



Hackathon GreenIT @ 42 Lausanne

Mesure de la consommation électrique d'une application

2-4/2/2024

Problèmes

- Isoler les activités sur une machine
- Estimer l'impact du monitoring lui-même
- Marge d'erreur potentiellement grande
- Convertir l'utilisation des ressources (CPU, GPU, RAM, I/O, ...) en kWh est forcément une estimation
- Très différent d'une machine à une autre

Interrogations

- À quel point la surcharge du système impacte la conso totale ?
- Comment créer un environnement stable pour avoir des mesures consistantes et précises ?
- Opérations dans le code: *while* versus *for*, etc.
- FLOPS (*floating operations/second*) comme unité de mesure ?

Observations

- La mesure de l'utilisation du CPU, RAM, ... est instantanée
- *Activity monitor* sépare l'usage du CPU selon que le process est lié au système ou aux activités de l'utilisateur, notamment les apps lancées
- Le choix du langage importe beaucoup, si l'on veut utiliser moins d'énergie, on devrait plutôt s'orienter vers C, Rust, C++, ...

Outils existants

Hardware

- JouleScope:



Outils existants

Hardware

Caractéristiques (*SmartWatts*, Kieni et al. 2020):

- Mesure de la consommation peu importe le système
- Précis mais peu granulaire (données agrégées uniquement)

Outils existants

Software

- Apps:
 - Mac Power Monitor
 - HWInfo
 - ...
- Outils de fabricants:
 - vTune (Intel), uProf (AMD), RAPL (Intel)

Outils existants

Software

- Problème (*SmartWatts*, Kieni et al. 2020) : taux d'erreur élevé par rapport aux outils hardware
- Explication: les métriques traditionnelles d'usage du CPU, mémoire, disque, réseau, ... calculées par le kernel sont peu précises.

Outils existants

Software (...)

- Solution: observer certaines actions précises du système (*“number of retired instructions, cache misses, non-halted cycles..”*) qui sont fortement corrélées avec la conso. du CPU + utilisation d’outils hardware pour corréler ces métriques avec la consommation (via des modèles statistiques)

Outils existants

Software

- Outils natifs Apple/OS X:
 - *Activity Monitor, Xcode, Instruments*: Energy Impact (score)
 - **Powermetrics** : Rapport en milliWatts pour CPU, GPU, ANE*

***ANE**: Apple Neural Engine (puce Apple pour le machine learning)

Mesure

Cibles

- Ressources:
 - CPU
 - GPU
 - ANE (Apple Neural Engine)
- Utilisation combinée

Mesure

Durée des tests

- Charge au repos (*idle*): 2 min

Justification: par contrainte de temps, 2 min pour simplifier nos tests. En théorie, 15-30min idéal. Risque: système n'a pas le temps de se stabiliser pendant les 2 min.

- Charge au travail: 2 min

Mesure

Intervalles d'observation

- Intervalles au repos et au travail: 200 ms (300 observations/min)

Justification: adapté aux applications dynamiques (100-500ms)
=> plus de granularité (facilite la détection de problèmes au sein de l'application). Applications à charge de travail plus stable: 1000-2000 ms. Et pour la cohérence des mesures, même intervalle au repos et au travail.

Mesure

Benchmarking

- Repos: networking éteint (Wifi, Bluetooth, Ethernet, autres apps)
- Travail: même environnement, script lance un jeu vidéo plein écran avec personnage qui danse

Mesure

Statistiques

- Moyenne simple de puissance (sur la durée de test)
- Variance (*std deviation*)
- Valeurs extrêmes (min, max)
- Consommation électrique combinée et par ressource (CPU, GPU, ANE)

Merci !

