

# Aran, une Librairie pour la Méthode des Multipôles

Pierre Gay

10 février 2014

# Plan

## Le problème des N-corps

Le problème des N-corps

## La méthode des Multipôles Rapide

La méthode des Multipôles Rapide

Détails

## La librairie Aran

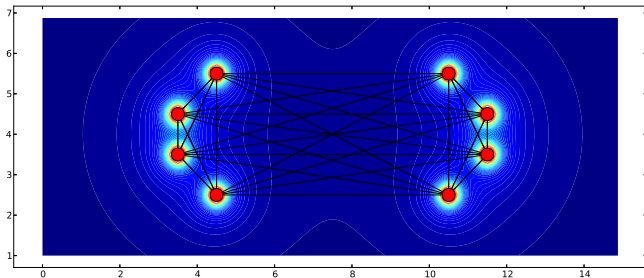
Implémentation

Parallélisme

## Etude de performances

## Conclusion

# Particules



- ▶ Des particules en grand nombre :  $N$
- ▶ Un champ / potentiel :  $\frac{1}{r^k}$ ,  $\frac{r^2 + \alpha^2}{(r^2 + \alpha^2)^{3/2}}$ ,  $\frac{e^{-k_0 r}}{r}$ , ...
- ▶ Des interactions directes : temps de calcul en  $O(N^2)$

## Domaines d'application

Le problème des N-corps est présent dans beaucoup de domaines scientifiques :

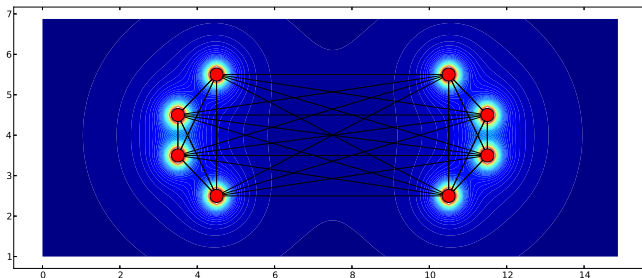
- ▶ Astrophysique (Spherical Particle Hydrodynamics)
- ▶ Chimie (dynamique moléculaire)
- ▶ Mécanique des fluides (Méthode des Vortex)
- ▶ Equations intégrales (Helmholtz, Maxwell, ...)
- ▶ ...

## Méthodes d'accélération

Il y a beaucoup de méthodes pour s'affranchir de la barre des  $O(N^2)$  :

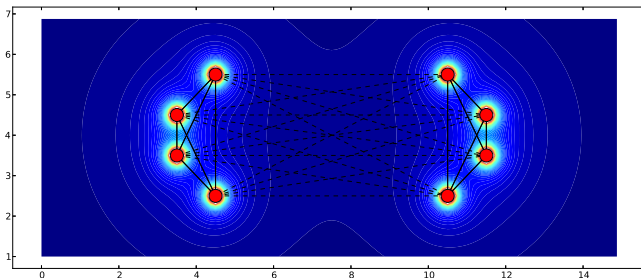
- ▶ Tree codes - Barnes & Hut
- ▶ Fast Fourier Transform
- ▶ **Fast Multipole Method (FMM) - Greengard & Rokhlin**
- ▶ ...

## Diminuer les interaction lointaines



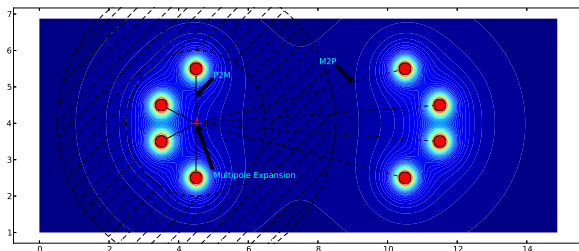
- ▶ Grouper les particules pour séparer les interactions selon leur portée

## Diminuer les interaction lointaines



- ▶ Grouper les particules pour séparer les interactions selon leur portée
- ▶ Remplacer les interactions entre groupes de particules *lointains* par des **approximations** du potentiel

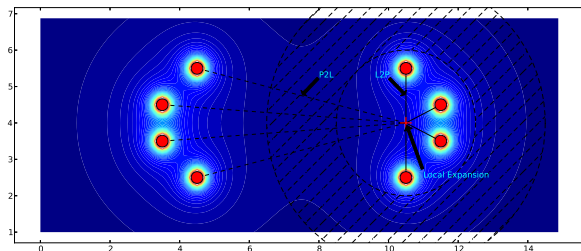
## Expansion Multipôle



- ▶ L'expansion Multipôle exprime le potentiel émis par un groupe de particules *sources* proches d'un point central (opérateur  $P2M$ )
- ▶ On peut exprimer le potentiel d'une expansion multipôle pour des particules *destinations* suffisamment lointaines (opérateur  $M2P$ )

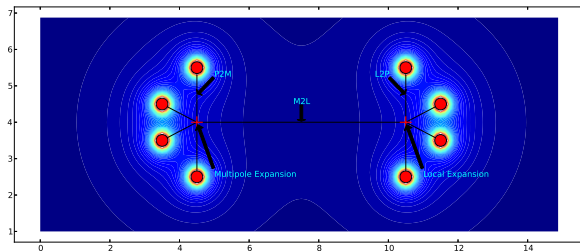


## Expansion Locale



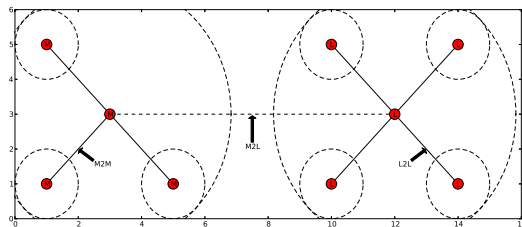
- ▶ On peut aussi définir l'expansion Locale pour des particules *sources* extérieures à une certaine zone (opérateur  $P2L$ )
- ▶ L'expansion Locale est valable pour des particules *destinations* suffisamment proches de son centre (opérateur  $L2P$ )

## Translation Multipôle vers Local ( $M2L$ )



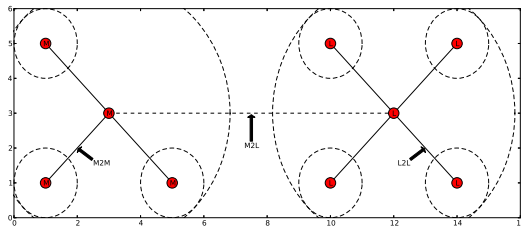
- ▶ On introduit un opérateur  $M2L$  qui permet d'utiliser à la fois les expansions Multipôles et Locales en transférant le potentiel de l'une vers l'autre
- ▶ L'influence de plusieurs expansions Multipôles seront accumulées dans une même expansion Locale

## Niveaux multiples



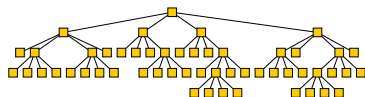
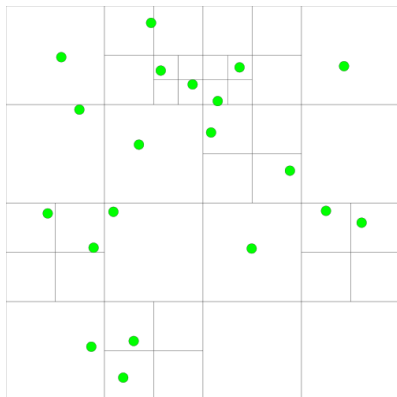
- ▶ Des opérateurs  $M2M$  permettent de regrouper les expansions Multipôles en expansions de plus grande portée
- ▶ Les expansions Locales de plus grande portée correspondants seront reportées dans les expansions Locales plus petites grâce à un opérateur  $L2L$

## Niveaux multiples



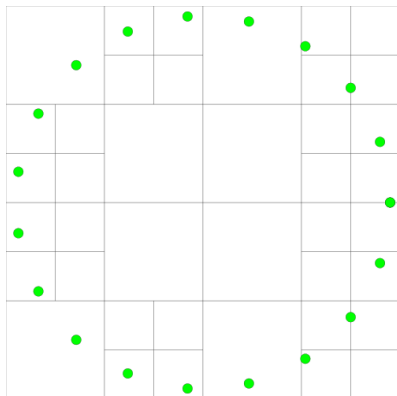
- ▶ Des opérateurs  $M2M$  permettent de regrouper les expansions Multipôles en expansions de plus grande portée
- ▶ Les expansions Locales de plus grande portée correspondants seront reportées dans les expansions Locales plus petites grâce à un opérateur  $L2L$
- ▶ L'empilement des niveaux donne une complexité en  $O(N)$

## QTree / OcTree I



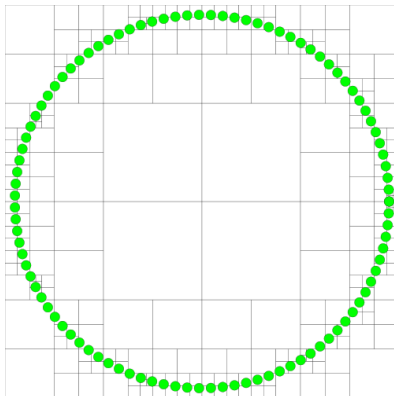
- ▶ Les arbres quaternaires en 2D (resp. octaux en 3D) fournissent la décomposition des particules en groupes
- ▶ On insère les particules dans une boîte jusqu'à ce que le nombre de particules dépasse un paramètre  $s$ , la boîte est divisée en 4 (resp. 8)

# Arbres adaptatifs



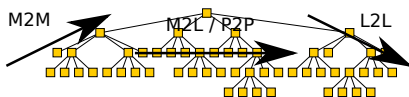
- ▶ La hiérarchie des arbres QTree et OcTree fournit la structure multi-niveaux pour l'algorithme
- ▶ On place un développement Multipôle et un développement Local centre de chaque boîte (feuilles non vides et nœuds internes)

# Arbres adaptatifs



- ▶ La hiérarchie des arbres QTree et OcTree fournit la structure multi-niveaux pour l'algorithme
- ▶ On place un développement Multipôle et un développement Local centre de chaque boîte (feuilles non vides et nœuds internes)
- ▶ Ils s'adaptent à la distribution des particules

## Récapitulatif : étapes de la méthode



- ▶ Passe montante
  - ▶ Accumulation des potentiels des particules dans les expansions Multipôles des feuilles  $\rightarrow P2M$
  - ▶ Remontée des potentiels dans la hiérarchie de l'arbre  $\rightarrow M2M$
- ▶ interactions proches et lointaines
  - ▶ Interactions proches pour boîtes voisines  $\rightarrow P2P$
  - ▶ Interactions lointaines pour boîtes convenablement séparées  $\rightarrow M2L$
- ▶ Passe descendante
  - ▶ Descente des potentiels Locaux dans l'arbre  $\rightarrow L2L$
  - ▶ Expression des potentiels (ou des champs) des expansions Locales des feuilles vers les particules  $\rightarrow L2P$



## Développement en 2D

- ▶ Champ :  $\frac{1}{r}$  (assimilé à  $\frac{1}{z}$ ,  $z = x + iy$ )
- ▶ Séries de Laurent
- ▶ Expansion Multipôle :  $\{a_k\}_{0 \leq k \leq K}$

$$M2P : \phi(z) = \sum_{k=0}^K \frac{a_k}{(z - z_M)^{k+1}}$$

- ▶ Expansion Locale :  $\{b_k\}_{0 \leq k \leq K}$

$$L2P : \phi(z) = \sum_{k=0}^K b_k (z - z_L)^k$$

## Développement en 3D I

- ▶ Champ :  $\frac{1}{r^2}$ , potentiel :  $\frac{1}{r}$
- ▶ Développement en Harmoniques Sphériques

$$Y_l^m(\theta, \phi) = \sqrt{\frac{2l+1}{4\pi} \frac{(l-m)!}{(l+m)!}} P_l^m(\cos\theta) e^{im\phi}$$

- ▶ Expansion Multipôle :  $\{\hat{Y}_l^m(r\vec{M})\}_{0 \leq |m| \leq l \leq K}$

$$M2P : \Phi(\vec{r}) = \sum_{l=0}^K \frac{1}{r^{l+1}} \sum_{m=-l}^l \hat{Y}_l^m(r\vec{M}) Y_l^m(\theta_{\vec{r}-r\vec{M}}, \phi_{\vec{r}-r\vec{M}})$$

## Développement en 3D II

- Expansion Locale :  $\{\check{Y}_l^m(\vec{r}_L)\}_{0 \leq |m| \leq l \leq K}$

$$L2P : \Phi(\vec{r}) = \sum_{l=0}^K r^l \sum_{m=-l}^l \check{Y}_l^m(\vec{r}_L) Y_l^m(\theta_{\vec{r}-\vec{r}_L}, \phi_{\vec{r}-\vec{r}_L})$$

- Exemple d'opérateur de translation : M2L (complexité en  $O(K^4)$ )

$$\check{Y}_l^m(\vec{r}_L) = (-1)^l \sum_{n=0}^K \sum_{o=-n}^n \frac{1}{|\vec{r}_L - \vec{r}_M|^{l+n+1}} \frac{\beta_l \beta_{l+n} \alpha_l^m \alpha_n^o}{\beta_n \alpha_{l+n}^{m+o}} \cdot \frac{Y_{l+n}^{m+o}(\theta_{\vec{r}_L-\vec{r}_M}, \phi_{\vec{r}_L-\vec{r}_M}) \hat{Y}_l^m(\vec{r}_M)}{Y_{l+n}^{m+o}(\theta_{\vec{r}_L-\vec{r}_M}, \phi_{\vec{r}_L-\vec{r}_M}) \hat{Y}_l^m(\vec{r}_M)}$$

# La librairie Aran

- ▶ Une librairie C pour la méthode des Multipôles en 2D et 3D
- ▶ Composé de 2 librairies
  - ▶ LibVsg - primitives géométriques et arbres quaternaires et octaux
  - ▶ LibAran - noyaux numériques (séries de Laurent, harmoniques sphériques, ...)
- ▶ Version parallèle MPI pur

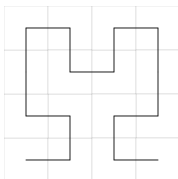
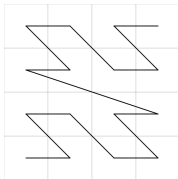
## Quelques détails d'implémentation

- ▶ Insertion, recherche, déplacement et suppression de particules avec adaptation automatique de l'arbre
- ▶ Utilisation de l'optimisation Point&Shoot ou Kylin pour des opérateurs de translation en  $O(K^3)$
- ▶ Opérateurs P2L et M2P pour accélérer les calculs sur des distributions de particules déséquilibrées
- ▶ Pas (encore ?) de conditions limites périodiques
- ▶ Pas (encore ?) d'autre potentiel que Coulomb
- ▶ Pas d'accélération des opérateurs de translation par ondes planes (3D)

## Décomposition parallèle d'un arbre quaternaire/octal

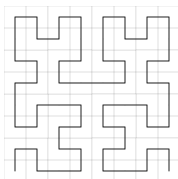
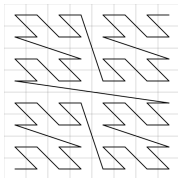
- ▶ Les feuilles de l'arbre sont présentes sur un processeur et un seul. Une feuille est donc *locale* pour un processeur quand elle est stockée par celui-ci ou *distante* sinon
- ▶ Les nœuds internes peuvent être soit :
  - ▶ *locaux* si tous leurs descendants sont locaux
  - ▶ *distants* si tous leurs descendants sont situés sur un même processeur distant
  - ▶ *partagés* sinon (les nœuds partagés sont connus de tous les processeurs)
- ▶ Effectuer la décomposition de domaine d'un arbre consiste à déterminer quelles feuilles doivent être stockées sur quel processeur

## Répartition des feuilles



- ▶ La méthode la plus simple : trouver une numérotation des feuilles et distribuer aux processeurs des segments de numéros contigus.
  - ▶ Ordre de Morton (Z-order)
  - ▶ Ordre de Peano-Hilbert
- ▶ L'ajout, la suppression ou le déplacement de particules peut amener à un déséquilibre de la décomposition initiale.
- ▶ Il suffit de ré-appliquer la décomposition pour revenir à une répartition idéale

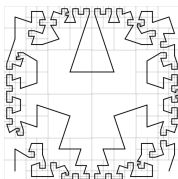
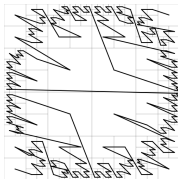
## Répartition des feuilles



- ▶ La méthode la plus simple : trouver une numérotation des feuilles et distribuer aux processeurs des segments de numéros contigus.
  - ▶ Ordre de Morton (Z-order)
  - ▶ Ordre de Peano-Hilbert
- ▶ L'ajout, la suppression ou le déplacement de particules peut amener à un déséquilibre de la décomposition initiale.
- ▶ Il suffit de ré-appliquer la décomposition pour revenir à une répartition idéale



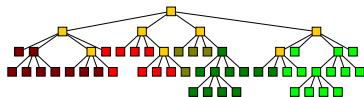
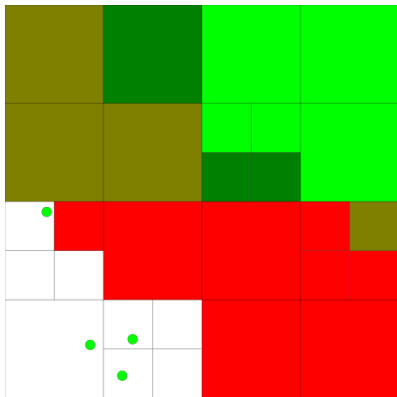
## Répartition des feuilles



- ▶ La méthode la plus simple : trouver une numérotation des feuilles et distribuer aux processeurs des segments de numéros contigus.
  - ▶ Ordre de Morton (Z-order)
  - ▶ Ordre de Peano-Hilbert
- ▶ L'ajout, la suppression ou le déplacement de particules peut amener à un déséquilibre de la décomposition initiale.
- ▶ Il suffit de ré-appliquer la décomposition pour revenir à une répartition idéale

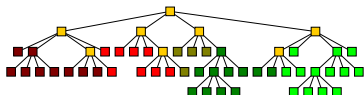
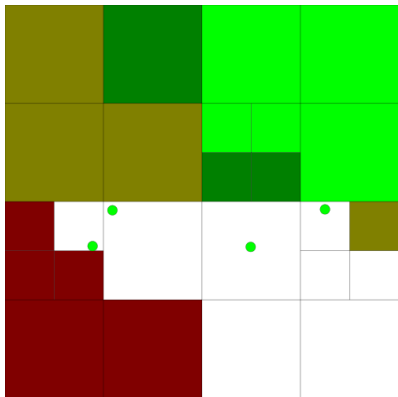
# La décomposition parallèle en images

Vue du processeur 0



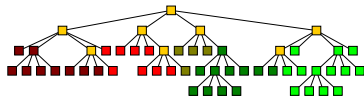
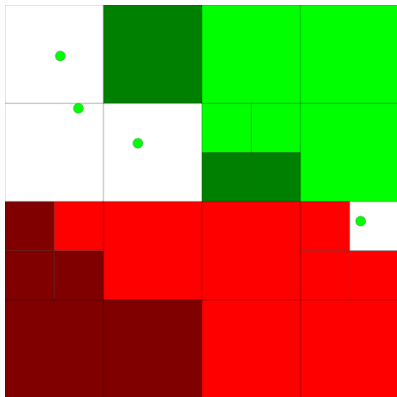
# La décomposition parallèle en images

Vue du processeur 1



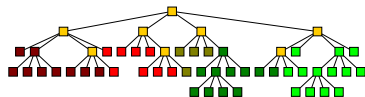
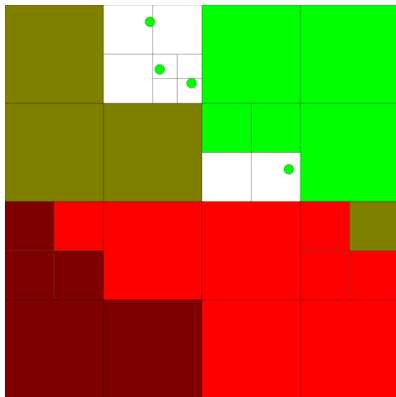
# La décomposition parallèle en images

Vue du processeur 2



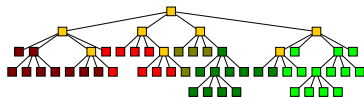
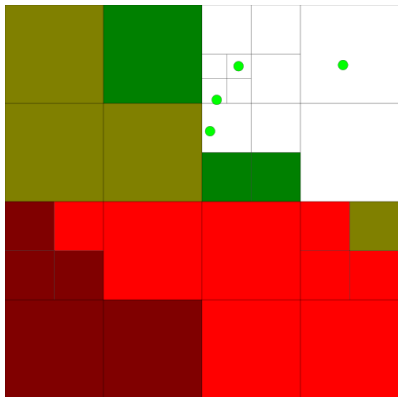
# La décomposition parallèle en images

Vue du processeur 3



# La décomposition parallèle en images

Vue du processeur 4



## L'algorithme FMM distribué I

- ▶ Calcul des interactions entre objets situés sur des processeurs différents (P2P / M2L)
  - ▶ Lors du parcours de la partie locale de l'arbre, l'un des processeurs trouve un objet qui nécessite des interactions avec des objets *distants*; il envoie l'objet vers l'autre processeur (qui vérifie régulièrement l'arrivée de ce genre de messages)
  - ▶ Le calcul a lieu sur le deuxième processeur qui renvoie le résultat vers le premier
- ▶ Les envois de messages sont non-bloquants. Les processeurs ne savent pas a priori combien d'objets il doit recevoir en tout
- ▶ Le parcours M2L + P2P se termine donc par une étape de synchronisation qui peut être pénalisante si les charges des processeurs sont déséquilibrées

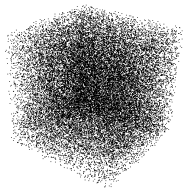
## L'algorithme FMM distribué II

- ▶ Opérations de type Allreduce pour la partie *partagée* de l'arbre
  - ▶ Passe montante : il faut transmettre le résultat des opérations M2M d'un nœud *local* vers un parent *partagé*
  - ▶ A la fin du parcours M2L + P2P, certaines interactions M2L ont pu intervenir avec des nœuds *partagés*; il faut transmettre ces résultats sur tous les processeurs
- ▶ La nature dissimulée des nœuds *distants* engendre de grandes quantités de messages inutiles; il faut introduire l'information sur la profondeur minimum du sous-arbre contenu par un nœud *distant*
- ▶ Diverses optimisations permettent de réduire les pertes de performance (concaténation de messages en attente, omission des retours de résultats lors d'envois inutiles, ...)

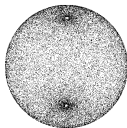


## Cas tests

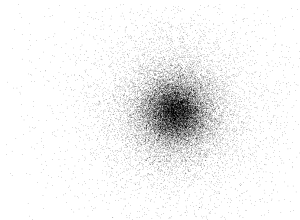
On présente une étude de performances sur trois distributions de particules différentes :



Cube (*random*)



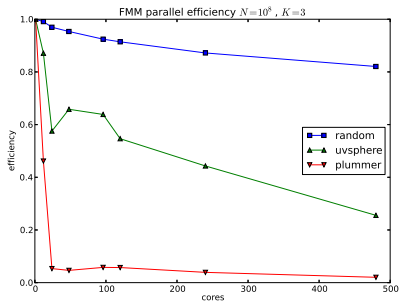
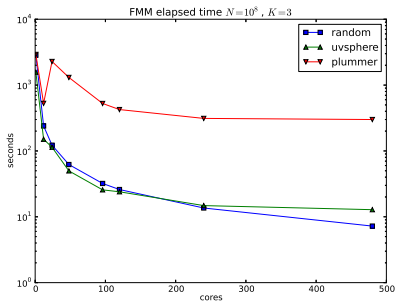
Surface d'une sphère  
(*uvsphere*)



Plummer (*plummer*)  
$$\rho(r) = \frac{1}{(1 + \frac{r^2}{a^2})^{\frac{5}{2}}}$$

Les calculs sont effectués pour deux valeurs différentes du paramètre  $K$  :  $K = 3$  et  $K = 8$

# Scalabilité forte

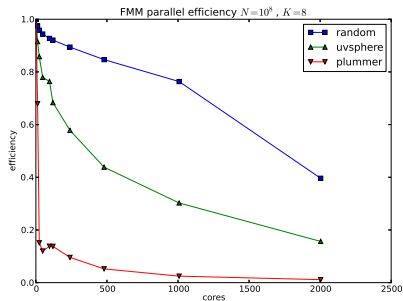
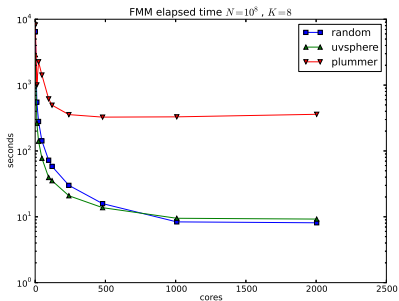


100 millions de particules

Paramètre de troncature :  $K = 3$

Les distributions in-homogènes sont problématiques (surtout *plummer*) : déséquilibre des charges

# Scalabilité forte

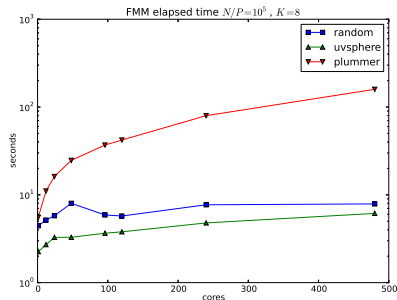
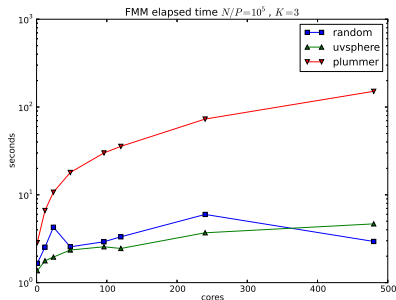


100 millions de particules

Paramètre de troncature :  $K = 8$

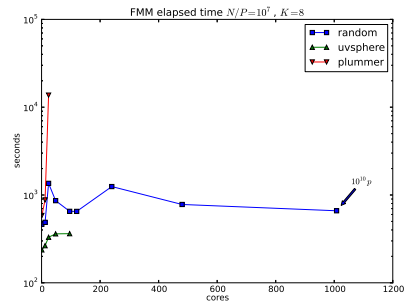
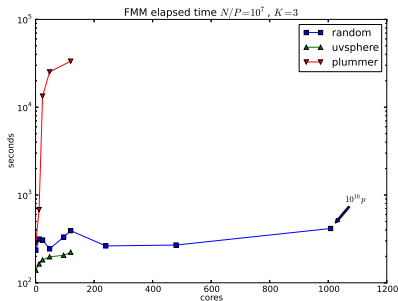
Les distributions in-homogènes sont problématiques (surtout *plummer*) : déséquilibre des charges

# Scalabilité faible I



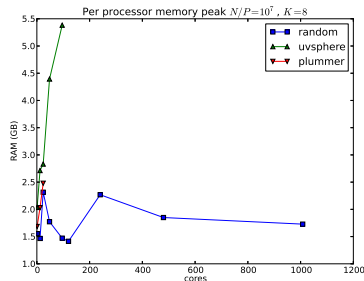
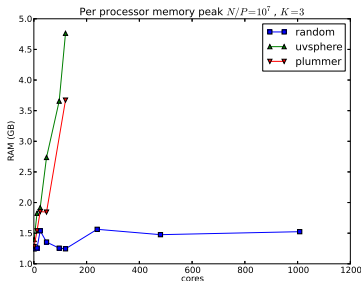
Nombre de particules par processeur :  $N/P = 10^5$

# Scalabilité faible II



Nombre de particules par processeur :  $N/P = 10^7$

## Consommation de mémoire



- ▶ La consommation pour la distribution *random* est raisonnable
- ▶ Les distributions *uvsphere* et *plummer* posent problème à cause de la part importante de la partie *partagée* de l'arbre qui est dupliquée sur chaque processeur

## Travaux futurs

- ▶ Amélioration de l’empreinte mémoire
- ▶ Amélioration de la performance parallèle par équilibrage de charge
  - ▶ Migration dynamique des calculs
  - ▶ Pondération de la répartition parallèle après une première résolution
- ▶ Version *multithread* → parallélisme hybride
- ▶ Implémentation d’autres potentiels
- ▶ ...

## Conclusion

- ▶ On a présenté la librairie Aran pour la résolution parallèle du problème des N-corps par la Méthode des Multipôles Rapides
- ▶ On a montré les performances parallèles :
  - ▶ Efficacité parallèle importante (0.75) pour des problèmes de l'ordre de 100 millions de particules sur 1000 tâches MPI
  - ▶ Capacité de résolution de problèmes jusqu'à 10 milliards de particules
  - ▶ Mais les distributions in-homogènes sont encore problématiques
- ▶ Le code d'Aran est libre et disponible sous LGPL
- ▶ → Fork me on GitHub : <https://github.com/pigay/aran>
- ▶ *Computer time for this study was provided by the computing facilities MCIA (Mésocentre de Calcul Intensif Aquitain) of the Université de Bordeaux and of the Université de Pau et des Pays de l'Adour*



# Questions

- ▶ Des questions ?
- ▶ Merci de votre attention