

# Un million d'atomes en chimie quantique

Anthony Scemama <[scemama@irsamc.ups-tlse.fr](mailto:scemama@irsamc.ups-tlse.fr)>

Mathias Rapacioli

Labratoire de Chimie et Physique Quantiques

IRSAMC (Toulouse)

Nicolas Renon

CALMIP (Toulouse)



# HPC pour la Chimie

Approches standard:

- Approches quantiques pour les molécules:  $O(N^3)$  -  $O(N!)$
- Méthodes semi-empiriques : méthodes quantiques paramétrées  $O(N^3)$
- Mécanique Moléculaire : méthodes Newtoniennes paramétrées,  $O(N^2)$

Traitement de gros systèmes (dizaines de milliers d'atomes):

- Pour toute méthode, il faut réduire le scaling : **linear scaling** ( $O(N)$ ).
- Les interactions électron-électron, électron-noyau et noyau-noyau sont *locales*

Le travail présenté ici:

- DFTB (Density Functional Tight Binding) : approche semi-empirique de la DFT (théorie de la fonctionnelle de la densité). Code **deMon-Nano**.
- Développement d'une méthode de calcul  $O(N)$  et implémentation OpenMP
- Résultats jusqu'à 1.5 millions d'atomes sur 376 cores, (~1.8 TB RAM, ~1.2 MB/atome)

# Algorithme

Hartree-Fock ou DFT: résolution de  $\mathbf{H C} = \mathbf{E S C}$

- H: matrice Hamiltonienne
- C: orbitales moléculaires (vecteurs)
- E: Énergie du système
- S: Matrice de recouvrement (base non-orthonormale)

Résolution de type  $AX = \lambda BX$  (routine DSYGV Lapack,  $O(N^3)$ )

H dépend de C, donc résolution *auto-cohérente* (SCF: self-consistent field).

- Préconditionnement de la matrice à diagonaliser par des produits de matrices creuses
- Produit de 2 matrices creuses: succession de petits produits dense x creux
- Diagonalisation partielle de H par une méthode Jacobi approchée
- La méthode converge vers les bons vecteurs/valeurs propres (voir résultats numériques à la fin).

# Implémentation

## Pourquoi OpenMP?

- La DFTB est une méthode paramétrée donc le calcul des matrices est très rapide. Une approche MPI aurait un rapport communication/calcul trop élevé
- Avec notre algorithme, il y aurait beaucoup de communications collectives de type *one-to-all* ou *all-to-one*
- Il est toujours possible d'utiliser le parallélisme massif ultérieurement en distribuant des calculs indépendants (Monte Carlo, dynamique, etc)

## Problèmes principaux

- Calcul:  $O(N^3)$  ->  $O(N)$ , Stockage:  $O(N^2)$  ->  $O(N)$
- Autant de calcul que d'accès mémoire, donc *memory-bound*
- Implémentation optimisée, donc peu de Flops et très dépendant des **latences mémoire**

# Latences mémoire

Sur Altix-UV : 24 blades bi-processeurs Intel Xeon E7-8837 @ 2.67 GHz (Westmere EX):

- 1 cycle CPU: 0.37 ns
- Au maximum 8 flops/cycle (SP), donc 1 flop ~ 0.047 ns
- L1 : 1.5 ns (4 cycles)
- L2 : 3.7 ns (10 cycles)
- L3 : 16 ns (43 cycles)
- RAM : 195 / 228 / 570 / 670 / 760 / 875 / 957 ns (520-2555 cycles)
- Accès réguliers à la mémoire : prefetch hardware
- **Le prefetch fonctionne entre blades** : comm. asynchrones

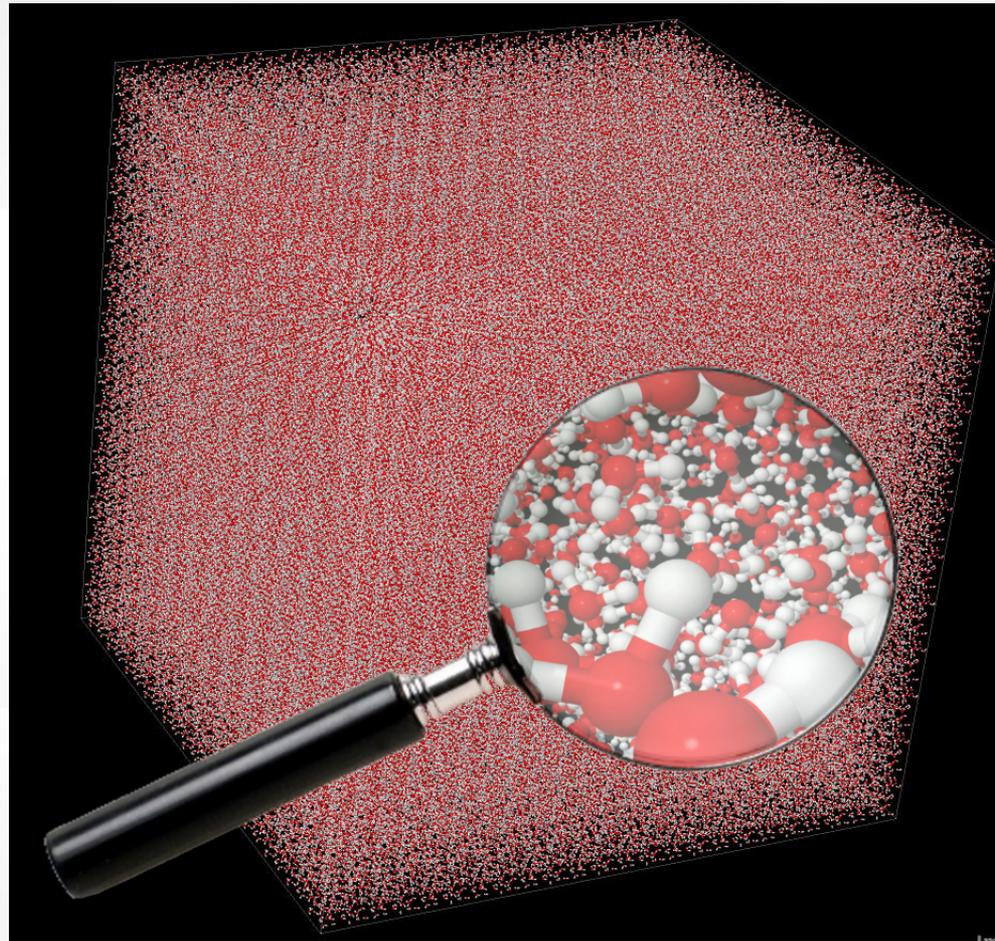
Autres:

- IB + MPI : 1070 ns
- Sur une machine Sandy Bridge : 86 / 155 ns

# Stratégie

- Objectif : Amener les données le plus vite possible sur les coeurs
- Réduire le plus possible le stockage, éventuellement re-calculer
- Placement des processus et politique d'allocation de mémoire : *omplace*
- Chaque processeur utilise le plus possible de la mémoire proche : allocations et initialisation des tableaux dans les blocs OpenMP
- Utilisation au maximum d'accès réguliers à la mémoire (stride 1) pour déclencher les prefetchers
- Réutiliser le plus possible les données qui sont dans les caches
- Synchroniser les threads le moins souvent possible
- Scheduling statique si possible pour garder l'accès local à la mémoire ou dynamique dans les cas de mauvais équilibrage

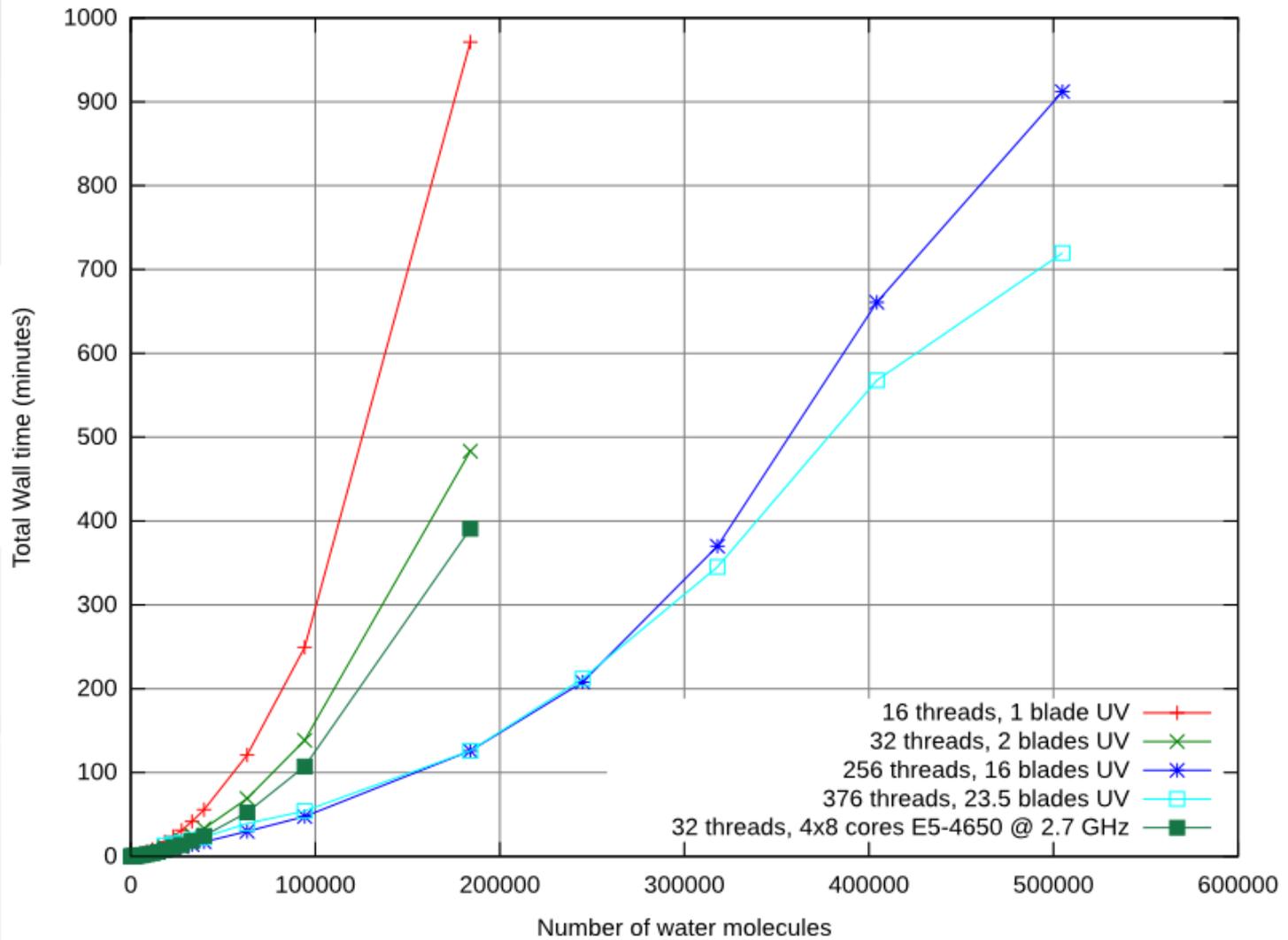
# Résultats



Comparaison de la précision numérique de la méthode en fonction du critère de convergence.

	$\epsilon = 10^{-5}$	$\epsilon = 10^{-6}$
E(Lapack)	-8996.7819247	-8996.7823133
E(deMon)	-8996.7819236	-8996.7823133
Erreur	1.2e-10	2.2e-12
t(Lapack) (s)	22710.1	25010.4
t(deMon) (s)	91.2	155.8

- Converge vers la bonne valeur
- Erreur de la méthode en dessous de l'erreur de convergence



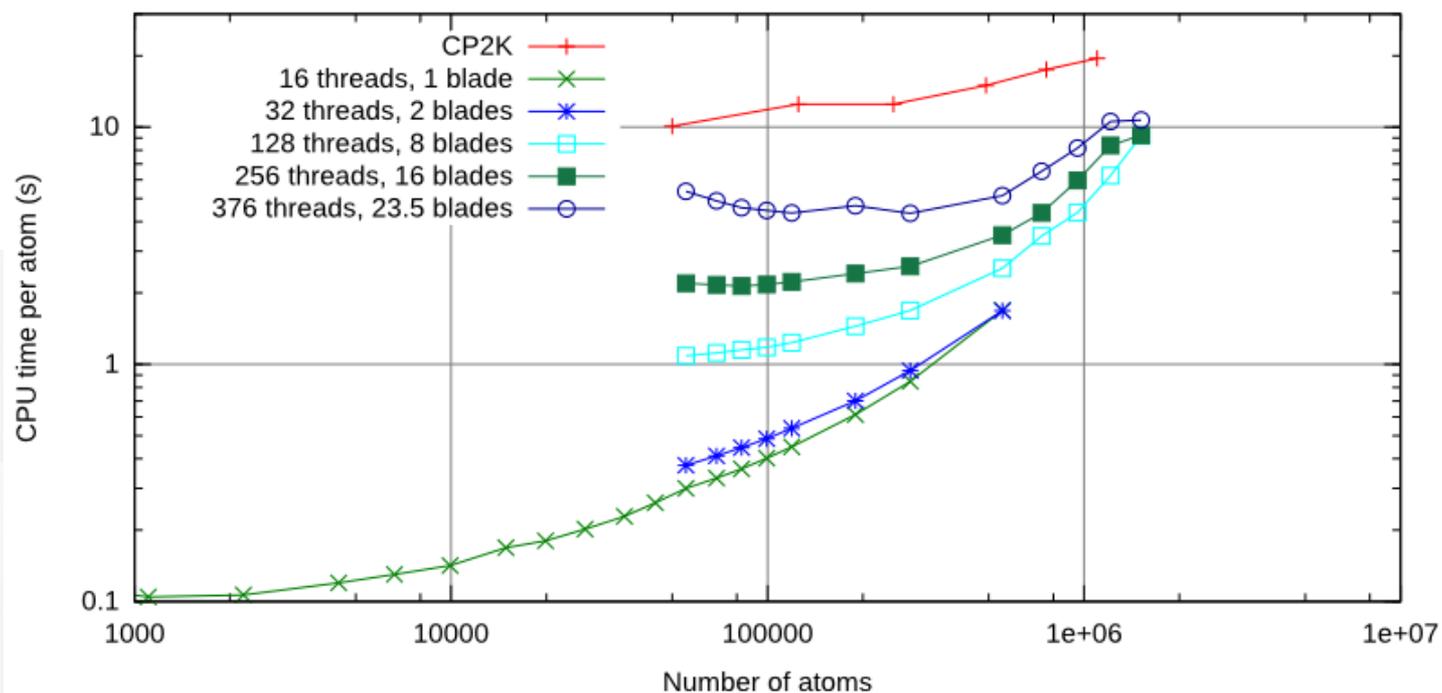
# Comparaison avec une implémentation MPI

Code CP2K, utilisant une bibliothèque d'algèbre linéaire creuse MPI:

**Linear Scaling Self-Consistent Field Calculations with Millions of Atoms in the Condensed Phase**

Joost VandeVondele, Urban Borštnik, and Jürg Hutter

*Journal of Chemical Theory and Computation* **8** (10), 3565-3573 (2012)



# Perspectives

Etude des épingles à cheveux dans des brins d'ADN dans l'eau avec une équipe du CEMES (N. Tarrat et al).

