



GAT Workshop

Use Case Scenarios

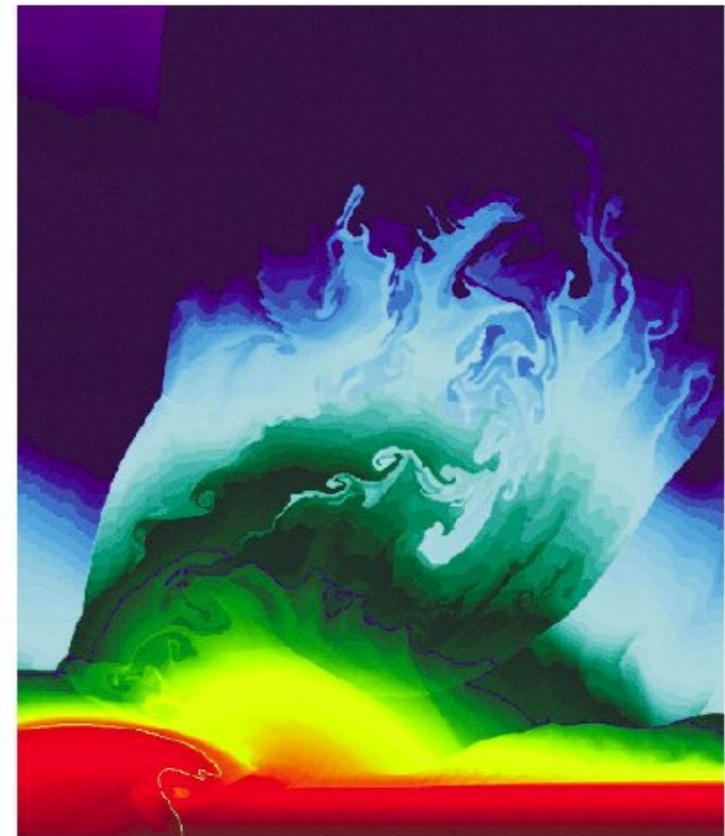
Ed Seidel

Director, Center for
Computation &
Technology, LSU
& Albert-Einstein-Institut
Potsdam, Germany

Co-Chair,
Applications Research Group, GGF
TWG-Apps, EU Gridstart Project

PHYSICS TOMORROW

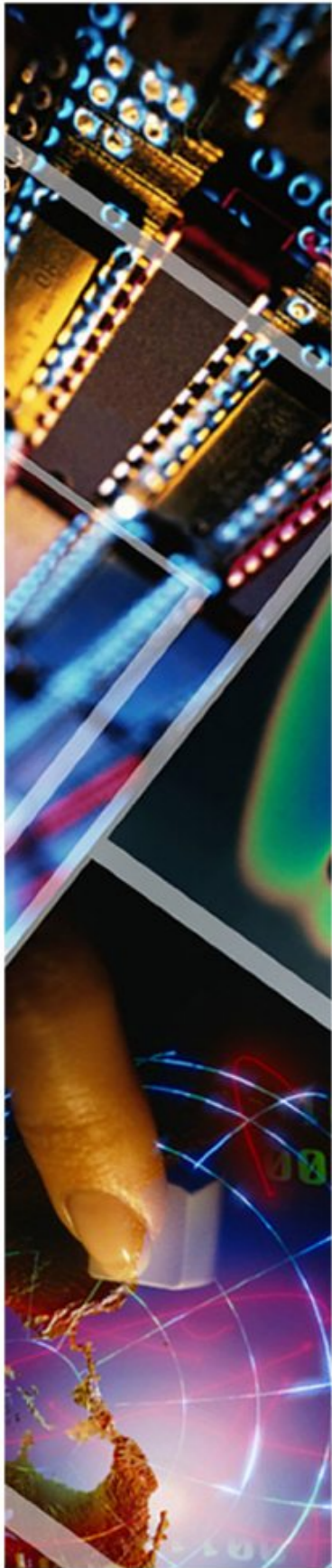
FEBRUARY 2002



THE POWER OF GRID COMPUTING

A Missing Component for Grid Computing

- Grid Aware Applications
 - Most apps developed before Grid
 - Have no “awareness”
 - Take limited advantage
- Develop new generation of Grid Applications
 - Not just task farming/resource selection, totally new apps
 - Apps find Grid services, or even provide them
 - Ease of Use: automatically find resources, given need!
 - Distributed simulations: use as many machines as needed!
 - Remote Viz and Steering, tracking: watch what happens!
 - Grid Enabled Modular Toolkits/Libraries for Grid Computation: Provide to Scientist/Engineer
 - Plug your Science/Eng. Applications in!



What do we want to do on Grids?

- Collaboration
 - Distributed communities share resources
- “Standard Things”
 - Task Farming, Resource Brokering, Distributed jobs...
- “New Scenarios”
 - Apps abstracted, become services
 - Dynamic apps decide their future, find their services (data, resources, people)
 - Apps distributed, spawned, task farmed, controlled, monitored with other apps/people





GridLab Project

www.gridlab.org

Funded by the EU (5+ M€), January 2002 – April 2005

Application and Testbed oriented

- Cactus Code, Triana Workflow, all the other applications that want to be Grid-enabled
- Gravitational Wave Astro (LIGO/GEO) and Numerical Relativity

- Main goal: develop a Grid Application Toolkit (GAT) and set of grid services/tools...:

- resource management (GRMS),
- data management,
- monitoring,
- adaptive components,
- mobile user support,
- security services,
- portals,



New Paradigms for Dynamic Grids

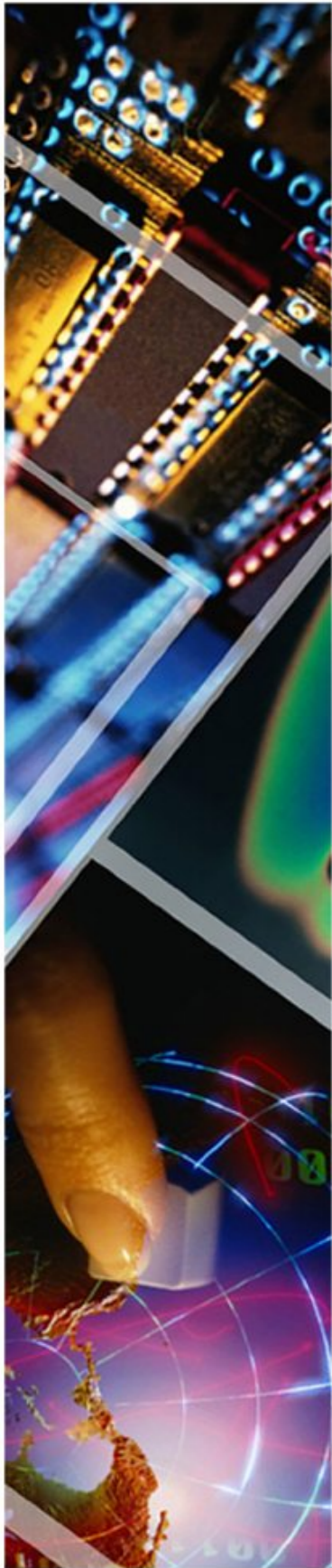
- Code/User/Infrastructure should be aware of environment
 - What Grid Services are available??
 - Resources available NOW, and what is their current state?
 - What is my allocation?
 - What is the bandwidth/latency between sites?
 - Where do I get my data from sensor network?
- Code/User/Infrastructure should be able to make decisions
 - A slow part of simulation runs asynchronously...spawn it off!
 - New, more powerful resources available...migrate there!
 - Machine went down...reconfigure and recover!
 - Need more memory (or less!): get it by adding(dropping) machines!
- Code/User/Infrastructure should be able to publish to central server for tracking, monitoring, steering...
 - Unexpected event...notify users!
 - Collaborators from anywhere all connect, examine simulation.





New Grid Applications

- Intelligent Parameter Surveys, Monte Carlos
 - May control other simulations!
- Dynamic Staging: move to faster/cheaper/bigger machine ("Grid Worm")
- Multiple Universe: clone to investigate steered parameter ("Grid Virus")
- Automatic Component Loading
 - Needs change, discover/load/execute component
- Automatic Convergence Testing
- Look Ahead
 - Spawn off and run coarser resolution to predict likely future
- Spawn Independent/Asynchronous Tasks
- Dynamic Load Balancing: inhomogeneous loads, multiple grids
- Inject dynamically acquired data

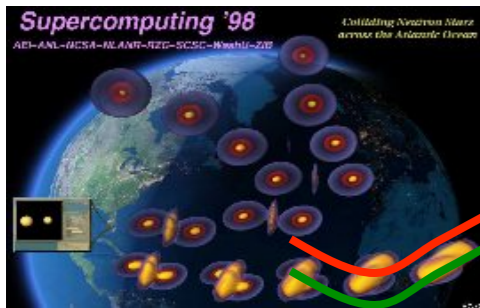




Cactus Worm: Illustration of basic scenario



- Cactus simulation (could be anything) starts, launched from a portal
- (Queries a Grid Information Server, finds available resources)
- Migrates itself to next site, according to some criterion
- Registers new location to GIS, terminates old simulation
- User tracks/steers, using http, streaming data, etc.....



- Continues around Europe...
- If we can do this, much of what we want can be done!

www.cactuscode.org

The composite image illustrates the Cactus Worm scenario. It features a satellite map of Europe with several red circular markers indicating simulation locations. A red arrow points from the text 'finds available resources' to one of these markers. In the foreground, there is a person at a computer workstation, a mobile PDA device, and a window titled 'Cactus' showing a list of 'Thorny problems in Numerical Relativity'.

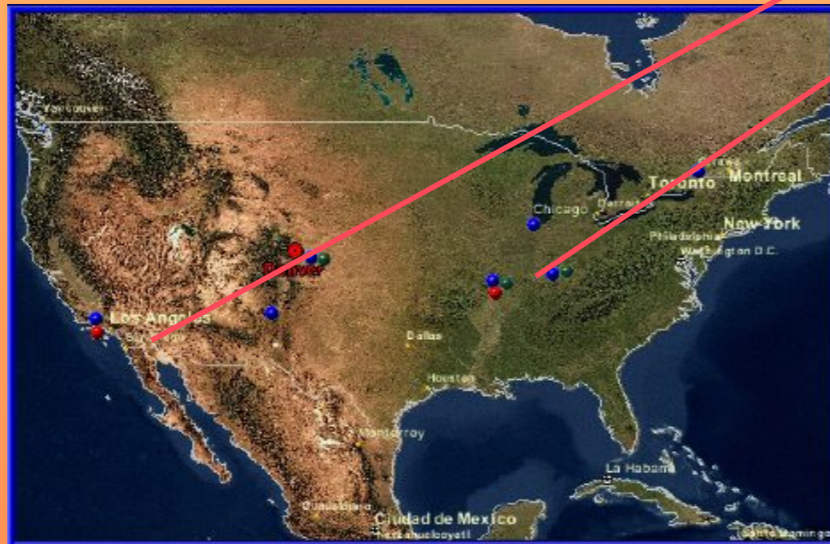
www.gridlab.org

Spawning on ARG Testbed

Main Cactus BH Simulation starts here



All analysis tasks spawned automatically to free resources worldwide



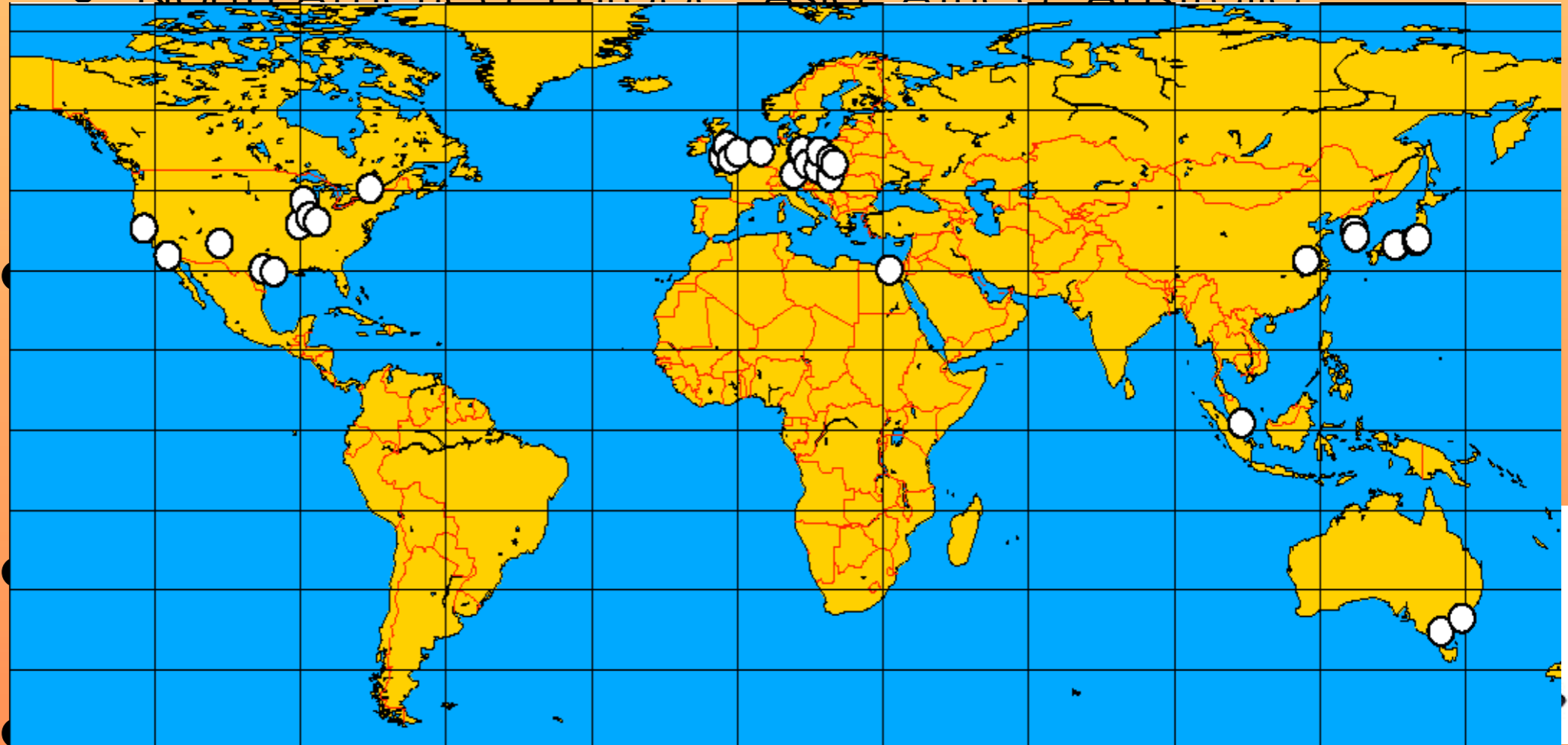
User only has to invoke “Spawner” thorn...





Global Grid Testbed Collaboration

- 5 continents
 - North America Europe Asia Africa Australia



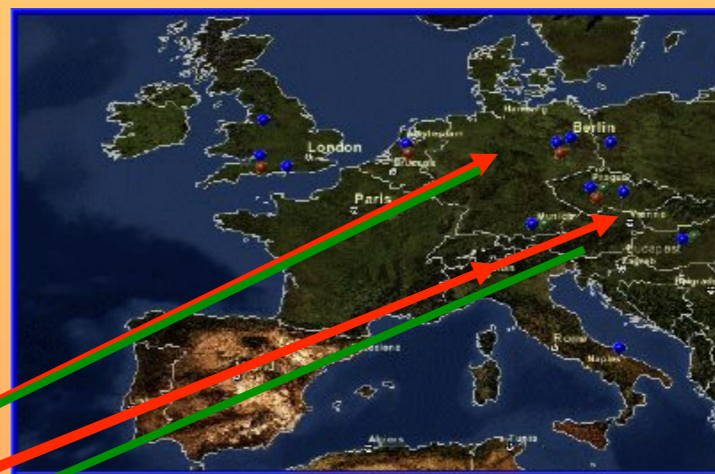
- And yet, all run same Grid infrastructure, and many can be used for applications...
- 4 different applications



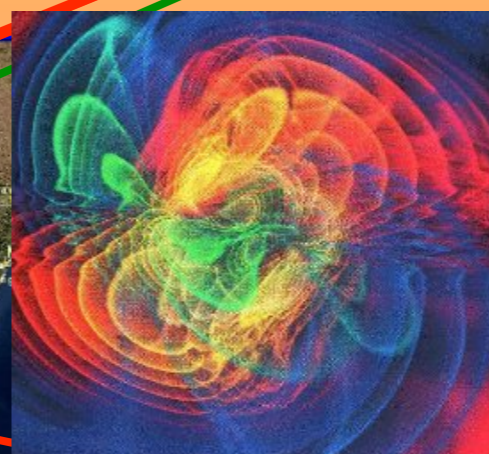
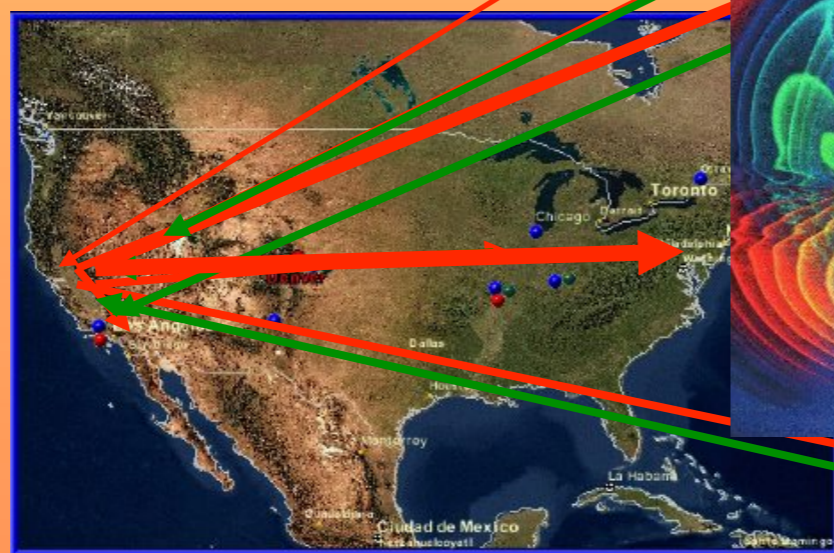


Task Farm/Steering Combo

Main Cactus BH Simulation starts in Berkeley



zens of small jobs
t out to test
ameters



Data returned for main job

Huge job generates remote data
to be visualized in Baltimore





Remote Viz/Steering watch/control simulation live...



Homepage: Running CACTUS Status Information

www.CactusCode.org

Available options:

Option	Description	Unit
AASTEP
AASTEP
AASTEP



Any Viz Client:
LCA Vision, OpenDX

Remote
Viz data

Streaming HDF5
Autodownsamples

Changing any steerable parameter

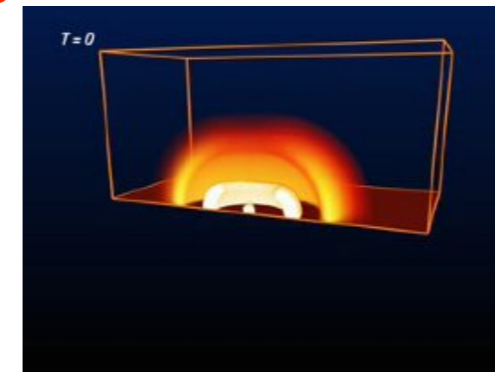
- *Parameters*
- *Physics, algorithms*
- *Performance*



www.cactuscode.org

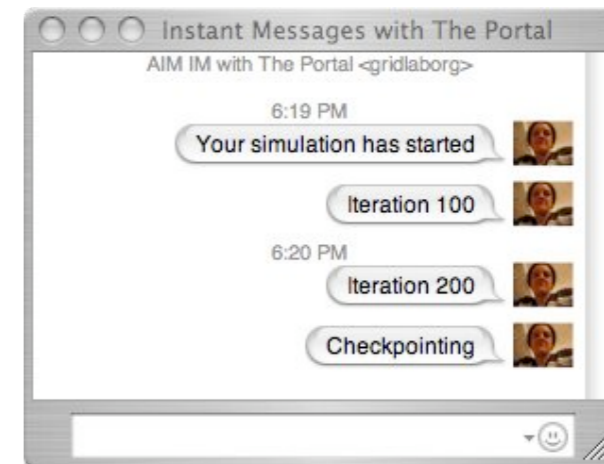
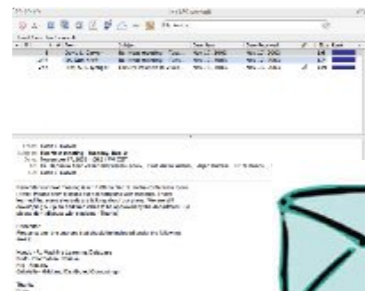


Remote
Viz data



Amira gridlab.org

Notification





Why we wrote GridLab

- Grids provide new paradigms, scenarios for working
- These scenarios are hard!
 - Even harder to do properly
 - fault tolerant, infrastructure, OS independent, modular, re-usable
- Researchers need them, but have no tools to build such apps
- The tools and interfaces need to become standardized



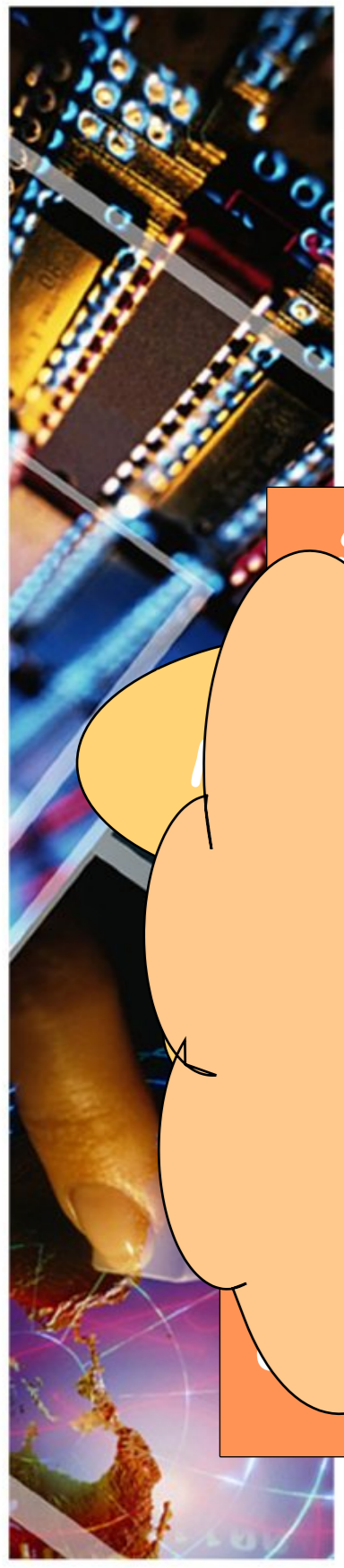
How to build Grid Apps?

Application

"Is there a better resource I could be using?"

"Where can I get my data?"

The Grid



S

y

1,

.

e

In

e?



Grid Application Toolkit

Application

“Is there a better resource I could be using?”

GAT_FindResource()

GAT

The Grid





Philosophy

- Application makes GAT API calls for operations which may be Grid-related.
- Application links against the GAT Engine
- Application runs irrespective of actual underlying infrastructure deployment
 - Engine loads adaptors which are valid in the environment extant when the application starts
 - Adaptors try to do Grid operations on request, on failure another adaptor provided function may be called.
- Application can thus be compiled, linked and tested without any Grid services
- Same application executable can run in a full Grid environment.



SAGA GGF Working Group

- Simple API for Grid Applications
- Aims to define standard APIs for commonly needed Grid operations
- Various implementations to be created
 - GridLab GAT with GAT Engine and adaptors
 - Others
- Leaders
 - Keith Jackson, LBNL
 - Tom Goodale, LSU
 - Stephen Pickles, Manchester

Summary

- Grids are coming
 - A lot of hype, promise, but definitely coming
- New applications must be created to take advantage
 - New paradigms
 - New Tools
 - Think creatively! I was inspired 4 years ago here!
- We are in the early phases
 - Similar to parallel computing decade ago
 - MPI, HPF, etc ---> GAT, other Grid App development tools
- Help us get this right!
 - This week: Workshop agenda very flexible: apps, API oriented
 - Future: need examples, follow through, adaptors

