

---

# FLOC: outil de mesure énergétique multi-composants

**Hernan Humberto ALVAREZ VALERA**<sup>1</sup>, **Franck RAVAT**<sup>2</sup>,  
**Jiefu SONG**<sup>2</sup>, **Philippe ROOSE**<sup>4</sup>,  
**Nathalie VALLES-PARLANGEAU**<sup>4</sup>

1. Domolandes, Saint Geours de marenne, France

*humberto.valera@domolandes.fr*

2. IRIT (CNRS 5505)- Université Toulouse Capitole, Toulouse, France

*prenom.nom@irit.fr*

3. LIUPPA, Université de Pau et des Pays de l'Adour, Anglet, France

*prenom.nom@univ-pau.fr*

---

## RÉSUMÉ.

Mesurer la consommation énergétique des applications est un préalable nécessaire à toute stratégie de réduction de l'empreinte écologique de l'IT. Cette consommation est directement liée à la charge générée sur différents composants matériels par des applications.

Cet article présente FLOC, un outil conçu pour mesurer et fournir des profils complets de consommation d'énergie des applications et des ensembles d'applications qui forment un système complexe. FLOC couvre des composants matériels clés tels que le CPU, la RAM, les cartes réseau et les dispositifs de stockage.

Nous avons démontré la pertinence de FLOC sur un benchmark de Data Lake.

*MOTS-CLÉS : consommation énergétique globale, outils de mesure logiciels, systèmes big data*

---

## 1. Introduction

Pour atteindre zéro émission d'ici 2050, une réduction de 55% de la consommation électrique est attendue. Concernant le numérique, l'un des leviers est lié à l'exécution des logiciels, dont la consommation énergétique augmente de 7% chaque année<sup>1</sup>. Afin de réduire efficacement cette consommation, il s'agit d'abord de la mesurer de manière pertinente. C'est-à-dire, il faut considérer les composants matériels impliqués

---

1. <https://www.digitalinformationworld.com/2020/02/the-global-energy-consumption-of-information-technologies-infographic.html>

dans l'exécution d'une application unique ou d'un ensemble d'applications formant un système complexe, tout en incluant leurs processus, sous-processus et threads.

Par exemple, concernant les Big Data Analytics, et dans le cas qui nous concerne, les Lacs de Données, les traitements dans les différentes zones impactent différemment les composants matériels. Durant la phase d'ingestion, la carte réseau gère le transfert des données, tandis que les dispositifs de stockage garantissent leur pérennité et accessibilité. Lors de la phase de stockage, ces dispositifs prennent en charge l'organisation, l'indexation, et la récupération des données. Enfin, durant la phase de nettoyage des données, le CPU est intensivement utilisé, tandis que la RAM offre un stockage temporaire pour les données en cours de traitement et les résultats intermédiaires. Ainsi, pour élaborer des méthodes d'optimisation holistiques, comment pouvons-nous mesurer l'intégralité de ce processus complexe ? Comment déterminer quelle phase est la plus coûteuse en terme énergétiques ? Quel composant matériel est le plus gourmand en énergie ?

Actuellement, deux méthodes de mesure existent : d'une part, les capteurs physiques évaluent précisément la consommation des dispositifs (ordinateurs, serveurs, etc.), mais ne distinguent pas la consommation des applications spécifiques ou celle des composants matériels distincts (CPU, RAM, etc.) (Jay *et al.*, 2023). D'autre part, des outils logiciels permettent d'évaluer la consommation d'un ensemble d'applications. Certains sont développés comme une couche au-dessus des interfaces de bas niveau des processeurs pour évaluer la consommation des processus (Noureddine, 2022 ; Petit, 2023) ou des fragments du code source (Hähnel *et al.*, 2012). Ces derniers ciblent seulement le CPU, négligeant des composants clés comme la RAM ou les périphériques d'entrée/sortie pour une analyse complète de la consommation énergétique (Alvarez-Valera *et al.*, 2024) . D'autres approches considèrent divers composants matériels mais se limitent à l'analyse d'une charge de travail spécifique pendant une période déterminée et, à notre connaissance, n'ont pas maintenu la compatibilité avec les nouvelles versions des systèmes d'exploitation. Enfin, certaines méthodes étudient la consommation de l'intégralité du trajet d'une requête client dans des environnements distribués (Anand *et al.*, 2023), mais elles ne sont pas adaptées pour analyser le comportement énergétique d'applications ou de systèmes complexes, tels que les systèmes de big data analytics, sur le long terme.

Nous proposons FLOC, un outil pour mesurer de façon complète la consommation énergétique d'applications individuelles ou des systèmes complexes, prenant en compte leurs processus, sous-processus, et threads, ainsi que divers composants matériels clés tels que le CPU, la RAM, la carte réseau, ou les dispositifs de stockage.

## 2. Fonctionnement de FLOC

FLOC repose sur la mesure de la charge induite des composants matériels lors de l'exécution d'une application, de frameworks ou d'un ensemble d'applications, ainsi que les processus, sous-processus et threads associés. FLOC se base sur des calculs spécifiques à chaque composant pour convertir ces charges en valeurs de puissance en

watts. Ce processus est répété à des intervalles définis  $i$  et sur une durée totale  $t$  spécifiée. Ainsi, FLOC calcule la puissance moyenne consommée par chaque composant pendant cette période, déterminant ainsi la consommation énergétique totale de l'application en joules (la puissance en watts multipliée par le temps en secondes). Pour le CPU, les calculs se basent sur le temps CPU alloué aux applications. Pour la RAM, ils considèrent le coût énergétique des opérations de lecture et d'écriture, alors que pour les disques et le réseau ce sont les opérations d'entrée/sortie qui sont évaluées.

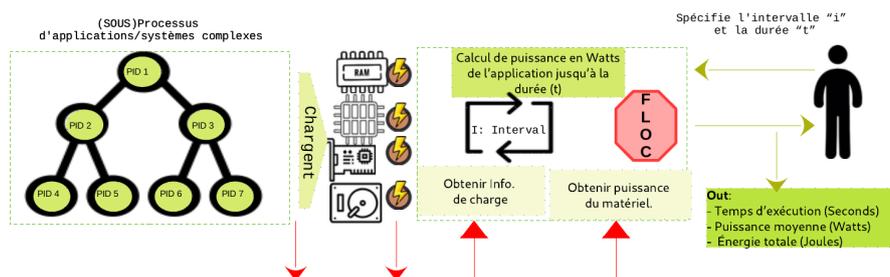


FIGURE 1. Fonctionnement de FLOC

Du point de vue de l'utilisateur (Figure 1), il saisit le nom de l'application, le PID du processus ou le nom du framework à évaluer, ainsi que l'intervalle  $i$ , la durée  $t$  et les composants d'intérêt. L'application enregistre la puissance moyenne en Watts (W) et l'énergie consommée en Joules (J) pour chacun des composants spécifiés. Si  $t$  est négatif, FLOC fonctionne indéfiniment jusqu'à la fin de tous les PIDs concernés.

### 3. FLOC en action sur un système de gestion et d'analyse de données

Nous avons choisi d'évaluer l'utilité et de démontrer la pertinence de FLOC au travers de la mesure de la consommation d'énergie d'un benchmark spécialement conçu pour évaluer les opérations Big Data dans un lac de données (Sawadogo, Darmont, 2023). C'est un système complexe dont l'exécution génère une charge sur les 4 composants matériels les plus importants, que FLOC est capable d'évaluer. Pendant sa phase d'ingestion, ce benchmark transforme de grands ensembles de données brutes (PDFs d'articles scientifiques et tables SQL) en textes indexables dans Elasticsearch. Il génère ensuite des métadonnées stockées dans Elasticsearch et MongoDB. Il optimise ces données en utilisant Neo4j pour les relations et MongoDB pour le traitement du texte, et applique des analyses de similarité avec SQLite et Neo4j.

L'expérience a consisté à exécuter et évaluer la consommation énergétique de la phase d'ingestion du benchmark, soit 18,75 Go. Cette phase comprend aussi bien l'intégration de données brutes que les mises à jour des métadonnées associées.

La figure 2 présente la consommation énergétique des opérations d'ingestion du benchmark. Elle révèle que, pour l'ingestion de données, le dispositif de stockage est le composant ayant le plus consommé d'énergie avec 197.46 J, soit 39.8% du total, suivi par la RAM à 111.95 J (22.6%) et la carte réseau à 86.39 J (17.4%). Ce qui est

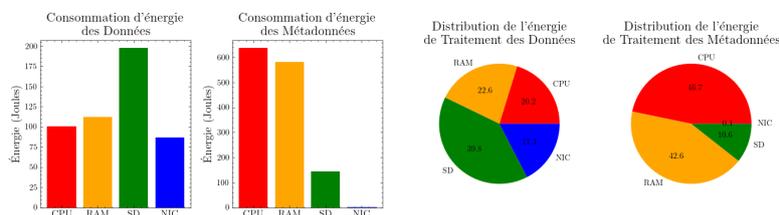


FIGURE 2. FLOC : Résultats du traitement des données et des métadonnées

prévisible étant donné que les opérations sont principalement de type E/S. Concernant le traitement des métadonnées, le CPU avec 637.27 J (46.7%) et la RAM avec 581.23 J (42.6%) sont nettement plus énergivores en raison de la complexité des opérations impliquées.

Grâce à **FLOC**, nous spécifions la consommation individuelle des composants matériels pour chacune des opérations effectuées par une application ou un système complexe. Ceci permet d'envisager des stratégies potentielles éco-responsables de répartition de charge, d'optimisation de codage, de traitement de données, etc.

#### 4. Travaux futurs

Actuellement, nous améliorons FLOC pour (i) permettre d'intégrer d'autres composants matériels et méthodes de mesure, comme RAPL pour le CPU et (ii) garantir sa compatibilité avec les différentes distributions GNU/Linux. Pour en savoir plus, vous pouvez visiter <https://github.com/humbertoAv/FLOC/tree/main>.

#### Bibliographie

- Alvarez-Valera H. H., Maurice A., Ravat F., Song J., Roose P., Valles-Parlangeau N. (2024). Energy measurement system for data lake: An initial approach. In *16th asian conference on intelligent information and database systems*. (Accepted for publication)
- Anand V., Xie Z., Stolet M., De Viti R., Davidson T., Karimipour R. *et al.* (2023). The odd one out: Energy is not like other metrics. , vol. 3.
- Hähnel M., Döbel B., Völp M., Härtig H. (2012). Measuring energy consumption for short code paths using rapl. *SIGMETRICS Perform. Eval. Rev.*
- Jay M., Ostapenco V., Lefevre L., Trystram D., Orgerie A.-C., Fichel B. (2023). An experimental comparison of software-based power meters: focus on cpu and gpu. In *2023 IEEE/ACM 23rd International Symposium on Cluster, Cloud and Internet Computing*.
- Noureddine A. (2022). Powerjoular and joularjx: Multi-platform software power monitoring tools. In *2022 18th International Conference on Intelligent Environments (IE)*.
- Petit B. (2023). *scaphandre*. Consulté sur <https://github.com/hubblo-org/scaphandre>
- Sawadogo P. N., Darmont J. (2023). Dlbench+: A benchmark for quantitative and qualitative data lake assessment. *Data & Knowledge Engineering*.