Parallélisation de l'inférence de domaines protéiques.

Clément Rezvoy
encadrement :
Frédéric Vivien Daniel Kahn

Laboratoire de l'Informatique du Parallélisme

Laboratoire de Biométrie et Biologie Évolutive

21 Juin 2007

Domaines protéiques

Domaine protéique : sous-partie structurale d'une protéine.

On regroupe en famille des domaines :

- adoptant une structure similaire
- présentant des similitudes de séquence

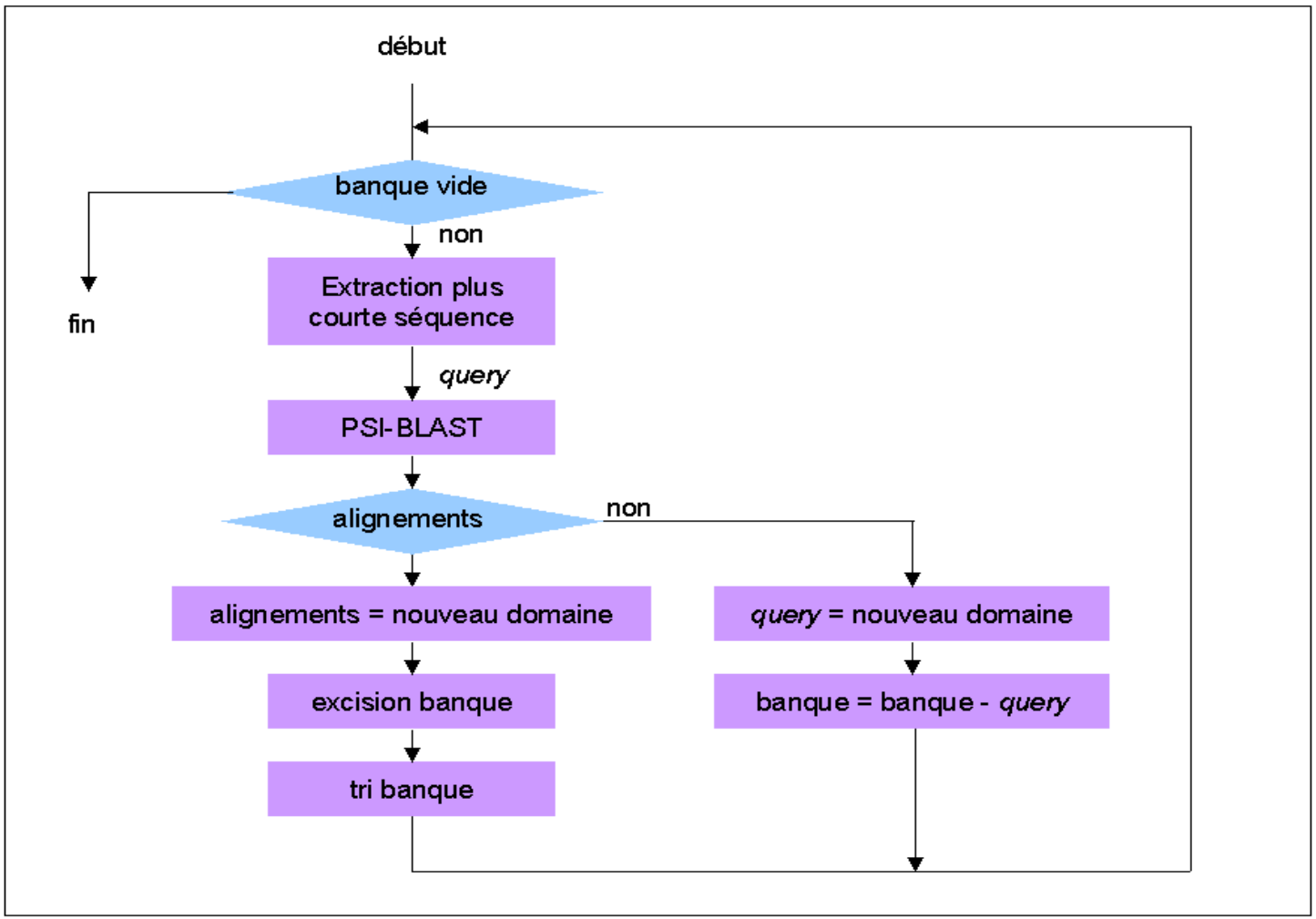
Utilisé pour :

- prédiction de propriétés
- étude de l'évolution d'une protéine
- *etc*

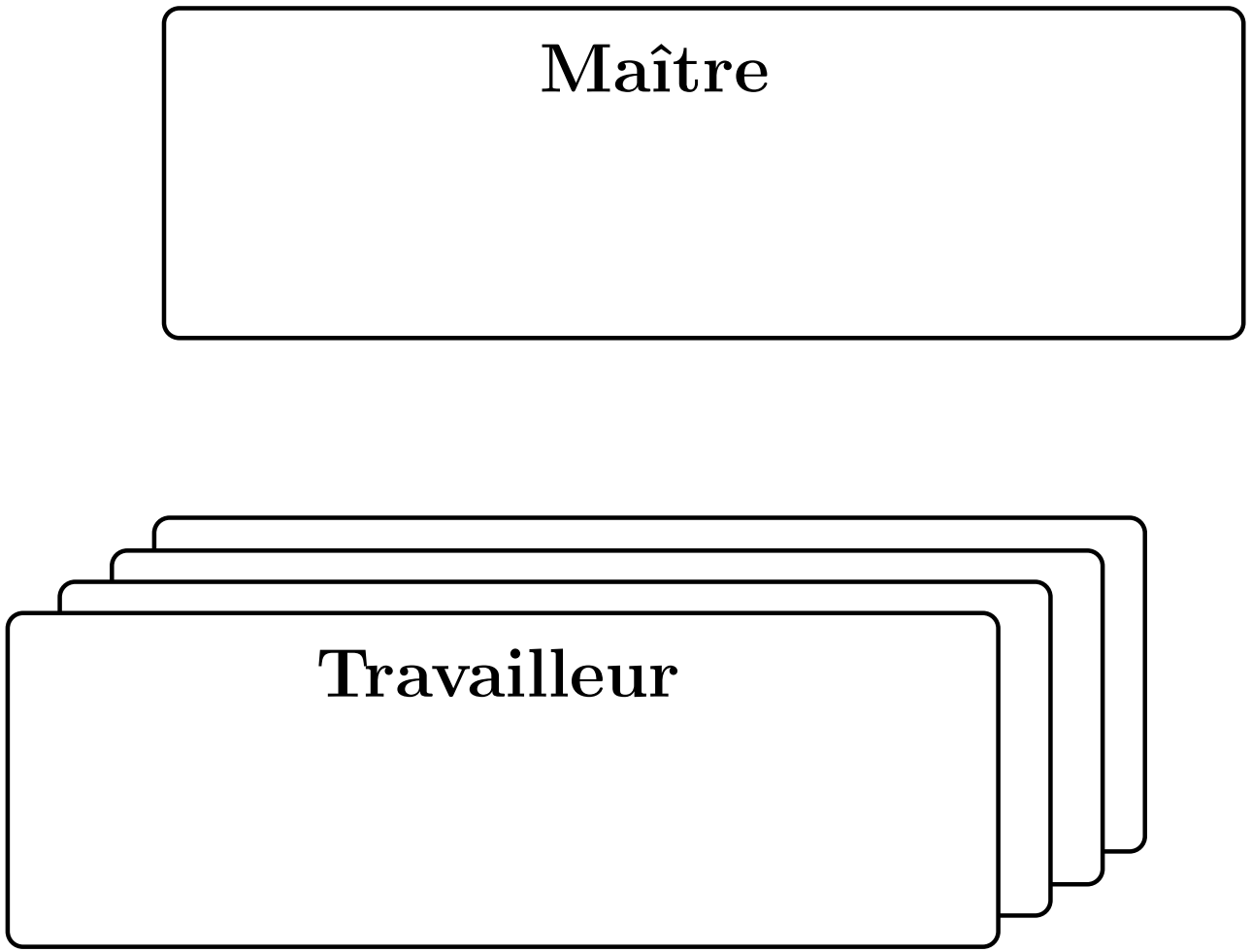
La base de données ProDom

- Base de données de familles de domaines protéiques
- Construite automatiquement par comparaisons de séquences
- À partir des séquences contenues dans Uniprot par l'algorithme MkDom2
- La complexité de MkDom2 est $\Theta(n^2)$
- Le temps de calcul augmente drastiquement :
  - 6 mois de calcul pour la version 2005.1
  - plus d'un an de calcul pour la version 2006.1 (en cours)
- Le temps de calcul séquentiel est devenu prohibitif

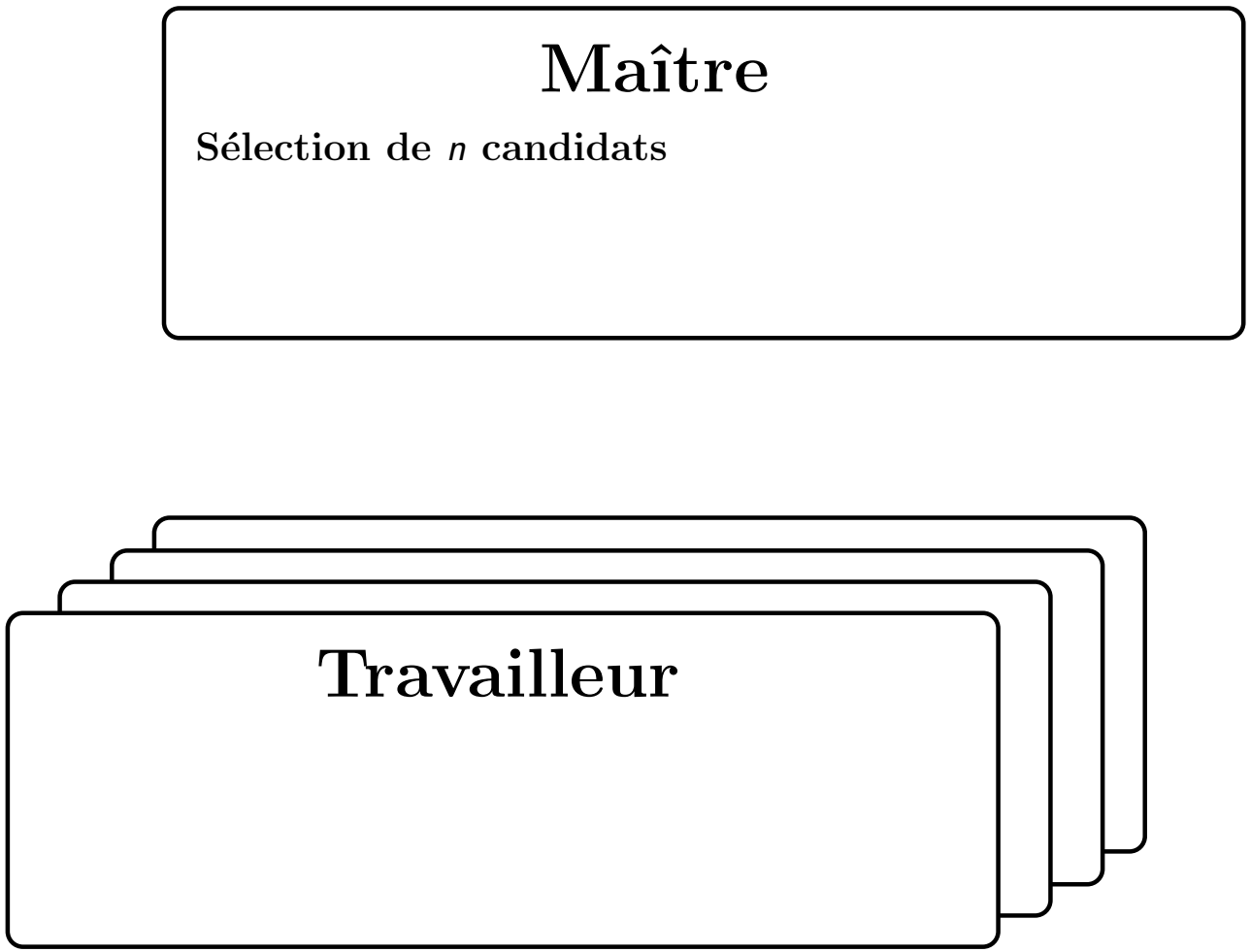
L'algorithme MkDom2 (Gouzy *et al.* 99)



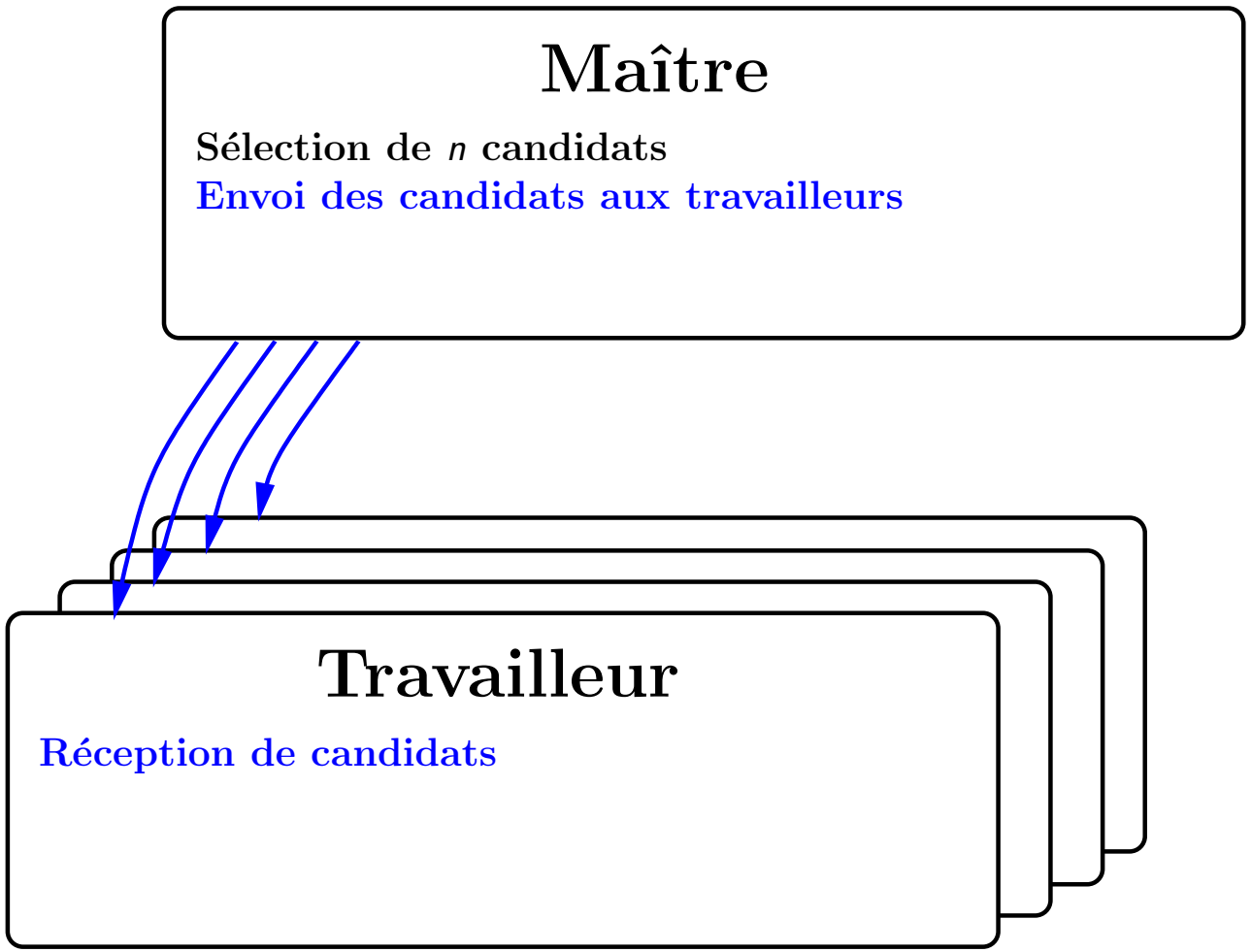
Parallélisation « maître-travailleurs »



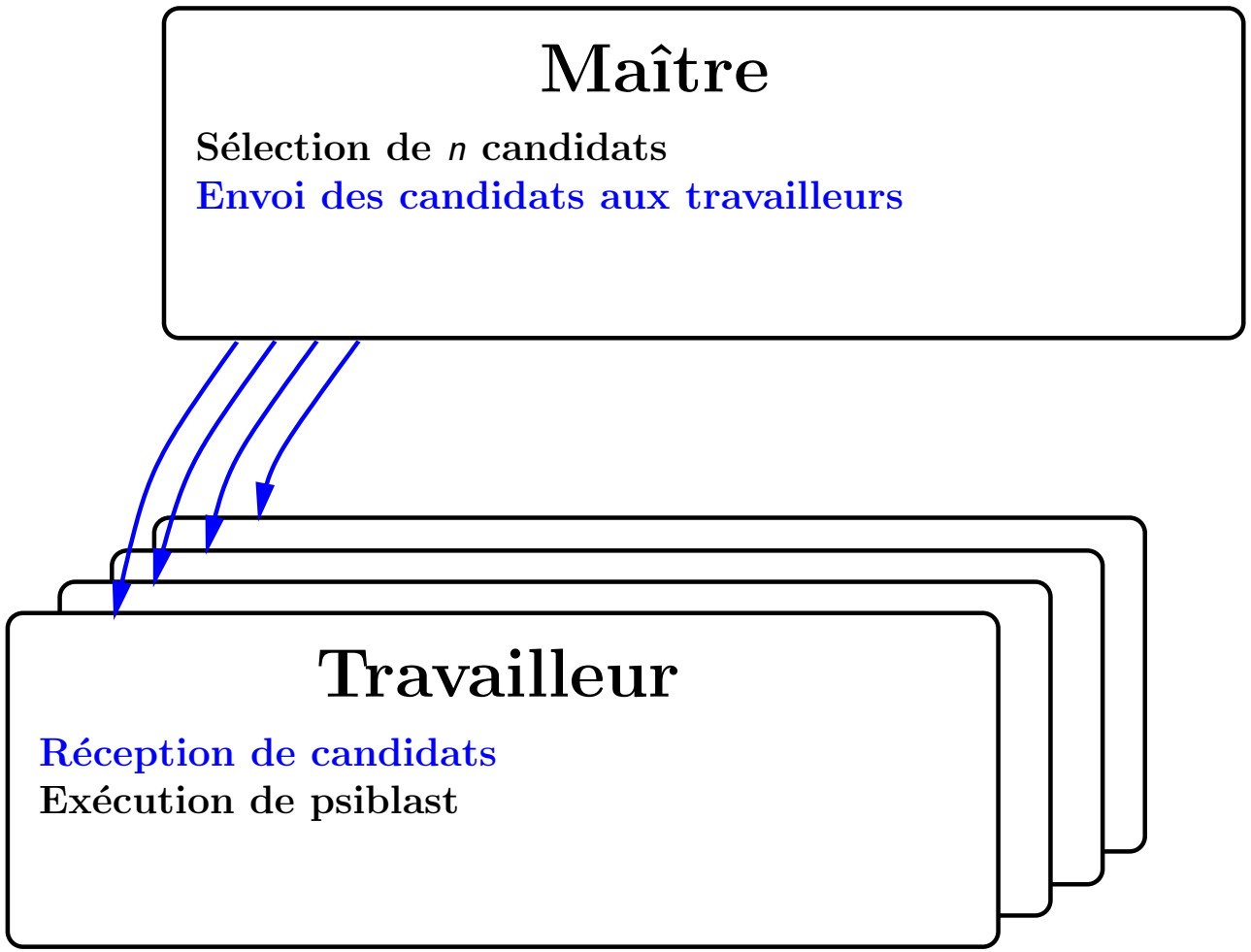
Parallélisation « maître-travailleurs »



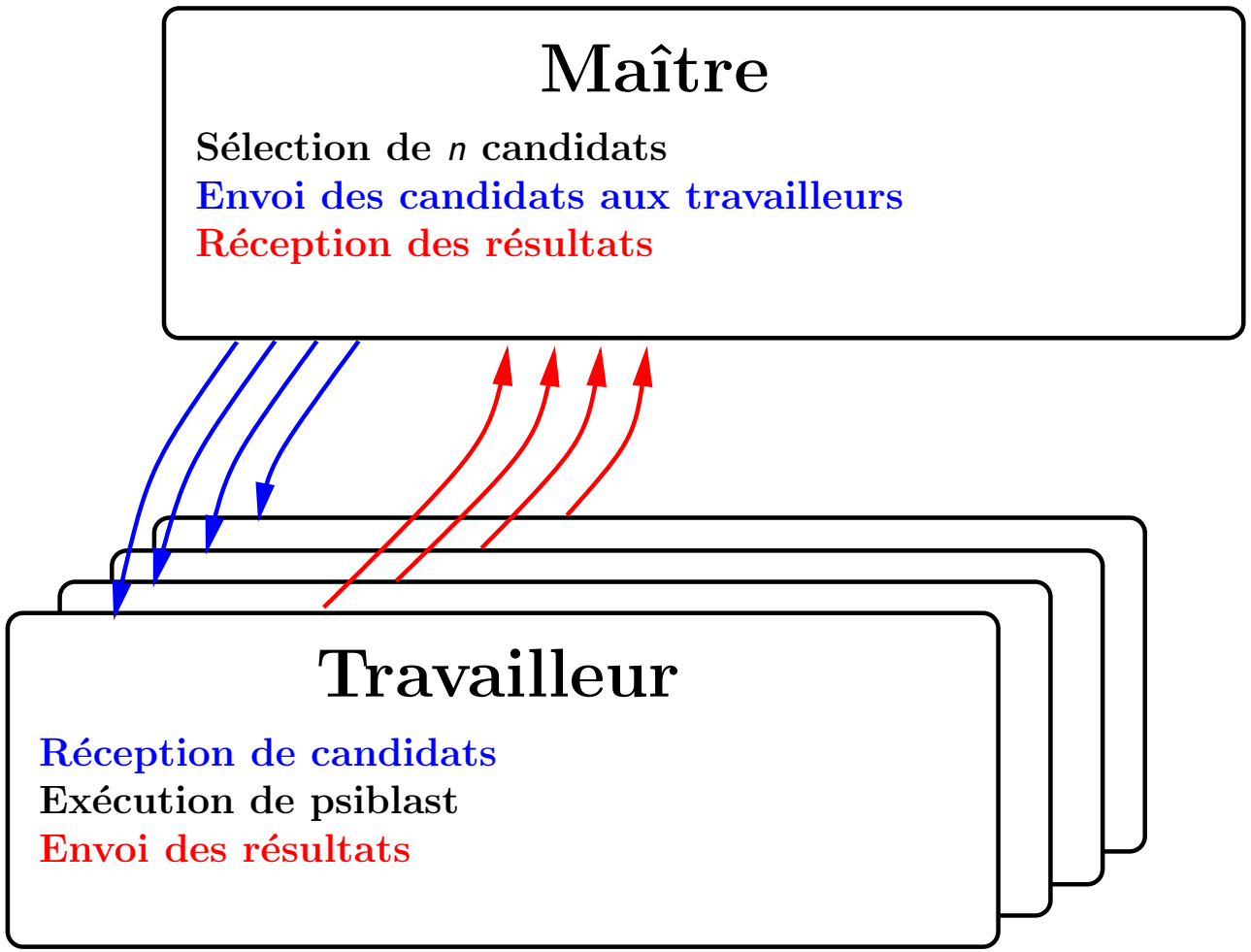
Parallélisation « maître-travailleurs »



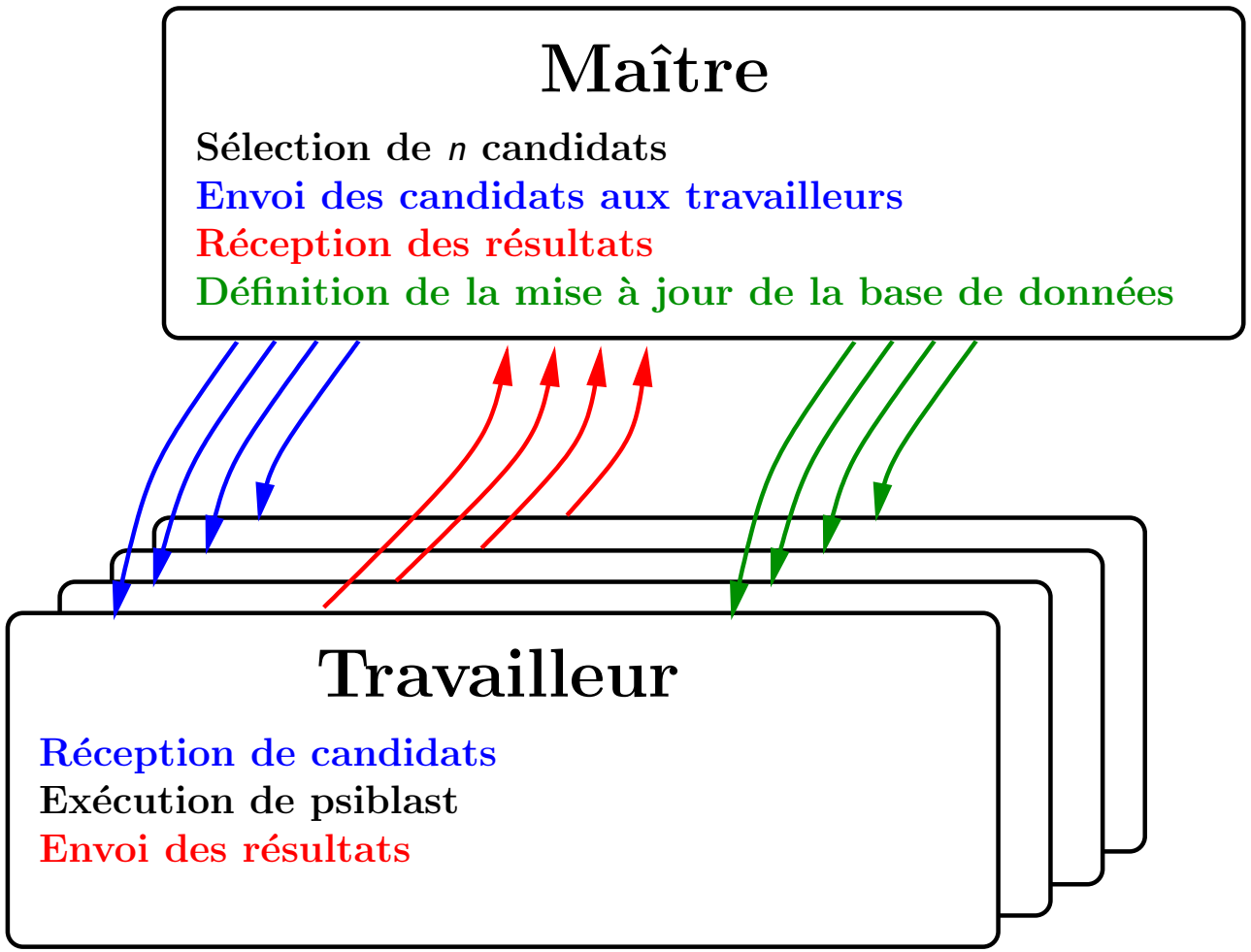
Parallélisation « maître-travailleurs »



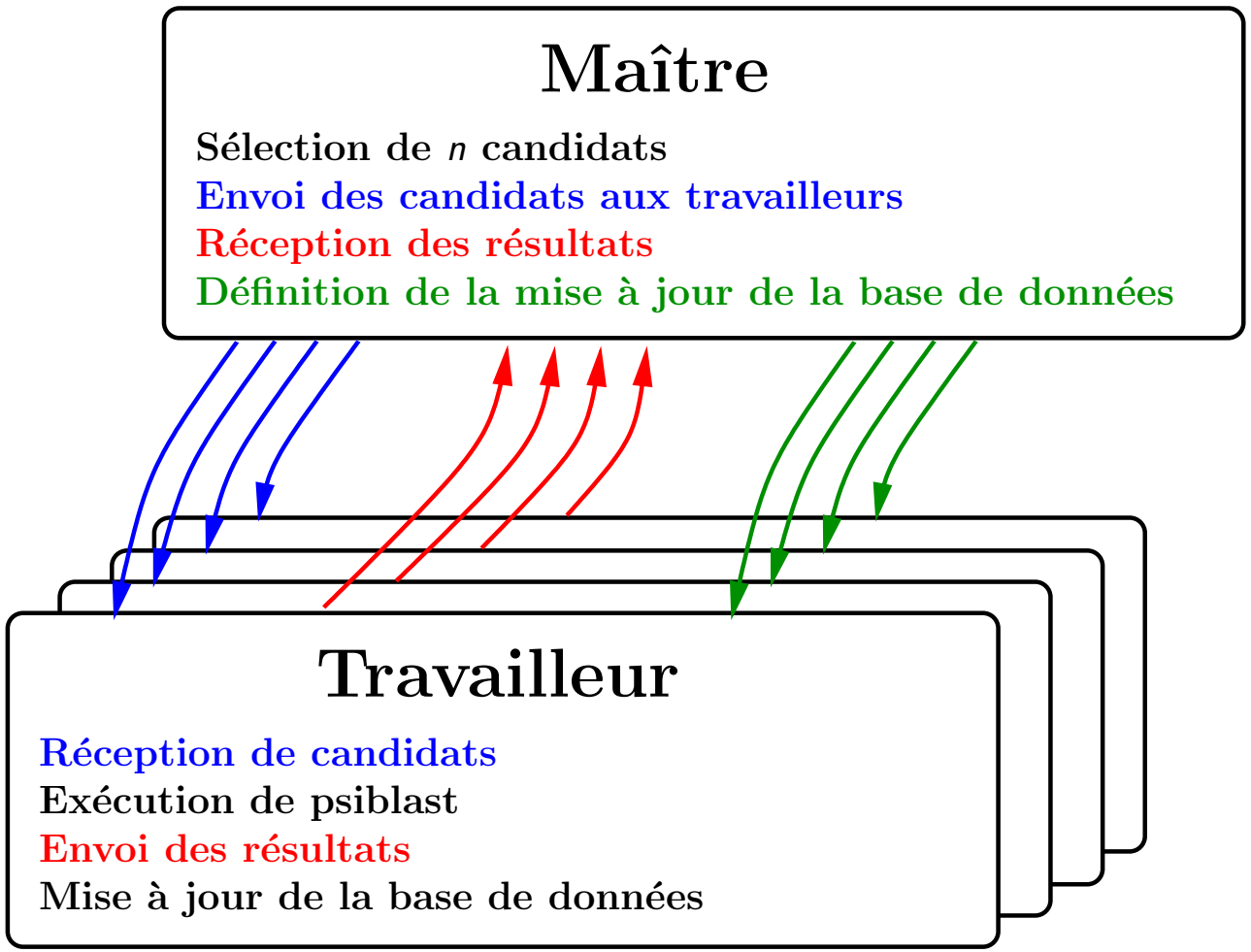
Parallélisation « maître-travailleurs »



Parallélisation « maître-travailleurs »

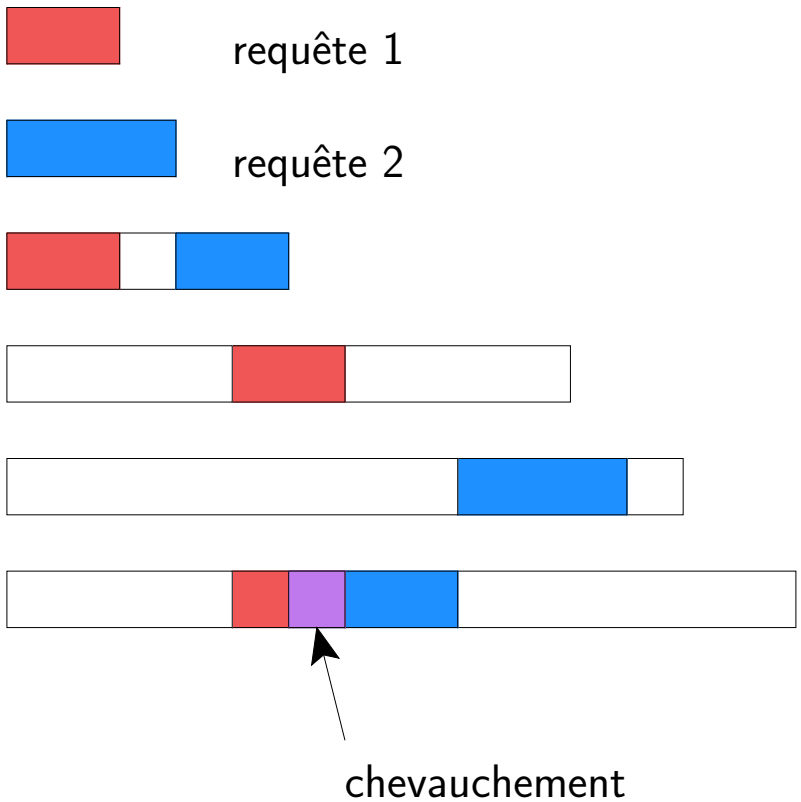


Parallélisation « maître-travailleurs »



Éviter les conflits entre requêtes

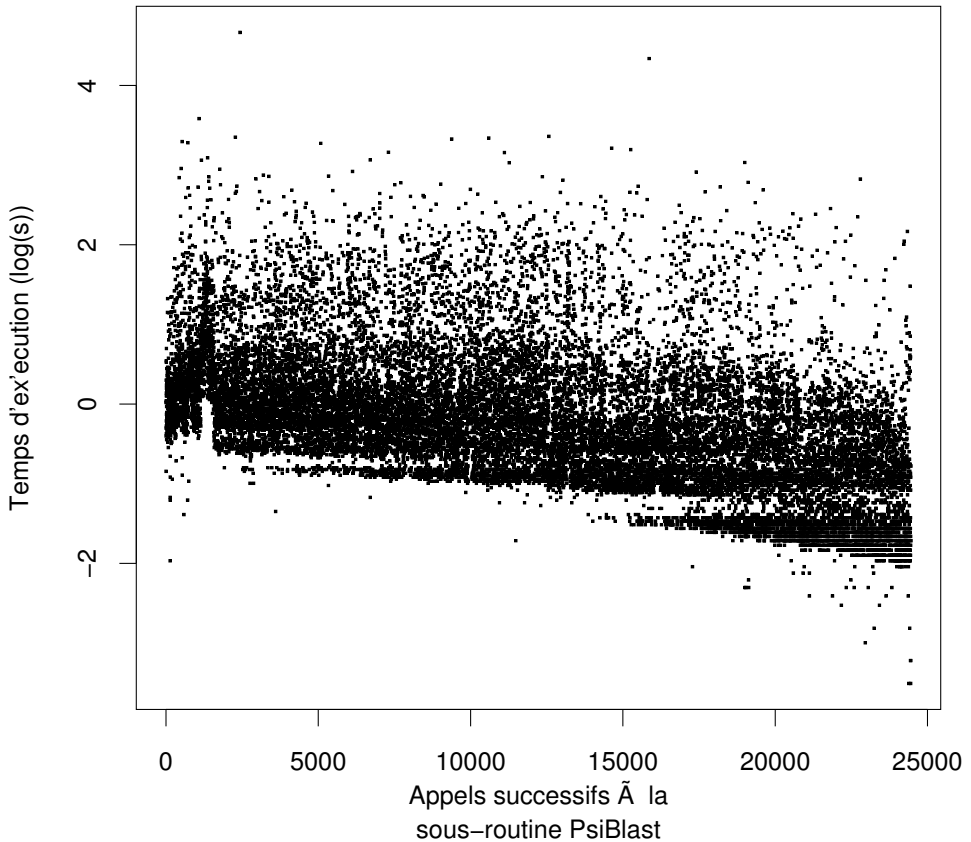
- Un résultat conflictuel doit être recalculé
- Prévoir les conflits pour les éviter
- Vérifications de l'indépendance des résultats *a posteriori*



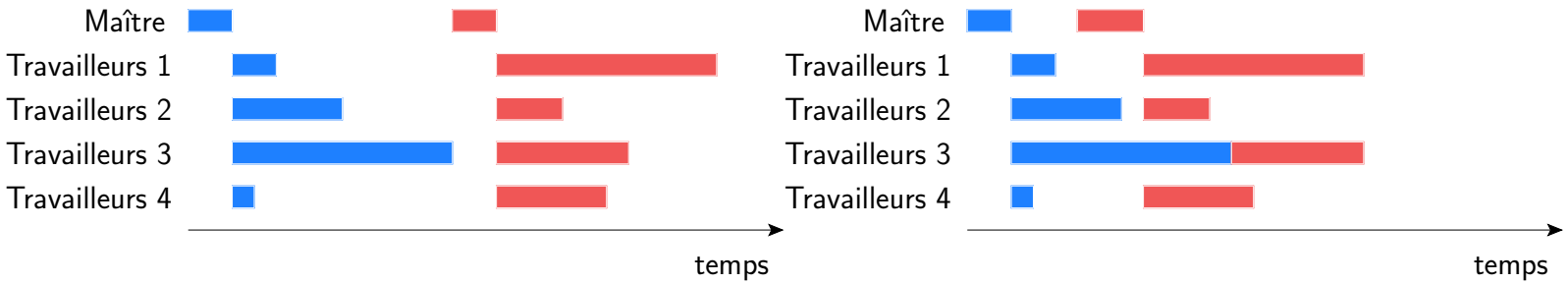
Équilibrage de charge entre travailleurs

Le temps d'exécution d'une recherche PSI-BLAST varie en fonction :

- de la puissance du processeur
- de la taille de la base de données
- du nombre d'itérations PSI-BLAST

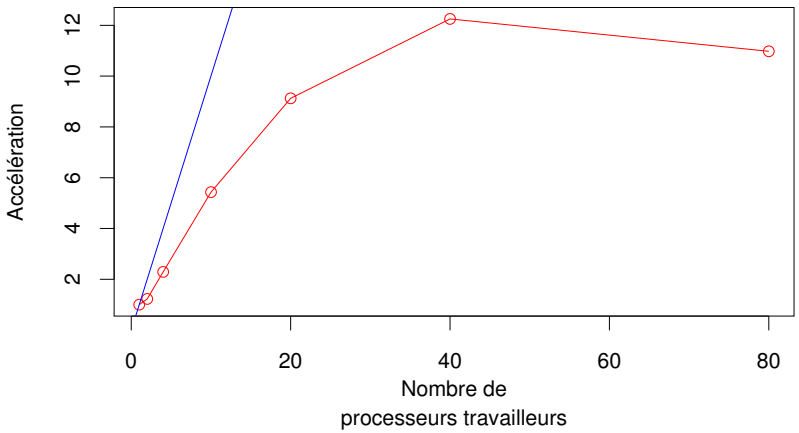


Optimisations

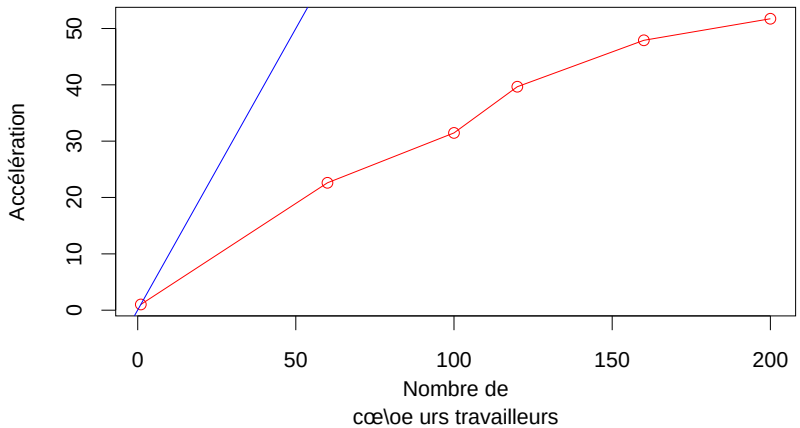


- Désynchroniser le travail du maître
- Décharger le plus possible le maître
- Limiter les accès disques
- Limiter les communications

Accélération observée



- Protéome complet de *C. elegans* ($\approx$ 11 Mo)
- 1 processeur : 4 h 22 min
- 40 processeurs : 18 min



- Protéomes complets eukaryotes ($\approx$ 92 Mo)
- 1 processeur : 6 j 8 h
- 200 cœurs : 2 h 56 min

Résultats qualitatifs

- Les résultats diffèrent de la version séquentielle.
  - Gestion des répétitions internes
  - Gestion des erreurs
  - Versions des programmes
- Les conflits sont peu fréquents ($< 1\%$ pour le protéome de *C. elegans* , $< 4\%$ sur l'ensemble des protéomes eukaryotes)

Conclusions

- La parallélisation s'avère efficace pour réduire le temps de calcul de MkDom2
- L'occurrence de conflits entre résultats est peu fréquente
- Le coût de la parallélisation est limité
- L'efficacité augmente avec la taille des données à traiter

À court terme :

- Affiner l'algorithme et le tester à plus grande échelle

À plus long terme :

- La complexité de l'algorithme parallèle est toujours en $\Theta(n^2)$
- Il faudra définir un nouvel algorithme permettant un traitement exhaustif à grande échelle des gros volumes de données à venir.