Projet Génie Logiciel :

Analyse des impacts énergétiques

Groupe 7 Equipe 34

BESSET Noé
FALL Rokhaya Yvette
GILLOT Aurélie
LESPINE Marilou
ROCHE Faustine

24 janvier 2024

Tables des matières

1 Efficience du procédé de fabrication

L'étude du processus de fabrication met l'accent sur les étapes de rédaction du code et de test qui pourraient bénéficier d'une optimisation. La mesure de la consommation électrique du processeur se présente comme un indicateur crucial lors de l'évaluation des différentes options d'optimisation. Dans notre projet, l'éditeur de texte est un outil omniprésent que nous utilisons constamment lorsque nous travaillons sur le projet GL.

En ce qui concerne l'évaluation de la consommation électrique d'une machine Intel, l'utilisation du RAPL (Limite moyenne de puissance fonctionnelle) est particulièrement utile. Le fichier /sys/class/powercap/intel-rapl/intel-rapl:0/energy_uj contient les microjoules (uJ) utilisés par le processeur. Ensuite, nous avons créé un script en python appliqué sur les données RAPL afin de convertir les microjoules en watts

Jusqu'à présent, une solution de mesure équivalente n'a pas été trouvée sur une machine AMD.

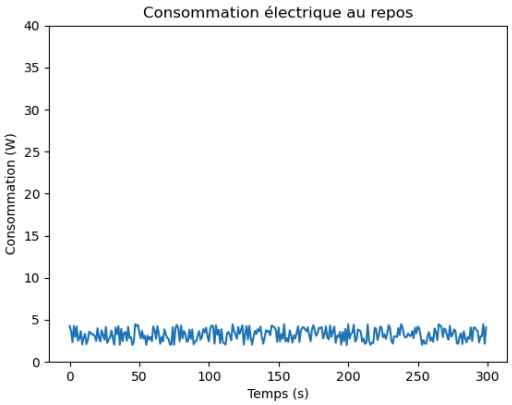
Le processus de décision entre les différentes options d'optimisation implique une évaluation des avantages et inconvénients de chaque approche. La mesure de la consommation électrique nous donne une solution d'optimisation quantifiable et concrète qui nous permet de prendre des décisions réfléchies sur la partie développement durable de ce projet.

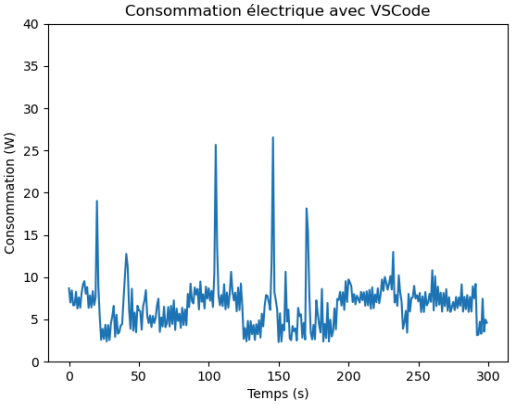
Efficience énergétique des outils de programmation

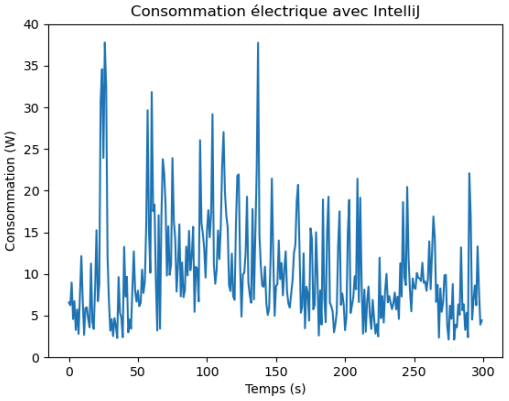
L'éditeur de texte est l'outil principal qui influence la consommation d'énergie lors de l'écriture de code. L'objectif est de choisir un éditeur de texte peu gourmand en énergie afin d'optimiser cette partie essentielle. Les éditeurs de texte maîtrisés par les membres du groupe sont les suivants : Visual Studio Code (VSCode) et IntelliJ.

Un benchmark a été établi afin de construire une comparaison objective entre ces deux éditeurs. En simulant l'utilisation de plusieurs fonctionnalités communes aux deux éditeurs, ce benchmark à pour but d'établir une courbe de consommation énergétique sur une période de 5 minutes. Ces fonctionnalités ont été choisies pour refléter des scénarios d'utilisation réalistes.

Les courbes de consommation d'énergie ont été créées en utilisant les résultats de ce test sur un ordinateur de l'ENSIMAG (DELL Precision 3430, Intel Core i5 8500). Dans le contexte spécifique de notre projet, l'analyse de ces courbes permettra de déterminer lequel des deux éditeurs, VSCode ou IntelliJ, consomme le plus d'énergie. En utilisant cette méthode, nous pouvons déterminer l'éditeur de texte à utiliser pour maximiser l'efficacité énergétique tout au long du processus de développement.







Voici les moyennes de consommation d'énergie obtenues :

Au repos : 3.29 W.
Visual Studio Code : 6.71 W
IntelliJ : 10.27 W

Des deux éditeurs de texte, VSCode est clairement le gagnant en termes d'efficacité énergétique. VSCode est le meilleur choix pour réduire la consommation d'énergie, consommant plus de 33 % moins d'énergie qu'IntelliJ. Par conséquent, il a été décidé que l'ensemble de l'équipe travaillera dorénavant sur VSCode, ce qui permettra d'économiser de l'énergie tout en maintenant une efficacité élevée dans le processus de développement.

Efficience énergétique du procédé de test

Lors du démarrage d'un test, la gestion de l'énergie est cruciale et il existe un certain nombre de méthodes qui peuvent être utilisées pour minimiser l'impact sur l'environnement.

- Premièrement, il est important de ne pas effectuer de tests inutiles, évitant ainsi de devoir répéter les tests et de répéter les tests dans des conditions similaires. Par exemple, si la condition « si » et la multiplication sont déjà évaluées séparément, il n'est pas nécessaire de tester les deux éléments en même temps. De plus, il est recommandé d'optimiser les tests en les combinant judicieusement pour vérifier les interactions entre les éléments individuels, réduisant ainsi le besoin de tests redondants.

- Lors de l'exécution de tests, il est recommandé d'exécuter uniquement les tests pertinents pour un cas spécifique.Lorsqu'une branche est poussée, il est plus économe en énergie d'exécuter uniquement la suite de tests pertinente à une partie spécifique du code que d'exécuter une "vérification mvn" générale. "mvn verify" est réservé à la fusion des branches dans master.

- De plus, la sélection de machines pour les tests de démarrage peut également contribuer à réduire l'impact énergétique. En analysant la consommation d'énergie lors de l'exécution de "mvn verify" sur différentes machines, nous pouvons déterminer quelle machine est la plus économe en énergie. Par exemple, les résultats obtenus pour la version du code sans objets montrent que l'ordinateur de Noé consomme 0,171 Wh, tandis que l'ordinateur de l'Ensimag consomme 0,316 Wh et l'ordinateur d'Aurélie consomme 0,251 Wh. Compte tenu de ces mesures, il a été décidé d'exécuter "mvn verify" sur la version avec-objet de la machine de Noé, optimisant ainsi l'efficacité énergétique du processus de test. Il convient toutefois de préciser que les ordinateurs Yvette équipés d'une architecture AMD ne peuvent pas être utilisés pour ces tests spécifiques.

2 Efficience du code produit

Un compilateur peut rendre le programme produit plus efficace et donc par la même occasion réduire l'impact énergétique indirect du compilateur lié au code compilé. Dans le cadre de notre projet, nous avons manqué de temps pour pouvoir améliorer l'efficacité du code produit mais nos pistes d'amélioration envisagés sont nombreuses :

- Constant folding : lorsque l'on a un calcul dans le programme .deca qui n'implique que des littéraux, il n'est pas nécessaire d'inclure dans le code produit l'opération car on peut très bien inclure uniquement le résultat de l'opération calculé à la compilation. Ainsi l'opération 3*3 ne sera représenté dans le .ass que comme un 9, ci qui permet une économie importante de calcul.

- Suppression de code mort : lorsqu'une portion de code à compiler ne sera pas accessible à l'exécution, il n'est pas nécessaire de l'inclure dans le code compilé. Ainsi, la partie else d'une condition if(true) ne sera pas traduire inutilement.

- Propagation de constantes : lorsqu'une variable (int, float, boolean) n'a jamais sa valeur de modifié lors de l'exécution du programme, il n'est pas nécessaire de l'enregistrer dans la pile et il est préférable de directement écrire la valeur de la variable dans le code compilé au lieu d'écrire des appelle à la pile coûteux en énergie.

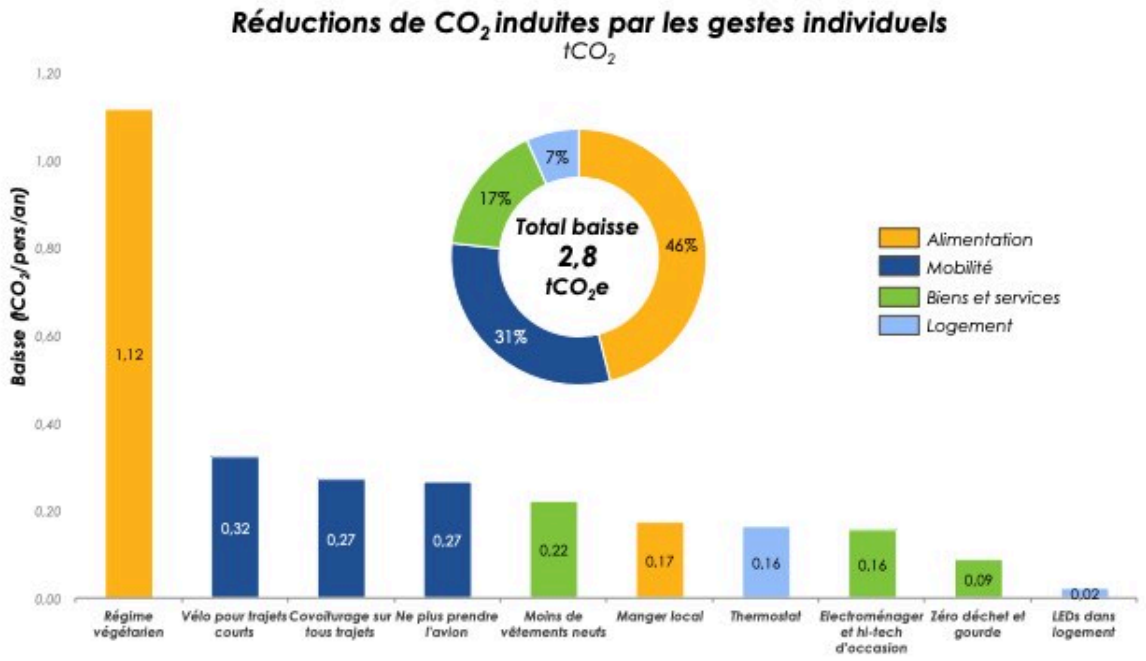
3 Utilisation du produit dans des projets à impacts

Lors de la conception du compilateur, nous avons observé qu'il était relativement polyvalent et qu'il était aisé de l'utiliser pour la sensibilisation de l'impact environnemental de chacun.

C'est pour cela que nous avons décidé de centrer notre démonstration sur cette cause. En effet, même si lors de son élaboration nous voulions au maximum adopter une attitude responsable et consciente de notre impact énergétique lors de ce projet, il nous tenait à coeur d'aller plus loin.

La démonstration montre que notre compilateur peut être un outil pour des projets à impact positif sur l'environnement par le biais de la sensibilisation des particuliers. En effet, notre application ECOHOUSE permet de promouvoir les comportements eco-responsables directement chez soi.

La technologie peut être un catalyseur puissant pour des initiatives environnementales. Ce compilateur DECA le prouve en ouvrant la voie à des possibilités créatives pour concevoir des outils et des solutions numériques axés sur l'environnement. Cela facilite le développement d'applications et de programmes liés à la durabilité et ainsi mobilise les individus vers des comportements plus respectueux de l'environnement.



A titre d'exemple, on peut observer ici les réductions de CO2 induites par les gestes individuels. Cela peut représenter une baisse totale de 2,8 tonnes de CO2 par personne. En

moyenne, un français est responsable de 9 tonnes d'émission de CO2 par an. Cela représente une réduction de presque un tiers, ce qui est non négligeable et doit être encouragé.