My View

Grid Computing: An Overview

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

LRIG, September 30, 2003

Ack: Slides borrowed from presentations by I. Foster & C. Kesselman (Globus), J.C. Kesler (MCNC), C. Goble (U. of Manchester)
Abstract

• The Grid is rapidly emerging as the dominant paradigm for wide area distributed computing. Its goal is to provide a service-oriented infrastructure that leverages standardized protocols and services to enable pervasive access to, and coordinated sharing of geographically distributed hardware, software, and information resources. Grid technologies and solutions are being rapidly developed and deployed by industry and academia and form the basis of the new national (and possibly global) Cyberinfrastructure, and are enabling a new generation of applications that are based on seamless and secure aggregations and interactions. In this talk I will introduce the vision of the Grid, and highlight key underlying technologies, emerging standards, current deployments, and open research issues/challenges.
Outline of my talk

• Grid computing?
  – vision, definitions, motivation, enablers, history, projects, …

• Grid computing issues, technologies, standards
  – requirements/challenges, platforms, GGF, OGSA, …

• Next steps
  – semantic (cognitive) grid, autonomic grid, …

• Summary, more information
Grids, The 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

• New idea?
  • a large part this vision was originally proposed by Fenando Corbato (The Multics Project, 1965, www.multicians.org)
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

Scientific American (Jan-2001)

“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)
Drivers: Evolution of the Scientific/Business Process

• Evolution of the scientific process
  – Pre-electronic
    • Theorize &/or experiment, alone or in small teams; publish paper
  – Post-electronic
    • Construct and mine very large databases of observational or simulation data
    • Develop computer simulations & analyses
    • Exchange information quasi-instantaneously within large, distributed, multidisciplinary teams

• Evolution of business process
  – Pre-Internet
    • Central corporate data processing facility
    • Business processes not typically compute-oriented
  – Post-Internet
    • Enterprise computing is highly distributed, heterogeneous, inter-enterprise (B2B)
    • Outsourcing becomes feasible => service providers of various sorts
    • Business processes increasingly computing- and data-rich

⇒ Need to manage dynamic, distributed infrastructures, services, and applications
⇒ Seamless aggregations and interactions

LRIG, September 30, 2003
The Grid according to The Experts

“Flexible, secure, coordinated resource sharing among dynamic collections of individuals, institutions, and resources.”

From *The Anatomy of the Grid* by Foster, Kesselman and Tuecke

“A grid is all about gathering together resources and making them accessible to users and applications.”

Dr. Andrew Grimshaw, CTO Avaki
The Grid…

“Resource sharing & coordinated problem solving in dynamic, multi-institutional virtual organizations”
The Grid: A Brief History

- Early 90s
  - Gigabit testbeds, metacomputing
- Mid to late 90s
  - Early experiments (e.g., I-WAY), academic software projects (e.g., Globus, Legion), application experiments
- 2002
  - Dozens of application communities & projects
  - Major infrastructure deployments
  - Significant technology base (esp. Globus Toolkit™)
  - Growing industrial interest
  - Global Grid Forum: ~500 people, 20+ countries
Contemporary Grid Projects

• Computer science research
  – *Wide variety of projects worldwide*
  – *Situation confused by profligate use of label*

• Technology development
  – *R&E: Condor, Discover, Globus, EU DataGrid, GriPhyN*
  – *Industrial: significant efforts emerging*

• Infrastructure development
  – *Persistent services as well as hardware*

• Application
  – *Deployment and production application*

• See [www.gridforum.org](http://www.gridforum.org) for a list of projects
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
- Unprecedented
  - scales, complexity, heterogeneity, dynamism, uncertainty, failure
  - unpredictability, lack of guarantees

Unprecedented scales, complexity, heterogeneity, dynamism, uncertainty, failure
unpredictability, lack of guarantees
For Example, why Grid security is hard

- Resources being used may be extremely valuable & the problems being solved extremely sensitive
- Resources are often located in distinct administrative domains
  - Each resource may have own policies & procedures
- The set of resources used by a single computation may be large, dynamic, and/or unpredictable
  - Not just client/server
- It must be broadly available & applicable
  - Standard, well-tested, well-understood protocols
  - Integration with wide variety of tools
Current Grid Platforms -- Market Segments (J.C. Kesler)

One Way to Categorize Grids:
• Toolkits
• Integrated Environments

Or Another Way to Look at Grids:
• Server Aggregation
• Desktop Aggregation
Where Current Platforms Fit in the Market

Desktop Aggregation
- Entropia
- United Devices
- Data Synapse
- Parabon

Server Aggregation
- Platform LSF Multi-Cluster
- Avaki
- IBM Grid Toolbox

- BOINC

- OGSA
- NMI
- Globus
The Early Adopter Market for Grid Technology

- **Private Sector**
  - Pharmaceuticals
  - Banking & Finance
  - Energy

- **Mix of Industry and Academia**
  - Life Sciences
  - Entertainment

- **Public Sector**
  - Academia
  - Government
  - National Labs

- **Desktop Aggregation**

- **Server Aggregation**

*(does anyone want this?)*
Grid Platform Example: Globus Toolkit V2

- Primary development occurred at Argonne National Labs
  - Principals were 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
Globus Toolkit V2 “Pillars”

- Resource Management (GRAM)
- Information Services (MDS)
- Data Management (GASS)
- Grid Security Infrastructure (GSI)
Grid Platform Example: AVAKI

- Original technology came from the Legion project at UVa (which was also used as part of NPACI); principal is Andrew Grimshaw (now CTO)
- Integrated solution - load and run
- Object-oriented architecture
- Data Grid (v3.0) - new architecture meant as the stepping stone to OGSA; implemented with J2EE
- Compute Grid (v2.6) - latest release of Legion-based technology; has compute and data grid integrated
- Comprehensive Grid: 3.0 Data + 2.6 Compute Grids
AVAKI 3.0 Data Grid Architecture

Other grids

interconnect

AVAKI Domain Controller

LDAP (User Info)

Grid Server (metadata)

Grid Server (metadata)

Data Access Server (NFS)

Share Server

Share Server

Share Server

Share Server

/dfm/edu
/data/ncbi
/home/edu
/data/riceblast

/dfm/edu
/local/data
/home/edu
/local/data

/grid
/grid/dfm/edu
/grid/home/edu
/grid/data
/grid/data/ncbi
/grid/data/riceblast

LRIG, September 30, 2003
Standardizing The Grid: The 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
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
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
The Grid Service = Interfaces/Behaviors + Service Data

Service data access
Explicit destruction
Soft-state lifetime

Binding properties:
- Reliable invocation
- Authentication

Implementation

Hosting environment/runtime ("C", J2EE, .NET, ...)

Standard:
- Notification
- Authorization
- Service creation
- Service registry
- Manageability
- Concurrency
+ application-specific interfaces
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
The Next Step: Semantic (Cognitive) Grid

- In a service oriented architecture, how do I?
  - Create, name, manage, discover services?
  - Render resources, data, sensors as services?
  - Negotiate service level agreements?
  - Express & negotiate policy?
  - Organize & manage service collections?
  - Establish identity, negotiate authentication?
  - Manage VO membership & communication?
  - Compose services efficiently?
  - Achieve interoperability?
The Next Step: Semantic (Cognitive) Grid

• Build on the semantic web:
  – The Semantic Web is an extension of the current Web in which information is given a well-defined meaning, better enabling computers and people to work in cooperation. It is the idea of having data on the Web defined and linked in a way that it can be used for more effective discovery, automation, integration and reuse across various applications. The Web can reach its full potential if it becomes a place where data can be processed by automated tools as well as people” - From the W3C Semantic Web Activity statement

• Grid Services + Ontologies + Knowledge Driven Services

• Examples
  – Knowledge driven matchmaking
  – Agent based service composition
  – High-level planning and resource discovery
  – Knowledge based provisioning

www.semanticgrid.org
The Next Step: Semantic (Cognitive) Grid

Richer semantics

Semantic Web

Classical Web

Semantic Grid

Classical Grid

More computation

LRIG, September 30, 2003  
Source: Norman Paton
The Next Step: Autonomic Grids (Motivations)

• Unprecedented
  – scales, complexity, heterogeneity, dynamism, uncertainty, failure 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

• Bottom line
  – the increasing system complexity is reaching a level beyond human ability to manage and secure

• A fundamental change is required in how applications are formulated, composed and managed
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 human body – 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?
    - e.g. FlyPhones, AORO/AutoMate, ROC, ELiza, etc.
- of course, there is a cost
  - lack of controllability, precision, guarantees, comprehensibility, …
AutoMate: Enabling Autonomic Applications (http://automate.rutgers.edu)

• **Objective:**
  - To enable the development of autonomic Grid applications that are context aware and are capable of self-configuring, self-composing, self-optimizing and self-adapting.

• **Overview:**
  - **Definition of Autonomic Components:**
    - definition of programming abstractions and supporting infrastructure that will enable the definition of autonomic components
    - autonomic components provide enhanced profiles or contracts that encapsulate their functional, operational, and control aspects
  - **Dynamic Composition of Autonomic Applications:**
    - mechanisms and supporting infrastructure to enable autonomic applications to be dynamically and opportunistically composed from autonomic components
    - compositions will be based on policies and constraints that are defined, deployed and executed at run time, and will be aware of available Grid resources (systems, services, storage, data) and components, and their current states, requirements, and capabilities
  - **Autonomic Middleware Services:**
    - design, development, and deployment of key services on top of the Grid middleware infrastructure to support autonomic applications
    - a key requirements for autonomic behavior and dynamic compositions is the ability of the components, applications and resources (systems, services, storage, data) to interact as peers
AutoMate: Architecture

- **Key components:**
  - Accord: Autonomic application framework
  - Rudder: Decentralized deductive engine
  - Squid: P2P discovery service
  - SESAME: Dynamic access control engine
  - Pawn: P2P messaging substrate
AutoMate: Architecture

- **AutoMate System Layer:**
  - builds on the Grid middleware and OGSA and extends core Grid services to support autonomic behavior
  - provide specialized services such as peer-to-peer semantic messaging, events and notification
- **AutoMate Component Layer:**
  - addresses the definition, execution and runtime management of autonomic components
  - provides supporting services such as discovery, factory, lifecycle, context, etc.
- **AutoMate Application Layer:**
  - builds on the component and system layers to support the autonomic composition and dynamic (opportunistic) interactions between components
- **AutoMate Engines:**
  - decentralized (peer-to-peer) networks of agents in the system.
    - context-awareness engine composed of context agents and services and provides context information at different levels to trigger autonomic behaviors
    - deductive engine composed of rule agents which are part of the applications, components, services and resources, and provides the collective decision making capability to enable autonomic behavior
    - trust and access control engine composed of access control agents and provides dynamic context-aware control to all interactions in the system
- **AutoMate Portals**
  - provide users with secure, pervasive (and collaborative) access to the different entities
  - using these portals users can access resource, monitor, interact with, and steer components, compose and deploy applications, configure and deploy rules, etc.
Summary

- Technology exponentials are changing the shape of scientific investigation & knowledge
  - More computing, even more data, yet more networking
- The Grid: Resource sharing & coordinated problem solving in dynamic, multi-institutional virtual organizations
  - On-demand, ubiquitous access to computing, data, and services
  - New capabilities constructed dynamically and transparently from distributed services
  - Many technical issues/challenges
- Evolving Grid standards, technologies, infrastructures, applications
  - GGF, OGSA, …
- Next steps
  - Semantic Grid, Autonomic Grid

Ack: Slides borrowed from presentations by I. Foster & C. Kesselman (Globus), J.C. Kesler (MCNC), C. Goble (U. of Manchester)
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
- Semantic Grid
  - http://www.semanticgrid.org
- AutoMate (Autonomic Grid)
  - http://automate.rutgers.edu
- TASSL (Rutgers)
  - http://www.caip.rutgers.edu/TASSL