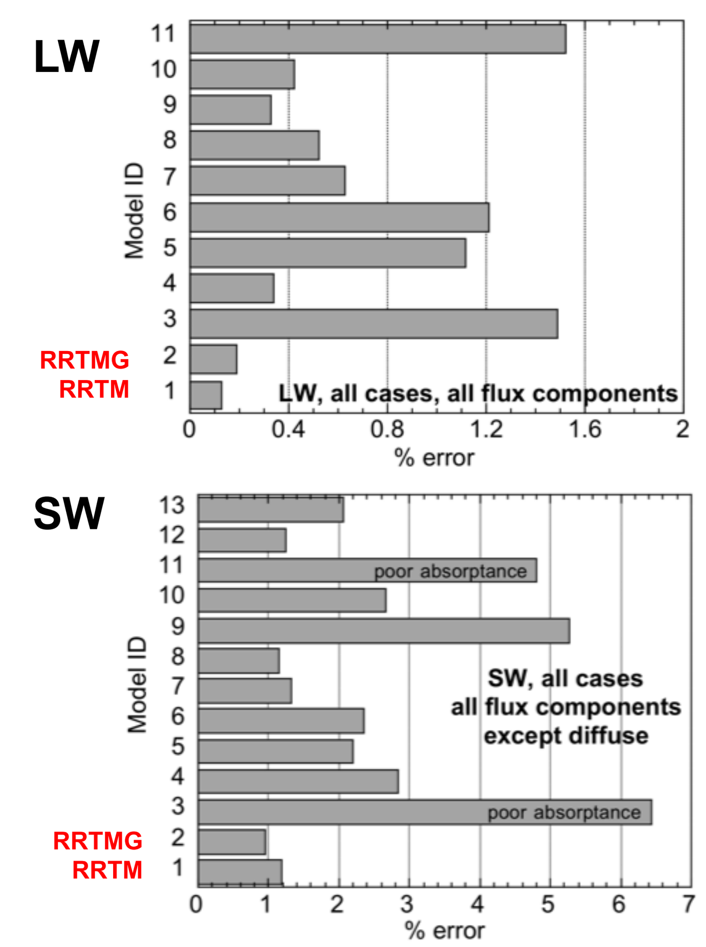
Enhancing Efficiency of the RRTMG Radiation Code with GPU and MIC Approaches for Numerical Weather Prediction Models

Michael J. Iacono[1], David Berthiaume[1], John Michalakes[2]

(1) Atmospheric and Environmental Research (miacono@aer.com) (2) NOAA – National Centers for Environmental Prediction

RRTMG, Radiative Transfer Model for GCMs

- Accurate calculations of radiative fluxes and cooling rates are key to accurate simulations of climate and weather in GCMs
- Radiative transfer (RT) calculations in GCMs constitute a significant fraction of the model's computations
  - As much as **30-50%** of execution time
- **RRTMG** is an accurate and fast RT code relative to RRTM, LBLRTM and measurements
  - (Iacono et al., JGR, 2008; Mlawer et al., JGR, 1997)
  - Available at rtweb.aer.com
- RRTMG is used in many dynamical models:
  - **WRF-ARW**: LW and SW implemented as physics options in v3.1 in 2009
  - NCAR CAM5 and CESM1 (LW in 2010, SW in 2010)
  - NASA GEOS-5 RRTMG to be next operational RT code
  - NCEP GFS (2003,2010), CFS (2004,2010), RUC (2008)
  - ECMWF IFA (2000,2007) and ERA40
  - ECHAM5 (2002)

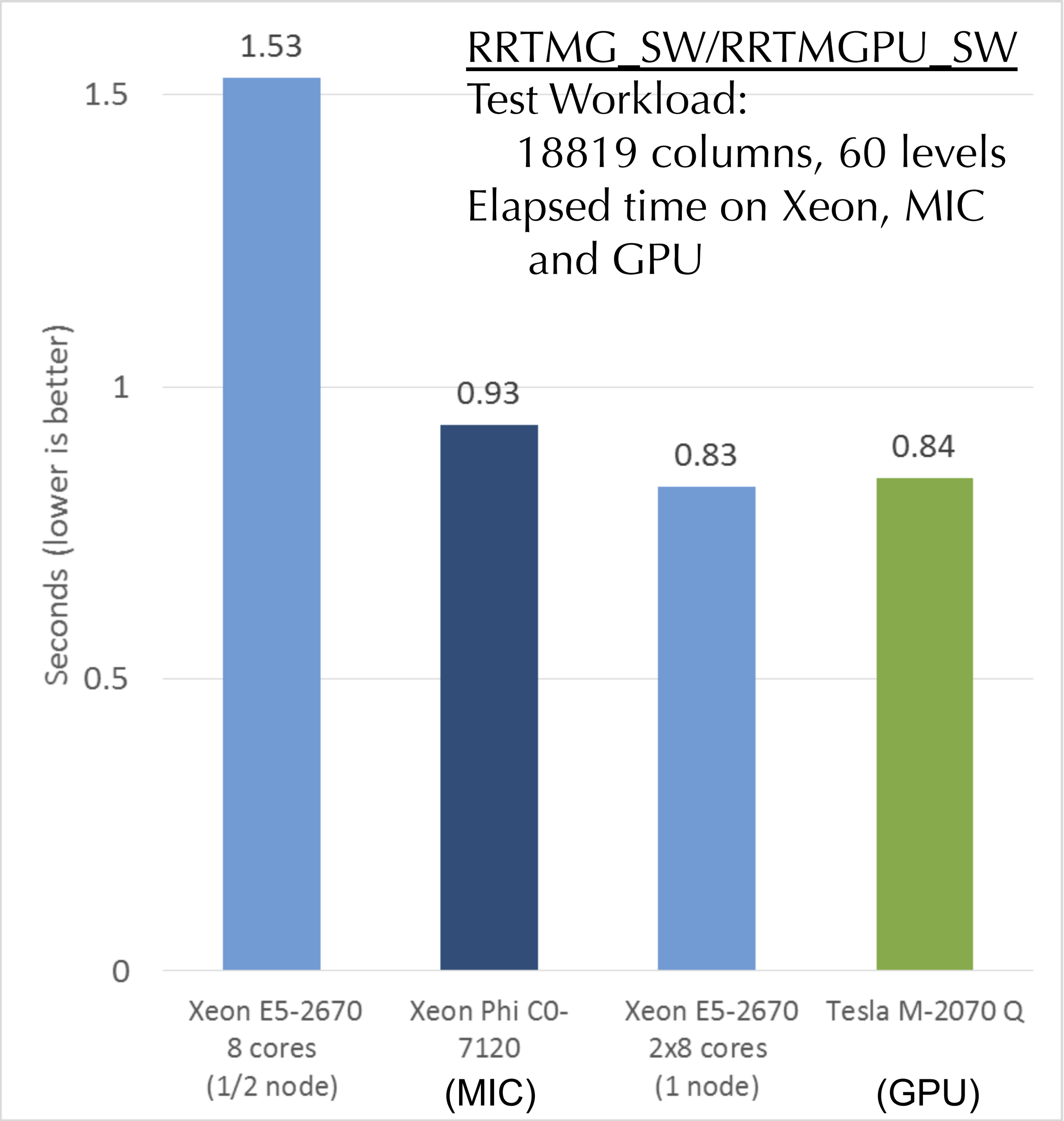


From Oreopoulos et al., JGR, 2012a:
The total percent error of each participating radiation code with respect to reference calculations for all LW (top) and SW (bottom) cases in the Continual Intercomparison of Radiation Codes (CIRC) project.

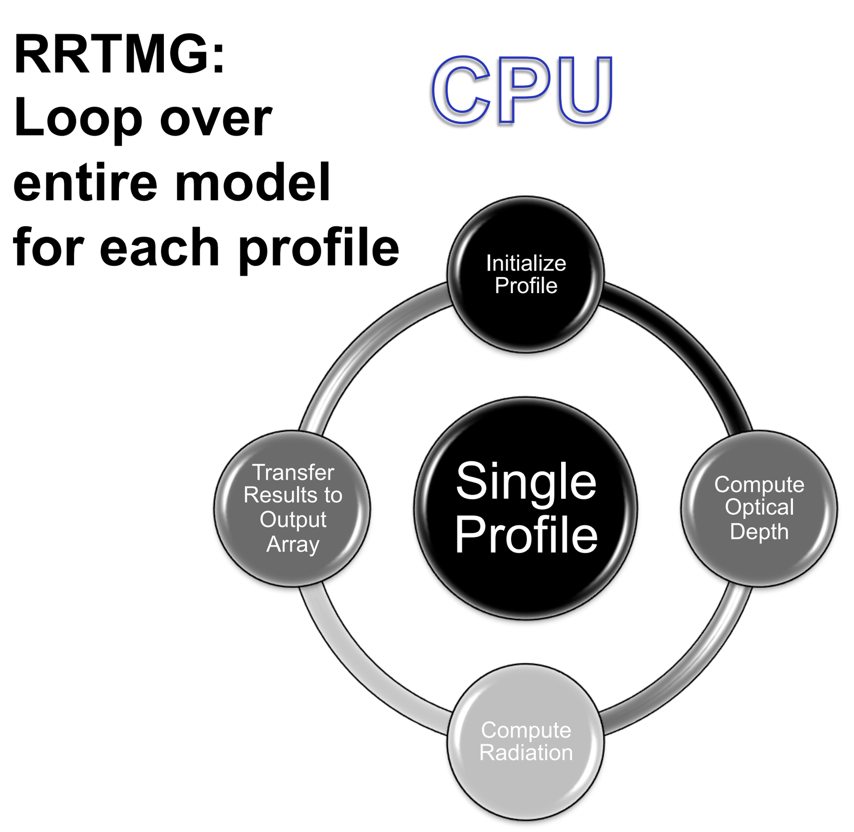
Computational savings will allow introduction of more sophisticated physics packages elsewhere in WRF.

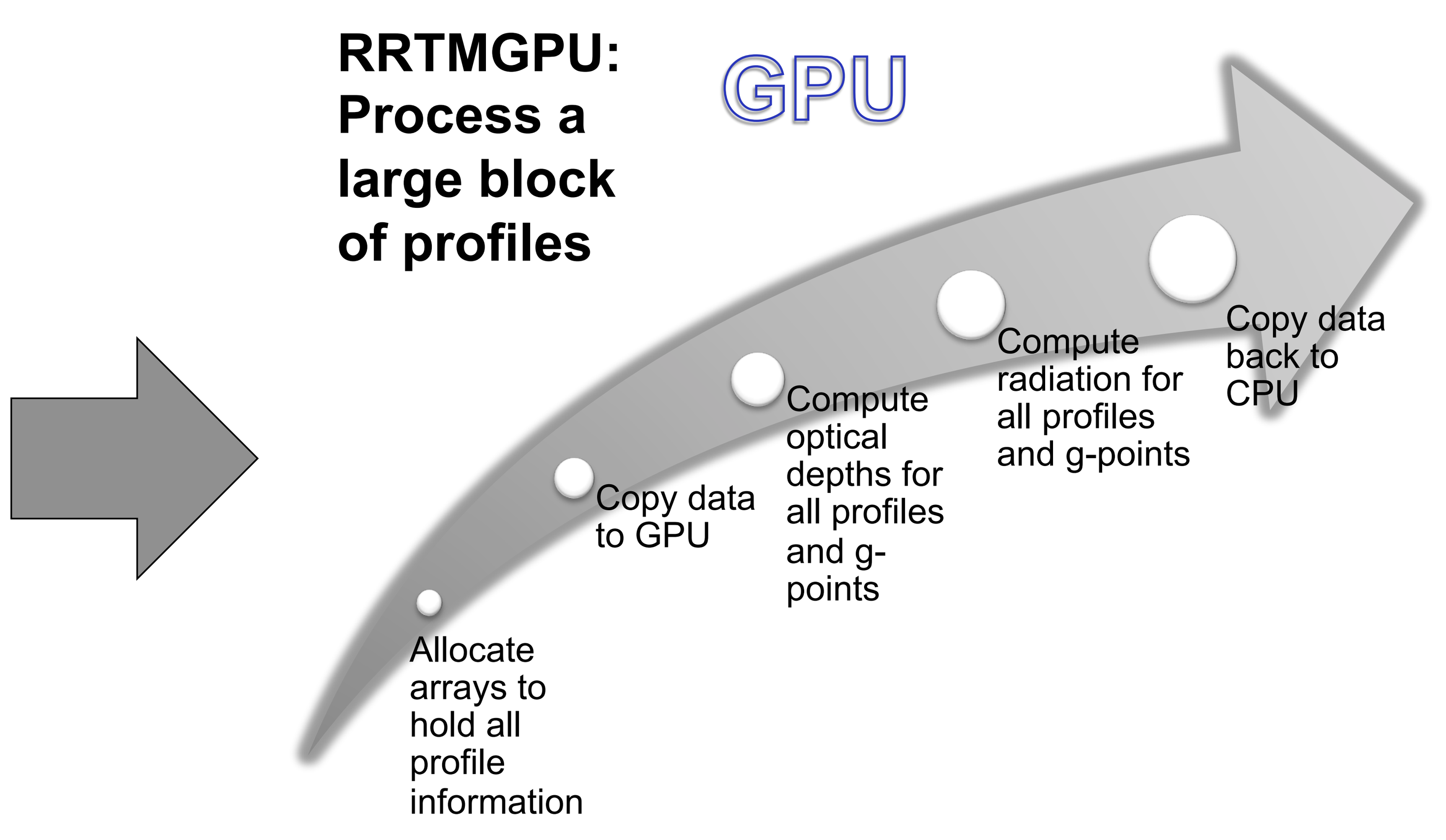
RRTMG Performance on MIC

- MIC: Intel Many Integrated Core architecture (Xeon Phi processors)
- MIC code modifications in progress at NOAA for NCEP NMM-B and GFS forecast models
- Column, layer and g-point loops reordered for best efficiency on MIC
- Performance comparison of total **elapsed time** (right) for RRTMG_SW on Xeon Sandybridge (8 and 16 cores, light blue), and Xeon Phi (MIC, dark blue), and for RRTMGPU_SW on a GPU (green); all single precision
- Xeon and GPU tested are not latest versions of vendor hardware



Restructuring RRTMG to Run Efficiently on Graphics Processing Units (GPUs)



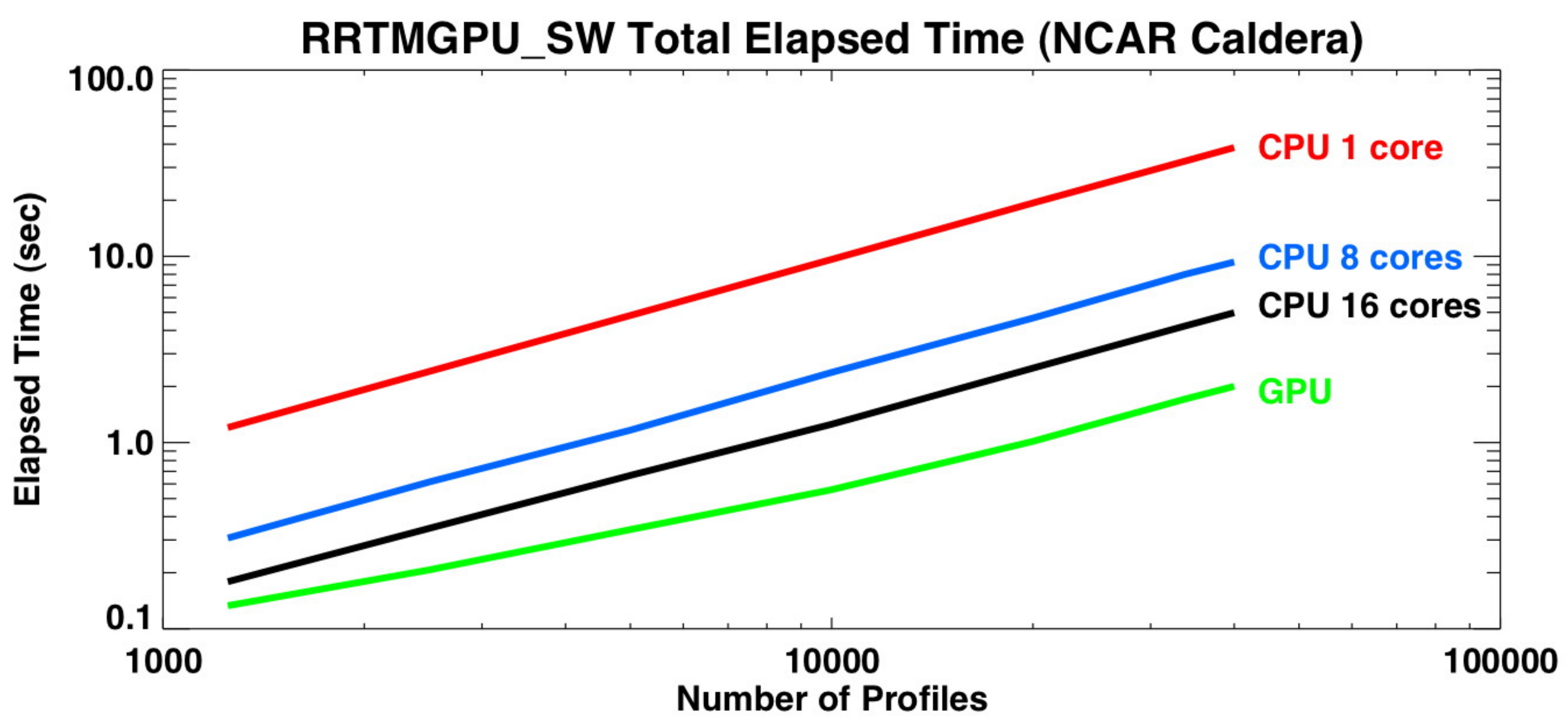


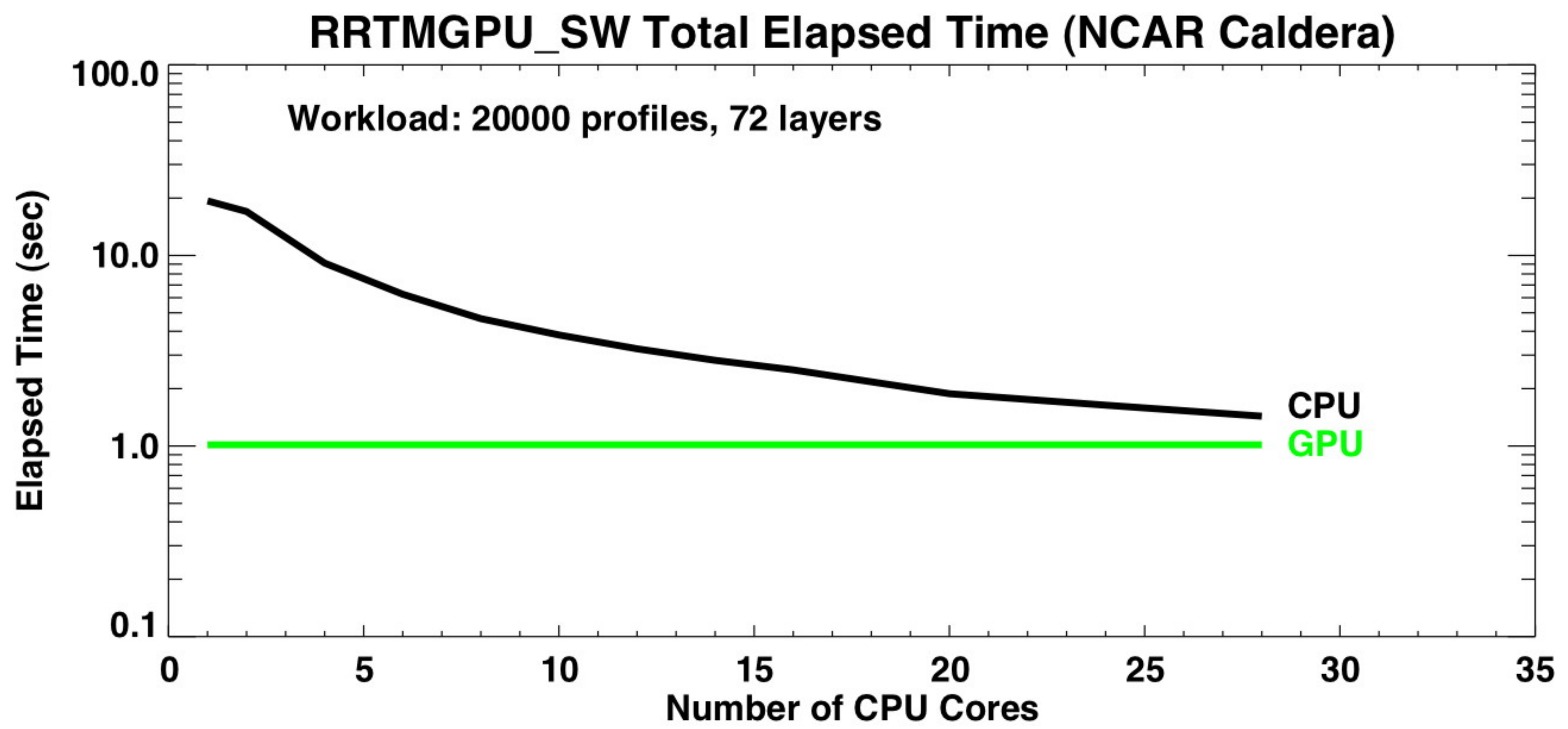
- In order for every profile to be run in parallel, arrays were padded to be multiples of 32, the size of a warp on a GPU, and reordered so that the fastest changing dimension would coincide with the thread layout to enable efficient memory coalescing.
- Algorithms were restructured so that g-points can be run in parallel, ensuring that even with a relatively low number of profiles, the GPU is always busy and therefore running efficiently.
- Look-up tables were removed and calculations were implemented within the main loop to avoid scattered memory access and enable more efficient execution on the GPU.
- Profile partitioning was implemented using the MPI API and multiple streams for running RRTMG on multiple GPUs in parallel.
- The main loop was restructured so that, instead of running a single profile at a time, the various subroutines for all of the profiles were run in parallel.

RRTMGPU Performance (Off-line)

Test Environment: **NCAR Caldera**

- **System** Configuration:
  - ➢ Compiler: PGI_v13.9 with CUDA Fortran (v5.0) and openACC; single precision used for all tests
  - ➢ Caldera CPU: 2.6 GHz Intel Xeon E5-2670 (SandyBridge)
  - ➢ Caldera GPU: NVIDIA Tesla M2070-Q, Compute Capability 2.0
- **Radiation** Configuration:
  - ➢ RRTMGPU_LW/SW running off-line on CPU and GPU
  - ➢ Input data generated for 1250 to 40000 clear and cloudy profiles
- **Radiation** Timing Performance:





RRTMGPU Performance (WRF)

Test Environment: **NCAR Caldera**

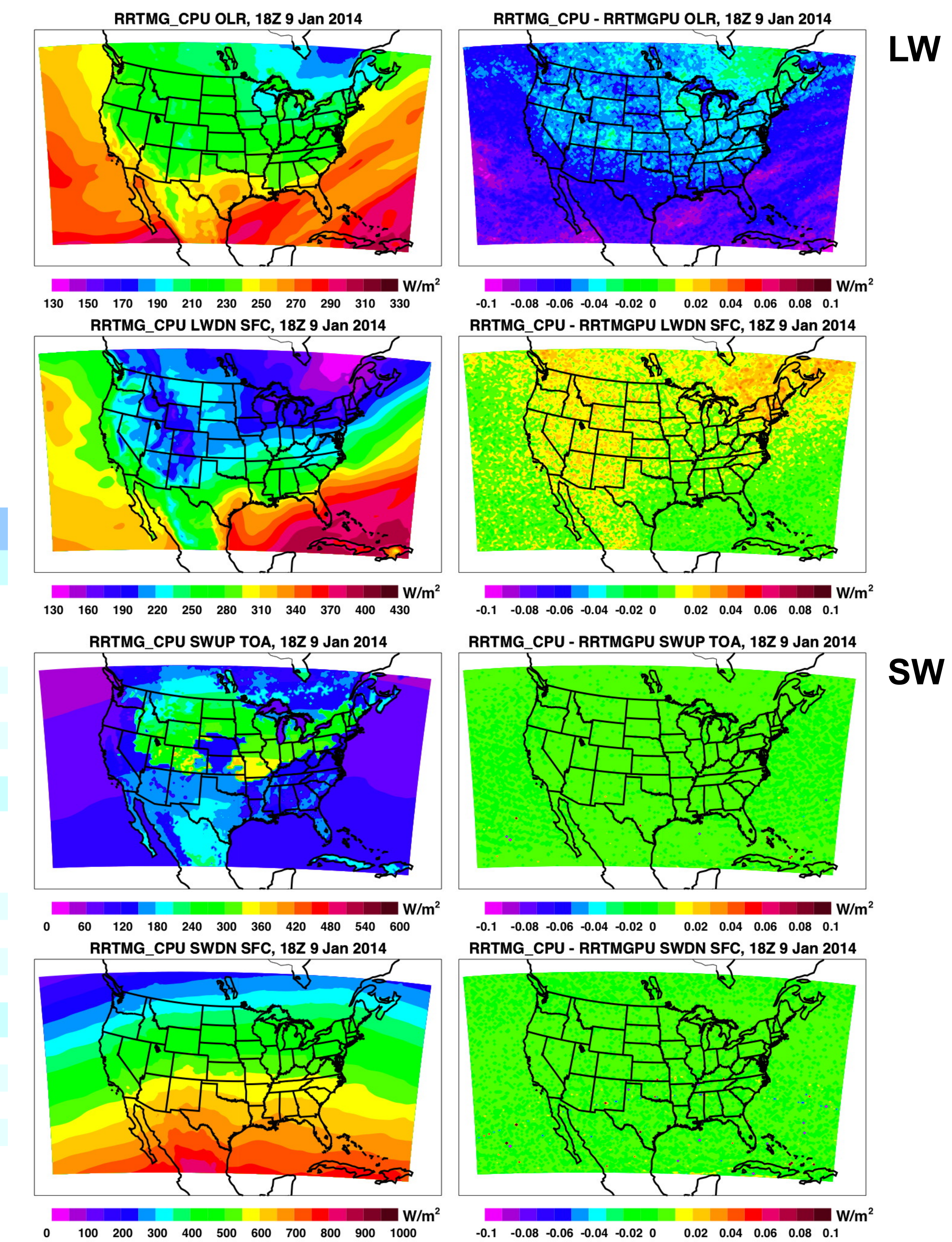
- **System** Configuration:
  - ➢ Same as off-line tests (see left)
- **WRF** Configuration: (Two runs: CPU/GPU)
  - ➢ WRF_v3.51 (configured for 1 and 8 CPU processors)
  - ➢ Old Rad: RRTMG_LW_v4.71, RRTMG_SW_v3.7
  - ➢ New Rad: RRTMGPU_LW (**includes physics changes** equal to RRTMG_LW_v4.85), RRTMGPU_SW (same physics as SW in WRF)
  - ➢ Single CONUS grid, **33750 grid points**, 29 layers, time step: 3 min., radiation time step: 30 min., 1-day forecast: 18Z, 9-10 Jan 2014
- **WRF** Radiation Timing Performance:

WRF RRTMG_CPU/RRTMGPU Performance Examples on NCAR/Caldera

WRF/CPU (1 Core; "serial")			**WRF/CPU (8 Cores; "dmpar")**		
Model	Elapsed Time (sec)	Time Fraction vs. WRF	Model	Elapsed Time (sec)	Time Fraction vs. WRF
LW	904.3	0.28	LW	116.6	0.23
SW	643.6	0.20	SW	90.2	0.18
LW+SW	1547.8	0.48	LW+SW	206.8	0.40
WRF	3210.7	1.00	WRF	512.6	1.00
WRF/CPU + RRTMGPU			**WRF/CPU + RRTMGPU**		
Model	Elapsed Time (sec)	Time Fraction vs. WRF	Model	Elapsed Time (sec)	Time Fraction vs. WRF
LW	70.2	0.04	LW	70.2	0.16
SW	55.4	0.03	SW	55.4	0.13
LW+SW	125.6	0.07	LW+SW	125.6	0.29
WRF	1944.2	1.00	WRF	431.4 (estim.)	1.00
Model	**CPU/GPU** Time Ratio		Model	**CPU/GPU** Time Ratio	
LW	12.9		LW	1.7	
SW	11.6		SW	1.6	
LW+SW	12.3		LW+SW	1.7	
WRF	1.7		WRF	1.2	

WRF/RRTMGPU Output Verification

- ➢ Small differences in LW fluxes are expected and due to physics changes
- ➢ No impact on SW fluxes from running on GPU (except through LW)



Summary

- RRTMGPU_LW/SW are working both offline and within WRF_v3.51 at NCAR,
- Running the radiation codes on the GPU presently requires the PGI compiler (v13.9), a recent version of CUDA Fortran (e.g. v5.0), openACC, and NVIDIA GPU hardware,
- Results show improved performance of RRTMGPU within WRF and a significant reduction in the fraction of total model time spent on radiative transfer,
- Additional speed-up is possible with further configuration refinement; specific performance improvement is also dependent on the GPU hardware; faster GPUs are available than the NVIDIA Tesla M-2070Q in use in Caldera,
- Separate tests on Xeon Phi (MIC) architecture also show improved performance relative to tests on the GPU and Xeon Sandybridge CPU

Future Work

- Timing improvement reported here is a preliminary result; it is essential to perform consistent comparisons between optimal CPU and GPU environments,
- Dependence of timing improvement on WRF grid size will be quantified,
- Further refinement of GPU application will be completed to determine optimal configuration,
- Version of RRTMGPU in use here is a transitional model; under separate funding (ONR) the radiation codes will be completely redesigned to further enhance their parallel processing capability and generalized application,
- Current RRTMGPU or a later version will be made available to NCAR for application to a future WRF release

Acknowledgment: This work is supported by the DOE Office of Science Earth System Modeling (ESM) SciDAC Program under grant DE-SC0007038.