Enhancing Efficiency of the RRTMG Radiation Code with Graphics Processing Units in the Weather Research and Forecasting Model

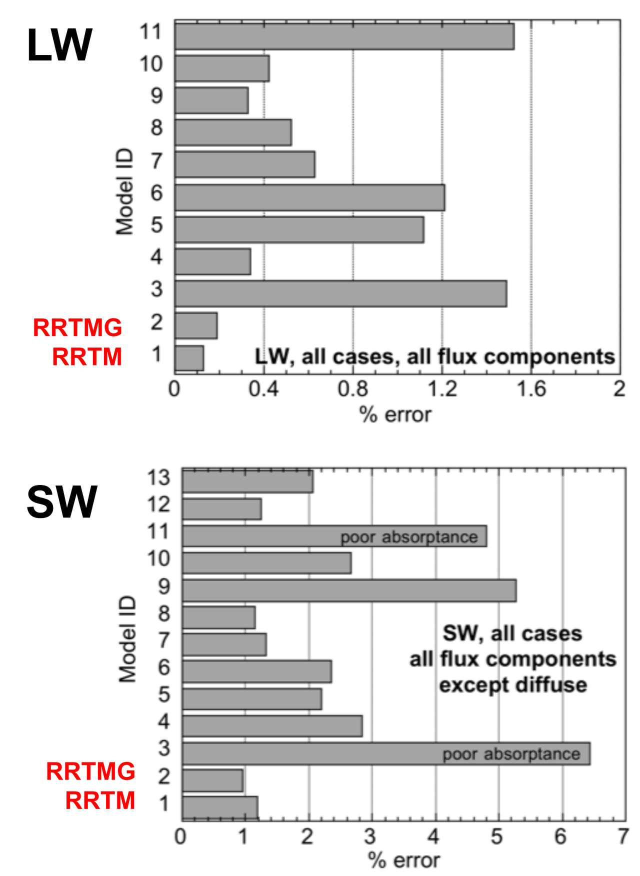
Michael J. Iacono[1], David Berthiaume[1], Eli Mlawer[1], John Michalakes[2]

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

RRTMG, Radiation Calculations in 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)



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.

- 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

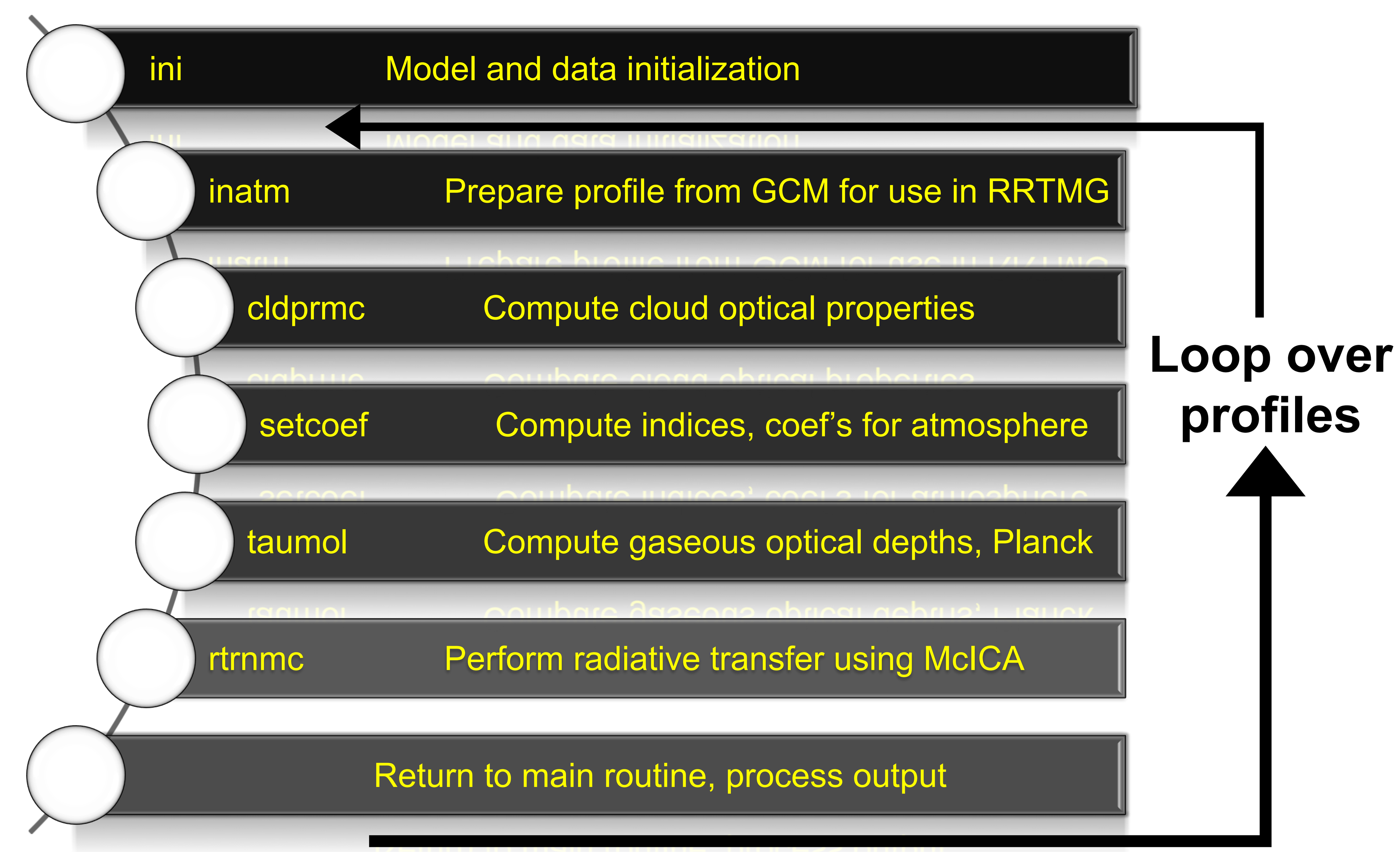
➢ NCEP GFS (2003,2010), CFS (2004,2010), RUC (2008)

➢ ECMWF IFA (2000,2007) and ERA40

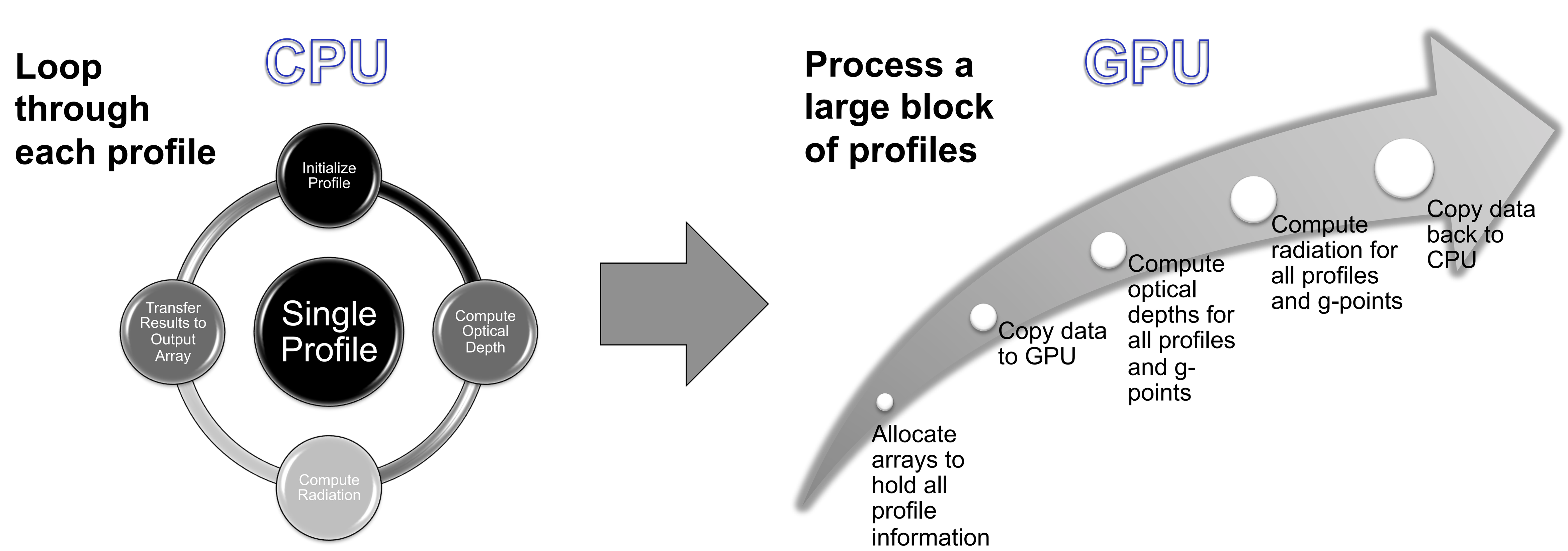
➢ ECHAM5 (2002)

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

Structure of RRTMG_LW/SW



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

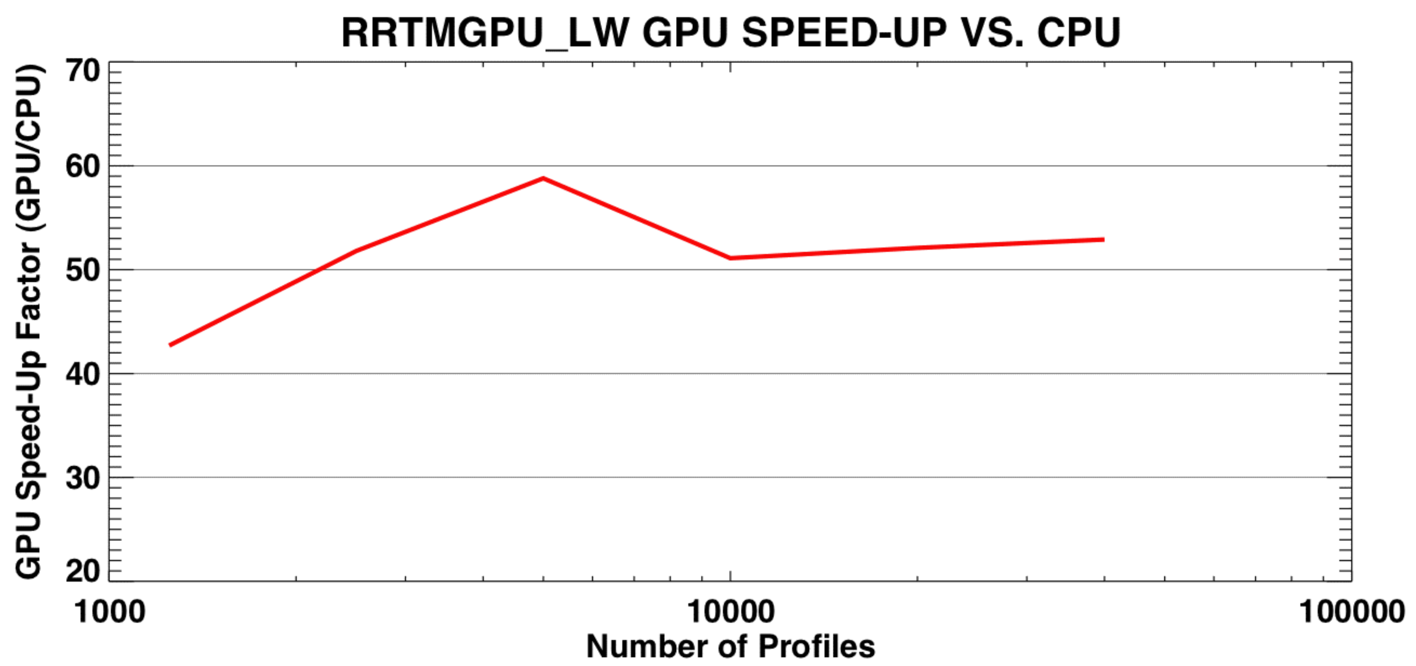


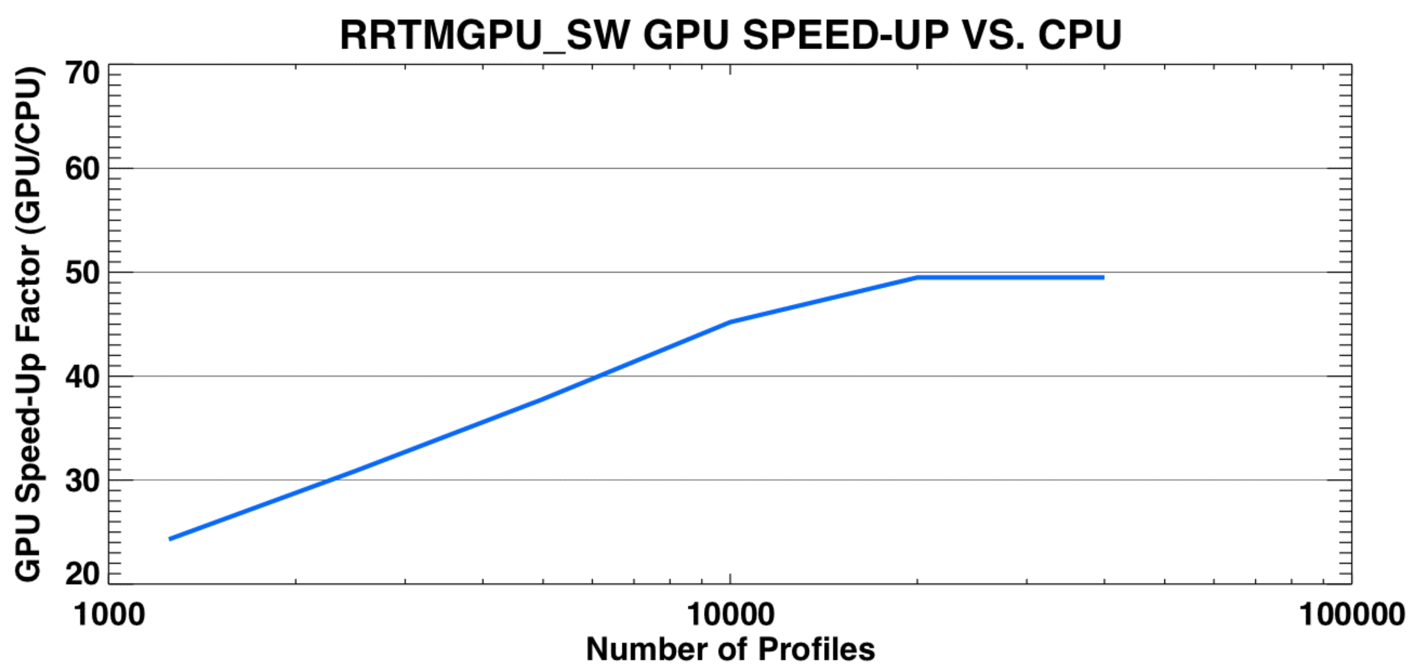
- In order for every profile to be run in parallel, arrays were padded to be multiples of 32, the size of a warp on a GPU, and reordered so that the fastest changing dimension would coincide with the thread layout to enable efficient memory coalescing.
- Algorithms were restructured so that g-points 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 (Offline)

Test Environment: **NCAR Caldera**

- **System** Configuration:
  - Compiler: PGI_v13.3 (LW), PGI_v13.9 (SW) with CUDA Fortran and openACC
  - Caldera CPU: 2.6 GHz Intel Xeon E5-2670 (SandyBridge)
  - Caldera GPU: NVIDIA Tesla M2070-Q,
  - Compiler Flags (CPU): -O3 –r4 –i4
  - Compiler Flags (GPU) –O3 –r4 –i4 –Kieee –acc –Mcuda -ta=nvidia,fastmath,cuda5.0
- **Radiation** Configuration:
  - RRTMGPU_LW/SW running offline on CPU and GPU
  - CMAKE compile build system used with PGI_v13.3 and v13.9
  - Input data generated for 1250 to 40000 clear and cloudy profiles
- **Radiation** Timing Performance:





RRTMGPU Performance (WRF)

Test Environment: **NCAR Caldera**

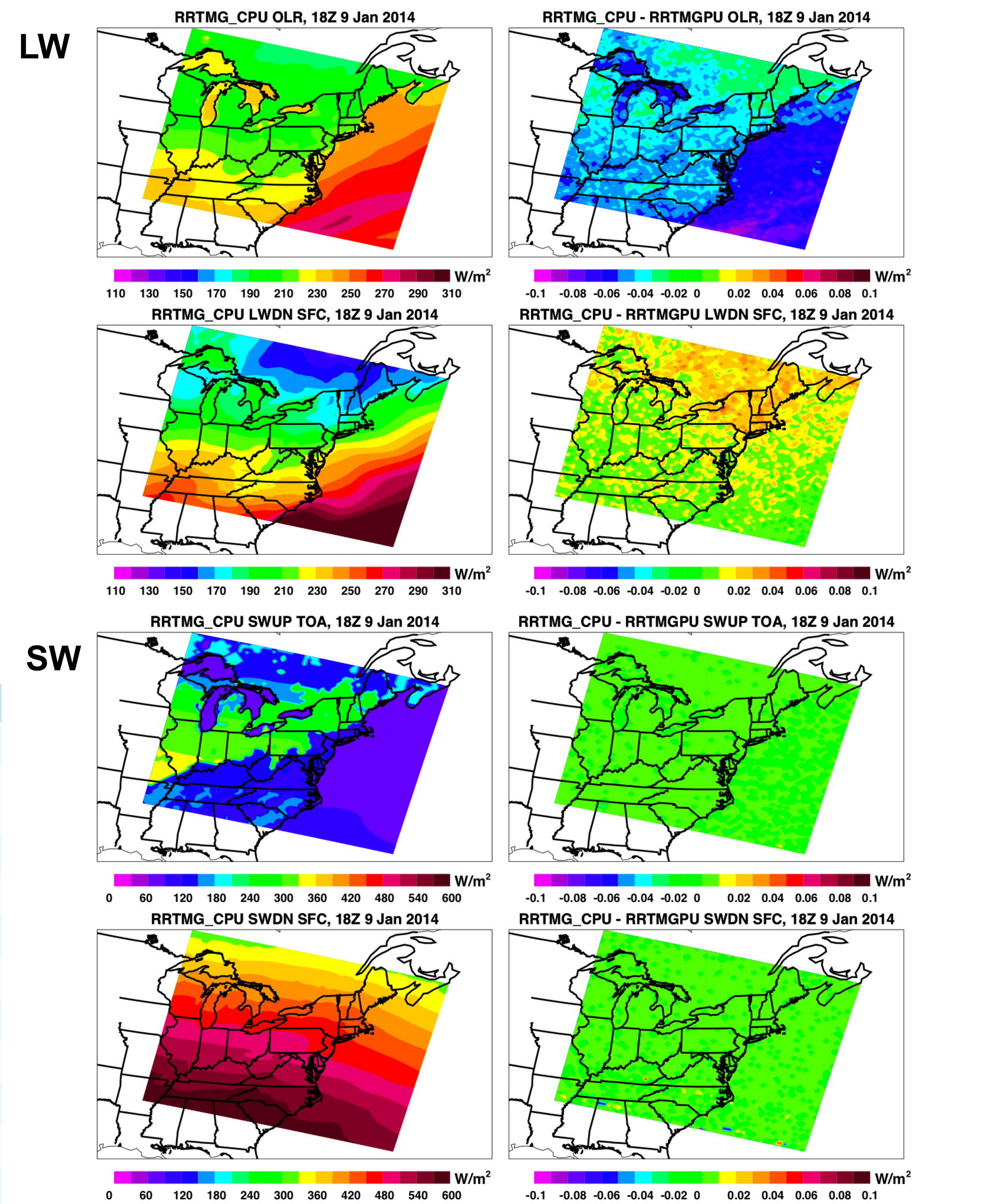
- **System** Configuration:
  - Compiler: PGI_v13.9 with CUDA Fortran (v5.0) and openACC
  - Caldera CPU: Two 2.6 GHz Intel Xeon E5-2670 (SandyBridge) per node
  - Caldera GPU: NVIDIA Tesla M2070-Q, with Compute Capability 2.0
  - Compiler Flags (CPU): -O3 –r4 –i4
  - Compiler Flags (GPU) –O3 –r4 –i4 –Kieee –acc –Mcuda -ta=nvidia,fastmath,cuda5.0,cc20
- **WRF** Configuration: (Two runs: CPU/GPU)
  - WRF_v3.51 (configured for 1 CPU processor)
  - 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 grid, 4380 grid points, 29 layers, time step: 3 min., radiation time step: 30 min., one-day forecast 9 Jan2014 18Z to 10 Jan 2014 18Z
- **WRF** Timing Performance (1 model day)

WRF RRTMG CPU/GPU Performance Example on NCAR Caldera

CPU			**GPU**		
Model	Time(sec)	Time vs. WRF	Model	Time(sec)	Time vs. WRF
LW	119.6	0.29	LW	10.3	0.04
SW	85.2	0.21	SW	6.5	0.03
LW+SW	204.8	**0.50**	LW+SW	16.8	**0.07**
WRF	411.4	1.00	WRF	241.4	1.00
CPU/GPU	Time Ratio		**GPU/CPU**	Time Ratio	
LW	11.56		LW	0.09	
SW	13.17		SW	0.08	
LW+SW	**12.18**		LW+SW	0.08	
WRF	1.70		WRF	0.59	

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), and NVIDIA GPU hardware,
- An initial speed-up of a factor of 12 from RRTMG to RRTMGPU has been achieved within a one-day WRF forecast on a single regional grid relative to a single processor,
- Additional speed-up is expected with further configuration refinement,
- WRF grid size is a significant factor in the potential speed-up; faster results are expected on larger forecast grids than used here,
- Specific performance improvement is also dependent on the GPU hardware available; faster GPUs are available than the NVIDIA Tesla M-2070Q in use in Caldera

Future Work

- Timing improvement reported here is a preliminary result; it will be essential to perform a fair comparison between the 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.