TP

Observation et outils de mesure énergétique

Préambule

Tous les outils de mesure d'énergie nécessitent soit des droits de superutilisateur sur la machine, soit du matériel supplémentaire (wattmètre). Les *hyperscalers* (AWS, Google Cloud, etc.) ne vous donnent pas accès à de tels outils. Néanmoins, le contenu présenté dans ce TP devrait être utile lors de (i) la phase de développement, où l'application s'exécute sur un ordinateur local, ou (ii) dans un cloud privé, où un contrôle plus approfondi du système/matériel est possible.

1 Objectifs du TP

- Utiliser des outils de monitoring d'énergie
- Visualiser quels composants sont inclus dans un outil de mesure
- Étudier l'impact du nombre de ressources dans la consommation énergétique
- Étudier l'impact du choix du datacentre dans l'émissions de CO2

2 Consommation électrique et impact environnemental

2.1 Qu'est-ce qui consomme ?

L'ordinateur sur lequel tourne votre programme consomme de l'électricité pour alimenter ses composants : processeur, barrettes de RAM, disque dur, carte réseau, ventilateurs, périphériques USB, etc. Il y a également des pertes dans l'alimentation électrique (rendement inférieur à 100%) et dans les circuits (perte sous forme de chaleur par effet Joule). La consommation des composants dépend parfois de leur utilisation. C'est par exemple le cas pour le CPU, car les CPUs modernes sont capables de s'éteindre partiellement lorsqu'ils sont peu sollicités.

La consommation d'un logiciel donné n'est pas directement mesurable, mais on peut l'estimer en attribuant la consommation du matériel au logiciel.

> ℹ️ Pour avoir une attribution plausible, il faut bien sûr tenir compte de l'utilisation du matériel par le logiciel.
>
> Intuitivement, si on exécute deux programmes en même temps sur la même machine :
> - un programme A qui ne fait que "dormir" (par exemple avec un `sleep(99999999)`)
> - un programme B de simulation numérique qui exécute en boucle des calculs matriciels
>
> On voit que le programme B devrait être "responsable" d'une plus grande part de la consommation électrique que le programme A.
>
> En connaissant la consommation du matériel et l'utilisation du matériel par le logiciel, on peut calculer une estimation de l'énergie consommée par chaque programme.

2.2 Comment mesurer ?

> ℹ️ Pour cette partie, vous devez impérativement être sous Linux, de préférence Ubuntu 22.04 LTS (les postes de l'IM2AG correspondent parfaitement à cette contrainte).

Le plus simple est de se concentrer sur le CPU et la RAM, qui disposent d'interfaces bas niveau permettant de lire leur consommation électrique à chaque petit intervalle de temps Δt (chez Intel, $\Delta t = 976\mu s$). Il existe plusieurs logiciels pour récupérer cette information, mais la plupart souffrent d'erreurs d'implémentation ou de limitations conceptuelles [1].

Pour ce TP, nous allons utiliser un nouvel outil : Alumet (Adaptive, Lightweight, Unified METrics). Alumet propose un cadre modulaire permettant de mesurer tout type de matériel, de récolter des informations sur le système, et de combiner les deux vers tout type de sortie (fichier, base de données). Un système de plugins permet de construire un outil de mesure "sur mesure". Alumet peut se déployer sur une seule machine ou dans un système distribué (Kubernetes par exemple).

Vous allez utiliser Alumet pour mesurer la consommation d'énergie de votre machine pendant que le scheduler s'exécute, et obtenir une estimation de la consommation d'un programme en exécution.

❓
1. Aller lire le README du dépôt GitHub. N'hésitez pas à mettre une petite étoile ⭐ !
2. Télécharger Alumet à utiliser pour ce TP, c'est le fichier `alumet-local-agent`.
3. Vérifier qu'Alumet tourne :

```
git clone https://github.com/danilo-carastan-santos/measuring-energy-alumet.git
cd Bin
./alumet-local-agent exec -- sleep 3
```

Vous devriez obtenir un fichier `alumet-output.csv` contenant des mesures. Utilisez la documentation d'Alumet pour comprendre le contenu du fichier.

2.3 Première approche avec TDP

Pour cette approche, il suffit de connaître le TDP (depuis l'anglais *Thermal Design Power*) du processeur utilisé et le temps d'exécution de l'application. Pour le premier, vous pouvez obtenir d'informations sur le processeur avec la commande ci-dessous.

```
cat /proc/cpuinfo
```

❓
- Pourquoi la commande dessus semble afficher plusieurs fois la même information ?
- En cherchant sur Internet, quel est le TDP du processeur utilisé ?

2.4 Observer la consommation énergétique avec Alumet et `stress`

Pour commencer, lançons un stress sur le CPU. Vous pouvez faire varier le nombre de *workers* avec l'argument -c. Nous allons utiliser la librairie Python `subprocess` qui permet lancer des commandes depuis un code Python.

Le code Python disponible dans le fichier `./Bin/alumet-stress-test.py` (code ci-dessous) permet de lancer Alumet et `stress`. Il est possible de lancer ce stress test avec la commande suivante.

```
python3 alumet-stress-test.py
```

1. Guillaume Raffin, Denis Trystram. Dissecting the software-based measurement of CPU energy consumption : a comparative analysis. 2024. hal-04420527v2

```python
import subprocess
from datetime import datetime
import time
import os

NB_CPUS = 1
PROC_TIME_SECONDS = 10
SLEEP_TIME_SECONDS = 1

ALUMET_RESULT_FILENAME = "../Results-CSV/"+datetime.now().strftime('%Y-%m-%d_%H:%M:%S')+"_alumet-output.csv"+".csv"

alumet_command = ["./alumet-local-agent"]

stress_command = ["./stress","-c",str(NB_CPUS),"-t",str(PROC_TIME_SECONDS)]

## Lancer Alumet et la commande stress

alumet_start = time.time()
## Lancer perf stat et ne pas attendre sa fin
alumet_pid = subprocess.Popen(alumet_command).pid

## Attendre un peu pour bien monitorer la commande stress
time.sleep(SLEEP_TIME_SECONDS)

stress_start = time.time()
## Lancer la commande stress et attendre sa fin
subprocess.run(stress_command)
stress_end = time.time()

## Attendre un peu pour bien monitorer la commande stress
time.sleep(SLEEP_TIME_SECONDS)

## Terminer Alumet en envoyant SIGINT (-2)
## Équivalent à Ctrl+c
subprocess.run(["kill", "-2", str(alumet_pid)])

## Renommer le CSV de sortie d'Alumet
os.rename("./alumet-output.csv", ALUMET_RESULT_FILENAME)
print("Done")
```

FIGURE 1 – Code source de `./Bin/alumet-stress-test.py`

❓
- Utiliser le script `./Bin/alumet-stress-test.py` pour obtenir des mesures lors de l'exécution de `stress`. Tracez une courbe (avec LibreOffice ou Python par exemple) de la **puissance** du CPU en fonction du temps (timestamp).
- À l'aide du même script, calculer la consommation énergétique du CPU pendant l'exécution de *stress*
- Comparer la consommation des différentes exécutions de `./Bin/alumet-stress-test.py`, en changeant la valeur de `NB_CPUS`.

⚠️ **Attention !** La sortie `rapl_consumed_energy_J` d'Alumet montre l'énergie consommée et non la puissance. La puissance est normalement mesurée en Watts = Joules / secondes. C'est à vous de faire la bonne conversion pour avoir la courbe de puissance.

ℹ️ Il se peut que vous obteniez des erreurs ou warnings à propos de priorité de threads. C'est dû à un manque de privilèges sur les machines de l'école. Ignorez-les, ça n'empêche pas Alumet de fonctionner.

En cas d'erreur plus grave, appeler le professeur pour trouver une alternative.

2.5 Empreinte carbone

Produire de l'électricité entraîne des émissions de gaz à effet de serre plus ou moins importantes en fonction du mode de production. L'empreinte carbone d'un datacentre dépend donc de ses sources d'électricité et, à moins qu'il ne soit totalement indépendant du réseau, de sa position.

❓
- Utiliser ElectricityMaps pour visualiser l'intensité carbone (en grammes d'équivalent CO2 par kWh d'électricité) de la consommation d'énergie par pays. Quelles sont les sources dites "bas carbone" ?
- Multiplier l'intensité carbone de la consommation d'électricité en France par la consommation du CPU lorsqu'il exécute `stress` pour obtenir une idée de son empreinte carbone en phase d'usage. Extrapoler à une année entière d'exécution de `stress` (24h/24 7j/7).
- Utiliser la liste d'intensité carbone de serveurs Google Cloud pour comparer l'intensité carbone de serveurs Google dans plusieurs régions
- Utiliser le simulateur de prix de machine virtuelles de Google Cloud pour comparer les émissions *versus* le prix de chaque région des datacentres Google Cloud.
- Essayer de trouver les mêmes types d'informations des points précédents pour d'autres *hyperscalers* (Amazon AWS, Microsoft Azure).

ℹ️
- Pour la liste d'intensité carbone de serveurs Google Cloud, Google utilise deux indicateurs : (1) le CFE% (*Carbon Free Energy*), qui est une pourcentage du total d'énergie utilisé qui provient d'une source à très faibles émissions [a], et (2) le *Grid Carbon Intensity*, qui est l'intensité carbone du reste de l'énergie qui n'est pas fournie par le CFE. Exemple : pour 10 kWh de consommation énergétique dans une région avec CFE = 40% et *Grid Carbon Intensity* = 456 gCO2eq/kWh. 10 kWh · 0.4 = 4 kWh d'énergie a été consommée à très faibles émissions, et le reste a produit 6 kWh · 456 gCO2eq/kWh = 2,73 kgCO2eq d'émissions.
- Pour le simulateur de prix de machine virtuelles, vous pouvez utiliser une configuration standard. Par exemple avec une machine virutelle `n1_standard_4`. Il suffit de cliquer sur *Add to estimate* et puis sur *Compute Engine*. La machine virtuelle `n1_standard_4` sera le choix par default.
- Pour les exercices avec les informations de Google Cloud, nous supposons qu'une machine virtuelle occupe entièrement une machine physique, ce qui n'est largement pas le cas. La consommation d'une machine virtuelle n'est qu'un pourcentage de la consommation totale de la machine physique (*host machine*). Google offre un service Empreinte Carbone qui donne une idée des émissions. À vous de voir s'il y a de services similaires pour les autres *hyperscalers*.

a. qu'on suppose qui est proche de zero

2.6 Un mot sur ce que nous n'avons pas mesuré

La production de gaz à effet de serre n'est pas le seul impact de l'informatique, et il n'y a pas que la phase d'usage qui compte. Il faut aussi fabriquer, transporter, puis gérer la fin de vie des équipements. Même en phase d'usage, la consommation électrique entraîne d'autres impacts : en plus de l'empreinte carbone, on devrait évaluer la consommation d'eau, de métaux, etc.