Quantitative Spectroscopy in 3D
From Serial to Parallel Computing

A Personal View

Lars Koesterke

Texas Advanced Computing Center

"Recent Directions in Astrophysical Quantitative Spectroscopy and Radiation Hydrodynamics"

In Honor of Dimitri Mihalas

Overview

Collaboration with: Carlos Allende Prieto — University of Texas (Austin), now England
Ivan Hubeny — Arizona
David Lambert — University of Texas (Austin)

Later with: Ivan Ramirez
Hans-Günter Ludwig
Martin Asplund
Remo Collet

Solar Abundances
Calculating the Solar Spectrum

ASSET: Advanced Spectrum Synthesis 3D Tool

What can Modern Software (Design) and Supercomputers do for you?

The World is 3D …

… and sometimes even time dependent

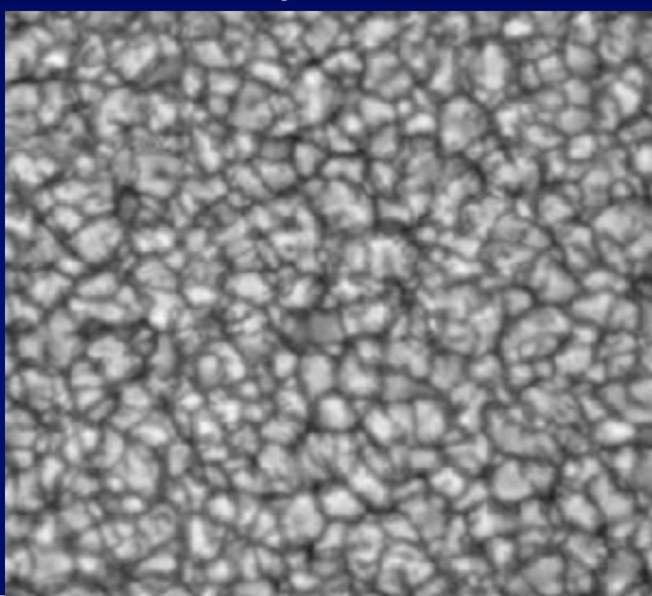
Example: The Sun

Analysis in 3D: 40% decrease of the Solar Oxygen Abundance

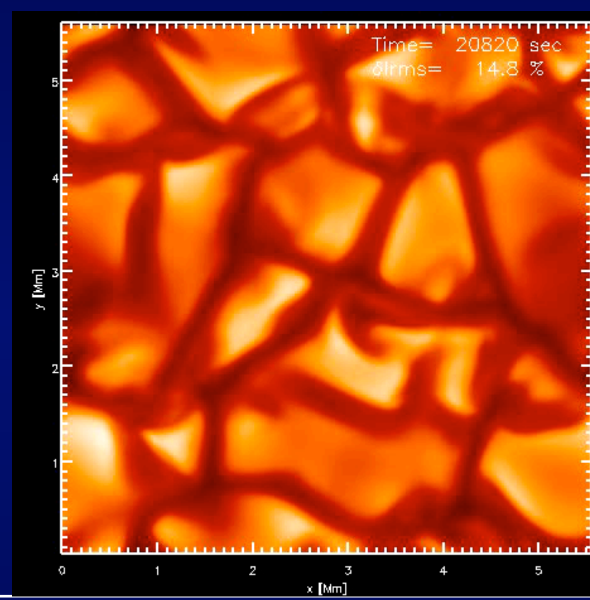
(Allende Prieto, Lambert, Asplund: ApJ 2001)

Huge Consequences and
upset Helioseismologists (who claim measurements down to the 0.01% level)

Solar Photosphere: Observation



Model



Hydro-Models by:

- Freytag, Ludwig & Steffen
- Asplund, Collet et al.
- … and many other groups

Challenges

- Quality/Value of 3D Hydro-Models still (somewhat) under debate
- 3D Spectra under debate
- Unresolved Discrepancies with Helioseismic Analysis
- 3D Hydro-Models full of hidden treasures …

Why? Calculating Spectra in 3D is *a bit* different than in 1D

Tremendous Task

- Arbitrary Opacities
- Molecules
- What about Scattering?
  (You'll need 2 codes!)
- Execution Speed

Problem Size (per frequency)

	1D	3D (bare minimum)
Rays:	3	$3 \times N_A \times N_x \times N_y \times N_S$
		$N_A = 8$ (Azimuth)
		$N_X = 50$ (Size in X)
		$N_Y = 50$ (Size in Y)
		$N_S = 100$ (# of Snapshots)

factor ≥ 2,000,000

… and all the Multi-D Interpolations

TACC

Task at Hand (2004−2007)

Write a Code that takes Hydro-Models and calculates Spectra …

- Arbitrary opacities, i.e. no constant background opacities, so that it can (in principle) cover large spectral regions
- Molecules, Scattering
- 1 spectral line (~200 frequencies) in a reasonable time frame (overnight)
  averaging over ~100 Snapshots
  on a medium-size Workstation (slow CPU & < 1GB of memory)

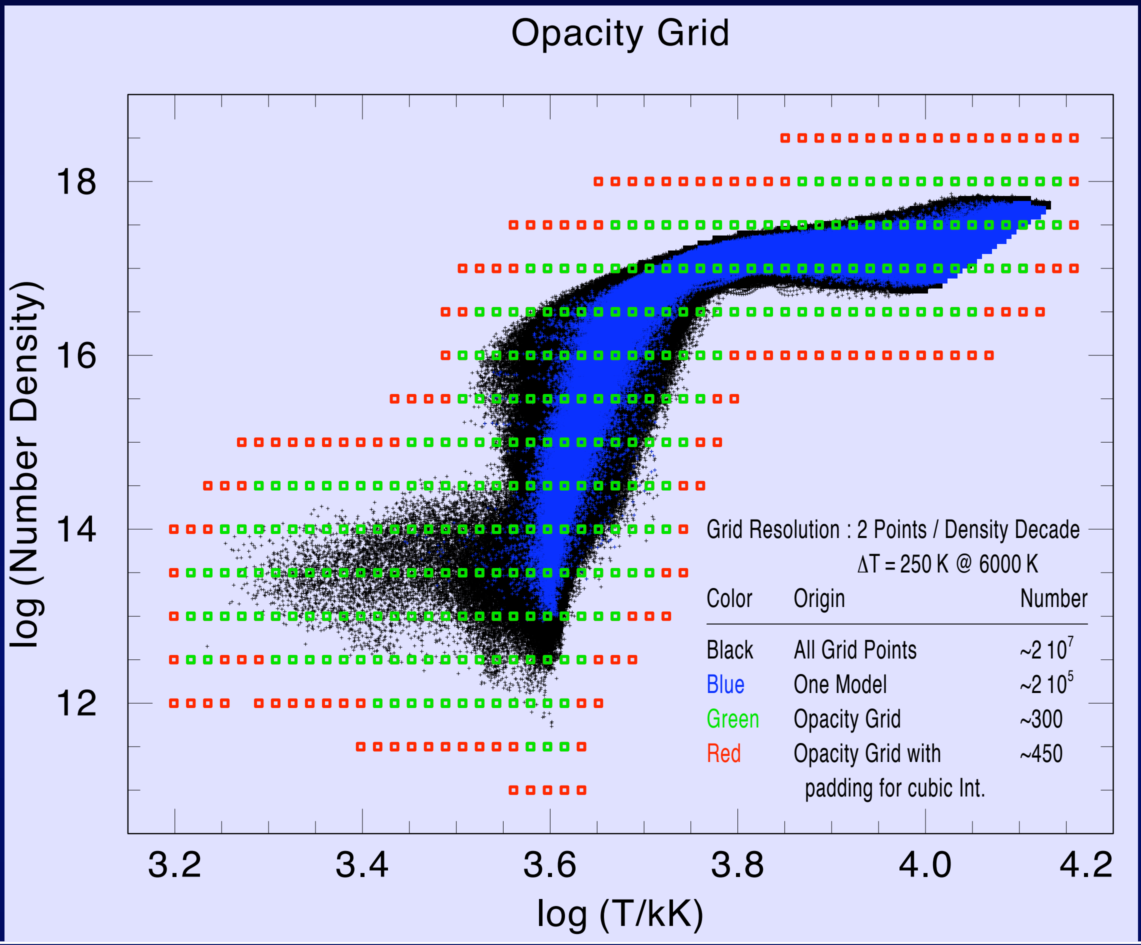
ASSET: Advanced Spectrum Synthesis 3D Tool

- 2 Codes:
  - Short-Characteristics for Mean Intensity (J)
  - Long-Characteristics for Intensity and Flux (I/F)
- Arbitrary Opacities (Opacity Grid, Modified Version of SYNSPEC)
- Cubic, Bezier Interpolations (Auer, Tübingen workshop 2002)
- Radiation Transfer by-the-book (no short-cuts)

Opacity Grid

Trick for Calculations in LTE

1 Snapshot: 50 x 50 x 82 ~ 200,000 grid points (Temperature and Density)
100 Snapshots: 20,000,000 grid points



Blue : 1 Snapshot
Black : 100 Snapshots
Green : Linear Interp.
Red : Cubic Interp.

Equidistant in T and ρ
Cubic Interpolation in 2D

Grid Resolution:
- 4 per decade in ρ
- 250K at 6000K
- ⇒ 450 grid points

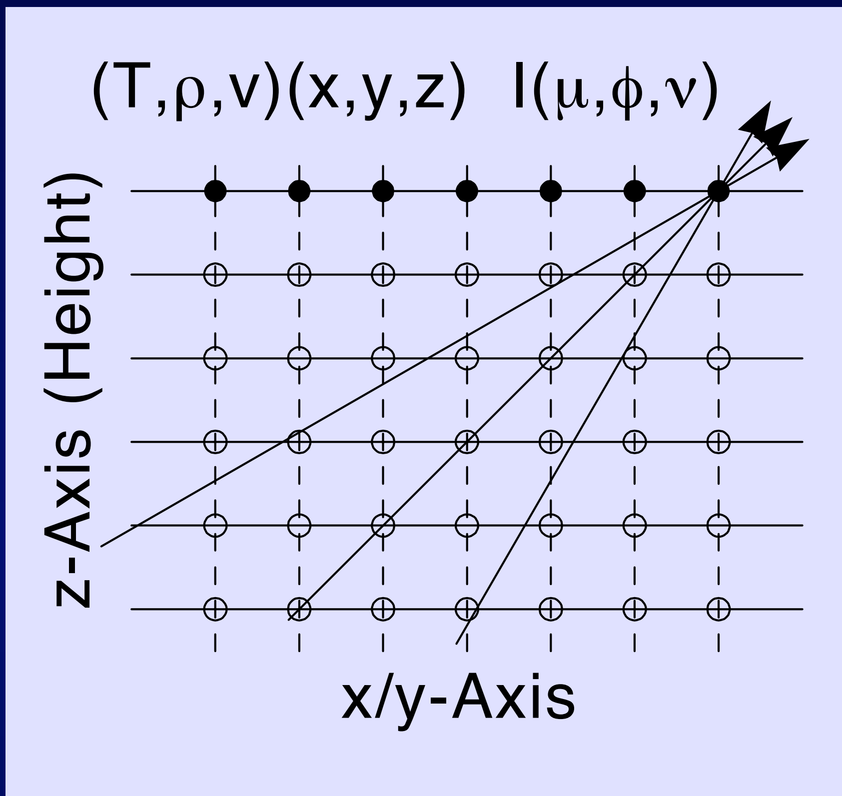
Residual about 0.01%

TACC

Radiation Transfer in 3D

Calculation of the Intensity: Long-Characteristics

- Data-Cube (X/Y/Z): Temperature, Density and Velocity
- Have Opacities and Source Functions ready for all Grid Points (circles)
- Integrate along Rays starting at outermost layer (filled circles)



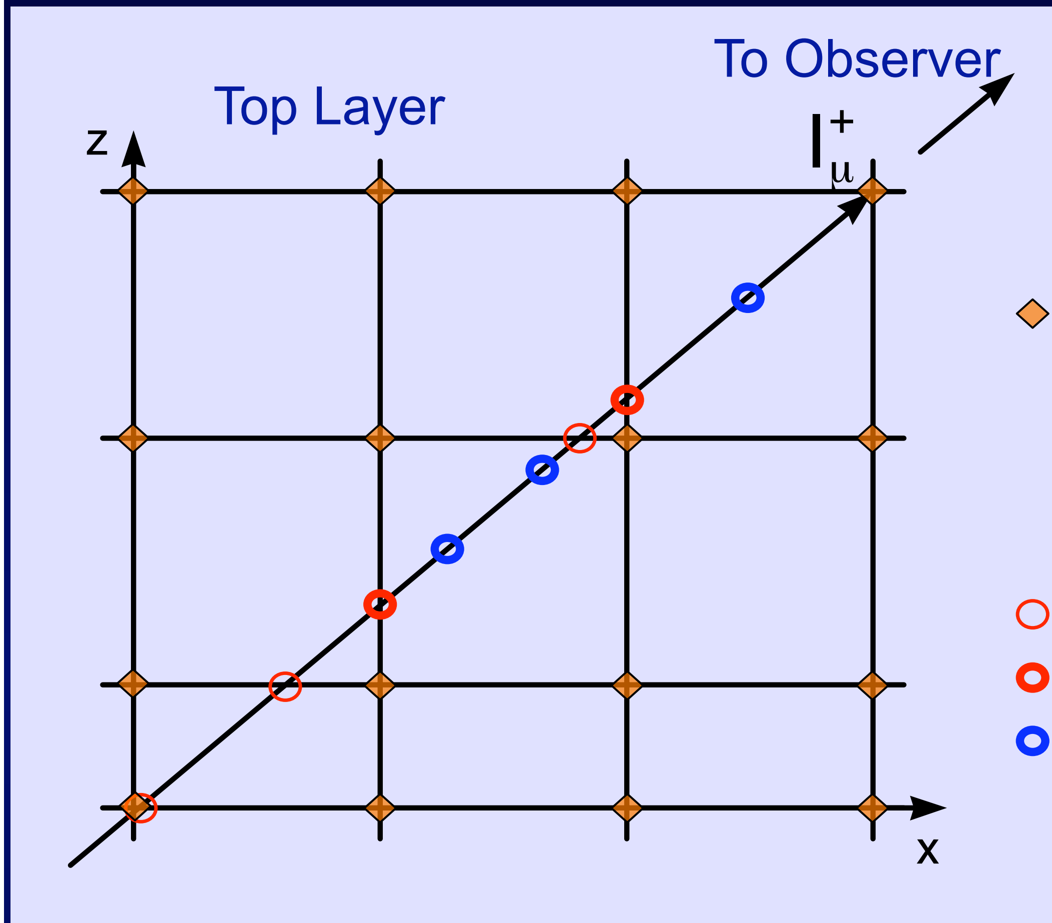
Rays do not hit Grid Points
⇒ Apply Interpolations

<u>Data is 4D</u> (X,Y,Z + Velocity)
⇒ Interpolate in 3 Dimensions
$4^3 = 64$ data points
21 interpolations in 1D

TACC

Calc. of the Intensity: Long-Characteristics

- Start at all Grid Points at Top Layer (X/Y plane)
- Follow Rays down to Tau ~ 20



Input Data:
Opacities, Source Function
Mean Radiation Field J
(from other code)

Insert Points where Rays intersect the Grid:
z-plane
x/y plane
} 2D space + 1D freq.
Velocity field: 3D space

TACC

2007: “New” Career at TACC

TACC: Texas Advanced Computing Center

TACC: Home of Ranger

Member of the High-Performance Computing Group
Performance Evaluation and Optimization

- User Support, Software Support
- Documentation, Training
- Research in Computer Science
- Research in Astrophysics

Fields of Expertise:
- Optimization
- Parallel Computing
- etc.

Requirements: Understanding of Parallel Software Paradigms, namely OpenMP and MPI

Learning by doing: Parallelization of ASSET with OpenMP and MPI

Ranger @ TACC

Sun Constellation Linux Cluster

(all members of US institutions can apply!)

- Compute power – 579 TeraFlops
  - High in Top500 list
  - 3,936 four socket Sun blades
  - 15,744 AMD “Barcelona” CPUs (2.3 GHz)
  - 62,976 Compute cores (2GB Memory/core)
  - 4 Flops/cycle (Streaming SIMD Extension)
- Memory – 123 TeraBytes
- Disk subsystem – Lustre
  - 1.7 PetaBytes
- Interconnect – Infiniband
  - 2 switches
  - 10 Gbps with 1.6-2.9 μsec latency
- Power – 3MW (0.6 for cooling)

… and Spur, a very powerful system for visualization with 32 GPUs

Why OpenMP and MPI?

… it's quite an effort!

- MPI: Message Passing Interface
  - Distributed Memory Machines (Ranger, any large cluster, etc.)
  - Each MPI task works independently on a chunk of frequencies
  - No sharing of Memory
- OpenMP: Open Multi-Processing
  - Shared Memory Machines (individual compute nodes on Ranger, any cluster or Desktop)
  - OpenMP threads work together on multiple CPUs/cores on a chunk of frequencies
  - Memory is shared!

Memory Requirements are high!

Standard Grid: 50 x 50 x 82

Enhanced Grid: 200 x 200 x 100

J, 48 Angles: 500MB ⇒ 8GB

I/F, 21 Angles: 800MB ⇒ 13GB

Ranger has 32GB of Memory in a blade: 4 quad-core CPUs with 2GB per core

With OpenMP full-size calculations can be done easily!

Another advantage is better load-balancing: fewer MPI tasks, larger frequency blocks

Optimization

Serial Optimization & OpenMP: on One Workstation

Original Design on Mac Mini: 180 frequencies = 1800s (100 Snapshots = 1 Day)

Laptop technology: slower dual-core CPU with smaller cache, slower Memory, slower Disk

+ Free Compiler: g95 (serial!)

Speedup	Measure
2.5	Laptop CPU ⇒ High-end CPU (Intel Penryn, AMD Barcelona) Compiler: g95 ⇒ Intel, PGI
3.7	OpenMP on 4 cores
5	Trading Memory for Operations (400MB ⇒ 800MB) Optimizing Memory Access (allows for more Optimization later on)
8	Serial Optimization
~400	Total Speedup: 10x Compiler, (Multicore) Hardware & OpenMP 40x Serial Optimization

Medium-Level Workstation with 4 cores: 180 freq., 100 Snapshots = 10 min.

Optimization I

Trading Memory for Operations, Optimizing Memory Access

a) Using 800MB (up from 400MB) avoids the recalculation of quantities
b) Rearrangement of Loop Structure improves Memory Access

The Memory Subsystem is the Bottleneck in any Computation!

- Memory Access has to be as "gentle" as possible
  "gentle" ⇒ Stride−1 access: Cache friendly and enables efficient Prefetching
- Unfortunately only the Frequency Loop facilitates Stride-1 Access

(Loop in X would be also possible)

Old Loop Structure (5 Loops)
- Frequency
  - Angles
    - Starting Point Y
      - Starting Point X
        - Along the Ray

New Loop Structure (6 Loops)
- Frequency Block (contains ~192 Frequencies, MPI)
  - Angles
    - Starting Point Y (OpenMP)
      - Starting Point X
        - Along the Ray
          - Frequencies in Block

Optimization II

- Compiler Options (particularly the Inline-Size; what to inline)
- Arrangement of Arrays in Memory
- Bundling of inner Routines for Interpolation (2D)
- Reverse Inlining (Beefing-up innermost Routines)
- Avoiding Divisions (beyond the obvious)
- Manual Loop Unrolling and Loop Blocking (Help the compiler!)
- Check Assembly Code: Type Conversion, Vectorization
- Save Operations (beyond the obvious)
  Tau positive & exp(−Tau) = exp(0−Tau) ⇒ count Tau negative & exp(Tau)
- Make use of SIMD: Single Instruction Multiple Data (4 Ops/cycle)
- Disable SIMD Alignment Check (Intel compiler directive)
- …

Optimization III

Every Challenge is also a Chance!

Routine for Radiation Transfer
- Hand-tuned and Highly Optimized (Manual Loop Blocking + Loop Unrolling)
- Highest Optimization of the Compiler (−O3 −fp-model fast)

Causes occasionally spurious floating-point exceptions

Solutions:
- Compilation with reduced Optimization ⇒ 20% Penalty overall
- No manual Loop Blocking and Unrolling ⇒ 20% Penalty overall
- Use of 2 Subroutines and Fortran2003 Exception Handling ⇒ wait/pay for Intel 11
- Custom-made EXP function

Custom EXP function:
- Chebychev Polynomial 5th order, almost double precision (5×10^{-14})
- Total of 12 (13) floating-point operations (add/mult)
- Fortran code (no Assembly!)
- At least 50% faster than built-in exp function (Overall Speed-up: 16%)

TACC

What can we do with a Speed-Up of 400?

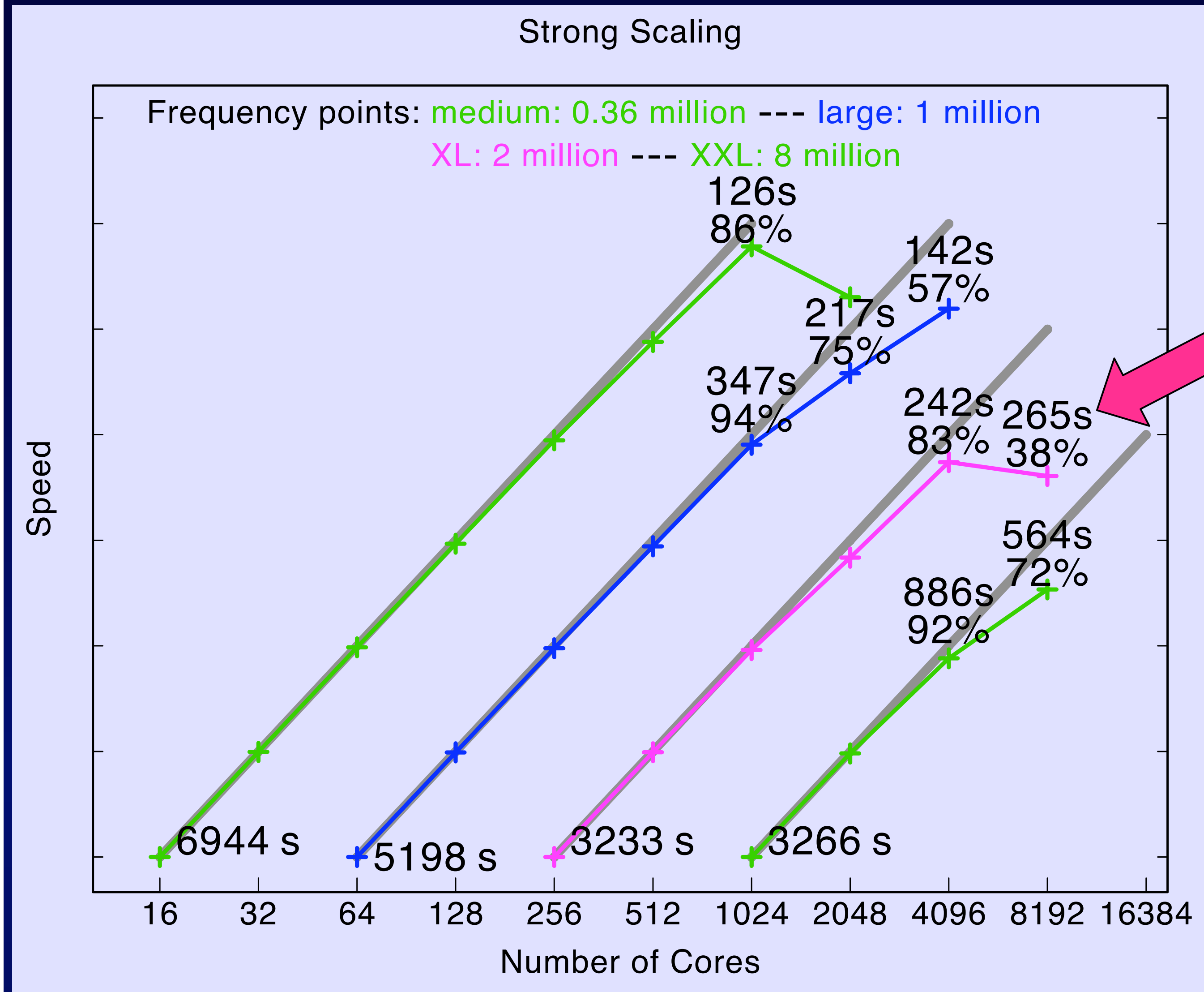
Calculating the whole Solar (Stellar) Spectrum!

- 2,000 − 30,000 Å
- Resolution 1.3 million (0.3 x Thermal Width @ T(min) of Fe-lines)
- 3.3 million frequency points

How long does it take?	1 Day per Snapshot on a Workstation
100 Snapshots, 100 Ranger blades (You may get away with 25)	2.5% of Ranger for 1 Day (which caused some problems when I asked for attending this conference)
(Without Serial Optimization (f ~ 40)	100% of Ranger for 1 Day)
Only partial Spectrum	1 Workstation is sufficient

TACC

Strong Scaling Experiment



Full Solar Spectrum:
Up to 4096 cores
6% of Ranger

Scaling breaks down
at Runtimes below
5 minutes

Problem is too small
or, Ranger is too big!

1D vs. 3D

We are now in 3D where we were in 1D in the early 90's!

- Desktop
  - Single line in a few minutes
  - Chunk of Spectrum (~500Å) overnight
- Small Cluster
  - Full Spectrum overnight

Get Ready to Explore the Spectra of 3D Hydro-Models!

Current and Future Activities

Current Activities:

- Calculation/Analysis of the full Spectrum of K-Dwarfs
  - with Ivan Ramirez, Carlos Allende Prieto, et al.
  - Asplund models
- Calculation/Analysis of the full Solar Spectrum
  - with Carlos Allende Prieto, Hans-Günter Ludwig, et al.
  - Asplund models and Co5Bold models

… but there are more Hydro models out there …

New Collaborations and Projects wanted!

- Martin Asplund, Remo Collet and Regner Trampedach have ~20 Hydro-Time series covering a substantial part of the HRD …

- There are other groups that might be interested in using ASSET …

It's a perfect opportunity: A powerful tool + Great access to compute resources

Activities: Code Development

- Upgrading ASSET to calculate population numbers and spectra in non-LTE
  - Using the “J” code in an Accelerated Lambda Iteration (ALI)
  - Upgrading the “I/F” code for spectra/population numbers in non-LTE
- Upgrading ASSET to Hydro-models with AMR (Adaptive Mesh Refinement)
  - I’d like to talk to some AMR experts at this conference to bounce some ideas around …

Funding

We collaborate with scientists and actively pursue funding opportunities!

If you need somebody for …

- Code Optimization (Serial & Parallel)
- Porting code to Large Clusters
- Improving Scalability
- Implementation of Parallel Computing (OpenMP & MPI)
- Code Development
- Radiation Transfer

… please let me know!

TACC

Thanks …

… and Thank You Dimitri!