ACI JEUNES CHERCHEUSES ET JEUNES CHERCHEURS

Appel à propositions 2002
(à transmettre en 2 exemplaires)

I. FICHE D'IDENTITE DU PROJET

Numéro de référence du projet :

JC9126 (décision 03 5 1152 du 2 septembre 2003)

Titre du projet :

Création d'une équipe de recherche en informatique sur la qualité du logiciel numérique et l'arithmétique des ordinateurs

Coordonnateur du projet :

Philippe LANGLOIS

Tél du coordonnateur du projet :

04 68 66 21 35

Mél du coordonnateur du projet:

langlois@univ-perp.fr

Laboratoire de rattachement du coordonnateur du projet:

Laboratoire ELIAUS : Electronique, Informatique, Automatique et Systèmes

Adresse postale du laboratoire :

Université de Perpignan
52 avenue Paul Alduy
66860 Perpignan cedex

Numéro d'unité :

EA 3679

Information de cadrage du projet :

Durée :

36 mois

Moyens obtenus dans le cadre de l'ACI (en euros TTC) :

Equipement : **10 000**
Fonctionnement : **36 000**
CDD : **3 000**

II. BILAN DETAILLE DU PROJET

1. Rappel des objectifs initiaux du projet

Renvoi à la fiche d'identification du projet (1/3) du dossier de candidature (page 3)

2. Rapport final

Renvois

- au compte rendu scientifique de fin d'opération (pages 4 à 23),
- à son Annexe (liste complète des publications entre 2003 et 2006, 4 pages)
- au rapport à mi-parcours (mai 2005, 19 pages).

3. Rapport financier

Renvoi à la Partie 3 du compte rendu scientifique de fin d'opération (page 18).

PROPOSITION ACI
« JEUNES CHERCHEUSES ET JEUNES CHERCHEURS » 2003

FICHE D'IDENTIFICATION (1/3)

Projet :

Titre du projet : **Création d'une équipe de recherche en Informatique à l'Université de Perpignan sur la *qualité du logiciel numérique et l'arithmétique des ordinateurs*.**

Chef de projet : **Philippe LANGLOIS**

N° d'identification : 9126

Résumé du projet (1/2 page maximum) :

Ph. Langlois, MCF HDR (07/01), classé premier sur un poste de Professeur à l'Université de Perpignan au concours de recrutement actuel, est actuellement le seul chercheur actif en Informatique dans cet établissement. Depuis son arrivée à l'Université de Perpignan en octobre 2001, il est membre d'un laboratoire de mathématiciens reconnus en analyse non linéaire (2 PR, 5 MCF, 2 doctorants, sections 25 et 26).

Avec l'appui de l'Université (Président : feral@univ-perp.fr, vice-président du Conseil Scientifique : polit@univ-perp.fr), Ph. Langlois propose de **créer, animer et développer une équipe de recherche en Informatique.** L'équipe proposée dans ce projet est à ce jour composée, outre le demandeur, de deux autres jeunes chercheurs : **D. Defour**, soutenance de thèse de l'ENS Lyon prévue en septembre 2003 et **S. Graillat**, allocataire normalien en première année de thèse à l'Université de Perpignan. Un second Professeur rejoindra l'équipe au 1er septembre 2003.

Le thème de recherche est l'**adéquation entre arithmétiques des ordinateurs et qualité numérique du logiciel scientifique**. Deux axes de travail sont définis pour les trois années à venir : l'amélioration automatique de la précision et le développement d'interactions entre calcul numérique et calcul symbolique. La caractéristique principale des approches développées est l'utilisation conjointe de propriétés fines d'arithmétique des ordinateurs et de techniques mathématiques de l'analyse en précision finie.

Les membres de l'équipe proposée sont bien insérés dans la communauté française de la thématique avec laquelle ils collaborent activement sur plusieurs sujets.

L'appui demandé pour 2003-2005 est de 49000 euros et d'une allocation de recherche pour septembre 2004.

COMPTE RENDU SCIENTIFIQUE DE FIN D'OPERATION
Action Concertée Incitative Jeunes Chercheurs 2003

Décision n°03 5 152 datée du 2 septembre 2003

Création d'une équipe de recherche en informatique sur la qualité du logiciel numérique et l'arithmétique des ordinateurs

Période concernée : 2003-2006

Résumé : Ce rapport présente les activités réalisées entre 2003 et 2006 dans le cadre de l'ACI jeunes chercheurs 2003 qui a conduit à la création de l'équipe de recherche DALI. Pour la période concernée, ce document inclus un rapport scientifique (Partie 1), une description des moyens humains et financiers impliqués (Partie 2) et un bilan quantitatif des activités scientifiques (Partie 3). La Partie 4 précise les objectifs scientifiques pour 2007-2010 et la liste complète des publications de l'équipe DALI entre 2003 et 2006 est présentée en Annexe.

Rédacteur : Philippe LANGLOIS, Professeur, responsable scientifique de l'ACI, U. de Perpignan.

DALI : Fiabilité numérique et haute performance en informatique

L'équipe de recherche DALI a été initiée et reconnue par une ACI jeune-chercheur (JC-9276, 2003-2006) du Ministère délégué à la Recherche. Durant le contrat quadriennal 2003-2006, la politique volontariste de la direction de l'UPVD a permis le développement de l'équipe DALI, dans un premier temps au sein du laboratoire MANO (Modélisation, Analyse Non linéaire, Optimisation) et depuis septembre 2004 au sein du laboratoire LP2A.Depuis septembre 2006, l'équipe DALI s'est associée aux deux équipes du laboratoire LP2A pour constituer le laboratoire ELIAUS : Electronique, Informatique, Automatique et Systèmes, reconnu comme EA 3679 pour le contrat quadriennal 2007-2010.

Une des forces de l'équipe DALI est l'unité de ses travaux de recherche qui visent à **améliorer la fiabilité et la performance des calculs**. Cet objectif est conforté par l'interaction, rare en France au sein d'une même équipe, d'experts en **arithmétique et architecture des ordinateurs**. Les projets de recherche en cours s'intéressent à la fois au matériel (processeurs généralistes, nouvelles architectures émergentes), au logiciel (arithmétique, fonctions élémentaires et précision), aux outils de certification (preuves formelles et assistants) et aux applications (algorithmique numérique, cryptographie, calcul formel, théorie du contrôle).

La structure pluridisciplinaire et la taille (des disciplines scientifiques) de l'UPVD incitent naturellement les enseignants-chercheurs en informatique de l'université à s'organiser en réseau, et ce depuis la création de l'équipe en 2003. L'équipe DALI travaille maintenant depuis le début de 2005 à la mise en place de **collaborations structurantes importantes en informatique**. Il existe déjà entre Perpignan, Montpellier, Toulouse, Barcelone, Gérone et Tarragone plusieurs actions et contacts thématiques. Cet aspect fait l'objet d'un PPF sur le projet Suréna proposé en partenariat avec l'Université Montpellier 2 dans le cadre.du contrat quadriennal 2007-2010.

Partie 1 : Rapport scientifique pour la période 2003-2006

Participants : P. Langlois, B. Goossens, M. Daumas (LIRMM), D. Defour, Ch. Nègre, D . Parello, V. Beaudenon (05-06), S. Graillat, N. Louvet.

L'équipe est constituée à ce jour de 2 professeurs (recrutés en 2003 et précédemment à l'INRIA et à Paris 7), 2 maîtres de conférences (recrutés en 2004 et 2005 et précédemment doctorants à l'ENS Lyon et à Paris 11), 2 doctorants (allocation couplée pour élève normalien en 2003 et allocation MRES en 2005) et d'un ATER financé au titre des dotations exceptionnelles pour la recherche en 2004 (précédemment doctorant à Montpellier 2).

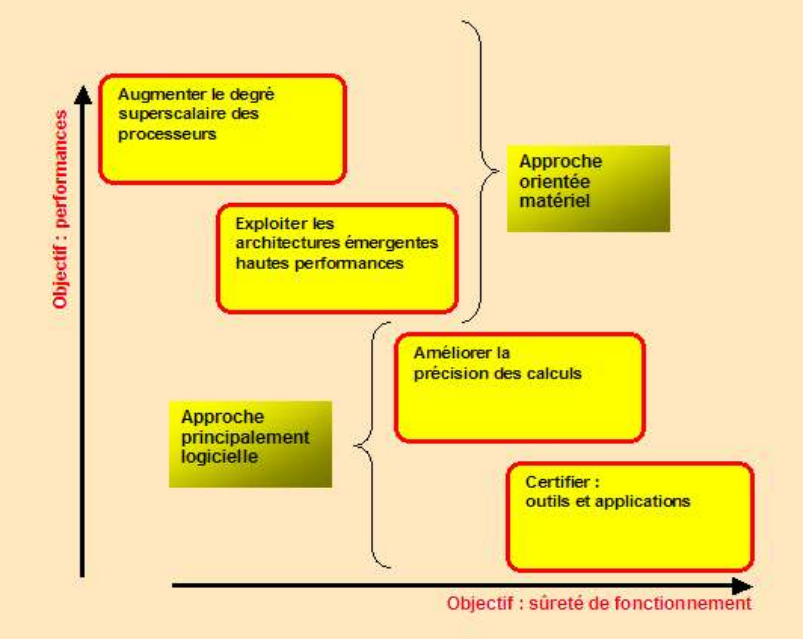
En se limitant par exemple aux revues internationales et aux actes de conférences internationales avec comité de lecture, 15 articles ont été publiés sur ces deux ans, soit 3 publications significatives par permanent et par an en tenant compte des arrivées. Cette expertise nous amène à assurer un cycle de cours depuis 2003 dans les Master recherche (anciennement DEA) d'informatique de l'UM2 (Ecole doctorale I2S de Montpellier) ainsi que de l'UPS (Toulouse).

L'année 2005-2006 confirme l'attractivité des travaux menés avec la mutation au LIRMM d'un chargé de recherche du CNRS et sa présence au sein de l'équipe DALI (voir paragraphes suivants) et le recrutement d'un maître de conférence en février 2006 (motivé par la qualité des candidatures au concours 2005).

L'équipe DALI se structure actuellement autour de 4 thèmes de recherche :

- Augmenter le degré superscalaire des processeurs,
- Exploiter les architectures émergentes haute-performances,
- Améliorer la précision des calculs,
- Certifier : outils et applications.

La figure suivante place les différents thèmes de recherche que nous allons détailler en fonction des apports attendus en terme de **sûreté de fonctionnement** et de **performances**.



Thème 1 : Augmenter le degré superscalaire des processeurs

Nouvelles micro-architectures haute performance.

Participants : David Defour, Bernard Goossens, David Parello.

La performance des micro-processeurs a connu une progression annuelle de 50% pendant deux décennies, soit un facteur 1500. Cela tenait à l'origine principalement à l'avancée technologique, ayant fait passer la finesse de gravure du millimètre au nanomètre. Deux avancées micro-architecturales majeures ont accéléré cette progression régulière de la performance. Tout d'abord, au milieu des années 80, l'organisation pipelinée des processeurs RISC a permis de quadrupler le nombre d'instructions exécutées par cycle (l'IPC est passé de 0,2 à 0,8) tout en multipliant la fréquence par quatre (à technologie égale, simplement en introduisant un pipeline à quatre étages).

Une dizaine d'années plus tard, l'ajout de la dimension superscalaire et surtout de l'exécution en désordre ont permis de doubler l'IPC (passant de 0,8 à environ 1,6) et de découper le pipeline plus finement pour en quadrupler le nombre d'étages (entre 10 et 20 étages), augmentant la fréquence d'horloge dans les mêmes proportions.

Mais, force est de constater que depuis une dizaine d'année, le degré superscalaire n'évolue plus et depuis deux ou trois ans, il en est de même du temps de cycle. Le peu de gain architectural est venu de l'utilisation du multithreading simultané[1] (ou hyperthreading dans la terminologie Intel). Il faut noter que les mesures effectuées par Eggers et Tullsen en 1995, qui faisaient apparaître une augmentation de l'IPC jusqu'à 8 threads, se basaient sur une microarchitecture employant un banc de registres architecturaux par thread alors que l'implémentation actuelle utilise un banc de registres de renommage partagé.

Cette panne de la progression des performances tient avant tout au fait que ce que l'on pourrait gagner en élargissant le degré superscalaire, on le perdrait probablement en étant contraint d'allonger le pipeline. Un allongement du pipeline accroît la pénalité des aléas de contrôle. Un tel allongement semble inévitable puisque l'augmentation du degré superscalaire implique une extension de la surface de la machine d'exécution en désordre et en premier lieu de son organe central qu'est le banc de registres de renommage. Aujourd'hui, le processeur Pentium 4 prend deux cycles pour accéder à son banc de registres entiers.

Il serait vain de compter sur le multithreading simultané pour franchir un nouveau seuil de performance. Au mieux, les aléas de contrôle et de données non masqués par l'exécution en désordre peuvent ils l'être par l'exécution d'un second thread. En revanche, le multithreading serait plutôt un frein à l'extensibilité superscalaire, renforçant la pression sur le banc de registres de renommage partagé. Enfin, il convient de remarquer que l'obstacle ne se situe pas dans le manque de parallélisme d'instruction: l'ILP moyen sur SPEC92 est de 80 instructions par cycle[2].

Le problème vient de la structure très centralisée de la machine d'exécution en désordre et de la taille de son banc de registres. Le banc de registres des machines à exécution en désordre a augmenté en surface, donc en temps d'accès, pour trois raisons. On a tout d'abord augmenté le nombre de registres pour capter plus d'ILP. On a parallèlement augmenté le nombre de ports d'accès pour la même raison. On a enfin doublé la largeur des données pour passer d'une

[1] S. Eggers, J. Emer, H. Levy, J. Lo, R. Stamm, D. Tullsen, Simultaneous multithreading: a foundation for next-generation processors, IEEE Micro, sept/oct 1997.

[2] J. Hennessy et D.A. Patterson, Computer architecture, a quantitative approach, Morgan Kaufmann, 3rd edition, 2003.

architecture 32 bits à une architecture 64 bits. La conséquence est qu'aujourd'hui, il faut deux cycles au Pentium 4 pour accéder à un registre entier. Augmenter le degré superscalaire conduit à augmenter le nombre de ports d'accès aux registres, ce qui allonge le pipeline. Comme nous l'avons dit un peu plus haut, on perd en longueur ce qu'on gagne en largeur.

Travaux et résultats

Le banc de registres d'instructions

Participant : Bernard Goossens.

Nous avons proposé un modèle d'extraction des instructions basé sur un banc de registres d'instruction. Le procédé permet de s'affranchir de la conservation des traces pour ne retenir que la description des blocs de base. Le banc de registres, par sa structure d'accès à ports multiples, permet de lire les blocs successifs pour construire les traces prédites.

Néanmoins une pièce essentielle manquait encore au dispositif pour être pleinement efficace: un producteur des adresses des blocs de base formant l'exécution du programme. Dans le cas du cache de trace[3], un prédicteur multi-sauts est employé pour fournir les directions des trois prochains sauts immédiats. L'adresse de leur cible n'est pas utile puisque le cache de trace fournit une trace déjà construite. En contrepartie, quand les directions prédites ne coïncident pas avec celles de la trace disponible, celle-ci doit soit être écartée, soit être coupée et l'adresse suivante calculée.

Il serait intéressant d'étudier un mécanisme de calcul ou de prédiction rapide (superscalaire) des adresses des blocs de base successifs. En particulier, la prédiction multi-sauts est encore peu efficace, donnant un taux acceptable de bonnes prédictions seulement pour le premier saut prédit.

L'ordonnancement distribué d'instructions

Participants : David Defour, Bernard Goossens.

Nous avons présenté un modèle d'exécution distribué des instructions, permettant de répartir les registres de renommage pour en réduire le temps d'accès. Pour cela, le lien producteur/consommateur entre les instructions et les registres a été inversé. Au lieu que ce soit les instructions qui adressent les registres, ce sont les registres qui produisent les numéros des instructions choisies pour lire ou écrire. Ainsi, chaque registre peut, par ses trois seuls ports, lire deux sources et écrire un résultat. Les registres sont regroupés en un nombre de bancs égal au degré superscalaire du processeur, donnant à l'ensemble une capacité d'accès d'un banc de registres traditionnel mais avec seulement trois ports par registre.

[3] E. Rotenberg, S. Bennett, J.E. Smith, Trace cache: a low latency approach to high bandwidth instruction fetching, Micro29, 1996.

Thème 2 : Exploiter les architectures émergentes hautes performances

Participants : David Defour, Marc Daumas (LIRMM), Bernard Goossens, Christophe Negre, David Parello.

Mots clés : carte graphique, GPU, GPGPU, virgule flottante, applications numériques.

Les processeurs graphiques sont principalement destinés aux calculs 3D issus des jeux et applications multimédia. Leurs performances ont été énormément améliorées depuis leur apparition dans les années 1990. Par exemple, la future PlayStation 3 embarquera un GPU (*Graphic Processing Unit*) programmable avec des performances de calcul flottant crête de 1.8 Tflops, soit 300 fois les 6.8 Gflops d'un Pentium 4 à 3 Ghz.

Cependant les applications multimédia ne constituent qu'une très faible proportion des tâches confiées à un ordinateur. Le GPU est donc sous-exploité. Cela explique l'émergence du *General-purpose computation on GPU* (GPGPU)[4] qui utilise la puissance de calcul et la forte bande passante des GPU pour d'autres applications. L'utilisation du GPU comme coprocesseur permet également de décharger le CPU d'une certaine quantité de travail et de bénéficier d'un « thread » supplémentaire quasi gratuitement. C'est pour ces raisons que l'on a vu apparaître des travaux liés à la cryptographie, la simulation, la fouille de donnée, la génomique et un peu d'algorithmes numériques[5].

Les unités de calcul des GPU travaillent sur des flots de données organisées en points et en fragments. Cette organisation des données ainsi que l'organisation très pipelinée des unités de calcul conditionne la recherche de nouveaux algorithmes et implémentations ciblant ces processeurs.

Cette thématique est récente. En septembre 2005, nous avons acquis une carte équipée d'un processeur Nvidia Geforce 7800 GTX. Il est capable d'effectuer 200 milliards d'opérations flottantes par seconde. De plus, il présente de nombreuses caractéristiques parmi lesquelles on trouve une bonne gestion des branchements et de nombreux registres qui devraient préfigurer du futur des GPU. Ce processeur nous permet de conduire des expérimentations sur la possibilité de transférer des applications cryptographiques du CPU vers le GPU en visant plus particulièrement une implémentation de IDEA[6].

Il est connu que les caractéristiques flottantes des GPU ne respectent pas la norme IEEE-754[7]. Nous nous intéressons à des méthodes permettant de caractériser le comportement des opérateurs flottants, y compris les opérateurs complexes comme le FMA ou les fonctions élémentaires. Notre objectif est de proposer des opérateurs logiciels permettant de garantir un comportement proche de celui imposé par la norme IEEE-754.

---

[4] **GPU Gems 2 - Programming Techniques for High-Performance Graphics and General-Purpose Computation**, Matt Pharr, 2005, Addison Wesley.

[5] Dinesh Manocha, **General-Purpose Computations Using Graphics Processors**, Computer, 38(8), pp. 85-88, 2005.

[6] **CryptoGraphics: Secret Key Cryptography Using Graphics Cards,** *Debra L. Cook, Angelos D. Keromytis, John Ioannidis, Jake Luck,* In the Proceedings of the RSA Conference, Cryptographer's Track (CT-RSA), LNCS 3376, Springer-Verlag, pages 334-350, February 2005, San Francisco, CA.

[7] **GPU floating-point paranoia,** *K. Hillesland, A. Lastra,* Proceedings of GP^2. 2004.

Thème 3 : Améliorer la précision des calculs

Participants : David Defour, Stef Graillat, Philippe Langlois, Nicolas Louvet

Plus de précision : pourquoi et comment

La précision du résultat d'un calcul en précision finie dépend de trois facteurs : le conditionnement du problème à résoudre, la stabilité de l'algorithme numérique et la précision de l'arithmétique utilisée. Cette dernière est souvent l'arithmétique flottante binaire IEEE-754 qui permet une précision maximale de l'ordre de 16 chiffres décimaux en format long (double précision). Cette précision n'est pas suffisante si on cherche une solution précise de problèmes mal conditionnés. En effet, le nombre de chiffres décimaux perdus par les (bons) algorithmes numériques évolue comme l'ordre de grandeur du conditionnement du problème : il ne subsiste plus que 7 chiffres significatifs dans la solution calculée en double précision IEEE d'un problème dont le conditionnement est de l'ordre de 10^{9}.

Plusieurs solutions permettent d'augmenter la précision de l'arithmétique utilisée. Une quadruple précision est proposée par quelques constructeurs (Sun) mais est hélas non portable. Des bibliothèques qui simulent une arithmétique flottante en précision arbitraire existent depuis assez longtemps (MP[8], ARPREC et MPFUN90[9]). La plupart de ces bibliothèques représentent les quantités flottantes par autant d'entiers que nécessaire. La norme IEEE-754 ayant réduit la complexité de l'arithmétique flottante, la bibliothèque de précision arbitraire MPFR développée par l'INRIA MPFR[10] étend le modèle IEEE en précision arbitraire. Disposer de bibliothèques implantant les fonctions transcendantes élémentaires de la meilleure façon pour une précision arbitraire est actuellement une des priorités importantes.

Certaines solutions mettent l'accent sur la rapidité (au détriment d'une précision arbitraire) en se limitant à une augmentation de précision fixée et en profitant des performances des unités matérielles flottantes. Une précision supérieure à la double précision IEEE était considérée par la plupart comme anecdotique jusqu'à très récemment. La nécessité de calculs intermédiaires en précision double de la précision courante est illustrée[11] par quelques algorithmes numériques très utilisés et ce afin de justifier le développement de BLAS en précision étendue (X-BLAS : eXtended Basic Linear Algebra Subroutines). Les réalisations les plus abouties sont proposées par un groupe de chercheurs de Berkeley autour des travaux de D.H. Bailey : les bibliothèques ``double-double'' et ``quad-double'' implantent respectivement deux ou quatre fois la double précision IEEE. L'arithmétique flottante est alors redéfinie sur des quantités représentées par la somme de deux ou quatre flottants en double précision (expansions de longueur deux ou quatre).

Toutes les solutions pour augmenter la précision présentées précédemment sont **logicielles**. Elles augmentent donc considérablement les temps de calcul : comptez par exemple un facteur 10 pour doubler la précision avec la bibliothèque ``double-double'' de Berkeley.

---

[8] Richard P. Brent. A fortran multiple-precision arithmetic package. *ACM Trans. Math. Softw.*, 4(1):57-70, 1978.

[9] http://crd.lbl.gov//~dhbailey/mpdist.

[10] http://www.mpfr.org/.

[11] Xiaoye S. Li, James W. Demmel, David H. Bailey, Greg Henry, Yozo Hida, Jummy Iskandar, William Kahan, Suh Y. Kang, Anil Kapur, Michael C. Martin, Brandon J. Thompson, Teresa Tung, and Daniel J. Yoo. Design, implementation and testing of extended and mixed precision BLAS. *ACM Transactions on Mathematical Software*, 28 (2):152-205, June 2002.

Travaux et résultats obtenus :

Motivation générale et originalité de l'approche développée

Notre objectif principal est de proposer une **alternative aux « double-double »** qui soit à la fois **performante, fiable** et **portable**.

En matière de performance, nous développons des algorithmes dont la solution est d'une précision similaire à celle qui serait obtenue avec une précision deux fois supérieure à la double précision IEEE, et ce en un temps significativement inférieur à celui fourni par les « double-double ».

Coté fiabilité, ces algorithmes sont accompagnés de deux bornes d'erreur. L'une, statique, est établie formellement *a priori* et fournit une majoration de l'erreur dans le pire cas qui prouve un comportement en précision deux fois supérieure. L'autre permet de majorer la précision de la solution calculée à la fois de façon dynamique et certifiée : cette borne, calculée avec l'arithmétique flottante du processeur, est un majorant effectif de l'imprécision résiduelle.

Afin de garantir la portabilité maximale, les algorithmes et les majorations statiques et dynamiques d'erreur proposés utilisent principalement le modèle de l'arithmétique IEEE-754. De façon complémentaire, les optimisations apportées par l'opérateur *fma* (présent dans la famille Intel IA et l'IBM RS6000) sont aussi considérées.

Chaque opération arithmétique (+,-,x,/,sqrt,*fma*) introduit une erreur d'arrondi élémentaire qui s'accumule en l'erreur globale de la solution d'un algorithme numérique. Notre approche consiste à corriger, de façon adaptée à chaque algorithme, la contribution d'ordre 1 de ces erreurs élémentaires dans l'erreur globale. Les propriétés des erreurs d'arrondi élémentaires et la mise en oeuvre de la correction permet la majoration théorique *a priori* de l'erreur résiduelle. Cette correction est accompagnée d'une majoration dynamique de l'erreur résiduelle.

Nous spécialisons ainsi la méthode de **correction automatique CENA**[12] **en des algorithmes compensés**.

La correction automatique des erreurs d'arrondis CENA

Participant : Philippe Langlois

L'intérêt de la correction des erreurs d'arrondi élémentaires pour améliorer la précision des algorithmes numériques est assurée par les résultats que nous avons obtenu avec la méthode de correction automatique CENA. Il en est de même pour la validation dynamique de la précision du résultat ainsi corrigé [27].

[12] Philippe Langlois. Automatic linear correction of rounding errors. *BIT*, 41(3):515-539, September 2001.

En effet, CENA automatise la correction des termes d'ordre 1 de l'erreur globale grâce à la différentiation automatique de l'algorithme et au calcul des erreurs d'arrondi élémentaires. La majoration dynamique de l'erreur résiduelle est obtenue par majoration dynamique de l'ensemble du processus de correction (différentiation automatique, calcul des erreurs élémentaires et des majorations éventuelles, correction). Les algorithmes corrigés exhibent un comportement numérique similaire à une exécution en précision double de la précision courante. Un tel comportement compensé est expliqué sans pouvoir être prouvé formellement dans le cadre général de la correction automatique. Parce qu'elle automatise la correction (l'algorithme est considéré comme une boîte noire), CENA est une méthode d'investigation où les performances sont secondaires.

La majoration dynamique de l'erreur résiduelle permet aussi d'invalider *a posteriori* le résultat non corrigé d'une certaine classe d'algorithmes. Nous avons identifié ces algorithmes qui admettent une erreur globale linéaire par rapport aux erreurs élémentaires. Certains algorithmes de base sont linéaires au sens de la correction CENA : citons, le produit scalaire, l'évaluation polynomiale et la résolution de système linéaire triangulaire.

Des algorithmes compensés pour l'évaluation polynomiale et la résolution de systèmes triangulaires

Participants : Stef Graillat, Philippe Langlois, Nicolas Louvet

L'instance de la correction CENA aux algorithmes qui propagent linéairement les erreurs d'arrondis génère **des algorithmes compensés**, c'est-à-dire qui permettent un résultat calculé d'une précision équivalente à celle obtenue avec une arithmétique deux fois plus précise. L'intérêt de la spécialisation d'une telle correction a été démontré récemment dans le cas du produit scalaire[13]. Une majoration du type ``erreur directe égale au conditionnement fois la précision au carré'' est prouvée ; les performances obtenues sont meilleures que le sur-coût théorique. Ce dernier point est justifié par les capacités super-scalaires des processeurs actuels.

Ces deux aspects justifient nos travaux actuels. Nous proposons le traitement complet de l'évaluation polynomiale par le schéma de Horner [155] et des premiers résultats sur la résolution de systèmes triangulaires. Le schéma de Horner compensé permet un calcul deux fois plus précis que celui donné par la précision courante, et ce pour un sur-coût raisonnable et deux fois moins élevé que les implantations optimisées des « doubles-double ». Cet algorithme est accompagné d'un calcul dynamique de borne d'erreur résiduelle, calcul certifié valide en arithmétique flottante. Le comportement compensé de cet algorithme est formellement prouvé. L'optimisation de ces résultats selon la disponibilité au non d'un opérateur fma sont détaillés dans [109]. Ce traitement complet du schéma de Horner constitue le résultat le plus remarquable de ces travaux [54].

Multiprécision et fonctions élémentaires :

Participant : David Defour

La précision fournie par la représentation machine est finie et fixée. Or, pour certaines applications, comme la cryptographie, ou un cas qui nous concerne plus, l'évaluation des fonctions élémentaires, cette précision n'est pas suffisante. Plusieurs solutions existent pour faire face à ce problème. Il est possible d'utiliser des logiciels mathématiques tels que Maple ou Mathematica. Ces logicielles présentent de nombreuses fonctionnalités mais sont payants, gourmands en

[13] Takeshi Ogita, Siegfried M. Rump, and Shin'ichi Oishi. Accurate sum and dot product. *SIAM J. Sci. Comput.*, 2005.

ressources et leur fiabilité n'est pas absolue. Le bug de Maple version 6 est un des exemples les plus évident. Pour des applications où la précision est le critère décisif, il est possible d'utiliser des bibliothèques multiprécisions dont la plus connue est GMP. Les opérateurs logiciels proposés par ces bibliothèques sont soit lents soit faciles à faire évoluer, mais rarement les deux. En effet, la principale façon d'obtenir des opérateurs performants est d'exploiter les caractéristiques matérielles des processeurs modernes accessibles avec de l'assembleur. Mais l'assembleur est est un langage peu évolutif et nécessite de gros investissement pour le développement logiciel.

La principale différence entre chaque bibliothèque multiprécision est le format de représentation interne des nombres multiprécisions. Nous proposons une bibliothèque composée d'opérateurs multiprécision dont le format de représentation ne nécessite pas d'accéder aux caractéristiques matérielles pour être performant. En effet, l'opération la plus coûteuse des opérateurs multiprécisions est la propagation de retenue. Cette opération nécessaire dans les autres bibliothèques casse le parallélisme disponible et requière un contrôle en langage assembleur pour conserver de bonne performance. Le format que nous utilisons permet de différer les propagations de retenues lors des additions et des multiplications et donc d'accélérer les algorithmes multiprécisions en les rendant simples, portables et efficaces. Cette bibliothèque appelée SCSlib (Software Carry Save) est aujourd'hui distribuée sous licence LGPL[14][15] et dans un souci de performance tient compte des caractéristiques des processeurs et des compilateurs actuels.

Cet axe de recherche, dans la continuité des travaux de thèse de David Defour, a naturellement accompagné son intégration comme ATER (09/03) puis maître de conférences (09/04) au sein de l'équipe DALI. L'orientation des travaux actuels de David Defour (voir thème Exploiter les architectures émergentes hautes performances) ne permettent pas de continuer à développer cet axe lors du contrat 2007-2010.

[14] D. Defour et F. de Dinechin, « Software Carry-Save : A case study for instruction-level parallelism », 7th Conference on Parallel Computing Technologies, PaCT-2003, pp 207-214, septembre 2003.

[15] D. Defour et F. de Dinechin, « Software carry-save for fast multiple-precision algorithms », 35th International Congress of Mathematical Software, ICMS, pp 29-40, août 2002.

Thème 4 : Certifier : outils et applications

Participants : Vincent Beaudenon, Marc Daumas (LIRMM), Stef Graillat, Philippe Langlois, Matthieu Martel (CEA).

Mots clés : virgule flottante, borne d'erreur, validation, arithmétique d'intervalles, méthodes formelles, Coq, PVS, Fluctuat.

On ne peut pas effectuer des calculs numériques (à virgule flottante) sans risque de rendre catastrophiques des erreurs d'arrondi ni de faire apparaître des situations exceptionnelles. Notre démarche vise donc à certifier formellement les programmes dont le comportement est convenable.

En l'absence de certification contraignante, cette dernière notion reste floue. L'exemple de l'échec du vol Ariane 501[16] montre, s'il en était besoin, qu'un programme peut être adapté numériquement à une situation mais pas à une autre. Notre travail vise aussi à faire disparaître ce flou dans les spécifications et à le rendre inacceptable pour des entreprises qui développent des logiciels critiques en terme de vies humaines ou des projets coûteux.

Il est vrai que les entreprises qui se sont engagées dans une véritable démarche de certification de leurs développements sont encore trop rares[17]. Notons que seuls deux projets[18] ont atteint le niveau maximal de qualité (EAL7) défini dans les critères communs et aucun d'eux ne fait usage d'algorithmes numériques.

Dans la pratique, les petits programmes au comportement numérique simple peuvent facilement et efficacement être certifiés avec l'arithmétique d'intervalles et un outil de preuve certifiée tel que Coq[19] ou PVS[20]. Il faut dans le même temps développer de petits programmes au comportement numérique évolué qui pourront être certifiés. Les outils de certification étant encore limités par leur puissance et leurs bibliothèques, seuls certains travaux peuvent être certifiés et il est important de mettre en balance les efforts nécessaires avec les retombées attendues pour chaque nouveau développement. Dans certains cas, la spécification ne pourra se faire qu'en introduisant de nouvelles notions comme les pseudozéros. Enfin, les programmes plus gros doivent être analysés par des outils sémantiques puissants tels que Fluctuat[21] avant de pouvoir être certifiés.

La génération automatique de preuve certifiée avec les outils Coq ou PVS

Les résultats validés avec Coq ou PVS sont exempts d'erreurs. Pour une entreprise vraiment engagée dans une démarche de qualité numérique, le nombre de défaillances dans son développement est probablement très petit. Elle ne voudra pas utiliser un outil qui peut être lui-

[16] J.-L. Lions *et al*., **Ariane 5 flight 501 failure report by the inquiry board**, tech. rep., European Space Agency, Paris, France, 1996.

[17] P. E. Ross, **The exterminators**, *IEEE Spectrum*, vol. 42, no. 9, pp. 36-41, 2005.

[18] http://newsroom.slb.com/press/newsroom/index.cfm?PRID=16261 et http://www.rockwellcollins.com/news/page6237.html.

[19] G. Huet, G. Kahn, et C. Paulin-Mohring, **The Coq proof assistant: a tutorial: version 8.0**, 2004.

[20] S. Owre, J. M. Rushby, et N. Shankar, **PVS: a prototype verification system**, in *11th International Conference on Automated Deduction* (D. Kapur, ed.), (Saratoga, New-York), pp. 748-752, Springer-Verlag, 1992.

[21] M. Martel, **Propagation of roundoff errors in finite precision computations: a semantics approach**, in *11th European Symposium on Programming*, (Grenoble, France), pp. 194-208, 2002.

même défaillant et ne pas détecter les quelques situations alarmantes qui pourraient rester dans le code. L'utilisation d'un outil de validation comme Coq ou PVS permet de garantir qu'aucun problème n'est indûment écarté.

Il est humainement impossible de se prémunir de toute erreur. Une erreur, si elle devait apparaître remettrait en cause la formalisation synthétique[22] et cela serait en soit un résultat intéressant. Nous n'osons pas émettre la possibilité que l'outil de validation des preuves soit incorrect, mais une telle possibilité sera encore plus intéressante scientifiquement.

Preuves et certification d'algorithmes évolués

Le développement et la preuve d'algorithmes numériquement évolués ne peuvent pas être automatisées telles celle de W. Kahan[23] et d'étudiants autour de lui à l'University of California à Berkeley qui sont aujourd'hui abondamment référencés tels que de P. Sterbenz[24], J. Coonen[25] ou D. Priest[26]. En ajoutant D. Knuth[27] à cette longue liste, on peut être tenté de croire que ce sujet ne peut être traité qu'autour de la baie de San Francisco ! Ce serait exclure les travaux novateurs de O. Moller[28] et T. Dekker[29], mais il semble qu'à quelques rares exceptions cette constatation empirique se vérifie.

Dans la pratique, les algorithmes et leur preuve de fonctionnement sont extrêmement ardus, non que les preuves soient particulièrement difficiles mais parce qu'une petite omission peut ruiner de longs développements. Ce risque est renforcé par le fait que l'arithmétique à virgule flottante regorge de pièges. Ainsi, on définit souvent un ulp comme « la distance entre les deux plus proches voisins représentables d'un nombre réel non représentable en machine ». Nous avons volontairement tronqué cette définition comme ce fut le cas par erreur dans la liste de discussion électronique du comité de révision de la norme IEEE 754. En fait un ulp est « la distance entre les deux plus proches voisins représentables de part et d'autre d'un nombre réel non représentable en machine ». Une omission de ce type peut aboutir à l'usage non justifié de théorèmes dont les hypothèses ne sont pas vérifiées.

Par la suite, il est difficile de faire confiance à un résultat nouveau sur ce sujet. Même des articles unanimement respectés tel celui de P. Tang[30] contiennent au moins une petite erreur. Il semble

---

[22] M. Daumas, L. Rideau, et L. Théry, **A generic library of floating-point numbers and its application to exact computing**, in *14th International Conference on Theorem Proving in Higher Order Logics*, (Edinburgh, Scotland), pp. 169-184, 2001.

[23] W. Kahan, **Further remarks on reducing truncation errors**, *Communications of the ACM*, vol. 8, no. 1, p. 40, 1965.

[24] P. H. Sterbenz, **Floating point computation**, Prentice Hall, 1974.

[25] J. T. Coonen, **Specification for a proposed standard for floating point arithmetic**, Memorandum ERL M78/72, University of California, Berkeley, 1978.

[26] D. M. Priest, **Algorithms for arbitrary precision floating point arithmetic**, in *Proceedings of the 10th Symposium on Computer Arithmetic* (P. Kornerup et D. Matula, eds.), (Grenoble, France), pp. 132-144, 1991.

[27] J. F. Reiser et D. E. Knuth, **Evading the drift in floating point addition**, *Information Processing Letters*, vol. 3, no. 3, pp. 84-87, 1975.

[28] O. Moller, **Quasi double-precision in floating point addition**, *BIT*, vol. 5, no. 1, pp. 37-50, 1965.

[29] T. J. Dekker, **A floating point technique for extending the available precision**, *Numerische Mathematik*, vol. 18, no. 3, pp. 224-242, 1971.

[30] P. T. P. Tang, **Table driven implementation of the exponential function in IEEE floating point arithmetic**, *ACM Transactions on Mathematical Software*, vol. 15, no. 2, pp. 144-157, 1989.

alors raisonnable de penser qu'un développement d'une dizaine de pages non certifié contient au mieux quelques erreurs et une preuve n'est plus suffisante. Ces petites erreurs, que le lecteur attend, font naître le doute dans son esprit. Sont-elles mineures ou remettent-elles en cause le résultat final ?

De nouveaux problèmes apparaissent quand on essaie de répondre à des problèmes exacts avec des algorithmes approchés. L'arithmétique à virgule flottante étant très efficace, la tentation est forte de l'utiliser dans des algorithmes du calcul formel. Quelle signification peut-on donner au résultat si l'on est rigoureux et si l'on n'accepte pas le flou et l'à peu près ?

Travaux et résultats

Symétries additives dans la structure de l'arithmétique à virgule flottante

Participants : Marc Daumas, Philippe Langlois.

Cette étude, très théorique, vise à mieux appréhender le rôle particulier du résultat 0 dans la méthode CENA de P. Langlois. Un symétrique additif b de a par rapport à c vérifie c = (a+b)/2. L'existence et l'unicité d'un tel b est une propriété de base en arithmétique exacte qui disparaît quand a et b sont des nombres à virgule flottante et quand le calcul de c est effectué dans une arithmétique de type IEEE-754. Nous présentons et nous prouvons des conditions sur l'existence, l'unicité et l'égalité avec le symétrique exact dans le cas où b et c sont de même signe.

Pseudozéros de polynômes

Participants : Stef Graillat, Philippe Langlois.

Étant donné ε > 0, l'ensemble des ε-pseudozéros d'un polynôme p est l'ensemble des zéros de tous les polynômes se trouvant à une distance (dans un sens à définir) inférieure à ε de p. Il s'agit d'un outil introduit par Mosier[31] en 1986 afin d'étudier la sensibilité des zéros de polynômes par rapport à des perturbations sur les coefficients. Les pseudozéros ont ensuite été étudiés par Trefethen et Toh[32] qui ont comparé la notion de pseudozéros à celle de pseudospectres de la matrice compagnon. Plus récemment, Stetter[33] a consacré un chapitre de son livre à cette notion.

Applications des pseudozéros en théorie du contrôle

Motivation : En théorie du contrôle et en automatique, on écrit classiquement les fonctions de transfert sous la forme H(p) = N(p)/D(p), ou N et D sont deux polynômes et ou p est le paramètre du système. Le système décrit par cette fonction de transfert est dit *stable* (dans le sens de Hurwitz) si tous les zéros de D sont à partie réelle négative (autrement dit si le polynôme D est stable au sens de Hurwitz). Puisque des incertitudes sur les coefficients sont inévitables dans les problèmes de la vie réelle (incertitudes sur les données, erreurs d'arrondi), il est utile de mesurer la distance d'un système stable au système instable le plus proche. Il s'agit en fait de mesurer la distance de D au polynôme instable le plus proche. Cette distance s'appelle le rayon de stabilité de D.

[31] Mosier, Ronald G. Root neighborhoods of a polynomial. *Math. Comp.* **47** (1986), no. 175, 265--273.

[32] Toh, Kim-Chuan; Trefethen, Lloyd N. Pseudozeros of polynomials and pseudospectra of companion matrices. *Numer. Math.* **68** (1994), no. 3, 403--425.

[33] Stetter, Hans J. Numerical polynomial algebra. *Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA,* 2004.

Contribution : Nous proposons un algorithme de calcul du rayon de stabilité. Le point clé de notre algorithme est l'utilisation des pseudozéros. L'algorithme proposé est de type symbolique-numérique (on voit aussi le terme d'algorithme hybride). Cela signifie que l'on utilise des méthodes formelles (ici les suites de Sturm) dans des procédures numériques. Une telle approche a été initiée[34] pour compter le nombre de valeurs propres imaginaires pures d'une matrice Hamiltonienne. Il semble néanmoins que de telles approches aient été très peu étudiées bien qu'elles fournissent des algorithmes efficaces et précis.

Zéros de polynômes d'intervalles

Motivation : Le concept de l'analyse par intervalles[35] est de calculer avec des intervalles de nombres réels plutôt qu'avec les nombres réels eux-mêmes. Tandis que l'arithmétique flottante est affectée par les erreurs d'arrondis et donc peut conduire à des résultats imprécis et faux, l'analyse par intervalles a l'avantage de donner des bornes rigoureuses pour la solution exacte. Une telle analyse se révèle très utile quand par exemple les paramètres ne sont connus qu'avec une certaine incertitude. Dans ce cas, on peut implémenter des algorithmes en utilisant l'arithmétique d'intervalles pour les paramètres incertains afin de produire un intervalle qui contient tous les résultats possibles.

Contribution : Nous nous intéressons à la notion de polynômes d'intervalle. Il s'agit de polynômes dont les coefficients ne sont plus des nombres réels ou complexes mais des intervalles réels. Cela va nous permettre de modéliser des incertitudes sur les coefficients d'un polynôme. Nous nous intéressons aux pseudozéros réels de polynômes réels avec des perturbations mesurées en norme infinie. Cela nous permet de donner une formule calculable et un outil pour tracer les pseudozéros d'un polynôme d'intervalle.

[34] Boyd, S.; Balakrishnan, V.; Kabamba, P. A bisection method for computing the H_∞ norm of a transfer matrix and related problems. *Math. Control Signals Systems* **2** (1989), no. 3, 207--219.

[35] L. Jaulin, M. Kieffer, O. Didrit, et E. Walter, **Applied interval analysis**, Springer, 2001.

Partie 2 : Moyens sur la période 2003-2006

Personnel permanent au 1er janvier 2003

Néant

Arrivées de membres permanents et doctorants entre septembre 2003 et décembre 2006

Septembre	PR 27	MCF 27	CR CNRS	Doctorant
2003	B. Goossens P. Langlois			S. Graillat
2004		D. Defour		N. Louvet
2005		D. Parello	M. Daumas (CR1 CNRS LIRMM, HDR) (01/06)	
2006		Ch. Nègre		

Doctorant ayant soutenu entre le 1/10/2003 et le 1/10/2006

Stef GRAILLAT. Thèse de doctorat, Université de Perpignan, Novembre 2005.

Titre : Fiabilité des algorithmes numériques : pseudosolutions structurées et précision
Rapporteurs : G. Villard (DR CNRS), S.M. Rump (PR, Allemagne)

Jury : M. Daumas (CR CNRS), Ph. Langlois (PR), S.M. Rump (PR, Allemagne), G. Villard (DR CNRS), E. Walter (DR CNRS), P. Zimmerman (DR INRIA).

Direction : Philippe LANGLOIS.

Financement : allocation couplée pour élève normalien.

Stef GRAILLAT est maître de conférences à l'Université Paris 6 depuis septembre 2006.

Doctorant en cours entre le 1/10/2003 et le 1/10/2006

Nicolas LOUVET. Doctorant depuis octobre 2004, soutenance prévue en octobre 2007.

Titre : Amélioration de la précision des algorithmes numériques par compensation
Rapporteurs : J.M. Muller (DR CNRS), N.J. Higham (PR, Grande Bretagne)

Direction : Philippe LANGLOIS.

Financement : allocation atribuée dans le cadre de l'ACI JC (03 5 152).

Utilisation des crédits entre le 1/10/2003 et le 1/10/2006

Type de dépense	*2003-2004* *TTC en €*	*2005* *TTC en €*	*2006* *TTC en €*
Equipements informatiques	10205	1527	3125
Logiciels	2250		
Missions	6824	13738	13000
Fournitures et matériels de bureau	1140	1254	4500
Bibliographie : livre, abonnements, photocopies.	1578	314	
Vacations			700
Total	**22477**	**16833**	**20625**

Autres ressources que l'ACI JC sur la période 2003-2006 (en euros TTC) :

- BQR de l'UPVD : 3500 (2004), 2000 (2005).
- Reversion fraction de la dotation la laboratoire MANO (plan quadriennal 03-06) : 5574 (2004).
- Aide région (complément 50% équipement) : 1400 (2004).

Partie 3 : Bilan quantitatif pour la période 2003-2006

Publications

La liste complète des publications est annexée en fin de ce rapport.

Articles dans des revues avec comité de lecture	Internationales	ACLI	9
	Nationales	ACLN	1
Articles dans des revues sans comité de lecture		SCL	0
Conférences invitées		INV	1
Communications avec actes	Internationales	ACTI	27
	Nationales	ACTN	2
Autres publications		AP	13

Thèses et habilitations à diriger les recherches soutenues

	2003 (après oct)	2004	2005	2006
Thèses			1	
Habilitations				

Collaborations

Collaborations universitaires

Etranger

- Technical University Hamburg-Harburg (Allemagne)
- Université de Girona (Espagne)
- Université de Tarragona (Espagne)
- Université de Californie Irvine (EU)
- University of Manchester (GB),
- Chuo University, Tokyo, (Japon)
- Université de Novosibirsk (Russie)

France

- Université Pierre et Marie Curie (LIP6, Paris)
- Université Montpellier 2 (LIRMM)
- Université Paul Sabatier (IRIT, Toulouse)
- CNRS-INRIA-ENS-Lyon, Projet Arénaire (LIP, Lyon)
- CNRS-INRIA-PCRI, Projet ALCHEMY (LRI, Orsay)
- CEA (LIST, Saclay)

Eléments de visibilité

Organisation de conférences :

- Présidence de la rencontre francophone conjointe Sympa-Renpar-CFSE 2006, Canet-en-roussillon, octobre 2006 et présidence du comité de programme de Sympa 2006.
- ACM SAC'06, « More accurate computation : methods and software », Dijon, avril 2006 : co-organisation P. LANGLOIS et S.M. RUMP (TU Hamburg-Harburg).
- Journées Arinews, Banyuls, novembre 2003
- Journées Arinews, Perpignan, novembre 2005.

Partie 4 : Objectifs scientifiques pour la période 2007-2010

Pour la période 2007-2010, l'équipe DALI est intégrée au sein du laboratoire ELIAUS, EA 3679 (DS STIC), Université de Perpignan.

Équipe DALI : Sûreté numérique et haute performance en informatique

Augmenter le degré superscalaire des processeurs

Participants : D. Defour, B. Goossens, D. Parello

Notre sentiment aujourd'hui est qu'il faut, à l'instar de ce qui fut fait dans les années 80 avec les architectures RISC, adapter l'architecture aux innovations micro-architecturales des machines à exécution en désordre. Le modèle à registre, adapté à un modèle d'exécution chargement/rangement, ne l'est pas aussi bien au modèle flot de donnée.

On constate en particulier que le compilateur et la micro-architecture se livrent à deux manipulations antagonistes du programme. Le compilateur mappe l'espace mémoire du programme sur un ensemble très réduit de registres, introduisant dans le code des opérations de chargement et de rangement des registres en mémoire. La micro-architecture, à l'exécution, expanse l'espace des registres en les renommant.

Le mécanisme de renommage est assez lourd à mettre en œuvre, avec une allocation, un pointage spéculatif et un pointage validé et une libération. Tout cela est, en définitive, très peu extensible, bridant les micro-architectures superscalaires.

Par ailleurs, les chargements et les rangements, qui peuvent représenter plus de 50% des instructions exécutées, sont des instructions totalement parasites. La mémoire hiérarchisée est depuis longtemps adaptée à un fonctionnement entièrement matériel, sans qu'il soit besoin de passer par l'intermédiaire d'un registre. La centralisation du banc de registres pourrait ainsi disparaître au profit d'un ensemble distribué de bancs mémoires.

La suppression des registres aurait de très nombreux avantages: simplification de l'algorithme d'exécution en désordre par la suppression du renommage, simplification du travail du compilateur en lui évitant l'allocation de registres, simplification du travail du système lors des changements de contextes. Le multithreading simultané pourrait être obtenu à coût quasi nul (un compteur de programme et un pointeur de pile), ce qui le rendrait plus extensible.

Nous avons montré comment on peut distribuer une instruction dans la micro-architecture, envoyant les sources et la destination vers leurs unités d'accès et l'opération vers son unité fonctionnelle. Le passage a`un modèle sans registre remplace le banc de registres distribué par un ensemble de bancs mémoires formant le premier niveau de la hiérarchie. Le renommage disparaît. La reprise de l'exécution après une fausse prédiction se simplifie: elle s'obtient en vidant le store buffer, les chargements spéculatifs invalidés servant la plupart du temps de mécanisme de prefetching.

Pour valider cette proposition, nous devons définir un nouveau jeu d'instructions mémoire-mémoire. Ensuite, nous devons proposer une implantation micro-architecturale en faisant ressortir le nouvel étagement du pipeline. Enfin, nous devons mesurer la performance potentielle et mettre en évidence l'extensibilité de notre solution comparativement au modèle registre-registre.

Exploiter les architectures émergentes hautes performances

Participants : David Defour, Marc Daumas (LIRMM), Bernard Goossens, Christophe Negre, David Parello.

De nombreux verrous subsistent à la démocratisation de GPU pour le calcul scientifique. Parmi ces verrous on retrouve le manque de communication de la part des fabricants sur le matériel, l'absence d'environnement de développement logiciel adapté ou l'absence d'opérations basiques avec un comportement certifié sur lesquels les développeurs peuvent s'appuyer.

Dans cette logique les processeurs graphiques ne proposent que des opérateurs flottants dont la précision maximale est proche de celle de la simple précision de la norme IEEE-754. Nous souhaitons très prochainement développer des schémas d'évaluations logiciels pour atteindre des opérateurs en précisions supérieures à celle disponible en matériel et notamment la double précision.

Nos premières investigations nous montrent que les GPU disposent d'unités de calcul évoluées pour certaines fonctions comme le sinus, cosinus, exponentielle… ainsi que le matériel pour l'interpolation de fonctions. Nous allons étudier comment de nouveaux schémas d'évaluation basés sur l'interpolation peuvent être proposés.

Les GPU vont certainement devenir les co-processeurs numériques de demain. Le bug du Pentium montre que l'arithmétique à virgule flottante est difficile à manier et à implémenter sur ordinateur. Cela est lié à la complexité des architectures des processeurs généralistes (CPU). Au contraire, les GPU sont construits autour d'une architecture plus simple, un long pipeline, ce qui laisse envisager la validation d'algorithmes basée sur l'outil de preuve formelle COQ.

Améliorer la précision des calculs

Participants : David Defour, Philippe Langlois, Nicolas Louvet

Nous étendrons les avantages des algorithmes compensés à la résolution des systèmes linéaires triangulaires, brique importante des bibliothèques d'algèbre linéaire numérique (BLAS et LAPACK).

Un algorithme compensé de résolution de systèmes triangulaires a été introduit [136]. Les premières mesures de performances confirment toutes les attentes. Cette remontée compensée concurrence très favorablement les meilleures solutions actuellement proposées, en l'occurrence, la résolution LAPACK[36] qui utilise le raffinement itératif en précision étendue. Nous travaillons actuellement à la preuve formelle du comportement compensé ainsi qu'à l'obtention d'un algorithme de calcul certifié de l'erreur résiduelle. Ces développements constituent le centre de la thèse de Doctorat de N. Louvet qui a démarré en octobre 2004.

Par ailleurs, nous poursuivrons aussi grâce à CENA, l'identification d'algorithmes qui bénéficient de la correction des erreurs d'arrondi pour ensuite les spécialiser. Nous disposerons ainsi de versions optimisées, fiables et portables d'algorithmes compensés. Un autre domaine d'extension de ce type de méthode nécessite de disposer (d'estimations) des erreurs d'arrondi élémentaires pour les fonctions numériques type logarithme, exponentielle, ... Les résultats connus sur ce type de problème sont ceux obtenus sur le dilemme du fabriquant de tables par J.M. Muller et ses

[36] E. Anderson, Z. Bai, C. Bischof, S. Blackford, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, and D. Sorensen. *LAPACK Users' Guide*. Society for Industrial and Applied Mathematics, Philadelphia, PA, third edition, 1999.

étudiants. Nous proposons d'étudier un calcul économique d'estimateurs d'ordre 1 de ces erreurs élémentaires.

Certifier : outils et applications

Participants : Vincent Beaudenon, Marc Daumas (LIRMM), Stef Graillat, Philippe Langlois, Matthieu Martel (CEA).

Les outils sont en train d'arriver à maturité. Les travaux des membres du groupe s'organisent comme une concentration verticale dans deux dimensions.

Dans la dimension des doctorants, nous sommes à même de proposer des développements adapté pour des étudiants aux intérêts scientifiques très variés. Ceux-ci peuvent contribuer efficacement à nos travaux par des développements en génie logiciel, au niveau matériel, en analyse numérique, en calcul symbolique, en sémantique ou dans les méthodes formelles.

Dans la dimension des applications, nous aurons très prochainement à notre disposition des outils différents et complémentaires capables d'envisager efficacement des problèmes variés. De plus, ces outils seront bientôt capables de prendre en compte des applications complètes avec des efforts raisonnables. Dans la pratique, la certification ne sera bientôt plus réservée à des grands groupes tels que l'Aérospatiale ou la NASA avec qui les participants collaborent. Nous pourrons envisager de certifier des applications de petites entreprises. Cela est d'autant plus intéressant que les industriels sont de plus en plus demandeurs de ce type de technologie.

Ce document (23 pages) est complété par une Annexe (4 pages)

A Perpignan, le 23/07/07