GPU Acceleration of the WRF Model for Time-Critical Satellite Data Assimilation Applications

Jarno Mielikainen, Bormin Huang (bormin@ssec.wisc.edu), Allen Huang

Space Science and Engineering Center, University of Wisconsin–Madison

Introduction

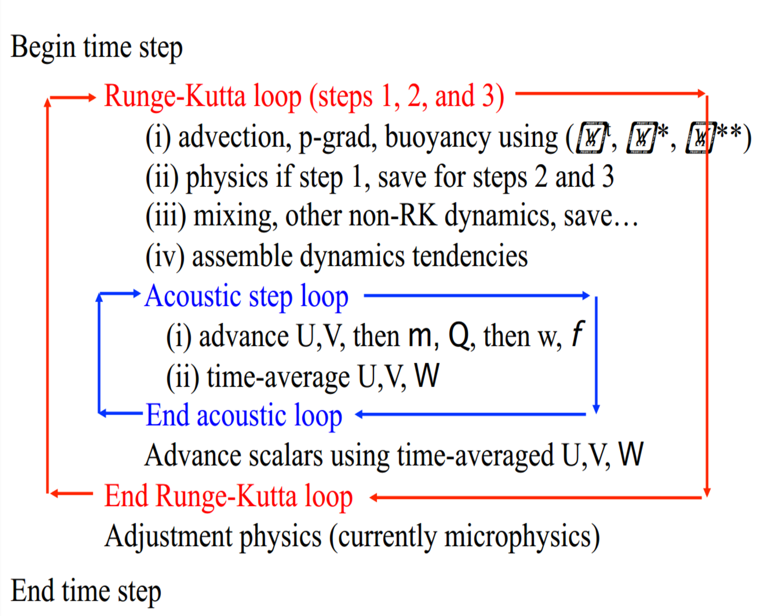
- •WRF is mesoscale and global Weather Research and Forecasting model
- •Designed for both operational forecasters and atmospheric researchers
- •WRF is currently in operational use at numerous weather centers around the world
- •WRF is suitable for a broad spectrum of applications across domain scales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

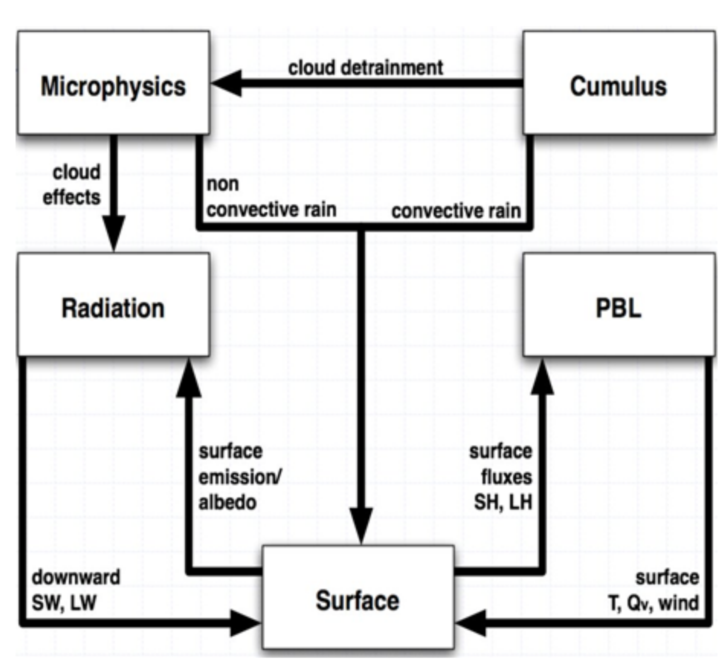
Operational/Realtime Forecasting Countries

153 Countries

WRF System Components



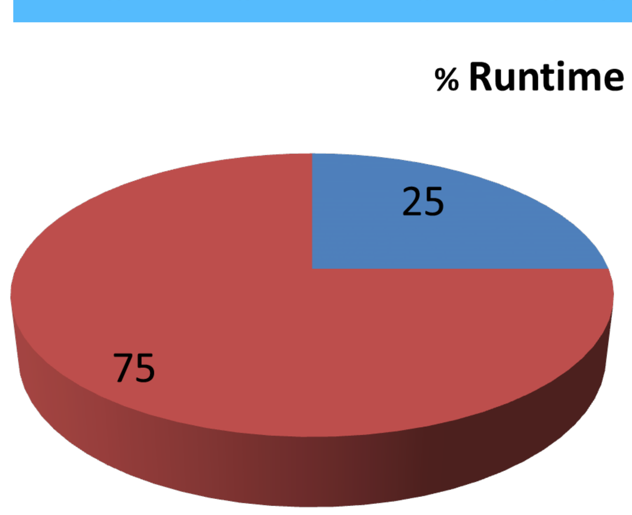
WRF model integration procedure



Jimy Dudhia: WRF physics options

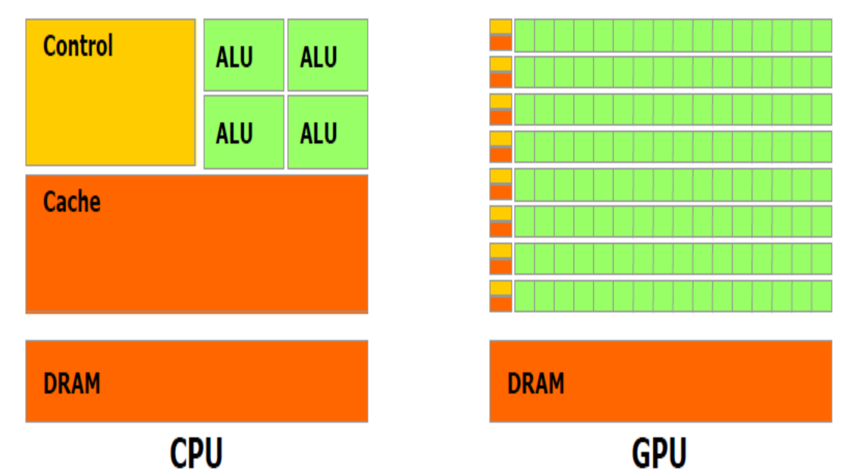
The WRF physics components are **microphysics**, cumulus parametrization, planetary boundary layer (PBL), land-surface model and shortwave/longwave radiation.

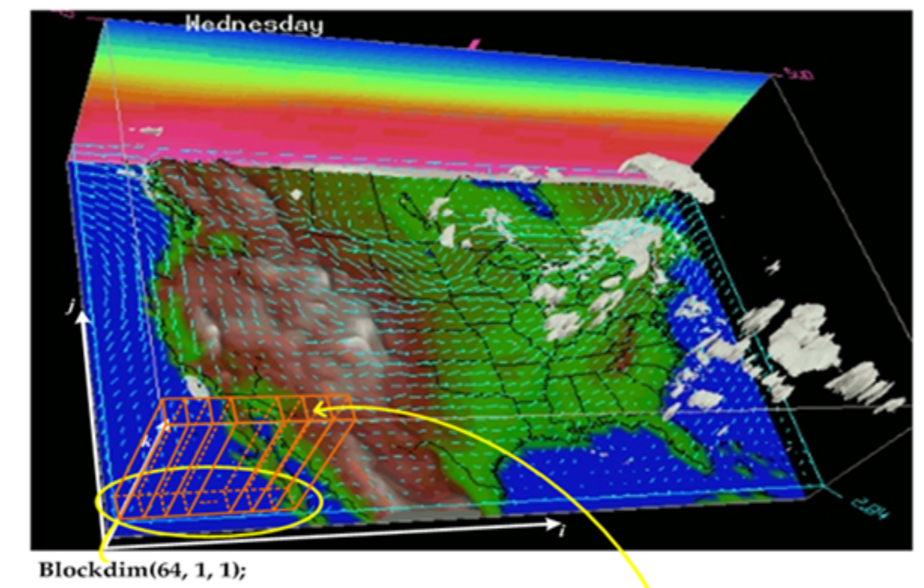
Performance Profile of WRF

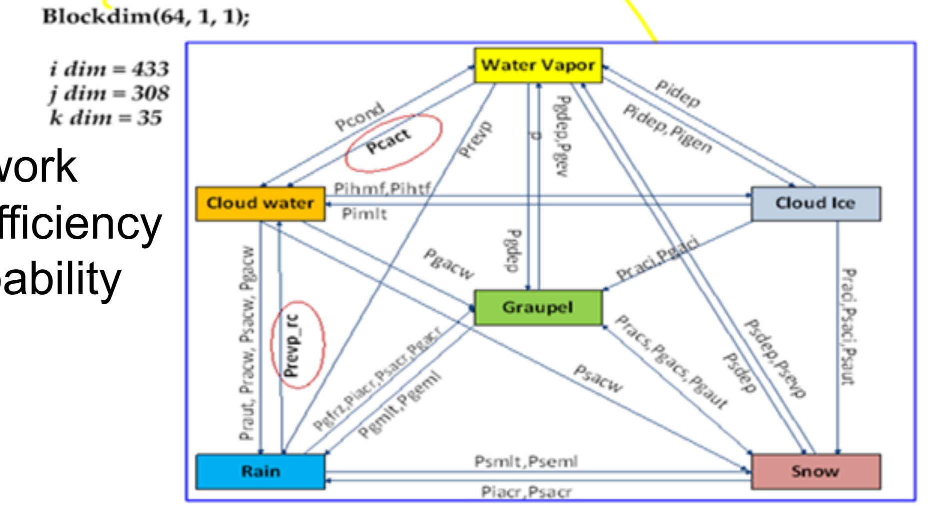


Jan. 2000 30km workload profiling by John Michalakes, "Code restructuring to improve performance in WRF model physics on Intel Xeon Phi", Workshop on Programming weather, climate, and earth-system models on heterogeneous multi-core platforms, September 20, 2013

Parallel Execution of WRF on GPU







- GPUs have more cores than CPUs
- GPUs are optimized for data parallel work
- GPU needs 1000s of threads for full efficiency
- GPUs have a higher floating-point capability than CPUs

CPUs focus on per-core performance - **Sandy Bridge**: 4 cores, **115.2 Gflops**, 90 Watts **(~1.4 GFlop /Watt),** Memory Bandwidth: **34.1 GB/s**

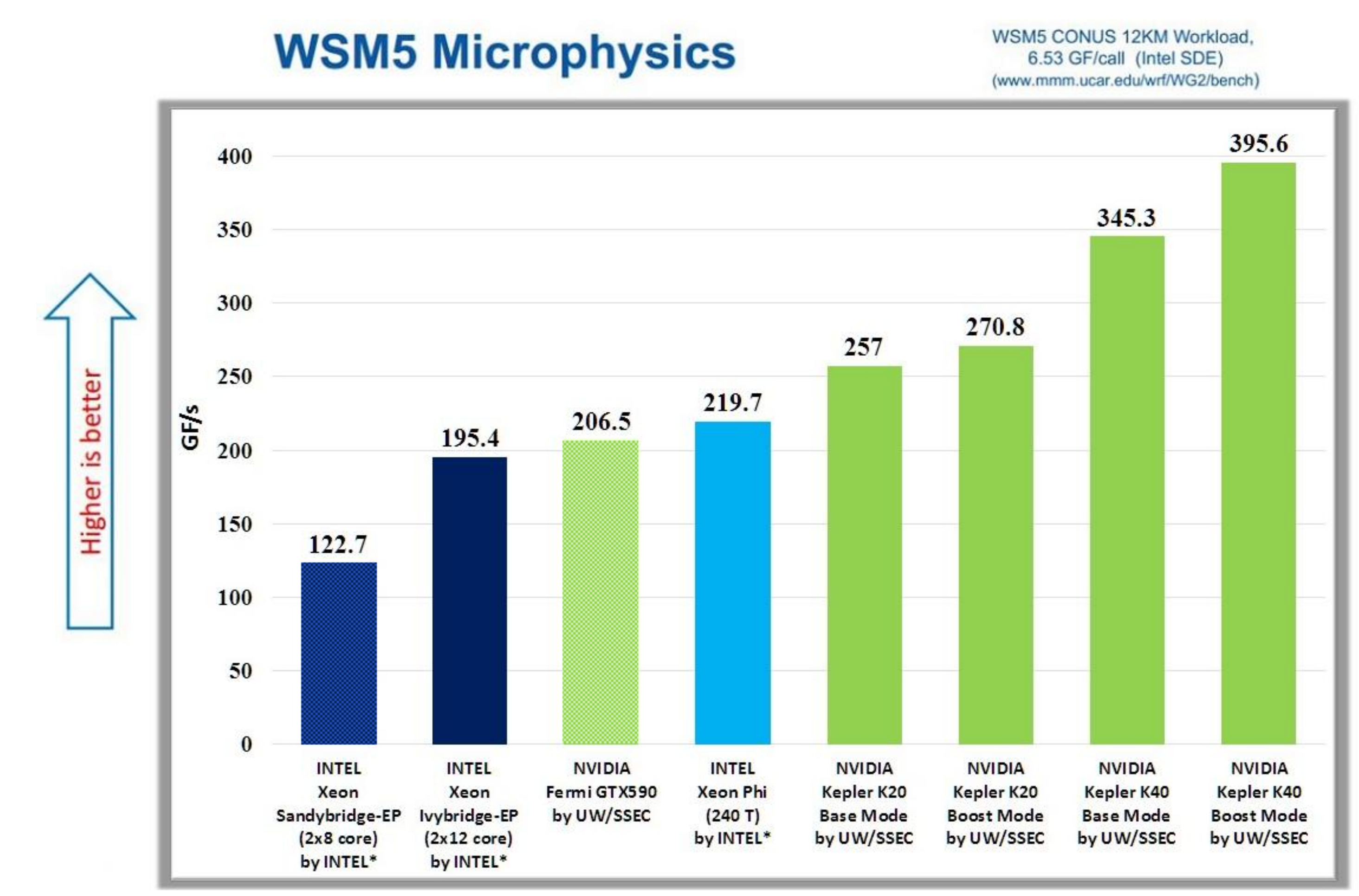
GPUs focus on parallel execution - **Tesla K40**: 2,880 cores, **4290 Gflops (Peak SP)**, 1430 Gflops (Peak DP), 235 Watts **(~18.3 Gflops / Watt),** Memory bandwidth: **288 GB/s**

GPU Speedups

Single threaded non-vectorized CPU code is compiled with gfortran 4.4.6

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

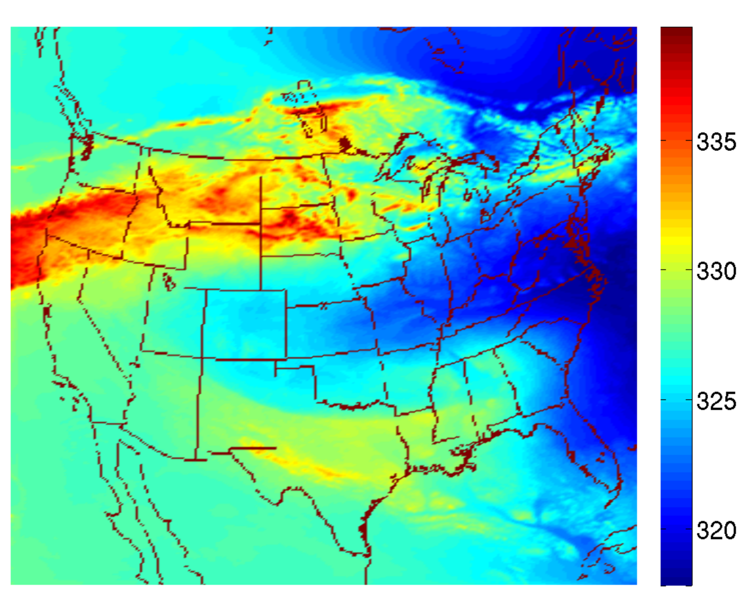
Performance Comparison

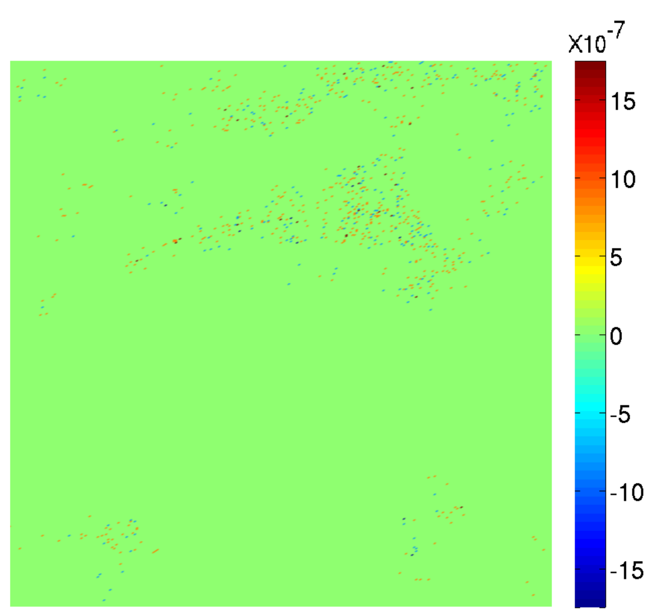


* *Code Restructuring to Improve Performance in WRF Model Physics on Intel Xeon Phi.* J. Michalakes. Workshop on Programming Weather, Climate and Earth System Models on Heterogeneous Multi-core Platforms, Boulder, Colorado, Sept. 19-20, 2013. (http://data1.gfdl.noaa.gov/multi-core/presentations/michalakes_5.pdf)

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() were replaced by routines from GNU C library
- -> bit-exact output compared to gfortran compiler on CPU
- -Small output differences for –fast-math (shown below)





Potential temperatures [K] Difference between CPU and GPU outputs

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 will be applied to OpenACC 2.0/OpenMP 4.0 optimization of WRF modules for GPU/Intel MIC