

CUDA Implementation of the Weather Research and Forecasting (WRF) Model



Bormin Huang

Space Science and Engineering Center

University of Wisconsin-Madison

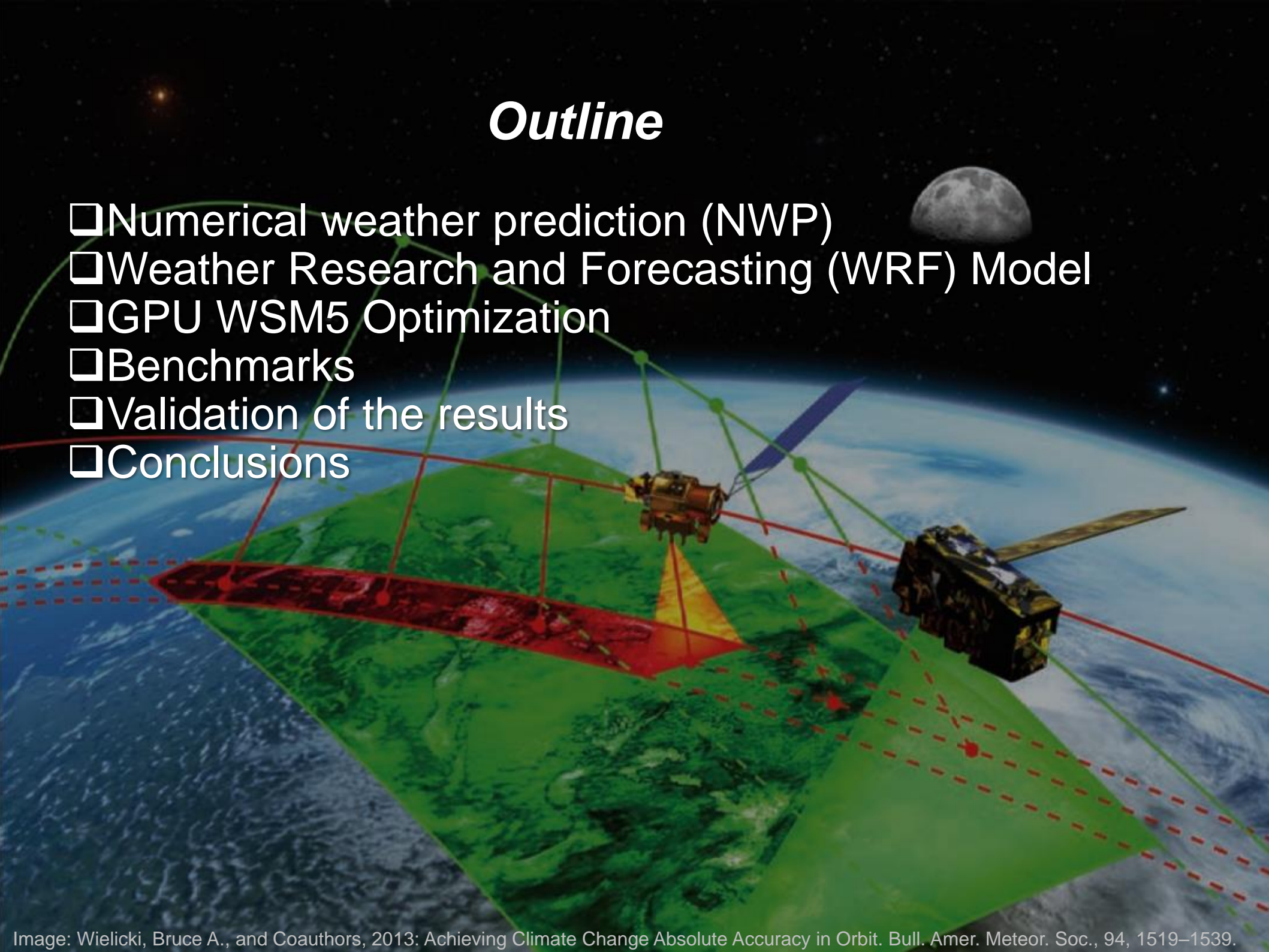


***SC13 nVIDIA Booth #613
Colorado Convention Center***



Outline

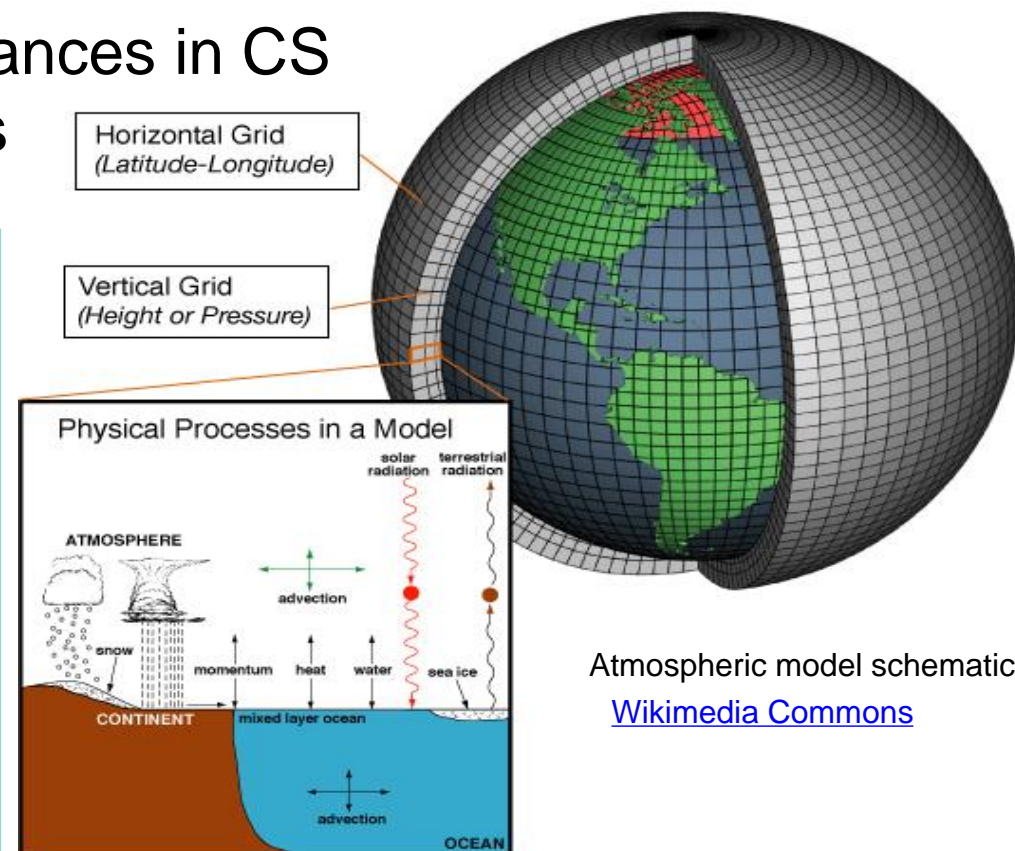
- ❑ Numerical weather prediction (NWP)
- ❑ Weather Research and Forecasting (WRF) Model
- ❑ GPU WSM5 Optimization
- ❑ Benchmarks
- ❑ Validation of the results
- ❑ Conclusions



What is numerical weather prediction (NWP)?

- Numerical weather prediction uses mathematical models of the atmosphere and oceans to predict the weather based on current weather conditions.
- First attempted in the 1920s
- Computer simulation in the 1950s -> NWP produced realistic results.
- Advances in NWP linked with advances in CS
- Major application in HPC business

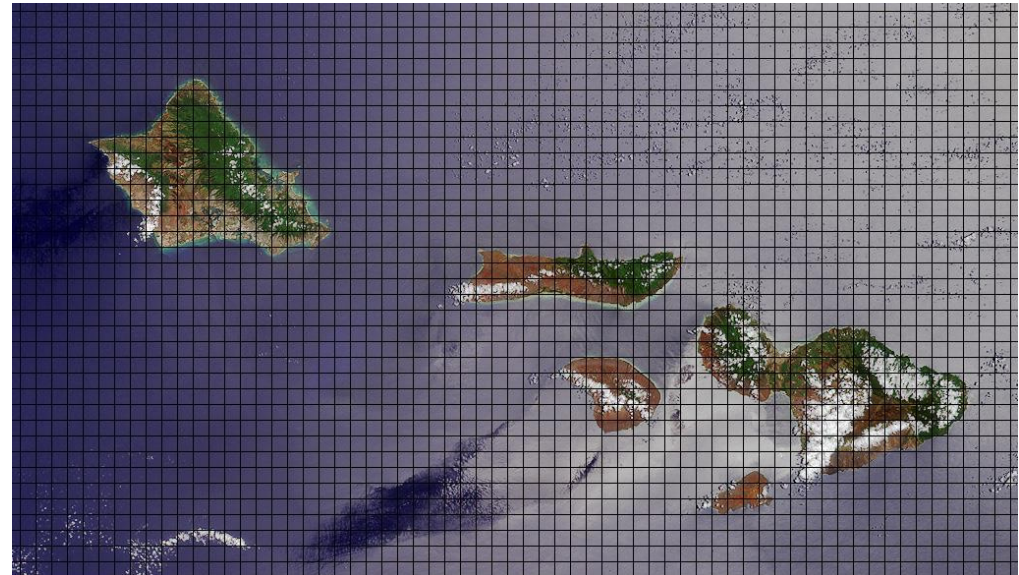
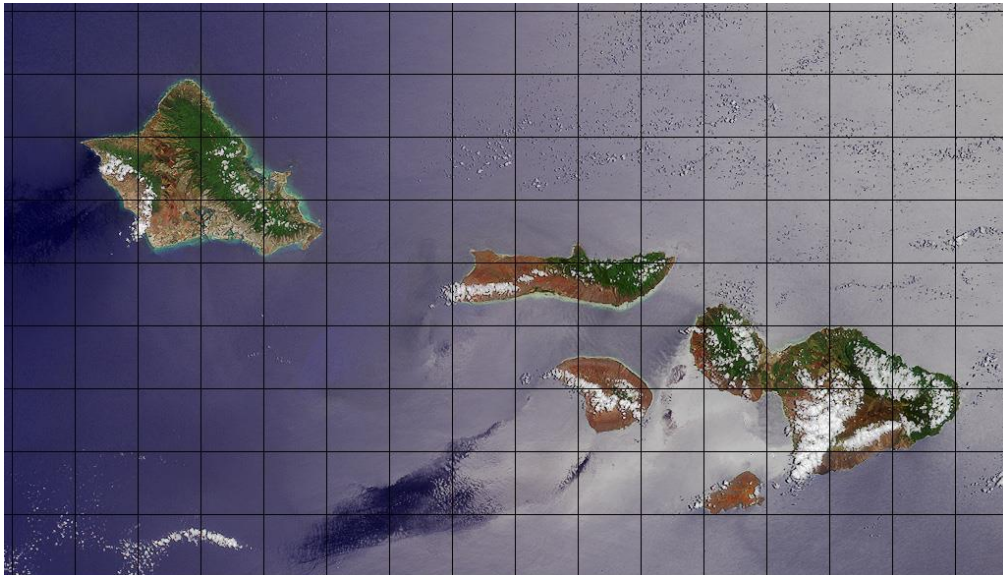
Weather models use systems of differential equations based on the laws of physics, fluid motion, and chemistry, and use a coordinate system which divides the planet into a 3D grid. Winds, heat transfer, solar radiation, relative humidity, and surface hydrology are calculated within each grid cell, and the interactions with neighboring cells are used to calculate atmospheric properties in the future.



Atmospheric model schematic
[Wikimedia Commons](#)

Grid spacing (resolution)

- Grid spacing (resolution) defines the scale of features you can simulate with the model
- “Global” vs. “regional”
 - regional = higher resolution over smaller domain




[Wikimedia Commons](#): NASA satellite photograph of the Hawaiian Islands


WRF Overview

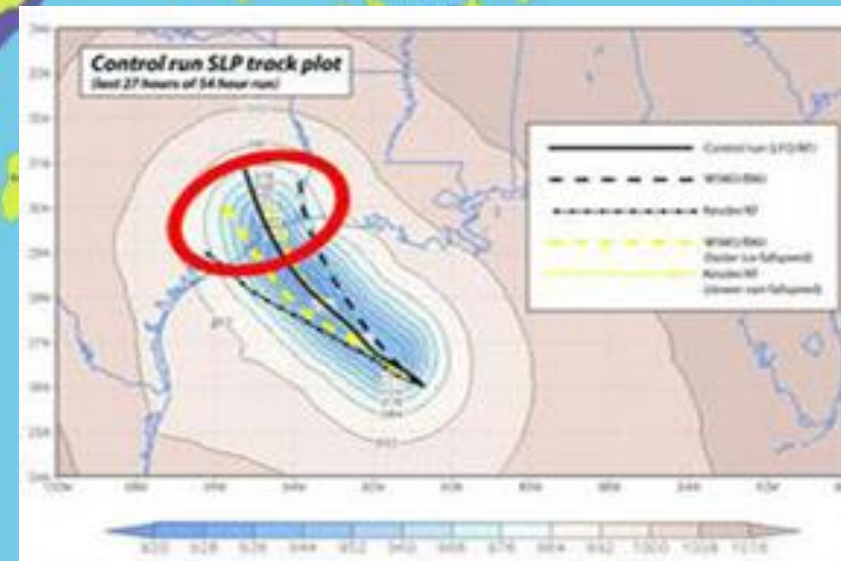
- WRF is mesoscale and global Weather Research and Forecasting model
- Designed for both operational forecasters and atmospheric researchers
- WRF is currently in operational use at numerous weather centers around the world
- WRF is suitable for a broad spectrum of applications across domain scales ranging from meters to hundreds of kilometers.

- Increases in computational power enables
 - Increased vertical as well as horizontal resolution
 - More timely delivery of forecasts
 - Probabilistic forecasts based on ensemble methods

- Why accelerators?
 - Cost performance
 - Need for strong scaling

 User Countries

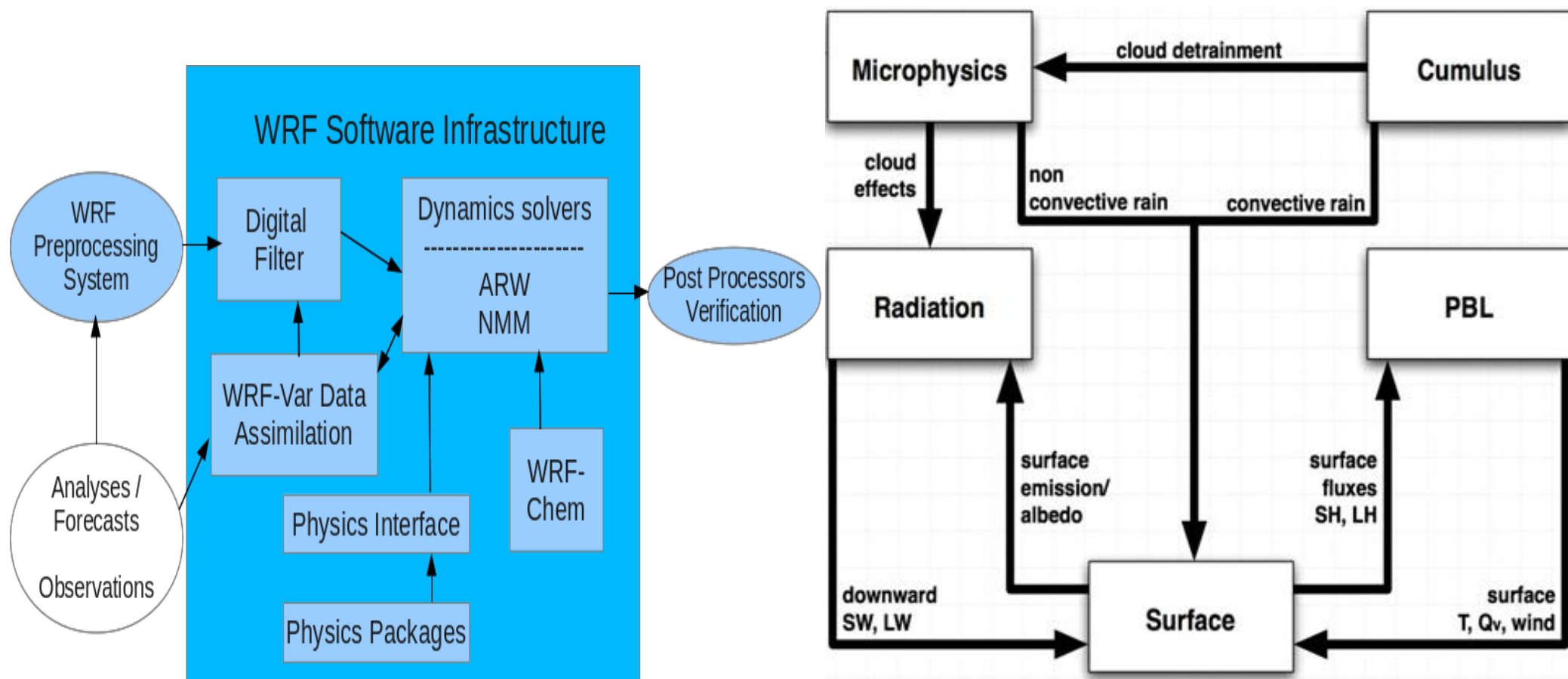
 Operational/Realtime Forecasting Countries



WRF simulation of Hurricane Rita (2005) tracks
[Wikimedia Commons](#)

153 Countries

WRF system components

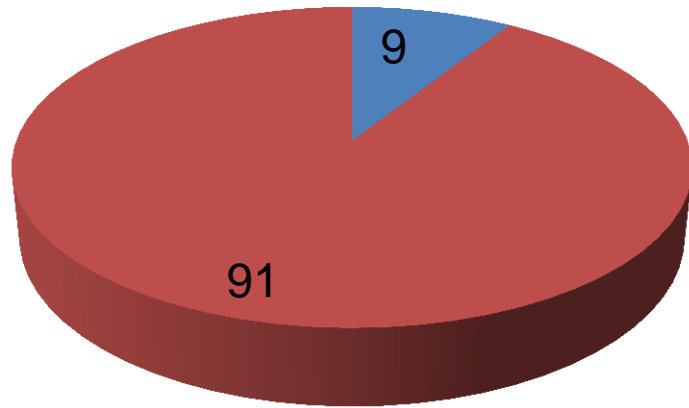


Jimmy Dudhia: WRF physics options

- The WRF physics categories are **microphysics**, cumulus parametrization, planetary boundary layer (PBL), land-surface model and radiation.

Performance Profile of WRF

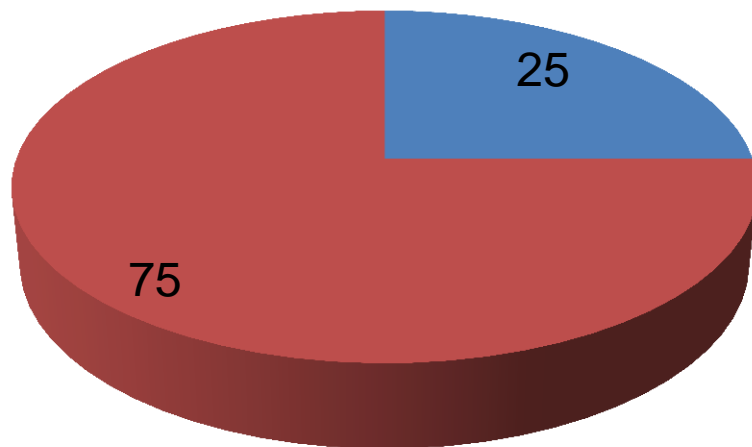
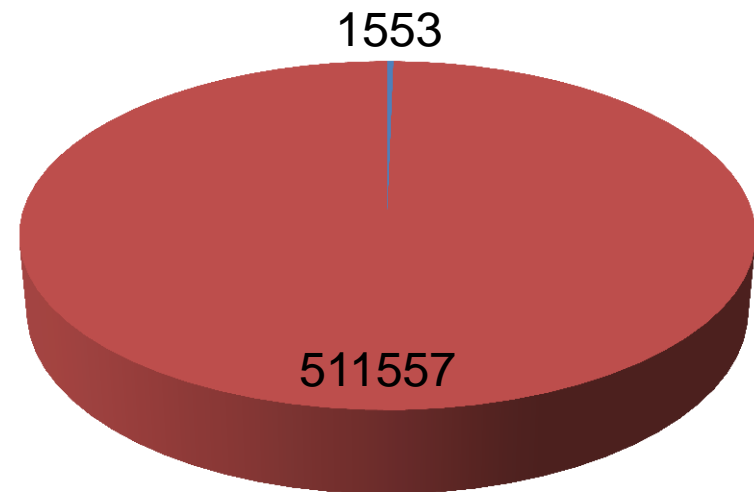
CONUS 12km workload *



% Runtime

■ WSM5
■ Others

Code lines (f90)

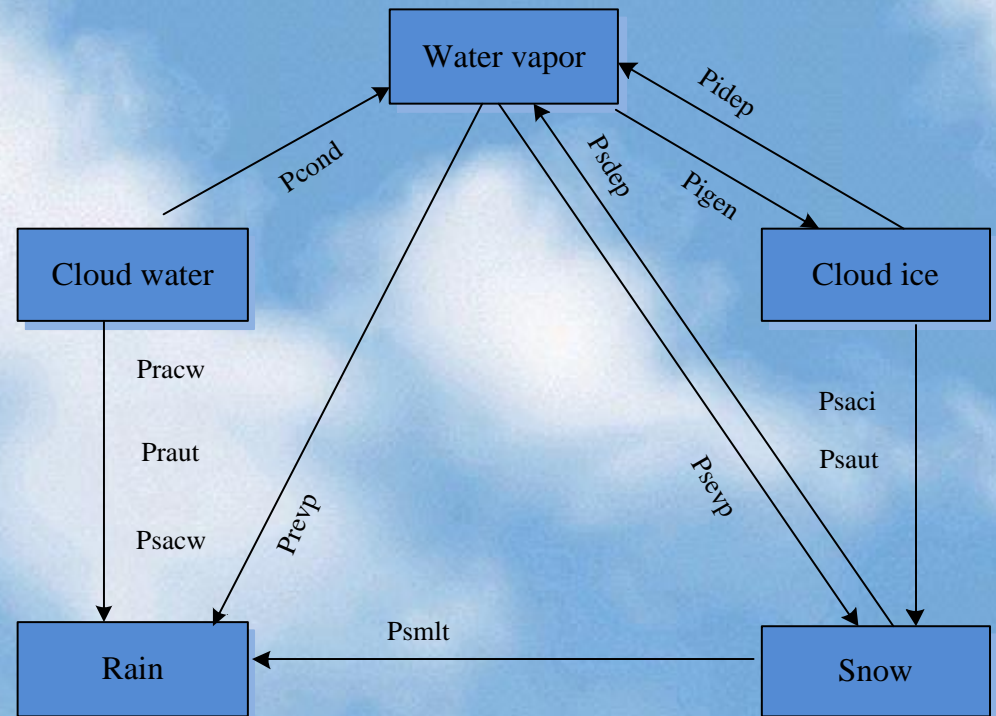


Jan. 2000, 30km workload *

* John Michalakes, "Code restructuring to improve performance in WRF model physics on Intel Xeon Phi", Workshop on Programming weather, climate, and earth-system models on heterogeneous multi-core platforms, September 20, 2013

WRF Microphysics

- Microphysics provides atmospheric heat and moisture tendencies.
- Microphysics includes explicitly resolved water vapor, cloud, and precipitation processes.
- Surface snowfall and rainfall are computed by microphysical schemes.
- Several bulk water microphysics schemes are available within the WRF, with different numbers of simulated hydrometeor classes and methods for estimating their size fall speeds, distributions and densities



Microphysics processes in the WSM5 scheme

Analyzing the WSM5 on CONUS 12 km domain

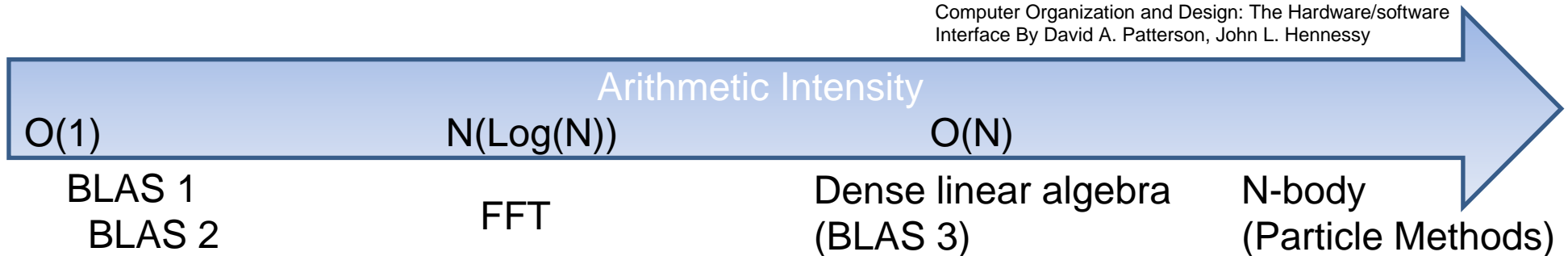
- Arithmetic intensity (=FLOPS / byte)
 - high arithmetic intensity -> computation bound
 - low arithmetic intensity -> memory bound
- WSM5 CONUS 12km workload: 24.25 billion instructions
- 7.30 billion memory reads
- 3.18 billion memory writes
 - > 0.83 instructions / byte

Measured using
cachegrind (valgrind)



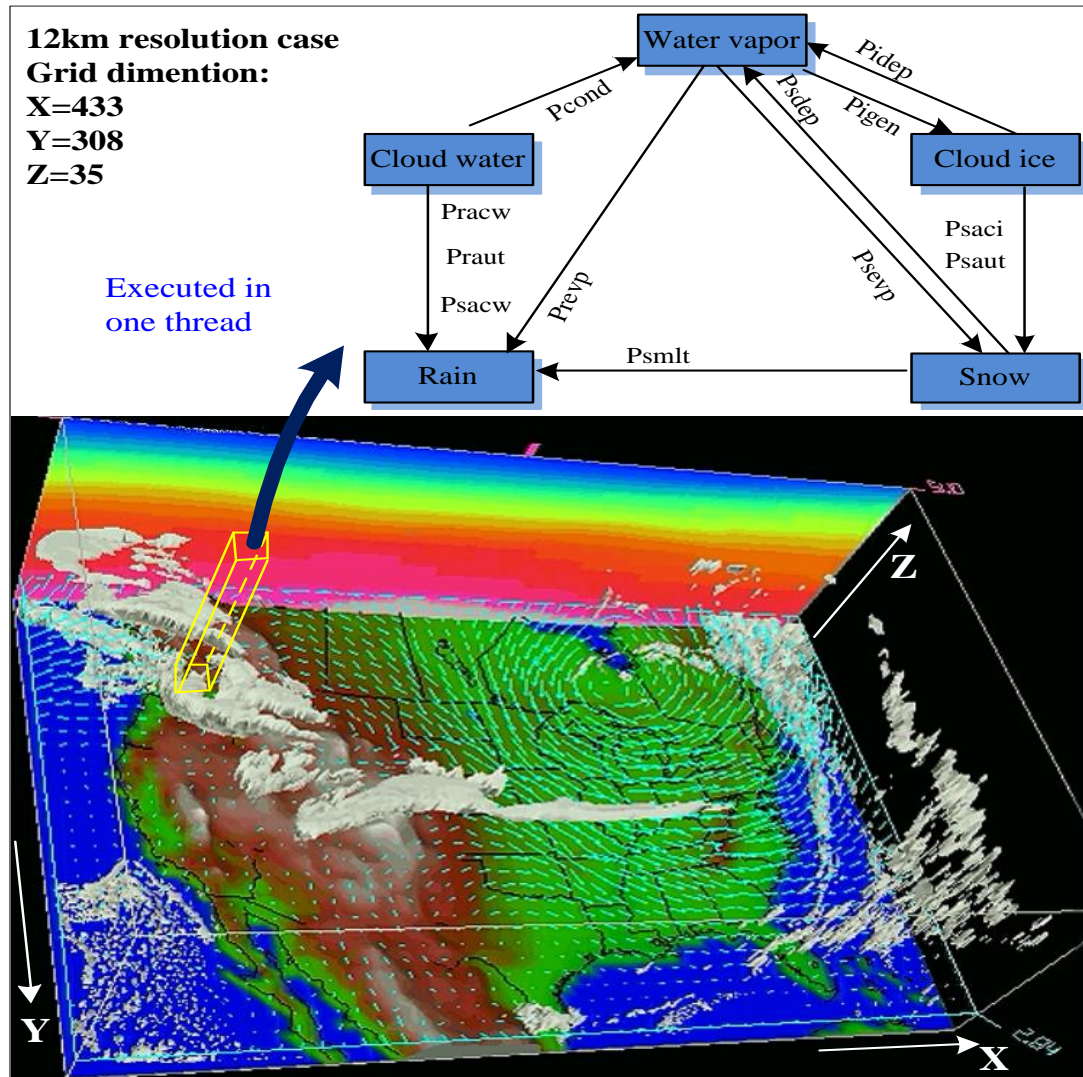
Tesla K20 delivers up to 3519 GFLOPS / 208 GB/s ~16.9 FLOPS/byte

Computer Organization and Design: The Hardware/software Interface By David A. Patterson, John L. Hennessy



Arithmetic intensity is relatively low -> reduce memory accesses

Parallelization of the computational domain



- WRF domain is 2d grid parallel to the ground
 - Multiple levels correspond to the vertical heights in the atmosphere
 - Vertical dependencies
- Columns are independent
 - Parallelizable in horizontal: two dimensions of parallelism to work with
 - Each thread computes one column at a grid point

Additional optimizations for CUDA C

Decreases processing time from 29.6 ms to 25.4 ms on K20

1. Seven additional temporaries were eliminated
2. Four additional loop fusions were performed
3. Several global arrays were prefetched from global memory to registers. Results were written back at the end of the loop.
4. Dead-code was eliminated
5. Removed computation of the same array thrice
6. After a loop-inversion, three loops were fused (2x)
7. Used `const __restrict__`* to utilize read-only cache

Analysis of WSM5 on Tesla K20

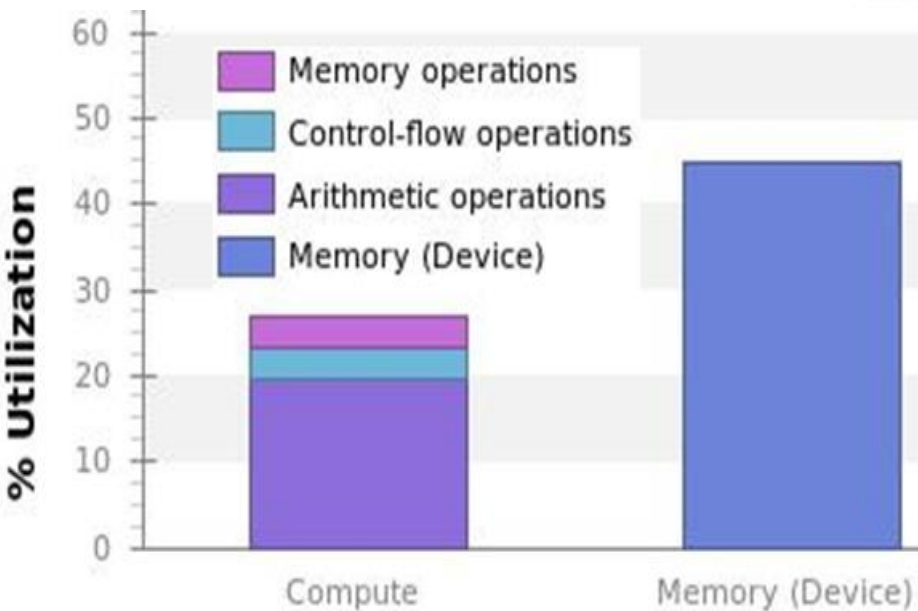
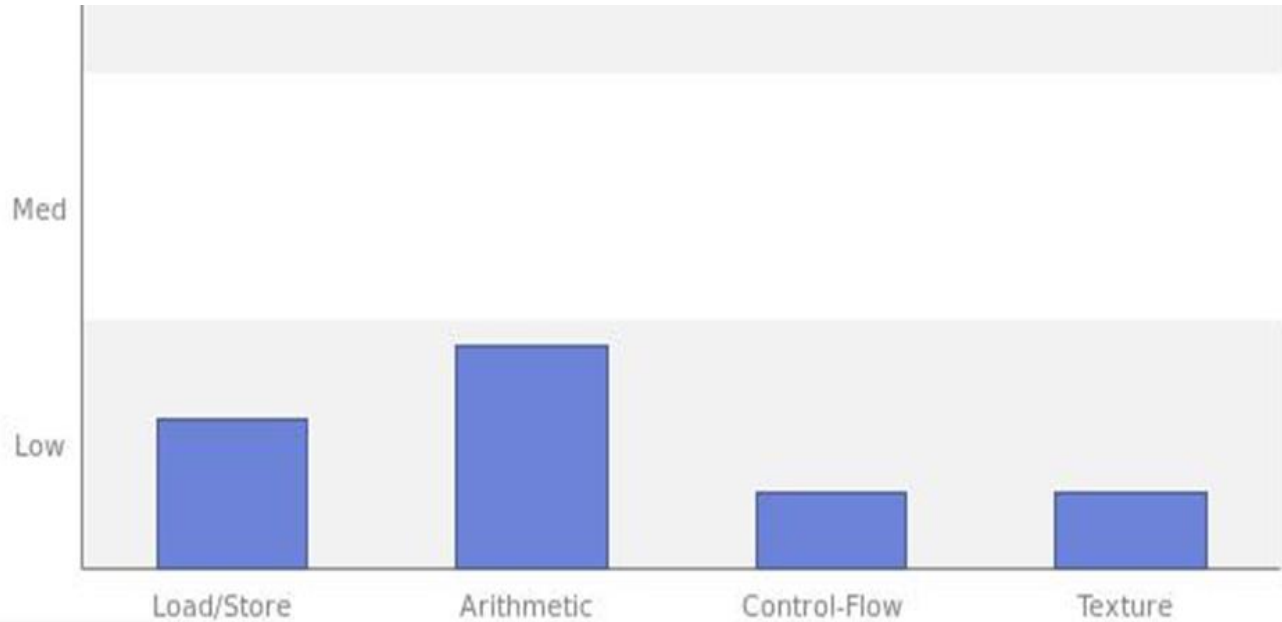
Metric Description	Old WSM5	New WSM5	
Processing time	29.6 ms	25.4 ms	14% faster
GFLOPS/s	220.5	257.0	
Registers per thread	56	62	Additional registers are used for data prefetching/temp. removal
Stack frame	0 bytes	8 bytes	
Spill stores	0 bytes	4 bytes	
Constant memory	840 bytes	784 bytes	7x64-bit pointers were removed
Achieved Occupancy	0.47	0.47	Increase in register usage didn't reduce occupancy
Executed IPC	1.17	1.30	Increased by loop fusion
L2 Hit Rate	46.18%	57.31%	Increased by temporary elimination
Texture Cache Hit Rate	53.30%	59.74%	
Global Load Transactions	25,283,839	24,217,376	Reduced by temporary elimination
Global Store Transactions	12,078,815	8,802,572	Reduced by temporary elimination
Global Load Throughput	93.9 GB/s	103.8 GB/s	

Limiting factors

Different type of instructions are executed on different function units within each SM.

Performance can be limited if a function unit is overused

Utilization Level



Achieved compute and memory bandwidth below 60% indicate latency issues

Kernel Performance is Bound by Instruction and Memory Latency

Benchmarking GPUs

GPU	Core clock	CUDA cores	Peak Single Precision Processing Power	Peak Double Precision Processing Power	Memory Bandwidth (ECC off)	Total Memory Size
Tesla K20 (Nov. 2012)	705 MHz (758 MHz *)	2496	3519 GFLOPS	1173 GFLOPS	208 GB/s	5 GB
Tesla K40 (Nov. 2013)	745 MHz (875 MHz *)	2880	3837 GFLOPS	1279 GFLOPS	288 GB/s	12 GB

- NVIDIA GPU Boost is a feature that makes use of the power headroom to run the SM clock to a higher frequency.
- The default clock is set to the base clock, which is necessary for some applications that are demanding on power (e.g., DGEMM), many application workloads are less demanding on power and can take advantage of a higher boost clock setting for added performance.

Memory Bandwidth and Utilization

K40 Base Mode

	Transactions	Bandwidth
L1/Shared Memory		
Local Loads	0	0 B/s
Local Stores	0	0 B/s
Shared Loads	0	0 B/s
Shared Stores	0	0 B/s
Global Loads	24217376	153.68 GB/s
Global Stores	8802572	50.49 GB/s
L1/Shared Total	33019948	204.17 GB/s
Texture Cache		
Reads	23942221	38.08 GB/s
L2 Cache		
Reads	59361222	94.2 GB/s
Writes	31836901	50.49 GB/s
Total	91198123	144.69 GB/s
Device Memory		
Reads	47782746	75.78 GB/s
Writes	29572926	46.91 GB/s
Total	77355672	122.68 GB/s

K40 Boost Mode

	Transactions	Bandwidth
L1/Shared Memory		
Local Loads	0	0 B/s
Local Stores	0	0 B/s
Shared Loads	0	0 B/s
Shared Stores	0	0 B/s
Global Loads	24217376	176.23 GB/s
Global Stores	8802572	57.92 GB/s
L1/Shared Total	33019948	234.15 GB/s
Texture Cache		
Reads	23978703	43.12 GB/s
L2 Cache		
Reads	59307590	107.94 GB/s
Writes	31836907	57.92 GB/s
Total	91144497	165.86 GB/s
Device Memory		
Reads	47765532	86.9 GB/s
Writes	29558128	53.77 GB/s
Total	77323660	140.67 GB/s

Nvidia K40 vs. Xeon Phi

	Xeon Phi *	Tesla K40
Processing Time	29.7 ms	16.5 ms
Concurrent CUDA threads	3840 (60 cores, 4 HT, 16 SIMD)	14336 (28 warps/MP, 16 MPs)
Vector Instructions	49.73%	100%
DRAM Write Throughput	33.5 GB/s	57.7 GB/s
DRAM Read Throughput	19.0 GB/s	93.3 GB/s

- Xeon Phi vectorized 1/2 of WSM5 – the other half utilizes only multiple cores
- Xeon Phi with a higher cache size/number of threads ratio can serve more memory requests from caches than K40
- K40 is able to hide latency better even with a higher usage of global memory than Xeon Phi
 - a larger number of concurrent threads allows for better latency hiding

* Xeon Phi optimization: John Michalakes, NOAA

Additional Optimization: I. Gokhale, L. Meadows, R. Sasanka, Intel Corp.

WSM5 Microphysics

WSM5 CONUS 12KM Workload,
6.53 GF/call (Intel SDE)
(www.mmm.ucar.edu/wrf/WG2/bench)

395.6

345.3

270.8

257

219.7

195.4

122.7

GF/s

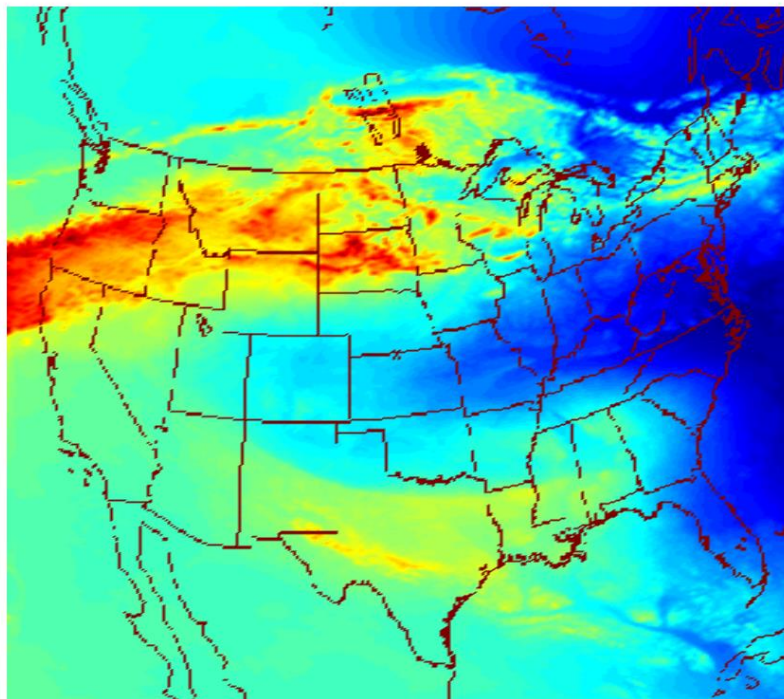
Higher is better

Processor	GF/s
INTEL Xeon Sandybridge-EP (2x8 core) by INTEL*	122.7
INTEL Xeon Ivybridge-EP (2x12 core) by INTEL*	195.4
INTEL Xeon Phi (240 T) by INTEL*	219.7
NVIDIA Kepler K20 Base Mode by UW/SSEC	257
NVIDIA Kepler K20 Boost Mode by UW/SSEC	270.8
NVIDIA Kepler K40 Base Mode by UW/SSEC	345.3
NVIDIA Kepler K40 Boost Mode by UW/SSEC	395.6

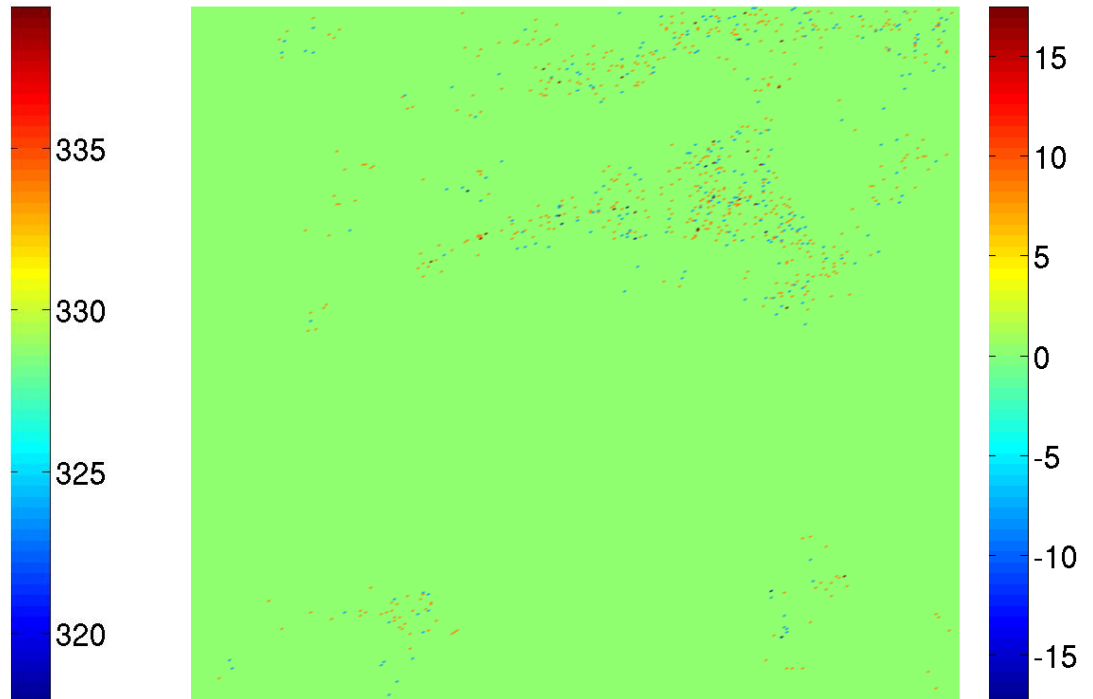
* Code Restructuring to Improve Performance in WRF Model Physics on Intel Xeon Phi. J. Michalakes. Workshop on Programming Weather, Climate and Earth System Models on Heterogeneous Multi-core Platforms, Boulder, Colorado, Sept. 19-20, 2013. (http://data1.gfdl.noaa.gov/multi-core/presentations/michalakes_5.pdf)

Code Validation

- Fused multiply-addition was turned off (--fmad=false)
- GNU C math library was used on GPU, i.e. powf(), expf(), sqrt() and logf() are replaced by library routines from GNU C library
-> bit-exact output
- Small output differences for -fast-math

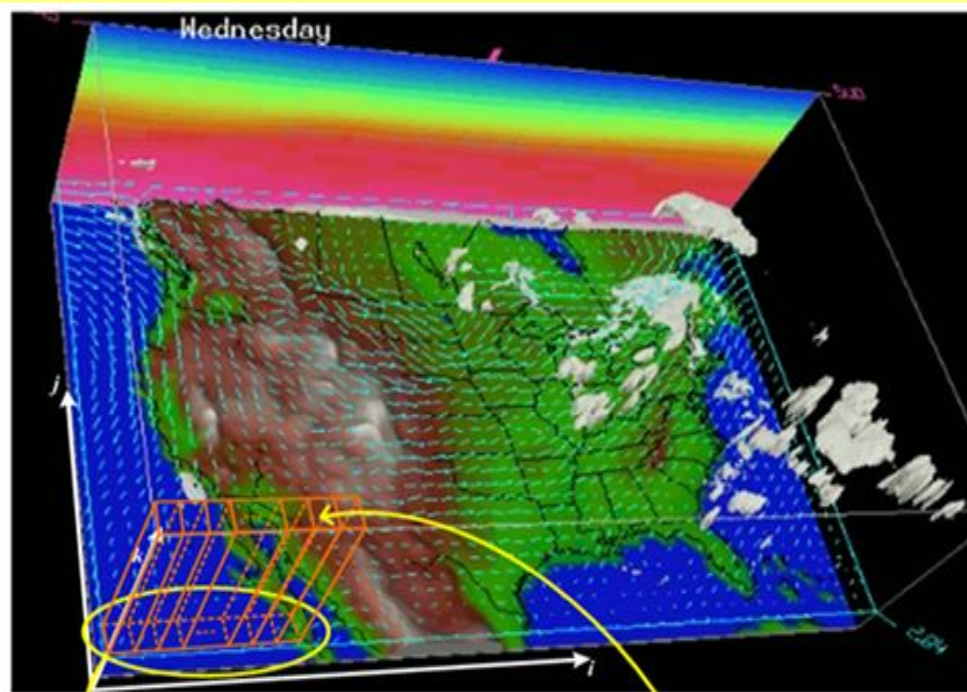


Potential temperature



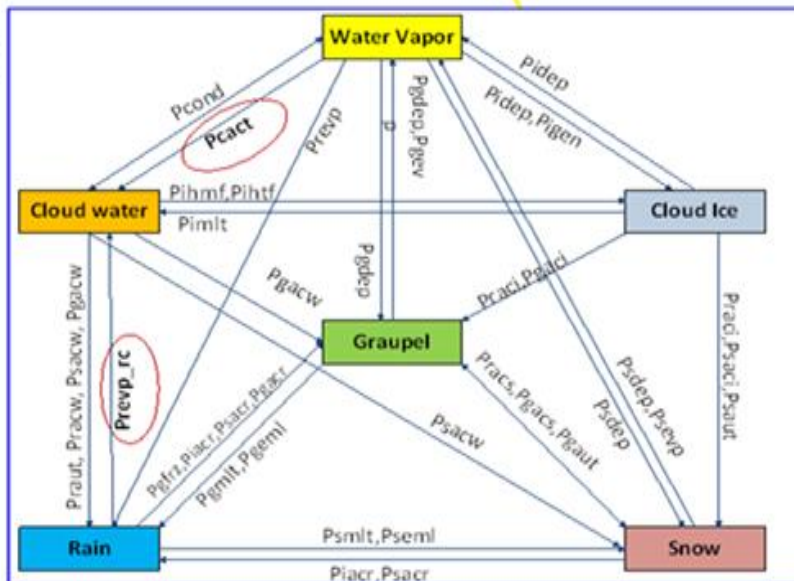
Difference between CPU and GPU outputs

GPU-accelerated WRF modules



Blockdim(64, 1, 1);

i dim = 433
 j dim = 308
 k dim = 35



WRF Module name	Speedup
Single moment 6-class microphysics	500x
Eta microphysics	272x
Purdue Lin microphysics	692x
Stony-Brook University 5-class microphysics	896x
Betts-Miller-Janjic convection	105x
Kessler microphysics	816x
New Goddard shortwave radiance	134x
Single moment 3-class microphysics	331x
New Thompson microphysics	153x
Double moment 6-class microphysics	206x
Dudhia shortwave radiance	409x
Goddard microphysics	1311x
Double moment 5-class microphysics	206x
Total Energy Mass Flux surface layer	214x
Mellor-Yamada Nakanishi Niino surface layer	113x
Single moment 5-class microphysics	350x
Pleim-Xiu surface layer	665x

Conclusions

- Great interest in the community in accelerators
- Continuing work on accelerating other WRF modules using CUDA C (~20 modules finished)
- Lessons learned during CUDA C implementation of WSM5 could be applied to OpenACC/OpenMP 4.0 optimization of WRF modules

Acknowledgement

Co-authors, Sponsors and Participants:

- ❖ Jarno Mielikainen, SSEC, Wisconsin-Madison
- ❖ Melin Huang, SSEC, Wisconsin-Madison
- ❖ Allen Huang, SSEC, Wisconsin-Madison

- ❖ Mitchell Goldberg, NOAA
- ❖ Ajay Mehta, NOAA

- ❖ Stan Posey, NVIDIA