

Sistema de ficheros de nueva generación para GNU/Linux

1. ¿Que es Btrfs?

- **Sistema de ficheros basado en “copy on write”**
- **Diseñado para reemplazar a los sistemas de ficheros ext***
- **Suple las principales carencias de los FS UNIX originales**

2. Historia

- **Sistema de ficheros creado por Oracle en 2007**
- **Disponible en el kernel Linux desde la versión 2.6.29**
- **Actualmente en desarrollo**

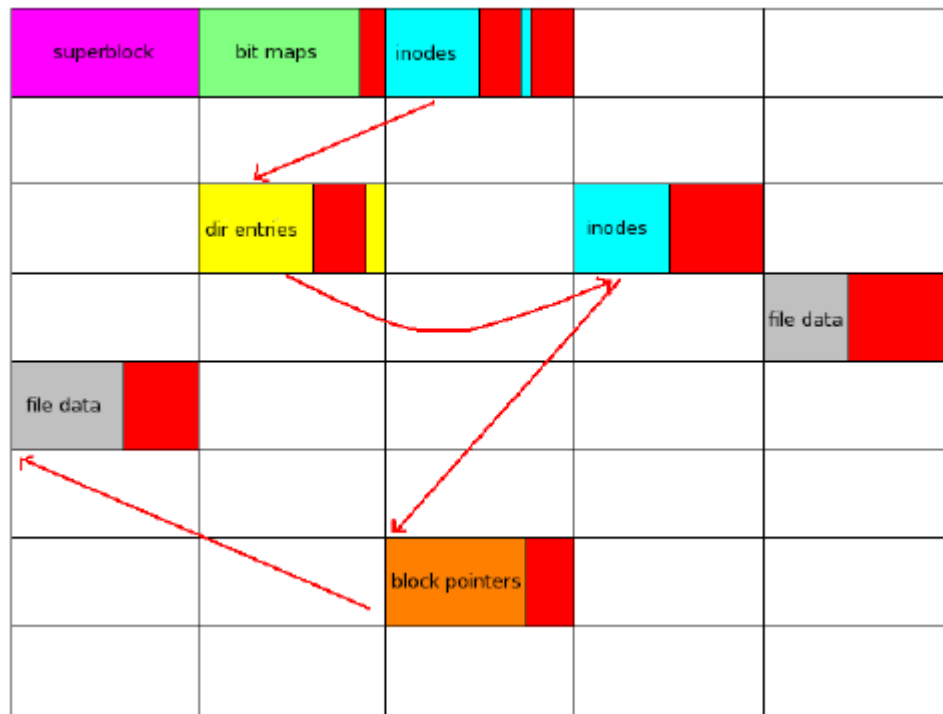
3. Principales características

- **Adición y eliminación de dispositivos en línea.**
- **Crecimiento y reducción de volúmenes en línea.**
- **Compresión transparente.**
- **Instantáneas y clonación.**
- **Desfragmentación en línea.**
- **Balanceo en línea.**
- **Soporte para RAID**
- **Conversión instantánea desde sistemas de ficheros ext3 y ext4.**
- **Opciones especiales para dispositivos SSD (Discos de Estado Sólido).**

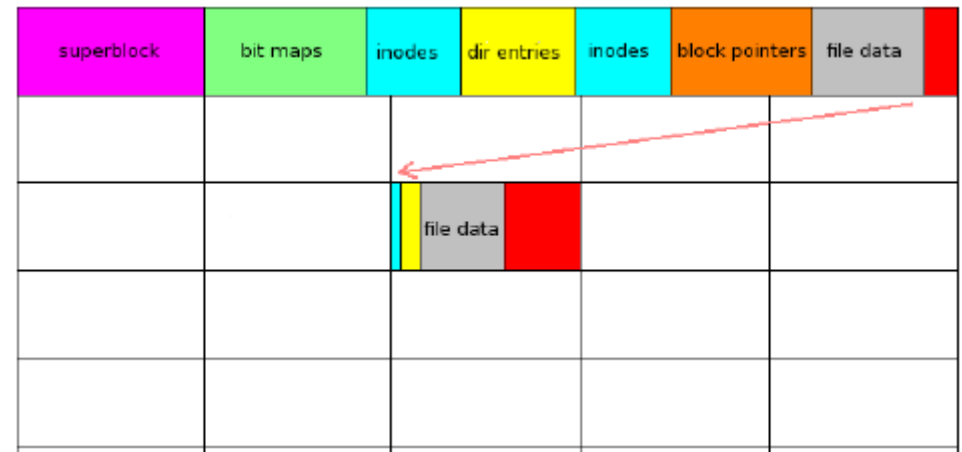
4. Organización de los ficheros en Btrfs

- **Cambios respecto a los FS UNIX tradicionales**
 - La tabla de inodes se estructura en forma de árbol
 - Si el directorio es pequeño, se incluye dentro de la hoja del inode en el árbol
 - Los datos del fichero se guardan dentro del mismo inode, en una estructura única dentro del mismo bloque
- **Ventajas:**
 - Menor uso de metadatos → menor espacio desperdiciado
 - Acceso mas rápido: se realiza una única búsqueda

4. Organización de los ficheros: Tradicional vs Btrfs



Tradicional

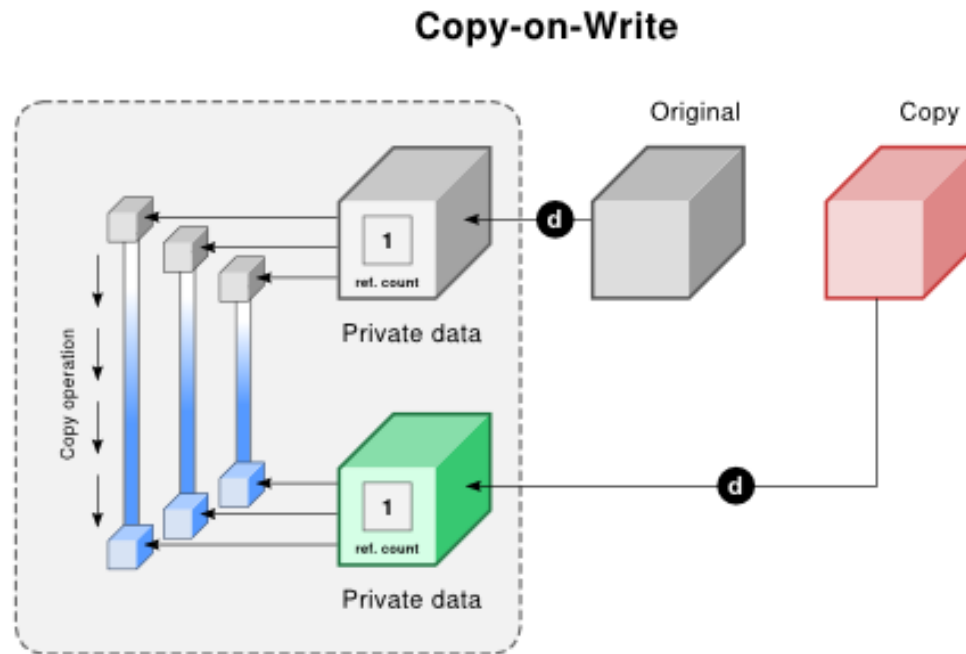


Btrfs

5. Características de Btrfs: Copy on Write

- **Permite evitar inconsistencias cuando varios recursos intentan acceder al mismo fichero**
 - Mientras los recursos leen, todos comparten el mismo puntero
 - Cuando un recurso comienza a escribir en el fichero:
 - Se genera una copia del fichero
 - El resto de recursos pasan a leer desde esa copia, en lugar del fichero original
- **El proceso es completamente transparente para los recursos**
 - Todos los recursos creen estar accediendo al mismo fichero

5. Características de Btrfs: Copy on Write

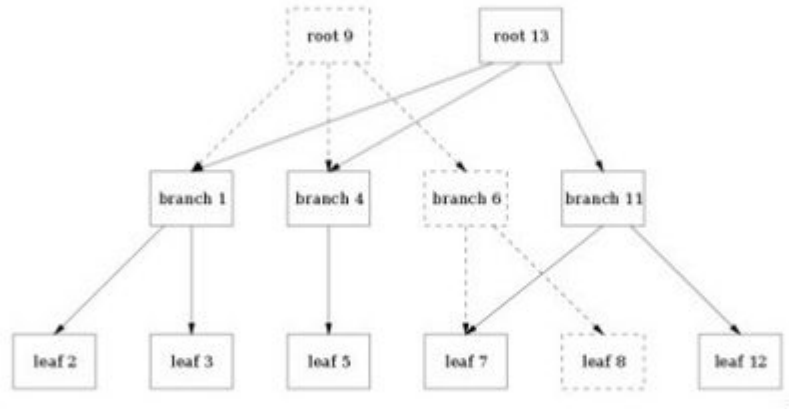


Esquema de funcionamiento de “Copy on Write”

5. Características de Btrfs: Copy on Write

BTRFS: Copy on Write

- how the append-only btree works
-



Funcionamiento de “Copy on Write” en Btrfs

6. Características de Btrfs: Volúmenes lógicos

- **Características:**

- Permiten crear un árbol de ficheros independiente del dispositivo de bloques
- Pueden tener propietarios distintos a root
- Pueden tener una cuota de bloques
- El máximo de subvolúmenes admitidos por el FS es 2^{64}

- **Diferencias respecto a LVM o ZFS:**

- Cada volumen y subvolumen tiene su propia raíz
- Los volúmenes no están representados como dispositivos de bloques
- No pueden contener otros sistemas de ficheros

6. Características de Btrfs: Volúmenes lógicos

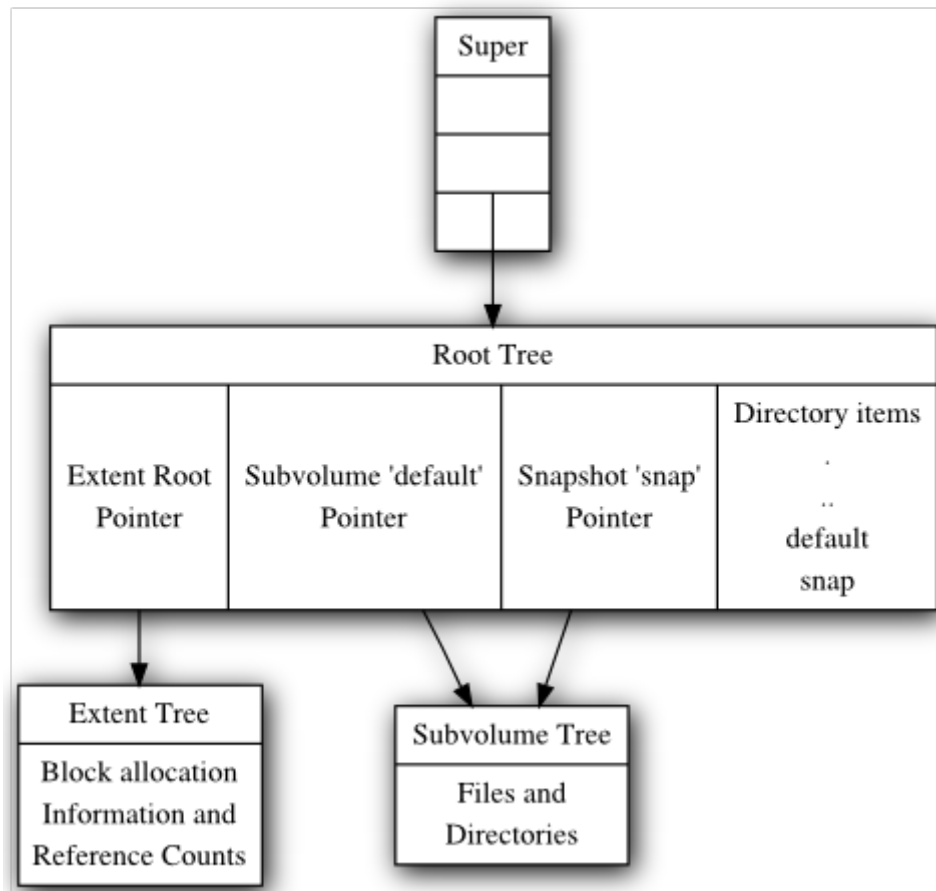
```
toplevel          (volume root directory)
+-- dir_1         (normal directory)
|  +-- file_2     (normal file)
|  |-- file_3     (normal file)
|-- subvol_a      (subvolume root directory)
   +-- subvol_b   (subvolume root directory, nested below subvol_a)
       |-- file_4  (normal file)
       |-- file_5  (normal file)
```

Ejemplo de estructura de volúmenes lógicos en Btrfs

7. Características de Btrfs: instantáneas

- **Replican el estado de un subvolumen en un momento determinado**
- **Basadas en subvolúmenes**
 - Subvolumen cuyo bloque raíz es inicialmente compartido con otro
 - Cuando se toma una instantánea, el número de referencias en el bloque raíz es incrementado
 - Los cambios hechos en la instantánea son ocultos a la raíz del subvolumen original
- **Son actualizables**
 - No hay límite en el número de actualizaciones de una instantánea
 - En caso de necesitar una instantánea de solo-lectura, se le puede aplicar una cuota al bloque raíz

8. Resumen: Estructura de ficheros en Btrfs



Organización de volúmenes y subvolúmenes en Btrfs

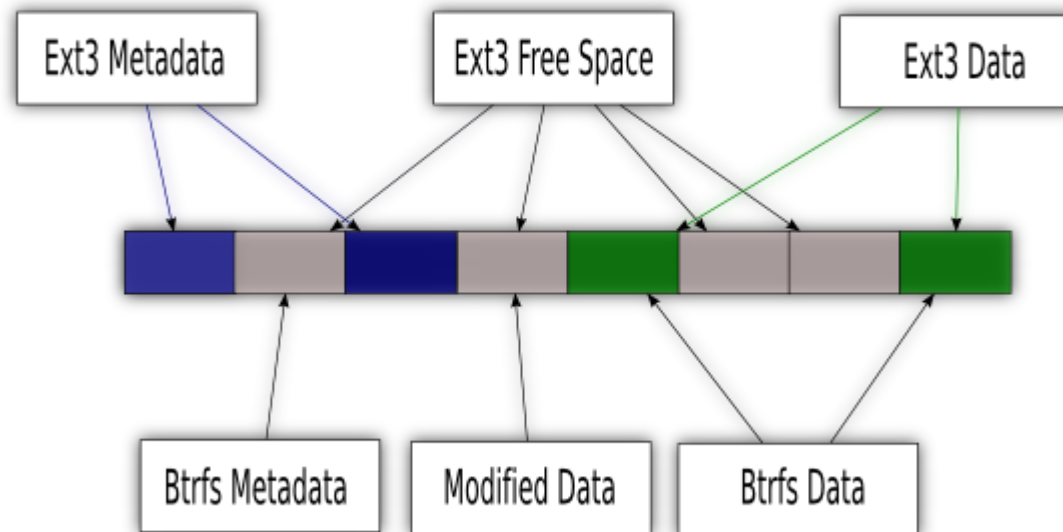
9. Características de Btrfs: RAID

- **Btrfs incluye herramientas para crear RAID de forma nativa en el sistema de ficheros**
 - En el momento de creación del sistema de ficheros con `mkfs`, el usuario puede indicar si va a utilizar un único dispositivo de bloques, o se desea configurar un RAID software
 - Los dispositivos se pueden añadir y eliminar “en línea”
- **Permite convertir un sistema de ficheros no-RAID en un RAID software, en dos simples pasos:**
 - Añadir un nuevo dispositivo
 - Activar el balanceador del nivel RAID deseado

10. Migración a Btrfs desde ext3/4

- **La herramienta `btrfs-convert` nos permite migrar un sistema de ficheros ext3/4 a btrfs, sin perder los datos**
- **La conversión mantiene la versión original de los bloques de datos de ficheros de ext3/4**
 - El programa de conversión crea una instantánea con todas las referencias a bloques usados por ext3/4
 - Si no se elimina esa instantánea, Btrfs solo utilizará los bloques marcados como libres en ext3/4
 - De esta forma, el proceso de conversión es reversible

10. Migración a Btrfs desde ext3/4



Sistema de ficheros ext3 convertido para funcionar con Btrfs

11. Conclusiones

- **Btrfs elimina la necesidad de herramientas externas para usos avanzados**
 - RAID, volúmenes lógicos: todo se incluye en el mismo sistema de ficheros
- **Sistema de ficheros ideado para ser rápido y flexible**
 - El uso de volúmenes lógicos como estructura principal aporta mucha versatilidad
 - Las mejoras en el acceso a ficheros permiten explotar al máximo la capacidad del dispositivo y la velocidad del mismo
- **La migración desde ext3/4 es bastante sencilla**
 - No se producen pérdidas de datos
 - El proceso es reversible

Referencias

- <https://es.wikipedia.org/wiki/Btrfs>
- <https://lwn.net/Articles/342892/>
- <https://www.quora.com/What-are-some-examples-from-Linux-kernel-source-implementing-copy-on-write-feature>
- <https://btrfs.wiki.kernel.org/index.php/SysadminGuide#Subvolumes>
- https://btrfs.wiki.kernel.org/index.php/Btrfs_design#Snapshots_and_Subvolumes
- <http://apimadrid.org/nibbler/documentos/articulos/pdf/btrfs.pdf>
- https://btrfs.wiki.kernel.org/index.php/Using_Btrfs_with_Multiple_Devices
- https://btrfs.wiki.kernel.org/index.php/Conversion_from_Ext3