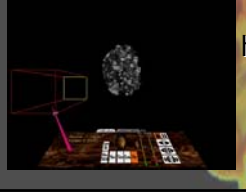
Introduzione al Grid Computing e al Globus Toolkit™

The Globus Project™
USC Information Sciences Institute
Argonne National Laboratory

http://www.globus.org



Sommario

- Introduzione al Grid Computing
- Alcune Definizioni
- La Filosofia della Architettura di Griglia
- Il Globus Toolkit (GT2)
  - Introduzione, Security, Resource Management, Information Services, Data Management
- Open Grid Services Architecture (GT3)

Il Problema della Griglia

- Condivisione flessible, sicura, coordinata condivisione di risorse tra gruppi dinamici di individui, instituzioni e sistemi.
  Da *"The Anatomy of the Grid: Enabling Scalable Virtual Organizations"*
- Permettere a comunità reali o virtuali ("virtual organizations") con obiettivi comuni di condividere risorse distribuite geograficamente - *assumendo l'assenza di…*
  - sito centrale,
  - controllo centrale,
  - completa conoscenza,
  - l'esistenza di relazioni affidabili.

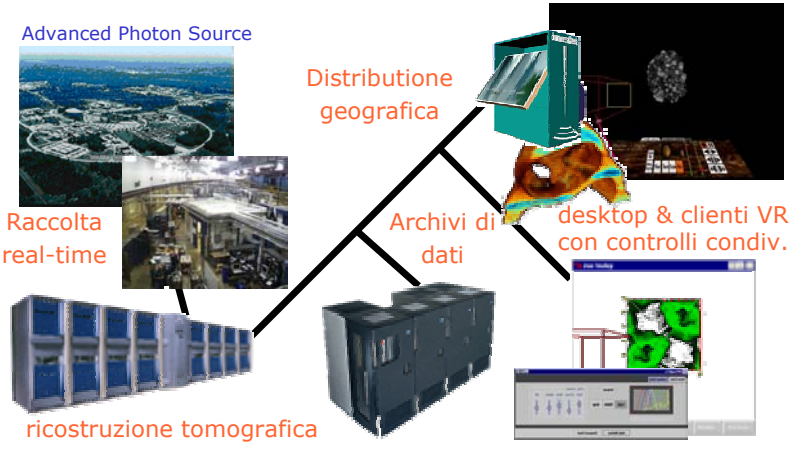
Elementi del Problema

- Condivisione di risorse
  - Computer, memorie, sensori, reti, …
  - Condivisione condizionale: problemi di fiducia, politiche, negoziazione, pagamento, …
- Coordinated problem solving
  - Oltre il client-server: analisi distribuita di dati, elaborazione distribuita, collaborazione, …
- Organizzazioni dinamiche, multi-istituzionali e virtuali
  - Comunità sovrapposte su strutture org. Classiche.
  - Grandi o piccole, statiche o dinamiche.

Perché usare le Griglie?

- Un biochimico usa 10.000 computer per analizzare 100.000 composti in un'ora.
- 1.000 fisici nel mondo usano in maniera integrata alcuni petabytes di dati.
- Ingegnerici civili collaborano per progettare, realizzare e analizzare esperimenti di terremoti.
- Scienziati del clima visualizzano, annotano, e analizzano terabyte di dati di simulazioni.
- Un team di gestione di emergenze integra dati real time, weather modelli di previsione e dati sulla popolazione.

Accesso Online a Strumenti Scientifici



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

Data Grid per la Fisica delle Alte Energie

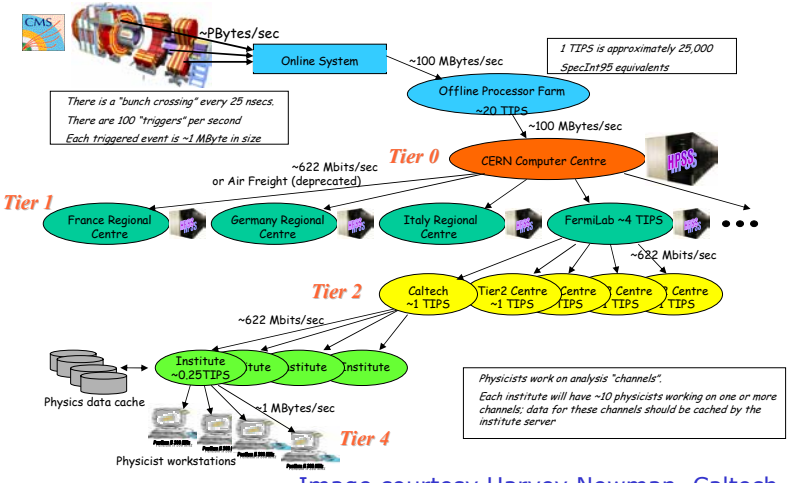
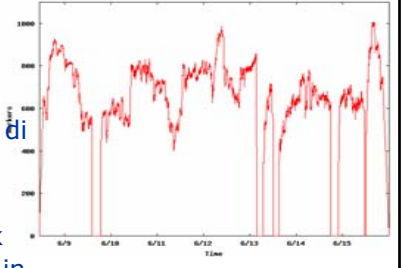


Image courtesy Harvey Newman, Caltech

Matematici Risolvono il NUG30

- Ricerca della soluzione del problema dell'assegnamento quadratico NUG30
- Una collaborazione informale di matematici e informatici
- Condor-G ha gestito 3.46E8 secondi di CPU in 7 gg. (peak 1009 processors) in U.S.A. e in Italia (8 sites)

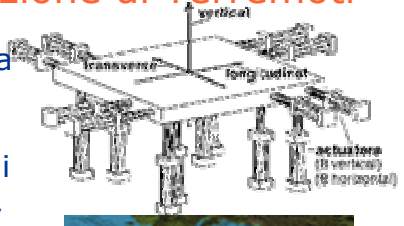


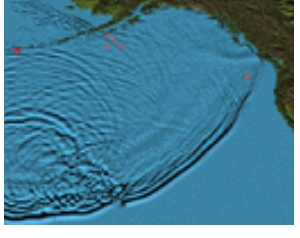
14,5,28,24,1,3,16,15,
10,9,21,2,4,29,25,22,
13,26,17,30,6,20,19,
8,18,7,27,12,11,23

MetaNEOS: Argonne, Iowa, Northwestern, Wisconsin

Rete per la Simulazione di Terremoti

- NEESgrid: infrastruttura nazionale per la collaborazione tra ingegneri civili e l'uso di dispositivi sperimentali, databases, computers, & altro
- Accesso on-demand ad esperimenti, dati, calcolo, archivi, e collaborazioni.





NEESgrid: Argonne, Michigan, NCSA, UIUC, USC

Home Computers Per l'Analisi di Farmaci per l'AIDS

- Comunità=
  - migliaia di utenti di I
  - Philanthropic computing vendor (Entropia)
  - Gruppo di Ricerca (Scripps)
- Obiettivo Comune= avanzamento nella Ricerca e nello studio dell'AIDS

Contesto più Generale

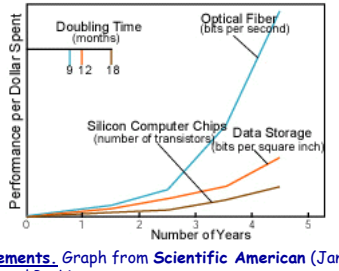
- Il "Grid Computing" ha molto in comune con i maggiori trend industriali
  - Business-to-business, Peer-to-peer, Application Service Providers, Storage Service Providers, Distributed Computing, Internet Computing…
- Problemi comuni non adeguatamente affrontati dalle tecnologie esistenti
  - Requisiti Complicati: "eseguire il programma X sul sito Y conforme alla politica di Comunità P, fornendo l'accesso ai dati in Z secondo la politica Q"
  - High performance: richieste particolari di sistemi avanzati con alte prestazioni.

Perchè Adesso?

- I miglioramenti di legge di Moore nei sistemi di elaborazione produce sistemi finali altamente funzionali.
- Internet e le diverse reti wired o wireless forniscono una connettività globale.
- I cambiamenti nel modo di lavorare in team e orientato alle soluzioni favoriscono questa soluzione.
- Le elevate prestazioni delle reti producono cambiamenti drammatici in termini geometrici e geografici.

Prestazioni Elevate delle Reti

- Prestazioni delle reti e dei calcolatori
  - La velocità dei calcolatori raddoppia ogni 18 mesi
  - La velocità delle reti raddoppia ogni 9 mesi
  - Differenza = un ordine di grandezza ogni 5 anni
- Dal 1986 al2000
  - Computers: x 500
  - Reti: x 340.000
- Dal 2001 al 2010
  - Computers: x 60
  - Reti: x 4000



Moore's Law vs. storage improvements vs. optical improvements. Graph from **Scientific American** (Jan-2001) by Cleo Vilett, source Vined Khoslan, Kleiner, Caufield and Perkins.

Il Globus Project™

- Stretta collaborazione con progetti "reali" di Griglia nella scienza e nell'industria
- Sviluppo e promozione dei protocolli standard e delle interfacce di griglia per permettere interoperabilità ed infrastruttura comune
- Il Globus Toolkit™: Open source, software di base di riferimento per la costruzione dell'infrastruttura e le applicazioni di griglia
  - GT2:
  - GT3: Nuova implementazione basata sui Grid Services (che estendono i Web Services)
- Global Grid Forum: Sviluppo di protocolli standard e API per Grid computing (www.ggf.org)

Principali Progetti di Grid

Nome	URL & Sponsors	Focus
DOE Science Grid	sciencegrid.org DOE Office of Science	Creare una Grid che fornisca l'accesso a risorse & applicazioni nei laboratori del DOE e di università partner.
Earth System Grid (ESG)	earthsystemgrid.org DOE Office of Science	Griglia per l'analisi di dati per modelli climatologici a larga scala
European Union (EU) DataGrid	eu-datagrid.org European Union	Creare e usare una Grid for nella fisica delle alte energie, le scienze ambientali e la bioinformatica

Principali Progetti di Grid

Nome	URL/Sponsor	Focus
Fusion Collaboratory	fusiongrid.org DOE Off. Science	Creare un ambiente collaborativo U.S.A. per ricerche sulla fusione
Globus Project™	globus.org DARPA, DOE, NSF, NASA, Msoft	Ricerca sulle Grid technologies; sviluppo e supporto del Globus Toolkit™; applicazioni e messa in uso (deployment)
Grid Research Integration Dev. & Support Center	grids-center.org NSF	Integrazione, deployment, supporto della NSF Middleware Infrastructure per ricerca e formazione.

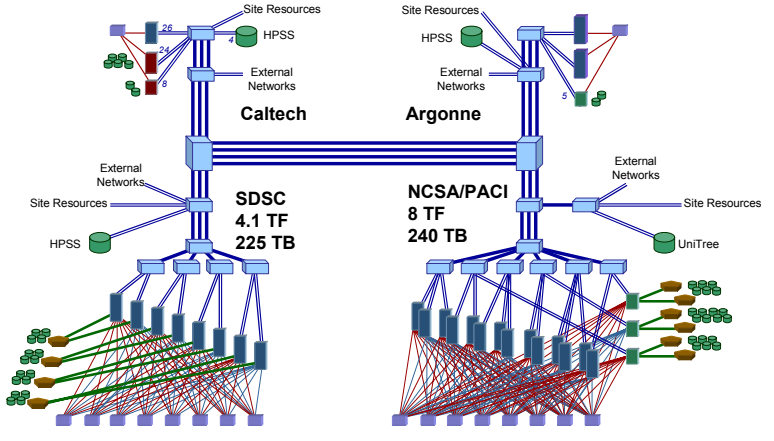
Principali Progetti di Grid

Nome	URL/Sponsor	Focus
Grid Physics Network	griphyn.org NSF	Tecnologia R&S per data analysis in esper. di fisica: ATLAS, CMS, LIGO, SDSS
Information Power Grid	ipg.nasa.gov NASA	Creare e usare una production Grid per aeroscienze e altre missioni NASA
International Virtual Data Grid Laboratory	ivdgl.org NSF	Creare una Data Grid intern. Per permettere esp. su larga scala di tecnologie & applicazioni Grid
Network for Earthquake Eng. Simulation Grid	neesgrid.org NSF	Creare e usare una production Grid per l'ingegneria dei terremoti
Particle Physics Data Grid	ppdg.net DOE Science	Creare e usare una production Grid per data analysis in esperim. di fisica nucleare e delle alte energie.

Principali Progetti di Grid

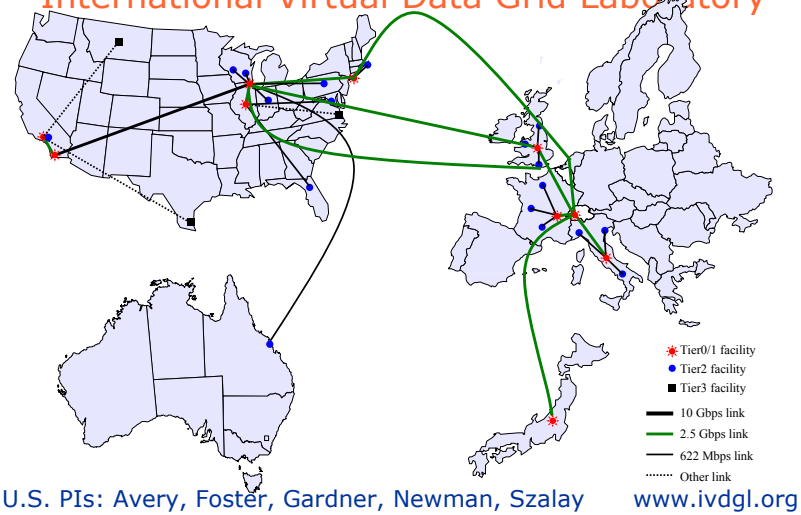
Nome	URL/Sponsor	Focus
TeraGrid	teragrid.org NSF	infrastruttura U.S.A. per la scienza che collega 4 siti a 40 Gb/s
UK Grid Support Center	grid-support.ac.uk U.K. eScience	Centro di supporto per progetti Grid nel Regno Unito

Il 13.6 TF TeraGrid: Elaborazione a 40 Gb/s

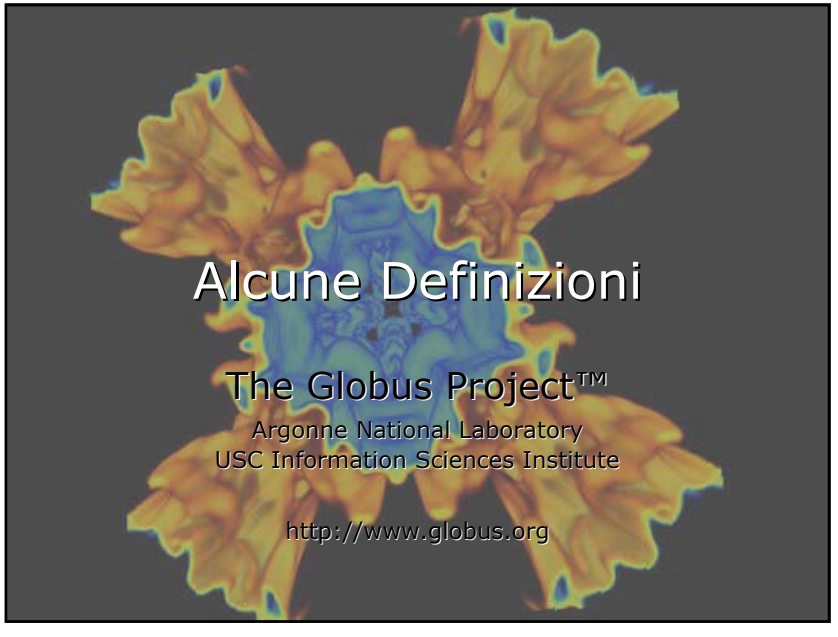


TeraGrid/DTF: NCSA, SDSC, Caltech, Argonne www.teragrid.org

iVDGL: International Virtual Data Grid Laboratory



U.S. PIs: Avery, Foster, Gardner, Newman, Szalay www.ivdgl.org



Alcune Importanti Definizioni

- Risorsa
- Network protocol
- Network enabled service
- Application Programmer Interface (API)
- Software Development Kit (SDK)
- Sintassi

- Non discusse, ma importanti: politiche

Risorsa

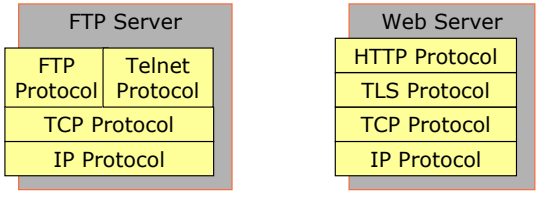
- Una entità da condividere
  - Es., computers, memorie, dati, software
  - Definita in termini di interfacce, non di dispositivi
  - Es. uno scheduler come LSF e PBS definisce una risorsa di calcolo come un cluster
  - Es., Open/close/read/write definiscono accessi ad un file system distribuito come NFS, AFS, DFS.

Network Protocol

- Una descrizione formale di formati di messaggi e un insieme di regole per lo scambio di messaggi.
  - Le regole possono definire sequenze di scambio di messaggi
  - Un protocollo può definire il cambio di stato nel punto finale, es., cambio di stato di un file system
  - I Protocolli possono prevedere più livelli.
- Esempi di protocolli
  - IP, TCP, TLS (era SSL), HTTP, Kerberos

Network Enabled Services

- Implementazione di un protocollo che definisce un insieme di capabilities
  - Il protocollo definisce l'interazione con il servizio
  - Tutti i servizi di rete richiedono protocolli
  - Non tutti i protocolli sono usati per fornire servizi (es. IP, TLS)
- Esempi: FTP e Web servers



Application Programming Interface

- Una specifica di un insieme di routine per facilitare lo sviluppo di applicazioni
  - Si riferiscono alla definizione, non all'implementazione
  - Es., vi sono implementazioni di MPI
- Specifiche spesso legate ad un linguaggio
  - Nome della routine, numero, ordine e tipo degli argomenti; mapping a costrutti del linguaggio
  - Comportamento o funzione della routine
- Esempi
  - GSS API (security), MPI (message passing)

Software Development Kit

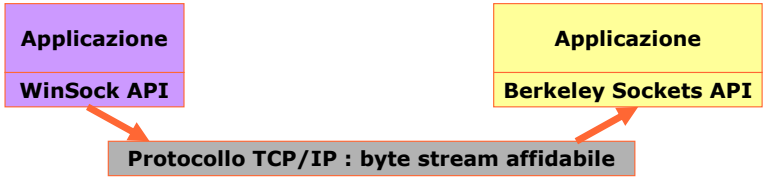
- Una particolare istanziazione di una API
- Un SDK consiste di librerie e strumenti
  - Fornisce una implementazione di una specifica di una API
- Possono esistere diversi SDK per una API
- Esempi di SDK
  - MPICH, Motif Widgets

Sintassi

- Regole per codificare l'informazione, es.
  - XML, Condor ClassAds, Globus RSL
  - X.509 certificate format (RFC 2459)
  - Cryptographic Message Syntax (RFC 2630)
- Distinta dai protocolli
  - Una sintassi puo' essere usata da molti protocolli (e.g., XML) e utile per molti scopi.
- Si possono avere sintassi a più livelli
  - Es., Condor ClassAds -> XML -> ASCII
  - Importante capire la stratificazione quando si paragonano e valutano più sintassi.

Un Protocollo può avere più API

- Le API TCP/IP includono i sockets BSD, Winsock, System V streams, …
- Il protocollo fornisce interoperability: programmi che fanno uso di API differenti possono scambiarsi informazioni.
- Io non ho bisogno di conoscere quali API sta usando un utente remoto.



Una API può avere più Protocolli

- MPI fornisce la portabilità: qualsiasi programma corretto compila e "gira" su una piattaforma.
- Non fornisce interoperabilità: tutti i processi devono comunicare tramite la stessa SDK
  - E.g., MPICH and LAM versions of MPI

API e Protocolli sono Entrambi Importanti

- API/SDK standard sono importanti
  - Permettono la *portabilità* delle applicazioni
  - Ma senza protocolli standard, interoperabilità è difficile (ogni SDK parla con ogni protocollo?)
- Protocolli standard sono importanti
  - Permettono *interoperabilità* tra siti diversi
  - Permettono una struttura condivisa
  - Ma senza API/SDK standard, la portabilità delle applicazioni è difficile (macchine differenti fanno uso di un protocollo in maniera differente)

Oggi: Focus sul Problema dei Sistemi

- Il problema dei sistemi
  - Facilitare l'uso coordinato di risorse diverse
  - Facilitare condivisione dell'infrastruttura : es., autorità di certificazione, info services
  - Richiede sistemi: protocolli, servizi
  - Es., porte/servizi/protocolli per accedere informazioni e allocare risorse
- Il problema della programmazione
  - Facilitare lo sviluppo di applic. Sofisticate.
  - Facilitare il code sharing
  - Richiede ambienti di programmazione: APIs, SDKs, tools

Problema dei Sistemi : Meccanismi di Condivisione di Risorse che...

- Affrontino problemi di sicurezza e politiche dei proprietari e degli utenti.
- Siano abbastanza flessibili per gestire risorse di tipo diverso e modalità di cooperazione diverse.
- Gestiscano elevate risorse, programmi e utenti.
- Operino efficientemente nel gestire grandi moli di dati e di computazione.

Aspetti del Problema dei Sistemi

1) Necessità di interoperabilità quando differenti gruppi condividono risorse
   - Diverse componenti, politiche, meccanismi
   - Es., notioni standard di identità, mezzi di comunicazione, descrizione di risorse
2) Necessità servizi di infrastruttura condivisi per evitare sviluppi e configurazioni ripetute
   - Es., una porta/servizio/protocollo per accesso remoto all'elaborazione, non uno per tool/applicazione
   - Es., Autorità di Certificazione : costose
- Necessità comune per protocolli e servizi

Quindi, una Vista Protocol-Oriented dell'Architettura di Grid orientata a …

- Sviluppo di protocolli e servizi di Grid
  - Accesso "Protocol-mediated" a risorse remote
  - Nuovi servizi: es., brokering di risorse
  - "On the Grid" = uso di protocolli Intergrid
  - Essenzialmente (estensioni di) protocolli esistenti
- Sviluppo di Grid APIs & SDKs
  - Interfacce a protocolli e servizi di Grid
  - Facilitare lo sviluppo di applicazioni attraverso astrazioni di più alto livello
- Il modello (largamente vincente) è Internet.

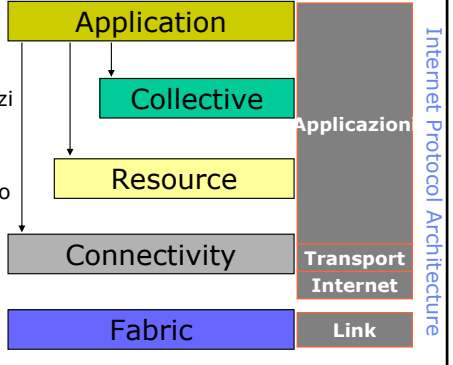
Architettura di Grid a Livelli
(per analogia con l'Architettura di Internet)

"Coordinare risorse multiple": servizidi infrastruttura ubiqui, servizi distribuiti application-specific

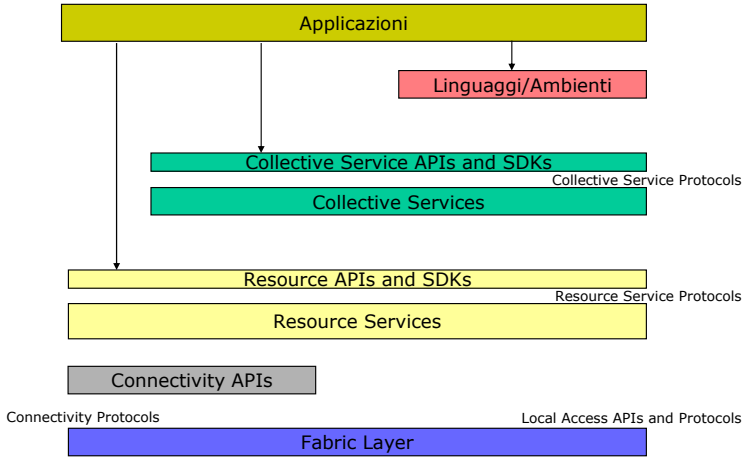
"Condividere risorse singole": negoziare l'accesso, controllare l'uso

"Parlare alle cose": comunicazione (protocolli Internet) & security

"Controllare le cose localmente": Accesso a, e controllo di, risorse



Protocolli, Servizi, e API ad Ogni Livello

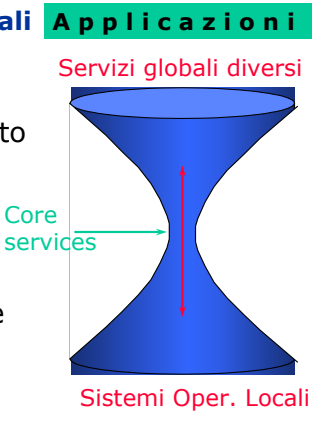


Aspetti Importanti

- Costruita sui protocolli e i servizi di Internet
  - Comunicazione, routing, risolutione dei nomi, ecc.
- "Stratificazione" qui è concettuale, non implica vincoli su chi può chiamare cosa
  - Protocolli/servizi/API/SDK sono, largamente auto-contenuti
  - Alcune cose sono fondamentali: es., comunicazione e sicurezza
  - E' vantaggioso per funzioni di più alto livello usare funzioni comuni di più basso livello.

Il Modello a Clessidra

- **Focus sui problemi architetturali**
  - Un insieme di core services come infrastruttura di base
  - Costruzione di solutioni di alto livello, domain-specific
- **Principi di progettazione**
  - Mantenere basso il costo di participazione
  - Permettere il controllo locale
  - Supportare l'adattamento
  - Modello "clessidra IP"



Livello di Connettività Protocolli & Servizi

- Comunicazione
  - protocolli Internet : IP, DNS, routing, ecc.
- Sicurezza: Grid Security Infrastructure (GSI)
  - Autenticazione uniforme, autorizzazione, e meccanismi di protezione di messaggi in ambienti multi-istituzioni
  - Singola iscrizione, delega, mapping dell'identità
  - Tecnologia a Chiave Pubblica, SSL, X.509, GSS-API
  - Infrastruttura di Supporto : Certificate Authorities, gestione di certificati & chiavi, …

GSI: www.gridforum.org/security

Livello di Risorse Protocolli & Servizi

- Grid Resource Allocation Mgmt (GRAM)
  - Allocazione Remota, prenotazione, monitoraggio, controllo delle risorse di calcolo
- Protocollo GridFTP (estenzioni FTP)
  - Accesso a dati e trasporto ad alte prestazioni
- Grid Resource Information Service (GRIS)
  - Accesso a informazioni di struttura e di stato
- Network reservation, monitoring, controllo
- Tutto costruito sul livello di connettività: GSI & IP

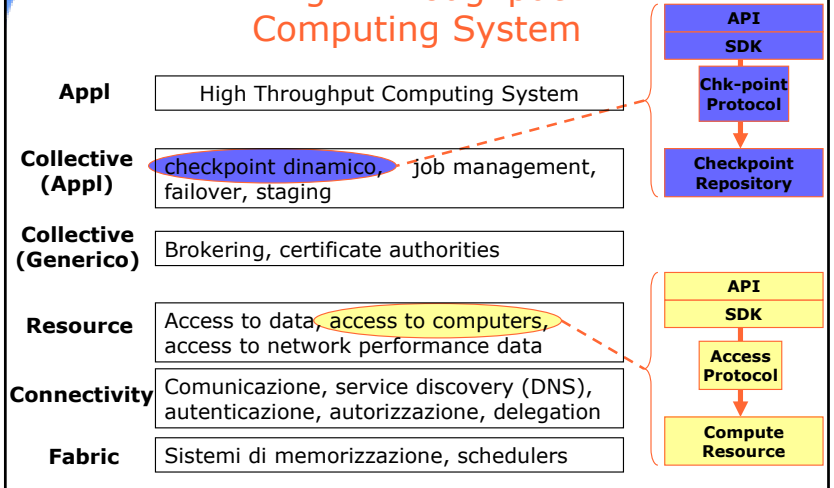
GridFTP: www.gridforum.org
GRAM, GRIS: www.globus.org

Livello Collective Protocolli & Servizi

- Index server (es. Monitoring and Discovery Service)
  - Viste personalizzate su collezioni di risorse dinamiche assemblate da una comunità
- Resource brokers (e.g., Condor Matchmaker)
  - Scoperta e allocazione di risorse
- Servizi di Replica Location e Management
- Servizi di gestione di Metadati
- Servizi di Co-reservation and co-allocation
- Servizi di Gestione di Workflow
- Ecc.

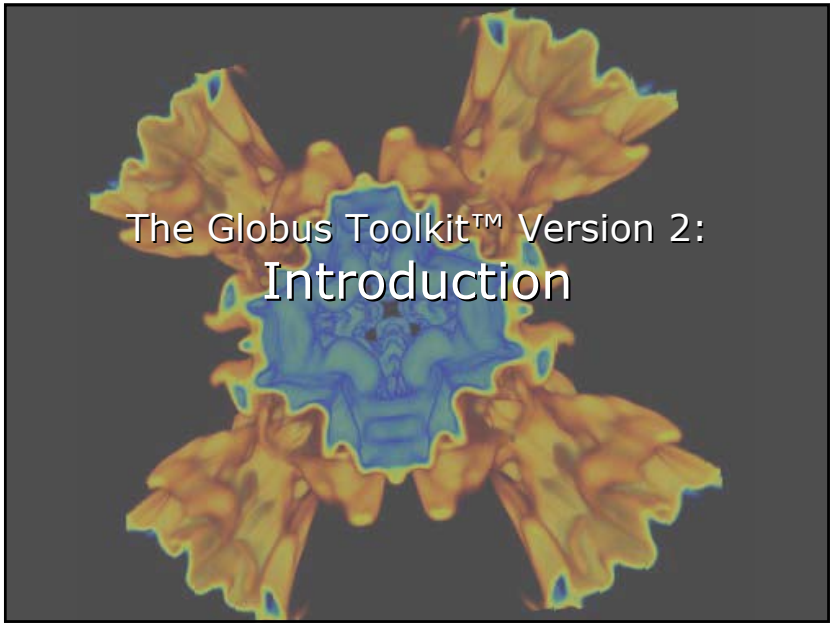
Condor: www.cs.wisc.edu/condor

Esempio: High-Throughput Computing System



Esempio: Grid Services per Applicazioni Data-Intensive

Applicaz.	Applicazioni Data Grid Specifiche per Disciplina
Collective (Appl)	Controllo di Coerenza, replica selection, task management, virtual data catalog, virtual data code catalog, …
Collective (Generic)	Replica catalog, replica management, co-allocazione, certificate authorities, metadata catalogs,
Resource	Accesso ai dati, accesso ai computers, accesso a dati di performance di rete, …
Connect	Comunicazione, service discovery (DNS), autenticazione, autorizzazione, delegation
Fabric	Sistemi di memorizzazione, clusters, reti, cache di rete, .



Globus Toolkit™ Version 2

- A software toolkit addressing key technical problems in the development of Grid enabled tools, services, and applications
  - Offer a modular "bag of technologies"
  - Enable *incremental* development of grid-enabled tools and applications
  - Implement standard Grid protocols and APIs
  - Make available under liberal open source license

Four Main Components

- Security
- Information Management
- Resource Management
- Data Management

General Approach

- Define Grid protocols & APIs
  - Protocol-mediated access to remote resources
  - Integrate and extend existing standards
  - "On the Grid" = speak "Intergrid" protocols
- Develop a reference implementation
  - Open source Globus Toolkit
  - Client and server SDKs, services, tools, etc.
- Grid-enable wide variety of tools
  - Globus Toolkit, FTP, SSH, Condor, SRB, MPI, …
- Learn through deployment and applications

Four Key Protocols

- The Globus Toolkit™ Version 2 centers around four key protocols
  - Connectivity layer:
    - *Security*: Grid Security Infrastructure (GSI)
  - Resource layer:
    - *Resource Management*: Grid Resource Allocation Management (GRAM)
    - *Information Services*: Grid Resource Information Protocol (GRIP)
    - *Data Transfer*: Grid File Transfer Protocol (GridFTP)

Security Terminology

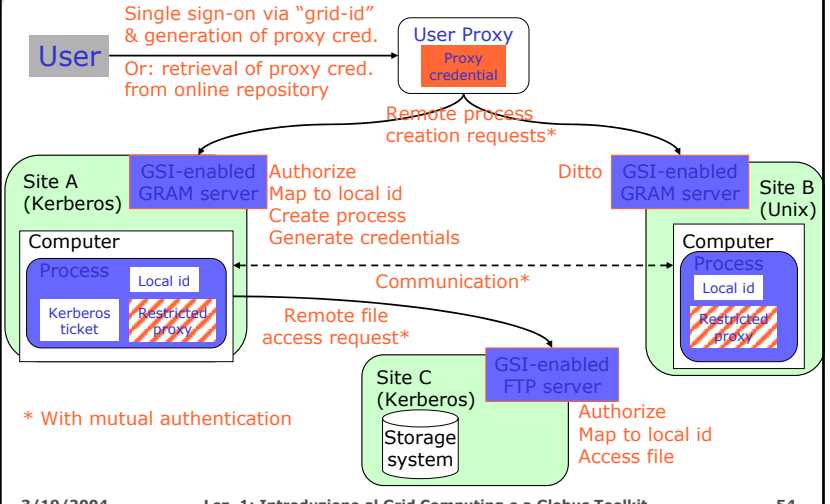
- Authentication: Establishing identity
- Authorization: Establishing rights
- Message protection
  - Message integrity
  - Message confidentiality
- Non-repudiation
- Digital signature
- Accounting
- Certificate Authority (CA)

Why Grid Security is Hard

- Resources being used may be valuable & the problems being solved sensitive
- Resources are often located in distinct administrative domains
  - Each resource has own policies & procedures
- Set of resources used by a single computation may be large, dynamic, and unpredictable
  - Not just client/server, requires delegation
- It must be broadly available & applicable
  - Standard, well-tested, well-understood protocols; integrated with wide variety of tools

GSI in Action

"Create Processes at A and B that Communicate & Access Files at C"



Grid Security Requirements

User View

1) Easy to use
2) Single sign-on
3) Run applications
   ftp,ssh,MPI,Condor,Web,…
4) User based trust model
5) Proxies/agents (delegation)

Resource Owner View

1) Specify local access control
2) Auditing, accounting, etc.
3) Integration w/ local system
   Kerberos, AFS, license mgr.
4) Protection from compromised resources

Developer View

API/SDK with authentication, flexible message protection, flexible communication, delegation, ...
Direct calls to various security functions (e.g. GSS-API)
Or security integrated into higher-level SDKs:
E.g. GlobusIO, Condor-G, MPICH-G2, HDF5, etc.

Grid Security Infrastructure (GSI)

- Extensions to standard protocols & APIs
  - Standards: SSL/TLS, X.509 & CA, GSS-API
  - Extensions for single sign-on and delegation
- Globus Toolkit reference implementation of GSI
  - SSLeay/OpenSSL + GSS-API + SSO/delegation
  - Tools and services to interface to local security
  - Tools for credential management
    - > Login, logout, etc.
    - > Smartcards
    - > MyProxy: Web portal login and delegation
    - > K5cert: Automatic X.509 certificate creation

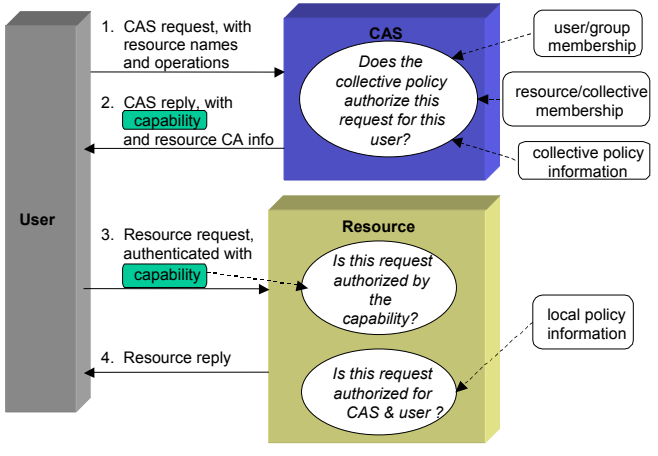
Other Globus Security Work

- Protection against compromised resources
  - Restricted delegation, smartcards
- Standardization
- Scalability in numbers of users & resources
  - Credential management
  - Online credential repositories ("MyProxy")
  - Account management
- Authorization
  - Policy languages
  - Community authorization

Community Authorization Service

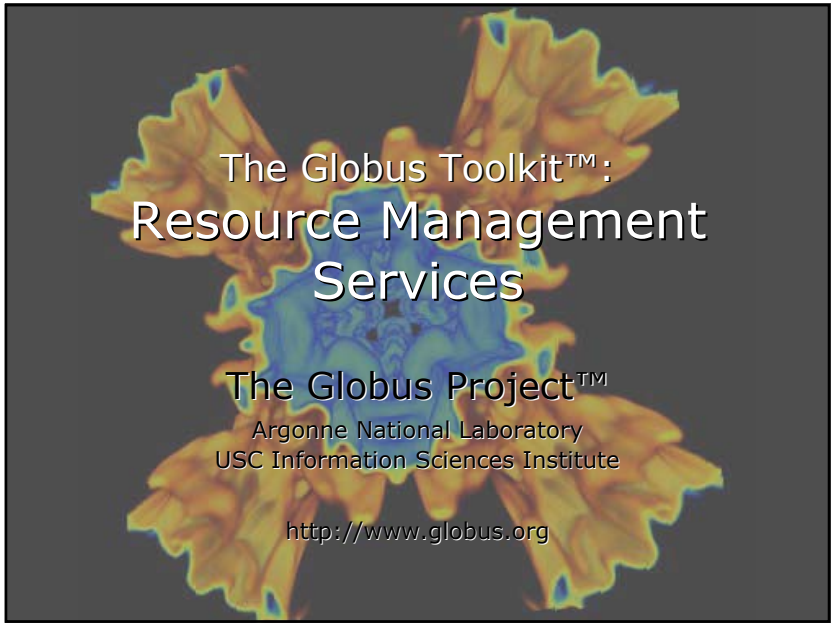
- Question: How does a large community grant its users access to a large set of resources?
  - Should minimize burden on both the users and resource providers
- Community Authorization Service (CAS)
  - Community negotiates access to resources
  - Resource outsources some authorization to CAS
  - CAS handles user registration, group membership…
  - User who wants access to resource asks CAS for a capability credential
  - Resources can also do local access control

Community Authorization



Security Summary

- GSI successfully addresses wide variety of Grid security issues
- Broad acceptance, deployment, integration with tools
- Standardization on-going in IETF & GGF
- Community Authorization Service to address community-based allocation of resources
  - Continuing development



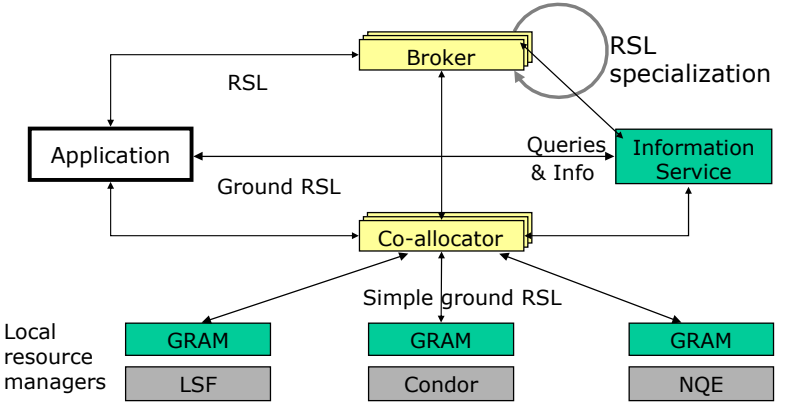
The Challenge

- Enabling secure, controlled remote access to heterogeneous computational resources and management of remote computation
  - Authentication and authorization
  - Resource discovery & characterization
  - Reservation and allocation
  - Computation monitoring and control
- Addressed by new protocols & services
  - GRAM protocol as a basic building block
  - Resource brokering & co-allocation services
  - GSI for security, MDS for discovery

Resource Management

- The Grid Resource Allocation Management (GRAM) protocol and client API allows programs to be started on remote resources, despite local heterogeneity
- Resource Specification Language (RSL) is used to communicate requirements
- A layered architecture allows application-specific resource brokers and co-allocators to be defined in terms of GRAM services
  - Integrated with Condor, PBS, MPICH-G2, …

Resource Management Architecture



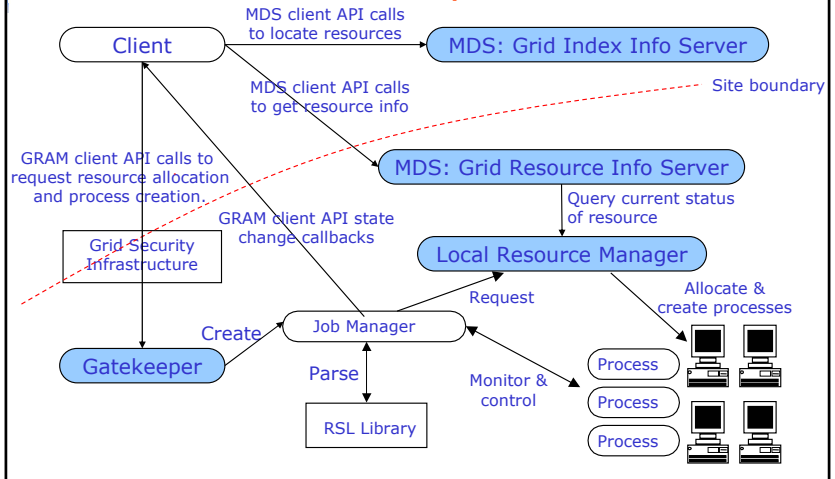
Resource Specification Language

- Common notation for exchange of information between components
  - Syntax similar to MDS/LDAP filters
- RSL provides two types of information:
  - Resource requirements: Machine type, number of nodes, memory, etc.
  - Job configuration: Directory, executable, args, environment
- Globus Toolkit provides an API/SDK for manipulating RSL

Globus Toolkit Version 2 Implementation

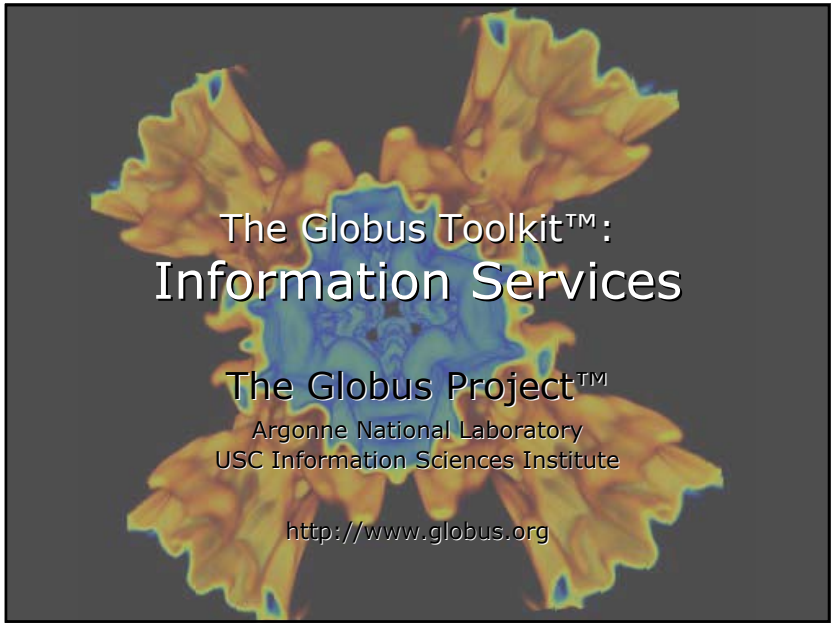
- Gatekeeper
  - Single point of entry
  - Authenticates user, maps to local security environment, runs service
  - In essence, a "secure inetd"
- Job manager
  - A gatekeeper service
  - Layers on top of local resource management system (e.g., PBS, LSF, etc.)
  - Handles remote interaction with the job

GRAM Components



Co-allocation

- Simultaneous allocation of a resource set
  - Handled via optimistic co-allocation based on free nodes or queue prediction
  - In the future, advance reservations will also be supported (already in prototype)
- Globus APIs/SDKs support the co-allocation of specific multi-requests
  - Uses a Globus component called the Dynamically Updated Request Online Co-allocator (DUROC)



Grid Information Services

- System information is critical to operation of the grid and construction of applications
  - What resources are available?
    - > Resource discovery
  - What is the "state" of the grid?
    - > Resource selection
  - How to optimize resource use
    - > Application configuration and adaptation?
- We need a general information infrastructure to answer these questions

Examples of Useful Information

- Characteristics of a compute resource
  - IP address, software available, system administrator, networks connected to, OS version, load
- Characteristics of a network
  - Bandwidth and latency, protocols, logical topology
- Characteristics of the Globus infrastructure
  - Hosts, resource managers

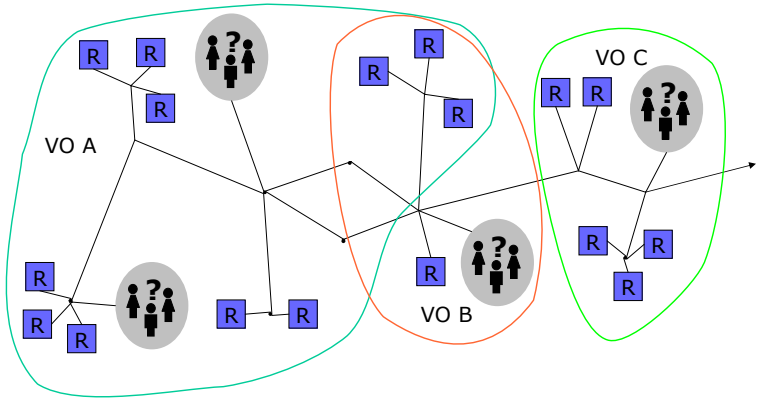
Grid Information: Facts of Life

- Information is always old
  - Time of flight, changing system state
  - Need to provide quality metrics
- Distributed state hard to obtain
  - Complexity of global snapshot
- Component will fail
- Scalability and overhead
- Many different usage scenarios
  - Heterogeneous policy, different information organizations, etc.

Grid Information Service

- Provide access to static and dynamic information regarding system components
- A basis for configuration and adaptation in heterogeneous, dynamic environments
- Requirements and characteristics
  - Uniform, flexible access to information
  - Scalable, efficient access to dynamic data
  - Access to multiple information sources
  - Decentralized maintenance

The GIS Problem: Many Information Sources, Many Views



What is a Virtual Organization?

- Facilitates the workflow of a group of users across multiple domains who share (some of) their resources to solve particular classes of problems
- Collates and presents information about these resources in a uniform view

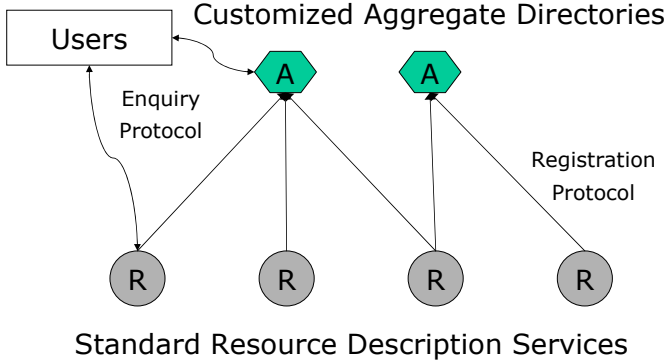
Two Classes Of Information Servers

- Resource Description Services
  - Supplies information about a specific resource (e.g. Globus 1.1.3 GRIS).
- Aggregate Directory Services
  - Supplies collection of information which was gathered from multiple GRIS servers (e.g. Globus 1.1.3 GIIS).
  - Customized naming and indexing

Information Protocols

- Grid Resource Registration Protocol
  - Support information/resource discovery
  - Designed to support machine/network failure
- Grid Resource Inquiry Protocol
  - Query resource description server for information
  - Query aggregate server for information
  - LDAP V3.0 in Globus 1.1.3

GIS Architecture



Monitoring and Discovery Service (MDS)

- Use LDAP as Inquiry
- Access information in a distributed directory
  - Directory represented by collection of LDAP servers
  - Each server optimized for particular function
- Directory can be updated by:
  - Information providers and tools
  - Applications (i.e., users)
  - Backend tools which generate info on demand
- Information dynamically available to tools and applications

Two Classes Of MDS Servers

- Grid Resource Information Service (GRIS)
  - Supplies information about a specific resource
  - Configurable to support multiple information providers
  - LDAP as inquiry protocol
- Grid Index Information Service (GIIS)
  - Supplies collection of information which was gathered from multiple GRIS servers
  - Supports efficient queries against information which is spread across multiple GRIS server
  - LDAP as inquiry protocol

Grid Resource Information Service

- Server which runs on each resource
  - Given the resource DNS name, you can find the GRIS server (well known port = 2135)
- Provides resource specific information
  - Much of this information may be dynamic
    - > Load, process information, storage information, etc.
    - > GRIS gathers this information on demand
- "White pages" lookup of resource information
  - Ex: How much memory does machine have?
- "Yellow pages" lookup of resource options
  - Ex: Which queues on machine allows large jobs?

Grid Index Information Service

- GIIS describes a class of servers
  - Gathers information from multiple GRIS servers
  - Each GIIS is optimized for particular queries
    - > Ex1: Which Alliance machines are >16 process SGIs?
    - > Ex2: Which Alliance storage servers have >100Mbps bandwidth to host X?
  - Akin to web search engines
- Organization GIIS
  - The Globus Toolkit ships with one GIIS
  - Caches GRIS info with long update frequency
    - > Useful for queries across an organization that rely on relatively static information (Ex1 above)
- Can be merged into GRIS

the globus project
www.globus.org

Data Management Problem

- "Enable a geographically distributed community [of thousands] to pool their resources in order to perform sophisticated, computationally intensive analyses on Petabytes of data"
- Note that this problem:
  - Is common to many areas of science
  - Overlaps strongly with other Grid problems
  - Sometimes term "data grid" is used, but this is a general grid problem

Requirements for Grid Data Management

- Terabytes or petabytes of data
  - Often read-only data, "published" by experiments
  - Other systems need to maintain data consistency
- Large data storage and computational resources shared by researchers around the world
  - Distinct administrative domains
  - Respect local and global policies governing how resources may be used
- Access raw experimental data
- Run simulations and analysis to create "derived" data products

Requirements for Grid Data Management (Cont.)

- Locate data
  - Record and query for existence of data
- Data access based on metadata
  - High-level attributes of data
- Support high-speed, reliable data movement
  - E.g., for efficient movement of large experimental data sets
- Support flexible data access
  - E.g., databases, hierarchical data formats (HDF), aggregation of small objects
- Data Filtering
  - Process data at storage system before transferring

Requirements for Grid Data Management (Cont.)

- Planning, scheduling and monitoring execution of data requests and computations
- Management of data replication
  - Register and query for replicas
  - Select the best replica for a data transfer
- Security
  - Protect data on storage systems
  - Support secure data transfers
  - Protect knowledge about existence of data
- Virtual data
  - Desired data may be stored on a storage system ("materialized") or created on demand

Grids for High Energy Physics

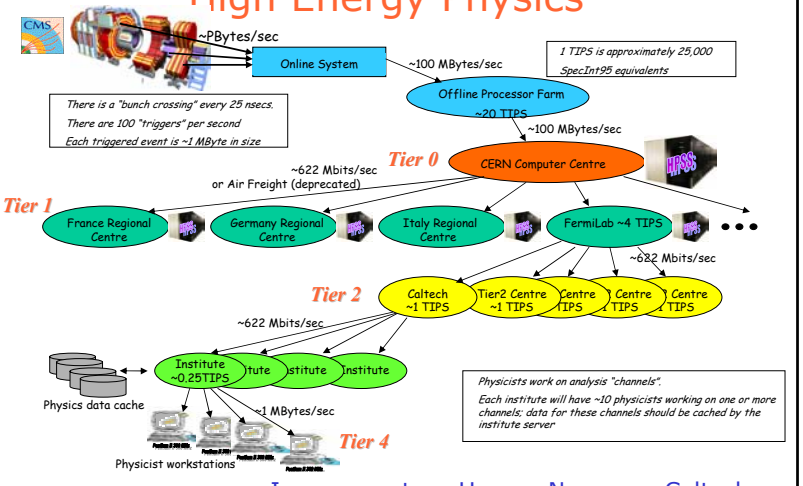


Image courtesy Harvey Newman, Caltech

Globus Toolkit Data Components

- GridFTP Data Transport Protocol
- Replica Location Service
- Metadata Catalog Service

GridFTP

- Data-intensive grid applications need to transfer and replciate large data sets (terabytes, petabytes)
- GridFTP Features:
  - Third party (client mediated) transfer
  - Parallel transfers
  - Striped transfers
  - TCP buffer optimizations
  - Grid security

GridFTP: Basic Approach

- FTP protocol is defined by several IETF RFCs
- Start with most commonly used subset
  - Standard FTP: get/put etc., 3rd-party transfer
- Implement standard but often unused features
  - GSS binding, extended directory listing, simple restart
- Extend in various ways, while preserving interoperability with existing servers
  - Striped/parallel data channels, partial file, automatic & manual TCP buffer setting, progress monitoring, extended restart

GridFTP Implementation

- The GT2 GridFTP is based on the wuftpd server and client
- Important feature is separation of control and data channels
- GridFTP is a Command Response Protocol
  - Issue a command
  - Get only responses to that command until it is completed
  - Then can issue another command

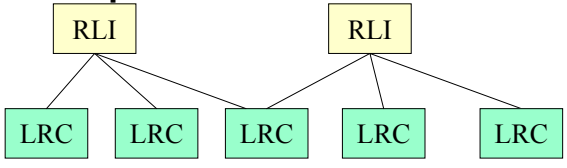
Replica Management in Grids

- Data intensive applications
  - Produce Terabytes or Petabytes of data
- Replicate data at multiple locations
  - Fault tolerance
  - Performance: avoid wide area data transfer latencies, achieve load balancing
- Issues:
  - Locating replicas of desired files
  - Creating new replicas
  - Scalability
  - Reliability

A Replica Location Service

- **A Replica Location Service (RLS)** is a distributed registry service that records the locations of data copies and allows discovery of replicas
- Maintains mappings between *logical* identifiers and *target names*
  - Physical targets: Map to exact locations of replicated data
  - Logical targets: Map to another layer of logical names, allowing storage systems to move data without informing the RLS
- RLS was designed and implemented in a collaboration between the Globus project and the DataGrid project

Replica Location Indexes



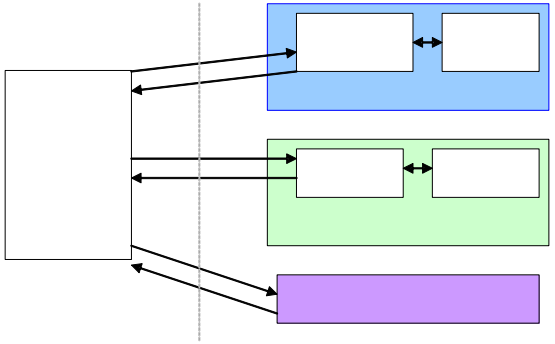
Local Replica Catalogs

- LRCs contain consistent information about logical-to-target mappings on a site
- RLIs nodes aggregate information about LRCs
- Soft state updates from LRCs to RLIs: relaxed consistency of index information, used to rebuild index after failures
- Arbitrary levels of RLI hierarchy

Metadata Services for Cataloguing and Discovery

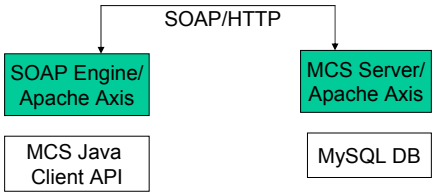
- Metadata is information that describes data sets
- Metadata Services
  - Store metadata attributes according to a specified schema
  - Answer queries for discovery of data with desired attributes
- Two types of metadata services
  - Distinguish between *logical* metadata and *physical* metadata
- Metadata Catalog Service
  - Stores logical metadata that describes contents of files and collections
  - Logical metadata is independent of a particular physical instance, applies to all replicas
  - Variables, annotations, some provenance information

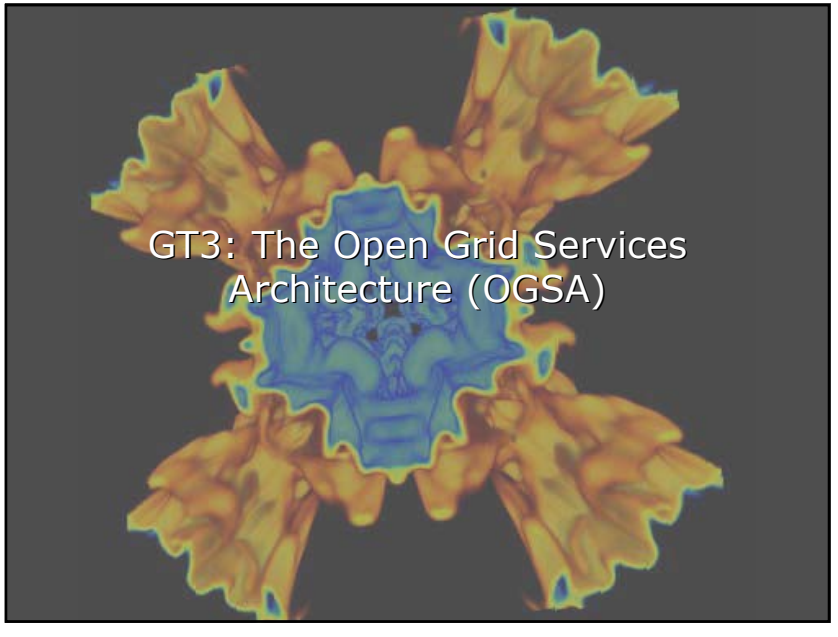
Typical Use of Data Services in Grids



MCS Data Model and Implementation

- Logical files, logical collections and logical views
- May associate pre-defined or user-defined attributes with files, collections or views
- Prototype is a centralized service based on open source web service and database technology





Globus Toolkit: Evaluation (+)

- Good technical solutions for key problems, e.g.
  - Authentication and authorization
  - Resource discovery and monitoring
  - Reliable remote service invocation
  - High-performance remote data access
- This & good engineering is enabling progress
  - Good quality reference implementation, multi-language support, interfaces to many systems, large user base, industrial support
  - Growing community code base built on tools

Globus Toolkit: Evaluation (-)

- Protocol deficiencies, e.g.
  - Heterogeneous basis: HTTP, LDAP, FTP
  - No standard means of invocation, notification, error propagation, authorization, termination, …
- Significant missing functionality, e.g.
  - Databases, sensors, instruments, workflow, …
  - Virtualization of end systems (hosting envs.)
- Little work on total system properties, e.g.
  - Dependability, end-to-end QoS, …
  - Reasoning about system properties

"Web Services"

- Increasingly popular standards-based framework for accessing network applications
  - W3C standardization; Microsoft, IBM, Sun, others
- WSDL: Web Services Description Language
  - Interface Definition Language for Web services
- SOAP: Simple Object Access Protocol
  - XML-based RPC protocol; common WSDL target
- WS-Inspection
  - Conventions for locating service descriptions
- UDDI: Universal Desc., Discovery, & Integration
  - Directory for Web services

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
  - In fact, much of our work is concerned with the management of service instances

OGSA Design Principles

- Service orientation to virtualize resources
  - Everything is a service
- From Web services
  - Standard interface definition mechanisms: multiple protocol bindings, local/remote transparency
- From Grids
  - Service semantics, reliability and security models
  - Lifecycle management, discovery, other services
- Multiple "hosting environments"
  - C, J2EE, .NET, …

OGSA Service Model

- System comprises (a typically few) persistent services & (potentially many) transient services
  - Everything is a service
- OGSA defines basic behaviors of services: fundamental semantics, life-cycle, etc.
- Key issues:
  - Globally unique Grid Service Handle
  - Dynamic service creation (factories)
  - Lifetime management
  - Service discovery
  - Service data elements: associate state with service during its lifetime
  - Query service data elements
  - Subscription/notification

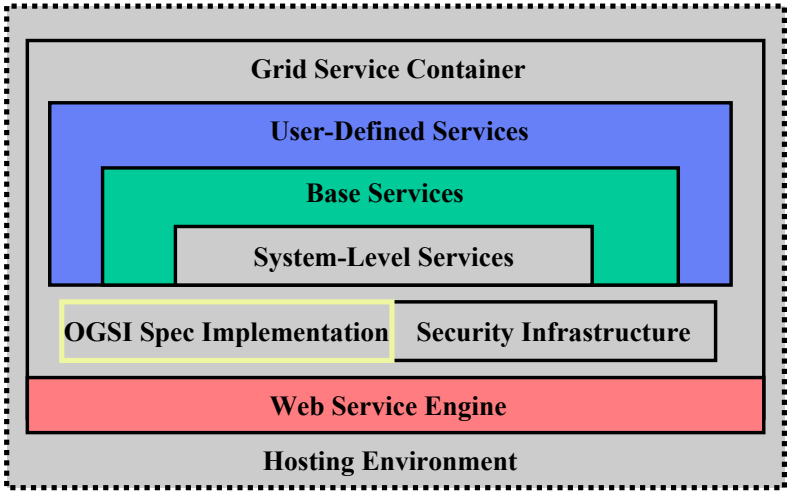
OGSA Development

- Standardization via the Global Grid Forum
  - Focus on RF licensing
- Wide industry interest
  - IBM, Sun, HP, SGI, Microsoft, Veritas, Oracle, …
- Open source reference implementation via Globus project
  - GT3.0 Alpha released in January
- Will be commercial products

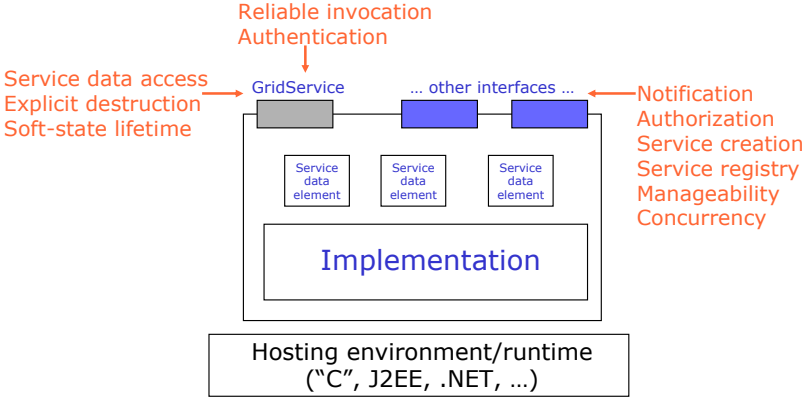
GT3
Architecture and Functionality

- Core
  - OGSI Implementation
  - Security Services
  - System-Level Services
  - Container
  - Hosting Environment
- Base Services
  - Resource Management
  - Information Services
  - Data Management
- User-Defined Services
  - Grid Service Development Framework
- Future Directions

GT-OGSA Grid Service Infrastructure



GT3 Core: The Grid Service = Interfaces + Service Data



GT3 Core: Notification and Subscription

- Our NotificationSourceProvider implementation allows any Grid Service to become a sender of notification messages
- A subscribe request on a NotificationSource triggers the creation of a NotificationSubscription service
- A NotificationSink can receive notification msgs from NotificationSources. Sinks are not required to implement the GridService portType
- Notifications can be set on SDEs

GT3 Core: OGSI Specification (cont.)

Factory portType

- Factories create services
- Factories are typically persistent services
- Factory is an optional OGSI interface

(Grid Services can also be instantiated by other mechanisms)

GT3 Core: OGSI Specification (cont.)

Service group portTypes

- A ServiceGroup is a grid service that maintains information about a group of other grid services
- The classic registry model can be implemented with the ServiceGroup portTypes
- A grid service can belong to more than one ServiceGroup
- Members of a ServiceGroup can be heterogenous or homogenous
- Service group portTypes are optional OGSI interfaces

GT3 Core: OGSI Specification (cont.)

Grid Service Handles (GSHs)

Globally unique

HandleResolver portType

- Defines a means for resolving a GSH (Grid Service Handle) to a GSR (Grid Service Reference)
  - A GSH points to a Grid Service
    (GT3 uses a hostname-based GSH scheme)
  - A GSR specifies how to communicate with the Grid Service
    (GT3 currently supports SOAP over HTTP, so GSRs are in WSDL format)

GT3 Core: Security Infrastructure

- Transport Layer Security/Secure Socket Layer (TLS/SSL)
  - To be deprecated
- SOAP Layer Security
  - Based on WS-Security, XML Encryption, XML Signature
- GT3 uses X.509 identity certificates for authentication
- It also uses X.509 Proxy certificates to support delegation and single sign-on, updated to conform to latest IETF/GGF draft

GT3 Core: Grid Service Container

Includes the OGSI Implementation, security infrastructure and system-level services, plus:

- Service activation, deactivation, construction, destruction, etc.
- Service data element placeholders that allow you to dynamically fetch service data values at query time
- Evaluator framework (supporting ByXPath and ByName notifications and queries)
- Interceptor/callback framework (allows one to intercept certain service lifecycle events)

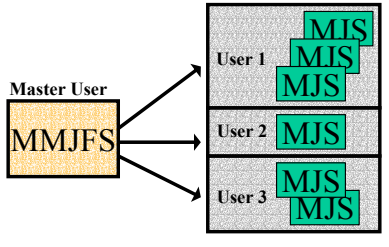
GT3 Core: Hosting Environment

GT3 currently offers support for four Java Hosting Environments:

- Embedded
- Standalone
- Servlet
- EJB

GT3 Base: Resource Management

- GRAM Architecture rendered in OGSA
- The MMJFS runs as an unprivileged user, with a small highly-constrained setuid executable behind it
- Individual user environments are created using Virtual Hosting

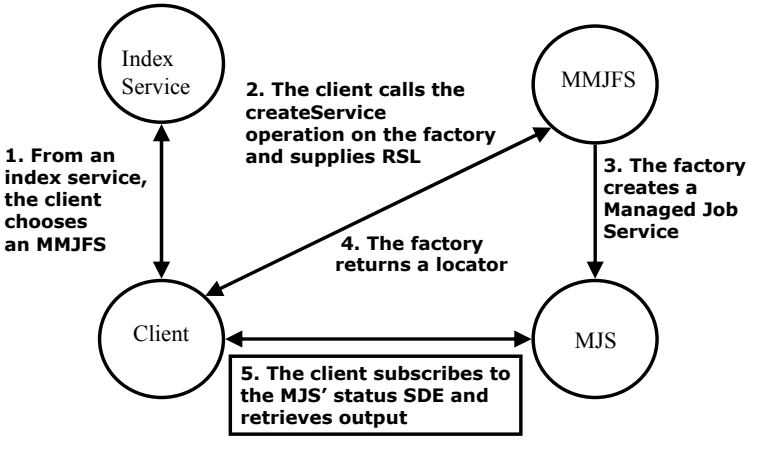


MMJFS: Master Managed Job Factory Service

MJS: Managed Job Service

User Hosting Env

GRAM Job Submission Scenario

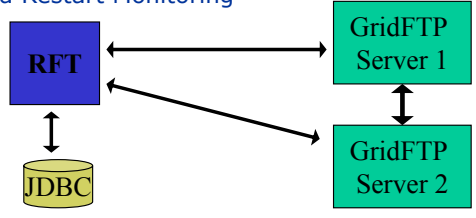


GT3 Base: Information Services

- Index Service as Caching Aggregator
  - Caches service data from other grid services
- Index Service as Provider Framework
  - Serves as a host for service data providers that live outside of a grid service to publish data

GT3 Base: Reliable File Transfer

- Reliably performs a third party transfer between two GridFTP servers
- OGSI-compliant service exposing GridFTP control channel functionality
- Recoverable Grid Service
  - Automatically restarts interrupted transfers from the last checkpoint
- Progress and Restart Monitoring



Summary

- The Grid problem: Resource sharing & coordinated problem solving in dynamic, multi-institutional virtual organizations
- Grid architecture: Emphasize protocol and service definition to enable interoperability and resource sharing
- Globus Toolkit™ Version 2: a source of protocol and API definitions, reference implementations
- GT3: Open Grid Services Architecture