JCS'99
Rennes, 8 juin - 11 juin 1999
Journée des Jeunes Chercheurs en Système

Amélioration du niveau de service par la distribution adaptative d'applications dans un environnement mobile

F. Le Mouël

IRISA / INRIA
Campus Universitaire de Beaulieu
35042 Rennes, Cedex, France

e-mail : flemouel@irisa.fr

Résumé

Le développement grandissant de l'informatique nomade est le résultat de percées technologiques dans plusieurs domaines : les systèmes d'informations en ligne, les réseaux de communication sans fil, l'informatique mobile. Cependant, les déplacements géographiques ainsi que les contraintes matérielles induisent une variation importante des ressources (puissance CPU, capacité mémoire, bande passante réseau, énergie, ...) qui doit être prise en compte pour conserver, voire améliorer, le niveau de service rendu à l'utilisateur. Nous proposons une solution, basée sur une distribution adaptative des applications, pour améliorer ce niveau de service rendu.

1 Introduction

Due à une informatisation sans cesse croissante du travail, l'informatique nomade se développe et prend de plus en plus d'ampleur. Ce développement est le résultat de percées technologiques dans plusieurs domaines : les systèmes d'informations en ligne permettent aujourd'hui d'avoir accès de plus en plus aisément à une masse d'informations en perpétuel renouvellement; les réseaux de communication sans fil (GSM, satellites), associés au développement de l'informatique mobile (notebook, PDA), permettent à l'utilisateur de se déplacer en gardant son propre environnement, offrant ainsi constamment un accès aux sources d'informations. Ces avancées apportent de nouveaux défis qui dépassent le cadre traditionnel des systèmes distribués [9]. En effet, les déplacements géographiques ainsi que les contraintes matérielles induisent une variation importante des ressources (puissance CPU, capacité mémoire, bande passante réseau, énergie, ...) qu'il doit être nécessaire de prendre en compte pour conserver, voire améliorer, le niveau de service rendu à l'utilisateur.

Tout d'abord, nous définissons la notion de niveau de service rendu dans le cadre de la mobilité (§2), et présentons brièvement comment les travaux existants satisfont ce niveau de service (§3). Nous proposons une solution, basée sur une distribution adaptative des applications, pour améliorer ce niveau de service rendu (§4). Enfin, la dernière partie conclue, examine les problèmes soulevés et indique les futurs axes de recherche.

2 Définition du niveau de service

Pour répondre aux contraintes inhérentes à la mobilité, il est nécessaire de définir la notion de niveau de service rendu à l'utilisateur dans ce contexte [1]. Ce niveau de service peut être défini comme étant composé de deux parties, la partie qualitative :

- qualité de l'interactivité utilisateur (temps de réponse, transparence des connexions, déconnexions, ...),
- qualité des données (cohérence, pertinence, dégradation, distillation, transformations, ...),
- disponibilité des ressources (puissance CPU, capacité mémoire, bande passante réseau, énergie, ...) et des services (services de bas niveau : surveillance des ressources, détection de changement d'environnement, ... ; services de haut niveau : météo, trafic routier, ...),
- sécurité des services (enregistrement utilisateur, authentification, protocoles sécurisés, ...),

et la partie quantitative : quantité des données (formats de différents types selon le moyen de consultation (ordinateur, téléphone, fax, ...)).

3 Approches existantes

De nombreux travaux ont été menés pour adapter les environnements informatiques aux contraintes de la mobilité. Ces approches s'étendent du niveau système, en masquant la mobilité à l'utilisateur, jusqu'au niveau applicatif, où l'application met en œuvre l'adaptation à la mobilité.

Les approches masquant la mobilité à l'utilisateur se placent au niveau système ou réseau et consistent en une émulation des protocoles des systèmes statiques. Par exemple, Mobile-IP [7] permet à une station mobile de garder la même adresse IP malgré ses déplacements. Cette approche a aussi été utilisée dans le système Coda [5] en permettant aux utilisateurs de fonctionner en mode déconnecté, masquant les déconnexions au moyen de techniques de cache. Ces approches permettent de ne faire aucune modification des applications existantes, mais possèdent néanmoins certaines limitations liées au fait qu'un masquage complet, ne prenant pas en compte le comportement de l'application, ne peut être effectué.

Les approches intégrant l'adaptation à l'application permettent de prendre en compte des informations liées à l'environnement et ainsi d'opérer une adaptation de leur comportement d'exécution. Par exemple, Odyssey [6], une plateforme logicielle d'adaptation, surveille les ressources, détecte les changements, et envoie des notifications aux applications intéressées, la mise en œuvre de l'adaptation à la mobilité étant implémentée dans l'application. Les adaptations dépendent aussi de l'application et peuvent agir sur les données considérées : dégradation, distillation des images [3], nombre d'images par seconde dans une vidéo [4], ...

Ces différentes approches agissent toutes sur un critère unique du niveau de service, et nous nous proposons d'améliorer globalement le niveau de service par une distribution des applications.

4 Distribution adaptative d'applications

Dans ces approches applicatives, l'adaptation à la mobilité se fait *"à la baisse"*, c'est à dire que le comportement de l'application s'adapte aux variations par une *dégradation* du niveau de service. Ces adaptations n'essaient pas de tirer parti de la mobilité, ni des ressources et services des stations et du réseau sur lesquels les connexions sont effectuées. En effet, un utilisateur mobile est surtout intéressé par les informations et les services correspondants à sa situation actuelle, locale, que nous définissons comme le contexte. Parmi ces informations contextuelles, les informations liées à l'état du réseau (nombre de stations, charges, vitesses des liens), ainsi que les informations concernant les services disponibles, peuvent permettre de lancer au mieux l'application considérée et cela de manière distribuée. Ensuite, lorsque l'utilisateur change d'environnement, les informations contextuelles changent et permettent d'adapter la distribution de manière optimale, en terme de niveau de service, au nouvel environnement.

Les techniques traditionnelles de distribution permettent d'améliorer chacun des points du niveau de service :

- elle influent sur la qualité de l'interactivité utilisateur :
  - soit en améliorant le temps de réponse,
  - soit en évitant la diminution du temps de réponse lorsque l'on se connecte à un environnement moins puissant que le précédent;
- elle influent sur la qualité et la quantité des données :
  - soit en améliorant la qualité en mettant à profit la mobilité, le contenu évoluant par exemple selon la localisation,
  - soit en évitant la dégradation à l'aide d'un environnement disposant de plusieurs stations.

Des techniques particulières de distribution permettent aussi :

- d'améliorer la disponibilité si les éléments distribués ne sont pas dépendants, les différentes tâches pouvant être effectuées en parallèle,
- d'augmenter la sécurité, d'une part vis à vis des environnements surveillant particulièrement leurs ressources, en équirépartissant cette utilisation des ressources, et d'autre part, en offrant aucune sémantique au niveau d'une entité distribuée [2] contre d'éventuelles attaques du code par le système ou par d'autres utilisateurs [8].

La distribution est donc un moyen pour l'application de fonctionner dans son mode normal, là où les variations de ressources l'auraient forcée à travailler en mode dégradé (cf figure 1a), ou permet à l'application de mieux satisfaire le niveau de service même si elle fonctionne en mode dégradé (cf figure 1b). Cette distribution permet aussi d'améliorer le fonctionnement normal lorsque les ressources le permettent (cf figure 1c).

5 Conclusion, problèmes soulevés et travaux futurs

Nous avons présenté dans ce papier comment une distribution adaptative des applications peut permettre de surmonter certaines contraintes liées à la mobilité, et d'améliorer le niveau de service rendu à l'utilisateur.

Cette distribution soulève néanmoins différents problèmes. Le traitement de l'adaptation à la mobilité au niveau de l'application amène des problèmes de reconfiguration d'applica-

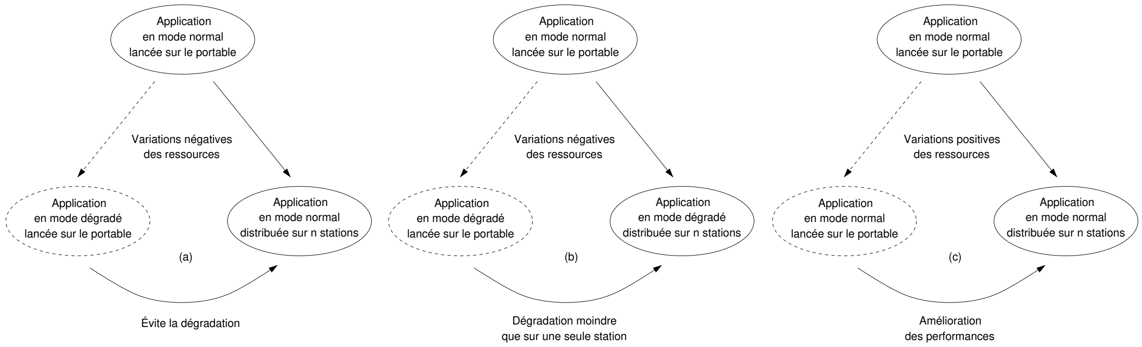

FIG. 1 – *Amélioration via la distribution*

tions. La distribution considérée peut être faite de manière statique ou dynamique soit à la conception, soit aux changements d'environnement, ou encore pendant l'exécution. De plus, certaines techniques particulières de distribution (décorrélation des éléments composants une application) doivent être mises en place pour prendre en compte la mobilité.

Contrairement aux approches citées en §3, nous envisageons d'intégrer ces fonctionnalités dans une approche middleware, permettant ainsi un masquage partiel de la mobilité, déchargeant l'application d'une partie de l'adaptation à la mobilité, mais n'empêchant pas l'utilisateur, au travers de préférences, de spécifier ses attentes en terme de niveau de service. Ce masquage partiel se retrouvera en particulier dans la distribution qui se fera de manière transparente mais restera sujette aux préférences de l'utilisateur.

Bibliographie

1. Davies (N.), Coulson (G.) et Blair (G.). – *Supporting Quality of Service in Heterogeneous Networks: From ATM to GSM.* – Rapport technique n. MPG-93-26, Computing Department, Lancaster University, 1993.
2. Fabre (J.C.), Deswarte (Y.) et Blain (L.). – *Tolérance aux fautes et sécurité par Fragmentation-Redondance-Dissémination.* – Rapport technique n. 95130, Laboratoire d'Analyse et d'Architecture des Systèmes, avril 1995.
3. Fox (A.), Gribble (S.D.), Chawathe (Y.) et Brewer (E.A.). – Adapting to Network and Client Variation Using Infrastructural Proxies: Lessons and Perspectives. *IEEE Personal Communications (invited submission)*, vol. 5, n. 4, août 1998, pp. 10–19.
4. Inouye (J.), Cen (S.), Pu (C.) et Walpole (J.). – System Support for Mobile Multimedia Applications. *In : Proc. of the 7th International Workshop on Network and Operating Systems Support for Digital Audio and Video*, pp. 143–154. – mai 1997.
5. Kistler (J.J.) et Satyanarayanan (M.). – Disconnected Operation in the Coda File System. *ACM Transactions On Computer Systems*, vol. 10, n. 1, février 1992, pp. 3–25.
6. Noble (B.D.), Satyanarayanan (M.), Tilton (J.E.), Flinn (J.) et Walker (K.R.). – Agile Application-Aware Adaptation for Mobility. *In : Proc. of the 16th ACM Symposium on Operating Systems Principles.* – octobre 1997.
7. Perkins (C.E.). – IP Mobility Support. – Request For Comments (RFC) 2002, octobre 1996.
8. Sander (T.) et Tschudin (C.F.). – Protecting Mobile Agents Against Malicious Hosts. *In : Mobile Agent Security*, éd. par Vigna (G.), pp. 44–60. – Springer-Verlag, 1998.
9. Satyanarayanan (M.). – Fundamental Challenges in Mobile Computing. *In : Proc. of the 5th Annual ACM Symposium on Principles of Distributed Computing.* – mai 1996.