High-Performance Weather Forecasting Model Advancement at SSEC using Accelerator Technology

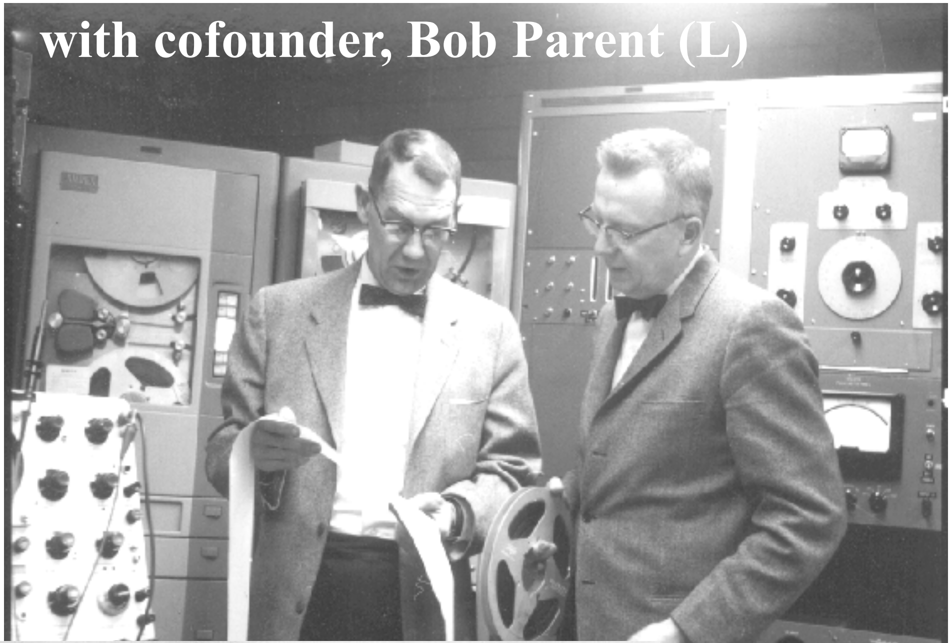
Current Status and Future Plan

Allen Huang, Bormin Huang, Jarno Mielikainen, Melin Huang
allenh@ssec.wisc.edu
Space Science and Engineering Center (SSEC)
University of Wisconsin-Madison

- ***About SSEC***
- ***Why HPC is relevant to SSEC – the Motivation***
- ***SSEC MIC & GPU accelerator R&D***
- ***Ways Forward***

16th ECMWF HPC Workshop
ECMWF, Reading, UK
30 October, 2014

SSEC works to maintain the spirit of exploration of its founder, Verner E. Suomi (1915-1995)



with cofounder, Bob Parent (L)

1959: 1st Meteorological Satellite Experiment

Earth Radiation Balance Observations on Explorer VII

1966: 1st Earth Imaging from GEO

Spin-scan Camera on 1st Advanced Technology Satellite

1980: 1st Infrared Sounder from GEO

VISSR Atmospheric Sounder on GOES-4

"Father of Satellite Meteorology"

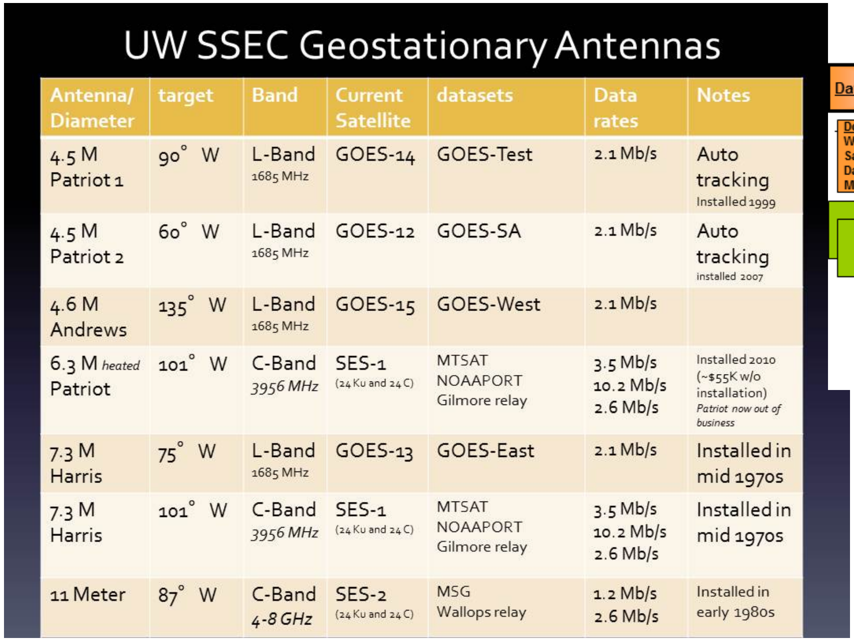
Weather Satellite renamed "Suomi NPP"

On 25 January 2012
NASA & NOAA renamed their newest Earth-observing satellite after UW-Madison space pioneer

Blue Marble
NPP VIIRS Image, GSFC

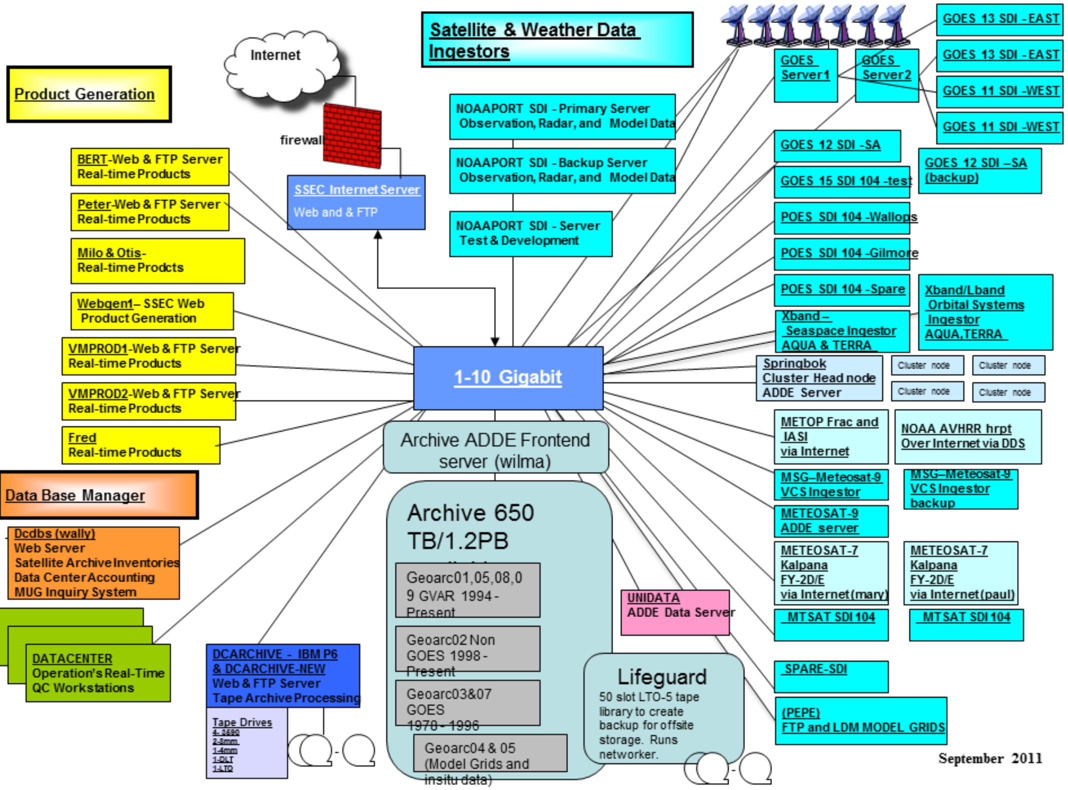
SSEC Data Center Infrastructure Serving Real-time Users around the clock

- Leveraging SSEC 30 years of meteorological satellite operation, currently real-time receive, process, and distributing 10 GEO and 12 LEO satellites;
- Served as NOAA GEO data archive center for more than 20 years;
- **World largest non-profit/non-governmental wx Satellite Center**



UW SSEC Geostationary Antennas

Antenna/ Diameter	target	Band	Current Satellite	datasets	Data rates	Notes
4.5 M Patriot 1	90° W	L-Band 1685 MHz	GOES-14	GOES-Test	2.1 Mb/s	Auto tracking Installed 1999
4.5 M Patriot 2	60° W	L-Band 1685 MHz	GOES-12	GOES-SA	2.1 Mb/s	Auto tracking installed 2007
4.6 M Andrews	135° W	L-Band 1685 MHz	GOES-15	GOES-West	2.1 Mb/s	
6.3 M *heated* Patriot	101° W	C-Band *3956 MHz*	SES-1 (24 Ku and 24 C)	MTSAT NOAAPORT Gilmore relay	3.5 Mb/s 10.2 Mb/s 2.6 Mb/s	Installed 2010 (~$55K w/o installation) *Patriot now out of business*
7.3 M Harris	75° W	L-Band 1685 MHz	GOES-13	GOES-East	2.1 Mb/s	Installed in mid 1970s
7.3 M Harris	101° W	C-Band *3956 MHz*	SES-1 (24 Ku and 24 C)	MTSAT NOAAPORT Gilmore relay	3.5 Mb/s 10.2 Mb/s 2.6 Mb/s	Installed in mid 1970s
11 Meter	87° W	C-Band *4-8 GHz*	SES-2 (24 Ku and 24 C)	MSG Wallops relay	1.2 Mb/s 2.6 Mb/s	Installed in early 1980s

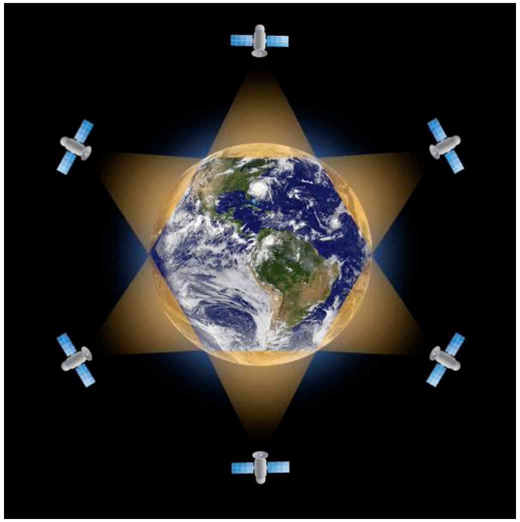


September 2011

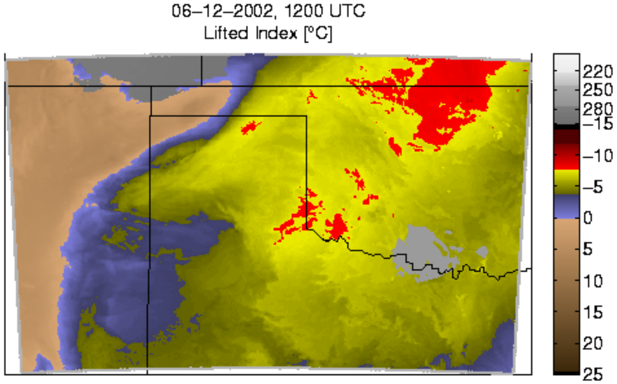
GeoMetWatch-STORM Brings the Advanced Science & Technology Together

Cutting Edge Sensor

Global Coverage

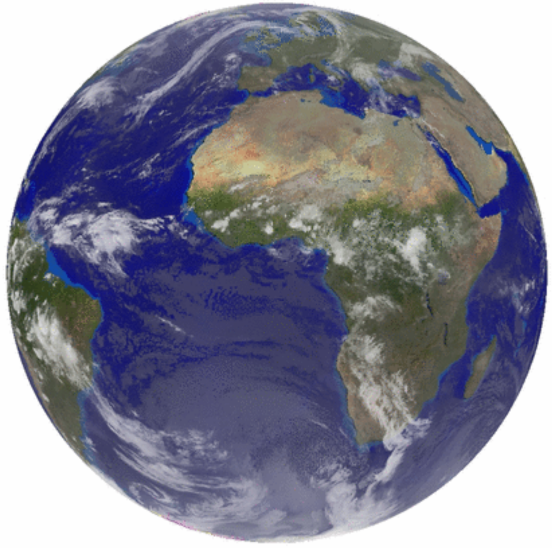


Early Monitoring & Warning

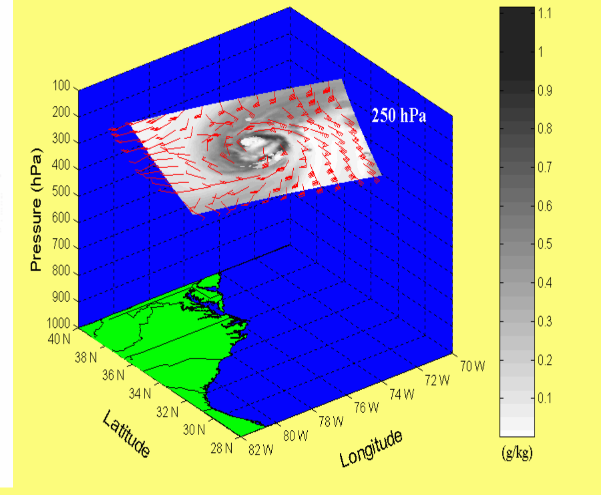


UW/CIMSS

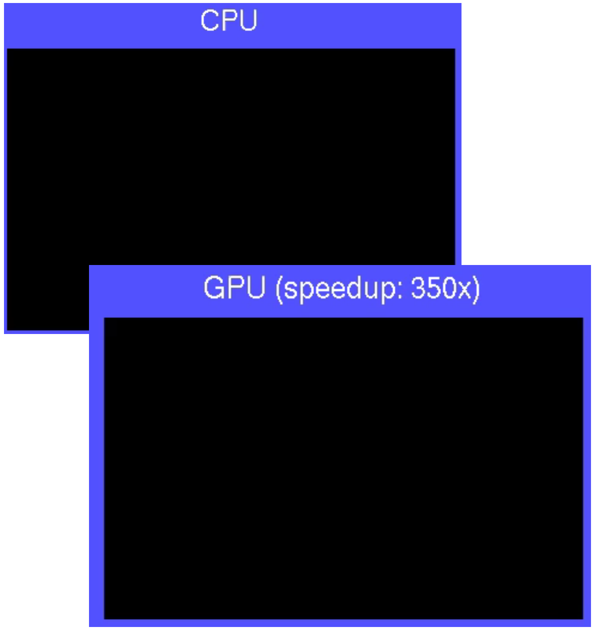
Large Domain High Temporal Observations



High Vertical Resolving Observations



High-Performance Forecasting Technology



Current High Vertical Resolution Satellite Sounding Systems Vs. STORM Characteristic Comparison

Current High Vertical Resolution Satellite Sounding Systems Vs. STORM Characteristic Comparison

Sensor	Observation per day	Field of View Sampling (km)	Temporal Frequency (hour)	ECMWF Forecast Model Error Reduction (%)
COSMIC Taiwan	~1,500 (COSMIC-2: ~10,000)	~200 x 200	>12	~8.5%
AIRS NASA	~2.9 Million	14 x 14	12	<12%
IASI EUMETSAT	~1.3 Million	12 x 12	12	~12%
CrIS NOAA	~29 Million	12 x 12	12	??
STORM-1* GMW	~135 Million per STORM	4 x 4 (2 x 2)	0.25-1.0	~12-15%? (1) ~30-40%? (6)

Note that STORM-1* has much higher observation density, spatial resolution, and much higher measurement frequency than any current high vertical resolution satellite sounding systems. All evidences shown have lead us to believe STORM would enhanced forecast model performance by a huge margin when compares with its countparts!

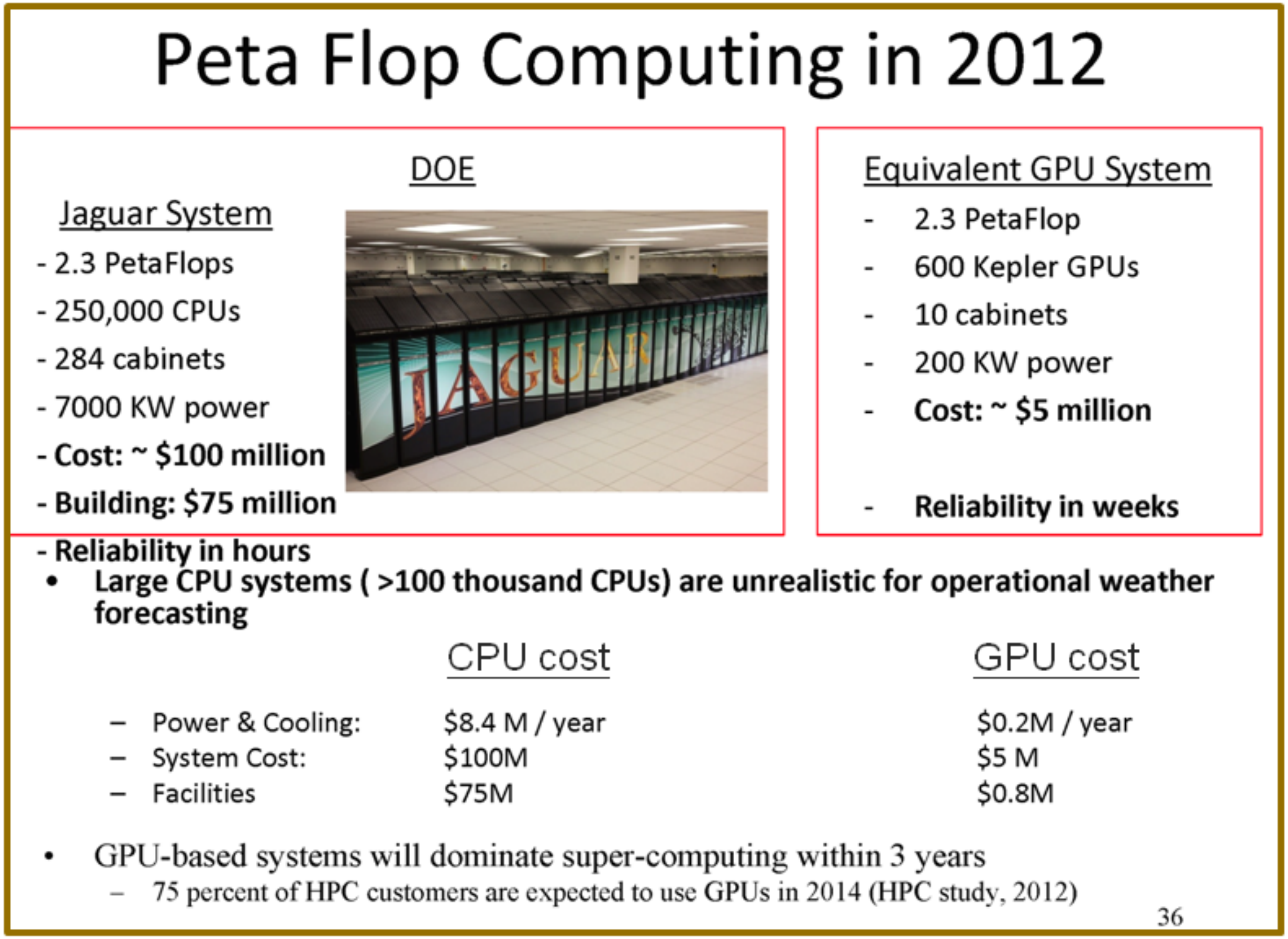
ESRL Global Model Plans

A presentation to:
ECMWF 15th Workshop
On HPC in Meteorology
October 2, 2012

Alexander E. MacDonald
Director
Earth System Research Lab
Boulder, Colorado

Deputy Assistant Administrator
NOAA Research

Slide #36



Peta Flop Computing in 2012

DOE

Jaguar System

- 2.3 PetaFlops
- 250,000 CPUs
- 284 cabinets
- 7000 KW power
- **Cost: ~ $100 million**
- **Building: $75 million**
- **Reliability in hours**

Equivalent GPU System

- 2.3 PetaFlop
- 600 Kepler GPUs
- 10 cabinets
- 200 KW power
- **Cost: ~ $5 million**
- **Reliability in weeks**

- **Large CPU systems (>100 thousand CPUs) are unrealistic for operational weather forecasting**

	CPU cost	GPU cost
– Power & Cooling:	$8.4 M / year	$0.2M / year
– System Cost:	$100M	$5 M
– Facilities	$75M	$0.8M

- GPU-based systems will dominate super-computing within 3 years
  - 75 percent of HPC customers are expected to use GPUs in 2014 (HPC study, 2012)

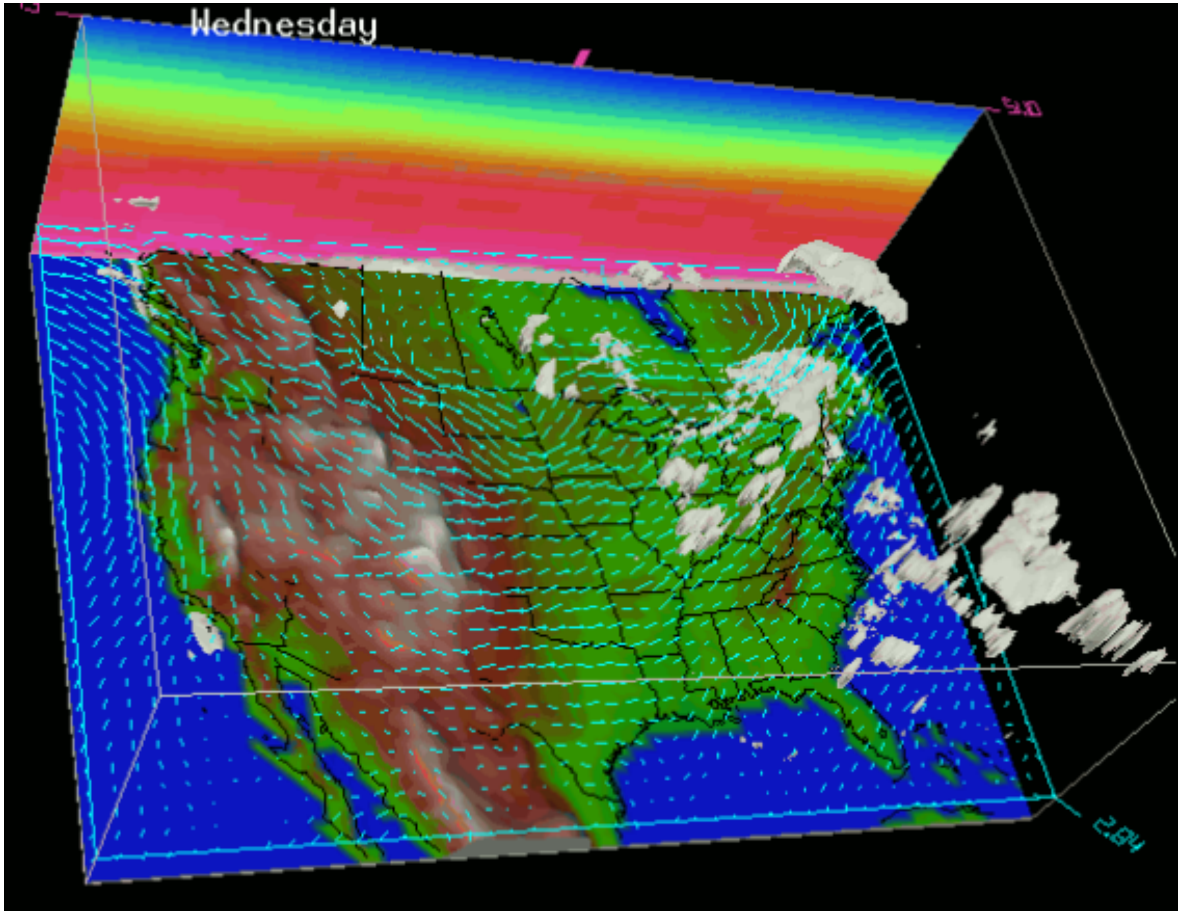
36

SSEC High Performance Computing Publications (2009-2014)

82 Journal & Conference Papers

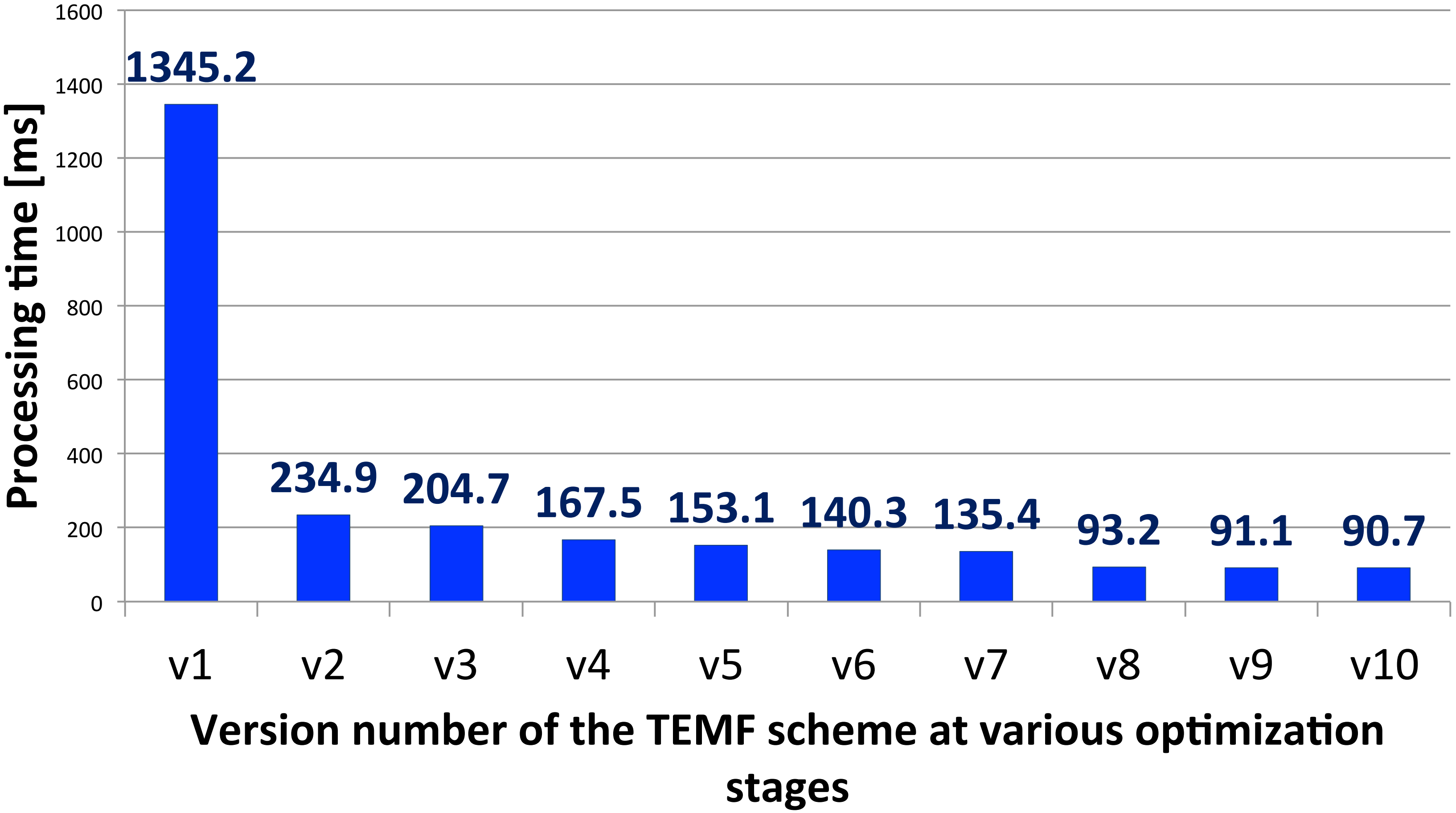
- Journal Papers: 36
  - GPU-based: 26
  - HPC: 10
- Conference Papers: 46
  - GPU-based: 38
  - MIC: 3
  - HPC: 5

CONtinental United States (CONUS) benchmark data set for 12 km resolution domain for October 24, 2001

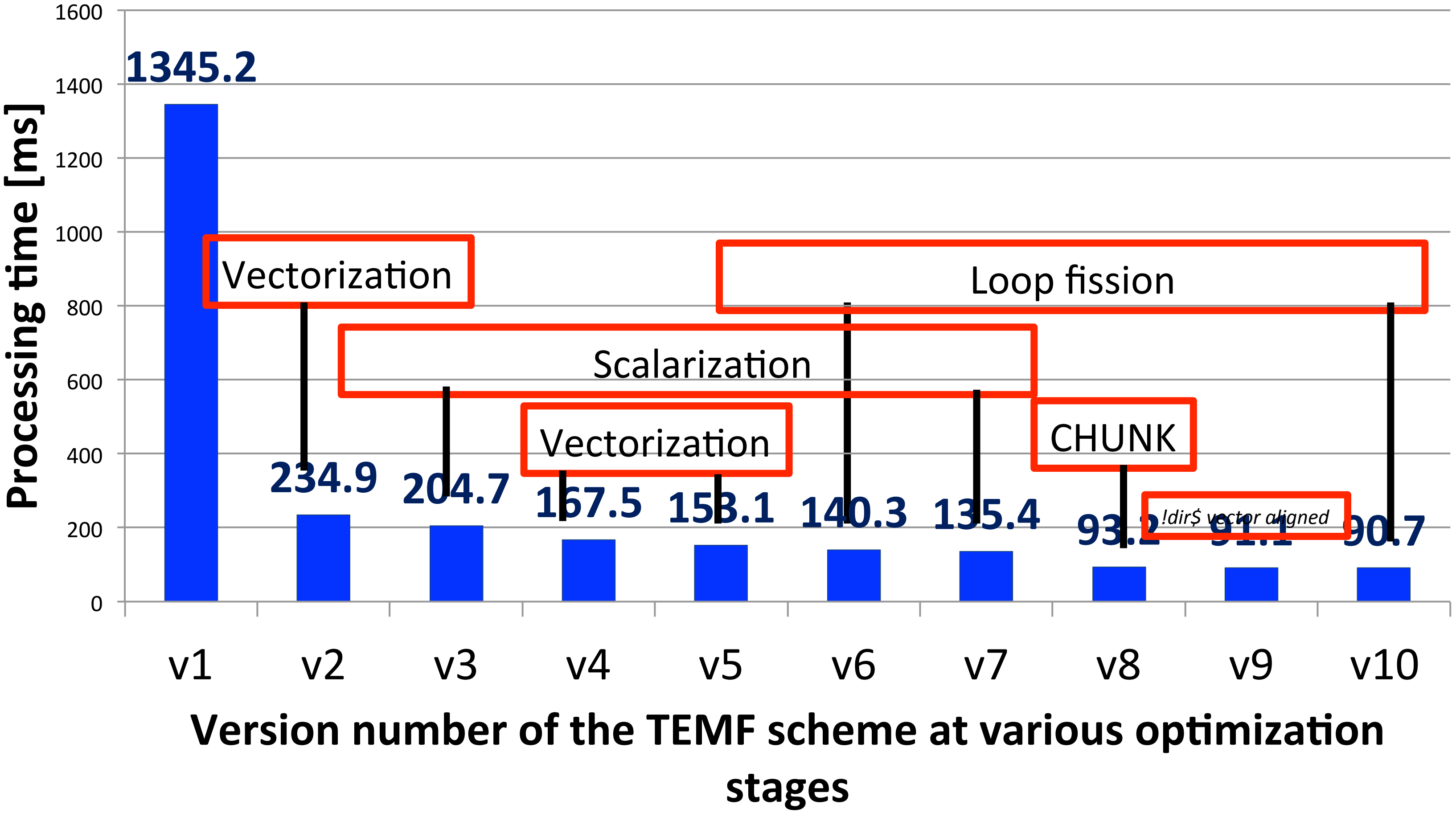


- The size of the CONUS 12 km domain is 433 x 308 horizontal grid points with 35 vertical levels.
- The test problem is a 12 km resolution 48-hour forecast over the Continental U.S. capturing the development of a strong baroclinic cyclone and a frontal boundary that extends from north to south across the entire U.S.

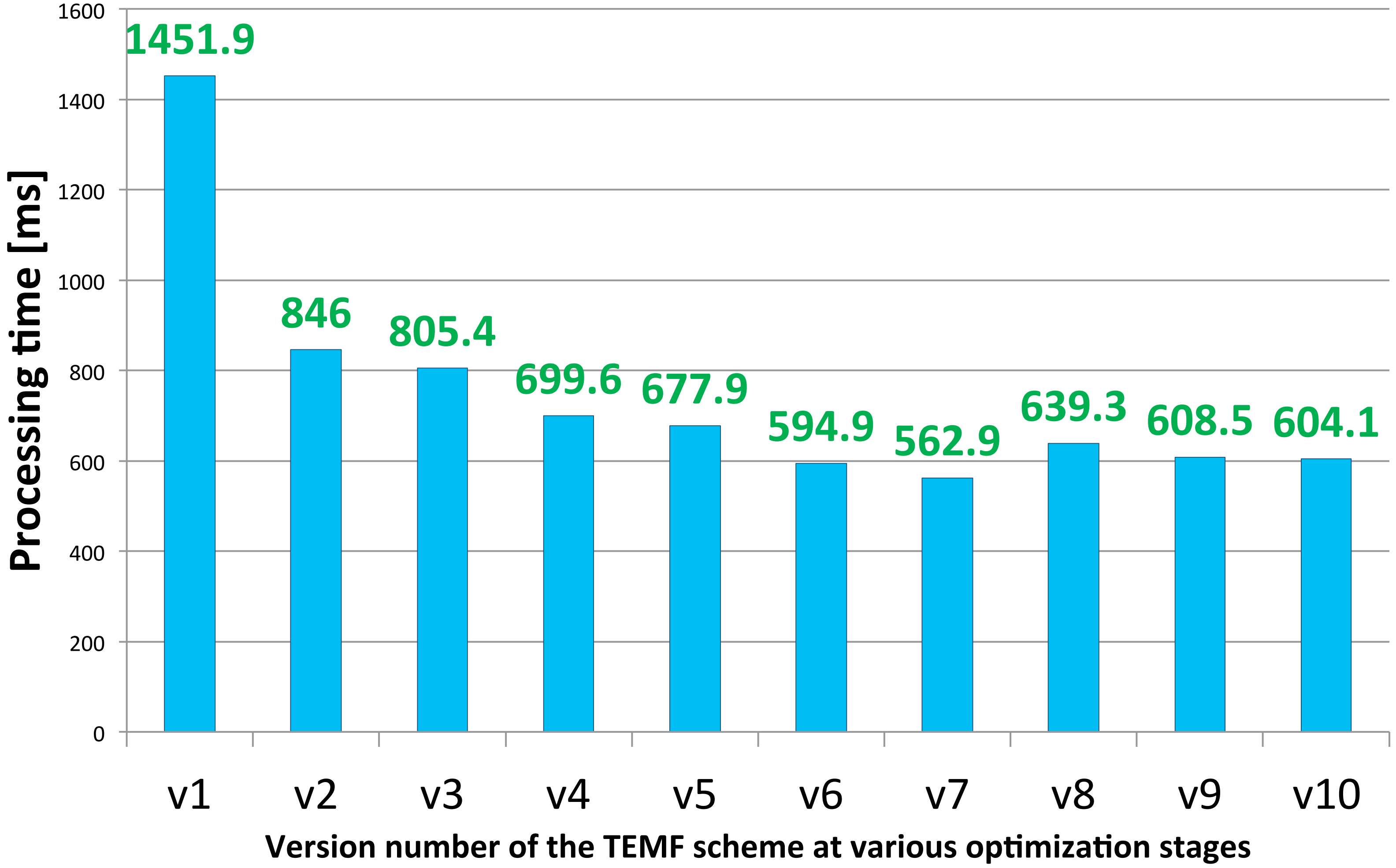
MIC processing time for the TEMF planetary boundary layer scheme



MIC processing time for the TEMF planetary boundary layer scheme



CPU processing time for the TEMF planetary boundary layer scheme

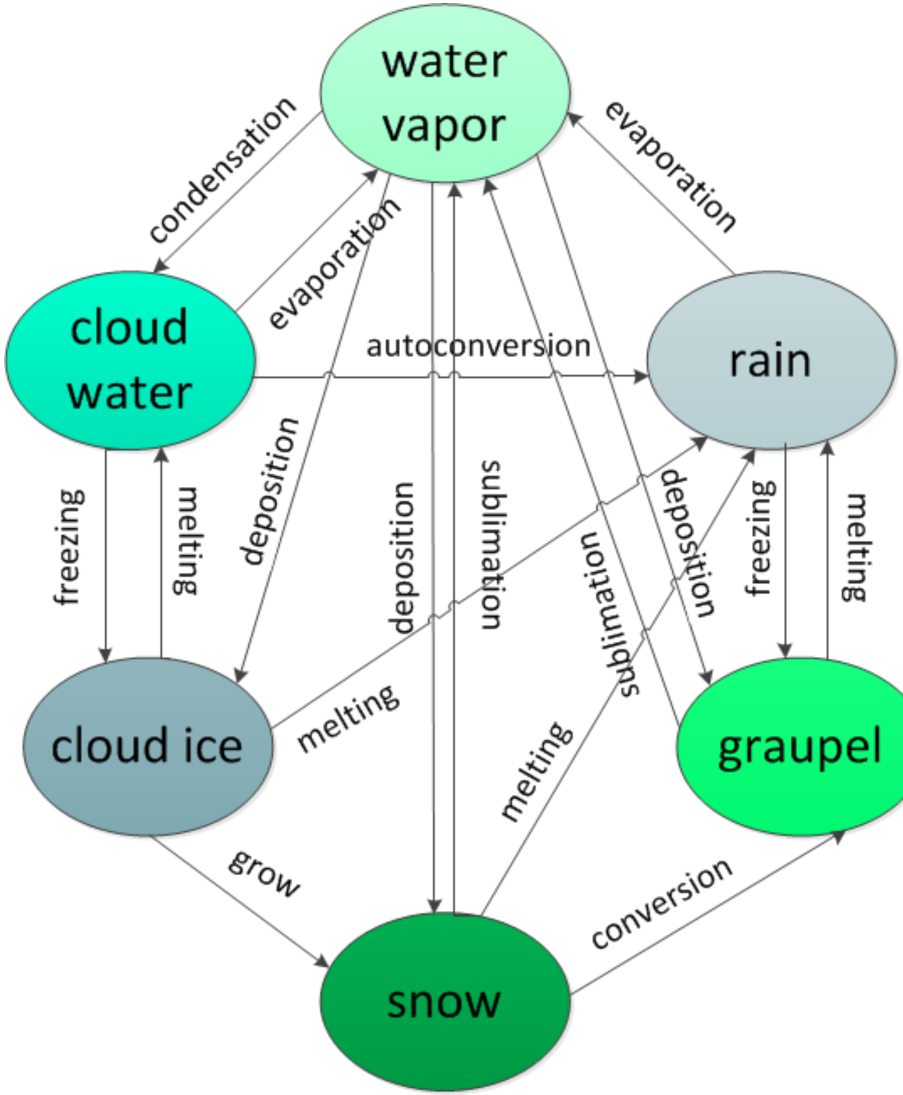


Thompson microphysics scheme

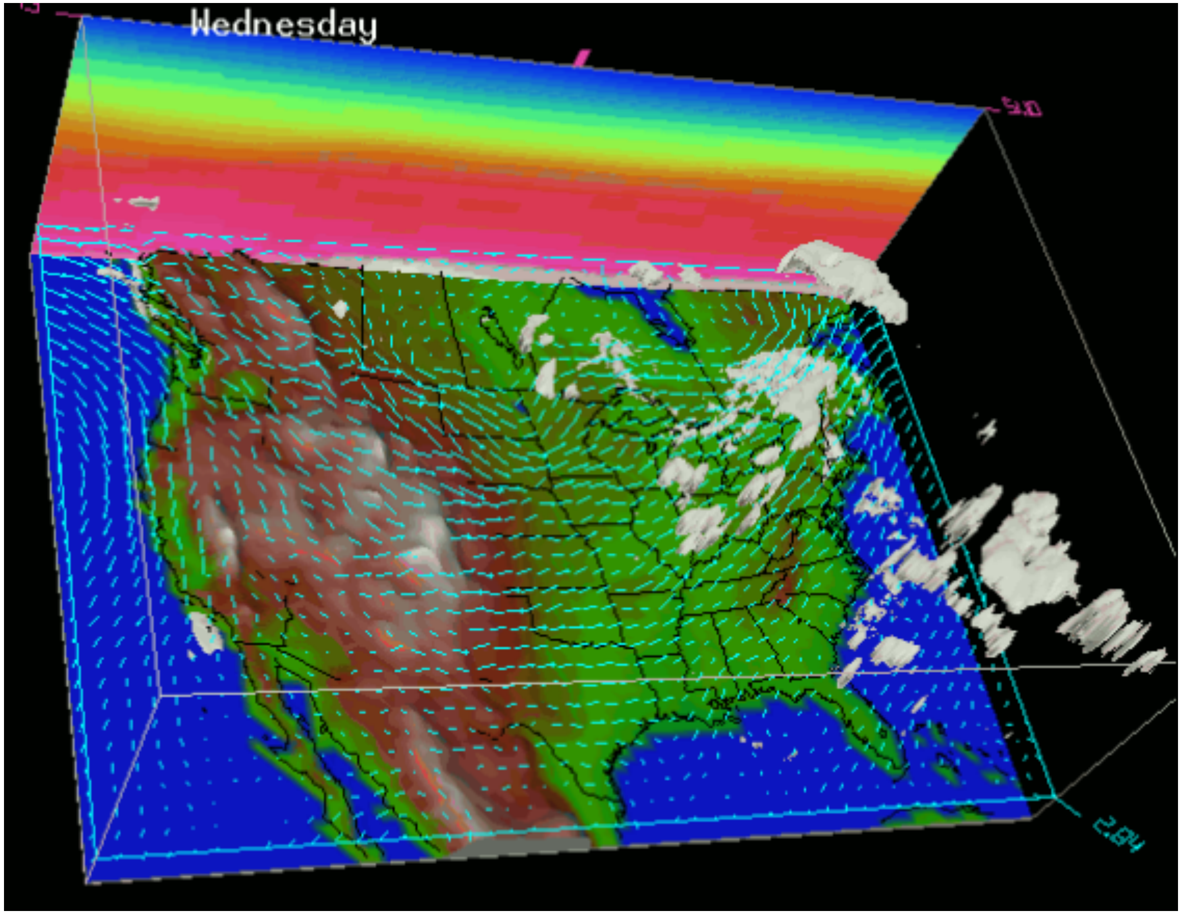
◆WRF v2.2 incorporated the new Thompson microphysics scheme

◆Includes water vapor, cloud water, rain, cloud ice, graupel, and snow

1. Water and ice species
2. Microphysical process

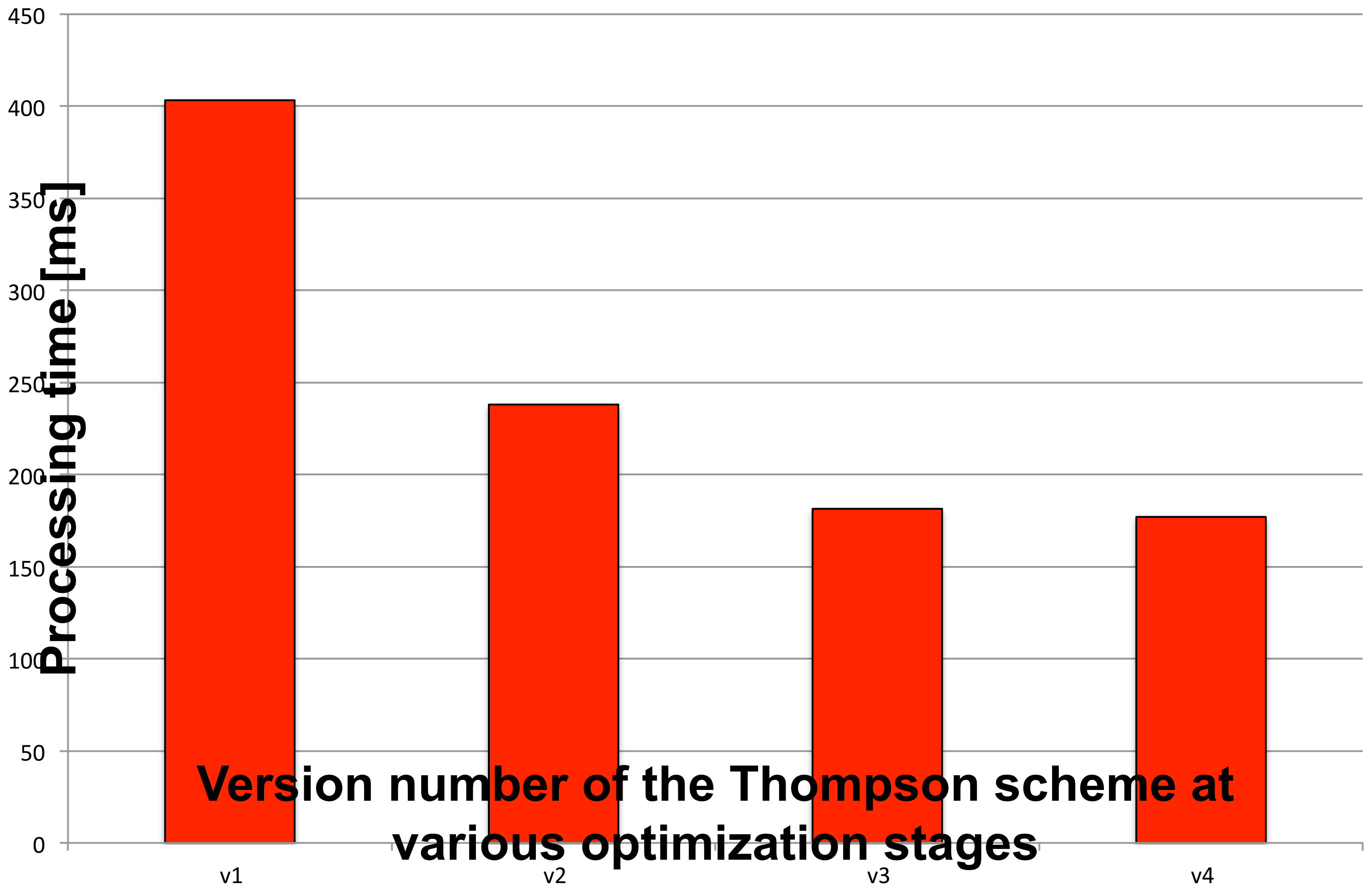


CONtinental United States (CONUS) benchmark data set for 12 km resolution domain for October 24, 2001

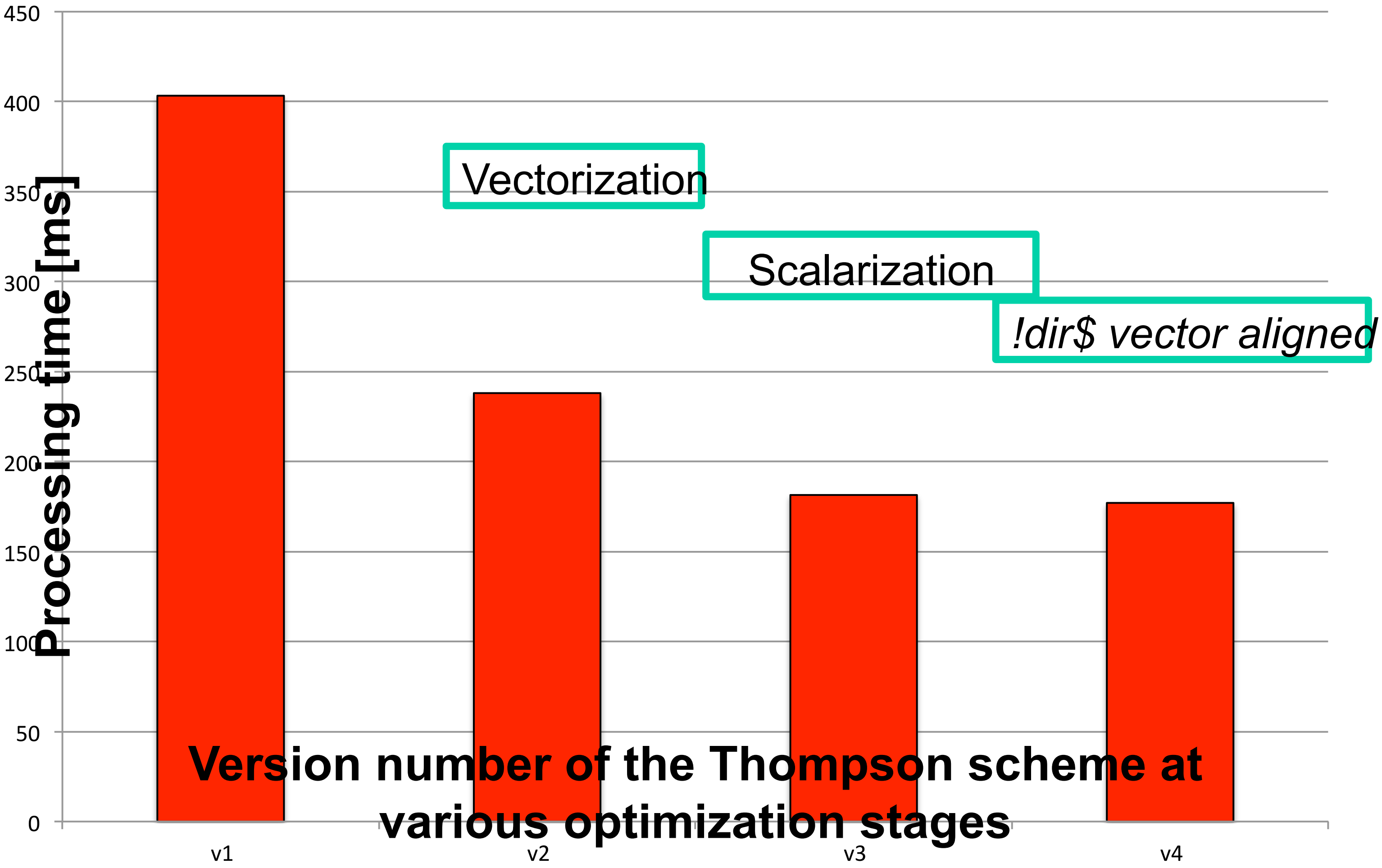


- The size of the CONUS 12 km domain is 433 x 308 horizontal grid points with 35 vertical levels.
- The test problem is a 12 km resolution 48-hour forecast over the Continental U.S. capturing the development of a strong baroclinic cyclone and a frontal boundary that extends from north to south across the entire U.S.

MIC processing time for the Thompson microphysics scheme



MIC processing time for the Thompson microphysics scheme



Vectorization

Scalarization

!dir$ vector aligned

5-Layer Thermal Diffusion Scheme - 1

- **Land Surface Models (LSMs) in WRF**
  -- are used to provide **heat** and **moisture** fluxes over land and sea-ice points – see **land surface process** illustration
- **5-layer thermal diffusion scheme** is one of LSMs based on MM5 with an energy budget made up of **sensible, latent,** and **radiative heat** fluxes

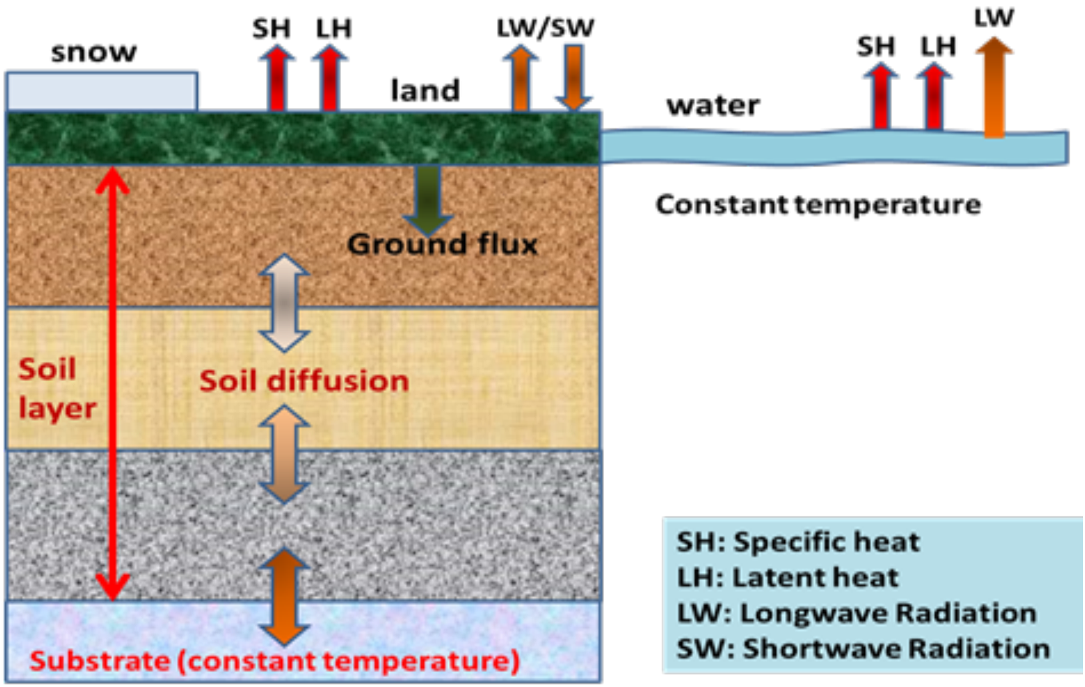


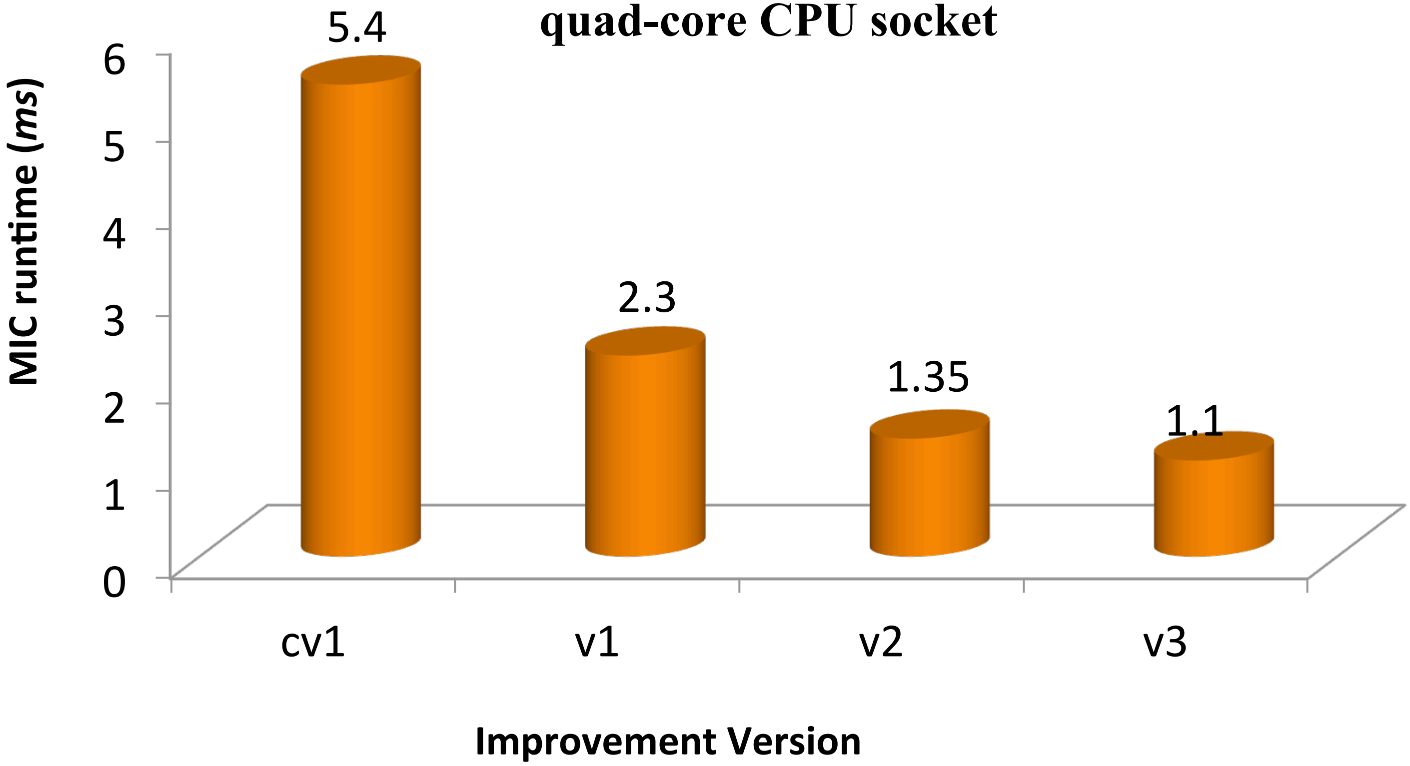
Illustration of Land Surface process

Reference:

(1) MM5 (Mesoscale Models)
(2) J. Dudhia, PSU/NCAR, 4950, 1996

Optimization on 5-Layer Thermal Diffusion Scheme – 5

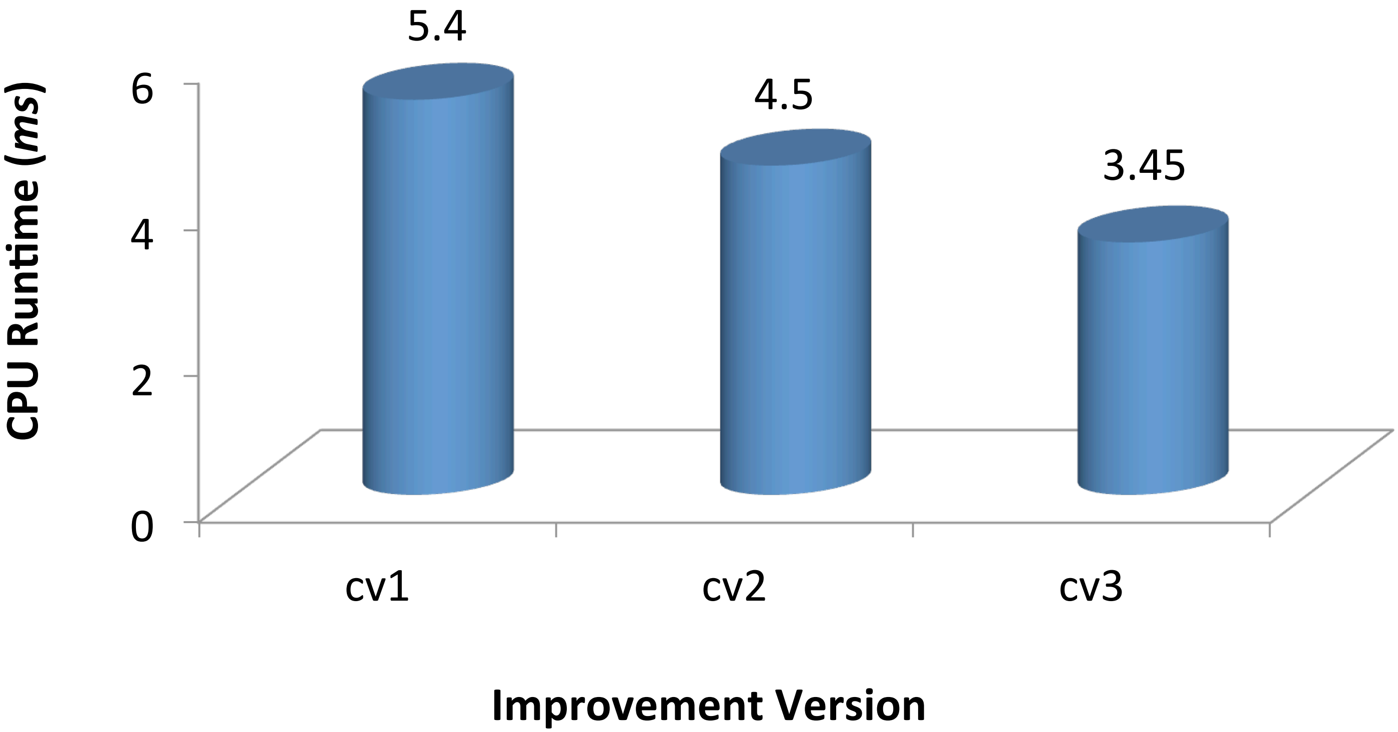
-- MIC-based Runtime Summary --



cv1: multi-threaded code running at one quad-core CPU socket

Optimization on 5-Layer Thermal Diffusion Scheme – 6

-- CPU-based Runtime Summary --



Optimization on 5-Layer Thermal Diffusion Scheme – 4

-- Runtime & Speedup Summary --

CPU/MIC-based code runtimes and speedups for various versions, where the **improvement factors are compared to cv1 and v1 respectively.**

Version	**Description**	**Runtime (*ms*)**	**Improvement factor**
cv1	**The first-version multi-threaded code running at one quad-core CPU socket with OpenMP before any improvement**	**5.4**	
v1	**The first-version multi-threaded code running at MIC before any improvement**	**2.3**	
cv2 / v2	**Multi-threading OpenMP, i-loop fusion running on one-quad core CPU socket and MIC**	**4.5 / 1.35**	
cv3 / v3	**+ data process in parallel with *CHUNK* = 64, and add !dir$ vector aligned, add !DIR$ SIMD**	**3.45 / 1.1**	**CPU-based: 1.6x** **MIC-based: 2.1x**

Yonsei University Planetary Boundary Layer (YSU PBL) Scheme - 1

- **YSU scheme** is one of PBL in WRF – see **PBL process** illustration
- **PBL process**

  -- is responsible for **vertical sub-grid-scale fluxes due to eddy transports** in the whole atmospheric column

  -- determines **flux profiles** within boundary and stable layers

  -- provides **atmospheric tendencies** of temp., moisture, horizontal momentum etc.

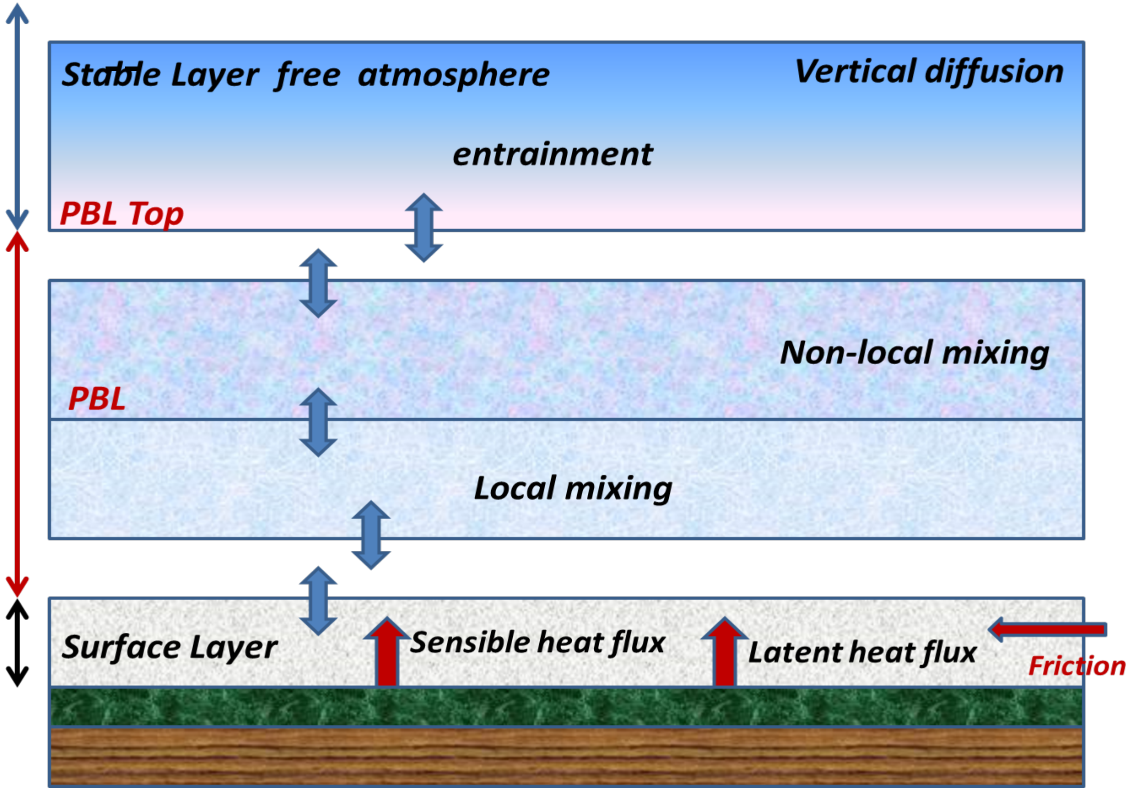
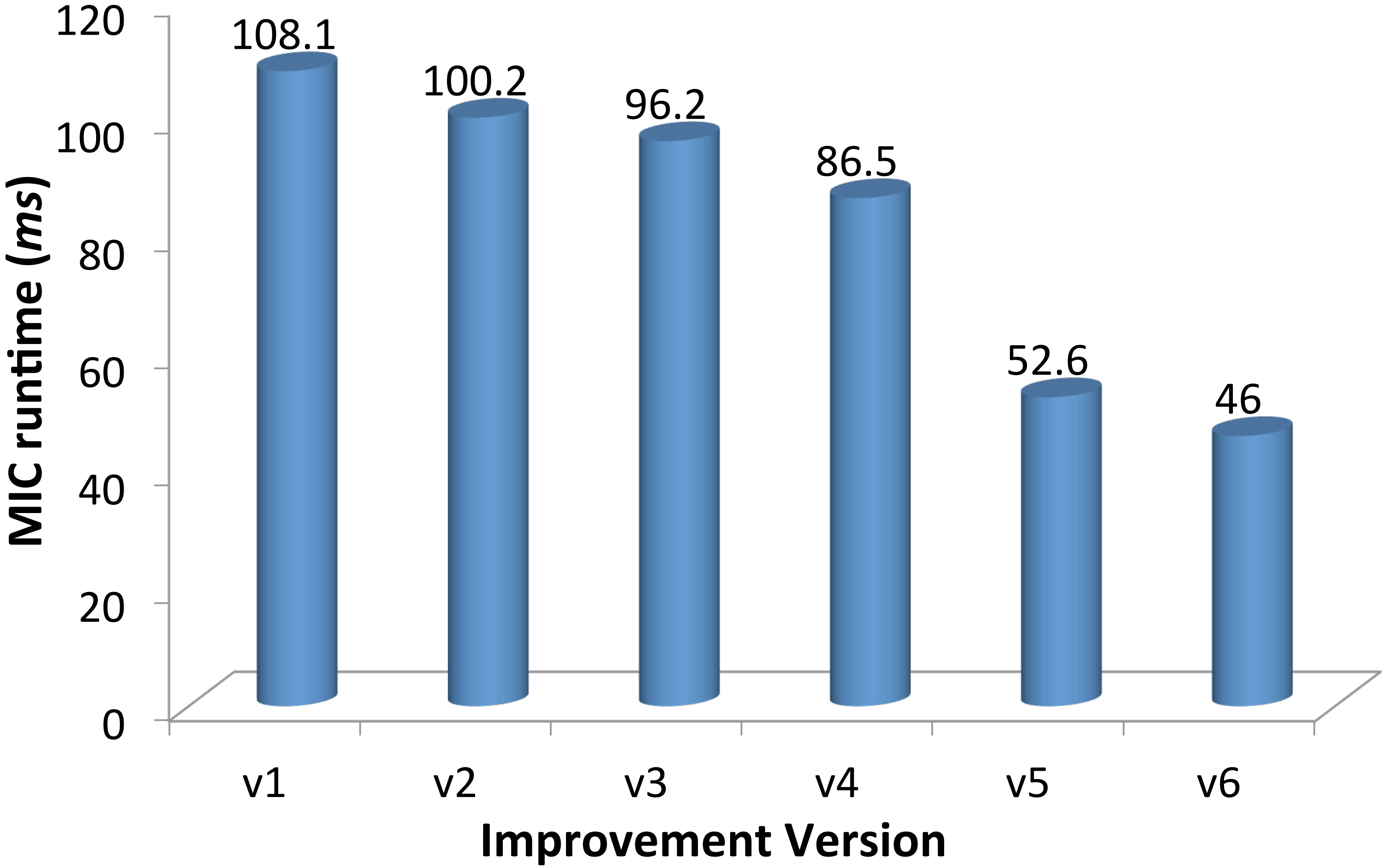


Illustration of PBL process

Reference:

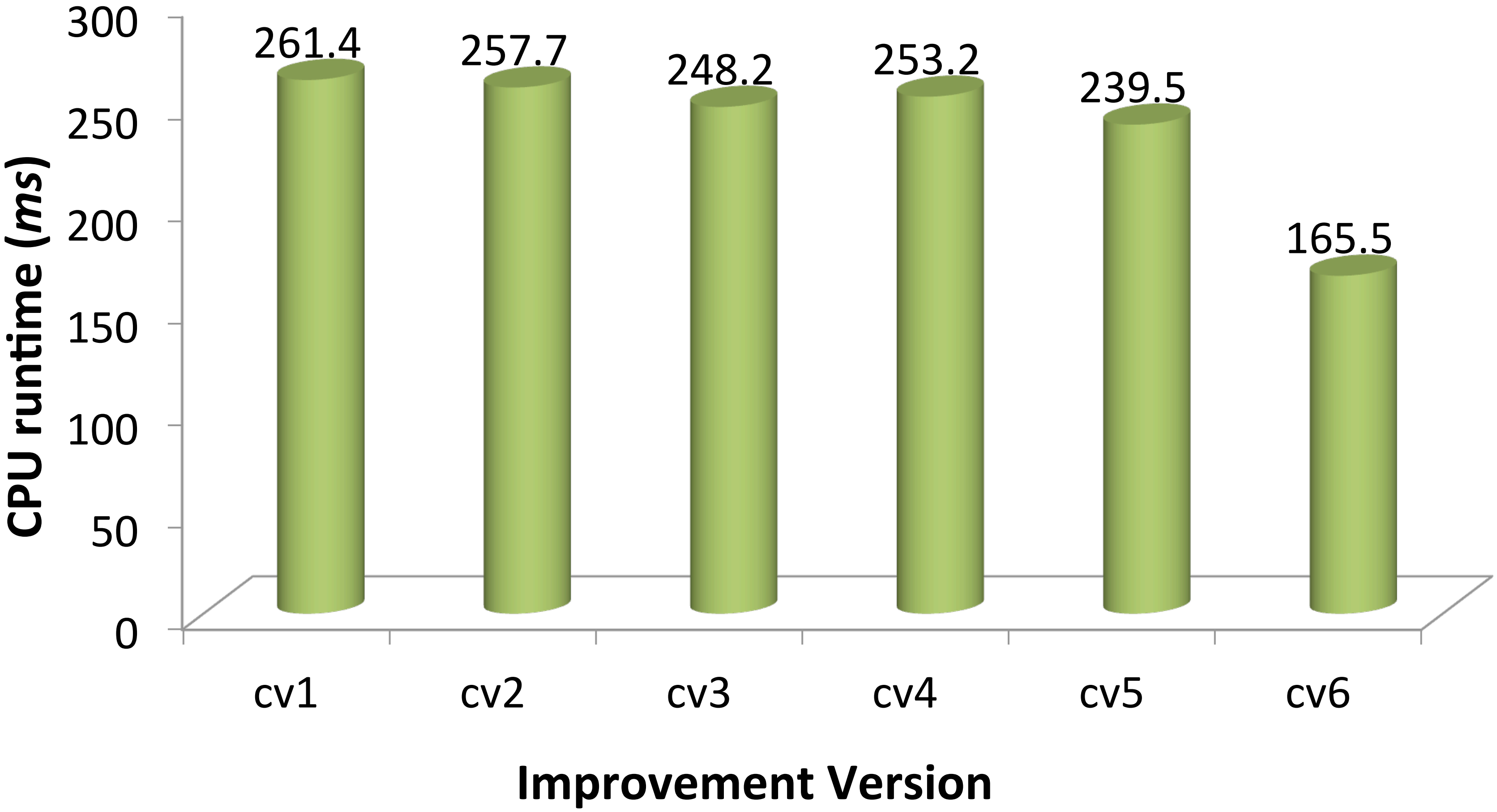
Hong, Noh, and Dudhia, Monthly Weather Review, **134**, 2318-2341, 2006

Optimization on YSU PBL – 8
-- MIC-based Runtime Summary --



Optimization on YSU PBL – 9

-- CPU-based Runtime Summary --



Note: our focus is to optimize the code with Intel MIC architecture, and thus sometimes it may have an impact on its performance when running on CPU.

Optimization on YSU PBL – 7

-- MIC code Runtime & Speedup Summary --

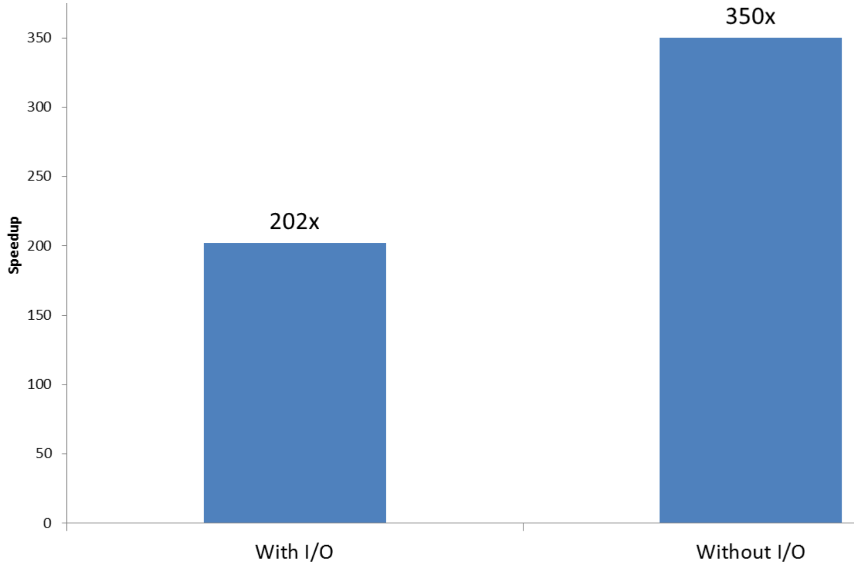
Improvement factors s are compared to cv1 and v1 respectively

Version	Description	Runtime (ms)	Speedup
cv1	The first-version multi-threaded code running on one quad-core CPU socket with OpenMP before any improvement	261.4	
v1	The first-version multi-threaded code running at MIC before any optimization	108.1	
cv2/v2	Multithreading OpenMP + loop fusion	257.7/100.2	
cv3/v3	+ SIMD	248.2/ 96.2	
cv4/v4	+ vectorization for subroutines "*tridi1n*" and "*tridin_ysu*"	253.2/ 86.5	
cv5/v5	+ data process in parallel with *CHUNK* = 64, and add !dir$ vector aligned, add !DIR$ SIMD	239.5/ 52.6	
cv6/v6	+ data process in parallel with *CHUNK* = 112	165.5/ 46.0	(1.6x) (2.4x)

Processing times – CPU Vs. GPU
Early Result (2009)

	Time [ms]
The original Fortran code on CPU	16928
CUDA C with I/O on GPU	83.6
CUDA C without I/O on GPU	48.3

Our experiments on the Intel i7 970 CPU running at 3.20 GHz and a single GPU out of two GPUs on NVIDIA GTX 590



The Fast Radiative Transfer Model

Without losing the generality of our GPU implementation, we consider the following radiative transfer model:

$$R_v = \varepsilon_v B_v(T_s)\,\tau_v(p_s) - \int_0^{p_s} B_v\left[T(p)\right]\frac{d\tau_v(p)}{dp}\,dp$$

with the regression-based transmittances:

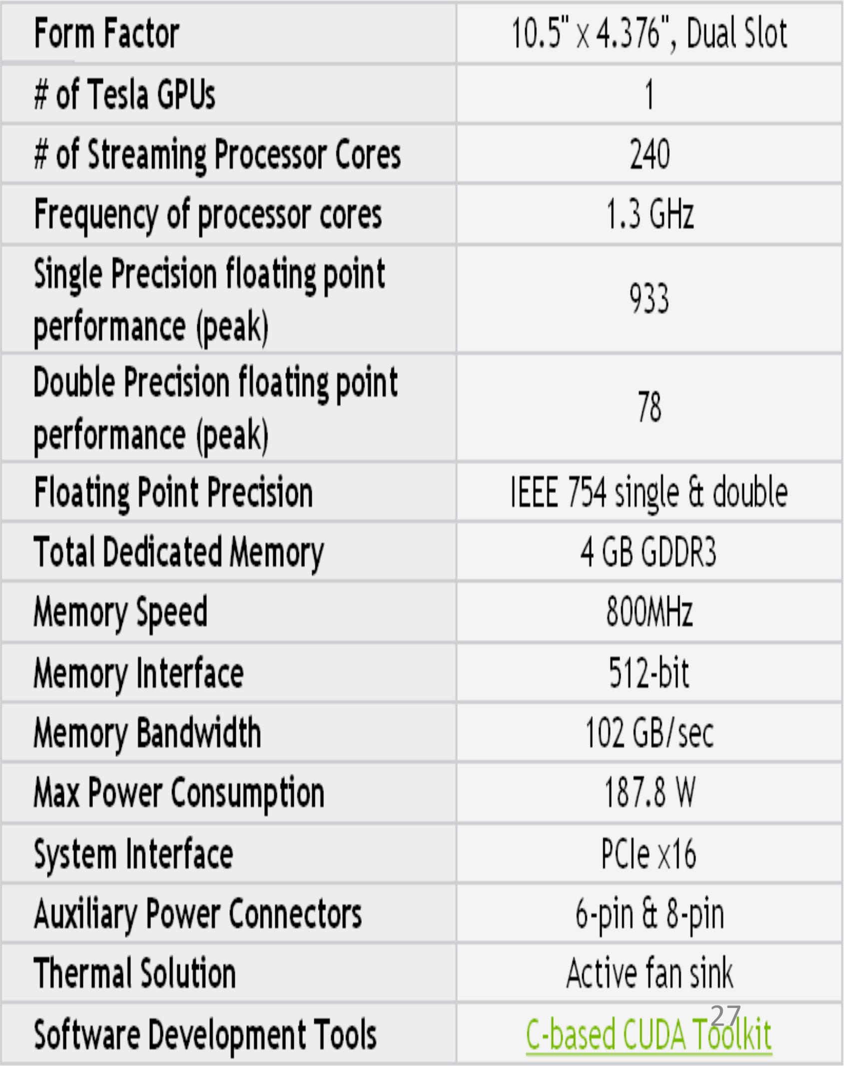
$$\begin{aligned}\tau_v(p_j) &= \tau_v^{\mathrm{Dry}}(p_j)\,\tau_v^{\mathrm{H_2O}}(p_j)\,\tau_v^{\mathrm{O_3}}(p_j)\,\tau_v^{\mathrm{CH_4}}(p_j)\,\tau_v^{\mathrm{CO}}(p_j)\\ &= \exp\left\{\sum_{k=1}^{j}\left[\sum_{l_d=1}^{m_f} C_{v l_d k}^{\mathrm{Dry}}\, X_{l_d k}^{\mathrm{Dry}} + \sum_{l_w=1}^{m_w} C_{v l_w k}^{\mathrm{H_2O}}\, X_{l_w k}^{\mathrm{H_2O}} + \sum_{l_o=1}^{m_o} C_{v l_o k}^{\mathrm{O_3}}\, X_{l_o k}^{\mathrm{O_3}} + \sum_{l_m=1}^{m_m} C_{v l_m k}^{\mathrm{CH_4}}\, X_{l_m k}^{\mathrm{CH_4}} + \sum_{l_c=1}^{m_c} C_{v l_c k}^{\mathrm{CO}}\, X_{l_c k}^{\mathrm{CO}}\right]\right\}\end{aligned}$$

Our GPU forward model is running on a low-cost personal super computer (~US$7000).

It has a quad-core 2.4 GHz AMD CPU, and 4 Nvidia Tesla 1.3 GHz GPUs with total 960 cores.

ServMax PSC-2 **960-Core** Personal Supercomputer

- **250 times faster** than Standard PCs and Workstations
- **4 Teraflops** of Compute Capability
- Delivering Cluster Level Computing Performance at Your Desk.



Form Factor	10.5" x 4.376", Dual Slot
# of Tesla GPUs	1
# of Streaming Processor Cores	240
Frequency of processor cores	1.3 GHz
Single Precision floating point performance (peak)	933
Double Precision floating point performance (peak)	78
Floating Point Precision	IEEE 754 single & double
Total Dedicated Memory	4 GB GDDR3
Memory Speed	800MHz
Memory Interface	512-bit
Memory Bandwidth	102 GB/sec
Max Power Consumption	187.8 W
System Interface	PCIe x16
Auxiliary Power Connectors	6-pin & 8-pin
Thermal Solution	Active fan sink
Software Development Tools	C-based CUDA Toolkit

RTTOV-7 GPU Work Update
(7/20/2010)

Tasks finished (single GPU version):

1. The single-input single-GPU RTTOV-7 IASI code (for computing one 8461-channel IASI radiance spectrum on 1 GPU), with 180x speedup.
2. The multi-input single-GPU RTTOV-7 IASI code (for computing five 8461-channel IASI radiance spectra on 1 GPU), with 368x speedup.

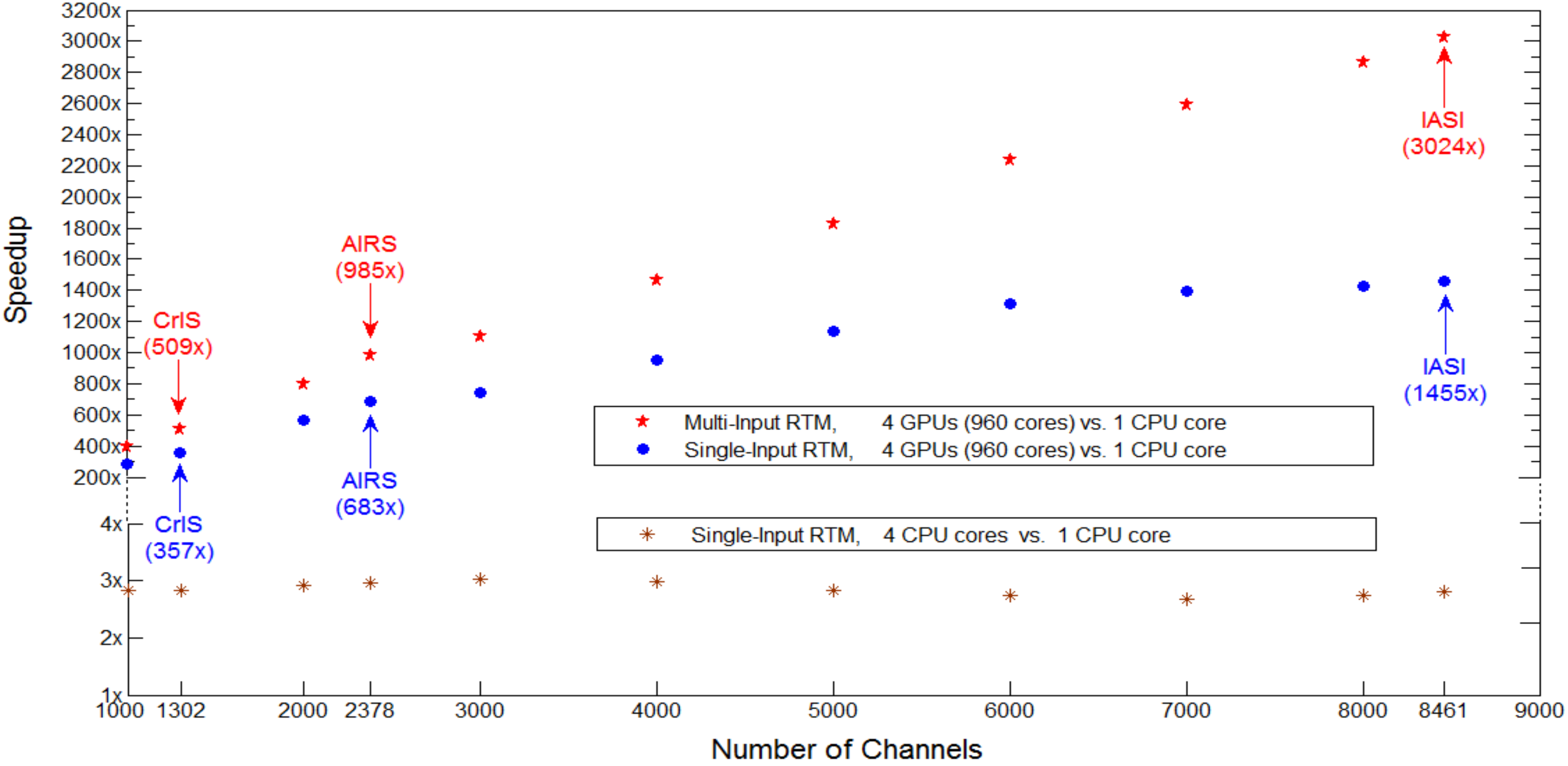
Execution Time in milliseconds (ms)	RTTOV-7 (single-input GPU)	RTTOV-7 (multi-input GPU)
1 CPU core	195 ms	195 ms
1 GPU (240 cores)	1.083 ms	0.53 ms
GPU Speedup	180x	368x

Note:

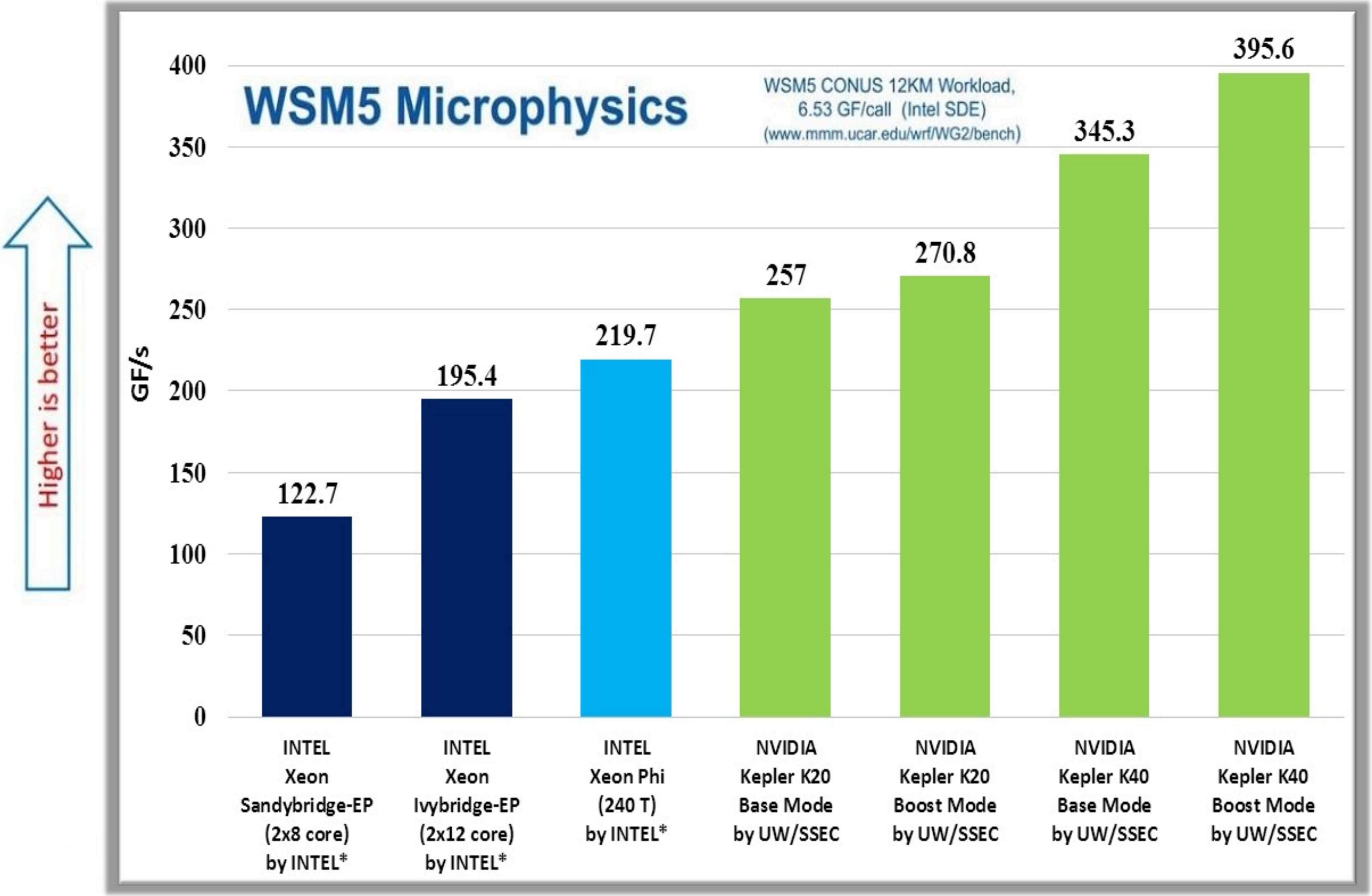
- ➢ Benchmark performed in a low-cost (~US$7000) personal computer with 1 quad-core 2.4 GHz AMD CPU and four 1.3 GHz 240-core NVIDIA Tesla GPUs.
- ➢ GPU speedup was with respect to 1 CPU core performance. The CPU code was compiled using gfortran with –O2 compiler switch.
- ➢ To compute one day's amount of 1,296,000 IASI spectra, the CPU code will take 2.925 days, whereas the single-input & multi-input GPU codes will take 23.39 & 11.44 minutes, respectively.

GPU-based Multi-input RTM

- ➢ A forward model to concurrently compute 40 radiance spectra was further developed to take advantage of GPU's massive parallelism capability.



To compute one day's amount of 1,296,000 IASI spectra,
the original RTM (with –O2 optimization) will take ~10 days on a 3.0 GHz CPU core;
the single-input GPU-RTM will take ~ 10 minutes (with 1455x speedup), whereas
the multi-input GPU-RTM will take ~ 5 minutes (with 3024x speedup).



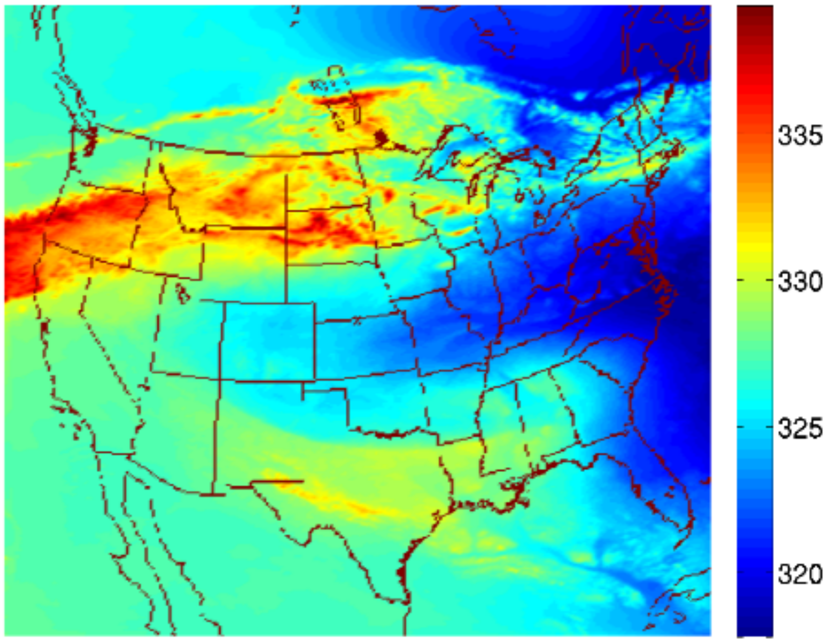
* *Code Restructuring to Improve Performance in WRF Model Physics on Intel Xeon Phi*. J. Michalakes. Workshop on Programming Weather, Climate and Earth System Models on Heterogeneous Multi-core Platforms, Boulder, Colorado, Sept. 19-20, 2013. (http://data1.gfdl.noaa.gov/multi-core/presentations/michalakes_5.pdf)

Code Validation

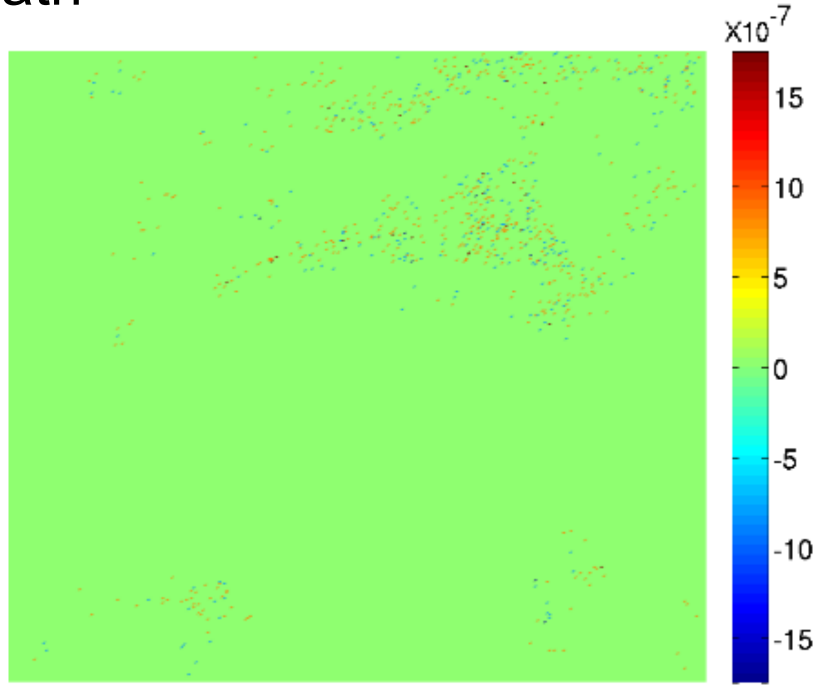
-Fused multiply-addition was turned off (--fmad=false)

-GNU C math library was used on GPU, i.e. powf(), expf(), sqrt() and logf() are replaced by library routines from GNU C library
-> bit-exact output

-Small output differences for –fast-math

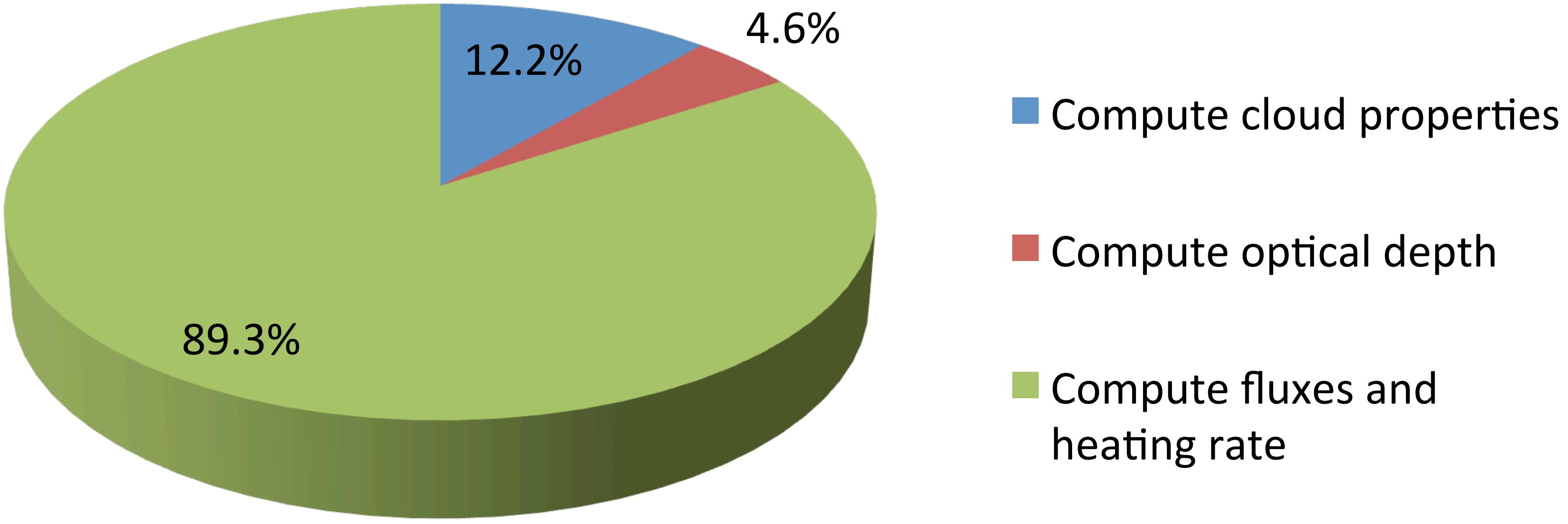


Potential temperature

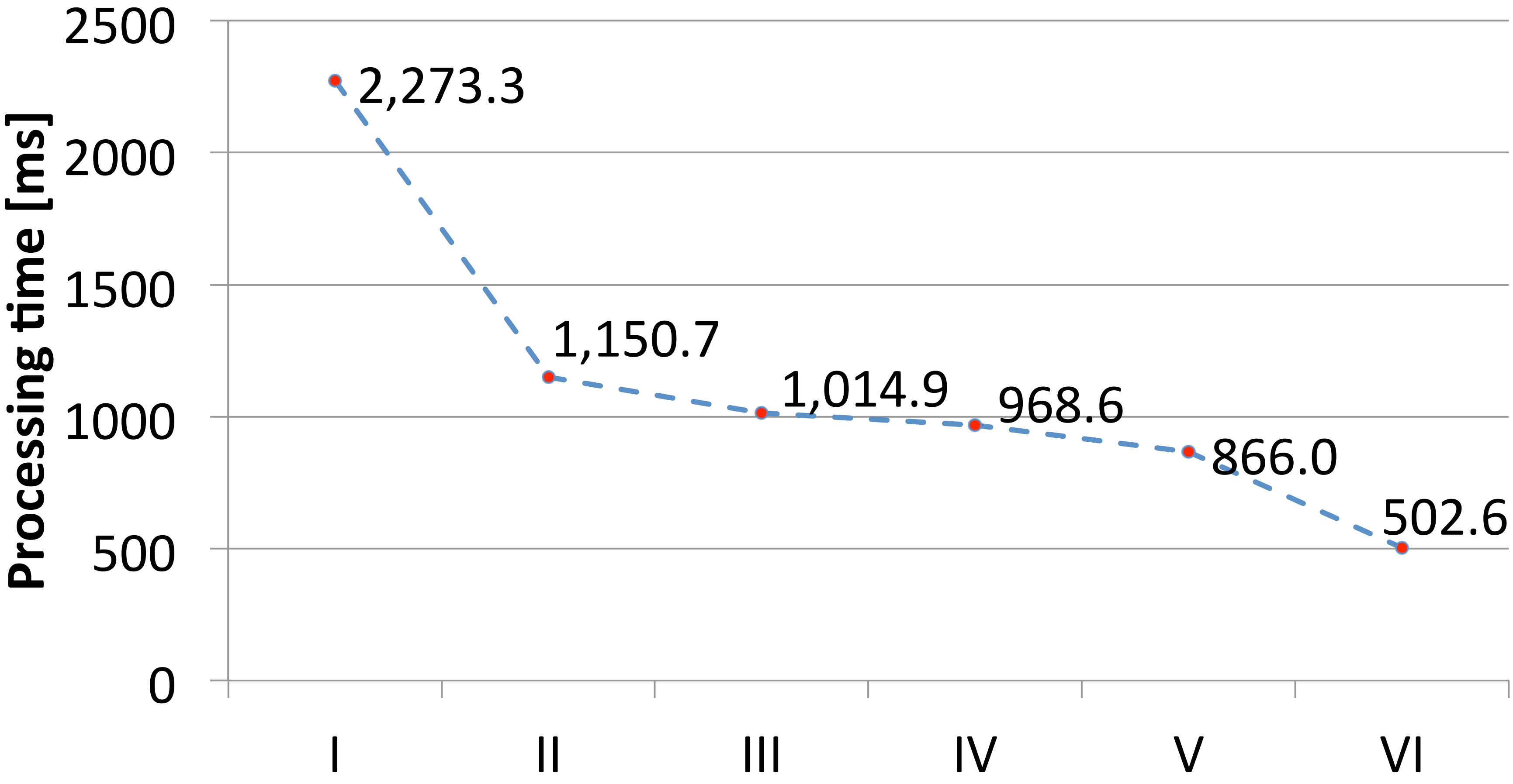


Difference between CPU and GPU outputs

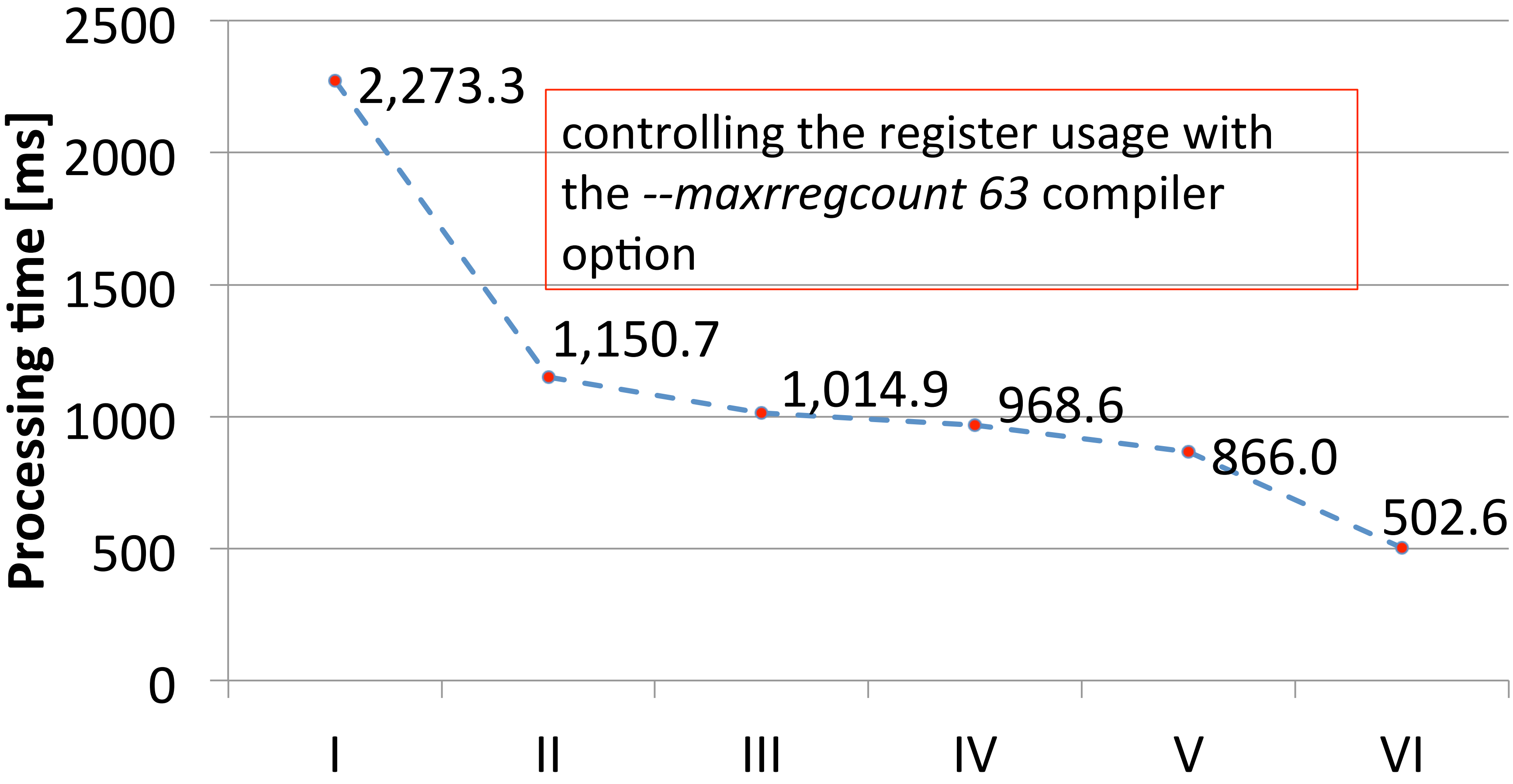
RRTMG_SW computing time on a CPU



RRTMG_SW processing time for a six GPU optimizations steps

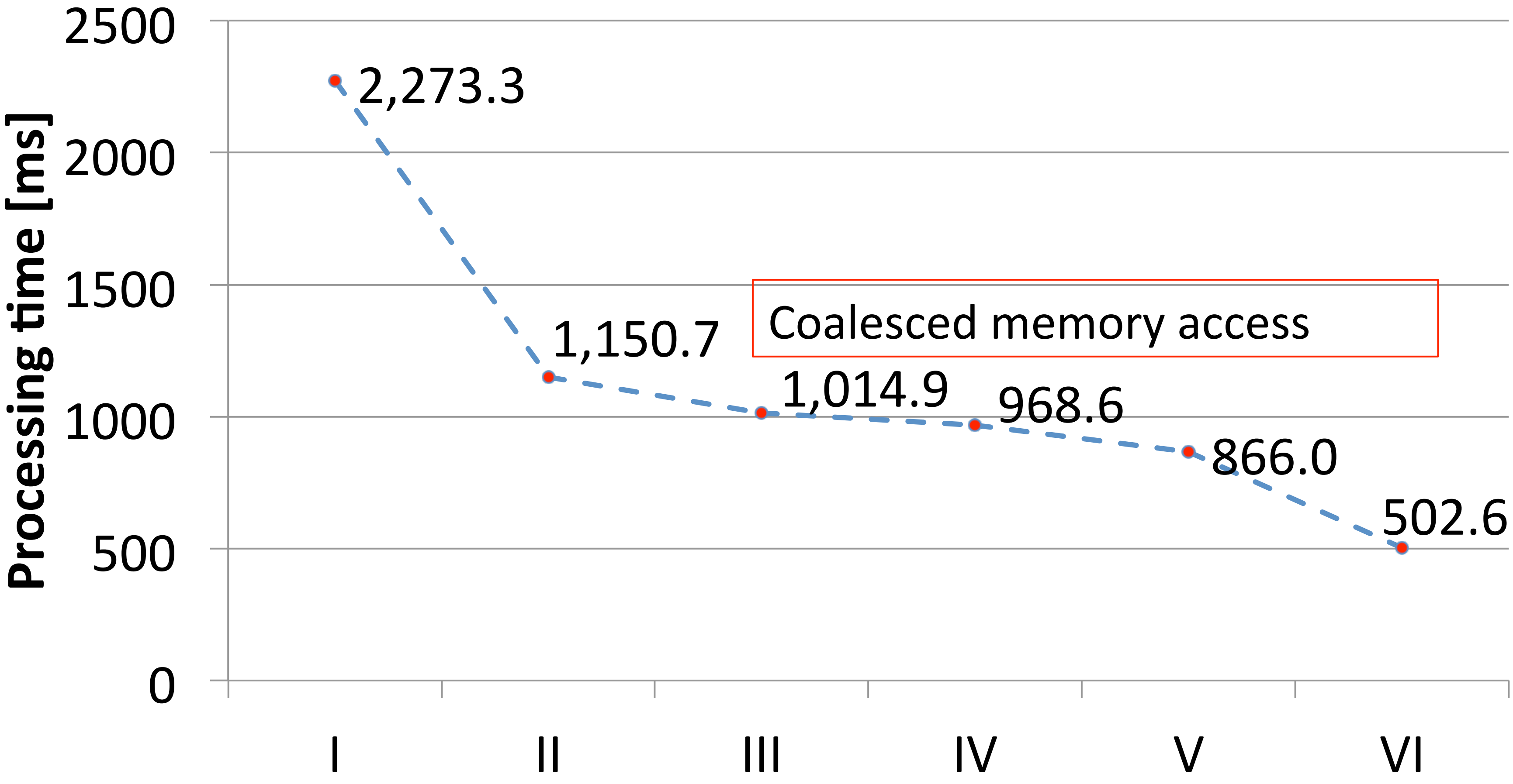


RRTMG_SW processing time for a six GPU optimizations steps

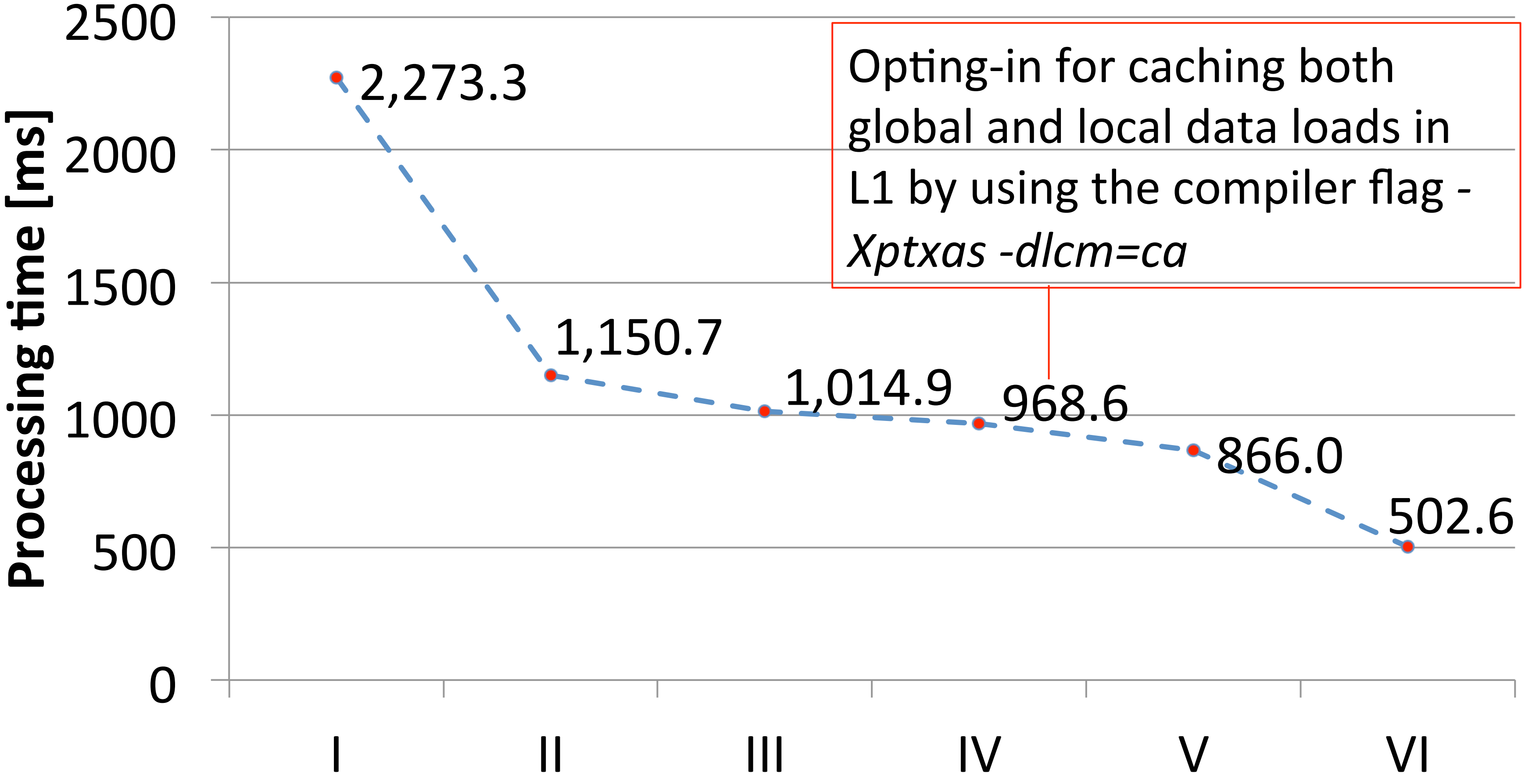


controlling the register usage with the *--maxrregcount 63* compiler option

RRTMG_SW processing time for a six GPU optimizations steps

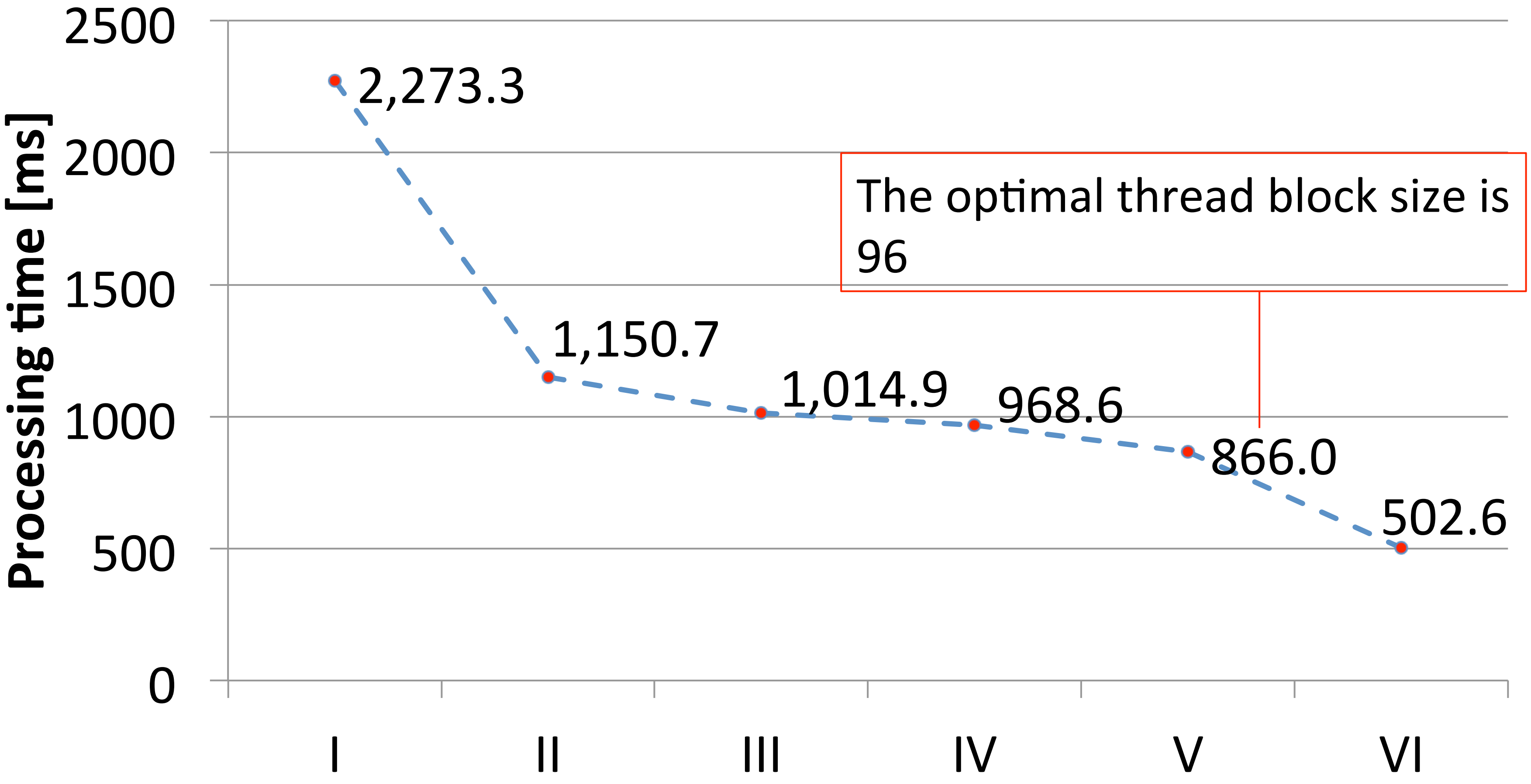


RRTMG_SW processing time for a six GPU optimizations steps



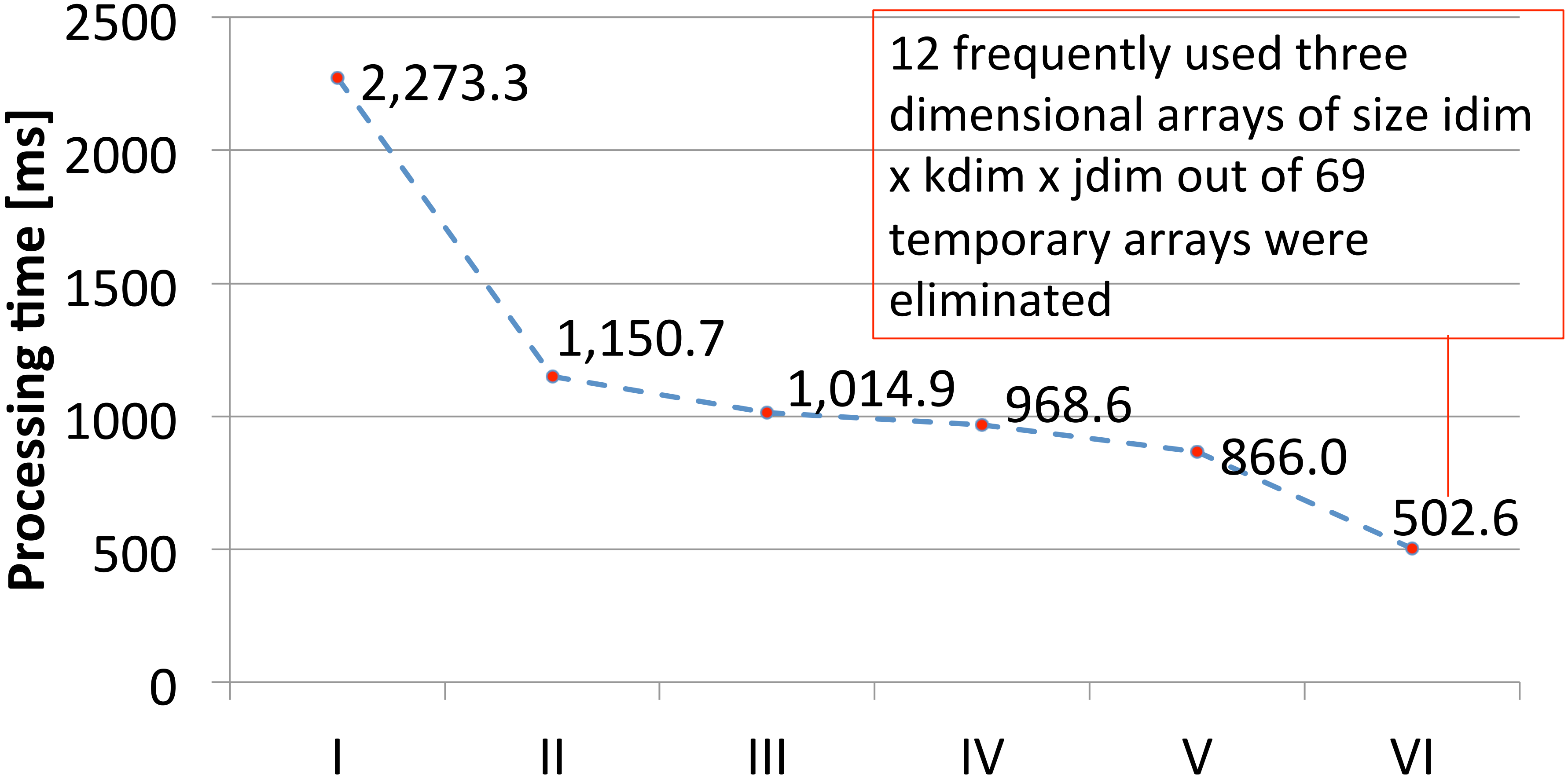
Opting-in for caching both global and local data loads in L1 by using the compiler flag -*Xptxas -dlcm=ca*

RRTMG_SW processing time for a six GPU optimizations steps



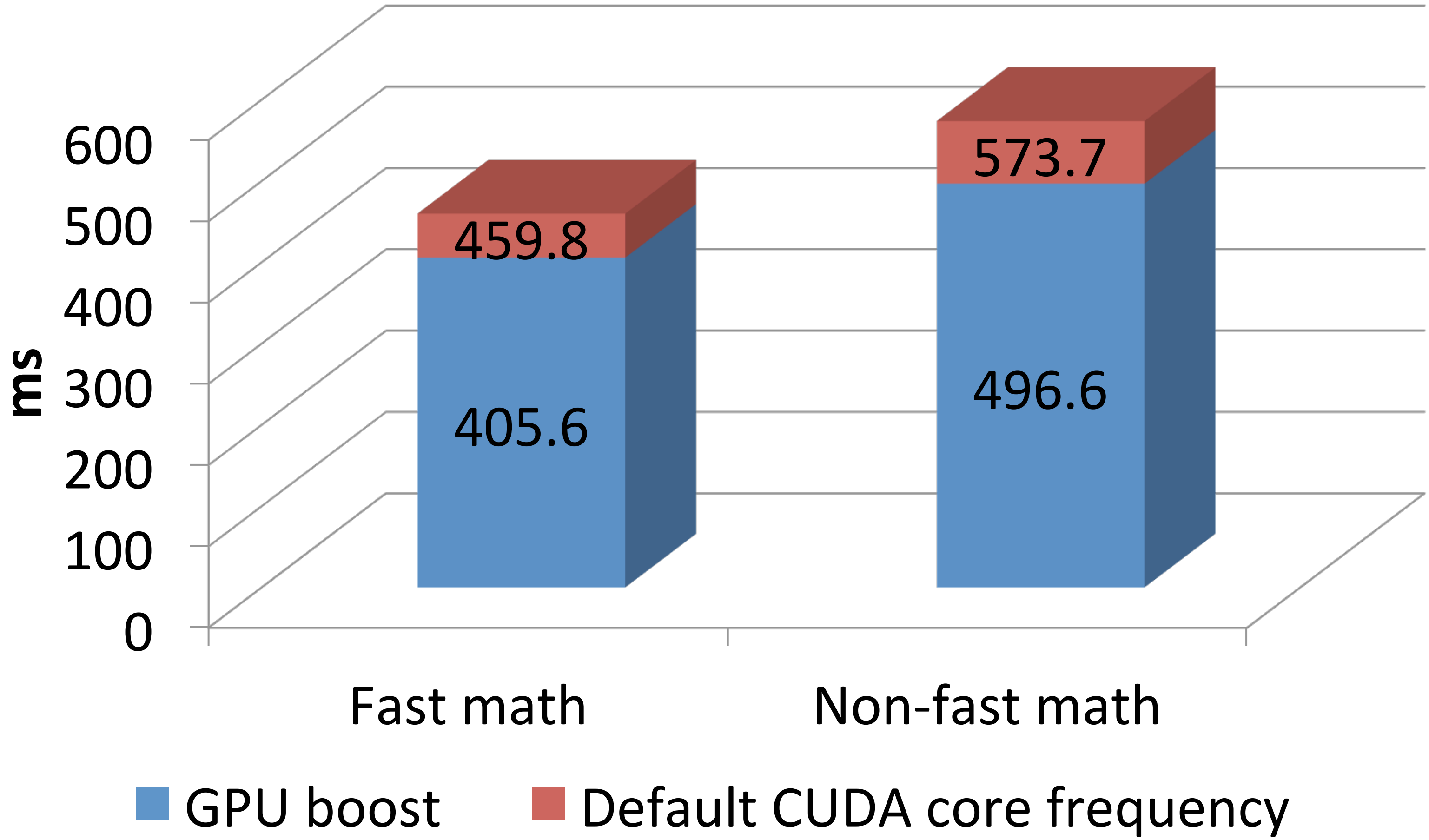
The optimal thread block size is 96

RRTMG_SW processing time for a six GPU optimizations steps

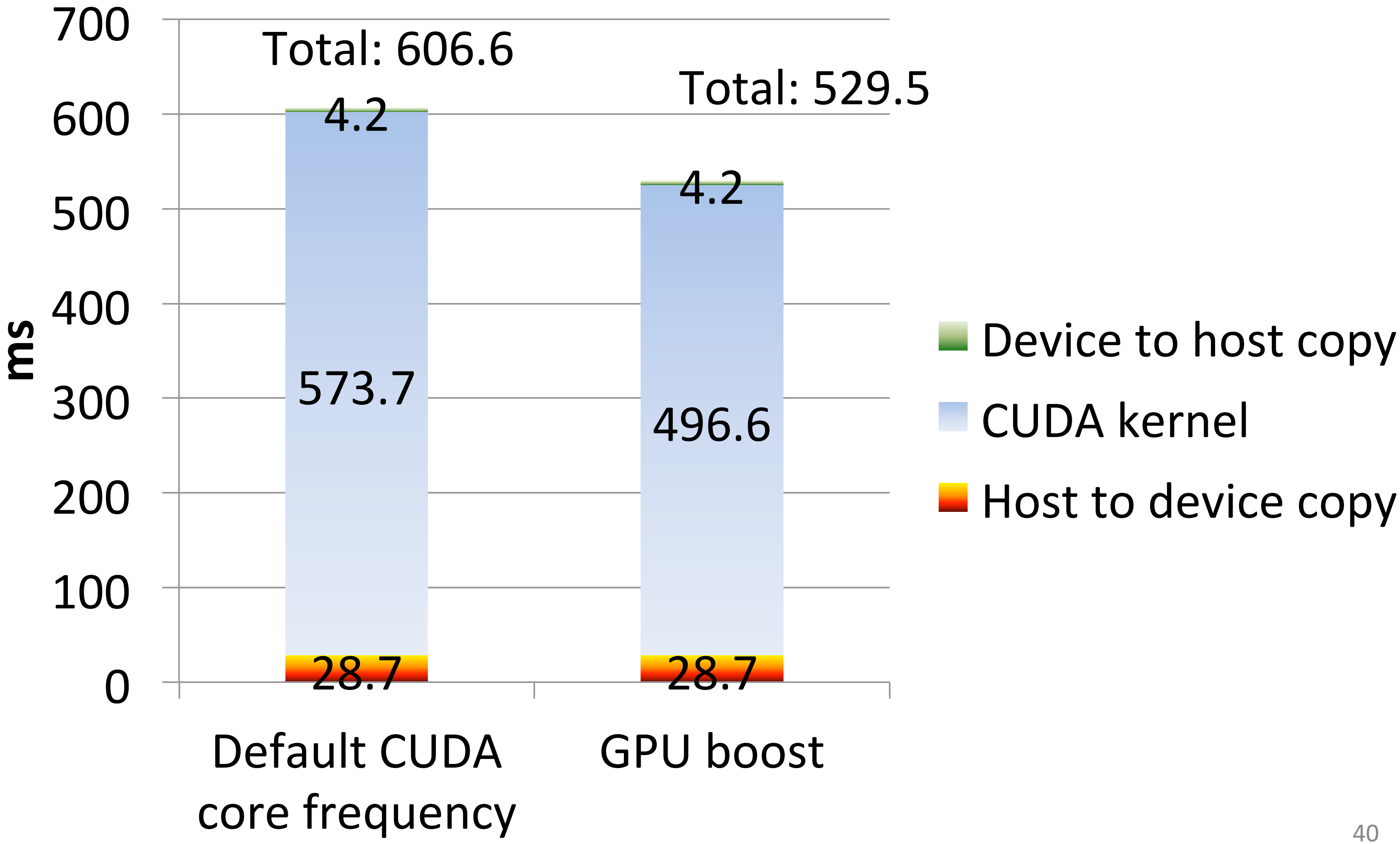


12 frequently used three dimensional arrays of size idim x kdim x jdim out of 69 temporary arrays were eliminated

Fast math and GPU boost features



Execution time of RRTMG_SW including data transfer



Comparing RRTMG_SW GPU implementations

AER: 18,819 profile calculations with **72** layers
0.84 s on **K20**
-> 22,404 profiles / second.

SSEC: 130,900 profile calculations with **35** layers
0.50s on **K40**
-> 261,800 profiles / second.

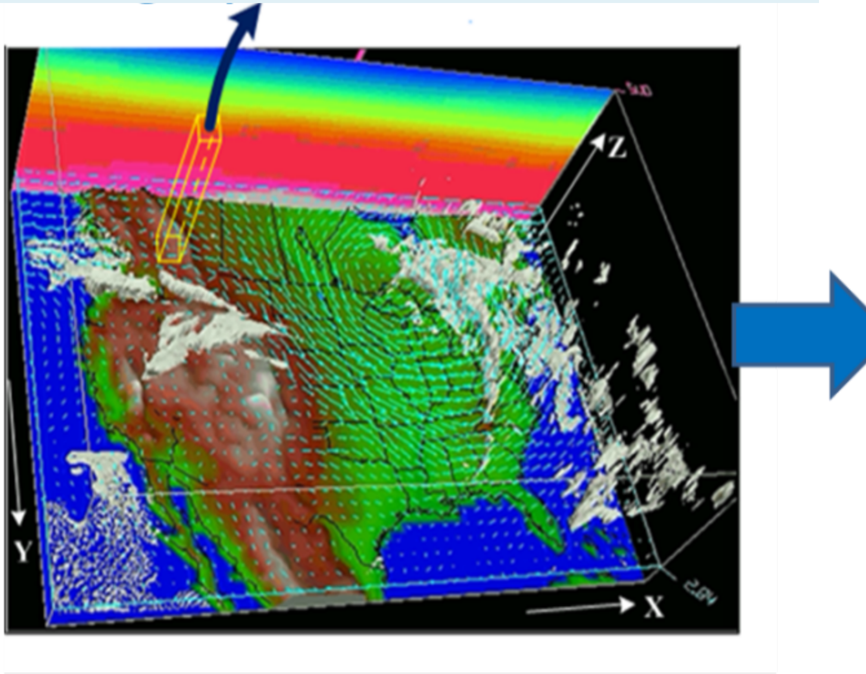
- K40 is ~2x faster than K20
- The number of layers has a non-linear impact on processing speed, which makes comparison of profiles/s metrics diffucult

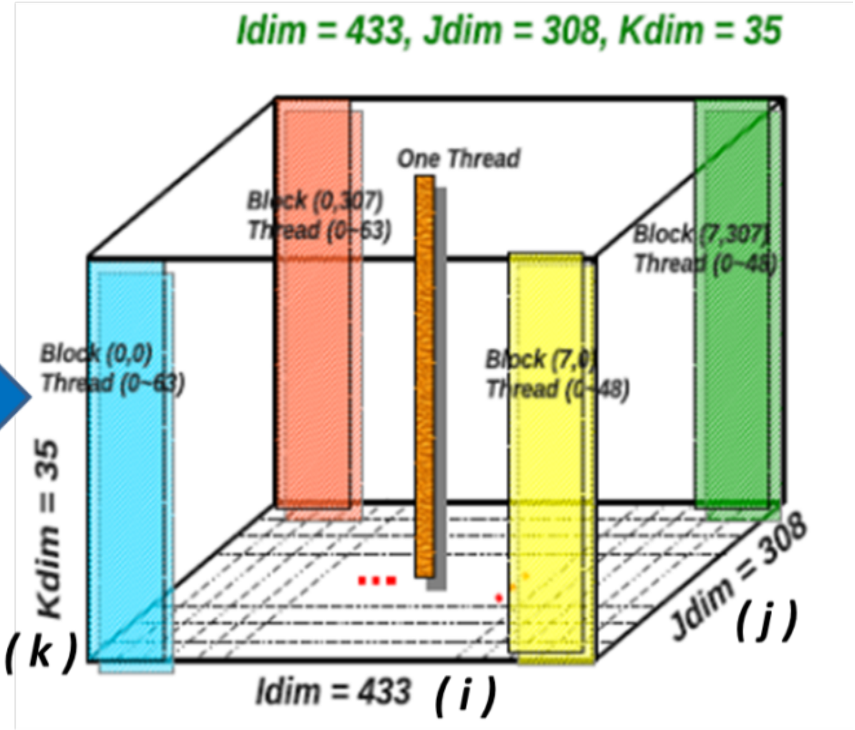
Implementation of YSU PBL in GPUs with CUDA Program - 2

To **test** whether the coding of **YSU PBL** in CUDA is correct, CONtinential United States (**CONUS**) benchmark data set is used –

433 x 308 horizontal grid points with **35** vertical levels

Equations describing YSU PBL scheme are executed in one thread for each grid point





Mapping of the CONUS domain onto one GPU thread-block-grid domain

Improvement of YSU PBL in GPU-Based Parallelism - 1

GPU runtime and speedup as compared to one-single-threaded CPU code for the **first CUDA version** YSU PBL module

	CPU runtime	GPU runtime	Speedup
One CPU core	1800.0 ms		
Non-coalesced		50.0 ms	36.0x
Coalesced		48.0 ms	37.5x

Improvement of YSU PBL in GPU-Based Parallelism – 2.1 with More L1 Cache than Shared Memory

Three configurations of memory between **shared memory** and **L1 cache** are :

(1) **48** KB **shared memory**, **16** KB **L1 cache** – **default**

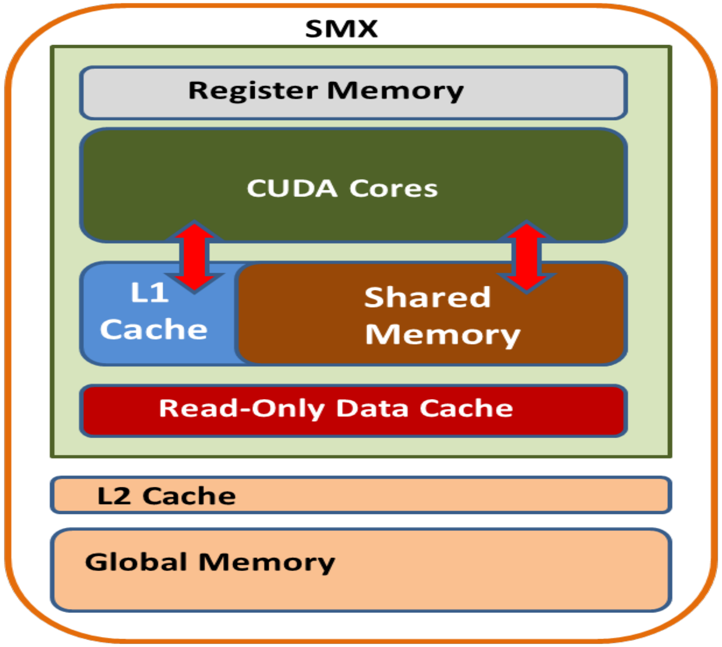
(2) **32** KB **shared memory**, **32** KB **L1 cache**

(3) **16** KB **shared memory, 48** KB **L1 cache**

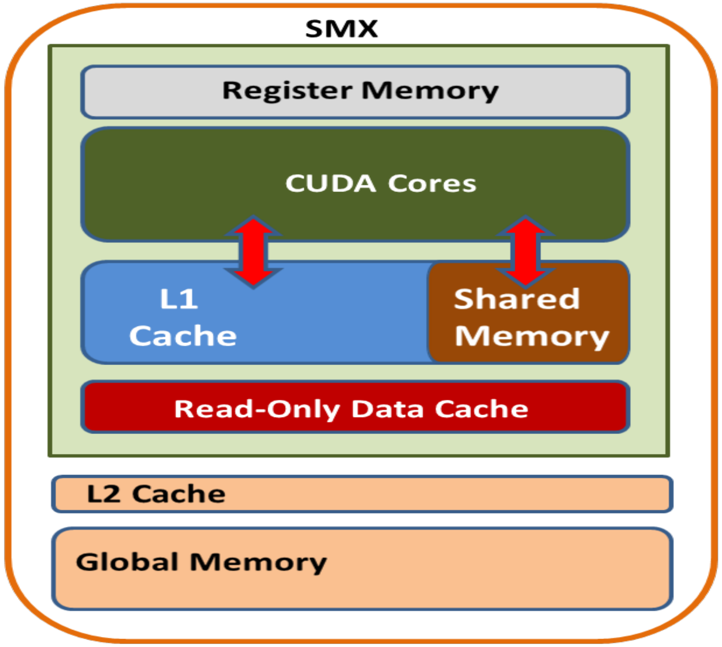
→ can be achieved by applying **"cudaFuncCachePreferL1"**

→ After increasing **L1 cache** with **"cudaFuncCachePreferL1" ,** the GPU runtime reduces and speedup increases

Improvement of YSU PBL in GPU-Based Parallelism – 2.2 with More L1 Cache than Shared Memory



Command ***cudaFuncCachePreferL1***
Is Not launched



Command ***cudaFuncCachePreferL1***
Is launched

	CPU runtime	GPU runtime	Speedup
One CPU core	1800.0 ms		
Non-coalesced		48.0 ms	37.5x
Coalesced		45.0 ms	40.0x

Improvement of YSU PBL in GPU-Based Parallelism – 3.2 with Scalarizing Temporary Arrays

By Scalarization, the **temporary arrays** are **reduced** from **68** down to **14** arrays –

→**This makes global memory access reduced a lot !!!**

	CPU runtime	GPU runtime	Speedup
One CPU core	1800.0 ms		
Non-coalesced		39.0 ms	46.2x
Coalesced		35.0 ms	51.4x

Improvement of YSU PBL in GPU-Based Parallelism – 4.1 with Releasing Vertical-Level Dependence

- The **leftover 14 local array variables** are tricky because the vertical-level (***k***) components are not independent with one another.
- The complication can be seen from the **differential equations** for those **prognostic variavles** $(C,\ u,\ v,\ \theta, q_v,\ q_c,\ q_i)$

→ **Firstly,** we have to **release the first z dependence** –

the ***k-th*** component depends on the ***(k-1)-th*** input

→ **Secondly, release the second z dependence** –

the ***k-th*** component needs inputs from ***(k+1)-th*** component

$$\frac{\partial C}{\partial t} = \boxed{\frac{\partial}{\partial z}\left[k_c\left(\frac{\partial C}{\partial z}-\gamma_c\right) - (\overline{w'c'})_h\left(\frac{z}{h}\right)^3\right]}$$

⬭ **First z-dependent component**

▢ **Second z-dependent component**

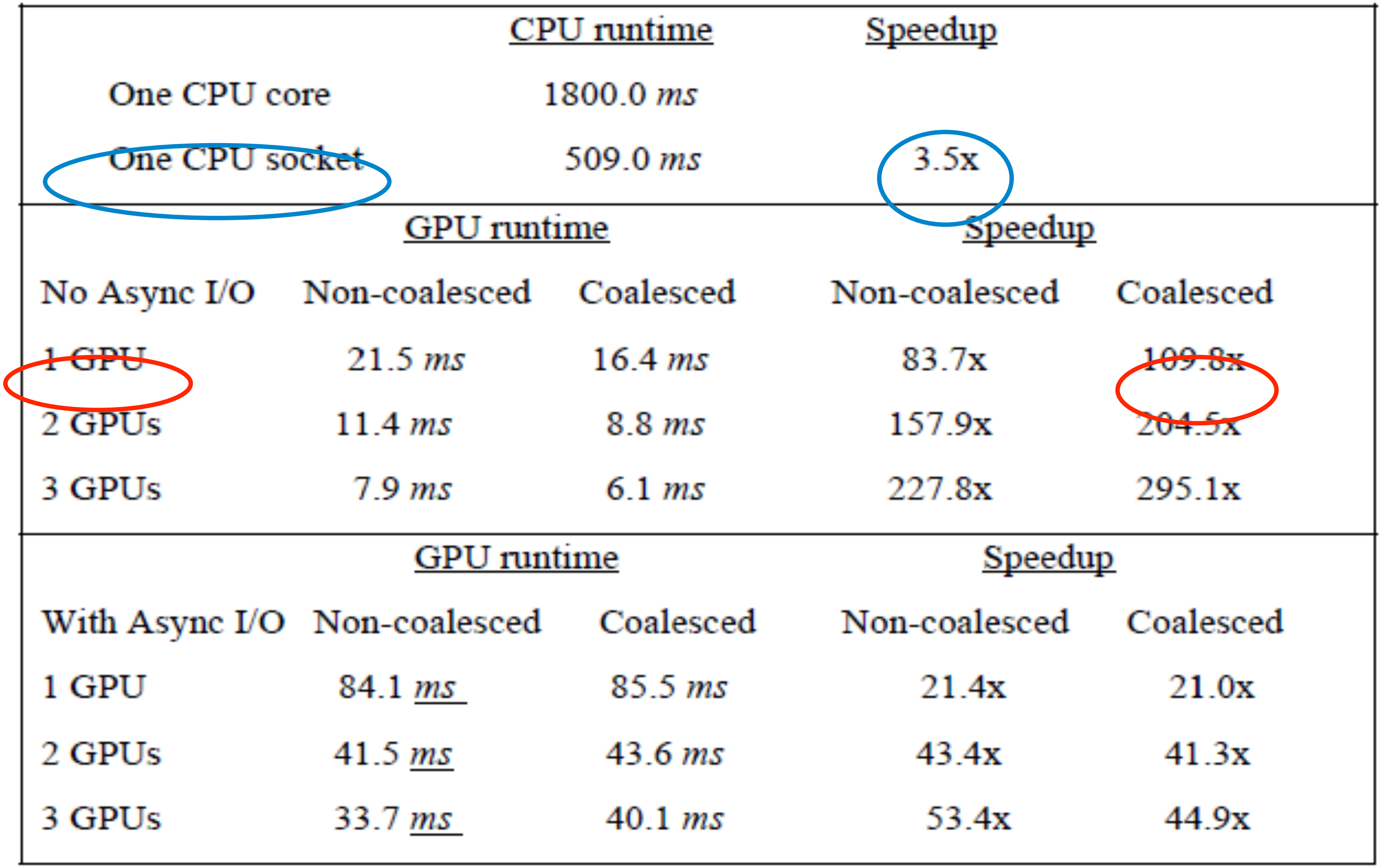
Improvement of YSU PBL in GPU-Based Parallelism – 4.3 with Releasing Vertical-Level Dependence

- About **1/3** of the Fortran codes appears to be such a dependence among the ***(k-1)-th***, ***k-th,*** and ***(k+1)-th*** components
- **This reduces the global memory access even much more !!!**

	CPU runtime	GPU runtime	Speedup
One CPU core	1800.0 ms		
Non-coalesced		21.53 ms	83.4x
Coalesced		16.45 ms	109.4x

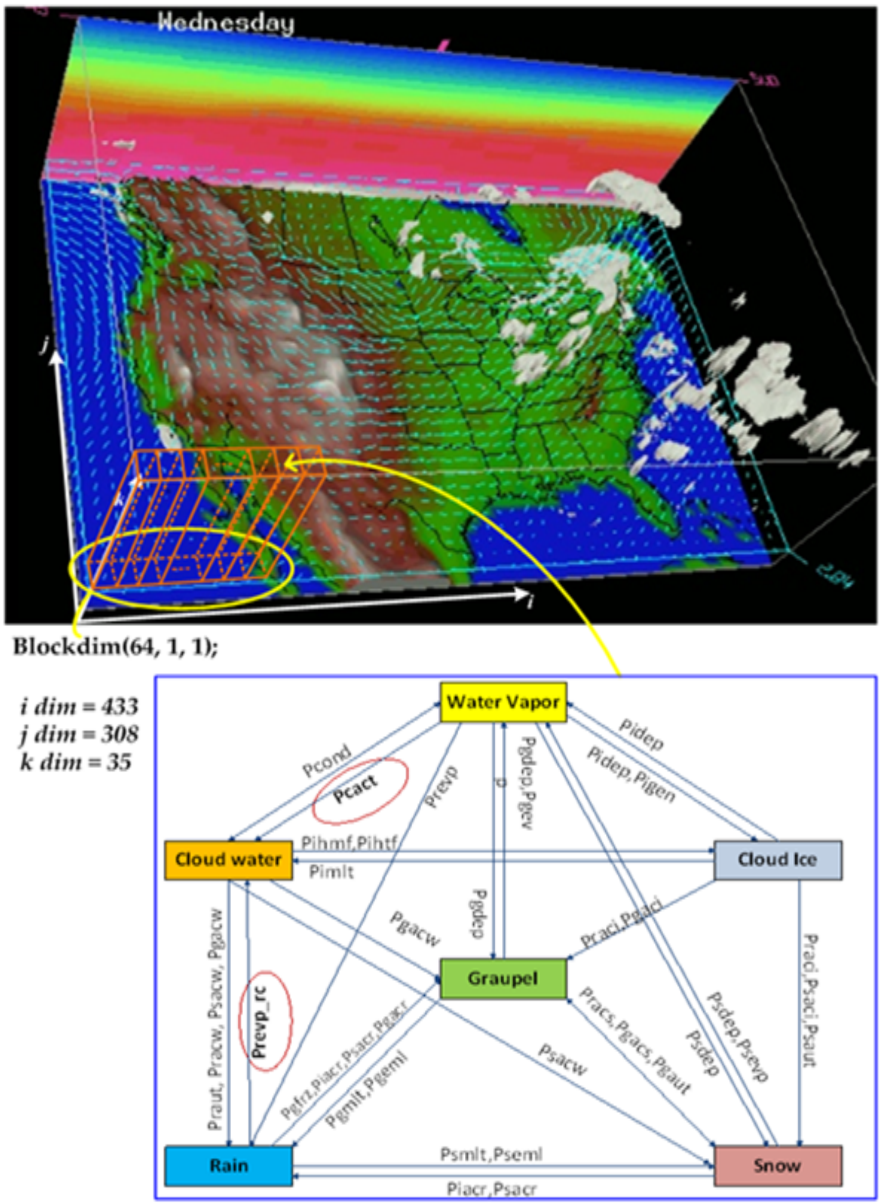
Note: So far, **63 registers/thread** and **64 threads/block** are used, and **No I/O** is involved

GPU Runtime & Speedups with Multi-GPU Implementations for YSU PBL Module



	CPU runtime		Speedup	
One CPU core	1800.0 *ms*			
One CPU socket	509.0 *ms*		3.5x	
	GPU runtime		Speedup	
No Async I/O	Non-coalesced	Coalesced	Non-coalesced	Coalesced
1 GPU	21.5 *ms*	16.4 *ms*	83.7x	109.8x
2 GPUs	11.4 *ms*	8.8 *ms*	157.9x	204.5x
3 GPUs	7.9 *ms*	6.1 *ms*	227.8x	295.1x
	GPU runtime		Speedup	
With Async I/O	Non-coalesced	Coalesced	Non-coalesced	Coalesced
1 GPU	84.1 *ms*	85.5 *ms*	21.4x	21.0x
2 GPUs	41.5 *ms*	43.6 *ms*	43.4x	41.3x
3 GPUs	33.7 *ms*	40.1 *ms*	53.4x	44.9x

CUDA-based GPU accelerated WRF modules



WRF Module name	Speedup vs. one thread on 1.8Ghz Sandy Bridge (gfortran v.4.6.2)
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

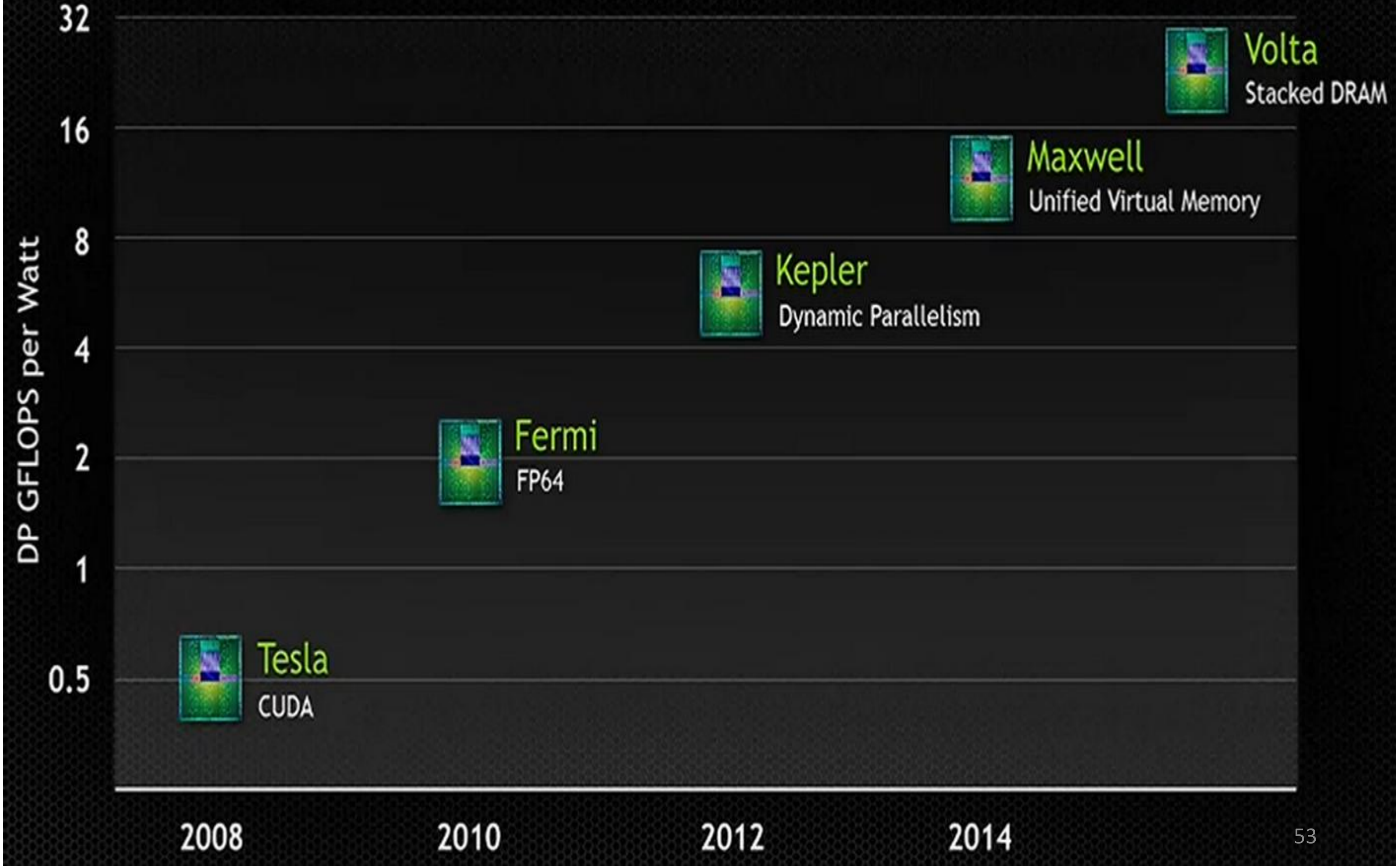
Radiation	**RRTMG LW**	**123x / 127x**	**JSTARS, 7, 3660-3667, 2014**
	RRTMG SW	**202x / 207x**	**Submitted to J. Atmos. Ocean. Tech.**
	Goddard SW	**92x / 134x**	**JSTARS, 5, 555-562, 2012**
	Dudhia SW	**19x / 409x**	
Surface	**MYNN SL**	**6x / 113x**	
	TEMF SL	**5x / 214x**	
	Thermal Diffusion LS	**10x / 311x [2.1 x]**	**(GPU) Submitted to JSATRS**
PBL	**YSU PBL**	**34x / 193x [2.4x]**	**(GPU) Submitted to GMD**
	TEMF PBL	**[14.8x]**	**(MIC) SPIE:doi:10.1117/12.2055040**
CU P	**Betts-Miller-Janjic (BMJ) convetion**	**55x / 105x**	

GPU speedup: speedup with IO / speedup without IO
MIC improvement factor in [ ]: w.r.t. 1st version multi-threading code before any improvement

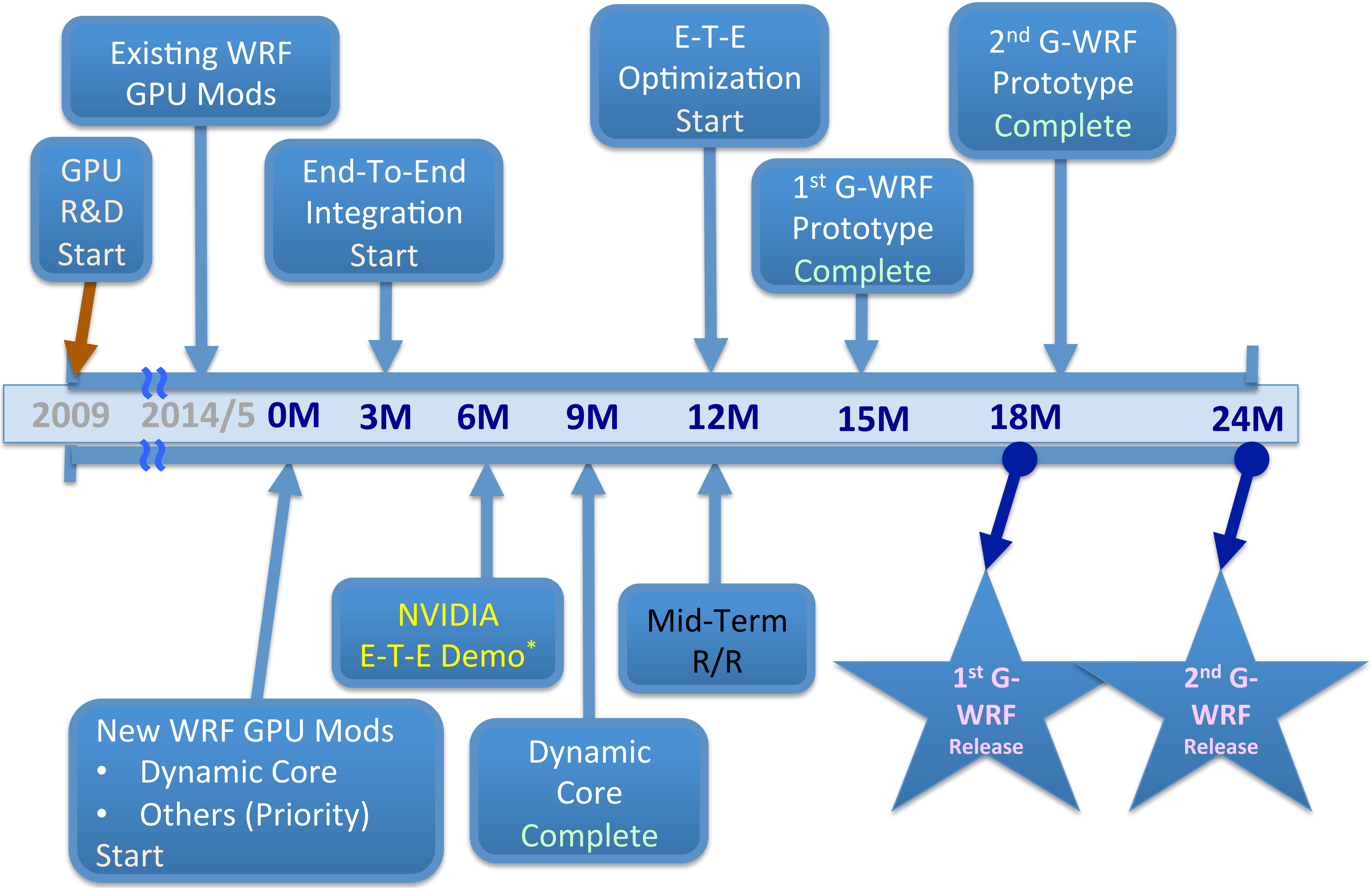
Cloud Microphysics

Kessler MP	70x / 816x	J. Comp. & GeoSci., 52, 292-299, 2012
Purdue-Lin MP	156x / 692x [4.2x]	(GPU) SPIE: doi:10.1117/12.901825
WSM 3-class MP	150x / 331x	
WSM 5-class MP	202x / 350x	JSTARS, 5, 1256-1265, 2012
Eta MP	37x / 272x	SPIE: doi:10.1117/12.976908
WSM 6-class MP	165x / 216x	Submitted to J. Comp. & GeoSci.
Goddard GCE MP	348x / 361x [4.7x]	(GPU) Accepted for publication in JSTARS
Thompson MP	76x / 153x [2.3x]	(MIC) SPIE: doi:10.1117/12.2055038
SBU 5-class MP	213x / 896x	JSTARS, 5, 625-633, 2012
WDM 5-class MP	147x / 206x	
WDM 6-class MP	150x / 206x	J. Atmo. Ocean. Tech., 30, 2896, 2013

GPU Roadmap


DP GFLOPS per Watt
32
16
8
4
2
1
0.5
Volta
Stacked DRAM
Maxwell
Unified Virtual Memory
Kepler
Dynamic Parallelism
Fermi
FP64
Tesla
CUDA
2008
2010
2012
2014

CUDA-GPU WRF Project Milestone (October, 2014)



2009 2014/5 0M 3M 6M 9M 12M 15M 18M 24M

GPU R&D Start

Existing WRF GPU Mods

End-To-End Integration Start

E-T-E Optimization Start

1st G-WRF Prototype Complete

2nd G-WRF Prototype Complete

New WRF GPU Mods
- Dynamic Core
- Others (Priority)

Start

NVIDIA E-T-E Demo*

Dynamic Core Complete

Mid-Term R/R

1st G-WRF Release

2nd G-WRF Release

Accelerator-based (GPU/MIC) WRF model development at SSEC/CIMSS UW-Madison

- **Intel** awarded SSEC a two-year grant to develop Intel MIC Xeon Phi Coprocessor based WRF using OpenMP/OpenACC common architecture
  - UW-Madison becomes one of the Intel Parallel Computing Center (IPCC)
  - **MIC WRF** Open Source with minimum changes to existing cores

- **NVIDIA**, the world largest GPU chip maker, has selected SSEC as one of their CUDA Research Center (CRC) and will fund SSEC to develop a GPU-CPU **Hybrid WRF** prototype using CUDA architecture

- **Tempo Quest CUDA GPU-WRF** project (proposal pending)
  - World fastest GPU-WRF with CUDA based unique architecture
  - Need time consuming code porting & optimization

SSEC Accelerator Technology Team, Collaborators & Sponsors

- SSEC, UW-Madison:
  - Bormin Huang, PhD
  - Jarno Mielikainen, PhD
  - Melin Huang, PhD
  - Allen Huang, PhD
  - Visiting Scholars from around the world
- NOAA:
  - Mitchell Goldberg and Ajay Mehta
- NASA: Tsengdar Lee
- NVIDIA: Stan Posey
- INTEL: Michael Greenfield
- Tempo Quest, Inc: Ed & Gene (proposal pending)

Thank you for your Attention!