Performance-related developments in WRF

John Michalakes
NOAA/EMC (I.M. Systems Group)

Mike Iacono, David Berthiaume
AER

Indraneil Gokhale
Intel Corp.

WRF Users Workshop

June 24, 2014

Outline

- Accelerators: GPU and MIC
- WRF software trends for HPC
- Conclusions

Accelerators

- Graphics Processing Units (GPUs)
- Intel Xeon Phi (Many Integrated Core, or “MIC”)
- Roughly the same:
  - cost (energy and dollars)
  - performance potential (~teraflop peak)
- Different programming models and approaches to parallelism
  - “Bare metal” programming with CUDA on GPU
  - Programming directives and Fortran
    - OpenMP and Vector directives on MIC
    - OpenACC on GPU
  - All approaches require restructuring of loops and data structures for performance, *which may also benefit code on host processor*

Accelerating NWP

- NOAA
  - NCEP
    - Adapting operational models, starting with NMM-B
    - Detailed analysis and performance modeling of NWP kernels
  - ESRL *(Govett, Henderson, Rosinksi, Middlecoff)*
    - OpenACC parallelization of WSM5 and YSU PBL on GPU
    - Adding MPAS physics to NIM dynamics on MIC
    - Leading charge on “single source” implementations for GPU & MIC
- NCAR *(Loft, Dennis, et al)*
  - New Intel Parallel Computing Center (IPCC) with U. Colorado
  - Ongoing accelerator work on WRF, MPAS, CESM
- U. Wisc. SSEC *(Huang, Mielikainen, et al)*
  - CUDA implementations of many WRF kernels for GPU
  - Also awarded an Intel Parallel Computing Center
- AER *(Iacono, Berthiaum)*
  - CUDA Fortran and OpenACC implementations of RRTMG on GPU
- Many others
  - NCAR Multi-core Workshop series: **http://data1.gfdl.noaa.gov/multi-core**

Accelerating RRTMG Radiation Physics

- AER Development of RRTM*GPU*
  - Originally funded by NASA for GEOS-5
  - DOE Climate Modeling SciDAC Program funding application to WRF
  - RRTMGPU_LW and SW implemented in WRF_v3.51 and testing in progress on NCAR Caldera
- NOAA/EMC
  - Porting and optimization of RRTMG in NMM-B and GFS to Intel MIC
  - MIC-restructured code ran 1.3X faster on host Xeons
- Apples-to-apples comparisons
  - Ported GPU version of AER's shortwave code to MIC
    - Converted OpenACC threading directives to OpenMP
    - We permuted the loop ordering to favor vectorization on MIC
  - Benchmarked an NMMB-like workload (from 4KM CONUS)

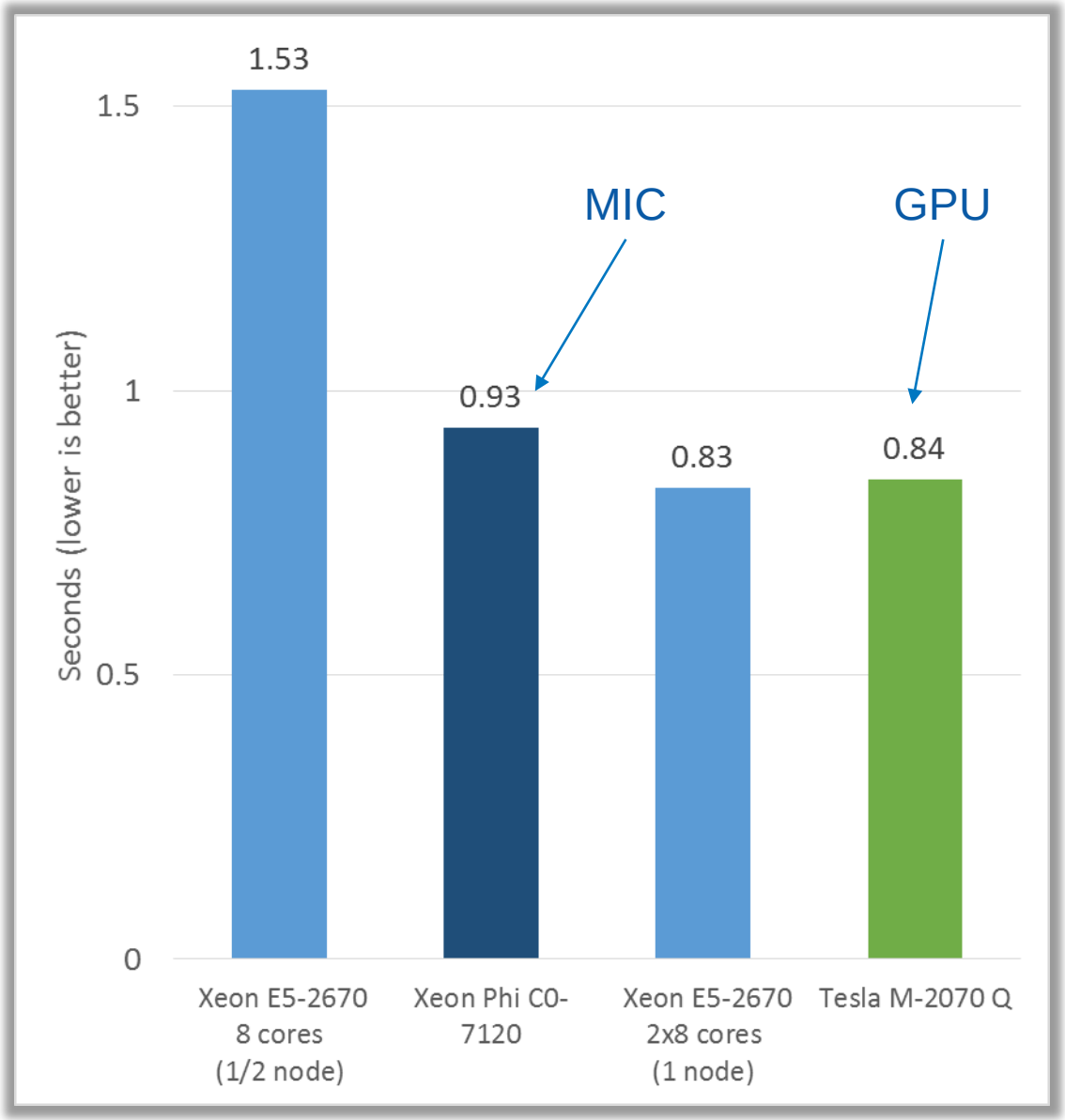
RRTMG Shortwave Performance

Test workload

18819 columns, 60 levels

Elapsed time on Xeon, MIC and GPU

(Note: Xeon and GPU are not newest versions of vendor hardware)

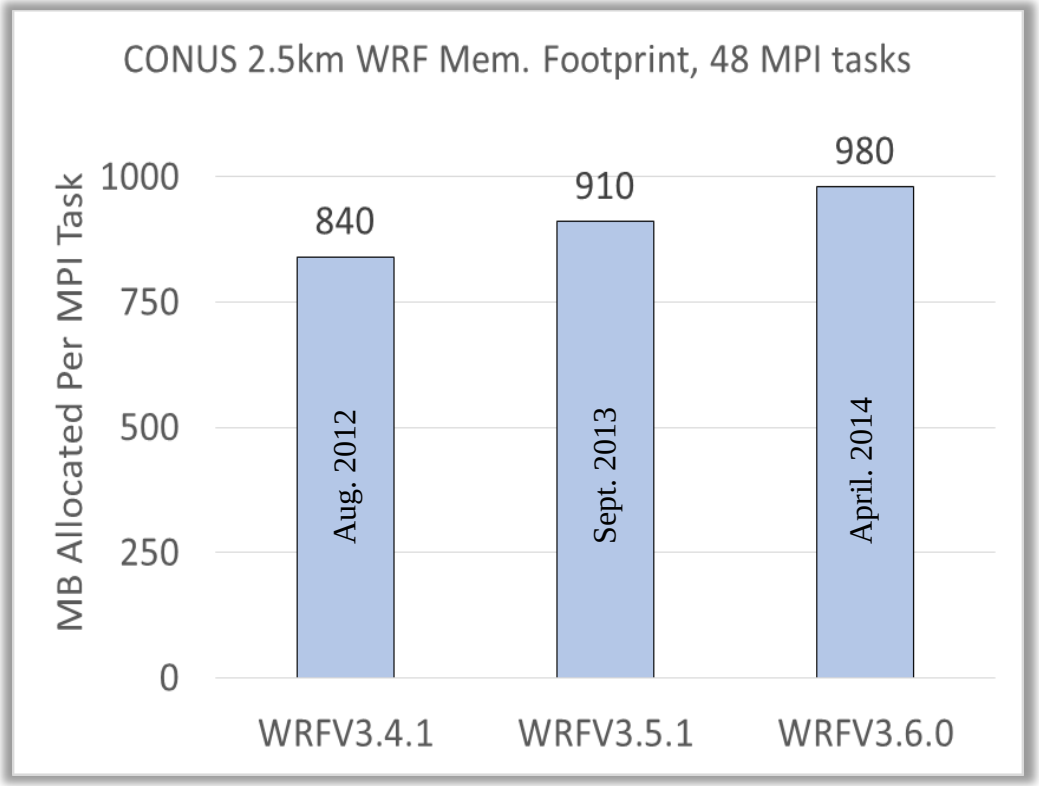


Accelerators: Summary

- Neither accelerator is living up to its name
  - Latency-bound: large per-thread memory footprint exceeds caches, accesses spill to memory, floating point ops stall
  - Likely similar effects on GPU
- Lesson for NWP on next generation architectures:
  - Wait for *next* next generation....
    - NERSC-8 “Cori” System – 9300 hostless MIC (Knights Landing) nodes
    - https://www.nersc.gov/users/computational-systems/nersc-8-system-cori/
  - Engineer codes for
    - Concurrency
    - Fine-grained parallelism
    - **Leaner memory footprint per thread**
- How’s WRF doing?

WRF Software Trends

- Three WRF releases
  - **Memory requirements**



SINCE 2012

8% Growth per year
(doubling time = 8.5 years)

21 more 3D arrays
47 more 2D arrays

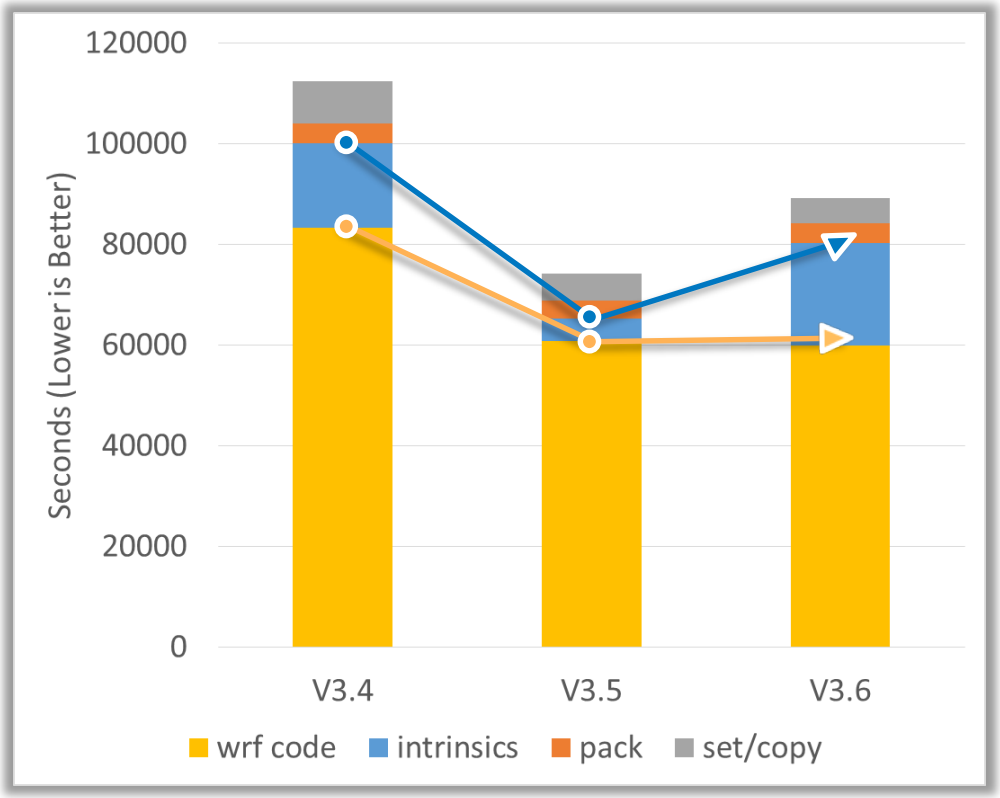
plus 2 special purpose arrays:

urb_param ≡ **4** 3D arrays
obs_savwt ≡ **10** 3D arrays

Most of this is from new features, some of which could be but have not been "packaged" in the registry

WRF Software Trends

- Three WRF releases
  - Memory requirements
  - Performance
  - **What's happening?**
    - **Intel-contributed Vectorization improvements from v3.4 to v3.5, especially for intrinsics (log and power)**
    - **Much of this appears to have been lost from v3.5 to v3.6**



Profiles of where time is spent (gprof)

Summary: evolving to next-gen HPC

- Accelerators show promise but too little return on current hardware
- Prepping applications for next gen. hardware underway
  - Increasing concurrency
  - Increasing vectorization
  - Decreasing memory footprint
- More attention needs to be paid to fine-grained parallelism going forward
- WRF is heading in the wrong direction on memory use but no clear evidence it's hurting performance **... yet**

Recommend

- Further study
- Test for performance and resource consumption
- Consider requirements for software redesign for scalability