Red Hat Enterprise Linux 8

Configuración de sistemas de archivos GFS2

Guía de configuración y gestión de los sistemas de archivos GFS2

Last Updated: 2021-03-08

Red Hat Enterprise Linux 8 Configuración de sistemas de archivos GFS2

Guía de configuración y gestión de los sistemas de archivos GFS2

Enter your first name here. Enter your surname here.
Enter your organisation's name here. Enter your organisational division here.
Enter your email address here.

Legal Notice

Resumen

Este manual proporciona información sobre la configuración y gestión de sistemas de archivos GFS2 para Red Hat Enterprise Linux 8.

Table of Contents

HACER QUE EL CÓDIGO ABIERTO SEA MÁS INCLUSIVO

Red Hat se compromete a sustituir el lenguaje problemático en nuestro código, documentación y propiedades web. Estamos empezando con estos cuatro términos: maestro, esclavo, lista negra y lista blanca. Debido a la enormidad de este esfuerzo, estos cambios se implementarán gradualmente a lo largo de varias versiones próximas. Para más detalles, consulte el mensaje de nuestro CTO Chris Wright .

PROPORCIONAR COMENTARIOS SOBRE LA DOCUMENTACIÓN DE RED HAT

Agradecemos su opinión sobre nuestra documentación. Por favor, díganos cómo podemos mejorarla. Para ello:

- Para comentarios sencillos sobre pasajes concretos:
  1. Asegúrese de que está viendo la documentación en el formato *Multi-page HTML*. Además, asegúrese de ver el botón **Feedback** en la esquina superior derecha del documento.
  2. Utilice el cursor del ratón para resaltar la parte del texto que desea comentar.
  3. Haga clic en la ventana emergente **Add Feedback** que aparece debajo del texto resaltado.
  4. Siga las instrucciones mostradas.
- Para enviar comentarios más complejos, cree un ticket de Bugzilla:
  1. Vaya al sitio web de Bugzilla.
  2. Como componente, utilice **Documentation**.
  3. Rellene el campo **Description** con su sugerencia de mejora. Incluya un enlace a la(s) parte(s) pertinente(s) de la documentación.
  4. Haga clic en **Submit Bug**.

CAPÍTULO 1. PLANIFICACIÓN DE LA IMPLANTACIÓN DE UN SISTEMA DE ARCHIVOS GFS2

El sistema de archivos Red Hat Global File System 2 (GFS2) es un sistema de archivos de cluster simétrico de 64 bits que proporciona un espacio de nombres compartido y gestiona la coherencia entre múltiples nodos que comparten un dispositivo de bloques común. Un sistema de archivos GFS2 pretende ofrecer un conjunto de características lo más parecido posible a un sistema de archivos local y, al mismo tiempo, reforzar la coherencia total del clúster entre los nodos. Para lograrlo, los nodos emplean un esquema de bloqueo en todo el clúster para los recursos del sistema de archivos. Este esquema de bloqueo utiliza protocolos de comunicación como TCP/IP para intercambiar información de bloqueo.

En algunos casos, la API del sistema de archivos de Linux no permite que la naturaleza agrupada de GFS2 sea totalmente transparente; por ejemplo, los programas que utilizan bloqueos POSIX en GFS2 deben evitar el uso de la función **GETLK**, ya que, en un entorno agrupado, el ID del proceso puede corresponder a un nodo diferente del clúster. Sin embargo, en la mayoría de los casos, la funcionalidad de un sistema de archivos GFS2 es idéntica a la de un sistema de archivos local.

El complemento de almacenamiento resistente de Red Hat Enterprise Linux (RHEL) proporciona GFS2 y depende del complemento de alta disponibilidad de RHEL para proporcionar la gestión de clústeres que requiere GFS2.

El módulo del kernel **gfs2.ko** implementa el sistema de archivos GFS2 y se carga en los nodos de cluster GFS2.

Para obtener el mejor rendimiento de GFS2, es importante tener en cuenta las consideraciones de rendimiento que se derivan del diseño subyacente. Al igual que un sistema de archivos local, GFS2 se basa en la caché de páginas para mejorar el rendimiento mediante el almacenamiento en caché local de los datos más utilizados. Para mantener la coherencia entre los nodos del clúster, el control de la caché lo proporciona la máquina de estado *glock*.

IMPORTANTE

Asegúrese de que su despliegue del complemento de alta disponibilidad de Red Hat satisface sus necesidades y puede ser soportado. Consulte con un representante autorizado de Red Hat para verificar su configuración antes de la implementación.

1.1. PARÁMETROS CLAVE DE GFS2 PARA DETERMINAR

Antes de instalar y configurar GFS2, tenga en cuenta las siguientes características clave de sus sistemas de archivos GFS2:

Nodos GFS2

Determine qué nodos del clúster montarán los sistemas de archivos GFS2.

Número de sistemas de archivos

Determina cuántos sistemas de archivos GFS2 se van a crear inicialmente. Más adelante se pueden añadir más sistemas de archivos.

Nombre del sistema de archivos

Cada sistema de archivos GFS2 debe tener un nombre único. Este nombre suele ser el mismo que el del volumen lógico LVM y se utiliza como nombre de la tabla de bloqueo DLM cuando se monta un sistema de archivos GFS2. Por ejemplo, esta guía utiliza los nombres de sistemas de archivos **mydata1** y **mydata2** en algunos procedimientos de ejemplo.

Revistas

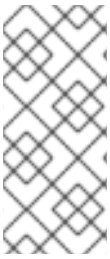
Determine el número de diarios para sus sistemas de archivos GFS2. GFS2 requiere un diario por cada nodo del clúster que necesite montar el sistema de archivos. Por ejemplo, si tiene un clúster de 16 nodos pero sólo necesita montar el sistema de archivos desde dos nodos, sólo necesitará dos diarios. GFS2 le permite añadir diarios dinámicamente en un momento posterior con la utilidad **gfs2_jadd** cuando los servidores adicionales montan un sistema de archivos.

Dispositivos de almacenamiento y particiones

Determine los dispositivos de almacenamiento y las particiones que se utilizarán para crear volúmenes lógicos (utilizando **lvmlockd**) en los sistemas de archivos.

Protocolo de tiempo

Asegúrese de que los relojes de los nodos GFS2 están sincronizados. Se recomienda utilizar el Protocolo de Tiempo de Precisión (PTP) o, si es necesario para su configuración, el software del Protocolo de Tiempo de Red (NTP) proporcionado con su distribución de Red Hat Enterprise Linux. Los relojes del sistema en los nodos GFS2 deben estar a pocos minutos de distancia entre sí para evitar la actualización innecesaria de la marca de tiempo del nodo. La actualización innecesaria de las marcas de tiempo de los nodos afecta gravemente al rendimiento del clúster.



NOTA

Puede ver problemas de rendimiento con GFS2 cuando se emiten muchas operaciones de creación y borrado desde más de un nodo en el mismo directorio al mismo tiempo. Si esto causa problemas de rendimiento en su sistema, debería localizar la creación y eliminación de archivos por parte de un nodo a directorios específicos de ese nodo en la medida de lo posible.

1.2. CONSIDERACIONES SOBRE LA COMPATIBILIDAD CON GFS2

Tabla 1.1, "Límites de soporte de GFS2" resume el tamaño máximo actual del sistema de archivos y el número de nodos que admite GFS2.

Tabla 1.1. Límites de soporte de GFS2

Parámetro	Máximo
Número de nodos	16 (x86, Power8 en PowerVM) 4 (s390x bajo z/VM)
Tamaño del sistema de archivos	100TB en todas las arquitecturas compatibles

GFS2 se basa en una arquitectura de 64 bits, que teóricamente puede acomodar un sistema de archivos de 8 EB. Si su sistema requiere sistemas de archivos GFS2 más grandes que los soportados actualmente, contacte con su representante de servicio de Red Hat.

NOTA

Aunque un sistema de archivos GFS2 puede ser implementado en un sistema independiente o como parte de una configuración de cluster, Red Hat no soporta el uso de GFS2 como un sistema de archivos de nodo único. Red Hat sí admite una serie de sistemas de archivos de nodo único de alto rendimiento que están optimizados para un nodo único y, por lo tanto, tienen generalmente una menor sobrecarga que un sistema de archivos de clúster. Red Hat recomienda el uso de estos sistemas de archivos en lugar de GFS2 en los casos en que sólo un nodo necesita montar el sistema de archivos. Para obtener información sobre los sistemas de archivos que soporta Red Hat Enterprise Linux 8, consulte Gestión de sistemas de archivos .

Red Hat seguirá dando soporte a los sistemas de archivos GFS2 de un solo nodo para montar instantáneas de los sistemas de archivos del clúster que puedan ser necesarias, por ejemplo, para realizar copias de seguridad.

Al determinar el tamaño de su sistema de archivos, debe tener en cuenta sus necesidades de recuperación. Ejecutar el comando **fsck.gfs2** en un sistema de archivos muy grande puede llevar mucho tiempo y consumir una gran cantidad de memoria. Además, en el caso de que se produzca un fallo en el disco o en el subsistema de disco, el tiempo de recuperación está limitado por la velocidad de los medios de copia de seguridad. Para obtener información sobre la cantidad de memoria que requiere el comando **fsck.gfs2**, consulte Determinación de la memoria necesaria para ejecutar fsck.gfs2 .

Mientras que un sistema de archivos GFS2 puede ser usado fuera de LVM, Red Hat sólo soporta sistemas de archivos GFS2 que son creados en un volumen lógico LVM compartido.

NOTA

Al configurar un sistema de archivos GFS2 como sistema de archivos de clúster, debe asegurarse de que todos los nodos del clúster tengan acceso al almacenamiento compartido. No se admiten configuraciones de clúster asimétricas en las que algunos nodos tienen acceso al almacenamiento compartido y otros no. Esto no requiere que todos los nodos monten el sistema de archivos GFS2.

1.3. CONSIDERACIONES SOBRE EL FORMATO DE GFS2

Esta sección ofrece recomendaciones sobre cómo formatear su sistema de archivos GFS2 para optimizar el rendimiento.

IMPORTANTE

Asegúrese de que su despliegue del complemento de alta disponibilidad de Red Hat satisface sus necesidades y puede ser soportado. Consulte con un representante autorizado de Red Hat para verificar su configuración antes de la implementación.

Tamaño del sistema de archivos: Cuanto más pequeño, mejor

GFS2 se basa en una arquitectura de 64 bits, que teóricamente puede albergar un sistema de archivos de 8 EB. Sin embargo, el tamaño máximo admitido actualmente de un sistema de archivos GFS2 para el hardware de 64 bits es de 100 TB.

Tenga en cuenta que aunque los sistemas de archivos de gran tamaño de GFS2 son posibles, eso no significa que sean recomendables. La regla general con GFS2 es que más pequeño es mejor: es mejor tener 10 sistemas de archivos de 1TB que un sistema de archivos de 10TB.

Hay varias razones por las que deberías mantener tus sistemas de archivos GFS2 pequeños:

- Se necesita menos tiempo para hacer una copia de seguridad de cada sistema de archivos.
- Se requiere menos tiempo si necesita comprobar el sistema de archivos con el comando **fsck.gfs2**.
- Se necesita menos memoria si se necesita comprobar el sistema de archivos con el comando **fsck.gfs2**.

Además, un menor número de grupos de recursos que mantener supone un mayor rendimiento.

Por supuesto, si haces tu sistema de archivos GFS2 demasiado pequeño, podrías quedarte sin espacio, y eso tiene sus propias consecuencias. Debes considerar tus propios casos de uso antes de decidir el tamaño.

Tamaño del bloque: Se prefieren los bloques por defecto (4K)

El comando **mkfs.gfs2** intenta estimar un tamaño de bloque óptimo basado en la topología del dispositivo. En general, los bloques de 4K son el tamaño de bloque preferido porque 4K es el tamaño de página (memoria) por defecto para Red Hat Enterprise Linux. A diferencia de otros sistemas de archivos, GFS2 realiza la mayoría de sus operaciones utilizando buffers de kernel de 4K. Si su tamaño de bloque es 4K, el kernel tiene que hacer menos trabajo para manipular los buffers.

Se recomienda utilizar el tamaño de bloque por defecto, que debería producir el mayor rendimiento. Puede que necesites utilizar un tamaño de bloque diferente sólo si necesitas un almacenamiento eficiente de muchos archivos muy pequeños.

Tamaño del diario: Por defecto (128MB) suele ser óptimo

Cuando se ejecuta el comando **mkfs.gfs2** para crear un sistema de archivos GFS2, se puede especificar el tamaño de los diarios. Si no se especifica un tamaño, será por defecto de 128MB, que debería ser óptimo para la mayoría de las aplicaciones.

Algunos administradores de sistemas podrían pensar que 128MB es excesivo y estarían tentados a reducir el tamaño del diario al mínimo de 8MB o a un más conservador 32MB. Aunque esto podría funcionar, puede afectar gravemente al rendimiento. Al igual que muchos sistemas de archivos con registro en el diario, cada vez que GFS2 escribe metadatos, éstos se consignan en el diario antes de colocarlos. Esto asegura que si el sistema se bloquea o pierde energía, recuperará todos los metadatos cuando el diario se reproduzca automáticamente en el momento del montaje. Sin embargo, no hace falta mucha actividad del sistema de archivos para llenar un diario de 8 MB, y cuando el diario está lleno, el rendimiento se ralentiza porque GFS2 tiene que esperar las escrituras en el almacenamiento.

Generalmente se recomienda utilizar el tamaño de diario por defecto de 128MB. Si su sistema de archivos es muy pequeño (por ejemplo, 5GB), tener un diario de 128MB puede ser poco práctico. Si tienes un sistema de archivos más grande y puedes permitirte el espacio, usar diarios de 256MB podría mejorar el rendimiento.

Tamaño y número de grupos de recursos

Cuando se crea un sistema de archivos GFS2 con el comando **mkfs.gfs2**, éste divide el almacenamiento en porciones uniformes conocidas como grupos de recursos. Intenta estimar un tamaño de grupo de recursos óptimo (que va de 32MB a 2GB). Puede anular el valor predeterminado con la opción **-r** del comando **mkfs.gfs2**.

El tamaño óptimo de su grupo de recursos depende de cómo vaya a utilizar el sistema de archivos. Ten en cuenta lo lleno que estará y si estará o no muy fragmentado.

Debería experimentar con diferentes tamaños de grupos de recursos para ver cuál resulta en un rendimiento óptimo. Es una buena práctica experimentar con un clúster de prueba antes de desplegar GFS2 en plena producción.

Si tu sistema de archivos tiene demasiados grupos de recursos, cada uno de los cuales es demasiado pequeño, las asignaciones de bloques pueden perder demasiado tiempo buscando decenas de miles de grupos de recursos para un bloque libre. Cuanto más lleno esté tu sistema de archivos, más grupos de recursos se buscarán, y cada uno de ellos requiere un bloqueo en todo el clúster. Esto conduce a un rendimiento lento.

Sin embargo, si su sistema de archivos tiene muy pocos grupos de recursos, cada uno de los cuales es demasiado grande, las asignaciones de bloques pueden competir más a menudo por el mismo bloqueo de grupo de recursos, lo que también afecta al rendimiento. Por ejemplo, si tiene un sistema de archivos de 10 GB que está dividido en cinco grupos de recursos de 2 GB, los nodos de su clúster se pelearán por esos cinco grupos de recursos con más frecuencia que si el mismo sistema de archivos estuviera dividido en 320 grupos de recursos de 32 MB. El problema se agrava si el sistema de archivos está casi lleno, ya que cada asignación de bloques podría tener que buscar en varios grupos de recursos antes de encontrar uno con un bloque libre. GFS2 intenta mitigar este problema de dos maneras:

- En primer lugar, cuando un grupo de recursos está completamente lleno, lo recuerda e intenta evitar comprobarlo para futuras asignaciones hasta que se libere un bloque del mismo. Si nunca borras archivos, la contención será menos severa. Sin embargo, si tu aplicación está constantemente borrando bloques y asignando nuevos bloques en un sistema de archivos que está mayormente lleno, la contención será muy alta y esto impactará severamente en el rendimiento.

- En segundo lugar, cuando se añaden nuevos bloques a un archivo existente (por ejemplo, al añadirlos) GFS2 intentará agrupar los nuevos bloques en el mismo grupo de recursos que el archivo. Esto se hace para aumentar el rendimiento: en un disco giratorio, las operaciones de búsqueda tardan menos tiempo cuando están físicamente juntas.

El peor escenario es cuando hay un directorio central en el que todos los nodos crean archivos, porque todos los nodos lucharán constantemente para bloquear el mismo grupo de recursos.

1.4. CONSIDERACIONES SOBRE LA AGRUPACIÓN

A la hora de determinar el número de nodos que contendrá su sistema, tenga en cuenta que existe un compromiso entre la alta disponibilidad y el rendimiento. Con un número mayor de nodos, se hace cada vez más difícil hacer escalar las cargas de trabajo. Por esta razón, Red Hat no soporta el uso de GFS2 para implementaciones de sistemas de archivos en cluster mayores a 16 nodos.

El despliegue de un sistema de archivos en cluster no es un reemplazo "drop in" para el despliegue de un solo nodo. Red Hat recomienda que permita un período de alrededor de 8-12 semanas de pruebas en las nuevas instalaciones con el fin de probar el sistema y asegurarse de que está trabajando en el nivel de rendimiento requerido. Durante este período, se puede resolver cualquier problema de rendimiento o funcionalidad y cualquier consulta debe dirigirse al equipo de soporte de Red Hat.

Red Hat recomienda a los clientes que estén pensando en implantar clusters que sus configuraciones sean revisadas por el soporte de Red Hat antes de la implantación para evitar posibles problemas de soporte más adelante.

1.5. CONSIDERACIONES SOBRE EL HARDWARE

Debe tener en cuenta las siguientes consideraciones de hardware al desplegar un sistema de archivos GFS2.

- Utilice opciones de almacenamiento de mayor calidad
  GFS2 puede funcionar en opciones de almacenamiento compartido más baratas, como iSCSI o Fibre Channel over Ethernet (FCoE), pero obtendrá un mejor rendimiento si adquiere un almacenamiento de mayor calidad y con mayor capacidad de caché. Red Hat realiza la mayoría de las pruebas de calidad, sanidad y rendimiento en el almacenamiento SAN con interconexión de canal de fibra. Como regla general, siempre es mejor desplegar algo que haya sido probado primero.

- Probar los equipos de la red antes de desplegarlos
  Un equipo de red más rápido y de mayor calidad hace que las comunicaciones del clúster y GFS2 funcionen más rápido y con mayor fiabilidad. Sin embargo, no es necesario adquirir el hardware más caro. Algunos de los conmutadores de red más caros tienen problemas para pasar los paquetes de multidifusión, que se utilizan para pasar los bloqueos (rebaños) de **fcntl**, mientras que los conmutadores de red básicos más baratos son a veces más rápidos y fiables. Red Hat recomienda probar los equipos antes de desplegarlos en plena producción.

CAPÍTULO 2. RECOMENDACIONES PARA EL USO DE GFS2

Esta sección ofrece recomendaciones generales sobre el uso de GFS2.

2.1. CONFIGURACIÓN DE LAS ACTUALIZACIONES DE ATIME

Cada inodo de archivo y de directorio tiene tres marcas de tiempo asociadas:

- **ctime**
- **mtime**
- **atime**

Si las actualizaciones de **atime** están habilitadas, como lo están por defecto en GFS2 y otros sistemas de archivos de Linux, entonces cada vez que se lee un archivo es necesario actualizar su inodo.

Dado que pocas aplicaciones utilizan la información proporcionada por **atime**, esas actualizaciones pueden requerir una cantidad significativa de tráfico de escritura y de bloqueo de archivos innecesario. Ese tráfico puede degradar el rendimiento; por lo tanto, puede ser preferible desactivar o reducir la frecuencia de las actualizaciones de **atime**.

Existen los siguientes métodos para reducir los efectos de la actualización de **atime**:

- Montar con **relatime** (atime relativo), que actualiza el **atime** si la actualización anterior **atime** es más antigua que la actualización **mtime** o **ctime**. Esta es la opción de montaje por defecto para los sistemas de archivos GFS2.
- Montar con **noatime** o **nodiratime**. El montaje con **noatime** desactiva las actualizaciones de **atime** tanto para los archivos como para los directorios de ese sistema de archivos, mientras que el montaje con **nodiratime** desactiva las actualizaciones de **atime** sólo para los directorios de ese sistema de archivos, Por lo general, se recomienda montar los sistemas de archivos GFS2 con la opción de montaje **noatime** o **nodiratime** siempre que sea posible, con preferencia por **noatime** cuando la aplicación lo permita. Para obtener más información sobre el efecto de estos argumentos en el rendimiento del sistema de archivos GFS2, consulte GFS2 Node Locking.

Utilice el siguiente comando para montar un sistema de archivos GFS2 con la opción de montaje **noatime** Linux.

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

BlockDevice

Especifica el dispositivo de bloque donde reside el sistema de archivos GFS2.

MountPoint

Especifica el directorio donde debe montarse el sistema de archivos GFS2.

En este ejemplo, el sistema de archivos GFS2 reside en **/dev/vg01/lvol0** y está montado en el directorio **/mygfs2** con las actualizaciones de **atime** desactivadas.

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

2.2. OPCIONES DE AJUSTE DEL VFS: INVESTIGACIÓN Y EXPERIMENTACIÓN

Como todos los sistemas de archivos de Linux, GFS2 se asienta sobre una capa llamada sistema de archivos virtual (VFS). El VFS proporciona buenos valores predeterminados para la configuración de la caché para la mayoría de las cargas de trabajo y no debería ser necesario cambiarlos en la mayoría de los casos. Sin embargo, si tienes una carga de trabajo que no está funcionando eficientemente (por ejemplo, la caché es demasiado grande o demasiado pequeña) entonces puedes ser capaz de mejorar el rendimiento utilizando el comando **sysctl**(8) para ajustar los valores de los archivos **sysctl** en el directorio **/proc/sys/vm**. La documentación de estos archivos se puede encontrar en el árbol de fuentes del kernel **Documentation/sysctl/vm.txt**.

Por ejemplo, los valores de **dirty_background_ratio** y **vfs_cache_pressure** pueden ajustarse en función de su situación. Para obtener los valores actuales, utilice los siguientes comandos:

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

Los siguientes comandos ajustan los valores:

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

Puedes cambiar permanentemente los valores de estos parámetros editando el archivo **/etc/sysctl.conf**.

Para encontrar los valores óptimos para sus casos de uso, investigue las distintas opciones de VFS y experimente en un clúster de prueba antes de desplegarlo en producción completa.

2.3. SELINUX EN GFS2

El uso de Security Enhanced Linux (SELinux) con GFS2 incurre en una pequeña penalización de rendimiento. Para evitar esta sobrecarga, puede optar por no utilizar SELinux con GFS2 incluso en un sistema con SELinux en modo de aplicación. Al montar un sistema de archivos GFS2, puede asegurarse de que SELinux no intente leer el elemento **seclabel** en cada objeto del sistema de archivos utilizando una de las opciones **context** como se describe en la página man **mount**(8); SELinux asumirá que todo el contenido del sistema de archivos está etiquetado con el elemento **seclabel** proporcionado en las opciones de montaje **context**. Esto también acelerará el procesamiento ya que evita otra lectura en disco del bloque de atributos extendidos que podría contener elementos de **seclabel**.

Por ejemplo, en un sistema con SELinux en modo de aplicación, puede utilizar el siguiente comando **mount** para montar el sistema de archivos GFS2 si el sistema de archivos va a contener contenido de Apache. Esta etiqueta se aplicará a todo el sistema de archivos; permanece en la memoria y no se escribe en el disco.

```
# mount -t gfs2 -o context=system_u:object_r:httpd_sys_content_t:s0
/dev/mapper/xyz/mnt/gfs2
```

Si no está seguro de si el sistema de archivos tendrá contenido Apache, puede utilizar las etiquetas **public_content_rw_t** o **public_content_t**, o puede definir una nueva etiqueta y definir una política en torno a ella.

Tenga en cuenta que en un clúster de Pacemaker siempre debe utilizar Pacemaker para gestionar un sistema de archivos GFS2. Puede especificar las opciones de montaje cuando cree un recurso de sistema de archivos GFS2.

2.4. CONFIGURACIÓN DE NFS SOBRE GFS2

Debido a la complejidad añadida del subsistema de bloqueo de GFS2 y a su naturaleza de clúster, la configuración de NFS sobre GFS2 requiere tomar muchas precauciones y una cuidadosa configuración. Esta sección describe las advertencias que debe tener en cuenta al configurar un servicio NFS sobre un sistema de archivos GFS2.

AVISO

Si el sistema de archivos GFS2 se exporta por NFS, debe montar el sistema de archivos con la opción **localflocks**. Dado que la utilización de la opción **localflocks** le impide acceder con seguridad al sistema de archivos GFS2 desde múltiples ubicaciones, y no es viable exportar GFS2 desde múltiples nodos simultáneamente, es un requisito de soporte que el sistema de archivos GFS2 se monte en un solo nodo a la vez cuando se utiliza esta configuración. El efecto previsto de esto es forzar que los bloqueos POSIX de cada servidor sean locales: no agrupados, independientes unos de otros. Esto se debe a que existen varios problemas si GFS2 intenta implementar bloqueos POSIX de NFS a través de los nodos de un clúster. Para las aplicaciones que se ejecutan en clientes NFS, los bloqueos POSIX localizados significan que dos clientes pueden mantener el mismo bloqueo simultáneamente si los dos clientes están montando desde diferentes servidores, lo que podría causar la corrupción de los datos. Si todos los clientes montan NFS desde un servidor, entonces el problema de que servidores separados concedan los mismos bloqueos de forma independiente desaparece. Si no está seguro de montar su sistema de archivos con la opción **localflocks**, no debería utilizar la opción. Póngase en contacto con el soporte de Red Hat inmediatamente para discutir la configuración apropiada para evitar la pérdida de datos. La exportación de GFS2 a través de NFS, aunque técnicamente se admite en algunas circunstancias, no se recomienda.

Para todas las demás aplicaciones GFS2 (no NFS), no monte su sistema de archivos utilizando **localflocks**, de modo que GFS2 gestione los bloqueos POSIX y los flocks entre todos los nodos del clúster (en todo el clúster). Si especifica **localflocks** y no utiliza NFS, los demás nodos del clúster no tendrán conocimiento de los bloqueos POSIX y flocks de los demás, por lo que no serán seguros en un entorno de clúster

Además de las consideraciones de bloqueo, debe tener en cuenta lo siguiente cuando configure un servicio NFS sobre un sistema de archivos GFS2.

- Red Hat sólo admite configuraciones de Red Hat High Availability Add-On que utilicen NFSv3 con bloqueo en una configuración activa/pasiva con las siguientes características. Esta configuración proporciona Alta Disponibilidad (HA) para el sistema de archivos y reduce el tiempo de inactividad del sistema, ya que un nodo fallado no da lugar a la necesidad de ejecutar el comando **fsck** al fallar el servidor NFS de un nodo a otro.
    - El sistema de archivos back-end es un sistema de archivos GFS2 que se ejecuta en un clúster de 2 a 16 nodos.
    - Un servidor NFSv3 se define como un servicio que exporta todo el sistema de archivos GFS2 desde un único nodo del clúster a la vez.

- El servidor NFS puede pasar de un nodo del clúster a otro (configuración activa/pasiva).
- No se permite el acceso al sistema de archivos GFS2 *except* a través del servidor NFS. Esto incluye tanto el acceso al sistema de archivos GFS2 local como el acceso a través de Samba o Clustered Samba. El acceso al sistema de archivos localmente a través del nodo de clúster desde el que se monta puede provocar la corrupción de los datos.
- No hay soporte de cuotas NFS en el sistema.

- La opción **fsid=** NFS es obligatoria para las exportaciones NFS de GFS2.
- Si surgen problemas con su clúster (por ejemplo, el clúster se queda sin capacidad y el fencing no tiene éxito), los volúmenes lógicos en clúster y el sistema de archivos GFS2 se congelarán y no será posible acceder a ellos hasta que el clúster se quede sin capacidad. Debe tener en cuenta esta posibilidad al determinar si una solución de conmutación por error simple como la definida en este procedimiento es la más adecuada para su sistema.

2.5. SERVICIO DE ARCHIVOS SAMBA (SMB O WINDOWS) SOBRE GFS2

Puede utilizar el servicio de archivos Samba (SMB o Windows) desde un sistema de archivos GFS2 con CTDB, que permite configuraciones activas/activas.

No se admite el acceso simultáneo a los datos del recurso compartido de Samba desde fuera de Samba. Actualmente no hay soporte para los arrendamientos de clústeres GFS2, lo que ralentiza el servicio de archivos de Samba. Para más información sobre las políticas de soporte para Samba, vea Políticas de soporte para RHEL Resilient Storage - Políticas generales de ctdb y Políticas de soporte para RHEL Resilient Storage - Exportación de contenidos gfs2 a través de otros protocolos en el Portal del cliente de Red Hat.

2.6. CONFIGURACIÓN DE MÁQUINAS VIRTUALES PARA GFS2

Cuando se utiliza un sistema de archivos GFS2 con una máquina virtual, es importante que los ajustes de almacenamiento de la máquina virtual en cada nodo estén configurados correctamente para forzar la desactivación de la caché. Por ejemplo, incluir estas configuraciones para **cache** y **io** en el dominio **libvirt** debería permitir que GFS2 se comporte como se espera.

```
<driver name='qemu' type='raw' cache='none' io='native'/>
```

Alternativamente, puedes configurar el atributo **shareable** dentro del elemento dispositivo. Esto indica que se espera que el dispositivo sea compartido entre dominios (siempre que el hipervisor y el SO lo soporten). Si se utiliza **shareable**, se debe utilizar **cache='no'** para ese dispositivo.

2.7. ASIGNACIÓN DE BLOQUES

Esta sección ofrece un resumen de los problemas relacionados con la asignación de bloques en los sistemas de archivos GFS2. Aunque las aplicaciones que sólo escriben datos no suelen preocuparse por cómo o dónde se asigna un bloque, un cierto conocimiento de cómo funciona la asignación de bloques puede ayudarle a optimizar el rendimiento.

2.7.1. Dejar espacio libre en el sistema de archivos

Cuando un sistema de archivos GFS2 está casi lleno, el asignador de bloques empieza a tener dificultades para encontrar espacio para asignar nuevos bloques. Como resultado, los bloques asignados por el asignador tienden a ser exprimidos al final de un grupo de recursos o en pequeñas porciones

donde la fragmentación del archivo es mucho más probable. Esta fragmentación de archivos puede causar problemas de rendimiento. Además, cuando un sistema de archivos GFS2 está casi lleno, el asignador de bloques GFS2 pasa más tiempo buscando entre varios grupos de recursos, y eso añade una contención de bloqueos que no necesariamente existiría en un sistema de archivos que tiene un amplio espacio libre. Esto también puede causar problemas de rendimiento.

Por estas razones, se recomienda no ejecutar un sistema de archivos que esté lleno en más de un 85%, aunque esta cifra puede variar dependiendo de la carga de trabajo.

2.7.2. Que cada nodo asigne sus propios archivos, si es posible

Al desarrollar aplicaciones para su uso con sistemas de archivos GFS2, se recomienda que cada nodo asigne sus propios archivos, si es posible. Debido a la forma en que funciona el gestor de bloqueos distribuidos (DLM), habrá más contención de bloqueos si todos los archivos son asignados por un nodo y otros nodos necesitan añadir bloques a esos archivos.

Con DLM, el primer nodo que bloquea un recurso (como un archivo) se convierte en el "maestro de bloqueo" para ese bloqueo. Otros nodos pueden bloquear ese recurso, pero tienen que pedir permiso al maestro de bloqueo primero. Cada nodo sabe para qué bloqueos es el maestro, y cada nodo sabe a qué nodo ha prestado un bloqueo. Bloquear un candado en el nodo maestro es mucho más rápido que bloquear uno en otro nodo que tiene que parar y pedir permiso al maestro del candado.

Como en muchos sistemas de archivos, el asignador de GFS2 intenta mantener los bloques de un mismo archivo cerca unos de otros para reducir el movimiento de las cabezas de disco y aumentar el rendimiento. Un nodo que asigna bloques a un archivo probablemente necesitará utilizar y bloquear los mismos grupos de recursos para los nuevos bloques (a menos que todos los bloques de ese grupo de recursos estén en uso). El sistema de archivos funcionará más rápido si el maestro de bloqueo del grupo de recursos que contiene el archivo asigna sus bloques de datos (es más rápido que el nodo que abrió primero el archivo haga toda la escritura de los nuevos bloques).

2.7.3. Preasignar, si es posible

Si se preasignan los archivos, se pueden evitar las asignaciones de bloques y el sistema de archivos puede funcionar de forma más eficiente. GFS2 incluye la llamada al sistema **fallocate(1)**, que puede utilizarse para preasignar bloques de datos.

CAPÍTULO 3. SISTEMAS DE ARCHIVOS GFS2

Esta sección proporciona información sobre los comandos y las opciones que se utilizan para crear, montar y ampliar los sistemas de archivos GFS2.

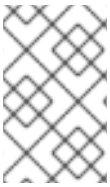
3.1. CREACIÓN DEL SISTEMA DE ARCHIVOS GFS2

Se crea un sistema de archivos GFS2 con el comando **mkfs.gfs2**. Se crea un sistema de archivos en un volumen LVM activado.

3.1.1. El comando mkfs de GFS2

La siguiente información es necesaria para ejecutar el comando **mkfs.gfs2** para crear un sistema de archivos GFS2 en clúster:

- Nombre del protocolo/módulo de bloqueo, que es **lock_dlm** para un clúster
- Nombre del clúster
- Número de diarios (se requiere un diario por cada nodo que pueda montar el sistema de archivos)



NOTA

Una vez que haya creado un sistema de archivos GFS2 con el comando **mkfs.gfs2**, no podrá disminuir el tamaño del sistema de archivos. Sin embargo, puede aumentar el tamaño de un sistema de archivos existente con el comando **gfs2_grow**.

El formato para crear un sistema de archivos GFS2 en cluster es el siguiente. Tenga en cuenta que Red Hat no admite el uso de GFS2 como sistema de archivos de un solo nodo.

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

Si lo prefiere, puede crear un sistema de archivos GFS2 utilizando el comando **mkfs** con el parámetro **-t** especificando un sistema de archivos de tipo **gfs2**, seguido de las opciones del sistema de archivos GFS2.

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

AVISO

La especificación incorrecta del parámetro *ClusterName:FSName* puede causar la corrupción del sistema de archivos o del espacio de bloqueo.

ClusterName

El nombre del cluster para el que se está creando el sistema de archivos GFS2.

FSName

El nombre del sistema de archivos, que puede tener de 1 a 16 caracteres. El nombre debe ser único para todos los sistemas de archivos de **lock_dlm** en el clúster.

NumberJournals

Especifica el número de diarios que debe crear el comando **mkfs.gfs2**. Se requiere un diario por cada nodo que monte el sistema de archivos. En el caso de los sistemas de archivos GFS2, se pueden añadir más diarios posteriormente sin que crezca el sistema de archivos.

BlockDevice

Especifica un dispositivo lógico o de otro tipo de bloque

Tabla 3.1, "Opciones de comando **mkfs.gfs2**" describe las opciones del comando **mkfs.gfs2** (banderas y parámetros).

Tabla 3.1. Opciones de comando mkfs.gfs2

Bandera	Parámetro	Descripción
-c	**Megabytes**	Establece el tamaño inicial del archivo de cambio de cuota de cada diario en **Megabytes**.
-D		Activa la salida de depuración.
-h		Ayuda. Muestra las opciones disponibles.
-J	**Megabytes**	Especifica el tamaño del diario en megabytes. El tamaño del diario por defecto es de 128 megabytes. El tamaño mínimo es de 8 megabytes. Los diarios más grandes mejoran el rendimiento, aunque utilizan más memoria que los diarios más pequeños.
-j	**Number**	Especifica el número de diarios que debe crear el comando **mkfs.gfs2**. Se requiere un diario por cada nodo que monte el sistema de archivos. Si no se especifica esta opción, se creará un diario. En el caso de los sistemas de archivos GFS2, se pueden añadir diarios adicionales más adelante sin que el sistema de archivos crezca.
-O		Evita que el comando **mkfs.gfs2** pida confirmación antes de escribir el sistema de archivos.

Bandera	Parámetro	Descripción
-p	**LockProtoName**	* Especifica el nombre del protocolo de bloqueo a utilizar. Los protocolos de bloqueo reconocidos son: * **lock_dlm** * **lock_nolock**
-q		Silencio. No mostrar nada.
-r	**Megabytes**	Especifica el tamaño de los grupos de recursos en megabytes. El tamaño mínimo de los grupos de recursos es de 32 megabytes. El tamaño máximo de los grupos de recursos es de 2048 megabytes. Un tamaño de grupo de recursos grande puede aumentar el rendimiento en sistemas de archivos muy grandes. Si no se especifica, **mkfs.gfs2** elige el tamaño del grupo de recursos basándose en el tamaño del sistema de archivos: los sistemas de archivos de tamaño medio tendrán grupos de recursos de 256 megabytes, y los sistemas de archivos más grandes tendrán RGs más grandes para mejorar el rendimiento.

Bandera	Parámetro	Descripción
-t	**LockTableName**	* Un identificador único que especifica el campo de la tabla de bloqueo cuando se utiliza el protocolo **lock_dlm**; el protocolo **lock_nolock** no utiliza este parámetro. * Este parámetro tiene dos partes separadas por dos puntos (sin espacios) como sigue **ClusterName:FSName** . * **ClusterName** es el nombre del clúster para el que se está creando el sistema de archivos GFS2; sólo los miembros de este clúster pueden utilizar este sistema de archivos. * **FSName** , el nombre del sistema de archivos, puede tener entre 1 y 16 caracteres, y el nombre debe ser único entre todos los sistemas de archivos del clúster.
-V		Muestra la información de la versión del comando.

3.1.2. Creación de un sistema de archivos GFS2

El siguiente ejemplo crea dos sistemas de archivos GFS2. Para ambos sistemas de archivos, lock_dlm` es el protocolo de bloqueo que utiliza el sistema de archivos, ya que se trata de un sistema de archivos en clúster. Ambos sistemas de archivos pueden utilizarse en el clúster denominado **alpha**.

Para el primer sistema de archivos, el nombre del sistema de archivos es **mydata1**. contiene ocho diarios y se crea en **/dev/vg01/lvol0**. Para el segundo sistema de archivos, el nombre del sistema de archivos es **mydata2**. Contiene ocho diarios y se crea en **/dev/vg01/lvol1**.

```
# mkfs.gfs2 -p lock_dlm -t alpha:mydata1 -j 8 /dev/vg01/lvol0
# mkfs.gfs2 -p lock_dlm -t alpha:mydata2 -j 8 /dev/vg01/lvol1
```

3.2. MONTAJE DE UN SISTEMA DE ARCHIVOS GFS2

NOTA

Siempre debe utilizar Pacemaker para gestionar el sistema de archivos GFS2 en un entorno de producción en lugar de montar manualmente el sistema de archivos con un comando **mount**, ya que esto puede causar problemas al apagar el sistema, como se describe en Desmontaje de un sistema de archivos GFS2 .

Antes de poder montar un sistema de archivos GFS2, el sistema de archivos debe existir, el volumen en el que existe el sistema de archivos debe estar activado y los sistemas de agrupación y bloqueo que lo soportan deben estar iniciados. Una vez cumplidos estos requisitos, puede montar el sistema de archivos GFS2 como lo haría con cualquier sistema de archivos de Linux.

Para el correcto funcionamiento del sistema de archivos GFS2, el paquete **gfs2-utils** debe estar instalado en todos los nodos que monten un sistema de archivos GFS2. El paquete **gfs2-utils** forma parte del canal Resilient Storage.

Para manipular las ACL de los archivos, debe montar el sistema de archivos con la opción de montaje **-o acl**. Si un sistema de archivos se monta sin la opción de montaje **-o acl**, los usuarios pueden ver las ACL (con **getfacl**), pero no pueden establecerlas (con **setfacl**).

3.2.1. Montaje de un sistema de archivos GFS2 sin especificar opciones

En este ejemplo, el sistema de archivos GFS2 en **/dev/vg01/lvol0** se monta en el directorio **/mygfs2**.

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

3.2.2. Montaje de un sistema de archivos GFS2 que especifica las opciones de montaje

El siguiente es el formato del comando para montar un sistema de archivos GFS2 que especifica las opciones de montaje.

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

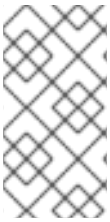
BlockDevice

Especifica el dispositivo de bloque donde reside el sistema de archivos GFS2.

MountPoint

Especifica el directorio donde debe montarse el sistema de archivos GFS2.

El argumento **-o option** consiste en opciones específicas de GFS2 (véase Tabla 3.2, "Opciones de montaje específicas de GFS2") o en opciones estándar aceptables de Linux **mount -o**, o una combinación de ambas. Los parámetros múltiples de **option** se separan con una coma y sin espacios.



NOTA

El comando **mount** es un comando del sistema Linux. Además de utilizar las opciones específicas de GFS2 descritas en esta sección, puede utilizar otras opciones estándar del comando **mount** (por ejemplo, **-r**). Para obtener información sobre otras opciones del comando **mount** de Linux, consulte la página del manual de Linux **mount**.

Tabla 3.2, "Opciones de montaje específicas de GFS2" describe los valores disponibles específicos de GFS2 **-o option** que se pueden pasar a GFS2 en el momento del montaje.

NOTA

Esta tabla incluye descripciones de opciones que se utilizan sólo con sistemas de archivos locales. Tenga en cuenta, sin embargo, que Red Hat no soporta el uso de GFS2 como un sistema de archivos de nodo único. Red Hat continuará apoyando los sistemas de archivos GFS2 de nodo único para montar instantáneas de sistemas de archivos de cluster (por ejemplo, para propósitos de respaldo).

Tabla 3.2. Opciones de montaje específicas de GFS2

Opción	Descripción
acl	Permite manipular las ACL de los archivos. Si un sistema de archivos se monta sin la opción de montaje **acl**, los usuarios pueden ver las ACL (con **getfacl**), pero no pueden establecerlas (con **setfacl**).
data=[ordered|writeback]	Cuando se establece **data=ordered**, los datos del usuario modificados por una transacción se vuelcan al disco antes de que la transacción se confirme en el disco. Esto debería evitar que el usuario vea bloques no inicializados en un archivo después de una caída. Cuando se establece el modo **data=writeback**, los datos del usuario se escriben en el disco en cualquier momento después de ser ensuciados; esto no proporciona la misma garantía de consistencia que el modo **ordered**, pero debería ser ligeramente más rápido para algunas cargas de trabajo. El valor por defecto es el modo **ordered**.
* **ignore_local_fs** * **Caution:** Esta opción debería *not* utilizarse cuando se comparten sistemas de archivos GFS2.	Obliga a GFS2 a tratar el sistema de archivos como un sistema de archivos multi-host. Por defecto, el uso de **lock_nolock** activa automáticamente la bandera **localflocks**.
* **localflocks** * **Caution:** Esta opción no debe utilizarse cuando se comparten sistemas de archivos GFS2.	Indica a GFS2 que deje que la capa VFS (sistema de archivos virtual) se encargue de todo el flock y el fcntl. La bandera **localflocks** es activada automáticamente por **lock_nolock**.
lockproto=LockModuleName	Permite al usuario especificar qué protocolo de bloqueo utilizar con el sistema de archivos. Si no se especifica **LockModuleName**, el nombre del protocolo de bloqueo se lee del superbloque del sistema de archivos.
locktable=LockTableName	Permite al usuario especificar qué tabla de bloqueo utilizar con el sistema de archivos.

Opción	Descripción
quota=[off/account/on]	Activa o desactiva las cuotas de un sistema de archivos. Configurar las cuotas en el estado **account** hace que las estadísticas de uso por UID/GID sean mantenidas correctamente por el sistema de archivos; los valores de límite y advertencia son ignorados. El valor por defecto es **off**.
errors=panic|withdraw	Cuando se especifica **errors=panic**, los errores del sistema de archivos causarán un pánico en el kernel. Cuando se especifica **errors=withdraw**, que es el comportamiento por defecto, los errores del sistema de archivos harán que el sistema se retire del sistema de archivos y lo haga inaccesible hasta el siguiente reinicio; en algunos casos el sistema puede seguir funcionando.
discard/nodiscard	Hace que GFS2 genere solicitudes de E/S de "descarte" para los bloques que han sido liberados. Estos pueden ser utilizados por el hardware adecuado para implementar el aprovisionamiento ligero y esquemas similares.
barrier/nobarrier	Hace que GFS2 envíe barreras de E/S cuando se vacía el diario. El valor por defecto es **on**. Esta opción se convierte automáticamente en **off** si el dispositivo subyacente no admite barreras de E/S. Se recomienda encarecidamente el uso de barreras de E/S con GFS2 en todo momento, a menos que el dispositivo de bloque esté diseñado de forma que no pueda perder el contenido de su caché de escritura (por ejemplo, si está en un SAI o no tiene una caché de escritura).
quota_quantum=*secs*	Establece el número de segundos durante los cuales un cambio en la información de cuota puede permanecer en un nodo antes de ser escrito en el archivo de cuota. Esta es la forma preferida de establecer este parámetro. El valor es un número entero de segundos mayor que cero. El valor por defecto es de 60 segundos. Las configuraciones más cortas resultan en actualizaciones más rápidas de la información de la cuota perezosa y menos probabilidad de que alguien exceda su cuota. Las configuraciones más largas hacen que las operaciones del sistema de archivos que involucran cuotas sean más rápidas y eficientes.

Opción	Descripción
statfs_quantum=*secs*	Establecer **statfs_quantum** a 0 es la forma preferida de establecer la versión lenta de **statfs**. El valor por defecto es de 30 segundos, que establece el período máximo de tiempo antes de que los cambios de **statfs** se sincronicen con el archivo maestro **statfs**. Esto puede ajustarse para permitir valores más rápidos y menos precisos de **statfs** o valores más lentos y precisos. Cuando esta opción se ajusta a 0, **statfs** siempre informará de los valores reales.
statfs_percent=*value*	Proporciona un límite en el porcentaje máximo de cambio en la información de **statfs** a nivel local antes de que se sincronice con el archivo maestro **statfs**, incluso si el período de tiempo no ha expirado. Si el ajuste de **statfs_quantum** es 0, este ajuste se ignora.

3.2.3. Desmontaje de un sistema de archivos GFS2

Los sistemas de archivos GFS2 que se han montado manualmente en lugar de automáticamente a través de Pacemaker no serán conocidos por el sistema cuando los sistemas de archivos se desmonten al apagar el sistema. En consecuencia, el agente de recursos GFS2 no desmontará el sistema de archivos GFS2. Tras el cierre del agente de recursos GFS2, el proceso de apagado estándar elimina todos los procesos de usuario restantes, incluida la infraestructura del clúster, e intenta desmontar el sistema de archivos. Este desmontaje fallará sin la infraestructura de clúster y el sistema se colgará.

Para evitar que el sistema se cuelgue cuando se desmonten los sistemas de archivos GFS2, debe hacer una de las siguientes cosas:

- Utilice siempre Pacemaker para gestionar el sistema de archivos GFS2.
- Si un sistema de archivos GFS2 se ha montado manualmente con el comando **mount**, asegúrese de desmontar el sistema de archivos manualmente con el comando **umount** antes de reiniciar o apagar el sistema.

Si su sistema de archivos se cuelga mientras se desmonta durante el apagado del sistema en estas circunstancias, realice un reinicio del hardware. Es poco probable que se pierdan datos, ya que el sistema de archivos se sincroniza antes en el proceso de apagado.

El sistema de archivos GFS2 puede desmontarse de la misma manera que cualquier sistema de archivos de Linux, utilizando el comando **umount**.

NOTA

El comando **umount** es un comando del sistema Linux. Puede encontrar información sobre este comando en las páginas del manual del comando de Linux **umount**.

Uso

```
umount MountPoint
```

MountPoint

Especifica el directorio en el que está montado el sistema de archivos GFS2.

3.3. COPIA DE SEGURIDAD DE UN SISTEMA DE ARCHIVOS GFS2

Es importante hacer copias de seguridad regulares de su sistema de archivos GFS2 en caso de emergencia, independientemente del tamaño de su sistema de archivos. Muchos administradores de sistemas se sienten seguros porque están protegidos por RAID, multipath, mirroring, snapshots y otras formas de redundancia, pero no hay nada suficientemente seguro.

Puede ser un problema crear una copia de seguridad ya que el proceso de copia de seguridad de un nodo o conjunto de nodos suele implicar la lectura de todo el sistema de archivos en secuencia. Si esto se hace desde un solo nodo, ese nodo retendrá toda la información en la caché hasta que otros nodos del cluster comiencen a solicitar bloqueos. Ejecutar este tipo de programa de copia de seguridad mientras el clúster está en funcionamiento tendrá un impacto negativo en el rendimiento.

La eliminación de las cachés una vez que se ha completado la copia de seguridad reduce el tiempo necesario para que los otros nodos recuperen la propiedad de sus bloqueos/cachés del clúster. Sin embargo, esto no es lo ideal, ya que los otros nodos habrán dejado de almacenar en caché los datos que estaban almacenando antes de que comenzara el proceso de copia de seguridad. Puede eliminar las cachés utilizando el siguiente comando una vez que se haya completado la copia de seguridad:

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

Es más rápido si cada nodo del clúster hace una copia de seguridad de sus propios archivos, de modo que la tarea se divide entre los nodos. Esto puede lograrse con un script que utilice el comando **rsync** en directorios específicos de cada nodo.

Red Hat recomienda hacer una copia de seguridad de GFS2 creando una instantánea de hardware en la SAN, presentando la instantánea a otro sistema y haciendo una copia de seguridad allí. El sistema de respaldo debe montar la instantánea con **-o lockproto=lock_nolock** ya que no estará en un cluster.

3.4. SUSPENDER LA ACTIVIDAD EN UN SISTEMA DE ARCHIVOS GFS2

Puede suspender la actividad de escritura en un sistema de archivos utilizando el comando **dmsetup suspend**. La suspensión de la actividad de escritura permite utilizar instantáneas de dispositivos basados en hardware para capturar el sistema de archivos en un estado consistente. El comando **dmsetup resume** finaliza la suspensión.

El formato del comando para suspender la actividad en un sistema de archivos GFS2 es el siguiente.

```
dmsetup suspender MountPoint
```

Este ejemplo suspende las escrituras en el sistema de archivos /**mygfs2**.

```
# dmsetup suspend /mygfs2
```

El formato del comando para finalizar la suspensión de la actividad en un sistema de archivos GFS2 es el siguiente.

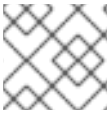
```
dmsetup reanudar MountPoint
```

Este ejemplo pone fin a la suspensión de las escrituras en el sistema de archivos /**mygfs2**.

```
# dmsetup resume /mygfs2
```

3.5. CRECIMIENTO DE UN SISTEMA DE ARCHIVOS GFS2

El comando **gfs2_grow** se utiliza para ampliar un sistema de archivos GFS2 después de que se haya ampliado el dispositivo donde reside el sistema de archivos. La ejecución del comando **gfs2_grow** en un sistema de archivos GFS2 existente llena todo el espacio libre entre el extremo actual del sistema de archivos y el final del dispositivo con una extensión de sistema de archivos GFS2 recién inicializada. Todos los nodos del clúster pueden entonces utilizar el espacio de almacenamiento extra que se ha añadido.



NOTA

No se puede disminuir el tamaño de un sistema de archivos GFS2.

El comando **gfs2_grow** debe ejecutarse en un sistema de archivos montado. El siguiente procedimiento aumenta el tamaño del sistema de archivos GFS2 en un clúster que está montado en el volumen lógico **shared_vg/shared_lv1** con un punto de montaje de **/mnt/gfs2**.

1. Realice una copia de seguridad de los datos del sistema de archivos.

2. Si no conoce el volumen lógico que utiliza el sistema de archivos que se va a ampliar, puede determinarlo ejecutando el comando **df *mountpoint*** comando. Esto mostrará el nombre del dispositivo en el siguiente formato:
/dev/mapper/*vg-lv*

   Por ejemplo, el nombre del dispositivo **/dev/mapper/shared_vg-shared_lv1** indica que el volumen lógico es **shared_vg/shared_lv1**.

3. En un nodo del clúster, amplíe el volumen subyacente del clúster con el comando **lvextend**, utilizando la opción **--lockopt skiplv** para anular el bloqueo normal del volumen lógico.

```
# lvextend --lockopt skiplv -L+1G shared_vg/shared_lv1
WARNING: skipping LV lock in lvmlockd.
Size of logical volume shared_vg/shared_lv1 changed from 5.00 GiB (1280 extents) to 6.00 GiB (1536 extents).
WARNING: extending LV with a shared lock, other hosts may require LV refresh.
Logical volume shared_vg/shared_lv1 successfully resized.
```

4. Si está ejecutando RHEL 8.0, en cada nodo adicional del clúster actualice el volumen lógico para actualizar el volumen lógico activo en ese nodo. Este paso no es necesario en los sistemas que ejecutan RHEL 8.1 y posteriores, ya que el paso se automatiza cuando se amplía el volumen lógico.

```
# lvchange --refresh shared_vg/shared_lv1
```

5. En un nodo del clúster, aumente el tamaño del sistema de archivos GFS2. No amplíe el sistema de archivos si el volumen lógico no se ha actualizado en todos los nodos, ya que de lo contrario los datos del sistema de archivos podrían no estar disponibles en todo el clúster.

```
# gfs2_grow /mnt/gfs2
FS: Mount point:             /mnt/gfs2
FS: Device:                  /dev/mapper/shared_vg-shared_lv1
```

```
FS: Size:                 1310719 (0x13ffff)
DEV: Length:              1572864 (0x180000)
The file system will grow by 1024MB.
gfs2_grow complete.
```

6. Ejecute el comando **df** en todos los nodos para comprobar que el nuevo espacio está ahora disponible en el sistema de archivos. Tenga en cuenta que el comando **df** puede tardar hasta 30 segundos en mostrar el mismo tamaño del sistema de archivos en todos los nodos

```
# df -h /mnt/gfs2
Filesystem                      Size  Used Avail Use% Mounted on
/dev/mapper/shared_vg-shared_lv1  6.0G  4.5G  1.6G  75% /mnt/gfs2
```

3.6. AÑADIR DIARIOS A UN SISTEMA DE ARCHIVOS GFS2

GFS2 requiere un diario por cada nodo de un cluster que necesite montar el sistema de archivos. Si añade nodos adicionales al clúster, puede añadir diarios a un sistema de archivos GFS2 con el comando **gfs2_jadd**. Puede añadir diarios a un sistema de archivos GFS2 de forma dinámica en cualquier momento sin necesidad de ampliar el volumen lógico subyacente. El comando **gfs2_jadd** debe ejecutarse en un sistema de archivos montado, pero sólo debe ejecutarse en un nodo del clúster. Todos los demás nodos perciben que se ha producido la expansión.

NOTA

Si un sistema de archivos GFS2 está lleno, el comando **gfs2_jadd** fallará, incluso si el volumen lógico que contiene el sistema de archivos ha sido ampliado y es más grande que el sistema de archivos. Esto se debe a que en un sistema de archivos GFS2, los diarios son archivos simples en lugar de metadatos incrustados, por lo que la simple ampliación del volumen lógico subyacente no proporcionará espacio para los diarios.

Antes de añadir diarios a un sistema de archivos GFS2, puede averiguar cuántos diarios contiene actualmente el sistema de archivos GFS2 con el comando **gfs2_edit -p jindex**, como en el siguiente ejemplo:

```
# gfs2_edit -p jindex /dev/sasdrives/scratch|grep journal
  3/3 [fc7745eb] 4/25 (0x4/0x19): File    journal0
  4/4 [8b70757d] 5/32859 (0x5/0x805b): File    journal1
  5/5 [127924c7] 6/65701 (0x6/0x100a5): File    journal2
```

El formato del comando básico para añadir diarios a un sistema de archivos GFS2 es el siguiente.

```
gfs2_jadd -j Number MountPoint
```

Number

Especifica el número de nuevos diarios que se van a añadir.

MountPoint

Especifica el directorio donde se monta el sistema de archivos GFS2.

En este ejemplo, se añade un diario al sistema de archivos en el directorio /**mygfs2**.

```
gfs2_jadd -j 1 /mygfs2
```

CAPÍTULO 4. GESTIÓN DE CUOTAS DE GFS2

Las cuotas del sistema de archivos se utilizan para limitar la cantidad de espacio del sistema de archivos que un usuario o grupo puede utilizar. Un usuario o grupo no tiene un límite de cuota hasta que se establece uno. Cuando un sistema de archivos GFS2 se monta con la opción **quota=on** o **quota=account**, GFS2 lleva un registro del espacio utilizado por cada usuario y grupo incluso cuando no hay límites establecidos. GFS2 actualiza la información de las cuotas de forma transaccional, por lo que las caídas del sistema no requieren que se reconstruyan los usos de las cuotas.

Para evitar una ralentización del rendimiento, un nodo GFS2 sincroniza las actualizaciones del archivo de cuotas sólo periódicamente. La contabilidad de cuotas difusa puede permitir que los usuarios o grupos superen ligeramente el límite establecido. Para minimizar esto, GFS2 reduce dinámicamente el período de sincronización a medida que se acerca al límite de cuota dura.

NOTA

GFS2 es compatible con las instalaciones de cuotas estándar de Linux. Para poder utilizarlo, tendrá que instalar el **quota** RPM. Esta es la forma preferida de administrar las cuotas en GFS2 y debería utilizarse para todas las nuevas implementaciones de GFS2 que utilicen cuotas. Esta sección documenta la administración de cuotas de GFS2 utilizando estas facilidades.

Para más información sobre las cuotas de disco, consulte las páginas **man** de los siguientes comandos:

- **quotacheck**
- **edquota**
- **repquota**
- **quota**

4.1. CONFIGURACIÓN DE CUOTAS DE DISCO GFS2

Para implementar las cuotas de disco, siga los siguientes pasos:

1. Configurar las cuotas en modo de aplicación o de contabilidad.
2. Inicializar el archivo de la base de datos de cuotas con la información de uso de bloques actual.
3. Asignar políticas de cuotas. (En el modo contable, estas políticas no se aplican)

Cada uno de estos pasos se discute en detalle en las siguientes secciones.

4.1.1. Configuración de las cuotas en modo de aplicación o en modo contable

En los sistemas de archivos GFS2, las cuotas están desactivadas por defecto. Para habilitar las cuotas para un sistema de archivos, monte el sistema de archivos con la opción **quota=on** especificada.

Para montar un sistema de archivos con cuotas habilitadas, especifique **quota=on** para el argumento **options** cuando cree el recurso del sistema de archivos GFS2 en un clúster. Por ejemplo, el siguiente comando especifica que el recurso GFS2 **Filesystem** que se está creando se montará con cuotas habilitadas.

```
# pcs resource create gfs2mount Filesystem options="quota=on" device=BLOCKDEVICE
directory=MOUNTPOINT fstype=gfs2 clone
```

Es posible hacer un seguimiento del uso del disco y mantener la contabilidad de cuotas para cada usuario y grupo sin aplicar los valores de límite y advertencia. Para ello, monte el sistema de archivos con la opción **quota=account** especificada.

Para montar un sistema de archivos con cuotas desactivadas, especifique **quota=off** para el argumento **options** cuando cree el recurso del sistema de archivos GFS2 en un clúster.

4.1.2. Creación de los archivos de la base de datos de cuotas

Después de montar cada sistema de archivos habilitado para cuotas, el sistema es capaz de trabajar con cuotas de disco. Sin embargo, el sistema de archivos en sí mismo aún no está preparado para soportar cuotas. El siguiente paso es ejecutar el comando **quotacheck**.

El comando **quotacheck** examina los sistemas de archivos con cuotas habilitadas y construye una tabla del uso actual del disco por sistema de archivos. La tabla se utiliza entonces para actualizar la copia del uso del disco del sistema operativo. Además, se actualizan los archivos de cuota de disco del sistema de archivos.

Para crear los archivos de cuotas en el sistema de archivos, utilice las opciones **-u** y **-g** del comando **quotacheck**; ambas opciones deben especificarse para que se inicialicen las cuotas de usuarios y grupos. Por ejemplo, si las cuotas están activadas para el sistema de archivos **/home**, cree los archivos en el directorio **/home**:

```
quotacheck -ug /home
```

4.1.3. Asignación de cuotas por usuario

El último paso es asignar las cuotas de disco con el comando **edquota**. Tenga en cuenta que si ha montado su sistema de archivos en modo contable (con la opción **quota=account** especificada), las cuotas no se aplican.

Para configurar la cuota de un usuario, como root en un prompt del shell, ejecute el comando

```
# edquota username
```

Realice este paso para cada usuario que necesite una cuota. Por ejemplo, si se habilita una cuota para la partición **/home** (**/dev/VolGroup00/LogVol02** en el ejemplo siguiente) y se ejecuta el comando **edquota testuser**, se muestra lo siguiente en el editor configurado por defecto para el sistema:

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

NOTA

El editor de texto definido por la variable de entorno **EDITOR** es utilizado por **edquota**. Para cambiar el editor, establezca la variable de entorno **EDITOR** en su archivo ~/**.bash_profile** con la ruta completa del editor de su elección.

La primera columna es el nombre del sistema de archivos que tiene una cuota habilitada para él. La

segunda columna muestra cuántos bloques está utilizando actualmente el usuario. Las dos columnas siguientes se utilizan para establecer límites de bloques blandos y duros para el usuario en el sistema de archivos.

El límite de bloques blandos define la cantidad máxima de espacio en disco que se puede utilizar.

El límite de bloques duros es la cantidad máxima absoluta de espacio en disco que un usuario o grupo puede utilizar. Una vez que se alcanza este límite, no se puede utilizar más espacio en disco.

El sistema de archivos GFS2 no mantiene cuotas para los inodos, por lo que estas columnas no se aplican a los sistemas de archivos GFS2 y estarán en blanco.

Si alguno de los valores está en 0, ese límite no está establecido. En el editor de texto, cambie los límites. Por ejemplo:

```
Disk quotas for user testuser (uid 501):
Filesystem              blocks    soft    hard    inodes  soft  hard
/dev/VolGroup00/LogVol02  440436  500000  550000
```

Para verificar que la cuota para el usuario se ha establecido, utilice el siguiente comando:

```
# quota testuser
```

También puedes establecer cuotas desde la línea de comandos con el comando **setquota**. Para obtener información sobre el comando **setquota**, consulte la página de manual **setquota**(8).

4.1.4. Asignación de cuotas por grupo

Las cuotas también se pueden asignar por grupos. Tenga en cuenta que si ha montado su sistema de archivos en modo contable (con la opción **account=on** especificada), las cuotas no se aplican.

Para establecer una cuota de grupo para el grupo **devel** (el grupo debe existir antes de establecer la cuota de grupo), utilice el siguiente comando:

```
# edquota -g devel
```

Este comando muestra la cuota existente para el grupo en el editor de texto:

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

El sistema de archivos GFS2 no mantiene cuotas para los inodos, por lo que estas columnas no se aplican a los sistemas de archivos GFS2 y estarán en blanco. Modifique los límites y guarde el archivo.

Para comprobar que se ha establecido la cuota de grupo, utilice el siguiente comando:

```
$ quota -g devel
```

4.2. GESTIÓN DE CUOTAS DE DISCO GFS2

Si se implantan las cuotas, necesitan cierto mantenimiento, sobre todo en forma de vigilancia para ver si se superan las cuotas y asegurarse de que éstas son exactas.

Si los usuarios exceden repetidamente sus cuotas o alcanzan constantemente sus límites blandos, el administrador del sistema tiene algunas opciones que tomar, dependiendo del tipo de usuarios que sean y del impacto del espacio en disco en su trabajo. El administrador puede ayudar al usuario a determinar cómo utilizar menos espacio en disco o aumentar la cuota de disco del usuario.

Puede crear un informe de uso del disco ejecutando la utilidad **repquota**. Por ejemplo, el comando **repquota /home** produce esta salida:

```
* 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 el informe de uso del disco para todos los sistemas de archivos con cuota (opción **-a**), utilice el comando

```
# repquota -a
```

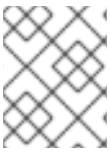
El **--** que aparece después de cada usuario es una forma rápida de determinar si se han superado los límites de bloque. Si se ha superado el límite suave de bloque, aparece un en lugar del primer **-** en la salida. El segundo **-** indica el límite de inodo, pero los sistemas de archivos GFS2 no admiten límites de inodo, por lo que ese carácter permanecerá como **-**. Los sistemas de archivos GFS2 no admiten un período de gracia, por lo que la columna **grace** permanecerá en blanco.

Tenga en cuenta que el comando **repquota** no es compatible con NFS, independientemente del sistema de archivos subyacente.

4.3. CÓMO MANTENER LA EXACTITUD DE LAS CUOTAS DE DISCO DE GFS2 CON EL COMANDO QUOTACHECK

Si habilita las cuotas en su sistema de archivos después de un período de tiempo en el que ha estado funcionando con las cuotas deshabilitadas, debe ejecutar el comando **quotacheck** para crear, comprobar y reparar los archivos de cuotas. Además, puede ejecutar el comando **quotacheck** si cree que sus archivos de cuotas pueden no ser precisos, como puede ocurrir cuando un sistema de archivos no se desmonta limpiamente después de un fallo del sistema.

Para obtener más información sobre el comando **quotacheck**, consulte la página de manual **quotacheck**.



NOTA

Ejecute **quotacheck** cuando el sistema de archivos esté relativamente inactivo en todos los nodos porque la actividad del disco puede afectar a los valores de cuota calculados.

4.4. SINCRONIZACIÓN DE CUOTAS CON EL COMANDO QUOTASYNC

GFS2 almacena toda la información de cuotas en su propio archivo interno en el disco. Un nodo de GFS2 no actualiza este archivo de cuotas con cada escritura del sistema de archivos, sino que, por defecto, actualiza el archivo de cuotas una vez cada 60 segundos. Esto es necesario para evitar la

contención entre los nodos que escriben en el archivo de cuotas, lo que provocaría una ralentización del rendimiento.

Cuando un usuario o grupo se acerca a su límite de cuota, GFS2 reduce dinámicamente el tiempo entre sus actualizaciones de archivos de cuota para evitar que se supere el límite. El período de tiempo normal entre sincronizaciones de cuotas es un parámetro ajustable, **quota_quantum**. Puede cambiar este valor predeterminado de 60 segundos utilizando la opción de montaje **quota_quantum=**, como se describe en la tabla "Opciones de montaje específicas de GFS2" en Montaje de un sistema de archivos GFS2 que especifica las opciones de montaje.

El parámetro **quota_quantum** debe establecerse en cada nodo y cada vez que se monte el sistema de archivos. Los cambios en el parámetro **quota_quantum** no son persistentes a través de los desmontajes. Puede actualizar el valor de **quota_quantum** con el parámetro **mount -o remount**.

Puede utilizar el comando **quotasync** para sincronizar la información de cuotas de un nodo con el archivo de cuotas en disco entre las actualizaciones automáticas realizadas por GFS2. Uso **Synchronizing Quota Information**

```
# `quotasync [-ug -a|mountpoint..a`].
```

u

Sincronizar los archivos de cuotas de usuarios.

g

Sincronizar los archivos de cuotas de grupo

a

Sincroniza todos los sistemas de archivos que están actualmente habilitados para cuotas y soportan la sincronización. Cuando -a está ausente, se debe especificar un punto de montaje del sistema de archivos.

mountpoint

Especifica el sistema de archivos GFS2 al que se aplican las acciones.

Puede ajustar el tiempo entre sincronizaciones especificando una opción de montaje **quota-quantum**.

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

MountPoint

Especifica el sistema de archivos GFS2 al que se aplican las acciones.

secs

Especifica el nuevo período de tiempo entre las sincronizaciones regulares de archivos de cuota por parte de GFS2. Los valores más pequeños pueden aumentar la contención y ralentizar el rendimiento.

El siguiente ejemplo sincroniza todas las cuotas sucias almacenadas en caché del nodo en el que se ejecuta al archivo de cuotas en disco para el sistema de archivos **/mnt/mygfs2**.

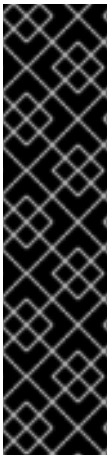
```
# quotasync -ug /mnt/mygfs2
```

El siguiente ejemplo cambia el período de tiempo por defecto entre las actualizaciones regulares de archivos de cuota a una hora (3600 segundos) para el sistema de archivos **/mnt/mygfs2** cuando se vuelve a montar ese sistema de archivos en el volumen lógico **/dev/volgroup/logical_volume**.

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

CAPÍTULO 5. REPARACIÓN DEL SISTEMA DE ARCHIVOS GFS2

Cuando los nodos fallan con el sistema de archivos montado, el registro en el diario del sistema de archivos permite una rápida recuperación. Sin embargo, si un dispositivo de almacenamiento pierde energía o se desconecta físicamente, puede producirse una corrupción del sistema de archivos. (El registro en el diario no puede utilizarse para recuperarse de los fallos del subsistema de almacenamiento). Cuando se produce ese tipo de corrupción, puede recuperar el sistema de archivos GFS2 utilizando el comando **fsck.gfs2**.



IMPORTANTE

El comando **fsck.gfs2** debe ejecutarse sólo en un sistema de archivos que esté desmontado de todos los nodos. Cuando el sistema de archivos se gestiona como un recurso de clúster de Pacemaker, puede desactivar el recurso del sistema de archivos, que desmonta el sistema de archivos. Después de ejecutar el comando **fsck.gfs2**, se vuelve a habilitar el recurso del sistema de archivos. El valor *timeout* especificado con la opción **--wait** del comando **pcs resource disable** indica un valor en segundos.

```
# pcs resource disable --wait=timeoutvalue resource_id
[fsck.gfs2]
# pcs resource enable resource_id
```

Para garantizar que el comando **fsck.gfs2** no se ejecute en un sistema de archivos GFS2 en el momento del arranque, puede establecer el parámetro **run_fsck** del argumento **options** al crear el recurso del sistema de archivos GFS2 en un clúster. Especificar **"run_fsck=no"** indicará que no se debe ejecutar el comando **fsck**.

5.1. DETERMINACIÓN DE LA MEMORIA NECESARIA PARA EJECUTAR FSCK.GFS2

La ejecución del comando **fsck.gfs2** puede requerir memoria del sistema más allá de la memoria utilizada por el sistema operativo y el kernel. Los sistemas de archivos más grandes, en particular, pueden requerir memoria adicional para ejecutar este comando.

La siguiente tabla muestra los valores aproximados de memoria que pueden ser necesarios para ejecutar sistemas de archivos **fsck.gfs2** en sistemas de archivos GFS2 de 1TB, 10TB y 100TB con un tamaño de bloque de 4K.

Tamaño del sistema de archivos GFS2	Memoria aproximada necesaria para ejecutar **fsck.gfs2**
1 TB	0.16 GB
10 TB	1.6 GB
100 TB	16 GB

Tenga en cuenta que un tamaño de bloque menor para el sistema de archivos requeriría una mayor cantidad de memoria. Por ejemplo, los sistemas de archivos GFS2 con un tamaño de bloque de 1K requerirían cuatro veces la cantidad de memoria indicada en esta tabla.

5.2. REPARACIÓN DE UN SISTEMA DE ARCHIVOS GFS2

A continuación se muestra el formato del comando **fsck.gfs2** para reparar un sistema de archivos GFS2.

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

-y

La bandera **-y** hace que todas las preguntas se respondan con **yes**. Con la bandera **-y** especificada, el comando **fsck.gfs2** no le pide una respuesta antes de hacer cambios.

BlockDevice

Especifica el dispositivo de bloque donde reside el sistema de archivos GFS2.

En este ejemplo, se repara el sistema de archivos GFS2 que reside en el dispositivo de bloque **/dev/testvg/testlv**. Todas las consultas de reparación se responden automáticamente con **yes**.

```
# 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
```

CAPÍTULO 6. MEJORA DEL RENDIMIENTO DE GFS2

Esta sección ofrece consejos para mejorar el rendimiento de GFS2.

Para obtener recomendaciones generales sobre la implantación y actualización de clusters de Red Hat Enterprise Linux utilizando el complemento de alta disponibilidad y Red Hat Global File System 2 (GFS2), consulte el artículo "Red Hat Enterprise Linux Cluster, High Availability, and GFS Deployment Best Practices" en el Portal del Cliente de Red Hat en https://access.redhat.com/kb/docs/DOC-40821.

6.1. DESFRAGMENTACIÓN DEL SISTEMA DE ARCHIVOS GFS2

Aunque no existe una herramienta de desfragmentación para GFS2 en Red Hat Enterprise Linux, puede desfragmentar archivos individuales identificándolos con la herramienta **filefrag**, copiándolos a archivos temporales y renombrando los archivos temporales para reemplazar los originales.

6.2. BLOQUEO DEL NODO GFS2

Para obtener el mejor rendimiento de un sistema de archivos GFS2, es importante entender parte de la teoría básica de su funcionamiento. Un sistema de archivos de un solo nodo se implementa junto con una caché, cuyo propósito es eliminar la latencia de los accesos al disco cuando se utilizan datos solicitados con frecuencia. En Linux, la caché de páginas (e históricamente la caché de búferes) proporciona esta función de caché.

Con GFS2, cada nodo tiene su propia caché de páginas que puede contener una parte de los datos en disco. GFS2 utiliza un mecanismo de bloqueo llamado *glocks* (se pronuncia gee-locks) para mantener la integridad de la caché entre nodos. El subsistema glock proporciona una función de gestión de la caché que se implementa utilizando el *distributed lock manager* (DLM) como capa de comunicación subyacente.

Los glocks proporcionan protección para la caché en base a cada nodo, por lo que hay un bloqueo por nodo que se utiliza para controlar la capa de caché. Si ese glock se concede en modo compartido (modo de bloqueo DLM: PR) entonces los datos bajo ese glock pueden ser almacenados en caché en uno o más nodos al mismo tiempo, de modo que todos los nodos pueden tener acceso local a los datos.

Si el glock se concede en modo exclusivo (modo de bloqueo DLM: EX) entonces sólo un único nodo puede almacenar en caché los datos bajo ese glock. Este modo es utilizado por todas las operaciones que modifican los datos (como la llamada al sistema **write**).

Si otro nodo solicita un glock que no puede ser concedido inmediatamente, entonces el DLM envía un mensaje al nodo o nodos que actualmente tienen los glocks que bloquean la nueva solicitud para pedirles que suelten sus bloqueos. La eliminación de los bloqueos puede ser (en comparación con la mayoría de las operaciones del sistema de archivos) un proceso largo. La eliminación de un glock compartido sólo requiere la invalidación de la caché, lo cual es relativamente rápido y proporcional a la cantidad de datos almacenados en la caché.

El abandono de un glock exclusivo requiere un vaciado del registro, y la escritura de cualquier dato modificado en el disco, seguido de la invalidación según el glock compartido.

La diferencia entre un sistema de archivos de un solo nodo y GFS2, por tanto, es que un sistema de archivos de un solo nodo tiene una única caché y GFS2 tiene una caché independiente en cada nodo. En ambos casos, la latencia para acceder a los datos almacenados en caché es de un orden de magnitud similar, pero la latencia para acceder a los datos no almacenados en caché es mucho mayor en GFS2 si otro nodo ha almacenado previamente esos mismos datos.

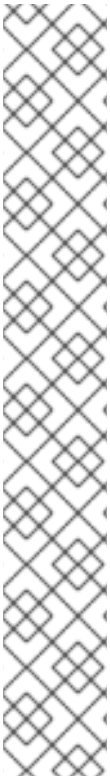
Operaciones como **read** (con buffer), **stat,** y **readdir** sólo requieren un glock compartido. Operaciones

como **write** (con buffer), **mkdir**, **rmdir**, y **unlink** requieren un glock exclusivo. Las operaciones de lectura/escritura de E/S directa requieren un glock diferido si no se está realizando ninguna asignación, o un glock exclusivo si la escritura requiere una asignación (es decir, ampliar el archivo, o rellenar agujeros).

Hay dos consideraciones principales de rendimiento que se derivan de esto. En primer lugar, las operaciones de sólo lectura se paralelizan muy bien en un clúster, ya que pueden ejecutarse independientemente en cada nodo. En segundo lugar, las operaciones que requieren un glock exclusivo pueden reducir el rendimiento, si hay varios nodos compitiendo por el acceso al mismo nodo(s). Por lo tanto, la consideración del conjunto de trabajo en cada nodo es un factor importante en el rendimiento del sistema de archivos GFS2, como cuando, por ejemplo, se realiza una copia de seguridad del sistema de archivos, como se describe en Copia de seguridad de un sistema de archivos GFS2 .

Otra consecuencia de esto es que recomendamos el uso de la opción de montaje **noatime** o **nodiratime** con GFS2 siempre que sea posible, con preferencia por **noatime** cuando la aplicación lo permita. Esto evita que las lecturas requieran bloqueos exclusivos para actualizar la marca de tiempo **atime**.

Para los usuarios que se preocupan por el conjunto de trabajo o la eficiencia de la caché, GFS2 proporciona herramientas que permiten supervisar el rendimiento de un sistema de archivos GFS2: Performance Co-Pilot y GFS2 tracepoints.



NOTA

Debido a la forma en que se implementa el almacenamiento en caché de GFS2, el mejor rendimiento se obtiene cuando se produce cualquiera de las siguientes situaciones:

- Un inodo se utiliza de forma de sólo lectura en todos los nodos.
- Un inodo se escribe o modifica desde un solo nodo.

Tenga en cuenta que la inserción y eliminación de entradas de un directorio durante la creación y eliminación de archivos cuenta como escritura en el inodo del directorio.

Es posible romper esta regla siempre que se rompa con relativa poca frecuencia. Ignorar esta regla con demasiada frecuencia dará lugar a una grave penalización del rendimiento.

Si **mmap**() un archivo en GFS2 con un mapeo de lectura/escritura, pero sólo lee de él, esto sólo cuenta como una lectura.

Si no se establece el parámetro **noatime mount** , las lecturas también darán lugar a escrituras para actualizar las marcas de tiempo de los archivos. Recomendamos que todos los usuarios de GFS2 monten con **noatime** a menos que tengan un requisito específico para **atime**.

6.3. PROBLEMAS CON EL BLOQUEO DE POSIX

Al utilizar el bloqueo Posix, debe tener en cuenta lo siguiente:

- El uso de Flocks dará lugar a un procesamiento más rápido que el uso de bloqueos Posix.
- Los programas que utilizan bloqueos Posix en GFS2 deben evitar el uso de la función **GETLK** ya que, en un entorno de clúster, el ID del proceso puede ser para un nodo diferente en el clúster.

6.4. AJUSTE DEL RENDIMIENTO CON GFS2

Por lo general, es posible alterar la forma en que una aplicación problemática almacena sus datos para obtener una considerable ventaja de rendimiento.

Un ejemplo típico de aplicación problemática es un servidor de correo electrónico. Estos suelen estar dispuestos con un directorio spool que contiene archivos para cada usuario (**mbox**), o con un directorio para cada usuario que contiene un archivo para cada mensaje (**maildir**). Cuando las peticiones llegan a través de IMAP, lo ideal es dar a cada usuario una afinidad con un nodo concreto. De esta forma, sus peticiones para ver y borrar mensajes de correo electrónico tenderán a ser servidas desde la caché de ese nodo. Obviamente, si ese nodo falla, la sesión puede reiniciarse en un nodo diferente.

Cuando el correo llega por medio de SMTP, de nuevo los nodos individuales pueden ser configurados para pasar el correo de un determinado usuario a un nodo particular por defecto. Si el nodo por defecto no está activo, entonces el mensaje puede ser guardado directamente en el spool de correo del usuario por el nodo receptor. Una vez más, este diseño está pensado para mantener determinados conjuntos de archivos en caché en un solo nodo en el caso normal, pero para permitir el acceso directo en caso de fallo del nodo.

Esta configuración permite el mejor uso de la caché de páginas de GFS2 y también hace que los fallos sean transparentes para la aplicación, ya sea **imap** o **smtp**.

Las copias de seguridad suelen ser otra área complicada. De nuevo, si es posible, es muy preferible hacer una copia de seguridad del conjunto de trabajo de cada nodo directamente desde el nodo que está almacenando en caché ese conjunto concreto de inodos. Si tienes un script de copia de seguridad que se ejecuta en un punto regular en el tiempo, y que parece coincidir con un pico en el tiempo de respuesta de una aplicación que se ejecuta en GFS2, entonces hay una buena posibilidad de que el clúster no esté haciendo el uso más eficiente de la caché de páginas.

Obviamente, si usted está en la posición de poder detener la aplicación para realizar una copia de seguridad, entonces esto no será un problema. Por otro lado, si una copia de seguridad se ejecuta desde un solo nodo, después de que se haya completado una gran parte del sistema de archivos se almacenará en caché en ese nodo, con una penalización de rendimiento para los accesos posteriores desde otros nodos. Esto se puede mitigar hasta cierto punto eliminando la caché de páginas VFS en el nodo de copia de seguridad después de que se haya completado la copia de seguridad con el siguiente comando:

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

Sin embargo, esta solución no es tan buena como asegurarse de que el conjunto de trabajo de cada nodo sea compartido, de sólo lectura en el clúster, o que se acceda a él principalmente desde un solo nodo.

6.5. RESOLUCIÓN DE PROBLEMAS DE RENDIMIENTO DE GFS2 CON EL VOLCADO DE BLOQUEOS DE GFS2

Si el rendimiento de su clúster se ve afectado por el uso ineficiente de la caché de GFS2, es posible que vea tiempos de espera de E/S grandes y crecientes. Puede hacer uso de la información de volcado de bloqueos de GFS2 para determinar la causa del problema.

Esta sección proporciona una visión general del volcado de bloqueos de GFS2.

La información de volcado de bloqueos de GFS2 puede obtenerse del archivo **debugfs** que puede encontrarse en el siguiente nombre de ruta, asumiendo que **debugfs** está montado en **/sys/kernel/debug/**:

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

El contenido del archivo es una serie de líneas. Cada línea que comienza con G: representa un glock, y las líneas siguientes, sangradas por un solo espacio, representan un elemento de información relacionado con el glock inmediatamente anterior en el archivo.

La mejor manera de utilizar el archivo **debugfs** es usar el comando **cat** para tomar una copia del contenido completo del archivo (puede tomar mucho tiempo si tiene una gran cantidad de RAM y muchos inodos en caché) mientras la aplicación está experimentando problemas, y luego mirar los datos resultantes en una fecha posterior.

NOTA

Puede ser útil hacer dos copias del archivo **debugfs**, una unos segundos o incluso uno o dos minutos después de la otra. Comparando la información del soporte en las dos trazas relacionadas con el mismo número de glock, se puede saber si la carga de trabajo está progresando (sólo es lenta) o si se ha quedado atascada (lo que siempre es un error y debe ser reportado al soporte de Red Hat inmediatamente).

Las líneas del archivo **debugfs** que comienzan con H: (holders) representan solicitudes de bloqueo concedidas o en espera de ser concedidas. El campo de banderas en la línea de titulares f: muestra cuál: La bandera 'W' se refiere a una solicitud en espera, la bandera 'H' se refiere a una solicitud concedida. Los glocks que tienen un gran número de solicitudes en espera son probablemente los que están experimentando una contención particular.

Tabla 6.1, “Banderas Glock” muestra los significados de las diferentes banderas de las glock y Tabla 6.2, “Banderas de soporte de Glock” muestra los significados de las diferentes banderas de los soportes de las glock.

Tabla 6.1. Banderas Glock

Bandera	Nombre	Significado
b	Bloqueo	Válido cuando la bandera de bloqueo está activada, e indica que la operación que se ha solicitado al DLM puede bloquearse. Esta bandera se borra para las operaciones de descenso y para los bloqueos "try". El propósito de esta bandera es permitir la recopilación de estadísticas del tiempo de respuesta del DLM independientemente del tiempo que tomen otros nodos para degradar los bloqueos.
d	Pendiente de degradación	Una solicitud de baja aplazada (remota)
D	Desplazar a	Una solicitud de baja (local o remota)

Bandera	Nombre	Significado
f	Descarga de troncos	El registro necesita ser comprometido antes de liberar esta glock
F	Congelado	Las respuestas de los nodos remotos se ignoran - la recuperación está en curso. Esta bandera no está relacionada con la congelación del sistema de archivos, que utiliza un mecanismo diferente, sino que se utiliza sólo en la recuperación.
i	Invalidación en curso	En el proceso de invalidación de páginas bajo este glock
I	Inicialmente	Se establece cuando el bloqueo DLM está asociado a esta glock
l	Bloqueado	La glock está en proceso de cambio de estado
L	LRU	Se establece cuando la glock está en la lista LRU
o	Objeto	Se establece cuando el glock está asociado a un objeto (es decir, un inodo para los glocks de tipo 2, y un grupo de recursos para los glocks de tipo 3)
p	Descenso de categoría en curso	El glock está en proceso de responder a una solicitud de degradación
q	En cola	Se establece cuando un portador está en la cola de una glock, y se borra cuando la glock está retenida, pero no hay portadores restantes. Se utiliza como parte del algoritmo que calcula el tiempo mínimo de retención de una cerradura.
r	Respuesta pendiente	La respuesta recibida del nodo remoto está a la espera de ser procesada

Bandera	Nombre	Significado
y	Dirty	Los datos necesitan ser lavados en el disco antes de liberar este glock

Tabla 6.2. Banderas de soporte de Glock

Bandera	Nombre	Significado
a	Async	No espere el resultado de glock (hará una encuesta para el resultado más tarde)
A	Cualquier	Se acepta cualquier modo de bloqueo compatible
c	No hay caché	Cuando se desbloquea, baja el bloqueo DLM inmediatamente
e	No caduca	Ignorar las solicitudes de cancelación de bloqueo posteriores
E	exactamente	Debe tener el modo de bloqueo exacto
F	Primero	Se establece cuando el titular es el primero en ser concedido para esta cerradura
H	Titular	Indica que se ha concedido el bloqueo solicitado
p	Prioridad	Colocar al titular de la cola en la cabeza de la cola
t	Prueba con	Una cerradura "try"
T	Prueba 1CB	Un bloqueo "try" que envía un callback
W	Espera	Se establece mientras se espera a que se complete la solicitud

Una vez identificado un glock que está causando un problema, el siguiente paso es averiguar a qué inodo se refiere. El número de glock (n: en la línea G:) lo indica. Es de la forma *type*/*number* y si *type* es 2, entonces el glock es un glock de inodo y *number* es un número de inodo. Para localizar el inodo, puede

ejecutar **find -inum *number*** donde *number* es el número de inodo convertido del formato hexadecimal del archivo glocks a decimal.

AVISO

Si ejecuta el comando **find** en un sistema de archivos cuando está experimentando contención de bloqueos, es probable que empeore el problema. Es una buena idea detener la aplicación antes de ejecutar el comando **find** cuando se buscan inodos en contención.

Tabla 6.3, "Tipos de Glock" muestra los significados de los diferentes tipos de glock.

Tabla 6.3. Tipos de Glock

Tipo de número	Tipo de cerradura	Utilice
1	Trans	Bloqueo de la transacción
2	Inodo	Metadatos y datos del inodo
3	Rgrp	Metadatos del grupo de recursos
4	Meta	El superbloque
5	Abrir	Detección del último inodo más cercano
6	Flock	**flock**(2) syscall
8	Cuota	Operaciones de contingencia
9	Diario	Diario mutex

Si el glock que se identificó era de un tipo diferente, lo más probable es que sea del tipo 3: (grupo de recursos). Si ve un número significativo de procesos esperando por otros tipos de glock bajo cargas normales, infórmelo al soporte de Red Hat.

Si ves un número de solicitudes en espera en el bloqueo de un grupo de recursos, puede haber varias razones para ello. Una de ellas es que haya un gran número de nodos en comparación con el número de grupos de recursos en el sistema de archivos. Otra es que el sistema de archivos puede estar casi lleno (lo que requiere, por término medio, más tiempo de búsqueda de bloques libres). La situación en ambos casos puede mejorarse añadiendo más almacenamiento y utilizando el comando **gfs2_grow** para ampliar el sistema de archivos.

6.6. HABILITACIÓN DEL REGISTRO DE DATOS EN EL DIARIO

Normalmente, GFS2 sólo escribe metadatos en su diario. El contenido de los archivos se escribe posteriormente en el disco mediante la sincronización periódica del kernel que vacía los búferes del sistema de archivos. Una llamada a **fsync()** en un archivo hace que los datos del archivo se escriban en el disco inmediatamente. La llamada regresa cuando el disco informa que todos los datos se han escrito con seguridad.

El registro en el diario de datos puede dar lugar a una reducción del tiempo de **fsync()** para archivos muy pequeños, ya que los datos del archivo se escriben en el diario además de los metadatos. Esta ventaja se reduce rápidamente a medida que aumenta el tamaño del archivo. La escritura en archivos medianos y grandes será mucho más lenta con el registro en el diario de datos activado.

Las aplicaciones que dependen de **fsync()** para sincronizar los datos de los archivos pueden ver mejorado su rendimiento si utilizan el registro en el diario de datos. El registro en el diario de datos puede activarse automáticamente para cualquier archivo GFS2 creado en un directorio marcado (y todos sus subdirectorios). También se puede activar o desactivar el registro en el diario de datos de los archivos existentes con longitud cero.

La activación del registro en el diario de datos en un directorio establece que el directorio "hereda jdata", lo que indica que todos los archivos y directorios creados posteriormente en ese directorio se registran en el diario. Puede activar y desactivar el registro en el diario de datos de un archivo con el comando **chattr**.

Los siguientes comandos habilitan el registro en el diario de datos en el archivo **/mnt/gfs2/gfs2_dir/newfile** y luego comprueban si el indicador se ha establecido correctamente.

```
# chattr +j /mnt/gfs2/gfs2_dir/newfile
# lsattr /mnt/gfs2/gfs2_dir
---------j--- /mnt/gfs2/gfs2_dir/newfile
```

Los siguientes comandos desactivan el registro en el diario de datos en el archivo **/mnt/gfs2/gfs2_dir/newfile** y luego comprueban si el indicador se ha establecido correctamente.

```
# chattr -j /mnt/gfs2/gfs2_dir/newfile
# lsattr /mnt/gfs2/gfs2_dir
------------- /mnt/gfs2/gfs2_dir/newfile
```

También puede utilizar el comando **chattr** para establecer la bandera **j** en un directorio. Cuando se establece este indicador para un directorio, todos los archivos y directorios creados posteriormente en ese directorio se registran en el diario. El siguiente conjunto de comandos establece la bandera **j** en el directorio **gfs2_dir**, y luego comprueba si la bandera se ha establecido correctamente. Después de esto, los comandos crean un nuevo archivo llamado **newfile** en el directorio **/mnt/gfs2/gfs2_dir** y luego comprueban si la bandera **j** ha sido establecida para el archivo. Dado que la bandera **j** está establecida para el directorio, entonces **newfile** también debería tener el registro en el diario habilitado.

```
# chattr -j /mnt/gfs2/gfs2_dir
# lsattr /mnt/gfs2
---------j--- /mnt/gfs2/gfs2_dir
# touch /mnt/gfs2/gfs2_dir/newfile
# lsattr /mnt/gfs2/gfs2_dir
---------j--- /mnt/gfs2/gfs2_dir/newfile
```

CAPÍTULO 7. DIAGNÓSTICO Y CORRECCIÓN DE PROBLEMAS EN LOS SISTEMAS DE ARCHIVOS GFS2

Esta sección proporciona información sobre algunos problemas comunes de GFS2 y cómo resolverlos.

7.1. SISTEMA DE ARCHIVOS GFS2 NO DISPONIBLE PARA UN NODO (LA FUNCIÓN DE RETIRADA DE GFS2)

La función GFS2 *withdraw* es una característica de integridad de los datos del sistema de archivos GFS2 que evita posibles daños en el sistema de archivos debido a un hardware o software de kernel defectuoso. Si el módulo del kernel de GFS2 detecta una incoherencia mientras se utiliza un sistema de archivos GFS2 en un nodo de clúster determinado, se retira del sistema de archivos, dejándolo indisponible para ese nodo hasta que se desmonte y se vuelva a montar (o se reinicie la máquina que detecta el problema). Todos los demás sistemas de archivos GFS2 montados siguen siendo totalmente funcionales en ese nodo. (La función de retirada de GFS2 es menos severa que un pánico del kernel, que hace que el nodo sea cercado)

Las principales categorías de incoherencias que pueden provocar una retirada de GFS2 son las siguientes:

- Error de consistencia del inodo
- Error de consistencia del grupo de recursos
- Error de consistencia del diario
- Error de consistencia de los metadatos del número mágico
- Error de consistencia del tipo de metadatos

Un ejemplo de una incoherencia que podría causar una retirada de GFS2 es un recuento de bloques incorrecto para el inodo de un archivo. Cuando GFS2 borra un archivo, elimina sistemáticamente todos los bloques de datos y metadatos a los que hace referencia ese archivo. Cuando lo hace, comprueba el recuento de bloques del inodo. Si el recuento de bloques no es 1 (lo que significa que todo lo que queda es el propio inodo del disco), eso indica una inconsistencia del sistema de archivos, ya que el recuento de bloques del inodo no coincide con los bloques reales utilizados para el archivo.

En muchos casos, el problema puede haber sido causado por un hardware defectuoso (memoria defectuosa, placa base, HBA, unidades de disco, cables, etc.). También puede haber sido causado por un error del kernel (otro módulo del kernel que sobrescribe accidentalmente la memoria de GFS2), o un daño real del sistema de archivos (causado por un error de GFS2).

En la mayoría de los casos, la mejor manera de recuperarse de un sistema de archivos GFS2 retirado es reiniciar o cercar el nodo. El sistema de archivos GFS2 retirado le dará la oportunidad de reubicar los servicios en otro nodo del clúster. Una vez reubicados los servicios, puede reiniciar el nodo o forzar un vallado con este comando.

```
# pcs stonith fence node
```

AVISO

No intente desmontar y volver a montar el sistema de archivos manualmente con los comandos **umount** y **mount**. Debe utilizar el comando **pcs**, de lo contrario Pacemaker detectará que el servicio del sistema de archivos ha desaparecido y cerrará el nodo.

El problema de consistencia que causó la retirada puede hacer que sea imposible detener el servicio del sistema de archivos, ya que puede hacer que el sistema se cuelgue.

Si el problema persiste después de un nuevo montaje, debe detener el servicio del sistema de archivos para desmontar el sistema de archivos de todos los nodos del clúster y, a continuación, realizar una comprobación del sistema de archivos con el comando fsck.gfs2 antes de reiniciar el servicio con el siguiente procedimiento.

1. Reinicie el nodo afectado.
2. Desactive el servicio de sistema de archivos no clonados en Pacemaker para desmontar el sistema de archivos de cada nodo del clúster.

```
# pcs resource disable --wait=100 mydata_fs
```

3. Desde un nodo del clúster, ejecute el comando **fsck.gfs2** en el dispositivo del sistema de archivos para comprobar y reparar cualquier daño en el sistema de archivos.

```
# fsck.gfs2 -y /dev/vg_mydata/mydata > /tmp/fsck.out
```

4. Vuelva a montar el sistema de archivos GFS2 desde todos los nodos volviendo a habilitar el servicio del sistema de archivos:

```
# pcs resource enable --wait=100 mydata_fs
```

Puede anular la función de retirada de GFS2 montando el sistema de archivos con la opción **-o errors=panic** especificada en el servicio del sistema de archivos.

```
# pcs resource update mydata_fs “options=noatime,errors=panic”
```

Cuando se especifica esta opción, cualquier error que normalmente provocaría la retirada del sistema fuerza un kernel panic en su lugar. Esto detiene las comunicaciones del nodo, lo que hace que el nodo sea cercado. Esto es especialmente útil para los clústeres que se dejan desatendidos durante largos períodos de tiempo sin supervisión o intervención.

Internamente, la función de retirada de GFS2 funciona desconectando el protocolo de bloqueo para garantizar que todas las operaciones posteriores del sistema de archivos den lugar a errores de E/S. Como resultado, cuando se produce la retirada, es normal ver una serie de errores de E/S del dispositivo de mapeo de dispositivos reportados en los registros del sistema.

7.2. EL SISTEMA DE ARCHIVOS GFS2 SE CUELGA Y REQUIERE EL REINICIO DE UN NODO

Si su sistema de archivos GFS2 se cuelga y no devuelve los comandos ejecutados en él, pero al reiniciar un nodo específico el sistema vuelve a la normalidad, esto puede ser indicativo de un problema de bloqueo o de un error. Si esto ocurre, reúna los datos de GFS2 durante una de estas ocurrencias y abra un ticket de soporte con el Soporte de Red Hat, como se describe en Recopilación de datos de GFS2 para la resolución de problemas.

7.3. EL SISTEMA DE ARCHIVOS GFS2 SE CUELGA Y REQUIERE EL REINICIO DE TODOS LOS NODOS

Si su sistema de archivos GFS2 se cuelga y no devuelve los comandos que se ejecutan en él, requiriendo que reinicie todos los nodos del clúster antes de utilizarlo, compruebe los siguientes problemas.

- Es posible que haya fallado una valla. Los sistemas de archivos GFS2 se congelan para garantizar la integridad de los datos en caso de que falle una valla. Compruebe los registros de mensajes para ver si hay algún vallado fallido en el momento del cuelgue. Asegúrese de que el cercado está configurado correctamente.
- Es posible que el sistema de archivos GFS2 se haya retirado. Busque en los registros de mensajes la palabra **withdraw** y compruebe si hay mensajes y rastros de llamadas de GFS2 que indiquen que el sistema de archivos se ha retirado. Una retirada es indicativa de una corrupción del sistema de archivos, un fallo de almacenamiento o un error. Cuando sea conveniente desmontar el sistema de archivos, deberá realizar el siguiente procedimiento:

  a. Reinicie el nodo en el que se produjo la retirada.

  ```
  # /sbin/reboot
  ```

  b. Detenga el recurso del sistema de archivos para desmontar el sistema de archivos GFS2 en todos los nodos.

  ```
  # pcs resource disable --wait=100 mydata_fs
  ```

  c. Capture los metadatos con el comando **gfs2_edit savemeta…**. Debe asegurarse de que hay espacio suficiente para el archivo, que en algunos casos puede ser grande. En este ejemplo, los metadatos se guardan en un archivo en el directorio **/root**.

  ```
  # gfs2_edit savemeta /dev/vg_mydata/mydata /root/gfs2metadata.gz
  ```

  d. Actualice el paquete **gfs2-utils**.

  ```
  # sudo yum update gfs2-utils
  ```

  e. En un nodo, ejecute el comando **fsck.gfs2** en el sistema de archivos para asegurar la integridad del sistema de archivos y reparar cualquier daño.

  ```
  # fsck.gfs2 -y /dev/vg_mydata/mydata > /tmp/fsck.out
  ```

  f. Una vez finalizado el comando **fsck.gfs2**, vuelva a habilitar el recurso del sistema de archivos para que vuelva a estar en servicio:

  ```
  # pcs resource enable --wait=100 mydata_fs
  ```

g. Abra un ticket de soporte con el Soporte de Red Hat. Infórmeles de que ha experimentado una retirada de GFS2 y proporcione los registros y la información de depuración generada por los comandos **sosreports** y **gfs2_edit savemeta**.
En algunos casos de retirada de GFS2, los comandos que intentan acceder al sistema de archivos o a su dispositivo de bloques pueden colgarse. En estos casos se requiere un reinicio duro para reiniciar el clúster.

Para obtener información sobre la función de retirada de GFS2, consulte Sistema de archivos GFS2 no disponible para un nodo (la función de retirada de GFS2).

- Este error puede ser indicativo de un problema de bloqueo o error. Recopile datos durante una de estas ocurrencias y abra un ticket de soporte con el Soporte de Red Hat, como se describe en Recopilación de datos de GFS2 para la resolución de problemas .

7.4. EL SISTEMA DE ARCHIVOS GFS2 NO SE MONTA EN EL NODO DE CLÚSTER RECIÉN AÑADIDO

Si añade un nuevo nodo a un clúster y descubre que no puede montar su sistema de archivos GFS2 en ese nodo, es posible que tenga menos diarios en el sistema de archivos GFS2 que nodos que intenten acceder al sistema de archivos GFS2. Debe tener un diario por cada host GFS2 en el que pretenda montar el sistema de archivos (a excepción de los sistemas de archivos GFS2 montados con la opción de montaje **spectator**, ya que éstos no requieren un diario). Puede añadir diarios a un sistema de archivos GFS2 con el comando **gfs2_jadd**. Añadir diarios a un sistema de archivos GFS2.

7.5. ESPACIO INDICADO COMO UTILIZADO EN EL SISTEMA DE ARCHIVOS VACÍO

If you have an empty GFS2 file system, the **df** command will show that there is space being taken up. This is because GFS2 file system journals consume space (number of journals * journal size) on disk. If you created a GFS2 file system with a large number of journals or specified a large journal size then you will be see (number of journals * journal size) as already in use when you execute the **df** command. Even if you did not specify a large number of journals or large journals, small GFS2 file systems (in the 1GB or less range) will show a large amount of space as being in use with the default GFS2 journal size.

7.6. RECOGIDA DE DATOS DE GFS2 PARA LA RESOLUCIÓN DE PROBLEMAS

Si su sistema de archivos GFS2 se cuelga y no devuelve los comandos que se ejecutan contra él y se ve en la necesidad de abrir un ticket con el Soporte de Red Hat, primero debería reunir los siguientes datos:

- El volcado de bloqueos GFS2 para el sistema de archivos de cada nodo:

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

- El volcado de bloqueo DLM para el sistema de archivos en cada nodo: Puede obtener esta información con el **dlm_tool**:

```
dlm_tool lockdebug -sv lsname.
```

En este comando, *lsname* es el nombre del espacio de cierre utilizado por DLM para el sistema de archivos en cuestión. Puedes encontrar este valor en la salida del comando **group_tool**.

- La salida del comando **sysrq -t**.

- El contenido del archivo /**var**/**log**/**messages**.

Una vez que haya reunido esos datos, puede abrir un ticket con el Soporte de Red Hat y proporcionar los datos que ha recopilado.

CAPÍTULO 8. DEPURACIÓN DE SISTEMAS DE ARCHIVOS GFS2 CON TRACEPOINTS GFS2 Y EL ARCHIVO DEBUGFS GLOCKS

Esta sección describe tanto la interfaz glock **debugfs** como los tracepoints de GFS2. Está pensada para usuarios avanzados que estén familiarizados con los aspectos internos del sistema de archivos que quieran aprender más sobre el diseño de GFS2 y cómo depurar problemas específicos de GFS2.

8.1. TIPOS DE TRACEPOINT GFS2

Actualmente hay tres tipos de tracepoints GFS2: *glock* (pronunciado \ "gee-lock") tracepoints, *bmap* tracepoints y *log* tracepoints. Estos pueden ser usados para monitorear un sistema de archivos GFS2 en ejecución y dar información adicional a la que puede ser obtenida con las opciones de depuración soportadas en versiones anteriores de Red Hat Enterprise Linux. Los tracepoints son particularmente útiles cuando un problema, como un cuelgue o un problema de rendimiento, es reproducible y por lo tanto la salida del tracepoint puede obtenerse durante la operación problemática. En GFS2, los glocks son el principal mecanismo de control de la caché y son la clave para entender el rendimiento del núcleo de GFS2. Los tracepoints bmap (mapa de bloques) pueden utilizarse para supervisar las asignaciones de bloques y el mapeo de bloques (búsqueda de bloques ya asignados en el árbol de metadatos del disco) a medida que se producen y comprobar cualquier problema relacionado con la localidad de acceso. Los tracepoints de registro hacen un seguimiento de los datos que se escriben y liberan del diario y pueden proporcionar información útil sobre esa parte de GFS2.

Los tracepoints están diseñados para ser lo más genéricos posible. Esto debería significar que no será necesario cambiar la API durante el transcurso de Red Hat Enterprise Linux 8. Por otro lado, los usuarios de esta interfaz deberían ser conscientes de que se trata de una interfaz de depuración y no forma parte del conjunto normal de API de Red Hat Enterprise Linux 8, y como tal Red Hat no garantiza que no se produzcan cambios en la interfaz de tracepoints de GFS2.

Los tracepoints son una característica genérica de Red Hat Enterprise Linux y su alcance va mucho más allá de GFS2. En particular, se utilizan para implementar la infraestructura **blktrace** y los tracepoints de **blktrace** pueden utilizarse en combinación con los de GFS2 para obtener una imagen más completa del rendimiento del sistema. Debido al nivel en el que operan los tracepoints, pueden producir grandes volúmenes de datos en un periodo de tiempo muy corto. Se han diseñado para que supongan una carga mínima para el sistema cuando están activados, pero es inevitable que tengan algún efecto. El filtrado de los eventos por diversos medios puede ayudar a reducir el volumen de datos y a centrarse en la obtención de sólo la información que es útil para entender cualquier situación particular.

8.2. TRACEPOINTS

Los tracepoints se encuentran en el directorio **/sys/kernel/debug/tracing/** suponiendo que **debugfs** esté montado en el lugar estándar del directorio **/sys/kernel/debug**. El subdirectorio **events** contiene todos los eventos de rastreo que se pueden especificar y, siempre que se cargue el módulo **gfs2**, habrá un subdirectorio **gfs2** que contiene otros subdirectorios, uno para cada evento GFS2. El contenido del directorio **/sys/kernel/debug/tracing/events/gfs2** debería ser más o menos el siguiente:

```
[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 activar todos los tracepoints GFS2, introduzca el siguiente comando:

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

Para habilitar un tracepoint específico, hay un archivo **enable** en cada uno de los subdirectorios de eventos individuales. Lo mismo ocurre con el archivo **filter**, que puede utilizarse para establecer un filtro de eventos para cada evento o conjunto de eventos. El significado de los eventos individuales se explica con más detalle a continuación.

La salida de los tracepoints está disponible en formato ASCII o binario. Este apéndice no cubre actualmente la interfaz binaria. La interfaz ASCII está disponible de dos maneras. Para listar el contenido actual del buffer del anillo, puede introducir el siguiente comando:

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

Esta interfaz es útil en los casos en los que se utiliza un proceso de larga duración durante un cierto período de tiempo y, después de algún evento, se desea consultar la última información capturada en el buffer. Una interfaz alternativa, **/sys/kernel/debug/tracing/trace_pipe**, se puede utilizar cuando se requiere toda la salida. Los eventos se leen de este archivo a medida que ocurren; no hay información histórica disponible a través de esta interfaz. El formato de la salida es el mismo desde ambas interfaces y se describe para cada uno de los eventos de GFS2 en las secciones posteriores de este apéndice.

Existe una utilidad llamada **trace-cmd** para leer los datos de los tracepoints. Para más información sobre esta utilidad, consulte el enlace en Sección 8.10, “Referencias”. La utilidad **trace-cmd** puede utilizarse de forma similar a la utilidad **strace**, por ejemplo para ejecutar un comando mientras se recopilan datos de trazas de varias fuentes.

8.3. GLOCKS

Para entender GFS2, el concepto más importante que hay que entender, y el que lo diferencia de otros sistemas de archivos, es el concepto de glocks. En términos del código fuente, un glock es una estructura de datos que reúne el DLM y la caché en una sola máquina de estado. Cada glock tiene una relación 1:1 con un único bloqueo del DLM, y proporciona el almacenamiento en caché de ese estado de bloqueo para que las operaciones repetitivas realizadas desde un único nodo del sistema de archivos no tengan que llamar repetidamente al DLM, y así ayudan a evitar el tráfico de red innecesario. Existen dos grandes categorías de glocks, los que almacenan en caché los metadatos y los que no. Los glocks de inodo y los glocks de grupo de recursos almacenan en caché los metadatos, los otros tipos de glocks no almacenan en caché los metadatos. El inode glock también participa en el almacenamiento en caché de los datos además de los metadatos y tiene la lógica más compleja de todos los glocks.

Tabla 8.1. Modos de Glock y modos de bloqueo DLM

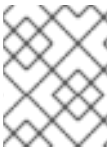
Modo Glock	Modo de bloqueo DLM	Notas
ONU	IV/NL	Desbloqueado (sin bloqueo DLM asociado a la glock o bloqueo NL dependiendo de la bandera I)
SH	PR	Bloqueo compartido (lectura protegida)
EX	EX	Cerradura exclusiva

Modo Glock	Modo de bloqueo DLM	Notas
DF	CW	Diferido (escritura concurrente) utilizado para la E/S directa y la congelación del sistema de archivos

Los glocks permanecen en memoria hasta que se desbloquean (a petición de otro nodo o a petición de la VM) y no hay usuarios locales. En ese momento se eliminan de la tabla hash de glock y se liberan. Cuando se crea un glock, el bloqueo DLM no se asocia con el glock inmediatamente. El bloqueo DLM se asocia con el glock en la primera petición al DLM, y si esta petición tiene éxito entonces la bandera 'I' (inicial) se establecerá en el glock. Tabla 8.4, “Banderas Glock” muestra los significados de las diferentes banderas del glock. Una vez que el DLM ha sido asociado con el glock, el bloqueo del DLM siempre permanecerá al menos en el modo de bloqueo NL (Nulo) hasta que el glock sea liberado. Una degradación del bloqueo DLM de NL a desbloqueado es siempre la última operación en la vida de un glock.

Cada glock puede tener un número de "holders" asociados, cada uno de los cuales representa una solicitud de bloqueo de las capas superiores. Las llamadas del sistema relacionadas con GFS2 ponen en cola y decaen los titulares del glock para proteger la sección crítica del código.

La máquina de estado glock se basa en una cola de trabajo. Por razones de rendimiento, los tasklets serían preferibles; sin embargo, en la implementación actual necesitamos enviar E/S desde ese contexto, lo que prohíbe su uso.



NOTA

Las colas de trabajo tienen sus propios tracepoints que pueden utilizarse en combinación con los tracepoints de GFS2.

Tabla 8.2, “Modos y tipos de datos de Glock” muestra qué estado puede ser almacenado en caché bajo cada uno de los modos de glock y si ese estado almacenado en caché puede estar sucio. Esto se aplica tanto a los bloqueos de inodos como a los de grupos de recursos, aunque no hay ningún componente de datos para los bloqueos de grupos de recursos, sólo metadatos.

Tabla 8.2. Modos y tipos de datos de Glock

Modo Glock	Datos de la caché	Metadatos de la caché	Datos sucios	Metadatos sucios
ONU	No	No	No	No
SH	Sí	Sí	No	No
DF	No	Sí	No	No
EX	Sí	Sí	Sí	Sí

8.4. LA INTERFAZ GLOCK DEBUGFS

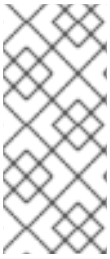
La interfaz de glock **debugfs** permite visualizar el estado interno de los glock y los soportes y también incluye algunos detalles resumidos de los objetos que se bloquean en algunos casos. Cada línea del archivo o bien comienza con G: sin sangría (que se refiere al propio glock) o bien comienza con una letra diferente, sangrada con un solo espacio, y se refiere a las estructuras asociadas al glock inmediatamente superior en el archivo (H: es un holder, I: un inodo, y R: un grupo de recursos) . He aquí un ejemplo de cómo podría ser el contenido de este archivo:

```
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]
```

El ejemplo anterior es una serie de extractos (de un archivo de aproximadamente 18 MB) generados por el comando **cat /sys/kernel/debug/gfs2/unity:myfs/glocks >my.lock** durante una ejecución del benchmark postmark en un sistema de archivos GFS2 de un solo nodo. Los glocks de la figura se han seleccionado para mostrar algunas de las características más interesantes de los volcados de glock.

Los estados de glock son EX (exclusivo), DF (diferido), SH (compartido) o UN (desbloqueado). Estos estados se corresponden directamente con los modos de bloqueo DLM, excepto UN, que puede representar el estado de bloqueo DLM nulo, o que GFS2 no mantiene un bloqueo DLM (dependiendo de la bandera I, como se ha explicado anteriormente). El campo s: del glock indica el estado actual del bloqueo y el mismo campo en el holder indica el modo solicitado. Si el bloqueo se concede, el soporte tendrá el bit H activado en sus banderas (campo f:). En caso contrario, tendrá activado el bit de espera W.

El campo n: (número) indica el número asociado a cada elemento. En el caso de los glocks, es el número de tipo seguido del número de glock, de modo que en el ejemplo anterior, el primer glock es n:5/75320; lo que indica un glock **iopen** que se relaciona con el inodo 75320. En el caso de los glocks de inodo y **iopen**, el número de glock es siempre idéntico al número de bloque de disco del inodo.



NOTA

Los números de glock (campo n:) en el archivo glocks de debugfs están en hexadecimal, mientras que la salida de tracepoints los lista en decimal. Esto es por razones históricas; los números de glock siempre se escribían en hexadecimal, pero se eligió el decimal para los tracepoints para que los números pudieran ser fácilmente comparados con la salida de otros tracepoints (de **blktrace** por ejemplo) y con la salida de **stat**(1).

El listado completo de todas las banderas, tanto para el titular como para el glock, se encuentra en Tabla 8.4, “Banderas Glock” y Tabla 8.5, “Banderas de soporte de Glock” . El contenido de los bloques de valores de los candados no está disponible actualmente a través de la interfaz de glock **debugfs**.

Tabla 8.3, "Tipos de Glock" muestra los significados de los diferentes tipos de glock.

Tabla 8.3. Tipos de Glock

Tipo de número	Tipo de cerradura	Utilice
1	trans	Bloqueo de la transacción
2	inode	Metadatos y datos del inodo
3	rgrp	Metadatos del grupo de recursos
4	meta	El superbloque
5	iopen	Detección del último inodo más cercano
6	rebaño	**flock**(2) syscall
8	cuota	Operaciones de contingencia
9	revista	Diario mutex

Una de las banderas más importantes de glock es la bandera l (locked). Este es el bit de bloqueo que se utiliza para arbitrar el acceso al estado de glock cuando se va a realizar un cambio de estado. Se establece cuando la máquina de estado está a punto de enviar una solicitud de bloqueo remoto a través del DLM, y sólo se borra cuando la operación completa se ha realizado. A veces esto puede significar que se haya enviado más de una solicitud de bloqueo, con varias invalidaciones que se producen entre las veces.

Tabla 8.4, "Banderas Glock" muestra los significados de las diferentes banderas de las glock.

Tabla 8.4. Banderas Glock

Bandera	Nombre	Significado
d	Pendiente de degradación	Una solicitud de baja aplazada (remota)
D	Desplazar a	Una solicitud de baja (local o remota)
f	Descarga de troncos	El registro necesita ser comprometido antes de liberar esta glock
F	Congelado	Respuestas de los nodos remotos ignoradas - la recuperación está en curso.

Bandera	Nombre	Significado
i	Invalidación en curso	En el proceso de invalidación de páginas bajo este glock
I	Inicialmente	Se establece cuando el bloqueo DLM está asociado a esta glock
l	Bloqueado	La glock está en proceso de cambio de estado
L	LRU	Se establece cuando el glock está en la lista LRU`
o	Objeto	Se establece cuando el glock está asociado a un objeto (es decir, un inodo para los glocks de tipo 2, y un grupo de recursos para los glocks de tipo 3)
p	Descenso de categoría en curso	El glock está en proceso de responder a una solicitud de degradación
q	En cola	Se establece cuando un portador está en la cola de una glock, y se borra cuando la glock está retenida, pero no hay portadores restantes. Se utiliza como parte del algoritmo que calcula el tiempo mínimo de retención de una cerradura.
r	Respuesta pendiente	La respuesta recibida del nodo remoto está a la espera de ser procesada
y	Dirty	Los datos necesitan ser lavados en el disco antes de liberar este glock

Cuando se recibe una devolución de llamada remota de un nodo que quiere obtener un bloqueo en un modo que entra en conflicto con el que se mantiene en el nodo local, entonces se establece uno u otro de los dos indicadores D (demote) o d (demote pending). Para evitar que se produzcan condiciones de inanición cuando hay contención en un determinado bloqueo, a cada bloqueo se le asigna un tiempo de retención mínimo. Un nodo que aún no ha tenido el bloqueo durante el tiempo mínimo de retención se le permite retener ese bloqueo hasta que el intervalo de tiempo haya expirado.

Si el intervalo de tiempo ha expirado, se activará la bandera D (demote) y se registrará el estado requerido. En ese caso, la próxima vez que no haya bloqueos concedidos en la cola de titulares, el bloqueo será degradado. Si el intervalo de tiempo no ha expirado, entonces se establece la bandera d

(demote pending) en su lugar. Esto también programa la máquina de estado para borrar d (demote pending) y establecer D (demote) cuando el tiempo mínimo de retención haya expirado.

La bandera I (inicial) se establece cuando a la glock se le ha asignado un bloqueo DLM. Esto sucede cuando la glock se utiliza por primera vez y la bandera I permanecerá entonces establecida hasta que la glock sea finalmente liberada (que el bloqueo DLM se desbloquee).

8.5. SOPORTES PARA GLOCK

enTabla 8.5, "Banderas de soporte de Glock" se muestran los significados de las diferentes banderas de los soportes de las glock.

Tabla 8.5. Banderas de soporte de Glock

Bandera	Nombre	Significado
a	Async	No espere el resultado de glock (hará una encuesta para el resultado más tarde)
A	Cualquier	Se acepta cualquier modo de bloqueo compatible
c	No hay caché	Cuando se desbloquea, baja el bloqueo DLM inmediatamente
e	No caduca	Ignorar las solicitudes de cancelación de bloqueo posteriores
E	Exactamente	Debe tener el modo de bloqueo exacto
F	Primero	Se establece cuando el titular es el primero en ser concedido para esta cerradura
H	Titular	Indica que se ha concedido el bloqueo solicitado
p	Prioridad	Colocar al titular de la cola en la cabeza de la cola
t	Prueba con	Una cerradura "try"
T	Prueba 1CB	Un bloqueo "try" que envía un callback
W	Espera	Se establece mientras se espera a que se complete la solicitud

Los indicadores de titular más importantes son H (titular) y W (espera), como se mencionó anteriormente, ya que se establecen en las solicitudes de bloqueo concedidas y en las solicitudes de bloqueo en cola, respectivamente. El orden de los titulares en la lista es importante. Si hay titulares concedidos, siempre estarán a la cabeza de la cola, seguidos de los titulares en cola.

Si no hay titulares concedidos, el primer titular de la lista será el que desencadene el siguiente cambio de estado. Dado que las solicitudes de baja se consideran siempre de mayor prioridad que las solicitudes del sistema de archivos, esto podría no siempre resultar directamente en un cambio de estado solicitado.

El subsistema glock soporta dos tipos de bloqueos "try". Estos son útiles tanto porque permiten la toma de bloqueos fuera del orden normal (con un retroceso y reintento adecuados) como porque pueden ser utilizados para ayudar a evitar recursos en uso por otros nodos. El bloqueo normal t (try) es justo lo que su nombre indica; es un bloqueo "try" que no hace nada especial. El bloqueo T (**try 1CB**), por otro lado, es idéntico al bloqueo t, excepto que el DLM enviará una única devolución de llamada a los actuales titulares de bloqueos incompatibles. Un uso del bloqueo T (**try 1CB**) es con los bloqueos **iopen**, que se utilizan para arbitrar entre los nodos cuando la cuenta **i_nlink** de un nodo es cero, y determinar cuál de los nodos será responsable de la desasignación del nodo. El glock de **iopen** se mantiene normalmente en estado compartido, pero cuando la cuenta de **i_nlink** llega a cero y se llama a →**evict_inode**(), solicitará un bloqueo exclusivo con T (**try 1CB**) establecido. Si el bloqueo es concedido, continuará con la asignación del inodo. Si el bloqueo no se concede, el nodo(s) que impedía(n) la concesión del bloqueo marcará(n) su(s) glock(s) con la bandera D (demote), que se comprueba en →**drop_inode**() para asegurar que la desasignación no se olvida.

Esto significa que los inodos que tienen un recuento de enlaces cero pero que todavía están abiertos serán desasignados por el nodo en el que se produce el **close**() final. Además, al mismo tiempo que el recuento de enlaces del nodo se reduce a cero, el nodo se marca como en el estado especial de tener un recuento de enlaces cero pero todavía en uso en el mapa de bits del grupo de recursos. Esto funciona como la lista de huérfanos del sistema de archivos ext3 en el sentido de que permite a cualquier lector posterior del mapa de bits saber que existe un espacio potencialmente recuperable, e intentar recuperarlo.

8.6. TRAPECIOS DE GLOCK

Los tracepoints también están diseñados para poder confirmar la corrección del control de la caché combinándolos con la salida de **blktrace** y con el conocimiento de la disposición en el disco. De este modo, es posible comprobar que cualquier E/S se ha emitido y completado bajo el bloqueo correcto, y que no hay carreras presentes.

El tracepoint **gfs2_glock_state_change** es el más importante para entender. Rastrea todos los cambios de estado del glock desde su creación inicial hasta el descenso final que termina con **gfs2_glock_put** y la transición final de NL a desbloqueado. La bandera l (bloqueada) de la glock siempre se establece antes de que se produzca un cambio de estado y no se borrará hasta después de que haya terminado. Durante un cambio de estado nunca hay titulares concedidos (la bandera de titular de glock H). Si hay titulares en cola, siempre estarán en el estado W (esperando). Cuando el cambio de estado se ha completado, los titulares pueden ser concedidos, que es la operación final antes de que la bandera l glock se borre.

El tracepoint **gfs2_demote_rq** lleva la cuenta de las peticiones de demote, tanto locales como remotas. Asumiendo que hay suficiente memoria en el nodo, las peticiones de demote locales raramente se verán, y la mayoría de las veces serán creadas por **umount** o por recuperaciones de memoria ocasionales. El número de peticiones de baja remotas es una medida de la contención entre nodos para un nodo o grupo de recursos en particular.

El tracepoint **gfs2_glock_lock_time** proporciona información sobre el tiempo que tardan las peticiones al DLM. La bandera de bloqueo (**b**) se introdujo en el glock específicamente para ser utilizada en combinación con este tracepoint.

Cuando a un titular se le concede un bloqueo, se llama a **gfs2_promote**, esto ocurre como las etapas finales de un cambio de estado o cuando se solicita un bloqueo que puede ser concedido inmediatamente debido a que el estado de glock ya tiene en caché un bloqueo de un modo adecuado. Si el titular es el primero en ser concedido para este glock, entonces la bandera f (primero) se establece en ese titular. En la actualidad, esto sólo lo utilizan los grupos de recursos.

8.7. BMAP TRACEPOINTS

El mapeo de bloques es una tarea fundamental para cualquier sistema de archivos. GFS2 utiliza un sistema tradicional basado en mapas de bits con dos bits por bloque. El objetivo principal de los tracepoints en este subsistema es permitir el control del tiempo que se tarda en asignar y mapear los bloques.

El tracepoint **gfs2_bmap** se llama dos veces para cada operación bmap: una vez al principio para mostrar la petición bmap, y otra al final para mostrar el resultado. Esto facilita el cotejo de las peticiones y los resultados y la medición del tiempo que se tarda en mapear bloques en diferentes partes del sistema de archivos, diferentes offsets de archivos, o incluso de diferentes archivos. También es posible ver cuáles son los tamaños medios de extensión que se devuelven en comparación con los solicitados.

El tracepoint **gfs2_rs** rastrea las reservas de bloques a medida que se crean, utilizan y destruyen en el asignador de bloques.

Para hacer un seguimiento de los bloques asignados, se llama a **gfs2_block_alloc** no sólo en las asignaciones, sino también en la liberación de bloques. Dado que todas las asignaciones están referenciadas según el inodo al que está destinado el bloque, esto puede utilizarse para rastrear qué bloques físicos pertenecen a qué archivos en un sistema de archivos activo. Esto es particularmente útil cuando se combina con **blktrace**, que mostrará patrones de E/S problemáticos que pueden ser referenciados de nuevo a los inodos relevantes usando el mapeo obtenido por medio de este tracepoint.

La E/S directa (**iomap**) es una política de caché alternativa que permite que las transferencias de datos de archivos se realicen directamente entre el disco y el buffer del usuario. Esto tiene ventajas en situaciones en las que se espera que la tasa de éxito de la caché sea baja. Tanto **gfs2_iomap_start** como **gfs2_iomap_end** trazan estas operaciones y pueden ser utilizadas para mantener un registro del mapeo que utiliza la E/S directa, las posiciones en el sistema de archivos de la E/S directa junto con el tipo de operación.

8.8. REGISTRO DE PUNTOS DE SEGUIMIENTO

Los puntos de seguimiento de este subsistema registran los bloques que se añaden y eliminan del diario (**gfs2_pin**), así como el tiempo que se tarda en consignar las transacciones en el registro (**gfs2_log_flush**). Esto puede ser muy útil cuando se trata de depurar los problemas de rendimiento del diario.

El tracepoint **gfs2_log_blocks** lleva la cuenta de los bloques reservados en el registro, lo que puede ayudar a mostrar si el registro es demasiado pequeño para la carga de trabajo, por ejemplo.

El tracepoint **gfs2_ail_flush** es similar al tracepoint **gfs2_log_flush** en el sentido de que mantiene un registro del inicio y el final de los vaciados de la lista AIL. La lista AIL contiene búferes que han pasado por el registro, pero que aún no han sido escritos de nuevo en su lugar y esto se vacía periódicamente con el fin de liberar más espacio de registro para su uso por el sistema de archivos, o cuando un proceso solicita un **sync** o **fsync**.

8.9. ESTADÍSTICAS DE GLOCK

GFS2 mantiene estadísticas que pueden ayudar a rastrear lo que está sucediendo dentro del sistema de archivos. Esto le permite detectar problemas de rendimiento.

GFS2 mantiene dos contadores:

- **dcount**, que cuenta el número de operaciones DLM solicitadas. Esto muestra cuántos datos han entrado en los cálculos de la media/varianza.
- **qcount**, que cuenta el número de operaciones de nivel **syscall** solicitadas. Generalmente **qcount** será igual o mayor que **dcount**.

Además, GFS2 mantiene tres pares de media/varianza. Los pares media/varianza son estimaciones exponenciales suavizadas y el algoritmo utilizado es el que se emplea para calcular los tiempos de ida y vuelta en el código de red.

Los pares de media y varianza mantenidos en GFS2 no están escalados, sino que están en unidades de nanosegundos enteros.

- srtt/srttvar: Tiempo de ida y vuelta suavizado para operaciones no bloqueantes
- srttb/srttvarb: Tiempo de ida y vuelta suavizado para operaciones de bloqueo
- irtt/irttvar: Tiempo entre peticiones (por ejemplo, tiempo entre peticiones DLM)

Una petición no bloqueante es aquella que se completará inmediatamente, sea cual sea el estado del bloqueo DLM en cuestión. Esto significa actualmente cualquier petición cuando (a) el estado actual del bloqueo es exclusivo (b) el estado solicitado es nulo o desbloqueado o (c) la bandera "try lock" está activada. Una solicitud de bloqueo cubre todas las demás solicitudes de bloqueo.

Los tiempos más grandes son mejores para los IRTT, mientras que los tiempos más pequeños son mejores para los RTT.

Las estadísticas se guardan en dos archivos **sysfs**:

- El archivo **glstats**. Este archivo es similar al archivo **glocks**, excepto que contiene estadísticas, con un glock por línea. Los datos se inicializan a partir de datos "por cpu" para el tipo de reloj para el que se crea el reloj (aparte de los contadores, que se ponen a cero). Este archivo puede ser muy grande.
- El archivo **lkstats**. Contiene estadísticas "por cpu" para cada tipo de glock. Contiene una estadística por línea, en la que cada columna es un núcleo de cpu. Hay ocho líneas por tipo de reloj, con los tipos que se suceden.

8.10. REFERENCIAS

Para más información sobre los tracepoints y el archivo GFS2 **glocks**, consulte los siguientes recursos:

- Para obtener información sobre las normas de bloqueo interno de las glock, consulte https://git.kernel.org/pub/scm/linux/kernel/git/torvalds/linux.git/tree/Documentation/filesystem glocks.rst.
- Para obtener información sobre el seguimiento de eventos, consulte https://git.kernel.org/pub/scm/linux/kernel/git/torvalds/linux.git/tree/Documentation/trace/eve

- Para obtener información sobre la utilidad **trace-cmd**, consulte http://lwn.net/Articles/341902/.

CAPÍTULO 9. SUPERVISIÓN Y ANÁLISIS DE SISTEMAS DE ARCHIVOS GFS2 MEDIANTE PERFORMANCE CO-PILOT (PCP)

Esta sección proporciona información sobre el uso de Performance Co-Pilot (PCP) para ayudar con la monitorización y para analizar los sistemas de archivos GFS2. La monitorización de los sistemas de archivos GFS2 en PCP es proporcionada por el módulo GFS2 PMDA en Red Hat Enterprise Linux que está disponible a través del paquete **pcp-pmda-gfs2**.

El PMDA de GFS2 proporciona una serie de métricas dadas por las estadísticas de GFS2 proporcionadas en el subsistema **debugfs**. Cuando se instala, el PMDA expone los valores dados en los archivos **glocks**, **glstats**, y **sbstats**. Estos informan de conjuntos de estadísticas sobre cada sistema de archivos GFS2 montado. El PMDA también hace uso de los tracepoints del kernel GFS2 expuestos por el Kernel Function Tracer (**ftrace**).

9.1. INSTALACIÓN DEL PMDA DE GFS2

Para funcionar correctamente, el PMDA GFS2 requiere que el sistema de archivos **debugfs** esté montado. Si el sistema de archivos **debugfs** no está montado, ejecute los siguientes comandos antes de instalar el GFS2 PMDA:

```
# mkdir /sys/kernel/debug
# mount -t debugfs none /sys/kernel/debug
```

El PMDA de GFS2 no está habilitado como parte de la instalación por defecto. Para hacer uso de la monitorización de métricas de GFS2 a través de PCP debe habilitarla después de la instalación.

Ejecute los siguientes comandos para instalar PCP y habilitar el PMDA de GFS2. Tenga en cuenta que el script de instalación de PMDA debe ejecutarse como root.

```
# yum install pcp pcp-pmda-gfs2
# cd /var/lib/pcp/pmdas/gfs2
# ./Install
Updating the Performance Metrics Name Space (PMNS) ...
Terminate PMDA if already installed ...
Updating the PMCD control file, and notifying PMCD ...
Check gfs2 metrics have appeared ... 346 metrics and 255 values
```

9.2. VISUALIZACIÓN DE INFORMACIÓN SOBRE LAS MÉTRICAS DE RENDIMIENTO DISPONIBLES CON LA HERRAMIENTA PMINFO

La herramienta **pminfo** muestra información sobre las métricas de rendimiento disponibles. Las siguientes secciones muestran ejemplos de diferentes métricas de GFS2 que puede mostrar con esta herramienta.

9.2.1. Examinar el número de estructuras glock que existen actualmente por sistema de archivos

Las métricas de glock de GFS2 ofrecen información sobre el número de estructuras de glock actualmente incrustadas para cada sistema de archivos GFS2 montado y sus estados de bloqueo. En GFS2, un glock es una estructura de datos que reúne el DLM y el almacenamiento en caché en una sola

máquina de estado. Cada glock tiene un mapeo 1:1 con un único bloqueo DLM y proporciona caché para los estados de bloqueo de manera que las operaciones repetitivas realizadas en un único nodo no tienen que llamar repetidamente al DLM, reduciendo el tráfico de red innecesario.

El siguiente comando **pminfo** muestra una lista del número de glocks por sistema de archivos GFS2 montado según su modo de bloqueo.

```
# pminfo -f gfs2.glocks

gfs2.glocks.total
    inst [0 or "afc_cluster:data"] value 43680
    inst [1 or "afc_cluster:bin"] value 2091

gfs2.glocks.shared
    inst [0 or "afc_cluster:data"] value 25
    inst [1 or "afc_cluster:bin"] value 25

gfs2.glocks.unlocked
    inst [0 or "afc_cluster:data"] value 43652
    inst [1 or "afc_cluster:bin"] value 2063

gfs2.glocks.deferred
    inst [0 or "afc_cluster:data"] value 0
    inst [1 or "afc_cluster:bin"] value 0

gfs2.glocks.exclusive
    inst [0 or "afc_cluster:data"] value 3
    inst [1 or "afc_cluster:bin"] value 3
```

9.2.2. Examinar el número de estructuras glock que existen por sistema de archivos por tipo

Las métricas glstats de GFS2 dan cuenta de cada tipo de glock que existe para cada sistema de archivos, un gran número de ellos será normalmente del tipo inodo (inodo y metadatos) o grupo de recursos (metadatos de grupo de recursos).

El siguiente comando **pminfo** muestra una lista del número de cada tipo de Glock por sistema de archivos GFS2 montado.

```
# pminfo -f gfs2.glstats

gfs2.glstats.total
    inst [0 or "afc_cluster:data"] value 43680
    inst [1 or "afc_cluster:bin"] value 2091

gfs2.glstats.trans
    inst [0 or "afc_cluster:data"] value 3
    inst [1 or "afc_cluster:bin"] value 3

gfs2.glstats.inode
    inst [0 or "afc_cluster:data"] value 17
    inst [1 or "afc_cluster:bin"] value 17

gfs2.glstats.rgrp
    inst [0 or "afc_cluster:data"] value 43642
```

```
    inst [1 or "afc_cluster:bin"] value 2053

gfs2.glstats.meta
    inst [0 or "afc_cluster:data"] value 1
    inst [1 or "afc_cluster:bin"] value 1

gfs2.glstats.iopen
    inst [0 or "afc_cluster:data"] value 16
    inst [1 or "afc_cluster:bin"] value 16

gfs2.glstats.flock
    inst [0 or "afc_cluster:data"] value 0
    inst [1 or "afc_cluster:bin"] value 0

gfs2.glstats.quota
    inst [0 or "afc_cluster:data"] value 0
    inst [1 or "afc_cluster:bin"] value 0

gfs2.glstats.journal
    inst [0 or "afc_cluster:data"] value 1
    inst [1 or "afc_cluster:bin"] value 1
```

9.2.3. Comprobación del número de estructuras glock que están en estado de espera

Las banderas más importantes del titular son H (titular: indica que el bloqueo solicitado está concedido) y W (espera: se establece mientras se espera a que la solicitud se complete). Estas banderas se establecen en las solicitudes de bloqueo concedidas y en las solicitudes de bloqueo en cola, respectivamente.

El siguiente comando **pminfo** muestra una lista del número de glocks con la bandera de soporte Wait (W) para cada sistema de archivos GFS2 montado.

```
# pminfo -f gfs2.holders.flags.wait

gfs2.holders.flags.wait
    inst [0 or "afc_cluster:data"] value 0
    inst [1 or "afc_cluster:bin"] value 0
```

Si ves un número de solicitudes en espera en el bloqueo de un grupo de recursos, puede haber varias razones para ello. Una de ellas es que haya un gran número de nodos en comparación con el número de grupos de recursos en el sistema de archivos. Otra es que el sistema de archivos puede estar casi lleno (lo que requiere, por término medio, más tiempo de búsqueda de bloques libres). La situación en ambos casos puede mejorarse añadiendo más almacenamiento y utilizando el comando **gfs2_grow** para ampliar el sistema de archivos.

9.2.4. Comprobación de la latencia de las operaciones del sistema de archivos mediante las métricas basadas en los puntos de seguimiento del núcleo

El PMDA de GFS2 soporta la recolección de métricas de los tracepoints del kernel de GFS2. Por defecto, la lectura de estas métricas está desactivada. La activación de estas métricas enciende los tracepoints del kernel GFS2 cuando se recopilan las métricas con el fin de rellenar los valores de la métrica. Esto podría tener un pequeño efecto en el rendimiento cuando estas métricas de Kernel Tracepoint están activadas.

PCP proporciona la herramienta **pmstore**, que permite modificar la configuración de PMDA basándose en los valores de las métricas. Las métricas de **gfs2.control.*** permiten alternar los tracepoints del kernel GFS2. El siguiente ejemplo utiliza el comando **pmstore** para habilitar todos los tracepoints del kernel GFS2.

```
# pmstore gfs2.control.tracepoints.all 1
gfs2.control.tracepoints.all old value=0 new value=1
```

Cuando se ejecuta este comando, el PMDA activa todos los tracepoints GFS2 en el sistema de archivos **debugfs**. Tabla 9.1, “Lista completa de métricas” explica cada uno de los tracepoints de control y su uso, Una explicación sobre el efecto de cada tracepoint de control y sus opciones disponibles también está disponible a través del interruptor de ayuda en **pminfo**.

La métrica de promoción de GFS2 cuenta el número de solicitudes de promoción en el sistema de archivos. Estas solicitudes se separan por el número de solicitudes que se han producido en el primer intento y "otros" que se conceden después de su solicitud de promoción inicial. Un descenso en el número de promociones a la primera con un aumento de "otras" promociones puede indicar problemas de contención de archivos.

La métrica de solicitudes de descenso de GFS2, al igual que la métrica de solicitudes de ascenso, cuenta el número de solicitudes de descenso que se producen en el sistema de archivos. Sin embargo, éstas también se dividen entre las solicitudes que provienen del nodo actual y las que provienen de otros nodos del sistema. Un gran número de peticiones de demote de nodos remotos puede indicar contención entre dos nodos para un determinado grupo de recursos.

La herramienta **pminfo** muestra información sobre las métricas de rendimiento disponibles. Este procedimiento muestra una lista del número de glocks con el indicador Wait (W) holder para cada sistema de archivos GFS2 montado. El siguiente comando **pminfo** muestra una lista del número de glocks con la bandera Wait (W) holder para cada sistema de archivos GFS2 montado.

```
# pminfo -f gfs2.latency.grant.all gfs2.latency.demote.all

gfs2.latency.grant.all
    inst [0 or "afc_cluster:data"] value 0
    inst [1 or "afc_cluster:bin"] value 0

gfs2.latency.demote.all
    inst [0 or "afc_cluster:data"] value 0
    inst [1 or "afc_cluster:bin"] value 0
```

Es una buena idea determinar los valores generales observados cuando la carga de trabajo se ejecuta sin problemas para poder notar cambios en el rendimiento cuando estos valores difieren de su rango normal.

Por ejemplo, podría notar un cambio en el número de solicitudes de promoción que esperan a completarse en lugar de completarse en el primer intento, lo que la salida del siguiente comando le permitiría determinar.

```
# pminfo -f gfs2.latency.grant.all gfs2.latency.demote.all

gfs2.tracepoints.promote.other.null_lock
     inst [0 or "afc_cluster:data"] value 0
     inst [1 or "afc_cluster:bin"] value 0

gfs2.tracepoints.promote.other.concurrent_read
```

```
    inst [0 or "afc_cluster:data"] value 0
    inst [1 or "afc_cluster:bin"] value 0

gfs2.tracepoints.promote.other.concurrent_write
    inst [0 or "afc_cluster:data"] value 0
    inst [1 or "afc_cluster:bin"] value 0

gfs2.tracepoints.promote.other.protected_read
    inst [0 or "afc_cluster:data"] value 0
    inst [1 or "afc_cluster:bin"] value 0

gfs2.tracepoints.promote.other.protected_write
    inst [0 or "afc_cluster:data"] value 0
    inst [1 or "afc_cluster:bin"] value 0

gfs2.tracepoints.promote.other.exclusive
    inst [0 or "afc_cluster:data"] value 0
    inst [1 or "afc_cluster:bin"] value 0
```

La salida del siguiente comando le permitiría determinar un gran aumento en las solicitudes remotas de degradación (especialmente si provienen de otros nodos del clúster).

```
# pminfo -f gfs2.tracepoints.demote_rq.requested

gfs2.tracepoints.demote_rq.requested.remote
    inst [0 or "afc_cluster:data"] value 0
    inst [1 or "afc_cluster:bin"] value 0

gfs2.tracepoints.demote_rq.requested.local
    inst [0 or "afc_cluster:data"] value 0
    inst [1 or "afc_cluster:bin"] value 0
```

La salida del siguiente comando podría indicar un aumento inexplicable de las descargas de registros.

```
# pminfo -f gfs2.tracepoints.log_flush.total

gfs2.tracepoints.log_flush.total
    inst [0 or "afc_cluster:data"] value 0
    inst [1 or "afc_cluster:bin"] value 0
```

9.3. LISTA COMPLETA DE MÉTRICAS DISPONIBLES PARA GFS2 EN PCP

Tabla 9.1, “Lista completa de métricas” describe la lista completa de métricas de rendimiento que ofrece el paquete **pcp-pmda-gfs2** para los sistemas de archivos GFS2.

Tabla 9.1. Lista completa de métricas

Nombre de la métrica	Descripción

Nombre de la métrica	Descripción
gfs2.glocks.*	Métricas relativas a la información recogida del archivo de estadísticas de glock (**glocks**) que cuentan el número de glocks en cada estado que existe actualmente para cada sistema de archivos GFS2 montado actualmente en el sistema.
gfs2.glocks.flags.*	Rango de métricas que cuentan el número de glocks que existen con las banderas de glocks dadas
gfs2.holders.*	Métricas relativas a la información recopilada del archivo de estadísticas de glock (**glocks**) que cuenta el número de glocks con titulares en cada estado de bloqueo que existe actualmente para cada sistema de archivos GFS2 montado actualmente en el sistema.
gfs2.holders.flags.*	Gama de métricas que cuentan el número de soportes de glocks con las banderas de soporte dadas
gfs2.sbstats.*	Métricas de tiempo relativas a la información recogida del archivo de estadísticas de superbloques (**sbstats**) para cada sistema de archivos GFS2 montado actualmente en el sistema.
gfs2.glstats.*	Métrica relativa a la información recogida del archivo de estadísticas de glock (**glstats**) que cuenta el número de cada tipo de glock que existe actualmente para cada sistema de archivos GFS2 montado actualmente en el sistema.
gfs2.latency.grant.*	Una métrica derivada que hace uso de los datos de los tracepoints **gfs2_glock_queue** y **gfs2_glock_state_change** para calcular una latencia media en microsegundos para que las solicitudes de concesión de glock se completen para cada sistema de archivos montado. Esta métrica es útil para descubrir posibles ralentizaciones en el sistema de archivos cuando la latencia de concesión aumenta.

Nombre de la métrica	Descripción
gfs2.latency.demote.*	Una métrica derivada que hace uso de los datos de los tracepoints **gfs2_glock_state_change** y **gfs2_demote_rq** para calcular una latencia media en microsegundos para que se completen las peticiones de demote de glock para cada sistema de archivos montado. Esta métrica es útil para descubrir posibles ralentizaciones en el sistema de archivos cuando aumenta la latencia de destitución.
gfs2.latency.queue.*	Una métrica derivada que hace uso de los datos del tracepoint **gfs2_glock_queue** para calcular una latencia media en microsegundos para que se completen las peticiones de la cola glock para cada sistema de archivos montado.
gfs2.worst_glock.*	Una métrica derivada que hace uso de los datos del tracepoint **gfs2_glock_lock_time** para calcular un "peor bloqueo actual" percibido para cada sistema de archivos montado. Esta métrica es útil para descubrir la posible contención de bloqueos y la ralentización del sistema de archivos si el mismo bloqueo se sugiere varias veces.
gfs2.tracepoints.*	Métricas relativas a la salida de los tracepoints GFS2 **debugfs** para cada sistema de archivos montado actualmente en el sistema. Cada subtipo de estas métricas (una de cada tracepoint GFS2) puede controlarse individualmente si se activa o desactiva mediante las métricas de control.
gfs2.control.*	Métricas de configuración que se utilizan para activar o desactivar el registro de métricas en el PMDA. Las métricas de control se activan mediante la herramienta **pmstore**.

9.4. REALIZACIÓN DE UNA CONFIGURACIÓN MÍNIMA DE PCP PARA RECOPILAR DATOS DEL SISTEMA DE ARCHIVOS

El siguiente procedimiento describe las instrucciones sobre cómo instalar una configuración mínima de PCP para recopilar estadísticas en Red Hat Enterprise Linux. Esta configuración implica añadir el número mínimo de paquetes en un sistema de producción necesario para recopilar datos para su posterior análisis.

El archivo **tar.gz** resultante de la salida de **pmlogger** puede analizarse utilizando otras herramientas de PCP y puede compararse con otras fuentes de información sobre el rendimiento.

1. Instale los paquetes PCP necesarios.

```
# yum install pcp pcp-pmda-gfs2
```

2. Activar el módulo GFS2 para PCP.
   # cd /var/lib/pcp/pmdas/gfs2 # ./Install

3. Inicie los servicios **pmcd** y **pmlogger**.
   # systemctl start pmcd.service # systemctl start pmlogger.service

4. Realiza operaciones en el sistema de archivos GFS2.

5. Detenga los servicios **pmcd** y **pmlogger**.
   # systemctl stop pmcd.service # systemctl stop pmlogger.service

6. Recoge la salida y la guarda en un archivo **tar.gz** cuyo nombre se basa en el nombre del host y la fecha y hora actuales.
   # cd /var/log/pcp/pmlogger # tar -czf $(hostname).$(date+%F-%Hh%M).pcp.tar.gz $(hostname)

9.5. REFERENCIAS

- Para obtener más información sobre la supervisión del rendimiento en GFS2 en general, consulte Depuración de sistemas de archivos GFS2 con tracepoints GFS2 y el archivo debugfs glocks.

- Para más información general sobre el uso de Performance Co-Pilot para supervisar el rendimiento del sistema, consulte Supervisión del rendimiento con Performance Co-Pilot en el Portal del cliente de Red Hat.

- Para obtener información general sobre Performance Co-Pilot en Red Hat Enterprise Linux, consulte el Índice de artículos, soluciones, tutoriales y libros blancos de Performance Co-Pilot(PCP) en el Portal del Cliente de Red Hat.