

WHAT'S NEW IN CUDA 8

Siddharth Sharma, Oct 2016

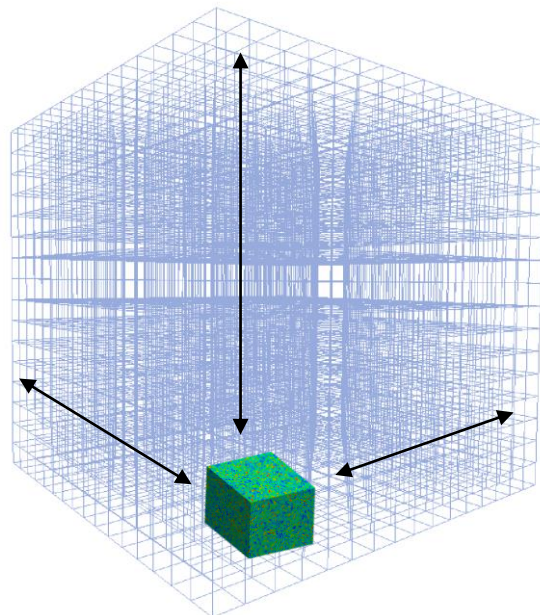


WHAT'S NEW IN CUDA 8

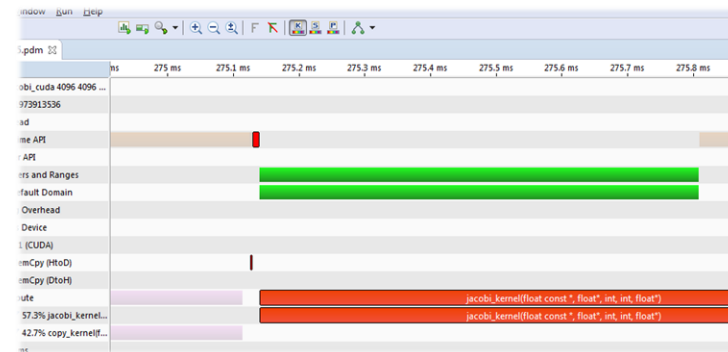
Why Should You Care



Run Computations Faster*



Solve Larger Problems**



Critical Path Analysis

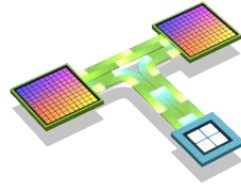
* HOOMD Blue v1.3.3 Lennard-Jones liquid benchmark
• K80 and P100 (PCIe); Base clocks; 4 GPUs per PCIe root complex
• 2x K80 indicates 2-GPU configuration (or 1x K80 board)
• CUDA 8 GA with r361.79 (K80) and r361.93.02 (P100)
• Host System: Intel Xeon Broadwell dual socket 22-core E5-2699 v4@2.2GHz 3.6GHz Turbo with CentOS 7.2 x86-64 and 256GB memory

** HPGMG-FV benchmark
• K80 and P100 (PCIe); Base clocks; 4 GPUs per PCIe root complex
• 2x K80 indicates 2-GPU configuration (or 1x K80 board)
• CUDA 8 GA with r361.79 (K80) and r361.93.02 (P100)
• Host System: Intel Xeon Broadwell dual socket 22-core E5-2699 v4@2.2GHz 3.6GHz Turbo with CentOS 7.2 x86-64 and 256GB memory

WHAT'S NEW IN CUDA 8

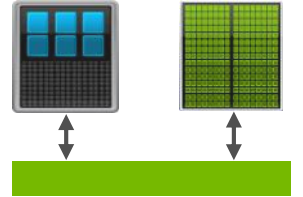
PASCAL ARCHITECTURE

- NVLINK
- HBM2 Stacked Memory
- Page Migration Engine



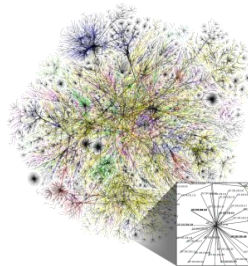
UNIFIED MEMORY

- Demand Paging
- New Tuning APIs
- Data Coherence & Atomics



LIBRARIES

- New nvGRAPH library
- Support for FP16, INT8



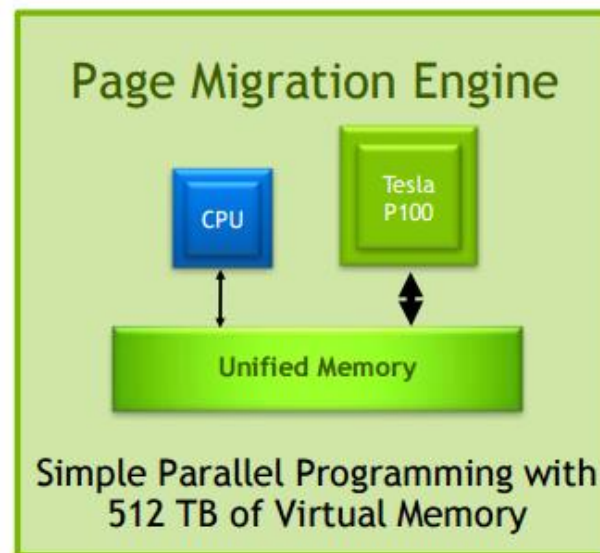
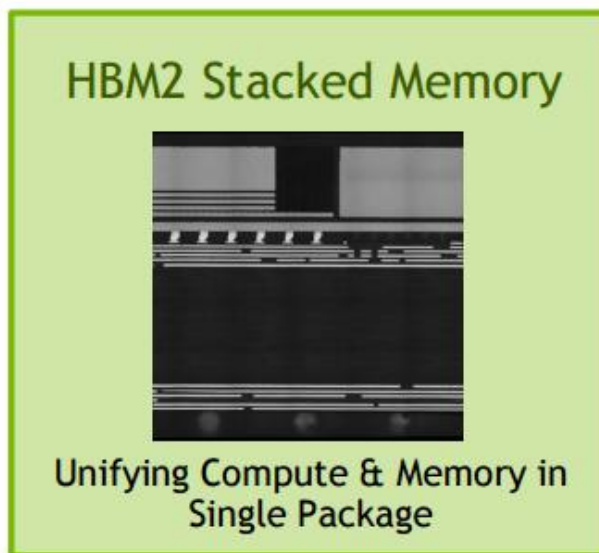
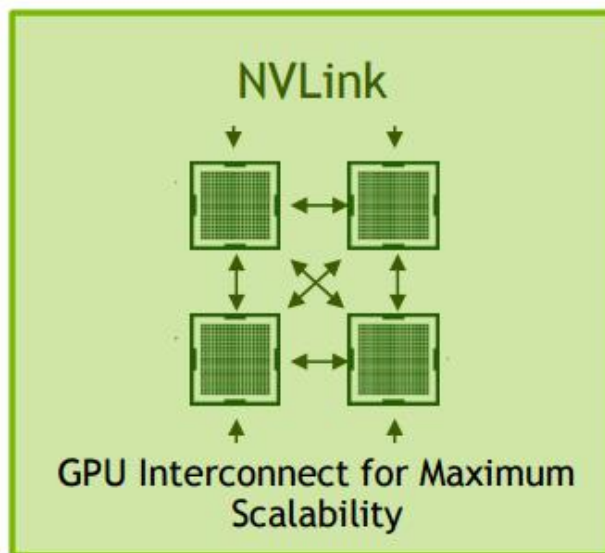
DEVELOPER TOOLS

- Critical Path Analysis
- NVCC Compile Time
- OpenACC Profiling



PASCAL ARCHITECTURE

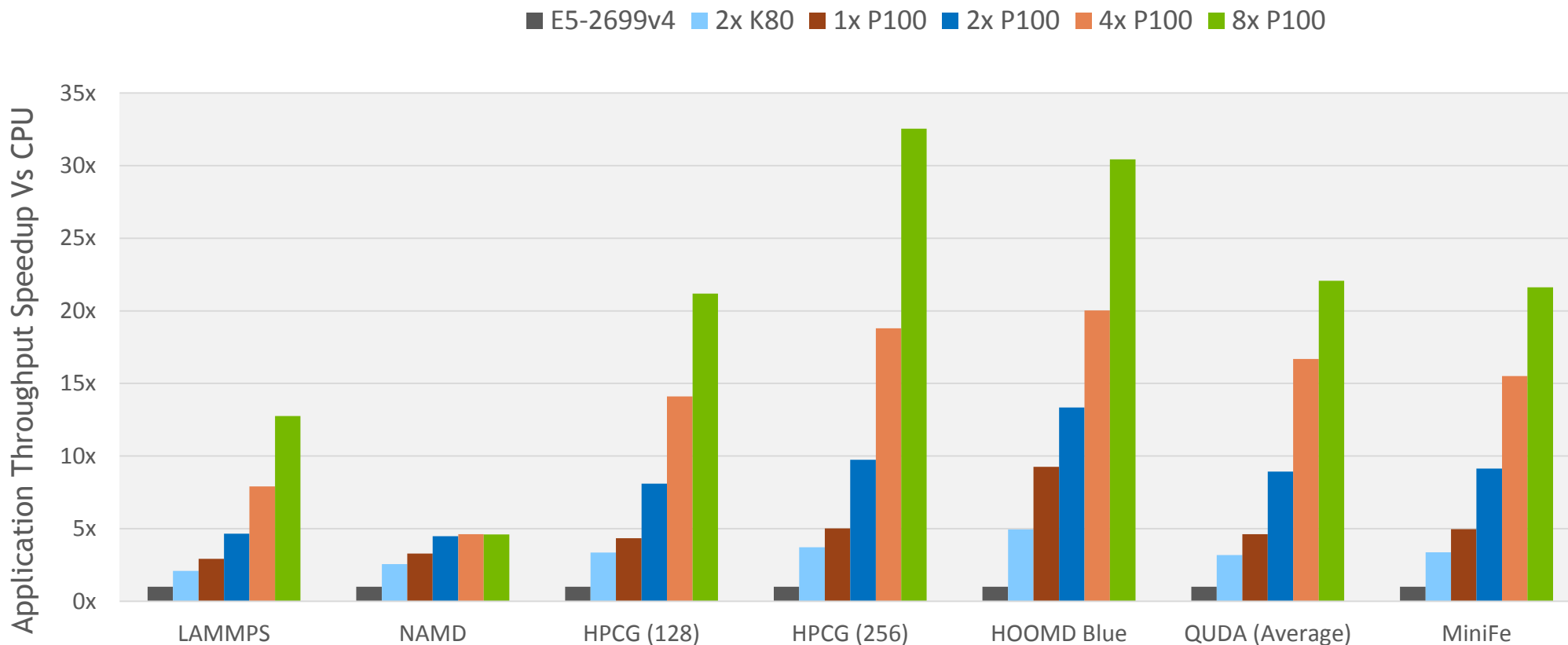
PASCAL ARCHITECTURE



Webinar: "Inside Pascal"

Mark Harris (NVIDIA), Lars Nyland (NVIDIA)
GPU Technical Conference 2016 - ID S6176

CUDA 8 ON P100: >3X FASTER THAN CPUs



- K80 and P100 (PCIe); Base clocks; 4 GPUs per PCIe root complex
- CUDA 8 GA with r361.79 (K80) and r361.93.02 (P100)
- Host System: Intel Xeon Broadwell dual socket 22-core E5-2699 v4@2.2GHz 3.6GHz Turbo with CentOS 7.2 x86-64 and 256GB memory

Performance may vary based on OS and software versions, and motherboard configuration

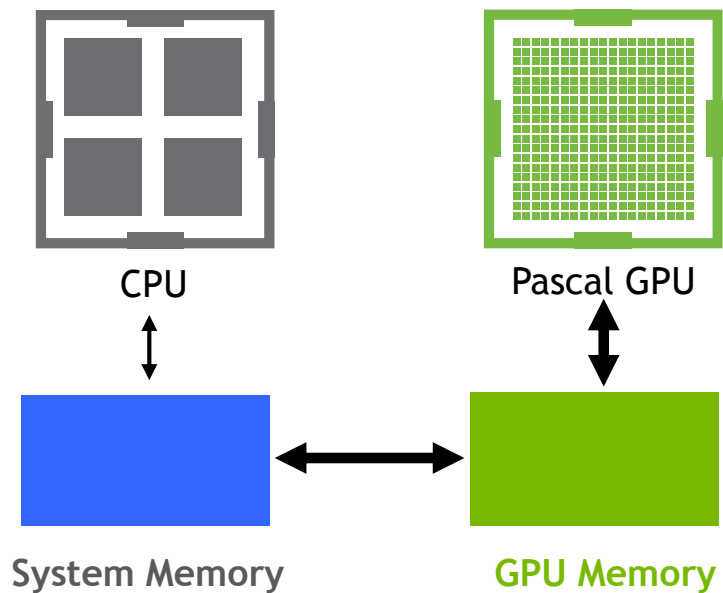
UNIFIED MEMORY

UNIFIED MEMORY

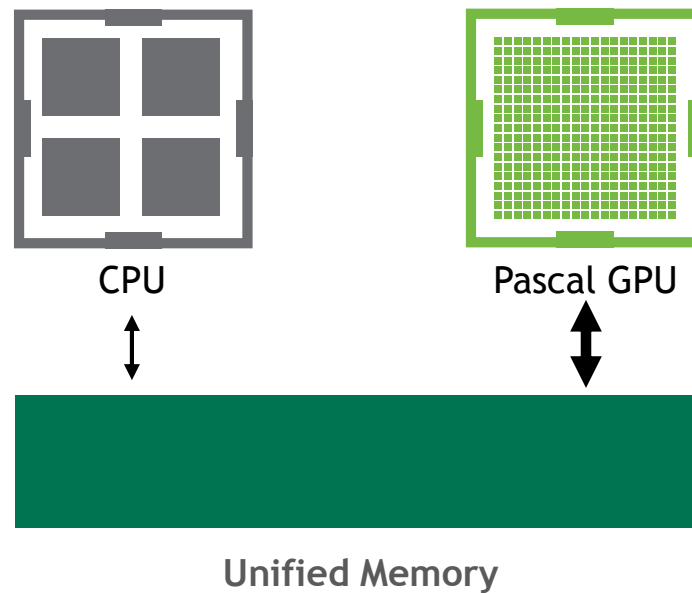
Implicit Memory Management



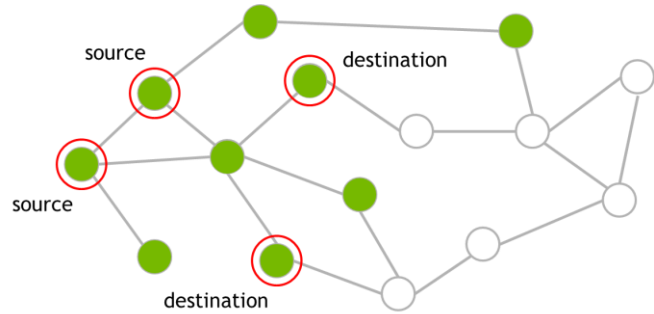
Past Developer View



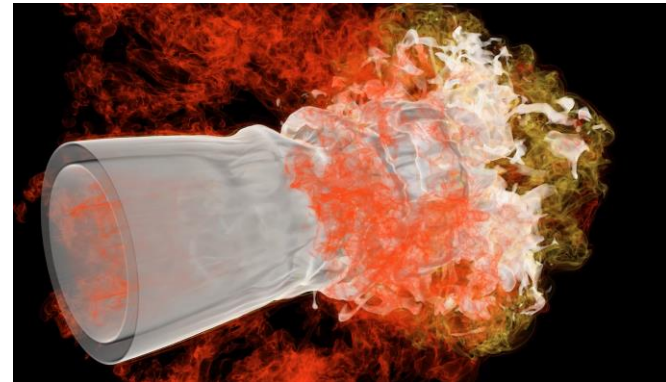
Starting with Kepler and CUDA 6



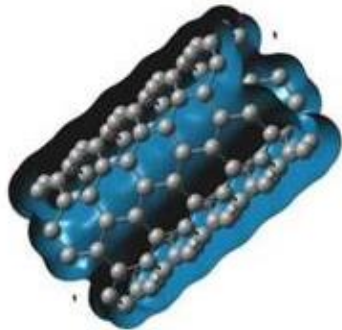
APPLICATIONS: LARGE VARIATIONS IN DATASET SIZES



Graph Analysis
Larger datasets



Combustion
More species & improved accuracy



Quantum Chemistry
Larger systems



Ray-tracing
Larger scenes to render

CUDA 8: PASCAL UNIFIED MEMORY

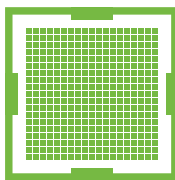
Easier Memory Management, APIs for High Performance



CUDA 8



CPU



Pascal GPU



Unified Memory

Allocate Beyond
GPU Memory Size

ENABLE LARGE
DATA MODELS

Oversubscribe GPU memory
Allocate up to system memory size

SIMPLER
DATA ACCESS

CPU/GPU Data coherence
Unified memory atomic operations

TUNE
UNIFIED MEMORY
PERFORMANCE

APIs for Pre-fetching & Read duplication
Usage hints via `cudaMemAdvise` API

CUDA 8 UNIFIED MEMORY – EXAMPLE

Allocating 4x more than P100 physical memory



```
void foo() {  
  
    // Allocate 64 GB  
    char *data;  
    size_t size = 64*1024*1024*1024;  
    cudaMallocManaged(&data, size);  
}
```

64 GB unified memory allocation on P100 with 16 GB physical memory

Transparent - No API changes

Works on Pascal & future architectures

CUDA 8 UNIFIED MEMORY – EXAMPLE

Accessing data simultaneously by CPU and GPU codes



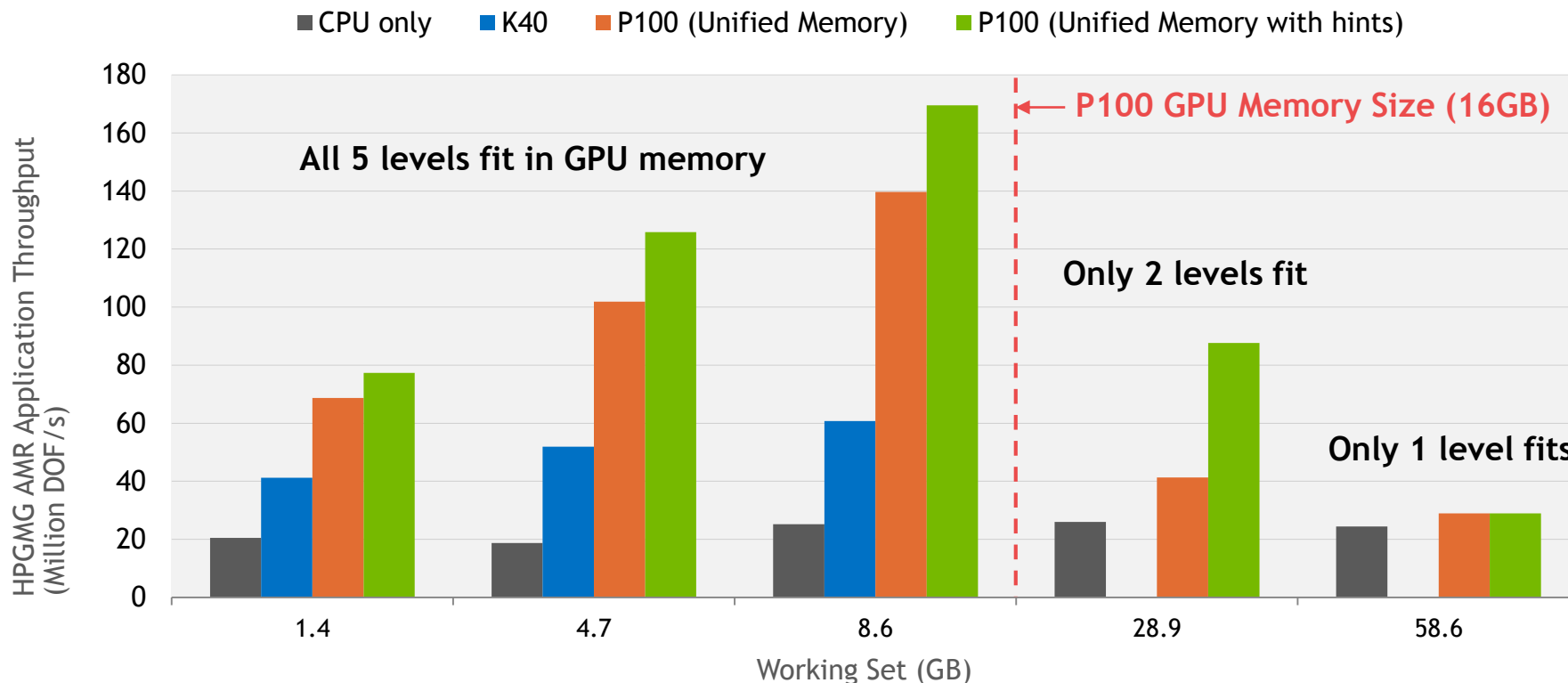
```
__global__ void mykernel(char *data) {  
    data[1] = 'g';  
}
```

```
void foo() {  
    char *data;  
    cudaMallocManaged(&data, 2);  
  
    mykernel<<<...>>>(data);  
    // no synchronize here  
    data[0] = 'c';  
  
    cudaFree(data);  
}
```

Both CPU code and CUDA kernel
accessing 'data' simultaneously

Possible with CUDA 8 unified
memory on Pascal

>3X SPEEDUP WITH UNIFIED MEMORY



- HPGMG AMR on 1xK40, 1xP100 (PCIe) with CUDA 8 (r361)
- CPU measurements with Intel Xeon Haswell dual socket 10-core E5-2650 v3@2.3 GHz 3.0 GHz Turbo, HT on
- Host System: Intel Xeon Haswell dual socket 16-cores E5-2630 v3@2.4GHz 3.2GHz Turbo

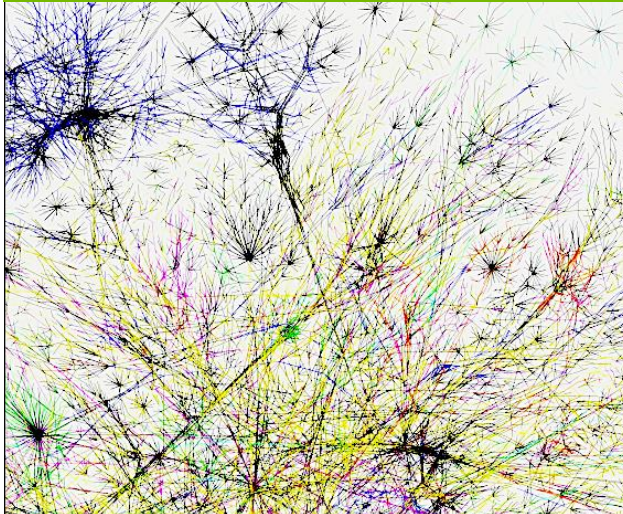
LIBRARIES

GRAPH ANALYTICS

Insight from connections in big data

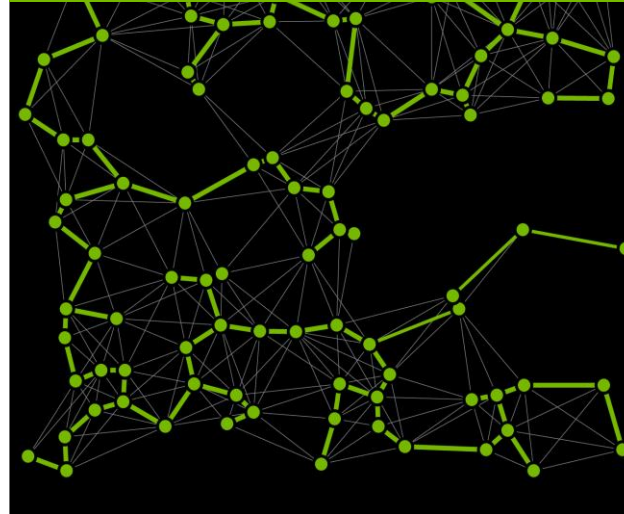


SOCIAL NETWORK ANALYSIS

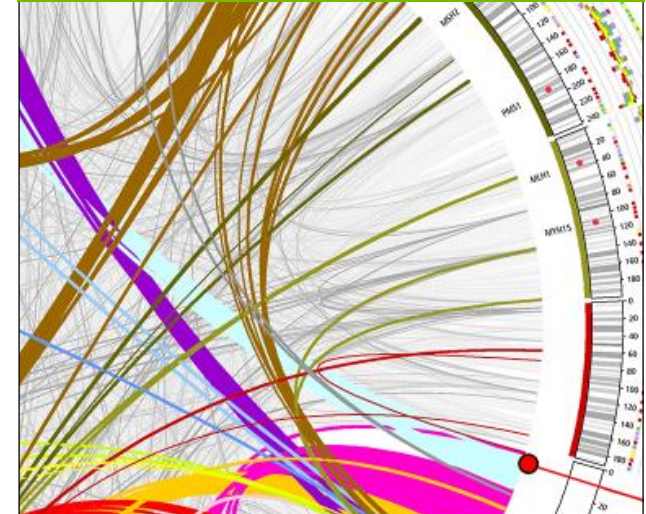


Wikimedia Commons

CYBER SECURITY / NETWORK ANALYTICS



GENOMICS



Circos.ca

... and much more: Parallel Computing, Recommender Systems, Fraud Detection, Voice Recognition, Text Understanding, Search

nvGRAPH

GPU Accelerated Graph Analytics

Parallel Library for Interactive and High Throughput Graph Analytics

Solve graphs with up to 2.5 Billion edges on a single GPU (Tesla M40)

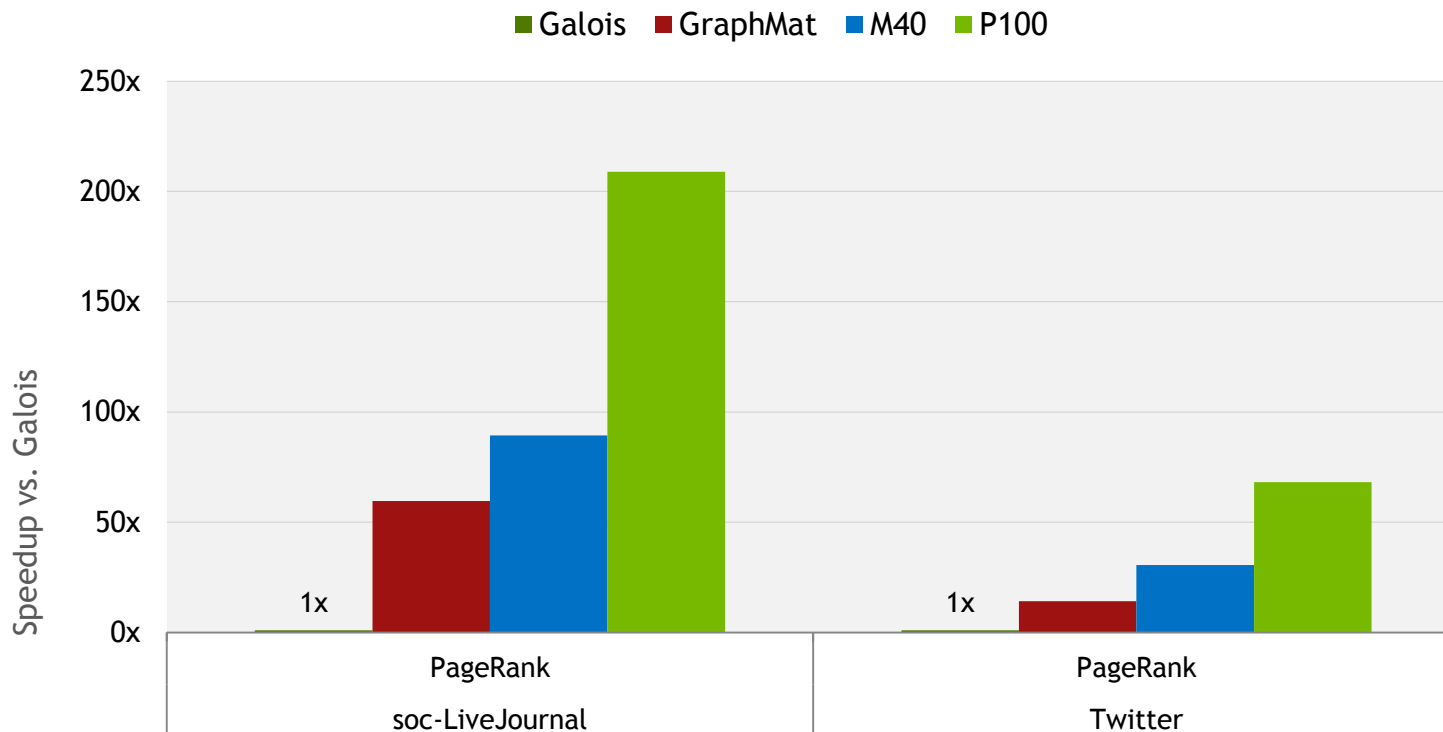
Includes — PageRank, Single Source Shortest Path and Single Source Widest Path algorithms

Semi-ring SPMV operations provides building blocks for graph traversal algorithms



PageRank	Single Source Shortest Path	Single Source Widest Path
Search	Robotic Path Planning	IP Routing
Recommendation Engines	Power Network Planning	Chip Design / EDA
Social Ad Placement	Logistics & Supply Chain Planning	Traffic sensitive routing

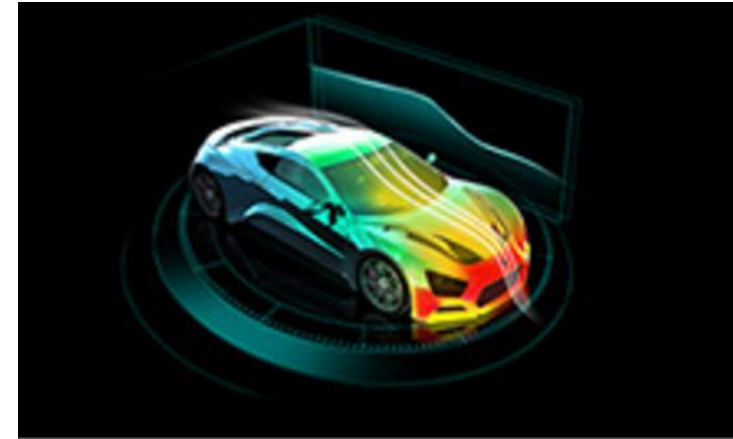
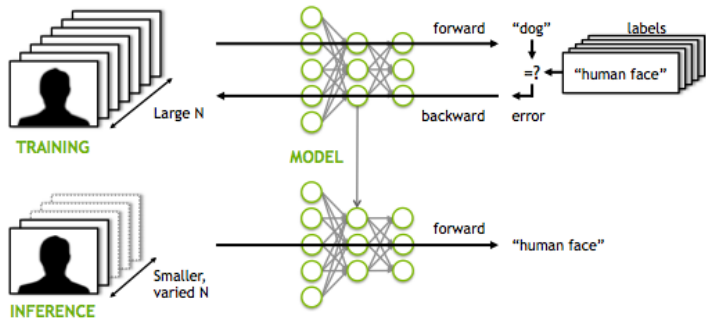
> 200X SPEEDUP ON PAGERANK VS GALOIS



- nvGRAPH on M40 (ECC ON, r352), P100 (r361), Base clocks, input and output data on device
- GraphMat, Galois (v2.3) on Intel Xeon Broadwell dual-socket 22-core/socket E5-2699 v4 @ 2.22GHz, 3.6GHz Turbo
- Comparing Average Time per Iteration (ms) for PageRank
- Host System: Intel Xeon Haswell single-socket 16-core E5-2698 v3 @ 2.3GHz, 3.6GHz Turbo
- CentOS 7.2 x86-64 with 128GB System Memory

Performance may vary based on OS and software versions, and motherboard configuration

HIGHER THROUGHPUT THROUGH LOWER PRECISION COMPUTATION



Deep Learning
cuBLAS: FP16 and INT8 GEMMS

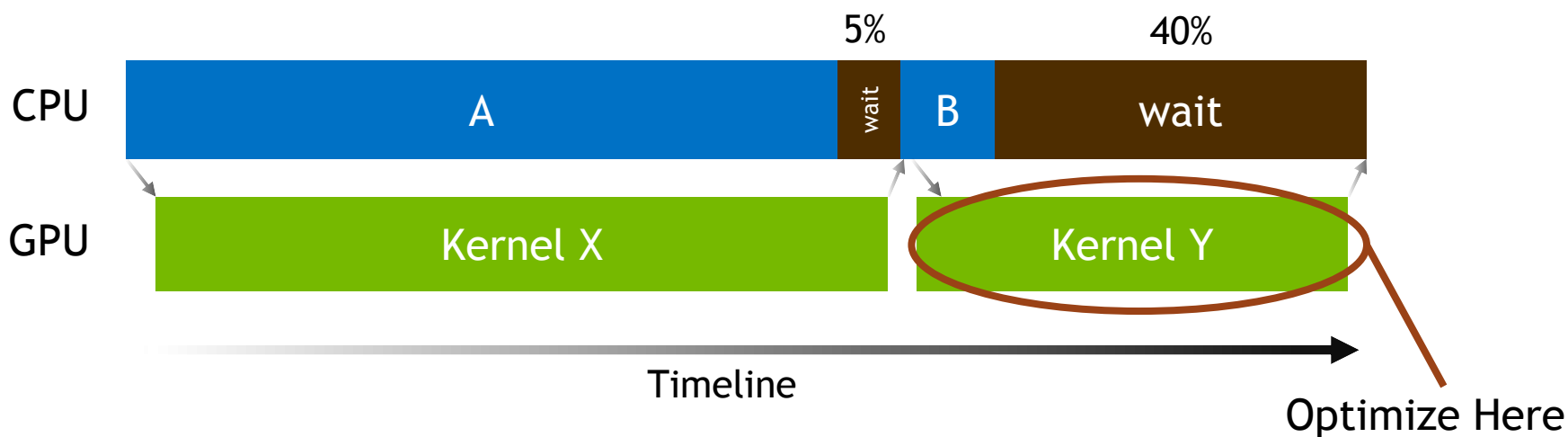
Radio Astronomy
cuFFT: native FP16 operations

Fluid Dynamics
cuSPARSE: FP16 CSR MV

DEVELOPER TOOLS

DEPENDENCY ANALYSIS

Easily find the critical kernel to optimize



The longest running kernel is not always the most critical optimization target

IDENTIFY BOTTLENECKS ON CRITICAL PATH



Visual Profiler and NVPROF

Unguided Analysis

The screenshot shows the Visual Profiler interface with the 'Dependency Analysis' tab selected. The left sidebar contains several analysis categories: 'Data Movement And Concurrency', 'Compute Utilization', 'Kernel Performance', 'Dependency Analysis' (highlighted), and 'NVLink'. The main results pane displays a table of functions on the critical path.

Function Name	Time on Critical Path (%)	Time on Critical Path	Waiting time
cudaMalloc	32.72 %	127.392 ms	0 ns
jacobi_kernel(float const *, float*, int, int, float*)	20.61 %	80.248 ms	0 ns
copy_kernel(float*, float const *, int, int)	17.46 %	68.004 ms	0 ns
<Other>	12.61 %	49.113 ms	0 ns
cudaMemcpy	10.75 %	41.844 ms	20.181 ms
[CUDA memcpy DtoH]	5.18 %	20.181 ms	0 ns
cudaSetupArgument	0.14 %	534.684 µs	0 ns
cudaFree	0.11 %	424.883 µs	0 ns
[CUDA memcpy HtoD]	0.10 %	400.25 µs	0 ns
cuDeviceGetAttribute	0.09 %	336.781 µs	0 ns
cudaGetDeviceProperties	0.08 %	319.677 µs	0 ns
cudaLaunch	0.05 %	192.598 µs	0 ns
cudaConfigureCall	0.05 %	186.452 µs	0 ns
cuDeviceTotalMem_v2	0.05 %	182.833 µs	0 ns
cuDeviceGetName	0.00 %	18.022 µs	0 ns
cudaSetDevice	0.00 %	12.933 µs	0 ns

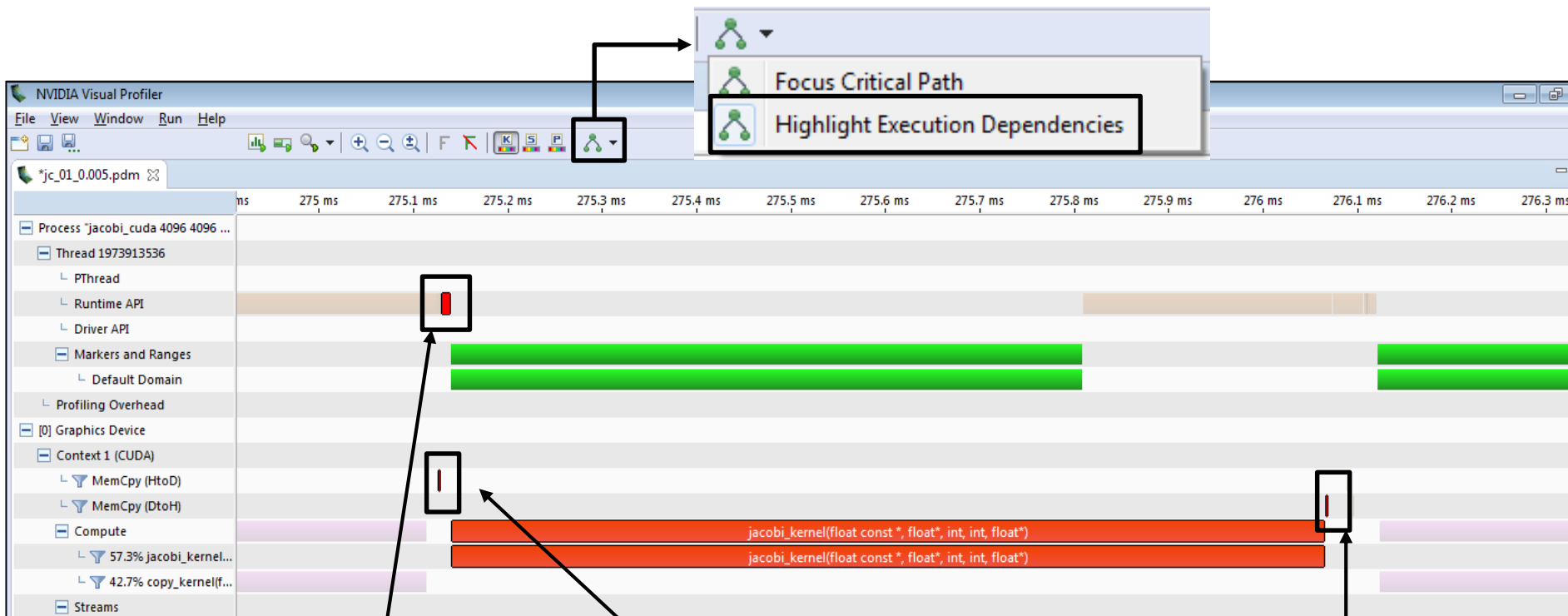
Dependency Analysis

Functions on critical path

IDENTIFY BOTTLENECKS ON CRITICAL PATH



Visual profiler and NVPROF



Launch copy_kernel MemCpy HtoD [sync]
Inbound dependencies

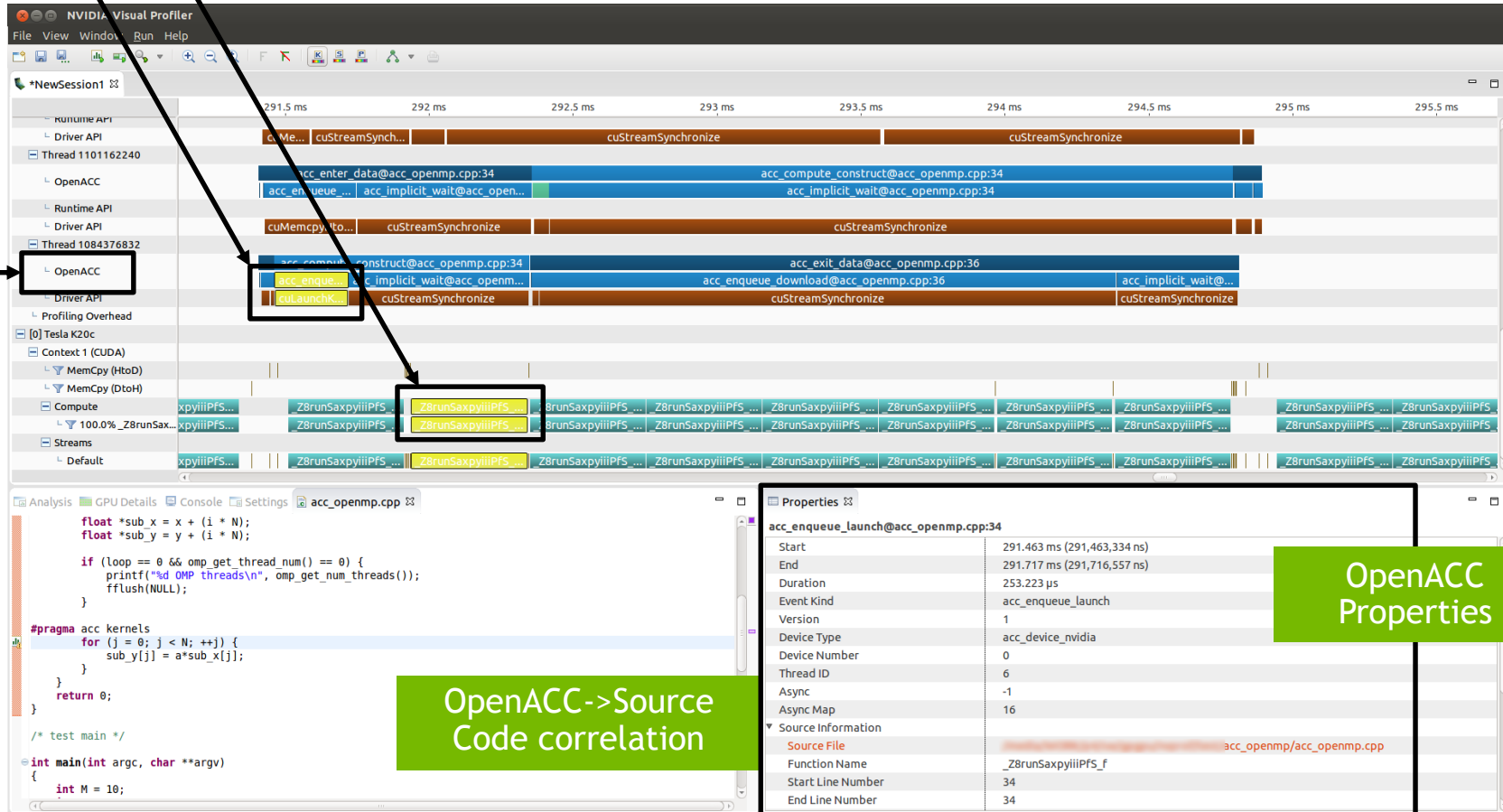
MemCpy DtoH [sync]
Outbound dependencies

OpenACC PROFILING



OpenACC->Driver
API->Compute
correlation

OpenACC
timeline



OpenACC->Source
Code correlation

OpenACC
Properties

PROFILE CPU CODE + GPU CODE IN VISUAL PROFILER



Profile execution times on host function calls

View CPU code function hierarchy

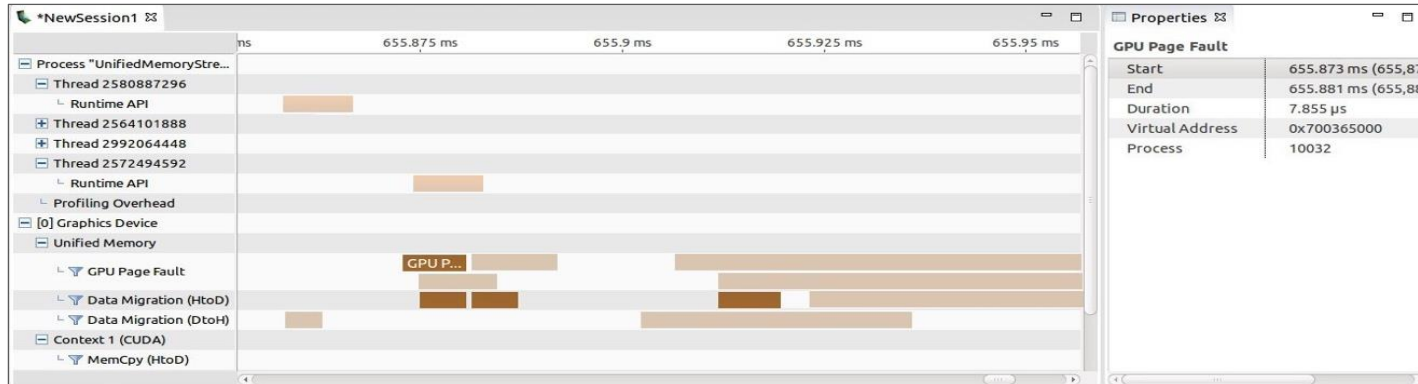
Event	%	Time
TOTAL		
bench_staggeredleapfrog2_	95.833%	689.695 ms
CCTKi_BindingsFortranWrapperBenchADM	95.833%	689.695 ms
CCTK_CallFunction	95.833%	689.695 ms
__open_nocancel	1.389%	9.996 ms
InitialFlat	1.389%	9.996 ms
_c_mcopy8	1.389%	9.996 ms

```
136 FSDX = 4.00/DX
137 FSDY = 4.00/DY
138 !$OMP PARALLEL DO
139 DO 100 J=1,N
140
141
142
143
144
145
146
147 100 CONTINUE
148
149
150 C
151 C PERIODIC CONTINUATION
152 C
153
154 DO 110 J=1,N
155
```

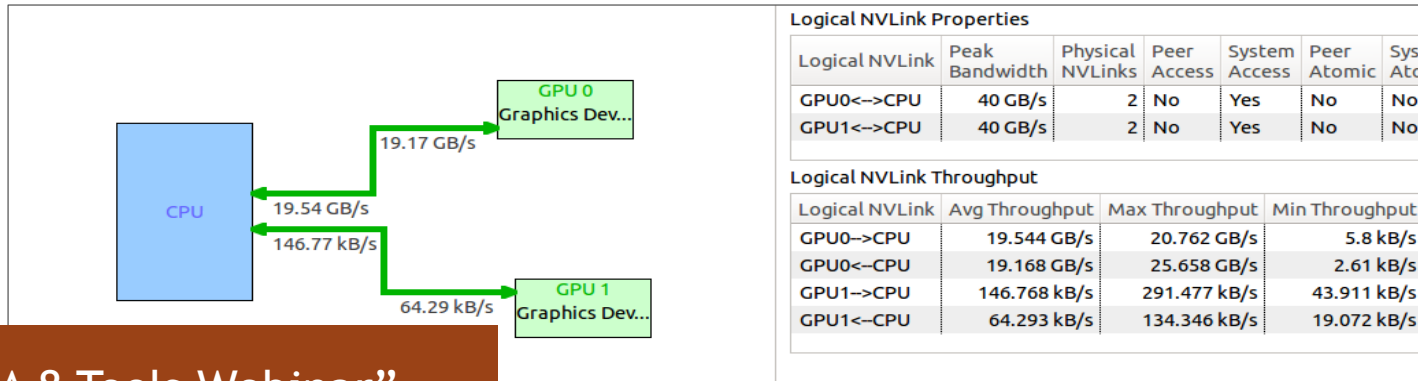
Multiple markers at this line

- Generated 6 prefetch instructions for the loop
- Generated vector sse code for the loop
- Generated 5 alternate versions of the loop
- 2 loop-carried redundant expressions removed with 2 operations and 4 arrays
- Intensity = 1.93

PROFILE UNIFIED MEMORY

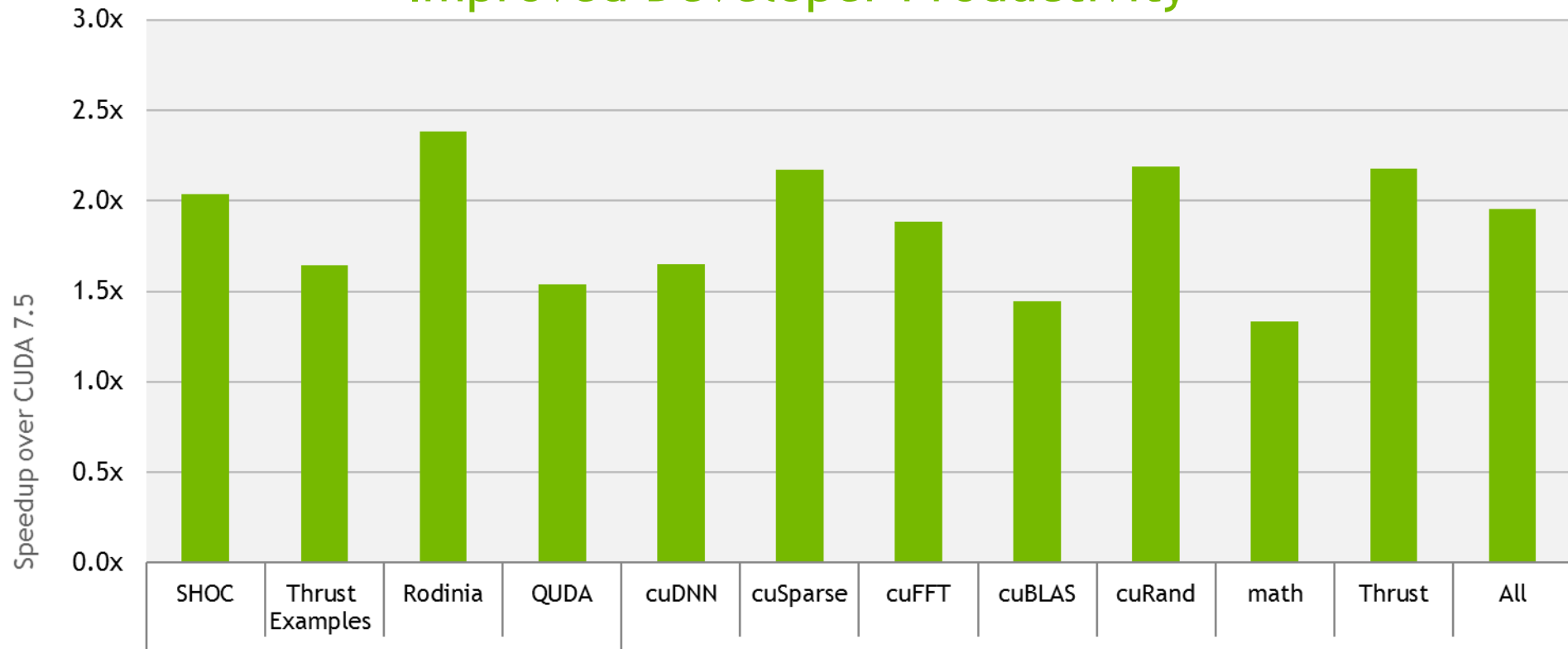


MONITOR NVLINK BANDWIDTH



COMPILE NVCC 2X FASTER

Improved Developer Productivity



NEW PLATFORMS SUPPORTED

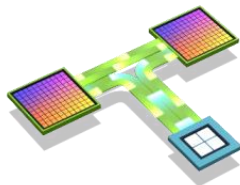


Platform	Operating Systems	Compilers
Windows	Windows Server 2012 R2	Microsoft Visual Studio 2015 Update 3 and VS Community 2015
Linux	Fedora 23, Ubuntu 16.04, SLES 1	PGI C++ 16.1/16.4, Clang 3.7, ICC 16.0
MAC	OS X 10.12	GCC 5.x

WHAT'S NEW IN CUDA 8

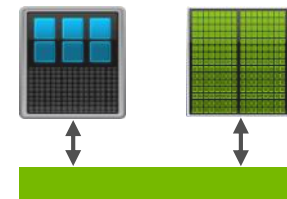
PASCAL ARCHITECTURE

- NVLINK
- HBM2 Stacked Memory
- Page Migration Engine



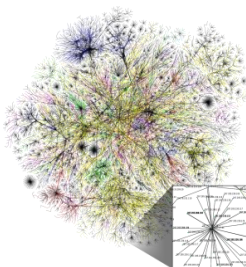
UNIFIED MEMORY

- Demand Paging
- New Tuning APIs
- Data Coherence & Atomics



LIBRARIES

- New nvGRAPH library
- Support for FP16, INT8



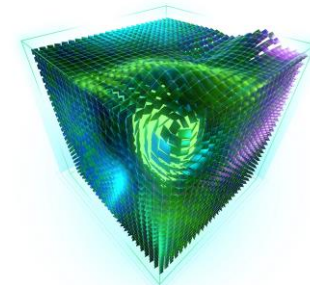
DEVELOPER TOOLS

- Critical Path Analysis
- NVCC Compile Time
- OpenACC Profiling



CUDA 8 - DOWNLOAD TODAY!

Everything You Need to Accelerate Applications



- CUDA Applications in your Industry: www.nvidia.com/object/gpu-applications-domain.htm
- Additional Webinars:
 - Inside PASCAL
 - CUDA 8 Performance Report
 - CUDA 8 Tools
 - CUDA 8 Unified Memory
- CUDA 8 Release Notes: www.docs.nvidia.com/cuda/cuda-toolkit-release-notes/index.html#abstract

THANK YOU

