Projet de Fin d'Étude

TELECOM SudParis

IP PARIS

Analyse de Performance de TensorFlow

Présenté par:
Fatma Bouzghaia
Mehdi Messaoudi

Encadré par:
François Trahay

TABLE DES MATIÈRES

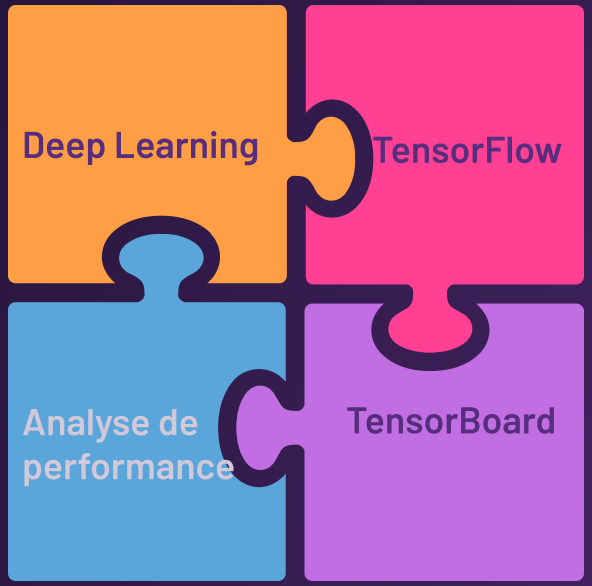
INTRODUCTION

Commençons par une introduction

Introduction

Domaine d'intelligence artificielle permet aux ordinateurs d'apprendre avec l'expérience

Observations systématiques pour améliorer les performances et la prise de décision



Framework qui permet de construire une architecture d'apprentissage en profondeur à grande échelle

Faire le profilage des modèles TensorFlow

OBJECTIF DU PROJET

Repérons notre cible!

But du projet ASR

- Étudier les outils d'analyse de performants existants.
- Évaluer l'impact de plusieurs optimisations sur des applications TensorFlow.

TECHNOLOGIES UTILISÉES

Parlons technologies!

TensorFlow

- Framework qui permet de faire de l'IA et de déployer des calculs sur GPU.
- Basé sur le calcul de graphes.
- Visualisation de la construction du réseau neuronal avec Tensorboard.

Les outils d’analyse de performance

TensorBoard

- Outil de profilage des modèles TensorFlow.
- Procure des informations concernant l'utilisation des ressources coûteuses.

NVIDIA Profiler

- Fournit des informations de profilage pour les applications CUDA.
- NVPROF: un outil de ligne de commande.
- NVVP: une interface visuelle.

RÉALISATIONS ET RÉSULTATS

Passons aux choses sérieuses

Étapes de la réalisation

Chercher des modèles

- Chercher des modèles de TensorFlow.
- Exécuter les applications sans optimisations.
- Évaluer l'application avec TensorBoard.

Appliquer les optimisations

- Optimiser les applications.
- Tracer les graphes et inspecter le profil donné par TensorBoard.

Interpréter les résultats

- Interpréter les tracés des temps d'exécution.
- Évaluer le profil donné et l'interpréter.

Nos applications

Une application ResNet-50 composée de 50 couches de profondeur.

Une application CNN composée de:

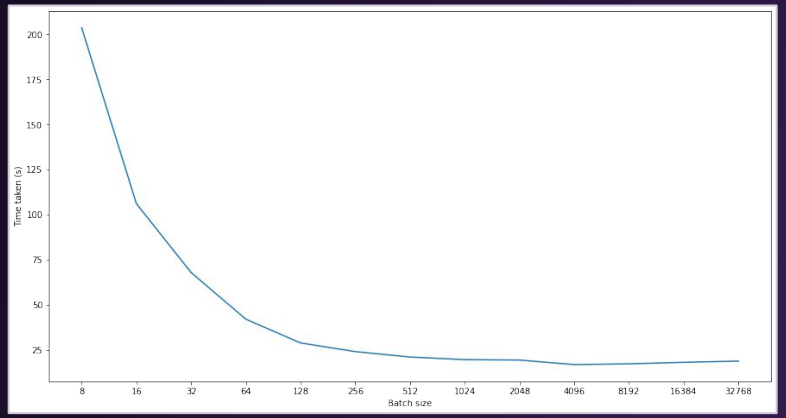
- 3 couches convolutives
- 2 couches de pooling
- 2 couches denses

La base de données CIFAR-10

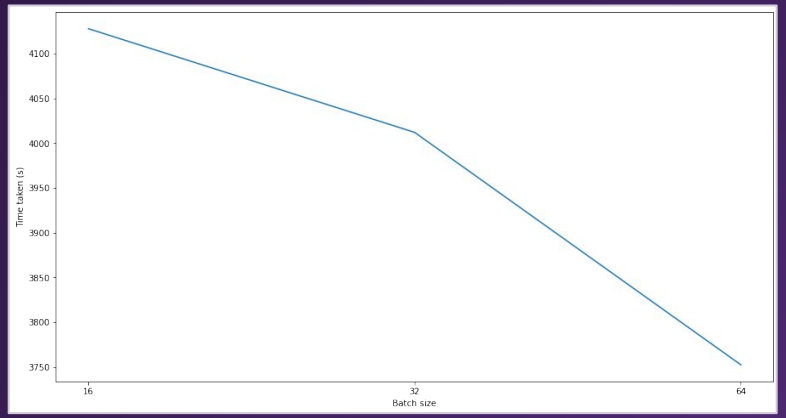
- CIFAR-10 se compose de 60,000 images divisées en 10 classes ayant chacune 6,000 images.
- 50,000 images pour l'entraînement
- 10,000 images pour les tests.

Nos différentes optimisations

Exécution avec une variation du batch size

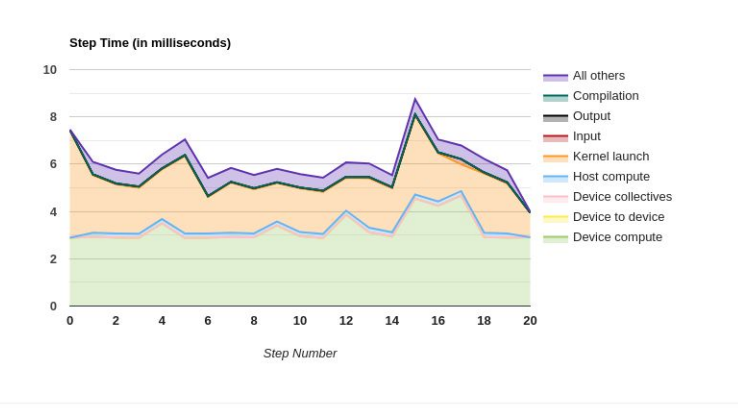


Variation du batch size pour l'application cnn

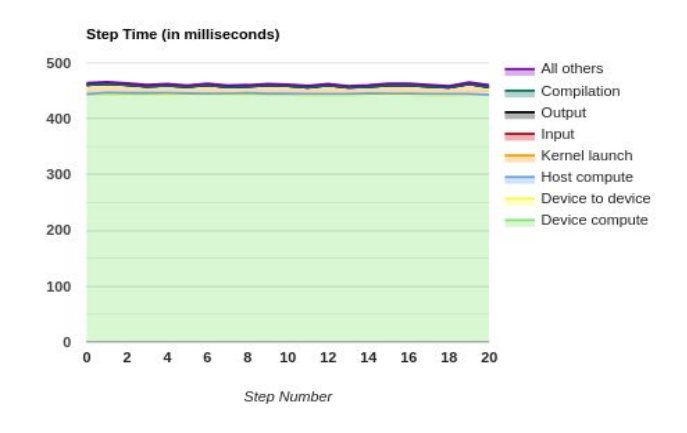


Variation du batch size pour l'application resnet-50

Interprétation du profil avec TensorBoard



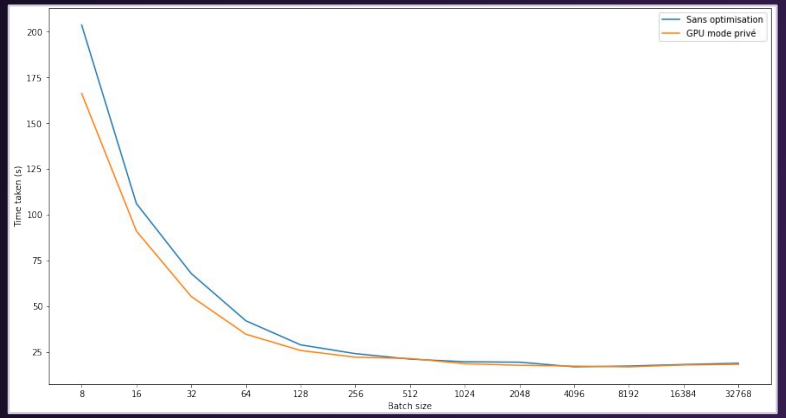
Profil de l'application cnn avec une variation du batch size



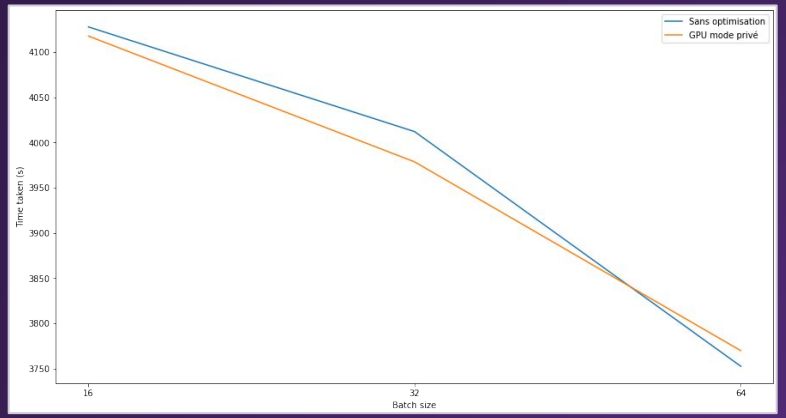
Profil de l'application resnet-50 avec une variation du batch size

Exécution avec le mode privée de la GPU

Pour chaque application et en variant le batch size, nous appliquons la première optimisation qui consiste à passer vers une GPU privée.

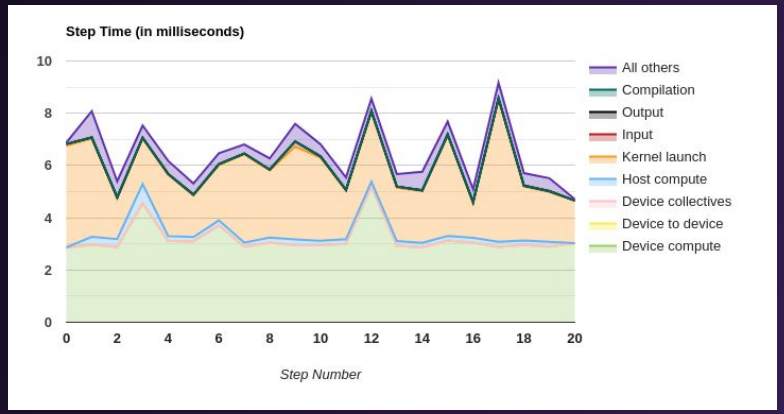


Temps d'exécution de l'application cnn avec une GPU privée

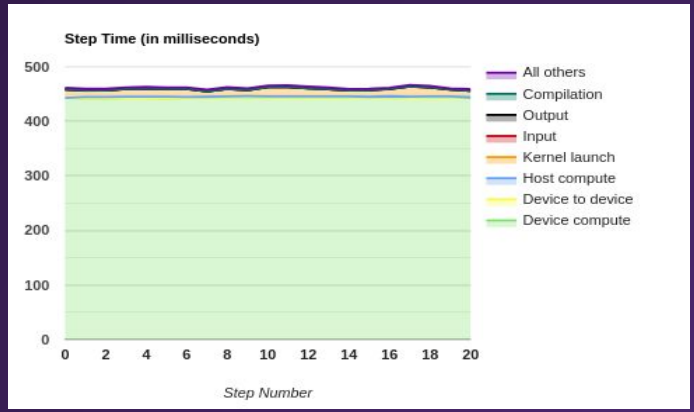


Temps d'exécution de l'application cnn avec une GPU privée

Interprétation du profil avec TensorBoard

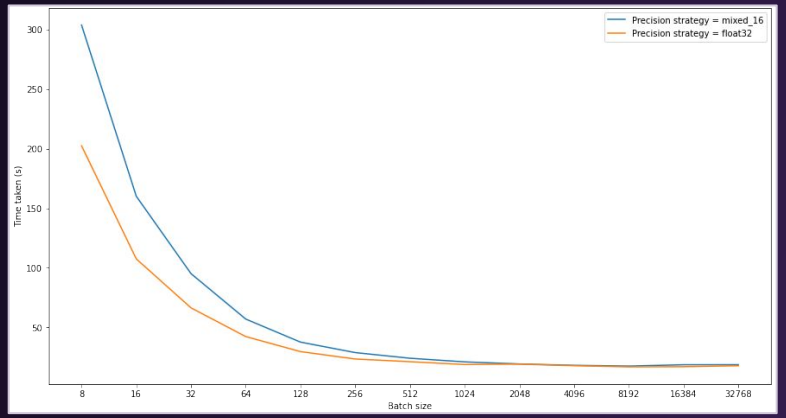


Profil de l'application cnn avec une GPU privée

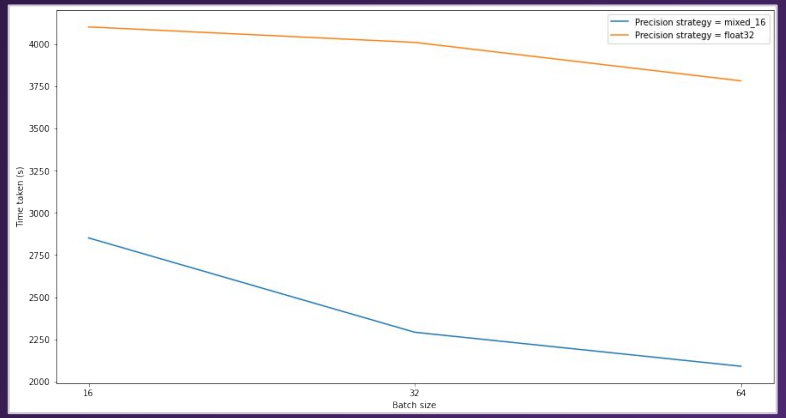


Profil de l'application resnet-50 avec une GPU privée

Exécution avec une stratégie de précision sans GPU privée

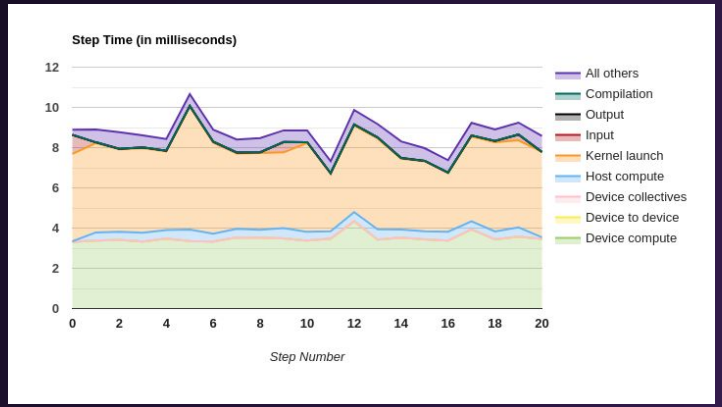


Temps d'exécution de l'application cnn avec une stratégie de précision

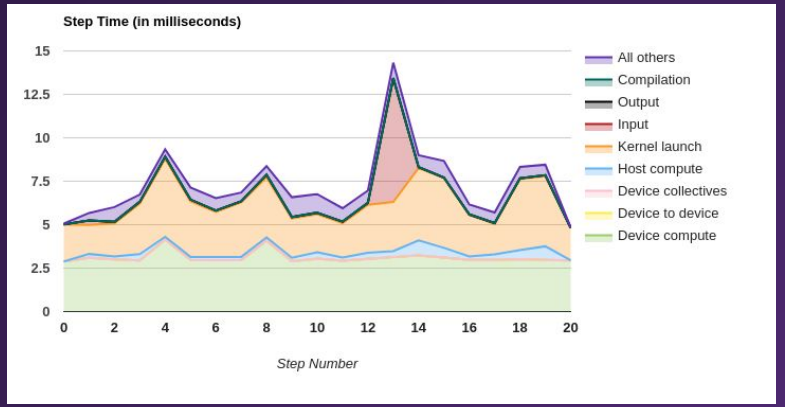


Temps d'exécution de l'application cnn avec une stratégie de précision

Interprétation du profil de l'application cnn

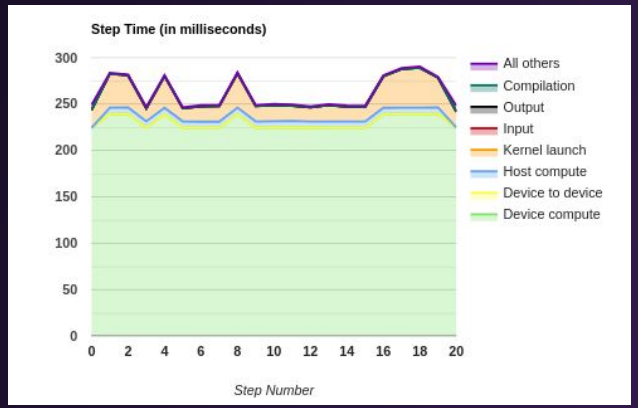


Profil de l'application cnn avec une stratégie mixed float16 sans GPU privée

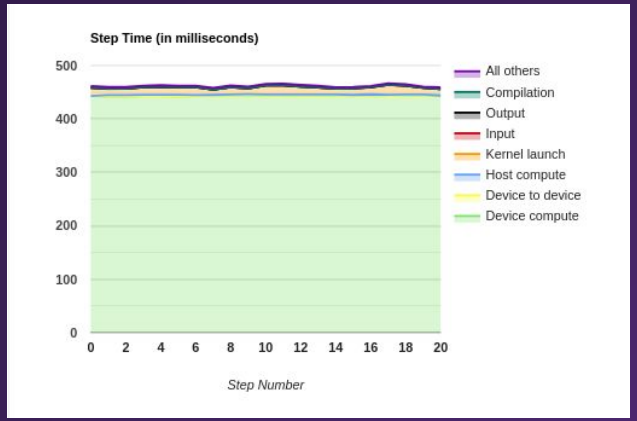


Profil de l'application cnn avec une stratégie float32 sans GPU privée

Interprétation du profil de l'application resnet-50

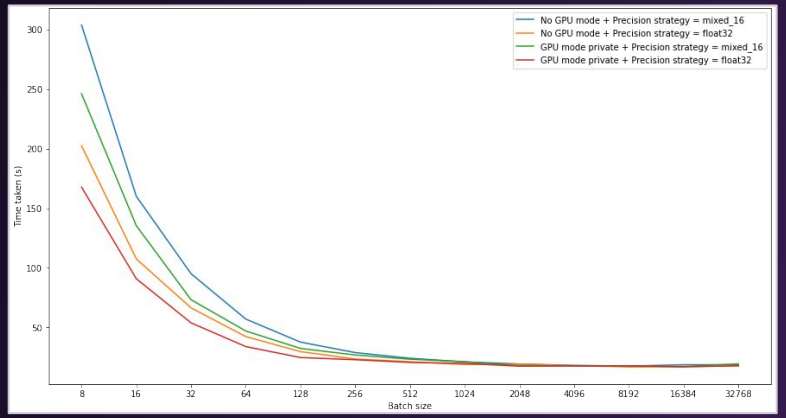


Profil de l'application resnet-50 avec une stratégie mixed float16 sans GPU privée

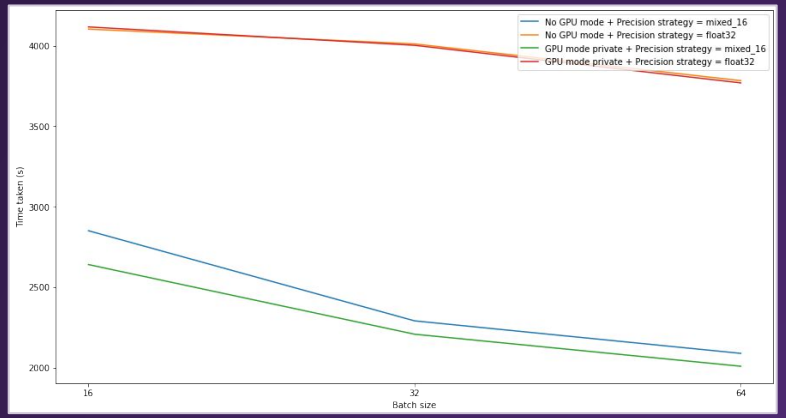


Profil de l'application resnet-50 avec une stratégie float32 sans GPU privée

Exécution avec une stratégie de précision et une GPU privée

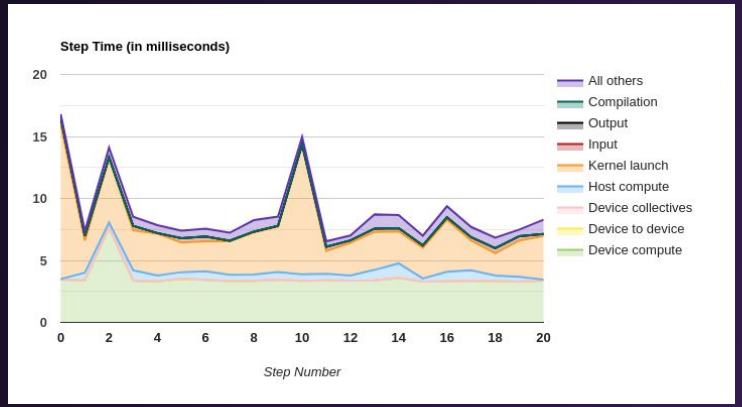


Temps d'exécution de l'application cnn avec une GPU privée et une stratégie de précision

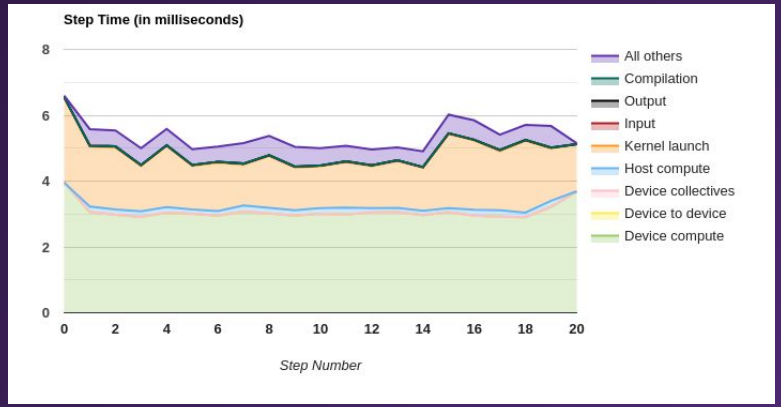


Temps d'exécution de l'application cnn avec une GPU privée et une stratégie de précision

Interprétation du profil de l'application cnn

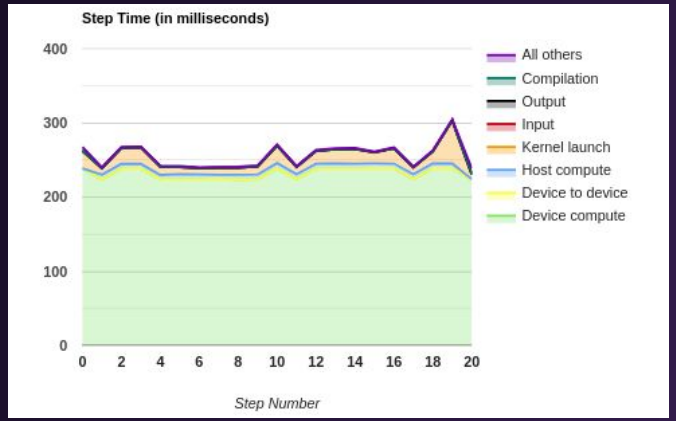


Profil de l'application cnn avec une stratégie mixed float16 avec GPU privée

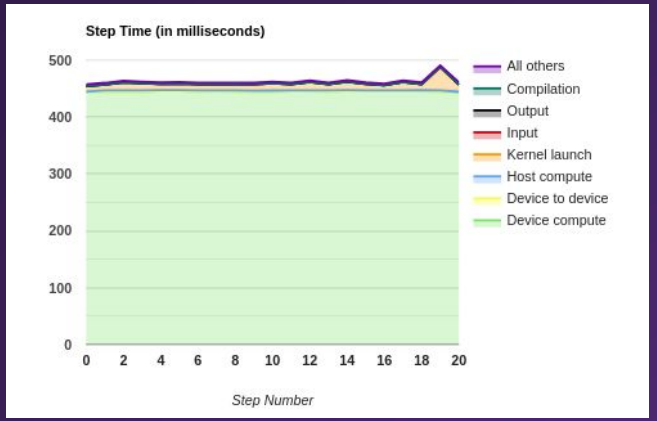


Profil de l'application cnn avec une stratégie float32 avec GPU privée

Interprétation du profil de l'application resnet-50

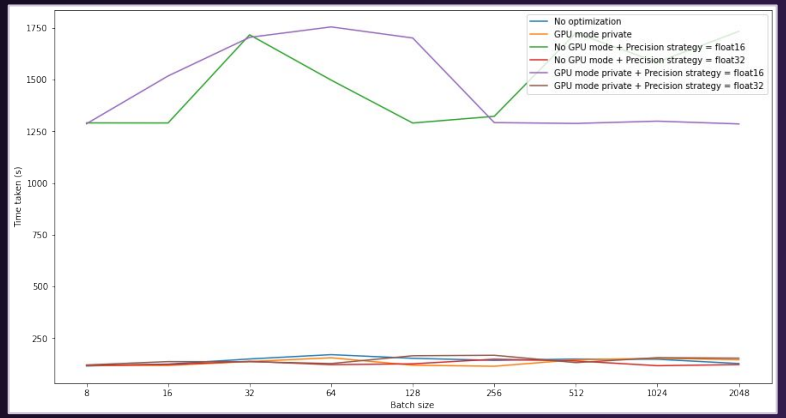


Profil de l'application resnet-50 avec une stratégie mixed float16 avec GPU privée

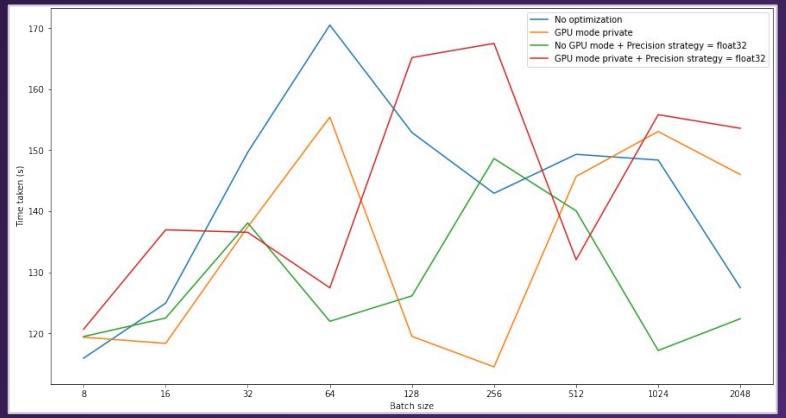


Profil de l'application resnet-50 avec une stratégie float32 avec GPU privée

Prélecture des données

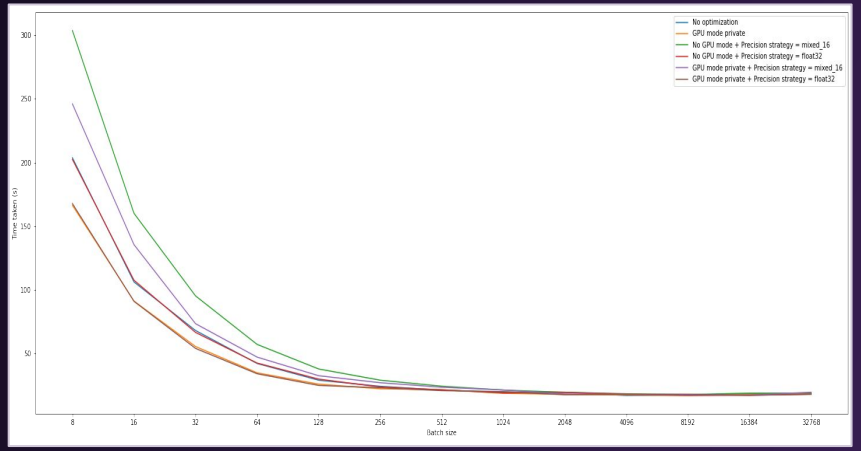


Temps d'exécution de l'application cnn avec une prélecture des données

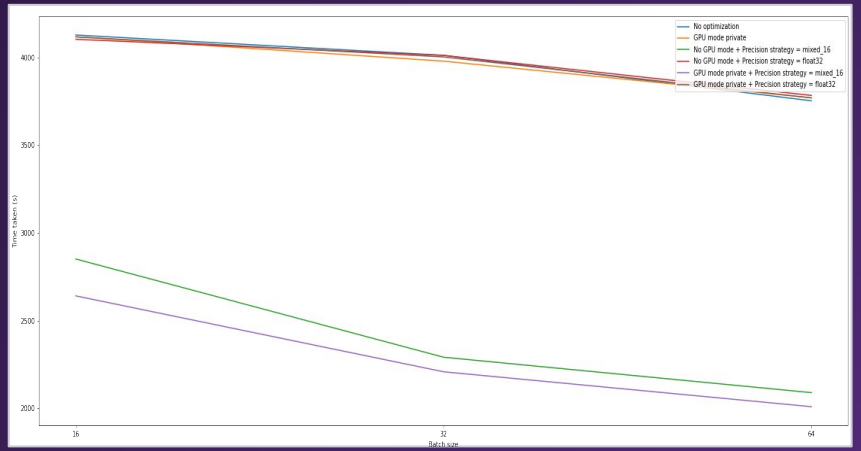


Zoom sur la partie inférieure du tracé représentant tous les temps d'exécution de l'application cnn

Récapitulatifs de toutes les optimisations

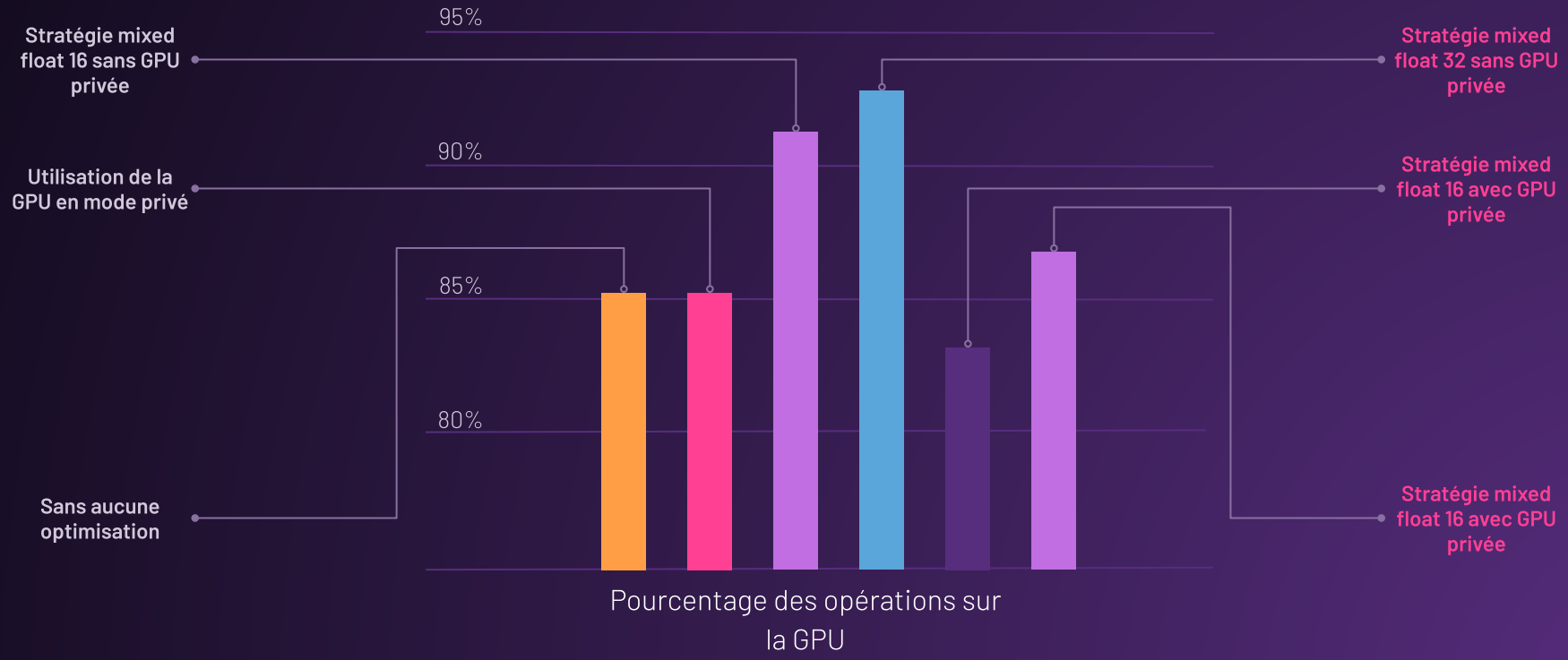


Temps d'exécution de toutes les optimisations pour l'application cnn

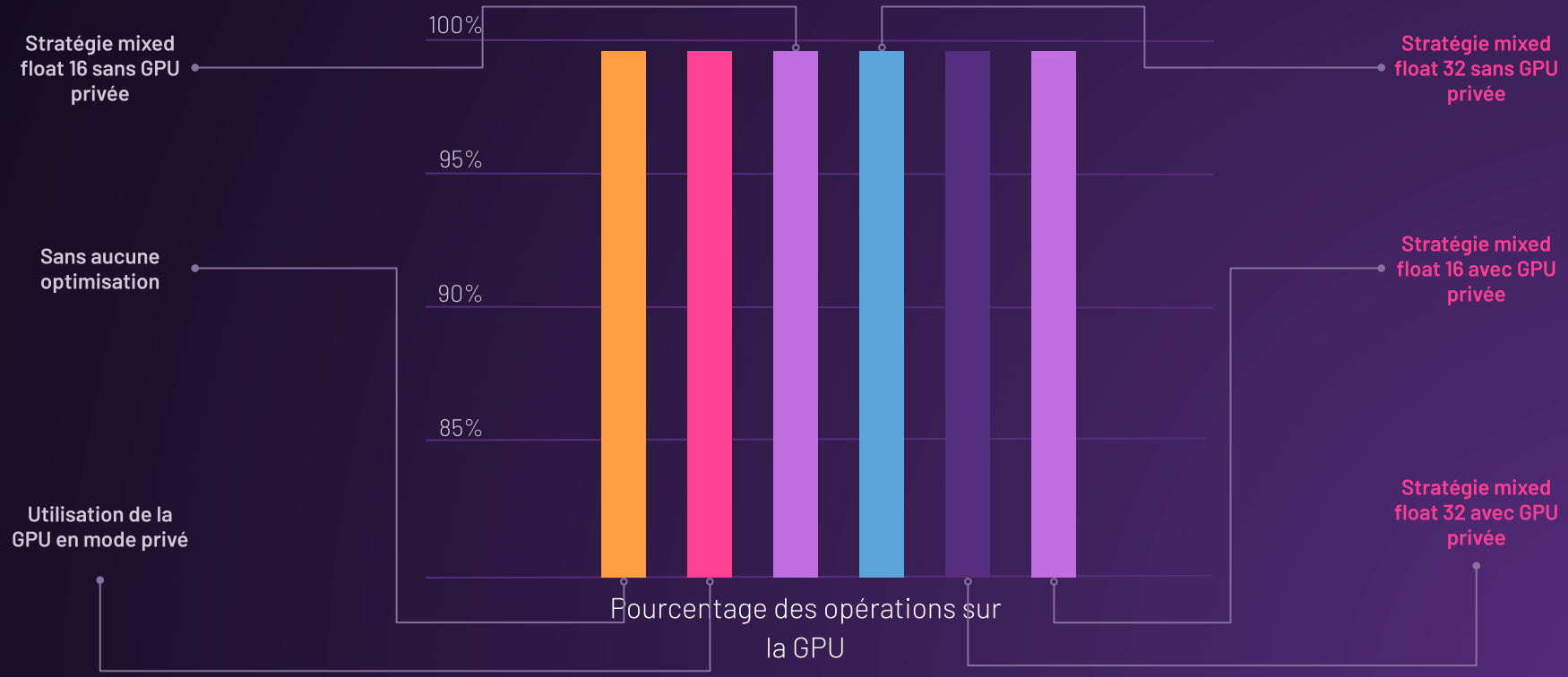


Temps d'exécution de toutes les optimisations pour l'application resnet-50

Résumé des TF Op Placement -> Device sur l'application cnn pour chaque optimisation



Résumé des TF Op Placement -> Device sur l'application resnet-50 pour chaque optimisation



Comparaisons des optimisations

Application simple cnn	Temps de pas moyen (en ms)	Temps de calcul sur la GPU (en ms)
Sans / avec GPU privée	6.1 / 6.5	3.2
Stratégie mixed float 16 sans / avec GPU privée	8.7 / 8.5	3.5 / 3.6
Stratégie float 32 sans / avec GPU privée	7.3 / 5.4	3.1

Application resnet	Temps de pas moyen (en ms)	Temps de calcul sur la GPU (en ms)
Sans / avec GPU privée	461.6 / 461.7	442.8 / 443
Stratégie mixed float 16 sans / avec GPU privée	261.8 / 255.8	230.1 / 231.8
Stratégie float 32 sans / avec GPU privée	431 / 462.4	443 / 444.7

DÉMONSTRATION

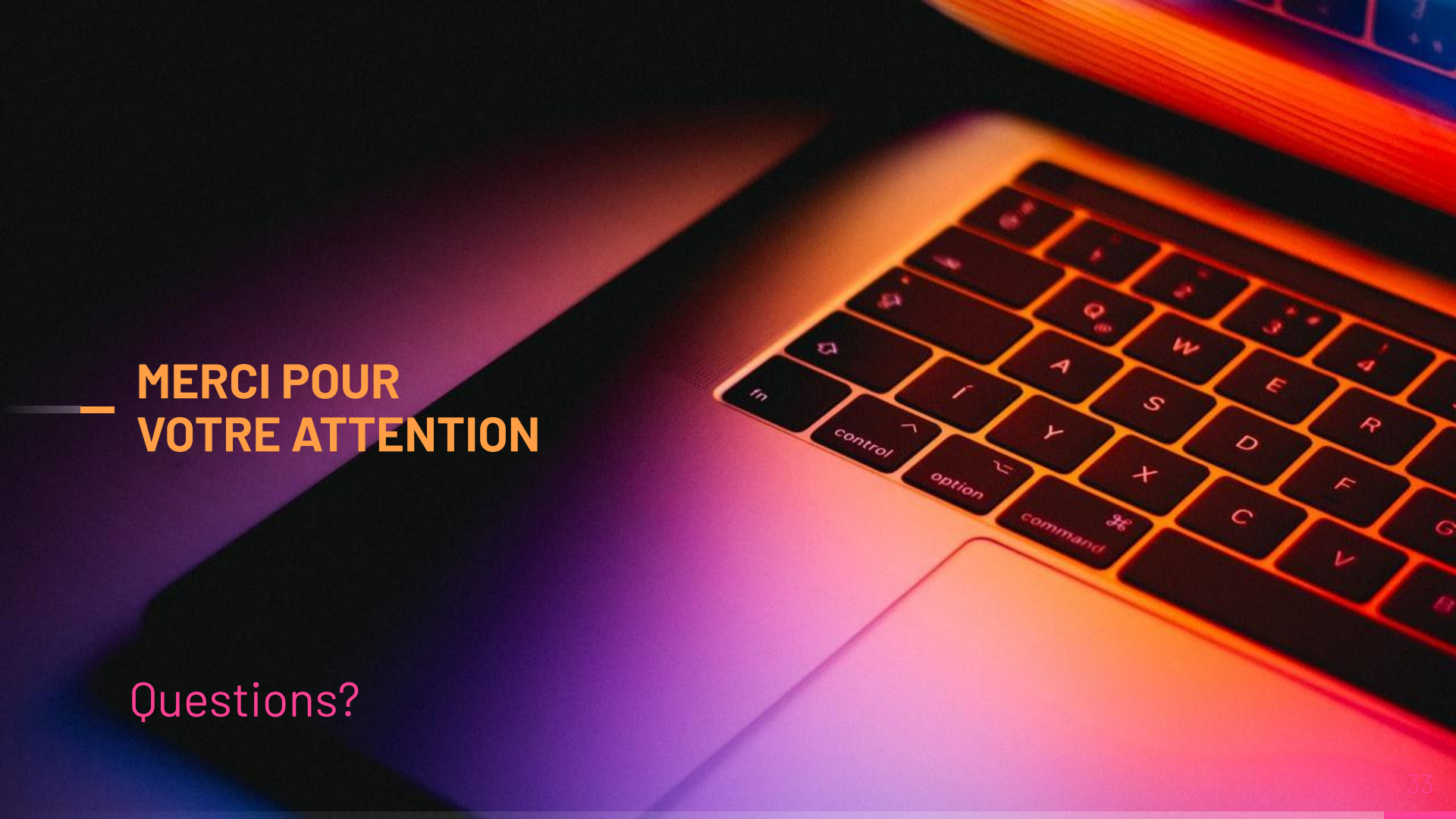
Laissons la magie opérer

CONCLUSION ET PERSPECTIVES

C’est ici que tout se termine...

- Les outils d'analyse de performance jouent un rôle primordial dans l'optimisation des projets Tensorflow (Temps et mémoire).
- Mise en valeur des profilers grâce à deux applications.
- Observation de l'impact des optimisations réalisées sur les performances des applications.

- Étudier le profil TensorBoard lors de la prélecture des données.
- Développer de nouvelles analyses, notamment en regardant ce qui se passe à bas niveau (au niveau de cuda).


MERCI POUR
VOTRE ATTENTION
Questions?