

Modélisation d'un Workflow par un programme mathématique

November 16, 2012

1 Introduction

Avantages :

- Solution optimale
- Algorithme de résolution courant
- Réoptimisation et modification du modèle “simples”

2 To Do

Les contraintes pour tout t à améliorer, chercher un schéma d'exploration .

3 Modélisation

3.1 Données

On dispose des données suivantes :

- I : l'ensemble des tâches i à exécuter, $i \in [1..n]$
- J : l'ensemble des machines j dans le cloud, $j \in [1..m]$
- T : l'ensemble des instants du problème, $t \in [0..t_{max}]$
- $d_{i,j}$: durée de la tâche i si nous l'exécutons sur la machine j
- $D_{i,j}$: durée du transfert de l'output de la tâche i vers la machine j
- $\Gamma(i)$: l'ensemble des successeurs de la tâche i

Remarque :

t_{max} est une majoration du temps d'exécution du problème. Il peut être obtenue de différentes manières :

- somme du temps d'exécution des tâches
- solution d'un algo polynomial

3.2 Variables

Le coeur de notre modélisation se situe dans les variables r_{ijk} définies

$$r_{i,j,k} = \begin{cases} 1 & \text{si la tâche } i \text{ est exécutée par la machine } j \text{ à l'instant } k \\ 0 & \text{sinon} \end{cases}$$

$$\gamma_{i,j} = \begin{cases} 1 & \text{si } \sum_k r_{ijk} \geq 1 \\ 0 & \text{sinon} \end{cases}$$

$$\gamma'_{i,j} = \begin{cases} 1 & \text{si } \sum_k r_{ijk} = d_{ij} \\ 0 & \text{sinon} \end{cases}$$

On introduit également :

x_i : date de début de la tâche i

3.3 Modèle Linéaire

Fonction Obj :

$\min u$

Sous contraintes :

Modéliser le max : on minimise le maximum des dates de fin :

$$u \geq x_{ij} + \sum_{t=0}^{t_{max}} r_{ijt}, \quad \forall (i, j) \in I \times J \quad (1)$$

Contraintes de précédences :

$$x_{ij} + \sum_{t=0}^{t_{max}} r_{ijt} + D_{ijk} \leq x_{ok} \quad \forall o \in \Gamma(i), \quad \forall (i, j, k) \in I \times J \times J \quad (2)$$

Contrainte : une tâche doit être exécutée

$$\sum_{j=0}^m \sum_{t=0}^{t_{max}} r_{ijt} \geq 1 \quad \forall (i, j) \in I \times J \quad (3)$$

Contrainte : une machine n'exécute qu'une tâche à la fois

$$\sum_{i=0}^n \sum_{t=0}^{t_{max}} r_{ijt} \leq 1 \quad \forall (i, t) \in I \times T \quad (4)$$

Pour (i, j) , si il existe un t tel que $r_{i,j,t}$ soit supérieur à 1, alors la somme des r_{ijt} vaut d_{ij} (le temps d'exécution d'une tâche sur une machine est d_{ij})

$$\sum_k r_{ijk} \geq 1 \Rightarrow \sum_k r_{ijk} = d_{ij}, \quad \forall (i, j) \in I \times J$$

Ce qui se traduit donc :

$$\gamma_{i,j} \leq \gamma'_{i,j} \quad (5)$$

Contraignons les variables γ_{ij} et γ_i , comme définit précédemment :

($\gamma_{ij} = 1$ si la somme des r_{ijt} est supérieur à 1, et $\gamma'_{ij} = 1$ si la somme des r_{ijt} vaut d_{ij})

$$\gamma_{i,j} \leq \sum_k r_{ijk} \leq M \cdot \gamma_{i,j}, \quad \forall (i,j) \in I \times J \quad (6)$$

$$d_{ij} \gamma'_{i,j} = \sum_k r_{ijk} \quad \forall (i,j) \in I \times J \quad (7)$$

(5''') Consécutivité d'exécution d'une tâche (non préemption)

4 Archives

On introduit également des variables auxiliaires :

- x_i : date de début de la tâche i
- $y_{i,j} = \begin{cases} 1 & \text{si la tâche } i \text{ est exécutée sur la machine } j \\ 0 & \text{sinon} \end{cases}$
- d_i : durée de la tâche i fixée

(1)(obsolète) d_i = durée de la tâche i

$$d_i = \sum_{j=0}^m y_{i,j} * d_{i,j}, \quad \forall i$$

(2)(obsolète) il existe un unique (i,j) tel que : $(r_{ijt}) = 1 \Rightarrow (y_{ij} = 1)$

$$y_{ij} - \frac{1}{duree} * \sum_{t=0}^{t_{max}} r_{ijt} = 0, \quad \forall i, j$$

(3)(obsolète) Une tâche est exécutée sur une seule machine

$$\sum_{j=0}^m y_{ij} = 1, \quad \forall i$$

Une tâche est exécutée une fois

$$\sum_{j=0}^m \sum_{k=0}^{t_{max}} r_{ijk} = d_i, \quad \forall i \quad (8)$$

Pour (i,j), on doit avoir $d_{i,j}$ variables $r_{i,j,k}$ supérieures à 1 (le temps d'exécution d'une tâche sur une machine est dij)

$$\sum_k r_{i,j,k} \geq 1 \Rightarrow \sum_k r_{i,j,k} = d_{i,j}, \forall (i,j) \in J \times M, \forall i,j$$

Avec les variables $\gamma_{i,j} = 1$ si la somme des r_{ijt} est supérieur à 1, et $\gamma'_{i,j} = 1$ si la somme des r_{ijt} vaut d_{ij}

Traductions :

- $\gamma_{i,j} = 1 \Leftrightarrow \sum_k r_{i,j,k} \geq 1$ se traduit par $\gamma_{i,j} \leq \sum_k r_{i,j,k} \leq M \cdot \gamma_{i,j}$
- $\gamma'_{i,j} = 1 \Leftrightarrow \sum_k r_{i,j,k} = d_{ij}$ se traduit par $d_{i,j} \gamma'_{i,j} = \sum_k r_{i,j,k}$
- la contrainte finale $\gamma_{i,j} = 1 \Rightarrow \gamma'_{i,j} = 1$ se traduit par $\gamma_{i,j} \leq \gamma'_{i,j}$