



Université Claude Bernard Lyon 1



PROJET D'OUVERTURE À LA RECHERCHE

-

DÉCROISSANCE NUMÉRIQUE : QUELS ÉCOSYSTÈMES LOGICIELS POUR L'INFORMATIQUE FRUGALE ?

-

CAHIER DES CHARGES

STAVIS THOMAS, PETIT LUCAS

Encadré par : MOY MATTHIEU, SALAGNAC GUILLAUME, MOREL LIONEL

Mots clés : DURABILITÉ, SOBRIÉTÉ NUMÉRIQUE, LANGAGES DE PROGRAMMATION, COMPILATION,
INFORMATIQUE MINIMALISTE OU EMBARQUÉE

Master 1 Informatique

Université Claude Bernard Lyon 1 – Site la Doua

1^{er} novembre 2022

Table des matières

1	Mise en contexte	1
2	Objectifs et besoins	1
3	Travail à réaliser	2
3.1	Préalable	2
3.2	Réalisation	3
4	Pistes à approfondir	3
4.1	Multi threading	3
4.2	Provenance des coûts	3
5	Calendrier	3

1 Mise en contexte

Les émissions de kgCO_2eq causées par l'exploitation de matières premières, la production et l'utilisation d'objets complexes, ainsi que le traitement de ces objets en fin de vie, induisent une augmentation de la concentration en gaz à effet de serre (GES) de l'atmosphère, qui est un des principaux vecteurs d'accroissement du réchauffement climatique et de toutes les conséquences néfastes qui lui sont associées. De plus, la planète comporte une quantité de ressources limitée, alors que le nombre d'objets rattachés au numérique connaît une croissance exponentielle qui ne peut être soutenue que par l'extraction de ressources rares et polluantes comme le cuivre, or, etc. C'est dans ce contexte de régulation des GES et de limitation de l'exploitation de ressources, pour éviter ou diminuer les conséquences néfastes, que nous nous penchons sur l'impact du numérique.

En effet, selon The Shift Project, en 2019, le numérique est responsable de 4 % des émissions mondiales de GES et cela continue de croître de 2 % an. Cela est en partie dû à la complexité des objets numériques : par exemple, la fabrication et l'utilisation d'un smartphone de 150 g émettent $85\text{kgCO}_2\text{eq}$. Or le numérique est un vaste champ dans lequel un très grand nombre de facteurs se distribuent les émissions. On les sépare fréquemment en 3 parties, utilisateurs, centres et réseaux, chacune divisée en 2 sous-parties, fabrication et utilisation.

Nous allons nous intéresser à la partie utilisation par les utilisateurs, plus précisément aux langages de programmation, environnements et technologies de développement utilisés pour créer les applications numériques.

Nous nous concentrerons sur la partie utilisation de mémoire vive (RAM) lors de la compilation, de l'interprétation et de l'exécution des programmes. Cela a pour but de définir quels sont les moyens de programmer qui nécessitent le moins de mémoire, donc qui nécessitent des terminaux avec moins de mémoire installée, donc qui nécessitent moins de matière première pour être construits, et donc qui ont une empreinte moins importante à la fabrication.

Ce travail s'inscrit dans un cadre où l'on prend le problème par un des fondamentaux du numérique, la RAM, car celle-ci va de pair avec les usages les plus simples des applications numériques.

2 Objectifs et besoins

Dans ce sujet d'ouverture à la recherche, la thématique étant vaste et le temps limité, nous essaierons à notre échelle de déterminer les technologies qui semblent être tenables dans le futur.

Tout d’abord, nous avons besoin d’outils de mesure précis en mémoire et en temps de la RAM lors des étapes qui nous intéressent, ainsi que d’une méthode de mesure efficace.

Nous avons besoin, aussi, de définir l’environnement de travail dans lequel nous allons effectuer ces mesures, pour s’assurer de la précision, de la reproductibilité et de l’écart de biais.

Une fois ces prérequis atteints, nous essaierons de réaliser ces objectifs :

- Déterminer une fonction de coût.
- Déterminer les langages, environnements et technologies à mesurer.
- Déterminer les programmes sur lesquels lancer les mesures.
- Compiler, interpréter et visualiser les données.

3 Travail à réaliser

3.1 Préalable

Afin d’atteindre les objectifs cités précédemment, il nous faudra dans un premier temps effectuer des recherches bibliographiques pour déterminer ce qui a déjà été fait dans le domaine.

Nous pourrions ainsi les prendre en compte dans nos expériences, voire les reproduire, et vérifier les résultats obtenus ainsi que la qualité de notre méthode.

Les tests variants selon la machine sur laquelle ils sont effectués, il faudra déterminer l’environnement d’expérience qui serait le plus pratique entre :

- Une machine virtuelle, afin de pouvoir spécifier la configuration de la machine, mais l’émulation pourrait introduire des facteurs de confusion.
- Une machine physique unique, mais la praticité est limitée.

Dans les 2 cas, nous documenterons la configuration de nos supports de tests (OS, architecture, CPU/GPU, RAM).

Nous observerons également la thématique de 3 points de vue différents, qui représentent nos cas d’utilisations futurs. Ainsi, nous regarderons ce qui est nécessaire pour :

- Un utilisateur lambda, n’ayant besoin que d’exécuter les programmes.
- Un développeur, un utilisateur devant compiler des programmes moyennement complexe.
- Un développeur toolchain, c’est-à-dire un utilisateur ayant la nécessité de compiler un compilateur d’un langage ainsi que l’ensemble des composants entrant dans l’utilisation de ce langage.

Enfin, pour déterminer les langages les plus valides pour une utilisation future, nous avons également besoin de déterminer les variables que nous voulons analyser (quantité de mémoire utilisée, temps d’exécution, etc.) afin de déterminer une fonction de coût dans un cas donné.

Concernant ce coût, nous ferons la distinction entre la compilation, qui est propre aux développeurs, et l’exécution ou l’interprétation qui sont nécessaires à tous.

Nous avons également en tête que les résultats obtenus seront marqués par l’implémentation que nous aurons faite. Pour cela, nous essaierons au mieux de les rendre similaires en termes de structure et de complexité d’un langage à l’autre.

Enfin, il est clair que nos résultats auront toujours une marge d’incertitude due aux nombreux facteurs qui entrent en jeu dans la compilation et l’exécution d’un programme. Nous espérons toutefois avoir au moins un ordre de grandeur du coût de chaque langage.

3.2 Réalisation

Nous pourrions alors effectuer différents tests qui sont présentés ci-dessous, avec cette matrice à 2 entrées. On y retrouve les langages candidats ainsi que les programmes de tests.

	Hello World	Sudoku Solver	Toolchain
C			
Java			
Forth			
Rust			

- Hello World : programme très simple représentant les performances de base d'un langage.
- Sudoku Solver : représente les programmes plus complexes nécessitant des traitements.
- Toolchain : dépendances nécessaires à la compilation d'un compilateur du langage, l'utilisation d'un langage impliquent de connaître le coût de sa toolchain.

4 Pistes à approfondir

4.1 Multi threading

Le multi threading est une composante intéressante de l'informatique, mais elle n'est pas supportée par tous les langages (C, Forth), et peut ne pas être adaptée en termes de consommation de ressources.

C'est un point que nous gardons en tête si le temps nous permet de l'approfondir, notamment en différenciant 2 de ses aspects : le parallélisme et la concurrence.

4.2 Provenance des coûts

Dans le cas où certains langages seraient plus gourmands que d'autres, il pourrait être intéressant de déterminer quels éléments de ces langages sont responsables de cette consommation plus importante.

Ces éléments de réponse pourraient nous aider à définir un peu mieux ce que seraient les outils de développement de demain.

5 Calendrier

En supposant un rendu de mi-projet fin janvier, nous prévoyons :

- En 1ère moitié de projet :
 - * D'avoir un environnement de test ainsi que des outils de mesure.
 - * De sélectionner les langages de programmation ainsi que les programmes à tester.
 - * De tester nos expériences avec un programme du type "Hello World!" pour chaque langage.
 - * D'écrire et tester des programmes moyens.

- En 2ème moitié de projet :
 - * De s'intéresser à la toolchain de chaque langage.
 - * D'approfondir le sujet si possible.
 - * De rédiger le rapport final et se pencher sur la vidéo de vulgarisation.
- En tâche de fond :
 - * De sélectionner et lire des articles permettant de stimuler nos recherches.