

2. MFLOPS, Millions of FLOating Point operations per Second

Calcula los MFLOPS para cada tamaño y nivel de optimización (velocidad de las operaciones suma y producto de coma flotante efectuados por el programa). La cifra cambia según el tamaño de la matriz, ¿por qué?

Los cálculos están en el Excel.

Esto se debe a que una menor carga de datos supone un mayor aprovechamiento de la cache ya que no se producen tantos misses.

¿Por qué la cifra que habéis obtenido al ejecutar gauss es muy inferior a los MFLOPS de pico anunciados por la propaganda? ¿Qué tipo de programas rondaría los MFLOPS de pico?

Esto se debe a que gauss solo tiene 2 operaciones de coma flotante y además tiene muchos accesos de memoria, por lo que afecta al tiempo de ejecución y por tanto el numerador es más pequeño y el denominador más grande.

Para acercarse se requeriría un programa con las características opuestas a gauss, muchas operaciones de coma flotante y muy pocas de acceso a memoria.

3. MIPS, Millions of Instructions Per Second

Para cada nivel de optimización y arquitectura, ¿cuántas instrucciones correspondientes al cuerpo del bucle L/U ($A[i][j] = \dots$) se ejecutan en cada iteración?

Calcular las velocidades en MIPS suponiendo que los tiempos medidos en el apartado anterior se deben exclusivamente a la ejecución de las instrucciones correspondientes al cuerpo del bucle L/U.

Ambas respuestas están en el Excel adjunto.

¿Por qué la cifra que habéis obtenido al ejecutar gauss es muy inferior a los MIPS de pico anunciados por la propaganda? ¿Qué tipo de programas conseguiría los MIPS de pico?

Se debe a que las instrucciones de acceso a memoria y de coma flotante suponen un gran coste en tiempo respecto al resto.

Aquellos con pocas instrucciones de este tipo, por ejemplo, programas de operaciones lógicas y/o enteras.

4. CPI (Ciclos Por Instrucción)

Conociendo el tiempo de ciclo del procesador, calculad el CPI de este programa. ¿Para qué tamaño y nivel de optimización se obtiene el mayor CPI? ¿Y el menor? ¿Cómo se explican las diferencias, si existen? ¿Menor CPI implica siempre menor tiempo de ejecución?

CPI calculado en el Excel.

Mayor CPI en matrices grandes con O2 y el menor en matrices pequeñas y al ser pequeñas el nivel de optimización no hace mucho cambio.

Dado que las matrices más grandes requieren más accesos a memoria, habrá más misses por parte de la cache pues no caben enteras en las rápidas y como hay más misses, habrá más ciclos, incrementando el CPI.

5. Ancho de banda

Rellenad una tabla en la que se especifiquen, para las versiones optimizadas de las dos arquitecturas, los bytes de cada tipo leídos/escritos en una iteración del bucle interno (ver la siguiente tabla).

Rellenad ahora otra tabla en la que, para cada tamaño de matriz y para cada arquitectura, se especifiquen los tiempos de ejecución y los MBps de cada tipo.

Ambas tablas están en el Excel adjunto.