# HIGH PERFORMANCE COMPUTING WITH ACCELERATORS

**A SEMINAR REPORT**

***Submitted by***

**AKSHITA K.A**

***In partial fulfillment for the award of the degree***

***Of***

**B-TECH DEGREE**

**In**

COMPUTER SCIENCE & ENGINEERING

**SCHOOL OF ENGINEERING**

**COCHIN UNIVERSITY OF SCIENCE & TECHNOLOGY KOCHI- 682022**

SEPTEMBER, 2010

## ACKNOWLEDGEMENT

First of all I thank **GOD** almighty for his grace and mercy that guided me throughout the seminar

I am extremely grateful to **Dr. David Peter, HOD, Division of Computer Science,** for providing me with the best facilities, work guidance and encouragement. I would like to thank my coordinator, **Mr.Sudheep Elayidom, Sr. Lecturer, Division of Computer Science,** and my guide Mrs**. Sheena S, Lecturer, Division of Computer Science, SOE** for all the help and support extended to me.

I, on this occasion, remember the valuable suggestions and prayers offered by my family members and friends which were inevitable for the successful completion of my seminar

# ABSTRACT

**High-performance computing (HPC)** uses supercomputers and computer clusters to solve advanced computation problems. HPC has come to be applied to business uses of cluster-based supercomputers, such as data warehouses, line-of-business (LOB) applications, and transaction processing.

In the past few years, a new class of HPC systems has emerged. These systems employ unconventional processor architectures—such as IBM's Cell processor and graphics processing units (GPUs)—for heavy computations and use conventional central processing units (CPUs) mostly for non-compute-intensive tasks, such as I/O and communication. Prominent examples of such systems include the Los Alamos National Laboratory's Cell-based Roadrunner) and the Chinese National University of Defence Technology's ATI GPU-based Tianhe-1 cluster.

The main reason computational scientists consider using accelerators is because of the need to increase application performance to either decrease the compute time, increase the size of the science problem that they can compute, or both. The HPC space is challenging since its dominated by applications that use 64 bit floating point calculations and have frequent data reuse. As the size of conventional HPC systems increase, their space and power requirements and operational cost quickly outgrow the available resources and budgets. Thus, metrics such as flops per machine footprint, flops per watt of power, or flops per dollar spent on the hardware and its operation are becoming increasingly important. Accelerator-based HPC systems look particularly attractive considering these metrics.

Types of accelerators in use

1. General purpose Graphical Processing units( GPGPUs) - a specialized microprocessor that offloads and accelerates 3D or 2D graphics rendering from the microprocessor.

2. Field Programmable Gate arrays( FPGAs)- an array of logic gates that can be hardware-programmed to fulfil user-specified tasks.

3 .Clear Speed Floating point accelerators

4. IBM Cell processors.

# TABLE OF CONTENTS

# CHAPTER 1
# INTRODUCTION

**High-performance computing (HPC) uses** supercomputers and computer clusters to solve advanced computation problems. Today, computer systems approaching the teraflops-region are counted as HPC-computers. The term is most commonly associated with computing used for scientific research or computational science. A related term, high-performance technical computing (HPTC), generally refers to the engineering applications of cluster-based computing (such as computational fluid dynamics and the building and testing of virtual prototypes). Recently, HPC has come to be applied to business uses of cluster-based supercomputers, such as data warehouses, line-of-business (LOB) applications, and transaction processing.

High-performance computing (HPC) is a term that arose after the term "supercomputing." HPC is sometimes used as a synonym for supercomputing; but, in other contexts, "supercomputer" is used to refer to a more powerful subset of "high-performance computers," and the term "supercomputing" becomes a subset of "high-performance computing." In computing, hardware acceleration is the use of hardware to perform some function faster than is possible in software running on the general purpose CPU. The hardware that performs the acceleration, when in a separate unit from the CPU, is referred to as a hardware accelerator, or often more specifically as graphics accelerator or floating-point accelerator, etc. Those terms, however, are older and have been replaced with less descriptive terms like video card or graphics card.

Unlike supercomputers, HPC systems usually don’t use custom designed hard-ware and are therefore much more affordable than supercomputers, while at thesame time delivering the same performance. Accelerator-based high-performance computing (HPC) resources are used among computational scientists from the geosciences, computational chemistry, and astronomy and astrophysics communities.

# CHAPTER 2
# STATE OF AFFAIRS

In the past few years, a new class of HPC systems has emerged. These systems employ unconventional processor architectures—such as IBM's Cell processor and graphics processing units (GPUs)—for heavy computations and use conventional central processing units (CPUs) mostly for non-compute-intensive tasks, such as I/O and communication. Prominent examples of such systems include the Los Alamos National Laboratory's Cell-based RoadRunner (ranked second on the December 2009 TOP500 list) and the Chinese National University of Defense Technology's ATI GPU-based Tianhe-1 cluster (ranked fifth on the same TOP500 list).

Currently, there's only one large GPU-based cluster serving the US computational science community—namely, Lincoln, a TeraGrid resource available at NCSA. This will be augmented in the near future by Keeneland, a Georgia Institute of Technology system funded by NSF Track 2D HPC acquisition program. On the more exotic front, Novo-G cluster, which is based on Altera field-programmable gate array (FPGA), is deployed at the University of Florida's NSF Center for High-Performance Reconfigurable Computing (CHREC). By all indications, this trend toward the use of unconventional processor architectures will continue, especially as new GPUs, such as Nvidia's Fermi, are introduced. The top eight systems on the November 2009Green500 list of the world’s most energy efficient supercomputers are accelerator-based.

Despite hardware system availability, however, the computational science community is currently split between early adopters of accelerators and skeptics. The skeptics' main concern is that new computing technologies are introduced frequently, and domain scientists simply don't have time to chase after developments that might fade away quickly.In particular, researchers working with mature and large-scale codes are typically reluctant to practice on the bleeding edge of computing technologies. From their perspective, the accelerator-based systems' long-term viability is a key question that prevents them from porting codes to these systems. Many such codes have been around

much longer than the machines they were originally designed to run on. This continues to be possible because the codes were written using languages (C and Fortran) supported by a range of HPC systems.

With the introduction of application accelerators, new languages and programming models are emerging that eliminate the option to port code between "standard" and "non-standard" architectures. The community fears that these new architectures will result in the creation of many code branches that are not compatible or portable. Mature codes have also been extensively validated and trusted in the community; porting them to newly emerging accelerator architectures will require yet another round of validation. In contrast, early adopters argue that existing HPC resources are insufficient—at least for their applications—and they're willing to rewrite their codes to take advantage of the new systems' capabilities. They're concerned about (but willing to endure) the complexity of porting existing codes or rewriting them from scratch for the new architectures. They're also concerned about (but willing to deal with) the limitations and issues with programming and debugging tools for the accelerators.

Early adopters aren't overly concerned about code portability, because in their view, efforts such as OpenCL and the development of standard libraries (such as Magma, a matrix algebra library for GPU and multicore architectures) will eventually deliver on cross-platform portability. Many early adopters are still porting code kernels to a single accelerator, but a growing number of teams are starting to look beyond simple kernels and single accelerator chips.

# CHAPTER 3
# ACCELERATORS

## Introduction

For many years microprocessor single thread performance has increased at rates consistent with Moore's Law for transistors. In the 1970s-1990s the improvement was mostly obtained by increasing clock frequencies. Clock speeds are now improving slowly and microprocessor vendors are increasing the number of cores per chip to obtain improved performance. This approach is not allowing microprocessors to increase single thread performance at the rates customers have come to expect. Alternative technologies include:

· General Purpose Graphical Processing Units (GPGPUs)

· Field Programmable Gate Arrays (FPGAs) boards

· ClearSpeed's floating-point boards

· IBM's Cell processors

These have the potential to provide single thread performance orders of magnitude faster than current "industry standard" microprocessors from Intel and AMD. Unfortunately performance expectations cited by vendors and in the press are frequently unrealistic due to very high theoretical peak rates, but very low sustainable ones.

Many customers are also constrained by the energy required to power and cool today's computers. Some accelerator technologies require little power per Gflop/s of performance and are attractive from this reason alone. Others accelerators require much more power than can be provided by systems such as blades. Finally, the software development environment for many of the technologies is cumbersome at best to nearly non-existent at worst. An ideal accelerator would have the following characteristics:

· Much faster than standard microprocessors for typical HPC workloads

· Improves price/performance

· Improves performance/watt

· Is easy to program

The HPC space is challenging since it is dominated by applications that use 64-bit floating-point calculations and these frequently have little data reuse. HPCD personnel are also doing joint work with software tool vendors to help ensure their products work well for the HPC environment. This report gives an overview of accelerator technologies, the HPC applications space, hardware accelerators. recommendations on which technologies hold the most promise, and speculations on the future of these technologies.

### 3.2 Accelerator background

Accelerators are computing components containing functional units, together with memory and control systems that can be easily added to computers to speed up portions of applications. They can also be aggregated into groups for supporting acceleration of larger problem sizes. Each accelerator being investigated has many (but not necessarily all) of the following features.

· A slow clock period compared to CPUs

· Aggregate high performance is achieved through parallelism

· Transferring data between the accelerators and CPUs is slow compared to the memory bandwidth available for the primary processors

· Needs lots of data reuse for good performance

· The fewer the bits, the better the performance

· Integer is faster than 32-bit floating-point which is faster than 64-bit floating-point

· Learning the theoretical peak is difficult

· Software tools lacking

· Requires programming in languages designed for the particular technology

### 3.3 High Performance Computing Considerations

Metrics

There are many metrics that can be used to measure the benefit of accelerators. Some important ones to consider are:

· Price/performance (want to increase Gflop/s / $) – the more costly the accelerator, the faster it must be to succeed

· Computational density for system (want to increase Gflop/s / cubic meter) – accelerators can improve this significantly.
· Power considerations (want to increase Gflop/s / watt) – some technologies require very little power while other require so much they can't be used in low power systems
· Cluster system Mean Time Between Failure (want to increase Gflop/s * MTBF) – if accelerators allow a reduction in node count, the MTBF may improve significantly.

# CHAPTER 4
# TYPES OF ACCELERATORS IN USE

## 4.1 GPU

## 4.2 Introduction

A graphics processing unit or GPU (also occasionally called visual processing unit or VPU) is a specialized microprocessor that offloads and accelerates 3D or 2D graphics rendering from the microprocessor. It is used in embedded systems, mobile phones, personal computers, workstations, and game consoles. Modern GPUs are very efficient at manipulating computer graphics, and their highly parallel structure makes them more effective than general-purpose CPUs for a range of complex algorithms. In a personal computer, a GPU can be present on a video card, or it can be on the motherboard. More than 90% of new desktop and notebook computers have integrated GPUs, which are usually far less powerful than those on a dedicated video card.

A GPU (Graphics Processing Unit) is a processor attached to a graphics card dedicated to calculating floating point operations. A graphics accelerator incorporates custom microchips which contain special mathematical operations commonly used in graphics rendering. The efficiency of the microchips therefore determines the effectiveness of the graphics accelerator. They are mainly used for playing 3D games or high-end 3D rendering. A GPU implements a number of graphics primitive operations in a way that makes running them much faster than drawing directly to the screen with the host CPU. The most common operations for early 2D computer graphics include the BitBLT operation, combining several bitmap patterns using a RasterOp, usually in special hardware called a *"blitter"*, and operations for drawing rectangles,triangles, circles, and arcs. Modern GPUs also have support for 3D computer graphics, and typically include digital video–related functions.

The model for GPU computing is to use a CPU and GPU together in a heterogeneous co-processing computing model. The sequential part of the application runs on the CPU and

the computationally-intensive part accelerated by the GPU. From the user's perspective, the application just runs faster because it is using the high-performance of the GPU to boost performance.

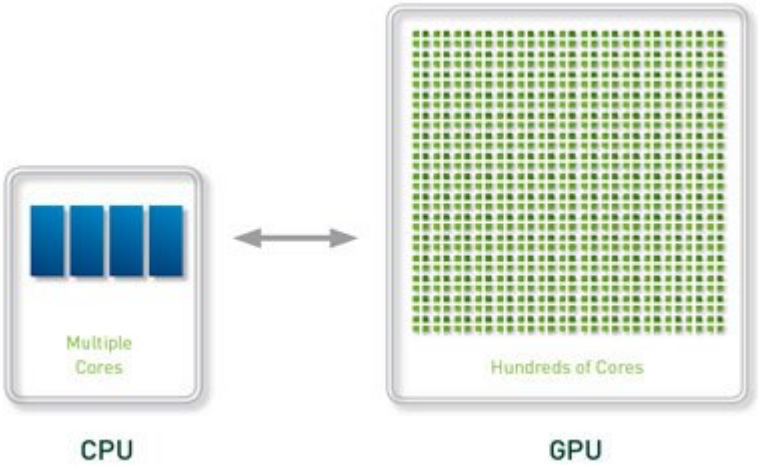


**Fig 4.1.1 CPU AND GPU**

The IBM Professional Graphics Controller was one of the very first 2D/3D graphics accelerators available for the IBM PC.

As the processing power of GPUs has increased, so has their demand for electrical power. High performance GPUs often consume more energy than current CPUs. Another characteristic of high performance GPUs is that they require a lot of power (and hence a lot of cooling). So they're fine for a workstation, but not for systems such as blades that are heavily constrained by cooling. However, floating-point calculations require much less power than graphics calculations. So a GPU performing floatingpoint code might use only half the power of one doing pure graphics code. Most GPUs achieve their best performance by operating on four-tuples each of which is a 32-bit floating-point number. These four components are packed together into a 128-bit word which isoperated on as a group. So it's like a vector of length four and similar to the SSE2 extensions on x86 processors. The ATI R580 has 48 functional units each of which can perform a 4-tuple per cycle and each of those can perform a MADD instruction. At a frequency of 650

MHz, this results in a rate of 0.65 GHz × 48 functional units × 4 per tuple × 2 flops per MADD = 250 Gflop/s. The recent NVIDIA G80 GPU takes a different approach since it includes 32-bitfunctional units instead of 128-bit ones. Each of the 128 scalar units runs at 1.35 GHz and can perform a single 32-bit floating-point MADD operation so its theoretical peak is 1.3 GHz × 128 functional units × 2 flops per MADD = 345 Gflop/s. Unfortunately GPUs tend to have a small number of registers so measured rates are frequently less than 10% of peak. GPUs do have veryrobust memory systems that are faster (but smaller) than that of CPUs. Maximum memory per GPU is about 1 GB and this memory bandwidth may exceed 40 GB/s.

Today, parallel GPUs have begun making computational inroads against the CPU, and a subfield of research, dubbed GPU Computing or GPGPU for *General Purpose Computing on GPU*, has found its way into fields as diverse as oil exploration, scientific image processing, linear algebra[4], 3D reconstruction and even stock options pricing determination. Nvidia's CUDA platform is the most widely adopted programming model for GPU computing, with OpenCL also being offered as an open standard.

### 4.1.2 GPU Classes

The GPUs of the most powerful class typically interface with the motherboard by means of an expansion slot such as PCI Express (PCIe) or Accelerated Graphics Port (AGP) and can usually be replaced or upgraded with relative ease, assuming the motherboard is capable of supporting the upgrade. A few graphics cards still use Peripheral Component Interconnect (PCI) slots, but their bandwidth is so limited that they are generally used only when a PCIe or AGP slot is not available.

A dedicated GPU is not necessarily removable, nor does it necessarily interface with the motherboard in a standard fashion. The term "dedicated" refers to the fact that dedicated graphics cards have RAM that is dedicated to the card's use, not to the fact that *most* dedicated GPUs are removable. Dedicated GPUs for portable computers are most commonly interfaced through a non-standard and often proprietary slot due to size

and weight constraints. Such ports may still be considered PCIe or AGP in terms of their logical host interface, even if they are not physically interchangeable with their counterparts. Technologies such as SLI by NVIDIA and CrossFire by ATI allow multiple GPUs to be used to draw a single image, increasing the processing power available for graphics.

**Integrated graphics solutions**

Integrated graphics solutions, shared graphics solutions, or Integrated graphics processors (IGP) utilize a portion of a computer's system RAM rather than dedicated graphics memory. Computers with integrated graphics account for 90% of all PC shipments. These solutions are less costly to implement than dedicated graphics solutions, but are less capable. Historically, integrated solutions were often considered unfit to play 3D games or run graphically intensive programs but could run less intensive programs such as Adobe Flash. Examples of such IGPs would be offerings from SiS and VIA circa 2004. However, today's integrated solutions such as AMD's Radeon HD 3200 (AMD 780G chipset) and NVIDIA's GeForce 8200 (nForce 710|NVIDIA nForce 730a) are more than capable of handling 2D graphics from Adobe Flash or low stress 3D graphics. However, most integrated graphics still struggle with high-end video games. Chips like the Nvidia GeForce 9400M in Apple's MacBook and MacBook Pro and AMD's Radeon HD 3300 (AMD 790GX) have an improved performance, but still lag behind dedicated graphics cards. Modern desktop motherboards often include an integrated graphics solution and have expansion slots available to add a dedicated graphics card later.

As a GPU is extremely memory intensive, an integrated solution may find itself competing for the already relatively slow system RAM with the CPU, as it has minimal or no dedicated video memory. System RAM may be 2 Gbit/s to 12.8 Gbit/s, yet dedicated GPUs enjoy between 10 Gbit/s to over 100 Gbit/s of bandwidth depending on the model. Older integrated graphics chipsets lacked hardware transform and lighting, but newer ones include it.

**Hybrid solutions**

This newer class of GPUs competes with integrated graphics in the low-end desktop and notebook markets. The most common implementations of this are ATI's HyperMemory and NVIDIA's TurboCache. Hybrid graphics cards are somewhat more expensive than integrated graphics, but much less expensive than dedicated graphics cards. These share memory with the system and have a small dedicated memory cache, to make up for the high latency of the system RAM. Technologies within PCI Express can make this possible. While these solutions are sometimes advertised as having as much as 768MB of RAM, this refers to how much can be shared with the system memory.

**Stream Processing and General Purpose GPUs (GPGPU)**

A new concept is to use a general purpose graphics processing unit as a modified form of stream processor. This concept turns the massive floating-point computational power of a modern graphics accelerator's shader pipeline into general-purpose computing power, as opposed to being hard wired solely to do graphical operations. In certain applications requiring massive vector operations, this can yield several orders of magnitude higher performance than a conventional CPU. The two largest discrete (GPU designers, ATI and NVIDIA, are beginning to pursue this new approach with an array of applications. Both nVidia and ATI have teamed with Stanford University to create a GPU-based client for the Folding@Home distributed computing project, for protein folding calculations. In certain circumstances the GPU calculates forty times faster than the conventional CPUs traditionally used by such applications.

Recently NVidia began releasing cards supporting an API extension to the C programming language CUDA ("Compute Unified Device Architecture"), which allows specified functions from a normal C program to run on the GPU's stream processors. This makes C programs capable of taking advantage of a GPU's ability to operate on large matrices in parallel, while still making use of the CPU when appropriate. CUDA is also the first API to allow CPU-based applications to access directly the

resources of a GPU for more general purpose computing without the limitations of using a graphics API.

Since 2005 there has been interest in using the performance offered by GPUs for evolutionary computation in general, and for accelerating the fitness evaluation in genetic programming in particular. Most approaches compile linear or tree programs on the host PC and transfer the executable to the GPU to be run. Typically the performance advantage is only obtained by running the single active program simultaneously on many example problems in parallel, using the GPU's SIMD architecture. However, substantial acceleration can also be obtained by not compiling the programs, and instead transferring them to the GPU, to be interpreted there. Acceleration can then be obtained by either interpreting multiple programs simultaneously, simultaneously running multiple example problems, or combinations of both. A modern GPU (*e.g.* 8800 GTX or later) can readily simultaneously interpret hundreds of thousands of very small programs.

### 4.1.3 Hardware

There are two dominant producers of high performance GPU chips: NVIDIA and ATI. ATI was purchased by AMD in November 2006. Until recently both GPU companies were very secretive about the internals of their processors. However, now both are opening up their architecture to encourage third party vendors to produce better performing product. ATI's has their Close To Metal (CTM) API. This is claimed to be an Instruction Set Architecture (ISA) for ATI GPUs so that software vendors can develop code using the CTM instructions instead of writing everything in graphics languages. This will make software development easier and will lead to improved performance. NVIDIA is taking a different approach in that they've announced their CUDA program for their latest generation GPUs. CUDA started with the C language, added some new extensions and produced a compiler for the language. Software vendors will write code for CUDA instead of graphics code to achieve improved performance. It remains to be seen which approach is best.

AMD has also announced the Fusion program which will place CPU and GPU cores on a single chip by 2009. An open question is whether the GPU component on the Fusion chips will be performance competitive with ATI's high power GPUs.

### 4.1.4 Software

Most GPU programs are written in a shader language such as OpenGL (Linux, Windows) or HLSL (Windows). These languages are very different from C or Fortran or other common high level languages usually used by HPC scientists. Hence arose the need to explore other languages that would be more acceptable to HPC users.

The most popular alternative to shader languages are streams languages – so named because they operate on streams (vectors of arbitrary length) of data. These are well suited for parallelism and hence GPUs since element in a stream can be operated on by a different functional unit. The first two streams languages for GPUs were BrookGPU and Sh (now named RapidMind). BrookGPU is a language that originated in the Stanford University Graphics Lab to provide a general purpose language for GPUs. This language contains extensions to C that can be used to operate on the four-tuples with single instructions. This effort is currently in maintenance mode because its creator has left Stanford so our team is not pursuing it. However in October 2006, PeakStream announced their successor to BrookGPU. Although they claim their language is really C++ with new classes it looks like a new language. They have created some 32-bit floating-point mathematical routines and we're in the process of evaluating them. PeakStream also is working closely with AMD/ATI, but not with NVIDIA.

The other language we investigated for programming GPUs is RapidMind. This is effort started at the University of Waterloo and led to founding the company RapidMind to productize the language and compilers. This is a language that is embedded in C++ programs and allows GPUs to be abstracted without directly programming in a shader language. While this language is based in graphics programming it is also a general purpose language that can be used for other technical applications. Also, the user does not have to directly define the data passing between the CPU and GPU as the RapidMind

compiler takes care of setting up and handling this communication. Since this language was the only viable GPU language suitable for our market, the authors began a series of technical exchanges with RapidMind personnel. RapidMind has also simplified the syntax of their language to make it easier to use.

**FPGAs**

**4.2.1 Introduction**

Field Programmable Gate Arrays ( FPGPAs) have a long history in embedded processing and specialized computing. These areas include DSP, ASIC prototyping, medial imaging, and other specialized compute intensive areas.

An important differentiator between FPGAs and other accelerators is that they are programmable. You can program them for one algorithm and then reprogram them to do a different one. This reprogramming step may take several milliseconds, so it needs to be done in anticipation of the next algorithm needed to be most effective. FPGA chips seem primitive compared to standard CPUs since some of the things that are basic on standard processors require a lot of effort on FPGAs. For example, CPUs have functional units that perform 64-bit floating-point multiplication as opposed to FPGAs that have primitive low-bit multiplier units that must be pieced together to perform a 64-bit floating-point multiplication. Also, FPGAs aren't designed to hold a large number of data items and instructions, so users have to consider exactly how many lines of code will be sent to the FPGA. Thousands of lines, for example, would exceed the capacity of most FPGAs.
Compared to modern CPUs, FPGAs run at very modest speeds – on the order of 200-600 MHz. This speed is dependent on the overall capability of the device and the complexity of the design being targeted for it. The key to gaining performance from an FPGA lies in the ability to highly pipeline the solution and having multiple pipelines active concurrently.

Running code on FPGAs is cumbersome as it involves some steps that are not necessary for CPUs. Assume an application is written in C/C++. Steps include:

· Profile to identify code to run on FPGA

· Modify code to use FPGA C language (such as Handel-C, Mitrionics, etc.)

· Compile this into a hardware description language (VHLD or Verilog)

· Perform FPGA place-and-route and product FPGA “bitfile”

· Download bitfile to FPGA

· Compile complete application and run on host processor and FPGA

For example, the latest generation and largest Xilinx Virtex-5 chip has 192 25x18 bit primitive multipliers. It takes 5 of these to perform a 64-bit floating-point multiply and these can run at speeds up to 500 MHz. So an upper limit on double precision multiplication is [192/5] * 0.5 = 19 Gflop/s. A matrix-matrix multiplication includes multiplications and additions and the highest claim seen for a complete DGEMM is about 4 Gflop/s, although numbers as high as 8 Gflop/s have been reported for data local to the FPGA. Cray XD-1 results using an FPGA that is about half the size of current FPGAs show DGEMM and double precision 1-d FFTs performing at less than 2 Gflop/s. Single precision routines should run several times faster. FPGAs are very good at small integer and floating-point calculations with a small number of bits. The manager of one university reconfiguration computing site noted: "If FPGAs represent Superman, then double precision calculations are kryptonite."

One HPC discipline that is enamored with FPGAs is astronomy. The current largest very long baseline Interferometry system, LOFAR, has at its heart an IBM Blue Gene system with a peak of 34 Tflop/s. Most of the processing on the Blue Gene systems uses 32-bit floating-point calculations. The next generation system, SKA, to be delivered in the late 2010s will need processing power in the 10-100 Petaflop/s range. The most time consuming algorithms don’t need 32-bit computations - 4-bit complex data and calculations are sufficient. Therefore many of these astronomers are experimenting with FPGAs since three FPGA chips can produce performance that exceeds the equivalent of a Tflop/s. FPGAs belong to a class of products known as Field Programmable Logic devices (FPLD). The traditional and dominant type of FPLD is FPGAs. Recently other types of FPLD have emerged including FPOA (Field Programmable Object Arrays) and FPMC (Field ProgrammableMultiCores).

### 4.2.2 Hardware

The dominant FPGA chip vendors are Xilinx and Altera. Both companies produce many different types of FPGAs. Some FPGAs are designed to perform integer calculations while others are designed for floating-point calculations. Each type comes in many different sizes, so most HPC users would be in interested in the largest (but most expensive) FPGA that is optimized for floating-point calculations. Other chip companies are the startup Velogix (FPGAs) and MathStar (FPOAs).

### 4.2.3 Software

Once again the software environment is not what the HPC community is used to using. There's a spectrum of FPGA software development tools. At one end is the popular hardware design language Verilog used by hardware designers. This has very good performance, but the language is very different from what HPC researches expect. Some vendors have solutions that are much closer to conventional C++. The conventional wisdom is that the closer to standard C that the solution is, the worse the resulting application performs. The reason for this is that to make the best use of FPGAs, users should define exactly how many bits they would like to use for each variable and calculation. The smaller the number of bits, the less space is required on the die, so more can be contained on a chip, and hence the better the performance. One company used by multiple HPC vendors is Celoxica. Its Handel C language allows users to exactly define the data size for all variable and calculations. The HPC accelerator team has begun implementing HPC algorithms in Handel-C to gauge its easy of use and performance.
Another language that has potential is Mitrionics' Mitrion-C programming language for FPGAs.There are also other FPGA C language variants such as Impulse C and Dime-C.

## 4.3 CLEARSPEED'S FLOATING POINT ACCELERATORS

ClearSpeed Technology produces a board that is designed to accelerate floating-point calculations. This board plugs into a PCI-X slot, has a clock cycle of 500 MHz, and contains 96 floating-point functional units that can each perform a double precision multiply-add in one cycle. Therefore their board has a theoretical peak of 96 Gflop/s. In

late 2006 ClearSpeed previewed boards that are connected to systems by a PCI-e slot. This will help performance get closer to their peak rates.

Clearspeed has a beta release of a software development kit that includes a compiler. There are two ways to use the ClearSpeed boards. One is to make a call to a routine from their math library. This library contains an optimized version of the matrix-matrix multiply subprogram DGEMM. The other way to access the boards is to write routines using the ClearSpeed accelerator language Cn. See the "Investigations Finding" for more performance information. The first accelerator enhanced system to make the Top500 list is the TSUBAME grid cluster in Tokyo. It is entry 9 on the November 2006 list and derives about ¼ of its performance from ClearSpeed boards and the rest from Opteron processors.

In stark contrast to nVidia and ATI/AMD ClearSpeed offered single and double precision from the very beginning, aiming specifically at general purpose computational acceleration but at a cost. One of the key ClearSpeed points is very low power and therefore potential for very high computing density.

### 4.4 IBM Cell

#### 4.4.1.Introduction

Cell is a microprocessor architecture jointly developed by Sony Corporation Sony Computer Entertainment, IBM, and Toshiba an alliance known as "STI". Cell is shorthand for Cell Broadband Engine Architecture, commonly abbreviated *CBEA* in full or *Cell BE* in part. Cell combines a general-purpose Power Architecture core of modest performance with treamlined coprocessing elements which greatly accelerate multimedia and vector processing applications, as well as many other forms of dedicated computation.It was initially designed for the Playstation 3. Although the Cell was not on the original list of technologies to evaluate it has become the mind share leader of acceleration technologies. The QS22 based on the PowerXCell 8i processor is used for the IBM Roadrunner supercomputer.

The Cell Broadband Engine—or *Cell* **(Refer Fig 4.4.1.1)** as it is more commonly known—is a microprocessor designed to bridge the gap between conventional desktop processors (such as the Athlon 64, and Core 2 families) and more specialized high-performance processors, such as the NVIDIA and ATI graphics-processors (GPUs). The longer name indicates its intended use, namely as a component in current and future digital distribution systems; as such it may be utilized in high-definition displays and recording equipment, as well as computer entertainment systems for the HDTV era. Additionally the processor may be suited to digital imaging systems (medical, scientific, *etc.*) as well as physical simulation (*e.g.*, scientific and structural engineering modeling).

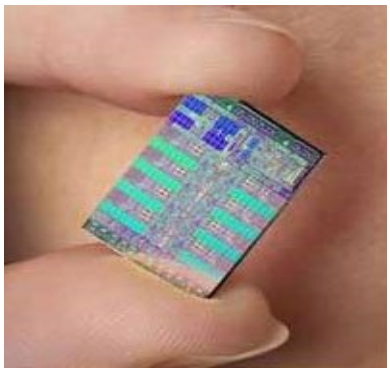

**Fig 4.4.1.1 IBM Cell Processor**

To achieve the high performance needed for mathematically intensive tasks, such as decoding/encoding MPEG streams, generating or transforming three-dimensional data, or undertaking Fourier analysis of data, the Cell processor marries the SPEs and the PPE via EIB to give access, via fully cache coherent DMA (direct memory access), to both main memory and to other external data storage. To make the best of EIB, and to overlap computation and data transfer, each of the nine processing elements (PPE and SPEs) is equipped with a DMA engine. Since the SPE's load/store instructions can only access its own local memory, each SPE entirely depends on DMAs to transfer data to and from the main memory and other SPEs' local memories. A DMA operation can transfer either a single block area of size up to 16KB, or a list of 2 to 2048 such blocks. One of the major design decisions in the architecture of Cell is the use of DMAs as a central means of

intra-chip data transfer, with a view to enabling maximal asynchrony and concurrency in data processing inside a chip.

### 4.4.2 Architecture

Cell has a total of nine cores and is a heterogeneous multiprocessor with aunique design, boasting an impressive theoretical peak performance of over 200 Gflops. Heterogeneous refers to the nine cores, which are of two different types, each specializing in different tasks. This is a completely different approach than other multi-core processors from e.g. Intel and AMD, where all cores are of the same type and therefore have the same strength and weaknesses. Initial research by IBM and others have shown that Cell outperforms these commodity processors, by several factors, for certain types of scientific kernels and can achieve near peak performance. Faster clock speeds, bigger cache, hyper-threading and out of order execution are just some of the more complex measures that have been taken to increase processor performance, and are all meant to decrease latency and execution time.

The first major commercial application of Cell was in Sony's PlayStation 3 game console. Mercury Computer Systems has a dual Cell server, a dual Cell blade configuration, a rugged computer, and a PCI Express accelerator board available in different stages of production. Toshiba has announced plans to incorporate Cell in high definition television sets. Exotic features such as the XDR memory subsystem and coherent Element Interconnect Bus (EIB) interconnect appear to position Cell for future applications in the supercomputing space to exploit the Cell processor's prowess in floating point kernels. IBM has announced plans to incorporate Cell processors as add-on cards into IBM System z9mainframes, to enable them to be used as servers for MMORPGs.

### 4.4.3 HPC and Cell

Several typical HPC kernels have been ported to the Cell processors and their performance compared to other leading commodity processors. Initial result are very impressive, showing that Cell can be up to 25x faster then other leading commodity

processors from Intel and AMD. Results also show that near peak performance is achievable on the Cell processor. Besides the HPC kernels, a small subset of a well know real-world image library has also been ported to the Cell processor. This has resulted in a Cell extension for the library, with optimized library functions, which take advantage of Cell's unique architecture. The extended library encapsulates the complexity of the Cell processor and is very easy to use and requires almost no knowledge about the Cell processor. Comparing the performance of the extended library functions on the Cell processor, with the performance of the original library functions on the commodity processors, shows that the extended Cell library can perform up to 30x times better. Thus initial results are very encouraging and show that Cell has real potential for image processing workloads.

Multi-processor configurations are often used in HPC systems and their performance are therefore interesting to explore. Results show that Cell is very suitable in a dual Cell processor configuration, with performance scaling linearly for nearly all kernels explored. Results show that the IBM XLC compiler outperforms Sony's GCC compiler signif-icantly, however, both compilers are still in very early development and therefore manually written optimizations are essential to reach Cell's peak performance. Finally a HPC cluster has been designed using Cell processors, which has a theoretical double precision peak performance of over one petaflops. This impressive performance can be achieved with only 5000 enhanced Cell Blade servers, and consequently only a very simple network topology is needed to support this. The Cell HPC cluster is estimated to cost roughly $9 million.

### 4.4.4 Cell High Performance Computing Cluster

Since the emergence of the first computers, the appetite for more computing performance as proved to be insatiable. For every year that goes by, companies and research institutes around the world crave for more and more performance. Traditionally this performance has been delivered by monolith supercomputers, however, in recent times monolith supercomputers have given way to High Per-formance Computing (HPC) systems, which are now the most prevalent way of achieving large amounts of computing performance.

According to International Data Corporation (IDC) the HPC market grew more than 102% from 2002 to 2006 and IDC expects that it will grow by an additional 9.1% annually[66, 67]. This turn toward HPC is due to the relatively low cost and high performance of commercial off-the-shelf hardware, which means that supercomputer performance can now be achieved at a fraction of the cost it used to.

A hybrid Cell cluster also has the advantage that cluster applications based on GP processors can run directly on the hybrid Cell HPC cluster, without porting and/or recompilation. One could then port key parts of the applications to use the Cell processors as accelerators. Therefore due to the limitations of the PPE and the above benefits of hybrid clusters, it will be a requirement that the Cell HPC cluster designed in this chapter must, in addition to Cell processors, also consist of a number of GP processors.

The requirements of the Cell HPC cluster can now be listed:

• Must consist of mainly Cell processors.

• Must be a hybrid and consist of a number of general purpose (GP) processors.

• Should roughly cost $9 million (the official price of MDGRAPE-3).

• Should deliver the same theoretical peak performance as MDGRAPE-3, around one petaflop.

• Must have a simple network infrastructure and should be based on either Ethernet or InfiniBand.

At the center of this is commodity processors, where prices have plummeted over the years, while at the same time performance has increased. Commodity processors are now standard in many HPC systems and developers are on constant lookout for new affordable processors which will increase the performance of their next generation HPC systems. Consequently the Cell processor is very interesting. Cell outperforms commodity processors from Intel and AMD by several magnitudes, both in terms of theoretical peak performance and actual performance. Cell is therefore an ideal candidate for HPC.

**PlayStation 3 cluster**

The considerable computing capability of the PlayStation 3's *Cell* microprocessors has raised interest in using multiple, networked PS3s for various tasks that require affordable high-performance computing.

**PS3 Clusters**

The NCSA has already built a cluster based on the PlayStation 3. Terra Soft Solutions has a version of Yellow Dog Linux for the PlayStation 3,and sells PS3s with Linux pre installed, in single units, and 8 and 32 node clusters. In addition, RapidMind is pushing their stream programming package for the PS3.

**Single PS3**

Even a single PS3 can be used to significantly accelerate some computations. Marc Stevens, Arjen K. Lenstra, and Benne de Weger have demonstrated using a single PS3 to perform an MD5 bruteforce in a few hours. They say: "Essentially, a single PlayStation 3 performs like a cluster of 30 PCs at the price of only one" (in November 2007).

The Computational Biochemistry and Biophysics Lab in Barcelona has launched a distributed computing project called PS3GRID. This project is expected to run sixteen times faster than an equivalent project on a standard PC. Like most distributed computing projects, it is designed to run only when the computer is idle. eHiTS Lightning is the first virtual screening and molecular docking software for the PS3. It was released by SimBioSys as reported by Bio-IT World in July 2008. This application runs up to 30x faster on a single PS3 than on a regular single CPU PC, and it also runs on PS3 clusters, achieving screening of huge chemical compound libraries in a matter of hours or days rather than weeks, which used to be the standard expectation. In March 28 2010, Sony announced it would be disabling the ability to run other operations.

**The $2 million PlayStation 3 Supercomputer**

When Sony launched the PlayStation 3 in 2006, the company touted its IBM Cell processor as one of the key features that distinguished the PS3 from other consoles. This

"supercomputer on a chip," would allow developers "for the first time can create games closer to actual intelligence instead of artificial intelligence," Sony claimed.

Although game developers are only now beginning to take advantage of the PS3's processing Ability, the United States Air Force has taken the claim literally. Stars and Stripes newspaper announced a $2 million government project to create a research supercomputer using 2,000 PS3s. The project will be headed by the Air Force Research Laboratory in Rome, New York.

# CHAPTER 5
# (HPC) - SUPERCOMPUTING WITH NVIDIA TESLA

## 5.1 Overview

The NVIDIA® Tesla™ 20-series is designed from the ground up for high performance computing. Based on the next generation CUDA GPU architecture codenamed “Fermi”, which is the third generation CUDA architecture. it supports many “must have” features for technical and enterprise computing. These include ECC memory for uncompromised accuracy and scalability, support for C++ and 8x the double precision performance compared Tesla 10-series GPU computing products. When compared to the latest quad-core CPU, Tesla 20-series GPU computing processors deliver equivalent performance at 1/20th the power consumption and 1/10th the cost.

## 5.2 CUDA Architecture

CUDA is NVIDIA’s parallel computing architecture that enables dramatic increases in computing performance by harnessing the power of the GPU (graphics processing unit).
With millions of CUDA-enabled GPUs sold to date, software developers, scientists and researchers are finding broad-ranging uses for CUDA, including image and video processing, computational biology and chemistry, fluid dynamics simulation, CT image reconstruction, seismic analysis, ray tracing, and much more. Computing is evolving from "central processing" on the CPU to "co-processing" on the CPU and GPU. To enable this new computing paradigm, NVIDIA invented the CUDA parallel computing architecture that is now shipping in GeForce, ION, Quadro, and Tesla GPUs, representing a significant installed base for application developers.

In the consumer market, nearly every major consumer video application has been, or will soon be, accelerated by CUDA, including products from Elemental Technologies, MotionDSP and LoiLo, Inc. CUDA has been enthusiastically received in the area of scientific research. For example, CUDA now accelerates AMBER, a molecular dynamics simulation program used by more than 60,000 researchers in academia and pharmaceutical companies worldwide to accelerate new drug discovery.

An indicator of CUDA adoption is the ramp of the Tesla GPU for GPU computing. There are now more than 700 GPU clusters installed around the world at Fortune 500 companies ranging from Schlumberger and Chevron in the energy sector to BNP Paribas in banking.

The next generation CUDA architecture (codename: "*Fermi*") which is standard on NVIDIA's released (GeForce 400 Series [GF100] (GPU) 2010-03-27) GPU is designed from the ground up to natively support more programming languages such as C++. It has eight times the peak double-precision floating-point performance compared to Nvidia's previous-generation Tesla GPU.

**Fig 5.2.1 Tesla GPU Computing Processor**
**The GPU Computing Processor transforms a standard workstation into a personal supercomputer.**

**Fig 5.2.2 Tesla Personal Supercomputer**
**The Tesla personal supercomputer delivers cluster level computing performance on your desk - 250 times faster than standard PCs and workstations.**

Your own Supercomputer Dedicated computing resource for every computational researcher and technical professional. Cluster Performance on your Desktop The performance of a cluster in a desk topsystem. Four Tesla GPU computing processors deliver close to 4 Teraflops of performance. Massively Parallel Many Core GPU Architecture 240 parallel processor cores per GPU that can execute thousands of

concurrent threads. Solve Large-scale Problems using Multiple GPUs Scale your application to multiple GPUs and harness the performance of thousands of processor cores to solve large-scale problems. 4 GB High-Speed Memory per GPU Enables larger datasets to be stored locallyfor each processor to maximize benefit from the 102GB/s memory transfer speeds and minimize data movement around the system. IEEE 754 Floating Point Precision (single-precision and double-precision) Provides results that are consistent across platforms and meet industry standards. 64-bit ALUs for Double-Precision Math Meets the precision requirements of your most demanding applications with 64-bit ALUs.

The GPU Computing System seamlessly fits into enterprise server clusters and scales to solve the most complex computing problems.

# CHAPTER 6
# JUNE 2010 TOP 500 LIST

## 6.1 Introduction

The TOP500 project ranks and details the 500 (non-distributed) most powerful known computer systems in the world. The project was started in 1993 and publishes an updated list of the supercomputers twice a year. The project aims to provide a reliable basis for tracking and detecting trends in high-performance computing and bases rankings on HPL, a portable implementation of the High-Performance LINPACK benchmark for distributed-memory computers.

## 6.2 Top Highlights

- A Chinese system called Nebulae, build from a Dawning TC3600 Blade system with Intel X5650 processors and NVidia Tesla C2050 GPUs is now the fastest in theoretical peak performance at 2.98 PFlop/s and No. 2 with a Linpack performance of 1.271 PFlop/s. That means it performs 1,270 trillion calculations a second. How long does it take an average PC to match its performance ? Answer: about 9hrs !
- China keeps increasing its number of systems to 24 and is now tied with Germany (steadily declining) for spot No. 4 after the USA, UK and France.
- The Jaguar system at Oak Ridge National Laboratory managed to hold the No. 1 spot with 1.75 PFlop/s Linpack performance even as it’s peak performance is lower than the Chinese Nebulae system.
- The most powerful system in Europe is an IBM BlueGene/P system at the German Forschungszentrum Juelich (FZJ) which dropped to No. 5.
- Intel dominates the high-end processor market 81.6 percent of all systems and over 90 percent of quad-core based systems.
- The Intel Core i7 (Nehalem-EP) processors increased their presence in the list with 186 systems compared with 95 in the last list.

- Quad-core processors are used in 85 percent of the systems and 5 percent use already processors with six or more cores.
- Other notable systems are:
- The Tianhe-1 system at No. 7, which is a hybrid design with Intel Xeon processors and AMD GPUs. The TH-1 uses AMD GPUs as accelerators. Each node consists of two AMD GPUs attached to two Intel Xeon processors.
- IBM regains the lead in market share by total systems from Hewlett-Packard, IBM also stays ahead by overall installed performance.
- The Cray's XT system series remains very popular for big research customers with 10 systems in the TOP50 (20 percent).

**General highlights from the Top 500 since the last edition:**

- Quad-core processor based systems dominate the TOP500 as 425 systems are using them. 48 systems are still using dual-core processors but already 25 systems are using processors with 6 or more cores.
- The entry level to the list moved up to the 24.7 TFlop/s mark on the Linpack benchmark, compared to 20 TFlop/s six months ago.
- The last system on the newest list was listed at position 357 in the previous TOP500 just six months ago. This turnover rate is far below average and might reflect the economic situation as well as an upcoming new product cycle in the HPC market.
- Total combined performance of all 500 systems has grown to 32.4 PFlop/s, compared to 27.6 PFlop/s six months ago and 22.6 PFlop/s one year ago.
- The entry point for the top 100 increased in six months from 47.72 TFlop/s to 52.84 TFlop/s.
- The average concurrency level in the TOP500 is 10,267 cores per system up from 9,174 six month ago and 8,210 one year ago.
- A total of 408 systems (81.6 percent) are now using Intel processors. This is slightly up from six months ago (402 systems, 80.4 percent). Intel continues to provide the processors for the largest share of TOP500 systems.

- They are now followed by the AMD Opteron family with 47 systems (9.4 percent), up from 42.
- The share of IBM Power processors is slowly declining with now 42 systems (8.4 percent), down from 52.
- 424 systems are labeled as clusters, making this the most common architecture in the TOP500 with a stable share of 85 percent.
- Gigabit Ethernet is still the most-used internal system interconnect technology (244 systems), due to its widespread use at industrial customers, followed by InfiniBand technology with 205 systems.
- However, Infiniband based system account for twice as much performance (15.8 PFlop/s) than Gigabit Ethernet ones (7.8 PFlop/s).
- IBM and Hewlett-Packard continue to sell the bulk of systems at all performance levels of the TOP500.
- HP lost its narrow lead in systems and has now 185 systems (37 percent) compared to IBM with 198 systems (39.8 percent). HP had 210 systems (42 percent) six months ago, compared to IBM with 186 systems (37.2 percent).
- IBM remains the clear leader in the TOP500 list in performance with 33.6 percent of installed total performance (down from 35.1 percent), compared to HP with 20.4 percent (down from 23 percent).
- In the system category, Cray, SGI, and Dell follow with 4.2 percent, 3.4 percent and 3.4 percent respectively.
- In the performance category, the manufacturers with more than 5 percent are: Cray (14.8 percent of performance) and SGI (6.6 percent), each of which benefits from large systems in the TOP10.
- HP (167) and IBM (128) sold together 295 out of 302 systems at commercial and industrial customers and have had this important market segment clearly cornered for some time now.
- The U.S. is clearly the leading consumer of HPC systems with 282 of the 500 systems (up from 277). The European share (144 systems – down from 152) is still substantially larger then the Asian share (57 systems – up from 51).

- Dominant countries in Asia are China with 24 systems (up from 21), Japan with 18 systems (up from 16), and India with 5 systems (up from 3).
- In Europe, UK remains the No. 1 with 38 systems (45 six months ago). France passed Germany and has now 29 (up from 26). Germany is still now the No. 3 spot with 24 systems (27 six months ago).

# CHAPTER 7
# CONCLUSIONS AND PREDICTIONS

There are multiple families of accelerators suitable for executing applications from portions of HPC space. These include GPGPUs, FPGAs, ClearSpeed and the Cell processor. Each type is good for specific types of applications, but they all need applications with a high calculation to memory reference ratio. They are best at the following:

· GPGPUs: graphics, 32-bit floating-point

· FPGAs: embedded applications, applications that require a small number of bits

· Clearspeed: matrix-matrix multiplication, 64-bit floating-point

· Cell: graphics, 32-bit floating-point

Common traits for accelerators today include slow clock frequencies, performance is through parallelism, low bandwidth connections to CPU, and the lack of standard software tools.

Future trends include

• faster clock periods, better nm technology, more parallelism improved interconnect bandwidth (PCI-e Gen2, later HT) and latency and the ability to achieve a higher percentage of the interconnect performance

• multicore/multichip accelerators

• heterogeneous processors, that is, some number of CPU cores + some number of accelerators cores of various types

• a better software development environment – users need a standard C compiler which generates code for whichever accelerator is most appropriate and a complete software tool chain that includes debuggers and profilers.

HPC is experiencing a new cycle of innovation in which high performance serial and parallel processing cores are tightly coupled with special-purpose hardware accelerators to enable unprecedented performance levels. Tightly integrated Intel Larrabee and Advanced Micro Devices (AMD) Fusion are just over the horizon, while loosely coupled

multicore systems with many-core GPUs attached are being assembled into large HPC systems with impressive demonstrated capabilities for specific applications. The challenge now is to migrate many more applications to these systems as well as to develop new algorithms that can take full advantage of them while ensuring portability to new generations of platforms.

## REFERENCES

1 http://www.computer.org/portal/web/csdl/magazines/cise#4

2 http://www.hp.com/techservers/hpccn/hpccollaboration/ADCatalyst/downloads/accelerators.pdf

3 http://en.wikipedia.org/wiki/Graphics_processing_unit

4 http://www.scientificcomputing.com/HPC-Future.aspx

5 http://www.nvidia.com/object/fermi_architecture.html

6 http://www.xilinx.com/support/documentation/white_papers/wp375_HPC_Using_FPGAs.pdf

7 http://www.top500.org/lists/2010/06