

Systèmes Répartis

Rapport final

Le projet :

Ce projet vise à concevoir un système distribuant des paires de type "*mot : occurrence*" à travers plusieurs machines pour établir un tri global des données. Il implique deux processus Map-Shuffle-Reduce : pour le comptage des occurrences des mots et pour le tri, visant à organiser les paires par ordre croissant des occurrences et, en cas d'égalité, par ordre alphabétique des mots. Enfin le but est d'en déduire l'évolution du taux de parallélisation du cluster grâce à la loi d'Amdahl.

Objectif de ce rapport :

- Expliquer les évolutions apportées dans le code afin d'intégrer les documents du Common Crawl et de réaliser le processus de Map-Shuffle-Reduce sur cet ensemble de documents.
- Déduire le taux de parallélisation du cluster grâce à la loi d'Amdahl

Table des matières :

- I. Le programme et son évolution
 - A. Point de départ
 - B. Map - Shuffle - Reduce de comptage
 - C. Map - Shuffle - Reduce de tri
 - D. Difficultés rencontrées et solutions apportées
- II. Analyse du taux de parallélisation
 - A. Compréhension de la loi d'Amdahl
 - B. Contexte de l'analyse
 - C. Calcul du taux d'accélération
 - D. Calcul du taux de parallélisation et conclusion
- III. Annexes

Autres pièces intégrées au rendu :

- Le script envoyeur, client.py
- Le script recepteurs, serveurs.py
- Le script de déploiement deploy.sh
- Le fichier machines.txt avec la liste des machines utilisées (dans l'ordre d'utilisation)
- Le résultat du dernier run avec 12 machines
- Screenshots des logs en annexes de ce rapport

I. Le programme et son évolution

Dans ce rapport, nous nommerons la machine 0 "client" et le reste des machines distantes "serveurs".

A. Point de départ

Je suis parti du code fonctionnel développé en session de live coding, réalisant un Map - Shuffle - Reduce avec une dizaine de mots en input et trois machines.

Mon objectif fut d'abord d'adapter les deux scripts client.py et serveurs.py ainsi que le script de déploiement pour :

- intégrer 1 à 10 fichiers .warc.wet d'environ 300Mo par fichier en input du programme. Ces fichiers sont situés sur le serveur central de l'école,
- écrire les résultats du Map et du Shuffle sur un fichier temporaire en RAM, plus rapide, et lire ces fichiers temporaires pour le Reduce,
- créer un fichier .txt avec les résultats du Reduce et un avec ceux du tri dans le répertoire `/home/users/login` situé sur le disque dur de chaque machine qui permet de stocker 800 Go,
- envoyer les résultats finaux triés au client qui les écrit dans un fichier en local afin de vérifier le résultat,
- limiter les logs / print dans les boucles qui surchargent le terminal et font crasher VSCode,
- ajouter un timer pour évaluer la performance du code et son évolution en fonction du nombre de machines ajoutées au cluster.

Pour optimiser l'envoi aux machines distantes des fichiers .warc.wet volumineux et éviter une surcharge de la socket, j'ai choisi de les diviser en blocs de 50 Mo.

Pour la phase de map, le client distribue les mots de chaque bloc entre les différentes machines (serveurs) en utilisant une stratégie round-robin, via la socket et le thread d'envoi déjà en place. Chaque machine les liste dans le fichier temporaire sous la forme : "*mot* : 1". Elles effectuent ensuite une redistribution des mots pour préparer la phase de Reduce. Nous verrons plus loin la stratégie que j'ai choisi d'adopter pour ce Shuffle. Enfin, elles comptabilisent la fréquence des mots reçus et stockent le résultat dans le répertoire du disque dur `/home/users/adeformel-24`. La phase de tri est quant à elle détaillée ci-dessous.

B. Fonctionnement du premier shuffle

Communication :

L'enjeu des Shuffle est de coordonner l'envoi et la réception des paires mot - occurrence entre les machines distantes.

Dans un premier temps, j'ai gardé le code original mais celui-ci n'intégrait pas de phase de réception. Lorsque j'ai adapté le code en écrivant les mots en RAM, j'ai provoqué un blocage de la socket 2 qui n'était jamais vidée. J'ai donc ajouté une phase de réception avec `recv()` qui a permis de débloquer la socket 2. Afin de fluidifier et coordonner à la fois l'envoi et la réception sur la même socket 2, j'ai créé deux threads, comme pour la socket 1.

Fonctionnement :

Mon objectif lors du premier Shuffle est de faciliter la phase de Reduce en regroupant sur les mêmes machines les mêmes mots. Dans un système distribué, la difficulté est que chaque machine n'a accès qu'aux blocs de données qu'elle a stockés et n'a pas de visibilité sur les blocs détenus par les autres machines.

Pour contrer cette difficulté, j'ai utilisé une fonction de partitionnement appliquée aux clés : le partitionnement déterministe.

Le principe est le suivant : toutes les données avec la même clé iront à la même partition et les partitions sont préalablement déterminées pour chaque machine. Elles savent donc quelles partitions envoyer à quelles machines. Pour cela, toutes les machines doivent utiliser le même algorithme de partitionnement, afin de garantir que chaque paire clé-valeur soit attribuée à la même machine, quel que soit l'endroit où cette clé est stockée. La fonction produit toujours le même résultat pour une même clé, indépendamment de la machine sur laquelle elle est exécutée.

Pour appliquer cette stratégie, j'ai codé la fonction `choisir_machine_pour_mot`. Une fois la partition définie, les paires "*mot* : 1" sont envoyées aux machines désignées par la fonction de hachage via un thread d'envoi grâce à la fonction `envoyer_paires_phase_2`. Cette fonction hash utilise la clé (donc le mot) et applique un modulo avec le nombre total de machine. Ainsi si la machine 1 calcule ``hash(mot) % len(machines_reçues)`` pour une certaine clé et obtient la machine 2, elle sait que la machine 2 est responsable de cette clé. Si la machine 2 fait le même calcul pour la même clé, elle arrivera également à la machine 2 et gardera la clé dans son fichier temporaire. Cela garantit que la décision d'attribution pour chaque clé est la même sur toutes les machines.

Côté réception, avec la fonction `recevoir_paires_phase_2`, les paires mot - occurrence ainsi distribuées sont réceptionnées grâce à un thread de réception puis ajoutées par chaque machine au fichier temporaire.

Une fois le Shuffle terminé, les machines réalisent la phase de reduce et comptent la fréquence de chaque mot sous la forme d'une paire "*mot* : *occurrence*". Un fichier de résultat est créé dans le disque dur de chaque machine, l'output path -> cf fonction `reduce_phase_3`.

C. Implémentation de la phase de tri

La consigne pour le tri était d'utiliser l'occurrence maximum : chaque machine envoie son max au client, le client en déduit le max des max, et l'envoie aux machines qui crée des bacs de tri (autant de bacs que de machines). Par exemple, si le maximum est 100 et qu'on a 3 machines dans le cluster : le premier bac sera 0-33 et sera géré par la machine 1, le deuxième sera 33-66 géré par la machine 2 et le troisième sera 66-100 géré par la machine 3. Les clés sont alors distribuées selon leur occurrence au bac correspondant grâce à un deuxième shuffle. Enfin chaque machine réalise un tri local final : par ordre croissant, et pour occurrence égale, par ordre alphabétique. Elle envoie le résultat ainsi trié au client qui concatène le tout dans le bon ordre dans un fichier de résultat.

J'ai testé cette approche mais elle crée un déséquilibre de charges conséquent. En effet 95% des paires ont des occurrences < 200, sachant que lors des premiers tests le maximum était d'environ 800 000 pour le mot "the". Ainsi 95% des paires vont à la machine 1 et les 5% restants sont répartis entre la 2 et la 3. Cela rend le tri local impossible car toutes les paires sont concentrées sur une unique machine, nécessitant une puissance de calcul bien supérieure à ses capacités.

Pour résoudre ce problème, j'ai décidé de modifier l'approche de tri, en utilisant la notion de quantiles. Supposons que nous avons X machines dans le cluster. Chaque machine commence par trier ses paires mot-occurrence en local. Elle calcule les X-1 quantiles qui déterminent les bacs de tri. Ces deux étapes sont réalisées grâce à la fonction `tri_local_et_definir_quantiles`. Elle envoie ensuite ses quantiles au client qui joue le rôle de coordinateur et qui va déterminer les seuils globaux en sélectionnant les minimums des quantiles reçus. Cette étape permet d'aligner les limites de répartition des bacs entre toutes les machines et d'assurer une répartition de charge équilibrée entre les machines. Le client envoie ces seuils aux machines distantes qui vont se répartir en bacs (machine 1 pour quantile 1, machine 2 quantile 2, etc...) et réaliser le tri. Comme codé dans la fonction `shuffle2_sort`, chaque machine conserve les paires correspondant à sa propre classe et les paires qui appartiennent aux autres classes sont envoyées aux machines responsables. Il faudra donc organiser l'envoi et la réception des paires. J'ai utilisé ici à nouveau un thread d'envoi et un thread de réception associés à la socket 2 et basés sur les fonctions `envoyer_paires_phase_5` et `recevoir_paires_phase_5`. En cas de quantiles égaux, les paires liées à ces bacs sont également distribuées entre les machines responsables de ces quantiles. Par exemple, si trois des seuils sont égaux à 1, les paires avec une occurrence de 1 sont équitablement réparties entre ces trois bacs. Pour finir, chaque machine trie ensuite localement les paires qui lui ont été attribuées par ordre croissant d'occurrence, puis par ordre alphabétique des mots pour les occurrences égales. La fonction `trier_paires_localement` permet de réaliser ce tri. Chaque machine génère alors un fichier `tri_shuffle2_results_tp-1a201-31_vX`, stocké sur le disque dur, qui contient les résultats triés de son bac. Cette approche m'a permis de

- limiter la redistribution des paires réduisant ainsi la surcharge de la socket,
- m'assurer que chaque machine traite un segment bien défini et équilibré.

Afin de rassembler les résultats, chaque machine envoie son fichier trié au client. Ce dernier s'assure que les machines envoient leurs fichiers une par une, pour garantir que les paires soient réunies dans le bon ordre.

Pour terminer proprement, chaque machine attend un signal de fin ('FIN') du client après avoir envoyé son fichier de résultats, puis ferme ses connexions.

D. Difficultés rencontrées et solutions apportées

Problèmes de Synchronisation des Envois :

Lors de la phase d'envoi des résultats des machines au client, certaines machines rencontraient des erreurs de type `"Connection reset by peer"`. Cela était dû à un manque de synchronisation, où plusieurs machines tentaient d'envoyer leurs fichiers en même temps,

ce qui surchargeait les connexions. La mise en place d'un signal de synchronisation ('GO') envoyé par le client à chaque machine avant d'envoyer les résultats a permis de garantir que chaque machine envoie ses résultats au bon moment. Cette synchronisation a assuré que les envois soient effectués de manière ordonnée, une machine à la fois.

Quantiles égaux et distribution équitable :

Lorsque plusieurs quantiles étaient égaux (comme ce fut le cas pour les occurrences égales à 1, très nombreuses), une machine pouvait recevoir une charge disproportionnée de paires avec des valeurs identiques, menant à un déséquilibre. Une logique de répartition équitable a été implémentée pour distribuer ces paires entre les machines responsables de ces quantiles. Chaque valeur a été attribuée de manière circulaire aux machines responsables, garantissant ainsi une distribution équitable.

Tri incohérent des paires :

La logique d'abord codée attribuait toutes les occurrences supérieures aux quantiles à la dernière machine par défaut. Cela a provoqué une situation où des valeurs plus petites (comme 3 ou 4) ont été envoyées à la dernière machine, créant une rupture dans l'ordre attendu. Pour corriger le problème, j'ai ajusté la logique de répartition des paires : lorsqu'une occurrence est supérieure à tous les quantiles disponibles, on va vérifier à quelle machine elle devrait être attribuée en itérant sur les quantiles dans l'ordre.

II. Analyse du taux de parallélisation

Avec une analyse du taux de parallélisation et la loi d'Amdahl, je vais essayer de comprendre comment l'augmentation du nombre de machines dans le cluster influence les performances globales. En effet, cette loi est particulièrement utile pour déterminer la limite des bénéfices possibles lorsque des ressources supplémentaires sont ajoutées à un système réparti.

A. La loi d'Amdahl

La loi d'Amdahl est utilisée pour modéliser les gains de performance lorsque des ressources de calcul supplémentaires sont ajoutées à un système parallèle. Elle indique que, même en augmentant indéfiniment le nombre de processeurs ou de machines, les gains de performances seront limités par la portion du programme qui ne peut pas être parallélisée. Ainsi j'ai mesuré le taux d'accélération pour chaque nombre de machines du cluster, me permettant d'en déduire le taux de parallélisation du programme.

B. Contexte de l'analyse

Le calcul des temps d'exécution pour l'analyse a été réalisé à distance via le VPN de l'école. Il est basé sur 10 fichiers provenant du Common Crawl dans le répertoire /cal/commoncrawl du serveur central. Ce nombre reste constant tout au long du processus. C'était le minimum pour observer une réelle évolution sur le taux d'accélération. Il intègre de 3 à 12 machines.

Le temps d'exécution est calculé par le client. Il démarre le compteur après la division du fichier combiné en blocs et la création de la socket, au moment d'envoyer les blocs aux machines. Il s'arrête à la fin du programme, après avoir récolté le résultat de chaque machine.

Les résultats que j'ai obtenu :

Nombre de documents	Nombre de machines	Temps d'exécution
10	3	741
10	4	703
10	5	656
10	6	612
10	7	530
10	8	519
10	9	503
10	10	509
10	11	572
10	12	567

C. Calcul des accélérations

La taux d'accélération pour N machines est donné par :
temps d'exécution avec 3 machines (le minimum, la baseline) / temps d'exécution avec N machines

Les taux d'accélération du cluster :

- Pour 4 machines : $741/703 = 1.05$
- Pour 5 : $741/656 = 1.13$
- Pour 6 : $741/612 = 1.21$
- Pour 7 : $741/530 = 1.4$
- Pour 8 : $741/519 = 1.43$
- Pour 9 : $741/503 = 1.47$
- Pour 10 : $741/509 = 1.46$
- Pour 11 : $741/572 = 1.3$
- Pour 12 : $741/567 = 1.31$

On observe que l'accélération augmente avec le nombre de machines, mais de manière non linéaire. Les gains de performance sont plus importants lorsque le nombre de machines passe de 3 à 7, mais ils deviennent progressivement moins significatifs par la suite.

A partir de 9 machines, on remarque que l'accélération stagne et même diminue légèrement (comme pour 11 et 12 machines). Cela est cohérent avec la loi d'Amdahl, qui montre que la partie non parallélisable du programme limite les gains.

D. Analyse du taux de parallélisation

D'après la formule de la loi d'Amdahl, pour calculer le taux de parallélisation du programme, il faut déterminer quelle portion du programme peut être parallélisée et quelle portion est nécessairement séquentielle. A partir de la formule du taux d'accélération calculé ci-dessus, nous pouvons isoler P, le taux de parallélisation.

Exemple détaillé pour 7 machines :

$$S(7) = 1.4$$

D'après la formule du taux d'accélération, $P = N \times (S(N) - 1) / S(N) \times (N - 1)$

En remplaçant par les valeurs, on obtient :

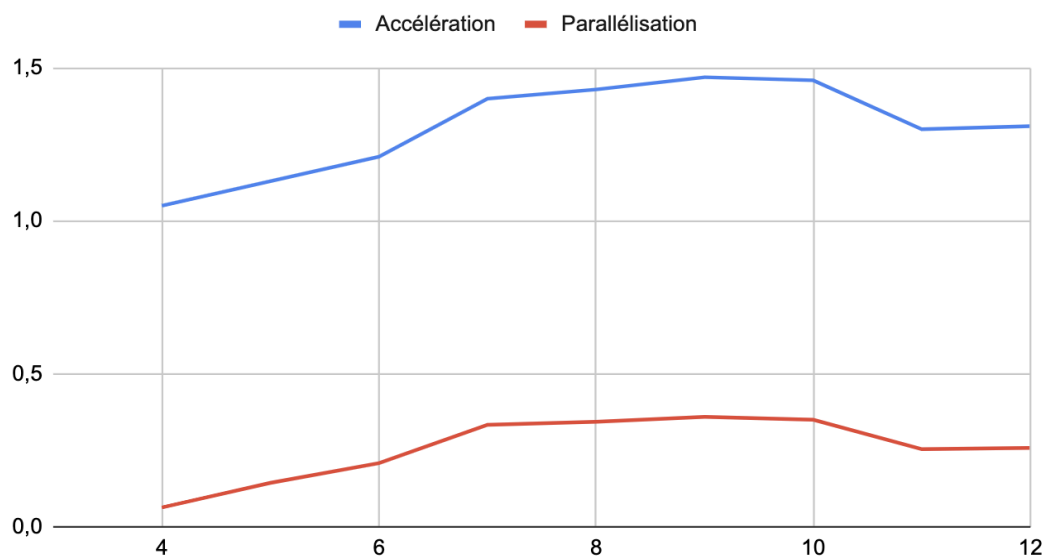
$$P = 7 \times 0.4 / 1.4 \times 6 = 0.33$$

Les résultats :

Nombre de machines	Temps d'exécution	Accélération	Parallélisation
3	741		
4	703	1,05	0,06
5	656	1,13	0,14
6	612	1,21	0,21
7	530	1,4	0,33
8	519	1,43	0,34
9	503	1,47	0,36
10	509	1,46	0,35
11	572	1,3	0,25
12	567	1,31	0,26

Conclusion : au maximum 36% du programme est parallélisable, ce qui signifie que 64% du programme est séquentiel. Ainsi, même en augmentant le nombre de machines, les gains de performance seront limités. Pour obtenir de meilleures performances, il serait donc nécessaire de réduire la partie séquentielle du programme ou d'augmenter la proportion de code parallélisable.

Taux d'accélération et de parallélisation en fonction du nombre de processeurs



III. Annexes

Screenshots des logs lors du déploiement

CÔTÉ CLIENT

```
→ TP python3 client.py
Les 10 fichiers ont été combinés.
Divisé le fichier en 66 blocs de 50 Mo.
Connexion établie avec tp-1a201-30
Connexion établie avec tp-1a201-31
Connexion établie avec tp-1a201-32
Connexion établie avec tp-1a201-05
Connexion établie avec tp-1a201-08
Connexion établie avec tp-1a201-23
Connexion établie avec tp-1a201-24
Connexion établie avec tp-1a201-25
Connexion établie avec tp-1a201-26
Connexion établie avec tp-1a201-10
Connexion établie avec tp-1a201-20
Connexion établie avec tp-1a201-21
Connexion établie avec tp-1a201-16
Envoyé la liste des machines à tp-1a201-30
Envoyé la liste des machines à tp-1a201-31
Envoyé la liste des machines à tp-1a201-32
Envoyé la liste des machines à tp-1a201-05
Envoyé la liste des machines à tp-1a201-08
Envoyé la liste des machines à tp-1a201-23
Envoyé la liste des machines à tp-1a201-24
Envoyé la liste des machines à tp-1a201-25
Envoyé la liste des machines à tp-1a201-26
Envoyé la liste des machines à tp-1a201-10
Envoyé la liste des machines à tp-1a201-20
Envoyé la liste des machines à tp-1a201-21
Envoyé la liste des machines à tp-1a201-16
Envoyé un bloc à tp-1a201-30
Envoyé un bloc à tp-1a201-30
Envoyé un bloc à tp-1a201-30
Envoyé un bloc à tp-1a201-30
Envoyé un bloc à tp-1a201-30
Envoyé un bloc à tp-1a201-31
Envoyé un bloc à tp-1a201-31
Envoyé un bloc à tp-1a201-31
Envoyé un bloc à tp-1a201-31
Envoyé un bloc à tp-1a201-32
Envoyé un bloc à tp-1a201-32
Envoyé un bloc à tp-1a201-32
Envoyé un bloc à tp-1a201-32
Envoyé un bloc à tp-1a201-05
Envoyé un bloc à tp-1a201-05
Envoyé un bloc à tp-1a201-05
Envoyé un bloc à tp-1a201-05
Envoyé un bloc à tp-1a201-08
Envoyé un bloc à tp-1a201-08
```

```
Envoyé un bloc à tp-1a201-20
Envoyé un bloc à tp-1a201-21
Envoyé un bloc à tp-1a201-21
Envoyé un bloc à tp-1a201-21
Envoyé un bloc à tp-1a201-21
Envoyé un bloc à tp-1a201-16
Envoyé un bloc à tp-1a201-16
Envoyé un bloc à tp-1a201-16
Envoyé un bloc à tp-1a201-16
Envoyé un bloc à tp-1a201-16
Envoyé 'FIN PHASE 1' à tp-1a201-30
Envoyé 'FIN PHASE 1' à tp-1a201-31
Envoyé 'FIN PHASE 1' à tp-1a201-32
Envoyé 'FIN PHASE 1' à tp-1a201-05
Envoyé 'FIN PHASE 1' à tp-1a201-08
Envoyé 'FIN PHASE 1' à tp-1a201-23
Envoyé 'FIN PHASE 1' à tp-1a201-24
Envoyé 'FIN PHASE 1' à tp-1a201-25
Envoyé 'FIN PHASE 1' à tp-1a201-26
Envoyé 'FIN PHASE 1' à tp-1a201-10
Envoyé 'FIN PHASE 1' à tp-1a201-20
Envoyé 'FIN PHASE 1' à tp-1a201-21
Envoyé 'FIN PHASE 1' à tp-1a201-16
Envoyé 'GO PHASE 2' à tp-1a201-30
Envoyé 'GO PHASE 2' à tp-1a201-31
Envoyé 'GO PHASE 2' à tp-1a201-32
Envoyé 'GO PHASE 2' à tp-1a201-05
Envoyé 'GO PHASE 2' à tp-1a201-08
Envoyé 'GO PHASE 2' à tp-1a201-23
Envoyé 'GO PHASE 2' à tp-1a201-24
Envoyé 'GO PHASE 2' à tp-1a201-25
Envoyé 'GO PHASE 2' à tp-1a201-26
Envoyé 'GO PHASE 2' à tp-1a201-10
Envoyé 'GO PHASE 2' à tp-1a201-20
Envoyé 'GO PHASE 2' à tp-1a201-21
Envoyé 'GO PHASE 2' à tp-1a201-16
'FIN PHASE 2' reçu de tp-1a201-30
'FIN PHASE 2' reçu de tp-1a201-31
'FIN PHASE 2' reçu de tp-1a201-32
'FIN PHASE 2' reçu de tp-1a201-05
'FIN PHASE 2' reçu de tp-1a201-08
'FIN PHASE 2' reçu de tp-1a201-23
'FIN PHASE 2' reçu de tp-1a201-24
'FIN PHASE 2' reçu de tp-1a201-25
'FIN PHASE 2' reçu de tp-1a201-26
'FIN PHASE 2' reçu de tp-1a201-10
'FIN PHASE 2' reçu de tp-1a201-20
'FIN PHASE 2' reçu de tp-1a201-21
'FIN PHASE 2' reçu de tp-1a201-16
Envoyé 'GO PHASE 3' à tp-1a201-30
Envoyé 'GO PHASE 3' à tp-1a201-31
```



```
Envoyé 'GO PHASE 3' à tp-1a201-25
Envoyé 'GO PHASE 3' à tp-1a201-26
Envoyé 'GO PHASE 3' à tp-1a201-10
Envoyé 'GO PHASE 3' à tp-1a201-20
Envoyé 'GO PHASE 3' à tp-1a201-21
Envoyé 'GO PHASE 3' à tp-1a201-16
'FIN PHASE 3' reçu de tp-1a201-30
'FIN PHASE 3' reçu de tp-1a201-31
'FIN PHASE 3' reçu de tp-1a201-32
'FIN PHASE 3' reçu de tp-1a201-05
'FIN PHASE 3' reçu de tp-1a201-08
'FIN PHASE 3' reçu de tp-1a201-23
'FIN PHASE 3' reçu de tp-1a201-24
'FIN PHASE 3' reçu de tp-1a201-25
'FIN PHASE 3' reçu de tp-1a201-26
'FIN PHASE 3' reçu de tp-1a201-10
'FIN PHASE 3' reçu de tp-1a201-20
'FIN PHASE 3' reçu de tp-1a201-21
'FIN PHASE 3' reçu de tp-1a201-16
Envoyé 'GO PHASE 4' à tp-1a201-30
Envoyé 'GO PHASE 4' à tp-1a201-31
Envoyé 'GO PHASE 4' à tp-1a201-32
Envoyé 'GO PHASE 4' à tp-1a201-05
Envoyé 'GO PHASE 4' à tp-1a201-08
Envoyé 'GO PHASE 4' à tp-1a201-23
Envoyé 'GO PHASE 4' à tp-1a201-24
Envoyé 'GO PHASE 4' à tp-1a201-25
Envoyé 'GO PHASE 4' à tp-1a201-26
Envoyé 'GO PHASE 4' à tp-1a201-10
Envoyé 'GO PHASE 4' à tp-1a201-20
Envoyé 'GO PHASE 4' à tp-1a201-21
Envoyé 'GO PHASE 4' à tp-1a201-16
'FIN PHASE 4' reçu de tp-1a201-30
'FIN PHASE 4' reçu de tp-1a201-31
'FIN PHASE 4' reçu de tp-1a201-32
'FIN PHASE 4' reçu de tp-1a201-05
'FIN PHASE 4' reçu de tp-1a201-08
'FIN PHASE 4' reçu de tp-1a201-23
'FIN PHASE 4' reçu de tp-1a201-24
'FIN PHASE 4' reçu de tp-1a201-25
'FIN PHASE 4' reçu de tp-1a201-26
'FIN PHASE 4' reçu de tp-1a201-10
'FIN PHASE 4' reçu de tp-1a201-20
'FIN PHASE 4' reçu de tp-1a201-21
'FIN PHASE 4' reçu de tp-1a201-16
Quantiles minimaux déterminés :[1, 1, 1, 1, 1, 1, 2, 2, 3, 4, 10]
Envoyé 'GO PHASE 5' et seuils à tp-1a201-30
Envoyé 'GO PHASE 5' et seuils à tp-1a201-31
Envoyé 'GO PHASE 5' et seuils à tp-1a201-32
```

```
Quantiles minimaux déterminés :[1, 1, 1, 1, 1, 1, 2, 2, 3, 4, 10]
Envoyé 'GO PHASE 5' et seuils à tp-1a201-30
Envoyé 'GO PHASE 5' et seuils à tp-1a201-31
Envoyé 'GO PHASE 5' et seuils à tp-1a201-32
Envoyé 'GO PHASE 5' et seuils à tp-1a201-05
Envoyé 'GO PHASE 5' et seuils à tp-1a201-08
Envoyé 'GO PHASE 5' et seuils à tp-1a201-23
Envoyé 'GO PHASE 5' et seuils à tp-1a201-24
Envoyé 'GO PHASE 5' et seuils à tp-1a201-25
Envoyé 'GO PHASE 5' et seuils à tp-1a201-26
Envoyé 'GO PHASE 5' et seuils à tp-1a201-10
Envoyé 'GO PHASE 5' et seuils à tp-1a201-20
Envoyé 'GO PHASE 5' et seuils à tp-1a201-21
Envoyé 'GO PHASE 5' et seuils à tp-1a201-16
'FIN PHASE 5' reçu de tp-1a201-30
'FIN PHASE 5' reçu de tp-1a201-31
'FIN PHASE 5' reçu de tp-1a201-32
'FIN PHASE 5' reçu de tp-1a201-05
'FIN PHASE 5' reçu de tp-1a201-08
'FIN PHASE 5' reçu de tp-1a201-23
'FIN PHASE 5' reçu de tp-1a201-24
'FIN PHASE 5' reçu de tp-1a201-25
'FIN PHASE 5' reçu de tp-1a201-26
'FIN PHASE 5' reçu de tp-1a201-10
'FIN PHASE 5' reçu de tp-1a201-20
'FIN PHASE 5' reçu de tp-1a201-21
'FIN PHASE 5' reçu de tp-1a201-16
Envoyé 'GO PHASE 6' à tp-1a201-30
Envoyé 'GO PHASE 6' à tp-1a201-31
Envoyé 'GO PHASE 6' à tp-1a201-32
Envoyé 'GO PHASE 6' à tp-1a201-05
Envoyé 'GO PHASE 6' à tp-1a201-08
Envoyé 'GO PHASE 6' à tp-1a201-23
Envoyé 'GO PHASE 6' à tp-1a201-24
Envoyé 'GO PHASE 6' à tp-1a201-25
Envoyé 'GO PHASE 6' à tp-1a201-26
Envoyé 'GO PHASE 6' à tp-1a201-10
Envoyé 'GO PHASE 6' à tp-1a201-20
Envoyé 'GO PHASE 6' à tp-1a201-21
Envoyé 'GO PHASE 6' à tp-1a201-16
Client : Début de la réception des résultats des serveurs. Résultat enregistré dans 'resultat_v13.txt'.
Client : Résultats de tp-1a201-30 ajoutés au fichier 'resultat_v13.txt'.
Client : Signal de fin envoyé à tp-1a201-30.
Client : Résultats de tp-1a201-31 ajoutés au fichier 'resultat_v13.txt'.
Client : Signal de fin envoyé à tp-1a201-31.
Client : Résultats de tp-1a201-32 ajoutés au fichier 'resultat_v13.txt'.
Client : Signal de fin envoyé à tp-1a201-32.
Client : Résultats de tp-1a201-05 ajoutés au fichier 'resultat_v13.txt'.
```

```

Envoyé 'GO PHASE 6' à tp-1a201-23
Envoyé 'GO PHASE 6' à tp-1a201-24
Envoyé 'GO PHASE 6' à tp-1a201-25
Envoyé 'GO PHASE 6' à tp-1a201-26
Envoyé 'GO PHASE 6' à tp-1a201-10
Client : Début de la réception des résultats des serveurs. Résultat enregistré dans 'resultat_v10.txt'.
Client : Résultats de tp-1a201-30 ajoutés au fichier 'resultat_v10.txt'.
Client : Signal de fin envoyé à tp-1a201-30.
Client : Résultats de tp-1a201-31 ajoutés au fichier 'resultat_v10.txt'.
Client : Signal de fin envoyé à tp-1a201-31.
Client : Résultats de tp-1a201-32 ajoutés au fichier 'resultat_v10.txt'.
Client : Signal de fin envoyé à tp-1a201-32.
Client : Résultats de tp-1a201-05 ajoutés au fichier 'resultat_v10.txt'.
Client : Signal de fin envoyé à tp-1a201-05.
Client : Résultats de tp-1a201-08 ajoutés au fichier 'resultat_v10.txt'.
Client : Signal de fin envoyé à tp-1a201-08.
Client : Résultats de tp-1a201-23 ajoutés au fichier 'resultat_v10.txt'.
Client : Signal de fin envoyé à tp-1a201-23.
Client : Résultats de tp-1a201-24 ajoutés au fichier 'resultat_v10.txt'.
Client : Signal de fin envoyé à tp-1a201-24.
Client : Résultats de tp-1a201-25 ajoutés au fichier 'resultat_v10.txt'.
Client : Signal de fin envoyé à tp-1a201-25.
Client : Résultats de tp-1a201-26 ajoutés au fichier 'resultat_v10.txt'.
Client : Signal de fin envoyé à tp-1a201-26.
Client : Résultats de tp-1a201-10 ajoutés au fichier 'resultat_v10.txt'.
Client : Signal de fin envoyé à tp-1a201-10.
Client : Fin de la réception des résultats des serveurs.Fichier complet : 'resultat_v10.txt'.
Durée totale du processus : 508.92 secondes

```

CÔTE RÉCEPTEURS :

```

'PHASE 2 tp-1a201-21' : En attente de connexion...
'PHASE 2 tp-1a201-10' : Connexion acceptée de ('137.194.140.209', 37806)
'tp-1a201-10' : Connexion enregistrée pour ('137.194.140.209', 37806)
'PHASE 2 tp-1a201-10' : En attente de connexion...
tp-1a201-16:Connexion établie avec tp-1a201-30
tp-1a201-16:Connexion établie avec tp-1a201-31
tp-1a201-16:Connexion établie avec tp-1a201-32
tp-1a201-16:Connexion établie avec tp-1a201-05
tp-1a201-16:Connexion établie avec tp-1a201-08
tp-1a201-16:Connexion établie avec tp-1a201-23
tp-1a201-16:Connexion établie avec tp-1a201-24
tp-1a201-16:Connexion établie avec tp-1a201-25
tp-1a201-16:Connexion établie avec tp-1a201-26
tp-1a201-16:Connexion établie avec tp-1a201-10
tp-1a201-16:Connexion établie avec tp-1a201-20
tp-1a201-16:Connexion établie avec tp-1a201-21
tp-1a201-16:Connexion établie avec tp-1a201-16
'PHASE 2 tp-1a201-16' : Connexion acceptée de ('137.194.140.209', 43768)
'tp-1a201-16' : Connexion enregistrée pour ('137.194.140.209', 43768)
'PHASE 2 tp-1a201-16' : En attente de connexion...
Buffer de la socket vidé.
'tp-1a201-16' : Début de l'envoi des paires clé-valeur pour la phase 2.
'tp-1a201-16' : Début de la réception des paires clé-valeur des autres machines pour la phase 2.
Buffer de la socket vidé.
'tp-1a201-26' : Début de l'envoi des paires clé-valeur pour la phase 2.
Buffer de la socket vidé.
'tp-1a201-21' : Début de l'envoi des paires clé-valeur pour la phase 2.
Buffer de la socket vidé.
Buffer de la socket vidé.
'tp-1a201-21' : Début de la réception des paires clé-valeur des autres machines pour la phase 2.
'tp-1a201-24' : Début de l'envoi des paires clé-valeur pour la phase 2.
'tp-1a201-26' : Début de la réception des paires clé-valeur des autres machines pour la phase 2.
'tp-1a201-20' : Début de l'envoi des paires clé-valeur pour la phase 2.
tp-1a201-21: Début de la phase 3
'REDUCE PHASE 3 tp-1a201-21' : Calcul des occurrences des mots
tp-1a201-26: Début de la phase 3
'REDUCE PHASE 3 tp-1a201-26' : Calcul des occurrences des mots
'tp-1a201-24' : Début de la réception des paires clé-valeur des autres machines pour la phase 2.
'tp-1a201-20' : Début de la réception des paires clé-valeur des autres machines pour la phase 2.
tp-1a201-24: Début de la phase 3
Buffer de la socket vidé.
Buffer de la socket vidé.
'REDUCE PHASE 3 tp-1a201-24' : Calcul des occurrences des mots
Buffer de la socket vidé.
tp-1a201-20: Début de la phase 3
'REDUCE PHASE 3 tp-1a201-20' : Calcul des occurrences des mots
'tp-1a201-05' : Début de l'envoi des paires clé-valeur pour la phase 2.

```

```
tp-1a201-05: Début de la phase 3
'REDUCE PHASE 3 tp-1a201-05' : Calcul des occurrences des mots
'tp-1a201-10' : Début de la réception des paires clé-valeur des autres machines pour la phase 2.
tp-1a201-25: Début de la phase 3
'REDUCE PHASE 3 tp-1a201-25' : Calcul des occurrences des mots
tp-1a201-10: Début de la phase 3
'REDUCE PHASE 3 tp-1a201-10' : Calcul des occurrences des mots
tp-1a201-16: Début de la phase 3
'REDUCE PHASE 3 tp-1a201-16' : Calcul des occurrences des mots
'PHASE 2 tp-1a201-16' : Connexion acceptée de ('137.194.140.203', 51356)
'tp-1a201-16' : Connexion enregistrée pour ('137.194.140.203', 51356)
'PHASE 2 tp-1a201-16' : En attente de connexion...
tp-1a201-31:Connexion établie avec tp-1a201-16
Buffer de la socket vidé.
'tp-1a201-31' : Début de l'envoi des paires clé-valeur pour la phase 2.
'tp-1a201-31' : Début de la réception des paires clé-valeur des autres machines pour la phase 2.
tp-1a201-31: Début de la phase 3
'REDUCE PHASE 3 tp-1a201-31' : Calcul des occurrences des mots
'PHASE 2 tp-1a201-16' : Connexion acceptée de ('137.194.140.224', 42068)
'tp-1a201-16' : Connexion enregistrée pour ('137.194.140.224', 42068)
'PHASE 2 tp-1a201-16' : En attente de connexion...
tp-1a201-30:Connexion établie avec tp-1a201-16
Buffer de la socket vidé.
'tp-1a201-30' : Début de l'envoi des paires clé-valeur pour la phase 2.
'PHASE 2 tp-1a201-16' : Connexion acceptée de ('137.194.140.223', 53070)
'tp-1a201-16' : Connexion enregistrée pour ('137.194.140.223', 53070)
'PHASE 2 tp-1a201-16' : En attente de connexion...
'tp-1a201-30' : Début de la réception des paires clé-valeur des autres machines pour la phase 2.
tp-1a201-30: Début de la phase 3
'REDUCE PHASE 3 tp-1a201-30' : Calcul des occurrences des mots
tp-1a201-32:Connexion établie avec tp-1a201-16
Buffer de la socket vidé.
'tp-1a201-32' : Début de l'envoi des paires clé-valeur pour la phase 2.
tp-1a201-08:Connexion établie avec tp-1a201-16
Buffer de la socket vidé.
'tp-1a201-08' : Début de l'envoi des paires clé-valeur pour la phase 2.
'tp-1a201-32' : Début de la réception des paires clé-valeur des autres machines pour la phase 2.
'tp-1a201-08' : Début de la réception des paires clé-valeur des autres machines pour la phase 2.
tp-1a201-32: Début de la phase 3
'REDUCE PHASE 3 tp-1a201-32' : Calcul des occurrences des mots
'PHASE 2 tp-1a201-16' : Connexion acceptée de ('137.194.140.225', 53592)
'tp-1a201-16' : Connexion enregistrée pour ('137.194.140.225', 53592)
'PHASE 2 tp-1a201-16' : En attente de connexion...
'PHASE 2 tp-1a201-16' : Connexion acceptée de ('137.194.140.201', 55606)
'tp-1a201-16' : Connexion enregistrée pour ('137.194.140.201', 55606)
'PHASE 2 tp-1a201-16' : En attente de connexion...
tp-1a201-08: Début de la phase 3
'REDUCE PHASE 3 tp-1a201-08' : Calcul des occurrences des mots
tp-1a201-23:Connexion établie avec tp-1a201-16
Buffer de la socket vidé.
'tp-1a201-23' : Début de l'envoi des paires clé-valeur pour la phase 2.
'tp-1a201-23' : Début de la réception des paires clé-valeur des autres machines pour la phase 2.
```

```
'tp-1a201-23' : Début de l'envoi des paires clé-valeur pour la phase 2.
'tp-1a201-23' : Début de la réception des paires clé-valeur des autres machines pour la phase 2.
'PHASE 2 tp-1a201-16' : Connexion acceptée de ('137.194.140.216', 35218)
'tp-1a201-16' : Connexion enregistrée pour ('137.194.140.216', 35218)
'PHASE 2 tp-1a201-16' : En attente de connexion...
tp-1a201-23: Début de la phase 3
'REDUCE PHASE 3 tp-1a201-23' : Calcul des occurrences des mots
Résultats du reduce stockés dans /home/users/adeformel-24/reduce_results_tp-1a201-25_v6.txt
'REDUCE PHASE 3 tp-1a201-25' : Calcul des occurrences terminé.
'PHASE 3 tp-1a201-25' : Fin de la phase 3
Résultats du reduce stockés dans /home/users/adeformel-24/reduce_results_tp-1a201-31_v11.txt
'REDUCE PHASE 3 tp-1a201-31' : Calcul des occurrences terminé.
'PHASE 3 tp-1a201-31' : Fin de la phase 3
Résultats du reduce stockés dans /home/users/adeformel-24/reduce_results_tp-1a201-26_v5.txt
'REDUCE PHASE 3 tp-1a201-26' : Calcul des occurrences terminé.
'PHASE 3 tp-1a201-26' : Fin de la phase 3
Résultats du reduce stockés dans /home/users/adeformel-24/reduce_results_tp-1a201-05_v10.txt
'REDUCE PHASE 3 tp-1a201-05' : Calcul des occurrences terminé.
'PHASE 3 tp-1a201-05' : Fin de la phase 3
Résultats du reduce stockés dans /home/users/adeformel-24/reduce_results_tp-1a201-10_v4.txt
'REDUCE PHASE 3 tp-1a201-10' : Calcul des occurrences terminé.
'PHASE 3 tp-1a201-10' : Fin de la phase 3
Résultats du reduce stockés dans /home/users/adeformel-24/reduce_results_tp-1a201-24_v7.txt
'REDUCE PHASE 3 tp-1a201-24' : Calcul des occurrences terminé.
'PHASE 3 tp-1a201-24' : Fin de la phase 3
Résultats du reduce stockés dans /home/users/adeformel-24/reduce_results_tp-1a201-20_v3.txt
'REDUCE PHASE 3 tp-1a201-20' : Calcul des occurrences terminé.
'PHASE 3 tp-1a201-20' : Fin de la phase 3
Résultats du reduce stockés dans /home/users/adeformel-24/reduce_results_tp-1a201-16_v1.txt
'REDUCE PHASE 3 tp-1a201-16' : Calcul des occurrences terminé.
'PHASE 3 tp-1a201-16' : Fin de la phase 3
'tp-1a201-31' : Phase 4 de tri démarrée
'tp-1a201-16' : Phase 4 de tri démarrée
Résultats du reduce stockés dans /home/users/adeformel-24/reduce_results_tp-1a201-21_v2.txt
'REDUCE PHASE 3 tp-1a201-21' : Calcul des occurrences terminé.
'PHASE 3 tp-1a201-21' : Fin de la phase 3
'tp-1a201-21' : Phase 4 de tri démarrée
'tp-1a201-05' : Phase 4 de tri démarrée
'tp-1a201-20' : Phase 4 de tri démarrée
'tp-1a201-10' : Phase 4 de tri démarrée
'tp-1a201-25' : Phase 4 de tri démarrée
'tp-1a201-26' : Phase 4 de tri démarrée
'tp-1a201-24' : Phase 4 de tri démarrée
Résultats du reduce stockés dans /home/users/adeformel-24/reduce_results_tp-1a201-32_v11.txt
'REDUCE PHASE 3 tp-1a201-32' : Calcul des occurrences terminé.
'PHASE 3 tp-1a201-32' : Fin de la phase 3
'tp-1a201-32' : Phase 4 de tri démarrée
Résultats du reduce stockés dans /home/users/adeformel-24/reduce_results_tp-1a201-30_v11.txt
'REDUCE PHASE 3 tp-1a201-30' : Calcul des occurrences terminé.
'PHASE 3 tp-1a201-30' : Fin de la phase 3
'tp-1a201-30' : Phase 4 de tri démarrée
Résultats du reduce stockés dans /home/users/adeformel-24/reduce_results_tp-1a201-08_v9.txt
```

```

'REDUCE PHASE 3 tp-1a201-08' : Calcul des occurrences terminé.
'PHASE 3 tp-1a201-08' : Fin de la phase 3
'tp-1a201-08' : Phase 4 de tri démarrée
Résultats du reduce stockés dans /home/users/adeformel-24/reduce_results_tp-1a201-23_v8.txt
'REDUCE PHASE 3 tp-1a201-23' : Calcul des occurrences terminé.
'PHASE 3 tp-1a201-23' : Fin de la phase 3
'tp-1a201-23' : Phase 4 de tri démarrée
'tp-1a201-21' : Tri local des résultats du reduce terminé.
'tp-1a201-21' : Quantiles définis : [1, 1, 1, 1, 1, 1, 1, 2, 2, 3, 6, 13]
'tp-1a201-26' : Tri local des résultats du reduce terminé.
'tp-1a201-26' : Quantiles définis : [1, 1, 1, 1, 1, 1, 1, 2, 2, 3, 5, 13]
'tp-1a201-16' : Tri local des résultats du reduce terminé.
'tp-1a201-16' : Quantiles définis : [1, 1, 1, 1, 1, 1, 1, 2, 2, 3, 6, 13]
'tp-1a201-31' : Tri local des résultats du reduce terminé.
'tp-1a201-31' : Quantiles définis : [1, 1, 1, 1, 1, 1, 1, 2, 2, 3, 5, 12]
'tp-1a201-05' : Tri local des résultats du reduce terminé.
'tp-1a201-05' : Quantiles définis : [1, 1, 1, 1, 1, 1, 1, 2, 2, 4, 6, 13]
'tp-1a201-10' : Tri local des résultats du reduce terminé.
'tp-1a201-10' : Quantiles définis : [1, 1, 1, 1, 1, 1, 1, 2, 2, 3, 6, 13]
'tp-1a201-20' : Tri local des résultats du reduce terminé.
'tp-1a201-20' : Quantiles définis : [1, 1, 1, 1, 1, 1, 1, 2, 2, 3, 6, 13]
'tp-1a201-21' : Quantiles envoyés au client
'tp-1a201-26' : Quantiles envoyés au client
'tp-1a201-16' : Quantiles envoyés au client
'tp-1a201-24' : Tri local des résultats du reduce terminé.
'tp-1a201-24' : Quantiles définis : [1, 1, 1, 1, 1, 1, 1, 2, 2, 3, 5, 13]
'tp-1a201-31' : Quantiles envoyés au client
'tp-1a201-05' : Quantiles envoyés au client
'tp-1a201-10' : Quantiles envoyés au client
'tp-1a201-20' : Quantiles envoyés au client
'tp-1a201-25' : Tri local des résultats du reduce terminé.
'tp-1a201-25' : Quantiles définis : [1, 1, 1, 1, 1, 1, 1, 2, 2, 3, 4, 10]
'tp-1a201-24' : Quantiles envoyés au client
'tp-1a201-25' : Quantiles envoyés au client
'tp-1a201-32' : Tri local des résultats du reduce terminé.
'tp-1a201-32' : Quantiles définis : [1, 1, 1, 1, 1, 1, 1, 2, 2, 4, 6, 14]
'tp-1a201-32' : Quantiles envoyés au client
'tp-1a201-08' : Tri local des résultats du reduce terminé.
'tp-1a201-08' : Quantiles définis : [1, 1, 1, 1, 1, 1, 1, 2, 2, 3, 6, 13]
'tp-1a201-08' : Quantiles envoyés au client
'tp-1a201-30' : Tri local des résultats du reduce terminé.
'tp-1a201-30' : Quantiles définis : [1, 1, 1, 1, 1, 1, 1, 2, 2, 3, 5, 11]
'tp-1a201-23' : Tri local des résultats du reduce terminé.
'tp-1a201-23' : Quantiles définis : [1, 1, 1, 1, 1, 1, 1, 2, 2, 3, 6, 14]
'tp-1a201-30' : Quantiles envoyés au client
'tp-1a201-23' : Quantiles envoyés au client
'tp-1a201-30' : Seuils globaux reçus : [1, 1, 1, 1, 1, 1, 1, 2, 2, 3, 4, 10]
'tp-1a201-23' : Seuils globaux reçus : [1, 1, 1, 1, 1, 1, 1, 2, 2, 3, 4, 10]
'tp-1a201-08' : Seuils globaux reçus : [1, 1, 1, 1, 1, 1, 1, 2, 2, 3, 4, 10]
'tp-1a201-32' : Seuils globaux reçus : [1, 1, 1, 1, 1, 1, 1, 2, 2, 3, 4, 10]
'tp-1a201-05' : Seuils globaux reçus : [1, 1, 1, 1, 1, 1, 1, 2, 2, 3, 4, 10]
'tp-1a201-31' : Seuils globaux reçus : [1, 1, 1, 1, 1, 1, 1, 2, 2, 3, 4, 10]

```

```

'tp-1a201-20' : Seuils globaux reçus : [1, 1, 1, 1, 1, 1, 1, 2, 2, 3, 4, 10]
'tp-1a201-16' : Seuils globaux reçus : [1, 1, 1, 1, 1, 1, 1, 2, 2, 3, 4, 10]
'tp-1a201-21' : Seuils globaux reçus : [1, 1, 1, 1, 1, 1, 1, 2, 2, 3, 4, 10]
'tp-1a201-32' : Paires lues depuis le fichier de résultats du reduce.
'tp-1a201-21' : Paires lues depuis le fichier de résultats du reduce.
'tp-1a201-23' : Paires lues depuis le fichier de résultats du reduce.
'tp-1a201-16' : Paires lues depuis le fichier de résultats du reduce.
'tp-1a201-26' : Paires lues depuis le fichier de résultats du reduce.
'tp-1a201-08' : Paires lues depuis le fichier de résultats du reduce.
'tp-1a201-20' : Paires lues depuis le fichier de résultats du reduce.
'tp-1a201-05' : Paires lues depuis le fichier de résultats du reduce.
'tp-1a201-10' : Paires lues depuis le fichier de résultats du reduce.
'tp-1a201-31' : Paires lues depuis le fichier de résultats du reduce.
'tp-1a201-24' : Paires lues depuis le fichier de résultats du reduce.
'tp-1a201-25' : Paires lues depuis le fichier de résultats du reduce.
'tp-1a201-30' : Paires lues depuis le fichier de résultats du reduce.
'tp-1a201-32' : Paires triées selon les seuils et réparties équitablement.
'tp-1a201-32' : Fichier de résultats du shuffle2 créé : /home/users/adeformel-24/tri_shuffle2_results_tp-1a201-32_v11.txt
'tp-1a201-21' : Paires triées selon les seuils et réparties équitablement.
'tp-1a201-23' : Paires triées selon les seuils et réparties équitablement.
'tp-1a201-32' : Paires locales écrites dans le fichier de résultat.
'tp-1a201-32' : Début de l'envoi des paires clé-valeur pour la phase 5.
'tp-1a201-32' : Début de la réception des paires clé-valeur des autres machines pour la phase 5.
'tp-1a201-32' : Début du tri final
'tp-1a201-26' : Paires triées selon les seuils et réparties équitablement.
'tp-1a201-21' : Fichier de résultats du shuffle2 créé : /home/users/adeformel-24/tri_shuffle2_results_tp-1a201-21_v2.txt
'tp-1a201-16' : Paires triées selon les seuils et réparties équitablement.
'tp-1a201-23' : Fichier de résultats du shuffle2 créé : /home/users/adeformel-24/tri_shuffle2_results_tp-1a201-23_v8.txt
'tp-1a201-26' : Fichier de résultats du shuffle2 créé : /home/users/adeformel-24/tri_shuffle2_results_tp-1a201-26_v5.txt
'tp-1a201-21' : Paires locales écrites dans le fichier de résultat.
'tp-1a201-21' : Début de l'envoi des paires clé-valeur pour la phase 5.
'tp-1a201-21' : Début de la réception des paires clé-valeur des autres machines pour la phase 5.
'tp-1a201-21' : Début du tri final
'tp-1a201-08' : Paires triées selon les seuils et réparties équitablement.
'tp-1a201-20' : Paires triées selon les seuils et réparties équitablement.
'tp-1a201-16' : Fichier de résultats du shuffle2 créé : /home/users/adeformel-24/tri_shuffle2_results_tp-1a201-16_v1.txt
'tp-1a201-23' : Paires locales écrites dans le fichier de résultat.
'tp-1a201-23' : Début de l'envoi des paires clé-valeur pour la phase 5.
'tp-1a201-23' : Début de la réception des paires clé-valeur des autres machines pour la phase 5.
'tp-1a201-23' : Début du tri final
'tp-1a201-26' : Paires locales écrites dans le fichier de résultat.
'tp-1a201-26' : Début de l'envoi des paires clé-valeur pour la phase 5.
'tp-1a201-26' : Début de la réception des paires clé-valeur des autres machines pour la phase 5.
'tp-1a201-26' : Début du tri final
'tp-1a201-10' : Paires triées selon les seuils et réparties équitablement.
'tp-1a201-05' : Paires triées selon les seuils et réparties équitablement.
'tp-1a201-08' : Fichier de résultats du shuffle2 créé : /home/users/adeformel-24/tri_shuffle2_results_tp-1a201-08_v9.txt
'tp-1a201-20' : Fichier de résultats du shuffle2 créé : /home/users/adeformel-24/tri_shuffle2_results_tp-1a201-20_v3.txt
'tp-1a201-16' : Paires locales écrites dans le fichier de résultat.
'tp-1a201-16' : Début de l'envoi des paires clé-valeur pour la phase 5.
'tp-1a201-16' : Début de la réception des paires clé-valeur des autres machines pour la phase 5.
'tp-1a201-16' : Début du tri final

```