특집

GPU 병렬 컴퓨팅 기술을 이용한 개인용 수퍼 컴퓨터 현황과 전망

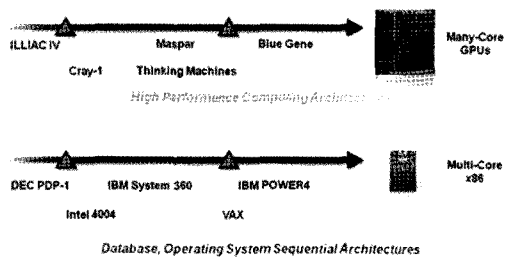
- CUDA기술에 대한 이해 -

이주석·류헌곤 (NVIDIA Korea)

I. 서 론

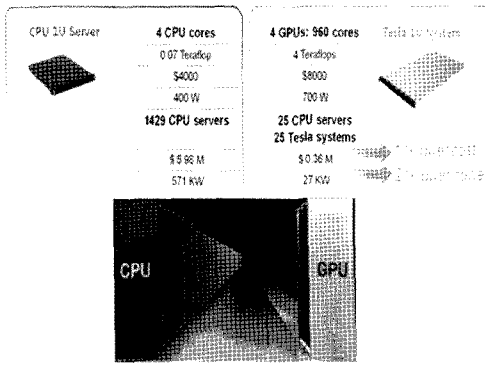
1. 그래픽 카드를 이용한 병렬 컴퓨팅 기술의 발전

컴퓨터의 진화는 architecture상으로 볼 때 크게 serial과 parallel적인 방식 두 가지를 병행하면서 발전을 거듭해 왔다. 전자는 1960년 12월 PDP-1과 1964년 IBM 360 system을 기반으로 시작이 되었으며 1965년 PDP-8에 이르러 최초로 미니 컴퓨터로서의 모습을 드러내게 된다. 1971년 Intel의 4004로 시작된 Micro Processor는 1991년 12월에 CMOS CPU도입과 더불어 본격적인 속도 경쟁에 불을 붙이게 된다. 한편 parallel적인 진화는 1966년 UIUC에 의해 고안된 ILLIAC IV 가 최초의 large-scale병렬 array 컴퓨터로서 등재가 됨으로써 시작이 된다. 그것은 64개의 프로세서를 가지는 SIMD 머신이었으며, 1976년 CRAY I는 최초의 commercial vector machine으로서 향후 1983년 NASA MPP, 1986년 CM-1 그리고 1990년대 MasPar MP-1에 지대한 영향을 미치게 된다. 왜냐하면 그것을 통해 SIMD machine처럼 프로그램이 가능해졌기 때문이다.

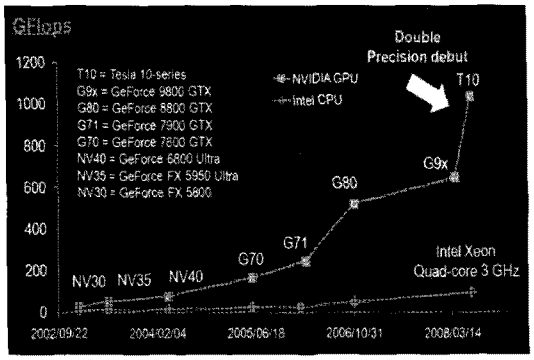


〈그림 1〉 computer architecture의 진화

엔비디아 (www.nvidia.com)는 이러한 진화 라인상에서 2007년 프로그램이 가능한 그래픽 카드 G80을 출시하면서 진화에 큰 획을 긋게 된다. X86 기반의 CPU clock speed가 한계점에 도달하면서 Dual Core로의 전환이 이루어지고 있던 2006년 말, CUDA(Compute Unified Device Architecture)라는 기술을 2006년 10월 웹에서 공개하면서 본격적으로 개인용 Super컴퓨터의 시대를 열어가게 된다. 공개 3개월 만에 CUDA 드라이버 및 SDK를 다운 받은 사람은 6만명이 넘었으며, 일본 동경공업대학은 2009년 Tesla라는 엔비디아 수퍼 컴퓨터 그래픽 카드와 CUDA를 적용하여 아시아에서 가장 빠른 수퍼 컴퓨터 센터를 구축했다. Standford 대학의 Folding@home을 시작



〈그림 2〉 data center with Tesla S1070 for 100TFlops

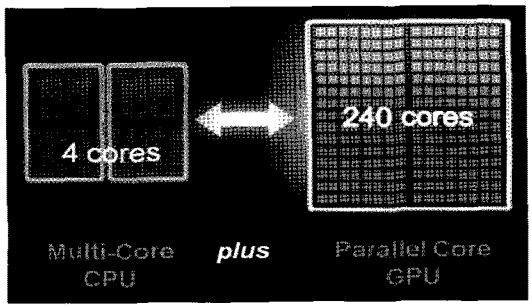


〈그림 3〉 데이터 처리 속도, CPU vs GPU

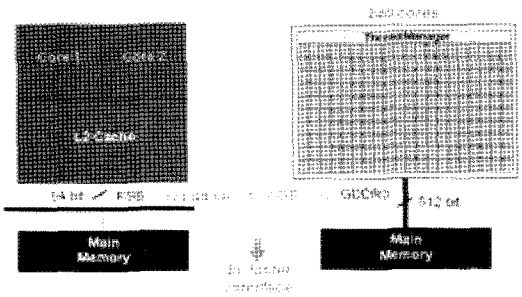
으로 SETI@home, Einstein@home으로 확산이 되고 있는 이 기술은 유럽 주요 대학에서도 적용이 되고 있으며 대만에 있는 대학교 및 중국 칭화대에서도 200 node 이상이 구축이 되면서 새로운 패러다임에 입각한 수퍼 컴퓨터의 속도 경쟁이 시작이 되고 있다.

II. Personal Super Computer 시대의 개막

엔비디아의 David Kirk박사는 병렬 컴퓨팅 솔

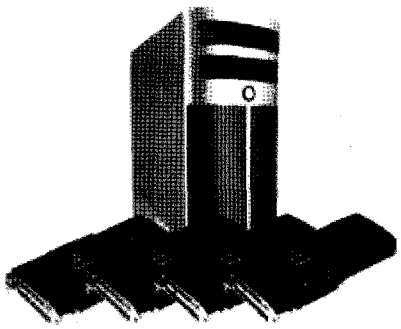


〈그림 4〉 Heterogeneous Computing, 240 cores in C1060



〈그림 5〉 memory interface speed, CPU vs GPU

루션을 소개하는 자리에서 "만약에 당신의 컴퓨터가 20~30% performance 우위를 보인다 해도 비용적 측면을 생각해서 컴퓨터를 교체할지 말지에 대한 고민을 할 것이다. 그런데 100~200배까지 빨라진다면? 이러한 상황에서도 새로운 기술을 도입하지 않는다는 것은 단지 그 사람이 그 사실을 몰랐거나 게으른 것이다"라고 기술했다. Massive한 parallel computing능력을 자랑하는 오늘 날의 GPU는 High Performance Computing (HPC)에 있어서 가장 적합한 솔루션이라고 평가되고 있으며 세상에서 가장 빠르게 확산이 되고 있는 기술 중의 하나이다. 특히 학술, 계산 분야에 있어서의 성과 및 속도는 가속화되고 있으며, Heterogeneous computing을 통해 시스템을 최적화 함으로써 CPU는 operating system, task 처리와 같은 순차적인 업무를 위



〈그림 6〉 Tesla Personal Super Computer with 4 C1060 graphic cards, 4TeraFlops, 16GB memory

주로, GPU는 massive한 data를 처리하게 된다. GPU는 CPU대비 10배 가까운 메모리 인터페이스 speed와 240개의 core에서 동시에 data를 처리 함으로서 최대 200배 이상 까지도 계산 속도를 높일 수 있으며 또한 시스템을 최적화 시킬 수 있다.

<그림 6>에서 보는 것처럼 single precision 1 Tera Flops 성능을 갖는 Tesla C1060을 Desk Top에 있는 PCIeX16에 4대를 장착 함으로서 집에서 또는 연구실 책상에서 4 Tera Flops의 개인용슈퍼컴퓨터를 사용할 수 있게 된 것이다.

III. CUDA에 대한 이해

1. 러닝커브

CUDA 프로그래밍을 접하는 경우 <표 1>에서 보는 바처럼 크게 5가지의 경우가 대부분이다. CUDA는 SIMD의 확장형인 SIMT기반으로 병렬프로그래밍을 해야하는데, 순차적 프로그래밍에 익숙한 개발자나 연구자의 경우, CUDA에 대한 접근이 어렵게 느껴질 수 있다. 하지만, 병렬프로그래밍이나 CG 프로그래밍, 특히, 벡터 프로그래밍을 경험한 연구자는 CUDA를 쉽게 접근할 수 있다.

〈표 1〉 CUDA 개발을 접근하는 경우

Step1 : Serial Programming(Numerical Analysis) Step2 : CUDA programming
Step1 : CG programming(OpenGL, DirectX, Cg) Step2 : CUDA Programming
Step1 : Vector Programming(MMX, SSE) Step2 : CUDA programming
Step1 : CPU Parallel Computing(OpenMP, MPI) Step2 : CUDA Programming
Step1 : CPU Parallel Computing(OpenMP, MPI) Step2 : Heterogeneous HPC(CS, IBM Cell) Step3 : CUDA Programming

또한 다음의 <표 2>는 실제 연구 분야에 CUDA를 적용할 경우 개발기간을 설명하고 있다. 특히, 병렬 프로그래밍 기법에 이미 익숙한 경우 앞에서 설명한 대로, 보다 빠르게 연구에 접목할 수 있다. 특히, CUDA는 C언어의 확장으로 되어 있기 때문에, C언어에 익숙한 개발자는 최소 1주일

〈표 2〉 CUDA 교육 러닝커브

내 용	기 간
병렬 프로그래밍 기법 습득	1달-2달
CUDA GPU H/W 아키텍쳐 이해	1일
CUDA S/W 아키텍쳐 이해	1주일
CUDA 프로그래밍 기법 습득	1-2주
컴파일러 세팅	1일
순차 알고리즘 분석(in house code)	1주-1달
CUDA 병렬화	1-2주일
CUDA 최적화	2달

정도면 CUDA 프로그래밍의 전반적인 기법을 익힐 수 있다.

CUDA가 유용한 연구분야는 이미지 프로세스, 토모그래피, 기계비전, 3D, CG 분야 등 비쥬얼 컴퓨팅 분야 뿐만 아니라, 연산이 많은 행렬연산, 편미분방정식 풀이, 천체 시뮬레이션, 정렬알고리즘, 탐색알고리즘(바이러스, 유전자, 단백질), DB분석계산과 신호 처리 등 다양한 분야에서 응용될 수 있다.

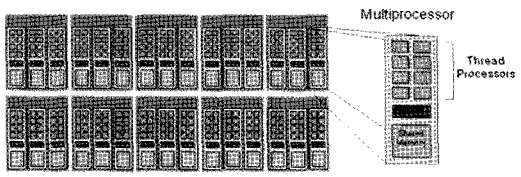
2. H/W 아키텍쳐

NVIDIA의 계산 전용 그래픽카드인 Tesla C1060의 스펙은 다음과 같다.

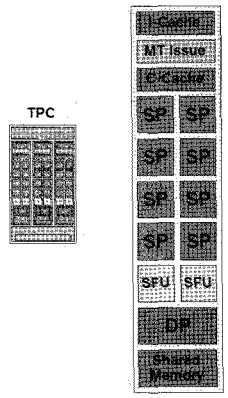
GPU의 아키텍쳐를 살펴보면, NVIDIA 최신 그래픽칩셋인 GT200 계열의 경우 30개의 멀티프로세서가 있고, 각각의 멀티프로세서는 8개의

〈표 3〉 Tesla C1060 사양

Processor	1 x Tesla T10
Number of cores	240
Core Clock	1.296 GHz
Floating Point	933 Gflops SP
	78 GFlops SP
On-board memory	4.0 GB
Memory bandwidth	102 GB/sec peak
Memory I/O	512-bit, 800MHz GDDR3
Form factor	Full ATX
	Dual slot wide
System I/O	PCIe x16 Gen2
Typical power	160 W



〈그림 7〉 NVIDIA GPU 구조



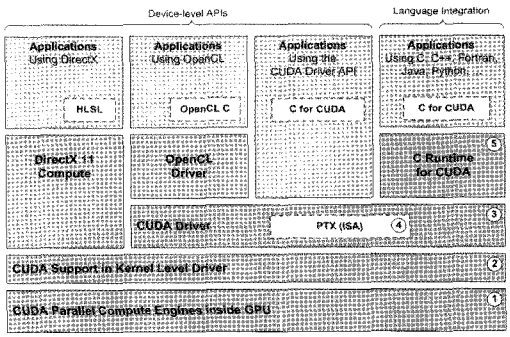
〈그림 8〉 멀티프로세서 구조

쓰레드 프로세서가 장착되어 있다. 따라서 240개의 코어로 작동되고 있다.

멀티프로세서는 8개의 쓰레드 프로세서(SP)와 삼각함수등의 계산에 사용되는 2개의 SFU 유닛과, 1개의 배정도 정밀도 연사을 처리하는 DP 유닛으로 구성되어있다. 특히, 각각의 멀티프로세서는 1만 6천개의 32bit register를 생성된 쓰레드가 분할하여 사용하고 각각 16KB의 shared memory를 사용한다.

3. S/W 아키텍쳐

CUDA는 NVIDIA의 그래픽 하드웨어를 제어

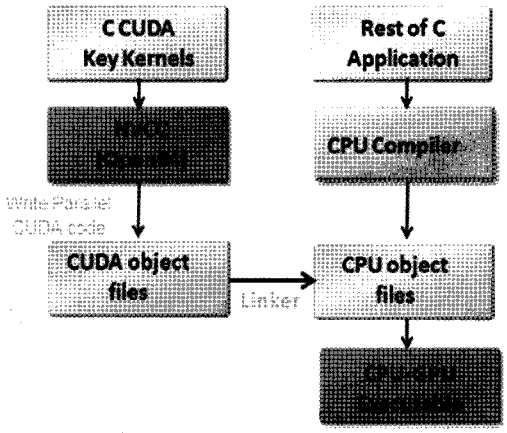


〈그림 9〉 CUDA S/W 아키텍쳐 프레임워크

하는 추상적인 명령어 셋이라고 생각해도 된다. 아래의 그램은 NVIDIA 그래픽카드를 제어하는 CUDA runtime 및 driver API와 CUDA driver 와의 관계 및 OpenCL과 DirectX와의 관계를 나타내고 있다. 특히, OpenCL을 사용하기 위해서 CUDA 드라이버의 설치는 필수적이다.

4. CUDA toolkit 컴파일 환경

CUDA는 그래픽 하드웨어를 제어할 수 있는 명령어셋을 C언어에 추가하여, 추가된 명령어는



〈그림 10〉 CUDA toolkit 컴파일 환경

〈표 4〉 CUDA H/W, S/W 용어간 관계

H/W	S/W
CPU server	Host
GPU chip	Kernel, Grids
Multiprocessor	Blocks
SP (core)	Threads
PCI-e slot	Memcpy()
4GB memory	Global memory
	Local Memory
Register	Register
Shared memory	Shared memory

NVCC 컴파일러를 통해 GPU컨트롤이 가능한 assemble 코드인 PTX인 assemble 코드로 변환하여 사용할 수 있도록 한다. 이를 <그림 10>으로 나타내면 다음과 같다.

CUDA의 S/W는 CUDA H/W 구성과 밀접한 관련을 가지고 있다. <표 4>는 하드웨어와 소프트웨어 간의 용어의 관계를 설명하고 있다.

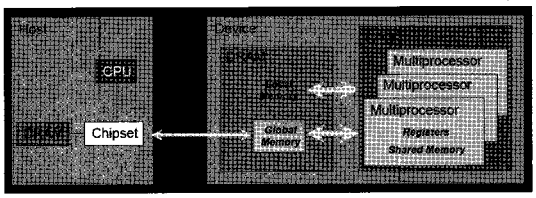
5. 메모리 관리 선행어

CUDA 프로그래밍에서는 항상 CPU메모리의 데이터를 GPU 메모리로 옮겨 작업을 실행시키고, 그 결과를 받아와야 한다. 이때 사용되는 메모리 관리 선행어는 다음과 같다.

〈표 5〉 메모리 관리 선행어

__device__	global 메모리에 변수가 만들어 진다.
__const__	global 메모리에 변수가 만들어 진다. 단 읽기 전용으로cache가 작동된다.
__shared__	shared 메모리에 변수가 만들어진다.

다음 <그림 11>은 GPU에서의 메모리 관리의 흐름도를 나타내고 있다.



〈그림 11〉 메모리 제어

6. 함수 관리 선행어

CUDA함수는 크게 3가지로 나뉜다. device함

〈표 6〉 함수 관리 선행어

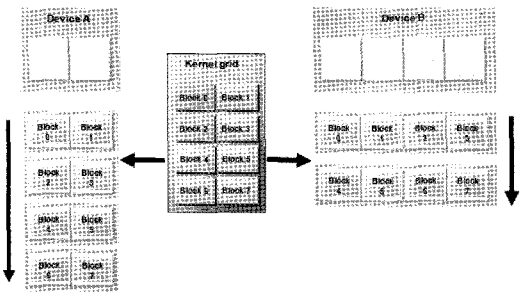
__global__	kernel을 실행시키는 함수를 정의할 때 사용
__device__	device영역에서 GPU 병렬처리 명령어들로 구성된 함수를 정의할 때 사용
__host__	C/C++기본 코드에서 사용되는 함수를 만들 때 사용 (생략해도 됨)

수, global 함수, host함수이다.

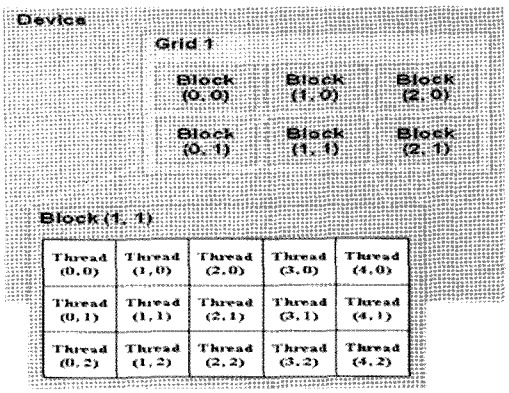
7. Triple Bracket

Triple Bracket <<<A,B>>>의 역할은 __global__ 함수를 실행시킨 후 블록과 쓰레드로 나누어 병렬로 실행시키는 역할을 한다. 이때, block과 thread가 실행되는데, 하나의 멀티프로세서에 최대한 들어갈 수 있는 block의 갯수는 active block이라고 불리우는데, 이는 하나의 쓰레드에서 사용하는 레지스터의 갯수, 전체 블럭에서 사용하는 shared memory의 크기, 블럭당 thread 갯수에 의해 결정된다. 최대 active block은 8이다. 이러한, block은 H/W적으로 자동 스케쥴링이 되고, 멀티프로세서가 많은 GPU로 업그레이드할 경우 코드 수정없이 속도 향상이 보장된다.

Block은 1차원과 2차원의 인덱스를 가질 수 있고, thread는 1차원, 2차원, 3차원의 인덱스를



〈그림 12〉 GPU 칩 업그레이드에 의한 속도 향상



〈그림 13〉 Block과 thread의 인덱스

가질 수 있다. 따라서, 이미지, 행렬 연산등에 Block과 thread의 고유 인덱스를 통해 하나의 픽셀과 원소의 접근을 쉽게 제어할 수 있다.

240개의 코어를 이러한 thread block 기반 위에 병렬프로처리를 통해 다양한 알고리즘을 가속처리하게 된다. 보다 자세한 내용은 NVIDIA CUDA Prgoramming Guide와 CUDA 사이트 http://www.nvidia.com/cuda를 참고하면 된다.

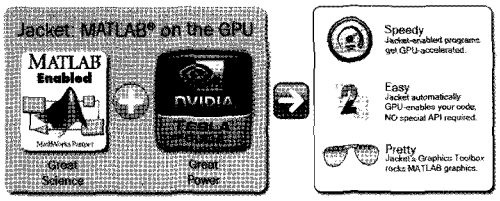
Ⅳ. 연구 성과 및 적용 사례

1. 과학, 연구를 목적으로 개발된 프로그램을 CUDA로 전환 시작

CUDA라는 언어가 C언어와 유사해서 배우기는 쉽지만 실질적으로 병렬화 알고리즘을 이해해서 프로그램을 CUDA로 바꿔주는 것이 그렇게 쉬운 작업만은 아니다. 때문에 이를 통한 연구성과는 직접 프로그램을 만들거나 개발을 해서 사용을 하고 있는 대학가를 중심으로 나타나게 된다. 한편 CUDA전환이 이루어진 알려진 프로그램의 경우 최소 20배에서 최대 137배 가까이

Algorithm	Field	Speedup
2-Electron Repulsion Integral	Quantum Chemistry	130X
Lattice Boltzmann	CFD	123X
Euler Solver	CFD	16X
Gromacs	Molecular Dynamics	137X
Lattice QCD	Physics	30X
Multifrontal Solver	FEA	20X
Nbody	Astrophysics	100X
Simultaneous Iterative Reconstruction Technique	Computed Tomography	32X

〈그림 14〉 CUDA전환을 통한 알고리즘 속도 개선



〈그림 15〉 MATLAB plug in using CUDA, Accelereyes / www.miruware.com제공

빨라지는 것을 확인할 수 있게 된다.

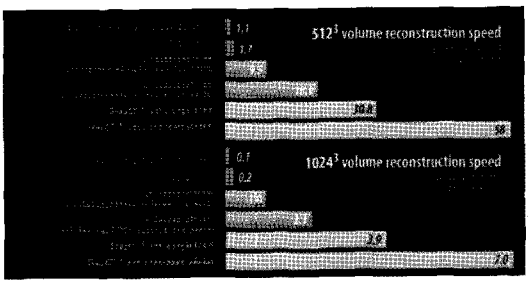
기존에 많이 사용하고 있는 함수 중에서 CUDA 함수가 Plug in 형태로 제공이 되는 경우도 있는데 대표적인 것이 MATLAB, IDL 등이다

2. 연구 학술 분야에서의 성과

몇 가지 대표적인 사례를 소개 드리고자 한다. 자세한 사항은 NVIDIA CUDA zone에서 확인이 가능하다.

가. 의료분야

미국의 의료 장비 업체는 발 빠른 움직임을 보이고 있다. GE healthcare의 CT장비에는 CUDA 기술이 적용이 되었으며 frame rate이 2배 정도 빨라졌다. Techniscan의Ultr-sound, Digisens의 Tomography에도 적용이 되어있다. 일반적으로 simulation을 필요로 하는 의료 분야로의 적용은 무궁무진하다고 볼 수 있다. Cancer



〈그림 16〉 volume reconstruction speed, CPU vs GPU

research(12배), MRI(100배), Brain Circuit simulation(130배) 및 neuron network, Virus simulation이 대표적인 case이다.

나. EDA분야

빠른 처리 속도를 가장 필요로 하는 분야라 할 수 있다. 반도체뿐 만이 아니라 회로 설계를 하고 simulation을 하는 과정은 상당히 많은 시간을 필요로 한다. 10층 기판 이상을 사용하는 회로설계에 대한 simulation은 길게는 한 달 가까이 걸리는 경우도 있는데 이런 곳에 CUDA기술이 적용이 되고 있다. Synopsis TCAD, Sentauras(20배), Agilent Spice simulation ADS(7배), Nas-centric Fast Spice simulation, OmegaSim(40배), Gauda Optical Proximity Correction(200배)가 빨라지는 것이 확인이 되었다. CST xFDT에도 CUDA는 적용이 되어 EMI 해석 tool로서 사용이 되고 있다.

다. Finance

향후 국내 수요가 발생할 수 있는 분야이다. 이미 Wall street 및 해외 주요 은행에서는 CUDA 기술 및 Tesla server가 적용이 되고 있는데 특히 옵션pricing, Risk Analysis, Algorithmic

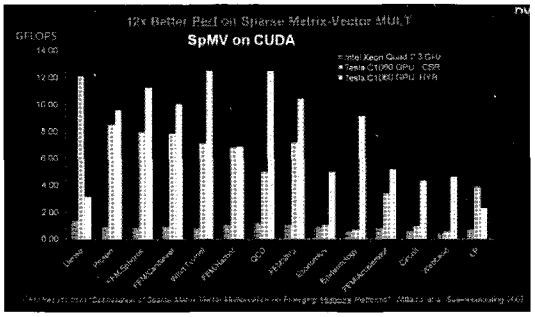
trading 분야로의 적용이 두르러 지고 있다. Monte Carlo simulation을 사용하는데 있어서 병렬컴퓨팅의 기술은 그 위력을 발휘하는데 최소 100배에서 최대 400배까지고 속도 개선이 이루어진다. SciComp (Decatives pricing modeling), Hanweck(Options pricing & risk analysis), Aqumin(3D visualization of market data), Exegy(High-volume Tickers & Risk Analysis), QuantCatalyst(Pricing & Hedging Engine), Oneye (Algorithmic Trading)등에 적용이 되고 있다.

라. 군사, Security 분야

표면으로 나타나고 있지는 않지만 가장 신기술 채택이 빠르게 이루어지고 있으며 성과가 나오고 있는 분야이다. 암호 해독은 50배 이상 속도가 빨라지게 되었으며 위성 사진촬영분석에도 적용이 되어 있다. Security 분야 에서는 홍채인식, 패턴인식, 지문인식, image processing GIS 등에 적용이 되고 있다.

마. Science

자연과학분야, 천체물리 대표적인 N-body simulation을 통해 100배 이상 빠르게 빅뱅현상을 분석할 수 있으며 블랙홀 이론에 대한 검증이 시작이 되고 있다. 일반 desk Top 에서는 확인이 힘든 카오스 이론에서의 Fractal 구조 (Mandelbrot) 현상을 빠르게 확인할 수 있고 유체물리 및 공기역학 관련 현상도 개인용 수퍼 컴퓨터에서 실시간 simulation이 가능하다. CFD 관련, Navier-Stokes, Lattice Boltzman, 3D Euler Solver에 적용 및 개발이 진행이 되고 있다. Quantum



〈그림 17〉 Sparx Metrix-Vector Mult, CPU vs GPU

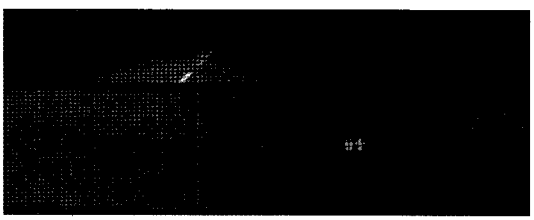
Chemistry 및 Molecular Dynamics관련하여서는 현재 CUDA 가능한 solution으로는 NAMD/VMD(271배), HOOMD, ACE-MD, MD-GPU 등이 있으며 개발이 진행이 되고 있는 solution으로는 LAMMPS, CHARMM, GROMACS, AMBER를 들 수 있다. 기상 및 기후 관련 WRF의 적용은 향후 기상컴퓨터에 지대한 영향을 줄 것이라 생각이 든다. Ocean modeling 및 Tsunami modeling 또한 연구 분야이다. Bio-Infomatics분야에서는 GPU HMMER, MUMmerGPU sequencing이 가능하며 Protein docking 및 Sequence alignment가 가능하다.

바. Oil & Gas, 지질 탐사

미국 및 인도에서는 유전탐사 분야 적용이 활발하게 이루어지고 있으며 지질 탐사 영역으로까지 확대가 되고 있다. Kirchoff and Reverse Time Migration, 3D seismic imaging for prestack depth migration, Spectral decomposition and inversion, 3D data visualization, 3D Seismic processing software, Prestack data processing, GIS 등이 적용 분야이다.

3. 국내 도입 사례

최근 국내 학교 및 연구소에서 CUDA를 적용한 solution 개발 및 연구 사례가 늘고 있다. 고려대학교 물리학과 나노광학lab에서는 FDTD solver CUDA coding이 완료되어 기존 CPU system 대비 200배 빠른 system을 구축하고 있다. 서울대학교 컴퓨터 공학과에서는 CT scan 3D volume reconstruction 및 SOFA기술로의 적용을 위해 개발이 한창이며 또한 유전체 정보 연산에 적용을 하고 있다. 연세대학교 전기전자과에서는 뇌파 simulation, 금융수학과에서는 Monte Carlo simulation, 천문학과 N-body연구가 Tesla 및 CUDA 기반 위에서 진행이 되고 있다. ETRI(edge detection for AR, 100배 향상), 한양대학교(image processing for Fax machine), 한국기상연구소/강릉대학교(WRF with CUDA, ocean modeling), 이화여자대학교 (Molecular Dynamics, 물리엔지), Human RH/nBiz(LCD defect inspection), 한양대학교(이동 통신OFDM), Varo Vision(H.264 encoding), Anycast(주조해석 simulation)이 진행되고 있다. 산업 IT의 융합 분야로 발전이 이루어지고 있으며 자동차에 있어서의 충돌 및 위험 감지 시스템 그리고 야간 주행 시 물체 인식에 대한 실시간 분석이 가능해 지고 있다. 개인용 수퍼 컴퓨터가 있기에 가능한 이야기이다.



〈그림 18〉 CUDA를 이용한 야간 차량 주행 중 물체 식별, 자동차 embedded GPU 적용

V. 현황과 전망

이미 미국에서만 CUDA를 이용해서 솔루션을 개발하고 있는 벤처업체가 60 개 이상이 탄생이 되었다. 또한 전세계적으로 CUDA를 강의하는 대학교도 100 군데가 넘어서고 있다. 그만큼 GPU를 이용한 병렬 컴퓨팅 기술은 획기적이며 많은 사람들에게 그 가능성과 미래를 제시하고 있다고 해도 과언이 아니다. 하지만 막상 병렬 컴퓨팅을 구현하기 위한 resource는 턱없이 부족하며 아직도 많은 application들, 그리고 속도개선이 이루어져야 할 분야가 산재해 있다. 에너지 재생 기술, 개인용 의료기기, 인공지능, 경제 분석, 핸드폰으로의 병렬 컴퓨팅 기술 적용 등은 향후 주목을 받을 만한 분야로 예상이 되고 있다. 평상 시 아무렇지도 않게 지나갔던 일들, 또는 습관에 젖어서 당연하다는 듯이 오랜 시간 결과를 기다렸던 일들이 바로 새로운 기술의 적용 분야가 될 것다. 미래에는 더욱 massive한 데이터를 처리를 해야 하며 빠른 시간 안에 그 결과를 얻어 내야만 하는 상황이 전개될 것이기 때문에 더더욱 병렬 컴퓨팅의 기술, Heterogeneous computing의 필요성은 대두가 될 것이다.

VI. 결 론

상대적으로 S/W infra가 빈약하고 대부분의 application을 해외로부터 도입을 해서 사용하고 있는 한국이 작업의 효율성이 떨어지고 연구결과가 느려지는 것은 당연하다. 다행인 것은 일부 선구자들의 노력에 의해 점차 infra 구축이 이루어지고 있다는 것이다. 이미 NVIDIA CUDA가 가능한 H/W infra는 전세계 1억대 이상 구축이

되어 있다. 이를 기회로 삼아 소프트 인력 양성 및 솔루션이 개발이 되었으면 하는 바램이다.

참고문헌

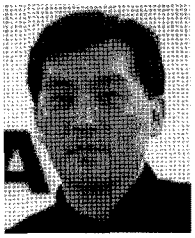
[1] J. Nickolls, I. Buck, K. Skadron, and M. Garland, "CUDA : Scalable parallel programming," ACM Queue, Apr 20008.

[2] J. Michalakes, M. Vachharajani, "GPU Acceleration of Numerical Weather Prediction," Parallel and Distributed Processing, 2008. IPDPS 2008. IEEE International Symposium on.

[3] M. S. Friedrichs, P. Eastman, V. Vaidyanathan, M. Houston, S. LeGrand, A. L. Beberg, D. L. Ensign, C. M. Bruns, V. S. Pande (2009). "Accelerating molecular dynamic simulation on graphics processing units". Journal of Computational Chemistry.

[4] NVIDIA CUDA Programming Guide, NVIDIA Corporation, Mar., 2009, version 2.2. Available : http://www.nvidia.com/CUDA

저자소개



이 주 석

1991년 2월 연세대학교 물리학과 졸업
2001년 10월 ~ 2003년 4월 RFMicro Devices 기술영업
2003년 5월 ~ 2006년 1월 인텔코리아 CISD 영업
2006년 2월 ~ 2009년 4월 엔비디아 코리아 Professional Soluion Group 영업 총괄

주관심 분야 : 수퍼 컴퓨터

류 현 곤

1997년 3월 연세대학교 상경대학 상경계열(경영학/경제학 이중전공) 졸업
2006년 3월 ~ 2008년 2월 연세대학교 이과대학 수학과 대학원 석사과정 졸업
2006년 연세대학교 이과대학 수학과 수리과학연구소 연구소원(RA)
2008년 3월 ~ 2008년 6월 인턴(Barrier 옵션 관련 계산엔진 개발 및 연구)
2008년 3월 ~ 현재 연세대학교 이과대학 수학과 대학원 박사과정 재학 중
2008년 ~ 현재 연세대학교 이과대학 수학과 연세금융 퀀트연구센터 연구원(RA)
2009년 1월 ~ 현재 엔비디아 코리아 SE

주관심 분야 : 병렬처리 알고리즘, 몬테카를로 시뮬레이션, 편미분 방정식