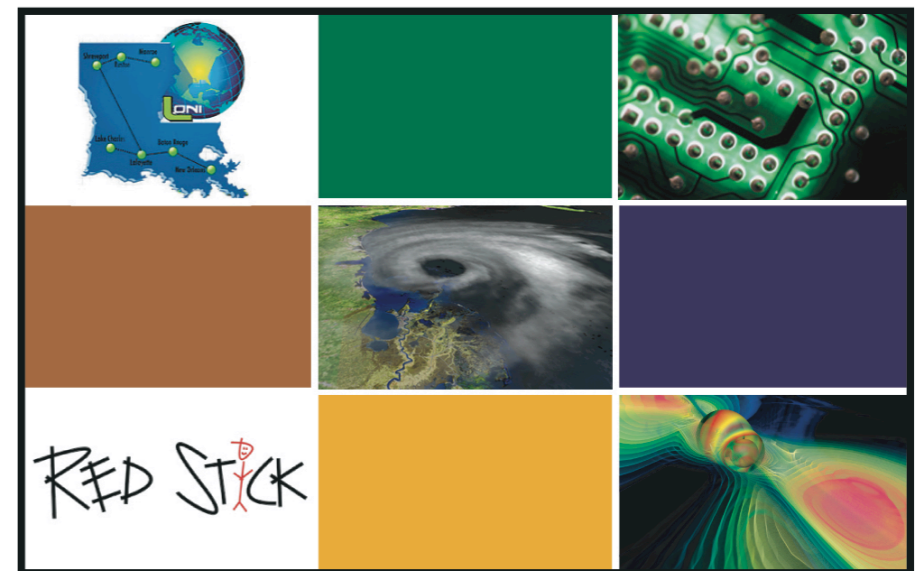


Cactus Tools for Petascale Computing

Erik Schnetter
Reno, November 2007



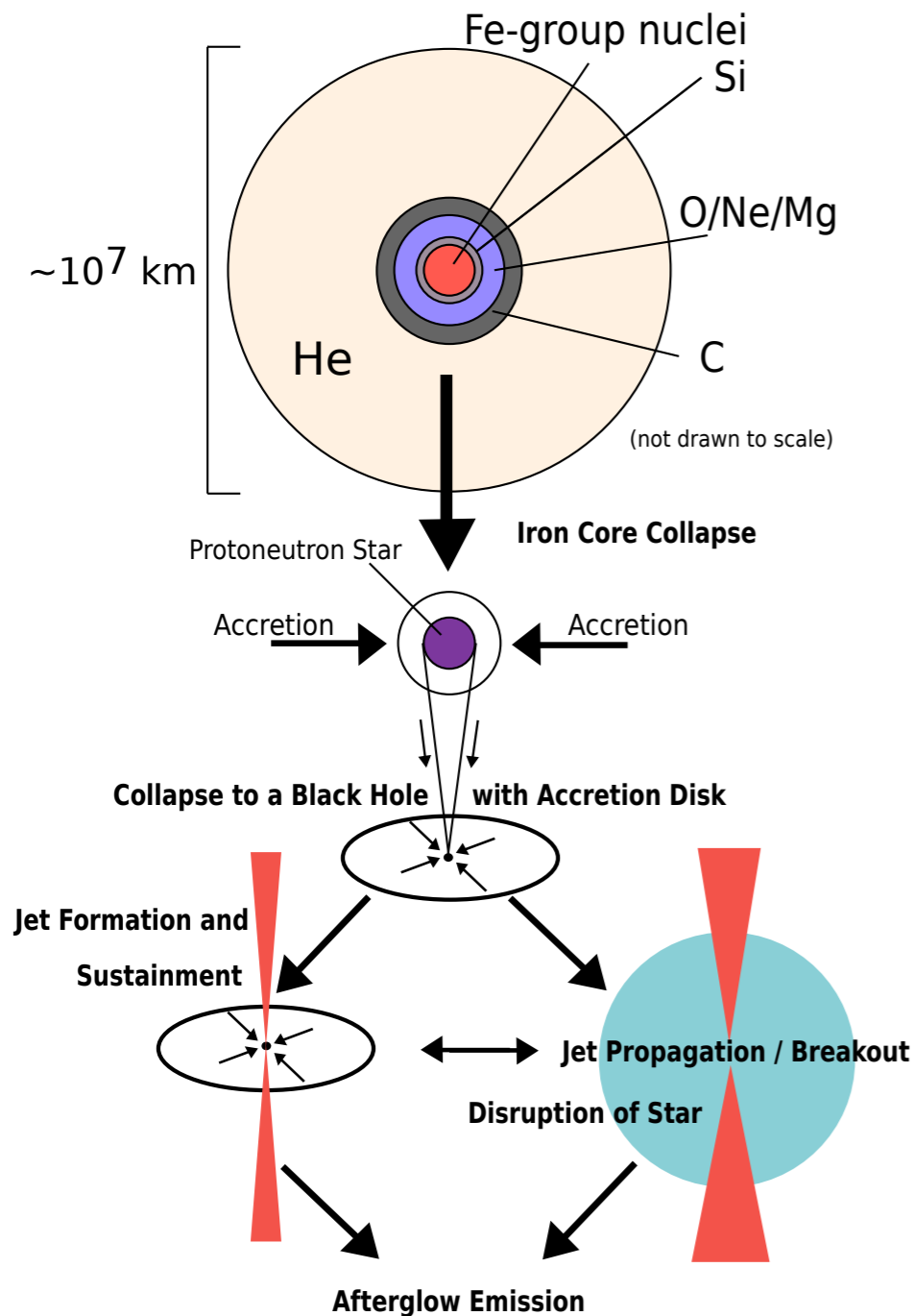
CENTER FOR COMPUTATION
& TECHNOLOGY



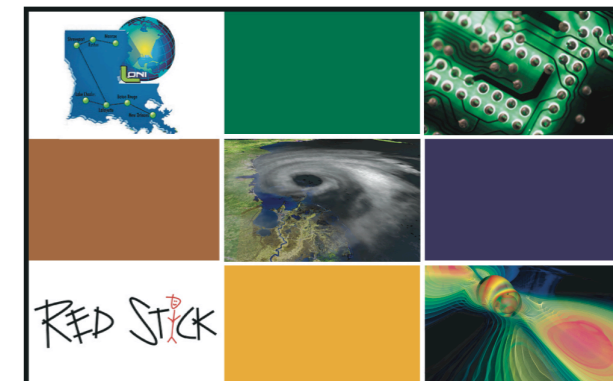


CENTER FOR COMPUTATION
& TECHNOLOGY

Gamma Ray Bursts



- Most energetic events known in universe
- Grand challenge in astrophysics; likely to be detected by LIGO in coming years
- Combines many fields of physics
- Requires (at least) petascale computing



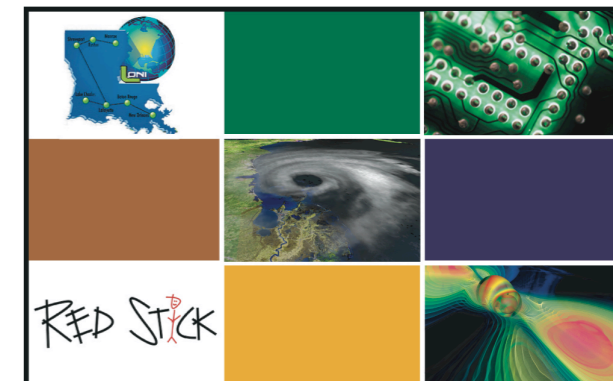


CENTER FOR COMPUTATION
& TECHNOLOGY

Petascale Needs of Gamma Ray Bursts

Conservative total estimates [See Ott et al., Mardi Gras conference, Baton Rouge 2008]:

- Large scale differences: ~25 m near central black hole, ~1000 km overall domain
- 3 TByte memory (with 10 levels of AMR)
- 300 million time steps
- in total 6 million Pflop (6 Zflop)
- ~70 days at sustained 1 Pflop/s





CENTER FOR COMPUTATION
& TECHNOLOGY

Addressing Petascale Challenges

1. Expect ~1 M CPUs, need everything parallel (Amdahl): use performance modelling to improve codes
2. More cores/node tighten memory bottleneck: use dynamic, adaptive cache optimisations
3. Probably less memory/processor than today: use hybrid schemes (MPI + OpenMP) to reduce overhead
4. Hardware failures “guaranteed”: use fault tolerant infrastructure

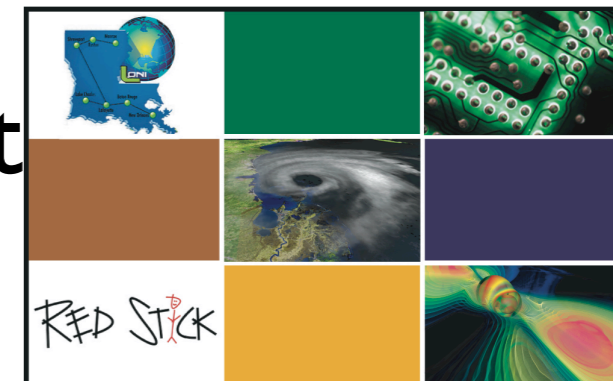




CENTER FOR COMPUTATION
& TECHNOLOGY

Cactus

- Framework for HPC: code development, simulation control, visualisation
- Manage increased complexity with higher level abstractions, e.g. for inter-node communication, intra-node parallelisation
- Active user community, 10+ years old
- Supports collaborative development





CENTER FOR COMPUTATION
& TECHNOLOGY

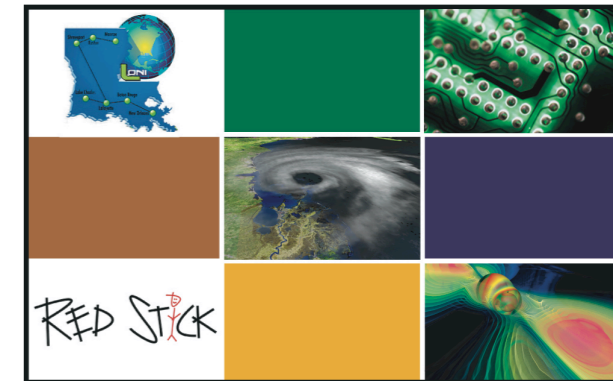
Cactus Framework

parallelism
 memory management
 I/O
 SOR solver
your computational tools
 multigrid
 interpolation
 reduction

extensible APIs
 ANSI C
 parameters
 schedule
 grid variables
 make system
 error handling

coordinates
 boundary conditions
 AMR
 CFD
 wave equation
 Einstein equations
your physics
 remote steering

Core *flesh* with plug-in *thorns*

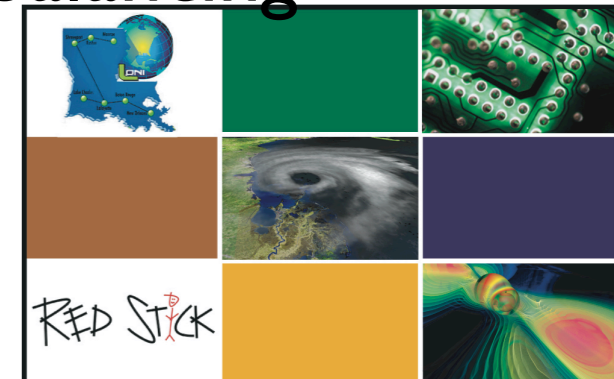




CENTER FOR COMPUTATION
& TECHNOLOGY

I. Performance Modelling

- Need tools to help make applications scale
- Developing on 1 M processors is expensive -- need to run on 10 k processors, then model
- Separate physics and infrastructure components; model them separately
- Cactus can automatically insert calliper points, with high-level knowledge about application
- Run-time profiling of computational and science thorns enables dynamic load balancing

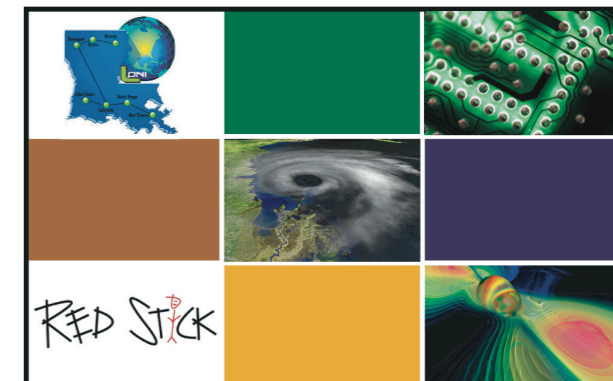




CENTER FOR COMPUTATION
& TECHNOLOGY

2. Dynamic and Adaptive Optimisation Methods

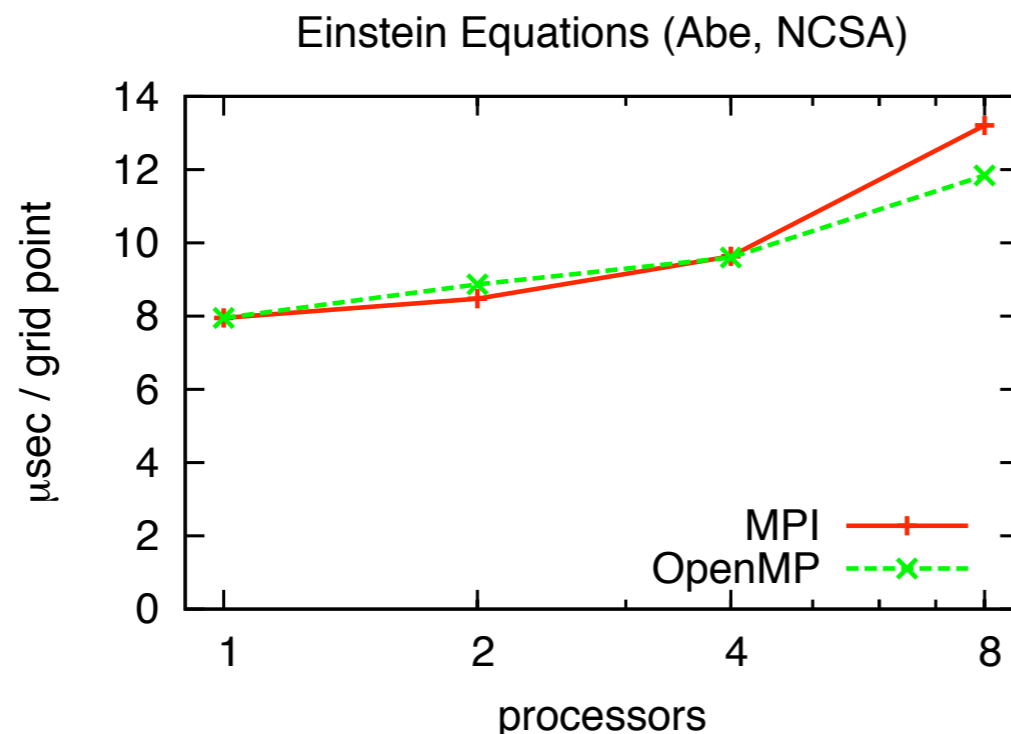
- Must choose parameters for OpenMP parallelisation, loop tiling, etc. very carefully
- Off-line parameter searches are expensive and only test test cases
- Better: monitor behaviour and update parameters at run time
- Adapts to changes in simulation (grid sizes, different physics) and environment (hardware, compiler)
- Offer safe and simple API to application programmers





3. Hybrid Communication Models

- Use MPI between nodes, OpenMP within nodes
- Reduce “MPI” memory overhead
- Common address space enables more cache optimisations
- Cactus framework offers abstraction layer for parallelisation: basic OpenMP features work as black box

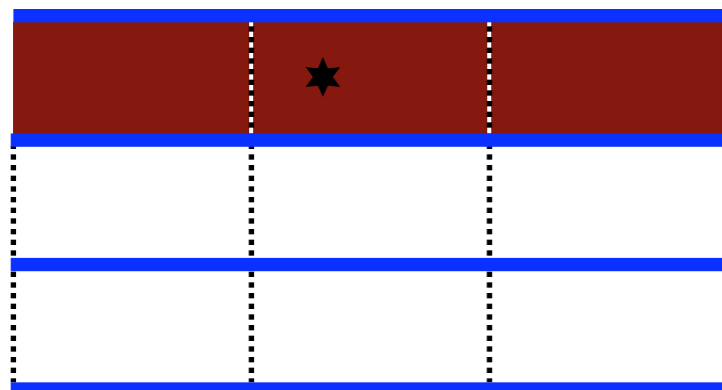




4. Fault Tolerance

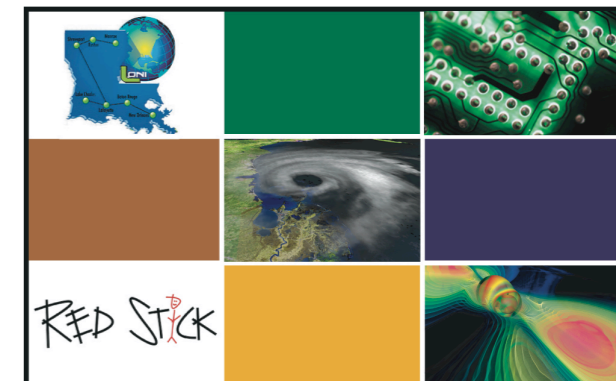
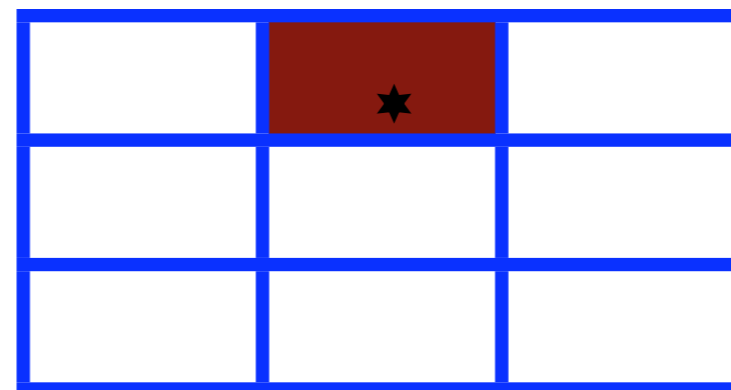
- Need checkpointing/recovery on steroids, need to cope with partial failure
- Be able to change number of active nodes
- Example: keep log of inter-processor messages, so that a lost node can be replaced
- Contain failure, continue simulation

Regular checkpointing



↑
time

“Cubicle” checkpointing





CENTER FOR COMPUTATION
& TECHNOLOGY

XiRel: Improve Computational Infrastructure

- Sponsored by NSF PIF; collaboration between LSU/PSU/RIT/AEI
- Enhance and create new physics infrastructure for numerical relativity
- Improve mesh refinement capabilities in Cactus, based on Carpet
- Develop common data and metadata management methods, with numrel as driver application
- Prepare numerical relativity codes for petascale architectures

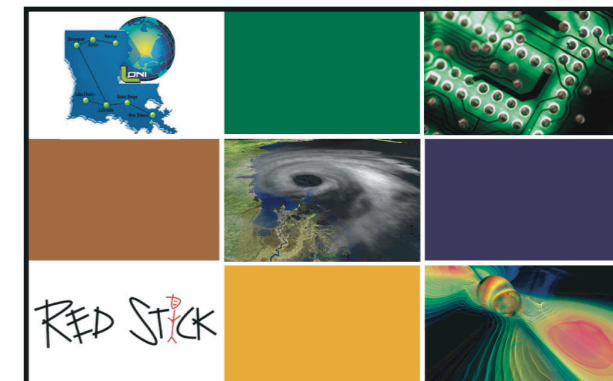




CENTER FOR COMPUTATION
& TECHNOLOGY

Application-Level Debugging and Profiling

- Sponsored by NSF SDCI
- As framework, Cactus has complete overview over programme and execution schedule
- Need to debug simulation at level of interacting components, in production situations, at scale
- Grid function declarations have rich semantics -- use this for visual debugging
- Combine profiling information with execution schedule, place calliper points automatically





CENTER FOR COMPUTATION
& TECHNOLOGY

Cactus, Eclipse, Blue Waters

Source code

cvcs/svn
edit
compile
debug

Performance data

gather
process
display

Online databases

Configuration files
Performance data

Simulations

submit
monitor
steer

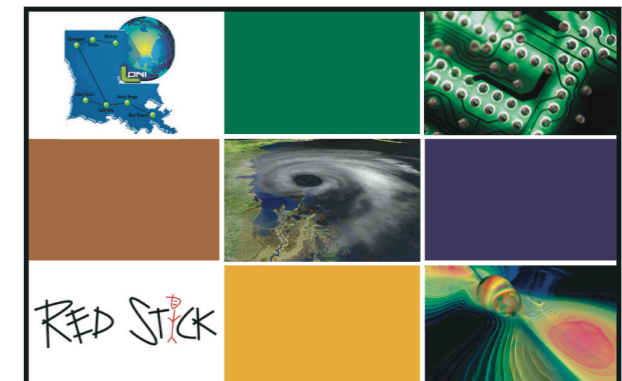
local
remote





CENTER FOR COMPUTATION
& TECHNOLOGY

Queen Bee

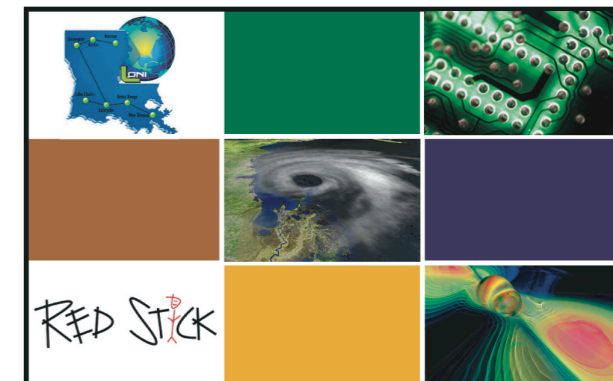




CENTER FOR COMPUTATION
& TECHNOLOGY

Queen Bee

- 50.4 GFlop/s peak, 34.8 GFlop/s Linpack
- 668 nodes, 8 cores/node
- 2.33 GHz dual quad-core CPUs
- 8 GByte memory per node
- Infiniband interconnect

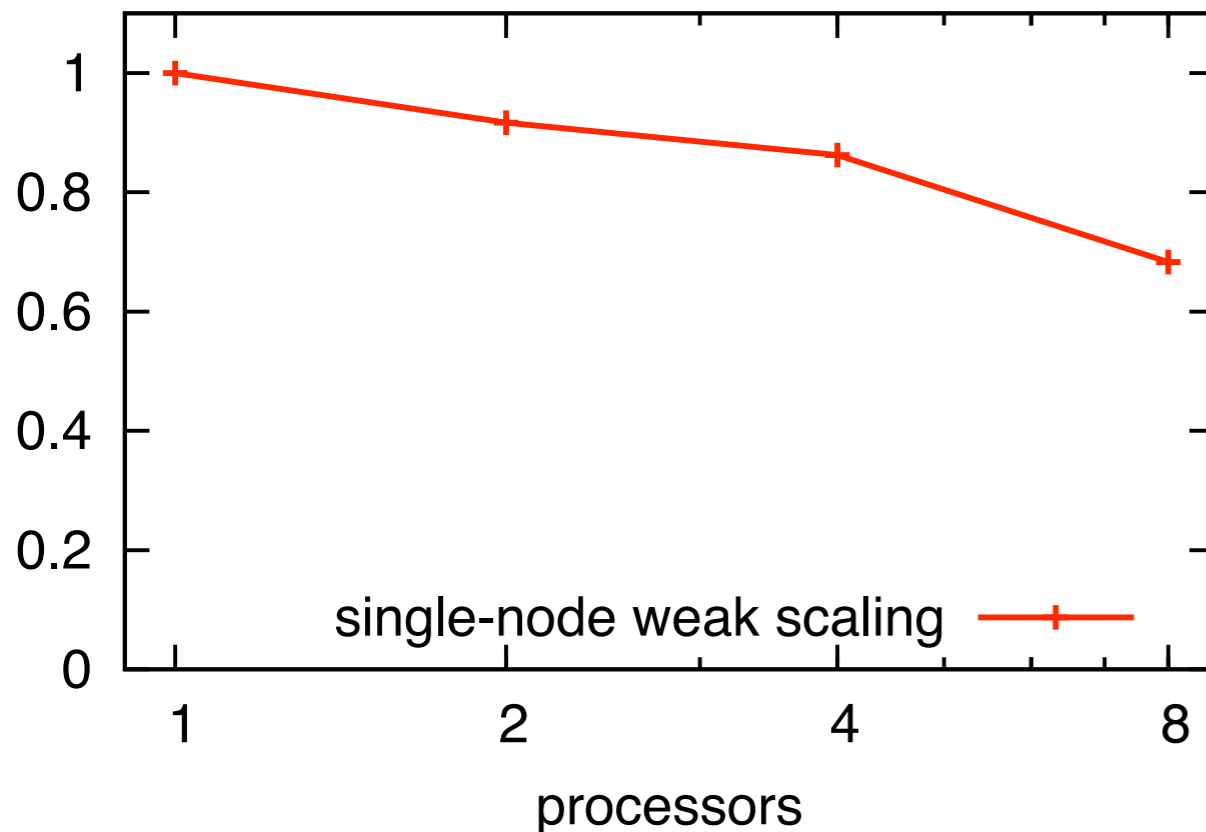




CENTER FOR COMPUTATION
& TECHNOLOGY

Single-Node Scaling

Einstein Equations (Queen Bee, LONI)



Parallelisation with OpenMP

- Full Einstein equations, 65^3 grid points per processor
- Scaling limited by cache performance
- 8th core still increases performance (but not linearly)
- Need advanced, dynamic cache optimisations



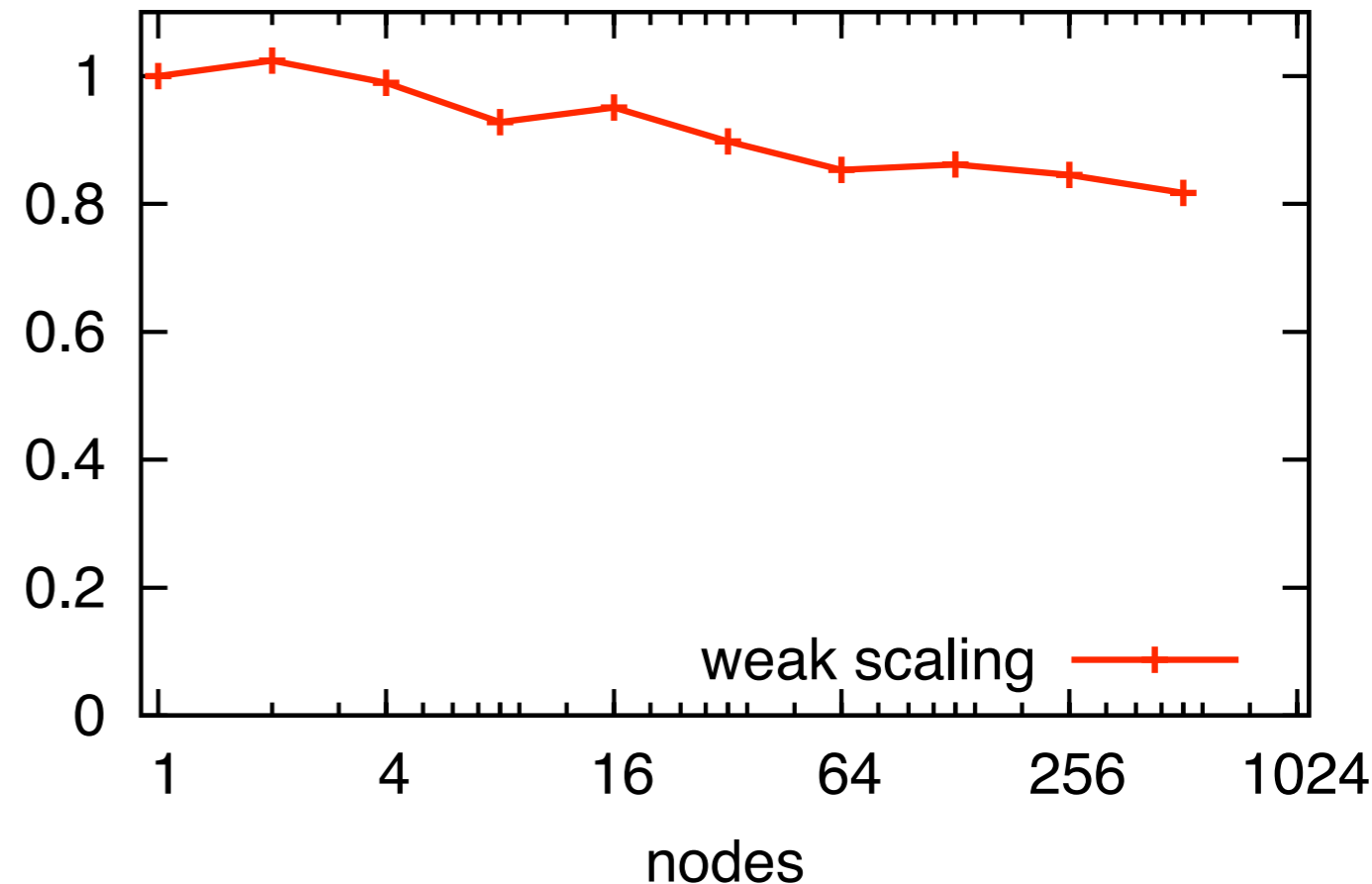


CENTER FOR COMPUTATION
& TECHNOLOGY

Queen Bee Scaling

- Full Einstein equations, 65^3 grid points per processor
- Hybrid scheme combining MPI + OpenMP, reducing parallelisation overhead
- Code machine-generated, including some cache optimisations, potential for more advanced transformations

Einstein Equations (Queen Bee, LONI)



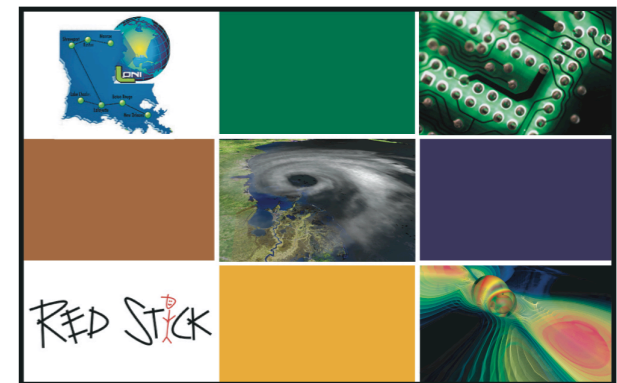
512 nodes = 4096 cores





CENTER FOR COMPUTATION
& TECHNOLOGY

Queen Bee Interactive Demo





CENTER FOR COMPUTATION
& TECHNOLOGY

Questions? Comments?

