



LABORATOIRE D'INFORMATIQUE DE PARIS 6

Systemes Répartis et Coopératifs

**De l'amélioration des performances
par la réplication des objets :
une approche par estampillage**

J.-Ph. Lesot

THÈSE de DOCTORAT de l'UNIVERSITÉ PARIS 6
LIP6 1999 / 006 mars 1999

Résumé : L'amélioration des performances d'un service distribué vis-à-vis de ses utilisateurs est un critère important de satisfaction. Or, les canaux de communication introduisent des temps de propagation et de latence qui pénalisent le client. L'utilisation de portables connectés par intermittence ou avec des débits faibles et irréguliers ajoute des contraintes supplémentaires. Notre travail a comme objectif principal d'améliorer les performances des clients dans un système distribué à objets. Notre contribution principale est d'une part un cadre de travail (architecture partielle d'un système à objets utilisant la réplication active) et d'autre part un protocole d'estampilles d'ordonnancement. Tout d'abord, le protocole des estampilles d'ordonnancement permet à une requête de transporter avec elle toutes les informations nécessaires à sa livraison dans un ordre imposé par un modèle de cohérence. L'implantation de ce modèle se résume alors à utiliser ce protocole pour déterminer un ordre sur les requêtes en les identifiant et les estampillant. Cela dissimule aux concepteurs de serveurs les problèmes de propagation, de stockage, de remise en ordre et de livraison des requêtes à l'application. Nous espérons ainsi augmenter l'utilisation d'un modèle de cohérence strictement nécessaire, voire d'un modèle spécifique, à l'application en proposant une solution simple et efficace. Ensuite, dans notre cadre de travail, chaque objet gestionnaire fournit, indépendamment des autres, des fonctionnalités précises permettant à chacun de tirer partie de son environnement, et d'être facilement interchangeable. Entre ces différents objets, les communications sont les plus asynchrones possibles et les interactions sont au maximum locales. Enfin, l'implantation d'un prototype montre le coût limité du protocole des estampilles d'ordonnancement, ainsi que sa facilité d'utilisation pour implanter des modèles de cohérence et des mécanismes de synchronisation.

Mot clés : systèmes distribués, systèmes à objets, performance, cohérence, réplication active, synchronisation.

Abstract: Speeding up distributed application is a significant quality criteria for users. Communication channels have associated propagation time as well as latency which slow clients down. The use of wireless appliances connected on a discontinuous basis or with a low and/or irregular bandwidth add even more constraints to the system. Our main goal is to improve server performance in an distributed objects' system. Our main contributions are (i) a framework, a partial architecture design of an objects' system using active replication, and (ii) an invocation scheduling stamp protocol. Firstly, the invocation scheduling stamp protocol allows a query to carry all the necessary information required to respect a delivery order fixed by a coherency model. The implementation of this model is simply done using our protocol to establish an order between queries by identifying and stamping them. This method hides the query propagation, spooling, reordering and delivery problems to application. Thus, we hope to increase the use of a strictly necessary coherency model, or even the use of application specific model, by suggesting a simple and efficient way. Secondly, in our framework, each manager provides a focused functionality, independently of other managers, allowing each one to take advantage of its environment and to be easily interchangeable. Asynchronous communications between those different managers are favoured as well as local interactions. Finally, we have built a prototype of the framework. Its shows the low cost of the invocation scheduling stamp protocol and how easy it is to implement coherency models and synchronization mechanisms with it.

Keywords: distributed systems, objects' systems, performance, coherency, active replication, synchronization.

Table des matières

Table des figures	vii
Liste des tableaux	ix
Introduction	1
1 Communication dans les systèmes distribués	5
1.1 Définition	5
1.2 Motivations	6
1.2.1 Partage de ressources	6
1.2.2 Amélioration des performances	6
1.2.3 Disponibilité et fiabilité	6
1.2.4 Extensibilité	6
1.3 Modèles du système	6
1.3.1 Synchrone	6
1.3.2 Asynchrone	7
1.3.3 Partiellement synchrone	7
1.4 Modèles de programmation	7
1.4.1 Passage de messages	7
1.4.1.1 De un vers un	7
1.4.1.2 De un vers plusieurs	9
1.4.1.3 Conclusion	12
1.4.2 Appel de procédures à distance	12
1.4.2.1 Passage de paramètres	12
1.4.2.2 Hétérogénéité	12
1.4.2.3 Sémantique	13
1.4.2.4 Synchrone <i>vs</i> asynchrone	13
1.4.2.5 Appel de procédures parallèles à distance	14
1.4.2.6 Conclusion	14
1.4.3 Invocation de méthodes à distance	14
1.4.4 Mémoire virtuellement partagée	15
1.4.4.1 Structure et granularité	16
1.4.4.2 Cohérence	16
1.4.4.3 Hétérogénéité	17
1.4.4.4 Tolérance aux fautes	17
1.4.4.5 Conclusion	17

1.5	Conclusion	17
2	La réplication	19
2.1	Types de réplication	19
2.1.1	Réplication passive	20
2.1.2	Réplication active	20
2.1.3	Réplication semi-active	21
2.2	Comparaison des types de réplication	21
2.2.1	Tolérance au fautes	21
2.2.2	Coûts réseaux	22
2.2.3	Coûts processeurs	22
2.2.4	Performance	22
2.2.4.1	Replication passive	22
2.2.4.2	Réplication active	23
2.2.4.3	Requêtes concurrentes	23
2.3	Modèles de cohérence	25
2.3.1	Cohérences sans synchronisation	27
2.3.1.1	Cohérence atomique	27
2.3.1.2	Cohérence séquentielle et objet	27
2.3.1.3	Cohérence causale	30
2.3.1.4	Cohérence PRAM et processeur	31
2.3.1.5	Cohérence hybride	34
2.3.1.6	Cohérence « à la longue »	34
2.3.2	Cohérences avec synchronisation	34
2.3.2.1	Cohérence faible	35
2.3.2.2	Cohérence au relâchement	36
2.3.2.3	Cohérence à l'entrée	40
2.3.3	Cohérences de session	40
2.3.3.1	« Lire ses écritures »	42
2.3.3.2	« Lectures monotones »	42
2.3.3.3	« Les écritures suivent les lectures »	42
2.3.3.4	« Écritures monotones »	43
2.3.4	Cohérence « de bout en bout »	43
2.3.5	Résumé	44
2.4	Systèmes	44
2.4.1	Grapevine	45
2.4.2	<i>Lazy replication</i>	45
2.4.3	Psync	46
2.4.4	Bayou	47
2.4.5	Isis	48
2.4.6	Arjuna	48
2.4.7	Globe	49
2.4.8	CORBA	50
2.4.8.1	Par intégration	50
2.4.8.2	Par service	51
2.4.8.3	Par interception	51
2.5	Conclusion	52

3	Estampilles d'Ordonnancement : principes et utilisations	53
3.1	Définitions et terminologies	53
3.1.1	Objets	53
3.1.2	Invocations	54
3.2	Estampilles d'ordonnancement	54
3.2.1	Relation d'ordre strict	54
3.2.2	Graphe de dépendance	55
3.2.3	Estampilles d'ordonnancement	55
3.2.3.1	Principe	55
3.2.3.2	Représentation du graphe	56
3.2.4	Liste d'estampilles	56
3.2.4.1	Principe	56
3.2.4.2	Représentation du graphe	57
3.2.5	Graphe de dépendance direct	57
3.2.6	Utilisations	58
3.2.6.1	Notation	58
3.2.6.2	Architecture	59
3.2.6.3	Maintien de l'ordre programme	60
3.2.6.4	Maintien de la cohérence atomique objet	60
3.3	Optimisations	62
3.3.1	Diffusion des lectures	62
3.3.1.1	Problème	62
3.3.1.2	Estampilles surchargées	64
3.3.1.3	Estampilles invalides	67
3.3.1.4	Dépendance de propagation	68
3.3.1.5	Graphe de dépendance transformé	69
3.3.1.6	Représentation du graphe	70
3.3.1.7	Utilisation	70
3.3.2	Taille de la liste d'estampilles	70
3.3.2.1	Problème	70
3.3.2.2	Groupe d'invocations	71
3.3.2.3	Connaissance du cardinal	74
3.3.2.4	Représentation du graphe	75
3.3.2.5	Groupe d'invocations <i>vs</i> liste d'estampilles	76
3.3.3	Utilisations	76
3.3.3.1	Maintien de la cohérence atomique objet « inverse »	77
3.3.3.2	Gestion d'exclusion mutuelle	77
3.4	Principe général et propriétés	79
3.4.1	Vivacité	81
3.4.2	Sûreté	81
3.5	Travaux similaires	82
3.6	Conclusion	83

4	Estampilles d'Ordonnancement : cadre de travail	85
4.1	Vue d'ensemble	85
4.1.1	Gestionnaires	85
4.1.2	Interactions client / serveur	86
4.2	Rôle des gestionnaires	87
4.2.1	Plate-forme sous-jacente	87
4.2.2	Objet serveur	87
4.2.3	Objet contrôle	90
4.2.4	Gestionnaire d'estampilles	91
4.2.5	Gestionnaire d'ordonnancement	91
4.2.6	Gestionnaire de propagation	95
4.3	Parcours d'une invocation	95
4.3.1	Du client à la copie primaire du serveur	95
4.3.1.1	Performance	99
4.3.1.2	Tolérance aux fautes	101
4.3.2	De la copie primaire aux autres copies du serveur	102
4.3.2.1	Performance	103
4.3.2.2	Tolérance aux fautes	103
4.3.3	Des copies synchrones à la copie primaire du serveur	105
4.3.4	Gestion de la propagation « à la demande »	105
4.3.5	Exécution et retour des paramètres au client	107
4.3.5.1	Performance	107
4.3.5.2	Erreur d'exécution locale	108
4.4	Cycle de vie	110
4.4.1	Barrières d'invocations	110
4.4.2	Création d'une copie	111
4.4.2.1	Performance	112
4.4.2.2	Tolérance aux fautes	112
4.4.3	Destruction d'une copie	112
4.4.3.1	Destruction « douce »	112
4.4.3.2	Destruction « violente »	113
4.4.4	Recyclage des identificateurs	113
4.4.4.1	Performance	114
4.4.4.2	Tolérance aux fautes	114
4.4.4.3	Déconnexion importante	115
4.5	Autres interactions	115
4.5.1	Interactions entre les gestionnaires d'estampilles	115
4.5.2	Interactions avec un répartiteur de charge	116
4.5.3	Interactions avec d'autres gestionnaires	117
4.6	Conclusion	117
5	Estampilles d'Ordonnancement : implantation	119
5.1	Implantation d'un prototype	119
5.2	De la théorie à la pratique	120
5.2.1	Finitude des ressources et de l'espace des identificateurs	120
5.2.2	Degré entrant	121
5.2.3	Degré entrant distribué	121

5.3	Gestionnaire d'ordonnancement	122
5.3.1	État d'un identificateur	122
5.3.2	Transition d'état	122
5.3.2.1	Transitions du cycle	125
5.3.2.2	Transitions arrières	126
5.3.2.3	Transitions pour l'abandon	126
5.3.3	Structures de données	127
5.3.4	Algorithmes	127
5.3.4.1	Ajouter une nouvelle invocation	127
5.3.4.2	Délivrer une invocation prête	128
5.3.4.3	Délivrer un identificateur attendu	129
5.3.4.4	Notifier la fin de l'exécution d'une invocation	129
5.3.5	Analyse du coût d'une invocation	129
5.4	Gestionnaire d'estampilles	131
5.4.1	Librairie d'identificateurs	131
5.4.2	Exemple de gestionnaires d'estampilles	134
5.4.2.1	Modèle de cohérence « à la longue »	134
5.4.2.2	Modèle de cohérence « lire ses écritures »	134
5.4.2.3	Modèle de cohérence atomique objet	138
5.4.2.4	Exclusion mutuelle	141
5.5	Évaluation	145
5.6	Conclusion	147
	Conclusion	149
	Bibliographie	151