Data Management Challenges in Modelling Core-Collapse Supernovae

Collapsing a Star without Collapsing an Infrastructure

Eric S. Myra
DEPT. OF PHYSICS & ASTRONOMY
STATE UNIVERSITY OF NEW YORK AT STONY BROOK

SDM Center All-Hands Meeting
2-3 March 2005
Salt Lake City, UT

Ground to be covered…

- A bit of astrophysics
- Challenges with data and where they're leading
- Our efforts and goals with SPA (Scientific Process Automation)
- Challenges of the next stages

Collaborators for this work

- Doug Swesty (SUNY at Stony Brook)
- Jim Lattimer (SUNY at Stony Brook)
- Amy Irwin (SUNY at Stony Brook)
- Dennis Smolarski (Santa Clara University)
- Polly Baker (Indiana Univ)
- Ed Bachta (Indiana Univ)

- Terence Critchlow (LLNL)
- Xiaowen Xin (LLNL)

Our Workflow (and opportunities for SDM-based improvements)

NATURE

PHYSICS

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{v}) = 0$$

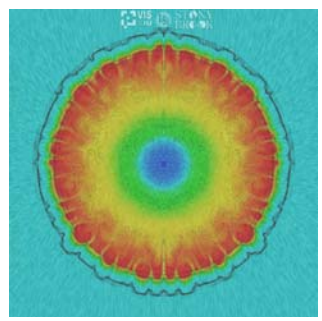
$$\frac{\partial \rho Y_e}{\partial t} + \nabla \cdot (\rho Y_e \mathbf{v}) = -m_b \sum_f \int d\varepsilon \left(\frac{\mathscr{S}_\varepsilon}{\varepsilon} - \frac{\bar{\mathscr{S}}_\varepsilon}{\varepsilon} \right),$$

$$\frac{\partial E}{\partial t} + \nabla \cdot (E \mathbf{v}) + P \nabla \cdot \mathbf{v} = -\sum_f \int d\varepsilon \left(\mathscr{S}_\varepsilon + \bar{\mathscr{S}}_\varepsilon \right),$$

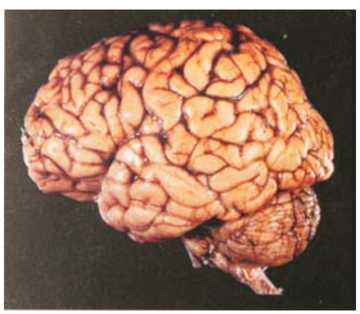
$$\frac{\partial \rho v_i}{\partial t} + \nabla \cdot (\rho v_i \mathbf{v}) + (\nabla P)_i + \nabla \cdot \left\{ \sum_f \int d\varepsilon \left(\chi_\varepsilon E_\varepsilon + \bar{\chi}_\varepsilon \bar{E}_\varepsilon \right) \right\} + \rho (\nabla \Phi)_i = -\sum_f \int d\varepsilon \, (A_\varepsilon)_i$$

NUMERICS

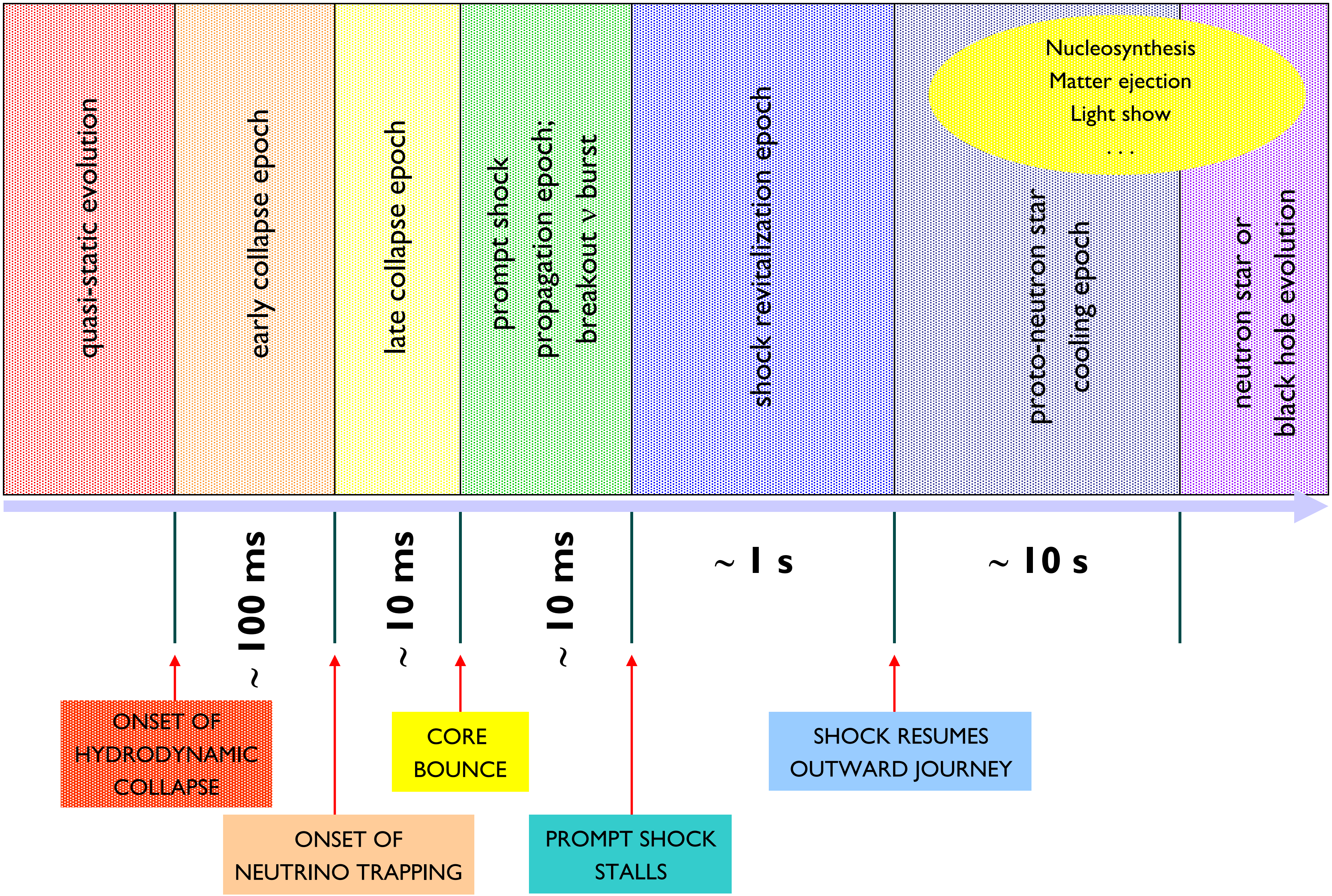
VISUALIZATION



INTERPRETATION

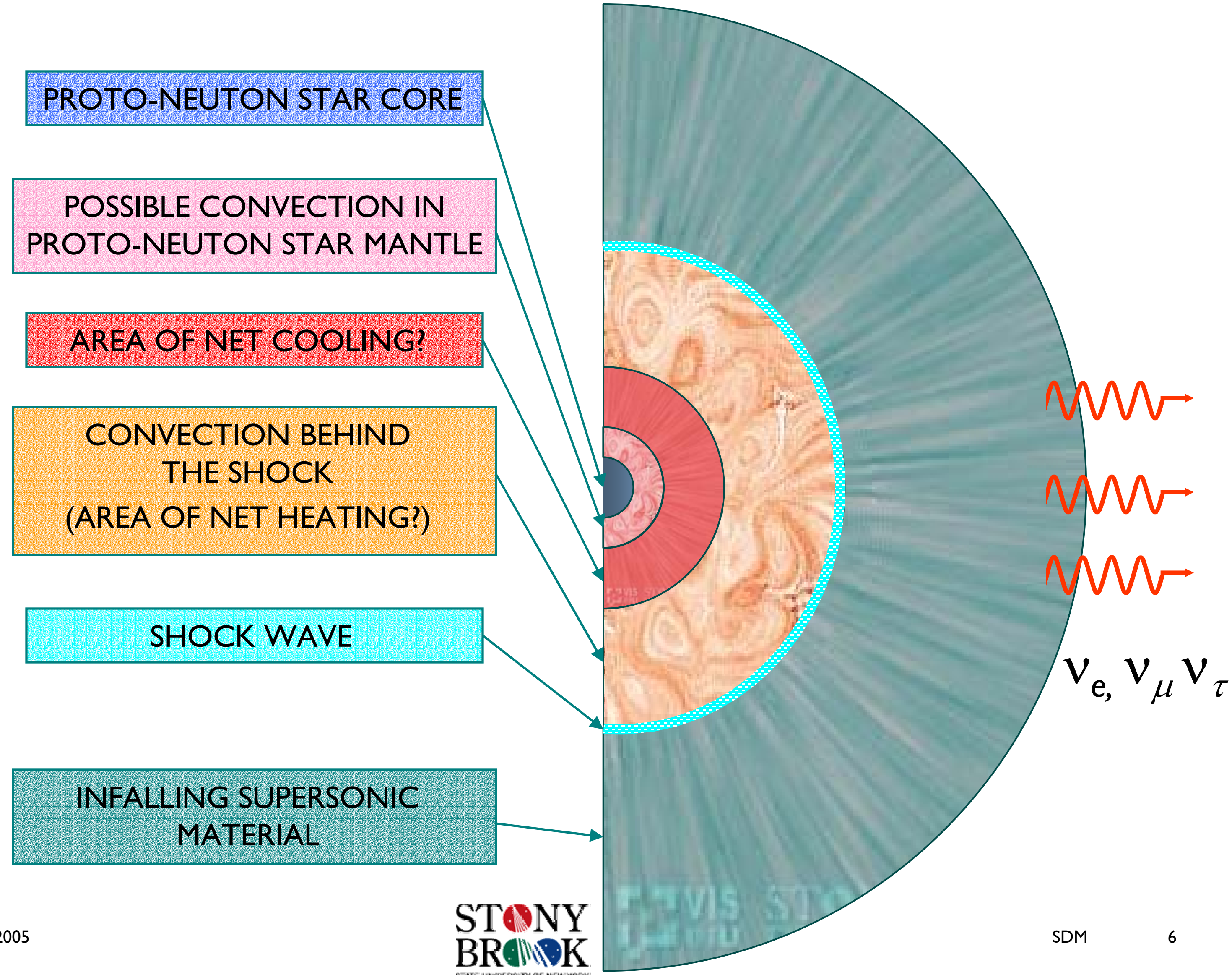


Hypothetical Timescales for Supernova Processes



Multi-dimensionality in Core-Collapse Supernovae:

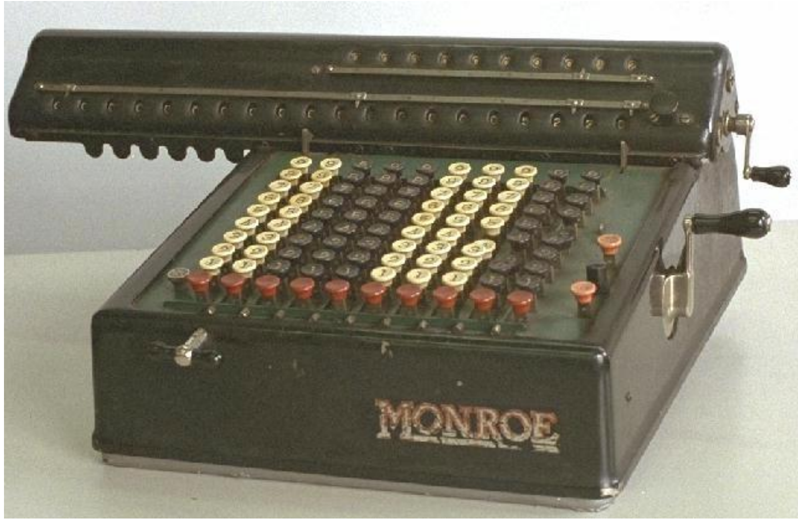
How convection complicates the picture



Data Management Challenges

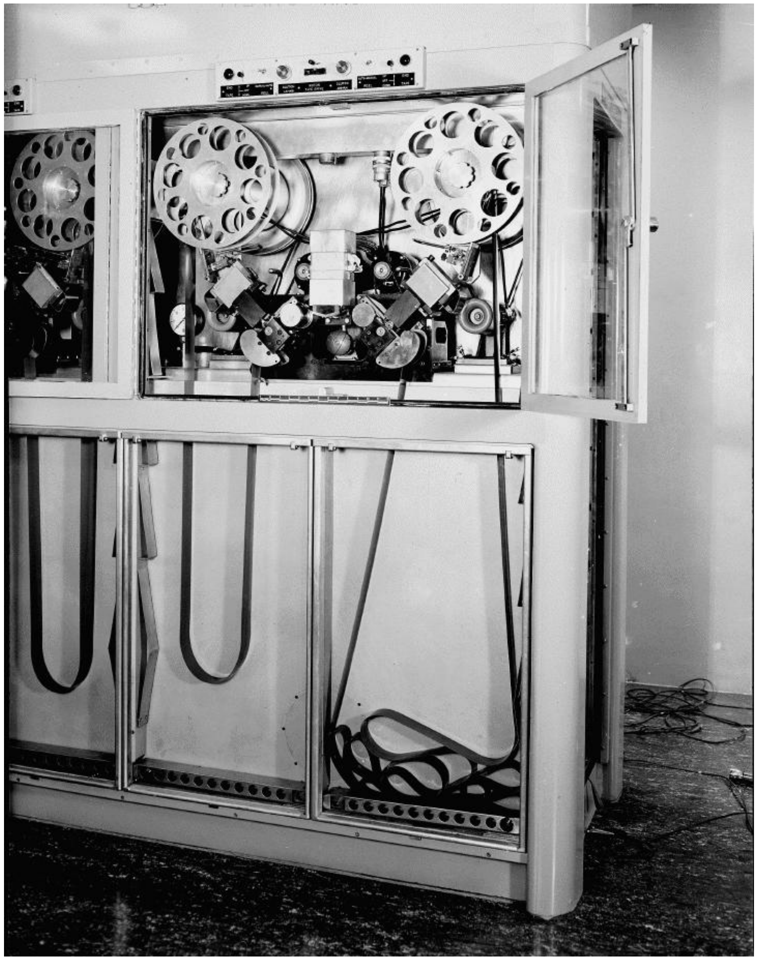
- Generation of datasets
  - computational, I/O performance
- Storage of datasets
  - scratch space, tertiary storage, local storage (placement issues)
- Transmission of datasets
  - bandwidth, time, labor
- Manipulation and re-inflation of datasets
  - extraction of the science
- Replication of datasets
  - how many copies floating about?

We have issues with computing speed



- Computers are never fast enough.
- Queues are long.
- Turnaround is slow.
- We anticipate our needs growing faster than capability growth of systems in the coming years.
- ***Automated job management desirable as job volume grows…***

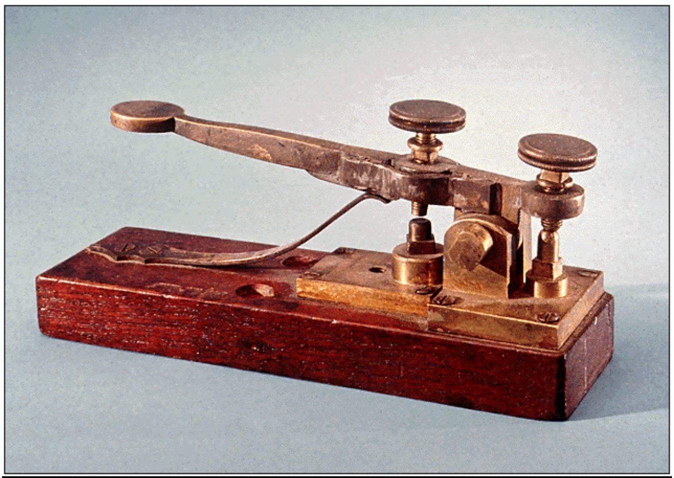
We have issues with I/O



- We are dealing with scalability issues in parallel I/O.
- Temporary disk space a problem on all HPC systems.
- Data volumes soon to explode.
  - better I/O
  - more runs
  - larger problems

- ***Automated migration of data highly desirable…***

We have issues with networking

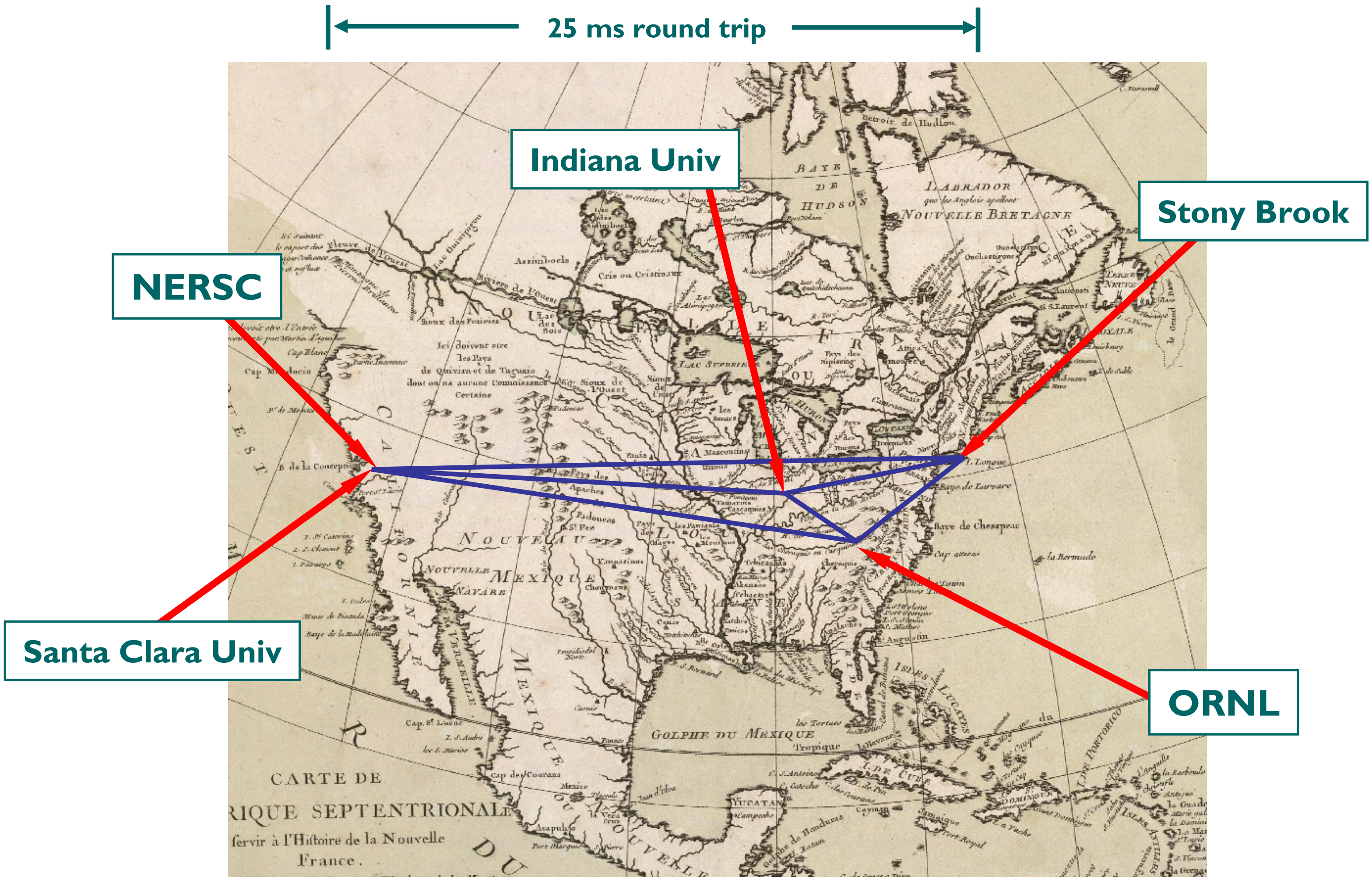


Stony Brook's connectivity retains many primitive aspects.

- Logistical networking is a major benefit.
- Networking limitations dictate which data are local and which remain remote.
- Without workflow management, much human time must be devoted to manually managing files.

- ***The latency-hiding features of automatic workflow management is highly desirable…***

We have issues with geography:
Distributed computing, analysis, collaboration



A pioneering byte of data faces a tortuous transcontinental journey.

Characteristics of Our Nuclear Astrophysical Simulation Data

- **Origin**:
  - from hydrodynamic, thermodynamic, magnetohydrodynamic, or radiation-transport components of a simulation
- **Disk Access Patterns:**
  - data written and read primarily from structured or block-structured AMR grids
  - unstructured grid or particle data is possible in the future
  - writes and reads done via parallelized I/O (MPI-I/O + HDF5)
  - large number of processes (¿ 1024)
  - write once, read multiple times (but on a different system)

The Current File Situation

- **Checkpoint files:**
  - Captures the minimal state of the model required for restarting a simulation
  - Also serves as the visualization dump file for post-processing
  - With 256x256 x 20 x 6 grid, sized at **70 MB per file**
  - Typical current debugging run: 500 such files → **35 GB per run**
  - Full production run: 10 000 files → **700 GB per run**

- **"Diagnostic" files:**
  - Post-processed inflation of checkpoints to recover physically interesting quantities
  - Current typical size: **> 200 MB → 100 GB – 2 TB per run**
  - Currently, *non-permanent* data

A nuisance, but not data-management-at-the-frontier stuff!

Within the next year or two…

- **Checkpoint files:**
  - With 128x128x128 x 20 x 6 grid: **2 GB per file**
  - Typical debugging run: 500 such files → **1 TB per run**
  - Full production run: 10 000 files → **20 TB per run**
- **Diagnostic files:**
  - Predicted typical size: **> 6 GB → 3 – 60 TB per run**
  - Non-permanent data – *whew!*
- **"Results" files:**
  - Plots, movies, formatted data that have permanent value.
  - Size yet undetermined
  - From the electronic workbook to the electronic library
  - File size less of an issue than annotation, ability to "mine" for features, *etc.*

This is looking nastier!

Further out…

- **Checkpoint files:**
  - With (128x128x128) x (16x16x20) x 6 grid: **500 GB per file**
  - Typical debugging run: 500 such files → **250 TB per run**
  - Full production run: 10 000 files → **5 PB per run**

- **Diagnostic files:**
  - Post-processing explosion of checkpoint to recover physically interesting quantities
  - Predicted typical size: **> 2 TB → 0.8 – 20 PB per run**

Goals for Workflow management:
How we hope SPA will improve our lives

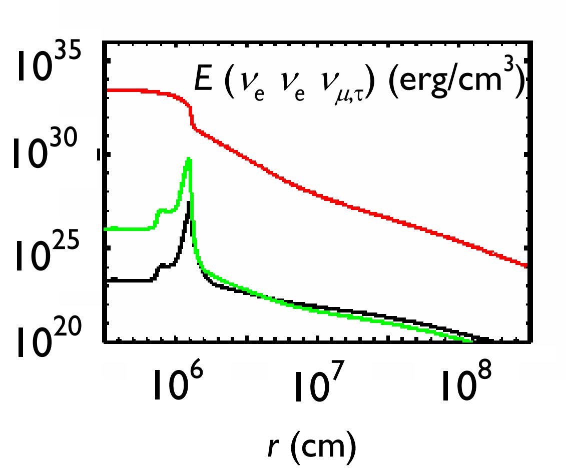
Provide automation for:

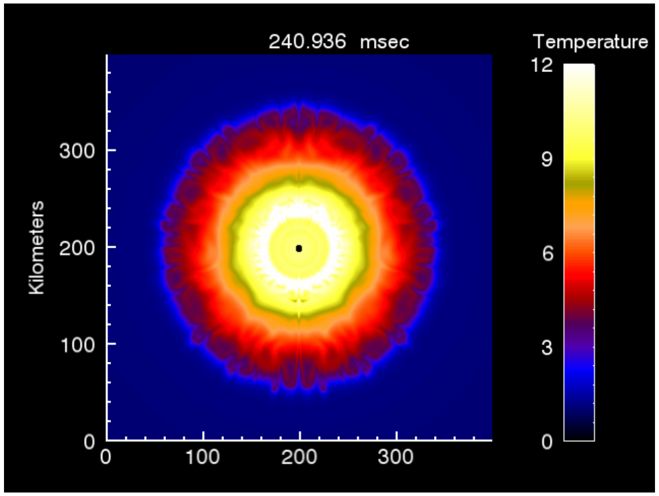
- job monitoring
- migration of files to
  - tertiary (long-term) storage
  - local sites for analysis
- zeroth-pass analysis
  - to confirm model validity and the value of continuing a simulation run (in near real time)
- first-pass analysis
  - to do “standard” analysis to identify areas where human intervention will pay off

Automated Zeroth-Pass Analysis

Working in pilot!

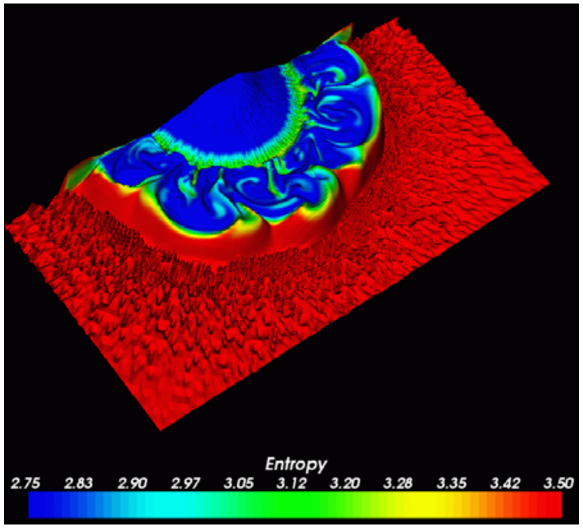
- ***Purpose:*** to confirm model validity and the value of continuing a simulation run (in near real time)
  - log files; line plots (*a la PGPLOT);* simple 2-D plots (*IDL*)



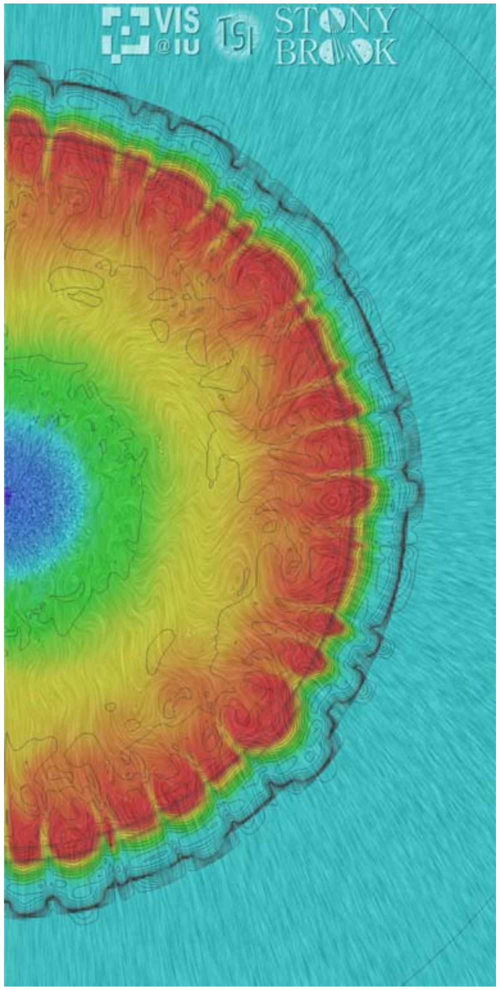


Automated First-Pass Analysis
Doable, but not done yet…

- ***Purpose:*** to do "standard" analysis to identify areas where human intervention will pay off
  - 2- and 3-D plots (*IDL* and *VTK*)
  - movies
  - feature recognition, data mining, *etc.*?

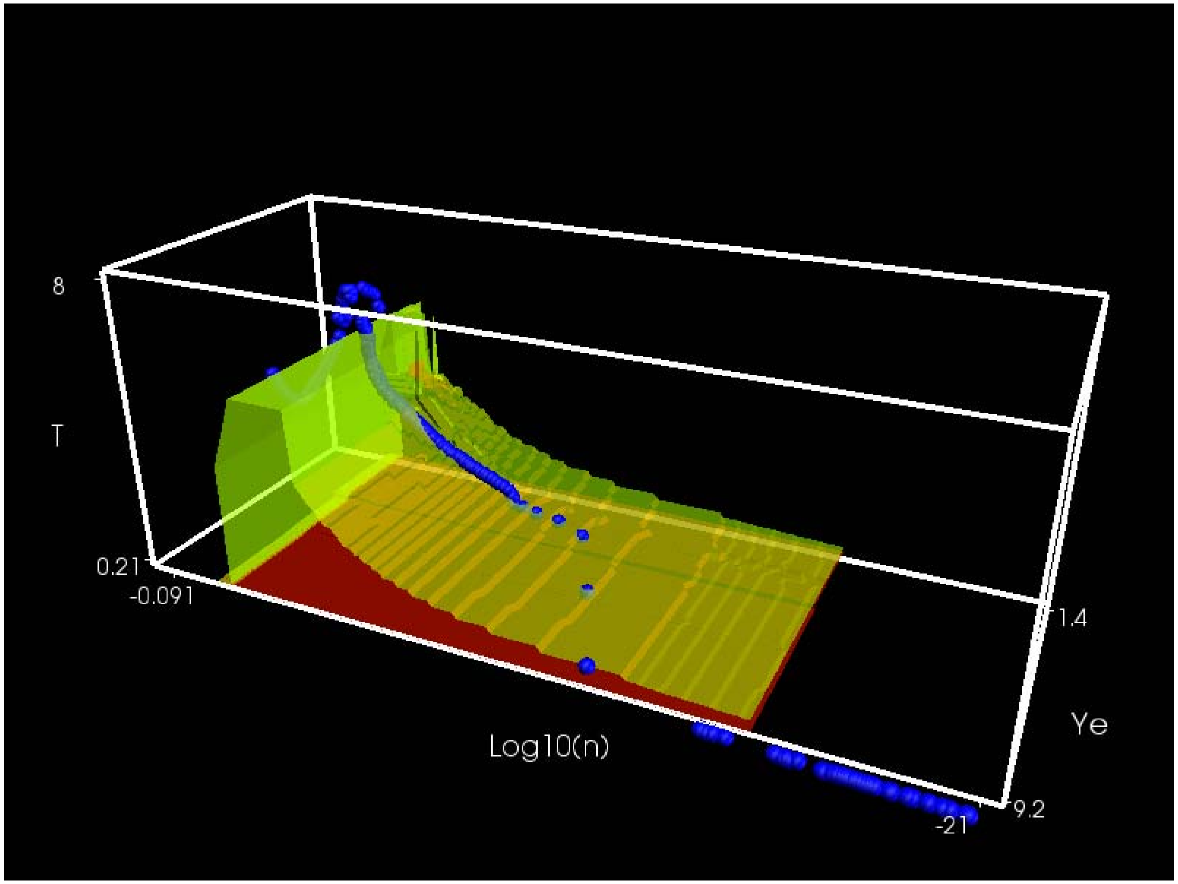


2- and 3-D imaging with contouring and LEA texturing



Automated First-Pass Analysis (cont'd)

VTK-based movies in parameter space



Our requirements for workflow management tools

- Has to be easy enough for us to learn and retain
  - must have short learning curve (for us, grad students, *etc.*)
  - helps to have good tutors!
- Must be modular
  - workflows change
  - what tasks we automate will change
  - the complexity we want to support will change
- Must be extensible without lots of consultation
- Needs to support data standards (HDF5, netCDF, MPI-IO)
- A usable command-line interface is desirable

Some observations so far regarding SPA

Initial experiences have been positive:

- Looks highly workable for our purposes
- Our experience suggests that it can be modified for diverse environments
- Perhaps commercializable…

Suggestions:

- Can it be written in python? (Java + X11 = slow)
- Would like workflows to be easier to maintain and modify remotely
- Perhaps more of a client/server architecture would help?
  - Don't want to bring up remote GUI (latency)
  - Automated workflow has its biggest value to the traveller where connectivity can be iffy

Variations on a theme

Q: *Where is it best do post-processing?*

A: For non-interactive processing, distance doesn't matter!

- We are experimenting with off-screen rendering with great success.
- Could *VTK* be built on the visualization systems at NERSC to support off-screen rendering?
  - could save a lot of time, bandwidth, and downstream disk space
- Can we think about the same thing for more advanced tools?
- Sophisticated post-processing may itself turn into a true HPC application.

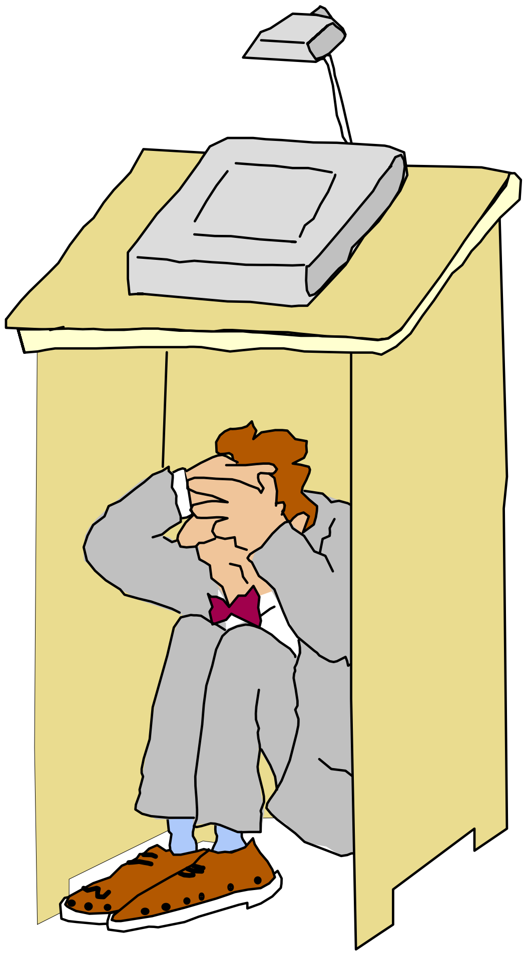
How best to work with us (and people like us)

- Small is beautiful
  - Monolithic toolsets are less likely to be used
  - Small component-based utilities are best

- Human contact is nice
  - Technology can't help without cooperation in its implementation

- Tutorials and examples for pinheads are great
  - Don't assume we know more than we do
  - Need to crawl before we walk, *etc*….

Questions?



emyra@mail.astro.sunysb.edu

http://www.astro.sunysb.edu/emyra