

66:20 Organización de Computadoras

Trabajo práctico 2: Arquitecturas de memoria

Fecha de presentación 05/11/2009

1. Objetivos

Familiarizarse con la medición de comportamiento utilizando herramientas de profiling y con la estimación de parámetros de la cache a partir de patrones de acceso.

2. Alcance

Este trabajo práctico es de elaboración grupal, evaluación individual, y de carácter obligatorio para todos alumnos del curso.

3. Recursos

Para la realización de este trabajo práctico, se requiere la herramienta de debugging y profiling Valgrind [1], la cual puede conseguirse en <http://valgrind.org/>.

Esta herramienta es de fácil instalación, y corre en ambientes Linux-x86. En particular, se utilizará el módulo Cachegrind para simulación de memorias caché.

En <http://valgrind.org/docs/manual/index.html> podrán hallar abundante documentación.

4. Software a Realizar

4.1. Objetivo

Determinar el tamaño de bloque, la cantidad de vias y el tamaño total de una memoria cache L1 de datos. Esta puede ser simulada utilizando la herramienta cachegrind o si no se simula ninguna cache, determinar la propia del procesador donde se ejecute el suite de benchmarking. El trabajo es de libre realización, pero se aconseja seguir el modelo que se propone

en la siguiente sección. Se pueden utilizar los lenguajes C o C++ para el desarrollo del suite de benchmarking.

4.2. Modelo propuesto

- Realizar un módulo que parsee los datos del archivo de salida del módulo cachegrind para que obtenga la tasa de miss de la cache L1.
- Realizar un módulo que ejecute un lazo para obtener el tamaño del bloque de la cache midiendo la tasa de miss que proporciona la herramienta parser del módulo cachegrind realizada en el primer paso.
- Realizar un módulo que ejecute un lazo para obtener el tamaño total de la cache bajo estudio midiendo la tasa de miss que proporciona la herramienta parser del módulo cachegrind realizada en el primer paso.
- Realizar un módulo que ejecute un lazo para determinar la cantidad de vías de la cache midiendo la tasa de miss que proporciona la herramienta parser del módulo cachegrind realizada en el primer paso.
- Realizar un programa que ejecute los módulos anteriores en orden, proporcionando los datos que sean necesarios para cada módulo y provea en su salida la información de la cache.

4.3. Parámetros del programa principal

Se deberán implementar las opciones:

- -V, --version
- -h, --help

En caso de existir errores, estos deben ser impresos por *stderr*.

4.4. Salida

La salida de la corrida del benchmark debe ser por *stdout* y tener el siguiente formato:

```
$>tp2
Cache de datos L1:
#Vias:           X  Vias
Tamaño:          Y  Bytes
Tamaño de bloque: Z  Bytes

$>
```

4.5. Pruebas

Se deberá correr el suite de benchmarking simulando las caches indicadas a continuación mostrando la salida con un screenshot de la pantalla:

- 1: Cache Direct Mapped, 16Bytes de línea, 8Kb tamaño total
- 2: Cache 2WSA, 32Bytes de línea, 32Kb tamaño total
- 3: Cache Fully Associative, 64Bytes de línea, 16Kb tamaño total

5. Informe

El informe deberá incluir

- Este enunciado.
- Descripción de los módulos realizados.
- Explicación del funcionamiento de cada módulo (Detallando cómo se calculan los distintos parámetros de la memoria cache).
- Código de los módulos realizados.
- Modelo de uso del suite de benchmarking.
- Pruebas realizadas para la comprobación del suite de benchmarking.

6. Fecha de entrega

La última fecha de entrega es el Jueves 3 de Diciembre de 2009.

7. Referencias

Referencias

- [1] Valgrind, <http://valgrind.org/>.