NCAR Weather and Climate Code
Optimized on Nvidia CUDA

Tony Heller and Dan Connors - CSU
Rich Loft - NCAR
The atmosphere can be mathematically partitioned into a 2-D grid across the earth's surface, with a third dimension consisting of atmospheric layers.

NCAR's widely used WRF (Weather Research and Forecast) model uses this geometry.

Ideally suited for parallelization, particularly on a GPU.
• GPUs were designed for doing lots of simultaneous math computations. Video games need to calculate lighting and geometry for millions of triangles - at least 30 frames per second.

• Weather models have similar requirements - millions of symmetrical calculations.

• GPUs use SIMT (Single Instruction Multiple Threads.)

• The idea of using GPUs for general purpose computing has been around for about a decade. GPGPU

Why Graphics Processors?
• Cuda is Nvidia's parallel programming interface to their graphics processors (GPUs)

• Nvidia GPUs consist of blocks of "multiprocessors" each containing a number of "thread processors"

• Each vertical atmospheric column is assigned to a multiprocessor

• Each atmospheric layer within a column is assigned to a thread processor
High Level Cuda Architecture
Typical WRF Forecast
Barra"cuda"
Gap Between GPUs and CPUs is Increasing

Cuda Memory Interactions
- Shared memory allows neighbors within a multi-processor to communicate.

- There is very little communication between adjacent WRF layers during a single time step. This allows most of the state to be kept in device registers.

- There are a limited number of device registers, so some state has to be spilled over into shared memory.

- Device memory is much slower, and is only used for communication with the CPU. Communication can be overlapped with computation.
Typical calculations include rain, snow, clouds, ice etc. Everything that affects - or is - the weather. 32 bit computing is adequate for weather forecasting.

Information from each time step is passed on spatially and temporally.

WRF is written in Fortran and was ported to Cuda initially by John Michalakes at NCAR.

My effort involved restructuring and optimizing the code.
Typical WRF Cuda Code

```
// practw: accretion of cloud water by rain [HL A40] [LFO 51]
//        (C->R)
//posix_points

denfac = shared_denfac[thread_index];

if(qr > QCRMIN && qc > QMIN)
{
    pracw = min(PACRR * rslope3r * rslopebr *qc * denfac, qc / dtcld);
}
```
● On a dual Nvidia 9800GX2 processor system, the code ran about 500X faster than an equivalent C version on an AMD Athlon Dual Core 4600+

● On a very low cost dual 8600 system, it ran about 100X faster than on the Athlon system. Not bad for an $80 add on to the system - which also makes games run faster

● Precision was identical to the 5th decimal place
Future of GPU Computing

- Nvidia Cuda
- AMD FireStream
- Intel Larrabee
- Converging on OpenCL standard?
- Climate models need 1,000 current computing performance
- Need 64 bit math
- GPUs can deliver both
- I have done some initial evaluations with NCAR, and they appear promising

Future of Climate Modeling?