
Algorithmes parallèles de simulation physique pour la synthèse d'images : application à l'animation de textiles

Florence Zara

Directeur de thèse : Brigitte Plateau (ID-IMAG)

Co-encadrants : Jean-Marc Vincent (ID-IMAG)

François Faure (GRAVIR-IMAG)

Laboratoire ID-IMAG, projet INRIA-APACHE
en collaboration avec le laboratoire GRAVIR-IMAG, projet INRIA-EVASION
et partiellement financé par la région Rhône-Alpes, projet SAPPE



Introduction (1)

Animation interactive de textiles en temps réel



- A partir d'un vêtement donné, comment en faire une animation ?
 - Comment modéliser cette robe ?
 - Comment la faire bouger ?
 - Comment interagir avec elle ?
- Objectif
 - Animation réaliste en temps réel
 - Interaction avec le vêtement

Introduction (2)

Intérêt de l'animation de textiles en temps réel

- Industries de la confection et du tissage
 - Croquis + patron + vêtement + essayage
 - Prototypage virtuel
 - Modélisation 3D et essayage virtuel
 - Diminution des coûts de mise au point

- Applications dans le monde du loisir
 - Films d'animation, jeux en réseau, RV
 - Habillage de personnages animés



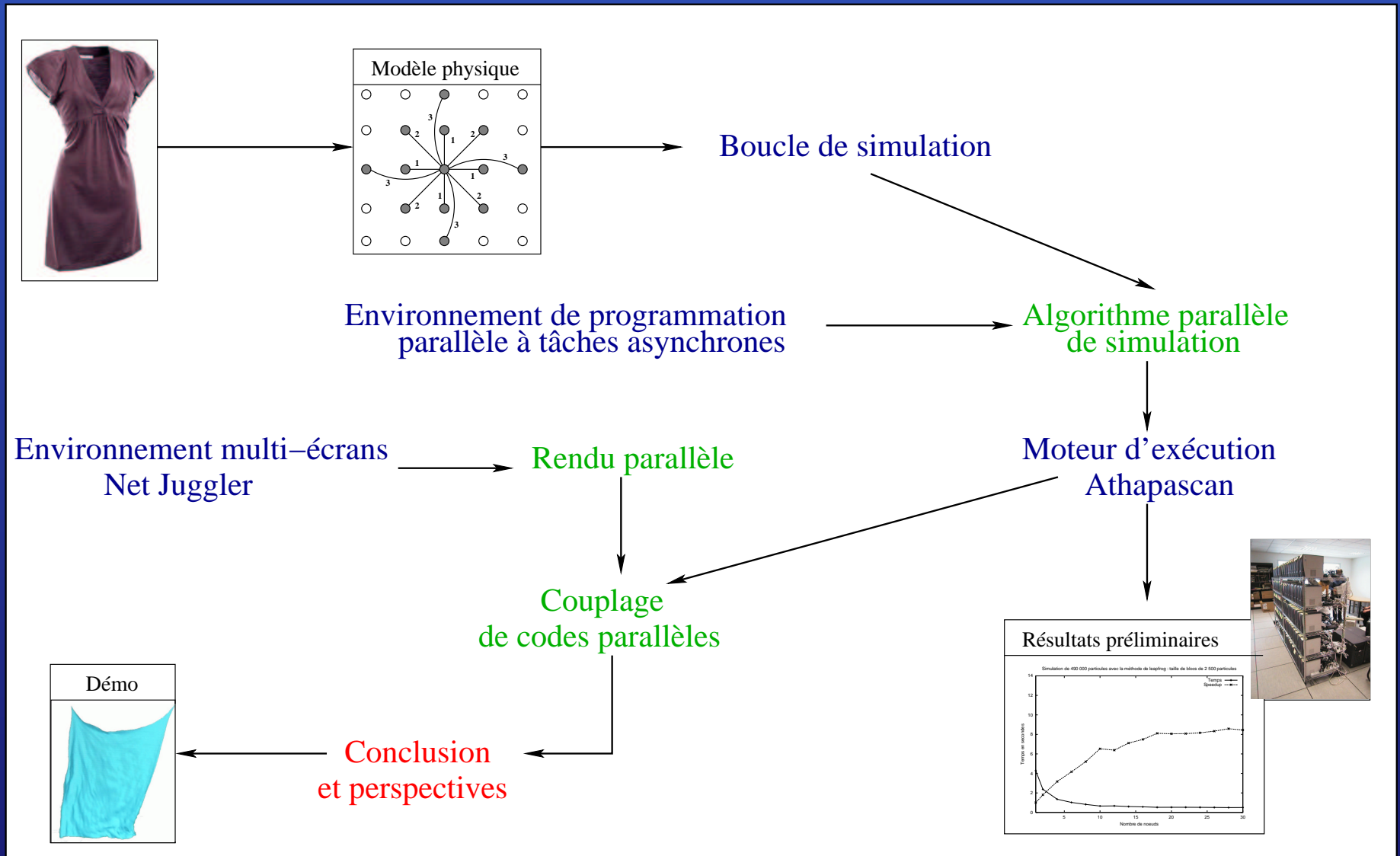
Problématique (1)

- Temps d'exécution et taille du problème trop importants pour du temps réel
- Approche du monde de l'informatique graphique
 - Simplification des algorithmes
→ Animations peu réalistes
- Approche du monde du parallélisme
 - Utilisation de plusieurs machines
 - Environnements de programmation parallèle
 - Parallélisme exprimé à base de tâches de calculs
 - ▲ Problème décomposé en sous-problèmes
 - ▲ Sous-problèmes traités en parallèle
 - Résultats récupérés en fin d'exécution

Problématique (2)

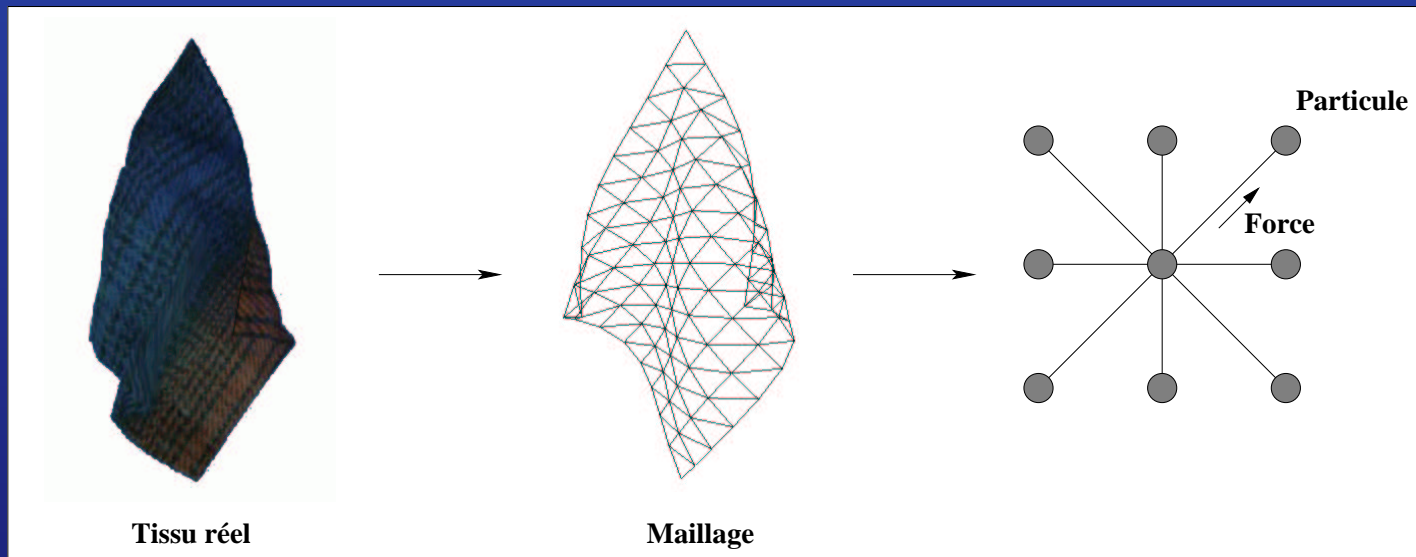
- Solution proposée
 - Parallélisation des algorithmes de synthèse d'animation pour obtenir des animations en temps réel
 - Utilisation de l'environnement ATHAPASCAN
- Difficultés
 - **Intrinsèques à la parallélisation**
 - ▲ Décomposition de l'animation de textiles en tâches de calcul
 - ▲ Ordonnancement de ces tâches sur les processeurs
 - **Liées à l'animation de textiles**
 - ▲ Visualiser les résultats au cours de la simulation
 - ▲ Interagir avec la simulation parallèle

Plan de la présentation



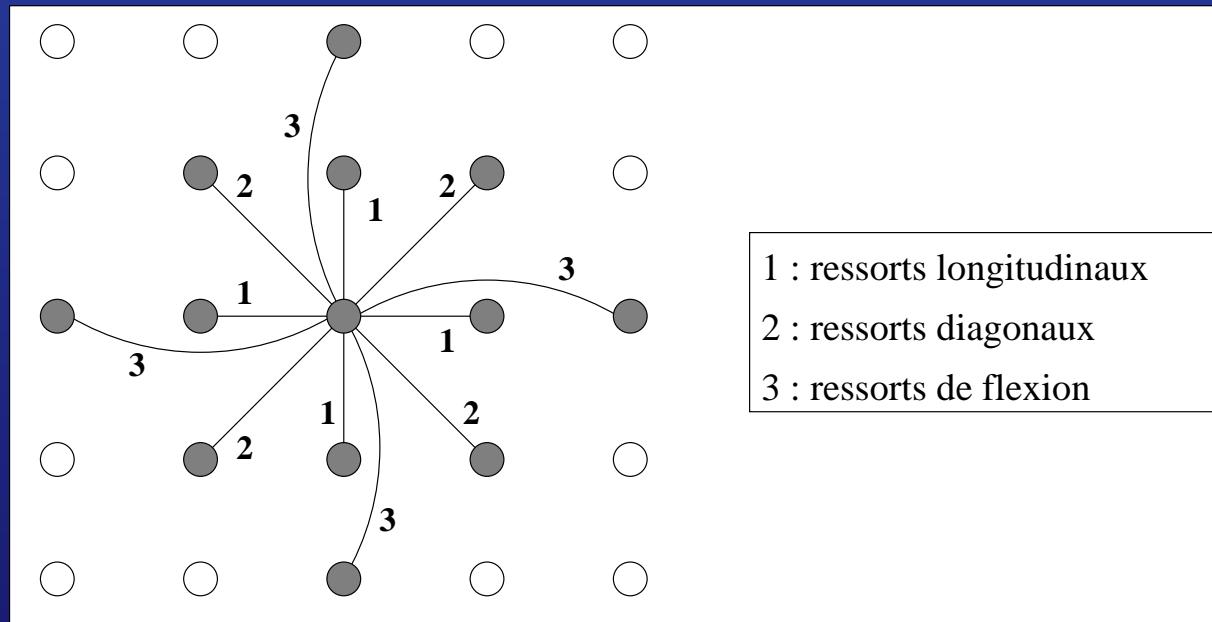
Modèle physique

- Modèle physique basé sur un **systeme de particules**
 - Discrétisation du textile en un maillage polygonal
 - Sommets appelés particules
 - Topologie définit les interactions



Modèle des interactions

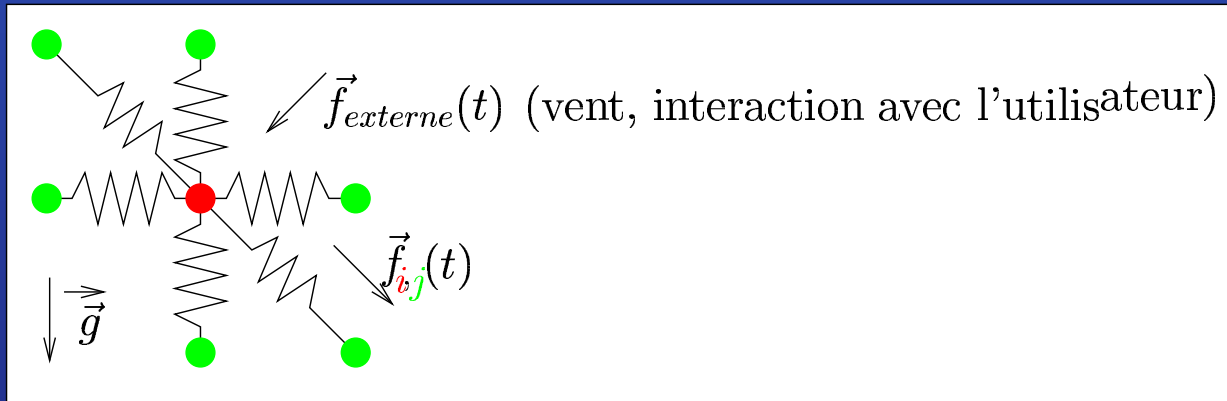
- Définition des interactions entre particules
 - Système **masses-ressorts**
 - ▲ Particules connectées par des ressorts
 - ▲ Paramètres physiques des ressorts : k , l_0 et ν
 - Paramètres de la simulation
 - ▲ Maillage + caractéristiques des ressorts



Champs de forces

- Calcul des forces exercées sur chaque particule

- $\vec{f}_i(t) = \sum_{j|j \text{ voisin de } i} \vec{f}_{i,j}(t) + \text{gravité} + \text{interactions}$



- Seulement ressorts diagonaux et longitudinaux
→ Ressorts de flexion rajoutés par la suite
- Calcul des accélérations
 - Équation du mouvement de Newton

Équation du mouvement

- Intégration pour obtenir les vitesses et les positions
 - Système différentiel associé aux N particules

$$\begin{cases} x''(t) &= M^{-1} f(t, x(t), \dot{x}(t)) \\ x'(t_0) &= v_0 \\ x(t_0) &= x_0 \end{cases}$$

→ f dépend de la définition des interactions

- Multitude de méthodes d'intégration existantes
 - Privilégie la **stabilité numérique** à la précision
 - **Complexité du calcul** séquentielle et parallèle

Méthodes d'intégration numérique

- Trois méthodes d'intégration implantées
 - Méthodes explicites
 - ▲ Euler explicite
 - ▲ Störmer-Verlet/leapfrog (symplectique)
 - h_{expl} contrôlé en taille pour garantir la stabilité
 - h_{expl} expérimentalement petit
 - Facile à découper en tâches de calculs
 - Méthode implicite
 - ▲ Euler implicite
 - Inconditionnellement stable : h_{impl} grand
 - Résolution d'un système linéaire creux
 - Plus difficile à paralléliser

→ Choix entre la complexité de parallélisation / stabilité

Boucle de la simulation

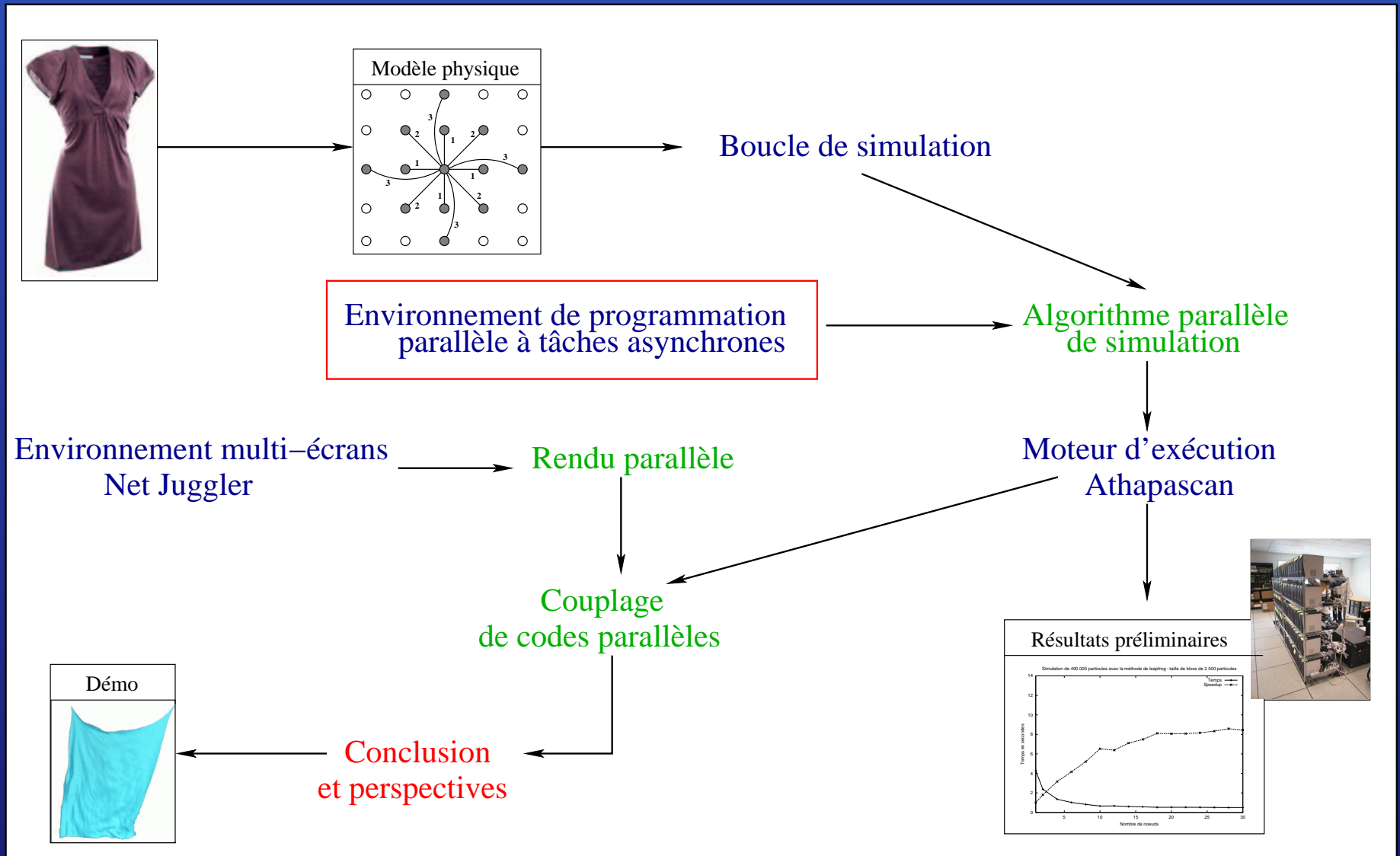
■ Repeter

- Forces exercees par les ressorts
- Forces externes (gravite, vent)
- Calculer les positions et vitesses par integration des accelerations
- Afficher

■ Fin

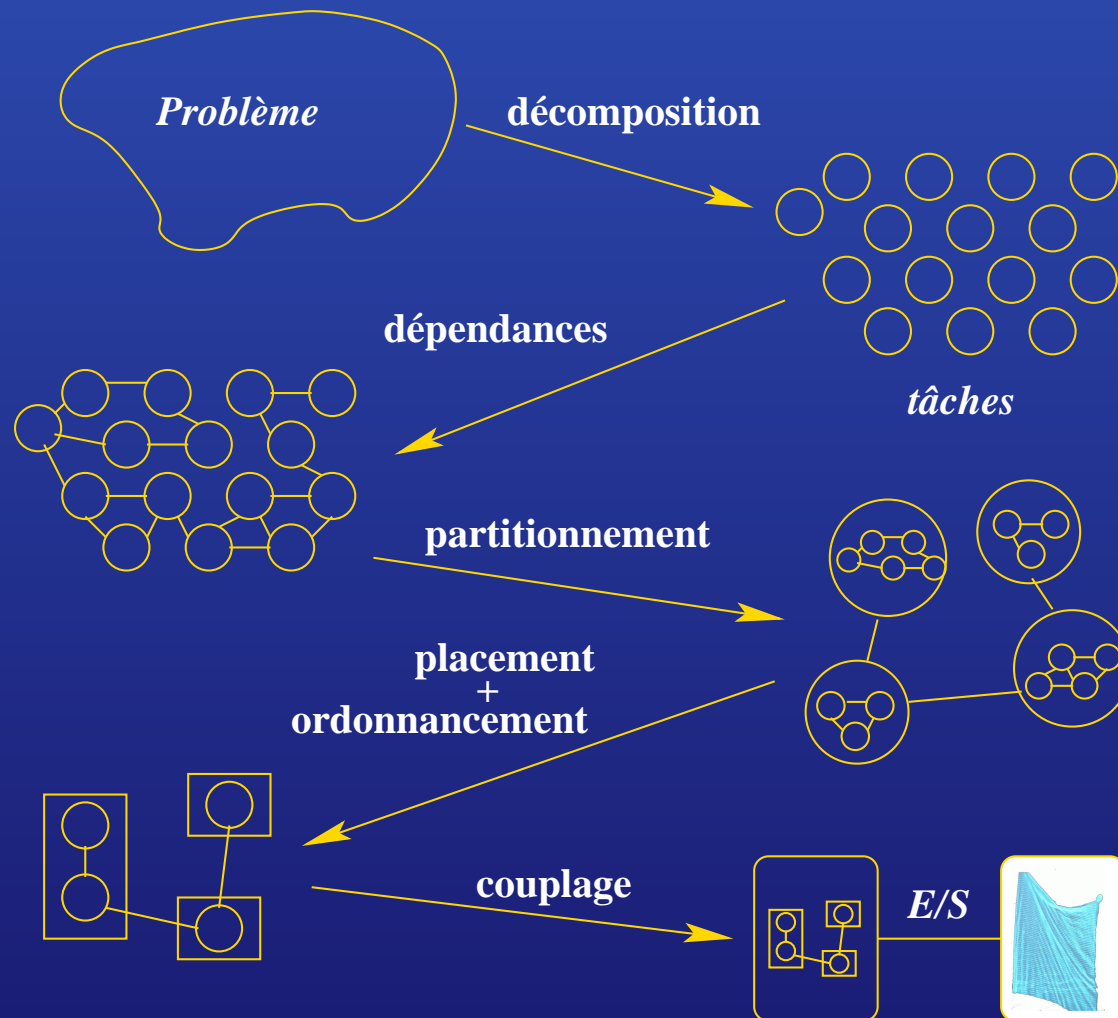
→ Objectif : parallelisation de la boucle de simulation

Plan de la présentation



Conception d'algorithmes parallèles

- Méthodologie pour paralléliser la simulation



ATHAPASCAN

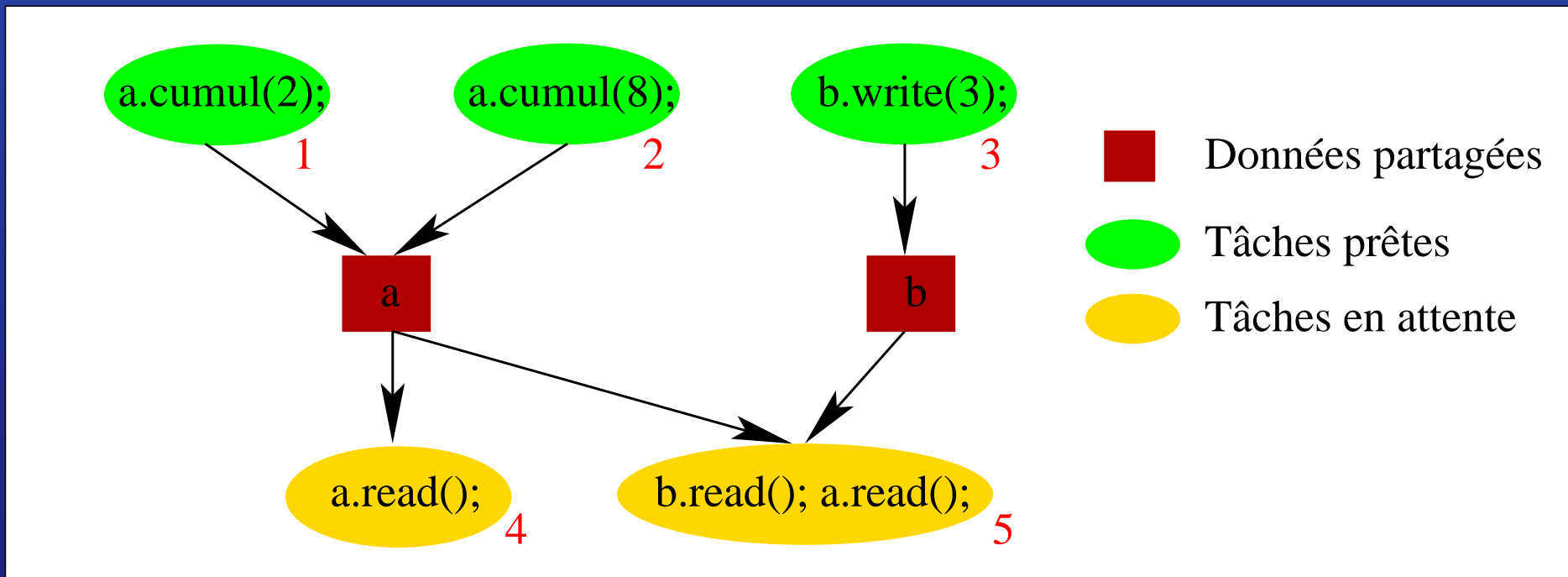
- Environnement de programmation parallèle
 - Facile à utiliser avec 2 mots-clés (`a1::Fork` et `a1::Shared`)
 - Assure la **portabilité** des programmes
 - ▲ Exécutable en séquentiel, en SMP et en distribué
 - ▲ Nombre de noeuds variable
- **Parallélisme explicite**
 - **Granularité explicite au niveau des calculs**
 - ▲ Utilisateur décompose l'application en **tâches**
→ Tâches exécutées en parallèle par ATHAPASCAN
 - **Granularité explicite au niveau des données**
 - ▲ Tâches prennent en paramètres des **données partagées**
 - ▲ Données partagées spécifiées par l'utilisateur
→ Définissent la granularité de l'algorithme parallèle

Interface de programmation de haut niveau

- Typage explicite des accès aux objets partagés effectués par les tâches
 - Concurrents : lecture (R), écriture cumulée (CW)
 - Exclusifs : écriture (W), lecture/écriture (RW)
 - Synchronisation implicite entre les tâches
 - Ordre d'exécution des tâches à respecter
 - ▲ Ordre définit par les dépendances entre les tâches
 - ▲ Issues des accès effectués aux données partagées
- ATHAPASCAN détermine l'ordre d'exécution des tâches et le site de placement

Graphe de flots de données

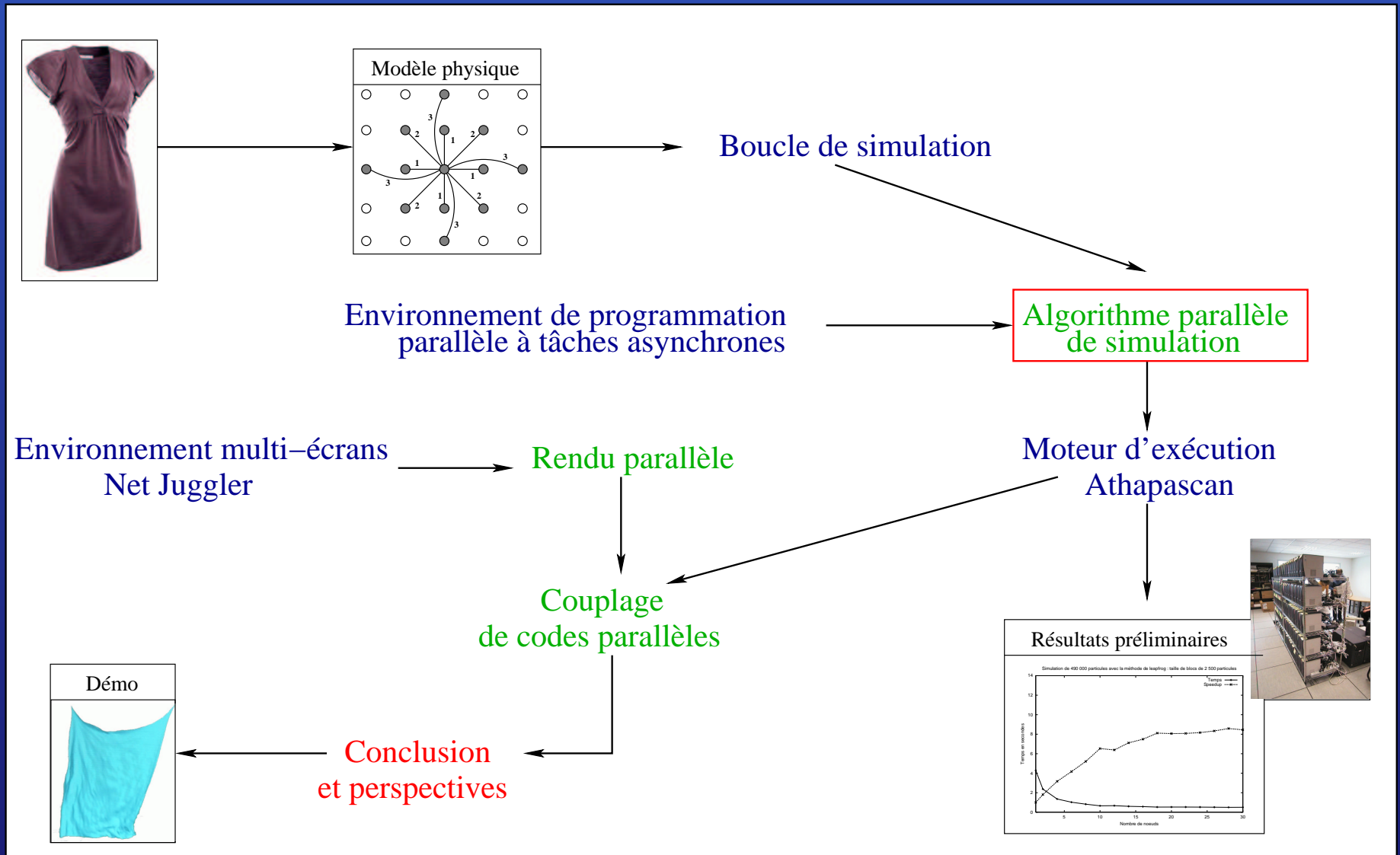
- Ordre basé sur le **graphe de flots de données**
 - Construit dynamiquement durant l'exécution
 - Décrit les dépendances entre les tâches et les données



Ordonnancement

- Ordonnancement semi-statique du GFD à effectuer
 - Placement des données sur les processeurs
 - Ordonnancement des tâches : date + n° processeur
→ Effectués dans un souci de performance
 - Ordonnancement locale des tâches
→ Respecte les dépendances
- Heuristiques d'ordonnancement fournies par ATHAPASCAN
 - Cyclic
 - LPTF (coûts associés aux tâches)
 - ORB (coûts + localisation spatiale des données)
 - Basées sur un partitionnement du graphe (Scotch)
 - Possibilité de spécialiser l'ordonnancement

Plan de la présentation

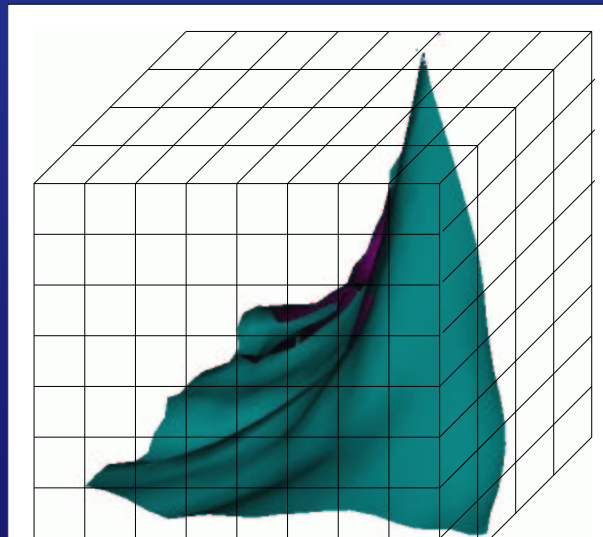


Décomposition

- Décomposition du problème en sous-problèmes
 - Basée sur une **décomposition des données**
 - Engendre une **décomposition en tâches**
 - Relations de précédences entre ces tâches
- Objectif de la décomposition
 - Tâches homogènes en terme de charge de travail
 - Facilite l'équilibrage de la charge sur les processeurs
- **Deux types de décomposition**
 - Décomposition de l'**espace de simulation**
 - Décomposition de l'**espace géométrique de l'objet**

Décomposition de l'espace de simulation

- Espace de simulation décomposé en boîtes
- Chaque tâche traite une partie de l'espace
 - Adapté aux interactions issues du voisinage 3D
 - Ne tient pas compte des interactions au sein de l'objet
- Grande variabilité du nombre de particules par boîte
 - Déséquilibre de la charge de travail par tâche
→ Difficile à ordonnancer efficacement



Décomposition de l'objet

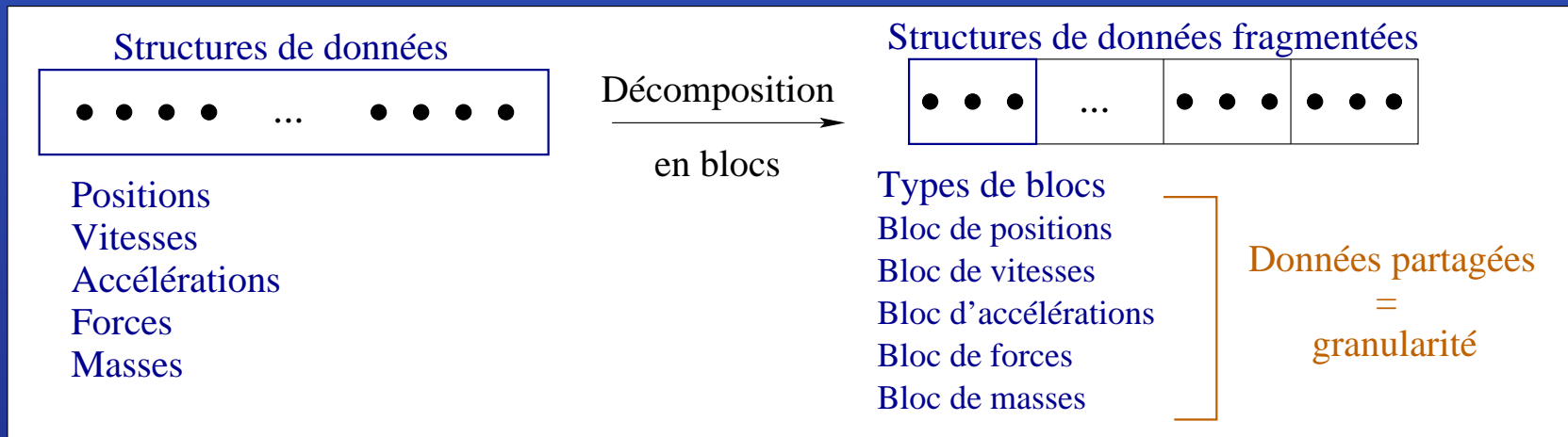
- Textile décomposé en sous-ensembles de taille équivalente



- Chaque tâche traite un sous-ensemble du textile
 - Répartition uniforme de la charge de calcul par tâche
- Perte de l'espace de simulation
 - Ne tient pas compte du voisinage 3D

Calcul sur les blocs

■ Structures de données décomposées en blocs



■ Tâche de calcul traite les données d'un bloc

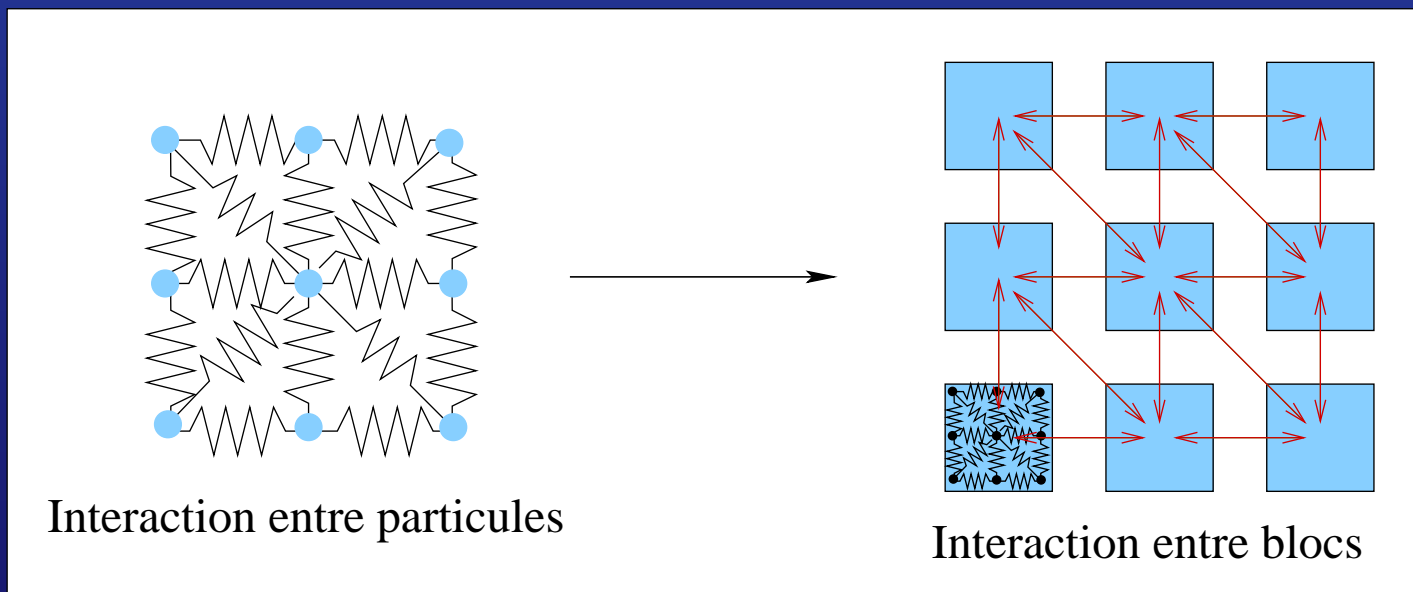
→ Tâches effectuant le même type de calcul sont exécutées en concurrence pour l'ensemble des blocs

■ Différents types de tâches de calculs

- Calcul des forces, des accélérations et des intégrations
→ Relation de précédence entre ces types de tâches

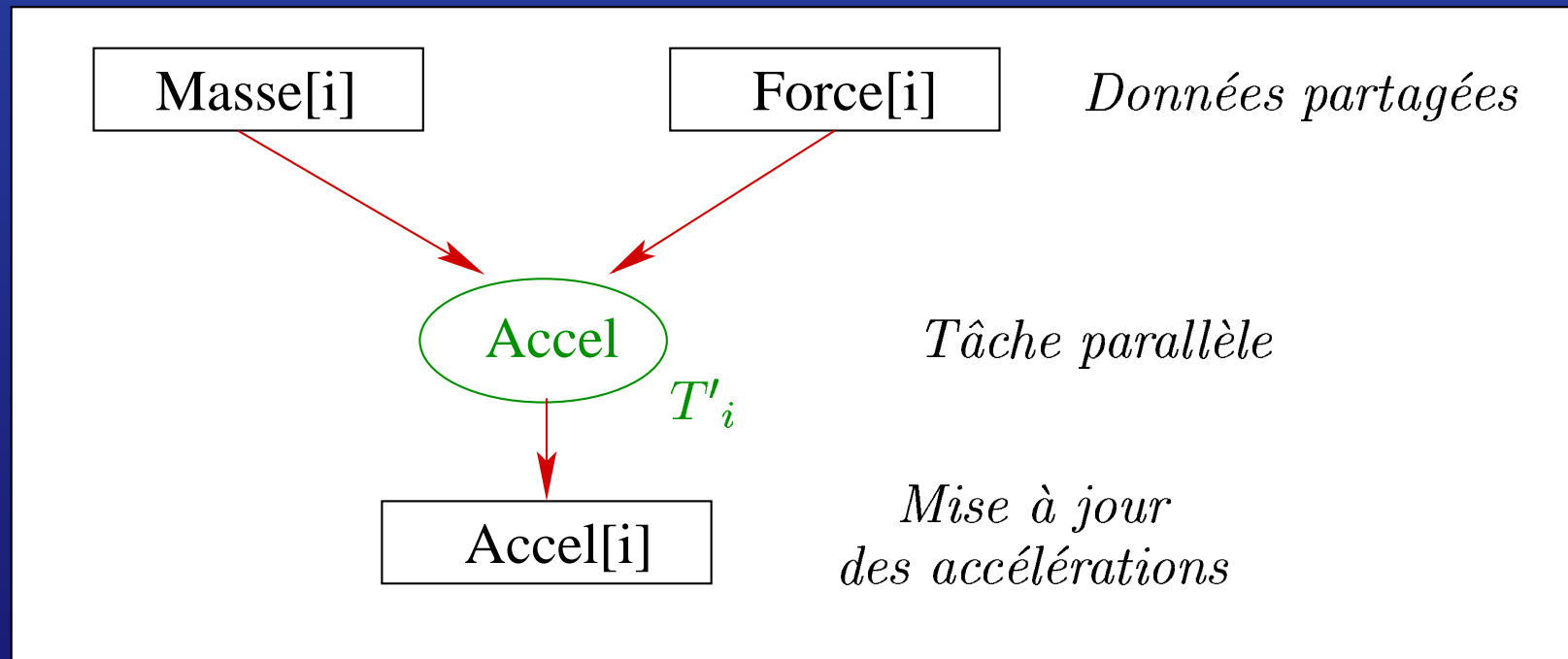
Graphe de tâches

- Tâches dépendent des données d'un ou de deux blocs
 - Tâches indépendantes des autres blocs
 - ▲ Accélération, intégrations explicites
 - Tâches localement dépendantes
 - ▲ Forces entre deux blocs, intégration implicite
 - ▲ Voisinage des particules → voisinage entre blocs
 - ▲ Nécessite connaissance sur les blocs adjacents



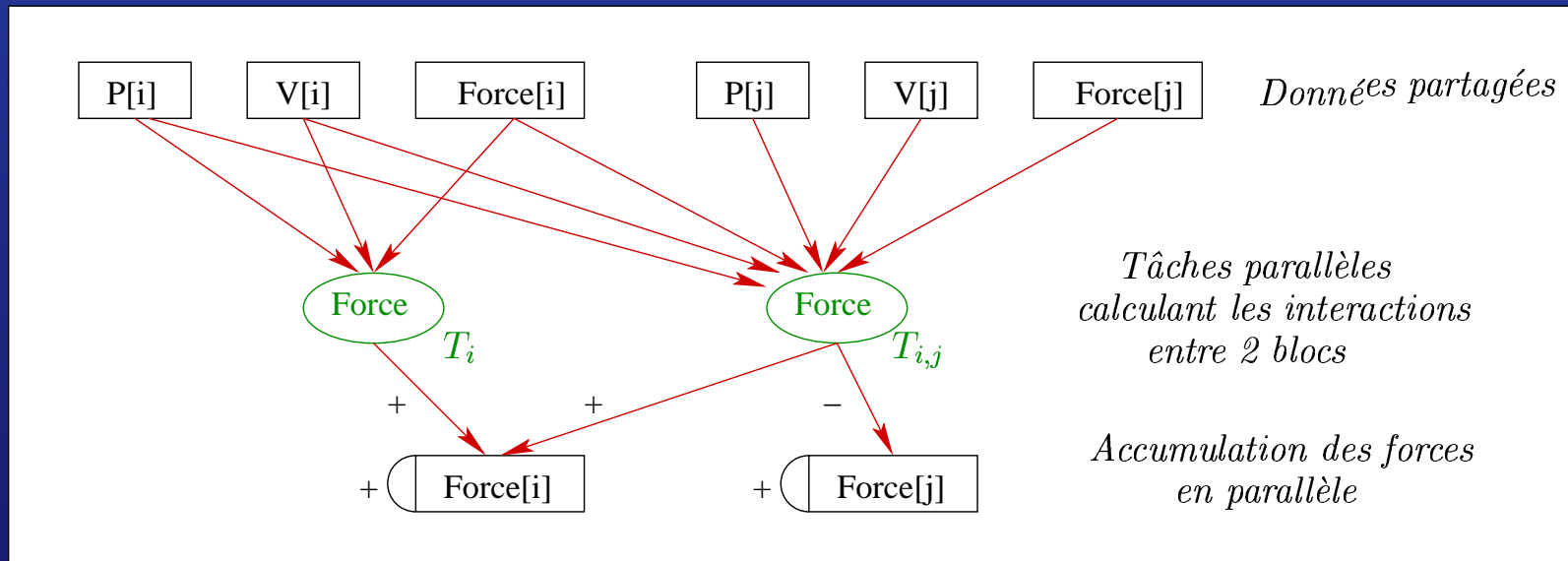
Tâche localement indépendante

- Calcul des **accélérations** du bloc i
 - Masses, forces et accélérations du bloc i accessibles
 - Calcul du bloc i indépendant des autres blocs
 - ▲ Création de la tâche T'_i
 - ▲ Mise à jour de la donnée partagée du bloc i



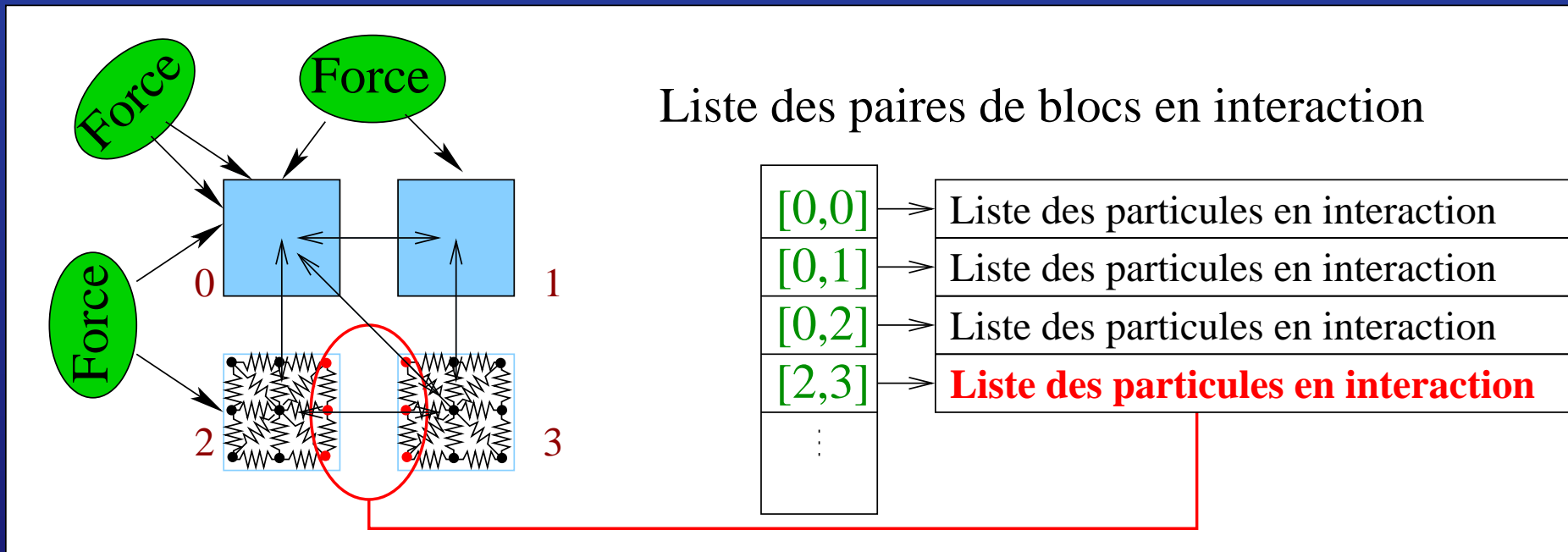
Tâche localement dépendante

- Calcul des **forces d'interaction** du bloc i
 - Interaction du bloc i avec blocs i et j
 - Tâches créées pour chaque paire de blocs
 - ▲ T_i et $T_{i,j}$ exécutées en parallèle
 - Prise en compte de la symétrie des calculs : $f_{i,j} = -f_{j,i}$
 - Accumulation des résultats dans les 2 données partagées

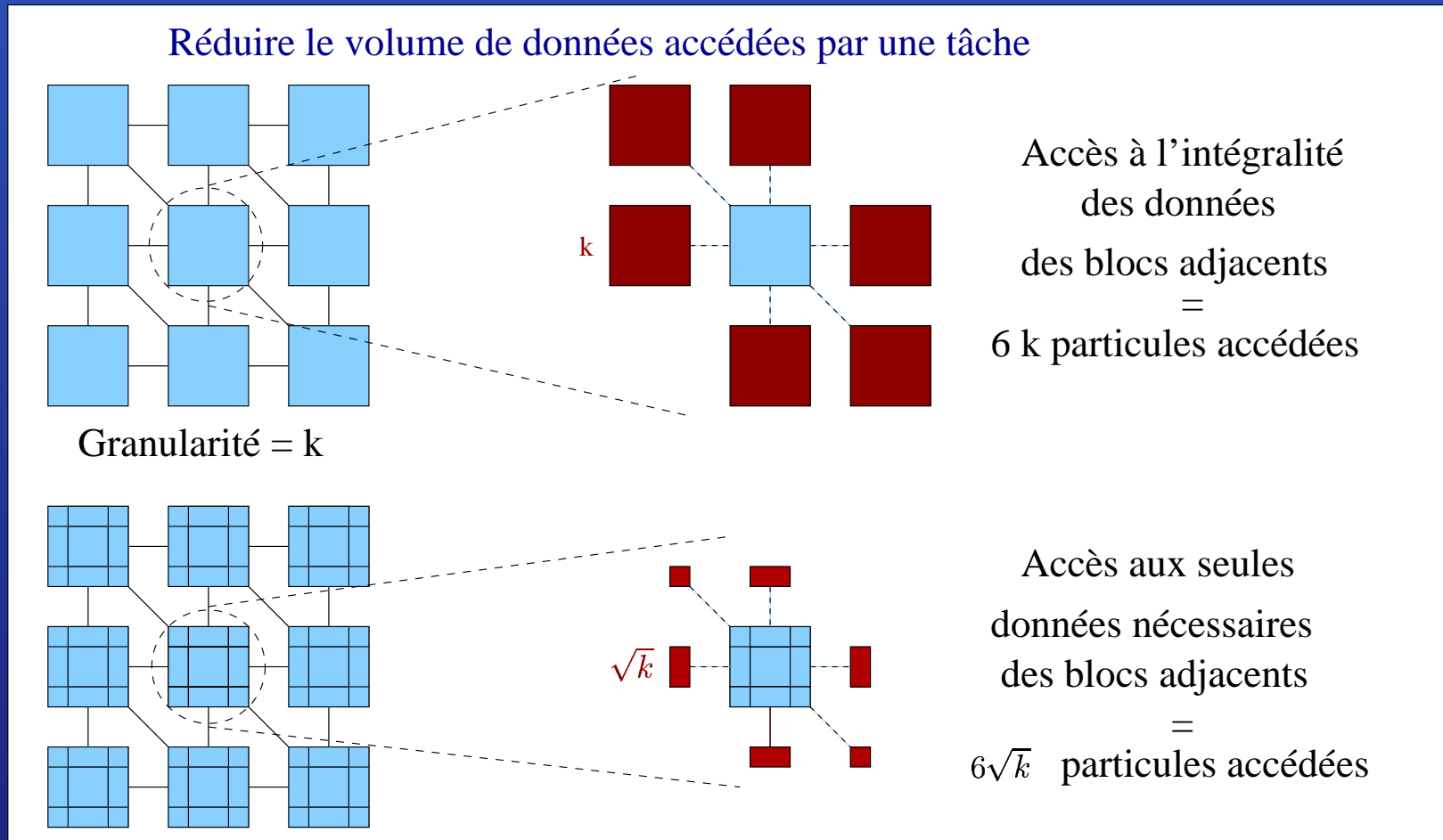


Implantation

- Nécessite la **connaissance des blocs adjacents**
 - A l'initialisation élaboration de
 - ▲ la liste des paires de blocs ayant des interactions
 - ▲ et pour une paire de blocs donnée
→ la liste des particules en interaction dans ces blocs

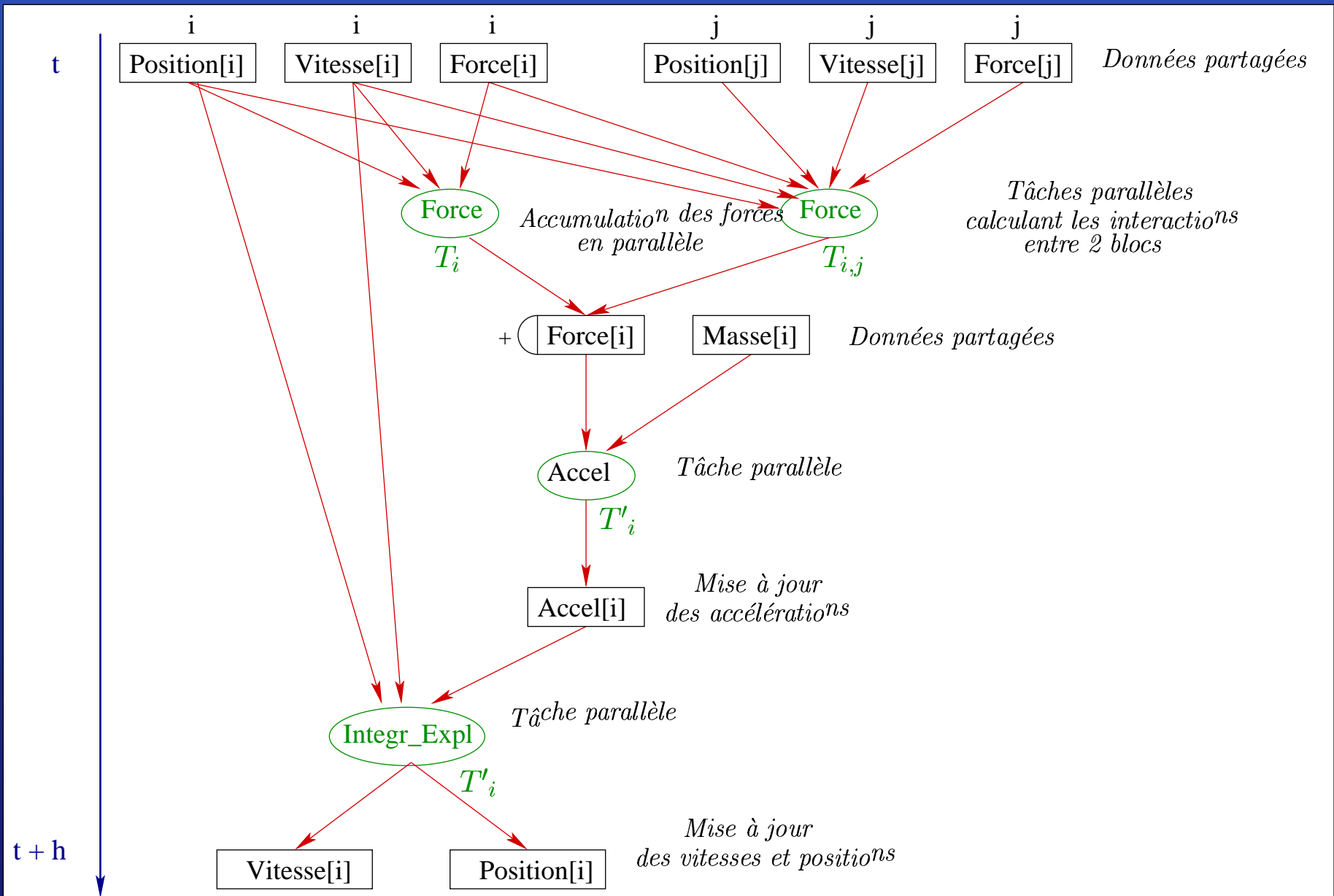


Réduction des coûts des accès

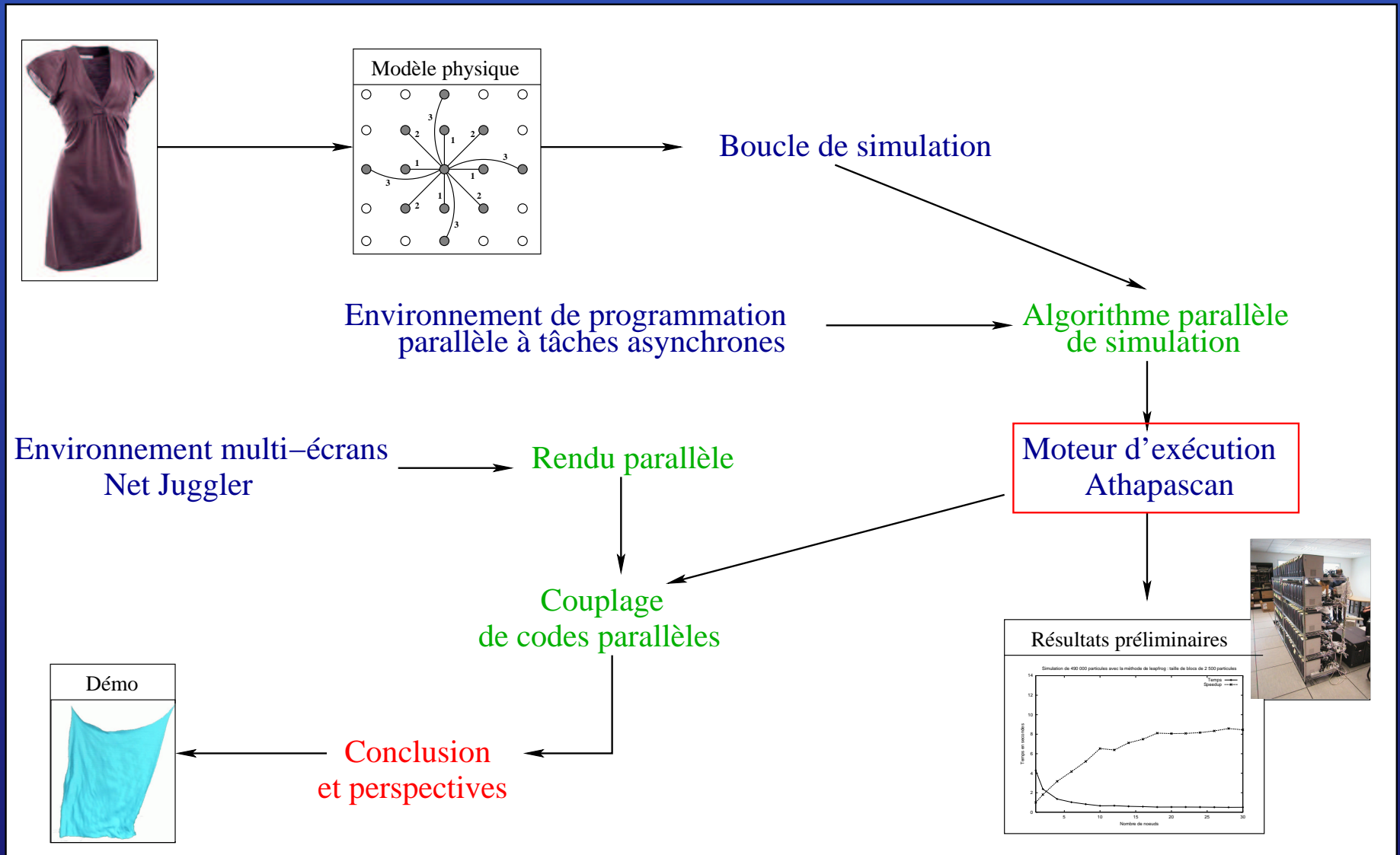


→ Si la granularité augmente d'un facteur k
→ les accès augmentent seulement d'un facteur \sqrt{k}

GFD pour 1 bloc sur une itération

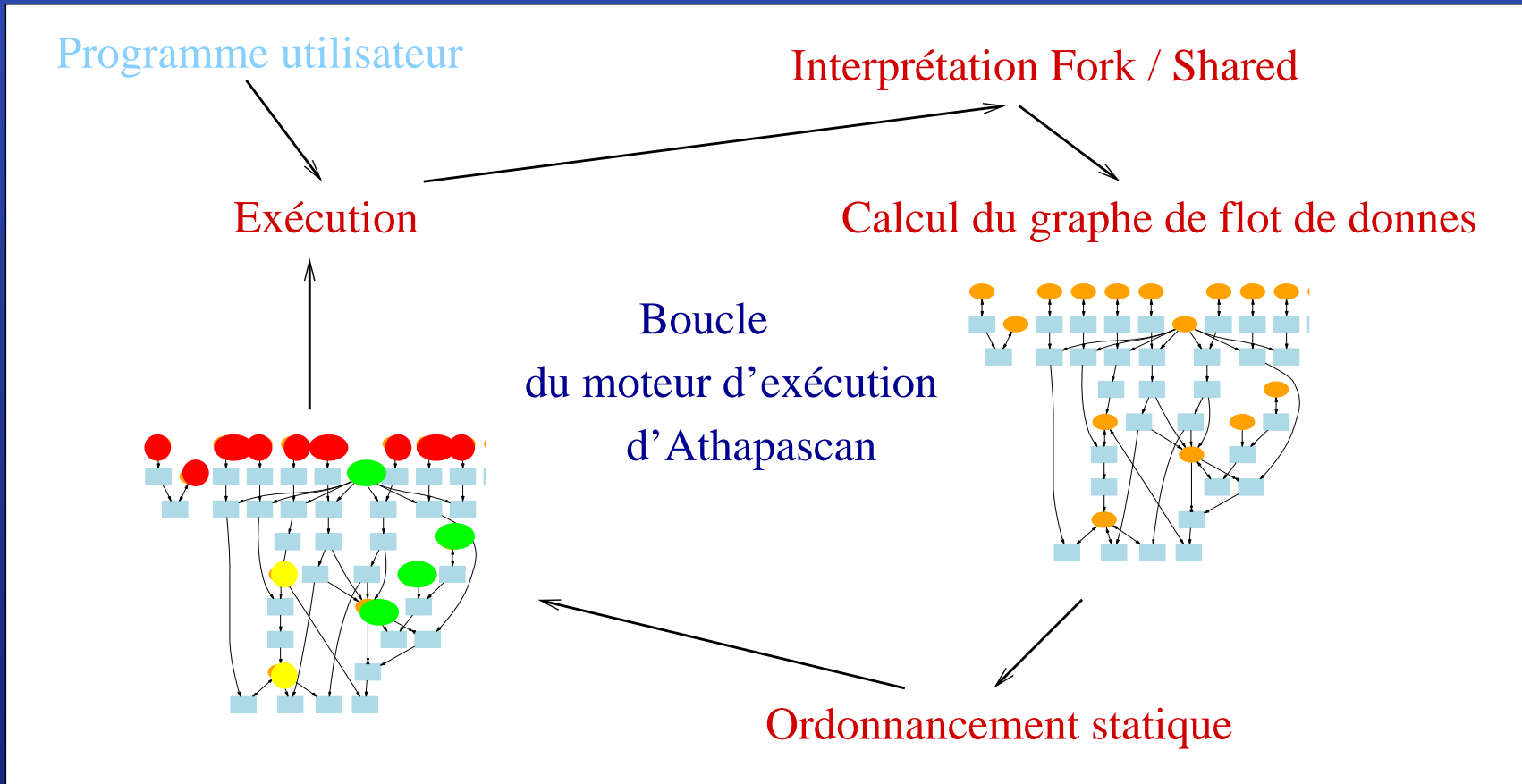


Plan de la présentation



Moteur de l'exécution

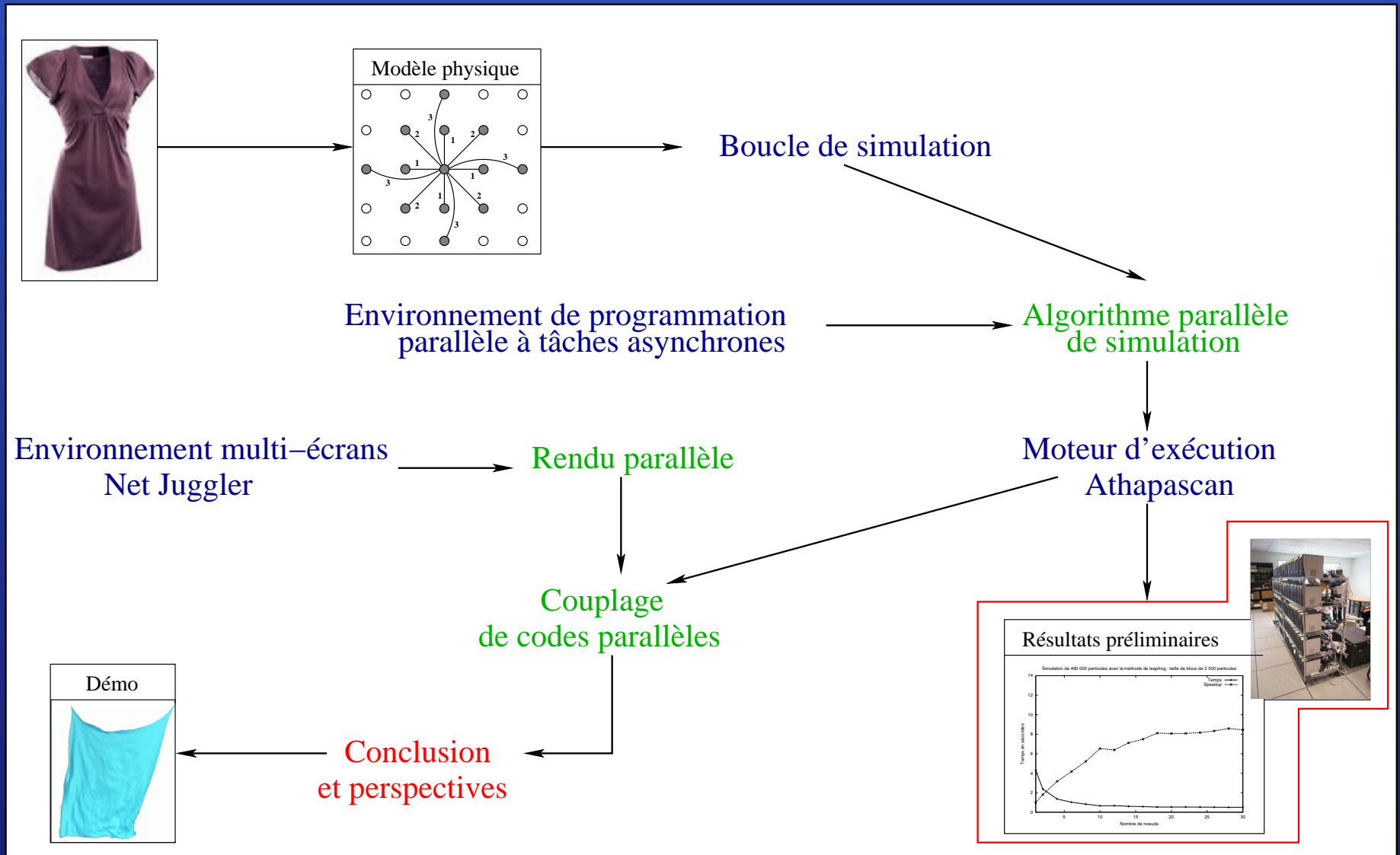
- Phases de l'exécution d'un programme ATHAPASCAN



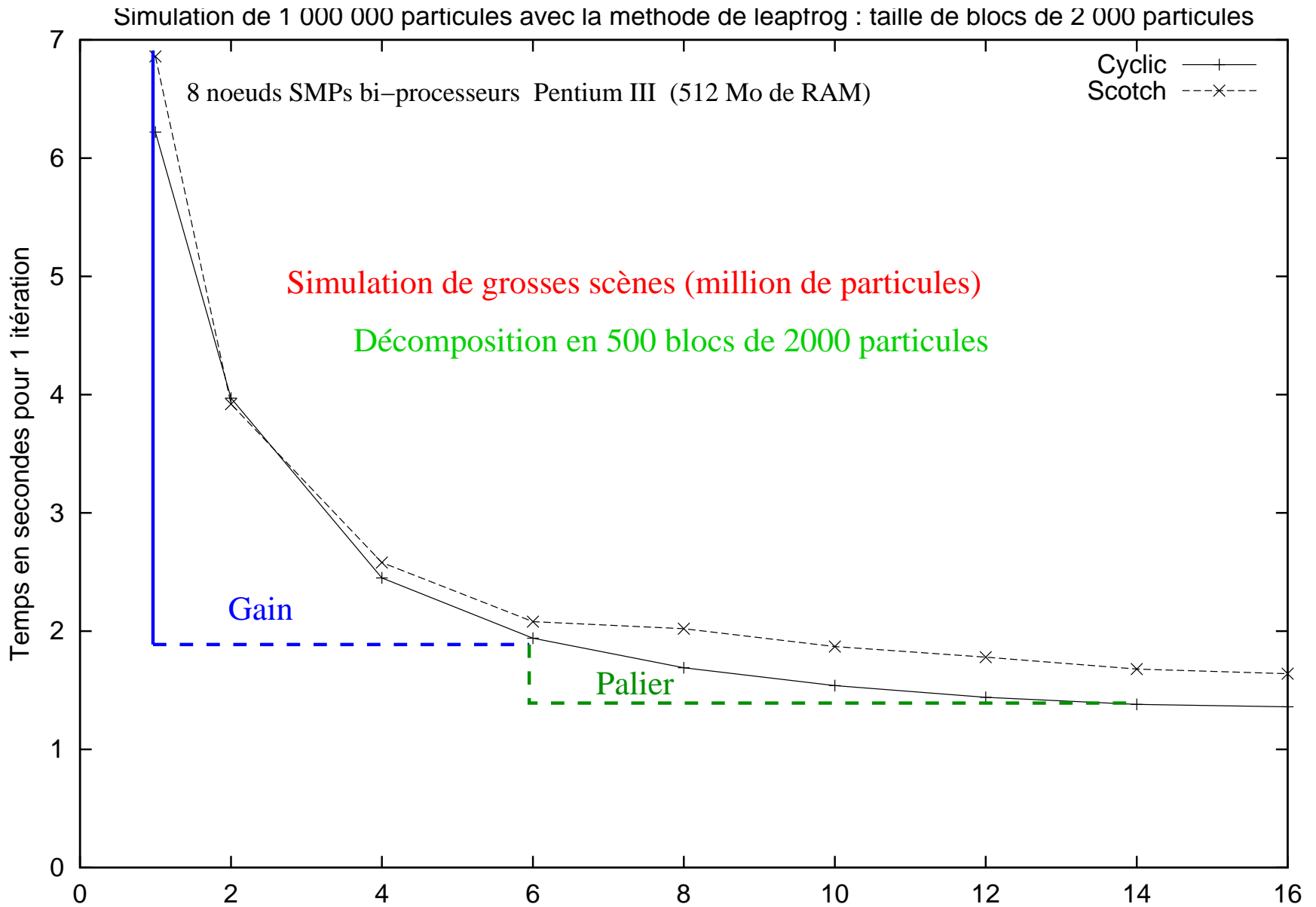
Optimisation

- Cas de la **boucle infinie** de la simulation
 - Construction du GFD pour plusieurs itérations
 - Ordonnancement du GFD construit
 - Exécution selon l'ordonnancement
 - **Nombre d'itérations à choisir**
- Cas de l'animation de textiles
 - Interactions entre blocs et particules changent pas
 - GFD semblable d'une itération à l'autre
 - GFD réutilisé d'une itération à l'autre
 - **Ôte le surcoût de la construction du GFD**

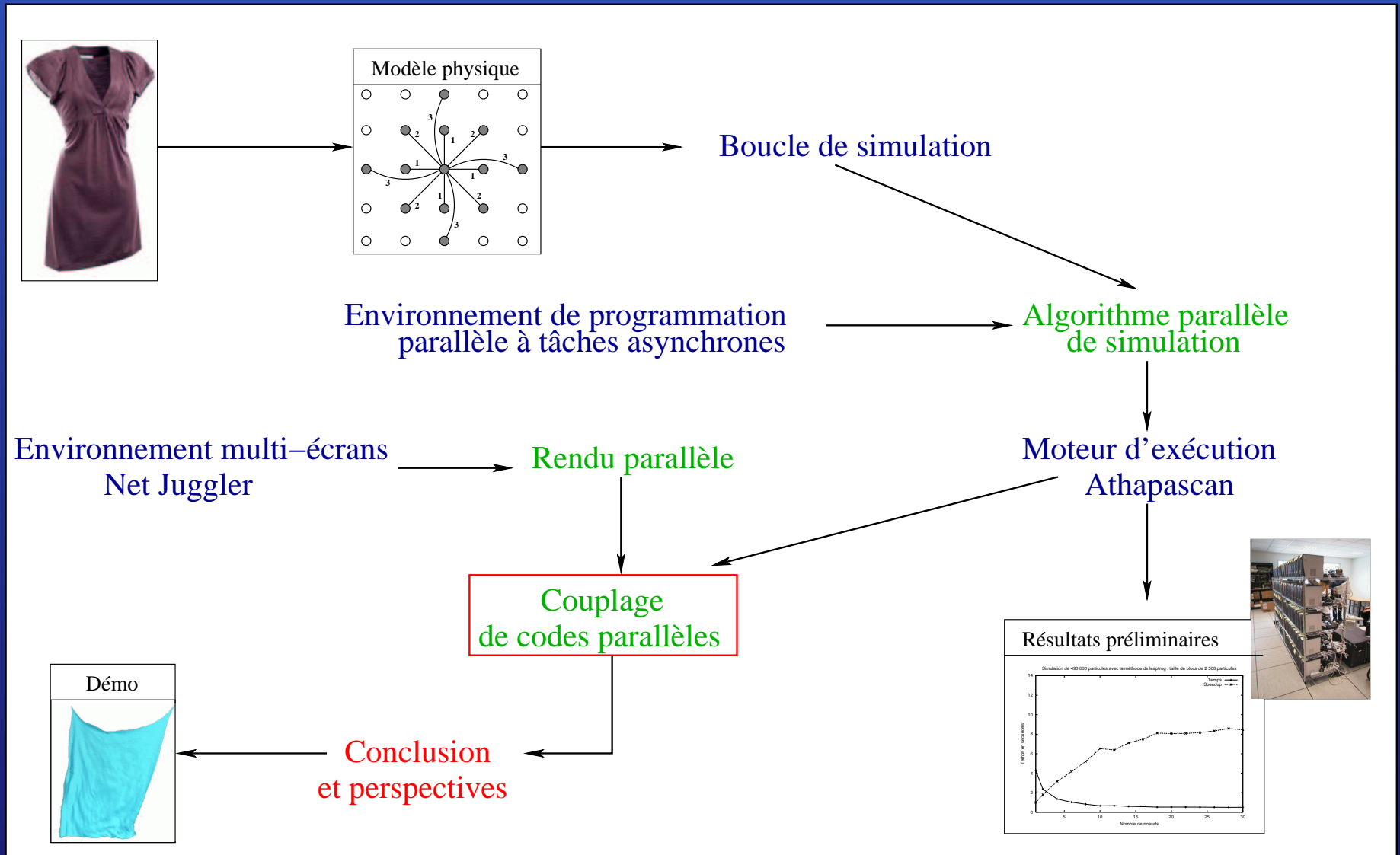
Plan de la présentation



Résultats préliminaires



Plan de la présentation



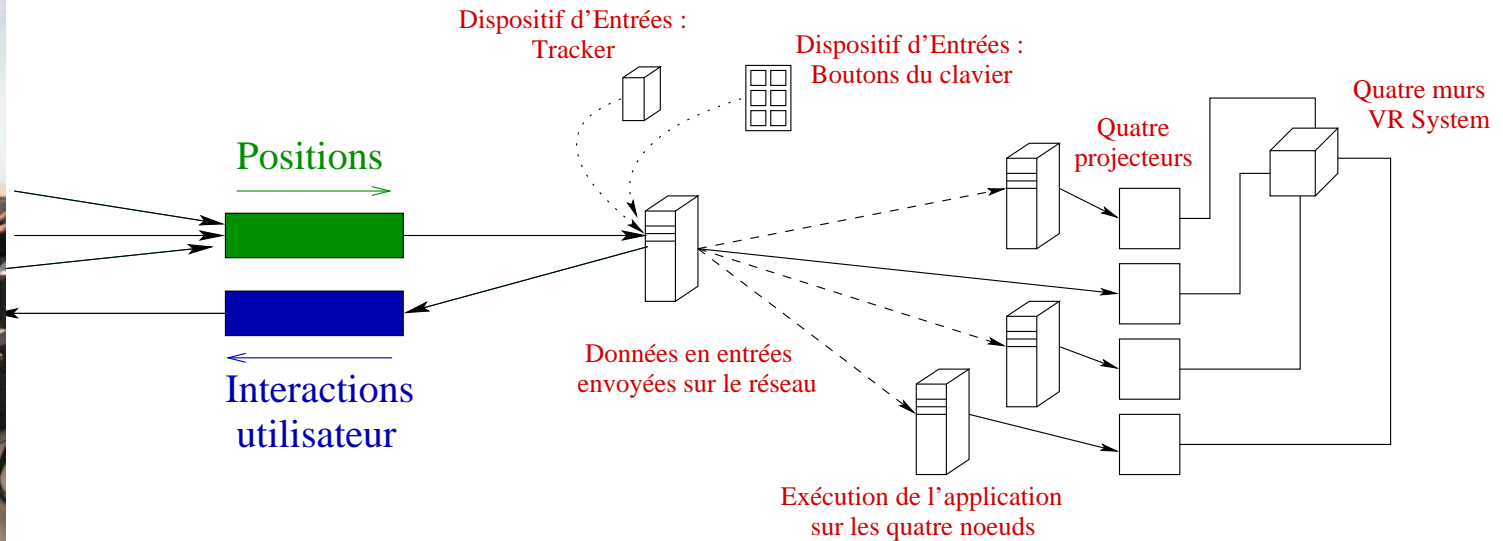
Couplage de codes parallèles

- Couplage entre une simulation et un rendu parallèle
 - Parallélisation du rendu via Net Juggler
 - Synchronisation logicielle entre les écrans

Problème de la récupération des données sur la grappe

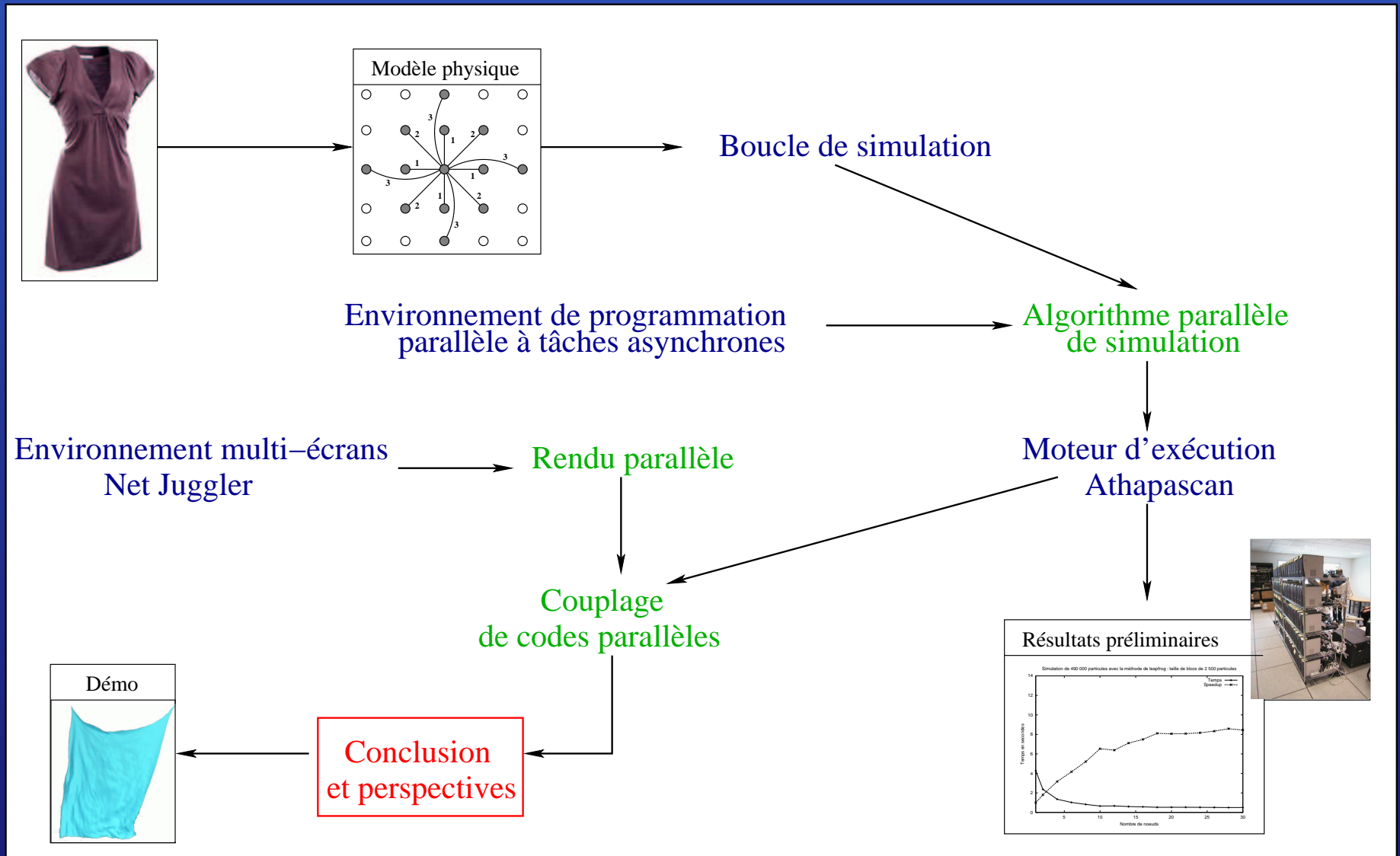


Simulation parallèle
via Athapascan



Visualisation multi écrans
via Net Juggler (LIFO, ID-IMAG)

Plan de la présentation



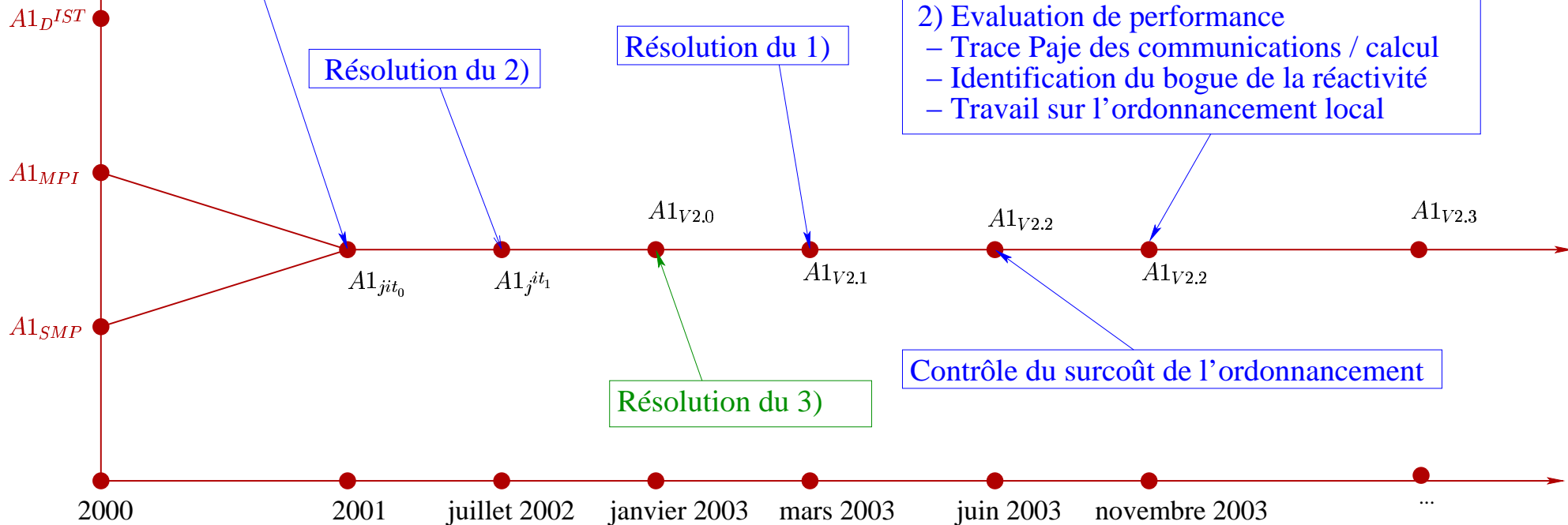
Conclusions (1)

- Plate-forme expérimentale d'animations interactives de textiles exécutées en parallèle
- Modèle physique et sa parallélisation
 - Caractéristiques du modèle
 - ▲ Interaction physique
 - ▲ Dépendances locales (peu de voisins)
 - ▲ Topologie du maillage varie peu
 - Caractéristiques de la parallélisation
 - ▲ Décomposition en blocs
 - ▲ Liste des interactions
 - ▲ Raffinement de la décomposition
 - ▲ Construction du graphe
- Schéma de type décomposition de domaine à grain fin
- Applicable à d'autres types d'animations de cette classe

Conclusions (2) : ATHAPASCAN

Problèmes

- 1) Construction du GFD dupliqué sur les processeurs
- 2) Manque de réactivité au niveau des communications
- 3) Contention du multi-thread



Conclusions (3)

■ Implantation

- ATHAPASCAN
- Rendu parallèle
 - ▲ Utilisation de Net Juggler
- Couplage de codes parallèles
 - ▲ Données éparpillées sur les noeuds de la grappe
→ Récupération des données sur la grappe
 - ▲ Récupération de l'interaction de l'utilisateur
→ Interaction entre codes parallèles
 - ▲ Contraintes de temps réel

Perspectives (1)

- Complexification du modèle physique
 - Ajout ressorts de flexion
 - Emploi d'autres modèles d'interactions
 - ▲ Modèles d'interaction continu
 - Emploi d'autres méthodes d'intégration
- Application générique
- Ne modifie pas la parallélisation

Perspectives (2) : collisions

■ Ajout des collisions dans la boucle de simulation

- Repete
 - ▲ Calculs des états des particules
 - ▲ Tant Que (collisions ou auto-collisions) Faire
 - Modifier les interactions
 - Rectifier positions et vitesses
 - ▲ Fin
 - ▲ Afficher
- Fin

■ Hiérarchie de boîtes englobantes

→ Boîtes englobantes des blocs de particules

■ Modification des interactions avec l'ajout de ressorts

→ Mais interactions toujours d'ordre local

■ Modification du GFD d'une itération à l'autre

→ Structures de données prévues pour cette dynamique

Perspectives (3)

- Projets sur le couplage de codes parallèles
 - Grimage, MOAIS
- Environnement de RV : FlowVR
 - Permet de visualiser et d'interagir avec plusieurs simulations distribuées
 - Gestion des contraintes entre simulations
 - ▲ Interactivité
 - ▲ Cohérence spatiale entre les écrans
 - ▲ Cohérence temporelle entre les pas d'itérations
 - ▲ Contraintes de temps réel (complexité, précision, affichage)

Vidéo issue du couplage

Algorithmes parallèles de simulation physique pour la synthèse d'images : application à l'animation de textiles

Florence Zara

Directeur de thèse : Brigitte Plateau (ID-IMAG)

Co-encadrants : Jean-Marc Vincent (ID-IMAG)

François Faure (GRAVIR-IMAG)

Laboratoire ID-IMAG, projet INRIA-APACHE
en collaboration avec le laboratoire GRAVIR-IMAG, projet INRIA-EVASION
et partiellement financé par la région Rhône-Alpes, projet SAPPE



Laboratoire
Informatique et
Distribution



CENTRE NATIONAL
DE LA RECHERCHE
SCIENTIFIQUE



Institut National
Polytechnique
de Grenoble



INSTITUT NATIONAL
DE RECHERCHE EN
INFORMATIQUE ET
EN AUTOMATIQUE



GRENOBLE 1
UNIVERSITÉ
JOSEPH FOURIER
SCIENCES, TECHNOLOGIE, MÉDECINE