

## Analyse de la consommation des algorithmes

### Introduction

L'objectif de ce rapport est d'analyser la consommation énergétique ou encore la complexité énergie des algorithmes. « *La consommation énergétique du numérique augmente de 8,5% par an ...* »<sup>1</sup>. Il est indéniable de nos jours que le numérique consomme de plus en plus d'énergie, soit à travers des *data centers* ou encore de l'augmentation d'appareils informatiques personnels, comme les smartphones et les ordinateurs portables, qui fonctionnent à l'aide de batteries. Avec l'émergence du IA, il est d'autant plus important de trouver des solutions afin d'optimiser les processus et réduire la consommation d'énergie des algorithmes qui demandent de plus de plus de puissance de CPU et GPU.

La complexité algorithmique se traduit par les ressources nécessaires permettant de résoudre un problème en entrée au moyen de l'exécution d'un algorithme. Les principales ressources mesurées sont le temps, l'espace mémoire ou encore la complexité énergie, une mesure sur laquelle nous nous concentrons dans le cadre de cette analyse.

Afin d'évaluer la consommation d'énergie nécessaire pour exécuter un algorithme, nous pouvons utiliser des outils qui permettent de monitorer la quantité d'énergie utilisée par le logiciel et donc, dans beaucoup de cas, par le CPU (l'unité centrale de traitement). Nous nous concentrons, dans le cadre de ce rapport, à analyser l'énergie consommée par le CPU, et pour le faire nous pouvons utiliser différentes mesures, comme le pourcentage de batterie utilisée, la température du CPU, sachant que plus celle-ci est élevée, plus l'usage du CPU est grand, ou encore tout simplement le pourcentage de CPU utilisé.

### Paramètres utilisés

Dans notre cas, nous nous focalisons sur une mesure d'analyse de consommation, soit l'énergie en Wh utilisée par le CPU afin de rouler un algorithme. Cette énergie est elle-même évaluée à partir du pourcentage de batterie restant au moment  $t$ . Avant d'aller plus loin dans notre analyse, nous rappelons quelques paramètres utilisés dans l'algorithme de monitoring utilisé dans le cadre de cette étude :

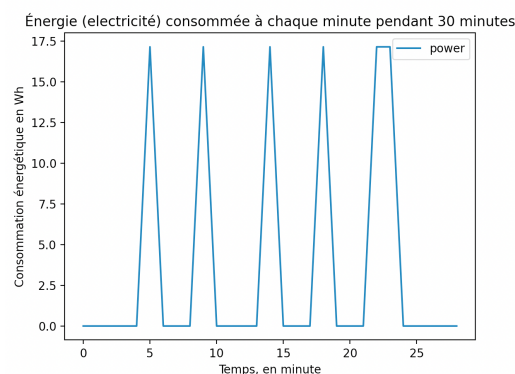
1. **Condition de la batterie:** L'état de santé des batteries varient pour chaque ordinateur. Cette différence est dû à différents facteurs, par exemple, '*l'âge*' d'un ordinateur, le type d'ordinateur, de processeur, la fréquence d'utilisation de cet ordinateur et le type de tâches donné à celui-ci. Plus les tâches sont '*lourdes*', plus l'état de santé de la batterie se détériore rapidement. Une autre raison de développer des algorithmes plus optimisés qui consomment moins! Dans notre cas, la condition de la batterie est autour de 52Wh initialement et est actuellement autour de 28.6 Wh, ce qui correspond à, à peu près, 55% de sa capacité originale.
2. **Critère d'arrêt :** Dans notre cas, nous utilisons un critère d'arrêt équivalent au nombre de minutes ou d'heures choisies pour faire notre approximation. La période utilisée, ou encore le temps entre deux mesures, en secondes, est de 60 secondes.
3. **Puissance électrique, *power\_est*.** La puissance électrique d'un appareil est mesurée en Watt. Il s'agit de la puissance d'un appareil capable de fournir une énergie d'un joule. Dans notre cas, nous calculons la puissance de notre ordinateur en nous basant sur l'état de la batterie de celui-ci (dans notre cas 28.6Wh sur 52Wh). Plus la batterie est de bonne condition plus la puissance de l'ordinateur est grande. Nous cherchons, dans le cadre de cette analyse, à recadrer la puissance électrique de l'ordinateur par rapport à la période de l'expérience et la différence du niveau de charge de l'ordinateur à chaque itération (dans notre cas à chaque minute) de l'expérience. Cela nous permet d'avoir la puissance électrique associée à la différence de charge de l'ordinateur, nous donnant ainsi 17.16W pour 1% de baisse de charge batterie. Il est important de préciser que cette puissance a été ramenée sur une période d'une heure nous permettant ainsi d'utiliser la relation suivante : un appareil de 1 Watt consommera 1 Wh d'électricité si elle reste allumée pendant une heure. Nous pouvons ainsi dire que la puissance électrique de notre processeur de 17.16W consommera une énergie de 17.16Wh pendant une heure. Nous notons, de plus, qu'un algorithme ayant une complexité énergie plus grande consommera plus d'énergie en Wh. C'est équivalent à dire que plus un algorithme consomme de l'énergie, plus le niveau de charge de l'ordinateur baissera, sachant que chaque baisse de niveau de charge (d'un pourcent) est équivalent à 17.16Wh d'énergie consommée. Dans notre cas, nous illustrons chaque baisse d'un pourcent de charge par un pic (la figure ci-dessous) de 17.16 Wh consommé. Plus il y aura des pics durant l'expérience, plus l'algorithme consommera de l'électricité.

<sup>1</sup> C. Villani, 'Donner un sens à l'intelligence artificielle pour une stratégie nationale et européenne ' Mission parlementaire du 8 septembre 2017 au 8 mars 2018.

## Application

Nous cherchons maintenant à appliquer notre algorithme de *monitoring* de consommation d'énergie avec les paramètres tel que précisés ci-dessus. Notre temps d'étude est fixé à 30 min. Nous choisissons d'exécuter un algorithme simple, connu, celui du factoriel, en utilisant une méthode itérative (qui, nous le rappelons, a une complexité temps,  $\Theta(n)$ , et qui prend un temps non-négligeable à calculer le factoriel d'un nombre naturel,  $n$ , lorsque celui-ci est grand). Nous précisons le niveau de charge de la batterie juste avant le 'lancement' de l'algorithme: 73%. Nous faisons également l'hypothèse que l'algorithme de *monitoring* ne consomme pas d'énergie et nous nous assurons qu'aucun autre calcul ou processus ne soit exécuté en même temps. Ainsi, nous supposons que la diminution du niveau de charge, est proportionnelle à l'exécution de l'algorithme du factoriel – itérative.

Nous affichons ci-dessous l'énergie consommée en Wh, à chaque minute pendant les 30 minutes de l'expérience. Nous rappelons que cette énergie de 17.16Wh est proportionnel à la puissance électrique en Watt de l'ordinateur associée à la différence du niveau de charge de l'ordinateur (soit 17.16W pour 1% de baisse de charge).



Chaque pic représente le moment où le niveau de charge baisse d'un pourcent (qui est équivalent à une perte d'énergie de 17.16Wh). Nous remarquons, qu'au début, l'algorithme consomme moins comparé à la fin. En effet, au début, cela prend 5 minutes pour que 17.16Wh soit consommé, puis entre 5 et 20 minutes, cela prend en moyenne 3-4 minute pour qu'un autre 17.16Wh soit utilisé, et finalement à partir de 23 minutes, 2\*17.16Wh d'énergie est consommé. Au total, en 30mins, l'algorithme a consommé 102.96Wh d'énergie pour une machine de puissance électrique de 17.16W par pourcentage de niveau de charge de la batterie.

Aux fins de l'analyse, nous faisons ce même test sur un algorithme modifié, le factoriel récursive, qui a également une complexité temps de  $\Theta(n)$ , mais qui est plus rapide que la version itérative lorsque  $n$  est grand. Nous ne notons aucun changement dans l'énergie consommé. De plus, nous analysons également d'autres algorithmes connus de recherche linéaire et recherche binaire, qui ont respectivement  $\Theta(n)$  et  $\Theta(\log n)$  en complexité temps. Bien qu'il y ait une différence dans les complexités temps, nous ne remarquons pas de différences notables au niveau de la complexité d'énergie.

Ces tests nous montrent qu'un algorithme ayant une plus petite complexité temps et mémoire n'aura pas forcément une petite complexité énergie. En effet, bien qu'un algorithme prenne plus de temps à être exécuté il peut ne pas exploiter toute la puissance de l'ordinateur et, de façon proportionnelle, il ne consommera pas beaucoup d'énergie. Nous notons par ailleurs l'article de MIT, 'Energy-Efficient Algorithms'<sup>2</sup>, qui utilise des techniques spécifiques comme la subroutine réversible ou encore la restructuration des structures de données et le mouvement des pointeurs (au lieu de leur destruction et création) afin de réduire le coût énergétique d'un algorithme.

---

<sup>2</sup> Demaine, Erik D., Jayson Lynch, Geronimo J. Mirano, and Nirvan Tyagi. "Energy-Efficient Algorithms." Proceedings of the 2016 ACM Conference on Innovations in Theoretical Computer Science - ITCS '16 (2016), Cambridge, Massachusetts, USA, January 14-17, 2016, pp. 321-332.