter Text, der sich je nach gegebenen Umständen ändert. Zum Beispiel:

Um sich mit einer Remote-Maschine via SSH zu verbinden, geben Sie an einem Shell-Prompt **ssh *username@domain.name*** ein. Falls die Remote-Maschine **example.com** ist und Ihr Benutzername auf dieser Maschine John lautet, geben Sie also **ssh *john@example.com*** ein.

Der Befehl **mount -o remount *file-system*** hängt das angegebene Dateisystem wieder ein. Um beispielsweise das **/home**-Dateisystem wieder einzuhängen, verwenden Sie den Befehl **mount -o remount /home**.

Um die Version des derzeit installierten Pakets zu sehen, verwenden Sie den Befehl **rpm -q *package***. Die Ausgabe sieht wie folgt aus: ***package-version-release***.

Note the words in bold italics above — *username*, *domain.name*, *file-system*, *package*, *version* and *release*. Each word is a placeholder, either for text you enter when issuing a command or for text displayed by the system.

Neben der Standardbenutzung für die Darstellung des Titels eines Werks, zeigt der Kursivdruck auch die erstmalige Nutzung eines neuen und wichtigen Begriffs an. Zum Beispiel:

Publican ist ein *DocBook* Publishing-System.

#### **4.2. Konventionen für Seitenansprachen (engl.: pull-quotes)**

Ausgaben des Terminals und Auszüge aus dem Quellcode werden visuell vom umliegenden Text hervorgehoben.

Eine Ausgabe, die an das Terminal gesendet wird, wird in den Schrifttyp **nichtproportional Roman** gesetzt und demnach wie folgt präsentiert:

```
books      Desktop  documentation  drafts  mss    photos  stuff  svn
books_tests Desktop1  downloads      images  notes  scripts svgs
```

Auszüge aus dem Quellcode werden ebenfalls in den Schrifttyp **nichtproportional Roman** gesetzt, doch wird zusätzlich noch die Syntax hervorgehoben:

```
package org.jboss.book.jca.ex1;

import javax.naming.InitialContext;

public class ExClient
{
    public static void main(String args[])
        throws Exception
    {
        InitialContext iniCtx = new InitialContext();
        Object          ref    = iniCtx.lookup("EchoBean");
        EchoHome        home   = (EchoHome) ref;
        Echo            echo   = home.create();

        System.out.println("Created Echo");

        System.out.println("Echo.echo('Hello') = " + echo.echo("Hello"));
    }
}
```

### 4.3. Anmerkungen und Warnungen

Zu guter Letzt verwenden wir drei visuelle Stile, um die Aufmerksamkeit auf Informationen zu lenken, die andernfalls vielleicht übersehen werden könnten.



#### Anmerkung

Eine Anmerkung ist ein Tipp, ein abgekürztes Verfahren oder ein alternativer Ansatz für die vorliegende Aufgabe. Das Ignorieren von Anmerkungen sollte keine negativen Auswirkungen haben, aber Sie verpassen so vielleicht einen Trick, der Ihnen das Leben vereinfachen könnte.



#### Wichtig

Die Wichtig-Schaukästen lenken die Aufmerksamkeit auf Dinge, die sonst leicht übersehen werden können: Konfigurationsänderungen, die nur für die aktuelle Sitzung gelten oder Dienste, für die ein Neustart nötig ist, bevor eine Aktualisierung wirksam wird. Das Ignorieren von Wichtig-Schaukästen würde keinen Datenverlust verursachen, kann aber unter Umständen zu Ärgernissen und Frustration führen.



#### Warnung

Eine Warnung sollte nicht ignoriert werden. Das Ignorieren von Warnungen führt mit hoher Wahrscheinlichkeit zu Datenverlust.

## Kapitel 1. Überblick über GFS2

Das Red Hat GFS2-Dateisystem ist Bestandteil des Resilient Storage Add-Ons. Es ist ein natives Dateisystem, das sich direkt mit der Dateisystem-Schnittstelle des Linux-Kernels (VFS-Schicht) verbindet. Wenn GFS2 als Cluster-Dateisystem implementiert wird, verwendet es verteilte Metadaten und multiple Journale. Red Hat unterstützt die Verwendung des GFS2-Dateisystems nur wie vom Hochverfügbarkeits-Add-On implementiert.



### Anmerkung

Obwohl das GFS2-Dateisystem sowohl auf einem eigenständigen System als auch als Teil einer Cluster-Konfiguration implementiert werden kann, unterstützt Red Hat den Einsatz von GFS2 nicht für ein Ein-Knoten-System. Red Hat unterstützt jedoch eine Reihe von leistungsstarken Ein-Knoten-Dateisystemen, die für den Einsatz auf einzelnen Knoten optimiert sind und dadurch meist einen geringeren Overhead als ein Cluster-Dateisystem haben. Red Hat empfiehlt den Einsatz eines dieser Dateisysteme anstelle von GFS2 in Fällen, in denen nur ein einzelner Knoten das Dateisystem einhängen muss.

Red Hat unterstützt auch weiterhin Ein-Knoten-GFS2-Dateisysteme zum Einhängen von Snapshots von Cluster-Dateisystemen (z.B. zwecks Datensicherung).



### Anmerkung

Red Hat unterstützt GFS2 nicht für Cluster-Dateisystem-Deployments mit mehr als 16 Knoten.

GFS2 basiert auf einer 64-bit Architektur, die theoretisch ein 8 EB Dateisystem handhaben kann. Allerdings beträgt die derzeit maximal unterstützte Größe für ein GFS2-Dateisystem für 64-bit Hardware 100 TB. Die derzeit maximal unterstützte Größe für ein GFS2-Dateisystem für 32-bit Hardware beträgt 16 TB. Falls Ihr System ein größeres GFS2-Dateisystem erfordert, setzen Sie sich bitte mit Ihrem Red Hat Service-Vertreter in Verbindung.

Wenn Sie die Größe Ihres Dateisystems festlegen, sollten Sie auch Ihre Ansprüche an die Wiederherstellung berücksichtigen. Das Ausführen des **fsck.gfs2**-Befehls auf einem sehr großen Dateisystem kann eine lange Zeit in Anspruch nehmen und zudem viel Speicherkapazität verbrauchen. Zudem wird bei einem Ausfall der Festplatte oder eines Untersystems die Wiederherstellungszeit durch die Schnelligkeit Ihres Sicherungsmediums begrenzt. Weitere Informationen über die Menge an Arbeitsspeicher, die der **fsck.gfs2**-Befehl benötigt, finden Sie in [Abschnitt 3.11, „Reparieren eines Dateisystems“](#).

Wenn Red Hat GFS2-Knoten als Teil eines Clusters konfiguriert werden, können diese mithilfe der Konfigurations- und Verwaltungs-Tools des Hochverfügbarkeits-Add-Ons konfiguriert und verwaltet werden. Red Hat GFS2 ermöglicht in diesem Fall eine gemeinsame Verwendung von Daten unter den GFS2-Knoten in einem Cluster, wobei eine einzelne, konsistente Ansicht des Dateisystem-Namensraums über die GFS2-Knoten hinweg geboten wird. Dies ermöglicht es Prozessen auf verschiedenen Knoten, GFS2-Dateien in der gleichen Art und Weise gemeinsam zu verwenden, wie Prozesse desselben Knotens Dateien auf dem lokalen Dateisystem gemeinsam verwenden können - und das ohne erkennbaren Unterschied. Weitere Information über das Hochverfügbarkeits-Add-On finden Sie im Handbuch *Cluster-Administration*.

Obwohl ein GFS2-Dateisystem auch außerhalb von LVM eingesetzt werden kann, unterstützt Red Hat nur solche GFS2-Dateisysteme, die auf einem logischen CLVM-Datenträger angelegt wurden. CLVM ist Bestandteil des Resilient-Storage-Add-Ons und ist eine clusterweite Implementierung von LVM, aktiviert vom CLVM-Daemon **clvmd**, der logische LVM-Datenträger im Cluster verwaltet. Der Daemon erlaubt die Verwendung von LVM2, um logische Datenträger im Cluster zu verwalten und ermöglicht es allen Knoten

im Cluster, die logischen Datenträger gemeinsam zu verwenden. Für mehr Information zur Datenträgerverwaltung mit LVM werfen Sie bitte einen Blick auf das Handbuch *Administration des Logical Volume Manager*.

Das `gfs2.ko` Kernel-Modul implementiert das GFS2-Dateisystem und wird auf GFS2 Cluster-Knoten geladen.



### Anmerkung

Wenn Sie ein GFS2-Dateisystem als ein Cluster-Dateisystem konfigurieren, müssen Sie sicherstellen, dass alle Knoten im Cluster Zugriff auf den gemeinsamen Speicher haben. Asymmetrische Cluster-Konfigurationen, bei denen einige Knoten Zugriff auf den Speicher haben und andere nicht, werden nicht unterstützt. Es ist jedoch nicht nötig, dass alle Knoten das GFS2-Dateisystem auch tatsächlich selbst einhängen.

Dieses Kapitel liefert einige kurze, grundlegende Informationen zum besseren Verständnis von GFS2. Es umfasst die folgenden Abschnitte:

- ▶ [Abschnitt 1.1, „Neue und veränderte Features“](#)
- ▶ [Abschnitt 1.2, „Bevor Sie GFS2 einrichten“](#)
- ▶ [Abschnitt 1.3, „Unterschiede zwischen GFS und GFS2“](#)
- ▶ [Abschnitt 1.4, „GFS2-Knotensperrung“](#)

## 1.1. Neue und veränderte Features

Dieser Abschnitt führt die neuen bzw. veränderten Features des GFS2-Dateisystems und der GFS2-Dokumentation auf, die in der ersten Release und nachfolgenden Releases von Red Hat Enterprise Linux 6 enthalten sind.

### 1.1.1. Neue und veränderte Features für Red Hat Enterprise Linux 6.0

Red Hat Enterprise Linux 6.0 führt die folgenden Änderungen und Aktualisierungen an Dokumentationen und Features ein.

- ▶ Für die Red Hat Enterprise Linux 6 Release unterstützt Red Hat nicht länger den Einsatz von GFS2 als Ein-Knoten-Dateisystem.
- ▶ In der Red Hat Enterprise Linux 6 Release wurde der `gfs2_convert`-Befehl zur Aktualisierung von einem GFS- auf ein GFS2-Dateisystem verbessert. Weitere Informationen über diesen Befehl finden Sie in [Anhang B, Konvertieren eines Dateisystems von GFS auf GFS2](#).
- ▶ Die Red Hat Enterprise Linux 6 Release unterstützt die Einhängoptionen **discard**, **nodiscard**, **barrier**, **nobarrier**, **quota\_quantum**, **statfs\_quantum** und **statfs\_percent**. Weitere Informationen über das Einhängen eines GFS2-Dateisystems finden Sie in [Abschnitt 3.2, „Einhängen eines Dateisystems“](#).
- ▶ Die Red Hat Enterprise Linux 6 Version dieses Dokuments enthält einen neuen Abschnitt, [Abschnitt 1.4, „GFS2-Knotensperrung“](#). Dieser Abschnitt beschreibt einige Interna des GFS2-Dateisystems.

### 1.1.2. Neue und veränderte Features für Red Hat Enterprise Linux 6.1

Red Hat Enterprise Linux 6.1 führt die folgenden Änderungen und Aktualisierungen an Dokumentationen und Features ein.

- ▶ Ab der Red Hat Enterprise Linux 6.1 Release unterstützt GFS2 die standardmäßigen Linux-

Funktionen für Festplattenkontingente. GFS2-Kontingentverwaltung ist in [Abschnitt 3.5, „Verwaltung von GFS2-Festplattenkontingenten“](#) dokumentiert.

In früheren Releases von Red Hat Enterprise Linux erforderte GFS2 den `gfs2_quota`-Befehl zur Verwaltung von Festplattenkontingenten. Die Dokumentation für den `gfs2_quota`-Befehl finden Sie in [Anhang A, GFS2-Kontingentverwaltung mit dem `gfs2\_quota`-Befehl](#).

- Dieses Dokument enthält nun ein neues Kapitel, [Kapitel 4, Diagnose und Behebung von Problemen mit GFS2-Dateisystemen](#).
- Im gesamten Dokument wurden kleinere Fehler korrigiert und einige technische Sachverhalte verdeutlicht.

## 1.2. Bevor Sie GFS2 einrichten

Bevor Sie GFS2 installieren und einrichten, bestimmen Sie die folgenden Schlüsselcharakteristika Ihres GFS2-Dateisystems:

### GFS2-Knoten

Legen Sie fest, welche Knoten im Cluster die GFS2-Dateisysteme einhängen sollen.

### Anzahl der Dateisysteme

Legen Sie fest, wie viele GFS2-Dateisysteme zu Beginn erstellt werden sollen. (Es können später mehr Dateisysteme hinzugefügt werden.)

### Namen der Dateisysteme

Legen Sie einen eindeutigen Namen für jedes Dateisystem fest. Jeder Name muss eindeutig für alle `lock_dlm`-Dateisysteme im Cluster sein. Jeder Dateisystemname muss in Form einer Parametervariable vorliegen. Für dieses Handbuch werden beispielsweise die Dateisystemnamen `mydata1` und `mydata2` für einige Beispielfahrer verwendet.

### Journal

Legen Sie die Anzahl der Journale für Ihre GFS2-Dateisysteme fest. Für jeden Knoten, der ein GFS2-Dateisystem einhängt, ist ein Journal erforderlich. GFS2 erlaubt Ihnen, Journale später dynamisch hinzuzufügen, wenn nachträglich weitere Server das Dateisystem einhängen. Für Informationen zum Hinzufügen von Journalen in ein GFS2-Dateisystem siehe [Abschnitt 3.7, „Hinzufügen von Journalen zu einem Dateisystem“](#).

### Speichergeräte und Partitionen

Legen Sie die Speichergeräte und Partitionen fest, die zum Erstellen von logischen Datenträgern (via CLVM) in den Dateisystemen verwendet werden sollen.



## Anmerkung

Es kann mit GFS2 unter Umständen zu Leistungseinbußen kommen, wenn gleichzeitig von mehr als einem Knoten in demselben Verzeichnis viele Operationen zum Erstellen und Löschen ausgeführt werden. Falls dies auf Ihrem System Probleme verursacht, sollten Sie die Dateierstellung und -löschung in einem Verzeichnis möglichst lokal auf einem Knoten durchführen.

## 1.3. Unterschiede zwischen GFS und GFS2

Dieser Abschnitt listet die Verbesserungen und Änderungen auf, die GFS2 gegenüber GFS bietet.

Die Migration von GFS auf GFS2 erfordert, dass Sie Ihre GFS-Dateisysteme mithilfe des **gfs2\_convert**-Dienstprogramms auf GFS2 konvertieren. Informationen über das **gfs2\_convert**-Dienstprogramm finden Sie in [Anhang B, Konvertieren eines Dateisystems von GFS auf GFS2](#).

### 1.3.1. GFS2-Befehlsnamen

Im Allgemeinen ist die Funktionalität von GFS2 mit der von GFS identisch. Allerdings verweisen die Namen der Dateisystembefehle auf GFS2 anstelle von GFS. [Tabelle 1.1, „GFS- und GFS2-Befehle“](#) zeigen die GFS-Befehle samt entsprechender GFS2-Befehle.

**Tabelle 1.1. GFS- und GFS2-Befehle**

| GFS-Befehl                                      | GFS2-Befehl                      | Beschreibung   |
|---|----------------------------------|--|
| <b>mount</b>                                    | <b>mount</b>                     | Einhängen eines Dateisystems. Das System ermittelt selbst, ob das Dateisystem ein GFS- oder ein GFS2-Dateisystemtyp ist. Für mehr Informationen zu den GFS2-Einhängeoptionen siehe die <code>gfs2_mount(8)</code> -Handbuchseite |
| <b>umount</b>                                   | <b>umount</b>                    | Aushängen eines Dateisystems   |
| <b>fsck</b>                                     | <b>fsck</b>                      | Überprüfen und Reparieren eines ausgehängten Dateisystems  |
| <b>gfs_fsck</b>                                 | <b>fsck.gfs2</b>                 |  |
| <b>gfs_grow</b>                                 | <b>gfs2_grow</b>                 | Erweitern eines eingehängten Dateisystems  |
| <b>gfs_jadd</b>                                 | <b>gfs2_jadd</b>                 | Hinzufügen eines Journals zu einem eingehängten Dateisystem  |
| <b>gfs_mkfs</b>                                 | <b>mkfs.gfs2</b>                 | Erstellen eines Dateisystems auf einem Speichergerät   |
| <b>mkfs -t gfs</b>                              | <b>mkfs -t gfs2</b>              |  |
| <b>gfs_quota</b>                                | <b>gfs2_quota</b>                | Verwalten von Festplattenkontingenten auf einem eingehängten Dateisystem   |
| <b>gfs_tool</b>                                 | <b>gfs2_tool</b>                 | Konfigurieren, Einstellen oder Anzeigen von Informationen über ein Dateisystem   |
| <b>gfs_edit</b>                                 | <b>gfs2_edit</b>                 | Anzeigen, Drucken oder Bearbeiten der internen Struktur eines Dateisystems. Der <b>gfs2_edit</b> -Befehl kann für GFS-Dateisysteme wie auch für GFS2-Dateisysteme verwendet werden   |
| <b>gfs_tool setflag<br/>jdata/inherit_jdata</b> | <b>chattr +j<br/>(bevorzugt)</b> | Aktivieren des Journaling auf einer Datei oder einem Dateisystem   |
| <b>setfacl/getfacl</b>                          | <b>setfacl/getfacl</b>           | Anzeigen oder Einstellen der Zugriffskontrollliste einer Datei oder eines Verzeichnisses   |
| <b>setfattr/getfattr</b>                        | <b>setfattr/getfattr</b>         | Anzeigen oder Einstellen der erweiterten Parameter einer Datei   |

Eine vollständige Liste der unterstützten Optionen für die GFS2-Dateisystembefehle finden Sie in den Handbuchseiten der jeweiligen Befehle.

### 1.3.2. Weitere Unterschiede zwischen GFS und GFS2

Dieser Abschnitt listet die zusätzlichen Unterschiede hinsichtlich der Administration von GFS und GFS2 Administration auf, die nicht im [Abschnitt 1.3.1, „GFS2-Befehlsnamen“](#) beschrieben sind.

## Kontextabhängige Pfade

GFS2-Dateisysteme unterstützen keine kontextabhängigen Pfade (Context-Dependent Path Names oder kurz CDPNs), die es Ihnen ermöglichen, symbolische Links zu erzeugen, die auf verschiedene Zieldateien oder -verzeichnisse zeigen. Um dieselbe Funktionalität in GFS2-Dateisystemen zu erreichen, können Sie die **bind**-Option des **mount**-Befehls verwenden. Weitere Informationen über Bind-Mounts und kontextabhängige Pfade in GFS2 finden Sie in [Abschnitt 3.12, „Bind Mounts und kontextabhängige Pfade“](#).

## gfs2.ko-Modul

Das Kernel-Modul, welches das GFS-Dateisystem implementiert, ist **gfs.ko**. Das Kernel-Modul, welches das GFS2-Dateisystem implementiert, ist **gfs2.ko**.

## Erzwingen von Speicherkontingenten in GFS2

In GFS2-Dateisystemen werden Festplattenkontingente standardmäßig nicht erzwungen, so dass dieses Verhalten explizit aktiviert werden muss. Informationen über das Aktivieren und Deaktivieren von Festplattenkontingenten finden Sie in [Abschnitt 3.5, „Verwaltung von GFS2-Festplattenkontingenten“](#).

## Daten-Journaling

GFS2-Dateisysteme unterstützen die Verwendung des **chattr**-Befehls, um das **j**-Flag auf einer Datei oder einem Verzeichnis zu setzen bzw. zu entfernen. Wird das **+j**-Flag auf einer Datei gesetzt, so wird das Daten-Journaling auf dieser Datei aktiviert. Wird das **+j**-Flag auf einem Verzeichnis gesetzt, bedeutet dies, dass für alle nachfolgend in diesem Verzeichnis erstellten Dateien oder Verzeichnisse ebenfalls das Daten-Journaling aktiviert ist ("inherit jdata"). Die Verwendung des **chattr**-Befehls ist die bevorzugte Methode zum Aktivieren und Deaktivieren des Daten-Journalings auf einer Datei.

## Journale dynamisch hinzufügen

In GFS-Dateisystemen sind Journale eingebettete Metadaten, die sich außerhalb des Dateisystems befinden, weshalb zunächst der logische Datenträger mit dem Dateisystem vergrößert werden muss, bevor Journale hinzugefügt werden können. In GFS2-Dateisystemen sind Journale einfache (versteckte) Dateien. Dies bedeutet, dass für GFS2-Dateisysteme Journale dynamisch hinzugefügt werden können, sobald weitere Server das Dateisystem einhängen, sofern noch genügend Platz auf dem Dateisystem für zusätzliche Journale vorhanden ist. Werfen Sie einen Blick auf [Abschnitt 3.7, „Hinzufügen von Journalen zu einem Dateisystem“](#) für Informationen über das Hinzufügen von Journalen zu einem GFS2-Dateisystem.

## atime\_quantum-Parameter entfernt

Das GFS2-Dateisystem unterstützt nicht den **atime\_quantum**-Parameter, der vom GFS-Dateisystem dazu verwendet wird um festzulegen, wie oft **atime**-Aktualisierungen erfolgen. Stattdessen unterstützt GFS2 die **relatime** und **noatime** Einhängeoptionen. Die **relatime**-Einhängeoption wird empfohlen, um ein ähnliches Verhalten zu erreichen, wie mit dem **atime\_quantum**-Parameter in GFS eingestellt werden kann.

## The data= Option des mount-Befehls

Wenn Sie ein GFS2-Dateisystem einhängen, können Sie die **data=ordered** oder **data=writeback**-Option des **mount**-Befehls angeben. Wenn **data=ordered** gesetzt ist, werden die Benutzerdaten, die von einer Transaktion verändert wurden, auf die Festplatte gespeichert, bevor die Transaktion auf die Festplatte festgeschrieben wird. Dies sollte verhindern, dass der Benutzer nach einem Absturz in einer Datei nicht initialisierte Blöcke sieht. Wenn **data=writeback** gesetzt ist, werden die Benutzerdaten zu einem beliebigen Zeitpunkt nach deren Änderung auf die Festplatte geschrieben. Dies garantiert zwar nicht dieselbe Konsistenz wie der **ordered**-Modus, sollte aber bei einer gewissen Auslastung etwas schneller sein. Standardmäßig ist der **ordered**-Modus gesetzt.

## Der gfs2\_tool-Befehl

Der **gfs2\_tool**-Befehl unterstützt eine andere Reihe von Optionen als der **gfs\_tool**-Befehl für GFS:

- ▶ Der **gfs2\_tool**-Befehl unterstützt einen **journals**-Parameter, der sämtliche Informationen über die derzeit konfigurierten Journale ausgibt, einschließlich der Anzahl an Journalen auf dem Dateisystem.
- ▶ Der **gfs2\_tool**-Befehl unterstützt nicht das **counters**-Flag, das der **gfs\_tool**-Befehl zum Anzeigen von GFS-Statistiken verwendet.
- ▶ Der **gfs2\_tool**-Befehl unterstützt nicht das **inherit\_jdata**-Flag. Um ein Verzeichnis als "inherit jdata" zu kennzeichnen, können Sie das **jdata**-Flag auf dem Verzeichnis setzen oder Sie können den **chattr**-Befehl verwenden, um das **+j**-Flag auf dem Verzeichnis zu setzen. Die Verwendung des **chattr**-Befehls ist die bevorzugte Methode, um das Daten-Journaling auf einer Datei zu aktivieren bzw. zu deaktivieren.

## Der gfs2\_edit-Befehl

Der **gfs2\_edit**-Befehl unterstützt eine andere Reihe von Optionen für GFS2 als der **gfs\_edit**-Befehl für GFS. Werfen Sie für weitere Informationen über die einzelnen Optionen, die jede Version des Befehls unterstützt, einen Blick auf die Handbuchseiten von **gfs2\_edit** und **gfs\_edit**.

### 1.3.3. Verbesserungen der GFS2-Leistung

Viele Features des GFS2-Dateisystems bringen zwar keine Veränderung an der Benutzerschnittstelle des GFS-Dateisystems mit sich, verbessern jedoch die Leistung des Dateisystems.

Ein GFS2-Dateisystem bietet in den folgenden Aspekten eine verbesserte Dateisystemleistung:

- ▶ Bessere Leistung bei intensiver Benutzung eines einzelnen Verzeichnisses
- ▶ Schnellere synchrone I/O-Operationen
- ▶ Schnellere Cache-Lesevorgänge (kein Overhead zwecks Sperrung)
- ▶ Schnellere direkte I/O mit bereits zugewiesenen Dateien (vorausgesetzt, I/O-Umfang ist relativ groß, z.B. 4 M Blöcke)
- ▶ Allgemein schnellere I/O-Operationen
- ▶ Deutlich schnellere Ausführung des **df**-Befehls aufgrund von schnelleren **statfs**-Aufrufen.
- ▶ Der **atime**-Modus wurde verbessert, um die Anzahl der von **atime** generierten Schreibvorgänge im Vergleich zu GFS zu verringern.

GFS2-Dateisysteme bieten in den folgenden Aspekten eine breitere und allgemeinere Unterstützung.

- ▶ GFS2 ist Teil des Upstream-Kernels (integriert in 2.16.19)
- ▶ GFS2 unterstützt folgende Features:
  - SELinux erweiterte Attribute.
  - Die **lsattr()** und **chattr()** Attributeinstellungen über standardmäßige **ioctl()**-Aufrufe.
  - Nanosekündliche Timestamps

Ein GFS2-Dateisystem bietet folgende Verbesserungen für die Gesamteffizienz des Dateisystems:

- ▶ GFS2 verbraucht weniger Kernel-Speicher
- ▶ GFS2 braucht keine Metadaten-Generierungsnummern.  
Das Zuweisen von GFS2-Metadaten erfordert keine Lesevorgänge. Kopien der Metadatenblöcke in mehreren Journalen werden von "revoking blocks" vom Journal verwaltet, bevor die Sperrung wieder aufgehoben wird.
- ▶ GFS2 beinhaltet eine deutlich einfachere Protokollverwaltung, die nichts über nicht verlinkte Inodes oder Änderungen der Speicherkontingente weiß.
- ▶ Die **gfs2\_grow**- und **gfs2\_jadd**-Befehle verwenden Sperren, um zu verhindern, dass mehrere

Instanzen gleichzeitig ausgeführt werden.

- » Der ACL-Code wurde für Aufrufe wie `creat()` und `mkdir()` vereinfacht.
- » Nicht verlinkte Inodes, Änderungen der Festplattenkontingente und `statfs`-Änderungen werden wiederhergestellt, ohne dass das Journal neu eingehängt werden muss.

## 1.4. GFS2-Knotensperrung

Um die beste Leistung aus einem GFS2-Dateisystem herauszuholen, ist es wichtig, die grundlegende Theorie für dessen Einsatz zu verstehen. Ein Ein-Knoten-Dateisystem wird zusammen mit einem Cache implementiert, um die Latenz der Plattenzugriffe für häufig abgerufene Daten zu verringern. In Linux erfüllt der Page-Cache (sowie historisch der Buffer-Cache) diese Caching-Funktion.

Bei GFS2 besitzt jeder Knoten seinen eigenen Page-Cache, der einen Teil der Festplattendaten enthalten kann. GFS2 verwendet ein Sperrverfahren namens *Glocks* (ausgesprochen Dschie-Locks), um die Integrität der Caches zwischen den Knoten zu bewahren. Das Glock-Untersystem liefert eine Cache-Verwaltungsfunktion, die unter Verwendung des *Distributed Lock Manager* (DLM) als zugrunde liegende Kommunikationsschicht implementiert ist.

Glocks schützen den Cache auf Inode-Basis, das heißt es gibt eine Sperre pro Inode, die zur Steuerung der Caching-Schicht verwendet wird. Wird das Glock im gemeinsamen Modus (DLM-Sperrmodus: PR) zugewiesen, so dürfen die Daten unter diesem Glock auf einem oder mehreren Knoten gleichzeitig gecacht werden, so dass alle Knoten lokalen Zugriff auf diese Daten haben.

Wird das Glock im exklusiven Modus (DLM-Sperrmodus: EX) zugewiesen, so darf nur ein einziger Knoten die Daten unter diesem Glock cachen. Dieser Modus wird von allen Operationen genutzt, die Daten verändern (z.B. der `write`-Systemaufruf).

Falls ein anderer Knoten ein Glock anfragt, das jedoch nicht sofort zugewiesen werden kann, so sendet der DLM eine Nachricht an den bzw. die Knoten, der/die derzeit das Glock halten und dadurch die Anfrage blockieren, damit diese ihre Sperren verwerfen. Das Verwerfen von Glocks kann (gemessen an den Standards der meisten Dateisystemoperationen) ein langwieriger Vorgang sein. Das Verwerfen eines gemeinsam verwendeten Glocks erfordert lediglich, dass der Cache für ungültig erklärt wird (Invalidierung), was vergleichsweise schnell und proportional zur Menge der gecachten Daten ist.

Das Verwerfen eines exklusiven Glocks erfordert die Bereinigung der Protokolle und das Zurückschreiben sämtlicher veränderter Daten auf die Festplatte, gefolgt von der Invalidierung des Caches wie beim gemeinsam verwendeten Glock.

Der Unterschied zwischen einem Ein-Knoten-Dateisystem und GFS2 besteht darin, dass das Ein-Knoten-Dateisystem über einen einzigen Cache verfügt während GFS2 über separate Caches auf jedem Knoten verfügt. In beiden Fällen ist die Latenz beim Zugriff auf gecachte Daten ähnlich groß, allerdings ist die Latenz beim Zugriff auf nicht gecachte Daten deutlich größer mit GFS2, wenn ein anderer Knoten vorher bereits dieselben Daten gecacht hatte.



## Anmerkung

Aufgrund der Art, wie Caching in GFS2 implementiert wird, erreichen Sie die beste Leistung in einer dieser beiden Fälle:

- ▶ Ein Inode wird auf allen Knoten nur zum Lesezugriff verwendet
- ▶ Schreibzugriffe auf einen Inode erfolgen von einem einzigen Knoten aus.

Beachten Sie, dass das Einfügen und Entfernen von Einträgen aus einem Verzeichnis bei der Dateierstellung und -löschung als Schreibvorgang auf dem Verzeichnis-Inode zählt.

Es ist möglich, diese Regel zu ignorieren, sofern dies nur selten geschieht. Wird diese Regel zu oft ignoriert, sind erhebliche Leistungseinbußen die Folge.

Falls Sie `mmap()` für eine Datei auf einem GFS2-Dateisystem mit einem Lese/Schreib-Mapping setzen, diese jedoch nur lesen, zählt dies als Lesevorgang. Auf GFS dagegen zählt dies als Schreibvorgang, so dass GFS2 sehr viel skalierbarer ist mit `mmap()` I/O.

Falls Sie den `noatime mount`-Parameter nicht setzen, führen Lesevorgänge auch zu Schreibvorgängen, um die Timestamps der Datei zu aktualisieren. Wir empfehlen im Allgemeinen, dass alle GFS2-Benutzer mit `noatime` einhängen sollten, sofern sie nicht besondere Anforderungen an `atime` haben.

### 1.4.1. Leistungsoptimierung mit GFS2

Normalerweise ist es möglich, für eine problematische Applikation die Art und Weise anzupassen, wie diese ihre Daten speichert, um dadurch eine deutlich verbesserte Leistung zu erreichen.

Ein typisches Beispiel für eine problematische Applikation ist ein E-Mail-Server. Diese verfügen oft über ein Spool-Verzeichnis, das Dateien für jeden Benutzer enthält (**mbox**) bzw. über ein Verzeichnis für jeden Benutzer, das eine Datei pro Nachricht enthält (**maildir**). Wenn Anfragen über IMAP eingehen, wird idealerweise jedem Benutzer eine Affinität zu einem bestimmten Knoten zugewiesen. Auf diese Weise werden deren Anfragen zum Ansehen und Löschen von E-Mails tendenziell vom Cache auf diesem Knoten bedient. Falls dieser Knoten ausfällt, kann die Sitzung natürlich auf einem anderen Knoten neu gestartet werden.

Wenn E-Mail über SMTP eingeht, können individuelle Knoten so eingestellt werden, dass sie die E-Mails eines bestimmten Benutzers standardmäßig an einen bestimmten Knoten weiterleiten. Falls dieser Knoten nicht läuft, kann der empfangende Knoten die Nachricht direkt im Mail-Spool-Verzeichnis des Benutzers speichern. Dieser Aufbau dient wie gesagt dazu, eine bestimmte Gruppe von Dateien im Normalfall vorrangig auf einem einzigen Knoten zwischenspeichern, erlaubt jedoch direkten Zugriff im Falle eines Knotenausfalls.

Dieser Aufbau nutzt den Page-Cache von GFS2 am besten aus und macht darüber hinaus Ausfälle für die Applikation transparent, egal ob **imap** oder **smtp**.

Die Datensicherung (oder "Backup") ist ebenfalls oft problematisch. Falls möglich, ist es unbedingt vorzuziehen, das Working Set eines jeden Knotens direkt von dem Knoten zu sichern, der diese bestimmte Gruppe von Inodes cacht. Falls Sie zur Datensicherung ein Skript nutzen, das regelmäßig zu einem bestimmten Zeitpunkt ausgeführt wird, welches mit einer Spitze in der Antwortzeit einer auf GFS2 laufenden Applikation zusammenfällt, dann zeigt dies mit ziemlicher Sicherheit an, dass der Cluster den Page Cache nicht so effizient wie möglich nutzt.

Falls Sie die Möglichkeit haben, die Applikation zur Datensicherung zu stoppen, stellt dies natürlich kein Problem dar. Wird die Datensicherung nur von einem Knoten durchgeführt, wird andererseits nach Abschluss ein großer Teil des Dateisystems auf diesem Knoten gecacht, was Leistungseinbußen für nachfolgende Zugriffe von anderen Knoten zur Folge hat. Bis zu einem gewissen Grad kann dies vermieden werden, indem Sie nach Abschluss der Datensicherung den VFS Page Cache auf dem

Backup-Knoten bereinigen, und zwar mithilfe des folgenden Befehls:

```
echo -n 3 >/proc/sys/vm/drop_caches
```

Allerdings besteht die bessere Lösung darin sicherzustellen, dass das Working Set auf jedem Knoten entweder gemeinsam hauptsächlich für Lesezugriffe im Cluster verwendet wird, oder dass hauptsächlich von nur einem einzigen Knoten darauf zugegriffen wird.

#### 1.4.2. Fehlersuche und -bereinigung der GFS2-Leistung mit GFS2 Lock Dump

Falls die Leistung Ihres Clusters unter ineffizienter Verwendung des GFS2-Caching leidet, sehen Sie große, steigende I/O-Wartezeiten. Sie können die Informationen des GFS2 Lock Dump verwenden, um der Ursache des Problems auf den Grund zu gehen.

Die GFS2-Lock-Dump-Informationen können Sie der **debugfs**-Datei entnehmen, die sich - vorausgesetzt, das **debugfs** ist unter **/sys/kernel/debug/** eingehängt - in folgendem Pfad befindet:

```
/sys/kernel/debug/gfs2/fsname/glocks
```

Die Datei enthält eine Reihe von Zeilen. Jede Zeile, die mit einem G: beginnt, steht für ein Glock, und die folgenden Zeilen - um eine Leerstelle eingerückt - stehen für eine Information, die sich auf das in der Datei darüberstehende Glock bezieht.

Erstellen Sie am besten mithilfe des **cat** -Befehls eine Kopie des gesamten Inhalts der **debugfs**-Datei (dies kann eine längere Zeit dauern, falls Sie eine große Menge RAM und viele gecachte Inodes haben), während bei der Applikation Probleme auftreten, und schauen Sie sich die so erstellten Daten zu einem späteren Zeitpunkt an.



#### Tipp

Es kann hilfreich sein, zwei Kopien der **debugfs**-Datei zu erstellen, einige Sekunden oder ein, zwei Minuten nacheinander. Vergleichen Sie in den zwei Kopien nun die Halterinformationen bezüglich derselben Glock-Nummer. Anhand dessen sollten Sie feststellen können, ob die Arbeitslast Fortschritte macht (also nur langsam ist), oder ob sie hängen geblieben ist (was immer auf einen Fehler hindeutet und umgehend an den Red Hat Support gemeldet werden sollte).

Jede Zeile in der **debugfs**-Datei, die mit H: (Halter) beginnt, steht für eine Sperranfrage, die entweder bereits gewährt wurde oder darauf wartet, gewährt zu werden. Das Flags-Feld auf der Halterzeile f: zeigt an, was von beidem der Fall ist: Das 'W' Flag kennzeichnet eine wartende Anfrage, das 'H' Flag kennzeichnet eine gewährte Anfrage. Diejenigen Glocks mit einer hohen Anzahl wartender Anfragen sind wahrscheinlich diejenigen mit Konflikten.

[Tabelle 1.2, „Glock-Flags“](#) zeigt die Bedeutungen der verschiedenen Glock-Flags und [Tabelle 1.3, „Glock-Halter-Flags“](#) zeigt die Bedeutung der verschiedenen Glock-Halter-Flags.

**Tabelle 1.2. Glock-Flags**

| Flag | Name                   | Bedeutung  |
|------|------------------------|--|
| d    | Pending demote         | Eine wartende Anfrage zum Herabstufen (remote)   |
| D    | Demote                 | Eine Anfrage zum Herabstufen (lokal oder remote)   |
| f    | Log flush              | Das Protokoll muss festgeschrieben werden, bevor dieses Glock freigegeben werden kann            |
| F    | Frozen                 | Antworten von Remote-Knoten werden ignoriert - Wiederherstellung läuft.                          |
| i    | Invalidate in progress | Seiten unter diesem Glock werden derzeit invalidiert   |
| l    | Initial                | Gesetzt, wenn DLM-Sperre mit diesem Glock verknüpft ist  |
| l    | Locked                 | Das Glock ändert derzeit den Status  |
| p    | Demote in progress     | Das Glock antwortet derzeit auf eine Anfrage zum Herabstufen                                     |
| r    | Reply pending          | Von Remote-Knoten erhaltene Antwort wartet auf Verarbeitung                                      |
| y    | Dirty                  | Daten müssen auf die Festplatte überschrieben werden, bevor dieses Glock freigegeben werden kann |

**Tabelle 1.3. Glock-Halter-Flags**

| Flag | Name      | Bedeutung   |
|------|-----------|---|
| a    | Async     | Nicht auf Glock-Ergebnis warten (Ergebnis wird später abgerufen)              |
| A    | Any       | Jeder kompatible Sperrmodus ist zulässig                                      |
| c    | No cache  | Wenn nicht gesperrt, sofort DLM-Sperre herabstufen                            |
| e    | No expire | Nachfolgende Anfragen zur Aufhebung der Sperre ignorieren                     |
| E    | exact     | Muss exakten Sperrmodus haben   |
| F    | First     | Gesetzt, wenn der Halter der Erste ist, der für diese Sperre zugelassen wurde |
| H    | Holder    | Zeigt an, dass die angefragte Sperre gewährt wurde                            |
| p    | Priority  | Reiht Halter am Anfang der Warteschlange ein                                  |
| t    | Try       | Eine "try"-Sperre   |
| T    | Try 1CB   | Eine "try"-Sperre, die einen Callback sendet                                  |
| W    | Wait      | Gesetzt, während auf den Abschluss einer Anfrage gewartet wird                |

Nachdem Sie herausgefunden haben, welches Glock das Problem verursacht, müssen Sie nun feststellen, auf welchen Inode es sich bezieht. Dies wird angezeigt durch die Glock-Nummer (n: in der G: Zeile) im Format **type/number**. Falls **type** 2 ist, ist das Glock ein Inode-Glock und die **number** ist eine Inode-Nummer. Um den Inode zu finden, können Sie den Befehl **find -inum number** nutzen, wobei **number** die Inode-Nummer ist, die vom Hex-Format in der Glocks-Datei nach Dezimal umgewandelt wurde.



## Warnung

Falls Sie den **find**-Befehl auf einem Dateisystem ausführen, während dort Sperrkonflikte auftreten, verschlimmern Sie dadurch das Problem wahrscheinlich noch. Wenn Sie nach Inode-Konflikten suchen, ist es ratsam, die Applikation zu beenden, bevor Sie **find** ausführen.

[Tabelle 1.4, „Glock-Typen“](#) zeigt die Bedeutungen der verschiedenen Glock-Typen.

**Tabelle 1.4. Glock-Typen**

| Typennummer | Sperrtyp | Verwendung                   |
|-------------|----------|------------------------------|
| 1           | Trans    | Transaktionssperre           |
| 2           | Inode    | Inode-Metadaten und Daten    |
| 3           | Rgrp     | Ressourcengruppen-Metadaten  |
| 4           | Meta     | Der Superblock               |
| 5           | Iopen    | Inode Last-Closer-Erkennung  |
| 6           | Flock    | <b>flock(2)</b> Systemaufruf |
| 8           | Quota    | Kontingentoperationen        |
| 9           | Journal  | Journal-Mutex                |

Falls das identifizierte Glock einem anderen Typ angehört, ist dies höchstwahrscheinlich Typ 3: (Ressourcengruppe). Falls Sie eine erhebliche Anzahl von Prozessen sehen, die unter normaler Auslastung auf andere Glock-Typen warten, melden Sie dies bitte dem Red Hat Support.

Falls Sie eine Anzahl wartender Anfragen sehen, die in der Warteschlange für eine Ressourcengruppensperre sind, könnte dies eine Reihe von Ursachen haben. Zum einen gibt es unter Umständen eine hohe Anzahl von Knoten im Vergleich zur Anzahl von Ressourcengruppen im Dateisystem. Zum anderen könnte das Dateisystem unter Umständen beinahe voll sein (wodurch die Suche nach freien Blöcken durchschnittlich länger braucht). In beiden Fällen kann die Situation verbessert werden, indem mehr Speicher hinzugefügt wird und das Dateisystem mithilfe des **gfs2\_grow**-Befehls vergrößert wird.

## Kapitel 2. Erste Schritte

Dieses Kapitel beschreibt, wie Sie bei der Erstinstallation von GFS2 vorgehen müssen und beinhaltet folgende Abschnitte:

- ▶ [Abschnitt 2.1, „Vorbereitung“](#)
- ▶ [Abschnitt 2.2, „Aufgaben zur erstmaligen Einrichtung“](#)
- ▶ [Abschnitt 2.3, „Implementierung eines GFS2-Clusters“](#)

### 2.1. Vorbereitung

Bevor Sie Red Hat GFS2 einrichten, versichern Sie sich, dass Sie sich über die Schlüsseleigenschaften von GFS2-Knoten im Klaren sind (siehe [Abschnitt 1.2, „Bevor Sie GFS2 einrichten“](#)). Versichern Sie sich des Weiteren, dass die Systemuhren der GFS2-Knoten synchronisiert sind. Es wird empfohlen, die Network Time Protokoll (NTP) Software zu benutzen, die in Ihrer Red Hat Enterprise Linux Distribution enthalten ist.



#### Anmerkung

Die Systemuhren der GFS2-Knoten dürfen nur ein paar Minuten voneinander abweichen, um unnötiges Aktualisieren der Inode-Timestamps zu vermeiden. Unnötige Aktualisierungen der Inode-Timestamps haben schwerwiegende Auswirkungen auf die Cluster-Leistung.

### 2.2. Aufgaben zur erstmaligen Einrichtung

Das erstmalige Einrichten von GFS2 umfasst die folgenden Aufgaben:

1. Einrichten der logischen Datenträger
2. Erstellung eines GFS2-Dateisystems
3. Einhängen der Dateisysteme

Führen Sie die folgenden Schritte aus, um GFS2 zum ersten Mal einzurichten.

1. Erstellen Sie mithilfe von LVM einen logischen Datenträger für jedes Red Hat GFS2-Dateisystem.



#### Anmerkung

Sie können die in der Red Hat Cluster Suite enthaltenen **init.d**-Skripte verwenden, um das Aktivieren und Deaktivieren logischer Datenträger zu automatisieren. Weitere Informationen über **init.d**-Skripte finden Sie im Handbuch *Konfiguration und Verwaltung eines Red Hat Clusters*.

2. Erzeugen Sie auf den in Schritt 1 erstellen logischen Datenträgern GFS2-Dateisysteme. Wählen Sie einen eindeutigen Namen für jedes Dateisystem. Weitere Informationen über das Erstellen von GFS2-Dateisystemen finden Sie in [Abschnitt 3.1, „Erstellung eines Dateisystems“](#).

Sie können eines der folgenden Befehlsformate verwenden, um ein geclustertes GFS2-Dateisystem zu erstellen:

```
mkfs.gfs2 -p lock_dlm -t ClusterName:FSName -j NumberJournals BlockDevice
```

```
mkfs -t gfs2 -p lock_dlm -t LockTableName -j NumberJournals BlockDevice
```

Weitere Informationen über das Erstellen von GFS2-Dateisystemen finden Sie in [Abschnitt 3.1, „Erstellung eines Dateisystems“](#).

3. Hängen Sie auf jedem Knoten die GFS2-Dateisysteme ein. Weitere Informationen über das Einhängen von GFS2-Dateisystemen finden Sie in [Abschnitt 3.2, „Einhängen eines Dateisystems“](#).

Verwendung der Befehle:

```
mount BlockDevice MountPoint
```

```
mount -o acl BlockDevice MountPoint
```

Die Einhängeoption **-o acl** erlaubt die Veränderung der Datei-ACLs. Wird ein Dateisystem ohne die Einhängeoption **-o acl** eingehängt, können Benutzer die ACLs zwar einsehen (mittels **getfacl**), dürfen diese jedoch nicht verändern (mittels **setfacl**).



### Anmerkung

Sie können die im Red Hat Hochverfügbarkeits-Add-On enthaltenen **init.d**-Skripts verwenden, um das Ein- und Aushängen von GFS2-Dateisystemen zu automatisieren.

## 2.3. Implementierung eines GFS2-Clusters

Die Implementierung eines Cluster-Dateisystems stellt keinen sofortigen Ersatz für eine Ein-Knoten-Implementierung dar. Wir empfehlen Ihnen, etwa 8-12 Wochen zum Testen der neuen Installationen einzuplanen, um das System gründlich zu testen und sicherzustellen, dass es die benötigte Leistung bringt. Während dieser Zeit können jegliche Probleme mit der Leistung oder der Funktionalität beseitigt werden, und für auftretende Fragen sollten Sie sich mit dem Red Hat Support-Team in Verbindung setzen. Wir empfehlen es Kunden außerdem in Erwägung zu ziehen, Cluster zu implementieren, deren Konfiguration vor dem Einsatz vom Red Hat Support überprüft wurde, um spätere Support-Probleme zu vermeiden.

## Kapitel 3. Verwaltung von GFS2

Dieses Kapitel beschreibt die Aufgaben und Befehle zur Verwaltung von GFS2 und besteht aus folgenden Abschnitten:

- ▶ [Abschnitt 3.1, „Erstellung eines Dateisystems“](#)
- ▶ [Abschnitt 3.2, „Einhängen eines Dateisystems“](#)
- ▶ [Abschnitt 3.3, „Aushängen eines Dateisystems“](#)
- ▶ [Abschnitt 3.5, „Verwaltung von GFS2-Festplattenkontingenten“](#)
- ▶ [Abschnitt 3.6, „Vergrößern eines Dateisystems“](#)
- ▶ [Abschnitt 3.7, „Hinzufügen von Journalen zu einem Dateisystem“](#)
- ▶ [Abschnitt 3.8, „Datenjournale“](#)
- ▶ [Abschnitt 3.9, „Konfiguration der `atime`-Aktualisierung“](#)
- ▶ [Abschnitt 3.10, „Unterbrechung der Aktivität eines Dateisystems“](#)
- ▶ [Abschnitt 3.11, „Reparieren eines Dateisystems“](#)
- ▶ [Abschnitt 3.12, „Bind Mounts und kontextabhängige Pfade“](#)
- ▶ [Abschnitt 3.13, „Einhänge-Reihenfolge für Bind Mounts und Dateisysteme“](#)
- ▶ [Abschnitt 3.14, „Die GFS2-Rückzugfunktion“](#)

### 3.1. Erstellung eines Dateisystems

Sie erstellen ein GFS2-Dateisystem mit dem `mkfs.gfs2`-Befehl. Sie können dazu auch den `mkfs`-Befehl mit der `-t gfs2`-Option verwenden. Ein Dateisystem wird auf einem aktivierten LVM-Datenträger erstellt. Folgende Informationen sind erforderlich, um den `mkfs.gfs2`-Befehl auszuführen:

- ▶ Sperrprotokoll/Modulname (das Sperrprotokoll für einen Cluster ist `lock_dlm`)
- ▶ Clustername (falls es als ein Teil einer Cluster-Konfiguration läuft)
- ▶ Anzahl der Journale (es ist jeweils ein Journal erforderlich für jeden Knoten, der evtl. das Dateisystem einhängen soll)

Beim Erstellen eines GFS2-Dateisystems können Sie den `mkfs.gfs2`-Befehl direkt verwenden, oder Sie können den `mkfs`-Befehl mit dem `-t`-Parameter verwenden und ein Dateisystem vom Typ `gfs2` spezifizieren, gefolgt von den GFS2-Dateisystemoptionen.



#### Anmerkung

Sobald Sie mit dem `mkfs.gfs2`-Befehl ein GFS2-Dateisystem erstellt haben, können Sie die Größe dieses Dateisystems nicht mehr nach unten korrigieren. Sie können ein vorhandenes Dateisystem jedoch mithilfe des `gfs2_grow`-Befehls vergrößern, wie in [Abschnitt 3.6, „Vergrößern eines Dateisystems“](#) beschrieben.

### Verwendung

Beim Erstellen eines geclusterten GFS2-Dateisystems können Sie eines der folgenden Formate verwenden:

```
mkfs.gfs2 -p LockProtoName -t LockTableName -j NumberJournals BlockDevice
```

```
mkfs -t gfs2 -p LockProtoName -t LockTableName -j NumberJournals BlockDevice
```

Beim Erstellen eines lokalen GFS2-Dateisystems können Sie eines der folgenden Formate verwenden:



### Anmerkung

Für die Red Hat Enterprise Linux 6 Release unterstützt Red Hat nicht länger den Einsatz von GFS2 als Ein-Knoten-Dateisystem.

```
mkfs.gfs2 -p LockProtoName -j NumberJournals BlockDevice
```

```
mkfs -t gfs2 -p LockProtoName -j NumberJournals BlockDevice
```



### Warnung

Sie sollten sich unbedingt mit der Verwendung der ***LockProtoName*** und ***LockTableName*** Parameter vertraut machen. Unsachgemäße Verwendung der ***LockProtoName*** und ***LockTableName*** Parameter kann zur Beschädigung des Dateisystems oder des Lock Space führen.

#### ***LockProtoName***

Spezifiziert den Namen des zu verwendenden Sperrprotokolls. Das Sperrprotokoll für einen Cluster ist ***lock\_dlm***.

#### ***LockTableName***

Dieser Parameter wird für ein GFS2-Dateisystem in einer Cluster-Konfiguration spezifiziert. Er besteht aus zwei Teilen, durch einen Doppelpunkt voneinander getrennt (ohne Leerstellen), also: ***ClusterName:FSName***

- ▶ ***ClusterName***, der Name des Clusters, für den das GFS2-Dateisystem erstellt wird.
- ▶ ***FSName***, der Dateisystemname, darf zwischen 1 und 16 Zeichen lang sein. Der Name muss eindeutig für alle ***lock\_dlm***-Dateisysteme im Cluster sein und für alle Dateisysteme (***lock\_dlm*** und ***lock\_nolock***) auf jedem lokalen Knoten.

#### ***Number***

Spezifiziert die Anzahl der Journale, die vom ***mkfs.gfs2***-Befehl erzeugt werden sollen. Es wird jeweils ein Journal für jeden Knoten benötigt, der das Dateisystem einhängt. Für GFS2-Dateisysteme können später mehr Journale hinzugefügt werden, ohne dass das Dateisystem vergrößert werden muss, wie im [Abschnitt 3.7, „Hinzufügen von Journalen zu einem Dateisystem“](#) beschrieben.

#### ***BlockDevice***

Spezifiziert einen logischen oder physischen Datenträger.

## Beispiele

In diesen Beispielen ist **lock\_dlm** das vom Dateisystem verwendete Sperrprotokoll, da es sich hierbei um ein geclustertes Dateisystem handelt. Der Clustername ist **alpha** und der Dateisystemname ist **mydata1**. Das Dateisystem beinhaltet 8 Journale und wird auf **/dev/vg01/lvol0** erstellt.

```
mkfs.gfs2 -p lock_dlm -t alpha:mydata1 -j 8 /dev/vg01/lvol0
```

```
mkfs -t gfs2 -p lock_dlm -t alpha:mydata1 -j 8 /dev/vg01/lvol0
```

In diesen Beispielen wird ein zweites **lock\_dlm**-Dateisystem erstellt, welches im Cluster **alpha** verwendet werden kann. Der Dateisystemname lautet **mydata2**. Das Dateisystem beinhaltet 8 Journale und wird auf **/dev/vg01/lvol1** erstellt.

```
mkfs.gfs2 -p lock_dlm -t alpha:mydata2 -j 8 /dev/vg01/lvol1
```

```
mkfs -t gfs2 -p lock_dlm -t alpha:mydata2 -j 8 /dev/vg01/lvol1
```

## Vollständige Optionen

[Tabelle 3.1, „Befehloptionen: mkfs.gfs2“](#) beschreibt die **mkfs.gfs2**-Befehloptionen (Flags und Parameter).

**Tabelle 3.1. Befehlsoptionen: mkfs.gfs2**

| Flag | Parameter            | Beschreibung   |
|------|----------------------|--|
| -c   | <i>Megabytes</i>     | Legt die anfängliche Größe der Kontingentänderungsdatei eines jeden Journals auf <b>Megabytes</b> fest.  |
| -D   |                      | Aktiviert Debugging-Ausgabe.   |
| -h   |                      | Hilfe. Zeigt verfügbare Optionen an.   |
| -J   | <i>MegaBytes</i>     | Spezifiziert die Größe des Journals in Megabytes. Die standardmäßige Journalgröße beträgt 128 Megabytes, die minimale Größe ist 8 Megabytes. Größere Journale verbessern die Leistung, allerdings verbrauchen sie mehr Speicher als kleinere Journale.   |
| -j   | <i>Number</i>        | Spezifiziert die Anzahl der Journale, die vom <b>mkfs.gfs2</b> -Befehl erzeugt werden sollen. Es wird jeweils ein Journal für jeden Knoten benötigt, der das Dateisystem einhängt. Wird diese Option nicht spezifiziert, wird ein Journal angelegt. Für GFS2-Dateisysteme können später mehr Journale hinzugefügt werden, ohne dass das Dateisystem vergrößert werden muss.  |
| -O   |                      | Verhindert, dass der <b>mkfs.gfs2</b> -Befehl nach einer Bestätigung fragt, bevor das Dateisystem geschrieben wird.  |
| -p   | <i>LockProtoName</i> | Spezifiziert den Namen des zu verwendenden Sperrprotokolls. Anerkannte Sperrprotokolle sind u.a.:<br><br><b>lock_dlm</b> — das standardmäßige Sperrmodul, erforderlich für ein geclustertes Dateisystem.<br><br><b>lock_nolock</b> — wird verwendet, wenn GFS2 als lokales Dateisystem fungiert (auf nur einem Knoten).  |
| -q   |                      | Es wird keinerlei Ausgabe angezeigt.   |
| -r   | <i>MegaBytes</i>     | Spezifiziert die Größe der Ressourcengruppen in Megabytes. Die minimale Ressourcengruppengröße beträgt 32 MB, die maximale Ressourcengruppengröße 2048 MB. Eine große Ressourcengruppengröße kann unter Umständen die Leistung von sehr großen Dateisystemen steigern. Falls dies nicht spezifiziert wird, wählt <b>mkfs.gfs2</b> die Ressourcengruppengröße auf Basis der Dateisystemgröße aus: Durchschnittlich große Dateisysteme erhalten 256 MB große Ressourcengruppen, und größere Dateisysteme erhalten für eine bessere Leistung größere Ressourcengruppen. |
| -t   | <i>LockTableName</i> | Ein eindeutiger Bezeichner, der das Sperrtabellenfeld spezifiziert, wenn Sie das <b>lock_dlm</b> -Protokoll verwenden; das <b>lock_nolock</b> -Protokoll verwendet diesen Parameter nicht.<br><br>Dieser Parameter besteht aus zwei Teilen, durch einen  |

Doppelpunkt voneinander getrennt (keine Leerstellen), also: **ClusterName:FSName**.

**ClusterName** ist der Name des Clusters, für den das GFS2-Dateisystem erzeugt wird; nur Mitgliedern des Clusters ist es erlaubt, dieses Dateisystem zu benutzen. Der Clustername wird in der **/etc/cluster/cluster.conf**-Datei mithilfe des **Cluster Configuration Tool** festgelegt, und wird im **Cluster Status Tool** in der Cluster-Verwaltungsoberfläche der Red Hat Cluster Suite angezeigt.

**FSName**, der Dateisystemname, darf 1 bis 16 Zeichen lang sein. Der Name muss unter all den Dateisystemen im Cluster eindeutig sein.

|           |                  |   |
|-----------|------------------|---|
| <b>-u</b> | <b>MegaBytes</b> | Spezifiziert die ursprüngliche Größe der nicht verlinkten Tag-Datei eines jeden Journals. |
| <b>-V</b> |                  | Zeigt Befehlsversionsinformationen an.  |

## 3.2. Einhängen eines Dateisystems

Bevor Sie ein GFS2-Dateisystem einhängen können, muss das Dateisystem existieren (siehe [Abschnitt 3.1, „Erstellung eines Dateisystems“](#)), der Datenträger, auf dem das Dateisystem existiert, muss aktiviert sein und das unterstützende Clustering- und Sperrsystem muss gestartet sein (siehe *Konfiguration und Verwaltung eines Red Hat Clusters*). Nachdem dies sichergestellt wurde, können Sie das GFS2-Dateisystem genauso einhängen, wie Sie es von jedem anderen Linux-Dateisystem gewohnt sind.



### Anmerkung

Sollten Sie versuchen, ein GFS2-Dateisystem einzuhängen, wenn der Cluster Manager (**cman**) nicht gestartet wurde, wird die folgende Fehlermeldung ausgegeben:

```
[root@gfs-a24c-01 ~]# mount -t gfs2 -o noatime /dev/mapper/mpathap1 /mnt
gfs_control join connect error: Connection refused
error mounting lockproto lock_dlm
```

Um die Datei-ACLs zu verändern, müssen Sie das Dateisystem mit der **-o acl** Einhängeloption einhängen. Wird ein Dateisystem ohne die Einhängeloption **-o acl** eingehängt, können Benutzer die ACLs zwar einsehen (mittels **getfacl**), dürfen diese jedoch nicht verändern (mittels **setfacl**).

## Verwendung

### Einhängen ohne ACL-Veränderung

```
mount BlockDevice MountPoint
```

## Einhängen mit ACL-Veränderung

```
mount -o acl BlockDevice MountPoint
```

### **-o acl**

GFS2-spezifische Option, um Veränderungen an Datei-ACLs zu erlauben.

### ***BlockDevice***

Spezifiziert das Blockgerät, auf dem sich das GFS2-Dateisystem befindet.

### ***MountPoint***

Spezifiziert das Verzeichnis, in dem das GFS2-Dateisystem eingehängt werden soll.

## Beispiel

In diesem Beispiel ist das GFS2-Dateisystem auf `/dev/vg01/lvol0` im `/mygfs2`-Verzeichnis eingehängt.

```
mount /dev/vg01/lvol0 /mygfs2
```

## Vollständige Verwendung

```
mount BlockDevice MountPoint -o option
```

Der Parameter **-o option** besteht aus GFS2-spezifischen Optionen (siehe [Tabelle 3.2, „GFS2-spezifische Einhängeoptionen“](#)), oder akzeptablen Standard-Linux **mount -o** Optionen, oder einer Kombination aus beiden. Mehrere **option**-Parameter werden durch ein Komma getrennt ohne Leerzeichen.



### Anmerkung

Der **mount**-Befehl ist ein Linux-Systembefehl. Zusätzlich zu den in diesem Abschnitt beschriebenen GFS2-spezifischen Optionen können Sie andere standardmäßige **mount** Befehlsoptionen verwenden (z.B. **-r**). Weitere Informationen über andere **mount**-Befehlsoptionen für Linux finden Sie auf der **mount**-Handbuchseite.

[Tabelle 3.2, „GFS2-spezifische Einhängeoptionen“](#) beschreibt die verfügbaren GFS2-spezifischen **-o option** Werte, die zum Zeitpunkt des Einhängens an GFS2 übergeben werden können.



### Anmerkung

Diese Tabelle enthält Beschreibungen der Optionen, die ausschließlich mit lokalen Dateisystemen verwendet werden. Beachten Sie jedoch, dass Red Hat für die Red Hat Enterprise Linux 6 Release den Einsatz von GFS2 nicht für ein Ein-Knoten-System unterstützt. Red Hat unterstützt auch weiterhin Ein-Knoten-GFS2-Dateisysteme zum Einhängen von Snapshots von Cluster-Dateisystemen (z.B. zwecks Datensicherung).



Tabelle 3.2. GFS2-spezifische Einhängoptionen

| Option  | Beschreibung  |
|---|---|
| <b>acl</b>  | Erlaubt das Verändern von Datei-ACLs. Wird ein Dateisystem ohne die Einhängoption <b>-o acl</b> eingehängt, können Benutzer die ACLs zwar einsehen (mittels <b>getfacl</b> ), dürfen diese jedoch nicht verändern (mittels <b>setfacl</b> ).  |
| <b>data=[ordered writeback]</b>   | Wenn <b>data=ordered</b> gesetzt ist, werden die Benutzerdaten, die von einer Transaktion verändert wurden, auf die Festplatte gespeichert, bevor die Transaktion auf die Festplatte festgeschrieben wird. Dies sollte verhindern, dass der Benutzer nach einem Absturz in einer Datei nicht initialisierte Blöcke sieht. Wenn <b>data=writeback</b> gesetzt ist, werden die Benutzerdaten zu einem beliebigen Zeitpunkt nach deren Änderung auf die Festplatte geschrieben. Dies garantiert zwar nicht dieselbe Konsistenz wie der <b>ordered</b> -Modus, sollte aber bei einer gewissen Auslastung etwas schneller sein. Standardmäßig ist der <b>ordered</b> -Modus gesetzt. |
| <b>ignore_local_fs</b><br><br><b>Vorsicht:</b> Diese Option sollte <i>nicht</i> eingesetzt werden, wenn GFS2-Dateisysteme gemeinsam verwendet werden. | Zwingt GFS2 dazu, das Dateisystem als Multihost-Dateisystem zu behandeln. Standardmäßig schaltet <b>lock_nolock</b> das <b>locallocks</b> -Flag an.   |
| <b>locallocks</b><br><br><b>Vorsicht:</b> Diese Option sollte nicht eingesetzt werden, wenn GFS2-Dateisysteme gemeinsam verwendet werden.             | Weist GFS2 an, dass die VFS (Virtual File System) Schicht alle <b>flocks</b> und <b>fcntl</b> ausführen soll. Das <b>locallocks</b> -Flag wird automatisch durch <b>lock_nolock</b> eingeschaltet.  |
| <b>lockproto=LockModuleName</b>   | Erlaubt es dem Benutzer zu spezifizieren, welches Sperrprotokoll mit dem Dateisystem benutzt werden soll. Falls <b>LockModuleName</b> nicht spezifiziert wurde, wird der Name des Sperrprotokolls vom Dateisystem-Superblock gelesen.   |
| <b>locktable=LockTableName</b>  | Erlaubt es dem Benutzer zu spezifizieren, welche Sperrtabelle mit dem Dateisystem benutzt werden soll.  |
| <b>quota=[off/account/on]</b>   | Aktiviert oder deaktiviert Festplattenkontingente für das Dateisystem. Werden die Festplattenkontingente in den <b>account</b> -Status gesetzt, führt das dazu, dass die Verbrauchsstatistiken pro UID/GID korrekt vom Dateisystem gepflegt werden; Grenzen und Warnwerte werden ignoriert. Der Standardwert ist <b>off</b> .   |
| <b>errors=panic withdraw</b>  | Wenn <b>errors=panic</b> spezifiziert ist, führen Fehler im Dateisystem zu einer Kernel-Panic. Gemäß Standardverhalten (was der Einstellung von <b>errors=withdraw</b> gleichkommt) zieht sich das System aus dem Dateisystem zurück und macht es bis zum nächsten Neustart nicht erreichbar; in einigen  |

|                             |  |
|-----------------------------|--|
|                             | Fällen kann das System weiterhin ausgeführt werden. Weitere Informationen über die GFS2-Rückzugfunktion finden Sie in <a href="#">Abschnitt 3.14 „Die GFS2-Rückzugfunktion“</a> .  |
| <b>discard/nodiscard</b>    | Veranlasst GFS2 dazu, "discard" I/O-Anfragen für Blöcke zu generieren, die freigegeben wurden. Von geeigneter Hardware kann dies dazu eingesetzt werden, um "Thin Provisioning" und ähnliche Schemata zu implementieren.   |
| <b>barrier/nobarrier</b>    | Veranlasst GFS2 dazu, I/O-Sperren zu senden, wenn das Journal auf die Festplatte übertragen wird. Der Standardwert lautet <b>on</b> . Diese Option wird automatisch auf <b>off</b> gesetzt, wenn das zugrunde liegende Gerät keine I/O-Sperren unterstützt. Die Verwendung von I/O-Sperren mit GFS2 wird grundsätzlich dringend empfohlen, es sei denn, das Blockgerät ist derart aufgebaut, dass es die Inhalte seines Schreib-Caches nicht verlieren kann (z.B. falls es an eine unterbrechungsfreie Stromversorgung angeschlossen ist oder über keinen Schreib-Cache verfügt).  |
| <b>quota_quantum=secs</b>   | Legt die Anzahl an Sekunden fest, für die eine Änderung an den Informationen der Festplattenkontingente auf einem Knoten verbleiben darf, bevor diese in die Kontingentdatei übertragen wird. Dies ist die bevorzugte Methode zum Einstellen dieses Parameters. Der Parameter ist ein ganzzahliger Wert größer als Null. Der Standard beträgt 60 Sekunden. Kleinere Werte führen zu schnelleren Aktualisierungen der Kontingentinformationen und einer geringeren Wahrscheinlichkeit, dass Benutzer oder Gruppen ihre Kontingente überschreiten. Größere Werte machen Dateisystemoperationen im Zusammenhang mit Kontingenten schneller und effizienter. |
| <b>statfs_quantum=secs</b>  | Das Einstellen von <b>statfs_quantum</b> auf 0 ist die bevorzugte Methode zum Festlegen der langsamen Version von <b>statfs</b> . Der Standardwert beträgt 30 Sekunden, was die maximale Zeit festlegt, bevor <b>statfs</b> -Änderungen mit der Master- <b>statfs</b> -Datei synchronisiert werden. Dies kann angepasst werden, um schnellere, weniger genaue <b>statfs</b> -Werte oder langsamere, genauere Werte zu ermöglichen. Wenn diese Option auf 0 gesetzt ist, wird <b>statfs</b> immer die echten Werte berichten.   |
| <b>statfs_percent=value</b> | Implementiert eine maximale Grenze prozentualer Änderungen an den <b>statfs</b> -Informationen auf lokaler Basis, bevor diese mit der Master- <b>statfs</b> -Datei synchronisiert werden, selbst wenn die festgelegte Zeitspanne noch nicht überschritten wurde. Falls die <b>statfs_quantum</b> -Einstellung 0 lautet, wird diese Einstellung ignoriert.  |

### 3.3. Aushängen eines Dateisystems

Das GFS2-Dateisystem kann auf dieselbe Art und Weise ausgehängt werden, wie jedes andere Linux-Dateisystem auch — mithilfe des **umount**-Befehls.



### Anmerkung

Der **umount**-Befehl ist ein Linux-Systembefehl. Informationen zu diesem Befehl finden Sie auf der Handbuchseite des **umount**-Befehls.

## Verwendung

```
umount MountPoint
```

### *MountPoint*

Spezifiziert das Verzeichnis, in dem das GFS2-Dateisystem derzeit eingehängt ist.

## 3.4. Spezielle Überlegungen zum Einhängen von GFS2-Dateisystemen

GFS2-Dateisysteme, die manuell eingehängt wurden, statt automatisch durch einen Eintrag in der **fstab**-Datei, sind dem System nicht bekannt, wenn dieses beim Herunterfahren die Dateisysteme aushängt. Infolgedessen wird das GFS2-Skript das GFS2-Dateisystem nicht aushängen. Nachdem das GFS2-Shutdown-Skript ausgeführt wurde, beendet der standardmäßige Shutdown-Prozess alle verbleibenden Benutzerprozesse, einschließlich der Cluster-Infrastruktur, und versucht das Dateisystem auszuhängen. Dieses Aushängen schlägt jedoch ohne die Cluster-Infrastruktur fehl, so dass sich das System aufhängt.

Um ein Aufhängen des Systems zu verhindern, wenn die GFS2-Dateisysteme ausgehängt werden, haben Sie zwei Möglichkeiten:

- Verwenden Sie grundsätzlich einen Eintrag in der **fstab**-Datei, um das GFS2-Dateisystem einzuhängen.
- Falls ein GFS2-Dateisystem manuell mit dem **mount**-Befehl eingehängt wurde, hängen Sie dies auch manuell wieder mit dem **umount**-Befehl aus, bevor Sie einen Neustart durchführen oder das System herunterfahren.

Falls sich Ihr Dateisystem beim Aushängen während des System-Shutdowns unter diesen Umständen aufhängt, führen Sie einen Hardware-Neustart aus. Es ist unwahrscheinlich, dass Daten verloren gehen, da das Dateisystem zu einem früheren Zeitpunkt im Shutdown-Vorgang synchronisiert wurde.

## 3.5. Verwaltung von GFS2-Festplattenkontingenten

Dateisystemkontingente werden dazu verwendet, um die Menge des Dateisystemspeichers zu begrenzen, den ein Benutzer oder eine Gruppe verbrauchen darf. Ein Benutzer oder eine Gruppe hat standardmäßig kein begrenztes Kontingent, bevor nicht eines gesetzt wird. Wenn ein GFS2-Dateisystem mit der **quota=on** oder der **quota=account**-Option eingehängt wird, verfolgt GFS2 den Speicherverbrauch von jedem Benutzer und jeder Gruppe nach, selbst wenn keine Grenzen gesetzt wurden. GFS2 aktualisiert die Kontingentinformationen auf transaktionale Weise, so dass nach einem Systemabsturz der Kontingentverbrauch nicht rekonstruiert werden muss.

Um nachteilige Auswirkungen auf die Leistung zu vermeiden, synchronisiert ein GFS2-Knoten die Aktualisierungen an der Kontingentsdatei nur periodisch. Diese demzufolge etwas ungenaue Nachverfolgung der Kontingente ermöglicht es Benutzern oder Gruppen unter Umständen, ihre gesetzten Grenzen leicht zu überschreiten. Um diesen Effekt zu minimieren, verkleinert GFS2 die zeitlichen Abstände der Synchronisation dynamisch, wenn sich die "harte" Kontingentgrenze allmählich nähert.



### Anmerkung

Ab der Red Hat Enterprise Linux 6.1 Release unterstützt GFS2 die standardmäßigen Linux-Funktionen für Festplattenkontingente. Um diese nutzen zu können, müssen Sie das **quota**-RPM installieren. Dies ist die bevorzugte Methode zur Verwaltung von Festplattenkontingenten auf GFS2 und sollte für alle neuen GFS2-Deployments, die Kontingente nutzen, verwendet werden. Dieser Abschnitt dokumentiert die GFS2-Kontingentverwaltung mithilfe dieser Funktionen. In früheren Releases von Red Hat Enterprise Linux erforderte GFS2 den **gfs2\_quota**-Befehl zur Verwaltung von Festplattenkontingenten. Werfen Sie für Informationen über die Verwendung des **gfs2\_quota**-Befehls bitte einen Blick auf [Anhang A, GFS2-Kontingentverwaltung mit dem gfs2\\_quota-Befehl](#).

## 3.5.1. Konfiguration von Festplattenkontingenten

Um Festplattenkontingente zu implementieren, führen Sie die folgenden Schritte aus:

1. Stellen Sie das Erzwingen oder das Berechnen von Kontingenten ein.
2. Initialisieren Sie die Kontingentdatenbankdatei mit den aktuellen Informationen zum Blockverbrauch.
3. Weisen Sie Kontingentrichtlinien hinzu. (Wurde lediglich der Berechnungsmodus eingestellt, werden diese Richtlinien jedoch nicht erzwungen.)

Jeder dieser Schritte wird in den nachfolgenden Abschnitten detailliert behandelt.

### 3.5.1.1. Einrichten von Kontingenten für das Erzwingen oder die Berechnung

In GFS2-Dateisystemen sind Kontingente standardmäßig deaktiviert. Um Kontingente für ein Dateisystem zu aktivieren, hängen Sie das Dateisystem mit der Option **quota=on** ein.

Es ist möglich, den Festplattenverbrauch nachzuverfolgen und Kontingente für alle Benutzer und Gruppen zu berechnen, ohne die Grenz- und Warnwerte zu erzwingen. Hängen Sie dazu das Dateisystem mit der Option **quota=account** ein.

## Verwendung

Um ein Dateisystem mit aktivierten Kontingenten einzuhängen, hängen Sie das Dateisystem mit der Option **quota=on** ein.

```
mount -o quota=on BlockDevice MountPoint
```

Um ein Dateisystem mit aktivierter Kontingentberechnung einzuhängen, wobei die Kontingentgrenzen jedoch nicht erzwungen werden, hängen Sie das Dateisystem mit der Option **quota=account** ein.

```
mount -o quota=account BlockDevice MountPoint
```

Um ein Dateisystem mit deaktivierten Kontingenten einzuhängen, hängen Sie das Dateisystem mit der

Option **quota=off** ein. Dies ist die Standardeinstellung.

```
mount -o quota=off BlockDevice MountPoint
```

### **quota={on|off|account}**

**on** - Spezifiziert, dass Kontingente aktiviert sind, wenn das Dateisystem eingehängt wird.

**off** - Spezifiziert, dass Kontingente deaktiviert sind, wenn das Dateisystem eingehängt wird.

**account** - Spezifiziert, dass die Benutzer- und Gruppenverbrauchsstatistik im Dateisystem gepflegt werden soll, selbst wenn die Kontingentgrenzen nicht erzwungen werden.

### **BlockDevice**

Spezifiziert das Blockgerät, auf dem sich das GFS2-Dateisystem befindet.

### **MountPoint**

Spezifiziert das Verzeichnis, in dem das GFS2-Dateisystem eingehängt werden soll.

## Beispiele

In diesem Beispiel ist das GFS2-Dateisystem **/dev/vg01/lvo10** im **/mygfs2**-Verzeichnis eingehängt mit aktivierten Kontingenten.

```
mount -o quota=on /dev/vg01/lvo10 /mygfs2
```

In diesem Beispiel ist das GFS2-Dateisystem **/dev/vg01/lvo10** im **/mygfs2**-Verzeichnis eingehängt mit Kontingentberechnung, jedoch ohne Erzwingen von Kontingentgrenzen.

```
mount -o quota=account /dev/vg01/lvo10 /mygfs2
```

### 3.5.1.2. Erstellen der Kontingent-Datenbankdateien

Nachdem jedes Dateisystem mit aktivierten Kontingenten eingehängt wurde, kann das System nun mit Kontingenten arbeiten. Allerdings ist das System selbst noch nicht dazu eingerichtet, Kontingente zu unterstützen. Der nächste Schritt ist daher das Ausführen des **quotacheck**-Befehls.

Der **quotacheck**-Befehl untersucht die Dateisysteme mit aktivierten Kontingenten und erzeugt eine Tabelle des aktuellen Festplattenverbrauchs pro Dateisystem. Die Tabelle wird dann verwendet, um für das Betriebssystem eine laufend aktualisierte Kopie des Verbrauchs zu pflegen. Zudem werden die Kontingentdateien des Dateisystems aktualisiert.

Um die Kontingentdateien auf dem Dateisystem zu erstellen, verwenden Sie die **-u** und **-g**-Optionen des **quotacheck**-Befehls; beide Optionen müssen angegeben werden, damit sowohl Benutzer- als auch Gruppenkontingente initialisiert werden. Wenn Kontingente beispielsweise auf dem **/home**-Dateisystem aktiviert sind, erstellen Sie die Dateien im **/home**-Verzeichnis:

```
quotacheck -ug /home
```

### 3.5.1.3. Kontingente pro Benutzer zuweisen

Weisen Sie abschließend die Festplattenkontingente mit dem **edquota**-Befehl zu. Beachten Sie, dass Kontingente nicht erzwungen werden, wenn Sie Ihr Dateisystem nur im Berechnungsmodus eingehängt haben (durch Angabe der **quota=account**-Option).

Um ein Festplattenkontingent für einen Benutzer zu konfigurieren, führen Sie als Root in einer Shell den folgenden Befehl aus:

```
edquota username
```

Führen Sie diesen Schritt für jeden Benutzer durch, dem ein Kontingent zugewiesen werden soll. Wenn beispielsweise in der **/etc/fstab**-Datei Kontingente für die **/home**-Partition (**/dev/Vo1Group00/LogVo102** im nachfolgenden Beispiel) aktiviert sind und Sie den **edquota testuser**-Befehl ausführen, sehen Sie Folgendes im Standard-Editor des Systems:

```
Disk quotas for user testuser (uid 501):
Filesystem          blocks      soft      hard    inodes    soft      hard
/dev/Vo1Group00/LogVo102  440436      0         0
```



### Anmerkung

Der **edquota**-Befehl nutzt den Texteditor, der durch die **EDITOR**-Umgebungsvariable definiert wurde. Um den Editor zu ändern, setzen Sie die **EDITOR**-Umgebungsvariable in Ihrer **~/.bash\_profile**-Datei auf den vollständigen Pfad zum Editor Ihrer Wahl.

Die erste Spalte zeigt den Namen des Dateisystems, auf dem Kontingente aktiviert wurden. Die zweite Spalte zeigt, wie viele Blöcke der Benutzer derzeit verbraucht. Die nächsten zwei Spalten legen die weichen und harten Blockgrenzen für den Benutzer auf dem Dateisystem fest.

Die weiche Blockgrenze definiert den maximalen Speicherplatz, der verbraucht werden kann.

Die harte Grenze definiert den absolut maximalen Speicherplatz, den ein Benutzer oder eine Gruppe verwenden kann. Sobald diese Grenze erreicht wurde, kann kein weiterer Festplattenplatz verbraucht werden.

Das GFS2-Dateisystem pflegt keine Kontingente für Inodes, diese Spalten sind für GFS2-Dateisysteme demnach irrelevant und bleiben leer.

Falls einer der Werte auf 0 gesetzt ist, wurde diese Grenze nicht gesetzt. Ändern Sie im Texteditor die Grenzen auf die gewünschten Werte. Zum Beispiel:

```
Disk quotas for user testuser (uid 501):
Filesystem          blocks      soft      hard    inodes    soft      hard
/dev/Vo1Group00/LogVo102  440436    500000    550000
```

Um zu überprüfen, ob das Kontingent wie gewünscht für den Benutzer festgelegt wurde, führen Sie den folgenden Befehl aus:

```
quota testuser
```

#### 3.5.1.4. Kontingente pro Gruppe zuweisen

Kontingente können auch pro Gruppe zugewiesen werden. Beachten Sie, dass Kontingente nicht erzwungen werden, wenn Sie Ihr Dateisystem nur im Berechnungsmodus eingehängt haben (durch Angabe der **account=on**-Option).

Um ein Gruppenkontingent für die **devel**-Gruppe festzulegen (die Gruppe muss bereits existieren, ehe das Gruppenkontingent festgelegt werden kann), verwenden Sie den folgenden Befehl:

```
edquota -g devel
```

Dieser Befehl zeigt die vorhandenen Kontingente für die Gruppe im Texteditor:

```
Disk quotas for group devel (gid 505):
Filesystem          blocks      soft      hard      inodes     soft      hard
/dev/VolGroup00/LogVol102  440400      0          0
```

Das GFS2-Dateisystem pflegt keine Kontingente für Inodes, diese Spalten sind für GFS2-Dateisysteme demnach irrelevant und bleiben leer. Verändern Sie die Grenzen wie gewünscht und speichern Sie die Datei.

Um zu überprüfen, ob das Kontingent wie gewünscht für die Gruppe festgelegt wurde, führen Sie den folgenden Befehl aus:

```
quota -g devel
```

### 3.5.2. Verwaltung von Festplattenkontingenten

Wenn Kontingente implementiert sind, erfordern diese ein gewisses Maß an Pflege — hauptsächlich sollte überwacht werden, ob Kontingente überschritten werden und ob die gesetzten Grenzen noch angemessen sind.

Sollten Benutzer wiederholt ihre Kontingentgrenzen ausreizen oder dauerhaft die weichen Grenzen überschreiten, muss der Systemadministrator abhängig von der Art des Benutzers und der Auswirkungen des begrenzten Speicherplatzes auf ihre Arbeit die Entscheidung treffen, wie vorzugehen ist. Der Administrator kann dem Benutzer entweder dabei helfen, seinen Speicherplatzverbrauch zu verringern, oder er kann das Festplattenkontingent des Benutzers erhöhen.

Sie können mithilfe des **repquota**-Befehls einen Bericht des Festplattenverbrauchs generieren. Beispielsweise erzeugt der Befehl **repquota /home** die folgende Ausgabe:

```
*** Report for user quotas on device /dev/mapper/VolGroup00-LogVol102
Block grace time: 7days; Inode grace time: 7days
  Block limits  File limits
User  used soft hard  grace used soft hard  grace
-----
root   --    36    0    0          4    0    0
kristin --   540    0    0         125    0    0
testuser -- 440400 500000 550000    37418    0    0
```

Um einen Bericht zum Festplattenverbrauch für alle (Option **-a**) Dateisysteme mit aktivierten Kontingenten zu erhalten, führen Sie den folgenden Befehl aus:

```
repquota -a
```

Der Bericht ist zwar recht einfach verständlich, aber einige Punkte sollten dennoch erläutert werden. Die Zeichen **--** nach jedem Benutzer bieten einen schnellen Weg um festzustellen, ob die Blockgrenzen überschritten wurden. Falls die weiche Blockgrenze überschritten wurde, erscheint ein **+** anstelle des ersten **-** in der Ausgabe. Das zweite **-** repräsentiert die Inode-Grenze; da GFS2-Dateisysteme jedoch keine Inode-Grenzen unterstützen, bleibt an dieser Stelle stets das Zeichen **-**. GFS2-Dateisysteme unterstützen keine Schonfristen, die **grace**-Spalte bleibt demzufolge leer.

Beachten Sie, dass der **repquota**-Befehl nicht über NFS unterstützt wird, unabhängig vom zugrunde liegenden Dateisystem.

### 3.5.3. Genauigkeit von Kontingenten pflegen

Falls Sie Kontingente auf Ihrem Dateisystem aktivieren, nachdem Sie einige Zeit mit deaktivierten Kontingenten gearbeitet haben, sollten Sie den **quotacheck**-Befehl ausführen, um die Kontingentdateien zu erstellen, zu überprüfen und zu reparieren. Außerdem sollten Sie den **quotacheck**-Befehl ausführen, wenn Sie den Eindruck haben, dass Ihre Kontingentdateien nicht genau sind, was in Situationen der Fall sein kann, in denen ein Dateisystem nach einem Systemabsturz nicht ordnungsgemäß ausgehängt wurde.

Weitere Informationen über den **quotacheck**-Befehl finden Sie auf der **quotacheck**-Handbuchseite.



#### Anmerkung

Führen Sie **quotacheck** zu einem Zeitpunkt aus, an dem das Dateisystem auf allen Knoten relativ untätig ist, da Festplattenaktivität andernfalls die Berechnung der Kontingentwerte stören kann.

### 3.5.4. Synchronisation von Kontingenten mit dem **quotasync**-Befehl

GFS2 speichert alle Kontingentinformationen in seiner eigenen, internen Datei auf der Festplatte. Ein GFS2-Knoten aktualisiert diese Kontingentdatei nicht bei jedem Schreibvorgang im Dateisystem, sondern aktualisiert sie standardmäßig nur alle 60 Sekunden. Dies ist notwendig, um Konflikte zwischen Knoten zu vermeiden, die in die Kontingentdatei schreiben, was sich andernfalls nachteilig auf die Leistung auswirken würde.

Wenn ein Benutzer oder eine Gruppe sich seiner bzw. ihrer Kontingentgrenze nähert, verkürzt GFS2 dynamisch die zeitlichen Abstände zwischen den Aktualisierungen der Kontingentdatei, um das Überschreiten der Grenze zu verhindern. Der normale Zeitabstand zwischen Kontingentsynchronisationen ist ein anpassbarer Parameter, **quota\_quantum**, und kann mithilfe des **gfs2\_tool**-Befehls geändert werden. Der Zeitabstand ist standardmäßig auf 60 Sekunden gesetzt. Ebenso muss der **quota\_quantum**-Parameter auf jedem Knoten und jedes Mal, wenn das Dateisystem eingehängt wird, gesetzt werden. (Änderungen am **quota\_quantum**-Parameter sind nicht über Aus- und Einhängvorgänge hinweg persistent.)

Sie können den **quotasync**-Befehl verwenden, um zwischen den automatischen Aktualisierungen durch GFS2 die Kontingentinformationen manuell von einem Knoten auf die Kontingentdatei auf der Festplatte zu synchronisieren.

## Verwendung

### Synchronisation der Kontingentinformationen

```
quotasync [-ug] -a|mntpnt...
```

**u**

Synchronisiert die Benutzer-Kontingentdateien.

**g**

Synchronisiert die Gruppen-Kontingentdateien.

**a**

Synchronisiert alle Dateisysteme, die derzeit Kontingente aktiviert haben und die Synchronisation unterstützen. Wenn `-a` nicht angegeben ist, sollte ein Einhängpunkt im Dateisystem spezifiziert werden.

***mntpnt***

Spezifiziert das GFS2-Dateisystem, auf dem diese Aktion durchzuführen ist.

## Anpassen der Zeitabstände zwischen Synchronisationen

```
gfs2_tool settune MountPoint quota_quantum Seconds
```

***MountPoint***

Spezifiziert das GFS2-Dateisystem, auf dem diese Aktion durchzuführen ist.

***Seconds***

Spezifiziert den Zeitabstand zwischen regulären Synchronisationen der Kontingentdatei durch GFS2. Kleinere Werte können zu mehr Konflikten führen und sich nachteilig auf die Leistung auswirken.

## Beispiele

Dieses Beispiel synchronisiert alle gecachten, veränderten Kontingente von dem Knoten, auf dem der Befehl ausgeführt wird, auf das Dateisystem `/mnt/mygfs2`.

```
# quotasync -ug /mnt/mygfs2
```

Dieses Beispiel ändert den standardmäßigen Zeitabstand zwischen den regelmäßigen Aktualisierungen der Kontingentdatei auf eine Stunde (3600 Sekunden) für das Dateisystem `/mnt/mygfs2` auf einem einzigen Knoten.

```
gfs2_tool settune /mnt/mygfs2 quota_quantum 3600
```

### 3.5.5. Referenzen

Werfen Sie für weitere Informationen über Festplattenkontingente bitte einen Blick auf die `man`-Handbuchseiten der folgenden Befehle:

- ▶ `quotacheck`
- ▶ `edquota`
- ▶ `repquota`
- ▶ `quota`

## 3.6. Vergrößern eines Dateisystems

Der **gfs2\_grow**-Befehl wird zum Erweitern eines GFS2-Dateisystems verwendet, nachdem der Speicher, auf dem das Dateisystem liegt, erweitert wurde. Startet man den **gfs2\_grow**-Befehl auf einem existierenden GFS2-Dateisystem, wird sämtlicher freier Platz zwischen dem aktuellem Ende des Dateisystems und dem Ende des Geräts mit einer neu initialisierten GFS2-Dateisystemserweiterung gefüllt. Wenn der Füllvorgang abgeschlossen ist, wird der Ressourcenindex des Dateisystems aktualisiert. Alle Knoten im Cluster können dann den zusätzlich neu hinzugefügten Speicherplatz verwenden.

Der **gfs2\_grow** Befehl muss auf dem eingehängten Dateisystem ausgeführt werden, braucht jedoch nur auf einem Knoten im Cluster ausgeführt werden. Die anderen Knoten merken, dass eine Erweiterung stattgefunden hat und benutzen den neuen Speicherplatz automatisch.



### Anmerkung

Nachdem Sie ein GFS2-Dateisystem mit dem **mkfs.gfs2**-Befehl erstellt haben, können Sie es nicht mehr verkleinern.

## Verwendung

```
gfs2_grow MountPoint
```

### *MountPoint*

Spezifiziert das GFS2-Dateisystem, auf dem diese Aktion durchzuführen ist.

## Anmerkungen

Bevor Sie den **gfs2\_grow**-Befehl ausführen:

- ▶ Sichern Sie alle wichtigen Daten auf dem Dateisystem.
- ▶ Ermitteln Sie den Datenträger, der von dem zu erweiternden Dateisystem verwendet wird, indem Sie den Befehl **df *MountPoint*** ausführen.
- ▶ Erweitern Sie den zugrunde liegenden Cluster-Datenträger mit LVM. Mehr Informationen über die Verwaltung von LVM-Datenträgern finden Sie im Handbuch *Administration des Logical Volume Manager*.

Führen Sie nach dem **gfs2\_grow**-Befehl nun den **df**-Befehl aus, um zu überprüfen, ob der neue Speicher jetzt im Dateisystem verfügbar ist.

## Beispiele

In diesem Beispiel wird das Dateisystem auf dem **/mygfs2fs**-Verzeichnis erweitert.

```
[root@dash-01 ~]# gfs2_grow /mygfs2fs
FS: Mount Point: /mygfs2fs
FS: Device:      /dev/mapper/gfs2testvg-gfs2testlv
FS: Size:       524288 (0x80000)
FS: RG size:    65533 (0xffffd)
DEV: Size:      655360 (0xa0000)
The file system grew by 512MB.
gfs2_grow complete.
```

## Vollständige Verwendung

```
gfs2_grow [Options] {MountPoint | Device} [MountPoint | Device]
```

### *MountPoint*

Spezifiziert das Verzeichnis, in dem das GFS2-Dateisystem eingehängt ist.

### *Device*

Spezifiziert den Geräteknoten des Dateisystems.

[Tabelle 3.3, „Verfügbare GFS2-spezifische Optionen zum Erweitern eines Dateisystems“](#) beschreibt die GFS2-spezifischen Optionen, die Sie beim Erweitern eines GFS2-Dateisystems verwenden können.

**Tabelle 3.3. Verfügbare GFS2-spezifische Optionen zum Erweitern eines Dateisystems**

| Option              | Beschreibung  |
|---------------------|---|
| -h                  | Hilfe. Zeigt kurze Informationen zur Verwendung an.   |
| -q                  | Verringert die Ausführlichkeit der Ausgabe.   |
| -r <b>MegaBytes</b> | Spezifiziert die Größe der neuen Ressourcengruppe. Die standardmäßige Größe ist 256 MB.                                   |
| -T                  | Test. Führt sämtliche Berechnungen durch, schreibt jedoch keine Daten auf die Platte und erweitert das Dateisystem nicht. |
| -v                  | Zeigt Befehlsversionsinformationen an.  |

## 3.7. Hinzufügen von Journalen zu einem Dateisystem

Der **gfs2\_jadd**-Befehl wird dazu verwendet, um Journale zu einem GFS2-Dateisystem hinzuzufügen. Sie können jederzeit Journale zu einem GFS2-Dateisystem dynamisch hinzufügen, ohne den zugrunde liegenden logischen Datenträger zu erweitern. Der **gfs2\_jadd**-Befehl muss auf einem eingehängten Dateisystem ausgeführt werden, allerdings braucht er nur auf einem Knoten im Cluster ausgeführt zu werden. Alle anderen Knoten merken automatisch, dass eine Erweiterung stattgefunden hat.



### Anmerkung

Falls ein GFS2-Dateisystem voll ist, schlägt der **gfs2\_jadd**-Befehl fehl, selbst wenn der logische Datenträger, der das Dateisystem enthält, erweitert wurde und größer als das Dateisystem ist. Der Grund hierfür liegt in den Journalen eines GFS2-Dateisystems, denn bei den Journalen handelt es sich um einfache Dateien statt um eingebettete Metadaten, so dass ein einfaches Erweitern des zugrunde liegenden logischen Datenträgers nicht mehr Platz für die Journale verfügbar macht.

Bevor Sie Journale zu einem GFS-Dateisystem hinzufügen, können Sie die **journals**-Option des **gfs2\_tool**-Befehls verwenden um herauszufinden, wie viele Journale das GFS2-Dateisystem derzeit enthält. Das folgende Beispiel zeigt die Anzahl und die Größe der Journale im unter **/mnt/gfs2** eingehängten Dateisystem an.

```
[root@roth-01 ../cluster/gfs2]# gfs2_tool journals /mnt/gfs2
journal2 - 128MB
journal1 - 128MB
journal0 - 128MB
3 journal(s) found.
```

## Verwendung

```
gfs2_jadd -j Number MountPoint
```

### *Number*

Spezifiziert die Anzahl der neu hinzuzufügenden Journale.

### *MountPoint*

Spezifiziert das Verzeichnis, in dem das GFS2-Dateisystem eingehängt ist.

## Beispiele

In diesem Beispiel wird ein Journal zum Dateisystem im **/mygfs2** Verzeichnis hinzugefügt.

```
gfs2_jadd -j1 /mygfs2
```

In diesem Beispiel wurden zwei Journale zum Dateisystem im **/mygfs2** Verzeichnis hinzugefügt.

```
gfs2_jadd -j2 /mygfs2
```

## Vollständige Verwendung

```
gfs2_jadd [Options] {MountPoint | Device} [MountPoint | Device]
```

### *MountPoint*

Spezifiziert das Verzeichnis, in dem das GFS2-Dateisystem eingehängt ist.

### *Device*

Spezifiziert den Geräteknoten des Dateisystems.

[Tabelle 3.4, „GFS2-spezifische Optionen zum Hinzufügen von Journalen“](#) beschreibt die GFS2-spezifischen Optionen, die Sie beim Hinzufügen von Journalen zu einem GFS2-Dateisystem verwenden können.

**Tabelle 3.4. GFS2-spezifische Optionen zum Hinzufügen von Journalen**

| Flag      | Parameter        | Beschreibung  |
|-----------|------------------|---|
| <b>-h</b> |                  | Hilfe. Zeigt kurze Informationen zur Verwendung an.   |
| <b>-J</b> | <i>MegaBytes</i> | Spezifiziert die Größe der neuen Journale in Megabytes. Die standardmäßige Journalgröße ist 128 Megabytes. Die minimale Größe ist 32 Megabytes. Um Journale mit verschiedenen Größen zum Dateisystem hinzuzufügen, muss der <b>gfs2_jadd</b> -Befehl für jede gewünschte Journalgröße ausgeführt werden. Die angegebene Größe wird abgerundet, so dass es ein Vielfaches der Journalsegmentgröße ist, die bei der Erstellung des Dateisystems festgelegt wurde. |
| <b>-j</b> | <i>Number</i>    | Spezifiziert die Anzahl der neuen Journale, die vom <b>gfs2_jadd</b> -Befehl hinzugefügt werden. Der standardmäßige Wert ist 1.   |
| <b>-q</b> |                  | Verringert die Ausführlichkeit der Ausgabe.   |
| <b>-v</b> |                  | Zeigt Befehlsversionsinformationen an.  |

### 3.8. Datenjournale

Gewöhnlich schreibt GFS2 nur Metadaten in die Journale. Dateiinhalte werden nachträglich auf die Festplatte geschrieben, wenn der Kernel seine regelmäßige Synchronisation durchführt, die alle Dateisystempuffer auf die Platte schreibt. Ein **fsync()**-Aufruf auf einer Datei verursacht, dass die Dateidaten sofort auf die Platte geschrieben werden. Der Aufruf kehrt zurück, wenn die Platte meldet, dass sämtliche Daten sicher geschrieben wurden.

Das Pflegen von Datenjournalen kann die benötigte Zeit für **fsync()**-Operationen für sehr kleine Dateien verkürzen, da zusätzlich zu den Metadaten auch die Dateidaten in das Journal geschrieben werden. Dieser Vorteil verringert sich jedoch rapide bei größeren Dateien. Das Schreiben in normal große oder große Dateien ist sehr viel langsamer bei aktivierten Datenjournalen.

Applikationen, die auf **fsync()** zur Synchronisation von Dateidaten angewiesen sind, erzielen eine verbesserte Leistung durch die Verwendung der Daten-Journale. Das Daten-Journaling kann automatisch für jede GFS2-Datei aktiviert werden, die in einem entsprechend gekennzeichneten Verzeichnis (und all seinen Unterverzeichnissen) erstellt wird. Vorhandene Dateien mit Null-Länge können ebenfalls das Datenjournaling aktiviert oder deaktiviert haben.

Ist das Datenjournaling auf einem Verzeichnis aktiviert, wird das Verzeichnis auf "inherit jdata" gesetzt, was bedeutet, dass alle in diesem Verzeichnis erstellten Dateien und Unterverzeichnisse journalisiert werden. Sie können das Daten-Journaling auf einer Datei mithilfe des **chattr**-Befehls aktivieren bzw. deaktivieren.

Die folgenden Befehle aktivieren das Daten-Journaling auf der **/mnt/gfs2/gfs2\_dir/newfile**-Datei und überprüfen anschließend, ob das Flag ordnungsgemäß gesetzt wurde.

```
[root@roth-01 ~]# chattr +j /mnt/gfs2/gfs2_dir/newfile
[root@roth-01 ~]# lsattr /mnt/gfs2/gfs2_dir
-----j--- /mnt/gfs2/gfs2_dir/newfile
```

Die folgenden Befehle deaktivieren das Daten-Journaling auf der **/mnt/gfs2/gfs2\_dir/newfile**-Datei und überprüfen anschließend, ob das Flag ordnungsgemäß gesetzt wurde.

```
[root@roth-01 ~]# chattr -j /mnt/gfs2/gfs2_dir/newfile
[root@roth-01 ~]# lsattr /mnt/gfs2/gfs2_dir
-----j---- /mnt/gfs2/gfs2_dir/newfile
```

Sie können den **chattr**-Befehl auch dazu verwenden, um das **j**-Flag auf einem Verzeichnis zu setzen. Wenn Sie dieses Flag auf einem Verzeichnis setzen, werden alle nachfolgend in diesem Verzeichnis erstellten Dateien und Unterverzeichnisse journalisiert. Die folgenden Befehle setzen das **j**-Flag auf dem **gfs2\_dir**-Verzeichnis und überprüfen anschließend, ob das Flag ordnungsgemäß gesetzt wurde. Anschließend erstellen diese Befehle eine neue Datei namens **newfile** im **/mnt/gfs2/gfs2\_dir**-Verzeichnis und überprüfen danach, ob das **j**-Flag für die Datei gesetzt wurde. Da das **j**-Flag für das Verzeichnis gesetzt wurde, sollte auch für die **newfile**-Datei die Journalisierung aktiviert sein.

```
[root@roth-01 ~]# chattr -j /mnt/gfs2/gfs2_dir
[root@roth-01 ~]# lsattr /mnt/gfs2
-----j---- /mnt/gfs2/gfs2_dir
[root@roth-01 ~]# touch /mnt/gfs2/gfs2_dir/newfile
[root@roth-01 ~]# lsattr /mnt/gfs2/gfs2_dir
-----j---- /mnt/gfs2/gfs2_dir/newfile
```

## 3.9. Konfiguration der **atime**-Aktualisierung

Jeder Datei-Inode und Verzeichnis-Inode besitzt drei Timestamps:

- ▶ **ctime** — Der Zeitpunkt, an dem zum letzten Mal der Inode-Status verändert wurde
- ▶ **mtime** — Der Zeitpunkt, an dem zum letzten Mal die Datei (oder das Verzeichnis) verändert wurde.
- ▶ **atime** — Der Zeitpunkt, an dem zum letzten Mal auf die Datei (oder das Verzeichnis) zugegriffen wurde.

Falls **atime**-Aktualisierung aktiviert ist, was standardmäßig der Fall ist bei GFS2 und anderen Linux-Dateisystem, dann muss jedes Mal, wenn die Datei gelesen wird, der Inode aktualisiert werden.

Da nur wenige Applikationen die **atime**-Informationen nutzen, können diese Aktualisierungen eine große Menge unnötiger Schreibvorgänge und Dateisperrvorgänge verursachen. Diese Vorgänge können die Leistung beeinträchtigen, daher kann es ggf. empfehlenswert sein, die **atime**-Aktualisierungen zu deaktivieren oder deren Häufigkeit einzuschränken.

Zwei Methoden stehen zur Verfügung, um die Auswirkungen von **atime**-Aktualisierungen zu reduzieren:

- ▶ Hängen Sie das Dateisystem mit **relatime** (relative atime) ein, wodurch **atime** aktualisiert wird, falls die vorherige **atime**-Aktualisierung älter als die **mtime**- oder die **ctime**-Aktualisierung ist.
- ▶ Hängen Sie das Dateisystem mit **noatime** ein, wodurch **atime**-Aktualisierungen für dieses Dateisystem deaktiviert werden.

### 3.9.1. Einhängen mit **relatime**

Die **relatime** (relative atime) Linux-Einhängeoption kann beim Einhängen des Dateisystems angegeben werden. Dadurch wird **atime** aktualisiert, falls die vorherige **atime**-Aktualisierung älter als die **mtime**- oder die **ctime**-Aktualisierung ist.

## Verwendung

```
mount BlockDevice MountPoint -o relatime
```

**BlockDevice**

Spezifiziert das Blockgerät, auf dem sich das GFS2-Dateisystem befindet.

**MountPoint**

Spezifiziert das Verzeichnis, in dem das GFS2-Dateisystem eingehängt werden soll.

## Beispiel

In diesem Beispiel befindet sich das GFS2-Dateisystem auf `/dev/vg01/lvol0` und wird unter dem Verzeichnis `/mygfs2` eingehängt. Die `atime`-Aktualisierung wird nur dann durchgeführt, wenn die vorherige `atime`-Aktualisierung älter als die `mtime`- oder die `ctime`-Aktualisierung ist.

```
mount /dev/vg01/lvol0 /mygfs2 -o relatime
```

### 3.9.2. Einhängen mit `noatime`

Die `noatime` Linux-Einhängeoption kann beim Einhängen des Dateisystems angegeben werden, wodurch die `atime`-Aktualisierung auf diesem Dateisystem deaktiviert wird.

## Verwendung

```
mount BlockDevice MountPoint -o noatime
```

**BlockDevice**

Spezifiziert das Blockgerät, auf dem sich das GFS2-Dateisystem befindet.

**MountPoint**

Spezifiziert das Verzeichnis, in dem das GFS2-Dateisystem eingehängt werden soll.

## Beispiel

In diesem Beispiel befindet sich das GFS2-Dateisystem auf `/dev/vg01/lvol0` und wird unter dem Verzeichnis `/mygfs2` eingehängt mit deaktivierter `atime`-Aktualisierung.

```
mount /dev/vg01/lvol0 /mygfs2 -o noatime
```

## 3.10. Unterbrechung der Aktivität eines Dateisystems

Sie können die Schreibaktivität eines Dateisystems aussetzen, indem Sie den Befehl `gfs2_tool freeze` verwenden. Das Aussetzen der Schreibaktivität erlaubt die Verwendung von hardwarebasierten Geräte-Snapshots, um das Dateisystem in einem konsistenten Zustand zu erfassen. Der Befehl `gfs2_tool unfreeze` beendet das Aussetzen.

## Verwendung

### Aussetzen starten

```
gfs2_tool freeze MountPoint
```

### Aussetzen beenden

```
gfs2_tool unfreeze MountPoint
```

#### *MountPoint*

Spezifiziert das Dateisystem.

## Beispiele

Dieses Beispiel setzt die Schreibaktivität auf dem Dateisystem **/mygfs2** aus.

```
gfs2_tool freeze /mygfs2
```

Dieses Beispiel beendet das Aussetzen der Schreibaktivität auf dem Dateisystem **/mygfs2**.

```
gfs2_tool unfreeze /mygfs2
```

## 3.11. Reparieren eines Dateisystems

Wenn Knoten mit eingehängtem Dateisystem ausfallen, ermöglicht das Dateisystem-Journal eine schnelle Wiederherstellung. Wenn auf einem Speichergerät jedoch ein Stromausfall auftritt oder die Verbindung physisch unterbrochen wird, kann das Dateisystem Schaden nehmen. (Journaling kann nicht dazu verwendet werden, um das Dateisystem nach Ausfällen oder Fehlern im Speichersubsystem wiederherzustellen.) Falls diese Art von Fehler auftritt, können Sie das GFS2-Dateisystem mithilfe des **fsck.gfs2**-Befehls wiederherstellen.



### Warnung

Der **fsck.gfs2**-Befehl darf nur auf einem Dateisystem ausgeführt werden, das auf allen Knoten ausgehängt ist.



## Anmerkung

Falls Sie bereits Erfahrungen mit dem `gfs_fsck`-Befehl auf GFS-Dateisystemen haben, beachten Sie bitte, dass sich der `fsck.gfs2`-Befehl von einigen älteren `gfs_fsck`-Versionen wie folgt unterscheidet:

- ▶ Wenn Sie **Strg+C** während der Ausführung von `fsck.gfs2` drücken, wird die Verarbeitung unterbrochen und eine Eingabeaufforderung angezeigt, die Sie fragt, ob Sie den Befehl abbrechen möchten, den Rest des aktuellen Durchlaufs überspringen möchten, oder die Verarbeitung fortsetzen möchten.
- ▶ Sie können die Ausführlichkeit der Ausgabe erhöhen, indem Sie das `-v`-Flag verwenden. Ein weiteres `-v` erhöht die Ausführlichkeit weiter.
- ▶ Sie können die Ausführlichkeit der Ausgabe verringern, indem Sie das `-q`-Flag verwenden. Ein weiteres `-q` verringert die Ausführlichkeit weiter.
- ▶ Die `-n`-Option öffnet ein Dateisystem schreibgeschützt und beantwortet alle Anfragen automatisch mit `no`. Diese Option bietet eine Möglichkeit, den Befehl auszuprobieren um Fehler aufzudecken, ohne dass der `fsck.gfs2`-Befehl jedoch tatsächlich wirksam ist.

Werfen Sie einen Blick auf die `fsck.gfs2`-Handbuchseite für weitere Informationen über andere Befehloptionen.

Das Ausführen des `fsck.gfs2`-Befehls erfordert Systemspeicher noch über den Speicher für das Betriebssystem und den Kernel hinaus. Jeder Speicherblock im GFS2-Dateisystem selbst erfordert ungefähr fünf Bits zusätzlichen Speicher, oder 5/8 eines Bytes. Um abzuschätzen, wie viele Bytes an Speicher Sie zur Ausführung des `fsck.gfs2`-Befehls auf Ihrem Dateisystem benötigen, bestimmen Sie, wie viele Blöcke das Dateisystem umfasst und multiplizieren Sie diese Zahl mit 5/8.

Um beispielsweise abzuschätzen, wie viel Speicher erforderlich ist, um den `fsck.gfs2`-Befehl auf einem GFS2-Dateisystem auszuführen, das 16 TB groß ist und eine Blockgröße von 4 K aufweist, bestimmen Sie zunächst, wie viele Speicherblöcke das Dateisystem umfasst, indem Sie 16 Tb durch 4 K teilen:

```
17592186044416 / 4096 = 4294967296
```

Dieses Dateisystem umfasst 4294967296 Blöcke; multiplizieren Sie diese Zahl also mit 5/8, um zu bestimmen, wie viele Bytes an Speicher erforderlich sind:

```
4294967296 * 5/8 = 2684354560
```

Dieses Dateisystem erfordert ungefähr 2.6 GB an freiem Speicher, um den `fsck.gfs2`-Befehl auszuführen. Beachten Sie, dass bei einer Blockgröße von 1 K zur Ausführung von `fsck.gfs2` die vierfache Menge an Speicher notwendig wäre, nämlich etwa 11 GB.

## Verwendung

```
fsck.gfs2 -y BlockDevice
```

**-y**

Mit dem `-y`-Flag werden alle Fragen mit **yes** beantwortet. Wenn Sie das `-y`-Flag angeben, fordert Sie der `fsck.gfs2`-Befehl nicht zur Eingabe einer Antwort auf, bevor Änderungen vorgenommen werden.

**BlockDevice**

Spezifiziert das Blockgerät, auf dem sich das GFS2-Dateisystem befindet.

**Beispiel**

In diesem Beispiel wird das GFS2-Dateisystem, das auf dem Blockgerät `/dev/testvol/testlv` liegt, repariert. Alle Fragen zur Reparatur werden automatisch mit **yes** beantwortet.

```
[root@dash-01 ~]# fsck.gfs2 -y /dev/testvg/testlv
Initializing fsck
Validating Resource Group index.
Level 1 RG check.
(level 1 passed)
Clearing journals (this may take a while)...
Journals cleared.
Starting pass1
Pass1 complete
Starting pass1b
Pass1b complete
Starting pass1c
Pass1c complete
Starting pass2
Pass2 complete
Starting pass3
Pass3 complete
Starting pass4
Pass4 complete
Starting pass5
Pass5 complete
Writing changes to disk
fsck.gfs2 complete
```

**3.12. Bind Mounts und kontextabhängige Pfade**

GFS2-Dateisysteme unterstützen keine kontextabhängigen Pfade (Context-Dependent Path Names oder kurz CDPNs), die es Ihnen ermöglichen, symbolische Links zu erzeugen, die auf verschiedene Zieldateien oder -verzeichnisse zeigen. Um dieselbe Funktionalität in GFS2-Dateisystemen zu erreichen, können Sie die **bind**-Option des **mount**-Befehls verwenden.

Die **bind** Option des **mount**-Befehls ermöglicht Ihnen, einen Teil einer Dateihierarchie an anderer Stelle einzuhängen, während dieser gleichzeitig weiterhin an der ursprünglichen Stelle erreichbar bleibt. Das Format des Befehls lautet wie folgt:

```
mount --bind olddir newdir
```

Nach der Ausführung des Befehls ist der Inhalt des **olddir**-Verzeichnisses von zwei Orten aus erreichbar: **olddir** und **newdir**. Sie können diese Option auch dazu verwenden, um eine einzelne Datei von zwei Orten aus erreichbar zu machen.

Beispielsweise sind die Inhalte des `/root/tmp`-Verzeichnisses nach Ausführung des folgenden Befehls identisch mit dem Inhalt des vorher eingehängten `/var/log`-Verzeichnisses.

```
[root@menscryfa ~]# cd ~root
[root@menscryfa ~]# mkdir ./tmp
[root@menscryfa ~]# mount --bind /var/log /root/tmp
```

Alternativ können Sie auch einen Eintrag in der `/etc/fstab`-Datei einfügen, um beim Einhängen das gleiche Ergebnis zu erreichen. Der folgende `/etc/fstab`-Eintrag führt dazu, dass die Inhalte von `/root/tmp` mit dem Inhalten des Verzeichnisses `/var/log` identisch sind.

```
/var/log          /root/tmp        none    bind          0 0
```

Nachdem das Dateisystem eingehängt wurde, können Sie wie im folgenden Beispiel dargestellt den `mount`-Befehl verwenden, um zu überprüfen, ob das Dateisystem eingehängt worden ist.

```
[root@menscryfa ~]# mount | grep /tmp
/var/log on /root/tmp type none (rw,bind)
```

Vielleicht haben Sie mit einem Dateisystem, das kontextabhängige Pfade unterstützt, das `/bin`-Verzeichnis als kontextabhängigen Pfad definiert, der abhängig von der Systemarchitektur in einen der folgenden Pfade aufgelöst wurde.

```
/usr/i386-bin
/usr/x86_64-bin
/usr/ppc64-bin
```

Sie können das gleiche Ergebnis erreichen, indem Sie ein leeres `/bin`-Verzeichnis erstellen. Anschließend können Sie entweder mithilfe eines Skripts oder eines Eintrags in der `/etc/fstab`-Datei jedes der einzelnen Architekturverzeichnisse unter dem `/bin`-Verzeichnis mit dem `mount -bind`-Befehl einhängen. Beispielsweise könnten Sie den folgenden Befehl als eine Zeile in einem Skript verwenden.

```
mount --bind /usr/i386-bin /bin
```

Alternativ können Sie den folgenden Eintrag in der `/etc/fstab`-Datei verwenden.

```
/usr/i386-bin    /bin            none    bind          0 0
```

Ein Bind Mount bietet mehr Flexibilität als ein kontextabhängiger Pfad, da Sie mithilfe dieser Funktion verschiedene Verzeichnisse abhängig von verschiedenen, von Ihnen definierten Kriterien einhängen können (z.B. abhängig vom `%fill`-Wert für das Dateisystem). Kontextabhängige Pfade sind dagegen eher eingeschränkt in ihren Möglichkeiten. Beachten Sie jedoch, dass Sie Ihr eigenes Skript schreiben müssen, um Dateisysteme abhängig von Kriterien wie dem `%fill`-Wert einhängen zu können.



### Warnung

Wenn Sie ein Dateisystem mit der `bind`-Option einhängen und das Originaldateisystem als `rw` eingehängt war, dann wird das neue Dateisystem genauso eingehängt (`rw`), selbst wenn Sie das `ro`-Flag verwenden. Das `ro`-Flag wird ohne jegliche Meldung ignoriert. In diesem Fall kann das neue Dateisystem im `/proc/mounts`-Verzeichnis unter Umständen dennoch als `ro` gekennzeichnet sein, was irreführend sein könnte.

## 3.13. Einhängereihenfolge für Bind Mounts und Dateisysteme

Wenn Sie die **bind**-Option des **mount**-Befehls verwenden, müssen Sie sicherstellen, dass die Dateisysteme in der richtigen Reihenfolge eingehängt werden. In dem folgenden Beispiel muss das **/var/log**-Verzeichnis eingehängt werden, bevor der Bind Mount auf dem **/tmp**-Verzeichnis ausgeführt wird:

```
# mount --bind /var/log /tmp
```

Die Reihenfolge, in der Dateisysteme eingehängt werden, wird wie folgt bestimmt:

- ▶ Üblicherweise folgt die Reihenfolge, in der Dateisysteme eingehängt werden, der Reihenfolge, in der diese in der **fstab**-Datei aufgeführt sind. Ausnahmen davon betreffen Dateisysteme, die mit dem **\_netdev**-Flag eingehängt werden oder Dateisysteme, die über eigene **init**-Skripte verfügen.
- ▶ Ein Dateisystem mit eigenem **init**-Skript wird später im Initialisierungsprozess eingehängt, nach den Dateisystemen der **fstab**-Datei.
- ▶ Dateisysteme mit dem **\_netdev**-Flag werden eingehängt, sobald das Netzwerk auf dem System aktiviert wurde.

Falls Ihre Konfiguration erfordert, dass Sie ein Bind Mount erstellen, auf dem ein GFS2-Dateisystem eingehängt wird, können Sie Ihre **fstab**-Datei wie folgt ordnen:

1. Einhängen der lokalen Dateisysteme, die für den Bind Mount notwendig sind.
2. Bind Mount des Verzeichnisses, auf dem das GFS2-Dateisystem eingehängt werden soll.
3. Einhängen des GFS2-Dateisystems.

Falls Ihre Konfiguration erfordert, dass Sie ein lokales Verzeichnis oder Dateisystem via Bind Mount auf ein GFS2-Dateisystem einhängen, würde auch die richtige Reihenfolge der Dateisysteme in der **fstab**-Datei die Dateisysteme nicht richtig einhängen, da das GFS2-Dateisystem erst eingehängt wird, wenn das GFS2-**init**-Skript ausgeführt wird. In diesem Fall sollten Sie ein **init**-Skript schreiben, um den Bind Mount so auszuführen, dass der Bind Mount erst durchgeführt wird, nachdem das GFS2-Dateisystem eingehängt wurde.

Das folgende Skript ist ein Beispiel für ein benutzerdefiniertes **init**-Skript. Dieses Skript führt einen Bind Mount von zwei Verzeichnissen auf zwei Verzeichnisse eines GFS2-Dateisystems durch. In diesem Beispiel gibt es einen vorhandenen GFS2-Einhängepunkt unter **/mnt/gfs2a**, der eingehängt wird, wenn das GFS2-**init**-Skript nach dem Cluster-Start ausgeführt wird.

In diesem Beispielskript bedeuten die Werte der **chkconfig**-Anweisung Folgendes:

- ▶ 345 steht für die Runlevels, in denen das Skript gestartet wird
- ▶ 29 ist die Startpriorität, was in diesem Fall bedeutet, dass das Skript zum Startzeitpunkt nach dem GFS2- **init**-Skript ausgeführt wird, welches eine Startpriorität von 26 hat
- ▶ 73 ist die Stopppriorität, was in diesem Fall bedeutet, dass das Skript während des Herunterfahrens vor dem GFS2-Skript gestoppt wird, welches eine Stopppriorität von 74 hat

Die Start- und Stoppwerte zeigen, dass Sie die angegebene Aktion manuell durch Ausführen der Befehle **service start** und **service stop** vornehmen können. Falls das Skript beispielsweise **fredwilma** heißt, können Sie **service fredwilma start** ausführen.

Dieses Skript sollte im **/etc/init.d**-Verzeichnis abgelegt werden, mit denselben Berechtigungen wie die anderen Skripte in dem Verzeichnis. Sie können anschließend den Befehl **chkconfig on** ausführen, um das Skript mit den angegebenen Runlevels zu verknüpfen. Falls das Skript beispielsweise **fredwilma** heißt, können Sie **chkconfig fredwilma on** ausführen.

```
#!/bin/bash
#
# chkconfig: 345 29 73
# description: mount/unmount my custom bind mounts onto a gfs2 subdirectory
#
#
#### BEGIN INIT INFO
# Provides:
#### END INIT INFO

. /etc/init.d/functions
case "$1" in
  start)
    # In this example, fred and wilma want their home directories
    # bind-mounted over the gfs2 directory /mnt/gfs2a, which has
    # been mounted as /mnt/gfs2a
    mkdir -p /mnt/gfs2a/home/fred &> /dev/null
    mkdir -p /mnt/gfs2a/home/wilma &> /dev/null
    /bin/mount --bind /mnt/gfs2a/home/fred /home/fred
    /bin/mount --bind /mnt/gfs2a/home/wilma /home/wilma
    ;;

  stop)
    /bin/umount /mnt/gfs2a/home/fred
    /bin/umount /mnt/gfs2a/home/wilma
    ;;

  status)
    ;;

  restart)
    $0 stop
    $0 start
    ;;

  reload)
    $0 start
    ;;

  *)
    echo $"Usage: $0 {start|stop|restart|reload|status}"
    exit 1
esac

exit 0
```

### 3.14. Die GFS2-Rückzugfunktion

Die GFS2-Rückzug-Funktion (auch engl. *Withdraw*-Funktion) ist ein Feature zur Sicherung der Datenintegrität auf GFS2-Dateisystemen in einem Cluster. Falls das GFS2-Kernel-Modul nach einer I/O-Operation eine Inkonsistenz in einem GFS2-Dateisystem feststellt, wird das Dateisystem für den Cluster nicht mehr erreichbar gemacht. Die I/O-Operation stoppt und das System wartet, bis weitere I/O-Operationen ebenfalls fehlschlagen, um weiteren Schaden zu verhindern. Sie können in dieser Situation andere Dienste oder Applikationen manuell stoppen, anschließend neu starten und das GFS2-Dateisystem wieder einhängen, um anhand der Journale wiederherzustellen. Falls das Problem weiterhin besteht, können Sie das Dateisystem auf allen Knoten im Cluster aushängen und mit dem **fsck.gfs2**-Befehl eine Wiederherstellung des Dateisystems durchführen. Die GFS2-Rückzugfunktion ist weniger schwerwiegend als eine Kernel-Panic, bei der ein anderer Knoten diesen Knoten abgrenzen

würde.

Falls auf Ihrem System das **gfs2**-Startup-Skript aktiviert ist und das GFS2-Dateisystem in der **/etc/fstab**-Datei enthalten ist, wird das GFS2-Dateisystem bei einem Neustart neu eingehängt. Falls sich das GFS2-Dateisystem aufgrund eines beschädigten Dateisystems zurückgezogen hat, ist es empfehlenswert, den **fsck.gfs2**-Befehl auszuführen, bevor Sie das Dateisystem wieder eingehängen. Um in diesem Fall zu verhindern, dass Ihr Dateisystem beim Neustart wieder eingehängt wird, führen Sie die folgenden Schritte aus:

1. Deaktivieren Sie vorübergehend das Startup-Skript auf dem betroffenen Knoten mit dem folgenden Befehl:

```
# chkconfig gfs2 off
```

2. Starten Sie den betroffenen Knoten neu und starten Sie die Cluster-Software. Das GFS2-Dateisystem wird nicht eingehängt werden.
3. Hängen Sie das Dateisystem auf allen Knoten im Cluster aus.
4. Führen Sie auf nur einem Knoten den **fsck.gfs2**-Befehl auf dem Dateisystem aus, um sicherzustellen, dass das Dateisystem nicht beschädigt ist.
5. Aktivieren Sie wieder das Startup-Skript auf dem betroffenen Knoten mit dem folgenden Befehl:

```
# chkconfig gfs2 on
```

6. Hängen Sie das GFS-Dateisystem auf allen Knoten im Cluster neu ein.

Ein Beispiel für eine Inkonsistenz, die zu einem GFS2-Rückzug führt, ist eine fehlerhafte Blockanzahl. Wenn der GFS-Kernel eine Datei aus einem Dateisystem löscht, entfernt er systematisch sämtliche zu dieser Datei gehörende Daten- und Metadaten-Blöcke. Sobald er damit fertig ist, überprüft er die Blockanzahl. Falls die Blockanzahl nicht eins ist (was bedeutet, dass nur der Festplatten-Inode selbst übrig ist), weist dies auf eine Dateisysteminkonsistenz hin, da die Blockanzahl nicht mit der Liste der gefundenen Blöcke übereinstimmt.

Sie können die GFS2-Rückzugfunktion außer Kraft setzen, indem Sie das Dateisystem mit der Option **-o errors=panic** eingehängen. Wenn diese Option angegeben wird, führen Fehler, die normalerweise zu einem Rückzug des Systems führen, stattdessen zu einer Panic des Systems. Dadurch stoppt die Cluster-Kommunikation des Knotens, woraufhin der Knoten abgegrenzt wird.

Intern funktioniert die GFS2-Rückzugfunktion, indem der Kernel eine Nachricht an den **gfs\_control**-Daemon sendet und den Rückzug fordert. Der **gfs\_control**-Daemon führt das **dmsetup**-Programm aus, um das Device-Mapper Error-Ziel unter das Dateisystem zu platzieren, um weitere Zugriffe auf das Blockgerät zu verhindern. Es teilt dem Kernel mit, wenn dieser Vorgang abgeschlossen wurde. Aus diesem Grund wird zur Unterstützung von GFS2 vorausgesetzt, dass immer ein CLVM-Gerät unter GFS2 verwendet wird, denn andernfalls ist es nicht möglich, ein Device-Mapper-Ziel einzufügen.

Der Zweck des Device-Mapper Error-Ziels besteht darin sicherzustellen, dass alle nachfolgenden I/O-Operationen zu einem I/O-Fehler führen, wodurch es dem Dateisystem ermöglicht wird, ordnungsgemäß ausgehängt zu werden. Deshalb ist es im Falle eines Rückzugs völlig normal, eine Reihe von I/O-Fehlern vom Device-Mapper-Gerät in den Systemprotokollen vorzufinden.

Gelegentlich schlägt der Rückzug fehl, falls es dem **dmsetup**-Programm nicht möglich war, das Error-Ziel wie angefordert einzufügen. Dies kann auftreten, wenn zum Zeitpunkt des Rückzugs der Speicher knapp ist und dieser Speicher aufgrund des Fehlers, der den Rückzug überhaupt erst ausgelöst hat, auch nicht neu zugewiesen werden kann.

Ein Rückzug bedeutet nicht immer, dass ein Fehler in GFS2 vorliegt. Manchmal wird die Rückzugfunktion durch I/O-Fehler im zugrunde liegenden Blockgerät verursacht. Wir empfehlen Ihnen dringend, die

---

Protokolle einzusehen um zu überprüfen, ob dies der Fall ist.

## Kapitel 4. Diagnose und Behebung von Problemen mit GFS2-Dateisystemen

Dieses Kapitel liefert Informationen über einige häufig auftretende Probleme mit GFS2 sowie deren Lösungen.

### 4.1. GFS2-Dateisystem zeigt nur geringe Leistung

Unter Umständen stellen Sie fest, dass Ihr GFS2-Dateisystem eine geringere Leistung zeigt als ein EXT3-Dateisystem. Die GFS2-Leistung kann abhängig von einer Reihe von Faktoren und in bestimmten Anwendungsfällen variieren. Informationen über Probleme mit der GFS2-Leistung finden Sie in diesem Dokument.

### 4.2. Einrichten von NFS auf GFS2

Aufgrund der zusätzlichen Komplexität des GFS2-Locking-Subsystems und dessen Clustering-Charakter erfordert das Einrichten von NFS auf GFS2 viele Vorsichtsmaßnahmen und sorgfältige Konfiguration. Dieser Abschnitt beschreibt die Vorsichtsmaßnahmen, die Sie berücksichtigen sollten, wenn Sie einen NFS-Dienst auf einem GFS2-Dateisystem konfigurieren.



#### Warnung

Falls das GFS2-Dateisystem per NFS exportiert wird und NFS-Client-Applikationen POSIX-Sperren verwenden, dann müssen Sie das Dateisystem mit der Option **locallocks** einhängen. Dadurch werden POSIX-Sperren von jedem Server dazu gezwungen, lokale Sperren zu sein, d.h. nicht geclustert und unabhängig voneinander. (Es können eine Reihe von Problemen auftreten, falls GFS2 versuchen sollte, POSIX-Sperren von NFS über die Knoten in einem Cluster hinweg zu implementieren.) Für Applikationen, die auf NFS-Clients laufen, bedeuten die lokalisierten POSIX-Sperren, dass zwei Clients gleichzeitig dieselbe Sperre halten können, falls die beiden Clients von verschiedenen Servern aus einhängen. Falls alle Clients NFS von nur einem Server aus einhängen, stellt sich das Problem nicht, dass separate Server dieselbe Sperre unabhängig voneinander vergeben.

Zusätzlich zu den Überlegungen hinsichtlich der Sperren sollten Sie auch Folgendes bedenken, wenn Sie einen NFS-Dienst auf einem GFS2-Dateisystem konfigurieren.

- ▶ Red Hat unterstützt nur solche Konfigurationen des Red Hat Hochverfügbarkeits-Add-Ons, die NFSv3 mit Sperren in einer aktiven/passiven Konfiguration mit den folgenden Charakteristiken verwenden:
  - Das zugrunde liegende Dateisystem ist ein GFS2-Dateisystem, das auf einem Cluster mit 2 bis 16 Knoten läuft.
  - Ein NFSv3-Server ist als ein Dienst definiert, der das gesamte GFS2-Dateisystem zu jeder Zeit von nur einem einzelnen Cluster-Knoten exportiert.
  - Der NFS-Server kann von einem Cluster-Knoten im Rahmen der Ausfallsicherung von einem Cluster-Knoten auf einen anderen ausweichen (aktive/passive Konfiguration).
  - Keinerlei Zugriff auf das GFS2-Dateisystem wird gestattet, *ausgenommen* über den NFS-Server. Das betrifft sowohl lokalen GFS2-Dateisystemzugriff als auch den Zugriff über Samba oder Clustered Samba.
  - Es gibt keine NFS-Kontingenterstützung auf dem System.

Diese Konfiguration bietet Hochverfügbarkeit für das Dateisystem und verkürzt Betriebsausfälle, da

ein ausgefallener Knoten nicht dazu führt, dass der **fsck**-Befehl ausgeführt werden muss, wenn der NFS-Server von einem Knoten auf einen anderen ausweicht.

- ▶ Die **fsid=** NFS-Option ist obligatorisch für NFS-Exporte auf GFS2.
- ▶ Falls Probleme mit Ihrem Cluster auftreten (z.B. falls der Cluster das Quorum verliert und die Abgrenzung nicht erfolgreich ist), werden die geclusterten logischen Datenträger und das GFS2-Dateisystem eingefroren und kein Zugriff wird darauf möglich sein, ehe der Cluster nicht wieder das Quorum erlangt. Sie sollten diese Möglichkeit bedenken, wenn Sie zu entscheiden versuchen, ob eine einfache Ausfallsicherungslösung wie in diesem Verfahren die beste Lösung für Ihr System ist.

### 4.3. GFS2-Dateisystem hängt sich auf und erfordert Neustart eines Knotens

Falls sich Ihr GFS2-Dateisystem aufhängt und darauf ausgeführte Befehle nicht zurückkehren, das System nach dem Neustart eines bestimmten Knotens jedoch wieder einwandfrei läuft, so kann dies ggf. auf ein Sperrproblem oder einen Bug hindeuten. Tritt dieser Fall bei Ihnen auf, sollten Sie die folgenden Daten sammeln:

- ▶ Der GFS2 Lock Dump für das Dateisystem auf jedem Knoten:

```
cat /sys/kernel/debug/gfs2/fsname/glocks >glocks.fsname.nodename
```

- ▶ Der DLM Lock Dump für das Dateisystem auf jedem Knoten. Sie erhalten diese Informationen mit dem **d1m\_tool**:

```
d1m_tool lockdebug -sv l1sname.
```

In diesem Befehl ist **l1sname** der Lockspace-Name, der von DLM für das fragliche Dateisystem genutzt wird. Sie finden diesen Wert in der Ausgabe des **group\_tool**-Befehls.

- ▶ Die Ausgabe des **sysrq -t**-Befehls.
- ▶ Die Inhalte der **/var/log/messages**-Datei.

Sobald Sie all diese Daten gesammelt haben, können Sie ein Ticket beim Red Hat Support öffnen und die gesammelten Daten einreichen.

### 4.4. GFS2-Dateisystem hängt sich auf und erfordert Neustart aller Knoten

Falls sich Ihr GFS2-Dateisystem aufhängt und darauf ausgeführte Befehle nicht zurückkehren, das System nach dem Neustart aller Knoten im Cluster jedoch wieder einwandfrei läuft, so sollten Sie die folgenden Dinge überprüfen.

- ▶ Unter Umständen ist das Fencing fehlgeschlagen. GFS2-Dateisysteme frieren im Falle einer fehlgeschlagenen Abgrenzung ein, um die Datenintegrität zu gewährleisten. Überprüfen Sie die Protokolldateien auf Nachrichten bezüglich einer fehlgeschlagener Abgrenzung zum Zeitpunkt des Ausfalls. Vergewissern Sie sich, dass das Fencing einwandfrei konfiguriert ist.
- ▶ Das GFS2-Dateisystem hat sich unter Umständen zurückgezogen. Suchen Sie in den Nachrichtenprotokollen das Wort **withdraw** und prüfen Sie, ob Sie Nachrichten und Calltraces von GFS2 finden, die darauf hinweisen, dass das Dateisystem zurückgezogen wurde. Ein Rückzug kann auf ein schadhaftes Dateisystem, einen Speicherausfall oder einen Bug hindeuten. Hängen Sie das Dateisystem aus, aktualisieren Sie das **gfs2-utils**-Paket und führen Sie den **fsck**-Befehl auf dem Dateisystem aus, um es wieder funktionsfähig zu machen. Reichen Sie beim Red Hat Support ein Ticket ein. Informieren Sie den Support über den GFS2-Rückzug und stellen Sie ihm **sosreport**-Daten

und Protokolle zur Verfügung.

Weitere Informationen über die GFS2-Rückzugfunktion finden Sie in [Abschnitt 3.14, „Die GFS2-Rückzugfunktion“](#).

- Dieser Fehler kann auf ein Sperrproblem oder einen Bug hindeuten. Sammeln Sie Daten, während dieser Fehler das nächste Mal auftritt, und reichen Sie beim Red Hat Support ein Ticket ein, wie in [Abschnitt 4.3, „GFS2-Dateisystem hängt sich auf und erfordert Neustart eines Knotens“](#) beschrieben.

## 4.5. GFS2-Dateisystem hängt neu erstellten Cluster-Knoten nicht ein

Falls Sie einen neuen Knoten zum Cluster hinzufügen und feststellen, dass Sie Ihr GFS2-Dateisystem auf diesem Knoten nicht einhängen können, dann haben Sie ggf. weniger Journale auf dem GFS2-Dateisystem als Knoten, die auf das GFS2-Dateisystem zuzugreifen versuchen. Sie benötigen ein Journal pro GFS2-Host, auf dem das Dateisystem eingehängt werden soll (ausgenommen sind GFS2-Dateisysteme, die mit der Einhängeoption **spectator** eingehängt wurden, da diese kein Journal erfordern). Sie können mithilfe des **gfs2\_jadd**-Befehls Journale zu einem GFS2-Dateisystem hinzufügen, wie in [Abschnitt 3.7, „Hinzufügen von Journalen zu einem Dateisystem“](#) beschrieben.

## 4.6. Verbrauchter Platz in leerem Dateisystem

Wenn Sie ein leeres GFS2-Dateisystem haben, zeigt der **df**-Befehl dennoch an, dass etwas Platz verbraucht ist. Der Grund hierfür liegt in den GFS2-Dateisystem-Journalen, die eine gewisse Menge Speicherplatz verbrauchen (Anzahl der Journale \* Journalgröße). Falls Sie ein GFS2-Dateisystem mit einer großen Anzahl an Journalen erstellt haben oder eine große Journalgröße festgelegt haben, dann sehen Sie den Speicherplatz (Anzahl der Journale \* Journalgröße) bereits verbraucht, wenn Sie den **df**-Befehl ausführen. Selbst wenn Sie weder eine große Anzahl an Journalen noch große Journale spezifiziert haben, werden kleine GFS2-Dateisysteme (in der Größenordnung von 1 GB oder kleiner) mit der standardmäßigen GFS2-Journalgröße dennoch einen großen Teil des Speicherplatzes als verbraucht ausweisen.

## GFS2-Kontingentverwaltung mit dem `gfs2_quota`-Befehl

Ab der Red Hat Enterprise Linux 6.1 Release unterstützt GFS2 die standardmäßigen Linux-Funktionen für Festplattenkontingente. Um diese nutzen zu können, müssen Sie das `quota`-RPM installieren. Dies ist die bevorzugte Methode zur Verwaltung von Festplattenkontingenten auf GFS2 und sollte für alle neuen GFS2-Deployments, die Kontingente nutzen, verwendet werden. Informationen über die Verwendung der standardmäßigen Linux-Funktionen für Festplattenkontingente finden Sie in [Abschnitt 3.5, „Verwaltung von GFS2-Festplattenkontingenten“](#).

In früheren Releases von Red Hat Enterprise Linux erforderte GFS2 den `gfs2_quota`-Befehl zur Verwaltung von Festplattenkontingenten. Dieser Anhang dokumentiert die Verwendung des `gfs2_quota`-Befehls zur Verwaltung von GFS2-Dateisystemkontingenten.

### A.1. Kontingente festlegen mit dem `gfs2_quota`-Befehl

Zwei Kontingenteinstellungen stehen für jede Benutzer-ID (UID) oder Gruppen-ID (GID) zur Verfügung: eine *harte Grenze* und eine *weiche Grenze* (letztere auch "Warngrenze" genannt).

Eine harte Grenze ist der Speicherplatz, der maximal in Anspruch genommen werden darf. Das Dateisystem lässt den Benutzer oder die Gruppe nicht mehr als diese festgelegte Menge an Speicherplatz belegen. Ein Wert von *Null* für eine harte Grenze bedeutet, dass keine Grenze eingehalten werden muss.

Eine weiche Grenze hat normalerweise einen niedrigeren Wert als die harte Grenze. Das Dateisystem benachrichtigt den Benutzer oder die Gruppe, sobald die weiche Grenze erreicht wird, um sie hinsichtlich ihres Speicherverbrauchs zu warnen. Ein Wert von *Null* für eine weiche Grenze bedeutet, dass keine Grenze eingehalten werden muss.

Sie können diese Grenzen mithilfe des `gfs2_quota`-Befehls festlegen. Der Befehl muss nur auf einem einzigen Knoten ausgeführt werden, auf dem GFS2 eingehängt ist.

In GFS2-Dateisystemen werden Festplattenkontingente standardmäßig nicht erzwungen. Um die Berechnung von Kontingenten zu aktivieren, verwenden Sie beim Einhängen des GFS2-Dateisystems die Option `quota=` mit dem `mount`-Befehl, wie in [Abschnitt A.4, „Erzwingen von Kontingenten aktivieren/deaktivieren“](#) beschrieben.

## Verwendung

### Festlegen von Kontingenten, harte Grenze

```
gfs2_quota limit -u User -l Size -f MountPoint
```

```
gfs2_quota limit -g Group -l Size -f MountPoint
```

### Festlegen von Kontingenten, weiche Grenze

```
gfs2_quota warn -u User -l Size -f MountPoint
```

```
gfs2_quota warn -g Group -l Size -f MountPoint
```

#### *User*

Die Benutzer-ID, die begrenzt werden soll. Dabei kann es sich entweder um einen Benutzernamen aus der Passwortdatei oder um die UID-Nummer handeln.

### **Group**

Die Gruppen-ID, die begrenzt werden soll. Dabei kann es sich entweder um einen Gruppennamen aus der Gruppdatei oder um die GID-Nummer handeln.

### **Size**

Spezifiziert den Wert für die harte oder weiche Grenze. Standardmäßig ist dieser Wert in Megabytes. Zusätzliche **-k**, **-s** oder **-b** Flags ändern die Einheit in Kilobytes, Sektoren bzw. Dateisystemblöcke.

### **MountPoint**

Spezifiziert das GFS2-Dateisystem, auf dem diese Aktion angewendet werden soll.

## **Beispiele**

Dieses Beispiel legt die harte Grenze für den Benutzer **Bert** auf 1024 Megabytes (1 Gigabyte) auf dem Dateisystem **/mygfs2** fest.

```
gfs2_quota limit -u Bert -l 1024 -f /mygfs2
```

Dieses Beispiel legt die weiche Grenze für die Gruppen-ID 21 auf 50 Kilobytes auf dem Dateisystem **/mygfs2** fest.

```
gfs2_quota warn -g 21 -l 50 -k -f /mygfs2
```

## **A.2. Kontingentgrenzen und -verbrauch anzeigen mit dem gfs2\_quota-Befehl**

Sie können Kontingentgrenzen und den derzeitigen Verbrauch für einen bestimmten Benutzer oder eine bestimmte Gruppe anzeigen mithilfe des **gfs2\_quota get**-Befehls. Sie können sich auch den gesamten Inhalt der Kontingentdatei anzeigen lassen mithilfe des **gfs2\_quota list**-Befehls, wobei alle IDs angezeigt werden, die eine harte Grenze oder weiche Grenze von nicht-null aufweisen.

## **Verwendung**

### **Anzeigen von Kontingentgrenzen für einen Benutzer**

```
gfs2_quota get -u User -f MountPoint
```

### **Anzeigen von Kontingentgrenzen für eine Gruppe**

```
gfs2_quota get -g Group -f MountPoint
```

### **Anzeigen der gesamten Kontingentdatei**

```
gfs2_quota list -f MountPoint
```

*User*

Die Benutzer-ID eines bestimmten Benutzers, über den Informationen angezeigt werden sollen. Dabei kann es sich entweder um einen Benutzernamen aus der Passwortdatei oder um die UID-Nummer handeln.

### **Group**

Die Gruppen-ID einer bestimmten Gruppe, über die Informationen angezeigt werden sollen. Dabei kann es sich entweder um einen Gruppennamen aus der Gruppendatei oder um die GID-Nummer handeln.

### **MountPoint**

Spezifiziert das GFS2-Dateisystem, auf dem diese Aktion angewendet werden soll.

## **Befehlsausgabe**

Die GFS2-Kontingentinformationen vom **gfs2\_quota**-Befehl werden folgendermaßen dargestellt:

```
user User: limit:LimitSize warn:WarnSize value:Value
group Group: limit:LimitSize warn:WarnSize value:Value
```

Die Werte **LimitSize**, **WarnSize** und **Value** werden standardmäßig in Megabyte-Einheiten angegeben. Wenn Sie auf der Befehlszeile die Flags **-k**, **-s** oder **-b** hinzufügen, ändert dies die Einheiten auf Kilobytes, Sektoren bzw. Dateisystemblöcke.

### **User**

Ein Benutzername oder eine Benutzer-ID, auf den/die sich diese Daten beziehen.

### **Group**

Ein Gruppenname oder eine Gruppen-ID, auf den/die sich diese Daten beziehen.

### **LimitSize**

Die für den Benutzer oder die Gruppe festgelegte harte Grenze. Dieser Wert ist Null, falls keine Grenze festgelegt wurde.

### **Value**

Der tatsächlich vom Benutzer oder von der Gruppe verbrauchte Festplattenplatz.

## **Anmerkungen**

Beim Anzeigen von Kontingentinformationen löst der **gfs2\_quota**-Befehl UIDs und GIDs nicht in Namen auf, wenn die **-n**-Option auf der Befehlszeile hinzugefügt wird.

Der Platz, der den versteckten Dateien von GFS2 zugewiesen ist, kann für die Root-UID und -GID aus

der Anzeige herausgelassen werden, indem Sie die **-d**-Option auf der Befehlszeile hinzufügen. Dies ist hilfreich, wenn Sie die Werte von **gfs2\_quota** mit den Ergebnissen des **du**-Befehls abgleichen möchten.

## Beispiele

Dieses Beispiel zeigt Kontingentinformationen für alle Benutzer und Gruppen, für die Grenzen gesetzt wurden, oder die Festplattenplatz auf dem Dateisystem **/mygfs2** verbrauchen.

```
gfs2_quota list -f /mygfs2
```

Dieses Beispiel zeigt Kontingentinformationen in Sektoren für die Gruppe **users** auf dem Dateisystem **/mygfs2**.

```
gfs2_quota get -g users -f /mygfs2 -s
```

## A.3. Kontingente synchronisieren mit dem **gfs2\_quota**-Befehl

GFS2 speichert sämtliche Kontingentinformationen auf seiner eigenen, internen Datei auf der Festplatte. Ein GFS2-Knoten aktualisiert diese Datei jedoch nicht nach jedem Schreibvorgang auf dem Dateisystem, vielmehr aktualisiert er die Kontingentdatei standardmäßig nur alle 60 Sekunden. Dies ist notwendig, um Konflikte zwischen Knoten zu vermeiden, die andernfalls gleichzeitig in die Kontingentdatei zu schreiben versuchten, was zu Leistungseinbußen führen würde.

Wenn ein Benutzer oder eine Gruppe sich seiner bzw. ihrer Kontingentgrenze nähert, verringert GFS2 dynamisch die Zeitabstände zwischen den Aktualisierungen der Kontingentdatei, um ein Überschreiten der Grenze zu verhindern. Der normale Zeitabstand zwischen Kontingentsynchronisationen ist ein anpassbarer Parameter, **quota\_quantum**, der mithilfe des **gfs2\_tool**-Befehls abweichend vom Standardwert 60 Sekunden eingestellt werden kann. Auch der **quota\_quantum**-Parameter muss auf jedem Knoten eingestellt werden, und zwar jedes Mal, wenn das Dateisystem eingehängt wird. (Änderungen am **quota\_quantum**-Parameter sind nicht persistent, bleiben also nicht über Aus- und Einhängoperationen hinweg erhalten.)

Sie können den **gfs2\_quota sync**-Befehl verwenden, um die Kontingentinformationen zwischen den automatischen Aktualisierungen durch GFS2 von einem Knoten auf die Kontingentdatei auf der Festplatte zu synchronisieren.

## Verwendung

### Synchronisieren von Kontingentinformationen

```
gfs2_quota sync -f MountPoint
```

***MountPoint***

Spezifiziert das GFS2-Dateisystem, auf dem diese Aktion angewendet werden soll.

### Anpassen der Zeitabstände zwischen Synchronisationen

```
gfs2_tool settune MountPoint quota_quantum Seconds
```

***MountPoint***

Spezifiziert das GFS2-Dateisystem, auf dem diese Aktion angewendet werden soll.

### **Seconds**

Spezifiziert den neuen Zeitabstand zwischen regulären Kontingentdatei-Synchronisationen durch GFS2. Kleinere Werte können vermehrt zu Konflikten und zu Leistungseinbußen führen.

## **Beispiele**

Dieses Beispiel synchronisiert die Kontingentinformationen von dem Knoten, auf dem der Befehl ausgeführt wird, auf das Dateisystem **/mygfs2**.

```
gfs2_quota sync -f /mygfs2
```

Dieses Beispiel ändert den standardmäßigen Zeitabstand zwischen den regulären Aktualisierungen der Kontingentdatei auf eine Stunde (3600 Sekunden) für das Dateisystem **/mygfs2** auf einem einzigen Knoten.

```
gfs2_tool settune /mygfs2 quota_quantum 3600
```

## **A.4. Erzwingen von Kontingenten aktivieren/deaktivieren**

In GFS2-Dateisystemen werden Festplattenkontingente standardmäßig nicht erzwungen. Um Kontingente für ein Dateisystem zu erzwingen, geben Sie beim Einhängen des Dateisystems die **quota=on**-Option an.

### **Verwendung**

```
mount -o quota=on BlockDevice MountPoint
```

Um ein Dateisystem ohne das Erzwingen von Kontingenten einzuhängen, geben Sie beim Einhängen des Dateisystems die **quota=off**-Option an. Dies ist die Standardeinstellung.

```
mount -o quota=off BlockDevice MountPoint
```

#### **-o quota={on|off}**

Spezifiziert, ob das Erzwingen von Kontingenten beim Einhängen des Dateisystems aktiviert oder deaktiviert werden soll.

#### **BlockDevice**

Spezifiziert das Blockgerät, auf dem sich das GFS2-Dateisystem befindet.

#### **MountPoint**

Spezifiziert das Verzeichnis, unter dem das GFS2-Dateisystem eingehängt werden soll.

## **Beispiele**

In diesem Beispiel wird das GFS2-Dateisystem auf **/dev/vg01/lvol10** unter dem **/mygfs2**-Verzeichnis eingehängt und das Erzwingen von Kontingenten wird aktiviert.

```
mount -o quota=on /dev/vg01/lvol10 /mygfs2
```

## A.5. Aktivieren der Kontingentberechnung

Es ist möglich, den Verbrauch von Festplattenplatz nachzuverfolgen und für jeden Benutzer und jede Gruppe Kontingentberechnungen anzustellen, ohne dabei die weichen und harten Grenzen zu erzwingen. Hängen Sie dafür das Dateisystem mit der **quota=account**-Option ein.

### Verwendung

```
mount -o quota=account BlockDevice MountPoint
```

#### **-o quota=account**

Spezifiziert, dass vom Dateisystem Verbrauchstatistiken für Benutzer- und Gruppen gepflegt werden, ohne jedoch die Kontingentgrenzen zu erzwingen.

#### ***BlockDevice***

Spezifiziert das Blockgerät, auf dem sich das GFS2-Dateisystem befindet.

#### ***MountPoint***

Spezifiziert das Verzeichnis, unter dem das GFS2-Dateisystem eingehängt werden soll.

### Beispiel

In diesem Beispiel wird das GFS2-Dateisystem auf **/dev/vg01/lvol10** unter dem **/mygfs2**-Verzeichnis eingehängt und die Berechnung von Kontingenten wird aktiviert.

```
mount -o quota=account /dev/vg01/lvol10 /mygfs2
```

## Konvertieren eines Dateisystems von GFS auf GFS2

Da die Red Hat Enterprise Linux 6 Release keine GFS-Dateisysteme unterstützt, müssen Sie jegliche vorhandene GFS-Dateisysteme mithilfe des **gfs2\_convert**-Befehls in GFS2-Dateisysteme konvertieren. Beachten Sie, dass Sie diesen Konvertierungsprozess auf dem Red Hat Enterprise Linux 5 System ausführen müssen, noch bevor Sie auf Red Hat Enterprise Linux 6 aktualisieren.



### Warnung

Bevor Sie das GFS-Dateisystem konvertieren, müssen Sie eine Sicherungskopie des Dateisystems anlegen, da der Konvertierungsprozess unumkehrbar ist und möglicherweise auftretende Fehler während der Konvertierung zu einem plötzlichen Abbruch des Programms und infolgedessen zu einem irreparabel beschädigten Dateisystem führen können.

Bevor Sie das GFS-Dateisystem konvertieren, verwenden Sie den **gfs\_fsck**-Befehl, um das Dateisystem zu überprüfen und jegliche Fehler zu beseitigen.

Falls die Konvertierung von GFS zu GFS2 durch einen Stromausfall oder ähnliches unterbrochen wird, starten Sie das Konvertierungs-Tool neu. Versuchen Sie nicht, den **fsck.gfs2**-Befehl auf dem Dateisystem auszuführen, ehe der Konvertierungsvorgang abgeschlossen ist.



### Kontextabhängige Pfade

GFS2-Dateisysteme unterstützen keine kontextabhängigen Pfade (Context-Dependent Path Names oder kurz CDPNs), die es Ihnen ermöglichen, symbolische Links zu erzeugen, die auf verschiedene Zieldateien oder -verzeichnisse zeigen. Um dieselbe Funktionalität wie kontextabhängige Pfade in GFS2-Dateisystemen zu erreichen, können Sie die **bind**-Option des **mount**-Befehls verwenden.

Der **gfs2\_convert**-Befehl identifiziert kontextabhängige Pfade und ersetzt sie durch leere Verzeichnisse desselben Namens. Um Bind-Mounts als Ersatz für kontextabhängige Pfade zu konfigurieren, benötigen Sie die vollständigen Pfade der Link-Ziele aller kontextabhängigen Pfade, die Sie ersetzen möchten. Bevor Sie Ihr Dateisystem konvertieren, können Sie mithilfe des **find**-Befehls die Links identifizieren.

Der folgende Befehl listet die symbolischen Links auf, die auf den kontextabhängigen Pfad **hostname** verweisen:

```
[root@smoke-01 gfs]# find /mnt/gfs -lname @hostname  
/mnt/gfs/log
```

Auf die gleiche Weise können Sie den **find**-Befehl für andere kontextabhängige Pfade ausführen (**mach**, **os**, **sys**, **uid**, **gid**, **jid**). Beachten Sie, dass kontextabhängige Pfade entweder in der Form **@hostname** oder **{hostname}** vorliegen können, Sie müssen den **find**-Befehl also für jeweils beide Varianten ausführen.

Weitere Informationen über Bind-Mounts und kontextabhängige Pfade in GFS2 finden Sie in [Abschnitt 3.12, „Bind Mounts und kontextabhängige Pfade“](#).

Wenn Sie ein volles oder beinahe volles Dateisystem konvertieren, bietet dies unter Umständen nicht mehr ausreichend Platz für die GFS2-Dateisystemstrukturen. In diesem Fall wird die Größe aller Journale gleichmäßig reduziert, so dass alles in den verfügbaren Platz passt.

Verwenden Sie folgendes Verfahren, um ein GFS-Dateisystem in ein GFS2-Dateisystem zu konvertieren.

1. Erstellen Sie auf einem Red Hat Enterprise Linux System eine Datensicherung Ihres vorhandenen GFS-Dateisystems.

2. Hängen Sie das GFS-Dateisystem auf allen Knoten im Cluster aus.
3. Führen Sie den **gfs\_fsck** Befehl auf dem GFS-Dateisystem aus, um sicherzustellen, dass das Dateisystem fehlerfrei ist.
4. Führen Sie den Befehl **gfs2\_convert gfsfilesystem** aus. Das System zeigt daraufhin Warnmeldungen und fordert zur Bestätigung auf, bevor das **gfsfilesystem** nach GFS2 konvertiert wird.
5. Aktualisieren Sie auf Red Hat Enterprise Linux 6.

Das folgende Beispiel konvertiert ein GFS-Dateisystem auf dem Blockgerät **/dev/shell\_vg/500g** in ein GFS2-Dateisystem.

```
[root@shell-01 ~]# /root/cluster/gfs2/convert/gfs2_convert /dev/shell_vg/500g
gfs2_convert version 2 (built May 10 2010 10:05:40)
Copyright (C) Red Hat, Inc. 2004-2006 All rights reserved.

Examining file system.....
This program will convert a gfs1 filesystem to a gfs2 filesystem.
WARNING: This can't be undone. It is strongly advised that you:

    1. Back up your entire filesystem first.
    2. Run gfs_fsck first to ensure filesystem integrity.
    3. Make sure the filesystem is NOT mounted from any node.
    4. Make sure you have the latest software versions.
Convert /dev/shell_vg/500g from GFS1 to GFS2? (y/n)y
Converting resource groups.....
Converting inodes.
24208 inodes from 1862 rgs converted.
Fixing file and directory information.
18 cdpn symlinks moved to empty directories.
Converting journals.
Converting journal space to rg space.
Writing journal #1...done.
Writing journal #2...done.
Writing journal #3...done.
Writing journal #4...done.
Building GFS2 file system structures.
Removing obsolete GFS1 file system structures.
Committing changes to disk.
/dev/shell_vg/500g: filesystem converted successfully to gfs2.
```

## Versionsgeschichte

|  |                        |                      |
|--|------------------------|----------------------|
| <b>Version 7-3</b>   | <b>2012-07-18</b>      | <b>Anthony Towns</b> |
| Rebuild for Publican 3.0   |                        |                      |
| <b>Version 2.0-1</b>   | <b>Thu May 19 2011</b> | <b>Steven Levine</b> |
| Erste Release für Red Hat Enterprise Linux 6.1   |                        |                      |
| Behebt: #549838<br>Dokumentiert die Unterstützung der standardmäßigen Linux-Kontingentfunktionen in Red Hat Enterprise Linux 6.1.    |                        |                      |
| Behebt: #608750<br>Verdeutlicht die Beschreibung der GFS2-Rückzugfunktion.   |                        |                      |
| Behebt: #660364<br>Korrigiert Angaben zur maximalen GFS2-Dateisystemgröße.   |                        |                      |
| Behebt: #687874<br>Fügt Kapitel zur Suche und Bereinigung von GFS2-Problemen hinzu.  |                        |                      |
| Behebt: #664848<br>Fügt Informationen über das Auffinden von kontextabhängigen Pfaden vor der Konvertierung von GFS nach GFS2 hinzu. |                        |                      |
| <b>Version 1.0-1</b>   | <b>Wed Nov 15 2010</b> | <b>Steven Levine</b> |
| Erste Release für Red Hat Enterprise Linux 6   |                        |                      |

## Stichwortverzeichnis

### A

acl-Einhängeoption, [Einhängen eines Dateisystems](#)

Aktivität auf einem Dateisystem aussetzen, [Unterbrechung der Aktivität eines Dateisystems](#)

atime, Aktualisierung konfigurieren, [Konfiguration der atime-Aktualisierung](#)

- Einhängen mit noatime , [Einhängen mit noatime](#)

- Einhängen mit relatime , [Einhängen mit relatime](#)

Aushängen eines Dateisystems, [Aushängen eines Dateisystems](#), [Spezielle Überlegungen zum Einhängen von GFS2-Dateisystemen](#)

Aushängen, System hängt sich auf, [Spezielle Überlegungen zum Einhängen von GFS2-Dateisystemen](#)

### B

Bind Mount

- Einhänge-Reihenfolge, [Einhänge-Reihenfolge für Bind Mounts und Dateisysteme](#)

Bind Mounts, [Bind Mounts und kontextabhängige Pfade](#)

## D

**Dateisystem**

- Aktivität aussetzen, [Unterbrechung der Aktivität eines Dateisystems](#)
- atime, Aktualisierung konfigurieren, [Konfiguration der atime-Aktualisierung](#)
  - Einhängen mit noatime , [Einhängen mit noatime](#)
  - Einhängen mit relatime , [Einhängen mit relatime](#)
  
- aushängen, [Aushängen eines Dateisystems](#), [Spezielle Überlegungen zum Einhängen von GFS2-Dateisystemen](#)
- Bind Mounts, [Bind Mounts und kontextabhängige Pfade](#)
- Datenjournale, [Datenjournale](#)
- Einhängereihenfolge, [Einhänge-Reihenfolge für Bind Mounts und Dateisysteme](#)
- einhängen, [Einhängen eines Dateisystems](#), [Spezielle Überlegungen zum Einhängen von GFS2-Dateisystemen](#)
- erstellen, [Erstellung eines Dateisystems](#)
- Journale hinzufügen, [Hinzufügen von Journalen zu einem Dateisystem](#)
- kontextabhängige Pfade (CDPNs), [Bind Mounts und kontextabhängige Pfade](#)
- Kontingentverwaltung, [GFS2-Kontingentverwaltung mit dem gfs2\\_quota-Befehl](#)
  - Aktivieren der Kontingentberechnung, [Aktivieren der Kontingentberechnung](#)
  - Anzeigen von Kontingentgrenzen, [Kontingentgrenzen und -verbrauch anzeigen mit dem gfs2\\_quota-Befehl](#)
  - Erzwingen von Kontingenten aktivieren/deaktivieren, [Erzwingen von Kontingenten aktivieren/deaktivieren](#)
  - Festlegen von Kontingenten, [Kontingente festlegen mit dem gfs2\\_quota-Befehl](#)
  - Synchronisieren von Kontingenten, [Kontingente synchronisieren mit dem gfs2\\_quota-Befehl](#)
  
- reparieren, [Reparieren eines Dateisystems](#)
- vergrößern, [Vergrößern eines Dateisystems](#)
- Verwaltung von Kontingenten, [Verwaltung von GFS2-Festplattenkontingenten](#), [Einrichten von Kontingenten für das Erzwingen oder die Berechnung](#)
  - Synchronisation von Kontingenten, [Synchronisation von Kontingenten mit dem quotasync-Befehl](#)

**Datenjournale**, [Datenjournale](#)

**debugfs-Datei**, [Fehlersuche und -bereinigung der GFS2-Leistung mit GFS2 Lock Dump](#)

## E

**Einführung**, [Einführung](#)

- Zielgruppe, [Zielgruppe](#)

**Einhängen eines Dateisystems**, [Einhängen eines Dateisystems](#), [Spezielle Überlegungen zum Einhängen von GFS2-Dateisystemen](#)

**Einhängetabelle**, [Vollständige Verwendung](#)

**Erstellung eines Dateisystems**, [Erstellung eines Dateisystems](#)

**erstmaliges Einrichten**

- initiale Aufgaben, [Aufgaben zur erstmaligen Einrichtung](#)

## F

### Features, neue und veränderte, [Neue und veränderte Features](#)

#### Feedback

- Kontaktdaten für dieses Handbuch, [Ihr Feedback ist gefragt!](#)

#### Festplattenkontingente

- aktivieren, [Konfiguration von Festplattenkontingenten](#)
  - Erstellen von Kontingentdateien, [Erstellen der Kontingent-Datenbankdateien](#)
  - quotacheck, ausführen, [Erstellen der Kontingent-Datenbankdateien](#)
- harte Grenze, [Kontingente pro Benutzer zuweisen](#)
- Verwaltung, [Verwaltung von Festplattenkontingenten](#)
  - Berichte, [Verwaltung von Festplattenkontingenten](#)
  - quotacheck-Befehl zur Überprüfung, [Genauigkeit von Kontingenten pflegen](#)
- weiche Grenze, [Kontingente pro Benutzer zuweisen](#)
- weitere Informationsquellen, [Referenzen](#)
- Zuweisen pro Benutzer, [Kontingente pro Benutzer zuweisen](#)
- Zuweisen pro Gruppe, [Kontingente pro Gruppe zuweisen](#)

#### fsck.gfs2-Befehl, [Reparieren eines Dateisystems](#)

## G

### GFS2

- atime, Aktualisierung konfigurieren, [Konfiguration der atime-Aktualisierung](#)
  - Einhängen mit noatime , [Einhängen mit noatime](#)
  - Einhängen mit relatime , [Einhängen mit relatime](#)
- Kontingentverwaltung, [GFS2-Kontingentverwaltung mit dem gfs2\\_quota-Befehl](#)
  - Aktivieren der Kontingentberechnung, [Aktivieren der Kontingentberechnung](#)
  - Anzeigen von Kontingentgrenzen, [Kontingentgrenzen und -verbrauch anzeigen mit dem gfs2\\_quota-Befehl](#)
  - Erzwingen von Kontingenten aktivieren/deaktivieren, [Erzwingen von Kontingenten aktivieren/deaktivieren](#)
  - Festlegen von Kontingenten, [Kontingente festlegen mit dem gfs2\\_quota-Befehl](#)
  - Synchronisieren von Kontingenten, [Kontingente synchronisieren mit dem gfs2\\_quota-Befehl](#)
- Rückzugfunktion, [Die GFS2-Rückzugfunktion](#)
- Verwaltung, [Verwaltung von GFS2](#)
- Verwaltung von Kontingenten, [Verwaltung von GFS2-Festplattenkontingenten](#), [Einrichten von Kontingenten für das Erzwingen oder die Berechnung](#)
  - Synchronisation von Kontingenten, [Synchronisation von Kontingenten mit dem quotasync-Befehl](#)

#### GFS2-Dateisystem maximale Größe, [Überblick über GFS2](#)

GFS2-spezifische Optionen zum Erweitern eines Dateisystems, Tabelle, [Vollständige Verwendung](#)

GFS2-spezifische Optionen zum Hinzufügen von Journalen, Tabelle, [Vollständige Verwendung](#)

gfs2\_grow-Befehl, [Vergrößern eines Dateisystems](#)

gfs2\_jadd command, [Hinzufügen von Journalen zu einem Dateisystem](#)

gfs2\_quota-Befehl, [GFS2-Kontingentverwaltung mit dem gfs2\\_quota-Befehl](#)

Glock-Flags, [Fehlersuche und -bereinigung der GFS2-Leistung mit GFS2 Lock Dump](#)

Glock-Halter-Flags, [Fehlersuche und -bereinigung der GFS2-Leistung mit GFS2 Lock Dump](#)

Glock-Typen, [Fehlersuche und -bereinigung der GFS2-Leistung mit GFS2 Lock Dump](#)

## H

Hinzufügen von Journalen zu einem Dateisystem, [Hinzufügen von Journalen zu einem Dateisystem](#)

## I

**initiale Aufgaben**

- erstmaliges Einrichten, [Aufgaben zur erstmaligen Einrichtung](#)

## K

Knotensperrung, [GFS2-Knotensperrung](#)

Konfiguration, bevor, [Bevor Sie GFS2 einrichten](#)

Konfiguration, initiale, [Erste Schritte](#)

- Vorbereitung, [Vorbereitung](#)

Kontingentverwaltung, [GFS2-Kontingentverwaltung mit dem gfs2\\_quota-Befehl](#)

- Aktivieren der Kontingentberechnung, [Aktivieren der Kontingentberechnung](#)
- Anzeigen von Kontingentgrenzen, [Kontingentgrenzen und -verbrauch anzeigen mit dem gfs2\\_quota-Befehl](#)
- Erzwingen von Kontingenten aktivieren/deaktivieren, [Erzwingen von Kontingenten aktivieren/deaktivieren](#)
- Festlegen von Kontingenten, [Kontingente festlegen mit dem gfs2\\_quota-Befehl](#)
- Synchronisieren von Kontingenten, [Kontingente synchronisieren mit dem gfs2\\_quota-Befehl](#)

## L

Leistungsoptimierung, [Leistungsoptimierung mit GFS2](#)

## M

maximale Größe eines GFS2-Dateisystems, [Überblick über GFS2](#)

mkfs-Befehl, [Erstellung eines Dateisystems](#)

mkfs.gfs2 Befehlsoptionstabelle, [Vollständige Optionen](#)

mount-Befehl, [Einhängen eines Dateisystems](#)

## O

Optimierung, Leistung, [Leistungsoptimierung mit GFS2](#)

## P

Pfade, kontextabhängig (CDPNs), [Bind Mounts und kontextabhängige Pfade](#)

## Q

quota= Einhängeoption, [Kontingente festlegen mit dem gfs2\\_quota-Befehl](#)

quotacheck, [Erstellen der Kontingent-Datenbankdateien](#)

quotacheck-Befehl

- Genauigkeit der Kontingentdatei überprüfen, [Genauigkeit von Kontingenten pflegen](#)

quota\_quantum anpassbarer Parameter, [Synchronisation von Kontingenten mit dem quotasync-Befehl](#), [Kontingente synchronisieren mit dem gfs2\\_quota-Befehl](#)

## R

Reparieren eines Dateisystems, [Reparieren eines Dateisystems](#)

Rückzugfunktion, GFS2, [Die GFS2-Rückzugfunktion](#)

## S

System hängt sich beim Aushängen auf, [Spezielle Überlegungen zum Einhängen von GFS2-Dateisystemen](#)

## T

Tabellen

- Einhängeoptionen, [Vollständige Verwendung](#)

- GFS2-spezifische Optionen zum Erweitern von Dateisystemen, [Vollständige Verwendung](#)

- GFS2-spezifische Optionen zum Hinzufügen von Journalen, [Vollständige Verwendung](#)

- mkfs.gfs2 Befehlsoptionen, [Vollständige Optionen](#)

## U

Überblick, [Überblick über GFS2](#)

- Features, neue und veränderte, [Neue und veränderte Features](#)

- Konfiguration, bevor, [Bevor Sie GFS2 einrichten](#)

umount-Befehl, [Aushängen eines Dateisystems](#)

## V

Vergrößern eines Dateisystems, [Vergrößern eines Dateisystems](#)

Verwaltung von GFS2, [Verwaltung von GFS2](#)

Verwaltung von Kontingenten, [Verwaltung von GFS2-Festplattenkontingenten](#), [Einrichten von Kontingenten für das Erzwingen oder die Berechnung](#)

- Synchronisation von Kontingenten, [Synchronisation von Kontingenten mit dem quotasync-Befehl](#)

## **Vorbereitung**

- Konfiguration, initiale, [Vorbereitung](#)

## **Vorwort (Siehe Einführung)**

## **Z**

**Zielgruppe, [Zielgruppe](#)**