Cactus Code and Grid Programming

Here at GGF1:
Gabrielle Allen, Gerd Lanfermann,
Thomas Radke, Ed Seidel

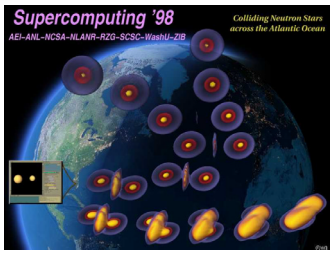
Max Planck Institute for Gravitational Physics,
(Albert Einstein Institute)

Cactus: Parallel, Collaborative, Modular Application Framework

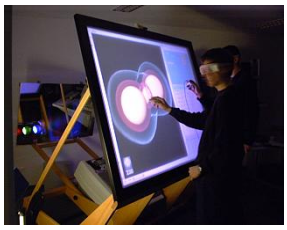
http://www.CactusCode.org

- Open source PSE for scientists and engineers ... USER DRIVEN ... easy parallelism, no new paradigms, flexible, Fortran, legacy codes.
- Flesh (ANSI C) provides code infrastructure (parameter, variable, scheduling databases, error handling, APIs, make, parameter parsing)
- Thorns (F77/F90/C/C++) 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
- Everything implemented as a swappable thorn ... use best available infrastructure without changing application thorns.
- Collaborative, remote and Grid tools
- Computational Toolkit: existing thorns for (Parallel) IO, elliptic, MPI unigrid driver, coordinates, interpolations, and more.
- Integrate other common packages and tools, HDF5, PETSc, GrACE ..

Grid-Enabled Cactus



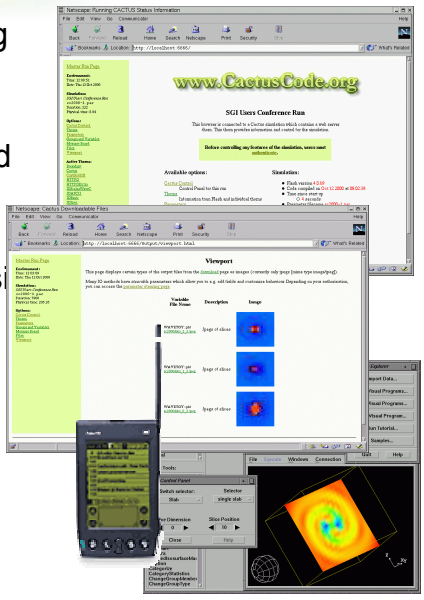
- Cactus and its ancestor codes have been using Grid infrastructure since 1993 ... motivated by simulation requirements ...
- Support for Grid computing was part of the design requirements for Cactus 4.0 (experiences with Cactus 3)
- Cactus compiles out-of-the-box with Globus [using globus device of MPICH-G(2)]



- Design of Cactus means that applications are unaware of the underlying machine/s that the simulation is running on ☺ applications become trivially Grid-enabled
- Infrastructure thorns (I/O, driver layers) can be enhanced to make most effective use of the underlying Grid architecture
- Involved in lots of ongoing Grid projects ....

Working and Used "User" Grid Tools

- Remote monitoring and steering of simulations (thorn HTTPD).
- Just adding point-and-click visualization to Web Server, and embedded visualization.
- Remote visualization ... data streamed to local site for analysis with local clients "Window into simulation"
- Notification by Email at start of simulation, including details of where to connect (URLs) for other tools.
- Checkpointing and restart between machines.



Grand Picture

Remote Viz in St Louis

Remote Viz and steering from Berlin

Remote steering and monitoring from airport

Viz of data from previous simulations in SF café

IsoSurfaces

HDF5

http

DataGrid/DPSS Downsampling

T3E: Garching

Origin: NCSA

Globus

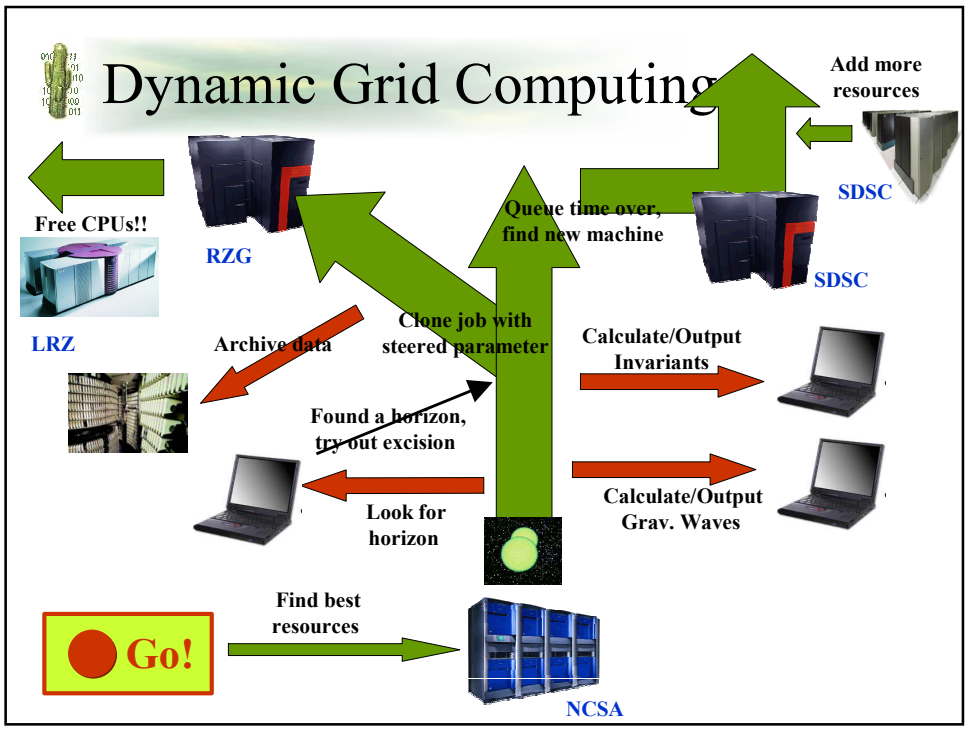
Simulations launched from Cactus Portal

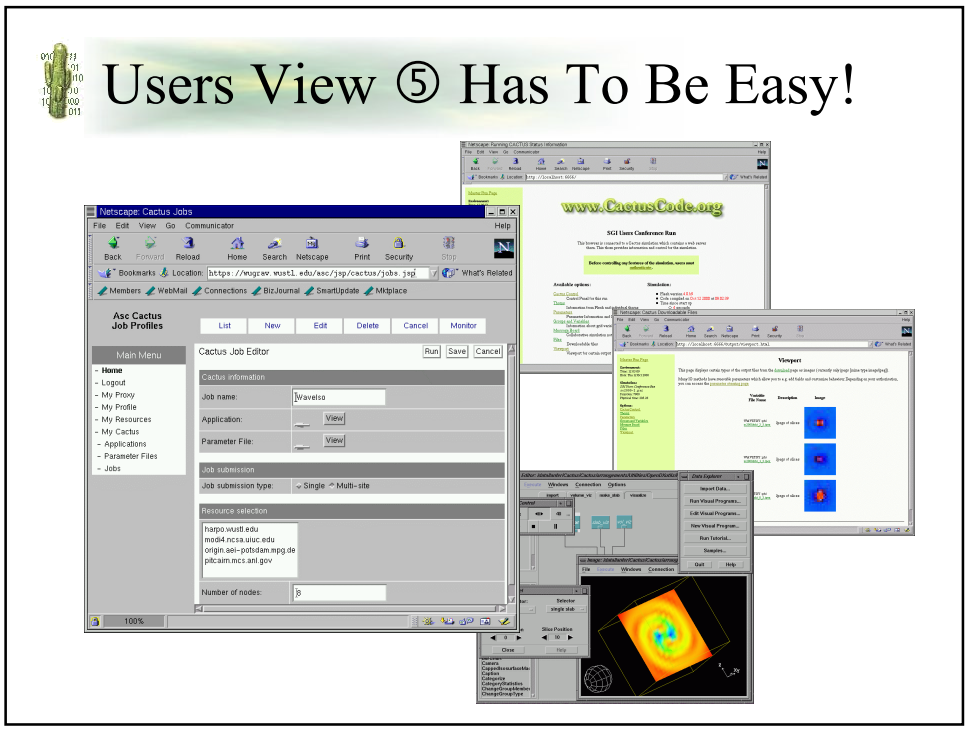
Grid enabled Cactus runs on distributed machines

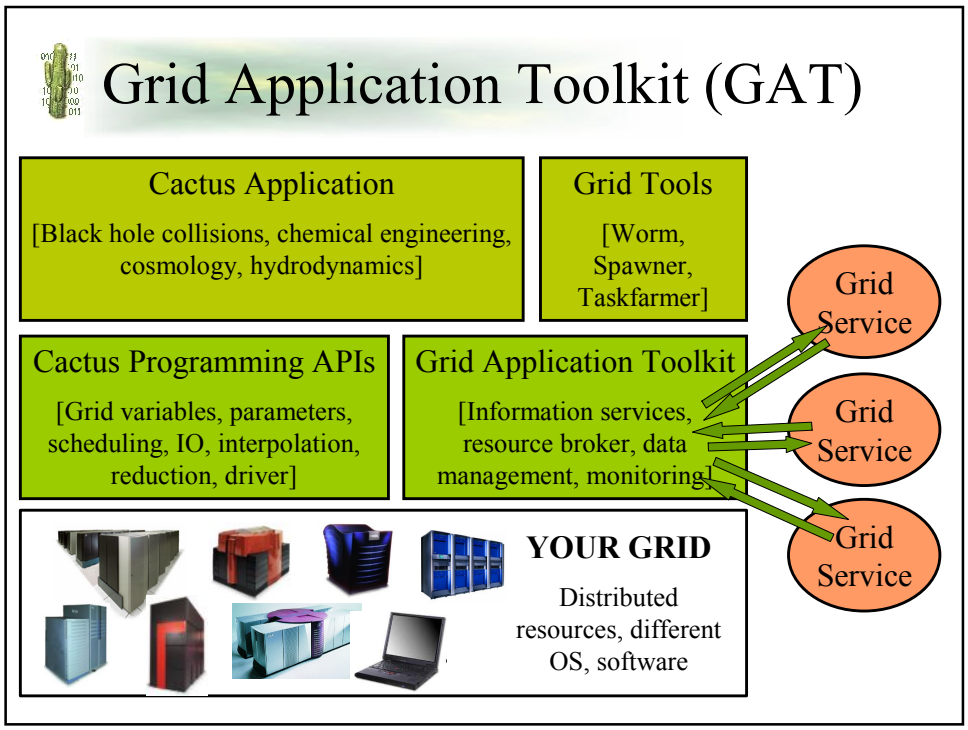
Cactus Worm

- Egrid Test Bed: 10 Sites
- Simulation starts on one machine, seeks out new resources (faster/cheaper/bigger) and migrates there, etc, etc
- Uses: Cactus, Globus
- Protocols: gsissh, gsiftp, streams or copies data
- Queries Egrid GIIS at each site
- Publishes simulation information to Egrid GIIS

- Demonstrated at Dallas SC2000
- Development proceeding with KDI ASC (USA), TIKSL/GriKSL (Germany), GrADS (USA), Application Group of Egrid (Europe)
- Fundamental dynamic Grid application !!!
- Leads directly to many more applications


Dynamic Grid Computing
Add more resources
SDSC
SDSC
Free CPUs!!
RZG
Queue time over, find new machine
LRZ
Archive data
Clone job with steered parameter
Calculate/Output Invariants
Found a horizon, try out excision
Look for horizon
Calculate/Output Grav. Waves
Go!
Find best resources
NCSA


Users View ☹ Has To Be Easy!
Netscape: Cactus Jobs
Asc Cactus Job Profiles
Cactus Job Editor
www.CactusCode.org



Grid Application Toolkit (GAT)

- Application developer should be able to build simulations with tools that easily enable dynamic grid capabilities
- Want to build programming API to:
  - Query/Publish to Information Services
    - Application, Network, Machine and Queue status
  - Resource Brokers
    - Where to go, how to decide?, how to get there?
  - Move and manage data, compose executables
    - Gsiscp, gsiftp, streamed HDF5, scp, GASS, ...
  - Higher level grid tools:
    - Migrate, Spawn, Taskfarm, Vector, Pipeline, Clone, ...
  - Notification
    - Send me an Email, SMS, fax, page, when something happens.
- Much more.

Usability

- Standard User:
  - ThornList, parameter file, (configuration file, batch script)
- Grid User:
  - Cactus Application Portal (ASC Project, Version 2 soon)
- Application Developer:
  - Already Grid aware plug-and-play thorns for I/O, communication (MPICH-G)
  - Planned **Grid Application Toolkit** with flexible API for using Grid services and tools (Resource selection, data management, monitoring, information services, ...)
  - Higher level tools built from basic components: Worm/Migrator, Taskfarmer, Spawner ☺ can be applied to any application.
- Grid Infrastructure Developer:
  - GAT will provide the same interface to different implementations of Grid services, allowing tools to be easily tested and compared
  - As soon as new tools are available they can straightaway be used by Grid-enabled applications.

Portability

- Ported to many different architectures (everything we have access to), new architectures are usually no problem since use standard programming languages and packages.
- Cactus data types used to ensure compatibility between machines (Distributed simulations have been successfully performed between many different architectures).
- Architecture independent checkpointing. Can checkpoint a Cactus run using 32 processors on machine A, and restart on machine B using 256 processors.

Interoperability

- We want to be able to make use of Grid services as they become available.
- Developing APIs for accessing grid services from applications, either as a library or remote call, e.g.

  Grid_PostInfo(id, machine, tag, info, service)

  where service could be mail, SMS, HTTPD, GIS, etc
- Lower level thorns will interpret to call the given service.
- Use function registration/overloading as appropriate, to call a range of services.
- Key point is we will be able to easily add a new service without changing application code.

Reliable Performance

- Initial implementation of remote performance monitoring using PAPI library and tools (thorn PAPI)
- Since Cactus APIs manages storage assignment and communication it will be possible to make use of this information dynamically.
- The Grid Application Toolkit will include APIs for performance monitoring (Grid and Application) and contract specification.
- Projects underway to develop application code instrumentation, both from supplied data (Cactus configuation files) and dynamically generated data.

Reliability and Fault Tolerance

- Initial implementation of Cactus Worm was quickly coded for SC2000, with no emphasise on fault tolerance
- These experiments showed how important fault tolerance was, as well as detailed log files.
- The Grid Application Toolkit will have error checking and reporting, eg check a machine is up before sending data to it.
- But still a problem ... what to do when one machine (e.g. In a cluster) fails during simulation?

Security and Privacy

- We want it ⑤
  - Passwords, Certificates, Multiple certificates?
  - Who can access information about a simulation?
  - Who can steer a simulation?
  - How does the resource broker know about my resources?
  - Collaborative issues, groups
  - Who can connect to this socket?
- How can we include in GAT? Has to be easy for users,
  - Scientists don't care too much about security
  - They don't keep private stuff on remote resources
- Cactus Grid Tools send data, information through sockets, problems with firewalls.