NVIDIA GPU COMPUTING TRENDS IN HW, SW AND SYSMGMT

CARLO NARDONE, Sr. Solution Architect EMEA

ICTP Workshop on Accelerated HPC in Computational Sciences, May-June 2015
THE WORLD LEADER IN VISUAL COMPUTING
AGENDA

1 Intro
2 Future GPU Generation
3 Development Software Trends
4 Tesla Platform System Management
Power for CPU-only Exaflop Supercomputer

= Power for the Bay Area, CA (San Francisco + San Jose)

HPC’S BIGGEST CHALLENGE: POWER
US TO BUILD TWO FLAGSHIP SUPERCOMPUTERS

SUMMIT
150-300 PFLOPS
Peak Performance

SIERRA
> 100 PFLOPS
Peak Performance

IBM POWER9 CPU + NVIDIA Volta GPU
NVLink High Speed Interconnect
>40 TFLOPS per Node, >3,400 Nodes
2017

Major Step Forward on the Path to Exascale
JETSON TK1

THE WORLD’S 1st EMBEDDED SUPERCOMPUTER

Development Platform for Embedded Computer Vision, Robotics, Medical

Tegra K1 SoC
Quad core A15 + Kepler GPU
192 CUDA Enabled cores
326 Gflops @ 5 Watt
$192
ENERGY COST
DP FMA FLOP
ON CPU

Payload Arithmetic 20pJ

Overhead 980pJ
1 Intro
2 Future GPU Generation
3 Development Software Trends
4 Tesla Platform System Management
TESLA PLATFORM DRIVES INNOVATION

Unlocking New Opportunities in the HPC ecosystem

X86, Power and ARM

NVLink

2 GPUs per Node

GPU

x86 CPU

3 GPUs per Node

GPU

GPU

x86 CPU

4 GPUs per Node

GPU

GPU

x86 CPU

NVLINK 20GB/s

PCIe Gen3 x16
PASCAL: NEXT GENERATION TESLA GPU

Peak Performance

>3 TeraFLOPS

Stacked Memory

4x Higher Bandwidth (~1 TB/s)
Larger Capacity (16 GB)

NVLink High-Speed Interconnect

80 GB/sec
POWER CPU & GPU-to-GPU Interconnect

Unified Memory

Single Memory Space
Lower Developer Effort
PASCAL GPU FEATURING NVLINK AND STACKED MEMORY

NVLINK
- GPU high speed interconnect
- 80-200 GB/s

3D Stacked Memory
- 4x Higher Bandwidth (~1 TB/s)
- 3x Larger Capacity
- 4x More Energy Efficient per bit
3D STACKED MEMORY

3D Chip-on-Wafer integration
Many X bandwidth
2.5X capacity
4X energy efficiency
TSV (THROUGH SILICON VIA)

The Enabling Technology for 3D Memory Stack
HBM (HIGH BANDWIDTH MEMORY) STANDARD

1\textsuperscript{st} Gen HBM
- 2Gb per DRAM die
- 1Gbps speed /pin
- 128GB/s Bandwidth
- 4 Hi Stack (1GB)
  - x1024 IO
  - 1.2V VDD
  - KGSD w/ μBump

2\textsuperscript{nd} Gen HBM
- 8Gb per DRAM die
- 2Gbps speed/pin
- 256GBps Bandwidth/Stack
- 4/8 Hi Stack (4GB/8GB)
NVLINK and Unified Memory
BANDWIDTH BOTTLENECKS

- PCI Express: 16GB/sec
- CPU Memory: 60GB/sec
- GPU Memory: 288GB/sec
FUTURE INTERCONNECT: NVLINK

Differential with embedded clock
PCIe programming model (w/ DMA+)
Unified Memory
Cache coherency in Gen 2.0
5 to 12X PCIe
5X MORE BANDWIDTH FOR SCALING
UNIFIED MEMORY

Traditional Developer View of Heterogenous System

- System Memory
- GPU Memory

Developer View With Unified Memory

- Unified Memory

GPU Memory
SIMPLIFIED MEMORY MANAGEMENT CODE

**CPU Code**

```c
void sortfile(FILE *fp, int N) {
    char *data;
    data = (char *)malloc(N);
    fread(data, 1, N, fp);
    qsort(data, N, 1, compare);
    use_data(data);
    free(data);
}
```

**CUDA 6 Code with Unified Memory**

```c
void sortfile(FILE *fp, int N) {
    char *data;
    cudaMallocManaged(&data, N);
    fread(data, 1, N, fp);
    qsort<<<...>>>(data, N, 1, compare);
    cudaDeviceSynchronize();
    use_data(data);
    cudaFree(data);
}
```
UNIFIED MEMORY

Traditional Developer View of Heterogenous System

System Memory

GPU Memory

Developer View With Unified Memory

Unified Memory with NVLink

Share Data Structures at CPU Memory Speeds, not PCIe speeds

Oversubscribe GPU Memory
void sortfile(FILE *fp, int N) {
    char *data;
    data = (char *)malloc(N);
    fread(data, 1, N, fp);
    qsort(data, N, 1, compare);
    use_data(data);
    free(data);
}

void sortfile(FILE *fp, int N) {
    char *data;
    data = (char *)malloc(N);
    fread(data, 1, N, fp);
    qsort<<<...>>>>(data,N,1,compare);
    cudaDeviceSynchronize();
    use_data(data);
    free(data);
}
AGENDA

1. Intro
2. Future GPU Generation
3. Development Software Trends
4. Tesla Platform System Management
VISION: MAINSTREAM PARALLEL PROGRAMMING

Enable more programmers to write parallel software
Give programmers the choice of language
Embrace and evolve key language standards
C++ PARALLEL ALGORITHMS LIBRARY

std::vector<int> vec = ...  

// previous standard sequential loop
std::for_each(vec.begin(), vec.end(), f);

// explicitly sequential loop
std::for_each(std::seq, vec.begin(), vec.end(), f);

// permitting parallel execution
std::for_each(std::par, vec.begin(), vec.end(), f);

• Complete set of parallel primitives: for_each, sort, reduce, scan, etc.

• ISO C++ committee voted unanimously to accept as official tech. specification working draft

N3960 Technical Specification Working Draft:  

Prototype:  
https://github.com/n3554/n3554
INLINE PARALLELISM

Language features enable parallelism in-line with sequential code

Example: Parallel Black-Scholes kernel in (future) standard C++

- Standard parallel algorithms library (Projected C++17)
- Lambda, std::begin/end (C++11)
- Can substitute vendor-specific execution policy (std::par → nvidia::gpu)

```c++
std::for_each(std::par, std::begin(options), std::end(options), [](Option &i) {
    const double d1 = (log((i.S/i.X))+(i.r+i.v*i.v/2)*i.T)/(i.v*sqrtf(i.T));
    const double d2 = d1-i.v*sqrtf(i.T);

    i.call = i.S * CND(d1)-i.X * exp(-i.r*i.T)*CND(d2);
    i.put = i.X * exp(-i.r*i.T) * CND(-d2) - i.S * CND(-d1);
});
```
DEVELOPER PLATFORM WITH OPEN ECOSYSTEM
ACCELERATE APPLICATIONS ACROSS MULTIPLE CPUs

Libraries
- AmgX
- cuDNN
- cuBLAS
- OpenCV
- Thrust

Compiler Directives
- OpenACC

Programming Languages
- C/C++
- Fortran
- Python
- Java
- x86
- ARM
- Power

NVIDIA
DROP-IN ACCELERATION WITH GPU LIBRARIES

Up to 10x speedups out of the box

Automatically scale with multi-GPU libraries

75% of developers use GPU libraries to accelerate their application
AmgX
A simple path to accelerated core solvers, providing up to 10x acceleration in the computationally intense linear solver portion of simulations, and is very well suited for implicit unstructured methods.

cuDNN
NVIDIA cuDNN is a GPU-accelerated library of primitives for deep neural networks, it is designed to be integrated into higher-level machine learning frameworks.

cuFFT
NVIDIA CUDA Fast Fourier Transform Library (cuFFT) provides a simple interface for computing FFTs up to 10x faster, without having to develop your own custom GPU FFT implementation.

cuBLAS-XT
cuBLAS-XT is a set of routines which accelerate Level 3 BLAS (Basic Linear Algebra Subroutine) calls by spreading work across more than one GPU.

NPP
NVIDIA Performance Primitives is a GPU accelerated library with a very large collection of 1000’s of image processing primitives and signal processing primitives.

CHOLMOD
GPU-accelerated CHOLMOD is part of the SuiteSparse linear algebra package by Prof. Tim Davis. SuiteSparse is used extensively throughout industry and academia.

CULA Tools
GPU-accelerated linear algebra library by EM Photonics, that utilizes CUDA to dramatically improve the computation speed of sophisticated mathematics.

MAGMA
IMSL®

IMSL Fortran Numerical Library
Developed by RogueWave, a comprehensive set of mathematical and statistical functions that offloads work to GPUs.

cuSOLVER
A collection of dense and sparse direct solvers which deliver significant acceleration for Computer Vision, CFD, Computational Chemistry, and Linear Optimization applications.

cuSPARSE
NVIDIA CUDA Sparse (cuSPARSE) Matrix library provides a collection of basic linear algebra subroutines used for sparse matrices that delivers over 8x performance boost.

cuBLAS
NVIDIA CUDA BLAS Library (cuBLAS) is a GPU-accelerated version of the complete standard BLAS library that delivers 6x to 17x faster performance than the latest MKL BLAS.

ArrayFire
Comprehensive, open source GPU function library. Includes functions for math, signal and image processing, statistics, and many more. Interfaces for C, C++, Java, R and Fortran.

cuRAND
The CUDA Random Number Generation library performs high quality GPU-accelerated random number generation (RNG) over 8x faster than typical CPU only code.

CUDA Math Library
An industry proven, highly accurate collection of standard mathematical functions, providing high performance on NVIDIA GPUs.

Thrust
A powerful, open source library of parallel algorithms and data structures. Perform GPU-accelerated sort, scan, transform, and reductions with just a few lines of code.
**NVBIo**
A GPU-accelerated C++ framework for High-Throughput Sequence Analysis for both short and long read alignment.

**NVIDIA VIDEO CODEC SDK**
Accelerate video performance with this complete set of NVIDIA video codec tools, which includes the NVENC H.264 hardware encoding API as well as NVCUVID CUDA decoding API.

**HiPLAR**
HiPLAR (High Performance Linear Algebra in R) delivers high performance linear algebra (LA) routines for the R platform for statistical computing using the latest software libraries for heterogeneous architectures.

**OpenCV**
OpenCV is the leading open source library for computer vision, image processing and machine learning, and now features GPU acceleration for real-time operation.

---

**Geometry Performance Primitives (GPP)**
GPP is a computational geometry engine that is optimized for GPU acceleration, and can be used in advanced Graphical Information Systems (GIS), Electronic Design Automation (EDA), computer vision, and motion planning solutions.

**Paralution**
A library for sparse iterative methods with special focus on multi-core and accelerator technology such as GPUs.

**Triton Ocean SDK**
Triton provides real-time visual simulation of the ocean and bodies of water for games, simulation, and training applications.
DEEP LEARNING WITH cuDNN

cuDNN is a library for deep learning primitives

Applications

Frameworks

Caffe
torch
Kaldi
theano
cuDNN

Tesla
TX-1
GPUs
Titan

Caffe (GPU) 14x
Caffe (CPU*) 11x
Baseline Caffe compared to Caffe accelerated by cuDNN on K40
OpenACC
OpenACC is a specification for high-level, compiler directives for expressing parallelism for accelerators

- Aims to be performance portable to a wide range of accelerators
- Multiple Vendors, Multiple Devices, One Specification

The OpenACC specification was first released in November 2011

- Original members: CAPS, Cray, NVIDIA, Portland Group

OpenACC 2.0 was released in June 2013, expanding functionality and improving portability
Program myscience
... serial code ...
!$acc kernels
do k = 1,n1
do i = 1,n2
... parallel code ...
enddo
enddo
!$acc end kernels
... 
End Program myscience

Simple Compiler hints

Compiler Parallelizes code

Works on many-core GPUs & multicore CPUs

Inspired by OpenMP, but more descriptive than prescriptive
OPENACC 2.0 VS OPENMP 4.0
APPLYING OPENACC TO SOURCE CODES

*Exploit* GPU with LESS effort; maintain ONE legacy source code

Examples: REAL-WORLD application tuning using directives

- **ELAN**
  - Computational Electro-Magnetics
  - Goals: optimize w/ less effort, preserve code base
  - Kernels 6.5X to 13X faster than 16-core Xeon
  - Overall speedup 3.2X

- **COSMO**
  - Weather
  - Goal: preserve *physics* code (22% of runtime), augmenting *dynamics* kernels in CUDA
  - Physics speedup 4.2X vs. multi-core Xeon

- **GAMESS CCSD(T)**
  - Molecular Modeling
  - Goals: 3X speedup (2 kernels = 98% of runtime); scale to 1536 nodes
  - Overall speedup 3.1X vs. 8-core Interlagos

Results from EMGS, MeteoSwiss/CSCS, NCSA/Cray/NVIDIA
OPENACC RESOURCES

- www.openacc.org
- PGI OpenACC C/Fortran compiler free trial: www.pgroup.com
- See latest preso at GTC15
  - S5192: Intro to OpenACC
  - S5195: Advanced OpenACC
  - S5196: Comparing OpenACC and OpenMP
CUDA 7 Highlights
CUDA 7

Production Release: Mar 2015 at GTC15

- C++11 language features
  - Increases productivity with lambdas, auto, and more
- New cuSOLVER library
  - Accelerates key LAPACK routines and direct sparse solvers
- Runtime Compilation
  - Advanced feature used to generate highly optimized kernels at runtime

![cuSOLVER Speedup vs CPU on LAPACK Cholesky Factorization Routine](chart)

---

- cuSOLVER 7.0, K40
- MKL 11.0.4, i7-3930K CPU @ 3.20GHz
CUDA RUNTIME COMPILATION
Preview Feature in CUDA 7.0

- Compile CUDA kernel source at run time
  - Compiled kernels can be cached on disk

*Run-time* C++ template specialization
- Before: Compile templates multiple times—once for each datatype that *might* be needed by any user
- Now: only compile for datatypes a specific user needs
- Reduces compile time and compiled code size
- No compromise on performance

Simplifies deployment of DSLs that target CUDA-C instead of NVVM IR
HYPERQ/MPI (MPS): MULTIPLE GPUs PER NODE

MPS Server efficiently overlaps work from multiple ranks to each GPU

$lrank=$OMPI_COMM_WORLD_LOCAL_RANK

case ${lrank} in
  [0]) export CUDA_VISIBLE_DEVICES=0; numactl --cpunodebind=0 ./executable;;
  [1]) export CUDA_VISIBLE_DEVICES=1; numactl --cpunodebind=1 ./executable;;
  [2]) export CUDA_VISIBLE_DEVICES=0; numactl --cpunodebind=0 ./executable;;
  [3]) export CUDA_VISIBLE_DEVICES=1; numactl --cpunodebind=1 ./executable;;
esac
Mark Harris GTC15 preso S5820
NVIDIA TESLA PLATFORM

Data Center Infrastructure

- System Solutions
- Communication
- Infrastructure Management

- GPU Accelerators
  - GPU Boost
- Interconnect
  - GPU Direct
  - NVLink
- System Management
  - NVML

Development

- Programming Languages
  - C/C++
  - Fortran
  - OpenACC
  - python
- Development Tools
  - PGI
  - VAMPiR
- Software Solutions
  - allinea DDT
  - Kitware
  - MATLAB
  - Rogue Wave Software

- Compiler Solutions
  - LLVM
- Profile and Debug
  - CUDA Debugging API
- Libraries
  - cuBLAS

Tesla Accelerated Computing Platform
NVIDIA TESLA ECOSYSTEM

Accelerated Solutions
GPUs are accelerating many applications across numerous industries.
Learn more

Numerical Analysis Tools
Applications with high arithmetic density can enjoy amazing GPU acceleration.
Learn more

GPU-Accelerated Libraries
Adding acceleration to your application can be as easy as calling a library function.
Learn more

Language and APIs
GPU acceleration can be accessed from most popular programming languages.
Learn more

Performance Analysis Tools
Find the best solutions for analyzing your application’s performance profile.
Learn more

Debugging Solutions
Powerful tools can help debug complex parallel applications in intuitive ways.
Learn more

Key Technologies
Learn more about parallel computing technologies and architectures.
Learn more

Cluster Management
Managing your GPU cluster will help achieve maximum performance.
Learn more

Job Scheduling
Scheduling jobs on your GPU Cluster can be simple and intuitive.
Learn more
Scheduling jobs on your GPU Cluster can be simple and intuitive with industry leading solutions now with NVIDIA GPU support.

**IBM Platform LSF**
A powerful workload management platform for demanding, distributed HPC environments. It provides a comprehensive set of intelligent, policy-driven scheduling features that enable you to utilize all of your compute infrastructure resources and ensure optimal application performance.

**PBS Professional**
The flagship product in Altair’s award-winning PBS Works suite, PBS Professional is an EAL3+ security-certified HPC workload management product proven for over 20 years at thousands of global sites. PBS Professional offers powerful, policy-based and topology aware scheduling, million-core scalability, and other capabilities for easily managing any HPC system – from small departmental clusters to the largest, most complex systems on the planet.

**Moab Cluster Suite.**
Collectively Moab and the open-source TORQUE resource manager provide an intelligent workload-driven solution that delivers advanced policy management, scheduling and reporting tools for many of today’s most advanced systems.

**Grid Engine**
An industry-leading distributed resource management (DRM) system used by hundreds of companies worldwide to build large compute cluster infrastructures for processing massive volumes of workload. A highly scalable and reliable DRM system, Grid Engine enables companies to produce higher-quality products, reduce time to market, and streamline and simplify the computing environment.
TORQUE

An open source resource manager providing control over batch jobs and distributed compute nodes. It is a community effort based on the original PBS project and, with more than 1,200 patches, has incorporated significant advances in the areas of scalability, fault tolerance.

SLURM

Slurm is an open-source workload manager designed specifically to satisfy the demanding needs of high performance computing. Slurm is in widespread use at government laboratories, universities and companies world wide. As of the November 2014 Top 500 computer list, Slurm was performing workload management on six of the ten most powerful computers in the world including the GPU giant Piz Daint, utilizing over 5,000 NVIDIA GPUs.
GPU-AWARE CLUSTER MGMT

IBM Platform HPC
A complete high performance computing (HPC) management solution in a single product. It includes a rich set of out-of-the-box features that empowers high performance technical computing users by reducing the complexity of their HPC environment and improving their time-to-solution.

Bright Cluster Manager
A totally integrated, single solution for deploying, testing, provisioning, monitoring and managing GPU clusters. With Bright Cluster Manager, a cluster administrator can easily install and manage multiple clusters simultaneously.

Ganglia
An open-source, scalable, distributed monitoring system for high-performance computing systems such as clusters and Grids. It is carefully engineered to achieve very low per-node overheads and high concurrency. Ganglia is currently in use on thousands of clusters around the world and can scale to handle clusters with several thousand of nodes.

StackIQ Boss for HPC with CUDA Pallet
Build and deploy clusters that leverage NVIDIA GPUs for general purpose computing. By integrating the CUDA Pallet with StackIQ Boss for HPC, users benefit from rapid configuration, and reliable, predictable performance from their cluster thanks to the parallel Avalanche installer, database driven library, and central operator’s console.
MONITORING SYSTEM WITH NVML SUPPORT

Examples: Ganglia, Nagios, Bright Cluster Manager, Platform HPC

Or write your own plugins using NVML
GTC15 preso S5144