Temps Réel, Tolérance aux Fautes et Cohérence dans les Systèmes Distribués

Pascale MINET

Résumé du rapport d'habilitation à diriger des recherches

Université de Versailles Saint-Quentin-en-Yvelines

soutenu le 23 septembre 1998

RESUME

Mes travaux concernent la prise en compte des contraintes **Temps Réel** (ex: échéances de terminaison des tâches, échéances de remise des messages), des contraintes liées à la **Tolérance aux Fautes** (ex: nature et nombre de défaillances de processus/réseau à tolérer pendant une durée donnée), et des contraintes de **Cohérence** (ex: respect d'invariants portant sur des objets persistants) dans les systèmes distribués.

Pour présenter mes travaux, j'utilise le cadre méthodologique fourni par **la méthode TRDF** (Temps Réel, Traitement Distribué, Tolérance aux Fautes) *[Le Lann & al 93a]* [Le Lann 96] [Le Lann 98]. J'ai contribué à la mise en oeuvre de cette méthode dans cinq contrats avec des industriels. Les problèmes étudiés sont regroupés en deux classes: ceux relatifs à la communication et ceux relatifs au traitement distribué.

Problèmes relatifs à la communication

J'ai activement contribué à la normalisation :

- **norme militaire française : GAM-T-103** pour les services transport temps réel, GAM-T-111 pour les protocoles transport temps réel associés, dont j'ai évalué les performances par simulation *[Minet 89a]*;
- **norme européenne ETSI : HIPERLAN** pour réseau local radio; ma contribution concerne la définition des services HIPERLAN et la spécification du protocole de **routage** *[Minet 94]*. L'originalité de ce protocole réside dans le concept des relais multipoint, qui permet une gestion efficace de la mobilité et des communications multipoint.

J'ai encadré deux thèses portant entre autres sur la **diffusion atomique uniforme au sein d'un groupe**, en modèle à Délais Bornés Connus (DBC). Une diffusion atomique est une diffusion fiable avec un même ordre de remise des messages pour tous les processus corrects du groupe. Les défaillances tolérées, en nombre borné, sont de type arrêt pour les processus et de type omission pour le réseau.

- **Le premier protocole, ABP** (Atomic Broadcast Protocol), proposé dans [Anceaume 93], s'appuie sur la vue du groupe. La tolérance aux fautes est de type détection/ recouvrement. Ses performances ont été évaluées par simulation.
- **Le deuxième protocole, plus exactement la famille de protocoles** de diffusion atomique proposée dans [George 98], ne s'appuie pas sur la vue du groupe et offre un ordre total global préservant localement/globalement l'ordre FIFO/causal/ chronologique. La tolérance aux fautes est basée sur le masquage. Les contraintes temps réel sont garanties sous réserve de satisfaire les conditions de faisabilité temps réel.

Dans le cadre du projet ATR, j'ai contribué à la conception de la composante temps réel de deux algorithmes de diffusion atomique uniforme *[Charron-Bost & al 97b]*. L'un est conçu en modèle DBC et l'autre est conçu en modèle à Délais Non Bornés (DNB), destiné à être plongé en fonctionnement opérationnel, dans un environnement DBC.

Problèmes relatifs au Traitement Distribué

Nous supposons que toute tâche prise isolément préserve la cohérence du système. La cohérence peut être détruite par les exécutions conflictuelles des tâches et par les défaillances. Nous nous sommes donc intéressés à la gestion des conflits et à la gestion des défaillances sous contraintes temps réel. Nous avons étudié deux approches pour gérer les conflits : l'évitement et la détection/résolution.

L'approche évitement des conflits est basée sur le maintien d'un ordre total global à chaque classe d'équivalence de l'ensemble des tâches. Cet ordre peut être fourni par l'ordonnancement ou par la diffusion atomique. Nous avons étudié deux modèles de tâches :

- **le modèle de tâches en étoile** (ce modèle n'induit pas de synchronisation entre les serveurs de la tâche). Dans *[George & al 97]*, chaque serveur exécute les tâches selon l'ordre de leurs demandes d'activation. Cet ordonnancement garantit le respect des contraintes temps réel, sous réserve de satisfaire les conditions de faisabilité nécessaires et suffisantes.
- **le modèle de tâches en graphe**. Avec ce modèle, l'établissement de conditions de faisabilité nécessaires et suffisantes se heurte à une combinatoire élevée (nombre de scénarios pires cas et longueur de ces scénarios), et ce même pour un ordonnancement périodique prédéfini. Un algorithme de construction des scénarios pires cas, ainsi que quelques propriétés permettant de réduire la longueur des scénarios testés et leur nombre sont donnés dans [Kamoun 97]. La formalisation de ce problème fait l'objet d'une thèse. Afin de réduire cette combinatoire, une approche classique consiste à introduire des constructions simplificatrices. On obtient alors des majorants des temps de réponse et donc des conditions de faisabilité seulement suffisantes (voir les travaux du groupe Reflecs dans le cadre du contrat ORECA *[Anceaume & al 96a]*).

L'approche détection/résolution des conflits utilise :

- **soit le verrouillage** : le verrouillage avec évitement des interblocages peut être utilisé par un contrôle d'admission ; voir par exemple *[George & al 98a]* pour un modèle de

tâches en arbre sans objet persistant, les conditions de faisabilité, obtenues selon l'approche holistique, sont seulement suffisantes.

• **soit l'estampillage** : voir par exemple l'étude SIGMA (*[Minet & al 87b]* et *[Boudenant & al 84]*) où le contrôle d'admission des tâches est réalisé par un couplage ordonnancement/contrôle de concomitance : une tâche acceptée par le système (tâche dite garantie) est sûre, en l'absence de défaillance, d'accéder à des valeurs cohérentes et de respecter son échéance. Les tâches sont apériodiques. Elles ont une structure en graphe, chaque action du graphe a une date de démarrage au plus tôt et une date de démarrage au plus tard. Ces dates, imposées par l'applicatif, sont utilisées par le système, ainsi que l'ordre sériel équivalent, pour décider de l'admission d'une tâche. L'ordre sériel équivalent est l'ordre des échéances absolues de terminaison des tâches.

Vis-à-vis des défaillances, un système peut devoir :

• **offrir le même service qu'en l'absence de défaillance**. Nous avons considéré des tâches répliquées sur plusieurs serveurs. Ceci correspond au modèle de tâches en étoile.

•• **en redondance active** : les solutions décrites dans l'approche évitement des conflits sont applicables (voir par exemple *[George & al 97]*).

•• **en redondance semi-active**, une diffusion fiable des points de reprise maintenant le synchronisme de vue du groupe destinataire est proposée.

• **maintenir l'atomicité malgré les défaillances.** Un protocole de validation atomique est utilisé. L'impact sur les temps de réponse des tâches peut être évalué en incluant la validation atomique dans le modèle des tâches.

• **détecter les défaillances des serveurs et les compenser.** Des indications sont données pour évaluer l'impact des détections/compensations sur les temps de réponse des tâches.