

Primera tarea de AOC, por Jose Antonio Lorenzo Abril

Con este programa, obtenemos, variando el tamaño de los bucles, cualquier entero hasta 16777216.000, que es precisamente 2^{24} , representado como float.

Esto quiere decir que el problema lo encontramos a partir de este número.

Su representación en coma flotante es 0 10010111 000000000000000000000000.

Ahora bien, 1.0 es 0 01111111 000000000000000000000000.

Para sumarlos, deben tener el mismo exponente, es decir, lo que haremos será $2^{24} + (1 \cdot 2^{-24}) \cdot 2^{24}$, y aquí es donde se produce el error.

La precisión de los float es, precisamente, 24 bits. De forma que, al sumar estos dos números, estamos sumando algo que produce una diferencia menor que la precisión de la máquina, luego el número no cambia, no importa cuantas veces hagamos esta suma.

Si cambiamos el 1.0 por un 100.0, llegamos hasta 2147483648.000, donde este mismo error se produce para el número 100.

Para calcular los gigaflops, simplemente hacemos

$$\frac{MAXFLOPS_ITERS \cdot LOOP_COUNT}{tiempo \cdot 10^9} = \frac{10^7 \cdot 64}{tiempo \cdot 10^9} = \frac{0.64}{tiempo} = \frac{0.64}{1.885687} = 0.3394 \text{ GFlops}$$

donde el tiempo ha sido obtenido con la función clock() de C:

```
clock_t begin = clock();
/* bucles */
clock_t end = clock();
double time_spent = (double)(end - begin) / CLOCKS_PER_SEC;
```