Sistemas Operativos

FIEC, ESPOL, Dra. Cristina Abad R.

Proyecto Parcial

Resumen del proyecto

Título: Evaluación de una caché de metadatos para sistemas de archivos

Estudiantes: 1 ó 2 por proyecto

Puntaje: 30 puntos (30% de la nota del parcial)

Fecha de entrega: Martes 9 de septiembre, 15h30

Fecha de sustentación: Martes 9 de septiembre, 15h30

Lenguajes de programación permitidos:
Java, C, C++, python

Sistemas operativos permitidos:
Linux o Windows

Interfaz gráfica: Ninguna

Descripción y alcance del proyecto

Antecedentes

Los sistemas de archivos (por ejemplo, ext3) almacenan los metadatos asociados a un archivo de manera independiente de los datos o contenido del archivo. Estos metadatos incluyen el nombre del archivo, su longitud, y su ubicación en la jerarquía de archivos (ej.: `/home/user/cabad/varios.txt`).

Dichos metadatos son almacenados en disco, en archivos con una estructura propietaria o en una base de datos ligera. Sin embargo, consultar esta base de datos cada vez que un usuario quiere abrir un archivo resulta costoso debido a lo lento que es leer del disco duro.

Por esta razón, los sistemas de archivos modernos, utilizan cachés en memoria que contienen información parcial de los metadatos. Cuando un usuario desea abrir un archivo, primero se consulta los metadatos en la caché, y solamente si no se encuentra la información en la misma (cache miss), entonces se consulta los metadatos en disco.

Descripción del problema

El sistema de archivos ficticio OperativosFS actualmente almacena todos los metadatos de la estructura jerárquica de directorios en memoria. Sus diseñadores están considerando implementar una caché para dicha información, y desean evaluar el rendimiento de cuatro políticas de desalojo de caché (cache eviction policies): FIFO, LRU, LFU y una seleccionada por Ud. (por ejemplo, ver políticas descritas en: http://en.wikipedia.org/wiki/Cache_algorithms y http://en.wikipedia.org/wiki/Page_replacement_algorithm).

Alcance

Para evaluar las cuatro políticas, a Ud. le han encargado que implemente un programa que pueda ser ejecutado en la línea de comando, y que dado un archivo con un workload específico, calcule la tasa de fallos de la cache (cache miss rate).

Los parámetros de la línea de comando deben recibirse en el siguiente orden:

```
<workload_file> <policy> <cache size (number of entries)> <extra parameters may go here>
```

Por ejemplo, si evaluamos la política LRU con una caché de 50000 elementos, el resultado debe ser algo así:

```
$java cacheSimulator workload.txt LRU 50000
Evaluando una caché LRU con 50000 entradas.
Resultados:
      Miss rate:               x.x% (W misses out of Q references)
      Miss rate (warm cache):  y.y% (M misses out of Q-50000 references)
```

Dataset (workload)

Como no tenemos un dataset o trace de un workload de usuarios de OperativosFS, vamos a usar traces de la Wikipedia para evaluar nuestra caché. Para esto, deben bajarse el primer dataset disponible en: http://www.wikibench.eu/wiki/2007-10/ y procesarlo de tal manera que seleccionen solamente los archivos del servidor `upload.wikimedia.org`, y obtengan la ruta de cada uno de dichos archivos. Si están trabajando en un computador con Linux, puede usar los siguientes comandos para obtener dicho dataset:

```
wget http://www.wikibench.eu/wiki/2007-10/wiki.1191201596.gz
gunzip wiki.1191201596.gz
cut -f3 -d' ' wiki.1191201596 | grep upload.wikimedia.org | cut -c28- > workload.txt
```

Dicho dataset representará un workload como el siguiente (NOTA: la siguiente es la salida de `head workload.txt`):

```
/wikipedia/commons/thumb/e/e0/Icono_aviso_borrar.png/50px-Icono_aviso_borrar.png
/wikipedia/commons/thumb/4/44/North.svg/17px-North.svg.png
/wikipedia/commons/thumb/d/d6/Wikiquote-logo-en.svg/49px-Wikiquote-logo-en.svg.png
/wikipedia/commons/thumb/b/bc/Flag_of_Grenada.svg/20px-Flag_of_Grenada.svg.png
/wikipedia/en/e/e3/Wikiversity-logo-41px.png
/wikipedia/en/6/67/Sixthfleet.gif
/wikipedia/en/thumb/4/4a/Commons-logo.svg/50px-Commons-logo.svg.png
/math/8/2/a/82af55fa8a0ab107af1ffa03e4852b28.png
/wikipedia/commons/thumb/2/27/Flag_of_Moldova.svg/22px-Flag_of_Moldova.svg.png
/wikipedia/en/1/18/Monobook-bullet.png
```

Para que confirmen que tienen el número de líneas correcto: `wc -l workload.txt` debe darles como resultado 3721736 líneas de texto.

Resultados esperados

Ud. debe realizar un análisis de rendimiento de cada uno de las políticas de desalojo de caché implementadas, y graficar sus resultados de tal manera que podamos comparar cuál política nos da mejores resultados. Un gráfico típico para evaluar cachés es graficar el tamaño de la caché (eje de las X) versus el miss rate de la misma (eje de las y). Por ejemplo: https://upload.wikimedia.org/wikipedia/commons/thumb/0/07/Cache%2Cmissrate.png/400px-Cache%2Cmissrate.png.OJO: esa es una caché diferente, para un workload diferente, así que su gráfico deberá tener otros valores en el eje de las X y de las Y, pero lo importante es que se pueda ver cómo el miss rate disminuye a medida que nuestra caché es más grande.

Entregables

1. Documento **impreso**, a ser entregado durante la sustentación del proyecto. Secciones del documento: Introducción, metodología (incluyendo una descripción de las estructuras de datos utilizadas por su programa), resultados, conclusiones.
2. Código fuente, archivos ejecutables y screenshots de cómo correr el cliente y servidor; entregar vía SidWeb. Ya deben haber sido subidos a SidWeb antes de la sustentación. No acepto código en pen drives, CD o vía email.

Rúbrica

Total: 30 puntos

1. Documento: 8 puntos
    a. Sección de metodología describe y justifica correctamente cada una de las estructuras de datos utilizadas en cada uno de los 4 algoritmos: 4 puntos
    b. Resultados: gráfico correctamente presentado (total: 4 puntos)
        i. Suficientes datapoints: 2 puntos
        ii. Rango de ejes de las X y de las Y bien seleccionados: 1 punto

iii. Sección de Resultados y Conclusiones concuerdan con resultados representados gráficamente y demuestran entendimiento de los mismos: 1 punto

2. Implementación: 16 puntos
    a. Algoritmo de desalojo produce la cantidad correcta de fallos (cache misses): 4 puntos por cada algoritmo (12 puntos en total)
    b. Algoritmo adicional correctamente implementado y su selección justificada formalmente: 4 puntos
3. Eje de rendimiento (performance): 6 puntos
    a. Correré en mi servidor Linux sus programas, con la siguiente configuración (donde política es FIFO, luego LRU y luego LFU):

    ```
    time su_programa workload.txt <política> 600000
    ```

    Y registraré los tiempos que les toma a cada uno ejecutar esta prueba. Luego, para cada grupo obtendré el tiempo promedio de las tres corridas (una para cada política) y compararé dicho tiempo promedio con el de cada uno de los demás grupos. El puntaje a ser asignado será el siguiente:
        - Grupo más rápido: 6 puntos + 5 puntos extra
        - Cualquier grupo cuyo rendimiento está una desviación estándar (o más) por sobre el promedio de los tiempos de todos los grupos: 6 puntos.
        - Cualquier grupo cuyo rendimiento está en el rango (media-std.dev., media+std.dev.): 4 puntos
        - Cualquier grupo con un rendimiento en el rango (media- 2*std.dev., media - std. dev]: 2 puntos
        - Cualquier grupo con un rendimiento de (-infinito, media - 2*std.dev]: 0 puntos

    NOTA: para estas pruebas, si desean que corra su programa con alguna optimización en la línea de comando (por ejemplo, más memoria u algún tipo en particular del recolector de basura en Java), deben indicarlo en el README y utilizaré dicha configuración en mis pruebas.
4. Sustentación: 10 puntos
    a. Estudiante demuestra haber participado activamente en el desarrollo del proyecto y demuestra entender su implementación y resultados: 10 puntos
    b. Estudiante demuestra haber participado parcialmente el desarrollo del proyecto y demuestra entender parcialmente su implementación y resultados: 7.5 puntos
    c. Estudiante demuestra participación limitada en el proyecto y/o no logra comprender los resultados obtenidos: 5 puntos
    d. Estudiante demuestra una colaboración mínima en el desarrollo del proyecto: 2.5 puntos
    e. Estudiante no colaboró en el desarrollo del proyecto: 0 puntos
5. Nota Final: (Nota documento + nota implementación + nota de rendimiento) * (Nota sustentación)/10