Ordonnancement régulé pour systèmes embarqués fiables

Sujet de stage ENS-L3, année 2005-2006

Alain Girault
INRIA Rhône-Alpes, projet POP-ART

18 janvier 2006

1 Modèle

On considère une application représentée par un DAG (N, E) de tâches avec contraintes de précédence : les nœuds du DAG sont les tâches (ensemble N, avec $|N| = n$) et les arcs sont les dépendances de données entre les tâches (ensemble E). Les n tâches sont notées t_i, avec $i \leq i \leq n$. La dépendance de donnée entre t_i et $t_{i'}$, si elle existe, est notée $t_i \triangleright t_{i'}$.

On considère de plus une architecture répartie hétérogène complètement connectée, représentée par un graphe (P, L) dont les nœuds sont les processeurs (ensemble P avec $|P| = m$) et les arcs sont les liens de communication bidirectionnels (ensemble L avec $|L| = m(m-1)/2$ puisque l'architecture est complètement connectée). Les m processeurs sont notés p_j, avec $1 \leq j \leq m$. Le lien de communication entre p_j et $p_{j'}$ est noté $l_{j,j'}$.

Le temps d'exécution de chaque tâche sur chaque processeur est donné par la fonction ρ_p :

$$\begin{array}{rcl} \rho_p : N \times P \times \Delta & \rightarrow & \mathbb{R} \\ (t_i, p_j, d) & \rightarrow & \rho_p(t_i, p_j, d) \\ & & \text{aussi noté } \rho_p(i, j, d) \text{ par commodité} \end{array} \quad (1)$$

où Δ est un ensemble fini et totalement ordonné de niveaux de qualité. On notera $\preceq$ l'ordre sur Δ :

$$\forall 1 \leq i \leq n \text{ et } \forall i \leq j \leq m, d_1 \prec d_2 \Longrightarrow \rho_p(i, j, d_1) < \rho_p(i, j, d_2) \quad (2)$$

L'hétérogénéité de l'architecture est représentée par le fait que :

$$\forall 1 \leq i \leq n \text{ et } \forall d \in \Delta, j \neq j' \Longrightarrow \rho_p(i, j, d) \neq \rho_p(i, j', d) \quad (3)$$

Le temps de communication de chaque dépendance de donnée sur chaque lien de communication est donné par la fonction ρ_l :

$$\begin{array}{rcl} \rho_l : \quad E \times L & \rightarrow & \mathbb{R} \\ (t_i \triangleright t_{i'}, l_{j,j'}) & \rightarrow & \rho_l(t_i \triangleright t_{i'}, l_{j,j'}) \end{array} \quad (4)$$

L'hétérogénéité de l'architecture est représentée par le fait que :

$$\forall t_i \triangleright t_{i'} \in E, l \neq l' \Longrightarrow \rho_l(t_i \triangleright t_{i'}, l) \neq \rho_l(t_i \triangleright t_{i'}, l') \tag{5}$$

Enfin, la probabilité de défaillance de chaque processeur $p \in P$ suit une loi de Poisson de paramètre constant λ_p, appelé taux de défaillance de p. Ainsi, la probabilité de défaillance de p pendant le temps t est $e^{-\lambda_p t}$. L'hétérogénéité de l'architecture est représentée par le fait que :

$$i \neq i' \Longrightarrow \lambda_{p_i} \neq \lambda_{p_{i'}} \tag{6}$$

En revanche, on suppose que les liens de communication sont fiables.

2 Problème

On désire ordonnancer de façon répétitive le DAG (N, E) sur l'architecture répartie (P, L), de telle sorte que chaque itération satisfasse les contraintes suivantes :

1. Le temps d'exécution doit être inférieur à une borne fixe donnée.
2. La fiabilité doit être au moins égale à une consigne qui est susceptible de changer à chaque itération.
3. Le niveau de qualité de chaque tâche doit être le plus haut possible.

On désire donc faire de l'« ordonnancement régulé », en se servant de la fiabilité comme valeur de régulation. La fiabilité d'un ordonnancement peut être ajustée en choisissant d'exécuter une tâche donnée sur un ou plusieurs processeurs de l'architecture. Comme la contrainte de temps d'exécution est constante pour chaque itération (contrainte numéro 1), le niveau de qualité des tâches servira à contre-balancer l'accroissement du temps d'exécution dû à la duplication de certaines tâches, duplication rendue nécessaire pour satisfaire la contrainte de fiabilité (contrainte numéro 2). Également, si une ou plusieurs fautes des processeurs se produit, il faut recalculer un nouvel ordonnancement sur l'architecture réduite, là encore en ajustant les niveaux de qualité des tâches pour continuer à satisfaire la contrainte de temps d'exécution. Enfin, si un ou plusieurs processeurs subit un ralentissement, cela sera répercuté sur la fonction ρ_p et le nouvel ordonnancement devra en tenir compte, toujours en ajustant les niveaux de qualité des tâches.

Dans tous les cas mentionnés ci-dessus, le fait de baisser les niveaux de qualité des tâches conduit le système à opérer en mode dégradé. L'intérêt de l'approche suivie ici est que la dégradation est régulée automatiquement.

Une des difficultés de ce travail provient du fait qu'il faut recalculer un nouvel ordonnancement à chaque itération afin de satisfaire la contrainte changeante de fiabilité. On s'orientera vers des algorithmes d'ordonnancement heuristiques, vers l'utilisation d'ordonnancements précalculés, et vers des algorithmes de recherche locale d'ordonnancements fiables à partir d'un ordonnancement donné.