



Short Introduction to Parallelism, Grid and ProActive, Distributed Objects and Components (GCM)

Denis Caromel, et al.

<http://ProActive.ObjectWeb.org>

OASIS Team

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



ProActive

a complete solution for Enterprise GRID

Programming

Wrapping

Composing

Deploying

Including:

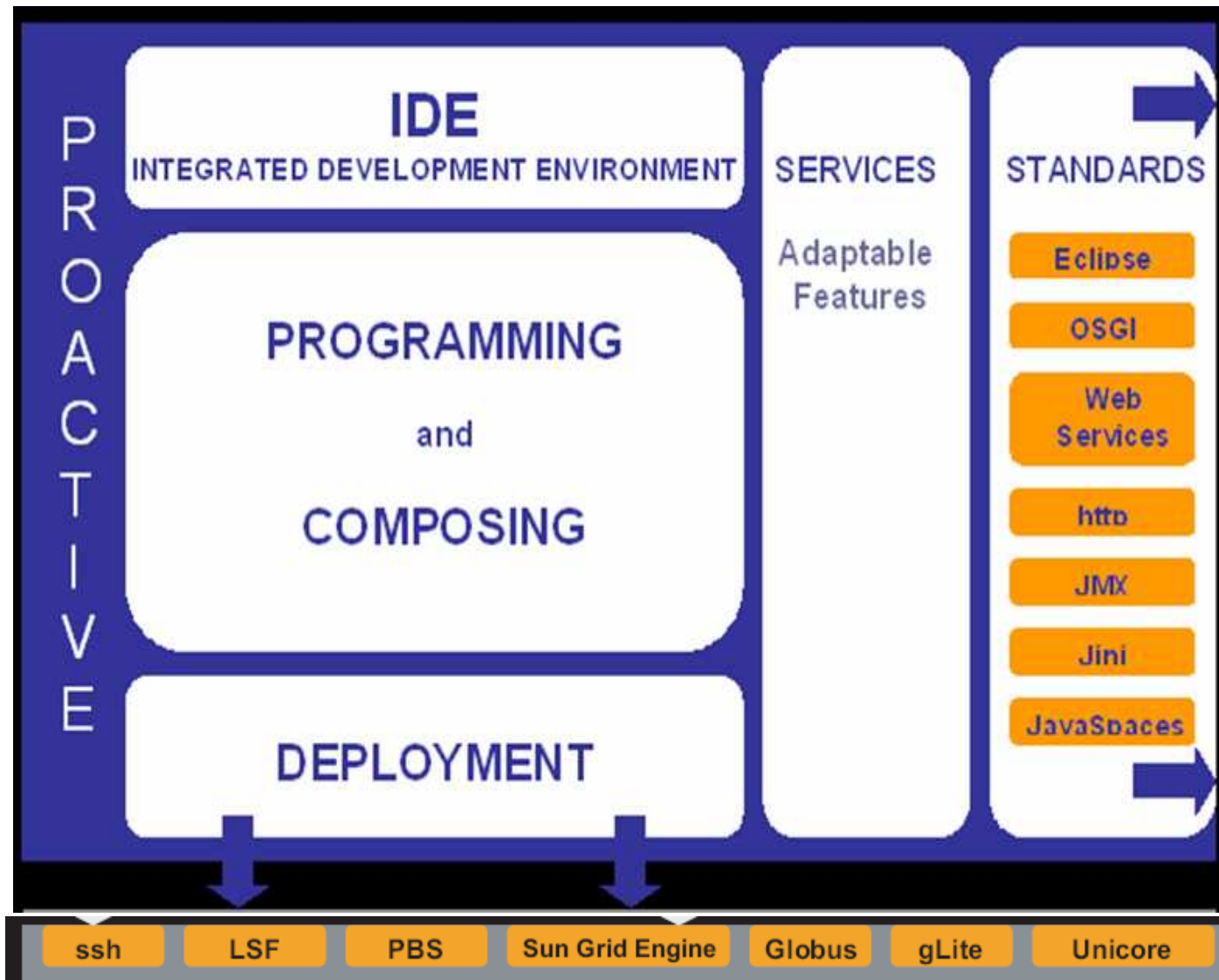
P2P, File Transfer, Branch & Bound,
Load Balancing, Fault Tolerance



Overview



ProActive's Framework in a nutshell



Open
Source
+
PROFESSIONAL
SUPPORT

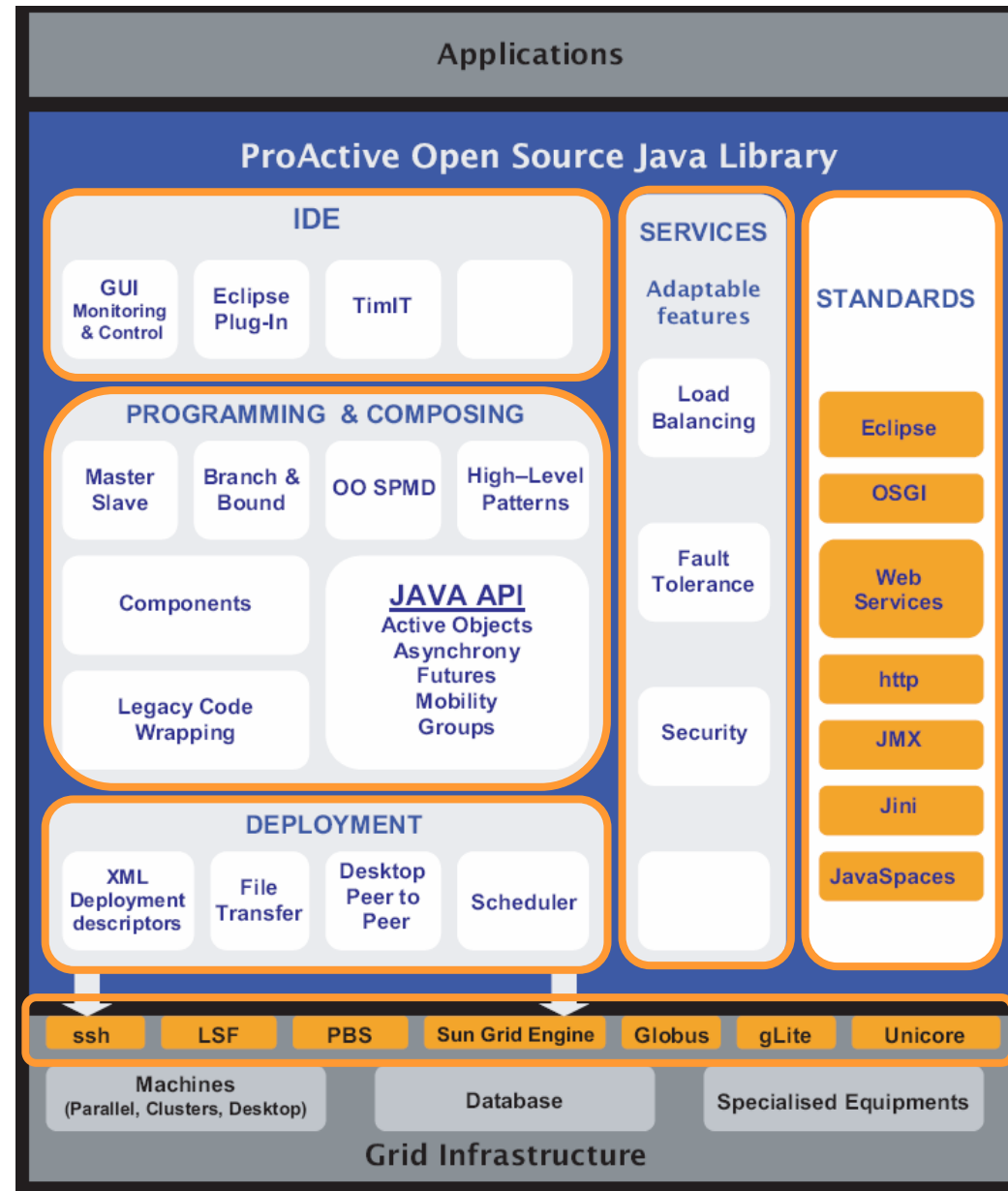


Inside *ProActive*

IDE

PROGRAMMING
&
COMPOSING

DEPLOYMENT

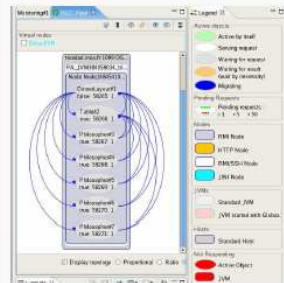


ProActive GUIs

IC2D Eclipse, **TimIt**, ...

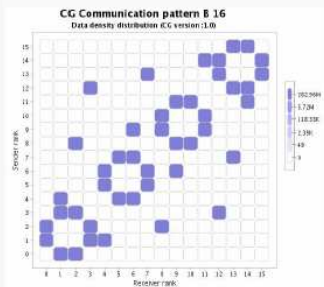
Graph of Distributed References

Display the references between active objects; in short it can easily spot cyclic dependencies.



TimIt Pattern

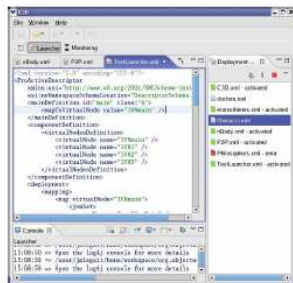
Visualize both number and size of communications between distributed entities.



Launcher and Scheduler

Launching applications directly from the GUI.

Allow seamless deployment on local machines (rsh, ssh...), as well as on third-party schedulers (such as PlatformTM LSF, PBS, Globus, Sun GE, gLite, Unicore...). The toolbox also provides a simple generic scheduler interface that lets you deploy your own scheduler, or integrate directly within your application.



GRID Scilab

ProActive interfaces with the Scilab Scientific Free Software Package

It allows the deployment of Scilab instances on distributed nodes, and manages scripts to be executed in parallel.



For more information on ProActive and Professional Services, please visit <http://ProActive.ObjectWeb.org/>
Contact Denis Caramel, Head of OASIS Team
Phone: +33 492 387 631; Facsimile: +33 492 387 971
Denis.Caramel@inria.fr
Joint Project University of Nice Sophia Antipolis - INRIA - CNRS - I3S
ProActive is a member of the ObjectWeb Open Source Consortium

ProActive
Programming, Composing, Deploying on the Grid
INRIA - INSTITUT NATIONAL DE RECHERCHE
EN INFORMATIQUE ET EN AUTOMATIQUE
2004, Route des Lucioles, BP 93
06902 Sophia Antipolis Cedex, France

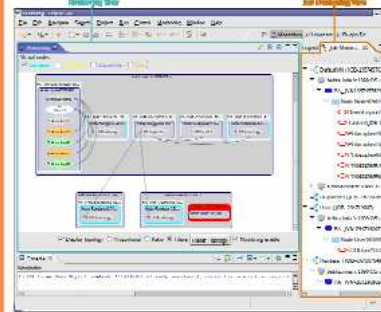
GRID IDE: Integrated Development Environment

ProActive
Programming, Composing, Deploying on the Grid

IC2D: View, Manage, Control
Distribution and Parallelism

Grid IDE Overview

ProActive features an integrated full-fledged Grid IDE (Integrated Development Environment) named IC2D. It features graphical monitoring and control, programming wizards, debugging and optimizing tools which all contribute to **high-productivity grid development**. Moreover, enterprise developers can work in a familiar setting within **Eclipse plugins**; increasing both productivity and reducing the need for new training.

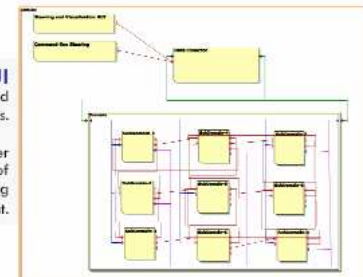


Graphical Monitoring View

A graphical environment for remote monitoring and steering of distributed and grid applications.

It provides a graphical visualisation for hosts, Java Virtual Machines, and active objects, including the topology and the volume of communications.

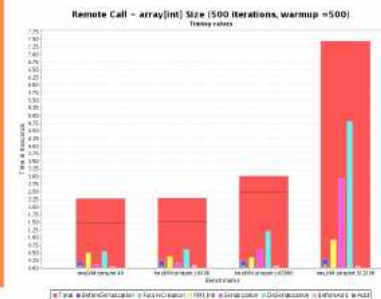
Job monitoring (right side) provides a tree representation of all these entities.



Fractal GUI

A graphical tool to edit Fractal and Grid Component configurations.

The GUI allows for Client/Server interface connections, and renaming of Virtual Nodes for controlling mapping and co-allocation at deployment.



TimIt Bench

A complete solution to benchmark and optimize applications' performance.

TimIt is able to produce a large variety of statistics, advanced timers with hierarchical capabilities.

TimIt automatically generates statistical charts.



Interface with The Scientific Free Software SciLab

The screenshot displays the 'Grid Scilab ToolBox' window, which manages tasks across multiple SciLab engines. The interface is divided into several sections:

- Command:** A tree view on the left lists 'Scilab Engines' from Engine0 to Engine5.
- Pending Tasks:** A table showing tasks waiting to be executed.
- Executing Tasks:** A table showing tasks currently running on specific engines.
- Terminated Tasks:** A table showing tasks that have completed or failed.
- Operations:** A log at the bottom showing the sequence of actions performed.
- ToolBox Legend:** A separate window on the right that defines the status icons used in the task tables.

Pending Tasks Table:

Id Task	Script	Priority	Awaited Time(ms)	State
Task22	test_scilab.sce	Normal	29000	Pending (green circle)
Task23	test_scilab.sce	Normal	16000	Pending (green circle)
Task24	test_scilab.sce	-	-	Cancelled (red X)

Executing Tasks Table:

Id Task	Script	Id Engine	Global Time(ms)	State
Task14	test_scilab.sce	-	-	Cancelled (red X)
Task15	test_scilab.sce	Engine5	53014	Pending (green circle)
Task16	test_scilab.sce	-	-	Cancelled (red X)
Task17	test_scilab.sce	Engine0	48063	Pending (green circle)
Task18	test_scilab.sce	Engine1	45020	Pending (green circle)

Terminated Tasks Table:

Id Task	Script	Execution Time(ms)	Global Time(ms)	State
Task5	test_scilab.sce	2	1059	Succeeded (green check)
Task6	test_scilab.sce	2	1055	Succeeded (green check)
Task7	test_scilab.sce	2	1059	Succeeded (green check)
Task8	test_scilab.sce	40	1042	Succeeded (green check)
Task9	test_scilab.sce	5	1038	Succeeded (green check)
Task10	test_scilab.sce	6	1068	Succeeded (green check)
Task11	test_scilab.sce	1	1068	Aborted (red X)
Task12	test_scilab.sce	1	1040	Aborted (red X)

ToolBox Legend:

- Pending Tasks:**
 - Green circle: Pending
 - Red X: Cancelled
- Executing Tasks:**
 - Green circle: Executing
 - Red X: Killed
- Terminated Tasks:**
 - Green check: Succeeded
 - Red X: Aborted

Operations Log:

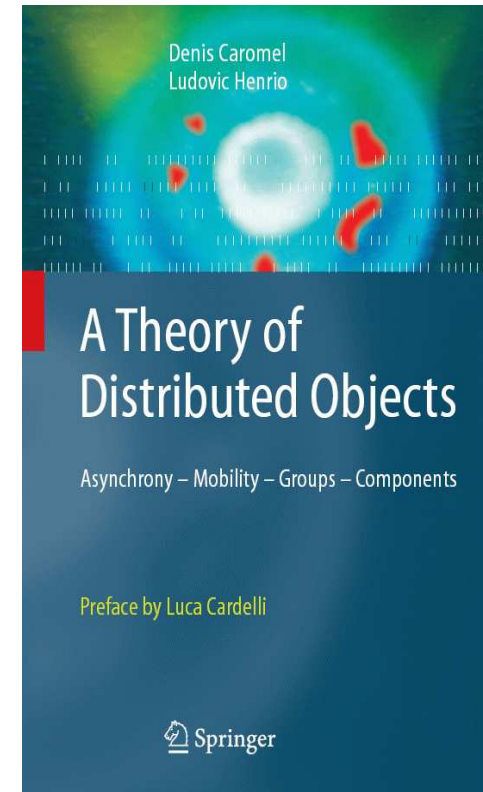
- > Deployment is running: /auto/sea/u/sea/0/user/amangin/ProActive_RELEASE/descriptors/examples/ProActiveScilabLocal.xml
- > Deployment is successful: /auto/sea/u/sea/0/user/amangin/ProActive_RELEASE/descriptors/examples/ProActiveScilabLocal.xml
- > Add new Scilab Task :Task0
- > Execute Scilab Task :Task0
- > Terminate Scilab Task :Task0
- > Add new Scilab Task :Task1
- > Execute Scilab Task :Task1
- > Terminate Scilab Task :Task1

The interface also features a 'ProActive' logo and a 'Scilab' logo at the bottom, along with a 'Clear' button for the operations log.

THEORY

A Theory of Distributed Objects

D. Caromel, L. Henrio,
Springer 2005, Monograph

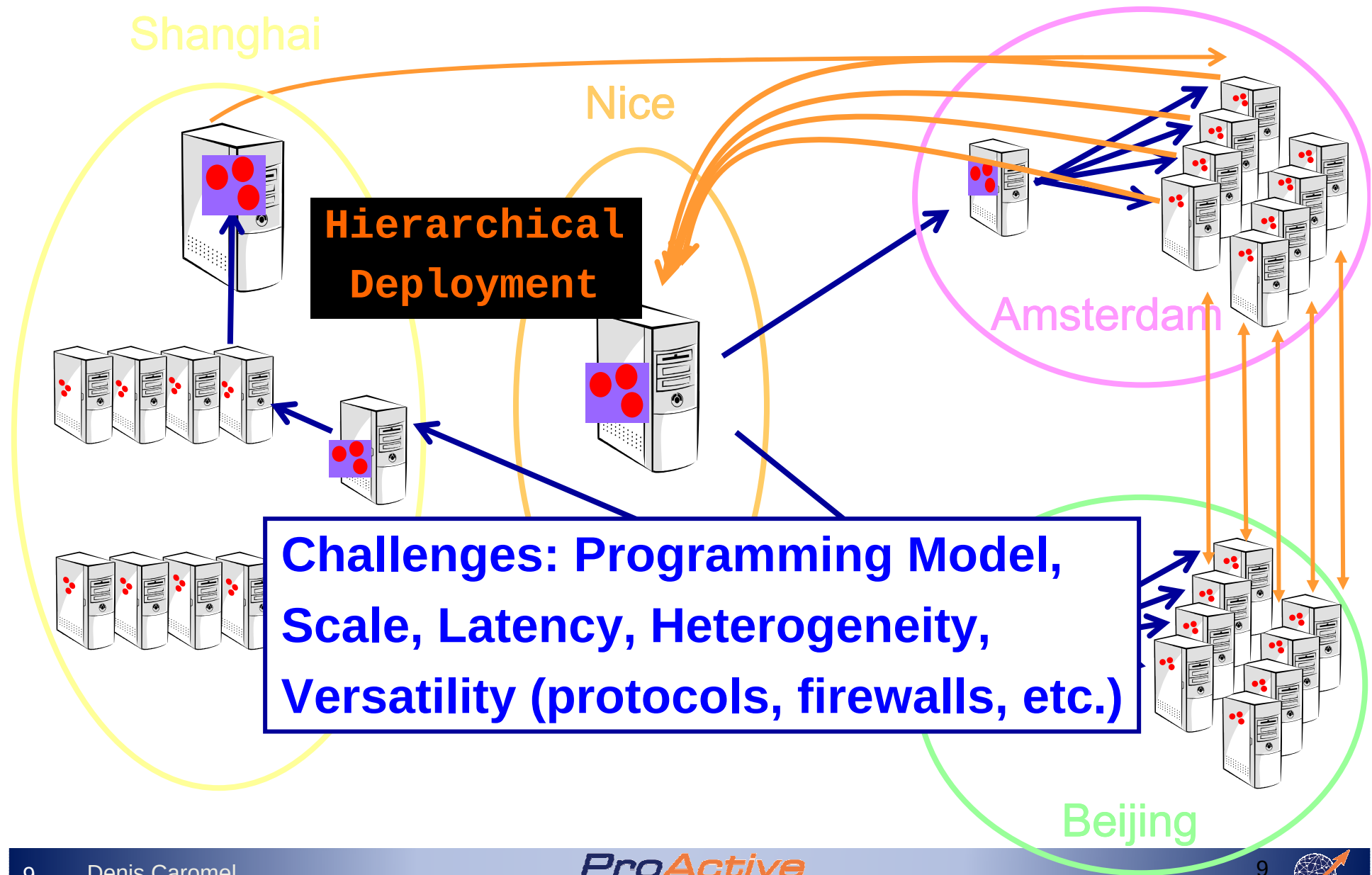


A Calculus:

ASP: Asynchronous Sequential Processes

- Based on Sigma-Calculus (Abadi-Cardelli)
- Formal Proofs of determinism
- Releases a few important implementation constraints

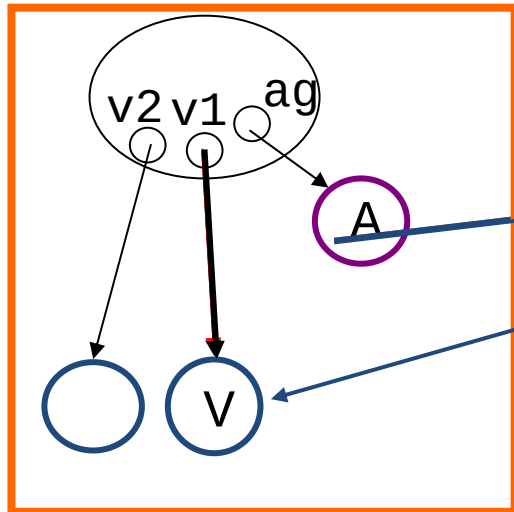
Grid Computing with *ProActive*



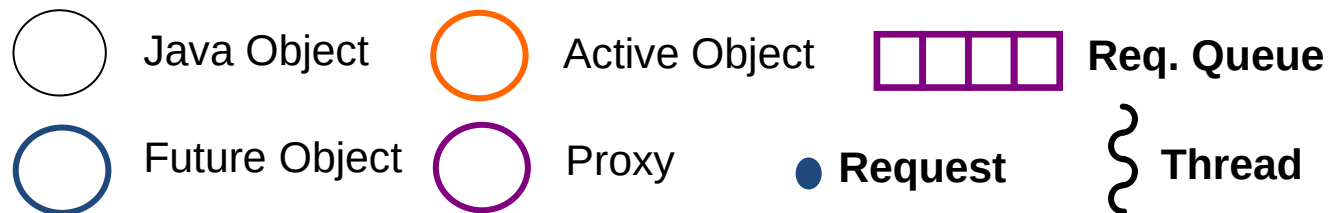
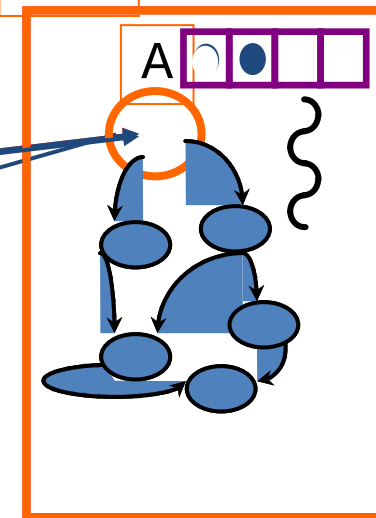
ProActive : Active objects

- A ag = **newActive** ("A", [...], VirtualNode)
- V v1 = ag.foo (param);
- V v2 = ag.bar (param);
- ...
- v1.bar(); **//Wait-By-Necessity**

JVM



JVM

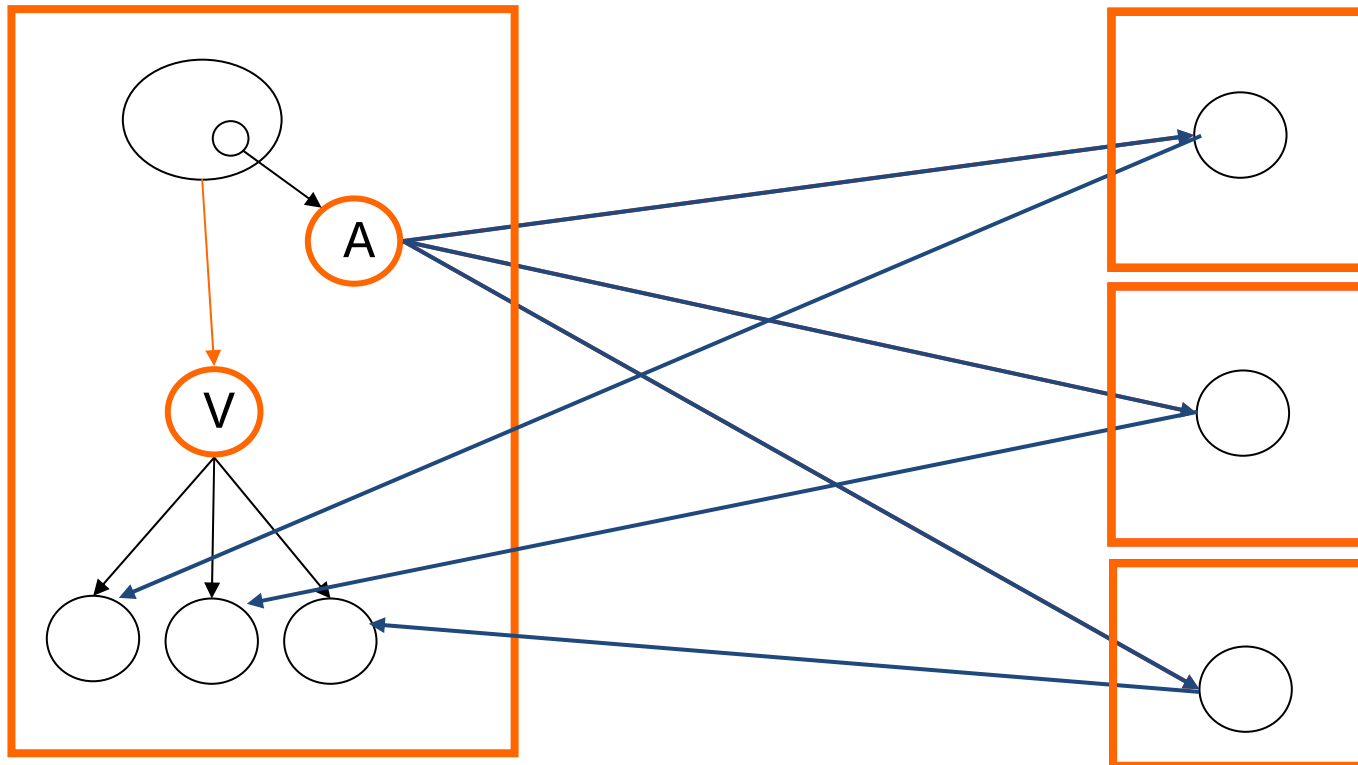


Wait-By-Necessity
is a
Dataflow
Synchronization



Creating AO and Groups

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



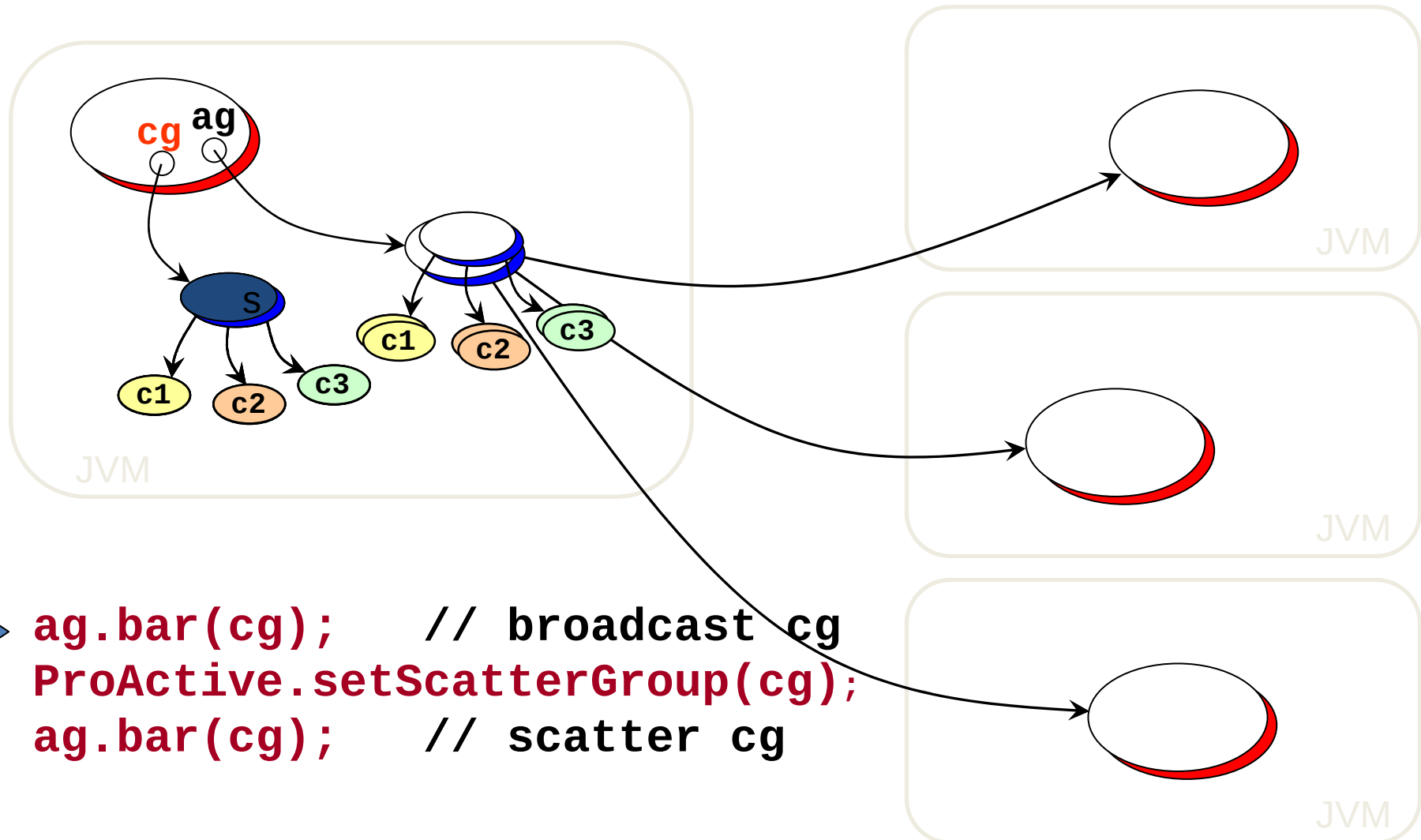
○ Typed Group ○ Java or Active Object

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**

4.2 IC2D

Interactive Control & Debug for Distribution

Eclipse

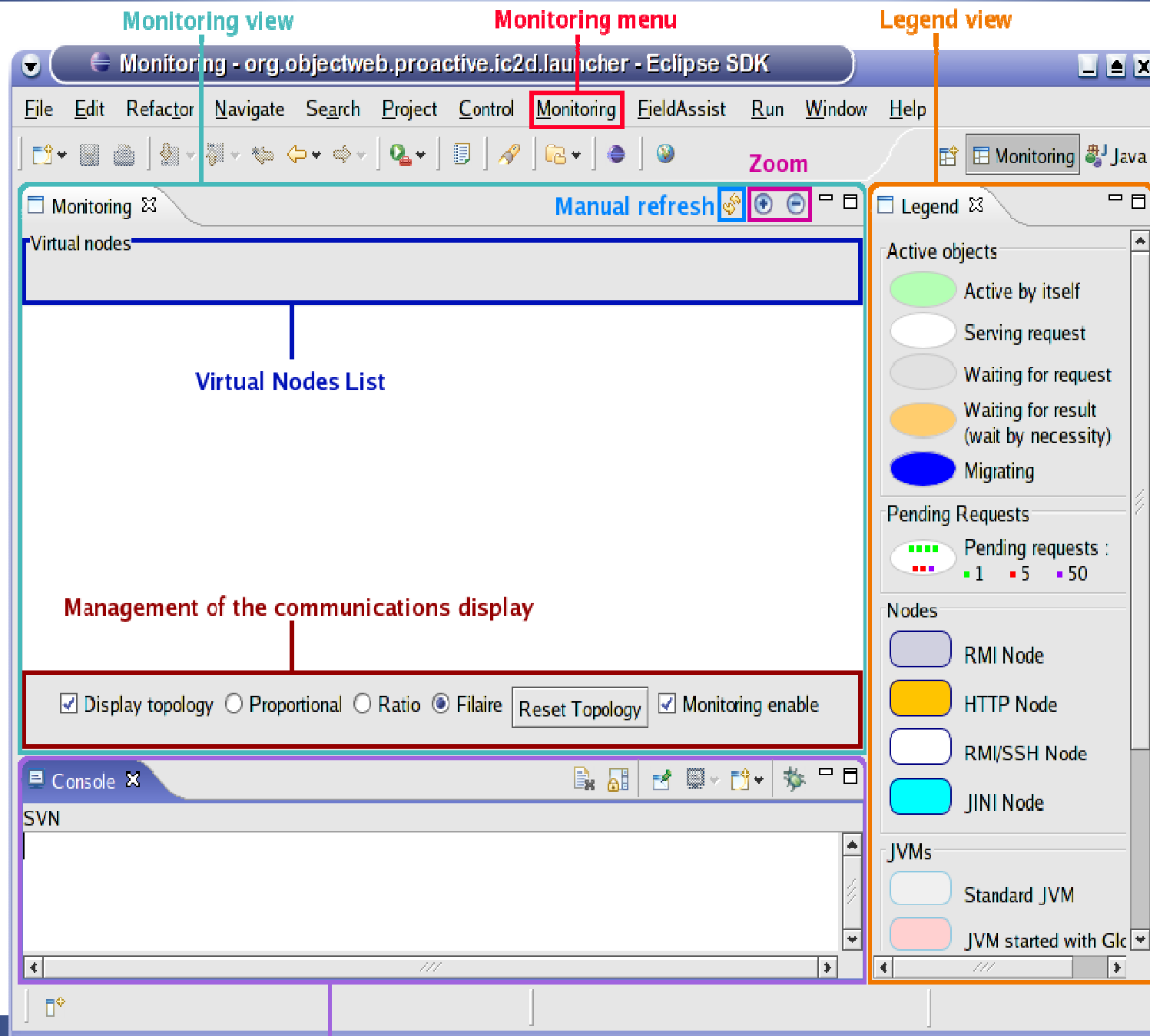
GUI for the GRID



ProActive

Eclipse UI





Monitoring View

Job Monitoring View

The screenshot displays the Eclipse IDE's Monitoring and Job Monitoring views. The top toolbar includes menus for File, Edit, Navigate, Search, Project, Run, Control, Monitoring, Window, and Help. The main workspace is divided into two panes.

Monitoring View (Left Pane):

- Virtual nodes:** A list of nodes with checkboxes: ☒ **Renderer**, ☐ **DefaultVN**, ☐ **Dispatcher**, and ☐ **User**.
- Diagram:** A complex graph showing the relationships between various components. On the left, a group of nodes includes **DinnerLayout#2**, **Table#3**, **Philosopher#4**, **Philosopher#5**, **Philosopher#6**, **Philosopher#7**, and **Philosopher#8**. These are connected to a central set of nodes: **Node Renderer-127...** (labeled **C3D Rendering...**), **Node Dispatcher-5...** (labeled **C3D Dispatche...**), **Node User1602644...** (labeled **C3DUser#13**), and **Node Renderer1307...** (labeled **C3DRendering...**). Below this, there are three more nodes: **Node Renderer1174...** (labeled **C3DRendering...**), **Node Renderer151...** (labeled **C3DRendering...**), and **Node Node-4551863...** (labeled **C3DRendering...**).
- Legend:** A list of nodes with checkboxes: ☒ **Renderer**, ☐ **DefaultVN**, ☐ **Dispatcher**, and ☐ **User**.
- Buttons:** ☒ **Display topology**, ☐ **Proportional**, ☐ **Ratio**, ☒ **Filaire**, **Reset Topology**, and ☒ **Monitoring enable**.

Job Monitoring View (Right Pane):

- Legend:** A list of nodes with checkboxes: ☒ **Renderer**, ☐ **DefaultVN**, ☐ **Dispatcher**, and ☐ **User**.
- Job List:** A hierarchical list of jobs and nodes:
 - DefaultVN (JOB-1357457629)**
 - bebita.inria.fr:1099:OS un...**
 - PA_JVM1357457629_...**
 - Node Node6056249...**
 - DinnerLayout#2**
 - Table#3 (JOB-1357457629)**
 - Philosopher#4 (JOB-1357457629)**
 - Philosopher#5 (JOB-1357457629)**
 - Philosopher#6 (JOB-1357457629)**
 - Philosopher#7 (JOB-1357457629)**
 - Philosopher#8 (JOB-1357457629)**
 - sidonie.inria.fr:1099:OS un...**
 - Dispatcher (JOB-1672076495)**
 - User (JOB-294719007)**
 - bebita.inria.fr:1099:OS un...**
 - PA_JVM-294719007_...**
 - Node User1602644...**
 - C3DUser#13 (JOB-294719007)**
 - Renderer (JOB-1672076495)**
 - bebita.inria.fr:1099:OS un...**
 - PA_JVM-1631909824_...**

Console (Bottom):

```
Monitoring
15:09:15 => NodeObject id=Node-455186381 already monitored, check for new active objects
```



Launcher - TestDeployment.xml - Eclipse SDK

File Edit Navigate Search Project Run Window Help

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

Information

Kill the application

Launcher console

13:51:03 => File selected : /user/jnlegait/home/workspace/org.objectweb.proactive

13:51:07 => File selected : /user/jnlegait/home/workspace/org.objectweb.proactive

13:54:05 => /user/jnlegait/home/workspace/org.objectweb.proactive.ic2d.launcher,

13:54:05 => Open the log4j console for more details

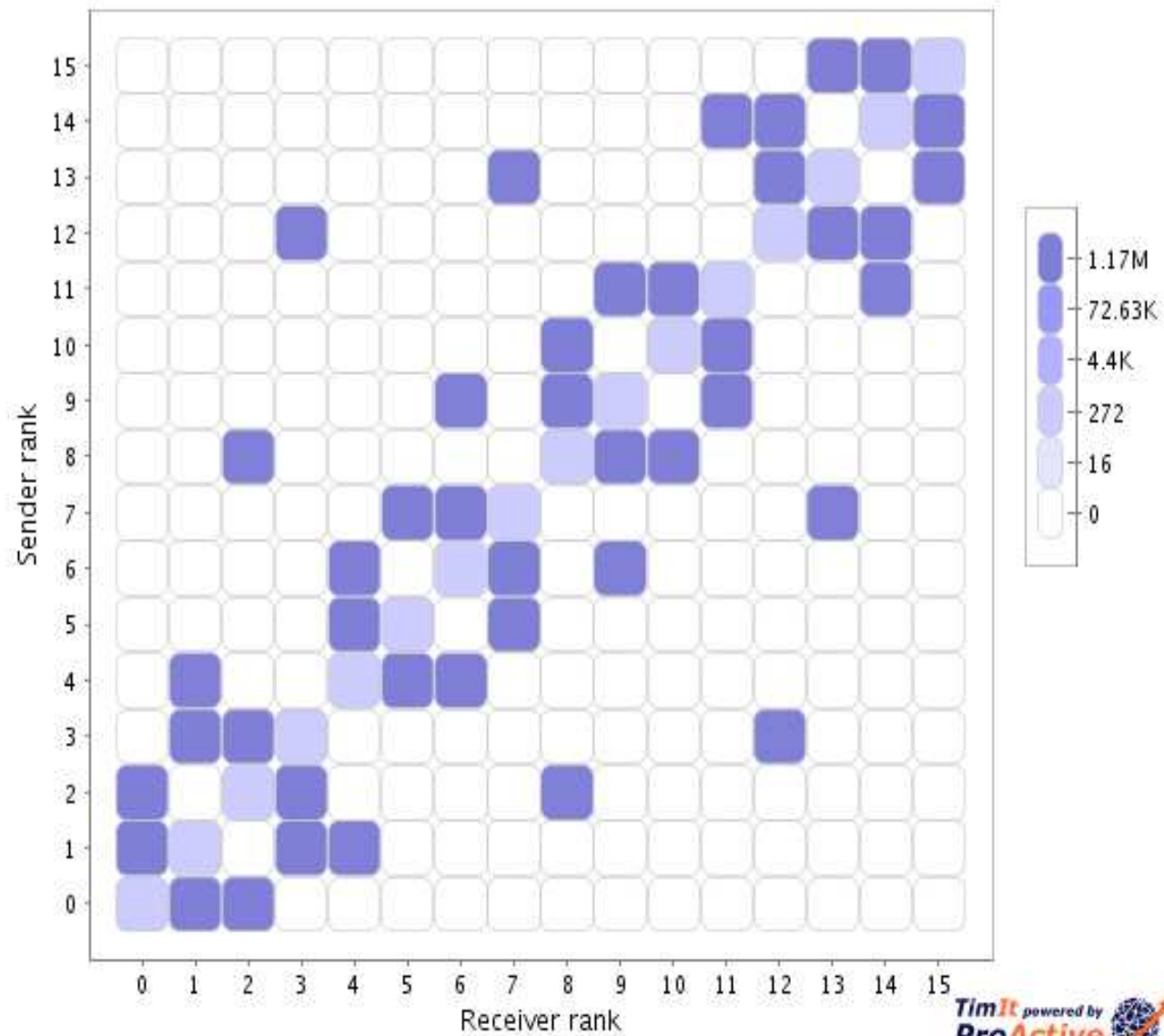


TimIt



CG Communication pattern S 16

Data density distribution (CG version :1.1)

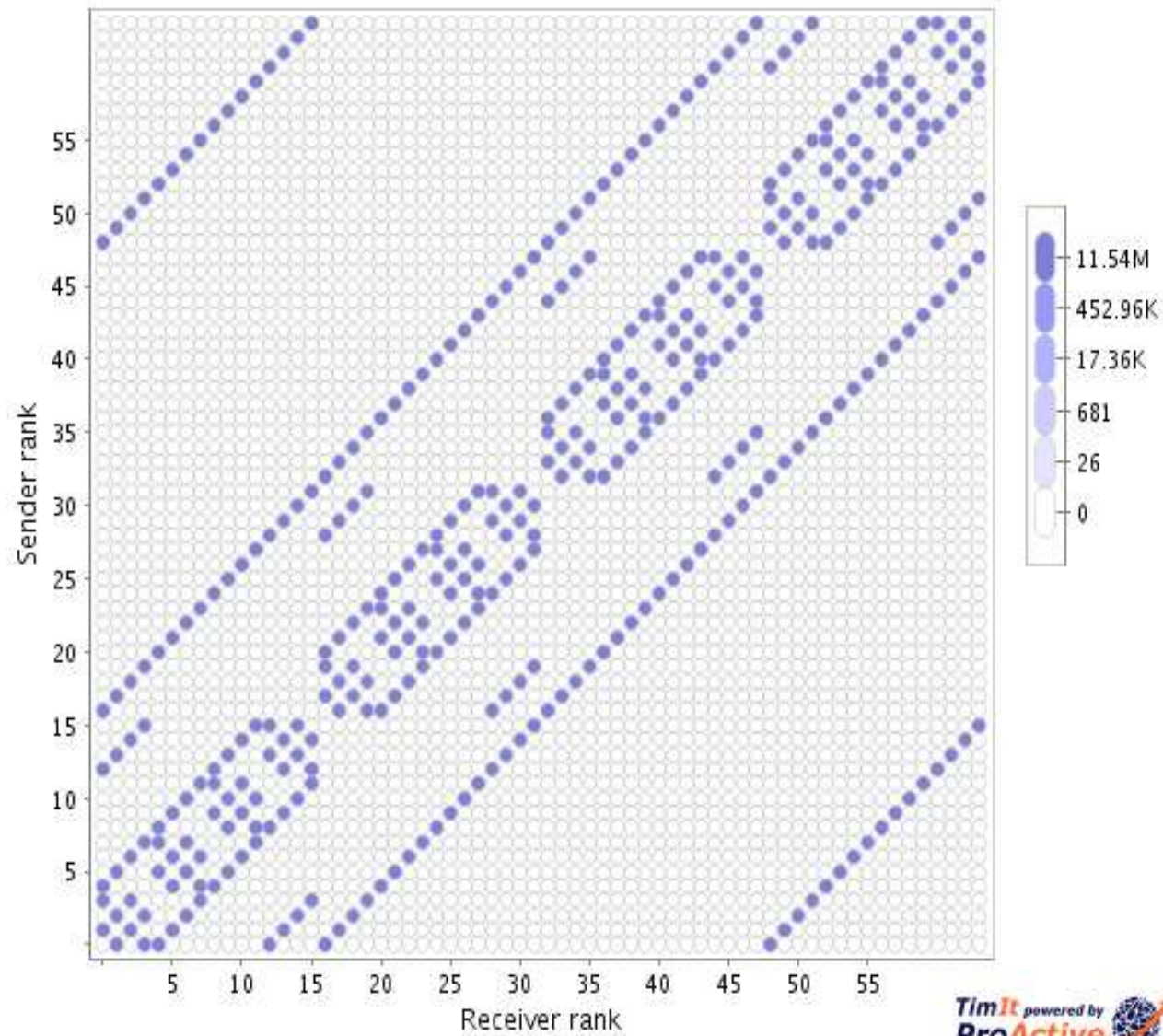


TimIt powered by
ProActive



MG Communication pattern C 64

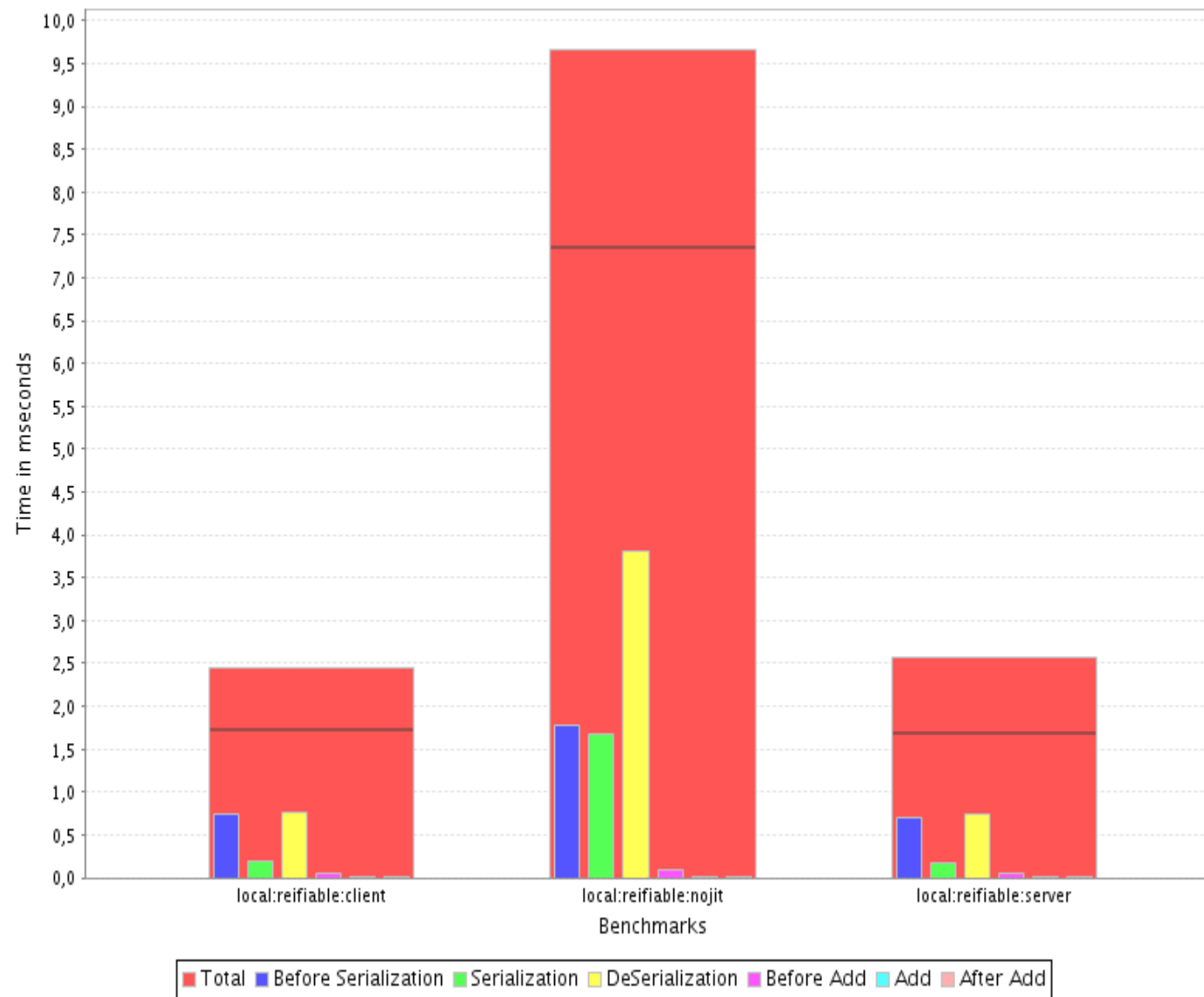
Data density distribution (MG version :1.0)



TimIt powered by
ProActive

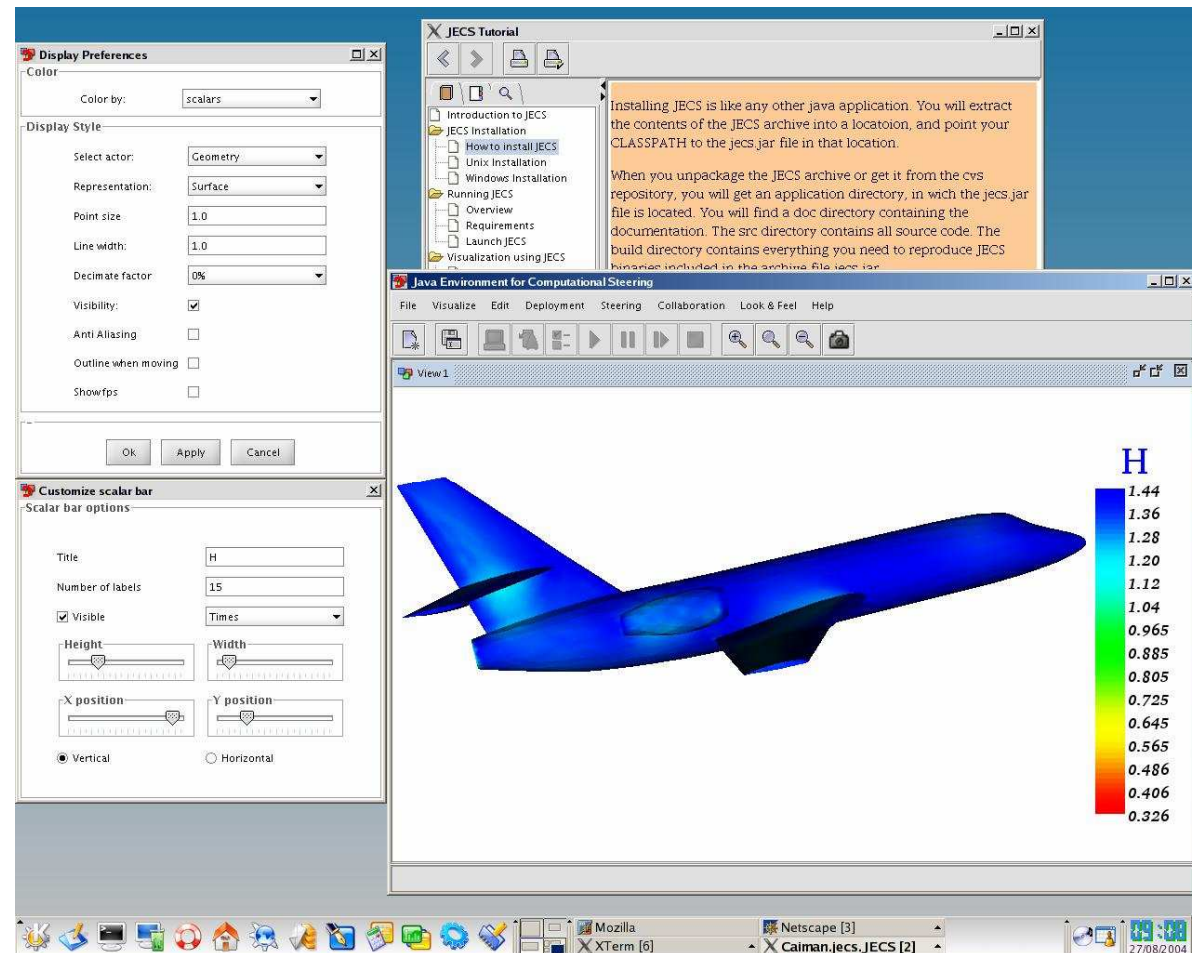


Remote Call – JIT options (1000 iterations, warmup =1000)

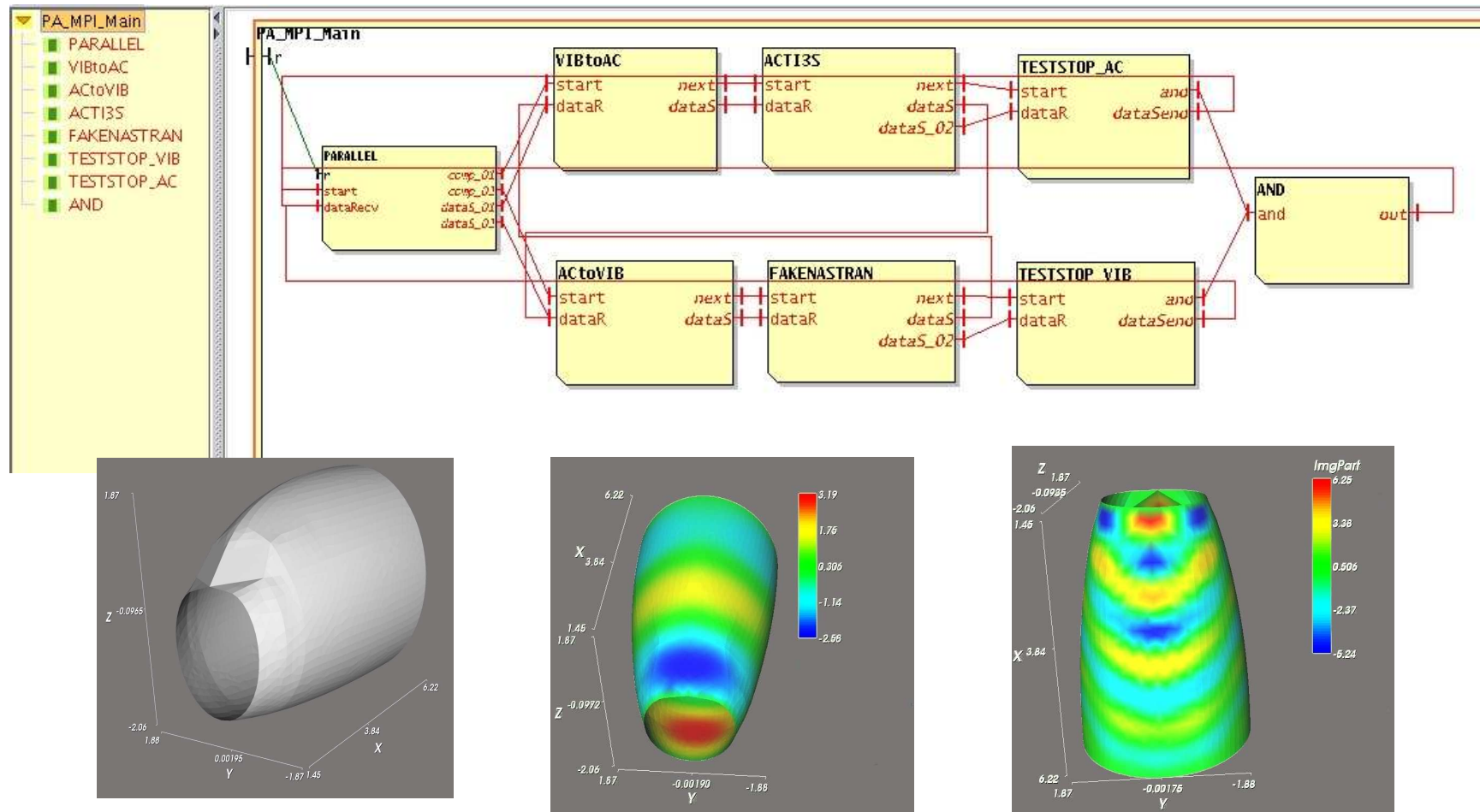


JECS : 3D Electromagnetism

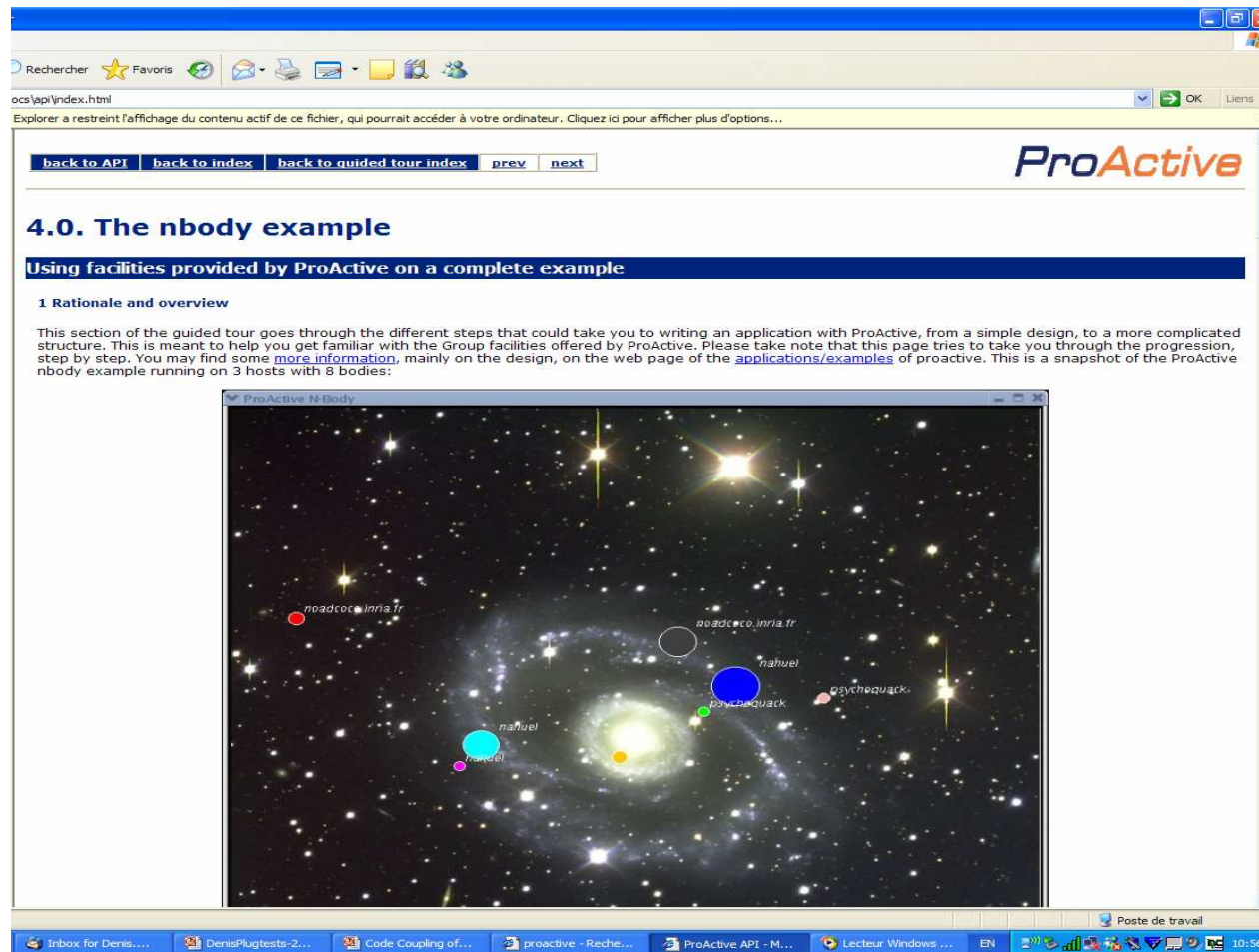
A Generic Version of Jem3D

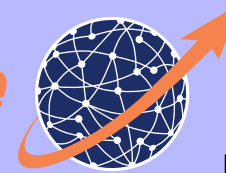


Code Coupling : Vibro Acoustic (courtesy of EADS)



N-Body Particules



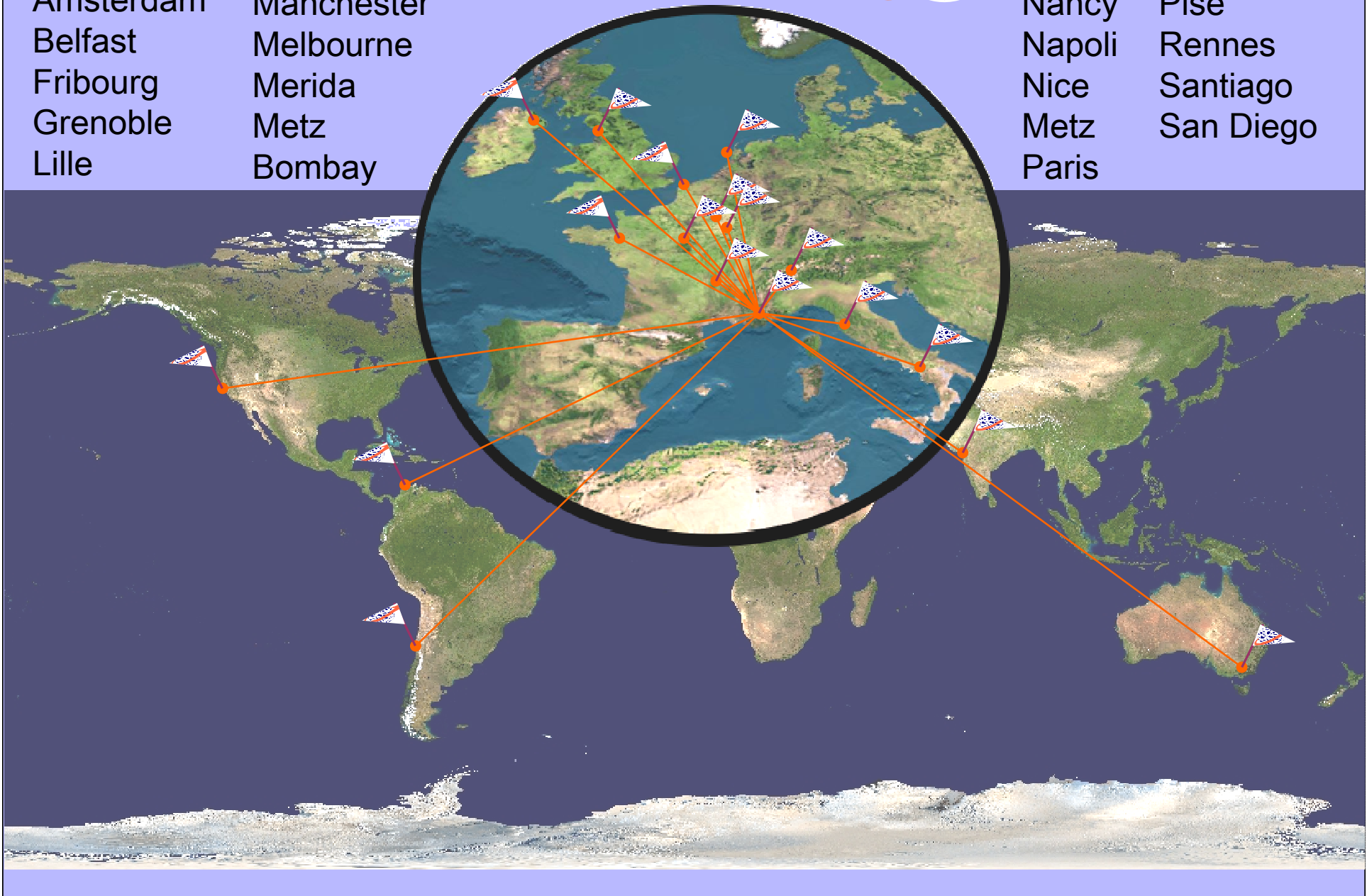


Amsterdam
Belfast
Fribourg
Grenoble
Lille

Manchester
Melbourne
Merida
Metz
Bombay

Nancy
Napoli
Nice
Metz
Paris

Pise
Rennes
Santiago
San Diego

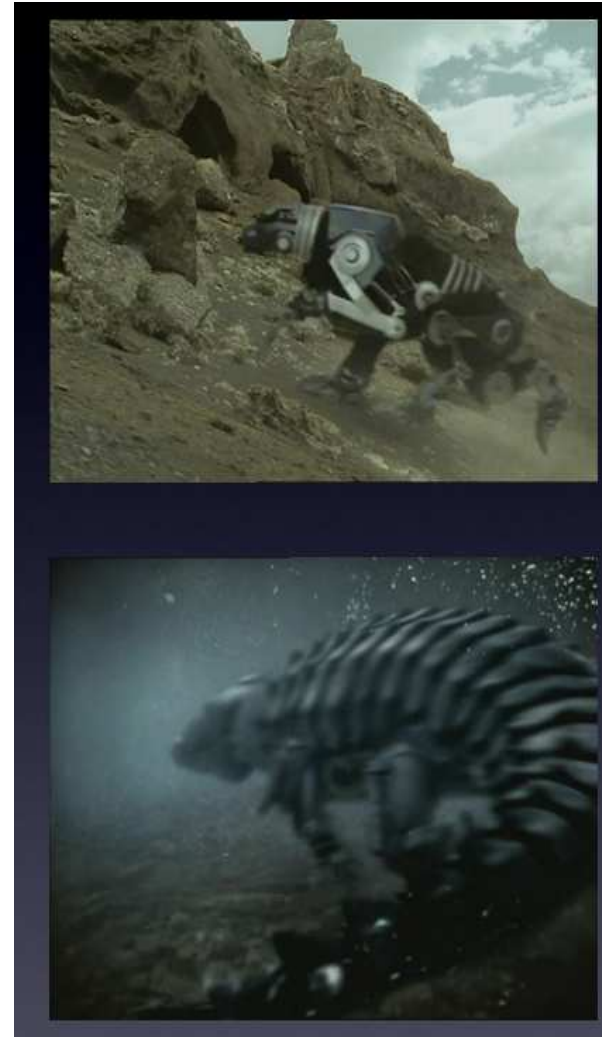


Mikros Image: Post Production

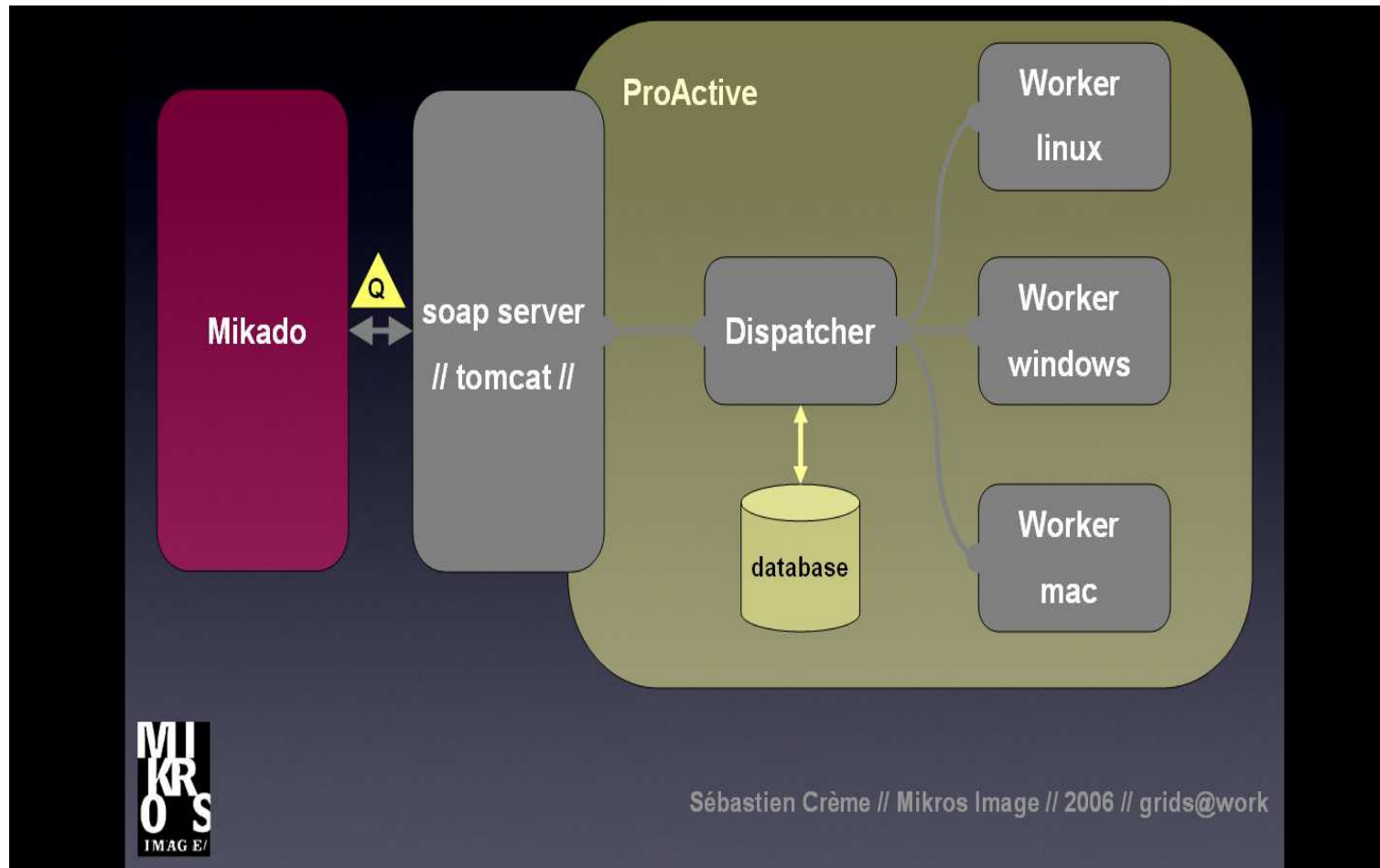
Frames Making of Nissan



Sébastien Crème // Mikros Image // 20



Mikros Image: Post Production



Conclusions and A Few Directions



A Strong Programming Model + Components + Env

NOT discussed so far: Web Service, FT, Exceptions

FACTS AND FIGURES

53 years of computation in 6 months in Desktop P2P

Deployed at once on 2111 CPUs (**PLUGTESTS** on ssh, Globus, LSF, ...)

(Close to) Beating Fortran on an Electromagnetic Application

PERSPECTIVES:

Behavior Specification, Fully Integrated Debugger

A great alchemy for the Grid + SOA:

Asynchrony + Wait By Necessity + Groups + Components



ProActive and SOA Integration

