

Globus

- From version 1.0 in 1998 to the 2.0 release in 2002 and the latest 3.0, the emphasis is to provide a set of components that can be used either independently or together to develop applications
- The Globus Toolkit version 2 (GT2) design is highly related to the architecture proposed by Foster *et al.*
- The Globus Toolkit version 3 (GT3) design is based on grid services, which are quite similar to web services. GT3 implements the *Open Grid Service Infrastructure* (OGSI).
- GT4 is also based on grid services, but with some changes in the standard
- GT5 provides an API multithreaded implementation based on an asynchronous event model

Globus

- Toolkit with a set of components that implement basic services:
 - Security
 - resource location
 - resource management
 - data management
 - resource reservation
 - Communication

Core Globus Services

- Communication Infrastructure (Nexus)
- Information Services (MDS)
- Remote File and Executable Management (GASS, RIO, and GEM)
- Resource Management (GRAM)
- Security (GSS)

Communications (Nexus)

- Communication library (ANL & Caltech)
 - Asynchronous communications
 - Multithreading
 - Dynamic resource management

Communications (Nexus)

- 5 basic abstractions
 - Nodes
 - Contexts (Address spaces)
 - Threads
 - Communication links (global pointers)
 - Remote service requests
- Startpoints and Endpoints

Communications (Nexus)

Source; technologies for ubiquitous supercomputing...Foster et al, (CCPE 1997)

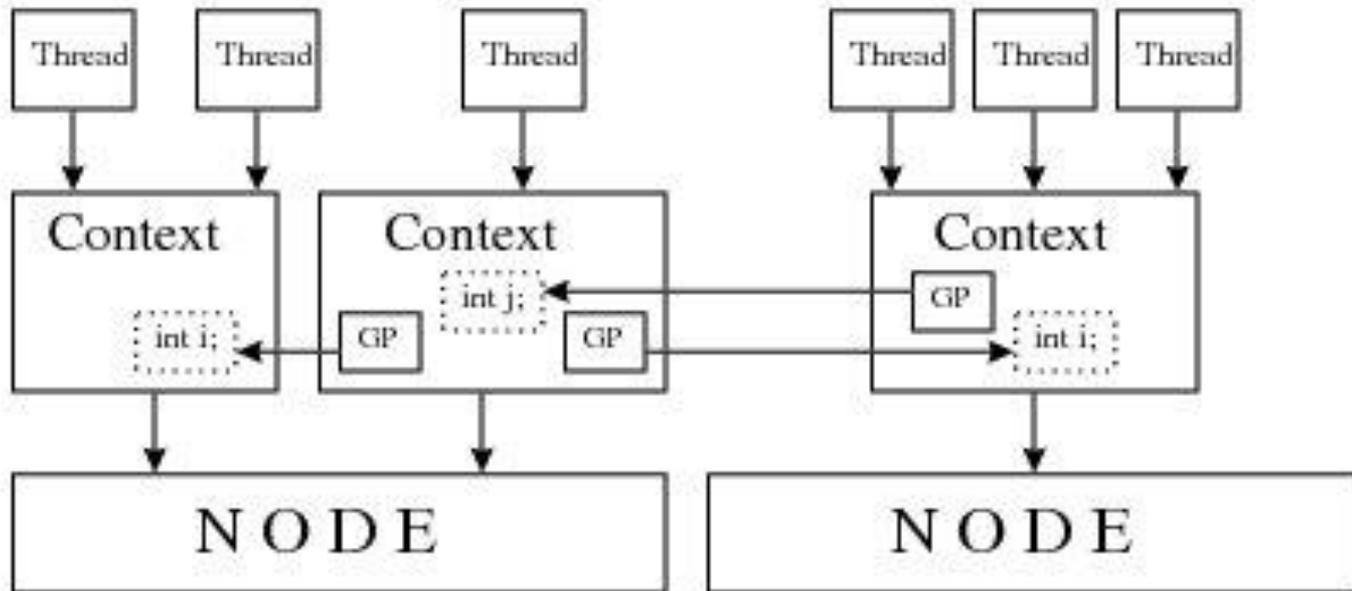


Figure 1. Nodes, Contexts, Threads, and Global Pointers

A Remote Service Request takes a GP, a proc name and data

Transfers the data to the context referenced by the GP

Remotely invokes the specified procedure (data and local portion of the GP arguments)

Information Services

(Metacomputing Directory Service - MDS)

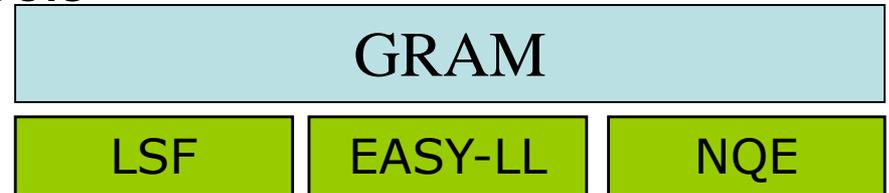
- Required information
 - Configuration details about resources
 - Amount of memory
 - CPU speed
 - Performance information
 - Network latency
 - CPU load
 - Application specific information
 - Memory requirements

Remote file and executable management

- **Global Access to Secondary Storage (GASS)**
 - basic access to remote files, operations supported include remote read, remote write and append
- **Remote I/O (RIO)**
 - distributed implementation of the MPI-IO, parallel I/O API
- **Globus Executable Management (GEM)**
 - enables loading and executing a remote file through the GRAM resource manager

Resource management

- Resource Specification Language (RSL)
- Globus Resource Allocation Manager (GRAM)
 - provides a standardized interface to all of the various local resource management tools that a site might have in place
- DUROC
 - provides a co-allocation service
 - it coordinates a single request that may span multiple GRAMs.



DUROC: Dynamically-Updated Request Online Coallocator

Authentication Model

- Authentication is done on a “user” basis
 - Single authentication step allows access to all grid resources
- No communication of plaintext passwords
- Most sites will use conventional account mechanisms
 - You must have an account on a resource to use that resource

Grid Security Infrastructure

- Each user has:
 - a Grid user id (called a Subject Name)
 - a private key (like a password)
 - a certificate signed by a Certificate Authority (CA)
- A “gridmap” file at each site specifies grid-id to local-id mapping

Certificate Based Authentication

- User has a certificate, signed by a trusted “certificate authority” (CA)
 - Certificate contains user name and public key
 - Globus project operates a CA

“Logging” onto the Grid

- To run programs, authenticate to Globus:
% grid-proxy-init
Enter PEM pass phrase: *****
- Creates a temporary, short-lived credential for use by our computations
Private key is not exposed past grid-proxy-init

Simple job submission

- globus-job-run provides a simple RSH compatible interface

```
% grid-proxy-init
  Enter PEM pass phrase: ****
% globus-job-run host program [args]
```