

Taller de Microarquitectura

Organización del Computador 2

Segundo Cuatrimestre 2014

1. Introducción

El objetivo de este taller es poder simular como funcionan algunos de los componentes internos de un procesador, utilizando la librería PIN de Intel que permite ejecutar instrucción a instrucción un binario cualquiera, poder pausar y simular una ejecución considerando una organización distinta a la nativa o con otros parámetros.

En este taller nos vamos a concentrar en como son los accesos a la memoria caché y en el sistema de predicción de saltos de un procesador. Vamos a simular el comportamiento de memorias caché reales de distintas arquitecturas y comprobar la eficiencia de distintos predictores de saltos sencillos.

2. Preparación Preliminar

Se deberán bajar de la página de Orga2 el paquete que incluye la librería PIN y los archivos del simulador de cache y del predictor de saltos desde el siguiente link:

<http://www.manuelferreria.com.ar/pin.tar.gz>

Se deberán compilar los simuladores también, invocando el script provisto: `./make_taller.sh`

3. Memoria Caché

3.1. El simulador

Para correr el simulador de memoria caché, se deberá invocar desde la carpeta donde esten copiados todos los archivos de la librería PIN de la siguiente manera:

```
./cachesim.sh [parametros] -- [binario]
```

Donde **parametros** puede ser una combinación de tamaños de L1, L2 y las asociatividades de las cachés. El resultado de las simulaciones se escribe en un archivo `cache.out`. Al ejecutar el simulador se corre el código que esta en el archivo: `pin/source/tools/Taller/cache.cpp`.

Para poder ver los parametros posibles, hacer `./cachesim.sh -h -- ls`

3.2. Tarea

Se deberán simular las memorias caché de los siguientes procesadores:

- AMD Geode LX (L1 64Kb, asociativa a 16 vias, 32B por linea, L2 128Kb, asociativa a 16 vias, 64B por linea)

- AMD Athlon64 X2 (L1 64Kb, asociativa a 16 vias, 64B por linea, L2 512Kb, asociativa a 16 vias, 64B por linea)
- Intel Pentium 3 (L1 32Kb, asociativa a 8 vias, 32B por linea, L2 512Kb, asociativa a 8 vias, 32B por linea)
- Intel Pentium 4 Cedar Mill(L1 16Kb asociativa a 8 vias, 64B por linea, L2 2Mb, asociativa a 8 vias, 64B por linea)

Para simularlas, se deberán utilizar los distintos parámetros que se proveen del simulador, y se deberán probar distintos programas para evaluar su comportamiento. Algunos sugeridos son: `ls`, `locate`, `bash`, `gedit`, `firefox`.

Además, se deberán implementar modificaciones al código C del simulador para estimar el tiempo total de ejecución considerando unicamente los *delay* de respuesta de cada una de las memorias cache. Tomando como datos que un HIT sobre L1 demora 1 ns, un MISS de L1 demora 1 ns, un HIT de L2 demora 4 ns y un MISS de L2 demora 100 ns. Estimar los tiempos totales por cada procesador para los distintos procesos ejecutados y hacer hipotesis sobre su performance en el mundo real.

4. Predictor de saltos

4.1. El simulador

Para correr el simulador de predicción de saltos, se deberá invocar desde la carpeta donde esten copiados todos los archivos de la libreria PIN de la siguiente manera:

```
./jgpsim.sh [parametros] -- [binario]
```

Donde `parametros` puede definir el tamaño de la tabla BHT usada en algunos predictores. Al ejecutar el simulador se corre el código que esta en el archivo: `pin/source/tools/Taller/jmp.cpp`

Para poder ver los parametros posibles, hacer `./jgpsim.sh -h -- ls`

4.2. Tarea

Se deberá modificar el simulador para agregar el mecanismo de predicción de saltos denominado, contador con saturación de dos bits, y se deberá comparar la eficacia de este método con los otros métodos ya implementados en el código fuente y descriptos en la clase, variando el tamaño de la BHT.

5. Resolución, informe y forma de entrega

Se recomienda realizar este taller de forma grupal, en grupos de hasta 3 personas. Este taller es optativo y no tiene carácter de evaluación, por lo que NO deberán entregar un informe.

La idea del taller es poder resolverlo en clase, obtener conclusiones comparativas sobre los efectos de la memoria caché, el predictor de saltos y sobre el impacto de estos en la performance de las aplicaciones.