Automatic Deployment of MPI Applications on a Computational Grid

Sébastien Lacour, Christian Pérez
IRISA / INRIA, France

Computational Grids

- Compute and storage resources:
  - Geographically distributed
  - Interconnected over a WAN
  - Not dedicated to one application
- Network bandwidth increase
- Potentially huge computer power
- Issues: security, heterogeneity
  - Compute resources
  - Network technology, performance, and topology / hierarchy
- Complex environment
Computational Grid Usage

- One of the goals: usage transparency
- In particular for application deployment
MPI Parallel Applications on Computational Grids

- MPI implementations for grids:
  - MPICH-G2, MagPIe, PACX-MPI, etc.

- Topology-aware collective operations:
  - Take network hierarchy into account
  - Optimize Broadcast, Reduce, Barrier, Gather, etc.
  - Minimize communications on slow networks

- Provide access to the underlying network topology (MPICH-G2)
  - MPI programmer can optimize his parallel algorithm
  - Dynamically create groups of communications
**MPI Deployment on Grids: Complexity Accumulation (1)**

- Select heterogeneous grid resources
  - OS and architecture compatibility
- Map application processes on selected compute nodes
- Select compatible compiled executables
- Upload executables, stage input files in
- Launch processes on remote computers
MPI Deployment on Grids: Complexity Accumulation (2)

- Set the configuration parameters
  - Provide network topology information to the MPI library
- MPICH-G2: environment variables
- MagPIe, PACX-MPI: description file to stage in
- All that **manually**...
  - Too complicated for grids!

No way!
MPICH-G2 Example: RSL

(& (resourceManagerContact="cluster.teragrid.org")
  (count=10)
  (environment=(GLOBUS_DUROC_SUBJOB_INDEX 0)
    (LD_LIBRARY_PATH "/usr/globus/lib")
    (GLOBUS_LAN_ID my_LAN))
  (executable="/homes/users/smith/myapp_i386")
)

(& (resourceManagerContact="node.othersite.edu")
  (count=20)
  (environment=(GLOBUS_DUROC_SUBJOB_INDEX 1)
    (GLOBUS_LAN_ID my_LAN))
  (directory="/home/ux394/")
  (executable="/home/ux394/mpi_proc_sparc")
)
Automatic Application Deployment on Grids

- Our objective: hide all that complexity
- Automatic deployment tool

Input:
- Packaged application (self-described)
- Description of grid resources

Run the application automatically
- Cluster: `mpirun -machinefile ... -np 16 my_appl`
- Grid: `grid_deploy -resources ... -application my_appl`
Automatic Deployment: Overview

Resource Description

Application Description

User-level Constraints

Deployment Planning

Deployment Plan Execution

Input of the Deployment Tool: Application Package

- MPI application
  - Packaged (ZIP file)
  - Self-described
    - Number of MPI processes
    - Various compiled implementations
- Ongoing work
  - Count, groups

Application type: MPI
Process count: 16

Executable
- Sparc Solaris
- PowerPC AIX
- i386 Linux
Input of the Deployment Tool: Grid Resource Description

- Distributed information
  - OS, architecture, CPU #
  - Network topology and performance characteristics
- A Network Topology Description Model for Grid Application Deployment, Grid2004, Pittsburgh, PA, Nov. 2004
Input of the Deployment Tool: User-Level Constraints

- Keep a certain level of control on the automatic deployment process
- Not specific to the application ("user's comfort")
  - Minimize execution time
  - Run the application close to a visualization site
- Example: no groups under 8 processes
Deployment Planning

- Heart of the automatic deployment tool
- Select grid resources
- Place application processes on computers
- Select compiled executables
  - Check OS and architecture compatibility
- Select a launch method (SSH, Globus GRAM)
- Produce a deployment plan
Deployment Plan Execution

Executable
Sparc
Solaris

Executable
i386
Linux

10-node cluster
Sparc/Solaris

10-node cluster
i386/Linux

6-node cluster
PowerPC/AIX

Internet

Less than 8 nodes
(no group under 8 processes)
Application Configuration

- Set configuration parameters
  - Provide network topology information
    - The planner has this information: placement decisions were based on it
  - MPICH-G2 (3 network hierarchy levels)
    - processes in a cluster
    - clusters in a LAN (local-area network)
    - LANs in a WAN (wide-area network)
  - MagPIe, PACX-MPI: 2 network hierarchy levels
ADAGE

- Automatic Deployment of Applications in a Grid Environment
  - http://www.irisa.fr/paris/ADAGE/
- Simple user-level constraints
- Already gained experience with distributed component-based applications
  - Deploying CORBA Components on a Computational Grid: General Principles and Experiments Using the Globus Toolkit, CD2004, Edinburgh, Scotland
MPI Application Deployment on a Grid with ADAGE

- Same simplicity as on a single cluster
  - `grid_deploy -resource my_grid`
    - `application my_appl.zip`
    - `usr_lvl my_usr_lvl_contraints`
- Resource selection among "my_grid"
- Placement and implementation selection among "my_appl.zip"
- Network topology information configuration
Conclusion

- MPI applications on computational grids
  - Complex to deploy
  - Need configuration: network topology
- Automatic deployment of MPI applications
  - Deployment planning
    - Resource selection, process placement, launch method selection, implementation selection
  - Transmit topology information to application
- Validation in ADAGE
Perspectives

- How to package MPI applications?
  - Ongoing work
- How about parallel components?
  - Distributed components made of an MPI program
- Re-deployment
  - Checkpoint/restart after failure or ETA
  - MPI-2 standard has MPI_Comm_spawn
Questions?

Thank you!