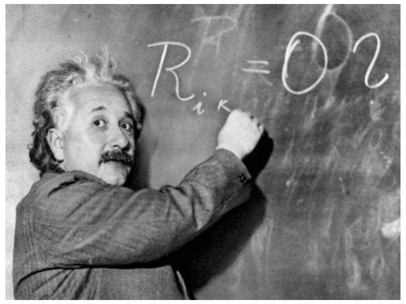
# Large Scale Problem Solving Using Automated Code Generation and Distributed Visualisation

SCALE 2009, Shanghai, China, May 2009

W. Benger, E. Bentivegna, P. Diener, J. Ge, A. Hutanu, R. Kooima, K. Liu, R. Paruchuri, E. Schnetter, J. Tao, C. Toole, G. Allen

LSU
CENTER FOR COMPUTATION & TECHNOLOGY

## PROGRAMMING PRODUCTIVITY
### Translating complex science into code



**Kranc**
Automated code generation through Mathematica

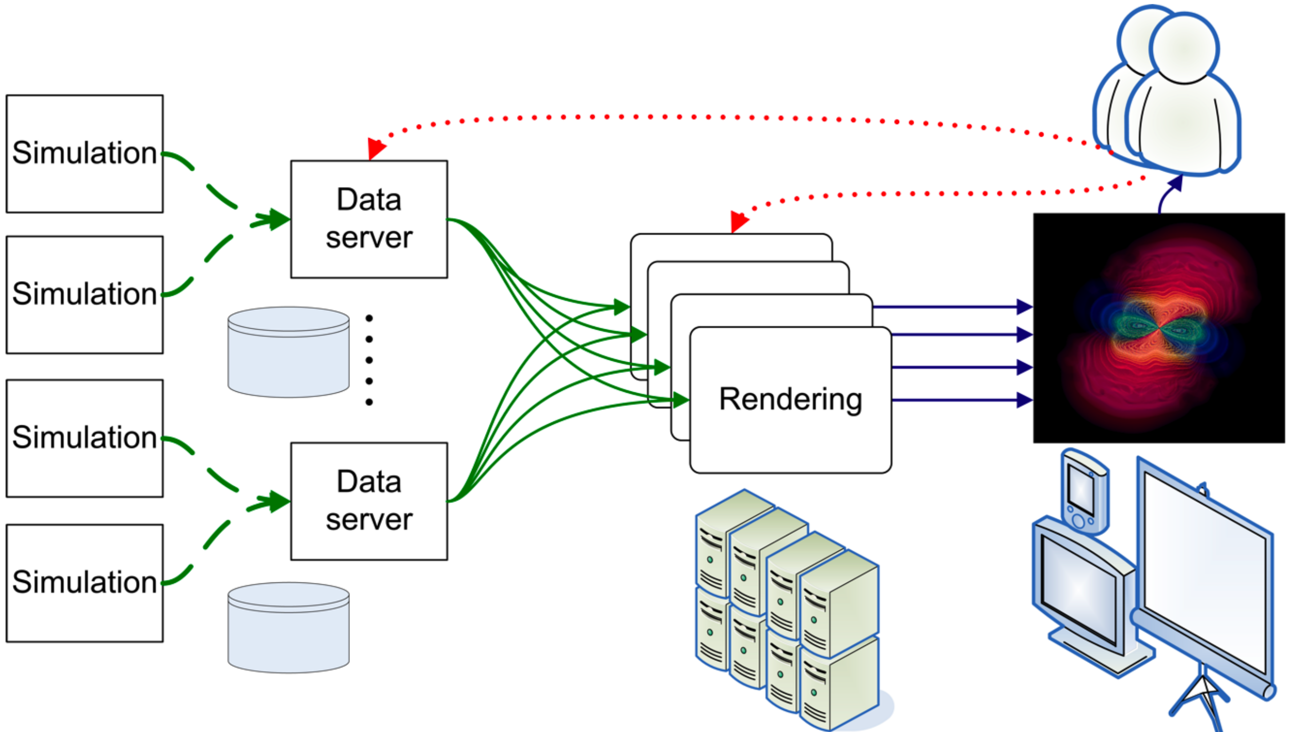
**Cactus**
Portable, modular, extensible scientific software framework

## SCALABILITY TO LARGE NUMBER OF PROCESSORS
### Efficient utilization of high-concurrency machines

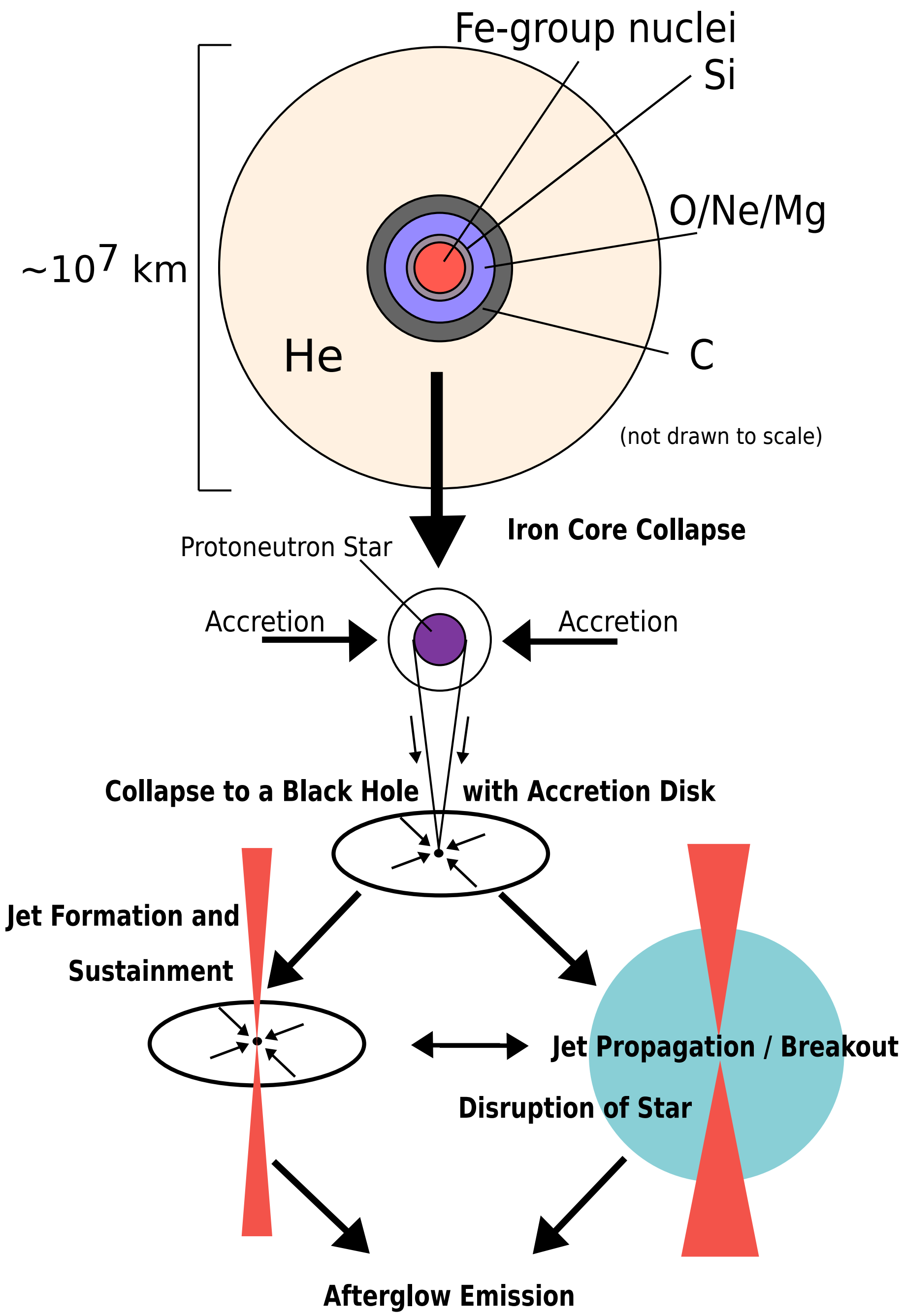
**Cactus**
framework

**Carpet**
Adaptive mesh refinement

## I/O BANDWIDTH AND INTERACTIVE VISUALIZATION
### Moving and analyzing large datasets



# Motivation:
# Gamma-Ray Burst Grand Challenge

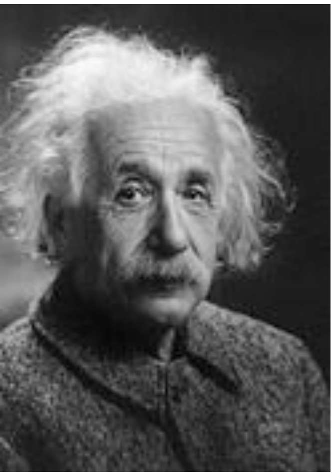



ACM, doi:10.1145/1341811.1341831

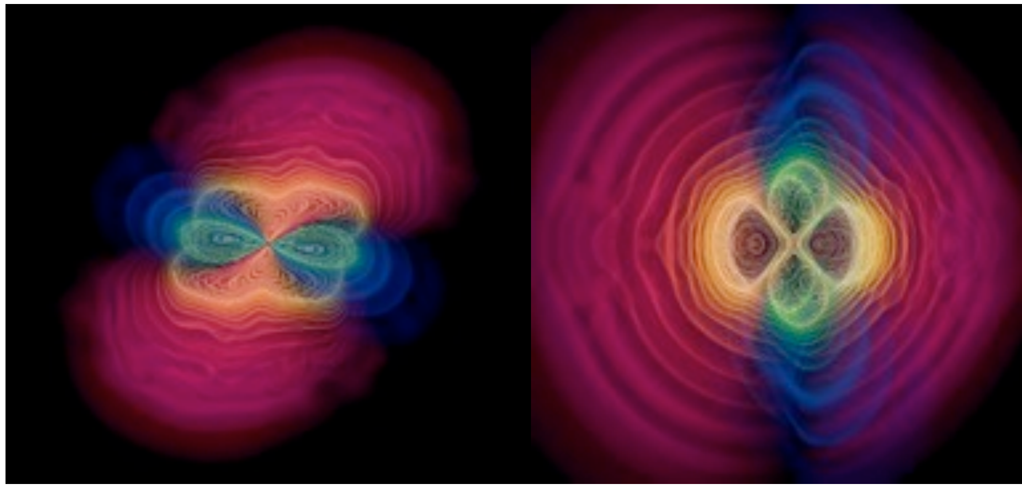
- Most energetic events in the universe
- Mechanism still a riddle; grand challenge in astrophysics
- Modelling requires expertise in many fields of physics (general relativity, magneto-hydrodynamics, neutrinos, ...)
- Requires petascale computing

CENTER FOR COMPUTATION & TECHNOLOGY

# McLachlan: Solving the Einstein Equations



- Einstein Equations: complex system of PDEs, almost impossible to code by hand
- Automated code generation with Kranc: creating complete, optimised Cactus modules
- Enables modifying the implementation without accidental errors
- Employing curvilinear coordinate systems, higher order discretisation methods, mesh refinement



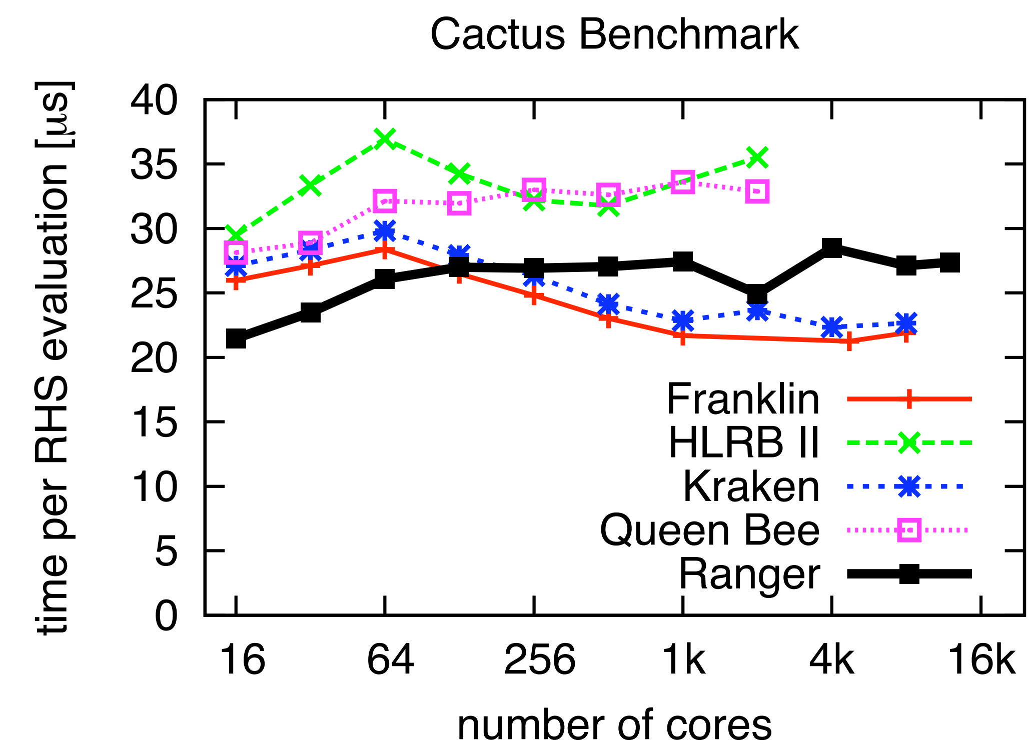
http://www.cct.lsu.edu/~eschnett/Kranc/
http://www.cct.lsu.edu/~eschnett/McLachlan/

LSU
CENTER FOR COMPUTATION & TECHNOLOGY

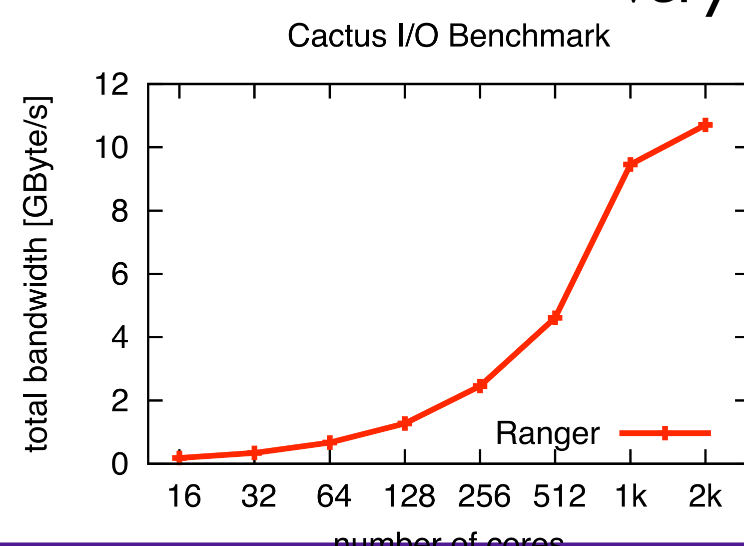
# Cactus/Carpet: Scalable Adaptive Mesh Refinement

- Cactus: Software framework for HPC
  - Collaborative code development
  - Modular mechanism for multi-physics coupling
- Carpet: Scalable Adaptive Mesh Refinement (AMR)
  - Hybrid parallelisation combining MPI and OpenMP
  - Efficient I/O layer

http://www.cactuscode.org
http://www.carpetcode.org



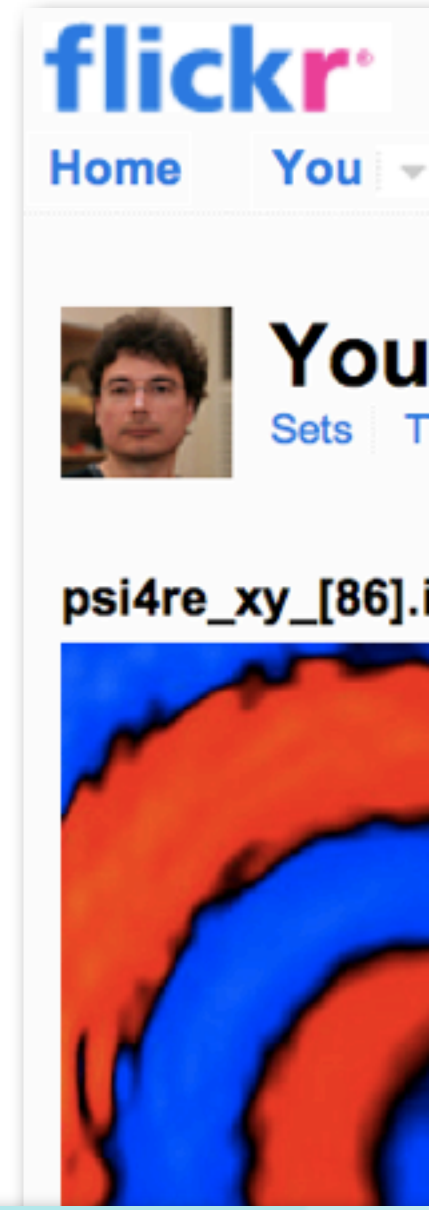

Weak scaling benchmark, 9 levels of mesh refinement, very good parallel scaling




LSU
CENTER FOR COMPUTATION & TECHNOLOGY

# Interacting with Scalable Simulations

- Typical large simulations are a **black box**
- However, need to interact with simulation:
  - examine current state
  - announce interesting events
- Debug physics, ensure good performance, detect errors early
- Web server built into simulation, announce to portals, Twitter, Flickr

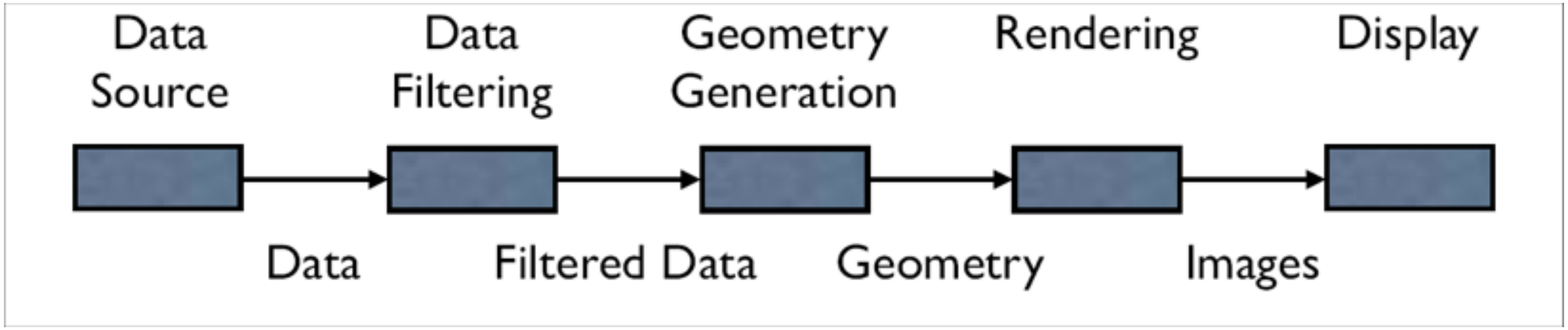



# Visualization Challenge

- Simulations generate large data (TBs->PBs)
  - –Visualization capabilities behind simulations
- Visualization system must be designed to be
  - –Interactive (>5 fps)
  - –Responsive (<2 second update)
  - –Handle large data (scale to deal with PB files)
  - –High resolution
  - –Able to enable collaborative visualization

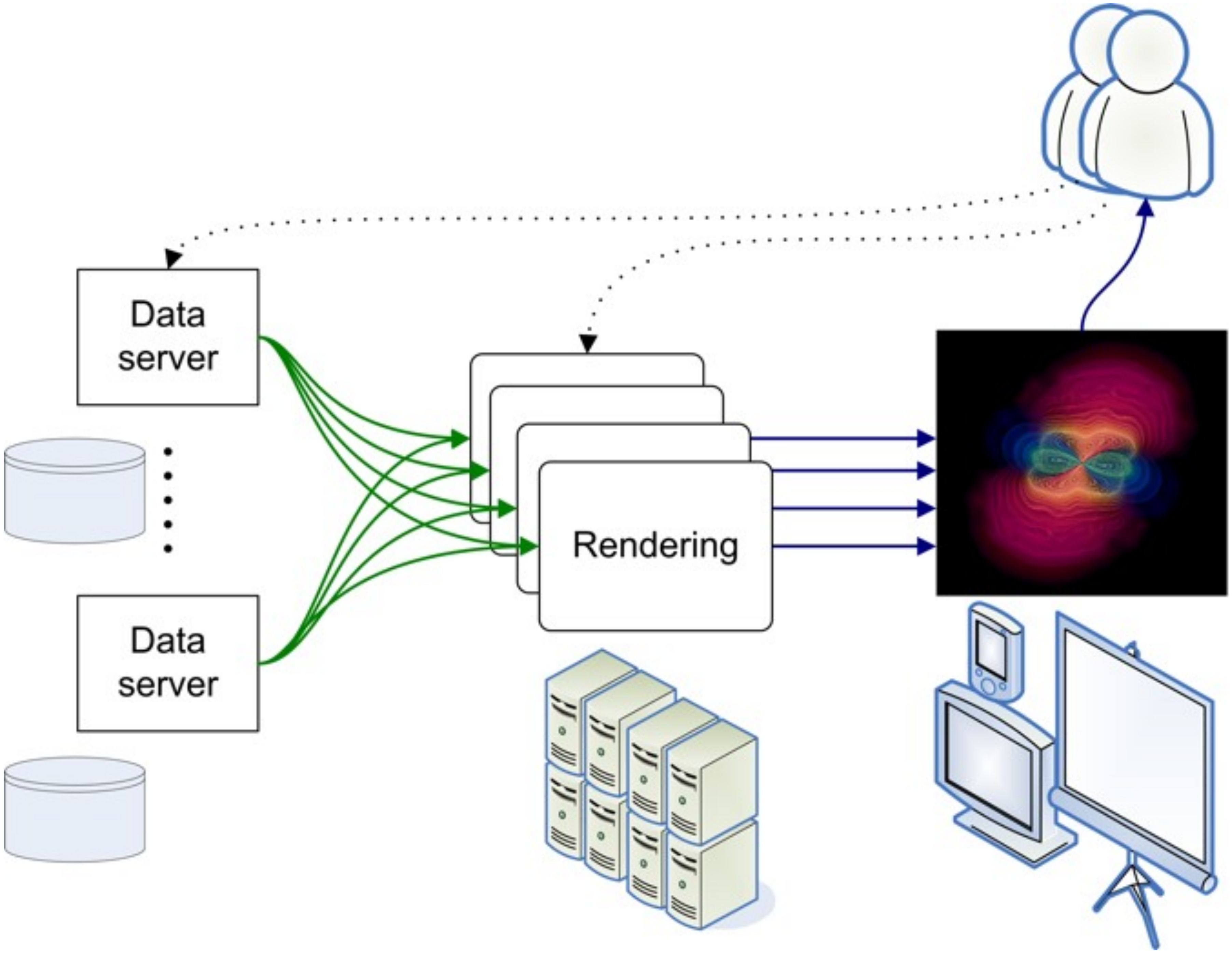
# Our Method

- Visualization pipeline




- Distributed resources to **improve** visualization application
- Integrated application development approach – optimize **all resources**

# Overall Architecture



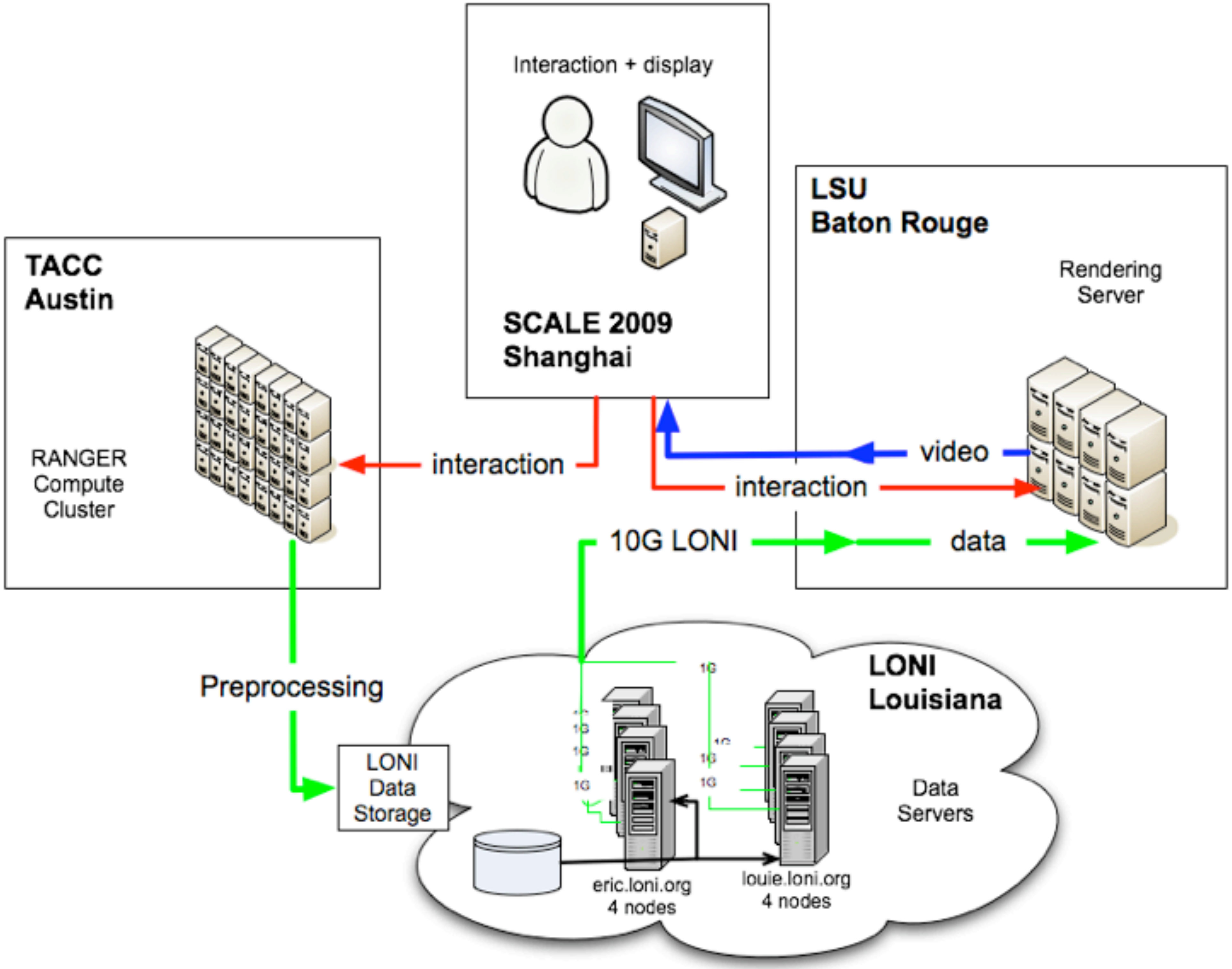

- New **Parallel** renderer
- New **Distributed** data server
- New interaction using **tangible devices**
- Using SAGE for streaming

CENTER FOR COMPUTATION & TECHNOLOGY

# Scalable, Fast I/O

- Disk I/O rate limited for loading/browsing through a large dataset
- Our approach –
  - *Use remote machines to store data in main memory*
  - *Transfer over 10G network faster than from local disks*
- Details of our distributed data server:
  - network transport protocols that support high transmission rate: UDT (UDP-based) library
  - new remote data access system that supports a high number of operations/second

# SCALE09 Demonstration




- 2048 core simulation code
- 8 node renderer at LSU
- Remote LONI nodes as data servers
- $1024^3$ spatial resolution (1GByte/ timestep)
- 20 timesteps cached remotely
- 2s/timestep remote load; 12.8s local load
- Scalable Approach!!!

LSU
CENTER FOR COMPUTATION
& TECHNOLOGY

# Acknowledgements


A. Clary (Comp. Eng.)
O. Korobkin (Physics)
H. Liu (LSU HPC)
T. Minyard (TACC)
J. Ucan (CCT)
A. Yates (CCT)
D. Waters (CCT)
S. White (LSU HPC)


XiRel

Alpaca


CyberTools

LSU
CENTER FOR COMPUTATION
& TECHNOLOGY

NSF

TeraGrid

TACC


LONI