

Les agents mobiles – Une introduction

Nicolas ESPOSITO

DASSAULT SYSTÈMES – Recherche et nouvelles technologies
9 quai Marcel DASSAULT – 92150 SURESNES – FRANCE
Tél. +33 (0)1 55 49 80 01 – Fax +33 (0)1 40 99 43 90

Sujet de recherche depuis plusieurs années, les agents mobiles ne sont toujours pas arrivés dans notre quotidien. Nous verrons que l'approche est très puissante, mais qu'elle est aussi très exigeante.

1 Exemple en client/serveur

Grâce à la généralisation d'Internet, tout le monde connaît le modèle client/serveur. Vous demandez une page HTML, un serveur vous l'envoie (voir figure 1-a). Le serveur peut aussi *calculer* votre page. Par exemple, il peut exécuter un script CGI qui consulte une base de données de chaussures de sport ; puis, il vous envoie la liste des modèles qui sont disponibles dans votre taille (voir figure 1-b).

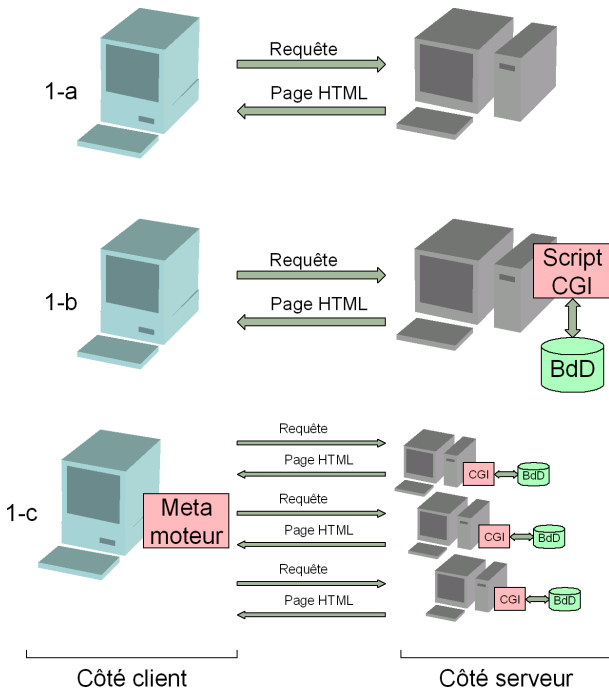


FIG. 1 : Modèles client/serveur

Pour comparer les prix de plusieurs magasins, vous devez trouver les sites, vous rendre sur chacun d'eux, effectuer vos recherches et noter ce que vous y trouvez. Une telle démarche prend un temps que beaucoup jugeront trop élevé. Une solution consiste à ajouter un intermédiaire : un meta moteur de recherche qui gère le commerce électronique (voir figure 1-c). Il va interroger

pour vous chaque magasin. Il en connaît les langages d'interrogation et les critères proposés. Après quelques instants de recherche, vous aurez la liste des dix modèles les moins chers.

Cette solution présente encore d'importantes contraintes. La machine cliente doit être connectée pendant toute la recherche, elle se charge de tous les calculs et le trafic sur le réseau est maximal : la requête est envoyée à chaque fois et les serveurs renvoient tous leur page complète de résultats.

Si l'on déplace le meta moteur sur un site Web, le client se trouve alors dans une situation beaucoup plus intéressante : la requête est envoyée une seule fois et un nombre restreint de résultats est renvoyé. Mais finalement, nous ne faisons que déplacer le problème. Le serveur du site qui fera les recherches rencontrera les mêmes contraintes. Un autre type d'architecture est donc à envisager. Nous allons voir comment les agents mobiles peuvent répondre à ce type de problème.

2 Exemples à base d'agents mobiles

Dans un modèle à agents mobiles, le client n'exécute plus toutes les tâches, il délègue le travail. On considère ici l'agent mobile comme un ensemble code/données autonome et capable de se déplacer entre les différents environnements d'exécution des machines hôtes d'un réseau (voir figure 2).

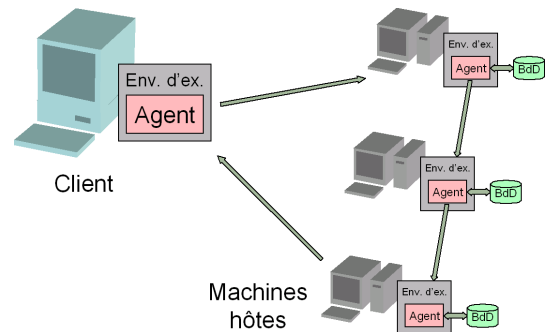


FIG. 2 : Modèle à agent mobile

Tout ce qui est fait en mode client/serveur pourra être fait avec des agents mobiles. Une partie des applications ne nécessiteront pas ce nouveau modèle, mais

certaines se révéleront plus efficaces en l'utilisant. Voici quelques exemples :

- la collecte d'informations avec filtrage (personnalisation de recherche) ;
- la surveillance (si tel évènement se produit sur le réseau, alors l'agent prévient le client) ;
- la distribution d'informations (les données ne partent qu'une seule fois) ;
- la négociation (l'agent part avec une offre du client et revient avec le meilleur prix obtenu) ;
- le calcul parallèle (l'agent peut se dupliquer sur différents serveurs avec les différentes parties d'un calcul à effectuer).

3 L'agent

Nous définirons plus tard la notion d'environnement d'exécution. Considérons pour l'instant qu'il s'agit de l'environnement qui permet à un agent d'être accueilli sur un hôte.

Un agent mobile peut être considéré comme une entité logique disposant des facultés suivantes :

Comportement. Un agent est pourvu d'un programme qui lui dicte quelles tâches il doit accomplir. On dit qu'il agit par *délégation* pour le client. Un agent peut ainsi venir avec une compétence particulière sur un hôte, par exemple avec une feuille de style qu'il appliquera au document à envoyer.

Autonomie. Un agent agit indépendamment du client. Il décide lui-même là où il va et ce qu'il doit y faire, en fonction du comportement qui lui a été donné. Cela implique que l'on ne peut pas toujours prévoir l'itinéraire des agents et c'est pour cela qu'on utilise des *proxies* pour faire abstraction de leur localisation.

Mémoire. Un agent dispose d'un potentiel de mémorisation permettant par exemple de recollecter des informations.

Communication. Un agent doit pouvoir interagir avec les différents environnements (le client et les hôtes) et les autres agents (locaux ou distants).

Sécurité. Un agent doit se protéger des attaques extérieures : par rapport aux autres agents, à ses environnements d'exécution et pendant son déplacement sur le réseau. Son intégrité et la confidentialité des données qu'il transporte doivent être assurés. Par exemple, on ne doit pas pouvoir lui *voler* un numéro de carte de crédit ou modifier son programme. On ne doit pas non plus pouvoir pratiquer la mutation d'un agent à son insu. C'est ainsi que l'on est amené à crypter les agents.

Malgré de sérieuses précautions à prendre en terme de sécurité, les caractéristiques des agents mobiles laissent apparaître un potentiel à répondre efficacement à certains types de problèmes.

Tout d'abord, les agents mobiles vont permettre de spécialiser les serveurs, pour que ces derniers répondent à des besoins qu'ils ne connaissent pas forcément à l'avance. Nous allons pouvoir les adapter afin qu'ils fournissent des services spécifiques. Une fois ces services installés, on peut aussi envisager qu'ils restent sur les serveurs.

Ensuite, les agents mobiles vont limiter l'utilisation du réseau au strict minimum. En général, on aura simplement deux échanges : l'envoi d'un agent et la réception des résultats. Les temps de connexion au réseau seront plus courts et on peut très bien imaginer des machines nomades – connectées de façon intermittente – qui seront déconnectées pendant toute la durée du travail de l'agent. Cette approche nous permettra aussi d'alléger la charge de calcul pour le client. Puisque l'agent agira par délégation pour le client, les calculs seront distribués sur les différents serveurs hôtes.

En ce qui concerne les performances, des études montrent que dans certains types d'applications et à partir d'un certain seuil (puisque'il faut tenir compte de la taille de l'agent en lui même), les agents mobiles deviennent beaucoup plus intéressants que le modèle client/serveur [1].

4 L'environnement d'exécution

Toute machine hôte qui accueille un agent doit disposer d'un environnement d'exécution qui permette aux agents de s'exécuter et d'accéder aux différents services de l'hôte, par exemple à une base de données ou à un algorithme de cryptage.

L'environnement d'exécution pose le problème de sa cohérence sur le réseau. Il nécessite une grande homogénéité des configurations logicielles sur chacun des hôtes. Même si un standard existait, il faudrait encore gérer les différences entre les versions. Dans le domaine de l'interopérabilité entre les agents, il existe trois standards : Mobile Agent System Interoperability Facility (MASIF) de l'OMG (l'organisme qui gère la norme CORBA), Knowledge Query and Manipulation Language (KQML) et Foundation of Intelligent Physical Agents (FIPA). Mais les plates-formes à agents mobiles qui se conforment à ces normes sont rares.

Si l'agent est un exécutable (par exemple du langage C compilé), il faudra en plus gérer l'homogénéité des plates-formes matérielles et des systèmes d'exploitation. Java et les langages interprétés ne souffrent évidemment pas de cette contrainte.

L'environnement d'exécution doit aussi se protéger des attaques, par exemple d'agents virus. Il nous faut donc un mécanisme d'authentification des agents pour gérer les droits qu'ils auront sur l'environnement. Cela permettra notamment à un hôte d'interdire l'accès à certaines informations pour tous les agents n'appartenant pas à un groupe autorisé. On pourra aussi gérer la répartition du temps de calcul entre les agents en leur affectant des priorités.

5 Implémentations

Chaque implémentation des agents mobiles présente ses spécificités, ses avantages et ses inconvénients. Nous allons ici en citer deux, parmi les plus connus.

Telescript [2] de General Magic est connu pour être la première implémentation commerciale des agents mobiles. Il s'agit d'un langage interprété que General Magic a remplacé par Odyssey (basé sur Java). On y trouve, en plus du concept d'agent, les notions suivantes : *places* (lieux virtuels), *travel* (pour aller de *place* en *place*), *meetings* (quand deux agents se rencontrent sur une *place*), *connections* (quand deux agents ne sont pas sur la même *place* et qu'ils communiquent), *authorities* (pour définir le niveau d'autorisation) et *permits* (possibilités d'exécuter des instructions ou d'accéder à des ressources).

Le plus connu des projets du monde Java nous vient de chez IBM Japon, il s'agit d'Aglets [3]. Ce nom a été créé à partir des mots *Agent* et *Applet*, ce qui exprime assez clairement ce qu'est Aglet. Pour créer un agent mobile avec Aglets, il suffit d'écrire une classe qui étende la classe *Aglet*. Cette classe dispose de méthodes à définir telles que *onCreation*, *onArrival*, *onDisposing*, etc. On a donc affaire à des objets Java (sérialisables et clonables) qui ont en plus la faculté de se déplacer.

Conclusion

Nous avons vu qu'une standardisation des agents mobiles permettrait de garantir une cer-

taine hétérogénéité des environnements d'exécution. Mais il faudrait alors une norme si vaste que les implémentations ne pourraient pas la couvrir. Il semble donc que nous soyons contraints à travailler avec des implémentations spécialisées qui répondent à des besoins bien particuliers. Il est encore trop tôt pour programmer son propre agent afin qu'il aille travailler pour nous dans le monde entier.

Néanmoins, on notera le très bon positionnement de Java comme langage de développement. Son interprétation dans une machine virtuelle résoud le problème de l'hétérogénéité matérielle et du système d'exploitation. La sérialisation qui permet de sauver l'état d'un objet, par exemple avant son envoi sur le réseau, la possibilité de clonage et les dispositifs de sécurité renforcent cette position.

Références

- [1] ISMAIL (Leila) et HAGIMONT (Daniel), « Spécialisation de serveurs par des agents mobiles », *Colloque International sur les Nouvelles Technologies de l'Information (NOTERE '98)*, Montréal, Canada, novembre 1998, <http://sirac.inrialpes.fr/~hagimont/papers/98-notere-PUB.ps.gz>.
- [2] James E. WHITE, « Mobile Agents », *Software Agents*, AAAI Press – MIT Press, 1997.
- [3] IBM Aglets Software Development Kit – Home Page, <http://www.tr1.ibm.co.jp/aglets/>.