Bilan du projet Apache 1996-2004

Les indiens

1 Thématiques scientifiques

Apache est un acronyme pour « Algorithmique Parallèle et pArtage de CHargE », ce qui est la devise du projet en 1996. Notre objectif était de contribuer au développement d'une programmation parallèle facile, portable et efficace sur les architectures parallèles existantes ou émergentes. Le travail de recherche a porté sur les domaines suivants qui, au départ du projet, comptaient les plus importantes lacunes :

Algorithmique et applications parallèles : le projet s'est attaqué à l'élaboration d'algorithmes et applications parallèles réalistes dans les domaines du calcul scientifique (océanographie, dynamique moléculaire) ou du calcul formel. Ce travail était nécessaire à l'époque du fait du faible corpus d'algorithmes parallèles et d'implantations fortement liées aux architectures cibles (processeurs vectoriels, machines SIMD,..).

Ordonnancement statique et en ligne : tout calcul parallèle se ramène à assigner des tâches coopérantes à des processeurs de façon à réduire le temps de calcul global. Élaborer un bon ordonnancement nécessite de tenir compte des enchaînement des pas de calcul et des pas de coopération (synchronisation, communication) qui dépendent totalement de l'architecture cible. L'ambition du projet était double. D'une part, il s'agissait d'élaborer des stratégies d'ordonnancement efficace adaptable à toute architecture et d'autre part, déterminer les propriétés exigées d'une application pour être ordonnancée au mieux.

Modèle et environnement de programmation parallèle : il n'existait aucun environnement de programmation (langage, compilateur, lanceur,..) permettant l'expression « directe » d'un algorithme parallèle et l'automatisation de la production et l'exécution du code machine correspondant. Un des objectifs du projet était la conception et la validation d'un tel environnement de programmation parallèle.

Modèle de machine et environnement exécutif : il n'existait aucun environnement exécutif standard compatible avec les diverses architectures disponibles. Un des objectif du projet a été de se donner un modèle d'architecture le plus « général » possible, de développer pour ces architectures un noyau exécutif portable et l'environnement de programmation de « bas niveau » correspondant.

Dès le départ, le projet s'est fixé une démarche expérimentale afin de tester au plus vite les concepts élaborés. Cette démarche devait s'appuyer sur le déploiement d'une série de plateformes expérimentales significatives de l'état de l'art en architecture de machine et largement ouvertes à la communauté scientifique intéressée par le calcul parallèle. Elle impliquait un travail important d'évaluation de performances de système et applications parallèles. L'effort humain nécessaire à cette démarche comme les contacts scientifiques et industriels extérieurs ont donc fait partie de la genèse du projet. Le domaine de recherche était largement ouvert au départ. Plusieurs pistes plus ou moins exclusives les unes des autres étaient possibles. Le projet a effectué initialement un certain nombre d'hypothèses de travail scientifiques et techniques qu'il a fallu confronter avec les choix initiaux et solutions apportées par d'autres projets.

1.1 Modèle de machine parallèle cible

Deux modèles de machine étaient en confrontation et le sont toujours. Le premier est une architecture multiprocesseur à mémoire commune[1]. Le second modèle est celui d'une « grappe » de processeurs sans mémoire commune interconnectés par un réseau. La première de ces architectures était la plus ancienne. Elle se caractérise par un modèle exécutif « bien connu » par ses concepts de programmation concurrente (processus léger, exclusion mutuelle, moniteur), ses opérateurs d'interaction (verrou, sémaphore, condition) et une utilisation dans la réalisation de systèmes d'exploitation et de serveurs INTERNET. Sur une telle architecture, un programme parallèle s'exprime en termes de processus se partageant des calculs sur des données communes et se coordonnant selon les propriétés de ses données globales.

La seconde architecture plus récente est liée à l'apparition des réseaux d'interconnexion standard efficace (LAN 802) ou propriétaire à haut débit et faible latence permettant d'interconnecter des processeurs standard. Cette méthode permet une puissance de calcul cumulée forte pour un coût raisonnable. Dans ce cadre, un programme parallèle nécessite d'expliciter le placement de calculs et données sur les processeurs. De même la progression des calculs nécessite d'expliciter les échanges de données entre les différents processeurs.

Chacune de ces architectures offre des avantages/inconvénients propres et fait face à des problèmes spécifiques. La première architecture offre un parallélisme de grain fin (pas de boucle d'itération, routine de calcul) exploitable directement par les noyaux exécutifs actuels. Par contre, elle posait le problème du passage à l'échelle de dizaines de processeurs[2] c'est à dire essentiellement la conception d'un mécanisme de mémoire partagée efficace[3]. Le second type d'architecture passait facilement à l'échelle de centaines de processeurs. Cependant il n'existait aucun accord sur modèle standard de noyau exécutif et en particulier sur le jeu d'opérateurs de communication le plus approprié à la programmation parallèle.

Dès le départ, le projet APACHE a considéré que la machine « cible » du projet devait être la grappe de multiprocesseurs SMP (CLUMP[4]) car l'évolution de l'architecture des machines évoluait selon 2 axes indépendants. La progression des techniques d'intégration conduisait à des SMP de faible arité[5] pour un coût inférieur au même nombre de proceseurs indépendant. A partir de ce seuil, il était plus facile et moins cher de les assembler via un réseau d'interconnexion standard LAN802 ou dédié.

1.2 Environnement exécutif

Le premier axe du projet a donc été une contribution à la définition et l'implantation d'un environnement exécutif pour CLUMP. Les lacunes dans ce domaine portaient sur l'absence de noyau exécutif et d'outil de lancement de programme. Le travail s'est initialement porté sur les problèmes de conception d'un noyau exécutif efficace puis sur les outils de lancement.

Noyau exécutif : L'approche choisie pour la définition d'un noyau exécutif a simplement été de « virtualiser » l'architecture cible. Celle-ci étant un réseau physique de noeuds SMP, le noyau exécutif simule un réseau « complètemenent maillé » de noeuds SMP virtuels. Le principe d'implantation est trivial. Un multiprocesseur virtuel est représenté par un « processus lourd » du système hôte d'un noeud physique. Le parallélisme de ce noeud est implanté par des « processus légers ». Les processeurs d'un même noeud virtuel peuvent communiquer via les concepts et techniques « classiques ». La coopération entre noeuds différents se fait par l'utilisation d'un protocole de communication approprié à définir. Cette approche a été partagée par différents projets (PM2 (LIFL,LIP), Nexus (ANL),...). Elle s'oppose aux approches « homogènes » qui se limitent à gérer un seul type de parallélisme. Une de ces approches considère un modèle de type réseau de monoprocesseurs et se focalise sur la conception d'un noyau de communication entre noeuds (PVM, MPI). Une autre approche « simule » une mémoire virtuelle globale distribuée sur les noeuds de façon à réutiliser les concepts et techniques de programmation parallèle

[1] Avec deux variantes dite Symetric MultiProcessors (SMP) ou Cache Coherent - Uniform Memory Access (CC-UMA) ou CC-NUMA selon que les temps d'accès à la mémoire sont homogènes ou non.

[2] Aujourd'hui de centaines de proceseurs.

[3] Réseau d'interconnexion proceseurs-mémoire et maintien de la cohérence des caches.

[4] Cluster of MultiProcessors.

[5] 2 au début du projet puis 4 aujourd'hui.

des multiprocesseurs à mémoire commune. C'est le cas des projets Treadmarks(RICE), MOM(IRISA) ou DOSMOS(LIP). Dans la mesure où il était déjà connu que les performances d'un calcul parallèle sont très dépendantes de l'adéquation des grains de calcul aux latences de calcul et de communication, nous pensions que l'approche APACHE était plus appropriée à l'exploitation des différents niveaux de parallélisme d'une machine. En particulier le parallélisme entre les calculs et les communications devrait être exploitable par des techniques de multiprogrammation. Un important travail initial a donc été consacré à la vérification de cette hypothèse. La validité de cette approche a été confirmée expérimentalement par la réalisation de noyaux exécutifs (Athapascan0) à destination de différentes grappes (IBM SPx, CrayT3E, cluster d'ALPHA, PC ou Sun...) et même CC-NUMA (SGI Origin2000, Sun Enterprise) et leur utilisation pour des applications de calcul scientifique (Dynamique moléculaire en particulier). Les limites de cette approche ont été aussi clairement identifiées. La première limite est la difficulté de programmer un algorithme parallèle exploitant au mieux ces différents niveaux de parallélisme : parallélisme calcul-communication, parallélisme SMP et parallélisme entre noeuds. La seconde limite n'est pas propre à notre approche. Celle-ci a aussi mis en évidence la faible réactivité des systèmes hôtes aux événement de communication réseau. Ceci s'est traduit par l'impossibilité d'exploiter la faible latence des réseaux modernes (Myrinet, SCI, Quadrics). Les modifications nécessaires des systèmes hôtes et noyaux exécutifs sont attaquées aujourd'hui par les constructeurs de grappes (IBM, Compaq, BULL) et certains projets (projet RunTime INRIA FUTUR/LABRI).

Lancement et mouvement de données : Dès la mi-projet, les expériences ont mis en évidence les insuffisances des outils de lancement de programme parallèle et de mouvement de données au sein d'une grappe de grande taille. La cause en était une implantation parallèle « naïve » sur des protocoles standard de l'INTERNET (rsh,ssh,ftp..). APACHE a exploité la compétence acquise en algorithmique parallèle pour proposer une implantation exploitant au mieux les sources de parallélisme du système. APACHE a donc conçu le lanceur parallèle Taktuk. Il repose sur des protocoles standard (rsh,ssh) à la différence d'autres projets qui se sont lancés dans la définition de protocoles adhoc. Cette implantation portable s'est révélée aussi efficace sinon plus que les autres approches à l'exception de celles qui incorporent ces outils au niveau matériel (COMPAQ-QUADRICS). Dans le cadre d'un comportement fixe de l'architecture hôte (mono-application sur grappe homogène), la solution proposée s'adapte automatiquement à l'architecture et est optimale. La préservation de cette propriété en cas de comportement variable (cas de plusieurs applications) reste un problème de recherche. Dans la conception d'un système de fichiers distribués sur grappe, la même démarche a été utilisée pour proposer une version « grappe » du protocole standard NFS. « NFSp » préserve l'interface applicative pour pouvoir être utilisé de façon transparente par tous les systèmes clients NFS. Par contre, le serveur NFS a été parallélisé et distribué de façon à pouvoir exploiter la bande passante cumulée entre clients et serveurs. Cette exploitation optimale dépend d'une répartition appropriée des données sur les serveurs pour un ensemble de clients donnés. Ceci est aussi un problème de placement à résoudre.

1.3 Modèle et environnement de programmation parallèle

Dès le début, le projet avait conscience de la difficulté de définir précisément puis de programmer un algorithme au niveau d'un noyau exécutif. L'expression de la découpe des données et des calculs élémentaires dans les différentes représentations nécessaires à l'exploitation des différents types de parallélismes disponibles (recouvrement des communications, threads,...) ainsi que l'expression des règles de placement et d'ordonnancement de ces calculs dépassent de loin l'expression même des calculs « utiles ».

Modèle de programmation parallèle : L'automatisation de ces aspects pouvait se baser sur deux modèles de calcul parallèles. Le premier modèle était le modèle BSP (Bulk Synchronous Parallel processing) qui consistait à concevoir un algorithme parallèle comme un enchaînement alterné de phases de calculs parallèles indépendants et de phases d'échanges de données. Ce modèle était particulièrement apte à rendre compte de calcul itératif sur des gros ensembles de données de structure régulière. Le second modèle « Macro Data Flow » est issu des travaux de conception des machines à flot de données.

Un algorithme parallèle y est vu comme un graphe où les noeuds sont des opérateurs et les arêtes les données circulant entre ces calculs. Une fois un modèle choisi, il fallait se donner un langage de description de l'algorithme parallèle. Un grand nombre d'approches étaient envisageables. L'approche minimale aujourd'hui dominante consistait à offrir une bibliothèque « carrossant » un noyau exécutif pour le modèle BSP et les règles de programmation permettant de passer d'un algorithme BSP à un programme BSP. Ces bibliothèques BSP ont été portées sur des architectures à mémoire commune ou distribuée. C'est là une des justifications principales des opérateurs de communication collective de MPI. L'approche maximale consistait à éliminer le problème de la conception de l'algorithme parallèle. Un algorithme était décrit de façon traditionnelle séquentielle et un compilateur parallèliseur doit « trouver » le parallélisme possible. Destinée à récupérer les codes FORTRAN existants, les projets HPF (High Performance Fortran) se sont donné un modèle BSP et l'objectif d'extraire cette type de parallélisme automatiquement d'un programme FORTRAN. Cette approche séduisante a buté sur la difficulté de la tâche d'extraction du parallélisme c'est-à-dire du calcul de l'indépendance des différentes étapes séquentielles. Cet échec s'est traduit par la définition d'une extension des langages FORTRAN et C où l'expression du parallélisme BSP était explicite : OpenMP. OpenMP a été immédiatement implanté sur des machines à mémoire commune. Une implantation efficace sur grappe reste à faire. Le projet APACHE a suivi l'approche « Macro Data Flow » qui était plus appropriée à rendre compte de structures de calculs irrégulières telles que celles rencontrées dans les domaines du calcul formel qui était un des domaines d'application du projet. Plutôt que de définir un nouveau langage de programmation comme SISAL ou CILK, nous avons choisi d'utiliser la généricité de C++ pour proposer une « extension générique » permettant une expression aisée d'un parallélisme « Macro Data Flow » : Athapascan. Le projet s'est alors focalisé sur une implantation à destination de CLUMP (CLuster of MUltiProcessors). Athapascan est aujourd'hui un des rares exemples d'environnement où un programme peut tourner sur une architecture à mémoire commune ou une grappe sans modification du texte source.

Implantation d'un moteur exécutif de haut niveau : Des problèmes demeurent dans le domaine des applications irrégulières pour exploiter efficacement les parallélismes d'une grappe. Ce sont des problèmes techniques d'implantation efficace des synchronisations « Macro Data Flow » et des problèmes d'ordonnancement dans le cadre d'applications générant dynamiquement des tâches parallèles irrégulières de façon inconnue a priori. Athapascan comme CILK ont montré la validité de l'approche « Macro Data Flow » tout mettant en évidence le problème critique de l'ordonnancement « en ligne » des applications irrégulières (calcul formel, bioinformatique). La solution retenue en fin du projet se base sur une hiérarchisation du calcul qui soit proche de l'architecture CLUMP ce qui permet d'optimiser l'allocation des données au plus près des calculs. Si cette solution a été montrée efficace pour les algorithmes possédant une bonne localité structurelle (cache oblivious problem), des problèmes demeurent afin de gérer mieux et plus généralement l'affinité des calculs et des données. Cela concerne aussi le calcul d'un bon ordonnancement. La conception d'un tel moteur exécutif de haut niveau est un domaine de recherche actif.

1.4 Ordonnancement statique et en ligne

L'efficacité d'un programme parallèle suppose de placer correctement les tâches élémentaires de calcul ou de communication sur les processeurs et d'assurer l'enchaînement optimal de ces tâches sur les différents processeurs. Quand les tâches et leurs dépendances sont connues a priori, il possible de construire un ordonnancement a priori. Dans le cas contraire, on doit se rabattre sur une heuristique d'ordonnancement à la volée. Le projet a donc maintenu une activité permanente sur l'étude d'ordonnancements dans le cadre général du calcul parallèle (ordonnancement/placement de tâches, de communication etc..) comme dans les cadres particuliers de la conception de l'environnement de programmation parallèle Athapascan ou le développement d'applications parallèles (Océanographie, dynamique moléculaire,..). Les résultats du projet en matière d'ordonnancement sont visibles au niveau national et international. Les résultats les plus récents portent sur l'ordonnancement de « tâches malléables » dans la mesure où la propriété de malléabilité implique

que l'ordonnanceur peut décomposer une tâche en sous-tâches à un niveau arbitraire. Cette propriété de décomposabilité étant une des propriétés du graphe des tâches « Macro Data Flow » d'Athapascan, il devrait être possible d'utiliser cette technique d'ordonnancement dans ce cadre.

1.5 Algorithmique et applications parallèles

Des études algorithmiques ont été conduites dans différents domaines et ont servi dans le développement d'applications parallèles. Dans le domaine des communications collectives (diffusion,accumulation ou échange), les résultats ont été utilisés ultérieurement dans les développements d'outils comme le lanceur de programme parallèle applicatifs ou les applications de simulation de tombés de tissus et de recherche en optimisation combinatoire. Dans le domaine du calcul formel, des méthodes de résolution parallèle d'équations ont été développées et validées. Dès le début du projet, une activité de parallélisation d'applications scientifiques (climat, dynamique moléculaire, calcul formel, simulation pour la visualisation, ...) a permis de promouvoir et de valider en vrai grandeur l'utilisation de grappes pour le calcul intensif ainsi que les environnements et les outils de programmation parallèle issus du projet. Elle a permis la promotion du calcul parallèle sur grappe auprès de la communauté scientifique. Le consortium CIMENT est le lieu privilégié d'échange et de coopération de tous les acteurs de la communauté scientifique grenobloise impliqués dans le calcul parallèle intensif.

1.6 Expérimentation et évaluation de performances

Le besoin d'évaluation des solutions avancées par le projet APACHE a nécessité le déploiement d'une suite de plateformes matérielles/logicielles, le développement de maquettes démonstratives ou prototypes, la mesure des performances obtenues et leur interprétation. Au cours de sa vie, le projet a donc déployé une série de plateformes de type grappe caractéristique de l'évolution de la technologie. A coté de diverses petites plateformes de PC destinés à l'évaluation de réseaux haut débit- faible latence (Myrinet ou SCI), on trouve des plateformes utilisables pour du calcul intensif et ouverte à la communauté :

- IBM/SP1 de 32 noeuds,
- cluster de 200 iVectra HP,
- cluster de 100 Itanium2 HP.

Les première et troisième opérations ont été financées dans le cadre des plans État-Région. La seconde est une donation de la société HP. Cette plateforme de 200 processeurs était la seule en France permettant des expérimentations informatiques de grande taille et a donné au site Grenoblois une grande visibilité. Ces opérations ont été effectuées dans le cadre de coopération étroite avec des industriels :

- intégration de threads dans MPI (IBM Yorktown Height),
- outils d'administration et déploiement de grappe (projet RNTL CLIC avec Mandrake et Bull),
- exploitation de jachères de ressources dans un Intranet (HP Labs Grenoble),
- études et évaluation de la mise en oeuvre de grappe de CC-NUMA Itanium Linux (action Bull/INRIA LIPS).

A cette activité de déploiement et d'exploitation s'est associée une activité d'instrumentation, de mesures et d'analyse de ces mesures en vue d'établir la validité des solutions proposées c'est-à-dire :

- l'efficacité atteinte relativement à l'efficacité théoriquement atteignable ou les solutions concurrentes,
- la propriété de « passage à l'échelle » de la solution proposée.

La très grande difficulté d'interprétation des mesures pour trouver les points noirs comme les cycles d'inactivité des ressources nous a amené à concevoir un outil de représentation visuelle paramétrable des traces temporelles des processus et de leurs interactions : Pajé. Cet outil s'est révélé très utile et adaptable hors du contexte du calcul parallèle (Analyse du comportement de SoC Thomson SMT, analyse de serveur d'EJB-BULL, Jonathan CORBA - FranceTelecom).

2 Bilan et ouverture scientifiques

Le projet a contribué de façon significative à valider et promouvoir les grappes pour le calcul intensif et les techniques d'exploitation de ce parallélisme : processus léger, recouvrement calcul-communication par multiprogrammation légère. Il a montré que le modèle « Macro Data Flow » était un modèle facilitant l'expression d'un parallélisme indépendant de l'architecture de machine et, par là même, fédérateur car implantable sur multiprocesseur et grappes. Les résultats du projet en matière d'ordonnancement sont visibles au niveau national et international. Le projet a échoué à promouvoir Athapascan comme alternative à MPI du fait des difficultés d'implantation d'un prototype robuste et de l'absence d'un ordonnanceur efficace en toutes circonstances. Ces problèmes de fond sont communs aux approches d'autres projets cherchant à promouvoir une programmation parallèle facile et efficace comme le projet OpenMP. Ce thème scientifique est toujours d'actualité.

Les outils de lancement parallèle de programmes (Taktuk) et système de fichiers distribué (NFSp) ont été incorporé dans une distribution Linux pour cluster (Mandrake Clustering). Taktuk a servi de base à l'outil d'installation automatique et de mise à jour des systèmes sur une grappe.

Au cours du projet, d'autres problèmes intéressants sont apparus et ont été explorés :

- L'analyse du comportement des programmes parallèles en vue de détecter les cycles d'attente inutile s'est révélé difficile. Le projet s'est impliqué dans la conception et la réalisation d'un outil de visualisation graphique (Pajé) du comportement d'un programme parallèle. Cet outil est basé sur le modèle des traces temporelles utilisé en algorithmique distribué. Il rend compte les relations de causalité. Ce modèle est enrichi par une « datation globale » des évènements de la trace. Ceci a nécessité la réalisation d'une « horloge globale » pour grappe.

- L'utilisation du parallélisme dans le domaine de la « Réalité virtuelle » a posé le problème du couplage d'applications de simulation physique (dynamique moléculaire, tombé de tissu) avec de la visualisation graphique et où le grain de la simulation est guidée par le point de vue du spectateur. Le parallélisme est introduit dans cette boucle d'interaction au niveau de la simulation physique, dela capture (multiples sources) comme de l'affichage (multiples opérateurs graphiques / PC).

- Le déploiement des plateformes du projet a mis en évidence les difficultés d'exploitation de grappes, de grille ou des jachères de ressources d'Intranet de grande taille de plusieurs milliers de noeuds (Intranet HP : jachère constante de 20000 noeuds). Ces problèmes concernent tous les aspects de la vie d'un tel système distribué : installation locale des systèmes et des applications, configuration correcte et interopérable des noeuds, sécurité, ordonnancement des travaux soumis, disponibilité variable des ressources. Un objectif ambitieux est de concevoir un système s'« auto-organisant » en fonction d'objectifs fonctionnels et non fonctionnels (sécurité, politique d'allocation des ressources par exemple) et pouvant passer à l'échelle d'un très grand nombre de noeuds. Concrètement, il s'agit de pouvoir construire un «squelette d'infrastructure » intelligente qui puisse évoluer dynamiquement par ajout/retrait spontané de noeuds. Ce problème est commun aux divers projets de calcul global (Global Computing) ou système distribué « pair à pair ». Le projet Grid5000 de construction d'une plateforme expérimentale distribuée au niveau national de 5000 noeuds est une cible excellente pour ce travail.

3 Heurs et malheurs

Le bilan d'un projet a tendance à faire apparaître son évolution comme lisse et continue. La réalité est tout à fait différente. En plus des réussites ou échecs de recherche, sa vie est rythmée par des évènements souvent contingents. Certains de ceux-ci ont eu des impacts décisifs sur la suite du projet. Citons en quelques exemples.

MPI multithread : Le projet s'était investi en coopération « informelle » avec IBM sur la définition d'un MPI multithread. Dans ce cadre un thésard avait été envoyé aux USA pour y contribuer. L'abandon du projet par IBM et le caractère non public de la maquette produite a contraint APACHE à se rabattre sur un palliatif et à la perte de l'investissement effectué.

Plateforme de 200 PC : L'installation des HP Labs à Grenoble et sa recherche de contact avec la communauté scientifique se sont traduits par une donation d'une grappe de 200 PC (Icluster1) et une offre attractive pour une grappe de 100 biprocesseurs Itanium (Icluster2). En France, la plateforme Icluster1 était la seule plateforme ouverte à la communauté et permettant des expérimentations informatiques de grande taille. Elle a donné au site Grenoblois une grande visibilité. L'INRIA nous a aidé par deux ingénieurs associés qui ont développé le c?ur du middleware d'exploitation.

Action Bull/Inria LIPS : L'abandon par IBM de AIX comme système de base de l'Itanium a contraint BULL à axer son activité cluster Itanium pour le HPC autour de Linux. Cela s'est traduit par un renforcement de la coopération du projet avec BULL en particulier au niveau des financements de thèse (3 thèses terminées et 4 en cours).

Jachère de calcul : La disparition des HPLabs Grenoble s'est traduite par l'arrêt d'un projet de recherche d'exploitation des jachères à très grande échelle (Intranet HP). Celui-ci redémarre avec des ambitions moindres dans le cadre du projet RNTL IGGI.

Dans tous ces cas, heureux ou malheureux, les causes étaient sans rapport avec l'activité scientifique du projet mais correspondait à des réorganisation structurelles des partenaires. Des exemples identiques peuvent être trouvés au sein des institutions de recherche nationale. Citons la réduction du nombre de bourses de thèses qui a pu être compensée par des bourses industrielles, APACHE étant, par pure chance, en phase de transfert industriel.