Critical Grid Research Issues: Perspective and Lessons from Large-Scale Grids

Andrew A. Chien, Moderator

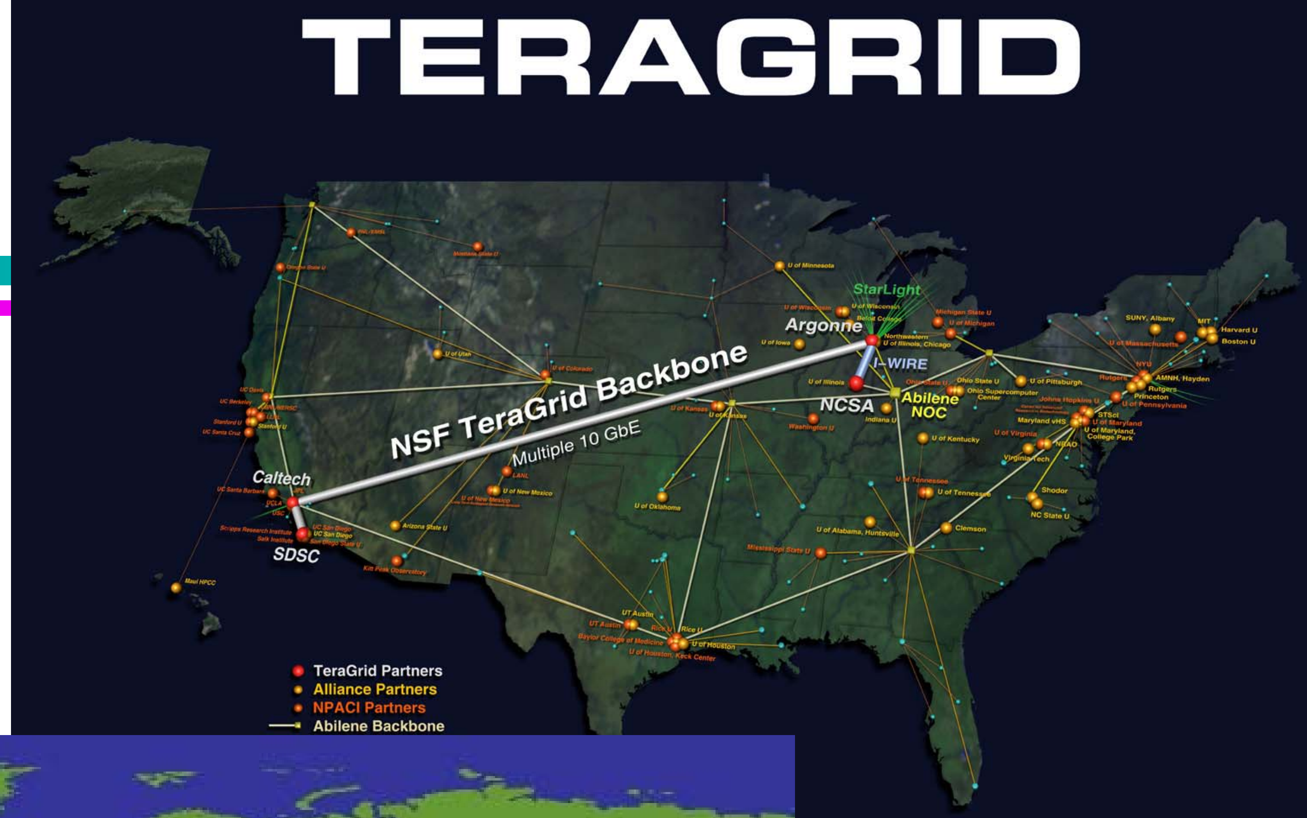
HPDC-13 Panel

June 6, 2004

Grids, Grids, Everywhere!

NASA Information Power Grid

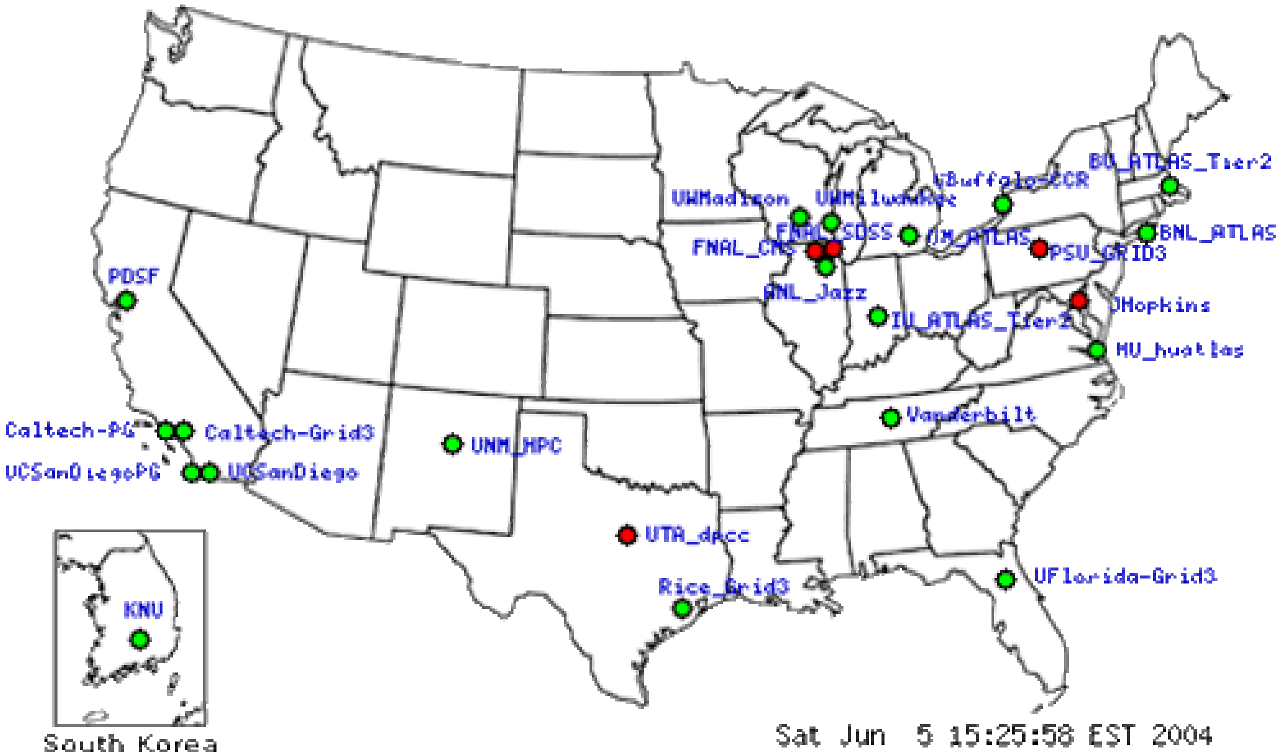
UK e-Science Grid



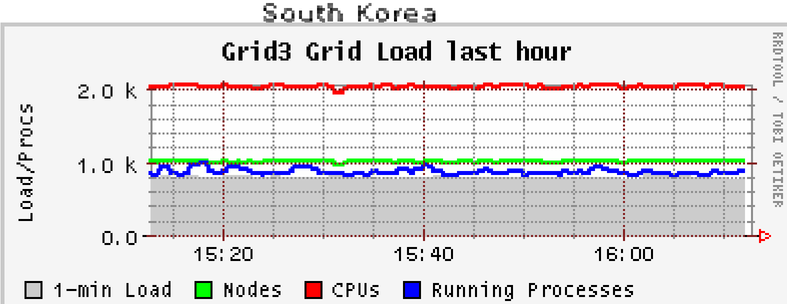
Planetlab

… and Grid2003!

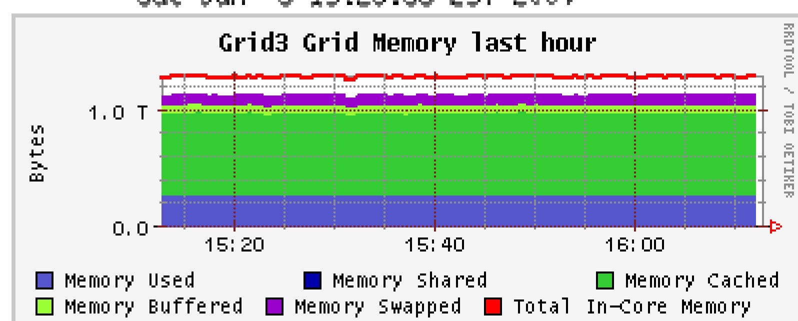
Grid2003



BU_ATLAS_Tier2
UWMadison
FNAL_SDSS
FNAL_CMS
BNL_ATLAS
PSU_GRID3
ANL_Jazz
PDSF
IU_ATLAS_Tier2
JHopkins
Caltech-PG
Caltech-Grid3
Vanderbilt
UNM_HPC
UCSanDiego
UTA_dpcc
Rice_Grid3
UFlorida-Grid3
KNU
South Korea
Sat Jun 5 15:25:58 EST 2004



Grid3 Grid Load last hour
Load/Procs
2.0 k
1.0 k
0.0
15:20
15:40
16:00
1-min Load
Nodes
CPUs
Running Processes



Grid3 Grid Memory last hour
Bytes
1.0 T
0.0
15:20
15:40
16:00
Memory Used
Memory Shared
Memory Cached
Memory Buffered
Memory Swapped
Total In-Core Memory

HPDC Research Maturing

Hypothesis

Design, Model

Simulation

Operation

Application Use

Measurement

Large-scale Experimentation

Dynamic Behavior
and Feedback

- Learn from Large-scale Production Grids
- What is Reality for Grid Systems? What is Not?
- What Works? What Doesn’t? What are the Hard Problems?
- Measurements, Use, Experience to Inform Research.

Panel Members

- Grid2003 Rob Gardner, U Chicago
- Planetlab Jeff Chase, Duke
- Condor Miron Livny, U Wisconsin
- Globus Ian Foster, U Chicago

- Andrew Chien, UCSD (Moderator)

Panel Charge and Organization

- Top 5 Things Learned (5 minutes each)
  - » What ARE major problems (and need extensive research)
  - » What are NOT major problems
  - » Two "takeaways" for every HPDC researcher
- Panel response (5 minutes)
- Questions / Comments from Audience

Experience and Lessons from Production Grids

Rob Gardner

University of Chicago

not major problems

- bringing sites into single purpose grids
- simple computational grids for highly portable applications
- specific workflows as defined by today's JDL and/or DAG approaches
- centralized, project-managed grids *to a particular scale, yet to be seen*

major problems

- Site, service providing perspective:
  - maintaining multiple “logical” grids with a given resource; maintaining robustness; long term management; dynamic reconfiguration; platforms
  - complex resource sharing policies (department, university, projects, collaborative), user roles
- Application perspective:
  - challenge of building integrated distributed systems
  - end-to-end debugging of jobs, understanding faults
  - collection, understanding of faults
  - limited workflows and interfaces, data exchange with other grids

three takeaways

- “think outside your grid”
- application developers/integrators do more complex things than simple computations
  - especially when complex, distributed datasets are involved
  - process activities/states need propagation to enable high level, intelligent decision making
- need to think of new ways to build and manage persistent infrastructures
  - favor decentralized, “entrepreneurial” models

Experience and Lessons from Production Grids

Jeff Chase
Duke University

http://www.cs.duke.edu/~chase

Grids are federated utilities

- Grids should preserve the control and isolation benefits of private environments.
- There's a threshold of comfort that we must reach before grids become truly practical.
  - Users need service contracts.
  - Protect users from the grid (security cuts both ways).
- Many dimensions:
  - decouple Grid support from application environment
  - decentralized trust and accountability
  - data privacy
  - dependability, survivability, etc.

Grids Need "Underware"

- Shift focus away from "meta-computing " middleware and toward underware and infrastructure services.
  - Enable user control over application environment.
  - Instantiate complete environment down to the metal.
  - OS is just another replaceable component.
  - Examples of "underware":
    - Virtual machines (Xen, Collective, JVM, etc.)
    - Net-booted physical machines (Cluster-on-Demand)
  - Innovate below OS and alongside it (infra-services).
  - Allot physical resources to each container/slice.

Grids Need Accountability

- Grid clients interact with many different components in different trust domains.
- Deep new trust management concerns go beyond basic support for authentication and secure communication.
- How to establish a Rule of Law in the Wild West?
- "Trust But Verify":
  - Non-repudiable actions: signed RPCs, etc.
  - Record/audit actions to detect deviant behavior.
  - Assign/prove responsibility when things go wrong.
  - Grounding in socio-legal-economic framework?

“Non-Problems”

- Technology advances are enabling new ways to transcend differences across sites.
  - Old: meta-APIs to “paper over” varying local facilities.
  - New: hide differences behind familiar low-level APIs.
  - API-free grid: focus on application-independent ways to grid-enable (“utilify”) applications?
- Grid “plumbing” is shifting to service frameworks and standardization efforts.
  - Plumbing is a technology; we just need to agree on pipes, threading, etc.
  - Focus on architecture: what/where are the hooks for policy, monitoring, diagnosis, adaptation, control?

“Takeaways”

- Underware
- Accountability

http://www.cs.duke.edu/~chase

Experience and Lessons from Production Grids

Miron Livny
Computer Sciences Department
University of Wisconsin-Madison
miron@cs.wisc.edu

Condor

not major problems
(but often studied extensively in rsch community)

- Performance
  - Meta scheduling
  - Grid economy
  - Communication overhead
  - Reservations
  - Predictions

are major problems
(and could benefit from extensive rsch in community)

- Trouble Shooting
  - Authentication
  - Software layers
  - Remote debugging
- Resource allocation (load control)
  - Storage
  - Connections
  - File descriptors

the two things "takeaways" you learned that you'd transplant into every researcher's head

- Robustness first performance later (information and control flow hold the key)
- Never assume that what you know is still true (always be prepared to react to the unexpected)

Experience and Lessons from Production Grids

Ian Foster
Argonne National Laboratories
and University of Chicago

Five Major Problems

- Troubleshooting & problem determination
  - Trace problems to causes; instrumentation
- Autonomic management
  - Manage scope of problems, provide QoS
- Trust and security
  - Could yet be a showstopper
- Application models
  - Integrating on-demand resources
- Heterogeneous schema
  - Integrating data, services, etc.

Five Non-Problems

- Scalability to millions of devices
  - We don't live in exponential regimes
- Basic resource access, monitoring, etc.
  - But that doesn't stop attempts to reinvent …
- Identifying interesting Grid applications
  - There are many of them
- Compilers and programming languages
  - At least not so far
- Coming up with problems
  - There are many more than 5!

Implications of Large-Scale Deployments for Grid Research

- It becomes <u>possible</u> to evaluate new ideas in realistic contexts and at realistic scales
  - ⇒ Will become <u>obligatory</u> for serious research
  - Places constraints on what is studied
  - Need consensus on platforms & workloads
- We can identify <u>real</u> problems associated with Grid creation, operation, & use
  - Again, makes research harder in some sense, but also more relevant