satellites pour fournir des données mises à jour. On accède à *StormCast* par le *World Wide Web* (la «toile d'araignée mondiale» véhiculée par le réseau *Internet*) à l'adresse http://www.cs.uit.no/

ne provient que des lignes de communication, la mise à jour finira par se faire. En revanche, si le serveur qui ne répond plus est hors service, l'ordinateur qui fait la mise à jour attendra indéfiniment ; pendant ce temps, le système sera indisponible. Si l'on se contente d'ordonner à l'ordinateur de cesser l'attente et de ne mettre à jour que le serveur qui fonctionne, le système principal et celui de secours ne seront plus identiques : des erreurs surviendront si le système tente d'utiliser le serveur périmé.

Le marché financier NASDAQ résout ce problème grâce à deux serveurs centraux d'échanges boursiers. Pour éviter toute confusion, un seul des deux serveurs est actif à un instant donné. Les administrateurs du NASDAQ décident eux-mêmes quand basculer sur le serveur de remplacement. Malheureusement, très peu de systèmes distribués peuvent compter sur la sagesse d'un opérateur humain pour détecter les pannes et décider le changement d'un serveur ; les programmeurs doivent généralement automatiser cette décision, afin que la transition survienne sans heurts.

En outre, les systèmes répartis à haute disponibilité gèrent souvent un grand nombre de serveurs et de programmes. Pour ce faire, les systèmes tiennent à jour la liste de leurs membres et gardent ainsi la trace de chaque programme et de son état d'activité. Un programme qui, pour quelque raison que ce soit, ne répond pas est jugé défectueux. En reconnaissant une défaillance en un site, le système change alors sa configuration et redistribue le travail vers les sites opérationnels.

Le système NASDAQ illustre également un second problème de fiabilité des systèmes distribués. Le retard de deux heures dans le début des échanges financiers, en 1994, aurait été évité si les opérateurs avaient immédiatement enclenché le système de secours. Ils choisirent pourtant d'attendre, car ils craignaient qu'une erreur logicielle n'ait causé le mauvais fonctionnement du système principal. Dans ce cas, la panne se serait produite également sur le système de secours, comme ce fut le cas en 1994 pour la panne du système de la compagnie de téléphone *AT&T*, mentionnée auparavant. Comme il est impossible de garantir qu'un logiciel soit exempt d'erreurs, on doit trouver des moyens de réduire le risque que les versions de secours d'un serveur critique ne tombent en panne à leur tour après la défaillance du serveur principal.

À cette fin, les programmeurs ont inventé la duplication active : un programme fait des copies de secours des programmes et des fichiers vitaux en utilisant des «groupes de processus». Un groupe de processus est un ensemble de programmes qui coopèrent. Un système réparti peut comporter plusieurs groupes de processus, et les programmes peuvent appartenir à plusieurs de ces groupes. Chaque groupe est identifié par un nom semblable à un nom de fichier et possède sa propre liste actualisée de participants (l'ensemble des programmes qui coopèrent). En outre, le groupe de processus permet d'envoyer des messages à ses membres. Cette fonction de transmission des messages garantit que chaque membre du groupe reçoit chaque message dans le même ordre, même si l'émetteur tombe en panne pendant la transmission, ce qui est vital pour de nombreuses applications.

Lorsqu'un programme particulier est vital pour maintenir la disponibilité du système, ce dernier crée un groupe de programmes qui sont, chacun, une copie de l'original. Pour garder à jour les données manipulées par le programme dupliqué, le système envoie un message au groupe de programme. Chaque membre réagit en mettant à jour sa copie du programme. Comme tous les programmes «voient» les mêmes mises à jour dans le même ordre, ils restent dans des états cohérents.

La duplication active permet à un système de tolérer les erreurs, car n'importe quel membre du groupe peut traiter n'importe quelle requête : quand une machine tombe en panne, la tâche est confiée à un site opérationnel. En outre, quand une requête ne modifie pas les données, elle est satisfaite par un site unique au lieu de mettre tout le système à contribution. De cette façon, de multiples tâches sont menées en même temps par différents programmes, ce qui accélère l'application en utilisant le traitement parallèle.

Naturellement, si tous les membres d'un groupe de processus traitaient de manière erronée un message, ils tomberaient tous en panne simultanément. Toutefois, les erreurs de ce genre sont exceptionnelles, car les programmeurs ont observé que les erreurs qui passent le plus facilement inaperçues lors des tests du logiciel sont plutôt celles où intervient l'ordre de réception des données. Ces erreurs sont provoquées uni-

quement par des séquences d'événements improbables. Lorsqu'un système utilise la duplication active, les différentes copies voient les mêmes mises à jour dans le même ordre ; toutefois, les mises à jour sont seulement une petite partie des requêtes reçues par un programme. La plupart du temps, les programmes dupliqués travaillent en parallèle, chaque programme traitant son propre ensemble de requêtes dans un ordre spécifique. Dès lors, même si une erreur logicielle échappait aux tests et interférait avec un membre d'un logiciel réparti, il serait improbable qu'elle mette hors service tous les membres d'un groupe de traitement en même temps.

L'idée de la duplication active est simple, mais sa mise en œuvre est difficile. La gestion dynamique des listes de participants, qui changent continuellement, et les communications avec les groupes de processus sont sujettes aux erreurs systèmes et à la perte de messages. Bien que l'informatique répartie se soit imposée au cours de la dernière décennie, la duplication active n'est sortie que récemment des laboratoires de recherche.

La boîte à outils des réseaux robustes

Durant ces dernières années, plus d'une dizaine d'équipes d'ingénieurs ont mis au point des logiciels qui assurent la robustesse des systèmes répartis. Tous offrent une haute disponibilité grâce à la duplication active, mais ils diffèrent sur leurs priorités : les uns privilégient la rapidité ; d'autres, la sécurité.

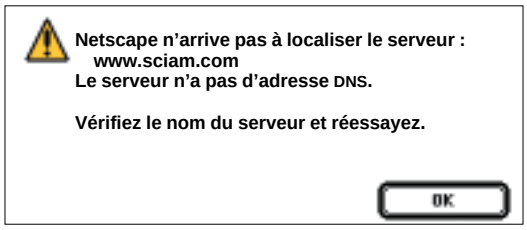
À l'Université Cornell, nous avons réalisé deux logiciels. L'un d'entre nous (K. Birman) a dirigé l'équipe qui a produit le logiciel *Isis* en 1987 ; puis, ensemble, nous avons construit le logiciel *Horus*, achevé en 1994. Les noms de ces programmes indiquent leurs propriétés : la déesse égyptienne Isis ressuscita le dieu Osiris après qu'il avait été tué, lors d'un duel, par Seth, dieu de la guerre ; Horus, fils d'Isis, vainquit finalement Seth. De même, les logiciels *Isis* et *Horus* établissent le fonctionnement d'un système distribué interrompu par une défaillance.

Des logiciels comme *Isis* utilisent un assortiment de fonctions logicielles, ou «outils», qui dupliquent et mettent à jour les données, gardent la trace des groupes de processus et aident à gérer l'appartenance de ces groupes. *Isis* peut aussi répartir le traitement des données entre les serveurs (une procédure nommée partage des tâches). Les systèmes répartis qui utilisent le partage des tâches présentent la plupart des avantages du parallélisme sans recourir à des ordinateurs parallèles spécialisés. En divisant le travail entre de nombreux serveurs fonctionnant de concert, *Isis* permet aux systèmes d'effectuer rapidement de lourdes tâches. De même, lorsqu'une application particulière nécessite davantage de puissance de calcul, on peut ajouter un serveur, en conservant la technique de partage des tâches pour le nouveau groupe. Les performances d'*Isis* et d'*Horus* surprennent souvent les programmeurs, qui considèrent qu'un système robuste est nécessairement lent et onéreux.

La duplication active a été utilisée pour de nombreuses applications : télécommunications, marchés financiers, banques... En Norvège, des informaticiens ont mis au point un système de surveillance de l'environnement

Le *Web* s'emmêle

La majeure partie du *World Wide Web* est invisible. De nombreux utilisateurs ne voient qu'un programme de navigation et des serveurs distants, où les documents à charger sont stockés. Cependant les sites du réseau et le réseau lui-même, qui relie ces sites, est composé de millions d'autres programmes et de serveurs ; des dizaines d'entre eux coopèrent pour fournir un seul document. Par exemple, quand on cherche une information dans un serveur du centre de coordination européen du réseau à l'Institut national de recherche en informatique et en automatique, le nom «www.inria.fr» doit être converti en une adresse numérique que les logiciels reconnaissent. Cette tâche est effectuée par une série de programmes de conversion d'adresses avant que l'adresse correcte soit localisée. En outre, toute requête d'accès à un site du réseau passe normalement par un certain nombre de relais, qui sauvegardent une copie des documents fréquemment demandés, afin de diminuer la charge de travail des serveurs *Web* distants. Ainsi, quand un intermédiaire proche a stocké une copie du document recherché, l'utilisateur évite un long transfert de fichier à travers le réseau.



Ce système de relais est classique dans les réseaux, mais c'est une source de défaillances : quand un serveur de nom ne répond pas, ou quand un relais échoue, ou quand le serveur *Web* tombe en panne, la requête initiale n'aboutit pas. Aussi l'incontournable message d'erreur du *Web* qui signale qu'un serveur est bloqué ou occupé *(ci-dessous)* peut-il être trompeur : la surcharge ou la panne d'un nombre quelconque de programmes intermédiaires peut aussi conduire à ce message.

Le réseau *Internet* dans sa globalité peut connaître des pannes partielles assez semblables à celles des réseaux téléphonique et électrique. Ainsi, fin 1995, l'un des principaux serveurs de temps du réseau *Internet*, à Atlanta, fut surchargé de façon intermittente. Pendant ce temps, personne sur le *Web* ne pouvait obtenir de documents de serveurs dont l'adresse n'était pas déjà connue du système local. Ce type de pannes gêne un grand nombre de personnes dans le monde entier. Même si une connexion avec un serveur réussit, cela n'exclut pas des erreurs ultérieures. Les copies faites par les programmes intermédiaires du *Web* ne sont pas actualisées lorsque le document original l'est. Aussi les utilisateurs n'ont-ils aucune garantie que la version qu'ils consultent est la dernière mise à jour. Si les documents nécessaires ne sont pas gardés à jour, certaines applications dépendantes du temps perdent alors leur fiabilité. D'un côté, les programmes mandataires du *Web* améliorent la fiabilité du réseau *Internet* en diminuant sa charge de travail, mais, de l'autre, ils diminuent la fiabilité en créant l'éventualité de l'obtention d'informations périmées.

Les projets vitaux, sur le réseau *Internet* ou sur d'autres systèmes informatiques répartis, requerront la garantie que les informations extraites du système soient précises, mises à jour et toujours disponibles. On peut éviter des problèmes en sauvegardant les informations vitales par le biais de la duplication active décrite dans l'article.