Changement de contexte pour tâches virtualisées à l'échelle des grappes

Fabien Hermenier Adrien Lèbre Jean-Marc Menaud

ASCOLA group, École des Mines de Nantes

Conférence Francophone en Système d'Exploitation,
9 septembre 2009, Toulouse

Motivations

Les grappes de serveurs

- Ensemble de machines interconnectées par un réseau performant
- Support pour l'exécution d'applications variées embarquées dans des tâches

Le gestionnaire de ressources

- Orchestre l'exécution des tâches
- Ordonnancement : Quelles tâches exécuter et où ?
- Réservation de ressources en accord avec la description de la tâche

Motivations

Traditionnellement :

- Allocation statique de ressources
- Exécution d'un trait
- sous-exploitation des ressources de la grappe

Cependant, des approches dynamiques existent :

- co-scheduling, gang scheduling, ...
- Requiert des mécanismes complexes et lourds pour manipuler les tâches dynamiquement : migration, suspension/reprise, ...
- Des mécanismes délicats à utiliser efficacement

Motivations

Une approche à base de machines virtuelles ?

- Chaque processus d'une tâche est embarqué dans une VM
- migration, suspension, reprise, ...
- Consolidation dynamique

Un support en vogue

Les approches actuellement manquent d'abstractions :

- des stratégies d'ordonnancement différentes
- les mêmes problèmes de manipulation des VMs

Proposition

Changement de contexte dans les grappes à base de VMs :

- Généralisation des concepts de manipulation des VMs avec **Entropy**[CFSE'06,VEE'09]
- Le développeur de l'ordonnanceur se focalise uniquement sur la sélection des tâches à exécuter
- Entropy s'occupe du reste :
  - Détection des actions à exécuter
  - Planification des actions assurant leur faisabilité
  - Recherche du plan le plus rapide possible

Le changement de contexte

... dans les ordinateurs

- Suspend une tâche utilisant le CPU et le ré-alloue à une autre tâche
- Une tâche suspendue peut être ré-affectée à un autre CPU (machine SMP)

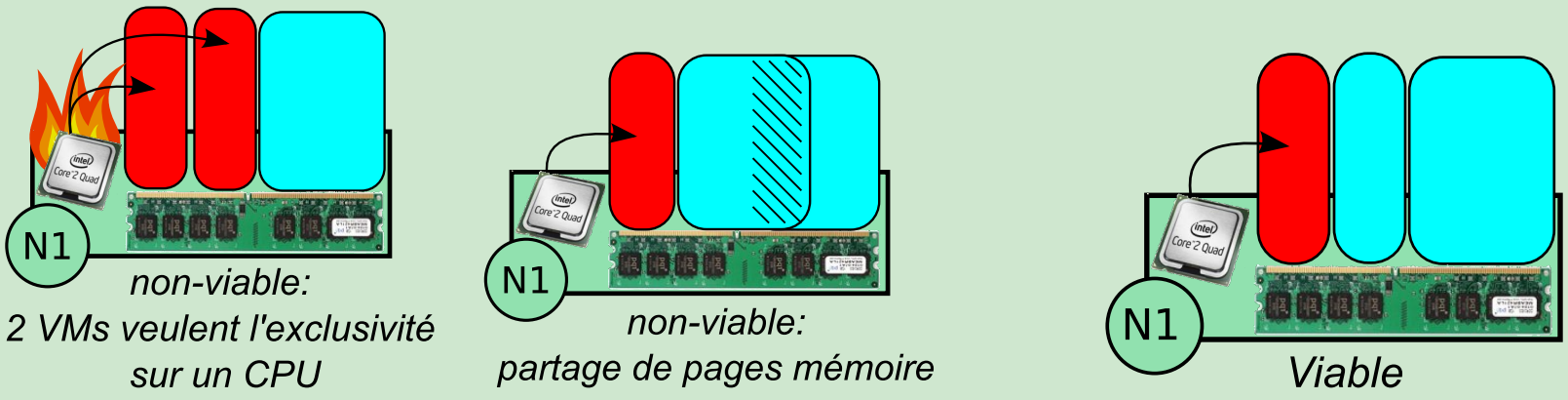
... dans les grappes ?

- Une tâche peut impliquer plusieurs VMs
- Migration des VMs d'une tâche entre différents nœuds
- Suspension et reprise d'exécution d'une tâche
- Un air de déjà vu avec le partitionnement temporel

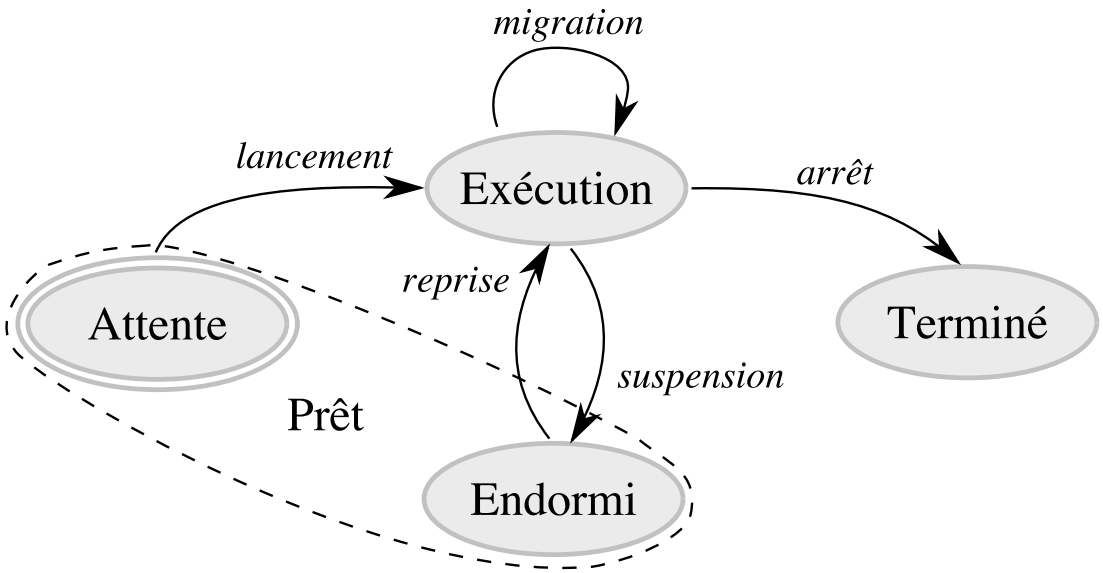
Une configuration

- Chaque VM en cours d'exécution est hébergée sur un nœud
- Chaque VM en cours d'exécution requiert une quantité fixe de mémoire, une quantité variable de ressources CPU
- Une configuration est viable ssi tous les besoins CPU et mémoire des VMs sont satisfaits.

Example

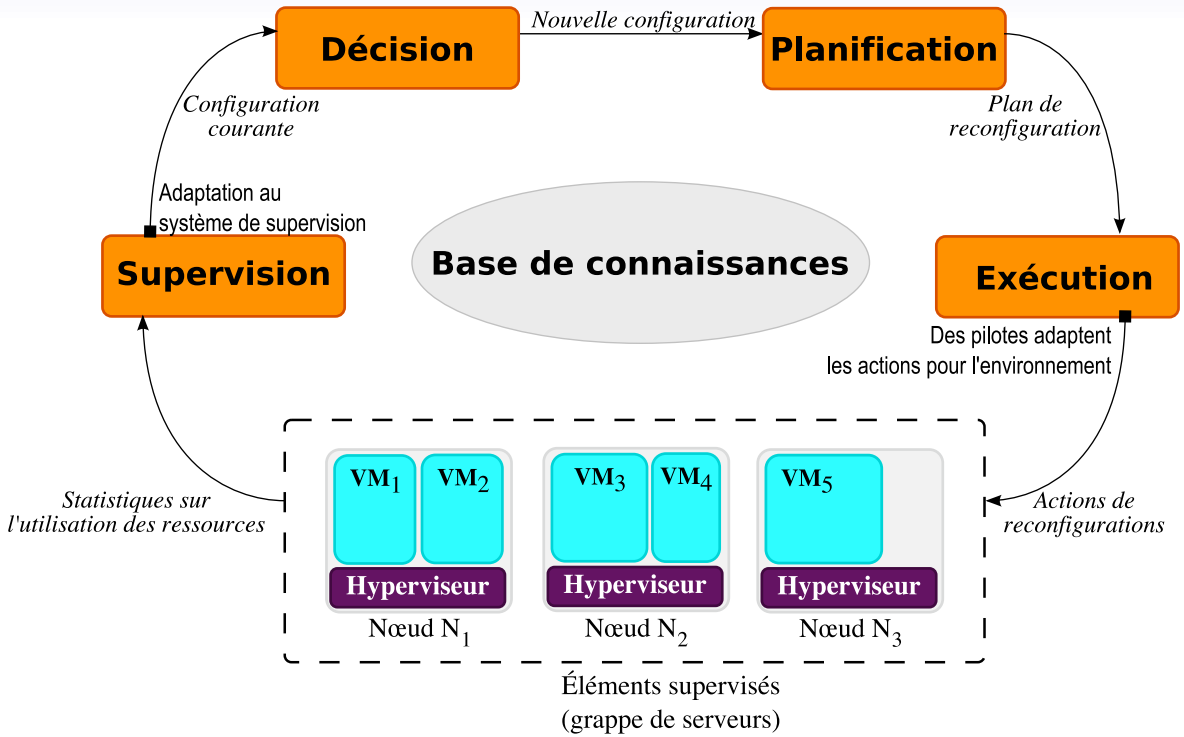


Cycle de vie d'une machine virtuelle



Les VMs appartenant à une même tâche sont nécessairement dans le même état

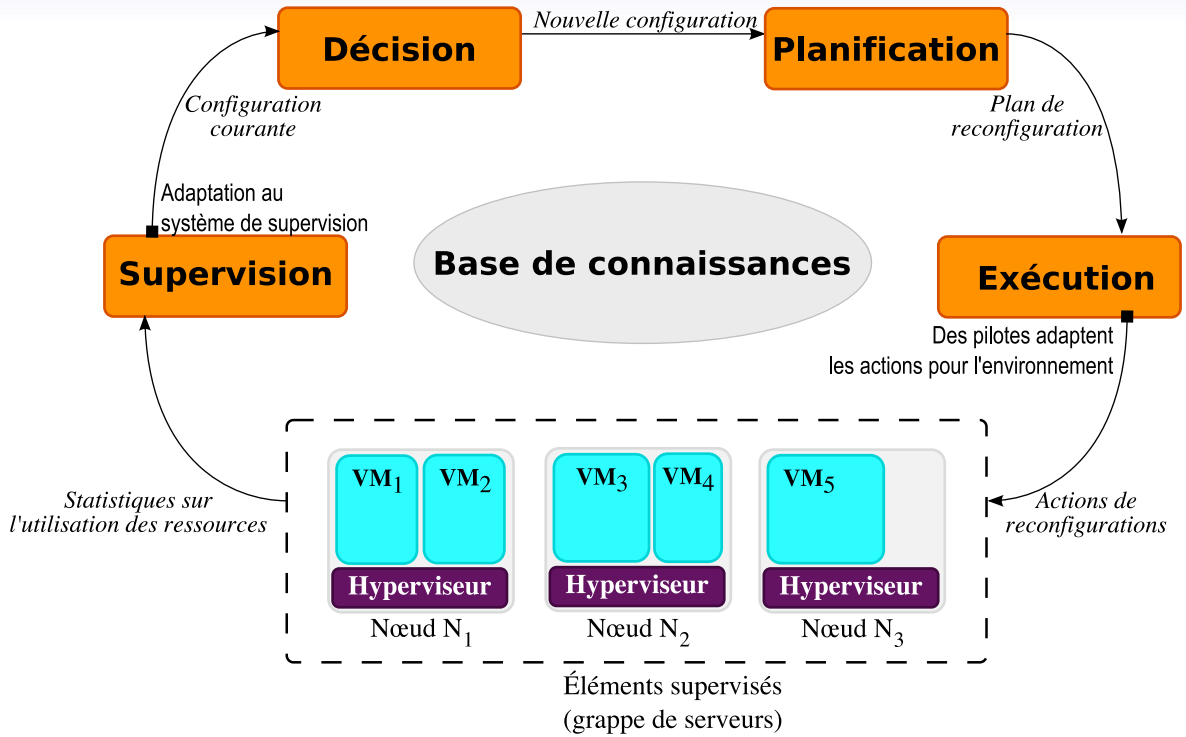
Boucle de contrôle d'Entropy



Supervision

- Utilisation d'un système de supervision traditionnel (par ex. Ganglia)
- Extrait la configuration courante

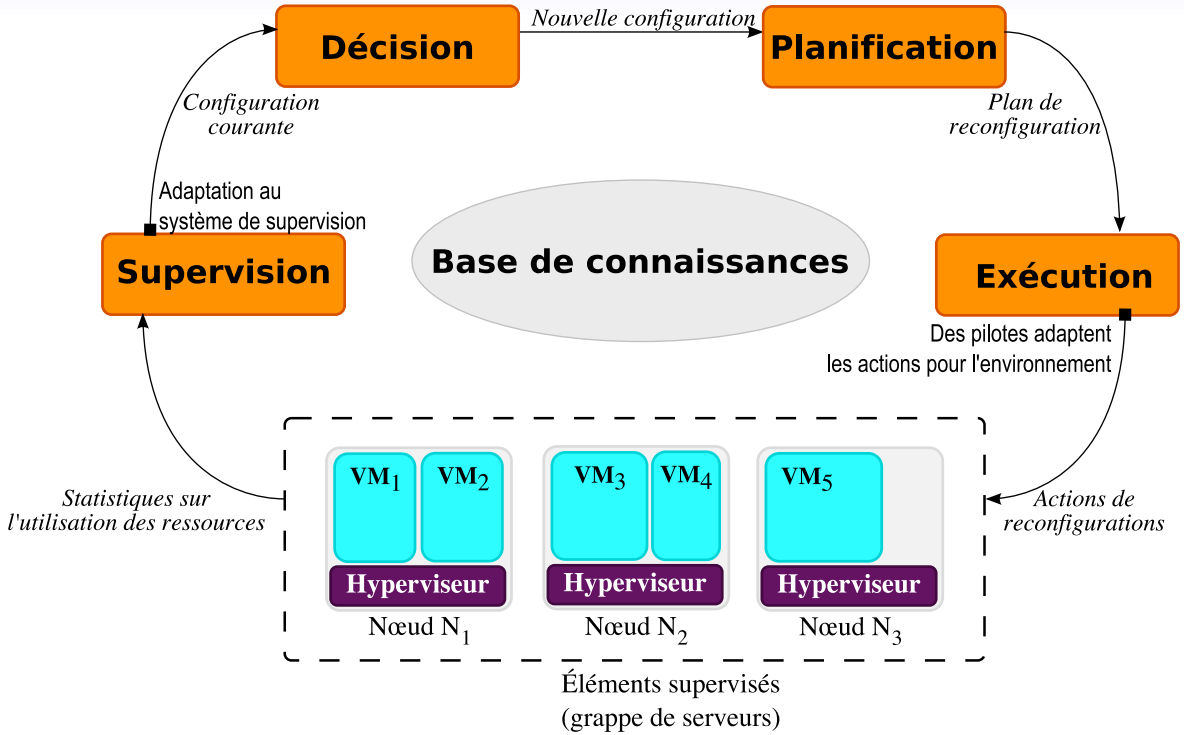
Boucle de contrôle d'Entropy



Décision

- À la charge du développeur
- Calcul une configuration viable indiquant les tâches à exécuter.

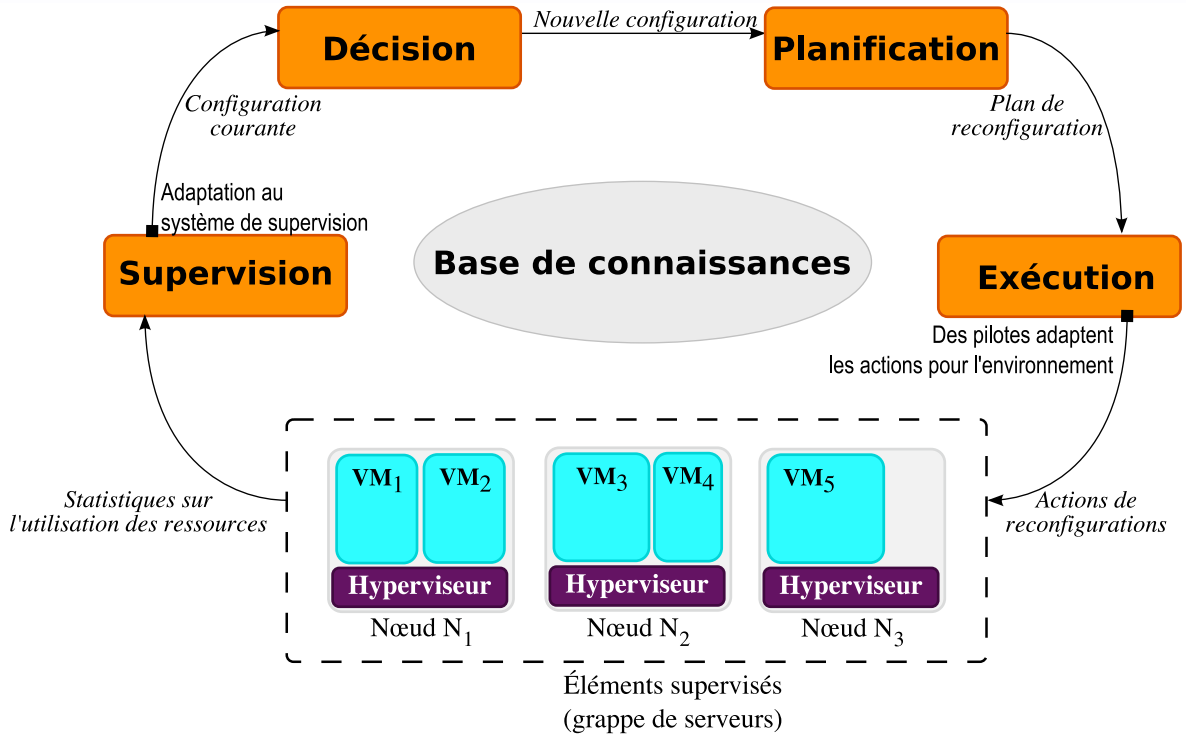
Boucle de contrôle d'Entropy



Planification du changement de contexte

- Calcul la transition entre l'état courant et la nouvelle configuration
- Réduit le temps d'exécution du plan à un minimum

Boucle de contrôle d'Entropy



Exécution

- Exécute les actions sur la grappe au travers de pilotes (actuellement pilotes SSH et Xen)

Module de décision

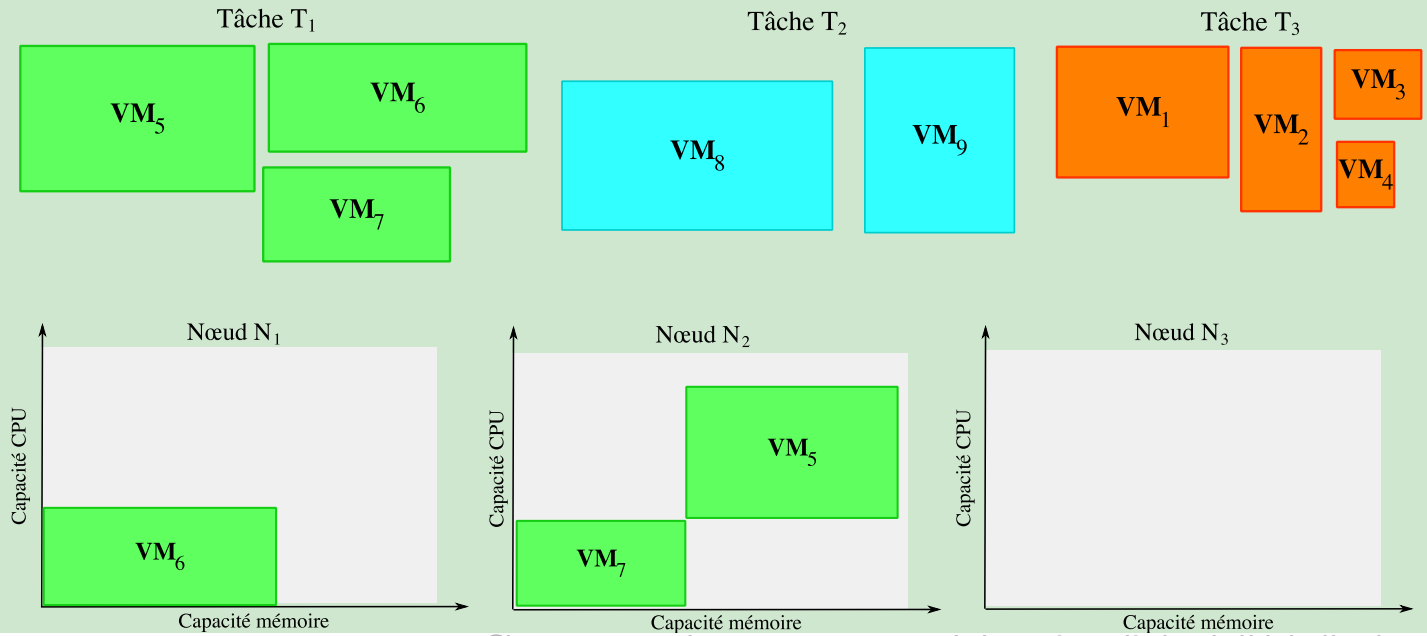
- Le développeur implémente sa propre stratégie d'ordonnancement
- Indique avec une configuration viable exemple, la liste des tâches qu'il souhaite exécuter
- Utilise la configuration courante de la grappe et des algorithmes spécifiques au besoin (FCFS, gang-scheduling, ...)

Exemple d'algorithme d'ordonnancement

Principe

- Allocation des ressources CPU à la volée, au besoin
- Ordre de file non-strict type « First Fit » avec priorité
- Partitionnement spatial et temporel

Example

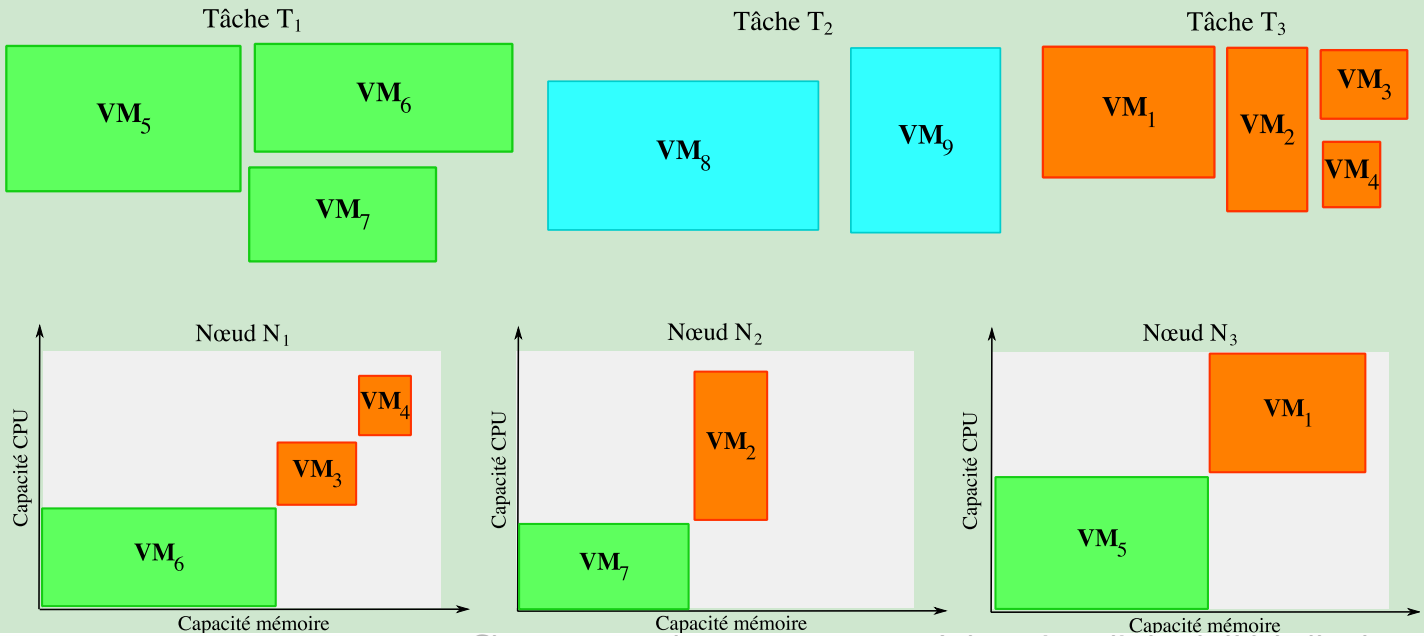


Exemple d'algorithme d'ordonnancement

Principe

- Allocation des ressources CPU à la volée, au besoin
- Ordre de file non-strict type « First Fit » avec priorité
- Partitionnement spatial et temporel

Example



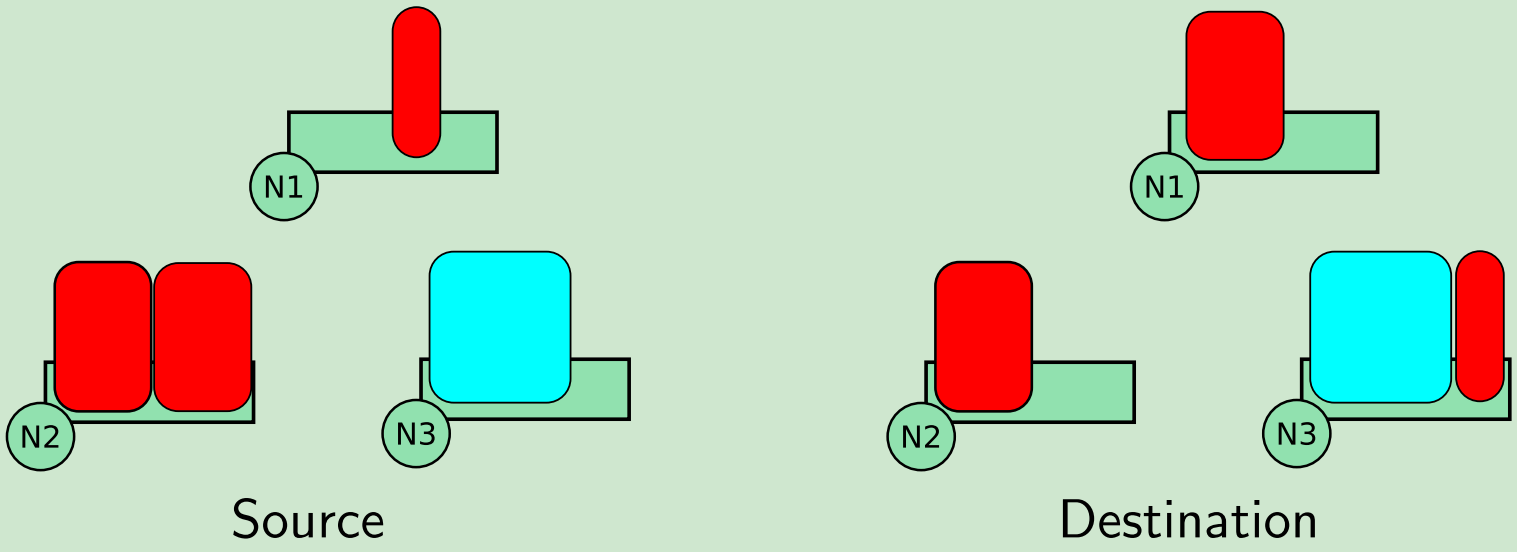
Planification

- Détection des actions à exécuter pour assurer le changement de contexte
- Des actions libèrent des ressources :
  suspension, arrêt, migration
- Des actions consomment des ressources :
  lancement, reprise, migration
- La portée d'une action est limitée à la VM concernée :
  implication sur le maintien de l'état d'une application distribuée

Planification

Des dépendances purement séquentielles

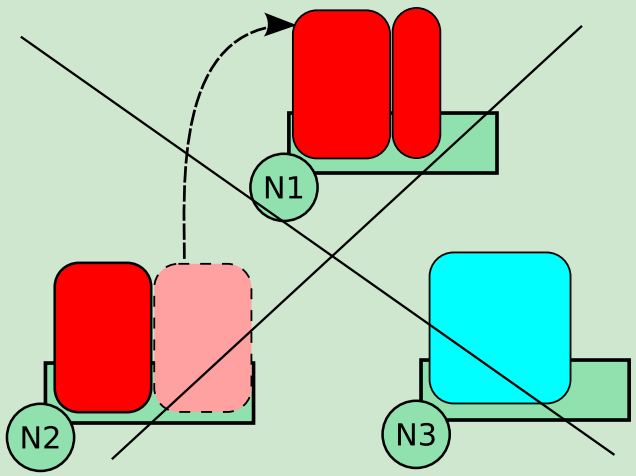
Example



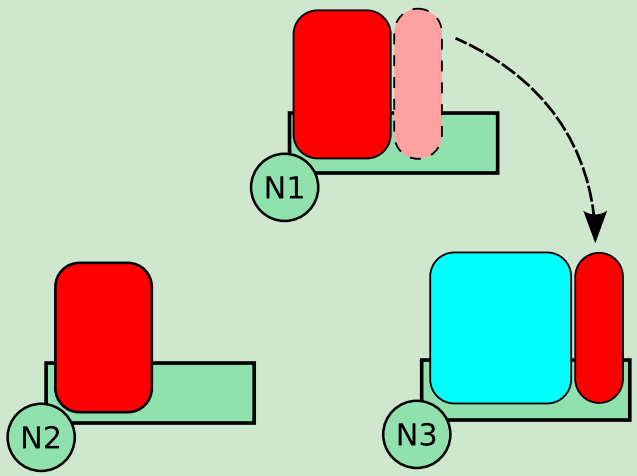
Planification

Des dépendances purement séquentielles

Example



(1) non-faisable

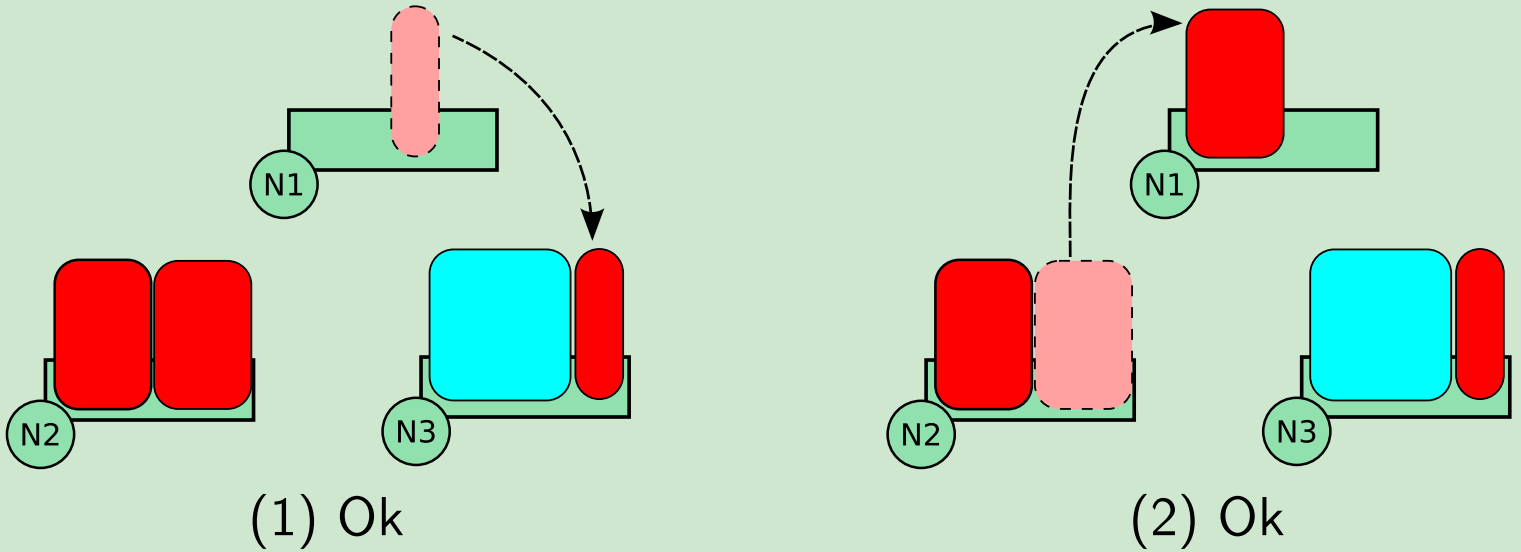


(2)

Planification

Des dépendances purement séquentielles

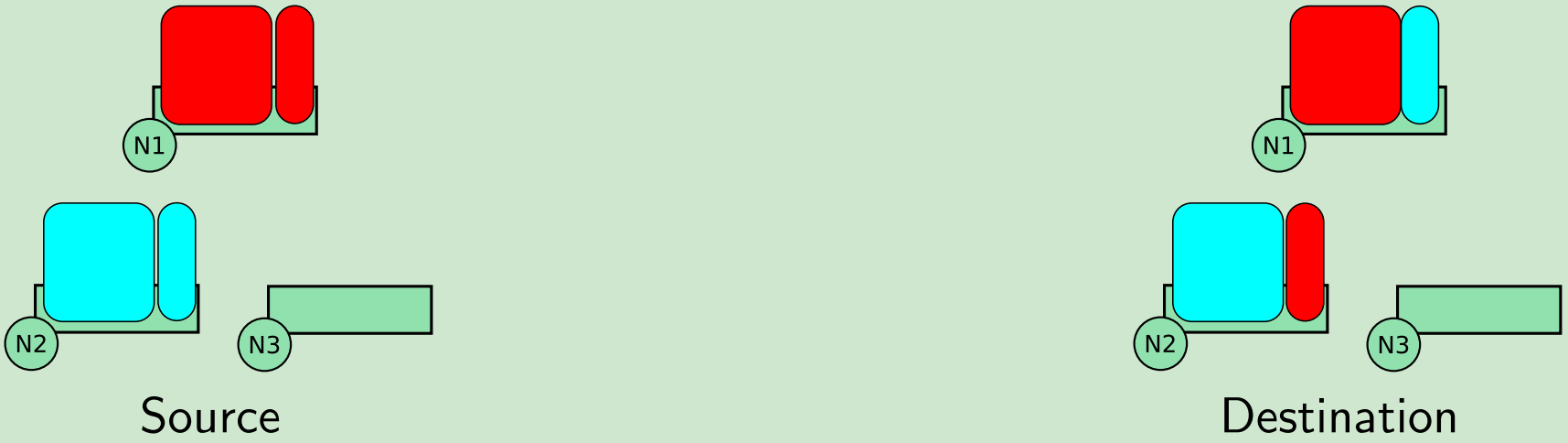
Example



Planification

Des dépendances cycliques :

Example



Une situation plus commune :

$a = 2, b = 3;$

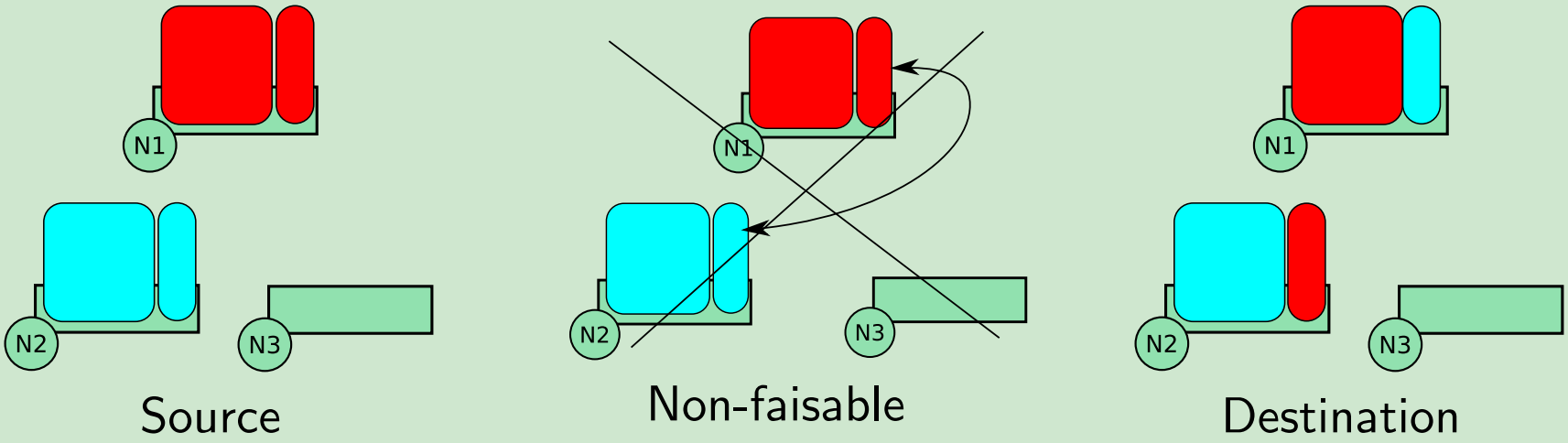
...

$a = 3, b = 2;$

Planification

Des dépendances cycliques :

Example



Une situation plus commune :

$a = 2, b = 3;$

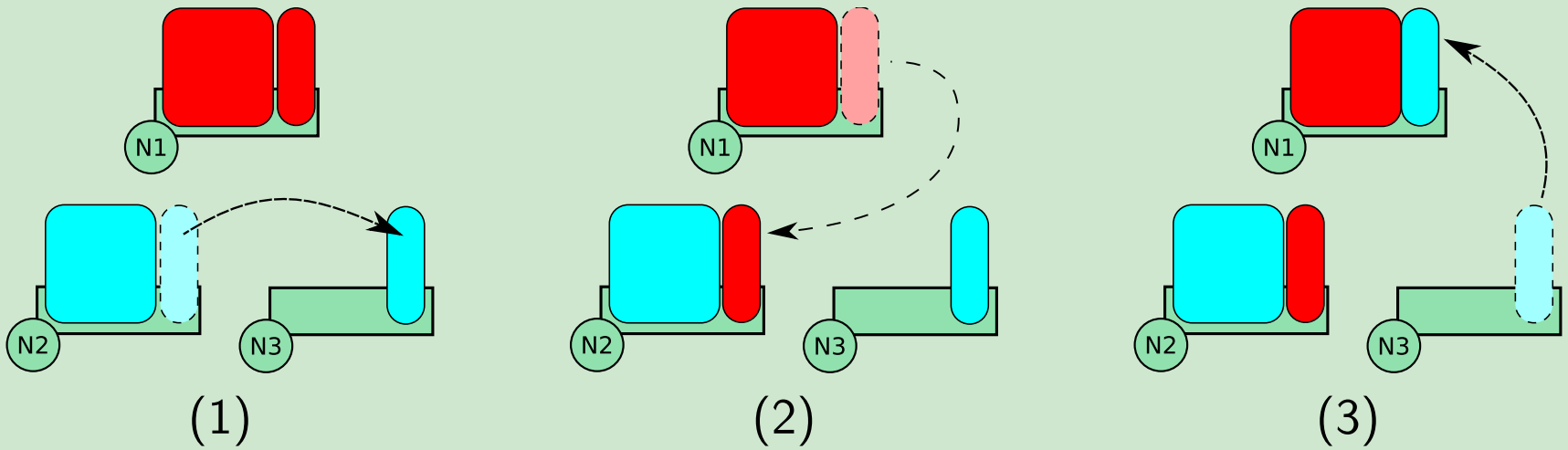
$a = b \ \& b = a?$

$a = 3, b = 2;$

Planification

Des dépendances cycliques :

Example



Une situation plus commune :

$a = 2, b = 3;$

$c = a; a = b; b = c$

$a = 3, b = 2;$

Planification

Le plan de reconfiguration

- Ordonne les actions pour assurer leur faisabilité
- Exécute les actions dès que possible
- Parallélise au mieux l'exécution des actions
- Maintiens la cohérence des états de VMs appartenant à une même tâche (post-traitement)

Optimisation du changement de contexte

Coût d'une action ?

- Exécuter une action à un coût (temps, performances)
- Une action est exécutée dans un contexte précis

Coût d'un changement de contexte ?

Évalué par une fonction de coût qui favorise :

- une exécution des actions au plus tôt
- un nombre d'actions réduit
- les actions qui s'exécutent rapidement

Optimisation du changement de contexte

Principe

Calcul d'une configuration viable :

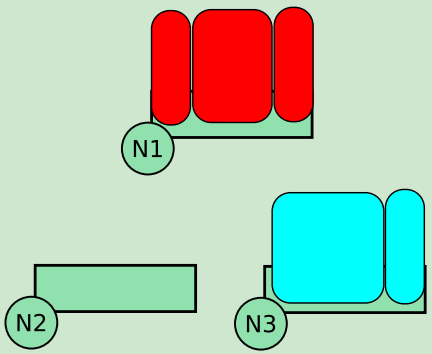
- qui respecte les états des tâches souhaités par l'algorithme d'ordonnancement
- impliquant un coût de changement de contexte réduit

Approche : la programmation par contraintes

- Méthode de définition et de résolution de problèmes combinatoires (ordonnancement, placement, ...)
- Processus de résolution générique et flexible
- Le temps de calcul d'une solution est important, contre-balancé par sa qualité [VEE'09]

Optimisation du changement de contexte

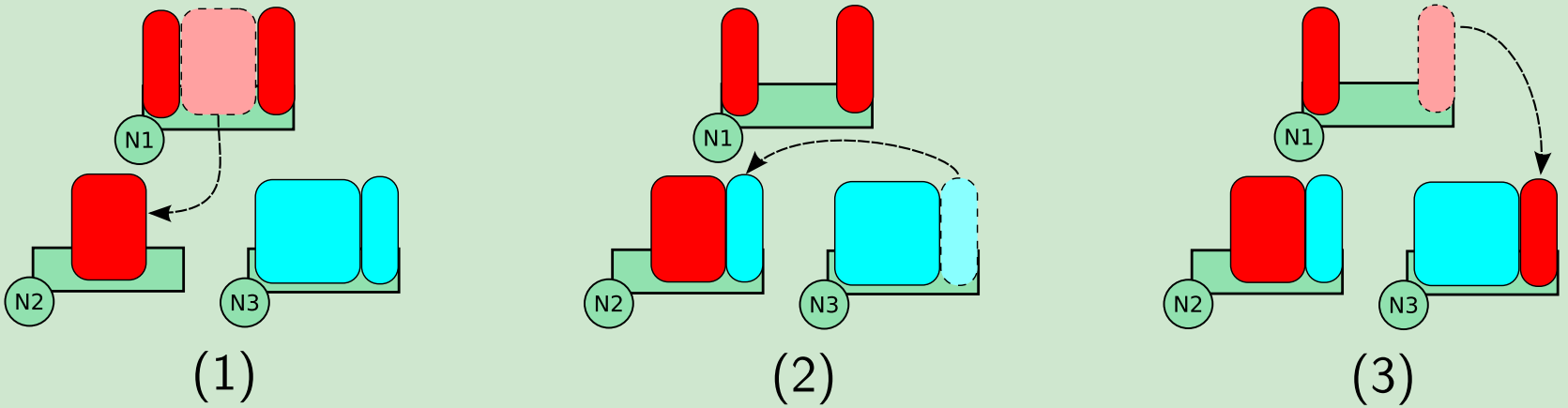
Example



(1)

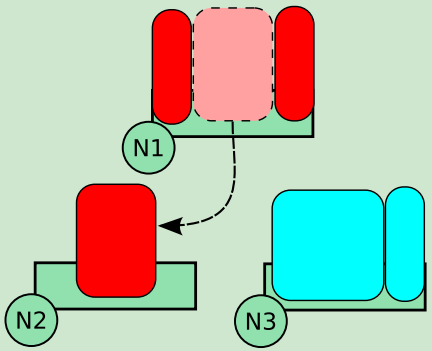
Optimisation du changement de contexte

Example

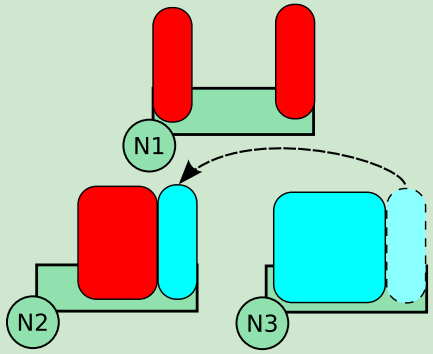


Optimisation du changement de contexte

Example

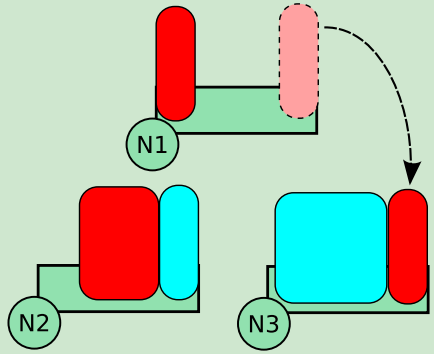


(1)

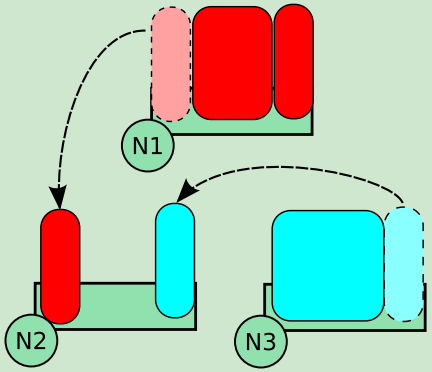


(2)

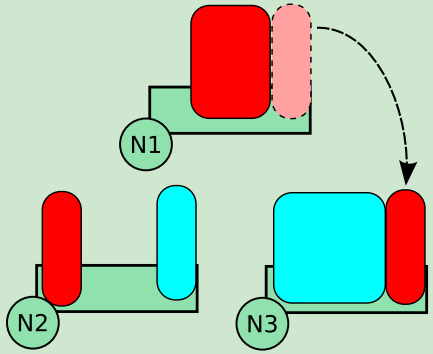
$cout = 9$



(3)



(1)



(2)

$cout = 4$

Expérimentation sur grappe

La grappe « Pastel »

- 11 nœuds de calcul
- 3 nœuds de stockage distribuent les images des VMs
- 1 nœud de service exécute Entropy

Expérimentation sur grappe

La grappe « Pastel »

- 11 nœuds de calcul
- 3 nœuds de stockage distribuent les images des VMs
- 1 nœud de service exécute Entropy

Méthodologie

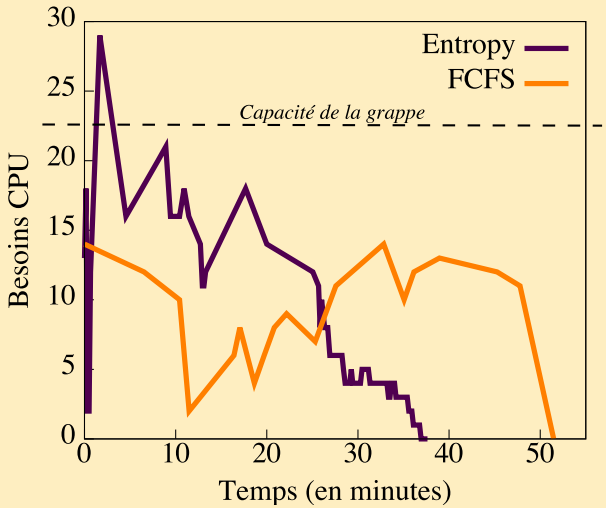
- Une file d'attente de 8 tâches (8 applications NASGrid)
- Chaque tâche utilise 9 VMs
- Comparaison de l'exécution des tâches avec 2 algorithmes d'ordonnancement :
  - FCFS : Premier arrivé, premier servi
  - Entropy + notre algorithme

Expérimentation sur grappe

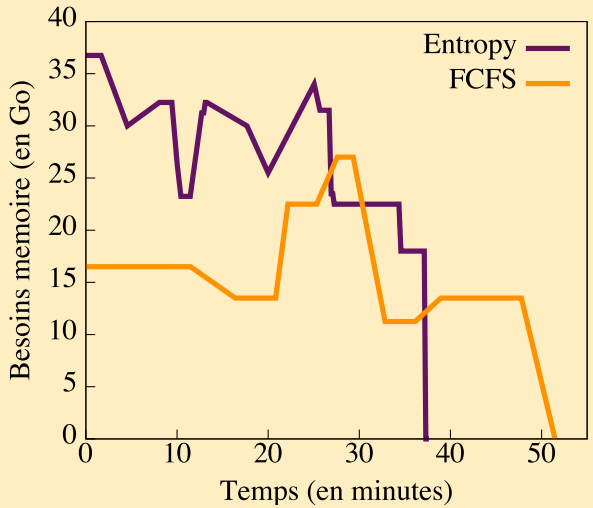
Bénéfices

- Meilleure utilisation des ressources CPU et mémoire
- L'auto-adaptation permet de solutionner les cas de surcharges CPU
- Réduit le temps total d'exécution des tâches

Taux d'utilisation des ressources



(e) CPU



(f) Mémoire

Expérimentation sur grappe

Bénéfices

- Meilleure utilisation des ressources CPU et mémoire
- L'auto-adaptation permet de solutionner les cas de surcharges CPU
- Réduit le temps total d'exécution des tâches

Temps cumulé d'exécution

- FCFS : 250 minutes
- Entropy : 150 minutes

Conclusion

- Des gestionnaires de ressources à base de VMs proposent une gestion dynamique de celles-ci pour améliorer le taux d'utilisation des grappes
- Des stratégies d'ordonnancement différentes mais des besoins identiques pour la manipulation des VMs

L'ordonnancement avec Entropy

- Permet d'implémenter des algorithmes d'ordonnancement avancés
- Le développeur ne se soucie pas du changement de contexte
- Preuve de concept avec l'implémentation d'un premier algorithme

Questions ?

http ://entropy.gforge.inria.fr

- statut expérimental pour cette version (1.2)
- Sources et binaires disponibles en LGPL
- Mailing-Lists, IRC : #entropy@FreeNode, ...