Performance Measurement and Modeling in Computational Grids

Ian Foster, Noam Freedman, Brian Toonen
Argonne National Laboratory
University of Chicago
Grid Applications

- High-performance computing in local or wide area networks, for applications that are
  - Distributed by design: e.g., collaborative environments, distributed data analysis, computer-enhanced instruments
  - Distributed by implementation: e.g., metacomputing, high-throughput computing
- Common challenge is to achieve & maintain performance guarantees in heterogeneous, dynamic environments
CNeph: Cloud Detection

"10 Gflop/sec, 5 Mb/sec, 10 minutes; rendering, 1 GB storage"

C. Lee et al., Aerospace Corp.
Performance-Robust Grid Applications

- Increase robustness of grid environment by using implicit or explicit models of application and system performance to
  - Identify resources required to meet application performance requirements
  - Select from among problem specification, algorithm, code variants
  - Establish hierarchical performance contracts
  - Select and manage adaptation strategies when performance contracts are violated
Example: Online Data Analysis

Broker(s) → Candidate resources

{net a: 100 Mb/s, MPP 1: 40 nodes, net b: 30 Mb/s, CPU: 0.5}

Reservation Co-allocator → Object creation Co-allocator

Requirements

Information Service

Exclusive

100 Mb/s

40 nodes

50 Mb/s

30 Mb/s

0.5 CPU

Modify Reservation

Online monitor

{ResHandles}

{ObjHandles}

{ResHandles}

Example: Online Data Analysis
Grids and Delphi, contd.

- **Goal:** Establish a technology framework for creation of performance-robust applications
- **Build on existing technology base**
  - Globus toolkit: Grid infrastructure
  - Paradyn/Pablo: Instrumentation, analysis
  - Autopilot: Sensor technology
  - HPC++, MPI: Application programming
- **Initial focus is on instrumentation and application studies**
Globus Toolkit
(Argonne/USC-ISI)

- A set of components providing core services required for grid applications
  - Information, resource management, security, communication, fault detection, data access, etc.

- Used to implement higher-level tools (e.g., MPI, CC++, CAVERN, HPC++, WebFlow, Apples, NEOS, NetSolve) and applications
Using the Globus Toolkit

Chemical engineering
Cosmology
Environmental hydrology
Molecular biology
Scientific instrumentation

Nanomaterials

Numerical libraries
HPC++
Legion
MPI
Web tools
CAVERN soft

Resource brokers

Condor
SWIG
Nimrod

Accounting
Communications
Information
Scheduling
Fault detection

Security
Instrumentation
QoS
Data access

Convex
SGI
SP
NT
LSF
Kerberos

Many Worlds
Recent Accomplishments

- Creation of GUSTO computational grid
  - 20+ sites, thousands of processors
- New resource management, security, monitoring, communication technologies
- Largest distributed interactive simulation
  - 100,000 entities on 13 supercomputers
- Release of Globus toolkit v1.0
- International experiments
- 1997 GII “Next Generation” award
The Globus Project
www.globus.org

GUSTO Computational Grid Testbed
as of November 1997

16 sites, 330 computers, 3600 nodes, 2 Teraflop/s, 10 application partners
Delphi and Grids: Initial Steps

- Develop information infrastructure for structural and performance information
- Allow for instrumentation of end-user applications under Globus
  - Paradyn-instrumented applications
- Develop Paradyn instrumentation for the Globus communication library (Nexus)
  - Bandwidth, loss rates, latency, jitter, etc.
- Study performance characteristics of real grid applications
MDS Object Class Example

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
}
Globus RM Architecture
Brokering, Co-allocation, Scheduling, Monitoring
Transparent Instrumentation

User request
allocate resource + create processes
(arch=sgi)(count=3)
(exec="myprog")...

GRAM Client Library

Gatekeeper

Job Manager

Local resources
Paradyn Integration

User request
(allocate resource + create processes)
(arch=sgi)(count=3)
(exec="myprog")
(paradyn=PDContact)
...

GRAM Client Library

Gatekeeper

Job Manager

Local resources

Paradyn
Resource Co-allocation

User request \rightarrow \text{Co-allocator}

\rightarrow \text{(multi-component application)}

\rightarrow \text{Individual GRAM requests}

\rightarrow \text{GRAM}

\rightarrow \text{Resource}

\rightarrow \text{GRAM}

\rightarrow \text{Resource}

\rightarrow \text{GRAM}

\rightarrow \text{Resource}
Paradyn Integration

User request → Co-allocator → (multi-component application) → Individual GRAM requests → Resource → GRAM → Resource → GRAM → Resource → Paradyn
Nexus Transform Modules

- Communication link: startpoints → endpoints
- Comm methods selected on per-CL basis
- User-managed transforms for encryption, compression ... and instrumentation

**Diagram:**
- **Send**
  - Prepend a module-specific header and possibly mutate the message
- **SP**
- **EP**
  - Pull out header, un-mutate the message, do computations
  - Invocation
Nexus Instrumentation

- Insertion of sequence numbers and timestamps for the header
- Computation of loss rates, latency, jitter, etc. on the endpoint.
- Dynamic insertion of transforms?

Diagram:
- send
  - SP
    - Expand message with sequence number, timestamp
  - EP
    - Compute loss rate, jitter, etc.
- invocation
Application Studies

- **Distributed computing**
  - SF-Express battle-field simulation
  - MPICH-G applications

- **Remote I/O**
  - RIO remote MPI-I/O library
  - Tardis remote data access library

- **Collaborative environments**
  - CAVERNsoft collaborative tele-immersion
SF-Express: Distributed Interactive Simulation

- Performance issues:
  - Organization of computational structure
  - Network requirements
  - Choice of communication methods and protocols
  - Scalability

P. Messina et al., Caltech
Application Example: CAVERNsoft Collaborative Design

- Allows users to manipulate shared virtual space
- Multiple flows
  - Control, Text, Video, Audio, Database, Simulation, Tracking, Haptics, Rendering
  - 1-10000 Mb/s
- Complex QoS issues
CAVERNsoft Applications

CAVE5D, ODU

NICE, UIC

Images courtesy Jason Leigh, EVL/UIC
Next Steps

- Integration with application-level tools
  - HPC++, MPI
- Performance models of individual components
  - See Dennis Gannon’s talk
- More detailed application studies
- Experiments in adaptation