# Accelerating Scientific Applications with GPUs


Presented by

**Jeremy Meredith**
**Kyle Spafford**

Future Technologies Group
Computer Science and Mathematics Division


# The GPU: Ideal candidate for scientific computing

- A massively parallel coprocessor
- Superior memory bandwidth
- Immense raw computing power
- SPMD processing with tens of thousands of threads

# CUDA

- NVIDIA's programming model makes GPUs more accessible than ever
- Decompose parallel sections of applications into kernels
- Execute those kernels on the GPU

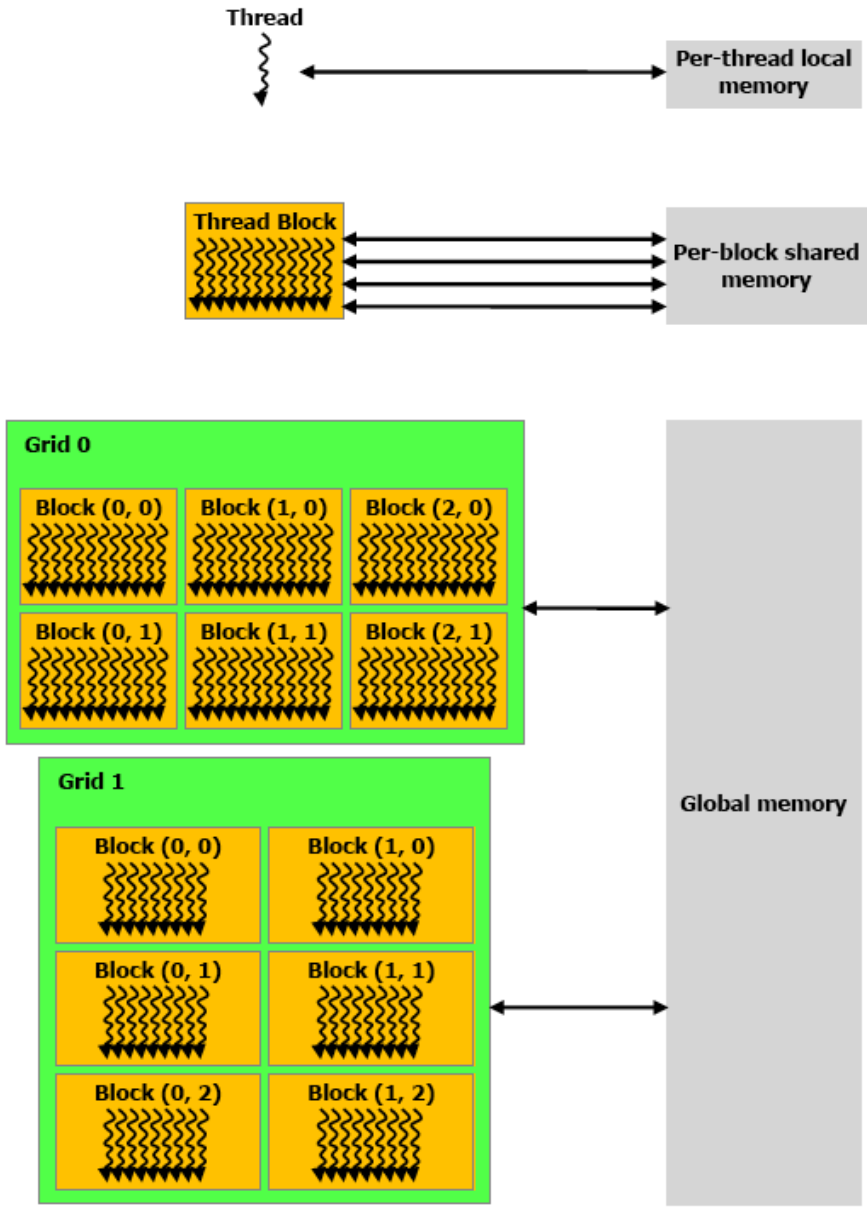



Image from NVIDIA CUDA Programming Guide

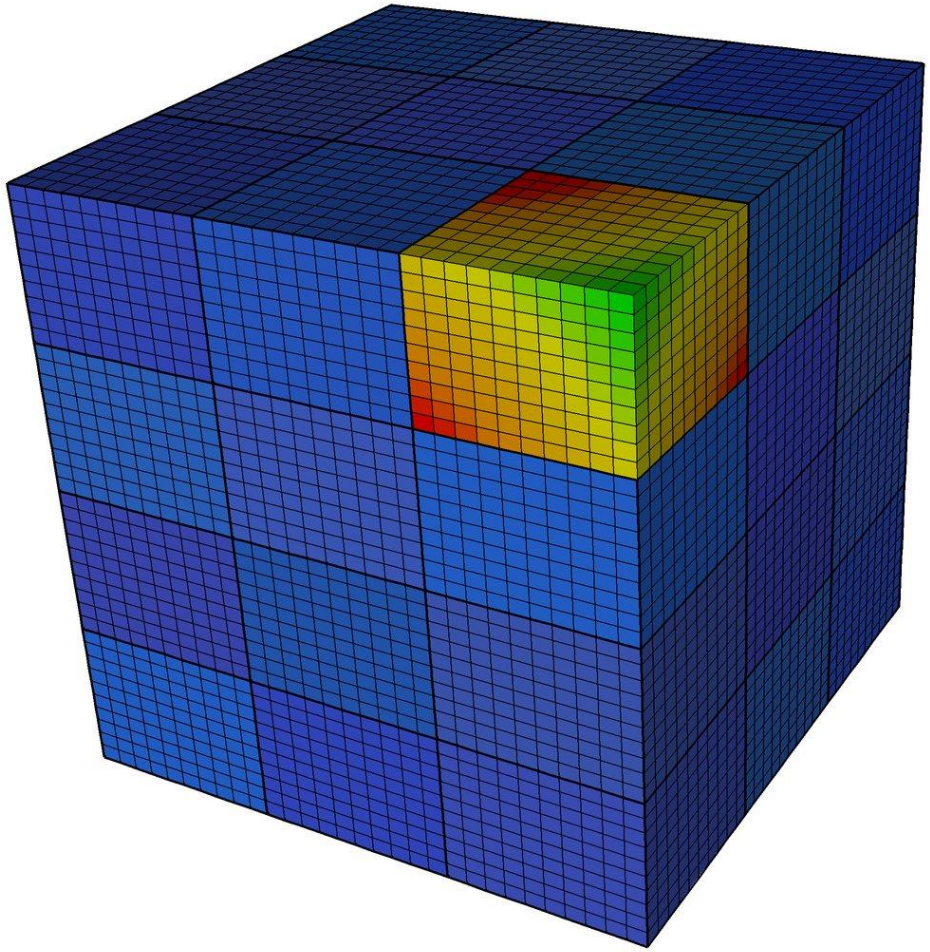
# The S3D simulation code

- S3D is a direct parallel solver for the Navier-Stokes equation, complete with detailed chemistry
- It is used to simulate combustion
- A faster version of S3D would allow scientists to simulate larger chemical domains at higher resolution, for longer periods of time

Image from: "Direct numerical simulation of a lifted ethylene-air flame."
C. S. Yoo, J. H. Chen, et al.

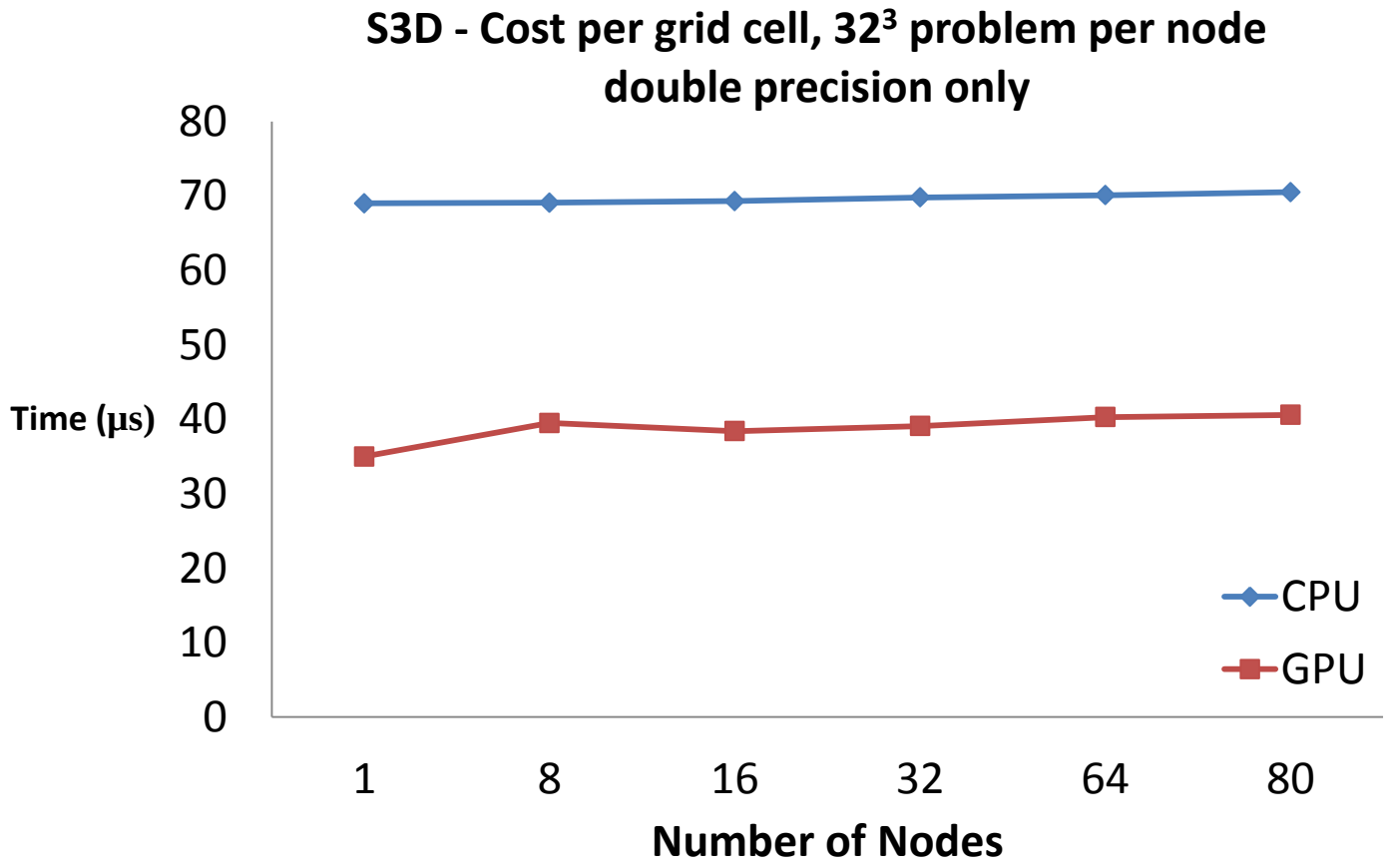
# Mapping S3D to CUDA

- S3D operates on a regular 3-D mesh, which we decomposed spatially into blocks
- Each block is further divided into threads, which solve for an individual point in space



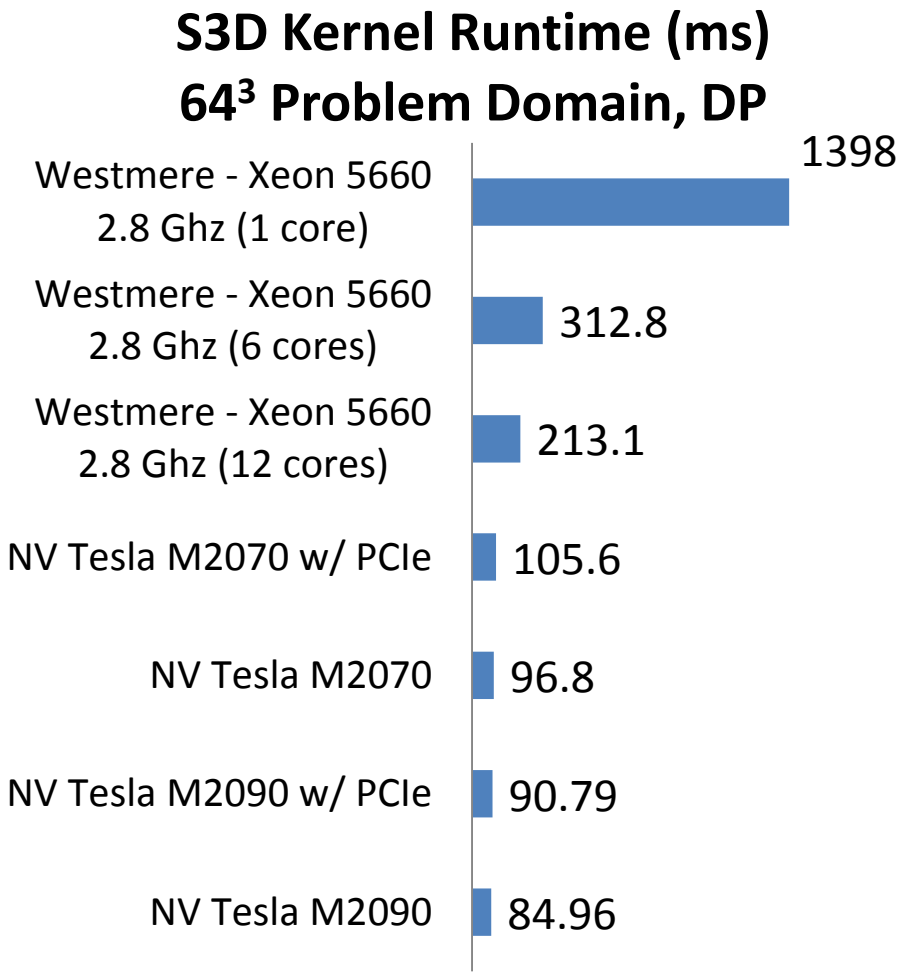
# Accelerating S3D's chemistry

**We implemented an accelerated version of S3D's Getrates kernel. The graph below shows the normalized runtime on Keeneland using one Tesla M2070 GPU or Westmere CPU per node.**

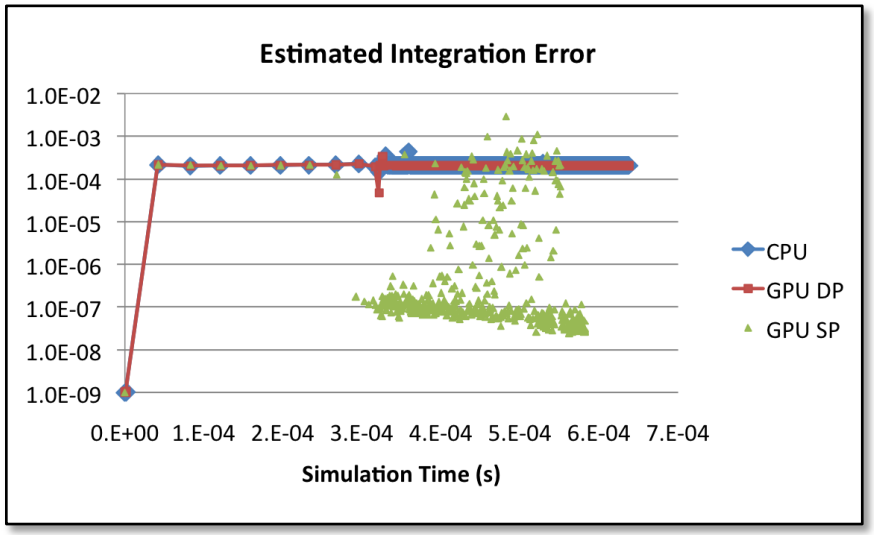


# Detailed Kernel Performance

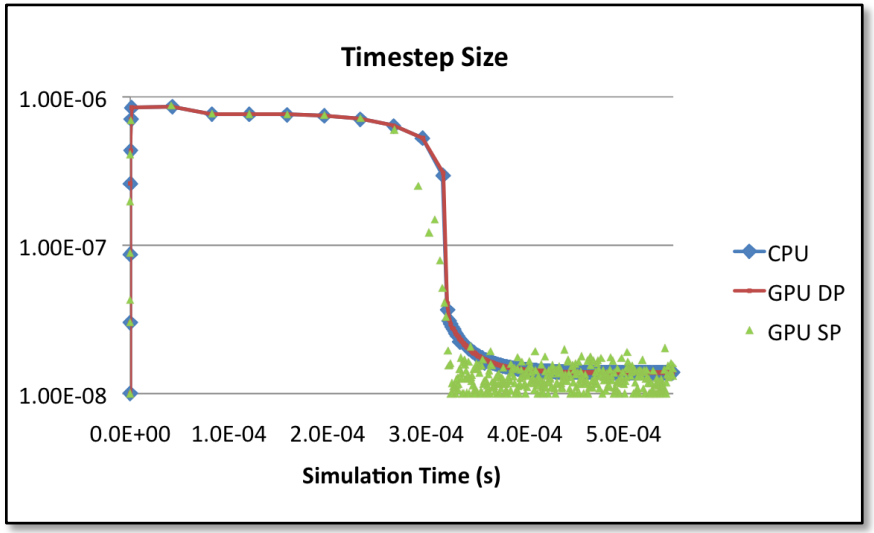
- This chart compares the performance of an optimized CPU version to the GPU version on the NSF Keeneland system.
- Performance measured using CUDA 4.0 and double precision.

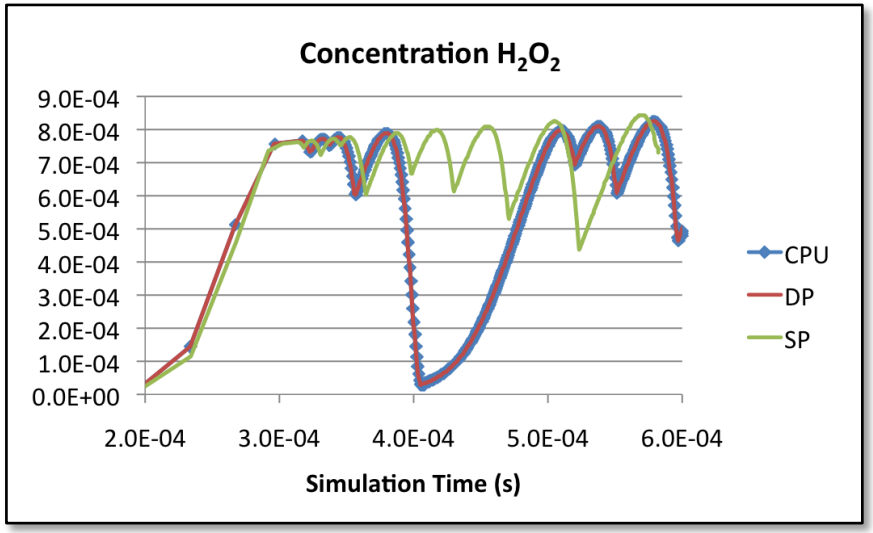


# Accuracy vs performance in S3D

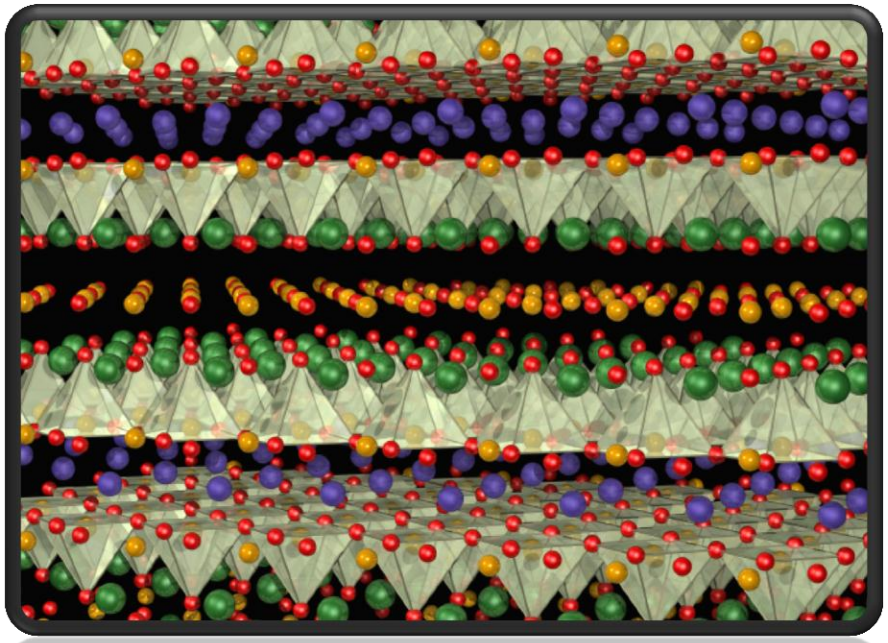


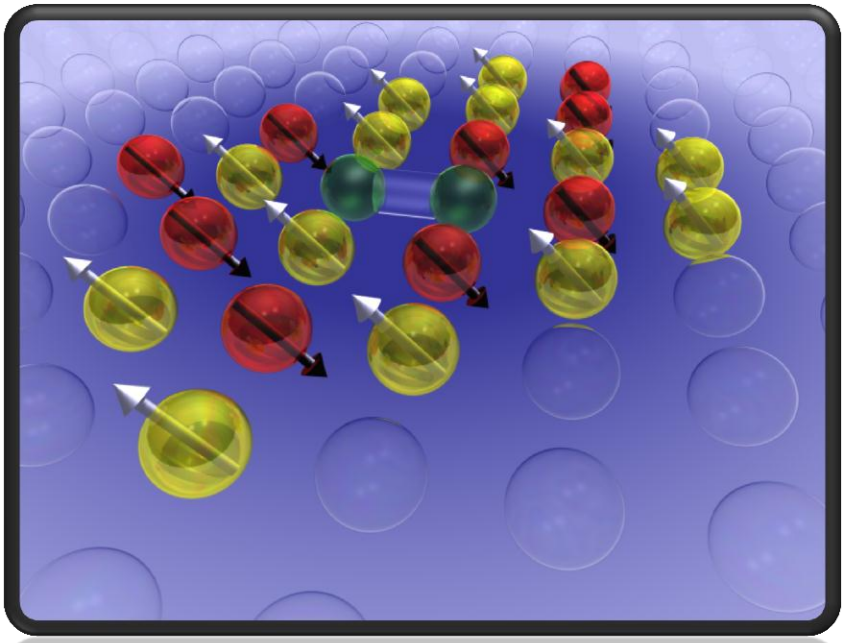

- Single precision is faster, but results in higher integration error
- Time step size shrinks to compensate
- It takes more iterations to simulate the same physical process
- Thus, the trade-off between precision and runtime is complex



# The DCA++ simulation code



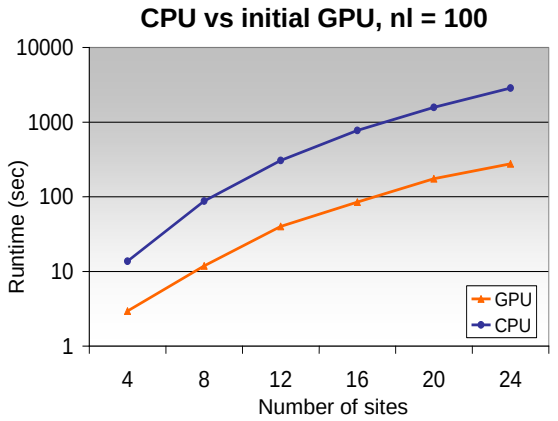


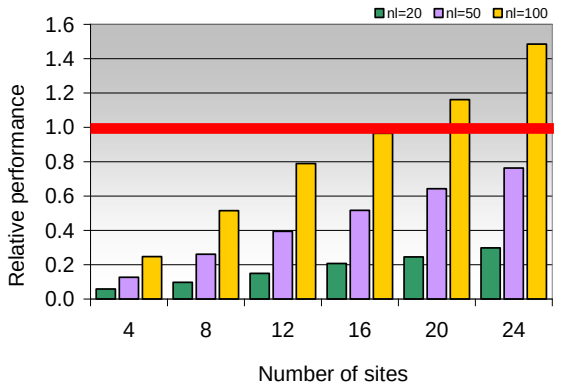
## DCA++ is a Quantum Monte Carlo simulation code

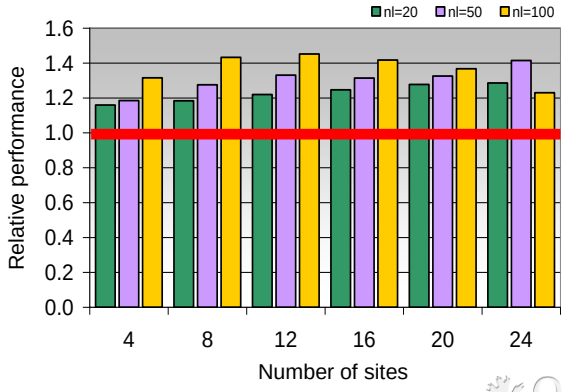
- Studies of high-temperature superconductivity and other materials science
- Implements Dynamic Cluster Approximation

# Mapping DCA++ to CUDA

- GPU BLAS offload alone resulted in significant speedups
- Further improvement by minimizing data transfers was only moderate because some functions remained on the CPU
- Porting remaining functions to GPU helped or hurt performance depending on problem size



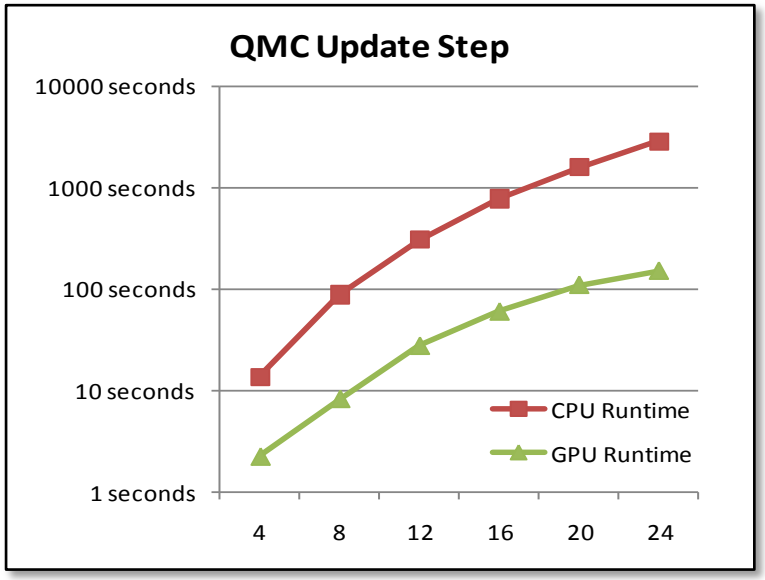



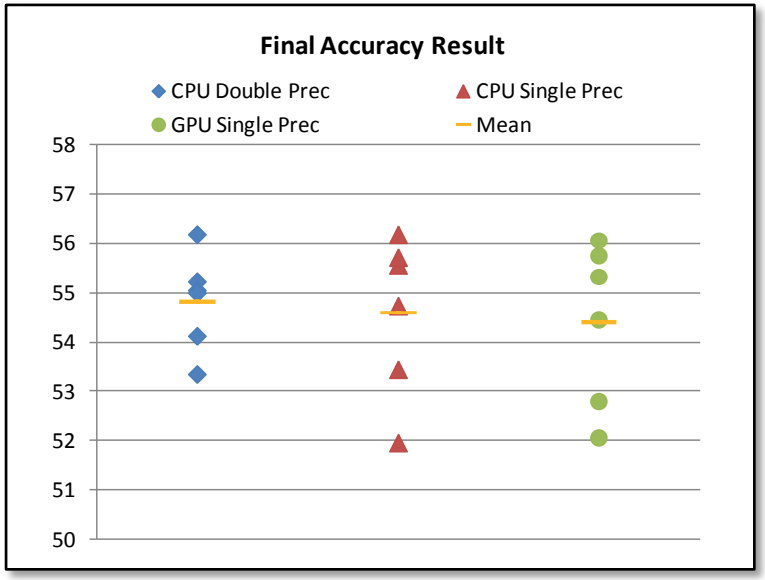



# DCA++ results

- QMC Update step showed almost 20× speedup after acceleration
- Accuracy remained within acceptable levels even when using single precision algorithms
- Full parallel GPU accelerated code showed 5× improvement

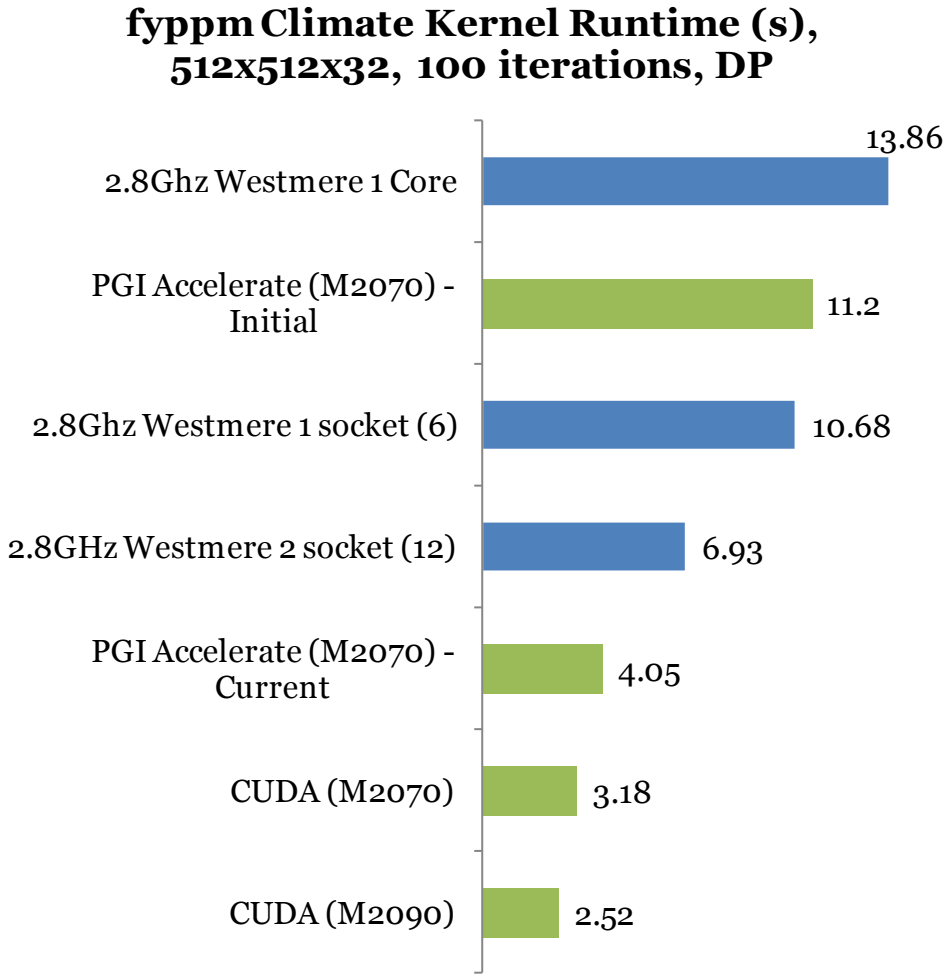
| Parallel Speedup | Median Speedup | Maximum Speedup |
|---|---|---|
| Warm-up phase | 13.46× | 17.28× |
| One QMC step | 13.32× | 17.10× |
| One measurement | 1.00× | 1.00× |
| One iteration | 4.96× | 5.56× |
| Whole code | 5.35× | |



# NOAA GFDL Climate Model Results

- Domain: Climate and weather modeling Cristopher Kerr, NOAA
- Evaluating performance-productivity tradeoff of hand-written CUDA and directive-based acceleration
- Measured kernel performance on Keeneland system and next-gen Sandy Bridge
- Initially legacy FORTRAN
- Currently running on multiple GPUs using OpenMP



# Moving forward

- Productivity and accuracy remain critical goals
  - OpenCL improves code portability
    - Open standard for parallel heterogeneous computing
    - Abstractions are similar to CUDA
    - www.khronos.org/opencl
  - GPU improvements in precision
    - Full double-precision support in latest GPUs
    - Speed is improving relative to single precision performance
- References


K. Spafford, J. Meredith, J.S. Vetter, J. Chen, R. Grout, R. Sankaran. "Accelerating S3D: A GPGPU Case Study," *Proceedings of the Seventh International Workshop on Algorithms, Models, and Tools for Parallel Computing on Heterogeneous Platforms (HeteroPar 2009),* Delft, The Netherlands.

J.S. Meredith, G. Alvarez, T.A. Maier, T.C. Schulthess, J.S. Vetter, "Accuracy and Performance of Graphics Processors: A Quantum Monte Carlo Application Case Study," *Parallel Computing,* Volume 35, Issue 3, March 2009.

G. Alvarez, M.S. Summers, D.E. Maxwell, M. Eisenbach, J.S. Meredith, J.M. Larkin, J. Levesque, T. A. Maier, P.R.C. Kent, E.F. D'Azevedo, T.C. Schulthess, "New algorithm to enable 400+ TFlop/s sustained performance in simulations of disorder effects in high-Tc superconductors," SuperComputing, 2008. (Gordon Bell Prize winner)

# Contacts


## Jeremy Meredith

**Future Technologies Group**
**Computer Science and Mathematics Division**
**(865) 241-5842**
**jsmeredith@ornl.gov**

## Kyle Spafford

**Future Technologies Group**
**Computer Science and Mathematics Division**
**(865) 574-0005**
**spaffordkl@ornl.gov**