

Documentation d'analyse énergétique

Introduction

Dans ce bilan énergétique, on se pose deux questions principales:

- L'impact sur l'environnement, c'est-à-dire dans le cas de l'énergie, surtout le réchauffement climatique et donc les émissions de CO2.
- L'impact sur l'utilisation de ressources limitées, tel que les ressources fossiles, qui pourraient à terme disparaître ou ne plus être accessible.

On applique ces questions à trois domaines:

- Le code assembleur produit par le compilateur. C'est-à-dire l'impact énergétique de l'exécution du code produit sur la machine abstraite ima.
- Le compilateur lui-même. C'est à dire l'impact énergétique de compiler un fichier deca avec notre compilateur
- Le processus de fabrication du compilateur. C'est-à-dire l'impact de l'exécution des tests par exemple, qui est de loin l'activité la plus "lourde" énergétiquement du projet, mais également l'impact de l'utilisation des outils informatiques (ordinateur portable...).

On présentera les moyens utilisés pour évaluer l'impact environnemental de la fabrication de notre compilateur, ainsi que les moyens utilisés pour limiter cet impact.

Moyen global utilisé pour évaluer la consommation d'énergie

Le travail nécessaire pour un compilateur est essentiellement lié au cpu.

De plus, le travail fait par les programmes deca compilés est également très lié au cpu.

Pour estimer l'énergie consommée nous avons donc utilisé la commande `/usr/bin/time` qui nous permet de connaître le temps cpu de l'exécution d'un programme ainsi que [l'enveloppe thermique](#) (TDP) du processeur qui est utilisé pour les exécuter. Lorsque l'on fera des approximations on essaiera toujours de prendre l'estimation la plus haute possible par sécurité.

Quantité d'énergie consommée

Sachant que nous utilisons tous les pc de l'ensimag à distance, qui sont sous linux et qui ont tous des processeurs similaires (pour simplifier on considère qu'ils ont tous la même enveloppe thermique). On utilise alors la commande `lscpu` qui nous permet de trouver le modèle exacte du processeur.

```
[cathelib@ensipc276 gl01]$ lscpu
Architecture:          x86_64
CPU op-mode(s):        32-bit, 64-bit
Byte Order:            Little Endian
CPU(s):                4
On-line CPU(s) list:   0-3
Thread(s) per core:    1
Core(s) per socket:    4
Socket(s):              1
NUMA node(s):          1
Vendor ID:              GenuineIntel
CPU family:             6
Model:                 58
Model name:             Intel(R) Core(TM) i5-3570 CPU @ 3.40GHz
```

On visite alors le site d'intel

<https://ark.intel.com/content/www/fr/fr/ark/products/65702/intel-core-i5-3570-processor-6m-cache-up-to-3-80-ghz.html>

qui nous donne les spécifications exactes du processeur, et surtout l'information qui nous intéresse, l'enveloppe thermique est de **77 W**.

Tout au long de ce document pour obtenir la consommation énergétique en Watt heure on prend en compte le temps total d'exécution du compilateur / de la machine ima / des tests ... et on effectue le calcul suivant :

$\frac{nbSeconde}{3600} \times TDP = \text{Watt heure consommés.}$

Quantité de CO2 émise

Sachant que nous sommes tous en France pour la plupart sauf Benjamin qui est en Suisse actuellement, nous prenons la quantité de CO2 par KWh pour la France.

<https://www.rte-france.com/eco2mix/les-emissions-de-co2-par-kwh-produit-en-france#>

Sur le site de rte (Réseau de transport d'électricité), on constate que cette estimation varie entre 23 et 92g de CO2 par KWH (entre les dates du 01/12/2020 et du 20/01/2021)

Par simplification, on part donc sur l'estimation haute de 100g de CO2 par KWh

Le calcul qui sera utilisé est donc le suivant:

$CO2/KWH \times KWH = 100g \times KWH$

Prix

Pour obtenir le prix c'est assez simple, on constate que le prix de l'électricité en France est de 15 centimes par KWh.

(https://particulier.edf.fr/content/dam/2-Actifs/Documents/Offres/Grille_prix_Tarif_Bleu.pdf)

Donc le calcul qui sera utilisé est le suivant

$KWh \times 0.15$

Critique de cette estimation

Du fait que nous ne sommes pas des expert en énergie et en climat, que nous disposons de moyens (pas de wattmètre pour mesurer la consommation d'énergie exacte de l'ordinateur par exemple) et de temps limité (car la priorité est de faire le meilleur compilateur possible), nos estimations sont imparfaites et critiquables. Ci-après vous trouverez quelques critiques évidentes auxquelles nous avons pensé. Il y a très probablement d'autres aspects qui ne sont pas corrects avec notre estimation également.

Quantité d'énergie consommée

C'est probablement la partie la moins précise de notre estimation. En effet, nous ne prenons en compte que le processeur, alors qu'il y a d'autres éléments à prendre en compte (disque dur, mémoire vive, efficacité de l'alimentation...)

Quantité de CO2

La quantité de CO2 émise par toute activité informatique dépend grandement du mix électrique du pays dans laquelle l'électricité est consommée. La France est un des pays qui a le "meilleurs" mix électrique au monde (en termes d'émission de CO2). Voir par exemple en Europe

<https://www.lemonde.fr/blog/huet/2019/05/06/electricite-et-co2-le-tableau-europeen/>. Cela est dû en grande partie à l'énergie nucléaire qui est très déployée en France.

Si on se positionne dans un autre pays, par exemple l'Allemagne on est à plus de 400g de CO2 par KWh (de ce n'est pas une estimation haute, contrairement au 100g précédent)

Efficiency du code produit

Tout d'abord nous avons prêté énormément d'attention à la page [Palmarès Projets GL 2021](#) qui nous a permis de vérifier si nous n'étions pas sur le mauvais chemin. Vers le milieu du projet, nous étions vers le milieu du classement, mais à la toute fin nous avons réussi à obtenir une place dans le top 10, grâce à une optimisation des opérations entières par puissances de 2, remplacées par des shifts (instructions SHL et SHR). Cela est particulièrement utile pour le programme Syracuse42.

Efficiency du compilateur

Ci dessous nous avons listé les temps de compilation pour un programme simple en deca java et c, compilé respectivement avec notre decac, javac et gcc.

<pre>real 0m0.548s user 0m1.079s sys 0m0.127s</pre> <p>while.deca (decac)</p>	<pre>{ int i=0; while(i<100){ println(i); } }</pre>
<pre>real 0m0.086s user 0m0.042s sys 0m0.022s</pre> <p>while.c (gcc)</p>	<pre>int main(){ int i = 0; while(i<100){ printf("%d\n",i); ++i; } return 0; }</pre>

	}
<pre>real 0m0.961s user 0m2.476s sys 0m0.196s</pre> <p>While.java (javac)</p>	<pre>public class While{ public static void main(String []args){ int i = 0; while(i<100){ System.out.println(i); ++i; } } }</pre>

On remarque que gcc est de loin le plus efficace mais que decac est tout de même devant javac.

Impact énergétique de l'extension

Fonction trigonométriques

Pour voir l'impact énergétique des fonctions trigonométrique nous avons décidé de comparer les implémentations de ces fonctions avec celles de la bibliothèque standard math.h en C. Les tests ont été faits sur toutes les fonctions mais ils donnent toujours les mêmes résultats.

<pre>real 0m0.103s user 0m0.092s sys 0m0.011s</pre>	<pre>#include "Math.decah" { Math m=new Math(); println(m.cos(1.0)); //ou sin, asin, atan }</pre>
<pre>real 0m0.002s user 0m0.001s sys 0m0.001s</pre>	<pre>#include <stdio.h> #include<math.h> int main(){ printf("%lf\n",cos(1.0)); //ou sin, asin, atan return 0.0; }</pre>

On remarque très vite le facteur 100 de différence.

Évidemment il ne faut pas oublier que notre code est exécuté dans un simulateur (ima), mais la conclusion est qu'il faut éviter d'utiliser cela en production. Des tests sur un vrai processeur seraient intéressants.

ulp

Comme expliqué dans la documentation sur l'extension, pour calculer l'ULP nous avons utilisé la définition suivante (qui correspond à ce qui est demandé dans le sujet) :

L'ulp de deca doit se comporter exactement comme Math.ulp en java

Hors, contrairement à java il est impossible d'accéder à la représentation IEEE754 directement. Plus précisément, il est impossible d'accéder aux champs (c'est à dire la mantisse, le signe et l'exposant) de la représentation binaire individuellement. Nous avons donc décidé de retrouver cette représentation avec un algorithme (ce n'est pas très compliqué), qui est certes simple, mais qui a un coût (en temps processeur et donc en énergie). Cela est dommage car l'information est déjà présente en mémoire vive mais cela nous semble inévitable.

Efficiency du procédé de fabrication

Impact énergétique des tests

Energie consommée par un test complet

Ci dessous, un tableau récapitulatif des mesures avec la commandes time

<pre>real 4m41.719s user 9m58.632s sys 1m7.464s</pre>	./src/test/script/master-all-test.sh
<pre>real 0m3.714s user 0m3.858s sys 0m0.610s</pre>	mvn clean
<pre>real 0m12.094s user 0m23.939s sys 0m1.287s</pre>	mvn compile (après clean)
<pre>real 0m7.502s user 0m17.642s sys 0m0.975s</pre>	mvn compile (pas de clean)
<pre>real 0m13.030s user 0m19.680s sys 0m1.424s</pre>	mvn test-compile (après clean)

<pre>real 0m13.134s user 0m20.155s sys 0m1.448s</pre>	mvn test-compile (pas de clean)
---	---------------------------------

Nous ne prendrons en compte que l'exécution de tous les tests (script master-all-test), car le reste est négligeable en comparaison.

$$4m41s = 281s$$

$$\frac{281}{3600} \times 77 = 6 \text{ Watt heure}$$

Prix d'un test complet

$$0.006 \times 0.15$$

C'est à dire à 0,0009 €.

Emission de CO2 d'un test

$$100g \times 0.006 = 0.6g$$

Impact énergétique global des tests

Si on prend l'hypothèse, excessive, que notre groupe va faire tourner tous les tests (avec le script master-all-test) 10000 fois alors on arrive à:

$$0.6g \times 10^4 = 6kg$$

Impact énergétique global du projet

Pour obtenir une estimation (imprécise et haute) de l'impact énergétique global du projet on prend les deux hypothèses suivantes:

-l'activité la plus lourde est l'exécution des tests, cette activité est indépendante du reste du travail

-le reste du temps la consommation électrique des ordinateurs des membres du projet est dans la moyenne française.

Tout d'abord on calcul la quantité d'heures travaillées:

4 semaines de 35h avec 5 personnes.

$$4 \times 5 \times 35 = 700 \text{ heures travaillées.}$$

Comme le projet est en distanciel, on estime que toutes les heures travaillées ont été passées devant un ordinateur.

La plupart des membres du projet ont un ordinateur portable. Une estimation haute de la consommation d'un ordinateur portable est 100 Watt .

cf(<https://www.happ-e.fr/actualites/les-eco-gestes/quelle-est-consommation-electrique-dun-pc>). Sachant que en réalité on utilise deux ordinateurs pour travailler (le client, notre ordinateur + le pc ensimag ensipcxxx¹ auquel on est connecté à distance), on peut doubler cette estimation à 200Watt.

On obtient donc.

¹ Bien qu'on puisse questionner la "responsabilité" de notre projet pour ces émissions là car ces ordinateurs sont de toute façon allumés en permanence

$700 \times 200 = 140000$ Watt heures ou 140KWh

c'est à dire

$100g \times 140 = 14000g$ Ou 14Kg de CO2.

Pour obtenir l'estimation de l'impact total on additionne l'impact des tests + l'impact de l'utilisation des ordinateurs.

$14 + 6 = 20$ Kg de CO2

Comparé à l'émission moyenne par habitant

https://fr.wikipedia.org/wiki/Liste_des_pays_par_%C3%A9missions_de_dioxyde_de_carbone_par_habitant

Sachant que, en France, un habitant émet en moyenne 5 tonnes (tonnes métrique c'est à dire 1000kg) de CO2 par an alors on peut voir que c'est assez faible. Pour être précis, cela représente

$$\frac{20}{5000} = 0.004 = 0.4\%$$

des émission d'un français moyens

Toutefois si on généralise cela à toutes les équipes (57)

$$57 \times 20 = 1140kg$$

Soit plus d'1 tonne de CO2 ce qui n'est pas négligeable.

Prix total en électricité du projet

On multiplie la quantité d'électricité par le nombre d'équipes ainsi que par le prix au KWh.

$$140KWh \times 57 \times 0.15 = 1197 \text{ €}$$

C'est une estimation haute mais ça reste beaucoup d'argent.

Moyens mis en oeuvre pour diminuer l'impact énergétiques sans nuire à la qualité de la validation

Tests lourds

Certains tests sont extrêmement lourds et il serait très long et très consommateur en énergie de faire ces derniers sur chaque fichier deca.

Par exemple, le test de l'option -r est quasiment impossible à faire sur toute la base de test et pas seulement pour des raisons énergétiques, mais aussi pour des raisons de temps.

Pour cette raison nous sommes limité à 1 fichier sur ce test

Fréquences des tests

Nous nous sommes engagés à limiter la fréquence avec laquelle nous exécutons le script master-all-test.sh (qui exécute tous les tests). Quand nous voulions tester une fonctionnalité spécifique, nous utilisions uniquement le test approprié et non pas master-all-test.

Conclusion

D'après nos expérimentations et calculs, le moyen le plus efficace de réduire son impact énergétique serait de diminuer le nombre d'heures travaillées, ou du moins, le nombre d'heures où les ordinateurs sont allumés. Une suggestion serait peut être de ne pas laisser les ordinateurs de l'ensimag (ensipcxxx) allumés en permanence. Toutefois cela est compliqué dans la période actuelle car nous sommes beaucoup à les utiliser pour travailler à distance. Nous pouvons toutefois conclure que l'impact énergétique de notre projet est assez minime en lui même avec un certains niveau de confiance², mais nous réalisons que le projet dans son ensemble (c'est-à-dire toutes les équipes) produit une quantité non négligeable de CO2.

² Même si notre estimation était fausse d'un facteur 10 alors ça nous amènerait à 200kg de CO2, ce qui reste faible.