



# Red Hat Enterprise Linux 6

## Global File System 2

Red Hat Global File System 2

Edizione 7

Last Updated: 2017-10-12



# Red Hat Enterprise Linux 6 Global File System 2

---

Red Hat Global File System 2

Edizione 7

## Nota Legale

Copyright © 2014 Red Hat, Inc. and others.

This document is licensed by Red Hat under the [Creative Commons Attribution-ShareAlike 3.0 Unported License](#). If you distribute this document, or a modified version of it, you must provide attribution to Red Hat, Inc. and provide a link to the original. If the document is modified, all Red Hat trademarks must be removed.

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, OpenShift, 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.

Node.js ® is an official trademark of Joyent. Red Hat Software Collections is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack ® Word Mark and OpenStack logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

## Sommario

Questo manuale fornisce le informazioni necessarie per i processi d'installazione, di configurazione e di gestione del Red Hat GFS2 (Red Hat Global File System 2) per Red Hat Enterprise Linux 6.

## Indice

<b>INTRODUZIONE</b> .....	<b>5</b>
1. A CHI È RIVOLTO	5
2. DOCUMENTAZIONE CORRELATA	5
3. ABBIAMO BISOGNO DEI VOSTRI COMMENTI!	6
 <b>CAPITOLO 1. PANORAMICA SUL GFS2</b> .....	 <b>7</b>
1.1. FUNZIONI NUOVE E MODIFICATE	8
1.1.1. Funzioni nuove e modificate di Red Hat Enterprise Linux 6.0	8
1.1.2. Funzioni nuove e modificate di Red Hat Enterprise Linux 6.1	8
1.1.3. Funzioni nuove e modificate di Red Hat Enterprise Linux 6.2	9
1.1.4. Funzioni nuove e modificate di Red Hat Enterprise Linux 6.3	9
1.1.5. Funzioni nuove e modificate di Red Hat Enterprise Linux 6.4	9
1.1.6. Funzionalità nuove e aggiornate per Red Hat Enterprise Linux 6.6	9
1.2. PRIMA D'IMPOSTARE IL GFS2	10
1.3. INSTALLAZIONE DI GFS2	10
1.4. DIFFERENZE TRA GFS E GFS2	11
1.4.1. Nomi dei comandi di GFS2	11
1.4.2. Differenze aggiuntive tra GFS e GFS2	12
Context-Dependent Path Names	12
Modulo gfs2.ko	12
Come abilitare il Quota Enforcement in GFS2	13
Data Journaling	13
Come aggiungere i journal dinamicamente	13
parametro atime_quantum rimosso	13
data= opzione del comando mount	13
Il comando gfs2_tool	13
Il comando gfs2_edit	14
1.4.3. Miglioramenti delle prestazioni di GFS2	14
 <b>CAPITOLO 2. CONSIDERAZIONI OPERATIVE E CONFIGURAZIONE DEL GFS2</b> .....	 <b>16</b>
2.1. CONSIDERAZIONI SULLA FORMATTAZIONE	16
2.1.1. Dimensione del file system: è meglio avere una dimensione più piccola	16
2.1.2. Dimensione blocco: È preferito il default di (4K) blocchi.	16
2.1.3. Numero di journal: Uno per ogni nodo che esegue il montaggio	17
2.1.4. Dimensione journal: Predefinita (128MB), generalmente risulta essere ottimale	17
2.1.5. Dimensione e numero dei gruppi delle risorse	17
2.2. FRAMMENTAZIONE DEL FILE SYSTEM	18
2.3. PROBLEMATICHE RELATIVE ALL'ASSEGNAZIONE DEL BLOCCO	18
2.3.1. Avere spazio disponibile nel file system	18
2.3.2. Se possibile ogni nodo deve assegnare i propri file	18
2.3.3. Se possibile, preassegnate	19
2.4. CONSIDERAZIONI SUL CLUSTER	19
2.5. CONSIDERAZIONI SULL'USO	19
2.5.1. Opzioni di montaggio: noatime e nodiratime	19
2.5.2. Opzioni ottimizzazione DLM: Aumento dimensioni della tabella DLM	20
2.5.3. Opzioni di ottimizzazione VFS: Ricerca e Sperimentazione	20
2.5.4. SELinux: Evitare l'uso di SELinux su GFS2	20
2.5.5. Impostazione di NFS attraverso GFS2	20
2.5.6. Samba (SMB o Windows) File Serving attraverso GFS2	21
2.6. BACKUP DEL FILE SYSTEMA	22
2.7. CONSIDERAZIONI HARDWARE	22
2.8. PROBLEMATICHE DELLE PRESTAZIONI: CONTROLLARE IL PORTALE CLIENTI DI RED HAT	23

2.9. BLOCCO DEI NODI GFS2	23
2.9.1. Problematichette con il Posix Locking	24
2.9.2. Regolazione delle prestazioni con GFS2	24
2.9.3. Risoluzione problematiche relative alle prestazioni di GFS2 con il GFS2 Lock Dump	25
<b>CAPITOLO 3. PER INIZIARE</b> .....	<b>29</b>
3.1. PREREQUISITI	29
3.2. COMPITI INIZIALI PER L'IMPOSTAZIONE	29
<b>CAPITOLO 4. GESTIONE DEL GFS2</b> .....	<b>31</b>
4.1. CREAZIONE DI UN FILE SYSTEM	31
Utilizzo	31
Esempi	33
Opzioni complete	33
4.2. MONTAGGIO DI UN FILE SYSTEM	35
Utilizzo	35
Esempio	36
Utilizzo completo	36
4.3. COME SMONTARE UN FILE SYSTEM	39
Utilizzo	39
4.4. CONSIDERAZIONI PARTICOLARI DURANTE IL MONTAGGIO DEI FILE SYSTEM GFS2	39
4.5. GESTIONE QUOTA DEL GFS2	39
4.5.1. Configurazione dei disk quota	40
4.5.1.1. Impostazione dei quota in modalità Enforcement o Accounting.	40
Utilizzo	40
Esempi	41
4.5.1.2. Creazione dei file del Quota Database	41
4.5.1.3. Assegnazione dei quota per utente	41
4.5.1.4. Assegnazione dei quota per un gruppo	42
4.5.2. Gestione del Disk Quota	43
4.5.3. Mantenimento di un quota accurato	43
4.5.4. Sincronizzazione dei Quota con il comando quotasync	44
Utilizzo	44
Esempi	45
4.5.5. Riferimenti	45
4.6. COME ESPANDERE UN FILE SYSTEM	45
Utilizzo	46
Commenti	46
Esempi	46
Utilizzo completo	46
4.7. COME AGGIUNGERE I JOURNAL AD UN FILE SYSTEM	47
Utilizzo	47
Esempi	48
Utilizzo completo	48
4.8. DATA JOURNALING	48
4.9. CONFIGURAZIONE DEGLI AGGIORNAMENTI ATIME	49
4.9.1. Montaggio con relatime	50
Utilizzo	50
Esempio	50
4.9.2. Montaggio con noatime	50
Utilizzo	51
Esempio	51
4.10. SOSPENSIONE DI UNA ATTIVITÀ SU DI UN FILE SYSTEM	51

Utilizzo	51
Esempi	51
4.11. COME RIPRISTINARE UN FILE SYSTEM	51
Utilizzo	53
Esempio	53
4.12. MOUNT BIND E CONTEXT-DEPENDENT PATH NAMES	54
4.13. MOUNT BIND E ORDINE DI MONTAGGIO DI UN FILE SYSTEM	55
4.14. FUNZIONE WITHDRAW DI GFS2	57
<b>CAPITOLO 5. DIAGNOSI E CORREZIONI DEI PROBLEMI CON I FILE SYSTEM GFS2</b>	<b>59</b>
5.1. IL FILE SYSTEM GFS2 CON PRESTAZIONE LENTA	59
5.2. SOSPENSIONE DEL FILE SYSTEM GFS2 E RIAVVIO DI UN NODO	59
5.3. SOSPENSIONE DEL FILE SYSTEM GFS2 E RIAVVIO DI TUTTI I NODI	59
5.4. IMPOSSIBILE MONTARE IL FILE SYSTEM GFS2 SU UN NODO APPENA AGGIUNTO AL CLUSTER	60
5.5. SPAZIO INDICATO COME USATO IN UN FILE SYSTEM VUOTO	60
<b>CAPITOLO 6. CONFIGURAZIONE DI UN FILE SYSTEM GFS2 IN UN CLUSTER PACEMAKER</b>	<b>61</b>
<b>APPENDICE A. GESTIONE QUOTA DEL GFS2 CON IL COMANDO GFS2_QUOTA</b>	<b>63</b>
A.1. IMPOSTAZIONE DEI QUOTA CON IL COMANDO GFS2_QUOTA	63
Uso	63
Esempi	64
A.2. COME VISUALIZZARE L'UTILIZZO ED I LIMITI DEL QUOTA CON IL COMANDO GFS2_QUOTA	64
Uso	64
Output del comando	65
Commenti	65
Esempi	65
A.3. SINCRONIZZAZIONE DEL QUOTA CON IL COMANDO GFS2_QUOTA	66
Uso	66
Esempi	66
A.4. COME ABILITARE/DISABILITARE IL QUOTA ENFORCEMENT	67
Uso	67
Esempi	67
A.5. COME ABILITARE LA CONTABILITÀ DEI QUOTA	67
Uso	68
Esempio	68
<b>APPENDICE B. CONVERSIONE DI UN FILE SYSTEM DA GFS A GFS2</b>	<b>69</b>
B.1. CONVERSIONE DEI CONTEXT-DEPENDENT PATH NAMES	69
B.2. PROCEDURA CONVERSIONE DA GFS A GFS2	70
<b>APPENDICE C. TRACEPOINT GFS2 E DEBUGFS GLOCK FILE</b>	<b>71</b>
C.1. TIPI DI TRACEPOINT GFS2	71
C.2. TRACEPOINT	71
C.3. GLOCK	72
C.4. L'INTERFACCIA DEBUGFS DI GLOCK	73
C.5. HOLDER DI GLOCK	76
C.6. TRACEPOINT DI GLOCK	78
C.7. TRACEPOINT BMAP	78
C.8. TRACEPOINT DI LOG	78
C.9. STATISTICHE DI GLOCK	79
C.10. RIFERIMENTI	79
<b>APPENDICE D. DIARIO DELLE REVISIONI</b>	<b>81</b>

**INDICE ANALITICO** ..... **83**



# INTRODUZIONE

Questa guida fornisce le informazioni relative alla configurazione e gestione di Red Hat GFS2 (Red Hat Global File System 2), incluso nel Resilient Storage Add-On.

## 1. A CHI È RIVOLTO

Questo libro è rivolto principalmente agli amministratori dei sistemi Linux con una conoscenza delle seguenti attività:

- Procedure di amministrazione dei sistemi Linux, incluso la configurazione del kernel
- Installazione e configurazione delle reti per lo storage condiviso, come ad esempio le Fibre Channel SAN

## 2. DOCUMENTAZIONE CORRELATA

Per maggiori informazioni su come utilizzare Red Hat Enterprise Linux, consultare le seguenti risorse:

- *Installation Guide*— Fornisce le informazioni relative all'installazione di Red Hat Enterprise Linux 6.
- *Deployment Guide*— Fornisce le informazioni relative all'implementazione, configurazione e amministrazione di Red Hat Enterprise Linux 6.
- *Storage Administration Guide*— Fornisce le informazioni necessarie ad una gestione efficiente dei dispositivi di storage e file system su Red Hat Enterprise Linux 6.

Per maggiori informazioni su come utilizzare High Availability Add-On e Resilient Storage Add-On per Red Hat Enterprise Linux 6, consultare le seguenti risorse:

- *Panoramica sull'High Availability Add-On*— Fornisce una panoramica molto dettagliata sul Red Hat High Availability Add-On.
- *Amministrazione del Cluster*— Fornisce le informazioni relative all'installazione, configurazione e gestione di High Availability Add-On.
- *Amministrazione del Logical Volume Manager*— Fornisce una descrizione del Logical Volume Manager (LVM), ed include le informazioni su come eseguire LVM in un ambiente clusterizzato.
- *DM Multipath* — Fornisce le informazioni su come utilizzare il Device-Mapper Multipath di Red Hat Enterprise Linux.
- *Amministrazione Load Balancer*— Fornisce le informazioni necessarie per la configurazione di sistemi ad elevate prestazioni con il Load Balancer Add-On, un set di componenti software integrati il quale fornisce i Linux Virtual Servers (LVS) per il bilanciamento del carico IP su di un set di server reali.
- *Note di rilascio*— Fornisce le informazioni sulla release corrente dei prodotti di Red Hat.

La documentazione di Red Hat Cluster Suite ed altri documenti di Red Hat sono disponibili nelle versioni HTML, PDF, e RPM sul CD di documentazione di Red Hat Enterprise Linux, e online su <https://access.redhat.com/site/documentation/>.

### 3. ABBIAMO BISOGNO DEI VOSTRI COMMENTI!

Se individuate degli errori di battitura o se pensate di poter contribuire al miglioramento di questa guida, contattateci subito. Inviare un report in Bugzilla: <http://bugzilla.redhat.com/> sul componente **Red Hat Enterprise Linux 6** e **doc-Global\_File\_System\_2**. Quando inviate un bug report assicuratevi di indicare l'identificatore del manuale:

```
rh-gfs2(EN)-6 (2014-10-8T15:15)
```

Se avete dei suggerimenti per migliorare la documentazione, cercate di essere il più specifici possibile. Se avete trovato un errore, vi preghiamo di includere il numero della sezione e alcune righe di testo, in modo da agevolare la ricerca dell'errore stesso.

## CAPITOLO 1. PANORAMICA SUL GFS2

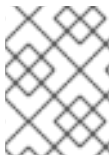
Il file system GFS2 di Red Hat è incluso nel Resilient Storage Add-On. Esso è un file system nativo che interfaccia direttamente con il file system del kernel di Linux (VFS layer). Quando implementato come file system del cluster, GFS2 implementa i metadati distribuiti e journal multipli. Red Hat supporta l'uso di file system GFS2 come implementato nella High Availability Add-On.



### NOTA

Anche se il file system GFS2 può essere implementato in un sistema standalone o come parte di una configurazione del cluster, per la release di Red Hat Enterprise Linux 6, Red Hat non supporta l'uso del GFS2 come file system con nodo singolo. Red Hat supporta un numero di file system di nodi singoli ad elevate prestazioni ottimizzati per i singoli nodi e quindi con un overhead più basso rispetto ad un file system del cluster. Red Hat consiglia l'uso dei suddetti file system rispetto al GFS2 nel caso in cui si verifica la necessità di un montaggio del file system da parte del nodo.

Red Hat continuerà il suo supporto di file system GFS2 con nodo singolo per il montaggio delle snapshot dei file system del cluster (per esempio per scopi di backup)



### NOTA

Red Hat non supporta l'uso del GFS2 per implementazioni del file system del cluster maggiori a 16 nodi.

GFS2 si basa su una architettura a 64-bit la quale può in teoria ospitare un file system di 8 EB. Tuttavia la dimensione massima supportata corrente di un file system GFS2 per un hardware a 64-bit è 100 TB. La dimensione massima corrente supportata di un file system GFS2 per un hardware a 32-bit è 16TB. Se il vostro sistema richiede file system GFS2 maggiori è consigliato contattare un rappresentante di Red Hat.

Quando determinate la dimensione del vostro file system è consigliato considerare i vostri requisiti per un processo di ripristino. L'esecuzione del comando `fsck.gfs2` su di un file system molto grande, richiederà molto tempo e consumerà una quantità molto grande di memoria. Altresì, nell'evento di un errore del disco o del sottosistema-disco, il tempo di ripristino sarà limitato dalla velocità del vostro dispositivo di backup. Per informazioni sulla quantità di memoria richiesto dal comando `fsck.gfs2` consultate [Sezione 4.11, «Come ripristinare un file system»](#).

Quando configurati in un cluster i nodi di Red Hat GFS2 possono essere configurati e gestiti con i tool di configurazione e gestione di High Availability Add-On. Red Hat GFS2 permette una condivisione dei dati tra i nodi GFS2 in un cluster con una vista singola ed uniforme dello spazio del nome del file system attraverso i nodi GFS2. Ciò permette ai processi presenti sui diversi nodi di condividere i file GFS2, in modo simile alla condivisione dei file su di un file system locale con nessuna differenza da parte dei processi presenti sullo stesso nodo. Per informazioni su High Availability Add-On consultare la *Configurazione e gestione di un Red Hat Cluster*

Mentre un file system GFS2 può essere usato esternamente al LVM, Red Hat supporta solo i file system GFS2 creati su di un volume logico CLVM. CLVM è incluso nel Resilient Storage Add-On. Esso è una implementazione di LVM dell'intero cluster abilitato dal demone `clvmd`, il quale gestisce i volumi logici LVM in un cluster. Il demone rende possibile l'utilizzo di LVM2 per la gestione dei volumi logici attraverso il cluster, permettendo a tutti i nodi presenti nel cluster stesso di condividere i volumi logici su ogni directory presente sul sistema. Per informazioni sul volume manager LVM, consultate la *Guida dell'amministratore di LVM*.

Modulo kernel `gfs2.ko` implementa il file system GFS2, ed è caricato sui nodi del cluster GFS2.



## NOTA

Quando configurate un file system GFS2 come un file system del cluster assicuratevi che tutti i nodi presenti in un cluster abbiano accesso allo storage condiviso. Configurazioni asimmetriche nelle quali solo alcuni nodi hanno un accesso allo storage condiviso, non saranno supportate. Ciò non richiede il montaggio da parte di tutti i nodi del file system GFS2 stesso.

Questo capitolo contiene alcune informazioni di base abbreviate come background per aiutarvi a comprendere meglio il GFS2. Sono presenti le seguenti sezioni:

- [Sezione 1.1, «Funzioni nuove e modificate»](#)
- [Sezione 1.2, «Prima d'impostare il GFS2»](#)
- [Sezione 1.4, «Differenze tra GFS e GFS2»](#)
- [Sezione 1.3, «Installazione di GFS2»](#)
- [Sezione 2.9, «Blocco dei nodi GFS2»](#)

## 1.1. FUNZIONI NUOVE E MODIFICATE

Questa sezione riporta le funzioni nuove e quelle modificate del file system GFS2 insieme alla relativa documentazione inclusi con la release iniziale e quella successiva di Red Hat Enterprise Linux 6.

### 1.1.1. Funzioni nuove e modificate di Red Hat Enterprise Linux 6.0

Red Hat Enterprise Linux 6.0 include le seguenti modifiche ed aggiornamenti.

- Per la release di Red Hat Enterprise Linux 6, Red Hat non supporta l'uso del GFS2 come file system con un singolo nodo.
- Per la release di Red Hat Enterprise Linux 6 il comando `gfs2_convert` per l'aggiornamento da GFS ad un file system GFS2 è stato migliorato. Per informazioni su questo comando consultare [Appendice B, Conversione di un file system da GFS a GFS2](#)
- La release Red Hat Enterprise Linux 6 supporta le opzioni di mount `discard`, `nodiscard`, `barrier`, `nobarrier`, `quota_quantum`, `statfs_quantum`, e `statfs_percent`. Per informazioni sul montaggio di un file system GFS2 consultare [Sezione 4.2, «Montaggio di un file system»](#).
- La versione di Red Hat Enterprise Linux 6 di questo documento contiene una nuova sezione [Sezione 2.9, «Blocco dei nodi GFS2»](#). Questa sezione riporta alcune informazioni più dettagliate dei file system GFS2.

### 1.1.2. Funzioni nuove e modificate di Red Hat Enterprise Linux 6.1

Red Hat Enterprise Linux 6.1 include le seguenti modifiche ed aggiornamenti.

- Con Red Hat Enterprise Linux 6.1 è ora disponibile il supporto delle funzioni standard dei quota di Linux. La gestione del quota del GFS2 è documentata in [Sezione 4.5, «Gestione quota del GFS2»](#).

Per release precedenti di Red Hat Enterprise Linux, GFS2 richiedeva l'uso del comando `gfs2_quota` per gestire i quota. La documentazione di `gfs2_quota` è ora disponibile in [Appendice A, Gestione quota del GFS2 con il comando `gfs2\_quota`](#).

- Questo documento contiene ora un nuovo capitolo, [Capitolo 5, Diagnosi e correzioni dei problemi con i file system GFS2](#).
- Sono stati apportati su tutto il documento piccole correzioni e chiarimenti.

### 1.1.3. Funzioni nuove e modificate di Red Hat Enterprise Linux 6.2

Red Hat Enterprise Linux 6.2 include le seguenti modifiche e gli aggiornamenti relativi alla documentazione ed alle funzioni.

- Con la versione di Red Hat Enterprise Linux 6.2, GFS2 è in grado di supportare l'uso del comando `tunegfs2`, il quale sostituisce alcune delle funzioni del comando `gfs2_tool`. Per maggiori informazioni consultare la pagina man di `tunegfs2`.

Le seguenti sezioni sono state aggiornate e forniscono ora le procedure amministrative che non necessitano del comando `gfs2_tool`:

- [Sezione 4.5.4, «Sincronizzazione dei Quota con il comando `quotasync`](#). e [Sezione A.3, «Sincronizzazione del quota con il comando `gfs2\_quota`](#) descrivono ora come modificare il parametro `quota_quantum` dal proprio valore predefinito di 60 secondi usando l'opzione di montaggio `quota_quantum=`.
- [Sezione 4.10, «Sospensione di una attività su di un file system»](#) descrive ora come sospendere l'attività di scrittura per un file system, utilizzando il comando `dmsetup suspend`.
- Questo documento include una nuova appendice, [Appendice C, Tracepoint GFS2 e debugfs glock File](#), la quale descrive l'interfaccia `debugfs` di `glock` e i tracepoint di GFS2. È rivolto agli utenti esperti che desiderano approfondire le conoscenze sul design e i metodi per risolvere le problematiche specifiche al file system GFS2.

### 1.1.4. Funzioni nuove e modificate di Red Hat Enterprise Linux 6.3

Per la versione di Red Hat Enterprise Linux 6.3, questo documento contiene un nuovo capitolo, [Capitolo 2, Considerazioni operative e configurazione del GFS2](#). Il suddetto capitolo rende disponibili i consigli per ottimizzare le prestazioni di GFS2, e per la creazione, utilizzo e gestione di un file system GFS2.

In aggiunta sono state apportate piccole correzioni e chiarimenti su tutto il documento.

### 1.1.5. Funzioni nuove e modificate di Red Hat Enterprise Linux 6.4

È stato aggiornato nella versione Red Hat Enterprise Linux 6.4 il [Capitolo 2, Considerazioni operative e configurazione del GFS2](#).

### 1.1.6. Funzionalità nuove e aggiornate per Red Hat Enterprise Linux 6.6

Per la versione di Red Hat Enterprise Linux 6.6, questo documento contiene un nuovo capitolo, [Capitolo 6, Configurazione di un file system GFS2 in un cluster Pacemaker](#). Il suddetto capitolo fornisce le fasi necessarie per impostare un cluster Pacemaker che include un file system GFS2.

In aggiunta sono state apportate piccole correzioni e chiarimenti su tutto il documento.

## 1.2. PRIMA D'IMPOSTARE IL GFS2

Prima di poter installare ed impostare GFS2, è necessario essere a conoscenza delle seguenti caratteristiche dei vostri file system GFS2:

### Nodi GFS2

Determina quali nodi presenti nel cluster monteranno i file system GFS2.

### Numero di file system

Determina il numero iniziale di file system GFS2 da creare. (È possibile aggiungere successivamente un numero superiore di file system.)

### Nome del file system

Determina un nome unico per ogni file system. Il nome deve essere unico per tutti i filesystem `lock_dlm` attraverso il cluster. Ogni nome del file system deve essere sottoforma di una variabile del parametro. Per esempio, questo manuale utilizza i nomi del file system `mydata1` e `mydata2` in alcuni esempi.

### Journal

Determina il numero di journal per i file system GFS2. Per ogni nodo che esegue il montaggio del file system GFS2 è necessario un journal. GFS2 permette di aggiungere dinamicamente i journal nelle fasi successive, poichè i server aggiuntivi eseguiranno il montaggio di un file system. Per maggiori informazioni su come aggiungere journal ad un file system GFS2 consultate la [Sezione 4.7, «Come aggiungere i journal ad un file system»](#).

### Dispositivi di storage e partizioni

Determina i dispositivi di storage e le partizioni da usare per la creazione di volumi logici (via CLVM) nei file system.



### NOTA

Con GFS2 è possibile avere problemi di prestazione quando numerose operazioni di creazione e rimozione sono eseguite contemporaneamente da più di un nodo nella stessa directory. Se ciò causa problemi di prestazione del sistema, e quando possibile, è consigliato localizzare la rimozione e la creazione di file da parte di un nodo per le directory specifiche al nodo in questione.

Per maggiori informazioni sulla creazione, uso e gestione di un file system GFS2 consultare [Capitolo 2, Considerazioni operative e configurazione del GFS2](#).

## 1.3. INSTALLAZIONE DI GFS2

Oltre ai pacchetti richiesti per il Red Hat High Availability Add-On, sarà necessario installare il pacchetto `gfs2-utils` per GFS2 e `lvm2-cluster` per il Clustered Logical Volume Manager (CLVM).

I pacchetti `lvm2-cluster` e `gfs2-utils` appartengono al canale ResilientStorage il quale deve essere abilitato prima di installare i pacchetti.

Usare il seguente comando `yum install` per installare i pacchetti del software di Red Hat High Availability Add-On:

```
# yum install rgmanager lvm2-cluster gfs2-utils
```

Per informazioni generali sulla gestione del cluster e su Red Hat High Availability Add-On, consultare il manuale *Amministrazione del cluster*.

## 1.4. DIFFERENZE TRA GFS E GFS2

Questa sezione elenca i miglioramenti e le modifiche offerte da GFS2 rispetto a GFS.

La migrazione da GFS a GFS2 necessita di una conversione del vostro file system GFS in GFS2 attraverso l'utilità `gfs2_convert`. Per informazioni sull'utilità `gfs2_convert`, consultare [Appendice B, Conversione di un file system da GFS a GFS2](#)

### 1.4.1. Nomi dei comandi di GFS2

In generale, la funzionalità di GFS2 è identica al GFS. Tuttavia i nomi dei comandi del file system specificano GFS2 invece di GFS. [Tabella 1.1, «Comandi GFS2 e GFS»](#) mostra il GFS equivalente ed i comandi GFS2.

Tabella 1.1. Comandi GFS2 e GFS

Comando GFS	Comando GFS2	Descrizione
<code>mount</code>	<code>mount</code>	Esegue un mount del file system. Il sistema è in grado di determinare se il file system è di tipo GFS o GFS2. Per informazioni sulle opzioni di montaggio di GFS2 consultate la pagina man di <code>gfs2_mount(8)</code> .
<code>umount</code>	<code>umount</code>	Smonta un file system.
<div style="border: 1px solid black; padding: 2px; margin-bottom: 2px;"><code>fsck</code></div> <div style="border: 1px solid black; padding: 2px;"><code>gfs_fsck</code></div>	<div style="border: 1px solid black; padding: 2px; margin-bottom: 2px;"><code>fsck</code></div> <div style="border: 1px solid black; padding: 2px;"><code>fsck.gfs2</code></div>	Controlla e ripara un file system non montato.
<code>gfs_grow</code>	<code>gfs2_grow</code>	Espande un file system montato.
<code>gfs_jadd</code>	<code>gfs2_jadd</code>	Aggiunge un journal ad un file system montato.
<div style="border: 1px solid black; padding: 2px; margin-bottom: 2px;"><code>gfs_mkfs</code></div> <div style="border: 1px solid black; padding: 2px;"><code>mkfs -t gfs</code></div>	<div style="border: 1px solid black; padding: 2px; margin-bottom: 2px;"><code>mkfs.gfs2</code></div> <div style="border: 1px solid black; padding: 2px;"><code>mkfs -t gfs2</code></div>	Crea un file system su di un dispositivo di storage.

Comando GFS	Comando GFS2	Descrizione
<b>gfs_quota</b>	<b>gfs2_quota</b>	Gestisce il quota su un file system montato. Con Red Hat Enterprise Linux 6.1 GFS2 supporta le funzioni standard del quota di Linux. È ora disponibile il supporto delle funzioni standard dei quota di Linux. La gestione del quota del GFS2 è documentata in <a href="#">Sezione 4.5, «Gestione quota del GFS2»</a> .
<b>gfs_tool</b>	<b>tunegfs2</b>  parametri di montaggio  <b>dmsetup suspend</b>	Configura, regola, o raccoglie informazioni relative al file system. Con Red Hat Enterprise Linux 6.2, <b>tunegfs2</b> è supportato. È presente altresì il comando <b>gfs2_tool</b> .
<b>gfs_edit</b>	<b>gfs2_edit</b>	Mostra, stampa o modifica le strutture interne del file system. Il comando <b>gfs2_edit</b> può essere usato sia per i file system GFS che per i file system GFS2.
<b>gfs_tool setflag jdata/inherited_jdata</b>	<b>chattr +j</b> (preferito)	Abilita il journaling su di un file o directory.
<b>setfacl/getfacl</b>	<b>setfacl/getfacl</b>	Imposta o acquisisce l'access control list del file per un file o directory.
<b>setfattr/getfattr</b>	<b>setfattr/getfattr</b>	Imposta o acquisisce gli attributi estesi di un file.

Per un elenco completo delle opzioni supportate per i comandi del file system GFS2, consultare le pagine man dei comandi in questione.

### 1.4.2. Differenze aggiuntive tra GFS e GFS2

Questa sezione riassume le differenze aggiuntive presenti nella gestione di GFS e GFS2 non riportate in [Sezione 1.4.1, «Nomi dei comandi di GFS2»](#).

#### Context-Dependent Path Names

I file system GFS2 non forniscono alcun supporto per i context-dependent path name, i quali permettono all'utente di creare link simbolici per directory o file con destinazione variabile. Per questa funzionalità in GFS2 è possibile utilizzare l'opzione **bind** del comando **mount**. Per informazioni sui bind mounts ed i nomi del percorso che dipendono dal contesto in GFS2, consultare [Sezione 4.12, «Mount Bind e Context-Dependent Path Names»](#).

#### Modulo gfs2.ko

Il modulo del kernel che implementa il file system GFS è **gfs.ko**. Il modulo del kernel che implementa il file system GFS2 è **gfs2.ko**.



## Come abilitare il Quota Enforcement in GFS2

Nei file system GFS2 il quota enforcement viene disabilitato per default e deve essere esplicitamente abilitato. Per informazioni su come abilitare e disabilitare il quota enforcement consultare la [Sezione 4.5, «Gestione quota del GFS2»](#).

## Data Journaling

I file system GFS2 supportano l'utilizzo del comando `chattr` per impostare e rimuovere il flag `j` su di un file o directory. L'impostazione del flag `+j` su di un file abilita il data journaling su quel file. L'impostazione del flag `+j` su di una directory significa "inherit jdata", il quale indica che tutti i file e le directory successivamente create nella directory in questione vengono salvati all'interno del journal. L'utilizzo del comando `chattr` rappresenta il modo migliore per abilitare e disabilitare il data journaling su di un file.

## Come aggiungere i journal dinamicamente

Nei file system GFS i journal sono metadati interni presenti esternamente al file system che rendono necessaria l'estensione della dimensione del volume logico che contiene il file system prima di aggiungere i journal. Nei file system GFS2 i journal sono file semplici (nascosti). Ciò significa che sarà possibile aggiungere dinamicamente i journal poichè i server supplementari possono eseguire il mount di un file system fino a quando è presente spazio sufficiente sul file system per i journal aggiuntivi. Per informazioni su come aggiungere i journal ad un file system GFS2, consultate la [Sezione 4.7, «Come aggiungere i journal ad un file system»](#).

## parametro `atime_quantum` rimosso

Il file system GFS2 non supporta il parametro regolabile `atime_quantum`, il quale può essere utilizzato dal file system GFS per specificare la cadenza degli aggiornamenti `atime`. GFS2 supporta le opzioni di montaggio `relatime` e `noatime`. L'opzione di montaggio `relatime` è consigliata per avere un comportamento simile all'impostazione del parametro `atime_quantum` in GFS.

## `data=` opzione del comando `mount`

Quando si esegue il montaggio del file system GFS2, è possibile specificare l'opzione `data=ordered` o `data=writeback` del comando `mount`. Una volta impostato `data=ordered`, i dati dell'utente modificati da una transazione verranno scaricati sul disco prima di confermare la transazione sul disco stesso. Tale operazione dovrebbe impedire una visualizzazione da parte dell'utente, dei blocchi non inizializzati all'interno di un file dopo il verificarsi di un crash. Se `data=writeback` è stato impostato i dati verranno scritti in qualsiasi momento sul disco, dopo che lo stesso è stato già utilizzato. Tale operazione non garantisce una consistenza simile a quella garantita dalla modalità `ordered`, ma dovrebbe essere più veloce sotto alcuni carichi di lavoro. La modalità predefinita è `ordered`.

## Il comando `gfs2_tool`

Il comando `gfs2_tool` supporta un set diverso di opzioni per GFS2 rispetto al comando `gfs_tool` per GFS:

- Il comando `gfs2_tool` supporta un parametro `journals` il quale stampa le informazioni relative al journal attualmente configurato, incluso il numero di journal contenuti da un file system.
- Il comando `gfs2_tool` non supporta il flag `counters`, usato dal comando `gfs_tool` per visualizzare le statistiche di GFS.

- Il comando `gfs2_tool` non supporta il flag `inherit_jdata`. Per impostare il flag di una directory in modo da indicare "inherit jdata", impostare il flag `jdata` sulla directory o usare il comando `chattr` per impostare il flag `+j`. L'utilizzo del comando `chattr` rappresenta il modo migliore per abilitare o disabilitare il data journaling su di un file.



## NOTA

Con Red Hat Enterprise Linux 6.2, GFS2 è in grado di supportare l'uso del comando `tunegfs2`, il quale sostituisce alcune delle funzioni del comando `gfs2_tool`. Per maggiori informazioni consultare la pagina `man (8) di tunegfs2`. Le funzioni `set tune` e `get tune` del comando `gfs2_tool`, sono state sostituite dalle opzioni della linea di comando di `mount`, è possibile ora una loro impostazione mediante il file `fstab` quando necessario.

## Il comando `gfs2_edit`

Il comando `gfs2_edit` supporta un set diverso di opzioni per GFS2 rispetto al comando `gfs_dit` per GFS. Per informazioni su opzioni specifiche supportate da ogni versione del comando consultare le pagine `man gfs2_edit` e `gfs_edit`.

### 1.4.3. Miglioramenti delle prestazioni di GFS2

Sono presenti numerose funzionalità nei file system GFS2 le quali non differiscono nell'interfaccia utente rispetto ai file system GFS e garantiscono una migliore prestazione del file system.

Un file system GFS2 fornisce una migliore prestazione del file system nei termini di seguito riportati:

- Migliore prestazione sotto condizioni di utilizzo intenso in una directory singola.
- Operazioni I/O sincrone più veloci
- Letture dati in cache più veloci (senza locking overhead)
- I/O diretto più veloce con file preassegnati (con una dimensione I/O ragionevolmente grande, come ad esempio 4M di blocchi)
- Operazioni I/O più veloci in generale
- Esecuzione più veloce del comando `df`, dovuto a chiamate `statfs` più veloci.
- La modalità `atime` è stata migliorata in modo da ridurre il numero di operazioni I/O di scrittura generati da `atime` quando confrontato con GFS.

Il file system GFS2 fornisce un supporto più ampio e dettagliato nelle seguenti modalità:

- GFS2 è parte del kernel upstream (integrato in 2.6.19).
- GFS2 supporta le seguenti caratteristiche.
  - attributi estesi del file (`xattr`)
  - le impostazioni degli attributi `lsattr()` e `chattr()` tramite le chiamate standard `ioctl()`.
  - timestamp in nanosecondi

Un file system GFS2 migliora l'efficienza interna del file system.

- GFS2 utilizza una quantità di memoria minore del kernel.
- GFS2 non necessita di alcun numero di generazione dei metadata.

L'assegnazione dei metadata GFS2 non necessita di alcun processo di lettura. Le copie delle sezioni dei metadata nei journal multipli, vengono gestite attraverso la revoca delle sezioni dal journal prima del rilascio del blocco.

- GFS2 include un log manager molto più semplice il quale non è a conoscenza degli inode non collegati o delle modifiche dei quota.
- I comandi `gfs2_grow` e `gfs2_jadd` utilizzano il locking per evitare l'esecuzione simultanea di istanze multiple.
- Il codice ACL è stato semplificato per chiamate `creat()` e `mkdir()`.
- Gli inode non collegati, le modifiche dei quota e le modifiche `statfs` vengono ripristinati senza montare nuovamente il journal.

## CAPITOLO 2. CONSIDERAZIONI OPERATIVE E CONFIGURAZIONE DEL GFS2

Global File System 2 (GFS2) permette a diversi computer ("nodi") in un cluster di condividere lo stesso storage. Per raggiungere questo tipo di cooperazione e mantenere una certa consistenza dei dati tra i nodi, viene utilizzato uno schema di locking per l'intero cluster per le risorse del file system. Questo schema utilizza alcuni protocolli di comunicazione come TCP/IP, per lo scambio di informazioni.

Per migliorare le prestazioni seguire i consigli riportati in questo capitolo, inclusi quelli relativi alla creazione, utilizzo e gestione di un file system GFS2.



### IMPORTANTE

Assicuratevi che l'implementazione di Red Hat High Availability Add-On possa essere supportata ed in grado di soddisfare i vostri requisiti. Consultate un rappresentante autorizzato di Red Hat per una verifica della configurazione prima dell'impiego.

## 2.1. CONSIDERAZIONI SULLA FORMATTAZIONE

Questa sezione fornisce le informazioni su come formattare il file system GFS2 per il miglioramento delle prestazioni.

### 2.1.1. Dimensione del file system: è meglio avere una dimensione più piccola

GFS2 si basa su una architettura a 64-bit la quale può in teoria ospitare un file system di 8 EB. Tuttavia la dimensione massima supportata corrente di un file system GFS2 per un hardware a 64-bit è 100 TB. La dimensione massima corrente supportata di un file system GFS2 per un hardware a 32-bit è 16TB.

Anche se è possibile creare un file system GFS2 molto grande, questa operazione non è consigliata. La regola generale è quella di usare dimensioni più piccole: è meglio avere 10 file system di 1TB che uno di 10TB.

Diversi sono i motivi per i quali è consigliato avere una dimensione piccola per i file system GFS2:

- È necessario un minor tempo per il back up di ogni file system.
- Quantità di tempo minore per il controllo del file system con il comando `fsck.gfs2`.
- Minore quantità di memoria necessaria per controllare il file system con il comando `fsck.gfs2`.

Numero minore di gruppi di risorse per migliori prestazioni.

Se il GFS2 è troppo piccolo potreste avere problemi di spazio e quindi dover risolvere problemi in tal senso. Prima di decidere la misura del file system determinarne il tipo di utilizzo.

### 2.1.2. Dimensione blocco: È preferito il default di (4K) blocchi.

Con Red Hat Enterprise Linux 6 il comando `mkfs.gfs2` tenta di stimare una dimensione ottimale del blocco in base al tipo di dispositivo usato. In generale, blocchi da 4K sono la dimensione preferita poichè 4K è la dimensione predefinita della pagina (memoria) per Linux. Diversamente da altri file system, GFS2 esegue la maggior parte delle sue operazioni usando kernel buffer da 4K. Se la dimensione del blocco che utilizzate è di 4K, il kernel dovrà lavorare meno per manipolare i buffer.

È consigliato usare la dimensione predefinita del blocco per avere migliori prestazioni. Usare una dimensione diversa solo se è necessario avere uno storage efficiente per numerosi file di piccole dimensioni.

### 2.1.3. Numero di journal: Uno per ogni nodo che esegue il montaggio

GFS2 ha bisogno di un journal per ogni nodo del cluster che deve montare il file system. Per esempio, se siete in presenza di un cluster a 16 nodi ma avete bisogno di montare solo il file system di due nodi, in tal caso avrete bisogno solo di due journal. Se è necessario eseguire il montaggio da un terzo nodo, sarà sempre possibile aggiungere un journal con il comando `gfs2_jadd`. Con GFS2 è possibile aggiungere i journal in modo istantaneo.

### 2.1.4. Dimensione journal: Predefinita (128MB), generalmente risulta essere ottimale

Quando eseguite il comando `mkfs.gfs2` per creare un file system GFS2, è possibile specificare la dimensione del journal. Se non specificate la dimensione del journal verrà impostato come default 128MB. Questa dimensione dovrebbe essere ottimale per la maggior parte delle applicazioni.

Alcuni amministratori di sistema possono pensare che 128MB siano eccessivi e ridurre la dimensione a 8MB o a 32MB. Anche se questa operazione può funzionare, essa può abbassare le prestazioni. Come numerosi altri file system con journal, ogni qualvolta GFS2 scrive i metadati, quest'ultimi verranno salvati sul journal prima di poterli utilizzare. Ciò significa che se il sistema si arresta inaspettatamente o perde alimentazione, sarà possibile recuperare tutti i metadati quando si riutilizza il journal al momento del montaggio. Tuttavia un journal di 8 MB può non avere più spazio disponibile in tempi molto brevi, e quando il journal è pieno, le prestazioni diminuiscono poichè GFS2 dovrà attendere prima di poter utilizzare lo storage.

È generalmente consigliato l'uso della dimensione predefinita del journal di 128MB. Se il file system è molto piccolo (5GB) l'uso di 128MB potrebbe non essere pratico. Se al contrario il file system è più grande, l'utilizzo di 256MB potrebbe aumentare le prestazioni.

### 2.1.5. Dimensione e numero dei gruppi delle risorse

Al momento della creazione di un GFS2 tramite il comando `mkfs.gfs2`, lo storage verrà suddiviso in sezioni uniformi conosciute come gruppi di risorse. Durante questo processo verrà stimata una dimensione del gruppo ottimale (che varia da 32MB a 2GB). È possibile annullare l'impostazione predefinita con l'opzione `-r` del comando `mkfs.gfs2`.

La dimensione del gruppo di risorse ottimale dipende dal tipo di uso del file system. Considerate il tipo di utilizzo e la sua frammentazione.

Controllate quale delle dimensioni del gruppo di risorse è quella ideale per le prestazioni. Per fare questo utilizzate un cluster di prova prima di implementare GFS2 in ambienti di produzione.

Se il file system presenta un numero molto alto di gruppi (ognuno dei quali risulta essere molto piccolo), l'assegnazione del blocco potrà richiedere una quantità di tempo molto elevata a causa della ricerca del blocco libero tra migliaia (o centinaia di migliaia) di gruppi. Più il file system è pieno, maggiore sarà il numero di gruppi da ricercare, e ognuno di essi avrà bisogno di un blocco dell'intero cluster. Questa operazione rallenterà notevolmente le prestazioni.

Se invece il file system presenta un numero limitato di gruppi (ognuno dei quali è molto grande), si potrà verificare più frequentemente una contesa per lo stesso blocco del gruppo di risorse, abbassando così il livello delle prestazioni. Per esempio, se siete in possesso di un file system di 10GB con 5 gruppi di risorse da 2GB, i nodi presenti nel cluster si contenderanno più spesso i cinque gruppi,

rispetto ad un file system composto da 320 gruppi da 32MB. Il problema potrebbe diventare più acuto se il file system è quasi pieno poiché durante l'assegnazione del blocco sarà necessario controllare ogni gruppo di risorse prima di poter trovarne uno con un blocco disponibile. GFS2 cerca di mitigare questo problema in due modi:

- Come prima cosa quando un gruppo di risorse è pieno cercherà di non controllare l'assegnazione futura (fino a quando il blocco non risulta disponibile). Se i file non verranno mai rimossi, la contesa risulterà essere meno severa. Tuttavia, se l'applicazione usata cancella costantemente i blocchi assegnandone di nuovi su un file system quasi pieno, la contesa risulterà essere molto alta con un conseguente rallentamento delle prestazioni.
- Secondo, quando nuovi blocchi vengono aggiunti ad un file esistente, GFS2 cercherà di raggruppare i nuovi blocchi nello stesso gruppo di risorse del file. Questa operazione viene eseguita per aumentare le prestazioni: su un disco in rotazione, le ricerche richiederanno minor tempo se risultano essere fisicamente vicini.

Nei casi peggiori, quando è presente una directory centrale nella quale tutti i nodi creano i file, e i nodi entrano in contesa cercando di bloccare lo stesso gruppo di risorse.

## 2.2. FRAMMENTAZIONE DEL FILE SYSTEM

Red Hat Enterprise Linux 6.4 introduce alcuni miglioramenti nella gestione della frammentazione del file in GFS2. Con Red Hat Enterprise Linux 6.4 le azioni di scrittura simultanea risultano in una frammentazione minore, e quindi in una migliore prestazione durante questi carichi di lavoro.

Anche se non è ancora presente uno strumento di deframmentazione per GFS2 su Red Hat Enterprise Linux, è possibile deframmentare singoli file identificandoli con lo strumento filefrag, compiandoli su file temporanei, modificandone i nomi e sostituendo gli originali. (È possibile eseguire questa procedura anche su versioni precedenti di Red Hat Enterprise Linux 6.4 se i processi di scrittura vengono eseguiti in maniera sequenziale).

## 2.3. PROBLEMATICHE RELATIVE ALL'ASSEGNAZIONE DEL BLOCCO

Questa sezione riporta un sommario delle problematiche relative all'assegnazione del blocco nei file system GFS2. Anche se alle applicazioni che scrivono dati generalmente non interessa come e dove viene assegnato un blocco, una conoscenza generica sul processo di assegnazione potrebbe assistere l'utente a migliorare le prestazioni.

### 2.3.1. Avere spazio disponibile nel file system

Quando un file system è quasi pieno il processo di assegnazione del blocco si complica notevolmente. Ne deriva che, i blocchi assegnati vengono posizionati alla fine del gruppo di risorse o in sezioni molto piccole dove la frammentazione è molto più probabile. La frammentazione del file può causare problemi alle prestazioni. In aggiunta, quando un GFS2 è quasi pieno l'assegnatore del blocco richiede una quantità di tempo maggiore per la ricerca attraverso gruppi di risorse multipli. Questa operazione genererà una contesa del blocco che potrebbe essere evitata se in presenza di un file system con molto spazio disponibile. Tutto questo può causare un peggioramento delle prestazioni.

Per questi motivi non è consigliato utilizzare un file system con un utilizzo uguale o maggiore dell'85%. Da notare che questa figura potrebbe variare in base al carico di lavoro.

### 2.3.2. Se possibile ogni nodo deve assegnare i propri file

A causa del funzionamento del distributed lock manager (DLM), si possono verificare un numero più elevato di contese se tutti i file vengono assegnati da un nodo, e altri nodi devono aggiungere i blocchi ai file in questione.

Con GFS (versione 1), tutti i blocchi erano gestiti da un gestore centrale e la sua funzione principale era quella di controllare il locking su tutto il cluster. Il grand unified lock manager (GULM) poteva però rappresentare un punto singolo d'errore. Con il nuovo schema di locking di GFS2, DLM estende i blocchi su tutto il cluster. In presenza di un errore in un nodo del cluster, i rispettivi blocchi verranno ripristinati da altri nodi.

Con DLM, il primo nodo che esegue il blocco di una risorsa (come un file) diventa il "lock master" per quel blocco. Altri nodi saranno in grado di bloccare la risorsa in questione, ma essi dovranno chiedere prima il permesso del lock master. Ogni nodo è a conoscenza dei lock master relativi e ogni nodo è a conoscenza del nodo al quale ha prestato il blocco. Il locking di un blocco sul nodo master è più veloce di un locking su un altro nodo che deve arrestarsi e chiedere il permesso al lock master.

Come in numerosi file system l'assegnatore del GFS2 prova a mantenere i blocchi di un file vicini l'un l'altro, riducendo così il movimento delle testine di un disco e aumentando le prestazioni. Un nodo che assegna i blocchi ad un file probabilmente avrà bisogno di utilizzare e bloccare gli stessi gruppi di risorse per i nuovi blocchi (a meno che tutti i blocchi in quel gruppo sono in uso). Il file system sarà più veloce se il lock master del gruppo contenente il file assegna i propri blocchi dati (e cioè, è più veloce se si ha il nodo che per primo ha aperto il file, eseguire le azioni di scrittura dei nuovi blocchi).

### 2.3.3. Se possibile, preassegnate

Se i file sono preassegnati, sarà possibile non allocare il blocco e avere una esecuzione del file system più efficiente. Le versioni più recenti di GFS2 includono la chiamata del sistema `faallocate(1)`, utilizzabile per preassegnare i blocchi di dati.

## 2.4. CONSIDERAZIONI SUL CLUSTER

Quando determinate il numero dei nodi presenti in un sistema, è importante tener presente il compromesso tra prestazioni e alta disponibilità. Con un numero elevato di nodi può diventare sempre più difficile gestire i carichi di lavoro. Per questo motivo Red Hat non supporta l'uso di GFS2 per implementazioni in un cluster maggiori di 16 nodi.

L'implementazione di un file system del cluster non è una sostituzione "semplice" per un nodo singolo. È consigliato un periodo di prova di 8-12 settimane delle nuove installazioni per testare il sistema ed assicurare un suo corretto funzionamento ad un livello di prestazione desiderato. Durante questo periodo qualsiasi problema funzionale o di prestazione può essere risolto e qualsiasi domanda inviata al team di supporto di Red Hat.

È consigliato altresì agli utenti che desiderano implementare i cluster, un controllo delle configurazioni da parte del team di supporto di Red Hat prima dell'impiego per evitare qualsiasi problema di supporto futuro.

## 2.5. CONSIDERAZIONI SULL'USO

La seguente sezione rende disponibili alcuni consigli generali sull'uso di GFS2.

### 2.5.1. Opzioni di montaggio: `noatime` e `nodiratime`

In generale è consigliato montare i file system GFS2 con `noatime` e `nodiratime`. Ciò permetterà a GFS2 di utilizzare una quantità di tempo minore per l'aggiornamento degli inode del disco per ogni accesso.

## 2.5.2. Opzioni ottimizzazione DLM: Aumento dimensioni della tabella DLM

DLM utilizza diverse tabelle per la gestione, coordinazione e inoltro delle informazioni relative al lock tra i nodi presenti in un cluster. L'aumento della dimensione delle tabelle DLM può migliorare le prestazioni. Con Red Hat Enterprise Linux 6.1, e versioni più recenti, le dimensioni predefinite delle tabelle sono state aumentate. È importante sapere che è possibile aumentare manualmente le dimensioni con il seguente comando:

```
echo 1024 > /sys/kernel/config/dlm/cluster/lkbtbl_size
echo 1024 > /sys/kernel/config/dlm/cluster/rsbtbl_size
echo 1024 > /sys/kernel/config/dlm/cluster/dirtbl_size
```

Questi comandi non sono persistenti e non saranno implementati dopo il riavvio, per questo motivo è necessario aggiungerli ad uno degli script d'avvio, ed eseguirli prima di montare qualsiasi file system GFS2. In caso contrario le modifiche verranno ignorate.

Per informazioni più dettagliate sul blocco del nodo GFS2 consultare [Sezione 2.9, «Blocco dei nodi GFS2»](#).

## 2.5.3. Opzioni di ottimizzazione VFS: Ricerca e Sperimentazione

Come tutti i file system di Linux GFS2 è posizionato sopra un livello chiamato virtual file system (VFS). È possibile regolare il VFS in modo da migliorare le prestazioni del GFS2 usando il comando `sysctl(8)`. Per esempio i valori per `dirty_background_ratio` e `vfs_cache_pressure` possono essere regolati in modo desiderato. Per avere i valori correnti usare i seguenti comandi:

```
sysctl -n vm.dirty_background_ratio
sysctl -n vm.vfs_cache_pressure
```

I seguenti comandi regolano i valori:

```
sysctl -w vm.dirty_background_ratio=20
sysctl -w vm.vfs_cache_pressure=500
```

È possibile modificare in modo permanente i valori di questi parametri modificando il file `/etc/sysctl.conf`.

Per ottenere i valori ottimali per i casi desiderati, controllate le varie opzioni VFS e sperimentatele in un cluster di prova prima di implementarle in ambienti di produzione.

## 2.5.4. SELinux: Evitare l'uso di SELinux su GFS2

Security Enhanced Linux (SELinux) è fortemente consigliato per motivi di sicurezza nella maggior parte delle situazioni, ma il suo utilizzo con GFS2 non è supportato. SELinux archivia le informazioni utilizzando attributi estesi per ogni oggetto del file system. La lettura, scrittura e il mantenimento di questi attributi estesi è possibile, ma al tempo stesso può rallentare considerevolmente GFS2. Disattivare SELinux sui file system GFS2.

## 2.5.5. Impostazione di NFS attraverso GFS2

A causa di una ulteriore complessità del sistema di bloccaggio del GFS2 e della sua natura clusterizzata, l'impostazione di NFS attraverso GFS2 richiede un certo numero di precauzioni ed una configurazione attenta. Questa sezione riporta le informazioni da considerare durante la configurazione del servizio NFS attraverso un file system GFS2.





## AVVERTIMENTO

Se il file system GFS2 è esportato con NFS e le applicazioni client NFS utilizzano i blocchi POSIX allora montare il file system con l'opzione `locallocks`. L'effetto desiderato è quello di forzare i blocchi POSIX di ogni server in modo da essere locali: es. non-clusterizzati, indipendenti tra loro. (Si possono verificare un certo numero di problemi se GFS2 cerca di implementare i blocchi POSIX da NFS sui nodi di un cluster.) Per applicazioni in esecuzione su client NFS, in presenza di blocchi POSIX localizzati due client possono avere contemporaneamente lo stesso blocco se i due client eseguono il montaggio da server differenti. Se tutti i client montano NFS da un server allora il problema di server separati che conferiscono indipendentemente gli stessi blocchi non sussiste. Se non siete sicuri di montare il file system con l'opzione `locallocks` allora non utilizzate la suddetta opzione; è più sicuro avere i blocchi in funzione seguendo una modalità clusterizzata.

In aggiunta alle considerazioni relative al blocco considerare anche quanto di seguito riportato durante la configurazione di un servizio NFS attraverso un file system GFS2.

- Red Hat supporta solo le configurazioni Red Hat High Availability Add-On che utilizzano NFSv3 con una configurazione del blocco attiva/passiva con le seguenti caratteristiche:
  - Il file system backend è un file system GFS2 in esecuzione su un cluster con 2 a 16 nodi.
  - Un server NFSv3 viene definito come un servizio che esporta l'intero file system GFS2 da un nodo per volta del cluster.
  - Il server NFS può passare (failover) da un nodo ad un altro del cluster (configurazione attiva/passiva).
  - Non è permesso alcun accesso al file system GFS2 ad eccezione del server NFS. Ciò include l'accesso al file system GFS2 e attraverso Samba o Samba Clusterizzato.
  - Non è presente alcun supporto del quota NFS sul sistema.

Questa configurazione rende disponibile una HA per il file system e riduce il tempo di inattività poiché il server NFS esegue il failover da un nodo ad un altro senza l'esecuzione del comando `fsck`.

- L'opzione NFS `fsid=` è obbligatoria per le esportazioni NFS di GFS2.
- In presenza di alcune problematiche con il cluster (per esempio, se il cluster non ha più un quorum ed il fencing fallisce), i volumi logici clusterizzati ed il file system GFS2 verranno interrotti e non sarà più possibile un loro accesso fino a quando il cluster avrà nuovamente il suo quorum. Considerare questa impostazione se il tipo di failover sopra descritto è quello più appropriato per il vostro sistema.

### 2.5.6. Samba (SMB o Windows) File Serving attraverso GFS2

Con la versione Red Hat Enterprise Linux 6.2 è disponibile il Samba (SMB o Windows) file serving da un file system GFS2 con CTDB, il quale permette di avere configurazioni active/active. Per informazioni sulla configurazione Samba clusterizzato consultare il documento *Amministrazione del cluster*.

Non è supportata l'operazione d'accesso simultaneo dei dati in una condivisione Samba da una posizione esterna. Attualmente non è disponibile alcun supporto per i lease del cluster GFS2, con un conseguente rallentamento del Samba file serving.

## 2.6. BACKUP DEL FILE SYSTEMA

È importante eseguire backup a intervalli regolari del file system GFS2 in casi di emergenza, senza considerare la dimensione del file system. Numerosi amministratori si sentono protetti poiché utilizzano il RAID, multipath, mirroring, istantanee e altre forme di ridondanza.

Può essere problematico creare un backup poiché questo processo eseguito nei confronti di un nodo, o gruppo di nodi, generalmente richiede la lettura dell'intero file system in sequenza. Se si esegue il processo da un nodo singolo, il nodo in questione manterrà tutte le informazioni in cache, fino a quando altri nodi nel cluster iniziano a richiedere i lock. L'esecuzione di questo tipo di programma durante le funzioni normali di un cluster, avrà un impatto negativo sulle prestazioni.

L'abbandono della cache dopo il completamento del backup riduce il tempo necessario ad altri nodi per ottenere una cache/lock del cluster. Questa situazione ancora non è quella ideale poiché gli altri nodi avranno sospeso la memorizzazione in cache dei dati, prima di poter iniziare il processo di backup. Per abbandonare la cache usare il seguente comando dopo il completamento del backup:

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

Il processo potrebbe essere più veloce se ogni nodo esegue il backup dei propri file, in questo modo il processo verrà suddiviso tra i nodi presenti nel cluster. Per fare questo usare uno script che utilizzi il comando `rsync` su directory specifiche ai nodi.

Il modo migliore per eseguire un backup del GFS2 è di creare una istantanea dell'hardware sul SAN, presentare l'istantanea ad un altro sistema ed eseguire il backup. Il sistema usato per il backup dovrebbe montare l'istantanea con `-o lockproto=lock_no1ock`, poiché quest'ultima non sarà all'interno del cluster.

## 2.7. CONSIDERAZIONI HARDWARE

Prima di implementare un file system GFS2 considerare quanto di seguito riportato.

- Usare migliori opzioni per l'archiviazione

GFS2 è in grado di operare su opzioni di storage condiviso più semplici come iSCSI o Fibre Channel over Ethernet (FCoE), ma per migliori prestazioni utilizzare opzioni di archiviazione migliori con una capacità di memorizzazione in cache maggiore. Red Hat esegue la maggior parte delle prove sulle prestazioni, qualità e di integrità su storage SAN con interconnessione Fibre Channel. Come regola generale è sempre meglio una implementazione precedentemente testata.

- Eseguire il test degli strumenti di rete prima del loro utilizzo

Strumenti più veloci e di migliore qualità velocizzano le comunicazioni e il GFS2 di un cluster, aumentandone l'affidabilità. Tuttavia non è necessario acquistare l'hardware più costoso. Alcuni degli interruttori di rete più costosi presentano alcuni problemi nell'inoltro dei pacchetti multicast, usati a loro volta per passare gli `fcntl lock (flock)`, mentre interruttori di rete meno costosi sono talvolta più veloci e più affidabili. È sempre meglio testare l'hardware prima di implementarlo in un ambiente di produzione.

## 2.8. PROBLEMATICHE DELLE PRESTAZIONI: CONTROLLARE IL PORTALE CLIENTI DI RED HAT

Per informazioni sulle implementazioni e aggiornamento dei cluster Red Hat Enterprise Linux utilizzando un High Availability Add-On e Red Hat Global File System 2 (GFS2), consultare "Red Hat Enterprise Linux Cluster, High Availability e GFS Deployment Best Practices" sul Portale clienti di Red Hat su <https://access.redhat.com/site/articles/40051>.

## 2.9. BLOCCO DEI NODI GFS2

Per ottenere le migliori prestazioni da un file system GFS2 è importante comprendere alcune delle teorie operative di base. Un file system con un solo nodo viene implementato insieme ad una cache, tale operazione serve per eliminare la latenza presente a causa dell'accesso al disco quando si utilizzano frequentemente determinati dati. Con Linux il page cache (ed il buffer cache) forniscono queste funzioni di caching.

Con GFS2 ogni nodo presenta la propria page cache la quale può contenere alcune porzioni di dati sul disco. GFS2 utilizza un meccanismo di bloccaggio chiamato *glocks* per mantenere l'integrità della cache tra i nodi. Il sistema secondario di glock fornisce una funzione di gestione della cache implementata tramite il *distributed lock manager* (DLM) come livello di comunicazione sottostante.

Glock fornisce una protezione per la cache per ogni inode, in questo modo è presente un blocco per inode usato per controllare il caching. Se si conferisce glock in modalità condivisa (DLM lock mode: PR) i dati presenti nel glock in questione potranno essere archiviati contemporaneamente su uno o più nodi, in questo modo tutti i nodi potranno avere un accesso locale ai dati.

Se si conferisce glock in modalità esclusiva (DLM lock mode: EX) allora solo un unico nodo potrà archiviare i dati in cache sotto il glock in questione. Questa modalità viene usata da tutte le operazioni che modificano i dati (come ad esempio una chiamata del sistema *write*).

Se un altro nodo richiede un glock, il quale a sua volta non potrà essere immediatamente conferito, DLM invia un messaggio al nodo o nodi che attualmente presentano un glock i quali bloccano la nuova richiesta, richiedendone il loro rilascio. Il rilascio di glock (in base agli standard di numerose operazioni del file system) è una operazione molto lunga. Il rilascio di un glock condiviso richiede la sola rimozione della convalida della cache, tale processo è relativamente veloce e proporzionale alla quantità di dati archiviati in cache.

Il rilascio di un glock esclusivo richiede un processo di azzeramento molto lungo e di riscrittura dei dati modificati sul disco, il tutto seguito dalla rimozione della convalida simile al glock condiviso.

La differenza presente tra un file system con un solo nodo e GFS2 è che il primo presenta una cache singola mentre il secondo (GFS2) presenta una cache separata su ogni nodo. In entrambi i casi la latenza per l'accesso ai dati archiviati in cache è simile, ma la latenza per l'accesso ai dati non archiviati in cache è maggiore in GFS2 se un altro nodo ha precedentemente archiviato gli stessi dati.



## NOTA

A causa del metodo usato per l'implementazione del caching di GFS2 le migliori prestazioni si ottengono quando:

- Un inode viene usato in sola lettura su tutti i nodi.
- Un inode viene scritto o modificato solo da un nodo singolo.

Da notare che l'inserimento e la rimozione delle voci da una directory durante la creazione e la cancellazione di un file conta come la scrittura sull'inode della directory.

È possibile non rispettare questa regola se la stessa viene generalmente osservata. Se la regola spesso non viene rispettata l'utente andrà incontro a problemi molto seri di prestazione.

Se eseguite il `mmap()` di un file su GFS2 con una mappatura lettura/scrittura, ma eseguite solo la lettura, tale operazione conta solo come lettura. Al contrario su GFS l'operazione conta come scrittura, in questi termini GFS2 è molto più scalabile con `mmap()` I/O.

Se non impostate il parametro `noatime mount`, allora i processi di lettura causeranno l'aggiornamento dei timestamp del file anche da parte del processo di scrittura. È consigliato a tutti gli utenti di GFS2 di eseguire il montaggio usando `noatime` a meno che non siano presenti requisiti specifici per `atime`.

### 2.9.1. Problematiche con il Posix Locking

Prima di utilizzare Posix locking considerate quanto di seguito riportato:

- L'uso di Flocks permette di avere una processazione più veloce rispetto ai lock di Posix.
- I programmi che utilizzano i lock di Posix in GFS2 non devono utilizzare la funzione `GETLK`, poichè in un ambiente clusterizzato l'ID del processo potrebbe essere per un nodo diverso nel cluster.

### 2.9.2. Regolazione delle prestazioni con GFS2

Generalmente è possibile alterare il metodo attraverso il quale un'applicazione problematica archivia i propri dati, così facendo sarà possibile aumentare considerevolmente le prestazioni.

Un esempio tipico di una applicazione problematica può essere un server di posta elettronica. Generalmente disposti con una directory di spool contenente i file per ogni utente (`mbox`), o con una directory per ogni utente contenente un file per ogni messaggio (`maildir`). Quando arrivano le richieste attraverso IMAP, l'organizzazione ideale è quella di conferire ad ogni utente una affinità con un nodo specifico. In questo modo le richieste per la visualizzazione e rimozione dei messaggi di posta elettronica verranno serviti dalla cache presente sul nodo interessato. Ovviamente se il nodo fallisce, la sessione potrà essere riavviata su un altro nodo.

Se la posta arriva tramite SMTP i nodi individuali possono essere impostati in modo da passare il messaggio di un particolare utente ad un nodo specifico per default. Se il nodo predefinito non è in funzione, il messaggio potrà essere salvato direttamente nello spool di posta dell'utente da parte del nodo ricevente. Questo tipo di organizzazione è intesa a mantenere set particolari di file archiviati in cache solo su di un nodo in casi normali, e permettere un accesso diretto nel caso di errore del nodo.

Questo tipo di impostazione permette l'uso migliore del page cache di GFS2 rendendo altresì gli errori trasparenti per l'applicazione, sia `imap` che `smtp`.

Il backup rappresenta spesso un'altra area spinosa. Se possibile è fortemente consigliato eseguire il backup del set di ogni nodo direttamente dal nodo che esegue il caching del set di inode. Se siete in possesso di uno script di backup il quale viene eseguito in determinati periodi e coincide con il punto massimo del tempo di risposta di una applicazione in esecuzione su GFS2, allora molto probabilmente il cluster non utilizzerà nel modo più efficace il page cache.

Ovviamente se siete in grado di arrestare l'applicazione per poter eseguire il backup allora non avrete alcun problema. Se il backup viene eseguito da un solo nodo dopo il completamento di una sezione molto grande del file system, esso verrà archiviato su quel nodo con un conseguente deterioramento delle prestazioni per accessi successivi eseguiti da altri nodi. Tale tendenza può essere parzialmente ridotta rilasciando il VFS page cache sul nodo di backup dopo il completamento del backup con il seguente comando:

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

Tuttavia questa non rappresenta la soluzione migliore come può essere per esempio la procedura attraverso la quale il set in funzione su ogni nodo sia condiviso, per la maggior parte in sola lettura sul cluster, o accesso principalmente da un singolo nodo.

### 2.9.3. Risoluzione problematiche relative alle prestazioni di GFS2 con il GFS2 Lock Dump

Se si verifica un deterioramento delle prestazioni del cluster a causa di un uso inefficiente del caching del GFS2, potrete notare un aumento dei tempi di attesa I/O. A tale scopo usate le informazioni presenti nel lock dump di GFS2 per determinare la causa del problema.

Questa sezione fornisce solo una breve descrizione del lock dump di GFS2. Per informazioni più dettagliate consultare [Appendice C, Tracepoint GFS2 e debugfs glock File](#)

Le informazioni presenti nel lock dump del GFS2 sono disponibili nel file `debugfs` attraverso il seguente nome del percorso, assumendo che `debugfs` sia stato montato su `/sys/kernel/debug/`:

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

Il contenuto è una serie di linee. Ogni linea che inizia con G: rappresenta un glock, e linee seguenti, rappresentano le informazioni relative al glock che le precede sul file.

Il modo migliore per usare il file `debugfs` è quello di utilizzare il comando `cat` per avere una copia del contenuto completo del file (potrebbe richiedere molto tempo se siete in possesso di una quantità di RAM molto grande ed un numero elevato di inode presenti in cache) se l'applicazione presenta qualche problema, consultando le informazioni presenti in un secondo momento.



#### NOTA

Potrebbe essere utile creare due copie del file `debugfs` una dopo qualche secondo o minuto dalla precedente. Confrontando le informazioni dell'holder nelle due tracce relative allo stesso numero di glock sarete in grado di verificare se il carico di lavoro prosegue senza alcun problema (lentamente) o se si è fermato (in questo caso siamo in presenza di un bug da riportare al supporto di Red Hat immediatamente).

Le linee nel file `debugfs` che iniziano con H: (holder 'detentori') rappresentano le richieste del blocco

conferite o in attesa di essere assegnate. Il campo dei flag sulla linea f: mostra quale: Il flag 'W' si riferisce alla richiesta in attesa, 'H' invece si riferisce alla richiesta assegnata. I glock che possiedono un numero elevato di richieste d'attesa nella maggior parte dei casi sono quelli con maggiore contesa.

[Tabella 2.1, «Flag di glock»](#) mostra il significato dei diversi flag di glock mentre [Tabella 2.2, «Flag holder di glock»](#) mostra il significato dei diversi flag degli holder di glock nell'ordine presente nei dump di glock.

**Tabella 2.1. Flag di glock**

Flag	Nome	Significato
b	Blocco	Valido se impostato un flag locked, indica che l'operazione richiesta dal DLM potrebbe eseguire un blocco. Questo flag viene annullato per operazioni di declassamento e per lock "try". Lo scopo di questo flag è quello di permettere la raccolta di statistiche sul tempo di risposta DLM, indipendentemente dal tempo necessario per altri nodi di eseguire operazioni di demote dei lock.
d	Pending demote	Una richiesta di retrocessione (remota) rinviata
D	Demote (retrocessione)	Una richiesta di retrocessione (locale o remota)
f	Log flush	È necessario eseguire il commit del log prima di rilasciare questo glock
F	Frozen	Le repliche dei nodi remoti verranno ignorate - il recupero è in corso. Questo flag non si riferisce al freeze del file system, il quale utilizza un meccanismo diverso, ma viene usato solo per operazioni di ripristino.
i	Invalidate in progress	La rimozione in questo glock della convalida delle pagine è in corso
l	Initial	Impostato quando il blocco DLM è associato con questo glock
l	Locked	Il glock è in procinto di cambiare stato
L	LRU	Impostato quando glock è sull'elenco LRU
o	Oggetto	Impostato quando glock è associato con un oggetto (cioè un inode per il tipo 2 di glock, e un gruppo di risorse per il tipo 3 di glock)
p	Demote in progress	Glock è in procinto a rispondere ad una richiesta di retrocessione

Flag	Nome	Significato
q	In coda	Impostato quando un holder è in coda per un glock ed è rimosso quando un glock è occupato, quando altri holder non sono disponibili. Usato come parte dell'algoritmo per il calcolo dell'intervallo minimo per un glock.
r	Reply pending	La risposta ricevuta da un nodo remoto è in attesa di processazione
y	Dirty	È necessario azzerare i dati sul disco prima di rilasciare questo glock

Tabella 2.2. Flag holder di glock

Flag	Nome	Significato
a	Async	Non attendere il risultato di glock (il risultato verrà richiesto più avanti)
A	Qualsiasi	Qualsiasi modalità di blocco compatibile è accettabile
c	No cache	Quando sbloccato, retrocedi immediatamente il blocco DLM
e	No expire	Ignora le richieste di cancellazione del blocco successive
E	exact	Deve avere una modalità di blocco esatta
F	First	Imposta quando l'holder è il primo ad essere conferito per questo blocco
H	Holder	Indica che il blocco richiesto è stato conferito
p	Priorità	Metti l'holder in cima alla coda
t	Prova	Un blocco "try"
T	Try 1CB	Un blocco "try" che invia una callback
M	Wait	Imposta mentre in attesa del completamento della richiesta

Dopo aver identificato il glock che causa il problema, la fase successiva è quella di trovare l'inode relativo. Il numero di glock (n: sulla riga G:) riporta questa informazione. Il suo formato è *type/number* e se *type* risulta essere 2, allora saremo in presenza di un glock inode ed il *number* è un numero inode. Per rintracciare l'inode eseguire `find -inum number` dove *number* è il numero inode modificato da un formato esadecimale nel file di glock in un formato decimale.



## NOTA

Se eseguite `find` su di un file system in presenza di una contesa del blocco, molto probabilmente non farete altro che peggiorare il problema esistente. È consigliato arrestare l'applicazione prima di eseguire `find` se siete alla ricerca di inode contesi.

Tabella 2.3, «Tipi di glock» mostra il significato dei diversi tipi di glock.

Tabella 2.3. Tipi di glock

Numero tipo	Tipo di blocco	Uso
1	Trans	Blocco transazione
2	Inode	Dati e metadati di Inode
3	Rgrp	Metadati del gruppo di risorse
4	Meta	Il superblocco
5	lopen	Rilevamento ultimo nodo che ha utilizzato inode
6	Flock	<code>flock(2)</code> syscall
8	Quota	Operazioni quota
9	Diario	Journal mutex

Se il glock identificato era di un tipo diverso, molto probabilmente sarà di tipo 3: (gruppo risorse). Se visualizzate un numero elevato di processi in attesa per altri tipi di glock in presenza di carichi normali, riportatelo al supporto di Red Hat.

Se visualizzate un certo numero di richieste in coda in un blocco del gruppo delle risorse controllate se è presente un numero elevato di nodi rispetto al numero di gruppi delle risorse presenti nel file system. Oppure controllate se il file system è quasi pieno (causando un tempo più lungo per le ricerche di blocchi liberi). In entrambi i casi sarà possibile aggiungere più storage ed usare il comando `gfs2_grow` per estendere il file system.



## CAPITOLO 3. PER INIZIARE

Questo capitolo descrive le procedure per una impostazione iniziale di GFS2 e contiene le seguenti sezioni:

- [Sezione 3.1, «Prerequisiti»](#)
- [Sezione 3.2, «Compiti iniziali per l'impostazione»](#)

### 3.1. PREREQUISITI

Completare le seguenti fasi prima di impostare un GFS2 di Red Hat:

- Siate a conoscenza delle caratteristiche più importanti dei nodi GFS2 (consultare la [Sezione 1.2, «Prima d'impostare il GFS2»](#)).
- Assicuratevi che gli orologi presenti sui nodi del GFS2 siano sincronizzati. È consigliato utilizzare il software Network Time Protocol (NTP) presente con la distribuzione Red Hat Enterprise Linux.



#### NOTA

Gli orologi del sistema presenti all'interno dei nodi del GFS2 devono essere impostati con qualche minuto di sfasamento tra loro, in modo da evitare un aggiornamento time-stamp dell'inode non necessario. Tali aggiornamenti non necessari possono influire negativamente sulle prestazioni del cluster

- Per poter utilizzare il GFS2 in un ambiente clusterizzato sarà necessario configurare il sistema per un utilizzo del Clustered Logical Volume Manager (CLVM), un insieme di estensioni per il LVM Logical Volume Manager. Per poter usare il CLVM è necessario eseguire il software Red Hat Cluster Suite, incluso il demone `clvmd`. Per informazioni su come utilizzare CLVM consultare la *Gestione del Logical Volume Manager*. Per informazioni su come installare e gestire il Red Hat Cluster Suite consultare l'*Amministrazione del Cluster*.

### 3.2. COMPITI INIZIALI PER L'IMPOSTAZIONE

L'impostazione GFS2 iniziale consiste nelle seguenti fasi:

1. Impostazione dei volumi logici.
2. Creazione di un file system GFS2.
3. Montaggio dei file system.

Seguite queste fasi per una impostazione iniziale di GFS2.

1. Utilizzando LVM, creare un volume logico per ogni file system GFS2 di Red Hat.



#### NOTA

È possibile utilizzare gli script `init.d` inclusi in Red Hat Cluster Suite per attivare e disattivare automaticamente i volumi logici. Per maggiori informazioni sugli script `init.d`, consultate la *Configurazione e Gestione di un Red Hat Cluster*

2. Creare i file system GFS2 sui volumi logici creati durante la Fase 1. Scegliere un nome unico per ogni file system. Per maggiori informazioni sulla creazione di un file system GFS2, consultare la [Sezione 4.1, «Creazione di un file system»](#).

È possibile utilizzare uno dei seguenti formati per la creazione di un file system GFS2 clusterizzato:

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

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

Per maggiori informazioni su come creare un file system GFS2 consultare la [Sezione 4.1, «Creazione di un file system»](#).

3. Ad ogni nodo montare i file system GFS2. Per maggiori informazioni sul montaggio di un file system GFS2 consultare la [Sezione 4.2, «Montaggio di un file system»](#).

Utilizzo del comando:

```
mount BlockDevice MountPoint
```

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

L'opzione `-o acl` mount permette la manipolazione dei file ACL. Se un file system è stato montato senza l'opzione di mount `-o acl`, gli utenti saranno in grado di visualizzare gli ACL (con `getfacl`), ma non saranno in grado di impostarli (con `setfacl`).



#### NOTA

Utilizzare gli script `init.d` inclusi con il Red Hat High Availability Add-On per automatizzare i processi di montaggio e smontaggio dei file system GFS2.

## CAPITOLO 4. GESTIONE DEL GFS2

Questo capitolo descrive i compiti ed i comandi per la gestione di GFS2 e comprende le seguenti sezioni:

- [Sezione 4.1, «Creazione di un file system»](#)
- [Sezione 4.2, «Montaggio di un file system»](#)
- [Sezione 4.3, «Come smontare un file system»](#)
- [Sezione 4.5, «Gestione quota del GFS2»](#)
- [Sezione 4.6, «Come espandere un file system»](#)
- [Sezione 4.7, «Come aggiungere i journal ad un file system»](#)
- [Sezione 4.8, «Data Journaling»](#)
- [Sezione 4.9, «Configurazione degli aggiornamenti `atime`»](#)
- [Sezione 4.10, «Sospensione di una attività su di un file system»](#)
- [Sezione 4.11, «Come ripristinare un file system»](#)
- [Sezione 4.12, «Mount Bind e Context-Dependent Path Names»](#)
- [Sezione 4.13, «Mount bind e ordine di montaggio di un file system»](#)
- [Sezione 4.14, «Funzione `Withdraw` di GFS2»](#)

### 4.1. CREAZIONE DI UN FILE SYSTEM

Creazione di un file system GFS2 con il comando `mkfs.gfs2`. È possibile utilizzare `mkfs` specificando l'opzione `-t gfs2`. Così facendo verrà creato un file system sul volume LVM attivato. Le seguenti informazioni sono necessarie per eseguire il comando `mkfs.gfs2`:

- Nome modulo/protocollo di blocco (il protocollo di blocco per un cluster è `lock_dlm`)
- Nome del cluster (quando in esecuzione come parte di una configurazione del cluster)
- Numero di journal (un journal necessario per ogni nodo che monterà il file system)

Durante la creazione di un file system GFS2 è possibile utilizzare direttamente `mkfs.gfs2`, oppure il comando `mkfs` con il parametro `-t` specificando un filesystem di tipo `gfs2` seguito dalle opzioni del file system `gfs2`.



#### NOTA

Una volta creato il file system GFS2 con il comando `mkfs.gfs2` non sarà possibile diminuire la dimensione del file system. Sarà possibile tuttavia aumentare la dimensione di un file system esistente con il comando `gfs2_grow` come descritto in [Sezione 4.6, «Come espandere un file system»](#).

#### Utilizzo

Durante la creazione di un filesystem GFS2 clusterizzato è possibile utilizzare uno dei seguenti formati:

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

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

Durante la creazione di un file system GFS2 locale sarà possibile usare uno dei seguenti formati:



### NOTA

Per la release di Red Hat Enterprise Linux 6, Red Hat non supporta l'uso del GFS2 come file system a nodo singolo.

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

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



### AVVERTIMENTO

Assicuratevi di essere a conoscenza su come utilizzare i parametri *LockProtoName* e *LockTableName*. Un utilizzo non appropriato dei parametri *LockProtoName* e *LockTableName* potrebbe causare la corruzione del file system o di lock space.

#### *LockProtoName*

Specifica il nome del protocollo di blocco da usare. Il suddetto protocollo per un cluster è `lock_dlm`.

#### *LockTableName*

Questo parametro viene specificato per il file system GFS2 in una configurazione del cluster. Esso presenta due sezioni separate da due punti (senza spazio) nel seguente modo:

*ClusterName* : *FSName*

- *ClusterName*, il nome del cluster per il quale è stato creato il file system GFS2.
- *FSName*, il nome del file system, può contenere da 1 a 16 caratteri ed il nome deve essere unico per tutti i file system `lock_dlm` presenti nel cluster, e per tutti i filesystem (`lock_dlm` e `lock_nolock`) su ogni nodo locale.

#### *Number*

Specifica il numero di journal da creare con il comando `mkfs.gfs2`. È necessaria l'implementazione di un journal per ogni nodo che monta il file system. Per i file system GFS2, è possibile aggiungere un numero maggiore di journal senza espandere il file system, come descritto nella [Sezione 4.7, «Come aggiungere i journal ad un file system»](#).

## BlockDevice

Specifica un volume logico o fisico.

## Esempi

In questo esempio, `lock_dlm` è il protocollo di blocco usato dal file system poichè esso è un file system clusterizzato. Il nome del cluster è `alpha`, ed il nome del file system è `mydata1`. Il file system contiene otto journal e viene creato su `/dev/vg01/lvol0`.

```
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 questo esempio viene creato un secondo file system `lock_dlm`, il quale può essere usato nel cluster `alpha`. Il nome del file system è `mydata2`. Il file system contiene otto journal e viene creato su `/dev/vg01/lvol1`.

```
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
```

## Opzioni complete

Tabella 4.1, «Opzioni del comando: `mkfs.gfs2`» descrive le opzioni del comando `mkfs.gfs2` (flag e parametri).

Tabella 4.1. Opzioni del comando: `mkfs.gfs2`

Flag	Parametro	Descrizione
<code>-c</code>	<i>Megabytes</i>	Imposta la dimensione iniziale di ogni quota change file del journal in <i>Megabytes</i> .
<code>-D</code>		Abilita l'output di debugging.
<code>-h</code>		Aiuto. Mostra le opzioni disponibili.
<code>-J</code>	<i>MegaBytes</i>	Specifica la dimensione del journal in megabytes. La dimensione predefinita del journal è 128 megabytes. La dimensione minima è 8 megabytes. Journal più grandi migliorano le prestazioni anche se utilizzano una memoria maggiore rispetto ai journal più piccoli.
<code>-j</code>	<i>Number</i>	Specifica il numero di journal da creare con il comando <code>mkfs.gfs2</code> . Sarà necessario un journal per ogni nodo che monta il file system. Se questa opzione non viene specificata, verrà creato un solo journal. Per i file system GFS2, è possibile aggiungere un numero maggiore di journal senza espandere il file system.

Flag	Parametro	Descrizione
-O		Impedisce al comando <code>mkfs.gfs2</code> di chiedere la conferma prima della scrittura sul file system.
-p	<i>LockProtoName</i>	<p>Specifica il nome del protocollo di blocco da usare. I protocolli di blocco riconosciuti includono:</p> <p><b>lock_dlm</b> – Il modulo di blocco standard necessario per un file system clusterizzato.</p> <p><b>lock_nolock</b> – Usato quando GFS2 si comporta come un file system locale (un solo nodo).</p>
-q		Quiet. Non mostrare niente.
-r	<i>MegaBytes</i>	Specifica la dimensione delle risorse dei gruppi in megabytes. La dimensione minima della risorsa del gruppo è 32 MB. La dimensione massima è di 2048 MB. Una dimensione della risorsa del gruppo molto grande potrebbe aumentare le prestazioni su file system molto grandi. Se non specificato, <code>mkfs.gfs2</code> sceglie la dimensione in base alla dimensione del file system: una dimensione media dei file system contiene circa 256MB di risorsa dei gruppi, file system più grandi avranno delle RG più grandi per migliori prestazioni.
-t	<i>LockTableName</i>	<p>Un identificatore unico il quale specifica il campo della tabella di blocco durante l'utilizzo del protocollo <b>lock_dlm</b>; il protocollo <b>lock_nolock</b> non utilizza questo parametro.</p> <p>Questo parametro presenta due campi separati da due punti (senza alcuno spazio): <b>ClusterName : FSName</b>.</p> <p><b>ClusterName</b> è il nome del cluster per il quale è stato creato il file system GFS2; solo gli appartenenti a questo cluster possono utilizzare questo file system. Il nome del cluster viene impostato nel file <code>/etc/cluster/cluster.conf</code> tramite <b>Cluster Configuration Tool</b>, e mostrato sul <b>Cluster Status Tool</b> nella GUI di gestione del cluster di Red Hat Cluster Suite.</p> <p><b>FSName</b>, il nome del file system, può contenere da 1 a 16 caratteri, ed il nome deve essere unico tra tutti i file system presenti nel cluster.</p>

Flag	Parametro	Descrizione
-u	<i>MegaBytes</i>	Specifica la dimensione iniziale del file dell'etichetta non collegato di ogni journal.
-V		Mostra le informazioni sulla versione del comando

## 4.2. MONTAGGIO DI UN FILE SYSTEM

Prima di poter montare un file system GFS2, il file system deve essere presente (consultate la [Sezione 4.1, «Creazione di un file system»](#)), il volume interessato deve essere attivato, ed i sistemi di blocco e di clustering devono essere avviati (consultare la *Configurazione e Gestione di un Red Hat Cluster*). Una volta soddisfatti i suddetti requisiti sarà possibile montare il file system GFS2 in modo simile ad ogni file system di Linux.



### NOTA

È possibile visualizzare il seguente errore se si cercherà di montare un file system GFS2 quando il Cluster Manager (*cman*) non è stato ancora avviato:

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

Per manipolare i file ACL è necessario montare il file system con l'opzione di montaggio `-o acl`. Se un file system è stato montato senza l'opzione `-o acl`, gli utenti saranno in grado di visualizzare le ACL (con `getfacl`), senza però poterle impostare (con `setfacl`).

### Utilizzo

#### Montaggio senza manipolazione ACL

```
mount BlockDevice MountPoint
```

#### Montaggio con manipolazione ACL

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

#### -o acl

Opzione specifica al GFS2 che permettere la manipolazione dei file ACL.

#### *BlockDevice*

Specifica il dispositivo a blocchi dove risiede il file system GFS2.

#### *MountPoint*

Specifica la directory dove il file system GFS2 deve essere montato.

## Esempio

In questo esempio il file system GFS2 presente su `/dev/vg01/lvo10` viene montato sulla directory `/mygfs2`.

```
mount /dev/vg01/lvo10 /mygfs2
```

## Utilizzo completo

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

L'argomento `-o option` consiste in opzioni specifiche a GFS2 (consultare la [Tabella 4.2, «Opzioni di mount specifiche al GFS2»](#)) o da opzioni `mount -o` basate sugli standard di Linux, o tramite una combinazioni di entrambi. Le opzioni *option* multiple vengono separate da virgole e non da spazi.



### NOTA

Il comando `mount` è un comando Linux. In aggiunta all'utilizzo delle opzioni specifiche al GFS2 descritte in questa sezione, è possibile utilizzare altre opzioni standard del comando `mount` (per esempio, `-r`). Per informazioni su altre opzioni del comando `mount` di Linux, consultare la pagina man di `mount`.

[Tabella 4.2, «Opzioni di mount specifiche al GFS2»](#) descrive i valori `-o option` che possono essere passati al GFS2 al momento del montaggio.



### NOTA

Questa tabella riporta le opzioni usate solo con i file system locali. Da notare che per la release di Red Hat Enterprise Linux 6, Red Hat non supporta l'uso del GFS2 come file system a nodo singolo. Red Hat continuerà il supporto dei file system GFS2 a nodo singolo per il montaggio delle snapshot dei file system del cluster (per esempio a scopi di backup).

**Tabella 4.2. Opzioni di mount specifiche al GFS2**

Opzione	Descrizione
<code>acl</code>	Permette la manipolazione dei file ACL. Se un file system è stato montato senza l'opzione di montaggio <code>acl</code> , gli utenti saranno in grado di visualizzare le ACL (con <code>getfacl</code> ), senza però poterle impostare (con <code>setfacl</code> ).



Opzione	Descrizione
<b>data=[ordered writeback]</b>	<p>Una volta impostato <b>data=ordered</b>, i dati dell'utente modificati da una transazione verranno scaricati sul disco prima di confermare la transazione sul disco stesso. Tale operazione dovrebbe impedire una visualizzazione da parte dell'utente dei blocchi non inizializzati all'interno di un file dopo il verificarsi di un crash. Se invece è stato impostato <b>data=writeback</b>, i dati verranno scritti in qualsiasi momento sul disco, dopo che lo stesso è stato sporcato. Tale operazione non garantisce una consistenza simile a quella garantita dalla modalità <b>ordered</b>, ma dovrebbe essere più veloce sotto alcuni carichi di lavoro. La modalità predefinita è <b>ordered</b>.</p>
<div style="border: 1px solid black; padding: 5px;"> <b>ignore_local_fs</b> </div> <div style="border: 1px solid black; padding: 5px; margin-top: 5px;"> <b>Attenzione:</b> Questa opzione <i>non</i> dovrebbe essere usata quando i file system GFS2 risultano essere condivisi.         </div>	<p>Forza GFS2 a trattare il file system come se fosse un file system multihost. Per default utilizzando <b>lock_nolock</b> si abilita automaticamente il flag <b>localcaching</b>.</p>
<div style="border: 1px solid black; padding: 5px;"> <b>localflocks</b> </div> <div style="border: 1px solid black; padding: 5px; margin-top: 5px;"> <b>Attenzione:</b> Questa opzione non dovrebbe essere utilizzata quando i file system GFS2 sono condivisi.         </div>	<p>Indica a GFS2 di far eseguire flock e fcntl al livello VFS (virtual file system). Il flag <b>localflocks</b> viene automaticamente abilitato da <b>lock_nolock</b>.</p>
<b>lockproto=LockModuleName</b>	<p>Permette all'utente di specificare il protocollo di blocco da usare con il file system. Se <b>LockModuleName</b> non risulta essere specificato, il nome del protocollo di blocco viene letto dal superblocco del file system.</p>
<b>locktable=LockTableName</b>	<p>Permette all'utente di specificare la tabella di blocco da usare con il file system.</p>
<b>quota=[off/account/on]</b>	<p>Abilita o disabilita i quota per un file system. L'impostazione dei quota in modo da essere in uno stato <b>account</b>, causa una gestione corretta delle statistiche di utilizzo per UID/GID da parte dei file system; i valori limiti e di avvertimento vengono ignorati. Il valore predefinito è <b>off</b>.</p>

Opzione	Descrizione
<b>errors=panic withdraw</b>	Se specificate <b>errors=panic</b> gli errori relativi al file system causeranno un kernel panic. Il comportamento predefinito, simile a <b>errors=withdraw</b> , è quello per il sistema di ritirarsi dal file system e renderlo inaccessibile fino al processo di riavvio successivo; In alcuni casi il sistema può restare in esecuzione. Per informazioni sulla funzione di 'withdraw' di GFS2 consultare <a href="#">Sezione 4.14, «Funzione Withdraw di GFS2»</a> .
<b>discard/nodiscard</b>	Causa la generazione da parte di GFS2 delle richieste I/O "discard" per i blocchi che sono stati liberati. Utilizzabili da hardware idonei per implementare il thin provisioning e schemi simili.
<b>barrier/nobarrier</b>	Causa l'invio da parte di GFS2 delle I/O barrier durante l'azzeramento del journal. Il valore predefinito è <b>on</b> . Questa opzione viene <b>disattivata</b> automaticamente se il dispositivo sottostante non supporta le I/O barrier. L'uso delle I/O barrier con GFS2 è altamente consigliato in ogni momento a meno che il dispositivo a blocchi è stato creato in modo tale da non poter perdere il contenuto del write cache (per esempio, se si trova su di un UPS e se sprovvisto di un write cache).
<b>quota_quantum=secs</b>	Imposta il numero di secondi entro i quali una modifica delle informazioni relative ai quota è in grado di trascorrere in un nodo prima di essere scritta sul quota file. Esso rappresenta il metodo preferito per impostare questo parametro. Il valore è un numero intero di secondi maggiore di zero, per impostazione predefinita 60 secondi. Impostazioni più corte risultano in aggiornamenti più rapidi delle informazioni relative al lazy quota ed una possibilità più bassa di superare il quota. Impostazioni più lunghe rendono le operazioni dei file system più veloci ed efficienti.
<b>statfs_quantum=secs</b>	L'impostazione di <b>statfs_quantum</b> su 0 rappresenta il metodo preferito per impostare la versione lenta di <b>statfs</b> . Il valore predefinito è 30 secondi il quale imposta il periodo di tempo massimo prima che le modifiche <b>statfs</b> vengono sincronizzate sul file <b>statfs</b> master. Tale impostazione può essere modificata in modo da permettere valori <b>statfs</b> più veloci ma meno accurati o valori più lenti e più accurati. Se impostata su 0 <b>statfs</b> riporterà sempre i valori veri.
<b>statfs_percent=value</b>	Fornisce localmente un balzo sulla modifica in percentuale massima delle informazioni di <b>statfs</b> prima di sincronizzarle sul file <b>statfs</b> master anche se il periodo di tempo non è scaduto. Se impostato su <b>statfs_quantum</b> allora questa impostazione viene ignorata.

### 4.3. COME SMONTARE UN FILE SYSTEM

Il file system GFS2 può essere smontato allo stesso modo di un qualsiasi file system di Linux – utilizzando il comando `umount`.



#### NOTA

Il comando `umount` è un comando del sistema Linux. Le informazioni su questo comando sono disponibili nelle pagine man del comando `umount` di Linux.

#### Utilizzo

```
umount MountPoint
```

#### *MountPoint*

Specifica la directory sulla quale è attualmente montato il file system GFS2.

### 4.4. CONSIDERAZIONI PARTICOLARI DURANTE IL MONTAGGIO DEI FILE SYSTEM GFS2

I file system GFS2 montati manualmente e non automaticamente tramite una voce nel file `fstab` non saranno riconosciuti dal sistema quando i file system verranno smontati al momento dell'arresto del sistema. Ne risulterà che lo script del GFS2 non smonterà il file system GFS2. Dopo aver eseguito lo script d'arresto di GFS2, il processo di arresto standard eliminerà tutti i processi dell'utente restanti, incluso l'infrastruttura del cluster provando così a smontare il file system. Questo processo fallirà senza alcuna sospensione del sistema e dell'infrastruttura del cluster.

Per impedire la sospensione del sistema durante lo smontamento dei file system GFS2 eseguire una delle seguenti azioni:

- Usare sempre una voce nel file `fstab` per montare il file system GFS2.
- Se un file system GFS2 è stato montato manualmente con il comando `mount`, assicuratevi di smontare il file system manualmente con il comando `umount` prima del reboot o dell'arresto del sistema.

Se si verifica in queste circostanze una sospensione del file system durante la sua rimozione nel processo d'arresto del sistema, eseguire un reboot dell'hardware. Tale processo non dovrebbe causare alcuna perdita di dati poichè il file system è stato precedentemente sincronizzato durante l'arresto.

### 4.5. GESTIONE QUOTA DEL GFS2

I quota del File-system sono utilizzati per limitare la quantità di spazio utilizzato da un utente o gruppo. Un utente o gruppo non possiedono alcun limite quota fino a quando non ne viene impostato uno. Durante il montaggio di un file system GFS2 con l'opzione `quota=on` o `quota=account` GFS2 controlla lo spazio utilizzato da ogni utente o gruppo, anche quando non è implementato alcun limite. Il GFS2 aggiorna le informazioni dei quota in modo tale da non aver bisogno di una ricostruzione dell'utilizzo del quota dopo il crash del sistema.

Per impedire un rallentamento delle prestazioni un nodo GFS2 sincronizza periodicamente gli aggiornamenti con il quota file. L'accounting quota "fuzzy" permette agli utenti o gruppi di eccedere leggermente i limiti impostati. Per minimizzare questa tendenza, GFS2 riduce dinamicamente il periodo

di sincronizzazione quando si è prossimi ad un limite quota "hard".



## NOTA

Con Red Hat Enterprise Linux 6.1 è ora disponibile il supporto delle funzioni standard dei quota di Linux. Per il suo utilizzo installare l'RPM `quota`. Questo è il metodo preferito per gestire i quota sul GFS2 ed è consigliato il suo uso per tutte le nuove implementazioni di GFS2 che utilizzano il quota. Questa sezione documenta la gestione del quota sul GFS2 utilizzando le suddette funzioni.

Nelle release precedenti di Red Hat Enterprise Linux GFS2 richiedeva l'uso del comando `gfs2_quota` per la gestione dei quota. Per informazioni su questo comando consultare [Appendice A, Gestione quota del GFS2 con il comando `gfs2\_quota`](#).

### 4.5.1. Configurazione dei disk quota

Per implementare un disk quota seguire le seguenti fasi:

1. Impostare i quota in modalità enforcement o accounting.
2. Inizializzare il file del database quota con le informazioni correnti relative all'uso del blocco.
3. Assegnare le politiche del quota. (In modalità accounting queste politiche non sono implementate.)

Ogni fase viene riportata in dettaglio nelle sezioni seguenti.

#### 4.5.1.1. Impostazione dei quota in modalità Enforcement o Accounting.

Nei file system GFS2, il quota enforcement è disabilitato per default. Per abilitare il quota enforcement per un file system montare il file system specificando l'opzione `quota=on`.

È possibile controllare l'utilizzo del disco ed il conteggio dei quota per ogni utente e gruppo senza implementare valori limiti e di avvertimento. Per fare questo montare il file system specificando l'opzione `quota=account`.

#### Utilizzo

Per montare un file system con un quota abilitato, montare il file system specificando l'opzione `quota=on`.

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

Per montare un file system con un conteggio dei quota anche se i limiti del quota non sono stati impostati, montare il file sistem con l'opzione `quota=account`.

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

Per montare un file system con un quota disabilitato, montare il file system specificando l'opzione `quota=off`. Questa risulta essere l'impostazioni predefinita.

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

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

**on** - Specifica che il quota è stato abilitato quando il file system è montato.

**off** - Specifica che il quota è stato disabilitato quando il file system è montato.

**account** - Specifica la gestione delle statistiche dell'utente e del gruppo da parte del file system, anche se i limiti di quota non sono implementati.

### ***BlockDevice***

Specifica il dispositivo a blocchi dove risiede il file system GFS2.

### ***MountPoint***

Specifica la directory dove il file system GFS2 deve essere montato.

## **Esempi**

In questo esempio il file system GFS2 su `/dev/vg01/lvol0` è montato sulla directory `/mygfs2` con quota accounting abilitato.

```
mount -o quota=on /dev/vg01/lvol0 /mygfs2
```

In questo esempio il file system GFS2 su `/dev/vg01/lvol0` è montato sulla directory `/mygfs2` con un quota accounting gestito ma non implementato.

```
mount -o quota=account /dev/vg01/lvol0 /mygfs2
```

### **4.5.1.2. Creazione dei file del Quota Database**

Dopo aver montato ogni file system abilitato al quota il sistema sarà in grado di operare con i disk quota. Tuttavia il file system stesso non sarà in grado di supportare i quota. La fase successiva è quella di eseguire il comando `quotacheck`.

Il comando `quotacheck` esamina i file system abilitati al quota e compila una tabella per l'uso corrente del disco per file system. La tabella viene usata per aggiornare la copia relativa all'uso del disco del file system. In aggiunta i file del disk quota del file system vengono aggiornati.

Per creare i quota file sul file system usare le opzioni `-u` e `-g` del comando `quotacheck`; per poter inizializzare entrambe le opzioni le stesse devono essere specificate. Per esempio, se i quota sono stati abilitati per il file system `/home`, creare i file nella directory `/home`:

```
quotacheck -ug /home
```

### **4.5.1.3. Assegnazione dei quota per utente**

L'ultima fase è quella di assegnare i disk quota con il comando `edquota`. Se avete montato il file system in modalità accounting (con l'opzione `quota=account`), i quota non verranno implementati.

Per configurare il quota per un utente, come utente root in un prompt della shell, eseguire il comando:

```
edquota username
```

Eeguire la seguente fase per ogni utente che ha bisogno di un quota. Per esempio, se un quota è abilitato in `/etc/fstab` per la partizione `/home (/dev/VolGroup00/LogVol102` nell'esempio di seguito riportato) e si esegue il comando `edquota testuser`, verrà mostrato il seguente nell'editor configurato come il default per il sistema:

```
Disk quotas for user testuser (uid 501):
Filesystem          blocks      soft      hard      inodes      soft
hard
/dev/VolGroup00/LogVol102  440436      0         0
```



## NOTA

L'editor di testo definito dalla variabile d'ambiente **EDITOR** sarà usato da `edquota`. Per cambiare l'editor impostare la variabile **EDITOR** nel file `~/.bash_profile` per il percorso completo dell'editor desiderato.

La prima colonna rappresenta il nome del file system con un quota abilitato. La seconda colonna mostra il numero di blocchi attualmente usato dall'utente. Le due colonne successive sono usate per impostare i limiti hard e soft del blocco per l'utente sul file system.

Il limite soft del blocco definisce la quantità massima di spazio del disco utilizzabile.

Il limite hard del blocco è la quantità massima assoluta di spazio del disco utilizzabile da un utente o gruppo, Una volta raggiunto questo limite non sarà più possibile usare alcuno spazio.

Il file system GFS2 non mantiene alcun quota per gli inode, per questo motivo le suddette colonne non sono applicate ai file system GFS2 e saranno vuote.

Se qualsiasi valore viene impostato su 0, allora quel limite non viene impostato. Nell'editor di testo modificare i limiti desiderati. Per esempio;

```
Disk quotas for user testuser (uid 501):
Filesystem          blocks      soft      hard      inodes      soft
hard
/dev/VolGroup00/LogVol102  440436    500000    550000
```

Per verificare l'impostazione del quota per un utente usare il comando;

```
quota testuser
```

### 4.5.1.4. Assegnazione dei quota per un gruppo

Un quota può essere assegnato in base al gruppo. Se avete montato il file system in modalità accounting (con l'opzione `account=on`) i quota non verranno implementati.

Per impostare un quota al gruppo `devel` (è necessario che il gruppo esista per poter impostare un quota), usare il seguente comando:

```
edquota -g devel
```

Questo comando mostra il quota esistente per il gruppo nell'editor di testo;

```
Disk quotas for group devel (gid 505):
Filesystem                blocks      soft      hard    inodes    soft      hard
/dev/VolGroup00/LogVol02 440400      0         0
```

Il file system GFS2 non mantiene alcuna quota per gli inode, per questo motivo le suddette colonne non sono applicate ai file system GFS2 e saranno vuote. Modificare i limiti e successivamente salvare il file.

Per verificare se il quota del gruppo è stato impostato usare il seguente comando:

```
quota -g devel
```

#### 4.5.2. Gestione del Disk Quota

Se avete implementato una quota allora sarà necessario una sua gestione – nella maggior parte dei casi sarà necessario controllare se il quota eccede i propri limiti e allo stesso tempo la sua accuratezza.

Naturalmente se gli utenti eccedono i rispettivi quota o raggiungono spesso i limiti soft sarà necessario per un amministratore di sistema fare delle scelte in base al tipo di utenti ed alla quantità di spazio usato. In tal caso l'amministratore potrà assistere l'utente ad usare una quantità di spazio minore o aumentare il disk quota dell'utente stesso.

Sarà possibile creare un rapporto sull'utilizzo del disco eseguendo **repquota**. Per esempio il comando **repquota /home** emette il seguente output:

```
*** Report for user quotas on device /dev/mapper/VolGroup00-LogVol02
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
```

Per visualizzare il rapporto sull'uso del disco per tutti i file system abilitati al quota (opzione **-a**), usare il seguente comando:

```
repquota -a
```

Anche se la lettura del rapporto è semplice è necessario affrontare alcuni dei suoi contenuti. Il **--** visualizzato dopo ogni utente è un modo semplice per determinare se sono stati superati i limiti del blocco. Se i limiti soft sono stati superati allora sarà possibile visualizzare un segno **+** al posto del primo **-**. Il secondo **-** indica il limite dell'inode, ma i file system GFS2 non supportano i suddetti limiti quindi quel carattere resterà immutato (**-**). I file system GFS2 non supportano alcun grace period quindi la colonna **grace** resterà vuota.

Da notare che il comando **repquota** non è supportato attraverso NFS, indipendentemente dal file system sottostante.

#### 4.5.3. Mantenimento di un quota accurato

Se abilitate i quota sul file system dopo un periodo di tempo nel quale siete stati in esecuzione con quota disabilitati, allora eseguire il comando **quotacheck** per creare, controllare e riparare i quota file. Eseguire altresì **quotacheck** se credete che i quota file non siano accurati, ciò si può verificare quando

un file system viene smontato incorrettamente dopo un arresto inaspettato del sistema.

Per maggiori informazioni sul comando `quotacheck` consultare la pagina man di `quotacheck`.



#### NOTA

Eseguire `quotacheck` quando il file system è in sospensione (idle) su tutti i nodi poiché l'attività del disco potrebbe sfasare i valori elaborati dei quota.

#### 4.5.4. Sincronizzazione dei Quota con il comando `quotasync`

GFS2 archivia tutte le informazioni relative al quota nel proprio file interno presente sul disco. Un nodo GFS2 non aggiorna il quota file durante ogni processo di scrittura del file system; esso al contrario, aggiorna il quota file ogni 60 secondi. Tale processo è necessario per evitare contrasti tra i nodi durante la scrittura sul quota file, con un relativo rallentamento delle prestazioni.

Prima del raggiungimento dei limiti quota da parte di un utente o gruppo, GFS2 riduce dinamicamente l'intervallo tra un aggiornamento del file quota e l'altro, in modo da evitare il superamento del limite. L'intervallo di tempo medio tra i vari processi di sincronizzazione è regolabile, `quota_quantum`. È possibile modificare il valore predefinito di 60 secondi usando l'opzione di montaggio `quota_quantum=` come descritto in [Tabella 4.2, «Opzioni di mount specifiche al GFS2»](#). Altresì, il parametro `quota_quantum` deve essere impostato su ogni nodo e ogni qualvolta viene montato il file system. Le modifiche del parametro `quota_quantum` non verranno mantenute dopo i processi `umount`. È possibile aggiornare il valore `quota_quantum` con `mount -o remount`.

È possibile utilizzare il comando `quotasync` per sincronizzare le informazioni relative al quota da un nodo al quota file sul disco tra gli aggiornamenti automatici eseguiti dal GFS2.

#### Utilizzo

##### Sincronizzazione informazioni quota

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

**u**

Sincronizzazione dei quota file dell'utente.

**g**

Sincronizzazione dei quota file del gruppo

**a**

Sincronizzare tutti i file system attualmente abilitati al quota e supporto della sincronizzazione. Senza `-a` sarà necessario specificare il mount point del file system.

***mntpnt***

Specifica il file system GFS2 al quale vengono applicate le azioni.

##### Regolazione del periodo tra le sincronizzazioni

```
mount -o quota_quantum=secs,remount BlockDevice MountPoint
```



### *MountPoint*

Specifica il file system GFS2 al quale vengono applicate le azioni.

### *secs*

Specifica il nuovo periodo tra le normali sincronizzazioni del file quota del GFS2. I valori più piccoli possono rallentare le prestazioni.

## Esempi

In questo esempio vengono sincronizzati i quota 'sporchi' presenti in cache del nodo sul quale sono eseguiti, sul quota file su disco per il file system `/mnt/mygfs2`.

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

In questo esempio viene modificato il periodo predefinito tra aggiornamenti regolari del quota file in una ora (3600 secondi) per il file system `/mnt/mygfs2`, quando si esegue il rimontaggio del file system sul volume logico `/dev/volgroup/logical_volume`.

```
# mount -o quota_quantum=3600,remount /dev/volgroup/logical_volume
/mnt/mygfs2
```

### 4.5.5. Riferimenti

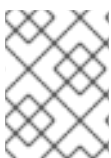
Per maggiori informazioni sui quota disk consultare le pagine *man* dei seguenti comandi:

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

## 4.6. COME ESPANDERE UN FILE SYSTEM

Il comando `gfs2_grow` viene usato per espandere un file system GFS2 dopo aver ingrandito il dispositivo sul quale risiede il file system. Con l'esecuzione di un comando `gfs2_grow` su di un file system GFS2 esistente, verrà utilizzato tutto lo spazio rimasto tra la parte finale del file system, e la parte finale del dispositivo con una nuova estensione del file system GFS2 inizializzata. Una volta completata tale operazione, verrà aggiornato l'indice delle risorse per il file system. Tutti i nodi presenti nel cluster potranno usare lo spazio di storage appena aggiunto.

Il comando `gfs2_grow` deve essere eseguito su di un file system montato, e su di un unico nodo del cluster. Tutti i nodi restanti noteranno l'avvenuta espansione ed automaticamente inizieranno ad usare il nuovo spazio.



### NOTA

Una volta creato un file system GFS2 con il comando `mkfs.gfs2` non sarà possibile diminuire la dimensione del file system.

## Utilizzo

```
gfs2_grow MountPoint
```

### *MountPoint*

Specifica il file system GFS2 al quale vengono applicate le azioni.

## Commenti

Prima di eseguire il comando `gfs2_grow`:

- Eseguite il backup dei dati più importanti sul file system.
- Determinate il volume da espandere usato dal file system tramite il comando `df MountPoint`.
- Espandere il volume del cluster con LVM. Per informazioni sull'amministrazione dei volumi LVM consultate *Amministrazione del Logical Volume Manager*

Dopo aver eseguito il comando `gfs2_grow`, eseguite `df` per controllare che il nuovo spazio sia disponibile nel file system.

## Esempi

In questo esempio viene riportata l'espansione del file system presente sulla directory `/mygfs2fs`.

```
[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.
```

## Utilizzo completo

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

### *MountPoint*

Specifica la directory sulla quale è stato montato il file system GFS2.

### *Device*

Specifica il nodo del dispositivo del file system.

[Tabella 4.3, «Opzioni specifiche a GFS2 disponibili durante l'espansione di un file system»](#) descrive le opzioni specifiche a GFS2 utilizzabili durante l'espansione di un file system GFS2.

**Tabella 4.3. Opzioni specifiche a GFS2 disponibili durante l'espansione di un file system**

Opzione	Descrizione
<b>-h</b>	Help. Visualizza un breve messaggio relativo all'utilizzo
<b>-q</b>	Quiet. Diminuisce il livello di verbosità
<b>-r MegaBytes</b>	Specifica la dimensione della nuova risorsa del gruppo. La dimensione predefinita è 256MB.
<b>-T</b>	Test. Esegue tutti i calcoli, ma non esegue la scrittura dei dati sul disco e non espande il file system.
<b>-V</b>	Mostra le informazioni sulla versione del comando

## 4.7. COME AGGIUNGERE I JOURNAL AD UN FILE SYSTEM

Il comando `gfs2_jadd` viene usato per aggiungere i journal ad un file system GFS2. È possibile aggiungere journal ad un file system GFS2 in modo dinamico in qualsiasi momento, senza espandere il volume logico. Il comando `gfs2_jadd` deve essere eseguito su di un file system montato, e solo su di un nodo nel cluster. Tutti i nodi restanti noteranno l'avvenuta espansione.



### NOTA

Se il file system GFS2 è pieno `gfs2_jadd` fallirà anche se il volume logico contenente il file system è stato esteso e risulterà più grande del file system. Tale comportamento si verifica poiché in un file system GFS2 i journal sono file semplici e non metadati 'embedded', per questo motivo l'estensione del volume logico sottostante non fornirà alcuno spazio per i journal.

Prima di aggiungere i journal ad un file system GFS è possibile utilizzare l'opzione `journals` di `gfs2_tool` per sapere il numero di journal presenti nel file system GFS2. I seguenti esempi riportano il numero e la dimensione dei journal nel file system montato su `/mnt/gfs2`.

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

### Utilizzo

```
gfs2_jadd -j Number MountPoint
```

#### *Number*

Specifica il numero di nuovi journal da aggiungere.

#### *MountPoint*

Specifica la directory sulla quale è stato montato il file system GFS2.

## Esempi

In questo esempio, viene aggiunto un journal al file system sulla directory `/mygfs2`.

```
gfs2_jadd -j1 /mygfs2
```

In questo esempio, vengono aggiunti due journal al file system sulla directory `/mygfs2`.

```
gfs2_jadd -j2 /mygfs2
```

## Utilizzo completo

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

### *MountPoint*

Specifica la directory sulla quale è stato montato il file system GFS2.

### *Device*

Specifica il nodo del dispositivo del file system.

Tabella 4.4, «Opzioni specifiche al GFS2 disponibili per l'aggiunta dei journal» descrive le opzioni specifiche al GFS2 utilizzabili durante l'aggiunta dei journal ad un file system GFS2.

Tabella 4.4. Opzioni specifiche al GFS2 disponibili per l'aggiunta dei journal

Flag	Parametro	Descrizione
<b>-h</b>		Help. Visualizza un breve messaggio sull'uso.
<b>-J</b>	<i>MegaBytes</i>	Specifica la dimensione dei nuovi journal in megabyte. La dimensione predefinita del journal è 128 megabyte. La dimensione minima è 32 megabyte. Per aggiungere journal con dimensioni diverse sul file system, eseguire il comando <b>gfs2_jadd</b> per ogni journal. La dimensione specificata durante la creazione del file system, viene arrotondata per difetto rendendola così multipla della dimensione del segmento del journal.
<b>-j</b>	<i>Number</i>	Specifica il numero di nuovi journal da aggiungere tramite il comando <b>gfs2_jadd</b> . Il valore predefinito è 1.
<b>-q</b>		Quiet. Diminuisce il livello di verbosità
<b>-v</b>		Mostra le informazioni sulla versione del comando

## 4.8. DATA JOURNALING

Solitamente GFS2 scrive sul proprio journal solo i metadati. I contenuti del file vengono scritti successivamente sul disco dal processo periodico di sincronizzazione del kernel, il quale svuota i

buffer del file system. Una chiamata `fsync()` su di un file, causerà la scrittura immediata dei dati del file sul disco. La chiamata ritorna quando il disco riporta il completamento della scrittura di tutti i dati.

Il data journaling può causare una riduzione del periodo di `fsync()`, in modo particolare per file più piccoli, poichè i dati del file vengono scritti sul journal in aggiunta ai metadati. Questo vantaggio si riduce velocemente con l'aumento della dimensione del file. La scrittura su file medio-grandi sarà più lento se il data journaling è stato abilitato.

Le applicazioni che si affidano al `fsync()` per eseguire la sincronizzazione dei dati del file possono avere migliori prestazioni attraverso il data journaling. Il data journaling può essere abilitato automaticamente per qualsiasi file GFS2 in una directory con opzione corretta (insieme a tutte le sottodirectory relative). I file esistenti con una lunghezza allo zero possono avere la funzione di journaling dei data abilitata o disabilitata.

Abilitando il data journaling su di una directory, quest'ultima verrà impostata su "inherit jdata", il quale indica che tutti i file e le directory successivamente create nella directory in questione verranno salvate all'interno dei journal. È possibile abilitare o disabilitare il data journaling su di un file con il comando `chattr`:

I seguenti comandi disabilitano il data journaling sul file `/mnt/gfs2/gfs2_dir/newfile`, e successivamente controllano l'impostazione corretta del flag.

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

I seguenti comandi disabilitano il data journaling sul file `/mnt/gfs2/gfs2_dir/newfile`, e successivamente controllano l'impostazione corretta del flag.

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

È possibile utilizzare `chattr` per impostare il flag `j` su di una directory. Quando impostate il suddetto flag per una directory, tutti i file e le directory successivamente create nella directory in questione, verranno salvate nel journal. L'insieme di comandi imposta il flag `j` sulla directory `gfs2_dir`, e controlla se il flag è stato impostato correttamente. Altresì, i comandi creeranno un nuovo file chiamato `newfile` nella directory `/mnt/gfs2/gfs2_dir`, e successivamente controlleranno l'impostazione del flag `j` per il file. Poichè il flag `j` è stato impostato per la directory, anche il `newfile` dovrebbe avere il journaling abilitato.

```
[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
```

## 4.9. CONFIGURAZIONE DEGLI AGGIORNAMENTI ATIME

Ogni inode del file e della directory può avere tre tipi di timestamp:

- `ctime` – Ultimo cambiamento dello stato di un inode

- `mtime` – Ultima modifica dei dati di un file (directory)
- `atime` – L'ultimo accesso ai dati di un file (o directory)

Se gli aggiornamenti `atime` sono stati abilitati, come per impostazione predefinita sul GFS2 ed altri file system di Linux, ogni qualvolta il file viene letto, l'inode corrispondente deve essere aggiornato.

Poichè un numero limitato di applicazioni utilizzano le informazioni fornite da `atime`, i suddetti aggiornamenti possono richiedere una quantità non necessaria di processi di scrittura e di attività di file locking. Tale traffico può influenzare negativamente le prestazioni, e per questo motivo è consigliato disabilitare o ridurre la frequenza degli aggiornamenti `atime`.

Per questo motivo sono disponibili due metodi attraverso i quali è possibile ridurre gli effetti dovuti agli aggiornamenti `atime`:

- Montaggio con `relatime` (atime relativo), il quale aggiorna `atime` se l'aggiornamento `atime` precedente è più vecchio rispetto all'aggiornamento `mtime` o `ctime`.
- Montaggio con `noatime` il quale disabilita gli aggiornamenti `atime` sul file system in questione.

#### 4.9.1. Montaggio con `relatime`

L'opzione di montaggio di Linux `relatime` (atime relativo) può essere specificata quando il file system è stato montato. Ciò specifica che `atime` viene aggiornato se l'aggiornamento `atime` precedente è più vecchio rispetto all'aggiornamento `mtime` o `ctime`.

##### Utilizzo

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

##### *BlockDevice*

Specifica il dispositivo a blocchi dove risiede il file system GFS2.

##### *MountPoint*

Specifica la directory dove il file system GFS2 deve essere montato.

##### Esempio

In questo esempio, il file system GFS2 risiede su `/dev/vg01/lvol0` ed è montato sulla directory `/mygfs2`. Gli aggiornamenti `atime` si verificano solo se l'aggiornamento `atime` precedente è più vecchio rispetto all'aggiornamento `mtime` o `ctime`.

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

#### 4.9.2. Montaggio con `noatime`

L'opzione di montaggio di Linux `noatime` può essere specificata se il file system è stato montato; questa opzione disabilita gli aggiornamenti `atime` sul file system in questione.

## Utilizzo

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

### *BlockDevice*

Specifica il dispositivo a blocchi dove risiede il file system GFS2.

### *MountPoint*

Specifica la directory dove il file system GFS2 deve essere montato.

## Esempio

In questo esempio il file system GFS2 risiede su `/dev/vg01/lvol0` e viene montato sulla directory `/mygfs2` con aggiornamenti `atime` disabilitati.

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

## 4.10. SOSPENSIONE DI UNA ATTIVITÀ SU DI UN FILE SYSTEM

È possibile sospendere la scrittura su di un file system utilizzando il comando `dmsetup suspend`. Tale sospensione permette l'utilizzo di istantanee del dispositivo basato sull'hardware per catturare il file system in uno stato consistente. Il comando `dmsetup resume` termina lo stato di sospensione.

## Utilizzo

### Inizio sospensione

```
dmsetup suspend MountPoint
```

### Fine sospensione

```
dmsetup resume MountPoint
```

### *MountPoint*

Specifica il file system.

## Esempi

In questo esempio vengono sospesi i processi di scrittura sul file system `/mygfs2`.

```
# dmsetup suspend /mygfs2
```

In questo esempio viene terminata la sospensione dei processi di scrittura sul file system `/mygfs2`.

```
# dmsetup resume /mygfs2
```

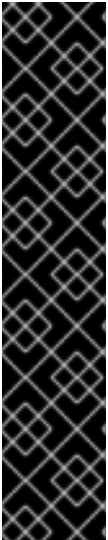
## 4.11. COME RIPRISTINARE UN FILE SYSTEM

Se si verifica il fallimento dei nodi ed il file system è stato montato, il journaling del file system permette un processo di ripristino molto rapido. Tuttavia, se un dispositivo di storage perde alimentazione o è fisicamente scollegato, in questi casi si potrebbe verificare una corruzione del file system. (Il Journaling non può essere usato per eseguire il ripristino da errori che riguardano il sottosistema di storage.) Se si verifica questo tipo di corruzione è possibile ripristinare il normale funzionamento del GFS2 utilizzando il comando `fsck.gfs2`.



## IMPORTANTE

Il comando `fsck.gfs2` deve essere eseguito solo su di un file system non montato da tutti i nodi.



## IMPORTANTE

Non eseguire il controllo del file system GFS2 al momento dell'avvio utilizzando il comando `fsck.gfs2`. Il comando `fsck.gfs2` non è in grado di determinare durante questo processo se il file system sia stato montato da un altro nodo presente nel cluster. Eseguire il comando `fsck.gfs2` manualmente dopo il riavvio del sistema.

Per essere sicuri che il comando `fsck.gfs2` non venga eseguito su un file system GFS2 al momento dell'avvio, modificare `/etc/fstab` in modo tale che le ultime due colonne per il mount point del file system GFS2 riportino "0 0" e non "1 1" (o qualsiasi altro numero), come riportato nel seguente esempio:

```
/dev/VG12/lv_svr_home /svr_home gfs2
defaults,noatime,nodiratime,noquota 0 0
```



## NOTA

Se avete precedentemente utilizzato il comando `gfs_fsck` sui file system GFS, vi informiamo che il comando `fsck.gfs2` differisce da alcune versioni precedenti di `gfs_fsck` nel modo di seguito riportato:

- La selezione di `Ctrl+C` durante l'esecuzione di `fsck.gfs2` interrompe e mostra un prompt il quale richiederà all'utente se abortire il comando, saltare il resto della fase corrente o continuare la processazione.
- È possibile aumentare il livello di verbosità utilizzando il flag `-v`. Aggiungendo un secondo flag `-v` aumenterete ulteriormente tale livello.
- È possibile diminuire il livello di verbosità utilizzando il flag `-q`. Aggiungendo un secondo flag `-q` diminuirete ulteriormente tale livello.
- L'opzione `-n` apre un file system in modalità di sola lettura, e risponderà automaticamente `no` ad ogni richiesta. L'opzione fornisce un modo attraverso il quale si cercherà di utilizzare il comando per rilevare la presenza di errori, senza però permettere al comando `fsck.gfs2` di essere implementato.

Consultate la pagina `man` di `fsck.gfs2` per informazioni aggiuntive su altre opzioni del comando.

L'esecuzione del comando `fsck.gfs2` necessita di una memoria del sistema superiore alla memoria usata per il sistema operativo e dal kernel. Ogni blocco della memoria nel file system GFS2 ha bisogno



di circa cinque bit di memoria aggiuntiva o 5/8 di byte. Per questo motivo per una stima di byte della memoria necessari per eseguire il comando `fsck.gfs2` sul file system, determinare il numero di blocchi contenuti dal file system e moltiplicare quel numero per 5/8.

Per esempio, per determinare approssimativamente la quantità di memoria necessaria per eseguire il comando `fsck.gfs2` su di un file system GFS2 di 16TB con una dimensione del blocco di 4K, determinare prima il numero di blocchi di memoria contenuti nel file system, e dividere successivamente 16Tb con 4K.

```
17592186044416 / 4096 = 4294967296
```

Poichè questo file system contiene 4294967296 blocchi, moltiplicare quel numero con 5/8 in modo da determinare il numero di byte della memoria necessari:

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

Questo file system ha bisogno di 2.6GB circa di memoria libera per eseguire il comando `fsck.gfs2`. Se la dimensione del blocco era 1K, l'esecuzione del comando `fsck.gfs2` avrà bisogno di quattro volte la memoria, o approssimativamente di 11GB.

## Utilizzo

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

**-y**

Con il flag `-y` tutte le domande verranno risposte con un `yes`. Con `-y`, il comando `fsck.gfs2` non vi richiederà alcuna risposta prima di eseguire le modifiche.

## *BlockDevice*

Specifica il dispositivo a blocchi dove risiede il file system GFS2.

## Esempio

In questo esempio il file system GFS2 che risiede sul dispositivo a blocchi `/dev/testvol/testlv` viene riparato. Tutte le richieste di riparazione fatte vengono risposte automaticamente con `yes`.

```
[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
```

## 4.12. MOUNT BIND E CONTEXT-DEPENDENT PATH NAMES

I file system GFS2 non forniscono alcun supporto per i Context-Dependent Path Name (CDPN), i quali permettono all'utente di creare link simbolici che indicano file o directory a destinazione variabili. Per questa funzionalità in GFS2, è possibile utilizzare l'opzione **bind** del comando **mount**.

L'opzione **bind** del comando **mount** permette di rimontare parte della gerarchia del file su di una posizione diversa, pur restando disponibile nella posizione originale. Di seguito viene riportato il formato del comando:

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

Dopo aver eseguito questo comando i contenuti della directory *olddir* saranno disponibili in due posizioni: *olddir* e *newdir*. È possibile utilizzare questa opzione per rendere disponibile un file singolo in due posizioni.

Per esempio, dopo aver eseguito i seguenti comandi i contenuti di `/root/tmp` risulteranno identici ai contenuti della directory `/var/log` precedentemente montata.

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

Alternativamente potrete utilizzare una entry all'interno del file `/etc/fstab` in modo da avere gli stessi risultati al momento dell'esecuzione del **mount**. La suddetta entry di `/etc/fstab` permetterà ai contenuti di `/root/tmp` di essere identici ai contenuti della directory `/var/log`.

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

Dopo aver montato il file system è possibile utilizzare il comando **mount** per controllare il montaggio del file system, come riportato nel seguente esempio.

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

Con un file system in grado di supportare il Context-Dependent Path Name, potrete definire la directory `/bin` come un Context-Dependent Path Name risolvibile in uno dei seguenti percorsi, a seconda dell'architettura del sistema.

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

È possibile ottenere la stessa funzionalità creando una directory `/bin` vuota. Successivamente utilizzando uno script o una entry nel file `/etc/fstab`, è possibile montare ogni directory individuale dell'architettura sulla directory `/bin`, con un comando `mount -bind`. Per esempio, è possibile usare il seguente comando all'interno di uno script.

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

Alternativamente potrete utilizzare la seguente entry all'interno del file `/etc/fstab`.

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

Un `mount bind` può fornire una maggiore flessibilità rispetto ad un Context-Dependent Path Name, poichè l'utente sarà in grado di usare questa caratteristica per montare directory diverse in base al criterio da lui definito (come ad esempio il valore di `%fill` per il file system). I Context-Dependent Path Names risultano molto più limitati nel contenuto. Da notare tuttavia che sarà necessario scrivere il proprio script per eseguire il `mount` in base al valore di `%fill`.



#### AVVERTIMENTO

Se montate un file system utilizzando l'opzione `bind` ed il file system originale è stato precedentemente montato in modalità `rw`, il nuovo file system sarà anch'esso montato in modalità `rw`, anche se utilizzate il flag `ro`; il flag `ro` verrà ignorato senza generare alcun messaggio. In questo caso, il nuovo file system potrebbe essere contrassegnato in modo non corretto come `ro` nella directory `/proc/mounts`.

## 4.13. MOUNT BIND E ORDINE DI MONTAGGIO DI UN FILE SYSTEM

Se usate l'opzione `bind` del comando `mount` assicuratevi che i file system siano montati nell'ordine corretto. Nel seguente esempio la directory `/var/log` deve essere montata prima di eseguire il `mount bind` sulla directory `/tmp`:

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

L'ordine di montaggio dei file system è determinato nel modo seguente:

- In generale il montaggio dei file system è determinato dall'ordine nel quale il file system appare all'interno del file `fstab`. Eccezioni a questa regola sono i file system montati con il flag `_netdev` o i file system che possiedono i propri script `init`.
- Un file system con il proprio script `init` è montato nelle fasi successive del processo di inizializzazione, dopo il file system presente nel file `fstab`.
- File system montati con il flag `_netdev` sono montati dopo aver abilitato una rete sul sistema.

Se la configurazione necessita di una creazione di un `mount bind` sul quale montare un file system GFS2 sarà possibile ordinare il file `fstab` nel modo seguente:

1. Montate i file system locali necessari per il mount bind.
2. Eseguire il mount bind della directory sulla quale montare il file system GFS2.
3. Montare il file system GFS2.

Se la configurazione richiede l'esecuzione di un mount bind di una directory locale o file system su di un file system GFS2, anche se elencherete i file system nell'ordine corretto nel file `fstab` non sarà possibile montare i file system correttamente poiché il file system GFS2 non verrà montato fino a quando non sarà eseguito lo script `init` di GFS2. In questo caso è consigliato scrivere uno script `init` per eseguire il mount bind in modo tale che questa operazione non sia eseguita prima del montaggio del file system GFS2.

Il seguente script è un esempio di script `init` personalizzato. Esso esegue un mount bind di due directory su due directory di un file system GFS2. In questo esempio è presente un mount point GFS2 esistente su `/mnt/gfs2a`, montato al momento dell'esecuzione dello script `init` di GFS2, dopo l'avvio del cluster.

In questo script d'esempio i valori dell'istruzione `chkconfig` indicano quanto di seguito riportato:

- 345 indica i run level nei quali verrà iniziato lo script
- 29 è la priorità d'avvio la quale in questo caso indica che lo script verrà eseguito al momento dell'avvio dopo lo script `init` di GFS2, il quale avrà una priorità di 26
- 73 è la priorità d'arresto, in questo caso indica che lo script verrà arrestato durante il processo di shutdown prima dello script GFS2, il quale ha una priorità d'arresto pari a 74.

I valori d'avvio e d'arresto indicano la possibilità di eseguire manualmente l'azione indicata tramite l'esecuzione di un comando `service start` ed un `service stop`. Per esempio, se lo script è `fredwilma` allora sarà possibile eseguire `service fredwilma start`.

Questo script dovrà essere archiviato nella directory `/etc/init.d` con gli stessi permessi di altri script presenti in quella directory. Successivamente sarà possibile eseguire un comando `chkconfig on` per collegare lo script ai run level indicati. Per esempio se lo script è `fredwilma` allora sarà possibile eseguire `chkconfig fredwilma on`.

```
#!/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

```

## 4.14. FUNZIONE WITHDRAW DI GFS2

La funzione *withdraw* di GFS2 è una funzione per l'integrità dei dati dei file system GFS2 in un cluster. Se il modulo del kernel GFS2 rileva una inconsistenza nel file system GFS2 dopo una operazione I/O, il file system non sarà disponibile al cluster. L'operazione I/O viene arrestata ed il sistema attende la fine di eventuali errori, impedendo il verificarsi di ulteriori danni. Se si verifica questo comportamento sarà possibile arrestare manualmente qualsiasi altro servizio o applicazione, e successivamente riavviare e rimontare il file system GFS2 per eseguire nuovamente i journal. Se il problema persiste sarà possibile smontare il file system da tutti i nodi nel cluster ed eseguire un ripristino con il comando `fsck.gfs2`. La funzione *withdraw* di GFS è meno severa di un kernel panic, il quale potrebbe causare l'esecuzione del fencing di un nodo da parte di un altro nodo.

Se il sistema è stato configurato con lo script d'avvio `gfs2` abilitato ed il file system GFS2 è stato incluso nel file `/etc/fstab`, il file system GFS2 verrà rimontato al momento del reboot. Se si verifica il ritiro del file system GFS2 a causa di una presunta corruzione del file system, è consigliato eseguire il comando `fsck.gfs2` prima di rimontare il file system stesso. In questo caso per non rimontare il file system al momento dell'avvio eseguire la seguente procedura:

1. Disabilitare momentaneamente lo script d'avvio sul nodo interessato con il seguente comando:

```
# chkconfig gfs2 off
```

2. Riavviare il nodo interessato avviando il software del cluster. Il file system GFS2 non verrà montato.
3. Smontare il file system da ogni nodo nel cluster.

4. Eseguire `fsck.gfs2` sul file system solo da un nodo in modo da assicurarsi che il file system non sia corrotto.
5. Abilitare nuovamente lo script d'avvio sul nodo interessato eseguendo il seguente comando:  

```
# chkconfig gfs2 on
```
6. Rimontare il file system GFS2 da tutti i nodi nel cluster.

Un esempio di inconsistenza in grado di causare il ritiro di un GFS2 è un conteggio dei blocchi incorretto. Quando il kernel del GFS rimuove un file da un file system esso rimuove sistematicamente tutti i blocchi relativi ai metadati ed ai dati associati con quel file. Una volta terminato verrà controllato il conteggio dei blocchi. Se il conteggio è diverso da uno (ciò significa che tutto ciò che è rimasto è il solo inode del disco), ciò indicherà una inconsistenza del file system poiché il conteggio non corrisponde all'elenco di blocchi trovati.

È possibile sovrascrivere la funzione `withdraw` del GFS2 montando il file system specificando l'opzione `-o errors=panic`. Se la suddetta opzione è stata specificata, qualsiasi errore che normalmente è in grado di causare il ritiro del sistema causerà un processo di `panic` del sistema stesso. Tale operazione arresterà le comunicazioni con il cluster del nodo, causando così l'isolamento del nodo stesso.

Internamente la funzione `withdraw` di GFS2 richiede un invio di un messaggio da parte del kernel al demone `gfs_controld` che richiede la revoca. Il demone `gfs_controld` esegue il programma `dmsetup` per posizionare il target di errore del device mapper sotto il file system impedendone l'accesso al dispositivo a blocchi. Successivamente verrà indicato al kernel che questa operazione è stata completata. Questo è il motivo per il quale il requisito per il supporto del GFS2 è quello di usare sempre un dispositivo CLVM sotto il GFS2, in caso contrario non sarà possibile inserire alcun target per il device mapper.

Lo scopo di un target d'errore per il device mapper è quello di assicurare che tutte le operazioni future di I/O possano risultare in un errore I/O il quale permetterà al file system un processo di smontaggio corretto. Ne risulta che, quando si verifica un processo di revoca, sarà normale visualizzare un certo numero di errori I/O del device mapper presenti nei log del sistema.

Occasionalmente la revoca può fallire se il programma `dmsetup` non è in grado di inserire un target d'errore come richiesto. Ciò si può verificare se è presente memoria insufficiente al momento della revoca e quindi sarà impossibile recuperare alcuna memoria a causa del problema che ha innescato la revoca stessa.

Una revoca non indica sempre la presenza di un errore in GFS2. Talvolta la funzione `withdraw` può essere innescata dagli errori I/O del dispositivo per il dispositivo a blocchi sottostante. A tal proposito è fortemente consigliato controllare i log del sistema.

## CAPITOLO 5. DIAGNOSI E CORREZIONI DEI PROBLEMI CON I FILE SYSTEM GFS2

Questo capitolo fornisce le informazioni sulle problematiche comuni relative al GFS2 e sulla loro risoluzione.

### 5.1. IL FILE SYSTEM GFS2 CON PRESTAZIONE LENTA

In certi casi il file system GFS2 potrà avere prestazioni più lente rispetto al file system EXT3. Prestazioni lente del GFS2 possono essere causate da un certo numero di fattori e durante usi particolari. In questo documento sono riportate le informazioni necessarie su come affrontare le suddette problematiche.

### 5.2. SOSPENSIONE DEL FILE SYSTEM GFS2 E RIAVVIO DI UN NODO

Se il file system GFS2 entra in sospensione e non ritorna alcun comando appena eseguito e riavviando un nodo specifico il sistema torna al suo stato normale, ciò potrebbe indicare un problema di blocco o la presenza di un bug. In tal caso raccogliere le seguenti informazioni:

- Il `gfs2 lock dump` per il file system su ogni nodo:

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

- Il DLM lock dump per il file system su ogni nodo: Per ottenere questo tipo di informazione usare `dlm_tool`:

```
dlm_tool lockdebug -sv lname.
```

In questo comando *lname* è il nome usato da DLM per il file system in questione. Il suddetto valore è disponibile all'interno dell'output del comando `group_tool`.

- L'output del comando `sysrq -t`.
- I contenuti del file `/var/log/messages`.

Una volta raccolti i dati richiesti aprire un ticket con il Red Hat Support e fornire le informazioni necessarie.

### 5.3. SOSPENSIONE DEL FILE SYSTEM GFS2 E RIAVVIO DI TUTTI I NODI

Se il file system GFS2 entra in sospensione e non ritorna alcun comando appena eseguito richiedendo il riavvio di tutti i nodi nel cluster, controllare quanto di seguito riportato:

- La presenza di un fencing fallito. I file system GFS2 verranno sospesi per assicurare l'integrità dei dati se si verifica il fallimento del processo di fencing. Controllare i log dei messaggi per la presenza di processi di fencing falliti al momento della sospensione. Assicurarsi che il fencing sia stato configurato correttamente.
- Il file system GFS2 potrebbe essere stato revocato. Controllate i log per la presenza della parola `withdraw` e di messaggi del GFS2 i quali indicano che il file system è stato revocato. La presenza di `withdraw` può indicare una corruzione del file system, il fallimento dello storage o

un bug. Smontate il file system, aggiornate il pacchetto `gfs2-utils` ed eseguite il comando `fsck` sul file system per ripristinare il servizio. Aprite un ticket con il Red Hat Support. Informatevi della presenza di una revoca GFS2 e fornite i log necessari.

Per informazioni sulla funzione `withdraw` di GFS2 consultare [Sezione 4.14, «Funzione Withdraw di GFS2»](#).

- Questo errore potrebbe indicare la presenza di un blocco o di un bug. Raccogliere le informazioni durante uno di questi eventi ed aprire un ticket con il Red Hat Support, come descritto in [Sezione 5.2, «Sospensione del file system GFS2 e riavvio di un nodo»](#) .

## 5.4. IMPOSSIBILE MONTARE IL FILE SYSTEM GFS2 SU UN NODO APPENA AGGIUNTO AL CLUSTER

Se aggiungete un nuovo nodo al cluster e successivamente non siete in grado di montare il file system GFS2 sul nodo in questione allora potreste essere in presenza di un numero minore di journal sul file system GFS2 rispetto ai nodi che tentano di accedere al file system stesso. È necessario avere un journal per host GFS2 sul quale desiderate montare il file system (con l'eccezione di file system GFS2 montati con il set di opzioni di montaggio `spectator` poiché l'uso del journal non è richiesto). È possibile aggiungere i journal ad un file system GFS2 con il comando `gfs2_jadd` come descritto in [Sezione 4.7, «Come aggiungere i journal ad un file system»](#) .

## 5.5. SPAZIO INDICATO COME USATO IN UN FILE SYSTEM VUOTO

Se il file system GFS2 è vuoto il comando `df` mostrerà che lo spazio sarà stato usato. Ciò si verifica poiché i journal del file system GFS2 consumano spazio (numero di journal \* dimensione del journal) sul disco. Se avete creato un file system GFS2 con un numero di journal molto elevato o se avete specificato una dimensione molto grande del journal allora sarà possibile visualizzare che (numero di journal \* dimensione di journal) lo spazio è già stato utilizzato quando eseguirete `df`. Anche se non specificate un numero molto elevato o una dimensione molto grande dei journal, file system GFS2 piccoli (1GB o minori) mostreranno una quantità molto grande di spazio usato con la dimensione del journal predefinita.



## CAPITOLO 6. CONFIGURAZIONE DI UN FILE SYSTEM GFS2 IN UN CLUSTER PACEMAKER

La seguente procedura riporta le fasi necessarie per impostare un cluster Pacemaker con un file system GFS2.

Dopo aver installato il software, GFS2 e i pacchetti LVM clusterizzati su ogni nodo, avviare i servizi `cman`, `clvmd` e `pacemaker` su ogni nodo e creare il cluster Pacemaker. È necessario configurare il fencing per il cluster. Per informazioni su come configurare un cluster Pacemaker consultare la sezione *Configurazione di Red Hat High Availability Add-On con Pacemaker*

1. Impostare il parametro globale di Pacemaker `no_quorum_policy` su `freeze`.



### NOTA

Per impostazione predefinita il valore di `no-quorum-policy` è impostato su `stop`, e indica che una volta perso il quorum tutte le risorse sulla partizione restante verranno immediatamente arrestate. Generalmente questa impostazione predefinita è quella più sicura e ottimale, ma diversamente dalla maggior parte delle risorse, GFS2 ha bisogno di un quorum per funzionare. Dopo la perdita di un quorum entrambe le applicazioni che utilizzano GFS2 eseguono il montaggio, questa operazione però non potrà essere arrestata correttamente. Qualsiasi tentativo di arrestare le risorse senza alcun quorum fallirà, isolando il cluster ogni qualvolta il quorum verrà perso.

Per risolvere questo problema impostare `no-quorum-policy=freeze` durante l'utilizzo di GFS2. Ciò significa che alla perdita del quorum, la partizione restante non eseguirà alcuna operazione fino a quando non sarà ristabilito il quorum.

```
# pcs property set no-quorum-policy=freeze
```

2. Dopo aver controllato l'impostazione del tipo 3 di locking nel file `/etc/lvm/lvm.conf`, abilitando così il supporto del locking clusterizzato, creare il LV clusterizzato e formattare il volume con un file system GFS2. Create un numero di journal sufficiente per ogni nodo presente nel cluster.

```
# pvcreate /dev/vdb
# vgcreate -Ay -cy cluster_vg /dev/vdb
# lvcreate -L5G -n cluster_lv cluster_vg
# mkfs.gfs2 -j2 -p lock_dlm -t rhel7-demo:gfs2-demo
/dev/cluster_vg/cluster_lv
```

3. Configurazione di una risorsa `clusterfs`.

Non aggiungere il file system al file `/etc/fstab` poichè esso verrà gestito come una risorsa del cluster Pacemaker. È possibile specificare le opzioni di montaggio come parte della configurazione delle risorse utilizzando `options=options`. Eseguire il comando `pcs resource describe Filesystem` per tutte le opzioni di configurazione.

Questo comando usato per la creazione delle risorse del cluster specifica l'opzione di montaggio `noatime`.

■

```
# pcs resource create clusterfs Filesystem
device="/dev/cluster_vg/cluster_lv" directory="/var/mountpoint"
fstype="gfs2" "options=noatime" op monitor interval=10s on-
fail=fence clone interleave=true
```

**4. Verificare che GFS2 sia stato montato come previsto.**

```
# mount |grep /mnt/gfs2-demo
/dev/mapper/cluster_vg-cluster_lv on /mnt/gfs2-demo type gfs2
(rw,noatime,seclabel)
```

**5. (Facoltativo) Riavviare tutti i nodi del cluster per verificare la persistenza di gfs2 e il ripristino.**

## APPENDICE A. GESTIONE QUOTA DEL GFS2 CON IL COMANDO

### GFS2\_QUOTA

Con Red Hat Enterprise Linux 6.1 è ora disponibile il supporto delle funzioni standard dei quota di Linux. Per il suo utilizzo installare l'RPM `quota`. Questo è il metodo preferito per gestire i quota su GFS2 ed è consigliato il suo uso per tutte le nuove implementazioni di GFS2 che utilizzano il quota. Per informazioni sull'uso delle funzioni standard del quota di Linux consultare [Sezione 4.5, «Gestione quota del GFS2»](#).

Nelle versioni precedenti di Red Hat Enterprise Linux GFS2 richiedeva l'uso del comando `gfs2_quota` per gestire i quota. Questa appendice documenta l'uso del comando `gfs2_quota` per la gestione dei quota del file system GFS2.

### A.1. IMPOSTAZIONE DEI QUOTA CON IL COMANDO `GFS2_QUOTA`

Per ogni user ID (UID) o group ID (GID) sono disponibili due tipi di impostazioni del quota: *hard limit* e *soft limit*.

Un *hard limit* è la quantità di spazio che può essere utilizzato. Il file system impedirà all'utente o al gruppo di utilizzare una quantità maggiore di spazio. Un valore zero dell'*hard limit* significa che non è presente alcun limite.

Un *limite soft* è un valore più basso rispetto a quello *hard*. In questo caso il file system notificherà all'utente o gruppo quando verrà raggiunto il *limite soft* informandoli della quantità di spazio utilizzato. Un valore zero significa che non è presente alcun limite.

Per impostare i limiti usare il comando `gfs2_quota`. Utilizzare il comando su un nodo singolo sul quale è montato il GFS2.

Per impostazione predefinita il quota enforcement non è impostato sui file system GFS2. Per abilitare la contabilità dei quota usare `quota=` del comando `mount` durante il montaggio del file system GFS2 come riportato in [Sezione A.4, «Come abilitare/disabilitare il Quota Enforcement»](#).

#### Uso

##### Impostazione dei quota, Hard Limit

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

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

##### Impostazione dei quota, Warn Limit

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

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

#### User

Un user ID da limitare o avvertire. Esso potrà essere un nome utente del file password o il numero UID.

### **Group**

Un group ID da limitare o avvertire. Esso potrà essere un nome del gruppo del file del gruppo o il numero GID.

### **Size**

Specifica il nuovo valore per un limite o un avviso. Per impostazione predefinita il valore è espresso in unità di megabyte. I flag **-k**, **-s** e **-b** modificano il valore in kilobyte, settori e blocchi del file system.

### **MountPoint**

Specifica il file system GFS2 al quale vengono applicate le azioni.

## **Esempi**

In questo esempio viene impostato un hard limit per l'utente *Bert* su 1024 megabytes (1 gigabyte) sul file system */mygfs2*.

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

In questo esempio viene impostato un soft limit per il group ID 21 su 50 kilobyte sul file system */mygfs2*.

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

## **A.2. COME VISUALIZZARE L'UTILIZZO ED I LIMITI DEL QUOTA CON IL COMANDO `GFS2_QUOTA`**

I limiti del quota e l'utilizzo corrente possono essere visualizzati per un utente specifico o gruppo tramite il comando `gfs2_quota get`. I contenuti del quota file possono essere visualizzati usando il comando `gfs2_quota list`, in tal caso tutti gli ID con un soft limit, hard limit non zero, o un valore sono elencati.

### **Uso**

#### **Visualizzazione dei limiti di quota per un utente**

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

#### **Visualizzazione dei limiti di quota per un gruppo**

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

#### **Visualizzazione dell'intero Quota File**

```
gfs2_quota list -f MountPoint
```

#### **User**

Un user ID per la visualizzazione delle informazioni relative ad un utente specifico. Esso potrà essere un nome utente del file password o il numero UID.

### **Group**

Un group ID per la visualizzazione delle informazioni relative ad un gruppo specifico. Esso potrà essere il nome del gruppo o il numero GID.

### **MountPoint**

Specifica il file system GFS2 al quale vengono applicate le azioni.

## **Output del comando**

Le informazioni relative al quota del GFS2 ottenute utilizzando il comando `gfs2_quota` sono visualizzate nel modo seguente:

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

I valori di *LimitSize*, *WarnSize*, e *Value* sono rappresentati per impostazione predefinita in unità di megabyte. L'aggiunta dei flag `-k`, `-s`, o `-b` alla linea di comando modificherà i valori in kilobyte, settori e blocchi del sistema.

### **User**

Un nome utente o ID ai quali vengono associati i dati.

### **Group**

Un nome del gruppo o ID ai quali vengono associati i dati.

### **LimitSize**

L'hard limit impostato per l'utente o il gruppo. Questo valore è zero se non è stato impostato alcun limite.

### **Value**

L'ammontare dello spazio del disco usato dall'utente o gruppo.

## **Commenti**

Quando visualizzate le informazioni sul quota il comando `gfs2_quota` non risolve gli UID e GID in nomi se l'opzione `-n` viene aggiunta alla linea del comando.

Lo spazio assegnato ai file nascosti di GFS2 può essere escluso dai valori visualizzati per il root UID e GID aggiungendo l'opzione `-d` alla linea del comando. Tale impostazione è utile quando si cerca di far corrispondere i numeri del `gfs2_quota` con i risultati di un comando `du`.

## **Esempi**

Questo esempio mostra le informazioni relative al quota per tutti gli utenti e gruppi con un limite o che utilizzano qualsiasi spazio sul file system `/mygfs2`.

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

Questo esempio mostra le informazioni del quota in settori per il gruppo `users` sul file system `/mygfs2`.

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

### A.3. SINCRONIZZAZIONE DEL QUOTA CON IL COMANDO `GFS2_QUOTA`

GFS2 archivia tutte le informazioni relative al quota nel proprio file interno sul disco. Un nodo del GFS2 non aggiorna il quota file per ogni processo di scrittura del file system; per impostazione predefinita aggiorna il quota file ogni 60 secondi. Tale processo è necessario per evitare contrasti tra i nodi durante la scrittura sul quota file, con un relativo rallentamento delle prestazioni.

Prima del raggiungimento dei limiti quota da parte di un utente o gruppo, GFS2 riduce dinamicamente l'intervallo tra un aggiornamento del file quota e l'altro, in modo da evitare il superamento del limite. L'intervallo di tempo medio tra i vari processi di sincronizzazione è regolabile, `quota_quantum`. È possibile modificare il valore predefinito di 60 secondi usando l'opzione di montaggio `quota_quantum=` come descritto in [Tabella 4.2, «Opzioni di mount specifiche al GFS2»](#). Altresì, il parametro `quota_quantum` deve essere impostato su ogni nodo e ogni qualvolta viene montato il file system. Le modifiche del parametro `quota_quantum` non verranno mantenute dopo i processi `umount`. È possibile aggiornare il valore `quota_quantum` con `mount -o remount`.

Sarà possibile utilizzare il comando `gfs2_quota sync` per sincronizzare le informazioni del quota di un nodo con il quota file sul disco tra gli aggiornamenti automatici eseguiti da GFS2.

#### Uso

##### Sincronizzazione informazioni del quota

```
gfs2_quota sync -f MountPoint
```

##### *MountPoint*

Specifica il file system GFS2 al quale vengono applicate le azioni.

##### Regolazione ora tra le sincronizzazioni

```
mount -o quota_quantum=secs,remount BlockDevice MountPoint
```

##### *MountPoint*

Specifica il file system GFS2 al quale vengono applicate le azioni.

##### *secs*

Specifica il nuovo periodo tra le normali sincronizzazioni del quota file del GFS2. I valori più piccoli possono rallentare le prestazioni.

#### Esempi

Questo esempio sincronizza le informazioni del quota del nodo sul quale viene eseguito sul file system `/mygfs2`.

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

In questo esempio viene modificato il periodo predefinito tra aggiornamenti regolari del quota file in una ora (3600 secondi) per il file system `/mnt/mygfs2`, quando si esegue il rimontaggio del file system sul volume logico `/dev/volgroup/logical_volume`.

```
# mount -o quota_quantum=3600,remount /dev/volgroup/logical_volume
/mnt/mygfs2
```

## A.4. COME ABILITARE/DISABILITARE IL QUOTA ENFORCEMENT

Nei file system GFS2 il quota enforcement è disabilitato per impostazione predefinita. Per abilitare il quota enforcement per un file system montate il file system specificando l'opzione `quota=on`.

### Uso

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

Per montare un file system con un quota enforcement disabilitato montare il file system specificando l'opzione `quota=off`. Questa è l'impostazione predefinita.

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

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

Specifica che il quota enforcement è abilitato o disabilitato quando il file system è montato.

### *BlockDevice*

Specifica il dispositivo a blocchi sul quale risiede il file system GFS2.

### *MountPoint*

Specifica la directory dove montare il file system GFS2.

### Esempi

In questo esempio il file system GFS2 su `/dev/vg01/lvol0` è montato sulla directory `/mygfs2` con un quota enforcement abilitato.

```
# mount -o quota=on /dev/vg01/lvol0 /mygfs2
```

## A.5. COME ABILITARE LA CONTABILITÀ DEI QUOTA

È possibile controllare l'utilizzo del disco e la contabilità dei quota per ogni utente e gruppo senza implementare valori limiti e di avvertimento. Per fare questo montare il file system specificando l'opzione `quota=account`.

## Uso

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

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

Specifica la gestione delle statistiche dell'utente e del gruppo da parte del file system, anche se i limiti di quota non sono implementati.

### ***BlockDevice***

Specifica il dispositivo a blocchi sul quale risiede il file system GFS2.

### ***MountPoint***

Specifica la directory dove montare il file system GFS2.

## Esempio

In questo esempio il file system GFS2 presente su `/dev/vg01/lvol0` viene montato sulla directory `/mygfs2` con una contabilità del quota abilitata.

```
# mount -o quota=account /dev/vg01/lvol0 /mygfs2
```



## APPENDICE B. CONVERSIONE DI UN FILE SYSTEM DA GFS A GFS2

Poichè la release Red Hat Enterprise Linux 6 non supporta i file system GFS sarà necessario aggiornare qualsiasi file system GFS esistente a GFS2 con il comando `gfs2_convert`. Da notare che è necessario eseguire la suddetta procedura su di un sistema Red Hat Enterprise Linux 5 prima di eseguire l'aggiornamento a Red Hat Enterprise Linux 6.



### AVVERTIMENTO

Prima di eseguire la conversione del file system GFS eseguire il back up del file system poichè il processo di conversione è irreversibile e qualsiasi errore presente durante questo processo, può causare una interruzione inaspettata del programma con una conseguente inoperabilità del file system.

Prima di convertire il file system GFS sarà necessario usare il comando `gfs_fsck` per controllare il file system e correggere qualsiasi errore.

Se la conversione da GFS a GFS2 viene interrotta a causa di una interruzione dell'alimentazione o a causa di altri motivi, riavviate il tool di conversione. È importante non eseguire il comando `fsck.gfs2` sul file system fino a quando il processo di conversione non sia terminato.

Durante la conversione di file system pieni o quasi pieni si potrebbe verificare la presenza di spazio insufficiente per le strutture dei dati del file system GFS2. In questi casi la dimensione di tutti i journal viene ridotta in modo uniforme in modo da avere spazio sufficiente.

### B.1. CONVERSIONE DEI CONTEXT-DEPENDENT PATH NAMES

I file system GFS2 non forniscono alcun supporto per i Context-Dependent Path Name (CDPN), i quali permettono all'utente di creare link simbolici che indicano file o directory a destinazione variabili. Per questa funzionalità in GFS2, è possibile utilizzare l'opzione `bind` del comando `mount`.

Il comando `gfs2_convert` identifica CDPN sostituendoli con directory vuote aventi lo stesso nome. Per configurare i `bind mount` in modo da sostituire i CDPN, sarà necessario sapere i percorsi completi delle destinazioni dei link relativi ai CDPN che state sostituendo. Prima di convertire i file system usare il comando `find` per identificare i link.

Il seguente comando elenca i link simbolici che indicano ad un `hostname` CDPN:

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

In modo simile è possibile eseguire il comando `find` per altri CDPN (`mach`, `os`, `sys`, `uid`, `gid`, `jid`). Poichè i nomi CDPN possono avere un formato `@hostname` o `{hostname}` eseguire il comando `find` per ogni variante.

Per maggiori informazioni sui bind mount e sui nomi del percorso dipendenti dal contesto con GFS2 consultare [Sezione 4.12, «Mount Bind e Context-Dependent Path Names»](#).

## B.2. PROCEDURA CONVERSIONE DA GFS A GFS2

Usare la seguente procedura per convertire un file system GFS in un file system GFS2.

1. Su di un sistema Red Hat Enterprise Linux eseguire un backup del vostro file system GFS esistente.
2. Smontate il file system GFS da tutti i nodi presenti nel cluster.
3. Eseguite il comando `gfs_fsck` sul file system GFS, in modo da assicurarvi che il file system non sia corrotto.
4. Eseguire `gfs2_convert gfsfilesystem`. Il sistema mostrerà le domande di conferma insieme ai messaggi d'avviso prima delle conversione di `gfsfilesystem` in GFS2.
5. Aggiornamento al Red Hat Enterprise Linux 6.

Il seguente esempio converte un file system GFS sul dispositivo a blocchi `/dev/shell_vg/500g` in un file system GFS2.

```
[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.
```

## APPENDICE C. TRACEPOINT GFS2 E DEBUGFS GLOCK FILE

La seguente appendice descrive sia l'interfaccia `debugfs` di `glock` che i `tracepoint` GFS2. Essa è rivolta agli utenti esperti con conoscenze approfondite dei file system, interessati a sviluppare ulteriormente le loro competenze sul design del GFS2 e su metodi di debug delle problematiche specifiche.

### C.1. TIPI DI TRACEPOINT GFS2

Sono attualmente presenti tre tipi di `tracepoint` GFS2: `glock` (pronunciato "gee-lock"), `bmap` e `log`. I suddetti `tracepoint` vengono utilizzati per controllare l'esecuzione di un file system GFS2, e per ottenere le informazioni aggiuntive disponibili con le opzioni di debug supportate nelle versioni precedenti di Red Hat Enterprise Linux. I `tracepoint` sono particolarmente utili nella riproduzione degli errori, tipo una sospensione o problematiche relative alle prestazioni; i loro output sono disponibili durante le operazioni in questione. Con il GFS2 `glock` rappresenta il meccanismo di controllo della cache primario ed è un elemento importantissimo per la comprensione delle prestazioni degli elementi principali di GFS2. I `tracepoint` `bmap` (block map) possono essere utilizzati per il monitoraggio durante l'assegnazione dei blocchi e per la loro mappatura (ricerca dei blocchi già assegnati nell'albero dei metadati sul disco), eseguendo una ricerca di possibili errori relativi all'accesso. I `tracepoint` `log` registrano i dati scritti su o rilasciati dal journal, e forniscono informazioni utili in questo processo del GFS2.

I `tracepoint` sono stati ideati per essere il più generici possibile. Ciò significa che non sarà necessario modificare una API durante il ciclo di Red Hat Enterprise Linux 6. È importante tener presente che questa è una interfaccia di debug e non fa parte dell'insieme normale delle API di Red Hat Enterprise Linux 6, e per questo motivo Red Hat potrebbe rendere disponibili alcune modifiche per l'interfaccia dei `tracepoint` di GFS2.

I `tracepoint` rappresentano una funzione generica di Red Hat Enterprise Linux 6 e il loro utilizzo va oltre GFS2. In particolare essi vengono usati durante l'implementazione dell'infrastruttura di `blktrace`. I `tracepoint` di `blktrace` possono essere utilizzati in combinazione con i `tracepoint` di GFS2, ottenendo quindi informazioni più dettagliate sulle prestazioni del sistema. A causa del livello operativo dei `tracepoint`, essi possono generare un volume elevato di dati in periodi molto brevi. I `tracepoint` sono stati ideati in modo da utilizzare carichi di lavoro minimi, ma al tempo stesso essi possono avere qualche effetto. Il filtraggio degli eventi può aiutare a ridurre il volume dei dati, facilitando così l'uso di informazioni utili alla comprensione di situazioni particolari.

### C.2. TRACEPOINT

I `tracepoint` sono disponibili nella directory `/sys/kernel/debug/tracing/` se `debugfs` è stato montato nella posizione standard in `/sys/kernel/debug`. La directory `events` contiene tutti gli eventi rilevanti specificati e, in presenza del modulo `gfs2`, sarà presente una sottodirectory `gfs2` contenente ulteriori directory, una per ogni evento GFS2. I contenuti di `/sys/kernel/debug/tracing/events/gfs2` dovrebbero somigliare al seguente:

```
[root@chywoon gfs2]# ls
enable                gfs2_bmap            gfs2_glock_queue    gfs2_log_flush
filter                gfs2_demote_rq      gfs2_glock_state_change gfs2_pin
gfs2_block_alloc     gfs2_glock_put      gfs2_log_blocks     gfs2_promote
```

Per abilitare tutti i `tracepoint` GFS2 eseguire il seguente comando:

```
[root@chywoon gfs2]# echo -n 1
>/sys/kernel/debug/tracing/events/gfs2/enable
```

Per abilitare un tracepoint specifico è disponibile un file `enable` in ogni sottodirectory dell'evento. Situazione simile per il file `filter`, il quale può essere utilizzato per impostare un filtro per ogni evento o insieme di eventi. Il significato dei singoli eventi viene affrontato in modo più dettagliato qui di seguito.

L'output dei tracepoint è disponibile in formato ASCII o binario. L'appendice attualmente non riporta le informazioni relative all'interfaccia binaria. L'interfaccia ASCII è disponibile in due modi. Per elencare il contenuto corrente del ring buffer eseguire il seguente comando:

```
[root@chywoon gfs2]# cat /sys/kernel/debug/tracing/trace
```

L'interfaccia è utile se si utilizza un processo complesso per un determinato periodo di tempo e, dopo alcuni eventi, desiderate consultare le informazioni fino a quel punto acquisite nel buffer. Una interfaccia alternativa, `/sys/kernel/debug/tracing/trace_pipe`, può essere utilizzata quando è necessario consultare tutto l'output. Gli eventi vengono letti da questo file quando vengono generati; tramite questa interfaccia non sono disponibili le informazioni relative alla cronologia. Il formato dell'output è simile in entrambe le interfacce ed è descritto per ogni evento GFS2 in altre sezioni.

L'utilità `trace-cmd` è disponibile per la lettura dei dati relativi ai tracepoint. Per maggiori informazioni su questa utilità consultare il link in [Sezione C.10, «Riferimenti»](#). `trace-cmd` può essere utilizzata in modo simile a `strace`, per esempio per l'esecuzione di un comando durante la raccolta delle informazioni desiderate.

### C.3. GLOCK

Per capire meglio GFS2 è importante comprendere il concetto più significativo, e quello che lo rende diverso da altri file system, e cioè il concetto di glock. In termini di codice sorgente un glock rappresenta una struttura dati che raggruppa DLM e la memoria in cache in una sola macchina. Ogni glock ha un rapporto di 1:1 con un DLM lock singolo, e fornisce la memoria in cache per il relativo stato, così facendo le operazioni ripetitive eseguite da un singolo nodo del file system non invocano ripetutamente DLM, limitando così il traffico non necessario di rete. Sono presenti due categorie di glock, quelli che salvano i dati in cache e quelli che non eseguono questa operazione. Gli inode glock e i glock del gruppo di risorse eseguono il salvataggio in cache dei metadati, altri tipi di glock non eseguono questa operazione. L'inode glock può eseguire entrambe le operazioni di salvataggio dei dati e metadati in cache, e presenta la logica più complessa tra tutti i glock.

Tabella C.1. Modalità glock e DLM lock

Modalità glock	Modalità DLM lock	Note
UN	IV/NL	Unlocked (nessun DLM lock associato con glock o con NL lock in base al flag I)
SH	PR	Lock condiviso (lettura protetta)
EX	EX	Lock esclusivo
DF	CW	Posticipato (scrittura simultanea) isato per I/O Diretto e per la sospensione del file system

Glock è in grado di restare in memoria fino a quando non vengono sbloccati (per la richiesta di un altro

nodo o di una VM) e non sono presenti utenti locali. A quel punto essi verranno rimossi dalla tabella hash di glock e resi disponibili. Al momento della creazione di un glock il DLM lock non verrà immediatamente associato. DLM lock viene associato con glock previa prima richiesta ricevuta da DLM, e se la richiesta ha avuto successo, il flag "I" (iniziale) verrà impostato su glock. [Tabella C.4, «Flag di glock»](#) mostra i significati dei diversi flag di glock. Dopo aver eseguito l'associazione il DLM lock avrà sempre, come minimo, una modalità di NL (Null) fino a quando non si libererà glock. Un declassamento di DLM lock da NL a unlocked, è sempre l'ultima operazione nel ciclo di vita di glock.



#### NOTA

Questo aspetto particolare del comportamento di DLM lock è diverso da quello in Red Hat Enterprise Linux 5, il quale talvolta sbloccava i DLM lock assegnati ai glock, per questo motivo Red Hat Enterprise Linux 5 ha un meccanismo diverso per assicurare che i LVB (lock value block) siano preservati quando necessario. Il nuovo schema usato da Red Hat Enterprise Linux 6 è stato reso possibile grazie all'unione del modulo lock `lock_dlm` (da non confondere con DLM stesso) con GFS2.

Ogni glock può avere un certo numero di "holder" ad esso associati, ognuno dei quali rappresenta una richiesta di lock dai livelli superiori. Le invocazioni del sistema relative al GFS2, mettono in coda (o rimuovono) gli holder dal glock per proteggere la sezione critica del codice.

La glock state machine si basa su una workqueue. Per regioni relative alle prestazioni, è preferibile usare tasklet; tuttavia nell'implementazione attuale è necessario inviare I/O dal contesto attuale il quale ne proibisce il loro uso.



#### NOTA

Workqueue hanno i propri tracepoint i quali possono essere usati insieme con i tracepoint di GFS2 se desiderato

[Tabella C.2, «Tipi di dati e modalità di glock»](#) mostra lo stato nel quale vengono memorizzati in cache le modalità di glock, indicando anche eventuali errori. Questa operazione riguarda sia i lock del gruppo di risorse che quelli degli inode, anche se non vi è alcun componente dati per i lock del gruppo di risorse, ma solo metadati.

**Tabella C.2. Tipi di dati e modalità di glock**

Modalità glock	Dati della cache	Metadati della cache	Dati corrotti	Metadati corrotti
UN	No	No	No	No
SH	Si	Si	No	No
DF	No	Si	No	No
EX	Si	Si	Si	Si

## C.4. L'INTERFACCIA DEBUGFS DI GLOCK

L'interfaccia `debugfs` di glock permette di visualizzare lo stato interno dei glock e degli holder,

include anche, in alcuni casi, un sommario degli oggetti bloccati. Ogni riga del file può iniziare con G: con doppio spazio (si riferisce a glock stesso) oppure con una lettera diversa, e con spazio singolo, e si riferisce alle strutture associate con glock immediatamente sopra nel file (H: rappresenta un holder, I: un inode e R: un gruppo di risorse). Di seguito viene riportato un esempio del contenuto di questo file:

```
G:  s:SH n:5/75320 f:I t:SH d:EX/0 a:0 r:3
   H: s:SH f:EH e:0 p:4466 [postmark] gfs2_inode_lookup+0x14e/0x260 [gfs2]
G:  s:EX n:3/258028 f:yI t:EX d:EX/0 a:3 r:4
   H: s:EX f:tH e:0 p:4466 [postmark] gfs2_inplace_reserve_i+0x177/0x780
   [gfs2]
   R: n:258028 f:05 b:22256/22256 i:16800
G:  s:EX n:2/219916 f:yfI t:EX d:EX/0 a:0 r:3
   I: n:75661/219916 t:8 f:0x10 d:0x00000000 s:7522/7522
G:  s:SH n:5/127205 f:I t:SH d:EX/0 a:0 r:3
   H: s:SH f:EH e:0 p:4466 [postmark] gfs2_inode_lookup+0x14e/0x260 [gfs2]
G:  s:EX n:2/50382 f:yfI t:EX d:EX/0 a:0 r:2
G:  s:SH n:5/302519 f:I t:SH d:EX/0 a:0 r:3
   H: s:SH f:EH e:0 p:4466 [postmark] gfs2_inode_lookup+0x14e/0x260 [gfs2]
G:  s:SH n:5/313874 f:I t:SH d:EX/0 a:0 r:3
   H: s:SH f:EH e:0 p:4466 [postmark] gfs2_inode_lookup+0x14e/0x260 [gfs2]
G:  s:SH n:5/271916 f:I t:SH d:EX/0 a:0 r:3
   H: s:SH f:EH e:0 p:4466 [postmark] gfs2_inode_lookup+0x14e/0x260 [gfs2]
G:  s:SH n:5/312732 f:I t:SH d:EX/0 a:0 r:3
   H: s:SH f:EH e:0 p:4466 [postmark] gfs2_inode_lookup+0x14e/0x260 [gfs2]
```

Il seguente esempio riporta alcuni estratti (da un file di circa 18MB) generati dal comando `cat /sys/kernel/debug/gfs2/unity:myfs/glocks >my.lock` durante l'esecuzione di un postmark benchmark su un file system GFS2 di un nodo. Nell'figura sono stati selezionati i glock in modo da mostrare alcune delle funzioni più interessanti dei glock dump.

Gli stati relativi al glock possono essere EX (exclusive), DF (deferred), SH (shared) o UN (unlocked). I suddetti stati corrispondono direttamente con le modalità DLM lock ad eccezione di UN, il quale può rappresentare uno stato DLM null lock oppure indicare che GFS2 non presenta alcun DLM lock (a seconda del flag I come precedentemente affrontato). Il campo s: indica lo stato corrente di lock, lo stesso campo nell'holder indica la modalità richiesta. Se si conferisce un lock, l'holder presenterà la lettera H nei propri flag (f: field). In caso contrario presenterà la lettera W.

Il campo n: (numero) indica il numero associato con ogni oggetto. Per glock, rappresenta il numero del tipo seguito dal numero di glock come nell'esempio riportato, il primo glock è n:5/75320; cioè un glock `iopen` relativo all'inode 75320. In presenza di inode e glock `iopen`, il numero di glock è sempre identico al numero del blocco a disco dell'inode.



## NOTA

I numeri di glock (campo n:) nel file glock di debugfs sono esadecimali, mentre gli output di tracepoint li rappresenta con valori decimali. Questo viene fatto per motivi cronologici; i numeri di glock sono sempre stati scritti in esadecimali, ma i valori decimali sono stati scelti per i tracepoint in modo da facilitare il processo di confronto con gli altri output (per esempio di `blktrace`) e con output di `stat(1)`.

L'elenco completo di tutti i flag, sia per l'holder che per glock, sono disponibili in [Tabella C.4, «Flag di glock»](#) e [Tabella C.5, «Flag holder di glock»](#). Il contenuto dei blocchi per il valore del blocco non è attualmente disponibile tramite l'interfaccia `debugfs` di glock.

[Tabella C.3, «Tipi di glock»](#) mostra il significato dei diversi tipi di glock.

Tabella C.3. Tipi di glock

Numero tipo	Tipo di lock	Uso
1	trans	Lock di transazione
2	inode	Dati e metadati di Inode
3	rgrp	Metadati del gruppo di risorse
4	meta	Il superblocco
5	iopen	Rilevamento ultimo nodo che ha utilizzato inode
6	flock	<b>flock(2)</b> syscall
8	quota	Operazioni quota
9	journal	Journal mutex

Uno dei flag più importanti di glock è **l** (locked). Questo flag regola l'accesso allo stato di glock per eseguire la modifica dello stato stesso. Esso viene impostato quando la macchina degli stati è in procinto di inviare una richiesta remota di lock tramite DLM, viene annullato solo quando l'operazione è stata completamente eseguita. Talvolta ciò può implicare l'invio di più di una richiesta di lock con conseguenti processi di annullamento.

Tabella C.4, «Flag di glock» mostra il significato dei diversi flag.

Tabella C.4. Flag di glock

Flag	Nome	Significato
d	Pending demote	Una richiesta di declassamento (remota) rinviata
D	Demote (declassamento)	Una richiesta di declassamento (locale o remota)
f	Log flush	È necessario eseguire il commit del log prima di rilasciare questo glock
F	Frozen	Le risposte dai nodi remoti sono ignorate - il ripristino è in corso.
i	Invalidate in progress	La rimozione in questo glock della convalida delle pagine è in corso
l	Initial	Impostato quando il DLM lock è associato con questo glock
l	Locked	Il glock è in procinto di cambiare stato

Flag	Nome	Significato
L	LRU	Impostato quando glock è nell'elenco LRU
o	Oggetto	Impostato quando glock è associato con un oggetto (cioè un inode per il tipo 2 di glock, e un gruppo di risorse per il tipo 3 di glock)
p	Demote in progress	Glock è in procinto a rispondere ad una richiesta di declassamento
q	In coda	Impostato quando un holder è in coda per un glock ed è rimosso quando un glock è occupato, quando altri holder non sono disponibili. Usato come parte dell'algoritmo per il calcolo dell'intervallo minimo per un glock.
r	Reply pending	La risposta ricevuta da un nodo remoto è in attesa di processazione
y	Dirty	È necessario azzerare i dati sul disco prima di rilasciare questo glock

Alla ricezione di una chiamata remota da un nodo che desidera ottenere un lock in una modalità in conflitto con quella presente sul nodo locale, verrà impostato uno dei seguenti flag, D (demote) o d (demote pending). Per evitare condizioni di "starvation" in presenza di una contesa per un lock particolare, ogni lock riceve un intervallo. In presenza di un nodo al quale non è stato conferito un lock per la durata prevista dell'intervallo, esso avrà diritto al lock fino a quando l'intervallo scade.

Se l'intervallo è scaduto, allora verrà impostato il flag D (demote) e registrato lo stato necessario. In tal caso in assenza di lock nella coda degli holder, il lock verrà declassato. Se l'intervallo non è ancora scaduto verrà impostato il flag d (demote pending). Questa operazione indica alla macchina degli stati di eliminare d (demote pending) e impostare D (demote) dopo la scadenza dell' intervallo.

Il flag l (initial) viene impostato quando è stato assegnato al glock un DLM lock. Ciò si verifica al primo utilizzo di glock ed il flag rimarrà impostato fino alla disponibilità di glock (DLM lock è sbloccato "unlocked").

## C.5. HOLDER DI GLOCK

Tabella C.5, «Flag holder di glock» mostra i significati dei diversi flag holder di glock.

Tabella C.5. Flag holder di glock

Flag	Nome	Significato
a	Async	Non attendere il risultato di glock (il risultato verrà richiesto più avanti)
A	Qualsiasi	Qualsiasi modalità di blocco compatibile è accettabile



Flag	Nome	Significato
c	No cache	Quando unlocked, declassa immediatamente DLM lock
e	No expire	Ignora le richieste di cancellazione successive di lock
E	Exact	Deve avere una modalità lock esatta
F	First	Imposta quando un holder è il primo a essere conferito per questo lock
H	Holder	Indica che il lock richiesto è stato conferito
p	Priorità	Metti l'holder in cima alla coda
t	Prova	Un lock "try"
T	Try 1CB	Un lock "try" che invia una callback
M	Wait	Imposta mentre in attesa del completamento della richiesta

I flag più importanti sono H (holder) e W (wait) come precedentemente indicato, poichè essi vengono conferiti rispettivamente in base alle richieste di lock conferite e richieste messe in coda. L'ordine degli holder nell'elenco è importante. Se si conferiscono alcuni holder, essi si troveranno sempre nei primi posti della coda, seguiti da qualsiasi altro holder.

In assenza di holder, il primo presente nell'elenco sarà colui che avvierà la modifica alla fase successiva. Poichè le richieste di declassamento hanno sempre una priorità più alta rispetto alle richieste del file system, questo processo potrebbe non sempre risultare in una modifica allo stato richiesto.

Il sottosistema di glock supporta due tipi di lock "try". Entrambi sono molto utili poichè permettono di modificare l'ordine normale dei lock (con operazioni idonee di back-off e retry), e possono essere usate per non utilizzare le risorse in uso da altri nodi. Il lock normale t (try) è semplicemente un lock "try" e non esegue alcuna azione speciale. T (try 1CB) invece, è simile al lock t ad eccezione del fatto che DLM invia una chiamata singola agli holder dei lock correnti non compatibili. Un tipo di utilizzo del lock T (try 1CB) è con `ioopen`, utilizzati per una mediazione tra i nodi quando il conteggio `i_nlink` di un inode è zero, e per determinare i nodi responsabili per la modifica dell'assegnazione dell'inode. Il glock `ioopen` normalmente presenta uno stato di shared (condiviso), ma se il conteggio `i_nlink` è zero e `->delete_inode()` viene invocato, esso richiederà un lock esclusivo con l'insieme T (try 1CB). Il processo di modifica dell'assegnazione dell'inode continuerà se viene conferito un lock. In caso contrario, i nodi che impedivano il conferimento di un lock creeranno un glock con il flag D (demote), che verrà revisionato in `->drop_inode()`, per assicurare che la modifica dell'assegnazione non venga dimenticata.

Ciò significa che agli inode con un conteggio di zero link, ma che risultano ancora aperti, verrà modificata l'allocazione dal nodo sul quale si è verificato l'ultimo `close()`. Altresì, quando il conteggio del link dell'inode viene diminuito fino a zero, l'inode sarà contrassegnato con uno stato speciale di zero link, ma ancora in uso nel bitmap del gruppo di risorse. Ciò funziona come l'elenco di orfani del file system3 ext3, e cioè permette a qualsiasi lettore del bitmap di sapere se è presente spazio disponibile in grado di essere recuperato.

## C.6. TRACEPOINT DI GLOCK

I tracepoint sono in grado di confermare il funzionamento corretto del controllo della cache combinandoli con l'output di `blktrace`, e utilizzando la conoscenza della disposizione del disco. È quindi possibile controllare che qualsiasi I/O dato sia stato emesso e completato con un lock corretto, al tempo stesso cerca di rilevare la presenza di eventuali corse critiche.

Il tracepoint `gfs2_glock_state_change` è quello più importante poichè lo si utilizza per il controllo di ogni modifica sullo stato di glock, dalla creazione iniziale fino al declassamento finale che termina con `gfs2_glock_put`, e NL finale in transizione su `unlocked`. Il flag di glock I (`locked`) viene sempre impostato prima della modifica dello stato, e non verrà rimosso se non nella parte finale. Durante la modifica dello stato (flag del detentore di glock H) non verrà mai conferito un holder. In presenza di holder nella coda, essi avranno sempre uno stato W (`waiting`). Dopo la modifica dello stato sarà possibile conferire un holder, questa è l'operazione finale prima della rimozione del flag I.

Il tracepoint `gfs2_demote_rq` controlla le richieste di declassamento, sia locali che remote. In presenza di memoria sufficiente sul nodo, le richieste di declassamento locali sono visualizzate raramente, e molto spesso vengono create a causa di operazioni di smontaggio o di reclamo della memoria. Il numero delle richieste di declassamento remote misura il livello di contesa tra i nodi per un gruppo di risorse o un inode particolare.

Al conferimento di un lock per un holder verrà invocato il comando `gfs2_promote`, ciò si verifica durante la fase finale di una modifica dello stato, o a causa di una richiesta di lock che può essere conferito immediatamente a causa dello stato di glock, il quale a sua volta memorizza in cache il lock con una modalità idonea. Dopo questa operazione verrà impostato il flag f (`first`). I gruppi di risorse utilizzano questo processo.

## C.7. TRACEPOINT BMAP

La mappatura del blocco è un compito importante in qualsiasi file system. GFS2 utilizza un sistema basato sul bitmap con due bit per blocco. In questo sottosistema lo scopo principale dei tracepoint è quello di monitorare il tempo necessario per assegnare e mappare i blocchi.

Per ogni operazione `bmap` il tracepoint `gfs2_bmap` viene invocato due volte: la prima volta all'inizio per visualizzare la richiesta `bmap`, la seconda per visualizzare il risultato. Ciò facilita la corrispondenza delle richieste e dei risultati, e la misurazione del tempo necessario per mappare i blocchi nelle diverse parti del file system, in offset diversi del file o anche in file diversi. È altresì possibile visualizzare le dimensioni medie ritornate rispetto a quelle richieste.

Per controllare i blocchi assegnati `gfs2_block_alloc` viene invocato non solo durante le operazioni di assegnazione, ma anche durante il processo di liberazione dei blocchi stessi. Poichè tutte le allocazioni si riferiscono all'inode per le quali è inteso il blocco, ciò può essere utilizzato per controllare quali blocchi fisici appartengono ai file di un file system attivo. Questa operazione è particolarmente utile se usata insieme con `blktrace`, il quale è in grado di mostrare i percorsi I/O con problematiche relative che potrebbero far riferimento agli inode rilevanti, utilizzando la mappatura disponibile tramite questo tracepoint.

## C.8. TRACEPOINT DI LOG

I tracepoint presenti in questo sottosistema controllano i blocchi aggiunti e rimossi dal journal (`gfs2_pin`), e il tempo necessario per salvare le operazioni sul log ( `gfs2_log_flush`). Questa operazione può essere particolarmente utile durante la risoluzioni di problematiche relative alle prestazioni del journal.

Il tracepoint `gfs2_log_blocks` controlla i blocchi riservati nel log, i quali possono assistere durante la visualizzazione se il log è troppo piccolo rispetto al carico di lavoro.

Il tracepoint `gfs2_all_flush` (Red Hat Enterprise Linux 6.2 e versioni più recenti) è simile al `gfs2_log_flush` poichè esso raccoglie le informazioni sull'inizio e la fine delle operazioni azzeramento dell'elenco ALL. L'elenco contiene i buffer riportati nel log, non ancora riscritti, e viene periodicamente azzerato per rendere disponibile più spazio utilizzabile dai file system, o per la richiesta di sync o fsync di un processo.

## C.9. STATISTICHE DI GLOCK

GFS2 mantiene le statistiche necessarie per il controllo di attività all'interno di un file system. Questa operazione aiuta l'utente a individuare potenziali problematiche delle prestazioni.

GFS2 mantiene due tipi di contatori:

- `dcount` è in grado di contare il numero di operazioni DLM richieste, e mostra la quantità di dati necessari per i calcoli di media/varianza.
- `qcount` esegue un conteggio del numero di operazioni `syscall` richieste. Generalmente `qcount` sarà uguale o maggiore di `dcount`.

GFS2 altresì è in grado di mantenere tre coppie di media/varianza. Queste coppie rappresentano stime esponenziali regolari, e l'algoritmo è quello usato per calcolare i tempi di round trip in un codice di rete. Le coppie di media e varianza presenti in GFS2 sono unità intere di nanosecondi.

- `srtt/srttvar`: Tempo medio di round trip per operazioni di non-blocco
- `srttb/srttvarb`: Tempo medio di round trip per operazioni di blocco
- `irtt/irttvar`: Tempo Inter-request (per esempio, tempo trascorso tra le richieste DLM)

Una richiesta di non-blocco è una richiesta che verrà completata subito senza considerare lo stato del DLM lock. Ciò significa in ogni richiesta quando (a) lo stato corrente di lock è esclusivo (b) lo stato richiesto è null o unlocked oppure (c) quando viene impostato il flag "try lock". Una richiesta di blocco si occupa di tutte le altre richieste in questione.

Tempi più lunghi sono ottimali per IRTT, mentre tempi più piccoli sono migliori per RTT.

Le statistiche vengono archiviate in due file `sysfs`:

- Il file `glstats`. Questo file è simile a `glocks` ma contiene le informazioni sulle statistiche, con un glock per riga. I dati vengono inizializzati "per cpu" per il tipo di glock per il quale è stato creato (a parte i contatori, che sono azzerati). Questo file può essere molto grande.
- Il file `lkstats`. Al suo interno sono disponibili le informazioni "per cpu" per ogni tipo di glock. Ogni riga contiene un tipo d'informazione nella quale ogni colonna rappresenta una cpu core. Sono presenti otto righe per ogni tipo di glock, con ogni tipologia che si sussegue l'un l'altra.

## C.10. RIFERIMENTI

Per maggiori informazioni sui tracepoint e sul file `glocks` di GFS2, consultare le seguenti risorse:

- Questa appendice è stata parzialmente adattata da una presentazione di Steve Whitehouse alla Linux Symposium 2009, disponibile su <http://git.kernel.org/>

[p=linux/kernel/git/torvalds/linux-2.6.git;a=blob;f=Documentation/filesystems/gfs2-glocks.txt;h=0494f78d87e40c225eb1dc1a1489acd891210761;hb=HEAD](http://git.kernel.org/?p=linux/kernel/git/torvalds/linux-2.6.git;a=blob;f=Documentation/filesystems/gfs2-glocks.txt;h=0494f78d87e40c225eb1dc1a1489acd891210761;hb=HEAD).

- Per informazioni sulle regole per il locking interne consultare <http://git.kernel.org/?p=linux/kernel/git/torvalds/linux-2.6.git;a=blob;f=Documentation/filesystems/gfs2-glocks.txt;h=0494f78d87e40c225eb1dc1a1489acd891210761;hb=HEAD>.
- Per informazioni sugli eventi di tracciamento consultare <http://git.kernel.org/?p=linux/kernel/git/torvalds/linux-2.6.git;a=blob;f=Documentation/trace/events.txt;h=09bd8e9029892e4e1d48078de4d076e24e>
- Per informazioni su `trace-cmd` consultare <http://lwn.net/Articles/341902/>.

## APPENDICE D. DIARIO DELLE REVISIONI

<b>Revisione 7.1-3.2</b> traduzione completata	<b>Wed Feb 4 2015</b>	<b>Francesco Valente</b>
<b>Revisione 7.1-3.1</b> Translation files synchronised with XML sources 7.1-3	<b>Wed Feb 4 2015</b>	<b>Francesco Valente</b>
<b>Revisione 7.1-3</b> Aggiornato per l'implementazione di sort_order nella pagina di apertura di RHEL 6.	<b>Tue Dec 16 2014</b>	<b>Steven Levine</b>
<b>Revisione 7.0-9</b> Versione per la release 6.6 GA	<b>Wed Oct 8 2014</b>	<b>Steven Levine</b>
<b>Revisione 7.0-8</b> Versione per la release 6.6 Beta	<b>Thu Aug 7 2014</b>	<b>Steven Levine</b>
<b>Revisione 7.0-4</b> Risolve #1102591 Aggiunge la procedura per la configurazione del GFS2 in un cluster Pacemaker	<b>Thu Jul 17 2014</b>	<b>Steven Levine</b>
<b>Revisione 7.0-3</b> Risolve #1035119 Aggiornamento tabella dei flag di Glock e aggiunta di una sezione sulle statistiche	<b>Wed Jul 16 2014</b>	<b>Steven Levine</b>
<b>Revisione 7.0-1</b> Prima bozza per la release 6.6	<b>Thu Jun 5 2014</b>	<b>Steven Levine</b>
<b>Revisione 6.0-6</b> Versione per la release 6.5 GA	<b>Wed Nov 13 2013</b>	<b>Steven Levine</b>
<b>Revisione 6.0-5</b> Versione per la release 6.5 Beta	<b>Fri Sep 27 2013</b>	<b>Steven Levine</b>
<b>Revisione 6.0-3</b> Risolve #960841 Chiarisce la mancanza di supporto per SELinux con i file system GFS2.	<b>Fri Sep 27 2013</b>	<b>Steven Levine</b>
<b>Revisione 6.0-1</b> Aggiunta una nota su Samba e GFS2	<b>Fri Sep 06 2013</b>	<b>Steven Levine</b>
<b>Revisione 5.0-7</b> Versione per la release 6.4 GA	<b>Mon Feb 18 2013</b>	<b>Steven Levine</b>
<b>Revisione 5.0-5</b> Versione per la release 6.4 Beta	<b>Mon Nov 26 2012</b>	<b>Steven Levine</b>
<b>Revisione 5.0-4</b> Risolve #860324 Aggiorna il capitolo sulle considerazioni operative e configurazione di GFS2 con piccoli chiarimenti.  Risolve #807057 Aggiunge alcuni consigli relativi alla consultazione di un rappresentante autorizzato di Red Hat per la verifica della configurazione, prima di una implementazione.	<b>Tue Nov 13 2012</b>	<b>Steven Levine</b>
<b>Revisione 5.0-1</b>	<b>Mon Oct 15 2012</b>	<b>Steven Levine</b>

Aggiornato il capitolo sulle considerazioni operative.

**Revisione 4.0-2**                                      **Thu Mar 28 2012**                                      **Steven Levine**  
Versione per la release 6.3 GA

**Revisione 4.0-1**                                      **Thu Mar 28 2012**                                      **Steven Levine**  
Resolve: #782482, #663944  
Aggiunge un nuovo capitolo sulla configurazione del GFS2 e sulle considerazioni relative alla funzione.

Resolve: #757742  
Chiarifica la necessità di utilizzare GFS2 con CLVM.

Resolve: #786621  
Corregge piccoli errori di battitura.

**Revisione 3.0-2**                                      **Thu Dec 1 2011**                                      **Steven Levine**  
Release per il GA di Red Hat Enterprise Linux 6.2

**Revisione 3.0-1**                                      **Mon Sep 19 2011**                                      **Steven Levine**  
Versione iniziale della release Red Hat Enterprise Linux 6.2 Beta

Resolve: #704179  
Documenta il supporto per il comando **tunegfs2**.

Resolve: #712390  
Aggiunge una nuova appendice per i tracepoint di GFS2.

Resolve: #705961  
Risolve piccoli errori di battitura.

**Revisione 2.0-1**                                      **Thu May 19 2011**                                      **Steven Levine**  
Release iniziale per Red Hat Enterprise Linux 6.1

Resolve: #549838  
Documenta il supporto per le funzioni standard del quota di Linux in Red Hat Enterprise Linux 6.1.

Resolve: #608750  
Chiarisce la descrizione della funzione `withdraw` del GFS2.

Resolve: #660364  
Corregge le informazioni relative alla dimensione massima del file system GFS2.

Resolve: #687874  
Aggiunge un nuovo capitolo sul troubleshooting del GFS2.

Resolve: #664848  
Aggiunge le informazioni sulla ricerca dei Context-Dependent Path Name prima della conversione da GFS a GFS2.

**Revisione 1.0-1**                                      **Wed Nov 15 2010**                                      **Steven Levine**  
Release iniziale per Red Hat Enterprise Linux 6

---

# INDICE ANALITICO

## A

atime, configurazione aggiornamenti, [Configurazione degli aggiornamenti atime](#)  
montaggio con noatime , [Montaggio con noatime](#)  
montaggio con relatime , [Montaggio con relatime](#)

## B

blocco dei nodi, [Blocco dei nodi GFS2](#)

## C

chi è rivolto, [A chi è rivolto](#)

comando fsck.gfs2, [Come ripristinare un file system](#)

comando gfs2\_grow, [Come espandere un file system](#)

comando gfs2\_jadd, [Come aggiungere i journal ad un file system](#)

comando gfs2\_quota, [Gestione quota del GFS2 con il comando gfs2\\_quota](#)

comando mkfs, [Creazione di un file system](#)

comando mount, [Montaggio di un file system](#)

comando quotacheck

controllo accuratezza del quota con, [Mantenimento di un quota accurato](#)

comando umount, [Come smontare un file system](#)

come aggiungere i journal ad un file system, [Come aggiungere i journal ad un file system](#)

come espandere un file system, [Come espandere un file system](#)

come riparare un file system, [Come ripristinare un file system](#)

come smontare un file system, [Come smontare un file system](#), [Considerazioni particolari durante il montaggio dei file system GFS2](#)

commento

informazioni di contatto per questo manuale, [Abbiamo bisogno dei vostri commenti!](#)

compiti iniziali

impostazione, iniziale, [Compiti iniziali per l'impostazione](#)

compiti prerequisiti

configurazione, iniziale, [Prerequisiti](#)

configurazione iniziale

compiti prerequisiti, [Prerequisiti](#)

configurazione, iniziale, [Per iniziare](#)

configurazione, prima, [Prima d'impostare il GFS2](#)

Considerazioni sulla configurazione, [Considerazioni operative e configurazione del GFS2](#)

Context-Dependent Path Names (CDPNs)

Conversione da GFS a GFS2, [Conversione dei Context-Dependent Path Names](#)

creazione di un file system, [Creazione di un file system](#)

## D

data journaling, [Data Journaling](#)

debugfs, [Tracepoint GFS2 e debugfs glock File](#)

Dimensione massima del file system GFS2, [Panoramica sul GFS2](#)

dimensione massima, file system GFS2, [Panoramica sul GFS2](#)

disk quota

assegnazione per gruppo, [Assegnazione dei quota per un gruppo](#)

assegnazione per utente, [Assegnazione dei quota per utente](#)

come abilitare

creazione file del quota, [Creazione dei file del Quota Database](#)

esecuzione di quotacheck , [Creazione dei file del Quota Database](#)

gestione di, [Gestione del Disk Quota](#)

comando quotacheck, utilizzo per il controllo, [Mantenimento di un quota accurato](#)

riporto, [Gestione del Disk Quota](#)

hard limit, [Assegnazione dei quota per utente](#)

risorse aggiuntive, [Riferimenti](#)

soft limit, [Assegnazione dei quota per utente](#)

disk quotas

come abilitare, [Configurazione dei disk quota](#)

## F

file debugfs, [Risoluzione problematiche relative alle prestazioni di GFS2 con il GFS2 Lock Dump](#)

file system

atime, configurazione aggiornamenti, [Configurazione degli aggiornamenti atime](#)

montaggio con noatime , [Montaggio con noatime](#)

montaggio con relatime , [Montaggio con relatime](#)

come aggiungere i journal, [Come aggiungere i journal ad un file system](#)

context-dependent path names (CDPN), [Mount Bind e Context-Dependent Path Names](#)

creazione, [Creazione di un file system](#)

data journaling, [Data Journaling](#)

espandere, [Come espandere un file system](#)

gestione dei quota

sincronizzazione dei quota, [Sincronizzazione dei Quota con il comando quotasync](#)

gestione del quota, [Gestione quota del GFS2 con il comando gfs2\\_quota](#)

come abilitare la contabilità dei quota, [Come abilitare la contabilità dei quota](#)

come abilitare/disabilitare il quota enforcement, [Come abilitare/disabilitare il Quota Enforcement](#)



impostazione dei quota, [Impostazione dei quota con il comando gfs2\\_quota](#)  
 sincronizzazione dei quota, [Sincronizzazione del quota con il comando gfs2\\_quota](#)  
 visualizzazione dei limiti del quota, [Come visualizzare l'utilizzo ed i limiti del quota con il comando gfs2\\_quota](#)

gestione quota, [Gestione quota del GFS2](#) , [Impostazione dei quota in modalità Enforcement o Accounting](#).

montaggio, [Montaggio di un file system](#), [Considerazioni particolari durante il montaggio dei file system GFS2](#)

mount bind, [Mount Bind e Context-Dependent Path Names](#)

ordine di montaggio, [Mount bind e ordine di montaggio di un file system](#)

riparazione, [Come ripristinare un file system](#)

smontare, [Come smontare un file system](#), [Considerazioni particolari durante il montaggio dei file system GFS2](#)

sospensione attività, [Sospensione di una attività su di un file system](#)

flag glock, [Risoluzione problematiche relative alle prestazioni di GFS2 con il GFS2 Lock Dump](#) , [L'interfaccia debugfs di glock](#)

flag holder di glock, [Risoluzione problematiche relative alle prestazioni di GFS2 con il GFS2 Lock Dump](#), [Holder di glock](#)

funzione withdraw, GFS2, [Funzione Withdraw di GFS2](#)

funzioni, nuove e modificate, [Funzioni nuove e modificate](#)

## G

gestioen del quota

visualizzazione dei limiti del quota, [Come visualizzare l'utilizzo ed i limiti del quota con il comando gfs2\\_quota](#)

gestione dei quota

sincronizzazione dei quota, [Sincronizzazione dei Quota con il comando quotasync](#)

gestione del GFS2, [Gestione del GFS2](#)

gestione del quota, [Gestione quota del GFS2 con il comando gfs2\\_quota](#)

come abilitare la contabilità dei quota, [Come abilitare la contabilità dei quota](#)

come abilitare/disabilitare il quota enforcement, [Come abilitare/disabilitare il Quota Enforcement](#)

impostazione dei quota, [Impostazione dei quota con il comando gfs2\\_quota](#)

sincronizzazione dei quota, [Sincronizzazione del quota con il comando gfs2\\_quota](#)

gestione quota, [Gestione quota del GFS2](#) , [Impostazione dei quota in modalità Enforcement o Accounting](#).

GFS2

atime, configurazione aggiornamenti, [Configurazione degli aggiornamenti atime](#)

montaggio con noatime , [Montaggio con noatime](#)

montaggio con relatime , [Montaggio con relatime](#)

Considerazioni sulla configurazione, [Considerazioni operative e configurazione del GFS2](#)

Funzionamento, [Considerazioni operative e configurazione del GFS2](#)

funzione withdraw, [Funzione Withdraw di GFS2](#)

gestione, [Gestione del GFS2](#)

gestione dei quota

sincronizzazione dei quota, [Sincronizzazione dei Quota con il comando quotasync](#)

gestione del quota, [Gestione quota del GFS2 con il comando gfs2\\_quota](#)

come abilitare la contabilità dei quota, [Come abilitare la contabilità dei quota](#)

come abilitare/disabilitare il quota enforcement, [Come abilitare/disabilitare il Quota Enforcement](#)

impostazione dei quota, [Impostazione dei quota con il comando gfs2\\_quota](#)

sincronizzazione dei quota, [Sincronizzazione del quota con il comando gfs2\\_quota](#)

visualizzazione dei limiti del quotas, [Come visualizzare l'utilizzo ed i limiti del quota con il comando gfs2\\_quota](#)

gestione quota, [Gestione quota del GFS2](#) , [Impostazione dei quota in modalità Enforcement o Accounting.](#)

glock, [Tracepoint GFS2 e debugfs glock File](#)

I

impostazione, iniziale

compiti iniziali, [Compiti iniziali per l'impostazione](#)

introduzione, [Introduzione](#)

a chi è rivolto, [A chi è rivolto](#)

M

montaggio di un file system, [Montaggio di un file system](#), [Considerazioni particolari durante il montaggio dei file system GFS2](#)

mount bind, [Mount Bind e Context-Dependent Path Names](#)

ordine di montaggio, [Mount bind e ordine di montaggio di un file system](#)

O

opzione di montaggio quota= , [Impostazione dei quota con il comando gfs2\\_quota](#)

opzione di mount acl, [Montaggio di un file system](#)

opzioni specifiche a GFS2 per la tabella d'espansione dei file system, [Utilizzo completo](#)

opzioni specifiche al GFS2 per l'aggiunta della tabella dei journal, [Utilizzo completo](#)

P

panoramica, [Panoramica sul GFS2](#)

configurazione, prima, [Prima d'impostare il GFS2](#)

funzioni, nuove e modificate, [Funzioni nuove e modificate](#)

parametro regolabile quota\_quantum, [Sincronizzazione dei Quota con il comando quotasync](#) ,  
[Sincronizzazione del quota con il comando gfs2\\_quota](#)

path names, context-dependent (CDPN), [Mount Bind e Context-Dependent Path Names](#)

Posix locking, [Problematiche con il Posix Locking](#)

prefazione (vedi introduzione)

## Q

quotacheck , [Creazione dei file del Quota Database](#)

## R

regolazione delle prestazioni, [Regolazione delle prestazioni con GFS2](#)

regolazione, prestazioni, [Regolazione delle prestazioni con GFS2](#)

## S

smontare, sospensione del sistema, [Considerazioni particolari durante il montaggio dei file system GFS2](#)

sospensione del sistema durante lo smontaggio, [Considerazioni particolari durante il montaggio dei file system GFS2](#)

sospensione dell'attività su di un file system, [Sospensione di una attività su di un file system](#)

## T

tabella di montaggio, [Utilizzo completo](#)

tabella opzioni del comando mkfs.gfs2, [Opzioni complete](#)

tabelle

- opzioni del comando mkfs.gfs2, [Opzioni complete](#)

- opzioni di montaggio, [Utilizzo completo](#)

- opzioni specifiche a GFS2 per l'espansione dei file system, [Utilizzo completo](#)

- opzioni specifiche al GFS2 per l'aggiunta dei journal, [Utilizzo completo](#)

Tipi di glock, [Risoluzione problematiche relative alle prestazioni di GFS2 con il GFS2 Lock Dump](#) ,  
[L'interfaccia debugfs di glock](#)

tracepoint, [Tracepoint GFS2 e debugfs glock File](#)