ENS DE LYON

THESE

pour l'obtention du grade de Docteur, délivré par

l'ECOLE NORMALE SUPERIEURE DE LYON

Ecole Doctorale N°512
InfoMaths - Informatique et Mathématiques de Lyon

Discipline : INFORMATIQUE (Informatique)

Soutenue publiquement le 19 juin 2026, par :

Maxime AGUSTI

Modélisation et estimation de la consommation électrique en environnement bare metal : une approche pragmatique et non-intrusive pour le suivi en milieu industriel

Modeling and estimating power consumption in bare-metal environments: A pragmatic and non-intrusive approach for monitoring in industrial settings

---

Après avis de :

Patricia STOLF, Professeure des universités, IRIT	Rapporteure
Jean-Marc MENAUD, Professeur des universités, IMT-Atlantique	Rapporteur

Devant le jury composé de :

Sophie CHABRIDON, Professeure, Telecom SudParis	Présidente
Patricia STOLF, Professeure des universités, IRIT	Rapporteure
Jean-Marc MENAUD, Professeur des universités, IMT-Atlantique	Rapporteur
Anne-Cécile ORGERIE, Directrice de recherche, Irisa	Examinatrice
Sébastien MONNET, Professeur des universités, Université Savoie Mont Blanc	Examinateur
Eddy CARON, Professeur des universités, Université Claude Bernard Lyon 1 / ENS de Lyon	Directeur de thèse
Laurent LEFÈVRE, Directeur de recherche, INRIA, ENS Lyon	Co-directeur de thèse

Laboratoire de l'Informatique du
Parallélisme
ENS de Lyon
46, allée d'Italie
69364 LYON CEDEX 07

École doctorale InfoMaths
7 avenue Jean Capelle
Bât. Blaise Pascal
69621 VILLEURBANNE Cedex

Résumé

La croissance de la demande en calcul numérique et l'essor de l'intelligence artificielle placent l'efficacité énergétique au cœur des enjeux stratégiques des centres de données. Si la mesure de la consommation globale est une pratique courante, l'obtention d'une granularité fine à l'échelle du serveur individuel se heurte à des verrous économiques et opérationnels. Le déploiement massif de wattmètres physiques représente un coût prohibitif, tandis que les estimations logicielles traditionnelles nécessitent un agent de collecte dans le système d'exploitation, rendue impossible dans les environnements de serveurs *bare metal* (machines physiques louées sans couche de virtualisation intermédiaire) ou de GPU en *pass-through* en raison des impératifs de sécurité et d'isolation.

Cette thèse propose une approche de métrologie énergétique pragmatique et non-intrusive, fondée sur l'exploitation des canaux auxiliaires thermiques. L'hypothèse centrale est que la température des composants, accessible via les interfaces de gestion matérielle standard, constitue une variable de substitution fiable pour reconstituer la consommation électrique sans outrepasser l'étanchéité entre l'opérateur et le client.

Nos travaux s'articulent autour de trois contributions réalisées sur l'infrastructure de production d'OVHcloud. Premièrement, nous introduisons la méthode PowerHeat, capable d'estimer la puissance d'un serveur *bare metal* via la température de son processeur. Cette approche intègre une compensation spécifique pour neutraliser l'inertie et le bruit thermique inhérents aux systèmes de refroidissement liquide.

Deuxièmement, pour lever le verrou du passage à l'échelle sur un parc hétérogène et non instrumenté, nous développons la méthodologie PPEM-BM. En combinant une bibliothèque de modèles de référence et des techniques d'apprentissage du classement, cette méthode permet de transférer la capacité d'estimation vers des milliers de serveurs dépourvus de capteurs de puissance en sélectionnant dynamiquement le modèle le plus pertinent selon le profil matériel et thermique.

Enfin, nous étendons cette approche aux GPU exploités en mode *pass-through*. Face à l'opacité de ces composants pour l'hébergeur, notre modélisation thermique permet de retrouver une visibilité sur leur consommation et leur taux d'usage réel. L'application de cette méthode sur un cluster de production révèle notamment une sous-utilisation massive des ressources, ouvrant la voie à de nouvelles stratégies de consolidation.

En transformant une dissipation thermique en source d'information, cette thèse démontre la faisabilité d'une observabilité énergétique logicielle précise, déployable massivement sans surcoût matériel.

Table des matières

Table des figures v

1 Introduction 1
- 1.1 Contexte . . . 1
- 1.2 Problématique de recherche . . . 2
- 1.3 Contexte industriel . . . 4
- 1.4 Contributions . . . 5
- 1.5 Publications . . . 6
- 1.6 Organisation du manuscrit . . . 6

2 État de l'art 9
- 2.1 La mesure de consommation énergétique dans les centres de données . . . 10
  - 2.1.1 Enjeux stratégiques et normatifs . . . 10
  - 2.1.2 Optimisation de l'infrastructure et gestion des capacités . . . 10
  - 2.1.3 Maîtrise des coûts et efficacité opérationnelle . . . 11
  - 2.1.4 Méthodes de mesure et d'estimation . . . 12
  - 2.1.5 Limites et opportunités . . . 13
- 2.2 Le défi de l'estimation en environnement *bare metal* . . . 14
  - 2.2.1 Le modèle de responsabilité partagée du *bare metal* . . . 14
  - 2.2.2 La contrainte de non-intrusivité . . . 14
  - 2.2.3 Estimation de puissance : l'apport des canaux auxiliaires . . . 15
  - 2.2.4 Limites et opportunités . . . 16
- 2.3 Modélisation de la puissance par la température . . . 16
  - 2.3.1 Approches classiques en refroidissement par air . . . 17
  - 2.3.2 Défis physiques : Inertie et courants de fuite . . . 17
  - 2.3.3 La rupture technologique du refroidissement liquide . . . 18
  - 2.3.4 Limites et opportunités . . . 18
- 2.4 Généralisation des modèles de puissance . . . 19
  - 2.4.1 Limites et opportunités . . . 21
- 2.5 Modélisation de puissance des GPU . . . 21
  - 2.5.1 Mécanismes de partage et de virtualisation . . . 21
  - 2.5.2 Défis de la mesure et modélisation logicielle . . . 22
  - 2.5.3 Limites et opportunités . . . 23

3 Modélisation non-intrusive de la puissance d'un serveur *bare metal* par la température du processeur 25
3.1 Introduction . . . 25
3.1.1 Mesure en environnement *bare metal* : un défi complexe . . . 26
3.1.2 La température CPU comme proxy de la puissance . . . 27
3.1.3 Structure du chapitre . . . 28
3.2 Méthodologie : PowerHeat . . . 28
3.2.1 Le banc d'expérimentation . . . 28
3.2.2 Collecte des données . . . 30
3.2.3 Approches de modélisation comparées . . . 30
3.2.4 Algorithme et validation . . . 31
3.3 Résultats et Analyse . . . 32
3.3.1 Observations des mesures thermiques et électriques des serveurs . . . 33
3.3.2 Performance des modèles . . . 35
3.3.3 Analyse de cas spécifiques . . . 37
3.3.4 Discussion . . . 40
3.4 Conclusion . . . 41

4 Généralisation des modèles de puissance : La méthodologie PPEM-BM 45
4.1 Introduction . . . 45
4.2 Méthodologie . . . 46
4.2.1 Constitution de la bibliothèque de modèles . . . 47
4.2.2 L'évaluation croisée pour mesurer la portabilité . . . 47
4.2.3 La fonction de sélection par apprentissage du classement . . . 48
4.2.4 Processus d'estimation . . . 49
4.3 Expérience . . . 49
4.3.1 Plateforme expérimentale . . . 49
4.3.2 Méthodes de comparaison . . . 51
4.3.3 Protocole de validation . . . 52
4.4 Résultats et discussions . . . 53
4.4.1 Performances globales . . . 53
4.4.2 Analyse du modèle LTR . . . 54
4.4.3 Représentativité des résultats . . . 56
4.5 Conclusion . . . 58

5 Modélisation de la puissance des GPU dans le *cloud* : Le cas du *pass-through* 61
5.1 Introduction . . . 61
5.2 Les modes de partage des GPU . . . 63
5.2.1 Allocation temporelle . . . 63
5.2.2 Allocation spatiale . . . 64
5.2.3 Allocation *pass-through* . . . 64
5.3 Inférence par capteurs de températures . . . 65

5.3.1 Acquisition de données via IPMI 65
5.3.2 Dispositifs expérimentaux 65
5.3.3 Sélection des capteurs pertinents 66
5.3.4 Modélisation en environnements refroidis par air 67
5.3.5 Modélisation en environnements refroidis par eau 69
5.4 Cas d'étude : analyse des GPU d'un cluster IaaS 70
5.4.1 Contexte de l'expérimentation et méthodologie 70
5.4.2 Analyse du taux d'occupation réel 70
5.4.3 Limites des métriques classiques 71
5.5 Conclusion 73

6 Conclusion générale **75**
6.1 Limitations et discussions 76
6.2 Perspectives de recherches 76

Bibliographie **79**

Table des figures

1.1 Consommation d'énergie totale des centres de données aux États-Unis [1, 2] 2

2.1 Points de mesure pour l'évaluation du PUE selon la norme ISO 30134 [3]. . 11

3.1 Représentation du banc d'expérimentation et de son système de refroidissement. 29
3.2 Représentation de la méthode d'évaluation des modèles par validation croisée. 32
3.3 Évolution de la consommation électrique et de la température CPU des serveurs sur 5 jours. 33
3.4 Performances des modèles `ABS_TEMP` et `REL_TEMP` pour chaque serveur et coefficient de corrélation de Pearson des entrées des modèles. 35
3.5 Performances des modèles `ABS_TEMP` et `REL_TEMP` en fonction de l'écart-type de puissance (écart-type mobile avec une fenêtre temporelle de 1 heure). 36
3.6 Comparaison des estimations de puissance des modèles `ABS_TEMP` et `REL_TEMP` avec la consommation réelle. 38
3.7 Analyse détaillée du serveur S_{15} : évolution temporelle des signaux (gauche) et corrélation puissance-température (droite). 39
3.8 Analyse détaillée du serveur S_{05} : évolution temporelle des signaux (gauche) et corrélation puissance-température (droite). 39
3.9 Analyse détaillée du serveur S_{09} : évolution temporelle des signaux (gauche) et corrélation puissance-température (droite). 40

4.1 Vue d'ensemble de la méthodologie PPEM-BM. 47
4.2 Répartition des configurations matérielles des 1076 serveurs. 50
4.3 Consommation électrique en fonction de la configuration matérielle 51
4.4 Distribution des taux d'erreur d'estimation de puissance de chaque approche. 54
4.5 Évolution du taux d'erreur en fonction du rang du modèle. 55
4.6 Distributions des configurations matérielles des serveurs. 56
4.7 Performances des approches en fonction de la gamme de serveurs. 57

5.1 Trois paradigmes d'allocation de GPU. 63
5.2 Coefficient de corrélation de Pearson entre les capteurs IPMI et la puissance des GPU. 66
5.3 Températures mesurées avant et après correction (4 × A100) en fonction du taux de charge. 67

5.4 Températures mesurées avant et après correction (8 × A100) en fonction du taux de charge. 68
5.5 Température d'un GPU refroidi à l'eau en fonction de la puissance et du taux d'utilisation. 69
5.6 Fonction de répartition de l'utilisation des GPU sur le cluster IaaS. 71
5.7 Matrice de corrélation des métriques des machines virtuelles. 72

Liste des abréviations

BMC *Baseboard Management Controller*
CPU *Central Processing Unit*
DRAM *Dynamic Random Access Memory*
DTC *Direct-To-Chip*
DVFS *Dynamic Voltage and Frequency Scaling*
GPU *Graphics Processing Unit*
H-PACE *Hardware-Profiled Average Consumption Estimator*
IA Intelligence Artificielle
IaaS *Infrastructure-as-a-Service*
IOMMU *Input-Output Memory Management Unit*
IPMI *Intelligent Platform Management Interface*
LLM *Large Language Model*
LTR *Learning To Rank*
MAPE *Mean Absolute Percentage Error*
MIG *Multi-Instance GPU*
MLaaS *Machine Learning as a Service*
NVML *NVIDIA Management Library*
OS *Operating System*
PDU *Power Distribution Unit*
PMBus *Power Management Bus*
PPEM-BM *Portable Power Estimation Methodology for Bare Metal*
PSU *Power Supply Unit*
PUE *Power Usage Effectiveness*
RAPL *Running Average Power Limit*
TDP *Thermal Design Power*
UPS *Uninterruptible Power Supply*
VFIO *Virtual Function I/O*

VM *Virtual Machine*
XGBoost *eXtreme Gradient Boosting*

Chapitre 1

Introduction

Sommaire

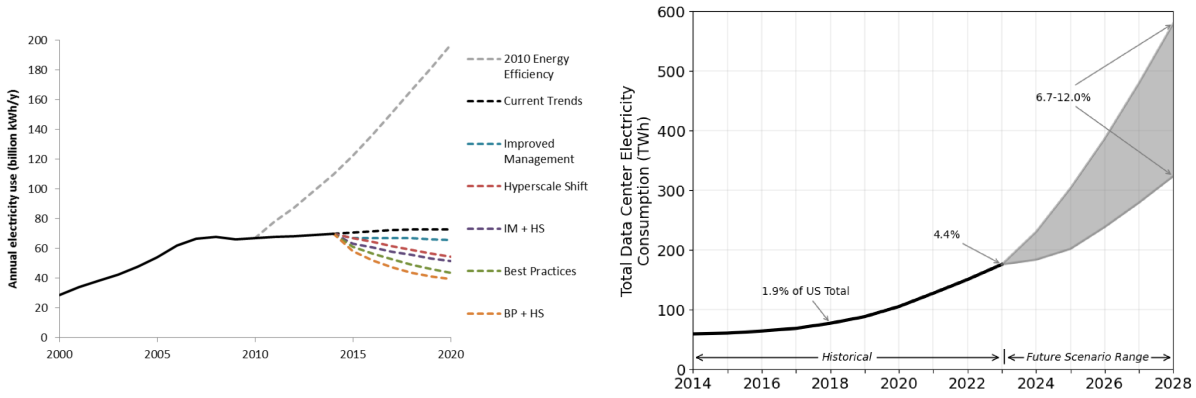
1.1 Contexte 1
1.2 Problématique de recherche 2
1.3 Contexte industriel 4
1.4 Contributions 5
1.5 Publications 6
1.6 Organisation du manuscrit 6

1.1 Contexte

En 2024, les centres de données ont consommé 415 TWh soit environ 1,5 % de la demande mondiale en électricité [4]. L'Agence Internationale de l'Énergie (IEA) projette que la demande pourrait plus que doubler d'ici 2030 pour atteindre 945 TWh. Cette dynamique marque une rupture avec la tendance observée au cours de la précédente décennie. En effet, entre 2010 et 2018, malgré une expansion historique du trafic internet, des charges de travail des serveurs et des capacités de stockage, la consommation électrique des centres de données n'a que très peu augmenté, de l'ordre de 6% sur toute la période, se maintenant autour de 1% de la consommation mondiale d'électricité [5].

La comparaison des projections issues de deux rapports du Lawrence Berkeley National Laboratory sur la consommation des centres de données aux États-Unis (Figure 1.1) illustre cette rupture. En 2016, les prévisions suggéraient une croissance modérée, voire une stabilisation grâce à la généralisation des stratégies d'efficacité énergétique ; certains scénarios envisageaient même une diminution. En 2024, les analyses révèlent une augmentation nette et continue depuis 2018, avec des prévisions indiquant une forte accélération potentielle.

Le principal levier de la maîtrise de la consommation entre 2010 et 2018 fut la transi-



(a) Projections en 2016 [1]

(b) Projections en 2024 [2]

FIGURE 1.1 – Consommation d'énergie totale des centres de données aux États-Unis [1,2]

tion massive des infrastructures d'entreprise, souvent sous-optimisées, vers des centres de données hyperscales. Cette migration a favorisé la consolidation des serveurs grâce à la virtualisation [6], remplaçant de nombreux serveurs physiques sous-utilisés par un nombre plus restreint de machines mieux exploitées. Simultanément, des avancées technologiques majeures, telles que l'adoption de techniques de refroidissement passif (*free cooling*) et les progrès continus en matière d'efficacité des processeurs [7] ont considérablement réduit l'énergie nécessaire pour chaque opération de calcul.

La nouvelle dynamique après 2018 résulte d'une convergence de facteurs. D'abord, les gains d'efficacité matérielle, notamment sur les processeurs et les systèmes de refroidissement ralentissent et ne parviennent plus à eux seuls à compenser l'augmentation globale de la demande [5, 8]. Ensuite, les progrès technologiques passés ont induit un effet rebond : en rendant les services numériques plus performants et moins coûteux, ils ont stimulé l'émergence et l'adoption de nouveaux usages, tels que le streaming vidéo en haute définition, le *cloud gaming* et le *big data*, absorbant ainsi une part significative des économies d'énergie réalisées [9].

À cela s'ajoute désormais la consommation énergétique intense de l'Intelligence Artificielle (IA), et particulièrement les modèles d'IA générative entraînés et exécutés par des accélérateurs spécialisés comme les GPU (Graphics Processing Units) [10, 11]. La démocratisation rapide de ces services d'IA auprès du grand public et des entreprises crée ainsi une demande grandissante en ressources, devenant l'un des principaux moteurs de l'augmentation actuelle de la consommation énergétique des centres de données [4].

1.2 Problématique de recherche

L'augmentation critique de la demande énergétique impose aux centres de données une pression croissante pour maîtriser leur empreinte. Si la mesure de la consommation globale d'un centre de données est aujourd'hui une pratique courante [12], la capacité à descendre à

une granularité fine — celle du serveur individuel ou du composant — reste un défi majeur pour les opérateurs [13]. Pourtant, cette métrologie fine est le prérequis indispensable à l'optimisation de l'infrastructure et au respect des nouvelles normes environnementales.

Le premier obstacle est d'ordre matériel et économique. La mesure directe via des wattmètres matériels au niveau de chaque serveur est le moyen le plus précis, mais son déploiement généralisé est freiné par un coût prohibitif et une grande complexité opérationnelle [14].

Le second obstacle est lié à la nature des alternatives logicielles. Les modèles d'estimation existants, comme les wattmètres logiciels, reposent traditionnellement sur la collecte de métriques d'utilisation du système nécessitant l'installation d'agents au sein du système d'exploitation [15]. Or, dans le contexte des offres *bare metal*, où l'utilisateur bénéficie d'un accès logiciel total, cette approche devient inaccessible. Le modèle de responsabilité partagée impose une frontière stricte : le fournisseur d'infrastructure n'a aucun accès au système du client. Le serveur devient alors une "boîte noire" dont l'activité interne est invisible pour l'hébergeur, rendant les méthodes intrusives impossibles à mettre en œuvre pour des raisons de confidentialité et de sécurité.

Le troisième obstacle réside dans l'hétérogénéité et la scalabilité. Même si une méthode d'estimation est valide pour un serveur donné, la diversité du parc informatique et les disparités environnementales posent le défi de la généralisation [16]. Comment estimer la consommation de milliers de serveurs aveugles (sans capteurs de puissance) sans avoir à calibrer individuellement chaque machine ?

Enfin, l'émergence des GPU pour les charges de travail d'intelligence artificielle ajoute un niveau de complexité. En mode *pass-through*, couramment utilisé en environnement *cloud*, l'hyperviseur perd la visibilité sur le GPU, retrouvant le problème de la "boîte noire" au niveau du composant.

Dès lors, la question centrale qui guide ces travaux de recherche peut se formuler ainsi : *Comment estimer de manière précise et non-intrusive la consommation électrique des serveurs et des accélérateurs dans un centre de données de production hétérogène, sans outre-passer l'isolation stricte entre l'opérateur et le client ?*

Pour répondre à cette problématique, cette thèse explore l'hypothèse selon laquelle les canaux auxiliaires thermiques, et plus spécifiquement la température des composants accessible via les interfaces de gestion matérielle (*Intelligent Platform Management Interface* (IPMI)), peuvent constituer une variable de substitution fiable pour reconstituer la consommation électrique.

Cette problématique générale se décline en trois sous-questions de recherche qui structureront ce manuscrit :

— Dans quelle mesure la température *Central Processing Unit* (CPU) peut-elle être utilisée pour modéliser la puissance d'un serveur *bare metal*, et comment isoler le signal utile des perturbations introduites par les systèmes de refroidissement liquide

modernes ?

— Comment dépasser la nécessité d'un calibrage individuel pour construire des modèles portables, capables d'estimer la consommation de serveurs inconnus ou non instrumentés au sein d'un parc hétérogène ?

— Comment adapter ces approches thermiques au cas spécifique des GPU en mode *pass-through* ?

1.3 Contexte industriel

Les travaux présentés dans ce manuscrit s'inscrivent dans le cadre du défi FrugalCloud [17], une initiative de recherche collaborative entre Inria et OVHcloud soutenue par le plan national France Relance. Ce programme, articulé autour de sept projets de recherche et de cinq équipes-projets Inria (Avalon, Inocs, Myriads, Spirals et Stack), explore de nouvelles solutions pour la conception de services *cloud* plus économes en consommation électrique et avec des impacts environnementaux réduits. Le défi se décline en trois axes complémentaires : l'éco-conception de services numériques ; leviers d'efficience énergétique ; et enfin l'optimisation des ressources pour une utilisation frugale du numérique — axe dans lequel s'insèrent nos travaux.

Le contexte industriel offert par OVHcloud présente des spécificités qui ont façonné nos problématiques de recherche :

Une infrastructure massive et hétérogène. Avec un parc de plus de 450 000 serveurs en production, dont la majorité dédiée à son offre de serveurs *bare metal*, OVHcloud est confronté à une grande diversité matérielle. Ce parc est le résultat de deux décennies de développement technologique. De plus, le modèle d'intégration verticale d'OVHcloud, qui conçoit et assemble ses propres serveurs, introduit des configurations matérielles spécifiques absentes des catalogues standards, rendant les modèles pré-établis de la littérature souvent inopérants.

Un refroidissement liquide industriel. La généralisation du refroidissement liquide sous la forme *direct-to-chip water-cooling* à une telle échelle est une signature technologique d'OVHcloud [18]. Cette technologie est efficace énergétiquement mais est encore peu documentée dans la littérature scientifique par rapport au refroidissement par air. La compréhension de la thermodynamique propre à ce système propriétaire a constitué un prérequis indispensable pour développer des modèles thermiques robustes.

L'objectif final pour le partenaire industriel est de bâtir les fondations d'un véritable système de métrologie à grande échelle, capable d'alimenter trois piliers stratégiques : la mesure précise de l'efficacité énergétique, la comptabilité carbone fine et l'optimisation capacitaire des infrastructures électriques.

1.4 Contributions

Cette thèse explore l'hypothèse suivante : les variations de température des composants informatiques permettent d'estimer leur consommation électrique. L'exploration s'articule autour de trois contributions :

Une méthodologie de modélisation de puissance non-intrusive. Notre première contribution évalue la faisabilité de l'estimation de puissance via la température dans un environnement de production. Notre méthodologie, PowerHeat, modélise la consommation d'un serveur *bare metal* en se basant sur la température de son processeur. Cette approche traite spécifiquement du refroidissement aux centres de données modernes : l'influence du refroidissement liquide. En isolant les fluctuations thermiques dues à la charge de calcul de celles générées par le circuit de refroidissement, nous montrons qu'il est possible d'atteindre une précision d'estimation élevée, sans jamais accéder aux métriques internes du système d'exploitation du client.

Une stratégie de généralisation de modèles de puissance. La méthode PowerHeat confirme le lien physique entre température et puissance, mais elle présente une limite opérationnelle : elle nécessite une phase d'apprentissage individuel pour chaque serveur, ce qui est inapplicable à l'échelle d'un parc de plusieurs milliers de machines non instrumentées. Notre deuxième contribution explore la portabilité des modèles avec la méthodologie PPEM-BM. Nous proposons une approche innovante qui ne cherche plus à entraîner un modèle par serveur, mais à sélectionner intelligemment le modèle le plus adapté au sein d'une bibliothèque de modèles existants. En utilisant des techniques d'apprentissage du classement, cette méthode permet d'estimer la consommation de serveurs "aveugles" (jamais instrumentés) en transférant les connaissances acquises sur un sous-ensemble de serveurs de référence.

L'extension aux accélérateurs graphiques en mode *pass-through*. L'essor de l'intelligence artificielle a déplacé le centre de gravité de la consommation énergétique vers les GPU. Dans les environnements *cloud*, ces composants sont souvent mis à disposition via le mode pass-through, qui occulte totalement leur visibilité logicielle pour l'hébergeur. Notre troisième contribution adapte l'approche thermique à ces architectures spécifiques. Nous développons des modèles capables d'inférer la puissance ainsi que le taux d'utilisation réel de GPU Nvidia H100 et A100, en contournant les interférences thermiques propres aux châssis haute densité. Appliquée à un cluster de production, cette méthode permet de révéler une sous-utilisation massive des ressources allouées, ouvrant la voie à de nouvelles stratégies de consolidation énergétique.

1.5 Publications

Les articles suivants ont été publiés dans le cadre de conférences internationales :

1. **Maxime Agusti**, Eddy Caron, Benjamin Fichel, Laurent Lefèvre, Olivier Nicol, Anne-Cécile Orgerie, "*PowerHeat : A non-intrusive approach for estimating the power consumption of bare metal water-cooled servers*", 2024 IEEE Green Computing & Communications (GreenCom).

2. **Maxime Agusti**, Eddy Caron, Benjamin Fichel, Laurent Lefèvre, Olivier Nicol, Anne-Cécile Orgerie, "*PPEM-BM : Portable Power Estimation Methodology for Bare Metal Servers*", 2025 IEEE 31th International Conference on Parallel and Distributed Systems (ICPADS).

3. Pierre Jacquet, **Maxime Agusti**, Eddy Caron, Camille Coti, Marcos Dias De Assunção, Laurent Lefèvre, Anne-Cécile Orgerie, "*Untangling GPU Power Consumption : Job-Level Inference in Cloud Shared Settings*", EUROSYS 2026-European Conference on Computer Systems.

1.6 Organisation du manuscrit

Après cette introduction, le présent manuscrit de thèse est organisé de la manière suivante :

Le Chapitre 2 dresse un panorama des enjeux de la métrologie énergétique dans les centres de données modernes, en analysant les limites des approches actuelles de mesures physiques et estimations logicielles. Ce chapitre définit spécifiquement les contraintes imposées par le modèle *bare metal* et le principe de non-intrusivité qui en découle. Il explore enfin les opportunités offertes par les canaux auxiliaires thermiques et les défis physiques posés par les technologies de refroidissement liquide.

Le Chapitre 3 présente PowerHeat, une méthodologie permettant d'estimer la consommation d'un serveur individuel à partir de la température de son processeur. Ce chapitre explore l'hypothèse selon laquelle la température est une variable de substitution fiable de la puissance et introduit des mécanismes de compensation du bruit thermique induit par les boucles de refroidissement liquide.

Le Chapitre 4 s'attaque au verrou du passage à l'échelle. Partant du constat que l'entraînement individuel des modèles est irréalisable sur un parc massif, ce chapitre introduit l'approche PPEM-BM. Il décrit comment nous avons construit une bibliothèque de modèles de référence et utilisé des algorithmes d'apprentissage du classement pour transférer la capacité d'estimation vers des serveurs dépourvus de capteurs de puissance.

Le Chapitre 5 étend notre approche aux accélérateurs graphiques, composants devenus critiques avec l'essor de l'intelligence artificielle. Ce chapitre traite spécifiquement du mode *pass-through*, qui rend le GPU totalement opaque pour l'hébergeur. Il détaille l'adaptation des modèles thermiques pour ces architectures, tant en refroidissement par air que liquide, et présente une analyse de l'usage d'un cluster de production.

Le Chapitre 6 synthétise les apports de cette thèse et propose des perspectives de recherche pour poursuivre l'amélioration de l'observabilité énergétique des infrastructures informatiques.

Chapitre 2

État de l'art

Sommaire

2.1 La mesure de consommation énergétique dans les centres de données . **10**
2.1.1 Enjeux stratégiques et normatifs 10
2.1.2 Optimisation de l'infrastructure et gestion des capacités . . . 10
2.1.3 Maîtrise des coûts et efficacité opérationnelle 11
2.1.4 Méthodes de mesure et d'estimation 12
2.1.5 Limites et opportunités 13
2.2 Le défi de l'estimation en environnement *bare metal* **14**
2.2.1 Le modèle de responsabilité partagée du *bare metal* 14
2.2.2 La contrainte de non-intrusivité 14
2.2.3 Estimation de puissance : l'apport des canaux auxiliaires . . . 15
2.2.4 Limites et opportunités 16
2.3 Modélisation de la puissance par la température **16**
2.3.1 Approches classiques en refroidissement par air 17
2.3.2 Défis physiques : Inertie et courants de fuite 17
2.3.3 La rupture technologique du refroidissement liquide 18
2.3.4 Limites et opportunités 18
2.4 Généralisation des modèles de puissance **19**
2.4.1 Limites et opportunités 21
2.5 Modélisation de puissance des GPU **21**
2.5.1 Mécanismes de partage et de virtualisation 21
2.5.2 Défis de la mesure et modélisation logicielle 22
2.5.3 Limites et opportunités 23

2.1 La mesure de consommation énergétique dans les centres de données

Le suivi précis de la consommation électrique constitue aujourd'hui un levier stratégique majeur pour les opérateurs de centres de données. Cette section expose un état des lieux de la métrologie énergétique dans les infrastructures modernes et explore les impératifs et les approches techniques actuelles de la mesure physique directe et de l'estimation logicielle.

2.1.1 Enjeux stratégiques et normatifs

En 2025, les pratiques des opérateurs concernant la mesure énergétique restent hétérogènes. D'après les sondages de l'Uptime Institute, 84 % des opérateurs mesurent la consommation électrique globale de leurs équipements informatiques ou du centre de données dans son ensemble [12], mais seuls 19 % déclarent être capables de le faire à l'échelle individuelle du serveur [13]. Or, cette granularité fine est souvent le pré-requis indispensable à l'application efficace des nouvelles réglementations et des stratégies d'optimisation avancées.

Sur le plan réglementaire, la directive européenne 2023/1791 (EED) impose désormais aux propriétaires et exploitants de centres de données d'une puissance installée d'au moins 500 kW de rendre publiques leurs performances énergétiques [19]. La définition de ces indicateurs de performance est encadrée par la norme ISO 30134 [3]. Cette norme définit notamment trois catégories de mesure pour le *Power Usage Effectiveness* (PUE), tel qu'illustré par la Figure 2.1. La catégorie 3, recommandée pour les installations performantes visant un PUE inférieur ou égal à 1,2, exige une mesure précise au niveau de l'unité de distribution électrique des équipements IT, soulignant la nécessité d'une métrologie fine.

2.1.2 Optimisation de l'infrastructure et gestion des capacités

Au-delà de la conformité, la mesure individuelle permet une gestion proactive et une planification optimisée de l'infrastructure électrique. Cette infrastructure (câblage, disjoncteurs, onduleurs, *Power Distribution Unit* (PDU), etc.) représente une ressource coûteuse, dont l'investissement est estimé entre 10 et 20 dollars par watt installé [14].

Historiquement, les opérateurs dimensionnent cette infrastructure en se basant sur la puissance nominale des serveurs, adoptant une stratégie conservatrice qui surestime largement la consommation réelle [1, 14]. Les mesures sur le terrain démontrent que la

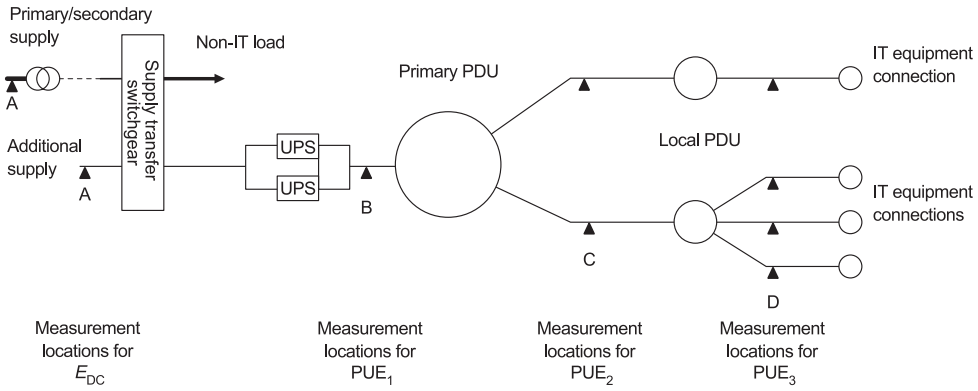


FIGURE 2.1 – Points de mesure pour l'évaluation du PUE selon la norme ISO 30134 [3].

consommation maximale atteinte en production est souvent bien inférieure à ces valeurs théoriques, laissant une marge de puissance inutilisée considérable [14].

L'identification de cette marge par la mesure réelle permet d'augmenter la densité de serveurs hébergés pour un même budget énergétique [20]. Cette pratique de sur-allocation parie sur le fait statistique que tous les serveurs n'atteindront pas leur pic de charge simultanément. Toutefois, cette approche comporte des risques : une surcharge non maîtrisée peut déclencher les disjoncteurs et provoquer des interruptions de service [20, 21]. Une surveillance fine, combinée à des techniques de limitation de puissance dynamique (*power capping*), est alors nécessaire pour maintenir la consommation sous les seuils de sécurité [20].

2.1.3 Maîtrise des coûts et efficacité opérationnelle

La mesure individuelle est également un vecteur de rentabilité économique. L'évolution volatile des coûts énergétiques impacte directement les marges bénéficiaires, en particulier pour les fournisseurs de services *cloud*. Une estimation précise est nécessaire pour garantir que les coûts opérationnels n'érodent pas le retour sur investissement de l'infrastructure.

Dans les environnements multi-locataires, la mesure par serveur ou par machine virtuelle permet de transiter d'une facturation forfaitaire vers une facturation à l'usage réel [22]. Elle permet également des stratégies de tarification dynamique, incitant par exemple les clients à planifier leurs tâches flexibles lors des périodes où l'électricité est moins chère ou moins carbonée [23, 24]. Enfin, la mesure permet d'identifier les machines sous-utilisées ou "zombies" afin de consolider les charges de travail sur un nombre réduit de serveurs physiques et d'éteindre les ressources excédentaires, réduisant ainsi drastiquement la consommation globale [1, 25].

2.1.4 Méthodes de mesure et d'estimation

La caractérisation de la consommation d'énergie des centres de données repose sur deux grandes familles d'approches : les méthodes physiques directes et les estimations logicielles basées sur des modèles.

Mesures physiques directes

La méthode la plus précise consiste à instrumenter le circuit électrique pour mesurer directement le courant et la tension [26–28]. Au niveau du rack ou du serveur, cette mesure est généralement réalisée via des unités de distribution d'alimentation intelligentes (Smart PDU) ou des wattmètres externes intercalés entre la prise murale et l'alimentation. Bien que fournissant une vérité terrain fiable, les PDU offrant une mesure individuelle par prise représentent un surcoût significatif [29].

Alternativement, la plupart des serveurs modernes intègrent des circuits de mesure directement sur la carte mère ou au sein des blocs d'alimentation (*Power Supply Unit* (PSU)) [26, 30]. Ces données sont exposées via le contrôleur de gestion - *Baseboard Management Controller* (BMC) - à travers des protocoles standardisés comme IPMI ou Redfish [26]. Selon la sophistication du firmware et du matériel, le BMC peut rapporter la puissance globale du châssis, voire la consommation spécifique de sous-composants tels que le processeur (CPU), la mémoire ou les ventilateurs.

Interfaces et modélisation logicielles

Pour obtenir une granularité plus fine, nécessaire à l'optimisation logicielle, les constructeurs ont développé des interfaces dédiées. L'interface *Running Average Power Limit* (RAPL) d'Intel et AMD est largement utilisée pour estimer la consommation du CPU et de la mémoire *Dynamic Random Access Memory* (DRAM) [31]. Elle s'appuie sur des registres spécifiques au modèle (MSR) alimentés soit par des mesures physiques internes, soit par des modèles micro-architecturaux prédictifs [32]. De même, pour les accélérateurs graphiques, la bibliothèque *NVIDIA Management Library* (NVML) permet d'interroger la consommation instantanée du GPU [28]. L'accès à ces interfaces (RAPL, NVML) nécessite toutefois généralement des privilèges administrateur (root) sur le système hôte [28, 32].

En l'absence de capteurs physiques dédiés, l'usage de modèles énergétiques s'offre comme une solution de substitution. Cette approche représente une alternative à la fois flexible et peu coûteuse, qui exploite les indicateurs d'activité du système pour en estimer la consommation électrique [15, 27]. L'hypothèse fondamentale est la corrélation entre l'activité logicielle observable et la consommation électrique matérielle.

Les approches de modélisation varient en complexité :

— **Modèles linéaires simples :** Ils supposent une relation proportionnelle entre l'utilisation du CPU (u) et la puissance : $P = P_{idle} + (P_{max} - P_{idle}) \times u$ [14].

— **Modèles non-linéaires :** Pour capturer des comportements complexes comme la variation dynamique de fréquence et de tension (*Dynamic Voltage and Frequency Scaling* (DVFS)) [33], des modèles polynomiaux sont utilisés, la puissance dynamique étant souvent proportionnelle au cube de la fréquence [15].

— **Apprentissage automatique :** Des techniques avancées comme les réseaux de neurones [34], *eXtreme Gradient Boosting* (XGBoost) [35] ou forêts aléatoires [36] exploitent des dizaines de compteurs de performance matérielle (PMC) pour apprendre les relations complexes entre l'activité micro-architecturale et l'énergie, offrant souvent une précision supérieure aux modèles analytiques [15].

La construction de ces modèles nécessite quasi systématiquement une phase de calibrage, durant laquelle le système exécute des charges de travail contrôlées sous la surveillance d'un wattmètre physique, afin de déterminer les coefficients spécifiques au matériel hôte [37].

2.1.5 Limites et opportunités

Malgré la diversité des méthodes existantes, plusieurs verrous limitent la généralisation de la mesure individuelle dans les centres de données de production.

Premièrement, le déploiement massif de PDU intelligents dans un centre de données existant est économiquement et techniquement complexe [23, 27]. Le coût matériel est prohibitif pour des milliers de machines, et le remplacement des PDU nécessite souvent une interruption de service pour modifier le câblage électrique [29]. Cette solution est donc principalement réservée aux nouveaux déploiements.

Deuxièmement, si les capteurs intégrés (BMC/IPMI) semblent être une solution idéale car sans surcoût matériel, ils souffrent de problèmes de précision et d'hétérogénéité [30]. Il existe fréquemment un écart entre la valeur rapportée par le BMC et la mesure réelle à la prise [28]. De plus, la fréquence d'échantillonnage et le type de données disponibles dépendent fortement du constructeur et de la version du firmware, compliquant la standardisation [30]. Si l'adoption du *Power Management Bus* (PMBus) [38] a amélioré la communication entre les PSU et la carte mère sur les serveurs récents, un parc informatique contient souvent une part importante de serveurs anciens non équipés ou incompatibles.

Troisièmement, la modélisation énergétique logicielle, bien qu'efficace pour combler l'absence de capteurs physiques, présente une contrainte majeure d'intrusivité. La collecte des métriques nécessite l'exécution d'un agent logiciel au niveau du système d'exploitation - *Operating System* (OS) - de la machine cible [15, 30]. Cette approche est inopérante dans les environnements de colocation ou *bare metal* où l'opérateur n'a aucun accès à l'OS du client [30]. De plus, les modèles souffrent d'une faible portabilité : un modèle calibré pour une architecture spécifique ne peut être transféré tel quel sur une autre génération de serveur, nécessitant un recalibrage coûteux en temps [34, 39, 40].

Face à ces limites — coût des PDU, hétérogénéité des BMC et intrusivité des mo-

dèles logiciels — il existe une opportunité pour développer des méthodes de supervision "non-intrusives". L'utilisation de données accessibles hors de l'OS, telles que les capteurs thermiques et de puissance globale via IPMI, ouvre la voie à de nouvelles approches de modélisation respectant les contraintes d'exploitation des opérateurs modernes.

2.2 Le défi de l'estimation en environnement *bare metal*

Cette section définit les spécificités du modèle *bare metal*, en s'ancrant notamment dans le contexte opérationnel d'OVHcloud, partenaire industriel de cette thèse.

2.2.1 Le modèle de responsabilité partagée du *bare metal*

Le serveur *bare metal* est un modèle de service d'infrastructure — *Infrastructure-as-a-Service* (IaaS) — où le client loue une machine physique dans son intégralité, sans couche de virtualisation intermédiaire imposée par le fournisseur. Ce modèle, tel que commercialisé par les principaux fournisseurs de *cloud*, repose sur un paradigme *"single-tenant"* : la machine est dédiée à un locataire unique, garantissant une isolation physique stricte par rapport aux autres utilisateurs, contrairement aux environnements virtualisés classiques où la cohabitation est la norme. Cette architecture définit une frontière stricte entre les responsabilités du fournisseur et celles du client :

— **Le client (Locataire) :** Il bénéficie d'un accès *root* ou administrateur complet. Il dispose d'une liberté totale sur la pile logicielle : choix du système d'exploitation, installation d'hyperviseurs pour créer un *cloud* privé, ou modifications du noyau pour optimiser ses applications. En contrepartie, il assume l'entière responsabilité de la gestion, de la mise à jour et de la sécurisation de son environnement logiciel.

— **Le fournisseur (Hébergeur) :** Propriétaire de la machine, il en assure l'hébergement dans son centre de données. Sa responsabilité se limite strictement à l'infrastructure physique : mise à disposition du serveur, maintenance des composants (CPU, mémoire, disques, etc.), et garantie des services utilitaires (alimentation électrique, refroidissement, connectivité réseau).

2.2.2 La contrainte de non-intrusivité

La séparation des responsabilités inhérente au modèle bare metal impose au fournisseur une contrainte fondamentale : la **non-intrusivité**. Pour des raisons contractuelles, de sécurité et de confidentialité des données, le fournisseur s'engage à ne jamais accéder à l'environnement logiciel du client. Par conséquent, toute méthode de monitoring nécessitant l'installation d'agents logiciels sur l'OS du locataire (introspecteurs de processus,

lecture de compteurs de performance, etc.) est proscrite. L'observation de l'état du serveur par le fournisseur doit donc s'opérer exclusivement par des canaux externes, dits *"out-of-band"*. Le périmètre de supervision accessible se limite aux éléments d'infrastructure sous le contrôle direct de l'hébergeur. Ce périmètre couvre quatre domaines distincts :

1. **Matériel :** Via des interfaces de gestion standardisées comme IPMI, le fournisseur collecte les événements matériels pour surveiller la santé des composants et anticiper les pannes (défaillance disque, erreurs mémoire).
2. **Thermique :** Pour garantir les conditions opérationnelles, le fournisseur observe en continu les sondes de température situées aux points critiques (processeurs, carte mère) afin de prévenir les dégradations liées à la surchauffe.
3. **Électrique :** La stabilité de l'alimentation est surveillée via les onduleurs (*Uninterruptible Power Supply* (UPS)) ou les PDU.
4. **Réseau :** Le monitoring s'effectue au niveau des ports des commutateurs réseaux de l'infrastructure. L'observation du trafic entrant/sortant permet d'assurer la qualité de service et d'activer des mécanismes de protection périphériques, notamment contre les attaques par déni de service (DDoS).

2.2.3 Estimation de puissance : l'apport des canaux auxiliaires

La littérature académique sur l'estimation de puissance en "boîte noire" est rare. Néanmoins, le domaine de la sécurité informatique a exploré ces problématiques sous l'angle offensif des canaux auxiliaires (*Side-Channel Attacks*). Les attaquants, confrontés à des systèmes isolés ou sécurisés, cherchent à exfiltrer des informations ou à déduire l'état d'une machine en observant les effets secondaires physiques de son activité computationnelle. Plusieurs vecteurs d'attaque exploitent les corrélations entre l'activité numérique et les grandeurs physiques observables de l'extérieur :

Islam *et al.* [41] propose l'utilisation de microphones pour écouter le bruit ambiant d'un centre de données. Le bruit généré par les ventilateurs des serveurs est directement corrélé à leur charge thermique, et donc à leur consommation électrique. En filtrant ce signal, il est possible de déduire le niveau de consommation des serveurs voisins. Ces informations permettent de synchroniser des charges de travail pour orchestrer des *Power Attacks*, visant à surcharger l'infrastructure électrique.

Islam *et al.* [42] montre que la consommation électrique des serveurs induit des fluctuations de tension sur les lignes d'alimentation, causées spécifiquement par la commutation haute fréquence des circuits de correction du facteur de puissance intégrés aux blocs d'alimentation. Un attaquant branché sur le même réseau électrique peut analyser ces ondulations de tension pour inférer la consommation globale des locataires légitimes. Cela permet de lancer une attaque de puissance précisément au moment où la consommation est élevée, maximisant les chances de provoquer une panne.

Islam *et al.* [21] montre qu'un attaquant, en analysant la température de l'air entrant dans ses propres serveurs, peut estimer la charge de travail de ses voisins. Dans un centre de données, la chaleur dégagée par un serveur victime peut recirculer ou influencer les capteurs environnants. L'utilisation d'un filtre de Kalman à état augmenté permet d'analyser les variations de température à l'entrée de ses serveurs et d'en déduire la consommation électrique globale des autres locataires, malgré le bruit des données.

2.2.4 Limites et opportunités

Bien que les travaux dans le domaine des attaques par canaux auxiliaires visent des objectifs offensifs, les méthodes employées valident techniquement la possibilité d'estimer l'activité et la puissance d'un serveur via des observations indirectes. La contrainte de non-intrusivité du bare metal nous prive des compteurs de performance logiciels, mais nous laisse accès aux sondes physiques via le BMC. En particulier, la relation forte entre la puissance consommée et la dissipation thermique, exploitée par les attaques par canaux auxiliaires, suggère une opportunité.

La prise en compte des capteurs de température des serveurs, accessibles via IPMI, s'avère être un indicateur candidat privilégié pour construire un modèle d'estimation de puissance non-intrusif, respectant les contraintes strictes de l'environnement bare metal.

2.3 Modélisation de la puissance par la température

La modélisation de la consommation énergétique des serveurs par le biais de la thermométrie repose sur des principes fondamentaux de la thermodynamique. En régime permanent, la première loi de la thermodynamique définit que l'énergie entrant dans un système isolé est équivalente à l'énergie qui en sort. Dans le contexte des équipements informatiques, l'énergie électrique consommée est presque intégralement dissipée sous forme de chaleur par effet Joule [1].

La consommation totale d'un système peut être décomposée en deux composantes principales [43] :

— **La puissance statique** est consommée dès la mise sous tension du système, indépendamment de toute activité de calcul. Elle est principalement causée par les courants de fuite au niveau des transistors.

— **La puissance dynamique** résulte directement de l'activité des circuits, c'est-à-dire de la commutation des transistors nécessaire pour effectuer les calculs.

En conséquence, mesurer la chaleur dissipée par un serveur offre, en théorie, une méthode pour estimer sa consommation électrique. Tout le défi scientifique réside cependant

1. Une infime partie de l'énergie électrique est convertie en rayonnement électromagnétique ou acoustique.

dans la mesure précise de cette chaleur dissipée.

2.3.1 Approches classiques en refroidissement par air

La chaleur générée par l'effet Joule réchauffe les composants du serveur ainsi que son environnement immédiat, typiquement l'air circulant dans le châssis. Les serveurs modernes sont équipés de multiples capteurs de température situés sur les cartes mères et les composants les plus énergivores (CPU, mémoire, stockage) pour surveiller cet échauffement.

L'approche de Rezaei *et al.* [44] exploite les capteurs de température situés à l'entrée (T_{in}) et à la sortie (T_{out}) du flux d'air du serveur. La différence de température (ΔT) entre ces deux points permet de quantifier l'apport calorifique du serveur à l'air ambiant. Le modèle thermique reliant la puissance du serveur (P_{srv}) à ce différentiel de température est formulé par une relation linéaire de la forme :

$$P_{srv} \approx \alpha \cdot P_{idle}(T_{out} - T_{in}) + P_{idle} \tag{2.1}$$

Où P_{idle} représente la consommation au repos et α est un coefficient de calibration spécifique au matériel. Ces deux paramètres doivent être mesurés et calculés préalablement au déploiement opérationnel du serveur.

Pour affiner la précision de ces modèles, la température interne des composants, et notamment celle du processeur, est utilisée comme un "proxy" pour estimer la puissance.

Un proxy [2] est défini comme une variable de substitution qui remplace une variable d'intérêt lorsque celle-ci est non observable ou difficilement mesurable. Pour être considérée comme un proxy valide, cette variable doit présenter une forte corrélation — pas nécessairement linéaire — avec la variable utile.

Dans ce contexte, bien que la température soit un indicateur précieux, sa relation avec la consommation électrique est complexe. L'analyse expérimentale menée par Jin *et al.* [45] souligne que cette relation n'est pas toujours strictement linéaire. Elle peut être plus fidèlement modélisée par des fonctions linéaires par morceaux. Par exemple, selon le mode de gestion énergétique activé (ex : mode "Performance"), la consommation peut être proportionnelle au ΔT jusqu'à un certain seuil (ex : 80°C), au-delà duquel le comportement thermique diverge.

2.3.2 Défis physiques : Inertie et courants de fuite

La modélisation thermique se heurte à deux contraintes physiques majeures : l'inertie thermique et la dépendance des courants de fuite à la température.

2. https://fr.wikipedia.org/wiki/Proxy_(variable)

Premièrement, contrairement à la puissance électrique qui varie quasi instantanément avec la charge de calcul, la température est un indicateur "tardif" [46]. L'inertie thermique des matériaux introduit un décalage temporel significatif entre un pic d'activité du processeur et l'augmentation de température détectable par les sondes.

Deuxièmement, une part significative de la consommation, la puissance de fuite, dépend de manière exponentielle de la température [47]. Concrètement, cela implique une boucle de rétroaction positive : plus l'environnement du serveur est chaud, plus la température de jonction interne augmente, ce qui accroît les courants de fuite et donc la consommation globale. L'étude de Jin *et al.* [45] a montré qu'une augmentation de la température d'entrée d'air de 20°C à 35°C entraînait une hausse de la consommation maximale comprise entre 3,16 % et 5,18 % selon le mode de fonctionnement du serveur.

2.3.3 La rupture technologique du refroidissement liquide

Pour répondre aux impératifs d'efficacité énergétique, les opérateurs de centres de données *hyperscale* modernes, tels qu'OVHcloud, adoptent massivement des systèmes de refroidissement liquide de type *Direct-To-Chip* (DTC) [2].

Ce paradigme utilise des plaques froides (*cold plates*, appelées aussi *water block*) traversées par un liquide caloporteur, généralement de l'eau, fixées directement sur les composants à haute densité thermique comme les CPU et les GPU [48]. Les autres composants (mémoire, stockage, carte-mère) restent généralement refroidis par un flux d'air.

L'eau possédant une capacité calorifique volumique environ 3300 fois supérieure à celle de l'air, le DTC dissipe la chaleur avec une efficacité nettement meilleure [49]. L'adoption de cette technologie modifie radicalement la thermodynamique du serveur. L'étude comparative de Latif *et al.* [50] sur des nœuds de calcul équipés de GPU NVIDIA H100 illustre cet impact : sous pleine charge, les GPU refroidis par liquide maintiennent une température de jonction stable entre 41°C et 50°C, tandis que ceux refroidis par air fluctuent fortement entre 54°C et 72°C.

2.3.4 Limites et opportunités

Bien que la littérature ait expérimenté l'utilisation des capteurs thermiques pour l'estimation de puissance, les approches existantes se concentrent sur des infrastructures refroidies par air. Le passage au refroidissement liquide introduit de nouvelles variables et contraintes :

— La température du composant ne dépend plus uniquement de l'air ambiant, mais également de la température et du débit du liquide traversant la plaque froide.

— Les capteurs standards intégrés aux serveurs ne permettent généralement pas de mesurer ni la température du liquide à l'entrée de la plaque, ni son débit.

— La stabilité thermique accrue offerte par le *water cooling* resserre l'amplitude des variations de température, réduisant la quantité d'information disponible pour entraîner des modèles prédictifs.

À notre connaissance, aucune méthodologie de modélisation de puissance par la température ciblant spécifiquement des serveurs *bare metal* refroidis à l'eau n'a été formalisée dans l'état de l'art. Il existe donc une opportunité de recherche, particulièrement pertinente dans le contexte industriel d'OVHcloud qui opère la quasi-totalité de ses 450 000 serveurs avec cette technologie.

Le véritable défi, pour l'application à grande échelle de ces modèles réside dans une double hétérogénéité. D'une part, l'hétérogénéité matérielle d'un parc informatique aux configurations variées. D'autre part, l'hétérogénéité environnementale où chaque serveur évolue dans des conditions de température qui lui sont propres, dépendant de sa charge, de sa position physique dans le centre de données et des serveurs qui l'entourent.

2.4 Généralisation des modèles de puissance

Le passage à l'échelle de la modélisation de la consommation énergétique des serveurs dans les centres de données modernes soulève une problématique plus complexe : celle de la généralisation. Elle désigne la capacité d'une approche d'estimation à fournir des prédictions précises lorsqu'elle est appliquée à des données, des charges de travail ou des configurations matérielles qui diffèrent de celles observées durant sa phase d'entraînement [51]. Traditionnellement, les approches d'apprentissage automatique reposent sur l'hypothèse que les données d'entraînement (source) et de test (cible) sont indépendantes et identiquement distribuées. Cependant, dans les environnements réels tels que le *cloud*, cette hypothèse est fréquemment invalidée par la diversité des infrastructures et des environnements, entraînant une dégradation significative de la performance des modèles sur les nouvelles données [16].

Nous distinguons plusieurs dimensions de la généralisation dans les systèmes informatiques, chacune présentant des défis spécifiques :

— **La généralisation inter-serveurs :** Elle correspond à la capacité d'un modèle entraîné sur un serveur "source" à prédire fidèlement la consommation d'un serveur "cible" présentant une configuration matérielle différente. Cette hétérogénéité constitue un obstacle majeur à la portabilité [52]. De surcroît, les variations inhérentes à la fabrication, même entre des composants matériels théoriquement identiques, peuvent introduire des erreurs fondamentales qui limitent la transférabilité directe des modèles [53].

— **La généralisation inter-charges de travail :** La nature des tâches exécutées influence considérablement le profil énergétique. Un modèle entraîné exclusivement sur des charges de travail intensives en CPU peut ainsi échouer à généraliser sur des

tâches sollicitant majoritairement la mémoire [54].

— **La généralisation temporelle :** Il s'agit de la capacité d'un modèle à maintenir sa précision dans la durée, en dépit du vieillissement du matériel ou de l'évolution technologique. Les études montrent que les prédictions prospectives fondées sur des données historiques perdent rapidement en précision au-delà d'un certain horizon temporel [51].

Face à ces défis, plusieurs approches ont été proposées dans la littérature pour améliorer la robustesse et la portabilité des estimations.

Dans le contexte de serveurs hétérogènes, Mo *et al.* [16] aborde le problème où les données ne respectent pas l'hypothèse de distribution identique. Cette méthode utilise la régression par adaptation de domaine pour aligner les distributions de probabilité conjointes des données de consommation entre un serveur source, riche en données, et un serveur cible pauvre en données étiquetées. L'avantage majeur de cette technique réside dans sa capacité à opérer sans nécessiter un historique conséquent sur la cible.

Pour traiter l'hétérogénéité des charges de travail, Mo *et al.* [54] propose une solution au problème de décalage de domaine, typique lorsqu'un modèle CPU-intensif est appliqué à une charge I/O-intensive. En exploitant l'apprentissage par transfert multi-sources, l'approche transfère les connaissances acquises sur plusieurs charges de travail sources vers une charge cible. Cela permet de généraliser le modèle à de nouveaux profils applicatifs sans recourir à une coûteuse campagne de collecte de données pour chaque nouvelle tâche.

D'autres travaux se concentrent sur l'évolution du matériel. Plutôt que de réentraîner un modèle de zéro lors d'une mise à jour matérielle ou d'un changement d'échelle, ModelMap [55] cherche à modéliser la transformation entre l'ancien et le nouveau système. L'approche construit une fonction de mappage qui relie mathématiquement le modèle hérité (source) au nouveau modèle (cible), assurant ainsi une continuité dans l'estimation.

Enfin, face à la diversité croissante des architectures, Hydra [56] part du constat qu'aucun modèle unique — qu'il soit linéaire ou basé sur des réseaux de neurones — n'est performant dans toutes les conditions. Si les modèles d'apprentissage profond offrent une grande précision au prix d'un coût de calcul élevé, les modèles linéaires restent rapides mais parfois imprécis. Hydra introduit un sélecteur de modèle léger, basé sur une forêt aléatoire, qui surveille les statistiques du système en temps réel. Ce sélecteur classifie l'état du serveur et active dynamiquement le modèle de puissance le plus approprié, optimisant ainsi le compromis entre précision et coût d'inférence. Au lieu de chercher un modèle universel qui se généraliserait mal, cette approche généralise le système de prédiction lui-même par une adaptation dynamique.

2.4.1 Limites et opportunités

L'analyse de l'état de l'art révèle des avancées significatives dans les techniques de transfert et d'adaptation de domaine. Cependant, aucune des méthodes identifiées ne mentionne explicitement une validation ou une conception spécifique pour des environnements refroidis à l'eau. Bien que techniquement déployables sur de tels serveurs, ces approches nécessiteraient vraisemblablement un ré-entraînement complet pour intégrer les dynamiques thermiques propres à ce mode de refroidissement.

Néanmoins, l'architecture proposée par Hydra, combinant plusieurs modèles et un mécanisme de sélection, présente des perspectives particulièrement intéressantes pour les environnements *bare metal* hétérogènes. Le principe d'un sélecteur capable de classifier l'état du serveur pour mapper la situation au modèle le plus performant permet de répondre efficacement à l'hétérogénéité conjointe des tâches et des matériels.

Il existe dès lors une opportunité pour proposer une nouvelle méthodologie de généralisation. Celle-ci pourrait s'inspirer de l'approche multi-modèles, tout en étant spécifiquement adaptée aux contraintes et aux variables disponibles dans les environnements de production refroidis à l'eau, comblant ainsi le vide identifié dans la littérature actuelle.

2.5 Modélisation de puissance des GPU

L'arrivée des grands modèles de langage - *Large Language Model* (LLM) - a provoqué une explosion de la demande en unités de traitement graphique (GPU). L'entraînement et l'inférence de modèles comme GPT-4 ou BLOOM nécessitent des infrastructures à très grande échelle [50, 57–59], rendant la modélisation et la maîtrise de leur consommation énergétique plus importantes que jamais.

2.5.1 Mécanismes de partage et de virtualisation

Pour la plupart des entreprises, l'accès aux ressources GPU s'opère majoritairement via des fournisseurs de services IaaS telles qu'Amazon Web Services (AWS), Google Cloud Platform (GCP), Microsoft Azure et OVHcloud. Ces opérateurs permettent aux utilisateurs de louer des instances de calcul où les GPU sont connectés à des *Virtual Machine* (VM), fournissant ainsi la puissance nécessaire au sein d'environnements virtualisés flexibles [60, 61].

Le mécanisme de connexion aux ressources de calcul s'effectue généralement via un bus PCI-Express, canal par lequel transitent les instructions et les données entre la VM et l'accélérateur [62]. Cependant, pour optimiser l'utilisation de ces accélérateurs coûteux, plusieurs technologies ont été développées pour permettre leur partage ou leur division entre plusieurs charges de travail. On distingue principalement deux modes de gestion multi-utilisateurs :

— **Le partage temporel :** Ce mode alloue le moteur de calcul du GPU aux différentes machines virtuelles de manière séquentielle, à tour de rôle. Bien qu'efficace pour des charges légères, il peut introduire des latences dues à la commutation de contexte [63].

— **Le partitionnement matériel :** La technologie *Multi-Instance GPU* (MIG) permet de partitionner physiquement un unique GPU en plusieurs instances isolées. Chaque instance dispose de sa propre bande passante mémoire et de ses propres unités de calcul, garantissant une isolation stricte des performances et permettant à plusieurs locataires de partager le matériel de manière sécurisée [63].

2.5.2 Défis de la mesure et modélisation logicielle

Du point de vue de l'opérateur *cloud*, la virtualisation et le partage des ressources GPU soulèvent une problématique complexe liée à la métrologie de l'énergie. Si la technologie MIG assure une isolation efficace de la mémoire et des performances, le matériel ne fournit pas, à ce jour, de mesure de la consommation d'énergie au niveau de la partition. Les registres matériels ne rapportent que la consommation globale du GPU, rendant opaque la contribution individuelle de chaque locataire [63].

Cette limitation impose le recours à des modèles de puissance logiciels pour estimer la consommation spécifique de chaque instance. Cette estimation est indispensable pour déterminer la "juste part" énergétique imputable à chaque utilisateur et établir des modèles de facturation équitables. Plusieurs approches ont été proposées dans la littérature récente pour répondre à ce besoin :

— **Approches basées sur l'apprentissage automatique :** Vamja *et al.* [63] proposent des méthodes logicielles exploitant des métriques de profilage légères (via DCGM) couplées à des algorithmes d'apprentissage automatique (XGBoost). Cette approche permet de désagréger la consommation globale pour estimer celle de chaque partition MIG individuelle avec une précision accrue.

— **Sondes logicielles et monitoring :** Jay *et al.* [28] ont mené une étude comparative d'outils de mesure logicielle. Ces outils, capables d'estimer la consommation au niveau du processus ou du conteneur, sont évalués selon leur précision et le surcoût qu'ils induisent sur le système hôte, guidant ainsi les opérateurs dans le choix de la solution de monitoring la plus adaptée.

— **Modélisation par requête :** Dans une optique applicative, Luccioni *et al.* [64] se concentrent sur la mesure de l'énergie par requête pour divers modèles d'IA. Ces données sont essentielles pour passer d'une tarification basée sur le temps d'occupation à une tarification basée sur la consommation réelle et l'empreinte carbone.

— **Simulation micro-architecturale :** GPUWattch [65] intègre des modèles de puissance directement dans le simulateur de performance GPGPU-Sim [66]. Ils décom-

posent le GPU en composants micro-architecturaux (registres, caches, unités d'exécution) pour calculer la puissance dynamique et statique en fonction de l'activité simulée cycle par cycle.

2.5.3 Limites et opportunités

Bien que les travaux existants offrent des solutions robustes pour le monitoring granulaire, ils reposent majoritairement sur l'accès aux compteurs de performance ou à l'interface de gestion logicielle NVML depuis l'hôte. Cependant, dès lors que la connexion PCIe est dédiée exclusivement à une machine virtuelle, l'hyperviseur (et par extension l'opérateur *cloud*) perd la capacité de communiquer directement avec le GPU via les pilotes standards. L'opérateur se trouve alors dépossédé de sa capacité à interroger les sondes énergétiques internes. Le GPU devient, du point de vue de l'hôte, une "boîte noire" dont le fonctionnement interne est inaccessible, et dont on ne peut observer que les effets externes : la consommation électrique globale mesurée à la prise ou la dissipation thermique.

Cette situation, qui présente des similarités avec la gestion de serveurs *bare metal* mais à l'échelle du composant, crée un angle mort dans la supervision énergétique des centres de données. Il existe donc un besoin pour les opérateurs de développer des méthodes non intrusives capables de "restaurer" cet indicateur de consommation sans transgresser l'isolation de la VM cliente.

Chapitre 3

Modélisation non-intrusive de la puissance d'un serveur *bare metal* par la température du processeur

Sommaire

3.1 Introduction **25**
3.1.1 Mesure en environnement *bare metal* : un défi complexe . . . 26
3.1.2 La température CPU comme proxy de la puissance 27
3.1.3 Structure du chapitre 28
3.2 Méthodologie : PowerHeat **28**
3.2.1 Le banc d'expérimentation 28
3.2.2 Collecte des données 30
3.2.3 Approches de modélisation comparées 30
3.2.4 Algorithme et validation 31
3.3 Résultats et Analyse **32**
3.3.1 Observations des mesures thermiques et électriques des serveurs 33
3.3.2 Performance des modèles 35
3.3.3 Analyse de cas spécifiques 37
3.3.4 Discussion 40
3.4 Conclusion **41**

3.1 Introduction

Le suivi précis de la consommation électrique au niveau du serveur individuel est un levier stratégique majeur pour les opérateurs de centres de données. Cette granularité de

mesure est le fondement sur lequel reposent trois piliers de la gestion moderne : la conformité aux réglementations environnementales, la mise en œuvre de stratégies d'optimisation de l'efficacité énergétique, et la gestion prévisionnelle des infrastructures techniques électriques et de refroidissement (c.f. Section 2.1, page 10).

Pourtant, malgré cet enjeu stratégique, ce suivi individuel est loin d'être généralisé. Une enquête menée en 2023 par l'Uptime Institute sur l'efficacité énergétique a révélé que seuls 48 % des opérateurs suivent activement la demande énergétique moyenne de leurs équipements informatiques, soulignant un décalage important entre le besoin et la réalité du terrain [13].

3.1.1 Mesure en environnement *bare metal* : un défi complexe

Le principal obstacle à une instrumentation énergétique complète est d'abord d'ordre économique. Le déploiement de wattmètres ou PDU intelligents sur chaque serveur représente un surcoût matériel et d'installation non négligeable (environs 80 $ par serveur) [67], particulièrement rédhibitoire à large échelle. À ces freins financiers s'ajoutent des contraintes opérationnelles : dans un centre de données existant et en production, l'installation de nouveaux équipements de mesure de puissance implique une interruption de service, car il faut physiquement débrancher puis rebrancher chaque serveur. Dans le cas spécifique de la location de serveurs *bare metal*, une telle interruption est souvent impossible pour des raisons contractuelles, où la continuité de service est un engagement de l'opérateur (c.f. Section 2.2.1, page 14).

Conscients de ces contraintes, les fabricants intègrent désormais dans les serveurs récents des capteurs de consommation électrique, soit au niveau de la carte mère, soit directement dans les blocs d'alimentation (c.f. Section 2.1.4, page 12). Ces mesures sont alors accessibles à distance et de manière non-intrusive via l'interface du BMC. Cependant, cette solution ne couvre pas le vaste parc de serveurs historiques encore en opération. Pour le reste de l'infrastructure existante, il faudra attendre la fin des contrats de location et le renouvellement progressif du matériel pour bénéficier de cette fonctionnalité, un processus qui peut prendre plusieurs années.

Face à ces problématiques d'instrumentation matérielle, une autre voie a été largement explorée par la recherche : l'estimation de la consommation d'énergie via des modèles logiciels (c.f. Section 2.1.4, page 12 et Section 2.3, page 16). Ces modèles, souvent appelés wattmètres logiciels, établissent une corrélation mathématique entre la consommation électrique réelle d'un serveur et des indicateurs de performance système tels que le taux d'utilisation CPU, le nombre d'opérations disque, ou encore le trafic réseau. Cependant, ces approches présentent un inconvénient majeur dans notre contexte : la collecte de ces indicateurs nécessite l'installation d'un agent logiciel, et donc un accès intrusif au système d'exploitation de chaque serveur. Cette contrainte rend ces approches particulièrement inadaptées aux environnements de location *bare metal*, où l'opérateur n'a, par contrat, aucun droit d'accès aux systèmes d'exploitation des serveurs de ses clients.

L'enjeu est donc de développer une méthode d'estimation de la puissance qui soit à la fois précise, et surtout, non-intrusive, afin de s'adapter aux contraintes du parc existant et du modèle *bare metal*.

3.1.2 La température CPU comme proxy de la puissance

Le point de départ de notre approche s'appuie sur une pratique courante au sein des opérateurs de centres de données : la supervision des serveurs par le biais des capteurs matériels via l'interface du BMC (IPMI ou RedFish). Parmi ces capteurs, la température CPU est une métrique généralement disponible. Les opérateurs, notamment en environnement *bare metal*, la collectent pour la supervision du refroidissement et de prévention des surchauffes.

Cette métrique est particulièrement intéressante car la température d'un CPU est intrinsèquement liée à sa consommation d'énergie. En effet, plus un processeur travaille, plus il consomme de l'électricité. Par effet Joule, cette énergie électrique est quasi-intégralement dissipée sous forme de chaleur, ce qui fait mécaniquement augmenter sa température. Partant de ce principe, et sachant que plusieurs travaux de recherche ont déjà montré une forte corrélation entre puissance CPU, activité CPU et la consommation électrique globale du serveur [14, 28], notre hypothèse est qu'en observant finement la température CPU, il est possible d'estimer la consommation électrique de l'ensemble du serveur.

Cependant, la température mesurée n'est pas uniquement le reflet de la charge de travail instantanée. Elle est influencée par deux facteurs principaux : l'échauffement propre dû à l'activité du processeur, et l'efficacité du système de refroidissement. Dans le cas spécifique du water-cooling, ce second facteur est potentiellement important. La température de l'eau de refroidissement qui circule au contact du processeur peut jouer un rôle majeur et varier en fonction de plusieurs facteurs comme la charge thermique globale du centre de données voire de la météo, modulant ainsi la température CPU observée indépendamment de son activité propre.

Notre hypothèse est donc double :

1. La température CPU seule est un proxy fiable — au sens d'une variable de substitution — pour estimer la consommation électrique globale du serveur.
2. La précision de cette estimation peut être significativement améliorée si l'on parvient à isoler les variations de températures dues à la charge du serveur de celles provoquées par le système de refroidissement.

Périmètre de l'étude : Le cas des serveurs sans GPU Il est important de souligner que l'approche présentée dans ce chapitre s'applique aux serveurs dépourvus de GPU. L'intégration d'accélérateur graphique, par la consommation électrique massive qui peut surpasser celle du CPU, altère le profil énergétique du serveur et invalide la corrélation

directe avec la seule activité du processeur. Notre étude se concentre donc sur les serveurs "classiques" (processeur, mémoire, stockage), qui constituent toujours la majorité du parc global [2], tout comme chez notre partenaire industriel OVHcloud. La problématique spécifique de l'estimation électrique des GPU sera explorée au Chapitre 5, page 61.

3.1.3 Structure du chapitre

La Section 3.2 détaillera la méthodologie employée, que nous nommerons PowerHeat, en décrivant le banc d'expérimentation, le processus de collecte des données et les approches de modélisation. La Section 3.3 analysera les résultats obtenus, en évaluant la performance des notre approche conformément à nos hypothèses. Enfin, la Section 3.4 présentera une conclusion sur la validité de l'approche PowerHeat, tout en introduisant la problématique de la portabilité des modèles, qui motivera la contribution présentée au Chapitre 4.

3.2 Méthodologie : PowerHeat

L'objectif est de construire un modèle d'estimation de la consommation électrique d'un serveur en utilisant des données thermiques non-intrusives. Nous présentons d'abord l'infrastructure de test, puis le processus de collecte de données, et enfin les approches de modélisation et l'algorithme d'apprentissage utilisés.

3.2.1 Le banc d'expérimentation

Notre étude a été menée sur une infrastructure de production dans un centre de données d'OVHcloud. Dans cette infrastructure nous avons identifié un rack composé de 19 serveurs *bare metal* identiques et partageant la même boucle de refroidissement liquide.

Caractéristiques matérielles

L'homogénéité du matériel est un point clé pour permettre une comparaison juste des résultats. Chaque serveur possède les caractéristiques suivantes :

— **Carte-mère :** SuperMicro X10SDV-4C-TLN2F

— **Processeur :** 1 × CPU Intel Xeon D-1521 (4 cœurs, 8 threads)

— **Mémoire :** 16 Go de RAM

— **Stockage :** 4 disques durs SATA de 6 To et 1 SSD NVMe de 500 Go

— **Gestion :** Puce BMC Aspeed AST2400, supportant l'IPMI 2.0

Système de refroidissement

Les serveurs sont refroidis par une technologie de type *direct-to-chip water-cooling*. Chaque serveur est équipé d'un *waterblock* en cuivre parcouru par de l'eau et placé directement sur le processeur pour absorber la chaleur.

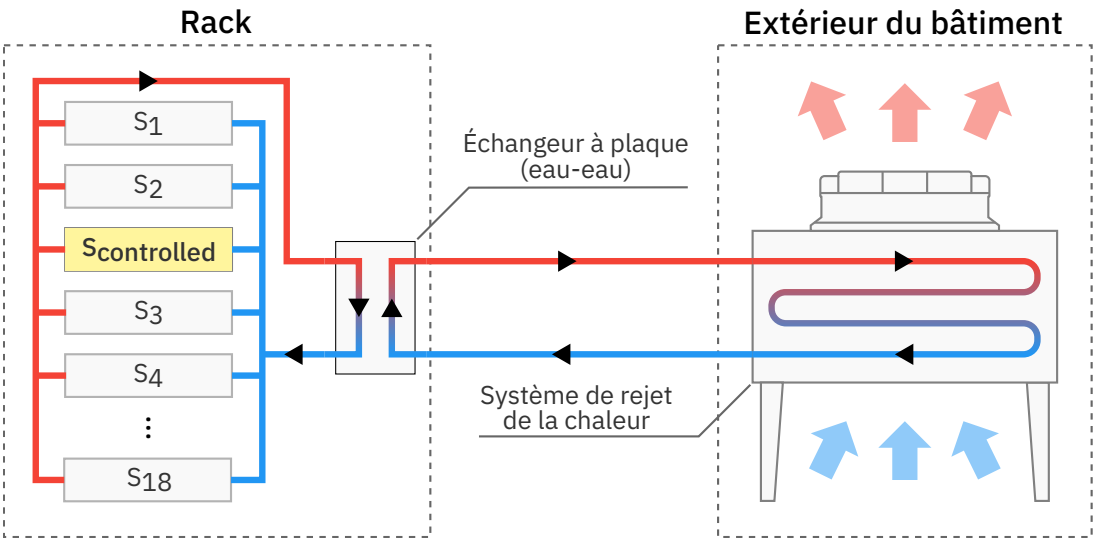


FIGURE 3.1 – Représentation du banc d'expérimentation et de son système de refroidissement.

L'eau chauffée est ensuite évacuée via un circuit partagé par les 19 serveurs du rack (Figure 3.1). La chaleur est échangée avec un circuit d'eau secondaire puis rejetée à l'extérieur du bâtiment. Cette topologie est très efficace sur le plan énergétique mais rend la température du CPU de chaque serveur dépendante non seulement de sa propre charge, mais aussi de la température de l'eau à l'entrée du *waterblock*, qui elle-même est influencée par la charge globale des autres serveurs.

Le serveur de contrôle

Parmi les 19 serveurs, 18 étaient utilisés par des clients externes, avec des charges de travail inconnues et variées, reflétant un cas d'usage de production typique. Le 19ème serveur, que nous nommerons $S_{controlled}$, a été placé sous notre contrôle exclusif. Ce serveur a été maintenu intentionnellement dans un état de repos (idle) constant pendant toute la durée de l'expérience. L'hypothèse est que, sa propre dissipation thermique étant stable et minimale, les variations de la température de son CPU sont principalement dues aux fluctuations de la température de l'eau du circuit de refroidissement commun. $S_{controlled}$ sert ainsi de capteur indirect et non-intrusif de la température de l'eau.

3.2.2 Collecte des données

Deux flux de données distincts ont été collectés en parallèle pour les 19 serveurs sur une période continue de 5 jours (du lundi au vendredi).

1. **Consommation électrique :** La consommation électrique de chaque serveur a été mesurée à l'aide d'un smart PDU. Ces PDU mesurent le courant consommé par chaque prise avec une fréquence de 5 secondes. Les données brutes en ampères ont été converties en Watts en utilisant une tension constante de 230 V et un facteur de puissance estimé à 0,95, typique pour ce type d'alimentation de serveur [68]. Ces mesures de puissance constituent la "vérité terrain" (ground truth) que nos modèles chercheront à prédire.
2. **Température du CPU :** La température du processeur de chaque serveur a été collectée via des requêtes IPMI à distance adressées à la puce BMC. La température instantanée a été relevée toutes les 10 secondes, intervalle minimal d'interrogation de notre système IPMI.

Un travail de pré-traitement des données a été nécessaire pour synchroniser ces deux flux. Les mesures de puissance (toutes les 5 secondes) ont été agrégées par une moyenne pour correspondre aux horodatages des mesures de température (toutes les 10 secondes), créant ainsi un jeu de données unifié où chaque point de donnée représente un instant t avec une valeur de puissance et une valeur de température pour chaque serveur.

3.2.3 Approches de modélisation comparées

Pour évaluer nos hypothèses nous avons défini deux approches de modélisation distinctes. Pour chaque serveur i (parmi les 18 serveurs clients), nous cherchons à prédire sa puissance P_i.

Modèle `ABS_TEMP` : La relation directe

Le premier modèle, le plus simple, est `ABS_TEMP`. Il se fonde sur une relation directe entre la température absolue du processeur (T_i) et la puissance du serveur (P_i).

$$P_i = f(T_i) \tag{3.1}$$

Ce modèle repose sur l'idée que, malgré les fluctuations de la température de l'eau, l'impact de la charge de travail du serveur sur sa propre température est suffisamment dominant pour permettre une prédiction robuste. La fonction f est un modèle de régression entraîné par un algorithme d'apprentissage (*machine learning*). Ce modèle a l'avantage d'être minimaliste : il ne requiert qu'un seul capteur de température par serveur et n'a aucune dépendance externe.

Modèle `REL_TEMP` : L'isolation de la charge de travail

Le second modèle, `REL_TEMP`, met en œuvre une **compensation thermique** visant à décorréler la température du CPU de l'influence du liquide de refroidissement. Pour réaliser cette compensation, il utilise la température du serveur de contrôle $S_{controlled}$ comme une ligne de base thermique. Nous définissons une température relative, θ_i, qui est la différence entre la température du serveur i et celle du serveur de contrôle.

$$\theta_i = T_i - T_{controlled} \tag{3.2}$$

L'idée est que si $T_{controlled}$ capture les variations de la température de l'eau, la soustraction de cette valeur à T_i devrait annuler ce bruit de fond, ne laissant que l'échauffement dû à la charge de travail propre au serveur i. Le modèle `REL_TEMP` cherche donc à prédire la puissance à partir de cette température relative :

$$P_i = g(\theta_i) \tag{3.3}$$

Il est important de souligner que la dépendance à un serveur "témoin" pour mesurer les variations de température de l'eau a été une nécessité expérimentale pour pallier l'absence de capteurs de température du liquide de refroidissement sur notre infrastructure. S'agissant d'une infrastructure de production hébergeant de vrais clients d'OVHcloud, il n'était pas possible de la modifier.

3.2.4 Algorithme et validation

Algorithme : Gradient Boosting (XGBoost)

Comme nous l'avons vu, notre objectif est d'estimer la consommation électrique d'un serveur, une valeur continue exprimée en Watts. En *machine learning*, ce type de problème est appelé une régression. La tâche consiste à entraîner un modèle à prédire une variable continue (la puissance) à partir de caractéristiques d'entrée (ici, la température absolue ou relative).

Clarification terminologique : le terme "prédire" est utilisé en machine learning pour désigner la sortie d'un modèle, et n'implique pas forcément de notion de prédiction "dans le futur". Dans ce chapitre, nous utiliserons conjointement le terme "estimation" pour différencier une puissance "estimée", prédite par un modèle, d'une "mesure" obtenue par un capteur physique.

La relation entre la température et la puissance est supposée non-linéaire : des effets de seuil dus à l'activation de ventilateurs ou à des paliers de fréquence du CPU peuvent introduire des effets complexes. Nous avons donc opté pour un algorithme de type Gradient Boosting, et plus spécifiquement l'implémentation XGBoost [69], reconnue pour ses

performances dans de nombreuses tâches de régression.

Un modèle distinct a été entraîné pour chaque approche (`ABS_TEMP` et `REL_TEMP`) et pour chacun des 18 serveurs. L'optimisation des hyperparamètres de XGBoost (tels que le taux d'apprentissage, la profondeur maximale des arbres, etc.) a été réalisée automatiquement à l'aide de la librairie `hyperopt` [70] et de l'algorithme TPE (Tree of Parzen Estimators) pour trouver la meilleure configuration pour chaque modèle.

Méthode de validation : Validation croisée (K-fold)

Pour évaluer la capacité de généralisation des modèles et éviter le sur-apprentissage, nous avons utilisé une méthode de validation croisée à K-blocs (*K-fold cross-validation*) [71], avec $K = 5$, représentée par la Figure 3.2. Le jeu de données de 5 jours de chaque serveur a été divisé en 5 sous-ensembles, de sorte que chaque sous-ensemble contienne les mesures successives d'un jour de données. Ce découpage temporel a été spécifiquement choisi pour isoler les variations induites par le cycle journalier des températures environnementales du centre de données et des activités humaines.

Le processus d'entraînement a été répété 5 fois : à chaque fois, un sous-ensemble différent était utilisé pour le test et les 4 autres pour l'entraînement. La performance finale du modèle est la moyenne des performances obtenues sur les 5 blocs de test. Cette méthode garantit que chaque point de donnée a été utilisé une fois pour le test, offrant une estimation bien plus robuste de la performance du modèle sur des données nouvelles que ne le ferait une simple division entraînement/test.

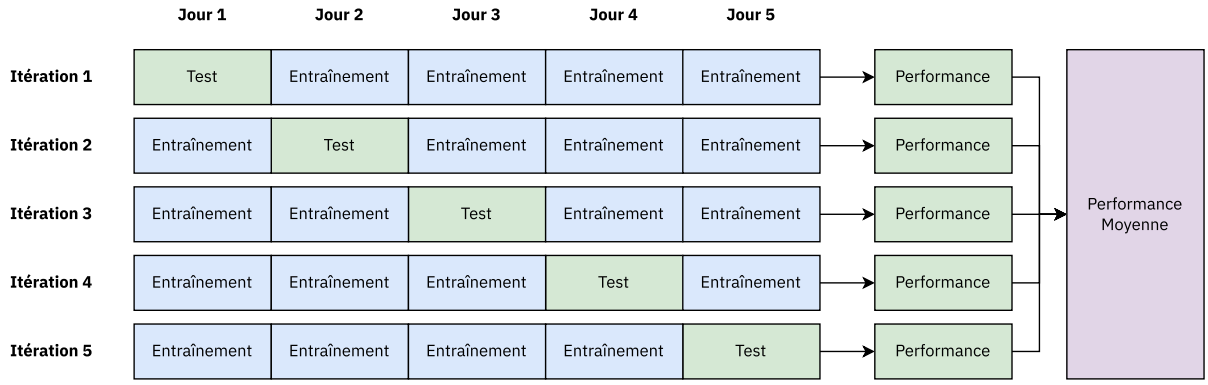


FIGURE 3.2 – Représentation de la méthode d'évaluation des modèles par validation croisée.

3.3 Résultats et Analyse

Cette section présente les observations des mesures de puissance et de température CPU ainsi que les résultats de performance obtenus par les deux approches de modélisation, `ABS_TEMP` et `REL_TEMP`, sur les 18 serveurs de production.

3.3.1 Observations des mesures thermiques et électriques des serveurs

L'expérience s'est déroulée sur 5 jours, durant lesquels une énergie totale de plus de 165 kWh a été consommée par l'ensemble des 19 serveurs, avec une puissance moyenne de 72,4 W par serveur. La Figure 3.3 présente l'évolution des consommations électriques et des températures CPU individuelles.

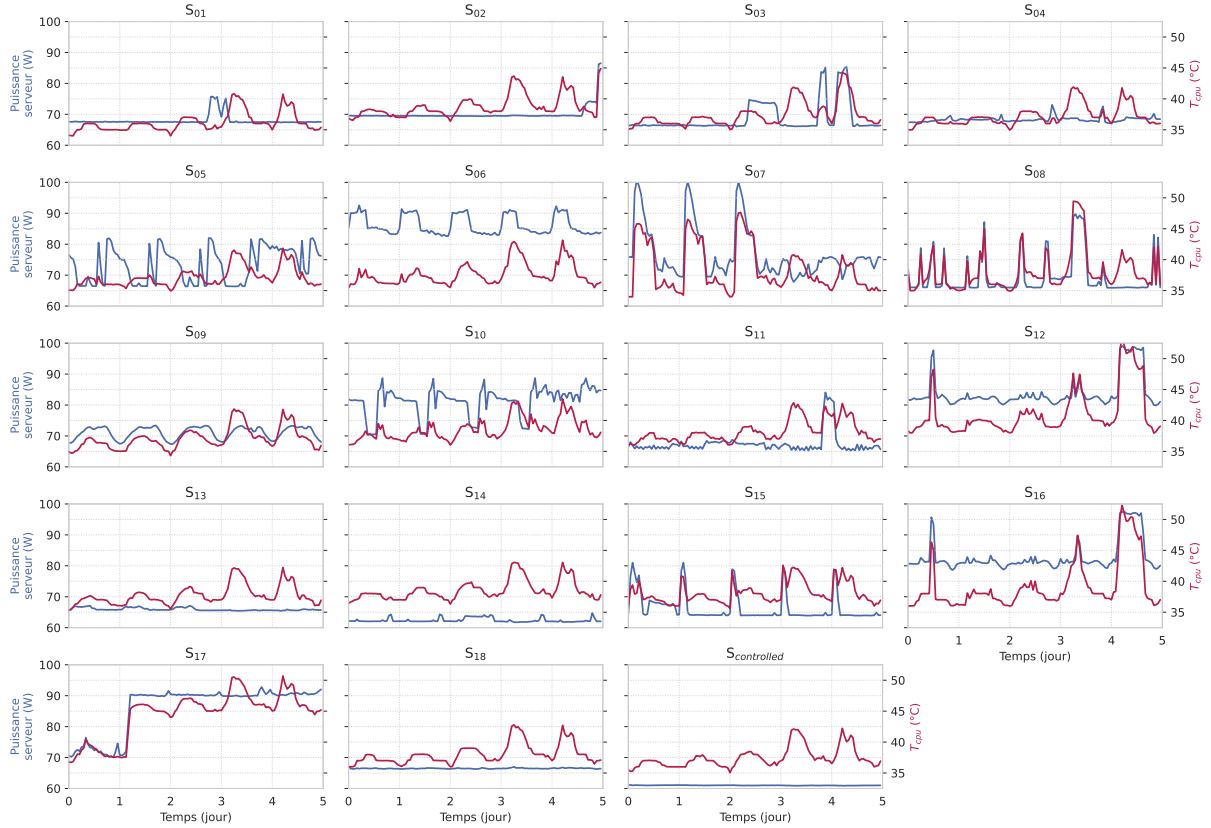


FIGURE 3.3 – Évolution de la consommation électrique et de la température CPU des serveurs sur 5 jours.

On observe une hétérogénéité des charges de travail, avec des puissances fluctuant globalement entre 61 W et 99 W. Le serveur de contrôle, $S_{controlled}$, se distingue par la consommation la plus faible, maintenue à un niveau constant de 61 W, confirmant son état de repos (idle). À l'opposé, certains serveurs comme S_{07}, S_{12} et S_{16} affichent des pics de consommation très marqués, témoignant d'une activité intense et sporadique.

Les températures CPU, révèlent une dynamique particulièrement intéressante : une oscillation quasi-sinusoïdale avec une période d'un jour. Cette observation soutient l'intérêt de la division du jeu de données en fonction du cycle journalier. Les niveaux de température minimum et maximum de ce cycle varient quotidiennement, avec une amplitude maximale lors du troisième jour.

L'observation la plus significative est que la température du CPU de $S_{controlled}$ fluctue

visiblement en suivant ce même cycle journalier, et ce, malgré sa charge de travail nulle et sa consommation électrique constante. Cette corrélation entre le temps et la température renforce notre hypothèse selon laquelle un capteur de température CPU d'un serveur idle peut effectivement servir de capteur indirect pour observer les variations de température de l'eau de refroidissement.

On remarque également que $S_{controlled}$ n'est pas le serveur avec la température CPU la plus basse, bien qu'il soit celui qui consomme le moins. Ce constat, a priori contre-intuitif, peut s'expliquer par plusieurs facteurs non liés à la charge [72] :

— Un biais systématique dû à des différences de calibration entre les capteurs de température des différents processeurs.

— L'influence du positionnement physique dans le rack. La chaleur ayant tendance à monter, les serveurs situés en haut du rack peuvent être exposés à un environnement légèrement plus chaud. La position de $S_{controlled}$ n'est pas connue mais il est probable qu'il ne soit pas positionné au plus bas.

— De minimes différences matérielles (ex : application de la pâte thermique, contact du waterblock sur le CPU) pouvant influencer l'efficacité de la dissipation thermique d'un serveur à l'autre.

Cependant, le fait que $S_{controlled}$ ne soit pas le plus froid n'invalide pas son rôle de témoin. Pour notre modèle `REL_TEMP`, l'important n'est pas la valeur absolue de sa température, mais l'hypothèse que l'effet des variations de température de l'eau sur chaque CPU est homogène. $S_{controlled}$ fournit ainsi une ligne de base dynamique essentielle pour isoler l'échauffement propre à la charge de chaque serveur.

Enfin, la comparaison des deux graphiques renforce l'hypothèse d'une forte corrélation entre puissance et température. Le serveur S_{18}, par exemple, voit sa puissance osciller autour de 70-75 W le premier jour avant de se stabiliser sur un plateau à près de 90 W pour le reste de la période. Sa température CPU suit un schéma identique, tout en superposant l'oscillation journalière observable sur $S_{controlled}$. De même, les pics de consommation électrique très nets des serveurs S_{07}, S_{12}, et S_{16} se traduisent par des pics de température tout aussi visibles sur leurs processeurs respectifs.

Cette analyse exploratoire a permis d'observer la pertinence des signaux thermiques pour estimer la puissance. Elle confirme l'existence d'une relation visible entre la charge d'un serveur et la température CPU, tout en soulignant l'impact non négligeable des variations du système de refroidissement.

L'étape suivante consiste à évaluer de manière quantitative la capacité de nos deux approches de modélisation, `ABS_TEMP` et `REL_TEMP`, à exploiter cette relation pour fournir des estimations de puissance précises.

3.3.2 Performance des modèles

Pour évaluer la précision des prédictions de puissance, nous avons utilisé l'Erreur Absolue Moyenne en Pourcentage - *Mean Absolute Percentage Error* (MAPE), une métrique couramment utilisée qui exprime l'erreur de prédiction en pourcentage de la valeur réelle, la rendant facilement interprétable.

$$MAPE = \frac{1}{n}\sum_{t=1}^{n}\left|\frac{A_t - F_t}{A_t}\right| \tag{3.4}$$

Où A_t est la valeur réelle (mesurée par le PDU) et F_t est la valeur prédite par le modèle.

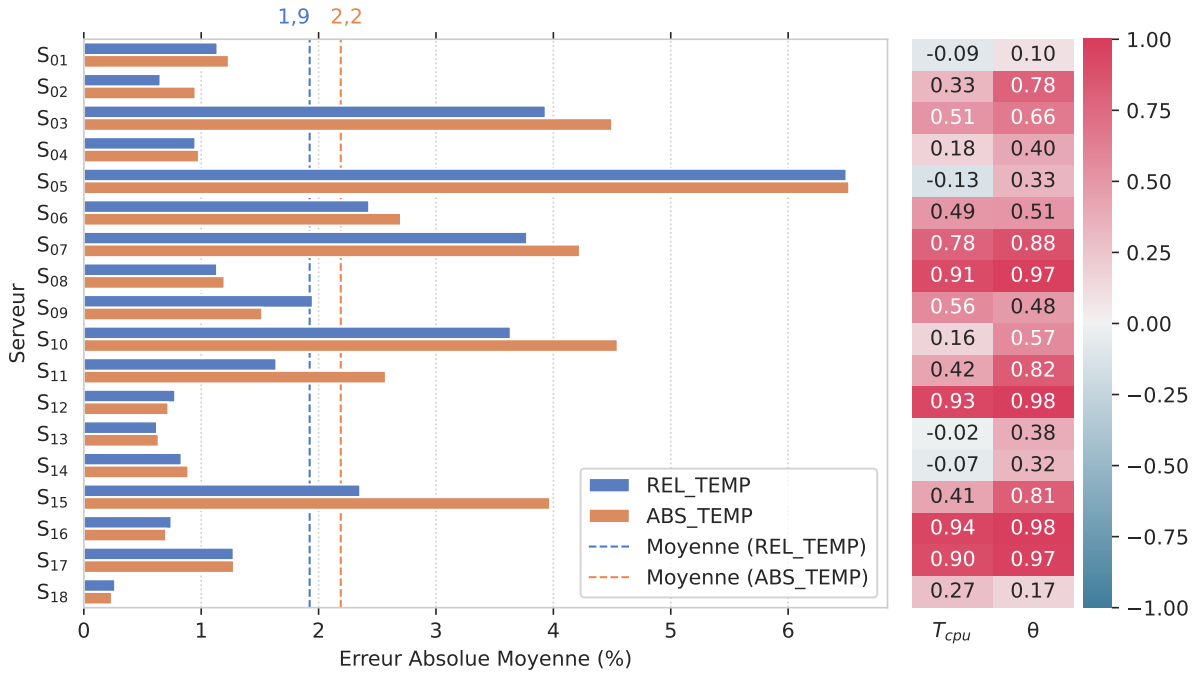


FIGURE 3.4 – Performances des modèles `ABS_TEMP` et `REL_TEMP` pour chaque serveur et coefficient de corrélation de Pearson des entrées des modèles.

La Figure 3.4 présente les performances obtenues pour chaque serveur. Le premier constat est que les deux approches permettent d'estimer la puissance avec une bonne précision. En moyenne sur les 18 serveurs, le modèle `ABS_TEMP` atteint une erreur de 2,2 %, tandis que le modèle `REL_TEMP` obtient une erreur légèrement inférieure de 1,9 %. Par ailleurs, on observe que l'approche `REL_TEMP` est quasi-systématiquement plus performante pour chaque serveur individuel.

Pour comprendre l'origine de cette amélioration, la partie droite de la Figure 3.4 représente la corrélation de Pearson entre la puissance et les variables d'entrée de chaque modèle (T_{cpu} et θ). Hormis pour S_{18}, il apparaît que la corrélation entre la puissance et la température relative (θ) est toujours plus élevée qu'avec la température absolue (T_{cpu}).

Pour certains serveurs, comme S_{01}, S_{05}, S_{13} et S_{14}, le coefficient avec la température absolue est quasi nul. En calculant la température relative, on parvient à obtenir un signal mieux corrélé à la puissance, expliquant ainsi les meilleures performances globales du modèle `REL_TEMP`.

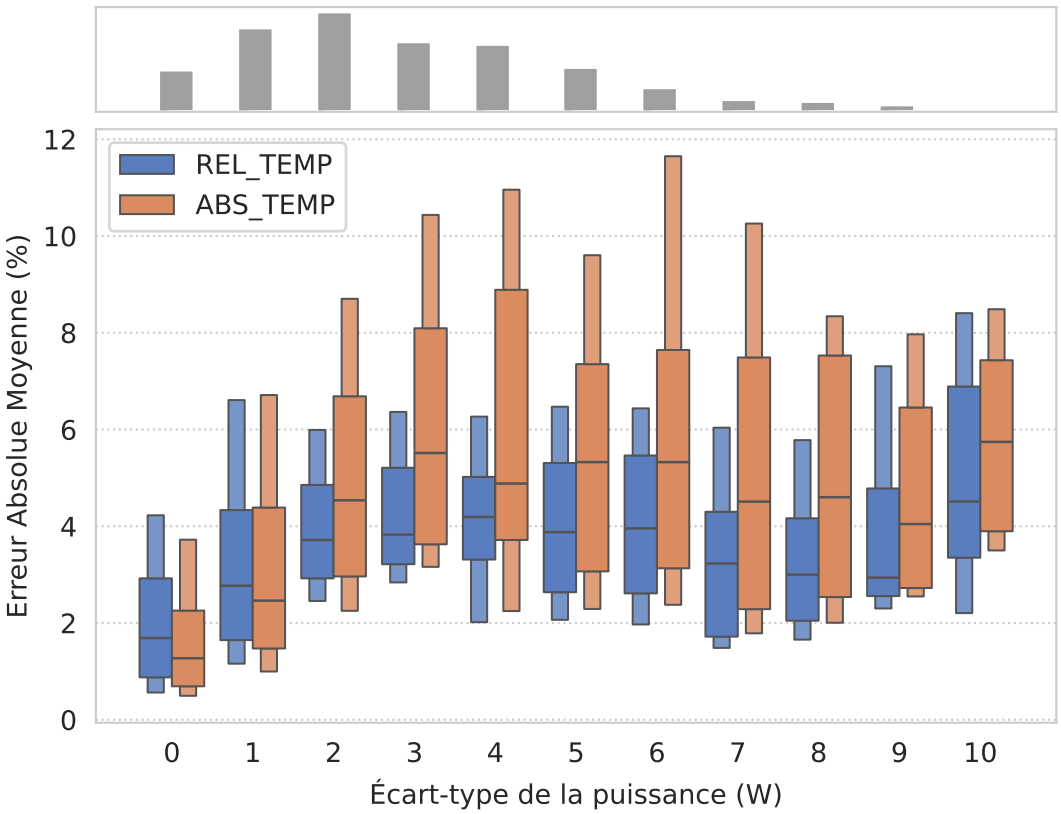


FIGURE 3.5 – Performances des modèles `ABS_TEMP` et `REL_TEMP` en fonction de l'écart-type de puissance (écart-type mobile avec une fenêtre temporelle de 1 heure).

Au-delà de l'erreur moyenne globale, nous avons analysé la performance des modèles en fonction de la complexité de la tâche d'estimation. Cette complexité est directement liée à la variabilité de la consommation électrique : plus celle-ci fluctue, plus la prédiction est difficile. Nous avons quantifié cette dynamique en calculant l'écart-type mobile de la puissance sur une fenêtre glissante d'une heure pour chaque serveur.

Cette métrique permet de distinguer les périodes de charge stable (faible écart-type), où la prédiction est a priori plus aisée, des périodes de forte activité dynamique (écart-type élevé), qui représentent un défi plus important pour les modèles.

La Figure 3.5 analyse l'évolution de l'erreur de prédiction en fonction de cette variabilité. On y observe plusieurs points clés. Premièrement, l'erreur de prédiction des deux modèles augmente avec la variabilité de la puissance puis la médiane se stabilise autour de 4 à 6 % pour `ABS_TEMP` et 3 et 5 % pour `REL_TEMP`. Deuxièmement, l'erreur médiane de `REL_TEMP` est systématiquement inférieure à celle de `ABS_TEMP` lorsque la variabilité augmente. Enfin, la distribution des erreurs du modèle `REL_TEMP` est visiblement plus

resserrée, indiquant que les prédictions sont plus fiables et consistantes notamment face aux charges dynamiques.

L'analyse globale des performances met en évidence une erreur moyenne faible et un avantage pour l'approche `REL_TEMP`. Toutefois, les résultats (Figure 3.4) montrent des différences de performances entre les serveurs. La section suivante examine en détail le comportement des estimations de trois cas spécifiques.

3.3.3 Analyse de cas spécifiques

Cette section analyse en détail le comportement de nos modèles sur des profils de charge spécifiques. La Figure 3.6 présente la comparaison entre la puissance mesurée et les estimations des modèles `ABS_TEMP` et `REL_TEMP` pour l'ensemble des serveurs. Afin d'illustrer les forces et les limites de chaque approche, nous avons sélectionné trois cas particulièrement représentatifs :

— **Le serveur S_{15}**, qui illustre un cas idéal où la compensation thermique du modèle `REL_TEMP` permet d'isoler efficacement le signal de la charge et d'améliorer nettement la précision de l'estimation.

— **Le serveur S_{05}**, qui constitue le cas le plus difficile pour nos modèles. Il enregistre les performances les moins favorables, sans qu'une approche ne se distingue de l'autre, mettant en lumière les limites physiques de l'estimation par proxy thermique.

— **Le serveur S_{09}** met en évidence un cas d'exception où le filtrage thermique de `REL_TEMP` s'avère contre-productif, rendant le modèle `ABS_TEMP` plus performant.

S_{15} : Charge dynamique et forte corrélation

Le serveur S_{15} (Figure 3.7) présente un profil de consommation avec des pics de charge intenses et périodiques suivis de phases de faible activité. La température CPU (Figure 3.7a) reflète globalement cette activité électrique. Cependant, on observe certaines variations de température qui n'ont pas de correspondance électrique, comme le second pic au jour 3 et les fluctuations après le pic du jour 4. Le calcul de la température relative permet de filtrer en partie ces perturbations.

Les deux modèles suivent la consommation réelle avec une bonne fidélité. L'analyse des nuages de points (Figure 3.8b) montre que la relation puissance-température absolue est claire mais présente une grande dispersion notamment aux niveaux de puissance les plus bas. La température relative permet de corriger cette dispersion et la corrélation avec la puissance est bien plus resserrée. Ce cas illustre une configuration où la compensation thermique du modèle `REL_TEMP` est particulièrement efficace pour isoler l'échauffement lié à la charge.



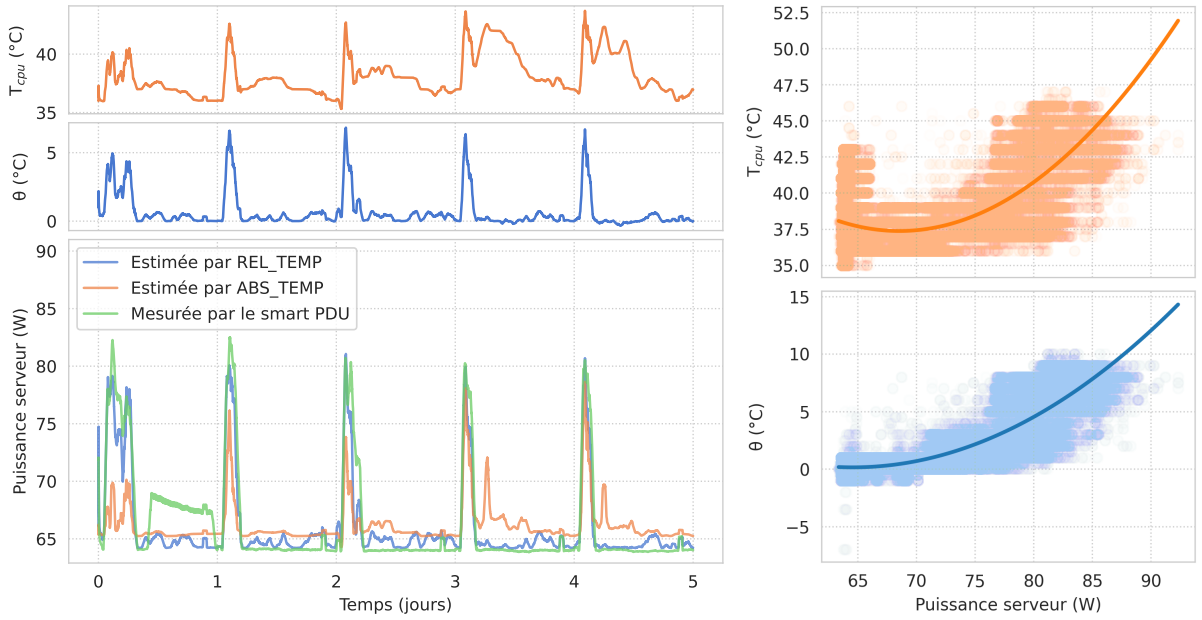
FIGURE 3.6 – Comparaison des estimations de puissance des modèles `ABS_TEMP` et `REL_TEMP` avec la consommation réelle.

S_{05} : Charge erratique

Le serveur S_{05} est le cas le plus difficile pour nos modèles ; il présente l'erreur de prédiction la plus élevée, avec 6,5 % d'erreur aussi bien pour l'approche `ABS_TEMP` que `REL_TEMP`. Comme l'illustre la Figure 3.8a, son profil de consommation est caractérisé par de nombreuses fluctuations et de faible amplitude. On constate que le signal de température du CPU est beaucoup plus lisse et ne reflète pas ces variations brèves. Par conséquent, les estimations des modèles parviennent à suivre la tendance de fond mais échouent à capturer les pics de consommation instantanés.

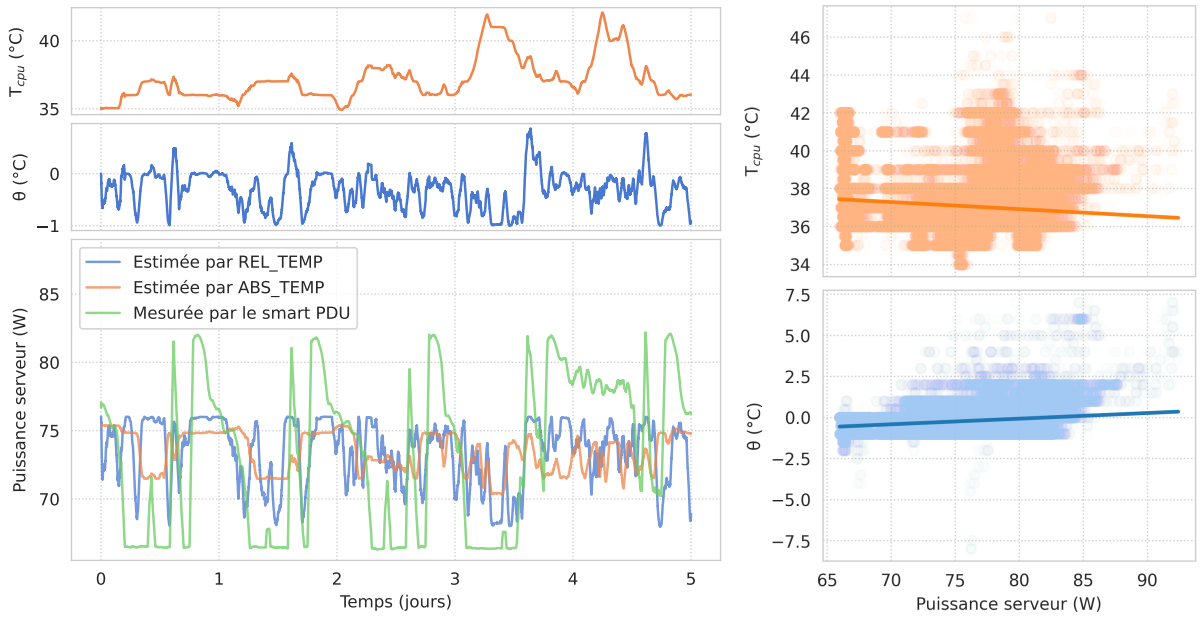
Ce phénomène met en lumière une limite physique de la méthode : l'inertie thermique, couplée à l'efficacité du système de refroidissement. La chaleur générée par un bref pic de puissance est évacuée par le *water-cooling* si rapidement que le processeur n'a pas le temps de monter significativement en température. Ces événements électriques deviennent donc invisibles pour le capteur thermique.

Cette invisibilité se traduit visuellement sur la Figure 3.8b, où la relation puissance-température apparaît comme un nuage dispersé, sans structure claire. L'absence de corrélation forte empêche le modèle d'apprendre une relation fiable. Cette mauvaise perfor-



(a) (b)

FIGURE 3.7 – Analyse détaillée du serveur S_{15} : évolution temporelle des signaux (gauche) et corrélation puissance-température (droite).

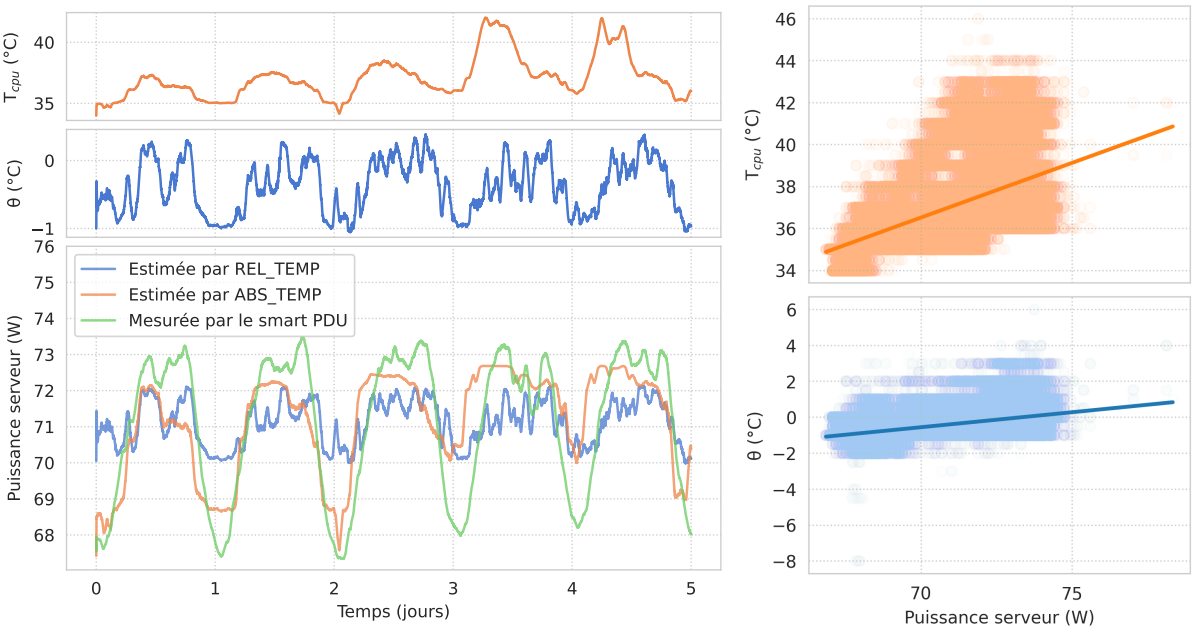


(a) (b)

FIGURE 3.8 – Analyse détaillée du serveur S_{05} : évolution temporelle des signaux (gauche) et corrélation puissance-température (droite).

mance est une limite inhérente à l'usage d'un proxy thermique lorsque le refroidissement est plus réactif que la charge à mesurer.

S_{09} : Limite de la compensation thermique



(a) (b)

FIGURE 3.9 – Analyse détaillée du serveur S_{09} : évolution temporelle des signaux (gauche) et corrélation puissance-température (droite).

Le cas du serveur S_{09} (Figure 3.9) révèle une limite importante de l'approche par température relative. En effet, l'amplitude de puissance de ce serveur est faible (environ 5 W) et les variations de sa température CPU sont perturbées par un signal thermique exogène, causé par les variations de température de l'eau du système de refroidissement. Ces variations exogènes apparaissent décalées par rapport à l'activité propre de S_{09}.

Le modèle `ABS_TEMP` parvient à suivre ces oscillations, tandis que le modèle `REL_TEMP` produit une estimation "aplatie". La corrélation (Figure 3.9b) montre une ligne quasi-verticale entre la puissance et la température relative. Ainsi, le calcul de la température relative semble avoir retiré une partie importante de l'information thermique, ce qui a pour effet de diminuer les performances de `REL_TEMP` avec 1,9 % d'erreur contre 1,5 % d'erreur pour `ABS_TEMP`.

3.3.4 Discussion

L'analyse des performances a montré que les deux approches, `ABS_TEMP` et `REL_TEMP`, permettent d'estimer la consommation électrique avec une bonne précision. Les résultats valident notre hypothèse principale : la température du processeur est un indicateur fiable de la puissance totale d'un serveur. L'approche `REL_TEMP`, avec une erreur moyenne de 1,9 %, se révèle quasi systématiquement plus performante que l'approche `ABS_TEMP` (2,2 %

d'erreur), confirmant ainsi notre seconde hypothèse selon laquelle la compensation des variations thermiques du système de refroidissement permet d'affiner la prédiction.

L'amélioration de la performance du modèle `REL_TEMP` s'explique par sa capacité à mieux isoler le signal thermique directement lié à la charge de travail. En soustrayant la température du serveur de contrôle, on filtre une partie significative du "bruit" thermique généré par le cycle journalier du système de refroidissement. L'augmentation quasi générale du coefficient de corrélation de Pearson entre la puissance et la température relative (θ) en est la principale illustration. Au-delà de l'erreur moyenne, l'avantage de `REL_TEMP` est particulièrement visible sur les charges dynamiques (Figure 3.5), où sa distribution d'erreur plus resserrée indique une fiabilité et une consistance supérieures.

L'analyse de cas spécifiques révèle cependant les limites de cette approche. Le serveur S_{05}, avec sa charge très erratique, illustre une limite physique fondamentale liée à l'efficacité du système de refroidissement. Pour des fluctuations de puissance brèves et de faible amplitude, le système de water-cooling évacue la chaleur si rapidement que le processeur n'a pas le temps de monter en température de manière significative. Ces variations électriques deviennent alors invisibles pour le capteur de température. Le signal est en quelque sorte "lissé" par la réactivité du refroidissement, et le modèle ne peut logiquement pas prédire des événements qui n'ont aucune signature dans les données d'entrée. Il ne s'agit pas d'une faiblesse de la modélisation, mais bien d'une limite inhérente à l'utilisation d'un proxy thermique dans un environnement à refroidissement très performant. Néanmoins, une plus grande résolution de mesure et une plus grande fréquence d'échantillonnage des signaux thermiques pourraient éventuellement améliorer cette la précision, bien que cela dépasse les limites techniques d'IPMI dans notre contexte.

Le cas du serveur S_{09} illustre une autre limite, cette fois méthodologique. L'échec relatif de `REL_TEMP` sur ce serveur s'explique par une corrélation entre la charge du serveur et le cycle du système de refroidissement. En tentant de supprimer ce qui est perçu comme du bruit, le modèle a en réalité retiré une partie de l'information pertinente.

D'un point de vue pratique, ces résultats offrent un compromis clair pour un opérateur. L'approche `ABS_TEMP` représente la solution la plus simple, sans aucune dépendance externe, et offre une précision déjà suffisante pour de nombreux cas d'usage comme la supervision générale. L'approche `REL_TEMP` implique une complexité supplémentaire (la nécessité de définir une ligne de base thermique) mais fournit une précision accrue, indispensable pour des applications plus exigeantes telles que la facturation à l'usage. Dans les deux cas, la méthode PowerHeat démontre qu'il est possible d'obtenir une estimation de puissance fine et de manière non-intrusive.

3.4 Conclusion

Ce chapitre a abordé la problématique de la mesure de consommation électrique à l'échelle du serveur *bare metal* individuel dans les centres de données, une pratique es-

sentielle mais souvent entravée par des contraintes économiques et opérationnelles. Pour répondre à ce besoin, nous avons présenté une méthode non-intrusive, nommée PowerHeat, qui estime la puissance d'un serveur en se basant sur la température de son processeur, une métrique généralement collectée par les opérateurs.

Les résultats expérimentaux ont confirmé la validité de notre hypothèse initiale : il existe une corrélation forte et exploitable entre la température du CPU et la consommation électrique globale d'un serveur. L'approche la plus directe, `ABS_TEMP`, a démontré sa capacité à estimer la puissance avec une erreur moyenne de 2,2 %, offrant déjà une solution simple et actionnable. De plus, notre seconde hypothèse a également été validée : en isolant l'échauffement propre à la charge des variations thermiques de l'infrastructure, la précision est améliorée. Le modèle `REL_TEMP`, qui met en œuvre cette compensation, a atteint une erreur moyenne de 1,9 % (min : 0,3 % et max : 6,5 %) et s'est montré plus fiable face à des charges de travail dynamiques.

La méthode PowerHeat offre un compromis : l'approche `ABS_TEMP` constitue une solution minimaliste et sans dépendance externe, adaptée à la supervision générale. L'approche `REL_TEMP`, bien que nécessitant une ligne de base thermique, fournit une précision accrue, indispensable pour des applications plus exigeantes comme la facturation à l'usage. Dans les deux cas, la méthode montre qu'il est possible d'obtenir une mesure de puissance fine sans déploiement matériel coûteux ni interruption de service. Elle ouvre la voie à un suivi énergétique fin, serveur par serveur, à faible coût, permettant d'alimenter des outils de supervision, d'optimisation énergétique, de facturation à l'usage ou de comptabilité carbone.

L'approche présentée dans ce chapite a une limite fondamentale : le processus nécessite, pour chaque serveur que l'on souhaite modéliser, une phase d'entraînement initiale. Cette phase d'entraînement requiert de disposer, pendant un temps donné, de mesures simultanées de la température (via IPMI) et de la consommation électrique réelle (via un wattmètre, typiquement sur un PDU intelligent). Une fois le modèle entraîné, le wattmètre n'est plus nécessaire.

Mais cela soulève plusieurs questions : comment estimer la puissance d'autres serveurs qui, par définition, ne possèdent pas et n'ont jamais possédé de wattmètre ? Dans quelle mesure un modèle peut-il estimer la puissance d'un autre serveur ? Peut-on développer une méthode permettant d'estimer la puissance de n'importe quel serveur de notre infrastructure ?

La véritable valeur d'une telle méthode réside dans sa capacité à être portable et à être généralisable. Le défi n'est plus seulement de modéliser un serveur instrumenté, mais de construire un modèle capable d'estimer la puissance d'autres serveurs sans jamais avoir eu accès aux données de consommation réelles.

Cette nouvelle problématique de la portabilité et de la généralisation constitue le cœur de la contribution qui sera présentée dans le chapitre suivant. Nous y explorerons des techniques pour construire un modèle unique et universel, entraîné sur un petit nombre de

serveurs de référence, et capable de s'adapter pour estimer avec précision la consommation de milliers d'autres serveurs "aveugles" avec des caractéristiques matérielles différentes.

Chapitre 4

Généralisation des modèles de puissance : La méthodologie PPEM-BM

Sommaire

4.1 Introduction **45**
4.2 Méthodologie **46**
4.2.1 Constitution de la bibliothèque de modèles 47
4.2.2 L'évaluation croisée pour mesurer la portabilité 47
4.2.3 La fonction de sélection par apprentissage du classement . . . 48
4.2.4 Processus d'estimation 49
4.3 Expérience **49**
4.3.1 Plateforme expérimentale 49
4.3.2 Méthodes de comparaison 51
4.3.3 Protocole de validation 52
4.4 Résultats et discussions **53**
4.4.1 Performances globales 53
4.4.2 Analyse du modèle LTR 54
4.4.3 Représentativité des résultats 56
4.5 Conclusion **58**

4.1 Introduction

Le chapitre précédent a démontré la viabilité de la méthode PowerHeat pour estimer avec précision la consommation électrique d'un serveur *bare metal* à partir de la température de son processeur. Cette approche non-intrusive propose une solution au problème de la mesure pour un serveur individuel, à condition de disposer d'une phase d'entraînement initiale où sa consommation réelle est mesurée.

Toutefois, cette condition même constitue la limite de l'approche PowerHeat dans le cadre d'un déploiement à grande échelle. Le processus nécessite, pour chaque serveur, un entraînement spécifique qui requiert un accès temporaire à des mesures de puissance réelles, typiquement via un wattmètre. Or, dans un contexte industriel gérant des centaines de milliers de machines, équiper chaque serveur, même temporairement, est opérationnellement et économiquement irréalisable. Chaque modèle PowerHeat a l'expertise de son propre serveur, mais il n'est pas conçu pour généraliser ses connaissances à une autre machine.

Cette spécificité soulève une nouvelle question, directement issue de la conclusion du chapitre précédent : comment peut-on estimer la consommation d'un serveur "cible", qui n'a jamais été équipé d'un wattmètre, en utilisant les connaissances acquises sur un parc de serveurs "sources" déjà modélisés ? Le défi n'est plus seulement de modéliser un serveur, mais de rendre cette modélisation portable. Il s'agit de développer une méthode qui permette de transférer la capacité d'estimation d'un groupe de serveurs instrumentés vers une flotte bien plus large de serveurs "aveugles".

Pour répondre à cette problématique, ce chapitre introduit la méthodologie PPEM-BM. L'approche PPEM-BM propose une solution pragmatique et innovante au défi de la généralisation.

Son principe ne consiste pas à construire un modèle unique et universel, mais à développer un mécanisme de sélection intelligent. L'idée est de constituer une "bibliothèque" de modèles PowerHeat pré-entraînés, puis, pour un nouveau serveur cible, de sélectionner le modèle le plus adéquat de cette bibliothèque. Cette sélection s'opère sur la base de caractéristiques facilement accessibles de manière non-intrusive : la configuration matérielle du serveur et des statistiques issues de son historique de température CPU.

Ce chapitre est structuré comme suit. La Section 4.2 détaille la méthodologie PPEM-BM. La Section 4.3 décrit le contexte expérimental et le protocole mis en place pour évaluer l'approche sur un parc de serveurs de production. La Section 4.4 analyse les résultats obtenus en comparant PPEM-BM à d'autres méthodes de référence. Enfin, la Section 4.5 conclue sur les apports de cette méthodologie et discute des perspectives pour de futurs travaux.

4.2 Méthodologie

La méthodologie PPEM-BM se décompose en trois étapes principales pour sa phase de construction, suivies d'un processus d'exploitation en cinq étapes pour l'estimation sur de nouveaux serveurs. La Figure 4.1 illustre le fonctionnement de la méthodologie.

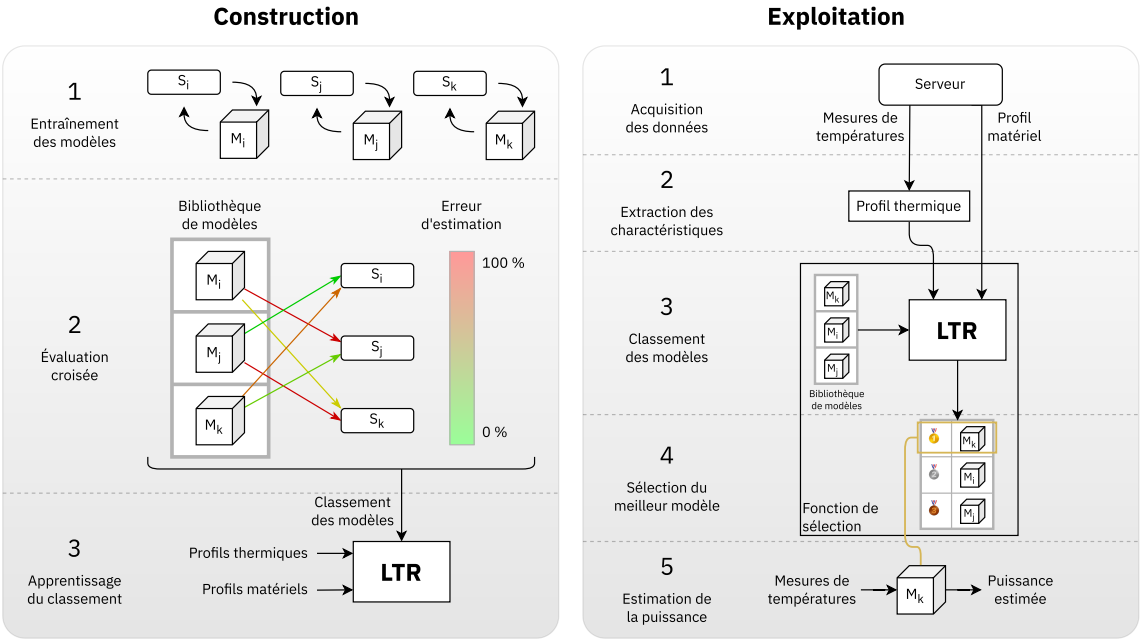


FIGURE 4.1 – Vue d'ensemble de la méthodologie PPEM-BM.

4.2.1 Constitution de la bibliothèque de modèles

Le point de départ de PPEM-BM est la création d'une collection de modèles de puissance individuels. Sur un sous-ensemble de serveurs équipés de capteurs de puissance, un modèle PowerHeat est entraîné pour chaque machine. Les approches `ABS_TEMP` et `REL_TEMP` telle que présentée dans le Chapitre 3 (page 25) peuvent être utilisées. Du fait de l'absence d'équipements permettant de mesurer la température de l'eau et donc de mettre en œuvre la compensation thermique à grande échelle (comme un capteur de température d'eau), l'expérience présentée dans ce chapitre s'appuie uniquement sur l'approche `ABS_TEMP`.

Chaque modèle apprend la relation spécifique entre la température CPU et la puissance pour un serveur donné. L'ensemble de ces modèles constitue la "bibliothèque de modèles", qui sert de base de connaissances pour l'estimation sur d'autres serveurs.

4.2.2 L'évaluation croisée pour mesurer la portabilité

Une fois la bibliothèque constituée, il est essentiel de comprendre dans quelle mesure un modèle est "portable". Pour ce faire, nous procédons à une évaluation croisée. Le principe est simple : chaque modèle de la bibliothèque, entraîné sur un serveur "source" A, est utilisé pour estimer la puissance de tous les autres serveurs "cibles" (B, C, D, etc.) du parc instrumenté. L'erreur de chaque estimation est calculée en comparant la prédiction à la consommation réelle mesurée sur le serveur cible.

Cette étape est cruciale car elle génère les données d'entraînement pour la phase suivante. Elle est cependant coûteuse en calcul, avec une complexité de $O(N^2)$ où N est le nombre de serveurs "sources". Pour chaque serveur cible de notre parc instrumenté, nous obtenons un classement complet de tous les modèles de la bibliothèque, du plus performant (erreur la plus faible) au moins performant (erreur la plus importante).

4.2.3 La fonction de sélection par apprentissage du classement

La question centrale est maintenant de savoir comment, pour un nouveau serveur dont on ne connaît pas la consommation, sélectionner le modèle le plus à même d'estimer sa consommation électrique. C'est le rôle de la fonction de sélection, que nous implémentons à l'aide d'une approche d'Apprentissage du Classement - *Learning To Rank* (LTR) [73].

LTR est une approche d'apprentissage automatique qui vise à classer des éléments en fonction de leur pertinence par rapport à une requête donnée. Un modèle LTR apprend à prédire l'ordre de pertinence des éléments (ici, les modèles de puissance) pour une requête donnée (ici, un serveur cible). Il est particulièrement utile dans des situations où il est important de comprendre non seulement quel est le meilleur élément, mais aussi pourquoi il est le meilleur.

Le choix d'un modèle LTR plutôt qu'une approche plus simple comme la classification vise à maximiser la qualité de l'apprentissage. Dans le cas d'une simple classification, l'algorithme chercherait uniquement à identifier le meilleur modèle source (celui arrivé au premier rang) pour un serveur cible donné. Cette approche limiterait les données d'entraînement à une seule relation positive par serveur cible, exposant le modèle à un sous-apprentissage et à un biais élevé. En revanche, en utilisant un modèle LTR, l'algorithme apprend des relations entre le serveur cible et l'ensemble des serveurs sources, en exploitant l'intégralité du classement de performance. Cela démultiplie la quantité d'informations disponibles pour l'entraînement, permettant au modèle de comprendre des nuances plus fines sur les raisons pour lesquelles un modèle est légèrement meilleur qu'un autre pour une cible donnée.

Dans le cadre de cette étude, nous avons choisi d'implémenter le modèle LTR en utilisant l'algorithme XGBoost [69]. Ce choix est motivé par la capacité de XGBoost à gérer efficacement des données structurées et à capturer des relations non linéaires complexes.

Le modèle LTR est entraîné sur les résultats de l'évaluation croisée. Pour chaque paire (serveur cible, modèle source), il apprend à prédire le rang de performance en se basant sur un ensemble de caractéristiques :

— **Les configurations matérielles :** Nombre de cœurs CPU, *Thermal Design Power* (TDP) du processeur, quantité de mémoire, nombre et type de disques de stockage, etc. Ces caractéristiques sont comparées entre le serveur source et le serveur cible.

— **Les statistiques thermiques :** Des indicateurs simples calculés sur l'historique de température CPU, tels que la moyenne, l'écart-type, les quartiles, le minimum et le maximum. Ces statistiques reflètent le profil thermique opérationnel du serveur.

4.2.4 Processus d'estimation

Une fois le système PPEM-BM entraîné, l'estimation de la puissance pour n'importe quel serveur "aveugle" de l'infrastructure suit le processus suivant :

1. **Acquisition des données :** Collecter l'historique récent de la température CPU du serveur cible via IPMI, ainsi que sa configuration matérielle.
2. **Extraction des caractéristiques :** Calculer les statistiques thermiques (moyenne, écart-type, etc.) à partir de l'historique de température.
3. **Classement des modèles :** Utiliser le modèle LTR entraîné pour classer tous les modèles de la bibliothèque en fonction de leur adéquation estimée pour le serveur cible.
4. **Sélection du meilleur modèle :** Choisir le modèle qui a obtenu le premier rang (Top-1) dans le classement.
5. **Estimation de la puissance :** Appliquer le modèle sélectionné à la série temporelle de température du serveur cible pour obtenir une estimation de sa consommation électrique.

4.3 Expérience

Pour évaluer la méthodologie PPEM-BM, nous l'avons mise en œuvre et testée sur un large parc de machines en conditions réelles de production.

4.3.1 Plateforme expérimentale

Notre étude s'appuie sur des données collectées à partir d'un parc de 1076 serveurs de production d'OVHcloud. Ce parc a la particularité d'être équipé de capteurs de puissance au niveau des blocs d'alimentation (PSU), permettant une mesure fiable de la consommation électrique réelle. Il présente une diversité de configurations matérielles, notamment en termes de processeurs, de quantité de mémoire et de dispositifs de stockage. Pour les besoins de cette expérience, nous avons collecté les séries temporelles de température CPU via IPMI et de puissance électrique à l'entrée des PSU pour chacun de ces serveurs. La campagne de collecte s'est déroulée sur deux mois, entre novembre et décembre 2024,

avec une fréquence d'échantillonnage d'une mesure de la valeur instantanée toutes les deux minutes pour chaque capteur.

La Figure 4.2 détaille la répartition des configurations matérielles au sein de ce parc de 1076 serveurs. Elle présente les distributions pour quatre caractéristiques principales :

— le nombre de cœurs CPU ;

— le TDP (Thermal Design Power) du processeur en Watts ;

— le nombre total de périphériques de stockage (tous types confondus : HDD, SSD, SSD NVMe) ;

— la quantité de mémoire vive en Gio.

L'analyse de ces histogrammes met en évidence une hétérogénéité notable, tout en faisant émerger des configurations prédominantes. Le serveur médian de cet échantillon s'établit à 24 cœurs CPU, avec un TDP de 180W, 192 Gio de mémoire et 5 périphériques de stockage.

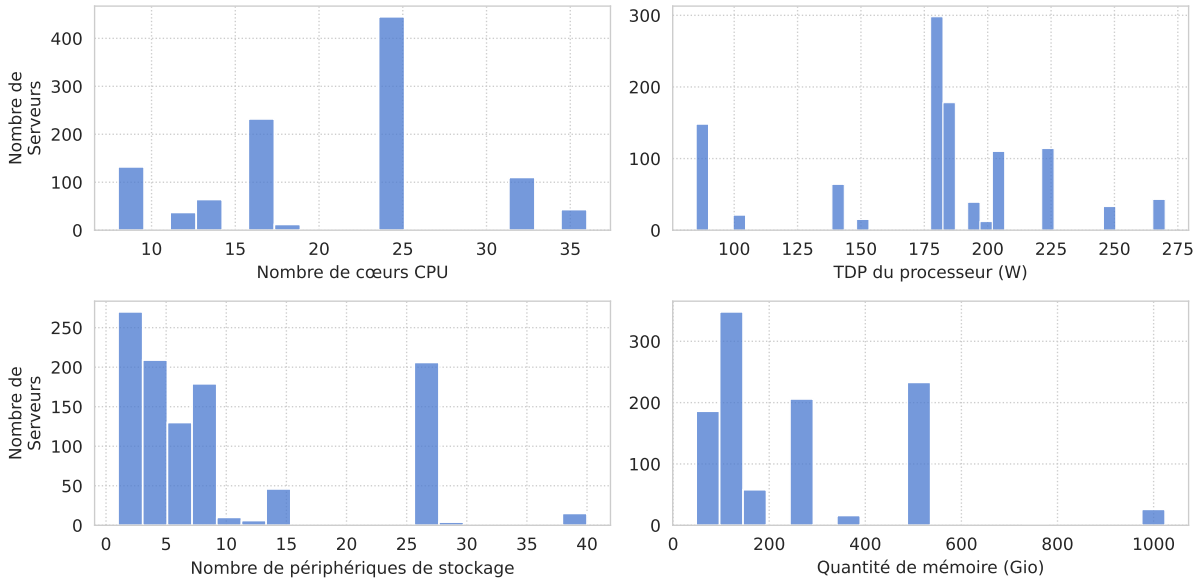


FIGURE 4.2 – Répartition des configurations matérielles des 1076 serveurs.

Pour compléter cette description, la Figure 4.3 met en lumière la relation entre les spécifications matérielles et la consommation électrique. Elle illustre la puissance totale observée sur le parc en fonction des quatre mêmes caractéristiques matérielles. Pour chaque graphique, la consommation est décomposée en trois quantiles de population : 5ème quantile, médiane ; 95ème quantile.

Ces graphiques confirment la corrélation positive : plus les composants sont nombreux ou performants (augmentation du nombre de cœurs CPU, du TDP, du nombre de périphériques de stockage ou de la quantité de mémoire), plus la consommation électrique globale du serveur augmente.

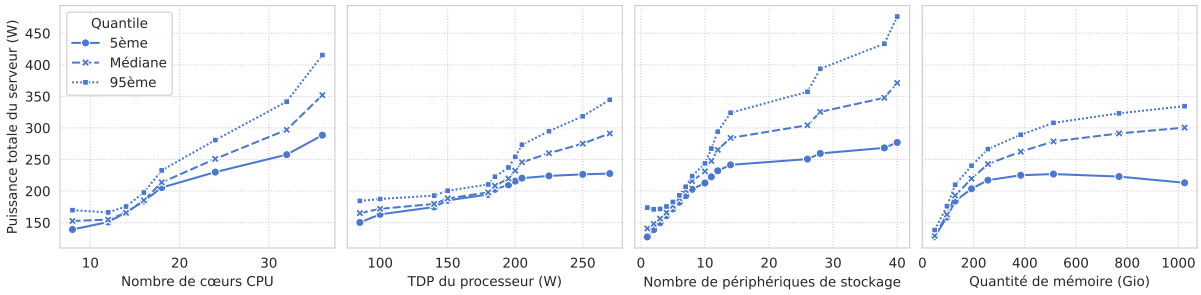


FIGURE 4.3 – Consommation électrique en fonction de la configuration matérielle

4.3.2 Méthodes de comparaison

Pour évaluer rigoureusement la performance de PPEM-BM, nous l'avons comparée à trois autres approches qui servent de points de référence :

PowerHeat : L'approche non-portable, décrite en détail au Chapitre 3. Elle établit la performance optimale théorique en entraînant un modèle de régression par Gradient Boosting pour chaque serveur. Ce modèle apprend la corrélation spécifique entre la température du CPU (obtenue via IPMI) et la puissance électrique globale de cette machine. Pour établir cette "borne inférieure", la performance est mesurée en entraînant le modèle sur les 80 % premières données temporelles du serveur et en le validant sur les 20 % restantes. L'erreur de PowerHeat représente donc le plancher minimal atteignable avec un modèle spécialisé.

Boavizta : Il s'agit d'un outil open-source externe [74], plus précisément la BOAVIZTA-API, conçu pour estimer l'impact environnemental des services numériques, y compris la consommation électrique d'un serveur. Le modèle Boavizta base son estimation sur la configuration matérielle (nombre de CPU, TDP, quantité de mémoire) et un taux d'utilisation global de la machine. Dans notre contexte, ce taux d'utilisation n'est pas connu. Pour pouvoir tout de même utiliser cet outil comme point de comparaison, nous avons estimé un taux d'utilisation u pour chaque serveur en nous basant sur les mesures de puissance réelles dont nous disposions, en utilisant l'Équation 4.1.

$$u = \frac{P_{moyenne} - P_{5\%}}{P_{95\%} - P_{5\%}} \tag{4.1}$$

Où :

— $P_{moyenne}$ représente la puissance moyenne observée ;

— $P_{5\%}$ le 5ème quantile : consommation quasi-à vide ;

— $P_{95\%}$ le 95ème quantile : consommation proche du pic de charge.

Les estimations fournies par l'API pour nos 1076 serveurs, en utilisant ce taux d'utilisation estimé, ont ensuite été comparées à la puissance moyenne réelle. Boavizta représente ainsi une approche générique, non spécialisée à notre infrastructure.

***Hardware-Profiled Average Consumption Estimator* (H-PACE) :** Afin d'établir un point de comparaison plus spécifique à notre infrastructure, nous avons développé notre propre baseline : H-PACE. L'objectif de cette approche est de quantifier la performance atteignable par un modèle qui se base uniquement sur la configuration matérielle. Le principe sous-jacent s'appuie sur la relation entre la configuration matérielle d'un serveur et sa consommation électrique : plus ses composants sont nombreux et puissants, plus sa consommation est élevée.

Nous avons entraîné un modèle d'apprentissage automatique à prédire la puissance moyenne d'un serveur à partir de ses caractéristiques statiques : nombre de cœurs et TDP du CPU, quantité de mémoire, et nombre de dispositifs de stockage. En étant entraîné sur les données des serveurs d'OVHcloud, H-PACE est intrinsèquement plus adapté à notre parc mais il reste fondamentalement limité : il ignore complètement une dimension importante de la consommation, à savoir l'usage et la charge de travail réels de la machine. Il ne peut donc, par conception, qu'estimer une consommation "typique" et non les variations dynamiques.

4.3.3 Protocole de validation

Pour évaluer et comparer les différentes approches, nous avons divisé l'ensemble des 1076 serveurs instrumentés en deux sous-ensembles :

— Un ensemble d'entraînement de 860 serveurs (80 %) ;

— Un ensemble de test de 216 serveurs (20 %).

Ce protocole nous permet de simuler un déploiement en conditions réelles où les modèles, entraînés sur un parc connu, doivent estimer la consommation de nouvelles machines "aveugles". L'évaluation des quatre méthodes a été menée comme suit :

1. Pour PPEM-BM : L'ensemble d'entraînement a été utilisé pour une double fonction : d'abord, pour construire la bibliothèque des modèles PowerHeat individuels, puis pour générer les données de l'évaluation croisée nécessaires à l'entraînement de la fonction de sélection LTR. L'ensemble de test a ensuite été utilisé pour l'évaluation finale. Pour chaque serveur de cet ensemble, nous avons simulé le cas d'un serveur "aveugle", appliqué le processus PPEM-BM (calcul des caractéristiques, classement par le LTR, sélection du meilleur modèle) et estimé sa puissance.
2. Pour H-PACE : L'approche généralisable a suivi le même protocole. Le modèle d'apprentissage a été entraîné pour prédire la puissance moyenne à partir des carac-

téristiques matérielles des serveurs de l'ensemble d'entraînement, puis a été utilisé pour prédire la consommation des serveurs de l'ensemble de test.

3. Pour Boavizta : La performance de cette approche a également été mesurée sur les serveurs de l'ensemble de test. Pour chacun de ces serveurs, nous avons calculé le taux d'utilisation u et interrogé l'API pour obtenir une estimation de puissance, qui a ensuite été comparée à la puissance moyenne réelle mesurée.

4. Pour PowerHeat : Le protocole est différent. La performance mesurée est celle des modèles propres aux serveurs de l'ensemble de test. Pour chacun des serveurs de l'ensemble, un modèle PowerHeat individuel a été entraîné en utilisant 80 % des mesures de température CPU et de puissance électrique propres au serveur. La performance a été calculée en testant ce modèle sur 20 % de mesures restantes de ce même serveur.

4.4 Résultats et discussions

L'ensemble des calculs a été effectué sur une seule machine équipée d'un processeur AMD Epyc 7642 de 48 cœurs et de 256 Go de mémoire. La collecte de données initiale a généré un total de 52.3 Mo de données brutes.

L'entraînement de l'ensemble des 1076 modèles PowerHeat s'est déroulé en 1 minute et 45 secondes (environ 2 heures et 45 minutes CPU). L'étape d'évaluation croisée, qui a impliqué l'exécution de 1.156.700 inférences (1075 modèles appliqués à 1076 serveurs), a nécessité 1 heure et 40 minutes (soit 160 heures CPU). L'entraînement du modèle LTR a duré 1 minute et 15 secondes (2 heures CPU). Enfin, l'entraînement du modèle de régression pour l'approche H-PACE a monopolisé 1 heure CPU.

L'intégralité du pipeline de traitement et de modélisation a été développée en Python [75], en s'appuyant sur les bibliothèques `pandas` [76] pour la manipulation des données, `xgboost` [69] pour l'entraînement des modèles individuels et du modèle LTR et `scikit-learn` [77] pour le modèle de l'approche H-PACE.

4.4.1 Performances globales

Pour évaluer les différentes approches, nous avons utilisé l'Erreur Absolue Moyenne en Pourcentage (MAPE). Cette métrique compare la puissance réellement consommée avec la puissance estimée par les modèles. Les résultats de cette comparaison sont présentés dans la Figure 4.4.

Les modèles PowerHeat atteignent une erreur moyenne très faible de 2,7 %. Le meilleur modèle descend à 0,1 % d'erreur, tandis que le pire atteint 23 %.

L'approche PPEM-BM obtient une moyenne de 11,6 % d'erreur. Elle surpasse légè-

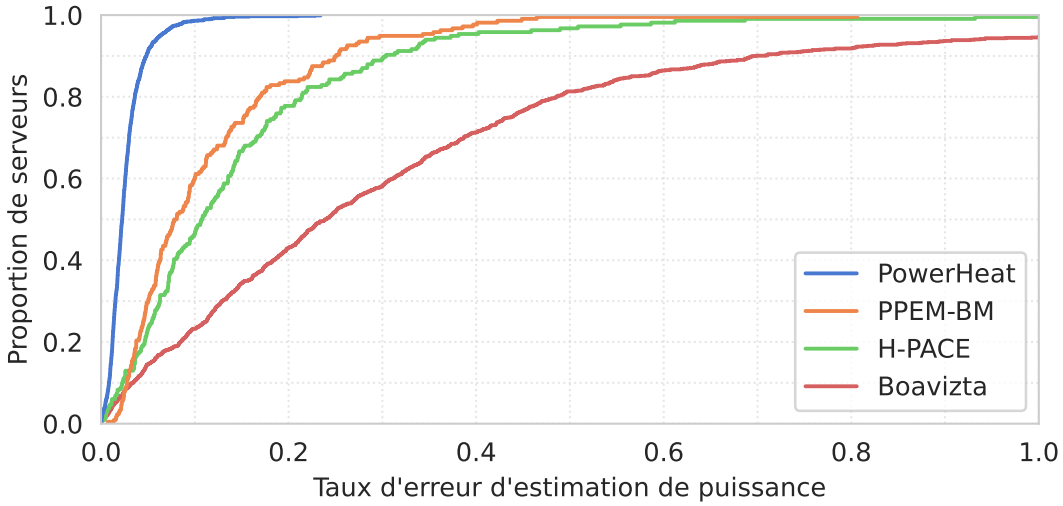


FIGURE 4.4 – Distribution des taux d'erreur d'estimation de puissance de chaque approche.

rement H-PACE (14,6 % d'erreur) et montre une amélioration notable par rapport à Boavizta (37,8 % d'erreur). En examinant les cas extrêmes, 1 serveur dépasse 50 % d'erreur avec PPEM-BM, contre 7 pour H-PACE et 201 pour Boavizta.

Nous constatons un écart important entre les performances de PPEM-BM (11,6 %) et le scénario idéal des modèles PowerHeat propres (2,7 %). Pour comprendre les raisons de cette différence, il est nécessaire d'analyser le comportement du modèle LTR, qui est au cœur de notre approche de sélection.

4.4.2 Analyse du modèle LTR

Pour chaque serveur de test, le modèle LTR évalue la pertinence de tous les modèles de puissance PowerHeat issus des 860 serveurs d'entraînement. Il attribue un score à chaque modèle : plus le score est élevé, plus le modèle est jugé pertinent. En triant les modèles par ce score de pertinence, nous leur associons un rang. Le modèle de rang 1 est considéré comme le plus pertinent, et celui de rang 860 comme le moins pertinent.

La Figure 4.5 analyse les performances des modèles en fonction de leur rang. Elle compare les résultats de l'évaluation croisée (où le rang est déterminé par l'inférence réelle de chaque modèle) avec les prédictions du modèle LTR.

Cette figure met en lumière deux points essentiels. Premièrement, les performances des modèles de puissance de rang 1 sont très bonnes, avec une moyenne d'erreur autour de 2,3 %. Ce résultat est équivalent aux performances des modèles PowerHeat entraînés spécifiquement pour ces serveurs. Deuxièmement, le modèle LTR parvient à suivre la tendance générale : plus le rang prédit augmente, plus l'erreur d'estimation s'élève. La dispersion des valeurs autour de la courbe reste contenue, indiquant un niveau d'incertitude

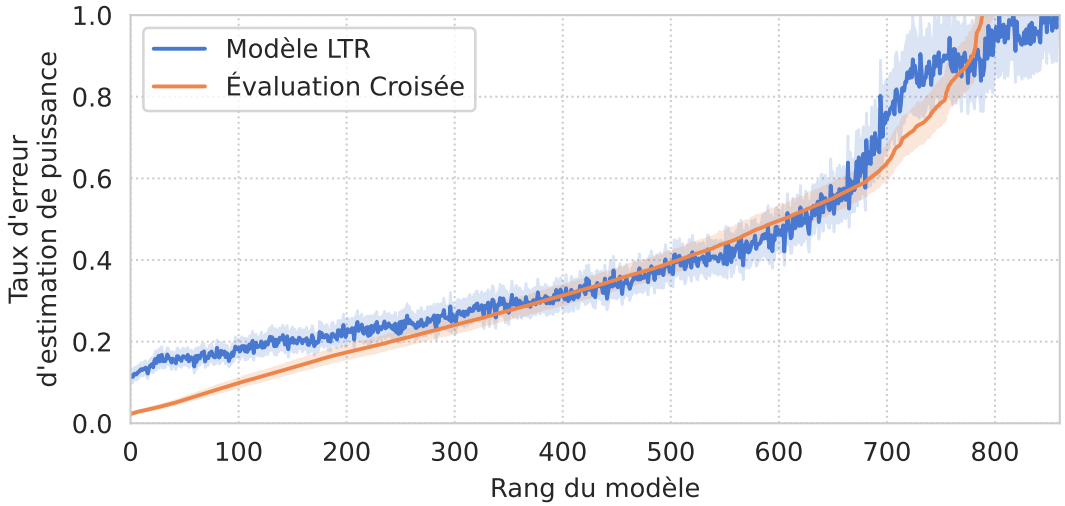


FIGURE 4.5 – Évolution du taux d'erreur en fonction du rang du modèle.

acceptable, particulièrement pour les modèles les mieux classés.

Cependant, un décalage important est visible pour les premiers rangs : le rang 1 prédit par le LTR ne correspond pas systématiquement au rang 1 de l'évaluation croisée. Ce décalage s'atténue autour du rang 400. Ce décalage initial est directement responsable de la sous-performance de l'approche globale.

En effet, l'implémentation de notre fonction de sélection est simple : elle choisit systématiquement le modèle prédit comme étant de rang 1 par le LTR. Par conséquent, les performances globales de l'approche PPEM-BM (11,6 %) correspondent à la moyenne des erreurs des modèles que le LTR a classés au premier rang.

Globalement, ces performances sont tout à fait satisfaisantes pour une première implémentation. L'évaluation croisée démontre que des modèles de puissance "tiers" (non spécifiques à un serveur) peuvent fournir d'excellentes estimations (2,3 % d'erreur en moyenne pour le rang 1 réel). De plus, l'utilisation d'un modèle LTR permet, dans une certaine mesure, de prédire l'ordre général de pertinence des modèles de la bibliothèque.

Ces résultats ouvrent deux perspectives d'amélioration :

— Une exploration des caractéristiques d'entraînement du modèle LTR (par exemple, la fonction de coût, les hyperparamètres) pourrait permettre de mieux coller à l'ordre de pertinence réel issu de l'évaluation croisée, en particulier pour les premiers rangs ;

— Une implémentation différente de la fonction de sélection pourrait permettre de choisir plus finement le modèle de puissance à utiliser, par exemple en ne se limitant pas au seul rang 1 prédit ou en combinant plusieurs modèles jugés pertinents.

4.4.3 Représentativité des résultats

L'objectif des travaux présentés dans ce chapitre est de concevoir une approche de modélisation énergétique destinée à être utilisée en production chez OVHcloud. Les 1076 serveurs utilisés dans cette expérience font partie d'une infrastructure de production comptant 132 011 serveurs.

Une contrainte majeure est que ces 1076 serveurs sont les seuls, au sein de cette vaste infrastructure, à disposer de capteurs matériels de puissance électrique. Tous les autres serveurs sont "aveugles". La question centrale est donc de savoir dans quelle mesure l'approche PPEM-BM, entraînée sur cet échantillon, peut être appliquée avec fiabilité à ces serveurs "aveugles".

Un aspect essentiel de l'approche PPEM-BM est la similarité matérielle, qui permet au LTR de trier les modèles de puissance par pertinence. La Figure 4.6 compare les distributions des caractéristiques matérielles des serveurs équipés de capteurs avec l'ensemble des autres serveurs de l'infrastructure.

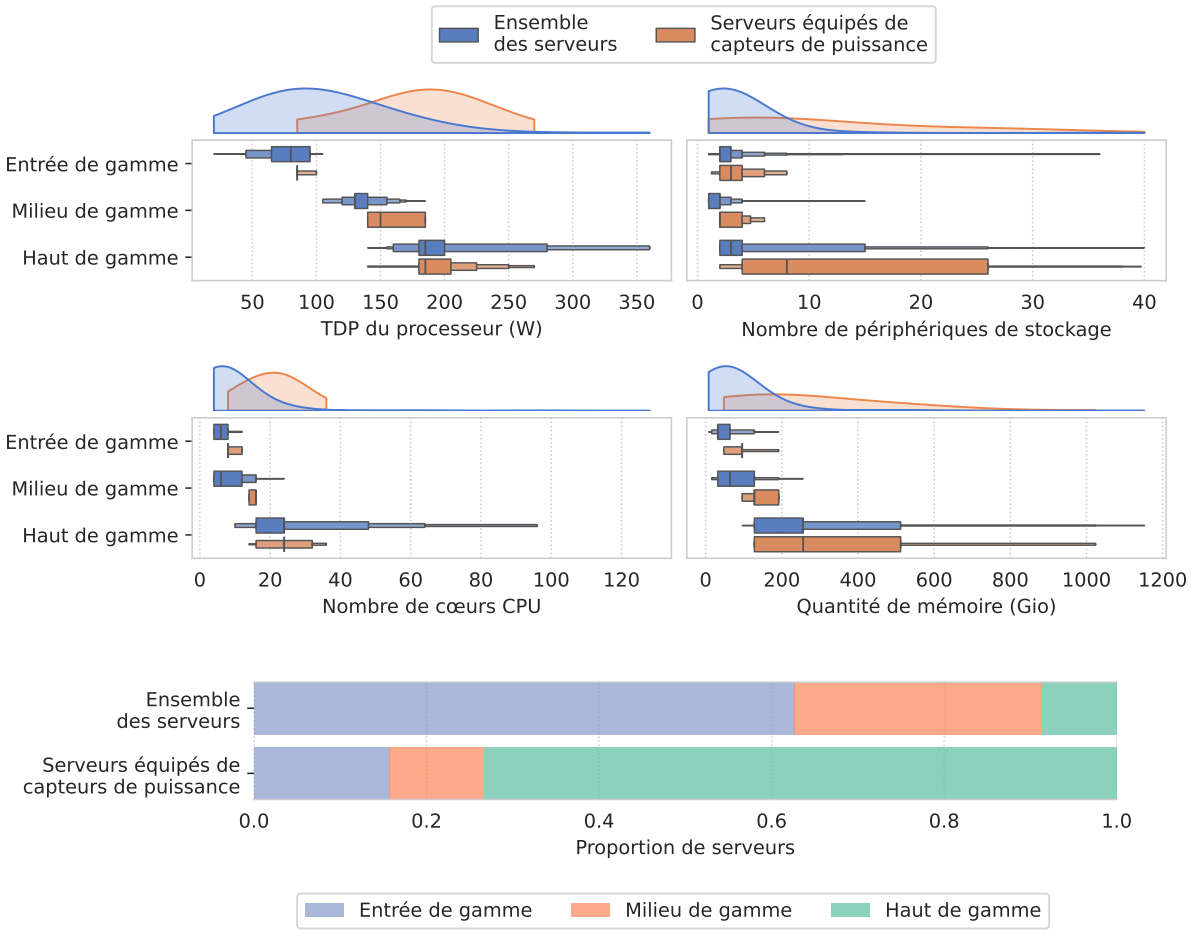


FIGURE 4.6 – Distributions des configurations matérielles des serveurs.

Nous avons découpé les configurations matérielles de l'ensemble des serveurs en trois

groupes : Bas de gamme, Milieu de gamme et Haut de gamme. Ce découpage a été réalisé de manière automatique en utilisant un algorithme de partitionnement de données (clustering non supervisé), spécifiquement l'algorithme K-moyennes (implémentation KMeans de la bibliothèque `scikit-learn` [77]), en se basant sur quatre caractéristiques matérielles (TDP, nombre de cœurs, quantité de mémoire, nombre de disques).

La Figure 4.6 révèle des distributions légèrement différentes entre les deux ensembles de serveurs. Plusieurs composants notables sont absents de notre échantillon de serveurs instrumentés :

— Les processeurs avec un TDP inférieur à 85 W ou supérieur à 270 W ;

— Les processeurs avec moins de 8 cœurs CPU ou plus de 36 ;

— Les serveurs avec moins de 48 Gio de RAM.

Ces composants absents ont pourtant une place importante dans l'ensemble de l'infrastructure. L'application de PPEM-BM sur ces serveurs "aveugles" pourrait donc générer des erreurs d'estimation importantes, puisqu'aucun modèle de puissance équivalent n'a été entraîné sur ce type de matériel. En l'état, il est probable que la consommation d'énergie de ces machines soit sur ou sous-estimée par le modèle LTR, qui sélectionnerait le modèle le moins inadapté.

Pour les composants présents dans les deux ensembles, les distributions montrent que certaines caractéristiques sont moins représentées (moins d'individus) dans notre échantillon. La répartition par gamme confirme ce biais : l'ensemble des serveurs est majoritairement représenté par des machines bas de gamme, à l'inverse des serveurs équipés de capteurs, qui tendent à être plus représentatifs des milieu de gamme et haut de gamme. Dans quelle mesure ce biais de représentativité impacte-t-il les performances de PPEM-BM ? La Figure 4.7 présente les performances de chaque approche, groupées par gamme de serveur.

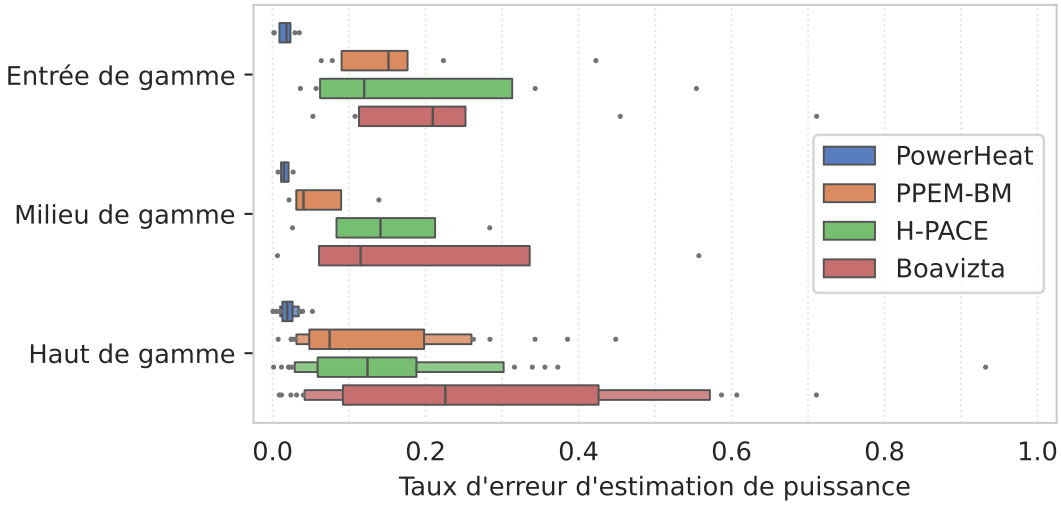


FIGURE 4.7 – Performances des approches en fonction de la gamme de serveurs.

Les performances des modèles PowerHeat sont constantes et peu dispersées quelle que soit la gamme. En revanche, la dispersion des erreurs est plus marquée pour les autres approches, en particulier pour Boavizta sur les serveurs haut de gamme. Concernant PPEM-BM, les performances pour les serveurs bas de gamme sont légèrement moins bonnes, avec une moyenne de 16,2 % d'erreur, contre 12,9 % pour les serveurs haut de gamme et un très bon 6,7 % pour les serveurs milieu de gamme.

Ces résultats montrent une certaine disparité des performances en fonction des gammes, mais ils ne confirment pas formellement que cela dépend du nombre d'individus présents dans chaque gamme de l'échantillon d'entraînement. Le milieu de gamme, le mieux estimé, n'est pas le plus représenté dans l'échantillon instrumenté.

Des travaux futurs pourraient s'intéresser à l'étude plus fine de l'impact de cette sous-représentativité matérielle. En particulier, ces travaux pourraient également se pencher sur la question suivante : sur quels serveurs "aveugles" devrions-nous installer des capteurs de puissance en priorité pour améliorer le plus efficacement les estimations globales des modèles et couvrir les "angles morts" matériels ?

4.5 Conclusion

Ce chapitre a abordé un défi majeur pour la gestion énergétique à grande échelle : comment estimer la consommation électrique de serveurs *bare metal* qui ne sont pas équipés de capteurs de puissance. La méthode PowerHeat, bien que précise, n'est pas "portable" car elle nécessite un entraînement spécifique pour chaque machine.

Nous avons introduit la méthodologie PPEM-BM, une approche pragmatique qui contourne cet obstacle. En constituant une bibliothèque de modèles de puissance issus d'un parc de serveurs instrumentés, et en utilisant un modèle d'apprentissage du classement (LTR), PPEM-BM apprend à sélectionner le modèle le plus pertinent pour un nouveau serveur "aveugle", en se basant uniquement sur son profil matériel et ses statistiques thermiques.

Les résultats de notre expérimentation sur 1076 serveurs de production sont encourageants. Avec un taux d'erreur moyen de 11,6 % , PPEM-BM surpasse les approches génériques comme Boavizta (37,8 %) et les modèles basés uniquement sur le matériel comme H-PACE (14,6 %). Cela démontre qu'il est possible de transférer une capacité d'estimation de puissance d'un parc de serveurs "sources" vers une flotte "aveugle".

Cependant, notre analyse révèle une marge de progression claire. L'erreur de 11,6 % est encore loin de la performance "idéale" d'un modèle PowerHeat dédié (2,7 %). Nous avons identifié que cet écart ne provient pas de la qualité des modèles dans la bibliothèque – le meilleur modèle disponible pour un serveur cible aurait offert une erreur de seulement 2,3 % – mais bien de la performance de la fonction de sélection LTR. Le modèle LTR parvient à classer correctement les modèles, mais il ne réussit pas toujours à identifier le

"Top-1" absolu. De plus, nous avons constaté que notre échantillon de serveurs instrumentés n'est pas parfaitement représentatif de l'ensemble de l'infrastructure, qui compte de nombreuses machines bas de gamme absentes de notre jeu de données d'entraînement.

Ces limites ouvrent des perspectives d'amélioration comme l'optimisation du LTR ou un enrichissement ciblé de la bibliothèque de modèles. PPEM-BM s'est avérée être une solution viable et efficace. Elle fournit une estimation de puissance robuste là où aucune mesure n'existait, en s'appuyant intelligemment sur les profils matériels et thermiques des serveurs.

Notre analyse s'est jusqu'à présent concentrée sur la consommation globale du serveur, fortement influencée par l'activité du processeur. Cependant, les centres de données modernes voient leur consommation énergétique de plus en plus dominée par un autre composant : le GPU. La demande croissante pour les charges de travail d'intelligence artificielle a rendu la compréhension de leur consommation essentielle.

Le défi de la mesure de puissance des GPU dans un contexte *cloud* est toutefois d'un autre ordre de complexité. Contrairement aux CPU dans un environnement *bare metal*, les GPU sont massivement partagés et virtualisés. Les outils de mesure standards ne sont souvent pas accessibles ou ne permettent pas de distinguer la consommation de plusieurs clients se partageant le même matériel.

Chapitre 5

Modélisation de la puissance des GPU dans le *cloud* : Le cas du *pass-through*

Sommaire

5.1 Introduction **61**
5.2 Les modes de partage des GPU **63**
5.2.1 Allocation temporelle 63
5.2.2 Allocation spatiale 64
5.2.3 Allocation *pass-through* 64
5.3 Inférence par capteurs de températures **65**
5.3.1 Acquisition de données via IPMI 65
5.3.2 Dispositifs expérimentaux 65
5.3.3 Sélection des capteurs pertinents 66
5.3.4 Modélisation en environnements refroidis par air 67
5.3.5 Modélisation en environnements refroidis par eau 69
5.4 Cas d'étude : analyse des GPU d'un cluster IaaS **70**
5.4.1 Contexte de l'expérimentation et méthodologie 70
5.4.2 Analyse du taux d'occupation réel 70
5.4.3 Limites des métriques classiques 71
5.5 Conclusion **73**

5.1 Introduction

Dans les chapitres précédents, nous avons présenté les méthodologies PowerHeat et PPEM-BM. Ces approches ont montré qu'il est possible d'estimer avec précision la consommation électrique de serveurs *bare metal* en utilisant la température du processeur comme

indicateur principal. Ces travaux ont été conçus pour des architectures de serveurs classiques (processeur, mémoire, stockage, réseau), où la consommation électrique fluctue majoritairement en fonction de l'intensité des calculs effectués par le processeur central.

Toutefois, ces dernières années ont marqué un tournant majeur dans l'écosystème des centres de données. L'essor de nouveaux usages, portés notamment par l'Intelligence Artificielle et l'apprentissage profond, a engendré une demande grandissante pour des ressources de calculs spécialisées : les accélérateurs graphiques (GPU).

Le profil énergétique des machines équipées de ces accélérateurs se démarque des serveurs classiques. La consommation du serveur n'est plus uniquement dictée par le CPU, mais elle est désormais partagée par ces coprocesseurs. Le GPU devient même parfois le consommateur principal d'énergie et l'intégration de sa consommation devient alors indispensable pour évaluer la consommation totale du serveur.

Les accélérateurs graphiques hauts de gamme, tels que les NVIDIA H100 [78] ou A100 [79], représentent un investissement massif, coûtant plusieurs dizaines de milliers d'euros par unité. Pour rentabiliser ces équipements, les opérateurs cherchent à maximiser leur taux d'occupation tout en évitant le gaspillage de ressources. Afin d'allouer la puissance de calcul nécessaire à chaque charge de travail, la mise à disposition des ressources GPU dans un environnement *cloud* s'opère selon différents modes d'allocation : partage temporel, partage spatial et mode *pass-through*.

Les travaux présentés dans ce chapitre sont issus de nos travaux de recherche dont l'article original [1] propose des analyses détaillées et des modèles énergétiques pour les trois modes de partage. Le présent chapitre se focalise sur l'allocation en *pass-through*, dont les problématiques liées à la non-intrusivité se rapprochent de celles des serveurs *bare metal* traitées précédemment.

En effet, si les modes partagés (temporel et spatial) permettent généralement au fournisseur de *cloud* de conserver une visibilité via le pilote (*driver*) installé sur l'hôte, le mode *pass-through* pose un problème de supervision.

En transférant le contrôle direct du périphérique à la VM du client, l'opérateur perd l'accès aux interfaces logicielles de monitoring usuelles. Du point de vue de l'opérateur *cloud*, le GPU devient une véritable "boîte noire". Les outils classiques comme `nvidia-smi` ou les API de gestion (NVML) ne sont plus interrogeables depuis l'hyperviseur car le périphérique est logiquement "détaché" du système hôte. L'opérateur ne dispose alors plus des métriques nécessaires pour mesurer l'énergie réelle consommée par le périphérique.

Face à cette perte de visibilité logicielle, nous proposons d'adapter la méthodologie PowerHeat au cas spécifique des GPU en mode *pass-through*.

Notre hypothèse est la suivante : les capteurs de température présents sur la carte mère

1. Pierre Jacquet, **Maxime Agusti**, Eddy Caron, Camille Coti, Marcos Dias De Assunção, Laurent Lefèvre, Anne-Cécile Orgerie, "*Untangling GPU Power Consumption : Job-Level Inference in Cloud Shared Settings*", EUROSYS 2026-European Conference on Computer Systems.

et directement sur GPU accessibles via le protocole IPMI peuvent servir de proxy fiable pour estimer la consommation électrique. Cette approche présente également l'avantage majeur d'être totalement non-intrusive : elle ne nécessite aucune installation d'agent dans la VM du client et respecte l'isolation stricte du mode *pass-through*.

Afin d'explorer notre hypothèse, ce chapitre est organisé de la manière suivante : la Section 5.2 présente les trois modes d'allocation (temporel, spatial, *pass-through*). La Section 5.3 détaille la méthodologie d'estimation dédiée au mode *pass-through*, centrée sur l'exploitation des données thermiques IPMI et présente la validation expérimentale des modèles sur des infrastructures refroidies par air et par eau. Enfin, la Section 5.4 propose une étude de cas concrète sur un cluster IaaS de production.

5.2 Les modes de partage des GPU

L'intégration des accélérateurs graphiques dans les offres de *Cloud Computing* s'appuie sur des mécanismes de partage visant à concilier les impératifs de performance, d'isolation et de rentabilité. La Figure 5.1 illustre les trois principaux paradigmes d'allocation observés dans l'industrie.

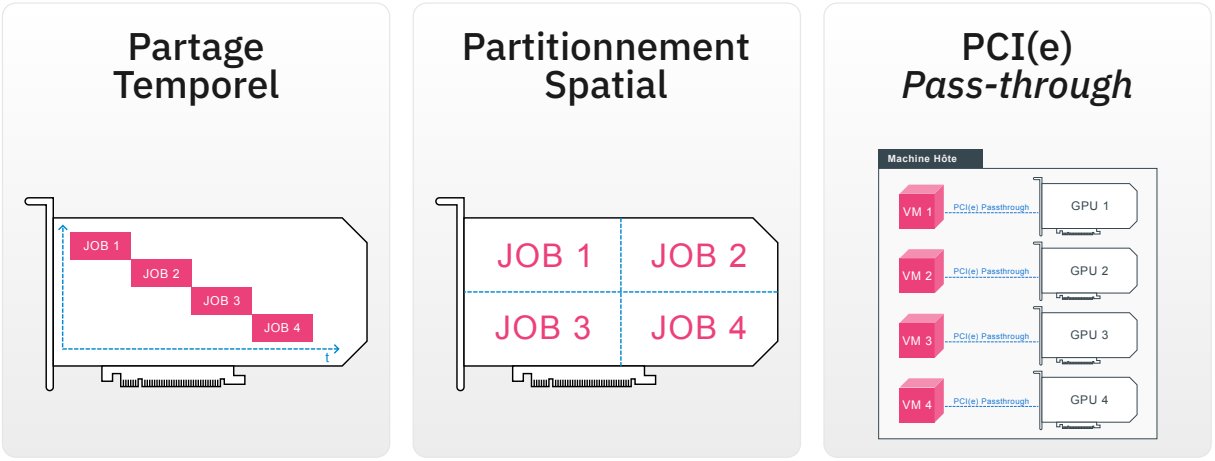


FIGURE 5.1 – Trois paradigmes d'allocation de GPU.

5.2.1 Allocation temporelle

L'allocation temporelle fonctionne sur un principe analogue à l'ordonnancement traditionnel des processus sur un cœur de CPU. Dans ce mode, le GPU est partagé séquentiellement entre plusieurs charges de travail.

Le mécanisme repose sur le changement de contexte. L'ordonnanceur (*scheduler*) du GPU alloue des tranches de temps d'exécution à chaque processus concurrent. À un instant t donné, les cœurs CUDA sont dédiés à un unique processus, mais l'opération de basculement entre les tâches génère une latence inhérente.

D'un point de vue technique, cette approche est couramment implémentée dans les environnements conteneurisés via des orchestrateurs comme Kubernetes. Le NVIDIA GPU Operator, par exemple, permet de définir des répliques logiques d'un même GPU physique, autorisant ainsi la soumission de plusieurs pods sur une seule carte. Ce mode est particulièrement adapté aux charges de travail sporadiques, aux environnements de développement ou aux services d'inférence légère - *Machine Learning as a Service* (MLaaS), où l'utilisation continue et exclusive de la puissance de calcul constituerait un gaspillage de ressources.

5.2.2 Allocation spatiale

Les architectures GPU modernes telles que les architectures Ampere et Hopper de NVIDIA ont introduit le partage spatial, commercialisé sous le nom de MIG.

Contrairement au découpage temporel, ce mode opère une division physique des ressources matérielles. Le GPU est segmenté en plusieurs instances totalement isolées, chacune disposant de ses propres multiprocesseurs de flux, de sa propre bande passante mémoire et de ses propres caches L2.

Cette approche garantit une isolation matérielle : l'exécution d'une tâche sur une partition n'impacte pas les performances des autres partitions. Ce déterminisme de la performance rend l'allocation spatiale idéale pour les environnements multi-locataires exigeants, où la sécurité et la stabilité des performances sont critiques, tout en permettant à l'opérateur de densifier l'usage de ses cartes haut de gamme.

5.2.3 Allocation *pass-through*

Le troisième mode, et celui qui nous intéresse particulièrement dans le cadre de ce chapitre, est l'allocation en *pass-through*. Il s'agit du mécanisme privilégié pour les offres IaaS destinées au calcul haute performance et à l'entraînement de modèles d'IA massifs.

Techniquement, ce mode repose sur les technologies de virtualisation des entrées-sorties assistées par matériel, telles que VT-d (Intel) ou AMD-Vi, et l'utilisation de l'*Input-Output Memory Management Unit* (IOMMU). Le contrôleur PCIe du GPU est détaché de l'hôte et assigné directement à l'espace d'adressage de la VM invitée via le module *Virtual Function I/O* (VFIO).

La VM accède ainsi au matériel avec des performances quasi-natives, sans la surcharge d'une couche de virtualisation intermédiaire. Cependant, cette architecture a une conséquence majeure pour l'opérateur : l'hyperviseur perd le contrôle et la visibilité sur le périphérique. Le pilote étant installé au sein du système invité (la VM du client), l'hôte ne peut plus interroger le GPU pour obtenir des métriques de télémétrie standard : taux d'utilisation, fréquence, consommation électrique. C'est cette spécificité architecturale qui transforme le GPU en "boîte noire" du point de vue de la supervision énergétique, justi-

fiant la nécessité d'une approche d'estimation par proxy thermique telle que développée dans la suite de ce chapitre.

5.3 Inférence par capteurs de températures

5.3.1 Acquisition de données via IPMI

Comme avec PowerHeat, notre hypothèse repose sur le principe physique de l'effet Joule : la puissance électrique dissipée par les circuits du GPU se transforme en chaleur, augmentant la température du composant et de son environnement immédiat. Bien que la relation entre puissance et température ne soit pas instantanée en raison de l'inertie thermique, nous postulons qu'elle est suffisamment corrélée pour servir de proxy fiable.

Pour valider cette hypothèse nous nous appuyons sur le protocole IPMI qui permet d'interroger le BMC de la carte mère indépendamment du système d'exploitation ou de l'état des machines virtuelles.

L'utilisation de l'IPMI présente un double avantage pour un environnement *cloud* multi-tenant :

— **Non-intrusivité totale :** Aucune installation d'agent logiciel n'est requise dans la VM du client. Le monitoring est invisible pour l'utilisateur final.

— **Indépendance logicielle :** La collecte de données reste fonctionnelle même en cas de dysfonctionnement de l'OS invité ou de saturation des ressources CPU de l'hôte.

5.3.2 Dispositifs expérimentaux

Pour évaluer notre approche nous l'avons expérimenté sur trois infrastructures différentes :

— **Banc d'essai (Grid'5000) - Air cooling - Densité Moyenne :** Un serveur HP Apollo 6500 Gen10 Plus équipé de 4 GPU NVIDIA A100.

— **Banc d'essai (Grid'5000) - Air cooling - Haute Densité :** Un serveur NVIDIA DGX équipé de 8 GPU NVIDIA A100.

— **Environnements de production (OVHcloud) - Water cooling :** Serveurs équipés de GPU NVIDIA A100 et H100.

Pour construire nos modèles, nous avons généré des charges de travail synthétiques contrôlées. Nous avons utilisé l'outil GPU-burn [80] pour stresser les cartes, couplé à la technologie MIG pour imposer des limites strictes de consommation. Cela nous a permis

de tester quatre niveaux de charge distincts (0 %, 30 %, 60 %, 100 %) maintenus chacun pendant 5 minutes, afin de capturer la dynamique thermique complète des serveurs.

5.3.3 Sélection des capteurs pertinents

Les serveurs modernes sont équipés de dizaines de capteurs de températures répartis sur la carte mère (CPU, RAM, admission d'air et sortie d'air, chipsets, etc.). Le défi consiste à identifier, parmi cette multitude de signaux, ceux qui capturent le plus fidèlement la signature thermique des GPU ciblés.

Pour opérer cette sélection, nous avons soumis des GPU aux charges de travail décrites précédemment, tout en enregistrant simultanément la puissance réelle consommée via `nvidia-smi` et les valeurs de tous les capteurs IPMI disponibles.

La Figure 5.2 présente la matrice de corrélation de Pearson obtenue lors de cette analyse.

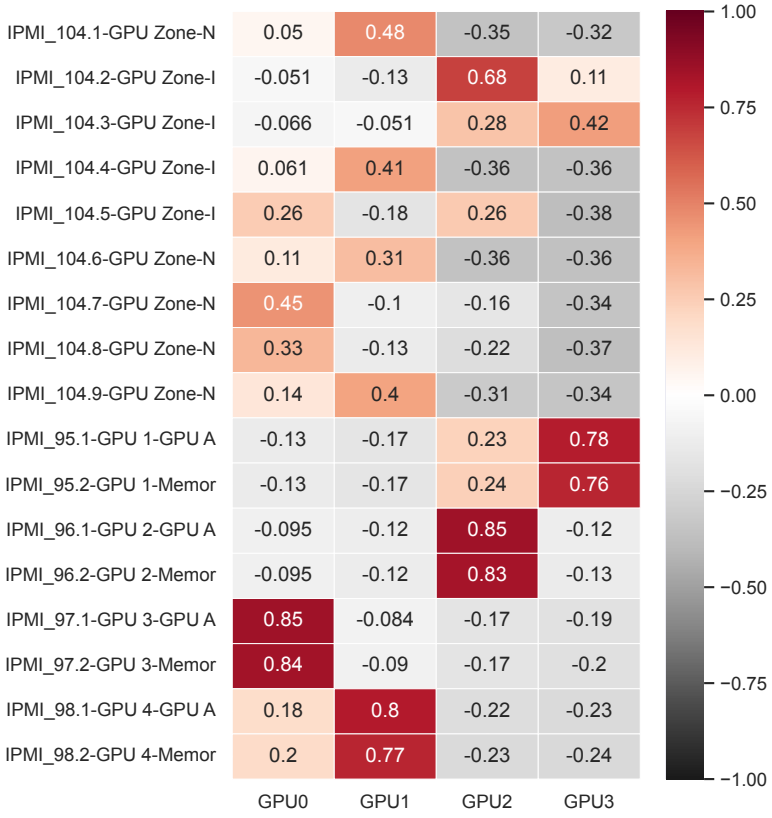


FIGURE 5.2 – Coefficient de corrélation de Pearson entre les capteurs IPMI et la puissance des GPU.

Les capteurs, identifiés en rouge vif sur la matrice, montrent une relation forte avec la consommation de puissance. Ils correspondent physiquement aux sondes situées à proximité immédiate des ports PCIe ou sur les régulateurs de tension alimentant les cartes. À

l'inverse, les autres capteurs ne réagissent que très faiblement aux variations de charge du GPU.

Sur la base de cette analyse, nous avons automatisé la sélection du "meilleur capteur candidat" pour chaque GPU : celui maximisant le coefficient de corrélation de Pearson avec la puissance réelle du périphérique.

5.3.4 Modélisation en environnements refroidis par air

L'utilisation de la température comme proxy énergétique se heurte à deux obstacles physiques majeurs, particulièrement exacerbés dans les serveurs haute densité refroidis par air :

— **L'inertie thermique :** La température met plusieurs secondes, voire minutes, à se stabiliser après un changement de charge (ex : passage de 0 W à 300 W).

— **L'interférence thermique :** Dans un châssis compact contenant 4 à 8 GPU, la chaleur dissipée par un accélérateur peut être captée par ses capteurs voisins, faussant leurs mesures respectives.

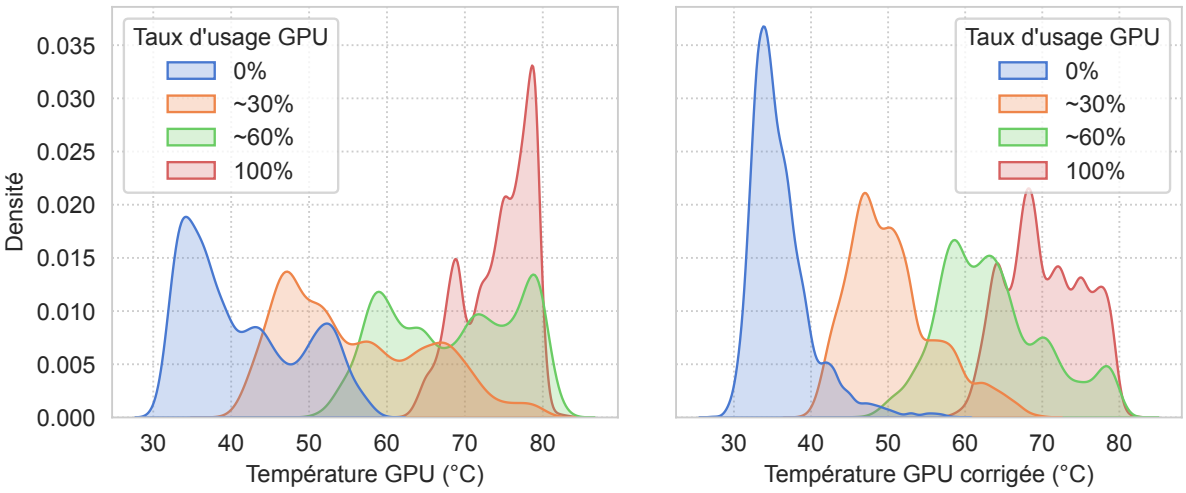


FIGURE 5.3 – Températures mesurées avant et après correction (4 × A100) en fonction du taux de charge.

Les Figures 5.3 et 5.4 représentent la fonction de densité (ou fonction de probabilité) de la température des GPU en fonction de la charge appliquée. L'axe de densité permet d'identifier les pics qui correspondent aux valeurs de température les plus fréquemment observées pour chaque niveau de charge.

Dans le cas spécifique du refroidissement par air, l'influence des serveurs voisins jouent un rôle significatif dans la dynamique des températures mesurées, entraînant une forte dispersion. Par exemple, lors de la mise en charge de quatre GPU A100 à 100 %, on

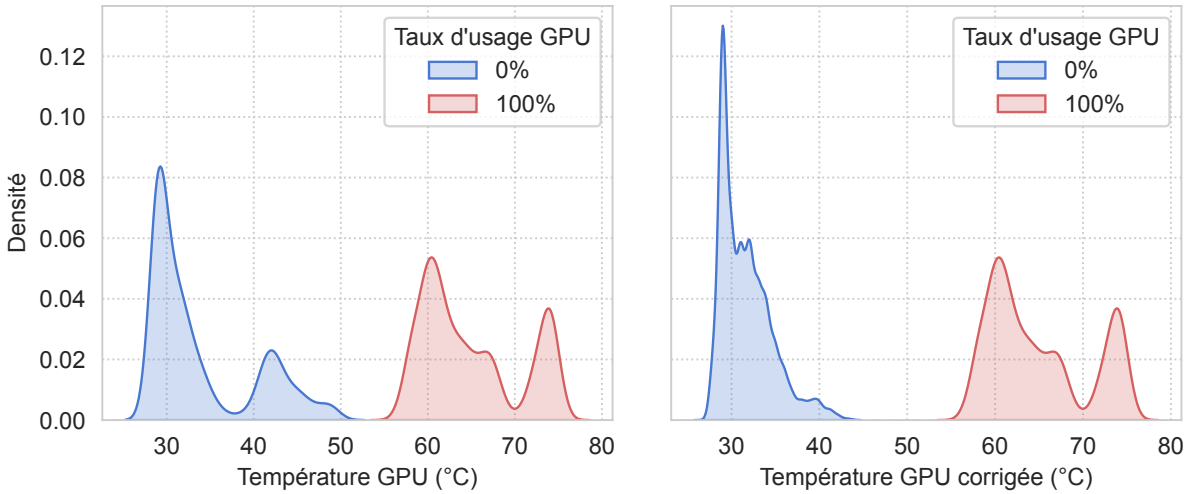


FIGURE 5.4 – Températures mesurées avant et après correction (8 × A100) en fonction du taux de charge.

observe des températures s'étalant sur une large plage allant d'environ 62 à 83 °C, comme l'illustre la partie gauche de la Figure 5.3.

Ce phénomène d'interférence explique le chevauchement significatif des plages de température entre l'état inactif (*idle*) et les états de charge partielle visible sur ces distributions brutes. Concrètement, un GPU inactif situé en aval d'un voisin fonctionnant à pleine puissance peut afficher une température de 50 °C, le rendant indiscernable d'un GPU actif.

Pour pallier ce biais, nous avons développé un algorithme de correction des interférences. La méthode procède en deux étapes :

1. **Identification du voisin influent :** Pour chaque GPU G_i, nous identifions le GPU voisin G_k dont la température est la plus corrélée à celle de G_i (hors autocorrélation).

2. **Correction par régression :** Nous appliquons une correction linéaire pour soustraire l'apport thermique du voisin, selon la formule 5.1.

$$T^{(i)}_{corrigee} = T^{(i)}_{mesuree} - \alpha \cdot (T^{(k)}_{mesuree} - T^{(k)}_{min}) \tag{5.1}$$

Où α est le coefficient d'influence déterminé lors de la phase de calibration.

Comme l'illustrent les graphiques de droite des Figures 5.3 et 5.4, l'application de cette correction par régression réduit significativement l'influence des voisins et permet de séparer distinctement les densités de probabilité. Après correction, il devient donc possible de mieux identifier le niveau de charge d'un GPU (0 %, 30 %, 60 %, 100 %) à partir de sa seule température, atteignant une précision de classification supérieure à 98 % pour l'état *idle*.

5.3.5 Modélisation en environnements refroidis par eau

Les centres de données modernes se tournent de plus en plus vers le refroidissement liquide de type *Direct-to-Chip Water Cooling* pour évacuer la chaleur des GPU de nouvelle génération dont le TDP peut dépasser les 400 W. Ce contexte modifie la dynamique thermique observée.

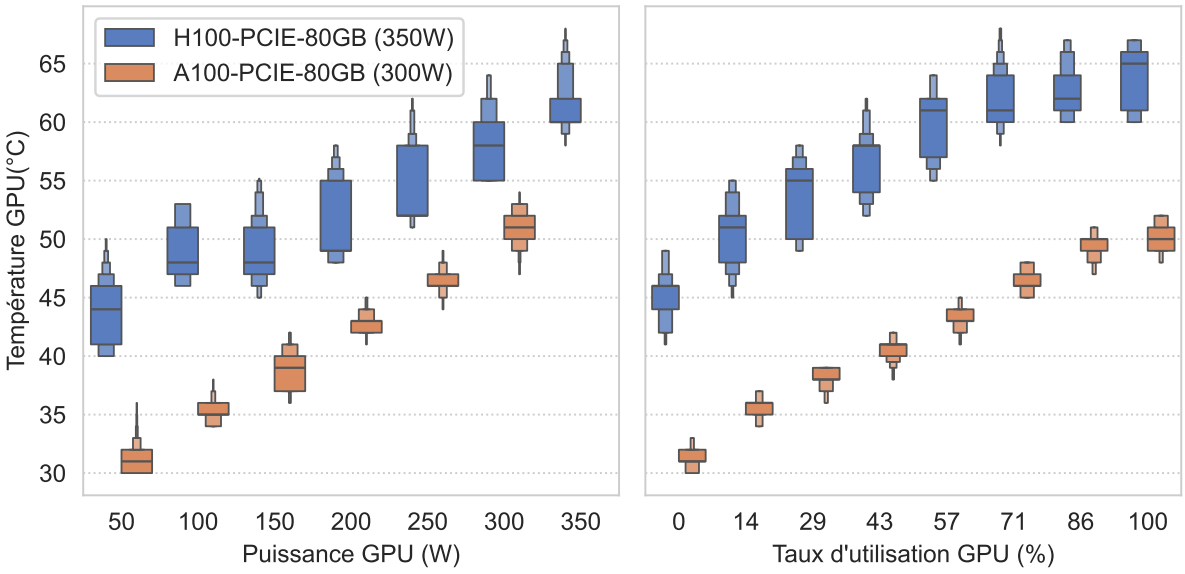


FIGURE 5.5 – Température d'un GPU refroidi à l'eau en fonction de la puissance et du taux d'utilisation.

Comme le montre la Figure 5.5, réalisée sur l'infrastructure de production d'OVHcloud, le refroidissement liquide maintient les composants dans une plage de température beaucoup plus restreinte et stable : entre 40 °C et 65 °C pour des H100 et 30 °C et 55 °C pour les A100.

Dans ce contexte, l'interférence entre voisins devient négligeable car l'eau capture la chaleur localement. Cependant, une nouvelle variable exogène apparaît : la température de l'eau en entrée, qui fluctue au cours de la journée et de la saison.

Pour adapter notre modèle, nous avons remplacé la correction spatiale par une correction temporelle de la ligne de base (*Baseline Correction*). Nous utilisons un algorithme de fenêtre glissante basé sur des opérations morphologiques d'érosion et de dilatation [81], et la bibliothèque python `pybaselines` [82], pour estimer et soustraire la température de base du circuit de refroidissement.

Une fois normalisée, la relation entre la température corrigée et la puissance consommée peut être modélisée par une régression polynomiale. Nos résultats expérimentaux démontrent une erreur moyenne (MAPE) de 5 % pour les GPU A100 et 13,3 % pour les H100, confirmant que la méthodologie est transposable aux environnements refroidis à l'eau.

5.4 Cas d'étude : analyse des GPU d'un cluster IaaS

Afin d'évaluer l'applicabilité industrielle de notre approche de modélisation thermique, nous avons appliqué le modèle développé dans la section précédente à un environnement de production réel. Cette étude de cas vise un double objectif : premièrement, valider que la température permet de lever le voile sur la consommation et l'activité des GPU dans un contexte "boîte noire", et deuxièmement, quantifier l'efficacité réelle de l'utilisation des ressources dans un centre de données commercial.

5.4.1 Contexte de l'expérimentation et méthodologie

L'expérimentation a été menée sur un cluster de production chez OVHcloud, composé de 176 GPU NVIDIA H100. Ces accélérateurs sont exploités dans le cadre d'une offre IaaS, où l'utilisateur final loue des VM avec un accès exclusif aux GPU via le mode *pass-through*.

Dans ce contexte, l'opérateur *cloud* ne dispose d'aucune visibilité logicielle sur la charge de travail exécutée par le client. L'infrastructure bénéficie d'un refroidissement liquide direct, similairement aux conditions de modélisation établies dans la Section 5.3.5.

Nous avons collecté les données de télémétrie sur une période de deux mois, couvrant un total de 865 VM distinctes. En moyenne, ces VM ont une durée de vie de 2 heures et 30 minutes et une allocation de 1,4 GPU (les configurations possibles étant de 1, 2 ou 4 GPU par VM).

Pour chaque instance, nous avons appliqué notre modèle d'inférence basé sur la température IPMI, préalablement corrigée par la méthode de fenêtre glissante décrite en Section 5.3.5. Cette température corrigée nous a ensuite permis d'en déduire le taux d'utilisation des cœurs de calcul, ainsi que d'estimer la puissance consommée. À notre connaissance, il s'agit de la première étude analysant l'usage réel d'un parc de GPU H100 en production IaaS sans intrusion logicielle.

5.4.2 Analyse du taux d'occupation réel

La Figure 5.6 présente la fonction de répartition cumulative de l'utilisation des GPU sur l'ensemble de la période. L'application de notre modèle sur ce vaste jeu de données révèle une sous-utilisation significative des ressources louées.

L'analyse de cette distribution met en évidence un profil d'usage très bas :

— 80 % des machines virtuelles présentent une utilisation médiane de leurs capacités de calcul GPU inférieure à 25 %.

— Même en considérant les pics d'activité (le 95ème centile de l'utilisation), 60 % des déploiements n'exploitent pas plus de 50 % de la puissance de calcul disponible.

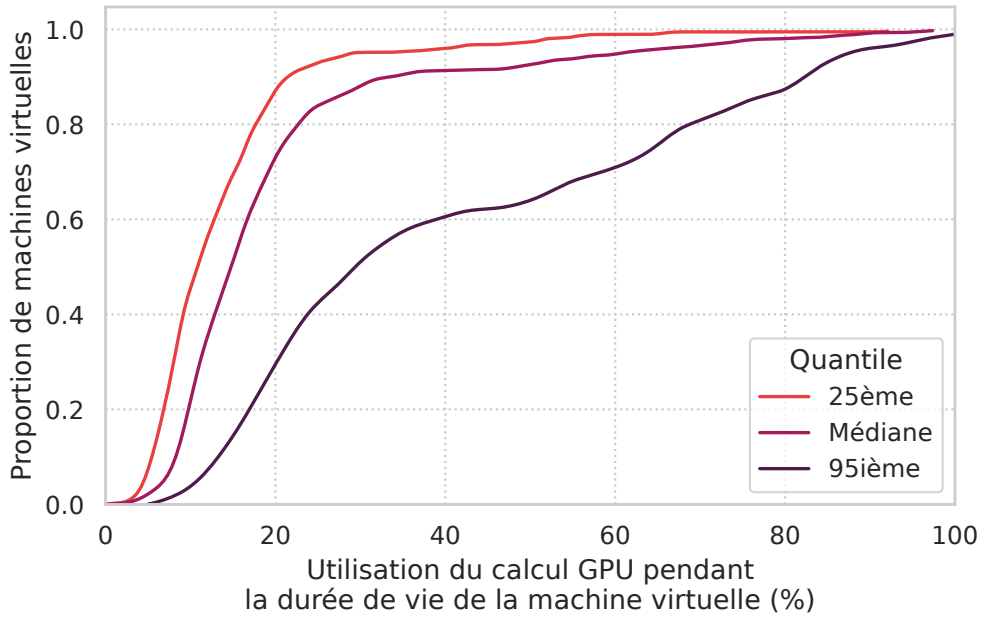


FIGURE 5.6 – Fonction de répartition de l'utilisation des GPU sur le cluster IaaS.

Plusieurs facteurs expliquent ce phénomène. D'une part, notre modèle estime l'utilisation réelle des cœurs de calcul, une métrique plus stricte que la définition temporelle de NVIDIA qui considère un GPU comme "utilisé" dès qu'un noyau CUDA est actif, même s'il est peu intensif. D'autre part, les charges de travail réelles comportent inévitablement des phases inactives (chargement de données, points de contrôle, goulots d'étranglement mémoire) qui ne saturent pas l'accélérateur.

Enfin, ces résultats suggèrent un surdimensionnement fréquent de la part des utilisateurs, qui réservent des cartes haut de gamme pour des tâches qui ne nécessitent pas leur pleine puissance continue. Ce constat valide empiriquement l'intérêt des modes de partage (temporel ou spatial) présentés en Section 5.2, qui permettraient de densifier ces usages sporadiques.

5.4.3 Limites des métriques classiques

Face à ce constat de sous-utilisation, une question se pose : l'opérateur pourrait-il détecter ces profils d'usage sans passer par une modélisation thermique complexe, en utilisant simplement les métriques déjà disponibles au niveau de l'hyperviseur ?

Pour y répondre, nous avons calculé la matrice de corrélation de Spearman entre l'usage GPU inféré par notre modèle et les métriques classiques des VM. Les résultats sont présentés dans la Figure 5.7.

L'analyse des corrélations montre qu'aucune métrique standard n'est un prédicteur fiable de l'activité GPU :

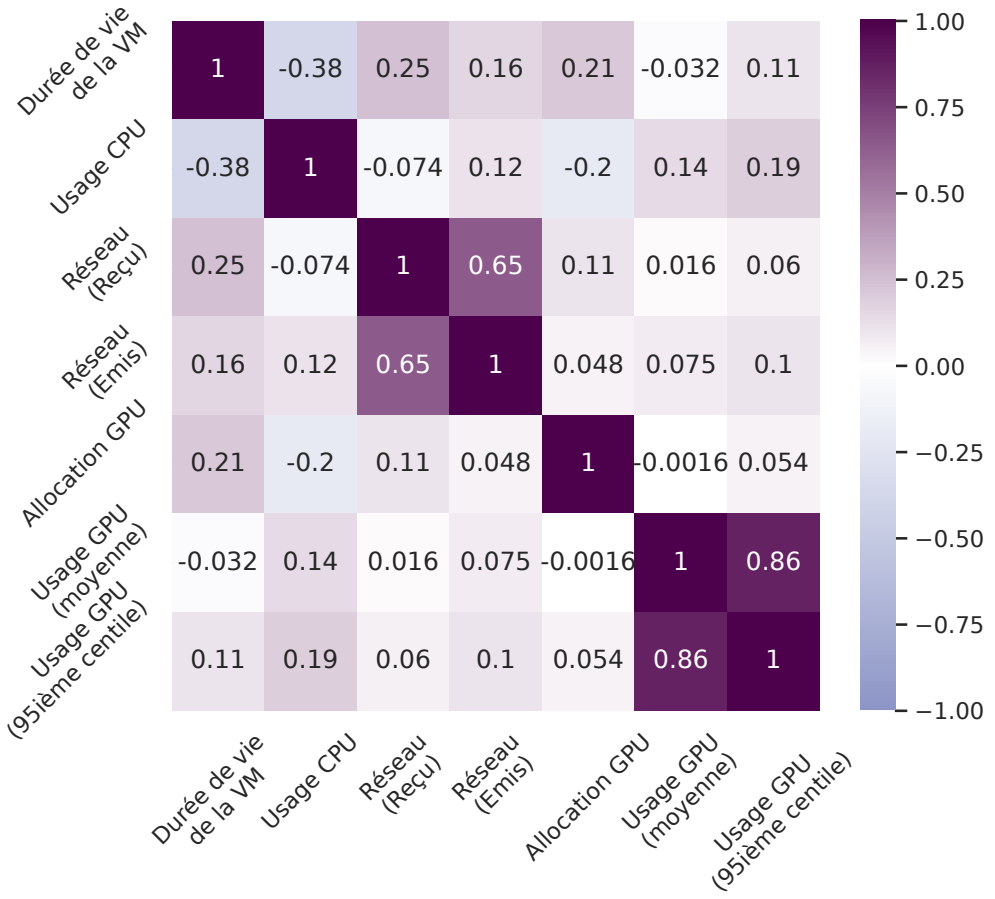


FIGURE 5.7 – Matrice de corrélation des métriques des machines virtuelles.

— **CPU vs GPU :** L'usage du CPU ne reflète pas fidèlement celui du GPU. Cela s'explique par le grand nombre de vCPU alloués aux VM diluant le signal et par l'asynchronisme des tâches (le CPU peut être en attente d'E/S pendant que le GPU calcule, et inversement).

— **Réseau vs GPU :** Bien que l'entraînement distribué génère du trafic, de nombreuses tâches d'inférence sont limitées par la mémoire ou le calcul, sans générer de trafic réseau continu corrélé à l'intensité du calcul.

— **Durée de vie vs Allocation :** Une très légère corrélation indique que les instances avec plus de GPU tendent à vivre plus longtemps, mais cela ne permet pas de déduire l'intensité d'usage.

L'absence de corrélation forte avec les métriques exogènes confirme que le GPU agit véritablement comme une "boîte noire" pour l'opérateur IaaS. L'approche proposée dans ce chapitre, utilisant la température comme proxy non-intrusif, s'avère donc être l'une des seules méthodes viables pour obtenir une visibilité sur la consommation énergétique et l'efficacité réelle des infrastructures accélérées en mode *pass-through*.

5.5 Conclusion

Les travaux présentés dans ce chapitre se sont attaqués à un verrou spécifique de la supervision énergétique dans le *cloud* : l'observabilité des accélérateurs graphiques en mode *pass-through*. Si ce mode d'allocation est privilégié dans les offres IaaS pour garantir des performances quasi-natives et une isolation stricte, il transforme le GPU en une "boîte noire" énergétique pour l'opérateur, le privant des métriques logicielles usuelles.

En réponse à cette perte de visibilité, nous proposons une méthodologie d'estimation non-intrusive, fondée sur l'exploitation des capteurs de température IPMI de la carte mère. Nos résultats montrent que la température, bien que sujette à l'inertie et aux interférences, constitue un proxy fiable de la consommation électrique, tant pour les infrastructures refroidies par air que par liquide. Notre approche repose sur deux mécanismes de correction : la régression spatiale pour neutraliser l'influence thermique des voisins en haute densité, et la correction de ligne de base par fenêtre glissante pour s'adapter aux fluctuations des circuits de refroidissement liquide.

L'application de ce modèle à un cluster de production de 176 GPU H100 a permis de révéler une réalité opérationnelle. L'analyse de 865 machines virtuelles a mis en évidence une sous-utilisation généralisée des ressources : 80 % des instances présentent un taux d'utilisation médian inférieur à 25 %. Ce gaspillage important, invisible via les métriques classiques (CPU, réseau) qui ne sont pas corrélées à l'activité GPU, confirme les limites du modèle d'allocation statique exclusif pour des charges de travail sporadiques ou mal dimensionnées.

Ces constats invitent à repenser le déploiement des ressources dans le *cloud*. En identifiant les cas de surdimensionnement, notre méthode ouvre la voie à une gestion plus souple des infrastructures. L'inefficacité du mode dédié justifie d'ailleurs l'adoption de mécanismes de partage. Bien que plus complexes à gérer, les modes de partage temporel et spatial deviennent alors indispensables pour mieux utiliser les serveurs et réduire l'impact carbone du calcul intensif.

Chapitre 6

Conclusion générale

La maîtrise de l'empreinte énergétique des centres de données est passée d'un enjeu économique d'optimisation des coûts à un impératif environnemental et réglementaire critique. Alors que la demande en calcul explose, portée par la transformation numérique et l'essor de l'intelligence artificielle, l'efficacité énergétique exige désormais une granularité fine, capable de descendre jusqu'au serveur et au composant.

Cependant, comme nous l'avons exposé en introduction, cette métrologie fine se heurte à un mur technique et économique dans les infrastructures existantes. Le déploiement massif de wattmètres physiques est prohibitif, et les modèles logiciels classiques sont rendus inopérants par les barrières d'isolation strictes du modèle *bare metal*.

Cette thèse a exploré une troisième voie : l'utilisation des canaux auxiliaires thermiques comme proxy non-intrusif de la puissance. Notre hypothèse centrale est que la trace thermique laissée par l'activité numérique, observable via les interfaces de gestion matérielle (IPMI), contient suffisamment d'information pour reconstituer la consommation électrique, sans jamais outre-passer la confidentialité du système client.

Nos travaux se sont articulés autour de trois contributions :

— Un proxy thermique en environnement refroidi par liquide : Dans le Chapitre 3, nous avons montré qu'il est possible d'estimer la consommation d'un serveur bare metal avec une bonne précision en utilisant la température du processeur. Nous avons spécifiquement adressé le défi posé par le water-cooling, dont l'efficacité thermique tend à "lisser" le signal utile. L'introduction d'une compensation thermique relative, basée sur un serveur témoin, a permis de décorréler les variations induites par la charge de calcul de celles exogènes liées à la boucle de refroidissement.

— Le passage à l'échelle par le transfert de modèles : Le Chapitre 4 a tenté de lever le verrou de la généralisation. Conscients que l'entraînement individuel des modèles est impossible sur un parc de plusieurs centaines de milliers de machines, nous avons conçu la méthodologie PPEM-BM. En combinant une bibliothèque de modèles de

référence et une approche d'apprentissage du classement (Learning to Rank), nous avons montré qu'il est possible d'assigner dynamiquement le modèle le plus pertinent à un serveur "aveugle". Cette méthode offre une solution viable pour déployer des "capteurs logiciels" massifs sans investissement matériel supplémentaire.

— L'extension aux accélérateurs opaques : Le Chapitre 5 a traité le cas critique des GPU utilisés pour l'IA. Face à l'opacité du mode pass-through, nous avons adapté nos modèles thermiques pour contourner les interférences spatiales (dans les châssis haute densité) et temporelles (inertie du refroidissement liquide). L'application de ces modèles sur un cluster de production a révélé une réalité opérationnelle marquante : une sous-utilisation massive des ressources, avec 80 % des machines virtuelles exploitant moins de 25 % des capacités de calcul des GPU H100 loués.

6.1 Limitations et discussions

Si ces travaux montrent la pertinence de l'approche thermique, ils soulignent également certaines limites physiques et méthodologiques.

La première limite est intrinsèque à la thermodynamique. L'inertie thermique agit comme un filtre passe-bas. Si nos modèles parviennent à estimer la consommation moyenne et les tendances énergétiques suffisantes pour la facturation ou le bilan carbone, ils peinent à capturer les pics de puissance brefs. Cette limitation restreint l'usage de nos méthodes pour des applications temps réel critiques, comme le *power capping* d'urgence pour la protection électrique, qui requièrent une réactivité à la milliseconde.

La seconde limite concerne la représentativité des données. La performance de la méthodologie PPEM-BM dépend directement de la richesse de la bibliothèque de modèles. Nos résultats ont montré que les serveurs "hors distribution" dont les configurations matérielles sont absentes du jeu d'entraînement subissent une dégradation de la qualité d'estimation. La diversité du parc matériel d'un fournisseur comme OVHcloud impose un effort constant de maintenance et d'enrichissement de cette bibliothèque.

6.2 Perspectives de recherches

Les contributions de cette thèse ouvrent la voie à des perspectives, allant de l'amélioration des modèles à leur exploitation pour une informatique plus durable.

Amélioration du Learning-to-Rank. L'analyse des résultats de PPEM-BM a montré que la fonction de sélection ne choisit pas toujours le modèle optimal absolu. Une piste consisterait à enrichir les caractéristiques d'entrée du LTR, non plus seulement avec des statistiques descriptives, mais avec des signatures thermiques spectrales (analyse fréquen-

tielle des variations de température) pour mieux caractériser la dynamique thermique du serveur cible.

Stratégies d'échantillonnage actif. Pour combler les lacunes de la bibliothèque de modèles, une stratégie d'instrumentation intelligente pourrait être déployée. Plutôt que d'équiper des serveurs au hasard, des algorithmes d'apprentissage actif pourraient identifier les configurations matérielles les plus incertaines et recommander le placement ciblé de PDU intelligents sur ces machines spécifiques, optimisant ainsi le ratio coût/information.

Modélisation physique de la boucle d'eau. Dans nos travaux, la température de l'eau a été traitée comme un bruit à soustraire. Une approche consisterait à modéliser explicitement la thermodynamique du centre de données. En couplant nos modèles de serveurs avec un jumeau numérique du système de refroidissement prenant en compte les débits, la position dans le rack, et les retours d'eau, il serait possible de transformer cette contrainte en une source d'information, améliorant la précision globale.

Approches hybrides. L'approche thermique pourrait être combinée avec d'autres canaux auxiliaires non-intrusifs. Par exemple, l'analyse du trafic réseau observable au niveau des switchs pourrait fournir des indices supplémentaires sur l'intensité de l'activité, permettant de clarifier certains états thermiques.

Vers une facturation incitative. La capacité d'estimer la consommation individuelle en bare metal rend possible le passage d'une facturation forfaitaire à une facturation à l'usage réel ou à l'empreinte carbone. Cette transition économique serait le levier le plus puissant pour inciter les clients à optimiser leur code ou à éteindre les machines inutilisées.

Pour conclure, ces travaux ont tenté de transformer la contrainte de l'opacité du bare metal en une opportunité de modélisation. En tirant parti des signaux thermiques, nous avons proposé une approche pour une observabilité énergétique globale et non-intrusive. Nous espérons que cette contribution méthodologique participera aux efforts nécessaires pour aligner le développement du *cloud* avec les enjeux environnementaux actuels.

Bibliographie

[1] Lawrence Berkeley National Laboratory, "2016 United States Data Center Energy Usage Report," 2016. Consulté le 2024-06-11.

[2] Lawrence Berkeley National Laboratory, "2024 United States Data Center Energy Usage Report," 2024. Consulté le 2024-06-11.

[3] ISO : the International Organization for Standardization, "ISO/IEC 30134-2 :2016/Amd 1 :2018." https://www.iso.org/standard/72702.html, 2018. Information technology — Data centres — Key performance indicators — Part 2 : Power usage effectiveness (PUE), [En ligne ; consulté le 21-juillet-2025].

[4] International Energy Agency, "Energy and AI," April 2025.

[5] E. Masanet, A. Shehabi, N. Lei, S. Smith, and J. Koomey, "Recalibrating global data center energy-use estimates," *Science*, vol. 367, no. 6481, pp. 984–986, 2020.

[6] L. M. Vaquero, L. Rodero-Merino, J. Caceres, and M. Lindner, "A break in the clouds : towards a cloud definition," *ACM sigcomm computer communication review*, vol. 39, no. 1, pp. 50–55, 2008.

[7] J. Koomey, S. Berard, M. Sanchez, and H. Wong, "Implications of historical trends in the electrical efficiency of computing," *IEEE Annals of the History of Computing*, vol. 33, no. 3, pp. 46–54, 2010.

[8] D. Donnellan, A. Lawrence, D. Bizo, P. Judge, J. O'Brien, J. Davis, M. Smolaks, J. Williams-George, and R. Weinschenk, "Uptime institute global data center survey results 2024," tech. rep., Uptime Institute, 2024.

[9] S. Lange, J. Pohl, and T. Santarius, "Digitalization and energy consumption. does ict reduce energy demand ?," *Ecological economics*, vol. 176, p. 106760, 2020.

[10] N. C. Thompson, K. Greenewald, K. Lee, G. F. Manso, *et al.*, "The computational limits of deep learning," *arXiv preprint arXiv :2007.05558*, vol. 10, p. 2, 2020.

[11] C. Morand, A.-L. Ligozat, and A. Névéol, "How Green Can AI Be ? A Study of Trends in Machine Learning Environmental Impacts," *arXiv preprint arXiv :2412.17376*, 2024.

[12] D. Donnellan, A. Lawrence, D. Bizo, O. Rogers, P. Judge, J. Davis, M. Smolaks, and R. Weinschenk, "Uptime institute global data center survey results 2025," tech. rep.,

Uptime Institute, 2025.

[13] J. O'Brien, "Pulling it power data with software," tech. rep., Uptime Institute, September 2024. Consulté le 15 septembre 2025.

[14] X. Fan, W.-D. Weber, and L. A. Barroso, "Power provisioning for a warehouse-sized computer," *ACM SIGARCH computer architecture news*, vol. 35, no. 2, pp. 13–23, 2007.

[15] C. Jin, X. Bai, C. Yang, W. Mao, and X. Xu, "A review of power consumption models of servers in data centers," *Applied Energy*, vol. 265, p. 114806, 2020.

[16] R. Mo, W. Lin, H. Zhong, M. Xu, and K. Li, "A cross-workload power prediction method based on transfer gaussian process regression in cloud data centers," *IEEE Transactions on Cloud Computing*, 2025.

[17] I. et OVHcloud, "Le Défi FrugalCloud : collaboration entre Inria et OVHcloud." https://graal.ens-lyon.fr/frugalcloud/index.html, 2021. Consulté le 2026-03-05.

[18] OVHcloud, "OVHcloud célèbre 20 ans de systèmes de refroidissement innovants et hautement efficaces au cœur de ses datacenters." https://corporate.ovhcloud.com/fr/newsroom/news/watercooling-anniversary/, 2023. Consulté le 2026-03-05.

[19] Parlement Européen, "Directive (UE) 2023/1791 du Parlement Européen et du Conseil du 13 septembre 2023 relative à l'efficacité énergétique et modifiant le règlement (UE)2023/955 (refonte)." https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=OJ%3AJOL_2023_231_R_0001&qid=1695186598766, Septembre 2023.

[20] Q. Wu, Q. Deng, L. Ganesh, C.-H. Hsu, Y. Jin, S. Kumar, B. Li, J. Meza, and Y. J. Song, "Dynamo : Facebook's data center-wide power management system," *ACM SIGARCH Computer Architecture News*, vol. 44, no. 3, pp. 469–480, 2016.

[21] M. A. Islam, S. Ren, and A. Wierman, "Exploiting a thermal side channel for power attacks in multi-tenant data centers," in *ACM SIGSAC Conference on Computer and Communications Security*, pp. 1079–1094, 2017.

[22] A. Kansal, F. Zhao, J. Liu, N. Kothari, and A. A. Bhattacharya, "Virtual machine power metering and provisioning," in *ACM symposium on Cloud computing*, pp. 39–50, 2010.

[23] C. Gu, H. Huang, and X. Jia, "Power metering for virtual machine in cloud computing-challenges and opportunities," *IEEE Access*, vol. 2, pp. 1106–1116, 2014.

[24] M. Colmant, M. Kurpicz, P. Felber, L. Huertas, R. Rouvoy, and A. Sobe, "Process-level power estimation in vm-based systems," in *European Conference on Computer Systems*, pp. 1–14, 2015.

[25] N. Jones *et al.*, "How to stop data centres from gobbling up the world's electricity," *nature*, vol. 561, no. 7722, pp. 163–166, 2018.

[26] Y. Wang, D. Nörtershäuser, S. Le Masson, and J.-M. Menaud, "An empirical study of power characterization approaches for servers," in *ENERGY 2019-The Ninth Inter-*

national Conference on Smart Grids, Green Communications and IT Energy-aware Technologies, pp. 1–6, 2019.

[27] W. Lin, W. Wu, H. Wang, J. Z. Wang, and C.-H. Hsu, "Experimental and quantitative analysis of server power model for cloud data centers," *Future Generation Computer Systems*, vol. 86, pp. 940–950, 2018.

[28] M. Jay, V. Ostapenco, L. Lefèvre, D. Trystram, A.-C. Orgerie, and B. Fichel, "An experimental comparison of software-based power meters : focus on CPU and GPU," in *2023 IEEE/ACM 23rd International Symposium on Cluster, Cloud and Internet Computing (CCGrid)*, pp. 106–118, IEEE, 2023.

[29] M. Dayarathna, Y. Wen, and R. Fan, "Data center energy consumption modeling : A survey," *IEEE Communications surveys & tutorials*, vol. 18, no. 1, pp. 732–794, 2015.

[30] M. S. Inukonda, A. R. Karpate, B. R. Tamma, S. Mittal, and P. Tammana, "Nascent : A non-invasive solution for detecting utilization of servers in bare-metal cloud," *IEEE Access*, vol. 10, pp. 12866–12881, 2022.

[31] E. Rotem, A. Naveh, A. Ananthakrishnan, E. Weissmann, and D. Rajwan, "Power-management architecture of the intel microarchitecture code-named sandy bridge," *IEEE Micro*, vol. 32, no. 2, pp. 20–27, 2012.

[32] G. Raffin and D. Trystram, "Dissecting the software-based measurement of CPU energy consumption : a comparative analysis," *IEEE Transactions on Parallel and Distributed Systems*, 2024.

[33] M. Weiser, B. Welch, A. Demers, and S. Shenker, "Scheduling for reduced CPU energy," in *Mobile Computing*, pp. 449–471, Springer, 1994.

[34] K. Chen, P. Kilpatrick, D. S. Nikolopoulos, and B. Varghese, "Cross architectural power modelling," in *2020 20th IEEE/ACM International Symposium on Cluster, Cloud and Internet Computing (CCGRID)*, pp. 390–399, IEEE, 2020.

[35] S. Choochotkaew, C. Wang, H. Chen, T. Chiba, M. Amaral, E. K. Lee, and T. Eilam, "A robust power model training framework for cloud native runtime energy metric exporter," *arXiv preprint arXiv :2407.00878*, 2024.

[36] R. A. Bridges, N. Imam, and T. M. Mintz, "Understanding GPU power : A survey of profiling, modeling, and simulation methods," *ACM Computing Surveys (CSUR)*, vol. 49, no. 3, pp. 1–27, 2016.

[37] D. Economou, S. Rivoire, C. Kozyrakis, and P. Ranganathan, "Full-system power analysis and modeling for server environments," in *Workshop on Modeling, Benchmarking, and Simulation (MoBS), held at the International Symposium on Computer Architecture*, 2006.

[38] R. V. White, "Pmbus : A decade of growth : An open-standards success," *IEEE Power Electronics Magazine*, vol. 1, no. 3, pp. 33–39, 2014.

[39] N. Lei, A. Shehabi, J. Lu, Z. Cao, J. Koomey, S. Smith, and E. Masanet, "Generalizable machine learning models for predicting data center server power, efficiency, and throughput," *arXiv preprint arXiv :2503.06439*, 2025.

[40] L. Ismail and H. Materwala, "Computing server power modeling in a data center : Survey, taxonomy, and performance evaluation," *ACM Computing Surveys (CSUR)*, vol. 53, no. 3, pp. 1–34, 2020.

[41] M. A. Islam, L. Yang, K. Ranganath, and S. Ren, "Why some like it loud : Timing power attacks in multi-tenant data centers using an acoustic side channel," *ACM on Measurement and Analysis of Computing Systems*, vol. 2, no. 1, pp. 1–33, 2018.

[42] M. A. Islam and S. Ren, "Ohm's law in data centers : A voltage side channel for timing power attacks," in *ACM SIGSAC Conference on Computer and Communications Security*, pp. 146–162, 2018.

[43] A.-C. Orgerie, M. D. d. Assuncao, and L. Lefevre, "A survey on techniques for improving the energy efficiency of large-scale distributed systems," *ACM Computing Surveys (CSUR)*, vol. 46, no. 4, pp. 1–31, 2014.

[44] M. Rezaei-Mayahi, M. Rezazad, and H. Sarbazi-Azad, "Temperature-aware power consumption modeling in hyperscale cloud data centers," *Future Generation Computer Systems*, vol. 94, pp. 130–139, 2019.

[45] C. Jin, X. Bai, X. Zhang, X. Xu, Y. Tang, and C. Zeng, "A measurement-based power consumption model of a server by considering inlet air temperature," *Energy*, vol. 261, p. 125126, 2022.

[46] S. Bhagavatula, *Fine-grained energy and thermal management using real-time power sensors*. PhD thesis, Purdue University, 2015.

[47] M. Zapater, J. L. Ayala, J. M. Moya, K. Vaidyanathan, K. Gross, and A. K. Coskun, "Leakage and temperature aware server control for improving energy efficiency in data centers," in *2013 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pp. 266–269, IEEE, 2013.

[48] M. Hnayno, A. Chehade, H. Klaba, H. Bauduin, G. Polidori, and C. Maalouf, "Performance analysis of new liquid cooling topology and its impact on data centres," *Applied Thermal Engineering*, vol. 213, p. 118733, 2022.

[49] R. Kong, H. Zhang, M. Tang, H. Zou, C. Tian, and T. Ding, "Enhancing data center cooling efficiency and ability : a comprehensive review of direct liquid cooling technologies," *Energy*, vol. 308, p. 132846, 2024.

[50] I. Latif, M. A. Shafique, H. Ullah, A. C. Newkirk, X. Yu, and A. Munir, "Cooling Matters : Benchmarking Large Language Models and Vision-Language Models on Liquid-Cooled Versus Air-Cooled H100 GPU Systems," *arXiv preprint arXiv :2507.16781*, 2025.

[51] V. Gohil, S. Dev, G. Upasani, D. Lo, P. Ranganathan, and C. Delimitrou, "The importance of generalizability in machine learning for systems," *IEEE Computer Architecture Letters*, vol. 23, no. 1, pp. 95–98, 2024.

[52] P. Garraghan, Y. Al-Anii, J. Summers, H. Thompson, N. Kapur, and K. Djemame, "A unified model for holistic power usage in cloud datacenter servers," in *International Conference on Utility and Cloud Computing*, pp. 11–19, 2016.

[53] J. C. McCullough, Y. Agarwal, J. Chandrashekar, S. Kuppuswamy, A. C. Snoeren, R. K. Gupta, *et al.*, "Evaluating the effectiveness of model-based power characterization," in *USENIX Annual Technical Conf*, vol. 20, pp. 19–20, 2011.

[54] R. Mo, W. Lin, H. Zhong, M. Xu, and K. Li, "A cross-workload power prediction method based on transfer gaussian process regression in cloud data centers," *IEEE Transactions on Cloud Computing*, 2025.

[55] F. Iorio, A. B. Hashemi, M. Tao, and C. Amza, "Transfer learning for cross-model regression in performance modeling for the cloud," in *2019 IEEE International Conference on Cloud Computing Technology and Science (CloudCom)*, pp. 9–18, IEEE, 2019.

[56] N. Bernard, H. Nguyen, A. Chandan, S. Jagdeeshan, N. Prabhugaonkar, R. Shah, and H. Jeon, "Hydra : Hybrid server power model," *arXiv preprint arXiv :2207.10217*, 2022.

[57] A. S. Luccioni, S. Viguier, and A.-L. Ligozat, "Estimating the carbon footprint of bloom, a 176b parameter language model," *Journal of machine learning research*, vol. 24, no. 253, pp. 1–15, 2023.

[58] S. Go, J. Park, S. More, H. Wu, I. Wang, A. Jezghani, T. Krishna, and D. Mahajan, "Characterizing the efficiency of distributed training : A power, performance, and thermal perspective," in *IEEE/ACM International Symposium on Microarchitecture*, pp. 626–642, 2025.

[59] S. Falk, D. Ekchajzer, T. Pirson, E. Lees-Perasso, A. Wattiez, L. Biber-Freudenberger, S. Luccioni, and A. van Wynsberghe, "More than Carbon : Cradle-to-Grave environmental impacts of GenAI training on the Nvidia A100 GPU," *arXiv preprint arXiv :2509.00093*, 2025.

[60] L. F. W. Anthony, B. Kanding, and R. Selvan, "Carbontracker : Tracking and predicting the carbon footprint of training deep learning models," *arXiv preprint arXiv :2007.03051*, 2020.

[61] R. Das and S. Rao, "Resource aware power modelling for compute-bound tasks on gpus," in *High Performance Computing and Cluster Technologies Conference*, pp. 1–10, 2024.

[62] K. Kasichayanula, D. Terpstra, P. Luszczek, S. Tomov, S. Moore, and G. D. Peterson, "Power aware computing on gpus," in *2012 Symposium on Application Accelerators in High Performance Computing*, pp. 64–73, IEEE, 2012.

[63] T. Vamja, K. Ray, F. George, and U. C. Devi, "On the Partitioning of GPU Power among Multi-Instances," *arXiv preprint arXiv :2501.17752*, 2025.

[64] S. Luccioni, Y. Jernite, and E. Strubell, "Power hungry processing : Watts driving the cost of ai deployment ?," in *ACM conference on fairness, accountability, and transparency*, pp. 85–99, 2024.

[65] J. Leng, T. Hetherington, A. ElTantawy, S. Gilani, N. S. Kim, T. M. Aamodt, and V. J. Reddi, "GPUWattch : Enabling energy optimizations in GPGPUs," *ACM SIGARCH computer architecture news*, vol. 41, no. 3, pp. 487–498, 2013.

[66] M. Khairy, Z. Shen, T. M. Aamodt, and T. G. Rogers, "Accel-sim : An extensible simulation framework for validated GPU modeling," in *2020 ACM/IEEE 47th Annual International Symposium on Computer Architecture (ISCA)*, pp. 473–486, IEEE, 2020.

[67] P. S. Gupta, Z. Talukder, T. A. Abir, P. Nguyen, and M. A. Islam, "Enabling low-cost server-level power monitoring in data centers using conducted emi," in *ACM Conference on Embedded Networked Sensor Systems*, pp. 237–250, 2023.

[68] F. H. Khan, T. D. Geist, B. Vairamohan, B. D. Fortenbery, and E. Hubbard, "Challenges and solutions in measuring computer power supply efficiency for 80 plus® certification," in *2009 Twenty-Fourth Annual IEEE Applied Power Electronics Conference and Exposition*, pp. 2079–2085, IEEE, 2009.

[69] T. Chen and C. Guestrin, "Xgboost : A scalable tree boosting system," in *ACM SIGKDD international conference on knowledge discovery and data mining*, pp. 785–794, 2016.

[70] J. Bergstra, R. Bardenet, Y. Bengio, and B. Kégl, "Algorithms for hyper-parameter optimization," *Advances in neural information processing systems*, vol. 24, 2011.

[71] P. Refaeilzadeh, L. Tang, and H. Liu, "Cross-validation," in *Encyclopedia of database systems*, pp. 532–538, Springer, 2009.

[72] F. C. Heinrich, T. Cornebize, A. Degomme, A. Legrand, A. Carpen-Amarie, S. Hunold, A.-C. Orgerie, and M. Quinson, "Predicting the energy-consumption of mpi applications at scale using only a single node," in *2017 IEEE international conference on cluster computing (CLUSTER)*, pp. 92–102, IEEE, 2017.

[73] T.-Y. Liu, "Learning to rank for information retrieval," *Foundations and Trends in Information Retrieval*, vol. 3, no. 3, pp. 225–331, 2009.

[74] T. Simon, D. Ekchajzer, A. Berthelot, E. Fourboul, S. Rince, and R. Rouvoy, "BoaviztAPI : a bottom-up model to assess the environmental impacts of cloud services," in *HotCarbon'24*, 2024.

[75] Python Software Foundation, *Python Language Reference, Version 3.12*, 2024.

[76] T. pandas development team, "pandas-dev/pandas : Pandas," Feb. 2020.

[77] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn : Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[78] NVIDIA, "GPU H100 Tensor Core." https://www.nvidia.com/fr-fr/data-center/h100/. Consulté le 2026-02-16.

[79] NVIDIA, "GPU NVIDIA A100 Tensor Core." https://www.nvidia.com/fr-fr/data-center/a100/. Consulté le 2026-02-16.

[80] V. Timonen, "Multi-GPU CUDA stress test." http://wili.cc/blog/gpu-burn.html, 2020. Consulté le 2026-02-16.

[81] Y. Chen and L. Dai, "An automated baseline correction method based on iterative morphological operations," *Applied spectroscopy*, vol. 72, no. 5, pp. 731–739, 2018.

[82] D. Erb, “pybaselines : A Python library of algorithms for the baseline correction of experimental data.” https://github.com/derb12/pybaselines. Consulté le 2026-02-16.