

SERVICES DE COMMUNICATION RÉPARTIS DANS LE SYSTÈME MASIX

Julien Simon
Julien.Simon@masi.ibp.fr

Franck Mével
Franck.Mevel@masi.ibp.fr

Laboratoire MASI, Institut Blaise Pascal
Université Pierre et Marie Curie
Paris VI

1 INTRODUCTION

Depuis quelques années, la conception de systèmes d'exploitation répartis s'oriente vers des approches à base de micro-noyau. En effet, les micro-noyaux offrent de nombreux avantages par rapport aux systèmes monolithiques, comme la portabilité, la modularité et la dynamicité.

Cependant, les systèmes basés sur un micro-noyau posent de nouveaux problèmes, liés notamment à la décentralisation des informations, qui entraîne une augmentation des communications entre les différentes composantes du système. Afin de résoudre ces problèmes de manière efficace, le système d'exploitation doit disposer de services de communication répartis adaptés.

Cet article décrit cet aspect du système réparti Masix [Card *et al.* 1993], basé sur le micro-noyau Mach [Accetta *et al.* 1986], en cours de développement au laboratoire MASI. Son objectif principal est l'exécution simultanée de plusieurs environnements, afin d'utiliser parallèlement sur une même station de travail des applications Unix, DOS, OS/2, Win32, ...

Masix permet également de mettre en commun les ressources d'un réseau (processeurs, mémoire, périphériques, fichiers), indépendamment des différents environnements qui peuvent s'exécuter sur chaque station.

Nous présentons tout d'abord la structure générale du système Masix puis nous étudions ses services de communication.

2 STRUCTURE DE MASIX

Masix est construit selon le modèle multi-serveurs [Stevenson et Julin 1995], qui lui confère une grande modularité. En effet, chaque serveur, qui offre des fonctionnalités orthogonales, peut être écrit et testé indépendamment des autres. De plus, il est possible d'ajouter ou de retirer dynamiquement un serveur au système. Masix est composé de deux couches, elles-mêmes formées de plusieurs serveurs :

- les environnements, qui implémentent les sémantiques des systèmes d'exploitation traditionnels : Unix, DOS, OS/2, Win32, ... ;
- en dessous, le système d'accueil générique réparti, qui factorise les services répartis dont ils ont besoin, et notamment les services de communication sous la forme d'un serveur réseau.

La modularité du système Masix pose certains problèmes, comme la gestion de l'état global du système. En effet, les informations qui le composent sont réparties entre le micro-noyau et les différents serveurs. De ce fait, de nombreuses communications ont lieu, d'une part entre les différents serveurs s'exécutant sur un noeud, et d'autre part entre les différents noeuds du réseau. Il est donc important de disposer de mécanismes adaptés afin de minimiser leur surcoût.

Le serveur réseau offre des services de communication susceptibles de satisfaire les besoins de toutes les composantes du système Masix, c'est à dire des serveurs du système d'accueil et des serveurs qui composent les différents environnements.

2.1 Transparence des communications

Afin de concevoir le système Masix de manière cohérente et d'en faciliter le développement, il est préférable que les communications inter-processus distantes s'effectuent selon les mêmes sémantiques que les communications locales offertes par Mach.

En effet, deux tâches qui souhaitent communiquer ne doivent en aucun cas préjuger de leur localisation respective, car certains mécanismes de Masix, comme la migration [Mével et Simon 1995], provoquent le déplacement d'une tâche d'un noeud du réseau vers un autre noeud. Par ailleurs, plusieurs occurrences d'un même serveur peuvent coexister sur le réseau, afin d'assurer une meilleure tolérance aux fautes. Dans ce cas, une tâche ne peut pas savoir a priori laquelle d'entre elles va traiter sa requête.

2.2 Support multi-protocoles

Le système Masix est un système multi-environnements. Par conséquent, le serveur réseau gère les protocoles propres à chaque environnement (TCP/IP, IPX, ...). De plus, il connaît à tout moment l'état des communications initiées par les applications qui utilisent cet environnement.

En effet, la terminaison d'un environnement peut nécessiter la fermeture de toutes les connexions gérées par ses protocoles réseau. Il est donc important que tous les protocoles utilisés par les environnements soient connus du serveur réseau. Pour ce faire, celui-ci possède une interface d'enregistrement et de désenregistrement dynamiques des protocoles [Tschudin 1991].

2.3 Fiabilité des communications

Un système réparti s'exécutant sur un réseau local est à la merci des défaillances des noeuds et des liens de communications. Ce problème se pose avec d'autant plus d'acuité dans Masix que celui-ci est composé de nombreux serveurs, qui peuvent chacun tomber en panne. Par conséquent, un serveur peut être répliqué afin de garantir la continuité de son service.

Masix dispose donc de primitives de communication de groupe fiables [Schiper et Sandoz 1993, Hadzilacos et Toueg 1994] permettant d'une part de détecter les défaillances puis d'y remédier, et d'autre part de préserver la cohérence des données répliquées.

2.4 Sécurité des communications

Dans Masix, les problèmes traditionnels de sécurité des communications réseau prennent de l'ampleur. En effet, les serveurs échangent par le réseau des informations privées, qui ont trait à l'état global du système. Si ces informations sont interceptées ou altérées, la sécurité et l'intégrité du système peuvent s'en trouver gravement compromises. C'est pourquoi le serveur réseau met à la disposition des environnements des services de chiffrement.

De plus, grâce à son interface d'enregistrement, ce serveur contrôle l'accès aux périphériques réseau, puisqu'une application ne peut communiquer qu'au travers des protocoles enregistrés auprès du serveur. Ceci empêche une application mal intentionnée d'utiliser son propre protocole et d'accéder ainsi librement au réseau.

2.5 Performances

La plupart des mesures montrent que les systèmes basés sur un micro-noyau ont des performances réseau inférieures à celles des systèmes monolithiques [Maeda et Bershad 1992]. On pourrait donc en déduire que les micro-noyaux sont intrinsèquement inadaptés aux architectures réparties.

Or, [Maeda et Bershad 1993] montrent qu'il est possible d'atteindre les performances d'un système monolithique en décomposant les fonctionnalités d'un protocole entre le serveur réseau et une librairie projetée dans l'espace d'adressage des applications.

La librairie contient les services dont les performances sont critiques, comme l'envoi ou la réception d'un message, ce qui permet d'éviter de coûteux changements de contexte entre l'application et le serveur réseau. Ce dernier ne contient plus quant à lui que les services qui nécessitent son intervention, comme l'ouverture et la fermeture d'une connexion.

3 CONCLUSION

Les services répartis offerts par le serveur réseau du système Masix sont :

- l'extension des sémantiques de communication inter-processus de Mach à l'échelle d'un réseau local ;

- la gestion dynamique des protocoles utilisés par les environnements ;
- des primitives fiables de communication de groupe (broadcast, multicast) ;
- l'authentification des entités communicantes et le chiffrement des données transmises ;
- des performances proches de celles des systèmes monolithiques.

REFERENCES

- [Accetta *et al.* 1986] M. Accetta, R. Baron, W. Bolosky, D. Golub, R. Rashid, A. Tevanian, et M. Young. Mach: A New Kernel Foundation For UNIX Development. In *Proceedings of the USENIX 1986 Summer Conference*, June 1986.
- [Card *et al.* 1993] R. Card, E. Commelin, S. Dayras, et F. Mével. The MASIX Multi-Server Operating System. In *OSF Workshop on Microkernel Technology for Distributed Systems*, June 1993.
- [Hadzilacos et Toueg 1994] V. Hadzilacos et S. Toueg. *A Modular Approach to Fault-Tolerant Broadcasts and Related Problems*. Technical Report TR94-1425, Cornell University, 1994.
- [Maeda et Bershad 1992] C. Maeda et B. Bershad. Networking Performance for Microkernels. In *Proceedings of the 3rd Workshop on Workstation Operating Systems*, April 1992.
- [Maeda et Bershad 1993] C. Maeda et B. Bershad. Protocol Service Decomposition for High Performance Networking. In *Proceedings of the 14th ACM Symposium on Operating Systems Principles*, pages 244–255, December 1993.
- [Mével et Simon 1995] F. Mével et J. Simon. Répartition de Charge au dessus du Micro-noyau Mach : Intégration de la Migration dans le Système Masix. In *Actes des Journées de Recherche sur le Placement Dynamique et la Répartition de Charge*, pages 77–80, Mai 1995.
- [Schiper et Sandoz 1993] A. Schiper et A. Sandoz. Uniform Reliable Multicast in a Virtually Synchronous Environment. In *Proceedings of the 13th International Conference on Distributed Computing Systems*, pages 561–568, May 1993.
- [Stevenson et Julin 1995] J. Stevenson et D. Julin. Mach-Us: Unix on Generic OS Object Servers. In *Proceedings of the 1995 Usenix Conference*, January 1995.
- [Tschudin 1991] C. Tschudin. Flexible Protocol Stacks. In *Proceedings of the SIGCOMM '91 Symposium*, pages 197–204, September 1991.