# Numerical Relativity With Cactus

Gabrielle Allen and Thomas Radke

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

## Cactus in a Nutshell

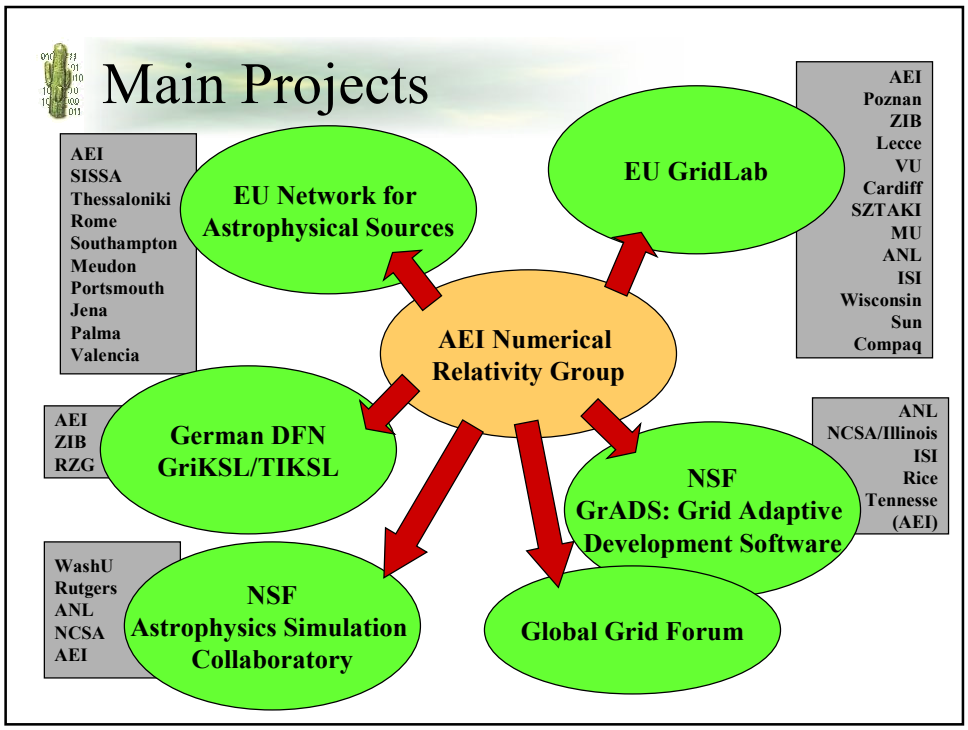
- Programming framework designed for scientists
- Designed, motivated by Numerical Relativity needs (very large-scale simulations, collaborative, portable, user-focused, coupling to viz, …)
- Keywords:
  - Modular, Portable, Parallel, Collaborative, Extensible, Easy-to-use
- The result of a whole communities expertise and experiences over many years (Grand Challenges, …)
- Cactus acts a the “main” routine of your code, it takes care of e.g. parallelism, IO, checkpointing, parameter file parsing for you (if you want), and provides different computational infrastructure such as parallel reduction operators, interpolators, coordinates, elliptic solvers, …

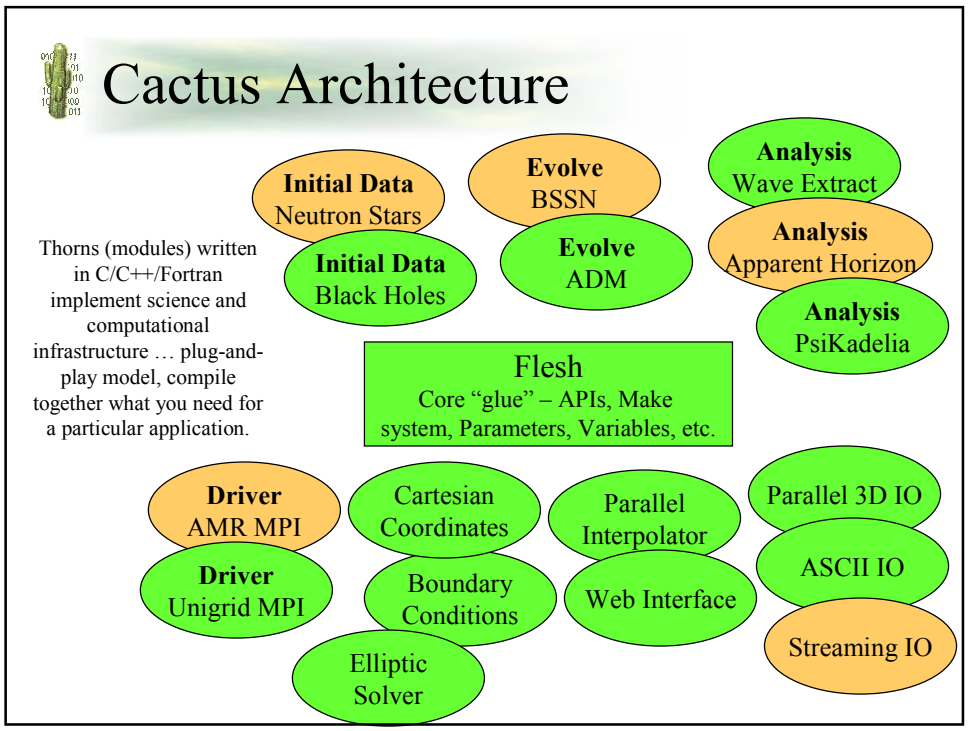
# Cactus for Numerical Relativity?

- Already a large and successful community of numerical relativity users
- Lets researchers concentrate on their science
- Best computer science infrastructure always available for everyone
- Many public numerical relativity thorns and tools
- Many more private numerical thorns being developed for current research
- Enables collaborations between groups
- Eases getting into the field
- Allows focusing of research on just one aspect of the field
- Use all available computing resources
- Numerical relativity community … code reuse, documentation, support community, continuously improving common code, connection to computer science projects, …

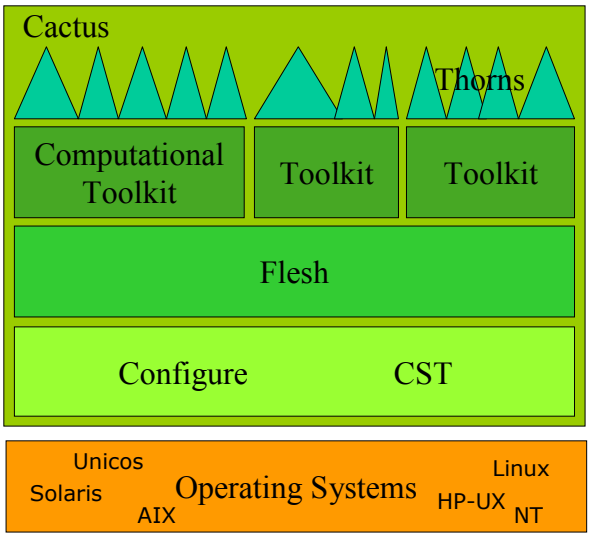
# Not Just Large Scale Simulations …

- Encourages/enforces “good” computing practices
- CVS (Source Code Versioning)
- Modularity
- Testsuites
- Documentation
- Code reuse
- Code sharing
- Infrastructure (Visualization, Computational Science)
- Contribute to and tie together different communities

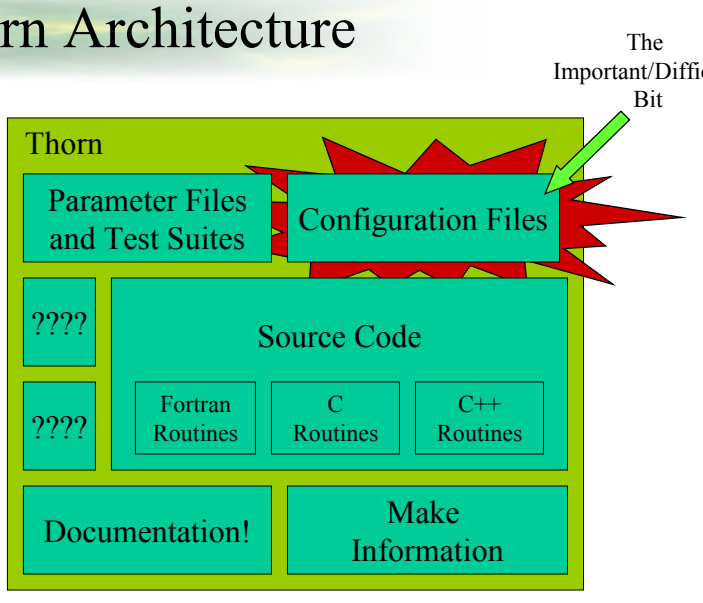

Main Projects
AEI
SISSA
Thessaloniki
Rome
Southampton
Meudon
Portsmouth
Jena
Palma
Valencia
EU Network for Astrophysical Sources
EU GridLab
AEI
Poznan
ZIB
Lecce
VU
Cardiff
SZTAKI
MU
ANL
ISI
Wisconsin
Sun
Compaq
AEI Numerical Relativity Group
AEI
ZIB
RZG
German DFN GriKSL/TIKSL
ANL
NCSA/Illinois
ISI
Rice
Tennesse
(AEI)
NSF GrADS: Grid Adaptive Development Software
WashU
Rutgers
ANL
NCSA
AEI
NSF Astrophysics Simulation Collaboratory
Global Grid Forum


Cactus Architecture
Thorns (modules) written in C/C++/Fortran implement science and computational infrastructure … plug-and-play model, compile together what you need for a particular application.
Initial Data Neutron Stars
Initial Data Black Holes
Evolve BSSN
Evolve ADM
Analysis Wave Extract
Analysis Apparent Horizon
Analysis PsiKadelia
Flesh
Core "glue" – APIs, Make system, Parameters, Variables, etc.
Driver AMR MPI
Driver Unigrid MPI
Cartesian Coordinates
Boundary Conditions
Elliptic Solver
Parallel Interpolator
Web Interface
Parallel 3D IO
ASCII IO
Streaming IO

# Cactus Architecture




# Thorn Architecture



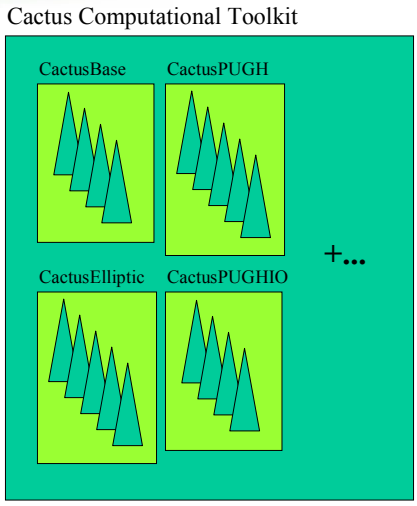
# Thorn Configuration Files

- Each thorn contains three configuration (ccl) files, which detail the thorn's interface with the Flesh and other thorns
- These files are parsed (Perl) during compilation by the CST (Cactus Specification Tool), which generates wrappers, argument lists (Fortran!), parameter lists, a run list etc.

| | |
|---|---|
| **interface.ccl** | **Variables, inheritance** |
| **param.ccl** | **Parameters** |
| **schedule.ccl** | **Routines, run order** |

# Thorn Collections

- For organizational convenience, thorns are grouped into *arrangements*
  - may have related functionality (e.g. IO or Maxwell solvers)
  - may have the same author
  - may contain everything needed for one problem
  - e.g. CactusBase, CactusPUGH
- We call a collection of arrangements, a toolkit



# How It Works …

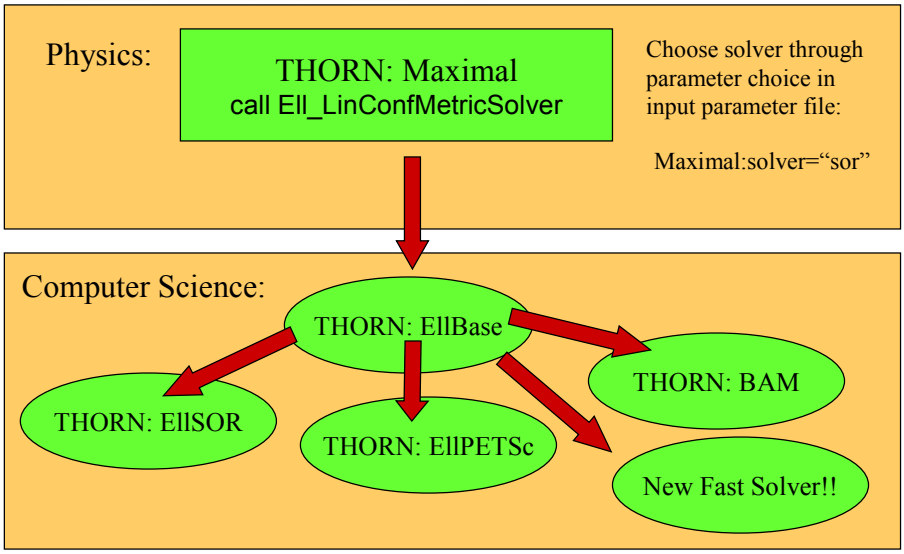
- E.g. Physics thorns just want to solve an elliptic equation (maximal slicing), they don't want to know the computational details, they just want the best available solver …

$$\nabla^2{}_g \alpha + \mathrm{M}\,\alpha = \mathrm{N}$$

call Ell_LinConfMetricSolver(ierr, cctkGH, imetric, ifield,
im, in, abstol, reltol, solver)

- Choose actual solver with a parameter
  solver = "petsc", "sor", "bam", …
- If a better solver becomes available, no physics source code needs to be changed, just link in new thorn

# Elliptic Solvers



# Support

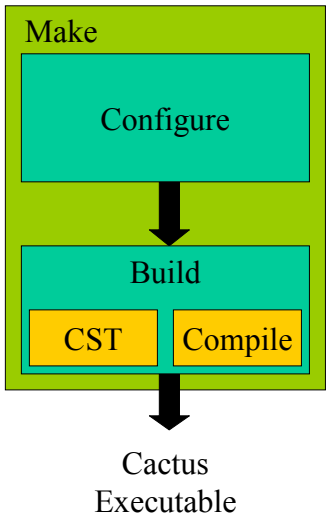
- Documentation: look at web pages for complete list, currently User's Guide, Thorn Guide, FAQ, HOWTO's, Visualization Tools, Releases, Specs, …
- Mailing Lists: Join
  - news@cactuscode.org,
  - users@cactuscode.org,
  - developers@cactuscode.org
- Mailing Lists: For each distributed arrangement of thorns
- Bug and feature request tracking system (see web pages)
- Email: cactusmaint@cactuscode.org
- [Email: allen@aei.mpg.de]

# Distribution

- Code distributed primarily from the AEI using CVS
  - www.cactuscode.org
- Two versions always available: "stable" (Beta 10) and "development"
- Script available for easy checkout from a "ThornList"

  GetCactus MyThornList.th
- Major problem for new users is how to find out which thorns are needed?
  - Get a ThornList from a colleague or web page.
  - Generate a ThornList from a parameter file: MakeThornList isco.par
  - Gradually getting more scripts to make this easier
  - Other suggestions?

# Creating Executable




Two Steps:

1. Configure: Set up machine and compiler dependent information.
2. Build: Create code from thorn configuration files. Cycle through each thorn creating object library then link together.

# Creating the ISCO Executable

- Create a ThornList for the executable: isco.th
- Choose a configuration name (here isco, describing the type of application we are going to run)
- Configure:
  - gmake isco-config THORNLIST=isco.th
  - gmake isco-config THORNLIST=isco.th MPI=NATIVE
- Make:
  - gmake isco
  - gmake isco WARN=yes TJOBS=4
- Executable: exe/cactus_isco
- Many Options: See Users Guide or gmake help

# isco.th

```
!DESC "isco_small.par [Fri Jul 13 09:47:11 CEST 2001]"

!REPOSITORY_LOCATION cvs.cactuscode.org
!REPOSITORY_NAME /cactusdevcvs
!REPOSITORY_TYPE pserver
CactusBase/Boundary
CactusBase/CartGrid3D
CactusBase/IOASCII

Etc …
```

**ThornList:**

- List of thorns to compile into an executable

  **gmake isco-config isco.th**

- Optionally contains directives for checking out from CVS

  **GetCactus isco.th**

- Get from a colleague, make your own, or generate from a parameter file

  **MakeThornList isco.par –o=isco.th**

# Testing the ISCO Executable

- Many thorns include sample parameter files and output which is rerun to compare new output.
- Tests:
  - Additional thorns, code changes, still give expected result
  - Same answer on different number of processors
  - Same answer on different machines
  - gmake isco-testsuite
- Copy example parameter files from each thorn (which can be run with the thorns in this executable) to “examples” directory
  - gmake isco-examples

# Parameter File

- Cactus executables are run using a parameter file
- Each thorn defines the parameters it uses, their ranges, a description, and a default value (param.ccl)
- If parameter is not set in parameter file, default is used
- Parameter checking at run time …

einstein::initial_data = “ultimate black hole data”

WARNING[L1,P0] (Cactus): ParameterSetKeyword: Unable to set keyword Einstein::initial_data - Ultimate black holes not in any active range

WARNING level 0 in thorn Cactus processor 0

(line 145 of c:\home\allen\CACTUS\Cactus\configs\isco\build\Cactus\main\SetParams.c):

-> CCTKi_SetParameter: Range error setting parameter einstein::initial_data to Ultimate black holes

# Parameter File: isco.par

# Running Cactus

- Single processor
  ./exe/cactus_isco MyParFiles/isco.par
- Multiple processor (if compiled with MPI)
  mpirun -np 8 ./exe/cactus_isco MyParFiles/isco.par
- Other run options
  ./exe/cactus_isco –help
- Version information for each compiled source file with strings

# Computational Toolkit

**CactusBase**
- Boundary, IOUtil, IOBasic, CartGrid3D, IOASCII, Time

**CactusBench**
- BenchADM

**CactusConnect**
- HTTPD, HTTPDExtra

**CactusExample**
- WaveToy1DF77, WaveToy2DF77

**CactusElliptic**
- EllBase, EllPETSc, EllSOR, EllTest

**CactusPUGH**
- PUGHInterp, PUGH, PUGHSlab, PUGHReduce

**CactusPUGHIO**
- IOFlexIO, IOHDF5, IsoSurfacer

**CactusIO**
- IOJpeg

**CactusTest**
- TestArrays, TestCoordinates, TestInclude1, TestInclude2, TestComplex, TestInterp, TestReduce

**CactusWave**
- IDScalarWave, IDScalarWaveC, IDScalarWaveCXX, WaveBinarySource, WaveToyC, WaveToyCXX, WaveToyF77, WaveToyF90, WaveToyFreeF90

**CactusExternal**
- FlexIO, jpeg6b

**CactusUtils**
- NaNChecker

# Einstein Toolkit: CactusEinstein

- Part of supported distribution
- Infrastructure Thorns:
  - Einstein (parameters, standard variables)
- Evolution Thorns:
  - ADM
- Initial Data Thorns:
  - IDAnalyticBH
  - IDAxiBrillBH
  - IDBrillData
  - IDLinearWaves
- Analysis Thorns
  - PsiKadelia
  - Extract
  - TimeGeodesic
  - AHFinder
  - ADMConstraints

# Thorn Einstein

- The “glue” for our numerical relativity thorns
- Defines standard grid functions calculated by initial data and analysis thorns
  - gxx, gxy, gxz, gyy, gyz, gzz
  - kxx, kxy, kxz, kyy, kyz, kzz
  - alp, betax, betay, betaz
- Optional conformal factor (and derivatives)
- Optional mask for excision
- Standard parameters (e.g. evolution method, initial data)
- Lapse and shift choices
  - Mixed slicings

# Other Numerical Relativity Thorns

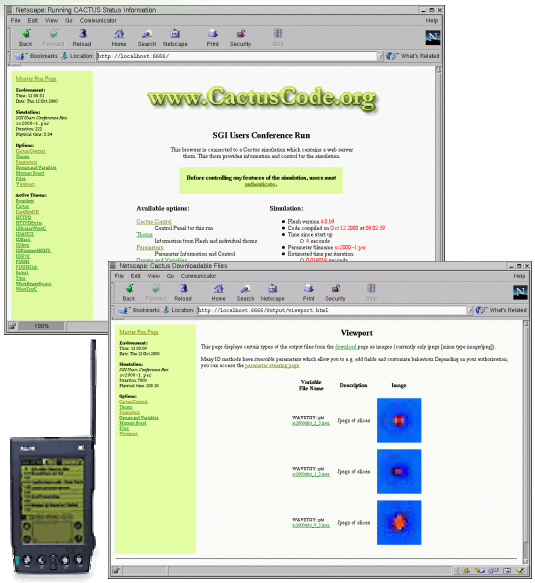
- Evolution
  - ADM_BSSN, ScalarWave, Maximal
  - MinDist_Shift (minimal distortian shift equations)
  - conf_adm (conformal traceless 3+1 form of evolution equations)
  - Mahc_Evolve (HRSC for relativistic fluid coupled to spacetime)
  - Newtonian_HRSC (Newtonian fluid coupled to Newtonian gravity)
- Elliptic
  - BAM_Elliptic (multigrid, also initial data)
  - BAM_VecLap (multigrid solver for vector elliptic equations)
- Tools
  - FishEye (coordinate transformations)
  - Cartoon2D (boundary conditions for solving 3D axisymmetric problems in 2D)
  - Mahc_Boundary
  - Resid_Quasi_Equil
  - EOS_Table_Reader

# Other Numerical Relativity Thorns II

- Initial Data
  - IDAxiOddBrillBH, DistortedBHIVP, RotatingDBHIVP, IDLinearBHIVP, AnalyticBH_PN
  - IVP (solves Hamiltonian and Momentum constraints)
  - Mahc_Init_Data
  - Quasi_Equil (Quasi-equilibrium neutron stars in binary orbit)
- Analysis:
  - Embedding (isometric embeddings of surfaces with spherical topology in flat space)
  - Zorro (different analysis quantities, direct waveform extraction, full Cauchy data extraction for Teukolsky evolution)
  - ADM_Constraints (alternative constraint evaluator)
  - Mahc_Analysis

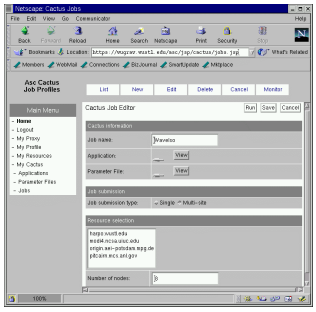
# Remote Monitoring: Thorn HTTPD

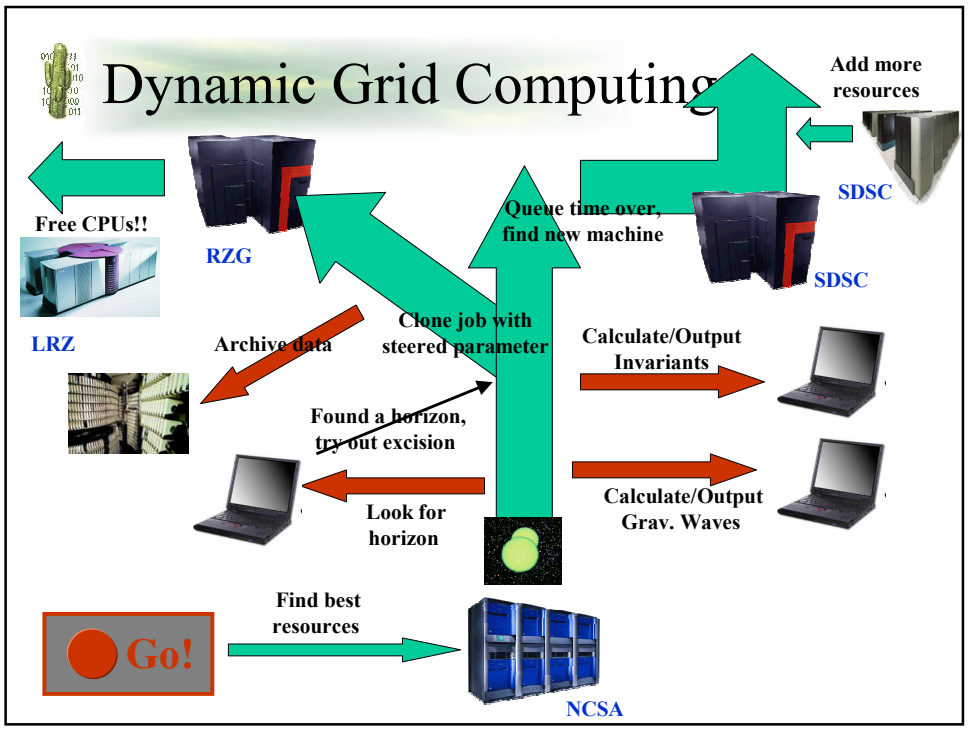
- Thorn which allows simulation any to act as its own web server
- Connect to simulation from any browser anywhere … collaborate
- Monitor run: parameters, basic inline visualization
- Automatic visualization with local clients (VizLauncher)
- Change steerable parameters
- See running example at www.CactusCode.org
- Wireless remote vizualization, monitoring and steering
- More features evolving … timing information, debugger …

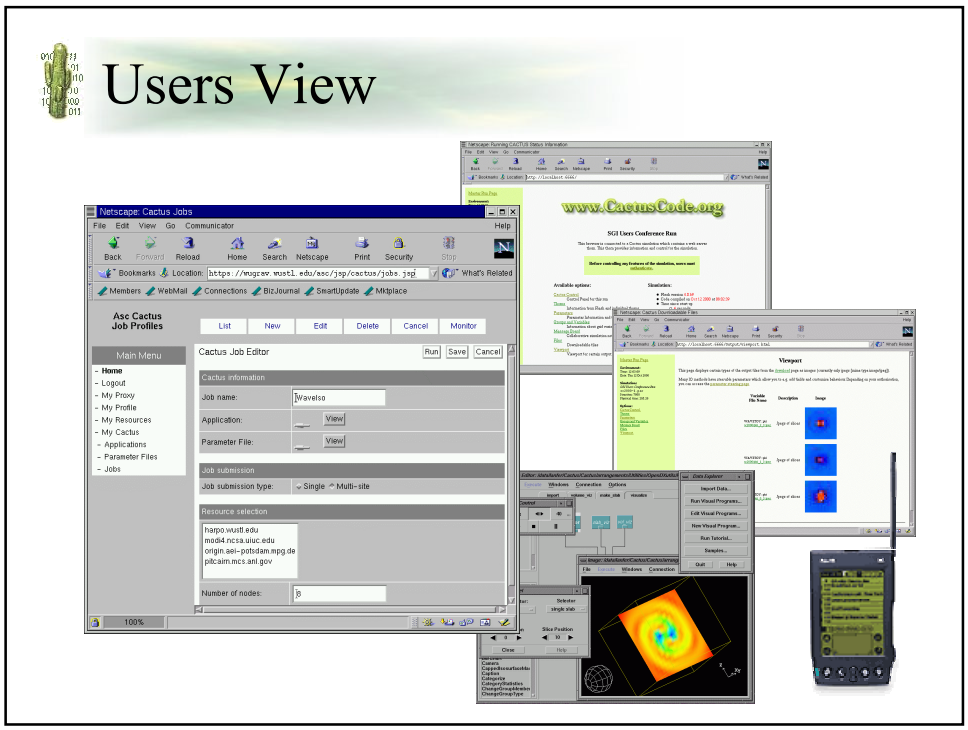



# Cactus/ASC Portal

- Locate all your available resources on the world wide grid
- One secure login
- Compose/Build/Find executables
- Manage parameter, configuration files (both personal and group)
- Submit jobs to queues tracks
- Track active jobs
- Collaborative features (project files, group accounting)
- Accessible from web browsers
- Building a “Cactus Virtual Machine Room”




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
Find best resources
Go!
NCSA


Users View
Netscape: Cactus Jobs
www.CactusCode.org
Asc Cactus Job Profiles
Cactus Job Editor
Main Menu