

Benchmark

Sobre Sistema de Archivos Ext3

Integrantes del grupo

Germán Hüttemann
Fernando Mancía
Cristhian Parra
Marcelo Rodas

1. Subsistema

Sistemas de archivos Ext3 para archivos del Sistema Operativo Linux.

2. Medidas

2.1. Descripción de Métricas medibles

Se ha optado en este trabajo, por un conjunto de métricas sencillas de medir, y que al mismo tiempo influyan de manera significativa en las operaciones más importantes del sistema de archivos.

A continuación se mencionan las métricas que efectivamente proponemos para la evaluación del sistema seleccionado según el criterio mencionado más arriba:

1. Cantidad de lecturas y escrituras por segundo: Consiste en la cantidad de operaciones de lectura y escritura realizadas por segundo. Se espera que las operaciones por segundo aumenten, ya que esto implicaría que una mayor cantidad de operaciones puedan ser realizadas en una misma unidad de tiempo.
2. Cantidad de bytes leídos y escritos por segundo: Consiste en la suma de los tamaños de los archivos leídos o escritos por segundo. Se espera que la cantidad de bytes procesados por segundo aumente, ya que esto implica que se tiene un mayor throughput de datos de entrada o de salida.
3. Latencia de lecturas y escrituras: Consiste en el tiempo que se tarda en realizar una lectura o una escritura de un archivo. Se espera que esta latencia disminuya, realizando operaciones de lectura o escritura de forma más rápida.
4. Latencia de creación y borrado de directorio: Consiste en el tiempo que se tarda en realizar la creación y borrado de un directorio.
5. Porcentaje de fragmentación del sistema de archivos: Consiste en el porcentaje de fragmentación causado por la administración de archivos del filesystem, debido a que se manejan archivos más grandes del tamaño de bloques y se guarda un archivo de forma fragmentada, es decir en más de un bloque de datos no contiguos. Mientras exista menor fragmentación, más rápido será el acceso al archivo.

2.2. Métricas para la evaluación del sistema

1. **BLs**: Cantidad de bytes leídos por segundo.
2. **BEs**: Cantidad de bytes escritos por segundo.
3. **LL**: Latencia de lectura.
4. **LE**: Latencia de escritura.
5. **tCD**: Tiempo de creación de directorios.
6. **tBD**: Tiempo de borrado de directorios.
7. **PF**: Porcentaje de fragmentación del sistema de archivos.

2.3. Monitorización de Métricas

Todas las métricas propuestas fueron monitoreadas por el propio benchmark. Para las métricas 1 y 2, lo que se realizó fue controlar los tiempos de lecturas y escrituras de archivos y controlar cuantos bytes se van escribiendo en todas las operaciones, luego se divide los bytes totales por el tiempo total de las operaciones. Para las métricas 3, 4, 5 y 6, lo que se realizará es calcular el tiempo que tarda en realizar las operaciones de lectura, escritura, creación de directorios y borrado de directorio respectivamente, luego se divide el tiempo calculado por la cantidad de operaciones correspondientes (o sea, se prácticamente se promedia la latencia de una operación).

En cuanto a la séptima métrica, encontramos que datos de fragmentación podrían ser obtenidos con los comandos `e2fsck`, `fsck`, sin embargo, finalmente optamos por implementar nuestra propia lógica de cálculo de la fragmentación

3. Parámetros

El sistema de archivos Ext3, es una extensión sistema Ext2, al cual agrega las siguientes características:

- Un sistema de Journaling.
- Crecimiento en line del sistema.
- Una versión especializada de BTrees para el manejo de directorios grandes, conocido como HTree.

En particular, el Journaling consiste en tener un registro de transacciones realizadas por el sistema de archivos, resultando en una masiva reducción del tiempo de recuperación del sistema de archivos, luego del apagado repentino del mismo.

3.1. Parámetros generales

Tamaño del bloque

- Descripción: El tamaño de bloque del sistema de archivos no es un parámetro variable online, ya que se determina al crear el mismo, sin embargo, creemos que será importante realizar pruebas sobre sistemas de archivos con características similares, pero con distintos tamaños de bloque.
- Influencia sobre el Rendimiento: Este parámetro tiene influencia en el tamaño máximo de los archivos que se guardan en el filesystem y también en el tamaño máximo del sistema de archivos. Sin embargo, estas dos limitaciones no son las que nos interesa estudiar, sino más bien nos interesa como influye este parámetro en la administración de los metadatos. Más precisamente, diferentes tamaños de bloque tendrán influencia en la manera en que Ext3 busca los bloques asociados a un archivo. La estructura de inodos de Ext3, contiene un array de 15 punteros de 32 bits. Los 12 primeros direccionan lo primeros doce bloques de un archivo y los 3 siguientes se utilizan para direccionar “bloques de indirección” que permiten direccionar los siguientes bloques del archivo. Si el tamaño de bloques es por ejemplo 4 KB, los primeros doce punteros permitirá direccionar hasta 48 KB de un archivo. Un tamaño menor, podría influir seriamente en el rendimiento, ya que se forzaría a buscar en los bloques de indirección constantemente, pero sería muy apropiado si a priori supiéramos que el sistema de archivos solo manejará archivos pequeños.
- Carga: La carga asociada a la prueba que determine la influencia de este parámetro debe ser una carga, fundamentalmente, de un alto porcentaje de lecturas, que nos permitan observar

como evoluciona el sistema cuando se consultan archivos grandes y se deben utilizar los bloques de indirección.

- **Valores:** El tamaño de bloques, junto con las limitaciones que este impone sobre el tamaño máximo de archivos y del filesystem se presentan en la siguiente tabla:

Tamaño de Bloque	Máximo Tamaño de Archivo	Máximo tamaño del filesystem
1 KB	16 GB	2 TB
2 KB	256 GB	8 TB
4 KB	2 TB	16 TB
8 KB*	2 TB	32 TB

Tabla 1. Tamaños posibles de Bloque para sistemas de archivos ext3

*Este último caso solo es posible para arquitecturas que soporten páginas de 8 KB. Probaremos los tres primeros valores.

3.2. Parámetros específicos de Ext3

Nivel de journaling

- **Descripción:** Ext3 se define como un “Journaled File System”, lo que significa que puede registrar los cambios a un “journal” (usualmente un archivo circular en un área dedicada) antes de hacer commit de los cambios al archivo real.
De esta manera, el sistema se vuelve menos propenso a volverse corrupto cuando hay alguna falla de alimentación, o el sistema se cuelga por algún motivo.
- **Influencia en el Rendimiento:** Este parámetro tiene una influencia directa en el rendimiento del sistema de archivos, porque de acuerdo al nivel de Journaling que se tenga, el costo de cada escritura en el disco podría multiplicarse por 2, ya que primero los cambios se colocan en el Journal y luego se confirman.
- **Valores:** El Journaling se realiza en tres niveles, lo cual se puede configurar al montar el sistema de archivos ya sea agregando la opción `data=<nivel>` en `/etc/fstab`, o mediante el comando `tune2fs`:

```
# tune2fs -o has_journal -o journal_data /dev/hdXY
```

 - **Journal:** En este nivel, tanto los metadatos como el contenido de los archivos se escriben en el journal antes de ser confirmados. En general, el rendimiento debería bajar debido a que los datos se escriben dos veces. Sin embargo, es el nivel más seguro que permite una rápida y confiable recuperación después de un crash.
 - **Ordered:** En este caso, solo los metadatos se escriben en el Journal. Se garantiza sin embargo que el contenido de los archivos se escribe antes que sus metadatos asociados sean marcados como confirmados en el journal. Suele ser el valor por defecto en las diferentes distribuciones de Linux.
 - **Writeback:** Como en el caso anterior, los metadatos se escriben en el journal. La diferencia está en que el contenido puede escribirse al archivo antes o después que el journal sea actualizado.

No se habla de la posibilidad de desactivar completamente el Journal porque en ese caso estaríamos bajando a la versión del sistema de archivos Ext2.

- **Carga:** Para notar en detalle como influye el journaling en el sistema de archivos, aumentaremos la cantidad de escrituras realizadas que forzarán al sistema a escribir constantemente en el journal.

Sync Interval

- **Descripción:** Se refiere al intervalo de tiempo entre cada sincronización del journal con los archivos reales. Se puede cambiar al montar el sistema de archivos utilizando `-o`

commit=Nsec

- Influencia sobre el Rendimiento: A mayores valores de tiempo, se mejora el rendimiento porque se reduce la sobrecarga de operaciones del sistema de archivos, sin embargo, se pierde en confiabilidad ante fallos del sistema.
- Valores: El valor por defecto es 5 segundos. Variaremos el parámetro con los valores 5 y 10, para ver su influencia.
- Carga: Alto porcentaje de escrituras sobre el filesystem.

Eliminar la actualización del access time para los archivos

- Descripción: Cada vez que un archivo es accedido, el sistema de archivos actualiza el atributo de tiempo de acceso del mismo. Esta actualización puede ser desactivada. Se puede manipular editando /etc/fstab con lo siguiente:
/dev/hda1 / ext3 defaults, errors=remount-ro, **noatime**, auto, rw, dev, exec, suid, nouser, data=writeback 0 1
- Influencia sobre el Rendimiento: La influencia en el rendimiento se da sobre todo porque se disminuye la sobrecarga de trabajo del filesystem. Es una buena opción para web servers, news servers u otros sistemas con altas tasas de acceso a archivos.
- Valores: Activado/desactivado. Realizaremos pruebas con ambas configuraciones.
- Carga: Alto porcentaje de lecturas sobre el filesystem.

3.3. Descripción de Parámetros para las Pruebas

1. Tamaño del Bloque (pTB):

Tiene influencia fundamentalmente en las lecturas. Cómo no se puede variar online, se probará con el valor máximo (4Kb) y con el valor mínimo (1Kb).

2. Nivel de journaling (pNJ):

Influye sobre todo en las escrituras. Se trata del nivel con el que las transacciones del filesystem serán protegidas para la recuperación, a través del archivo de journal.

Se probarán dos modos:

- El modo por defecto (ordered)
- El modo optimizado (writeback)

Para cambiar este parámetro, se necesita remontar el sistema de archivos:

```
# mount -t ext3 -o data=<modo> /dev/sda3 /vol01
```

También se puede cambiar el tipo de montaje por defecto:

```
# tune2fs -O has_journal -o journal_data /dev/hdXY
```

1. Sync Interval (pSI):

Intervalo de tiempo entre cada sincronización del journal con los archivos reales. Se puede cambiar al montar el sistema de archivos utilizando -o commit=Nsec.

A mayores valores de tiempo, se mejora el rendimiento porque se reduce la sobrecarga de operaciones del sistema de archivos, sin embargo, se pierde en confiabilidad ante fallos del sistema. Influye en operaciones de escritura.

Al montar:

```
# mount -t ext3 -o commit=Nsec /dev/sda2 /vol02
```

1. Eliminar la actualización del access time (pAT):

Influye en lecturas. Consiste en la actualización de este atributo ante cada acceso al archivo.

```
Se puede manipular editando /etc/fstab con lo siguiente:  
/dev/hda1 / ext3 defaults, errors=remount-ro, noatime, auto, rw, dev,  
exec, suid, nouser, data=writeback 0 1  
O al montar:  
mount -t ext3 -o noatime /dev/sda2 /vol02
```

4. Operaciones Básicas

Todo el sistema se sostiene sobre las siguientes operaciones primitivas sobre el sistema de archivos. De esta manera, el sistema permanece bien modularizado y eso lo suficientemente extensible como para generar nuevas versiones con mayor cantidad de pruebas.

1. Crear Archivo. **(CA)**
2. Leer Archivo. **(LA)**
 - Secuencialmente.
 - Aleatoriamente.
3. Escribir Archivo. **(EA)**
 - Secuencialmente.
 - Aleatoriamente.
4. Borrar Archivo. **(BA)**
5. Crear Directorio. **(CD)**
6. Borrar Directorio. **(BD)**

Obs: Todas las operaciones básicas, deben ser implementadas en *primitivas.c*

5. Operaciones de stress sobre el sistema de Archivos.

En base a las operaciones básicas, y teniendo en cuenta las métricas que deseamos medir, se realizó un plan de pruebas que pudiera poner bajo stress al sistema, de manera tal a sacar de cada conjunto de operaciones, las medidas adecuadas sobre las métricas de la forma más exacta posible. A partir de esto, se configuró el siguiente plan de prueba, el cual finalmente se ejecutó sobre el filesystem de prueba.

1. **c1000A:** Crear 1000 archivos para pruebas de re-lectura.
2. **l1000A:** Leer secuencialmente los 1000 archivos creados.
3. **rl1000A:** Releer secuencialmente los 1000 archivos creados.
4. **c2000A2m:** Crear 2000 archivos de 2 MB para accesos secuenciales y escrituras
5. **re2000A2m:** Reescribir aleatoriamente en los 2000 archivos.
6. **INSeq:** Leer los N archivos creados anteriormente, de forma secuencial.
7. **INAleat:** Leer los N archivos creados anteriormente, de forma aleatoria.
8. **cNDirs:** Crear 100000 directorios.
9. **bNDirs:** Borrar los 10000 directorios.
10. **c1000Avac:** Crear 1000 archivos vacíos.¹
11. **e1000A600b:** Escribir 600 bytes en cada archivo (repetir 1000 veces).¹
12. **t1000Amitad:** Truncar cada archivo a la mitad de su longitud.¹
13. **a1000A600b:** Agregar 600 bytes en cada archivo (repetir 1000 veces).¹

¹ Esta operación se realizará aparte exclusivamente para medir la fragmentación.

Observación: Los códigos asociados a cada uno de los elementos de más arriba, fueron asignados para su fácil asociación en los cuadros y explicaciones posteriores.

6. Relaciones de las operaciones con las Métricas y Parámetros

La siguiente tabla resume todas nuestras operaciones, y su relación con los parámetros y métricas que se midieron.

También se aclara, por cada Operación, a que grupo de de operaciones implementadas corresponde (Mix de Impl.). Estos grupos, constituyen una clasificación de operaciones orientadas a poner bajo stress cierto conjunto de métricas con mayor énfasis.

Operaciones	Operaciones Básicas	Métricas	Parámetros	Mix de Impl.
c1000A	CA, EA	BEs, LE	pTB, pNJ, pSI	mLect
l1000A	LA (S)	BLs, LL	pTB, pAT	mLect
rl1000A	LA (S)	BLs, LL	pTB, pAT	mLect
c2000A2m	CA, EA	BEs, LE	pTB, pNJ, pSI	mEscr
re2000A2m	EA (A)	BEs, LE	pTB, pNJ, pSI	mEscr
lNSeq	LA (S)	BLs, LL	pTB, pAT	mLect2
lNAleat	LA (A)	BLs, LL	pTB, pAT	mLect2
cNDirs	CD	tCD	pTB, pNJ, pSI	mDirs
bNDirs	BD	tBD	pTB, pNJ, pSI	mDirs
c1000Avac	--	PF	pTB	mFrag
e1000A600b	--	PF	pTB	mFrag
t1000Amitad	--	PF	pTB	mFrag
a1000A600b	--	PF	pTB	mFrag

Tabla 2. Tabla de Resumen de Operaciones, métricas y parámetros, de la forma en que están relacionados.

- “A” significa aleatorio.
- “S” significa secuencial.
- "Gn" son los grupos según los parámetros que influyen. Cada G incluye el mismo grupo de parámetros, y al mismo tiempo, incluye solo operaciones de lectura, escritura o la parte para fragmentación.
- En las últimas operaciones no se consideran operaciones básicas ya que lo que se quiere inducir es solamente el aumento de la fragmentación y no otras métricas.
- Los "Mixs" de implementación, permiten agrupar las operaciones que vamos a implementar según lo que en general se está probando en ese Mix. Así tenemos:
 1. **mLect:** 1^{ra} Mezcla de operaciones de lectura.
 2. **mEscr:** Mezcla de operaciones de escritura.
 3. **mLect2:** 2^{da} Mezcla de operaciones de lectura.
 4. **mDirs:** Mezcla de operaciones sobre Directorios.
 5. **mFrag:** Mezcla de operaciones para probar fragmentación.

7. Los Escenarios

Básicamente, se configuraron 2 escenarios de pruebas, para 2 sistemas de archivos con tamaños de bloque distinto. Esto significa en total, 4 escenarios, que se listan a continuación.

<i>Escenario</i>	<i>Tamaño de Bloque</i>	<i>Journal</i>	<i>Commit</i>	<i>Access time</i>
<i>Est4k</i>	4096 bytes	ordered	5 segs	Si
<i>Opt4k</i>	4096 bytes	writeback	10 segs	No
<i>Est1k</i>	1024 bytes	ordered	5 segs	Si
<i>Opt1k</i>	1024 bytes	writeback	10 segs	No

Tabla 3. Escenarios de Prueba utilizados.

Estos escenarios de prueba, se implementaron sobre una Laptop **Acer Aspire 5610**, procesador **Intel Core Duo T2050 de 1.6 Ghz.**, memoria de **1,5G** y un disco duro de **120 Gb 5400rpm**.

7.1. Particiones

Para implementar los escenarios de prueba, se construyeron dos particiones de prueba en el disco duro, cada uno de 6Gb.:

Particion 1:

```
# mke2fs -b 1024 -L vol101 -j
Cómo sería la Partición:
[root@localhost ~]# mke2fs -b 1024 -j -L vol102 -n /dev/sda3
mke2fs 1.40.8 (13-Mar-2008)
Warning: 256-byte inodes not usable on older systems
Etiqueta del sistema de ficheros=vol101
Tipo de SO: Linux
Tamaño del bloque=1024 (bitácora=0)
Tamaño del fragmento=1024 (bitácora=0)
655360 nodos i, 10482412 bloques
524120 bloques (5.00%) reservados para el superusuario
Primer bloque de datos=1
Máximo de bloques del sistema de ficheros=77594624
1280 bloque de grupos
8192 bloques por grupo, 8192 fragmentos por grupo
512 nodos i por grupo
Respaldo del superbloque guardado en los bloques:
      8193, 24577, 40961, 57345, 73729, 204801, 221185,
401409, 663553, 1024001, 1990657, 2809857, 5120001,
5971969
```

Particion 2:

```
# mke2fs -b 4096 -L vol102 -j
Cómo sería la partición:
[root@localhost ~]# mke2fs -b 4096 -L vol102 -j -n /dev/sda2
mke2fs 1.40.8 (13-Mar-2008)
Warning: 256-byte inodes not usable on older systems
Etiqueta del sistema de ficheros=vol102
Tipo de SO: Linux
Tamaño del bloque=4096 (bitácora=2)
Tamaño del fragmento=4096 (bitácora=2)
655360 nodos i, 2620603 bloques
131030 bloques (5.00%) reservados para el superusuario
Primer bloque de datos=0
Máximo de bloques del sistema de ficheros=2684354560
80 bloque de grupos
```

32768 bloques por grupo, 32768 fragmentos por grupo
8192 nodos i por grupo
Respaldo del superbloque guardado en los bloques:
32768, 98304, 163840, 229376, 294912, 819200,
884736, 1605632

7.2. Opciones de montaje:

Configuración Estándar:

```
$ mount -t ext3 /dev/sda2 /vol02
```

Por defecto, se usan las opciones journal=ordered, commit=5.

Configuración Optimizada:

```
$ mount -t ext3 -o data=writeback,commit=10,noatime /dev/sda2 /vol02
```

8. Resultados

En esta sección, presentamos un resumen de las tablas de resultados obtenidos en las pruebas que realizamos. Las tablas corresponden a la última prueba realizada, luego de una serie de ajustes que se realizaron en el código de benchmarking, para mejorar la confiabilidad de los resultados.

Las tablas muestran los resultados obtenidos por cada Operación realizada, sobre las métricas que se pretendían probar con esa operación. Para cada partición, se cuenta con la tabla de resultados de la configuración estándar vs. La tabla de resultados con la configuración optimizada.

En la sección de Conclusiones se analizan los resultados presentados en esta sección. Además se presenta en esta sección, como dato complementario, la salida del programa luego de la ejecución del benchmark.

8.1. Resultados en la partición con block size igual 1024.

testId	BEs	BLs	LE	LL	tCD	tBD	PF-%	PF-cant
1k – standard								
c1000A	6.432.968,9660	0,0000	0,0796					
l1000A		62.309.845,4424		0,0082				
rl1000A		62.226.543,5100		0,0082				
c2000A2m	6.099.308,9483		0,3279					
re2000A2m	184.212,4199		0,0278					
INSeq		17.426.460,3374		0,1148				
INAlcat		659.963,9082		0,0078				
cNDir					0,0035			
bNDir						0,0005		
mFrag							99,1540	1.000,0000

Tabla 4. Resultados de la configuración estándar en la partición con block_size 1K

Tabla 5. Resultados de la configuración optimizada en la partición con block_size 1K

testId	BEs	BLs	LE	LL	tCD	tBD	PF-%	PF-cant
1k – optimizado								
c1000A	7.446.514,5367		0,0688					
l1000A		64.370.128,2374		0,0080				
rl1000A		62.822.085,8896		0,0082				
c2000A2m	7.186.799,2871		0,2783					
re2000A2m	189.391,1371		0,0270					
INSeq		18.231.540,5652		0,1097				
INAleat		365.140,4935		0,0140				
cNDir					0,0038			
bNDir						0,0004		

testId	BEs	BLs	LE	LL	tCD	tBD	PF-%	PF-cant
Resultados								
c1000A	15,76%		115,76%					
l1000A		3,31%		103,31%				
rl1000A		0,96%		100,96%				
c2000A2m	17,83%		117,83%					
re2000A2m	2,81%		102,81%					
INSeq		4,62%		104,62%				
INAleat		-44,67%		55,33%				
cNDir					92,42%			
bNDir						122,28%		

Tabla 6. Comparación de resultados entre la configuración estándar y la optimizada de la partición con block_size 1k. Los porcentajes presentados en esta tabla indican la aceleración en el rendimiento sobre las métricas correspondientes para cada operación realizada.

Salidas de Programa.

Salida de Programa para la configuración estándar de la partición de block size = 1K:

```
Directorio de trabajo configurando en '/mnt/test/'...
Operaciones de Lectura/Re-lectura Secuencial...
  Creando 1000 arch de 512000 bytes. c1000A...79590 ms.
  Lectura de 1000 arch. de 512000 bytes. l1000A...8217 ms.
  Re-lectura de 1000 arch. de 512000 bytes. rl1000A...8228 ms.
Operaciones de Escritura Secuencial/Aleatoria...
  Creacion de 500 arch. de 2000000 bytes. rl1000A...163953 ms.
  Escritura aleatoria de 500 arch. de 2000000 bytes. rl1000A...13897 ms.
Operaciones de Lectura Secuencial/Aleatoria...
  Lectura secuencial de 500 arch de 5120 bytes...57384 ms.
  Lectura aleatoria de 500 arch de 5120 bytes...3879 ms.
Operaciones de Creacion/Borrado de Directorios...
  Creando 50000 directorios...177311 ms.
  Borrando 50000 directorios...24150 ms.
Operaciones de Fragmentacion...
  Creando 1000 archivos vacios...581 ms.
  Escribiendo 600 bytes en cada archivo (repetir 1000 veces)...409892 ms.
  Truncando cada archivo a la mitad de su longitud...291 ms.
  Agregando 600 bytes en cada archivo (repetir 1000 veces)...336119 ms.
Fin del test... 1283733 ms.
```

Salida de Programa para la configuración estándar de la partición de block size = 1K:

```
Directorio de trabajo configurando en '/mnt/test/'...
Operaciones de Lectura/Re-lectura Secuencial...
    Creando 1000 arch de 512000 bytes. c1000A...68757 ms.
    Lectura de 1000 arch. de 512000 bytes. l1000A...7954 ms.
    Re-lectura de 1000 arch. de 512000 bytes. rl1000A...8150 ms.
Operaciones de Escritura Secuencial/Aleatoria...
    Creacion de 500 arch. de 2000000 bytes. rl1000A...139144 ms.
    Escritura aleatoria de 500 arch. de 2000000 bytes. rl1000A...13517 ms.
Operaciones de Lectura Secuencial/Aleatoria...
    Lectura secuencial de 500 arch de 5120 bytes...54850 ms.
    Lectura aleatoria de 500 arch de 5120 bytes...7011 ms.
Operaciones de Creacion/Borrado de Directorios...
    Creando 50000 directorios...191870 ms.
    Borrando 50000 directorios...19751 ms.
Operaciones de Fragmentacion...
    Creando 1000 archivos vacios...495 ms.
    Escribiendo 600 bytes en cada archivo (repetir 1000 veces)...399917 ms.
    Truncando cada archivo a la mitad de su longitud...287 ms.
    Agregando 600 bytes en cada archivo (repetir 1000 veces)...502694 ms.
Fin del test... 1414736 ms.
```

8.2. Resultados en la partición con block size igual 4096.

testId	BEs	BLs	LE	LL	tCD	tBD	PF-%	PF-cant
4k – standard								
c1000A	9.299.622,2028		0,0551					
l1000A		63.342.818,2605		0,0081				
rl1000A		62.806.673,2090		0,0082				
c2000A2m	9.670.712,2480		0,2068					
re2000A2m	544.564,9862		0,0094					
INSeq		33.598.763,5655		0,0595				
INAleat		2.031.746,0317		0,0025				
cNDir					0,0007			
bNDir						0,0003		

Tabla 7. Resultados de la configuración estándar en la partición con block_size 4K

testId	BEs	BLs	LE	LL	tCD	tBD	PF-%	PF-cant
4k – optimizado								
c1000A	12.066.933,7733		0,0424					
l1000A		62.968.884,5161		0,0081				
rl1000A		62.568.740,0709		0,0082				
c2000A2m	12.338.824,1101		0,1621					
re2000A2m	537.250,7870		0,0095					
INSeq		38.848.529,5832		0,0515				
INAleat		688.542,2270		0,0074				
cNDir					0,0008			
bNDir						0,0003		

Tabla 8. Resultados de la configuración optimizada en la partición con block_size 4K

Tabla 9. Comparación de resultados entre la configuración estándar y la optimizada de la partición con block_size 4k. Los porcentajes presentados en esta tabla indican la aceleración en el rendimiento sobre las métricas correspondientes para cada operación realizada.

testId	BEs	BLs	LE	LL	tCD	tBD	PF-%	PF-cant
Resultados								
c1000A	29,76%		129,76%					
l1000A		-0,59%		99,41%				
rl1000A		-0,38%		99,62%				
c2000A2m	27,59%		127,59%					
re2000A2m	-1,34%		98,66%					
INSeq		15,62%		115,62%				
INAleat		-66,11%		33,89%				
cNDir					89,29%			
bNDir						108,17%		

Salidas de Programa.

Salida de Programa para la configuración estándar de la partición de block size = 4K:

```

Directorio de trabajo configurando en '/home/cparra/ext3bench/'...
Operaciones de Lectura/Re-lectura Secuencial...
    Creando 1000 arch de 512000 bytes. c1000A...55056 ms.
    Lectura de 1000 arch. de 512000 bytes. l1000A...8083 ms.
    Re-lectura de 1000 arch. de 512000 bytes. rl1000A...8152 ms.
Operaciones de Escritura Secuencial/Aleatoria...
    Creacion de 500 arch. de 2000000 bytes. rl1000A...103405 ms.
    Escritura aleatoria de 500 arch. de 2000000 bytes. rl1000A...4701 ms.
Operaciones de Lectura Secuencial/Aleatoria...
    Lectura secuencial de 500 arch de 5120 bytes...29763 ms.
    Lectura aleatoria de 500 arch de 5120 bytes...1260 ms.
Operaciones de Creacion/Borrado de Directorios...
    Creando 50000 directorios...34980 ms.
    Borrando 50000 directorios...16562 ms.
Operaciones de Fragmentacion...
    Creando 1000 archivos vacios...418 ms.
    Escribiendo 600 bytes en cada archivo (repetir 1000 veces)...412462 ms.
    Truncando cada archivo a la mitad de su longitud...297 ms.
    Agregando 600 bytes en cada archivo (repetir 1000 veces)...293146 ms.
Fin del test... 968500 ms.
```

Salida de Programa para la configuración estándar de la partición de block size = 4K:

```

Directorio de trabajo configurando en '/home/cparra/ext3bench/'...
Operaciones de Lectura/Re-lectura Secuencial...
    Creando 1000 arch de 512000 bytes. c1000A...42430 ms.
    Lectura de 1000 arch. de 512000 bytes. l1000A...8131 ms.
    Re-lectura de 1000 arch. de 512000 bytes. rl1000A...8183 ms.
Operaciones de Escritura Secuencial/Aleatoria...
    Creacion de 500 arch. de 2000000 bytes. rl1000A...81045 ms.
    Escritura aleatoria de 500 arch. de 2000000 bytes. rl1000A...4765 ms.
Operaciones de Lectura Secuencial/Aleatoria...
    Lectura secuencial de 500 arch de 5120 bytes...25741 ms.
    Lectura aleatoria de 500 arch de 5120 bytes...3718 ms.
Operaciones de Creacion/Borrado de Directorios...
    Creando 50000 directorios...39175 ms.
    Borrando 50000 directorios...15300 ms.
Operaciones de Fragmentacion...
    Creando 1000 archivos vacios...442 ms.
    Escribiendo 600 bytes en cada archivo (repetir 1000 veces)...394364 ms.
    Truncando cada archivo a la mitad de su longitud...289 ms.
    Agregando 600 bytes en cada archivo (repetir 1000 veces)...272434 ms.
Fin del test... 896278 ms.
```

9. Conclusiones

1. Las escrituras registran las mejoras previstas, ya que nuestras optimizaciones están pensadas fundamentalmente para reducir la sobrecarga de las operaciones de escritura, en las que se obtuvieron muy buenas ganancias es en la escritura secuencial de archivos, siendo estas de al menos un 15% para bloques de 1k y, de al menos un 27% para bloques de 4k. Se puede notar aquí que influye el tamaño de bloque al permitir mayores ganancias con mayores tamaños de bloques.
2. Por otro lado, la optimización orientada a las lecturas no permite obtener los resultados esperados. De hecho en cuanto a las lecturas, se registraron extrañas variabilidades en las relaciones entre los valores optimizados y los que no lo han sido.
Una razón probable por la cual la eliminación del *access time* no produce mejoras considerables, es que este parámetro es utilizado por el mecanismo de caching del sistema de archivo, y al eliminar su actualización, el algoritmo que decide que páginas mantener en la cache de acuerdo a cuales se acceden con mayor frecuencia, podría estar fallando y produciendo una cierta aleatoriedad en los resultados.
En general, las lecturas no mejoraron más de 5%, y cuando empeoraron no lo hicieron más de 0,6%. Sin embargo, hubieron picos, sobre todo en la lectura secuencial, en los que alcanzamos mejoras de 15%. Todo esto puede deberse a la influencia del algoritmo que controla el caching de las páginas de disco.
También respecto a la lectura, donde no se registró ninguna mejora, y muy por el contrario se produjeron grandes pérdidas, es en la lectura aleatoria. Suponemos que esto se debe fundamentalmente a la influencia que tiene sobre esta operación la constante recolocación (*seek*) del puntero de posición del fichero.
3. Nuestras actualizaciones no registraron mejoras en el tiempo de creación de directorios, aunque la pérdida de rendimiento es apenas del orden del 10%.
Sí se puede notar que el tiempo de creación de directorios es menor en el filesystem cuyo tamaño de bloque es menor. Esto era un resultado esperado, ya que al tener menor tamaño de bloque, los metadatos a escribir por cada creación de directorio son menos, ya que se reducen también el tamaño de los i-nodos.
También sucede que los archivos de directorio en sí, ocupan menos espacio y por lo tanto, se pueden notar mejorías en el borrado. De hecho, en el borrado de directorios, se pueden ver mejoras en las configuraciones optimizadas en relación a la estándar, lo que se puede deber fundamentalmente a las modificaciones en el nivel de *journaling*.
Hay más mejoras en cuanto a los directorios en sistemas de archivos con tamaños de bloque menor. También creemos que la variación del parámetro *dir index*, no utilizado en este trabajo, podría haber tenido algún efecto, negativo o positivo, sobre los resultados en cuanto a directorios se refiere.
4. En cuanto a la fragmentación, en realidad lo que se midió fue el porcentaje de utilización del disco en bloques. Obviamente, y como era de esperarse, este valor es mayor para sistemas de archivo con menor tamaño de bloque.
Sin embargo, la mejora obtenida no es tan significativa como podría esperarse. Esto posiblemente se deba a que los archivos creados para esta prueba fueron todos del mismo tamaño. Tal vez convendría utilizar archivos de tamaño aleatorio sobre un rango y dividir la prueba en dos, una para archivos pequeños y otra para archivos grandes, de manera a poder comparar mejor los resultados obtenidos.

10. Referencias.

- [1] <http://linuxplanet.com/linuxplanet/reports/4136/1/>. Datos introductorios sobre ext3
- [2] <http://wiki.archlinux.org/index.php/Ext3>. Filesystem_Tips Wiki con Tips para tunear ext3
- [3] <http://www.zip.com.au/~akpm/linux/ext3/ext3-usage.html>. Using the ext3 filesystem in 2.4 kernels
- [4] <http://en.wikipedia.org/wiki/Ext3>. Ext3 en Wikipedia.
- [5] <http://batleth.sapient-sat.org/projects/FAQs/ext3-faq.html>. Ext3 FAQ.
- [6] http://homepage.smc.edu/morgan_david/cs40/analyze-ext2.htm. Análisis de ext2
- [7] http://en.wikipedia.org/wiki/Journaling_file_system. Journaling File System
- [8] <http://www.redhat.com/archives/ext3-users/2007-May/msg00009.html>. Tips para tunear ext3
- [9] http://nakedape.cc/wiki/PlatformNotes_2fLinuxNotes#head-cc9fb9851ff9d491b28aba29e978fdd1e0d82271.
Notas varias sobre el sistema de archivos Ext3
- [10] <http://www.comentaomueve.com/2007/07/09/recopilacion-de-consejos-para-optimizar-ubuntu/>. Más notas
varias sobre ext3
- [11] <http://www.ibm.com/developerworks/linux/library/l-fs8.html#4>. Algunos datos interesantes