Acceleration of the Longwave Rapid Radiative Transfer Module using GPGPU

Mahesh Khadtare[1], Pragati Dharmale[2], Prakalp Somwanshi[3]

1 – I2IT, Pune, IN; 2 – SNHU, NH, US; 3 – CRL, Pune, IN

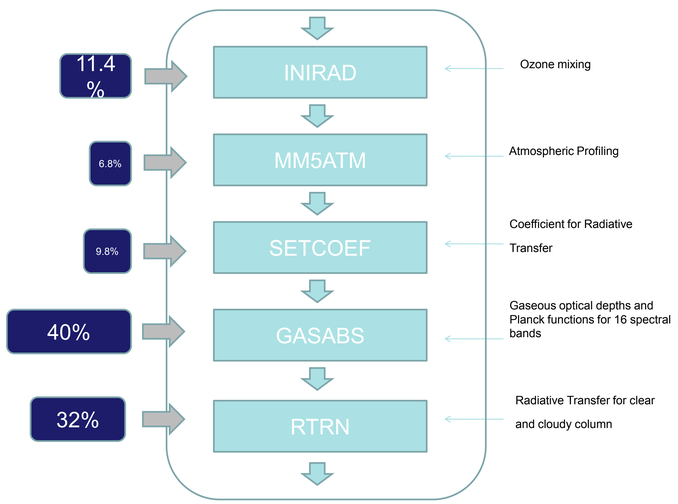
Abstract

This poster presents Weather Research and Forecast (WRF) model is a next-generation mesoscale numerical weather prediction system designed to serve both operational forecasting and atmospheric research communities. WRF offers multiple physics options, one of which is the Long-Wave Rapid Radiative Transfer Model (RRTM). Even with the advent of large-scale parallelism in weather models, much of the performance increase has came from increasing processor speed rather than increased parallelism. We present an alternative method of scaling model performance by exploiting emerging architectures like GPGPU using the fine-grain parallelism. We claim to get much more than 23.71x, performance gain by using asynchronous data transfer, use of texture memory and the techniques like loop unrolling.

Introduction

- WRF offers multiple physics options, one of which is the Long-Wave Rapid Radiative Transfer Model (RRTM).
- Longwave RRTM (Rapid Radiative Transport Model) is an optional model that computes the energy transfer through the atmosphere due to electromagnetic radiation.
  - Uses look-up tables for efficiency.
  - Separates calculation into 16 spectral bands.

RRTM Overview

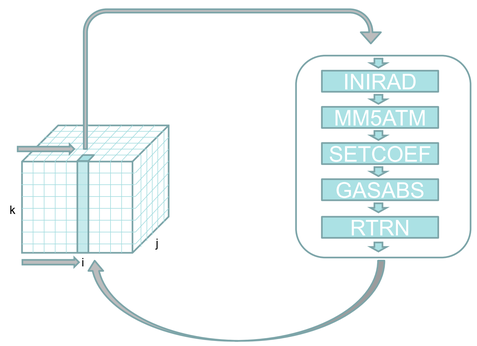


RRTM Present Status

RTM – Previous work

- RRTM proposed as benchmark kernel http://www.mmm.ucar.edu/wrf/WG2/GPU/
- Contains only CPU code.
- Nvidia claim to get 10X speed up but with different porting approach. (No implementation avaialble)

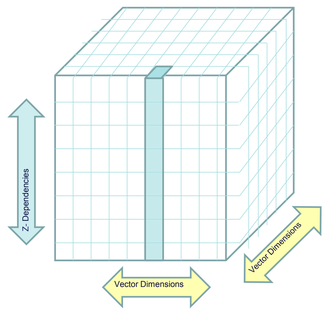
RRTM Sequential Approach



Implementation

Physics
- Predominantly vertical, "column physics"
- Perfectly parallel in the horizontal
- Many and varied dependencies in the vertical dimension

- RRTM only depends on data in the same vertical column.
  - GPU exploits this parallelism.

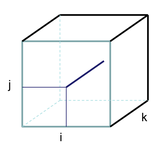


- The input data for this benchmark is a mesh of (x; z; y) = (73; 27; 60) elements.
- For any point in this three-dimensional mesh, radiation transfer only depends on data in the same vertical column.
- This independence between vertical columns was exploited for parallelism in developing the GPU version.

The original code and input data for the RRTM benchmark were obtained from the kernel Benchmark Page at NCAR. The code contains a driver for both CPU and GPU versions, as well as a serial CPU implementation of RRTM.

Data Layout

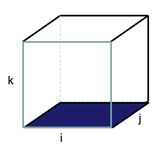
CPU Layout



A(i,k,j)
Outer loops over (i,j)

Extract 1D arrays in k and send to RRTM

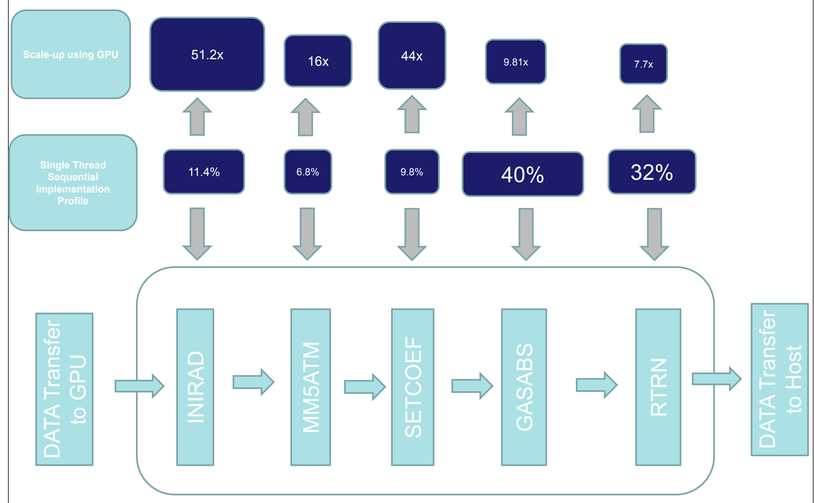
GPU Layout



Reorder to A(i,j,k) after transfer

All routines launch 'nx*ny' threads
each thread calculates one column in 'nz' dimension.

RRTM Scale-up after Porting



Results

CPU / GPU	Core Details	Time (sec)	Speedup
AMD Athlon X2 Dual Core	1	1067	1x
Tesla GeForce GTX 275	192	165	6.4x
Fermi C2050	448	98	10.88x
Fermi S2050	4x448	45	23.71x

Input data is on a (nx,nz,ny) mesh with nx = 73, nz = 28, ny = 60

References

[1] Online http://www.mmm.ucar.edu/wrf/WG2/GPU/
[3] NVIDIA Corporation, Compute Unified Device Architecture (CUDA), *http://developer.nvidia.com/object/cuda.html.*