Introduction au métacomputing et au global computing

Yves DENNEULIN
Projet APACHE - Laboratoire ID-IMAG

Motivation

- Développement d 'Internet
  - outils de communication : mail, news
  - serveurs de données : web
  - Interconnexion de dispositifs avancés (saisie, visualisation,…)
  - connexion d 'un grand nombre de machines
    - ⇨ large potentiel connecté de calcul inexploité
- Développement de serveurs de calcul
  - utilisation de composants standards
  - banalisation des techniques
    - processeurs puissants peu chers (Pentium, Alpha, PowerPC)
    - réseau rapide accessible
  - factorisation de moyens épars ?

⇨ Construction de supers calculateurs virtuels
  - métacomputing

Plan

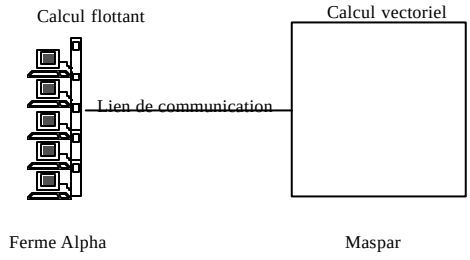
- Aspects matériels
  - infrastructure type pour le métacomputing
  - hétérogénéité
  - exemples américains et européens
- Les supports systèmes
  - support de l 'hétérogénéité
  - partage des ressources
  - accès distant
- Applications typiques de métacomputing
- Cas concret : le système Globus

Définition du métacomputing

- Applications parallèles traditionnelles
  - conçues pour une architecture
  - suivent
    - un modèle algorithmique
    - un modèle de programmation
    - un support d 'exécution
- Développement des réseaux
  - fédération de moyens distribués
  - hétérogénéité
    - modèles de calcul
    - Architecturale (calcul et visualisation)
    - réseaux et protocoles de communication
    - échelle de temps des communications
- Métacomputing = utiliser des ressources de calcul ou de stockage distribuées et hétérogènes

Exemple

- Petite échelle (un site)
  - code de Jacobi/Arnoldes



Exemple (2)

- Problèmes
  - vitesse du lien de communication
    - I/O
    - synchronisation
    - *idleness*
  - format des données
  - Algorithmique
  - Transport des données de/vers le calculateur
  - support
    - support (protocole) de communication commun : performances ? (TCP)
    - primitives de synchronisation

Global Computing

- Problématique semblable au métacomputing
  - Exploitation de ressources disséminées
- Mais…
  - Dispositifs de **nature** différente
    - Téléphones portables
    - PDA
    - Four à micro-ondes
    - …
  - Tout ce qui contient une puce capable de calculer!
- Exemple
  - Seti@home
  - Projet entropia : recherche médicale, physique nucléaire, etc.
  - XtremWeb en France

Global Computing (2)

- Problèmes
  - Dispositifs reliés de façon épisodique
    - Gérer la mobilité et la déconnexion
    - Granularité importante des calculs
  - Sécurité!!!
    - Accès au réseau
    - Accès limité à la machine hôte
      - Paramétré par le propriétaire de la machine
      - Principe de *sandboxing*
    - Possibilité de révocation immédiate de traitements en cours
      - Transfert (migration)
      - Destruction et perte de données => redondance des calculs
    - Prévention des fraudes
    - Protection du code
  - Rémunération

Peer to peer computing

- Sous-ensemble du global computing
  - Partage de puissance
  - Partage de fichiers
- Exemple
  - Napster
  - Gnutella
  - …
- Décentralisation
  - Du stockage
  - De la gestion des données et du calcul

Contents

- Framework
  - Hardware, system and administration issues
  - Application fields
- Platforms and programming environments
  - Heterogeneous communications (Nexus, Albatross)
  - Resource management (Globus, Legion)

Bibliography

- Books
  - *The Grid: Blueprint for a New Computing Infrastructure*
    I.Foster and C. Kesselman eds. Morgan-Kaufmann 1999
  - *High Performance Cluster Computing*
    Vol **1**: Architecture and Systems
    Vol **2**: Programming and Applications
    R. Buyya ed. Prentice Hall 1999.
- Journals
  - "*Blueprint for the future of high-performance computing*"
    CACM November 1997
  - *"The High-Performance computing continuum"*
    CACM November 1998

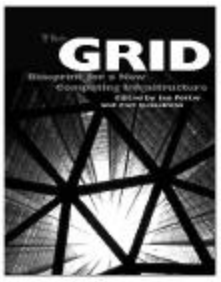
Bibliography (cont'd)

- Web
  - NPCACI *(National Partnership for Advanced Computational Infrastructure)*
    `http://www.npaci.edu`
  - "*An Overview of Computational Grids and Survey of a Few Research Projects*", Jack Dongarra
    `http://www.netlib.org/utk/people/JackDongarra/talks.html`
- LIP Report 99-36
  - "*Algorithms and Tools for (Distributed) Heterogeneous Computing: A Prospective Report*"
    `http://www.ens-lyon.fr/~yrobert`

The Grid: Blueprint for a New Computing Infrastructure

I. Foster, C. Kesselman (Eds), Morgan Kaufmann, 1999

- ISBN 1-55860-475-8
- 22 chapters by expert authors including Andrew Chien, Jack Dongarra, Tom DeFanti, Andrew Grimshaw, Roch Guerin, Ken Kennedy, Paul Messina, Cliff Neuman, Jon Postel, Larry Smarr, Rick Stevens, and many others



"A source book for the history of the future" -- Vint Cerf

Framework

Metacomputing

- Future of parallel computing
  distributed and *heterogeneous*
- Metacomputing = *Making use of distributed collections of heterogeneous platforms*
- Target = *Tightly-coupled high-performance distributed applications*
  (rather than loosely-coupled cooperative applications)

Metacomputing Platforms (1)

- Low end of the field
  Cluster computing with *heterogeneous networks* of workstations or PCs
  - Ubiquitous in university departments and companies
  - Typical poor man's parallel computer
  - Running a large PVM or MPI experiment (possibly all night long)
  - Make use of *all* available resources: slower machines **in addition** to more recent ones

Metacomputing Platforms (2)

- High end of the field
  Computational grid linking the most powerful supercomputers of the largest supercomputing centers through dedicated high-speed networks.
- Middle of the field
  Connecting medium size parallel servers (equipped with application-specific databases and application-oriented software) through fast but non-dedicated, thus creating a "*meta-system*"

Algorithmic and Software Issues (1)

Whereas the architectural vision is clear, the software developments are not so well understood

- Low end of the field: cope with heterogeneity
  Major algorithmic effort to be undertaken
- High end of the field
  - Logically assemble the distributed computers: extensions to PVM and MPI to handle distributed collection of clusters
  - Configuration and performance optimization
    - Inherent complexity of networked and heterogeneous systems
    - Resources often identified at runtime
    - Dynamic nature of resource characteristics

Algorithmic and Software Issues (2)

- High-performance computing applications must:
  - Configure themselves to fit the execution environment
  - Adapt their behavior to subsequent changes in resource characteristics
- Parallel run-time environments focused on strongly homogeneous architectures
  - Processor, memory, networks
- Homogeneity motivated the splendid research
  - Array and loop distribution, parallelizing compilers, HPF constructs, gang scheduling, MPI

However… Metacomputing platforms are strongly heterogeneous!

Tomorrow's Virtual Super-Computer (1)

- The web (and the associate data-base) is built using
  - A set of disks to store the data
  - A network infrastructure enabling a large number of users to access this data
- Metacomputing
  - using the computing power of the computers linked by Internet to execute various applications (numerically-intensive applications first, but many others to follow)
- Internet will slowly evolve into a virtual super-computer

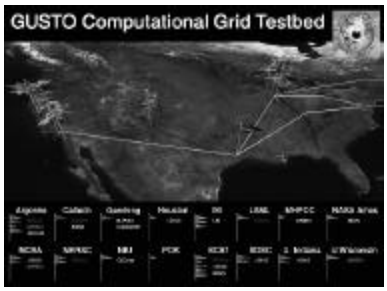
Tomorrow's Virtual Super-Computer (2)

- Metacomputing applications will execute on a hierarchical grid
  - Interconnection of clusters scattered all around the world
- A fundamental characteristic of the virtual super-computer
  - A set of strongly heterogeneous and geographically scattered resources

Hardware Platforms (1)

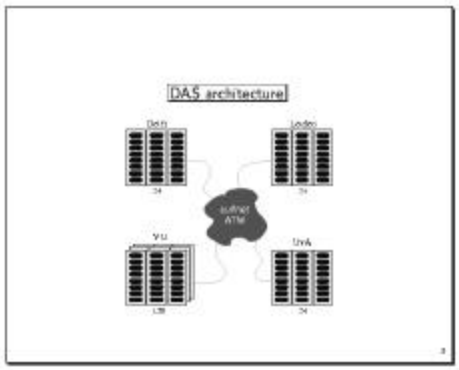
- Globus Ubiquitous Supercomputing Testbed Organization (GUSTO)
  - November 1998, **70** institutions, **3** continents
  - **17** sites, **330** supercomputers (over **3600** processors)
  - Aggregate global power in excess of **2 TeraFlops per second**!

Gusto



Hardware Platforms (2)

- Distributed ASCI Supercomputer (DAS)
  - Common platform for research
  - (Wide-area) parallel computing and distributed applications
  - November 1998, 4 universities, 200 nodes
  - Node
    - 200 MHz Pentium Pro
    - 128 MB memory, 2.5 GB disk
    - Myrinet 1.28 Gbit/s (full duplex)
    - Operating System: BSD/OS
  - ATM Network



Administrative Issues (1)

- Intensive computations on a set of processors across several countries and institutions
  - Strict rules to define the (good) usage of shared resources
- Respect of these rules must be guaranteed by the runtime, together with methods to migrate computations to other sites whenever some local request is raised

Administratives Issues (2)

- In a metacomputing context, a major difficulty is to avoid a large increase in the administrative overhead
  - Each user cannot have an account on each machine on the network
  - A single user (meta-user) cannot be the one and only one authorized user on the whole set of machines
  - Challenge = find a tradeoff that does not increase the administrative load while preserving the security of the users

Applications (1)

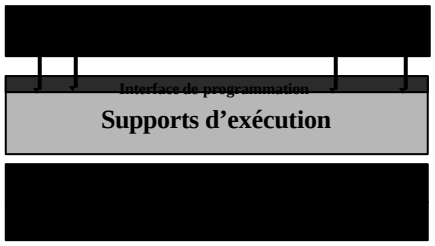
- All applications involving parallel computing Performance problems due to:
  - Using a network of heterogeneous machines
  - Relying on current (limited) programming environments
- "Classical applications" such as the grand challenges can be ported on metacomputing platforms
  - Forget fine-grain parallelism → deep hierarchy between all memory and communication layers
- Code coupling: nicest application for metacomputing
  - Large granularity
  - Loose exchanges between different component applications

Applications (2)

- Other applications out of the world of numerical (or scientific) computing
  - data-bases, decision-support systems
  - all kind of multimedia servers
- Best candidates for metacomputing are loosely-coupled applications:
  - All kinds of decompositions (functional, pipeline, data-parallel)
  - Applications needing various heterogeneous devices
  - Actual challenge = implementation of tightly-coupled applications

Platforms and Programming Environments

Supports d'exécution et environnements parallèles



Nouveaux défis pour les supports exécutifs

- Communications hétérogènes
  - Réseaux: étendus, locaux, grappes, …
  - Applications à hautes-performances
- Gestion de ressources
  - Spécification, localisation, allocation des ressources
  - Gestion dynamique
    - Choix au moment du lancement de l'application
    - Peuvent être remis en cause à chaque instant
- Administration système
  - Sécurité, utilisateurs
  - Politiques d'accès

Communications et métacomputing

- Structure complexe des réseaux, hétérogénéité des architectures et des systèmes
  - Des protocoles de communication multiples
- Haute-performance
  - Contrôle explicite des mécanismes de bas-niveau
- Recouvrement, comportement dynamique
  - Multithreading
- Facilité de programmation, portabilité
  - Paradigme et abstraction

Schémas et protocoles de communication

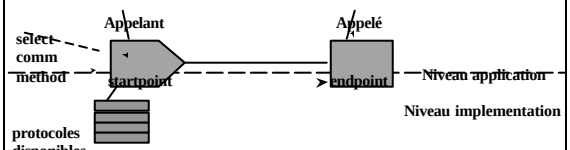
- Les applications sont basées sur des schémas de communication
  - Algorithmique
- Le protocole de communication le plus approprié peut dépendre de
  - la destination (Où)
  - du type des données (Quoi)
  - du moment (Quand)

Protocoles de communication

- Point à point, diffusion
- Fiable, non fiable
- Qualité de service
- TCP/IP, ATM AAL5, mémoire partagée, …
- Crypté, non crypté
- Compressé, non compressé

Etude de cas: Nexus/Globus

- **Un seul paradigme de programmation**
  - **RSR: Remote Service Request**



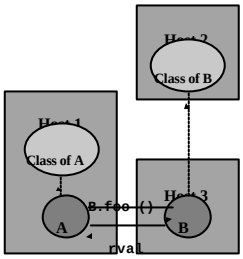
- Sélection du protocole de communication, soit automatique, soit manuelle

Etude de cas: Legion

- Legion is object-based
  - Everything is an object
- Legion's core "system" objects
  - Hosts
  - Classes LegionClass
- All system objects can be replaced

Legion: The object model

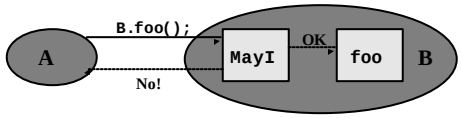
- Legion objects
  - belong to classes
  - are logically independent, address-space disjoint
  - communicate via non-blocking method calls
  - are *Active* or *Inert*
  - are named using LOIDs



Legion Object Identifiers (LOIDs)

- LOIDs name object are location independent string of bits
- LOIDs have many fields (arbitrary)
- Basic LOIDs have
  - type
  - "domain"
  - class identifier (integer)
  - instance identifier (integer)
  - public key (RSA)

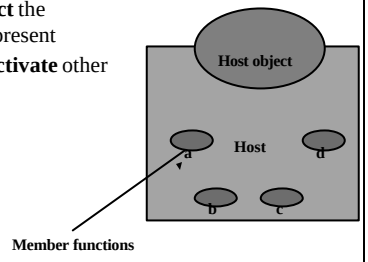
Access Control



- No single access control policy
  - Use a standard policy or define your own
- Security restricted to `MayI` function
  - Policies can be replaced and verified
  - use a simple access control language

Legion: Host objects

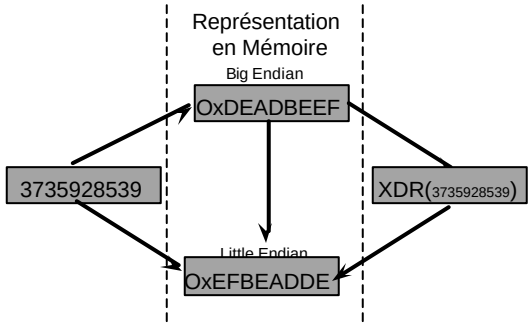
- Run on each "host" in the legion system
- **Guard** and **protect** the resources they represent
- **Activate** and **deactivate** other legion objects



Hétérogénéité des représentations de données

- Différents processeurs (Pentium, Sparc ,ALPHA,...)
  - Little Endian ↔ Big Endian
  - 32 bits ↔ 64 bits
- Solutions possibles :
  - Format universel de représentation de données
    - XDR, ASN.1...
    - Surcoût dû aux recopies
  - Conversion directe

Conversion directe de donnée



Scrutation des communications

- Contexte homogène
  - Comment et quand scruter le réseau ?
  - Compromis avec le calcul
- Contexte hétérogène
  - Plusieurs interfaces de communication
  - Différentes politiques de scrutation
  - Utilisation du multithreading

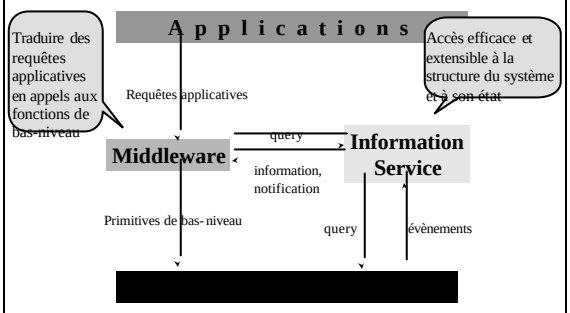
Communications Globales

- Exemple: les opérations MPI
  - Barrier, Broadcast, Alltoall
  - Scatter/Gather, Global reduction
- Réseaux hiérarchiques et hétérogènes
  - Efficacité: réduire les communications sur les réseaux lents
  - Algorithmes hiérarchiques
    - Arbres de diffusion
    - Aucune donnée n'est envoyée plusieurs fois vers le même cluster

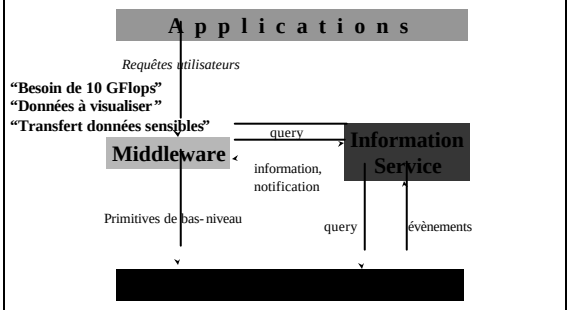
Problèmes de configuration

- Le choix d'une configuration appropriée est stratégique pour obtenir de bonnes perfs.
  - Comment une application détermine les ressources disponibles?
  - Quel est l'état des ressources ?
  - Comment peut-on optimiser une application à partir de cette configuration?
- Quelques éléments de réponse...

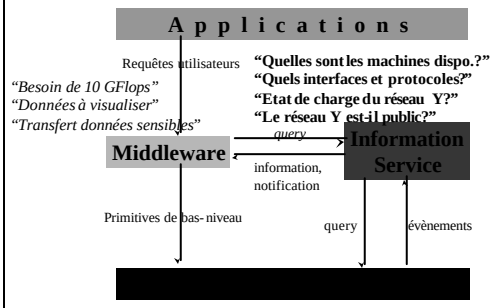
Principe de fonctionnement



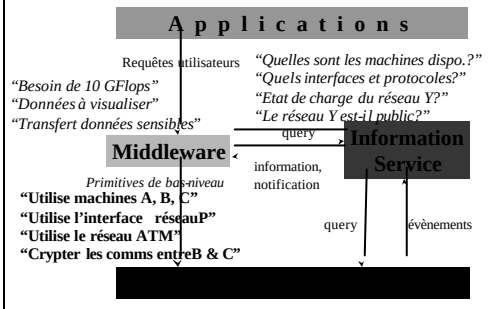
Exemple (1)



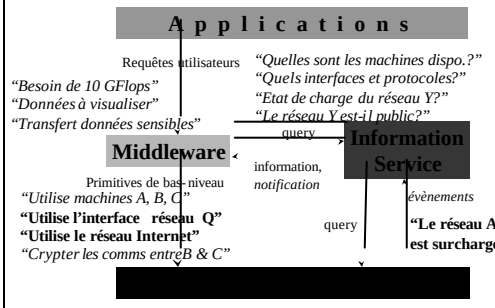
Exemple (2)



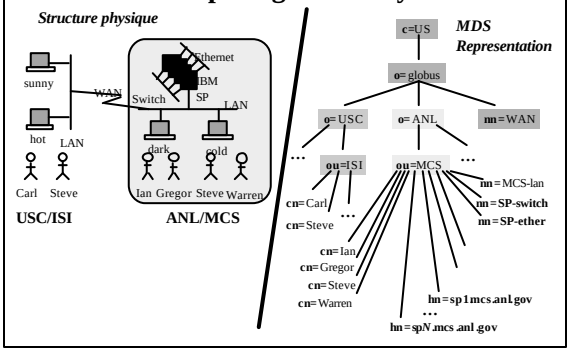
Exemple (3)

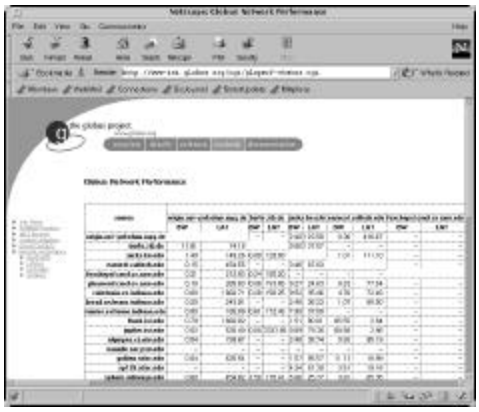


Exemple (4)



Metacomputing Directory Service

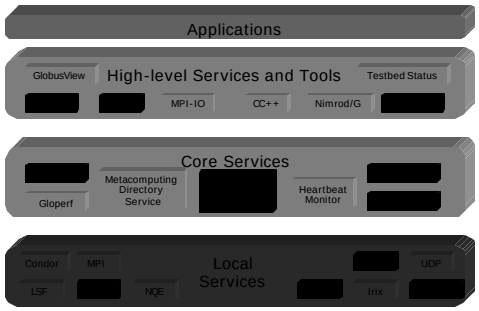




Services Globus

- Communication infrastructure (Nexus, IO)
- Information services (MDS)
- Network performance monitoring (Gloperf)
- Process monitoring (HBM)
- Remote file and executable management (GASS and GEM)
- Resource management (GRAM)
- Security (GSI)

Architecture Globus



Références

- Globus
  - http://www.globus.org
- Legion
  - http://www.cs.virginia.org/~legion
- Albatross
  - http://www.cs.vu.nl/~bal/albatross

Globus Toolkit

A User-level Tutorial To Grid Programming

Introduction

The Globus Project Team

http://www.globus.org

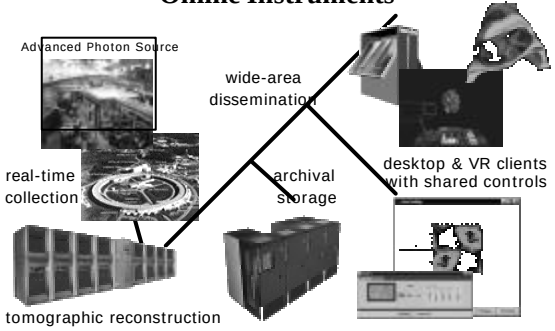
Overview

- Introduction to computational grids
- High-level overview of the Globus Toolkit
- Four components:
  - Security and remote process creation
  - Running across multiple resources with MPICH-G
  - Information services
  - Resource management and remote file access
- Case studies
- Other Globus services, and future directions

Why "The Grid"?

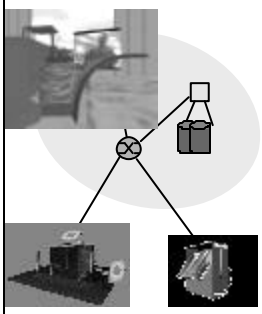
- New applications based on high-speed coupling of people, computers, databases, instruments, etc.
  - Online instruments
  - Collaborative engineering
  - Parameter studies
  - Browsing of remote datasets
  - Use of remote software
  - Data-intensive computing
  - Very large-scale simulation

Online Instruments



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

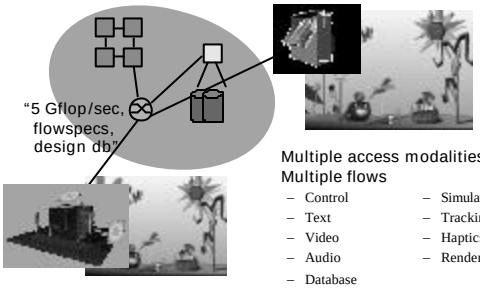
Collaborative Engineering



- Manipulate shared virtual space, with
  - Simulation components
  - Multiple flows: Control, Text, Video, Audio, Database, Simulation, Tracking, Haptics, Rendering
- Issues:
  - (un)reliable uni/multicast
  - Security
  - Reservation & QoS

CAVERNsoft: UIC, Electronic Visualization Laboratory

Tele-immersion

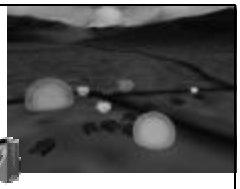


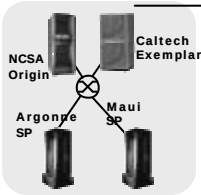
Multiple access modalities
Multiple flows

- Control
- Text
- Video
- Audio
- Database
- Simulation
- Tracking
- Haptics
- Rendering

Leigh et al: UIC, Electronic Visualization Laboratory

Distributed Supercomputing



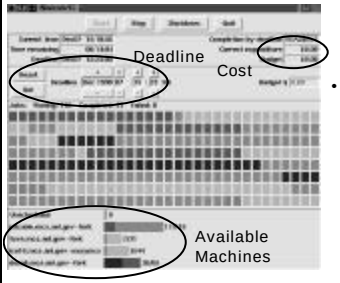


Issues:

- Resource discovery, scheduling
- Configuration
- Multiple comm methods
- Message passing (MPI)
- Scalability
- Fault tolerance

SF-Express Distributed Interactive Simulation: Caltech, USC/ISI

High-Throughput Computing



- Schedule many independent tasks
  - Parameter studies
  - Data analysis
- Issues:
  - Resource discovery
  - Data Access
  - Scheduling
  - Reservation
  - Security
  - Accounting
  - Code management

Nimrod-G: Monash University

Problem Solving Environments

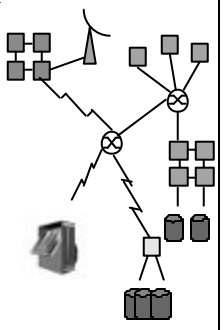
- Examples:
  - Problem solving env. for computational chemistry
  - Application web portals
- Issues:
  - Remote job submission, monitoring, and control
  - Resource discovery
  - Distributed data archive
  - Security
  - Accounting

ECCE': Pacific Northwest National Laboratory

The Grid

"Dependable, consistent, pervasive access to [high-end] resources"

- Dependable: Can provide performance and functionality guarantees
- Consistent: Uniform interfaces to a wide variety of resources
- Pervasive: Ability to "plug in" from anywhere



Evolution of a Concept

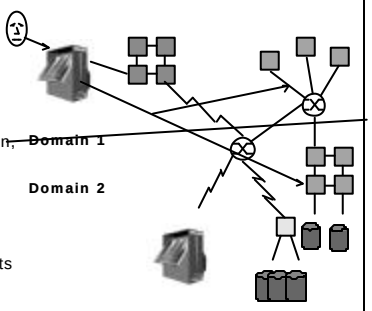
- Metacomputing: late 80s
  - Focus on distributed computation
- Gigabit testbeds: early 90s
  - Research, primarily on networking
- I-WAY: 1995
  - Demonstration of application feasibility
- NFS PACIs (National Technology Grid): 1998
- NASA Information Power Grid: 1999
- DOE ASCI DISCOM DRM: 1999
- European Grid: 2000

Technical Challenges

- Complex application structures, combining aspects of parallel, multimedia, distributed, collaborative computing
- Dynamic varying resource characteristics, in time and space
- Need for high & guaranteed "end-to-end" performance, despite heterogeneity and lack of global control
- Inter-domain issues of security, policy, payment

Issues

•Authenticate once
•Specify simulation (code, resources, etc.)
•Locate resources
•Negotiate authorization, acceptable use, etc.
•Acquire resources
•Initiate computation
•Steer computation
•Access remote datasets
•Collaborate on results
•Account for usage



Architectural Approaches

- Distributed systems: DCE, CORBA, Jini, etc.
  - Rich functionality eases app development
  - Complexity hinders deployment
    - especially in absence of global control
  - Performance difficulties
- Internet/Web Protocols and Tools
  - Simple protocols facilitate deployment
  - Missing functionality hinders app development
  - Performance difficulties

Standards & Commodity Tech

- Where appropriate, exploit standards and commodity technology in core infrastructure
  - LDAP, SSL/TLS, X.509, GSS-API, http, ftp, XML, SOAP, etc.
  - Provides leverage
- Interface with other common standards
  - CORBA, Java/Jini, DCOM, Web, etc
  - While our core infrastructure may not be built on one of these distributed architectures, we can and must cleanly interface with them

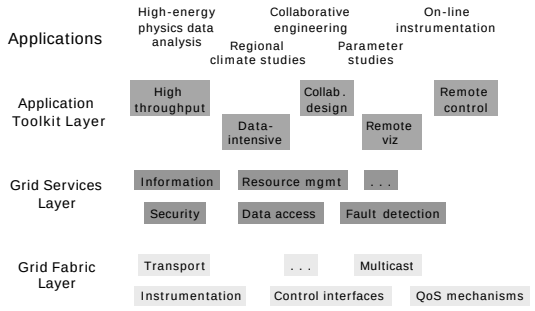
The Globus Project

- Basic research in grid-related technologies
  - Resource & data management, security, QoS, policy, communication, adaptation, etc.
- Development of Globus Toolkit
  - Core services for grid-enabled tools & apps
- Construction of production grids & testbeds
  - Multiple deployments to distributed organizations for production & prototyping
- Application experiments
  - Distributed applications, tele-immersion, etc.

Globus Project Participants

- Globus Project is a large community effort
  - Globus Toolkit core development
    - Argonne, USC/ISI, NCSA, SDSC
  - Globus Toolkit contributors
    - NASA, DOE ASCI DRM (SNL, LBNL, LLNL), Raytheon, and numerous others
  - Collaborators
    - University, lab, industrial, and international partners spanning many scientific and engineering disciplines
- Active in Grid Forum
  - http://www.gridforum.org

Grid Services Architecture

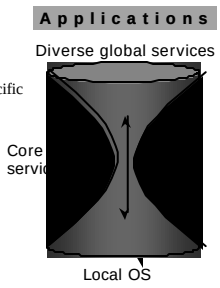


Globus Approach

- A toolkit and collection of services addressing key technical problems
  - Modular "bag of services" model
  - Not a vertically integrated solution
  - General infrastructure tools (aka middleware) that can be applied to many application domains
- Inter-domain issues, rather than clustering
  - Integration of intra-domain solutions
- Distinguish between local and global services

Globus Hourglass

- Focus on architecture issues
  - Propose set of core services as basic infrastructure
  - Use to construct high-level, domain-specific solutions
- Design principles
  - Keep participation cost low
  - Enable local control
  - Support for adaptation
  - "IP hourglass" model



Technical Focus & Approach

- Enable incremental development of grid-enabled tools and applications
  - *Model neutral*: Support many programming models, languages, tools, and applications
  - Evolve in response to user requirements
- Deploy toolkit on international-scale production grids and testbeds
  - Large-scale application development & testing
- Information-rich environment
  - Basis for configuration and adaptation

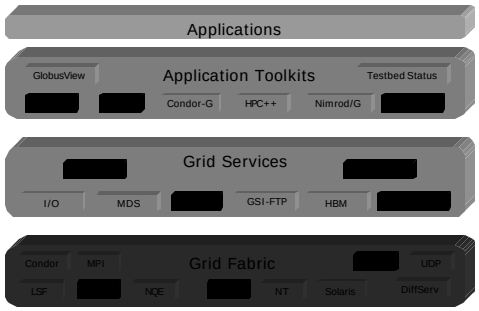
Globus Toolkit Grid Services

- Security (GSI)
- Resource management (GRAM)
- Information services (MDS)
- Remote file management (GASS)
- Communication (I/O, Nexus)
- Process monitoring (HBM)

Other Globus Project Grid Services

- Coming Soon
  - Data transfer (GSI-FTP)
  - Replica Management
    http://www.globus.org/datagrid
- Experimental Prototypes
  - Advanced Reservations & QoS (GARA)
  - Distributed Events & Logging

Layered Architecture



Sample of High-Level Services

- Resource brokers and co-allocators
  - DUROC, Nimrod/G, Condor-G, ASCI DRM
- Communication & I/O libraries
  - MPICH-G, PAWS, RIO (MPI-IO), PPFS, MOL
- Parallel languages
  - HPC++, CC++
- Collaborative environments
  - CAVERNsoft, ManyWorlds
- Others
  - MetaNEOS, NetSolve, LSA, AutoPilot, WebFlow

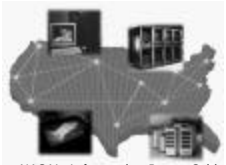
Condor-G: Condor for the Grid

- Condor is a high-throughput scheduler
- Condor-G uses Globus Toolkit libraries for:
  - Security (GSI)
  - Managing remote jobs on Grid (GRAM)
  - File staging & remote I/O (GSI-FTP)
- Grid job management interface & scheduling
  - Robust replacement for Globus Toolkit programs
    - Globus Toolkit focus is on libraries and services, not end user vertical solutions
  - Supports single or high-throughput apps on Grid
    - Personal job manager which can exploit Grid resources

Production Grids & Testbeds

- Production deployments underway at:
  - NSF PACIs National Technology Grid
  - NASA Information Power Grid
  - DOE ASCI
  - European Grid
- Research testbeds
  - EMERGE: Advance reservation & QoS
  - GUSTO: Globus Ubiquitous Supercomputing Testbed Organization
  - Particle Physics Data Grid

Production Grids & Testbeds



NASA's Information Power Grid

The Alliance National Technology Grid

GUSTO Testbed

Example Application Projects

- Computed microtomography (ANL, ISI)
  - Real-time, collaborative analysis of data from X-Ray source (and electron microscope)
- Hydrology (ISI, UMD, UT; also NCSA, Wisc.)
  - Interactive modeling and data analysis
- Collaborative engineering ("tele-immersion")
  - CAVERNsoft @ EVL
- OVERFLOW (NASA)
  - Large CFD simulations for aerospace vehicles

Example Application Experiments

- Distributed interactive simulation (CIT, ISI)
  - Record-setting SF-Express simulation
- Cactus
  - Astrophysics simulation, viz, and steering
  - Including trans-Atlantic experiments
- Particle Physics Data Grid
  - High Energy Physics distributed data analysis
- Earth Systems Grid
  - Climate modeling data management

Where We Are (June 2000)

- New results in QoS, data management, security, portals, tools, scheduling, etc.
- Globus Toolkit v1.1.3 released
  - Available on most Unix-es, partially on Win32
- Production deployment underway
  - NSF PACIs, NASA IPG, DOE ASCI DRM, …
- Many production and research applications and tools are leveraging this considerable investment in infrastructure
- Always looking for interesting applications