

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université des Sciences et de la Technologie Houari Boumediene (USTHB)
Faculté d'Électronique et Informatique



THÈSE

Présentée pour l'obtention du grade de **DOCTEUR**

En : **INFORMATIQUE**

Spécialité : **Systemes Informatiques**

Par : **ZEGHACHE Linda**

Sujet :

**Tolérance aux pannes dans les systèmes d'agents
mobiles transactionnels**

Soutenue publiquement, le 27/04/2014, devant le jury composé de :

M AHMED NACER Mohamed	Professeur	à l'USTHB	Président
M BADACHE Nadjib	Professeur	au CERIST	Directeur de thèse
M HURFIN Michel	Directeur de Recherche	à l'INRIA	Examineur 1
M ALIOUAT Makhlouf	Maître de Conférence/A	à l'Univ. Sétif1	Examineur 2
M ZAFOUNE Youcef	Maître de Conférence/A	à l'USTHB	Examineur 3
M TANDJAOUI Djamel	Maître de Recherche/A	au CERIST	Examineur 4

Table des matières

Introduction générale	1
1 Agents mobiles et tolérance aux pannes	5
1.1 Les agents mobiles : Concepts de base	6
1.1.1 Communication par messages vs. mobilité	7
1.1.2 Définition d'un agent mobile	8
1.1.3 Structure d'un agent mobile	9
1.1.4 Migration d'un agent mobile	9
1.1.5 Modèle d'exécution d'agents mobiles	10
1.1.6 Les plateformes d'agents mobiles	11
1.1.7 Avantages des agents mobiles	14
1.1.8 Agents mobiles et applications réparties	15
1.1.9 Limites des agents mobiles	18
1.2 La tolérance aux pannes	20
1.2.1 Sûreté de fonctionnement	20
1.2.2 La tolérance aux pannes dans les systèmes répartis	21
1.2.3 Gestion de la redondance	22
1.3 La tolérance aux pannes dans les systèmes d'agents mobiles	23
1.3.1 Modèles de fautes dans les systèmes d'agents mobiles	23
1.3.2 Propriétés des agents mobiles tolérants aux pannes	25
1.3.3 Les techniques de tolérance aux pannes pour les agents mobiles	25
1.3.4 Les approches de tolérance aux pannes des agents mobiles	27
1.4 Conclusion	33
2 Agents mobiles et support transactionnel	34
2.1 L'approche transactionnelle	34
2.1.1 Définition d'une transaction	34
2.1.2 Exécution d'une transaction	35
2.1.3 Propriétés d'une transaction	35
2.2 Les transactions distribuées	36
2.2.1 Définition	36
2.2.2 Moniteur de transaction	37
2.3 Les modèles de transactions avancés	38
2.3.1 Les transactions avec points de reprise et les transactions enchaînées	38

2.3.2	Modèle des transactions emboîtées	38
2.3.3	Modèles des sagas	39
2.3.4	Modèle des transactions flexibles	39
2.4	Les agents mobiles transactionnels	40
2.4.1	Le modèle d'exécution atomique	40
2.4.2	Agents mobiles transactionnels vs non transactionnels	41
2.4.3	Le modèle de fautes	41
2.4.4	Propriétés des agents mobiles transactionnels	42
2.5	Approches pour les agents mobiles transactionnels	43
2.6	Synthèse	47
2.7	Conclusion	48
3	La validation atomique non bloquante	49
3.1	Définition	49
3.2	Les protocoles de validation atomique dans les systèmes synchrones	50
3.2.1	Le protocole de validation à deux-phases	50
3.2.2	Le protocole de validation à trois phases	51
3.3	La validation atomique dans les systèmes asynchrones	53
3.3.1	Les détecteurs de pannes	53
3.3.2	L'élection de leader (détecteur de meneur)	54
3.3.3	Les protocoles de validation atomique non bloquante dans les systèmes asynchrones	55
3.4	Le problème de consensus	60
3.4.1	L'algorithme de Chandra et Toueg	61
3.4.2	Le protocole Paxos	62
3.4.3	Les optimisations du protocole Paxos : Le Paxos rapide	65
3.4.4	Fast Paxos adapté pour les consensus multiples intégrés (MIC Paxos)	67
3.5	Le consensus et la validation atomique	71
3.5.1	Un protocole de validation basé sur Paxos	71
3.6	Conclusion	72
4	Solution Proposée	73
4.1	Le modèle du système d'agents	74
4.1.1	Agent mobile	74
4.1.2	Agent de veille (Watch agent)	74
4.1.3	L'itinéraire	75
4.1.4	Places alternatives	75
4.1.5	Le gestionnaire de transactions (Transaction manager)	75
4.1.6	Les fautes d'infrastructure	75
4.1.7	L'indisponibilité de service	76
4.2	Le modèle transactionnel	76
4.2.1	La transaction distribuée	76
4.2.2	La requête	76
4.3	Le protocole de tolérance aux pannes	78
4.3.1	L'exploration de chemins et la réservation de services	78
4.3.2	Détection de fautes et recouvrement	81
4.3.3	Implémentation des agents	85

4.4	La validation atomique	89
4.5	Les services du gestionnaire de transactions	92
4.5.1	Une approche basée sur l'accord	93
4.5.2	Implémentation des services d'accord	96
4.6	Synthèse	103
4.7	Conclusion	104
5	Implémentation et Performance	105
5.1	Environnement de développement	105
5.2	Environnement de test	106
5.3	Intégration du protocole dans le code de l'agent mobile	106
5.4	Résultats de l'expérimentation	106
5.4.1	Panne de l'agent mobile	108
5.4.2	Panne de la place	109
5.4.3	Panne Sémantique	110
5.4.4	Panne de l'agent de veille	111
5.4.5	Panne catastrophique	112
5.4.6	La validation atomique	113
5.5	Fonctionnement du protocole en cas de pannes	115
5.6	Conclusion	118
	Conclusion générale	120
	Bibliographie	127
	A Les Publications	128

Résumé

Le concept d'agents mobile transactionnel a été introduit comme un amalgame entre la technologie d'agents mobile et le modèle transactionnel. durant son parcours, l'agent visite n places pour satisfaire les n requêtes de la transaction. Les nœuds visités gardent assez d'information afin de pouvoir valider ou annuler la transaction globale.

Les travaux de recherches sur les agents mobiles transactionnels ont identifiés deux problèmes que l'agent seul ne pas résoudre. Le premier est lié à la fiabilité des agents. Des agents et des nœuds peuvent tomber en pannes et donc, des mécanismes de détection de pannes et de recouvrement doivent être définis. La solution classique consiste à utiliser la dernière place visitée pour contrôler l'agent quand il migre vers une nouvelle place. Dans ce cas, le dernier nœud visité qui détient le mécanisme de détection risque de tomber en panne et doit être contrôlé à son tour. En conséquence, toutes les places visitées (y compris le client ayant initié la transaction) doivent former une chaîne où chaque nœud contrôle son successeur. Lorsque l'agent est en panne, un nouvel agent est créé par la place qui a suspecté la panne.

Il à noter que dans un système réparti asynchrone, où les détecteurs de pannes ne sont pas fiables, le lancement d'un nouvel agent ne garantit pas son unicité. De plus, le client, étant le premier nœud de la chaîne de contrôle, il doit rester connecté durant l'exécution de toute la transaction.

Le second problème pour lequel l'agent doit être assisté est lié la validation atomique de la transaction. Une fois l'agent aura identifié et visité les n nœuds capables de satisfaire les n requêtes de la transaction, il doit déléguer la supervision du processus de validation à une entité externe et surtout fiable. Pour plusieurs raisons, cette tâche ne devrait pas être faite uniquement par l'agent. Tout d'abord, par définition, un agent mobile est une entité qui exécute un code local sur le nœud où il se trouve. Cependant, un protocole de validation est un service distribué qui exige une communication entre l'initiateur et les n participants. D'autre part, la puissance du calcul d'un agent dépend de son nœud hôte qui peut avoir des capacité de calcul, de stockage ou des ressources limités. Enfin, le principe d'un protocole de validation atomique repose sur le fait qu'un coordinateur existe, ou généralement plusieurs, et que chaque coordinateur agit de d'une manière conforme à ce qui a été fait par les coordinateurs précédents. Pour répondre à toutes les difficultés avancées ci-dessus, une entité centralisée et surtout fiable peut être définie pour assister les agents mobiles dans leurs tâches de contrôle, de validation et de journalisation.

En effet, nous proposons une approche uniforme basée sur le concept d'accord pour fournir deux services, la disponibilité de la source (AS) et la validation atomique (AC), qui offrent un support pour l'exécution de l'agent d'une manière efficace, fiable et homogène.

Abstract

The concept of transactional mobile agent has been introduced as a mix between the agent technology and the transactional model. During its move, the agent discovers and visits n places that satisfy the n requests. During the agent's visit, a node records enough information so that it can subsequently either commit or abort the transaction.

Different works on transactional mobile agents have identified two problems that can not be solved by the agent alone. The first one is related to the reliability of the agent. Agents and nodes may fail. Failure detection and recovery procedures have to be defined. A classic solution consists in using the last visited node to monitor the state of an agent once it moves to another place. In that case, the last visited node which is hosting a failure detection mechanism becomes also a critical point of failure and thus, it has also to be monitored. One thing leading to another, all the nodes already visited by the agent (and the client who initiates the transaction) have to form a chain where each node observes its successor. When the agent is suspected to be crashed, a new agent is created by the node that suspects the occurrence of a failure.

Note that in an asynchronous distributed system where failure detectors are sometimes unreliable, the generation of a new agent in case of problems no more guarantees its uniqueness. Since the client is the first node in the monitoring chain, it has to remain connected during the whole execution of the transaction. The second problem for which the agent needs assistance is related to the atomic validation of the transaction. Once an agent has identified and visited n nodes able to satisfy the n requests of a transaction, it has to delegate the supervision of the validation process to an external reliable entity. For several reasons, this final task should not be made solely by the agent. First, by definition, an agent is a moving entity that executes a local code on the node where it is located. But a validation protocol is typically a distributed service that requires communications between its initiator and the n involved nodes. Second, the computing power of an agent depends on its hosting node which may have limited processing, storage and power resources. Third, a validation protocol relies on the fact that a single coordinator exists or, more generally, on the fact that any coordinator acts in a manner consistent with what has been done by previous coordinators. To address all the above remarks, a centralized and reliable entity can be defined to assist all the agents in their monitoring, validation and logging tasks.

Indeed, we propose a uniform approach based on the concept of agreement to provide two services, Availability of the Source (AS) and Atomic Commit (AC), that provide support for the agent execution in an efficient, reliable and homogeneous way.