fondé sur cette technique *(voir la figure 2)*. L'organisme français de contrôle du trafic aérien expérimente également cette technique pour l'appliquer à une nouvelle génération de logiciels de contrôle du trafic. Des usines de production ont utilisé les groupes de processus pour coordonner les tâches et pour reconfigurer les chaînes d'assemblage lorsque certains équipements de ces chaînes sont retirés pour être révisés.

Toutefois, à mesure que la complexité des applications augmente, les informaticiens découvrent que la duplication active a des limitations. Le partage des tâches n'est ni toujours possible, ni désirable : certains systèmes (notamment ceux où les données stockées dans un serveur changent très rapidement) ralentissent quand leurs éléments sont dupliqués. Par exemple, pour les vidéoconférences, la duplication active améliore la tolérance aux pannes du réseau de serveurs qui doivent continuer de fonctionner même quand certains participants sont déconnectés. En revanche, appliquée à la transmission de données vidéo à distance, cette technique ralentira le système sans améliorer sa fiabilité.

Le besoin de flexibilité nous a conduits à mettre au point *Horus*. *Horus* utilise à la fois la technique de duplication active et un traitement modulaire des tâches : un peu comme avec des *Lego*, les différentes pièces de construction, ou modules, d'*Horus* peuvent s'assembler selon n'importe quelle combinaison pour répondre aux besoins spécifiques d'un groupe de processus particulier. Un module chiffre les données, afin de fermer le réseau aux pirates ; un autre module identifie les erreurs de communication potentielles pouvant survenir lorsque des messages sont perdus ou altérés. Les programmeurs qui utilisent *Horus* décident des propriétés nécessaires à leur système et adaptent le système à l'usage projeté. En outre, *Horus* est extensible à l'aide de modules faits sur mesure pour des besoins spécifiques que nous n'avons pas rencontrés ni imaginés.

Horus a un nombre croissant d'utilisateurs dans le monde. À l'Université Cornell, Brian Smith l'a utilisé pour construire un système de vidéoconférences associé à des groupes de discussion et à des outils de travail coopératif assisté par ordinateur.

3. CES PASSAGERS, qui attendent leurs avions à l'aéroport de La Guardia, à New York, sont victimes d'une panne du système local de contrôle de trafic aérien faisant suite à une coupure de courant, en mai 1995. Les efforts pour rajeunir les logiciels américains vieillissants de contrôle de trafic aérien utilisant des systèmes répartis remontent aux années 1980.

Un manque de volonté?

Notre travail sur *Isis* et sur *Horus* nous a convaincus qu'une étude soignée assure la fiabilité d'un réseau informatique. Toutefois, la réalisation d'autoroutes de l'information robustes coûtera plus cher et prendra plus de temps que ne le voudraient les fabricants et les utilisateurs. Les logiciels de gestion des systèmes répartis sont habituellement construits avec la technique existante, qui n'a pas été conçue pour être fiable. En outre, les informaticiens doivent trouver des techniques nouvelles pour les très gros réseaux : un système extrêmement robuste lorsque 50 utilisateurs y accèdent simultanément peut se révéler trop lent, et donc non fiable, lorsque 5 000 personnes s'y connectent.

Bien que les programmeurs aient appliqué la technique de l'informatique répartie robuste avec succès dans certains cas, le public entend plus souvent parler des défaillances des systèmes non robustes. Ainsi, ces dernières années, des dizaines de rapports ont fait état de problèmes du système de contrôle du trafic aérien américain. Fin 1995, le système de Los Angeles tomba en panne, mettant les contrôleurs dans l'impossibilité de communiquer avec les avions ; une collision en plein ciel fut évitée de justesse.

Pis encore, les mises à jour des logiciels de contrôle de trafic aérien commandées, en 1982, par le ministère américain de l'aviation ont été retardées et les logiciels finalement proposés sont moins performants que ceux qui avait été initialement choisis : les aspects «haute disponibilité» et «distributivité» ont été presque éliminés. Pourtant, les contrôleurs aériens considèrent que le système existant est inapproprié, notamment parce qu'il lui manque une architecture de logiciel distribué et qu'il a perdu de sa fiabilité avec l'âge.

L'industrie logicielle est-elle en crise? Si une crise existe, c'est peut-être autant par manque de volonté que par manque de moyens. Tous les ingénieurs n'ont pas besoin de rendre robustes leurs logiciels réseaux : la pression du public pour accroître la fiabilité ne semble pas s'étendre au-delà de quelques applications particulièrement sensibles. Les sociétés qui commercialisent les logiciels d'informatique répartie précisent souvent, dans les garanties de leurs produits, que la technique de programmation employée n'est pas assez fiable pour un usage dans des applications critiques, avouant implicitement que la fiabilité n'est pas un objectif. C'est un peu comme si un constructeur d'automobiles vendait ses voitures en prévenant que les véhicules pourraient ne pas

Des langages aux systèmes

La recherche française étudie beaucoup les problèmes fondamentaux que posent les systèmes répartis. Par exemple, l'utilisation croissante du *Web* pour des transactions commerciales (en particulier bancaires) imposent des protocoles de communication très sûrs : le projet *Verified Internet Protocol* (protocole certifié pour *Internet*) établit une «preuve formelle» de la confidentialité des protocoles de transactions proposés sur le réseau *Internet*. Citons aussi la conception de modèles théoriques des systèmes distribués, qui part du constat que la plupart des langages disponibles pour écrire des applications réparties intègrent mal les fonctions primitives donnant accès à la distribution : l'obtention des programmes mobiles et la résistance aux pannes locales nécessitent des appels système complexes. Inspiré de la théorie des «processus mobiles» de R. Milner, le J-calcul *(Join calculus)* apporte un modèle simple de la distribution qui définit précisément le comportement des primitives de distribution en présence de migrations et de pannes. À l'aide de ce calcul, on espère créer un langage de programmation de haut niveau intégrant directement la programmation distribuée et mobile.

Sur le plan moins spéculatif des systèmes d'exploitations, la recherche française a été à l'origine de *Chorus*, une famille de composants de systèmes d'exploitation qui permet à partir d'un ensemble réduit de fonctionnalités de base, de créer un système d'exploitation adapté aux contraintes de l'application répartie à réaliser. Les travaux actuels de la recherche en systèmes d'exploitation répartis concernent les mécanismes de partage d'informations dans les grands systèmes répartis, la tolérance aux pannes, en particulier dans la vidéoconférence, ainsi que l'application des techniques de haute disponibilité aux distributeurs de films vidéo à la demande : la défaillance d'un serveur pendant la diffusion du film doit rester imperceptible au spectateur qui conserve ainsi l'illusion d'un service continu.

Le système français concurrent du logiciel *Isis* cité dans l'article est le système *SafeTech* qui consiste en une couche logicielle intermédiaire qui intercepte les communications de l'application et assure la tolérance aux pannes sans modification du programme d'origine.

En ce qui concerne plus spécifiquement le système *Horus* que nous décrivent K. Birman et R. van Renesse, la recherche française a joué un rôle indirect dans sa conception : une version d'*Horus* a été écrite en *Caml*, un langage de programmation mis au point par les chercheurs de l'Institut national de recherche en informatique et en automatique. Ce langage est utilisé pour l'enseignement de l'option informatique dans les classes préparatoires aux Grandes Écoles. Les chercheurs américains ont pris le risque d'utiliser un langage de haut niveau pour écrire une application proche du système, et la qualité du logiciel obtenu montre que ce pari est gagné.

Si les informaticiens qui ont écrit *Horus/ML* (la version *Caml* du logiciel *Horus*) ont ainsi bénéficié du travail fait en France, en récupérant par le réseau *Internet* un langage de programmation adapté à leur problème, réciproquement, les développeurs et les usagers du langage *Caml* tirent bénéfice de l'expérience, puisque *Horus/ML* permet maintenant de créer et de gérer en *Caml* des systèmes répartis.

Cette synergie du travail d'équipes distantes de plusieurs milliers de kilomètres est caractéristique de la recherche actuelle. C'est bien entendu le réseau mondial d'information qui permet cette coopération. Dans le cas d'*Horus* et de *Caml* cette coopération porte directement sur l'amélioration de la fiabilité des réseaux qui permettront bientôt à tous de communiquer plus sûrement et plus facilement. Ainsi l'existence du réseau et des systèmes distribués rend possible ces recherches sur le réseau et les systèmes distribués, sujets et acteurs à part entière de la recherche qui les concerne...

Pierre WEIS – INRIA–Rocquencourt

fonctionner sur autoroute ! Les équivalents informatiques des ceintures de sécurité et des coussins gonflables sont rares, car les ingénieurs et les usagers s'intéressent plus à la rapidité et à la convivialité qu'à la fiabilité.

Lorsqu'on pense à la fiabilité, on imagine souvent des systèmes lents et pesants, à l'opposé de ces autoroutes de l'information tant vantées. Pourtant la technique qui rend les systèmes répartis robustes n'est pas nécessairement lente ni désagréable à utiliser. Alors que les informaticiens trouvent sans cesse de nouvelles utilisations aux liaisons entre les ordinateurs, la question s'impose : ces ponts supporteront-ils l'énorme flux d'informations qu'on se propose de leur faire transmettre ? Nous pensons que les systèmes répartis robustes offrent un outil précieux pour relier rapidement et de manière fiable des ordinateurs. Dans notre future société de l'information, ces réseaux offriront une nouvelle manière de concevoir les affaires et les loisirs. Toutefois, si un système réparti n'est pas construit pour fonctionner de façon robuste, il peut être préférable de ne pas le construire du tout.

Kenneth BIRMAN est professeur d'informatique à l'Université Cornell. Robbert van RENESSE est chercheur dans cette même université.

J. GRAY, J. BARTLETT et R. W. HORST, *Fault Tolerance in Tandem Computer Systems*, in *The Evolution of Fault-Tolerant Computing*, sous la direction de A. Avizienis, H. Kopetz et J.C. Laprie, Springer-Verlag, 1987.

Ivars PETERSON, *Fatal Defect : Chasing Killer Computer Bugs*, Random House, 1995.

R. BALTER, J.P. BANÂTRE, S. KRAKOWIAK, *Construction des systèmes d'exploitation répartis*, Collection Didactique, 1991, *INRIA-UCIS* diffusion.

Jean-Marie RIFFLET, *La communication sous Unix*, 7, Ediscience International,1990.

Informations disponibles sur le *Web* :

- Le programme *Horus* : http://www.cs.cornell.edu/Info/Projects/horus/
- Partage d'informations : http://prof.inria.fr/
- Le langage *Caml* :http://pauillac.inria.fr/caml/caml-fra.html
- Serveurs de films vidéo :http://www.inria.fr/Rapports/solidor/node21.html

L'horreur des mines antipersonnel

GINO STRADA

Les mines antipersonnel tuent ou mutilent 26 000 personnes par an, surtout des civils et des enfants. Leur interdiction est urgente.

Au Rwanda, le terrible bain de sang avait cessé. Alphonsine rentrait chez elle, avec sa famille, quand elle marcha sur une mine : l'explosion lui déchiqueta les jambes et lui fractura l'avant-bras gauche. À l'hôpital de Kigali, nous avons réparé les dégâts du mieux que nous pouvions : nous avons amputé ses deux jambes au-dessus du genou. Sa sœur était touchée à la tête : un fragment métallique lui était entré dans le cerveau. Elle mourut six heures après l'opération, sans avoir repris conscience. Le père des deux fillettes, qui se tenait à plusieurs mètres d'elles, ne souffrait «que» de nombreuses petites plaies au thorax.

De nombreux civils, comme Alphonsine et sa famille, sont aujourd'hui la cible des mines, car la plupart des conflits actuels sont des guerres civiles, des luttes d'indépendance, des «épurations ethniques», des opérations terroristes. Les soldats utilisent fréquemment des armes dévastatrices dans des régions très peuplées. Beaucoup de groupes armés se mêlent à la population, sans uniforme pour éviter d'être identifiés, et ils utilisent parfois les civils comme bouclier. En outre, terroriser les populations civiles est une vieille stratégie militaire.

Depuis le début du siècle, de plus en plus de civils sont victimes de la guerre. Pendant la Première Guerre mondiale, 15 pour cent des tués étaient civils. Pendant la Seconde Guerre mondiale, les victimes civiles comptaient pour 65 pour cent des morts. Dans les conflits actuels, plus de 90 pour cent des blessés seraient des civils.

L'un des aspects les plus dramatiques de cette situation est la diffusion d'armes, telles que les mines antipersonnel, qui constituent une menace indiscriminée et durable. Elles ne distinguent pas le pied d'un combattant de celui d'un enfant qui joue. Elles ne reconnaissent ni cessez-le-feu, ni accord de paix. Elles tuent ou mutilent plusieurs décennies après la fin des hostilités. Ce sont des armes de destruction de masse à action lente.

Les mines ont été utilisées de diverses façons depuis le début du siècle, mais la stratégie militaire en a progressivement fait un usage de plus en plus important. Autrefois, elles interdisaient certaines régions à l'ennemi, canalisaient ses déplacements ou protégeaient des installations stratégiques. Aujourd'hui, elles empêchent les populations de s'approvisionner en eau, en bois ou en essence, leur interdisent d'emprunter les voies de communication, voire d'accéder aux cimetières. Dans de nombreux pays, les hélicoptères, l'artillerie ou d'autres moyens de largage à distance ont servi à disséminer des mines dans les villages et dans les champs, actes délibérés de terrorisme envers les populations civiles.

Une faible pression suffit

En termes techniques, une mine antipersonnel est un dispositif conçu pour tuer ou mutiler toute personne qui le déclenche. Les mines antichars, en revanche, sont spécifiquement conçues pour détruire les tanks et les véhicules, et eux seuls : elles n'explosent que lorsqu'elles subissent une pression de plusieurs centaines de kilogrammes. Les mines antipersonnel sont petites (parfois moins de dix centimètres de diamètre) et difficiles à détecter. Certaines ont une couleur et une forme qui les rendent quasi invisibles.

Une mine antipersonnel est activée quand une victime déclenche le mécanisme de mise à feu en appuyant directement sur la mine ou en tirant sur un fil déclencheur tendu au ras du sol. Un détonateur met alors à feu une charge auxiliaire (une petite quantité d'explosif puissant), qui fait enfin exploser la charge principale de la mine.

Ces dernières années, les industries de l'armement ont mis au point des mines en plastique, qui contiennent peu de métal et sont plus fiables, plus durables, plus difficiles à détecter... et aussi plus difficiles à désamorcer. En outre, des systèmes de largage à distance, par hélicoptère, par exemple, permettent de poser des milliers de mines en quelques minutes. Comme ce type de minage ne permet pas de dresser une carte du champ de mines, la localisation ultérieure est devenue très difficile.

Les mines antipersonnel sont faciles à fabriquer et peu coûteuses (entre 15 et 60 francs pièce). Environ 50 pays, y compris des pays en développement, ont produit quelque 350 modèles et les ont vendus aux divers groupes armés du monde.

Combien de mines posées de par le monde n'ont-elles pas explosé ? Selon l'Organisation des Nations unies et diverses associations humanitaires,

1. LA PETITE MINE SB-33, fabriquée en Italie, est montrée ici grandeur nature. Elle se confond si bien avec les pierres qu'elle en devient presque invisible. Lorsqu'une personne marche sur une telle mine à effet de souffle, son pied ou sa jambe sont arrachés.