Application Classes Motivating Widely Distributed Computing Environments

Multi-disciplinary Simulations

Multi-disciplinary simulations provide a good example of a class of applications that are very likely to require aggregation of widely distributed computing, data, and intellectual resources.

Such simulations – e.g. whole system aircraft simulation and whole system living cell simulation – require integrating applications and data that are developed by different teams of researchers frequently in different locations.
Motivating Applications - Multi-Disciplinary Simulations

The research teams are the only ones that have the expertise to maintain and improve the simulation code and/or the body of experimental data that drives the simulations. This results in an inherently distributed computing and data management environment.

Consider a vision for Aviation Safety:

How do we simulate the entire commercial airspace of the country?

(Yuri Gawdiak (VNAS) and Bill McDermott, NASA Ames, John Lytle and Gregory Follen, NASA Glenn (NPSS)).

This vision is being approached through a set of increasingly complex and computationally intensive integrations:
Component simulations are combined to get a system simulation.

Multiple system simulations are coupled to represent pieces of a device.
Whole device simulations are produced by coupling all of the subordinate system simulations.

Devices are inserted into a realistic environment.
Devices and environment are combined for operational systems simulation.

Clearly such simulations will need to use aggregated computing, data, instrument, and intellectual resources across multiple NASA Centers.

**Issues for combining component simulations**

+ wrapping the simulation code
+ composing these codes
+ locating and coordinating resources for executing the multiple components and managing the resulting data (which is likely to be distributed)
Multi-Disciplinary Simulations

Issues for multi system simulations

+ multi-Center interactions - component parameters maintained by discipline experts
+ multiple sources of data and data expertise
+ shared compute and data resources

Vision for the Information Power Grid

CORBA and IDL are useful for addressing the first two points.

IPG is intended to provide the basic framework for resource sharing and management across sites:

+ discovery
+ scheduling
+ access to, and
+ policy enforcement

with respect to computing systems, data management, and collaboration systems
**What are Grids?**

Grids are tools, middleware, and services for
+ providing a uniform look and feel to a wide variety of computing and data resources
+ supporting construction, management, and use of widely distributed application systems
+ facilitating human collaboration and remote access and operation of scientific and engineering instrumentation systems
+ managing and securing the computing and data infrastructure
**What Grids Will and Will Not Do**

Grids provide common resource access technology and operational services deployed across virtual organizations. This **allows the possibility of sharing resources, but does not automatically permit it:**

- Local authorization models are not changed by the Grid.
- Common Grid technology will allow common views of resources and uniform access to resources, thereby permitting very large application systems to be built, and if policy permits, to share resources across sites and organizations.
Grids will enable large scale applications based on:

+ *Loosely coupled computations*: Simulation parameter sweeps and certain types of experiment data analysis involve initiating and managing 100s, 1000s, and 10000s of processes. Grids provide the access and mechanisms for using large numbers of computing and data resources for this type of calculation.

+ *Large scale pipelined applications*: Multi-component simulations involve executing multiple, coupled, medium to large scale simulations on multiple computing resources. Grids provide co-scheduling and data stream management to support this.
Grids will not, in the near term, enable very large, single problems such as CFD calculations to be spread across distributed systems.

+ To accomplish this we will need new approaches and algorithms that are tolerant of high and variable latency. There is R&D going on to address this issue in the long term.

Grids will not provide a lot of “free” resources.

+ To produce a highly capable science Grid organizations must place major resources on the Grid.

---

**Approach and Goals for NASA’s IPG**

- Grids are built through collaborative efforts, and at the same time facilitate collaboration: IPG is a collaboration among several NASA Centers and the NSF Supercomputer Center consortia (PACIs), with the Grid Forum providing “coordination” of many institutions world wide

- Deployment of existing technology (Globus [1], Condor [23], Grid portals [13], etc.) is providing for relatively rapid impact – IT/ANCS computing and storage resources are providing the prototype production IPG environment
Approach and Goals

- **Strong security is being provided from the start** in order to address authentication, authorization, and infrastructure assurance in open science networks for both applications and Grid services

How is IPG Being Accomplished?

Persistent operational environment that encompasses significant resources across the three Centers - there are groups at NAS whose responsibilities are:

- Grid Information Services
  - MDS operations
  - configuration databases
- IPG systems
  - Globus configuration and deployment
  - large system integration
How is IPG Being Accomplished?

• Security
  + X.509 CA operation
  + security model
  + GSI services

• Remote Data Access
  + IPG Archival Storage, GSIftp, SRB/MCAT
  + security model

• Portals and monitoring

• User Services
  + consulting
  + support model
  + documentation
  + CORBA support

• Condor

• Testing
  + regression testing, verification suites,
    benchmarks, and reliability/sensitivity analysis
How is IPG Being Accomplished?

- PBS
  + Global queuing and user-level queue management capability on top of Globus

- Networking
  + IPG testbed connection Ames, GRC, LaRC
  + high-speed testbed

- Accounting
  + account management (automated generation and maintenance mechanisms)
  + standardized accounting records

The State of IPG

- Computing resources: \( \approx 600 \) CPU nodes in half a dozen SGI Origin 2000s and several workstation clusters at Ames, Glenn, and Langley, with plans for incorporating Goddard and JPL, and \( \approx 200 \) nodes in a Condor pool

- Wide area network interconnects of at least 100 mbit/s

- Storage resources: 50-100 Terabytes of archival information/data storage uniformly and securely accessible from all IPG systems via MCAT/SRB and GSIftp / Gridftp
State of IPG: Baseline Operational System (cont.)

- **Globus** providing the Grid common services
  - Programming and program execution support
    - Grid MPI (via the Globus communications library)
    - CORBA integrated with Globus
    - global job queue management
    - high throughput job manager
    - Condor [23] ("cycle stealing" computing)
- A stable and supported operational environment
- Several "benchmark" applications operating across IPG (multi-grid CFD code, parameter study)
- Multi-Grid operation (applications operating across IPG and NCSA)
• IPG Functionality Tasks
  + CORBA in the IPG environment
  + Integration of Legion
  + CPU resource reservation
  + High Throughput Computing
  + Programming Services
  + Distributed debugging
  + Grid enabled visualization
  + Parameter study frameworks
  + Network bandwidth reservation

• Characteristic Applications
  + OVERFLOW
  + NPSS
  + INS3D
  + molecular analysis

IPG has met it's first three Level-1 milestones
**References and Acronyms**

[1] Globus is a middleware system that provides a suite of services designed to support high performance, distributed applications. Globus provides:
- Resource Management: Components that provide standardized interfaces to various local resource management systems (GRAM) manage allocation of collections of resources (DUROC). All Globus resource management tools are tied together by a uniform resource specification language (RSL).
- Remote Access: Components that enable remote access to files (GASS and RIO) and executables (GEM).
- Security: Support for single sign-on, authentication, and authorization within the Globus system (GSI) and (experimentally) authorization (GAA).
- Fault Detection: Basic support for building fault detection and recovery into Globus applications.
- Information Infrastructure: Global access to information about the state and configuration of system components of an application (MDS).
- Grid programming services: Support writing parallel-distributed programs (MPICH-G), monitoring (HBM), etc.

www.globus.org provides full information about the Globus system.


[13] A collaborative effort to enable desktop access to remote resources including, supercomputers, network of workstations, smart instruments, data resources, and more - computingportals.org


[17] The Grid Forum (www.gridforum.org) is an informal consortium of institutions and individuals working on wide area computing and computational Grids. Current working groups include Security (authentication, authorization), Scheduling and Resource Management, Grid Information Services, Application and Tool Requirements, Advanced Programming Models, Grid User Services and Operations, Account Management, Remote Data Access, Grid Performance


“STACS is ... responsible for determining, for each query request, which events and files need to be accessed, to determine the order of files to be cached dynamically so as to maximize their sharing by queries, to request the caching of files from HPSS in tape optimized order, and to determine dynamically which files to keep in the disk cache to maximize file usage.”


[22] Storage Resource Broker (SRB) provides uniform access mechanism to diverse and distributed data sources. http://www.sdsc.edu/MDAS/

[23] Condor is a High Throughput Computing environment that can manage very large collections of distributively owned workstations. http://www.cs.wisc.edu/condor/

[24] SCIRun is a scientific programming environment that allows the interactive construction, debugging and steering of large-scale scientific computations. http://www.cs.utah.edu/~sci/software/

