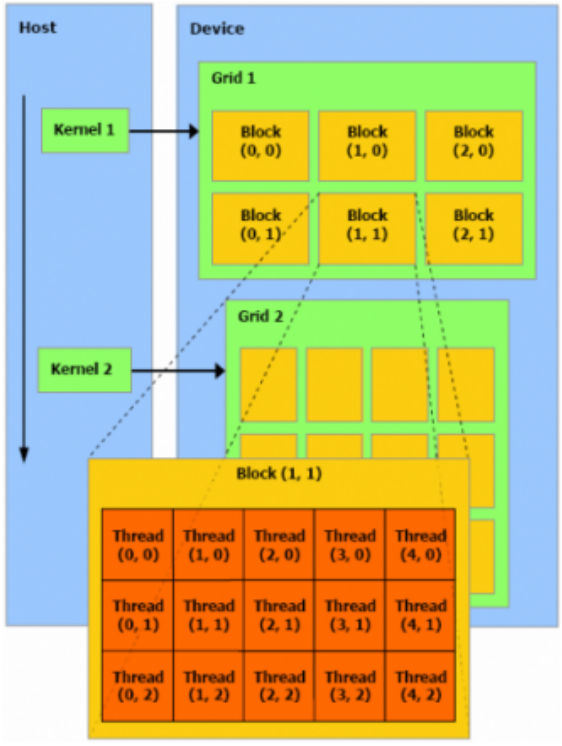
n°13
Septembre
2010

La lettre IN2P3 Informatique

Réseau des Informaticiens de l'IN2P3 et de l'IRFU

A LA UNE

Programmation GPU



Les cartes graphiques ont toujours fait l'objet de beaucoup d'investissements et d'améliorations, étant donné les fortes demandes au niveau des performances de ces composants et la concurrence qui existe entre les différents producteurs (notamment ATI/AMD et Nvidia). Les traitements graphiques sont massivement parallèles. C'est pourquoi les GPU sont adaptés au calcul parallèle. Ces processeurs graphiques sont en effet composés d'une grande quantité de processeurs qui permettent l'exécution d'instructions sur un grand nombre de données de manière simultanée, et ainsi de réduire considérablement les temps de traitement graphiques.

Les évolutions des processeurs graphiques ont contribué à l'amélioration des cartes graphiques au point de rendre leur puissance de calcul très intéressante pour des applications non liées au graphisme. En effet, la puissance de calcul brute offerte est plus élevée de par le grand nombre de processeurs présents sur ces cartes. C'est pourquoi il n'est pas rare d'obtenir de grands facteurs d'accélération entre CPU et GPU pour une même application. De plus, le coût d'une carte graphique face au coût d'un cluster de calcul rend ce mode de programmation très attractif.

Depuis quelques temps, Nvidia a fait un grand pas dans le High Performance Computing en (...)

lire la suite

Archives

Abonnement

Pour vous abonner/désabonner, suivez ce lien.

Proposer un article

Vous souhaitez proposer un article ? Envoyez un mail à LettreInformatique@in2p3.fr.

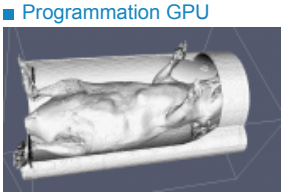
■ Nomination

Nouveau Directeur Adjoint au Centre de Calcul de l'IN2P3

Après presque 16 ans passés à l'IN2P3 – dont 10 ans au Centre de Calcul – Pierre-Etienne Macchi prend le poste de Directeur Adjoint du CC-IN2P3/CNRS

lire la suite

■ Programmation GPU

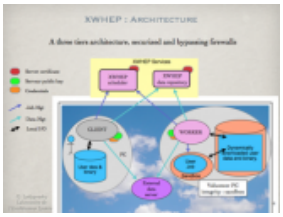


Reconstruction tomographique avec CUDA

Le groupe ImaBio de l'IPHC développe une plate-forme d'imagerie multimodale (AMISSA : A Multimodality Imaging System for Small Animal) qui combine trois techniques d'imagerie médicale : la micro tomodensitométrie X donne une représentation anatomique de l'organisme, tandis que les techniques d'imagerie nucléaire, micro tomographe à émission de positrons et micro tomographe à émission monophonique, fournissent une représentation dite fonctionnelle, reflétant la physiologie et le métabolisme des organes. Pour cet article, nous nous concentrerons uniquement sur le micro tomodensitomètre X.

lire la suite

■ XtremWeb-HEP



Une grille de PC interconnectée à EGEE

La grille de PC (Desktop Grid), apparue à la fin des années '90, a été rendue très populaire grâce au projet SETI@Home. Ce projet avait réussi à toucher le grand public qui a largement participé à l'aventure du calcul distribué en fournissant ses propres ressources aux applications scientifiques. Depuis d'autres projets ont émergé, tels que Boinc (qui a succédé à SETI), OurGrid, ou encore XtremWeb. Nous présentons ici les travaux menés autour de ce dernier afin de répondre au mieux aux attentes de la communauté scientifique.

lire la suite

■ Hommage

Décès d'Emmanuel Hornero

C'est avec une profonde tristesse que nous avons appris la disparition tragique de notre collègue et ami Emmanuel Hornero, à l'âge de 31 ans, le 1er août 2010 lors d'un accident de planeur près du Puy-en-Velay.

lire la suite

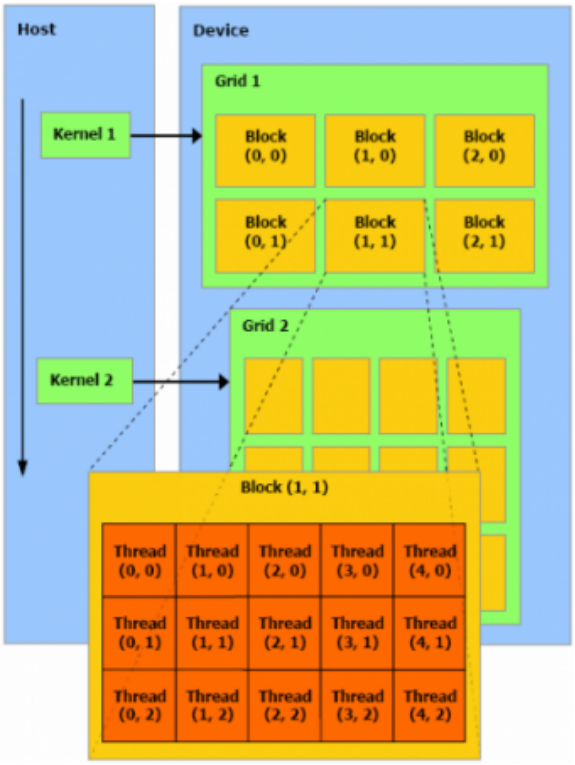
Equipe **Responsables éditoriaux** : Dominique Boutigny et Cristinel Diaconu
Comité de rédaction : Virginie Dutruel, Sébastien Grégoire, Eric Legay, Gaëlle Shifrin et Tiffany Thome

n°13
Septembre
2010

La lettre IN2P3 Informatique
Réseau des Informaticiens de l'IN2P3 et de l'IRFU

A LA UNE

Programmation GPU



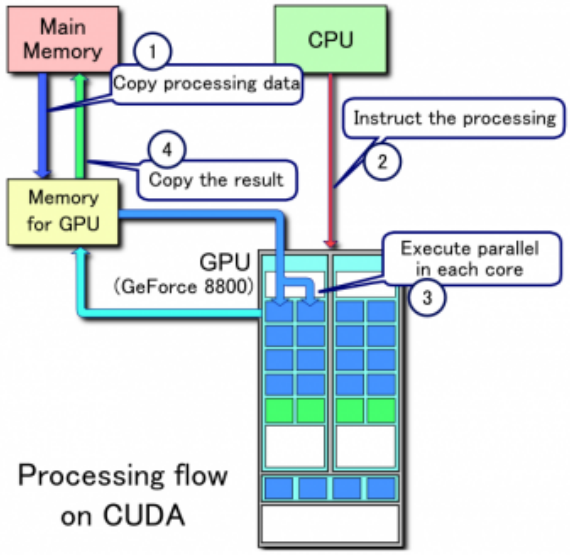
Les cartes graphiques ont toujours fait l'objet de beaucoup d'investissements et d'améliorations, étant donné les fortes demandes au niveau des performances de ces composants et la concurrence qui existe entre les différents producteurs (notamment ATI/AMD et Nvidia). Les traitements graphiques sont massivement parallèles. C'est pourquoi les GPU sont adaptés au calcul parallèle. Ces processeurs graphiques sont en effet composés d'une grande quantité de processeurs qui permettent l'exécution d'instructions sur un grand nombre de données de manière simultanée, et ainsi de réduire considérablement les temps de traitement graphiques.

Les évolutions des processeurs graphiques ont contribué à l'amélioration des cartes graphiques au point de rendre leur puissance de calcul très intéressante pour des applications non liées au graphisme. En effet, la puissance de calcul brute offerte est plus élevée de par le grand nombre de processeurs présents sur ces cartes. C'est pourquoi il n'est pas rare d'obtenir de grands facteurs d'accélération entre CPU et GPU pour une même application. De plus, le coût d'une carte graphique face au coût d'un cluster de calcul rend ce mode de programmation très attractif.

Depuis quelques temps, Nvidia a fait un grand pas dans le High Performance Computing en lançant CUDA (Compute Unified Device Architecture), une extension du langage C qui permet d'exploiter les capacités des cartes graphiques de la marque pour exécuter des instructions en parallèle.

Plus récemment, une collaboration entre les différents fournisseurs de CPU et de GPU a permis l'émergence du langage OpenCL (Open Computing Language). Ce langage, créé par le groupe Khronos (déjà à l'origine du standard OpenGL), est conçu de manière à pouvoir prendre en charge le parallélisme au niveau GPU mais aussi au niveau CPU.

Ensuite, les kernels disposent d'arguments spécifiques au découpage de la tâche totale à réaliser. Il s'agit là d'associer des identifiants aux tâches qui s'exécutent en parallèle afin d'assurer le lien entre les instructions à effectuer et les données à traiter. En CUDA, ces subdivisions s'appellent des grilles, en OpenCL, ce sont des NDRanges. Elles sont composées de : threads (CUDA) / work items (OpenCL) qui sont une instruction ou un groupe d'instructions qui correspond à l'instanciation d'un kernel. blocs (CUDA) / work groups (OpenCL) qui sont des groupes de threads / work items qui doivent être indépendants les uns des autres. Afin de se repérer, chaque bloc (respectivement work group) possède un identifiant unique 1D ou 2D au sein de la grille (respectivement NDRange) et chaque thread (respectivement work item) possède un identifiant unique, 1D, 2D ou 3D au sein du bloc (respectivement work group) auquel il appartient.



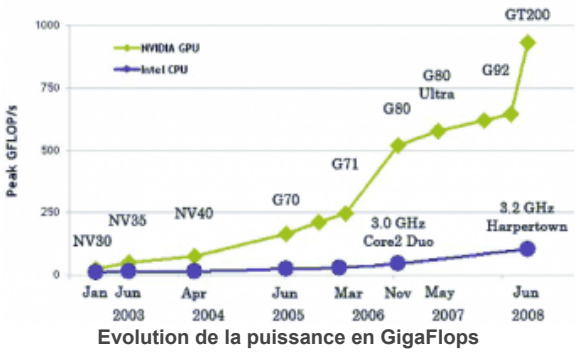
Enfin, une fois qu'un premier portage a été effectué, il n'est pas rare d'avoir à consacrer du temps à l'optimisation du programme, autant au niveau des kernels que de la gestion de la mémoire. Aussi, il est souvent nécessaire de spécialiser son application en fonction du matériel utilisé (modèle de la carte graphique, ...).

Tout ceci implique un effort de programmation supplémentaire à fournir lors du portage d'une application sur une architecture hybride CPU/GPU.

CUDA

Comme mentionné plus haut, cette bibliothèque de Nvidia est une extension au langage C pour la programmation GPU. La première version officielle est parue en février 2007. CUDA permet l'utilisation de deux niveaux d'API : une API bas niveau appelée CUDA driver API, une API plus haut niveau appelée CUDA runtime API ou CUDA C. L'API haut niveau permet une programmation plus aisée. L'initialisation, la gestion du contexte et des modules sont faites de manière implicite, ce qui raccourcit considérablement la longueur du code. Néanmoins, l'API bas niveau offre des fonctionnalités supplémentaires qui peuvent s'avérer intéressantes pour certains projets. Malheureusement, cet outil n'est disponible que pour les processeurs de la marque au caméléon, ce qui restreint considérablement la portabilité d'un code utilisant cette bibliothèque. Ceci est toutefois nuancé par le fait que Nvidia se partage l'essentiel du marché avec ATI/AMD. Cet inconvénient n'a pourtant pas arrêté les développeurs, qui ont fait apparaître toute une gamme d'outils

Ces deux langages facilitent le développement d'applications sur carte graphique et ont le même modèle de programmation. Néanmoins, le portage d'un code séquentiel sur une technologie GPU nécessite un temps supplémentaire de programmation non négligeable.



Evolution de la puissance en GigaFlops

La Programmation GPU

Tout d'abord, avant de porter une application sur une architecture hybride CPU/GPU, il faut déterminer les étapes les plus coûteuses d'un programme et déterminer si elles sont parallélisables. Il faut aussi garder à l'esprit que la programmation parallèle sur GPU suit un modèle SIMT (Single Instruction Multiple Thread).Ceci signifie que des groupes de threads doivent exécuter les mêmes instructions en même temps sur le GPU afin d'obtenir des gains de temps intéressants.

Lorsqu'on souhaite exécuter un morceau de code sur une carte graphique, il faut tout d'abord créer des kernels, qui ne sont autre que des morceaux de code parallèle exécutables sur le GPU, appelés par le CPU.

Les cartes graphiques disposent de leur propre mémoire RAM. Pour exécuter un kernel, il faut organiser de manière explicite les transferts de données nécessaire entre la carte mère et la carte graphique. De plus, la gestion de la mémoire est explicite sur une carte graphique : lors d'une opération de copie, il faut spécifier l'endroit où les données doivent être transférées (e.g. Mémoire globale, mémoire des textures, mémoire constante,...).

utilisant ces bibliothèques : CUBLAS pour l'algèbre linéaire, CuDPP pour les tris ou les réductions et d'autres encore. Ces outils s'avèrent être de puissants alliés lors du développement d'un code sur carte graphique. De plus, le support offert par la marque au caméléon est bien fourni de par les nombreux guides existants, et permet un apprentissage éclairé de l'utilisation de cette bibliothèque, sous réserve d'avoir de bonnes bases en C.
OpenCL Cette extension du C est issue d'un partenariat entre AMD, Apple, Intel et Nvidia, tous réunis au sein du Khronos Compute Working Group. Le but était de pouvoir programmer sur des systèmes parallèles hétérogènes visant des plateformes GPU mais aussi CPU-multicoeurs. Les premières versions d'OpenCL sont apparues au courant de l'année 2009, ce qui en fait un outil de programmation relativement jeune. Ici, il n'y a qu'une seule API disponible qui est équivalente à l'API bas niveau de CUDA (Driver API). La principale force d'OpenCL réside en sa portabilité. En effet, il est possible, avec ce langage, de programmer à la fois des CPU-multicoeurs mais aussi des GPU quelle qu'en soit la marque (AMD/ATI, Nvidia,...). Le deuxième point positif est le caractère public et open source de cet outil. Pour l'instant, ce langage ne dispose pas encore d'autant d'outil que son prédécesseur CUDA. Ceci le rend un peu moins populaire pour le moment au sein des développeurs. Il faut en effet développer ses propres outils à chaque besoin particulier (réduction, tri, multiplication matricielle, ...). Néanmoins, ces outils commencent à apparaître peu à peu.

Conclusion

Au niveau des performances, même si CUDA a parfois un léger avantage sur les GPU Nvidia, les deux outils se valent en terme de temps d'exécutions. Le plus gros point noir reste cependant la portabilité des performances. Afin d'avoir des gains de temps optimaux, il faut quasiment systématiquement programmer en fonction de son hardware qui a ses propres contraintes, aussi bien en CUDA qu'en OpenCL. Pour le futur de ces langages, ils continueront à perdurer tant que les GPU offriront une puissance de calcul brute à moindre coût face aux CPU. On peut néanmoins penser qu'OpenCL, grâce à sa portabilité, finira par dépasser CUDA en terme de nombre d'utilisateurs.

Matthieu Kuhn

n°13
Septembre
2010

La lettre IN2P3 Informatique
Réseau des Informaticiens de l'IN2P3 et de l'IRFU

Nouveau Directeur Adjoint au Centre de Calcul de l'IN2P3

Après presque 16 ans passés à l'IN2P3 – dont 10 ans au Centre de Calcul – Pierre-Etienne Macchi prend le poste de Directeur Adjoint du CC-IN2P3/CNRS

Diplômé en Informatique Industrielle et Instrumentation de l'école d'ingénieur de l'Université Joseph Fourier – Grenoble (38), Pierre-Etienne Macchi a rejoint l'IN2P3, il y a un peu plus de 15 ans, tout d'abord aux Services Centraux, en tant que Chef de Projet Logiciel, puis Chef du Service Informatique et Administration Générale. C'est en 2000 qu'il arrive au Centre de Calcul de l'IN2P3/CNRS (CC-IN2P3), tout d'abord au poste de Chef de Projet Bases de Données et Services Web, puis de Responsable du Service Systèmes d'Information et de Communication, avant de diriger l'équipe Développements.

Durant ces années au Centre de Calcul, il a pu, avec son équipe, mettre en place et opérer un certain nombre de technologies telles l'infrastructure d'hébergement web, les clusters de bases de données Oracle, les serveurs de bases de données Open Source (MySQL, PostgreSQL, OpenLDAP). Il a également participé à l'élaboration de la mise en œuvre d'outils à caractères collaboratifs pour la communauté : forge logicielle de l'IN2P3, outils de gestion des conférences (Indico), calendriers partagés…

En qualité de Chef d'équipe, il relevait directement du directeur du Centre de Calcul et tenait un rôle de conseil auprès de la Direction pour l'aider dans la réflexion et la mise en œuvre des différentes technologies liées à son domaine.

Pierre-Étienne Macchi va maintenant assurer la fonction de Directeur Adjoint du CC-IN2P3. Il remplace à ce poste Fabio Hernandez actuellement détaché à l'IHEP de Pékin *(voir **interview de Fabio Hernandez dans la lettre n°12 de juillet**)*.

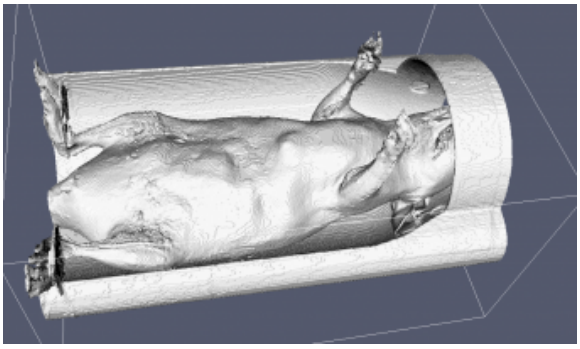
Ainsi, Pierre-Etienne Macchi a désormais pour missions de superviser l'ensemble des activités techniques et informatiques et de s'assurer de la cohérence des choix et de leur adéquation par rapport aux besoins et à la stratégie de l'unité. De plus, il est chargé de la coordination des interactions entre les différentes équipes, ainsi que du suivi des actions techniques décidées en Comité de Direction. Enfin, il doit assurer la planification des investissements techniques.

Souhaitons-lui une pleine réussite dans sa nouvelle fonction !

n°13
Septembre
2010

La lettre IN2P3 Informatique
Réseau des Informaticiens de l'IN2P3 et de l'IRFU

Reconstruction tomographique avec CUDA

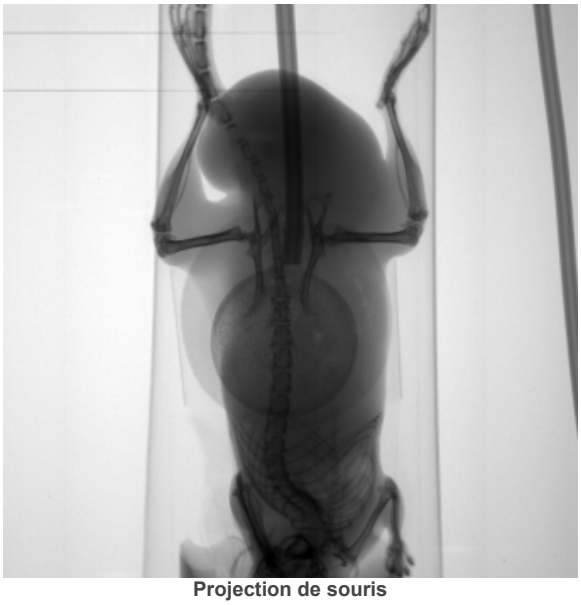


Le groupe ImaBio de l'IPHC développe une plate-forme d'imagerie multimodale (AMISSA : A Multimodality Imaging System for Small Animal) qui combine trois techniques d'imagerie médicale : la micro tomodensitométrie X donne une représentation anatomique de l'organisme, tandis que les techniques d'imagerie nucléaire, micro tomographe à émission de positrons et micro tomographe à émission monophonique, fournissent une représentation dite fonctionnelle, reflétant la physiologie et le métabolisme des organes. Pour cet article, nous nous concentrerons uniquement sur le micro tomodensitomètre X.

Notre micro tomodensitométre X a pour objectif d'imager un petit animal (rats, souris, petits oiseaux...). Il est composé d'un tube X et d'un détecteur en vis à vis qui tournent autour de l'animal. Typiquement, une acquisition recueille 768 projections d'une taille de 2048x2048 pixels où chaque valeur est codée sur des entiers courts : nous avons donc 6 Go de données en entrée. Notre but est de reconstruire le volume imagé à partir de ces projections dans un temps raisonnable, correspondant au temps nécessaire à l'acquisition des projections, soit 6 minutes environ. Le volume reconstruit est de l'ordre de 768*512*384 pixels codés sur des nombres décimaux en simple précision soit 540 Mo.

Le banc AMISSA

Il existe deux grandes classes d'algorithmes de reconstruction qui permettent de reconstruire un volume 3D à partir de ses projections : les algorithmes de type analytique et les algorithmes de type itératif. Les algorithmes analytiques consistent à rétro-projeter les valeurs mesurées, préalablement filtrées, dans un volume contenant l'objet à reconstruire, puis à un cluster de douze PC, permettant la reconstruction en quasi temps réel. Un gain de 20 a été observé sur le temps de calcul permettant de passer d'un temps de reconstruction de 2 heures à 6 minutes. L'analyse des 6 Go de données des projections issues du micro tomodensitomètre X et la reconstruction d'un volume 3D de 500 Mo s'exécutaient alors pendant un temps équivalent au temps nécessaire à un examen de type CT. Le processus de reconstruction pouvant démarrer dès la première projection acquise, l'image 3D était disponible dès la fin de l'examen. Cependant l'achat et la maintenance d'un tel cluster représente un coût humain et financier non négligeable : pour information, le coût lié à l'achat du cluster en 2004 était de l'ordre de 20 keuros. Nous avons donc décidé en 2008, d'explorer une nouvelle voie qui consiste à utiliser un processeur graphique comme coprocesseur de calcul généraliste.



Projection de souris

Les premières implémentations de calculs généralistes sur cartes graphiques datent d'une dizaine d'années. Le langage OpenGL (shaders) était alors utilisé : il était nécessaire d'adapter l'algorithme à implanter à un problème purement graphique. Dans le cadre de la tomographie, les premiers résultats obtenus étaient très encourageants ; cependant, la programmation de calculs généralistes sur cartes graphiques restait difficile à mettre en œuvre. L'apparition de CUDA et openCL ont permis une démocratisation de l'utilisation de processeurs graphiques pour le calcul parallèle. Depuis deux ans, bon nombre d'algorithmes de reconstruction tomographique ont été portés sur cartes graphiques et leur mise en œuvre sur des équipements commercialisés a fait son apparition.

Les premiers résultats, obtenus rapidement, ont été très encourageants. Nous avons rapidement atteint des performances en temps de calcul comparables voire meilleures que ce qui avait été observé avec le cluster de CPU avec une simple carte de jeux vidéo (Nvidia GeForce 8800GT). Ces premiers résultats ont été obtenus avec un PC, dédié pour le jeu, acheté à un prix de 1,8 keuros. Un algorithme de type itératif a également été porté sur GPU, en vue d'ajouter à la reconstruction des traitements de correction de phénomènes physiques tels que le durcissement de faisceau ou la diffusion, phénomènes qui entachent d'artéfacts les volumes reconstruits.

L'utilisation de cartes graphiques pour la reconstruction tomographique s'est avérée être un très bon choix : elle nous a

sommer les valeurs de ces rétro-projections pour les différents angles d'acquisition. La deuxième classe d'algorithmes concerne les algorithmes de type itératif. Ils consistent, de façon itérative, à réaliser une estimation du volume à reconstruire, à projeter ce volume et à comparer projections estimées et projections mesurées. Une mesure de l'erreur et une correction de l'estimation du volume est alors effectuée. Du fait du très grand nombre de données en entrée à traiter, les algorithmes utilisés sont très gourmands en temps de calcul. L'algorithme de Feldkamp, très utilisé dans le cas d'une reconstruction de type analytique nécessite un temps de calcul de plus d'une heure, si l'on considère nos volumes de données à traiter et une exécution sur un unique thread. Cependant, les algorithmes de reconstruction sont hautement parallélisables, ce qui nous a permis de réduire significativement le temps de calcul.

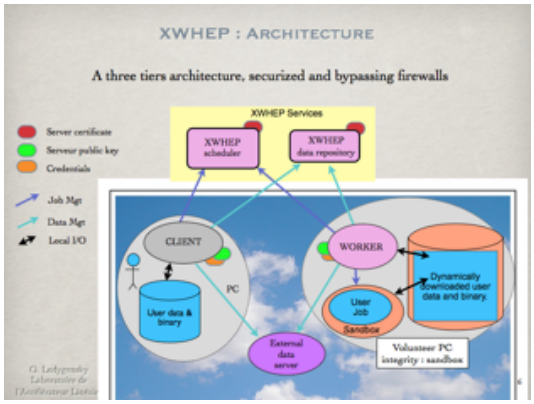
Il y a six ans, un gros effort d'optimisation logiciel a été effectué pour porter un code de reconstruction analytique (Feldkamp) sur permis de réduire considérablement nos frais d'infrastructure et d'obtenir de meilleurs résultats en terme de temps de calcul par rapport à une solution plus conventionnelle mettant en oeuvre un cluster de PCs. Si le portage d'un algorithme sur carte graphique est facilité par les langages de type CUDA ou openCL, l'optimisation du code pour une architecture de type processeur graphique reste un travail nécessaire si l'on souhaite tirer partie au mieux de la puissance de calcul du matériel. Cette phase d'optimisation nécessite du temps et une certaine expérience. La gestion des différentes mémoires présentes sur une carte graphique est un point crucial en ce qui concerne les performances d'un algorithme porté sur GPU. Cette gestion dépend à la fois du hardware utilisé et de l'algorithme mis en œuvre.

Damien Vintache

n°13
Septembre
2010

La lettre IN2P3 Informatique
Réseau des Informaticiens de l'IN2P3 et de l'IRFU

Une grille de PC interconnectée à EGEE



La grille de PC (Desktop Grid), apparue à la fin des années '90, a été rendue très populaire grâce au projet SETI@Home. Ce projet avait réussi à toucher le grand public qui a largement participé à l'aventure du calcul distribué en fournissant ses propres ressources aux applications scientifiques. Depuis d'autres projets ont émergé, tels que Boinc (qui a succédé à SETI), OurGrid, ou encore XtremWeb. Nous présentons ici les travaux menés autour de ce dernier afin de répondre au mieux aux attentes de la communauté scientifique.

Les grilles de PC (DG) permettent à tout un chacun de participer à la construction d'une ressource de calcul distribuée en proposant ses propres ordinateurs individuels sur la base du volontariat ; mais aussi d'utiliser la grille ainsi obtenue en soumettant ses propres calculs.

Le Laboratoire de l'Accélérateur Linéaire (LAL) s'est impliqué très tôt dans les projets de grilles informatiques. Il s'est intéressé aux grilles de PC à partir des années 2000 et a démontré les avantages que la physique des hautes énergies peut tirer des DG, malgré ses contraintes fortes (taille des données, temps de calcul). Après ces travaux, le LAL a continué sur cette voie et développé l'intergiciel XtremWeb-HEP (XWHEP), basé sur XtremWeb. Avec cette technologie, les objectifs fixés ont été atteints : stabilisation et pérennisation de l'intergiciel ; outils d'installation et d'administration ; sécurisation et connexion inter-grilles.

Nous détaillons ci-après la sécurisation.

La technologie XtremWeb ayant donné satisfaction, nous avons gardé dans XWHEP ses caractéristiques principales : architecture trois tiers ; utilisation des PC sur le mode « vol de cycle » ; support de l'hétérogénéité (Mac OS X, Linux et Windows) ; tolérance aux pannes ; passage des pare-feux. XWHEP étend la gestion des utilisateurs de XtremWeb en introduisant d'une part une meilleure gestion de l'authentification grâce à l'utilisation des certificats X509, et en proposant une gestion des droits d'accès. XWHEP implémente, d'autre part, la gestion des données qui était absente de XtremWeb. La figure 1 montre l'architecture obtenue. Un déploiement XWHEP est composé d'un ou plusieurs services publics : l'ordonnanceur, garant de la cohérence de la grille, et le « data repository », responsable de l'intégrité des données. Un déploiement est aussi composé de « workers » qui sont installés sur les PC volontaires et qui ont pour charge d'exécuter les jobs à calculer. Enfin, le client XWHEP est la partie émergée de l'iceberg : l'interface utilisateur permettant de gérer la grille, soumettre des jobs et des données, récupérer les résultats etc.

La sécurité est assurée à différents niveaux :

- communications : les clés électroniques des serveurs permettent leur authentification ainsi que le cryptage des communications ;
- ressources volontaires : le bac à sable permet d'assurer l'intégrité de la ressource ;
- authentification : les certificats X509 permettent d'authentifier les utilisateurs ;
- autorisation : la gestion des autorisations est gérée par l'ordonnanceur. Dans XWHEP, tout objet est défini avec un droit d'accès autorisant ou interdisant la lecture, l'écriture ou l'exécution au propriétaire, aux membres du groupe auquel appartient le propriétaire, et aux autres utilisateurs de la grille. Ces droits d'accès sont, dans leur définition, semblables à ceux que l'on peut trouver dans les systèmes de fichier Linux ;
- confinement : il est possible de confiner le déploiement en utilisant la gestion de groupe utilisateurs ; l'intergiciel assurant alors que les jobs, les binaires et les données ne seront distribués qu'au sein d'un même groupe.

Notre déploiement au LAL est aujourd'hui utilisé pour des applications qui vont de la physique à la recherche médicale. [*1*]

A titre d'exemple, entre juin 2009 et février 2010, plus de cinq cent mille jobs ont été exécutés.

Nous invitons toute personne intéressée par cette technologie à nous contacter, et nous pouvons proposer trois modes d'utilisation (parfaitement compatibles entre eux) :

- installation du client, afin d'utiliser les ressources existantes pour du calcul scientifique ;
- installation de workers afin d'accroître les ressources de calcul disponibles pour le calcul scientifique ;
- installation d'une grille de PC complète au sein de votre infrastructure.

Oleg Lodygensky, Etienne Urbah, Simon Dadoun - LAL IN2P3 / Gilles Fedak - LIP INRIA

[*1*] ***http://www.xtremweb-hep.org/lal/Xtr...***

n°13
Septembre
2010

La lettre IN2P3 Informatique
Réseau des Informaticiens de l'IN2P3 et de l'IRFU

Décès d'Emmanuel Hornero

C'est avec une profonde tristesse que nous avons appris la disparition tragique de notre collègue et ami Emmanuel Hornero, à l'âge de 31 ans, le 1er août 2010 lors d'un accident de planeur près du Puy-en-Velay.

Emmanuel avait commencé sa carrière au LPNHE en août 2004 comme CDD ingénieur expert en développement d'applications avant d'être titularisé ingénieur d'étude 2ème classe en décembre 2005. Avant cela il avait travaillé dans la société CIPAM à Clermont-Ferrand de 1998 à 2000 en tant qu'ingénieur concepteur de systèmes d'informations informatisés.

Emmanuel était diplômé de l'IUT de Clermont-Ferrand en 1998 (option Systèmes industriels), de l'Institut d'Ingénierie Informatique de Limoges en 2000 (diplôme d'ingénieur spécialité "Conception de Systèmes d'Informations Informatisés"), de l'Université de Bordeaux en 2001 (DESS en Imagerie), et de l'Université Paul Sabatier à Toulouse (ENSEEIHT) en 2004 (DEA Informatique de l'Image et du Langage).

Emmanuel était très fortement impliqué dans le développement d'expériences au sein de collaborations internationale (SUPERNOVA, ATLAS, HESS, SNDICE, ILC et LSST) et travaillait aussi régulièrement pour le service informatique du LPNHE. Sa collaboration avec le groupe SUPERNOVA avait commencé dès son arrivée au laboratoire lors d'un contrat à long terme. Dans le cadre du projet SNFactory, il avait contribué aux développements liés à la prise de données du spectrographe du télescope de Hawaii, au système de processing central implanté au Centre de Calcul de l'IN2P3 à Lyon et avait automatisé la mise à jour de la base de données à Lyon en fonction de l'état du déroulement de la chaîne de traitement des données (acquisition, transfert et traitement de données). A la demande du groupe SNLS, il avait appliqué la même démarche à une base de données hébergée sur une machine locale au LPNHE.

acquisitions de données de la nouvelle caméra HESS2. Dans la même année, Emmanuel avait proposé une interface de pilotage automatique du guidage et du positionnement d'un miroir dans le cadre d'un projet d'étude pour SNDICE.

En 2009, dans le cadre de la grille de Calcul Tiers2 pour ATLAS en production dans le laboratoire, Emmanuel avait été convié à une mission délicate consistant à apporter à la fois une expertise développement et un soutien aux physiciens désirant exploiter au mieux leurs logiciels de calcul et les ressources en stockage sur la grille locale du LPNHE. Dans la même période, Emmanuel avait participé aux tests et à la caractérisation d'une nouvelle carte d'acquisition prototype pour le groupe ILC. Il avait développé pour celui-ci une interface de dialogue avec la carte afin de lancer des séries de tests.

Au sein du service informatique, Emmanuel avait su s'adapter aux différentes missions qui lui avaient été confiées dans des contextes variés sur des projets techniques différents ; configuration d'un nouveau serveur d'impression, participation à la virtualisation des serveurs d'intérêt général, étude des performances de calcul des algorithmes scientifiques sur cartes graphiques, divers développements des portails WEB (doctorants, stagiaires) et développement d'une application graphique de simulation probabiliste et réelle sur la propagation de gerbes dans une cuve de Cherenkov dans le cadre de l'event display du groupe AUGER. Ce dernier travail avait fait l'objet d'une présentation aux Journées Informatique de l'IN2P3 à Lyon en septembre 2006. Dernièrement, Emmanuel avait initié des tests en laboratoire des librairies CUDA sur une carte graphique ce qui lui avait donné l'occasion de participer à des partages de compétences interlabos dans les domaines du calcul scientifique et de l'imagerie.

Emmanuel était unanimement apprécié par ses collègues. Tous louaient sa grande gentillesse et sa disponibilité. Doté d'un caractère égal, il était toujours de bonne humeur, souriant et prêt à rendre service.

Emmanuel était un homme passionné. Captivé depuis l'enfance par l'aviation il pratiquait le vol à voile depuis de nombreuses années. Il était même en passe de devenir instructeur. Il y a deux ans, il avait découvert le théâtre et s'y investissait depuis sans réserve. Ayant obtenu une inscription dans une école de théâtre, il devait passer à mi-temps à la rentrée pour pouvoir en suivre les cours. L'informatique était aussi l'une de ses passions et il était heureux d'avoir pu en faire sa vie professionnelle. Il s'était également engagé lors du mouvement de défense de la recherche en 2009.

Ces différents aspects de sa vie n'étaient pas cloisonnés mais communiquaient. Il n'hésitait pas par exemple à mettre ses compétences informatiques au service de ses autres activités. Il travaillait à la mise au point d'un logiciel de reconstruction de ses vols en planeur à partir des informations de son GPS, il avait mis sur pied et administrait le serveur web de sa compagnie de théâtre (Les productions de la fabrique), etc... Il partageait volontiers ces différentes passions avec ses collègues et savait leur communiquer son enthousiasme et sa fraîcheur.

Au sein de l'équipe informatique, et plus largement au sein du LPNHE, il occupait une place importante, tant sur le plan purement professionnel que sur le plan humain. Toujours prêt à participer avec enthousiasme à la vie du laboratoire, Emmanuel était en outre doté d'un humour subtil et n'hésitait pas à plaisanter et à partager sa bonne humeur avec ses collègues. Il participait ainsi grandement à la bonne ambiance au sein du service et au delà, au sein du laboratoire.

En 2008, Emmanuel avait intégré le groupe LSST dans le cadre d'un développement spécifique sous LabView pour les bancs de tests des nouveaux CCD et l'optimisation de la transformation et de la conversion des images brutes sauvegardées localement et répliquées au Centre de Calcul de Lyon. Parallèlement, Emmanuel avait contribué à l'étude du système de pilotage existant de la caméra HESS pour proposer un nouveau software permettant aux physiciens de tester, de paramétrer et lancer les

L'ensemble du personnel du LPNHE est sincèrement et profondément affecté par le départ prématuré d'Emmanuel. Il nous manquera énormément.

Reynald Pain