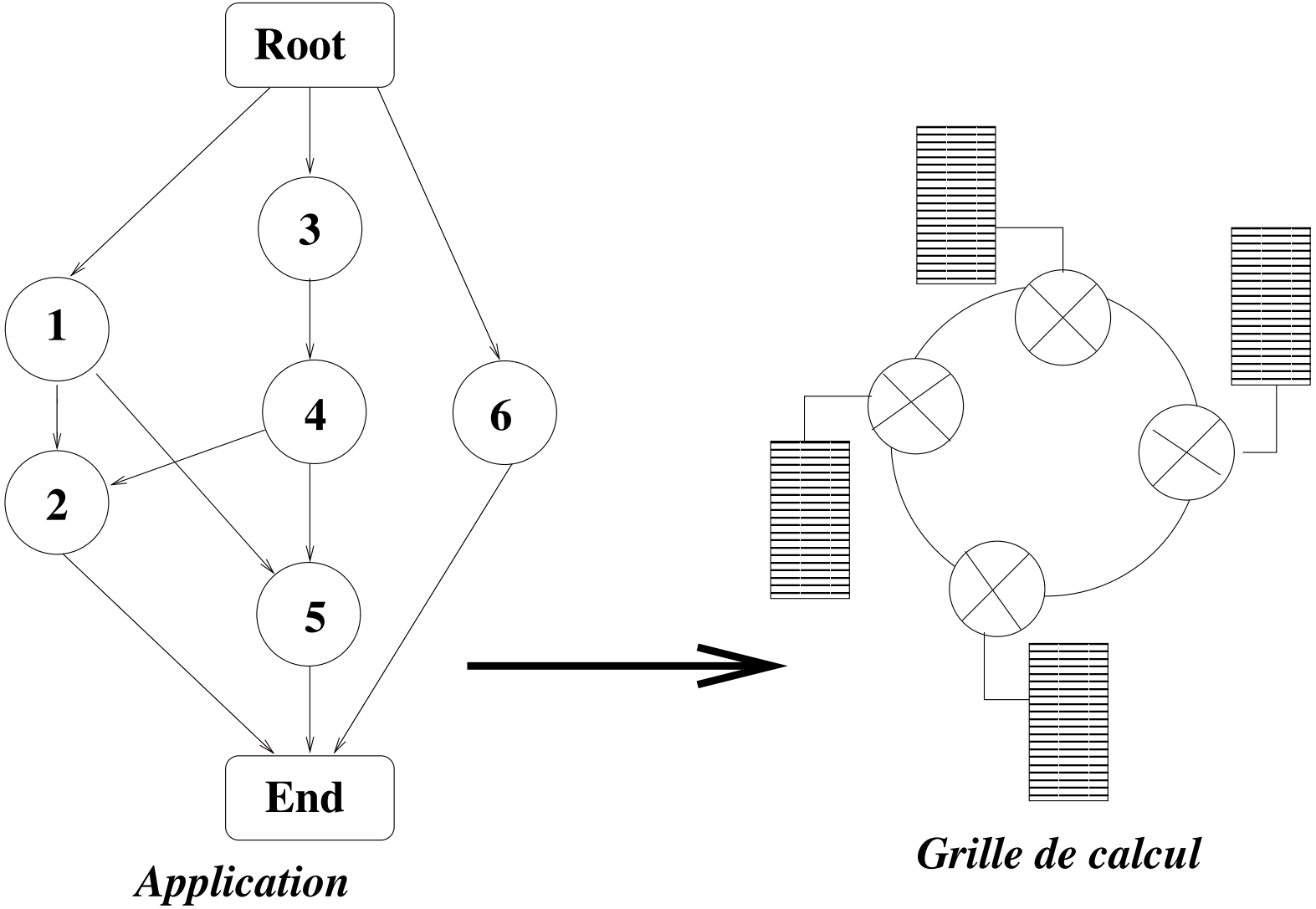
Ordonnancement de tâches parallèles sur Grid'5000 de la théorie à l'expérimentation

Pierre-François Dutot Frédéric Suter

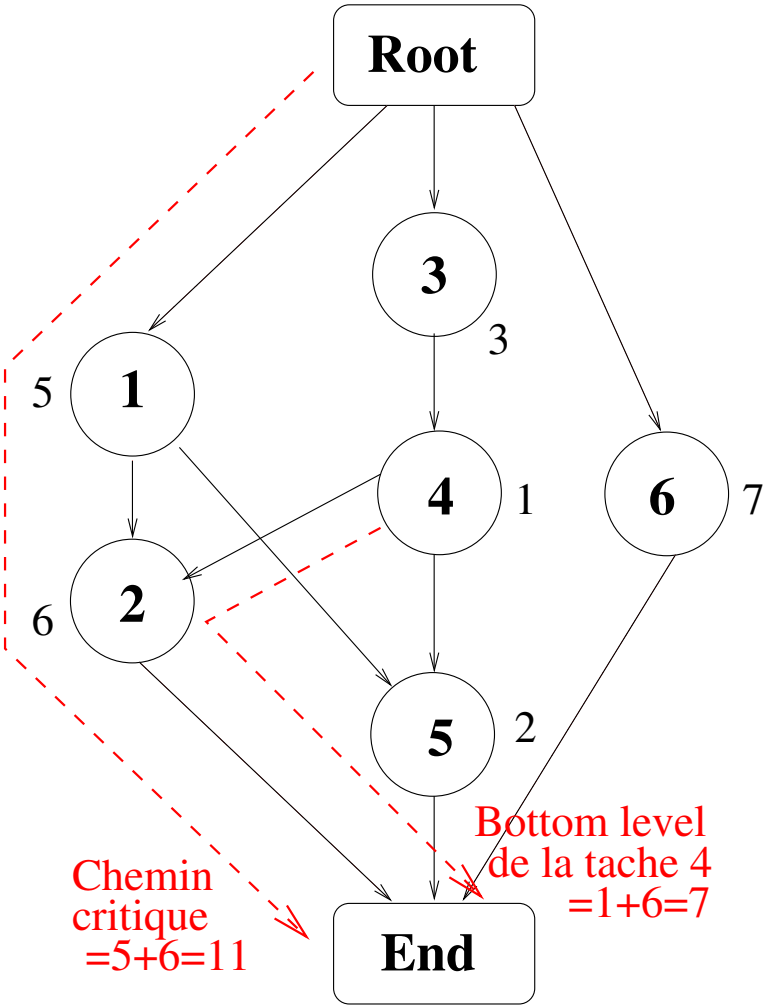
AlGorille / LORIA

Réunion MA0, 23 mars 2006

Introduction



Modèle d'applications



Modèle d'application

- Application parallèle

  $\longrightarrow$ graphe orienté acyclique $G = (V, E)$

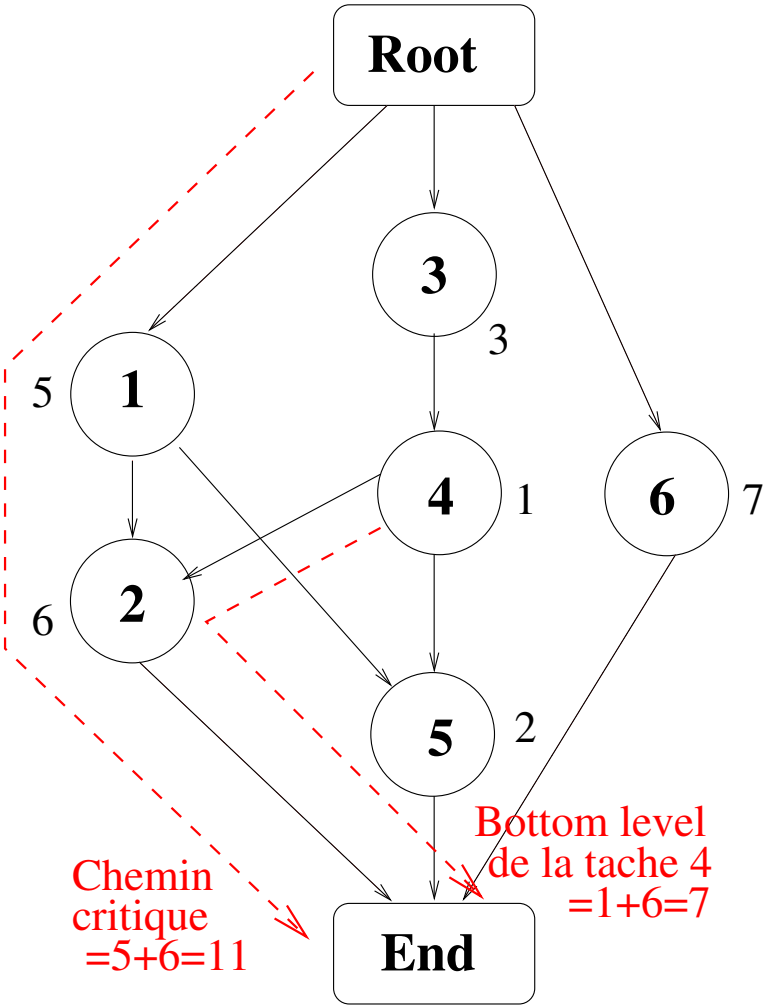
- On suppose que chaque tâche peut s'exécuter sur un ou plusieurs processeurs

  $\longrightarrow$ tâches parallèles

- Le temps d'exécution d'une tâche peut être modélisé par la loi d'Amdahl :

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

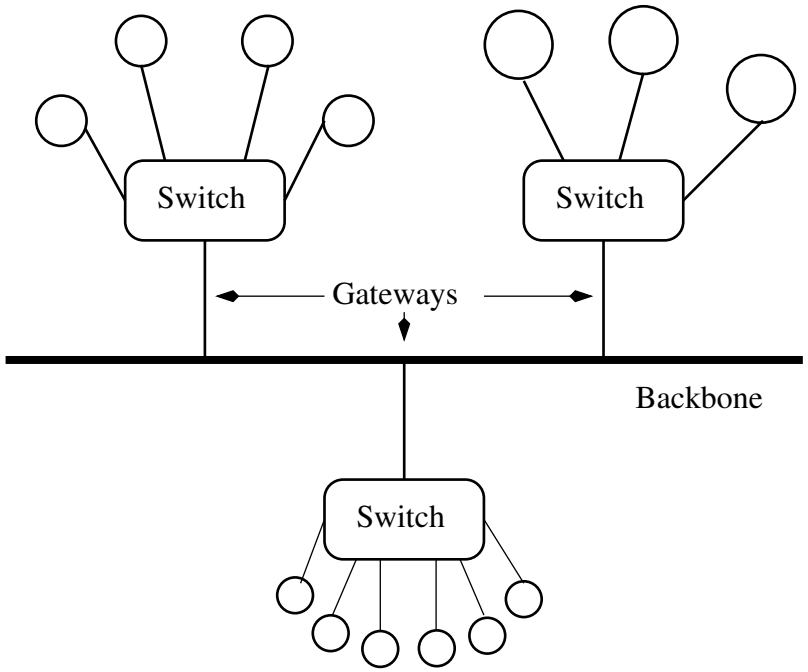
Modèle d'applications



Modèle d'application

- Application parallèle
  - $\longrightarrow$ graphe orienté acyclique $G = (V, E)$
- On suppose que chaque tâche peut s'exécuter sur un ou plusieurs processeurs
  - $\longrightarrow$ tâches parallèles

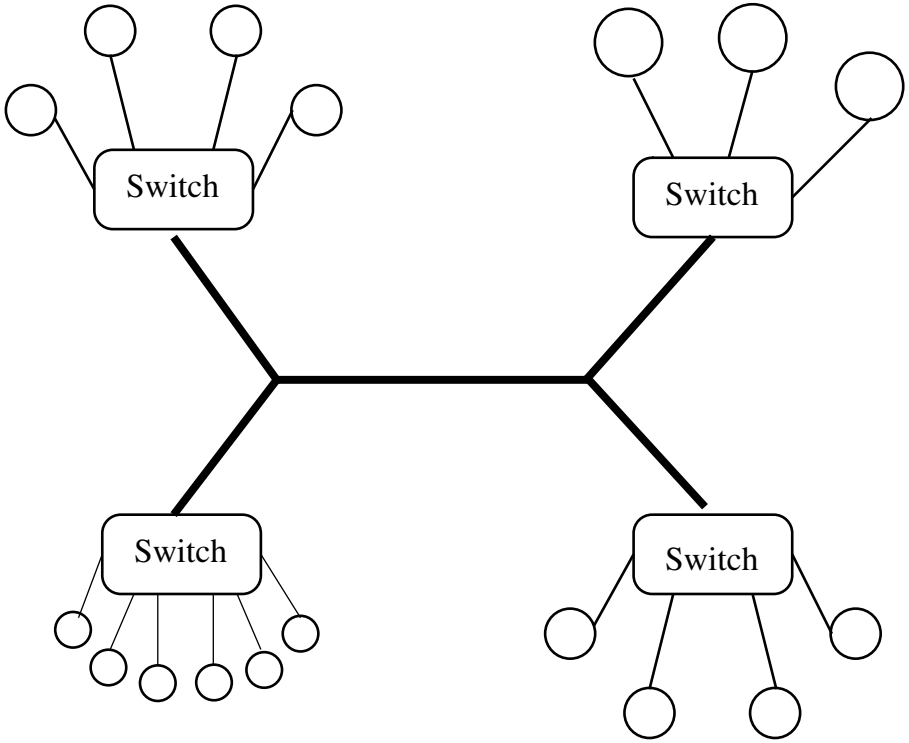
Modèle de plates-formes

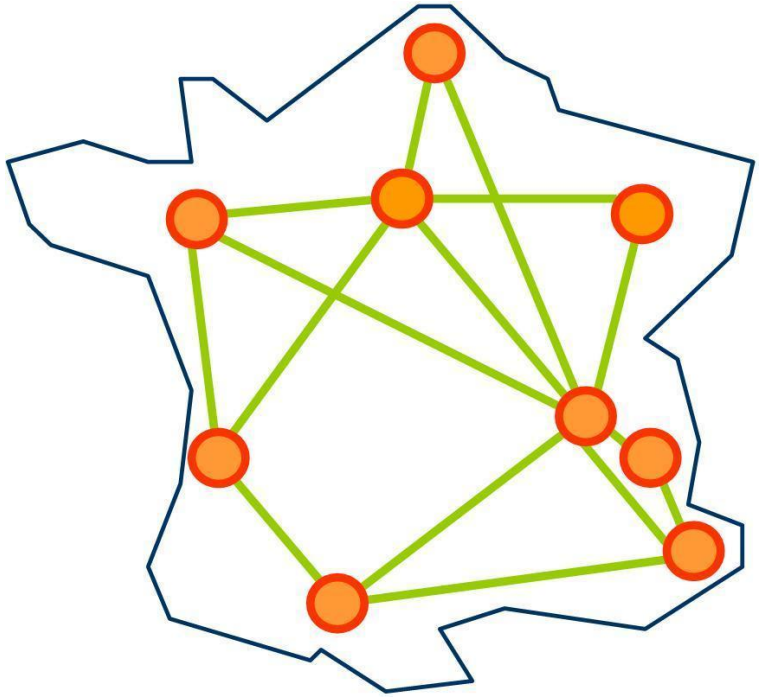


Grilles légères

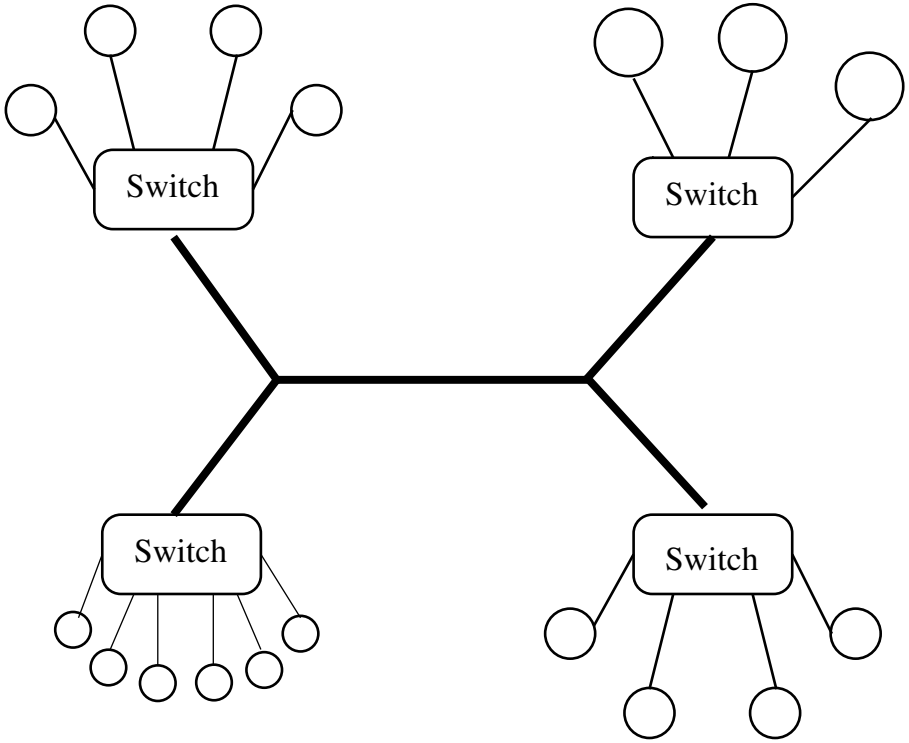
- Plusieurs grappes homogènes
- Hétérogénéité de l'ensemble
  - Réseaux d'interconnexion
  - Vitesse des processeurs
  - Capacité mémoire

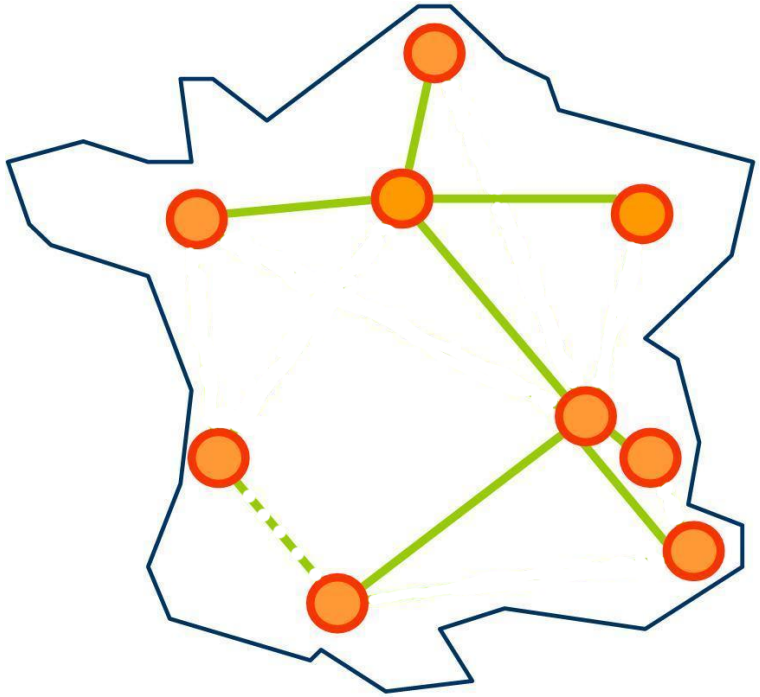
Nouveaux modèles de plates-formes





Nouveaux modèles de plates-formes





Problématique

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 :
  - Minimiser temps de complétion de l'application ?
  - Garder un bon compromis entre temps de complétion et puissance utilisée ?
  - Rester efficace ?

Pourquoi changer de modèle de plate-forme

- Pour l'instant
  - Réseau à plat → pas très réaliste
  - Placement statique → interférences entre tâches placées non prises en compte
- À l'avenir
  - Topologie complexe → influence des transferts sur le placement
  - Placement dynamique → le passé influence le choix

Hétérogénéité de Grid'5000

Pas si hétérogène que ça

- Matériel → beaucoup de bi-opterons (IBM/SUN)
- Matériel → tous de même génération
- Grappes → peu de différences de tailles
- Réseau → plus homogène avec Renater 4

→ Il est possible d'utiliser un sous-ensemble homogène de Grid 5000

  - De quelques centaines de processeurs
  - sur 4 à 5 sites

Rappel sur les heuristiques existantes

Séquentiel hétérogène + parallélisme

- Adaptation d'algorithmes d'ordonnancement de tâches séquentielles en milieu hétérogène
  - HEFT ⇒ M-HEFT [EuroPar'04]

Parallèle homogène + hétérogénéité

- Intégration de l'hétérogénéité dans des algorithmes d'ordonnancement de tâches parallèles en milieu homogène
  - CPA ⇒ HCPA et S-HCPA

Modifications à apporter

- Rendre dynamique la phase de placement
  - Objectif : prendre en compte les commmunications en cours
- Idée : tester les différents scénarios et garder le meilleur

Autres heuristiques

Il existe des algorithmes garantis avec précédences, mais sans communication

- sur un cluster
- sur plusieurs clusters identiques

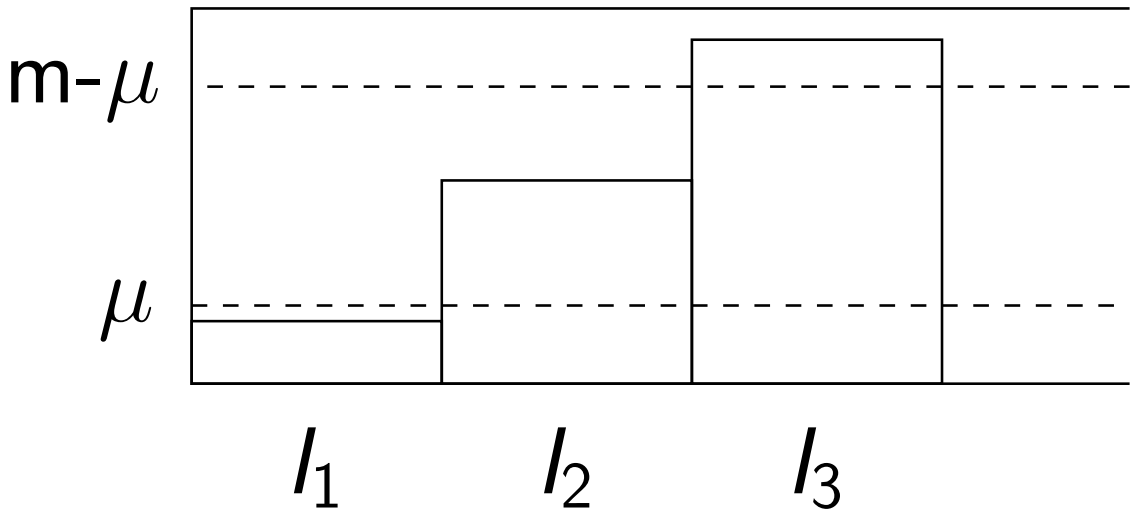
Autres heuristiques

Il existe des algorithmes garantis avec précédences, mais sans communication

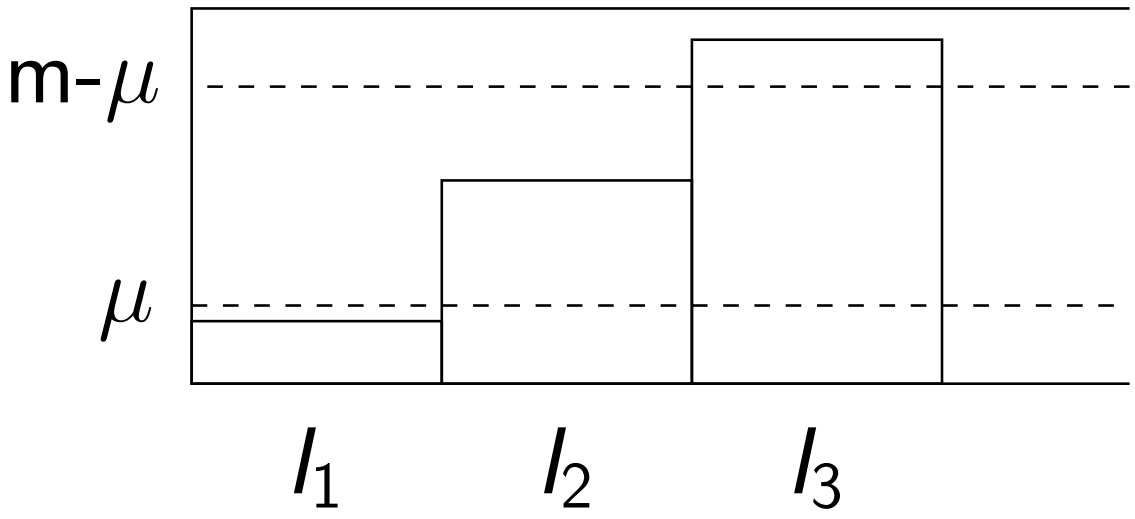
- sur un cluster
- sur plusieurs clusters identiques

Comment s'en inspirer ?

Ordonnancement garanti (2,62) [Lepere et al 01]



Ordonnancement garanti (2,62) [Lepere et al 01]



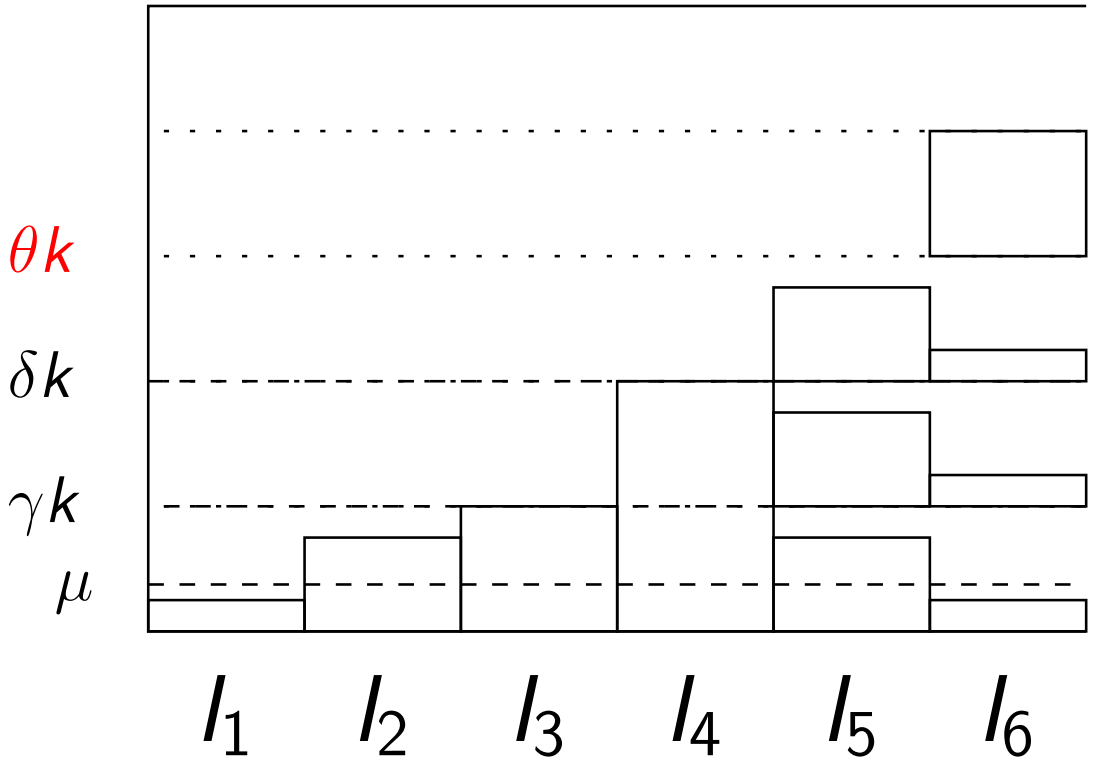
$$\omega = l_1 + l_2 + l_3 \quad (1)$$

$$\omega^* \geq L^*_{max} \geq l_1 + \frac{\mu}{m} l_2 \quad (2)$$

$$m\omega^* \geq W^* \geq l_1 + \mu l_2 + (m - \mu) l_3 \quad (3)$$

En hiérarchique homogène (au pire 5,64) [EuroPar'05]

Il y a 6 types d'intervalles :



Et sur Grid5000 ?

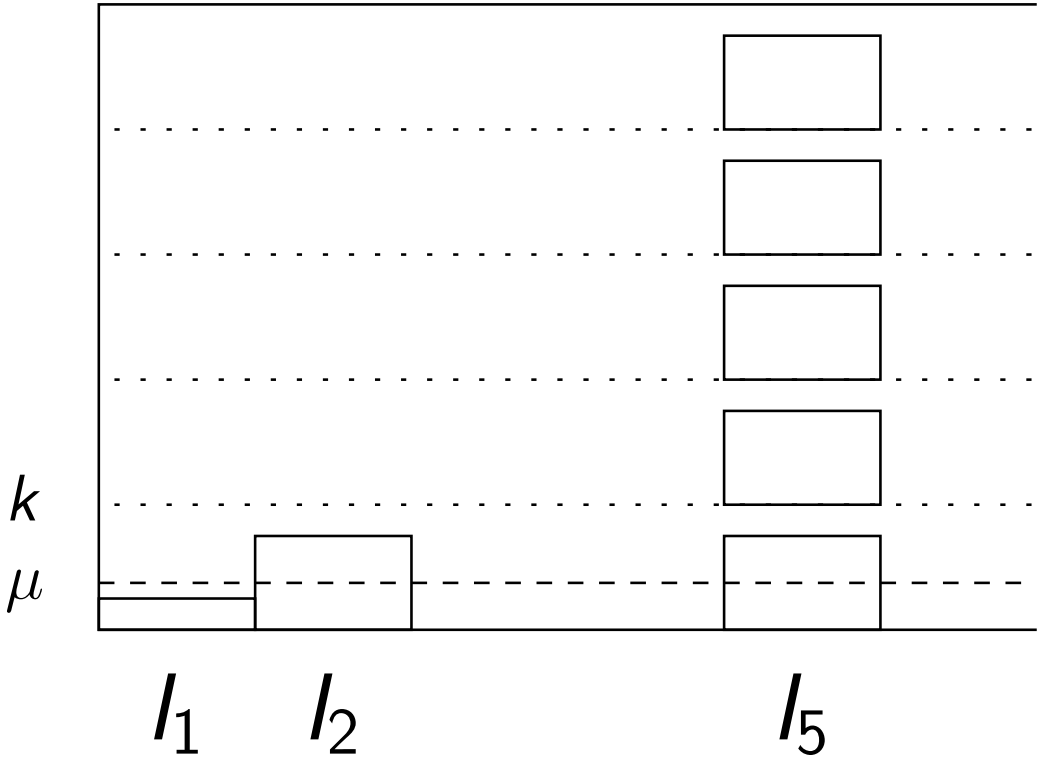
Trop de latence entre sites

Et sur Grid5000 ?

Trop de latence entre sites $\Rightarrow$ on supprime les «grandes» tâches

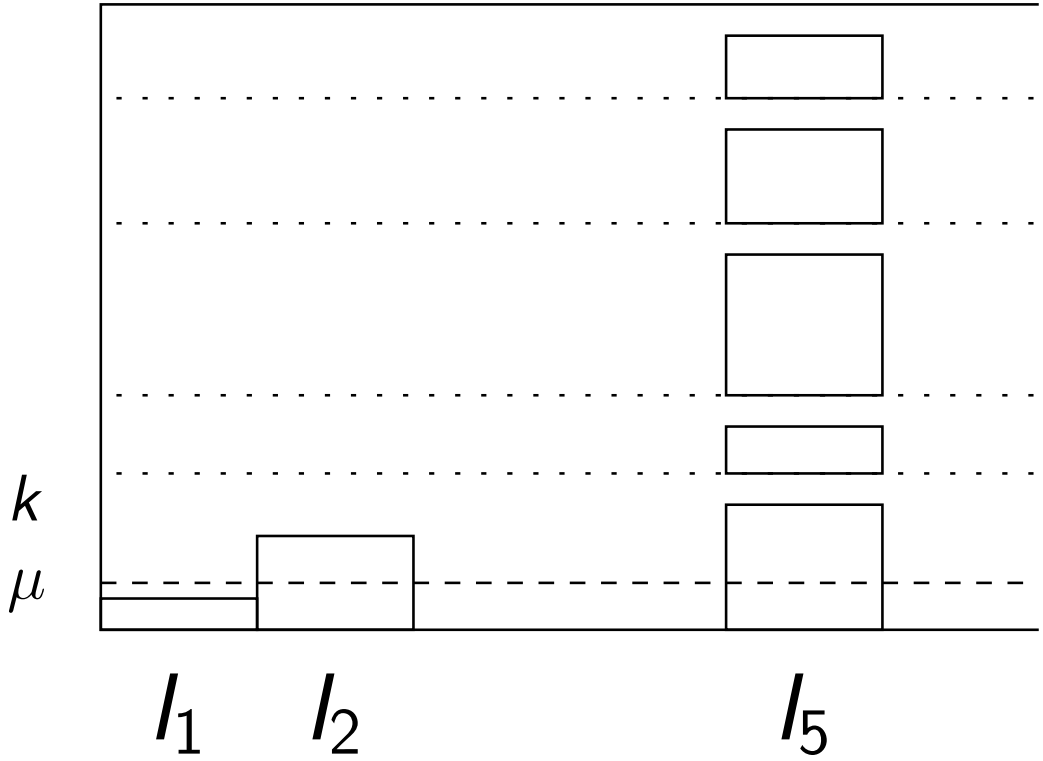
Et sur Grid5000 ?

Trop de latence entre sites ⇒ on supprime les «grandes» tâches



Et sur Grid5000 ?

Trop de latence entre sites ⇒ on supprime les «grandes» tâches



Fonctionne aussi avec des sites différents !

Ordonnancement hiérarchique

Quelques cas d'école

	Cas 1	Cas 2	Cas 3	Cas 4	Cas 5
Bordeaux	96	96		96	96
Lille	106				106
Lyon			112		112
Nancy	94	94		94	94
Rennes			128	128	128
Toulouse			116	116	116

p max	106	96	128	128	128
p tot	296	190	356	434	652
k	3	2	3	4	6

Ordonnancement hiérarchique

Quelques cas d'école

	Cas 1	Cas 2	Cas 3	Cas 4	Cas 5
p max	106	96	128	128	128
p tot	296	190	356	434	652
k	3	2	3	4	6
μ	57.76	48.17	69.45	70.53	77.35
garantie	3.35	2.98	3.36	3.78	4.36

Ordonnancement hiérarchique

1000 processeurs - 3 sites

	6*200	3*400	600+2*300	600+3*200
p max	200	400	600	600
p tot	1200	1200	1200	1200
k	6	3	3	4
μ	133.87	226.81	254.21	216.48 (*)
garantie	3.98	3.29	3.72	4.55 (*)

Toujours sans communications

Schéma de validation

Simulateur SimGrid

- Exécution de la simulation au fur et à mesure du placement
- A chaque nouvelle tâche à placer
  - Arrêter la simulation
  - Tester chacune des possibilités de placement
  - Garder la meilleure
  - Reprendre la simulation
- Problème : prédiction de performance idéale
  - Pratique mais irréaliste («tricherie»)

Schéma de validation

Simulateur SimGrid

- Exécution de la simulation au fur et à mesure du placement
- A chaque nouvelle tâche à placer
  - Arrêter la simulation
  - Tester chacune des possibilités de placement
  - Garder la meilleure
  - Reprendre la simulation
- Problème : prédiction de performance idéale
  - Pratique mais irréaliste («tricherie»)
- Solution : utiliser deux simulateurs distincts communiquants
  $\longrightarrow$ vue floue de l'ordonnanceur

Expérimentation

Devoirs pour Tchimou

- Modéliser un sous ensemble de Grid5000
- Comparer l'exécution SimGrid à l'exécution réelle
- Utiliser la simulation pour prendre les décisions d'ordonnancement
- Quelle est la meilleure heuristique ?
- Quels sont les meilleurs paramètres ?

Expérimentation

Devoirs pour Tchimou

- Modéliser un sous ensemble de Grid5000
- Comparer l'exécution SimGrid à l'exécution réelle
- Utiliser la simulation pour prendre les décisions d'ordonnancement
- Quelle est la meilleure heuristique ?
- Quels sont les meilleurs paramètres ?

Scrupules

Expérimentation

Devoirs pour Tchimou

- Modéliser un sous ensemble de Grid5000
- Comparer l'exécution SimGrid à l'exécution réelle
- Utiliser la simulation pour prendre les décisions d'ordonnancement
- Quelle est la meilleure heuristique ?
- Quels sont les meilleurs paramètres ?

Scrupules

On aimerait trouver une application réelle

- Ne pas gâcher des heures de CPU
- Ne pas modéliser les tâches