Enabling Science with Graphical Processing Units (GPUs)

Jacques Middlecoff and Mark Govett

NOAA - Earth System Research Laboratory

Introduction

NOAA's Earth System Research Laboratory (ESRL) is breaking new ground by investigation GPUs as a way to get the massive computing power needed for advancing new science.

Many fields of science, including data assimilation, weather models, and ensemble forecasting require ever more computing power. We believe GPUs can provide this power and we think ESRL's new Non-hydrostatic Icosahedral Model (NIM) is an excellent example of a code that requires the power of GPUs to perform as intended.

NIM is targeted to run at 4km resolution over the globe within 3 years. At this resolution, over 200,000 traditional CPU cores would be required to meet operational requirements (2% of real-time) and 200,000 cores is impractical due to power requirements, cost, and reliability.

To address the NIM computational requirements, we have been testing Nvidia GPUs and we have determined that GPUs give a 34X speedup over CPUs and so only 5,000 GPUs will be required to run NIM at 4km resolution in 2% of real time.

NVIDIA GPU Gives 34X Speedup

The GPU
- Highly parallel structure
- Very efficient at manipulating video graphics
- Much faster than CPUs for weather codes.

The GPU gives 34X speedup for NIM
- We have ported NIM dynamics to the GPU
- NIM dynamics is 34 times faster on GPU than on CPU

The GPU translator F2C-ACC
- NIVIDIA GPU requires it's own language called CUDA
- CUDA is a challenge since our codes are written in Fortran
- We have developed the tool F2C-ACC that translates Fortran into CUDA
- We used F2C-ACC to port some of the FIM and all of NIM dynamics to CUDA

Commercial compilers
- Under development
- Will compile Fortran directly into CUDA
- If adequate, may eventually augment or replace F2C-ACC

Summary

We have demonstrated that CPUs are inadequate for next generation weather codes leaving a void that can be filled by GPUs. We have used NIM to show that GPUs give a 34X performance improvement over CPUs and thus GPUs can enable new science. We have shown that F2C-ACC allows the efficient porting of Fortran code to CUDA.

NIM Computational Design

Preliminary version running by August 2010
- Using same infrastructure as ESRL's current model (FIM)
- Like FIM, uses SMS for very fast parallelization

NIM being developed specifically for the GPU
- Designed by team of meteorologists and software engineers
- Vertical method designed specifically for the GPU
- Exploiting fine grain parallelism

CPU 4 km NIM 1 Day Forecast Projected

CPUs	Points per Processor	Time (hours)	Percent of Real Time
1280	32768	60	250%
2560	16384	30	125%
5120	8192	16	67%
10240	4096	8.4	35%
20480	2048	5.0	21%
40960	1024	3.8	16%

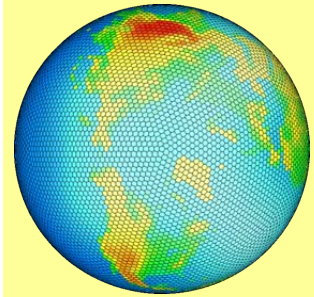
Assumes NIM time is 2X FIM time

GPU 4 km NIM 1 Day Forecast Projected

GPUs	Points per Processor	Time (hours)	Percent of Real Time
1280	32768	1.53	6.4%
2560	16384	0.81	3.4%
5120	**8192**	**0.46**	**1.9%**
10240	4096	0.27	1.1%
20480	2048	0.16	0.7%
40960	1024	0.12	0.5%

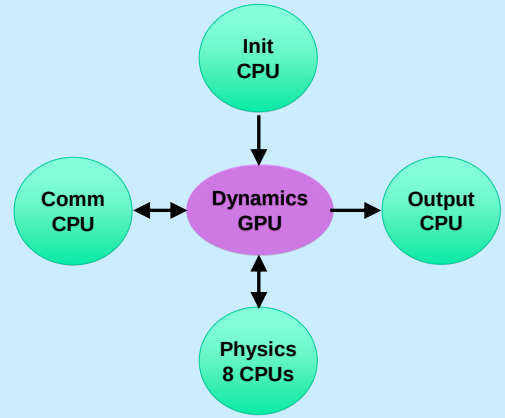
Assumes GPU is 34X CPU speed
Assumes 6X GPU speedup by 2012

The Icosohedral Grid



The FIM and NIM models are based on the icosahedral grid. This is a horizontal grid composed of mostly hexagons and ten pentagons that more accurately represent the globe and do not contain the well-known pole problems characteristic of other global grids.

How we got 34x Speedup



Everything on the GPU
- Break from widespread view of GPU as a kernel accelerator
- Kernel accelerator results in costly copying between CPU and GPU
- Current GPUs have large memories
- Goal is to run everything on GPU except init, comm, and output
- 34x speedup measured with no copies

Current Plans
- NIM dynamics is ported to the GPU and gives 34x speedup
- Physics is not ported to the GPU but does not run every time step
- Will run dynamics on the GPU and physics on 8 CPUs per GPU
- Expect 17x overall speedup over CPU

Future Work

- Get NIM running on the GPU with physics on multiple CPUs
- Experiment with GPU to GPU communications
- Port physics to the GPU
- Port FIM to the GPU