XtremWeb : une plate-forme de recherche sur le calcul global et pair à pair

F. Cappello, S. Djilali, Gilles Fedak, C. Germain, Oleg Lodygensky, V. Néri

XtremWeb : une plate-forme de recherche sur le Calcul Global et Pair à Pair

Franck Cappello — Abderrahmane Djilali — Gilles Fedak — Cécile Germain — Oleg Lodygensky — Vincent Néri

LRI, Université Paris Sud
F-91405 Orsay Cedex
fci@lri.fr

RÉSUMÉ. *La connection à grande échelle des ordinateurs aux réseaux de communication et les mécanismes qui permettent leur interopérabilité (standardisation des protocoles, homogénéisation des formats d'échange, systèmes de nomage efficace, sécurisation des sites) motivent des tentatives de globalisation des ressources et des données à partir de systèmes de Calcul Global, Pair à Pair ou de GRID. Ces systèmes sont les prémisses de l'émergence de systèmes d'exploitation à grande échelle. Dans cet article nous proposons d'abord de situer les systémes Pair à Pair par rapport aux autres structures de GRID en soulignant leur différences qui ont principalement une origine historique. Dans une deuxième partie nous présentons une rationalisation des systémes de Calcul Global et Pair à Pair. Nous détaillons ensuite la plate-forme XtremWeb qui constitue l'une des premières tentatives de généralisation du concept de parité entre les ressources appliqué à la globalisation du processeur, de la mémoire et du stockage. Nous terminons par la présentation de quelques points durs en recherche : la sécurisation des participants et la communication inter-noeud.*

ABSTRACT.

MOTS-CLÉS : *Calcul Global, Systèmes Pair à Pair, Architecture de systèmes distribués à grande échelle, Sécurité, Applications parallèles*

KEYWORDS: *Global Coputing, Peer-to-Peer Systems, Large Scale Distributed System Architecture, Security, Parallel Applications*

1. Introduction

La notion de GRID qui recouvre l'ensemble des dispositifs logiciels et matériels nécessaires pour l'exécution d'applications de grande taille sur plusieurs sites informatique et/ou sur une communauté de ressources (PC) est devenue indissociable du calcul numérique et du stockage à grande échelle. L'affluence rencontrée lors des récentes réunions du Global Grid Forum et du projet DataGRID montre la mesure de l'intérêt de la communauté internationale pour les GRID. Les expériences menées dans le cadre d'EuroGRID indiquent que les grands centres de calcul sont très actifs dans ce domaine et poussent les GRID vers la production. Le programme d'ACI GRID en France a permit de fédérer des équipes de recherche multi-sites autour de projets pluridisciplinaires avec des applications en physiques, biologie, géologie, des recherches fondamentales sur les mécanismes associés aux systèmes de GRID et des projets middleware d'envergure internationale.

Les projets étudiés et les infrastructures proposées émanent de différentes communautés et peuvent être classés en trois catégories : les systèmes de GRID, les systèmes de Calcul Global et les systèmes de partage de ressources entre Pairs (Peer-to-Peer).

Cette article examine plus particulièrement la problématique des systèmes de partage de ressources Pair à Pair appliqués au calcul. Nous commencons par situer les systèmes Pair à Pair par rapport aux systèmes de GRID traditionnels. Puis, nous proposons une rationalisation des systèmes Pair à Pair dans l'objectif de dégager leur caractéristiques principales. Nous présentons ensuite les composants principaux de ces systèmes en les illustrant par des exemples. La quatrième partie examine en détails le projet XtremWeb. La dernière partie considère des problèmes scientifiques durs relatifs au calcul et plus généralement à la globalisation des ressources à partir de systèmes Pair à Pair.

2. Systèmes Pair à Pair et GRID

Le Metacomputing [SMA 92] et les systèmes de GRID (Globalisation des Ressources Informatiques et des Données) [FOS 99], activement développés pour les applications scientifiques de calcul (EuroGRID) ou de stockage massif (DataGRID) ont pour ambition d'établir une infrastructure permettant 1) de faire interopérer des ressources de calcul, des grands instruments de mesure, des grandes bases de données et des centres de visualisation et de réalité virtuelle et 2) de facilité l'accès et l'organisation de ces ressources pour l'exécution d'applications scientifiques.

Les systèmes de Calcul Global (Global Computing) concernent aussi les applications scientifiques mais impliquent la participation de volontaires particuliers trouvant un intérêt social dans une expérience spécifique (lute contre le cancer, recherche pharmaceutique, recherche d'intelligence extraterrestre, généthon). Le principe du Calcul Global est différent de celui des GRID. Il s'agit d'utiliser un très grand nombre d'ordinateurs volontaires distribués géographiquement à l'échelle mondiale et communiquant uniquement par Internet pour exécuter des applications informatiques de

très grande taille. Le modèle de calcul est un parallélisme massif nécessitant peu de communications entre les sites. Le modèle d'exploitation est l'utilisation des ordinateurs d'institutions ou d'individus volontaires essentiellement pendant leurs périodes d'inactivité. Les systèmes de partage de ressources Pair à Pair ont, jusqu'à maintenant, concerné principalement la communauté des utilisateurs d'internet cherchant à échanger des documents multimédias (musiques, films, articles). Dans ces systèmes, toutes les machines du système peuvent remplir le rôle de client et de serveur ; d'où le terme de ServEnt dans Gnutella [KAN 01]. Les ressources partagées peuvent être de différentes natures : données, espaces de stockage, puissance CPU. Le rôle de l'infrastructure système est de mettre en relation directe les machines recherchant des ressources (les clients) et celles disposant des ressources recherchées (les serveurs), à un instant donné.

Dans un article récent [FOS 01], Ian Foster et al. suggèrent une fusion des systèmes de GRID, de Calcul Global et de Pair à pair. Il est très probable que les systèmes de Calcul Global et de Pair à pair vont fusionner simplement parce qu'ils reposent sur des contraintes identiques et ont recourt à des principes de fonctionnement similaires. La fusion avec les systèmes de Metacomputing (interconnexion de grands sites de calcul) est plus délicate à court terme essentiellement parce que les mécanismes à mettre en oeuvre dans les deux types de systèmes sont fondés sur des contraintes très différentes. Les deux sections suivantes présentes l'origine des différences entre ces systèmes dans une perspective historique.

2.1. *Les Grilles de calcul*

Les grilles de calcul classiques peuvent être considérées comme des extensions des grands centres de calcul nationaux ou de méso-informatique. Historiquement, les premières expériences dans ce domaines ont été réalisées lorsqu'il fut possible, aux États-Unis, de relier plusieurs grands sites de calcul grâce au réseau NFSNET [FOS 99] dont l'objectif était de construire et mettre en oeuvre un réseau haut débit (1,5 Mbis/s) reliant une douzaine de centres de calcul. Les réseaux Gigabit/s qui succédèrent à NFSNET et plus particulièrement vBNS furent l'occasion de lancer en 1995 un appel à projets pour démontrer la faisabilité d'applications réunissant une douzaine de laboratoires, l'utilisation d'un réseau haut débit et la visualisation par des environnements de réalité virtuelle. Les projets retenus furent regroupés dans le projet Iway. Ce projet avait la responsabilité de mettre en oeuvre l'infrastructure système permettant l'exécution des ces applications. Les infrastructures de grille de calcul actuelles descendent directement de cette progression incrémentale.

Il en résulte que globalement, les mécanismes systèmes mis en oeuvre sont des extensions des mécanismes existants : il s'agit d'autoriser l'accès et l'utilisation des sites à un ensemble d'utilisateurs, identifiés à l'avance. Par exemple, la sécurité repose sur des systèmes de login et de quota traditionnels et l'accès aux ressources est réalisé par l'intermédiaire de systèmes de batch avec des mécanismes de priorité pour répartir l'accès aux ressources suivant les droits des utilisateurs. L'ensemble d'outils système

Globus qui fournit ces mécanismes illustre parfaitement cette extension des principes utilisés dans les sites de calcul.

2.2. *Les systèmes de calcul global et les systèmes pair à pair*

Les premières expériences de Calcul Global ont eu lieu dans les années 90. Les domaines d'applications identifiés étaient essentiellement la cryptographie [DIS 97] [ADA], et certains problèmes mathématiques (nombres premiers de Mersenne [MER 97], calcul des décimales de Pi [PER 99]). Les infrastructures misent en oeuvre reposaient sur des protocoles comme le SMTP. Dans certains systèmes, les contributeurs devaient manuellement introduire de nouveaux paramètres de calcul dans l'application exécutée sur leur station de travail. Aucune mécanisme de sécurité n'était réellement mis en oeuvre et les contributeurs comme l'utilisateur des résultats se faisaient réciproquement confiance. Le projet SETI@home [AND 97], sans résoudre le problèmes de sécurité, a popularisé le concept de calcul à partir d'une fédération de PC volontaires connectés sur Internet.

Les fédérations de PCs ont fait émerger la notion de ressources distribuées et mutualisées, connue actuellement sous le terme de système Pair à Pair. En parallèle au Calcul Global, s'est donc développé une autre catégorie de systèmes distribués fondés sur le partage des ressources de stockage et la messagerie instantanée et non plus de calcul. Les projets initiateurs qui sont devenus vite populaire ont été Napster [PAN], Freenet [CLA 01] et Gnutella [SOL]. Dans ces systèmes, les ressources participantes fonctionnent à la fois comme des clients et des serveurs. Le système Napster a lui aussi démontré la faisabilité d'un système de mutualisation des ressources de stockage jusqu'à plusieurs millions de participants.

Les systèmes de Calcul Global et Pair à Pair se différencient des grilles de calcul principalement par quatre caractéristiques : 1) le nombre de ressources connecté est plusieurs ordres de grandeur plus grand (typiquement 100 000 ressources) et les ressources sont rarement parallèles (biprocesseur au manimum), 2) les ressources sont extrêmement volatiles, 3) les réseaux qui connectent les ressources sont des LAN, des Intranets et Internet et 4) les utilisateurs sont aussi en nombre très important (typiquement un utilisateur par ressource). Comparativement aux grilles de calcul, il faut composer avec une infrastructure matérielle déjà présente et évoluant de façon non coordonnée et en fonction d'impératifs sans relation évidente avec le calcul à grande échelle (les PCs sont plutôt conçus pour la bureautique et les jeux et les réseaux véhiculent les applications classiques d'Internet : mail, web, flux vidéo, audio).

L'ordre de grandeur et l'impossibilité d'action ou d'influence sur l'infrastructure engendre des problèmes spécifiques qui ne se présentent pas dans le cas des grilles de calcul. Ainsi 1) la sécurité doit être fondée sur des mécanismes extensibles à plusieurs centaines de milliers d'utilisateurs et de ressources et 2) le placement et l'ordonnancement des tâches doivent prendre en compte l'évolution à court terme (la connexion et déconnexion des ressources) et l'évolution à long terme (modification d'infrastruc-

ture) du système. Globalement, les techniques adaptées dans le cadre des grilles de calcul sont inapplicables dans le cadre du calcul à très grande échelle. Les fonctions traditionnellement associées à la procédure de login (identification, sélection des utilisateurs, droits d'accès, priorité d'exécution, protection du site et traçage des opérations) ne passent pas à l'échelle. Inversement, la confiance de l'utilisateur envers les résultats renvoyés et la facturation de l'utilisation des ressources ne peut pas reposer, comme dans le cas des grilles de calcul, sur le caractère institutionnel des sites accédés. Les sites de calcul sont des machines quelconques appartenant à des utilisateurs auxquels on ne peut pas accorder à priori une confiance absolue. Le calcul à très grande échelle repose donc sur la capacité de certifier les résultats ou d'être capable de discerner les bons et les mauvais résultats. De même, contrairement au cas des grilles de calcul, l'ordonnancement d'un très grand nombre de tâches sur une centaine de milliers de ressources ne peut être géré à long terme et coordonné à court terme par des mécanismes centralisés.

Ainsi, les GRID et les systèmes Pair à Pair ont des racines historiques très différentes qui expliquent leur différence d'organisation et de problématique propre. L'étude de la fusion de ces systèmes est une question ouverte. La spécification OGSA (Open Grid Services Architecture) proposé par le laboratoire d'Argonne comme servant de base à la version 3 de Globus est une première tentative dans cette direction.

3. Un état de l'art des plates-formes de Calcul Global

A la fin des années 90, plusieurs projets regroupés sous le terme de "Calcul basé sur Internet" (Web-based Computing) comme JET [PED 97], Bayanihan [SAR 98], SuperWeb [ALE 97], Javelin [NEA 99], Popcorn [NIS 98] ou Charlotte [BAR 96] ont vu le jour. Ces projets proposaient des environnements de développement pour les applications sur Internet. Le terme *Global Computing* était alors déjà utilisé pour représenter ces systèmes. La plupart sont nés avec le langage Java et exploitent certaines de ses caractéristiques comme 1) un langage intermédiaire (bytecode) qui permet aux applications d'être portables sans recompilation et 2) d'exécuter les applications dans environnement sécurisé pour la machine participante. Pour participer à une application distribuée, un utilisateur visite simplement une page Web dans laquelle est embarquée l'applet de calcul. Dans le principe, plusieurs de ces projets proposaient une vision du calcul sur Internet de type Pair à Pair dans laquelle toute machine offrant ses moyens de calcul pouvait aussi soumettre des requêtes de calcul. Le projet SuperWeb [ALE 97] avait identifié de nombreux problèmes scientifiques et techniques qui redeviennent d'actualité avec l'interêt actuel pour les systèmes de Calcul Global. Toutefois, plusieurs questions n'avaient pas été abordées comme l'exécution de code natif, la prise en compte de l'extrême volatilité des ressources notamment dans le contexte des communications entre les PCs participants.

Le lancement officiel du projet SETI@home est très récent (Avril 1999) mais le projet et le développement du code remonte à 1997. Avant son lancement officiel, le projet avait reçu environ 400 000 préinscriptions. Le dernier recensement indique 3

millions de machines participantes ou ayant participé à ce projet. SETI@home a démontré plusieurs points : 1) il est possible de fédérer une communauté de plusieurs centaines de milliers d'utilisateurs pour faire du calcul numérique, 2) le système développé soutient une production de calcul considérable (20 Teraflops) ; rappelons que cette valeur n'est pas directement comparable aux performances des machines parallèles classiques, 3) l'organisation de serveur de calcul et du collecteur de résultats permettent de satisfaire plusieurs millions de tâches par jour. Le projet a déjà produit l'équivalent 500000 années de calcul de PC. Notons toutefois que la limite à l'approche centralisée de SETI@home est le réseau. Deux éléments montrent la vulnérabilité d'un tel système : l'expérience SETI@home utilise à elle seule la moitié de la bande passante réseau du campus de Berkeley (information obtenue lors d'un entretien avec David Anderson) ; la coupure mécanique (vandalisme) de la fibre optique reliant les serveurs de l'expérience au campus a provoqué l'interruption d'activité pendant plusieurs dizaines d'heures.

Ces résultats ont ouvert la voie à des projets académiques et industriels. Il faut noter que la plupart des projets visent la mise en place de plate-formes de Calcul Global. Il existe actuellement dans le monde plusieurs dizaines de projets de ce type. Les plus connus sur le plan industriel sont Entropia, Parabon, United Devices, Popular Power. Ces projets sont multi-applications. Dans le monde académique, les projets sont souvent dédiés à une application (Xpulsar, Evolution, Distributed.net) ou visent la généralisation du concept de Calcul Global au Pair to Pair (COSM). Les projets académiques mono-application ont comme objectif principal la production de calcul. Ces plates-formes peuvent être considérées comme des spécialisations de plates-formes de Calcul Global généralistes. Leur spécialisation permet de résoudre beaucoup des problèmes de sécurité que nous détaillerons par la suite. Les projets industriels ne dévoilent ni leur architecture ni les mécanismes utilisés pour la sécurité des ressources. Certains projets commerciaux et la plupart des projets académiques généralistes proposent la description du protocole de communication entre les ressources et offrent le code source des programmes fonctionnant sur les ressources. En revanche, l'architecture générale et notamment les logiciels qui ne sont pas liés aux ressources ne sont pas dévoilés. Ces plates-formes ne peuvent donc pas être utilisées comme base pour conduire des expériences.

Comme nous l'avons évoqué précédemment, les systèmes de Calcul Global peuvent être considérés comme des extensions à l'échelle d'Internet du principe de vol de cycles. Sur les réseaux locaux, le vol de cycles a déjà été étudié dans les projets tels que Condor [LIT 88], Glunix [GHO 98] et Mosix [BAR 93]. Mais les contextes des réseaux locaux et Internet sont radicalement différents. Les techniques d'ordonnancement [AID 98, ROS 99] doivent aussi être adaptées à un environnement de Calcul Global en raison : 1) du nombre de ressources mis en jeu, 2) de l'extrême volatilité des ressources. Le projet Condor propose une évolution de son système pour les ressources connectées sur Internet. Ainsi Condor/G est une adaptation de Condor par dessus les mécanismes offerts par Globus.

Toutes les plates-formes de calcul global évoquées ci-dessus et toutes celles qui sont étudiées actuellement reposent sur une organisation centralisée. Des logiciels commercialisés par les sociétés Entropia, United Devices et Platform permettent néanmoins aux participants de soumettre des calcul.

Pour comprendre les relations possibles entre les systèmes de calcul global et les systèmes pair à pair, il est important d'examiner les concepts des ces systèmes et comment ils pourraient s'articuler dans un système de calcul global. Toutefois, la littérature et les annonces sur Internet fourmillent de propositions, de projets, de groupes de discussions autour du terme Peer-to-Peer (Pair à Pair). Pour un novice, il est très difficile de distinguer entre les argumentaires idéologiques et les conceptes fondamentaux. Un système Pair à Pair doit-il être fondamentalement totalement distribué ? L'anonymat est-il un principe discriminant les vrais systèmes Pair à Pair ? l'auto organisation et l'absence d'un point central (de points centraux) est-elle une caractéristique incontournable des systèmes Pair à Pair. Pour répondre à ces questions nous allons tenter de rationaliser l'approche des système Pair à Pair à travers une classification, la description des composants fondamentaux de ces systèmes et l'examen des technologies proposées.

4. Une classification des systémes Pair à Pair

Il n'existe pas de consensus sur la définition des systèmes Pair à Pair. C'est sans doute la marque d'un phénomène émergent très prometeur mais c'est aussi ce qui est à la base de ce manque de rationalité.

Pour mieux identifier et comprendre les propriétés de ces systèmes, nous proposons de les classer suivant différents paramètres discriminants. Il est intéressant de constater dans la littérature que les systèmes de Calcul Global sont considérés comme une forme de système Pair à Pair [Bar01]. Pour présenter les distinctions entre ces systèmes, nous intégrons dans notre classification les systèmes de Calcul Global.

Dans tous les cas, un système Pair à Pair suppose que les machines collaborent à un but commun qui peut être un calcul et/ou le stockage/partage d'information.

Le premier paramètre discriminant concerne la propriété pour une même machine de fonctionner alternativement ou simultanément comme un client et/ou un serveur du système. Les systèmes de Calcul Global comme SETI@home, Entropia, UnitedDevice ne respectent pas cette propriété contrairement à XtremWeb (http://www.xtremweb.net/), ActiveCluster de Platform et à tous les systèmes d'échange de fichiers Pair à Pair connus.

L'une des caractéristiques de certains systèmes Pair à Pair est, qu'une fois l'étape de mise en relation terminée, la transmission de données est réalisée sans intermédiaire, entre pairs. Ce n'est pas le cas pour FreeNet [LAN 01] ou la transmission de l'information est réalisée par l'infrastructure de mise en relation. Cette propriété est aussi absente dans les systèmes de Calcul Global. L'une des questions ouvertes sur

les systèmes de Calcul Global Pair à Pair concerne justement l'intérêt d'établir une connexion directe entre client et serveur de calcul.

Certains systèmes comme SETI@home, Napster et XtremWeb utilisent une architecture centralisée. Dans Naptser, le répertoire qui permet à un client d'identifier les serveurs potentiels est centralisé. Dans SETI@home et XtremWeb, l'ordonnanceur de calcul et le collecteur de résultats sont centralisés. A l'inverse, les architectures Pair à Pair récentes comme Gnutella, Freenet et Fastrack utilisent une infrastructure (backbone) distribuée pour remplir la fonction de mise en relation des clients et des serveurs.

La capacité d'auto-organisation du système est quelques fois associée au principe de communauté dynamique des systèmes de Calcul Global et des systèmes Pair à Pair. Cette propriété fait référence à la possibilité pour tous noeuds de rejoindre ou de quitter à tout moment le système. Du point de vue infrastructure système, l'auto-organisation signifie la propriété de reconfiguration dynamique et totalement distribuée du système en fonction du nombre et des caractéristiques des noeuds qui composent le système. Par exemple, FastTrack et Gnutella permettent à certains noeuds de jouer le rôle de supernoeuds qui serviront de répertoires de recherche pour améliorer les performances du système. L'introduction de caches dans le système permet aussi d'exploiter les propriétés de localité spatiale et temporelle des accès aux documents.

Le tableau suivant résume la classification des systèmes de Calcul Global et Pair à Pair. Le type d'opérations réalisées est mentionné par un C pour calcul et D pour données.

	Collaboration à un but commun	Rôle Client et serveur	Connexion directe	Infrastructure distribuée	Sysème auto organisé
Calcul Global	Oui / C				
XtremWeb I	Oui / C	Oui			
Napster	Oui / D	Oui	Oui		
Jabber	Oui / D	Oui	Non	Hiérarchique	
Freenet	Oui / D	Oui	Non	Oui	Oui
FastTrack	Oui / D	Oui	Oui	Hiérarchique	Super-noeud
Gnutella	Oui / D	Oui	Oui	Oui	Oui
P2P calcul	Oui / C+D	Oui	?	?	?

TAB. 1.

4.1. *Eléments fondamentaux d'architecture P2P par l'exemple*

A la base, Un système Pair à Pair est composé d'un système de recherche de ressources (ou système de mise en relation) et d'un système de transport de documents (lorsque l'application vise à communiquer des documents). Des éléments supplémentaires interviennent dans les systèmes récents comme les noeuds d'indexation. Enfin,

un système Pair à Pair doit fonctionner même lorsque des machines du système sont protégées par des pare-feux. Nous présentons ici ces éléments fondamentaux à partir d'exemples choisis d'architectures. Le système de recherche de ressources (ou système de mise en relation). Le système de recherche de ressources (ou de mise en relation) sert de mécanisme de recherche et d'identification d'un pair capable de fournir la ressource recherchée. La figure 1 présente les systèmes de recherche de ressources de Napster et de Gnutella dans sa version initiale.

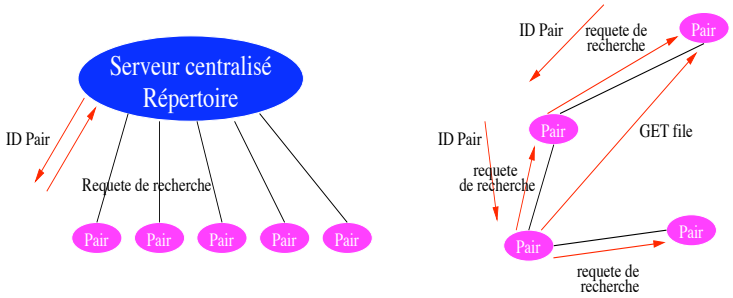


FIG. 1. *à gauche, l'architecture de Napster et à droite celle du système Gnutella (dans sa version initiale).*

Dans Napster le système de recherche de ressources est centralisé et repose sur un répertoire stockant des relations ressource-pair. Lorsqu'un pair recherche une ressource, il obtient auprès du répertoire les adresses IP de pairs disposants de la ressource.

Le système de recherche de ressources de Gnutella fonctionne d'une façon totalement distribuée. Un pair n'est connecté qu'à un ensemble limité de pairs voisins. Il s'agit de voisins logiques identifiés uniquement par leur adresse IP. L'ensemble des connexions forme une topologie logique qui n'a pas de correspondance avec la topologie physique d'Internet. Le système de mise en relation de Gnutella est construit pardessus cette topologie logique et fonctionne par diffusion en utilisant le protocole TCP. Un pair qui recherche un document (ou un service) diffuse une requête à ses voisins dans la topologie logique. La requête est propagée par les voisins à leurs propres voisins. Le protocole agit par inondation des noeuds (pairs) présents dans le système. Si un pair qui reçoit la requête possède le document recherché, il ne propage plus la requête et envoi une réponse à destination de l'émetteur de la requête, en indiquant son adresse IP. La réponse suit le parcours inverse de la requête en se propageant de Pair en Pair jusqu'au destinataire. Cette méthode fonctionne car tous les pairs propageant une requête retiennent l'origine et la destination (voisins immédiats) par lesquels la requête est passée. Gnutella possède un mécanisme pour limiter l'inondation : chaque requête possède une durée de vie (un compteur de sauts) qui est décrémentée à chaque fois qu'un pair propage la requête. Un mécanisme permet aussi d'éviter les boucles dans le protocole d'inondation.

Le mécanisme de recherche de ressources de FreeNet fonctionne aussi de manière totalement distribuée, selon une organisation différente de celle de Gnutella. Comme dans Gnutella, un noeud FreeNet ne connaît que des voisins logiques. Cependant, dans FreeNet toute ressource est cryptée et identifiable à partir d'une clé unique. Un noeud FreeNet stoke des relations Clés- adresses et le système de propagation de requêtes ne transmet la requête qu'au voisin possédant une clé proche de la clé de la ressource recherchée. Une clé peut représenter le cryptage du contenu de la ressource entière ou d'un ensemble de mots clés identifiant la ressource. Le système de transport de ressources. Dans Napster comme dans Gnutella, lorsque l'émetteur de la requête reçoit une réponse, il contacte directement le pair possédant le document recherché en utilisant l'adresse IP contenue dans la réponse. Dans Gnutella, le protocole de transfert consiste en une requête http (get) émise par le pair recherchant le document. Le pair qui possède le document reçoit la requête http sur un numéro de port différent de 80 et y répond en transmettant le document. FreeNet a été conçu pour minimiser la possibilité d'identifier les machines clientes et serveurs de documents. Il n'utilise pas de système de connexion directe pour éviter que le pair émetteur de la requête obtienne l'adresse IP du serveur. Le document est transmit au destinataire par le système recherche de ressource lui-même qui devient aussi le système de transport. En fait, chaque noeud FreeNet fonctionne comme un cache pouvant stocker dans chaque relation clé-adresse, la ressource correspondante. Le système de cache ne retient que les ressources les plus recherchées si bien que le temps de recherche d'une ressource souvent référencée est très rapide. En revanche les ressources les moins recherchées ont peu de chance de se trouver dans le système de caches et demandent plus de temps de recherche. Des mécanismes de transports plus sophistiqués peuvent être mis en oeuvre pour faciliter l'échange de fichiers très volumineux comme des films. Par exemple, Edonkey2000 l'émetteur de la requête peut solliciter plusieurs machines possédant un document (ou un segment du document) pour paralléliser le téléchargement. Il demande à chaque machine un segment différent du document. Dès qu'il a lui même reçu un premier segment, il peut servir de serveur pour d'autres machines recherchant ce segment du document.

Dans le protocole compatible Gnutella proposé par la société Clip2, chaque super-Pair agit comme un mini serveur Napster ; c'est à dire qu'il stocke un index des ressources et des pairs possédants ces ressources. Ainsi, au lieu de propager une requête, le super-Pair, s'il trouve une référence de la ressource recherchée dans son index, répond directement à l'émetteur de la requête. Les super-Pairs modifient uniquement le système de mise en relation. Le mécanisme de transport reste le même. Le système FasTrack fonctionne de manière analogue. L'un des problèmes liés à l'introduction de super-Pairs dans un système comme Gnutella est la cohérence des informations stockées dans l'index avec la réalité. En principe un super-Pair doit mettre à jour son index lorsqu'une machine disposant d'une ressource se retire du système et lorsqu'une machine ne dispose plus d'une ressource. La figure 2 présente les systèmes d'indexation de Clip2 pour Gnutella et de FastTrack avec les super-noeuds.

Dans tous les systèmes Pair à Pair, se pose le problème des pare feux qui protègent les pairs connectés au système. Ce problème concerne d'abord le système de

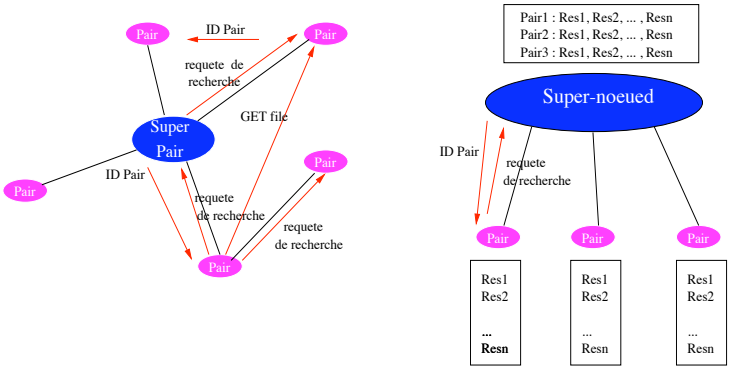


FIG. 2. *A gauche un super-Pair du système Clip2 et à droite un super-noeud du système FastTrack.*

mise en relation. En effet, pour mettre en oeuvre le système de mise en relation, il est nécessaire que des pairs acceptent des connexions externes. Si aucun pair n'accepte de connexion externe ou si leur nombre est insuffisant, le système ne pourra pas être mis en oeuvre ou sera très limité en performance. Le problème des pare feux se pose aussi pour le transport du document. Il arrive très souvent que le possesseur du document soit protégé par un pare feu. Dans ce cas, la requête http de l'émetteur de la requête sera bloquée (port fermé). Pour contourner le problème Gnutella prévoit un mécanisme qui permet à l'émetteur de la requête de recontacter le possesseur du document par le système de mise en relation en lui demandant d'exécuter lui-même une requête http (put) en direction du destinataire. Ce mécanisme appelé « push » inverse les rôles pour la transmission du document. Bien évidemment si l'émetteur de la requête de document est lui-même protégé par un pare feu, la transmission ne sera pas possible. C'est là une des limites intrinsèques des systèmes de transport de documents par connexion directe. La technique de transport utilisée dans FreeNet permet d'éviter, en partie, le problème des pares feux puisque les pairs ne communiquent par directement mais par l'intermédiaire du système de mise en relation.

4.2. *Technologies P2P*

Pour construire un système Pair à Pair, il existe plusieurs points de départ suivant le type de réalisations visées. Pour simplifier, nous pouvons considérer trois types de réalisations possibles : créer un protocole de système Pair à Pair, créer un système distribué Pair à Pair, créer une application reposant sur une infrastructure Pair à Pair.

Pour créer un protocole de système Pair à Pair, les technologies logicielles sont à la base celles associées à l'internet : protocoles TCP, http. Certains systèmes ont été construits pardessus le protocole de mail classique SMTP. Rapidement, le mode d'interaction par appel de procédures distantes s'est avéré comme particulièrement

adapté et des projets comme XtremWeb ont été construits par dessus Java RMI ou JINI. Depuis les premières expériences et les premiers systèmes, d'autres protocoles ont été proposés comme SOAP et XML RPC. SOAP (Simple Object Access Protocol) qui encapsule des RPC encodés au format XML dans des messages http (notons que SOAP n'est pas limité au protocole http) est retenu par beaucoup de projets comme protocole de base pour le développement. Indépendamment du protocole et du mode d'interaction, le système de mise en relation avec les mécanismes évoqués comme la durée de vie d'une requête, l'évitement de boucle dans le protocole de diffusion, le cache, etc. doivent être construits de toute pièce par le développeur. Il est à noter que le Peer-to-Peer Working Group, notamment soutenu par Intel, entend développer des technologies de base pour faciliter le développement de systèmes Pair à Pair en abordant par exemple les problèmes de la traduction des adresses IP (NAT) et des pare-feux, de la sécurité. Actuellement une bibliothèque de sécurisation des communications fondée sur OpenSSL est mise à disposition.

Pour simplifier la tâche du développeur, des environnements sont proposés permettant de construire des systèmes Pair à Pair plus facilement. Ainsi Jxta de SUN propose de structurer un système autour de 4 éléments : A) les « Peer pipes », mécanisme de connexion d'un pair à un autre et de partage d'informations sur le réseau et de manière distribuée, B) « les Peer groups » qui permettent la création de groupes de façon dynamique et le regroupement logique et cohérent de contenu, C) la possibilité de surveiller et de mesurer les interactions et de définir des politiques de contrôle entre pairs (Peer monitoring) et D) des mécanismes de sécurité permettant de garantir la confidentialité, l'identité et l'accès contrôlé aux services. Le projet COSM propose aussi de construire des environnements Pair à Pair plus simplement qu'en utilisant les protocoles de communication. Les projets Folding@home et Genome@home utilisent COSM comme plate-forme de base.

Le niveau supérieur dans la hiérarchie des environnements pour le développement de systèmes Pair à Pair consiste en des plates-formes complètes et opérationnelles dans lesquelles il « suffit » d'intégrer l'application désirée. FastTrack et XtremWeb proposent des plates-formes adaptées respectivement pour les applications d'échanges de documents et de calcul distribué. FastTrack est une plate-forme industrielle payante alors qu'XtremWeb est une plate-forme de recherche à code source ouvert. Tous les deux proposent des outils de configuration et d'administration permettant à l'administrateur d'installer simplement, de mettre en oeuvre et de contrôler le système. Ils incluent aussi les mécanismes de diffusion des programmes clients et offrent des interfaces utilisateurs.

Les problèmes que nous avons évoqués jusqu'à ce stade concernent globalement les applications d'échange de documents et de calcul Pair à Pair. Dans le cadre du calcul global, les systèmes Pair à Pair peuvent intervenir à plusieurs titres : a) permettre a tous serveurs d'agir aussi comme un client, b) mettre en oeuvre un système de mise en relation hiérarchique ou totalement distribué, c) mettre en oeuvre un mécanisme de transport de résultat offrant des propriété de cache, d'anonymat ou d'envois segmentés. Au delà, l'association du calcul global et des concepts Pair à Pair engendrent

de nouvelles questions comme la sécurité des participants et des applications, la circulation des données cohérente avec le placement des calculs et l'ordonnancement des tâches, l'exécution d'applications parallèles avec communication entre les participants.

Le projet XtremWeb, présenté dans la section suivante, examine cette problématique. Sa première version ne reprend des systèmes Pair à Pair que la possibilité pour tous serveurs d'agir aussi comme un client. Les recherches menées sur XtremWeb dans le cadre du projet d'ACI GRID CGP2P traitent les autres questions dont certaines sont présentées en détails dans la dernière section.

5. Le système de Calcul Global XtremWeb

Le projet XtremWeb vise à la conception et le développement d'un environnement de Calcul Global et de calcul Pair à Pair académique et pluri-disciplinaires.

L'environnement XtremWeb est une plate-forme généraliste, sécurisée et orientée hautes performances de calcul. La conception d'XtremWeb est organisée selon une perspective de dissémination de la technologie logicielle qui sera développée dans le projet. Ces logiciels devraient permettre à de centres de recherche, des Universités, des Ecoles et des Industriels d'installer et d'utiliser leur propre système de Calcul Global et calcul Pair à Pair pour leurs travaux de recherche ou pour la production de calcul.

L'un des objectifs de la plate-forme est d'imposer le moins de contraintes possibles sur l'environnement à mettre en oeuvre autour d'XtremWeb et les applications à exécuter sur XtremWeb. Généraliste signifie qu'à la différence de projets comme SETI@HOME, XtremWeb n'est pas dédié à une application particulière, mais configurable pour une classe d'applications.

Le système est construit uniquement à partir de standards éprouvés et des logiciels libres de type *open source* comme les langages C++, Java, le langage de scripts PERL, le protocole XML RPC, le gestionnaire de base de données MySQL et le serveur Web Apache. L'utilisateur d'XtremWeb peut ainsi compter sur un accès facile à ces logiciels et une pérénité du système.

L'orientation hautes performances d'XtremWeb signifie que l'on vise les applications du calcul intensif (entier ou flottant). Cela ne signifie pas qu'une plate-forme XtremWeb est comparable directement avec les ordinateurs recensés dans le top 500 par exemple. Cette comparaison n'est pas possible puisque a) le nombre de ressources considérées est beaucoup plus grand pour un système de Calcul Global, b) que les ressources sont par essence volatiles (peuvent se déconnecter à tout instant) et c) les performances de communication sont très faibles.

5.1. *Architecture d'XtremWeb*

XtremWeb est conçu pour fournir un environnement de Calcul Global pour la résolution de différentes applications sur des projets d'institutions, d'entreprises commerciales ou de communautés *open source*. La figure 3 donne un aperçu des différentes utilisations envisagées d'un système de Calcul Global comme XtremWeb.

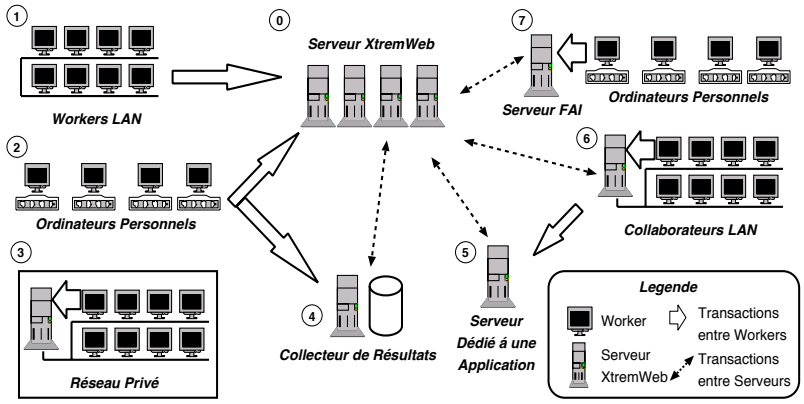


FIG. 3. *Vue d'ensemble d'un environnement de Calcul Global comme XtremWeb*

Des PCs volontaires collaborent avec le serveur XtremWeb (0). Ce sont soit des fermes de PCs appartenant à des institutions avec une haute connexion à Internet (1) soit des PCs familiaux avec une connexion intermittente (2). XtremWeb permet aux collaborateurs de mettre en place leur propre expérience de Calcul Global (5, 6) pour distribuer leurs applications. Le système lui-même est assez flexible pour que des serveurs soient dédiés à certaines tâches comme la collection des résultats (4). Dans une approche plus spéculative, on peut aussi penser à des serveurs qui collaborent (6) ou marchandent (7) pour agréger leurs ressources. Finalement une institution peut aussi mettre en place son *Intranet XtremWeb* (3) sans lien vers l'extérieur.

5.2. *Organisation générale*

L'architecture générale d'XtremWeb est centralisée, dans le sens où un serveur, qui peut être distribué, organise les calculs sur des ressources distantes géographiquement distribuées ainsi que les échanges d'informations entre ces ressources. A la différence de systèmes comme Gnutella ou Freenet, le système n'apparaît pas identique quel que soit le point de vue que l'on prenne : point de vue du serveur ou point de vue des ressources. Les ressources sont nommées Workers dans la suite.

Dans la première version d'XtremWeb, le serveur centralisé joue aussi le rôle de collecteur de résultats. Dans des versions ultérieures pour des systèmes à grande échelle, il sera nécessaire d'étudier la séparation des deux fonctions : serveur de tâches

et collecteur de résultats ainsi que la réalisation de ces deux fonctions par des serveurs répartis.

L'organisation centralisée qui semble pertinente dans le cas du Calcul Global avec un fonctionnement Maître-Esclaves puisque c'est la solution prise par tous les grands projets du domaines (SETI@home, Entropia, United Devices, Parabon, etc.) peut sembler inappropriée dans le cas du mode d'interaction Pair à Pair (Peer to Peer). En effet, les ressources étant sensées remplir tour à tour le rôle de serveur et de ressources, il semblerait plus efficace de distribuer totalement le système pour éviter les goulots d'étranglement.

5.3. *La plate-forme XtremWeb*

Le projet XtremWeb, comme tous les environnements de Calcul Global, doit satisfaire des contraintes sévères :

– Extensibilité : le système doit être extensible jusqu'à des centaines de milliers de noeuds avec une augmentation de performance correspondante,

– Hétérogénéité : les machines possèdent des configurations matérielles et systèmes très variées,

– Dynamicité : le système doit être capable de s'accommoder à une configuration qui varie constamment ainsi qu'à des communications dont la latence et le débit varient aussi,

– Disponibilité : le propriétaire d'une ressource doit pouvoir définir une politique qui limite la contribution de sa ressource, et le système de Calcul Global, doit respecter cette politique,

– Tolérance aux pannes : une défaillance du serveur de calcul ou du collecteur de résultats ne doit pas avoir d'incidence sur les machines de calcul. La perte d'une machine participante ou d'un ensemble de machines participantes est un événement intrinsèque du système,

– "Utilisabilité" : le système doit rester simple à utiliser, à déployer et à maintenir,

– Sécurité : comme tout environnement connecté à Internet, un système de Calcul Global doit assurer la sécurité du serveur, des machines de calcul et de l'application elle même contre les attaques.

Il convient de pondérer cette liste car pour la plupart des points, des solutions sont connues.

L'extensibilité des performances de calcul et la dynamicité de l'environnement sont deux contraintes connues dans d'autres cadres comme dans les architectures parallèles, les bases de données sur Internet et plus généralement dans les systèmes distribués. Dans XtremWeb, l'hétérogénéité ne constitue pas un sujet de recherche en soit. En effet, XtremWeb s'adresse principalement à des PCs sous Linux ou Windows.

En revanche, le travail d'adaptation d'XtremWeb aux différentes versions de ces systèmes est nécessaire.

La disponibilité est réglée de façon habituelle en intégrant XtremWeb sur les machines de calcul comme un pseudo économiseur d'écran. XtremWeb n'entre en action que lorsque la machine est inutilisée. Toutefois, l'encombrement mémoire d'une application peut provoquer une gène lors du retour de l'utilisateur. En effet, si l'application occupe toute la mémoire physique de la machine, les pages de mémoire virtuelle des applications de l'utilisateur seront stockées temporairement sur disque. Lors de son retour, l'utilisateur percevra un délai qui correspond au rétablissement en mémoire physique des pages de ses applications. Il faut donc comme dans le projet Glunix veiller à n'occuper qu'une part raisonnable de la mémoire physique (le projet Glunix a montré que cette part dépend de la notion de contrat social entre l'utilisateur et le système).

Les serveurs de calcul et les collecteurs de résultats des systèmes de Calcul Global reçoivent leurs interactions d'Internet. Leur fonctionnement s'apparente à celui des serveurs Web. Or, la tolérance aux pannes est un domaine théorique bien connu dans le domaine des serveurs Web. Dans le cas d'XtremWeb, nous souhaitons dépendre le moins possible de solutions propriétaires. Des solutions provenant du logiciel libre "open source" seront donc examinées et une étude spécifique pourra être menée le cas échéant.

S'il existe des solutions pour plusieurs des contraintes évoquées, les performances et la sécurité représentent des problèmes difficiles dans le cadre des systèmes de Calcul Global et Pair à Pair.

Deux modes sont possibles concernant l'exécution d'applications sur les ressources participantes. Le premier consiste à exécuter l'application sous la forme d'un byte code Java. Jet [PED 97] et Bayanihan [SAR 98] sont deux projets qui correspondent à ce mode. Il y a deux intérets : 1) la sécurité et 2) la portabilité. L'inconvénient majeur de cette approche réside dans les performances trop faibles d'exécution de l'application en comparaison d'un code binaire.

Le deuxième mode consiste à exécuter du code natif. Entropia, Distributed.net, United Devices, SETI@home, GIMPS et Bovine exécutent des programmes natifs. Les performances sont comparables à celles d'un programme s'exécutant localement puisque le code exécutable est directement celui généré par un compilateur C ou Fortran. La portabilité est assurée en préparant plusieurs versions de la même application adaptée à des machines et des systèmes d'exploitation différents. Le point critique dans cette approche est la sécurité.

La sécurité concerne les trois entités qui forment une exécution XtremWeb : le serveur, la ressource et l'application. Comme indiqué précédemment, les problèmes de sécurité et de hautes performances sont liés. La sécurité des serveurs d'applications connectés sur Internet est un sujet vaste et relativement bien connu. Les problèmes spécifiques du Calcul Global concernent la ressource et l'application.

XtremWeb doit assurer la sécurité des ressources participantes puisqu'elles sont connectées sur Internet. Les machines participantes ne doivent pas être corrompues involontairement ou volontairement par l'environnement XtremWeb ou par un programme se faisant passer pour lui. La sécurité de la ressource peut être prise en défaut lors du déroulement des protocoles de communication avec le serveur et lors de l'exécution du code natif d'une nouvelle application. La sécurisation des communications entre les ressources et le serveur fait appel aux techniques actuelles (cryptage des données, mécanisme d'authentification par clé publique et clé privée).

Le deuxième point de recherche sur la sécurité concerne l'intégrité de l'application. Une des difficultés rencontrées par les systèmes de Calcul Global est la certification des résultats d'exécution renvoyés vers le collecteur de résultats. L'identification des partenaires de la communication est réalisée par un système de clés classique. Ceci n'empêche pas une ressource participante de retourner ponctuellement des données erronées.

La sécurité des machines participantes et de l'application est examinée dans la partie consacrée à la recherche dans les systèmes pair à pair.

5.4. *Architecture du Worker*

Le Worker a deux principales fonctionnalités : fournir des ressources de la machine d'un utilisateur pour l'exécution d'un calcul XtremWeb et exécuter l'application fournie par le serveur. Pour qu'un projet de Calcul Global soit largement accepté auprès du public, il est nécessaire que le logiciel Worker assure au mieux le respect de l'utilisateur, c'est à dire qu'il soit configurable, non-intrusif et sécurisé.

L'utilisateur décide quand XtremWeb peut lancer une exécution et quelles sont les ressources (CPU, taille mémoire, espace disque) que cette exécution peut obtenir. La disponibilité d'une machine dépend 1) de la présence d'un utilisateur (détectée par l'activité clavier/souris) 2) de l'existence de tâches non- interactives (détectée par l'usage du processeur, de la mémoire ou des E/S) et 3) d'autres conditions telles que le jour et la nuit par exemple. L'utilisateur définit une politique de disponibilité en indiquant pour chaque ressource des seuils qui provoquent le lancement ou l'interruption de la participation au calcul global. Un cas particulier est la ressource réseau, qui est directement gérée par la couche de communication. Le worker s'adapte à une déconnection temporaire en sauvant dans une file d'attente les communications à effectuer. Lorsque le worker peut à nouveau contacter le serveur, il reprend les communications en attente. Nous appelons ce mode "calcul off-line".

La protection de l'ordinateur d'un utilisateur suggère que le calcul d'une tâche s'effectue dans un environnement virtuel sécurisé, typiquement la machine virtuelle Java. Néanmoins deux raisons incitent au support de l'exécution de code natif. D'abord beaucoup d'applications scientifiques utilisent des codes matures écrits en C ou Fortran, ensuite les exigences de performance imposent que le code soit natif à la plateforme d'exécution de l'utilisateur. Bien sûr, le code ne peut pas être testé sur tous

les paramètres possibles en particulier sur ceux susceptibles de contenir des attaques par dépassement de tampon (buffer overflow). L'exécution sur la machine du Worker doit donc être confinée dans un environnement où les opérations systèmes sont contrôlées [GOL 96b] pour interdire/autoriser la lecture ou l'écriture de fichiers ou de sockets, et empêcher l'exécution de processus. XtremWeb supporte le mécanisme de confinement d'exécution Subterfugue.

Le Worker identifie le serveur en utilisant la phase d'authentification. Une fois cette phase terminée, toutes les communications sont encryptées y compris le transfert du code au Worker. Avant d'exécuter un code, le Worker renvoie une signature du fichier binaire et le serveur vérifie qu'elle est identique à l'original. Enfin les résultats sont aussi encryptés avant d'être rapatriés vers le serveur de façon à prévenir l'altération des résultats par un tiers. En revanche, il est difficile de garantir que les résultats n'ont pas pu être modifiés par le Worker lui même. Dans ce cas, il appartient aux institutions de fournir par application des filtres ou des méthodes statistiques pour éliminer ou atténuer l'influence de résultats manipulés.

5.5. *Implémentation du Worker*

La figure 4 montre l'architecture du Worker. L'architecture s'organise autour d'une *réserve de travail* qui est vidée et remplie par un gestionnaire de communication et un exécuteur de tâches. Cette réserve contient les descriptions des tâches téléchargées par le worker et peut être sérialisée sur le disque de façon à ce que le worker puisse reprendre son travail aprés une interruption. Le gestionnaire de communication et l'exécuteur de travail sont implémentés de manière multithead et accèdent à la réserve de travail selon un model producteur/consommateur. Cela permet d'implémenter le recouvrement des calculs par les communications (éventuellement plusieurs communications simultannées) et le support des machines multiprocesseurs en lancant un exécuteur de tâches par processeurs. D'autre part un thread d'arrière plan s'exécutant avec une basse priorité scrute l'activité de l'ordinateur pour appliquer la politique de disponibilité. Lorsque l'ordinateur devient disponible il permet à l'exécuteur de tâche de reprendre une tâche interrompue ou lancer une nouvelle tâche. Lorsque le thread d'observation détecte une augmentation de la charge du système (en excluant celle induite par sa propre activité) ou la présence d'un utilisateur, il ordonne à l'exécuteur de tâche d'arréter ou de suspendre la tâche en cours.

Le Worker XtremWeb est principalement écrit en Java, les appels à des fonctionnalités spécifiques du système d'exploitation étant écrites en C et intégrées via la Java Native Interface. Le choix du langage Java permet au noyau du Worker XtremWeb d'être facilement porté sur différentes architectures. Actuellement le Worker est disponible en téléchargement pour Linux sur Intel x86 et Windows; et il est en projet de le porter sur Windows Pocket pour les agendas électroniques.

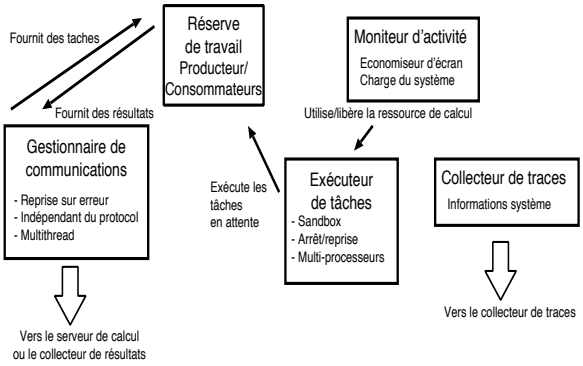


FIG. 4. *Architecture du Worker*

5.6. *Architecture des Serveurs*

Les serveurs XtremWeb sont responsables du support des Applications Globales en 1) stockant les fichiers binaires des applications pré-compilés pour différentes architectures et des tâches soumises par les clients et 2) en fournissant les tâches et les applications aux Workers désirant calculer.

Distribuer une application consiste à fournir un binaire pré-compilé pour différentes architectures et une description de l'application. La description de l'application comprend son nom, la manière dont l'application prend en compte ses paramètres et la forme que prennent ses résultats. Il est aussi possible de fournir différentes versions de l'application, la mise à jour est automatique sur le serveur.

Un client soumet une tâche en fournissant la référence de l'application et l'ensemble des paramètres qui définissent une tâche. Les paramètres peuvent être : différents fichiers, un fichier lu sur l'entrée standard de l'application ou des arguments sur la ligne de commandes.

Lorsqu'une tâche est enregistrée dans la base de données, elle reçoit un unique *identifiant de tâche*. Cet *identifiant de tâche* réfère la tâche pour toutes les opérations suivantes d'ordonnancement et de récupération des résultats. Un module de surveillance contient les informations sur l'ensemble des tâches telles que : a) les machines ayant traité la tâche ou la liste des machines si la tâche a été interrompue, b) le client qui a soumis la tâche et c) différentes prises de temps. Les utilisateurs et administrateurs d'XtremWeb peuvent interagir avec le système à travers une interface Web pour soumettre des tâches, suivre la progression des tâches, récupérer les résultats. Les utilisateurs peuvent obtenir des statistiques sur l'activité des machines participant à XtremWeb en tant que Worker.

5.6.1. *Répartition des tâches*

Le placement des tâches sur les Worker est effectué en deux étapes : la sélection des tâches et le placement des tâches. La sélection des tâches à partir de la liste des tâches, remplie par les requêtes des clients, se fait en fonction des applications prioritaires. La priorité est déterminée en attribuant des proportions de tâches à accomplir par application. Les tâches sont placées selon une méthode FIFO. Lorsqu'un Worker demande une tâche, le serveur sélectionne d'abord la tâche qui peut être exécutée par ce Worker particulier selon son environnement d'exécution et l'existence d'un binaire précompilé de l'application. Cette approche correspond au mode d'ordonnancement *eager* .

Le serveur détecte aussi les tâches abandonnées en détectant des dépassements de délai et les attribue à d'autres Workers si nécessaire. Les politiques de sélection et de placement des tâches sont configurables dynamiquement. De nouvelles politiques peuvent être définies (e.g. LIFO ou d'autres tenant compte des critères spécifiques de l'application et des ressources), et échangées en cours d'exécution.

5.6.2. *Implémentation*

L'implémentation de serveur utilise essentiellement des logiciels libres et le langage Java, disponibles sur la plupart des systèmes d'exploitation. Le logiciel de base de données utilisé est MySQL et les langages utilisés (Java, PHP et Perl) fournissent des accès généralisés aux bases de données (comme Java DataBase Connectivity ou Perl Database Interface). Changer de logiciel de base de données requiert peu de modifications du code.

5.7. *Protocole de communication entre Workers et serveurs*

Toutes les communications sont à l'initiative du Worker. Ceci facilite le déploiement en regard des protections réseau (firewall) qui pourraient bloquer les requêtes provenant de serveurs situés à l'extérieur d'un domaine d'administration.

Le protocole entre les Workers et le serveur est indépendant de la couche de communication. Celle-ci peut être généraliste comme les sockets TCP-UDP/IP, de plus haut niveau comme Java RMI (Remote Method Invocation) ou spécialisée comme SSL (Secure Socket Layer). Un Worker est une machine identifiée par son nom et son propriétaire. Lors de l'inscription, le serveur vérifie que le nom du propriétaire n'est pas déjà utilisé. Le nom de machine permet au même propriétaire d'inscrire plusieurs machines. L'unicité du nom de machine par rapport à un propriétaire est vérifiée lors de l'enregistrement de la machine.

La figure 5 présente le protocole entre un Worker et un serveur.

Le protocole consiste en quatre requêtes détaillées ci-dessous:

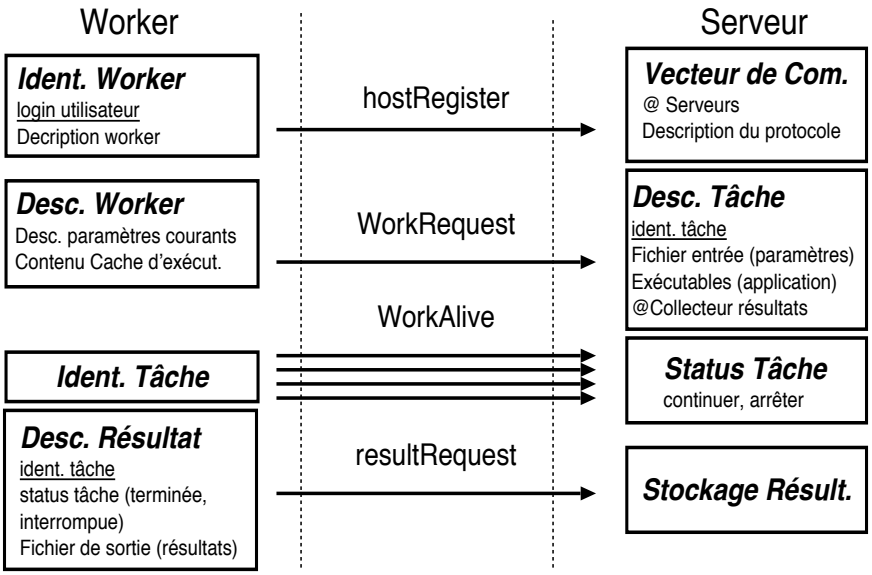


FIG. 5. *Le protocole entre le serveur et un Worker*

– La première requête *hostRegister* s'effectue vers le dernier serveur contacté ou vers le serveur racine du système. Cette première connexion authentifie le serveur au Worker. Le serveur renvoie un *vecteur de communication* qui spécifie la liste des serveurs susceptibles de fournir des tâches au Worker ainsi que la couche de communication (protocoles et ports) par laquelle ils peuvent être contactés. Dans le cas le plus simple, le serveur retourne son adresse.

– Puis le Worker demande une tâche au serveur à travers la requête *workRequest*. Le Worker fournit une description de son environnement d'exécution (système d'exploitation, architecture, etc.) et la liste des applications précédemment téléchargées et cachées dans un répertoire local. Selon cette information, le serveur sélectionne une tâche et renvoie au Worker une description de la tâche, les paramètres d'entrée de la tâche, le fichier binaire de l'application correspondant à l'environnement d'exécution du Worker si nécessaire et l'adresse d'un serveur capable de stocker les résultats.

– Durant tout le calcul, le Worker invoque périodiquement *workAlive* pour signaler son activité au serveur. Le serveur scrute ces appels en permanence pour implémenter un protocole de *timeout* . Si un Worker n'a pas appelé depuis un temps suffisamment long, le Worker est considéré indisponible et sa tâche peut être réattribuée à un autre Worker. En retour, le serveur envoie des messages de contrôles pour la tâche en cours : poursuite du calcul, arrêt du calcul, changement d'identification de la tâche; et des messages de contrôle pour le Worker : mise en état actif/inactif.

– A la fin du calcul, le Worker renvoie les résultats à l'adresse spécifiée (au collecteur de résultats ou au serveur principal si les deux fonctions sont agrégées), à travers l'appel *workResult*. Cet appel est dupliqué vers le serveur qui a fourni la tâche de façon à signaler l'aboutissement du travail, dans le cas ou le serveur de tâches et le collecteur de résultats sont deux entités distinctes.

Actuellement le protocole est implémenté sur les RMI Java et sur SSL.

5.8. *Une API Client pour XtremWeb*

L'API Client est une interface pour programmer des applications qui peuvent s'exécuter sur la plate-forme XtremWeb. Cette API permet à l'utilisateur du système de dialoguer avec le serveur pour spécifier un environnement d'exécution souhaité (nombre de Workers), envoyer des tâches vers le serveur et récupérer les resultats des tâches soumises. Comme pour le Worker, et pour les mêmes raisons de sécurité, les communications sont à l'initiative du client. Client et Woker peuvent fonctionner sur les mêmes machines.

Les principales fonctionnalités fournies par cette API sont :

La gestion de l'environnement d'exécution : ceci consiste à demander au serveur un nombre de Workers pour exécuter l'application. Ceci peut être étendu pour prendre en compte d'autre requêtes : types de machines, durée d'exécution total, durée d'exécution par machine etc.

La gestion de l'exécution : ceci consiste en quelques fonctions de base permettant de :

– soumettre une tâche au serveur,

– demander l'état d'avancement d'une tâche,

– récupérer le résultat d'exécution d'une tâche,

– arrêter l'exécution d'une tâche.

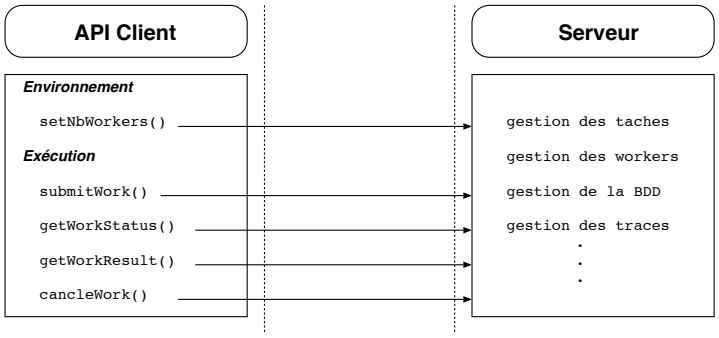


FIG. 6. *Fonctions de l'API Client*

Cette interface a déjà été utilisée pour programmer des applications de type maître-esclave. Ces applications ne nécessitent pas de communications entre les différentes tâches s'exécutant en parallèle. Un exemple de ce type d'applications est le programme test EP (Embarasingly Parallel) du benchmark NAS. EP ne nécessite aucune communication inter-processus; il s'agit d'une boucle qui parcours un vecteur. Cette boucle peut être parallélisée facilement, et ne nécessite qu'une seule communication globale (une réduction) en fin d'exécution. L'adaptation de ce programme à notre plate-forme, est réalisée à partir de la version parallèle NPB 2.3 écrite avec MPI (Message Passing Interface). Chaque processus MPI du programme EP NPB 2.3 est traduit en une tâche du programme XtremWeb. La tâche maître crée les tâches esclaves et les envoie au serveur qui les achemine à son tour vers les Workers disponibles. A la fin de l'exécution, la tâche maître récupère tout les résultats des tâches exécutées et effectue une opération de réduction.

Dans ce même cadre, d'autres types d'applications devraient être adaptées à XtremWeb : Out-of-Core, Branch-and-Bound, Application Parallèles communicantes, résolution de très grand problèmes creux, etc.

5.9. *Le moniteur XtremWeb*

L'administration du système XtremWeb consiste à contrôler la plate-forme complète (serveur et Workers) et à obtenir des informations sur leur fonctionnement (prise de traces d'activité, temps passé dans les différents mécanismes du serveur).

5.9.1. *Réglages du serveur.*

Quatre paramètres sont réglables au niveau du serveur :

– expected Workers permet de définir le nombre de Workers maximum qui seront utilisés

– trace frequency : ce paramètre, envoyé au serveur qui le transmettra à tous les Workers, est le nombre de secondes entre deux prises de traces,

– nb traces in trace file : ce paramètre, envoyé au serveur qui le transmettra à tous les Workers, est le nombre de traces par fichier.

L'application de contrôle du serveur permet de gérer la liste des adresses IP qui sont autorisées à se connecter au serveur. Cette liste concerne toutes les connexions au serveur : Worker, client et monitor lui-même.

5.9.2. *Réglages des Workers.*

Le réglage des Workers concerne principalement la liste des informations sur les Workers connectés au serveur ; cette liste est construite au lancement de l'application. Ces informations sur les Worker sont : identifiant du Worker, adresses IP et Mac, time zone, os type, cpu type, user name, alive, active et traces.

Alive indique que le Worker est connecté et répond correctement. Active indique qu'il est susceptible de recevoir des taches à exécuter. Cette fonctionnalité est contrôlable. L'administrateur peut décider d'arrêter immédiatement le Worker et de plus lui soumettre des tâches comme illustré par la figure 7. Traces indique que le Worker prend des traces d'activité. L'administrateur peut décider d' arrêter la prise de traces et d'ordonner le transfert des traces contenant les dernières prises au serveur.

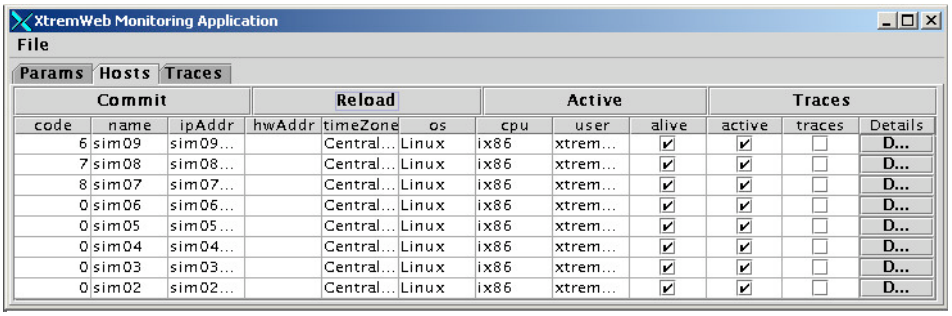


FIG. 7. *Interface de réglages des Workers.*

5.9.3. *Prise de traces.*

Dans le cadre du Calcul Global, il nous semble intéressant de pouvoir suivre, comprendre et connaître le comportement de machines candidates. En affinant ces connaissances, nous serons non seulement en mesure de simuler un système de Calcul Global avec plus de réalisme, mais nos prédictions quant aux capacités d'un tel système en seront d'autant plus justes.

Dans XtremWeb, les traces rassemblent les informations de taux d'activité des différentes parties des machines exécutant un Worker (cpu, mémoire, disques, interfaces réseau). Les traces sont prises à intervalle régulier et sont envoyées au serveur XtremWeb qui les stocke pour analyse ultérieure.

Ces traces dépendent de la machine sur laquelle le Worker est installé. Elles sont spécifiques au système d'exploitation. Pour la version Linux, ces traces sont lues dans */proc*. Elles comprennent les informations suivantes: heure de prise des traces, taux d'occupation CPU (user, nice; système, idle), nombre de process, mémoire (disponible, partagée ; nombre de buffers), taux de mémoire cachée, quantité de swap.

Les traces sont stokées sur le serveur et accessible par le client. Une première application permet l'affichage du taux CPU pour chaque Worker. Des outils d'analyse plus complexes peuvent être construits à partir des traces stockées sur le serveur.

5.10. *XtremWeb 1.0*

Un prototype d'XtremWeb est fonctionnel depuis Octobre 2000. Le serveur de tâches est situé au LRI comme le serveur de résultats. Les Workers sont des machines du LRI et des machines externes situées en dehors de l'Université. Les Workers exécutent le programme AIRES du projet Auger (voir section 7). En Novembre 2000, environ 80 machines étaient enregistrées comme Workers XtremWeb. Les machines connectées sont de type PCs. XtremWeb admet deux systèmes d'exploitation sur les Workers (Linux et Windows 98 et 2000). Des tests ont été effectués pour valider la procédure d'installation générique depuis des sites distants en Australie, USA et Europe. Le résultat du calcul effectué sur les Workers est renvoyé au serveur de résultats, mais il n'est pas stocké dans la version actuelle. Dans le cas du projet Auger, c'est l'interfaçage avec les autres éléments informatiques du projet qui déterminera le mode de stockage des résultats. XtremWeb est visible sur le site www.xtremweb.net

La première version *open source* d'XtremWeb a été rendue disponible au mois d'Avril 2001. L'ensemble de logiciels mis à disposition comprend le serveur de tâches/collecteur de résultats, les Workers Linux et Windows, la configuration d'un site Web et en démonstration l'application de synthèse d'images par lancés de rayons PovRay. Une interface de soumission de tâches est intégrée au Worker de cette première version d'XtremWeb. Les utilisateurs du système complet peuvent donc expérimenter le mode de calcul Pair à Pair avec l'application PovRay.

La page d'accueil donne accès à une page système permettant de consulter des informations sur les Workers enregistrés et des statistiques d'exécutions correspondant au système XtremWeb rattaché au site considéré. L'écran suivant (figure 8) présente une information disponible sur la page système : le nombre d'heures de calcul réalisées par jour (sur un groupe de 10 machines).

L'écran suivant (figure 9) montre les machines enregistrées comme Workers pour le système XtremWeb du LRI. On remarque que plusieurs systèmes d'exploitation sont supportés : Linux et Windows.

L'écran suivant (figure 10) montre le nombre de tâches réalisées et détaille ces tâches. La durée moyenne des tâches est de 30 minutes sur les simulations d'essai. Sur les deux premiers mois d'essai et sur un groupe réduit de machines, environ 3500 heures de calcul ont été fournies. Ceci montre le potentiel de calcul important du Calcul Global.

Toutes ces informations et statistiques fournissent à l'administrateur du système XtremWeb des éléments pour l'administration (retrait de machines participantes, vérification de l'enregistrement et de l'activité de machines participantes, performance du système en heures de calcul et exécutions par jour etc.).

L'écran suivant (figure 11) montre l'interface de soumission de scènes utilisée sur le Worker XtremWeb dans le mode d'utilisation pair à pair de l'application PovRay.

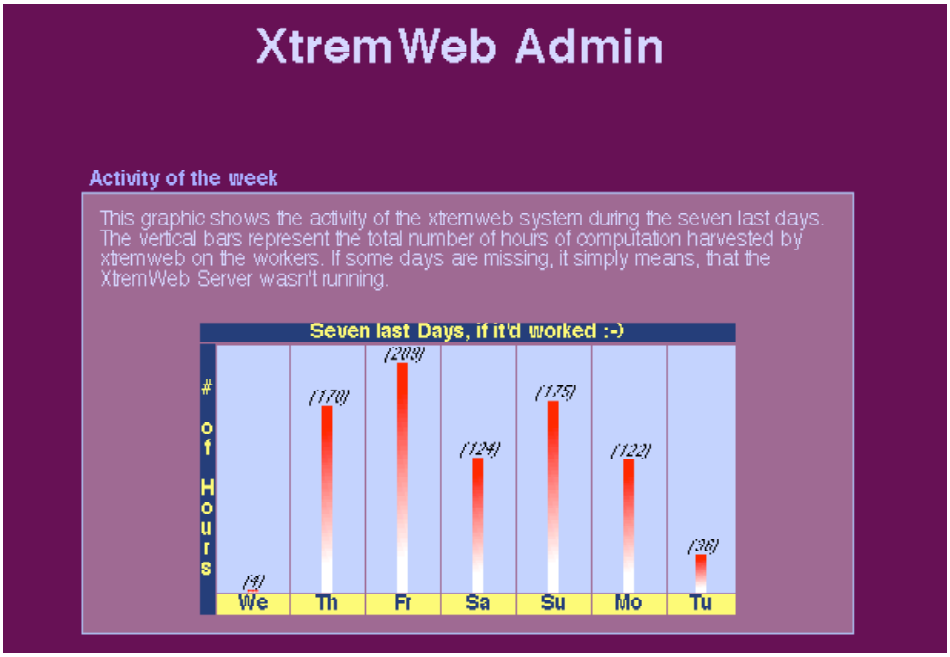


FIG. 8. *Représentation graphique du nombre d'heures de calcul réalisées par jour*

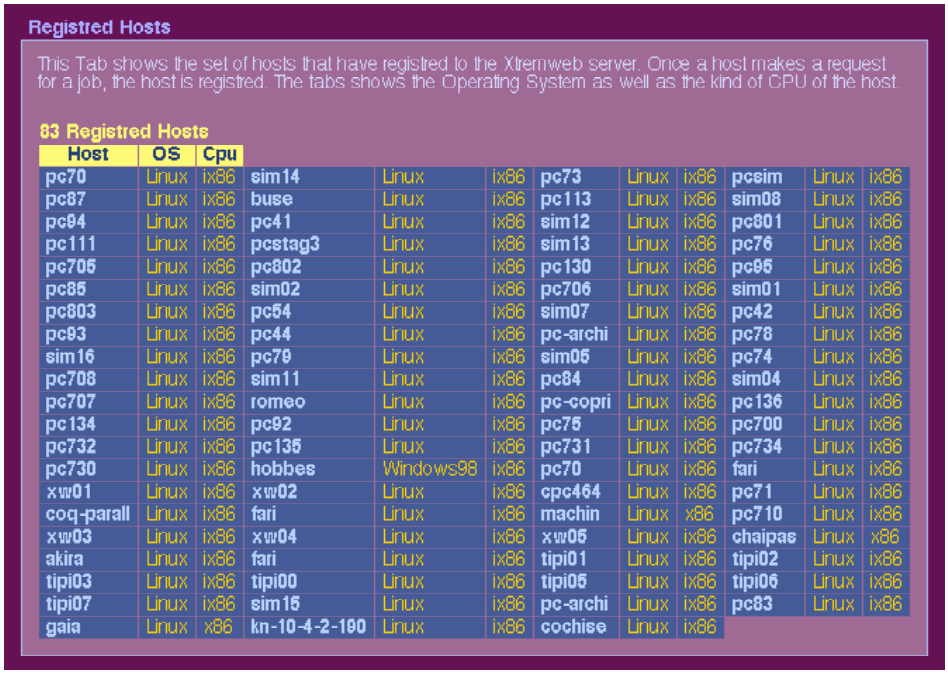
5.11. *Le projet Auger*

Le projet d'Observatoire Pierre Auger constitue la première application scientifique d'XtremWeb. L'objectif général du projet Auger est l'étude des particules de très haute énergie. Cette expérience implique des traitements informatiques dont certains relèvent du Calcul Global hautes performances.

5.11.1. *L'observatoire Pierre Auger*

L'Observatoire Pierre Auger est un effort international réunissant dix neuf pays, trente trois institutions, trois cents chercheurs avec un budget de $100M pour étudier les rayons cosmiques de très haute énergie, au-delà de 10^{19} eV. En fait, jusqu'à la détection fortuite de deux évènements au dessus de 10^{20} eV, la théorie ne prévoyait pas leur existence. L'origine de tels rayons cosmiques est complètement inconnue. C'est un des objectifs du projet que de déterminer si les sources de ces particules sont ponctuelles ou très dispersées dans l'univers.

De tels évènements sont extrêmement rares : au delà de 10^{20} eV, la fréquence d'apparition d'une particule sur un kilomètre carré étant de une par siècle. Les rayons cosmiques ne peuvent être directement observés depuis la surface de la terre. En revanche, lorsqu'une particule à haute énergie (primaire) entre dans l'atmosphère terrestre, les



Registred Hosts

This Tab shows the set of hosts that have registred to the Xtremweb server. Once a host makes a request for a job, the host is registred. The tabs shows the Operating System as well as the kind of CPU of the host

83 Registred Hosts

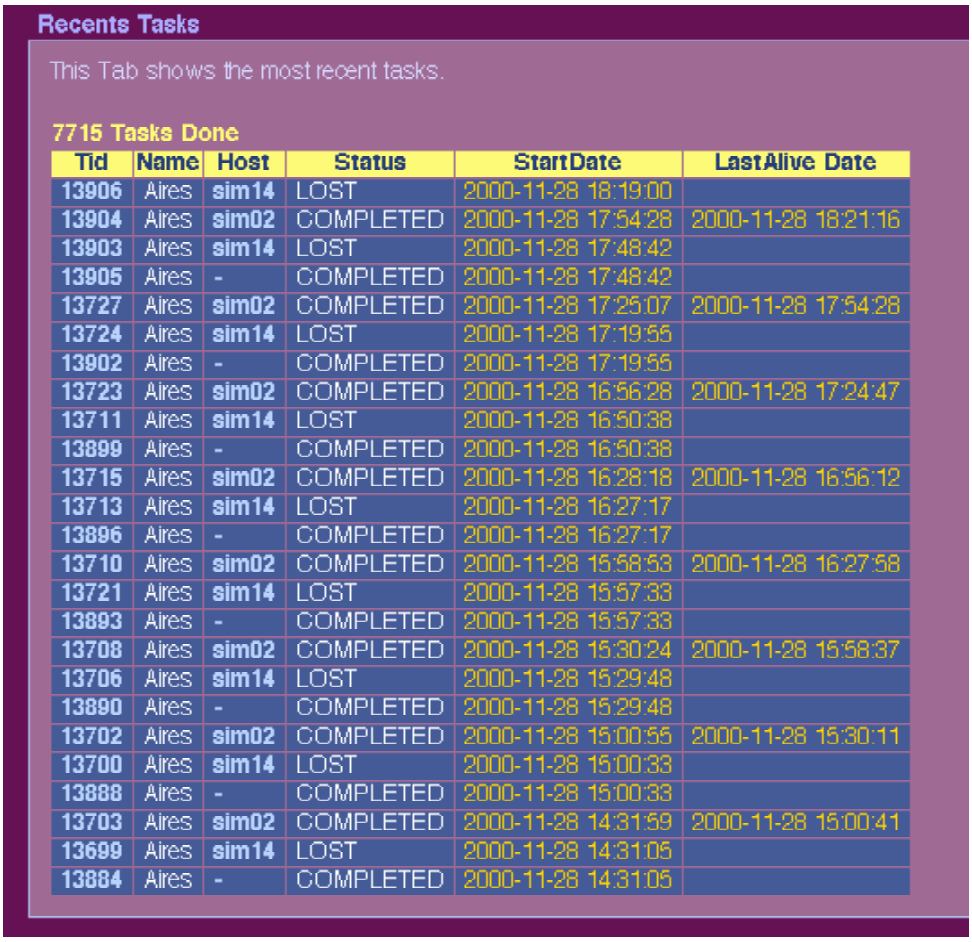
Host	OS	Cpu									
pc70	Linux	ix86	sim14	Linux	ix86	pc73	Linux	ix86	pcsim	Linux	ix86
pc87	Linux	ix86	buse	Linux	ix86	pc113	Linux	ix86	sim08	Linux	ix86
pc94	Linux	ix86	pc41	Linux	ix86	sim12	Linux	ix86	pc801	Linux	ix86
pc111	Linux	ix86	pcstag3	Linux	ix86	sim13	Linux	ix86	pc76	Linux	ix86
pc705	Linux	ix86	pc802	Linux	ix86	pc130	Linux	ix86	pc95	Linux	ix86
pc85	Linux	ix86	sim02	Linux	ix86	pc706	Linux	ix86	sim01	Linux	ix86
pc803	Linux	ix86	pc54	Linux	ix86	sim07	Linux	ix86	pc42	Linux	ix86
pc93	Linux	ix86	pc44	Linux	ix86	pc-archi	Linux	ix86	pc78	Linux	ix86
sim16	Linux	ix86	pc79	Linux	ix86	sim05	Linux	ix86	pc74	Linux	ix86
pc708	Linux	ix86	sim11	Linux	ix86	pc84	Linux	ix86	sim04	Linux	ix86
pc707	Linux	ix86	romeo	Linux	ix86	pc-copri	Linux	ix86	pc136	Linux	ix86
pc134	Linux	ix86	pc92	Linux	ix86	pc75	Linux	ix86	pc700	Linux	ix86
pc732	Linux	ix86	pc135	Linux	ix86	pc731	Linux	ix86	pc734	Linux	ix86
pc730	Linux	ix86	hobbes	Windows98	ix86	pc70	Linux	ix86	fari	Linux	ix86
xw01	Linux	ix86	xw02	Linux	ix86	cpc464	Linux	ix86	pc71	Linux	ix86
coq-parall	Linux	ix86	fari	Linux	ix86	machin	Linux	x86	pc710	Linux	ix86
xw03	Linux	ix86	xw04	Linux	ix86	xw05	Linux	ix86	chaipas	Linux	x86
akira	Linux	ix86	fari	Linux	ix86	tipi01	Linux	ix86	tipi02	Linux	ix86
tipi03	Linux	ix86	tipi00	Linux	ix86	tipi05	Linux	ix86	tipi06	Linux	ix86
tipi07	Linux	ix86	sim15	Linux	ix86	pc-archi	Linux	ix86	pc83	Linux	ix86
gaia	Linux	x86	kn-10-4-2-190	Linux	ix86	cochise	Linux	ix86			

FIG. 9. *Des machines enregistrées comme Worker pour le système XtremWeb du LRI*

collisions avec les molécules d'air créent des cascades de particules secondaires appelées *gerbes atmosphériques* qui sont observables au niveau du sol. Deux détecteurs géants, chacun d'une surface de 3000km^2, vont être construits un en Amérique du Nord et l'autre en Amérique du sud. L'objectif de ces détecteurs est de mesurer la direction, l'énergie et la composition des particules à haute énergie pendant plusieurs années (le projet est prévu pour une durée initiale de 20 ans).

Les gerbes atmosphériques doivent aussi être simulées numériquement, en particulier par le programme AIRES [SCI 95] (*Air Showers Extended Simulation*).

Les résultats simulés seront comparés aux observations pour déterminer les caractéristiques des particules primaires pendant toute la durée de l'expérience. Les données de la simulation numérique sont les paramètres physiques relatifs au contrôle de la simulation. Le résultat est la gerbe de particules simulée arrivant au niveau du sol. Le nombre de simulations indépendantes à exécuter est très grand : la simulation est basée sur un principe de Monte-Carlo, requérant de nombreuses exécutions avec les mêmes paramètres pour dégager des moyennes. Enfin des particules primaires avec différentes propriétés cinétiques et physiques doivent être simulées avec différents modèles physiques. Le besoin en puissance de calcul est d'au moins 10^6 heures équivalent PC à 300Mhz par an. A cette étape de notre travail et de l'expérience Auger, le



Recents Tasks

This Tab shows the most recent tasks.

7715 Tasks Done

Tid	Name	Host	Status	StartDate	LastAlive Date
13906	Aires	sim14	LOST	2000-11-28 18:19:00	
13904	Aires	sim02	COMPLETED	2000-11-28 17:54:28	2000-11-28 18:21:16
13903	Aires	sim14	LOST	2000-11-28 17:48:42	
13905	Aires	-	COMPLETED	2000-11-28 17:48:42	
13727	Aires	sim02	COMPLETED	2000-11-28 17:25:07	2000-11-28 17:54:28
13724	Aires	sim14	LOST	2000-11-28 17:19:55	
13902	Aires	-	COMPLETED	2000-11-28 17:19:55	
13723	Aires	sim02	COMPLETED	2000-11-28 16:56:28	2000-11-28 17:24:47
13711	Aires	sim14	LOST	2000-11-28 16:50:38	
13899	Aires	-	COMPLETED	2000-11-28 16:50:38	
13715	Aires	sim02	COMPLETED	2000-11-28 16:28:18	2000-11-28 16:56:12
13713	Aires	sim14	LOST	2000-11-28 16:27:17	
13896	Aires	-	COMPLETED	2000-11-28 16:27:17	
13710	Aires	sim02	COMPLETED	2000-11-28 15:58:53	2000-11-28 16:27:58
13721	Aires	sim14	LOST	2000-11-28 15:57:33	
13893	Aires	-	COMPLETED	2000-11-28 15:57:33	
13708	Aires	sim02	COMPLETED	2000-11-28 15:30:24	2000-11-28 15:58:37
13706	Aires	sim14	LOST	2000-11-28 15:29:48	
13890	Aires	-	COMPLETED	2000-11-28 15:29:48	
13702	Aires	sim02	COMPLETED	2000-11-28 15:00:55	2000-11-28 15:30:11
13700	Aires	sim14	LOST	2000-11-28 15:00:33	
13888	Aires	-	COMPLETED	2000-11-28 15:00:33	
13703	Aires	sim02	COMPLETED	2000-11-28 14:31:59	2000-11-28 15:00:41
13699	Aires	sim14	LOST	2000-11-28 14:31:05	
13884	Aires	-	COMPLETED	2000-11-28 14:31:05	

FIG. 10. *Informations sur les tâches réalisées*

projet XtremWeb n'est qu'une tentative de fournir des ressources complémentaires à la production classique par le calcul haute performance.

5.11.2. *Implémentation de AIRES sur XtremWeb*

Une exécution séquentielle de Aires dure entre 5 et 10 heures sur un PC 300Mhz, occupe un espace disque allant jusqu'à 100 Mo, produit un résultat de 10Mo et occupe en mémoire de l'ordre de 10Mo. Cette faible occupation mémoire s'explique par la présence dans le code de AIRES d'un mécanisme de sauvegarde temporaire sur disque. Ce mécanisme sert d'une part à réduire l'occupation mémoire en fonctionnant comme un système de pagination et d'autre part à sauvegarder des contextes

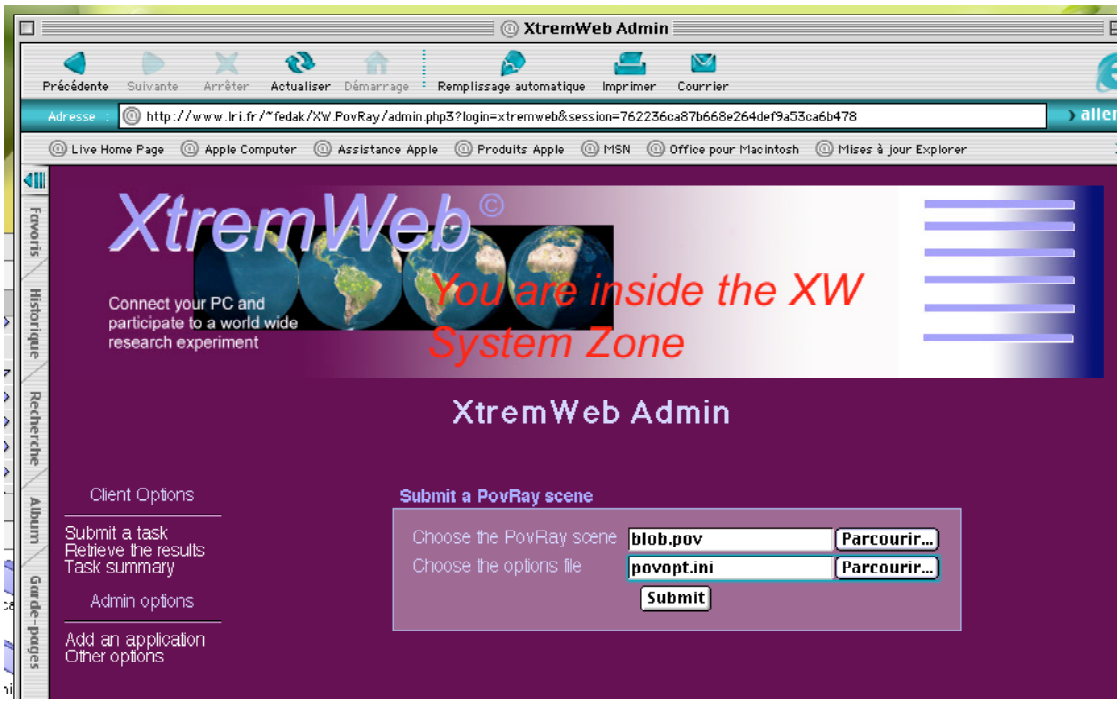


FIG. 11. *Interface de soumission de scènes dans le mode Pair à Pair d'XtremWeb*

intermédiaires de calcul permettant une reprise éventuelle du calcul en cas d'arrêt de l'application.

Le temps d'exécution d'AIRES dépend directement d'un paramètre d'affinage de la simulation. Des exécutions précises peuvent demander plus de 10 heures. Le temps d'exécution est prédit avec une précision raisonnable à partir des paramètres d'entrée et de la performance de la machine du Worker. Même en choisissant judicieusement les machines d'exécution en fonction de leur performance et de leur disponibilité, il existe une probabilité non négligeable qu'une machine cesse de calculer à n'importe quel moment de l'exécution d'une simulation. Nous pouvons supposer qu'un nombre significatif de longues simulations seront perdues si l'on conserve le mode de fonctionnement actuel qui perd complètement un calcul lorsque la machine est réutilisée par son utilisateur ou déconnectée du réseau.

6. Recherche dans les systèmes de Calcul Global P2P

Le projet XtremWeb du groupe Cluster et Grille dur LRI et le projet d'ACI GRID CGP2P (www.lri.fr/fci/CGP2P.html) abordent précisément les problèmes liés au cal-

cul Pair à Pair. Nous ne pouvons présenter ici que quelques uns. D'autres problèmes comme l'ordonnancement, la tolérance à la volatilité, les limites du domaine d'applications, l'interface utilisateur et l'aide à la décision sont tout aussi importants et ardus. Ceci souligne la richesse scientifique de ce thème de recherche.

6.1. *Besoin d'outils pour la validation*

L'engouement actuel pour les systèmes Pair à Pair se traduit par la naissance à un rythme élevé de protocoles, d'environnements de développement et de projets se référant aux systèmes Pair à Pair. Par exemple, le site SUN consacré à Jxta voit chaque mois augmenter le nombre de projets utilisant cette technologie comme base de développement. L'annonce de Plate-forme concernant son logiciel Active Cluster va sans doute mettre d'avantage l'accent sur ces systèmes et renforcer l'intérêt des industriels envers ces systèmes.

L'une des questions fondamentales actuelles, du point de vue scientifique mais aussi pratique, est l'efficacité de ses systèmes dans l'utilisation des ressources qu'ils mettent en oeuvre et leur résistance aux défaillances les plus diverses. En plus des pairs qui participent par principe à un système Pair à Pair d'autres ressources, comme les réseaux locaux, les réseaux métropolitains et les réseaux longue distance sont utilisées non plus seulement pour transporter des documents mais aussi pour mettre en relation les pairs d'un système. Des études sur le système Gnutella ont démontré l'utilisation excessive des réseaux par le système de mise en relation [RIP 01]. D'autres études ont montré la vulnérabilité de systèmes comme FreeNet pour les cas d'attaques ciblées sur les noeuds présentant le plus de connexions [HON 01]. Le projet SETI@home a rencontré plusieurs problèmes de défaillance. Le plus important a été la coupure accidentelle d'une partie du réseau de l'Université de Berkeley lors de travaux. Le serveur est resté déconnecter de ses millions de participants pendant quelques heures. Un autre type de défaillance à été relevé lorsque plusieurs participants ont commencé a retourner des résultats faux. L'optimisation de l'utilisation des ressources, de l'extensibilité du système, du dimensionnement de l'infrastructure, de la tolérance aux défaillances et de la résistance aux attaques ciblées du système Pair à Pair passe par une compréhension fine des phénomènes intervenants dans ces plates-formes. La modélisation des systèmes, leur simulation (ou leur émulation) sont des outils indispensables dans cette perspective.

Le problème majeur que rencontre la modélisation, la simulation et l'émulation des systèmes Pair à Pair est le nombre et la complexité des ressources à considérer. La dimension typique d'un système Pair à Pair est de plusieurs centaines de milliers de ressources. Est-il nécessaire de construire des modèles, des simulateurs et des émulateurs capables de prendre en compte les interactions de plusieurs centaines de milliers de ressources pour faire émerger les problèmes et trouver des solutions intéressantes ? En attendant de connaître un élément de réponse à cette question, beaucoup de chercheurs se posent la question de la construction de tels modèles, simulateurs ou émulateurs. Les tous prochains mois verront très certainement fleurir le nombre de ces

outils. Il existe déjà des prototypes ad-hoc, basés sur une reproduction très fidèle du comportement des réseaux sous-jacents. Ces simulateurs utilisent généralement NS (Network Simulator) et un générateur de topologies. D'autres simulateurs ne prennent pas en compte le réseau et s'intéressent à des phénomènes de seuils apparaissant dans les protocoles Pair à Pair, quelle que soit la topologie réelle. D'autres simulateurs s'intéressent à l'ordonnancement des tâches en prenant en compte des temps de calcul et de communication [CAS 00].

6.2. *Sécurité des participants*

La recherche de performance conduit à exécuter l'application sur les machines participantes dans le jeu d'instructions natif de la machine. Le problème de cette approche est la sécurité des participants. L'exécution d'un code binaire ne doit en aucun cas pouvoir conduire à une corruption de la machine ou de ses données.

Java apporte une solution de compromis à cette question. Sans compilation à la volée, le byte code Java est interprété par une machine virtuelle qui respecte la politique de sécurité par défaut ou imposée par le propriétaire de la machine. Pour améliorer les performances, certains environnements Java traduisent à la volée le byte code Java dans le format binaire de la machine d'exécution. Il est alors de la responsabilité du propriétaire de la machine de vérifier quelle politique de sécurité respectent ces environnements. L'un des problèmes liés à l'utilisation de Java réside dans la diversité des machines virtuelles existantes, de leur performance et de leur sécurité. En effet, il paraît difficile d'imposer à un participant une machine virtuelle précise. L'autre problème lié à l'utilisation de Java est que de nombreuses applications scientifiques ne sont pas écrites en Java.

Ces deux problèmes soulignent la nécessité de trouver une solution générale pour la protection des participants dans le cas de l'exécution de codes binaires résultats de la compilation de langages variés. La protection de la machine repose dans ce cas sur le système qui fonctionne sur cette machine. La problématique sous jacente est celle du « sandboxing »(enveloppement) de codes exécutables.

Il existe différentes voies pour parvenir à cet objectif. La première consiste à exécuter l'application en sécurisant le noyau du système d'exploitation de la machine d'exécution. Les systèmes Janus [GOL 96a] et Subterfugue reposent sur cette approche. Le système intercepte tous les appels systèmes réalisés par l'application, avant leur exécution. Ces appels sont analysés dynamiquement pour évaluer leur agressivité. Par exemple, on cherche à savoir si les fichiers visés sont bien dans l'espace de travail de l'application. Si ce n'est pas le cas l'application est stoppée.

Une autre approche, certainement la plus prometteuse, consiste à intégrer les dispositifs de sécurisé dans les fonctions du système d'exploitation. C'est ce que proposent les « Linux Security Modules » qui intègrent des points de vérification avant les sections sensibles des fonctions systèmes. Ces points de vérification prennent la forme de branchements, pris ou non selon la politique de sécurité, qui déroutent l'exé-

cution vers un module de sécurité que l'administrateur ajoute au noyau standard. Sans vérification, le coût des points de vérification est nul. Si un module de sécurité est ajouté, les performances des appels systèmes deviennent dépendants de la politique de sécurité et de son implémentation.

Il est important de noter que ces systèmes sont encore incomplètement maîtrisés.

6.3. *Certification de résultats*

L'une des problématiques complexes des systèmes de GRID (et en particulier des systèmes de calcul Pair à Pair) est la certification de résultats. Le problème provient de la possibilité qu'a un participant d'envoyer des résultats qui ne sont pas conformes à ceux prévus. Ces retours de résultats erronés peuvent être volontaires comme involontaires. Par exemple, dans le projet SETI@home, certains participants ont optimisé le calcul de la FFT qui constitue le coeur du calcul réalisé sur les machines participantes. Comme cette optimisation a été effectuée sans chercher à suivre des spécifications, le code optimisé ne répondait pas aux critères de qualité admissibles pour la FFT calculée sur les clients. Les codes optimisés ont donc envoyé des résultats faux.

Se prémunir contre des attaques volontaires ou involontaires sur les résultats de calcul est une question complexe. L'application ne peut pas compter sur les protocoles et la sécurisation des communications pour certifier un résultat. C'est le système ou l'application elle-même qui doivent offrir une parade à cette possibilité de corruption.

Il existe essentiellement deux approches possibles : l'approche système et l'approche application. L'approche système repose sur des techniques comme les exécutions redondantes avec vote et le test ponctuel. Une technique plus subtile qui en fait lie les deux précédentes consiste à attribuer une crédibilité à chaque résultat et aux machines fournissant les résultats. Selon cette technique, les ressources acquièrent une crédibilité de plus en plus grande à mesure que le nombre de vérifications ponctuelles validées augmente. De même les résultats acquièrent une crédibilité de plus en plus grande à mesure que le nombre de résultats concordants pour une même tâche augmente.

En complément de l'approche système, la certification de résultats peut être effectuée par l'application après réception du résultat. Plusieurs approches sont possibles suivants le type d'applications. Dans le cas ou le code source de l'application n'est pas ouvert, le cryptage des résultats au moment du calcul par l'application et avant leur stockage ou leur transmission permet de résoudre le problème simplement. Dans le cas général, où l'on admet que la machine effectuant le calcul dispose sous une forme ou sous un autre des sources de l'application et de la méthode de cryptage, le problème est beaucoup plus complexe et reste ouvert scientifiquement.

6.4. *Applications parallèles*

L'exécution d'applications parallèles sur Internet est tentante si l'on considère le nombre de ressources connectées. Plusieurs projets dont Javelin [NEA 99] et Bayanihan [SAR 01] ont examiné la possibilité de faire communiquer des applets Java et de construire des modèles de programmation par-dessus cette fonctionnalité. Notons que dans les deux cas, les échanges d'informations entre les applets passent systématiquement par un intermédiaire centralisé (le broker), ce qui n'est pas souhaitable dans le cas général.

Une des caractéristiques d'une organisation de type Calcul Global est l'extrême volatilité des PC participants. Cette caractéristique semble s'opposer aux principes qui sont à la base du modèle de programmation parallèle le plus utilisé pour les architectures parallèles à mémoire distribuée : le passage de messages explicites avec les fonctions d'envoi et réception classiques. En effet, dans un système de Calcul Global, il est difficile de garantir que les deux partenaires d'une communication seront encore actifs lorsque celle-ci sera exécutée.

Le principe de sites intermédiaires fonctionnant comme des mémoires associatives permet de concevoir un mécanisme de communication entre les machines participantes adapté à l'extrême volatilité des machines d'un système Pair à Pair. Une machine intermédiaire fiable joue le rôle de mémoire associative. Un élément de la mémoire associative est composé d'un indentifiant et d'une donnée. Pour communiquer l'émetteur dépose un message dans la mémoire associative en utilisant un identifiant. Le récepteur s'adresse à la mémoire associative pour lire le message en utilisant le même identifiant. Ce principe permet aux tâches communicantes d'être totalement asynchrones (le récepteur pouvant lire un message envoyé par l'émetteur ayant depuis cessé de participer). Dans le principe, l'émetteur et le récepteur peuvent être remplacés par d'autres machines en cas de défaillances puisque leurs actions sur l'environnement extérieur se résument aux accès en écriture ou en lecture à une mémoire associative qui, elle, est fiable. Des machines de remplacement accèderaient directement à la mémoire associative pour continuer le calcul.

Découpler les machines communicantes et sauvegarder les communications en cours dans la mémoire associative ne suffit pas. Il faut aussi mettre en oeuvre un mécanisme de reprise pour relancer globalement le calcul. Cette problématique est très complexe car les machines considérées ne sont connectées qu'à travers Internet ce qui rend en pratique déraisonnable la sauvegarde à distance de contexte d'exécution de plusieurs Mo. Sauvegarder localement (sur la machine elle même) le contexte d'exécution aurait aussi peu d'intérêt puisque qu'il est peu probable qu'une machine se déconnectant du système se reconnecte suffisamment rapidement pour que le calcul parallèle progresse à une vitesse raisonnable. Bref, le calcul parallèle dans les systèmes Pair à Pair pose des questions intéressantes et difficiles qui restent à l'heure actuelle des sujets de recherche.

7. Conclusion

Les systèmes de calcul Pair à Pair représentent une formidable opportunité pour la globalisation et le partage des ressources de calcul et des données. Le projet XtremWeb a constitué une des premières tentatives dans ce sens en construisant une plate-forme verticale complète. Cette plate-forme est déjà utilisée en production dans différents sites. Des protocoles comme JXTA, COSM ou .NET permettent maintenant de construire ce type de systèmes mais n'offrent pas l'ensemble des mécanismes nécessaires pour construire XtremWeb, par exemple.

L'architecture du système lui-même reste un problème ouvert. Le choix entre architecture centralisée comme pour SETI@home, hiérarchisée comme pour FastTrack ou complètement distribuée comme pour Gnutella et FreeNet n'est pas établi. Une modélisation , une simulation ou une émulation des mécanismes fondamentaux (découverte de ressources, ordonnancement, routage) est nécessaire pour déterminer scientifiquement l'approche la plus pertinente. Ceci dit, les paramètres de ces modélisations et simulations sont encore à préciser. De nombreuses recherches sont aussi nécessaires sur la sécurité, la tolérance à la volatilité des neouds, les applications parallèles, l'ordonnancement et les outils d'aide à la programmation. Le projet d'ACI GRID CGP2P (Calcul Global Pair à Pair) aborde ces questions et explore des pistes de solutions. Il faudra trouver des solutions efficaces à tous ces problèmes pour permettre, un jour, de globaliser à grande échelle les ressources informatiques, à partir de systèmes Pair à Pair.

8. Bibliographie

[ADA] ADAM L. BEBERG E. A., « Project Monarch », http://www.distributed.net/des/.

[AID 98] AIDA K., NAGASHIMA U., NAKADA H., MATSUOKA S., TAKEFUSA A., « Performance Evaluation Model for Job Scheduling in a Global Computing System », *7th IEEE Int. Symp on High Performance Distributed Computing*, 1998, p. 352–353.

[ALE 97] ALEXANDROV A. D., IBEL M., SCHAUSER K. E., SCHEIMAN C. J., « SuperWeb: Towards a global web-based parallel computing infrastructure », *Proceedings of the 11 th IEEE International Parallel Processing Symposium (IPPS)*, April 1997.

[AND 97] ANDERSON D., BOWYER S., COBB J., GEDYE D., SULLIVAN W. T., WERTHIMER D., « A New Major SETI Project Based on Project Serendip Data and 100,000 Personal Computers », *Astronomical and Biochemical Origins and the Search for Life in the Universe, Proc. of the Fifth Intl. Conf. on Bioastronomy*, 1997.

[BAR 93] BARAK A., GUDAY S., WHEELER R. G., « The MOSIX distributed operating system: load balancing for UNIX », vol. 672, New York, NY, USA, 1993, Springer-Verlag Inc., page 221.

[BAR 96] BARATLOO A., KARAUL M., KEDEM Z., WYCKOFF P., « Charlotte: Metacomputing on the Web », *Proceedings of the 9th Conference on Parallel and Distributed Computing Systems*, 1996.

[CAS 00] CASANOVA H., LEGRAND A., ZAGORODNOV D., BERMAN F., « Heuristics for Scheduling Parameter Sweep Applications in Grid Environments », *9th Heterogeneous Computing Workwhop HCW'00*, May 2000.

[CLA 01] CLARKE I., SANDBERG O., WILEY B., , HONG T., « Freenet: A Distributed Anonymous Information Storage and Retrieval System », *Proceedings of the Workshop on Design Issues in Anonymity and Unobservability,Springer*, http://freenet.sourceforge.net/, 2001.

[DIS 97] DISTRIBUTED.NET, « Project RC5 », http://www.distributed.net/rc5/, 1997.

[FOS 99] FOSTER I., KESSELMAN C., « The Grid: Blueprint for a Future Computing Infrastructure », Morgan Kaufmann Publishers, 1999. 8, 1999.

[FOS 01] FOSTER I., « The Anatomy of the Grid: Enabling Scalable Virtual Organizations, IJSA, 2001 », *International Jounal on Supercomputer Applications*, 2001.

[GHO 98] GHORMLEY D. P., PETROU D., RODRIGUES S. H., VAHDAT A. M., ANDERSON T. E., « GLUnix: A Global Layer Unix for a Network of Workstations », *Software Practice and Experience*, vol. 28, n° 9, 1998, p. 929–961.

[GOL 96a] GOLDBERG I., WAGNER D., THOMAS R., BREWER E., « A Secure Environment for Untrusted Helper Applications — Confining the Wily Hacker », *Proceedings of the 1996 USENIX Security Symposium*, 1996.

[GOL 96b] GOLDBERG I., WAGNER D., THOMAS R., BREWER E. A., « A Secure Environment for Untrusted Helper Applications », *Proceedings of the 6th Usenix Security Symposium*, San Jose, Ca., 1996.

[HON 01] HONG T., « *Peer-to-Peer: harnessing the power of disruptive technologies* », Chapitre Performance, O'Reilly, March 2001.

[KAN 01] KAN G., « *Peer-to-Peer: harnessing the power of disruptive technologies* », Chapitre Gnutella, O'Reilly, March 2001.

[LAN 01] LANGLEY A., « *Peer-to-Peer: harnessing the power of disruptive technologies* », Chapitre Freenet, O'Reilly, March 2001.

[LIT 88] LITZKOW M. J., LIVNY M., MUTKA M. W., « Condor - A Hunter of Idle Workstations », *Proceedings of the 8th International Conference on Distributed Computing Systems (ICDCS)*, Washington, DC, 1988, IEEE Computer Society, p. 104–111.

[MER 97] MERSENNE G., SEARCH P., « GIMPS Discovers 36th Known Mersenne Prime », Great Internet Mersenne Prime Search. Press Release, Sept. 1997. http://www.mersenne.org/2976221.htm., 1997.

[NEA 99] NEARY M., BRYDON S., KMIEC P., ROLLINS S., CAPELLO P., « Javelin++: Scalability Issues in Global Computing », *Proceedings of the ACM Java Grande 1999 Conference*, San Francisco, California, June 1999.

[NIS 98] NISAN N., LONDON S., REGEV O., CAMIEL N., « Globally distributed computation over the internet-the popcorn project », *Proceedings for the 18th International Conference on Distributed Computing Systems*, 1998.

[PAN] PANDE V., « Genome at home », http://genomeathome.stanford.edu/, Pande Group, Chemistry Department, Stanford University.

[PED 97] PEDROSO H., SILVA L. M., SILVA J. G., « Web-based metacomputing with JET », *Proceedings of the ACM 1997 PPoPP Workshop on Java for Science and Engineering Computation*, ACM, June 1997.

[PER 99] PERCIVAL C., « PiHex, A distributed effort to calculate Pi. », http://www.cecm.sfu.ca/projects/pihex/index.html, 1999.

[RIP 01] RIPEANU M., « Peer-to-Peer Architecture Case Study: Gnutella », *International Conference on Peer-to-peer Computing (P2P2001)*, august 2001.

[ROS 99] ROSENBERG A. L., « Guidelines for Data-parallel Cycle-Stealing in Networks of Workstations », *Journal of Parallel and Distributed Computing*, vol. 59, 1999, p. 31-53.

[SAR 98] SARMENTA L. F. G., HIRANO S., WARD S. A., « Towards Bayanihan: builing an extensible framework for Volonteer Computing using java », *Proc. ACM Workshop on Java for High-Performance Network Computing*, 1998.

[SAR 01] SARMENTA. L. F. G., « Sabotage-Tolerance Mechanisms for Volunteer Computing Systems », *ACM/IEEE International Symposium on Cluster Computing and the Grid (CCGrid'01)*, 2001.

[SCI 95] SCIUTTO S. J., « Aires : Air Showers Extended Simulation », http://www.fisica.unlp.edu.ar/auger/aires/, 1995.

[SMA 92] SMARR L., CATLETT C., « Metacomputing », 1992, Communications of the ACM, 35(6):44–52, June 1992.

[SOL] SOLUTION C. D. S., « Gnutella protocol specification v0.4 », http://gnutella.wego.com/.