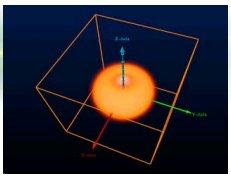
Cactus Grid Computing

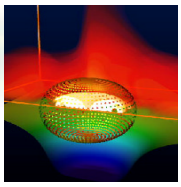
Gabrielle Allen

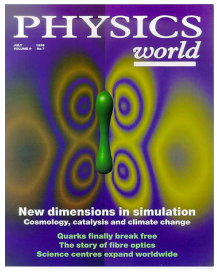
Max Planck Institute for Gravitational Physics,
(Albert Einstein Institute)

www.CactusCode.org

Cactus

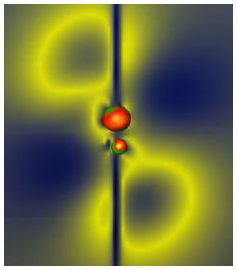


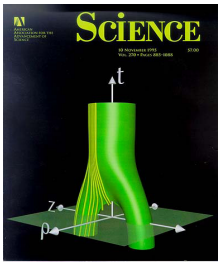


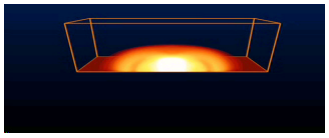


CACTUS is a freely available, modular, portable and manageable environment for collaboratively developing parallel, high-performance multi-dimensional simulations

THE GRID: Dependable, consistent, pervasive access to high-end resources





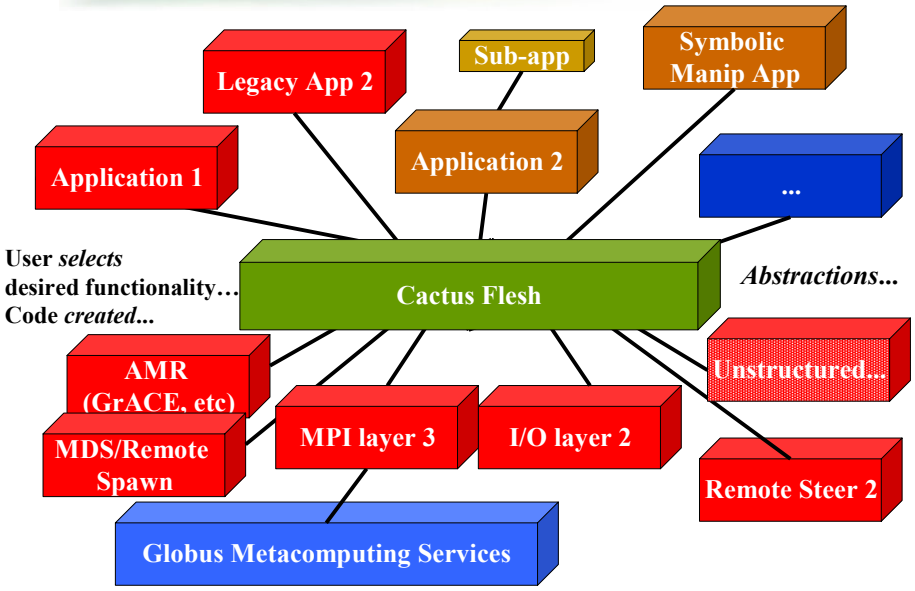


www.CactusCode.org

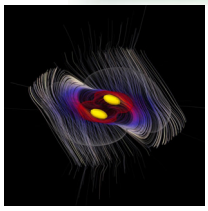
What is Cactus?

- Flesh (ANSI C) provides code infrastructure (parameter, variable, scheduling databases, error handling, APIs, make, parameter parsing)
- Thorns (F77/F90/C/C++/[Java/Perl/Python]) are **plug-in** and **swappable** modules or collections of subroutines providing both the computational instructructure and the physical application. Well-defined interface through 3 config files
- Just about anything can be implemented as a thorn: Driver layer (MPI, PVM, SHMEM, …), Black Hole evolvers, elliptic solvers, reduction operators, interpolators, web servers, grid tools, IO, …
- **User driven**: easy parallelism, no new paradigms, flexible
- Collaborative: thorns borrow concepts from OOP, thorns can be shared, lots of collaborative tools
- Computational Toolkit: existing thorns for (Parallel) IO, elliptic, MPI unigrid driver,
- Integrate other common packages and tools: HDF5, Globus, PETSc, PAPI, Panda, FlexIO, GrACE, Autopilot, LCAVision, OpenDX, Amira, ...
- Trivially Grid enabled!

Modularity of Cactus...

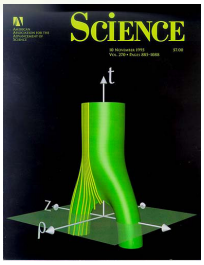


Motivation: Grand Challenge Simulations



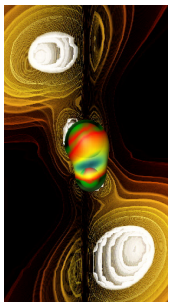
NASA Neutron Star Grand Challenge

- 5 US Institutions
- Towards colliding neutron stars



NSF Black Hole Grand Challenge

- 8 US Institutions, 5 years
- Towards colliding black holes



EU Network Astrophysics

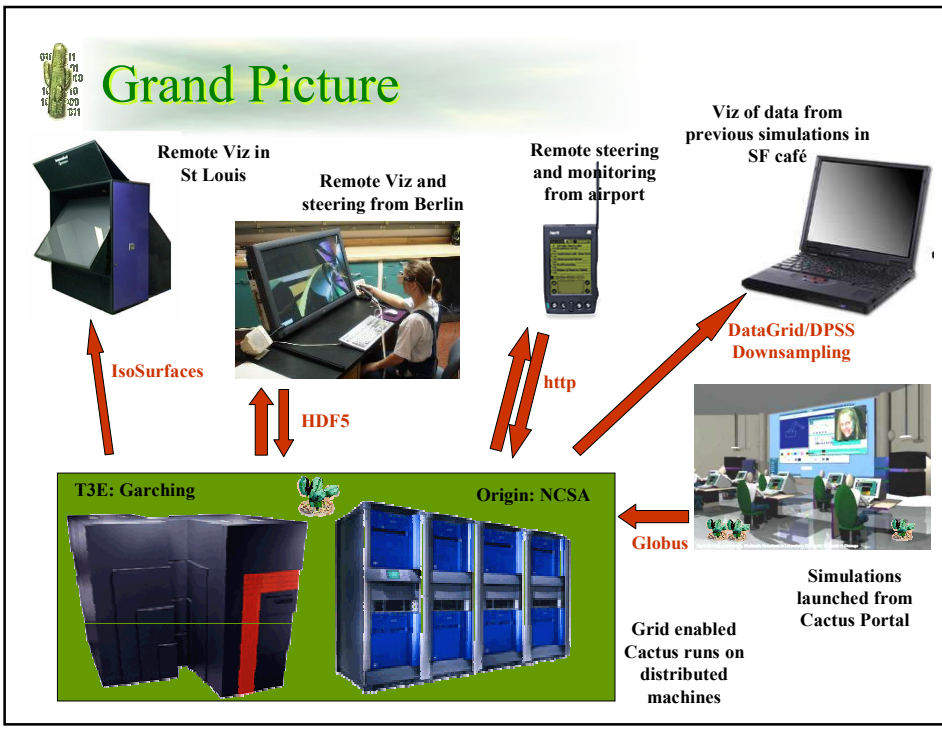
- 10 EU Institutions, 3 years
- Try to finish these problems …
- *Entire Community becoming Grid enabled*

Examples of Future of Science & Engineering

- Require Large Scale Simulations, beyond reach of any single machine
- Require Large Geo-Distributed Cross-Disciplinary Collaborations
- Require Grid Technologies, but not yet using them!
- Both Apps and Grids Dynamic…

Why Grid Computing?

- AEI Numerical Relativity Group has access to high-end resources in over ten centers in Europe/USA
- They want:
  - Bigger simulations, more simulations and faster throughput
  - Intuitive IO at local workstation
  - No new systems/techniques to master!!
- How to make best use of these resources?
  - Provide easier access … no one can remember ten usernames, passwords, batch systems, file systems, … great start!!!
  - Combine resources for larger productions runs (more resolution badly needed!)
  - Dynamic scenarios … automatically use what is available
  - Remote/collaborative visualization, steering, monitoring
- Many other motivations for Grid computing ...

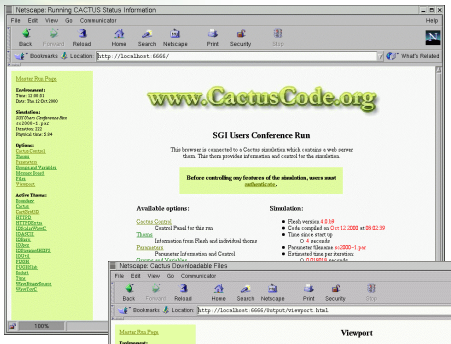


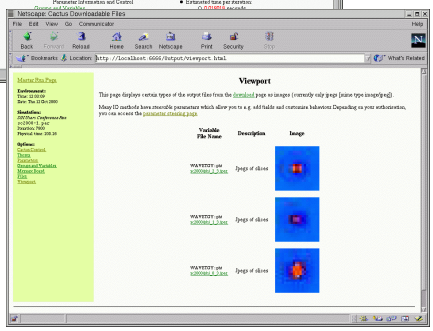
Cactus Grid Projects:

- User Portal (KDI Astrophysics Simulation Collaboratory)
  - Efficient, easy, access to resources … interfaces to everything else
- Collaborative Working Methods (KDI ASC)
- Large Scale Distributed Computing (Globus)
  - Only way to get the kind of resolution we really need
- Remote Monitoring (TiKSL/GriKSL)
  - Direct access to simulation from anywhere
- Remote Visualization (Live/Offline) (TiKSL/GriKSL)
  - Collaborative analysis during simulations/Viz of large datasets
- Remote Steering (TiKSL/GriKSL)
  - Live collaborative interaction with simulation (eg IO/Analysis)
- Dynamic, Adaptive Scenarios (GridLab/GrADs)
  - Simulation adapts to changing Grid environment
- **Make Grid Computing useable/accessible for application users !!**
  - GridLab: Grid Application Toolkit

Remote Monitoring/Steering: Thorn HTTPD

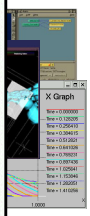
- Thorn which allows simulation any to act as its own web server
- Connect to simulation from any browser anywhere … **collaborate**
- Monitor run: parameters, basic visualization, ...
- Change *steerable* parameters
- See running example at www.CactusCode.org
- Wireless remote viz, monitoring and steering





Developments

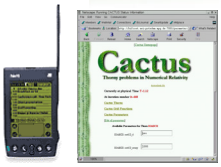
- VizLa… …)
  autor… …ht
  (exte…

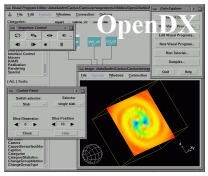


File Name	Variable	Description
output/phi_xy_[16].asc	WAVETOY::phi	Two-dimensional slice plots
output/phi_xz_[16].asc	WAVETOY::phi	Two-dimensional slice plots
output/phi_yz_[16].asc	WAVETOY::phi	Two-dimensional slice plots
output/phi_3D_diagonal.xg	WAVETOY::phi	One-dimensional line plots
output/phi_z_[16][16].xg	WAVETOY::phi	One-dimensional line plots
output/phi_y_[16][16].xg	WAVETOY::phi	One-dimensional line plots
output/phi_x_[16][16].xg	WAVETOY::phi	One-dimensional line plots
output/phi.h5	WAVETOY::phi	Full-dimensional variable contents
output/phi_norm2.xg	WAVETOY::phi	Reduction on Grid Functions

- Debu… …rovide
  their …
- Timin… …ws
  users to steer their simulation for better performance
  (switch of analysis/IO)

Remote Visualization

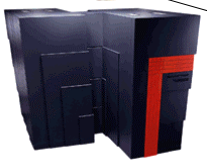


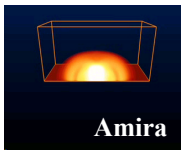


IsoSurfaces and Geodesics

All Remote Files VizLauncher (download)

Grid Functions Streaming HDF5 (downsampling to match bandwidth)





Use variety of local clients to view remote simulation data.
Collaborative, colleagues can access from anywhere.
Now adding matching of data to network characteristics

Remote Steering

Any Viz Client

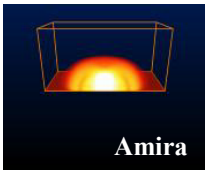
XML

HTTP

Remote Viz data

HDF5

Remote Viz data



Remote File Access

Viz in Berlin

Only what is needed

4TB at NCSA

Viz Client (Amira)
HDF5 VFD
DataGrid (Globus)
DPSS | FTP | HTTP

Visualization Client

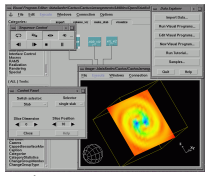
Downsampling, hyperslabs

DPSS Server | FTP Server | Web Server

Remote Data Server

Remote File Access

Chandos Hall (UK)



Visualization

HDF5 VFD/
GridFTP:
Clients use file URL
(downsampling,hyperslabbing)

Lapse for r<1, every other point

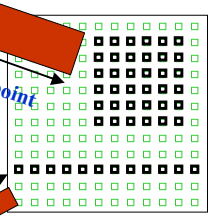
More Bandwidth Available

...work
...onitoring
Service

grr.psi on timestep 2

AEI (Germany)

Analysis of Simulation Data

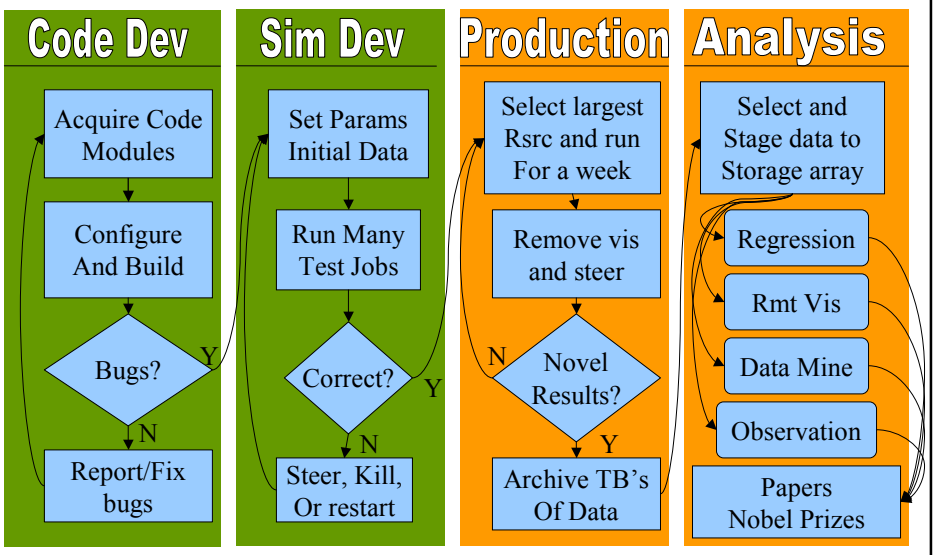


TB Data File

Remote Data Storage

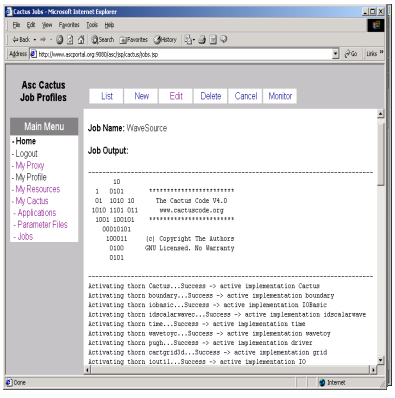
NCSA (USA)

Computational Physics: Complex Workflow



Cactus/ASC Portal

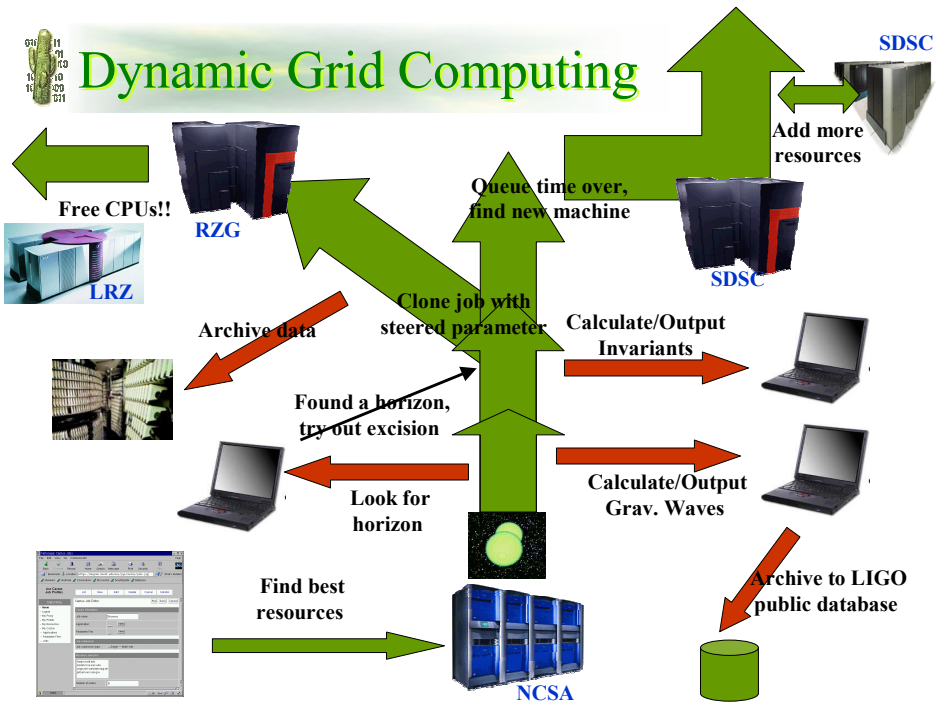
- KDI ASC Project (Argonne, NCSA, AEI, LBL, WashU)
- Technology: Web Based (end user requirement)
  - Globus, GSI, DHTML, Java CoG, MyProxy, GPDK, TomCat, Stronghold/Apache, SQL/RDBMS
- Portal should hide/simplify the Grid for users
  - Single access, locates resources, builds/finds executables, central management of parameter files/job output, submit jobs to local batch queues, tracks active jobs. Submission/management of distributed runs
- Accesses the ASC Grid Testbed



Grid Applications: Some Examples

- Dynamic Staging: move to faster/cheaper/bigger machine
  - "Cactus Worm"
- Multiple Universe
  - create clone to investigate steered parameter
- Automatic Convergence Testing
  - from intitial data or initiated during simulation
- Look Ahead
  - spawn off and run coarser resolution to predict likely future
- Spawn Independent/Asynchronous Tasks
  - send to cheaper machine, main simulation carries on
- Thorn Profiling
  - best machine/queue, choose resolution parameters based on queue
- Dynamic Load Balancing
  - inhomogeneous loads, multiple grids
- Intelligent Parameter Surveys
  - farm out to different machines
- …Must get application community to rethink algorithms…

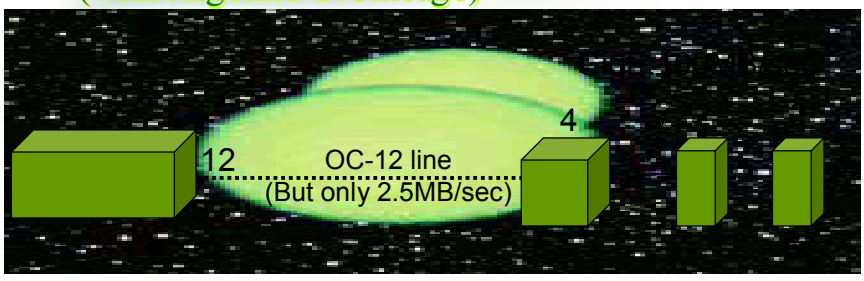
Dynamic Grid Computing

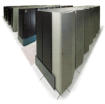


New Paradigms

- Dynamic Distributed apps with Grid-threads (gthreads)
- Code should be aware of its environment
  - What resources are out there NOW, and what is their current state?
  - What is my allocation?
  - What is the bandwidth/latency between sites?
- Code should be able to make decisions on its own
  - A slow part of my simulation can run asynchronously…spawn it off!
  - New, more powerful resources just became available…migrate there!
  - Machine went down…reconfigure and recover!
  - Need more memory…get it by adding more machines!
- Code should be able to publish this information to Portal for tracking, monitoring, steering…
  - Unexpected event…notify users!
  - Collaborators from around the world all connect, examine simulation.
- Two protypical examples:
  - Dynamic, Adaptive Distributed Computing
  - Cactus Worm: Intelligent Simulation Migration

Dynamic Adaptive Distributed Computation (with Argonne/U.Chicago)





SDSC IBM SP
1024 procs
5x12x17 =1020

NCSA Origin Array
256+128+128
5x12x(4+2+2) =480

This experiment:

- Einstein Equations (but could be any Cactus application)

Achieved:

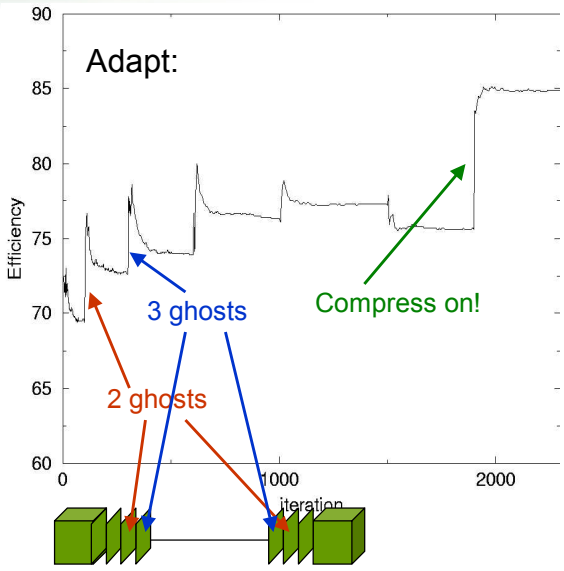
- First runs: 15% scaling
- With new techniques: 70-85% scaling, ~ 250GF

Distributing Computing

- Why do this?
  - Capability: Need larger machine memory than a single machine has
  - Throughput: For smaller jobs, can still be quicker than queues
- Technology
  - Globus GRAM for job submission/authentification
  - MPICH-G2 for communications (Native MPI/TCP)
  - Cactus simply compiled with MPICH-G2 implementation of MPI
    - gmake cactus MPI=globus
- New Cactus Communication Technologies
  - Overlap communication/communications
  - Simulation dynamically adapts to WAN network
    - Compression/Buffer size for communication
    - Extra ghostzones, communicate across WAN every N timesteps
  - Available generically: all applications/grid topologies

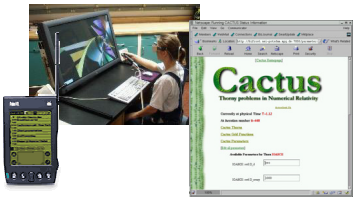
Dynamic Adaptation

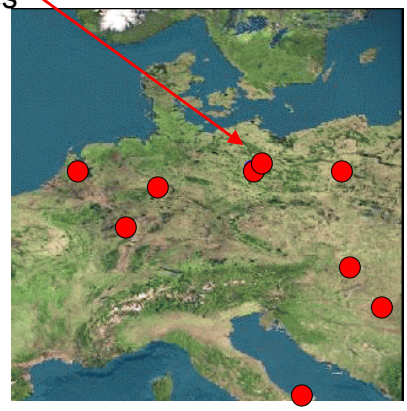
- Automatically adapt to bandwidth latency issues
- Application has NO KNOWLEDGE of machines(s) it is on, networks, etc
- Adaptive techniques make NO assumptions about network
- Issues:
  - More intellegent adaption algorithm
  - Eg if network conditions change faster than adaption…
- Next: Real BH hole run across Linux Clusters for high quality data for viz.



Cactus Worm: Basic Scenario
Live Demo at http://www.cactuscode.org

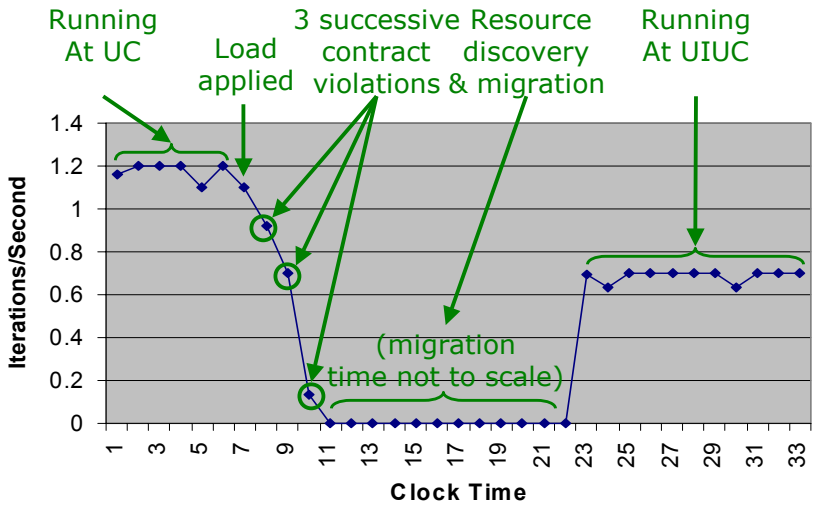
- Cactus simulation starts
- Queries a Grid Information Server, finds resources
- Makes intelligent decision to move
- Locates new resource & migrates
- Registers new location to GIS





- Continues around Europe…
- *Basic prototypical example of many things we want to do!*

Migration due to Contract Violation
(Foster, Angulo, Cactus Team…)

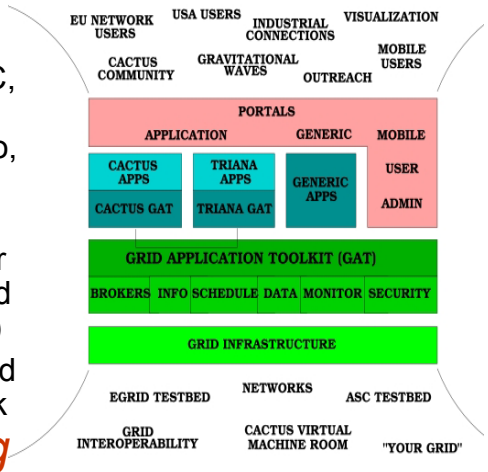


Grid Application Toolkit

- Application developer should be able to build simulations, such as these, with tools that easily enable dynamic grid capabilities
- Want to build programming API to easily incorporate:
  - Query information server (e.g. GIIS)
    - What's available for me? What software? How many processors?
  - Network Monitoring
  - Resource Brokering
  - Decision Routines (Thorns)
    - How to decide? Cost? Reliability? Size?
  - Spawning Routines (Thorns)
    - Now start this up over here, and that up over there
  - Authentication Server
    - Issues commands, moves files on your behalf (can't pass-on Globus proxy)
  - Data Transfer
    - Use whatever method is desired (Gsi-ssh, Gsi-ftp, Streamed HDF5, scp…)
  - Etc…
- Need to be able to test Grid tools as swappable plug-and-play

GridLab: Enabling Dynamic Grid Applications

- Large EU Project under negotiation with EC
- Members: AEI, ZIB, PSNC, Lecce, Athens, Cardiff, Amsterdam, SZTAKI, Brno, ISI, Argonne, Wisconsin, Sun, Compaq
- Grid Application Toolkit for application developers and infrastructure (APIs/Tools)
- Will be around 20 new Grid positions in Europe !! Look at *www.gridlab.org* for details



Credits