Des Multi-Cores aux Clouds avec ProActive Parallel Suite

D. Caromel, et al.

Agenda

1. **Background: INRIA, ActiveEon**
2. **ProActive Open Source:**
   Programming, Scheduling, Resourcing
3. **Cloud Seeding with GPU**
4. **ProActive PACA GRID:**
   Cloud Portal with GPUs

Parallelism is Complex:
Hiding Infra Details and Seamless Integration with Multi-Cores and GPU

1. Background

OASIS Team & INRIA

- A joint team, Now about 35 persons
- 2004: First ProActive User Group
- 2009, April: ProActive 4.1, Distributed & Parallel: From Multi-cores to Enterprise GRIDs

OASIS Team Composition (35)

- Researchers (5):
  - D. Caromel (UNSA, Det. INRIA
  - E. Madelaine (INRIA)
  - F. Baude (UNSA)
  - F. Huet (UNSA)
  - L. Henrio (CNRS)
- PhDs (11):
  - Antonio Cansado (INRIA, Coni
  - Brian Amedro (SCS-Agos)
  - Cristian Ruz (INRIA, Conicyt)
  - Elton Mathias (INRIA-Cordi)
  - Imen Filali (SCS-Agos / FP7 S
  - Marcela Rivera (INRIA, Conicy
  - Muhammad Khan (STIC-Asia)
  - Paul Naoumenko (INRIA/Régic
  - Viet Dung Doan (FP6 Bionets)
  - Virginie Contes (SOA4ALL)
  - Guilherme Pezzi (AGOS, CIFR
- + Visitors + Interns

Located in Sophia Antipolis, between Nice and Cannes,
Visitors Welcome!

ActiveEon
SCALE BEYOND LIMITS

Startup Company Born of INRIA

Activeeon
SCALE BEYOND LIMITS

Some Customers: CEA THALES

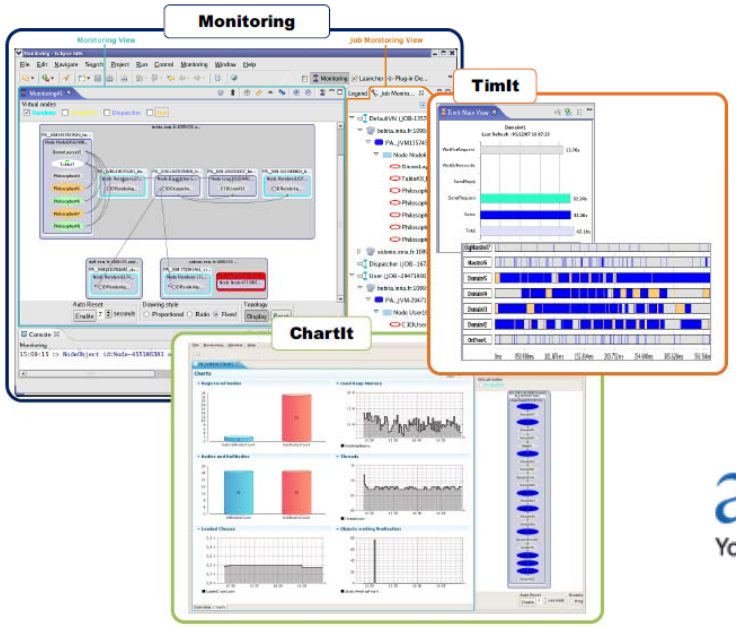
Some Partners: BULL Architect of an Open World, intel Software Partner, NVIDIA, Microsoft, CARRI Systems

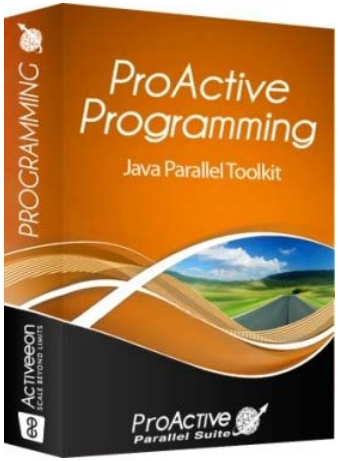
- Co-developing, Support for ProActive Parallel Suite
- Worldwide Customers: Fr, UK, Boston USA

2. ProActive Parallel Suite

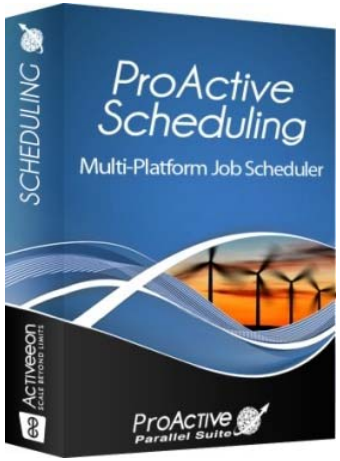
Product: ProActive Parallel Suite

ProActive Parallel Suite





Java Parallel Toolkit



Multi-Platform Job Scheduler

Resource Manager

amadeus
Your technology partner

Used in Production Today:
50 Cores → 300 Cores early 2010

Strong Differentiation:

- ❑Java Parallel Programming + Integration +
- ❑Portability: Linux, Windows, Mac +
- ❑Versatility: Desktops, Cluster, Grid, Clouds = Perfect Flexibility

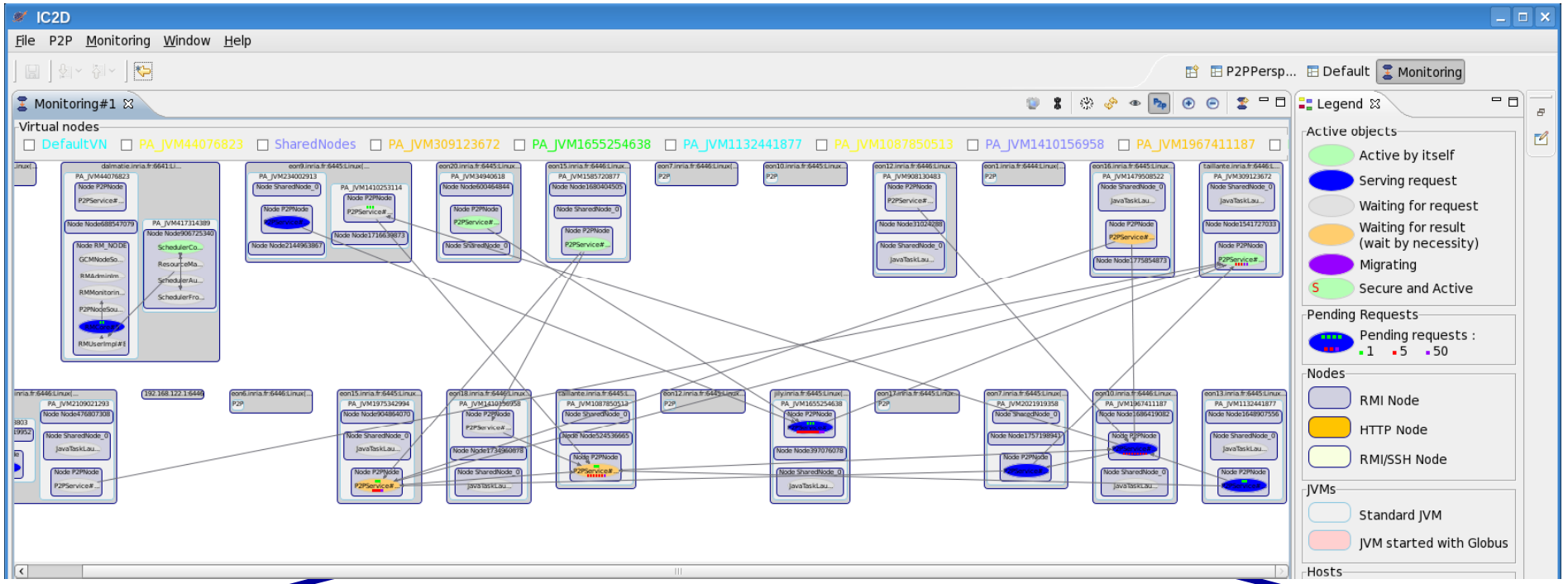
Activeeon SCALE BEYOND LIMITS

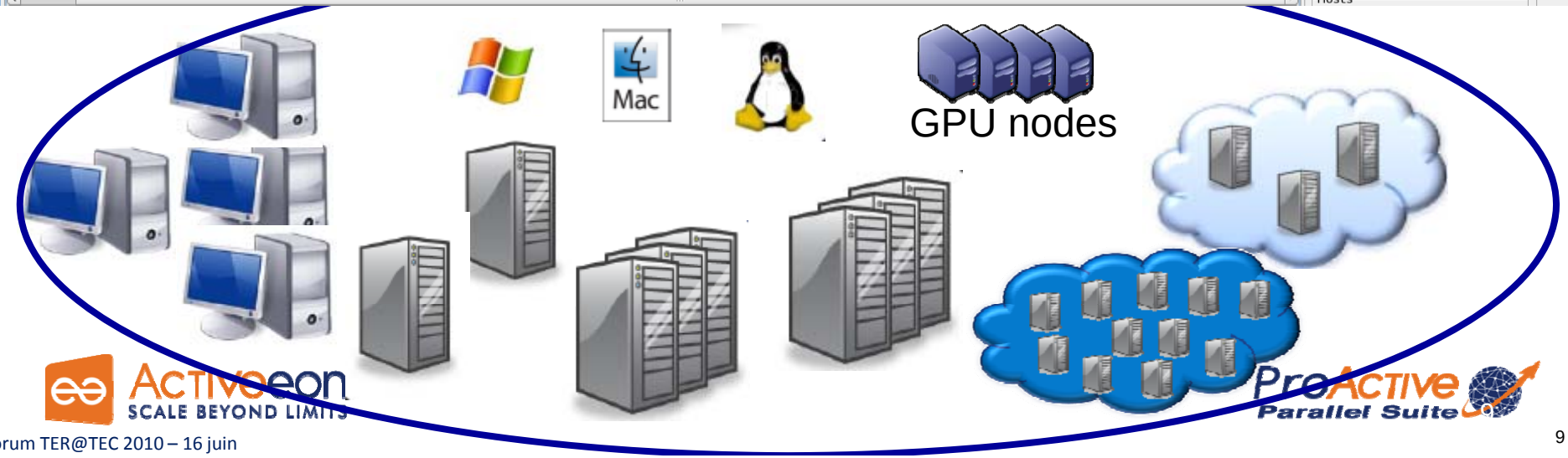
ProActive Parallel Suite

OW2 Consortium

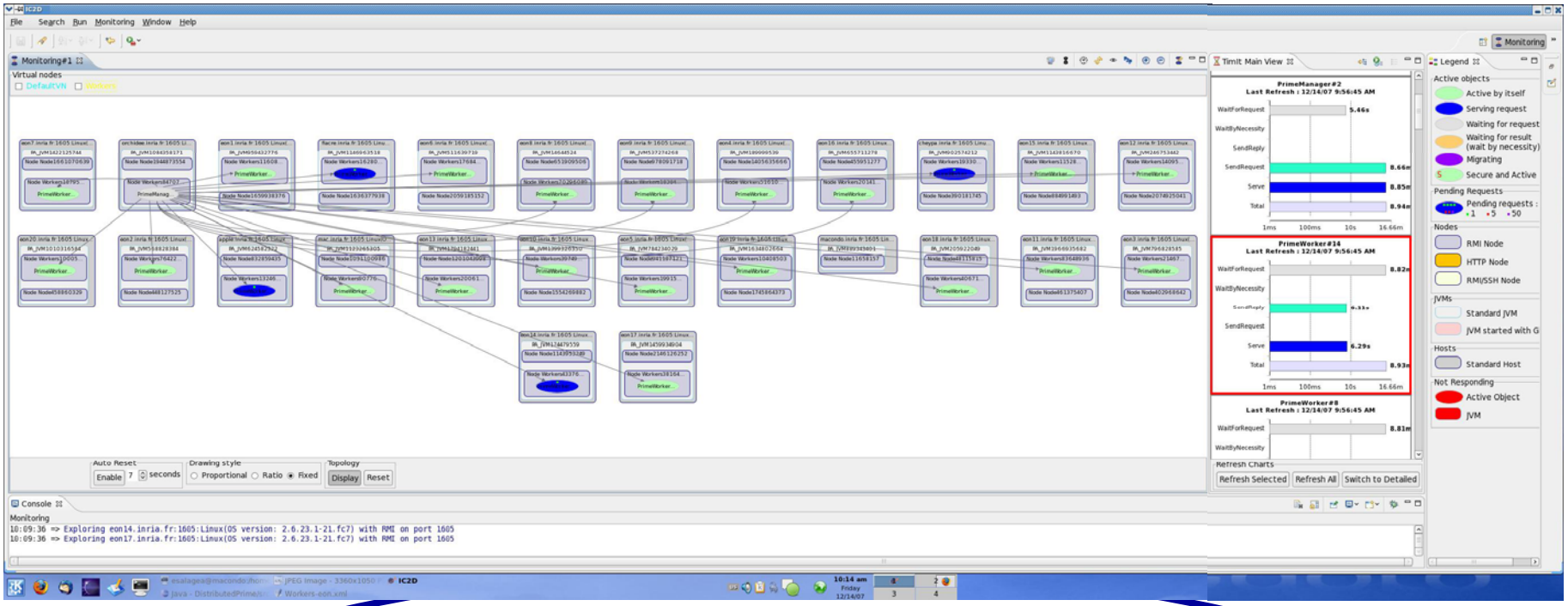
ProActive Programming: Active Objects

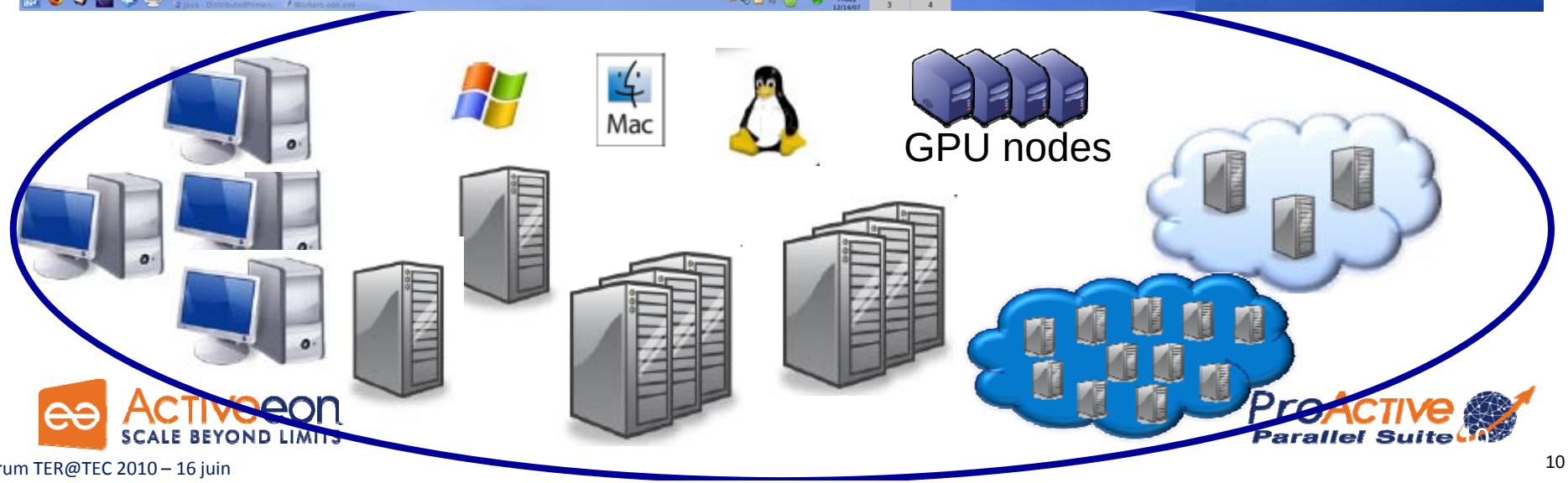
ProActive Programming View





ProActive Programming View

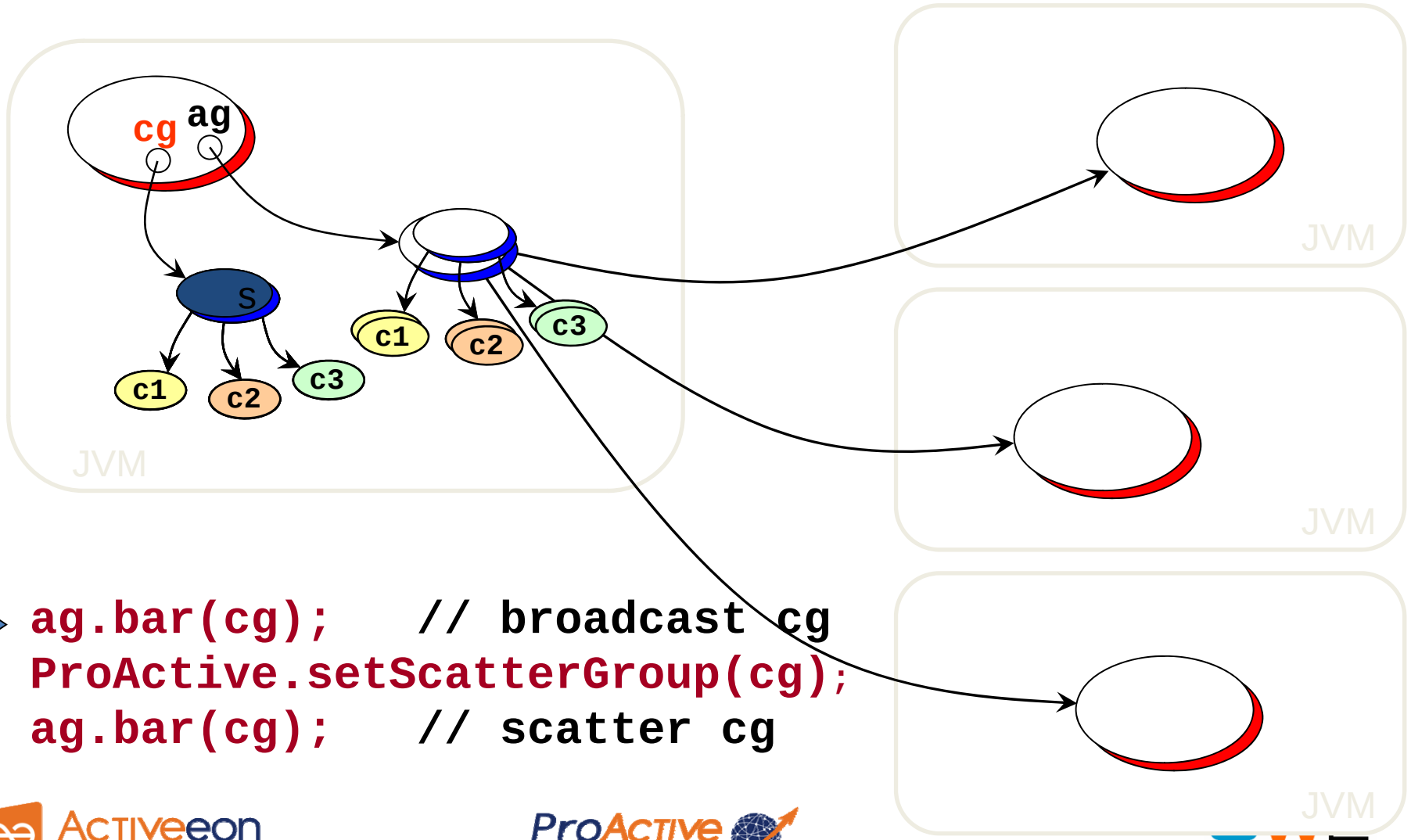




TYPED ASYNCHRONOUS GROUPS

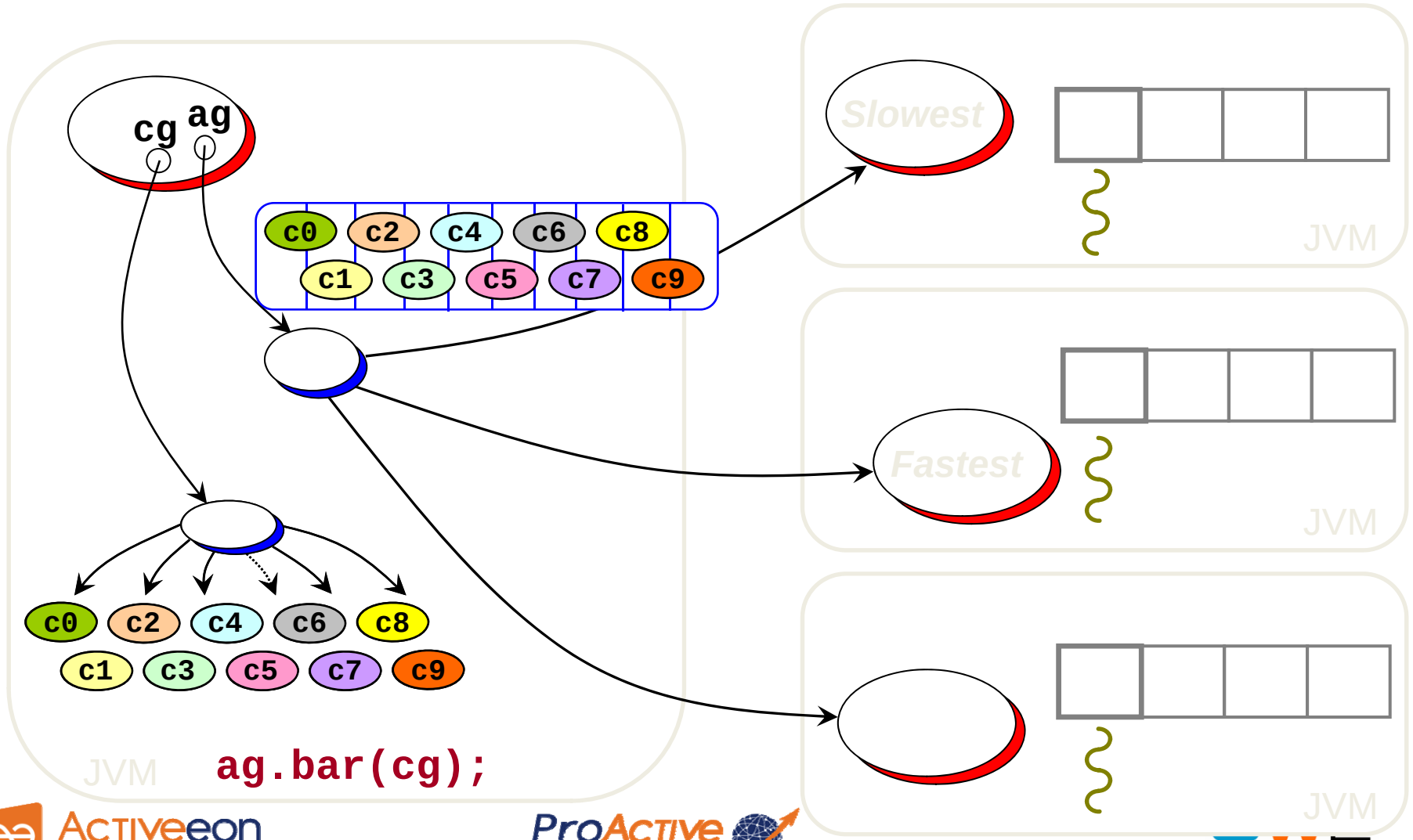
Broadcast and Scatter

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
```

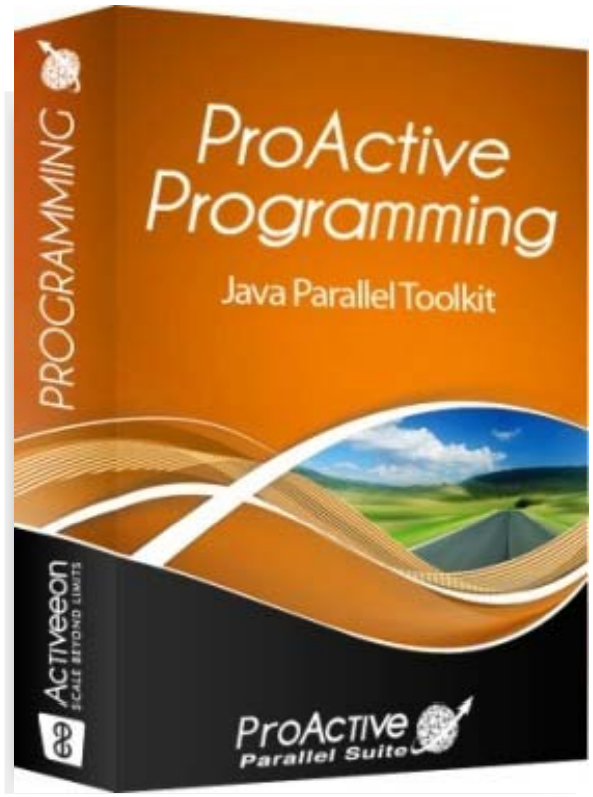
Dynamic Dispatch Group



Abstractions for Parallelism

The right Tool to do the Task right

ProActive Parallel Suite



- Workflows in Java
- Master/Workers
- SPMD
- Components
- …

Core API
Active Objects
Asynchrony
Futures
Groups
Mobile Agents
MOP / AOP

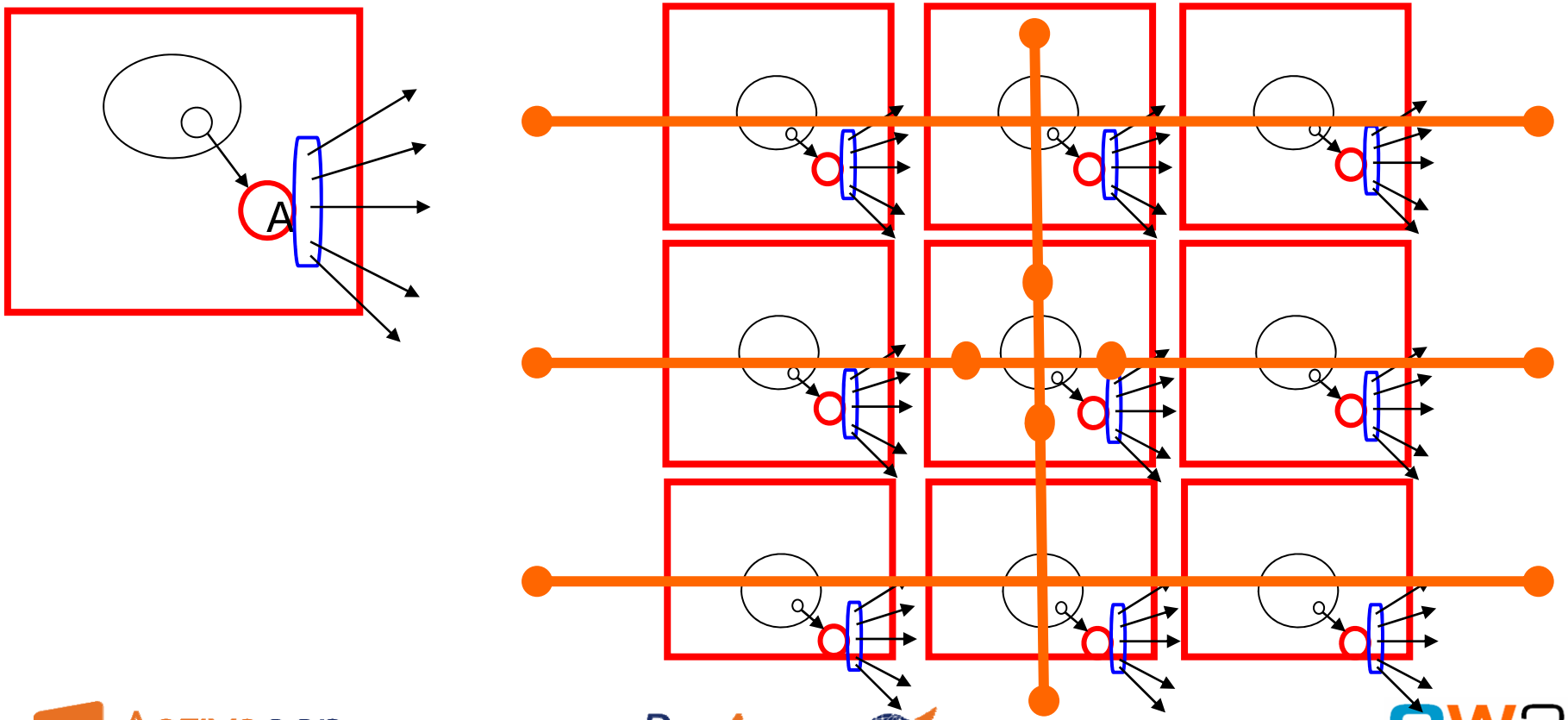
Object-Oriented SPMD

Key Point

- **"MPI and programming languages from the 60's will not make it"**
  - **Jack Dongarra,** 2/13/2009,
  - Wake Forest University talk
- **"It is time to get ride of MPI"**
  - **Alan Edelman, MIT,** 06/16/2010, Ter@tec
  - ScilabTec'10 Users' Day (30 mn. ago)

OO SPMD: Object-Oriented SPMD

```
A ag = newSPMDGroup ("A", […], VirtualNode)
        // In each member
        myGroup.barrier ("2D"); // Global Barrier
        myGroup.barrier ("vertical"); // Any Barrier
        myGroup.barrier ("north","south","east","west");
```

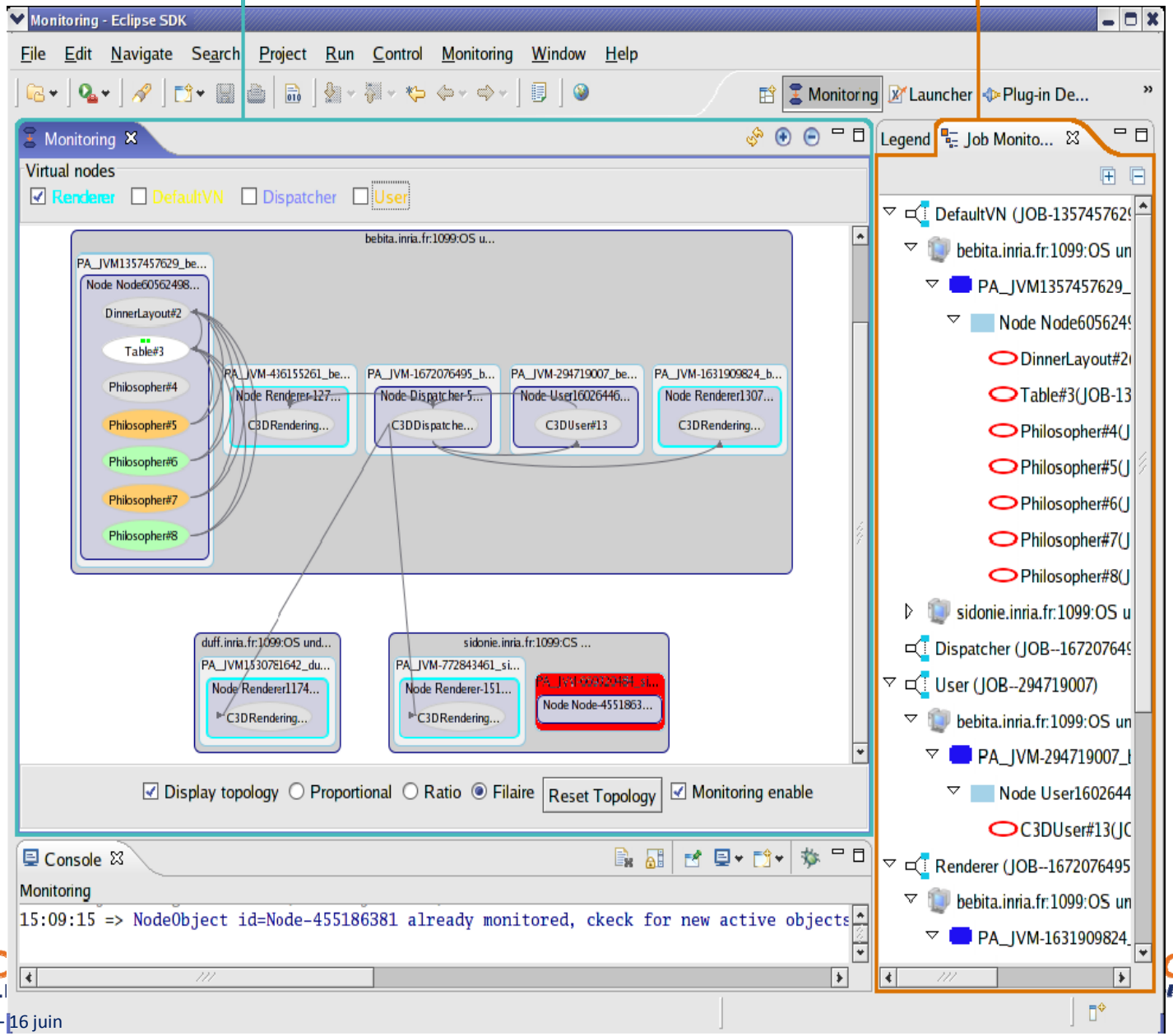


IC2D: Optimizing

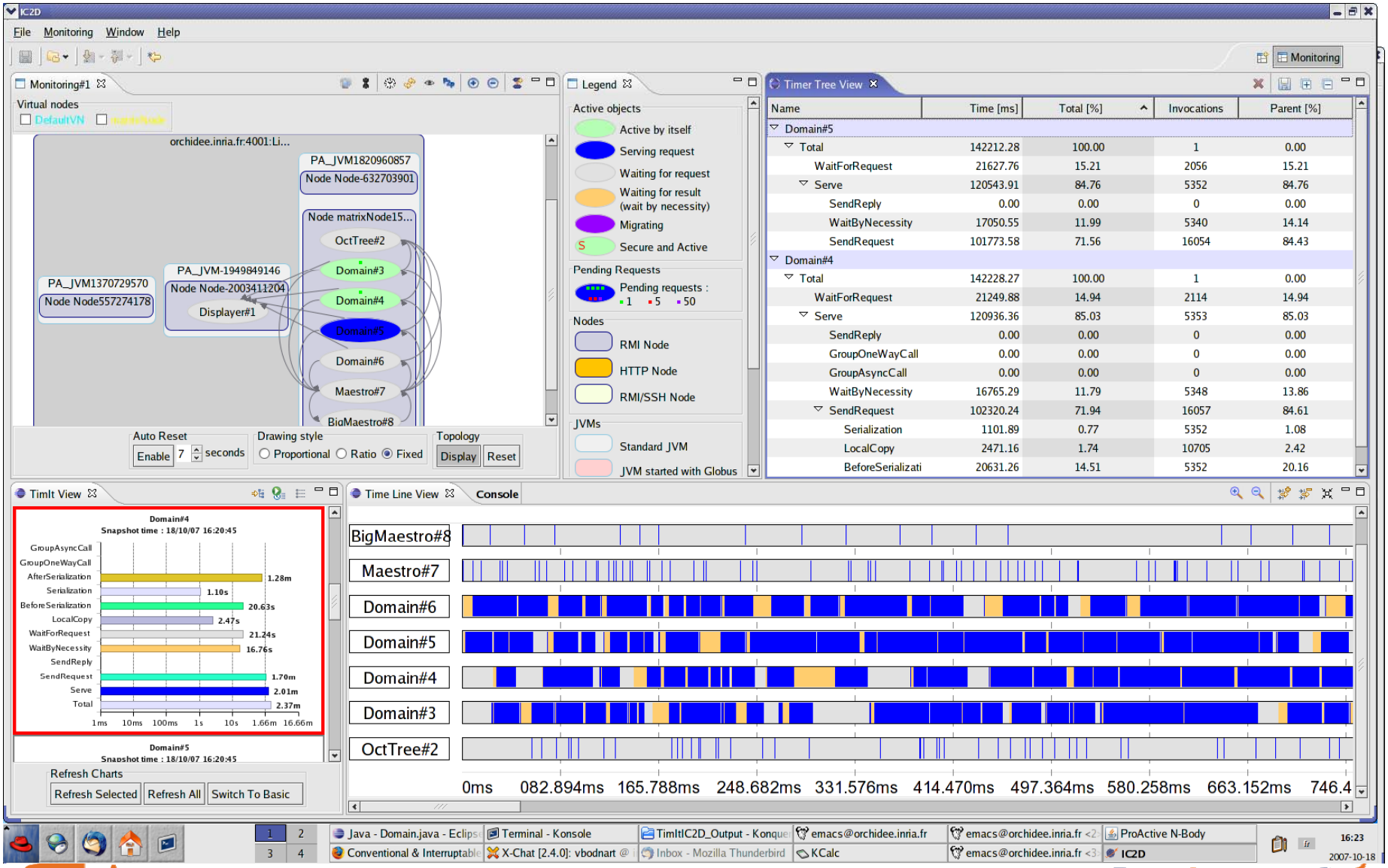
IC2D

Monitoring View

Job Monitoring View



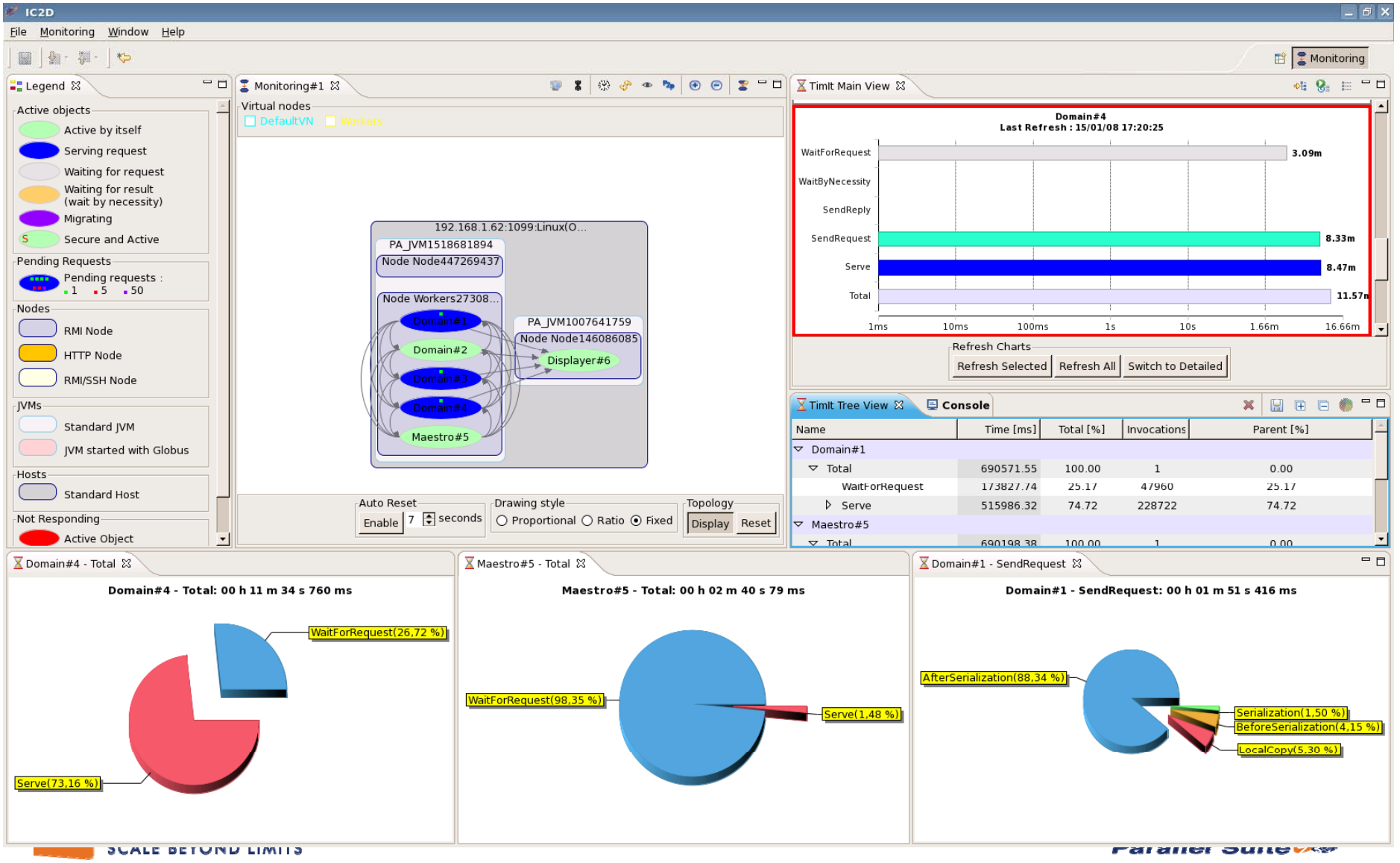
IC2D


IC2D
File Monitoring Window Help
Monitoring
Monitoring#1
Virtual nodes
DefaultVN
orchidee.inria.fr:4001:Li...
PA_JVM1820960857
Node Node-632703901
Node matrixNode15...
OctTree#2
Domain#3
Domain#4
Domain#5
Domain#6
Maestro#7
BigMaestro#8
PA_JVM1370729570
Node Node557274178
PA_JVM-1949849146
Node Node-2003411204
Displayer#1
Auto Reset
Enable 7 seconds
Drawing style
Proportional Ratio Fixed
Topology
Display Reset
Legend
Active objects
Active by itself
Serving request
Waiting for request
Waiting for result (wait by necessity)
Migrating
Secure and Active
Pending Requests
Pending requests : 1 5 50
Nodes
RMI Node
HTTP Node
RMI/SSH Node
JVMs
Standard JVM
JVM started with Globus
Timer Tree View
Name | Time [ms] | Total [%] | Invocations | Parent [%]
Domain#5
Total 142212.28 100.00 1 0.00
WaitForRequest 21627.76 15.21 2056 15.21
Serve 120543.91 84.76 5352 84.76
SendReply 0.00 0.00 0 0.00
WaitByNecessity 17050.55 11.99 5340 14.14
SendRequest 101773.58 71.56 16054 84.43
Domain#4
Total 142228.27 100.00 1 0.00
WaitForRequest 21249.88 14.94 2114 14.94
Serve 120936.36 85.03 5353 85.03
SendReply 0.00 0.00 0 0.00
GroupOneWayCall 0.00 0.00 0 0.00
GroupAsyncCall 0.00 0.00 0 0.00
WaitByNecessity 16765.29 11.79 5348 13.86
SendRequest 102320.24 71.94 16057 84.61
Serialization 1101.89 0.77 5352 1.08
LocalCopy 2471.16 1.74 10705 2.42
BeforeSerializati 20631.26 14.51 5352 20.16
TimIt View
Domain#4
Snapshot time : 18/10/07 16:20:45
GroupAsyncCall
GroupOneWayCall
AfterSerialization 1.28m
Serialization 1.10s
BeforeSerialization 20.63s
LocalCopy 2.47s
WaitForRequest 21.24s
WaitByNecessity 16.76s
SendReply
SendRequest 1.70m
Serve 2.01m
Total 2.37m
1ms 10ms 100ms 1s 10s 1.66m 16.66m
Domain#5
Snapshot time : 18/10/07 16:20:45
Refresh Charts
Refresh Selected Refresh All Switch To Basic
Time Line View
Console
BigMaestro#8
Maestro#7
Domain#6
Domain#5
Domain#4
Domain#3
OctTree#2
0ms 082.894ms 165.788ms 248.682ms 331.576ms 414.470ms 497.364ms 580.258ms 663.152ms 746.4
Java - Domain.java - Eclips
Terminal - Konsole
TimItIC2D_Output - Konquer
emacs@orchidee.inria.fr
emacs@orchidee.inria.fr <2
ProActive N-Body
Conventional & Interruptable
X-Chat [2.4.0]: vbodnart @
Inbox - Mozilla Thunderbird
KCalc
emacs@orchidee.inria.fr <3
IC2D
16:23
2007-10-18

ChartIt


PA_JVM251111462
Charts
Chart#1 [LoadedClassCount]
2,5 k
2,4 k
2,3 k
2,2 k
2,1 k
2,0 k
LoadedClassCount
Chart#3 [ThreadCount]
40
35
30
25
ThreadCount
Chart#2 [UsedHeapMemory]
7,0 M
6,0 M
5,0 M
4,0 M
UsedHeapMemory
Chart#4 [InternalBodiesCount, UserBodiesCount]
5
4
3
2
1
0
2
4
InternalBodiesCount
UserBodiesCount
Chart#5 [ThreadCount] Meter
0
5
10
15
20
25
30
35
ThreadCount
35.0
Chart#6 [AllBodiesCount, InternalBodiesCount, UserBodiesCount, HalfBodiesCount]
7
6
5
4
3
2
1
0
6
2
4
5
AllBodiesCount
InternalBodiesCount
UserBodiesCount
HalfBodiesCount

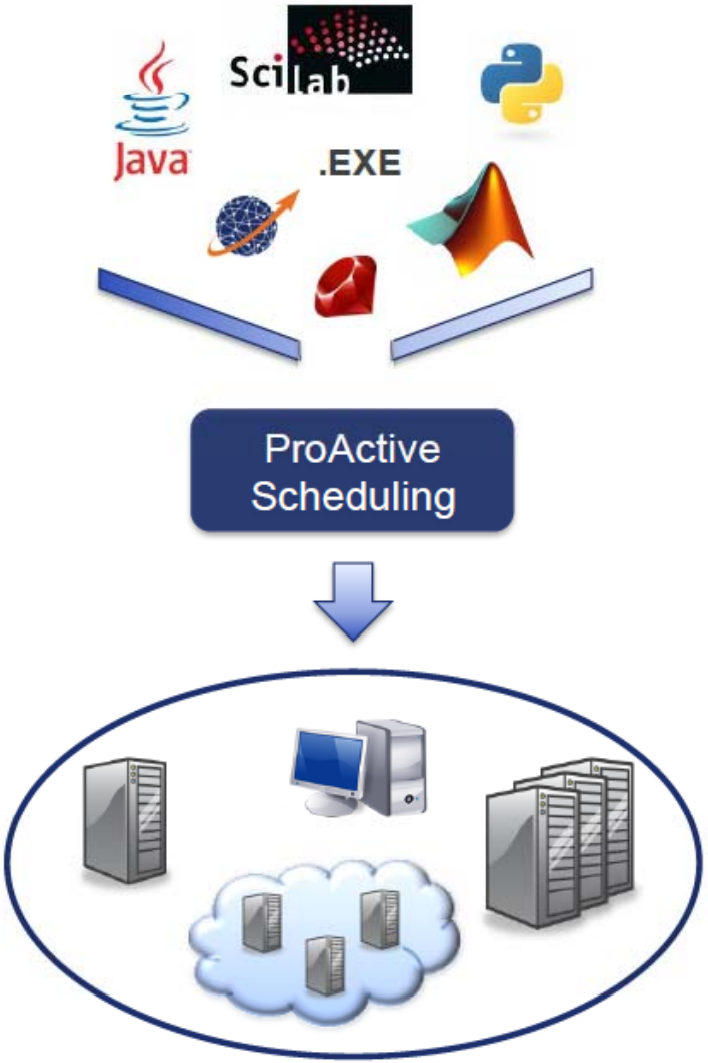
Pies for Analysis and Optimization



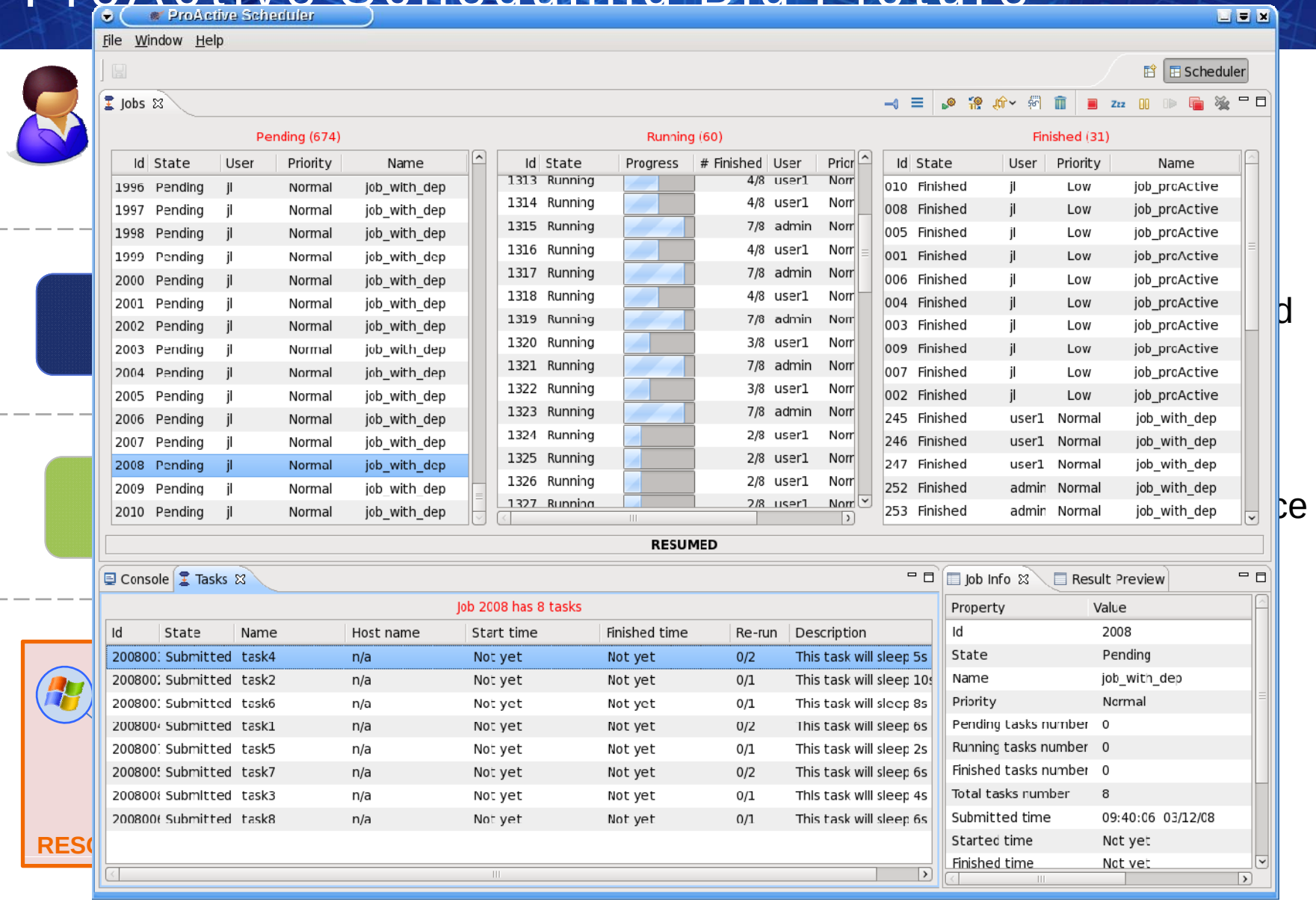
Video 1:
IC2D Optimizing
Monitoring, Debugging, Optimizing

Scheduling & Resourcing

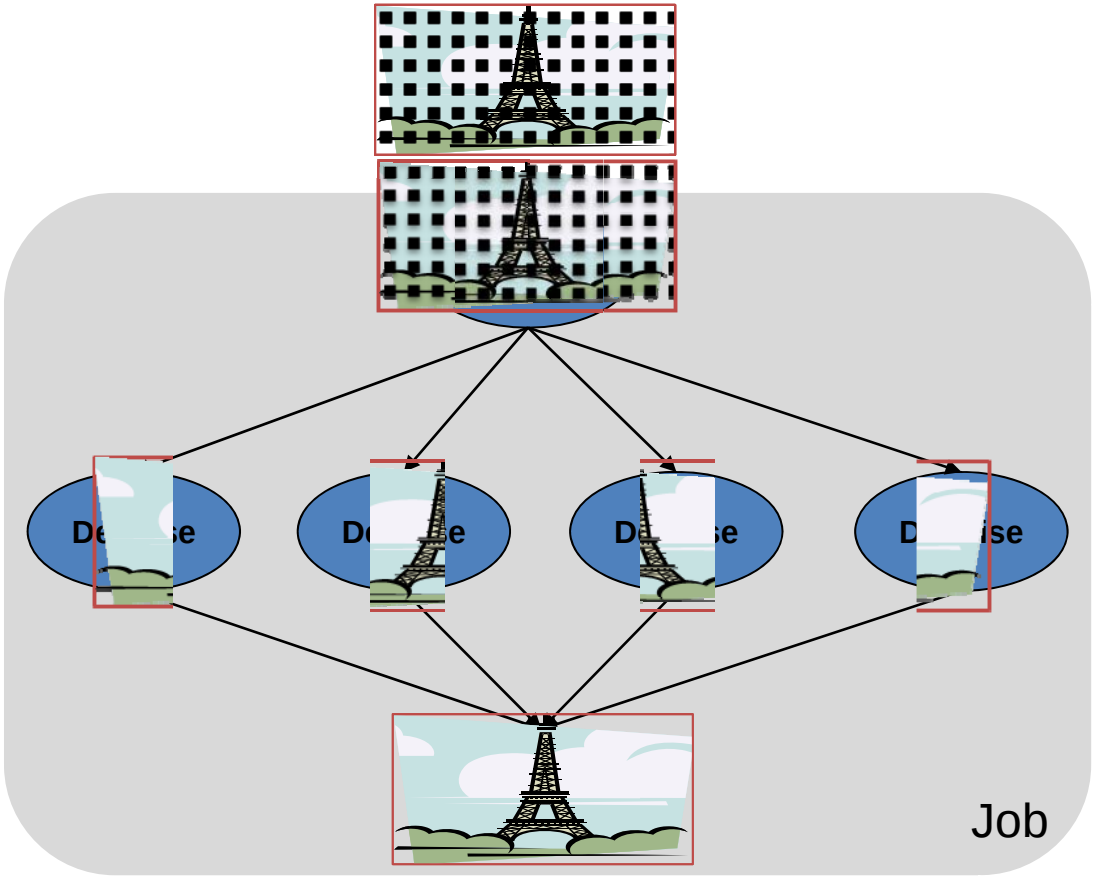
ProActive Scheduling



ProActive Scheduling Big Picture



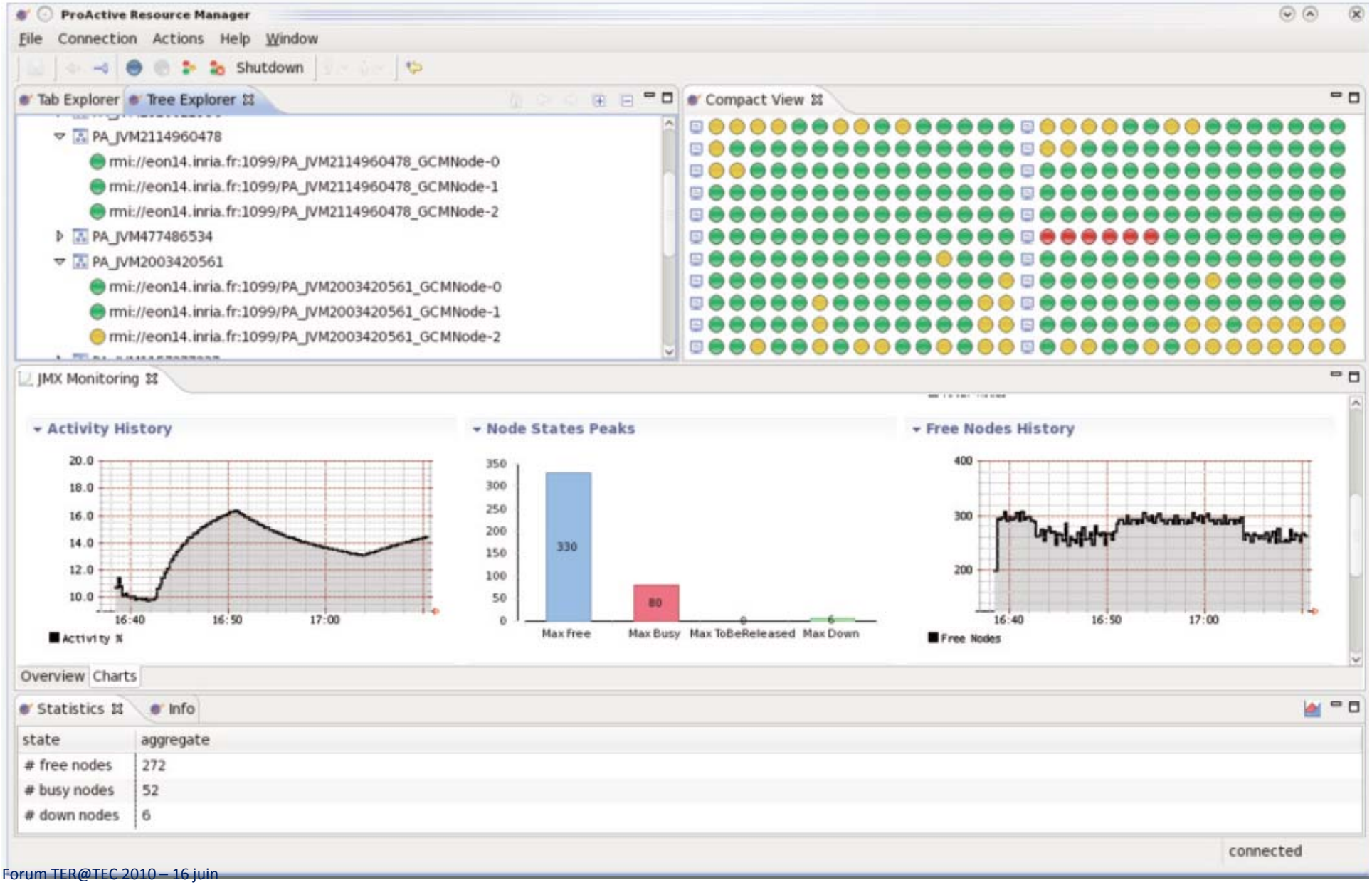
Workflow Example : Picture Denoising



•with selection on native executable availability (ImageMagik, GREYstoration)
  • Multi-platform selection and command generation
•with file transfer in pre/post scripts

ProActive Resourcing

RESOURCING User Interface



Clusters to Grids to Clouds:
e.g. on Amazon EC2

Node source Usecase : Configuration for external cloud with EC2

ProActive Scheduler

ProActive Resource Manager

Static Policy

LSF

Timing Policy 12/24

Desktops

Dynamic Workload Policy

EC2

Dedicated resources

Desktops

Amazon EC2

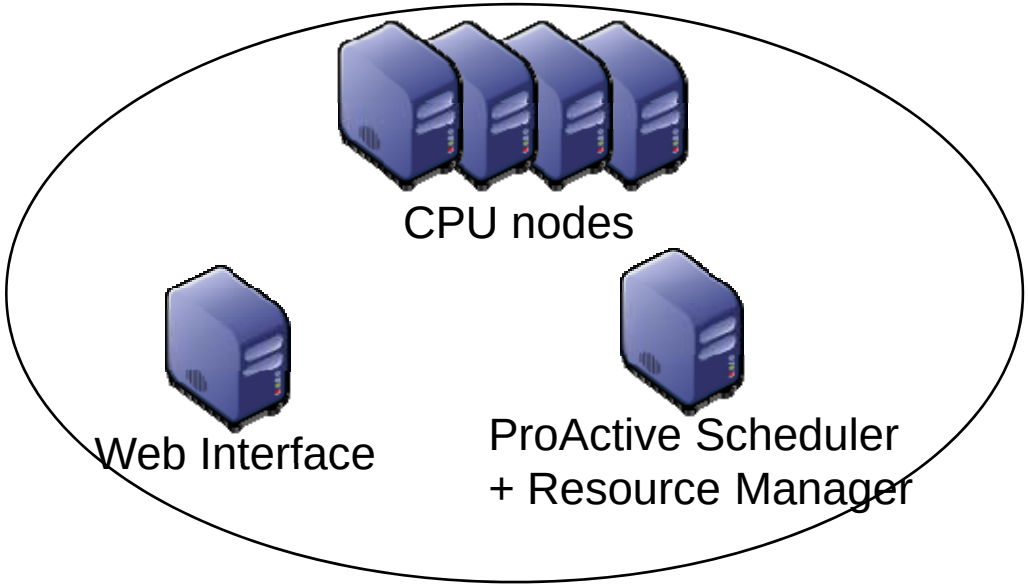
Video 2:
Scheduler, Resource Manager

3.
Cloud Seeding with GPU

Cloud Seeding with ProActive

- Amazon EC2 Execution
- *Cloud Seeding* strategy to mix heterogeneous computing resources :
  - External GPU resources

Cloud Seeding with ProActive

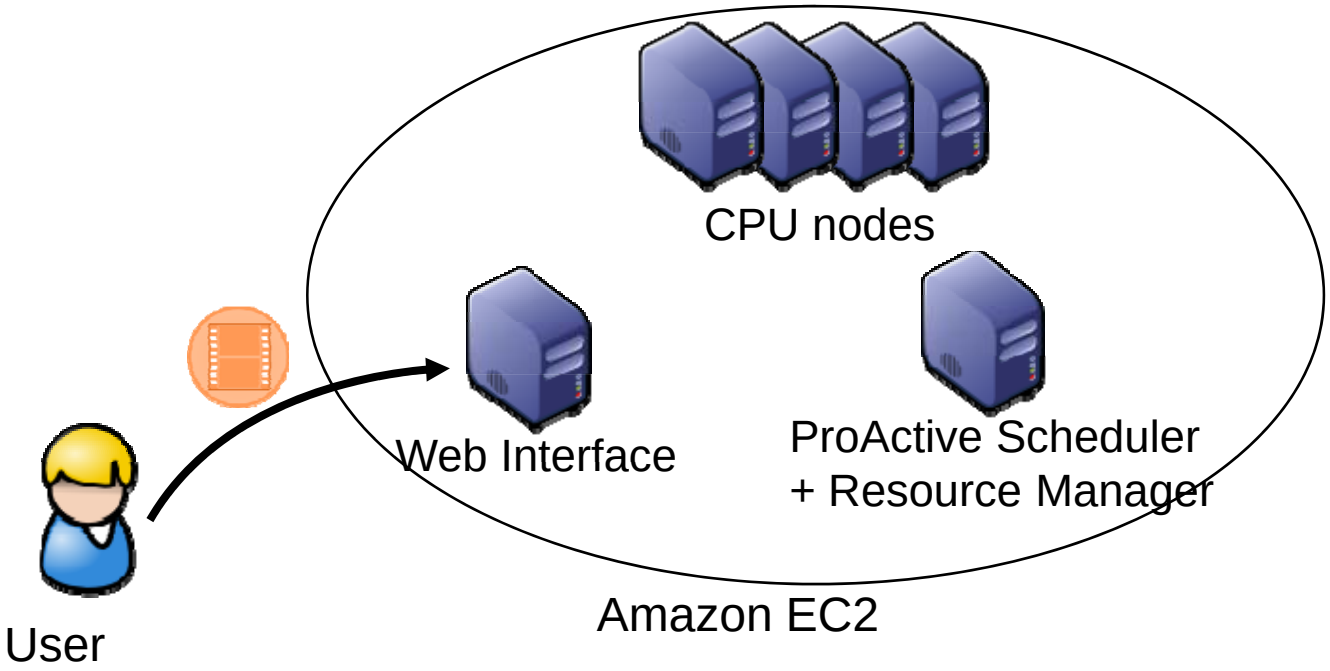


User

Noised video file

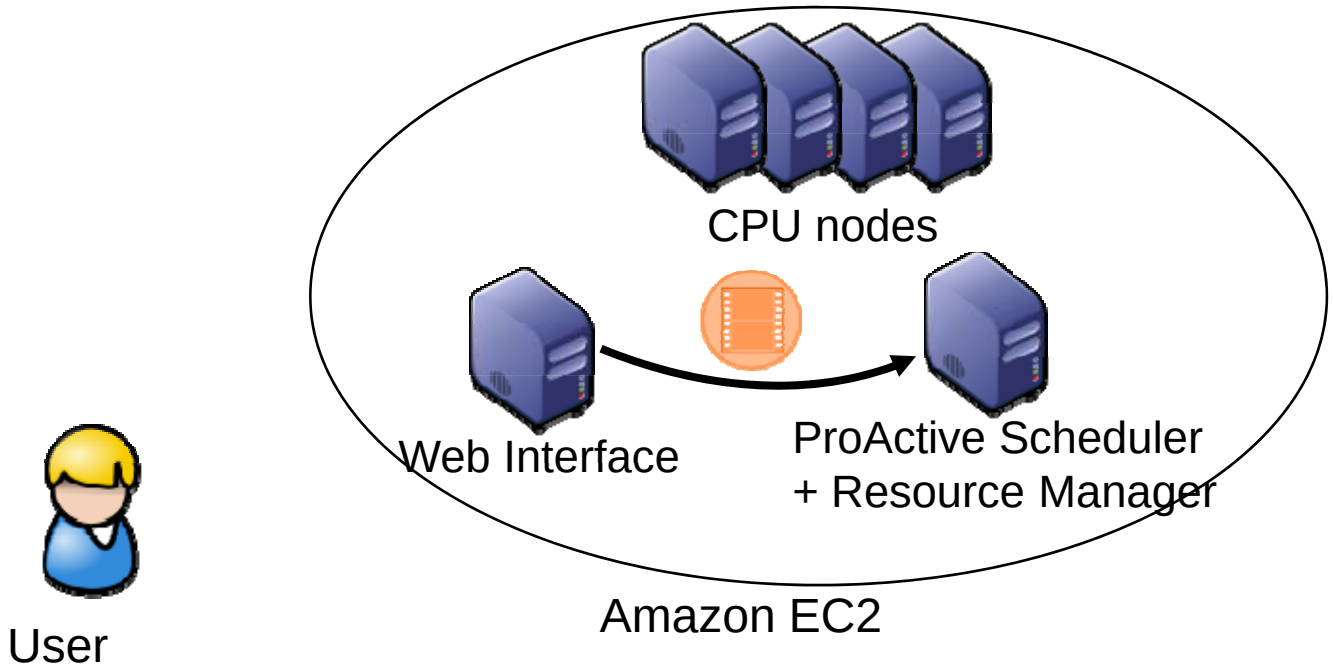
GPU nodes

Cloud Seeding with ProActive



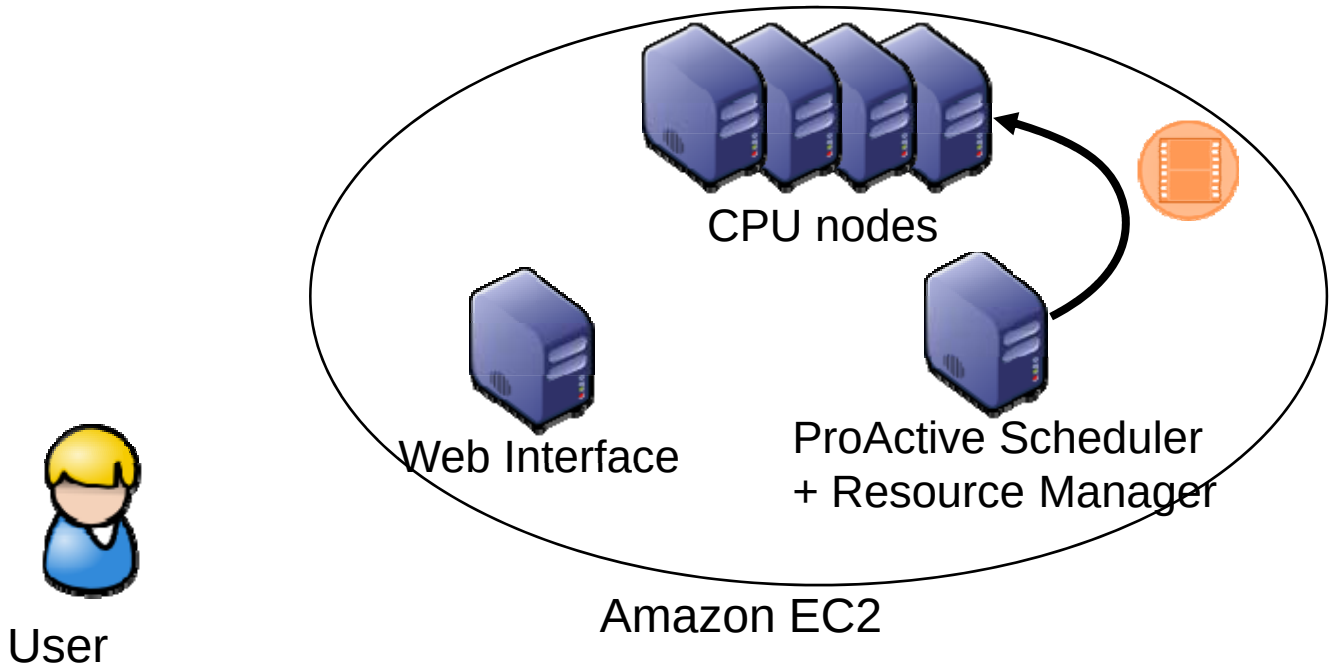
User submit its noised video to the web interface

Cloud Seeding with ProActive



Web Server submit a denoising job the ProActive Scheduler

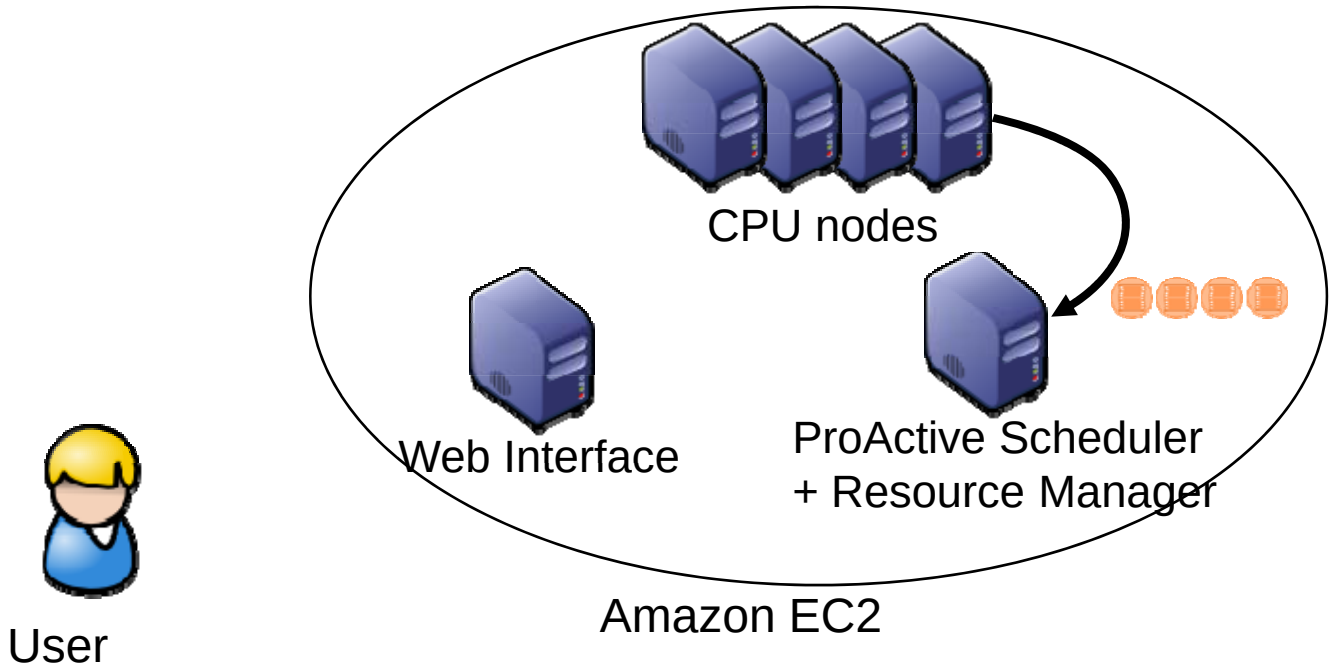
Cloud Seeding with ProActive



CPU nodes are used to split the video into smaller ones

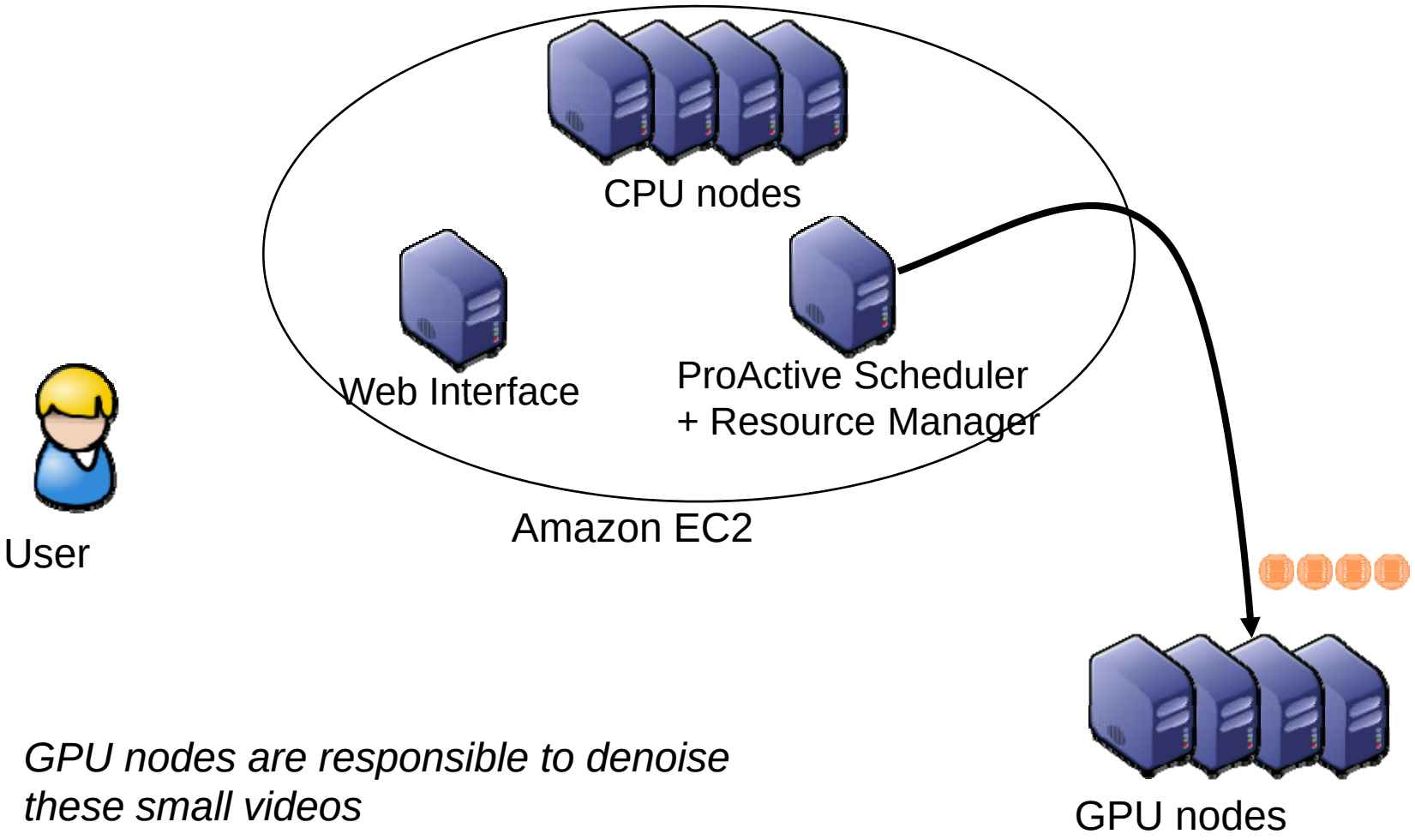
GPU nodes

Cloud Seeding with ProActive



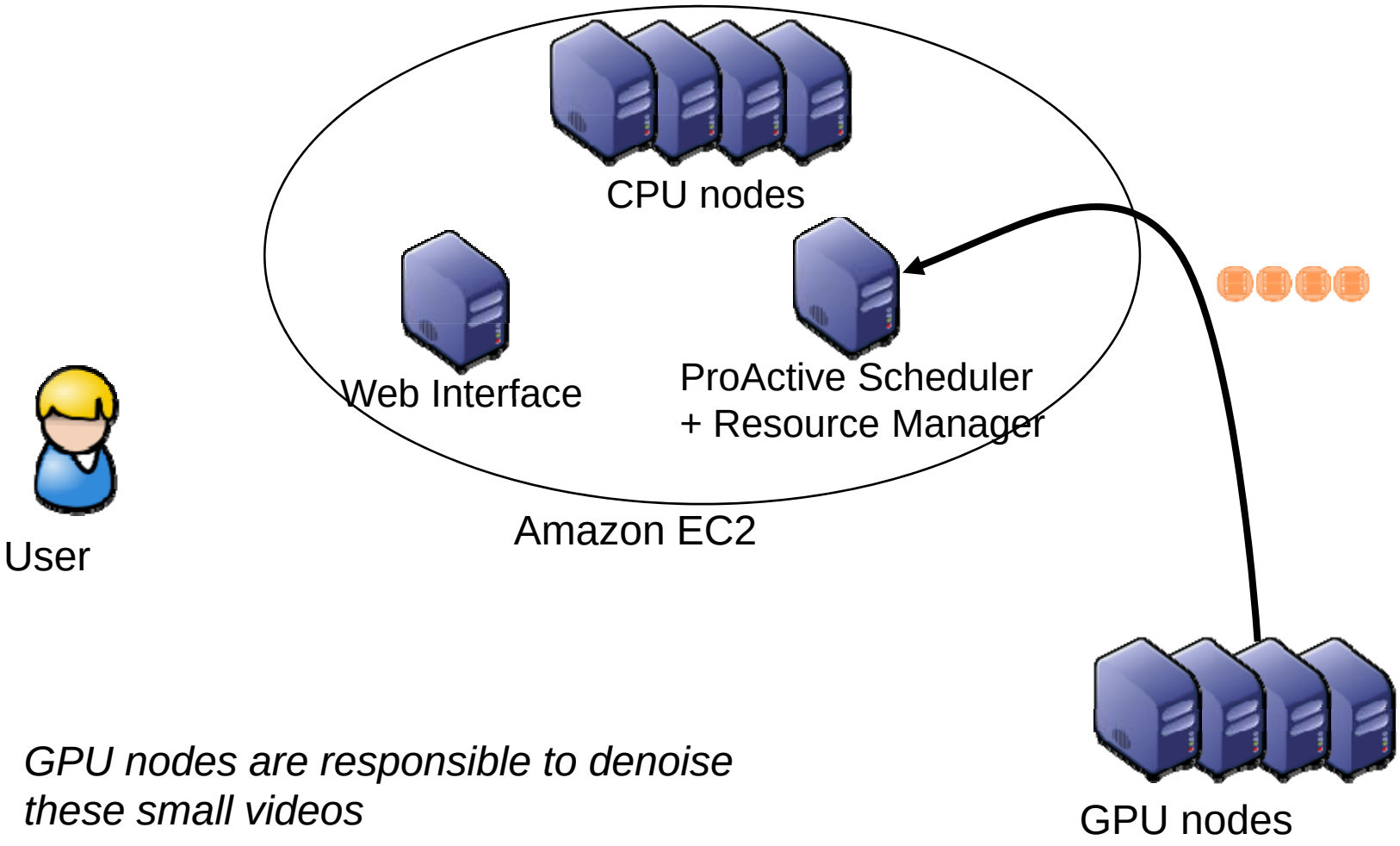
CPU nodes are used to split the video into smaller ones

Cloud Seeding with ProActive



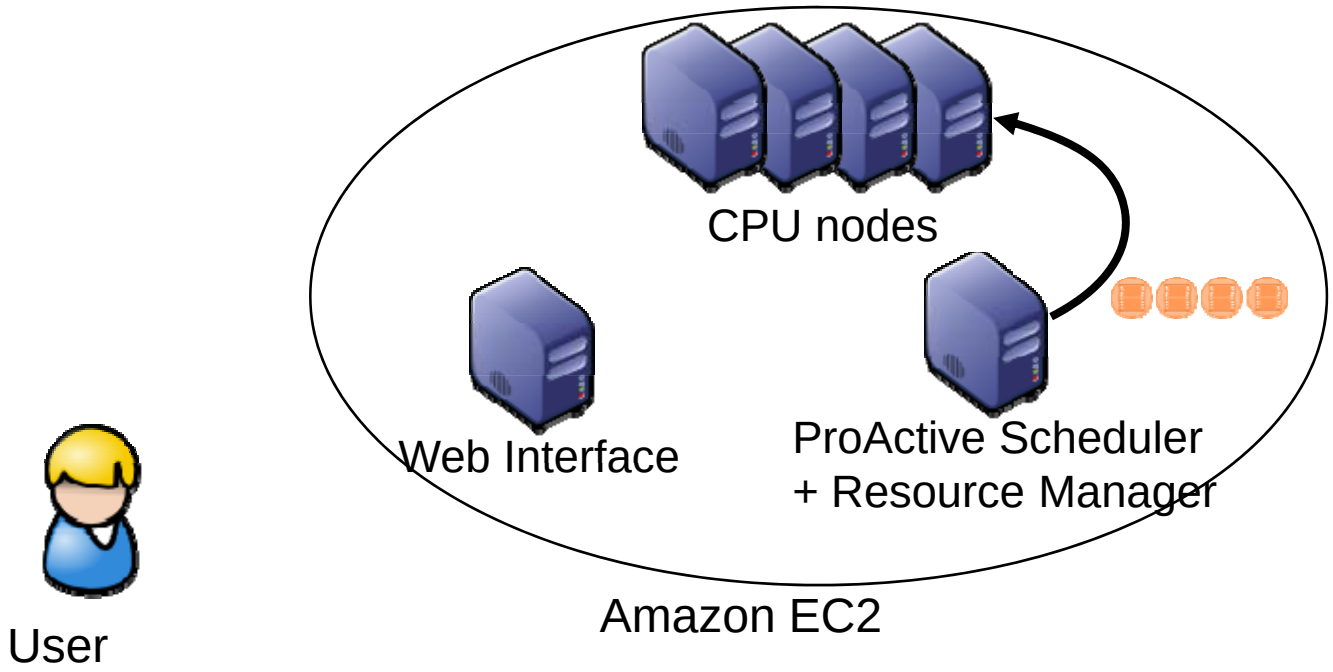
GPU nodes are responsible to denoise these small videos

Cloud Seeding with ProActive



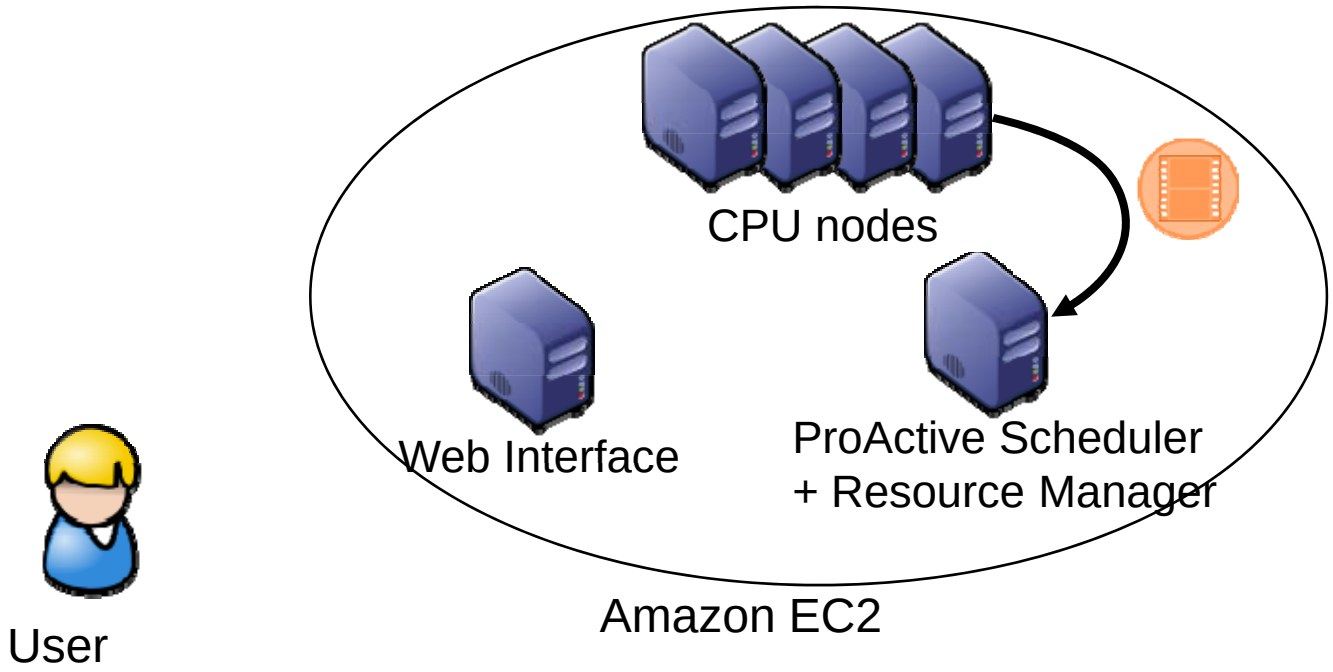
GPU nodes are responsible to denoise these small videos

Cloud Seeding with ProActive



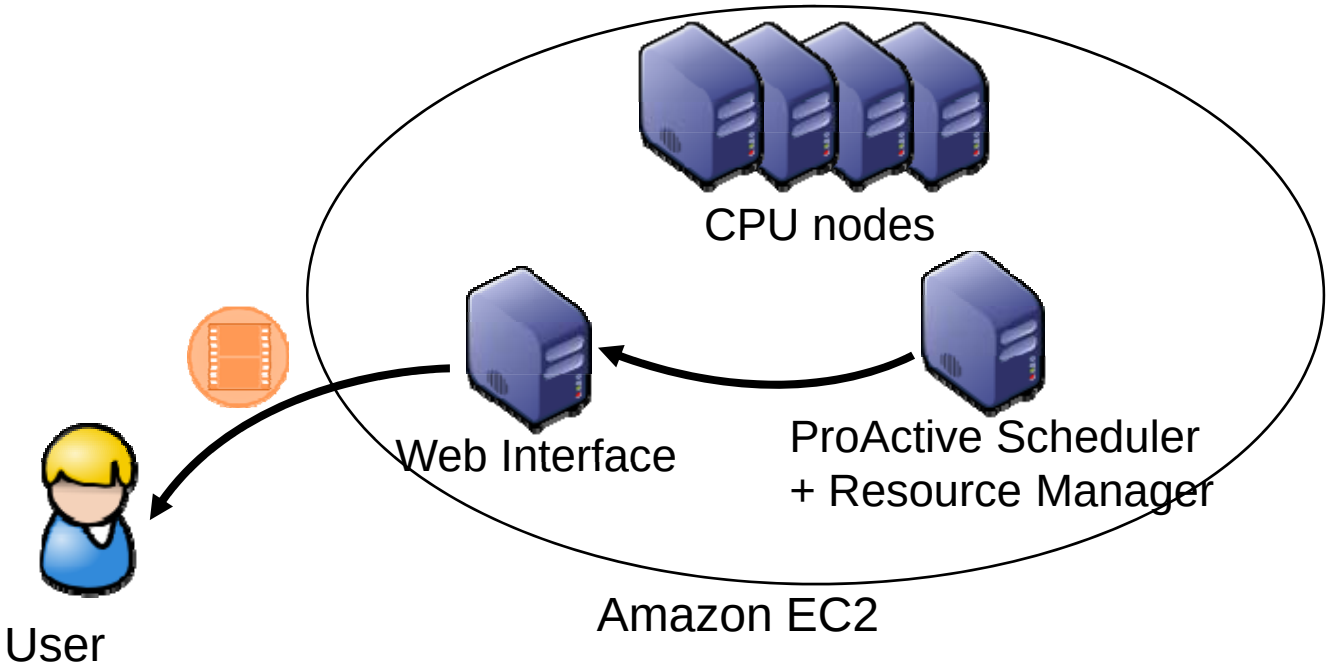
CPU nodes merge the denoised video parts

Cloud Seeding with ProActive



CPU nodes merge the denoised video parts

Cloud Seeding with ProActive



The final denoised video is sent back to the user

4.
ProActive PACA GRID: Cloud Portal with GPUs

ProActive Parallel Suite

ProActive PACA Grid

an INRIA, UNSA, CNRS-I3S, PACA Computing Cloud

Welcome | Monitor | Nodes configuration | Download | Tutorials | links

Main

- Welcome
- Monitor
- Nodes configuration
- Download
- Tutorials
- links

INRIA

Université Nice SOPHIA ANTIPOLIS

cnrs i3s sophia antipolis

Région PACA

Welcome

Welcome to ProActive PACA Grid web site

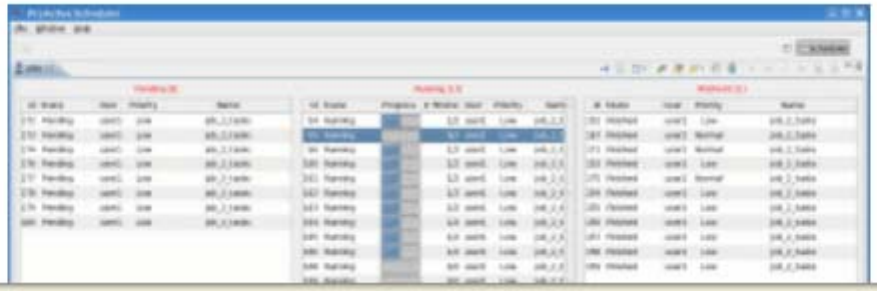
ProActive PACA Grid is a set of machines accessible via Graphical Interactive interfaces based on ProActive Parallel Suite (http://proactive.inria.fr). The machines are currently deployed within INRIA Sophia Antipolis networks. The Cloud aggregates dedicated machines, both Linux and Windows, and spare desktop machines, dynamically added during nights and week-ends. This Grid is available for INRIA and UNSA members that need to accelerate their scientific applications. Upon request, other PACA labs and SMEs can also access the ProActive PACA Grid.

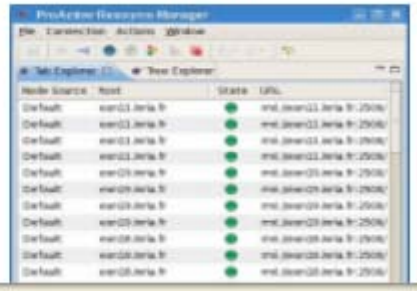
In production today : Download Graphical client in download page and schedule your jobs !

On going : Direct Web Access to CPER Cloud cluster by Java Web Start :

Web Access to the ProActive Scheduler :

Web Access to the ProActive Resource Manager :





The ProActive PACA Grid Platform (4)

Total:

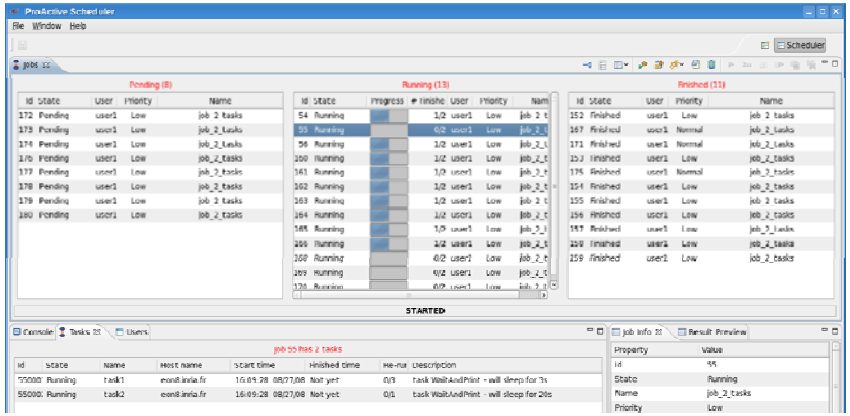
- 816 Cores
- 480 CUDA Cores
- 14.8TB Storage

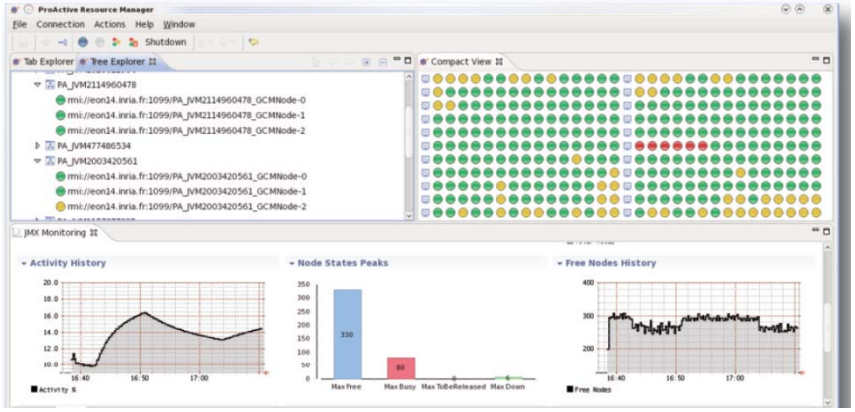
Publically Available Today

ProActive Parallel Suite

ProActive in Cloud Stack

ProActive Parallel Suite





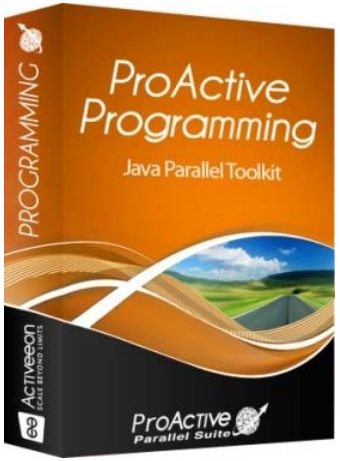
ProActive Parallel Suite

Conclusion

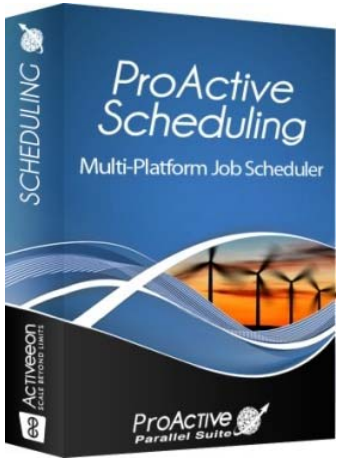
Conclusion

ProActive Parallel Suite

- **Portability:**
  Windows, Linux, Mac
- **Versatility:**
  Desktops, Grids, Clouds



Java Parallel Toolkit



Multi-Platform Job Scheduler

Resource Manager

Free Professional Open Source Software

OW2 Consortium

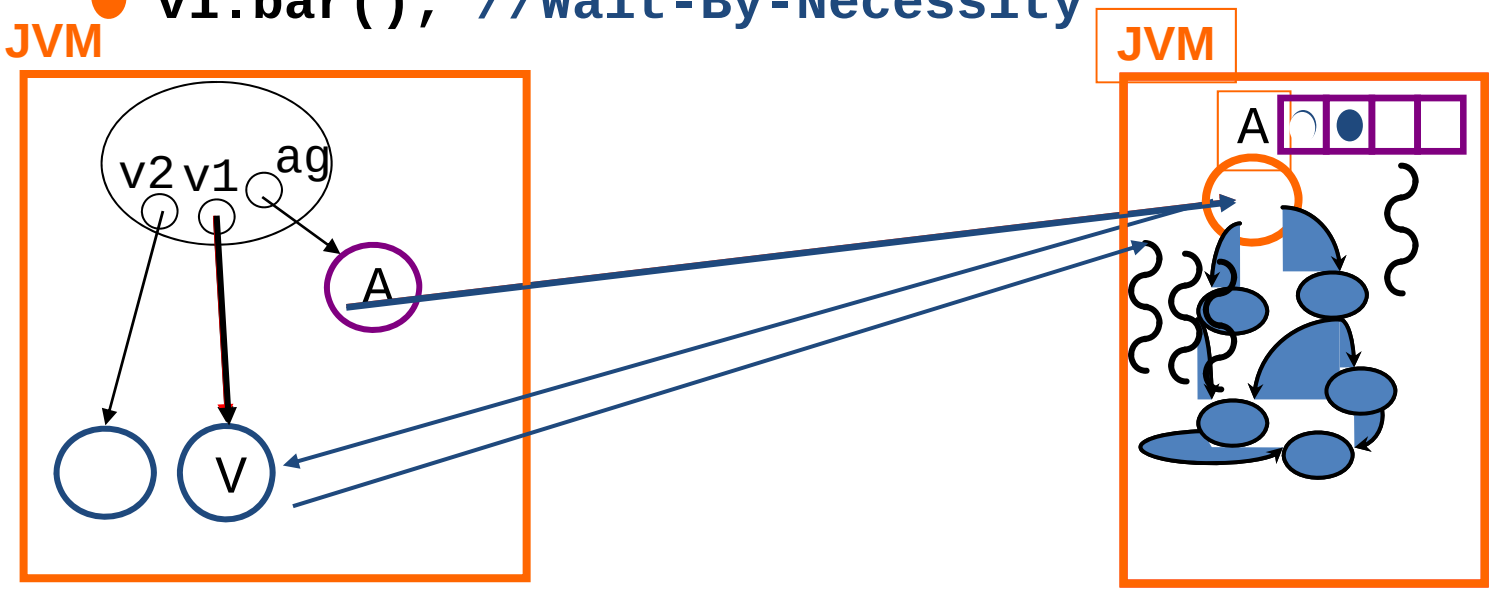
Manycores, Distribution and Cloud
Multi-Core: Ready for a strong evolution
Cloud: Smooth transition available (Desktop , Server, Cluster)

Activeeon SCALE BEYOND LIMITS

ProActive Parallel Suite

ProActive : Active objects

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



Java Object

Active Object

Req. Queue

Future Object

Proxy

Request

Thread

Wait-By-Necessity is a Dataflow Synchronization

ACTIVEEON SCALE BEYOND LIMITS

ProActive Parallel Suite

Use Case 1: Genomics

Resources set up

SOLID machine from Applied Biosystems

16 nodes

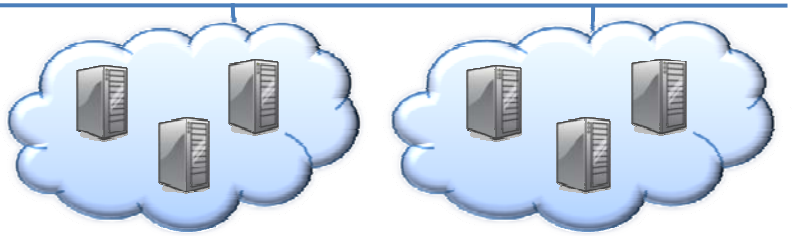
ProActive Parallel Suite

PBS

Cluster

Desktops

Nodes can be dynamically added!



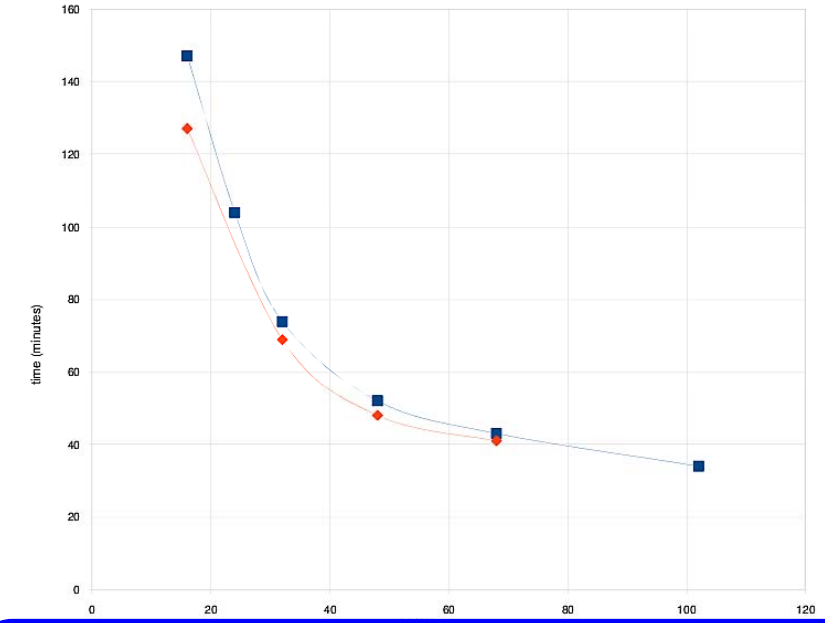
amazon web services™

EC2

Clouds

First Benchmarks

- The distributed version with ProActive of Mapreads has been tested on the INRIA cluster with two settings: the Reads file is split in either 30 or 10 slices
- Use Case: Matching 31 millions Sequences with the Human Genome (M=2, L=25)



4 Time FASTER from 20 to 100
Speed Up of 80 / Th.
Sequential : 50 h ➔ 35 mn

EC2 only test: nearly the same performances as the local SOLiD cluster (+10%)

For only $3,2/hour, EC2 has nearly the same perf. as the local SOLiD cluster (16 cores, for 2H30)

Benchmark: local vs. hybrid cloud

Use case: 3 runs performed in parallel containing a total of 28,5 millions of reads to be matched against the human genome

- SOLID nodes only
  Standard configuration using SOLID embedded nodes: 12

- Total computation time:
  12.5 hours

- SOLID and EC2 nodes
  12 SOLiD nodes
  12 EC2 machines
  (type: “mlarge”, 2 nodes each)

- Total computation time:
  8 hours

Gain: 4,5 hours (36% faster)
EC2 costs: $40

Benchmark: local vs. EC2 cloud

	Execution time (min)	Cost (US$)
Standard PBS config	300	NA
ProActive Amazon EC2	340	20 US$

For only $3,2/hour, the EC2 setup has nearly the same performances as the local SOLiD cluster

UC 2: Acceleration of Financial Valuations

A High Performance Solution

PRICING PARTNERS

ProActive Parallel Suite

- A Collaboration between Pricing Partners and ActiveEon

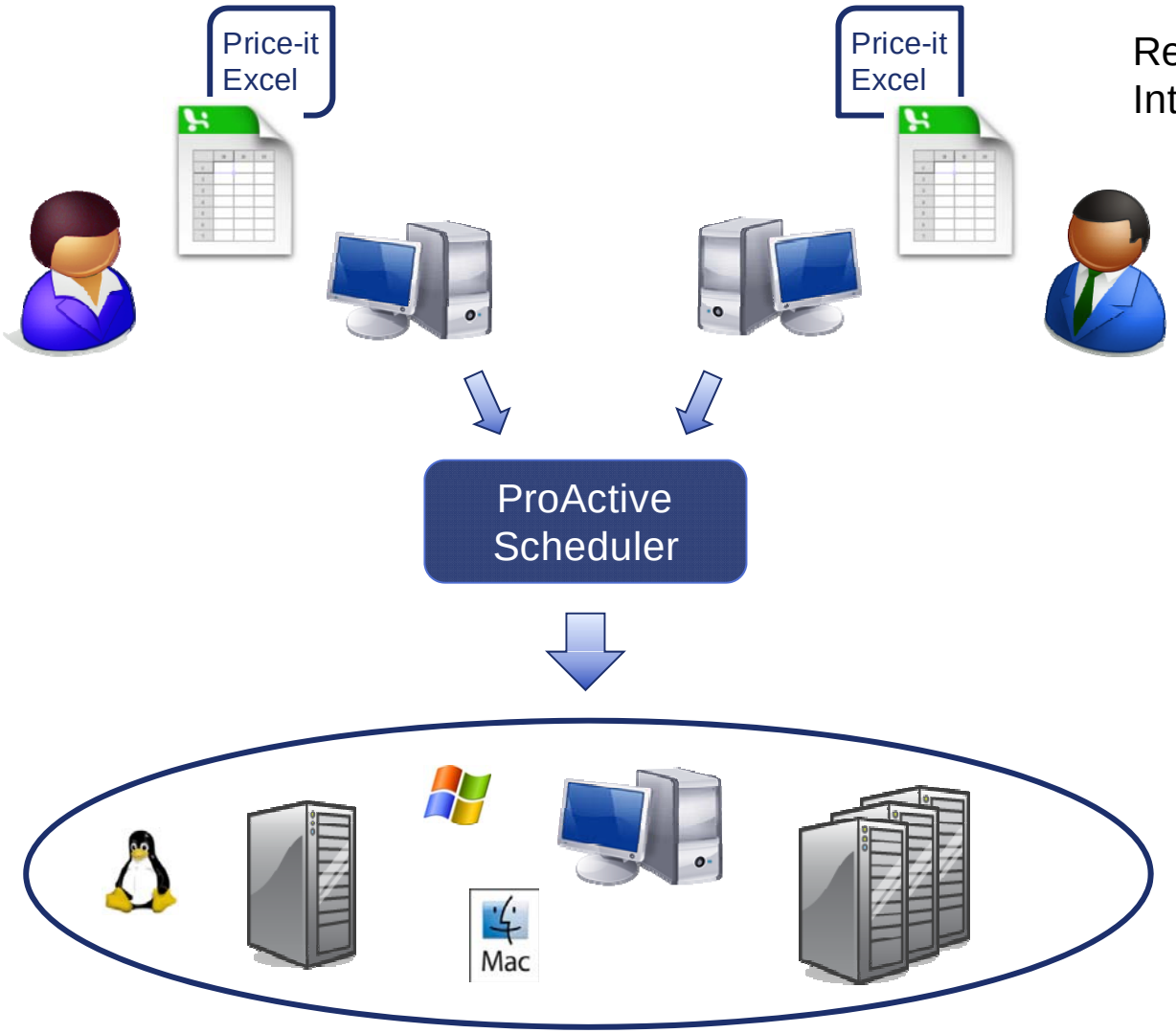
- Price-it® Excel Accelerated by *ProActive Parallel Suite®*
  - A Global Solution: fully integrated with the same functionalities and interface as Price-it Excel while increasing its computing power
  - High Quality Service: from both companies

Some Technical Facts

- Price-It®
  - C++ library developed by Pricing Partners
  - Pricing solution dedicated to highly complex financial derivatives

- Specification and Constraints
  - Accelerate Price-It® Excel product
    - Built on Price-It® library, this product integrates an interface with Excel for input data managem and results display
  - Focus on highly parallelizable Greek computation
  - Operating system: Windows

How Does it Work?
Price-it Computing Distribution



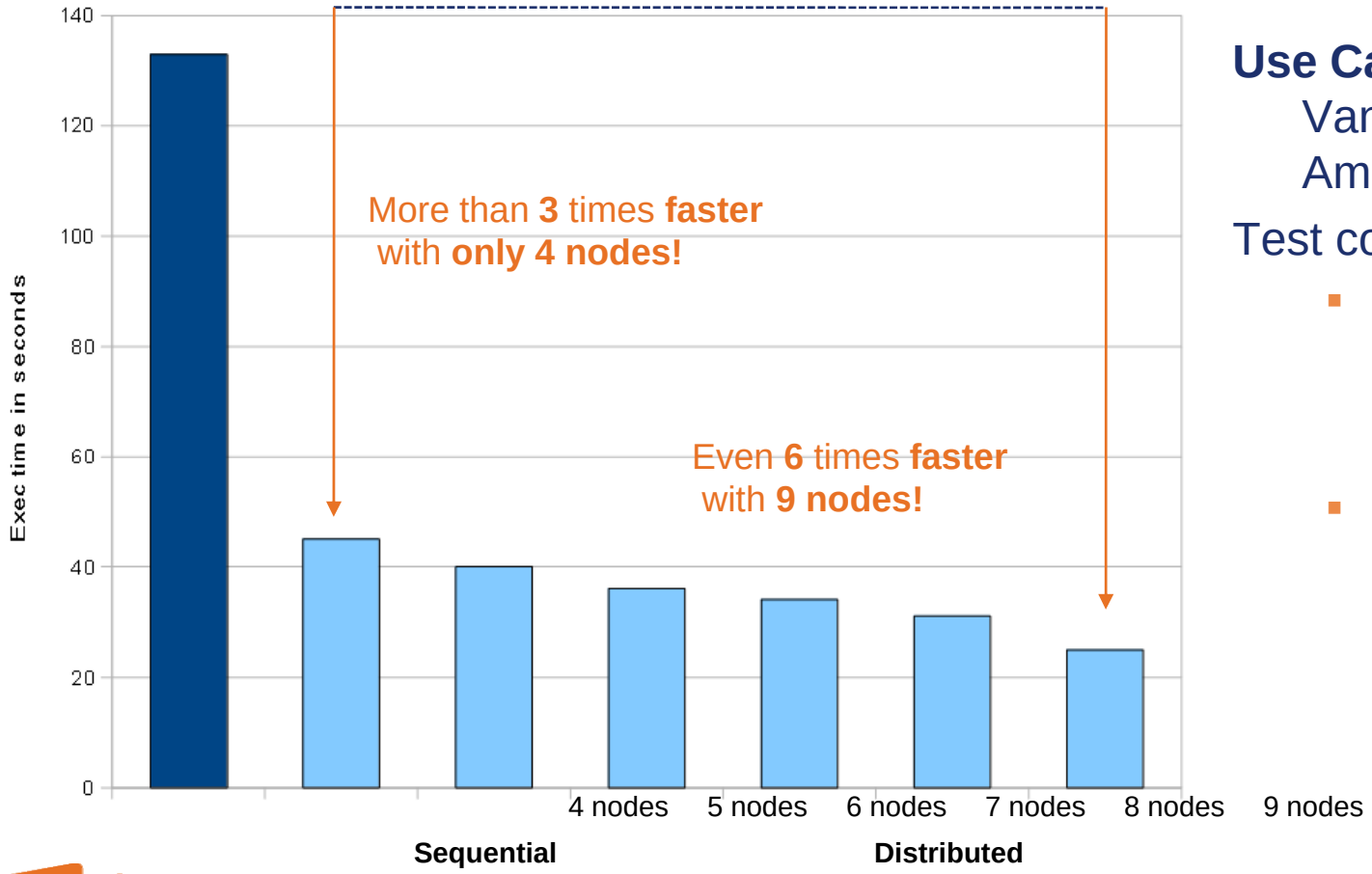
Regular Price-it Excel Interface

Automatic execution via job scheduler

Pool of shared resources

Accelerated Price-it Performances

- **Increased Productivity**: Reduces Price-it Execution Time by 6 or more!



Use Case: Bermuda Vanilla, Model American MC

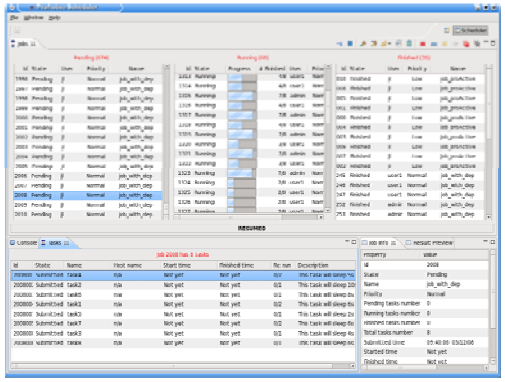
Test conditions:

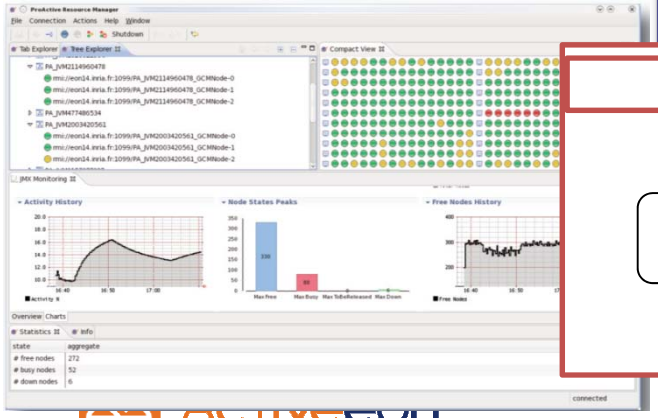
- One computation is split in 130 tasks that are distributed
- Each task uses 300ko

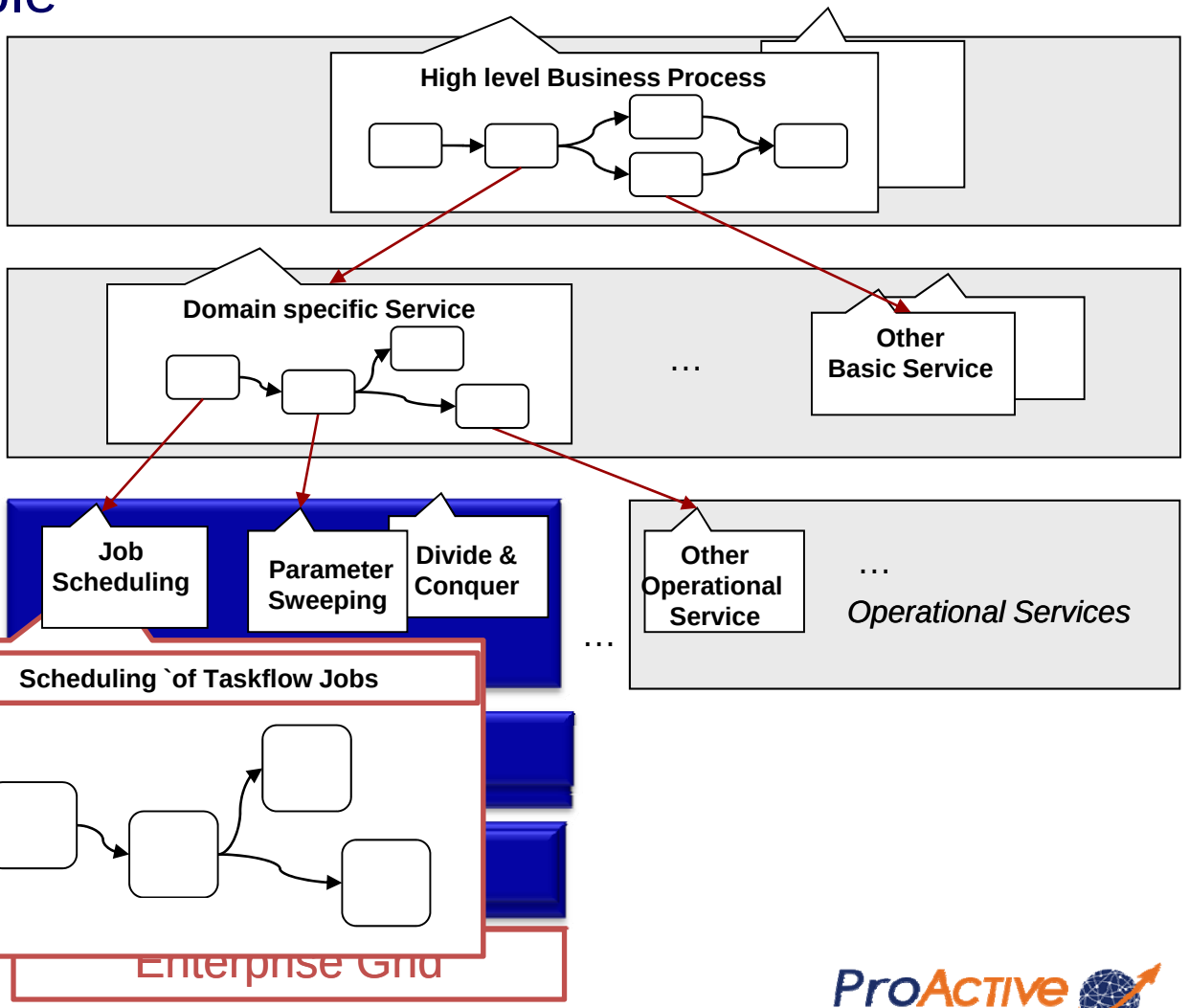
UC 3: SOA Analysis of Web Server Logs

Parallel Services

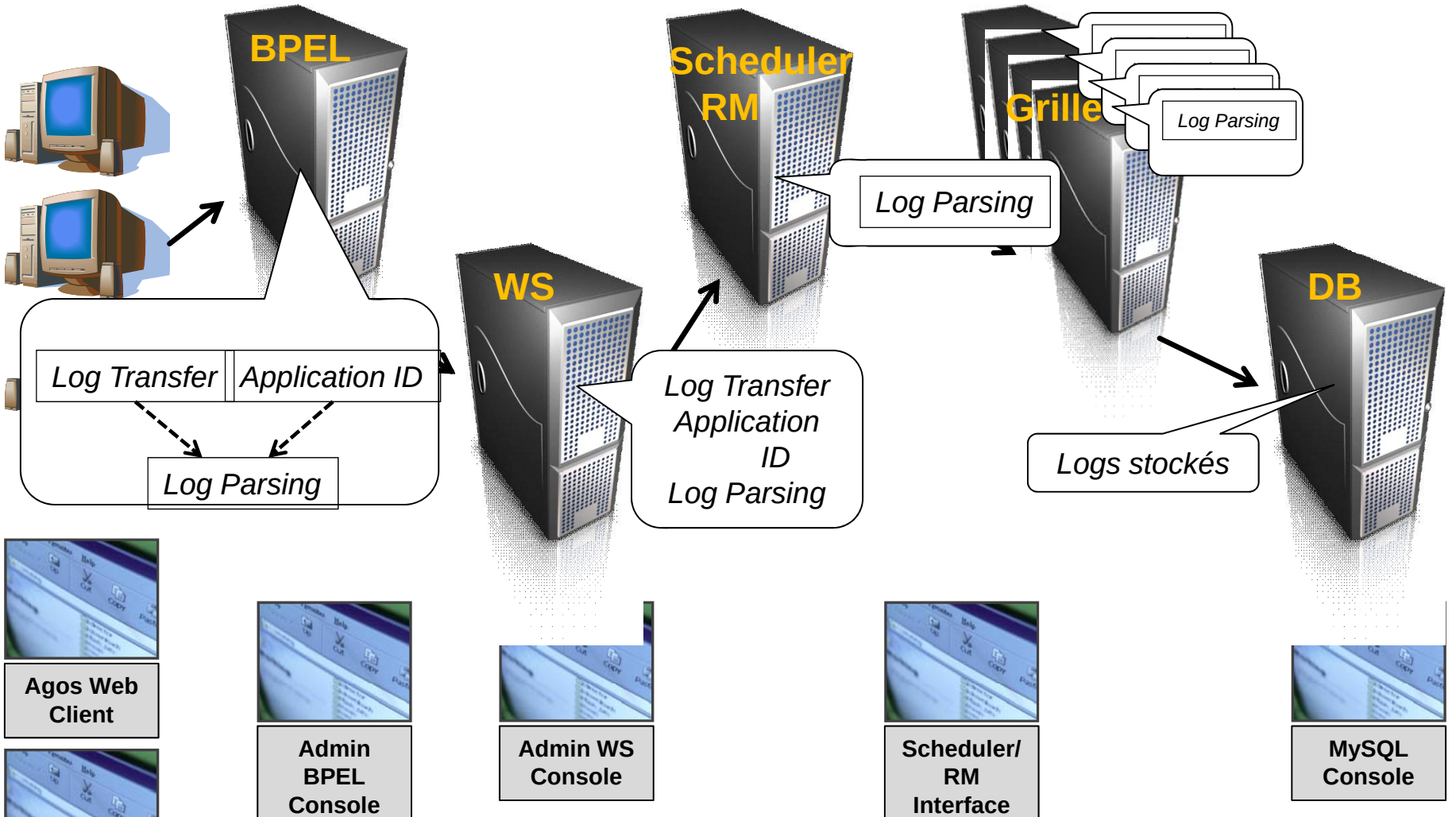
- Separation: BPEL – Parallel Serv. – Task Flow
- Standards et Portable
- Flexibility



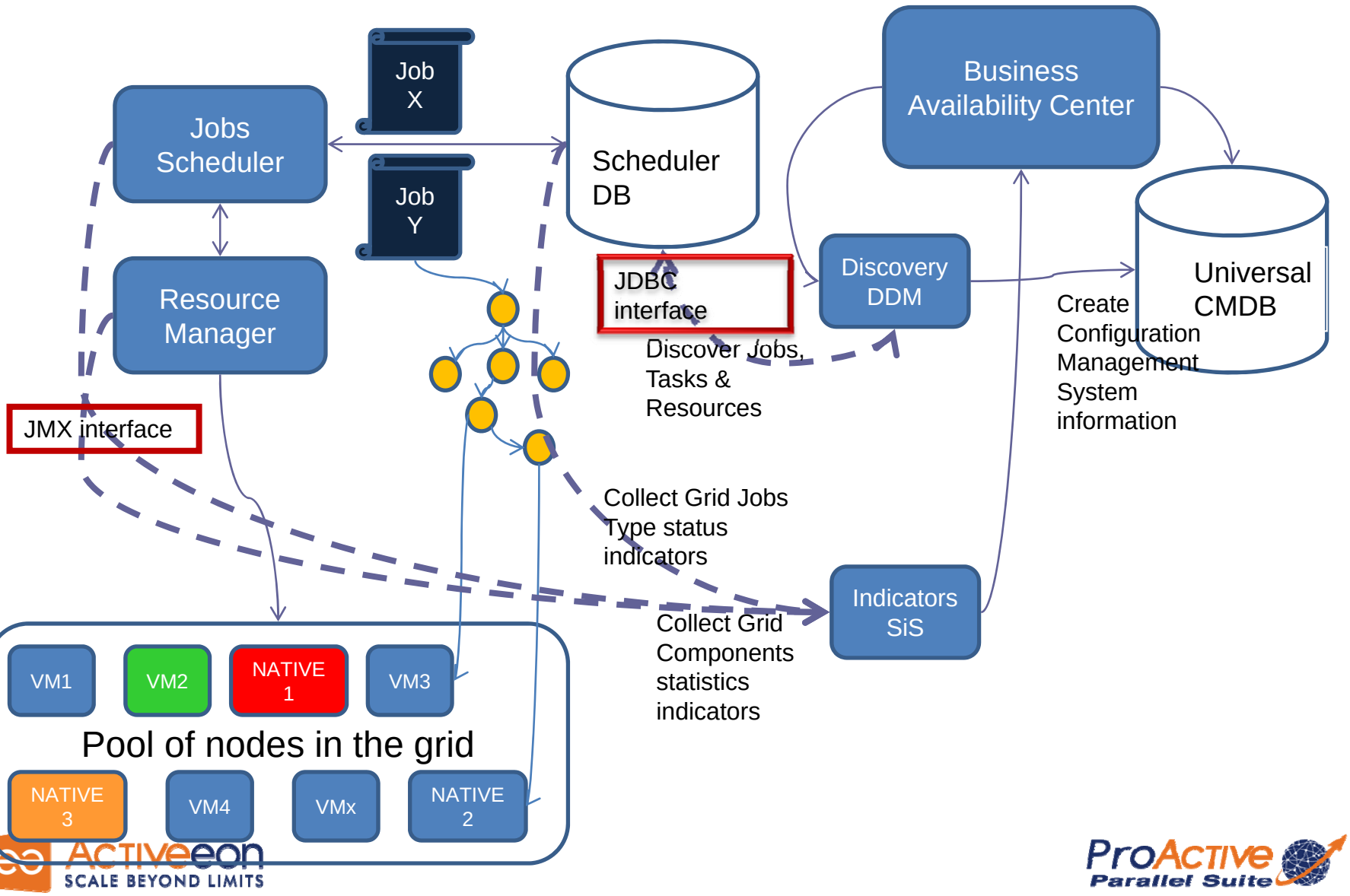




Cas d'études AMADEUS : Démonstration



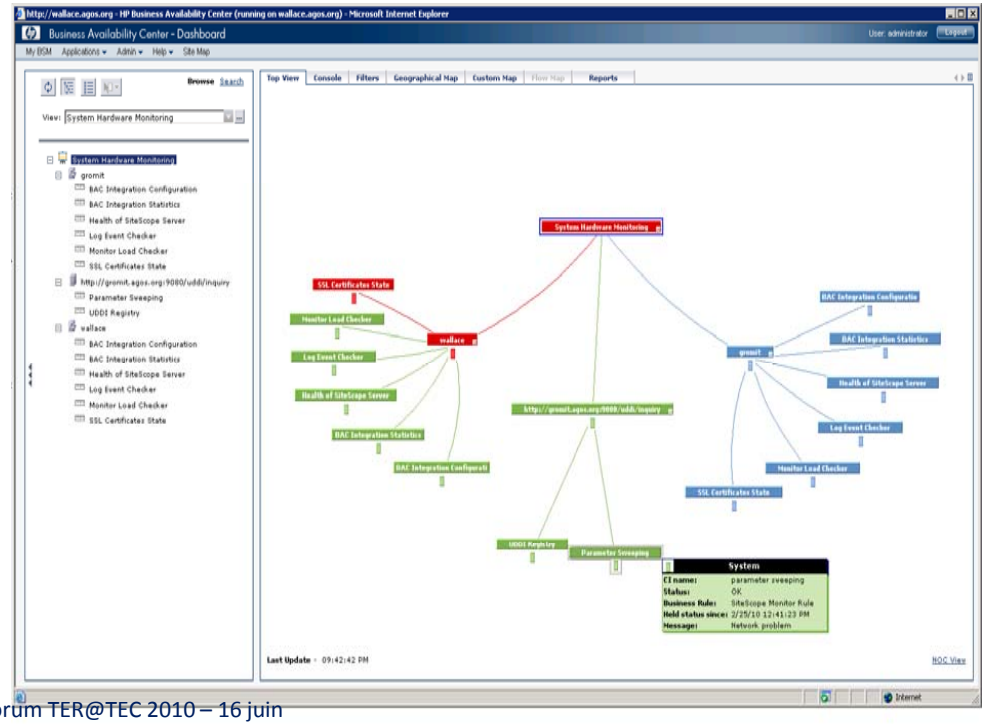
Grid Monitoring integration



AGOS Platform Management

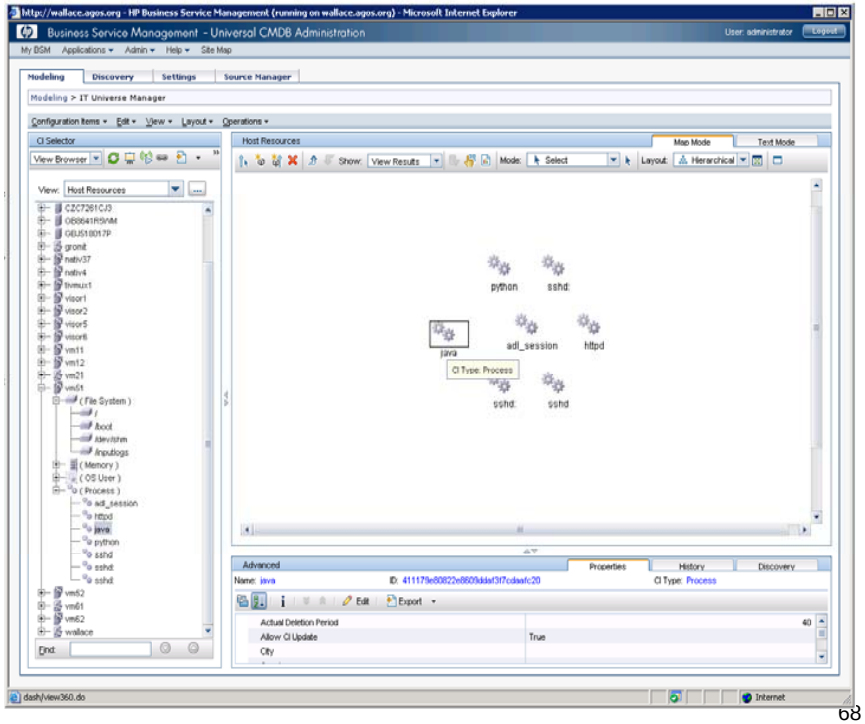
HP- Business Availability Center (HP-BAC)

- Monitoring of the entire platform
- Cover all layers in the scope
- Provide monitoring dashboard and reports



Tasks scheduler & Resources manager

- Integration with grid components
- Grid insights through indicator collection and running jobs on



AGOS: Grid Architecture for SOA

Building a Platform for Agile SOA with Grid

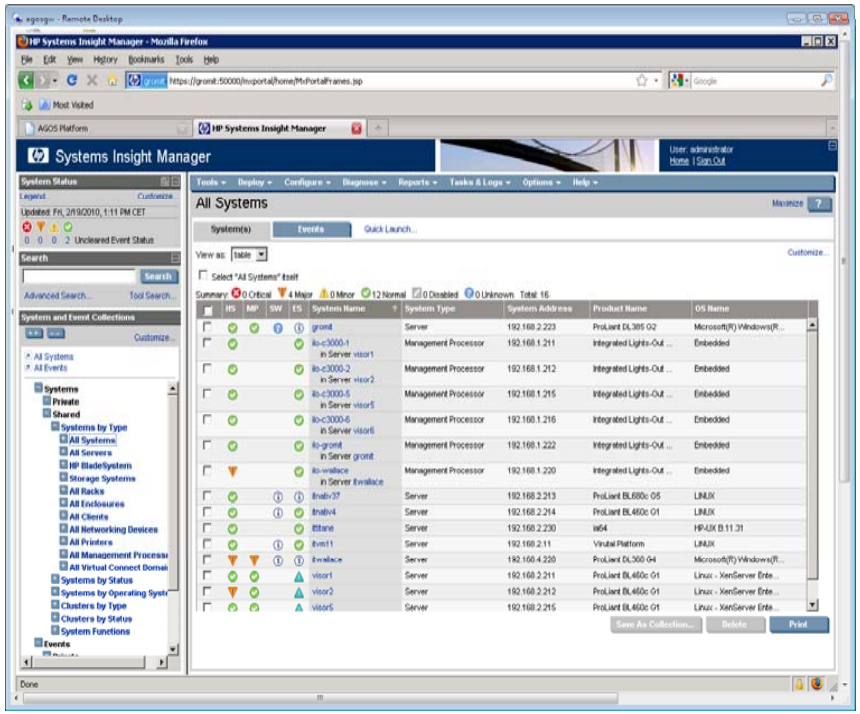
- **AGOS Solutions**

In Open Source with Professional Support

AGOS Infrastructure Management

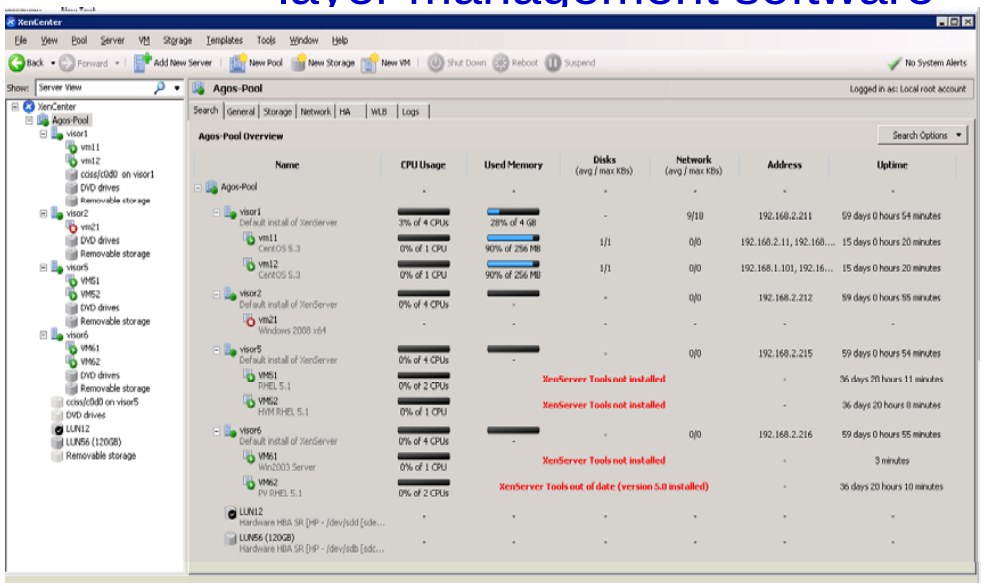
HP Systems Insight Manager (HP-SIM)

- Monitoring of entire infrastructure
- Communicates with upper layer management software (HP BAC)



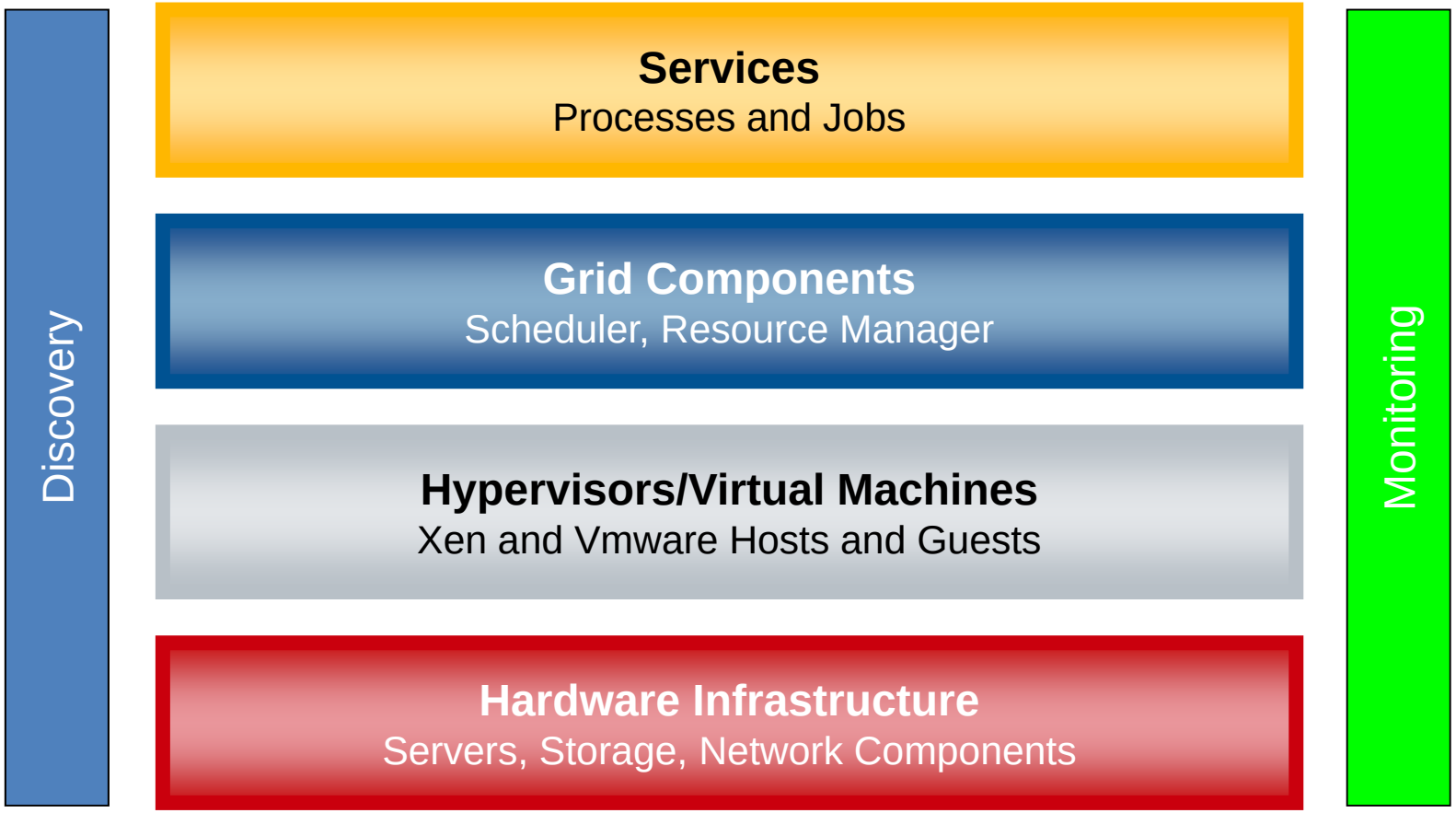
Citrix XenCenter

- Hypervisor and VM management
- Communicates with upper layer management software



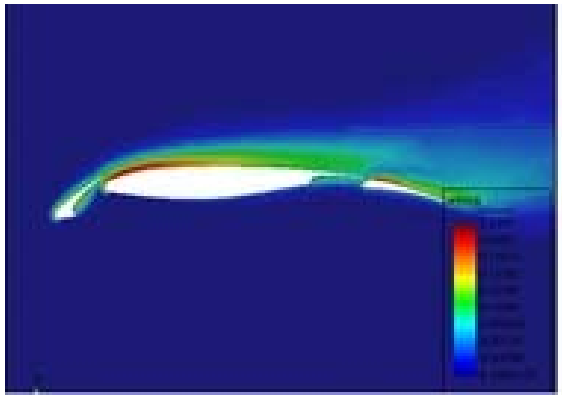
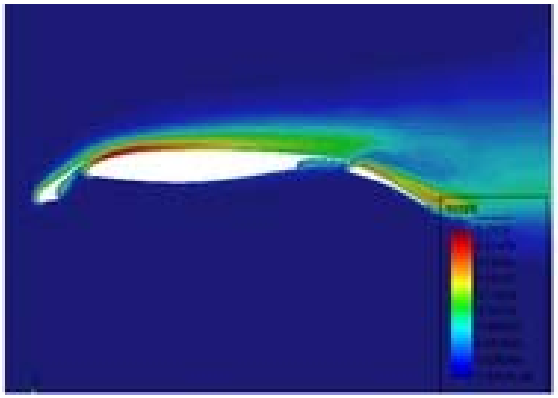
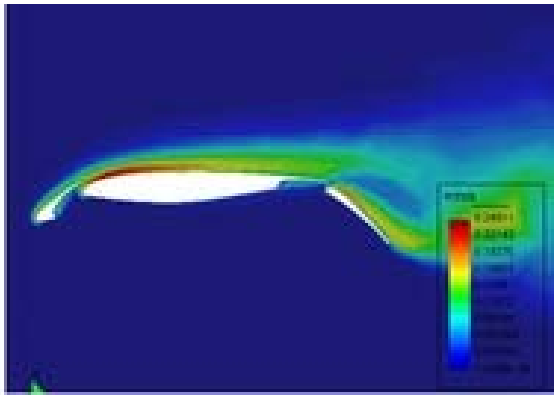
ProActive Parallel Suite

AGOS and HP Management tools Integration

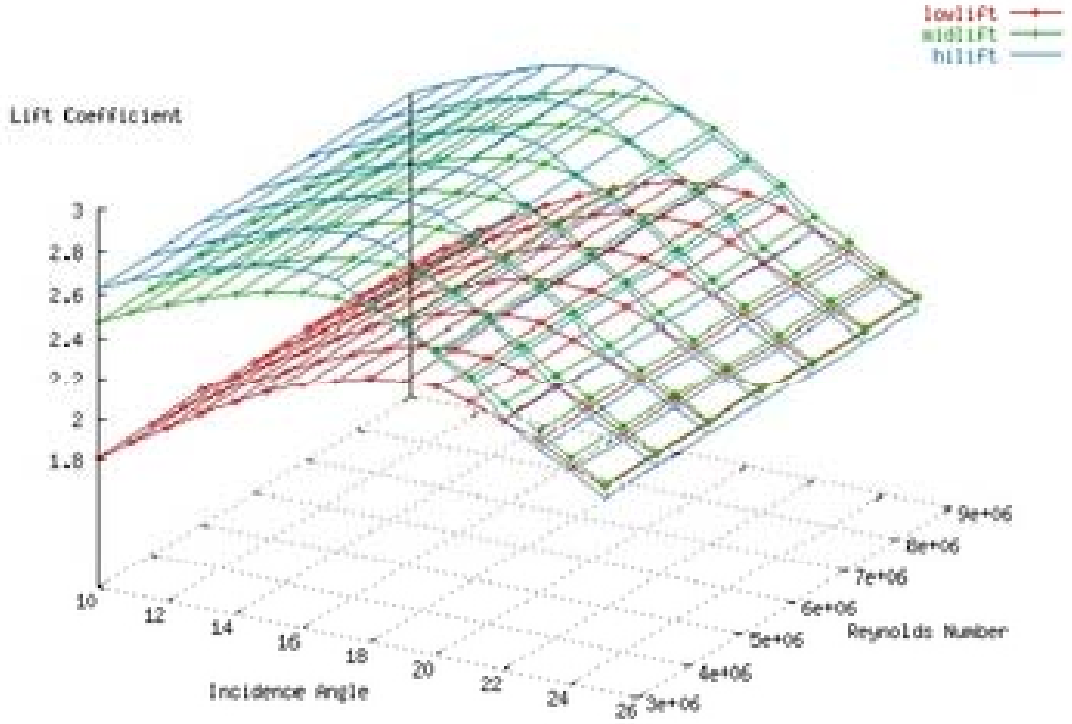


4. Use Case: Wings Optimization

Wing performance (GridSystems, Spain)

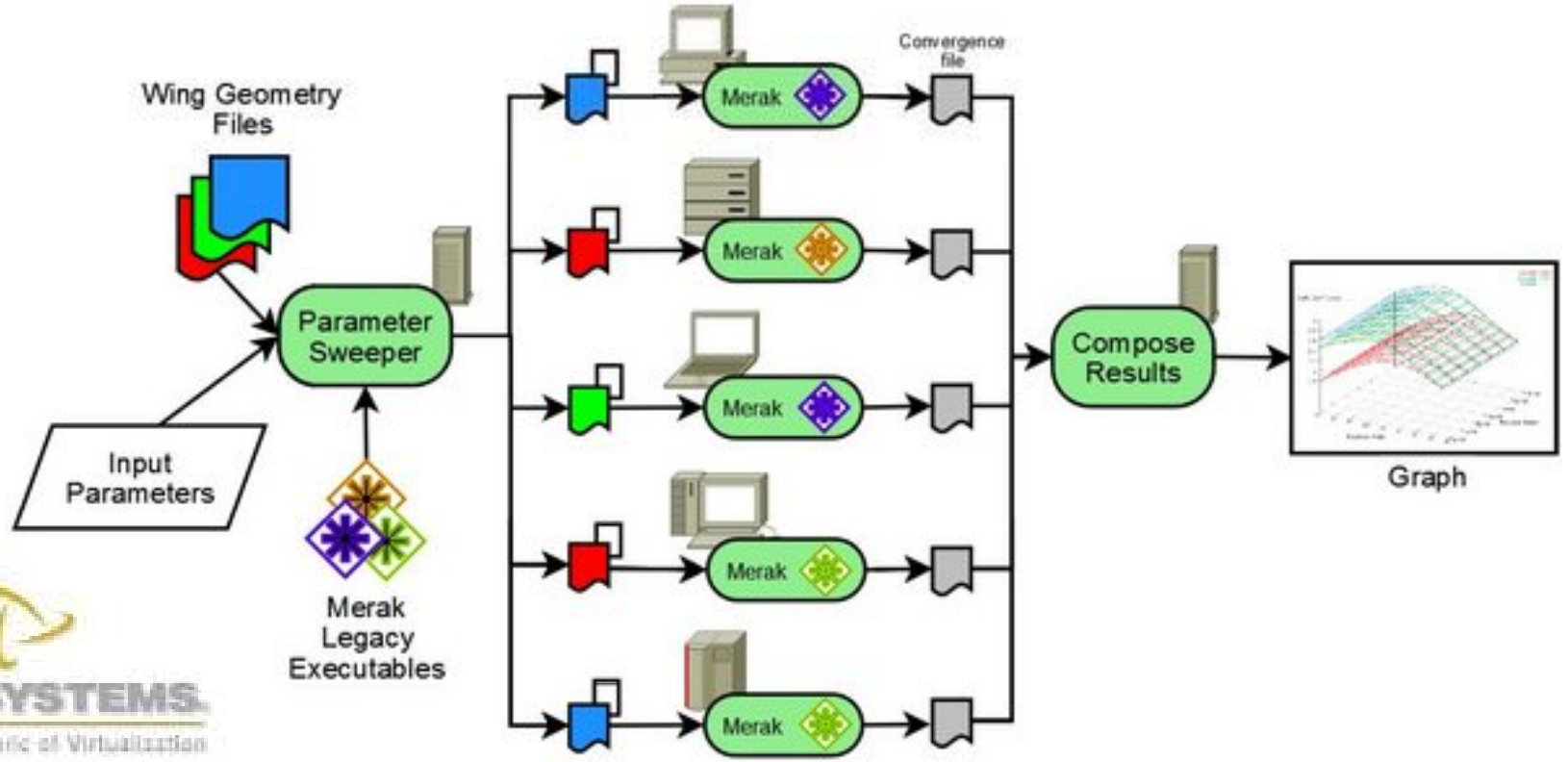
GRIDSYSTEMS
The Open Fabric of Virtualization

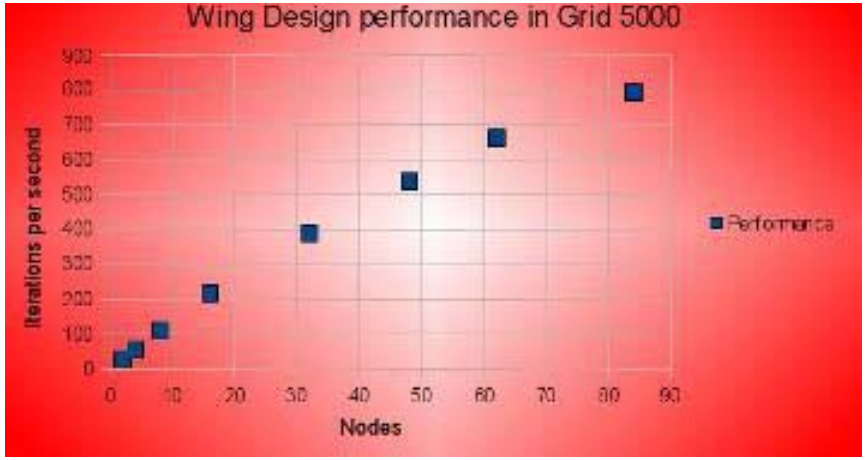


ACTIVEEON
SCALE BEYOND LIMITS

OW2 Consortium

Wing performance (GridSystems, Spain)





(2) ASP: Asynchronous Sequential Processes

$$\frac{(a,\sigma) \rightarrow_S (a',\sigma')}{\alpha[a;\sigma;\iota;F;R;f] \parallel P \longrightarrow \alpha[a';\sigma';\iota;F;R;f] \parallel P}\ (\textsc{local})$$

Local

$$\frac{\begin{array}{c}\gamma \text{ fresh activity} \qquad \iota' \notin dom(\sigma) \qquad \sigma' = \{\iota' \mapsto AO(\gamma)\} :: \sigma \\ \sigma_\gamma = copy(\iota'',\sigma) \qquad Service = (\text{ if } m_j = \emptyset \text{ then } FifoService \text{ else } \iota''.m_j())\end{array}}{\begin{array}{c}\alpha[\mathcal{R}[Active(\iota'',m_j)];\sigma;\iota;F;R;f] \parallel P \\ \longrightarrow \alpha[\mathcal{R}[\iota'];\sigma';\iota;F;R;f] \parallel \gamma[Service;\sigma_\gamma;\iota'';\emptyset;\emptyset;\emptyset] \parallel P\end{array}}\ (\textsc{newact})$$

Creating an Activity

$$\frac{\begin{array}{c}\sigma_\alpha(\iota) = AO(\beta) \qquad \iota'' \notin dom(\sigma_\beta) \qquad f_i^{\alpha\rightarrow\beta} \text{ new future} \qquad \iota_f \notin dom(\sigma_\alpha) \\ \sigma'_\beta = Copy\&Merge(\sigma_\alpha,\iota' \ ; \ \sigma_\beta,\iota'') \qquad \sigma'_\alpha = \{\iota_f \mapsto fut(f_i^{\alpha\rightarrow\beta})\} :: \sigma_\alpha\end{array}}{\begin{array}{l}\alpha[\mathcal{R}[\iota.m_j(\iota')];\sigma_\alpha;\iota_\alpha;F_\alpha;R_\alpha;f_\alpha] \parallel \beta[a_\beta;\sigma_\beta;\iota_\beta;F_\beta;R_\beta;f_\beta] \parallel P \longrightarrow \\ \alpha[\mathcal{R}[\iota_f];\sigma'_\alpha;\iota_\alpha;F_\alpha;R_\alpha;f_\alpha] \parallel \beta[a_\beta;\sigma'_\beta;\iota_\beta;F_\beta;R_\beta :: [m_j;\iota'';f_i^{\alpha\rightarrow\beta}];f_\beta] \parallel P\end{array}}\ (\textsc{request})$$

Sending a Request

$$\frac{R = R' :: [m_j;\iota_r;f'] :: R'' \qquad m_j \in M \qquad \forall m \in M,\ m \notin R'}{\alpha[\mathcal{R}[Serve(M)];\sigma;\iota;F;R;f] \parallel P \longrightarrow \alpha[\iota.m_j(\iota_r) \Uparrow f,\mathcal{R}[[]];\sigma;\iota;F;R' :: R'';f'] \parallel P}\ (\textsc{serve})$$

$$\frac{\iota' \notin dom(\sigma) \qquad F' = F :: \{f \mapsto \iota'\} \qquad \sigma' = Copy\&Merge(\sigma,\iota \ ; \ \sigma,\iota')}{\alpha[\iota \Uparrow (f',a);\sigma;\iota;F;R;f] \parallel P \longrightarrow \alpha[a;\sigma';\iota;F';R;f'] \parallel P}\ (\textsc{endservice})$$

Service

$$\frac{\sigma_\alpha(\iota) = fut(f_i^{\gamma\rightarrow\beta}) \qquad F_\beta(f_i^{\gamma\rightarrow\beta}) = \iota_f \qquad \sigma'_\alpha = Copy\&Merge(\sigma_\beta,\iota_f \ ; \ \sigma_\alpha,\iota)}{\begin{array}{l}\alpha[a_\alpha;\sigma_\alpha;\iota_\alpha;F_\alpha;R_\alpha;f_\alpha] \parallel \beta[a_\beta;\sigma_\beta;\iota_\beta;F_\beta;R_\beta;f_\beta] \parallel P \longrightarrow \\ \alpha[a_\alpha;\sigma'_\alpha;\iota_\alpha;F_\alpha;R_\alpha;f_\alpha] \parallel \beta[a_\beta;\sigma_\beta;\iota_\beta;F_\beta;R_\beta;f_\beta] \parallel P\end{array}}\ (\textsc{reply})$$

Sending a Reply

Key Point

- **“MPI and programming languages from the 60’s will not make it”**
- **Jack Dongarra,** 2/13/2009,
- Wake Forest University talk

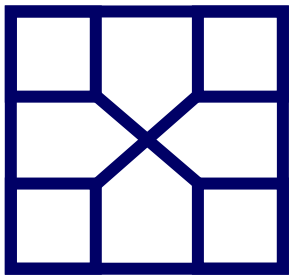
NAS Parallel Benchmarks

- Experimented on 3D ElectroMagnetism, and Nasa Benchmarks

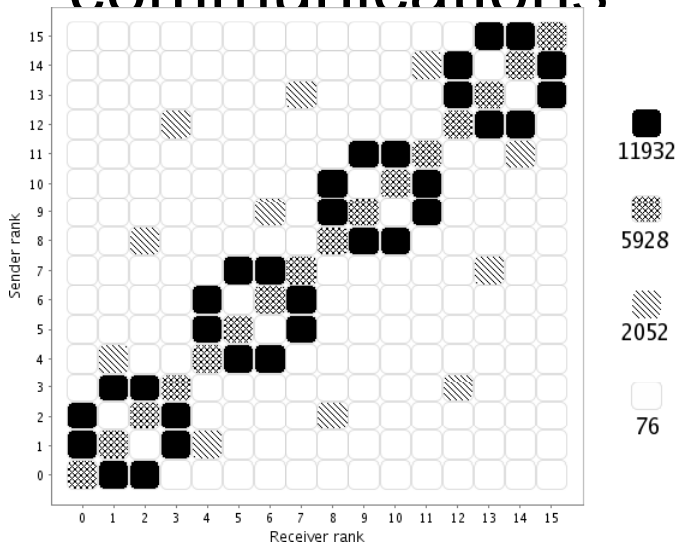
- **Designed by NASA to evaluate benefits of high performance systems**
- **Strongly based on CFD**
- **5 benchmarks (kernels) to test different aspects of a system**
- **2 categories or focus variations:**
  - **communication intensive and computation intensive**

Communication Intensive
CG Kernel (Conjugate Gradient)

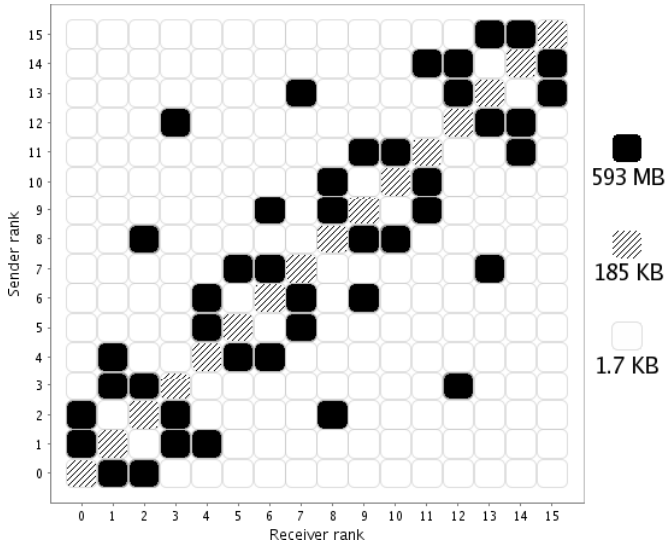
- Floating point operations
- Eigen value computation
- High number of unstructured communications



- 12000 calls/node
- 570 MB sent/node
- 1 min 32
- 65 % comms/WT

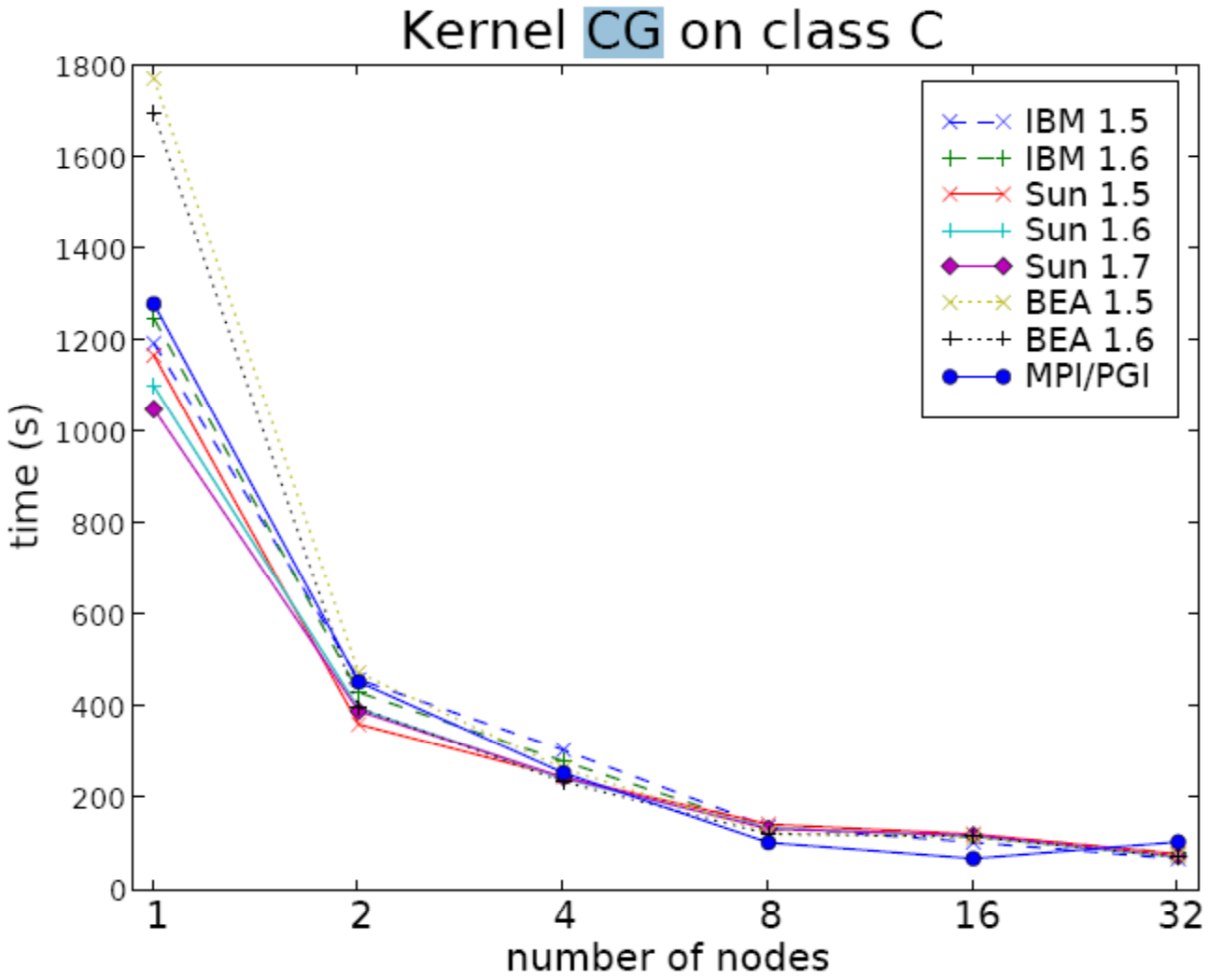


Message density distribution



Data density distribution

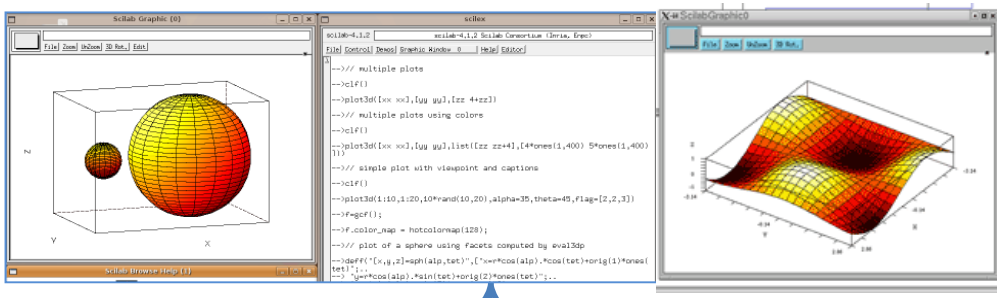
Communication Intensive
CG Kernel (Conjugate Gradient)



← Comparable Performances

3.
Parallel Scilab Simulations

Seamless Parallel & Distributed Scilab



Scilab

Static Policy

LSF

Timing Policy 12/24

Desktops

Dynamic Workload Policy

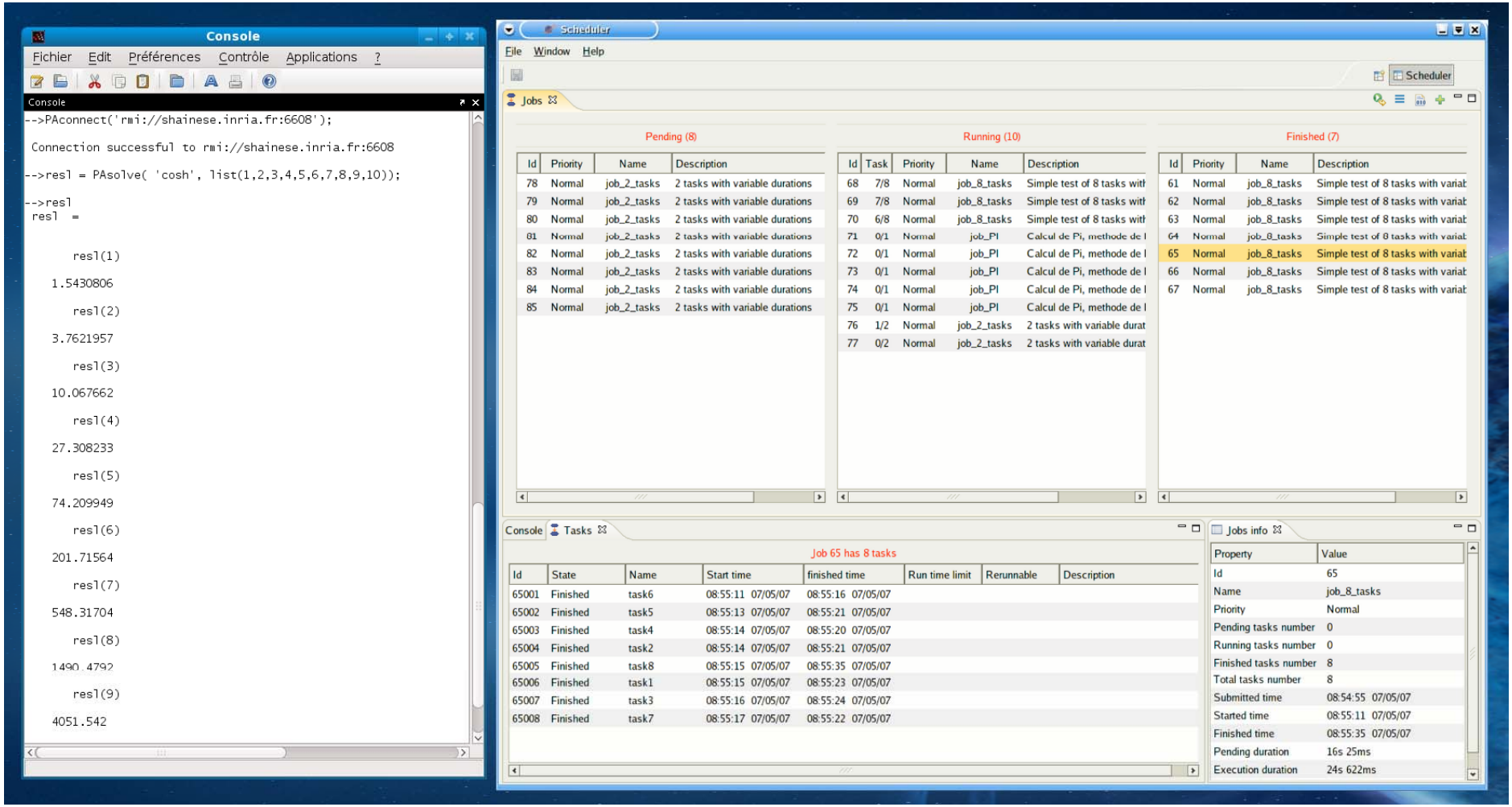
EC2

Dedicated resources

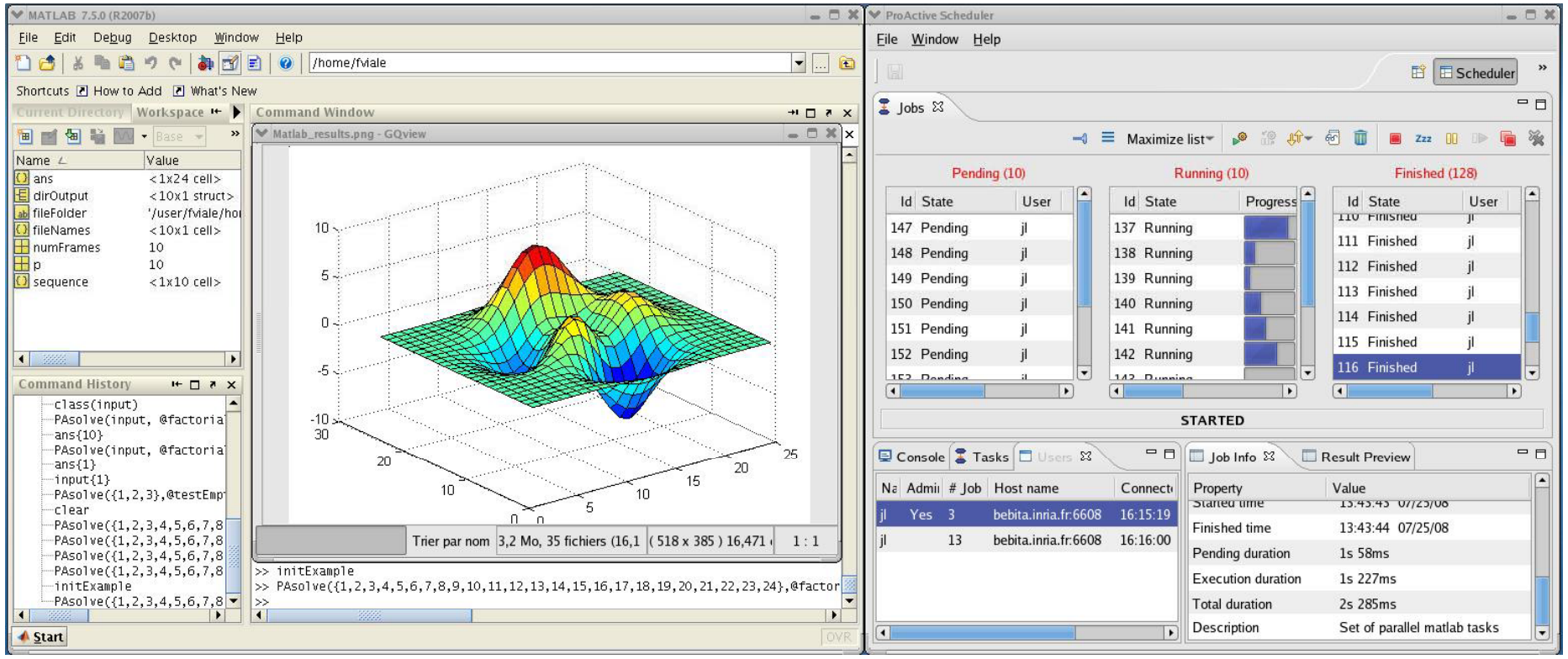
Desktops

Amazon EC2

Interface ProActive ⇔ Scilab



Interface ProActive ⇔ Matlab



Interface ProActive ⇔ Scilab / Matlab

- Primitives de haut niveau Scilab/Matlab pour distribuer les calculs :
  - f = PAeval(@foo,p1)
    // evalue f(p1) en asynchrone
    // f est un futur
  - L = PAsolve(@foo, {p1,p2, … pk})
    // evalue foo(p1), foo(p2) .. foo(pk) en asynchrone
    // L = f1,…fk est une liste de futurs

Synchronization des calculs

- Primitives d'attente :
  - r = PAwaitForAny(L)
    // bloque et attend jusqu'a l'obtention du
    // premier des fk et retourne le resultat r
  - lr = PAgetAny(L)
    // non-blocant, retourne une liste des resultats
    // deja obtenus

4. Use Case: OMD2

OMD2

Interfaces Open Sources
pour faciliter les
Optimisations Multi-Disciplinaires Distribuées

OMD2 OPTIMISATION MULTI DISCIPLINAIRE DISTRIBUÉE

OMD2 :

Interfaces Open Sources pour faciliter les Optimisations Multi-Disciplinaires Distribuées

OMD2 OPTIMISATION MULTI DISCIPLINAIRE DISTRIBUÉE

RENAULT

Laboratoire Roberval
Unité de recherche en mécanique

CENTRALE PARIS

Ecole Nationale Supérieure des Mines SAINT-ETIENNE

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

INRIA

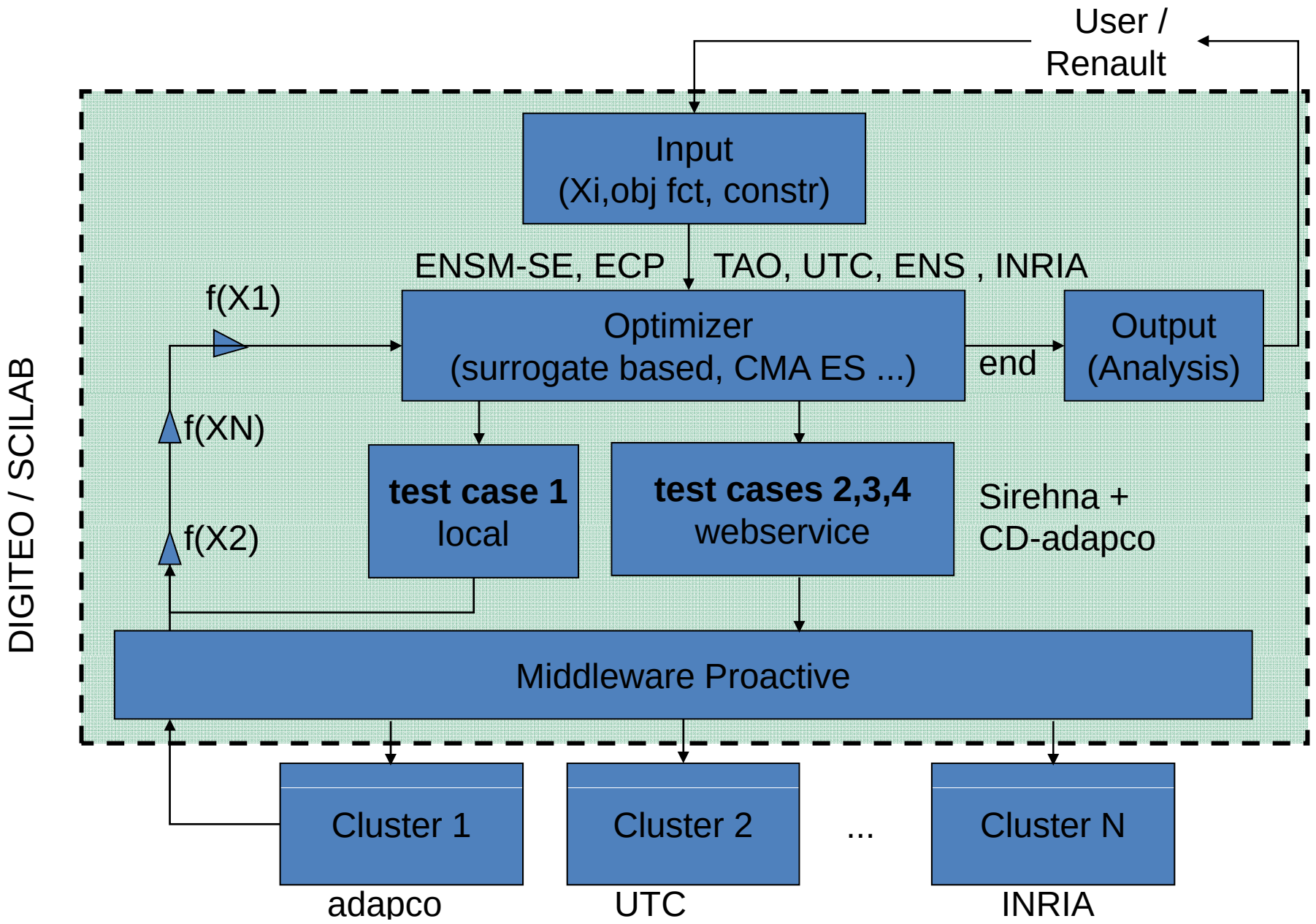
CD-adapco

digiteo
Research in information sciences and technologies

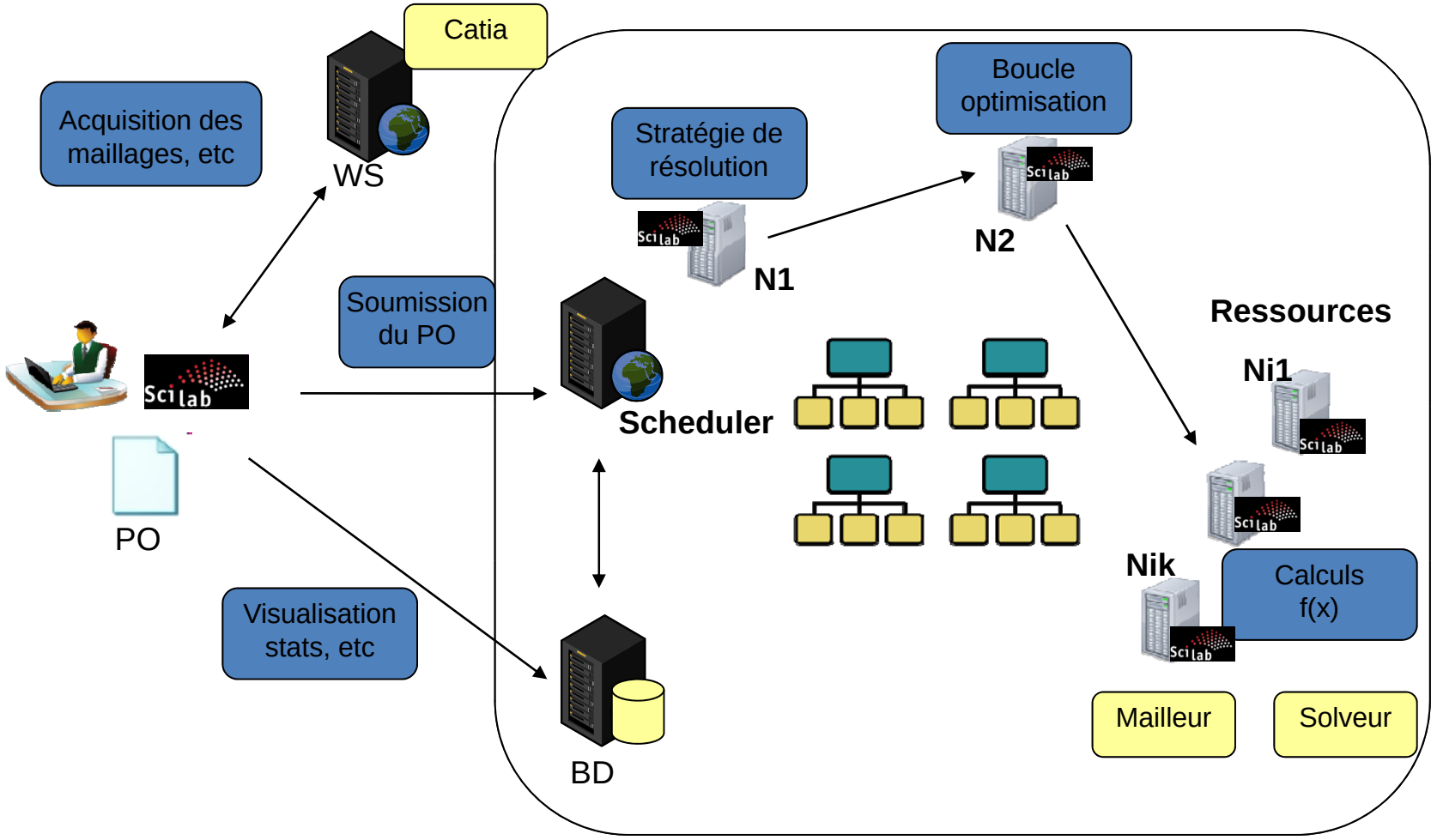
SIREHNA
a DCNS company

Activeeon
SCALE BEYOND LIMITS

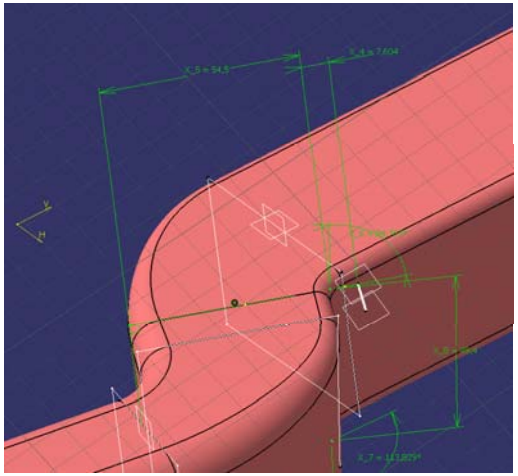
Plateforme OMD2, vue générale :



Distribution des calculs

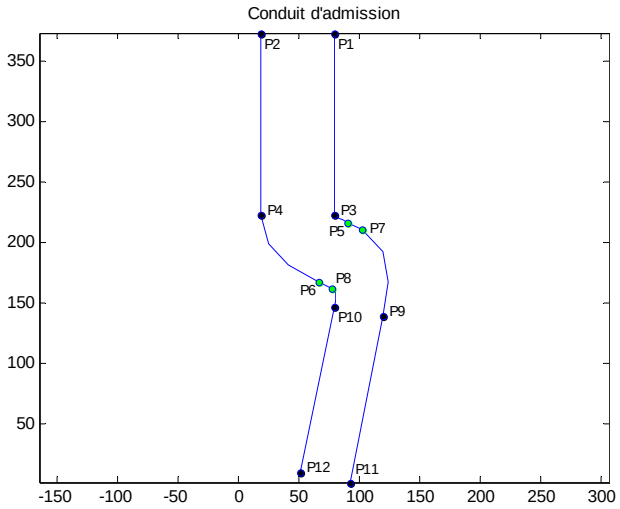


Les cas tests



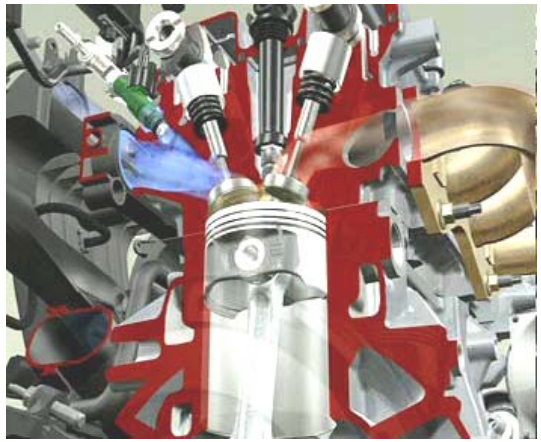
10min CPU

Conduit de climatisation 3D



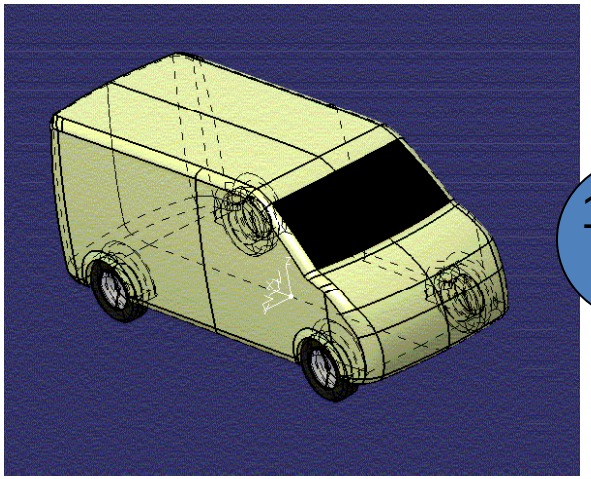
<1min CPU

Conduit de climatisation 2D



100h CPU

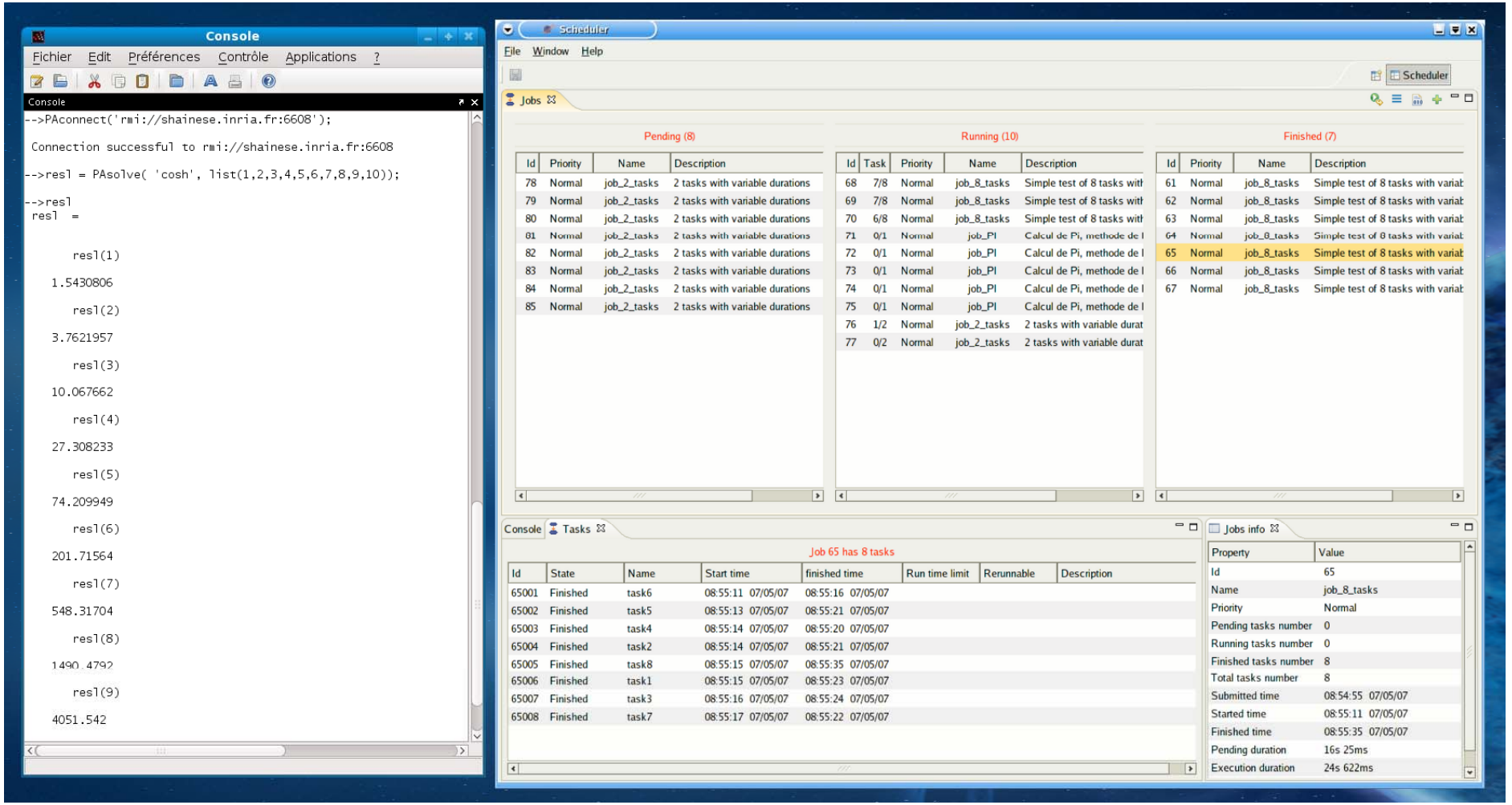
Culasse multi-disciplinaire



1000h CPU

Aéro externe

Interface ProActive ⇔ Scilab



Exemple d'utilisation : Boucle d'Optimisation

```
futureList = {};
PAconnect('//machine where the scheduler is running')
do {
        if size(futureList) < MAX_SENT {
                x=opt.ask(); // asks a new vector
                future = PAeval (@f, {x}); // asynchronously send f(x) to grid
                futureList = append(futureList, future); }
        else  {
                resultList = PAgetAny(futureList);   // Get Available results
                If resultList != void { // Updates the optimizer with new results
                        for i=1:size(resultList) {
                                [y,x] = resultList(i);
                                opt.tell(y,x);
                        }
                } else Sleep (t)
        }
}
while ~(opt.stop())
```