

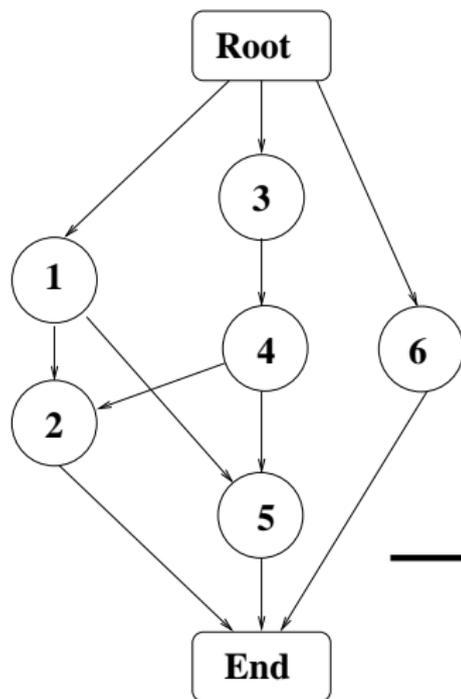
HCPA et S-HCPA: deux algorithmes d'ordonnement de tâches parallèles sur plates-formes hétérogènes

Tchimou N'takpé Frédéric Suter

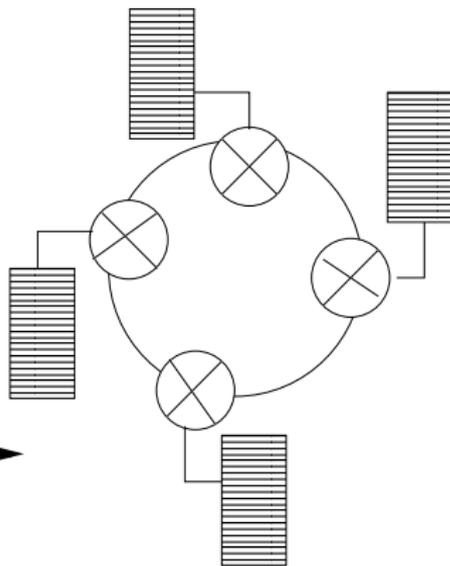
AlGorille / LORIA

Réunion MA0, 24 novembre 2005

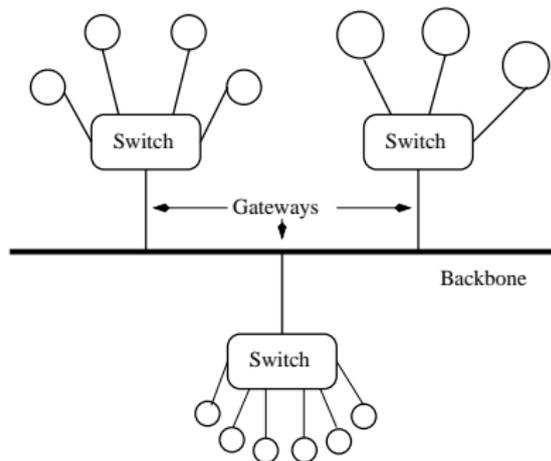
Introduction



Application

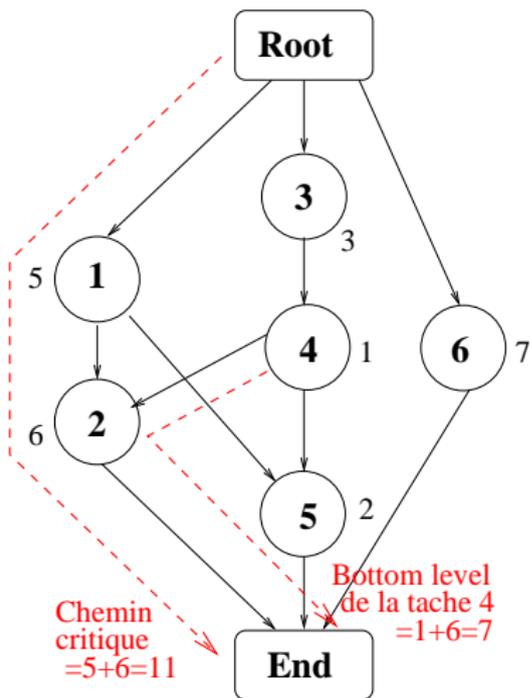


Grille de calcul



Grilles légères

- Plusieurs grappes homogènes
- Hétérogénéité de l'ensemble
 - Réseaux d'interconnexion
 - Vitesse des processeurs
 - Capacité mémoire



Modèle d'application

- Application parallèle
 - graphe orienté acyclique $G = (V, E)$
- On suppose que chaque tâche peut s'exécuter sur un ou plusieurs processeurs
 - tâches parallèles
- Le temps d'exécution d'une tâche peut être modélisé par la loi d'Amdahl :

$$T_{\omega}(t, N_p(t)) = \left(\alpha + \frac{1 - \alpha}{N_p(t)} \right) \cdot T_{\omega}(t, 1)$$

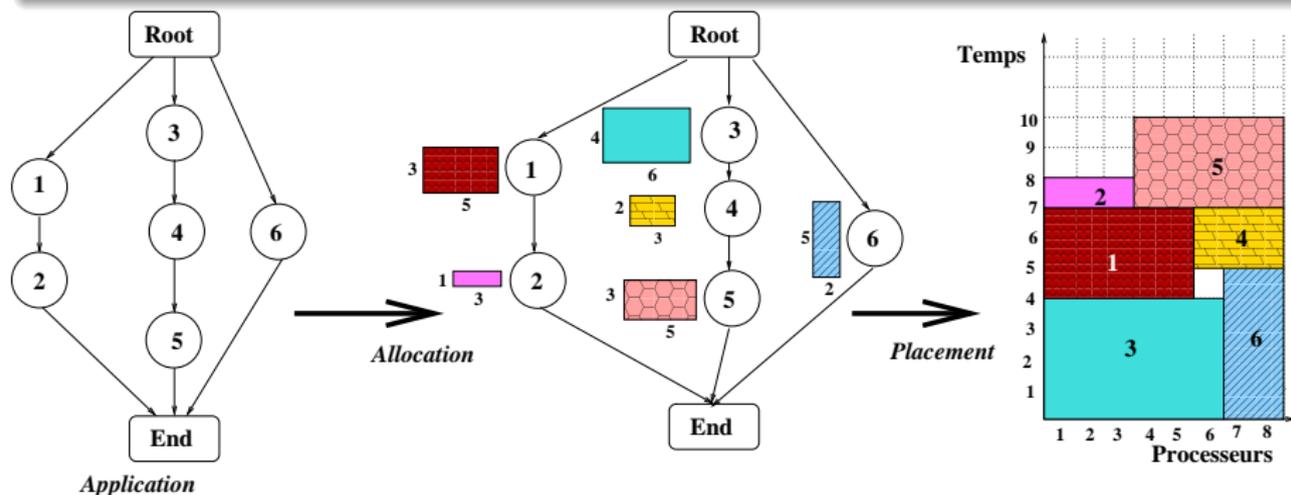
Ordonnancement de tâches parallèles en milieu hétérogène

- Pour chaque tâche on cherche :
 - le "bon" nombre de processeurs
 - la "bonne" grappe
- En respectant les contraintes :
 - de ressources
 - de précédences
- Objectifs :
 - 1 Minimiser temps de complétion de l'application
 - 2 Garder un bon compromis entre temps de complétion et puissance utilisée
 - 3 Rester efficace

- 1 Introduction
- 2 Travaux précédents
- 3 Adaptation de CPA aux plates-formes hétérogènes
- 4 Validation expérimentale
- 5 Résultats des simulations
- 6 Conclusion et Travaux futurs

Algorithmes en deux étapes (cas homogène)

- TSAS (*Two Step Allocation and Scheduling*) [Ramaswamy et al, TPDS-97]
- CPR (*Critical Path Reduction*) [Radulescu et al, IPDPS'01]
- CPA (*Critical Path and Area-based*) [Radulescu et al, ICPP'01]



Algorithmes en deux étapes (cas homogène)

- TSAS (*Two Step Allocation and Scheduling*) [Ramaswamy et al, TPDS-97]
- CPR (*Critical Path Reduction*) [Radulescu et al, IPDPS'01]
- CPA (*Critical Path and Area-based*) [Radulescu et al, ICPP'01]

Ordonnancement de tâches parallèles en milieu hétérogène

- Adaptation d'un algorithme d'ordonnancement de tâches séquentielles en milieu hétérogène [Casanova et al, Europar'04] (MHEFT - Mixed Parallel Heterogenous Earliest Finish Time)

- 1 Introduction
- 2 Travaux précédents
- 3 Adaptation de CPA aux plates-formes hétérogènes**
 - Principe général
 - Phase d'allocation
 - Phase de placement
- 4 Validation expérimentale
- 5 Résultats des simulations
- 6 Conclusion et Travaux futurs

Phase d'allocation

- Utilisation d'une grappe de référence
 - masquer l'hétérogénéité
 - Adaptation du chemin critique et du temps moyen d'occupation des processeurs
- Interdiction à une tâche de s'exécuter sur plusieurs grappes en même temps

Phase d'ordonnancement

- Deux approches retenues :
 - Temps de complétion minimum → HCPA
 - Approche de type *sufferage* → S-HCPA

Caractéristiques et utilisation

- Même puissance cumulée
- Vitesse des processeurs les plus lents
- 1 allocation de référence \iff C allocations réelles :

$$N_p^i(t) = \min\{P_i, \lceil f(N_p^{ref}(t), t, i) \rceil\}, \forall i$$

- Objectif : avoir le même temp d'exécution sur toutes les grappes
→ Modulo erreurs de discrétisation

Algorithme commun à HCPA et S-HCPA

- Allocations initiales
 - 1 processeur par tâche
- À chaque itération
 - Ajout d'un processeur à la tâche du chemin critique qui en bénéficie le plus
- Critères d'arrêt
 - Bon compromis trouvé entre temps de complétion et utilisation des processeurs
 - Chemin critique saturé

HCPA

- Minimiser la date de fin d'exécution de la tâche prête prioritaire
- Choisir l'allocation appropriée en tenant compte du coût des mouvements de données.
- Critère de priorité \rightarrow *bottom level*

S-HCPA

- Calcul des deux "meilleures" allocations pour chaque tâche prête
- Placer la tâche qui serait la plus pénalisée si on lui attribuait sa deuxième meilleure allocation au lieu de la première

- 1 Introduction
- 2 Travaux précédents
- 3 Adaptation de CPA aux plates-formes hétérogènes
- 4 Validation expérimentale**
 - Méthodologie
 - Génération des plates-formes
 - Génération des graphes de tâches
- 5 Résultats des simulations
- 6 Conclusion et Travaux futurs

Comment évaluer nos algorithmes ?

- 1 Difficile de comparer selon des critères théoriques
→ Recours aux expériences

Comment évaluer nos algorithmes ?

- 1 Difficile de comparer selon des critères théoriques
 - Recours aux expériences
- 2 Reproductibilité des conditions expérimentales indispensable
 - Simulation

Comment évaluer nos algorithmes ?

- 1 Difficile de comparer selon des critères théoriques
 - Recours aux expériences
- 2 Reproductibilité des conditions expérimentales indispensable
 - Simulation
- 3 Tester un grand nombre de cas
 - 200 grilles
 - 1296 graphes
 - 5 algorithmes (SEQ, CPA, HCPA, S-HCPA, MHEFT)
 - 1 296 000 simulations

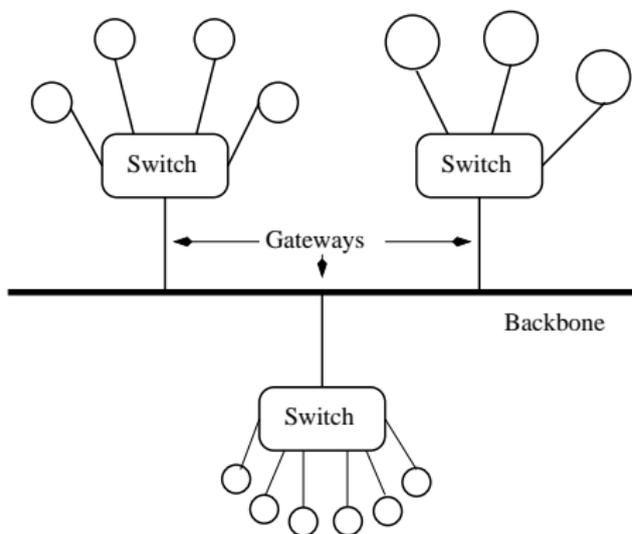
Comment évaluer nos algorithmes ?

- 1 Difficile de comparer selon des critères théoriques
 - Recours aux expériences
- 2 Reproductibilité des conditions expérimentales indispensable
 - Simulation
- 3 Tester un grand nombre de cas
 - 200 grilles
 - 1296 graphes
 - 5 algorithmes (SEQ, CPA, HCPA, S-HCPA, MHEFT)
 - 1 296 000 simulations
 - ⇒ Bon simulateur

Comment évaluer nos algorithmes ?

- 1 Difficile de comparer selon des critères théoriques
 - Recours aux expériences
- 2 Reproductibilité des conditions expérimentales indispensable
 - Simulation
- 3 Tester un grand nombre de cas
 - 200 grilles
 - 1296 graphes
 - 5 algorithmes (SEQ, CPA, HCPA, S-HCPA, MHEFT)
 - 1 296 000 simulations
 - ⇒ Bon simulateur → SimGrid bien sûr !

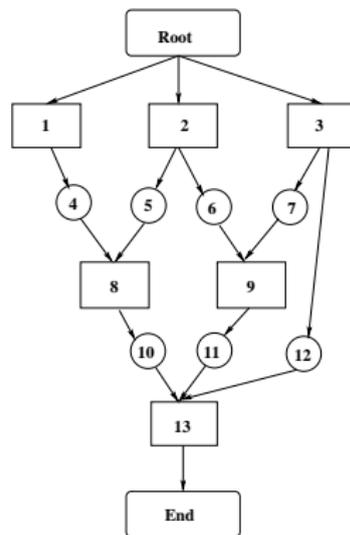
Plates-formes générées



nombre de grappes	1, 2, 4, 8
nombre minimum de processeurs par grappe	16
nombre maximum de processeurs par grappe	128
bornes inférieures des vitesses de processeurs (GFlops)	0.25, 0.5, 0.75 1
hétérogénéité	1, 2, 5
nombre d'échantillons	5

200 grilles générées

Graphes de tâches générés



nombre de tâches de calcul	10, 20, 50
taille minimum des données	2048
taille maximum des données	11268
complexité des tâches de calcul	$a \cdot N$, $a \cdot N \log N$, $a \cdot N^{3/2}$, aléatoire
valeur minimale de α	0.0
valeur maximale de α	0.2
largeur des graphes	0.1, 0.2, 0.8
densité des graphes	0.2, 0.8
régularité entre les niveaux	0.2, 0.8
longueur des sauts	1, 2, 4
nombre d'échantillons	3

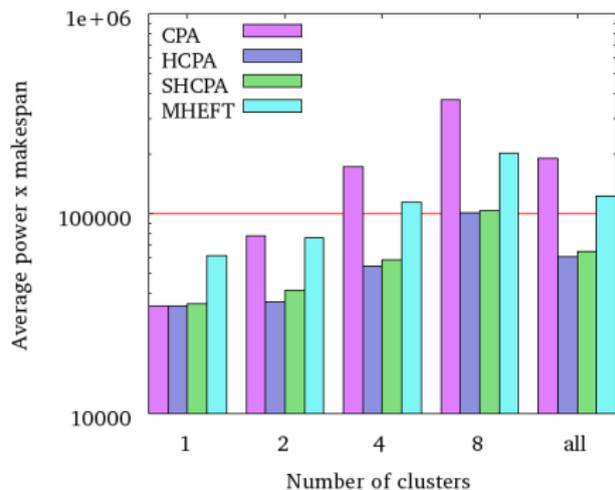
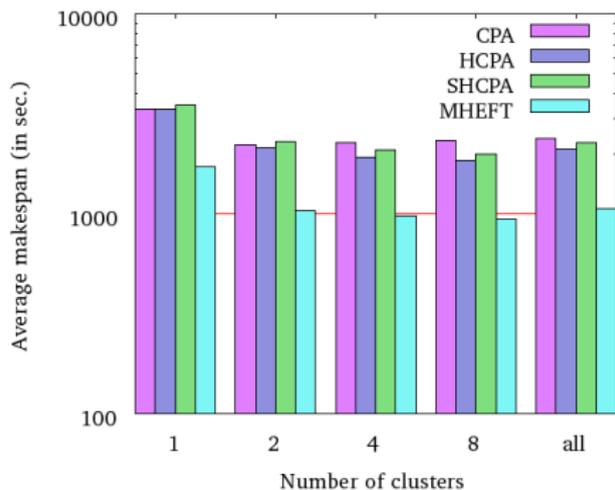
1296 graphes générés.

- 1 Introduction
- 2 Travaux précédents
- 3 Adaptation de CPA aux plates-formes hétérogènes
- 4 Validation expérimentale
- 5 Résultats des simulations**
 - Métriques d'évaluations
- 6 Conclusion et Travaux futurs

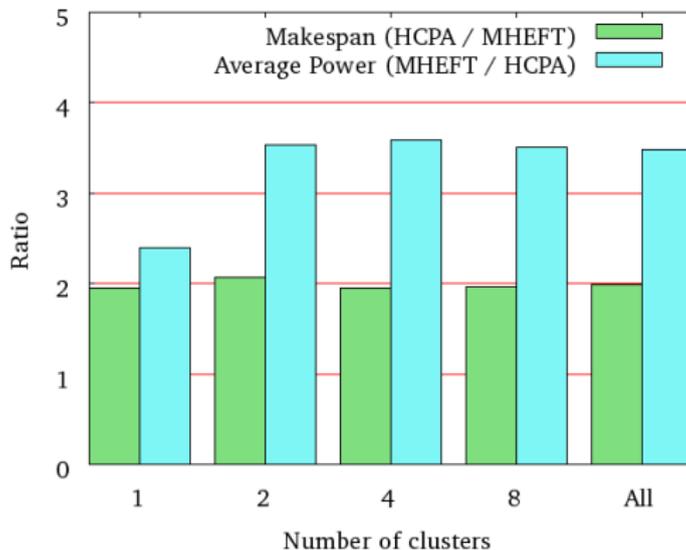
- Le *Makespan* des applications.
- Les compromis
 - *Makespan* × *pic de puissance utilisée*.
 - *Makespan* × *puissance moyenne utilisée*.
- Le gain d'un algorithme (ALG) par rapport à SEQ :

$$\text{Gain} = \frac{\text{Makespan de SEQ}}{\text{Makespan de ALG}}$$

Évolution du nombre de grappes

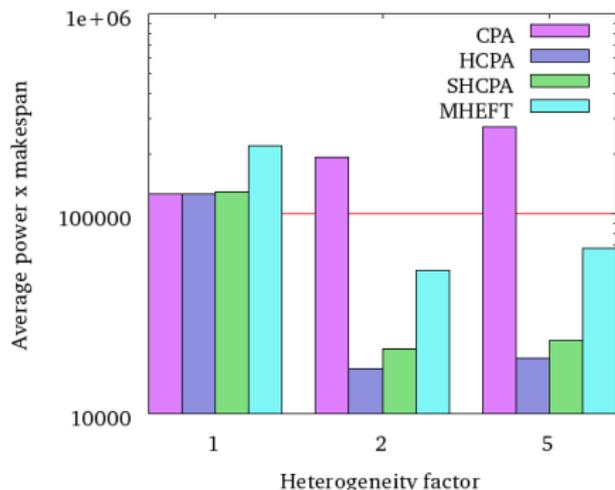
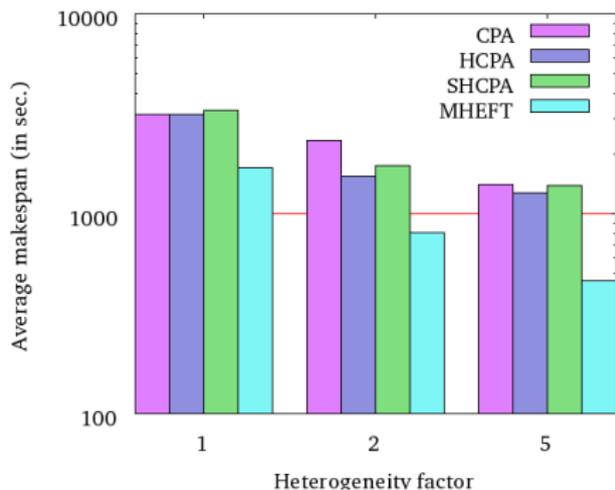


Complémentarité de MHEFT et HCPA



- MHEFT plus rapide mais HCPA moins gourmand.

Évolution de l'hétérogénéité



- 1 Introduction
- 2 Travaux précédents
- 3 Adaptation de CPA aux plates-formes hétérogènes
- 4 Validation expérimentale
- 5 Résultats des simulations
- 6 Conclusion et Travaux futurs**
 - Conclusion
 - Travaux futurs

- Conception de deux algorithmes
 - Extension au cas hétérogène d'un algorithme de la littérature
→ HCPA et S-HCPA
- Virtualisation de la plate-forme → grappe de référence
 - Ramène à la procédure d'allocation de CPA
→ Complexité de l'ordre de CPA.
- Évaluation de HCPA et de S-HCPA
 - Mêmes ordonnancements en milieu homogène (HCPA)
 - Nette amélioration en milieu hétérogène
 - Même si CPA est avantagé par le modèle de tâches

Court terme

- Proposer une version moins gourmande de MHEFT
- Tenir compte des communications intra-tâches
 - Abandonner le modèle de la loi de Amdahl.
 - Recenser des catégories de tâches réelles
 - Complexité en calcul.
 - Schéma des communications.
- Mettre un peu de théorie la dedans (OTaPHe + PFD)

Plates-formes hétérogènes et partagées

- Un ordonnanceur ou plusieurs ?
- Applications avec ou sans contrainte de temps ?
- Priorités entre applications ?
- Spécification des besoins et des envies et des capacités

Questions ?

ARC recherche post-doc !!