Institut Universitaire de Technologie Lyon 1
Département Informatique
43 Boulevard du 11 Novembre 1918 - 69622 Villeurbanne

Rapport de stage - 2eme année
DUT Informatique

Simuler l'empreinte environnementale des centres de données

9 Avril - 15 Juin 2018

Laboratoire de l'Informatique du Parallélisme
École Normale Supérieure
46 Allée d'Italie - 69364 Lyon

Maître de stage
Laurent Lefevre

Étudiant
Bastien Marsaud

Responsable pédagogique
Hamamache Kheddouci

Fiche technique

Le Laboratoire de l'Informatique du Parallélisme

Le Laboratoire de l'Informatique du Parallélisme est un laboratoire de recherche situé sur le site Monod de l'École Normale Supérieure de Lyon. Il regroupe 57 membres permanents, 20 membres temporaires et entre 40 et 50 doctorants autour de sujets très larges liés à l'informatique.

Le sujet du stage

Le sujet du stage est de concevoir un simulateur d'empreinte environnementale des centres de données. Ce stage à lieu dans le cadre d'un projet avec l'Institut d'aménagement et d'urbanisme de la région Île-de-France et l'Ecole d'architecture de la ville et des territoires. Ainsi il permettrait à terme d'aider les architectes dans la construction des centres de données et d'aider les urbanistes dans leur intégration sur le territoire.
L'une des perspective de ce projet serait d'aider à la construction de nouveaux centre de données en Île-de-France afin de répondre aux besoins massifs en traitement de données que nécessiterons les Jeux Olympiques 2024.

L'environnement du stage

Dans le cadre de ce stage je suis intégré au Laboratoire de l'Informatique du Parallélisme dans l'équipe AVALON. Tous les membres de l'équipe sont soit des chercheurs, soit des ingénieurs de recherche, soit des doctorant en informatique. Un camarade de ma promotion **Lucas Besnard** est lui aussi en stage dans l'équipe AVALON mais nous ne travaillons pas sur le même sujet.
Je travaille seul sur le projet, **Laurent Lefèvre**, mon maître de stage est bien entendu présent pour me donner les consignes, m'aiguiller et m'épauler dans ma réflexion mais ne participe pas au développement.

L'environnement de travail

La laboratoire possède des ordinateurs portables, mais comme ils ne sont pas très performants on m'a conseiller d'utiliser mon ordinateur personnel. La laboratoire m'a cependant fourni un deuxième écran.

Comme je n'avais aucune contraintes aux niveaux des technologies j'ai décidé d'utiliser celles avec lesquelles j'étais le plus à l'aise. Le projet en lui-même est développé en JAVA en utilisant la technologie JavaFX pour l'interface graphique, j'utilise Maven pour la gestion des librairies ainsi qu'Eclipse en tant qu'IDE. Pour versionner le code source j'utilise le protocole GIT couplé à un repository privé sur GitHub. Il était en effet compliqué de me créer un repository sur la plateforme interne à cause de formalités administratives.

Méthode de travail

Pour le bon déroulement du projet il était indispensable de faire une recherche bibliographique conséquente avant de commencer la phase de développement afin d'assimiler un certains nombre de notions spécifiques.

Remerciements

Je tiens à remercier toutes les personnes qui ont contribué de près ou de loin à la réussite de mon stage.

Je remercie tout d'abord **Mme. Jocelyne Debouté**, enseignante en expression communication et responsable des stages à l'IUT Lyon 1, qui fait un travail remarquable dans la recherche et le partage d'offres de stage. Grâce à qui j'ai pu postuler à ce stage et qui m'a beaucoup aidé lors de la validation de ma convention.

Je remercie chaleureusement mon maître de stage **M. Laurent Lefèvre**, chercheur à l'INRIA, membre de l'équipe AVALON au Laboratoire de l'Informatique du Parallélisme pour sa confiance et l'indépendance qu'il m'a accordée, mais surtout pour ses conseils avisés et sa bonne humeur à toute épreuve.

Je remercie **M. Sylvain Maurin**, mon voisin de palier, chercheur à l'Institut des Sciences Cognitives, **M. Cristophe Jaloux**, enseignant en mathématiques à l'IUT Lyon 1, **Mme Aude Joubert**, enseignante en mathématiques à l'IUT Lyon 1 et **M. Pierre-Antoine Champin**, enseignant-chercheur à l'IUT Lyon 1 pour m'avoir permis, à leurs manières, d'assister à la conférence EcoInfo *Que deviennent nos déchets électroniques ?* du 6 Avril dernier à Grenoble dans laquelle mon maître de stage, Laurent Lefèvre, est intervenu.

Je remercie **M. Issam Raïs**, avec qui je partageais mon bureau, doctorant sous la supervision de Laurent Lefèvre, pour sa bonne humeur entre deux lignes de thèse.

Je remercie **Mlle. Dorra Boughzala**, doctorante sous la supervision de Laurent Lefèvre pour avoir pris le temps de m'explique en détail le fonctionnement et les subtilités de la plateforme *Grid' 5000*.

Enfin, je remercie mes camarades de promotion **M. Simon Lecutiez**, **M. Valentin Gilles**, **M. Tom Befieux** et **M. Alex Pertuiset** pour ces deux années de DUT Informatique ainsi que pour leurs conseils avisés durant la rédaction et leur relecture de ce rapport de stage.

Sommaire

Fiche technique **1**

Remerciements **2**

Introduction **5**

1 L'environement du stage **6**

1.1 Le laboratoire et ses relations institutionnelles 6

1.1.1 Présentation générale 6

1.1.2 Organisation générale 6

1.1.3 Des métiers variés 7

1.1.4 Une production conséquente 9

1.2 L'équipe Avalon en détails 10

1.2.1 Les membres de l'équipe 10

1.2.2 Le contexte de création 11

1.2.3 La vision de l'équipe 11

1.3 Mon environement au sein du laboratoire 12

1.3.1 Les horaires de travail 12

1.3.2 Les locaux 13

1.3.3 L'environement matériel 13

1.3.4 L'environement technologique 13

1.3.5 L'environement humain 14

1.3.6 Les groupes de travail 15

2 La mission du stage **16**

2.1 Présentation de la mission 16

2.1.1 Le contexte 16

2.1.2 Les enjeux 17

2.1.3 L'outils demandé 17

2.2 La recherche bibliographique 18

2.2.1 Power Usage Effectiveness (PUE) 18

2.2.2 Carbon Usage Effectiveness (CUE) 19

2.2.3 Data Center energy Productivity (DCeP) 19

2.2.4 Green Energy Coefficient (GEC) 20

2.2.5 Energy Reuse Factor (ERF) 20

2.2.6 Efficacité énergétique de l'équipement (EEE) 20

2.2.7 KPI Global DCEM 21

2.2.8 Indice d'Efficacité du Transport Électrique 22

2.2.9 Indice de conformité de la température extérieure (CTE) 23

2.2.10 Indice de Pertinence Géographique 24

2.3 La développement 24

2.3.1 Le choix des outils 24

2.3.2 L'architecture générale . . . 25
2.3.3 L'architecture des contrôleurs . . . 26
2.3.4 La notion de simulation . . . 26
2.3.5 La gestion de la langue . . . 27
2.3.6 La réflexion autour de l'ergonomie . . . 27
2.3.7 Les éléments d'interface personnalisés . . . 28
2.3.8 La gestion des bases de données . . . 29
2.3.9 Le calcul des indicateurs . . . 29
2.3.10 La génération du rapport . . . 30
2.3.11 Créer un fichier exécutable . . . 30
2.3.12 Faire face aux bugs des librairies . . . 31

3 Mon expérience de stage **32**
3.1 Bilan du travail effectué . . . 32
3.1.1 Vue générale des fonctionnalités . . . 32

Conclusion **34**

Glossaire **35**

Table des figures **37**

Liste des tableaux **37**

Sources **38**

Annexes **40**

Introduction

Du 9 Avril au 15 Juin 2018 j'ai effectué mon stage de fin de deuxième année de DUT Informatique à l'IUT Lyon 1 au Laboratoire de l'Informatique du Parallélisme (LIP). Le Laboratoire de l'Informatique du Parallélisme est un laboratoire de recherche en informatique situé à l'ENS Lyon et regroupant des chercheurs, des ingénieurs et des doctorants autour de problématiques comme l'arithmétique en informatique, les architectures distribuées, l'optimisation des réseaux et des ressources ou encore l'analyse de la compilation. Il a permis, depuis l'année 2000, de mettre au jour environ 2500 publications.

Durant ma recherche de stage, même si j'utilisais des plateformes de recherche d'emplois en ligne, cette offre m'est parvenu via notre enseignante responsable des stages qui en envoyait régulièrement par mail. Cette offre a tout particulièrement attiré mon attention pour plusieurs raisons. Tout d'abord, il s'agissait de la seule opportunité qui permettait de travailler dans un laboratoire de recherche, les autres offres étant majoritairement des entreprises privées. Le milieu de la recherche m'a toujours attiré et comme j'essaye de plus en plus de l'intégrer à mon projet professionnel, cette offre me semblait une bon moyen d'y parvenir. Ensuite, les deux sujets proposés par l'offre portaient sur des problématiques environnementales, étant très attiré par ces questions depuis mon plus jeune âge et ayant suivi mon cursus de lycéen dans un lycée agricole j'étais très enthousiaste à travailler dans ce domaine. Ainsi ce stage m'apparaît comme une bonne opportunité de découvrir le monde de la recherche et de travailler sur un projet complexe, d'utilité publique sur des problématiques qui m'intéressent.

Ma mission durant ce stage est de concevoir un simulateur informatique d'empreinte environnementale des centres de données afin d'aider des architectes et des urbanistes dans leur démarche d'intégration des centres de données sur notre territoire. Pour y parvenir je devrai tout d'abord m'inscrire dans une démarche de recherche bibliographique poussée avant de pouvoir espérer commencer le développement.

TODO : Modifier le plan si il a changé
Dans ce rapport de stage, je présenterai tout d'abord l'environnement du stage : le laboratoire et ses relations institutionnelles, l'équipe AVALON dont je fais partie ainsi que ma place au sein de cette organisation. Ensuite j'expliquerais en détail ma mission, son contexte, les enjeux qu'elle porte, son but ainsi que les étapes que j'ai suivi pour y parvenir. Enfin je terminerai par détailler mon expérience durant ce stage, ce que j'ai appris et découvert techniquement et humainement.

1 L'environement du stage

L'environement de travail d'un stage dans un laboratoire de recherche est sans doute très différent d'un stage plus classique dans une entreprise privé. En effet, un laboratoire de recherche s'inscrit dans une organisation complexe et possède de nombreuses relations avec d'autres organisations ou d'autres personnes.
Dans cette partie nous présenterons tout d'abord ce qu'est le Laboratoire de l'Informatique du Parallélisme ainsi que ces relations avec d'autres institution, ensuite nous présenterons l'équipe Avalon dont je fais partie, et enfin nous terminerons par présenter la place que j'occupe au sein du laboratoire.

1.1 Le laboratoire et ses relations institutionnelles

Le Laboratoire de l'Informatique du Parallélisme (abrégé LIP) est un laboratoire de recherche en informatique situé principalement sur le site Monod de l'École Normale Supérieure de Lyon.

1.1.1 Présentation générale

Créé dans les années 80, le Laboratoire de L'Informatique du Parallélisme devient Unité de Recherche Associé (URA) de l'ENS Lyon en 1989, puis Unité Mixte de Recherche (UMR) en 1999 qui sera complété par l'Université Claude Bernard Lyon 1 en 2003. Le laboratoire est très tôt épaulé par l'Institut national de recherche en informatique et en automatique (Inria) qui est le principal acteur de la recherche en informatique et mathématique en France depuis 1967. Ainsi le laboratoire héberge plusieurs équipes-projets communes avec l'Inria. [9]

Le laboratoire compte 57 enseignant titulaires et chercheurs, entre 40 et 50 doctorants ainsi qu'une vingtaine de personnes sur des postes non permanents. L'équipe d'administration et l'équipe technique quant-à-elles sont épaulés par 12 ingénieurs.

Le LIP possède une bonne visibilité de ses équipes au niveau national et de plusieurs de ses membres au niveau international. Il occupe une place central dans le paysage de la recherche en informatique français pour quasiment l'ensemble de ses thématiques. La forte croissance du laboratoire lui a même obligé a s'étendre sur 2 autres sites : au sein de l'Institut Rhône-Alpin des Systèmes Complexes, et au sein de locaux appartenant à l'UCB à Gerland.

1.1.2 Organisation générale

Le laboratoire compose l'Unité Mixte de Recherche 5668 avec l'ENS Lyon (EnsL), l'Université Claude Bernard Lyon 1, l'Inria et le CNRS. Il est dirigé par **M. Patrick Baillot** qui est secondé par **M. Frédéric Vivien**. Le responsable en charge de l'appel à projets, de la valorisation de recherche et des relations internationales est **M. Eddy Caron** et le responsable en charge des thèses, de l'enseignement et des postes non permanents est **M. Damien Stehlé**.
L'équipe administrative et l'équipe en charge des moyens informatique quant-à-elles sont composés de différentes personne issues des institutions qui constituent l'Unité Mixte de Recherche 5668.

Le laboratoire héberge sept équipes de recherche dont cinq sont commune avec l'Inria : AriC, Avalon, CASH, DANTE, MC2, PLUME et ROMA chacune administrés par un chef d'équipe.

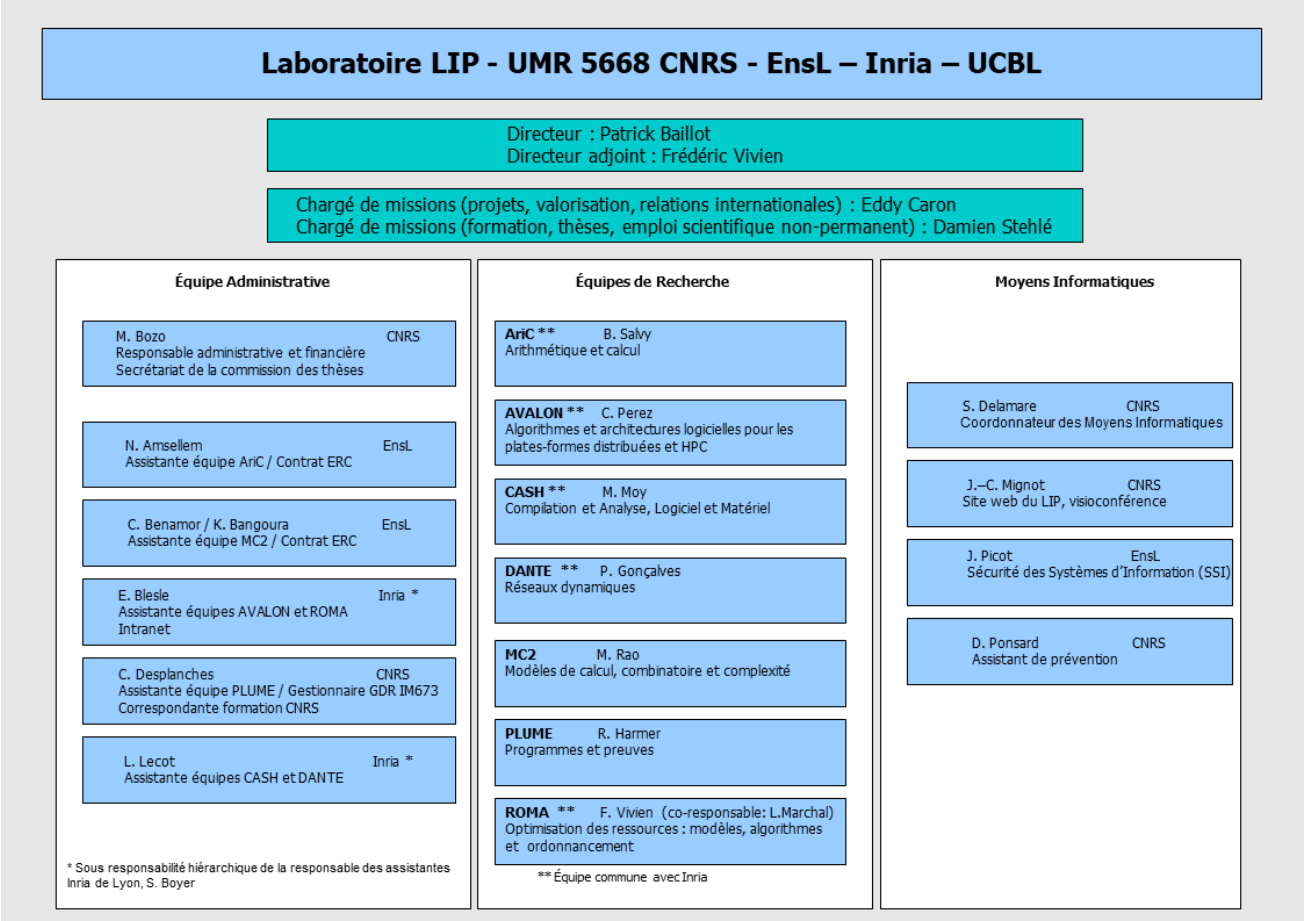

FIGURE 1 – Organigramme du LIP au 10 Avril 2018 [6]

1.1.3 Des métiers variés

Le projet du Laboratoire de l'Informatique du Parallélisme est de mettre en relation l'informatique fondamental et sa mise en œuvre pratique dans les institutions. Ainsi le laboratoire crée un lien fort entre informatique et d'autres sciences comme les mathématiques, les sciences du vivant ou, plus globalement, les sciences fondamentales.[1]

Les chercheurs du LIP possèdent un socle commun : l'algorithmie et l'utilisation efficace des ressources. Ils organisent leurs recherches autour de deux grands axes :

— La conception, l'utilisation et l'adéquation aux besoins des applications des futurs architectures de calcul (traitement de données, calcul fondamental) et de communication (réseaux)

— L'étude des modèles et des méthodes en informatique : compléxité, algorithmie, développement logiciel et matériel, avancée technologiques etc.

Ainsi, le laboratoire ne compte pas dans ses rang uniquement des expert de l'informatique. D'autres profils bien différents sont mis en valeurs dans les 7 équipes-projets que nous allons présenter.

AriC - Arithmetic and Computing

AriC est une grande équipe projet commune avec l'Inria composée d'une vingtaine de membres. Elle à pour but d'améliorer le calcul, en terme de performance, d'efficacité et de fiabilité. Ses 3 principaux projets de recherche portent sur les sujets suivants :

- Les réseaux euclidiens : algorithme et cryptologie,
- Les méthodes d'approximation efficaces en calcul formel,
- Le calcul fiable à haute performance, avec virgule flottante et précision d'au plus une centaine de bits.

Au delà de la recherche cette équipe à une véritable vocation à la diffusion et à la vulgarisation de leurs travaux, ce qui passe par la multiplication des interventions dans les lycées ou autre institutions d'enseignement et la publication d'articles et d'ouvrages [14].

Avalon - Algorithms and Software Architectures for Distributed and HPC Platforms

L'objectif de l'équipe commune Inria Avalon est de concevoir des modèles de programmations, des systèmes et des algorithmes pour exécuter des applications sur des ressources tout en satisfaisant les contraintes des utilisateurs (e.g. coût, performances) et des administrateurs (e.g. maximisé l'usage des ressource, minimiser la consommation d'énergie). L'équipe se concentre en particulier sur le profilage et la modélisation d'applications gourmandes en énergie et en données, la gestion des données et l'ordonnancement des applications sur des architectures de supercalculateurs [16].

CASH - Compilation and Analyses for Software and Hardware

La vision de l'équipe commune Inria CASH est d'utiliser l'architecture dataflow pour le traitement des données par les supercalculateurs. Son but est d'utiliser les caractéristiques particuliers du matériel informatique afin de fournir des couples matériel-logiciel efficaces énergétiquement au développeur final. Pour ce faire elle travaille sur les axes d'études suivants :

- Développer l'architecture dataflow,
- Améliorer les algorithmes de compilation,
- Développer la compilation matériel, qui consiste à transformer un programme informatique en un circuit électronique physique,
- Émuler les systèmes sur puce pour faciliter leur optimisation [17].

DANTE - Dynamic Network

L'objectif principal de l'équipe commune Inria DANTE est de poser des bases solides à la caractérisation des réseaux dynamiques et des processus dynamiques se produisant sur des réseaux à grande échelle. Afin de développer des outils d'une pertinence pratique en situation réelle, elle fonde ses études méthodologiques sur des jeux de données réelles. Ses 3 grands thèmes de recherche sont :

- Le traitement du signal basé sur les graphes,
- La théorie des graphes dynamiques,
- Les algorithmes distribués pour les réseaux dynamiques [18].

MC2 - Models of computation, Complexity, Combinatorics
L'équipe MC2 à pour but de comprendre les possibilités et les limitations des algorithmes efficace. Pour ce faire elle crée et analyse des algorithme jusqu'à leurs limites. Parmi les différents domaines des mathématiques au cœur de ses problématiques, l'équipe MC2 se concentre sur l'algèbre et l'analyse combinatoire. Ces deux domaines sont des sources de problèmes algorithmiques qui jouent un rôle clé dans la théorie de la complexité [19].

PLUME - Programs and Proof
Les recherches menées par l'équipe PLUME s'articulent autour de deux thèmes fortement imbriqués : les fondements logiques des langages de programmation et la théorie des systèmes informatiques. Elle met au centre de ses recherche la logique mathématique afin de trouver comment écrire des programmes sûrs ou comment vérifier formellement des systèmes informatique complexes [20].

ROMA - Resource Optimization : Models, Algorithms and Scheduling
L'équipe commune Inria ROMA vise à concevoir des modèles, des algorithmes et des stratégies d'ordonnancement pour optimiser l'exécution d'applications scientifiques sur des supercalculateurs. Plus spécifiquement, ROMA vise à obtenir la "meilleure" performance possible du point de vue de l'utilisateur (e.g. le temps d'exécution de l'application) tout en utilisant les ressources aussi efficacement que possible [21].

Ainsi, le Laboratoire de l'Informatique du Parallélisme possède un impressionnant savoir faire dans de nombreux domaines de l'informatique et produit de nombreuses ressources pour des institutions publiques et privés, comme nous allons le voir.

1.1.4 Une production conséquente

Le Laboratoire de l'Informatique du Parallélisme est plutôt prolifique dans la quantité de production. Depuis 2000, 2502 publications ont été publiés et sont disponibles sur des plateforme en ligne comme les archives ouvertes HAL. En effet le laboratoire est dans une véritable démarche de production de connaissances, mais cela ne l'empêche pas de s'autofinancer grâce à des contrats industriels et des projets institutionnels régionaux, nationaux et internationaux [9]. 2 brevets ont même été déposés.

Le laboratoire encourage également les ambitions d'entrepreneuriat de ses membres avec la création de 5 start-ups dans le domaine du numérique depuis 2010 [1].

Enfin, les différentes équipes du laboratoire produisent également de nombreuses ressources logicielles open-source à destinations des institutions, les industriels et même des particuliers que l'on peut retrouver sur les plateforme de partage de code source en ligne.

1.2 L'équipe Avalon en détails

L'équipe dont je fais partit durant ce stage est l'équipe Avalon. Cette équipe créée le 1er Février 2012 [15] est une équipe-projet du Laboratoire de l'Informatique du Parallélisme commune à l'Inria composée de 22 membres : 8 universitaires permanents, 2 universitaires temporaires, 4 membres permanents et 8 doctorants. Elle est située au troisième étage de l'aile sud du bâtiment M7 sur le site Monod de l'École Normale Supérieure de Lyon.

1.2.1 Les membres de l'équipe

Parmi les 22 membres de l'équipe, voici un rapide aperçu de ceux que j'ai côtoyé durant mon stage :

Universitaires permanents

Christian Perez
Chef de l'équipe
Chercheur sénior Inria

Eddy Carron
Responsable administratif
Enseignant chercheur

Laurent Lefèvre
Mon maître de stage
Chercheur Inria

Universitaires temporaires

Marcos Dias de Assuncao
Chercheur

Cyril Seguin
Chercheur PostDoc
Expert Cloud et Sécurité

Équipe

Evelyne Blesle
Assistante administrative

Simon Delamare
Ingénieur de recherche

Matthieu Imbert
Ingénieur de recherche

Doctorants

Issam Raïs
Thèse sur l'étude de la consommation énergétique des supercalculateurs

Dorra Boughzala
Thèse sur la simulation de la consommation d'énergie des architecture hétérogènes

Alexandre da Silva Veith
Thèse sur les algorithmes pour l'analyse des flux élastiques du Big-Data

Hadrien Croubois
Spécialisé dans les processus parallèles et les systèmes distribués

Felipe Rodrigo de Souza
Thèse sur les algorithmes de provisionnement des réseaux

Valentin Lorentz
Thèse sur la traçabilité énergétique des données

1.2.2 Le contexte de création

Formée le 1 Février 2012, l'équipe Avalon est une véritable réponse aux changements effrénés de l'informatique. L'évolution très rapide du matériel informatique en terme de communication, de traitement de données et de virtualisation à fait émerger des nouveaux besoins pour l'utilisateur. En effet la complexité des systèmes informatiques augmente ! Il existe aujourd'hui de nombreuses variétés de plateformes à grande échelles disponibles pour des chercheurs ou des industriels qui souhaitent satisfaire leurs besoins en traitement de données : agrégation de clusters, grand data-centers, supercalculateurs etc. Chacune de ces plateforme disposent de spécificités intrinsèques, d'accès et d'utilisation qui ont un impact important sur l'architecture et l'exécution des applications qui souhaitent les utiliser. Elles intègrent de nombreuses fonctionnalités obligatoires comme le sécurité, la virtualisation, le load-balancing ou autres qui augmentent encore plus leur complexité d'utilisation. C'est dans ce contexte que l'équipe Avalon a été créée, la réponse qu'elle apporte est d'aller plus loin dans l'abstraction de ces plateformes pour assurer à l'utilisateur une utilisation simplifiée tout en gardant l'ensemble des fonctionnalités disponibles. [15]

1.2.3 La vision de l'équipe

La vision de l'équipe Avalon est de considérer l'ensemble du système, de la ressource à l'application, afin de concevoir des outils simples à utiliser par les programmeurs tout en permettant une exploitation efficace des ressources. L'équipe se concentre en particulier sur la gestion de l'élasticité (i.e la capacité à s'adapter aux besoins des applications le plus rapidement possible)

des plateforme parallèles et distribués ainsi que leur efficacité énergétique. [15]

L'équipe souhaite pouvoir mettre à disposition d'autres équipes de recherche travaillant dans d'autres sciences des ressources informatiques à haute performance de manière simple à utiliser. Voici quelques exemples de disciplines qui pourrait avoir recours aux travaux de l'équipe Avalon :

- La biologie, avec par exemple le séquençage de l'ADN,
- L'étude du climat, qui demande une quantité de paramètres impressionnante pour simuler les changements de climat prochains,
- L'astrophysique, qui est demandeuse de simulations pour comprendre les phénomènes physique qui nous entoure, la formation des galaxies, pour étudier la matière noire etc.

1.3 Mon environement au sein du laboratoire

Dans cette partie nous allons voir dans quel environement de travail j'ai évolué tout au long de mon stage au sein du laboratoire. Nous présenterons tout d'abord comment s'est organisé le travail, ensuite dans quelle environement technique j'ai évolue et enfin quelles sont mes interactions avec les autres membres de l'équipe afin.

1.3.1 Les horaires de travail

Le travail au laboratoire est assez librement organisé. En effet, pendant toute ma période de stage j'ai eu, dans la limite d'effectuer 35 heures par semaines, des horaires libres. Cela m'a permis d'organiser facilement d'autres tâches importantes de cette période, notamment les entretiens avec les écoles d'ingénieurs pour mon admission en alternance, le forum d'entreprise obligatoire de Polytech ou encore mes entretiens individuels avec des entreprises dans le but de dénicher un contrat d'apprentissage.

Contrairement à lorsque j'étudiais à l'IUT où j'aurais farouchement défendu ma pause de midi de 2h ainsi que mes pauses de 20 minutes toutes les deux heures, je n'ai pas spécialement ressentis pendant ce stage le besoin de prendre de longue pauses. C'est pour cela, qu'après quelques jours, je me suis basé sur un rythme de travail de 9h–16h30.

Arriver à 9h est déjà une grande différence par rapport à l'IUT, c'est tout d'abord une heure plus tard, mais mon temps de trajet étant également plus court pour aller au laboratoire que pour aller à l'IUT, cela me permet de me lever 1h20 plus tard tous les matins.

Pour le temps de midi, une petite particularité du laboratoire est qu'il utilise pour manger le restaurant universitaire de l'ENS Lyon, ainsi pour ne pas faire la queue, les équipes ont décidés de descendre manger à 11h30. Tous les midis une partie de l'équipe Avalon descend ensemble manger pendent environs 30 minutes, puis remonte. Il reste alors 4h30 de travail pour arriver aux 7 heures de travail quotidien.

Je me suis fais la réflexion que ce rythme de travail n'est peut-être pas le plus adéquat en raison de son déséquilibre en le temps de travail du matin et de l'après midi. Mais je n'étais pas convaincu par le fait d'arriver à 8h et de repartir à 15h30.

1.3.2 Les locaux

Ce stage s'est déroulé dans les locaux du LIP, au troisième étage du bâtiment M7 du site Monod de l'ENS Lyon. Pratiquement tous les membres de l'équipe Avalon sont situés au même endroit : un grand couloir désert de part est d'autres des bureaux de deux, parfois trois, personnes.

Au début de mon stage j'étais situé dans le dernier bureau du couloir que je partageais avec **Issam Raïs**, doctorant sous la supervision de mon maître de stage Laurent Lefèvre. Le 22 mai, une réorganisation des bureaux à été faites, j'ai quitté Issam pour rejoindre, dans le bureau situé juste avant, **Lucas Besnard**, un camarade de promotion qui effectue également son stage au LIP, mais qui ne travaille pas sur le même sujet que moi.

Les bureaux sont relativement spacieux, possèdent une armoire commune ainsi qu'un casier individuel. Le principale problème est que l'équipe est situé dans l'aile Sud du bâtiment et que les bureaux sont équipé d'une grande baie vitré coté Sud. Lorsqu'il y a du soleil le travail devient très vite difficile en raison des températures qui grimpent très vite. Heureusement nous avions la possibilité de nous réfugier dans la bibliothèque de l'ENS, qui est climatisé, si cela devenait intenable.

Le bâtiment possède également des boxs de travail ou nous pouvions nous retrouver pour travailler à plusieurs ainsi qu'une grande salle de réunion. Après le couloir, un solarium était également à notre disposition avec des transats, qui m'ont bien été utiles lors de la lecture de publications scientifiques.

1.3.3 L'environement matériel

Le laboratoire pouvait me fournir un ordinateur portable ainsi qu'une station d'accueil pour effectuer nos travaux. Cependant ces machines n'étant pas très performantes on m'a conseillé d'utiliser mon ordinateur personnel.

On m'a par contre fournis un deuxième écran, qui est quelque-chose que je trouve de plus en plus indispensable. Il est d'ailleurs tombé en panne quelques semaines plus tard, probablement à cause de la chaleur, mais Issam m'a très gentiment prêté un de ses écrans qu'il n'utilisait pas.

1.3.4 L'environement technologique

L'équipe Avalon ne possède pas vraiment d'environement technologique propre, car chaque membre utilise les technologies les plus pertinentes pour chaque tâche. Cependant, elle utilise activement le plateforme *Grid'5000* que je vais présenter.

FIGURE 2 – Logo de la plateforme Grid'5000

Grid'5000 est un banc d'essai à grand échelle pour la recherche expérimentale en informatique. Cette plateforme met à disposition des chercheurs un très grande quantité de ressources informatiques : plus de 1000 serveurs, 8000 cœurs de processeur regroupés en clusters. Elle est utilisé par une communauté de plus de 500 utilisateurs et est hébergé sur une dizaine de sites en France. [10]

Elle est à la pointe de la technologie avec notamment une connexion au réseau de 10Gbit/s (5000 fois meilleur que la box qui vous connecte à Internet), des connectiques *Infiniband* qui permettent un débit jusqu'à 56Gbits/s ou encore les processeur ultra performant *Xeon PHI* du constructeur leader *Intel*.

La plateforme intègre de nombreux outils de monitoring et de mesure afin de permettre des expérimentations et leur interprétation très précise. Elle possèdent notamment un arsenal de Wattmètre qui mesurent au plus près de la machine la consommation du matériel et qui sont beaucoup utilisé par les membres de l'équipe Avalon, notamment par Issam et Dorra pour leurs thèses et mon camarade Lucas pour son stage.

Afin de nous former, Lucas et moi, à l'utilisation de cette plateforme nous avons été formé par **Dorra Boughzala**, qui venait d'effectuer une formation complète. Nous passé 2 à 4 heures par jour en sa compagnie pendant la première semaine de notre stage. Je n'avais jamais utilisé une telle plateforme, mais étant déjà familier avec les environnements Linux ainsi que certains concepts, j'ai pû assez facilement m'approprier l'outils, qui s'avère extrêmement intéressant et puissant.

1.3.5 L'environement humain

Évidemment, un laboratoire de recherche est international, un certain nombre des membres de l'équipe ne sont donc pas français. Tout le monde parle bien l'anglais ce qui permet de très bien se comprendre, mais la barrière de la langue se fait parfois ressentir lorsqu'on troque les interactions formelles pour des interactions plus amicales. Les interactions avec les différents membres de l'équipe et du laboratoire sont d'ailleurs très décontractés, tout le monde se tutoie, par exemple.

Pour ma mission je ne communiquais qu'avec Laurent, mon maître de stage qui prennait note de l'avancement du projet tout en m'aiguillait et en me proposant des pistes à suivre, parfois en me fournissant des publications scientifiques. Il n'a cependant pas participé à la phase de développement, où j'étais en toute autonomie.

1.3.6 Les groupes de travail

Lorsque l'équipe en ressent le besoin, des groupes de travail sont organisés. Toute l'équipe se retrouve ensemble dans une sale de réunion. Souvent elle procède à une *round table*, qui consiste à présenter ce sur quoi on travaille, ce que l'on à fait depuis la dernière *round table* et ce que l'on a prévu de faire pour la suite. Mais parfois il peu s'agir d'un membre de l'équipe qui souhaite présenter, une technologie, son travail ou tout autre chose aux autres.
Les membres mettent également par écrit ce qu'ils ont dit durant le groupe de travail, ce qui permet à Alexandre de mettre à jour le site internet de l'équipe Avalon.

2 La mission du stage

TODO : Rédiger l'introduction de la partie

2.1 Présentation de la mission

Ma mission pendant ce stage est de créer un simulateur informatique d'empreinte environnementale des centres de données. Nous allons présenter dans cette partie, en quoi cette mission s'intègre dans des problématique contemporaines et comment nous allons faire pour y répondre.

2.1.1 Le contexte

Face à la constante augmentation des besoins en ressources informatique, que ce soit de calcul, de stockage ou de communication ainsi que l'augmentation de leur qualité (disponibilité, rapidité etc.) de nombreux centre de données doivent être construit pour répondre à ces besoins. L'échelle de grandeur de ces centres de données doit être à l'échelle du besoin, ainsi ils sont de plus en plus imposant et construits de plus en plus proche des environement urbain qui concentrent les usages.

Cela n'est pas sans conséquence, les centres de données génèrent une très grosse empreinte environnementale. Leur consommation d'énergie est record, pouvant aller jusqu'au plus de 20GW, en effet l'informatique est très gourmand en énergie et nécessite d'être refroidis par des systèmes encore plus gourmands. Ils ont un très gros impact sur le territoire : grosse surface au sol occupée, génèrent de la chaleur, du bruit et des vibrations qui peuvent gêner le voisinage. Ils possèdent des groupes électrogènes de secours, qui apportent des risque supplémentaire notamment par le stockage de grande quantité de fioul et impactent le voisinages via leur test réguliers qui génèrent une grande quantité de fumée.

Une illustration de ce nouveau besoin en ressources informatiques au plus près des villes sont les jeux olympique 2024 à Paris. En effet, le trafic de données sur la région parisienne va grandement augmenter, de part les besoins des institutions sportive, mais également par celle des spectateurs qui sont aujourd'hui tous connecté à internet par leur smartphone. Il va donc falloir intégrer dans le territoire de nouveaux centres de données pour traiter ces nouvelles données. Il serait donc intéressant de fournir au urbanistes et aux architectes un outils qui permettrais de comparer les projets de centre de données entre eux et de mesure leur impact sur leur environnement.

Ainsi, l'Institut d'aménagement et d'urbanisme de la région Île-de-France et l'Ecole d'architecture de la ville et des territoires, acteurs de l'urbanisme en région parisienne, se sont associés avec le Laboratoire de l'Informatique du Parallélisme, qui n'a plus à prouver son expertise sur les sujets environnementaux en lien avec l'informatique, afin de s'entraider dans la réflexion de nouveaux projets de centre de données en environement urbain.

2.1.2 Les enjeux

Les travers des centres de données ne sont pas ignorés par les industriels et les institutions. Les pratiques de *GreenIT*, littéralement *informatique vert*, se développent de plus en plus. Le réchauffement climatique est un enjeu planétaire, et la réduction des émissions de gaz a effet de serre peut se faire via la diminution de la consommation d'énergie des centres de données qui représentait déjà, en France, 4% de la production d'électricité en 2015 [13]. Les industriels, eux ont tout de suite vu que la réduction de la consommation d'énergie implique obligatoirement la diminution des coûts. De plus ayant remarqué l'attrait du public et des institutions pour l'efficacité énergétique de leurs centres de données, ils peuvent rentabiliser leurs actions en faveur de l'efficacité énergétique par des campagnes de green-washing.
La population elle aussi devient méfiante à l'égar des nouveaux projets de centre de données et pointent les nuisances, parfois illégales, qu'ils génèrent sur le voisinage, jusqu'à porter plainte [3], ce qui porte préjudice aux société exploitantes.

Les porteurs de projets, quant-à-eux ne possèdent pas toujours des outils simples qui permettent de mesurer l'impact de leur projet sur l'environement. Ils n'ont d'ailleurs pas toujours de connaissances en informatique, il est donc primordiale d'apporter un outils simple, compréhensible pour tous, fiable et assez puissant pour prendre en compte toutes les caractéristiques des centres de données afin de les aider dans la construction de leur projet.

2.1.3 L'outils demandé

Pour répondre à ces besoins, nous avons imaginé un outils informatique permettant de simuler l'impact environnementale d'un centre de données. Cet outils devra tout d'abord être facilement compréhensible par un architecte ou un urbaniste car ils seront probablement les utilisateurs finaux. Le simulateur devra prendre en compte une multitude de paramètres décrivant un centre de données entrés par l'utilisateur, les compiler et les présenter à l'utilisateur sous une forme permettant des les analyser. Pour ce faire, nous utiliserons un certain nombre d'indicateur de *Green IT* reconnus par des institutions ou des organismes et les présenterons à l'utilisateur via un rapport détaillé généré automatiquement par l'outils.

Le simulateur devra être construit autour de trois grands blocs : le matériel informatique, le refroidissement, et les sources d'énergie électrique. L'utilisateur doit pouvoir entrer la liste du matériel informatique du centre de données, ce qui permettra de calculer la chaleur généré ainsi que la consommation d'énergie. Ensuite, il devra pouvoir choisir les paramètres du refroidissement afin de traiter la chaleur généré, ce qui fera augmenter la quantité d'électricité consommé par le site. Enfin il devra pouvoir choisir comment le centre de données est alimenté électriquement pour répondre à ses besoins énergétiques.

Au début du projet nous souhaitions un outils fonctionnel et puissant mais pas nécessairement beau ou très ergonomique, nous verrons par la suite que j'ai volontairement modifié cette direction afin de donner à l'utilisateur une meilleur expérience.

2.2 La recherche bibliographique

Avant d'espérer pouvoir commencer le développement de l'outils, mon maître de stage, Laurent Lefèvre, m'a conseillé de commencer par une grande recherche bibliographique afin d'intégrer les problématiques de Green IT et plus particulièrement de chercher des indicateurs qui pourraient nous intéresser pour mesurer l'empreinte environnementale des centres de données. Je me suis tout d'abord intéresser aux indicateurs fournis par l'organisme GreenGrid qui fut le premier à fournir ce genre d'indicateur de Green IT pour les centres de données.

Dans cette partie nous présenterons et critiquerons l'ensemble des indicateurs que j'ai retenu pour les intégrer dans le simulateur.

2.2.1 Power Usage Effectiveness (PUE)

Les centres de données effectuent de nombreuses dépenses énergétiques pour placer le matériel informatique dans des conditions optimales de fonctionnement : refroidissement de l'air, stabilité et qualité de l'électricité, sécurité du bâtiment etc. Il est alors intéressant de comparer la part d'énergie utilisée pour ces dépenses auxiliaires et la part allouée au travail utile du data-centre.

Le PUE est un indicateur créé par l'organisme GreenGrid qui montre la part d'énergie utilisée par tout ce qui entoure le matériel informatique par rapport à la part utilisée par le matériel informatique lui-même. Il est défini par la formule suivante :

$$PUE = \frac{Consommation\ de\ l'établissement}{Consommation\ de\ l'équipement\ informatique}$$

Ainsi un PUE de 3 indique que le centre de données utilise 3 fois plus d'énergie en infrastructure qu'en travail utile. Il existe également une variante du PUE sous forme d'indice, appelé DCE et calculé comme l'inverse du PUE, il permet d'obtenir une valeur bornée entre 0 et 1.

Le PUE s'est imposé durant les dernières années comme norme afin de comparer l'efficacité énergétique des data-centre et est même considéré par certains comme un critère « éco-responsable ». Pour plus de simplicité on a alors découpé le PUE en 7 classes énergétiques :

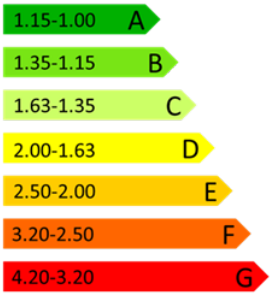

FIGURE 3 – Les classes énergétiques du PUE

Cependant, il ne faudrait pas lui donner des propriétés qu'il ne possède pas. Malgré la tendance actuelle accommodée au green-washing, le PUE n'est pas très bien choisi pour comparer des data-centres entre eux. Des éléments manquent pour une comparaison juste et efficace : localisation géographique, taux de disponibilité ou taux de charge [4]. De plus, le mode de calcul possède un biais : baisser la quantité d'énergie utilisée par l'équipement informatique, même si c'est pour une meilleur gestion de l'énergie fait augmenter le PUE. La virtualisation par exemple fait augmenter le PUE alors qu'elle offre une puissance de calcul par unité d'énergie plus importante [25]. Le même raisonnement peut être appliqué à la gestion de la charge, qui, en éteignant certains équipements informatiques, fait augmenter le PUE [22].

Cependant, s'il on prend le PUE comme il est, c'est à dire un indicateur qui montre la proportion d'énergie utilisé par l'architecture du centre de données par rapport à la consommation de l'équipement informatique et non pas un fer-de-lance de l'éco-conception des centres de données il reste pertinent pour notre simulateur.

2.2.2 Carbon Usage Effectiveness (CUE)

Cet indicateur est le deuxième indicateur de la famille xUE de l'organisme GreenGrid et aspire à devenir un standard dans l'industrie pour comparer les data-centres sur leurs rejets de dioxyde de carbone comme l'a été le PUE pour l'efficacité énergétique.

$$CUE = CEF \times PUE$$

Le CEF représente le facteur d'émission de dioxyde de carbonne pour un kWh, en France cette valeur vaut en moyenne 74 g/kWh [7]. Cependant cette valeur moyenne n'est pas très précise, en effet on peut imaginer que des centre de données utilise certains fournisseurs d'énergie afin de limiter leur émissions de CO2, ou produisent leur propre énergie sur site. Afin d'obtenir une valeur correcte du CUE il faudra donc calculer le CEF en fonction de la quantité de gaz à effet de serre émise par chaque source d'énergie qu'utilise le centre de données.

Le CUE permet d'évaluer la quantité de CO2 produite par le data-centre en fonction de sa consommation d'énergie. La valeur optimale du CUE est donc zéro.

2.2.3 Data Center energy Productivity (DCeP)

$$DCeP = \frac{Travail\ utile}{Consommation\ de\ l'établissement}$$

Le DCeP représente l'énergie nécessaire pour une opération. L'unité du travail utile est à définir selon l'usage de l'indicateur, on peut imaginer un flop, une exécution d'un programme, un client, une commande etc.

Grâce à cet indicateur on peut également évaluer la quantité de gaz à effet de serre rejetée dans

l'atmosphère par unité de travail utile en utilisant le facteur d'émission de dioxyde de carbone par kWh du centre de données (CEF).

$$CEF \times DCeP$$

Pour notre simulateur cet indicateur est difficile à exploiter car il est très difficile d'estimer la valeur du travail utile car c'est extrêmement spécifique. Cependant, il serait cependant dommage de ne pas donner la possibilité à l'utilisateur de le calculer si il en connait les détails.

2.2.4 Green Energy Coefficient (GEC)

$$GEC = \frac{Consommation\ énergie\ verte}{Consommation\ totale\ d'énergie}$$

Le GEC quantifie la part d'énergie renouvelable consommée par le data-centre. Ce coefficient permet de nuancer l'impact carbone du data-centre s'il utilise des énergies renouvelables. La consommation d'énergie verte ne correspond pas à la part d'énergie renouvelable injectée dans le réseau électrique du fournisseur d'énergie, la data-centre doit posséder, ou posséder les droits d'utilisation, de cette source d'énergie.

2.2.5 Energy Reuse Factor (ERF)

$$ERF = \frac{Énergie\ réutilisée\ à\ l'extérieur}{Consommation\ totale\ d'énergie}$$

Le ERF quantifie la quantité d'énergie réutilisée à l'extérieur du data-centre. Cette énergie peut prendre différentes formes : chaleur servant de chauffage, électricité résiduelle, mouvement etc. et est mesurée à la sortie du data-centre et convertie en kWh . Le but de cet indicateur est d'inciter les data-centres à réutiliser leur énergie résiduelle plutôt que la rejeter.

2.2.6 Efficacité énergétique de l'équipement (EEE)

Le Standard Performance Evaluation Corporation (SPEC) est un consortium proposant, entre autre, des benchmark du matériel informatique. Son indicateur *ssj_ops* représente le débit de calcul (le nombre d'opération par seconde) durant un benchmark effectué sous une machine virtuelle JAVA [23].

Le SPEC met également à disposition les mesures de consommation d'énergie pendant le benchmark ce qui lui permet de déduire l'efficacité énergétique de l'équipement en calculant son nombre d'opérations par unité d'énergie consommée (*ssj_ops_perwatt*). Le jeu de données contient 551 modèles de serveur commercialisés entre 2008 et 2018 [24].

On peut alors définir l'efficacité énergétique de l'équipement du data-centre par la moyenne de l'efficacité de chaque équipement :

$$EEE_{DC} = \frac{1}{n}\sum_{i=1}^{n} ssj\_ops\_perwatt_i$$

Même si cette initiative nous permet d'avoir un ordre d'idée de l'efficacité énergétique de l'équipement, nous devons garder à l'esprit que les mesures, envoyées par de multiples organismes, ne sont pas vérifiées par le SPEC et qu'elle sont effectuées sous un environnement JAVA qui n'est pas forcement représentatif de l'utilisation d'un serveur de centre de données.

2.2.7 KPI Global DCEM

Ce meta-indicateur normalisée par l'Europe [12] en réponse aux critiques à l'égard du PUE américain permet le suivi de la gestion énergétique d'un data-centre. Il est défini par 4 indicateurs que nous allons présenter qui permettent une interprétation plus fine des résultats.

Ce meta-indicateur se présente sous la forme d'un couple de valeur défini par (DC_G, DC_P), représentant respectivement la taille du data-centre ainsi que sa classe énergétique.

KPI Energy Consumption (KPI_{EC})
Cet indicateur représente l'énergie nécessaire pour le fonctionnement du data-centre pendant une année. Il prend en compte les dépenses énergétiques liées à l'infrastructure technique mais également celles liées à l'exploitation du data-centre : sécurité, gardiens, maintenance etc. Cependant il ne prend pas en compte la consommation d'énergie des bâtiments contenant les bureaux des employés qui mettent en place et utilisent le data-centre : chefs de projets, devOps etc.

Cet indicateur est très normalisé et détaillé dans son calcul, vous retrouverez en Annexe A le calcul exact du KPI_{EC}.

KPI Task Efficiency (KPI_{TE})
Cet indicateur représente le rapport entre la consommation d'énergie de tous les composants quels qu'ils soient (EC_{DC}) et celle des composants qui calculent, stockent ou transporte des données (EC_{HE}). Tous les composants transformant l'électricité ou améliorant la disponibilité doivent être pris en compte ainsi que tous les équipements en aval des sources d'énergie : éclairage, climatisation, sécurité, distribution électrique, extraction de la chaleur etc.

$$KPI_{TE} = \frac{EC_{DC}}{EC_{HE}}$$

Cet indicateur rappelle le PUE à la différence qu'il est beaucoup plus précis dans la liste des équipements à prendre en compte, la méthode de mesure (à l'extérieur du composant et au plus près de son entrée et de sa sortie pour ne pas prendre en compte les pertes.

KPI Energy Reuse (KPI_{REUSE})
Cet indicateur représente le rapport entre l'énergie réutilisée en dehors des équipements informatiques du data-centre (EC_{REUSE}) et l'énergie totale dépensée par le data-centre (EC_{DC}). Voici les

initiatives courantes pour la réutilisation de l'énergie :

— La production d'eau chaude,
— Le chauffage des bureaux ou d'appartements adjacents,
— Le préchauffage des diesels

$$KPI_{REUSE} = \frac{EC_{REUSE}}{EC_{DC}}$$

Datacenter Energy Consumption Gauge (DC_G)

Cet indicateur définis le gabarit de consommation d'énergie du data-centre. Il est défini par le tableau suivant :

DC_G	Intervalle du KPI_{EC}
S	$KPI_{EC} \leq 1\,\text{GWh}$
M	$1\,\text{GWh} < KPI_{EC} \leq 4\,\text{GWh}$
L	$4\,\text{GWh} < KPI_{EC} \leq 20\,\text{GWh}$
XL	$KPI_{EC} > 20\,\text{GWh}$

TABLE 1 – Calcul du Datacenter Energy Consumption Gauge (DC_G)

Datacenter Performance (DC_P)

Cet indicateur défini la performance énergétique du data-center, il est obtenu par la formule suivante :

$$DC_P = KPI_{TE} \times (1 - W_{REUSE} \times KPI_{REUSE}) \times (1 - W_{REN} \times KPI_{REN})$$

W_{REUSE} et W_{REN} sont deux facteurs de modération susceptibles de changer en fonction de la taille du data-center, leurs valeurs par défaut sont 0,5.

Une fois calculé on pourra définir la classe à partir du tableau suivant :

2.2.8 Indice d'Efficacité du Transport Électrique

$$ETE = 1 - \frac{\acute{E}nergie\ perdue}{Energie\ nominale\ de\ la\ ligne}$$

Cet indicateur permet de mesurer l'efficacité du transport de l'énergie jusqu'au data-centre. Pour son calcul nous devons prendre en compte sa situation géographique par rapport aux transformateurs électriques et ses interactions avec les lignes de transport.

Le calcul de l'énergie perdu

Pour calculer les pertes énergétiques nous utiliserons l'effet de peau [29] ainsi que l'effet corona [26]

Classe	DC_P	
	$\geq$	$<$
A		0,70
B	0,70	1,00
C	1,00	1,30
D	1,30	1,50
E	1,50	1,70
F	1,70	1,90
G	1,90	2,10
H	2,10	2,40
I	2,40	

TABLE 2 – Calcul du Datacenter Performance (DC_P)

qui sont les deux principaux type de pertes énergétiques dans les lignes électriques transportant du courant alternatif [11].

$$\acute{E}nergie\ perdue = P_{peau} + P_{corona}$$

Vous retrouverez en Annexe B la méthode de calcul de l'effet de peau et de l'effet corona.

Le calcul des distances

Afin de calculer les pertes énergétiques pendant le transport de l'énergie jusqu'au data-centre nous devons prendre en compte la topologie du réseau électrique. Grâce aux données RTE [8] nous pouvons en déduire où est situé le transformateur ou la ligne de bonne capacité la plus proche. Il nous suffira ensuite de calculer la distance entre le data-centre et cet équipement.

2.2.9 Indice de conformité de la température extérieure (CTE)

Les centre de données doivent maintenir leurs équipements informatiques à une température comprise entre 18 °C et 27 °C [2]. Plus la température de l'air extérieure est basse, moins il sera nécessaire de consommer de l'énergie pour refroidir l'air.

Il est alors intéressant de calculer la part du temps où la température extérieure ne dépasse pas un seuil, ce qui permettra de discriminer l'emplacement géographique du centre de données.
Le seuil peut être défini selon l'objectif que l'on souhaite atteindre.

- S_{best}, la température la plus basse conseillée (18 °C)
- S_{hardw}, la température maximale autorisée par le contrat de maintenance constructeur (généralement 24 °C)
- S_{worst}, la température la plus haute conseillée (27 °C)

$$CTE_{seuil} = \frac{Nombre\ d'heures[t° < S_{seuil}]}{24 \times 365}$$

Le nombre d'heures où la température ne dépasse pas le seuil devra être obtenu à partir d'un

jeu de données retraçant l'historique des températures sur l'année par emplacement géographique. Nous verrons plus tard dans ce rapport que l'accès à ces données est beaucoup plus compliqué que prévu.

2.2.10 Indice de Pertinence Géographique

Nous avons vu que plusieurs indicateurs discriminent la position géographique du data-centre : l'efficacité du transport électrique (ETE) qui diminue plus le data-centre est éloigné de la source d'énergie et l'indice de conformité de la température extérieur (CTE) qui diminue si l'emplacement géographique est dans un climat trop chaud par rapport à la température interne du data-centre. Pour plus de simplicité, il est intéressant de les regrouper sous un unique indicateur.

$$IPG = \frac{ETE + CTE}{2}$$

2.3 La développement

Une fois la recherche bibliographique à peu près terminé, nous avons une idée plus clair de à quoi le simulateur devrait ressembler. Nous présenterons dans cette partie les différentes tâches effectués lors du développement du simulateur. *TODO : GANT*

2.3.1 Le choix des outils

Pour le développement de ce simulateur j'ai décidé d'utiliser le langage JAVA. En effet, j'avais déjà une expérience sur ce langage qui à été consolidé par l'enseignement reçu en DUT. JAVA est également plateforme ce qui permet au simulateur de pouvoir fonctionner sous la plupart des systèmes d'exploitation.

Pour ma gestion des librairies j'ai utilisé la technologie Maven qui permet de gérer très facilement ses librairies ainsi que la compilation des projets Java. Même si il peut y a voir quelques fois des complications, cet outils est généralement très facile d'utilisation.

J'ai décidé d'utiliser pour l'interface utilisateur la technologie JavaFX qui est désormais intégré dans l'installation par défaut de Java. J'avais découvert cette technologie via mes camarades de promotions et l'est utilisé pour la première fois dans le projet de groupe du cours de *CPOA* où nous avions réaliser une application de gestion d'un tournois de tennis. Le choix de cette technologie est logique au vu de sa puissance et sa facilité d'utilisation en comparaison des technologies plus anciennes comme *AWT* ou *Swing*.
Cependant, je n'étais pas totalement satisfait du rendu graphique de JavaFX, c'est pour cela que j'ai cherché une alternative qui me permette d'embellir mes interface graphiques. C'est alors que j'ai découvert la librairie *JFoenix* qui implémente le Matérial Design, qui est un courant de design graphique numérique initié par Google, en JavaFX.

2.3.2 L'architecture générale

Avant de développer chaque fonctionnalité du simulateur il faut tout d'abord réfléchir à comment va s'organiser le code source. Le simulateur va prendre la forme d'une succession d'écrans dans lesquelles l'utilisateur pourra saisir des valeurs, une fois les valeurs vérifiés, l'utilisateur pourra passer à l'écran suivant. Une fois qu'un écran est complété, l'utilisateur peut revenir en arrière sur un autre écran afin de modifier ses valeurs.

Pour arriver à ce résultat, j'ai mis en pratique une notion que nous avons appris en DUT : le modèle MVC.

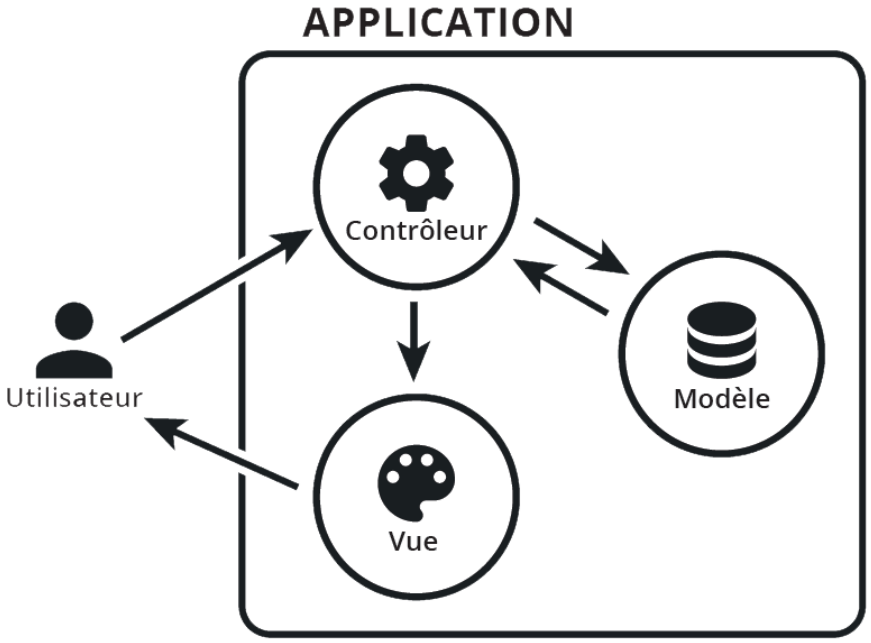

FIGURE 4 – Schéma du modèle MVC

Le modèle MVC (Modèle-vue-contrôleur) est un modèle d'architecture logiciel destiné aux interfaces graphiques. Il se découpe en 3 modules effectuant chacun une action spécifique :

- Le modèle qui s'occupe de récupérer les données à afficher dynamiquement,
- La vue qui contient toutes les composants graphiques,
- Le contrôleur qui s'occupe de récupérer et traiter les actions faites par l'utilisateur.

Ainsi chaque écran possède une vue et un contrôleur qui parfois va chercher des information dans un modèle. La vue est stocké sous forme de fichier FXML, qui est une implémentation du langage de balisage XML propre à JavaFX et qui nous permet de construire des vue à la manière d'une page web. Chaque vue, en plus des éléments graphiques, possède un nom, un numéro d'apparition ainsi qu'une référence vers leur contrôleur associé. Afin de charger toutes les vues en mémoire le plus facilement possible, on les stocke dans un même dossier, ainsi au lancement du simulateur il suffira de récupérer la liste des fichiers de ce dossier, de lire le FXML de chaque fichier afin de créer la vue et d'y associer le contrôleur associé grâce à la référence stockée dans le fichier.

2.3.3 L'architecture des contrôleurs

Pour que le simulateur soit fonctionnel j'ai besoin d'effectuer certaines actions à certains moments dans l'exécution du programme. Je dois par exemple initialiser les composants graphiques de la vue une seule fois au tout début du programme pour qu'ils soient fonctionnel. J'ai également besoin, parfois d'effectuer des actions à chaque fois que l'utilisateur va sur un écran. Je dois également pouvoir être capable de changer les lignes de textes lorsque l'utilisateur demande à changer de langue.
Pour ce faire j'ai créé un *interface* Java appelé `ViewController`, il s'agit d'une sorte de modèle que je peux appliquer à mes contrôleur afin qu'ils implémentent les modules que j'ai définis dans l'interface. Cet interface comporte donc 3 modules :

- L'initialisation, qui s'exécute une seule fois par exécution du simulateur, au chargement de la vue en mémoire,
- Le chargement, qui s'exécute à chaque fois que l'utilisateur change d'écran,
- L'application de la langue, qui s'exécute à lorsque l'utilisateur souhaite changer la langue.

Ces trois modules étant essentiels pour chaque écran, tous mes contrôleur de vue implémentent cet interface.

J'ai également créé un autre interface Java, appelé `FormViewController`, qui lui ne sera appliqué qu'aux vues qui comporte un formulaire car il permet en implémentant les 3 modules suivant de gérer les champs des formulaire :

- La sauvegarde, qui vérifie si les champs sont valides et lance la sauvegarde
- Le remplissage des champs, qui s'exécute lorsque l'on charge un sauvegarde en entrant les valeurs lues
- Le nettoyage des champs, qui supprime toutes les valeurs contenus dans les champs

2.3.4 La notion de simulation

Pour que l'utilisateur puisse gérer facilement ses différentes configuration de centre de données qu'il rentre dans le simulateur j'ai décidé d'avancer la notion de simulation. Une simulation est tout simplement une instance du simulateur, elle contient un nom, des auteurs ainsi que tous les paramètres entrés par l'utilisateur durant l'exécution du programme. Cette simulation peut être facilement stocké et lue depuis le disque dur de l'utilisateur.

D'un point de vue technique, la simulation représente un objet Java que nous sérialisons au format JSON afin de la stocker sur le disque. Il nous suffira ensuite de la dé-sérialiser pour la charger en mémoire. J'ai fais le choix du format JSON car il est facilement lisible par l'humain et qu'il est extrêmement utilisé par les développeurs, la plupart des langages proposent des librairie pour le lire. Ainsi les fichiers de sauvegarde du simulateur pourrons facilement être utilisé dans d'autres programmes, même si ils ne sont pas développés en Java.

2.3.5 La gestion de la langue

Le laboratoire étant international j'ai naturellement penser à implémenter la possibilité de changer la langue du simulateur. Pour ce faire chaque ligne de texte qui est affichée à l'écran est codé par un identifiant unique. Par exemple l'identifiant `OPEN_SIMULATION` représente la phrase « Ouvrir une simulation » en français et la phrase « Open a simulation » en anglais. Toutes les lignes de textes sont ainsi associé à leur identifiant dans un fichier de langue (que l'on appelle locale) situé dans un dossier de ressource. Chaque fichier contient également le nom de la langue, il nous suffit alors, comme pour les vue, de charger tous les fichiers du dossier afin de récupérer toutes les langues. Pour rajouter une langue au simulateur il faut donc simplement glisser dans le bon dossier un nouveau fichier de langue, sans avoir besoin de modifier le code source.

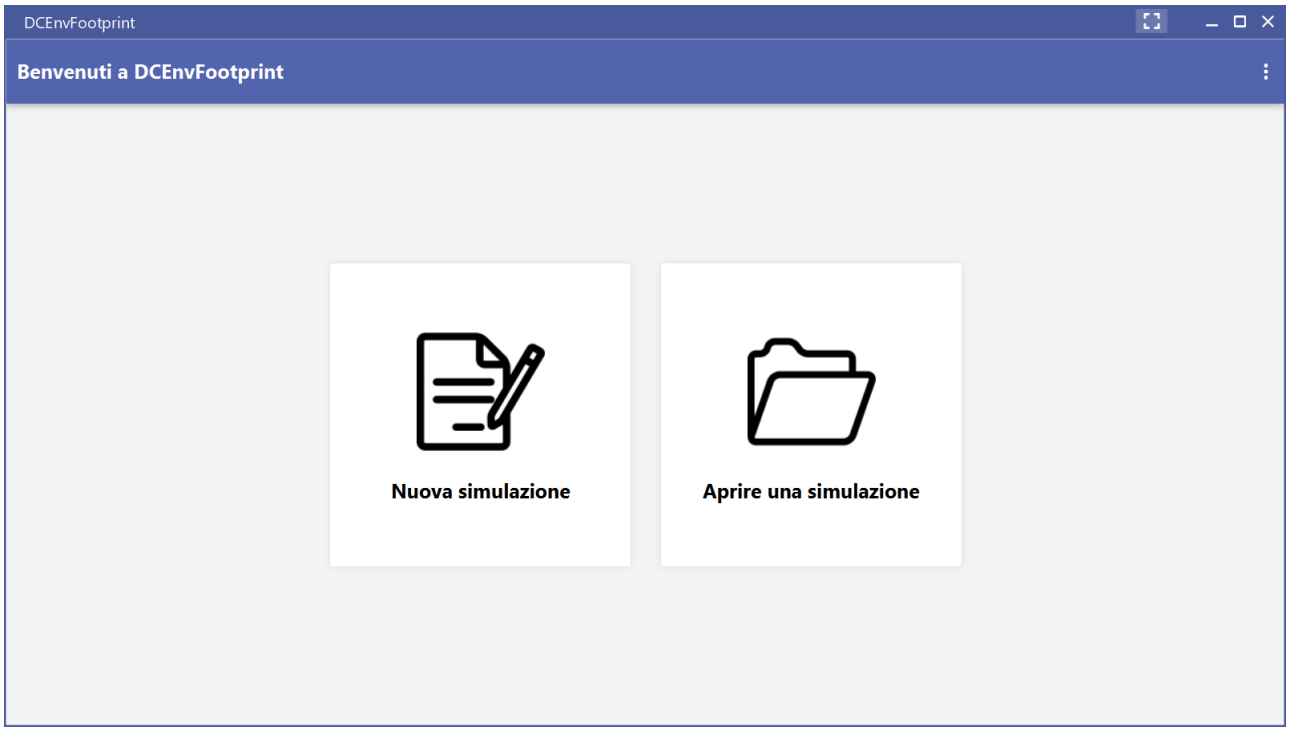

FIGURE 5 – Le simulateur réglé en italien

Le choix de la langue de l'utilisateur, qui se fait dans un menu en haut à droite, est sauvegardé. Pour ce faire nous utilisons un fonctionnalité de Java : les `Preferences`, qui permettent de stocker des couples clefs-valeurs très facilement sans se soucier du système d'exploitation. Ainsi lorsque l'utilisateur ferme et ré-ouvre le simulateur, la langue est reste inchangée. Ainsi le simulateur est disponible en anglais (plus ou moins exact) et en français.

2.3.6 La réflexion autour de l'ergonomie

Comme l'interaction avec les utilisateurs finaux à été quasiment inexistence, j'ai du pousser un peu plus loin la réflexion sur l'ergonomie que sur un projet plus classique où les retours réguliers des utilisateurs auraient permis de l'affiner. Le simulateur possède donc 2 menus, un menu latéral sur la gauche qui rappelle le nom de la simulation et affiche toutes les étapes de successives que l'utilisateur va suivre. L'utilisateur peut revenir à une étape qu'il a passé, mais ne peux pas aller sur les étapes grisés qui représente celles qu'il n'a pas encore remplis. J'ai également disposé un

deuxième menu dans le coin supérieur droit qui permet d'effectuer des actions simples comme créer une nouvelle simulation, en ouvrir une existante, sauvegarde la simulation courante, changer les paramètres de langues, afficher des informations générales sur le simulateur et quitter l'application.

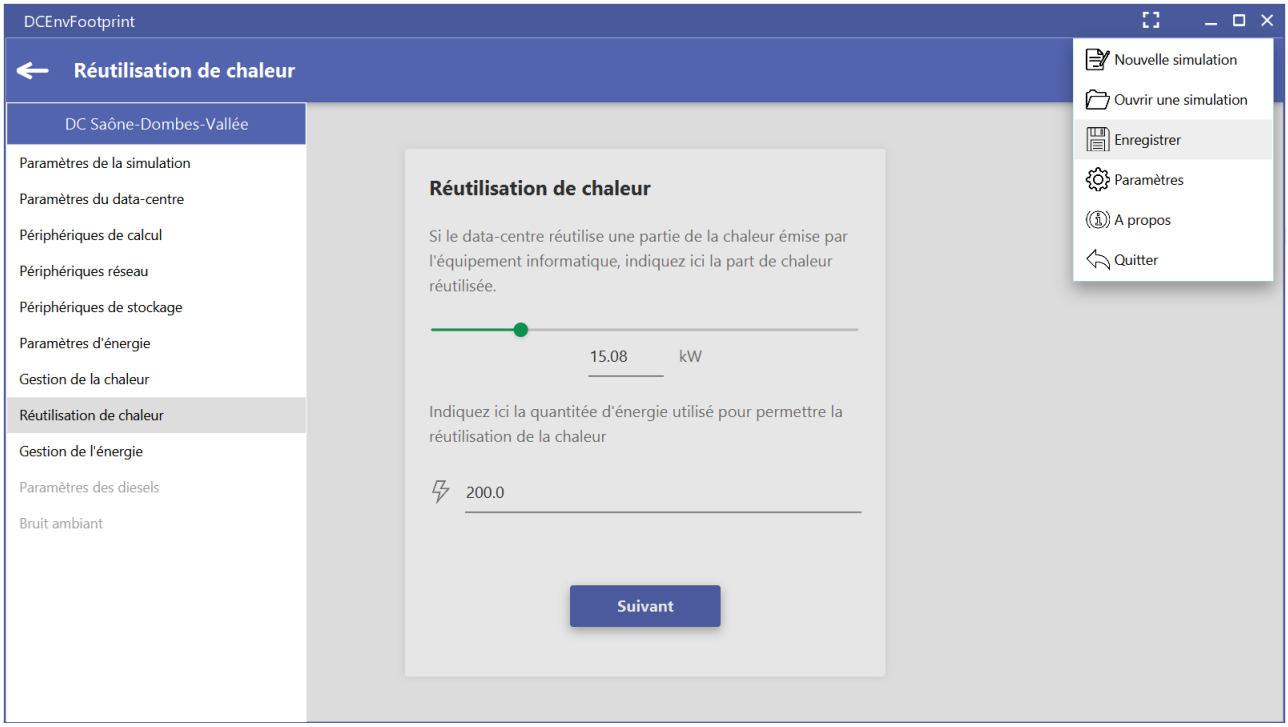

FIGURE 6 – Les menus du simulateur

Comme on peut le voir dans cet exemple, j'ai également fait le choix d'utiliser des composant d'interface graphique facile d'utilisation comme le slider tout en laissant à l'utilisateur le choix de précision avec un champs de texte qui est mis à jour et qui met à jour la valeur du slider ce qui permet un bon compromis.

2.3.7 Les éléments d'interface personnalisés

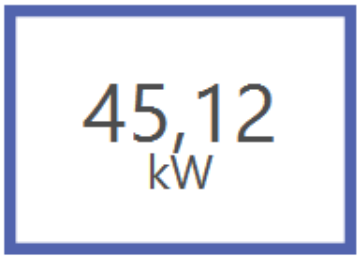

FIGURE 7 – Un indicateur de valeur avec unité

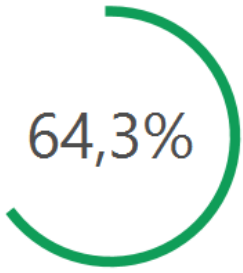

FIGURE 8 – Un indicateur de pourcentage

Afin de coller au mieux a ce que j'imaginais pour l'affichage de certains indicateur, j'ai dû personnaliser certains éléments d'interface. J'ai réécris des partie du code source d'éléments graphiques déjà implémenté dans JavaFX ou dans JFoenix et je les ai modifié pour que je puisse les utiliser par la suite. Ainsi j'ai créé un indicateur de valeur avec une unité personnalisable, qui affiche automatique le bon ordre de grandeur de la valeur stocké : si 1000 W sont stocké l'indicateur

affichera 1 kW. J'ai également créé cette afficheur de pourcentage qui est basé sur un élément graphique représentant un temps de chargement.

J'ai également modifié le comportement d'une bonne partie d'éléments graphiques, sans les récréer complètement, afin qu'ils effectuent le comportement voulue. J'ai par exemple modifié les listes déroulante pour quelles affichent un icône, un titre et un description là où elle ne peuvent normalement n'afficher qu'un petit libellé. Pour ce faire j'ai réécris le code source des `Factory` de ces éléments. Une `Factory` est un module de code source qui crée applique à un élément son comportement, par exemple une `CellFactory` gère le comportement des cellules d'un tableau. La difficulté de cet exercie réside dans le fait qu'il faut s'insipirer du code source original pour espérer ne pas casser le fonctionnement de base de l'élément graphique. La librairie JFoenix est certes bien documenté, mais sa documentation ne possède pas un niveau de détail assez précis pour ce genre d'action, il est donc nécessaire faire appel à la connaissance de la communauté d'utilisateur de la librairie, notemment via des forums de discussion comme *StackOverFlow*.

2.3.8 La gestion des bases de données

Pour certaines fonctionnalités le simulateur à besoins d'informations issues de base de données. C'est notamment le cas pour récupérer le poste électrique le plus proche dans le calcul de l'Indice d'Efficacité du Transport Electrique ou pour récupérer l'efficacité énergétique du matériel depuis la base de données *SPEC*.
Ces données sont le plus souvent stockés dans des structures de données facilement lisible par l'humain, mais qui ne sont pas très performant pour leur exploitation automatique par un algorithme, comme par exemple les classeurs Excel. Pourtant le simulateur à besoin de faire des requêtes sur ces données pour extraire ce dont il a besoin.

Quoi de mieux pour faire des requêtes que d'utiliser le langage SQL que nous avons appris à l'IUT ? Afin d'exploiter les bases de données j'ai utilisé la technologie *SQLite* qui est, comme son nom l'indique, une version légère de SQL. Son principal avantage est de pouvoir lire un base de données depuis un fichier directement sur la machine du client : donc pas besoin de serveur SQL pour répondre aux requêtes. De plus *SQLite* propose un outils en ligne de commande qui permet de transformer les fichiers `.csv` en fichier `.sqlite`, j'ai donc pu convertir facilement les bases de données dans le bon format pour pouvoir les utiliser par la suite.

2.3.9 Le calcul des indicateurs

Le calcul des indicateurs est au centre du simulateur, c'est ce qu'il va permettre de générer le rapport et de d'évaluer les l'empreinte environnementale des centres de donnes. Ainsi tous les indicateurs sont codé « en dure » dans le code source. Tous les indicateurs possèdent une architecture similaire. Chacun prend des paramètres d'entrées et génère un résultat qui est stocké afin de ne pas avoir à recalculer l'indicateur à chaque fois que l'on en a besoin. Toutes les étapes de calcul sont détaillés dans un dictionnaire associant une ligne de texte à une valeur afin de pouvoir retracer chaque étape du calcul de l'indicateur. Ainsi dans le rapport généré nous pouvons détailler

assez précisément chaque étape de calcul. Les indicateurs possèdent également une source, sous forme d'adresse web, que l'on intégrera au rapport. L'utilisateur n'aura qu'a cliquer sur le lien pour avoir toutes les informations techniques relatives au calcul de l'indicateur.

2.3.10 La génération du rapport

Pour générer le rapport j'avais tout d'abord pensé à utiliser des outils de Business Intelligence puissants intégré en Java. Je pensais que l'utilisation de ces outils allaient me simplifier la tâche. C'était sans compter sur leur complexité et leurs concepts qui ne collaient pas totalement au résultat que je souhaitait avoir. J'ai notamment essayé la librairie *JasperReport*, ce qui m'a fais perdre pas mal de temps. J'ai alors complètement changé de méthode, en partant du principe que le plus simple serait le mieux.
J'utilise désormais un fichier HTML qui me sert de modèle, cela me permet de faire la mise en forme du rapport directement dans le fichier modèle et non pas lors de la génération. Ce fichier contient un éventail de mots clefs, facilement identifiables dans son code source. Pour générer le rapport il me suffit de lire le fichier modèle et de remplacer tous les mots clefs par des valeurs calculés.
J'ai distingué deux types de mots clefs. Tout d'abord les mots clefs de langue, entouré par des accolades (e.g. {`SIMULATION_AUTHORS`}) qui correspondent à des entrées de textes présents dans les fichiers de langue. Ainsi le rapport peut être généré selon la langue choisie par l'auteur. Pour ce faire je recherche dans le fichier HTML toutes les entrées de langue présents dans le fichier de locale, et je les remplace par leur valeur associée. Le deuxième type de mots clefs sont les mots clefs de valeur (e.g `TOTAL_SERVER_AMOUNT`), non entouré d'accolade, ils sont remplacés au cas par cas par le simulateur selon ce qu'il doit être affiché.

Enfin, pour pouvoir générer un rapport au format PDF, j'enregistre sur le disque le modèle remplis avec les bonne valeur et j'utilise la librairie *iText* qui me permet de convertir un fichier HTML en un fichier PDF.

2.3.11 Créer un fichier exécutable

Pendant toute la phase de développement, je lance le simulateur depuis mon IDE Eclipse. Cependant si je souhaite le distribuer il faut le compiler dans un fichier exécutable. Pour ce faire j'au utilisé la librairie Maven qui me permet de compiler tout le code source dans un seul fichier exécutable : le fichier `.jar`. Je pensais que cette phase allait se passer sans encombres, mai j'ai fais face à quelques problèmes.

Tout d'abord, pour charger mes ressources : images, fichiers vues ou fichiers de langues. Je recherchais dans le dossier sur le disque comprenant ses fichiers. Lorsque le simulateur est compilé en un seul fichiers, ces fichiers de ressources ne sont plus dans un dossier spécifique sur le disque mais dans un dossier compressé situé à l'intérieur du fichier compilé. Je ne peux donc pas accéder au ressources avec les moyens habituels. Pour palier à ce problème j'ai utilisé la lecture par flux : au lieu de lire un fichier physique, je lis un flux de données extrais du fichier `.jar`.

J'ai également fais face à un problème avec le chargement et la sauvegarde des simulation. Lorsque que je lançais le simulateur à partir du fichier exécutable compilé j'ai remarqué que certains caractères ne s'affichaient pas correctement. Pour régler ce problème j'ai forcer le simulateur à écrire et lire les fichiers de ressource dans un encodage spécifique : l'*UTF-8*. Cet encodage est l'encodage le plus utilisé aujourd'hui et permet de coder la plupart des caractères utilisés dans le monde sans encombre. J'en ai déduis que l'environement dans lequel le simulateur était lancé depuis mon IDE avait comme encodage par défaut l'*UTF-8*, alors que quand je lance le simulateur depuis le fichier exécutable l'encodage par défaut est celui du système d'exploitation (pour Windows il s'agit de l'*ANSI*). Comme je n'avais pas donné de directive particulière pour l'encodage, le simulateur à simplement « choisi » l'encodage par défaut.

2.3.12 Faire face aux bugs des librairies

Pour fonctionner le simulateur utilise 7 librairies tierces pour réaliser différentes fonctions. Ces librairies sont souvent le fruit d'un travail dans le cadre non professionnel par des développeur passionnés qui souhait ajouter leur pierre à l'édifice du logiciel libre. Ainsi certains librairies ne sont pas forcément maintenu régulièrement ou beaucoup testé avant leur mise en production. Fatalement ces librairies contiennent donc des bugs que le développeur qui les utilise ne pas corriger. Cela a été le cas avec le simulateur où j'ai par exemple découvert un bug de la librairie *JFoenix* : les éléments graphique représentant un chargement (les *spinner*) peuvent normalement rester fixe en se stopper à une valeur bien précise. Cependant lorsque l'on recharge l'écran, ce comportement disparait ! On se retrouve alors avec des *spinner* qui tournent perpétuellement.

Afin que ce bug soit corriger dans le prochaine version de *JFoenix*, j'ai créé un ticket sur la page GitHub afin de signaler le bug. Les développeur de l'application y ont répondu et grâce à me contribution ce bug sera corrigé dans la prochaine version de *JFoenix*.

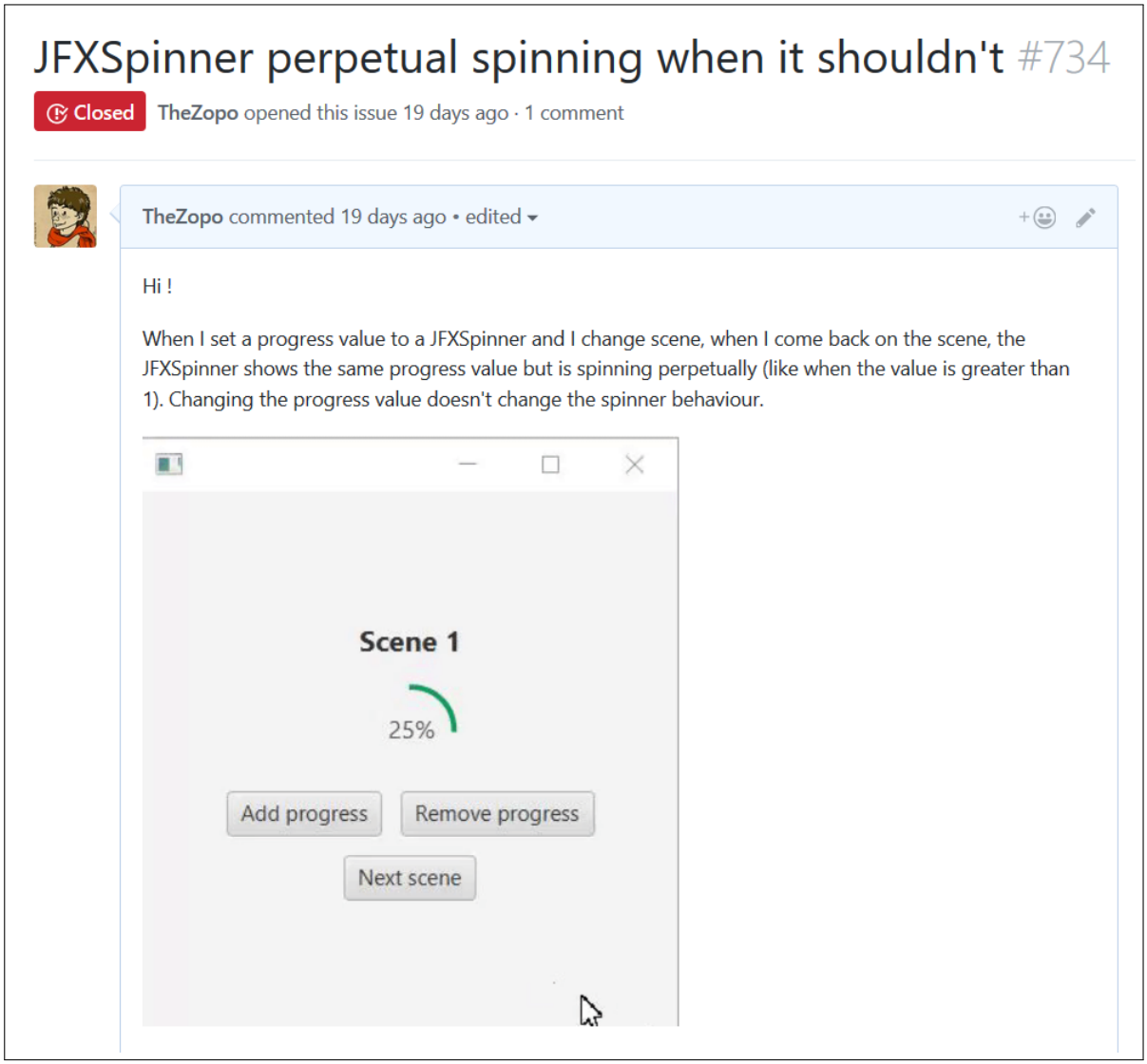

FIGURE 9 – Une partie du ticket que j'ai créé sur la page de JFoenix

3 Mon expérience de stage

TODO : Rédiger l'introduction de la partie

3.1 Bilan du travail effectué

A l'issue de ce stage le simulateur est quasiment terminé, il implémente pêle-mêle les fonctionnalités suivantes :

- Créer, ouvrir et sauvegarder une simulation identifié par un nom, une version et des auteurs,
- Implémente la gestion de la langue et permet de changer de l'une à l'autre à la volée,
- Reprend là où l'utilisateur s'est arrêté lors du chargement d'une simulation,
- Permet de choisir l'emplacement du centre de données depuis une adresse, des coordonnées GPS ou au clic sur une carte,
- L'utilisateur peut rentrer l'ensemble de son équipement de calcul informatique aidé par une autocomplétion des marques, des modèles et de la consommation des serveurs,
- L'utilisateur peut entrer la liste de son équipement réseau et de stockage,

— Calcule automatique la dissipation de chaleur créé par l'équipement informatique,
— L'utilisateur peut choisir et combiner des type de refroidissement selon ses besoins,
— Calcule automatique la quantité d'électricité nécessaire au fonctionnement du centre de données,
— Permet à l'utilisateur de choisir et combiner des sources d'énergie utilisés pour répondre aux besoins électriques,
— Génère un rapport détaillé au format PDF qui affiche le détail de tous les paramètres entrés par l'utilisateur durant la simulation et intègre le calcul d'un certain nombres d'indicateur de Green IT pour critiquer le centre de données.

Conclusion

Glossaire

algorithme distribué Algorithme s'éxécutant, généralement en parallèle, sur plusieurs sites. 8

analyse combinatoire Domaine des mathématiques étudiant les configurations de collections finies d'objets ou d'ensembles et le dénombrement. 9

benchmark Méthode visant à comparer différents équipements de même nature sous la même contrainte. 20

cluster Ensemble de serveurs indépendants regroupés en une seule entité pour l'utilisateur. 11, 14

complexité En informatique, désigne la quantité de ressources néscéssaire à l'éxécution d'un algorithme. 9

datacenter Centre de données. 11

dataflow Flux de données, indique que les données sont actives et traversent un programme de manière asynchrone contrairement à une architecture classique où elles attendent leur tour chargées en mémoire [28]. 8

Eclipse IDE multiplateforme et multilangage. 1

ENS École Normale Suppérieure. 5, 6

flop En informatique, réprésente une oppération mathématique effectuée par le processeur. 19

GIT Protocole de gestion de version centralisé, permet de stocker du code source en conservant la chronologie de toutes les modifications. 1

GitHub Plateforme en ligne de gestion de version utilisant le protocole GIT. S'est imposé en tant que réseau social pour développeur. 1

IDE Environement de développement intégré, ensemble d'outils dédiés au développement regroupés dans un même logiciel. 1

Inria Institut National de Recherche en Informatique et en Automatique. 6–10

JAVA Langage de programmation orienté objet et multiplateforme. 1, 20

JavaFX Bibliothèque interne à JAVA gérant l'interface graphique utilisateur. 1

LIP Laboratoire de l'Informatique du Parallélisme. 5–7

load-balancing Ensemble de techniques visant à distribuer une charge de travail entre différents serveurs. 11

Maven Outils de gestion de production. Facilite la gestion de bibliothèques. 1

repository Un dépôt centralisé et organisé de code source. 1

système sur puce Systeme embarqué sur une seule puce électronique. 8

travail utile En informatique, réprésente l'ensemble des opérations néscéssaire à la réalisation d'une tâche. 18–20

Unité de Recherche Associé Structure de recherche qui relève d'un autre organisme que le CNRS dans laquelle le CNRS lui-même est impliqué [5]. 6

Unité Mixte de Recherche Structure de recherche placée sous la responsabilité conjointe du ministère de la recherche et du CNRS [5]. 6

Table des figures

1 Organigramme du LIP au 10 Avril 2018 [6] 7
2 Logo de la plateforme Grid'5000 14
3 Les classes énergétiques du PUE 18
4 Schéma du modèle MVC 25
5 Le simulateur réglé en italien 27
6 Les menus du simulateur 28
7 Un indicateur de valeur avec unité 28
8 Un indicateur de pourcentage 28
9 Une partie du ticket que j'ai créé sur la page de JFoenix 32

Liste des tableaux

1 Calcul du Datacenter Energy Consumption Gauge (DC_G) 22
2 Calcul du Datacenter Performance (DC_P) 23

Sources

[1] Université Claude Bernard Lyon 1. Unité mixe de recherche 5668 - laboratoire de l'informatique du parallélisme (lip). https://www.univ-lyon1.fr/recherche/entites-de-recherche/laboratoire-de-l-informatique-du-parallelisme-lip--618125.kjsp, 2018.

[2] ASHRAE TC 9.9. 2011 thermal guidelines for data processing environments – expanded data center classes and usage guidance. http://ecoinfo.cnrs.fr/IMG/pdf/ashrae_2011_thermal_guidelines_data_center.pdf, 2011.

[3] Aurélie Barbaux. Plainte pour nuisances sonores contre un data center à paris. *L'usine digitale*, 09 Janvier 2016.

[4] Fabrice Coquio. Tribune dans green it « efficacité énergétique : que représente vraiment le pue ? ». https://www.greenit.fr/2014/04/04/efficacite-energetique-que-represente-vraiment-le-pue/, 4 Avril 2014.

[5] Direction de la recherche de Grenoble INP. Les labels des unités de recherche. pages 1–2, 19 Juillet 2012.

[6] Laboratoire de l'Informatique du Para. Organigramme du lip. http://www.ens-lyon.fr/LIP/index.php/organization-chart, 10 Avril 2018.

[7] RTE Réseau de transport de l'électricité. Chiffres clés. http://www.rte-france.com/fr/eco2mix/chiffres-cles, 2017.

[8] RTE Réseau de transport de l'électricité. Données d'infrastructures. https://opendata.rte-france.com/explore/?sort=modified&refine.keyword=Infrastructure, 2017.

[9] Pierre FRAIGNIAUD. Évaluation du HCERES sur l'unité : Laboratoire de l'Informatique du Parallélisme. *Haut Conseil de la Recherche et de l'Enseignement Supérieur*, pages 5–6, 2015.

[10] Communauté Grid'5000. Grid5000 :home. https://www.grid5000.fr/mediawiki/index.php/Grid5000:Home, 3 Mai 2018.

[11] Curt Harting. Ac transmission line losses. http://large.stanford.edu/courses/2010/ph240/harting1/, 24 Octobre 2010.

[12] European Telecommunications Standards Institute. Etsi gs oeu 001 v2.1.1. http://www.etsi.org/deliver/etsi_gs/OEU/001_099/001/02.01.01_60/gs_OEU001v020101p.pdf, Décembre 2014.

[13] Planetoscope. Energie consommée par les data centers. https://www.planetoscope.com/electronique/230-energie-consommee-par-les-data-centers.html, 2015.

[14] Équipe AriC. Aric : Arithmetic and computing. http://www.ens-lyon.fr/LIP/AriC/wp-content/uploads/2014/09/Slides_eval_Aeres_Aric.pdf, 2018.

[15] Équipe Avalon. Activity report 2012. https://raweb.inria.fr/rapportsactivite/RA2012/avalon/avalon.pdf, 2013.

[16] Équipe Avalon. The avalon research team. https://avalon.ens-lyon.fr/, 2018.

[17] Équipe CASH. Cash team. http://www.ens-lyon.fr/LIP/CASH/, 2018.

[18] Équipe DANTE. Team presentation. https://team.inria.fr/dante/, 2018.

[19] Équipe MC2. Mc2 : Modèles de calcul, complexité, combinatoire. http://www.ens-lyon.fr/LIP/MC2/, 2018.

[20] Équipe PLUME. Preuves & langages : Un manège enchanté. http://www.ens-lyon.fr/LIP/PLUME/, 2018.

[21] Équipe ROMA. Roma. http://www.ens-lyon.fr/LIP/ROMA/, 2018.

[22] Tom Raftery. Power usage efficiency (pue) is a poor data center metric. http://greenmonk.net/2011/12/13/power-usage-efficiency-is-a-poor-data-center-metric/, 13 Décembre 2011.

[23] Standard Performance Evaluation Corporation (SPEC). Specpower_ssj2008 result file fields. https://www.spec.org/power/docs/SPECpower_ssj2008-Result_File_Fields.html, 19 Novembre 2014.

[24] Standard Performance Evaluation Corporation (SPEC). Specpower_ssj2008. https://www.spec.org/power_ssj2008/results/power_ssj2008.html, 2007-2018.

[25] Jayabalan Subramanian. Going beyond power usage effectiveness (pue) for data center efficiency. http://www.datacenterjournal.com/going-beyond-power-usage-effectiveness-pue-for-data-center-efficiency/, 30 Mai 2012.

[26] Communauté Wikipédia. Effet corona. https://fr.wikipedia.org/wiki/Effet_corona, 1 Avril 2018.

[27] Communauté Wikipédia. Résitivité. https://fr.wikipedia.org/wiki/R%C3%A9sistivit%C3%A9, 1 Avril 2018.

[28] Communauté Wikipédia. Architecture dataflow. https://fr.wikipedia.org/wiki/Architecture_Dataflow, 29 Novembre 2016.

[29] Communauté Wikipédia. Effet de peau. https://fr.wikipedia.org/wiki/Effet_de_peau, 6 Mars 2018.

Annexes

A Calcul du KPI Energy Consumption (KPI_{EC}) 41

A.1 Consommation électrique du réseau (EC_{SP}) 41

A.2 Consommation électrique d'énergie fossile (EC_{FEN}) 41

A.3 Consommation électrique d'énergies renouvelables (EC_{REN}) 41

A.4 Consommation d'énergie thermique (EC_{TH}) 41

A.5 Facteur de conversion d'énergie thermique ver électrique (K_{TH}) 41

B Calcul de l'énergie perdue dans les lignes HTA 42

B.1 L'effet de peau 42

B.1.1 La résistivité du câble 42

B.1.2 L'épaisseur de peau du câble 43

B.1.3 L'inductance par unité de longueur 43

B.2 L'effet corona 44

Annexe A

A Calcul du KPI Energy Consumption (KPI_{EC})

Le KPI_{EC} est définie par la formule suivante que nous allons détailler :

$$KPI_{EC} = EC_{SP} + EC_{FEN} + EC_{REN} + (EC_{TH} \times K_{TH})$$

A.1 Consommation électrique du réseau (EC_{SP})

Il s'agit de la quantité d'énergie obtenue depuis le réseau électrique et qui est relevée sur les compteurs du fournisseur d'énergie. Cette valeur prend également en compte l'énergie issue d'une boucle interne de distribution d'électricité et les pertes dans les transformateurs.

A.2 Consommation électrique d'énergie fossile (EC_{FEN})

Il s'agit de la quantité d'énergie produite localement à partir de sources d'énergie fossile, comme par exemple des groupes électrogènes. Elle est mesurée à la sortie de la source d'énergie si elle est dédiée au data-centre ou à l'entrée de ce dernier si la source est partagée avec d'autres sites.

A.3 Consommation électrique d'énergies renouvelables (EC_{REN})

Il s'agit de la quantité d'énergie produite localement à partir de sources renouvelables. Le mode de mesure est le même que pour l'EC_{FEN}.

A.4 Consommation d'énergie thermique (EC_{TH})

Il s'agit de la quantité d'énergie thermique livrée, quelle soit chaude ou froide. On la mesure à l'entrée du data-centre via un compteur de calories.

A.5 Facteur de conversion d'énergie thermique ver électrique (K_{TH})

Il s'agit du facteur de conversion s'il est connu et certifié, si ce n'est pas le cas on utilisera la valeur par défaut $0,43$ qui correspond à une installation de référence.

Annexe B

B Calcul de l'énergie perdue dans les lignes HTA

B.1 L'effet de peau

L'effet de peau est un phénomène électromagnétique faisant qu'a haute fréquence, le courant est transmis dans la section du conducteur la plus proche de la surface. La densité du courant décroît exponentiellement à mesure que l'on s'éloigne de la périphérie du conducteur ce qui entraine une augmentation de la résistance de ce dernier et donc des pertes d'énergie.

La quantité d'énergie perdu est définie par la formule suivante :

$$P_{peau} = (1 - exp(\frac{-dR_l}{Lc})) \times P_{nominale}$$

Paramètre	Définition et unité
P_{peau}	Énergie perdue par effet de peau (kW)
d	Longueur du câble (m)
R_l	Résistivité du câble (Ω/m)
L	Inductance par mètre (H/m)
c	Célérité de la lumière dans le vide (3×10^8 m/s)
$P_{nominale}$	La puissance nominale de la ligne (kW)

B.1.1 La résistivité du câble

La résistivité du câble représente sa capacité à s'opposer à la circulation du courant électrique. Cela correspond à la résistance d'un tronçon du matériaux conducteur du câble d'un mètre de longueur et d'un mètre carré de section [27].

Elle est définie par la formule suivante :

$$R_l = \frac{I_B}{2\pi a \sigma \delta}$$

Paramètre	Définition et unité
R_l	Résistance par unité de longueur (Ω/m)
I_B	Coefficient de correction de Bessel (1,1)
a	Rayon du câble (m)
σ	Conductivité du métal (S/m)
δ	Épaisseur de peau du câble (m)

B.1.2 L'épaisseur de peau du câble

L'épaisseur de peau du câble détermine la largeur de la zone où se concentre le courant dans le câble.
Elle est définie par la formule suivante :

$$\delta = \frac{1}{\sqrt{\pi f \mu_0 \sigma}}$$

Paramètre	Définition et unité
δ	Épaisseur de peau du câble (m)
f	Fréquence du courant (Hz)
μ_0	Perméabilité magnétique du vide ($4\pi \times 10^{-7}$ H/m)
σ	Conductivité du métal (S/m)

B.1.3 L'inductance par unité de longueur

L'inductance est la quantification de l'énergie électromagnétique qui apparait lorsque le courant parcourt le câble.

Elle est définie par la formule suivante :

$$L = \frac{\pi}{\mu} \ln(\frac{d}{a})$$

Paramètre	Définition et unité
L	Inductance par mètre (H/m)
μ	Perméabilité magnétique ambiant ($\approx \mu_0$) (H/m)
d	Espace entre deux lignes (*A vérifier*) (m)
a	Rayon du câble (m)

B.2 L'effet corona

L'effet corona est dû à l'ionisation des molécules d'air le long des lignes électriques. Une partie de l'énergie est dissipée dans l'air. Cet effet varie selon les conditions métrologiques.

La quantité d'énergie perdu par effet corona est définie par la formule suivante :

$$P_{corona} = \frac{k_0}{k_d}(f+25)\sqrt{\frac{a}{d}}[V_0 - g_0 k_i a k_d \ln(\frac{d}{a})]^2 \times 10^{-5} \times l$$

Paramètre	Définition et unité
k_0	Constante (241)
k_d	Densité de l'air (hPa)
f	Fréquence (Hz)
a	Rayon du câble (cm)
d	Espace entre deux lignes (*A vérifier*) (cm)
V_0	Tension phase neutre ($Tension\ ligne\ ligne$/1.73 kV)
g_0	Rigidité diélectrique de l'air (21 kV/cm)
k_i	Facteur de correction du câble (*Se référer aux tables constructeur*)
l	Longueur du câble (km)

Rapport de stage - 2eme année
DUT Informatique

Simuler l'empreinte environnementale des centres de données

Bastien Marsaud
9 Avril - 15 Juin 2018

Résumé

Étudiant en deuxième année de DUT informatique à l'IUT Lyon 1, j'ai effectué mon stage de fin de cursus au Laboratoire de l'Informatique du Parallélisme à l'ENS Lyon. Très intéressé par le milieu de la recherche et par les problématiques environnementale, j'ai initié la conception d'un simulateur d'empreinte environnementale des centres de données. Ce simulateur permettra, à terme, d'aider les urbanistes et les architectes dans la conception et l'intégration des centre de données sur le territoire. Il sera capable, à partir d'une multitude de données entrées par l'utilisateur, de générer automatiquement un rapport complet présentant les caractéristiques du centre de données ainsi que ses scores pour les différents indicateurs permettant d'évaluer son empreinte environnementale.
Dans ce rapport de stage vous découvrirez le Laboratoire de l'Informatique du Parallélisme en tant qu'organisation, les différentes étapes du projet : de la phase de découverte du sujet et de conception d'une bibliographie à la phase de développement, mais également mon ressenti sur cette expérience dans le monde de la recherche et ce que j'ai découvert techniquement et humainement.

Mots clefs : informatique, green IT, datacenter, recherche, environnement, énergie, simulation, économie, urbanisme

IUT Lyon 1

Lip