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…

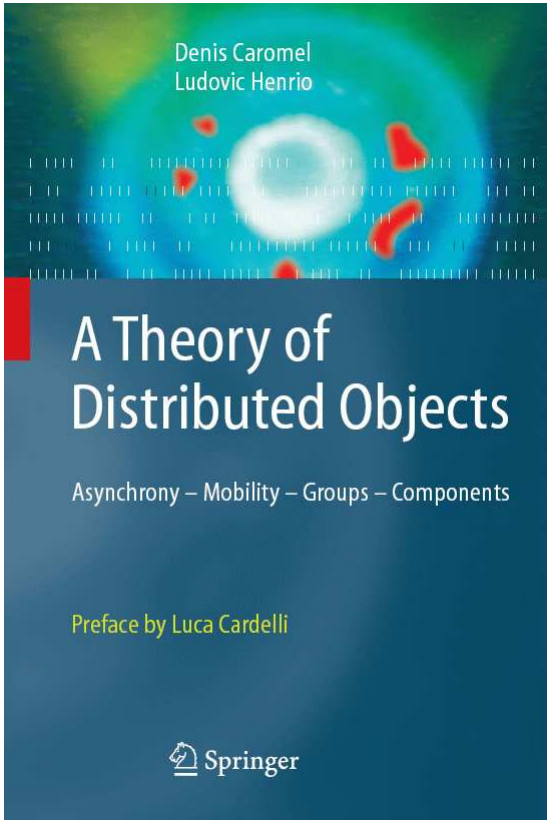
- Collaborations: ObjectWeb, CoreGRID, GridCOMP etc..

The library

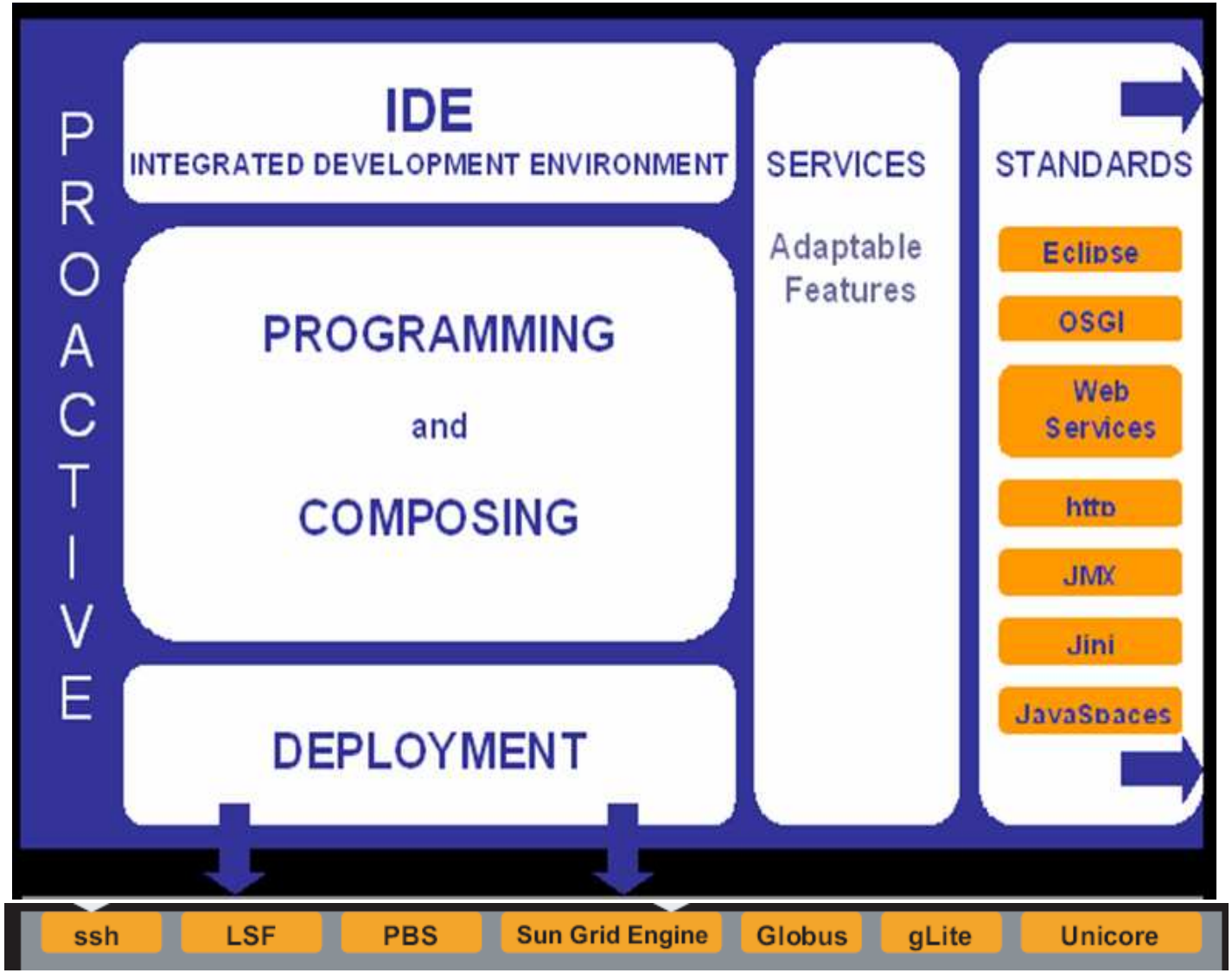
- Originates from work on Eiffel //
  - Started in 1999
- Official releases ~ every 6 months
  - ProActive 3.2.1 released in April 2007
  - (Version 3.9 soon)
- Metrics:
  - 2000 classes, ~ 300.000 LOC (160.000 NCLOC)
- Compliant with several (*de facto*) standards
- ProActive startup: ActiveEon
  - 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



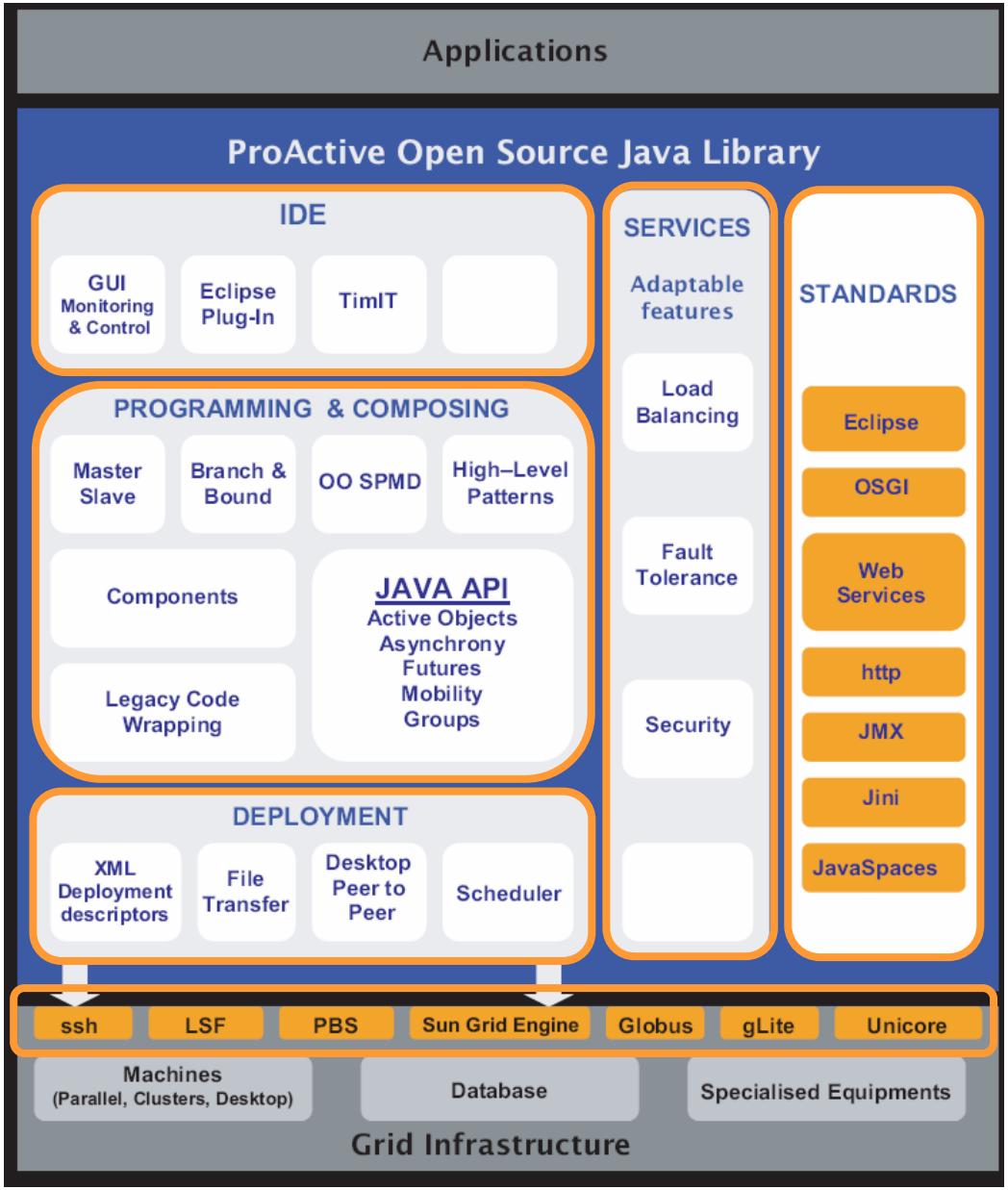
ProActive's Framework in a nutshell



Open
Source
+
PROFESSIONAL
SUPPORT

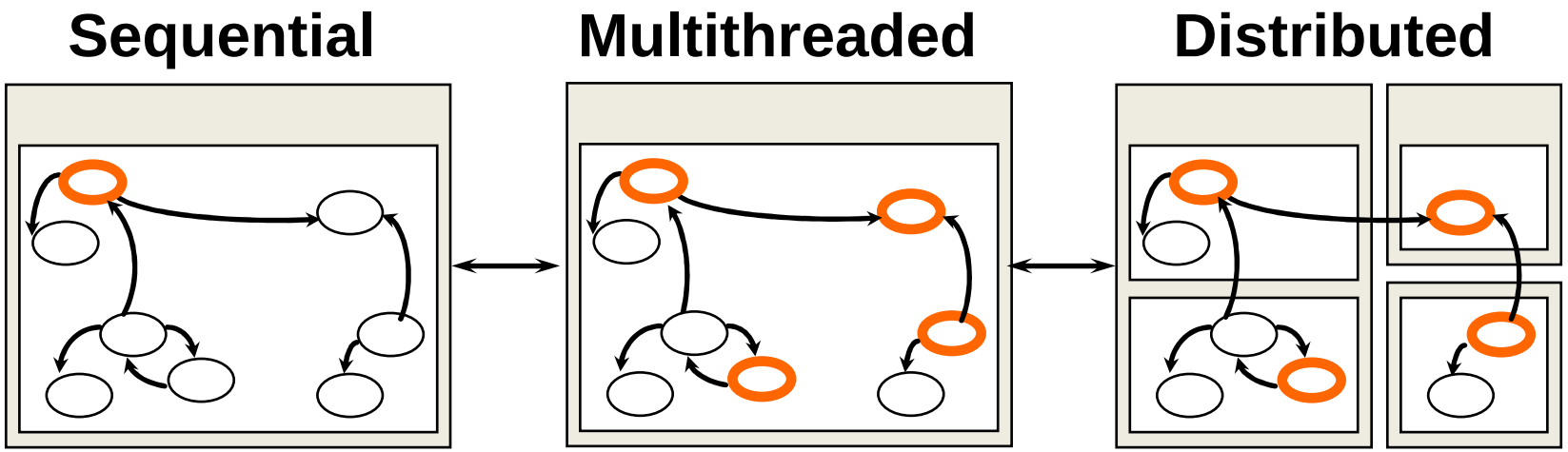
Inside *ProActive*

- IDE
- PROGRAMMING & COMPOSING
- DEPLOYMENT

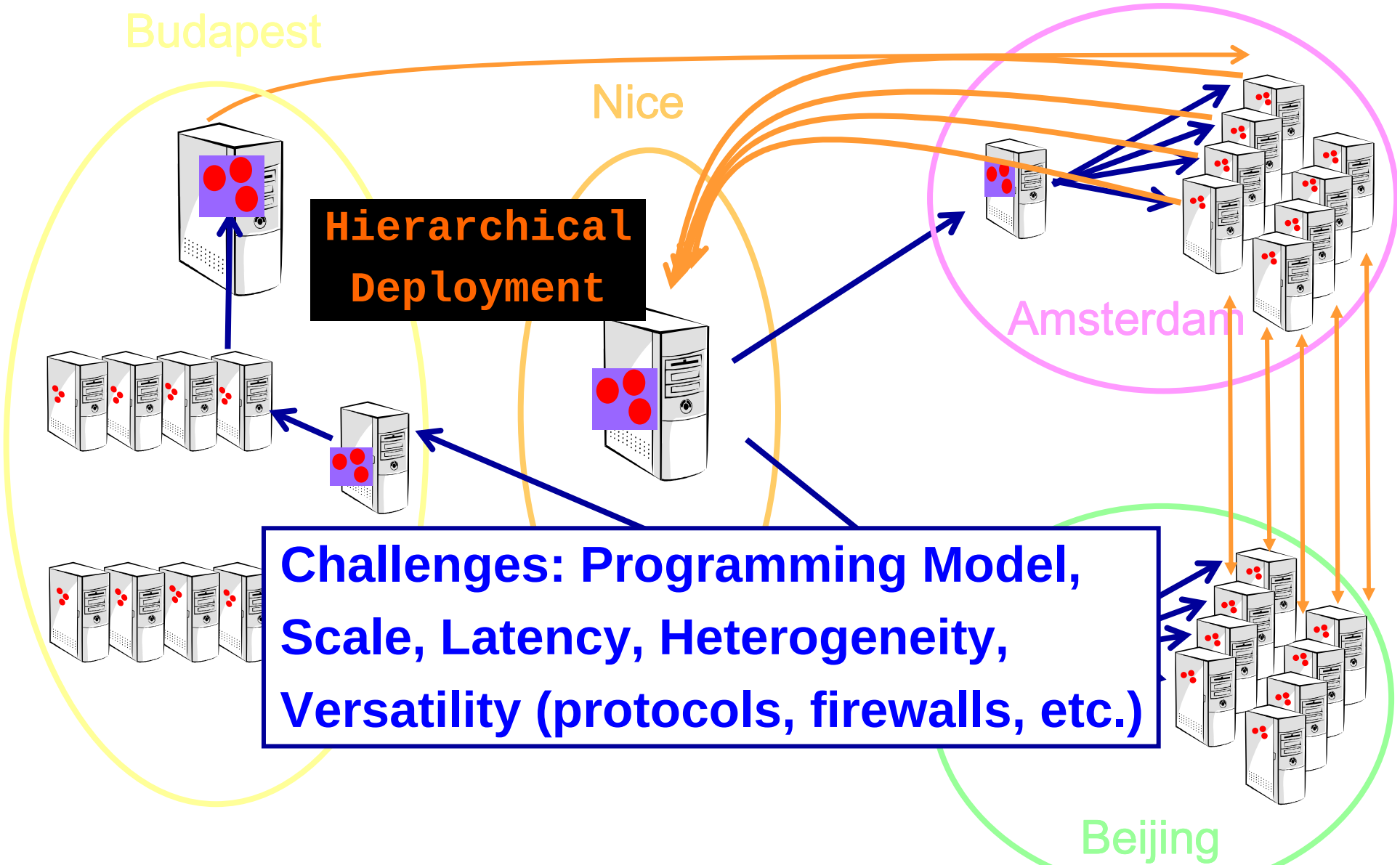


Rationale

- Distributed programming entities
- Parallel processes
- Asynchronism
- Synchronization facilities



Grid Computing with *ProActive*



Programming

ProActive: Model

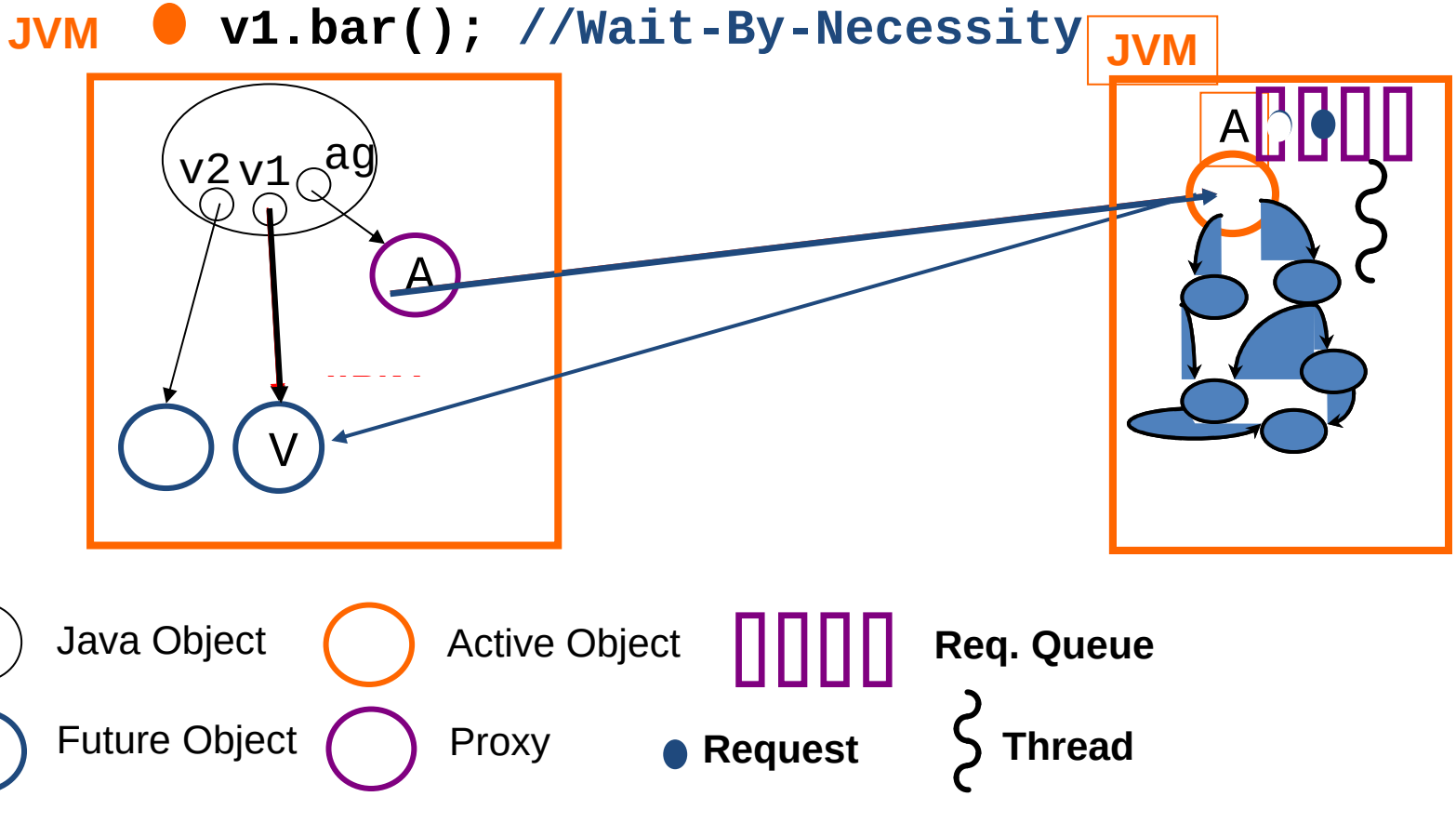
- Active objects : structuring entities (subsystems)
  - Passive objects (fields)
  - 1 thread / AO
  - Request queue
- Full control to serve incoming requests (reification)
- Sequential processing
- Typed entities (safe)
- Asynchronous Communication between active objects
- No shared passive objects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

Creation, Invocation and Sync.

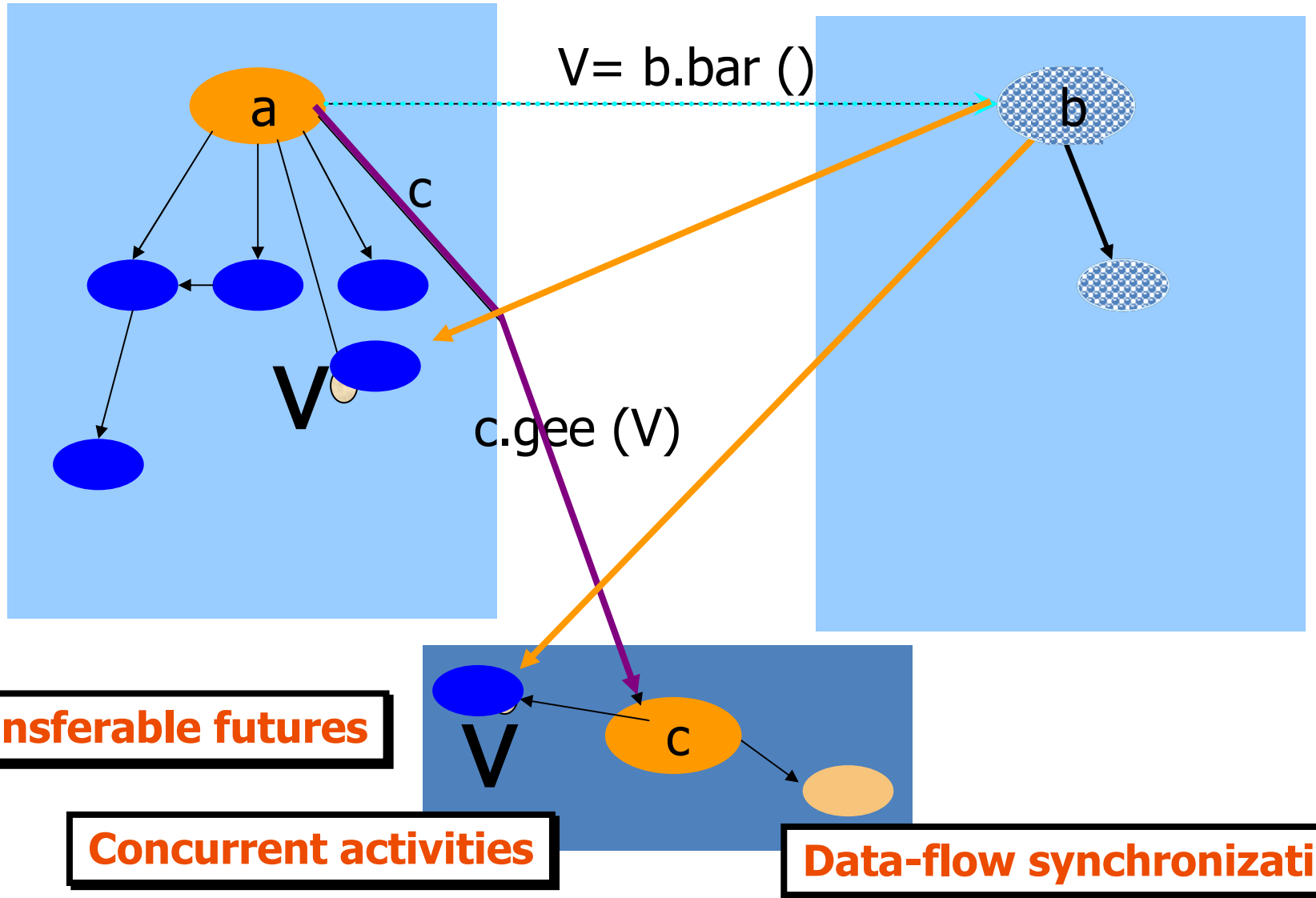
- `A ag = newActive ("A", […], VirtualNode)`
- `V v1 = ag.foo (param);`
- `V v2 = ag.bar (param);`

```
...
```

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



Automatic Continuations

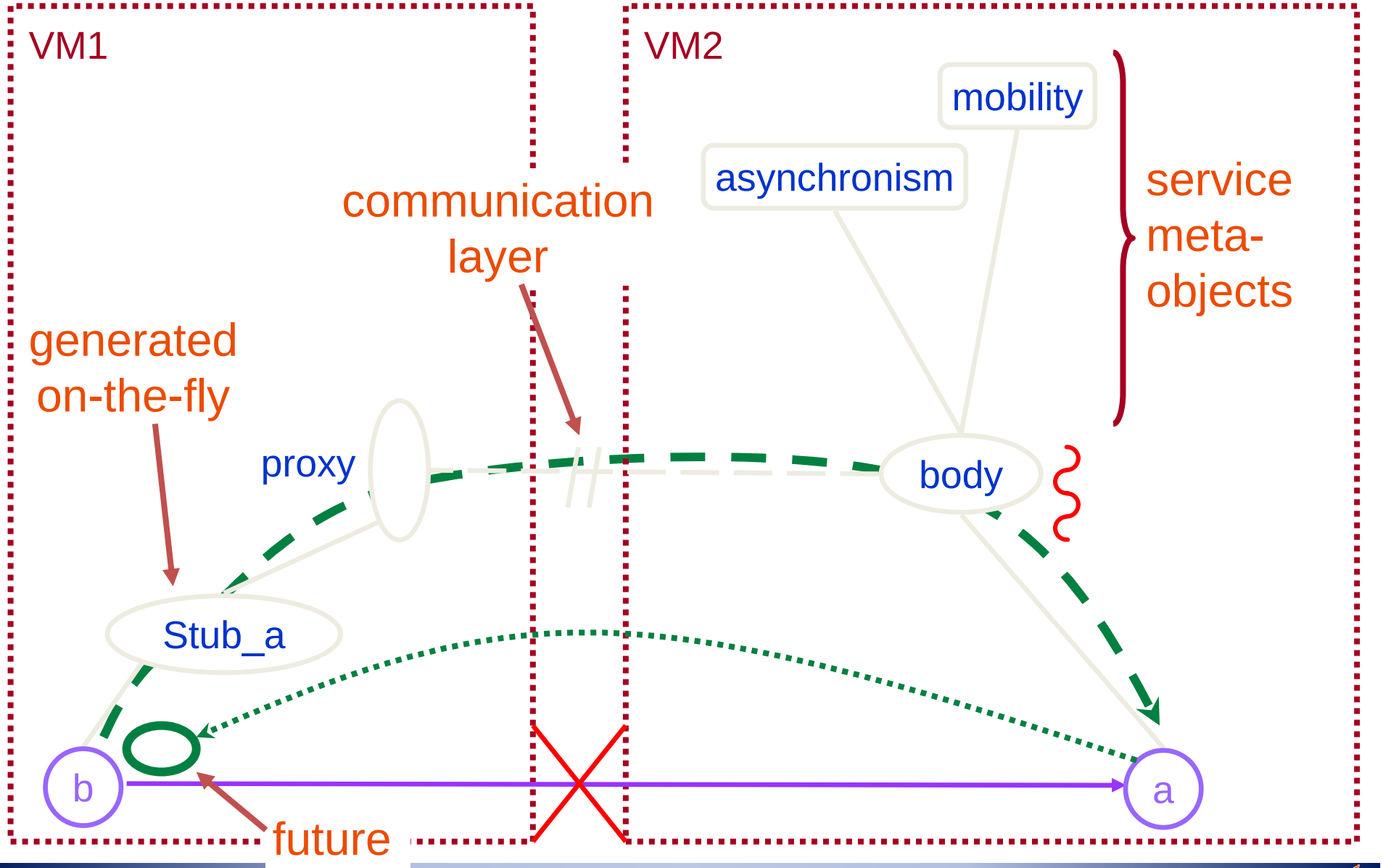


Explicit Synchronizations

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

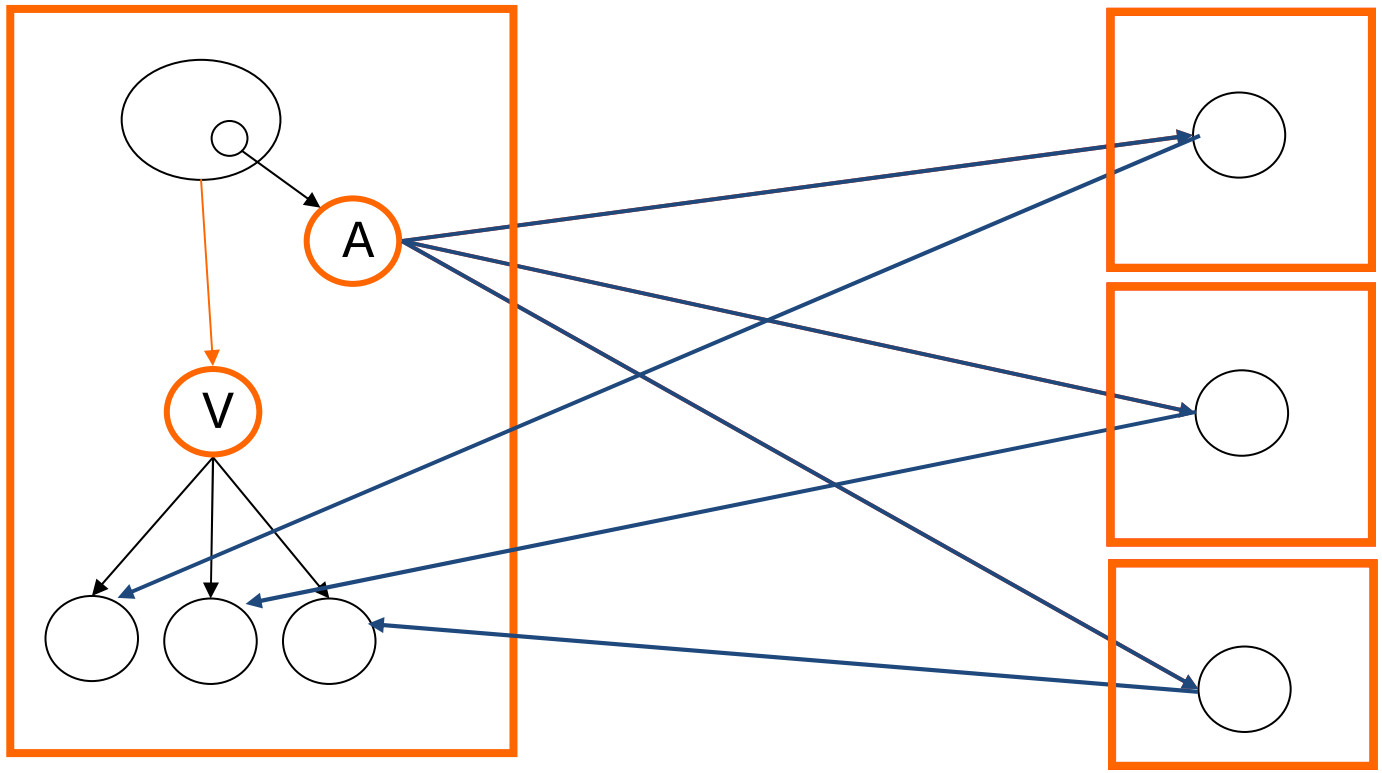
- Explicit Synchronization:
  - - ProActive.isAwaited (v); // Test if available
  - - ProActive.waitFor (v); // Wait until availab.
- Vectors of Futures:
  - - ProActive.waitForAll (Vector); // Wait All
  - - ProActive.waitForAny (Vector); // Get First

Architecture : a Meta-Object Protocol



Active Objects and Groups

```
A ag = newActiveGroup (“A”, […], VirtualNode)
V v = ag.foo(param);
…
v.bar(); //Wait-by-necessity
```

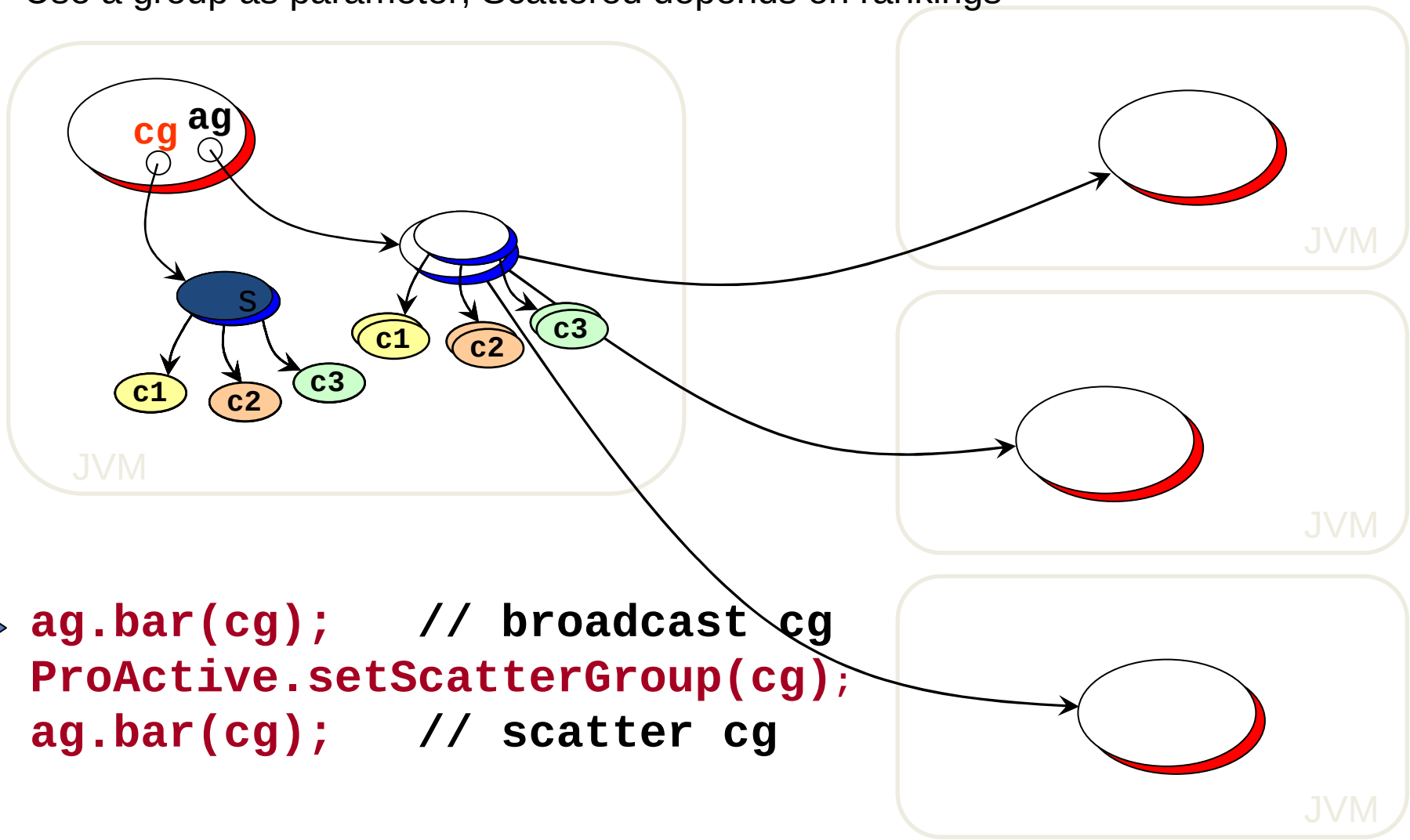


Typed Group Java or Active Object

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




```
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

XML deployment file ⇨

Virtual Node (VN)

Creation Protocols

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

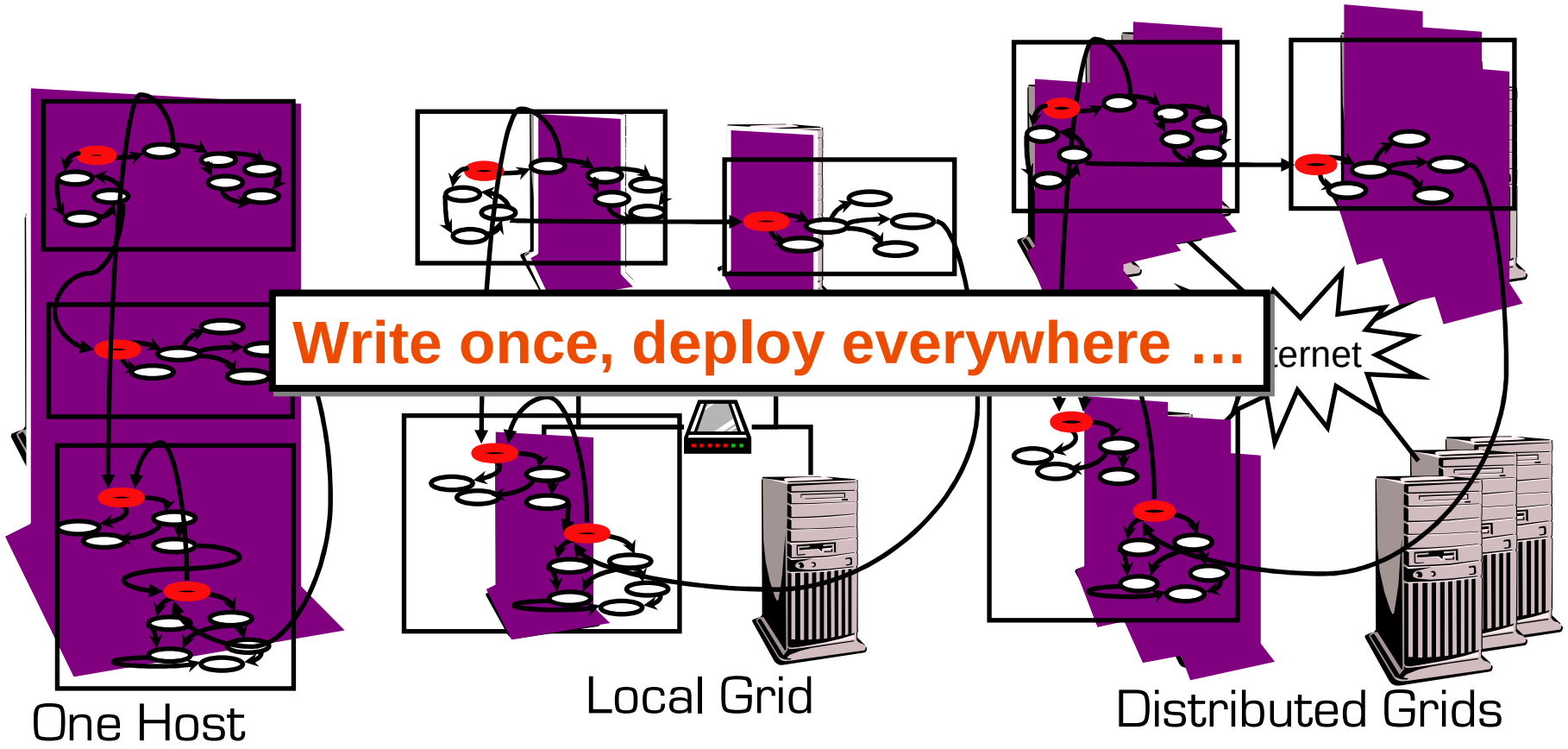
- Registry/Lookup and Comm. Protocols
  - rmi, http, rmissh, ibis, soap
- Files Transfers
  - scp, rcp
  - unicore, arc (nordugrid)
  - other protocols like globus, glite will be supported soon

Activating a XML Desc.

```
•.....
• <virtualNodesDefinition>
• <virtualNode name='Dispatcher'/>
 .....
```

```
•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

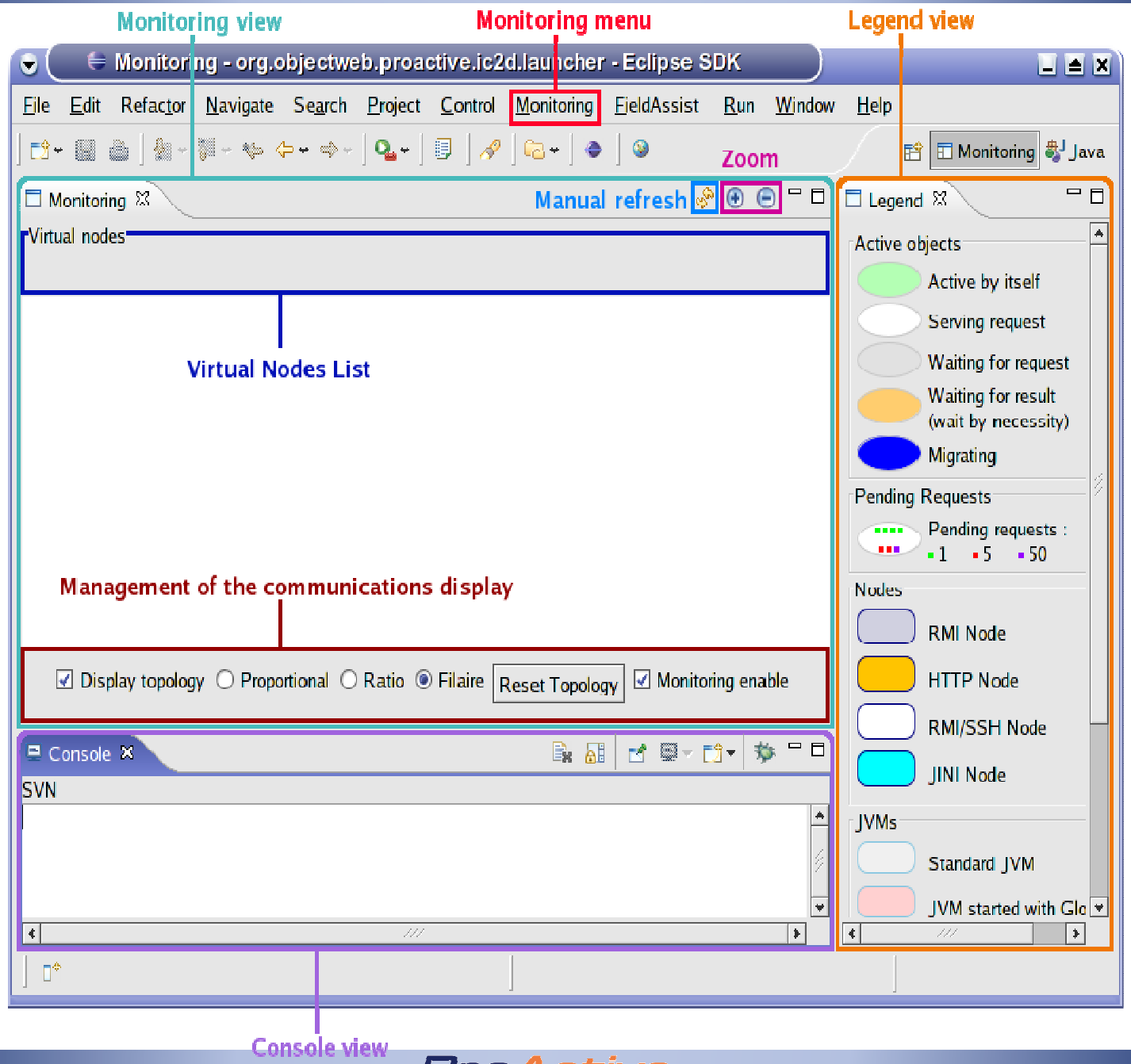


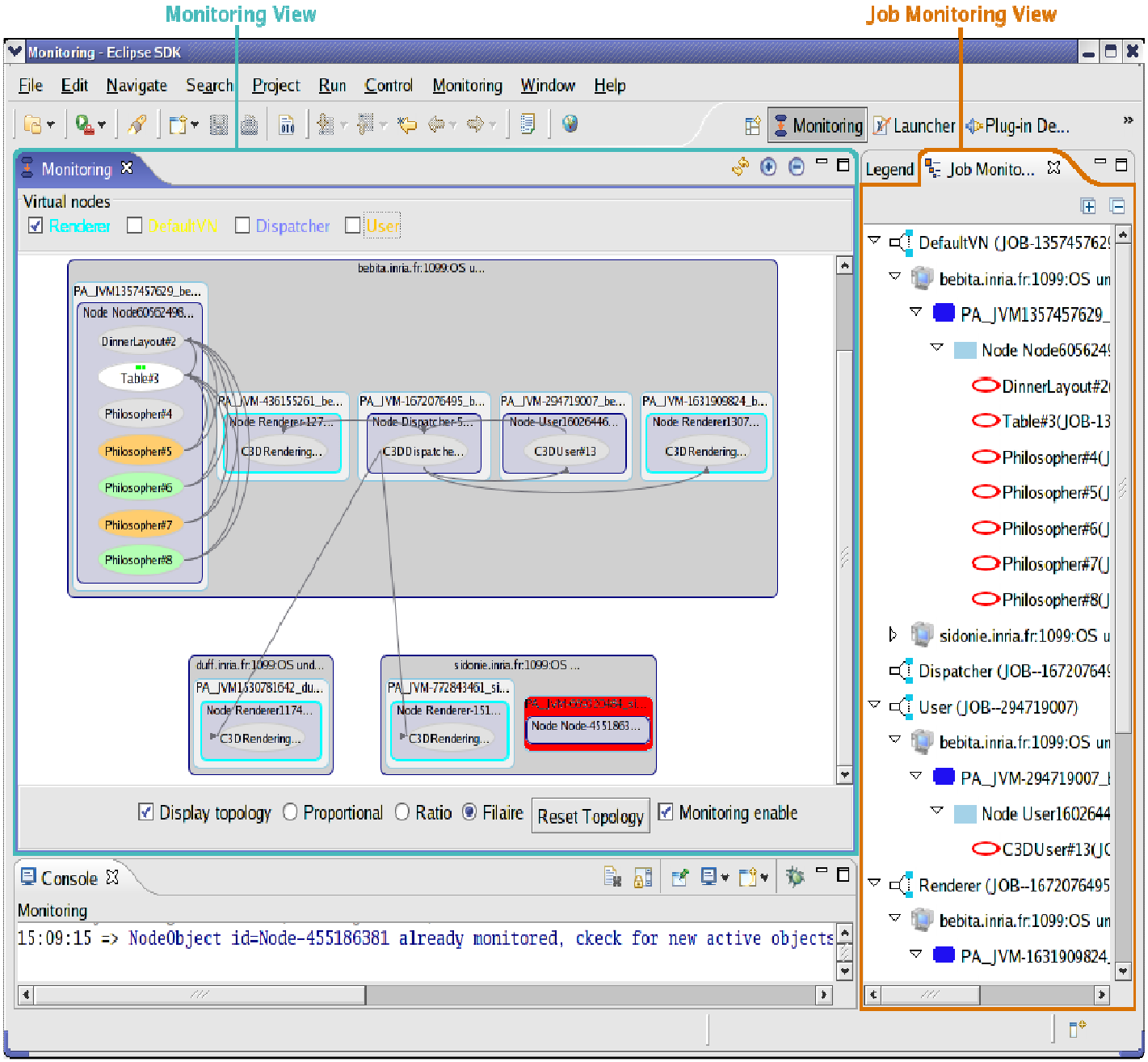
What else?

Other important features

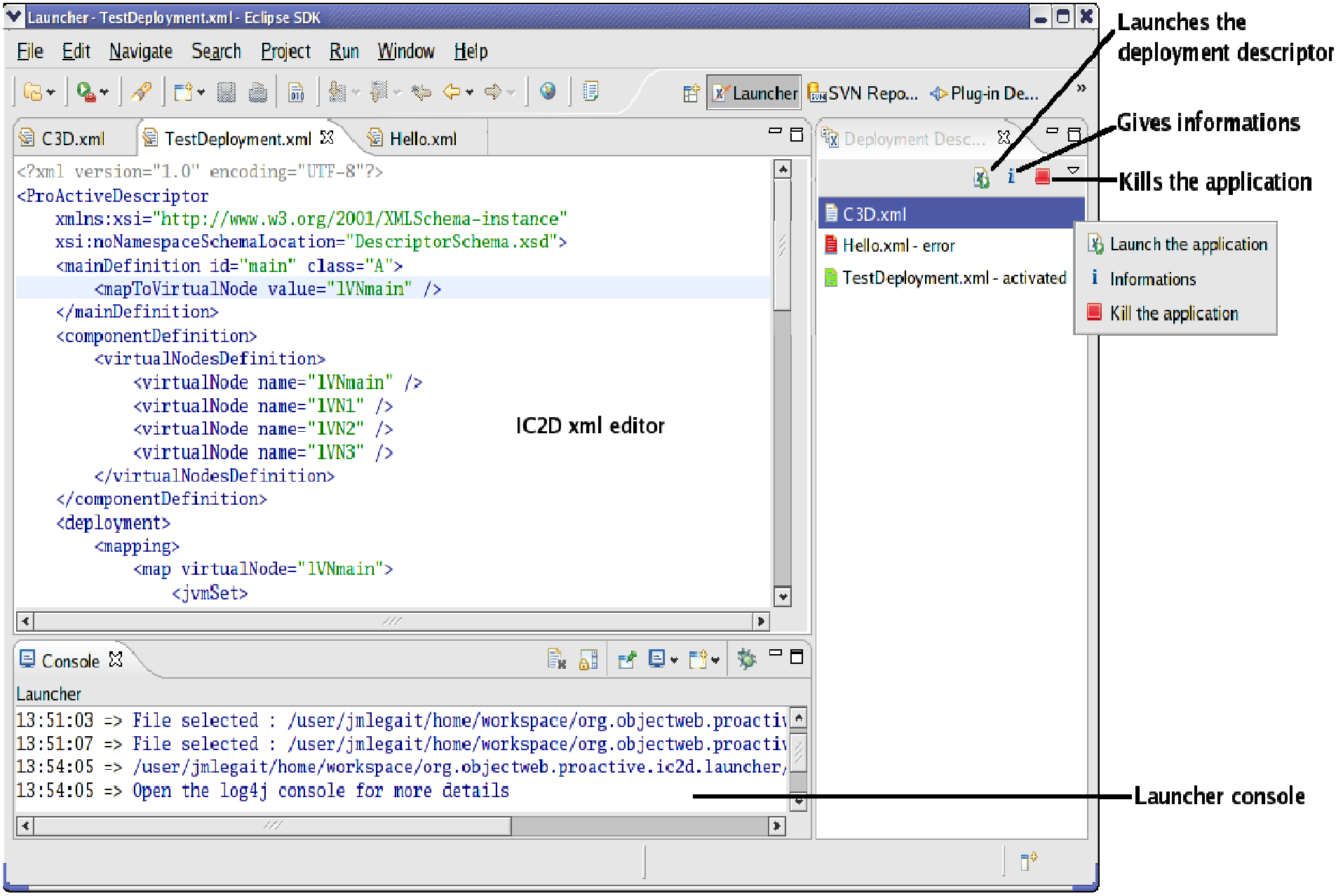
- Component framework, the ProActive/GCM (just later!)
- P2P environment + Branch & Bound API
- Legacy code wrapping (MPI!)
- Middleware services:
  - Fault Tolerance
  - SOA integration (OSGI compliancy)
  - Migration
  - Load Balancing
  - Security

GUIs and tools

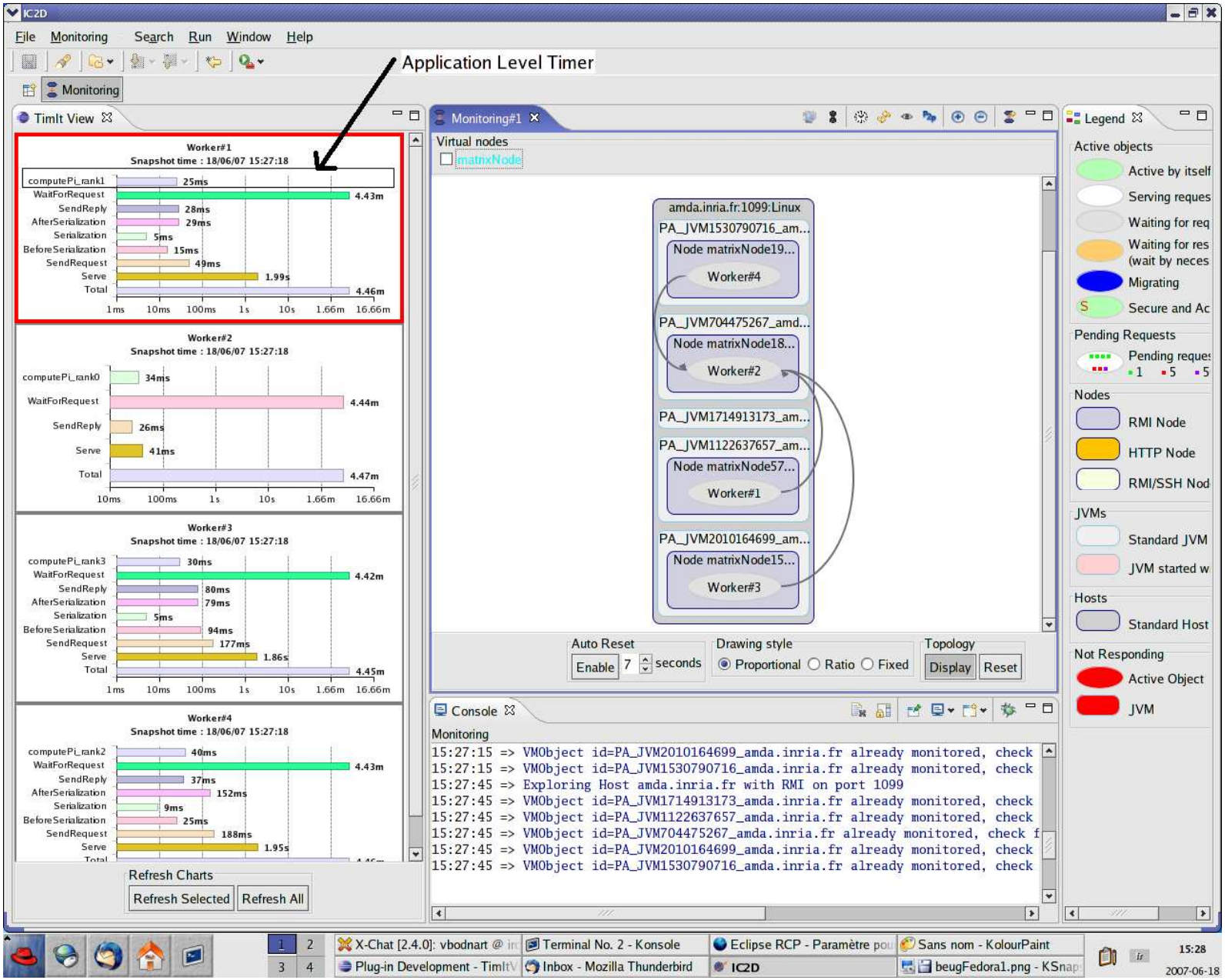

Monitoring view
Monitoring menu
Legend view
Monitoring - org.objectweb.proactive.ic2d.launcher - Eclipse SDK
File Edit Refactor Navigate Search Project Control Monitoring FieldAssist Run Window Help
Zoom
Monitoring
Java
Monitoring
Manual refresh
Virtual nodes
Virtual Nodes List
Management of the communications display
Display topology
Proportional
Ratio
Filaire
Reset Topology
Monitoring enable
Console
SVN
Console view
Legend
Active objects
Active by itself
Serving request
Waiting for request
Waiting for result
(wait by necessity)
Migrating
Pending Requests
Pending requests :
1
5
50
Nodes
RMI Node
HTTP Node
RMI/SSH Node
JINI Node
JVMs
Standard JVM
JVM started with Glo

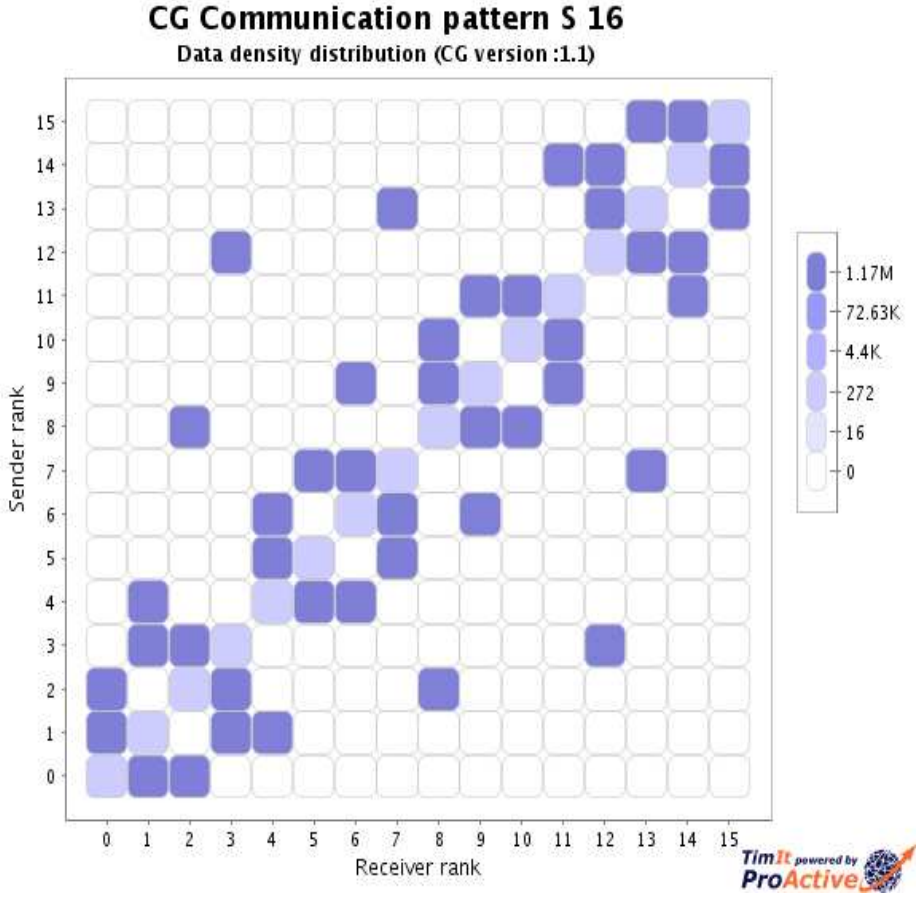


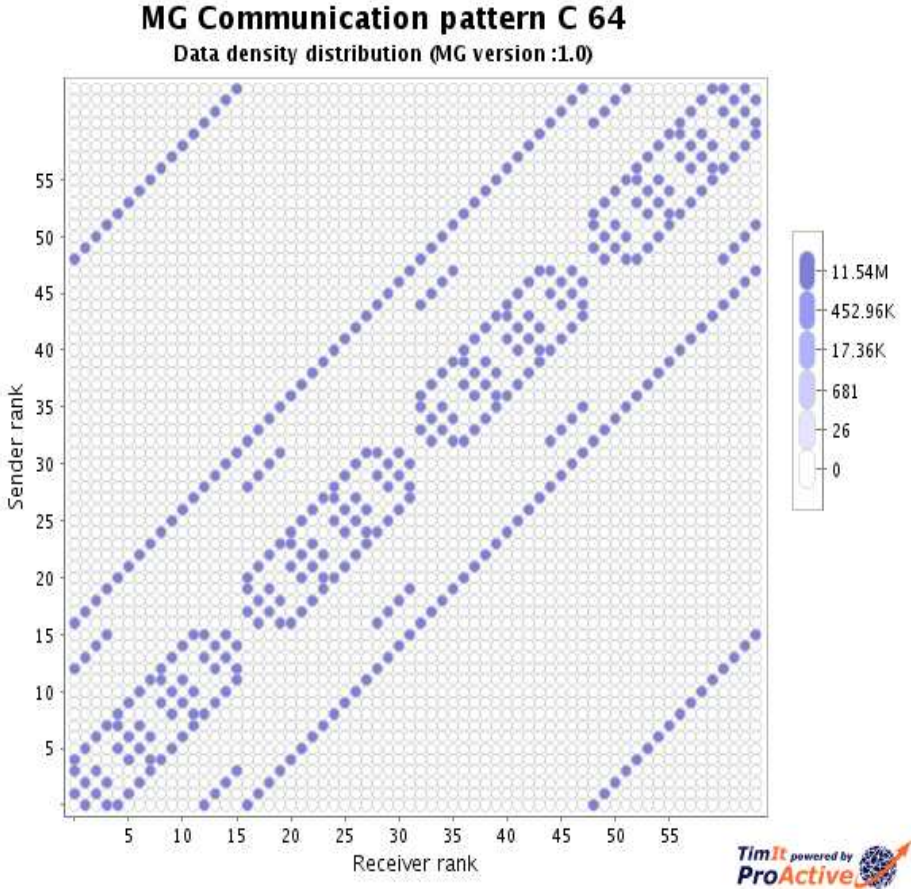
ProActive
Programming, Composing, Deploying on the Grid

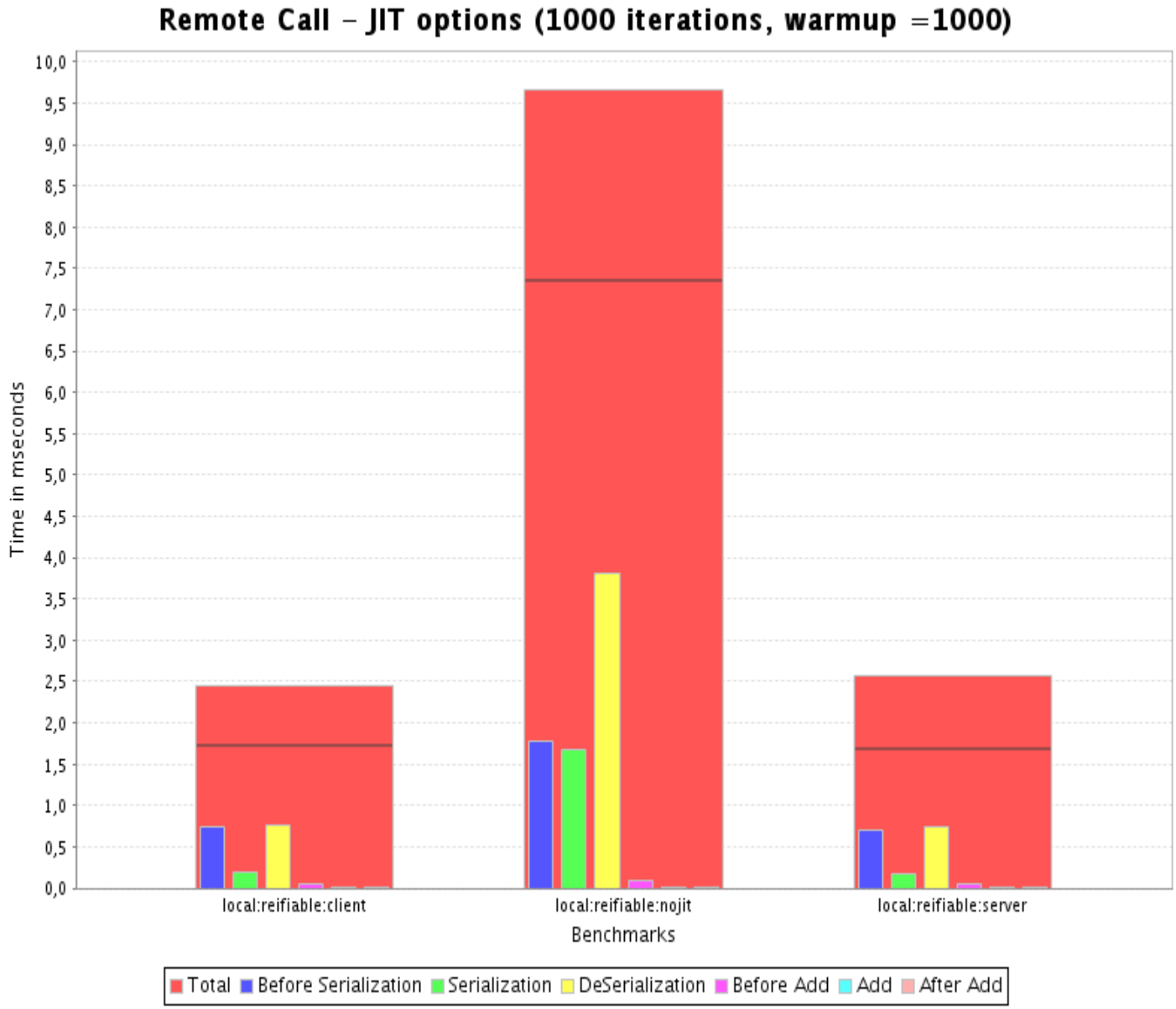

Launcher - TestDeployment.xml - Eclipse SDK
File Edit Navigate Search Project Run Window Help
Launcher
SVN Repo...
Plug-in De...
C3D.xml
TestDeployment.xml
Hello.xml
<?xml version="1.0" encoding="UTF-8"?>
<ProActiveDescriptor
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="DescriptorSchema.xsd">
<mainDefinition id="main" class="A">
<mapToVirtualNode value="lVNmain" />
</mainDefinition>
<componentDefinition>
<virtualNodesDefinition>
<virtualNode name="lVNmain" />
<virtualNode name="lVN1" />
<virtualNode name="lVN2" />
<virtualNode name="lVN3" />
</virtualNodesDefinition>
</componentDefinition>
<deployment>
<mapping>
<map virtualNode="lVNmain">
<jvmSet>
IC2D xml editor
Deployment Desc...
C3D.xml
Hello.xml - error
TestDeployment.xml - activated
Launch the application
Informations
Kill the application
Console
Launcher
13:51:03 => File selected : /user/jmlegait/home/workspace/org.objectweb.proacti
13:51:07 => File selected : /user/jmlegait/home/workspace/org.objectweb.proacti
13:54:05 => /user/jmlegait/home/workspace/org.objectweb.proactive.ic2d.launcher,
13:54:05 => Open the log4j console for more details
Launches the deployment descriptor
Gives informations
Kills the application
Launcher console

TimIt


IC2D
File Monitoring Search Run Window Help
Application Level Timer
Monitoring
TimIt View
Worker#1
Snapshot time : 18/06/07 15:27:18
computePi_rank1 25ms
WaitForRequest 4.43m
SendReply 28ms
AfterSerialization 29ms
Serialization 5ms
BeforeSerialization 15ms
SendRequest 49ms
Serve 1.99s
Total 4.46m
1ms 10ms 100ms 1s 10s 1.66m 16.66m
Worker#2
Snapshot time : 18/06/07 15:27:18
computePi_rank0 34ms
WaitForRequest 4.44m
SendReply 26ms
Serve 41ms
Total 4.47m
10ms 100ms 1s 10s 1.66m 16.66m
Worker#3
Snapshot time : 18/06/07 15:27:18
computePi_rank3 30ms
WaitForRequest 4.42m
SendReply 80ms
AfterSerialization 79ms
Serialization 5ms
BeforeSerialization 94ms
SendRequest 177ms
Serve 1.86s
Total 4.45m
1ms 10ms 100ms 1s 10s 1.66m 16.66m
Worker#4
Snapshot time : 18/06/07 15:27:18
computePi_rank2 40ms
WaitForRequest 4.43m
SendReply 37ms
AfterSerialization 152ms
Serialization 9ms
BeforeSerialization 25ms
SendRequest 188ms
Serve 1.95s
Refresh Charts
Refresh Selected
Refresh All
Monitoring#1
Virtual nodes
matrixNode
amda.inria.fr:1099:Linux
PA_JVM1530790716_am...
Node matrixNode19...
Worker#4
PA_JVM704475267_amd...
Node matrixNode18...
Worker#2
PA_JVM1714913173_am...
PA_JVM1122637657_am...
Node matrixNode57...
Worker#1
PA_JVM2010164699_am...
Node matrixNode15...
Worker#3
Auto Reset
Enable 7 seconds
Drawing style
Proportional Ratio Fixed
Topology
Display Reset
Console
Monitoring
15:27:15 => VMObject id=PA_JVM2010164699_amda.inria.fr already monitored, check
15:27:15 => VMObject id=PA_JVM1530790716_amda.inria.fr already monitored, check
15:27:45 => Exploring Host amda.inria.fr with RMI on port 1099
15:27:45 => VMObject id=PA_JVM1714913173_amda.inria.fr already monitored, check
15:27:45 => VMObject id=PA_JVM1122637657_amda.inria.fr already monitored, check
15:27:45 => VMObject id=PA_JVM704475267_amda.inria.fr already monitored, check f
15:27:45 => VMObject id=PA_JVM2010164699_amda.inria.fr already monitored, check
15:27:45 => VMObject id=PA_JVM1530790716_amda.inria.fr already monitored, check
Legend
Active objects
Active by itself
Serving reques
Waiting for req
Waiting for res (wait by neces
Migrating
Secure and Ac
Pending Requests
Pending reques 1 5 5
Nodes
RMI Node
HTTP Node
RMI/SSH Nod
JVMs
Standard JVM
JVM started w
Hosts
Standard Host
Not Responding
Active Object
JVM
X-Chat [2.4.0]: vbodnart @ in
Terminal No. 2 - Konsole
Eclipse RCP - Paramètre pou
Sans nom - KolourPaint
Plug-in Development - TimItV
Inbox - Mozilla Thunderbird
IC2D
beugFedora1.png - KSnap
15:28
2007-06-18

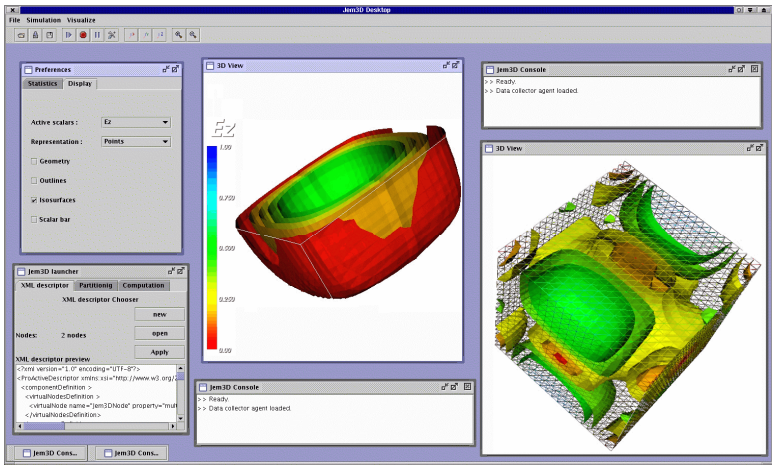

CG Communication pattern S 16
Data density distribution (CG version :1.1)
Sender rank
Receiver rank
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
1.17M
72.63K
4.4K
272
16
0
TimIt powered by ProActive

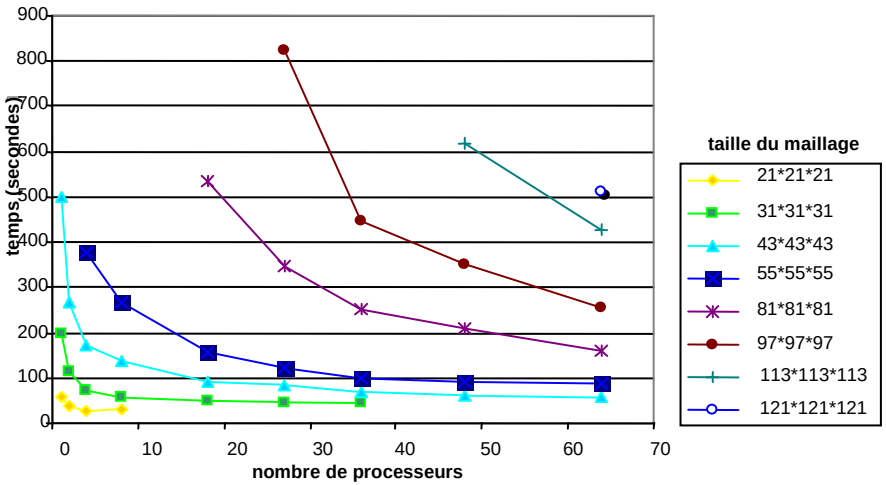

MG Communication pattern C 64
Data density distribution (MG version :1.0)
Sender rank
Receiver rank
5 10 15 20 25 30 35 40 45 50 55
11.54M
452.96K
17.36K
681
26
0
TimIt powered by ProActive


Remote Call – JIT options (1000 iterations, warmup =1000)
Time in mseconds
10,0
9,5
9,0
8,5
8,0
7,5
7,0
6,5
6,0
5,5
5,0
4,5
4,0
3,5
3,0
2,5
2,0
1,5
1,0
0,5
0,0
local:reifiable:client
local:reifiable:nojit
local:reifiable:server
Benchmarks
Total
Before Serialization
Serialization
DeSerialization
Before Add
Add
After Add

Some Applications

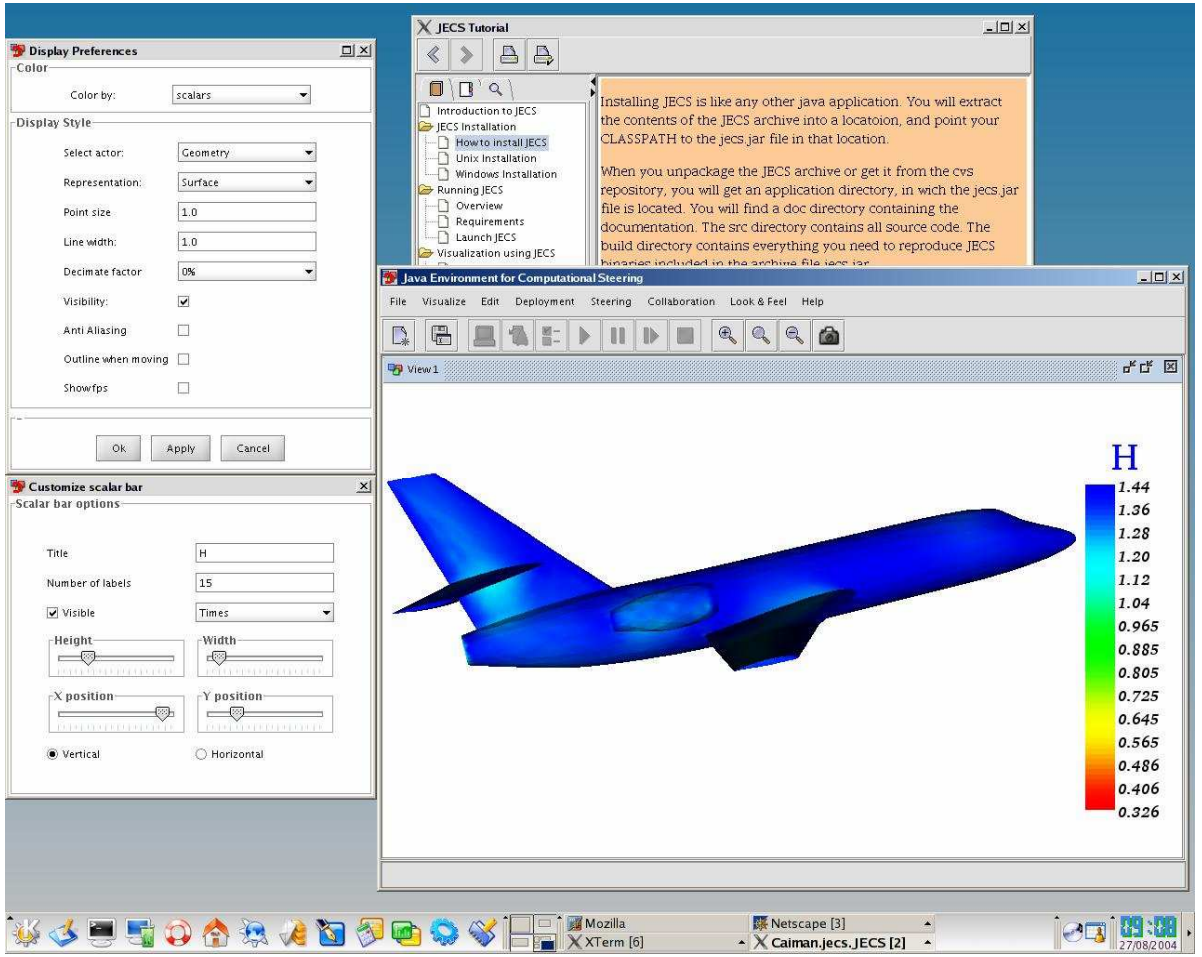
Java 3D Electromagnetism



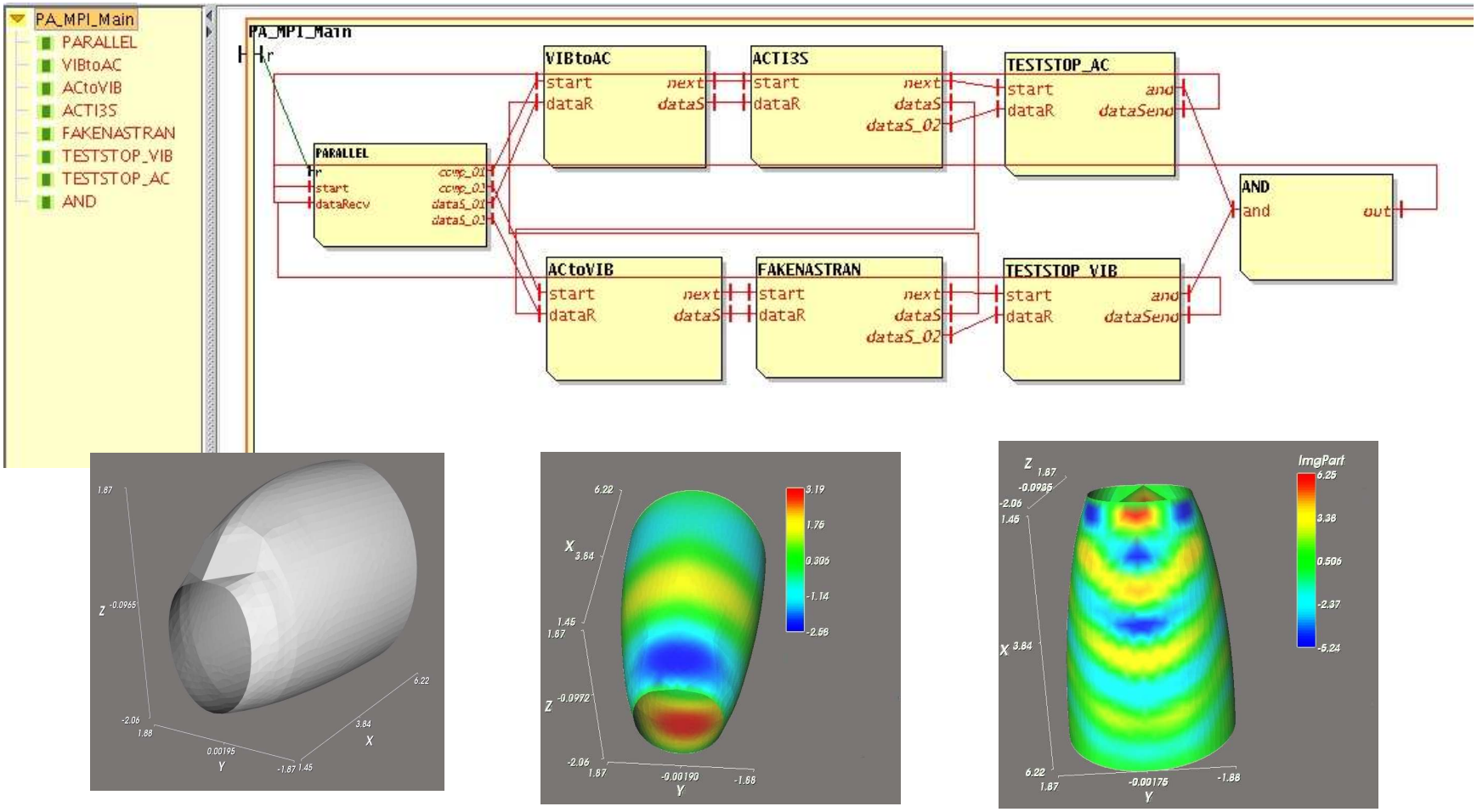


- Maxwell 3D equation solver, Finite Volume Method (FVM)
- Pre-existing **Fortran MPI** version: **EM3D** (CAIMAN team @ INRIA)
- 300+ machines at the same time (Intranet and cluster)
- Large data sets: 150x150x150 (100 million facets)

JECS : A Generic Version of Jem3D

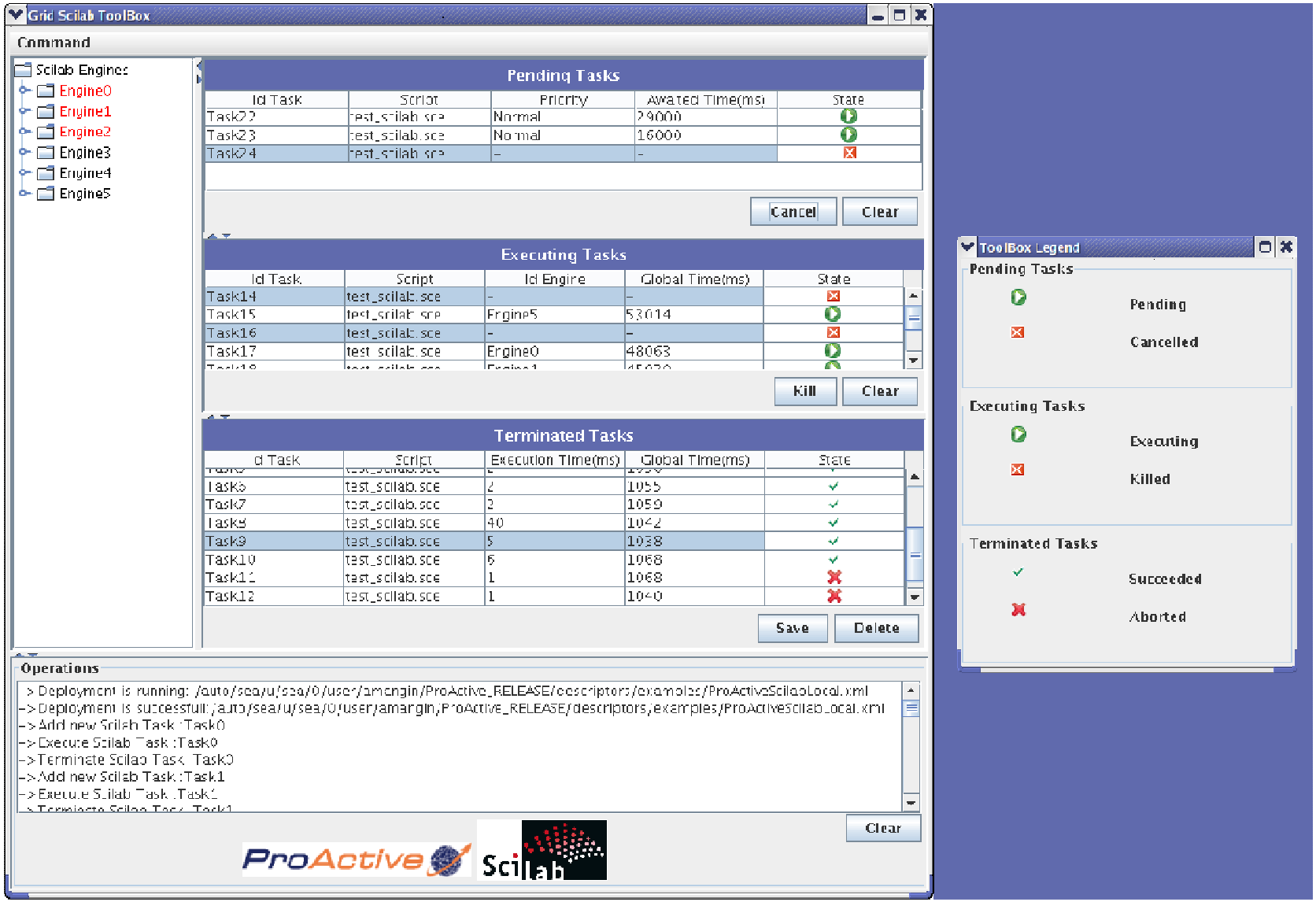


Code Coupling : Vibro Acoustic (courtesy of EADS)

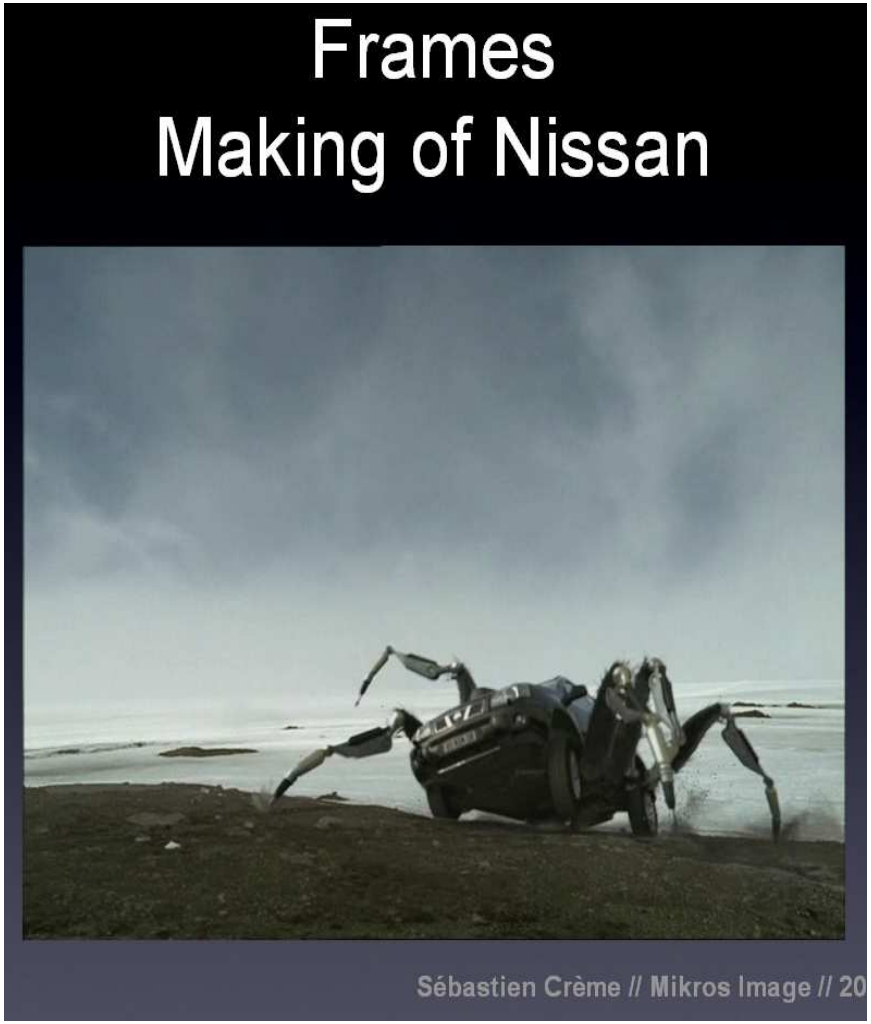


ProActive
Programming, Composing, Deploying on the Grid

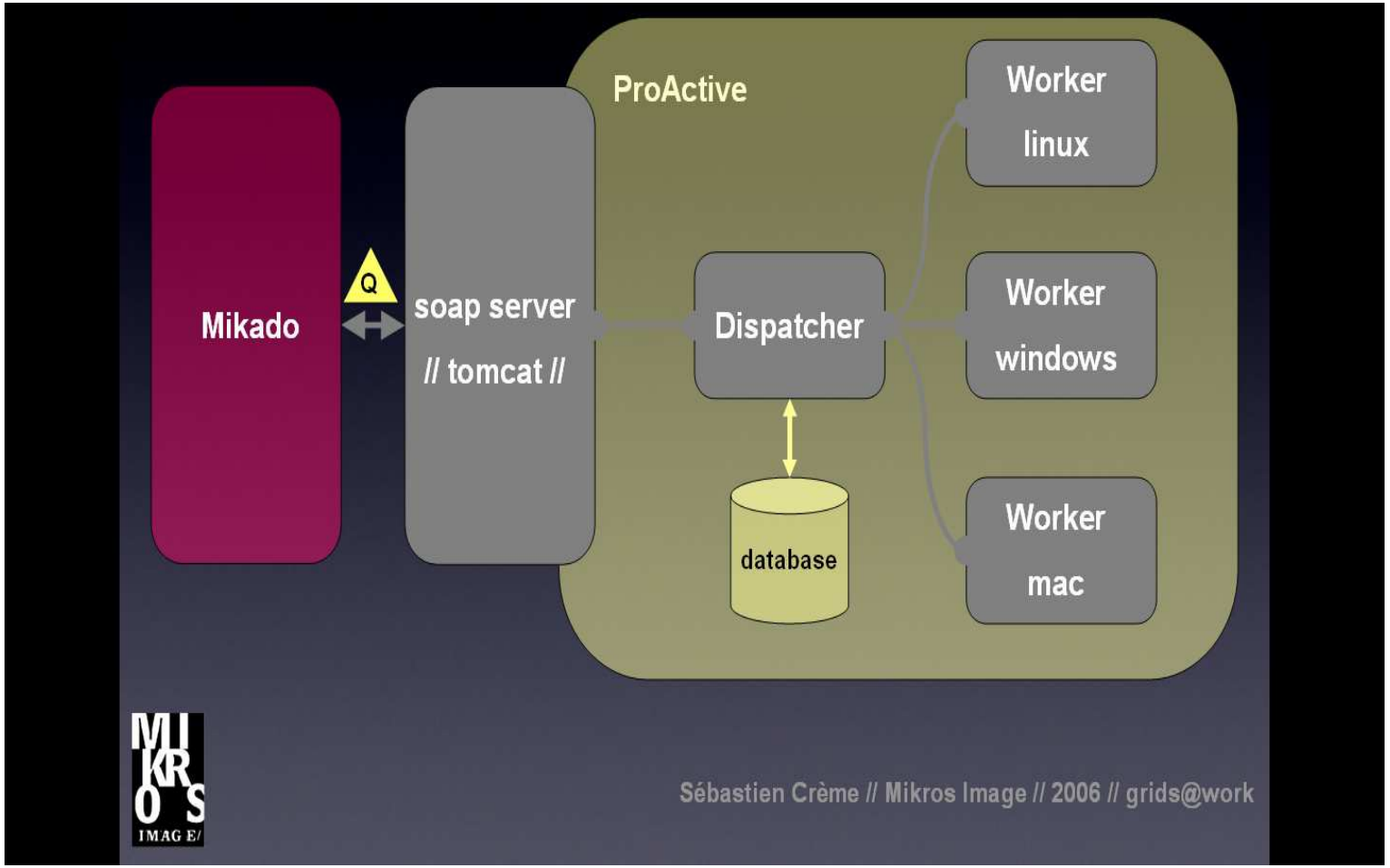
Scilab Grid Toolkit



Post Production movie processing



Post Production movie processing

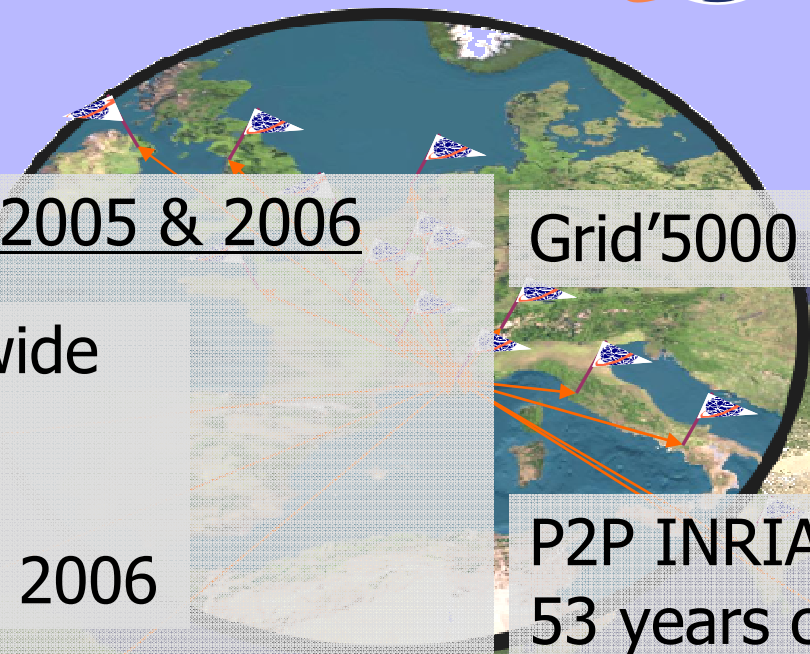


Large Scale Deployments

ObjectWeb Open Source Middleware

ProActive
Programming, Composing, Deploying on the Grid

PLUGTESTS THE INTEROPERABILITY SERVICE

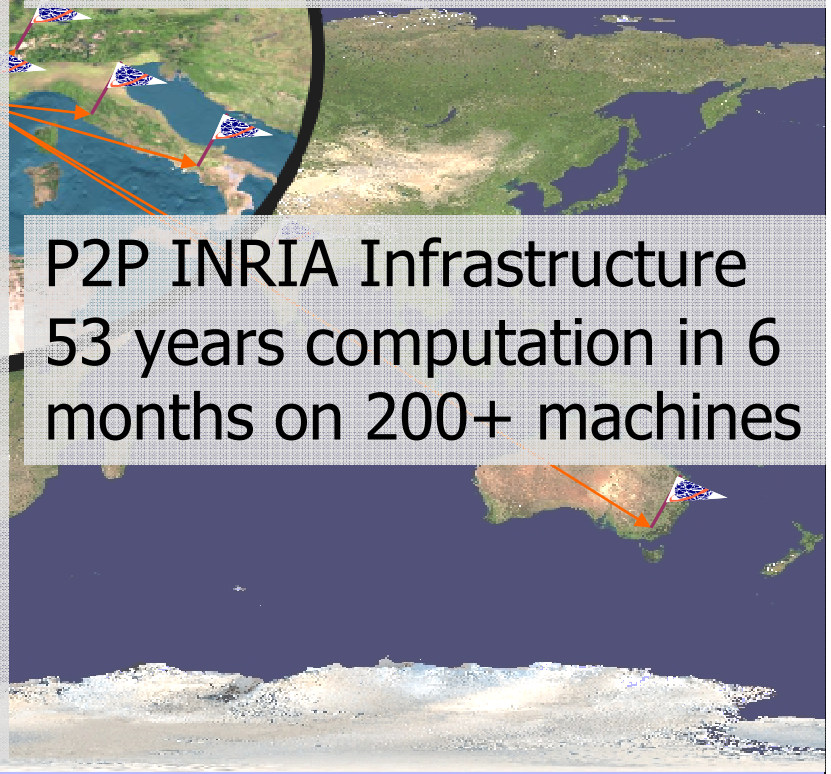


Grid Plugtests 2004, 2005 & 2006

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, Service

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…

http://proactive.objectweb.org

Let’s practice !