

Modulariser les ordonnanceurs de tâches : une approche structurelle

Simon ARCHIPOFF, Marc SERGENT

Inria Bordeaux - Équipe Runtime

Olivier Aumage - Samuel Thibault

Conférence d'informatique en Parallélisme, Architecture et Système

23 Avril 2014



université
de BORDEAUX



Inria

- ★ Machines de plus en plus complexes et hétérogènes
 - ▶ CPUs, GPUs, Xeon Phi ...
- ★ Comment les exploiter efficacement ?
 - ▶ Gestion des calculs
 - Allouer judicieusement les calculs aux unités
 - Déterminer la granularité adéquate
 - ▶ Gestion des données
 - Recouvrir les transferts entre espaces mémoire discrets
 - Éliminer les transferts redondants

- ★ Machines de plus en plus complexes et hétérogènes
 - ▶ CPUs, GPUs, Xeon Phi ...
- ★ Comment les exploiter efficacement ?
 - ▶ Gestion des calculs
 - Allouer judicieusement les calculs aux unités
 - Déterminer la granularité adéquate
 - Ordonnancement
 - ▶ Gestion des données
 - Recouvrir les transferts entre espaces mémoire discrets
 - Éliminer les transferts redondants
 - Mémoire virtuellement partagée

Une solution proposée par la communauté

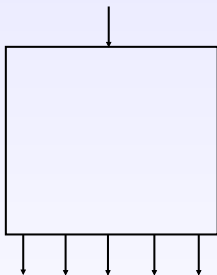
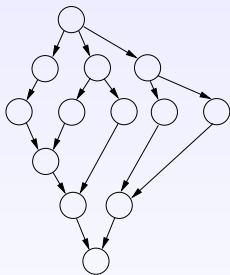
- ★ Supports d'exécution à base de tâches dépendantes : StarPU

- ★ Machines de plus en plus complexes et hétérogènes
 - ▶ CPUs, GPUs, Xeon Phi ...
- ★ Comment les exploiter efficacement ?
 - ▶ Gestion des calculs
 - Allouer judicieusement les calculs aux unités
 - Déterminer la granularité adéquate
 - **Ordonnancement**
 - ▶ Gestion des données
 - Recouvrir les transferts entre espaces mémoire discrets
 - Éliminer les transferts redondants
 - **Mémoire virtuellement partagée**

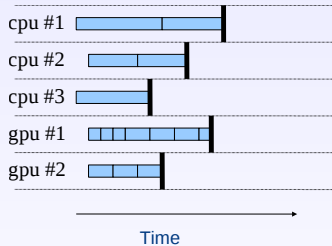
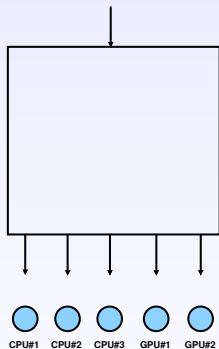
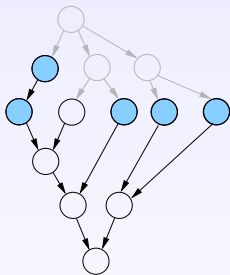
Une solution proposée par la communauté

- ★ Supports d'exécution à base de tâches dépendantes : StarPU
- ★ Ordonnanceurs de tâches

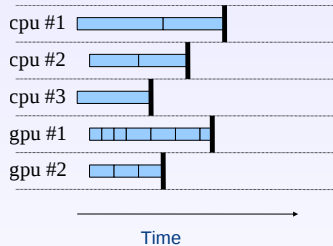
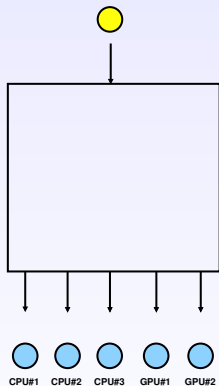
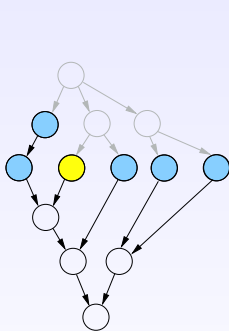
Fonctionnement d'un ordonnanceur de tâches



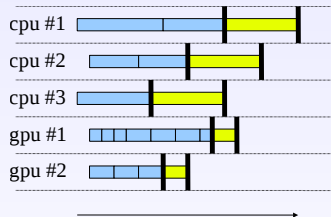
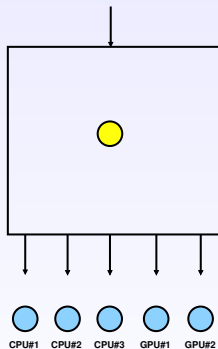
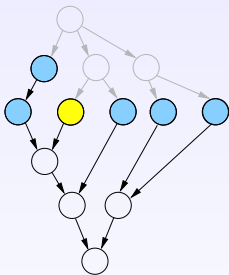
Fonctionnement d'un ordonnanceur de tâches



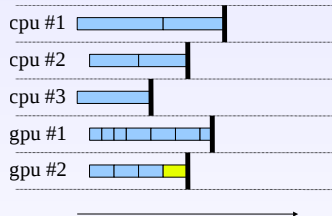
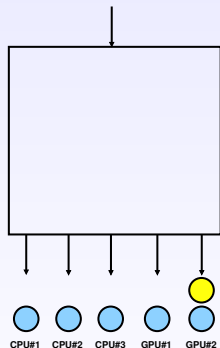
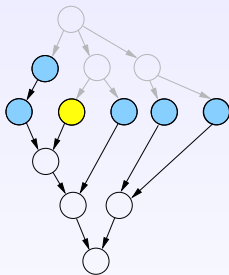
Fonctionnement d'un ordonnanceur de tâches



Fonctionnement d'un ordonnanceur de tâches



Fonctionnement d'un ordonnanceur de tâches



Ordonnanceurs existants

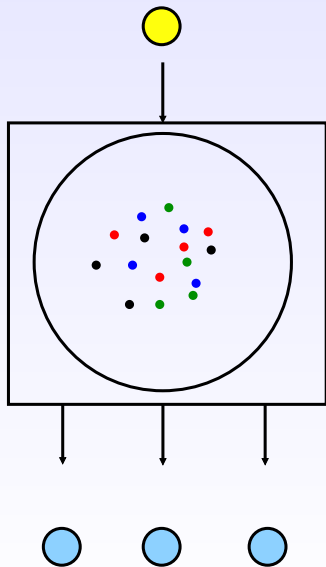
- ★ Monolithiques
 - ▶ Triviaux : glouton (Eager)
 - ▶ Évolués
 - Difficiles à implémenter

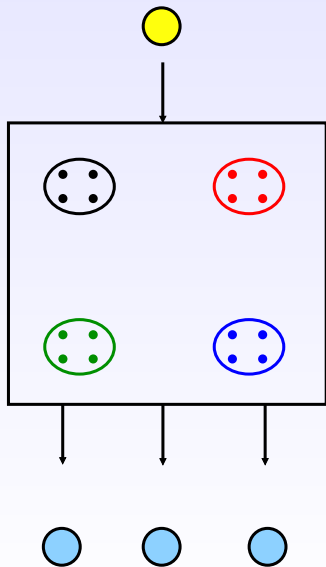
Ordonnanceurs existants

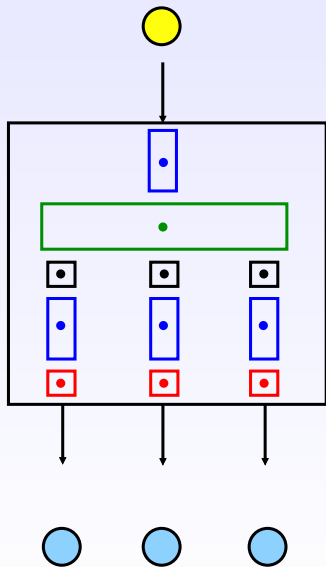
- ★ Monolithiques
 - ▶ Triviaux : glouton (Eager)
 - ▶ Évolués
 - Difficiles à implémenter

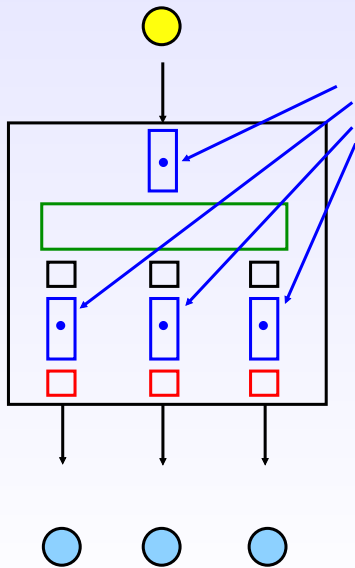
Idée

- ★ Assemblage
 - ▶ Modulaire
- ★ Quel est le bon paradigme ?
 - ▶ Facile à coder
 - ▶ Facile à assembler
 - ▶ Flexible
 - ▶ Puissant

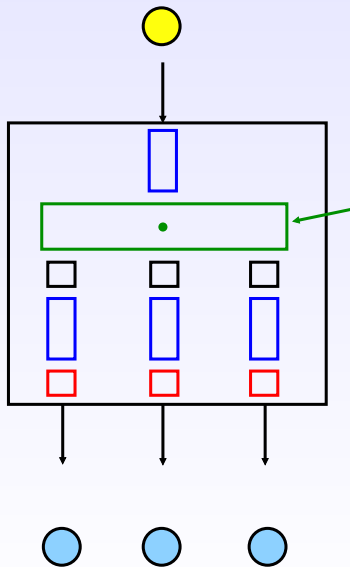




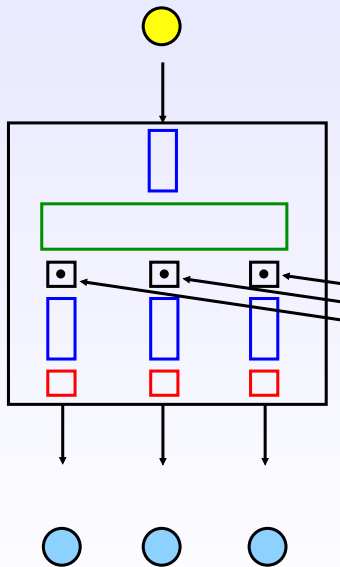




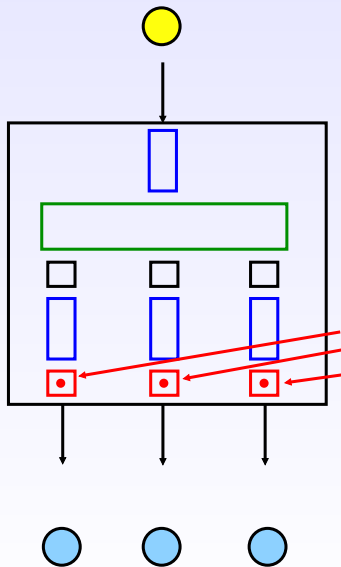
- ★ Réservoirs
 - ▶ Files d'attente



- ★ Réservoirs
 - ▶ Files d'attente
- ★ Aiguillage
 - ▶ Peut faire le choix d'ordonnement

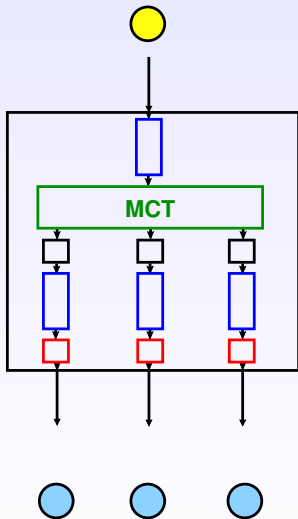


- ★ Réservoirs
 - ▶ Files d'attente
- ★ Aiguillage
 - ▶ Peut faire le choix d'ordonnancement
- ★ Effets de bord
 - ▶ Préchargement de données
 - ▶ Choix de l'implémentation



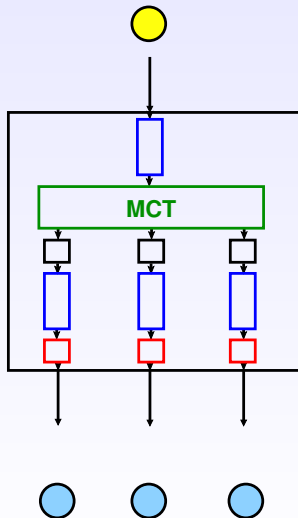
- ★ Réservoirs
 - ▶ Files d'attente
- ★ Aiguillage
 - ▶ Peut faire le choix d'ordonnancement
- ★ Effets de bord
 - ▶ Préchargement de données
 - ▶ Choix de l'implémentation
- ★ Workers
 - ▶ Modélisation des unités de calcul

★ Heft

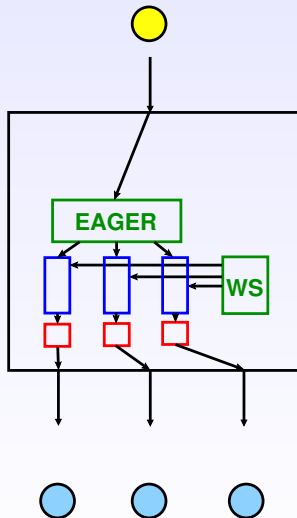


Exemples

★ Heft

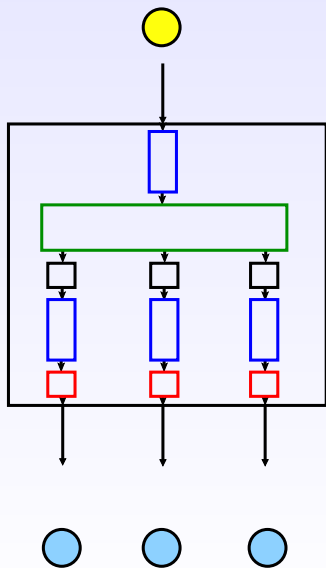


★ Vol de travail

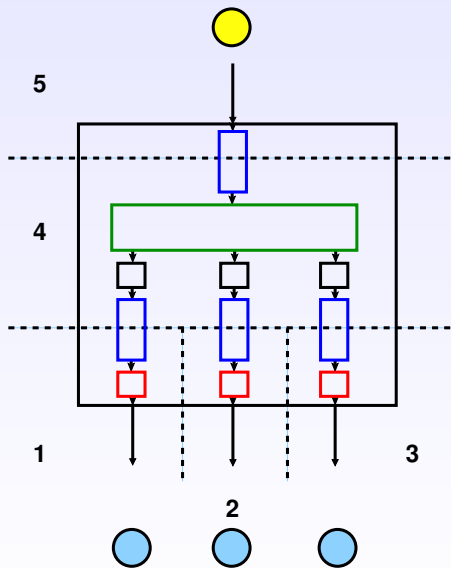


- ★ Assemblage valide

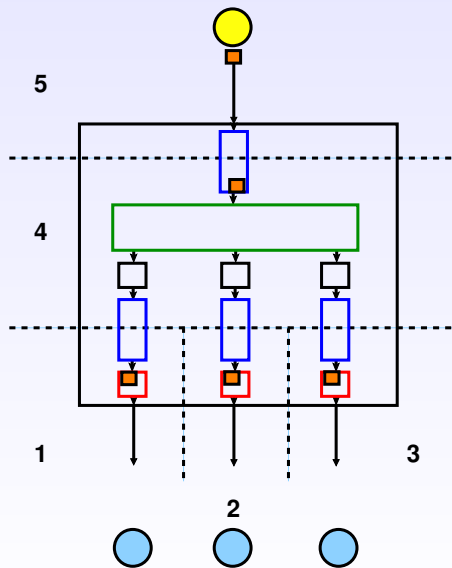
- ▶ Progression, vivacité
- ▶ Passage des tâches d'un composant à un autre



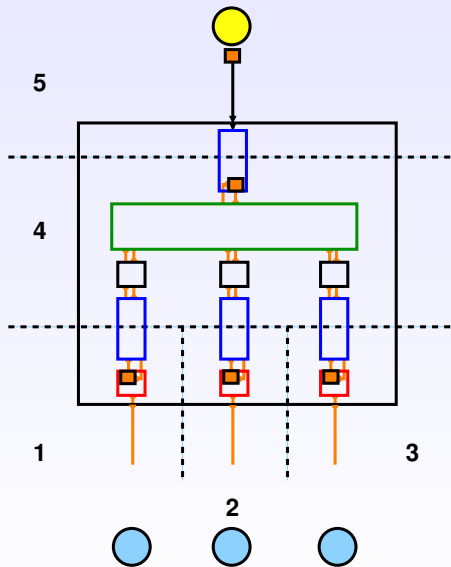
★ Zones d'ordonnement



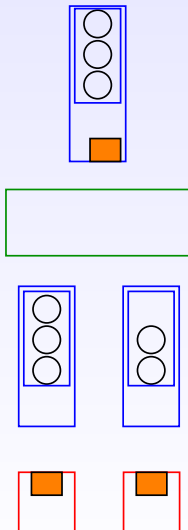
- ★ Zones d'ordonnancement
 - ▶ Délimitées par des bassins d'accumulation

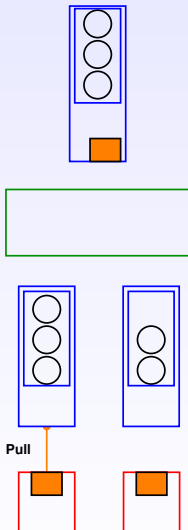


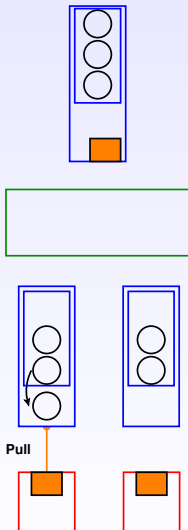
- ★ Zones d'ordonnement
 - ▶ Délimitées par des bassins d'accumulation
 - ▶ Une pompe par zone d'ordonnement

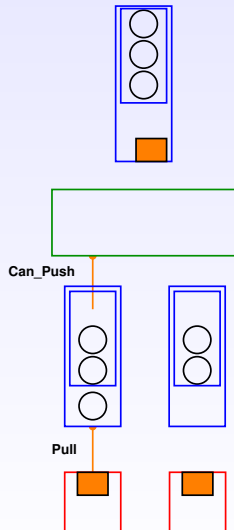


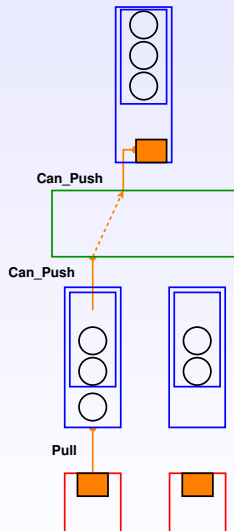
- ★ Zones d'ordonnancement
 - ▶ Délimitées par des bassins d'accumulation
 - ▶ *Une pompe par zone d'ordonnancement*
 - ▶ Push/Pull
 - ▶ Can_Push/Can_Pull

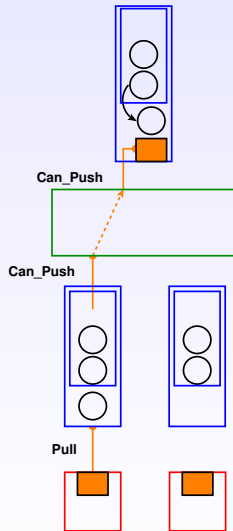


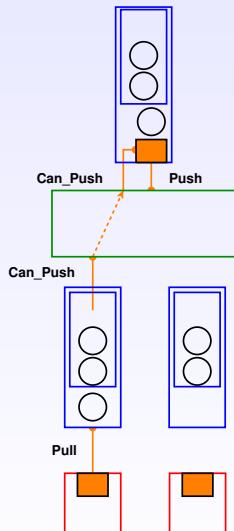


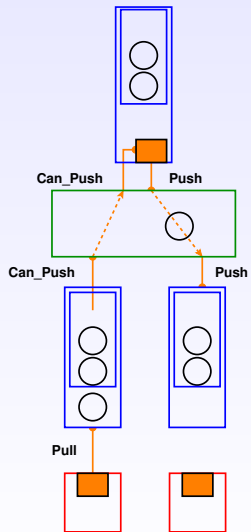


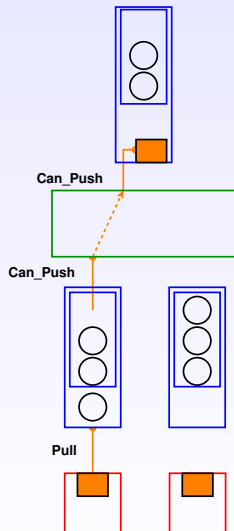


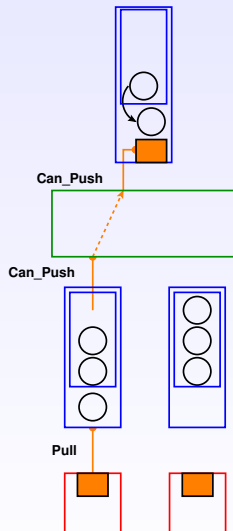


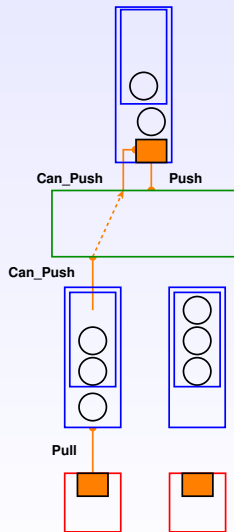


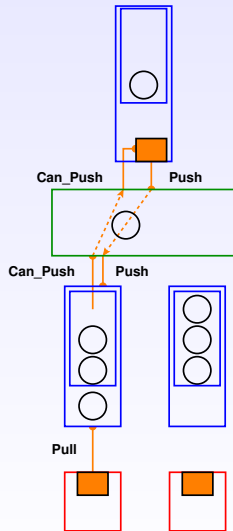


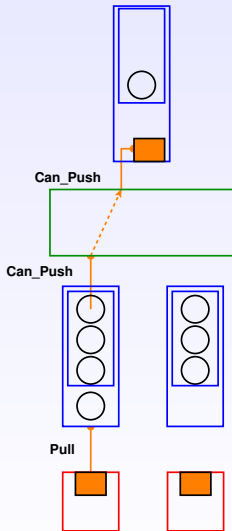


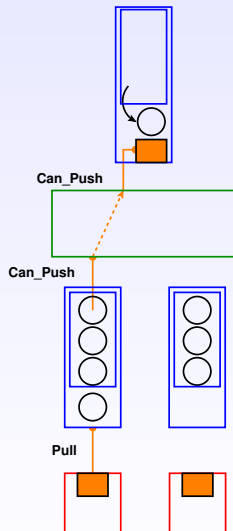


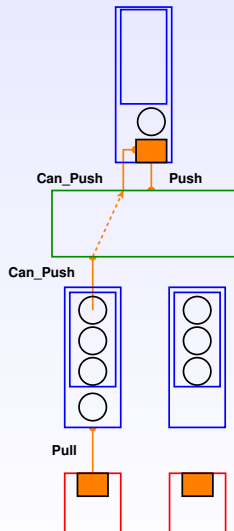


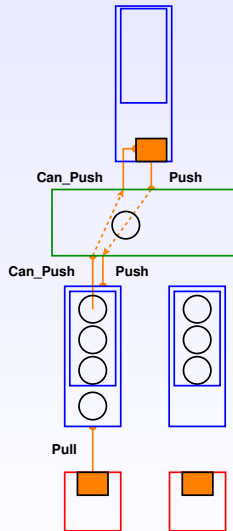


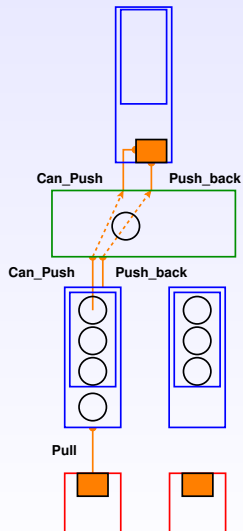


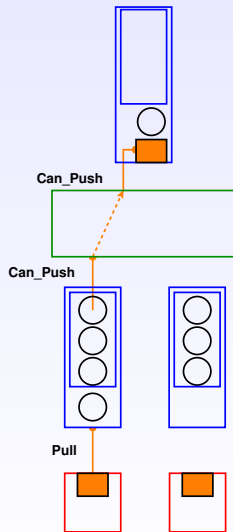


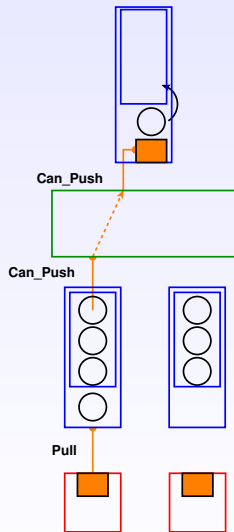


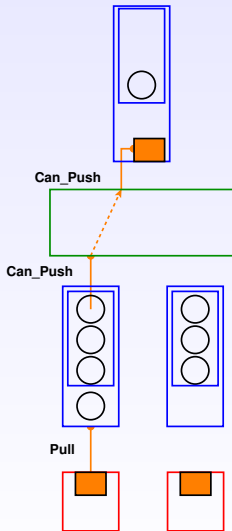


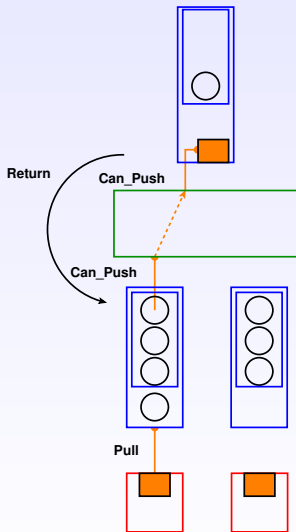


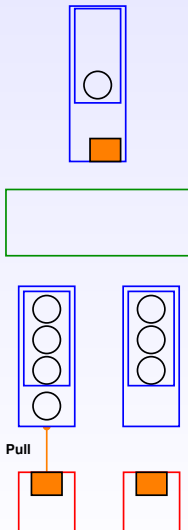


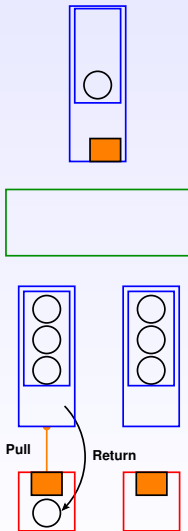


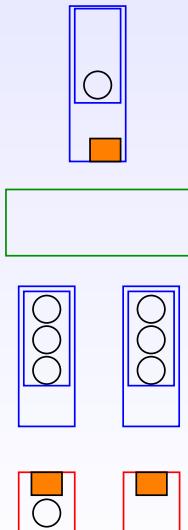






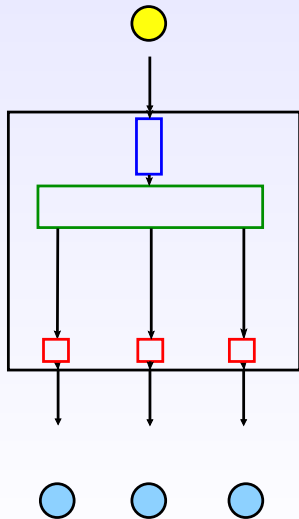






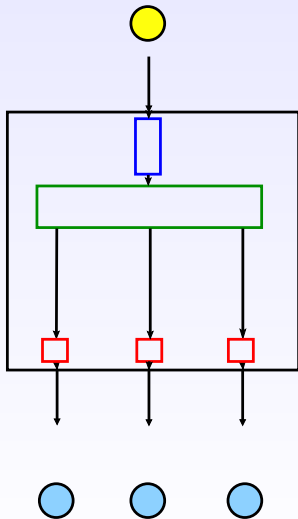
Une boîte à outils de composants

★ Priorités

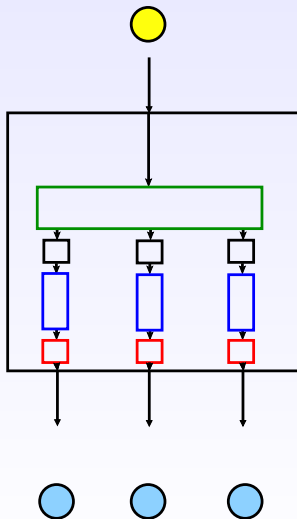


Une boîte à outils de composants

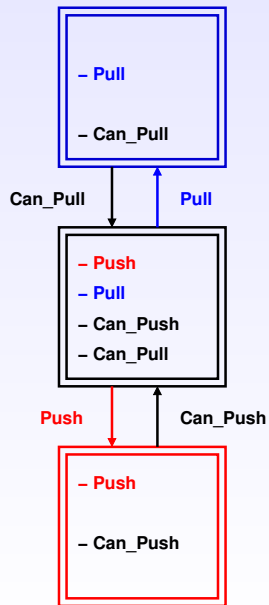
★ Priorités



★ Préchargement de données



- ★ Surcharger les méthodes de l'interface du composant



Objectifs

- ★ Non-régression des performances
- ★ Observer l'état des réservoirs

Objectifs

- ★ Non-régression des performances
- ★ Observer l'état des réservoirs

Objet de l'étude

- ★ Support d'exécution
 - ▶ StarPU

Objectifs

- ★ Non-régression des performances
- ★ Observer l'état des réservoirs

Objet de l'étude

- ★ Support d'exécution
 - ▶ StarPU
- ★ Ordonnanceurs comparés
 - ▶ DMDAS
 - ▶ Modular-HEFT

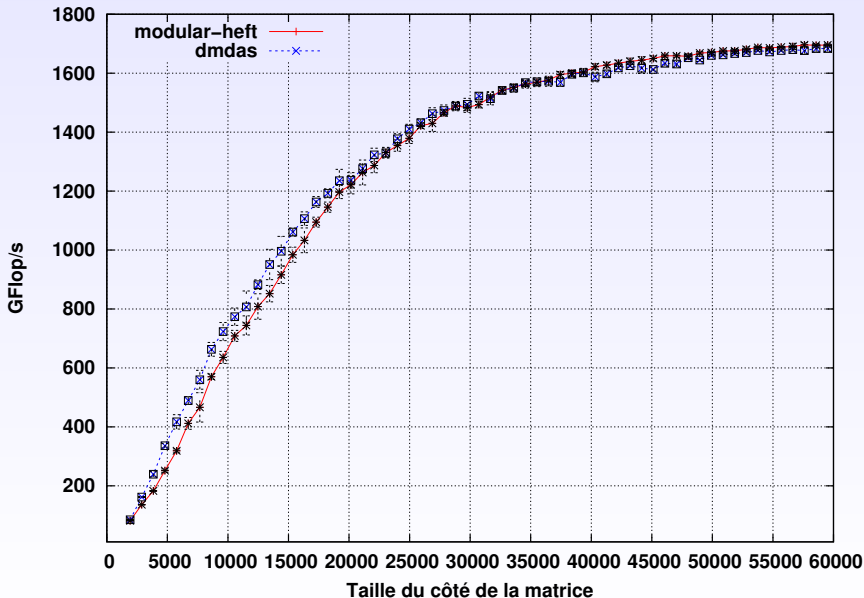
Cadre expérimental

- ★ PlaFRIM
- ★ 1 machine Mirage
 - ▶ 2 Hexa-core Westmere Intel Xeon X5650 : 12 cœurs
 - ▶ 3 NVIDIA Tesla M2070 GPU
 - ▶ 36 Go RAM

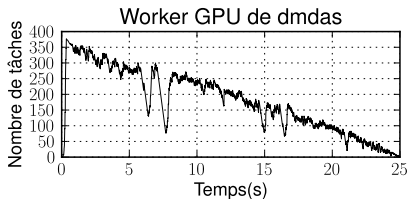
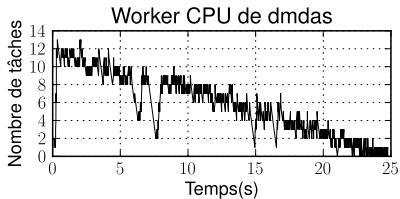
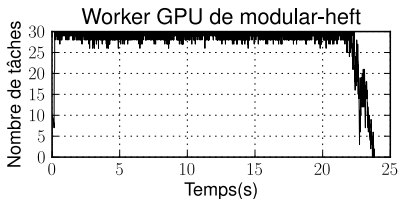
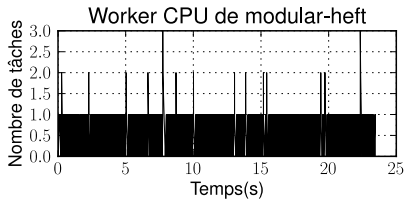
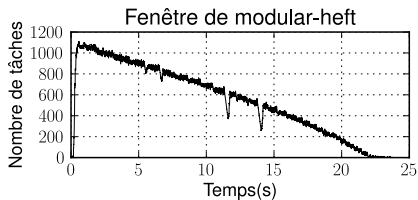
Cadre expérimental

- ★ PlaFRIM
- ★ 1 machine Mirage
 - ▶ 2 Hexa-core Westmere Intel Xeon X5650 : 12 cœurs
 - ▶ 3 NVIDIA Tesla M2070 GPU
 - ▶ 36 Go RAM
- ★ Factorisation de Cholesky
 - ▶ Magma-Morse
 - ▶ Simple-précision
 - ▶ Taille de tuilage : 960x960

Factorisation de Cholesky sur 1 machine Mirage



État des réservoirs pendant l'exécution



Contribution

- ★ Un modèle rationalisé d'ordonnanceurs de tâches
- ★ Modulaire
- ★ Pas de régression des performances
- ★ Une boîte à outils de composants

Contribution

- ★ Un modèle rationalisé d'ordonnanceurs de tâches
- ★ Modulaire
- ★ Pas de régression des performances
- ★ Une boîte à outils de composants

Perspectives

- ★ Bouchons
 - ▶ Ordonnancer les tâches par paquets
- ★ Travail avec des théoriciens de l'ordonnancement
- ★ Scalabilité
 - ▶ Composition d'heuristiques
- ★ DSL
 - ▶ Niveau d'abstraction supplémentaire pour le programmeur
 - ▶ Preuves automatiques de validité et de vivacité des ordonnanceurs

Influence de la taille des réservoirs sur les performances

