Projet SOR

Systèmes d'objets répartis

Localisation : *Rocquencourt*

Mots-clés : cache, chaînes de PSS, cohérence, liaison, mémoire partagée répartie, mesures de performance, mobilité, objet réparti, persistance, ramasse-miettes réparti, référence, réplication, système réparti, travail coopératif, WWW.

Accès World-Wide Web : http://www-sor.inria.fr/

1 Composition de l'équipe

Responsable scientifique

Marc Shapiro, directeur de recherche, INRIA

Responsable permanent

Mesaac Makpangou, chargé de recherche, INRIA

Secrétaire

Nelly Maloisel, assistante de projet, INRIA

Ingénieurs experts

Sytse Kloosterman, (à mi-temps)
Ian Piumarta
Fabio Riccardi

Collaborateurs extérieurs

Vincent Bouthors, ingénieur de recherche, Bull.
Patrick Duval, professeur assistant, Pôle Universitaire Léonard de Vinci.
Bertil Folliot, maître de conférences, Université Paris 7.

Chercheurs doctorants

Luciana Arantes, boursière du gouvernement brésilien, Université Paris 6.
Aline Baggio, boursière INRIA, Université Paris 6.
Xavier Blondel, boursier INRIA, CNAM.
Georges Brun-Cottan, boursier INRIA, Université Paris 6.
Paulo Ferreira, boursier JNICT (Portugal), Université Paris 6.
Yann Hervé, boursier MENESR, Université Paris 6.
Julien Maisonneuve, Université Paris 6.
Guillaume Pierre, boursier INRIA, Université d'Évry-Val d'Essonne.

Stagiaires

Cédric Adjih, Université Paris-Sud.
Fabrice Albrecht, Université Versailles Saint-Quentin.
Éric Bérenguier, École Supérieure d'Électricité.
Stéphane Dugelay, École Nationale Supérieure des Télécommunications.
Fabrice Le Fessant, École Polytechnique.
Laurent Martelli, CNAM - IIE.
Eberhard Osthus, Université Paris VI.

2 Présentation du projet

Le projet SOR a pour thème de recherche les mécanismes de partage d'information dans les systèmes répartis de grande échelle. Le partage d'information intéresse par exemple la conception assistée par ordinateur, le travail de groupe ou les logiciels d'entreprise. Nous nous plaçons dans une perspective « système », c'est-à-dire que nous recherchons des mécanismes généraux, orthogonaux entre eux, et indépendants d'un langage ou d'une classe trop étroite d'applications. Le projet SOR a aussi pour vocation de transférer dans l'industrie et le grand public les technologies développées par le projet.

En 1996, nos principaux axes de recherche ont été la gestion automatisée de la mémoire partagée répartie, la gestion des références dans des grands réseaux, la gestion de la réplication à grande échelle, et les outils pour le travail coopératif sur le World-Wide Web.

2.1 Axes de recherche

2.1.1 Ramassage de miettes en mémoire répartie persistante et partagée

Mots-clés : mémoire répartie, mémoire virtuelle partagée, ramasse-miettes.

L'écriture de programmes répartis reste une tâche difficile. Le paradigme de communication par messages asynchrones, qui est à la base de la répartition, est trop complexe à programmer. Le paradigme de l'appel de procédure distante (RPC pour *Remote Procedure Call*) est légèrement plus simple mais se révèle coûteux. De plus, il ne fait que faciliter la communication, sans résoudre les problèmes plus profonds de la répartition : parallélisme, réplication, cohérence, défaillances, coûts, etc.

Le paradigme de la mémoire partagée est plus facile d'utilisation. Une mise en œuvre de mémoire partagée au-dessus d'un système réparti, dite *mémoire répartie virtuellement partagée* (ou MRVP) est capable de masquer les problèmes notés ci-dessus[1]. Les MRVP constituent un domaine de recherche extrêmement actif dans le monde. Les MRVP sur réseau de stations de travail ont été utilisées jusqu'ici pour le calcul scientifique haute performance, en remplacement des multiprocesseurs massivement parallèles, trop chers.

Une MRVP facilite le partage des données entre programmes parallèles. Toutefois, il n'y a pas de données « persistantes » conservées en vue d'une utilisation ultérieure. La persistance est pourtant indispensable, aussi bien pour les tâches quotidiennes comme le traitement de texte, que pour les applications émergentes comme le travail coopératif. Les programmeurs gèrent aujourd'hui la persistance manuellement, par l'intermédiaire de fichiers ou de bases de données.

C'est pour résoudre ces problèmes que nous proposons une MRVP *persistante*, appelée Larchant. Dans Larchant, les données des différents programmes sont allouées dans une mémoire commune et partagées, comme en programmation centralisée, par accès par référence. Toute donnée devient persistante si nécessaire. Larchant permet de répartir et de rendre persistants, de façon presque transparente, des programmes ou des structures de données existantes, même écrits dans des langages primitifs comme C ou C++.

[1] Le parallélisme n'est pas masqué car il est vu comme un avantage, le traitement parallèle permettant d'améliorer les performances.

Un problème majeur d'une telle mémoire est le ramassage de miettes. Pour ce problème non résolu auparavant, nous proposons une solution satisfaisante, décrite ci-après (cf. §3.1)

Le système Larchant a été développé dans le projet SOR depuis 1992, notamment dans le cadre de la thèse de Paulo Ferreira [440]. En 1996, ce travail a porté principalement sur la finition et l'amélioration du prototype, les mesures de performances sur le prototype Larchant-BMX (cf. § 3.1), ainsi que sur la valorisation de ces recherches (projet PerDiS, cf. § 3.1.4, § 5.2.2).

2.2 Gestion des références dans les systèmes de grande échelle

Mots-clés : chaîne de PSS, liaison, ramasse-miettes.

Une *référence* est ce qui permet d'identifier, d'accéder et donc de partager un objet. Mécanisme de base de tout système, les références doivent être performantes et avoir une sémantique bien spécifiée. En particulier, elles doivent permettre le ramassage de miettes, être bien typées, et se comporter proprement en présence de fautes.

Au cours des années passées, nous avons défini et prototypé les Chaînes de Paires Souche-Scion (PSS), un mécanisme de références propre, efficace, tolérant les fautes non byzantines, et permettant le ramassage de miettes. Nous avons aussi proposé un protocole de liaison des références réparties, appélé Hobbes. Ces deux mécanismes représentent des avancées dans la gestion des références réparties et le ramassage de miettes dans les systèmes distribués. Les résultats sont désormais mûrs et formeront la base des développements du projet SOR.

2.3 Gestion de la réplication à grande échelle

Mots-clés : réplication, grande échelle, contrat de cohérence.

La réplication est essentielle pour le partage à grande échelle parce qu'elle permet d'augmenter la disponibilité des données et ce, malgré les surcharges, la mobilité des utilisateurs, les pannes, et les partitionnements du réseau. La réplication à grande échelle pose cependant un problème majeur : assurer la cohérence des réplicats tout en conservant des performances acceptables. Il y a lieu de rechercher un compromis entre contraintes de cohérence, tolérance aux fautes et performances. Malheureusement, aucun compromis ne satisfait les attentes de toutes les applications.

Nous avons proposé une technique de construction des systèmes de réplication qui permet d'adapter la politique de réplication, objet par objet, et application par application, selon la sémantique des premiers et les besoins des seconds. Notre proposition est centrée sur la notion de *contrat de cohérence* liant les entités applicatives utilisant un objet répliqué et le système de réplication ; il détermine le degré de cohérence et de disponibilité garanti aux utilisateurs. L'architecture est basée sur un protocole méta-objet (Meta-Object Protocol ou MOP).

Cette année, l'accent a été porté sur le raffinement et l'extension de la notion de contrat de cohérence afin de prendre en compte les aspects grande échelle (cf. §3.3).

2.4 Outils pour le travail coopératif sur le World-Wide Web

Mots-clés : travail coopératif.

Le succès du Web a fait naître de nombreux besoins de travail coopératif à grande échelle : édition collaborative d'un document, partage des espaces de travail, mise en commun des expertises dans un groupe (annotation de documents), partage d'information entre utilisateurs ou machines mobiles, etc.

Ces différentes applications n'ont pas toutes les mêmes attentes. De plus, les besoins varient notamment en fonction des caractéristiques des connexions entre les participants, du nombre de ces participants ainsi que de leurs localisations.

Pour répondre à ces besoins de façon générale, nous construisons un système de cache flexible, au-dessus duquel viendront se greffer des outils spécifiques à chacune des tâches coopératives que nous ciblons (cf. §3.4). À terme, nous voulons améliorer la qualité de service offerte aux applications coopératives.

2.5 Vie du projet

Nous avons tenu notre deuxième « retraite » en juin 1996. Nous avons invité deux personnes extérieures, Sacha Krakowiak, professeur à l'Université Fournier de Grenoble et Bertil Folliot, maître de conférences à l'Université Paris 7.

Trois collaborateurs extérieurs nous ont rejoint cette année. Patrick Duval (pôle universitaire Léonard de Vinci) et Vincent Bouthors (Bull) renforceront notre pôle de compétence Web, ce qui facilitera les transferts des technologies SOR dans le Web, notamment les techniques de réplication et de gestion de caches répartis (cf. §4.3). Bertil Folliot (université Paris 7 et MASI) renforcera l'encadrement de la recherche sur la MVRP et sur la cohérence des caches.

Le projet Esprit LTR PerDiS dont l'INRIA-SOR est coordinateur a été accepté par la commission européenne. Le projet a démarré en décembre 96.

Aline Baggio a séjourné pendant trois mois à l'Université Carnegie Mellon, dans l'équipe Coda du professeur Satyanarayanan (cf. §3.4.2).

3 Actions de recherche

3.1 Mémoire partagée persistante : Larchant

Mots-clés : mémoire partagée répartie, mesures de performance, ramasse-miettes réparti.

Larchant est une mémoire virtuelle répartie, persistante et partagée. Dans Larchant, les références sont des pointeurs mémoire, modifiés directement par les programmes utilisateurs. Larchant garantit la *persistance par atteignabilité* : les objets persistants sont tous ceux, et uniquement ceux, atteignables depuis une « racine de persistance ».

Cette action s'est organisée, cette année, autour des activités suivantes : la spécification et la preuve détaillée du ramasse-miettes, et la finition du prototype Larchant-BMX (§ 3.1.1) ; la création d'un environnement de programmation facilitant l'interface des applications avec Larchant (§ 3.1.2) ; et l'étude des caractéristiques des applications (§ 3.1.3). Nous avons aussi consacré beaucoup de temps à la préparation du projet PerDiS de généralisation et de valorisation des résultats de Larchant (§ 3.1.4, § 5.2.2).

3.1.1 Spécification, preuve détaillée, mise en œuvre, et mesures de surcoût du ramasse-miettes

Participants : Paulo Ferreira, Marc Shapiro, Fabio Riccardi, Xavier Blondel

Cette activité s'intéresse au problème de ramassage de miettes dans une mémoire répartie, partagée et persistante. Les deux grands principes de conception sont : éviter tout surcoût de communication et ne pas interférer avec le protocole de cohérence mémoire.

Nous avons établi que le ramasse-miettes peut se contenter d'une vue non cohérente de la mémoire, et peut donc travailler localement, sans causer d'activité de cohérence, ni interférer avec les applications. En 1994, nous avions défini un algorithme de ramasse-miettes par morceaux, respectant l'autonomie des machines, et n'imposant ni entrées-sorties, ni trafic réseau, ni prise de verrous. L'algorithme se prête donc aisément à l'utilisation sur systèmes de grande échelle et/ou fortement parallèles.

En 1995 et 1996, l'accent a été mis sur la preuve formelle de la correction de cet algorithme et la mesure des surcoûts [440, 444]. L'algorithme satisfait les propriétés d'innocuité (« safety ») et de vivacité

(« liveness »). Nous décrivons un modèle formel de comportement du réseau support, de la mémoire répartie, du ramasse-miettes réparti ainsi que des applications. Nous commençons par prouver par récurrence sur le nombre de pointeurs que l'algorithme est correct en l'absence de réplication des données. Nous étendons ensuite la preuve au cas où les données peuvent être répliquées sur plusieurs sites par récurrence sur le nombre de réplicats.

La théorie que nous avons développée établit des conditions suffisantes de correction du ramasse-miettes en présence de mémoire répliquée, même non cohérente. Les conditions portent sur l'ordre des actions du ramasse-miette (RM), sur l'ordre des livraisons des messages (causal), sur les interactions cohérence-RM, et sur la coopération inter-RM. Nous avons donc aussi complété le prototype Larchant-BMX, et mesuré le surcoût dû au ramasse-miettes, qui reste raisonnable bien que supérieur à nos prévisions.

Cette activité constitue l'essentiel de la thèse soutenue par Paulo Ferreira [440] en mai 1996.

3.1.2 Environnement de programmation pour Larchant

Participants : Eberhard Osthus, Fabrice Le Fessant, Marc Shapiro

L'un des objectifs principaux de Larchant est de simplifier la distribution d'applications existantes avec une modification minimale du code. Pour cela, nous avons développé un outil capable d'analyser statiquement les fichiers de code binaire des programmes utilisateurs pour en extraire toutes les informations dont le ramasse-miettes de Larchant aura besoin dynamiquement [450].

Cet outil généralise l'outil OCI *Object Class Information*, précédemment développé. Il permet de récupérer l'ensemble des informations relatives aux types, et en particulier la position et le type des pointeurs dans un objet. Le nouvel outil est adaptable à tout compilateur C++, et à toute application ayant besoin d'informations sur le type des objets ; il génère de façon complète toutes les informations sur la signature des objets.

Nous avons développé un mécanisme d'emballage-déballage générique, c'est-à-dire capable de transmettre un objet de n'importe quel type sur un canal de transmission, sans génération statique de code. L'algorithme d'emballage tient compte des cycles, de l'héritage (simple et multiple), des tableaux, et des objets emboîtés.

3.1.3 Mesure et caractérisation d'applications réparties

Participants : Cédric Adjih, Luciana Arantes, Marc Shapiro

Larchant offre de nouveaux services aux applications : persistance, ramasse-miettes, répartition des données, cohérence, etc. Les performances globales de ce système dépendent en partie de l'adéquation au comportement réel des applications des heuristiques de ramasse-miettes, de gestion de cache, et de regroupement des objets.

Dans le but de recueillir les informations nécessaires à l'évaluation de ces heuristiques, une première série de mesures sur des applications réelles a été faite. Dans le cadre de son stage de DEA [446], Cédric Adjih a réalisé un outil de trace et d'analyse du comportement mémoire d'une application quelconque. Cet outil fournit des informations détaillées sur le graphe des objets : combien d'objets sont alloués et désalloués, à quel moment, quelle est leur adresse, et quels sont les liens de ces objets entre eux.

Les mesures réalisées sur un certain nombre d'applications standards (par exemple gs, make, awk, ical, etc.) écrites en C ou C++ ont confirmé ce qui était connu sur les applications en Lisp ou Smalltalk, et ont aussi révélé des informations nouvelles :

- Les applications allouent beaucoup d'objets temporaires.
- Le degré incident vers un objet dépasse rarement 1.

- Un pointeur connecte en général deux objets ayant été alloués à des instants très proches l'un de l'autre.
- Les cycles de miettes existent. La plupart de ces cycles comportent peu d'objets, mais un petit nombre en comporte beaucoup.
- Les objets sont éparpillés en mémoire par l'allocateur : des objets liés par un pointeur, et alloués à des instants proches, sont parfois placés à des adresses éloignées en mémoire.

L'outil développé à cette occasion pourra servir à nouveau pour l'analyse de programmes de plus grande taille, pour l'analyse de bases de données persistantes, et pour l'analyse du graphe de documents dans le Web.

3.1.4 PerDiS

Participants : Fabio Riccardi, Marc Shapiro, Xavier Blondel, Luciana Arantes, Sytse Kloosterman

Larchant (§3.1) a montré tout l'intérêt, pour les programmeurs d'application, d'une mémoire répartie partagée persistante. La technologie éprouvée dans Larchant-BMX est bien adaptée aux applications de partage de données sur le long terme et dans des grands réseaux. Elle représente par exemple le support idéal pour la classe des applications d'« ingénierie coopérative ».

Dans cet esprit, nous avons pris contact avec une industrie potentiellement utilisatrice de cette technologie : *l'ingénierie coopérative pour le bâtiment*. La construction d'un grand bâtiment réunit des intervenants très divers, appartenant à plusieurs entreprises (généralement en compétition par ailleurs). Les participants sont très dispersés géographiquement.

Pour passes des outils de CAO mono-poste actuels à une véritable coopération, les technologies standards comme DCE ou CORBA ne donnent pas satisfaction. D'une part leur interface très pauvre oblige à réécrire complètement les applications. D'autre part le mécanisme d'accès distant souffre de mauvaises performances. Enfin et surtout, les problèmes de fond de la répartition (accès aux données, cohérence, tolérance aux pannes, etc.) ne sont pas résolus. Au contraire, la technologie de Larchant résoud ces problèmes, tout en permettant de conserver à peu près intacte la structure des programmes existants et ce, en maintenant des performances satisfaisantes. C'est l'objectif central du projet Esprit LTR PerDiS.

PerDiS, démarré en décembre 1996, mettra en œuvre une mémoire persistante répartie pour les applications d'ingénierie coopérative pour le bâtiment. Outre les problèmes déjà traités dans Larchant (cohérence et ramasse-miettes), PerDiS s'intéresse à la tolérance aux fautes, à la sécurité, au fonctionnement en grande échelle, et à l'utilisation de l'informatique mobile. Les spécifications et la mise en œuvre du système ont déjà pris une bonne avance. Les aspects industriels de ce projet sont développés en § 5.2.2.

3.2 Gestion des références réparties

Mots-clés : chaîne de PSS, désignation, liaison, mobilité, ramasse-miettes, références réparties.

Au cours des années passées, nous avons défini et prototypé les Chaînes de Paires Souche-Scion (PSS), un mécanisme de références propre, efficace, tolérant les fautes non byzantines, et permettant le ramassage de miettes. Nous avons aussi proposé un protocole de liaison des références réparties, appélé Hobbes.

Toutefois, pour transférer les résultats dans l'industrie et le grand public, plusieurs points devront être résolus. Sont nécessaires, tout d'abord une mise en œuvre stable intégrant les Chaînes de PSS et le protocole de liaison flexible, des mesures de performance et une documentation détaillée du prototype. Il sera ensuite indispensable de résoudre certains problèmes importants qui ont été délaissés jusqu'alors :

tolérance aux fautes et recouvrement, application aux réseaux de mobiles. Ces travaux sont actuellement en cours et se placent dans le cadre d'un contrat avec le CNET (cf. §3.2 et §4.1).

En 1996, le travail sur la gestion des références en réparti se décompose en trois actions : la réalisation d'un prototype des Chaînes de Paires Souche Scion (PSS) de qualité pré-industrielle (cf. §3.2.1), l'extension des Chaînes de PSS pour les réseaux de mobiles (cf. §3.2.2), et Hobbes (cf. §3.2.3) qui fait le lien entre la désignation et le placement des données ou calculs dans les environnements répartis.

3.2.1 Chaînes de Paires Souche Scion

Participants : Ian Piumarta, Laurent Martelli, Julien Maisonneuve

Les chaînes de PSS sont un mécanisme de désignation répartie. Depuis 1995, nous travaillons sur la mise en œuvre d'un prototype pré-industriel des Chaînes de PSS. Ce travail est maintenant très largement avancé, il ne manque plus actuellement que quelques outils d'aide au développement de systèmes basés sur les Chaînes de PSS. Le transfert de la technologie Chaînes de PSS est déjà en cours au sein du projet SOR et le sera à l'extérieur dès que ces outils de développement seront écrits. La documentation du système des Chaînes de PSS est actuellement en bonne voie.

Un nouveau moniteur graphique a été développé et permet désormais de surveiller les activités et interactions d'applications distribuées basées sur les Chaînes de PSS. Un langage de scripts, en cours de finition, fournit un environnement flexible dans lequel il est possible de réaliser les mesures de performance demandées par le contrat CNET (cf. §4.1).

3.2.2 Extension des Chaînes de PSS

Participants : Aline Baggio, Ian Piumarta

Le but de cette activité est de rendre les Chaînes de PSS utilisables sur machines mobiles ainsi que de faciliter la gestion de la mobilité grâce à ces mécanismes systèmes souples et efficaces.

Cette année, nous avons analysé l'intérêt des Chaînes de PSS pour la gestion de la mobilité [443], ainsi que les problèmes qui dérivent du référençage d'objets sur machines mobiles. Nous avons étendu le mécanisme des Chaînes de PSS en définissant des protocoles de déconnexion et reconnexion des Chaînes de PSS après déplacement, de localisation de mobiles, et de découverte de ressources [442]. Un prototype de Chaînes de PSS intégrant ces nouveaux protocoles est actuellement en cours de réalisation (protocoles réels, serveurs et clients simulés) et devrait être achevé dans les prochains mois.

Par la suite, ces travaux de recherche seront réutilisés dans le cadre d'une problématique plus large, axée sur le partage des informations sur le Web par des utilisateurs ou machines mobiles. Ce recentrage amènera à réutiliser les résultats et certains concepts abordés à l'Université Carnegie Mellon (cf. §3.4.2) et à collaborer plus étroitement avec les axes réplication à grande échelle (cf. §3.3) et outils pour le travail coopératif sur le Web (cf. §3.4).

Cette activité constitue une partie du travail de thèse d'Aline Baggio.

3.2.3 Hobbes : un modèle de liaison de références réparties

Participants : Julien Maisonneuve, Ian Piumarta, Marc Shapiro

Mots-clés : objet réparti.

Un modèle de liaison de références réparties, Hobbes, a été conçu et partiellement intégré dans le nouveau prototype des Chaînes de PSS. Hobbes étend le modèle de programmation des applications réparties, en alliant la transparence et la souplesse nécessaire à la prise en compte des avantages et des problèmes de la répartition. Il se base sur un mécanisme universel mais simple et efficace : un protocole de liaison flexible qui permet le choix par un serveur des mandataires le représentant chez son client.

Le protocole est ouvert, permet l'encapsulation des politiques de l'utilisateur à l'aide de mécanismes système, assure souplesse et versatilité. Il offre en même temps une transparence maximale pour un programme client et des moyens permettant au fournisseur de service de gérer des aspects tels que la tolérance aux pannes.

La spécification et la mise en œuvre de Hobbes constituent l'essentiel du travail de thèse de Julien Maisonneuve, soutenue en octobre 96 [441].

3.3 Gestion de la réplication à grande échelle

Mots-clés : cache, cohérence, réplication.

L'efficacité de la réplication à grande échelle repose sur la spécialisation du protocole de réplication selon les caractéristiques des applications. Compte tenu de la complexité du code de réplication, un enjeu important est de ne pas laisser cette spécialisation à la charge des programmeurs d'application mais de la déléguer au système.

Toutefois, laisser la responsabilité de la spécialisation au système pose un problème difficile parce qu'elle repose sur des informations sémantiques, le type des objets, les invariants applicatifs de cohérence, et les schémas d'accès aux données.

En 1996, nous avons mené deux activités : terminer une architecture d'abstraction de cohérence (cf. §3.3.1) ; analyser les comportements d'applications à grande échelle afin d'étendre notre architecture d'objet répliqué à de tels environnements (cf. §3.3.2).

3.3.1 Abstraction de cohérence : CORE

Participants : Georges Brun-Cottan, Mesaac Makpangou

Nous proposons une bibliothèque de gestionnaires de cohérence (CORE). Les gestionnaires de cohérence sont des composants système réalisant chacun un contrat de cohérence. Ces composants délèguent à des mandataires fournis par le programmeur de l'objet répliqué, l'évaluation d'un petit nombre de prédicats. Ces prédicats sont ceux pour lesquels la prise en compte du type de l'objet permet un gain de performance.

CORE supporte le modèle de réplication actif et total, c'est-à-dire que chaque client possède une copie privée de l'objet et que chaque copie effectue toutes les opérations de mise à jour demandées par tous les clients. Le problème de mise en œuvre essentiel, la factorisation du code réalisant la gestion de cohérence, a été résolu par l'utilisation d'une architecture basée sur un protocole méta-objet (Meta-Object Protocol).

Cette année, le prototype a été porté sur les différentes plates-formes utilisées par l'équipe, en particulier sur Solaris et OSF/1 qui offrent des environnements multi-tâches plus murs que celle de la plate-forme initiale (SunOS 4.1.x). Ce portage a, par ailleurs, bénéficié de la standardisation du langage C++. Dans le cadre de ce portage, nous avons réalisé un paquetage de « threads » portables offrant une interface objet, structuré comme un emballage autour des bibliothèques de gestion de « threads » existantes.

La spécification de la notion de contrat de cohérence, la définition de l'architecture CORE, son prototypage et son évaluation constituent la thèse de Georges Brun-Cottan qui sera soutenue au printemps 1997.

3.3.2 LaSCoW

Participants : Guillaume Pierre, Mesaac Makpangou

Mots-clés : réplication, travail coopératif, WWW.

Le passage à grande échelle des mécanismes de réplication se heurte à la non-uniformité de la qualité de service attendue par les différents utilisateurs. Afin de prendre en compte la diversité de leurs besoins,

nous avons conçu LaSCoW, un modèle de contrôle de concurrence hybride. Ce modèle permet le partitionnement de l'ensemble des utilisateurs d'un même objet répliqué en plusieurs domaines, chaque domaine mettant en œuvre une politique de réplication conforme aux besoins et aux possibilités de ses membres [445].

Afin d'évaluer l'intérêt de LaSCoW, nous avons décidé d'étudier son comportement dans le cadre de quelques applications, grâce à des techniques de simulation. Des simulations sont en cours afin d'analyser le comportement d'un groupe de caches coopératifs utilisant des environnements et protocoles variés.

Cette activité est financée par le consortium W3C et constitue une partie de travail de thèse de Guillaume Pierre.

3.4 Outils pour le travail coopératif sur le Web

Mots-clés : cache, travail coopératif, WWW.

Cette année, nous avons mené trois activités principales : l'intégration dans le système Squid d'un protocole de coopération pour caches Web et une évaluation systématique des politiques de coopération existantes (§3.4.1), le développement d'un cache Web déconnectable (cf. §3.4.2), et le développement d'un système d'annotations sur le Web (§3.4.3),

3.4.1 Caches coopérants

Participants : Mesaac Makpangou, Guillaume Pierre, Stéphane Dugelay, Éric Bérenguier, Patrick Duval

Les caches coopérants collaborent pour diminuer le nombre de requêtes qui nécessitent des accès aux serveurs sources. Toutefois, les politiques de coopération proposées actuellement sont coûteuses.

Cette année, nous nous sommes intéressés à la mise en œuvre et à l'évaluation d'un protocole que nous avons proposé en 1995. La mise en œuvre se fait par remplacement du protocole de coopération dans Squid. Cette approche incrémentale a deux objectifs : limiter le travail de programmation et faciliter la diffusion du résultat. La mise en œuvre commencée par Stéphane Dugelay pendant son stage [448], se poursuit avec Éric Bérenguier. Cette partie du travail est financée par le GIE Dyade.

En même temps, nous menons une évaluation comparative des qualités de service obtenues avec différentes politiques de coopération entre les caches Web. La qualité de service résultante dépend à la fois du schéma d'accès des utilisateurs, de la qualité des liaisons entre les serveurs cache et avec les serveurs, de la taille des caches, de la politique de remplacement, et de la politique de maintien de cohérence des documents en cache avec les originaux détenus par les serveurs.

Le taux de réussite, la mesure habituelle d'évaluation des caches, n'est pas un bon indicateur de qualité de service du World-Wide Web. Nous le remplaçons par des critères plus directement reliés à la qualité de service perçue par les utilisateurs : diminution de la latence, augmentation du débit utile, taux de documents périmés, trafic réseau induit, etc.

Afin d'assurer la vraisemblance de la simulation, nous soumettons des caches réels à une suite de requêtes réelles, extraites des journaux de véritables caches Web. Afin de simuler de façon réaliste un service de cache couvrant un organisme décentralisé, nous utilisons les journaux provenant de l'ensemble des serveurs caches activement en service dans l'INRIA.

Les résultats attendus de ce travail sont une meilleure compréhension du fonctionnement des différents protocoles de coopération entre caches, des conditions dans lesquelles ils sont pertinents, de la façon de structurer la coopération entre caches, et du gain que les techniques de caches coopératifs peuvent apporter aux utilisateurs.

L'activité évaluation est financée par le W3C et constitue une partie du travail de thèse de Guillaume Pierre.

3.4.2 Cache Web déconnectable

Participant : Aline Baggio

De début juin à début septembre, Aline Baggio a effectué un stage à l'Université Carnegie Mellon, dans le groupe de recherche Coda du Professeur Satyanarayanan. Ce stage a entre autres été l'occasion de concevoir un cache Web déconnectable pour machines mobiles.

À la suite de cette expérience, nous avons débuté une étude sur les besoins des applications mobiles et coopératives de partage d'information à grande échelle. Ce projet permettra d'intégrer à la fois les idées abordées à l'Université Carnegie Mellon sur les caches déconnectables, et les développements déjà entamés sur les Chaînes de PSS mobiles (cf. §3.2.2).

3.4.3 Partage d'annotations sur le Web

Participants : Vincent Bouthors, Fabrice Albrecht, Patrick Duval

L'objectif de cette activité est de permettre à un groupe de personnes d'annoter les documents disponibles sur le Web, et de partager ces annotations, afin de mieux exploiter les documents concernés. Ce type de méta-information permettra par exemple d'interdire l'accès de certains documents à certaines catégories d'utilisateurs, ou de faciliter la recherche de documents pertinents. Un exemple est l'utilisation par des enseignants, annotant les documents du Web et utilisant les annotations de leurs collègues. Ceci leur permettra par exemple de trouver les documents présentant un intérêt pédagogique pour leurs élèves.

Nous avons étudié différents systèmes d'annotation, en particulier celui développé par OSF Cambridge, mais il s'est avéré qu'ils avaient des objectifs trop différents des nôtres. Nous avons donc décidé de développer notre propre système[447].

Nous avons aussi réalisé un assistant capable de repérer les documents consultés par le browser Netscape. Il offre ainsi à l'utilisateur la faculté d'associer aux documents des mots clés, des notes et différents commentaires. Le couplage de cet assistant avec l'outil d'annotation est prévu pour les prochains mois.

Ce travail est financé par Dyade.

4 Actions industrielles

Au cours de l'année 1996, le contrat DEC-EERP de soutien à Larchant s'est terminé. Notre contrat CNET (§ 4.1) et la collaboration avec le consortium World-Wide Web (W3C) (§ 4.2) se poursuivent sans changement notable. Des contrats industriels nouveaux sont, soit d'ores et déjà approuvés (Dyade, § 4.3), soit en cours d'évaluation (Génie Phase II).

4.1 Contrat CNET « Chaînes de PSS »

Notre action sur la gestion de références distantes (§ 3.2) fait l'objet d'une collaboration avec le CNET, dans le cadre de sa consultation thématique « Systèmes Répartis et Réseaux Publics ». Nous sommes actuellement en phase de finition d'un nouveau prototype des Chaînes de PSS, de qualité pré-industrielle. Les problèmes de recherche restants incluent l'adaptation des Chaînes de PSS à des réseaux de très grande taille et leur utilisation pour l'informatique mobile et les agents (cf. §3.2.1).

4.2 W3C

L'action sur le travail coopératif conçoit et réalise une infrastructure destinée à augmenter la qualité de service fourni aux utilisateurs coopérant au travers du web. Plus précisément, nous nous attachons à améliorer les moyens d'accès à l'information et à la méta-information sur le Web.

Pour répondre à ce besoin, nous avons conçu LaSCoW, un modèle de contrôle de concurrence hybride, qui permet à un ensemble de participants de répliquer un document en respectant les besoins et possibilités de chacun. L'évaluation de l'intérêt de ce modèle pour quelques exemples d'applications coopératives est en cours (cf. §3.4.1).

4.3 Action WebTools de Dyade

Le but de l'action WebTools de Dyade est de développer des outils pour améliorer la qualité du travail coopératif au-dessus de Web. Cette année, nous avons démarré deux activités : le développement d'un système d'annotations coopératives (§3.4.3) et l'intégration dans le système Squid d'un protocole de coopération de caches Web (§3.4.1).

5 Actions nationales et internationales

5.1 Action nationale

Le projet SOR collabore, au niveau national, avec les équipes de recherche en systèmes répartis dans le cadre du PRC Parallélisme, Répartition, Systèmes (PRS).

5.2 Actions européennes

5.2.1 Broadcast-WG

Broadcast est un projet européen de recherche sur les systèmes répartis de grande échelle. L'ancien BRA Broadcast se prolonge par un Working Group dont nous restons membres, qui démarre en novembre 1996. La plupart des travaux de l'équipe sont inclus dans le champ d'intérêt de Broadcast-WG.

5.2.2 LTR PerDiS

Notre activité Larchant (§3.1) a montré l'intérêt d'une mémoire répartie partagée persistante, qui permet d'étendre de façon très simple des applications centralisées existantes, afin de les rendre réparties et persistantes. La technologie éprouvée dans Larchant-BMX (réplication et accès local aux données, transactions, persistance par atteignabilité) est bien adaptée aux applications de partage de données sur le long terme et dans des grands réseaux. Elle représente par exemple le support idéal pour la classe des applications d'« ingénierie coopérative ».

Le projet Esprit LTR PerDiS *(Persistent Distributed Store for Cooperative Engineering)* 22.533 rassemble l'INRIA Rocquencourt (projet SOR), l'INRIA Rhône-Alpes (projet SIRAC), l'INESC, Queen Mary and Westfield College (QMW), le Centre Scientifique et Technique du Bâtiment (CSTB), et IEZ, entreprise allemande leader dans les logiciels d'ingénierie pour l'industrie du bâtiment. Ce projet bénéficie du parrainage de Bull, Chorus-Systèmes, DEC, et Iona Technologies.

Ce projet, démarrant en décembre 1996, doit valoriser les résultats de Larchant et les généraliser au partage de données entre plusieurs entreprises sur un grand réseau. Les outils d'ingénierie coopérative existants d'IEZ et du CSTB deviendront des outils répartis grâce à la technologie Larchant. De plus, l'environnement PerDiS sera tolérant aux pannes et comportera des mécanismes assurant la sécurité et la confidentialité des données partagées. Les résultats de PerDiS seront diffusés sous forme de logiciel libre dès le 3ème trimestre 97.

5.3 Sejour d'Aline Baggio au CMU

De début juin à début septembre, Aline Baggio a effectué un stage à l'Université Carnegie Mellon, dans le groupe de recherche Coda du Professeur Satyanarayanan. Ce stage a entre autres été l'occasion de concevoir un cache Web déconnectable pour machines mobiles.

6 Diffusion des résultats

6.1 Actions d'enseignement

DEA Systèmes Informatiques de l'Université Paris 6, systèmes répartis avancés, Marc Shapiro (janvier-mars 1996).

École Nationale Supérieure des Télécommunications, Systèmes répartis, cours de 3ème année, Marc Shapiro (février 1996).

ENSTA, systèmes répartis, cours de 3ème année, Mesaac Makpangou et Georges Brun-Cottan (janvier - février 1996).

École Nationale Supérieure des Télécommunications de Bretagne, gestion des objets répartis, cours de 3ème année, Mesaac Makpangou (mai 1996)

Institut Supérieur de Technologie et de Management, systèmes d'exploitation, cours de 2ème année, Ian Piumarta, (décembre 1996 à mars 1997). Réseaux informatiques, cours de 2ème année, Marc Shapiro, Guillaume Pierre, Aline Baggio (décembre 1996 à mars 1997).

6.2 Jurys de thèse

Paulo Ferreira. *Larchant: garbage collection in a cached distributed shared store with persistence by reachability,* Université Paris 6, mai 1996 [440]. *Responsable de la thèse* : Marc Shapiro.

Julien Maisonneuve. *Hobbes : un modèle de liaison de références réparties,* Université Paris 6, octobre 1996 [441]. *Responsable de la thèse* : Marc Shapiro.

Anne-Marie Kermarrec. Université de Rennes-I, octobre 1996. *Rapporteur* : Marc Shapiro.

Karim Mazouni. École Fédérale Polytechnique de Lausane, novembre 1996. *Rapporteur* : Marc Shapiro.

6.3 Conférences internationales

USENIX 96, Usenix Annual Technical Conference, San Diego USA, janvier 1996. *Participant* : Mesaac Makpangou.

MMCN 96, Multimedia Computing and Networking, San Jose USA, janvier 1996. *Participant* : Mesaac Makpangou.

ERCIM/W4G, Fifth ERCIM/W4G Workshop « CSCW and the Web », Sankt Augustin (Allemagne), avril 1996. *Participants* : Patrick Duval, Guillaume Pierre.

ICDCS 96, Int. Conf. on Distributed Computing Systems, Hong-Kong, mai 1996. *Présentation d'une communication :* "Larchant: Persistence by Reachability in Distributed Shared Memory through Garbage Collection.", Paulo Ferreira [444].

PODC 1996, Principles of Distributed Computing, Philadelphia, mai 1996. *Participant* : Marc Shapiro.

POS 1996, Workshop on Persistent Object Systems, Cape May, mai 1996. *Membre du comité de programme* : Marc Shapiro.

IDEA Workshop, Conférence australo-européenne sur la persistance, Lamington National Park, Australie, juillet 1996. *Conférencier invité* : Marc Shapiro.

SIGOPS European Workshop sur le thème « Systems Support for World-Wide Applications ». Connemara, Irlande, septembre 1996. *« Position papers »* : Aline Baggio [442], Guillaume Pierre [445]. *Participant* : Marc Shapiro.

I-WOOOS 1996, International Workshop on Object Orientation and Operating Systems. Seattle, oct. 1996. *Participant* : Marc Shapiro.

OSDI 1996, Symposium on Operating Systems Design and Implementation Seattle, oct. 1996. *Membre du comité de programme* : Marc Shapiro. *Participant* : Mesaac Makpangou.

6.4 Conférences nationales

MVPR, Journées sur les mémoires virtuellement partagées en réparti, Bordeaux, mai 1996. *Co-organisateur, membre du comité de programme* : Marc Shapiro. *Conférencier invité* : Paulo Ferreira.

SAR/CNET, Séminaires Action scientifique systèmes et applications répartis, CNET, Issy-les-Moulineaux, 5-6 février 1996. *Présentation d'une communication* : Aline Baggio [443]. *Participant* : Guillaume Pierre.

CRAC'96, Journées de Recherche sur le contrôle réparti dans les applications coopératives, Paris, 30-31 mai 1996. *Participant* : Guillaume Pierre.

CAR, École d'été INRIA « Construction des applications réparties », Saint Malo, 2–7 septembre 1996. *Participant* : Guillaume Pierre.

6.5 Séminaires de recherche

Modeling a Cached Distributed Persistent Store and its Garbage Collector,

Séminaire à l'université du Maryland, mai 1996. Marc Shapiro.

Innocuité d'un ramasse-miettes en mémoire partagée répartie,

Séminaire à Australian National University (Canberra), juin 1996. Marc Shapiro.

Mise en œuvre de la persistance par atteignabilité dans une mémoire partagée répartie,

Séminaire à Sydney University (Australie), juillet 1996. Marc Shapiro.

Le ramasse-miettes et la persistance par atteignabilité en réparti : état de l'art et solutions.

Séminaire à l'université de Genève, novembre 1996. Marc Shapiro.

Broadcast Working Group,

réunion de lancement, Rennes, novembre 1996 (§ 5.2.1). *Participants* : Aline Baggio, Georges Brun-Cottan, Patrick Duval. *Présentations de communications :* Guillaume Pierre, Fabio Riccardi, Marc Shapiro.

6.6 Réunions diverses

Comité de Programme OSDI 1996, San Francisco (USA), juillet 1996. Marc Shapiro.

PerDiS Preliminary Platform, réunion de travail, Paris, septembre et décembre 1996 (§ 5.2.2). *Participants* : Marc Shapiro, Fabio Riccardi, Xavier Blondel, Luciana Arantes, Syste Kloosterman.

6.7 Organisation de colloques et de cours

Journées Mémoire Virtuelle Répartie 96 (Bordeaux) dans le cadre du PRC « Parallélisme, Réseaux, Systèmes ». *Co-organisateur* : Marc Shapiro. *Membre du comité de programme* : Paulo Ferreira.

OSDI 96 (Operating Systems Design and Implementation), **POS 96** (Persistent Object Systems), **TINA 96** (Telecommunications Intelligent Network Architecture). *Membre des comités de programme* : Marc Shapiro.

Unité « Systèmes distribués de traitement d'information », Institut Supérieur de Technologie et de Management. *Coordinateur des enseignements de l'unité* : Mesaac Makpangou.

ERSADS (European Research Seminar in Advanced Distributed Systems), *co-organisateur* : Marc Shapiro.

WWW5 (Fifth World-Wide Web Conference), *chairman des workshops* : Patrick Duval.

7 Publications

Thèses

[440] P. FERREIRA, *Larchant: ramasse-miettes dans une mémoire partagée répartie avec persistance par atteignabilité*, Thèse de doctorat, Université Paris 6, Pierre et Marie Curie, Paris (France), mai 1996.

[441] J. MAISONNEUVE, *Hobbes : un modèle de liaison de réfénces réparties*, thèse de doctorat, Université Paris 6, Paris (France), octobre 1996.

Communications à des congrès, colloques, etc.

[442] A. BAGGIO, I. PIUMARTA, « Mobile Host Tracking and Ressource Discovery », *in : Seventh ACM SIGOPS European Workshop*, Connemara (Irlande), septembre 1996.

[443] A. BAGGIO, « Environnements mobiles: caractéristiques et problèmes », *in : Séminaires Action Scientifiques Systèmes et Applications Répartis, CNET*, Issy-les-Moulineaux (France), février 1996.

[444] P. FERREIRA, M. SHAPIRO, « Larchant: Persistence by Reachability in Distributed Shared Memory through Garbage Collection », *in : Proc. 16th Int. Conf. on Dist. Comp. Syst. (ICDCS)*, Hong Kong, mai 1996, http://www-sor.inria.fr/SOR/docs/LPRDSMGC_icdcs96.html.

[445] G. PIERRE, M. MAKPANGOU, « A Flexible Hybrid Concurrency Control Model for Collaborative Editing in Large Scale Settings », *in : Seventh ACM SIGOPS European Workshop*, Connemara (Irlande), septembre 1996.

Rapports de recherche et publications internes

[446] C. ADJIH, *Caractérisation des applications*, Mémoire de DEA, Université Paris-Sud, Orsay (France), septembre 1996.

[447] F. ALBRECHT, *Système d'annotations réparti: application aux documents web*, Mémoire de DEA, Université Versailles Saint-Quentin, Versailles (France), septembre 1996.

[448] S. DUGELAY, *Caches coopératifs pour le World-Wide Web*, Mémoire d'ingénieur, École Nationale des Télécommunications, Paris (France), juin 1996.

[449] F. L. FESSANT, *Portage d'applications sur Larchant*, Mémoire de stage de deuxième année, École Polytechnique, Palaiseau (France), juillet 1996.

[450] E. OTHUS, *Adaptation de l'outil OCI pour Larchant*, Mémoire de DESS, Université Paris 6, France, novembre 1996.

8 Abstract

INRIA Projet SOR (*Systèmes d'objets répartis,* or Distributed Object Systems) studies operating system support for information sharing in large-scale distributed systems. The long-term goals of our research are: (i) distributed garbage collection; (ii) applying distributed systems mechanisms in the large scale. Our targeted application domains are cooperative work in large scale environments (e.g. World-Wide Web) and mobile computing. We seek system-level solutions, i.e. ones that are general, mutually orthogonal, scalable, application-independent and language-independent.

Our current focus is on the three following issues. First, the persistent distributed store (PDS) abstraction supports a simple API that programmers are already familiar with. Shared objects are mapped in memory and then accessed via pointers. Persistent objects are those that are accessible from some persistent root. In the past year we proposed Larchant, a PDS which incorporates a novel garbage collection algorithm. This year we concentrated on three actions: a formal proof of safety and liveness and the refinement of the current prototype of Larchant, a post-compiler that extracts type information directly from compiled code, and a tool that extracts the layout of objects in memory such as to characterize relationship between objects.

Second, we provide an efficient and semantically-correct remote reference mechanism. Our Stub-Scion Pair (SSP) Chains are a fault-tolerant reference mechanism for identifying and accessing (possibly mobile and/or replicated) objects remotely. We now have a modular, robust, efficient implementation of the SSP chains. Attached to the reference management activity is the implementation of a flexible binding protocol, called Hobbes. We also extends the initial mechanism with new protocols to support mobile computing.

Third, in order to provide programmers with systems that really match their needs, we developped CORE, a toolbox of components implementing multiple replication abstractions. CORE allows application programmers to customize the internal policies of the underlying infrastructure. Our initial targets are fragmented object support, coherence management and storage systems.

Finally, we are developping a set of tools to support large scale cooperative work. These include a flexible distributed cache system, a cooperative bookmarks system, and a mobile proxy cache.