

— MATIN —

Découverte dynamique de sources dans un groupe multicast

Mickaël Hoerdts (LSIIT, équipe réseaux et protocoles)

Dans cet exposé, nous présentons un protocole de découverte dynamique de canaux multicast nommé SSMSDP (Source Specific Multicast Source Discovery Protocol). Ce protocole permet d'offrir les mêmes propriétés de communication de groupe que celles offertes par le modèle ASM (Any Source Multicast) de Deering, mais en ne reposant que sur le modèle SSM (Source Specific Multicast) dans le réseau. Notre proposition tire partie des avantages du modèle SSM dans le réseau tout en rétablissant les propriétés puissantes du modèle ASM de Deering au niveau des hôtes. Une session SSMSDP est maintenue par un groupe dans lequel le nombre et la dynamique des récepteurs peuvent être très grands. Nous avons implémenté ce protocole sous la forme d'une librairie librement téléchargeable, puis nous avons évalué le temps de connexion d'une nouvelle source dans une session SSMSDP existante.

Nos résultats montrent que notre proposition est réaliste puisqu'elle permet de maintenir 1000 sources actives par session SSMSDP au même moment. Nous avons étendu notre implémentation à l'inter-domaine sur le réseau M6bone et nous montrons un temps de propagation des messages PIM-Join très élevés, ce qui entraîne un temps de connexion de plus d'une seconde dans nos tests. Néanmoins, nous pensons que notre proposition reste valable, puisque cette limite semble uniquement provenir d'un problème d'implémentation. Lors de travaux futurs, nous allons étendre notre protocole pour le rendre complètement décentralisé.

Amélioration de l'équité d'accès au canal 802.11

Jean Lorchat (LSIIT, équipe réseaux et protocoles)

Au cours de cette présentation, nous montrerons les défauts existants dans la couche MAC de la norme IEEE 802.11, en particulier du point de vue de la latence. Ces défauts empêchent toute garantie de délai nécessaire à l'établissement d'un schéma de qualité de service dans les réseaux sans fil, mais aussi à la diffusion de flux multimédia de façon satisfaisante, ou encore à l'utilisation du réseaux sans fil pour transporter la voix sur IP. Nous présenterons alors comment nous avons mis au point une couche de remplacement pour la partie MAC du standard 802.11 tout en conservant la couche PHYsique.

Cette couche est basée sur un multiplexage à base temporelle (TDMA) pour partager l'accès. L'obtention par les stations de l'accès au médium est décidé par le point d'accès suivant un ordonnancement déterministe qui permet d'exempter le réseau de toute collision. Ceci permet d'obtenir une borne supérieure sur la valeur de latence entre chaque trame d'un même flux. Enfin nous introduirons des résultats de simulation obtenus pour ce protocole, que nous comparerons alors avec le protocole MAC existant.

Gestion de la topologie dans une grille de calcul par l'intermédiaire d'une marche aléatoire

Cyril RABAT (CRéSTIC, équipe LICA)

Le GRID computing est une nouvelle approche pour le partage de ressources dans un réseau : la puissance de calcul, l'espace de stockage, des applications spécifiques, etc... Une grille de calcul comme SETI@home produit une puissance de près de 140 TFlops alors que l'actuel plus gros supercalculateur mondial BlueGene ne produit « que » 70 TFlops. Cependant, l'utilisation de nombreux ordinateurs (près de 4 millions pour SETI) induit de nombreux problèmes de communication. Il est essentiel de maintenir la grille dans le but de réussir le moindre calcul.

Pour parvenir à modéliser un réseau et dans le but d'être capable de développer des applications de GRID Computing, on propose un nouveau modèle décomposé en plusieurs couches de

fonctionnalité : routage, communication, gestion de la topologie... De plus, on présente une solution de gestion de la topologie complètement distribuée basée sur les marches aléatoires. Aucune structure virtuelle n'a besoin d'être maintenue et cette solution fonctionne sur les réseaux asynchrones. On présente aussi quelques résultats de simulations de cette solution.

— APRES-MIDI —

Etude de la notion de service sur la plate-forme de Grid Computing DIET Suphakit Niwattanakul (LIFL, équipe SDR)

Un des objectifs principaux de mes travaux est l'étude des concepts principaux et des technologies de services en vue d'intégrer de nouveaux environnements dans la plate-forme de Méta Computing DIET. DIET est un intergiciel de calcul de grille et une plate-forme de Méta Computing. Cette plate-forme est indépendante des langages de programmation. Actuellement les fournisseurs de services DIET sont dans un environnement CORBA. Une première partie de mon travail s'intéresse à la mise en oeuvre de l'accès à des services dans d'autres environnements, comme par exemple les environnements des Web Services.

De plus, la description des services doit être développée pour permettre une recherche plus "fine" des services et une comparaison des types de services. Pour cela nous envisageons d'introduire un Type Repository pour la description des types de services.

Mots-clés: Services, Description de services, Web Services, DIET, Méta Computing

Un intergiciel pair-à-pair pour l'exécution robuste de programmes parallèles Stéphane Genaud et Choopan Rattanapoka (LSIIT, équipe ICPS)

P2P-MPI est un intergiciel destiné aux grilles de calcul, qui propose un modèle de programmation de type passage de messages. Il apporte plusieurs nouveautés qui favorisent une utilisation et une administration simplifiée. L'une de ses caractéristiques est sa conception pair-à-pair, permettant à un participant de découvrir d'autres participants sans annuaire centralisé. P2P-MPI intègre aussi un mécanisme de réplication des processus afin d'accroître la robustesse d'une application. Cette réplication est totalement transparente pour le programmeur (seul le nombre de réplicas est précisé par l'utilisateur pour l'exécution de son application). Des tests de performances effectués sur deux des NAS benchmarks (IS et EP) sur une configuration à 24 PC reliés par un réseau fast-ethernet montrent que P2P-MPI (sans réplication) est aussi performant que des bibliothèques très utilisées (LAM et MPICH).

A Cellular Computing Library on Clusters and Grids: ParCeL-6 Mircea Ifrim et Didier Vagner (Supélec Metz)

The goal of ParCeL-6 is to decrease the development time for fine-grained applications. The implementation is done on coarse grained parallel and distributed architectures in order to take benefit of modern, generic and cheap machines. ParCeL-6 has a fine grained parallel programming model that is a successful compromise between the architecture requirements in order to reach high performances and the developer requirements to make quick developments. Currently, there are two submodels of ParCeL-6: ParCeL-6.1 for architectures supporting memory sharing paradigm and ParCeL-6.2 for architectures supporting message passing paradigm. We ran ParCeL-6.1 on a Linux cluster after we have installed a Distributed Shared Memory system on it. Kerrighed DSM helped us to transform this cluster in an "almost" shared memory system. The results showed that the Kerrighed DSM prefers the "embarrassingly" parallel computations applications in which cases it exhibits a very good speedup, but in the case of irregular memory accesses in the applications, a performance slowdown can be observed. ParCeL-6.1 worked on the DSM without any problems, but with bad performances.

On the other hand, we developed ParCeL-6.2 on MPI in order to have a library much more scalable. It cannot implement direct communication mode where cells can directly access to one another's output, but instead, it proposes a hybrid communication mode that try to afford the benefits of both direct and buffered mode. Performance measurement showed that the MPI's initialisation and barriers are very negligible towards the execution and routing phases. Finally we propose two different ways of investigating : the use of an intermediate library, SSCRAP, that group and manage efficiently MPI's communication, that help providing a high scalable version of ParCeL-6.2 that is planned to be tested on Grid eXplorer; and the use of ParCeL-6.2 on SSCRAP on MPI using a DSM to take advantage of the cluster-wide shared memory that can be very useful to interactivity and sequential parts of the application.