Grid Computing

Adam Belloum
Institute of Informatics
University of Amsterdam
a.s.z.belloum@uva.nl

High Performance computing Curriculum, Jan 2015 http://www.hpc.uva.nl/

UvA-SURFsara
Grid Computing

DISTRIBUTE THE WEALTH

- Grid Computing
- Distributed computing initiatives

Source: “Hype Cycle for Cloud Computing, 2009,” Gartner
First Generation Grids: Batch Computing

Focus on aggregation of many resources for massively (data-)parallel applications
Second Generation Grids: Service-Oriented Science

• Empower many more users by enabling
  – on-demand access to services

• Grids become an enabling technology for service oriented science (or business)
  – Grid infrastructures host services
  – Grid technologies used to build services

“Service-Oriented Science”, Science, 2005
Second Generation Grids: Service-Oriented Science (Best of Two Worlds)

Open Grid Services Architecture

- Applications on demand
- Secure and universal access
- Business integration

- Resources on demand
- Global Accessibility
- Vast resource scalability

Web Services

Grid Protocols

‘Open Grid Services Architecture Evolution, J.P. Prost, IBM Montpellier, France, Ecole Bruide 2004"
Second Generation Grids: Service-Oriented Science (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
  – Significant implications for how services are managed, named, discovered, and used
eScience: Applications that needs the Grid

• "eScience promotes innovation in collaborative, computationally or data intensive research across all disciplines, throughout the research lifecycle”

• Nowadays Scientific Applications are
  – CPU intensive
  – Produce/process Huge sets of Data
  – Requires access to geographically distributed and expensive instruments
Online Access to Scientific Instruments

Advanced Photon Source

tomographic reconstruction

real-time collection

wide-area dissemination

archival storage

desktop & VR clients with shared controls

DOE X-ray grand challenge: ANL, USC/ISI, NIST, U.Chicago

From the Grid tutorials available at: http://www.globus.org
CPU intensive Science: Optimization problem NUG30

- a quadratic assignment problem (QAP) known as NUG30
  - given a set of n locations and n facilities, the goal is to assign each facility to a location.
  - There are $n!$ possible assignments

- NUG30 proposed in 1968 as a test of computer capabilities, but remained unsolved because of its great complexity.
To solve these problems?

Grid Services
Harness multi-domain distributed

LHC
Application Specific Part
Potential Generic part
Management of comm. & computing

NUG30
Application Specific Part
Potential Generic part
Management of comm. & computing

Online Access
Application Specific Part
Potential Generic part
Management of comm. & computing

“VL-e project” UvA
Grid Services
Harness multi-domain distributed resources
The Grid Vision

“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

“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 paradigm

Principles and mechanisms for dynamic VOs
Leverage service oriented architecture (SOA)
Loose coupling of data and services
Open software, architecture

Computer science  Physics  Astronomy  Biology  Engineering  Biomedicine  Healthcare

1995  2000  2005  2010
The Grid paradigm and information integration

- Make resources usable and useful
- Name resources; move data around
- Make resources accessible over the network

Manage who can do what

Platform services

Data sources:
- Radiology
- Pathology
- Genomics
- Labs
- Medical records
- RHIO
The Grid paradigm and information integration

- Enhance user cognitive processes
- Incorporate into business processes
- Transform data into knowledge
- Integration
- Management
- Publication

Platform services
- Data sources
  - Radiology
  - Pathology
  - Genomics
  - Labs
  - Medical records
  - RHIO

Security and policy
The Grid paradigm and information integration

Value services
- Cognitive support
- Applications
- Analysis
- Integration
- Management
- Publication

Platform services

Data sources
- Radiology
- Pathology
- Genomics
- Labs
- Medical records
- RHIO

Security and policy
Increased functionality, standardization

Emergence of Open Grid Standards

- Globus Toolkit
  - Defacto standard
  - Single implementation
- Internet standards
  - Custom solutions
- Web services, etc.
- Computer science research
- Open Grid Services Arch
  - Real standards
  - Multiple implementations
  - Managed shared virtual systems

“Grid Computing and Scaling Up the Internet” I. Foster, IPv6 Forum, an
The Grid Middleware

• Software toolkit addressing key technical areas
  – Offer a modular “bag of technologies”
  – Enable incremental development of grid-enabled tools and applications
  – Define and standardize grid protocols and APIs

• Focus is on inter-domain issues, not clustering
  – Collaborative resource use spanning multiple organizations
  – Integrates cleanly with intra-domain services
  – Creates a “collective” service layer

Grid Middleware Definition

- Architecture identifies the fundamental system components, specifies purpose and function of these components, and indicates how these components interact with each other.

- Grid architecture is a protocol architecture, with protocols defining the basic mechanisms by which VO users and resources negotiate, establish, manage and exploit sharing relationships.

- Grid architecture is also a service standard-based open architecture that facilitates extensibility, interoperability, portability and code sharing.
"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
Examples of Grid Middleware

- Globus Toolkit (GT4.X) now (GT5.X)
  - www.globus.org
- Legion/Avaki
  - http://www.avaki.com/
  - http://legion.virginia.edu/
- Grid Sun engine
  - http://www.sun.com/service/sungrid/overview.jsp
- Unicore
  - http://www.unicore.org
The Grid Approach and Problem

- Flexible, secure, coordinated resource sharing among dynamic collections of individuals, institutions, and resources

- Enable communities (“Virtual Organizations”) to share geographically distributed resources as they pursue common goals -- assuming the absence of central location, central control, existing trust relationships.

From the Grid tutorials available at: http://www.globus.org
Typical Grid Scenario

Resources

Users
How does the Grid work?

1. **Client** locates resources.
2. Client requests resource info from **Grid Index Info Server**.
3. **Grid Index Info Server** queries the current status of the resource.
4. Client requests resource allocation and process creation.
5. **Gatekeeper** creates the request.
6. **Job Manager** parses the request.
7. **Local Resource Manager** processes the request and defines the processes.
8. Processes are allocated and created.

**Security Infrastructure** provides security for the Grid.

**Site boundary** is the boundary of the Grid system.
The Four components of a Grid infrastructure

- Resource Management
- Information services
- Data Management
- Security
Grid Security: Single Sign On, Delegation
Grid Security: Identity
Grid Security: Authentication
Security cross Grid (V.O.)

Resources providers can delegate some of the authority for maintaining a fine-grained access control policies to communities while still maintaining ultimate control over their resources.

Certification Authority (signs the certificates)

Lauroa Pearlman et al. A Community Authorization Service for Group Collaboration
outline

• e-Science
• Grid approach
• Grid computing
• Programming models for the Grid
• Grid-middleware
• Web Services
• Open Grid Service Architecture (OGSA)
Grid data management

- Data Management System
  - Replica Management
  - Replica Selection
  - Replica Location

- Storage System
- Metadata Repository
- Transport service
- Security

Low level Services (shared with other Grid Components)
A Data selection scenario

Attribute of Desired Data (1)

Logical File Names (2)

Location of One or more replicas (3)

Location of Selected replicas (8)

Sources and Destinations (6)

Candidates transfers

Performances Measurements And Predictions (7)

Sudharshan Vazhkudai “Replica Selection in the Globus Data Grid”
Create a data replica (step 1)

Client

- Initialize user proxy cert.

Proxy

- Create delegated credential resource
- Set termination time

Credential EPR returned

EPR: End Point Reference

Ann Chervenak, Robert Schuler “Globus Data Replication Services” USC ISI
Create a data replica (step 2)

Create Replicator resource
- Pass delegated credential EPR
- Set termination time

Replicator EPR returned

Access delegated credential resource

Ann Chervenak, Robert Schuler “Globus Data Replication Services” USC ISI
Create a data replica (step 3)

Client

Data Rep. Replicator

MDS Index

Delegation Credential

GridFTP Server

Replica Catalog

Replica Catalog

Replica Catalog

Add Replicator resource to Information service Index

Subscribe to Resource replicator changes for
- “Status” R1 [processing, finished]
- “Stage” R1 [discover, transfer, register]

EPR2

Conditions may trigger alerts or other actions (Trigger service not pictured)

Periodically polls Replicator R

Ann Chervenak, Robert Schuler “Globus Data Replication Services” USC ISI
Create a data replica (step 4)

Notification of “Stage” R1 value changed to “discover”

Replicator queries Replica Index to find catalogs that contain desired replica information

Replicator queries Replica Catalog(s) to retrieve mappings from logical name to target name (URL)
Create a data replica (step 5)

Create Transfer resource
- Pass credential EPR2
- Set Termination Time
- Transfer resource EPR3 returned

Access delegated credential resource

Notification of “Stage” R1 value changed to “transfer”

Data transfer between GridFTP Server sites

Periodically poll “ResultStatus” R3 When “Done”, get state information for each file transfer

Setup GridFTP Server transfer of file(s)

Ann Chervenak, Robert Schuler “Globus Data Replication Services” USC ISI
Create a data replica (step 6)

Notification of “Stage” R value changed to “register”

Replicator registers new file mappings in Replica Catalog

Replica Catalog sends update of new replica mappings to the Replica Index

Replica Catalog sends update of new replica mappings to the Replica Index

Replica Catalog

Replica Catalog

Replica Catalog

GridFTP Server

GridFTP Server

MDS Index R2

Delegation Credential R

RFT Transfer R3

Data Rep. Replicator R1
Create a data replica (step 7)

Client inspects Replicator state information for each replication in the request.

Notification of “Status” R1 value changed to “Finished”

Ann Chervenak, Robert Schuler “Globus Data Replication Services” USC ISI
Create a data replica (step 8)

- Client
- Service Container
- Data Rep. Replicator
  - R1
- RFT
  - Transfer
  - R3
- Delegation
  - Credential
  - R
- MDS
  - Index
  - R2

Termination time (set by client) expires eventually

Resources destroyed (Credential, Transfer, Replicator)

Ann Chervenak, Robert Schuler “Globus Data Replication Services” USC ISI
Grid Resource Management

- A user job enters a job queue,
- Scheduler decides when to start the job and resource allocation of the job.
Grid Job Scheduling Scenario

Queue

1.

2.

3.

4...

Scheduler

Schedule

Job-Queue

Compute Resource
What is Grid Computing

• Grid computing is the use of hundreds, thousands, or millions of geographically and organizationally disperse and diverse resources to solve:

=> problems that require more computing power than is available from a single machine or from a local area distributed system
Potential Grid Application

• An application which requires the grid solution is likely distributed (Distributed Computing) and fit in one of the following paradigms:
  – High throughput Computing
  – High performance Computing

Grid computing will be mainly needed for large-scale, high-performance computing.
Distributed Computing

• Distributed computing is a **programming model** in which processing occurs in **many geographically distributed places.**
  – Processing can occur wherever it makes the most sense, whether that is on a server, Web site, personal computer, etc.

• Distributed computing and grid computing either
  – **overlap** or distributed computing is a **subset** of grid computing

From “The Anatomy of the Grid: Enabling Scalable Virtual Organizations” Foster et al
High Throughput Computing

- HTC employs large amounts of computing power for very lengthy periods
  - HTC is needed for doing sensitivity analyses, parametric studies or simulations to establish statistical confidence.

- The features of HTC are
  - Availability of computing power for a long period of time
  - Efficient fault tolerance mechanism

- The key to HTC in grids
  - Efficiently harness the use of all available resources across organizations
High Performance Computing

• HPC brings enormous amounts of computing power to bear over relatively short periods of time.
  – HPC is needed for decision-support or applications under sharp time-constraint, such as weather modeling

• HPC applications are:
  – Large in scale and complex in structure.
  – Real time requirements.
  – Ultimately must run on more than one type of HPC system.
HPC/HTC requirements

- HPC/HTC requires a balance of computation and communication among all resources involved.
  - Managing computation,
  - communication,
  - data locality
Programming Model for the grid

• To achieve petaflop rates on tightly/loosely coupled grid clusters, applications will have to allow:
  – extremely large granularity or produce massive parallelism such that high latencies can be tolerated.

• This type of parallelism, and the performance delivered in a heterogeneous environment, is currently manageable by hand-coded applications.
Programming Model for the grid

• A programming model can be presented in different forms: a language, a library API, or a tool with extensible functionality.

• The successful programming model will
  – enable both high-performance and the flexible composition and management of resources.
  – influence the entire software lifecycle: design, implementation, debugging, operation, maintenance, etc.
  – facilitate the effective use of all manner of development tools, e.g., compilers, debuggers, performance monitors, etc.
Grid Programming Issues

- Portability, Interoperability, and Adaptability
- Discovery
- Performance
- Fault Tolerance
- Security
Programming models

• Shared-state models
• Message passing models
• RPC and RMI models
• Peer to Peer Models
• Web Service Models
• ...

References


• Ian Foster, The physiology of the grid: An open grid services architecture for distributed systems integration, (2002)