

Rapport HPC Lab 6 - Mesure de consommation énergétique - Samuel Roland

Ma machine

Extrait de l'output de `fastfetch`

```
OS: Fedora Linux 41 (KDE Plasma) x86_64
Host: 82RL (IdeaPad 3 17IAU7)
Kernel: Linux 6.13.5-200.fc41.x86_64
CPU: 12th Gen Intel(R) Core(TM) i7-1255U (12) @ 4.70 GHz
GPU: Intel Iris Xe Graphics @ 1.25 GHz [Integrated]
Memory: 15.34 GiB - DDR4
Swap: 8.00 GiB
```

Code

Je préférerais repartir sur une base propre et j'ai recodé un petit algo de somme de floats comme demandé. Tout le code est contenu dans `main.c`, une macro activée à la compilation permet de builder 2 exécutables pour les 2 variantes basique et SIMD.

Une fois compilé avec CMake comme d'habitude (`cmake . -Bbuild && cmake --build build/ -j 8`)

```
./build/floatsparty_basic
```

Ou la version SIMD

```
./build/floatsparty_simd
```

J'ai activé les flags `-O2 -g -ggdb -Wall -fno-tree-vectorize` pour éviter les autovectorisations mais optimiser un minimum. Il y a en plus `-mavx2` pour la version SIMD pour accéder aux intrinsèques.

J'ai pris une taille de 10000000 = 10 millions de float = 40 millions de bytes par défaut, mais celle-ci peut être changée en premier argument du programme, par ex.

```
./build/floatsparty_basic 1212
```

Mesures générales

```
perf stat -e power/energy-psys/ ./build/floatsparty_simd
```

Oups, il semble que cette événement ne soit pas supporté sur mon processeur

```
> perf list
...
msr/tsc/ [Kernel PMU event]
power/energy-cores/ [Kernel PMU event]
power/energy-gpu/ [Kernel PMU event]
power/energy-pkg/ [Kernel PMU event]
uncore_clock/clockticks/ [Kernel PMU event]
```

Je vais prendre `power/energy-pkg/` à la place comme cela me parait être ce qui mesure la zone la plus large (cores + autour) du CPU (et que le code ne tourne pas sur GPU), en espérant que les résultats soient similaire à l'événement `energy-psys`.

Comme on a pas hyperfine et qu'on a quand même 0.1-0.5 variations de joules entre chaque exécution, j'ai refait à l'aide de GPT-4.1 une fonction Fish pour mesurer facilement pour la suite et avoir une sorte d'hyperfine qui lance **10** fois et fait la moyenne, pour la variante donnée.

```
> sudo perf stat -e power/energy-pkg/ ./build/floatsparty_simd
Performance counter stats for 'system wide':
    2.34 Joules power/energy-pkg/
    0.125022687 seconds time elapsed

> sudo perf stat -e power/energy-pkg/ ./build/floatsparty_simd
Performance counter stats for 'system wide':
    2.24 Joules power/energy-pkg/
    0.125834400 seconds time elapsed
```

Une fois Fish installer il suffit de faire `source measure.fish`, pour avoir ensuite accès

```
> measure basic && sleep 1 && measure simd
Average for basic: 2.32 joules
Average for simd: 2.22 joules
```

On voit qu'on a quand même une différence de consommation avec un léger gain sur SIMD, c'est difficile de voir si cela fait beaucoup.

Ce qui est étonnant c'est que cette amélioration n'est pas toujours présente.

```
> measure basic && sleep 1 && measure simd
Average for basic: 2.24 joules
Average for simd: 2.21 joules
> measure basic && sleep 1 && measure simd
Average for basic: 2.24 joules
Average for simd: 2.32 joules
```

Même avec 40 itérations cette instabilité est toujours notable

```
> measure basic && sleep 1 && measure simd
Average for basic: 2.24 joules
Average for simd: 2.25 joules
> measure basic && sleep 1 && measure simd
Average for basic: 2.31 joules
Average for simd: 2.33 joules
```

C'est un peu déroutant ces mesures... Si je devais coder des améliorations j'aurai pas mal de peine à être sûr de l'amélioration.

Marqueurs avec perf

Pareil qu'avant, mais avec beaucoup plus de galère à automatiser à cause du passage en sudo de perf qui m'a fait perdre du temps de debug au niveau du script, avec les fd qui générait des "Bad file descriptor", parce que ça changeait d'utilisateur j' imagine ? A nouveau `source measure2.fish` va utiliser `perf-markers.sh`

Parfois la différence est minime, parfois elle est plus notable au niveau du gain. Je suis resté sur une moyenne avec 40 itérations.

```
> measure2 basic && sleep 1 && measure2 simd
Average for basic: .084915 joules
Average for simd: .085324 joules
> measure2 basic && sleep 1 && measure2 simd
Average for basic: .091502 joules
Average for simd: .086419 joules
```