GridCOMP – ProActive/GCM tutorial and and Hands-On Grid Programming

Cédric Dalmasso, Antonio Cansado and Denis Caromel

INRIA - OASIS Team
INRIA -- CNRS -- I3S -- Univ. of Nice Sophia-Antipolis, IUF

IV Grid@Work, Tsinghua University, Beijing
General agenda

- Talk: A short introduction to ProActive middleware
- Practical session: ProActive fundamentals
- Talk: ProActive / GCM
- Practical session: ProActive / GCM
- Talk: IDE
- Talk: Autonomic
- Practical session: Autonomic
Short Introduction to ProActive
Agenda (Update It)

- Overview
- Programming
- Deploying
- What else?
- GUIs and tools
- Applications
- Conclusion
Overview
The team : 30+ members

- OASIS Team at INRIA in Nice, France
- Joint team INRIA / CNRS / Univ. Nice

- Team leader: Denis Caromel
  - 3 professors
  - 2 researchers
  - 1 postdoc
  - 7 engineers
  - 7 PhD students
  - + Interns, visiting researchers…

- Collaborations: ObjectWeb, CoreGRID, GridCOMP etc..
The library

- Originates from work on Eiffel //
  - Started in 1999

- Official releases ~ every 6 months
  - ProActive 3.2.1 released in April 2007
  - (Version 3.9 soon)

- Metrics:
  - 2000 classes, ~ 300,000 LOC (160,000 NCLOC)

- Compliant with several (de facto) standards

- ProActive startup: ActiveEon
  - Training, Consulting, Integrating and Support
Theory

- Henrio & Caromel

- ASP Calculus:
  - Asynchronous Sequential Processes
  - Based on Sigma-Calculus (Abadi-Cardelli)

- Formal Proofs of determinism (in greek)

- Implemented in ProActive
ProActive’s Framework in a nutshell

Open Source + PROFESSIONAL SUPPORT
Inside ProActive

- IDE
- PROGRAMMING & COMPOSING
- DEPLOYMENT
Rationale

- Distributed programming entities
- Parallel processes
- Asynchronism
- Synchronization facilities
Grid Computing with \textit{ProActive}

Hierarchical Deployment

Challenges: Programming Model, Scale, Latency, Heterogeneity, Versatility (protocols, firewalls, etc.)
Programming
ProActive: Model

- Active objects: structuring entities (subsystems)
  - Passive objects (fields)
  - 1 thread / AO
  - Request queue
- Full control to serve incoming requests (reification)
- Sequential processing
- Typed entities (safe)
- Asynchronous Communication between active objects
- No shared passive objects - deep-copy of parameters
Creation, Invocation and Sync.

- A `ag = newActive ("A", [...], VirtualNode)`
- V `v1 = ag.foo (param);`
- V `v2 = ag.bar (param);`

... 

JVM
- `v1.bar(); //Wait-By-Necessity`
Automatic Continuations

V = b.bar ()

c.gee (V)

Transferable futures
Concurrent activities
Data-flow synchronization
Explicit Synchronizations

A ag = newActive ("A", [...], VirtualNode)
V v = ag.foo(param);
...
v.bar(); //Wait-by-necessity

• Explicit Synchronization:
  • - ProActive.isAwaited (v);   // Test if available
  • - ProActive.waitFor (v);   // Wait until available

• Vectors of Futures:
  • - ProActive.waitForAll (Vector); // Wait All
  • - ProActive.waitForAny (Vector); // Get First
Architecture: a Meta-Object Protocol

VM1

VM2

generated on-the-fly

communication layer

proxy

Stub_a

b

future

a

mobility

asynchronism

service meta-objects

Architecture : a Meta-Object Protocol
Active Objects and Groups

- A ag = newActiveGroup ("A", [...], VirtualNode)
- V v = ag.foo(param);
- ...

V

//Wait-by-necessity

Typed Group
Java or Active Object

Group, Type, and Asynchrony are crucial for Cpt. and GRID
Broadcast and Scatter

Broadcast is the default behavior
Use a group as parameter, Scattered depends on rankings

```java
ag.bar(cg);  // broadcast cg
ProActive.setScatterGroup(cg);
ag.bar(cg);  // scatter cg
```
Deploying
Deployment: an abstract model

- Problem:
  - Heterogeneous environments/protocols
  - Scalability issues (large number of hosts / latency)

- A key principle:
  - Separate design from deployment infrastructure
  - Nothing about infrastructure, protocols or physical resources in the app. code

Creation Protocols
- ssh, gsissh, rsh, rlogin
- lsf, pbs, sun grid engine, oar, prun
- globus(GT2, GT3 and GT4), unicore, glite, arc (nordugrid)

Registry/Lookup and Comm. Protocols
- rmi, http, rmissh, ibis, soap

Files Transfers
- scp, rcp
- unicore, arc (nordugrid)
- other protocols like globus, glite will be supported soon

XML deployment file ⇒ Virtual Node (VN)
Activating a XML Desc.

```
ProActiveDescriptor pad = ProActive.getProactiveDescriptor(String xmlFile);
// Returns a ProActiveDescriptor object from the xml file
VirtualNode dispatcher = pad.getVirtualNode("Dispatcher");
// Returns the VirtualNode Dispatcher as a java object
dispatcher.activate();
// Activates the VirtualNode
Node node = dispatcher.getNode();
// Returns the first node available among nodes mapped to the VirtualNode

C3DDispatcher c3dDispatcher = newActive("C3DDispatcher", param, node);
```
Same Application, Many Deployments

One Host

Local Grid

Distributed Grids

Write once, deploy everywhere …
What else?
Other important features

- Component framework, the ProActive/GCM (just later!)
- P2P environment + Branch & Bound API
- Legacy code wrapping (MPI!)
- Middleware services:
  - Fault Tolerance
  - SOA integration (OSGI compliancy)
  - Migration
  - Load Balancing
  - Security
GUIs and tools
CoreGRID Beijing’07

Monitoring View

Job Monitoring View

Display topology ○ Proportional ○ Ratio ○ Failure

Monitoring

15:09:15 → NodeObject id=Node-455195381 already monitored, check for new active objects.
IC2D xml editor

Launches the deployment descriptor
Gives Information
Kills the application

Launcher console

13:51:03 -> File selected: /user/jmlegait/home/workspace/org.objectweb.proactive
13:51:07 -> File selected: /user/jmlegait/home/workspace/org.objectweb.proactive.ic2d.launcher
13:54:05 -> /user/jmlegait/home/workspace/org.objectweb.proactive.ic2d.launcher
13:54:05 -> Open the log4j console for more details
Tim It
Remote Call – JIT options (1000 iterations, warmup =1000)
Some Applications
Java 3D Electromagnetism

- Maxwell 3D equation solver, Finite Volume Method (FVM)
- Pre-existing Fortran MPI version: EM3D (CAIMAN team @ INRIA)
- 300+ machines at the same time (Intranet and cluster)
- Large data sets: 150x150x150 (100 million facets)
JECS : A Generic Version of Jem3D
Code Coupling:
Vibro Acoustic (courtesy of EADS)
Scilab Grid Toolkit

[Image of a software interface showing a grid of tasks with columns for Task ID, Status, Engine, and other details. The interface includes options for pending, executing, and terminated tasks.]
Post Production movie processing

Frames
Making of Nissan
Post Production movie processing
Large Scale Deployments

**Grid Plugtests** 2004, 2005 & 2006

- 20 to 40 sites worldwide
- 100 GFlops in 2004
- 1700 GFlops in 2006
- 4130 cores (2111) in 2006

- IBM, Sun, Bull, Apple, x86, 64bits...
- Linux, Windows, Solaris, MacOS
- ssh, rsh, sshGSI, GRAM
- PBS, LSF, SGE, OAR, Globus, Prun

**Grid’5000 - DAS etc...**

- P2P INRIA Infrastructure
- 53 years computation in 6 months on 200+ machines
On-going activities

- Programming model
  - Grid Component Model, adaptive components
  - Model checking, formal verification of behavioral properties
  - High level parallelism patterns (skeletons)

- Deployment
  - OSGi gateways
  - MPI / native codes wrapping
  - Easier specification, Scheduler

- Middleware services
  - Security at application level
  - Distributed garbage collection

- Industrial strength product
  - Quality development process, Support, Services

---

ProActive/GCM User Group and Contest at GRIDs@Work 2007: IV GRID PLUGTESTS,
Joint European Union/China GRID
28 Oct.-2 Nov. 2007, Beijing, China
Conclusion

• Usability - High Level Abstractions - Latency

Strong programming model
formal model, active objects, groups, components

Versatile deployment framework
Interfaced with Grid & cluster standards

Pluggable middleware services
Mobility, fault tolerance, security etc…
Let’s practice!

http://proactive.objectweb.org