

# Bilan sur l'impact environnemental

Antoine Turkie, Augustin Chenevois, Aurélien Delory, Mihaja Razafimahefa, Sarah Ramaharobandro

# Table des matières

<b>Introduction.....</b>	<b>2</b>
<b>L'énergie informatique et matérielle : l'environnement et le numérique.....</b>	<b>2</b>
Les principales sources de consommation d'énergie.....	2
Les utilisateurs et leurs équipements informatiques.....	2
Les centres de données.....	3
Les services et applications numériques.....	3
<b>La consommation d'énergie selon les langages de programmation.....</b>	<b>4</b>
<b>La consommation énergétique globale de notre projet.....</b>	<b>5</b>
L'équilibre qualité et respect environnemental du projet.....	5
<b>Estimation de la consommation énergétique.....</b>	<b>6</b>
1) Le code java.....	6
2) Les tests.....	7
3) Les ordinateurs.....	7
4) Les écrans.....	7
<b>Bilan critique.....</b>	<b>8</b>

## **Introduction**

Dans ce document, nous allons parler de l'impact énergétique de notre projet.

Tout projet consomme de l'énergie sous différentes formes :

- de l'énergie humaine (énergie nécessaire à l'accomplissement des tâches)
- de l'énergie informatique et matérielle (consommation des serveurs informatiques)
- de l'énergie liée au déplacement pour se rendre au lieu de travail

Lors de ce projet, nous pouvons essentiellement tenter d'optimiser l'énergie informatique et matérielle... Mais aussi celle liée à nos déplacements ! Nous nous sommes donc engagés à tous venir à pied lors de ces quatre semaines de projet Génie Logiciel.

## **L'énergie informatique et matérielle : l'environnement et le numérique**

### **Les principales sources de consommation d'énergie**

Aujourd'hui, le numérique est responsable de 4% des émissions de gaz à effet de serre. Cette pollution est causée par trois sources principales :

### **Les utilisateurs et leurs équipements informatiques**

L'utilisation intensive des équipements informatiques comme les ordinateurs personnels entraîne une consommation d'énergie significative et une demande en électricité croissante. En outre, l'obsolescence rapide des appareils et la société de surconsommation pousse les consommateurs à changer régulièrement d'appareils et à exploiter de plus en plus les matières

premières de la Terre. Des déchets électroniques parfois non recyclés sont alors générés en grande quantité.

## **Les centres de données**

Le volume de données numériques créées ou répliquées à l'échelle mondiale a été multiplié par plus de trente au cours de la dernière décennie, passant de 2 zettaoctets en 2010 à 64 zettaoctets en 2020.

Toutes ces données sont exploitées et stockées par des Data Centers qui ont besoin d'une énergie considérable pour fonctionner. La température doit notamment y être optimale ce qui entraîne une utilisation importante des systèmes de refroidissement.

## **Les services et applications numériques**

Avant d'être stockées, les données doivent être transférées. Et l'utilisation croissante de services en ligne, de streaming vidéo, et d'applications cloud génère des transferts massifs qui contribuent à la consommation d'énergie des infrastructures réseau.

Face à toutes ces problématiques, il est devenu essentiel d'étudier et d'améliorer la consommation énergétique du numérique.

# La consommation d'énergie selon les langages de programmation

Six chercheurs de trois universités portugaises ont ainsi réalisé une étude intitulée Energy Efficiency Across programming Languages visant à évaluer la consommation énergétique de 27 langages de programmations.

Pour cela, chaque langage a été évalué sur 10 problèmes de référence. Voici les résultats obtenus :

	Energy
(c) C	1.00
(c) Rust	1.03
(c) C++	1.34
(c) Ada	1.70
(v) Java	1.98
(c) Pascal	2.14
(c) Chapel	2.18
(v) Lisp	2.27
(c) Ocaml	2.40
(c) Fortran	2.52
(c) Swift	2.79
(c) Haskell	3.10
(v) C#	3.14
(c) Go	3.23
(i) Dart	3.83
(v) F#	4.13
(i) JavaScript	4.45
(v) Racket	7.91
(i) TypeScript	21.50
(i) Hack	24.02
(i) PHP	29.30
(v) Erlang	42.23
(i) Lua	45.98
(i) Jruby	46.54
(i) Ruby	69.91
(i) Python	75.88
(i) Perl	79.58

La valeur en énergie est relative. Ainsi, Python consomme en moyenne 75.88 fois plus que le langage C.

Et comme nous pouvons le constater, les langages compilés (C, C++, Rust...) sont souvent les plus efficaces énergétiquement que les langages interprétés (Python, Perl...).

En effet, leurs exécutions demandent souvent plus de temps et par conséquent est plus gourmande en énergie même si ce n'est pas la seule raison.

Il faut aussi souligner que le programmeur a aussi un rôle important et qu'il doit écrire un code le plus efficace possible.

Enfin, d'après cet article, les langages compilés ont consommé en moyenne 120 J pour un temps de 5103 ms.

# **La consommation énergétique globale de notre projet**

## **L'équilibre qualité et respect environnemental du projet**

Le critère de qualité du rendu final était pour nous la plus grande priorité. Par conséquent, nous avons parfois négligé nos efforts concernant l'aspect environnemental.

C'est notamment le cas en ce qui concerne les tests. Pour garantir la robustesse du code, nous avons créé un script permettant de lancer tous les tests en même temps. Ainsi, à chaque modification, nous pouvions vérifier qu'il n'y avait pas d'effet de bord. Cependant, cela demandait forcément plus d'énergie.

Par contre, nous avons aussi fait des scripts permettant de tester uniquement chaque partie (syntaxe, context, codegen) pour ainsi limiter le nombre de tests et augmenter notre efficacité.

A la fin du projet, nous avons 437 tests et le script pour les lancer prenait plus de 2 minutes.

En outre, nous nous sommes aperçus à la fin du projet que lorsque nous exécutions `mvn test-compile`, `mvn-compile` s'exécutent aussi. Par conséquent, nous exécutions `mvn-compile` deux fois à chaque fois.

## **Estimation de la consommation énergétique**

Identifions chaque source de consommation d'énergie et leur impact dans le projet.

### **1) Le code java**

Supposons que :

- Le langage Java consomme comme la moyenne des langages à machine virtuelle indiqués dans l'article
- Une compilation du code java prend 7s
- Chaque personne du groupe compilait le code java 25 fois par jours

### **2) Les tests**

Supposons que :

- Les tests consomment la même quantité d'énergie que la compilation
- L'algorithme pour lancer les tests prenait 1min30s en moyenne dans le projet
- Chaque personne du groupe lançait la commande de test 15 fois jours

### **3) Les ordinateurs**

Supposons que :

- Un ordinateur consomme 300 Wh par heure
- Chaque personne travaillait 9 heures par jour, samedi et dimanche compris (ce qui est très réaliste heureusement/malheureusement)

### **4) Les écrans**

Supposons que la consommation d'un écran est de 39.5 kWh par an.

Enfin, supposons que le projet ait duré 21 jours. Après calcul, toutes les sources sont négligeables devant le fonctionnement des ordinateurs et nous avons consommé pendant 21 jours environ  $3 \cdot 10^5$  wh.

Cela représente environ l'énergie d'un réservoir (45 litres) d'essence.

## **Bilan critique**

Le projet Génie Logiciel consomme beaucoup d'énergie. Mais si l'on regarde bien les chiffres, la principale source de consommation d'énergie vient des ordinateurs. Et nous ne pouvons pas agir sur celle-ci.

Cependant, si tous les groupes faisaient des efforts, l'impact sur l'environnement ne serait peut-être pas si négligeable. Il est donc important que chacun trouve des solutions. C'est pour cela que nous avons créé des scripts de vérifications propres à chaque partie ou que nous avons décidé de tous venir à pied pendant ces quatre semaines et essayer d'optimiser le code au maximum afin de réduire le temps de calculs.