

— MATIN —

Equilibre de charge dynamique pour l'optimisation combinatoire
Christophe Jaillet (LICA)

Ces travaux s'inscrivent dans le cadre de la recherche combinatoire et de l'optimisation combinatoire. Les problèmes sont modélisés sous la forme de CSP, et nous proposons une méthode de résolution énumérative générique. L'arbre de recherche est élagué à la volée, par l'utilisation conjointe des contraintes et d'une fonction d'évaluation incrémentale. Pour la parallélisation, des tâches sont générées en développant l'arbre de recherche à une certaine profondeur, et seules celles qui sont utiles sont conservées. Dans le cadre des problèmes d'optimisation, nous avons proposé une stratégie originale d'équilibre de la charge, qui permet d'éviter les tâches obsolètes, par des sauts dynamiques en cours de résolution.

Nous avons également mis en évidence un phénomène de contention mémoire dans le cas de la programmation en mémoire partagée avec OpenMP, et mis au point une méthode d'allocation mémoire, basée sur une sémantique, qui corrige ce défaut sans surcoût.

Le problème pratique présenté ici est celui de la construction de règles de Golomb optimales. Il est considéré comme un challenge par la communauté académique. Les différents apports que nous avons amenés nous ont permis de montrer qu'il est possible de résoudre ces problèmes efficacement, par l'utilisation de machines parallèles ou de grilles de calcul.

Architecture logicielle sur grille tolérante aux pannes. Application aux calculs financiers

Sébastien Bezzine (CNAM de Lorraine & Supélec)

Les technologies de clusters et de grilles de calculs sont de plus en plus communément adoptées dans l'industrie, et notamment par l'industrie financière (banques et assurances). Cependant ces technologies souffrent toutes d'un problème inhérent aux systèmes distribués de grande taille : la fréquence des défaillances et des disparitions momentanées des ressources. La plupart des solutions actuelles gèrent la tolérance aux pannes sans intégrer les contraintes de temps : en cas de défaillance de certaines ressources les calculs sont relancés automatiquement mais les résultats sont finalement disponibles plus tard que prévu.

Pour pallier à ces faiblesses, nous développons une architecture logicielle sur grille qui soit tolérante aux pannes tout en respectant au mieux les contraintes de temps. Elle s'appuie sur des mécanismes de tolérance aux pannes à la fois au niveau applicatif et au niveau du middleware, et destinés à collaborer pour réduire les temps de récupération sur erreur. Pour réaliser ce projet, nous avons décidé d'utiliser ProActive, un middleware de grille et un environnement de développement se présentant sous la forme d'une bibliothèque Java développée par l'équipe OASIS de l'Université de Nice et de l'INRIA Sophia-Antipolis.

A ce jour nous avons développé une première architecture « d'objets actifs » se déployant sur une grille, détectant ses propres défaillances et se régénérant en incorporant si possible de nouvelles ressources. Une seule ressource est supposée fiable, ne se régénère pas en cas de défaillance, et garantit ainsi que l'on peut toujours stopper le système en cas d'urgence (en la détruisant !).

Ce projet se déroule principalement dans le cadre de l'ANR-CIGC GCPMF, en collaboration avec les équipes OASIS et OMEGA de Sophia-Antipolis.

Contrôle de ressources par multiroutage
Pascal Mérindol (LSIT)

Les protocoles de routage actuellement déployés (OSPF, RIP) ne profitent pas pleinement et équitablement des ressources que lui offre le réseau physique. En effet, d'une part ces protocoles et leurs algorithmes de recherche opératoire sous jacents (Dijkstra, Bellman-Ford) ne maximisent

pas la bande passante disponible entre deux points, et se contentent de calculer le meilleur chemin selon une certaine métrique entre une source et une destination. D'autre part, le principe de sous-optimalité du meilleur chemin dans un graphe, qui permet à OSPF de router au saut par saut alors que le calcul des routes se fait à la source avec l'algorithme de Dijkstra, entraîne un partage non équitable des ressources, aussi bien en termes de routeurs sollicités qu'en bande passante consommée. Le multiroutage permet de tirer profit du maillage du réseau physique et de partager plus équitablement les ressources qu'il propose. Dans cet article, nous proposons une solution de routage multichemins distribuée et extensible dont nous comparerons les performances en terme topologique aux différentes méthodes d'ingénierie de trafic basées sur du multiroutage, qu'elles soient distribuées ou réalisées à la source.

Evaluation de performance des algorithmes distribués à base de marches aléatoires

Devan Sohier (LICA)

Un système distribué est un ensemble d'ordinateurs interconnectés. Avec l'augmentation des puissances de calcul et l'importance croissante de l'internet, les systèmes distribués ont une taille toujours croissante : il est plus difficile de gérer leur topologie, qui est très fluctuante. Les algorithmes distribués à base de marches aléatoires est un algorithme utilisant un message spécial, le jeton, qui sert soit à récolter des informations (topologiques, par exemple), soit à conférer un privilège au site qui le détient, et qui circule suivant un schéma de marche aléatoire : à chaque étape, un site possède le jeton, qu'à l'étape suivante il transmet à un voisin choisi uniformément au hasard. Ce schéma n'utilise que des informations locales, ce qui permet à ces algorithmes de s'adapter aux changements topologiques du système.

Les principales quantités qui apparaissent dans l'évaluation de la complexité des algorithmes distribués à base de marches aléatoires sont les temps de percussion (temps moyen pour que le jeton atteigne un site donné), le temps de couverture (temps moyen pour que le jeton ait visité chacun des sites) et le temps de rencontre (temps moyen pour que deux jetons se rencontrent). Nous généralisons ces notions, initialement définies sur des graphes non-valués, au cas des graphes valués, de deux manières différentes (et fournissons les interprétations de ces deux notions). Nous généralisons également les bornes et théorèmes portant sur ces quantités, pour les deux versions présentées.

A partir d'un lien entre temps de percussion et résistances électriques, nous écrivons un algorithme calculant exactement les n^2 temps de percussion en une inversion de matrice (de taille n). Nous écrivons également les premiers algorithmes de calcul exact des temps de percussion et de couverture. Enfin, en analysant les variances des temps de percussion, nous modifions les algorithmes précédents pour calculer les variances des temps de percussion, de couverture et de rencontre pour un surcoût négligeable ($o(n^3)$).

Ces calculs peuvent en particulier être utilisés pour déterminer des topologies adaptées à la circulation de marches aléatoires, et pour hiérarchiser un graphe de manière à gérer efficacement ces changements topologiques à l'aide d'algorithmes distribués à base de marches aléatoires. Ce sont les espérances et les variances des grandeurs caractéristiques qui permettent de déterminer la taille à donner à chacun de ces sous-graphes.

— APRES-MIDI —

Métrieologie des Réseaux IP : Détection de Bottlenecks et Mesure de la Bande Passante Disponible

Ahmed Ait Ali (CRAN)

La bande passante disponible est un critère primordial pour le bon fonctionnement de nombreuses applications et technologies réseau. Il existe plusieurs outils qui permettent la mesure de cette métrique ; cependant on déplore l'absence d'études comparatives de ceux-ci. Dans la première partie de ce travail, nous proposons une étude expérimentale comparative des outils Pathload, Spruce, Pathchirp et IGI, puis nous expliquons et analysons les erreurs et les incertitudes dont ces outils font l'objet. Les résultats des expérimentations mettent en évidence les qualités de l'outil

Spruce. En effet, celui-ci semble offrir les meilleures performances en termes de précision, de rapidité de mesure et d'intrusivité. Dans la deuxième partie, nous proposons une nouvelle technique de détection de bottlenecks basée sur la méthode de paires de paquets récursives. Cette technique est implémentée dans l'outil TLFinder (Tight Link Finder) qui sonde le bottleneck afin de mesurer la bande passante disponible.

Contrôle du plan de gestion: approche basée sur la théorie du contrôle Abdelkader Lahmadi (LORIA)

Le plan de la gestion de réseaux et de services est de plus en plus complexe à cause de la complexité et la dynamique de systèmes gérés (réseaux mobiles, réseaux ad-hoc, réseaux personnels, etc). Dans ces environnements, le plan de gestion est souvent intégré dans le plan fonctionnel. Cette intégration a mis l'accent sur la performance du plan de gestion et sur son efficacité pour gérer les systèmes. Plusieurs travaux ont proposé des approches empiriques pour optimiser et contrôler le plan de gestion en agissant sur certains de ses paramètres (nombre d'agents, nombre d'objets gérés, fréquences de scrutation, filtrage de notifications, agrégation, etc). Afin d'avoir une approche unifiée et un cadre solide de contrôle et d'optimisation, nous proposons d'appliquer la théorie de contrôle sur le plan de gestion. Dans cet exposé nous présentons : les motivations supportant notre proposition, les enjeux et les verrous pour appliquer la théorie de contrôle sur le plan de gestion et une illustration de notre approche sur un algorithme de supervision de réseaux.