

Comparaison des Performances MPI : C vs Python

Chadha Sakka

December 15, 2024

1 Introduction

Le Message Passing Interface (MPI) est une norme permettant des communications parallèles efficaces dans des environnements distribués. Ce rapport vise à comparer les performances des implémentations MPI en **C** et **Python** (via `mpi4py`), en mettant l'accent sur la bande passante bi-directionnelle. L'objectif est de mettre en lumière les différences entre ces deux langages dans des contextes de calcul haute performance.

2 Méthodologie

2.1 Configuration Docker et Logicielle

Les benchmarks ont été réalisés dans un environnement Docker personnalisé avec la configuration suivante :

- **Version de Docker** : 27.3.1
- **Image Docker utilisée** : `mpi-python:latest`, construite à partir d'un Dockerfile incluant Open MPI et `mpi4py`.
- **Logiciels dans le conteneur** :
 - Python : version 3.9.21.
 - MPI pour Python : `mpi4py` version 4.0.1.
 - MPI pour C : Open MPI version 4.1.2.
- **Limites des ressources Docker** : Aucune limite spécifique n'a été imposée (toutes les ressources de l'hôte étaient disponibles).

L'image Docker a été construite pour inclure toutes les dépendances nécessaires à l'exécution des benchmarks, garantissant l'isolation et la reproductibilité des tests. Les tests ont été réalisés sur deux nœuds MPI identifiés par les adresses `10.0.1.2` et `10.0.1.4`. Les benchmarks utilisés sont :

- **C** : `osu_bibw` du package `osu-micro-benchmarks-5.9`.
- **Python** : `osu_bibw.py` modifié pour correspondre au test de bande passante bi-directionnelle.

Chaque test a mesuré la bande passante pour différentes tailles de messages, de 1 Byte à 4 MB. Les résultats ont été collectés sur un environnement Docker configuré avec Open MPI pour garantir une reproductibilité des tests.

3 Résultats et Analyse

3.1 Bande Passante Bi-Directionnelle

Les résultats obtenus pour les benchmarks de bande passante bi-directionnelle sont résumés dans le tableau 1 et illustrés dans la figure 1.

Table 1: Comparaison de la bande passante bi-directionnelle (MB/s)

Taille (B)	C - osu_bibw	Python - osu_bibw.py
1	0.18	0.06
2	0.37	0.13
4	0.74	0.25
8	1.48	0.50
16	2.98	1.00
32	5.95	1.98
64	11.84	3.99
128	23.04	7.87
256	46.01	15.67
1024	179.43	60.74
2048	340.01	117.12
4096	629.69	224.15
8192	1093.23	405.44
16384	1632.79	673.32
32768	2412.87	895.63
65536	1609.19	702.04
1048576	2908.66	1346.31

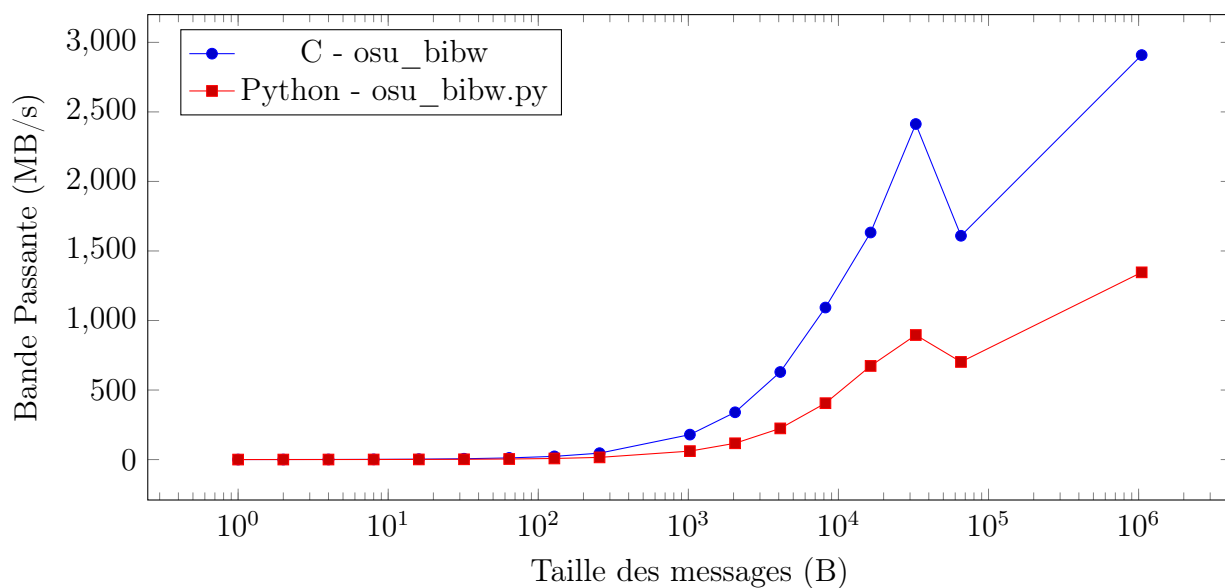


Figure 1: Comparaison de la bande passante bi-directionnelle entre C et Python

3.2 Analyse des Résultats

Les résultats montrent clairement que :

- Les performances en **C** surpassent largement celles en **Python** pour toutes les tailles de messages.
- La différence est particulièrement marquée pour les tailles de messages supérieures à 1 MB, où C atteint une bande passante proche des limites du matériel.
- Les résultats en Python montrent néanmoins une progression constante, indiquant que `mpi4py` est un outil valable pour des expérimentations légères ou du prototypage.

4 Conclusion

Ce rapport met en évidence l'écart de performance entre les implémentations MPI en C et Python. Les benchmarks montrent que les implémentations en C sont optimisées pour tirer parti des capacités matérielles, offrant une bande passante significativement supérieure, en particulier pour des tailles de messages importantes. En revanche, Python reste un choix utile pour les utilisateurs recherchant un développement rapide ou un environnement interactif.