

# Rapport de stage

Chadi AKEL

31 août 2012

# Sommaire

<b>I</b>	<b>Remerciements</b>	<b>4</b>
<b>II</b>	<b>Introduction</b>	<b>5</b>
<b>III</b>	<b>L'entreprise et son secteur d'activité</b>	<b>6</b>
<b>1</b>	<b>L'entreprise</b>	<b>6</b>
1.1	Historique . . . . .	6
1.1.1	1919-1931 : les origines . . . . .	6
1.1.2	1931-1962 : la construction d'une grande entreprise française . . . . .	7
1.1.3	1962-1982 : Alliances internationale . . . . .	10
1.1.4	1982-1990 : Dimension mondiale . . . . .	12
1.1.5	1991-1997 : Actionnariat, privatisation, intense adaptation de l'offre et de l'organisation . . .	13
1.2	Aujourd'hui . . . . .	15
1.2.1	Cap sur l'innovation . . . . .	15
1.3	Organisations . . . . .	17
1.3.1	Security Solutions . . . . .	17
1.3.2	Business Integration Solutions . . . . .	17
1.3.3	Computing Solutions . . . . .	17
1.3.4	Innovative products . . . . .	17
<b>2</b>	<b>Le secteur d'activité</b>	<b>17</b>
2.1	Présentation . . . . .	17
2.1.1	Secteur public . . . . .	17
2.1.2	Finance . . . . .	17
2.1.3	Défense . . . . .	17
2.1.4	Télécom . . . . .	17
2.2	Actualités . . . . .	17
2.2.1	Cloud Computing . . . . .	17
<b>IV</b>	<b>Le cadre du stage</b>	<b>18</b>
<b>3</b>	<b>L'équipe</b>	<b>18</b>
<b>4</b>	<b>Fonctionnement</b>	<b>18</b>
<b>V</b>	<b>Travaux effectués et apports du stage</b>	<b>19</b>

<b>5</b>	<b>Travaux effectués</b>	<b>19</b>
5.1	Les outils . . . . .	19
5.2	Les missions . . . . .	19
5.2.1	Knights Ferry . . . . .	19
5.2.2	OpenGPU . . . . .	30
5.2.3	Knights Corner . . . . .	31
5.3	Les tâches périphériques . . . . .	32
5.3.1	Wiki . . . . .	32
<b>6</b>	<b>Apports du stage</b>	<b>32</b>
6.1	Compétences acquises . . . . .	32
6.1.1	Autonomie . . . . .	32
6.1.2	Travail en équipe . . . . .	32
6.1.3	Organisation . . . . .	32
6.1.4	Compétences techniques . . . . .	33
6.2	Difficultés rencontrées et solutions apportées . . . . .	33
6.3	La vie en société . . . . .	33
6.3.1	L’articulation des différents départements . . . . .	33
6.3.2	Les relations humaines entre les employés . . . . .	34
<b>VI</b>	<b>Conclusion</b>	<b>35</b>

## Première partie

# Remerciements

Avant tout développement sur cette expérience professionnelle, il apparaît opportun de commencer ce rapport de stage par des remerciements, à ceux qui m'ont beaucoup appris au cours de ce stage et même à ceux qui ont eu la gentillesse de faire de ce stage un moment très profitable. Aussi, je remercie tous les membres de l'équipe, et plus particulièrement Gaetan Bayle et Jean-François Lemerre, qui m'ont formé et accompagné tout au long de cette expérience professionnelle avec beaucoup de patience et de pédagogie.

## Deuxième partie

# Introduction

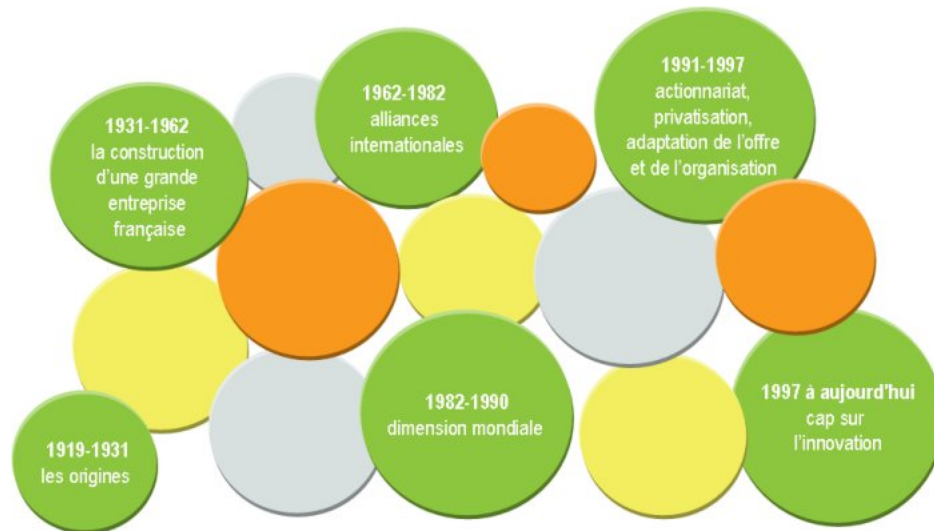
Du 1er avril au 30 septembre 2012, j'ai effectué un stage au sein de l'entreprise Bull située au Clayes-sous-bois dans les Yvelines. Bull étant un acteur important dans le domaine du calcul haute performance, j'ai naturellement été très enthousiaste à l'idée d'y faire mon stage. Le poste proposé permettait d'étudier les performances de l'accélérateur d'Intel (MIC) et d'autres GPUs de Nvidia ou AMD. Il était aussi question de travailler sur le projet OpenGPU en partenariat avec l'école centrale de Paris. Ce stage correspondait donc parfaitement à ma formation et l'étude des performances est un domaine que j'apprécie et sur lequel j'ai aussi eu l'occasion de travailler lors de mon précédent stage au laboratoire Exascale. [Ici présenter de manière globale l'entreprise qui sera plus détaillée dans la partie 1] Au cours de ce stage au département R&D dans l'équipe performance, j'ai pu m'intéresser aux accélérateurs tel que les GPU... [parler rapidement de la mission de mon stage qui sera plus détaillée dans la partie 2]

## Troisième partie

## L'entreprise et son secteur d'activité

## 1 L'entreprise

## 1.1 Historique



## 1.1.1 1919-1931 : les origines

**1919** : L'ingénieur norvégien Fredrik Rosing Bull a pour défi de concevoir une machine d'automatisation du traitement de statistiques pour la compagnie d'assurance Storebrand, qui est son employeur. En août **1921**, le prototype est présenté au conseil d'administration de Storebrand qui l'adopte. La presse spécialisée fait une bonne publicité à la machine de F.R. Bull. Une demi-douzaine d'exemplaires est livrée à diverses entreprises entre **1922** et **1925**. Ce succès est dû à la fois aux qualités techniques de la machine (notamment à sa simplicité) et au fait que son apparition met fin à l'emprise du système Hollerith (IBM), faisant ainsi baisser les prix et donnant le choix aux clients.

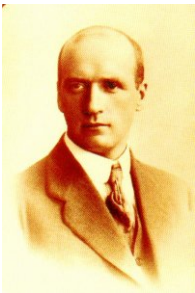


FIGURE 1 – Fredrik Rosing Bull 1882-1925

A la mort de F.R. en **1925**, l'initiateur principal de l'expansion européenne de Bull est Émile Genon, un belge qui vendait des machines à calculer. Il achète en **1927** les droits relatifs aux brevets de F.R. Bull pour dix pays d'Europe. Il entraîne la société H.W. Egli, établie en Suisse, à acquérir en **1928** les droits industriels touchant ces brevets hors des pays scandinaves et les machines sont fabriquées en Suisse dès **1928**.

En décembre **1929** la **première machine** fabriquée à Zürich est livrée aux laboratoires Sandoz. La recherche d'un marché national fort, capable d'absorber une production de type industriel, ainsi que la législation protégeant les brevets l'a conduit à s'implanter en France.

**1931** : En mars 1931 à Paris, la société H.W. Egli Bull, de droit français mais à majorité suisse, est fondée par trois partenaires : la société suisse H.W. Egli, la société Bull AG fondée l'année précédente à Zürich par Genon et enfin l'ATEIC (Association Technique d'Études industrielles et Comptables). Elle **vendra les machines Bull en France**. Elle s'installe avenue Gambetta dans le 20ème arrondissement de Paris ; le site est aujourd'hui occupé par le rectorat de l'académie de Paris.

### 1.1.2 1931-1962 : la construction d'une grande entreprise française

**1931** : Les études de Bull, sous la direction de Knut Andréas Knutsen, sortent la tabulatrice T30, 25 à 40% moins chère que la concurrence et capable d'imprimer 120 lignes à la minute. Un record mondial, **qui ne sera égalé que dix-huit ans plus tard** ! Cette performance est à l'origine du développement commercial de Bull.



FIGURE 2 – Tabulatrice T30

**1933** : Création de la **CMB** (Compagnie des Machines Bull). Née des brevets de machines mécanographiques de Frederik Rosing Bull et Knut Andréas Knutsen, elle a succédé à la société franco-suisse Egli-Bull. Deux ans plus tard (1935), elle détient 16% du marché français, devient le principal concurrent d'IBM en France et est commercialisée en Belgique, Suisse, Italie, Argentine et les pays Scandinaves. Sa tabulatrice est une des meilleures du marché et la plus rapide. Technologiquement, la CMB ne cesse d'innover. Elle jouit d'une excellente réputation de rapidité et de richesse fonctionnelle, et est bien adaptée aux applications européennes. L'entreprise construit environ trois équipements par mois et augmente régulièrement sa capacité de production.

**1935** : L'innovation exigeant des investissements coûteux, des pourparlers sont entrepris avec les pouvoirs publics en vue d'obtenir une aide pour le développement des études. E. Genon est mandaté par le Conseil d'Administration afin de poursuivre auprès de différentes firmes aux États-Unis les recherches d'accords de licence et de distribution. Il rencontre IBM qui lui fait une offre de collaboration amicale. Mais la compagnie préfère demander l'engagement du gouvernement français, dont la décision tarde à venir. Genon, sans avoir reçu l'autorisation du conseil d'administration, vend alors à IBM la majorité des actions de Bull A.G. (la société de commercialisation des machines Bull, qu'il dirigeait). Il y voit un moyen d'obtenir « une paix tacite » des brevets entre IBM et Bull et de développer Bull sur le plan international avec l'appui d'un groupe américain. L'intraitable Georges Vieillard, alors Directeur

de la CMB somme Genon de choisir : Bull ou IBM. **Après dix ans d'une intense activité souvent décisive**, Genon quitte Bull. De nouveaux acteurs entrent en scène : la famille Callies-Aussedat.

La Société des Papeteries Aussedat fournissait Bull en cartes mécanographiques utilisées par les machines. Depuis 1932, elle avait réalisé d'importants investissements dans ce domaine et était représentée au Conseil d'Administration de Bull par Jacques Callies. La menace d'une absorption de la compagnie par IBM inquiétait Aussedat car IBM exigeait de ses clients qu'ils lui achètent les cartes en exclusivité.

De même qu'il fallait éviter le rachat de Citroën par General Motors, il ne fallait pas que la Compagnie des Machines Bull tombe entre les mains des américains. Et, puisque l'État ne réagit toujours pas, la famille Callies décide d'accroître son engagement financier dans l'entreprise. Elle en prend la direction en la personne de Jacques Callies, ancien officier, nommé administrateur délégué de Bull en décembre 1935, puis Président-directeur Général. Il remplira cette fonction jusqu'à sa mort en novembre 1948 et aura comme successeur son frère Joseph, ingénieur aux papeteries Aussedat puis à la CMB. L'équipe qui animera et assurera pendant près de 30 ans l'expansion de la compagnie est désormais en place. (les Callies possédant 55% du capital).

**1936** : effectifs de 200 personnes.

Avec la mise au point, en **1938**, de la technique des cycles indépendants, les calculatrices électromécaniques Bull se sont améliorées et imposées peu à peu ; cette technique permettait de lancer de façon souple et optimisées des logiques de traitement spécifiques via le tableau de connexion. Les diverses machines permettent de lire, trier, comptabiliser et d'imprimer des milliers de données inscrites sur des cartes perforées. L'usage de ces cartes se poursuivra, avec l'électronique, jusqu'au début des années 80. La saisie des données sur cartes était à l'origine d'un nouveau métier disparu depuis, celui d'opérateur de perforation (le plus souvent opératrice d'ailleurs).

**1939** :



FIGURE 3 – Tabulatrice BS120

conception de la tabulatrice BS120 à cycles indépendants qui sera l'un des facteurs principaux de l'expansion de Bull pendant vingt ans.

partir de **1947**, l'activité exportatrice, interrompue par la guerre, reprend vigoureusement. Pendant les quinze années suivantes, le réseau international de Bull, va prendre une extension considérable et constituer une des grandes forces de la compagnie. Ainsi, en Belgique, **SOMECA, qui représentait Bull AG en 1930 devient en 1942**, la Société belge des machines Bull. En Suisse, la société Endrich A.G. partenaire de Bull depuis 1930, devient en 1947 une filiale sous le nom de Bull Lochkartenmaschinen A.G. En 1949, se conclut une association avec Olivetti pour créer une filiale de distribution en Italie : la société Olivetti-Bull. Dans les années quarante, Bull est implanté en Hollande, en Allemagne et en Amérique du Sud.

**1948** : Bull dépasse IBM sur le marché français avec 385 équipements installés. En seize ans, le nombre d'équipements installés sera multiplié par dix. Il s'agit d'une croissance essentiellement interne, due au développement des produits et des ventes. S'y ajoute l'absorption de certains sous-traitants de la compagnie. Cette période est à la fois celle où le marché de la mécanographie atteint son apogée, et celle pendant laquelle Bull, de même que ses concurrents, se convertit progressivement à l'électronique.



**1951 : début de l'aventure informatique encore plus audacieuse.** Bull présente,

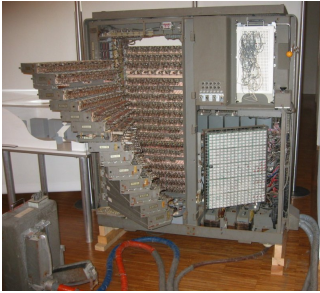


FIGURE 4 – Gamma 3

à Paris, au 3ème SICOB (Salon des Industries et du Commerce de Bureau) le Gamma 3, son premier calculateur électronique, relié aux machines de lecture et d'impression, il permet d'effectuer les calculs beaucoup plus rapidement. Succès technique et commercial, le Gamma 3 devient un modèle de référence et annonce le déclin de la mécanographie et le début de l'ère de l'informatique. Son programme, un simple enchaînement d'opérations, est câblé manuellement à l'aide de petites fiches électriques que l'on insère en séquence dans des trous numérotés donnant accès aux fonctions élémentaires de la machine. Il sera suivi par le Gamma 3 B à « tambour magnétique » utilisé aussi bien pour la gestion que pour le calcul scientifique ; le tambour contient des programmes et des données intermédiaires. C'est une unité de calcul rapide qui effectue des opérations comptables et scientifiques pour le compte d'une machine à cartes perforées (tabulatrice BS 120) à laquelle il est connecté. Effectuant **5 800 opérations par seconde**, il exécute les calculs

dans l'intervalle de lecture de 2 cartes consécutives, laissant la machine fonctionner à sa vitesse nominale de 150 cartes minutes. L'innovation technique réside dans l'utilisation intensive de diodes au cristal de germanium (**une première technologique**) : le Gamma 3 ne comporte plus que 400 tubes électroniques au lieu de 1 500 dans les appareils similaires, ce qui entraîne une réduction des coûts de fabrication et une fiabilité fortement améliorée.

**1952** : l'entreprise compte 2200 salariés. En **1956** le marché soviétique s'ouvre aux produits de la CMB. En 1960, Bull entre sur le marché de la République Populaire de Chine.

**En 1957** Bull développe le système de codage (**Caractères Magnétiques Codés à 7 bâtonnets**) encore utilisé aujourd'hui dans le traitement des chèques bancaires notamment.

**1960** : Bull lance « un grand frère » du Gamma 3 avec le Gamma 60, **premier ordinateur multitâche au monde**, doté d'une structure logique en avance de dix ans (20 fois plus rapide que le Gamma 3 et beaucoup plus puissant de par sa capacité multitâches). Les données sont initialement introduites à partir de cartes perforées à 80 colonnes, à la vitesse de 300 cartes à la minute ; elles sont alors stockées sur rubans et tambours magnétiques. Son développement conduit Bull à concevoir ses premiers éléments d'OS (Operating System). Il symbolise l'apparition du monde des informaticiens avec ses grandes salles climatisées et ses nouveaux métiers : les programmeurs, les analystes... Il préfigure les grands systèmes qui s'imposeront pendant les trente années suivantes et peut être considéré comme l'ancêtre du TERA 100. Une quinzaine de Gamma 60 seront vendus à des clients prestigieux parmi lesquels on peut citer : SNCF, EDF, Mitsubishi Corp. et le CEA.



FIGURE 5 – Gamma 60

Parallèlement Bull commercialisera le Gamma 10 conçu par Bull et véritable successeur du Gamma 3 et le Gamma 30 vendu sous licence RCA comme le feront de leur côté Siemens et ICL. Le dernier Gamma 60 européen sera retiré du service en 1974.

**1962** : lancement du Gamma 10, véritable successeur du Gamma 3 et dernier représentant de la génération « mécanographie à carte perforée » ; destiné aux applications de gestion, il utilise la technologie des grands systèmes (ordinateurs de 2ème génération). En octobre **1962** est signé un accord commercial avec Mitsubishi qui reçoit l'exclusivité de la vente du matériel Bull sur le marché japonais et acquiert un Gamma 60 puis des Gamma 10.

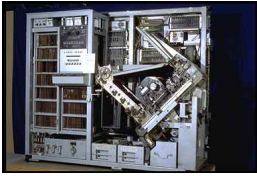


FIGURE 6 – Gamma 10

### 1.1.3 1962-1982 : Alliances internationale

**1962** : La CMB atteint le 17ème rang des sociétés françaises cotées en bourse. Elle doit cette performance à deux atouts majeurs : une gamme de produits adaptée aux besoins de la clientèle (dans les domaines de la comptabilité-gestion et du calcul scientifique) ; et la mise en place de services efficaces de formation, d'assistance technique et d'entretien. Ces prestations étaient le plus souvent intégrées dans le prix de location des machines, lesquelles n'étaient que très rarement vendues.

**1963** : fin 1963, la compagnie est présente dans 46 pays et exporte 60% de sa production. Elle occupe le deuxième rang mondial et le premier rang européen des industriels du traitement de l'information. **Bull détient 1/3 du marché français, 10% du marché européen.**

**1964** : La Compagnie des Machines Bull compte 15 600 salariés et réalise 60% de son chiffre d'affaires à l'exportation au travers d'un réseau de 22 filiales et agences servant 46 pays. C'est le **premier constructeur européen et le deuxième mondial**. Ce développement rapide nécessite des investissements importants tant pour le financement des locations que pour celui des études de nouveaux produits et technologies : passage à l'électronique, logiciels, élargissement de la gamme, etc...

En juillet 1964, le gouvernement autorise la prise de contrôle par le groupe américain General Electric, et la compagnie prend le nom de Bull General Electric, avec pour mission, au sein du nouvel ensemble, de concevoir et fabriquer les ordinateurs moyens et des périphériques.

L'histoire de Bull s'écrit désormais aussi de l'autre côté de l'Atlantique. L'apport de General Electric est déterminant sur le plan de la gestion financière, du management, de l'organisation et des équipes d'études et de la stratégie commerciale. GE apporte notamment son savoir faire en haut de gamme et plus particulièrement dans les multi-processeurs utilisés pour des applications critiques, prenant ainsi le relais du Gamma 60 dans le catalogue. C'est à partir de cette année que sont distribués les ordinateurs de General Electric et CMB. **En juillet 1969, deux ordinateurs Bull-GE assurent le contrôle des organes vitaux de la fusée Saturne qui emmène les astronautes d'Apollo XI faire leurs premiers pas sur la lune.** En France, le lancement en 1966 du Plan Calcul, prise de conscience de l'enjeu de la souveraineté informatique, aboutira à la création de la CII (Compagnie internationale informatique). En 1967, General Electric porte sa participation à 66%.

**1970** : Bull General Electric, dont les actionnaires sont la Compagnie des Machines Bull devenue société holding, et General Electric. Mais en 1970 General Electric décide de ne pas poursuivre son activité Informatique et cède celle-ci à un autre américain, Honeywell. Bull GE devient Honeywell-Bull.

**1973** : lancement du Micral N, le premier microordinateur commercialisé au monde par la société R2E qui sera rapidement rachetée par Bull. Développé à partir d'un microprocesseur du commerce, Intel 8008, il préfigure l'arrivée de l'informatique aussi bien dans les petites entreprises que chez les particuliers. Il marque aussi le début d'une collaboration avec Intel qui deviendra très régulière.



FIGURE 7 – Micral N

**1974** : Honeywell-Bull, a structuré une gamme complète, la série 60, allant du mini-ordinateur aux grands systèmes et pour laquelle ont été développés les systèmes d'exploitation GCOS. Les produits sont conçus et fabriqués tant aux Etats-Unis (PHOENIX le 66 et Boston le Mini 6), qu'en Europe (Pregnana le 62 et Paris le 61 et le 64 dont NEC acquiert la licence et en dérivera sa gamme de référence). Si ces produits sont des réussites, il n'en est pas moins vrai que la multiplicité des laboratoires d'études freine la mise en commun de nombreux développements et induit des coûts supplémentaires. Les produits équipent aussi bien les plus grandes organisations que des PME et tout particulièrement dans ce cas, les applicatifs prennent une part déterminante dans l'atteinte des objectifs des clients. La campagne de communication intitulée « L'informatique créative » reflète cette inflexion : Bull fait parler ses clients : les marques Charles Jourdan, Knoll, Seb ou les restaurants



FIGURE 8 – Serie 60

Jacques Borel accepteront ainsi de témoigner de leurs choix en faveur de Bull. Les partenariats avec les producteurs d'applicatifs que sont les SSII prennent une ampleur déterminante et consommatrice de financement.

**1976** : le gouvernement décide de regrouper Honeywell Bull avec les activités de la Compagnie Internationale pour l'Informatique (CII, qu'il avait créée en 1966). Elle **choisit l'arbre comme symbole fédérateur**. L'arbre est un symbole universel de savoir, de croissance et de vitalité. Il représente l'avenir et la croissance de Bull et de ses clients.

L'actionnaire majoritaire de la nouvelle compagnie CII Honeywell Bull redevient français (la CMB avec 53% du capital). En quatre ans, le chiffre d'affaires double mais le problème de financement de la croissance n'est pas résolu et l'entreprise souffre de sous-capitalisation. En matière de coûts d'études, ce sont deux nouvelles lignes de produit à faire converger avec les quatre précédemment citées ; un ambitieux programme est mis en place en ce sens dont la réalisation la plus significative sera l'architecture de réseau DSA.

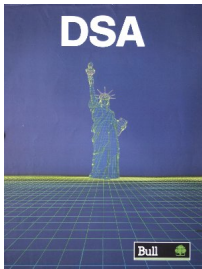


FIGURE 9 – DSA

**1979** : Saint-Gobain entre dans le capital de CMB et en devient majoritaire l'année suivante. Pour accélérer les progrès technologique, CII Honeywell Bull crée un « Centre de Recherche » et intensifie ses coopérations avec les milieux universitaires français et étrangers.



FIGURE 10 – CP8

Cette année-là commence la commercialisation de la Carte CP8, « première à carte micro-processeur » ; cette « carte à puce » (à distinguer de la simple carte à mémoire type paiement téléphonique) est destinée aussi bien à des applications monétiques, qu'au contrôle d'accès ou à des dossiers portables. En plus de cette carte, Bull va proposer progressivement des solutions complètes en maîtrisant également les terminaux de paiement ou les systèmes de transactions sécurisés. Cette activité florissante sera finalement cédée vingt ans plus tard.

Toujours en **1979** CII Honeywell Bull annonce DSA (Distributed Systems Architecture), son architecture de réseau. Alors que s'ouvre l'ère des réseaux d'ordinateurs, DSA était un ensemble de conventions et de protocoles assurant la mise en connexion des systèmes et terminaux au travers de réseaux de transmission publics ou privés. En permettant aux machines Bull de commencer à échanger leurs données depuis les quatre coins du monde, DSA était symbolique du début du rapprochement entre l'univers des télécommunications et celui de l'informatique. Basée sur des protocoles normalisés par l'ISO cette architecture, dans son approche, préluait au monde des « Open architectures ». Une compétence que Bull a beaucoup développée depuis.

En **mai 1981**, ce sont les équipes de Bull qui assurent, à la télévision, les **estimations du résultat de l'élection présidentielle**. Elles affichent pour la première fois, en direct, le visage numérisé du vainqueur : François Mitterrand.

#### 1.1.4 1982-1990 : Dimension mondiale

**1982** : l'État devient l'actionnaire principal de la Compagnie des Machines Bull. Pôle de développement de l'industrie informatique française, la compagnie voit sa compétence dans la mini-informatique renforcée par l'annonce du regroupement de CII Honeywell Bull, de la SEMS (filiale du groupe Thomson), de DAP (département des activités péri-informatique de Thomson) et de Transac (société d'informatique et de bureautique du groupe Alcatel). La constitution d'un nouveau groupe autour de CMB a permis le redressement et le développement de CII Honeywell Bull.

Cette même année, une nouvelle équipe menée par Jacques Stern et Francis Lorentz met en œuvre une politique de croissance externe et d'investissement important dans la R&D, l'outil industriel, la qualité et la force commerciale.

**1983 : choix du nom Bull.** Cinquante années d'efforts et de progrès ont permis à l'entreprise de conclure les alliances nécessaires, sauvegarder ses compétences, s'enrichir d'expérience multiples et conserver intacte la volonté de développer une informatique européenne capable de s'imposer sur les marchés mondiaux.

**1984** : François Mitterrand, Président de la République, vient chez Bull, le 2 mai 1984 célébrer la sortie du 1000ème DPS7. Cet ordinateur de la gamme DPS qui a, depuis 5 ans, remplacé le modèle 64 de la série 60 rencontre un très grand succès commercial. Bull redevenue française en 1976, est désormais le leader européen de l'informatique et de la bureautique, et met en œuvre en **1985** une stratégie sur trois axes :

- Offre à l'utilisateur d'informatique distribuée (poste de travail, serveurs départementaux, accès aux systèmes informatiques)
- Offre solution (approche sectorielle avec les SSII, ouverture au monde informatique, services associés)

- Coopération technologique et prospective (programmes de recherche européens, accords mondiaux sur produits à haute technologie)

Au début des années 80, Bull cherche une idée novatrice pour dynamiser son image de marque et son personnel. Elle décide de sponsoriser un projet, à l'époque très audacieux : **financer un voilier de course pour gagner la WHITBREAD**, la course à la voile la plus prestigieuse du moment. L'idée est de créer un état d'esprit au sein d'une équipe soudée. C'est ainsi que les voiliers de course furent appelés **ESPRIT D'EQUIPES**. C'est LIONEL PEAN qui eut la responsabilité d'amener ce projet à terme. Il remporta la WHITBREAD le 12 mai 1986, ce fut le premier voilier français à gagner cette course.

**En 1988** Bull sort un système DPS7 en technologie CMOS deux ans avant ses concurrents ; en **1991** le chip du haut de gamme de cette ligne est plus dense que celui réalisé par Intel pour ses propres processeurs ; il faudra attendre Itanium pour être dépassé par les chips des grands fondeurs. Mais les volumes ne suivent pas, souvent faute d'offre applicative laquelle se focalise progressivement sur les architectures IBM et Unix ; mais aussi le volume de vente des PC fournit aux fournisseurs de leurs CPU une assise incomparable pour financer les générations successives.

**1987** : à la suite de la décision de Honeywell de se retirer de l'informatique, Bull reprend les activités de son partenaire américain Honeywell Information Systems, exercées dans une nouvelle société, Honeywell Bull Inc. dont le siège est à Minneapolis. Bull (42,5%) Honeywell (42,5%) et NEC (15%) s'en partagent le capital, et en prend le contrôle en **1988** ; tout comme Zenith Data Systems, constructeur américain de micro-ordinateurs, en **1989**. Parallèlement à ces acquisitions, Bull s'appuie sur une politique de coopération intensive avec de nombreuses entreprises et institutions publiques et privées (aux échelles européenne et mondiale) en matière de produits, procédés ou pour l'élaboration de standards internationaux.

Dans ces années-là, Bull et Honeywell se retireront progressivement de la fabrication de périphériques disques puis bandes et enfin imprimantes, ces composants standardisés étant désormais achetés auprès de fournisseurs spécialisés qui, petit à petit, fourniront toute l'industrie y compris IBM. Bull Belfort ferme en 1991.

**1990** : coopération avec le groupe Videoton pour l'implantation de Bull en Hongrie et dans les pays de l'Est. Des coopérations similaires sont initialisées en Inde et au Brésil, sans générer d'inflexion significative des revenus.

#### 1.1.5 1991-1997 : Actionnariat, privatisation, intense adaptation de l'offre et de l'organisation

**1991** : ouverture du capital de CMB à France Télécom et NEC.

En 1991, Bull commence à revoir à la baisse les ambitions de la stratégie d'expansion. Le marché des systèmes GCOS commence en effet à s'éroder, en raison des coûts de développement encore élevés et de la limitation de l'effort commercial au parc existant.

Les systèmes ouverts se montrent décevants, la clientèle n'ayant plus ou presque de raison à rester fidèle au même fournisseur. Une tentative pour retrouver une économie de "mainframe" sur une base de système ouvert est tentée avec le programme Sagister, qui reste bien en deçà de ses espoirs.

Bull entreprend de s'allier avec un partenaire informatique pour une collaboration sur les systèmes UNIX. Après une investigation tous azimuts (incluant Intel, Digital Equipment et des plus petites sociétés), la recherche se porte sur IBM qui avait plus de raisons de respecter l'indépendance de Bull.

**1992** : Pour sa gamme UNIX, Bull doit sélectionner une architecture offrant les meilleurs potentiels en matière d'évolutivité et de partenariats. Bull adopte l'architecture PowerPC (commune à IBM et à Motorola) : un multi-

processeur destiné à être vendu par IBM et Bull, et fait l'objet d'une coopération technique dans le domaine du logiciel (Grenoble). Dans le cadre de cet accord, IBM prend 4,5% du capital de Bull.

**1994** : recapitalisation par l'Etat français et France Télécom, avec pour mission un redressement viable permettant une future privatisation.

**1995** : L'Etat réduit sa part chez Bull grâce à une ouverture du capital et à la constitution d'un socle d'actionnaires industriels durables, cohérent avec le développement du groupe et de ses métiers. Entrée de Motorola (17%), Dai Nippon Printing (DNP, 3,3%) et du personnel de Bull (4%). L'Etat conserve 37% (79,6% précédemment).

**1997** : la part du secteur public (Etat français 30,5% et France Télécom 18,5%) passe en février sous la barre des 50%, laissant la majorité aux actionnaires privés, aux industriels et aux salariés. Cette stratégie permet à Bull de concentrer ses efforts dans les secteurs à forte valeur ajoutée où sa compétence est reconnue et recherchée, et de s'appuyer sur ses partenaires industriels pour offrir des solutions globales à ses clients.

La privatisation s'achève avec l'ouverture du capital au public, réalisée en avril. Cette étape fait passer la part de l'Etat à 17,3%, France Télécom, Motorola et NEC portent leur participation à 17,7%, DNP à 5,5% et le flottant (institutionnels, individuels et salariés) à 24%.

Nomination en septembre de Guy de Panafieu à la présidence du Groupe. Maintenir les métiers traditionnels les plus rentables et développer les futurs métiers de croissance sont les priorités, exprimées dans le Plan Stratégique 2002. Bull décide de concentrer ses efforts dans les logiciels de sécurité, l'administration de systèmes et de réseaux, la carte à microcalculateurs et le commerce électronique.

Bull développe également ses compétences dans les services, dans les activités d'intégration de systèmes, d'infogérance, l'informatique en réseau, internet et intranet, secteurs en pleine croissance. Mais le coût de ces adaptations quelque peu tardives dépasse les capacités d'autofinancement sans pour autant atteindre la masse critique permettant d'être profitable.

Du point de vue de l'offre, les ambitions d'être un acteur de l'industrie du logiciel sont considérablement réduites et progressivement, Bull reconnaît que Linux (comme système d'exploitation) et Internet (dans les réseaux) s'avèrent incontournables. De même, dans le domaine des processeurs, Bull renonce progressivement à développer son offre PowerPC pour se tourner vers Intel.

En bas de gamme, son alliance avec NEC lui permet de commercialiser des machines x86. En haut de gamme, Bull prépare des machines Itanium pour lesquelles il développe ce qui est aujourd'hui connu comme la gamme NovaScale.

Pour cette gamme de produits, Bull a développé des émulateurs de ses systèmes GCOS de façon à ne plus avoir à supporter des dépenses de hardware pour les deux lignes de produits propriétaires subsistantes.



FIGURE 11 – Novascale

La gamme NovaScale a permis en outre à Bull de prendre une position unique en Europe dans le domaine très compétitif des superordinateurs (Tera 10).

Tout ceci implique que Bull apprenne de nouveaux métiers et tisse de nouvelles alliances sans pouvoir laisser de côté certains métiers traditionnels. C'est dans cette période que les études de Bull redéployent sur des processeurs standards Intel les architectures DPS de milieu et haut de gamme. Toujours dans cette période, les études réalisent les premiers clusters de multiprocesseurs (jusqu'à 16 processeurs) à la base des interconnexions des HPC d'aujourd'hui, et créent les premières « practices » ciblant quelques métiers clients grâce à des accords avec les grands fournisseurs de progiciels.

## 1.2 Aujourd'hui

### 1.2.1 Cap sur l'innovation

Depuis le début des années 1990, le Groupe a connu plusieurs restructurations, dont la dernière s'est achevée en 2004 sous l'impulsion du président Pierre Bonelli. (recapitalisation de 400 millions de francs avec l'aide de l'État français).

**Début 1999**, les effectifs de Bull étaient légèrement supérieurs à 20 000 personnes. Fin 2001, les effectifs de la société s'élèvent à 10 000 personnes.

**2000** : Depuis toujours soucieux d'apporter à ses clients les moyens de profiter des opportunités offertes par la technologie, Bull entend les aider à entrer dans la « nouvelle économie ». La campagne « Network of confidence » illustre parfaitement la démarche proposée : faire évoluer les infrastructures et transformer les processus pour s'intégrer au réseau et tirer parti de son potentiel, mais aussi mettre l'accent sur la sécurité. Bull a déjà compris que la confiance serait l'un des piliers de la société numérique alors en gestation.

**2001** : le Groupe vend son activité cartes à puce à Schlumberger, aujourd'hui Gemalto. Il vend également des activités de services en Europe à Steria (hors France). En matière d'offre, Bull confirme un engagement technologique novateur dans les technologies ouvertes, qui conduit **dès 2002** à la fondation du premier consortium mondial dédié aux logiciels d'infrastructure libres, ObjectWeb (aujourd'hui OW2), et au lancement en 2003 d'une nouvelle génération de serveurs ouverts pour les applications commerciales et scientifiques, NovaScale.

**2004** : Bull prend la commande d'un super-ordinateur de la part de la division simulation du CEA/DAM.

**2005** : Sortant victorieux d'une période complexe de restructuration, le groupe Bull prend conscience d'un déficit d'image et décide de reprendre la parole pour marquer son retour. Un nouveau logo, un slogan « Architecte d'un monde ouvert » qui définit son périmètre et la restructuration progressive de son offre autour de produits et solutions innovantes : NovaScale, globull, mobull, bullx, bullion, biodatacenter ... viennent renforcer sa position de leader européen des systèmes numériques critiques. Une offre de services globale est lancée, permettant de concevoir, bâtir et exploiter les applications critiques d'entreprise en s'appuyant sur toute la richesse fonctionnelle des logiciels libres. La signature de contrats d'envergure mondiale confirme le succès de cette stratégie ainsi que le potentiel technologique et commercial du Groupe.

En **Mars**, le gouvernement vend le reste de sa participation dans Bull. France Télécom reste l'actionnaire le plus important.

**Novembre 2005** : livraison du Tera 10, super-ordinateur au CE/DAM. Tera 10 forme un cluster de 544 noeuds de calcul NovaScale, comportant chacun huit processeurs double cœur Intel® Itanium® 2 de nouvelle génération.



L'ensemble constitue une capacité de traitement de 4532 processeurs double cœur et 30 Tera octets de mémoire centrale. La performance atteinte, mesurée sur 4000 processeurs, est de 42,9 Tera flops, démontrant ainsi la remarquable efficacité de l'infrastructure retenue pour Tera 10.

**L'année 2006** a été une année de transformation du Groupe, associant d'importantes avancées, notamment dans le calcul haute performance, les télécommunications et les services. La transformation du Groupe s'est poursuivie en 2007 avec des acquisitions ciblées – en particulier l'acquisition en Espagne de la société de services Siconet, et en France celle de Serviware, principal intégrateur de solutions pour le Calcul haute performance.

**En 2007**, Bull lance son programme 7i. Ce sont sept initiatives pour aider les entreprises à tirer profit d'un monde ouvert. Conjuguant le meilleur des services et des technologies ouvertes, Bull entend aider les entreprises à faire de leurs systèmes d'information (SI) un levier de création de valeur dans un monde connecté, en facilitant croissance, compétitivité et souveraineté.

**L'année 2008** a permis de poursuivre la voie de la transformation du Groupe et du développement de ses activités d'avenir, avec en particulier l'acquisition en Belgique de CSB Consulting, société de services informatiques, et en Allemagne de la société science+computing, leader dans les services et les solutions pour le Calcul haute performance. Bull emploie environ 7 800 salariés et recrute à nouveau fortement (1000 personnes en 2008), ce qui porte le total des salariés à 8 850 en 2009.

**En 2009** Bull confirme sa position d'acteur européen majeur de l'économie numérique avec le lancement d'innovations significatives : bullx (élu meilleur supercalculateur au monde), mobull (une révolution des centres de données), et des réalisations de très grande envergure (Chorus, supercalculateur pour le Forschungszentrum Jülich en Allemagne, hébergement et exploitation de mon.service-public.fr, etc.). Le chiffre d'affaires est de 1,1 milliard d'euros en 2009, réalisé à 52% en France.

**L'année 2010** a permis à Bull de prendre une nouvelle dimension, avec le lancement de nouveautés importantes (notamment la refonte de ses gammes de serveurs pour le calcul haute performance, les grands systèmes Windows/Linux et les mainframes), des réalisations de très grande envergure (livraison de Tera 100, premier supercalculateur européen), lancement d'une offre innovante pour le calcul à la demande (extreme factory) et l'acquisition du groupe Amesys, un leader européen dans la sécurité et les systèmes critiques.

**Novembre 2010** : le **supercalculateur Tera-100 du CEA se classe à la 6ème place mondiale** avec 1,05 pétaflops sur Linpack et 1,25 en puissance crête théorique. C'est le tout premier calculateur européen à passer la barre symbolique du pétaflops et il est constitué de 4 370 serveurs bullx pour un total de 17 480 processeurs octo-cœurs Intel Xeon 7500 (près de 140 000 cœurs en tout). Le chiffre d'affaires est de 1,25 milliard d'euros, réalisé à 56,3% en France. Le Groupe est présent dans 50 pays, sur tous les continents, et emploie 8750 collaborateurs.



croissance rentable.

Le plan stratégique BullWay 2011-2013, annoncé **fin 2010** par son nouveau PDG Philippe Vannier, a pour objectif de positionner Bull sous trois ans comme un leader européen des systèmes numériques critiques et de mettre le Groupe sur le chemin d'une



## 1.3 Organisations

La société [●] regroupe [ effectif / nombre ] d'employés (on compte environ [●] employés au sein de la société [●] dans le monde), qui sont divisés en [●] départements, tous sous la direction du Président-directeur général, [●] : Bull est organisé en quatre grande partie appelées Bussiness Lines

### 1.3.1 Security Solutions

### 1.3.2 Business Integration Solutions

### 1.3.3 Computing Solutions

### 1.3.4 Innovative products

Au sein de cette société de taille [●], il est aisé de percevoir l'interaction constante entre les différents services décrits plus haut. On distingue [●] circuits de création de richesse suivant qu'il procède [●] ou de [●].

## 2 Le secteur d'activité

### 2.1 Présentation

les concurrents, les besoins des consommateurs...  
Quatre secteurs de prédilections :

#### 2.1.1 Secteur public

#### 2.1.2 Finance

#### 2.1.3 Défense

#### 2.1.4 Télécom

### 2.2 Actualités

#### 2.2.1 Cloud Computing

Partenariat avec SFR pour une solution de Cloud (Computing?).

## Quatrième partie

# Le cadre du stage

### 3 L'équipe

Equipe performance secteur R&D (6 puis 5 membres cause départ Gaetan)  
Organigramme à présenter si possible.  
Expliquer le rôle de l'équipe.

### 4 Fonctionnement

??

## Cinquième partie

# Travaux effectués et apports du stage

La mission principale du stage a consisté à étudier l'accélérateur d'Intel. Le but étant d'évaluer les performances de la carte, mais aussi d'en comprendre le fonctionnement tant sur le plan matériel que logiciel. La carte présente dans nos bureaux, appelé Knights Ferry était une version alpha. Une version beta (Knights Corner) a été disponible plus tard dans les locaux d'Echiroles et j'ai également pu y effectuer des tests.

J'ai également travaillé sur le projet OpenGPU en partenariat notamment avec l'Ecole Centrale de Paris et le CEA, en faisant des tests comparatifs entre GPU et CPU dans une optique de consommation électrique.

Pour mener à bien ces missions, différents matériels étaient accessibles, de l'ordinateur personnel aux serveurs de calculs. Ces différents travaux m'auront aussi permis de mieux appréhender le travail en entreprise, de développer de nouvelles compétences ou d'en améliorer d'autres.

## 5 Travaux effectués

### 5.1 Les outils

- Ordinateur personnel (descriptif).
- Ordinateur, qui contenait l'accélérateur Intel (descriptif).
- Serveurs de calcul bull (description des lames... tableaux etc...).
- Accès au Knights Corner d'Echiroles (description machines echiroles).

### 5.2 Les missions

#### 5.2.1 Knights Ferry

Knights Ferry est le nom donné à la carte MIC (Many Integrated Cores) d'Intel dans sa version alpha (voir figure 12). Il s'agit d'un accélérateur assez similaire à un GPU dans le sens où l'on peut exécuter des noyaux de calcul sur la carte afin d'accélérer les programmes. La mission principale aura donc été de comprendre le fonctionnement de cette carte, d'en évaluer dans un premier temps les performances théoriques puis dans un deuxième temps les performances réelles sur des programmes de tests. De plus, toutes les connaissances et tous les résultats ont été systématiquement reportées sur un wiki interne.

**Documentation** Ce matériel étant nouveau et encore dans une version alpha, son fonctionnement tant au niveau matériel que logiciel était encore inconnu. Il aura donc fallu avant tout lire la documentation fournie. Ce qui m'a permis de connaître les propriétés hardware de la carte, comme le nombre de coeurs, de threads par coeur ou encore le nombre et la taille des registres. Ce travail a été indispensable pour estimer les performances théoriques de la carte et pouvoir corroborer par la suite les performances observées avec le matériel.

J'ai également pu noter la procédure d'installation des drivers et de manière générale l'utilisation de la carte à savoir comment démarrer le MIC, ou encore comment compiler un code pour la carte et les différents modes d'exécution. J'y ai également trouvé la liste des outils indispensables comme le débogueur (voir figure 13) ou encore un programme de monitoring (voir figure 14) ainsi que plusieurs outils d'information et de configuration.

**Drivers** La première chose à faire a donc été l'installation de la carte et des drivers puis de vérifier le bon fonctionnement de la carte. Cette étape s'est déroulée sans problème, en effet l'installation consistait en l'exécution



FIGURE 12 – Knights Ferry

d'un simple script python et les outils fournis m'ont permis de constater que la carte était correctement installée et reconnue.

**Benchmark pour le MIC** Dès lors, le but était de pouvoir évaluer les performances de l'accélérateur tels que le débit mémoire, la puissance de calcul, les latences etc...

Il existe de nombreux programmes de benchmark disponibles, l'idée était donc de porter les tests en notre possession sur le MIC, ce qui dans un premier temps était peut-être moins adapté au hardware particulier de la carte, mais nous permettrait d'appréhender le matériel et d'obtenir des résultats plus rapidement que s'il avait fallu écrire des tests spécifiques au MIC.

L'équipe avait donc un ensemble de tests, écrit en C, pour analyser les performances d'une machine sur chacun de ses composants. Il m'a donc été demandé de porter ses tests pour le MIC.

Contrairement au GPU classique NVidia ou AMD, des codes en C ou FORTRAN existant peuvent directement s'exécuter sur le MIC, sans modifier le code source, si le bon flag a été donné à la compilation. Ceci rend le portage de code simple et rapide. De cette manière le programme peut se lancer et s'exécuter intégralement sur la carte sans interaction avec l'hôte.

La plus grosse difficulté a donc été de compiler correctement ces tests. En effet, le simple fait de rajouter la bonne option de compilation n'était pas suffisante pour compiler la plupart des tests. Il aura fallu pour cela éditer les Makefiles inclure les bons headers et correctement linker les différentes bibliothèques spécifiques au MIC. D'autres programmes, une fois correctement compilés, ne s'exécutaient pas sur le MIC, à cause notamment de bibliothèque dynamique manquante sur la carte, mais aussi du manque de connaissance sur le fonctionnement du MIC. Cette première mission aura donc été une étape importante pour la compréhension et l'utilisation du Knights Ferry.

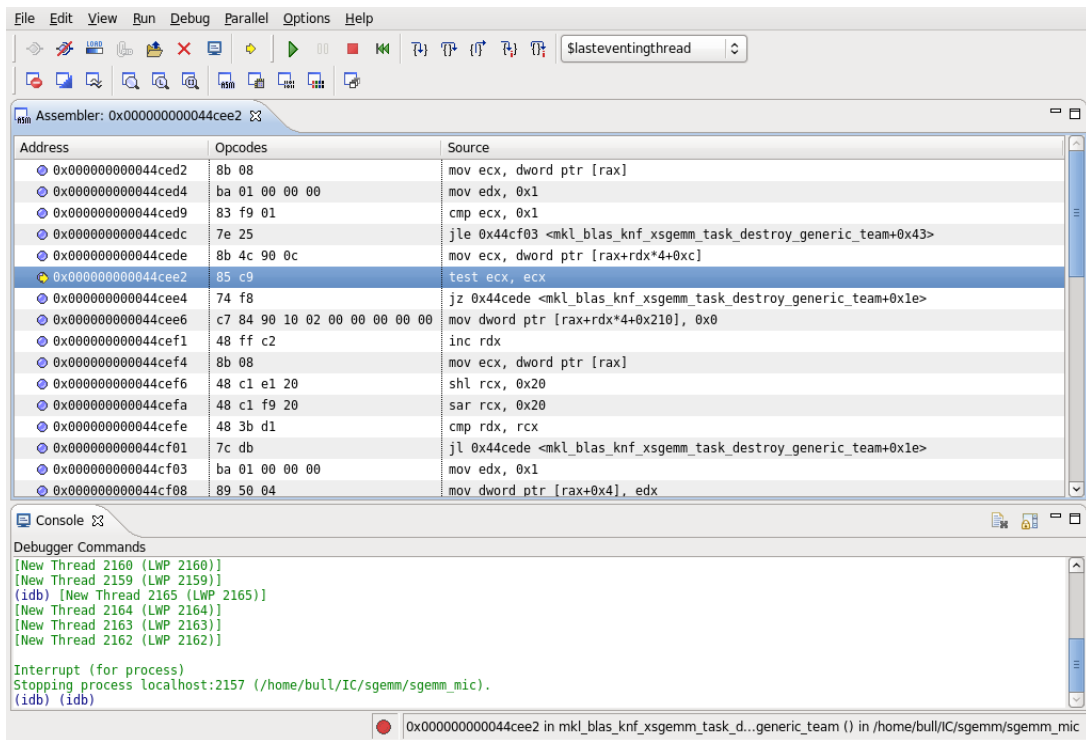


FIGURE 13 – Débogueur MIC d'Intel

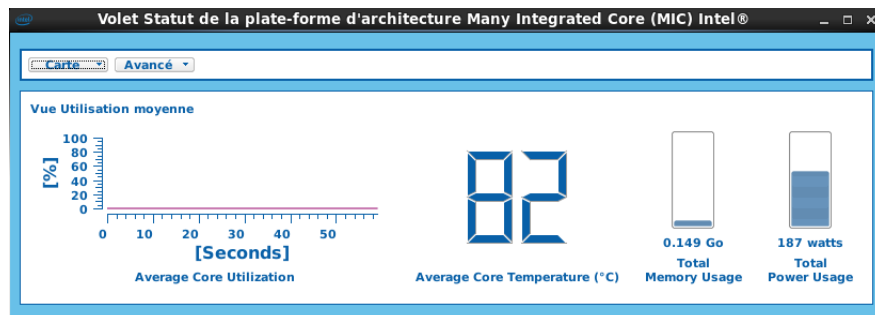


FIGURE 14 – Monitoring MIC

Parmi les programmes de tests standard qui permettent de classer les différentes machines SpecINT et SpecFP sont parmi les plus souvent utilisés. Il a donc semblé intéressant de les exécuter sur le MIC afin de pouvoir comparer la carte à d'autre solution de calcul.

CPU2000 est un ensemble de benchmark conçu pour tester les performances des CPU des serveurs. Le programme est écrit principalement en perl, bash et C. Les tests en eux mêmes sont écrits en C et C++ et il est divisé en deux parties : CINT2000 et CFP2000 pour la partie calculs flottants.

L'exécution de SPECInt peut se décomposer en quatre grandes étapes :

**Compilation :** Les différents programmes de test sont compilés et les paramètres sont donnés à l'aide d'un fichier de configuration édité au préalable pour la machine sur laquelle doit être exécuté SPECInt

**Configuration :** Etant donné les tests à exécuter et les paramètres choisis (taille des données en entrée, type de test à effectuer) les différents benches sont paramétrés.

**Exécution :** Les tests sont effectués et sont mesurés grâce à un binaire nommé "specinvoke".

**Résultat :** Les résultats des tests sont compilés et plusieurs fichiers de sortie sont créés (html, asc etc...) pour résumer les performances.

Il y avait alors deux possibilités pour utiliser SPECInt sur le MIC. Soit tout faire sur le MIC, soit ne faire que la partie exécution et mesures sur la carte.

La deuxième solution a été rapidement choisie car la première avait de nombreux inconvénients. En effet il aurait fallu recompiler l'ensemble de SPECInt pour MIC pour que les outils et binaires puissent s'exécuter dessus. Il aurait également fallu à chaque redémarrage de la carte, réinstaller SPECInt dessus ou alors créer une image du système pour le MIC avec SPECInt installé. Ces solutions étaient beaucoup trop complexes et le choix de la cross-compilation a donc été préféré.

Ainsi, s'il est précisé dans le fichier de configuration, que les tests sont destinés à un matériel de type MIC, alors la compilation se fera sur la machine hôte, mais pour Knights Ferry.

La configuration se fait de la même manière quelque soit le support des tests.

L'exécution doit évidemment se faire sur la carte, l'ensemble des fichiers nécessaires aux tests en question sont donc envoyés sur le MIC. A la fin de l'exécution, les fichiers de résultats sont rapatriés vers l'hôte.

Enfin les résultats sont analysés normalement.

Pour pouvoir réaliser cette procédure, il aura fallu comprendre précisément le fonctionnement et le déroulement des étapes de cette application. Identifier le rôle de chacun des modules et les appels des différentes fonctions afin de savoir exactement, où rajouter proprement les nouvelles fonctionnalités pour MIC. Ceci dans un but de les intégrer complètement dans le processus d'analyse de SpecInt. La plupart des traitements de haut niveau était écrit en Perl, un langage qui m'était inconnu jusque là, ce qui a rajouté une difficulté supplémentaire. Heureusement de très bon tutoriaux existent sur internet, et mes connaissances en Python et Bash m'ont permis d'appréhender rapidement les points essentiels du langage.

Une fois SPECInt fonctionnel pour MIC, l'intérêt a donc été de le comparer à un processeur généraliste de type Sandy Bridge, mais surtout d'en analyser les résultats et d'en comprendre les raisons. (Expliquer la démarche de comparaison, d'abord en mode speed, puis en mode rate)

Dans le même registre que SpecInt, SpecOMP est un ensemble de tests permettant de tester la capacité d'une machine à exécuter des applications multi-threadées. La base de l'application étant strictement la même que SpecInt, c'est-à-dire que l'ensemble de la chaîne de test est la même, et que seuls les tests sont différents, les mêmes modifications faites à SpecInt pour prendre en charge les MICs ont été faites sur SpecOMP.

Ces tests se prétaient très bien au Knights Ferry, qui avec ses 32 coeurs physiques et ses 128 coeurs virtuels était

```

#define ADD(r) vmadd213ps %v##r,%v##r,%v##r
#define MUL(r) vmadd213ps %v##r,%v##r,%v##r
.globl func
.type func, @function
func:
# %rdi : iter
# rdi, rsi, rdx, rcx, r8
.LBB1_func:
    pushq %rbp
    movq %rsp,%rbp
    movq %rdi,%rax

.Lloop:
    ADD(0)
    MUL(1)
    ADD(2)
    ...
    ADD(30)
    MUL(31)
    subq $1,%rax ## loop control
    jg.pt .Lloop ##

.Lexit:
    movq OPS,%rax
    leave
    ret

.LDWend_func:
.size func, .LDWend_func-func

.section .rodata
scalar
.type scalar, @object
.size scalar,8

scalar:
.byte 0
.byte 1071644672
.end

```

FIGURE 15 – Assembleur

un parfait candidat pour les applications multi-threadées. Le but était bien sûr de comparer les résultats obtenus sur MIC, avec ceux obtenus sur une machine Bull. (Expliquer le protocole de test, variation du nombre de thread pour voir l'impact sur les perfs etc...)

**Tests de performances** Pour étudier plus finement les performances il a fallu écrire des tests moins généralistes et plus spécifiques à certaine partie de la carte, comme le processeur ou la mémoire.

Les tests de calculs donnaient des résultats très inférieurs aux performances théoriques et performances attendues. Après analyse il s'est avéré que les codes n'étaient que très rarement vectorisés dû à un compilateur en version alpha, que les bibliothèques de calculs étaient toujours en cours de développement et donc pas encore optimisées pour le MIC et que la seule façon d'obtenir les performances maximums seraient d'utiliser des instruction de type FMA (fused multiply-add) qui réalisent en une instruction des multiplications et des additions sur un vecteur. En partant d'un code assembleur qui permettait de vérifier qu'un processeur supportait l'AVX, nous avons pour chacun des 32 registres de chaque processeur du MIC, exécuté l'instruction *vmadd213ps* qui réalise le produit de deux vecteurs 512 bits (soit 16 floats) puis additionne tous les éléments du vecteur résultat. Une seule instruction réalise donc 32 opérations simple précision. Comme on exécute cette instruction pour les 32 registres, l'ensemble du code réalise 1024 opérations simple précision (Voir figure 15). Ainsi les pipelines des processeurs sont bien remplis et tous les registres sont utilisés. Ce code a permis d'atteindre 90.75% de la performance crête théorique.

Toujours pour tester la puissance de calcul de la carte, mais sur un test moins artificiel, nous avons voulu voir les performances pour des calculs matriciels. SGEMM et DGEMM (pour sa version double précision) sont des fonctions de multiplications de matrices très bien implémentées et optimisées dans les bibliothèques de calcul

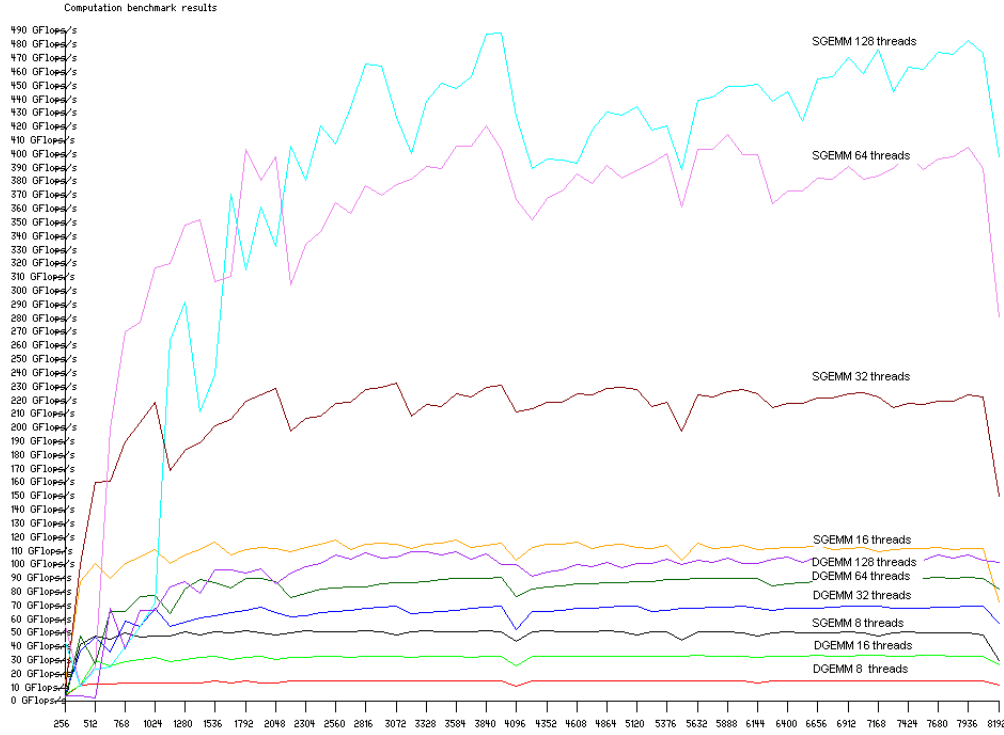


FIGURE 16 – Multiplication Matricielle

notamment dans la MKL (Maths Kernel Library) d'Intel.

La première implémentation du test s'exécutait sur la machine hôte et seule la partie de calcul était exécutée sur le MIC. Il était cependant difficile de pouvoir mesurer uniquement les temps de calculs. En effet l'implémentation pour le MIC de ce type de fonctionnement en sous-traitance, ne sépare pas clairement les parties transfert de données vers la carte des parties de calcul. Il était donc difficile de mesurer uniquement les calculs effectués sur le Knights Ferry.

La seconde version du test était donc complètement exécuté sur la carte. Pour pouvoir mieux visualiser les performances, j'ai implémenté une fonction d'affichage graphique dans le code afin d'avoir instantanément un résultat visuel (voir figure 16).

Un autre composant important à tester est la mémoire. J'ai donc commencé le développement d'un test de mesure de débit. Je me suis pour cela inspiré d'un code existant et qui consistait à faire des chargements mémoires de données de taille croissante. Ainsi le débit de chaque niveau de mémoire pouvait être mesuré. Le code de base étant écrit en assembleur, il a tout d'abord fallu le réécrire pour utiliser les instructions propres au MIC. Malheureusement certaines instructions ne pouvaient pas être remplacées. En conservant les anciennes, le matériel n'était pas complètement exploité et les débits mesurés étaient très loin de ceux espérés. J'ai donc réécrit le code en utilisant des intrinsèques et être ainsi sûr d'utiliser toutes les ressources de la carte. Mais encore une fois les débits



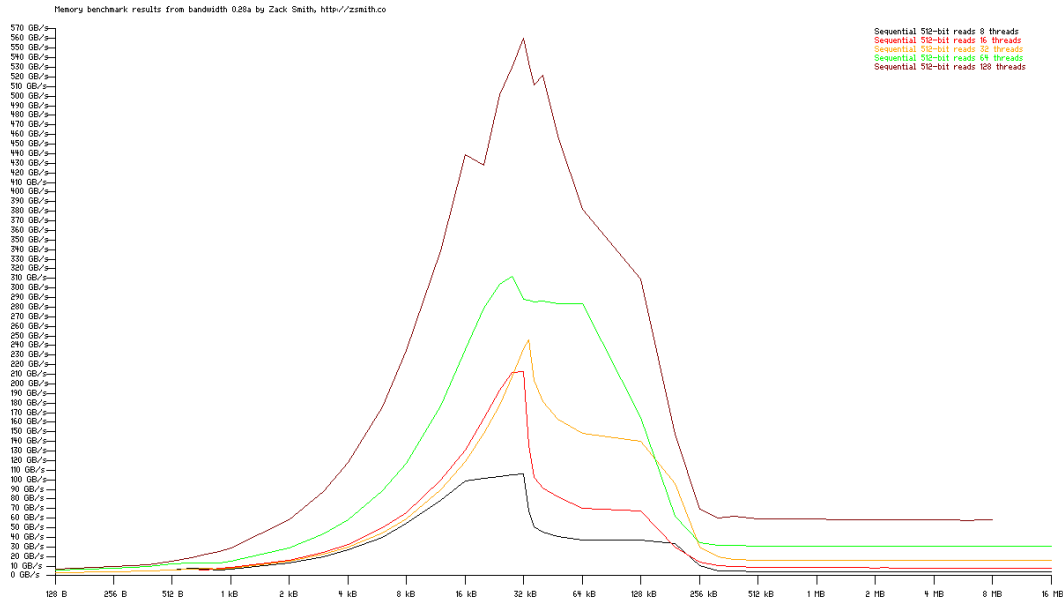


FIGURE 17 – Bandwidth MIC

n'étaient pas ceux annoncés. Après analyse, il a paru évident qu'il n'était pas possible de saturer le débit mémoire en faisant travailler qu'un seul processeur. J'ai donc exécuté le même programme sur plusieurs coeurs jusqu'à arriver à une limite supérieure, qui n'était plus très loin des débits théoriques (Voir figure 17). Cependant, contrairement aux tests sur Ivy Bridge (voir figure 18), les débits du cache sont erronés. Sans doute à cause du trop grand nombre de threads voulant accéder au cache et qui sont donc pénalisés.

Le deuxième métrique que l'on mesure généralement est la latence de la mémoire, à savoir le temps nécessaire pour accéder à une mémoire donnée (voir figure 19).

**Codes externes** Parmi les codes provenant d'organisme client (Ministère de la défense) ou partenaire (CEA), j'ai eu l'occasion de travailler sur deux applications très différentes, mais servant de mesure de référence pour évaluer les performances des machines dans des domaines très précis. L'un d'eux réalise des hachages cryptographiques et effectue des opérations sur des nombres entiers, alors que l'autre, est un programme de mécanique des fluides opérant sur des nombres flottants double précision.

La fonction de hachage implémentée dans le programme cryptographique, a été portée sur différentes architectures pour tirer partie au mieux des ressources disponibles. Ainsi il existe une version non vectorisée, une version 64 bits (SSE) une version 128 bits (AVX), et une version 256 bits (AVX2). Le MIC travaillant sur des vecteurs 512 bits, il s'agissait donc de créer une version 512 bits et ainsi d'étudier les performances de la carte sur des opérations en nombre entier et de voir si le très grand nombre de coeurs de la carte est profitable à ce type d'algorithme. Après avoir compris l'organisation et le fonctionnement du programme, la tâche la plus délicate aura été de trouver

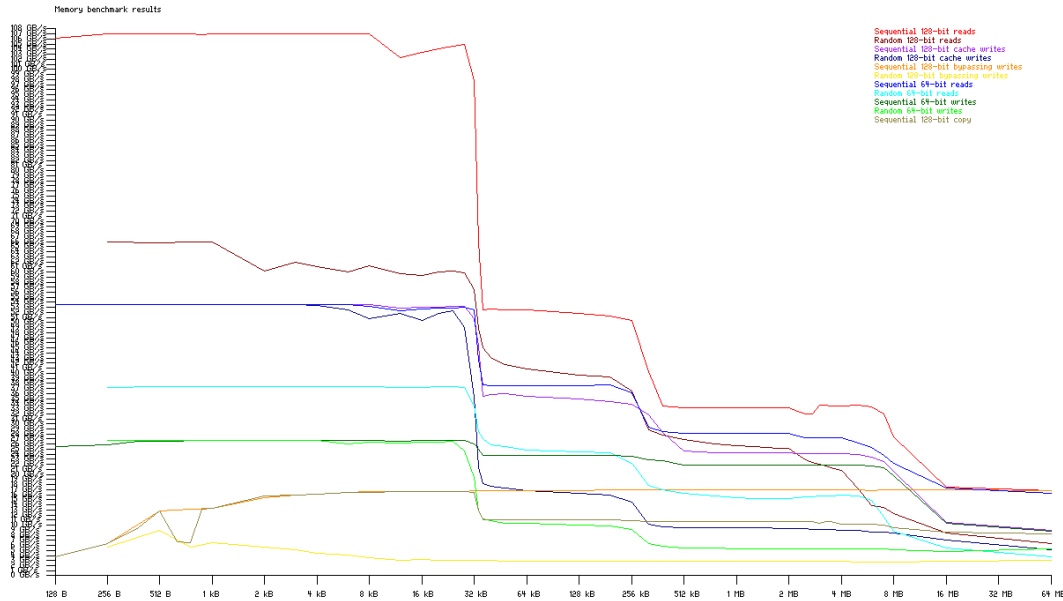


FIGURE 18 – Bandwidth Ivy Bridge

les informations utiles et nécessaires dans la documentation du MIC. En effet les opérations logiques et arithmétiques sont implémentées avec des intrinsics. Il aura fallu trouver les instructions équivalentes pour MIC et les remplacer (voir figure 20).

Les tests ont ensuite consistés à comparer les performances en fonction du nombre de coeurs utilisés et aussi en comparaison avec des processeurs généralistes de type Westmere ou Sandy Bridge.

Le code d'Hydro (un programme de calcul de mécanique des fluides) nous a été fourni par le CEA initialement dans le cadre du projet OpenGPU qui sera développé plus tard. Il s'agit d'une application dérivée d'une application beaucoup plus grosse RAMSES, qui avait été développée dans la division astrophysique du CEA par Romain Teyssier pour étudier la formation des galaxies. Ce code était écrit en Fortran et faisait un usage intensif des couches MPI et était particulièrement scalable. Hydro est une version simplifiée de Ramses, qui résout avec un maillage 2D des équations de dynamique des fluides, il est fondé sur une méthode numérique utilisant un schéma de Godunov de second ordre pour les équations d'Euler. La taille de ce code est intéressante, c'est-à-dire qu'il n'est ni trop petit, ce qui le rendrait non significatif, ni trop gros et donc pas trop compliqué, permettant ainsi de nombreuses variantes et expérimentations. Il effectue beaucoup de calculs en virgule flottante de façon réaliste. Ce code servant à comparer les performances entre CPU et GPU, il nous a paru intéressant de le porter également sur MIC et avoir ainsi une comparaison entre trois solutions de calculs différentes sur un code industriel.

Deux version du programme existait :

- Une version en C pour CPU
- Une version en CUDA pour GPU Nvidia

Je me suis donc basé sur la version CPU pour faire tout d'abord un portage simple vers le MIC et l'exécuter en mode natif, c'est-à-dire entièrement sur la carte et de manière autonome. Même s'il semblait évident que les performances seraient mauvaises avec ce type d'exécution (OS basique, gestion de la mémoire et des threads, quantité de mémoire) cela permettait néanmoins de vérifier que ce code pourrait fonctionner sur la carte.

La seconde étape était d'écrire le code à la manière de CUDA, c'est-à-dire en sous-traitant les noyaux de calcul à la carte et en laissant la machine hôte s'occuper de la gestion des entrées-sorties, des fichiers et des allocations mémoires. Il fallait pour cela analyser le code et trouver les point-chauds, autrement dit les parties de code où le programme passe le plus de temps à calculer. Grâce à ses nombreux coeurs de calcul, le Knights Ferry devrait permettre d'accélérer les calculs comparé à un processeur généraliste. Le concept était donc pour les parties de calcul, de transférer les données sur la carte, d'effectuer les calculs sur cette dernière et de renvoyer les résultats vers l'hôte. Malheureusement les résultats n'étaient pas convaincants et même en dégradation comparés à une exécution classique sur CPU. Il y avait plusieurs raisons à cela :

- Tout d'abord, la carte étant une version Alpha du produit, les performances en calcul flottant double précision étaient très mauvaises comme annoncé par Intel. Malgré le nombre de coeurs de calcul élevé, la carte était moins rapide qu'une lame Inca 3.
- La seconde raison pour expliquer ces performances, est le débit du port PCI-express. En effet le MIC partageait le port PCI-express avec une carte graphique pour l'affichage et le débit était donc deux fois inférieur au débit maximum. Les données transférées entre l'hôte et la carte faisant plusieurs centaines de MégaOctets, un temps considérable était perdu en transfert de données.

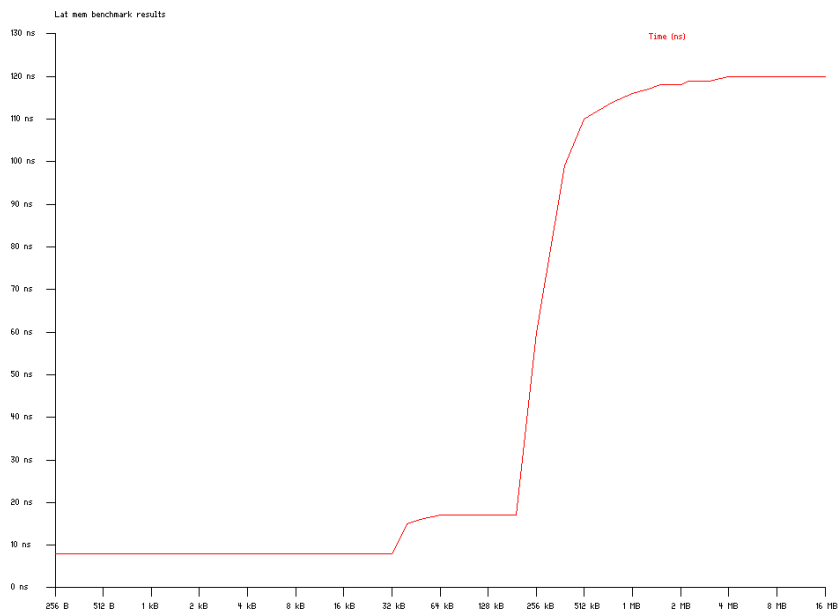


FIGURE 19 – Latence MIC

```

#define f1_512(x,y,z) {tmf=_mm512_xor_pi(x,y);tmf=_mm512_xor_pi(tmf,z);}
#define f2_512(x,y,z) {tmf=_mm512_xor_pi(y,z);tmf=_mm512_and_pi(tmf,x);tmf=_mm512_xor_pi(tmf,z);}
#define f3_512(x,y,z) {__m512i tmp2; tmf=_mm512_or_pi(x,y);tmf=_mm512_and_pi(tmf,z);tmp2=_mm512_and_pi(x,y);tmf=_mm512_or_pi(tmf,tmp2);}
#define f4_512(x,y,z) f1_512(x,y,z)

#define ROTL_512(n,X) {__m512i tm1,tm2,N,T; N=_mm512_set_16to16_pi(n,n,n,n,n,n,n,n,n,n,n,n,n,n,n,n);\
    T=_mm512_set_16to16_pi(32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,32);tm1=_mm512_sll_pi(X,N); tm2=_mm512_srl_pi(X,T-N);\
    X=_mm512_xor_pi(tm1,tm2);}

#define expand_512(W,i) {__m512i tm1,tm2; tm1=_mm512_xor_pi(W[i&15], W[(i-14)&15]);tm2=_mm512_xor_pi(W[(i-8)&15], W[(i-3)&15]);W[i&15]=_mm512_xor_pi(tm1,tm2);}

#define subRound512(a, b, c, d, e, f, k, data) {__m512i sum,tm,tmf;sum=_mm512_add_pi(data,e);f(b,c,d);tm=_mm512_set_16to16_pi(k,k,k,k,k,k,k,k);\
    tm=_mm512_add_pi(tm,tmf);tmf=a; ROTL_512(5,tmf);sum=_mm512_add_pi(sum,tmf);ROTL_512(30,b);e=_mm512_add_pi(sum,tm);}

```

FIGURE 20 – Intrinsics MIC

Pour tenter de palier au premier problème, la possibilité de passer les données en simple précision a été implémenté. Les performances se sont alors nettement améliorées, cependant, les temps de transfert restés trop importants pour espérer pouvoir être plus performant qu'un GPU, ou même qu'un CPU. Les améliorations envisagées pour tenter d'optimiser les transferts n'ont pas aboutie, en effet cela aurait nécessité de réécrire de grosses parties du programme et cette option n'était pas envisageable en temps.

**Outils** Jusqu'alors, l'ensemble des développements avaient été fait dans un but d'évaluer les performances du MIC. Afin de s'assurer que le MIC fonctionnait correctement et pour aller plus loin que les tests fournis par Intel qui se contentaient simplement de vérifier que la carte était reconnue, il m'a été demandé de développer deux tests :

- Un premier test pour évaluer le débit du port PCI-express.
- Et un deuxième pour s'assurer qu'aucune erreur n'était introduite lors des transferts entre l'hôte et la carte.

Encore une fois, plusieurs approches ont été possibles pour réaliser un test de débit du port PCI-express. Le procédure globale du test étant de créer un buffer de taille paramétrable et de l'envoyer de l'hôte vers le MIC, puis du MIC vers l'hôte et de mesurer le temps qui aura été nécessaire afin de calculer le débit dans chacun des sens. La fonction de mesure du temps utilisée est *gettimeofday*. Cette fonction mesure le nombre de secondes et microsecondes écoulées depuis le 1er janvier 1970 minuit. Ainsi, en faisant deux appels à cette fonction, l'un au début de la partie à mesurer et l'autre à la fin, nous pouvons en soustrayant les deux résultats obtenir le nombre de secondes écoulées.

Ainsi dans la version initiale, l'hôte relevé l'heure lors du début du transfert vers le MIC et le MIC relevé à son tour l'heure lorsqu'il avait fini de recevoir les données. Seulement voilà, les débits mesurés étaient tous négatifs. La raison à cela : une heure totalement erronée sur l'OS du MIC et surtout différente de celle sur l'hôte. Rien ne pouvant assurer que les deux heures soient identiques et le test se voulant le plus simple possible à mettre en place, (en évitant de demander à l'utilisateur de configurer l'heure sur le MIC) les mesures ont été faites différemment : l'hôte fait toujours appel à *gettimeofday* avant le début des transferts, mais aussi après les transferts (voir figure 21). cela est possible uniquement grâce aux communications asynchrones. C'est-à-dire que l'hôte reprend la main sur le programme que lorsque la communication est terminée. Ainsi on est sûr d'avoir relevé l'heure qu'une fois les transferts terminés et pas avant. Enfin le programme est paramétrable, on peut choisir la taille des buffers, le nombre de transferts à effectuer et une limite de temps à ne pas dépasser (voir figure 22).

```

gettimeofday(&tbegin_cpu, NULL); //Heure au début des transferts
for(j=0; j<n_transfer; j++)
{
    //Transfert des données vers le MIC
    #pragma offload target(mic)\
    in( buffer1:length(nb_elts) alloc_if(0) free_if(0) )\
    in( buffer2:length(nb_elts) alloc_if(0) free_if(0) )\
    in( buffer3:length(nb_elts) alloc_if(0) free_if(0) )
    {
    }
}
gettimeofday(&tend_cpu, NULL); //Heure à la fin des transferts
texec_cpu=(double)((tend_cpu.tv_sec-tbegin_cpu.tv_sec)+(tend_cpu.tv_usec-tbegin_cpu.tv_usec)*1e-6); //Calcul du temps écoulé

```

FIGURE 21 – Code pour mesurer le temps de transfert CPU vers MIC

```

[bull@mic-pc PCI_test]$ ./bw_test -n 10 -s 100 -v
+-----+
|Buffer size = 100 MB|
|Iterations = 10|
|Verbose = on|
+-----+

Starting main loop

MIC alive, iteration n°0:
cpu sent 5*300Mo in 0.578553 s ==> 2.71862 GB/s
gpu sent 5*300Mo in 0.575351 s ==> 2.73375 GB/s
MIC alive, iteration n°1:
cpu sent 5*300Mo in 0.571144 s ==> 2.75388 GB/s
gpu sent 5*300Mo in 0.574786 s ==> 2.73643 GB/s
MIC alive, iteration n°2:
cpu sent 5*300Mo in 0.572746 s ==> 2.74618 GB/s
gpu sent 5*300Mo in 0.572595 s ==> 2.7469 GB/s
MIC alive, iteration n°3:
cpu sent 5*300Mo in 0.573272 s ==> 2.74366 GB/s
gpu sent 5*300Mo in 0.573967 s ==> 2.74034 GB/s
MIC alive, iteration n°4:
cpu sent 5*300Mo in 0.573031 s ==> 2.74481 GB/s
gpu sent 5*300Mo in 0.574672 s ==> 2.73698 GB/s
MIC alive, iteration n°5:
cpu sent 5*300Mo in 0.572773 s ==> 2.74605 GB/s
gpu sent 5*300Mo in 0.575063 s ==> 2.73512 GB/s
MIC alive, iteration n°6:
cpu sent 5*300Mo in 0.573022 s ==> 2.74486 GB/s
gpu sent 5*300Mo in 0.576077 s ==> 2.7303 GB/s
MIC alive, iteration n°7:
cpu sent 5*300Mo in 0.573219 s ==> 2.74391 GB/s
gpu sent 5*300Mo in 0.573601 s ==> 2.74209 GB/s
MIC alive, iteration n°8:
cpu sent 5*300Mo in 0.572214 s ==> 2.74873 GB/s
gpu sent 5*300Mo in 0.575358 s ==> 2.73371 GB/s
MIC alive, iteration n°9:
cpu sent 5*300Mo in 0.567982 s ==> 2.76921 GB/s
gpu sent 5*300Mo in 0.573942 s ==> 2.74046 GB/s
[bull@mic-pc PCI_test]$

```

FIGURE 22 – Mesure débit PCIe avec un buffer de 100Mo, et 10 transferts

```
[bull@mic-pc PCI_test]$ ./mem_test -n 10 -s 100 -v -c
+-----+
|Buffer size = 100 MB|
|Iterations = 10|
|Continue on error = on|
|Verbose = on|
+-----+

Starting main loop
MIC alive, iteration n°0, Buffer on MIC ok, Buffer on CPU ok, PASSED (0h0m2s)
MIC alive, iteration n°1, Buffer on MIC ok, Buffer on CPU ok, PASSED (0h0m3s)
MIC alive, iteration n°2, Buffer on MIC ok, Buffer on CPU ok, PASSED (0h0m4s)
MIC alive, iteration n°3, Buffer on MIC ok, Buffer on CPU ok, PASSED (0h0m5s)
MIC alive, iteration n°4, Buffer on MIC ok, Buffer on CPU ok, PASSED (0h0m6s)
MIC alive, iteration n°5, Buffer on MIC ok, Buffer on CPU ok, PASSED (0h0m8s)
MIC alive, iteration n°6, Buffer on MIC ok, Buffer on CPU ok, PASSED (0h0m9s)
MIC alive, iteration n°7, Buffer on MIC ok, Buffer on CPU ok, PASSED (0h0m10s)
MIC alive, iteration n°8, Buffer on MIC ok, Buffer on CPU ok, PASSED (0h0m11s)
MIC alive, iteration n°9, Buffer on MIC ok, Buffer on CPU ok, PASSED (0h0m12s)

TEST PASSED : No errors detected
[bull@mic-pc PCI_test]$
```

FIGURE 23 – test PCIe avec un buffer de 100Mo, et 10 itérations

Le second outil de test, devait quant à lui, tester l'intégrité des données circulant sur le port PCI-express entre l'hôte et le MIC. Il existait déjà un test similaire développé au sein de l'équipe mais pour tester les communications entre un CPU et un GPU NVidia.

Le programme a été développé en C. On cré un buffer de N éléments numérotés de 0 à N-1 sur le CPU. Ce buffer est envoyé à la carte qui vérifie les données reçus, les incrémente et les renvoie au CPU. Ce dernier vérifie à son tour que les données sont correct, les incrémente et les renvoie vers le MIC et ainsi de suite. En connaissant les valeurs qui doivent être reçus, on peut ainsi s'assurer qu'aucune valeur n'aura été perdu ou modifiées durant les transfert. En cas d'erreur, le programme donne la valeur lue erronée et la valeur attendue, et l'adresse mémoire de l'élément. Nous pouvons comme pour le test de débit paramétrer l'exécution et choisir la taille du buffer, le nombre d'itérations, si le programme doit s'arrêter ou continuer en cas d'erreur et enfin imposer ou non une limite en temps (Voir figure 23).

### 5.2.2 OpenGPU

Le projet OpenGPU s'inscrit dans un but global de réduction de consommation d'énergie en utilisant des GPUs au lieu de processeurs généralistes. L'intérêt étant donc de savoir, si, à part des applications particulièrement choisies pour fonctionner correctement sur GPU, des applications réelles bénéficient du même gain de performances et de consommation.

Plusieurs partenaires ont ainsi pu se connecter aux machines construites par Bull pour y effectuer leur propre mesures et ainsi d'une part valider les outils de mesures conçus par Bull et d'autre part apporter leur contribution au projet.

Le but est de comparer les efficacités énergétiques des solutions à base de GPU avec celles des solutions sans GPU, et de vérifier ainsi que les GPUs peuvent apporter un gain énergétique important.

**Travaux et résultats** Le code qui nous a servi à effectuer les tests est le code d'Hydro qui a été décrit plus haut. Il a été implémenté en Fortran, en C, avec des couches MPI et OpenMP et porté en CUDA et OpenCL. Le gros intérêt de cette application, dans le cadre d'OpenGPU est qu'elle permet de comparer pour le même travail, le temps passé et la consommation électrique associée sur les architectures à base de GPU et celles qui en sont démunies, en utilisant donc uniquement la puissance délivrée par des processeurs Intel. Mon travail a donc consisté tout d'abord à compiler Hydro sur différentes lames :

- Une à base de Westmere sans GPU.
- Une à base de Sandy Bridge sans GPU.
- Et une avec deux GPUs NVidia.

Il a fallu pour cela adapter les Makefiles et configurer correctement l'environnement (variables d'environnement, version de MPI, etc). J'ai ensuite établie un protocole de test, c'est-à-dire déterminer les paramètres d'exécution afin que les trois mesures soient équivalentes. (Parler de la config pour utiliser plusieurs lames).

(AHHHHHHHH j'ai oublié de parler du problème de la macro qui faisait planter le programme avec la version 11 d'icc. Utilisation de gdb pour comprendre et installation de la version 12 d'icc!!)

Finalement, après plusieurs essais et paramétrages, les tests ont été effectués sur les machines et les résultats fournis aux autres membres de l'équipe.

La comparaison de la consommation énergétique entre différentes solutions, CPU de différents types, GPU, avec une ou plusieurs lames est un complément extrêmement intéressant aux mesures plus traditionnelles se ramenant à comparer la performance des systèmes en question. En effet, on ne mesure plus la performance seule, qui alimentait la course à la puissance, mais aussi la performance/watt, ce qui impose de trouver des solutions les moins énergivores possibles. On s'est aperçu que la méthode employée manquait un peu de précision et qu'il serait intéressant de différencier les consommations élémentaires des processeurs, de la mémoire, des GPUs. On pourrait ainsi s'attacher à mesurer la consommation d'algorithmes et donc comparer l'efficacité de certains algorithmes par rapport à d'autres.

Par contre ce souci d'optimisation énergétique conduit les différents fournisseurs à optimiser leurs différents processeurs. On vient de le voir récemment et on assiste plutôt à une explosion du nombre de cœurs sur les différentes puces, plutôt qu'à une course à la fréquence. Encore faut-il que la scalabilité des applications soit suffisante. Il ne sert à rien de multiplier le nombre de processeurs si les applications n'en tirent pas bien parti.

Dans cette logique d'économie d'énergie, d'autres solutions sont en train d'apparaître, on peut citer par exemple les MICs d'Intel, qui se positionnent sur le même marché que les GPUs, mais avec une philosophie différente de programmation, évitant les couches CUDA ou OpenCL.

Une autre piste pour l'efficacité énergétique est la démarche à base de processeurs ARM. L'idée sous-jacente est que chaque ARM consomme très peu (on économise la batterie du téléphone portable), tout en ayant une puissance de calcul non négligeable. En mettant en commun la puissance de nombreux processeurs ARM, on arrive à une puissance considérable en terme de calcul, pour une consommation électrique qui reste inférieure à d'autres solutions. Toutes ces solutions nécessitent une instrumentation efficace pour pouvoir juger de leur bien-fondé.

### 5.2.3 Knights Corner

Knights Corner est le nom donné à la version Beta du MIC. Cette dernière a été mise à disposition par Intel le 2ème trimestre 2012. De nombreuses modifications ont été apportées par rapport au Knights Ferry, aussi bien au niveau matériel que logiciel. Cette version étant plus proche de la version définitive que le Knights Ferry, l'intérêt était donc de voir si les applications de tests développées seraient compatibles avec la dernière version du MIC, dans le but d'en analyser les performances et de les comparer à celle du Knights Ferry.

**Documentation et drivers** La première étape a donc été de prendre connaissances des modifications apportées par Intel. D'une part côté matériel afin d'estimer les performances théoriques de la carte et d'autre part au niveau logiciel pour pouvoir modifier s'il le fallait les codes des programmes et des scripts existants.

**Le contexte** (Accès par ssh, communication avec Echiroles, etc...)

**Les tests** Dans l'ensemble les tests effectués sur Knights Ferry ont été refaits sur Knights Corner. Certaines parties de code ont cependant étaient réécrites dû surtout aux modifications logiciel faites sur le MIC. En effet l'assembleur a été modifié pour se rapprocher plus des instructions SSE et AVX. Les intrinsics ont également subi de légères modifications.

Ainsi le code du programme mesurant la puissance de calcul a été mis à jour puisque les instructions FMA n'avaient plus la même syntaxe. Quant au code de calcul cryptographique qui utilisait des intrinsics, certaines modifications ont dû être appliquées car des instructions présentes sur Knights Ferry ne l'était plus sur Knights Corner. Enfin, grâce à une quantité de mémoire plus importante, des tests supplémentaires ont pu être fait pour SpecInt et SpecOMP.

### 5.3 Les tâches périphériques

Au cours de mon stage, j'ai pu effectuer des tâches qui se situent à la périphérie du [●]. Dès lors qu'elles m'ont permis d'apprendre différents aspects de [●], il paraît approprié de s'y attarder. Il s'agit du [●] et de [●] :

#### 5.3.1 Wiki

Durant toute la durée de mon stage je me suis attaché à tenir à jour un wiki interne. J'y ai noté les aspects techniques concernant mes travaux sur le MIC afin que chacun puisse profiter de l'expérience et des connaissances acquises. Comme les descriptifs matériels et l'utilisation de la carte côté utilisateur (drivers, outils, compilateur). Mais aussi tous les résultats obtenus, pour chacun des tests pour les version Knights Ferry et Knights Corner ainsi que des explications sur la démarche suivie pour mener à bien les différents tests.

Plus généralement j'y ai noté tout ce qui pouvait sembler utile sur les petites difficultés rencontrées ou sur l'utilisation de certaines commandes linux par exemple.

## 6 Apports du stage

### 6.1 Compétences acquises

Durant mon stage chez Bull, j'ai eu la chance de pouvoir apprendre et améliorer de nombreuses compétences, aussi bien en autonomie, en organisation ou en compétences techniques.

#### 6.1.1 Autonomie

Même si l'équipe m'a évidemment toujours aidé lorsque je rencontrais des difficultés, l'absence au départ d'informations sur les MICs, m'a poussé à chercher les données dans de la documentation pas toujours très claire et à expérimenter pour apprendre à utiliser la carte. Les erreurs commises lors de ma première tâche m'ont d'ailleurs permis d'avancer et de maîtriser le sujet. J'ai, de manière plus générale, toujours essayé de chercher les solutions aux problèmes ou les informations utiles avant de demander autour de moi si vraiment je venais à être bloqué.

#### 6.1.2 Travail en équipe

#### 6.1.3 Organisation

Un des aspects ou je pense avoir le plus appris est l'organisation. Ceci est une conséquence direct du travail en équipe. Le fait de devoir être capable d'expliquer à tout moment ses démarches, l'avancée ou la planification de son travail, pousse à être organiser et à prévoir dans le temps son activité. Le fait de faire des tâches, de la recherche



d'informations ou des applications destinées à être utilisées ou à servir à d'autre personnes, oblige à une certaine rigueur. C'est pourquoi je me suis toujours efforcé de documenter au maximum mes travaux et de tenir à jour un wiki sur les résultats et avancées des tests effectués ou à venir.

D'un point de vue plus technique, et pour les mêmes raisons que précédemment, j'ai fait un effort pour fournir des applications simple d'utilisation avec des Makefiles et des Readme pour faciliter leur utilisation mais aussi pour écrire des codes clairs et commentés.

Seul la gestion du temps aura été plus laborieuse, avec parfois une alternance de surcharge de travail avec des moments plus creux (mois d'aout oblige?).

#### 6.1.4 Compétences techniques

La diversité et le nombre de tâche à accomplir nécessite déjà une certaine polyvalence technique et pousse donc à apprendre toujours de nouvelle chose. J'ai ainsi pu développer mes connaissances sur linux comme sur le noyau et le fonctionnement des modules mais aussi apprendre de nombreuses nouvelles commandes.

L'utilisation de débogueurs aura été aussi très enrichissante, une première fois avec l'interface du débogueur MIC d'Intel afin de déterminer les causes d'une erreur de segmentation sur un programme au comportement non déterministe et une seconde fois avec gdb pour tenter de déboguer Hydro, un programme parallèle, qui se terminait d'une manière inattendue et pouvoir mettre en cause la version du compilateur icc.

J'ai pu aussi améliorer mes connaissances en hardware de par la nature même d'une grosse partie de ma mission, d'analyser les performances d'un accélérateur, qui n'aurait pas pu se faire sans une bonne compréhension du fonctionnement matériel.

Enfin, j'ai appris de nombreuse choses en programmation. Tout d'abord un nouveau langage : le Perl, largement utilisé notamment sous Unix/linux pour sa gestion des fichiers très complète et que j'ai utilisé pour la compatibilité du MIC avec SpecInt et SpecOMP. J'ai également dû écrire des fonctionnalités de programme en assembleur, mais aussi lire de l'assembleur pour par exemple vérifier l'utilisation d'instructions 512 bits sous MIC notamment. La nécessité d'avoir un contrôle assez bas niveau dans des programmes écrit en C a conduit à l'utilisation d'intrinsics que j'avais très peu utilisés jusqu'alors.

## 6.2 Difficultés rencontrées et solutions apportées

### 6.3 La vie en société

Mon stage chez [●] a été très instructif. Au cours de ces [●] mois, j'ai ainsi pu observer le fonctionnement d'une [●]. Au-delà, de l'activité de chacun des services, j'ai pu apprendre comment s'articulent les différents départements d'une telle entreprise. Par ailleurs, les relations humaines entre les différents employés de la société, indépendamment de l'activité exercée par chacun d'eux, m'a appris sur le comportement à avoir en toute circonstance.

#### 6.3.1 L'articulation des différents départements

Comme il a été vu plus haut, [●] départements structurent la société [●]. Aussi, et au travers de l'analyse qui a pu être faite, il apparaît indéniable que tous ces services interviennent à un moment ou a un autre de la prise de décision. Pour autant, cette prise de décision mérite d'être étudiée spécifiquement, dès lors qu'elle est largement [●]. En effet, il est possible de comparer ce type de société [●]. L'expression la plus éloquente de ce partage de responsabilité est la décision de [●]. Aussi, toutes les semaines, les [résultats] étaient diffusés à tous les employés, de manière à ce qu'ils soient impliqués dans les résultats de l'entreprise (sachant que le but est chaque semaine d'avoir un maximum de [●] et à ce qu'ils connaissent [●]. [La circulation de l'information est ainsi un des points

forts que j'ai retenu de cette société, tant au niveau du travail collaboratif, que dans l'implication de tous dans le bon fonctionnement de la société.]

### **6.3.2 Les relations humaines entre les employés**

Au-delà du fonctionnement de l'entreprise, j'ai pu ressentir [●]. En effet, l'atmosphère au sein de la société était très [●]. J'ai ainsi constaté que la hiérarchie des fonctions de la société [●] était [●] dans les rapports entre les employés, favorisant par là [●]. A titre d'exemple, je [●]. [Au travers de cette convivialité, j'ai pu comprendre que l'activité d'une société est plus performante dans une atmosphère chaleureuse et bienveillante.]

## Sixième partie

# Conclusion

[La conclusion résume bien sur, dans une première partie, les principales conclusions de votre rapport de stage. Mais la conclusion permet aussi dans une deuxième partie de vous interroger sur la suite, sur l'avenir de l'entreprise, sur le service, et de mettre en perspective votre stage dans votre formation et dans projet professionnel.] A titre de conclusion, il semble intéressant de mettre en évidence les questions actuelles qui se posent sur l'avenir de l'industrie de [●], de savoir comment les acteurs économiques vont faire [●]. Au centre de cette question se trouve naturellement le problème de [●]. En effet, mon stage a été très bénéfique à cet égard : [●]. Il en résultera [●] La loi du [●] a donné un certain nombre de solutions aux dérives que pourraient prendre le [●]. Pour autant, plutôt que la [●], un changement de [●] apparaît plus à même à régler [●].