Grid Computing: An Evolving Vision

Manish Parashar
The Applied Software Systems Laboratory
Rutgers, The State University of New Jersey
http://www.caip.rutgers.edu/TASSL/

SOS, PPPL, February 14, 2004
Outline

• Grid computing – an evolving vision
  – definitions, drivers, enablers, …
• Grid computing – seamless aggregation
  – Globus v2
• Grid computing – seamless composition/interactions
  – Grid services, OGSA, Globus v3
• Grid computing – autonomic behaviors
  – AutoMate
• Summary and conclusions
Grid Computing

I.B.M. Making a Commitment to Next Phase of the Internet

By STEVE LOHR

I.B.M. is announcing today a new initiative to support and exploit a technology known as grid computing, which the company and much of the computer research community say is the next evolutionary step in the development of the Internet.

Merrill Lynch

Computing power on tap

Economist.com

Globus Grid Computing—the Next Internet

by John Roy/Steve Milunovich

The Internet was first a network and is now a communications platform. The next evolutionary step could be to a platform for distributed computing. This ability to manage applications and share data over the network is called “grid computing.”

Grid Computing

By M. Mitchell Waldrop

May 2002

Hook enough computers together and what do you get? A new kind of utility that offers supercomputer processing on tap.

Is Internet history about to repeat itself?
Defining Grid Computing …

• “… a concept, a network, a work in progress, part hype and part reality, and it’s increasingly capturing the attention of the computing community …” A. Applewhite, IEEE DS-Online

• “… grids are networks for computation – they are thinking, number-crunching entities. Like a decentralized nervous system, grids consist of high-end computers, servers, workstations, storage systems, and databases that work in tandem across private and public networks …” O. Malik, Red Herring

• “… a kind of hyper network that links computers and data storage owned by different groups so that they can share computing power…” USA Today

• ‘The Matrix’ crossed with ‘Minority Report’ … D. Metcalfe (quoted in Newsweek)

• “… use clusters of personal computers, servers or other machines. They link together to tackle complex calculations. In part, grid computing lets companies harness their unused computing power, or processing cycles, to create a type of supercomputer …” J. Bonasia, Investor’s Business Daily

• “… grid computing links far-flung computers, databases, and scientific instruments over the public internet or a virtual private network and promises IT power on demand…… All a user has to do is submit a calculation to a network of computers linked by grid-computing middleware. The middleware polls a directory of available machines to see which have the capacity to handle the request fastest…” A. Ricadela, Information Week
The Grid vision

• Imagine a world
  – *in which computational power (resources, services, data, etc.) is as readily available as electrical power*
  – *in which computational services make this power available to users with differing levels of expertise in diverse areas*
  – *in which these services can interact to perform specified tasks efficiently and securely with minimum of human intervention*
    • on-demand, ubiquitous access to computing, data, and services
    • new capabilities constructed dynamically and transparently from distributed services
  – *a large part of this vision was originally proposed by Fenando Corbato (The Multics Project, 1965, www.multicians.org)*
Grids – An evolving vision …

- Seamless aggregation
  - aggregation of capacity
  - aggregation of capability
- Seamless compositions and interactions
- Autonomic behaviors

Evolution

Revolution
Why the Grid? Revolution in Science

- **Pre-Internet:**
  - Theorize and/or experiment, alone or in small teams; then publish a paper.

- **Post-Internet:**
  - Construct and mine large databases of observational, experimental or simulation data,
  - Develop simulations and analysis,
  - Access and share specialized devices remotely,
  - Exchange information within distributed multidisciplinary teams.
Why the Grid? Revolution in Business

• Pre-Internet:
  – Central data processing facility.

• Post-Internet:
  – Enterprise computing is highly distributed, heterogeneous, inter-enterprise (B2B),
  – Business processes increasingly computing- and data-rich,
  – Outsourcing becomes feasible ➔ service providers of various sorts.
The Grid – Seamless aggregation

- Imaging Instruments
- Data Acquisition
- Advanced Visualization
- Analysis
- Computational Resources
- Large Scale Databases
Enabling Grid Computing - Exponentials

- Network vs. computer performance
  - Computer speed doubles every 18 months
  - Storage density doubles every 12 months
  - Network speed doubles every 9 months
  - Difference = order of magnitude per 5 years
- 1986 to 2000
  - Computers: x 500
  - Networks: x 340,000
- 2001 to 2010
  - Computers: x 60
  - Networks: x 4000

"When the network is as fast as the computer's internal links, the machine disintegrates across the net into a set of special purpose appliances"

(George Gilder)
The Grid – Seamless aggregation

- Coordinated resource sharing and problem solving in dynamic, multi-institutional virtual organizations.
  - A VO is a collection of users sharing similar needs and requirements in their access to processing, data and distributed resources and pursuing similar goals.

- Key concept:
  - Ability to negotiate resource-sharing arrangements among a set of participating parties (providers and consumers) and then to use the resulting resource pool for some purpose.
Computing-on-demand uses dynamically assigned (shared) pool of resources to support excess demand in flexible cost-effective fashion.
Online Access to Scientific Instruments

Advanced Photon Source

real-time collection

tomographic reconstruction

wide-area dissemination

archival storage

desktop & VR clients with shared controls

DOE X-ray grand challenge: ANL, USC/ISI, NIST, U.Chicago
Data Grids for High Energy Physics

There is a "bunch crossing" every 25 nsecs. There are 100 "triggers" per second. Each triggered event is ~1 MByte in size.

Tier 0
- CERN Computer Centre
  - Tier 0
  - ~622 Mbits/sec
  - ~100 MBytes/sec
  - ~20 TIPS
  - 1 TIPS is approximately 25,000 SpecInt95 equivalents

Tier 1
- France Regional Centre
- Germany Regional Centre
- Italy Regional Centre
- FermiLab ~4 TIPS
  - ~622 Mbits/sec

Tier 2
- Caltech ~1 TIPS
- Tier2 Centre ~1 TIPS
- Centre TIPS
- Centre TIPS
- Centre TIPS

Tier 3
- Institute ~0.25 TIPS
  - ~622 Mbits/sec
  - ~PBytes/sec

Tier 4
- Physics data cache
- ~1 MBytes/sec
- ~1 TIPI
- 1 TIPI

Physicists work on analysis "channels". Each institute will have ~10 physicists working on one or more channels; data for these channels should be cached by the institute server.

Image courtesy Harvey Newman, Caltech
Network for Earthquake Engineering Simulation

- NEESgrid: US national infrastructure to couple earthquake engineers with experimental facilities, databases, computers, & each other
- On-demand access to experiments, data streams, computing, archives, collaboration

NEESgrid: Argonne, Michigan, NCSA, UIUC, USC
SERVOGrid – Solid Earth Research Virtual Observatory will link Australia, Japan, USA ……

Repositories
Federated Databases

Sensor Nets

Streaming Data

Database

Loosely Coupled Filters

Closely Coupled Compute Nodes

Analysis and Visualization
DAME

- Airline
- Ground Station
- Maintenance Centre
- Engine Health (Data) Center

In flight data

~ Gigabyte per aircraft per Engine per transatlantic flight

~5000 engines

Global Network Such as SITA

Internet, e-mail, pager

Rolls Royce and UK e-Science Program
Distributed Aircraft Maintenance Environment
Evolving reference set of brains provides essential data for developing therapies for neurological disorders (multiple sclerosis, Alzheimer’s, etc.).

Today
- One lab, small patient base
- 4 TB collection

Tomorrow
- 10s of collaborating labs
- Larger population sample
- 400 TB data collection: more brains, higher resolution
- Multiple scale data integration and analysis
It takes a distributed virtual organization to design, simulate and build a complex system like an aircraft.
Virtual Observatory Astronomy Grid
Integrate Experiments

Radio
Far-Infrared
Visible

Visible + X-ray
Dust Map
Galaxy Density Map
Home Computers Evaluate AIDS Drugs

- Community =
  - 1000s of home computer users
  - Philanthropic computing vendor (Entropia)
  - Research group (Scripps)
- Common goal= advance AIDS research
Major Grid projects

• US projects:
  - PPDG, HEP, www.ppdg.net
  - iVDGL, www.ivdgl.org
  - TERAGRID (NSF), www.teragrid.org
    • IBM, Intel Qwest, Myricom, Sun Microsystems, Oracle.
  - National Middleware Initiative (NSF NMI), www.nsf-middleware.org
  - ESG, www.earthsystemgrid.org

• Asian projects:
  - TWGRID, www.twgrid.org
Major Grid projects

• Europe-based projects:
  – *EU DataGrid (EDG)*, www.edg.org
  – *LHC Computing GRID (LCG)*, cern.ch/lcg
  – *CrossGRID*, www.crossgrid.org
  – *DataTAG*, www.datatag.org
  – *GridLab*, www.gridlab.org
  – *EUROGRID*, www.eurogrid.org
Some Important Classes of Grids

- Computational Grids were origin of concepts and link computers across the globe – high latency stops this from being used as parallel machine.
- Knowledge and Information Grids link sensors and information repositories as in Virtual Observatories or Bioinformatics.
- Education Grids link teachers, learners, parents as a VO with learning tools, distant lectures etc.
- e-Science Grids link multidisciplinary researchers across laboratories and universities.
- Community Grids focus on Grids involving large numbers of peers rather than focusing on linking major resources – links Grid and Peer-to-peer network concepts.
Grid Technologies: Building an Open Grid infrastructure

• How do we:
  – Establish and evolve virtual organizations?
  – Manage security, trust, and privacy?
  – Manage and monitor distributed resources?
  – Federate data?
  – Discover available data, resources, services?
  – Provision diverse distributed activities?

• How do we do all these things in a robust, secure, scalable, interoperable fashion?
Technical Issues (obviously incomplete)

- Identity & authentication
- Authorization & policy
- Resource discovery
- Resource characterization
- Resource allocation
- (Co-)reservation, workflow
- Distributed algorithms
- Remote data access
- High-speed data transfer
- Performance guarantees
- Monitoring

- Adaptation
- Intrusion detection
- Resource management
- Accounting & payment
- Fault management
- System evolution
- Etc.
- Etc.
- ...
Grid Platform Example: Globus Toolkit (v1, v2)

• Primary development occurred at Argonne National Labs
  – Principals, Ian Foster and Carl Kesselman

• Open source
  – But architecture development was a closed process

• Toolkit approach: different “bundles” that can be installed depending upon what functions are desired

• API through CoG (Commodity Grid) kits
  – Java, Python, CORBA, Perl, Matlab, Web services, JSP

• www.globus.org
Layered Grid Architecture

“Coordinating multiple resources”: ubiquitous infrastructure services, app-specific distributed services

“Sharing single resources”: negotiating access, controlling use

“Talking to things”: communication (Internet protocols) & security

“Controlling things locally”: Access to, & control of, resources

Grid Evolution – From seamless aggregation to seamless composition/interaction

Electrical Chemical Lab Service

Computation service

Image processing service

Database service

Modalities e.g., MRI, CT, CR, Ultrasound, etc.
From Resources to Services: Managing Virtual Services

- Trying to manage total system properties:
  - e.g. Dependability, end-to-end QoS.
- “Resource” tends to connote a tangible entity to be consumed: CPU, storage, bandwidth, ...
- But many interesting services may be decoupled from any particular resource:
  - e.g. Finite element analysis service,
  - A service consumes resources, but how that happens is irrelevant to the client.
- “Service” forms a better base abstraction:
  - Can apply to physical or virtual.
Web Services

• At the heart of Web services is:
  – **WSDL**: Language for defining abstract service interfaces,
  – **SOAP (and friends)**: Binding from WSDL to bytes on the wire,
  – See Web Services case study for details.

• Web Services appears to offer a fighting chance at ubiquity

• But Web Services does not go far enough to serve a common base for the Grid…
Transient Service Instances

- “Web services” address discovery & invocation of persistent services
  - *Interface to persistent state of entire enterprise*
- In Grids, must also support transient service instances, created/destroyed dynamically
  - *Interfaces to the states of distributed activities*
  - *E.g. workflow, video conf., dist. data analysis*
- Significant implications for how services are managed, named, discovered, and used
Grid Evolution: Open Grid Services Architecture

• Service orientation to virtualize resources and unify resources/services/information
  – Everything is a service

• Embrace key Web services technologies: standard IDL, leverage commercial efforts
  – Standard interface definition mechanisms: multiple protocol bindings, local/remote transparency

• Include from Grids
  – Service semantics, reliability and security models
  – Lifecycle management, discovery, other services

• Multiple “hosting environments”
  – C, J2EE, .NET, …

• Result: standard interfaces & behaviors for distributed system management
Globus Toolkit v3 (GT3)
OGSA Technology

• Implement core OGSI interfaces:
  – Reference implementation of GS Specification.
• Support primary GT2 interfaces:
  – High degree of backward compatibility,
• Target multiple platforms & hosting environments:
  – J2EE, Java, C, .NET.
• New services:
  – SLA negotiation (GRAM-2), registry, replica location, community authorization, …
• Growing number of external contributions:
  – Database services (UK DAIS), Python hosting (LBNL), …

External adoption:
  – e.g., IBM, Avaki, UK e-Science projects.
The Grid World: Current Status

- Dozens of major Grid projects in scientific & technical computing/research & education
- Considerable consensus on key concepts and technologies
  - Open source Globus Toolkit™ a de facto standard for major protocols & services
  - Far from complete or perfect, but out there, evolving rapidly, and large tool/user base
- Industrial interest emerging rapidly
Standardizing The Grid: Global Grid Forum

- An open process for development of standards
  - Grid “Recommendations” process modeled after Internet Standards Process (IETF)
- A forum for information exchange
  - Experiences, patterns, structures
- A regular gathering to encourage shared effort
  - In code development: libraries, tools…
  - Via resource sharing: shared Grids
  - In infrastructure: consensus standards
- Research groups, working groups
- www.gridforum.org
Summarizing Current Trends: Smaller/Cheaper/Faster/Powerful/Connected ….

- Explosive growth in computation, communication, information and integration technologies
  - computing is ubiquitous, pervasive – communication is/will be
- Pervasive “anytime-anywhere” access environments
  - ubiquitous access to information via PCs, PDAs, Cells, smart appliances, etc. (billions of devices, millions of users)
  - peers capable of producing/consuming/processing information at different levels and granularities
  - embedded devices in clothes, phones, cars, mile-markers, traffic lights, lamp posts, refrigerators, medical instruments …
- “On demand” computational/storage resources, services
- The emerging national/global Cyberinfrastructure
The bad news ...

- **Unprecedented**
  - *scales, complexity, heterogeneity, dynamism and unpredictability, lack of guarantees*
    - *Millions of businesses, Trillions of devices, Millions of developers and users, Coordination and communication between them*

- The increasing system complexity is reaching a level beyond human ability to design, manage and secure
  - *programming environments and infrastructure are becoming unmanageable, brittle and insecure*

- A fundamental change is required in how applications are formulated, composed and managed
For example …

- Administration of individual systems is increasingly difficult
  - 100s of configuration, tuning parameters for DB2, WebSphere
- Heterogeneous systems are becoming increasingly connected
  - Integration becoming ever more difficult
- Architects can't intricately plan interactions among components
  - Increasingly dynamic; more frequently with unanticipated components
- More of the burden must be assumed at run time
  - But human system administrators can't assume the burden
    - 6:1 cost ratio between storage admin and storage
    - 40% outages due to operator error
- We need self-managing computing systems
  - Behavior specified by sys admins via high-level policies
  - System and its components figure out how to carry out policies
Autonomic Computing?

- Nature has evolved to cope with scale, complexity, heterogeneity, dynamism and unpredictability, lack of guarantees
  - self configuring, self adapting, self optimizing, self healing, self protecting, highly decentralized, heterogeneous architectures that work !!!
  - e.g. the autonomic nervous system
    - tells you heart how fast to beat, checks your blood’s sugar and oxygen levels, and controls your pupils so the right amount of light reaches your eyes as you read these words, monitors your temperature and adjusts your blood flow and skin functions to keep it at 98.6°F
    - coordinates - an increase in heart rate without a corresponding adjustment to breathing and blood pressure would be disastrous
    - is autonomic - you can make a mad dash for the train without having to calculate how much faster to breathe and pump your heart, or if you’ll need that little dose of adrenaline to make it through the doors before they close

- can these strategies inspire solutions?
- of course, there is a cost
  - lack of controllability, precision, guarantees, comprehensibility, …
- A.I. ? – duplication of human thought is not the ultimate goal
Autonomic Computing Characteristics

Self-protecting System designed to protect itself from any unauthorized access anywhere

Self-optimizing System designed to automatically manage resources to allow the servers to meet the enterprise needs in the most efficient fashion

Self-healing Autonomic problem determination and resolution

Self-configuring systems designed to define itself "on the fly"

www.research.ibm.com/autonomic
Project AutoMate: Autonomic behaviors on the Grid

- Enable the development of autonomic Grid applications that are context aware and are capable of self-configuring, self-composing, self-optimizing and self-adapting, anticipatory.
  - Definition of Autonomic Components/Services
  - Dynamic Composition of Autonomic Applications
  - Autonomic Middleware Services

- http://automate.rutgers.edu/
Autonomic Computing Architecture

- Based on distributed, component/service-oriented architectural approach
  - Components both provide and consume services
- Autonomic elements (components/services)
  - Responsible for policy-driven self-management of individual components
- Relationships among autonomic elements
  - Based on agreements established/maintained by autonomic elements
  - Governed by policies
  - Give rise to resiliency, robustness, self-management of system
Autonomic Elements: Structure

• Fundamental atom of the architecture
  – Managed element(s)
    • Database, storage system, server, software app, etc.
  – Plus one autonomic manager
• Responsible for:
  – Providing its service
  – Managing its own behavior in accordance with policies
  – Interacting with other autonomic elements
Autonomic Elements: Interactions

- Relationships
  - Dynamic, ephemeral
  - Formed by agreement
    - May be negotiated
  - Full spectrum
    - Peer-to-peer
    - Hierarchical
  - Subject to policies
Autonomic Elements: Composition of Autonomic Systems

- Workload Manager
- Arbiter
- Provisioner
- Planner
- Reputation Authority
- Registry
- Broker
- Network
- Monitor
- Server
- Workload Manager
- Sentinel
- Negotiator
- Registry
- Event Correlator
- Aggregator
- Monitor
- Database
- Server
- Storage
- Monitor
- Network
- Storage
- Database
AutoMate: Architecture

• Key components:
  – Autonomic application framework
  – Decentralized deductive engine
  – P2P content discovery service
  – Associative (content-based) Interactions
  – Dynamic access control engine
Autonomic Oil Well Placement

Contour plots of permeability, pressure contours with 3 wells, 2D profile.

Requires NYxNZ (450) evaluations. Minimum appears here.

VFSA solution: "walk": found after 20 (81) evaluations.
Client configures and launches IPARS Factory and VFSA Optimization peers on resource of choice.

IPARS Factory discovers and initializes VFSA Optimization Service.

VFSA Optimization Service generates new well placement.

One optimal well placement is determined, IPARS Factory launches IPARS run.

IPARS Factory gets initial guess from VFSA Optimization Service launches IPARS instance on resource of choice.

Clients can configure IPARS params.

 Scientists/Engineers collaboratively interact with IPARS.

IPARS connects to VFSA Optimization Services and presents revenue.
Autonomic Computational Science and Engineering
Autonomic Forest Fire Simulation

Predicts fire spread (the speed, direction and intensity of forest fire front) as the fire propagates, based on both dynamic and static environmental and vegetation conditions.
Autonomic Design of Nanomaterials

**INPUT: COMPOSITION/PROCESSING**

**Atomistics:** Simulate non-equilibrium solidification process, crystallization, diffusion and growth

- Interface Kinetics
- Thermodynamics
- Composition
- Atomistic Structure
- Force Fields

**Nanoscale:** Model the evolution and interaction of topologically complex nanosize metastable structures and their effective behavior

- Effective Particle Behavior
- Hardening Mechanisms
- Topological Characteristics
  - of Nanostructured Particles

**Microscale:** Model the collective behavior of assemblies of nanostructured particles

- Strength and Ductility
  - of Nanostuctured Particles

**Optimization:** Optimize metastable nanocomposition based on atomistic, nanoscale and macroscale properties.

**Computational Infrastructure:** To develop autonomic computational infrastructures and runtime management techniques for scalable parallel/distributing computing, automated interaction and data exchange between scales, real time sensing and computational response, collaborative monitoring and steering.

**OUTPUT: OPTIMIZED METASTABLE NANOCOMPOSITES**
Autonomic Living

- Autonomic living: autonomic peers opportunistically interact, coordinate and collaborate to satisfy goals?
  - your car in route to the airport estimates that given weather (from meteorological beacons), road conditions (from on-coming cars), traffic patterns (from the traffic light), warns that you will miss your flight and you will be better off taking the train – the station is coming up – do you want to rebook?
  - in a foreign country, your cell phone enlists a locally advertised GPS and translation service as you try to get directions
  - your clock/PDA estimates drive time to your next appointment and warns you appropriately
  - your eye glasses sends your current prescription as you happen to drive past your doctor or your PDA collects prices for the bike you promised yourself as you drive around
Online monitoring, analysis and self-healing

Vulnerability Index

Normal
Uncertain
Vulnerable

Network Services

Under attacks/Faults

Under normal state

Online Monitoring
Adaptive Analysis
Self Healing Engine

Data mining Statistic Engine

Real Network Running Environment
Summary

• Technology exponentials are changing the shape of scientific investigation & knowledge
  – *More computing, even more data, yet more networking*

• The Grid – An evolving vision
  – *Seamless aggregation*
  – *Seamless composition/interaction*
  – *Autonomic behaviors*
  – *Many technical issues/challenges*
  – *Evolving Grid standards, technologies, infrastructures, applications*
    • *GGF, OGSA, …*

• Project AutoMate – enabling autonomic Grid applications
For More Information

• Global Grid Forum
  – http://www.gridforum.org

• The Globus Project™
  – http://www.globus.org

• Open Grid Services Architecture
  – http://www.globus.org/ogsa

• AutoMate/Autonomic Computing
  – http://automate.rutgers.edu
  – http://www.autonomic-conference.org

• TASSL (Rutgers)
  – http://www.caip.rutgers.edu/TASSL

• M. Parashar
  – parashar@caip.rutgers.edu
The Team

• TASSL Rutgers University
  – Autonomic Computing Research Group
    • Viraj Bhat
    • Manish Agarwal
    • Nanyan Jiang
    • Hua Liu (Maria)
    • Zhen Li (Jenny)
    • Manish Mahajan
    • Vincent Matossian
    • Cristina Schmidt
    • Guangsen Zhang
  – Autonomic Applications Research Group
    • Sumir Chandra
    • Xiaolin Li
    • Dave Roberts
    • Taher Saif
    • Li Zhang

• CS Collaborators
  – HPDC, University of Arizona
    • Salim Hariri
  – Biomedical Informatics, The Ohio State University
    • Tahsin Kurc, Joel Saltz
  – CS, University of Maryland
    • Alan Sussman, Christian Hansen

• Applications Collaborators
  – CSM, University of Texas at Austin
    • Hector Klie, Mary Wheeler
  – IG, University of Texas at Austin
    • Mrinal Sen, Paul Stoffa
  – ASCI/CACR, Caltech
    • Michael Aivazis, Julian Cummings, Dan Meirom
  – CRL, Sandia National Laboratory, Livermore
    • Jaideep Ray, Johan Steensland