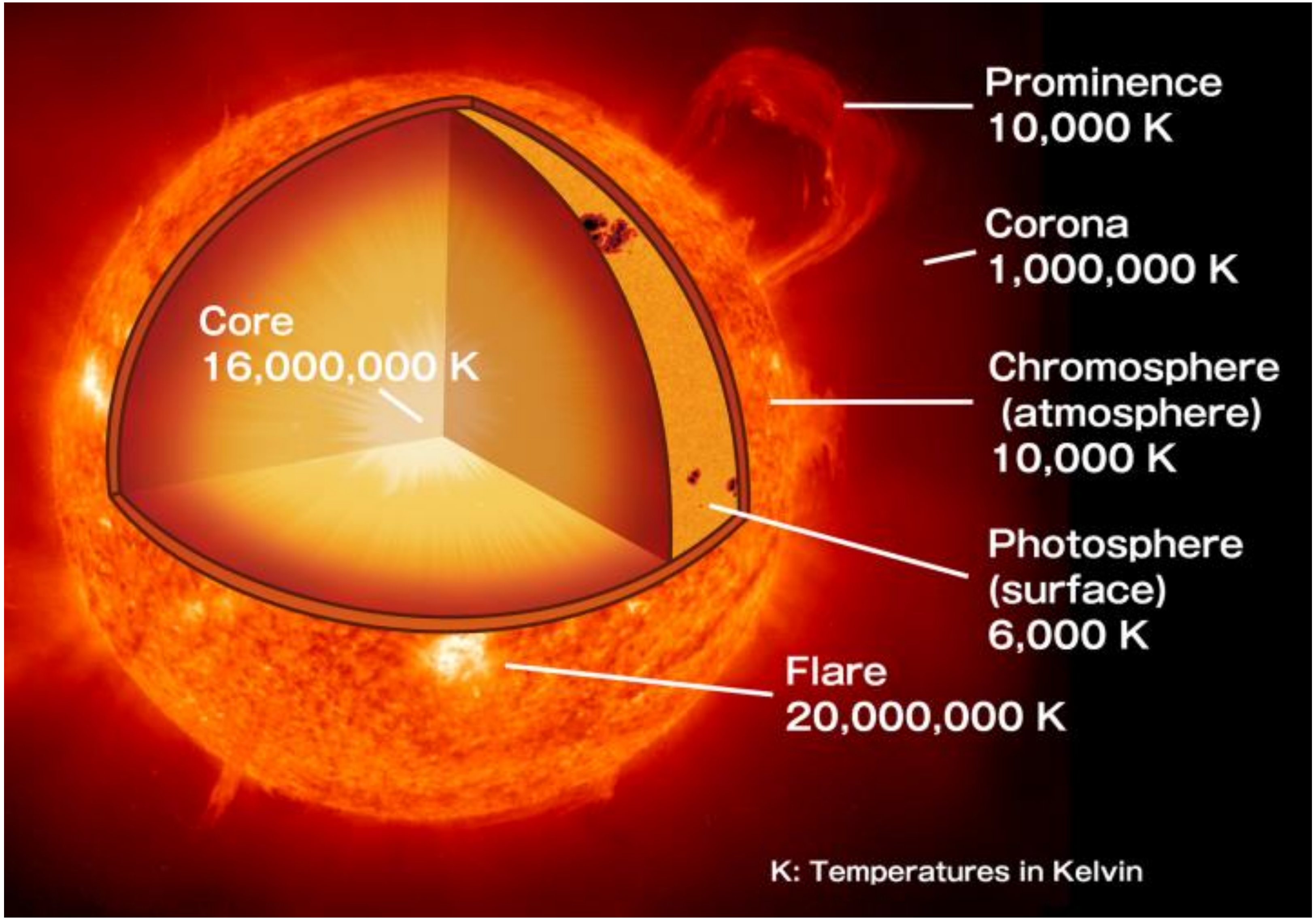
Porting MURaM (Max Planck University of Chicago Radiative MHD) to GPUs Using OpenACC

Rich Loft (Director TDD in CISL, NCAR)
Eric Wright (PhD student) & Sunita Chandrasekaran (Assistant Professor)
University of Delaware
loft@ucar.edu, {efwright, schandra}@udel.edu

Project in collaboration with NCAR, Max Planck for Solar System Research and University of Delaware

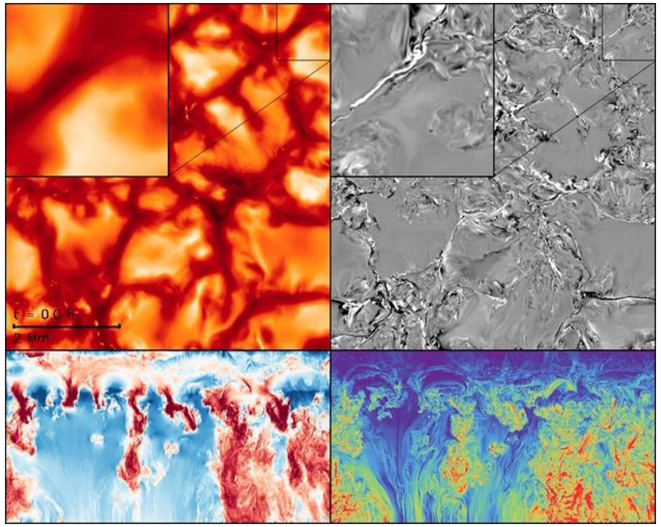
March 19, 2019 GTC 2019

Anatomy of the Sun



MURaM (Max Planck University of Chicago Radiative MHD)

- The primary solar model used for simulations of the upper convection zone, photosphere and corona.
- Jointly developed and used by HAO, the Max Planck Institute for Solar System Research (MPS) and the Lockheed Martin Solar and Astrophysics Laboratory (LMSAL).
- MURaM has contributed substantially to our understanding of solar phenomena.
- MURaM also plays a key role in interpreting high resolution solar observations.

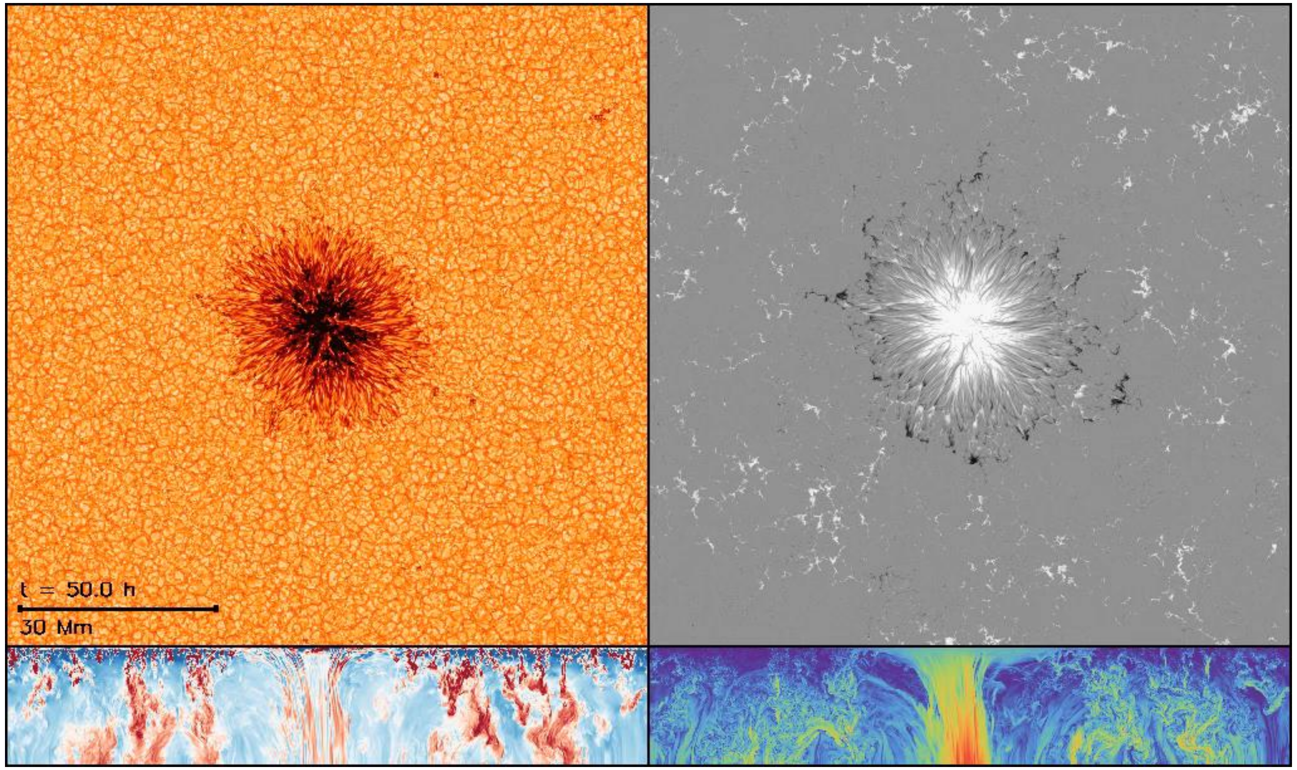


MURaM simulation of solar granulation

The Daniel K. Inouye Solar Telescope (DKIST), a ~$300M NSF investment, is expected to advance the resolution of ground based observational solar physics by an order of magnitude.

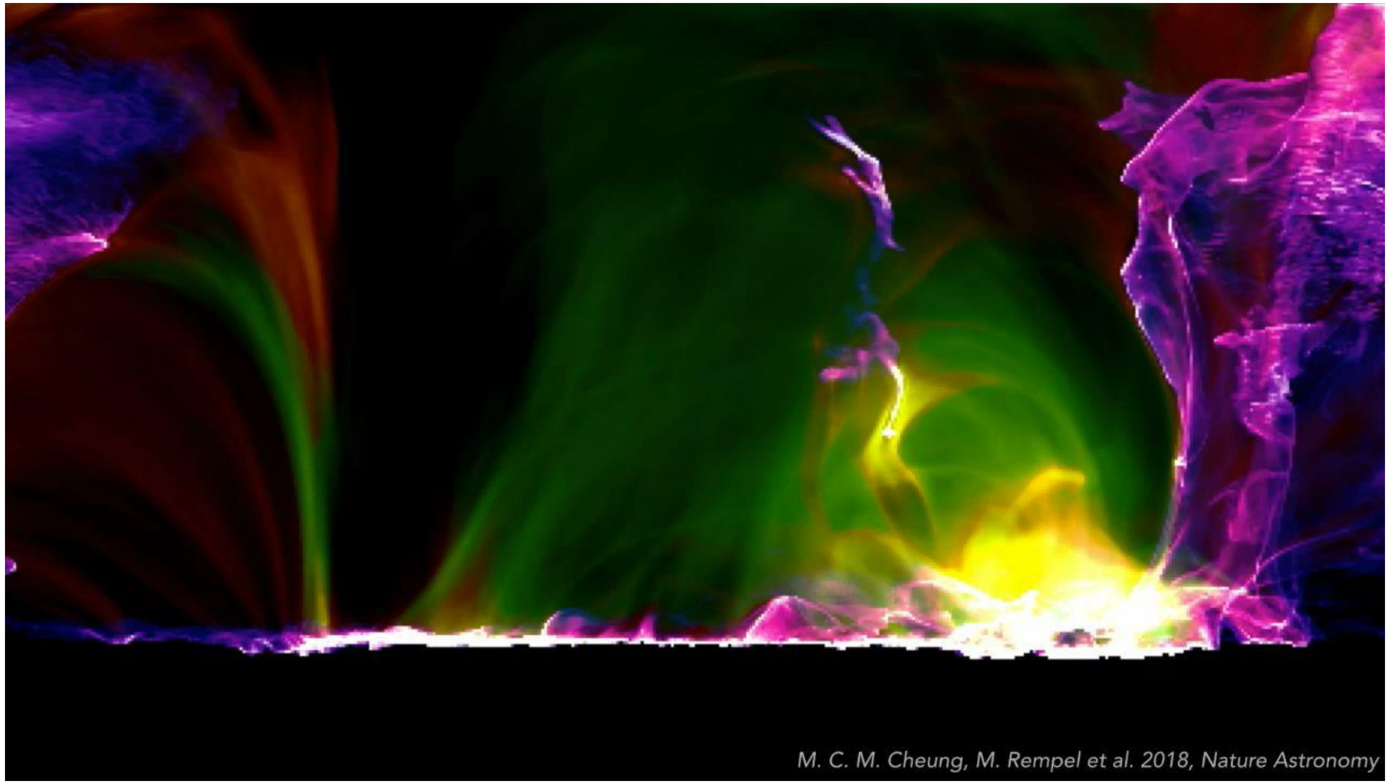
High Resolution Simulations of the Solar Photosphere

- Forward modeling of DKIST observables will require simulations with grid spacing of 4 km on a regular basis.
- Requires at least 10-100x increase in computing power compared to current baseline.



From data inspired to data driven simulations of solar eruptions

- Realistic simulations of the coupled solar atmosphere are an important tool to understand and even predict solar eruptions.
- Current models run about ~100x slower than real-time
- Data driven simulations of solar events would allow for analysis and prediction of ongoing solar events
- Future data assimilation applications will require ensemble runs (~10x)



Comprehensive model of entire life cycle of a solar flare (Cheung et al 2018)

Realistic Simulations of the Chromosphere

- The chromosphere is a difficult region to observe and model
- New instruments (DKIST 4m telescope and sunrise balloon observatory) will allow for unprecedented observations
- However, the modeling still must be brought up-to-date to these observational advancements.

Toward Predicting Solar Eruptions

- **How do we get MURaM 100x faster? GPU-accelerated RT is the key.**
  - RT solver speed up on GPUs **(~85x)** a CPU core via wavefront algorithms (Chandrasekaran et al.)
  - RT iteration counts are reduced **(~2x)** on GPU's bigger subdomains.
  - But, we can rewrite RT solver to do required wavelengths (these are embarrassingly parallel).
  - Last two points play to GPU's strength: data parallelism.
  - **Estimate 450 GPUs could achieve breakeven (1 simulated hour/hour)**
- **Thousands of GPUs would be required to do full data assimilation.**
  - Expensive but not unthinkable (Summit has 27,600!)
- **Requirements play to strengths of GPUs, and trend in their design.**

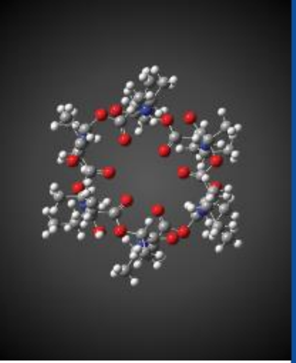
Planned MURaM Development

- Porting of the MURaM code to GPUs using OpenACC (collaboration between HAO, CISL, University of Delaware and MPS)
- Implementation of a more sophisticated chromosphere (HAO, MPS)
- Implementation of boundary conditions that allow data driven simulations of solar events (HAO, CU Boulder, LMSAL)
- New IO modules that allow for data compression during the IO process and runtime visualization.

Time to dive deeper into the computational science side of things! ☺

Roadmap

- Profiling, parallelization, re-profiling
- Optimizing CPU-GPU data transfer/management
- Focusing on the most important loop – Radiation Transport (RT)
  - Long term science goals
- Re-designing Radiation Transport Algorithm
- More profiling info to find & address performance challenges



GAUSSIAN 16

Mike Frisch, Ph.D.
President and CEO
Gaussian, Inc.

"Using OpenACC allowed us to continue development of our fundamental algorithms and software capabilities simultaneously with the GPU-related work. In the end, we could use the same code base for SMP, cluster/ network and GPU parallelism. PGI's compilers were essential to the success of our efforts."

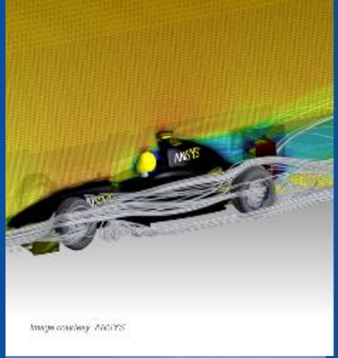
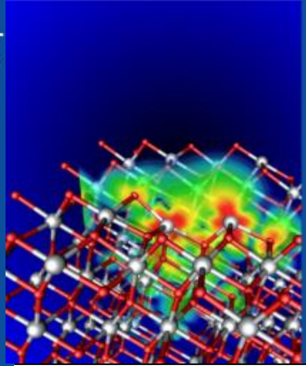


Image courtesy: ANSYS

ANSYS FLUENT

Sunil Sathe
Lead Software Developer
ANSYS Fluent

"We've effectively used OpenACC for heterogeneous computing in ANSYS Fluent with impressive performance. We're now applying this work to more of our models and new platforms."



VASP

Prof. Georg Kresse
Computational Materials Physics
University of Vienna

"For VASP, OpenACC is *the* way forward for GPU acceleration. Performance is similar and in some cases better than CUDA C, and OpenACC dramatically decreases GPU development and maintenance efforts. We're excited to collaborate with NVIDIA and PGI as an early adopter of CUDA Unified Memory."

COSMO

Dr. Oliver Fuhrer
Senior Scientist
Meteoswiss

"OpenACC made it practical to develop for GPU-based hardware while retaining a single source for almost all the COSMO physics code."

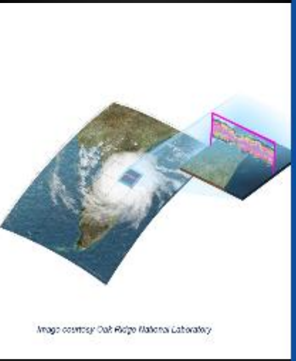


Image courtesy: Oak Ridge National Laboratory

E3SM

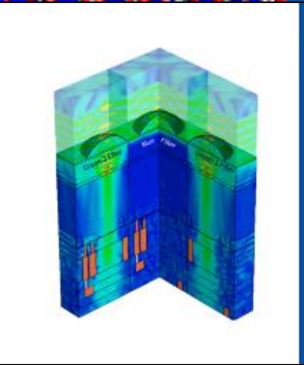
Mark A. Taylor
Multiphysics Applications
Sandia

"The CAAR project provided us with early access to Summit hardware and access to PGI compiler experts. Both of these were critical to our success. PGI's OpenACC support remains the best available and is competitive with much more intrusive programming model approaches."

NUMECA FINE/Open

David Gutzwiller
Lead Software Developer
NUMECA

"Porting our unstructured C++ CFD solver FINE/Open to GPUs using OpenACC would have been impossible two or three years ago, but OpenACC has developed enough that we're now getting some really good results."



SYNOPSYS

Dr. Lutz Schneider
Senior R&D Engineer
Synopsys Inc.

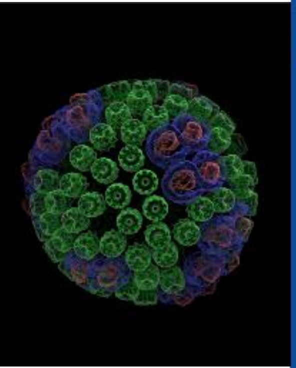
"Using OpenACC, we've GPU-accelerated the Synopsys TCAD Sentaurus Device EMW simulator to speed up optical simulations of image sensors. GPUs are key to improving simulation throughput in the design of advanced image sensors."


Image courtesy: NCAR

MPAS-A

Richard Loft
Director, Technology Development
NCAR

"Our team has been evaluating OpenACC as a pathway to performance portability for the Model for Prediction (MPAS) atmospheric model. Using this approach on the MPAS dynamical core, we have achieved performance on a single P100 GPU equivalent to 2.7 dual socketed Intel Xeon nodes on our new Cheyenne supercomputer."



VMD

John Stone
Senior Research Programmer
Beckham Institute
University of Illinois

"Due to Amdahl's law, we need to port more parts of our code to the GPU if we're going to speed it up. But the sheer number of routines poses a challenge. OpenACC directives give us a low-cost approach to getting at least some speed-up out of these second-tier routines. In many cases it's completely sufficient because with the current algorithms, GPU performance is bandwidth-bound."

GTC

Zhihong Lin
Professor and Principal Investigator
UC Irvine

"Using OpenACC our scientists were able to achieve the acceleration needed for integrated fusion simulation with a minimum investment of time and effort in learning to program GPUs."

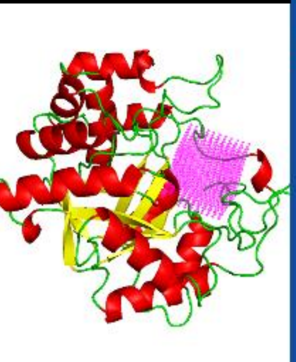
OpenACC
More Science, Less Programming


Map courtesy University of Tokyo

GAMERA

Takuma Yamaguchi, Kohei Fujita, Tsuyoshi Ichimura, Muneo Hori, Lalith Wijerathne,
The University of Tokyo

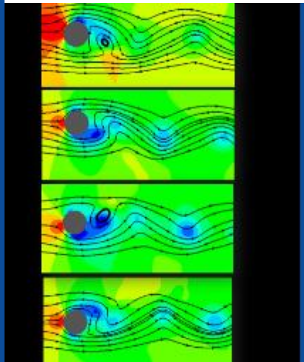
"With OpenACC and a compute node based on NVIDIA's Tesla P100 GPU, we achieved more than a 14X speed up over a K Computer node running our earthquake disaster simulation code"



SANJEEVINI

Abhilash Jayaraj
Project Scientist
Indian Institute of Technology
New Delhi

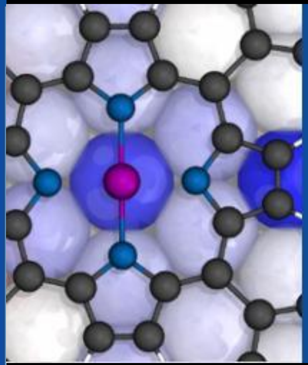
"In an academic environment maintenance and speedup of existing codes is a tedious task. OpenACC provides a great platform for computational scientists to accomplish both tasks without involving a lot of efforts or manpower in speeding up the entire computational task."



IBM-CFD

Somnath Roy
Assistant Professor
Mechanical Engineering Department
Indian Institute of Technology Kharagpur

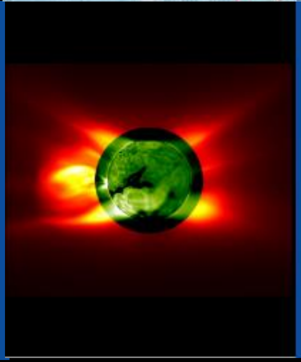
"OpenACC can prove to be a handy tool for computational engineers and researchers to obtain fast solution of non-linear dynamics problem. In immersed boundary incompressible CFD, we have obtained order of magnitude reduction in computing time by porting several components of our legacy codes to GPU. Especially the routines involving search algorithm and matrix solvers have been well-accelerated to improve the overall scalability of the code."



PWscf (Quantum ESPRESSO)

Filippo Spiga
Senior Contributor
Quantum ESPRESSO group

"CUDA Fortran gives us the full performance potential of the CUDA programming model and NVIDIA GPUs. While leveraging the potential of explicit data movement, !$CUF KERNELS directives give us productivity and source code maintainability. It's the best of both worlds."



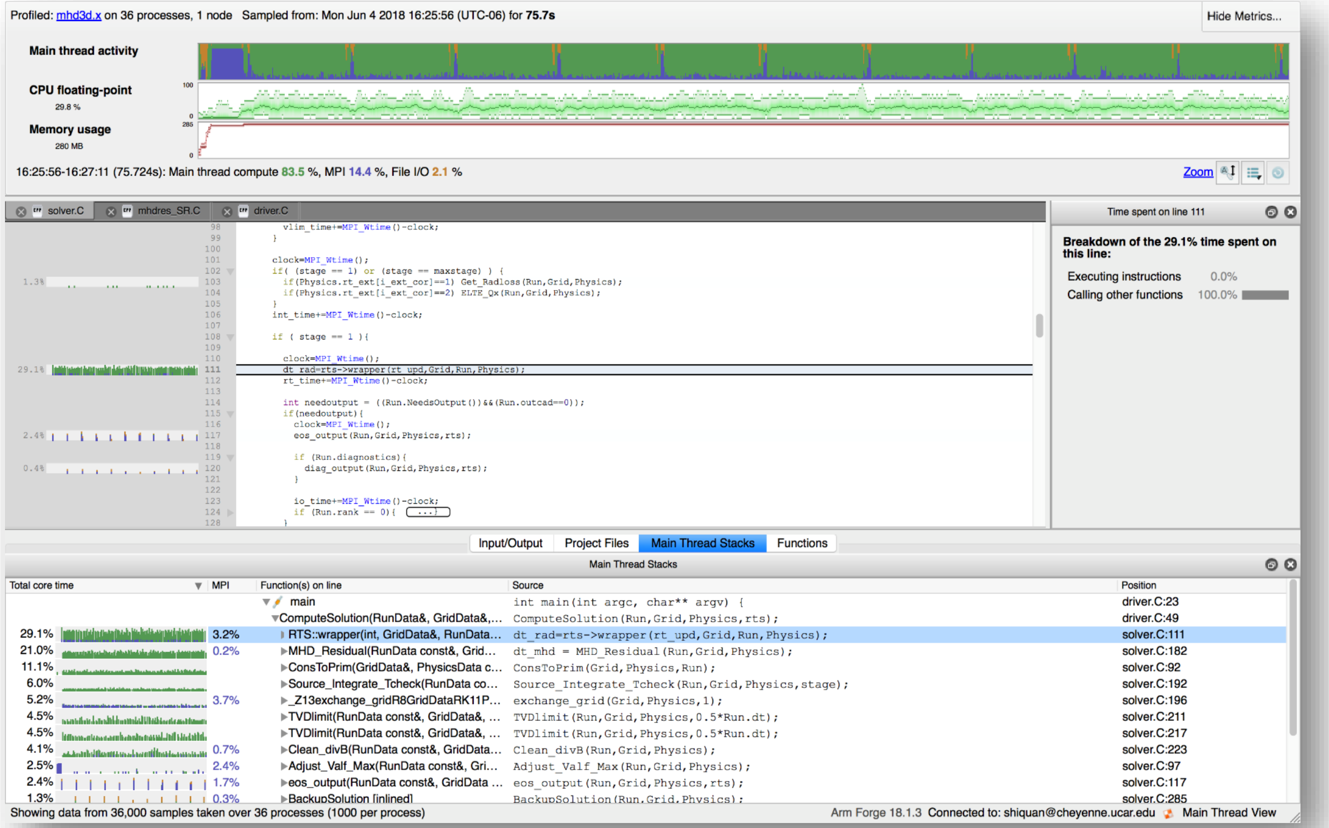
MAS

Ronald M. Caplan
Computational Scientist
Predictive Science Inc.

"Adding OpenACC into MAS has given us the ability to migrate medium-sized simulations from a multi-node CPU cluster to a single multi-GPU server. The implementation yielded a portable single-source code for both CPU and GPU runs. Future work will add OpenACC to the remaining model features, enabling GPU-accelerated realistic solar storm modeling."

Profiling Tools

- Several profiling techniques were used to obtain an initial, high-level view of the code
- Function call map
- Arm MAP for generalized performance metrics and MPI
- NVProf for GPU performance profiling



Experimental Setup

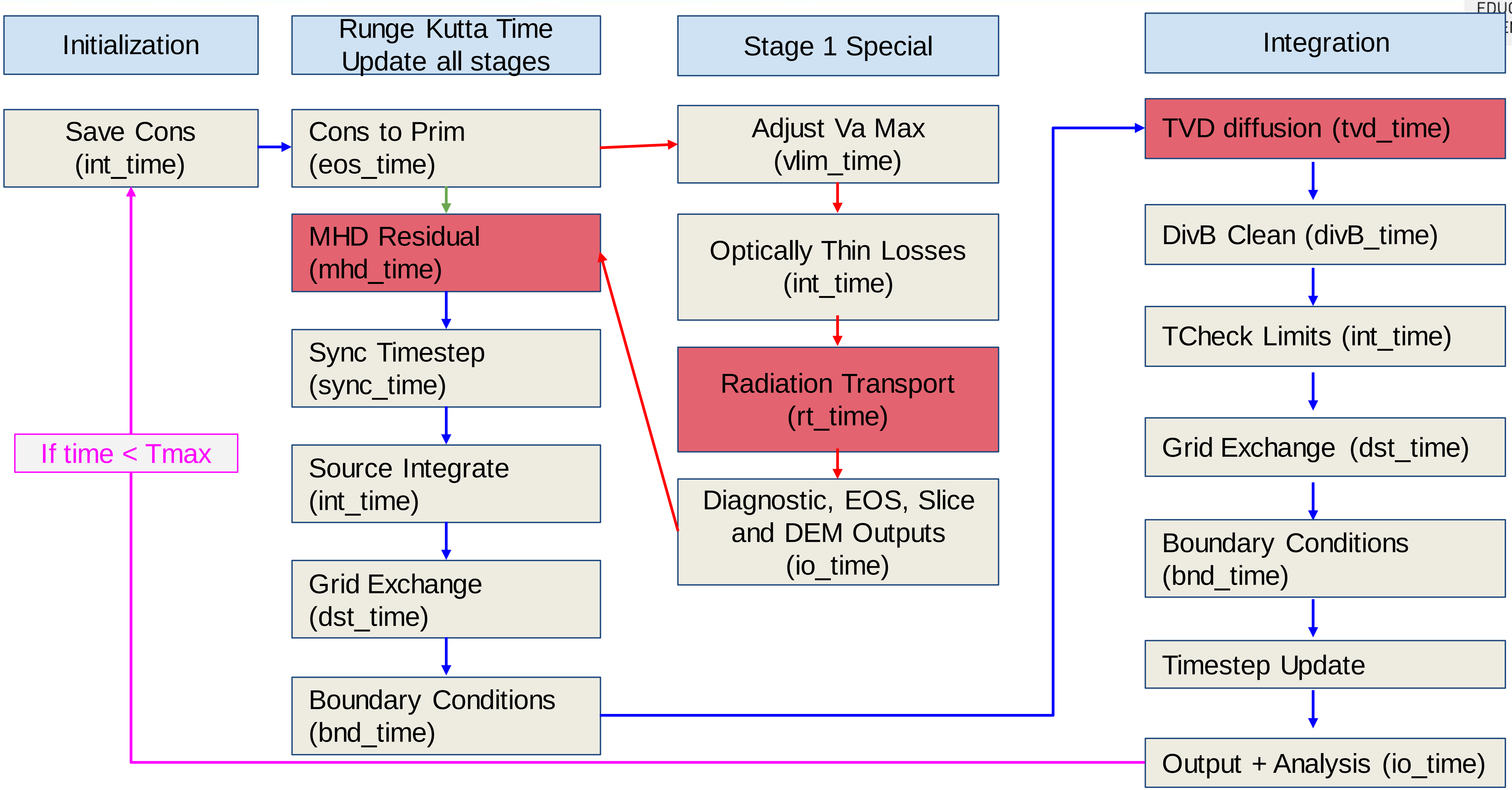
- NVIDIA PSG Cluster (hardware)
  - CPU: Intel Xeon E5-2698 v3 (16-core) and Xeon E5-2690 v2 (10-core)
  - GPU: NVIDIA Tesla V100 (4 GPUs per node)
- Software Used
  - PGI 18.4 (CUDA 9.1 and 9.2) and PGI 18.10
    - Results in this talk use PGI 18.4
  - icc 17.0.1/18.0.1

Thanks to NVIDIA for giving us access to their PSG system for our experiments!!!

Routine Descriptions

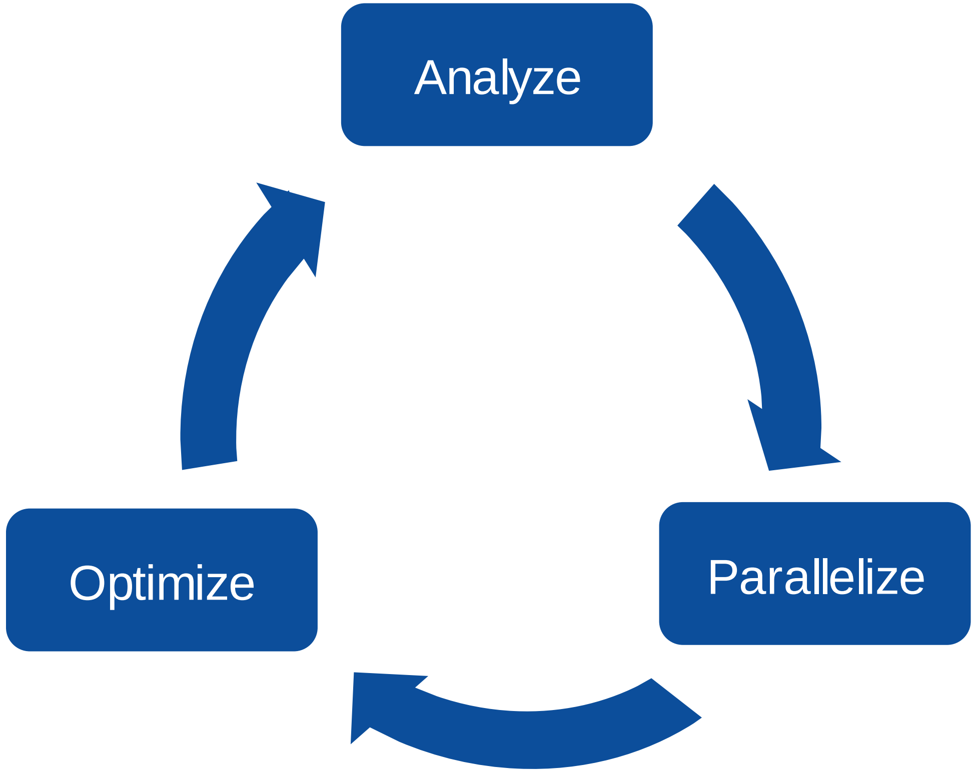
Name	Routine Summary:	Broadwell (v4) core (sec)
TVD Diffusion	Update diffusion scheme - using TVD slope + flux limiting.	7.36812
Magnetohydrodynamics	Calculate right hand side of MHD equations.	6.26662
Radiation Transport	Calculate radiation field and determine heating term (Qtot) required in MHD.	5.55416
Equation of State	Calculate primitive variables from conservative variables. Interpolate the equation of state tables.	2.26398
Time Integration	Performs one time integration.	1.47858
DivB Cleaner	Clean any errors due to non-zero div(B).	0.279718
Boundary Conditions	Update vertical boundary conditions.	0.0855162
Grid Exchange	Grid exchanges (only those in Solver)	0.0667914
Alfven Speed Limiter	Limit Maximum Alfven Velocity	0.0394724
Synchronize timestep	Pick minimum of the radiation, MHD and diffusive timesteps.	4.48E-05

Flowchart describes one timestep of the MHD equations: Function Description - (timer name)

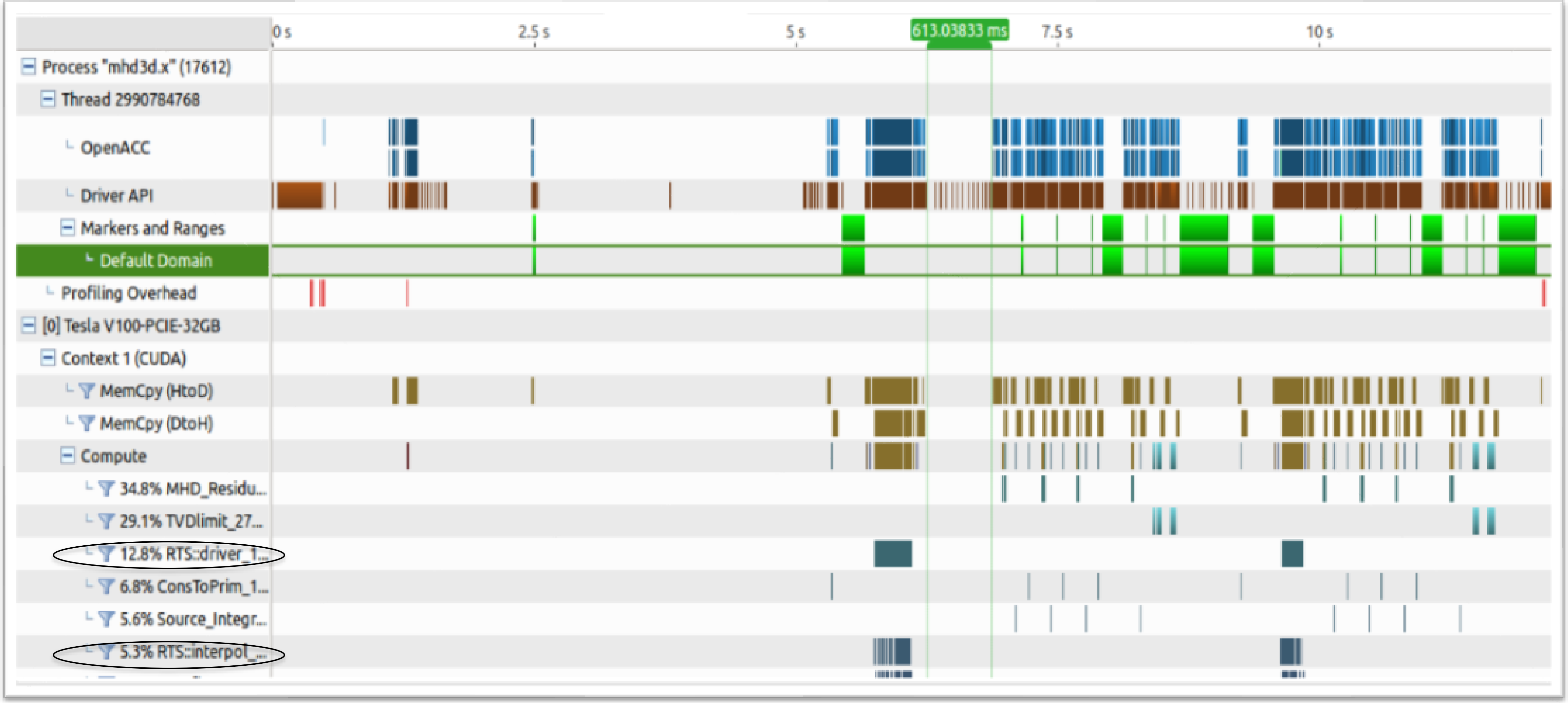


Profile driven parallelization

- Based on information gathered from profiling, implement simple development cycle:
  - Identify – which loops are currently the most impactful
  - Parallelize – the loop(s) for GPU execution
  - Verify – that our test cases pass with the new change
  - Reprofile/Optimize - until happy enough to move on



GPU Profile using nvprof



CUDA Occupancy Report

240x160x160 Dataset

Kernel Name	Theoretical Occupancy	Achieved Occupancy	Runtime % (GPU)
MHD	25%	24.9%	32.4%
TVD	31%	31.2%	31.6%
CONS	25%	24.9%	6.3%
Source_Tcheck	25%	24.9%	5.2%
Radiation Transport			
Driver	100%	10.2%	15.2%
Interpol	56%	59.9%	4.9%
Flux	100%	79%	1.5%

What did we learn so far?

- What is MURaM?
- What is the state of the project?
- What are the challenges identified?
- What problems do we still have to overcome?
  - Optimizing RTS
  - Learning from CUDA Occupancy Report

Toward Predicting Solar Eruptions

- **How do we get MURaM 100x faster? GPU-accelerated RT is the key.**
  - RT solver speed up on GPUs **(~85x)** a CPU core via wavefront algorithms (Chandrasekaran et al.)
  - RT iteration counts are reduced **(~2x)** on GPU's bigger subdomains.
  - But, we can rewrite RT solver to do required wavelengths (these are embarrassingly parallel).
  - Last two points play to GPU's strength: data parallelism.
  - **Estimate 450 GPUs could achieve breakeven (1 simulated hour/hour)**
- **Thousands of GPUs would be required to do full data assimilation.**
  - Expensive but not unthinkable (Summit has 27,600!)
- **Requirements play to strengths of GPUs, and trend in their design.**

Data Dependency Along Rays:

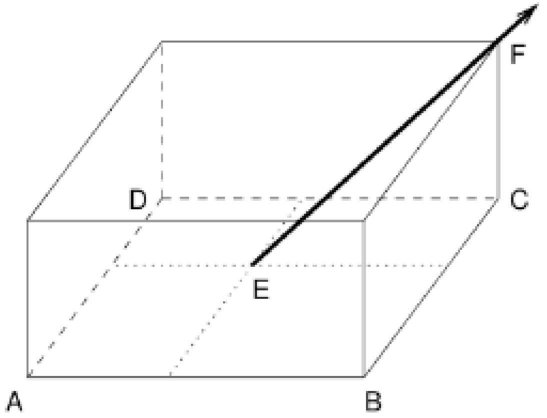


Figure 4.1: *The intensity at gridpoint F is obtained by solving the transfer equation along the short characteristic $\overline{EF}$. The intensity at the upwind point E is interpolated from the (already known) intensity values at the surrounding gridpoints, A to D.*

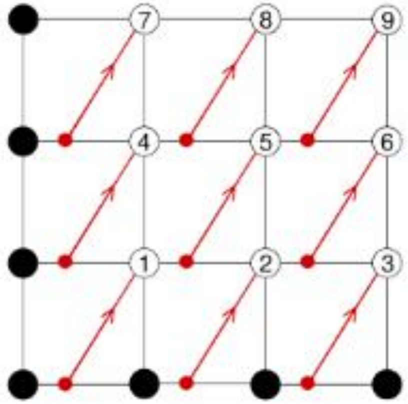


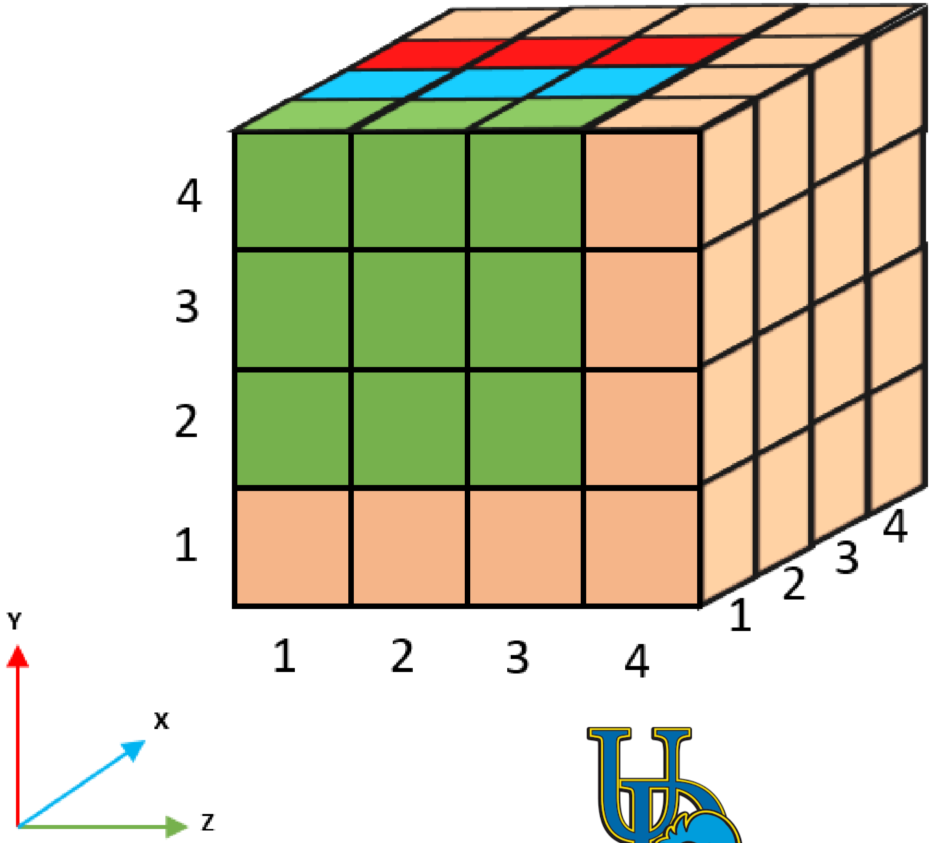
Figure 4.2: *The walking order of the Short Characteristics method in a 2D grid for a ray direction pointing into the upper right quadrant. Black circles represent gridpoints on the upwind boundaries, where the intensity values are assumed to be known.*

- Data dependency is along a plane for each octant, angle combo.
- Depends on resolution ratio, not known until run-time.
- Number of rays per plane can vary.

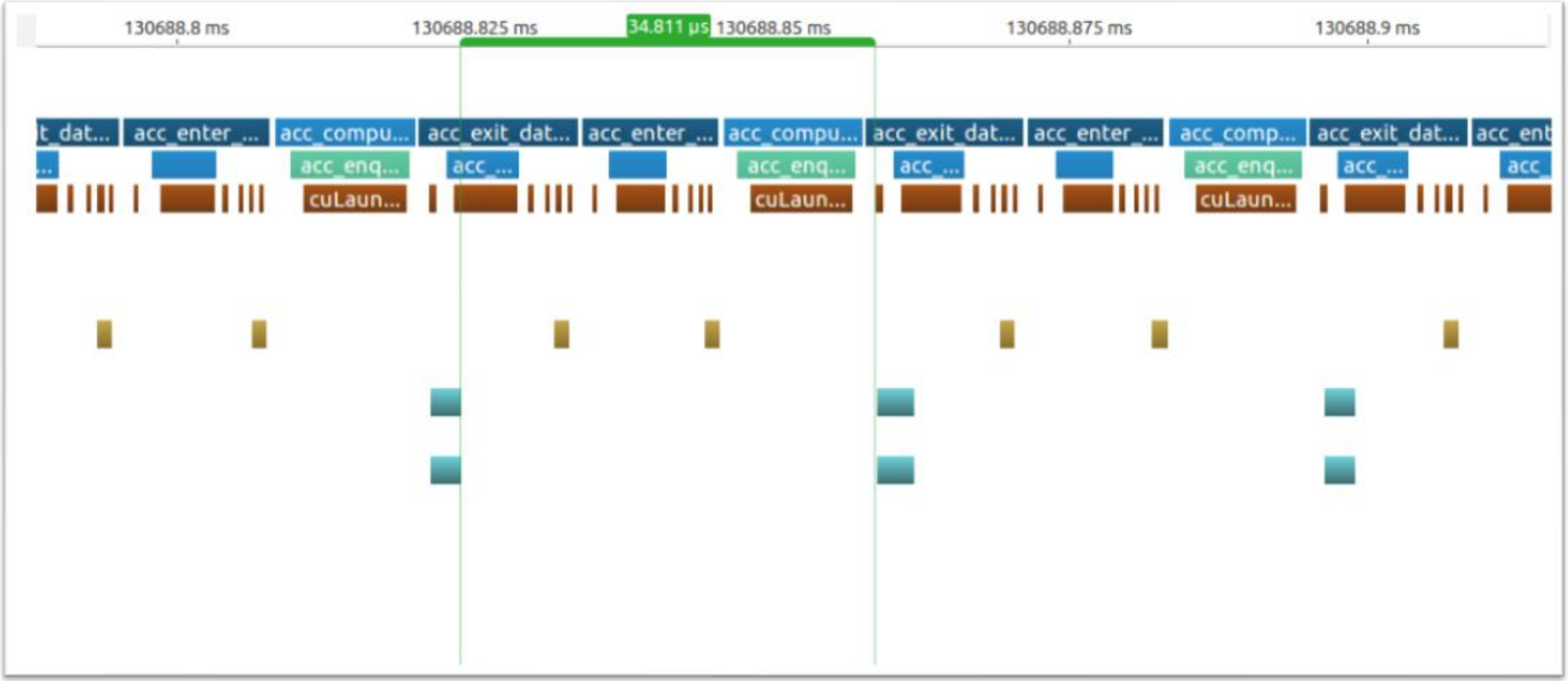
Vögler, Alexander, et al. "Simulations of magneto-convection in the solar photosphere-Equations, methods, and results of the MURaM code." Astronomy & Astrophysics 429.1 (2005): 335-351.

Solving RTS Data Dependency

- We can deconstruct the 3D grid into a series of 2D slices
- The direction of the slices is dependent on the X,Y,Z direction of the ray
- Parallelize within the slice, but run the slices themselves serially in predetermined order

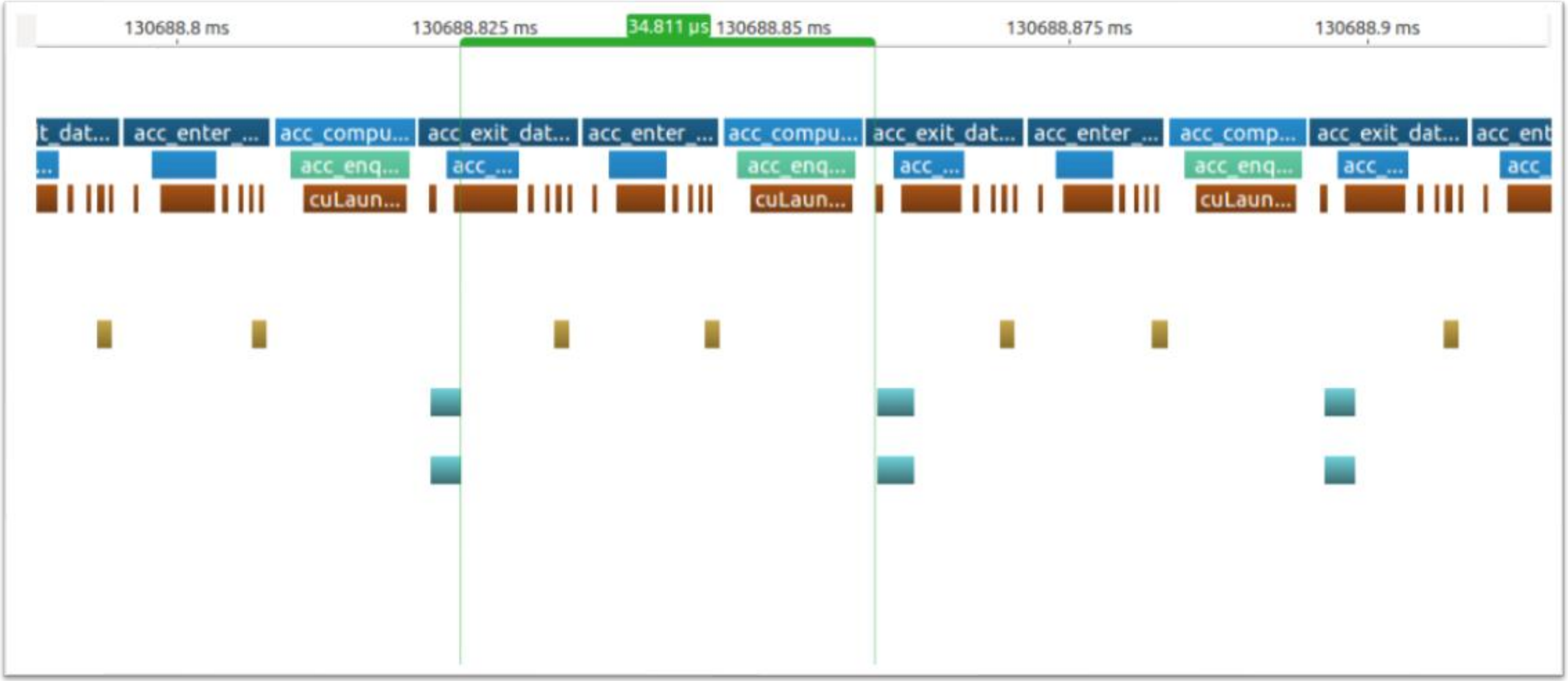


More profile driven optimizations

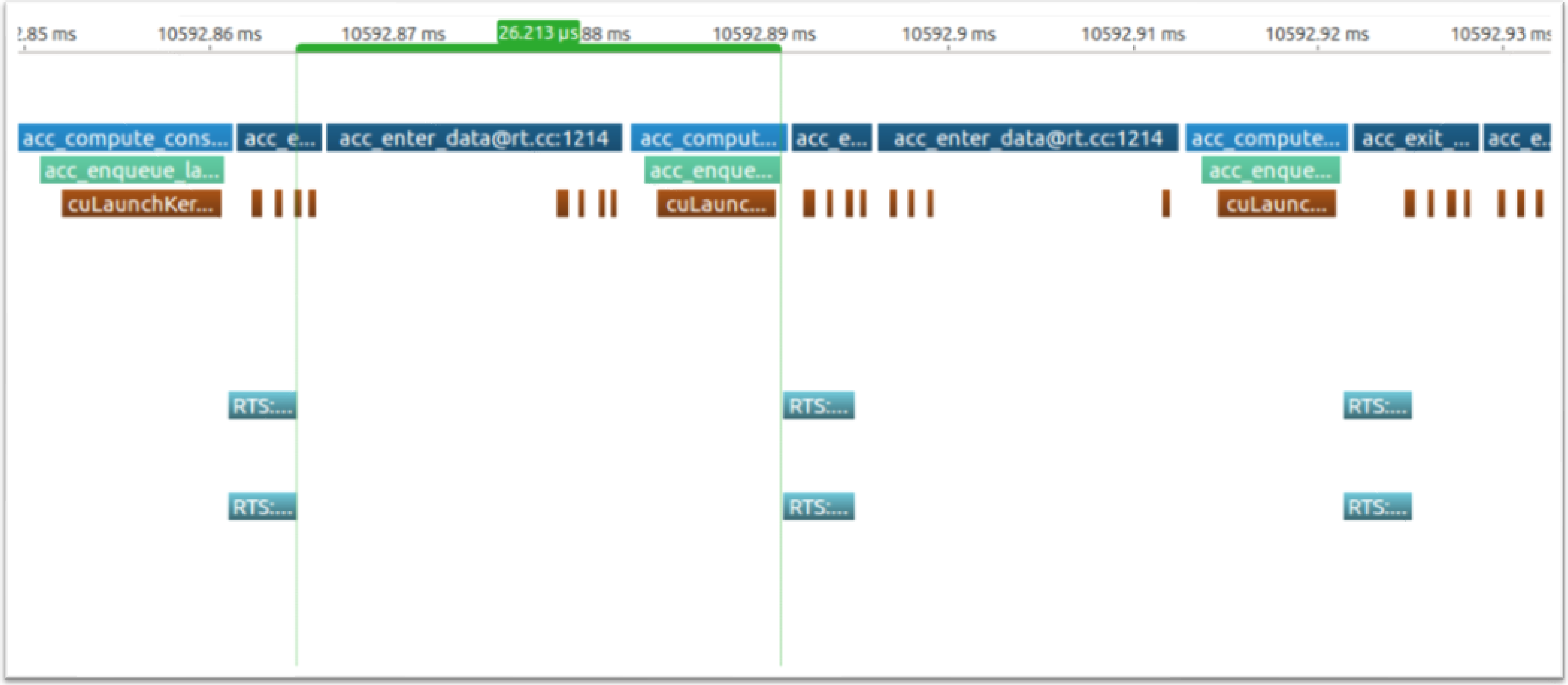


More profile driven optimizations

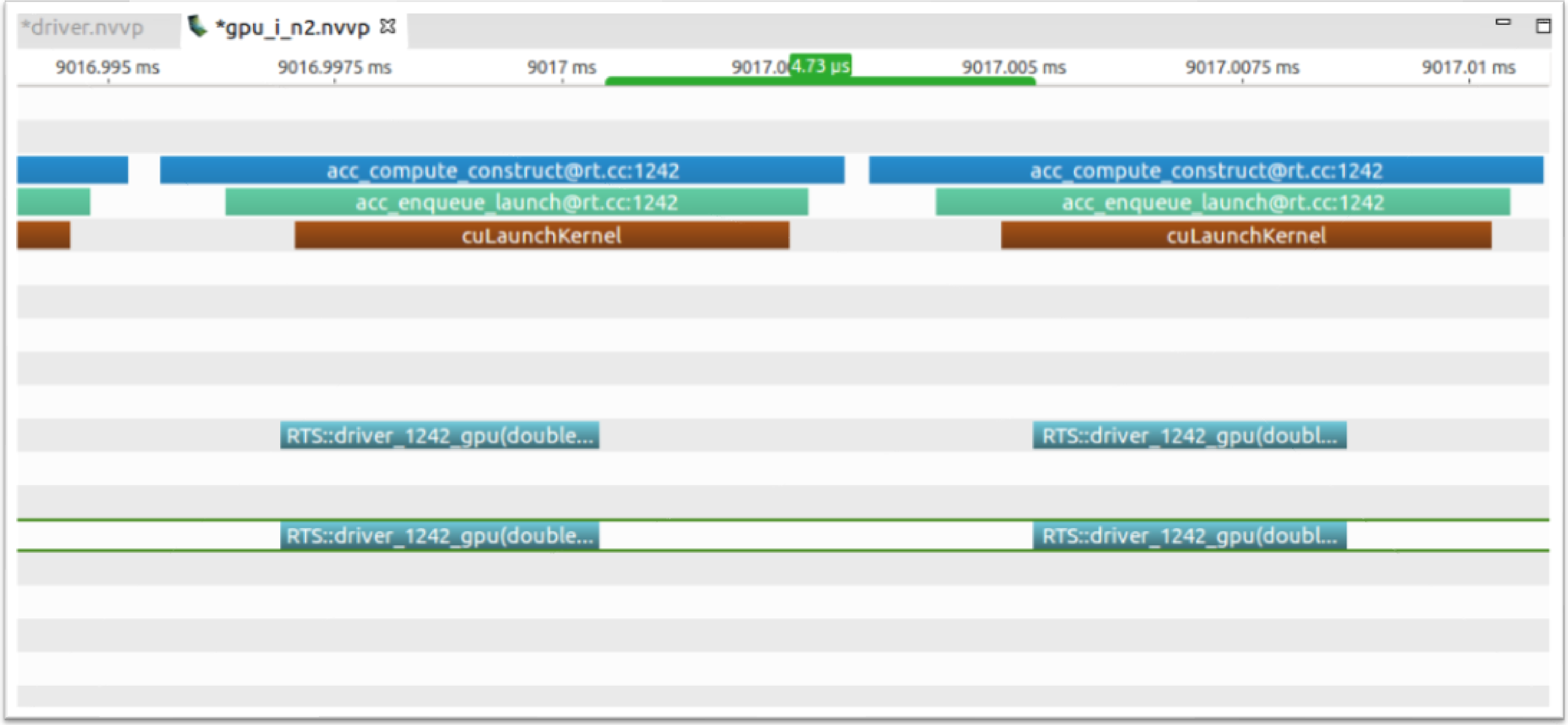
More profile driven optimizations

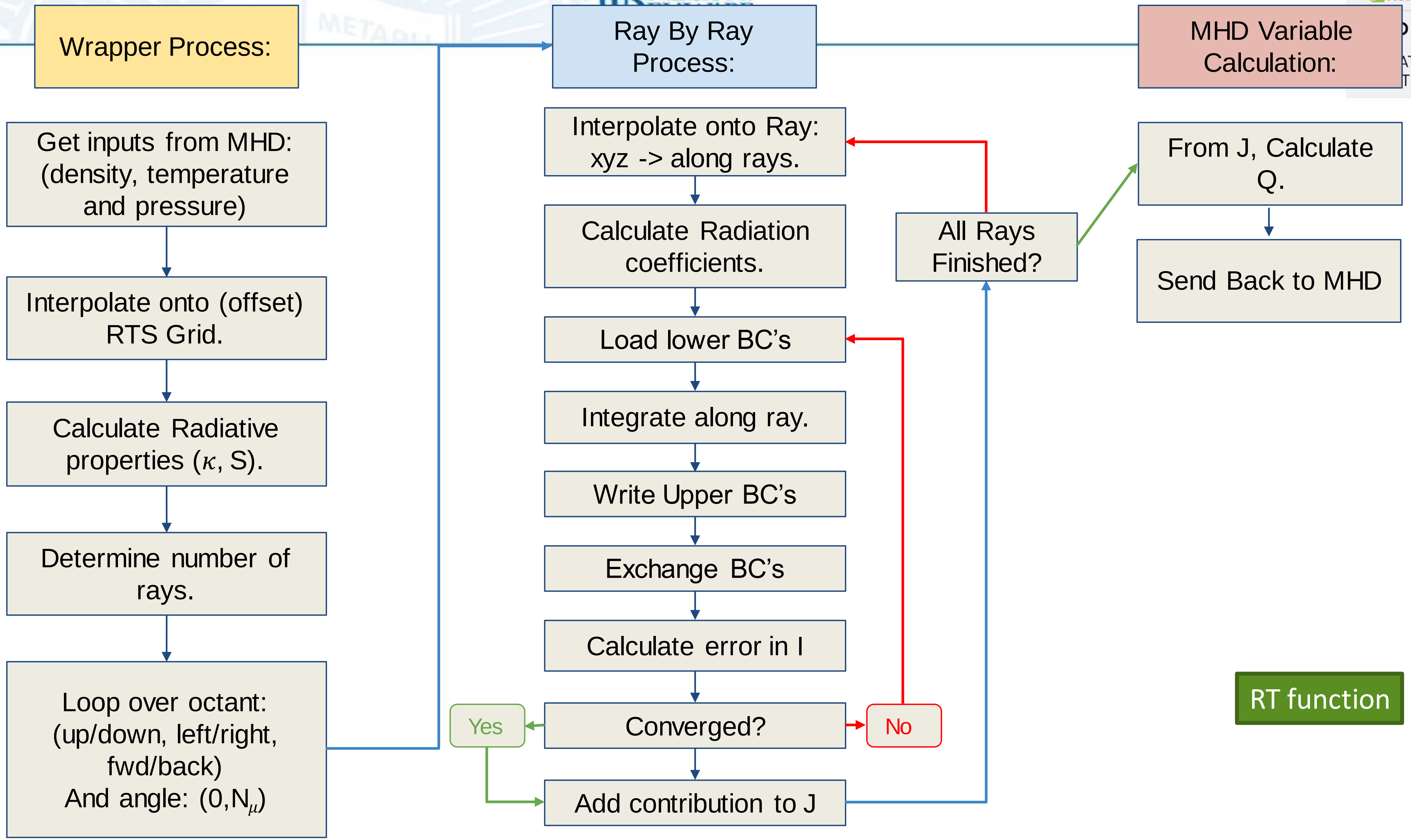


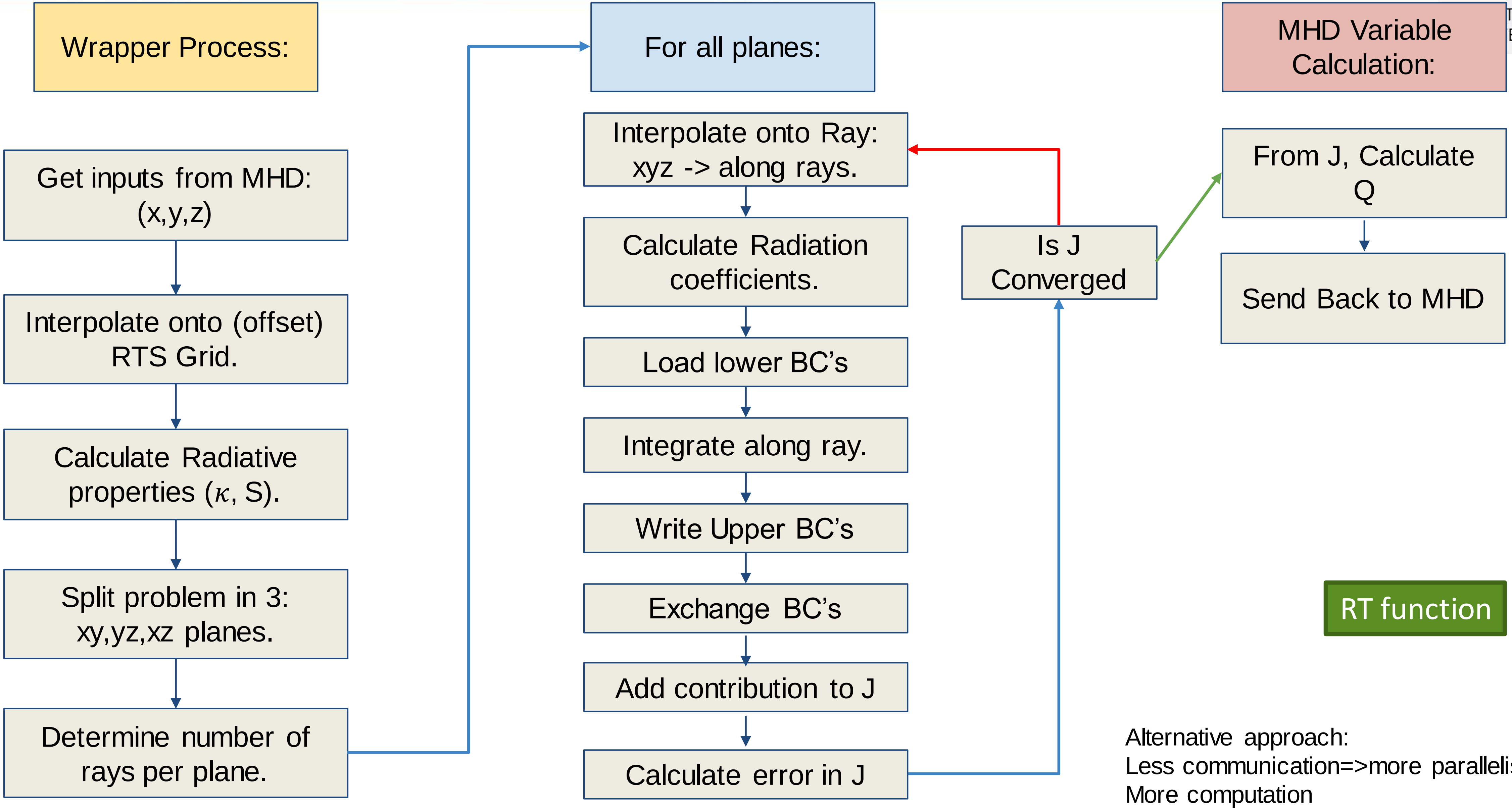
More profile driven optimizations



More profile driven optimizations




Wrapper Process:
Get inputs from MHD: (density, temperature and pressure)
Interpolate onto (offset) RTS Grid.
Calculate Radiative properties (κ, S).
Determine number of rays.
Loop over octant: (up/down, left/right, fwd/back) And angle: (0,Nμ)
Ray By Ray Process:
Interpolate onto Ray: xyz -> along rays.
Calculate Radiation coefficients.
Load lower BC's
Integrate along ray.
Write Upper BC's
Exchange BC's
Calculate error in I
Converged?
Yes
No
Add contribution to J
All Rays Finished?
MHD Variable Calculation:
From J, Calculate Q.
Send Back to MHD
RT function



Alternative approach:
Less communication=>more parallelism
More computation

What did we learn so far?

- What is MURaM?
- What is the state of the project?
- What are the challenges identified?
- What problems do we still have to overcome?
  - Optimizing RTS
- Learning from CUDA Occupancy Report

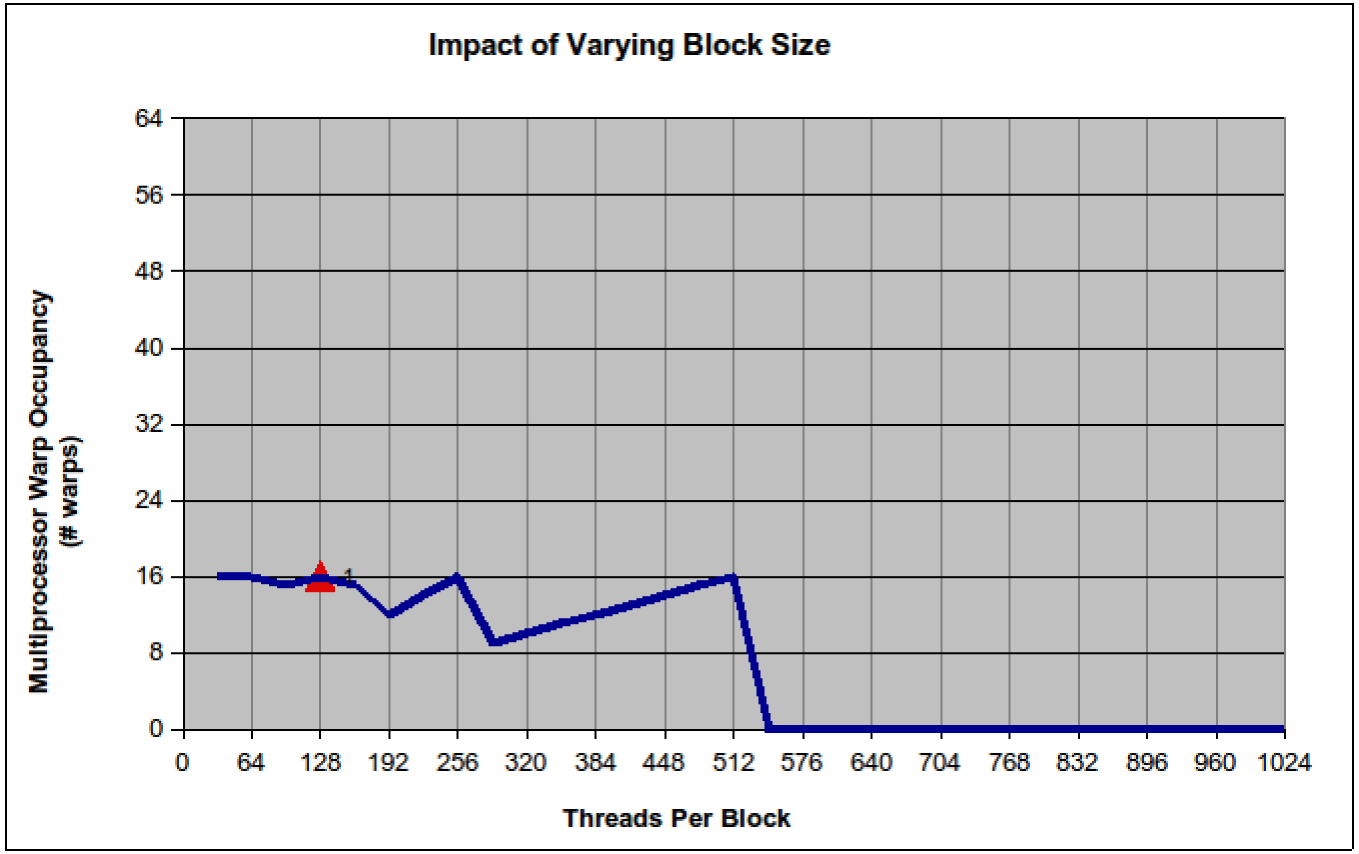
NVIDIA V100 specification

- 64 warps per multiprocessor (32 threads per warp)
- 65536 registers per multiprocessor
- 96 KB shared memory per multiprocessor
- Occupancy = # Used Warps / # Possible Warps
- How many warps we can use is dependent on how many registers and how much shared memory we use per thread

Thanks to NVIDIA for giving us access to their PSG system for our experiments!!!

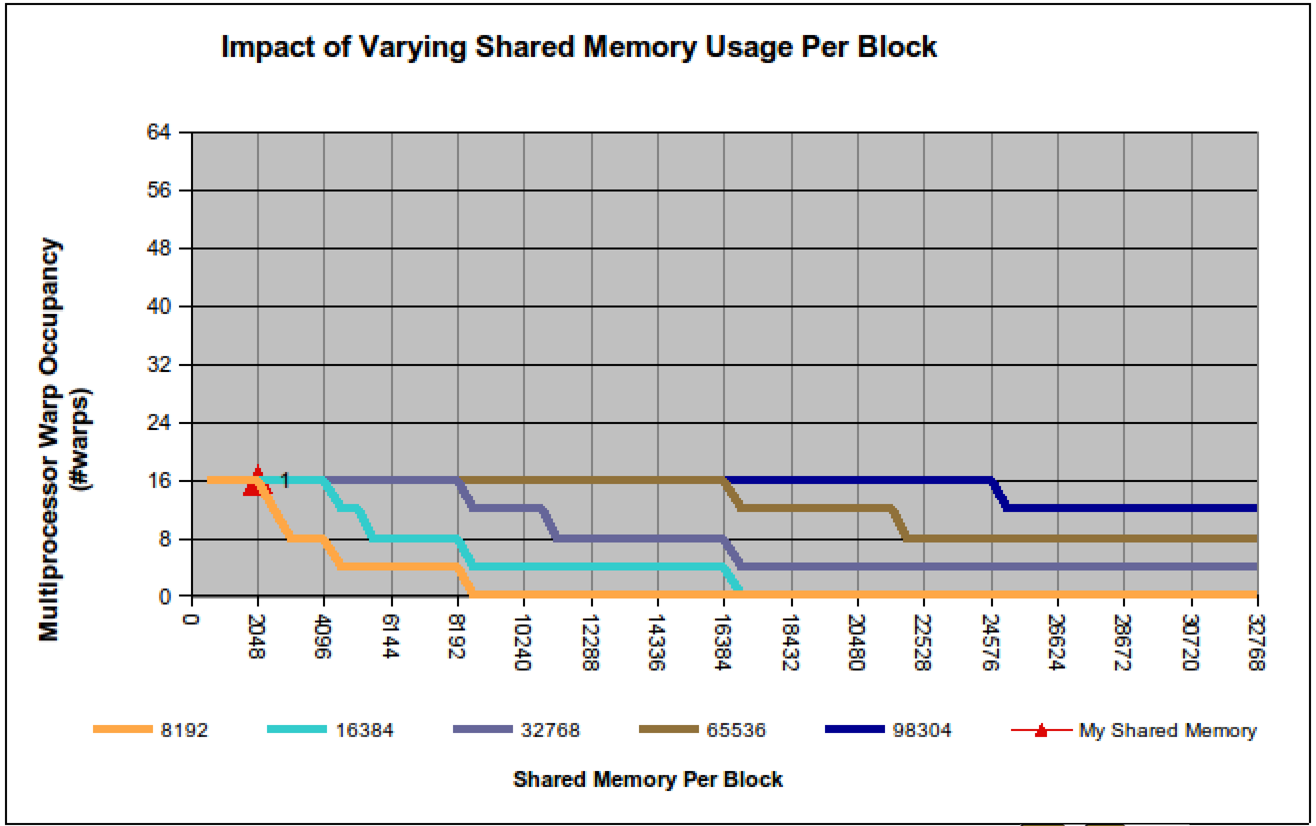
CUDA Occupancy Report

- Using the CUDA occupancy calculator
- We can see why our theoretical occupancy for MHD, TVD, etc is so low
- From this graph, we know that threads-per-block is not the problem



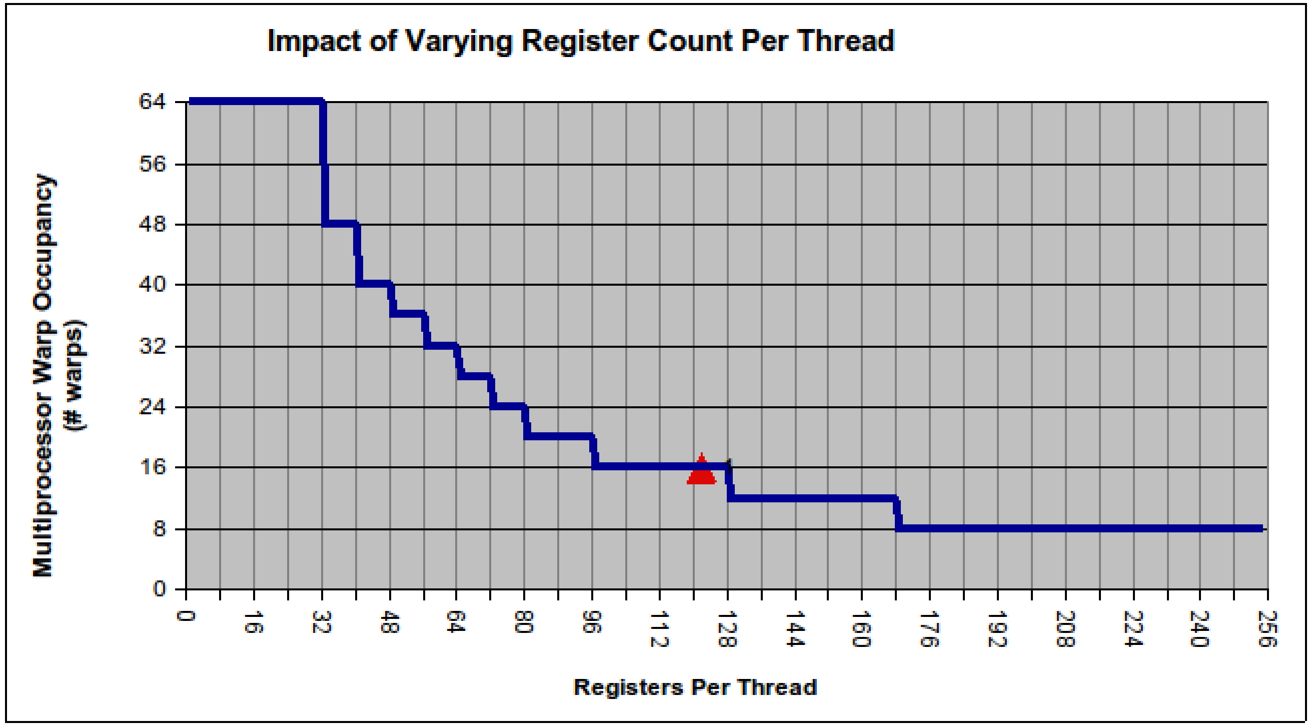
CUDA Occupancy Report

- We can also see that our shared memory usage is much lower than it could be
- One idea we have is to start moving some of our data to shared memory to get better usage



CUDA Occupancy Report

- Finally, we see that our problem is register usage per thread
- To get 100% occupancy, we would need to reduce register usage to 32 registers per thread



MURaM: Parallel processing using MPI

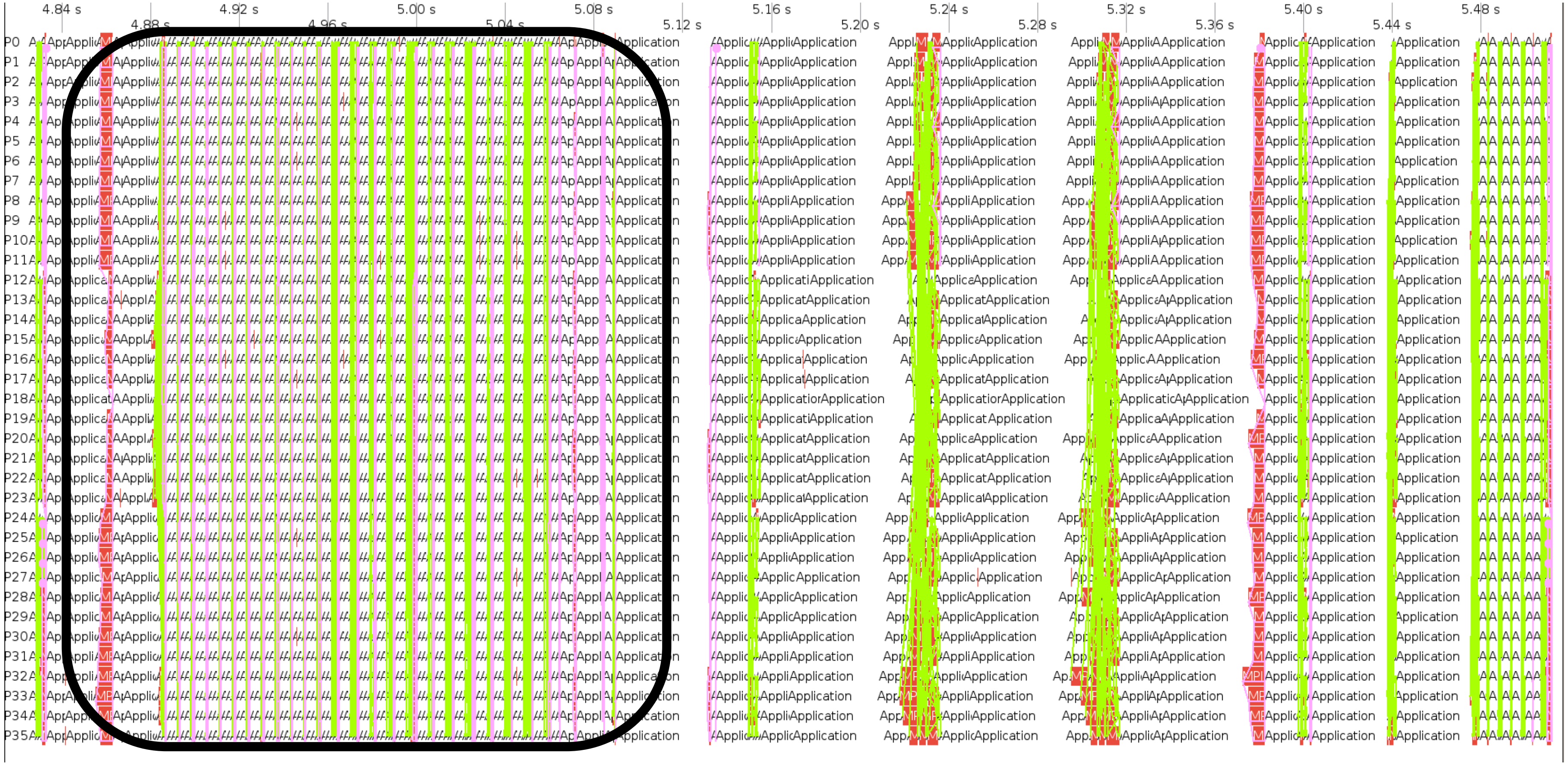
Introduction

- Three dimensional uniform cartesian grid.
- Grid is divided into smaller pieces based on MPI ranks and are processed independently.
- Halo information is communicated between the MPI ranks at a regular intervals.
- Large percentage of halo exchange happens in Radiative transfer.

Optimization strategies

- Overlap computation and communication in RTS.
- Perform a GPU to GPU direct communication without involving host.
- Optimize the packing and unpacking of communication data.

MURaM: MPI profiling using Intel Trace Analyzer



Communication in RTS for 24 rays

Results

Speedup of NVIDIA Volta V100 over ->	Singlecore	Full MPI node (32 cores)
RTS	27x	1.8x
TVD	36x	2x
MHD	18x	0.78x
Overall	15x	0.9x

Currently:

- 40% runtime is GPU
- 40% runtime is still CPU
- 20% is data movement
- As we finish porting the rest of the code to GPU, and optimizing the parts discussed today, we expect these results to improve significantly

Summary

- Accelerating the most significant routine – Radiation Transport, among other routines
  - Exploring how to optimize further
- Profiling and re-profiling – a Must
- Using directives enable maintenance of a single source code for both multicore and accelerators
  - Enables *new science*

Rich Loft, Eric Wright, Sunita Chandrasekaran
loft@ucar.edu, {efwright, schandra}@udel.edu