



# Red Hat Enterprise Linux 6

## Global File System 2

Red Hat Global File System 2

Edição 7

Last Updated: 2017-10-13



# Red Hat Enterprise Linux 6 Global File System 2

---

Red Hat Global File System 2

Edição 7

## Nota Legal

Copyright © 2014 Red Hat, Inc. and others.

This document is licensed by Red Hat under the [Creative Commons Attribution-ShareAlike 3.0 Unported License](https://creativecommons.org/licenses/by-sa/3.0/). 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.

## Resumo

Este livro fornece informações sobre configuração e manutenção do Red Hat GFS2 (Red Hat Global File System 2) para o Red Hat Enterprise Linux 6.

# Índice

<b>INTRODUÇÃO</b> .....	<b>5</b>
1. PÚBLICO ALVO	5
2. DOCUMENTAÇÃO RELACIONADA	5
3. PRECISAMOS DE SEU FEEDBACK	6
<b>CAPÍTULO 1. VISÃO GERAL DO GFS2</b> .....	<b>7</b>
1.1. RECURSOS NOVOS E ALTERADOS	8
1.1.1. Recursos Novos e Alterados para o Red Hat Enterprise Linux 6.0	8
1.1.2. Recursos Novos e Modificados para o Red Hat Enterprise Linux 6.1	8
1.1.3. Recursos Novos e Modificados para o Red Hat Enterprise Linux 6.2.	9
1.1.4. Recursos Novos e Alterados para o Red Hat Enterprise Linux 6.3	9
1.1.5. Recursos Novos e Modificados para o Red Hat Enterprise Linux 6.4	9
1.1.6. Recursos Novos e Modificados para o Red Hat Enterprise Linux 6.6	10
1.2. ANTES DE CONFIGURAR O GFS2	10
1.3. INSTALANDO O GFS2	10
1.4. DIFERENÇAS ENTRE O GFS E GFS2	11
1.4.1. Nomes de Comando do GFS2	11
1.4.2. Diferenças Adicionais Entre GFS e GFS2	12
1.4.2.1. Nomes de Caminhos Dependentes de Contexto	12
1.4.2.2. Módulo gfs2.ko	13
1.4.2.3. Ativando Imposição de Cota no GFS2	13
1.4.2.4. Diário de Dados	13
1.4.2.5. Adicionando Diários de forma Dinâmica	13
1.4.2.6. Removido parâmetro atime_quantum	13
1.4.2.7. A opção 'data=' do comando mount	13
1.4.2.8. O comando gfs2_tool	14
1.4.2.9. O comando gfs2_edit	14
1.4.3. Aprimoramentos de Desempenho do GFS2	14
<b>CAPÍTULO 2. CONFIGURAÇÃO DO GFS2 E CONSIDERAÇÕES OPERACIONAIS</b> .....	<b>16</b>
2.1. FORMATANDO CONSIDERAÇÕES	16
2.1.1. Tamanho do Sistema de Arquivo: Quanto Menor Melhor	16
2.1.2. Tamanho do Bloco: Prefere-se (4K) Blocos Padrão	16
2.1.3. Números de Diários: Um para cada Nó que Monta	17
2.1.4. Tamanho do Diário: Padrão (128 MB) É Geralmente o mais adequado	17
2.1.5. Tamanho e Número de Grupos de Recursos.	17
2.2. FRAGMENTAÇÃO DE SISTEMA DE ARQUIVO	18
2.3. PROBLEMAS DE ALOCAÇÃO DE BLOCOS	18
2.3.1. Deixe Espaço Livre no Sistema de Arquivo.	18
2.3.2. Aloque Cada Nó a Seus Próprios Arquivos, Se Possível	19
2.3.3. Pré-aloque, Se possível	19
2.4. CONSIDERAÇÕES DE CLUSTER	19
2.5. CONSIDERAÇÕES DE USO	19
2.5.1. Opções de Montagem: noatime e nodiratime	20
2.5.2. Opções de Ajuste do DLM: Aumentar Tamanhos da Tabela do DLM	20
2.5.3. Opções de Ajuste do VFS: Pesquisa e Experimentos	20
2.5.4. SELinux: Evite SELinux no GFS2	20
2.5.5. Configurando o NFS sob o GFS2.	21
2.5.6. Serviço de Arquivo do Samba (SMB ou Windows) sob GFS2	22
2.6. BACKUPS DO SISTEMA DE ARQUIVO	22
2.7. CONSIDERAÇÕES DO HARDWARE	22
2.8. PROBLEMAS DE DESEMPENHO: VERIFIQUE O RED HAT CUSTOMER PORTAL	23

2.9. BLOQUEIO DE NÓ GFS2	23
2.9.1. Problemas com o Bloqueio de Posix	24
2.9.2. Ajuste de Desempenho com GFS2	24
2.9.3. Resolução de Problemas do Desempenho do Despejo de Bloqueios GFS2	25
<b>CAPÍTULO 3. INICIANDO</b>	<b>30</b>
3.1. TAREFAS DE PRÉ-REQUISITOS:	30
3.2. TAREFAS INICIAIS DE CONFIGURAÇÃO	30
<b>CAPÍTULO 4. GERENCIANDO O GFS2</b>	<b>32</b>
4.1. CRIANDO UM SISTEMA DE ARQUIVO	32
4.1.1. Uso	33
4.1.2. Exemplos	34
4.1.3. Opções Completas	34
4.2. MONTAGEM DE SISTEMA DE ARQUIVO	36
4.2.1. Uso	37
4.2.2. Exemplo	37
4.2.3. Uso Completo	37
4.3. DESMONTANDO O SISTEMA DE ARQUIVO	40
4.3.1. Uso	40
4.4. CONSIDERAÇÕES ESPECIAIS AO MONTAR OS SISTEMAS DE ARQUIVO GFS2	40
4.5. GERENCIAMENTO DE COTA DO GFS2	41
4.5.1. Configurando Quotas de Disco	41
4.5.1.1. Definindo Quotas no Modo Enforcement ou Accounting	42
4.5.1.1.1. Uso	42
4.5.1.1.2. Exemplos	42
4.5.1.2. Criando os Arquivos de Banco de Dados de Quotas	43
4.5.1.3. Atribuindo Quotas por Usuário	43
4.5.1.4. Atribuindo Quotas por Grupo	44
4.5.2. Gerenciando Quotas de Disco	44
4.5.3. Mantendo Quotas Precisar	45
4.5.4. Sincronizando Quotas com o Comando quotasync	45
4.5.4.1. Uso	46
4.5.4.2. Exemplos	46
4.5.5. Referências	47
4.6. AUMENTANDO UM SISTEMA DE ARQUIVO	47
4.6.1. Uso	47
4.6.2. Comentários	47
4.6.3. Exemplos	48
4.6.4. Uso Completo	48
4.7. ADICIONANDO DIÁRIOS AO SISTEMA DE ARQUIVO	49
4.7.1. Uso	49
4.7.2. Exemplos	49
4.7.3. Uso Completo	49
4.8. DIÁRIO DE DADOS	50
4.9. CONFIGURANDO ATUALIZAÇÕES DO ATIME	51
4.9.1. Montar com o relatime	52
4.9.1.1. Uso	52
4.9.1.2. Exemplo	52
4.9.2. Montar com noatime	52
4.9.2.1. Uso	52
4.9.2.2. Exemplo	52
4.10. SUSPENDENDO A ATIVIDADE EM UM SISTEMA DE ARQUIVO	53

4.10.1. Uso	53
4.10.2. Exemplos	53
4.11. REPARANDO UM SISTEMA DE ARQUIVO	53
4.11.1. Uso	55
4.11.2. Exemplo	55
4.12. MONTAGENS BIND E NOMES DE CAMINHO DE CONTEXTO DEPENDENTE.	55
4.13. MONTA BINDS E ORGANIZA A MONTAGEM DE SISTEMA DE ARQUIVO	57
4.14. A FUNÇÃO DE RETIRADA DO GFS2	59
<b>CAPÍTULO 5. DIAGNOSTICANDO E CORRIGINDO PROBLEMAS COM OS SISTEMAS DE ARQUIVOS GFS2</b>	<b>61</b>
5.1. O SISTEMA DE ARQUIVOS GFS2 EXIBE UM BAIXO DESEMPENHO	61
5.2. O SISTEMA DE ARQUIVOS GFS2 SUSPENDE E REQUER REINICIALIZAÇÃO DE UM NÓ.	61
5.3. O SISTEMA DE ARQUIVOS GFS2 SUSPENDE E REQUER REINICIALIZAÇÃO DE TODOS OS NÓS	61
5.4. O SISTEMA DE ARQUIVOS GFS2 NÃO MONTA UM NÓ DE CLUSTER RECÉM ADICIONADO.	62
5.5. ESPAÇO INDICADO COMO USADO EM UM SISTEMA DE ARQUIVOS VAZIO	62
<b>CAPÍTULO 6. CONFIGURANDO UM SISTEMA DE ARQUIVO DO GFS2 EM UM CLUSTER DE PACEMAKER.</b>	<b>63</b>
<b>APÊNDICE A. GERENCIAMENTO DE QUOTA DO GFS2 COM O COMANDO GFS2_QUOTA</b> .....	<b>65</b>
A.1. DEFININDO QUOTAS COM O COMANDO GFS2_QUOTA	65
A.1.1. Uso	65
A.1.2. Exemplos	66
A.2. EXIBINDO LIMITES DE QUOTAS E USO COM O COMANDO GFS2_QUOTA	66
A.2.1. Uso	66
A.2.2. Resultado do Comando	67
A.2.3. Comentários	67
A.2.4. Exemplos	67
A.3. SINCRONIZANDO QUOTAS COM O COMANDO GFS2_QUOTA.	68
A.3.1. Uso	68
A.3.2. Exemplos	68
A.4. ATIVANDO/DESATIVANDO A IMPOSIÇÃO DE QUOTAS	69
A.4.1. Uso	69
A.4.2. Exemplos	69
A.5. ATIVANDO QUOTAS ACCOUNTING	69
A.5.1. Uso	69
A.5.2. Exemplo	70
<b>APÊNDICE B. CONVERTENDO UM SISTEMA DE ARQUIVO UTILIZANDO OS GFS E GFS2</b> .....	<b>71</b>
B.1. CONVERSÃO DE NOMES DE CAMINHOS DEPENDENTES DE CONTEXTO (CDPNS)	71
B.2. PROCEDIMENTO DA CONVERSÃO DO GFS PARA GFS2	72
<b>APÊNDICE C. OS TRACEPOINTS E O ARQUIVO DEBUGFS GLOCKS</b> .....	<b>74</b>
C.1. TIPOS DE TRACEPOINT DO GFS2.	74
C.2. TRACEPOINTS	74
C.3. GLOCKS	75
C.4. A INTERFACE DE DEBUGFS DO GLOCK	77
C.5. GLOCK HOLDERS	80
C.6. TRACEPOINTS DO GLOCK	81
C.7. TRACEPOINTS DE BMAP	81
C.8. TRACEPOINTS DOS LOGS	82
C.9. ESTATÍSTICA DE GLOCK	82
C.10. REFERÊNCIAS	83
<b>APÊNDICE D. HISTÓRICO DE REVISÕES</b> .....	<b>84</b>

ÍNDICE REMISSIVO ..... 86



# INTRODUÇÃO

Este livro fornece informações sobre a configuração e suporte do Red Hat GFS2 (Red Hat Global File System 2), incluído no Resilient Storage Add-On.

## 1. PÚBLICO ALVO

Este livro pretende atingir os administradores de sistema do Linux que já são familiarizados com as seguintes atividades:

- Os procedimentos da administração de sistema do Linux, incluindo a configuração do kernel
- Instalação e configuração de rede de armazenamento compartilhada, tal como Fibre Channel SANs.

## 2. DOCUMENTAÇÃO RELACIONADA

Para mais informações sobre como usar o Red Hat Enterprise Linux, consulte os seguintes recursos:

- *Guia de Instalação*— Fornece informações sobre a instalação do Red Hat Enterprise Linux 6.
- *Guia de Implementação*— Fornece informação sobre a implementação, configuração e administração do Red Hat Enterprise Linux 6.
- *Guia de Instalação*— Fornece informações sobre a instalação do Red Hat Enterprise Linux 6.

Para mais informações sobre a Adição com Alta Disponibilidade (High Availability Add-On) e sobre a Adição de Armazenamento Resiliente (Resilient Storage Add-On) para Red Hat Enterprise Linux 6, consulte os seguintes recursos:

- *Visão Geral do High Availability Add-On Overview*— Fornece uma visão geral de alto nível de Adição de Alta Disponibilidade.
- *Administração de Cluster (Cluster Administration)* — Fornece informações sobre a instalação, configuração e gerenciamento do Complemento de Alta Disponibilidade.
- *Administrador do Gerenciador de Volume Lógico*— Fornece uma descrição do Gerenciador de Volume Lógico (LVM), incluindo informações sobre como executar o LVM em um ambiente de cluster.
- *DM Multipath* — Fornece informações sobre como usar o recurso Device-Mapper Multipath (Dispositivo Mapeador Multipath) do Red Hat Enterprise Linux.
- *Linux Virtual Server Administration*— Fornece informações sobre configuração de sistemas de alto desempenho e serviços com a Adição de Balanceador de Carga (Load Balancer Add-On), um conjunto de componentes de software integrado que fornece Linux Virtual Servers (LVS) para balanceamento de carga de IP através um conjunto de servidores reais.
- *Notas de Lançamento*— Fornece informações sobre lançamento atual dos produtos da Red Hat.

A documentação do Red Hat Cluster Suite e outros documentos da Red Hat estão disponíveis nas versões HTML, PDF, e RPM na Red Hat Enterprise Linux. A documentação em CD e on-line pode ser encontrada no <https://access.redhat.com/site/documentation/>.

### 3. PRECISAMOS DE SEU FEEDBACK

Se você encontrar algum erro de digitação neste manual ou se você tem alguma idéia para tornar este manual melhor, nós gostaríamos de saber. Por favor envie um relatório ao Bugzilla:

<http://bugzilla.redhat.com/> sobre este produto **Red Hat Enterprise Linux 6** e o componente **doc-Global\_File\_System\_2**. Quando enviar um relatório de erro, lembre-se de mencionar o identificador deste manual:

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

Se você tiver uma sugestão para melhorar esta documentação, tente ser o mais específico possível. Se você encontrar um erro, inclua o número da seção e cole uma parte do texto para que possamos encontrá-lo com facilidade.

# CAPÍTULO 1. VISÃO GERAL DO GFS2

O sistema de arquivos GFS2 da Red Hat está incluído no Complemento de Armazenamento Resiliente. Ele é um sistema nativo de arquivos que lida diretamente com a interface do sistema de arquivos kernel do Linux (VFS layer). Quando implementado como um sistema de arquivos do cluster, o GFS2 implementa os metadados distribuídos e diários múltiplos. A Red Hat suporta o uso do sistema de arquivos GFS2 somente implementado do Complemento de Alta Disponibilidade.



## NOTA

Apesar do sistema de arquivos GFS2 pode ser implementado em um sistema isolado ou como parte de uma configuração de cluster, para o lançamento do Red Hat Enterprise Linux 6, a Red Hat não suporta o uso do GFS2 como um sistema de arquivos de nó único. A Red Hat suporta uma variedade de sistemas de arquivos de nó único de alto desempenho que são otimizados para um nó único e portanto possuem custos mais baixos do que um sistema de arquivos de cluster. A Red Hat recomenda usar estes sistemas de arquivos preferencialmente do que o GFS2 em casos onde somente um nó precisa montar um sistema de arquivos.

A Red Hat continuará a dar suporte para sistemas de arquivos GFS2 em nó único para montagem de snapshots dos sistemas de arquivos de cluster (por exemplo, para propósitos de backup).



## NOTA

A Red Hat não suporta o uso do GFS2 para implementações de sistemas de arquivos de cluster maiores de 16 nós.

O GFS2 é baseado em uma arquitetura de 16-bits que pode teoricamente acomodar um sistema de arquivos 8 EB. Entretanto, atualmente o tamanho máximo suportado de um sistema de arquivos GFS2 para hardware de 64-bits é de 100 TB. O tamanho máximo suportado atual de um sistema de arquivos GFS2 de 32-bits é de 16 TB. Se seu sistema requer sistemas de arquivos GFS2, contate seu representante de serviços Red Hat.

Quando determinar o tamanho de seu sistema de arquivos, você deve considerar suas necessidades de recuperação. Executando o comando `fsck.gfs2` em um sistema de arquivos muito grande pode levar um longo tempo e consumir uma grande quantidade de memória. Adicionalmente, no evento de falha de um disco ou subsistema de disco, o tempo de recuperação é limitado pela velocidade de sua mídia de backup. Para informações sobre a quantidade de memória que o comando `fsck.gfs2` requer, veja a [Seção 4.11, “Reparando um Sistema de Arquivo”](#).

Quando configurado em um cluster, os nós do Red Hat GFS2 podem ser configurados e gerenciados com as ferramentas do Complemento de Alta Disponibilidade. O Red Hat GFS2 então fornece compartilhamento de dados entre nós GFS2 em um cluster, com uma única visão consistente do espaço do nome do sistema de arquivos por todo os nós GFS2. Isto permite processos em nós diferentes para compartilhar arquivos GFS2 na mesma maneira que processos no mesmo nó podem partilhar arquivos em um sistema de arquivos local, sem uma diferença perceptível. Para informações sobre o Complemento de Alta Disponibilidade consulte *Configurando e Gerenciando um Cluster Red Hat* (Configuring and Managing a Red Hat Cluster).

Enquanto um sistema de arquivos GFS2 pode ser usado fora do LVM, a Red Hat suporta somente sistemas de arquivos GFS2 que são criados em um volume lógico CLVM. O CLVM é incluído no Complemento de Armazenamento Resiliente. Ele é uma implementação de cluster geral do LVM, ativada pelo daemon CLVM, `c1vmd`, que gerencia os volumes lógicos LVM em um cluster. O daemon

faz isso possível para usar o LVM2 para gerenciar volumes lógicos. Para informações sobre o gerenciador de volume LVM, veja *Administração do Gerenciador de Volume Lógico* (Logical Volume Manager Administration).

O módulo do kernel `gfs2.ko` implementa o sistema de arquivos GFS2 e é carregado em nós do cluster GFS2.



## NOTA

Quando você configurar um sistema de arquivos GFS2 como um sistema de arquivos de cluster, você deve garantir que todos os nós no cluster possuem acesso ao armazenamento compartilhado. Configurações de cluster assimétricas nas quais alguns nós possuem acesso ao armazenamento compartilhado e outros não são suportados. Isto não requer que todos os nós montem o sistema de arquivos GFS2.

Este capítulo fornece algumas informações básicas abreviadas como pano de fundo para ajudá-lo a entender o GFS2. Ele contém as seguintes seções:

- [Seção 1.1, “Recursos Novos e Alterados”](#)
- [Seção 1.2, “Antes de Configurar o GFS2”](#)
- [Seção 1.4, “Diferenças entre o GFS e GFS2”](#)
- [Seção 1.3, “Instalando o GFS2”](#)
- [Seção 2.9, “Bloqueio de Nó GFS2”](#)

## 1.1. RECURSOS NOVOS E ALTERADOS

Esta seção lista os novos e modificados recursos do sistema de arquivos GFS2 e da documentação GFS2 que são incluídos com os lançamentos iniciais e subsequentes do Red Hat Enterprise Linux 6.

### 1.1.1. Recursos Novos e Alterados para o Red Hat Enterprise Linux 6.0

O Red Hat Enterprise Linux 6.0 inclui a seguinte documentação e atualizações de recursos e mudanças.

- Para o lançamento do Red Hat Enterprise Linux 6, a Red Hat não dá suporte para o uso do GFS2 como um sistema de arquivos de nó único.
- Para o lançamento do Red Hat Enterprise Linux 6, o comando `gfs2_convert` que faz a atualização do GFS para um sistema de arquivos GFS2 foi melhorado. Para informações sobre este comando, veja o [Apêndice B, \*Convertendo um Sistema de Arquivo utilizando os GFS e GFS2\*](#)
- O lançamento do Red Hat Enterprise Linux 6 suporta as opções de montagem `discard`, `nodiscard`, `barrier`, `nobarrier`, `quota_quantum`, `statfs_quantum`, e `statfs_percent`. Para informações sobre montar um sistema de arquivos GFS2, veja a [Seção 4.2, “Montagem de Sistema de Arquivo”](#).
- A versão Red Hat Enterprise Linux 6 deste documento contém uma nova seção, a [Seção 2.9, “Bloqueio de Nó GFS2”](#). Esta seção descreve alguns dos sistemas de arquivos internos GFS2.

### 1.1.2. Recursos Novos e Modificados para o Red Hat Enterprise Linux 6.1

O Red Hat Enterprise Linux 6.1 inclui a seguinte documentação e atualizações de recursos e mudanças.

- A partir do lançamento do Red Hat Enterprise Linux 6.1, o GFS2 suporta as facilidades de quotas básicas do Linux. O gerenciamento de quotas GFS2 está documentado na [Seção 4.5, “Gerenciamento de Cota do GFS2”](#).

Para lançamentos anteriores do Red Hat Enterprise Linux, o GFS2 requer o comando `gfs2_quota` para gerenciar quotas. A documentação para o `gfs2_quota` é agora fornecida no [Apêndice A, Gerenciamento de Quota do GFS2 com o comando `gfs2\_quota`](#).

- Este documento agora contém um novo capítulo, o [Capítulo 5, Diagnosticando e Corrigindo Problemas com os Sistemas de Arquivos GFS2](#).
- Pequenas correções técnicas e clarificações que foram feitas através do documento.

### 1.1.3. Recursos Novos e Modificados para o Red Hat Enterprise Linux 6.2.

O Red Hat Enterprise Linux 6.2 inclui a seguinte documentação e atualização de recursos e mudanças.

- Desde o lançamento do Red Hat Enterprise Linux 6.2, o GFS2 suporta o comando `tunegfs2` que substitui alguns recursos do comando `gfs2_tool`. Para mais informações, consulte o man page de `tunegfs2`.

As seções a seguir foram atualizadas para fornecer procedimentos administrativos que não requerem o uso do comando `gfs2_tool`:

- [Seção 4.5.4, “Sincronizando Quotas com o Comando `quotasync`”](#). and [Seção A.3, “Sincronizando Quotas com o Comando `gfs2\_quota`”](#) agora descreve como mudar o parâmetro `quota_quantum` de seu valor padrão de 60 segundos utilizando a opção de montagem `quota_quantum=`.
- [Seção 4.10, “Suspendendo a Atividade em um Sistema de Arquivo”](#) agora descreve como suspender a edição da atividade de um sistema de arquivo utilizando o comando `dmsetup suspend`.
- Este documento inclui um novo apêndice, [Apêndice C, Os tracepoints e o Arquivo `debugfs\_glocks`](#). Este apêndice descreve a interface do glock `debugfs` e os pontos de rastreamento do GFS2. É destinado aos usuários mais avançados que possuam conhecimento do sistema de arquivo internos e que queiram saber mais sobre a criação do GFS2 e como depurar problemas específicos do GFS2.

### 1.1.4. Recursos Novos e Alterados para o Red Hat Enterprise Linux 6.3

Para o lançamento do Red Hat Enterprise Linux 6.3 release, este documento contém um novo capítulo, [Capítulo 2, Configuração do GFS2 e Considerações Operacionais](#). Este capítulo fornece recomendações para otimizar o desempenho do GFS2, incluindo recomendações para criar, utilizar e manter um sistema de arquivo GFS2.

Além disso, este documento apresenta pequenas explicações e correções.

### 1.1.5. Recursos Novos e Modificados para o Red Hat Enterprise Linux 6.4

Pois o lançamento do Red Hat Enterprise Linux 6.4, [Capítulo 2, Configuração do GFS2 e Considerações Operacionais](#) foi atualizado com pequenos esclarecimentos.

## 1.1.6. Recursos Novos e Modificados para o Red Hat Enterprise Linux 6.6

Para o lançamento do Red Hat Enterprise Linux 6.6, este documento contém um novo capítulo, [Capítulo 6, Configurando um Sistema de Arquivo do GFS2 em um Cluster de Pacemaker](#). Este capítulo fornece um guia de passos necessários para configurar um cluster do Pacemaker que inclui um sistema de arquivo do GFS2.

Além disso, este documento apresenta pequenas explicações e correções.

## 1.2. ANTES DE CONFIGURAR O GFS2

Antes de você instalar e configurar o GFS2, observe as seguintes características da chave de seus sistemas de arquivo do GFS2:

### Nós de GFS2

Determina quais nós no cluster irão montar os sistemas de arquivos GFS2.

### Número de sistemas de arquivo

Determine quantos sistemas de arquivos do GFS2 serão criados inicialmente. (Você poderá adicionar mais sistemas de arquivos mais tarde).

### Nome do Sistema de Arquivo

Determina um nome único para cada sistema de arquivo. O nome deve ser único para todos os sistemas de arquivos `lock_dlm` no cluster. Cada nome do sistema de arquivos é requerido na forma de uma variável de parâmetro. Por exemplo, este livro usa nomes de arquivos de sistemas `mydata1` e `mydata2` em alguns exemplos de procedimentos.

### Diários

Determina o número de diários (journals) para seus sistemas de arquivos GFS2. Um diário é requerido para cada nó que monta o sistema de arquivo GFS2. O GFS2 permite que você adicione diários dinamicamente mais tarde conforme servidores adicionais montam um sistema de arquivos. Para informações sobre adicionar diários a um sistema de arquivos GFS2, veja a [Seção 4.7, “Adicionando Diários ao Sistema de Arquivo”](#).

### Dispositivos de armazenamento e partições

Determine os dispositivos de armazenamento e partições a serem usadas para a criação de volumes lógicos (via CLVM) nos sistemas de arquivo.



### NOTA

Você poderá ter problemas de desempenho com o GFS2 quando muitas operações de criação e remoção são emitidas a partir de mais de um nó no mesmo diretório ao mesmo tempo. Se isto causar problemas de desempenho em seu sistema, você deveria localizar as criações e exclusões de arquivos por um nó à diretórios específicos daquele nó o máximo possível.

Para outras recomendações sobre a criação, uso e manutenção de um sistema de arquivo GFS2, consulte o [Capítulo 2, Configuração do GFS2 e Considerações Operacionais](#).

## 1.3. INSTALANDO O GFS2

Além dos pacotes necessários para o Red Hat High Availability Add-On, você precisa instalar o pacote `gfs2-utils` para GFS2 e para o pacote `lvm2-cluster` para o Clustered Logical Volume Manager (CLVM). Os pacotes `lvm2-cluster` e `gfs2-utils` são parte do canal `ResilientStorage`, que deve ser habilitado antes de instalar os pacotes.

Você pode usar o seguinte comando `yum install` para instalar os pacotes de software do High Availability Add-On:

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

Para informações gerais do Red Hat High Availability Add-On e administração do cluster, veja o manual do *Cluster Administration*.

## 1.4. DIFERENÇAS ENTRE O GFS E GFS2

Esta seção lista as melhorias e mudanças que o GFS2 oferece em relação ao GFS.

Migrar do GFS para GFS2 requer que você converta seus sistemas de arquivos GFS para GFS2 com o utilitário `gfs2_convert`. Para informações sobre o utilitário `gfs2_convert`, veja o [Apêndice B, Convertendo um Sistema de Arquivo utilizando os GFS e GFS2](#)

### 1.4.1. Nomes de Comando do GFS2

Em geral, a funcionalidade do GFS2 é idêntica ao GFS. Os nomes dos comandos do sistema de arquivos, entretanto especificam GFS2 ao invés de GFS. A [Tabela 1.1, “Comandos do GFS e GFS2”](#) mostra os comandos equivalentes do GFS e GFS2 e suas funcionalidades.

Tabela 1.1. Comandos do GFS e GFS2

Comando do GFS	Comando do GFS2	Descrição
<code>mount</code>	<code>mount</code>	Montar um sistema de arquivos. O sistema pode determinar se o sistema de arquivos é um sistema de arquivos do tipo GFS ou GFS2. Para informações sobre as opções de montagem do GFS2 veja a página <code>man gfs2_mount(8)</code> .
<code>umount</code>	<code>umount</code>	Desmonte um sistema de arquivo
<code>fsck</code> <code>gfs_fsck</code>	<code>fsck</code> <code>fsck.gfs2</code>	Verifique e repare um sistema de arquivo desmontado.
<code>gfs_grow</code>	<code>gfs2_grow</code>	Aumente um sistema de arquivo montado.
<code>gfs_jadd</code>	<code>gfs2_jadd</code>	Adicione um diário ao sistema de arquivo montado

Comando do GFS	Comando do GFS2	Descrição
<pre>gfs_mkfs</pre> <pre>mkfs -t gfs</pre>	<pre>mkfs.gfs2</pre> <pre>mkfs -t gfs2</pre>	Crie um sistema de arquivo em um dispositivo de armazenamento.
<b>gfs_quota</b>	<b>gfs2_quota</b>	Gerencie cotas em um sistema de arquivo montado. A partir do lançamento do Red Hat Enterprise Linux 6.1, o GFS2 suporta as facilidades de quotas básicas do Linux. O gerenciamento de quotas GFS2 está documentado na <a href="#">Seção 4.5, “Gerenciamento de Cota do GFS2”</a> .
<b>gfs_tool</b>	<b>tunegfs2</b>  montar parâmetros  <b>dmsetup</b> <b>suspend</b>	Configure, ajuste ou obtenha informações sobre um sistema de arquivo. O comando <b>tunegfs2</b> é suportado desde o lançamento do Red Hat Enterprise Linux 6.2. Também existe o comando <b>gfs2_tool</b> .
<b>gfs_edit</b>	<b>gfs2_edit</b>	Exiba, imprima ou edite estruturas internas de sistema de arquivo. O comando <b>gfs2_edit</b> pode ser usado para sistemas de arquivo do GFS, assim como para o sistema de arquivo GFS2.
<b>gfs_tool</b> <b>setflag</b> <b>jdata/inheri</b> <b>t_jdata</b>	<b>chattr +j</b> (preferido)	Ativar os diários em um arquivo ou diretório.
<b>setfacl/getfacl</b>	<b>setfacl/getfacl</b>	Defina ou obtenha a lista de controle de acesso ao arquivo para um arquivo ou diretório.
<b>setfattr/getfattr</b>	<b>setfattr/getfattr</b>	Defina ou obtenha os atributos estendidos de um arquivo.

Para uma listagem completa das opções suportadas para os comandos do sistema de arquivo do GFS2, veja as páginas `man`.

## 1.4.2. Diferenças Adicionais Entre GFS e GFS2

Esta seção resume as diferenças adicionais na administração do GFS e GFS2 que não são descritas na [Seção 1.4.1, “Nomes de Comando do GFS2”](#).

### 1.4.2.1. Nomes de Caminhos Dependentes de Contexto

Os sistemas de arquivo do GFS2 não fornecem suporte para os Nomes de Caminho de Contexto Dependente (CDPNs), o qual permite que você crie links simbólicos que apontam para os arquivos ou diretórios de destino de variáveis. Para esta funcionalidade do GFS2, você pode usar a opção `bind` do



comando `mount`. Para informações sobre `bind` de montagens e nomes de caminhos de contexto dependentes no GFS2, veja a [Seção 4.12, “Montagens Bind e Nomes de Caminho de Contexto Dependente.”](#)

#### 1.4.2.2. Módulo `gfs2.ko`

O módulo do kernel que implementa um sistema de arquivos GFS é `gfs.ko`. O módulo do kernel que implementa o sistema de arquivos GFS2 é `gfs2.ko`.

#### 1.4.2.3. Ativando Imposição de Cota no GFS2

Nos sistemas de arquivos GFS2, a imposição de quotas é desativada por padrão e deve ser ativada explicitamente. Para informações sobre ativar e desativar a imposição de quotas, veja a [Seção 4.5, “Gerenciamento de Cota do GFS2”](#).

#### 1.4.2.4. Diário de Dados

Os sistemas de arquivos GFS2 suporta o uso do comando `chattr` para definir e limpar o sinalizador `j` em um arquivo ou diretório. Definindo o sinalizador `+j` em um arquivo habilita o diário de dados nesse arquivo. Definindo sinalizador `+j` em um diretório significa "herdar `jdata`", que indica que todos os arquivos e diretórios subsequentemente criados nesse diretório obtenham diários. Usando o comando `chattr` é a maneira preferida para ativar e desativar o diários de dados em um arquivo.

#### 1.4.2.5. Adicionando Diários de forma Dinâmica

Em sistemas de arquivos GFS, diários são metadados incorporados que existem fora do sistema de arquivos, fazendo-se necessário estender o tamanho do volume lógico que contém o sistema de arquivos antes de adicionar os diários. Em sistemas de arquivos GFS2, diários são arquivos planos (embora escondidos). Isto significa que para sistemas de arquivos GFS2, os diários podem ser dinamicamente adicionados conforme servidores adicionais montam um sistema de arquivos, desde que haja espaço no sistema de arquivo para os diários de dados adicionais. Para informações sobre adicionar diários a um sistema de arquivos GFS2, veja a [Seção 4.7, “Adicionando Diários ao Sistema de Arquivo”](#).

#### 1.4.2.6. Removido parâmetro `atime_quantum`

O sistema de arquivos GFS2 não suporta o parâmetro ajustável `atime_quantum`, que pode ser usado pelo sistema de arquivos GFS para especificar com qual frequência as atualizações do `atime` ocorrem. No seu lugar o GFS2 suporta as opções de montagem `relatime` e `noatime`. A opção de montagem `relatime` é recomendada para alcançar comportamento similar para configurar o parâmetro `atime_quantum` no GFS.

#### 1.4.2.7. A opção '`data=`' do comando `mount`

Quando montar sistemas de arquivos GFS2, você pode especificar a opção `data=ordered` ou `data=writeback` do `mount`. Quando o `data=ordered` é configurado, os dados do usuários modificados por uma transação são liberados para o disco antes da transação ser enviada ao disco. Isto deve prevenir o usuário de ver blocos não inicializados em um arquivo depois de um travamento. Quando o `data=writeback` é definido, os dados do usuário são escritos no disco em qualquer momento após que ele é alterado. Isto não promove a mesma garantia de consistência como no modo `ordered`, mas deve ser um pouco mais rápido para algumas cargas de trabalho. O padrão é o modo `ordered` (ordenado).

### 1.4.2.8. O comando `gfs2_tool`

O comando `gfs2_tool` suporta um conjunto diferente de opções para o GFS2 do que o comando `gfs_tool` suporta para o GFS:

- O comando `gfs2_tool` suporta o parâmetro `journals` (diários) que mostra informações sobre os diários configurados atualmente, incluindo quantos diários o sistema de arquivos contém.
- O comando `gfs2_tool` não suporta o sinalizador `counters`, que o comando `gfs_tool` usa para mostrar estatísticas GFS.
- O comando `gfs2_tool` não suporta o sinalizador `inherit_jdata`. Para sinalizar um diretório como "inherit jdata" (herdar jdata), você pode definir o sinalizador `jdata` no diretório ou você pode usar o comando `chattr` para definir o sinalizador `+j` no diretório, Usando o comando `chattr` é a maneira preferida para ativar e desativas o diário de dados em um arquivo.



#### NOTA

Desde o lançamento do Red Hat Enterprise Linux 6.2, o GFS2 suporta o comando `tunegfs2` que substitui alguns recursos do comando `gfs2_tool`. Para mais informações, consulte o man page de `tunegfs2`. As funções `settune` e `gettune` do comando `gfs2_tool` foram substituídas pelas opções da linha de comando do comando `mount` o qual permite que sejam definidos por meio do arquivo `fstab` quando necessário.

### 1.4.2.9. O comando `gfs2_edit`

O comando `gfs2_edit` suporta um conjunto diferente de opções para o GFS2 do que o comando `gfs_edit` suporta para o GFS. Para informações sobre opções específicas de cada versão que o comando suporta, veja as páginas man do `gfs2_edit` e `gfs_edit`.

## 1.4.3. Aprimoramentos de Desempenho do GFS2

Existem muitos recursos dos sistemas de arquivos GFS2 que não resultam em uma diferença na interface do usuário dos sistemas de arquivos GFS mas que melhoram o desempenho do sistema.

Um sistema de arquivos do GFS2 fornece as seguintes melhorias no desempenho do sistema de arquivos nas seguintes maneiras:

- Melhor desempenho para uso pesado em um único diretório.
- Sincronia de operações de E/S mais rápida.
- Leitura em cache mais rápida (sem cabeçalho de bloqueio)
- E/S direta mais rápida com arquivos pré-alocados (considerando que o tamanho da E/S seja razoavelmente grande, como blocos de 4M)
- Operações de E/S em geral mais rápidas
- Execução mais rápida do comando `df`, por causa das chamadas mais rápidas do `statfs`.

- O modo `atime` foi melhorado para reduzir o número de edições de operações de E/S geradas pelo `atime` quando comparadas com o GFS.

Os sistemas de arquivo GFS2 fornecem suporte mais amplo e mais mainstream nas seguintes formas:

- GFS2 é parte do kernel superior (integrado ao 2.6.19).
- GFS2 suporta os seguintes recursos.
  - atributos de arquivo estendido (`xattr`)
  - as configurações dos atributos `lsattr()` e `chattr()` via chamadas padrão `ioctl()`.
  - carimbo de data e hora do nanosecond

Um sistema de arquivo do GFS2 fornece as seguintes melhorias para a eficiência interna do sistema de arquivo.

- GFS2 usa menos memória de kernel
- O GFS2 não requer número de geração de metadados.

A alocação dos metadados do GFS2 não requer leitura. As cópias dos blocos de metadados em diversos diários são gerenciadas por blocos de anulação de diários antes do lançamento do bloqueio.

- O GFS2 inclui um gerenciador de log mais simples que não sabe nada sobre inodes sem vínculo ou mudanças de cotas.
- Os comandos `gfs2_grow` e `gfs2_jadd` usam bloqueios para evitar que instâncias múltiplas sejam executadas ao mesmo tempo.
- O código ACL foi simplificado para chamadas como `creat()` e `mkdir()`.
- Inodes sem vínculos, mudanças de cotas e mudanças de `statfs` são recuperadas sem remontar o diário.

## CAPÍTULO 2. CONFIGURAÇÃO DO GFS2 E CONSIDERAÇÕES OPERACIONAIS

O sistema de arquivo (GFS2) do Global File System 2 permite que diversos computadores ("nós") em um cluster, compartilhem o mesmo armazenamento de forma cooperativa. Para alcançar esta cooperação e manter uma consistência de dados entre os nós, os nós empregam um esquema de bloqueio em todo o cluster para os recursos de sistema de arquivo. Este esquema de bloqueio utiliza protocolos de comunicação tais como o TCP/IP para troca de informações de bloqueio.

Você pode aprimorar o desempenho seguindo as recomendações descritas neste capítulo, incluindo recomendações para a criação, utilizando e mantendo um sistema de arquivo GFS2.



### IMPORTANTE

Certifique-se que sua implantação do Complemento de Alta Disponibilidade atenda suas necessidades e possa ser suportada. Consulte um representante autorizado Red Hat para verificar suas configurações antes da implementação.

## 2.1. FORMATANDO CONSIDERAÇÕES

Esta seção provê recomendações para como formatar seu sistema de arquivo do GFS2 para otimizar desempenho.

### 2.1.1. Tamanho do Sistema de Arquivo: Quanto Menor Melhor

O GFS2 é baseado em uma arquitetura de 16-bits que pode teoricamente acomodar um sistema de arquivos 8 EB. Entretanto, atualmente o tamanho máximo suportado de um sistema de arquivos GFS2 para hardware de 64-bits é de 100 TB e o tamanho máximo suportado atual de um sistema de arquivos GFS2 de 32-bits é de 16 TB.

Observe que mesmo que os sistemas de arquivo grandes do GFS2 sejam possíveis, não significa que são recomendados. Como via de regra, com o GFS2, quanto menor melhor será: é melhor ter 10 1TB sistemas de arquivo do que 10TB sistema de arquivo.

Existem diversas razões pela qual você deveria manter seus sistemas de arquivo GFS2 pequenos:

- É necessário menos tempo para fazer o back up de cada sistema de arquivo.
- É necessário menos tempo caso você precise verificar o sistema de arquivo com o comando `fsck.gfs2`.
- É necessário menos memória caso você precise verificar o sistema de arquivo com o comando `fsck.gfs2`.

Além disso, menos grupos de recurso para manutenção significa um melhor desempenho.

É claro que se você diminui muito seu sistema de arquivo GFS2, você pode não ter mais espaço e isto lhe trará consequências. Você deve considerar seus próprios casos de uso antes de decidir sobre o tamanho.

### 2.1.2. Tamanho do Bloco: Prefere-se (4K) Blocos Padrão

Desde o lançamento do Red Hat Enterprise Linux, o comando `mkfs.gfs2` tenta estimular um tamanho de bloco mais adequado baseado na topologia do dispositivo. Em geral, prefere-se blocos com 4K, pois

é o tamanho da página padrão (memória) para Linux. Ao contrário de alguns sistemas de arquivo, o GFS2 faz a maioria de suas operações utilizando os buffers do kernel de 4K. Se o tamanho do seu bloco é de 4K, o kernel dispenderá de menos trabalho para manipular os buffers.

Recomenda-se que você utilize o tamanho de bloco padrão, o qual deve gerar um maior desempenho. Você pode precisar utilizar um tamanho de bloco diferente somente se desejar armazenamento eficiente de muitos arquivos bem pequenos.

### 2.1.3. Números de Diários: Um para cada Nó que Monta

O GFS2 requer um jornal para cada nó no cluster que precise montar o sistema de arquivo. Por exemplo, se você possui um cluster de 16 nós, mas precisa montar somente o sistema de arquivo de dois nós, você precisará somente de dois diários. Se você precisar montar um terceiro nó, você pode adicionar um diário com o comando `gfs2_jadd`. Com o GFS2 você poderá adicionar diários imediatamente.

### 2.1.4. Tamanho do Diário: Padrão (128 MB) É Geralmente o mais adequado

Quando você executa o comando `mkfs.gfs2` para criar um sistema de arquivo GFS2, você pode especificar o tamanho dos diários. Se você não especificar um tamanho, ele terá 128MB como padrão, o qual deve ser mais adequado para a maioria dos aplicativos.

Alguns administradores de sistema devem achar que 128MB é excessivo, e podem tentar reduzir o tamanho do diário para um mínimo de 8MB ou 32MB como mais tradicional. Embora possa funcionar, ele pode impactar de forma severa o desempenho. Como muitos sistemas de arquivos de diário, todas as vezes que o GFS2 grava um metadado, o metadado fica comprometido com o diário antes que seja colocado em prática. Isto assegura que se o sistema travar, ou perder força, você pode recuperar todos os metadados quando o diário for reproduzido automaticamente durante a montagem. No entanto, não é necessário muita atividade de sistema de arquivo para preencher um diário de 8MB e quando o diário estiver cheio, o desempenho será mais lento pois o GFS precisa esperar para gravar no armazenamento.

Geralmente recomenda-se que utilize o tamanho de diário padrão de 128MB. Se seu sistema de arquivo for muito pequeno (por exemplo, 5GB), um diário de 128MB pode não ser prático. Se você possuir um sistema de arquivo maior e puder ter o espaço, o uso dos diários de 256MB pode aprimorar o desempenho.

### 2.1.5. Tamanho e Número de Grupos de Recursos.

Quando um sistema de arquivo GFS2 é criado com o comando `mkfs.gfs2`, ele divide o armazenamento em faixas uniformes conhecidas como grupos de recursos. Ele tenta estimar o tamanho do grupo de recurso mais adequado (variando entre 32MB até 2GB). Você pode sobrescrever o padrão como a opção `-r` do comando `mkfs.gfs2`.

Seu tamanho de grupo de recurso mais adequado depende de como você irá utilizar o sistema de arquivo. Pense no quão cheio estará e se será fragmentado ou não.

Você deveria experimentar com tamanhos de grupo de recursos diferentes para ver quais resultam em um desempenho mais avançado. É uma boa prática experimentar com um cluster de teste antes de implementar o GFS2 em total produção.

Se seu sistema de arquivo possuir muitos grupos de recursos (cada qual sendo muito pequeno), a alocação de blocos pode perder muito tempo tentando encontrar dezenas (ou centenas) de grupos de recursos para um bloco livre. O quanto mais cheio o sistema de arquivo estiver, mais os grupos de

recursos serão vasculhados e cada um deles requer um bloqueio em todo o cluster. Isto levará a um desempenho mais lento.

Caso seu sistema de arquivo possua muito poucos grupos de recursos (cada qual sendo muito grande), as alocações de blocos podem lutar mais vezes pelo mesmo bloqueio de grupo de recurso, o qual também impactará o desempenho. Por exemplo, se você possui um sistema de arquivo de 10GB que está dividido em cinco grupos de recursos de 2GB, os nós em seu cluster irão lutar por estes cinco grupos de recurso mais vezes do que se o mesmo sistema de arquivo fosse dividido em 320 grupos de recursos de 32MB. O problema é exacerbado se seu sistema de arquivos estiver quase completo, pois toda a alocação de bloco deve precisar procurar em diversos grupos de recursos antes que ele encontre um com bloco livre. O GFS2 tenta suavizar este problema de duas formas:

- Primeiro, quando um grupo de recurso estiver completamente cheio, ele se lembra disto e tenta evitar verificá-lo em alocações no futuro (até que um bloco seja liberado dele). Se você nunca remover arquivos, a contenção será menos severa. No entanto, se seu aplicativo remover constantemente blocos e alocar novos blocos em um sistema de arquivo que esteja quase cheio, a contenção será muito alta e terá um impacto no desempenho bastante severo.
- Em segundo lugar, quando novos blocos são adicionados a um arquivo existente (por exemplo, acrescentando) o GFS2 tentará agrupar os novos blocos no mesmo grupo de recursos como o arquivo. Isto é feito para aumentar o desempenho: em um disco rotativo, procura por menos tempo quando estão fisicamente juntos.

No pior dos casos, existirá um diretório central, no qual todos os nós criam arquivos porque todos os nós lutarão constantemente para bloquear o mesmo grupo de recurso.

## 2.2. FRAGMENTAÇÃO DE SISTEMA DE ARQUIVO

A Red Hat Enterprise Linux 6.4 apresenta melhorias no gerenciamento de fragmentação de arquivo no GFS2. Com o Red Hat Enterprise Linux 6.4, as gravações simultâneas resultam em menos fragmentações de arquivos e portanto em melhor desempenho para estas cargas de trabalho.

Embora não haja ferramenta de defragmentação para o GFS2 no Red Hat Enterprise Linux, você pode defragmentar arquivos individuais, identificando-os com as ferramentas de filefrag, copiando-os para arquivos temporários e mantendo-os para substituir os originais. (Este procedimento pode também ser feito em versões anteriores ao Red Hat Enterprise Linux 6.4 desde que a gravação seja feita sequencialmente).

## 2.3. PROBLEMAS DE ALOCAÇÃO DE BLOCOS

Esta seção fornece um sumário de problemas relacionados à alocação de blocos nos sistemas de arquivos do GFS2. Embora os aplicativos que somente gravam dados não se importam em como ou onde um bloco é alocado, um pouco de conhecimento sobre como a alocação de bloco funciona pode ajudar a otimizar desempenho.

### 2.3.1. Deixe Espaço Livre no Sistema de Arquivo.

Quando um sistema de arquivo do GFS2 estiver praticamente cheio, o alocador de bloco começa a ter dificuldades em encontrar espaço para novos blocos a serem alocados. Como resultado disto, os blocos dados pelo alocador tendem a ser espremidos no final de um grupo de recurso ou em pequenas faixas onde a fragmentação de arquivo é muito mais provável. Esta fragmentação de arquivo pode causar problemas de desempenho. Além disso, quando um GFS2 está quase cheio, o alocador de bloco do GFS2 gasta mais tempo procurar por grupos de recursos múltiplos, e isto adiciona contenção de bloqueio que não necessariamente estaria lá em um sistema de arquivo que tenha espaço amplo livre. Isto também pode causar problemas de desempenho.

Por estas razões, recomendamos que você não execute um sistema de arquivo que esteja com mais de 85 por cento cheio, embora este número possa variar dependendo da carga de trabalho.

### 2.3.2. Aloque Cada Nó a Seus Próprios Arquivos, Se Possível

Devido a forma que o gerenciador de bloqueio distribuído (DLM) funciona, existirá mais contenção de bloqueio se todos os arquivos forem alocados por um nó e outros nós precisarem adicionar blocos àqueles arquivos.

No GFS (versão 1), todos os bloqueios eram gerenciados por um gerenciador de bloqueio central, cuja função era controlar bloqueios nos clusters. Este Gerenciador de bloqueio Unificado Maior (GULM) era prolemático, pois era um único ponto de falha. O esquema de bloqueio de substituição do GFS2, DLM, espalha os bloqueios por todo o cluster. Se qualquer nó no cluster cair, seu bloqueio é recuperado por outros nós.

Com o DLM, o primeiro nó a bloquear um recurso (como um arquivo) se torna o "bloqueio mestre" para aquele bloqueio. Outros nós podem bloquear este recurso, mas eles terão que pedir permissão do bloqueio mestre primeiro. Cada nó sabe qual o bloqueio mestre para cada bloqueio, e cada nó sabe qual nó a quem ele emprestou um bloqueio. Bloquear uma trava no nó mestre é muito mais rápido do que bloquear um em outro nó que tenha que parar para pedir permissão o bloqueio mestre.

Como em muitos sistemas de arquivos, o alocador do GFS2 tenta manter bloqueios no mesmo arquivo perto um do outro para reduzir o movimento de cabeçalho de disco e aumentar o desempenho. Um nó que aloque travas à um arquivo, precisará provavelmente utilizar e bloqueiar o mesmo grupo de recurso para os novos blocos (a não ser que todos os blocos naquele grupo de recurso estejam em uso). O sistema de arquivo será executado mais rápido se o bloqueio mestre para o grupo de recurso que contém o arquivo, alocar seus blocos de dados (ou seja, é mais rápido que o nó, que abriu primeiro o arquivo, lide com a gravação dos novos blocos).

### 2.3.3. Pré-aloque, Se possível

Se os arquivos forem pré-alocados, as alocações de bloco podem ser evitadas e o sistema de arquivo podem funcionar com mais eficiência. Versões mais novas do GFS2 incluem a chamada de sistema `faallocate(1)`, a qual você pode utilizar para pré-alocar blocos de dados.

## 2.4. CONSIDERAÇÕES DE CLUSTER

Ao determinar o número de nós que seu sistema irá conter, note que existe uma negociação entre a alta disponibilidade e desempenho. Com um número maior de nós, se torna cada vez mais difícil escalar cargas de trabalho. Por esta razão, a Red Hat não suporta o uso do GFS2 para o sistema de arquivo do cluster, implementações maiores do que 16 nós.

A implementação de um sistema de arquivo cluster não é uma substituição "drop in" para uma implementação de nó único. Recomendamos que você permita um período de aproximadamente 8-12 semanas de testes em novas instalações para testar o sistema e certificar-se de que está funcionando no nível de desempenho requerido. Durante este período, qualquer problema de desempenho ou funcional pode ser solucionado e quaisquer dúvidas devem ser direcionadas à equipe de suporte da Red Hat.

Recomendamos que os clientes que estejam considerando a possibilidade de implementar clusters, peçam ao suporte da Red Hat para rever suas configurações antes da implementação para evitar qualquer problema de suporte possível mais tarde.

## 2.5. CONSIDERAÇÕES DE USO

Esta seção provê recomendações gerais sobre o uso do GFS2.

### 2.5.1. Opções de Montagem: `noatime` e `nodiratime`

Geralmente recomenda-se montar os sistemas de arquivo do GFS2 com os argumentos `noatime` e `nodiratime`. Isto permite que o GFS2 gaste menos tempo atualizando inodes de disco para cada acesso.

### 2.5.2. Opções de Ajuste do DLM: Aumentar Tamanhos da Tabela do DLM

O DLM utiliza diversas tabelas para gerenciar, coordenar e passar informações de bloqueio entre nós no cluster. Aumentar o tamanho das tabelas do DLM pode aumentar o desempenho. No Red Hat Enterprise Linux 6.1, os tamanhos padrão destas tabelas foram aumentados, mas você pode aumentá-los manualmente com os seguintes comandos:

```
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
```

Estes comandos não são persistentes e não sobreviverão uma reinicialização, portanto você precisa adicioná-los a um dos scripts de inicialização e você precisa executá-los antes de montar qualquer sistema de arquivo do GFS2, ou as mudanças serão ignoradas silenciosamente.

Para mais informações detalhadas sobre o bloqueio de nó do GFS2, consulte o [Seção 2.9, “Bloqueio de Nó GFS2”](#).

### 2.5.3. Opções de Ajuste do VFS: Pesquisa e Experimentos

Assim como todos os sistemas de arquivo do Linux, o GFS2 fica no topo da camada chamado sistema de arquivo virtual (VFS). Você pode ajustar a camada VFS para aumentar o desempenho do GFS2 adjacente, utilizando o comando `sysctl(8)`. Por exemplo, os valores para `dirty_background_ratio` e `vfs_cache_pressure` podem ser ajustados dependendo da sua situação. Para buscar os valores atuais, use os comandos a seguir:

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

Os comandos a seguir ajustam os valores:

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

Você poderá mudar permanentemente os valores destes parâmetros, editando o arquivo `/etc/sysctl.conf`.

Para encontrar valores que mais se adequam aos seus casos de uso, pesquise as diversas opções do VFS e experimente em um cluster teste antes de implementar em produção total.

### 2.5.4. SELinux: Evite SELinux no GFS2

O Security Enhanced Linux (SELinux) é altamente recomendado para razões de segurança na maioria dos casos, mas não é suportado para uso com o GFS2. O SELinux armazena informações utilizando atributos estendidos sobre todos os objetos de sistema de arquivo. Ler, gravar e manter estes



atributos estendidos é possível, mas diminui a velocidade do GFS2 consideravelmente. Você precisa desligar o SELinux nos sistemas de arquivo GFS2.

### 2.5.5. Configurando o NFS sob o GFS2.

Devido à complexidade adicionada do subsistema de bloqueio do GFS2 e sua natureza em cluster, configurar o NFS sob GFS2 requer precauções e configuração muito cuidadosa. Esta seção descreve os cuidados que deve tomar ao configurar um serviço NFS sob um sistema de arquivo GFS2.



#### ATENÇÃO

Se o sistema de arquivo GFS2 for NFS exportado e os aplicativos do cliente NFS usam os bloqueios POSIX, então você precisará montar o sistema de arquivo com a opção `locallocks`. O efeito pretendido deste, é forçar os bloqueios POSIX de cada servidor a serem locais: ou seja, que não estejam em cluster, independentes. (Existem vários problemas se o GFS2 tentar implementar os bloqueios POSIX a partir do NFS nos nós de um cluster). Para aplicativos que são executados em clientes NFS, os bloqueios POSIX localizados significam que dois clientes podem obter o mesmo bloqueio ao mesmo tempo se os dois clientes estiverem montando a partir de servidores diferentes. Se todos os clientes montarem NFS a partir de um só servidor, então o problema de servidores separados obtendo os mesmos bloqueios independentemente desaparece. Se você não estiver certo se deve montar seu sistema de arquivo com a opção `locallocks`, você não deve utilizar a opção; é sempre mais seguro que os bloqueios funcionem em uma base de cluster.

Além das considerações de bloqueio, você deve considerar os seguintes fatores ao configurar um serviço NFS sob um sistema de arquivo GFS2.

- A Red Hat suporta somente as configurações do Red Hat High Availability Add-On utilizando o NFSv3 com bloqueio em uma configuração ativa/passiva com as seguintes características:
  - O sistema de arquivo backend é um sistema de arquivo GFS2 executado em um cluster de 2 à 16 nós.
  - Um servidor do NFSv3 é definido como um serviço exportando todo o sistema de arquivo GFS2 de um único nó de cluster por vez.
  - O servidor de NFS pode falhar de um nó de cluster para outro (configuração ativa/passiva)
  - Nenhum acesso ao sistema de arquivo GFS2 é permitido exceto através do servidor NFS. Isto também inclui ambos acessos de sistema de arquivo GFS2 locais como o acesso através do Samba ou Samba em Cluster.
  - Não existe suporte de cota de NFS no sistema.

Esta configuração provê HA para o sistema de arquivo e reduz o tempo de baixa do sistema pois um nó falho não resulta no requerimento da execução do comando `fsck` ao falhar o servidor NFS de um nó para outro.

- A opção NFS `fsid=` é obrigatória para exportações de NFS do GFS2.

- Caso surja o problema com seu cluster (por exemplo, o cluster não possui quórum e fencing não é bem sucedido), os volumes lógicos em cluster e o sistema de arquivo GFS2 serão congelados e não será possível obter nenhum acesso até que o cluster tenha quórum. Você deve considerar a possibilidade ao determinar se uma solução de failover simples, tal como a definida neste procedimento, é a mais apropriada para seu sistema.

### 2.5.6. Serviço de Arquivo do Samba (SMB ou Windows) sob GFS2

Desde o lançamento do Red Hat Enterprise Linux 6.2 você pode utilizar o serviço de arquivo do Samba (SMB ou Windows) de um sistema de arquivo GFS2 com CTDB, que permite configurações ativa/ativa. Para mais informações sobre as configurações do Clustered Samba, veja o documento *Cluster Administration*.

Acesso simultâneo aos dados do compartilhamento do Samba, de fora do Samba, não é suportado. Não existe suporte atualmente para empréstimos do cluster GFS2, o qual reduz a velocidade do serviço de arquivo do Samba.

## 2.6. BACKUPS DO SISTEMA DE ARQUIVO

É importante fazer backups regulares de seu sistema de arquivo GFS2 em caso de emergência, não importando o tamanho do seu sistema de arquivo. Muitos administradores de sistema se sentem seguros porque são protegidos pelo RAID, multipath, espelhamento, snapshots e outras formas de redundância, mas tal segurança não existe.

Pode ser um problema para criar um backup pois o processo de salvamento de um nó ou configuração de nós geralmente envolve leitura de um sistema de arquivo todo na sequência. Se isto for feito de um nó único, este nó irá reter todas as informações no cache até que os outros nós no cluster comecem a requisitar bloqueios. Executar este tipo de programa de backup enquanto o cluster estiver em operação irá impactar o desempenho de forma negativa.

Retirar os caches depois que o backup estiver concluído reduz o tempo requerido por outros nós de reter a propriedade de seus bloqueios/caches de cluster. Isto ainda não é ideal, pois os outros nós irão interromper o processo de cache de dados no qual estavam antes do processo de backup ter iniciado. Você pode retirar caches utilizando o seguinte comando após a conclusão do backup:

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

É mais rápido se cada nó de um cluster salvar seus próprios arquivos, pois assim a tarefa é dividida entre os nós. Você deve ser capaz de concluir isto com um script que utiliza o comando `rsync` em diretórios de nó específico.

A melhor forma de fazer o backup do GFS2 é criar um snapshot de hardware no SAN, apresentar o snapshot a outro sistema e fazer o backup lá. O sistema de backup deve montar o snapshot com o `-o lockproto=lock_nolock`, pois ele não estará em um cluster.

## 2.7. CONSIDERAÇÕES DO HARDWARE

Você deve considerar os seguintes fatores de hardware ao implementar um sistema de arquivo do GFS2.

- Utilizar Opções de Armazenamento com Maior Qualidade

O GFS2 pode operar em opções de armazenamento compartilhado mais baratas, tais como iSCSI ou Canal de Fibra sob Ethernet (FCoE), mas você obterá um melhor desempenho se você comprar armazenamento de maior qualidade com maior capacidade de cache. A Red Hat

realiza a maioria dos testes de qualidade, sanidade e desempenho em armazenamento de SAN com a interconexão do Fibre Channel. Como via de regra, é sempre melhor implementar algo que tenha sido testado primeiro.

- Equipamento de Rede de Teste Antes da Implementação

O equipamento de rede mais rápido com maior qualidade faz com que a comunicação de cluster e GFS2 funcionem com maior velocidade e maior confiança. No entanto, você não precisa comprar o hardware mais caro. Alguns dos interruptores de redes mais caros têm problemas em passar pacotes de multicast, que são usados para passar os bloqueios (flocks) do `fcntl`, onde a mercadoria de interruptores de rede mais baratos é muitas vezes mais rápido e mais confiável. Geralmente é uma boa prática tentar equipamentos antes de implementá-los em produção total.

## 2.8. PROBLEMAS DE DESEMPENHO: VERIFIQUE O RED HAT CUSTOMER PORTAL

Para obter informações sobre as melhores práticas de implementação e atualização dos clusters do Red Hat Enterprise Linux, utilizando o Complemento de Alta Disponibilidade e Sistema de Arquivo Global da Red Hat 2 (GFS2), consulte o artigo "Red Hat Enterprise Linux Cluster, High Availability, e GFS Deployment Best Practices" no Red Hat Customer Portal em <https://access.redhat.com/site/articles/40051>.

## 2.9. BLOQUEIO DE NÓ GFS2

Para conseguir o melhor desempenho de um sistema de arquivos GFS2, é muito importante entender algumas teorias básicas de suas operações. Um sistema de arquivo de nó único é implementado junto com um cache, a razão do qual é eliminar a latência de acessos ao disco quando usar dados frequentemente requeridos. No Linux o cache de página (e historicamente o cache de buffer) fornece essa função de cache.

Com o GFS2, cada nó tem seu próprio cache de página que pode conter algumas porções de dados no disco. O GFS2 usa um mecanismo de bloqueio chamado *glocks* (pronunciado em inglês gee-locks) para manter a integridade do cache entre os nós. O subsistema glock fornece uma função de gerenciamento de cache que é implementada usando o *gerenciador de bloqueio distribuído* (DLM - distributed lock manager) como a camada de comunicação subjacente.

Os glocks fornecem proteção para o cache em uma base por nó, para que então haja um bloqueio por inode que é usado para controlar a camada de cache. Se esse glock é concedido para um modo compartilhado (DLM lock mode: PR) então os dados sob esse glock podem ser colocados em cache em um ou mais nós ao mesmo tempo, que então todos os nós possam ter acesso local aos dados.

Se o glock é concedido em modo exclusivo (DLM lock mode: EX) então somente um nó único pode fazer o cache dos dados sobre este glock. Este modo é usado por todas as operações que modificam os dados (tais como uma chamada de sistema `write`).

Se um outro nó solicita um glock que não pode ser feito imediatamente, então o DLM envia uma mensagem ao nó ou nós que atualmente possuem os glocks bloqueando a nova solicitação para pedir aos nós para remover seus bloqueios. Removendo glocks pode ser (pelo padrão da maioria dos sistemas operacionais) um longo processo. Removendo um glock compartilhado requer somente que o cache seja invalidado, que é relativamente rápido e proporcional ao tamanho dos dados em cache.

Despejar (drop) um glock exclusivo requer uma liberação de log e escrever de volta quaisquer dados alterados no disco, seguido pela invalidação como por glock compartilhado.

A diferença entre um sistema de arquivos de nó único e o GFS2 então, é que um sistema de arquivos de nó único possui um cache único e o GFS2 possui um cache separado em cada nó. Em ambos os casos, a latência para acessar os dados em cache é como uma ordem similar de magnitude mas a latência para acessar dados sem cache é muito maior no GFS2 se outro nó tiver anteriormente feito o cache dos mesmos dados.



## NOTA

Devido à maneira na qual o cache do GFS2 é implementada, o melhor desempenho é obtido quando qualquer umas das situações seguintes acontecem:

- Um inode é usado em uma forma de leitura somente em todos os nós
- Um inode é escrito ou modificado a partir de um nó único somente.

Note que inserindo e removendo entradas de um diretório durante criação e exclusão de arquivos conta como escrita no diretório inode.

É possível quebrar esta regra desde que ela seja quebrada relativamente com infrequencia. Ignorando esta regra com frequencia resultará em uma severa penalidade no desempenho.

Se você usar `mmap()` em um arquivo em GFS2 com mapeamento de escrita e leitura, mas somente leitura é feita, se contará apenas como leitura. No GFS entretanto, se conta como uma escrita, então o GFS2 é muito mais escalável com o `mmap()` de E/S.

Se você não ajudar o parâmetro `noatime mount`, então leituras também resultarão em escritas para atualizar o arquivo timestamps. Nós recomendamos que todos usuários GFS2 deveriam ser montados com `noatime` ao menos que eles tenham um requerimento específico para o `atime`.

### 2.9.1. Problemas com o Bloqueio de Posix

Ao utilizar o bloqueio de Posix, você deve levar em consideração:

- Ao utilizar o Flocks, você irá gerar processamento mais rápido do que se utilizar os bloqueios de Posix.
- Programas que utilizam os bloqueios Posix no GFS2 devem evitar utilizar a função `GETLK`, pois em um ambiente em cluster, o ID do processo pode ser para um nó diferente no cluster.

### 2.9.2. Ajuste de Desempenho com GFS2

É possível alterar a maneira em que um aplicativo problemático armazena seus dados para ganhar uma vantagem de desempenho considerável.

Um exemplo típico de aplicativo problemático é um servidor de emails. Estes são frequentemente definidos com um diretório de spool contendo arquivos para cada usuário (`mbox`), ou com um diretório para cada usuário contendo um arquivo para cada mensagem (`maildir`). Quando os pedidos chegam via IMAP, a preparação ideal é dar cada usuário uma afinidade a um nó particular. Desta maneira, seus pedidos para visualizar e deletar mensagens de email tenderão a ser servidos a partir do cache nesse nó. Obviamente se esse nó falhar, a sessão então pode ser reiniciada em um nó diferente.

Quando um email chega via SMTP, então novamente os nós individuais podem ser configurados para passar um email de determinado usuário para um nó em particular por padrão. Se o nó padrão não

estiver ativo, então a mensagem pode ser salva diretamente no pool de mail do usuário através no nó de recepção. Novamente esta estrutura pretende manter um conjunto determinado de cache de arquivos em somente um nó em situações normais, mas permitir acesso direto no caso de uma falha do nó.

Esta configuração permite o melhor uso da página de cache do GFS2 and também tornar falhas mais transparentes à aplicação, seja `imap` ou `smtp`.

O backup é muitas vezes uma área complicada. Novamente, se possível é recomendado fazer um backup do conjunto de execução de cada nó diretamente do nó que está fazendo cache desses conjuntos de inodes em particular, e isso parece coincidir com um aumento em tempo de resposta de uma aplicação rodando em GFS2, pois há uma boa chance que o cluster pode não estar fazendo o uso mais eficiente da página de cache.

Obviamente, se você possui permissão para parar a aplicação para realizar um backup, então isto não será um problema. De outra maneira, se um backup é executado a partir de um único nó, então depois dele ter completado uma grande parte do sistema de arquivos, ele cache nesse nó, com uma penalidade no desempenho pelo motivo de ter subsequente acessos dos outros nós. Isso pode ser aliviado a um certo ponto removendo o cache de página VFS no nó de backup depois do backup ter completado com o seguinte comando:

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

Entretanto, esta não é uma solução tão boa quanto certificar que o conjunto de execução em cada nó é tanto compartilhado, podendo ser lido somente pelo cluster ou acessado intensamente de um nó único.

### 2.9.3. Resolução de Problemas do Desempenho do Despejo de Bloqueios GFS2

Se o desempenho de seu cluster está comprometido por causa do uso ineficiente do cache do GFS2, você pode ver grandes e crescentes períodos de espera de E/S. Você pode usar as informações de despejo de bloqueio do GFS2 para determinar a causa do problema.

Esta seção provê uma visão geral do despejo do Bloqueio GFS2. Para uma descrição mais completa do despejo do Bloqueio GFS2, veja [Apêndice C, Os tracepoints e o Arquivo debugfs glocks](#)

As informações de despejo de bloqueio do GFS2 podem ser colhidas a partir do arquivo `debugfs` que pode ser encontrado no seguinte caminho, assumindo que o `debugfs` está montado em `/sys/kernel/debug/`:

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

O conteúdo do arquivo é uma série de linhas. Cada linha iniciando com G: representa um glock e as linhas seguintes, iniciadas com um único espaço, representam um ítem de formação relacionadas ao glock em questão nesse arquivo.

A melhor maneira de usar o arquivo `debugfs` é usar o comando `cat` para fazer uma cópia do conteúdo completo do arquivo (isso pode levar um longo tempo se você tiver uma grande quantidade de RAM e vários inodes em cache) enquanto o aplicativo estiver passando por problemas e então buscando através dos dados resultantes em uma data posterior.



## NOTA

Pode ser útil fazer duas cópias do arquivo `debugfs`, uma poucos segundos ou mesmo um ou dois minutos depois da outra. Comparando as informações do detentor nos dois arquivos, relacionados ao mesmo número glock, você poderá dizer se a carga de trabalho está fazendo progresso (que está somente lenta) ou se tornou parada (que é sempre um bug e deve ser avisado ao suporte Red Hat imediatamente).

Linhas no arquivo `debugfs` iniciando com o H: (holders - detentores) representam pedidos de bloqueios tanto concedidos ou em espera. O sinalizador 'W' se refere ao pedido em espera, o 'H' se refere a um pedido concedido. Os glocks que possuem um grande número de pedidos em espera são provavelmente aqueles que estão passando por determinadas contenções.

[Tabela 2.1, “Sinalizadores Glock”](#) mostra o significado de sinalizações de glock diferentes e [Tabela 2.2, “Sinalizadores do proprietário glock”](#) mostra os significados das sinalizações do proprietário diferentes do glock na ordem em que eles aparecem nos despejos do glock.

**Tabela 2.1. Sinalizadores Glock**

Sinaliz ação	Nome	Significado
b	Bloqueantes	Válido quando a sinalização bloqueada é definida, e indica que a operação, que foi requisitada do DLM, poderá bloquear. Esta sinalização é removida para operações de rebaixamento e para "tentativas" de bloqueios. O propósito desta sinalização é permitir reunir estatísticas do tempo de resposta de DLM independentes a partir do tempo tomado por outros nós para bloqueios de rebaixamento.
d	Rebaixamento pendente	Um pedido (remoto) de rebaixamento deferido
D	Rebaixamento	Um pedido de rebaixamento (local ou remoto)
f	Descarga de Log	O log precisa ser enviado antes de liberar esse glock
F	Congelado	Responde a partir de nós remotos ignorados - recuperação está em progresso. Esta sinalização não é relacionada ao congelamento de sistema de arquivo, o qual utiliza um mecanismo diferente, mas é utilizado somente na recuperação.
i	Invalidação em progresso	No processo de estar invalidando páginas sob este glock.
l	Inicial	Ajuste quando o bloqueio DLM é associado com este glock
l	Bloqueado	O glock está no processo de mudança de estado
L	LRU	Definido quando o glock está na lista do LRU

Sinalização	Nome	Significado
o	Object	Definido quando o glock é associado com um objeto (ou seja, um inode para glocks tipo 2, e um grupo de recurso para glocks tipo 3)
p	Rebaixamento em progresso	O glock está no processo de responder a um pedido de rebaixamento
q	Em fila	Definido quando um proprietário é enfileirado em um glock e limpo quando um glock é mantido, mas não houver nenhum outro proprietário. Usado como parte do algoritmo que calcula um tempo mínimo de espera para um glock.
r	Resposta pendente	Resposta recebida do nó remoto está aguardando processamento
y	Sujo	Os dados precisam descarregar para o disco antes de liberar este glock

Tabela 2.2. Sinalizadores do proprietário glock

Sinalização	Nome	Significado
a	Async	Não espere pelo resultado glock (fará um conjunto para enviar depois)
A	Qualquer	Qualquer modo de bloqueio compatível é aceitável
c	Sem cache	Quando desbloqueado, rebaixe o bloqueio DLM imediatamente
e	Não expirar data	Ignore pedidos de cancelamento de bloqueios subsequentes
E	exato	Deve ter um modo de bloqueio exato
F	Primeiro	Ajuste quando o proprietário é o primeiro a ser concedido para este bloqueio
H	Detentor	Indica que o bloqueio requerido é concedido
p	Prioridade	Enfileire os proprietários no início da fila
t	Tentar	Uma "tentativa" de bloqueio
T	Tentar 1CB	Uma "tentativa" que envia um callback

Sinalização	Nome	Significado
W	Espera	Ajuste enquanto espera por um pedido para completar

Ter identificado um glock que está causando um problema, o próximo passo é encontrar qual inode ele está relacionado. O número do glock (n: na linha G:) indica isso. Isso é na forma *tipo/número* e se o *tipo* é 2, então o glock é um glock inode e o *number* é um número inode. Para descobrir o inode, você pode então rodar `find -inum number` onde *número* é o número inode convertido do formato hex nos arquivos glocks em decimais.



#### NOTA

Se você executar o `find` em um sistema de arquivos quando estiver passando por uma contenção de bloqueio, você poderá tornar o problema pior. É uma boa idéia parar a aplicação antes de rodar o `find` quando você estiver buscando por inode sustentados.

A [Tabela 2.3, “Tipos de Glock”](#) mostra os significados dos diferentes tipos de glock.

**Tabela 2.3. Tipos de Glock**

Digite o número	Tipo de bloqueio	Use
1	Trans	Bloqueio de transação
2	Inode	Dados e metadados inode
3	Rgrp	Metadados de grupos de recursos
4	Meta	O superblock
5	lopen	Detecção mais próxima do último inode
6	Flock	<code>flock(2)</code> syscall
8	Quota	Operações de quota
9	Diário	Diário mutex

Se o glock que foi identificado fosse de um tipo diferente, então é provável que seja de um tipo 3: (grupo de recursos). Se você ver números significantes de processos esperando por outros tipos de glock sob carregamentos normais, então por favor relate isso ao suporte Red Hat.

Se você ver um número de pedidos em espera enfileirados em um bloqueio de grupo de recursos, poderão haver algumas razões para isso. Uma é que existem um grande número de nós comparados ao número de grupos de recursos no sistema de arquivos. Outra é que o sistema de arquivos podem estar



quase cheios (requerendo, na média, buscas mais longas para blocos disponíveis). A situação em ambos os casos pode ser melhorada adicionado mais armazenamento e usando o comando `gfs2_grow` para expandir o sistema de arquivos.

## CAPÍTULO 3. INICIANDO

Este capítulo descreve os procedimentos para a configuração inicial do GFS2 e contém as seguintes seções:

- [Seção 3.1, “Tarefas de Pré-requisitos:”](#)
- [Seção 3.2, “Tarefas Iniciais de Configuração”](#)

### 3.1. TAREFAS DE PRÉ-REQUISITOS:

Você precisa completar as seguintes tarefas antes de configurar o Red Hat GFS2:

- Certifique-se de que você anotou as características chave dos nós de GFS2 (consulte o [Seção 1.2, “Antes de Configurar o GFS2”](#)).
- Certifique-se de que os relógios no nós GFS2 estão sincronizados. Recomenda-se que você utilize o software Network Time Protocol (NTP) fornecido com seu Red Hat Enterprise Linux.



#### NOTA

O relógio do sistema nos nós de GFS2 deve ter uma diferença de quinze minutos entre um e o outro para evitar atualizações desnecessárias de carimbo de data e hora de inode. Esta atualização desnecessária de carimbo de data e hora de inode causa um sério impacto no desempenho do cluster.

- Para você utilizar o GFS2 em um ambiente em cluster, você precisa configurar seu sistema para usar o Clustered Logical Volume Manager (CLVM), um conjunto de extensões de cluster para o Gerenciador de Volume Lógico. Para utilizar o CLVM, o software Red Hat Cluster Suite, incluindo o daemon do `c1vmd` devem estar em execução. Para mais informações sobre como utilizar o CLVM, veja *Logical Volume Manager Administration*. Para informações sobre como instalar e administrar o Red Hat Cluster Suite, veja *Cluster Administration*.

### 3.2. TAREFAS INICIAIS DE CONFIGURAÇÃO

A configuração inicial do GFS2 consiste nas seguintes tarefas:

1. Configurando Volumes Lógicos:
2. Produzindo um sistema de arquivos GFS2.
3. Montando sistemas de arquivos.

Siga estes passos para configurar o GFS2 inicialmente.

1. Usando o LVM, crie um volume lógico para cada sistema de arquivo do Red Hat GFS2.



#### NOTA

Você pode usar os scripts `init.d` inclusos no Red Hat Cluster Suite para automatizar a ativação e desativação dos volumes lógicos. Para mais informações sobre os scripts `init.d`, consulte *Configurando e Gerenciando um Red Hat Cluster*.

2. Crie sistemas de arquivo GFS2 em volumes lógicos criados no Passo 1. Escolha um nome único para cada sistema de arquivo. Para mais informações sobre como criar um sistema de arquivo GFS2, consulte o [Seção 4.1, “Criando um Sistema de Arquivo”](#).

Você pode usar qualquer um dos formatos a seguir para criar um sistema de arquivo GFS2 em cluster:

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

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

Para mais informações sobre como montar um sistema de arquivo GFS2, consulte o [Seção 4.2, “Montagem de Sistema de Arquivo”](#).

3. Em cada nó, monte os sistemas de arquivos GFS2. Para mais informações sobre como montar um sistema de arquivo GFS2, consulte o [Seção 4.2, “Montagem de Sistema de Arquivo”](#).

Uso do comando:

```
mount BlockDevice MountPoint
```

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

A opção de montagem `-o acl` permite manipular ACLs de arquivos. Se um sistema de arquivo for montado sem a opção de montagem `-o acl` os usuários poderão visualizar as ACLs ( com o comando `getfacl`), porém, não poderão configurá-los ( com o comando `setfacl`).



#### NOTA

Você pode usar scripts `init.d` incluídos no Complemento de Alta Disponibilidade Red Hat para automatizar a montagem e desmontagem de sistemas de arquivo GFS2.

## CAPÍTULO 4. GERENCIANDO O GFS2

Este capítulo descreve as tarefas e comandos para gerenciar o GFS2 e consiste nas seguintes seções:

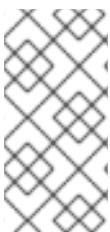
- [Seção 4.1, “Criando um Sistema de Arquivo”](#)
- [Seção 4.2, “Montagem de Sistema de Arquivo”](#)
- [Seção 4.3, “Desmontando o Sistema de Arquivo”](#)
- [Seção 4.5, “Gerenciamento de Cota do GFS2”](#)
- [Seção 4.6, “Aumentando um Sistema de Arquivo”](#)
- [Seção 4.7, “Adicionando Diários ao Sistema de Arquivo”](#)
- [Seção 4.8, “Diário de Dados”](#)
- [Seção 4.9, “Configurando atualizações do `atime`”](#)
- [Seção 4.10, “Suspendendo a Atividade em um Sistema de Arquivo”](#)
- [Seção 4.11, “Reparando um Sistema de Arquivo”](#)
- [Seção 4.12, “Montagens Bind e Nomes de Caminho de Contexto Dependente.”](#)
- [Seção 4.13, “Monta Binds e Organiza a Montagem de Sistema de Arquivo”](#)
- [Seção 4.14, “A Função de Retirada do GFS2”](#)

### 4.1. CRIANDO UM SISTEMA DE ARQUIVO

Você pode criar um sistema de arquivo do GFS2 com o comando `mkfs.gfs2`. Você pode também usar o comando `mkfs` com a opção `-t gfs2` especificada. Um sistema de arquivo é criado em um volume LVM ativado. As seguintes informações são necessárias para executar o comando `mkfs.gfs2`:

- Protocolo de bloqueio/ nome do módulo (o protocolo de bloqueio para um cluster é `lock_dlm`)
- Nome do Cluster (quando estiver sendo executado como parte de uma configuração de cluster)
- Número de diários (um diário necessário para cada nó que possa montar o sistema de arquivo)

Quando criar um sistema de arquivos GFS2, você pode usar o comando `mkfs.gfs2` diretamente ou você pode usar o comando `mkfs` com o parâmetro `-t` especificando um sistema de arquivos do tipo `gfs2`, seguido pelas opções do sistema de arquivos `gfs2`.



#### NOTA

Você não poderá reduzir mais o tamanho do sistema de arquivo, uma vez que criado com o comando `mkfs.gfs2`. No entanto, você poderá aumentar o tamanho de um sistema de arquivo existente com o comando `gfs2_grow`, conforme descrito no [Seção 4.6, “Aumentando um Sistema de Arquivo”](#).

### 4.1.1. Uso

Quando criar um sistema de arquivos GFS2, você pode usar quaisquer dos seguintes formatos:

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

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

Quando criar um sistema de arquivos GFS2 local, você pode usar quaisquer dos seguintes formatos:



#### NOTA

A Red Hat não suporta o uso do GFS2 como um sistema de arquivo com nó único, para a liberação do Red Hat Enterprise Linux 6.

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

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



#### ATENÇÃO

Tenha a certeza de que você esteja familiarizada com o uso dos parâmetros *LockProtoName* e *LockTableName*. O uso impróprio dos parâmetros *LockProtoName* e *LockTableName* pode fazer com que o sistema de arquivo ou espaço de bloqueio sejam corrompidos.

#### *LockProtoName*

Especifica o nome do protocolo de bloqueio a usar. O protocolo de bloqueio para um cluster é `lock_dlm`.

#### *LockTableName*

Este parâmetro é especificado para o sistema de arquivos do GFS2 em uma configuração de cluster. Ele possui duas partes separadas por uma vírgula (sem espaços) como se segue:

***ClusterName* : *FSName***

- ***ClusterName***, o nome do Red Hat cluster para o qual o sistema de arquivo GFS2 está sendo criado.
- ***FSName***, o nome do sistema de arquivos pode ter entre 1 e 16 caracteres. O nome deve ser único para todos os os sistemas de arquivos `lock_dlm` sobre o cluster e para todos sistemas de arquivos (`lock_dlm` e `lock_nolock`) em cada nó local.

#### *Número*

Especifica o número de diários a serem criados pelo comando `mkfs.gfs2`. É necessário um diário

para cada nó que monta o sistema de arquivos. Para os sistemas de arquivo do GFS2, você pode adicionar diários mais tarde sem aumentar o sistema de arquivos, conforme descrito na [Seção 4.7, “Adicionando Diários ao Sistema de Arquivo”](#).

### **BlockDevice**

Especifica um volume físico ou lógico.

## 4.1.2. Exemplos

Neste exemplo, o `lock_dlm` é o protocolo de bloqueio que o sistema de arquivo usa, desde que seja um sistema de arquivo em cluster. O nome do cluster é `alpha`, e o nome do sistema de arquivo é `mydata1`. O sistema de arquivo contém oito diários e é criado em `/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
```

Neste exemplo, um segundo sistema de arquivo `lock_dlm` é criado, o qual pode ser usado no cluster `alpha`. O nome do sistema de arquivo `mydata2`. O sistema de arquivo contém oito diários e é criado em `/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
```

## 4.1.3. Opções Completas

[Tabela 4.1, “Opções de Comando: `mkfs.gfs2`”](#) descreve as opções de comando `mkfs.gfs2` (sinalizadores e parâmetros).

**Tabela 4.1. Opções de Comando: `mkfs.gfs2`**

Sinalizador	Parâmetro	Descrição
<code>-c</code>	<i>Megabytes</i>	Estabelece o tamanho inicial de cada arquivo de mudança de cota de diário para <i>Megabytes</i> .
<code>-D</code>		Possibilita o resultado depurado.
<code>-h</code>		Ajuda. Exibe as opções disponíveis
<code>-J</code>	<i>MegaBytes</i>	Especifica o tamanho do diário em megabytes. O tamanho padrão do diário é de 128 megabytes. O tamanho mínimo é de 8 megabytes. Diários maiores aumentam o desempenho, embora usem mais memória do que os diários menores.

Sinalizador	Parâmetro	Descrição
-j	<i>Número</i>	Especifica o número de diários a serem criados pelo comando <code>mkfs.gfs2</code> . É necessário um diário para cada nó que monta o sistema de arquivo. Se esta opção não for especificada, será criado um diário. Para os sistemas de arquivo do GFS2, você pode adicionar diários mais tarde sem aumentar o sistema de arquivos.
-O		Evita que o comando <code>mkfs.gfs2</code> peça uma confirmação antes de escrever o sistema de arquivo.
-p	<i>LockProtoName</i>	<p>Especifica o nome do protocolo de bloqueio a usar. Bloqueios reconhecidos de protocolos incluem:</p> <p><b>lock_dlm</b> – O módulo de bloqueio padrão, necessário para um sistema de arquivo de cluster.</p> <p><b>lock_nolock</b> – Usado quando o GFS2 estiver agindo como um sistema de arquivo local (somente um nó).</p>
-q		Silencioso. Não exibe nada.
-r	<i>MegaBytes</i>	Especifica o tamanho de grupos de recursos em megabytes. O tamanho mínimo de grupo de recurso é de 32 MB. O tamanho máximo de grupo de recurso é de 2048MB. Um tamanho grande de grupo de recurso pode aumentar o desempenho em sistemas de arquivo muito grandes. Se ele não for especificado, o <code>mkfs.gfs2</code> escolhe o tamanho de grupo de recurso baseado no mesmo tamanho do sistema de arquivo: sistemas de arquivo de tamanho comum serão de grupos de recursos de 256MB, e maiores sistemas de arquivos terão RGs maiores para um melhor desempenho.

Sinalizador	Parâmetro	Descrição
-t	<b>LockTableName</b>	<p>Um identificador único que especifica o campo da tabela de bloqueio quando você usar o protocolo <b>lock_dlm</b>. O protocolo <b>lock_nolock</b> não usa este parâmetro.</p> <p>Este parâmetro possui duas partes separadas por vírgulas (sem espaço) como se segue: <b>ClusterName:FSName</b>.</p> <p><b>ClusterName</b> é o nome do Red Hat cluster para o qual o sistema de arquivo do GFS2 está sendo criado. Somente os membros deste cluster podem usar este sistema de arquivo. O nome do cluster é estabelecido no arquivo <b>/etc/cluster/cluster.conf</b> via <b>Cluster Configuration Tool</b> e exibido no <b>Cluster Status Tool</b> no (GUI) do gerenciamento do cluster do Red Hat Cluster Suite.</p> <p><b>FSName</b>, o nome do sistema de arquivo pode ter entre 1 e 16 caracteres e o nome deve ser único entre todos os sistemas de arquivo no cluster.</p>
-u	<b>MegaBytes</b>	Especifica o tamanho inicial de cada arquivo de marca sem vínculo de diário.
-V		Exibe as informações sobre a versão do comando.

## 4.2. MONTAGEM DE SISTEMA DE ARQUIVO

Antes que você possa montar um sistema de arquivo do GFS2, o sistema de arquivo deve existir (consulte a [Seção 4.1, “Criando um Sistema de Arquivo”](#)), o volume onde o sistema de arquivo existe deve ser ativado, e os sistemas de bloqueio e agrupamento (clustering) deve ser iniciado (consulte o [Configurando e Gerenciando um Red Hat Cluster](#)). Após estes requerimentos serem atendidos você deve montar o sistema de arquivo do GFS2 como você faria em qualquer sistema de arquivo do Linux.





## NOTA

Tentar montar um sistema de arquivos GFS2 quando o Gerenciador de Cluster (**cman**) não tiver sido iniciado produz a seguinte mensagem de erro:

```
[root@gfs-a24c-01 ~]# mount -t gfs2 -o noatime
/dev/mapper/mpathp1 /mnt
gfs_control join connect error: Connection refused
error mounting lockproto lock_dlm
```

Para manipular ACLs de arquivo, você deve montar o sistema de arquivo com a opção de montagem do **-o acl**. Se um sistema de arquivo for montado sem a opção de montagem **-o acl**, usuários podem visualizar as ACLs (com o **getfacl**), mas não podem estabelecê-los (com **setfacl**).

### 4.2.1. Uso

#### Montagem Sem a Manipulação da ACL

```
mount BlockDevice MountPoint
```

#### Montagem Sem a Manipulação da ACL

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

#### **-o acl**

Opção do GFS2 específico para permitir a manipulação das ACLs do arquivo.

#### *BlockDevice*

Especifica o dispositivo de bloco onde o sistema de arquivo do GFS2 reside.

#### *MountPoint*

Especifica o diretório onde o sistema de arquivo GFS2 deve ser montado.

### 4.2.2. Exemplo

Neste exemplo, o sistema de arquivo do GFS2 em **/dev/vg01/lvol0** é montado no diretório **/mygfs2**.

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

### 4.2.3. Uso Completo

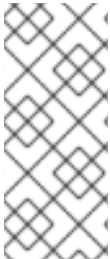
```
montar BlockDevice MountPoint -o option
```

O argumento **-o option** consiste das opções do GFS2 específico (consulte a [Tabela 4.2, “Opções de Montagem do GFS2 Específico”](#)) ou opções padrão aceitáveis **mount -o** do Linux, ou uma combinação de ambos. Parâmetros de *option* múltiplas são separadas por vírgulas e sem espaços.

**NOTA**

O comando `mount` é um comando do sistema Linux. Além de usar as opções do GFS2 específicas descritas nesta seção, você pode usar outras opções de comando padrão, `mount` (por exemplo `-r`). Para mais informações sobre outras opções de comando de `mount` do Linux, veja a página `man` do `mount` do Linux.

Tabela 4.2, “Opções de Montagem do GFS2 Específico” descreve os valores da `-o` option do GFS2 específico disponível, que pode ser passado para o GFS2 durante a montagem.

**NOTA**

Esta tabela inclui a descrição das opções que são usadas com os sistemas do arquivo local apenas. Perceba, no entanto, que na liberação do Red Hat Linux 6, a Red Hat não suporta o uso do GFS2 como um sistema de arquivo de nó único. A Red Hat continuará a suportar os sistemas de arquivo GFS2 de nó único para montagens de trechos para sistemas de arquivo com cluster (por exemplo, para propósitos de back-ups).

Tabela 4.2. Opções de Montagem do GFS2 Específico

Opções	Descrição
<code>acl</code>	Permite manipular ACLs de arquivo. Se um sistema de arquivo for montado sem a opção de montagem <code>acl</code> os usuários podem visualizar as ACLs (com <code>getfacl</code> ), mas não poderão estabelecê-las (com <code>setfacl</code> ).
<code>data=[ordered writeback]</code>	Quando o <code>data=ordered</code> for estabelecido, os dados de usuários modificados por uma transação, é enviado para o disco antes que a transação seja salva em um disco. Isto deve evitar que o usuário veja blocos não inicializados em um arquivo após um travamento. Quando o modo <code>data=writeback</code> for estabelecido, os dados do usuário serão escritos no disco logo após que for 'sujo'. Isto não oferece a mesma garantia de consistência quanto o modo <code>ordered</code> , mas deve ser um pouco mais rápido para algumas cargas de trabalho. O valor padrão é o modo <code>ordered</code> .
<div style="border: 1px solid black; padding: 5px; margin-bottom: 5px;"><code>ignore_local_fs</code></div> <p><b>Cuidado:</b> Esta opção <i>não</i> deve ser usada quando os sistemas de arquivos do GFS2 forem compartilhados.</p>	Força o GFS2 a tratar o sistema de arquivo como o sistema de arquivo multihost. Por padrão, o uso automático do <code>lock_nolock</code> ativa os sinalizadores <code>localcaching</code> e <code>localflocks</code> .

Opções	Descrição
<p><b>localflocks</b></p> <p><b>Cuidado:</b> Esta opção não deve ser usada quando os sistemas de arquivo do GFS2 forem compartilhadas.</p>	<p>Informa o GFS2 para deixar que a camada do VFS (sistema de arquivo virtual) faça todo o flock e fcntl. O sinalizador <b>localflocks</b> é ativado automaticamente pelo <b>lock_nolock</b>.</p>
<p><b>lockproto=LockModuleName</b></p>	<p>Permite que o usuário especifique qual protocolo de bloqueio usar com o sistema de arquivo. Se o <b>LockModuleName</b> não for especificado, o nome do protocolo de bloqueio é lido a partir do super bloco do sistema de arquivo.</p>
<p><b>locktable=LockTableName</b></p>	<p>Permite que o usuário especifique qual tabela de bloqueio usar com o sistema de arquivo.</p>
<p><b>quota=[off/account/on]</b></p>	<p>Ativa e desativa as cotas para um sistema de arquivo. Estabelecer as cotas para ficar no estado de <b>account</b> faz com que as estatísticas de uso por UID/GID para ser mantido corretamente pelo sistema de arquivo. Os valores de limite e avisos (warn) são ignorados. O valor de padrão é <b>off</b>.</p>
<p><b>errors=panic withdraw</b></p>	<p>Quando <b>errors=panic</b> forem especificados, os erros do sistema de arquivo causarão pânico no kernel. O comportamento padrão, que é o mesmo que especificar <b>errors=withdraw</b>, é para o sistema retirar o sistema de arquivo e fazê-lo inacessível até a próxima reinicialização. Em alguns casos o sistema pode continuar rodando. Consulte <a href="#">Seção 4.14, “A Função de Retirada do GFS2”</a>, para maiores informações sobre esta função de remoção do GFS2.</p>
<p><b>discard/nodiscard</b></p>	<p>Leva o GFS2 a gerar o "descarte" das solicitações para blocos que foram liberados. Ele podem ser usados pelo hardware adequado para implementar esquemas similares e configuração fina.</p>
<p><b>barrier/nobarrier</b></p>	<p>Leva o GFS2 a enviar barreiras de E/S quando esvaziando o diário. O valor padrão é <b>on</b>. Esta opção é automaticamente desligada caso o dispositivo adjacente não suporte as barreiras de E/S. O uso das barreiras de E/S com o GFS2 é altamente recomendável em todos os momentos, a não ser que o dispositivo do bloco seja designado de forma que ele não perca seu conteúdo de cache gravado (por exemplo, caso esteja acionado um UPS ou ele não possuir um cache gravado).</p>

Opções	Descrição
<b>quota_quantum=secs</b>	Define o número de segundos para qual uma mudança na informação de quota pode deixar em um nó antes de ser escrito no arquivo de quota. Esta é a maneira preferida para definir este parâmetro. O valor é um número inteiro de segundos maiores que zero. O padrão é 60 segundos. Definições menores resultam em atualizações mais rápidas da informação de quota lazy e probabilidade menor de alguém exceder sua quota. Definições mais longas fazem das operações do sistema de arquivos envolvendo quotas mais rápidas e mais eficientes.
<b>statfs_quantum=secs</b>	A configuração do <b>statfs_quantum</b> para 0 é a maneira preferida de determinar a versão mais lenta do <b>statfs</b> . O valor padrão é 30 segundos, o qual determina o período máximo de tempo antes das alterações <b>statfs</b> serem sincronizadas ao arquivo <b>statfs</b> mestre. Isto pode ser ajustado para permitir valores <b>statfs</b> mais rápidos, menos precisas ou valores menores e mais precisos.
<b>statfs_percent=value</b>	Fornece uma vinculação na alteração de porcentagem máxima na informação <b>statfs</b> de um local básico antes de ser sincronizado de volta ao arquivo <b>statfs</b> mestre, mesmo que o período de tempo não tenha excedido. Caso a configuração do <b>statfs_quantum</b> seja 0, esta configuração pode ser descartada.

### 4.3. DESMONTANDO O SISTEMA DE ARQUIVO

O sistema de arquivo do GFS2 pode ser desmontado da mesma forma como qualquer sistema de arquivo do Linux – usando o comando **umount**.



#### NOTA

O comando **umount** é o comando de sistema Linux. As informações sobre este comando podem ser encontradas na página man do comando **umount**.

#### 4.3.1. Uso

```
umount MountPoint
```

##### **MountPoint**

Especifica o diretório onde o sistema de arquivos do GFS2 foi montado.

### 4.4. CONSIDERAÇÕES ESPECIAIS AO MONTAR OS SISTEMAS DE ARQUIVO GFS2

Sistemas de arquivos GFS2 que foram montados manualmente em vez de automaticamente através de

uma entrada no arquivo `fstab` não serão conhecidos ao sistema quando os sistemas de arquivos são desmontados no desligamento do sistema. Como resultado, o script GFS2 não desmontará o sistema de arquivos GFS2. Depois do script de desligamento do GFS2 é executado, o processo de desligamento normal termina todos os processos do usuário remanescentes, incluindo a infra estrutura do cluster e tentativas de desmontar o sistema de arquivos. Esta desmontagem falhará sem a infraestrutura do cluster e o sistema parará.

Para evitar que o sistema trave quando os sistemas de arquivo GFS2 forem desmontados, você deverá realizar o seguinte:

- Sempre use uma entrada no arquivo `fstab` para montar o sistema de arquivo GFS2.
- Caso o sistema de arquivo GFS2 tenha sido montado manualmente com o comando `mount`, certifique-se de desmontar o sistema de arquivo manualmente com o comando `umount`, antes de re-iniciar ou encerrar o sistema.

Caso o seu sistema travar enquanto seja desmontado durante o encerramento do sistema sob estas circunstâncias, execute a reinicialização do hardware. Dificilmente os dados serão perdidos desde que o sistema de arquivo seja sincronizado anteriormente no processo de encerramento.

## 4.5. GERENCIAMENTO DE COTA DO GFS2

As quotas do sistema de arquivos são usadas para limitar a quantidade do espaço do sistema de arquivos que um usuário ou grupo pode usar. Um usuário ou grupo não possui um limite de quota até que um seja estabelecido. Quando um sistema de arquivos GFS2 estiver montado com a opção `quota=on` ou `quota=account`, o GFS2 mantém registro do espaço do usuário usado para cada usuário e grupo mesmo onde não há limites impostos. O GFS2 atualiza a informação de quotas de uma maneira transacional que onde travamentos do sistema não requerem uso de quotas para serem reconstruídos.

Para evitar uma redução de desempenho, um nó do GFS2 sincroniza atualizações para o arquivo de cota somente diariamente. A conta de cotas "difusas" (fuzzy) pode permitir que usuários ou grupos excedam um pouco do limite estabelecido. Para minimizar isto, o GFS2 reduz o período de sincronização, de forma dinâmica, quando um limite de cota "hard" é procurado.



### NOTA

A partir do lançamento do Red Hat Enterprise Linux 6.1, o GFS2 suporta a facilidade de quotas Linux padrões. Para usar isto você precisará instalar o RPM de `quota`. Esta é a maneira preferida para administrar quotas no GFS2 e deve ser usada para todas as novas implementações do GFS2 usando quotas. Esta seção documenta o gerenciamento de quotas GFS2 usando estas facilidades.

Para versões anteriores do Red Hat Enterprise Linux, o GFS2 requisitava o comando `gfs2_quota` para gerenciar quotas. Para informações sobre o uso do comando `gfs2_quota`, veja o [Apêndice A, Gerenciamento de Quota do GFS2 com o comando `gfs2\_quota`](#).

### 4.5.1. Configurando Quotas de Disco

Para implementar quotas de disco, use os seguintes passos:

1. Configure quotas em modo enforcement ou accounting
2. Inicialize o banco de dados de quota com a informação de uso de bloco atual.

3. Atribua políticas de quota (No modo accounting, estas políticas não são impostas).

Cada um destes passos é discutido em detalhes nas seguintes seções.

#### 4.5.1.1. Definindo Quotas no Modo Enforcement ou Accounting

Nos sistemas de arquivo do GFS2, quotas são desativada por padrão. Para ativar as quotas para um sistema de arquivos, monte o sistema de arquivos com a opção `quota=on` especificada.

É possível manter o controle do uso do disco e manter a conta de conta para todos os usuários e grupo sem impor o limite e valores warn. Para fazer isto, monte o sistema de arquivo com a opção `quota=account` especificada.

##### 4.5.1.1.1. Uso

Para montar um sistema de arquivos com as quotas ativadas, monte o sistema de arquivos com a opção `quota=on` especificada.

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

Para montar um sistema de arquivos as quotas accounting mantidas, mesmo que o limite de quotas não estiver imposto, monte do sistema de arquivos com a opção `quota=account` especificada.

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

Para montar um sistema de arquivos com quotas desativadas, monte o sistema de arquivos com a opção `quota=off` especificada. Esta é a configuração padrão.

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

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

**on** - Especifica que as quotas estão habilitadas quando o sistema de arquivos é montado.

**off** - Especifica que quotas estão desabilitadas quando o sistema de arquivos estiver montado.

**account** - Especifica que o usuário e as estatísticas de uso do grupo são mantidas pelo sistema de arquivos, mesmo que os limites de quota não são impostos.

#### ***BlockDevice***

Especifica o dispositivo de bloco onde o sistema de arquivo do GFS2 reside.

#### ***MountPoint***

Especifica o diretório onde o sistema de arquivo GFS2 deve ser montado.

##### 4.5.1.1.2. Exemplos

Neste exemplo, o sistema de arquivos GFS2 no `/dev/vg01/lvol0` é montado no diretório `/mygfs2` com quotas ativadas.

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

Neste exemplo, o sistema de arquivos GFS2 no `/dev/vg01/lvol0` é montado no diretório `/mygfs2` com a quota `accounting` mantida mas não imposta.

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

#### 4.5.1.2. Criando os Arquivos de Banco de Dados de Quotas

Depois que cada sistema de arquivos com quotas ativadas é montado, o sistema é capaz de funcionar com quotas de disco. Entretanto, o próprio sistema de arquivos não está pronto ainda para suportar quotas. O próximo passo é rodar o comando `quotacheck`

O comando `quotacheck` examina sistemas de arquivos com quotas habilitadas e constrói uma tabela de uso de disco atual por arquivo de sistema. A tabela é então usada para atualizar a cópia de uso do disco do sistema operacional. Além disso, os arquivos de quota de disco do sistema de arquivos são atualizados.

Para criar arquivos de quota no sistema de arquivos, use as opções `-u` e `-g` do comando `quotacheck`; ambas opções devem ser especificadas para quotas de usuário e grupos a serem inicializados. Por exemplo, se as quotas estiverem ativadas para o sistema de arquivos `/home`, crie os arquivos no diretório `/home`:

```
quotacheck -ug /home
```

#### 4.5.1.3. Atribuindo Quotas por Usuário

O último passo é atribuir quotas de disco com o comando `edquota`. Note que se você montou seu sistema de arquivos no modo `accounting` (com a opção `quota=account` especificada), as quotas não são impostas.

Para configurar a quota para um usuário, como `root` no prompt de shell, execute o comando:

```
edquota username
```

Realize este passo para cada usuário que precisa de uma quota. Por exemplo, se uma quota estiver ativada no `/etc/fstab` para a partição `/home` (`/dev/VolGroup00/LogVol02` no exemplo abaixo) e o comando `edquota testuser` é executado, o seguinte é mostrado no editor padrão do sistema:

```
Disk quotas for user testuser (uid 501):
Filesystem          blocks      soft      hard      inodes      soft
hard
/dev/VolGroup00/LogVol02 440436      0         0
```



#### NOTA

O editor de texto definido pela variável de ambiente `EDITOR` é usada pelo `edquota`. Para mudar o editor, defina a variável de ambiente `EDITOR` em seu arquivo `~/.bash_profile` para o caminho completo do editor de sua preferência.

Se a primeira coluna é o nome de seu sistema de arquivos que tem uma quota ativada. A segunda coluna mostra quantos blocos o usuário está atualmente usando. As próximas duas colunas são usadas para definir limites de bloco `soft` e `hard` para o uso do sistema de arquivos.

O limite de block soft define a quantidade máxima do espaço de disco que pode ser usada.

O limite de bloco hard é a quantidade máxima absoluta de espaço de disco que um usuário ou grupo pode usar. Uma vez este limite é alcançado, mais nenhum espaço de disco pode ser usado.

O sistema de arquivos GFS2 não mantém quotas para o inodes, então estas colunas não se aplicam aos sistemas de arquivos GFS2 e ficarão em branco.

Se qualquer dos valores são configurados para 0, o limite não é definido. Em um editor de texto, mude os limites desejados. Por exemplo:

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

Para verificar que a quota para o usuário foi definida, use o comando:

```
quota testuser
```

#### 4.5.1.4. Atribuindo Quotas por Grupo

Quotas também podem ser atribuídas por grupo. Note que se você montou seu sistema de arquivos no modo accounting (com a opção `account=on` especificada), as quotas não são impostas.

Para definir uma quota de grupo para o grupo `devel` (o grupo deve existir antes de definir a quota de grupo), use o seguinte comando:

```
edquota -g devel
```

Este comando exibe a quota existente para o grupo no editor de texto:

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

O sistema de arquivos GFS2 não mantém quotas para o inodes, então estas colunas não se aplicam para os sistemas de arquivos GFS2 e ficarão em branco. Modifique os limites e então salve o arquivo.

Para verificar que a quota de grupo foi definida, use o seguinte comando:

```
quota -g devel
```

#### 4.5.2. Gerenciando Quotas de Disco

Se as quotas são implementadas, elas precisam de alguma manutenção — a maioria para checar se as quotas estão excedidas e certificar que as quotas estão corretas.

É claro que se os usuários excederem suas quotas repetidamente ou consistentemente alcançarem seus limites soft, um administrador do sistema possui poucas escolhas a fazer dependendo de qual tipo os usuários são e quanto do espaço de disco impacta no seu trabalho. O administrador pode tanto ajudar o usuário a determinar como usar menos espaço de disco ou aumentar a quota de disco de usuário.



Você pode criar um relatório de uso de disco executando o utilitário `repquota`. Por exemplo, o comando `repquota /home` produz este resultado:

```
*** 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
```

Para ver o relatório de uso de disco para todos sistemas de arquivo com quotas ativadas (opção `a`), use o comando:

```
repquota -a
```

Enquanto o relatório é fácil de se ler, alguns pontos devem ser explicados. O `--` exibido após cada usuário é uma maneira rápida para determinar se um limite de bloco foi excedido. Se o limite `block soft` for excedido, um `+` aparece no lugar do primeiro `-` no resultado. O segundo `-` indica o limite `inode`, mas os sistemas de arquivos GFS2 não suportam limites `inode` que então o carácter permanecerá como `-`. Os sistemas de arquivos GFS2 não suportam um período de cortesia (`grace period`), então a coluna `cortesia` (`grace`) permanecerá em branco.

Note que o comando `repquota` não é suportado sobre o NFS, independente do sistema de arquivos subjacente.

### 4.5.3. Mantendo Quotas Precisar

Se você ativou as quotas em seu sistema de arquivos após um período de tempo onde você estava rodando com as quotas desativadas, você deve executar o comando `quotacheck` para criar, checar e reparar arquivos de quota. Adicionalmente, você pode querer rodar o comando `quotacheck` se você acha que seus arquivos de quotas podem não estar precisos, como pode ocorrer quando um sistema de arquivos não estiver montado adequadamente depois de um travamento do sistema.

Para mais informações sobre o comando `quotacheck`, veja a página [man quotacheck](#).



#### NOTA

Execute o `quotacheck` quando o sistema de arquivos estiver relativamente parado em todos os nós por causa que a atividade de disco pode descartar os valores de quota computados.

### 4.5.4. Sincronizando Quotas com o Comando `quotasync`

O GFS2 armazena toda informação de quota em seu arquivo interno próprio em disco. Um nó do GFS2 não atualiza este arquivo de quota para todas as escritas no sistema de arquivos; ao invés, por padrão, ele atualiza o arquivo de quota a cada 60 segundos. Isto é necessário para evitar contenção entre nós escrevendo no arquivo de quota, o qual causaria uma redução no desempenho.

Quando um usuário ou grupo se aproxima do limite de sua cota, o GFS2 reduz o tempo entre suas atualizações de cota arquivo, de forma dinâmica, para evitar que se exceda o limite. O tempo normal entre as sincronizações de cota é um parâmetro ajustável, `quota_quantum`. Por padrão, o tempo é de

60 segundos utilizando a opção de montagem `quota_quantum=` como descrito em [Tabela 4.2, “Opções de Montagem do GFS2 Específico”](#). O parâmetro de `quota_quantum` deve ser estabelecido em cada nó e todas as vezes que o sistema de arquivo for montado. Mudanças no parâmetro `quota_quantum` não são persistentes nas desmontagens. Você pode atualizar o valor de `quota_quantum` com `mount -o remount`.

Você pode usar o comando `quotasync` para sincronizar as informações de quota de um nó para um arquivo de quota em disco, entre atualizações automáticas realizadas pelo GFS2.

#### 4.5.4.1. Uso

##### Sincronizando Informações de Cotas

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

**u**

Sincronizar os arquivos de quota de usuário.

**g**

Sincronizar os arquivos de quota de grupos

**a**

Sincroniza todos os sistemas de arquivo que estão atualmente com quotas ativadas e suporte à sincronia. Quando um `-a` estiver ausente, o ponto de montagem do sistema de arquivos deve ser especificado.

**mntpnt**

Especifica o sistema de arquivo GFS2 para o qual as ações se aplicam.

##### Ajustando o Tempo entre as Sincronizações

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

**MountPoint**

Especifica o sistema de arquivo GFS2 para o qual as ações se aplicam.

**secs**

Especifica o novo período entre sincronizações de cota arquivo regulares pelo GFS2. Valores menores podem aumentar a contenção e diminuir o desempenho.

#### 4.5.4.2. Exemplos

Este exemplo sincroniza todas as quotas sujas no cache do nó que é rodado com o arquivo de quota `ondisk` para o sistema de arquivos `/mnt/mygfs2`.

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

Este exemplo muda o tempo padrão entre atualizações de quota arquivo regulares para uma hora (3600 segundos) para o sistema de arquivos `/mnt/mygfs2` na remontagem deste sistema de arquivo

no volume lógico `/dev/volgroup/logical_volume`.

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

#### 4.5.5. Referências

Para maiores informações sobre quotas de disco, consulte as páginas `man` dos seguintes comandos:

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

## 4.6. AUMENTANDO UM SISTEMA DE ARQUIVO

O comando `gfs2_grow` é usado para expandir um sistema de arquivo GFS2, depois que o dispositivo, onde o sistema de arquivo estiver, for expandido. A execução de um comando `gfs2_grow` em um sistema de arquivo GFS2 existente preenche todo o espaço livre entre o final do sistema arquivo atual e o final do dispositivo com uma extensão de sistema de arquivo GFS2 inicializada recentemente. Quando a operação de preenchimento for concluída, o índice dos recursos para o sistema de arquivo será atualizado. Todos os nós no cluster podem usar um espaço de armazenamento extra que tenha sido adicionado.

O comando `gfs2_grow` deve ser executado em um sistema de arquivo montado, mas precisa ser executado somente em um nó de um cluster. Todos os outros nós reconhecerão que a expansão ocorreu e automaticamente começarão a usar um novo espaço.



#### NOTA

Você não pode diminuir o tamanho do sistema de arquivo, depois da criação do sistema de arquivo com o comando `mkfs.gfs2`.

#### 4.6.1. Uso

```
gfs2_grow MountPoint
```

##### *MountPoint*

Especifica o sistema de arquivo GFS2 para o qual as ações se aplicam.

#### 4.6.2. Comentários

Antes de executar o comando `gfs2_grow`:

- Faça um backup dos dados importantes no sistema de arquivos.
- Determina o volume que é usado pelo sistema de arquivo a ser expandido executando o comando `df MountPoint`

- Expande o volume do cluster adjacente com o LVM. Para informações sobre como administrar os volumes LVM, veja o *Logical Volume Manager Administration*.

Depois de executar o comando `gfs2_grow`, execute o comando `df` para verificar se o novo espaço está disponível no sistema de arquivo.

### 4.6.3. Exemplos

Neste exemplo, o sistema de arquivo no diretório `/mygfs2fs`, é expandido.

```
[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.
```

### 4.6.4. Uso Completo

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

#### **MountPoint**

Especifica o diretório onde o sistema de arquivos do GFS2 foi montado.

#### **Device**

Especifica o nó de dispositivo do sistema de arquivo.

[Tabela 4.3, “Opções de GFS2 específico Disponíveis Ao Expandir um Sistema de Arquivos.”](#) descreve as opções específicas do GFS2 que podem ser usadas enquanto estiver expandindo um sistema de arquivos do GFS2.

**Tabela 4.3. Opções de GFS2 específico Disponíveis Ao Expandir um Sistema de Arquivos.**

Opções	Descrição
<code>-h</code>	Ajuda. Exibe uma mensagem curta de uso.
<code>-q</code>	Silencioso. Abaixa o nível de verbosidade.
<code>-r MegaBytes</code>	Especifica o tamanho do novo grupo de recursos. O tamanho padrão é de 256MB.
<code>-T</code>	Teste. Faz todos os cálculos, mas não edita nenhum dado no disco e não expande o sistema de arquivo.
<code>-V</code>	Exibe as informações sobre a versão do comando.

## 4.7. ADICIONANDO DIÁRIOS AO SISTEMA DE ARQUIVO

O comando `gfs2_jadd` é usado para adicionar diários ao sistema de arquivo GFS2. Você pode adicionar diários ao sistema de arquivo GFS2 de forma dinâmica em qualquer ponto, sem expandir o volume lógico adjacente. O comando `gfs2_jadd` deve ser executado em um sistema de arquivo montado, mas precisará ser executado somente em um nó no cluster. Todos os outros nós reconhecem que ocorreu uma expansão.



### NOTA

Caso um sistema de arquivo esteja cheio, o `gfs2_jadd` falhará, mesmo que o volume lógico que possui o sistema tenha sido estendido e seja maior que o sistema de arquivo. Isto deve-se ao fato de que em um sistema de arquivo, os jornais são arquivos simples ao invés de metadados incorporados, de forma que a simples extensão do volume lógico adjacente não fornecerá espaço para os diários.

Antes de adicionar os diários em um sistema de arquivo, você pode usar a opção `journals` do `gfs2_tool` para descobrir quantos diários existem no sistema de arquivo GFS2 atual. Os seguintes exemplos exibem o número e tamanho dos diários no sistema de arquivo montado em `/mnt/gfs2`.

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

### 4.7.1. Uso

```
gfs2_jadd -j Number MountPoint
```

#### *Número*

Especifica o número de diários a serem adicionados.

#### *MountPoint*

Especifica o diretório onde o sistema de arquivos do GFS2 foi montado.

### 4.7.2. Exemplos

Neste exemplo, um diário é adicionado ao sistema de arquivo no diretório `/mygfs2`.

```
gfs2_jadd -j1 /mygfs2
```

Neste exemplo, dois diários são adicionados no sistema de arquivo no diretório `/mygfs2`.

```
gfs2_jadd -j2 /mygfs2
```

### 4.7.3. Uso Completo

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

-

**MountPoint**

Especifica o diretório onde o sistema de arquivos do GFS2 foi montado.

**Device**

Especifica o nó de dispositivo do sistema de arquivo.

Tabela 4.4, “Opções do GFS2 Específicas Disponíveis para Adicionar Diários” descreve as opções do GFS2 específicas que podem ser usadas ao adicionar diários ao sistema de arquivo GFS2.

**Tabela 4.4. Opções do GFS2 Específicas Disponíveis para Adicionar Diários**

Sinalizador	Parâmetro	Descrição
-h		Ajuda. Exibe uma pequena mensagem de uso.
-J	<i>MegaBytes</i>	Especifica o tamanho dos novos diários em megabytes. O tamanho do diário padrão é de 128 megabytes. O tamanho mínimo é de 32 megabytes. Para adicionar diários de diferentes tamanhos ao sistema de arquivo, o comando <b>gfs2_jadd</b> deve ser executado para cada tamanho de diário. O tamanho especificado é arredondado para baixo para que seja um múltiplo do tamanho do segmento do diário que foi especificado quando o sistema de arquivo foi criado.
-j	<i>Número</i>	Especifica o número de diários novos a serem adicionados pelo comando <b>gfs2_jadd</b> . O valor padrão é de 1.
-q		Silencioso. Abaixa o nível de verbosidade.
-V		Exibe as informações sobre a versão do comando.

## 4.8. DIÁRIO DE DADOS

Geralmente, o GFS2 edita somente os metadados em seu diário. O conteúdo do arquivo é subsequentemente editado no disco pela sincronização periódica do kernel, a qual libera buffers de sistema de arquivo. Uma chamada **fsync ( )** em um arquivo, faz com que os dados do arquivo sejam editados no disco imediatamente. A chamada retorna quando o disco informa que todos os dados estão editados de maneira segura.

O diário de dados pode resultar em uma redução de tempo do **fsync ( )**, especialmente para arquivos pequenos, pois os dados do arquivo são editados no diário além dos metadados. Esta vantagem é inversamente proporcional ao tamanho do arquivo, sendo que a edição em arquivos médios e grandes levará mais tempo com o diário de dados acionado.

Os aplicativos que contam com o **fsync ( )** para sincronizar os dados do arquivo, podem melhorar o desempenho usando o diário de dados. O diário de dados pode ser ativado automaticamente para qualquer arquivo GFS2 criado em um diretório sinalizado ( e todos os seus subdiretórios). Os arquivos existentes com comprimento zero, também possuem diário de dados ativados ou desativados.

A ativação do diário de dados num diretório modifica o diretório para "inherit jdata", que indica que todos os arquivos e diretórios criados subsequentemente neste diretório passarão pelo diário. Você pode ativar e desativar o diário de dados num arquivo com o comando `chattr`.

Os comandos a seguir ativam o diário de dados no arquivo `/mnt/gfs2/gfs2_dir/newfile` e depois verifica se o sinalizador foi ajustado adequadamente.

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

Os comandos a seguir desativam o diário de dados no arquivo `/mnt/gfs2/gfs2_dir/newfile` e depois verifica se o sinalizador foi ajustado adequadamente.

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

Você pode usar o comando `chattr` para ajustar o sinalizador num diretório. Quando isto acontecer, todos os arquivos e diretórios criados subsequentemente naquele diretório passarão pelo diário. O seguinte conjunto de comandos ajusta o sinalizador `j` no diretório `gfs2_dir` e verifica se o sinalizador foi ajustado corretamente. Em seguida, os comandos criam um novo arquivo chamado `newfile` no diretório `/mnt/gfs2/gfs2_dir` e verifica se o sinalizador foi ajustado para o arquivo. Como o sinalizador `jdata` é estabelecido para o diretório, então o `newfile` também deve ter o diário ativado.

```
[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. CONFIGURANDO ATUALIZAÇÕES DO `ATIME`

Cada inode de arquivo e inode de diretório possui três carimbos de hora associados à ele:

- `ctime` – A última vez que o status do inode foi modificado
- `mtime` – A última vez que os dados do arquivo (ou diretório) foram modificados
- `atime` – A última vez que os dados do arquivo (ou diretório) foram acessados

Se as atualizações `atime` forem mantidas ativas, como são por padrão no GFS2 e em outros sistemas de arquivo do Linux, seu inode precisará ser atualizado todas as vezes que um arquivo for lido.

Como alguns aplicativos usam a informação provida pelo `atime`, estas atualizações podem requerer uma quantidade significativa de tráfego de edição desnecessário e tráfego de bloqueio de arquivo. Este tráfego pode diminuir o desempenho, por isso, pode ser melhor desativar as atualizações do `atime`.

Existem dois métodos disponíveis para reduzir os efeitos das atualizações do `atime`:

- Montar com o `relatime` (`atime` relativo), que atualiza o `atime`, caso a atualização `atime`

anterior seja mais antiga do `mtime` ou atualização `ctime`.

- Montar com o `noatime`, o qual desativa as atualizações do `atime` neste sistema de arquivo.

#### 4.9.1. Montar com o `relatime`

A opção de montagem do Linux (atime relativo) `relatime` pode ser especificada quando o sistema de arquivo for montado. Isto especifica que o `atime` é atualizado caso a atualização anterior `atime` for mais antiga que a atualização `mtime` ou `ctime`.

##### 4.9.1.1. Uso

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

##### ***BlockDevice***

Especifica o dispositivo de bloco onde o sistema de arquivo do GFS2 reside.

##### ***MountPoint***

Especifica o diretório onde o sistema de arquivo GFS2 deve ser montado.

#### 4.9.1.2. Exemplo

Neste exemplo, o sistema de arquivo do GFS2 reside no `/dev/vg01/lvol0` e é montado no diretório `/mygfs2`. As atualizações `atime` assumem comando apenas se a atualização `atime` anterior for mais velha que a atualização `mtime` ou `ctime`.

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

#### 4.9.2. Montar com `noatime`

Uma opção de montagem do Linux `noatime` pode ser especificada quando o sistema de arquivo é montado, o qual desativa as atualizações do `atime` neste sistema de arquivo.

##### 4.9.2.1. Uso

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

##### ***BlockDevice***

Especifica o dispositivo de bloco onde o sistema de arquivo do GFS2 reside.

##### ***MountPoint***

Especifica o diretório onde o sistema de arquivo GFS2 deve ser montado.

#### 4.9.2.2. Exemplo

Neste exemplo, o sistema de arquivo do GFS2 reside no `/dev/vg01/lvol0` e é montado no diretório `/mygfs2` com atualizações `atime` desativadas.



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

## 4.10. SUSPENDENDO A ATIVIDADE EM UM SISTEMA DE ARQUIVO

Você pode suspender a atividade de edição em um sistema de arquivo usando o comando `dmsetup suspend`. Suspender as atividades de edição, permite que os snapshots do dispositivo baseado em hardware, sejam usados para capturar o sistema de arquivo em estado consistente. O comando `dmsetup resume` finaliza a suspensão.

### 4.10.1. Uso

#### Inicializa a Suspensão

```
dmsetup suspend MountPoint
```

#### Encerra a Suspensão

```
dmsetup resume MountPoint
```

#### *MountPoint*

Especifica o sistema de arquivo.

### 4.10.2. Exemplos

Este exemplo suspende edições no sistema de arquivo `/mygfs2`.

```
# dmsetup suspend /mygfs2
```

Este exemplo finaliza a suspensão de edições no sistema de arquivo `/mygfs2`.

```
# dmsetup resume /mygfs2
```

## 4.11. REPARANDO UM SISTEMA DE ARQUIVO

Quando um nó falha com o sistema de arquivo montado, o diário do sistema de arquivo permite uma recuperação rápida. No entanto, se um dispositivo de armazenamento perder potência ou for desconectado fisicamente, pode ocorrer um corrompimento do sistema de arquivo. (O diário pode ser usado para recuperar das falhas de subsistemas de armazenamento). Quando este tipo de corrompimento ocorrer, você pode recuperar o sistema de arquivo do GFS2 usando o comando `gfs2_fsck`.



### IMPORTANTE

O comando `gfs2_fsck` deve ser executado somente em um sistema de arquivo que é desmontado de todos os nós.

## IMPORTANTE

Você não deve verificar o sistema de arquivo GFS2 durante a inicialização com o comando `fsck.gfs2`. Este comando não pode determinar se o sistema de arquivo foi montado por outro nó no cluster durante a inicialização. Você precisa executar o comando `fsck.gfs2` manualmente somente após o sistema inicializar.

Para certificar-se de que o comando `fsck.gfs2` não é executado em um sistema de arquivo do GFS2 durante a inicialização, modifique o arquivo `/etc/fstab` para que as duas colunas finais para um ponto de montagem do sistema de arquivo GFS2 exiba "0 0" ao invés de "1 1" (ou qualquer outro número), como no exemplo a seguir:

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

## NOTA

Se você tiver experiência prévia usando o comando `gfs_fsck` em um sistema de arquivo GFS, observe que o comando `gfs2_fsck` difere das versões anteriores ao `gfs_fsck` das seguintes formas:

- Pressionando `Ctrl+C` enquanto rodando o `fsck.gfs2`, irá interromper o processamento e exibirá uma janela perguntando se você deseja interromper o comando, pular o resto do passo atual ou continuar processando.
- Você pode aumentar o nível de verbosidade usando o sinalizador `-v`. Adicionar um segundo `-v` aumenta o nível novamente.
- Você pode diminuir o nível de verbosidade, usando o sinalizador `-q`. Adicionar um segundo `-q` diminui o nível novamente.
- A opção `-n` abre um sistema de arquivo como somente-leitura e não responde à quaisquer buscas automaticamente. A opção provê uma forma de tentar fazer o comando revelar erros sem realmente permitir que o comando `gfs2_fsck` tome efeito.

Consulte a página `man` do `gfs2_fsck` para informações adicionais sobre outras opções de comando.

A rodagem do comando `fsck.gfs2` requer a memória do sistema acima e além da memória usada para o sistema de operação e kernel. Cada bloco de memória no sistema de arquivo GFS2 requer aproximadamente cinco bites de memória adicional, ou 5/8 de um bite. Portanto, para estimar quantos bites de memória você precisará para rodar o comando `fsck.gfs2` no seu sistema de arquivo, determine quantos blocos o sistema de arquivo possui e multiplique aquele número por 5/8.

Por exemplo: para determinar aproximadamente quanta memória é necessária para rodar o comando `fsck.gfs2` num sistema de arquivo GFS2 que possui 16TB com um bloco de 4K, determine primeiro quantos blocos de memória o sistema de arquivo possui, dividindo 16Tb por 4K:

```
17592186044416 / 4096 = 4294967296
```

Uma vez que o sistema de arquivo possui 4294967296 blocos, multiplique este número por 5/8 para determinar quantos bites de memória são necessários:

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

Este sistema de arquivos requer aproximadamente 2.6GB de espaço de memória para rodar o comando `fsck.gfs2`. Perceba que o tamanho do bloco era 1K. A rodagem do comando `fsck.gfs2` requer quatro vezes mais de memória ou aproximadamente 11GB.

### 4.11.1. Uso

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

`-y`

O sinalizador `-y` faz com que todas as perguntas sejam respondidas com `sim`. Com o sinalizador `-y` especificado, o comando `gfs2_fsck` não lhe pede uma resposta antes de fazer as mudanças.

#### *BlockDevice*

Especifica o dispositivo de bloco onde o sistema de arquivo do GFS2 reside.

### 4.11.2. Exemplo

Neste exemplo, o sistema de arquivo do GFS2, o qual reside no dispositivo do bloco `/dev/testvol/testlv` é reparado. Todas as buscas para reparar são automaticamente respondidas com `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. MONTAGENS BIND E NOMES DE CAMINHO DE CONTEXTO DEPENDENTE.

Os sistemas de arquivo do GFS2 não fornece suporte para os Nomes de Caminhode Contexto

Dependente (CDPNs), o qual permite que você crie links simbólicos que apontam para os arquivos ou diretórios de destino de variantes. Para esta funcionalidade do GFS2, use a opção **bind** do comando **mount**.

A opção **bind** no comando **mount**, permite que você remonte parte de uma hierarquia de arquivo em locais diferentes enquanto estiver ainda disponível no local original. O formato deste comando é:

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

Após executar este comando, o conteúdo do diretório **olddir** estará disponível em dois locais: **olddir** e **newdir**. Você também pode usar esta opção para disponibilizar um arquivo individual em dois locais.

Por exemplo, após executar os seguintes comandos, o conteúdo do **/root/tmp** será idêntico ao conteúdo do diretório **/var/log** montado anteriormente.

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

Como forma alternativa, você pode usar uma entrada no arquivo **/etc/fstab** para obter o mesmo resultado durante a montagem. A entrada a seguir **/etc/fstab** resultará no conteúdo **/root/tmp** sendo idêntico ao conteúdo do diretório **/var/log**.

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

Depois que você houver montado o sistema de arquivo, use o comando **mount** para ver se o sistema de arquivo foi montado, como no exemplo a seguir.

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

Com um sistema de arquivo que suporta os Nomes de Caminho do Contexto Dependente, você deve definir o diretório **/bin** como um Nome de Caminho de Contexto Dependente que resolveria para um dos seguintes caminhos, dependendo da arquitetura do sistema.

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

Você pode obter esta mesma funcionalidade, criando um diretório **/bin** vazio. Depois, se usar um script ou uma entrada no arquivo **/etc/fstab**, poderá montar cada um dos diretórios de arquitetura individual no diretório **/bin** com um comando **mount -bind**. Por exemplo, você pode usar o seguinte comando como uma linha em um script.

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

Como forma alternativa, você pode usar a seguinte entrada no arquivo **/etc/fstab**.

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

Um `bind mount` pode fornecer maior flexibilidade do que o Nome do Caminho de Contexto Dependente, pois você pode usar este recurso para montar diretórios diferentes de acordo com qualquer critério que você definir (tal como o valor de `%fill` para o sistema de arquivo). Os Nomes de Caminho de Contexto Dependente são mais limitados no que podem abranger. Observe no entanto, que você precisará editar seu próprio script para montar de acordo com o critério tal qual o valor do `%fill`.



### ATENÇÃO

Quando você montar um sistema de arquivo com a opção `bind` o sistema de arquivo original já tiver sido montado como `rw`, o novo sistema de arquivo também será montado como `rw` mesmo que use o sinalizador `ro`. O `ro` é ignorado. Neste caso, o novo sistema de arquivo deve ser marcado como `ro` no diretório `/proc/mounts` o qual pode ser confuso.

## 4.13. MONTA BINDS E ORGANIZA A MONTAGEM DE SISTEMA DE ARQUIVO

Quando você usa a opção `bind` do comando `mount`, você deve certificar-se que os sistemas de arquivo são montados na ordem correta. No seguinte exemplo, o diretório `/var/log` deve ser montado antes de executar a montagem `bind` no diretório `/tmp`:

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

A ordem das montagens do sistema de arquivo está determinada abaixo:

- Em geral, a montagem do sistema de arquivos é determinada pela ordem em que os sistemas de arquivos aparecem no arquivo `fstab`. As exceções para esta ordenação são sistemas de arquivos montados com o sinalizador `_netdev` ou sistemas de arquivos que possuem seus próprios scripts `init`.
- Um sistema de arquivo com o próprio script `init` é montado mais tarde no processo de inicialização, após os sistemas de arquivo no arquivo `fstab`.
- Os sistemas de arquivo montados com o sinalizador `_netdev` são montados quando a rede é ativada no sistema.

Se sua configuração requer que você crie um `bind` de montagem na qual montar um sistema de arquivos GFS2, você pode ordenar seu arquivo `fstab` conforme a seguir:

1. Monte sistemas de arquivos locais que são requeridos para a
2. Faz o `bind` de montagem no diretório no qual se montará o sistema de arquivos GFS2.
3. Monte do sistema de arquivos GFS2.

Se sua configuração requer que você faça um `bind` de montagem em um diretório local ou sistema de arquivos em um sistema de arquivos GFS2, listando os sistemas de arquivos na ordem correta no arquivo `fstab` não montará os sistemas de arquivos corretamente já que o sistema de arquivos GFS2

não será montado até que o script `init` do GFS2 seja rodado. Neste caso, você deve escrever um script `init` para executar o bind de montagem, para que então esse não tome lugar até que o sistema GFS2 seja montado.

O script seguinte é um exemplo de um script `init` personalizado. Este script realiza um bind de montagem de dois diretórios em cima de dois diretórios de um sistema de arquivos GFS2. Neste exemplo, há um ponto de montagem GFS2 em `/mnt/gfs2a`, que é montado quando o script `init` do GFS2 roda, depois da inicialização do cluster.

Next script de exemplo, os valores da declaração `chkconfig` indicam o seguinte:

- 345 indica os níveis de execução em que o script será iniciado.
- 29 é a prioridade de início, que neste caso indica que o script será executado na hora da inicialização depois do script `init` do GFS2, que possui prioridade de início 26.
- 73 é a prioridade de parada, que neste caso indica que o script será parado durante o desligamento antes do script GFS2 que possui uma prioridade de parada de 74.

Os valores de parada e início indicam que você pode manualmente realizar a ação indicada executando um comando de **início de serviço** e **parada de serviço**. Por exemplo, se o script é chamado `fredwilma`, então você pode executar `service fredwilma start`.

O script deve ser colocado no diretório `/etc/init.d` com as mesmas permissões assim como os outros scripts nesse diretório. Você pode executar um comando `chkconfig on` para ligar o script ao níveis de execução indicados. Por exemplo, se o script é chamado `fredwilma`, então você pode executar `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
    ;;
esac
```

```

status)
    ;;

restart)
    $0 stop
    $0 start
    ;;

reload)
    $0 start
    ;;

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

exit 0

```

## 4.14. A FUNÇÃO DE RETIRADA DO GFS2

A função *withdraw* (retirada) do GFS2 é um recurso de integridade de dados dos sistemas de arquivos GFS2 em um cluster. Se o módulo do kernel GFS2 detecta uma inconsistência em um sistema de arquivos GFS2 depois de uma operação de E/S, o sistema de arquivos se torna indisponível ao cluster. A operação de E/S pára e o sistema aguarda por mais operações de E/S para dar como erro, prevenindo mais danos. Quando isso ocorre, você pode parar quaisquer outros serviços ou aplicações manualmente, depois do qual você pode reinicializar e remontar o sistema de arquivos GFS2 para re-executar os diários. Se o problema repetir, você pode desmontar o sistema de arquivos de todos os nós no cluster e realizar uma recuperação do sistema de arquivos com o comando `fsck.gfs2`. A retirada da função GFS é menos severa que um pânico do kernel, que causaria um outro nó fazer um fence neste nó.

Se seu sistema estiver configurado com o script de inicialização `gfs2` ativado e o sistema de arquivos GFS2 estiver incluído no arquivo `/etc/fstab`, o sistema de arquivos GFS2 será remontado quando você reinicializar. Se o sistema de arquivos GFS2 foi retirado porque percebeu uma corrupção do sistema de arquivos, é recomendado que você execute o comando `fsck.gfs2` antes de remontar o sistema de arquivos. Neste caso, para prevenir seu sistema de arquivos de remontar na inicialização, você pode seguir o seguinte procedimento:

1. Temporariamente desative o script de inicialização no nó afetado com o seguinte comando:

```
# chkconfig gfs2 off
```

2. Reinicialize o nó afetado, iniciando o software de cluster. O sistema de arquivos GFS2 não será montado.
3. Desmontar o sistema de arquivos em todos os nós no cluster.
4. Execute o comando `fsck.gfs2` no sistema de arquivos de um nó somente para assegurar que não há corrupção do sistema de arquivos.
5. Reative o script de inicialização no nó afetado executando o seguinte comando:

```
# chkconfig gfs2 on
```

## 6. Remonte o sistema de arquivos de todos os nós no cluster.

Um exemplo de uma inconsistência que submeteria uma remoção do GFS2 é uma contagem de blocos incorreta. Quando o kernel GFS deleta um arquivo de um sistema de arquivos, ele semanticamente remove todos os dados e blocos de metadados associados com esse arquivo. Quando isso é feito, ele checa a contagem de blocos. Se a contagem de blocos não é "um" (significando que tudo que restou é o próprio inode de disco), que indica que uma inconsistência do sistema de arquivos já que a contagem de bloco não correspondeu à lista de blocos encontradas.

Você pode sobrescrever a função de retirada do GFS2 montado o sistema de arquivos com a opção `-o errors=panic` especificada. Quando esta opção é especificada, quaisquer erros que normalmente causariam a retirada do sistema faz o sistema entrar em pânico. Isto pára as comunicações de cluster dos nós, que faz o nó entrar em fence.

Internamente, a maneira que a função de retirada do GFS2 trabalha é fazer o kernel enviar uma mensagem ao daemon `gfs_controld` requerendo a retirada. O daemon `gfs_controld` roda o programa `dmsetup` para colocar o alvo do erro do mapeador de dispositivos sob o sistema de arquivos prevenindo acesso ao dispositivo de bloco. Ele então diz ao kernel que a operação foi completada. Esta é a razão para o requerimento de suporte do GFS2 para sempre usar um dispositivo CLVM sob GFS2, já que de outra maneira não é possível inserir um alvo de mapeador de dispositivo.

O propósito do alvo de erro do mapeador de dispositivo é garantir que todas as operações de E/S futuras resultarão em um erro de E/S que permitirão ao sistema de arquivos ser desmontado de uma maneira ordenada. Como um resultado, quando a retirada ocorre, é normal ver erros de E/S do dispositivo mapeador relatados nos sistemas de logs.

Ocasionalmente, a retirada pode falhar se não for possível para o programa `dmsetup` inserir os alvos de erro conforme requeridos. Isto pode ocorrer se há uma falta de memória no ponto de retirada e mais memória não pode ser recuperada devido ao problema que desencadeou a retirada em primeiro lugar.

Uma retirada nem sempre significa que há um erro no GFS2. As vezes a função de retirada pode ser iniciada por erros no dispositivo de E/S relacionado ao bloco de dispositivo subjacente. É altamente recomendável verificar os logs para ver se esse é o motivo se uma retirada ocorrer.



## CAPÍTULO 5. DIAGNOSTICANDO E CORRIGINDO PROBLEMAS COM OS SISTEMAS DE ARQUIVOS GFS2

Este capítulo fornece informações sobre alguns problemas comuns do GFS2 e como lidar com eles.

### 5.1. O SISTEMA DE ARQUIVOS GFS2 EXIBE UM BAIXO DESEMPENHO

Você pode achar que seu sistema de arquivos GFS2 exibe um desempenho mais lento do que um sistema de arquivos ext3. O desempenho do GFS2 pode ser afetado por certas influências e também certos casos de uso. Informações que lidam com os problemas de desempenho do GFS2 são encontradas neste documento.

### 5.2. O SISTEMA DE ARQUIVOS GFS2 SUSPENDE E REQUER REINICIALIZAÇÃO DE UM NÓ.

Se seu sistema de arquivos GFS2 suspende e não retorna comandos de execução nele, mas reinicializar um nó específico retorna o sistema ao normal, isto pode ser um indicativo de um problema de bloqueio ou bug. Se isto ocorrer, você deve recolher os seguintes dados:

- O despejo do bloqueio gfs2 para o sistema de arquivos em cada nó:

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

- O despejo de bloqueio DLM para o sistema de arquivos em cada nó: Você pode obter essa informação com o `d1m_tool`:

```
d1m_tool lockdebug -sv l1name.
```

Neste comando, o `l1name` é o nome do espaço de bloqueio usado pelo DLM para o sistema de arquivos em questão. Você pode encontrar este valor no resultado do comando `group_tool`.

- O resultado do comando `sysrq -t`.
- Os conteúdos do arquivo `/var/log/messages`.

Uma vez que recolheu os dados, você pode abrir um tíquete com o Suporte Red Hat e fornecer os dados que você coletou.

### 5.3. O SISTEMA DE ARQUIVOS GFS2 SUSPENDE E REQUER REINICIALIZAÇÃO DE TODOS OS NÓS

Se seu sistema de arquivos GFS2 suspende e não retorna comandos, requerendo que você reinicialize todos os nós nos cluster antes de usa-los, cheque pelos seguintes problemas.

- Você pode ter tido uma falha no fence. Os sistemas de arquivos GFS2 congelarão para assegurar a integridade dos dados no evento de uma falha de fence. Cheque as mensagens de log para ver se existem quaisquer fences com falha no momento da suspensão. Assegure-se que o fence está configurado corretamente.
- O sistema de arquivos GFS2 pode ter sido retirado. Cheque pelas mensagens do log pela palavra `withdraw` e veja se há mensagens e traços do GFS2 indicando que o sistema de arquivos foi retirado. Uma retirada (`withdraw`) é um indicador de tanto corrupção do sistema

de arquivos, uma falha na armazenagem ou um bug. Desmonte o sistema de arquivos, atualize o pacote `gfs2-utils` e execute o comando `fsck` no sistema de arquivos para retorná-lo à operação. Abra um tíquete de suporte com o Suporte da Red Hat. Informe-os que você passou por uma retirada de GFS2 (`withdraw`) e forneça o `sosreports` com logs.

Para informações sobre a função GFS2 `withdraw`, veja a [Seção 4.14, “A Função de Retirada do GFS2”](#).

- Este erro pode ser um indicativo de um problema de bloqueio ou bug. Recolha os dados durante uma dessas ocorrências e abra um tíquete de suporte com o Suporte Red Hat, conforme descrito na [Seção 5.2, “O Sistema de Arquivos GFS2 Suspende e Requer Reinicialização de um Nó.”](#)

## 5.4. O SISTEMA DE ARQUIVOS GFS2 NÃO MONTA UM NÓ DE CLUSTER RECÉM ADICIONADO.

Se você adicionar um novo nó a um cluster e descobrir que não pode montar seu sistema de arquivos GFS2 neste nó, você pode ter menos diários no sistema de arquivos GFS2 do que nós tentando acessar o sistema de arquivos GFS2. Você deve ter um diário por host GFS2 que pretende montar no sistema de arquivos (com exceção dos sistemas de arquivos GFS2 montados com o conjunto de opções `spectator` definidas, já que estes não requerem diários). Você pode adicionar diários a um sistema de arquivos GFS2 com o comando `gfs2_jadd`, conforme descrito na [Seção 4.7, “Adicionando Diários ao Sistema de Arquivo”](#).

## 5.5. ESPAÇO INDICADO COMO USADO EM UM SISTEMA DE ARQUIVOS VAZIO

Se você possui um sistema de arquivos GFS2 vazio, o comando `df` mostrará que há espaço sendo tomado. Isto é por causa que os diários do sistema de arquivos consomem espaço (número de diários \* tamanho dos diários) no disco. Se você criou um sistema de arquivos GFS2 com um grande número de diários ou especificado um tamanho grande de diários, então você verá (número de diários \* tamanho dos diários) como já em uso quando você executar o `df`. Mesmo que você não especificou um número grande de diários ou grandes diários, sistemas de arquivos GFS2 pequenos (na faixa de 1GB ou menos) mostrarão uma quantidade grande de espaço sendo usadas com o tamanho de diário padrão.

## CAPÍTULO 6. CONFIGURANDO UM SISTEMA DE ARQUIVO DO GFS2 EM UM CLUSTER DE PACEMAKER.

O procedimento a seguir é um guia de passos necessários para configurar um cluster de Pacemaker que inclui o sistema de arquivo do GFS2.

Após instalar o software do cluster e GFS2 e pacotes de LVM em cluster em cada nó, inicie os serviços `cman`, `clvmd`, e `pacemaker` e crie o cluster do Pacemaker. Você precisa configurar o isolamento para o cluster. Para mais informações sobre configuração de um cluster Pacemaker, veja *Configuring the Red Hat High Availability Add-On with Pacemaker*.

1. Definir o parâmetro do Pacemaker global `no_quorum_policy` para `freeze`.



### NOTA

Por padrão, o valor do `no-quorum-policy` é definido para `stop`, indicando que uma vez que o quorum se perde, todos os recursos na partição restante serão interrompidos imediatamente. Geralmente este padrão é a opção mais segura e otimizada, mas ao contrário da maioria dos recursos, o GFS2 requer que o quorum funcione. Quando o quorum estiver perdido, as aplicações que estiverem utilizando as montagens GFS2 e a própria GFS2, não serão interrompidas corretamente. Qualquer tentativa de interrupção destes recursos sem o quorum, irá falhar, a qual resultará no isolamento de todo o cluster, todas as vezes que o quorum estiver perdido.

Para lidar com esta situação, você pode estabelecer o `no-quorum-policy=freeze` quando o GFS2 estiver sendo usado. Isto significa que quando o quorum se perde, a partição restante não fará nada até que o quorum retorne.

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

2. Após garantir que o tipo de bloqueio está definido para 3 no arquivo `/etc/lvm/lvm.conf` para suportar o bloqueio em cluster, crie o LV em cluster e o formato do volume com um sistema de arquivo GFS2. Assegure-se de que você possa criar agendas suficientes para cada um dos nós em seu 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. Configure um recurso `clusterfs`.

Você não precisa adicionar o sistema de arquivo ao arquivo `/etc/fstab` porque será gerenciado como um recurso de cluster Pacemaker. As opções de montagem podem ser especificadas como parte da configuração do recurso com o `options=options`. Execute o comando `pcs resource describe Filesystem` para opções de configurações completas.

Este comando de criação do cluster especifica a opção de montagem do `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. Verificar se o GFS2 está montado como o esperado

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

5. (Opcional) Reinicialize todos os nós de cluster para verificar a persistência e recuperação do gfs2

# APÊNDICE A. GERENCIAMENTO DE QUOTA DO GFS2 COM O COMANDO `GFS2_QUOTA`

A partir do lançamento do Red Hat Enterprise Linux 6.1, o GFS2 suporta as facilidades de quotas padrões do Linux. Para usar isso, você precisará instalar o RPM `quota`. Esta é a maneira preferida para administrar quotas no GFS2 e deve ser usada para todas as novas implementações do GFS2 usando quotas. Para informações sobre o uso das facilidades de quotas padrões do Linux, veja a [Seção 4.5, “Gerenciamento de Cota do GFS2”](#).

Para versões anteriores do Red Hat Enterprise Linux, o GFS2 requeria o comando `gfs2_quota` para gerenciar quotas. Este apêndice documenta o uso do comando `gfs2_quota` para gerenciar quotas de sistema de arquivos GFS2.

## A.1. DEFININDO QUOTAS COM O COMANDO `GFS2_QUOTA`

Duas definições de quotas estão disponíveis para cada ID de usuário (UID) ou ID de grupo (GID): um *hard limit* e um *soft limit*.

Um *hard limit* é a quantidade de espaço que pode ser usado. O sistema de arquivos não permitirá ao usuário ou grupo usar mais do que a quantidade de espaço de disco. Um valor de *hard limit* de zero significa que nenhum limite está imposto.

O *soft limit* é normalmente um valor menor do que o *hard limit*. O sistema de arquivos notificará o usuário ou grupo quando o *soft limit* é alcançado para avisá-los sobre a quantidade de espaço que estão usando. Um *soft limit* de valor zero significa que não há limite imposto.

Você pode definir limites usando o comando `gfs2_quota`. O comando somente precisa ser executado em um nó único onde o GFS2 estiver montado.

Por padrão, a imposição de quotas não é definida em sistemas de arquivos GFS2. Para ativar a contabilidade de quotas, use `quota=` do comando `mount` quando montar sistema de arquivos GFS2, conforme descrito na [Seção A.4, “Ativando/Desativando a Imposição de Quotas”](#).

### A.1.1. Uso

#### Definindo Quotas, Hard Limit

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

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

#### Definindo Quotas, Avisos de Limite

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

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

#### *User*

Uma ID de usuário para limitar ou avisar. Pode ser um nome de usuário de um arquivo de senha ou o número UID.

### **Group**

Um ID de grupo para limitar ou avisar. Pode ser tanto um nome de grupo do arquivo de grupo ou o número GID.

### **Size**

Especifica o novo valor de limite ou aviso. Por padrão, o valor é em unidades de megabytes. Os sinalizadores adicionais **-k**, **-s**, e **-b** mudam as unidades para kilobytes, setores e blocos do sistema de arquivos, respectivamente.

### **MountPoint**

Especifica o sistema de arquivos GFS2 ao qual as ações de aplicam.

## **A.1.2. Exemplos**

Este exemplo define o hard limit para o usuário *Bert* para 1024 megabytes (1 gigabyte) no sistema de arquivos */mygfs2*.

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

Este exemplo define o soft limit para o ID do grupo de 21 para 50 kilobytes no sistema de arquivos */mygfs2*.

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

## **A.2. EXIBINDO LIMITES DE QUOTAS E USO COM O COMANDO `GFS2_QUOTA`**

Limites de quotas e o uso atual podem ser exibidos para um usuário específico ou grupo usando o comando `gfs2_quota get`. Todo o conteúdo do arquivo de quota pode também ser exibido usando o comando `gfs2_quota`, que no caso todos os IDs com hard limits diferentes de zero, soft limits, ou valores são exibidos.

### **A.2.1. Uso**

#### **Exibindo Limites de Quotas para um Usuário**

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

#### **Exibindo Limites de Quotas para um Grupo**

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

#### **Exibindo Todo o Arquivo de Quotas**

```
gfs2_quota list -f MountPoint
```

#### **User**

Uma ID de usuário para mostrar informações sobre um usuário específico. Pode ser tanto um nome de usuário do arquivo de senha ou número UID.

**Group**

Uma ID de grupo para exibir informações sobre um grupo específico. Pode ser tanto um nome de grupo do grupo de arquivo ou o número GID.

**MountPoint**

Especifica o sistema de arquivos GFS2 ao qual as ações de aplicam.

**A.2.2. Resultado do Comando**

As informações de quota GFS2 do comando `gfs2_quota` são mostradas a seguir:

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

Os números (valores) do *LimitSize*, *WarnSize*, e *Value* estão em unidades de megabytes por padrão. Adicionar os sinalizadores `-k`, `-s`, ou `-b` à linha de comando muda as unidades para kilobytes, setores ou blocos de sistema de arquivos respectivamente.

**User**

Um nome de usuário ou ID o qual os dados são associados.

**Group**

Um nome de grupo ou ID a qual os dados são associados.

**LimitSize**

A definição hard limit para o usuário ou grupo. Este valor é zero se nenhum limite foi estabelecido.

**Value**

A quantidade atual de espaço em disco usado pelo usuário ou grupo.

**A.2.3. Comentários**

Quando exibir informações de quota, o comando `gfs2_quota` não determina os UIDs e GIDs em nomes se a opção `-n` é adicionada à linha de comando.

Espaço alocado para os arquivos escondidos do GFS2 podem ser deixados de fora os valores exibidos para o UID root e GID adicionando a opção `-d` à linha de comando. Isto é útil quando tentar corresponder os números do `gfs2_quota` com os resultados do comando `du`.

**A.2.4. Exemplos**

Este exemplo exibe informações de quotas para todos os usuários e grupos que têm um limite estabelecido ou estão usando qualquer espaço de disco do sistemas de arquivos `/mygfs2`.

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

Este exemplo exibe informações de quota em setores para usuários do grupo no sistema de arquivos `/mygfs2`.

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

### A.3. SINCRONIZANDO QUOTAS COM O COMANDO `GFS2_QUOTA`.

O GFS2 armazena todas as informações de quota em seu próprio arquivo interno no disco. Um nó GFS2 não atualiza este arquivo de quota para cada escrita no sistema de arquivos; em vez disso, por padrão ele atualiza o arquivo de quotas uma vez a cada 60 segundos. Isto é necessário para evitar contenção entre nós escrevendo no arquivo de quotas, que causaria uma lentidão no desempenho.

Quando um usuário ou grupo se aproxima do limite de sua cota, o GFS2 reduz o tempo entre suas atualizações de cota arquivo, de forma dinâmica, para evitar que se exceda o limite. O tempo normal entre as sincronizações de cota é um parâmetro ajustável, `quota_quantum`. Por padrão, o tempo é de 60 segundos utilizando a opção de montagem `quota_quantum=` como descrito em [Tabela 4.2, “Opções de Montagem do GFS2 Específico”](#). O parâmetro de `quota_quantum` deve ser estabelecido em cada nó e todas as vezes que o sistema de arquivo for montado. Mudanças no parâmetro `quota_quantum` não são persistentes nas desmontagens. Você pode atualizar o valor de `quota_quantum` com `mount -o remount`.

Você pode usar o comando `gfs2_quota sync` para sincronizar as informações de quota de um nó ao arquivo de quotas no disco entre as atualizações automáticas realizadas pelo GFS2.

#### A.3.1. Uso

##### Sincronizando Informações de Quotas

```
gfs2_quota sync -f MountPoint
```

##### *MountPoint*

Especifica o sistema de arquivos GFS2 ao qual as ações de aplicam.

##### Ajustando o Tempo entre Sincronizações

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

##### *MountPoint*

Especifica o sistema de arquivos GFS2 ao qual as ações de aplicam.

##### *secs*

Especifica um novo período de tempo entre sincronizações do arquivo de quota pelo GFS2. Valores menores podem aumentar a contenção e diminuir o desempenho.

#### A.3.2. Exemplos

Este exemplo sincroniza as informações de quota do nó em questão com o sistema de arquivos `/mygfs2`.



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

Este exemplo muda o tempo padrão entre atualizações de quota arquivo regulares para uma hora (3600 segundos) para o sistema de arquivos `/mnt/mygfs2` na remontagem deste sistema de arquivo no volume lógico `/dev/volgroup/logical_volume`.

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

## A.4. ATIVANDO/DESATIVANDO A IMPOSIÇÃO DE QUOTAS

Em sistema de arquivos GFS2, a imposição de quotas é desativada por padrão. Para habilitar a imposição de quotas para um sistema de arquivos, monte do sistema de arquivos com a opção `quota=on` especificada.

### A.4.1. Uso

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

Para montar um sistema de arquivos com imposição de quotas desativada, monte o sistema de arquivos com o a opção `quota=off` especificada. Isto é uma configuração padrão.

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

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

Especifica que a imposição de quotas está ativada ou desativada quando um sistema de arquivos é montado.

#### *BlockDevice*

Especifica o dispositivo de bloco onde um sistema de arquivos GFS2 reside.

#### *MountPoint*

Especifica o diretório onde o sistema de arquivos GFS2 deve ser montado.

### A.4.2. Exemplos

Neste exemplo, o sistema de arquivos GFS2 em `/dev/vg01/lvol0` está montado no diretório `/mygfs2` com imposição de quotas ativada.

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

## A.5. ATIVANDO QUOTAS ACCOUNTING

É possível acompanhar o uso de disco e manter quotas accounting para cada usuário e grupos sem impor valores limites e de aviso. Para fazer isso, monte o sistema de arquivos com a opção `quota=account` especificada.

### A.5.1. Uso

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

**-o quota=account**

Especifica que as estatísticas de uso do usuário e grupos são mantidas pelo sistema de arquivos, mesmo que os limites de quota não são impostos.

***BlockDevice***

Especifica o dispositivo de bloco onde um sistema de arquivos GFS2 reside.

***MountPoint***

Especifica o diretório onde o sistema de arquivos GFS2 deve ser montado.

**A.5.2. Exemplo**

Neste exemplo, o sistema de arquivos GFS2 no `/dev/vg01/lvol0` está montado no diretório `/mygfs2` com quota accounting ativada.

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

## APÊNDICE B. CONVERTENDO UM SISTEMA DE ARQUIVO UTILIZANDO OS GFS E GFS2

Já que o lançamento do Red Hat Enterprise Linux 6 não suporta sistemas de arquivos GFS, você deve atualizar quaisquer sistemas de arquivos GFS existentes para sistemas de arquivos GFS2 com o comando `gfs2_convert`. Note que você deve realizar este procedimento de conversão em um sistema Red Hat Enterprise 5 antes de atualizar para o Red Hat Enterprise Linux 6.



### ATENÇÃO

Antes de converter o sistema de arquivo GFS, você deve fazer um backup do sistema de arquivo, já que o processo de conversão é irreversível e quaisquer erros encontrados durante a conversão podem resultar numa interrupção brusca do programa e conseqüentemente um sistema de arquivo inutilizável.

Antes de converter o sistema de arquivos GFS, você deve usar o comando `gfs_fsck` para checar o sistema de arquivos e consertar quaisquer erros.

Se a conversão do GFS para o GFS2 é interrompida por uma falha de força ou qualquer outro problema, reinicie a ferramenta de conversão. Não tente executar o comando `fsck.gfs2` no sistema de arquivos até que a conversão estiver completa.

Quando converter sistemas de arquivos cheios ou quase cheios, é possível que não haja espaço suficiente disponível para colocar todas as estruturas do sistema de arquivos GFS2. Nesses casos, os tamanhos dos diários são reduzidos uniformemente de maneira que tudo caiba no espaço disponível.

### B.1. CONVERSÃO DE NOMES DE CAMINHOS DEPENDENTES DE CONTEXTO (CDPNS)

Os sistemas de arquivos do GFS2 não fornecem suporte para os Nomes de Caminho de Contexto Dependente (CDPNs), o qual permitem que você crie links simbólicos que apontam para os arquivos ou diretórios de destino de variantes. Para esta funcionalidade do GFS2, use a opção `bind` do comando `mount`.

O comando `gfs2_convert` identifica o CDPNs e os substitui com diretórios vazios com o mesmo nome. Para configurar as montagens de vinculação para substituir os CDPNs, entretanto, você precisa saber os caminhos completos para os links alvos dos CDPNs que você está substituindo. Antes de converter seu sistema de arquivos, você pode usar o comando `find` para identificar os links.

O seguinte comando lista os symlinks que apontam para um CDPN `hostname`:

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

Similarmente, você pode executar o comando `find` para outros CDPNs (`mach`, `os`, `sys`, `uid`, `gid`, `jid`). Note que já que os nomes CDPN podem ser na forma `@hostname` ou `{hostname}`, você precisará rodar o comando `find` para cada variante.

Para maiores informações sobre montagens bind e nomes de caminhos dependentes de contexto no GFS2, veja a [Seção 4.12, “Montagens Bind e Nomes de Caminho de Contexto Dependente.”](#) .

## B.2. PROCEDIMENTO DA CONVERSÃO DO GFS PARA GFS2

Use o seguinte procedimento para converter um sistema de arquivos GFS para um sistema de arquivos GFS2.

1. Em um sistema Red Hat Enterprise Linux, faça um backup de seu sistema de arquivos GFS existente.
2. Desmontar o sistema de arquivos GFS de todos os nós no cluster.
3. Executar o comando `gfs_fsck` no sistema de arquivo GFS para assegurar que o sistema de arquivos não foi corrompido.
4. Execute o `gfs2_convert gfsfilesystem`. O sistema irá exibir avisos e questões de confirmação antes de converter o `gfsfilesystem` para GFS2.
5. Atualize o Red Hat Enterprise Linux 6.

O seguinte exemplo converte um sistema de arquivos GFS em um dispositivo de bloco `/dev/shell_vg/500g` para um sistema de arquivos 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.
```

```
Examinando sistema de arquivo.....
```

```
Este programa irá converter um sistema de arquivo gfs1 para um sistema de
arquivo gfs2.
```

```
AVISO: Esta ação não poderá ser desfeita. Sugerimos que você:
```

1. Faça um Backup de todo o seu sistema de arquivo primeiro.
2. Execute o `gfs_fsck` primeiro para garantir a integridade do sistema de arquivo.
3. Assegure-se de que o sistema de arquivo NÃO foi montado de nenhum nó.
4. Assegure-se de que você possui as versões do software mais recentes.

```
Converta /dev/shell_vg/500g a partir do GFS1 para o GFS2? (y/n)y
```

```
Convertendo grupos de recursos.....
```

```
Convertendo inodes.
```

```
24208 inodes a partir do 1862 rgs convertido.
```

```
Reparando arquivo e informações de diretório.
```

```
18 cdpn symlinks mudaram para diretórios vazios.
```

```
Convertendo agendas.
```

```
Convertendo espaço de agendas para espaço rg .
```

```
Gravando agenda #1...concluído.
```

```
Gravando agenda #2...concluído.
```

```
Gravando agenda #3...concluído.
```

```
Gravando agenda #4...concluído.
```

```
Construindo as estruturas de sistema de arquivo do GFS2.
```

```
Removendo estruturas de sistema de arquivo GFS1 obsoletas.
```

```
Gravando mudanças ao disco.
```

| /dev/shell\_vg/500g: sistema de arquivo convertido com sucesso para gfs2.

## APÊNDICE C. OS TRACEPOINTS E O ARQUIVO DEBUGFS GLOCKS

Este apêndice descreve ambas interfaces do `glock debugfs` e dos tracepoints do GFS2. Estes são direcionados a usuários avançados que estão familiarizados com o sistema de arquivo interno que gostariam de aprender mais sobre criação de um GFS2 e como depurar problemas específicos ao GFS2.

### C.1. TIPOS DE TRACEPOINT DO GFS2.

Existem atualmente três tipos de tracepoints de GFS2: o tracepoint `glock` (pronuncia-se "gee-lock"), os tracepoints `bmap` e tracepoints `log`. Estes podem ser usados para monitorar um sistema de arquivo GFS2 em execução e fornecer informações adicionais ao que pode ser obtido com as opções de depuração suportados nas versões anteriores do Red Hat Enterprise Linux. Tracepoints são particularmente úteis quando um problema, tal como uma questão de trava ou desempenho, é reprodutível e, portanto, o resultado do tracepoint pode ser obtido durante uma operação problemática. Em GFS2, Glocks são os mecanismos de controle de cache primários e eles são a chave para entender o desempenho do núcleo do GFS2. Os (blocos mapa) tracepoints BMAP podem ser usados para monitorar as alocações de blocos e mapeamento de bloco (pesquisa de blocos já alocados na árvore de metadados em disco), a medida que acontecem e verificar quaisquer questões relacionadas com a localidade de acesso. Os tracepoints `log` mantém o controle dos dados que estão sendo gravados e liberados do diário e podem fornecer informações úteis sobre esta parte do GFS2.

Os tracepoints são projetados para serem o mais genérico possível. Isto deveria significar que não será necessário alterar a API no decurso do Red Hat Enterprise Linux 6. Por outro lado, os usuários dessa interface devem estar cientes de que esta é uma interface de depuração e não faz parte do Red Hat Enterprise Linux 6 API normal, e, como tal, a Red Hat não garante que as mudanças na interface dos tracepoints do GFS2 não ocorrerão.

Tracepoints são recursos genéricos do Red Hat Enterprise Linux 6 e seu alcance vai muito além do GFS2. Em particular, eles são utilizados para implementar a infraestrutura do `blktrace` e os tracepoints `blktrace` podem ser usados em combinação com os de GFS2 para obter uma imagem mais completa do desempenho do sistema. Devido ao nível em que os pontos de rastreamento operam, eles podem produzir grandes volumes de dados num curto período de tempo. Eles são criados para colocar uma carga mínima no sistema quando são ativadas, mas é inevitável que vão ter algum efeito. Filtragem de eventos através de uma variedade de meios que podem ajudar a reduzir o volume de dados e ajudar focar em obter apenas a informação que será útil para compreender qualquer situação em específico.

### C.2. TRACEPOINTS

Os tracepoints podem ser encontrados sob os diretórios `/sys/kernel/debug/tracing/` assumindo-se que `debugfs` é montado em um local padrão no diretório `/sys/kernel/debug`. O subdiretório `events` contém todos os eventos de rastreamento que podem ser especificados e considerando-se que o módulo `gfs2` esteja carregado, haverá um subdiretório `gfs2` que contém subdiretórios futuros, um para cada evento do GFS2. O conteúdo do diretório `/sys/kernel/debug/tracing/events/gfs2` deve se parecer com o seguinte:

```
[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
```

Para ativar todos os tracepoints do GFS2, execute o seguinte comando:

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

Para habilitar um tracepoint específico, existe um arquivo `enable` em cada um dos subdiretórios de eventos individuais. O mesmo é verdadeiro para o arquivo `filter` que pode ser utilizado para configurar um filtro de evento para cada evento ou conjunto de eventos. O significado dos eventos individuais é explicado em mais detalhes abaixo.

O resultado dos tracepoints está disponível em formato ASCII ou binário. Este apêndice não cobre a interface binária. A interface ASCII está disponível em duas formas. Para listar o conteúdo atual do buffer de anel, você pode executar o seguinte comando:

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

Esta interface é útil em casos que você estiver usando um processo de longa duração para um determinado período de tempo e, depois de algum evento, queira procurar pela última informação capturada no buffer. Uma interface alternativa, `/sys/kernel/debug/tracing/trace_pipe`, pode ser usado quando todo o resultado é necessário. Eventos são lidos a partir desse arquivo à medida que ocorrem, não há informações históricas disponíveis através desta interface. O formato de resultado é o mesmo de ambas as interfaces e é descrito para cada um dos GFS2 eventos nas seções deste apêndice.

O utilitário chamado `ttrace-cmd` está disponível para leitura de dados do tracepoint. Para mais informações sobre este utilitário, consulte o link em [Seção C.10, “Referências”](#). O utilitário `ttrace-cmd` pode ser utilizado de uma forma semelhante ao utilitário `strace`, por exemplo, para executar um comando enquanto reúne dados de rastreamento de diversas fontes.

### C.3. GLOCKS

Para entender GFS2, o conceito mais importante a entender, e que o distingue de outros sistemas de arquivos, é o conceito de Glocks. Em termos de código-fonte, uma glock é uma estrutura de dados que reúne a DLM e cache em uma máquina de estado único. Cada glock tem uma relação de 1:1 com um único bloqueio DLM, e fornece armazenamento em cache para aquele estado de bloqueio, assim as operações repetitivas realizadas a partir de um único nó do sistema de arquivos não precisam chamar repetidamente a DLM, e portanto ajudam a evitar tráfego de rede desnecessário. Existem duas amplas categorias dos Glocks, aqueles que realizam cache de metadados e aquelas que não. Os glocks inode e o grupo de recursos Glocks, ambos que realizam cache de metadados, outros tipos de Glock não realizam o cache de metadados. O inode Glock também está envolvido na realização de cache de dados, além de metadados e tem a lógica mais complexa de todos os Glocks.

Tabela C.1. Modos de glock e Modos de Bloqueio de DLM

Modo de GLock	Modo de bloqueio do DLM	Notas
UN	IV/NL	Desbloqueado (nenhum bloqueio DLM associado ao glock ou bloqueio de NL dependendo da bandeira I)
SH	PR	Bloqueio Compartilhado (leitura protegida)
EX	EX	Bloqueio exclusivo

Modo de GLock	Modo de bloqueio do DLM	Notas
DF	CW	Deferido (gravação simultânea) usada para o Direct I/O e congelamento de sistema de arquivo

Glocks permanecem na memória até que eles sejam desbloqueados (a pedido de outro nó, ou a pedido da VM) e não há usuários locais. Nesse ponto, eles são removidos da tabela hash glock e libertados. Quando um glock é criado, o bloqueio DLM não é associado com a Glock imediatamente. O bloqueio DLM torna-se associado com a glock no primeiro pedido para o DLM, e se esse pedido for bem sucedido, então a bandeira 'I' (inicial) será definida no glock. [Tabela C.4, “Sinalizadores Glock”](#) mostra os significados das bandeiras diferentes do glock. Depois da DLM ter sido associada ao Glock, o bloqueio DLM permanecerá sempre em pelo menos no modo de bloqueio NL (Nulo) até que o Glock seja libertado. Um rebaixamento de bloqueio DLM a partir de NL para desbloqueado é sempre a última operação na vida de um glock.



#### NOTA

Este aspecto específico do comportamento de bloqueio DLM mudou desde o Red Hat Enterprise Linux 5, o que às vezes desbloqueia o DLM anexado ao glock completamente, e portanto o Red Hat Enterprise Linux 5 tem um mecanismo diferente para garantir que LVBs (bloqueio de blocos de valor) são preservados onde for necessário. O novo esquema que o Red Hat Enterprise Linux 6 utiliza tornou-se possível devido à fusão do módulo de bloqueio `lock_dlm` (para não ser confundido com o próprio DLM) em GFS2.

Cada Glock pode ter um número de "holders" associados, cada um dos quais representa uma solicitação de bloqueio a partir das camadas mais altas. Chamadas do sistema relacionadas com a fila do GFS2 e holders que retiram da fila do glock para proteger a seção crítica do código.

A máquina de estado do glock é baseada em fila de trabalho. Por razões de desempenho, tasklets deve ter preferência. No entanto, na atual implementação precisamos submeter I/O daquele contexto que proíbe seu uso.



#### NOTA

Filas de trabalho possuem seus próprios tracepoints que podem ser utilizados em combinação com os tracepoints do GFS2, caso seja necessário.

[Tabela C.2, “Modos de Glock e Tipos de Dados”](#) mostra qual o estado que pode ser armazenado em cada um dos modos de glock e se esse estado em cache pode estar sujo. Isso se aplica tanto ao inode quanto para bloqueios de grupos de recursos, embora não haja nenhum componente de dados para as travas do grupo de recursos, somente os metadados.

**Tabela C.2. Modos de Glock e Tipos de Dados**

Modo de GLock	Cache Dados	Cache Metadado	Dados Sujos	Metadados Sujos
UN	Não	Não	Não	Não
SH	Sim	Sim	Não	Não



Modo de GLock	Cache Dados	Cache Metadado	Dados Sujos	Metadados Sujos
DF	Não	Sim	Não	Não
EX	Sim	Sim	Sim	Sim

## C.4. A INTERFACE DE DEBUGFS DO GLOCK

A interface do glock `debugfs` permite a visualização do estado interno dos Glocks e os "holders" e também inclui alguns detalhes em síntese dos objetos que estão sendo travados em alguns casos. Cada linha do arquivo ou começa G: sem recuo (que se refere ao próprio glock) ou começa com uma letra diferente, recuado com um único espaço, e refere-se às estruturas associadas ao glock imediatamente acima dele no arquivo (H: é um holder, I: um inode, e R: um grupo de recursos). Aqui está um exemplo de como o conteúdo deste arquivo pode parecer:

```
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]
```

O exemplo acima é uma série de trechos (a partir de um arquivo de aproximadamente 18MB) gerado pelo `cat/sys/kernel/debug/gfs2/unity:myfs/glocks >my.lock` durante uma execução do `postmark benchmark` em um sistema de arquivos GFS2 de nó único. Os glocks na figura foram selecionados, a fim de mostrar algumas das características mais interessantes dos despejos do glock.

A estados de glock são ou EX (exclusive), DF (diferido), SH (compartilhado) ou UN (desbloqueado). Esses estados correspondem diretamente aos modos de bloqueio DLM, exceto para UN, que pode representar o estado de bloqueio DLM nulo, ou que GFS2 não mantém um bloqueio DLM (dependendo da bandeira I, como explicado acima). O campo s: do glock indica o estado actual do bloqueio e o mesmo campo no holder indica o modo requerido. Se o bloqueio for concedido, o holderr terá o bit H definido em suas bandeiras (f: campo). Caso contrário, ele terá o conjunto de bits de espera W.

O campo n: (número) indica o número associado a cada item. Para glocks, este é o número do tipo seguido pelo número glock, assim, no exemplo acima, o primeira glock é n: 5/75320, isto é, um glock **Iopen**, que se relaciona com Inode 75320. No caso do inode e glocks **Iopen**, o número GLOCK é sempre idêntico ao número do bloco de disco do inode.



## NOTA

Os números glock (n: campo) no arquivo de glocks debugfs, estão em hexadecimal, enquanto que o resultado do tracepoints lista-os em decimal. Isto acontece por razões históricas, números glock foram sempre escritos em hexadecimal, mas decimal foi escolhido para os pontos de rastreamento, para que os números pudessem ser facilmente comparados com o outro resultado de tracepoint (a partir do `blktrace` por exemplo) e com resultado de `stat(1)`.

Uma listagem completa de todas as bandeiras para ambos holder e glock são definidas em [Tabela C.4, “Sinalizadores Glock”](#) e [Tabela C.5, “Sinalizadores do proprietário glock”](#). O conteúdo de blocos de valor de bloqueio não está disponível no momento via interface `debugfs` do glock.

A [Tabela C.3, “Tipos de Glock”](#) mostra os significados dos diferentes tipos de glock.

**Tabela C.3. Tipos de Glock**

Dígite o número	Tipo de bloqueio	Use
1	trans	Bloqueio de transação
2	inode	Dados e metadados inode
3	rgrp	Metadados de grupos de recursos
4	meta	O superblock
5	iopen	Detecção mais próxima do último inode
6	flock	<code>flock(2)</code> syscall
8	quota	Operações de quota
9	Diário	Diário mutex

Uma das bandeiras mais importantes glock é a bandeira I (bloqueado). Isto é o bit de bloqueio, que é utilizado para arbitrar o acesso ao estado Glock quando uma mudança de estado está a ser executada. Ele é definido quando a máquina de estado está prestes a enviar um pedido de bloqueio remoto via a DLM, e só apagada quando a operação toda tenha sido realizada. Às vezes, isso pode significar que mais do que um pedido de bloqueio terá sido enviado, com várias invalidações ocorrendo de tempo em tempo.

[Tabela C.4, “Sinalizadores Glock”](#) exhibe o significado das bandeiras de glock diferentes.

**Tabela C.4. Sinalizadores Glock**

Sinalizador	Nome	Significado
d	Rebaixamento pendente	Um pedido (remoto) de rebaixamento deferido

Sinalizador	Nome	Significado
D	Rebaixamento	Um pedido de rebaixamento (local ou remoto)
f	Descarga de Log	O log precisa ser enviado antes de liberar esse glock
F	Congelado	Respostas de nós remotos ignorados - recuperação em progresso.
i	Invalidação em progresso	No processo de estar invalidando páginas sob este glock.
I	Inicial	Ajuste quando o bloqueio DLM é associado com este glock
l	Bloqueado	O glock está no processo de mudança de estado
L	LRU	Definido quando o glock está na lista do LRU
o	Object	Definido quando o glock é associado com um objeto (ou seja, um inode para glocks tipo 2, e um grupo de recurso para glocks tipo 3)
p	Rebaixamento em progresso	O glock está no processo de responder a um pedido de rebaixamento
q	Em fila	Definido quando um proprietário é enfileirado em um glock e limpo quando um glock é mantido, mas não houver nenhum outro proprietário. Usado como parte do algoritmo que calcula um tempo mínimo de espera para um glock.
r	Resposta pendente	Resposta recebida do nó remoto está aguardando processamento
y	Sujo	Os dados precisam descarregar para o disco antes de liberar este glock

Quando uma chamada remota é recebida de um nó que deseja obter um bloqueio em um modo que entra em conflito com o que foi realizado no nó local, então uma das duas bandeiras D (rebaixar) ou d (rebaixar pendente) é definida. A fim de evitar condições de desespero, quando existe um bloqueio de contenção em particular, cada bloqueio recebe um tempo mínimo de espera. Um nó que ainda não tenha tido o bloqueio para o tempo mínimo de espera é permitido para manter aquele bloqueio até que o intervalo de tempo tenha expirado.

Se o intervalo de tempo expirou, então a bandeira D (demote) será definida e o estado requerido será gravado. Nesse caso, na próxima vez que não houver bloqueios concedidos na fila de holders, o bloqueio será rebaixado. Se o intervalo de tempo não houver expirado, a bandeira d (demote pending) será definida em seu lugar. Isso também agenda a máquina de estado para limpar o d (demote pending) e definir o D (demote), quando o tempo mínimo de espera expirar.

A bandeira I (initial) é definida quando o glock recebe um bloqueio do DLM. Isto acontece quando o glock é utilizado pela primeira vez e a bandeira I fica definida até que o glock é finalmente liberado (quando o bloqueio do DLM é desbloqueado).

## C.5. GLOCK HOLDERS

Tabela C.5, “Sinalizadores do proprietário glock” exhibe os significados de diferentes bandeiras de glock holders..

Tabela C.5. Sinalizadores do proprietário glock

Sinalizador	Nome	Significado
a	Async	Não espere pelo resultado glock (fará um conjunto para enviar depois)
A	Qualquer	Qualquer modo de bloqueio compatível é aceitável
c	Sem cache	Quando desbloqueado, rebaixe o bloqueio DLM imediatamente
e	Não expire	Ignore pedidos de cancelamento de bloqueios subsequentes
E	exato	Deve ter um modo de bloqueio exato
F	Primeiro	Ajuste quando o proprietário é o primeiro a ser concedido para este bloqueio
H	Detentor	Indica que o bloqueio requerido é concedido
p	Prioridade	Enfileire os proprietários no início da fila
t	Tentar	Uma "tentativa" de bloqueio
T	Tentar 1CB	Uma "tentativa" que envia um callback
W	Espera	Ajuste enquanto espera por um pedido para completar

As bandeiras holders mais importantes são H (holder) e W (wait) como mencionado anteriormente, uma vez que elas são definidas dependendo das solicitações de bloqueio concedidos e as solicitações de bloqueio em fila de espera, respectivamente. A ordenação dos holders na lista é importante. Se houver qualquer holder concedido, eles estarão sempre no topo da fila, seguido por quaisquer holders em fila de espera.

Se não houver holders concedidos, o primeiro holder da lista será aquele que desencadeia a próxima mudança de estado. Como as solicitações de rebaixamento (demote) sempre são considerados prioridade maior do que os pedidos do sistema de arquivos, estas podem nem sempre resultar diretamente em uma mudança para o estado requerido.

O subsistema do glock suporta dois tipos de bloqueio de "experimento". Eles são úteis tanto porque permitem a obtenção de bloqueios fora da ordem normal (com um adequado back-off e repetição)

como também porque pode ser usado para ajudar a evitar a utilização de recursos por outros nós. O bloqueio `t` (`try`) normal, é basicamente, apenas o que seu nome indica, é uma "tentativa" de bloqueio, que não faz nada de especial. O bloqueio `T` (`try 1CB`) por outro lado, é idêntico ao bloqueio `t`, exceto que o DLM irá enviar uma única chamada de retorno para os bloqueios holders de correntes incompatíveis. Uma das utilidades do bloqueio `T` (`try 1CB`) é com os bloqueios `iopen`, que são utilizados para arbitrar entre os nós quando a contagem de um `i_nlink` do inode é zero, e determinar qual dos nós será responsável por desalocar o inode. O glock `iopen` é normalmente realizado no estado compartilhado, mas quando a contagem do `i_nlink` torna-se zero e `->delete_inode()` é chamado, ele vai solicitar um bloqueio exclusivo com o conjunto do `T` (`try 1CB`). Ele continuará a desalocar o inode se o bloqueio for concedido. Se o bloqueio não for concedido o resultado será o(s) nó(s) que estavam impedindo a concessão do bloqueio marcando seu(s) glock(s) com a bandeira `D` (`demote`), que é verificada em `->drop_inode()` tempo, a fim de assegurar que o desalocação não seja esquecida.

Isso significa que inodes que possuem contagem de links zero, mas ainda estão abertos será desalocado pelo nó em que ocorre o último `close()`. Além disso, ao mesmo tempo que a contagem da ligação do inode é decrementado para zero o inode é marcado como estando no estado especial de possuir zero de contagem da link, mas ainda em uso no bitmap do grupo de recursos. Este funciona como lista órfã do arquivo `ext3 system3` na medida em que permite que qualquer leitor subsequente do bitmap saiba que existe espaço em potencial que pode ser recuperado, e tentar recuperá-lo.

## C.6. TRACEPOINTS DO GLOCK

Os tracepoints também são projetados para serem capazes de confirmar a exatidão do controle de cache, combinando-os com o resultado do `blktrace` e o conhecimento da disposição em disco. Em seguida, é possível verificar se qualquer `I/O` foi emitido e concluído sob o bloqueio correto, e que nenhum traço esteja presente.

O tracepoint `gfs2_glock_state_change` é o mais importante de se entender. Ele rastreia cada mudança de estado do glock a partir da criação inicial até o final do rebaixamento que finaliza com o `gfs2_glock_put` e o final `NL` para desbloquear a transição. A bandeira `I` (`locked`) do glock é sempre definida antes da mudança de estado ocorrer e não será explicada até que ela seja finalizada. Não existem holders obtidos (a bandeira `glock holder H`) durante uma mudança de estado. Caso existam holders em fila, eles estarão sempre em `W` (`waiting`). Quando a mudança de estado estiver concluída, os holders podem receber direitos, o que é o final da operação antes da bandeira do glock `I` ser removida.

O tracepoint `gfs2_demote_rq` mantém o controle de pedidos de rebaixamento local e remoto. Assumindo que não há memória suficiente no nó, os pedidos de rebaixamento local raramente serão vistos, e na maioria das vezes serão criados por `umount` ou pela recuperação de memória ocasional. O número de pedidos de rebaixamento remoto é uma medida de luta entre nós por um inode em particular ou grupo de recursos.

Quando um holder recebe um bloqueio, o `gfs2_promote` é chamado, isto ocorre a medida que as etapas finais de uma mudança de estado se aproximam, ou ainda quando um bloqueio é requisitado, o qual pode ser obtido imediatamente devido ao estado do glock estar realizando um cache de um bloqueio de um modo apropriado. Se o holder for o primeiro a receber para este glock, então a bandeira `f` (`flag`) será definida neste holder. Isto é utilizado atualmente somente pelos grupos de recurso.

## C.7. TRACEPOINTS DE BMAP

Mapeamento de bloco é uma tarefa fundamental para qualquer sistema de arquivos. GFS2 usa um sistema tradicional baseado em bitmap com dois bits por bloco. O principal objetivo dos tracepoints neste subsistema é permitir o monitoramento do tempo necessário para alocar e blocos do mapa.

O tracepoint `gfs2_bmap` é chamado duas vezes para cada operação `bmap`: uma vez no início para

apresentar o pedido `bmap`, e uma vez no final para exibir o resultado. Isto facilita ao coincidir com as solicitações e resultados e medir o tempo necessário para mapear blocos em diferentes partes do sistema de arquivos, diferentes deslocamentos de arquivo, ou até mesmo de arquivos diferentes. Também é possível ver qual a média de tamanhos de extensão retornados comparado àqueles que estão sendo solicitados.

Para manter o controle de blocos alocados, o `gfs2_block_alloc` é chamado não somente em alocações, mas como também em liberações de blocos. Como as alocações são todas referenciadas de acordo com os inodes para qual o bloco é pretendido, isto pode ser usado para rastrear quais blocos físicos pertencem à quais arquivos em um sistema de arquivo ao vivo. Isto é particularmente útil quando combinado com o `blktrace`, que exibirá modelos de I/O problemáticos, que podem ser referenciados de volta à inodes relevantes, utilizando o mapeamento obtido através deste tracepoint.

## C.8. TRACEPOINTS DOS LOGS

Os tracepoints no subsistema rastreiam blocos que são adicionados e removidos do diário (`gfs2_pin`), assim como tempo que o comprometimento de transações levou para o log (`gfs2_log_flush`). Isto pode ser bem útil ao tentar depurar os problemas de desempenho do diário.

O tracepoint `gfs2_log_blocks` mantém histórico dos blocos reservados no log, que podem ajudar a mostrar se o log é muito pequeno para a carga de trabalho, por exemplo.

O tracepoint `gfs2_ail_flush` (Red Hat Enterprise Linux 6.2 e posteriores) é semelhante ao tracepoint `gfs2_log_flush`, ou seja, observa o início e final de fluxos das listas AIL. Estas listas contém buffers que passaram pelo log, mas ainda não foram gravadas de volta em seu local e este é esvaziado para lançar mais espaço de log para ser usado pelos filesystems, ou quando um processo requer uma sincronização ou `fsync`.

## C.9. ESTATÍSTICA DE GLOCK

GFS2 mantém as estatísticas que podem ajudar rastrear o que está acontecendo dentro do sistema de arquivo. Isto permite que você identifique os problemas de desempenho.

GFS2 mantém dois contadores:

- `dcount`, que conta o número das operações de DLM requisitadas. Isto mostra quantos dados foram para o cálculo de média/variação.
- `qcount`, que conta o número de operações de nível do `syscall` requisitada. Geralmente, `qcount` será igual ou maior do que `dcount`.

Além disso, o GFS2 mantém três pares de média/variação. Os pares de média/variação são estimações exponenciais suavizadas e o algoritmo usado é um dos utilizados para calcular um número redondo de vezes em código de rede. Os pares de média e variação mantidos em GFS2 não são escalados, mas estão em unidades de nanosegundos inteiros.

- `srtt/srttvar`: Número redondo de vezes suavizados para operações sem bloqueio.
- `srttb/srttvarb`: Número redondo de vezes suavizados para operações de bloqueio.
- `irtt/irttvar`: Tempo de requisição interna (por exemplo, tempo entre as requisições de DLM)

Uma requisição sem bloqueio é uma que será concluída imediatamente, seja o estado de bloqueio de DLM que esteja em questão. Isto geralmente significa quaisquer requisições quando (a) o estado atual do bloqueio é exclusivo (b) o estado requisitado é nulo ou não foi bloqueado ou (c) a sinalização "tentar

bloqueio" foi definida. Uma requisição de bloqueio cobre todas as outras requisições de bloqueio.

Tempos mais longos são melhores para o IRTTs, e tempos mais curtos são melhores para o RTTs.

As estatísticas são mantidas em dois arquivos sysfs:

- O arquivo `glstats`. Este arquivo é semelhante ao arquivo `glocks`, exceto pelo fato de conter estatísticas, com um glock por linha. Os dados são inicializados a partir dos dados "per cpu", para aquele tipo de glock para o qual o glock foi criado (além dos contadores, que são zerados).
- O arquivo `lkstats`. Isto contém a stats do "per cpu" para cada tipo de glock. Ele contém uma estatística por linha, na qual cada coluna é um núcleo de cpu. Existem oito linhas por tipo de glock, com tipos seguindo um depois do outro.

## C.10. REFERÊNCIAS

Para mais informações sobre tracepoints e arquivo GFS2 `glocks`, consulte os seguintes recursos:

- Este apêndice foi parcialmente adaptado de um documento entregue por Steve Whitehouse no Linux Symposium 2009, o qual pode ser encontrado em <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>.
- Para informações sobre as regras de bloqueio internas do glock, veja <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>.
- Para informações sobre rastreamento de evento, veja <http://git.kernel.org/?p=linux/kernel/git/torvalds/linux-2.6.git;a=blob;f=Documentation/trace/events.txt;h=09bd8e9029892e4e1d48078de4d076e24e>.
- Para informações sobre o utilitário `trace-cmd` veja <http://lwn.net/Articles/341902/>.

## APÊNDICE D. HISTÓRICO DE REVISÕES

<b>Revisão 7.1-3.2</b> pt-BR is completed	<b>Tue Feb 10 2015</b>	<b>Glaucia Cintra</b>
<b>Revisão 7.1-3.1</b> Tradução de arquivos sincronizados com a versão 7.1-3 de fontes do XML	<b>Tue Feb 10 2015</b>	<b>Glaucia Cintra</b>
<b>Revisão 7.1-3</b> Atualizando para implementar o sort_order na página inicial do RHEL 6	<b>Tue Dec 16 2014</b>	<b>Steven Levine</b>
<b>Revisão 7.0-9</b> Versão de lançamento GA 6.6	<b>Wed Oct 8 2014</b>	<b>Steven Levine</b>
<b>Revisão 7.0-8</b> Lançamento da versão 6.6 Beta	<b>Thu Aug 7 2014</b>	<b>Steven Levine</b>
<b>Revisão 7.0-4</b> Resolve #1102591 Adiciona o procedimento para configurar o GFS2 em um cluster de Pacemaker	<b>Thu Jul 17 2014</b>	<b>Steven Levine</b>
<b>Revisão 7.0-3</b> Resolve #1035119 Atualiza a tabela das sinalizações do Glock e adiciona um seção nas estatísticas do Glock	<b>Wed Jul 16 2014</b>	<b>Steven Levine</b>
<b>Revisão 7.0-1</b> Primeiro rascunho para o lançamento do 6.6	<b>Thu Jun 5 2014</b>	<b>Steven Levine</b>
<b>Revisão 6.0-6</b> Versão de lançamento GA 6.5	<b>Wed Nov 13 2013</b>	<b>Steven Levine</b>
<b>Revisão 6.0-5</b> Versão de lançamento Beta 6.5	<b>Fri Sep 27 2013</b>	<b>Steven Levine</b>
<b>Revisão 6.0-3</b> Resolve #960841 Explica a falta de suporte para o SELinux com sistemas de arquivo do GFS2.	<b>Fri Sep 27 2013</b>	<b>Steven Levine</b>
<b>Revisão 6.0-1</b> Adicionando a nota sobre o Samba e GFS2	<b>Fri Sep 06 2013</b>	<b>Steven Levine</b>
<b>Revisão 5.0-7</b> Versão para o lançamento do GA 6.4	<b>Mon Feb 18 2013</b>	<b>Steven Levine</b>
<b>Revisão 5.0-5</b> Lançamento da versão 6.4 Beta	<b>Mon Nov 26 2012</b>	<b>Steven Levine</b>
<b>Revisão 5.0-4</b> Resolve #860324 Atualiza capítulo na configuração do GFS2 e Considerações Operacionais com pequenas explicações.  Resolve #807057 Adiciona a nota recomendando consulta com um representante autorizado Red Hat para verificar suas configurações antes da implementação.	<b>Tue Nov 13 2012</b>	<b>Steven Levine</b>
<b>Revisão 5.0-1</b>	<b>Mon Oct 15 2012</b>	<b>Steven Levine</b>





# ÍNDICE REMISSIVO

## A

adicionando diários no sistema de arquivo, [Adicionando Diários ao Sistema de Arquivo](#)

ajuste de desempenho, [Ajuste de Desempenho com GFS2](#)

ajuste, desempenho, [Ajuste de Desempenho com GFS2](#)

atime, configurando atualizações, [Configurando atualizações do atime](#)

montando com noatime, [Montar com o relatime](#), [Montar com noatime](#)

aumentando um sistema de arquivo, [Aumentando um Sistema de Arquivo](#)

## B

bandieras do glock holder, [Glock Holders](#)

Bloqueio de Posix, [Problemas com o Bloqueio de Posix](#)

## C

comando de montagens, [Montagem de Sistema de Arquivo](#)

comando desmontar, [Desmontando o Sistema de Arquivo](#)

comando fsck.gfs2, [Reparando um Sistema de Arquivo](#)

comando gfs2\_grow, [Aumentando um Sistema de Arquivo](#)

comando gfs2\_quota, [Gerenciamento de Quota do GFS2 com o comando gfs2\\_quota](#)

comando mkfs, [Criando um Sistema de Arquivo](#)

comando quotacheck

verificando a precisão de quotas com, [Mantendo Quotas Precisar](#)

configuração, antes, [Antes de Configurar o GFS2](#)

configuração, inicial, [Iniciando](#)

tarefas de pré-requisitos, [Tarefas de Pré-requisitos:](#)

configurações, iniciais

tarefas iniciais, [Tarefas Iniciais de Configuração](#)

Considerações de configuração, [Configuração do GFS2 e Considerações Operacionais](#)

Context-Dependent Path Names (CDPNs)

Conversão do GFS para GFS2, [Conversão de Nomes de Caminhos Dependentes de Contexto \(CDPNs\)](#)

criando um sistema de arquivo, [Criando um Sistema de Arquivo](#)

## D

debugfs, [Os tracepoints e o Arquivo debugfs glocks](#)

debugfs file, [Resolução de Problemas do Desempenho do Despejo de Bloqueios GFS2](#)

desmonta, trava o sistema, [Considerações Especiais ao Montar os Sistemas de Arquivo GFS2](#)

desmontando o sistema de arquivo, [Desmontando o Sistema de Arquivo](#), [Considerações Especiais ao Montar os Sistemas de Arquivo GFS2](#)

diário de dados, [Diário de Dados](#)

## F

feedback

informação de contato para este manual, [Precisamos de seu Feedback](#)

função de retirada, GFS2, [A Função de Retirada do GFS2](#)

## G

gerenciamento de cota, [Gerenciamento de Cota do GFS2](#), [Definindo Quotas no Modo Enforcement ou Accounting](#)

sincronizando cotas, [Sincronizando Quotas com o Comando quotasync](#)

gerenciamento de quotas, [Gerenciamento de Quota do GFS2 com o comando gfs2\\_quota](#)

ativando quotas accounting, [Ativando Quotas Accounting](#)

ativando/desativando imposição de quotas, [Ativando/Desativando a Imposição de Quotas](#)

definição de quotas, [Definindo Quotas com o comando gfs2\\_quota](#)

exibindo limites de quotas, [Exibindo limites de quotas e uso com o comando gfs2\\_quota](#)

sincronizando quotas, [Sincronizando Quotas com o Comando gfs2\\_quota](#).

gerenciando o GFS2, [Gerenciando o GFS2](#)

## GFS2

atime, configurando atualizações, [Configurando atualizações do atime](#)

montando com noatime, [Montar com noatime](#)

montando com o relatime, [Montar com o relatime](#)

Configuration considerations, [Configuração do GFS2 e Considerações Operacionais](#)

função de retirada, [A Função de Retirada do GFS2](#)

gerenciamento de quotas, [Gerenciamento de Quota do GFS2 com o comando gfs2\\_quota](#)

gerenciamento de cota

sincronizando cotas, [Sincronizando Quotas com o Comando quotasync](#)

gerenciamento de quotas

ativando quotas accounting, [Ativando Quotas Accounting](#)

ativando/desativando imposição de quotas, [Ativando/Desativando a Imposição de Quotas](#)

definindo quotas, [Definindo Quotas com o comando gfs2\\_quota](#)

exibindo limites de quotas, [Exibindo limites de quotas e uso com o comando gfs2\\_quota](#)

sincronização de quotas, [Sincronizando Quotas com o Comando gfs2\\_quota](#).

gerenciamento do cota, [Gerenciamento de Cota do GFS2](#), [Definindo Quotas no Modo Enforcement ou Accounting](#)

gerenciando, [Gerenciando o GFS2](#)

Operation, [Configuração do GFS2 e Considerações Operacionais](#)

gfs2\_jadd command, [Adicionando Diários ao Sistema de Arquivo](#)

**glock**, [Os tracepoints e o Arquivo debugfs glocks](#)

**glock flags**, [Resolução de Problemas do Desempenho do Despejo de Bloqueios GFS2](#) , [A interface de debugfs do glock](#)

**glock holder flags**, [Resolução de Problemas do Desempenho do Despejo de Bloqueios GFS2](#)

**glock types**, [Resolução de Problemas do Desempenho do Despejo de Bloqueios GFS2](#) , [A interface de debugfs do glock](#)

## I

**introdução**, [Introdução](#)

**público alvo**, [Público Alvo](#)

## M

**mkfs.gfs2** tabela de opções de comando, [Opções Completas](#)

**montagens bind**, [Montagens Bind e Nomes de Caminho de Contexto Dependente.](#)

**ordem de montagem**, [Monta Binds e Organiza a Montagem de Sistema de Arquivo](#)

**montagens do sistema de arquivo**, [Montagem de Sistema de Arquivo](#), [Considerações Especiais ao Montar os Sistemas de Arquivo GFS2](#)

## N

**node locking**, [Bloqueio de Nó GFS2](#)

**nomes de caminho, dependente de contexto (CDPNs)**, [Montagens Bind e Nomes de Caminho de Contexto Dependente.](#)

## O

**opção de montagem acl**, [Montagem de Sistema de Arquivo](#)

**Opções de GFS2 específicos para expandir tabelas de sistemas de arquivo**, [Uso Completo](#)

**Opções do GFS2 específicas para adicionar tabelas de diários**, [Uso Completo](#)

## P

**parâmetro ajustável quota\_quantum**, [Sincronizando Quotas com o Comando quotasync](#) , [Sincronizando Quotas com o Comando gfs2\\_quota.](#)

**prefácio** (ver introdução)

**público alvo**, [Público Alvo](#)

## Q

**quota= mount option**, [Definindo Quotas com o comando gfs2\\_quota](#)

**quotacheck** , [Criando os Arquivos de Banco de Dados de Quotas](#)

**quotas de disco**

**atribuindo por grupo**, [Atribuindo Quotas por Grupo](#)

**atribuindo por usuário**, [Atribuindo Quotas por Usuário](#)

**gerenciamento de**, [Gerenciando Quotas de Disco](#)

**quotacheck comando, usando para checar**, [Mantendo Quotas Precisar](#)

**reportar**, [Gerenciando Quotas de Disco](#)

habilitando, [Configurando Quotas de Disco](#)

criando arquivos de quotas, [Criando os Arquivos de Banco de Dados de Quotas](#)

habilitar

quotacheck, execução, [Criando os Arquivos de Banco de Dados de Quotas](#)

limite de hard, [Atribuindo Quotas por Usuário](#)

limite de soft, [Atribuindo Quotas por Usuário](#)

recursos adicionais, [Referências](#)

## R

recursos, novos e alterados, [Recursos Novos e Alterados](#)

reparando um sistema de arquivo, [Reparando um Sistema de Arquivo](#)

## S

sistema de arquivo

adicionando diários, [Adicionando Diários ao Sistema de Arquivo](#)

atime, configurando atualizações, [Configurando atualizações do atime](#)

montando com noatime , [Montar com noatime](#)

montando com relatime , [Montar com o relatime](#)

aumentando, [Aumentando um Sistema de Arquivo](#)

criando, [Criando um Sistema de Arquivo](#)

desmontando, [Desmontando o Sistema de Arquivo](#), [Considerações Especiais ao Montar os Sistemas de Arquivo GFS2](#)

diário de dados, [Diário de Dados](#)

gerenciamento de cota

sincronizando cotas, [Sincronizando Quotas com o Comando quotasync](#)

gerenciamento do cota, [Gerenciamento de Cota do GFS2](#), [Definindo Quotas no Modo Enforcement ou Accounting](#)

monta bind, [Montagens Bind e Nomes de Caminho de Contexto Dependente.](#)

montagem, [Montagem de Sistema de Arquivo](#), [Considerações Especiais ao Montar os Sistemas de Arquivo GFS2](#)

nomes de caminho dependente do contexto (CDPNs), [Montagens Bind e Nomes de Caminho de Contexto Dependente.](#)

ordem de montagem, [Monta Binds e Organiza a Montagem de Sistema de Arquivo](#)

reparando, [Reparando um Sistema de Arquivo](#)

suspendendo atividade, [Suspendendo a Atividade em um Sistema de Arquivo](#)

sistema de arquivos

gerenciamento de quotas, [Gerenciamento de Quota do GFS2 com o comando gfs2\\_quota](#)

ativando quotas accounting, [Ativando Quotas Accounting](#)

ativando/desativando imposição de quotas, [Ativando/Desativando a Imposição de Quotas](#)

definição de quotas, [Definindo Quotas com o comando gfs2\\_quota](#)  
exibindo limites de quotas, [Exibindo limites de quotas e uso com o comando gfs2\\_quota](#)  
sincronizando quotas, [Sincronizando Quotas com o Comando gfs2\\_quota](#).

suspendendo a atividade em um sistema de arquivo, [Suspendendo a Atividade em um Sistema de Arquivo](#)

## T

### tabelas

as opções do GFS2 específico para expandir sistemas de arquivo, [Uso Completo](#)  
opções de montagem, [Uso Completo](#)  
Opções do GFS2 específicas para adicionar diários, [Uso Completo](#)

tabelas de montagens, [Uso Completo](#)

### tables

mkfs.gfs2 opções de comando, [Opções Completas](#)

tamanho máximo do sistema de arquivo GFS2, [Visão Geral do GFS2](#)

tamanho máximo, sistema de arquivos GFS2, [Visão Geral do GFS2](#)

### tarefas de pré-requisitos

configuração, inicial, [Tarefas de Pré-requisitos:](#)

### tarefas iniciais

configuração inicial, [Tarefas Iniciais de Configuração](#)

tracepoints, [Os tracepoints e o Arquivo debugfs glocks](#)

trava o sistema na desmontagem, [Considerações Especiais ao Montar os Sistemas de Arquivo GFS2](#)

## V

visão geral, [Visão Geral do GFS2](#)

configuração, anterior, [Antes de Configurar o GFS2](#)

recursos, novos e alterados, [Recursos Novos e Alterados](#)