

ISSN - 2170-0656

CERIST NEWS

Bulletin d'information trimestriel

Quinzième numéro - Juin 2014

DOSSIER

AGENTS MOBILES

CENTRE DE RECHERCHE
SUR L'INFORMATION
SCIENTIFIQUE ET TECHNIQUE







Melle. ZEGHACHE LYNDA

Maître de recherche

Division réseaux
et systèmes distribués

L'informatique est en train de changer de manière assez profonde. Au départ, confinée dans les ordinateurs, elle devient de plus en plus diffuse et distribuée dans de multiples objets et fonctionnalités qui sont amenés à coopérer. La décentralisation est donc la règle et une organisation coopérative entre modules logiciels est un besoin. De plus, la taille, la complexité et l'évolutivité croissantes de ces nouvelles applications informatiques font qu'une vision centralisée, rigide et passive atteint ses limites.

Le modèle client/serveur, est actuellement le modèle incontournable pour le développement d'applications réparties. Néanmoins, ce paradigme n'arrive plus à satisfaire les nouveaux besoins des applications distribuées justifiés par l'évolution actuelle d'Internet. Nous pensons

que les services doivent être adaptés à chaque utilisateur : trouver le meilleur produit au meilleurs prix, trouver l'information pertinente en temps voulu, accéder au meilleur service et ce, au moindre coût. De plus, à l'ère de la mobilité et la communication nomade, le client/serveur impose des coûts qui freinent la généralisation de ces technologies au grand public.

Ces besoins se sont traduits par des recherches intensives autour des techniques de mobilité dans des environnements répartis afin de développer d'autres approches pouvant servir efficacement les besoins des utilisateurs et utiliser intelligemment les ressources du réseau. C'est ainsi que l'approche d'agents mobiles est apparue pour étendre et remplacer le modèle Client/Serveur.

Dans ce modèle, un agent est un

processus possédant un contexte d'exécution, incluant du code et des données, pouvant se déplacer de machine en machine (appelés serveurs) afin de réaliser la tâche qui lui est assignée. Les agents mobiles visent principalement des applications réparties sur des réseaux à grande distance, car ils permettent de répartir l'exécution sur plusieurs machines et de diminuer la charge sur des serveurs particuliers.

L'approche d'agents mobiles est mieux vue comme un outil général permettant de réaliser des applications distribuées quelconques. Elle constitue, en fait, un bon support pour les applications distribuées et un excellent support lorsqu'il s'agit d'environnement mobile.

5 Actualités

- Visite de lycéens de Bordj el Ghedir
- Annonce de la 2ème édition de la conférence internationale sur les Technologies de l'Information et de la Communication pour la gestion des catastrophes (ICT-DM'2015)

6 Événements

- IST Emag
- 6^{èmes} Journées d'Étude sur les Bibliothèques Universitaires algériennes (JEBU'14) sous le thème : L'exploitation de la production scientifique nationale, au moyen de la documentation universitaire.

9 Dossier - LA TECHNOLOGIE DES AGENTS MOBILES

Document spécial de 20 pages : 13/33

Un dossier élaboré par : **ZRGHACHE LYNDA**

Maître de Recherche A

Division réseaux et système distribués

27 Les Conseils de DZ - CERT

- Attention aux risques liés à l'utilisation des logiciels piratés.

31 Zoom sur un Projet

Agents mobiles transactionnels pour le E-commerce

38 CERIST Recherche & Formation

- Rapports de recherche internes

40 CERIST Bases de Données Documentaires

- SNDL

Visite de lycéens de Bordj el Ghedir

Dans le cadre des activités organisées par le lycée de Bordj El Ghedir, une visite pédagogique au CERIST a été effectuée par une vingtaine de lycéens, le mardi 22 avril 2014.

Au cours de cette journée, ces lycéens ont pu assister à une démonstration de la plateforme à distance « El Manhal » réalisée par le CERIST et présentée par Mme Dalila Bebbouchi, tuteur en ligne dans la formation de la PGS en Information Scientifique et Technique. Ils se sont ensuite rendus à la bibliothèque où M. Nourreddine Meftouh, responsable de la bibliothèque s'est chargé d'expliquer le fonctionnement et l'organisation de la bibliothèque d'une part, et le procédé de numérisation d'autre part. Ces lycéens ont ensuite visité le centre de calcul et ont clôturé leur journée par une visite au bloc pédagogique.



Annnonce de la 2^{ème} édition de la conférence internationale sur les Technologies de l'Information et de la Communication pour la gestion des catastrophes (ICT-DM'2015)

La Division Théorie et Ingénierie des Systèmes Informatique (DTISI) du CERIST a le plaisir d'annoncer que la 2^{ème} édition de la Conférence Internationale sur les Technologies de l'Information et de la Communication pour la gestion des catastrophes (ICT-DM'2015) sera organisée par une équipe locale en collaboration avec l'équipe du CERIST qui a organisé la première édition. Les détails concernant l'appel à contribution, les comités et le déroulement de l'évènement sont sur le site : <http://ict-dm.org>



The 2nd International Conference on Information and Communication Technologies for Disaster Management

In Rennes, Brittany, FRANCE • November 30 to December 2, 2015 •

IST Emag

L'Université M'Hamed Bougara de Boumerdès (assurant la coordination nationale et maghrébine du projet), a organisé en collaboration avec le CERIST (partenaire du projet) le dimanche 15 Juin 2014, au siège du CERIST à Ben Aknoun, le séminaire de clôture du projet Tempus IStEMag sous le thème : « Bilan final du projet IStEMag et la Signature de la Charte de pérennité du projet ». Ce Projet a été mené par un consortium international d'institutions universitaires Européennes (Belgique, France, Italie, UBTU et la Roumanie) et Maghrébines (Algérie, Tunisie, Maroc) et s'est donné pour objectif de développer des plates-formes des Archives Ouvertes Maghrébines en visant la mise en place des procédures techniques nécessaires pour assurer l'interopérabilité des différentes archives ouvertes.

Les activités mises en oeuvre dans le cadre de ce projet ont un impact important sur les capacités de recherche et d'enseignement dans



l'ensemble des universités du Maghreb. Au terme de cette journée une Charte de pérennité du projet IStEMAG a été approuvée. Elle a pour objet d'assurer, à l'issue du projet, la pérennité et la continuité des actions entamées. Par la présente charte, les signataires s'engagent à faire les efforts nécessaires et raisonnables afin d'assurer:

- La généralisation du projet et sa diffusion auprès d'autres acteurs locaux
- Son appropriation et sa prise en charge par les bénéficiaires locaux

- L'inscription des politiques adoptées dans une dynamique locale et notamment la politique d'archives ouvertes
- La prise en compte de la politique d'archives ouvertes maghrébine arrêtée tout en respectant les spécificités locales
- La durabilité et la visibilité du projet en garantissant la survie du portail fédérateur et des plateformes locales
- Le soutien technique par le maintien du comité technique créé à cet effet et la fourniture à ses membres des conditions adéquates pour la continuation des actions entamées et le transfert des compétences et du savoir-faire acquis



6^{èmes} Journées d'Etude sur les Bibliothèques Universitaires algériennes (JEBU'14) sous le thème : La production scientifique nationale.

L'exploitation de la production scientifique nationale, au moyen de la documentation universitaire, est l'enjeu principal des 6^{èmes} Journées d'étude sur les bibliothèques universitaires algériennes (JEBU' 14), ouvertes les 22 et 23 juin 2014 au - CERIST - Alger.

Dans la continuité de l'édition précédente, la thématique de politique documentaire a constitué la préoccupation de cette sixième édition de JEBU. La démarche, l'organisation et les outils ont constitué les éléments de débats de cette édition. Aussi et sous ce chapeau, les aspects formations et production scientifique nationale ont fait l'objet d'un débat et de

recommandations en vue de faire évoluer les actions entreprises durant les éditions précédentes. Il s'agit notamment de la finalisation et de la validation du programme de formation continue et des modalités de sa mise en œuvre, d'un côté et des modalités de prise en charge de la production scientifique nationale de l'autre. A l'issue de ces journées, les membres du comité de suivi des JEBU ont honoré les directeurs des bibliothèques universitaires qui ont accueilli et organisé les cycles de formations programmées, les bibliothécaires en retraite, les formateurs ayant assuré les six sessions de formation ainsi que le président du comité de suivi JEBU .



LE DOSSIER | LA TECHNOLOGIE DES AGENTS MOBILES

Document spécial de 17 pages : 09/26

Un dossier élaboré par :

ZRGHACHE LYNDA

Maître de Recherche

Division réseaux et systèmes distribués





Introduction

La multiplicité et la complexité des problèmes physiquement distribués et le besoin pressant de les résoudre en temps opportun et d'une manière efficace, placent les systèmes d'agents mobiles parmi les solutions les plus perspicaces. L'idée est de répartir la charge d'un domaine initial sur des entités. La mobilité, la communication, la coopération, la négociation, la coordination et l'organisation entre ces entités permettent d'améliorer le degré de contribution de chacune d'elle dans la résolution du problème globale.

Dans le cadre des systèmes répartis plusieurs mécanismes d'interactions ont été conçus pour faciliter la communication entre les entités des applications réparties.

Communication par message vs. mobilité

La communication entre les entités d'un système distribué peut être prise en charge par différents mécanismes. L'échange de messages fût le premier modèle proposé, qui permet aux processus de communiquer par l'envoi et la réception de messages. Ce paradigme est réalisable à travers des bibliothèques de bas niveau généralement difficiles à mettre en œuvre. De plus, cette méthode n'est pas directement intégrée aux langages de programmation et nécessite de définir un protocole de communication stricte. Ces deux contraintes ne permettent pas une programmation simplifiée aux développeurs.

Une abstraction de communication de haut niveau est fournie par le concept d'appel de procédure à distance ou RPC (Remote Procedure Call). Ce mécanisme est une abstraction permettant aux développeurs de ne pas manipuler directement le service de communication réseau mais d'appeler une procédure distante comme si elle était locale. Ainsi, un processus utilisant le RPC ne fait aucune différence entre appel local et distant. Notons qu'il faut connaître à l'avance la procédure que l'on souhaite utiliser et que cette procédure doit être disponible sur le site distant. Cette contrainte limite l'utilisation du RPC dans un système réparti ouvert. Il est parfois souhaitable d'envoyer la procédure vers un nœud distant (où se trouvent les données) et l'exécuter sur ce nœud.

- • • Ce niveau de flexibilité est assuré par un concept appelé évaluation à distance, qui est une extension naturelle du RPC. Avec l'évaluation à distance, un processus non seulement transmet le paramètre de la dite procédure mais aussi le code du procédé lui-même. Dans cette première utilisation de la mobilité, le client peut envoyer directement le code au serveur, comme dans le cas du rsh (Remote Shell) qui permet d'exécuter un script sur une machine distante.

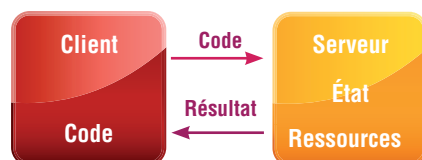


Figure1 : Evaluation à distance

Un autre schéma de mobilité appelé code à la demande est introduit. Le client dispose de l'unité d'exécution et des ressources mais va récupérer le code auprès du serveur. Il s'agit donc de l'inverse du cas précédent. Ainsi, un client adresse une requête uniquement pour récupérer un code précis afin de l'exécuter localement avec les ressources présentes. Cette méthode permet d'étendre les fonctionnalités d'une application directement chez le client (voir figure1.3). Les Applets Java reposent sur cette technologie de code mobile, il s'agit d'un programme chargé à partir d'une page Web pour être exécuté sur la machine du client.

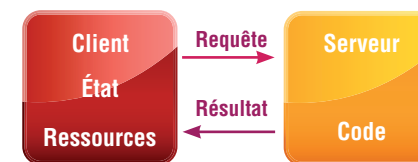


Figure 2 : Code à la demande

Par comparaison avec les deux schémas précédents qui introduisent la mobilité du code, l'agent mobile permet la mobilité du processus. Dans la mesure où le client a besoin d'interagir avec le serveur, ce même processus (code, état d'exécution et données) se déplace à travers le réseau pour continuer son exécution et pour interagir localement avec les ressources du serveur (voir figure 1.4). A la fin de l'exécution, l'agent mobile retourne éventuellement vers son client afin de lui fournir les résultats. Dans ce schéma, le savoir-faire appartient au client, l'exécution du code est initiée côté client et continuée sur les différentes machines visitées.

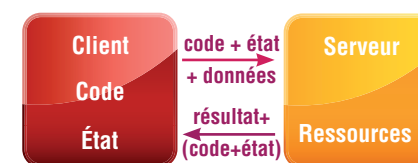


Figure 3 : Agent mobile

Définition d'un agent mobile

Les agents mobiles ont été introduits initialement en 1994 avec l'environnement Telescript qui permettait à des processus de choisir eux mêmes de se déplacer sur les sites d'un réseau afin de travailler localement sur les ressources. De cette définition découlent les propriétés suivantes :

- Comportement : Un agent est pourvu d'un programme qui lui dicte la tâche à accomplir. On dit qu'il agit par délégation pour le client. Un agent peut ainsi venir avec une compétence particulière sur une machine hôte.
- Autonomie : Un agent agit indépendamment du client. Il décide lui-même là où il va se déplacer et ce qu'il doit y faire, en fonction du comportement qui lui a été donné.
- Il n'est donc pas toujours possible de prévoir l'itinéraire des agents.
- Mémoire : Un agent mobile dispose d'une capacité de mémorisation lui permettant de récolter des informations sur les sites visités. Les informations mémorisées seront livrées par l'agent après son retour, au client.
- Environnement : L'environnement dans lequel l'agent évolue est constitué par :

- Les machines qui vont accueillir l'agent lors de son exécution ; ainsi l'agent utilise les ressources (unité de calcul, mémoire, etc...) disponibles sur chacune des machines afin d'accomplir la tâche demandée par le client,
- Les liens réseau que l'agent utilise pour se déplacer entre les différents sites,
- Les autres agents fixes et mobiles avec qui l'agent interagit afin de réaliser son but.

Attributs d'un agent mobile

Un agent mobile est une entité qui possède cinq composants: son état, son code, son interface, son identifiant et son autorité; quand un agent se déplace à travers le réseau, il transporte ses attributs.

L'état : quand un agent voyage, il transporte avec lui son état, ceci lui permet de reprendre son exécution quand il arrive à destination. L'état d'un agent peut être considéré comme une capture instantanée de son exécution avec ses données (les valeurs des variables).

Le code : comme n'importe quel autre programme, l'agent mobile a besoin d'un code pour pouvoir s'exécuter. Quand il se



déplace à travers le réseau, l'agent emporte son code avec lui. L'implémentation d'agent doit être à la fois exécutable et sans risque pour l'hôte de destination. Les langages interprétés offrent une indépendance par rapport aux plateformes ainsi qu'un environnement d'exécution contrôlé qui caractérise le mécanisme de sécurité puisqu'il limite l'accès aux ressources privées de l'hôte.

L'interface : un agent fournit une interface qui permet aux autres agents et aux autres systèmes d'interagir avec lui. Cette interface peut être un ensemble de méthodes qui permet aux autres agents et applications d'accéder aux méthodes de l'agent. L'identifiant : chaque agent possède un identifiant unique durant son cycle de vie, qui lui permet d'être identifié et localisé et qui soit géré par un service de nommage. Puisque l'identifiant est unique, il peut être utilisé comme clé dans les opérations qui exigent un moyen pour référencer une instance particulière d'agents.

L'autorité : une autorité est une entité dont l'identité peut être authentifiée par n'importe quel système auquel elle essaye d'accéder. Une autorité peut être soit une personne privée, soit une organisation. Elle est constituée d'un nom et d'autres attributs. Pour les agents, il existe principalement deux types d'autorité :

- Le fabricant (manufacturer) qui est le fournisseur du code d'implémentation de l'agent).

- Le propriétaire (owner) qui a la responsabilité du comportement de l'agent.

Migration d'un agent mobile

La migration permet le transfert d'un agent en cours d'exécution d'un site à un autre à travers le réseau. Cette migration comporte les étapes suivantes :

1. La sérialisation du contexte de l'agent et de son code, le résultat est inclus dans un message ;
2. Le processus de l'agent étant suspendu sur sa machine d'origine, toutes les communications avec d'autres agents sont donc suspendues et l'agent est détruit ;
3. L'envoi du contexte et du code de l'agent vers la machine de destination;
4. L'état de l'agent migrant est restauré sur la machine de destination, un nouveau processus léger est créé pour la poursuite de son exécution;
5. Les communications en cours sont redirigées vers la machine de destination, l'agent n'étant pas encore activé, ces messages seront traités après sa réactivation ;





6. La réactivation de l'agent sur la machine de destination, dès que la partie de son contexte nécessaire à son exécution est reçue ; la machine de destination s'occupe des liens dynamiques entre l'agent et son code. À partir de ce moment, la migration prend fin. Le processus sur la machine d'origine peut être définitivement effacé.

Les différentes stratégies de migration d'agents dépendent de la manière dont sont traitées ces diverses étapes et les données transférées avec l'agent lors de sa migration. Deux types de migrations ont été proposés dans les plateformes existantes :

A. La migration forte permet à un agent de se déplacer quelque soit l'état d'exécution dans lequel il se trouve. Dans ce type de migration l'agent se déplace avec son code, son contexte d'exécution et ses données. Dans ce cas, l'agent reprend son exécution après la migration exactement là où elle était avant son déplacement. La migration forte nécessite un mécanisme de capture instantanée de l'état d'exécution de l'agent. Elle peut être proactive ou réactive. Dans la migration proactive, la destination est déterminée par l'agent. Dans la migration réactive, la migration de l'agent est dictée par une partie ayant une relation avec l'agent mobile.

B. La migration faible ne fait transférer avec l'agent que son code et ses données. Sur le site de destination, l'agent redémarre son

exécution depuis le début en appelant la méthode qui représente le point d'entrée de l'exécution de l'agent, et le contexte d'exécution de l'agent est réinitialisé. Pour que l'agent se branche sur une instruction particulière de son code après sa migration, le programmeur doit inclure dans l'état de l'agent des moments privilégiés (explicites) dans le code de l'agent (point d'arrêt) pour pouvoir le relancer.

Modèle d'exécution d'agents mobiles

Même s'il existe de nombreuses approches pour la mise en œuvre de la migration dans les systèmes d'agents mobiles, la migration elle-même n'est qu'une partie du modèle d'exécution. L'exécution de l'agent mobile est constituée d'une séquence d'étapes appelés 'stades d'exécution'. Chaque étape est exécutée par l'agent mobile sur une place p_i , où une place p_i ($0 \leq i \leq n$) offre l'environnement d'exécution logique pour l'agent. La première et dernière place où l'agent mobile doit s'exécuter, sont appelées respectivement source et destination. La séquence de places p_0, p_1, \dots, p_n est appelée l'itinéraire de l'agent mobile. Un itinéraire statique est entièrement défini à la source et ne change pas au cours de l'exécution de l'agent, alors qu'un



- • • itinéraire dynamique subit des modifications par l'agent. Logiquement, un agent mobile s'exécute en une séquence de stades d'actions. L'agent a_i au stade S_i représente l'agent a qui a exécuté les actions sur les places p_0, \dots, p_{i-1} et s'apprête à s'exécuter sur la place p_i .

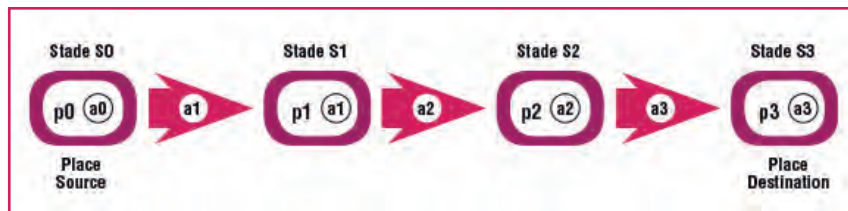


Figure 4 : Modèle d'exécution d'agents mobiles

Les plateformes d'agents mobiles

Lorsque l'on parle de plateformes, nous regroupons deux niveaux élémentaires. Le premier est le langage de programmation qui offre les primitives nécessaires à l'exécution, à la communication et à la mobilité ; il constitue le niveau de base. Le deuxième est l'intergiciel proprement dit qui offre des services de plus haut niveau, comme les annuaires, permettant aux agents la réalisation de leurs tâches.

Les langages sous-jacents

Le premier langage à proposer ces propriétés fut Telescript de GeneralMagic qui fut suivi de nombreux autres tels Obliq, Safe-Tcl, etc. Ils possèdent tous des qualités propres, comme la prise en compte de la mobilité forte par exemple, mais se voient supplantés par la popularité grandissante d'un langage à objets, à savoir Java. Une des premières plateformes à utiliser cette technologie fut Mole. En effet, grâce à sa machine virtuelle (JVM) largement répandue, Java a su s'imposer pour la programmation d'applications réparties dans des environnements hétérogènes en permettant une indépendance envers les réseaux et les systèmes d'exploitation. La migration est supportée grâce au mécanisme de sérialisation pour le transport d'objets de sites en sites, la sérialisation étant la transcription de l'état et de la structure complexe d'un objet dans un format particulier. Pourtant Java présente un inconvénient : il ne gère pas la mobilité forte. Malgré des travaux sur l'extension de la machine virtuelle, c'est toujours au développeur de gérer les différents états rencontrés après la migration de ses agents.

Les standards

L'émergence de nombreuses plate-formes expérimentales a rendu nécessaire la proposition d'une harmonisation grâce à la standardisation des différents concepts communs pouvant être identifiés. On peut trouver deux normes principales : il s'agit de la norme FIPA (Foundation for Intelligent Physical Agents) et de la norme MASIF (Mobile Agent System Interoperability Facility).

Lorsque deux normes existent au sein d'une même technologie, elles ont une tendance naturelle à s'opposer alors que, dans ce cas, elles penchent plus vers la complémentarité due à la différence de leurs domaines d'origine.

La norme MASIF a été spécifiée par l'Object Management Group (OMG) qui se préoccupe généralement de l'hétérogénéité entre les systèmes, comme dans le cas de CORBA. Dans cette optique, le but, dans la norme MASIF, est de décrire les notions élémentaires permettant l'échange des agents entre différentes plateformes. Pour ce faire, elle standardise la manière de gérer le code des agents, leur identification, la migration et l'adressage local.

En revanche, la communauté d'origine de FIPA étant celle des systèmes multi-agents, plus proche de l'intelligence artificielle, elle va se situer à un niveau plus élevé, c'est-à-dire, le niveau applicatif en décrivant les éléments nécessaires à la réalisation d'une application et principalement en détaillant la communication entre les agents. Le but est de décrire un

ACL (Agents Communication Language), les ontologies et des protocoles de négociation permettant ainsi de définir parfaitement les interactions entre les agents.

En prenant un peu de recul, on peut extraire des deux normes les différents éléments devant apparaître dans l'élaboration d'une plate-forme (figure 5) :

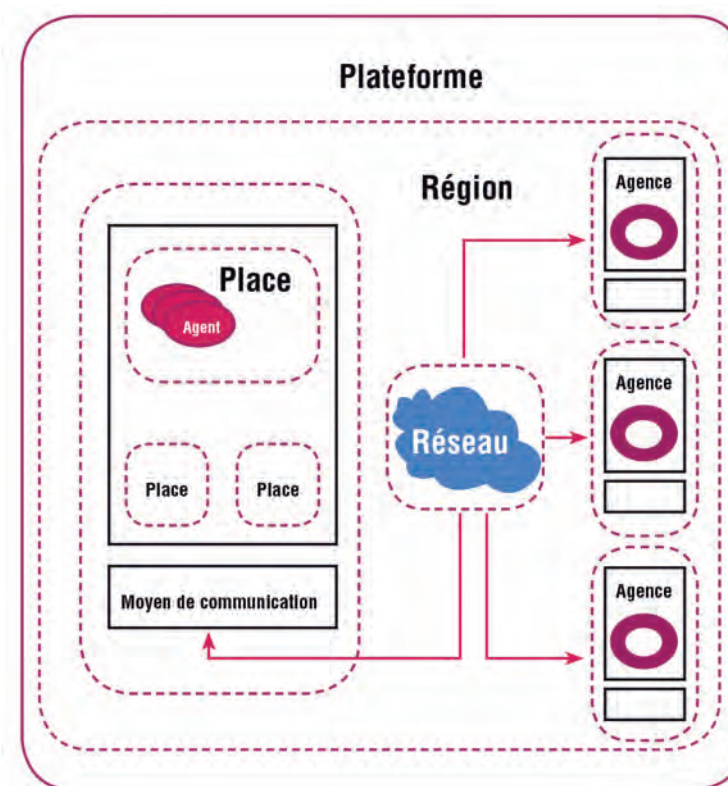


Figure 5 : Plateforme d'agents mobiles





Les agents : Ils sont définis par un identifiant unique au sein de la plate-forme et d'un état d'exécution comprenant le code et les données qu'ils transportent.

Les places : C'est l'élément qui sera en charge d'exécuter le code de l'agent. Une place peut contenir plus d'un agent. Une place définit un environnement d'exécution vers lequel un agent sera orienté pour gérer son exécution.

Les agences : Encore appelées «ports d'accueil» des agents, représentent les environnements où évoluent les agents. Une agence constitue le cœur de gestion de la plate-forme en assurant la délégation de l'exécution aux places qui la composent, l'administration, le contrôle, le transport effectif et la communication des agents grâce au système d'exploitation sous-jacent.

Les régions : Il s'agit du regroupement de plusieurs agences, pas nécessairement de même type, appartenant à un même domaine d'expertise, d'activité ou autre. Le but est de faciliter les activités d'administration. On peut associer par exemple une région à des politiques d'accès aux services.

Les plateformes existantes

Un agent mobile s'exécute sur une plateforme qui lui offre des services pour son déplacement, sa localisation et la communication avec les autres agents. Ainsi, la performance et la fiabilité de l'environnement d'exécution vont être déterminantes pour la qualité d'une application basée sur la technologie d'agents mobiles. Une centaine de plateformes existent à nos jours ; elles se différencient par le type de migration de l'agent (faible ou forte), par la sécurité et la protection de l'agent et du site d'accueil, par la fiabilité offerte lors de l'exécution (tolérance aux pannes), par la facilité de retrouver un agent mobile, par les facilités d'implémentations offertes aux programmeurs et par les services offerts pour la communication entre les agents. Dans le tableau suivant, une liste des plateformes d'agents mobiles les plus connues, issues des milieux industriels et commerciaux, et des travaux de recherche, est présentée.

	Plateforme	Institution	Standard	Langage	Mobilité	Communication
Issues des milieux Indus. et commerc.	Telescript	General Magic	—	Python	forte	Evènement
	Voyager	Objectspace	—	java	faible	Proxy
	Aglets	IBM	—	java	faible	Messages
	Grasshopper	GMD FOKUS and IKV++	MASIF, FIPA	java	faible	Evènements
	Jade	Telecom Italia	FIPA	java	faible	ACL
Issues des milieux de recherche	Agent Tcl	Dartmouth College	—	Tcl	forte	Messages
	Tacoma	Univ. of Tromsø, Cornell univ.	—	Tcl, Perl, C, Scheme, Python	forte	Fichier
	Ara	Univ. of Kaiserslautern	—	java	faible	Messages
	Javact	IRIT Lab	—	java	faible	Messages
	Nomads	DARPA, NASA, Univ of. Florida	—	Java aroma vm	forte	RMI

Avantages du paradigme Agent Mobile

Les premières améliorations apportées par les agents mobiles concernent le gain de performance dû à une meilleure utilisation des ressources physiques mises à disposition. L'amélioration interviendra à différents niveaux et permettra d'optimiser la tâche globale des agents.

1. Réduction du trafic réseau

Le déplacement des agents mobiles permet de réduire significativement, voir supprimer, les communications distantes entre les clients et les serveurs. En privilégiant les interactions locales, l'utilisation du réseau va se limiter principalement au transfert des agents. Cette situation présente trois principaux avantages.

Le premier avantage est la diminution de la consommation de bande passante. En effet, plusieurs études montrent qu'en comparaison de l'envoi à distance de requête (procédures et méthodes), la mise en place des agents mobiles permet d'obtenir une réduction significative de la charge réseau en termes du nombre total de données transférées. Cette diminution est constatée dans différents types d'applications nécessitant d'intenses échanges d'informations entre le client et le serveur.

Le second avantage notable est la diminution des temps de latence. Dans le contexte des réseaux à large échelle, la mise en place d'applications réparties, nécessitant de fréquentes interactions entre client et serveur, se heurte aux temps de latence propres aux communications réseaux. Il arrive fréquemment que le temps d'attente de la réponse d'une requête soit plus long que le temps de traitement nécessaire à la réalisation du service. En rapprochant client et serveur dans un même sous-réseau, voire sur un même site, on les place dans un environnement où les temps de réponse des interactions sont limités, ce qui permet de réduire d'autant les temps de latence.

Le dernier avantage à souligner vient des brèves périodes de communication. En réduisant le plus possible les communications distantes aux seuls transferts d'agents mobiles, on diminue considérablement les périodes de connexion entre deux sites. Cette diminution de la fenêtre d'utilisation des communications réseaux permet de moins se soucier des ruptures de liens physiques qui peuvent intervenir fréquemment dans les environnements sans fil.

2. Exécution asynchrone et autonome

Dans le modèle classique d'évaluation à distance, le client et le serveur doivent rester connectés tant que le service est en cours d'exécution. En plus des problèmes liés aux fluctuations des performances réseaux, certains services, nécessitant de longues phases de traitement, ne



- ● ● supportent pas facilement une rupture de connexion avec le client. Dans ce cas, ils doivent souvent redémarrer entièrement leurs calculs. Mais, le maintien du lien de communication peut s'avérer difficile dans des réseaux à large échelle ou sans fil. Avec les agents mobiles, un client peut déléguer les interactions avec le service sans maintenir une connexion de bout en bout. Avec cette possibilité d'un calcul indépendant, le client peut demander un service, se déplacer (ou simplement terminer une session) puis venir récupérer les résultats plus tard. Ce mode de fonctionnement est particulièrement intéressant lors du lancement à distance de simulations numériques.

3. Optimisation du traitement

L'optimisation des phases de traitement se produit à deux niveaux. Premièrement, comme nous l'avons déjà dit, en localisant les ressources et le savoir faire sur un même site, on supprime les phases de dialogue entre le client et le serveur qui sont perturbées par des temps de latence dus aux communications réseaux. Ensuite, le déplacement du savoir faire va permettre de déléguer les calculs sur des machines serveurs, un super-calculateur par exemple, qui sont généralement plus puissantes qu'une machine cliente. Cela est particulièrement vrai dans l'informatique nomade où la miniaturisation s'accompagne d'une perte de puissance significative.

4. Tolérance aux fautes physiques

En se déplaçant avec leur code et données propres, les agents mobiles peuvent s'adapter facilement aux erreurs systèmes. Ces erreurs peuvent être d'ordre purement physique, disparition d'un nœud par exemple, ou d'ordre plus fonctionnel, arrêt d'un service par exemple. Si on prend le cas d'un site perdant une partie de ses fonctionnalités, un service tombant en panne, l'agent pourra alors choisir de se déplacer vers un autre site contenant la fonctionnalité désirée. Ceci permet une meilleure tolérance aux fautes que le modèle statique classique.

Ces avantages font des agents mobiles une technologie appropriée et bénéfique pour différents domaines d'application.

Agents mobiles et applications réparties

Les principaux atouts des agents mobiles, à savoir, l'autonomie, l'adaptabilité et des interactions locales, sont très utiles lorsque l'on souhaite mettre en place des explorations de réseau, représenter un utilisateur nomade ou encore lorsque les environnements sont autonomes et décentralisés.

- • • Certaines applications réparties ne nécessiteront pas le modèle agent mobile, mais d'autres se révéleront plus efficaces en l'utilisant. Voici quelques exemples :

1. Le commerce électronique

Le commerce électronique peut se définir comme toute transaction dont les parties utilisent comme support de transmission un support électronique. Avec l'évolution du réseau des terminaux mobiles et la possibilité de les utiliser pour se connecter sur le réseau Internet, nous avons vu l'apparition d'une nouvelle forme de commerce électronique, le M-commerce. Le M-commerce ou mobile commerce est une forme du e-commerce où une transaction peut s'effectuer à partir d'un PC mais également à partir d'un terminal mobile comme un téléphone mobile ou un assistant personnel (smartphone). Le développement de l'ubiquitous computing (n ordinateurs pour 1 personne) amène à penser que les agents mobiles vont devenir un outil essentiel dans le cadre du commerce électronique sur l'Internet. En effet, les utilisateurs de ces services Web désirent continuer leurs transactions quel que soit le lieu où ils se trouvent, et quelle que soit la machine qu'ils utilisent. Cela implique une forte hétérogénéité des machines disponibles pour exécuter une seule et même transaction commerciale (ordinateur de bureau, téléphone portable, SetTopBox de télévision interactive). Dans ce

cadre, l'agent mobile va jouer le rôle de l'utilisateur, pour rassembler les informations, les adapter au terminal auquel elles sont destinées, et les transmettre pour permettre de poursuivre l'exécution de l'application. Ci-après nous citons quelques exemples d'agents mobiles dans le commerce en ligne :

MAGNET : acronyme de (Mobile Agents for Networked Electronic Trading) est un système de négociation électronique en réseau, basé sur le framework Aglets. Le site de l'acheteur maintient une liste de fournisseurs potentiels avec leurs listes de produits. L'acheteur qui est intéressé par l'acquisition d'un produit crée un agent mobile, spécifie les critères pour l'acquisition du produit, et distribue l'agent mobile aux fournisseurs potentiels. L'agent mobile se rend à chaque site de fournisseur, cherche les catalogues de produits selon des critères de l'acheteur, et revient à l'acheteur avec la meilleure affaire qu'il trouve. Soit l'acheteur confirme la transaction et procède à la transaction monétaire, soit il annule la requête.

Supplier driven marketplace : cet autre modèle de e-commerce utilisant les agents est appelé 'Supplier driven marketplace'. Cette approche est particulièrement intéressante pour les produits ayant une courte durée de vie. Un fournisseur crée et distribue un agent mobile pour les acheteurs



● ● ● potentiels, en lui donnant une liste de sites à visiter. L'agent mobile porte avec lui les informations sur les stocks disponibles et le prix du produit.

2. La recherche d'information sur le Web

Avec le développement du réseau Internet, la plupart des fournisseurs de services ont développé leur interface Web. Cette interface leur permet de présenter leurs produits et de donner des informations (numéro de téléphone, plan d'accès etc.) sur la société. L'adoption de cette solution permet aux utilisateurs du réseau Internet d'avoir accès au monde à travers leurs ordinateurs à moindre coût (Internet est presque gratuit). Actuellement, le modèle « client/serveur », où les échanges se font par envoi de messages à travers le réseau, est le modèle le plus utilisé. Ce modèle possède l'inconvénient d'augmenter le trafic sur le réseau et exige une connexion permanente. Dans le but de diminuer les échanges intermédiaires, nous pouvons envisager d'utiliser la technologie d'agents mobiles. Ainsi l'utilisateur crée un agent dont le but est de visiter les sites et de lui retourner les résultats.

3. L'administration du réseau

L'utilisation de la technologie d'agents mobiles pour l'administration du réseau permet de réduire les coûts de communication sur un réseau

à faible débit. L'exécution d'agents mobiles sur les éléments du réseau permet de répartir des analyses traditionnellement effectuées sur la plate-forme d'administration, ce qui offre l'avantage de diminuer la charge de cette dernière. Les agents mobiles peuvent aussi être utilisés pour réaliser de façon décentralisée des tâches d'administration impliquant plusieurs éléments de réseau. L'exécution de tâches répétitives comme par exemple l'installation ou la mise à jour de logiciels, peut être déléguée à des agents mobiles.

4. Le calcul distribué

Un programme parallèle peut être vu comme un ensemble de tâches réparties sur le réseau ; ces tâches utilisent le réseau pour communiquer et pour échanger des données. L'un des problèmes principaux de l'implantation d'un algorithme sur une architecture distribuée est celui de la répartition de charges sur les machines disponibles. Le placement dynamique des tâches nécessite la migration de ces dernières entre les différentes machines du réseau. Cette migration devient opérationnelle grâce à l'utilisation de la technologie des agents mobiles. Dans une application distribuée, une tâche est représentée par un agent et la terminaison de celle-ci correspond à la fin de la vie de l'agent. Les échanges d'informations entre les tâches correspondent à la communication entre les agents. Quant au déplacement d'une tâche, il correspond à la migration d'un agent à travers le réseau, il s'agit de ce fait, de représenter une tâche par un agent mobile.

5. L'informatique nomade

Une application distribuée est constituée de sites variés inter-connectés par des réseaux de divers types. En particulier, certains sites peuvent être mobiles (ordinateur portable, smartphone, etc.) et peuvent être connectés au réseau Internet par l'intermédiaire d'une connexion sans fil à faible débit. Une application distribuée intégrant des terminaux mobiles et des réseaux sans fil ajoute des contraintes inhabituelles dans les systèmes distribués traditionnels. Ainsi, les ordinateurs portables sont sujets à de fréquentes déconnexions volontaires (à l'initiative de l'utilisateur) ou involontaires (suite à une interruption temporaire de communication dans un réseau sans fil par exemple). De plus, les réseaux sans fil sont caractérisés par un faible débit, un coût de communication élevé et une faible qualité de service.

L'utilisation de la technologie d'agents mobiles permet de surmonter les problèmes liés à la déconnexion des sites. En effet, un terminal mobile crée un agent mobile et lui demande d'agir pour son compte. L'agent créé, après sa migration, peut s'exécuter dans le système même si le site mobile créateur fonctionne en mode déconnecté. Une fois le site du client connecté, il va contacter l'agent mobile afin de lui demander de revenir sur son site d'origine. Cependant, l'exécution d'un environnement d'agents mobiles dans le contexte d'un système d'informatique nomade pose plusieurs problèmes. Le premier est relatif à l'apparition et la disparition dynamique des places pouvant accueillir les agents.

Cette situation se produit lorsqu'une place s'exécute sur un site mobile et que son moyen de communication est déconnecté. Un second problème se pose au niveau du nommage des agents mobiles du fait de la mobilité des sites. Celle-ci entraîne un changement d'adresse IP ou même un changement de nom de site. Dans ce contexte, se pose le problème de la localisation des agents mobiles du fait de la mobilisation des sites d'accueil.

6. Le code mobile et les cartes à puce

La carte à puce a été conçue comme un médiateur électronique de confiance entre l'utilisateur et les infrastructures matérielles qu'il utilise. Une carte à microprocesseur n'est pas autonome en terme d'énergie et ne possède pas d'horloge interne, l'ensemble lui étant fourni par le lecteur auquel elle est connectée. En tant que telle, la carte migre de système d'information en système d'information au gré du déplacement de son porteur. Lors de chaque nouvelle connexion, elle assure la continuité du service et la persistance des informations critiques exploitées, mais surtout elle sécurise le code qu'elle véhicule. De ce point de vue, une carte à puce est vue sous la forme d'un serveur. Les nouvelles cartes à puce sont devenues des supports mobiles d'application. Ainsi, un programme placé dans une carte est amené à migrer et à s'exécuter sur différents terminaux d'accueil pour représenter le porteur de la carte. En ce sens, tout code encarté est



- un code mobile. Par ailleurs, les dernières générations de cartes permettant de charger de nouveaux programmes tout au long de leur cycle de vie, peuvent être elles-mêmes vues comme des structures d'accueil pour code mobile. Des travaux montrent la réalisation d'une plate-forme d'accueil pour code mobile sur une carte à puce, il s'agit de la plate-forme CAMILLE.

Limites des agents mobiles

Les agents mobiles ne pourront pas remplacer intégralement le modèle classique du client/serveur lequel s'applique parfaitement dans un grand nombre de cas. Les agents mobiles ne peuvent donc pas être utilisés à grande échelle sans avoir résolu leurs problèmes inhérents au développement, à la sécurité et à la fiabilité.

Le développement Le domaine des agents mobiles se heurte à de fortes contraintes lors des phases de développement.

La première d'entre elles est qu'il existe à l'heure actuelle un grand nombre d'intéressés pour agents mobiles qui possèdent leurs propres défauts et qualités. Avec cette offre pléthorique, il est difficile de parler de standardisation et aucune ne semble en-

LA TECHNOLOGIE DES AGENTS MOBILES

core s'imposer. En sachant que les agents doivent s'adapter aux conditions de l'environnement, les développeurs sont confrontés à un éventail de choix trop large.

La sécurité Le problème de sécurité des agents mobiles n'est pas encore intégralement résolu. C'est d'ailleurs le principal argument avancé pour expliquer la faible utilisation de ce paradigme. En effet, les agents mobiles représentent un nouveau champ d'investigation pour le domaine de recherche en sécurité, d'une part dans la protection des sites vis-à-vis des agents malveillants et d'autre part dans la protection des agents vis-à-vis des sites malveillants. **Protection des sites** La protection des sites contre des attaques menées par des agents malveillants est un problème qui est aujourd'hui bien maîtrisé. En effet, plusieurs solutions permettent maintenant de se prémunir d'éventuelles attaques et voici les méthodes les plus connues :

- **Bac à Sable** : La technique du bac à sable consiste à exécuter un agent à l'intérieur d'un environnement restreint, en interdisant l'accès au système de fichiers par exemple.

Ainsi, un site peut exécuter un agent douteux dans le bac à sable sans trop se soucier des problèmes de sécurité. Cette approche peut facilement se mettre en place en utilisant des interpréteurs de code dont les possibilités sont limitées. Pour illustrer cette technique



nous pouvons citer les applets Java exécutées à l'intérieur d'un navigateur Web.

- **Signature de code** : La signature de code intervient lors de la création d'un agent, son créateur le signant numériquement afin qu'il puisse s'identifier durant ses déplacements.

Cette technique permet d'obtenir une authentification de haut niveau pour les sites. Les applets Java adoptent désormais ce mode de fonctionnement. Ainsi si une applet est signée, elle est considérée comme un code de confiance et peut accéder à toutes les fonctionnalités de Java. Elle sera placée dans un bac à sable dans le cas contraire.

- **Contrôle d'accès** : Pour améliorer les deux précédentes techniques, on met en place une politique de contrôle d'accès plus complexe. On peut la voir comme un raffinement d'une politique de bac à sable générale vers une politique spécifique à chaque application ou classe d'agents. En fonction des agents, le site pourra autoriser l'accès à un ensemble précis de fonctionnalités. Le contrôle d'accès permet de mixer les deux techniques en offrant aux agents signés plus de fonctionnalités qu'un simple bac à sable sans pour autant accéder à toutes les fonctionnalités.

Protection des agents À l'opposé de la protection des sites, la protection des agents contre des sites malveillants ne dispose pas de solution éprouvée et reste encore aujourd'hui un champ de recherche ouvert. Pour comprendre ce que risque un agent lors de son exécution sur un site malveillant, nous pouvons référencer les éléments transportés pouvant être cible d'attaque :

- **Le code** : ensemble des instructions composant la tâche de l'agent.

- **Les données statiques** : données ne changeant pas durant les déplacements (la signature par exemple)

- **Les données collectées** : ensemble des résultats obtenus au cours des déplacements réalisés par l'agent depuis son lancement.

- **L'état courant** : ensemble de données servant à l'exécution courante de l'agent.

La sécurité des agents mobiles consiste alors à garantir les critères de confidentialité et d'intégrité de l'ensemble de ces éléments. Du point de vue des données, il est évident qu'un agent ne souhaite pas divulguer des informations critiques à n'importe quel site. Par exemple, un site



- malveillant pourrait récupérer la signature d'un code et l'utiliser pour créer un nouvel agent afin de s'introduire dans des environnements auxquels il n'a normalement pas accès.
- Pour le code, un agent transporte un savoir-faire propre à son concepteur qui pourrait tomber aux mains de ses concurrents. Le problème le plus important empêchant l'adoption actuelle des agents mobiles est sans nul doute la sécurité. En effet, même si la protection des sites est quasiment assurée, celle des agents reste un réel problème qui n'a pas de solution définitive. Différentes études ont été menées pour permettre d'obtenir un niveau de sécurité satisfaisant

La fiabilité Le modèle d'exécution de l'agent mobile implique son interaction avec plusieurs sites ; ce qui expose l'agent à une éventualité de disparition à cause de la défaillance ou de la déconnexion soudaine et imprévue d'un site sur lequel il s'exécute. La disparition d'un agent entraîne un dysfonctionnement de l'application basée sur ce dernier. Dans les systèmes d'agents mobiles, une erreur qui se produit ne peut être prise en compte par son site d'origine notamment dans le cas d'une déconnexion. Une application distribuée sûre doit pouvoir continuer de fonctionner en cas de défaillance d'une partie du système. Pour certains types d'applications, il est essentiel que les environnements d'exécution d'agents mobiles offrent des mécanismes de tolérance aux pannes.

Conclusion

Il existe plusieurs raisons qui justifient le choix d'utilisation des agents mobiles. En effet, les agents mobiles :

- réduisent la charge des réseaux ;
- surmontent le temps de latence sur le réseau ;
- s'exécutent de manière asynchrone et autonome ;
- s'adaptent dynamiquement aux environnements d'exécution ;

Plusieurs articles de synthèse, construits sur l'avis de spécialistes du domaine de la mobilité, exposent la vision des futures directions que devraient suivre les agents mobiles. En résumé, ils viennent confirmer que les agents ne pourront pas être utilisés à grande échelle sans avoir résolu leurs problèmes de sécurité et qu'ils ne remplaceront pas intégralement le modèle classique du client/serveur qui s'applique parfaitement dans un grand nombre de cas. Ils viennent aussi confirmer que les agents mobiles, par leur capacité d'autonomie et d'adaptation, méritent d'être étudiés dans des environnements dynamiques décentralisés constitués par les topologies hybrides.

Tout cela nous permet de conclure que les agents mobiles ne peuvent pas totalement remplacer l'ancien modèle du client/serveur mais reste une technologie prometteuse qui présente de nombreux avantages plus particulièrement dans le domaine des systèmes distribués.



LES CONSEILS DE DZ-CERT



Attention aux risques liés à l'utilisation des logiciels piratés.

Le piratage de logiciels est une affaire sérieuse, en plus d'enfreindre la loi et les droits de propriété intellectuelle des créateurs de logiciels, vous pouvez mettre votre PC à risque de dommages et de menaces de sécurité.

Il existe plusieurs méthodes pour obtenir et utiliser des logiciels contrefaits. Ceux couramment utilisés sont : l'obtention et l'utilisation de « clés contrefaites de produit », l'obtention de programmes « Générateur de clé » et les utiliser pour créer des clés de produits, et l'obtention des « outils de craque » et de les utiliser pour contourner les mécanismes d'attribution et d'activation de licences.

Qu'est-ce que le piratage logiciel ?

Le piratage logiciel est l'utilisation, la copie ou la distribution non autorisée d'un logiciel soumis à des droits d'auteur. Il peut prendre différentes formes :

- **Copie illicite de logiciels achetés en toute légitimité (piratage par l'utilisateur final) ;**
- **Accès illégal à un logiciel protégé (craquage) ;**
- **Reproduction et/ou distribution de contrefaçons ou de logiciels non autorisés (généralement par Internet).**

Les logiciels font partie des œuvres protégées par le code de la propriété intellectuelle. Dans ce bulletin, nous allons focaliser sur les risques techniques liés à l'utilisation des logiciels piratés.

Risques techniques d'utilisation des logiciels piratés :

- 1.** Les logiciels piratés peuvent provoquer une panne de votre système. Ils entraînent une perte de temps. Vous pouvez perdre des données ou des fichiers irremplaçables. Vous pouvez même détruire votre ordinateur et tous vos logiciels.
- 2.** Lorsque vous utilisez la copie illicite d'un logiciel, vous ne pouvez pas bénéficier des mises à jour du produit. Or, les mises à jour sont indispensables pour assurer la sécurité des logiciels : chaque année, les éditeurs publient des dizaines de « correctifs sécurité » ou des « patches » améliorant le fonctionnement de leurs logiciels. Si, cependant, vous utilisez des logiciels contrefaits, vous ne pourrez pas incorporer ces correctifs et serez vulnérable face à d'éventuelles attaques.
- 3.** Les logiciels piratés sont souvent incomplets et manquent de documentation. De plus, certains logiciels téléchargeables sur Internet sont en fait des versions bêta (de test) qui ne comportent pas toutes les fonctionnalités. Les pièces manquantes peuvent être parfois fatales : les versions bêta étant destinées à l'essai, rien ne garantit qu'elles ne mettent pas en péril le bon fonctionnement de votre ordinateur.

4. Les logiciels piratés peuvent ne pas fonctionner correctement ou être entièrement défectueux, ce qui entraîne une surconsommation des ressources de votre entreprise et une augmentation des coûts informatiques.

5. En cas de problème, vous ne pouvez pas avoir recours au support technique de l'éditeur.

6. Les logiciels contrefaits peuvent être malveillants, par exemple contenir des chevaux de Troie, des virus ou des spywares qui s'infiltreront sur votre ordinateur et font usage de vos informations personnelles sans votre autorisation, qu'il s'agisse de vos numéros de carte de crédit ou de comptes bancaires, de vos mots de passe ou de vos carnets d'adresses. Les informations volées peuvent être exploitées immédiatement par des usurpateurs d'identité.

De même, un vendeur qui propose de violer la loi ne s'arrêtera probablement pas au piratage de logiciels. Toute donnée de carte de crédit ou information personnelle que vous communiquez peut être exploitée par des voleurs d'identité.

Comment se mettre à l'abri des logiciels piratés ?

Pour éviter d'être victime des pirates de logiciels, il est recommandé :

- D'acheter les logiciels uniquement auprès de sociétés connues.
- Lorsque vous faites des achats en ligne, assurez-vous que le site

Web est légitime. Sur la page d'accueil du site de vente, cliquez sur l'icône de verrouillage qui apparaît sur le cadre de votre fenêtre de navigateur et affichez le certificat de sécurité.

- Si un prix affiché semble trop attractif, méfiez-vous. Méfiez-vous également des prix extrêmement faibles et vérifiez l'authenticité du site.

Quelques données...

- Selon une étude menée par Business Software Alliance (BSA), consortium regroupant les principaux éditeurs mondiaux, 42 % des logiciels bureautiques ou professionnels installés dans le monde seraient des logiciels piratés, Cependant, dans 50% des pays étudiés, plus de 60% des logiciels étaient piratés.

- Une étude intitulée «The Dangerous World of Counterfeit and Pirated Software» de IDC (International Data Corporation (IDC)) a analysé 270 sites internet et réseaux «Peer-to-Peer», 108 téléchargements de logiciels et 155 CD/DVD à travers le monde et a interrogé 2077 consommateurs et 258 directeurs informatiques et chefs des services informatiques au Brésil, en Chine, en Allemagne, en Inde, au Mexique, en Pologne, en Russie, en Thaïlande, au Royaume-Uni et aux Etats-Unis. Il a pu être constaté que, parmi les logiciels contrefaits, 45% d'entre eux proviennent d'internet (sites internet, réseaux «Peer-to-Peer»), 78% de ces logiciels ont des logiciels espion et 36% d'entre eux contiennent un Cheval de Troie et des logiciels publicitaires. Ainsi que :

- 64% des interrogés ayant utilisé des logiciels contrefaits ont déjà fait face à des problèmes de sécurité.
- Dans 45% des cas, les logiciels contrefaits ont ralenti les PC et ont dû être désinstallés
- 48% des interrogés ont indiqué que leur plus grande préoccupation liée à l'utilisation de faux logiciels était la perte de données.
- 29% d'entre eux étaient principalement préoccupés par le vol d'identité.

Zoom Sur un proje

A magnifying glass with a black handle and silver rim is positioned over a document. The word 'proje' is written in a large, bold, pink font and is the central focus of the magnifying glass. The background shows a document with some numbers like '371' and '344' and a pen tip.

**Agents mobiles transactionnels
pour le E-commerce**

ZRGHACHE LYNDA

Maître de Recherche

Division réseaux et systèmes distribués

Introduction

Le concept d'agents mobiles transactionnels a été introduit comme un amalgame entre la technologie d'agents mobiles 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 identifié deux problèmes que l'agent seul ne peut résoudre. Le premier est lié à la fiabilité des agents. Des agents et des nœuds peuvent tomber en panne et donc, des mécanismes de détection de pannes et de recouvrement doivent être définis.

Le second problème pour lequel l'agent doit être assisté est lié à la validation atomique de la transaction. Une fois que 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 capacités de calcul, de stockage ou des ressources limitées. 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 d'une manière conforme à ce qui a été fait par les coordinateurs précédents.

Objectifs

Afin de répondre à toutes les difficultés avancées ci-dessus, des mécanismes de tolérance aux pannes et de validation atomique doivent être définis pour assister les agents mobiles dans leurs tâches de contrôle, de validation et de journalisation.

L'objectif du présent projet consiste à développer des agents mobiles pour l'exécution des transactions réparties afin de renforcer le modèle transactionnel avec plus d'autonomie et de fiabilité par l'adoption de nouvelles sémantiques transactionnelles comme la compensation et l'alternative. Les agents mobiles transactionnels doivent satisfaire les propriétés suivantes :

- Décider de manière autonome du prochain site à visiter en cas de panne ou d'indisponibilité de service.



- Migrer entre un ensemble d'ordinateurs et manipuler des objets sur chaque nœud.
- Valider la transaction seulement si la condition de validation telle que l'atomicité est remplie sinon abandonner la transaction.

Scénario

Lorsque l'agent exécute une requête sur un nœud, le résultat ne dépend pas uniquement de l'état actuel du nœud mais aussi des informations fournies par l'environnement extérieur. Par exemple, les réseaux de capteurs, qui sont maintenant intégrés partout, peuvent fournir des données (température, bruit, mouvement, débit d'air, lumière, etc.) pouvant influencer l'exécution de la requête. Une transaction peut être utilisée pour organiser les activités d'un client dans un futur proche. Considérons le scénario suivant : Arrivant au centre commercial, le client lance une transaction pour exprimer ses besoins d'achats (figure1) :

avant le déjeuner, le client désire trouver une place à l'ombre dans le parking du centre commercial (requête P pour Parking). Ensuite, il veut prendre un petit déjeuner dans une terrasse avec beaucoup de monde mais qui soit calme et à condition que la table soit protégée

du soleil et du vent (requête R pour Restaurant). Aujourd'hui c'est l'anniversaire de son amie, après le déjeuner, il veut lui acheter un album d'un groupe de musique parmi les dix meilleurs albums de la semaine mais seulement s'il apparaît sur la liste de lecture d'au moins un client qui a le même âge que son amie (requête M pour Musique). De plus, il veut lui offrir un T-shirt noir sur lequel le nom du groupe est imprimé (requête T pour T-shirt).

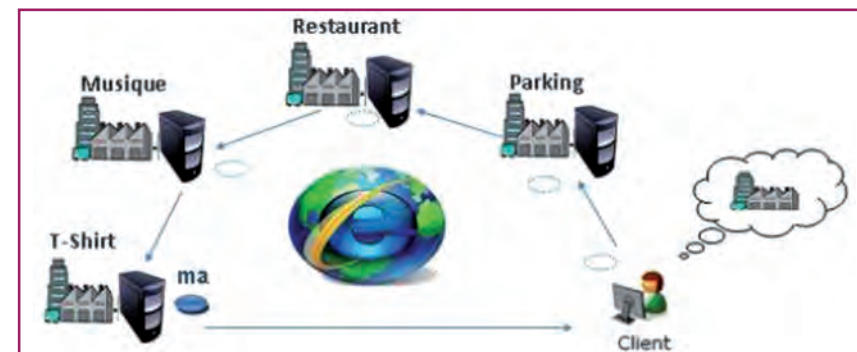


Figure 1

La transaction **P; R; M; T** est composée de quatre requêtes où chaque requête est exécutée sur un nœud qui offre le service demandé. Des capteurs sont utilisés pour assurer que les besoins spécifiés sont satisfaits (exemple, pas de bruit dans le restaurant). Des gens à proximité sont aussi consultés (accéder à leur liste de lecture). Les agents mobiles transactionnels sont bien adaptés à exécuter ce genre de scénario.

L'exploration de chemins et la réservation de services

Lorsque le client lance une transaction un agent mobile est créé. Cet agent migre vers plusieurs places pour exécuter les requêtes de la transaction.

Pour une requête donnée, plusieurs nœuds alternatifs, en mesure de satisfaire cette requête, sont identifiés. Le nombre de nœuds alternatifs dépend de la demande du client. Prenons la requête de réservation du restaurant dans l'exemple précédent, un client qui cherche un endroit pour manger est moins difficile à satisfaire qu'un client qui cherche un restaurant très fréquenté mais qui soit calme et avec une table protégée du soleil et du vent. Lorsque la demande du client est très précise et stricte elle risque de ne pas être satisfaite et par conséquent la transaction distribuée est annulée. Cependant, si la requête est plus large en termes de conditions de satisfaction, elle sera fort probablement satisfaite, ce qui augmente les chances de validation de la transaction globale. Pour cela, nous définissons pour chaque requête des niveaux de satisfaction. En définissant plusieurs niveaux de satisfaction pour chaque stade d'exécution le nombre de nœuds alternatifs augmente et la requête aura plus de chance d'être satisfaite. Dans notre exemple, si après avoir trouvé une place dans le parking dans le premier stade d'exécution **S0**, le client peut exiger un restaurant avec un menu qui ne dépasse pas 1500 DA. S'il est impossible de satisfaire cette demande, le client peut atténuer sa contrainte et accepter un menu inférieur à 2500 DA (une

requête alternative avec un niveau de satisfaction moins élevé). Il est important de définir les niveaux de satisfaction notamment dans un contexte transactionnel où les transactions distribuées peuvent être très longues (le nombre de stades d'exécution est très grand) et qui risquent d'être annulées à cause d'une requête non satisfaite et pour laquelle le client est prêt à faire des concessions.

Lorsque la place visitée est capable de satisfaire la requête de l'agent mobile, ce dernier effectue une pré-réservation du service où des ressources sont allouées à cette requête. Cette pré-réservation n'est validée qu'après la satisfaction de toutes les requêtes.

Cependant, si l'une des requêtes n'est pas satisfaite toute la transaction est annulée, cela veut dire que dans une transaction longue une pré-réservation peut être annulée après une longue durée d'allocation de ressources. Cette pénalisation des ressources qui ne sont pas validées à la fin peut être nuisible au fournisseur de service qui refuse de satisfaire d'autres requêtes demandant la même ressource. Pour contourner ce cas indésirable, nous supposons que le fournisseur de service garde le contrôle de ses ressources le plus longtemps possible, et ce, en allouant les ressources pour une période de temps limitée. Donc le fournisseur de service peut annuler la réservation (expiration du temps limite sans que la transaction soit décidée) et réallouer la ressource pour une autre requête. Le mécanisme d'allocation est flexible de telle sorte que le fournisseur de service peut réallouer les ressources à la requête annulée une nouvelle fois pour augmenter les chances de validation de la transaction si elle n'est pas encore décidée.

Un mécanisme de contrôle et de recouvrement

Afin d'assurer la progression de l'agent même en présence de pannes; nous définissons un mécanisme de contrôle créé et activé par l'agent mobile

La solution 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 est 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.

Les deux agents peuvent se rencontrer sur une place commune de l'itinéraire. Une place autorise un agent à exécuter une requête s'il s'agit de la première exécution où si l'agent est plus récent. Nous parlons **d'agent plus récent** s'il est

créé sur une place alternative suite à la suspicion d'une panne de l'agent contrôlé. La détection anticipée d'une double exécution permet de diminuer les réservations multiples des mêmes services pour une même transaction. Afin de distinguer l'agent le plus récent du moins récent nous utilisons un système de marquage qui permet d'associer un **tag** à l'agent mobile à chaque stade d'exécution. La figure 2 illustre les scénarios possibles dans le cas de la transaction **P; R;M; T**. Les fausses suspicions sont désignées par des croix en lignes pointillées alors que les vraies pannes sont désignées par des croix en lignes continues.

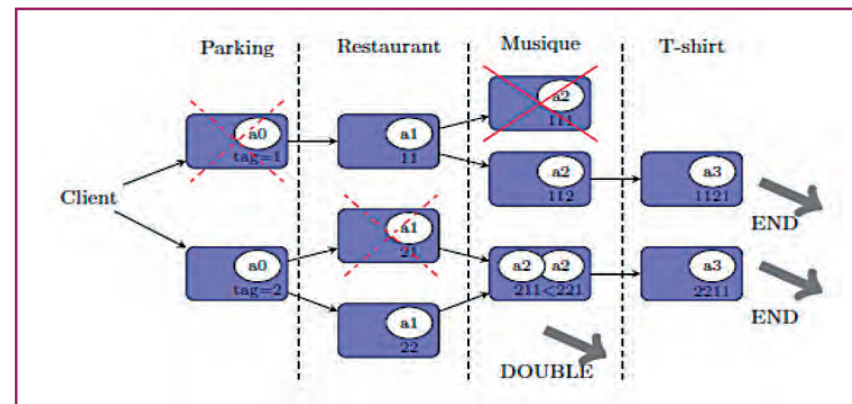


Figure 2

La validation atomique

Le protocole de validation atomique est exécuté par une entité que nous appelons gestionnaire de transaction (Transaction Manager) noté **TM**.

- Le gestionnaire de transaction est destiné à gérer l'exécution de la transaction depuis son lancement jusqu'à sa terminaison par une validation ou un abandon. Deux services sont fournis par le gestionnaire de transaction : La disponibilité de la source (Availability of the source, AS) et la validation atomique (Atomic Commitment, AC). Des copies sont définies pour assurer les tâches du TM en cas de panne. Une cohérence totale doit être maintenue entre les différentes copies du TM. Pour cela, chaque réplique doit être informée de toutes les actions effectuées par le TM afin de pouvoir reprendre l'exécution à partir d'un état cohérent, en cas de panne.

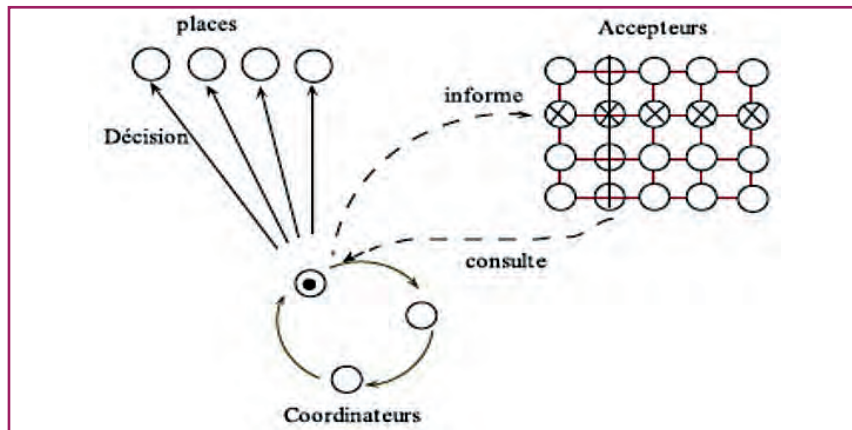


Figure 2

Nous avons utilisé un protocole de consensus comme brique de base dans l'implémentation du TM. Nous avons défini un service d'accord

appelé **AgreementService**. Ce service implémente un protocole de la famille paxos, le protocole Paxos-MIC, pour construire une séquence de décisions sur laquelle les deux services **AS** et **AC** sont implémentés. Ce protocole garantit la persistance de toutes les décisions prises. Cela permet d'obtenir un gestionnaire de transaction fiable qui supervise le comportement d'un agent mobile transactionnel dès sa création, grâce au service **AS** et jusqu'à sa validation grâce au service **AC**.

Collaborations

Dans le cadre de ce projet, nous avons travaillé avec :

Mr. Hurfin Michel, Maître de Conférences à l'INRIA (Institut national de recherche en informatique et en automatique) - Rennes - France.

Melle Moise Izabela, Docteur à l'IRISA (Institut de recherche en informatique et systèmes aléatoires) - Rennes - France.



FORMATION

Plusieurs sessions de formations sur Excel niveau I et niveau II au profit du personnel de soutien à la recherche du centre et du personnel de l'Office Algérien Interprofessionnel des Céréales (OAIC) ont eu lieu du mois de mars 2014 jusqu'au mois de septembre 2014. Au total, plus de soixante dix employés du centre ont été formés.

Soutenance de doctorat

Melle Zeghache Linda, attachée de recherche dans la Division Réseaux et Système distribués a soutenu sa thèse de doctorat intitulée :

« **Tolérance aux pannes dans les systèmes d'agents mobiles transactionnels** » **le dimanche 27 avril 2014 avec mention très honorable.**

RAPPORTS DE RECHERCHE INTERNES

([http : // www.cerist.dz/publications](http://www.cerist.dz/publications))

Setitra Insaf , Meziane Abdelkrim, A simple approach to distinguish between Maghrebi and Persian calligraphy in old manuscripts. Alger: CERIST, 2014-04-07. ISRN : CERIST-DSISM/RR--14-000000007--dz

<http://dl.cerist.dz/handle/CERIST/605>

Mohand Oussaid Linda , Ait-Sadoune Idir , Ait-Ameur Yamine , Ahmed Nacer Mohamed, modélisation formelle d'IHM multi-modales en sortie avec B Événementiel. Alger: CERIST, 2014-04-23. ISRN : CERIST-DSISM/RR--14-0000000012--dz

<http://dl.cerist.dz/handle/CERIST/651>



- ● ● Doudou Messaoud , M. Barcelo-Ordinas, Jose , Djenouri Djamel, Garcia-Vidal, Jorge, Badache Nadjib,Game ,Theoretical Approach for Energy-Delay Balancing in Distributed Duty-Cycled MAC Protocols of Wireless Networks. Alger: CERIST, 2014-04-24. ISRN : CERSIT-DTISI/RR--14-0000000013—dz
<http://dl.cerist.dz/handle/CERIST/652>
- Setitra Insaf, Larabi Slimane,Background subtraction algorithms with post processing A review.Alger: CERIST: 2014-04-28. ISRN: CERIST-DSISM/RR--14-0000000014—dz
<http://dl.cerist.dz/handle/CERIST/654>
- Benkhelifa Imane , Lamini Chakib , Azouz Hichem ,Moussaoui Samira,Prediction-based Localization for Mobile Wireless Sensor Networks. Alger: CERIST: 2014-06-05. ISRN : CERIST-DTISI/RR--14-0000000015--dz
<http://dl.cerist.dz/handle/CERIST/669>
- Keddari Djalila , Mellah Hakima , Benna Amel,Using tags Associated to Resources to Build Ontologies .Alger: CERIST: 2014-06-18. ISRN : CERIST-DSISM/RR--14-0000000016—dz
<http://dl.cerist.dz/handle/CERIST/672>
- Khider Hadjer , Benna Amel, Collaboration sociale dans un échange inter-intra entreprises. Alger: CERIST: 2014-06-22. ISRN: CERIST-DSISM/RR--14-0000000019—dz
<http://dl.cerist.dz/handle/CERIST/677>
- Djenouri Youcef, Gheraibai, Youcef, Mehdi, Malika, Bendjoudi Ahcène, Nouali-Taboudjemat Nadia, An Efficient Measure for Evaluating Association Rules.Alger: CERIST: 2014-06-24. ISRN: CERIST/DTISI/RR--14-0000000018—dz
<http://dl.cerist.dz/handle/CERIST/676>
- Djenouri Youcef, Bendjoudi Ahcène, Mehdi Malika, Nouali-Taboudjemat Nadia, Habbas Zineb, Parallel Association Rules Mining Using GPUs and Bees Behaviors.Alger: CERIST: 2014-06-24. ISRN: CERIST/DTISI/RR--14-0000000017—dz
<http://dl.cerist.dz/handle/CERIST/675>

CERIST

Bases de données documentaires

Accessibles sur : www.cerist.dz



Le CERIST permet l'accès à une documentation électronique nationale et internationale couvrant tous les domaines scientifiques et techniques grâce au Système National de la Documentation en Ligne (SNDL). Ce système concerne les chercheurs, les enseignants chercheurs et les étudiants.

De plus amples informations sont disponibles sur le site : www.sndl.cerist.dz

CERISTNEWS

The screenshot shows the SndL website interface. At the top, there is a navigation menu with links: A PROPOS, ACTUALITES, BASES DE DONNEES, PORTAILS, FORMATIONS, CONTACTS, and a Connexion button. Below the menu, there are four category buttons: SCIENCES & TECHNIQUES, SCIENCES DE LA VIE & DE LA TERRE, SCIENCES HUMAINES & SOCIALES, and PLURIDISCIPLINAIRES, each with a 'Plus' button. A large image of laboratory glassware is displayed. Below the image, there is a search prompt: 'Pour effectuer une recherche, CLIQUEZ ICI'. At the bottom, there are three columns of text: 'A Propos Du SNDL ?' with a keyboard and globe image, 'Charte SNDL' with a list of resource categories, and 'Actualités et Nouveautés' with a list of news items.

Directeur de publication

Pr. BADACHE Nadjib

Dossier : **LE DOCUMENT MULTIMEDIA**

Réalisé Par : **Dr. Zeghache Lynda**

Maître de Recherche – CERIST –

Division Réseaux et Système distribués

Rubrique : **Les Conseils de DZ - CERT**

L'ÉQUIPE DZ-CERT

Rubrique : **Zoom sur un Projet**

- CERIST -

Comité de communication et de rédaction

BEBBOUCHI Dalila

BENNADJI Khedidja

DJETTEN Fatiha

Photographies

ALIMIHOUB Dahmane

Réalisation graphique

BOUKEZOULA Mohamed Amine

Publié par **le CERIST**

5, rue des 3 Frères Aissou. Ben Aknoun. BP 143, 16030 - Alger

Tél : +213 (21) 91 62 05 – 08 / Fax : +213 (21) 91 21 26

E - mail : vrr@mail.cerist.dz

www.cerist.dz

Impression

ANEP

ISSN : 2170-0656 / DÉPÔT LÉGAL : 2690-201



Le Bulletin CERISTNEWS

CENTRE DE RECHERCHE SUR L'INFORMATION SCIENTIFIQUE ET TECHNIQUE - CERIST
5, Rue des Trois Frères Aissou, Ben - Aknoun - BP 143. 16030 - Alger
Tél : +213 (21) 91 62 05 - 08 / Fax : +213 (21) 91 21 26

www.cerist.dz