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 exectuer, 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'executons 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 tache i

Remarque :

t_{max} est une majoration du temps d'execution du problème. Il peut être obtenue de différentes manieres :
- somme du temps d'execution des taches
- solution d'un algo polynomial

3.2 Variables

Le coeur de notre modélisation se situe dans les variables $r_{i,j,k}$ définies

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

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

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

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 :

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

Contraintes de précédences :

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

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

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

Ce qui se traduit donc :

$$\gamma_{i,j} \leq \gamma'_{i,j} \tag{3}$$

Contraingnons les variables $\gamma_{i,j}$ et $\gamma_{i,j}$ comme définit précédement :
($\gamma_{i,j} = 1$ si la somme des r_{ijt} est supérieur à 1, et $\gamma'_{i,j}$ =1 si la somme des r_{ijt} vaut d_{ij})

$$\gamma_{i,j} \leq \sum_k r_{i,j,k} \leq M.\gamma_{i,j},\ \forall (i,j) \in I \times J \tag{4}$$

$$d_{i,j}\gamma'_{i,j} = \sum_k r_{i,j,k}\ \ \forall (i,j) \in I \times J \tag{5}$$

(5"') Consécutivité d'execution d'une tâche (non préemption)

4 Archives

On introduit également des variables auxiliaires :

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

(1)(obsolète) di = durée de la tache i

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

(2)(obsolète) il existe un unique (i,j) tel que : $(rijt) = 1 => (y_{ij} = 1)$

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

(3)(obsolète) Une tache est executé sur une seule machine

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

Une tâche est executée une fois

$$\sum_{j=0}^{m} \sum_{k=0}^{t_{max}} r_{ijk} = di, \ \forall i \tag{6}$$

Pour (i,j), on doit avoir $d_{i,j}$ variables $r_{i,j,k}$ supperieures à 1 (le temps d'execution 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 $\gamma_{i,j} \leq \sum_k r_{i,j,k} \leq M.\gamma_{i,j}$
- $\gamma'_{i,j} = 1 \Leftrightarrow \sum_k r_{i,j,k} = dij$ 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 $\gamma_{i,j} \leq \gamma'_{i,j}$