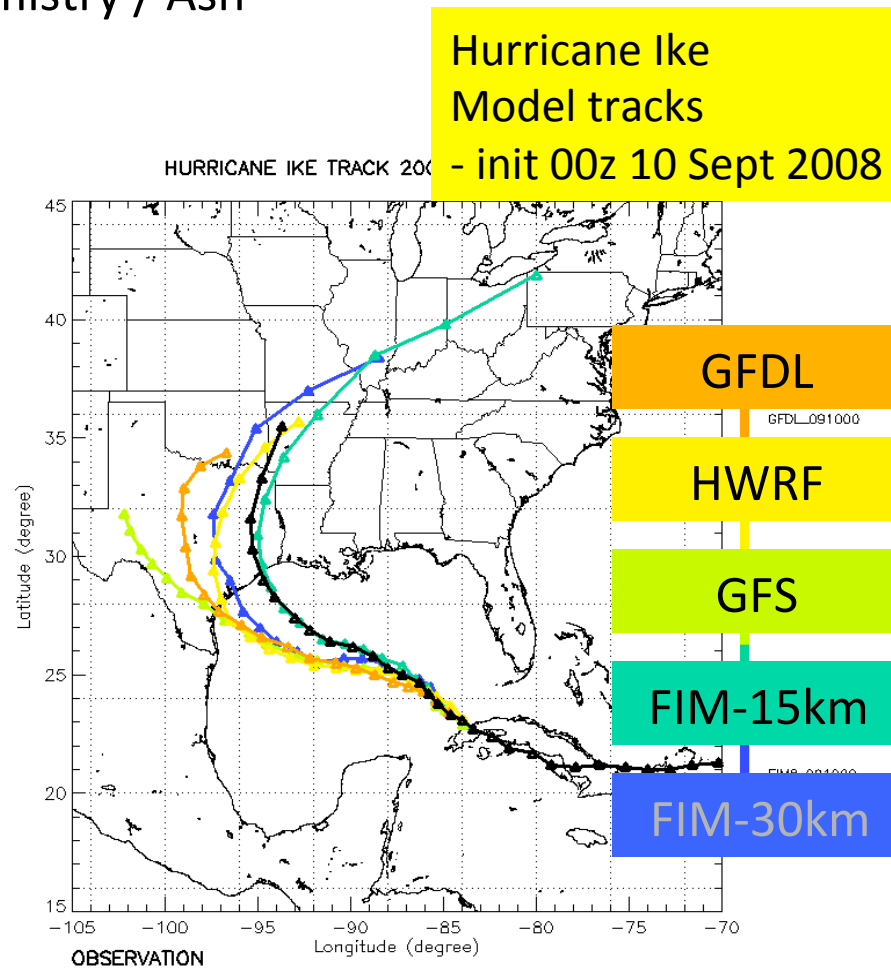
Using GPUs to Run Next-Generation Weather Prediction Models

Mark Govett
Jacques Middlecoff, Tom Henderson,
Craig Tierney, Paul Madden, Jim Rosinski

May 2011

Model Development Activities

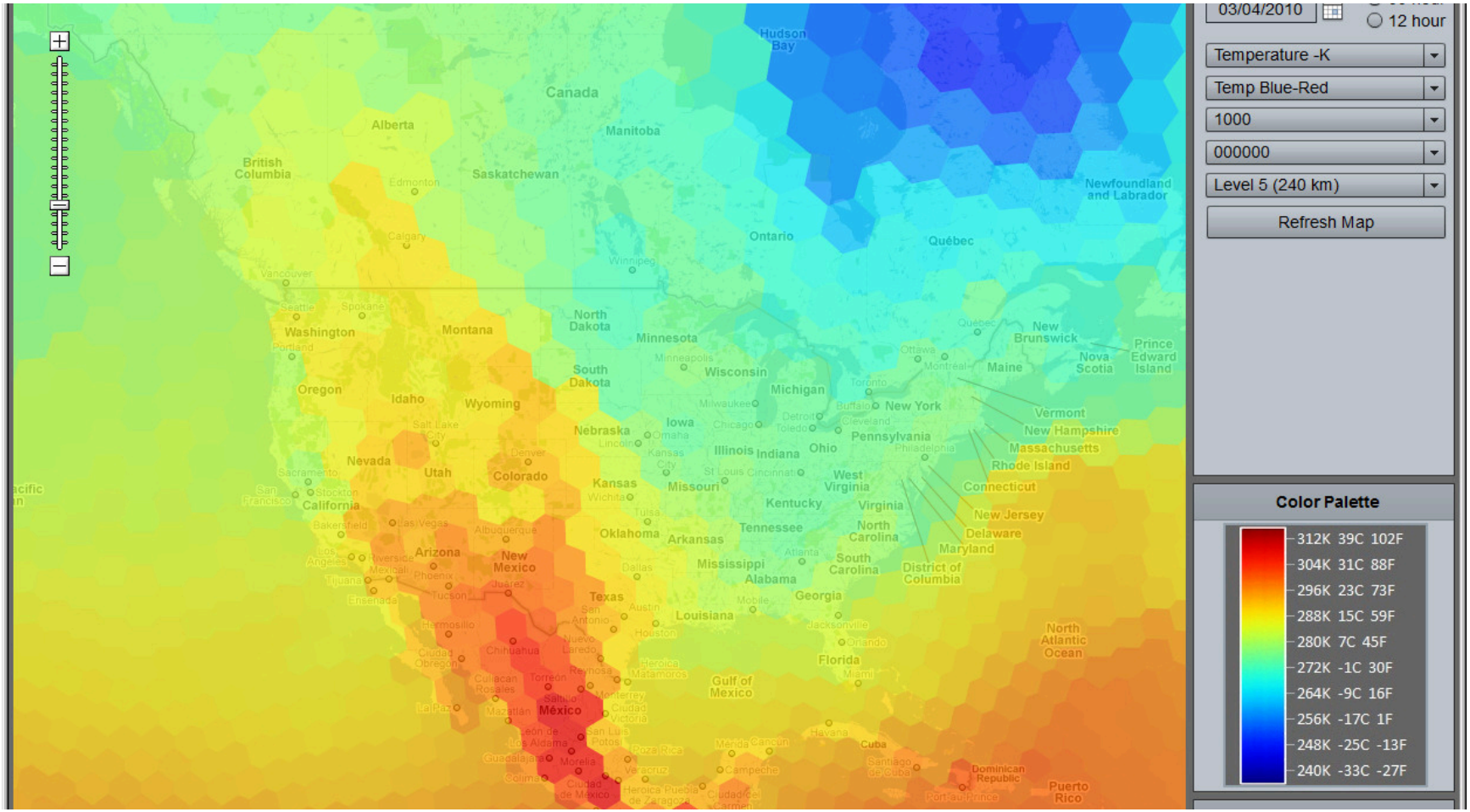
- Regional, Local Models (1-5 KM)
  - NOAA HRRR, WRF-ARW, WRF-NMM, etc
    - Hurricanes, Aviation, Fires, Chemistry / Ash
  - Ensembles (15-30KM)
- Global Models (10-30 KM)
  - NOAA FIM model
  - Improved hurricane forecasts



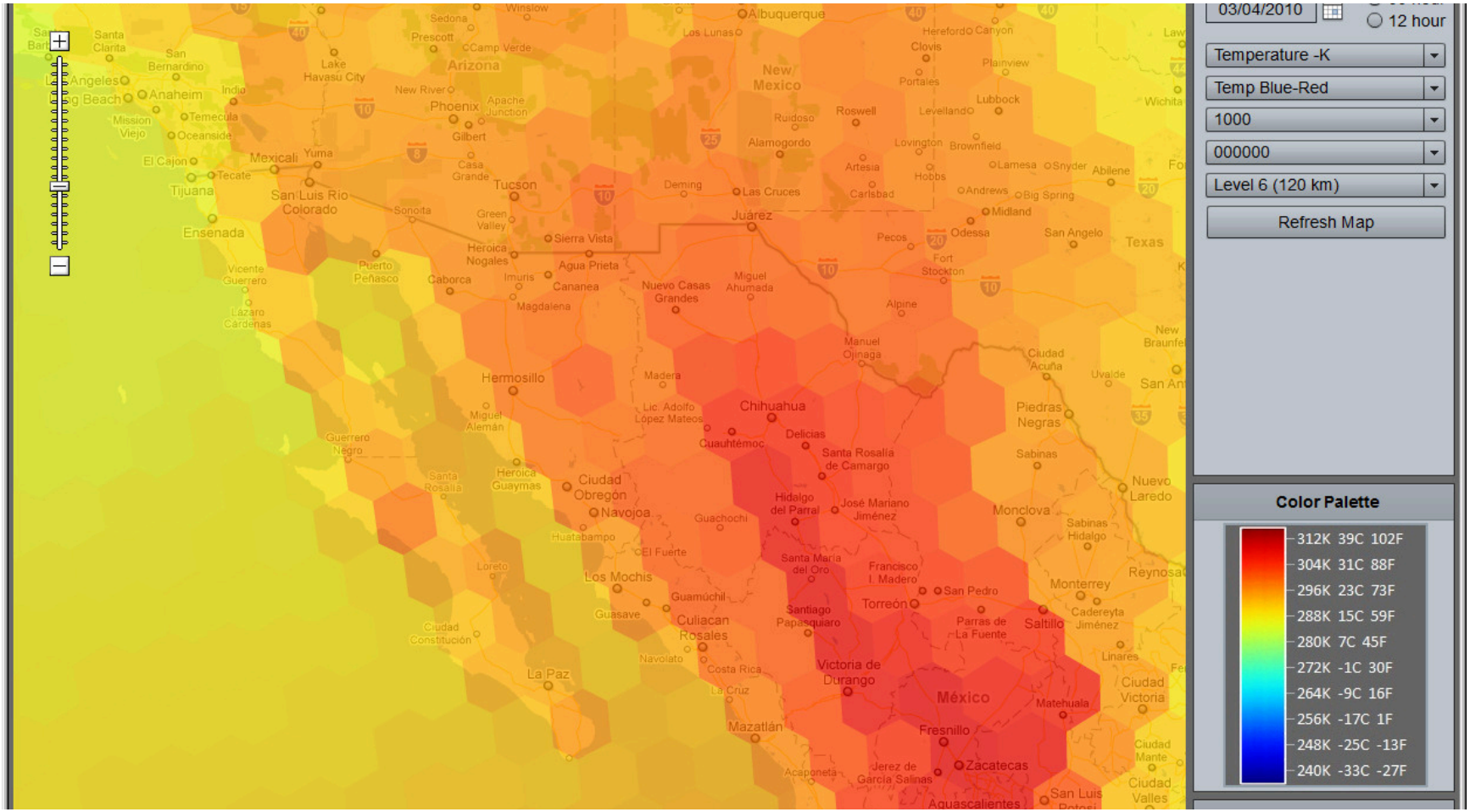
Computing Requirements

- 3000 cores:
  - 15KM Global FIM
- 126,000 cores
  - 21 member 30 KM ensemble

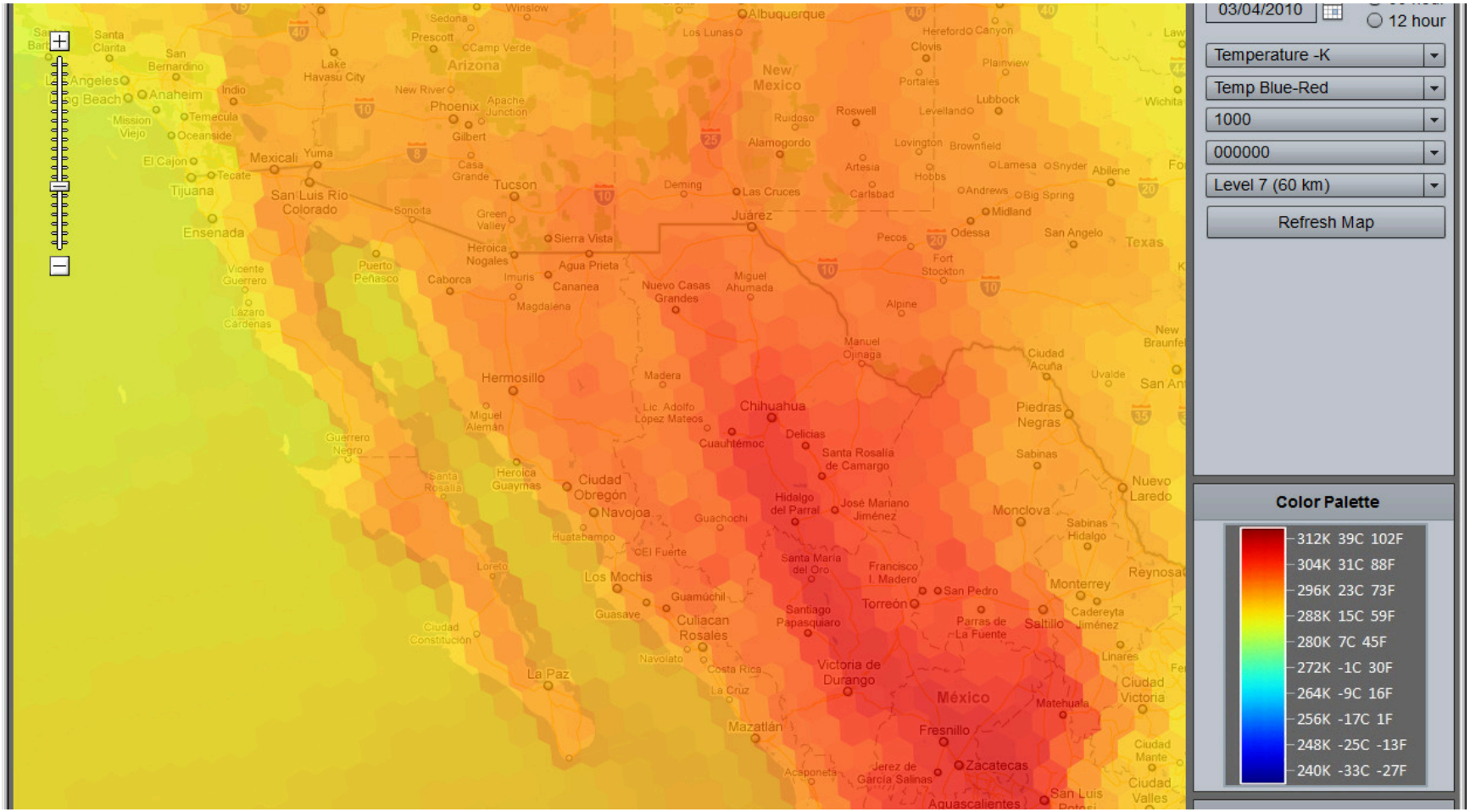
FIM 240KM Resolution on GE



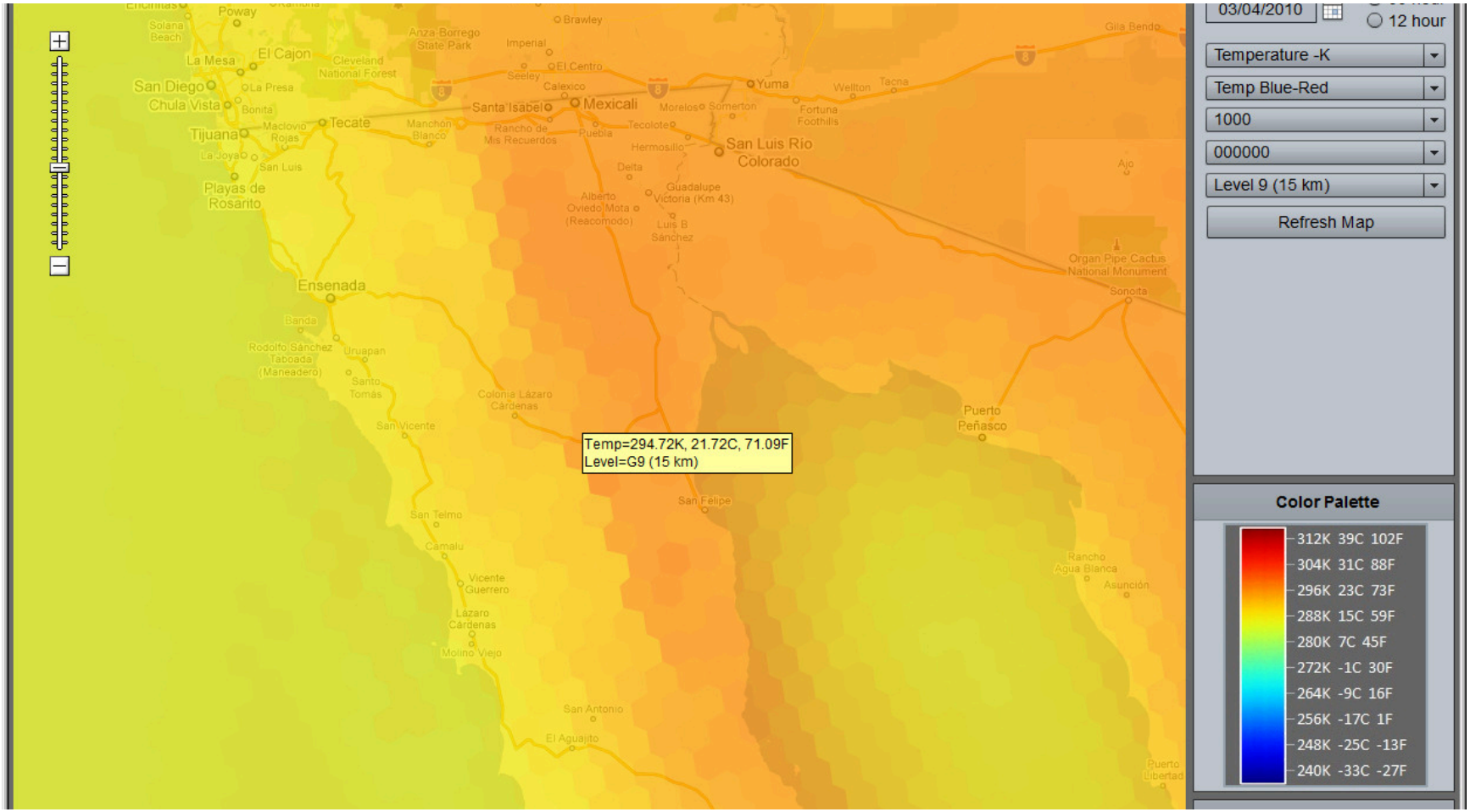
FIM 120 KM


03/04/2010
12 hour
Temperature -K
Temp Blue-Red
1000
000000
Level 6 (120 km)
Refresh Map
Color Palette
312K 39C 102F
304K 31C 88F
296K 23C 73F
288K 15C 59F
280K 7C 45F
272K -1C 30F
264K -9C 16F
256K -17C 1F
248K -25C -13F
240K -33C -27F
Albuquerque
Arizona
New Mexico
Texas
Phoenix
Tucson
Juárez
Hermosillo
Chihuahua
Ciudad Obregón
Navojoa
Los Mochis
Culiacan Rosales
Mazatlán
La Paz
Victoria de Durango
Torreón
Saltillo
Monterrey
Fresnillo
Zacatecas
México

FIM 60KM Resolution



FIM @ 15KM resolution



Computing Requirements

- Availability of computing limit advances in modeling
  - 7 day forecast takes 3.5 hours (2% real-time)
  - 15 KM global model run requires ~1500 cores
    - 8 KM requires ~12,000 cores
    - 4 KM - - - > 96,000 cores
  - Ensembles
    - A 21 member 10KM FIM ensemble required 126,000 cores on Jaguar
- Convective Initiation (CI)
  - Proposed modest experiments due to lack of computing resources

Global Cloud Resolving Models (GCRM)

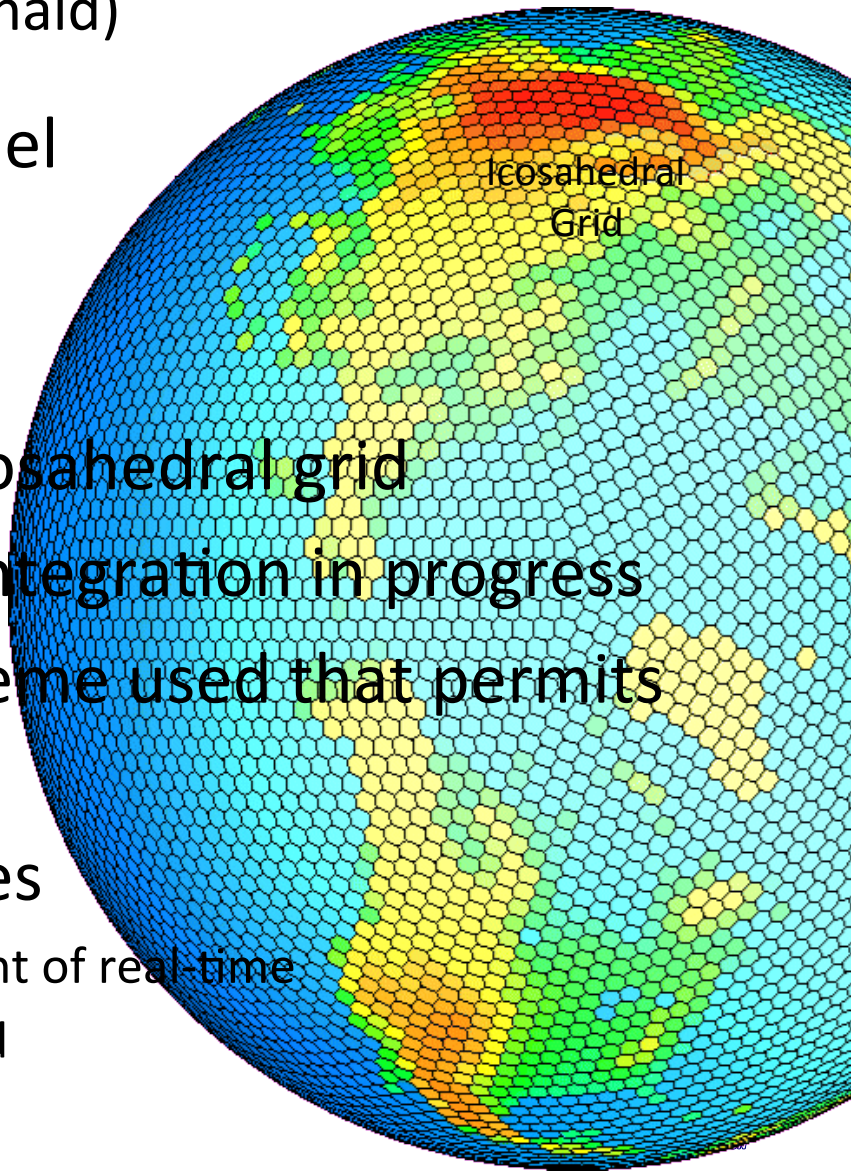
< 2KM horizontal resolution

- Benefits
  - Weather and climate scales (5-100 day forecasts)
    - Improved hurricane track and intensity prediction
    - Regional climate impacts
- Active developments in the research community
  - NICAM: University of Tokyo
  - GCRM: Colorado State University
  - GFDL: Finite Volume Cubed Sphere
  - NIM: NOAA Earth System Research Laboratory
- Computing Requirements
  - CSU's 4KM GCRM was run on 80,000 cores of DOE Jaguar
    - Simulations ran at ~50 percent of real-time

Non-hydrostatic Icosahedral Model (NIM)

(Lee, MacDonald)

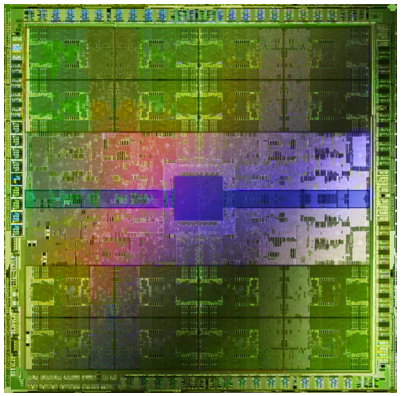
- Global Weather Forecast Model
- Non-hydrostatic
- Targeting 2KM resolution
- Uniform, hexagonal-based, icosahedral grid
- Dynamics complete, physics integration in progress
- Novel indirect addressing scheme used that permits concise, efficient code
- Designed for GPU architectures
  - 5000 GPUs needed to run in ~2 percent of real-time
  - ~ 200,000 CPU cores would be needed



GPU / Multi-core Technology

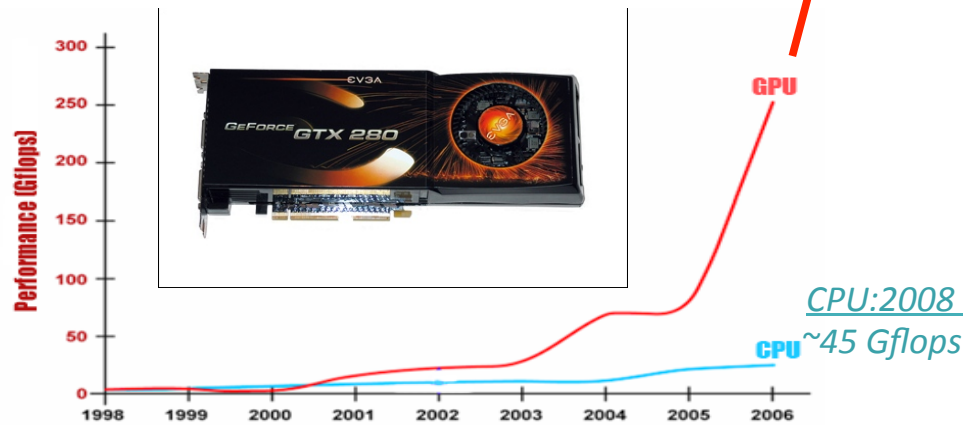
- NVIDIA: Fermi chip first to support HPC
  - Formed partnerships with Cray, IBM on HPC systems
  - #1, #3 systems on TOP500 (Fermi, China)
- AMD/ATI: Primarily graphics currently
  - #7 system on TOP500 (AMD-Radeon, China)
  - Fusion chip in 2011 (5 TeraFlops)
- Intel: Knights Ferry (2011), 32-64 cores

NVIDIA: Fermi (2010)



✧ 1.2 TeraFlops
✧ 8x increase in double precision
✧ 2x increase in memory bandwidth
✧ Error correcting memory

NVIDIA: Tesla (2008)



Hardware Cost Benefit

DOE Jaguar System

- 2.3 PetaFlops
- 250,000 CPUs
- 284 cabinets
- 7-10 MW power
- Cost: ~ $100 million
- Building: $75 million

Reliability in hours

Equivalent GPU System

- 2.3 PetaFlop
- 2000 Fermi GPUs
- 20 cabinets
- 1.0 MW power
- Cost: ~ $10 million
- **Reliability in weeks**

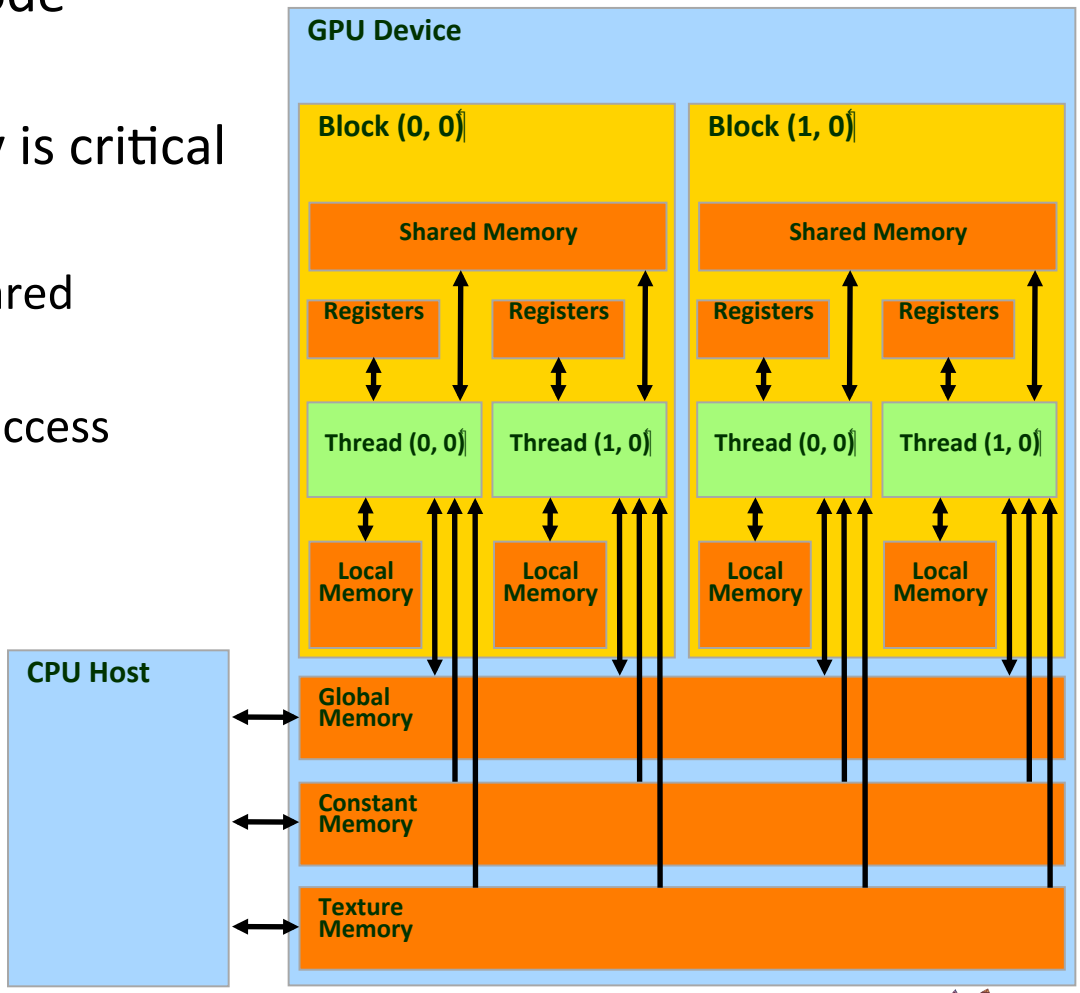
- Large CPU systems (>100 thousand cores) are unrealistic for operational weather forecasting
  - Hardware Cost: ~ 10x
    - ~15x including facilities cost
  - Power Consumption: 7-10x
    - CPU: 7MW @ .05/KWH => $3M / year
  - Reliability: 25-50x
    - Fewer components

Valmont
Power Plant
~200 MegaWatts
Boulder, CO

Application Performance

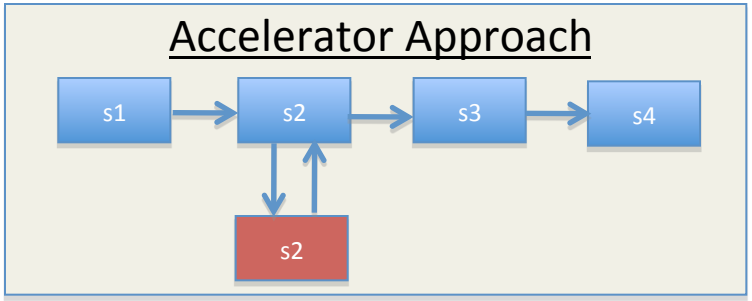
- Highly dependent on code
  - 20-50x is possible
- Efficient use of memory is critical to good performance
  - 1-2 cycles to access shared memory & registers
  - Hundreds of cycles to access global memory

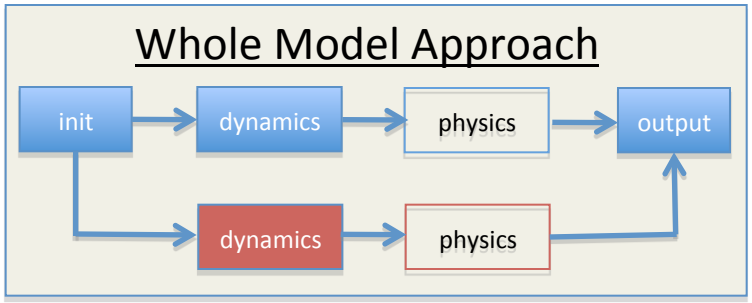
Memory	Tesla	Fermi
Shared	16K	64K
Constant	16K	64K
Global	1-2GB	4-6GB



GPU Parallelization Approaches

- Accelerator Approach
  - Target select routines for acceleration
  - Copy between CPU and GPU can be significant
  - Can be a step-wise approach to parallelization





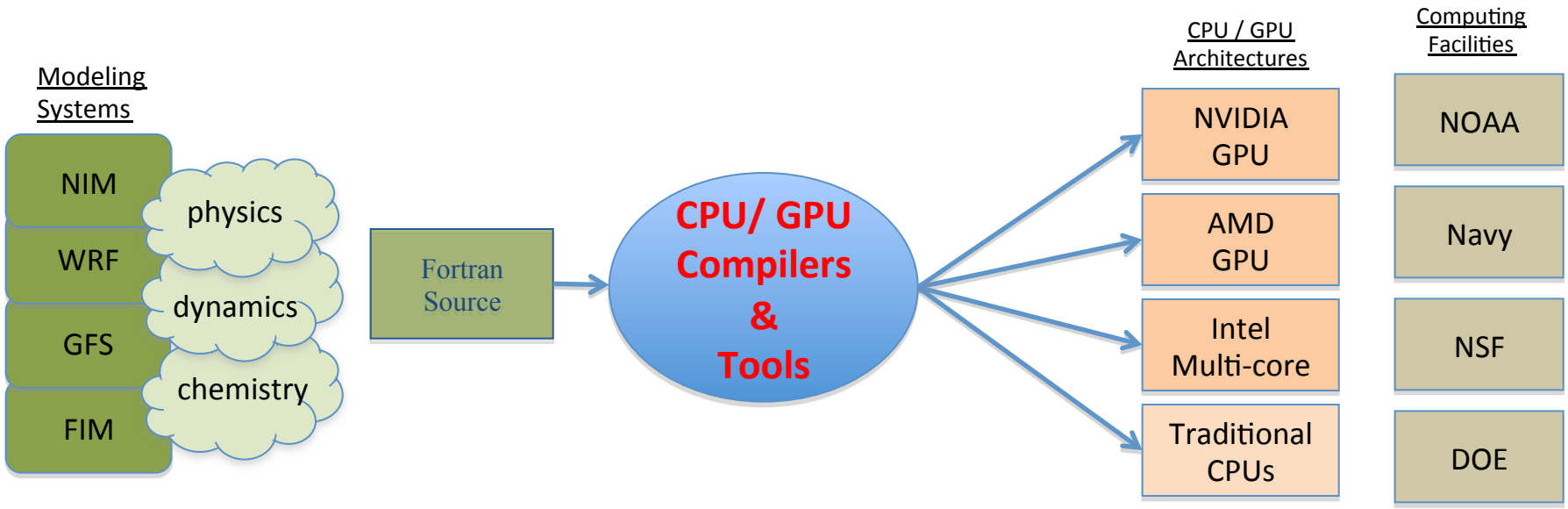
- Whole Model Approach
  - Keep data resident on the GPU
    - Communications only needed for input, output and for inter-process communications
  - May requires code conversion, code restructuring or rewriting

GPU Programming Approaches

- Language Approach
  - CUDA, OpenCL, CUDA Fortran, etc.
  - User control over coding and optimizations
    - Vendor specific optimizations
    - May not be portable across architectures
- Directive-based Approach
  - Appear as comments in the source
    - !ACC$DO VECTOR (1)
  - Maintain a single source
    - Scientists can modify themselves
    - Can run in serial, parallel, CPU, GPU architectures
  - Compilers can analyze and generate efficient code
    - Dependent on maturity

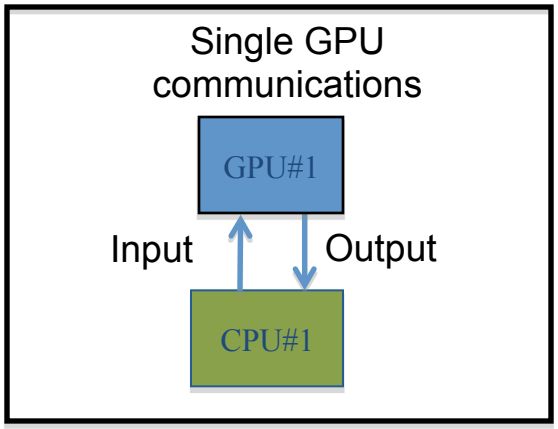
Performance and Portability

- Models are becoming increasingly complex
- Increasingly important to maintain a single source
  - Operations, research, collaboration
  - Especially for models under active development



Code Parallelization (2009)

- Developed the Fortran-to-CUDA compiler (F2C-ACC)
  - Commercial compilers were not available in 2008
  - Converts Fortran 90 into C or CUDA-C
  - Some hand tuning was necessary
- Parallelized NIM model dynamics
  - Tesla Chip, Intel Harpertown (2008)
  - Result for a single GPU
  - Communications only needed for I/O

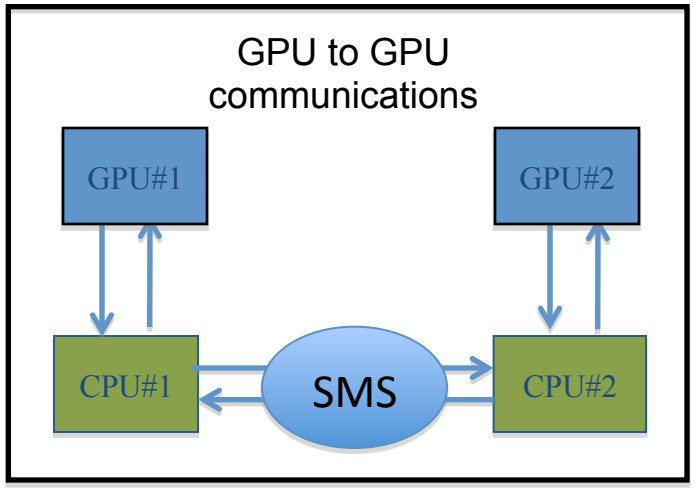


NIM Dynamics (version 160)

Resolution	HorizPts	Harpertown	Tesla	Nehalem	Fermi
G4-480km	2562	2.13	0.079 (26.9)	1.45	0.054 (26.7)
G5-240km	10242	8.81	0.262 (33.5)	5.38	0.205 (26.2)

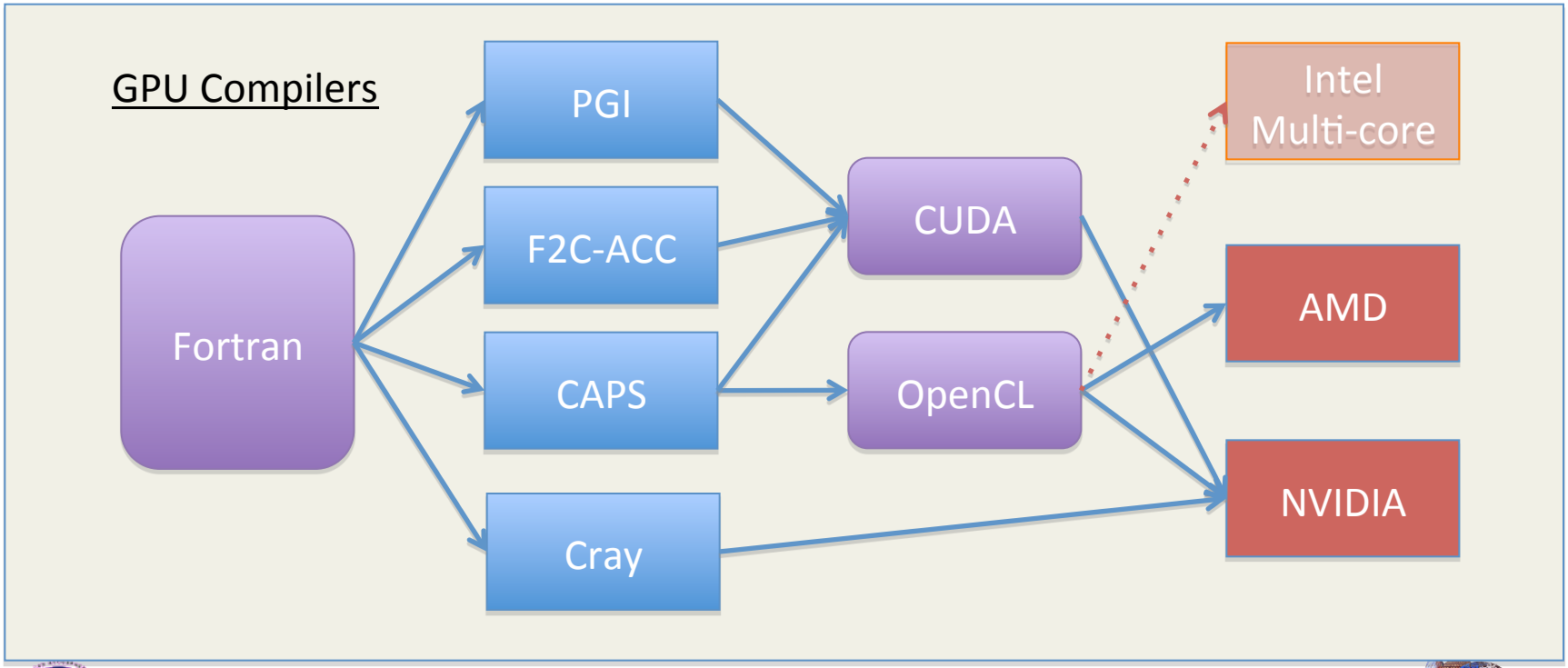
Model Parallelization (2010)

- Updated NIM Model Parallelization
  - Active model development
  - Code optimizations on-going
- Evaluate Fortran GPU compilers
  - Use F2C results as benchmark
- Evaluate Fermi
- Run on Multiple GPUs
  - Modified F2C-ACC GPU compiler
  - Uses MPI-based Scalable Modeling System (SMS)
  - Testing on 10 Tesla & 10 Fermi GPUs



GPU Compilers and Portability

- **The goal is to maintain a single source**
  - Fortran code base + parallelization directives
  - Optimal performance can require special coding
- Growing number of vendor compilers & tools
  - We continue to evaluate them



Fortran GPU Compiler Comparison (2009)

Using NIM G4 - 2562 horizontal points, 96 vertical levels
Tesla GPU vs. Intel Harpertown CPU Core

	Harpertown CPU Time	F2C-ACC CUDA-C Tesla GPU Speedup	HMPP Tesla GPU Speedup	PGI Tesla GPU Speedup
vdmints	88.86	34	37	11
vdmintv	37.73	38	38	24
flux	17.97	25	17	24
vdn	12.77	22	17	--
diag	5.13	29	60	52
force	5.34	46	28	12
trisol	8.46	4	6	--
TOTAL	194.25	(7.48) 26	(7.49) 26	

Fortran GPU Compiler Comparison (2011)

Using NIM G4 - 10242 horizontal points, 96 vertical levels

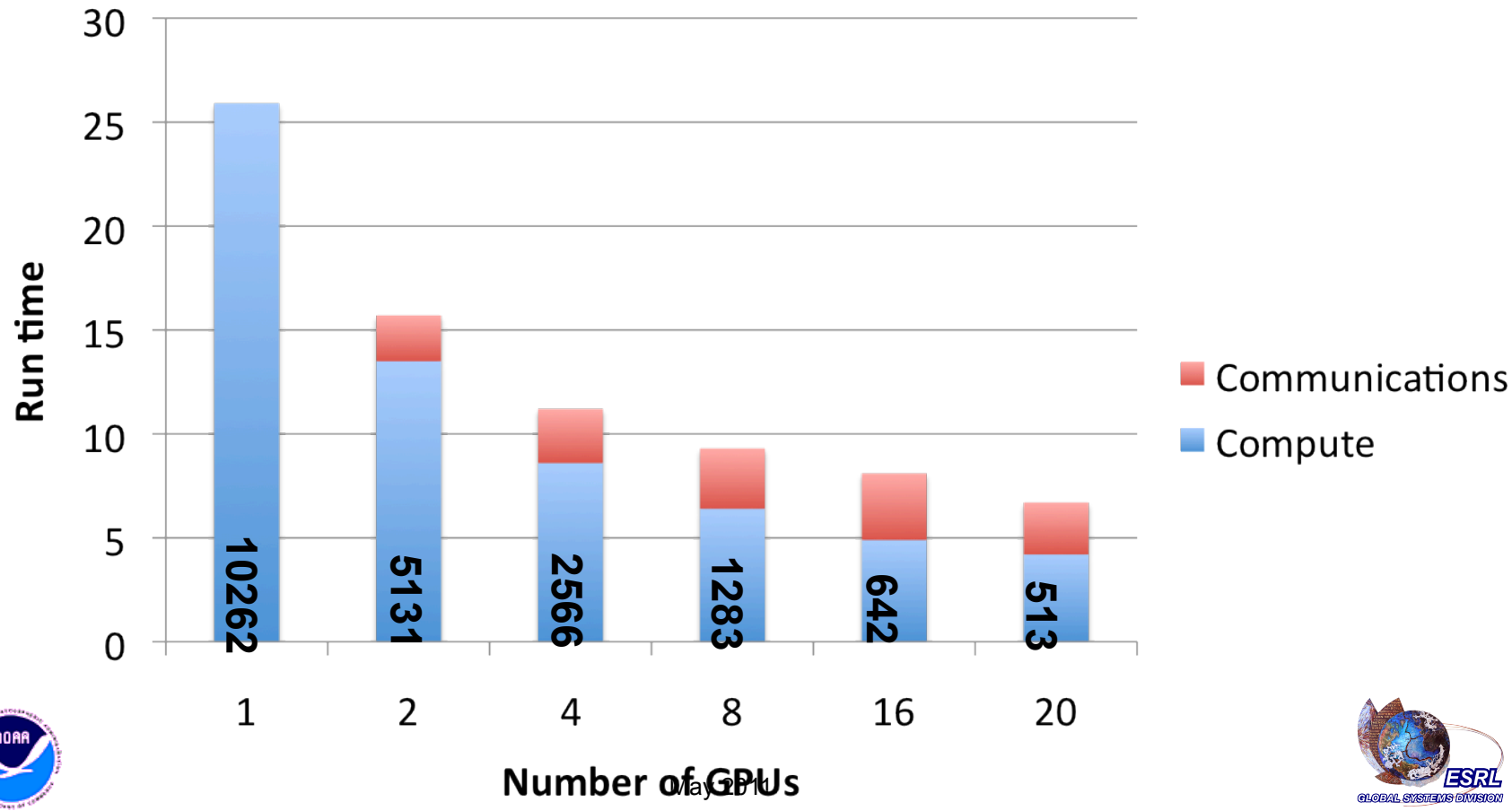
Fermi GPU vs. Intel Westmere CPU Core

	Westmere single core runtime	F2C-ACC CUDA-C Speedup	HMPP Speedup	PGI Speedup
vdmints	223.38	29		
vdmintv	91.95	32		
flux	48.99	38		
vdn	17.71	9		
diag	18.13	43		
force	16.54	48		
trisol	14.52	10		
TOTAL	457.83s	(17.8s) 25		

Waiting for compiler upgrades

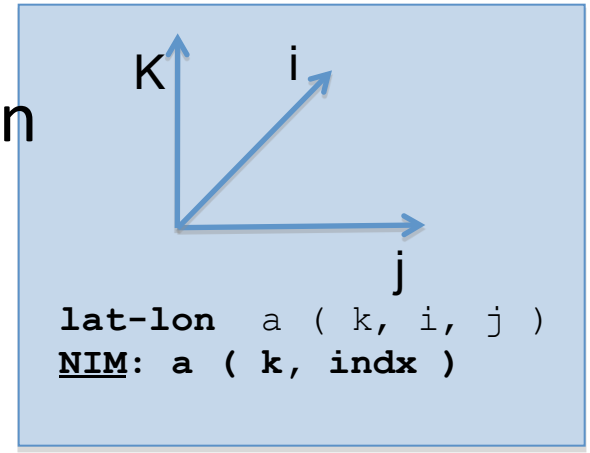
Parallel GPU performance (2011)

- Number of points / GPU is decreasing as number of GPU increase
  - Optimal performance will be 10-20K points / GPU
- Communications time remains the same



Conclusion

- Code design a dominant factor in performance
  - Data alignment led to a 10x improvement
- Data dependencies guide parallelization
- Performance tools are improving
  - Help to determine speedup potential
- Debugging is challenging
  - Potentially millions of threads to manage
- Commercial Fortran GPU compilers are maturing
  - F2C-ACC is largely a language translator, limited analysis
  - As CAPS, PGI mature, they should do more analysis and optimization and generate more efficient code



NIM Model Development

- NIM dynamics completed
  - 25x performance boost compared to CPU
    - 6x for a socket-to-socket comparison w/ fermi vs. westmere
- Physics integration in progress
  - NIM + GFS physics is working (hydrostatic scales)
  - NIM + WRF physics is in progress (non-hydrostatic)
- GPU parallelization of WRF Physics
  - PBL – in progress, working with vendors to improve their compilers
    - Legacy codes are tougher to work with
  - Radiation, Micro-physics, etc. will follow

NIM Model Testing

- Comparison between NIM + GFS, FIM + GFS
  - Request for NOAA RDHPCS WV resources
    - 30 KM resolution, 7 day forecasts run daily for 1 year
- NIM + WRF Physics @ non-hydrostatic scales
  - NOAA Climate Machine @ Oak Ridge
    - 8km resolution 1x / day, output every 12 hours, 30 day simulations
    - 4km resolution 2x / month, output every 3 hours, 7 day simulations
- NIM + GPU
  - Plans to augment our small GPU cluster significantly