

Practica 1 Andreu AIC

Parte 1

Esta practica es para la ley de Amdahl. Voy a poner un simil para saber que es cada cosa...

$$S' = \frac{1}{(1-F) + \frac{F}{S}} \quad S': \text{Aceleracion global} \quad S: \text{Aceleracion local}$$

Supongamos que tenemos un disco duro HDD y lo sustituimos por un SSD, el SSD va 5 veces mas rapido que el HDD. Al sustituir el componente tenemos una aceleracion local (S) de 5.

Pero el ordenador no utiliza el 100% del tiempo el SSD por tanto hay que calcular una aceleracion global, en la cual hay que tener en cuenta la memoria y sustituir la fraccion de tiempo en la que se usa el SSD (F).

El simil es el siguiente...

En nuestra practica tenemos un programa en C llamado **scalar.c**, que se puede ejecutar de dos maneras (eso seria el disco duro). Una mas rapida y una mas lenta. Eso nos dara una aceleracion local al utilizar la mejora (la rapida). **Asi obtenemos la aceleracion local la S**. Restando una latencia en la formula

Despues tenemos otro archivo. **Matrix.c** que eso seria el equivalente al ordenador en el ejemplo anterior, se puede ejecutar de dos formas tambien y a partir de él sacamos la **aceleracion global (S')**

Despues de realizar cambios sobre scalar.c y matrix.c obtenemos unos archivos compilados que se ejecutan de distintas maneras y nos dan unos tiempos para la formula

----- Calculo de la S: (scalar.c)

El tiempo antes de la mejora se obtiene: T_{std}

```
amutpor@alumno.upv.es@ldsic-vdi05:~/W/AIC/prac1$ gcc -O0 -o scalar-std scalar.c
amutpor@alumno.upv.es@ldsic-vdi05:~/W/AIC/prac1$ time ./scalar-std 100000 1024
Rep = 100000 / size = 1024

real    0m0,805s
user    0m0,801s
sys      0m0,000s
```

El tiempo tras la mejora es: T_{sse}

```
amutpor@alumno.upv.es@ldsic-vdi05:~/W/AIC/prac1$ time ./scalar-sse 100000 1024
Rep = 100000 / size = 1024

real    0m0,689s
user    0m0,685s
sys      0m0,000s
```

Ahora calcular la sobrecarga, que es lo que tardan los bucles para restarla en la aceleracion y tener un tiempo mas fiel. T_{load}

```
amutpor@alumno.upv.es@ldsic-vdi05:~/W/AIC/prac1$ time ./scalar-load 100000 1024
Rep = 100000 / size = 1024

real    0m0,572s
user    0m0,565s
sys     0m0,004s
```

Con estos datos se obtiene la aceleracion local.

$$S = \frac{T_{std} - T_{load}}{T_{sse} - T_{load}} = \frac{0,805 - 0,572}{0,689 - 0,572} = 1,99$$

Calculo de la S': (matrix.c)

El tiempo antes de la mejora se obtiene: $T_{mat-std}$

```
amutpor@alumno.upv.es@ldsic-vdi05:~/W/AIC/prac1$ gcc -O0 -msse -o matrix-std matrix.c -lm
amutpor@alumno.upv.es@ldsic-vdi05:~/W/AIC/prac1$ time ./matrix-std 1 1024
Rep = 1 / size = 1024

real    0m3,600s
user    0m3,584s
sys     0m0,012s
```

El tiempo tras la mejora es: $T_{mat-sse}$

```
amutpor@alumno.upv.es@ldsic-vdi05:~/W/AIC/prac1$ gcc -O0 -msse -o matrix-sse matrix.c -lm
amutpor@alumno.upv.es@ldsic-vdi05:~/W/AIC/prac1$ time ./matrix-sse 1 1024
Rep = 1 / size = 1024

real    0m2,350s
user    0m2,338s
sys     0m0,008s
```

$$S' = \frac{T_{matstd}}{T_{matsse}} = \frac{3,600}{2,350} = 1,53$$

Calculo de la F:

Con los datos anteriores se puede sacar la F despejando:

$$S' = \frac{1}{(1-F) + \frac{F}{S}} \rightarrow \frac{1}{(1-F) + \frac{F}{1,99}} = 1,53 \quad 1 = 1,53 - 1,53F + 0,76F \rightarrow 0,77F = 0,53 \rightarrow F = 0,68$$

Despejando la F sale que tiene valor 0,68

Calculo experimental de la fraccion de tiempo local (F_{exp})

Primero obtener el T_{mat_res} . Esto es algo que se explica en el boletín.

```
amutpor@alumno.upv.es@ldsic-vdi05:~/W/AIC/prac1$ gcc -O0 -msse -o matrix-res matrix.c -lm
amutpor@alumno.upv.es@ldsic-vdi05:~/W/AIC/prac1$ time ./matrix-res 1 1024
Rep = 1 / size = 1024

real    0m0,752s
user    0m0,750s
sys     0m0,000s
```

Esta matriz en concreto permite aplicar una F experimental, no es algo que se pueda hacer a menudo pero se sacaría así:

$$F_{exp} = \frac{T_{matstd} - T_{matres}}{T_{matstd}} = \frac{3,600 - 0,752}{3,600} = 0,79$$

Comparando la obtenida con Amdahl que era 0,68 no dista tanto de la experimental 0,79.

Hay diferencia porque en cada ejecución, los tiempos salían muy dispares, eso se debe a que el sistema operativo puede sacar de ejecución el proceso y hace que aumente el tiempo considerablemente.

Parte 2

dhystone con 10 000 000 iteraciones. $T_{dhystone}$

```
real    0m11,763s
user    0m0,710s
sys     0m0,000s
```

whetstone con 10 000 iteraciones T_{whet-h}

```
amutpor@alumno.upv.es@ldsic-vdi05:~/W/AIC/prac1$ time ./whetstone 10000

real    0m0,337s
user    0m0,166s
sys     0m0,000s
```

Compilador C con tiempo T_{gcc}

```
real    0m21,572s
user    0m12,643s
sys     0m2,415s
```

Aplicacion xv con tiempo T_{xv}

```
amutpor@alumno.upv.es@ldsic-vdi05:~/W/AIC/prac1$ time xv-310a/xv -wait 5 mundo.jpg
pg
real    0m6,183s
user    0m0,895s
sys     0m0,101s
```