



CMSC 818S Grid Computing

Globus: A Metacomputing Infrastructure Toolkit *Ian Foster and Carl Kesselman*

Binh Viet Nguyen

Contents

- Metacomputing
- Globus metacomputing infrastructure Toolkit
- Globus Toolkit Components
- Higher-Level Services
- Globus Testbeds and Experiences
- Conclusion

Metacomputing

- Metacomputing: a networked virtual supercomputer, constructed dynamically from geographically distributed resources linked by high-speed networks.

Introduction

- Globus: Metacomputing infrastructure toolkit providing basic capabilities and interfaces
- Object in Globus: Address problems of configuration and performance optimization in metacomputing environments
- AWARE: Higher-level service layered on Globus toolkit

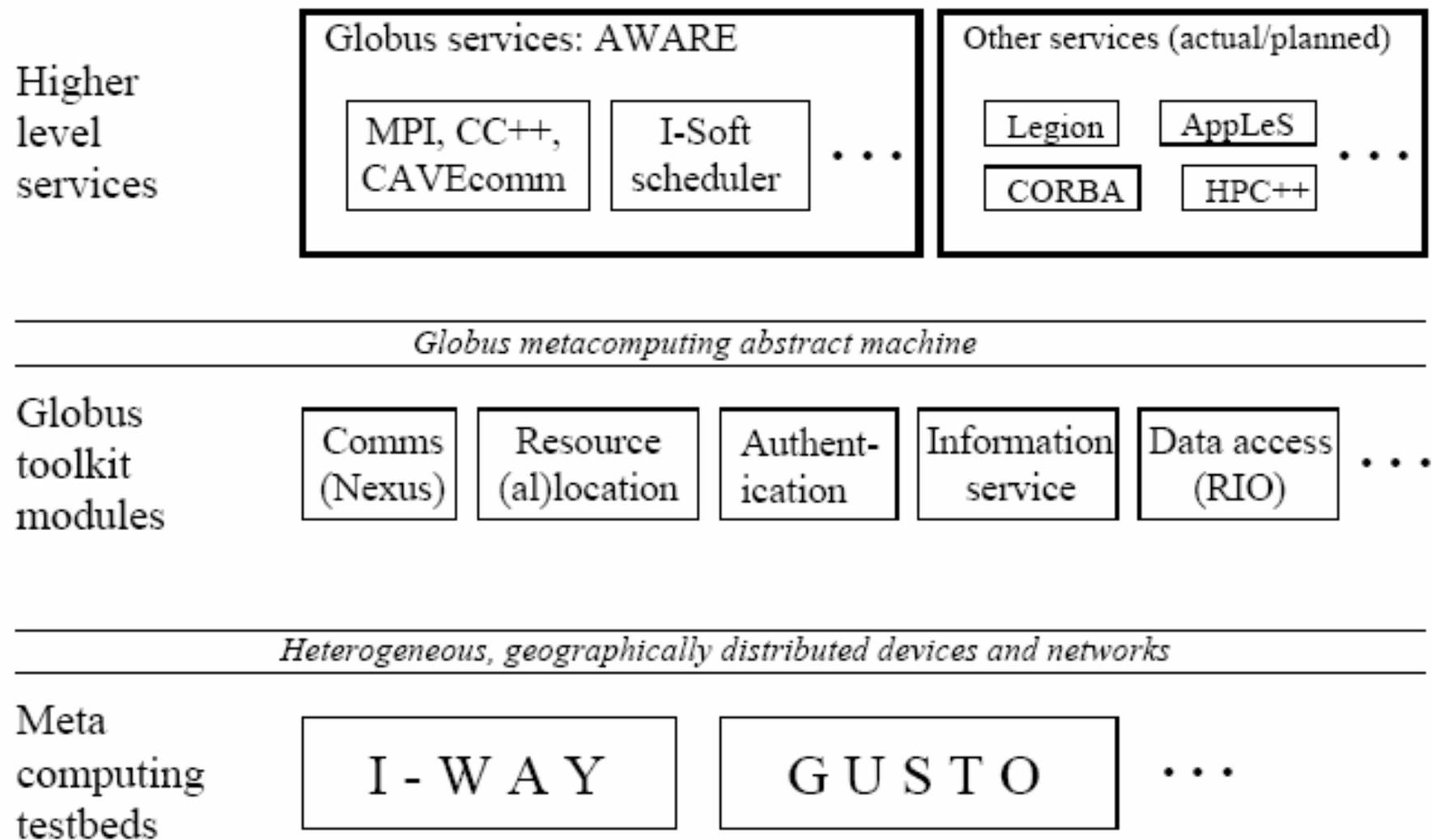


FIG. 1. *The Globus toolkit*

Motivation for metacomputing

- Need to access resources not located in a single computer
- Economic consideration: the resources in question – for example, supercomputers – are too expensive to be replicated
- Unique resources – such as specialized databases and people – cannot be replicated

I-WAY

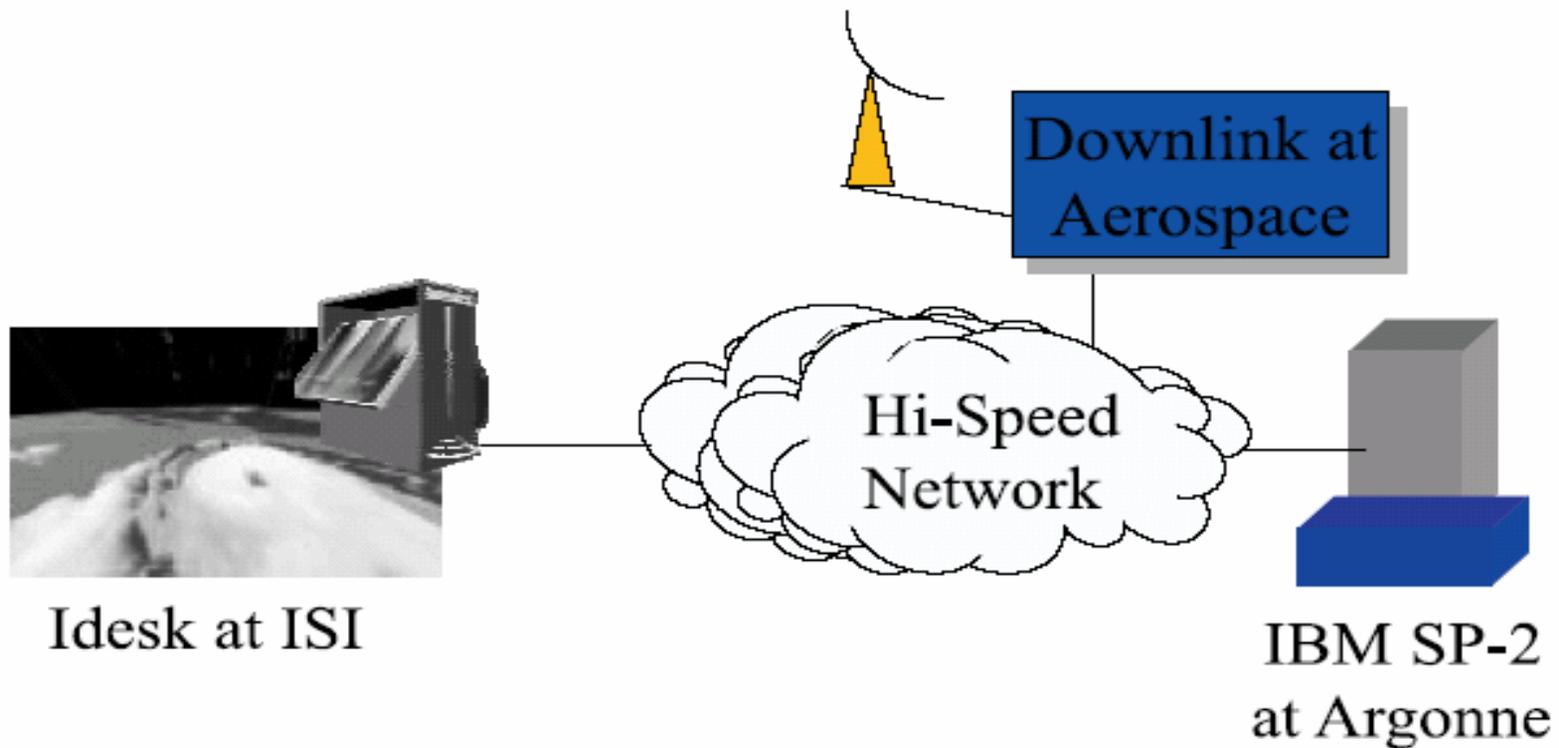


FIG. 2. This figure shows a networked supercomputing system used during the I-WAY experiment for the real-time analysis of data from a meteorological satellite. The satellite downlink is located in El Segundo, California; the supercomputer at Argonne, Illinois; and the visualization engine and graphics device in Los Angeles.

Application classes of metacomputing

- Desktop supercomputing
 - Application of high-end graphics capabilities with remote supercomputer and data set
- Smart instruments
 - Connect users to instruments such as telescope, microscope, satellite downlinks.
- Collaborative environments
 - Multiple virtual environments for interaction of users in different places
- Distributed supercomputing
 - Multiple computers to tackle hard problems

Distributed Supercomputing

- Scheduled mode
 - resources, once acquired, are dedicated to an application
 - is required for tightly coupled simulations, particularly those with time constraints
- Unscheduled mode
 - applications use otherwise idle resources that may be reclaimed if needed
 - Appropriate for loosely coupled applications that can adapt to time-varying resources

Metasystem Characteristics (1)

- Scale and the need for selection:
 - resources will be selected for particular applications according to criteria such as connectivity, cost, security, and reliability.
- Heterogeneity at multiple levels:
 - Heterogeneity can arise at multiple levels, ranging from physical devices, through system software, to scheduling and usage policies.

Metasystem Characteristics (2)

- Unpredictable structure
 - Metacomputing applications can be required to execute in a wide range of environments, constructed dynamically from available resources.
 - Geographical distribution and complexity are other factors that make it difficult to determine system characteristics such as network bandwidth and latency *a priori*

Metasystem Characteristics (3)

- Dynamic and unpredictable behavior
 - In metacomputing environments, resources – especially networks- are likely to be shared. The performance behavior can vary over time.
 - For example: network latency, bandwidth, and jitter may vary as traffic is rerouted. Suffered from network and resources failures
 - Not possible to guarantee even minimum quality of service requirements.

Metasystem Characteristics (4)

- Multiple administrative domains
 - Resources used by metacomputing applications often are not owned or administered by a single entity.
 - Network security problem as different entities may use different authentication mechanisms, authorization schemes, and access policies.
 - The need to execute user-specified code at different sites introduces additional concerns.

Globus Metacomputing infrastructure toolkit (1)

- The development of low-level mechanisms that can be used to implement higher-level services
- Techniques that allow higher-level services to observe and guide the operation of these mechanisms.
- This approach can reduce the complexity and improve the quality of metacomputing software by allowing single low-level infrastructure to be used for many purposes.
- Long-term goal of Globus is to address the problems of configuration and performance optimization in metacomputing environments

Globus Metacomputing infrastructure toolkit (2)

- The currently identified modules are:
 - Resource Location and Allocation
 - Communications
 - Unified resource information service
 - Authentication interface
 - Process creation
 - Data access

Globus Metacomputing infrastructure toolkit (3)

- Resource location and allocation: Provides mechanisms for
 - Expressing application resource requirements
 - Identifying resources that meet these requirements
 - Scheduling resources once have been located.
- In some situation – for example, on some supercomputers – location and allocation must be done in a single step

Globus Metacomputing infrastructure toolkit (5)

- Communications:
 - Provides basic communication mechanisms.
 - Must permit the efficient implementation of a wide range of communication methods, including message passing, RPC, distributed shared memory, stream-based, and multicast.
 - Must be cognizant of network quality of service parameters such as jitter, reliability, latency, and bandwidth.

Globus Metacomputing infrastructure toolkit (6)

- Unified resource information service
 - Provides a uniform mechanism for obtaining real-time information about metasytem structure and status
 - Must allow components to post as well as receive information
 - Support for scoping and access control is required

Globus Metacomputing infrastructure toolkit (7)

- *Authentication interface:*
 - Provides basic authentication mechanism that can be used to validate the identity of both users and resources
- *Process creation*
 - Initiating computation on a resource once it has been located and allocated.
 - Setting up executables, creating an execution environment, starting an executable, passing arguments, integrating new process into the rest of computation, and managing termination and process shutdown

Globus Metacomputing infrastructure toolkit (8)

- *Data access*
 - Provide high-speed remote access to persistent storage such as files.
 - Some data resources such as databases may be accessed via distributed database technology or CORBA (Common Object Request Broker Architecture)
 - Address the problem of achieving high performance when accessing parallel file systems and network-enabled I/O devices such as High Performance Storage System (HPSS)

Resource-Aware Services and Applications

- Globus Toolkit modules provided interfaces to do:
- *Rule-based selection*
 - Identify selection points at which choices from among alternatives (resources, parameter values, etc.) are made. E.g. “use TCP packet size X”, “use TCP over ATM”
- *Resource property inquiry*
 - Information provided by the uniformed information service can be used to guide selection processes within both Globus modules and applications that use these modules.

Resource-Aware Services and Applications (2)

- *Notification*

- A notification mechanism allows a higher-level service or application to specify constraints on the quality of service delivered by a Globus service and to name a call-back function that should be invoked if these constraints are violated.
- This can be used, for example, to switch between networks when one becomes loaded

Resource-Aware Services and App

- Toolkit modules provided interfaces to do:
 - Rule-based selection
ex: *"use TCP packet size X over ATM"*
 - Resource properly inquiry
ex: *"use ATM if loverload is low,
otherwise; use internet"*
 - Notification
specify constrains via call-back function

Communications (1)

- Based on the Nexus communication library.
- Nexus define 5 basic abstractions: nodes, threads, communication links, and remote service requests
- If multiple startpoints are bound to an endpoint, incoming communications are interleaved.
- If a startpoint is bound to multiple endpoints, communications result in multicast.

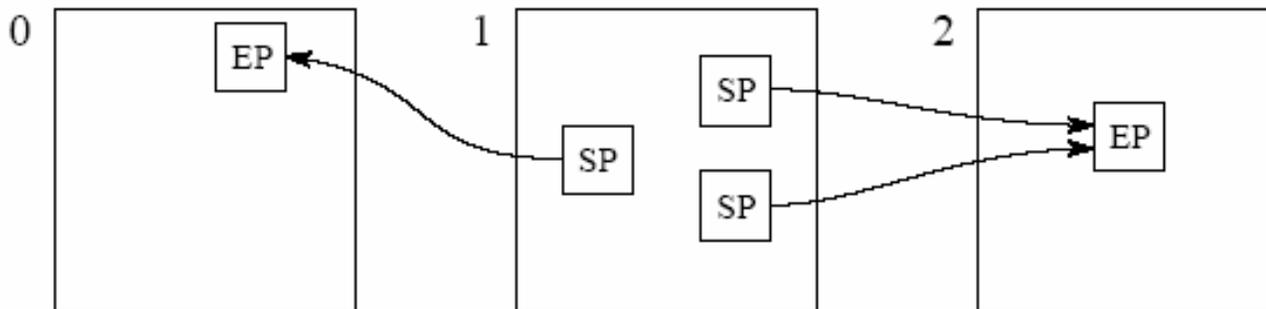


FIG. 3. *Nexus communication mechanisms. The figure shows three contexts (address spaces) and three communication links. Three startpoints in context 1 reference endpoints in contexts 0 and 2.*

Communications (2)

- A startpoint can be copied between processors, causing new communication links to be created that mirror the links associated with the original startpoint. Startpoints can be used as global names for objects and used anywhere in a distributed system.
- A communication link support a single communication operation: an asynchronous *remote service request (RSR)*

Communications (3)

- The Nexus interface and implementation support rule-based selection of the methods – such as protocol, compression method, and quality of service - used to perform communication.
- Different communication methods can be associated with different communication links, with selection rules determining which method should be used when a new link is established

Metacomputing Directory Service (1)

- Provide information:
 - Configuration details about resources such as the amount of memory, CPU speed.
 - Instantaneous performance information, such as point-to-point network latency, CPU load
 - Application-specific information, such as memory requirements
- Information can be obtained from multiple sources, such as NIS, SNMP.

Metacomputing Directory Service (2)

- Build on the data representation and application programming interface defined by Lightweight Directory Access Protocol (LDAP).
- Information is structured as a set of entries which comprises zero or more attribute-value pairs. The type of an entry, called its object class, specifies mandatory and optional attributes.

Metacomputing Directory Service (3)

GlobusHost OBJECT CLASS

SUBCLASS OF GlobusResource

MUST CONTAIN {

 hostname :: cis,

 type :: cis,

 vendor :: cis,

 model :: cis,

 OSType :: cis,

 OSversion :: cis

}

MAY CONTAIN {

 networkNode :: dn,

 totalMemory :: cis,

 totalSwap :: cis,

 dataCache :: cis,

 instructionCache :: cis

}

GlobusResource OBJECT CLASS

SUBCLASS OF top

MUST CONTAIN {

 administrator :: dn

}

MAY CONTAIN {

 manager :: dn,

 provider :: dn,

 technician :: dn,

 description :: cis,

 documentation :: cis

}

FIG. 4. *Simplified versions of the MDS object classes GlobusHost and GlobusResource*

Authentication Methods

- Use the Generic Security System (GSS) which defines a standard procedure and API for obtaining credentials (password or certificates), for mutual authentication (client and server), and for message-oriented encryption and decryption.
- GSS is independent of any particular security mechanism and can be layered on top of different security methods, such as Kerberos and SSL

Data Access Services (1)

- RIO (remote I/O) interface is based on the abstract I/O device (ADIO) interface. ADIO defines an interface for opening, closing, reading and writing parallel files.
- RIO extends ADIO by adding transparent remote access and global naming using a URL-based naming scheme. RIO remote access features use Nexus mechanism.

Data Access Services (2)

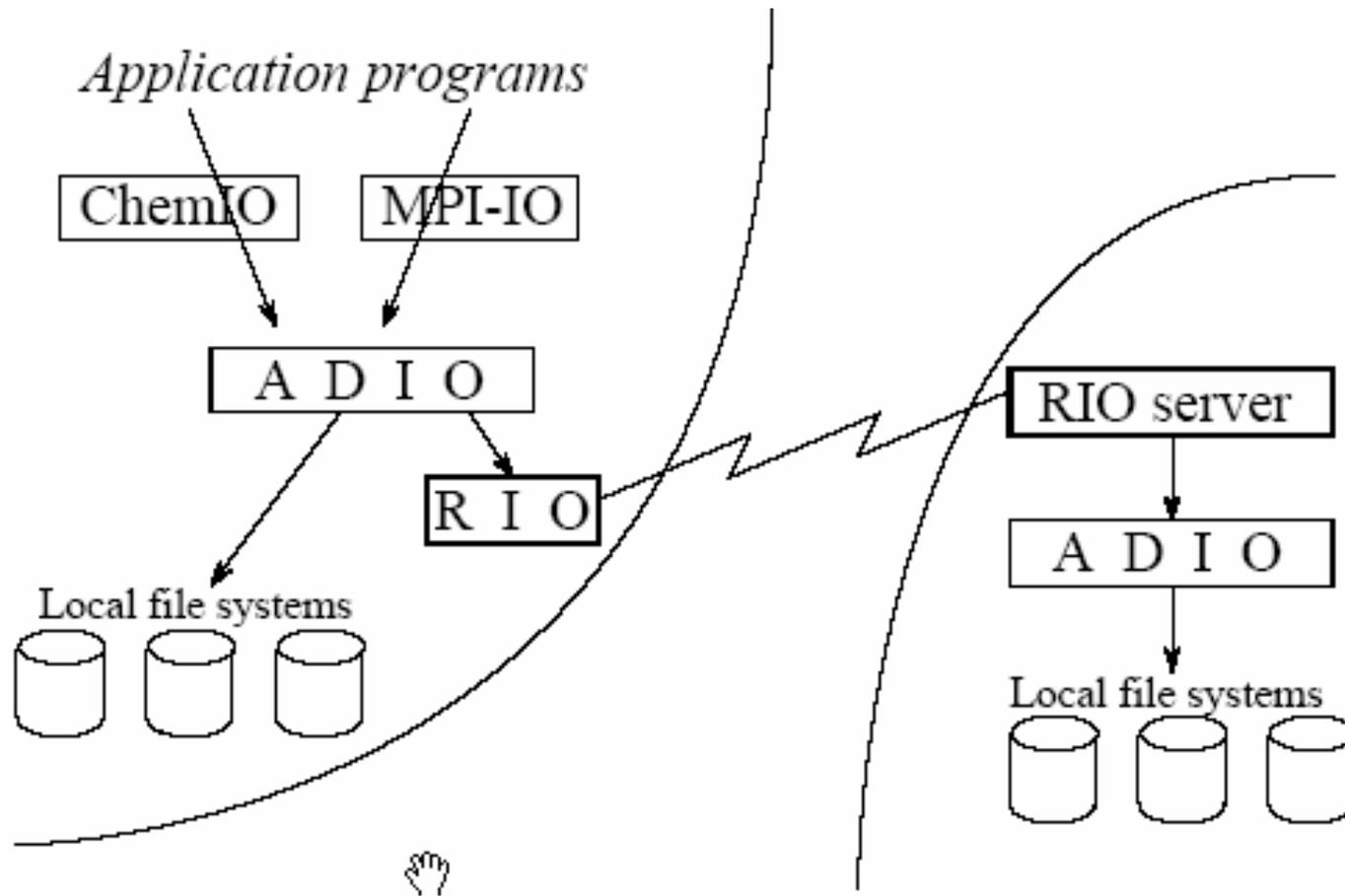


FIG. 5. *RIO mechanisms for high-performance access to remote file systems*

Higher-Level Service (1)

- *Parallel Programming Interfaces*
 - Numerous parallel programming interfaces have been adapted to use Globus authentication, process creation, and communication services
 - This interfaces include a complete implementation of MPI; Compositional C++, Fortran M, nPerl, and NexusJava

Higher-Level Service (2)

- Unified Certificate-based Authentication
 - Define a global, public key-based authentication space for all users and resources.
 - Provide a centralized authority that defines system-wide names (“accounts”) for users and resources.
 - Doesn’t address authorization: resource can use their usual mechanism to determine the users to which they will grant access

Globus Testbeds and Experience

- I-WAY: connect supercomputers and resources at 17 sites across North America
- GUSTO (the Globus Ubiquitus Supercomputing Testbed)

Conclusion

- Globus is attacking the metacomputing software problems
- Contribution of Globus project
 - Definition of core metacomputing system architecture
 - Development of a framework for metacomputing environment
 - Demonstration in testbed such as I-WAY