Empowering Cloud-Resolving Models Through GPU and Asynchronous I/O

IS&T
11 AM, February 18
Building 3

Wei-Kuo TAO (PI)
Thomas L. Clune (Co-PI)
Shujia Zhou (Co-PI)
Toshihisa Matsui (Co-I)
Xiaowen Li (Co-I)
Xiping Zeng (Co-I)

Earth Science Technology Office

Project Objectives

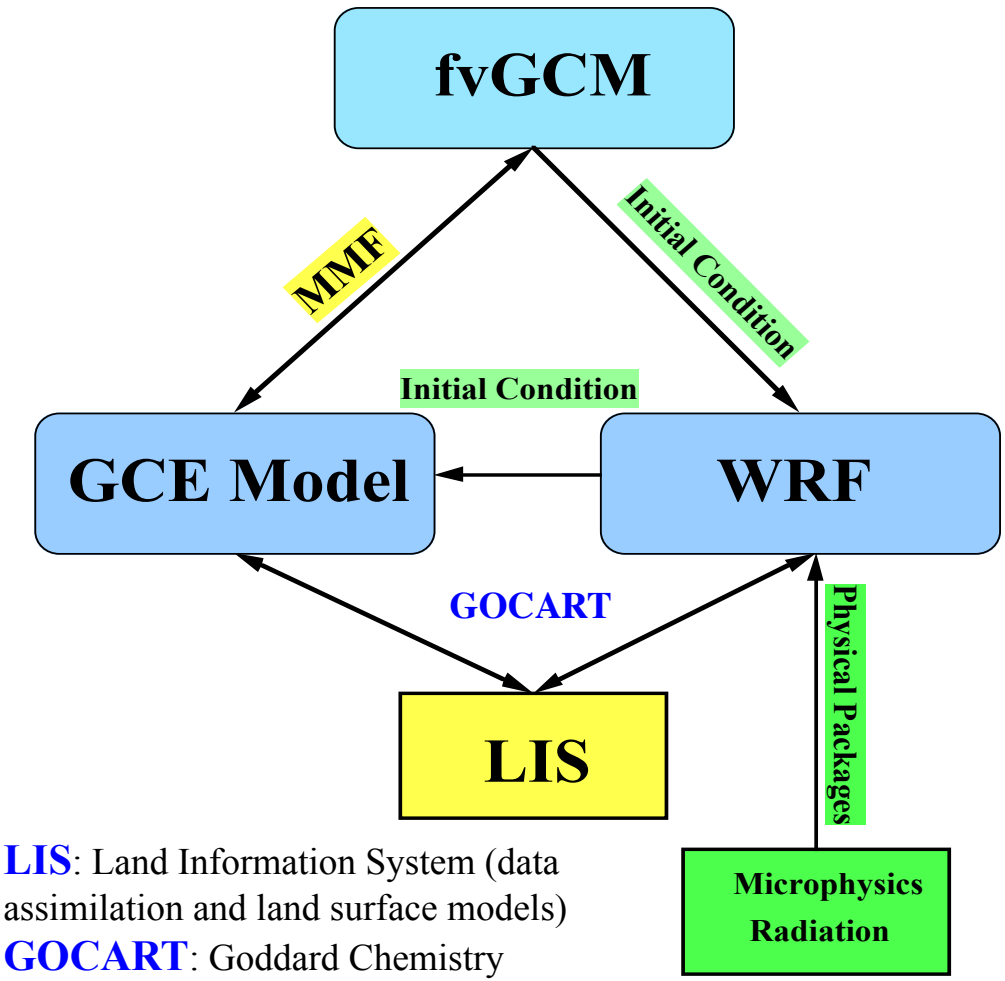
- Improve NASA cloud resolving models' computational performance by porting computationally-intensive components (Radiation and Microphysics) to Graphics Processing Units (GPUs)

- Develop an Asynchronous I/O tool to offload output data from compute node to reduce the idle time of computing processors

- Develop a data compression mechanism to further enhance the Asynchronous I/O tool

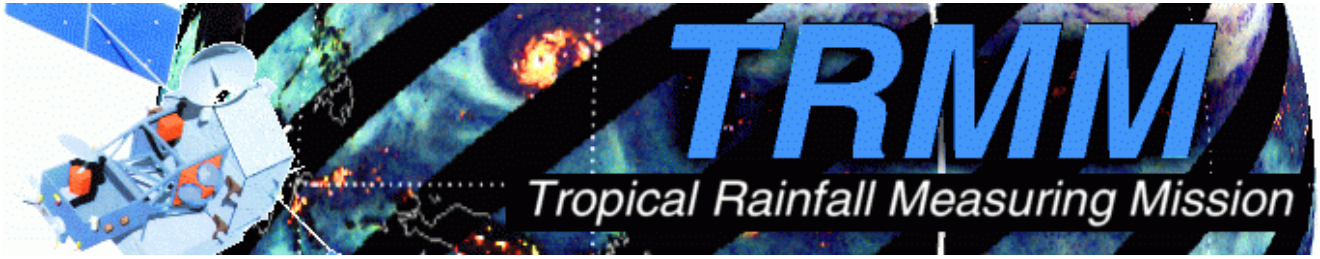
NASA Cloud Resolving Models

- **Multi-scale modeling system developed at Goddard with unified physics** from:
  1. **Goddard Cumulus Ensemble model** (GCE), a cloud-resolving model (CRM)
  2. **NASA Unified Weather Research and Forecasting Model** (NU-WRF), a region-scale model, and
  3. Coupled GEOS4/5-GCE, the GCE coupled to a general circulation model (or GCM known as **Goddard Multi-scale Modeling Framework or MMF**).
- Same parameterization schemes all of the models for microphysical processes, long- and short-wave radiative transfer, and land-surface processes, to study explicit cloud-radiation, cloud-aerosol and cloud-surface interactive processes.
- ***Coupled with multi-sensor simulators for comparison and validation of NASA high-resolution satellite data.***



LIS: Land Information System (data assimilation and land surface models)
GOCART: Goddard Chemistry Aerosol Radiation and Transport Model

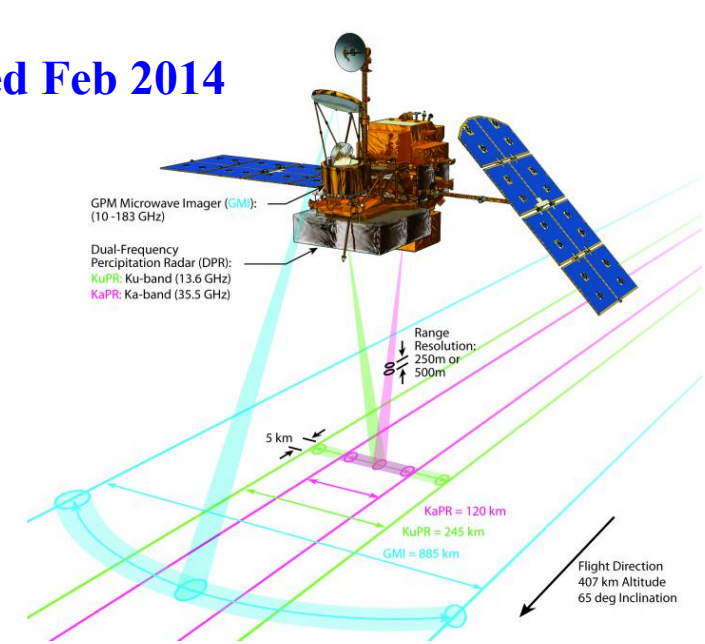
Tao, W.-K., S. Lang, X. Zeng, X. Li, T. Matsui, K. Mohr, D. Posselt, J. Chern, C. Peters-Lidard, P. Norris, I.-S. Kang, A. Hou, K.-M. Lau, I. Choi, M. Yang, 2014: The Goddard Cumulus Ensemble (GCE) Model: Improvements and Applications for Studying Precipitation Processes. An invited paper - *Atmos. Res.,* **143**, 392-424.



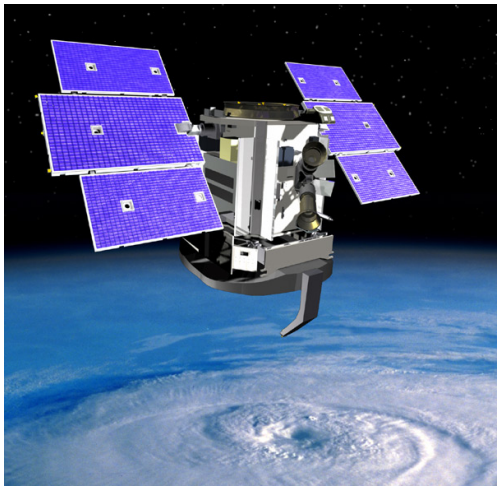
Launched in 1997

GPM

Launched Feb 2014



CloudSat



Launched 28 April 2006

CaPPM

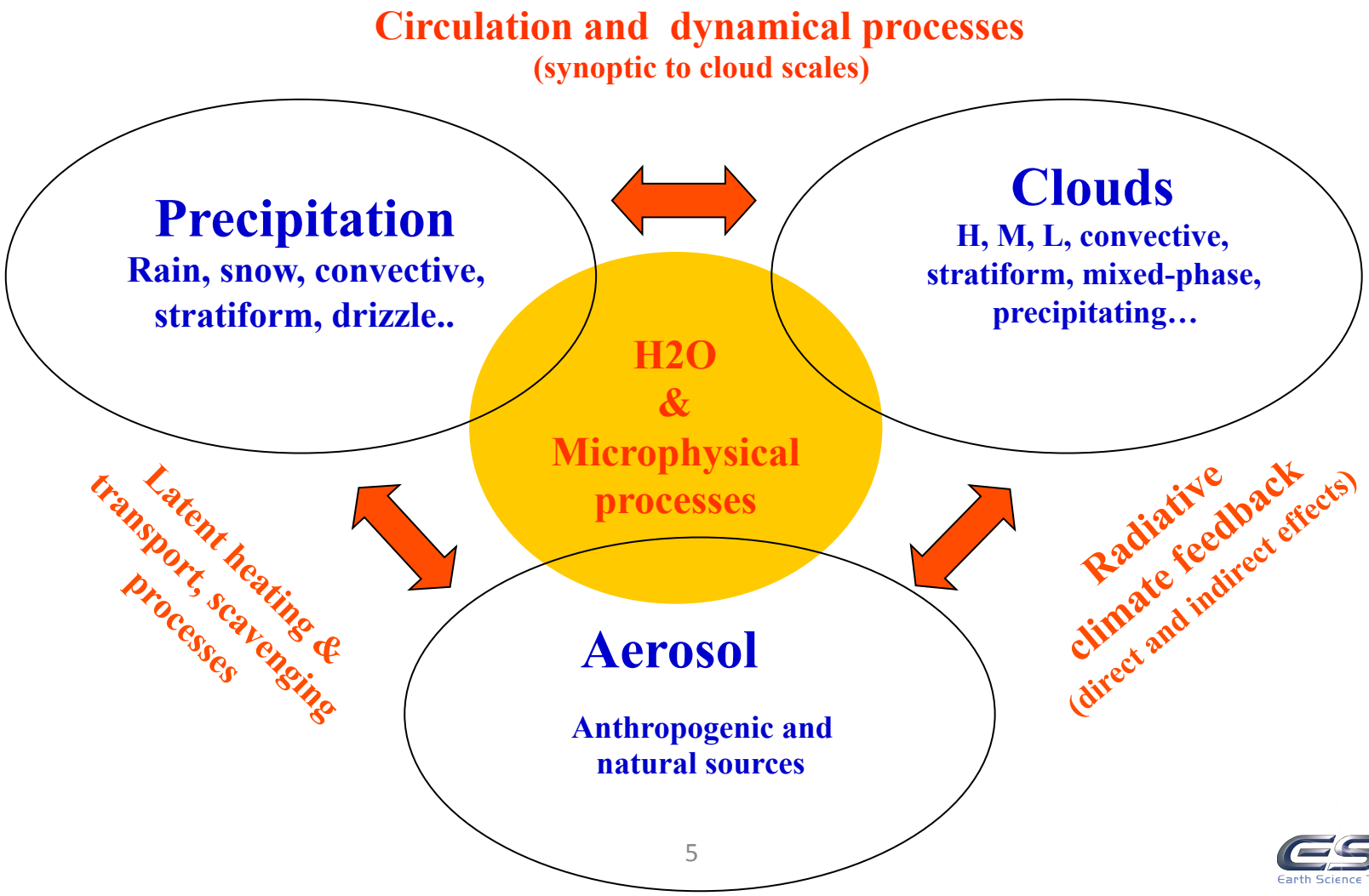
Cloud and Precipitation Processes Mission

Current/future global cloud resolving models (km resolution) need to use cloud / precipitation processes developed in cloud resolving models.

ESTO
Earth Science Technology Office

An Integrated Approach to Atmospheric Water Cycle and Climate Change Research

(satellite observations, field campaigns, modeling, data processing and applications)



5

CPU for radiation and microphysics

	Total CPU hours	Total CPU number	Radiation	Microphysics	Dynamics
Bulk (radiation every 10 steps)	198	64	25.1%	10.0%	63.9%
Bulk (radiation every step)	658	64	77.2%	3.0%	19.5%
Bin (radiation every 10 step)	44,697	1024	0.12%	45.8%	54.1%
Bin (radiation every step)	47,138	1024	1.16%	45.8%	53.0%

CPU times for 3D GCE simulations for a convective case on the NASA Pleiades computer. The domain size is 256x256x41, total integration time is 24 hours with 3 seconds time step. Dynamics includes the advection of all variables as well as the pressure solver.

Spectral bin microphysics scheme cost is about 326 CPU time compared with 1-Moment bulk run

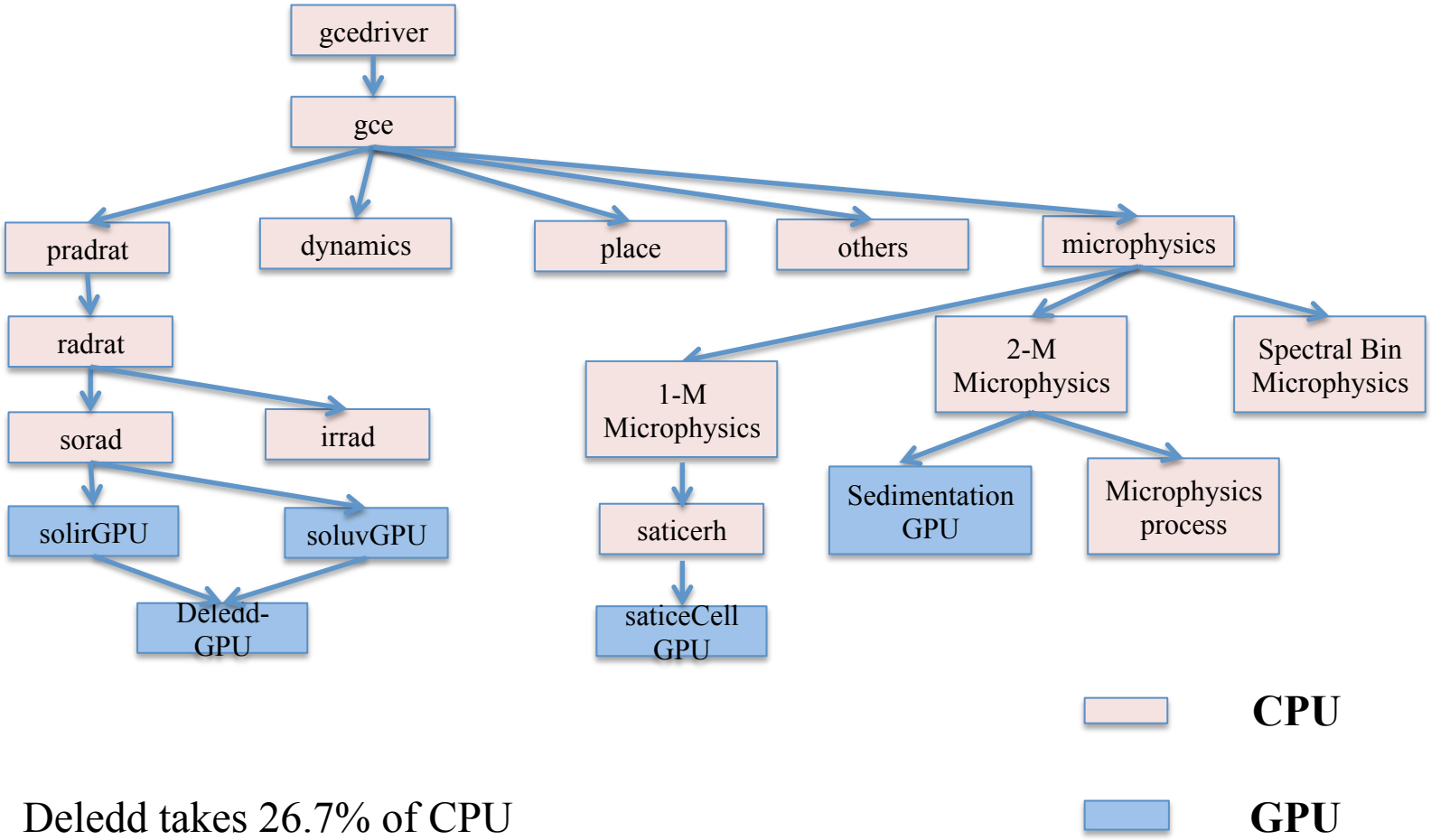
I/O data requirements Microphysical Scheme

Estimations based on the domain size of 256 x 256 x 41 grid points, for a total 5-days integration time, using FORTRAN binary format

	single output	current output frequency	current data amount	**desired** output frequency	desired data amount
Dynamics	0.12 G	1 hr	14.4 G	5 min	0.17 T
Bulk microphysics	0.15 G	1 hr	18.0 G	5 min	0.22 T
Bin microphysics	5.4 G	1 hr	648 G	5 min	**7.78 T**
Statistics	0.4 G	simulation period	0.4 G	simulation period	0.4 G
Total	18 G	------	680.8 G	------	8.17 T

Goddard MMF: 5 TBs for 1 year run with hourly CRM output and 45% wall time for output

GCE Code Structure



Deledd takes 26.7% of CPU

Radiation Integration

- Since the performance of GPU's version of infra radiation is not satisfactory, we integrate the GPU version of solar radiation into the updated GCE
- Experiment configuration
  - Simulation Case: TRMM LBA Feb 23
  - 128x128x41grid points, 3 simulation hours
  - CUDA Fortran
  - CUDA 4.0 driver
  - OpenACC GPU

Deledd() Performance and Precision Comparison

- Deledd() takes ~26.9% computation time of radiation
- There is a precision difference
  - E.g., in solir() routine, all-sky flux (downward minus upward), flx
    - 0.7947214478638445 without GPU
    - 0.7949332959524538 with GPU
- For the configuration of 128x128 columns, performance comparison against one CPU core

Processor	Time (micro second)	Speedup
CPU	68850	
OpenACC with IO	14440	4.77X
CUDA Fortran with IO	11898	5.78X
CUDA Fortran without IO	220	321X

142X (U. Wisconsin)

Performance

	Intel Fortran CPU (Second)	PGI CUDA Fortran GPU 128x128 in GPU 64x4 Threads (Second)	Speedup
Main()	37339.9	24581.8	1.52
Radiation()	15303.6	6013.3	2.54

Note:

1. **Timing includes copy-in and copy-out operations**
2. Performance is not sensitive to computing resource configuration
3. The numerical results for precipitation and SW are almost identical
4. Timing in previous performance report is on solir() and soluv() rather than radiation() and main()

Previous Results

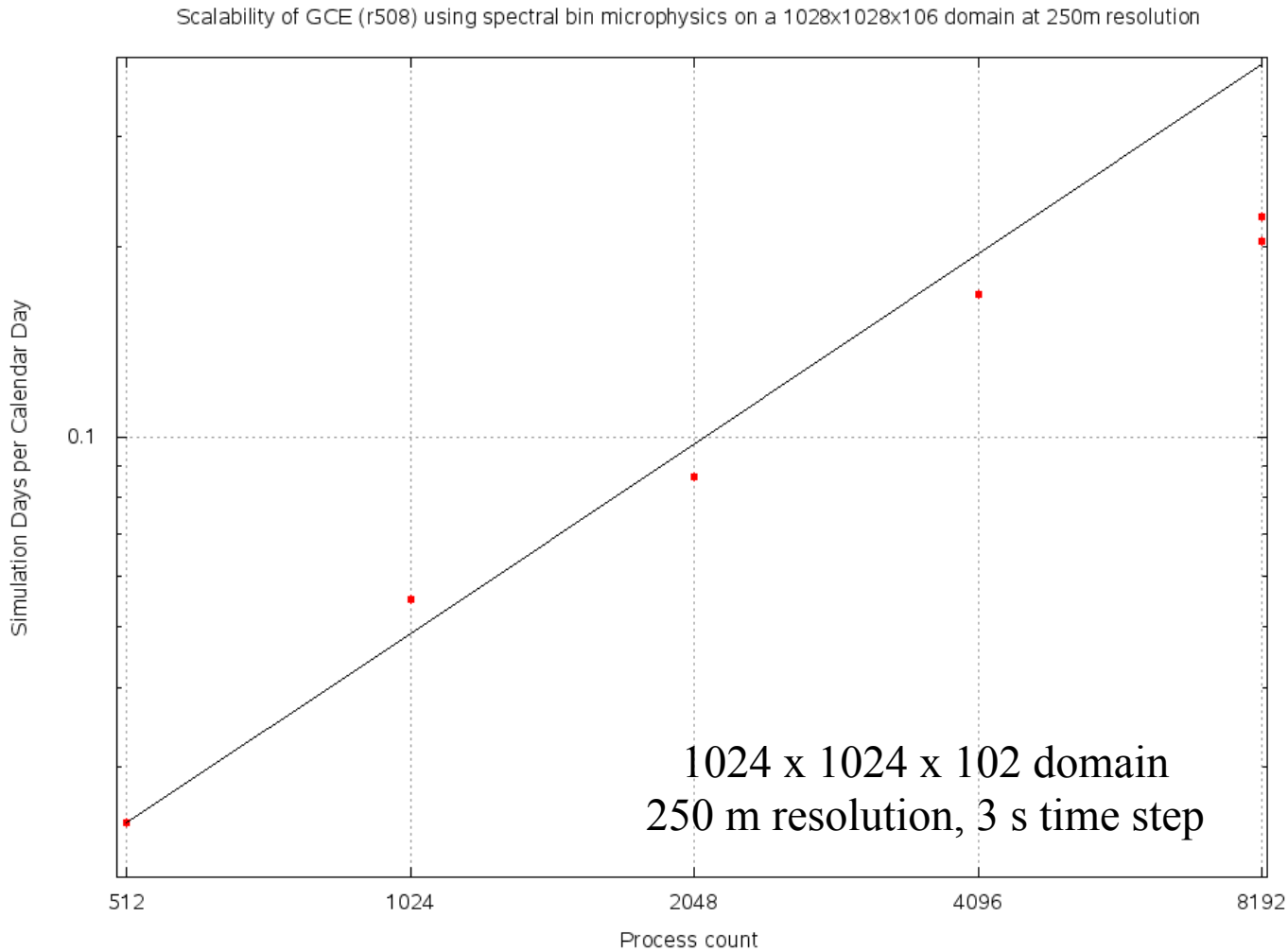
	CPU/GPU
Soluv() (Micro second)	19,406,336/ 2,450,730 = **7.92**
Solir() (Micro second)	71,337,870 / 7,859,607 = **9.08**

Port One- and Two-Moment Microphysics to GPU

- One-Momentum Scheme
  - Improved the original code for better parallel computation
  - Ported the improved code into GPU
- Two-Momentum Scheme
  - Developed several solutions to overcome the GPU limitations in compiler and hardware for **large numbers of input/output array variables (3D, ~20) as well as temporary array variables (1D, ~110)**
  - Obtained the consistent numerical results between CPU and GPU
    - Extraordinarily large numbers of input/output array variables as well as temporary array variables hinder performance gain
      - Copy-in and copy-out operations take considerable time
      - **Overly using GPU memory degrades performance**

Microphysics Scheme		Time (Seconds)	Ratio
One Moment Scheme	CPU (original code)	22.46	
	CPU (improved code)	8.25	
	GPU	3.06 (including I/O)	X 2.7 (to improved coc X 3.7 (to original code
Two Moment Scheme	CPU	0.03332	
	GPU (no copy-in/-out)	0.11206	X 0.297
	GPU (with copy-in/-out)	0.14776	X 0.225

GCE Scalability with Bin Physics and Parallel IO



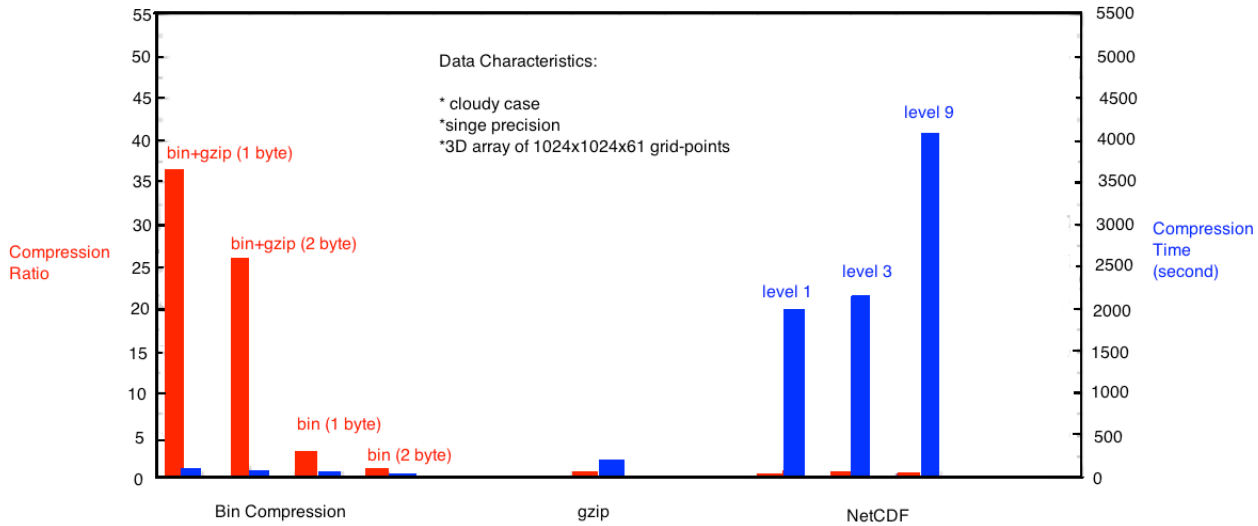
As the number of processes increases, the domain size decreases and the percentage of communication cost (halo update) increase. Consequently, scalability is not linear. **Without parallel IO, *1024x1024x102* run fails due to memory limitation of a single processor core.**

Simulations were carried out in NASA NAS Pleiades. Use ivybridge nodes with 16 ranks per node evenly distributed on the two sockets

Preliminary Results on Data Compression

Red Bars: Compression Ratio (ratio of original size to final size)

Blue Bars: Time used for data compression



Three Compression Methods Used

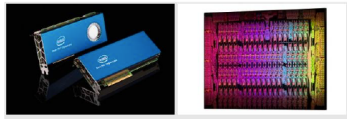
1. gzip (middle)
2. NetCDF with different compression levels (right)
3. Proposed two-stage data compression (i.e., bin method + gzip; left)

Although **netCDF4** provides a mechanism for lossless compression, much higher compression ratios can be achieved through lossy compression schemes.

We are pursuing a hybrid approach developed by the Global Modeling and Assimilation Office that compresses data prior to exporting through netCDF.

Hybrid MPI/OpenMP

for off-line radiation code

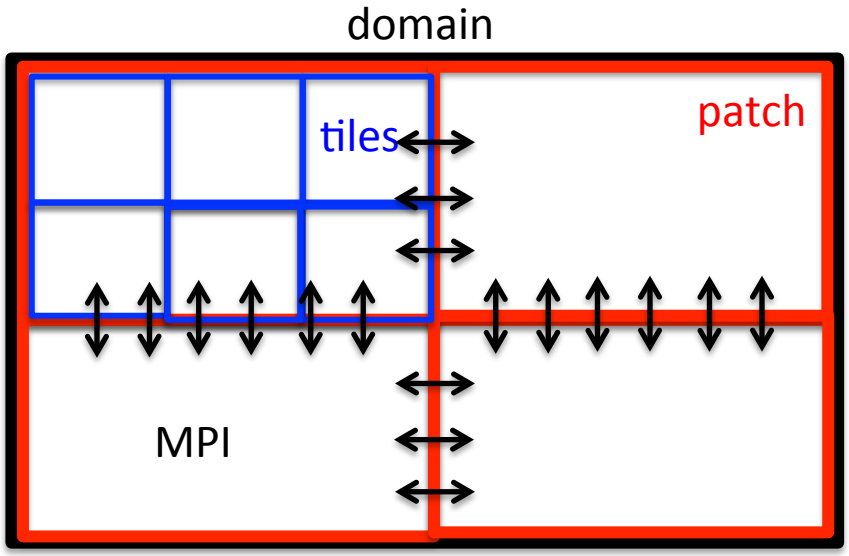


- Developed off-line framework of **NU-WRF Goddard Radiation scheme** under **Hybrid MPI/ OpenMP** structure.

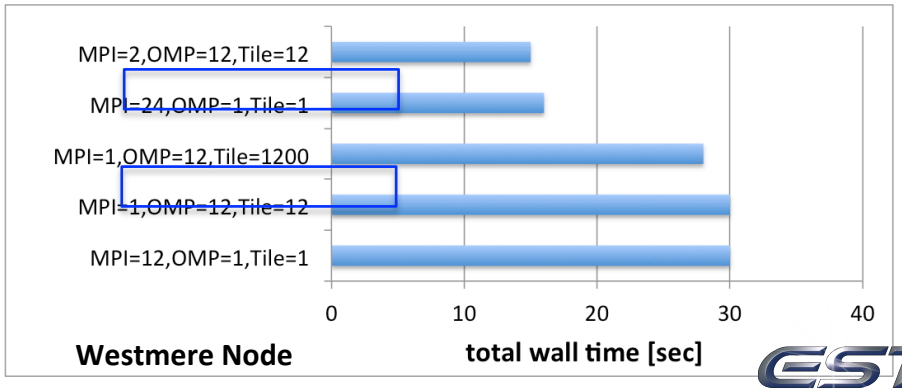
  This hybrid structure will work for both **MIC** node as well as convectional **CPU** node.

- Established a new benchmark of off-line **MPI/OpenMP** for scaling test, and result shows the equivalent performance (or slight gain) between MPI and Hybrid MPI/OpenMP with small number of node.

- Result ensures that hybrid structure will benefit ultra-large-scale simulation due to less MPI process and less memory use (~10%).



MPI process between patches.
OpenMP process between tiles.
Used for CPU node (tile # > 28) & MIC node (tile # >244)



Conclusions/Lessons Learned

- GPU acceleration
  - For compute-intensive components such as solar radiation, acceleration on component level is considerable. However, overall performance improvement is hindered by multiple-level (e.g., 4) drivers.
  - For complex components such as one-moment and two-moment scheme, performance gain requires significant code reengineering: (1) reduce number of variables transferring between CPU and GPU, (2) reduce number of temporary variables and make them scalar
- Parallel I/O
  - MPI IO can make GCE's IO in parallel and consequently enable large-domain calculations with bin physics scheme
  - AsyncIO can further improve parallel I/O
- Data compression
  - Lossy compression can significantly reduce data size

Happy Chinese New Year (February 18, 2015) Year of Sheep/Goat