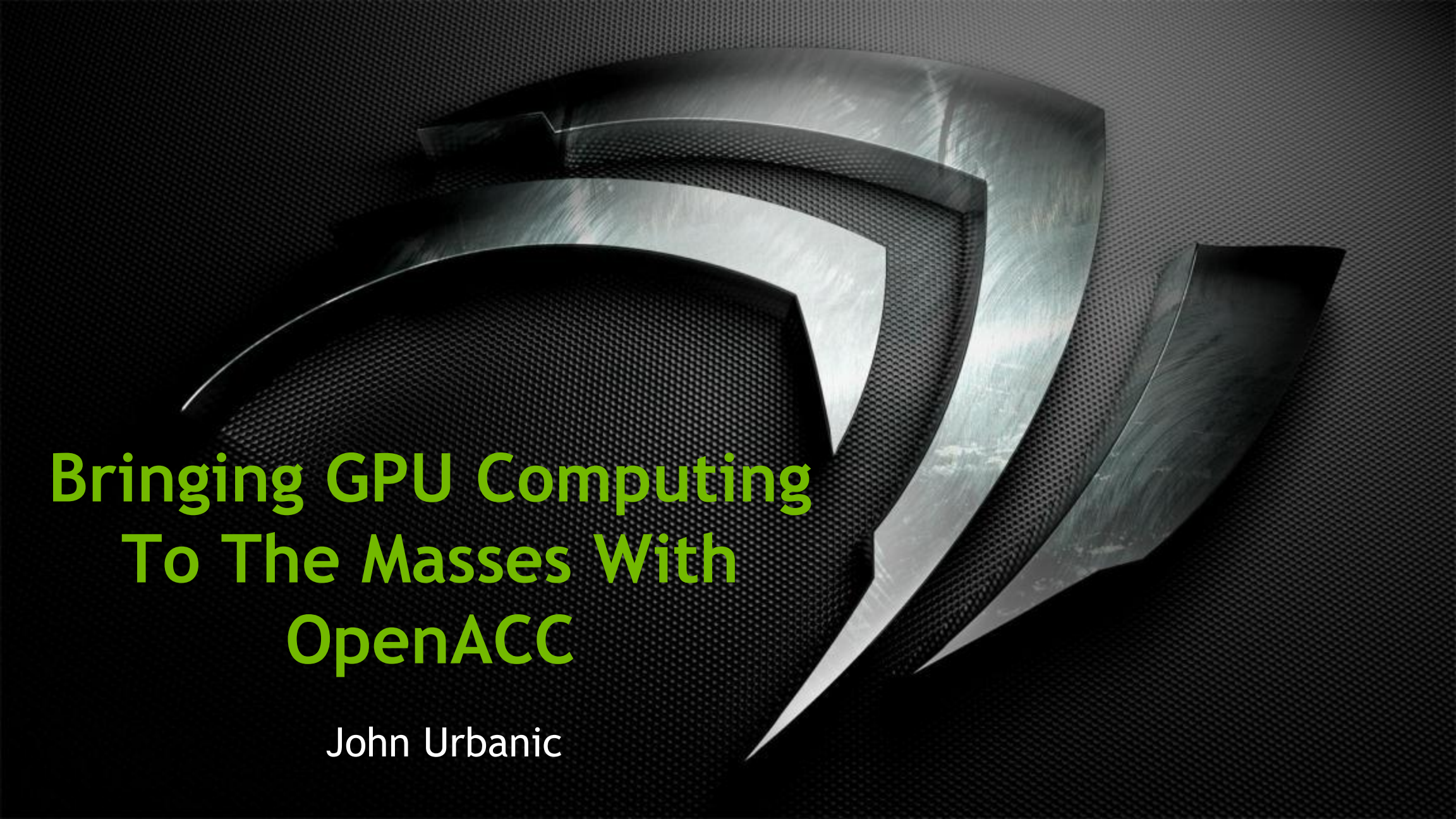
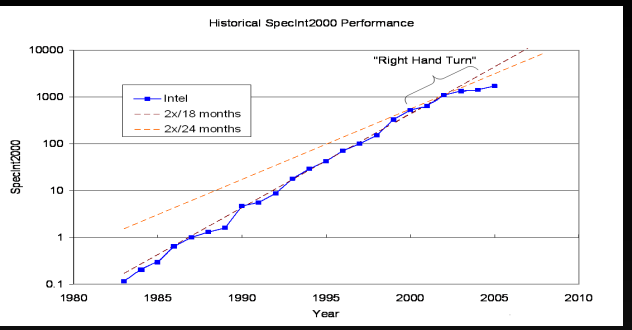
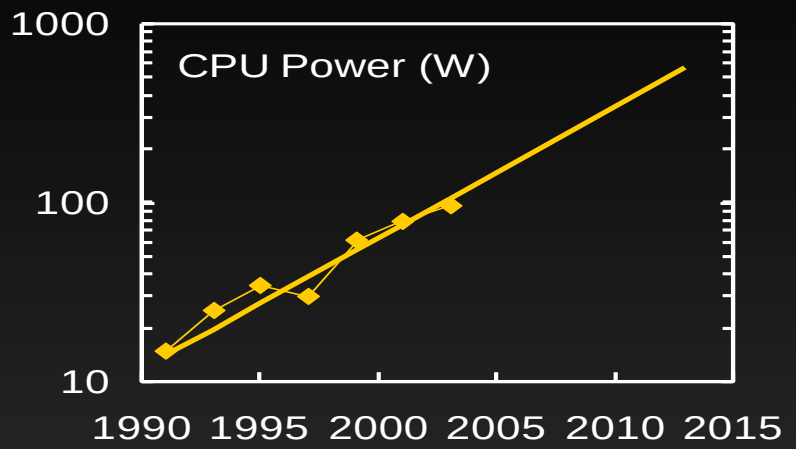


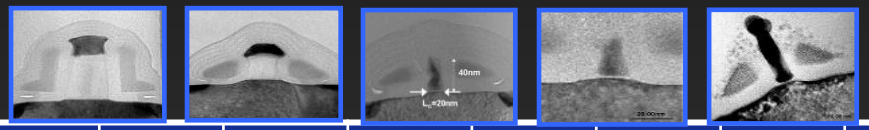
Bringing GPU Computing To The Masses With OpenACC

John Urbanic

What you (may) already know…







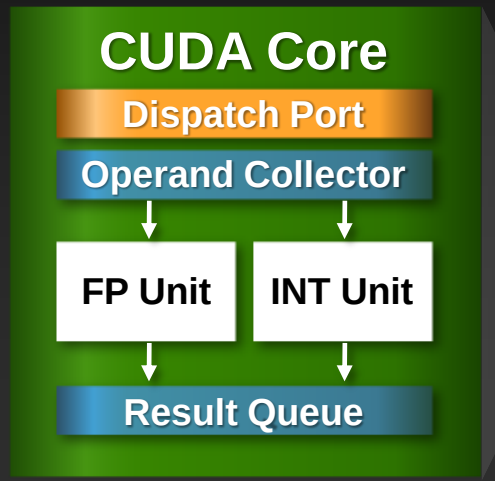
High Volume Manufacturing	2004	2006	2008	2010	2012	2014	2016	2018
Feature Size	90nm	65nm	45nm	32nm	22nm	16nm	11nm	8nm
Integration Capacity (Billions of Transistors)	2	4	8	16	32	64	128	256

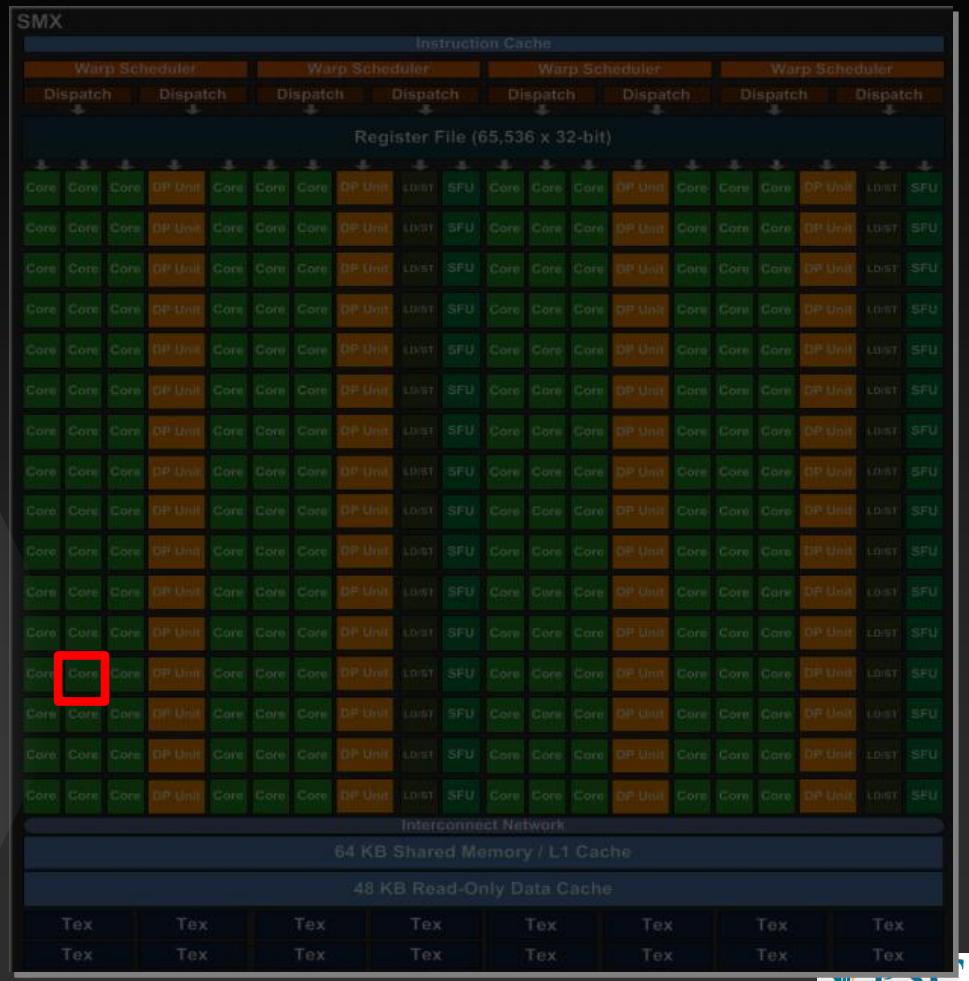
Want a lot of cores?: GPU Architecture (Kepler)

- 192 SP CUDA Cores per SMX
  - 192 fp32 ops/clock
  - 192 int32 ops/clock
- 64 DP CUDA Cores per SMX
  - 64 fp64 ops/clock
- 4 warp schedulers
  - Up to 2048 threads concurrently
- 32 special-function units
- 64KB shared mem + L1 cache
- 48KB Read-Only Data cache
- 64K 32-bit registers

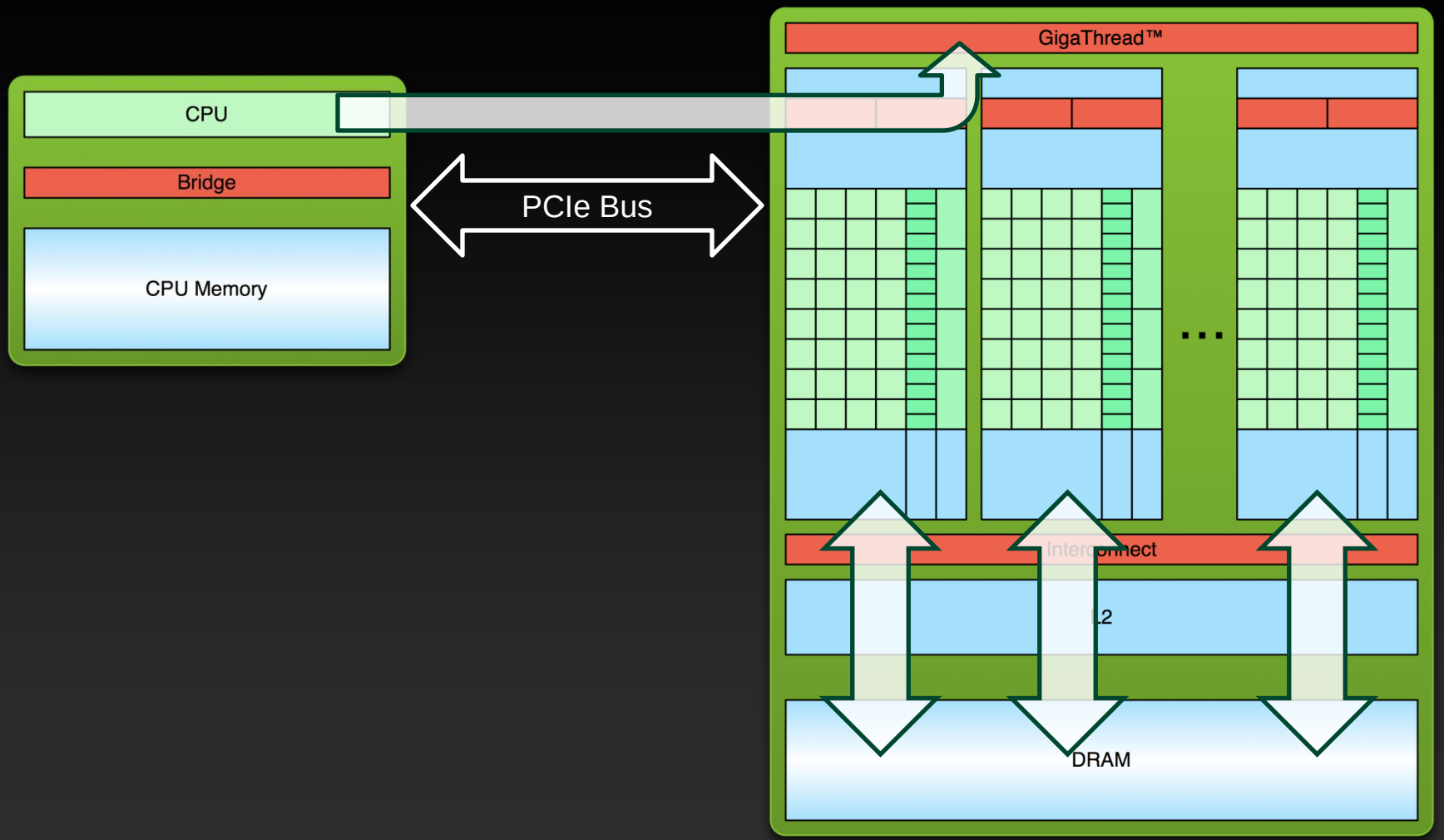
Kepler CUDA Cores

- Floating point & Integer unit
  - IEEE 754-2008 floating-point standard
  - Fused multiply-add (FMA) instruction for both single and double precision
- Logic unit
- Move, compare unit
- Branch unit



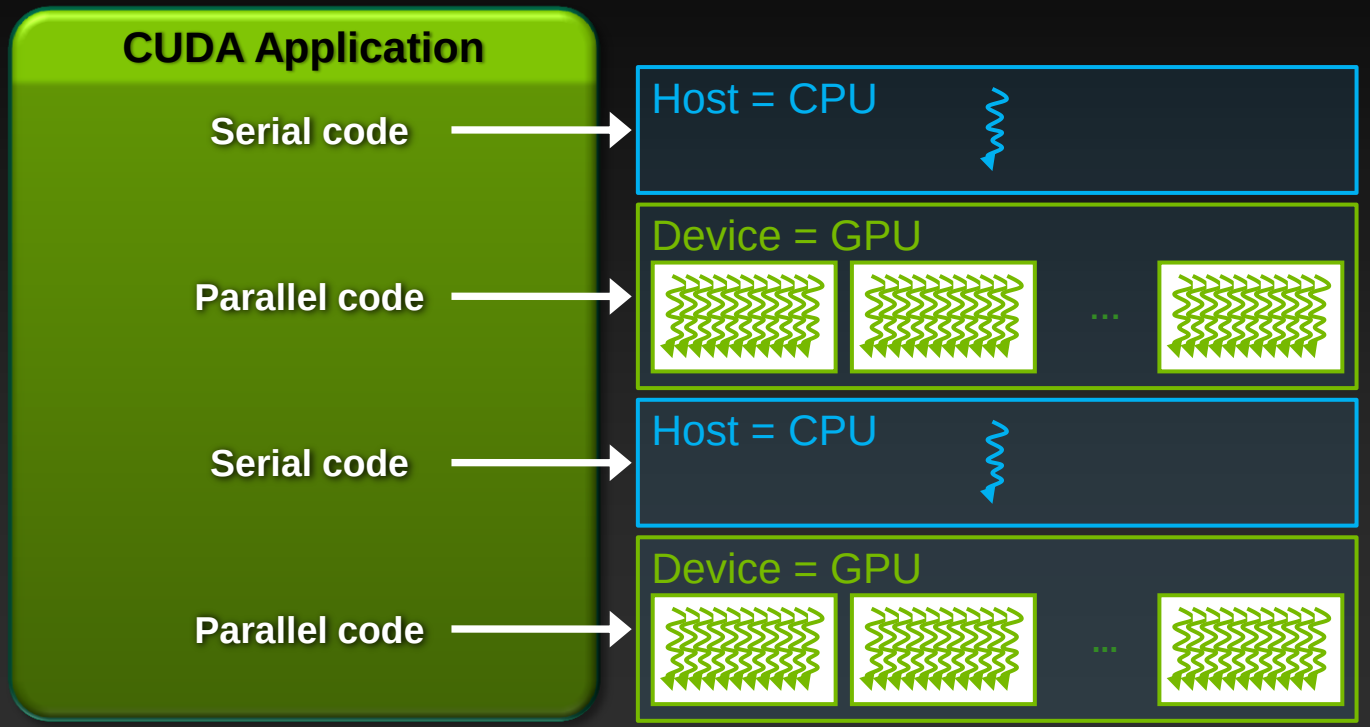


Data Flow Gets Complicated

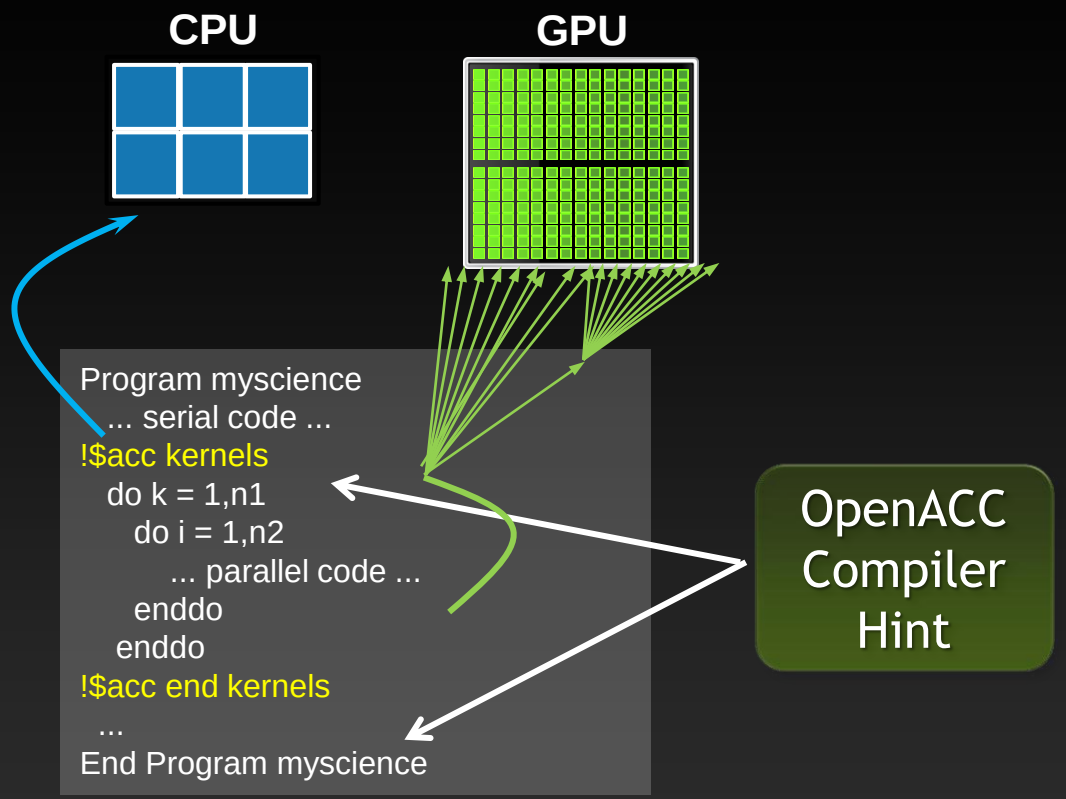


Anatomy of a CUDA Application

- Serial code executes in a Host (CPU) thread
- Parallel code executes in many Device (GPU) threads across multiple processing elements



OpenACC Directives To The Rescue



Simple Compiler hints

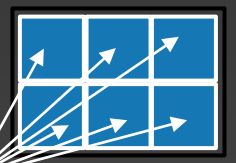
Compiler Parallelizes code

Works on many-core GPUs & multicore CPUs

Familiar to OpenMP Programmers

OpenMP

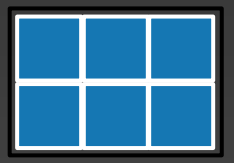
CPU



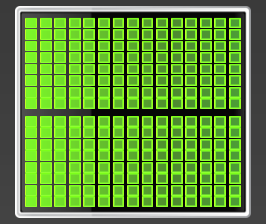
```
main() {
  double pi = 0.0; long i;

  #pragma omp parallel for reduction(+:pi)
  for (i=0; i<N; i++)
  {
    double t = (double)((i+0.05)/N);
    pi += 4.0/(1.0+t*t);
  }

  printf("pi = %f\n", pi/N);
}
```

OpenACC

CPU



GPU



```
main() {
  double pi = 0.0; long i;

  #pragma acc kernels
  for (i=0; i<N; i++)
  {
    double t = (double)((i+0.05)/N);
    pi += 4.0/(1.0+t*t);
  }

printf("pi = %f\n", pi/N);
}
```

Simple example code

```
int main(int argc, char **argv)
{
  int N = 1<<20; // 1 million floats

  if (argc > 1)
    N = atoi(argv[1]);

  float *x = (float*)malloc(N * sizeof(float));
  float *y = (float*)malloc(N * sizeof(float));

  for (int i = 0; i < N; ++i) {
    x[i] = 2.0f;
    y[i] = 1.0f;
  }

  saxpy(N, 3.0f, x, y);

  return 0;
}
```

```
#include <stdlib.h>

void saxpy(int n,
           float a,
           float *x,
           float *restrict y)
{
#pragma acc kernels
for (int i = 0; i < n; ++i)
    y[i] = a * x[i] + y[i];
}
```

Compare: *partial* CUDA C SAXPY Code

Just the subroutine

```
__global__ void saxpy_kernel( float a, float* x, float* y, int n ){
  int i;
  i = blockIdx.x*blockDim.x + threadIdx.x;
  if( i <= n ) x[i] = a*x[i] + y[i];
}

void saxpy( float a, float* x, float* y, int n ){
  float *xd, *yd;
  cudaMalloc( (void**)&xd, n*sizeof(float) );
  cudaMalloc( (void**)&yd, n*sizeof(float) ); cudaMemcpy( xd, x,
   n*sizeof(float),
                    cudaMemcpyHostToDevice );
  cudaMemcpy( yd, y, n*sizeof(float),
                    cudaMemcpyHostToDevice );
  saxpy_kernel<<< (n+31)/32, 32 >>>( a, xd, yd, n );
  cudaMemcpy( x, xd, n*sizeof(float),
                    cudaMemcpyDeviceToHost );
  cudaFree( xd ); cudaFree( yd );
}
```

OpenACC Specification and Website

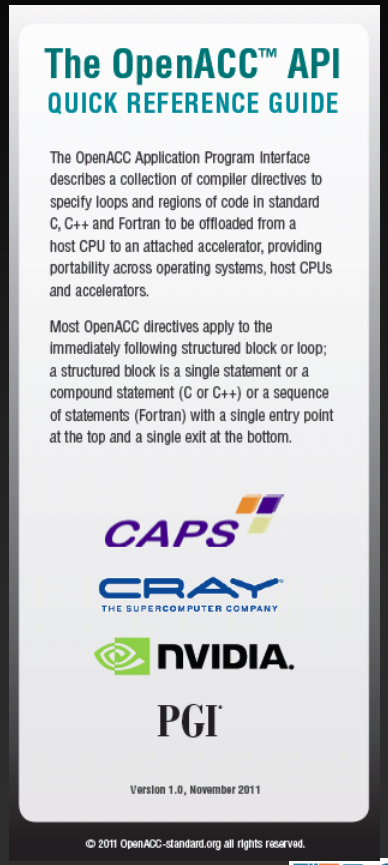
- Full OpenACC 1.0 Specification available online

  http://www.openacc-standard.org

- Quick reference card also available

- Implementations available now from PGI, Cray, and CAPS

- Will be rolled into OpenMP 4.0



Small Effort. Real Impact.

Large Oil Company

3x in 7 days

Solving billions of equations iteratively for oil production at world's largest petroleum reservoirs

UNIVERSITY of HOUSTON
CULLEN COLLEGE of ENGINEERING

Univ. of Houston

Prof. M.A. Kayali

20x in 2 days

Studying magnetic systems for innovations in magnetic storage media and memory, field sensors, and biomagnetism

THE UNIVERSITY OF MELBOURNE

Uni. Of Melbourne

Prof. Kerry Black

65x in 2 days

Better understand complex reasons by lifecycles of snapper fish in Port Phillip Bay

УГАТУ

Ufa State Aviation

Prof. Arthur Yuldashev

7x in 4 Weeks

Generating stochastic geological models of oilfield reservoirs with borehole data

COMPUTING FOR SCIENCE fs

GAMESS-UK

Dr. Wilkinson, Prof. Naidoo

10x

Used for various fields such as investigating biofuel production and molecular sensors.

* Using the PGI Accelerator Compiler

http://www.nvidia.com/object/gpu-directives-successes.html for many more.

PSC

How did PSC get involved?

Right people at the right place:

We've been working with PGI on their OpenACC predecessor for a few years and…

We've been dealing with NVIDIA as a potential collaborator for a while and…

We have a large user base that has been considering GPGPU computing, so…

We said "Let's do an OpenACC Workshop."

Results?

We had a very successful workshop.

NVIDIA was pleased. They got some great PR and even raffled away some nice hardware to those folks providing official Application Success Stories.

There was more demand, so we decided to try it again at XSEDE ‘12.

XSEDE

Extreme Science and Engineering Discovery Environment

Also very successful.

Meanwhile…

The Virtual School of Computational Science and Engineering (VSCSE)
asked PSC to be a satellite site for their summer programs, which we did.

Funding and support for the Virtual School are provided by the
- Great Lakes Consortium for Petascale Computation (GLCPC)
- National Science Foundation (NSF)
- State of Illinois,
- Committee on Institutional Cooperation (CIC),
- Internet2 Commons.

While their content and delivery was fine, their successful use of 2 way HD teleconferencing in conjunction with other channels was really pioneering.

Maybe we could address this pent-up demand for OpenACC training with this approach?
But, bigger…

Keeneland – Full Scale System

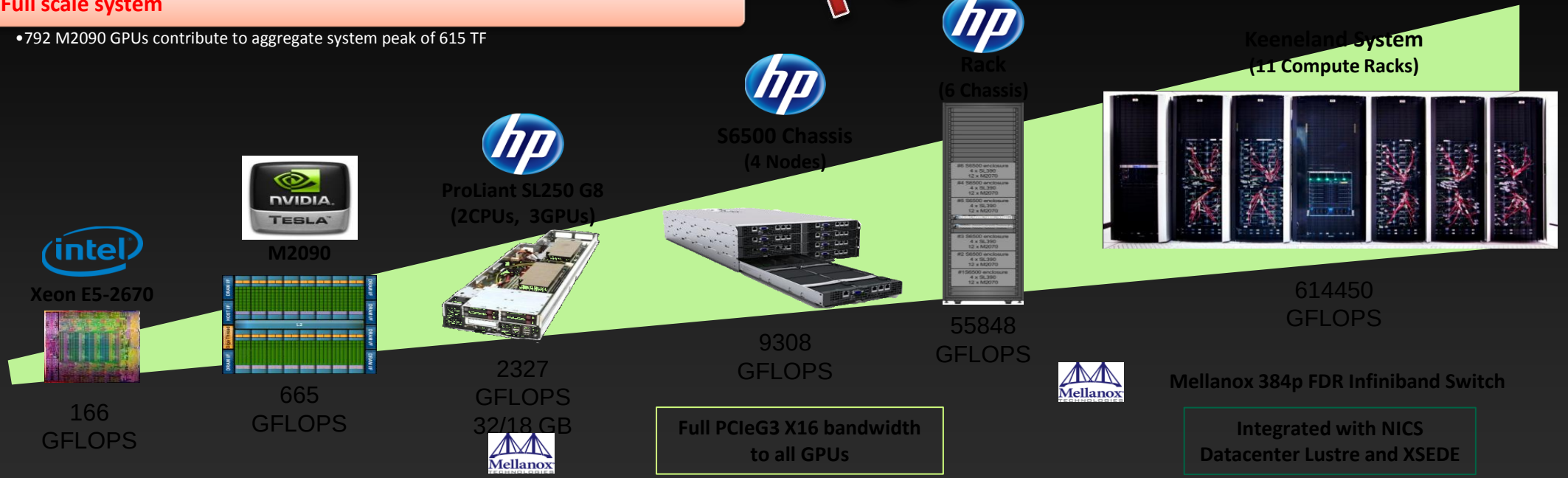
Initial Delivery system installed in Oct 2010

- 201 TFLOPS in 7 racks (90 sq ft incl service area)
- 902 MFLOPS per watt on HPL (#12 on Green500)
- Upgraded April 2012 to 255 TFLOPS
- Over 200 users, 100 projects using KID

Full scale system

- 792 M2090 GPUs contribute to aggregate system peak of 615 TF

Full Scale



J.S. Vetter, R. Glassbrook *et al.*, "Keeneland: Bringing heterogeneous GPU computing to the computational science community," IEEE Computing in Science and Engineering, 13(5):90-5, 2011, http://dx.doi.org/10.1109/MCSE.2011.83.

How big?

The hosts:

Pittsburgh Supercomputing Center
National Institute for Computational Sciences
Georgia Tech
Internet2

Our satellite sites:

National Center for Supercomputing Applications
Pennsylvania State University
University of Oklahoma
University of South Carolina
University of Kentucky
Louisiana State University
University of Utah
University of California, Los Angeles

Many more turned away for this time.

Each site with full hands-on workstations, two way AV links for questions and help, and local TA's.

How did we do?

150+ attendees at all sites.

2 full days of lecture and hands-on.

No downtimes, no major real-time hiccups.

Evaluations are extremely positive. Projects in works.

What Next?

We are in post production to turn this workshop in to an on-line seminar.

We are already producing another OpenACC workshop in January to accommodate all of the remote sites we had to turn away. We will also take the opportunity to update this one even further.

We will use our new expertise in remote delivery to conduct workshops on related subject such as MPI and OpemMP.

Particularly helpful parties.

NICS: Bruce Loftis

PGI: Doug Miles, Michael Wolfe

NVIDIA: Roy Kim, Mark Harris, Mark Ebersole

And others that I am doubtless overlooking.

Don't Forget

If you found this interesting, and potentially useful, please visit our January 15th and 16th workshop site to see if you want to attend remotely:

http://www.psc.edu/index.php/training/openacc-gpu-programming

also readily findable from

psc.edu