

Apports du couplage et de la monotonie dans la simulation temps parallèle

Jean-Michel Fourneau, Maël Guiraud, Yann Strozecki

DAVID, Université de Versailles Saint Quentin

June 26, 2018



Introduction Simulation temps parallèle

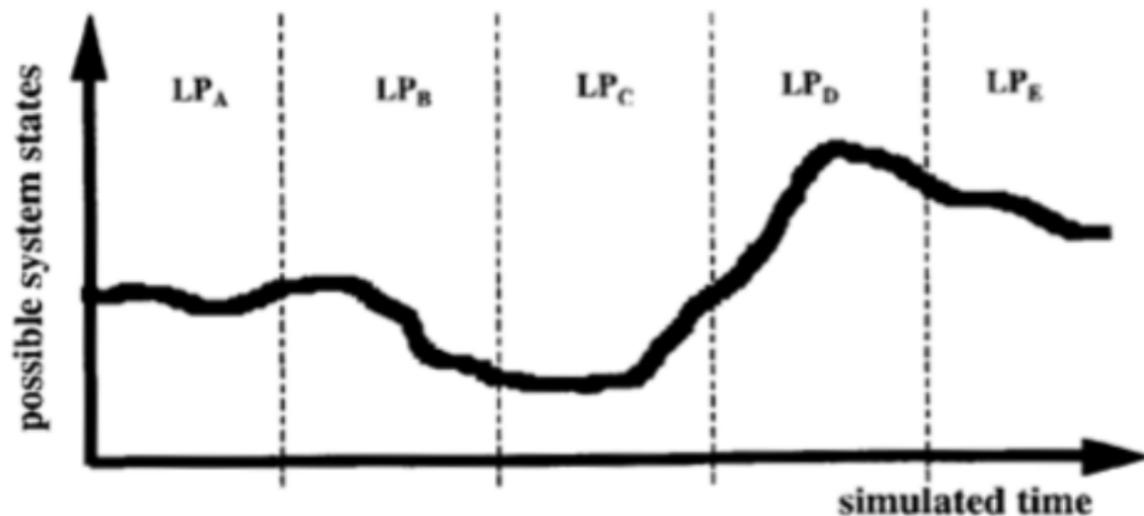


Figure 6.2 Sample path for a simulation computation.

Richard M Fujimoto. *Parallel and distributed simulation systems*. Vol. 300.
Wiley New York, 2000

Introduction Simulation temps parallèle

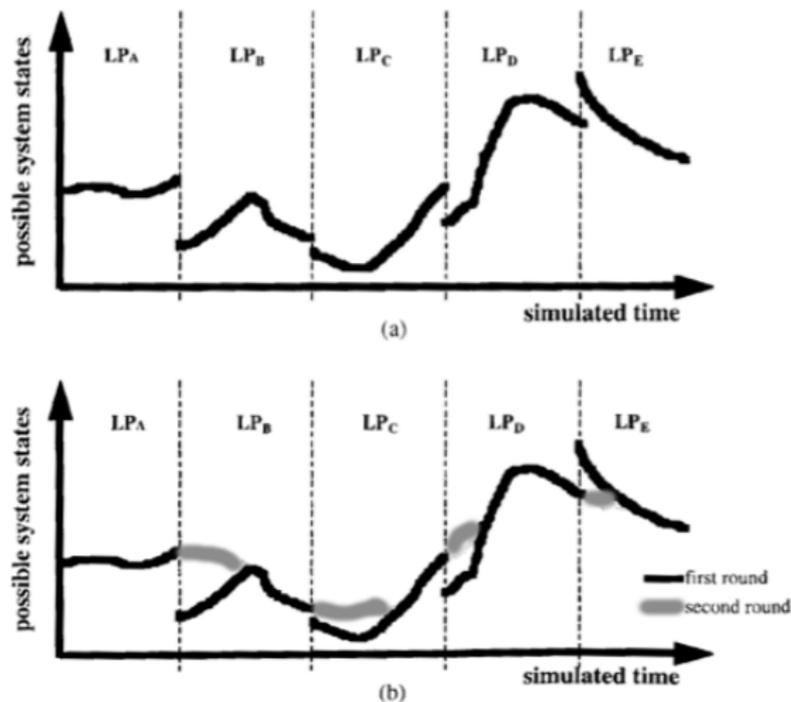


Figure 6.3 Time parallel simulation using fix-up computations. (a) Computation performed in first round; (b) fix-up computation in second round.

Hypothèse sur les états

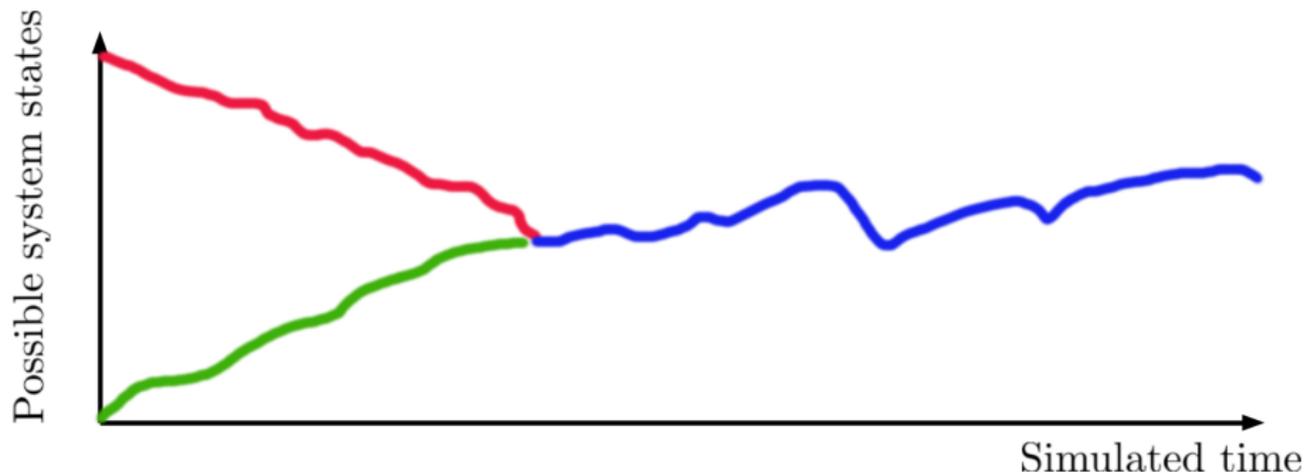
Pour améliorer la simulation temps parallèle , on fait les suppositions suivantes:

- Un ordre partiel rapidement calculable sur l'ensemble des états
- Une monotonie de la dynamique de la chaîne de Markov
- Un état minimum et un état maximum

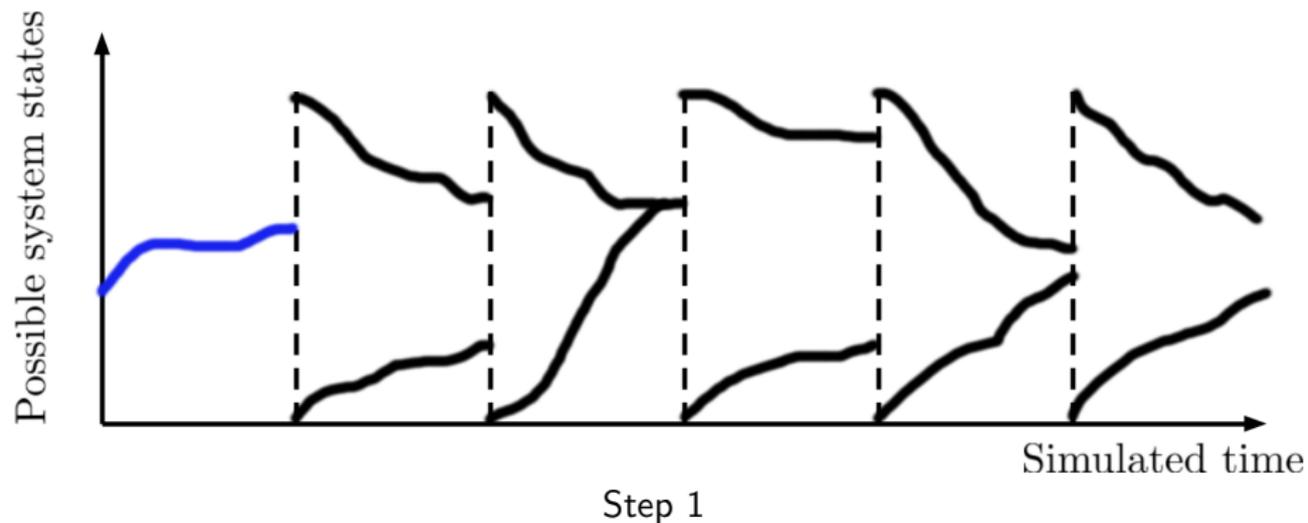
Algorithme à deux bornes (Speculative sandwich)

Idée de l'algorithme :

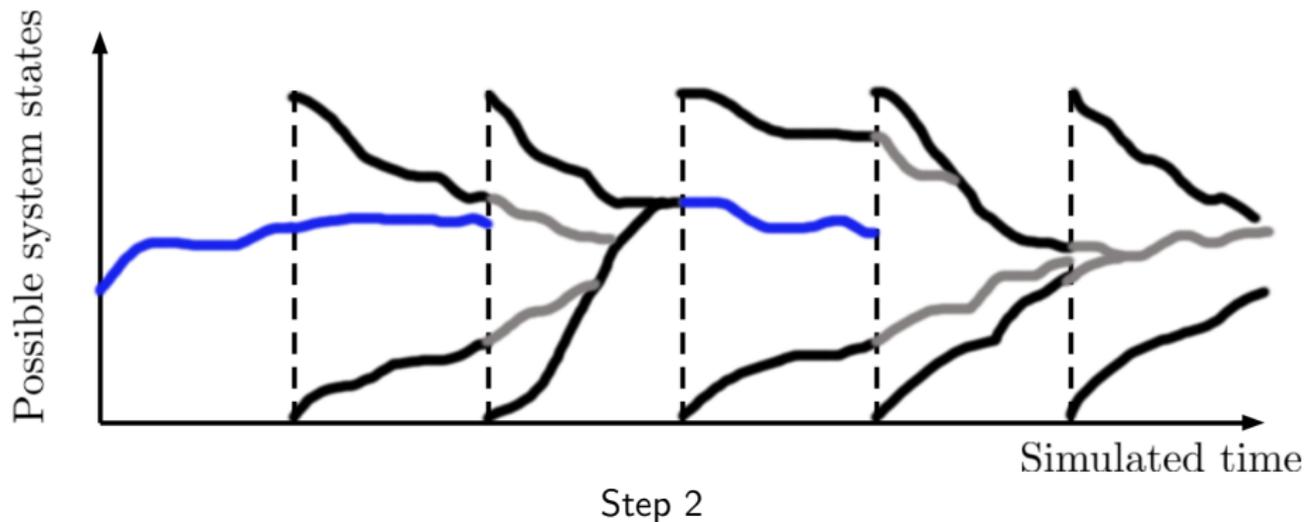
Utiliser une **borne inférieure** et une **borne supérieure** jusqu'au **couplage** des deux trajectoires.



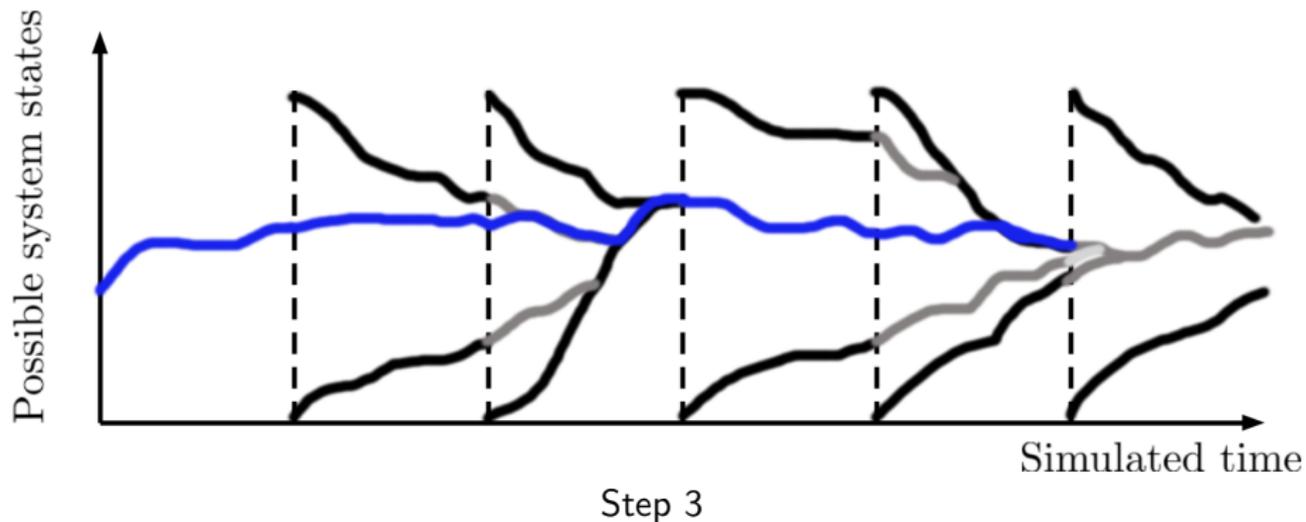
Algorithme à deux bornes (Speculative sandwich)



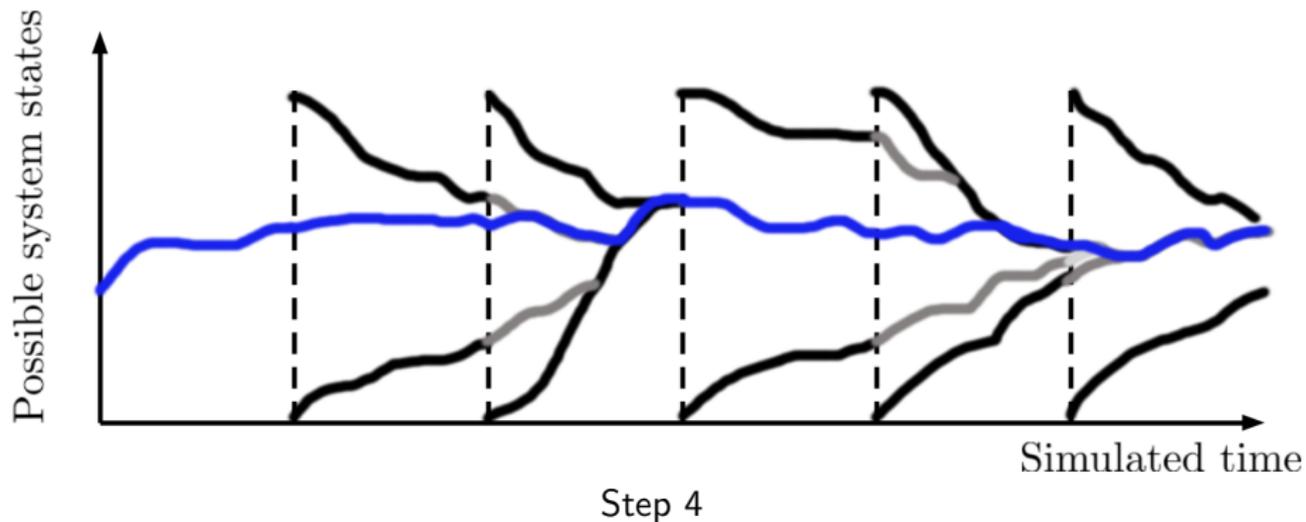
Algorithme à deux bornes (Speculative sandwich)



Algorithme à deux bornes (Speculative sandwich)



Algorithme à deux bornes (Speculative sandwich)



Variantes de l'algorithme :

- Choix de l'intervalle à traiter par un processeur :
 - Plus petit indice possible
 - Privilégier l'intervalle le plus en "retard" / le plus loin de coupler
 - Chaque processeur calcul la même zone d'intervalles
- Choix du nombre d'intervalles

Optimisations

- Ne pas calculer le dernier intervalle tant que l'avant dernier n'a pas couplé
- Ne pas recalculer un intervalle couplé, même si ses bornes de départs sont meilleures

Algorithme à une borne (One bound)

On comparera l'algorithme avec une seule ou deux bornes.

Utiliser une seule borne pour la simulation temps parallèle :

- Divise le temps de calcul d'un intervalle par deux
- Plus de couplage, mais monotonie augmente la chance que les trajectoires se croisent

Spécificités technique

Implémentation de l'algorithme sur des raspberry-pi 3 b connectés en réseau local en ethernet

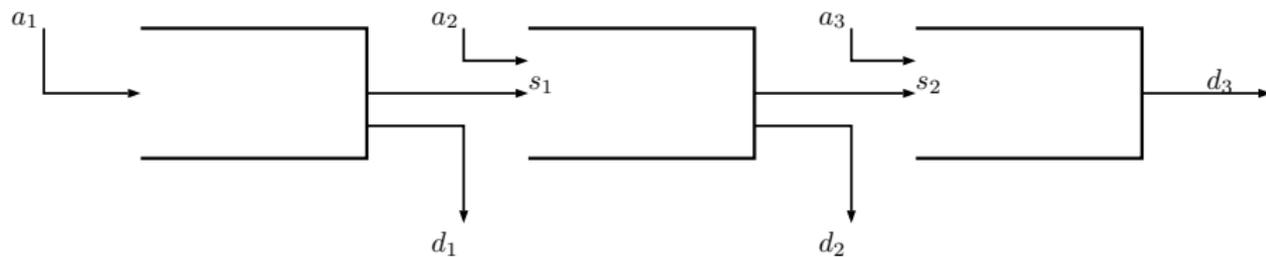
Protocole réseau

Utilisation des sockets en C, avec TCP/IP

→ Temps réseau coûte cher à cause de la taille des messages et de TCP

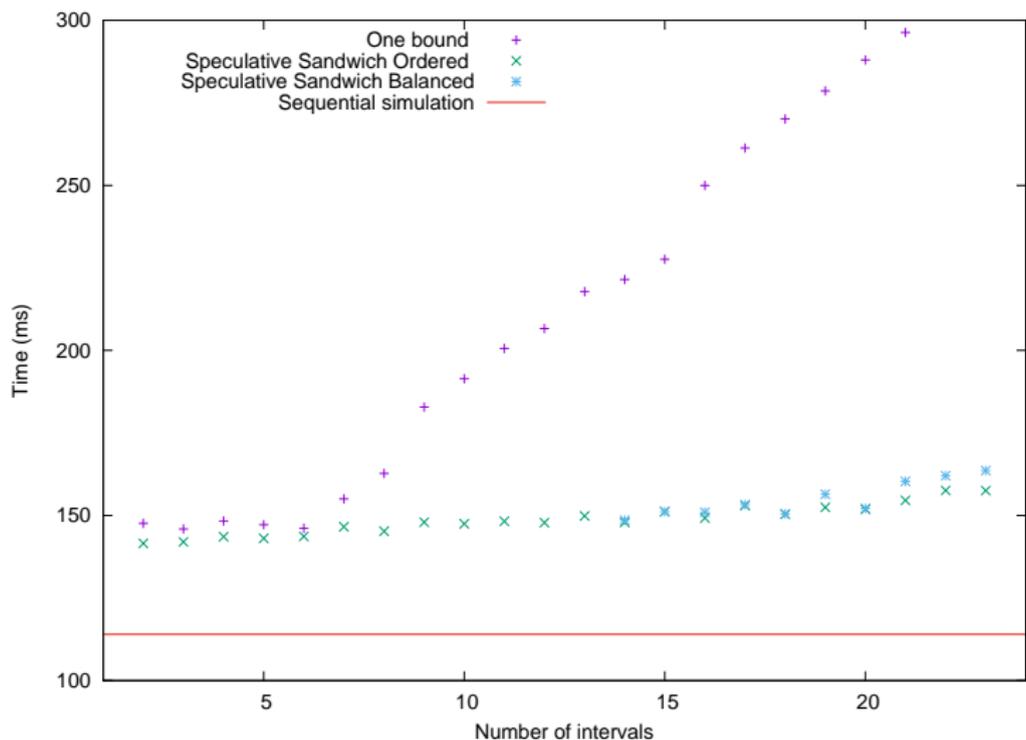
Résultats sur 7 esclaves et 1 maître.

Experiences sur un tandem de files d'attentes

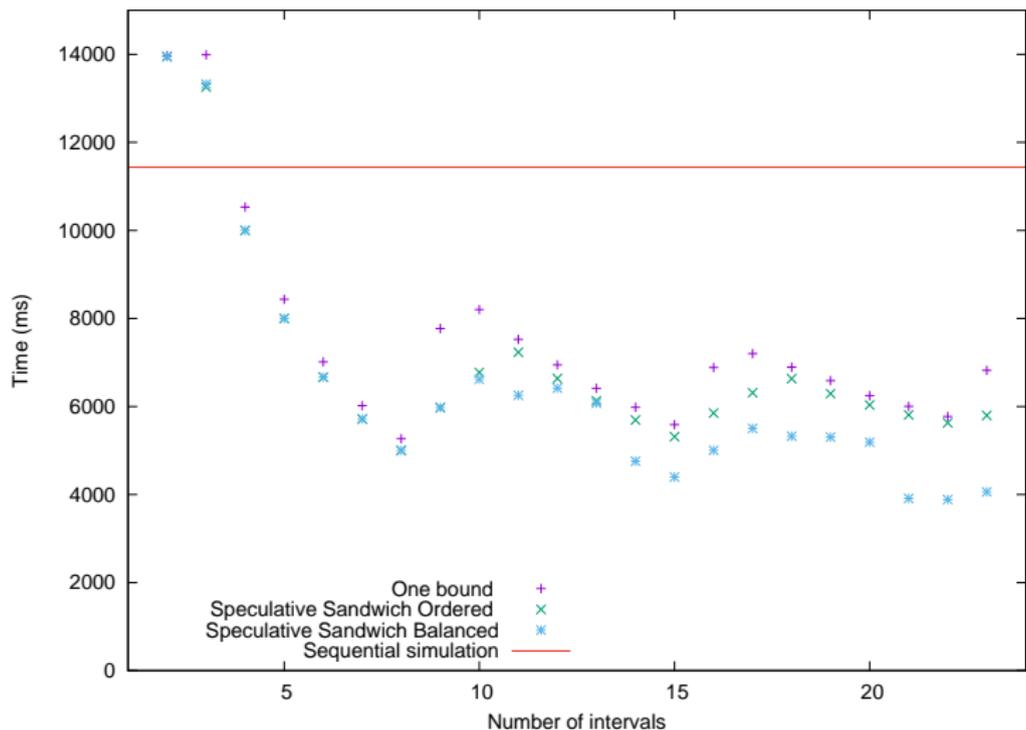


Les paramètres a_i , d_i et s_i et le nombre de files sont fixés de façon à avoir un grand temps de couplage.

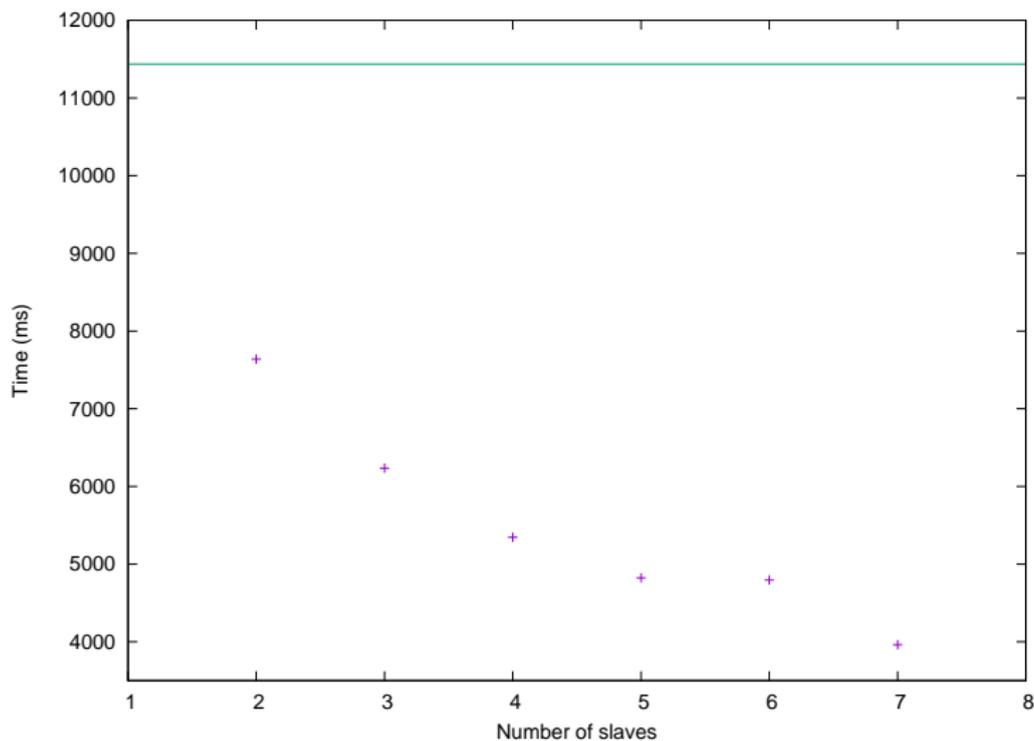
Résultats - Temps de couplage $>$ temps de simulation



Résultats - Temps de couplage \simeq taille d'un intervalle



Impact du nombre de processeurs



Utilisation d'une machine multiprocesseur et d'une implémentation OpenMP:

- On supprime les temps de communication réseau.
- Besoin de gérer les locks : coûteux.

Accélération de presque **le nombre de processeur** quand les deux bornes couplent très vite !

Conclusion

Algorithme plus performant que le séquentiel.

Programmé sur OpenMP et sur plusieurs machines.

Comment choisir les intervalles à calculer, estimer le nombre d'intervalles à choisir?

Merci de votre attention.