Bridging
Multi-Core and Distributed Computing:
all the way up to the Clouds

D. Caromel, et al.

Agenda

1. **Background: OASIS, ActiveEon**
2. **ProActive Overview :**
   Programming, Scheduling, Resourcing
3. **Use Case: Genomics**
4. **Cloud Seeding**

Parallelism+Distribution
with Strong Model: Speed & Safety

Key Objectives

- Parallel Programming Model and Tools
  - Badly needed
  - for the masses
  - for new architectures: Multi-Cores & Clouds
- As Effective as possible:
  - Efficient
  - However <u>Programmer Productivity</u> is first KSF
- For both Multi-cores and Distributed
  - Actually the way around
- Handling of ``Large-scale'': up to 4 000 so far

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

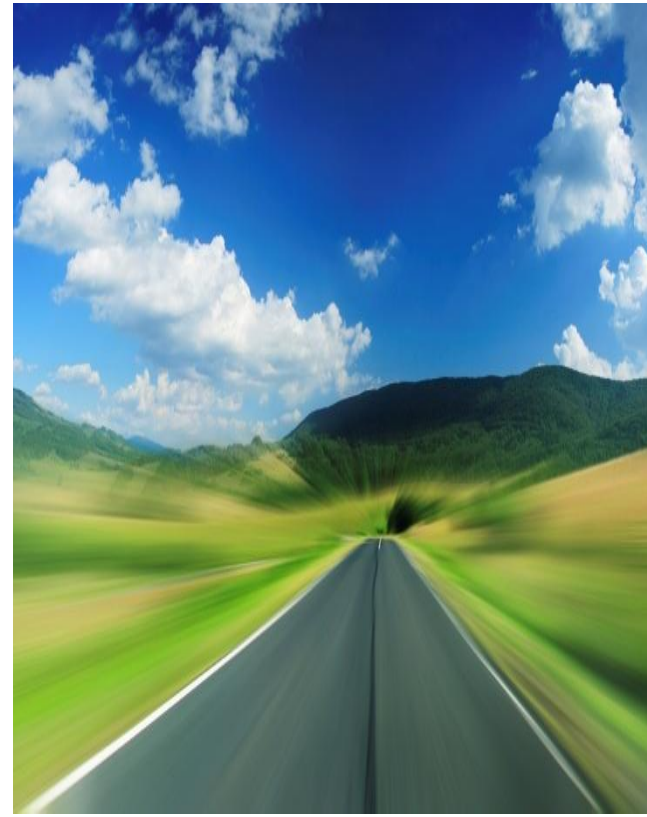
ActiveEon SCALE BEYOND LIMITS
ProActive Parallel Suite
OW2 Consortium

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/Régio
  - Viet Dung Doan (FP6 Bionets)
  - Virginie Contes (SOA4ALL)
  - Guilherme Pezzi (AGOS, CIFR
- + Visitors + Interns

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

Startup Company Born of INRIA



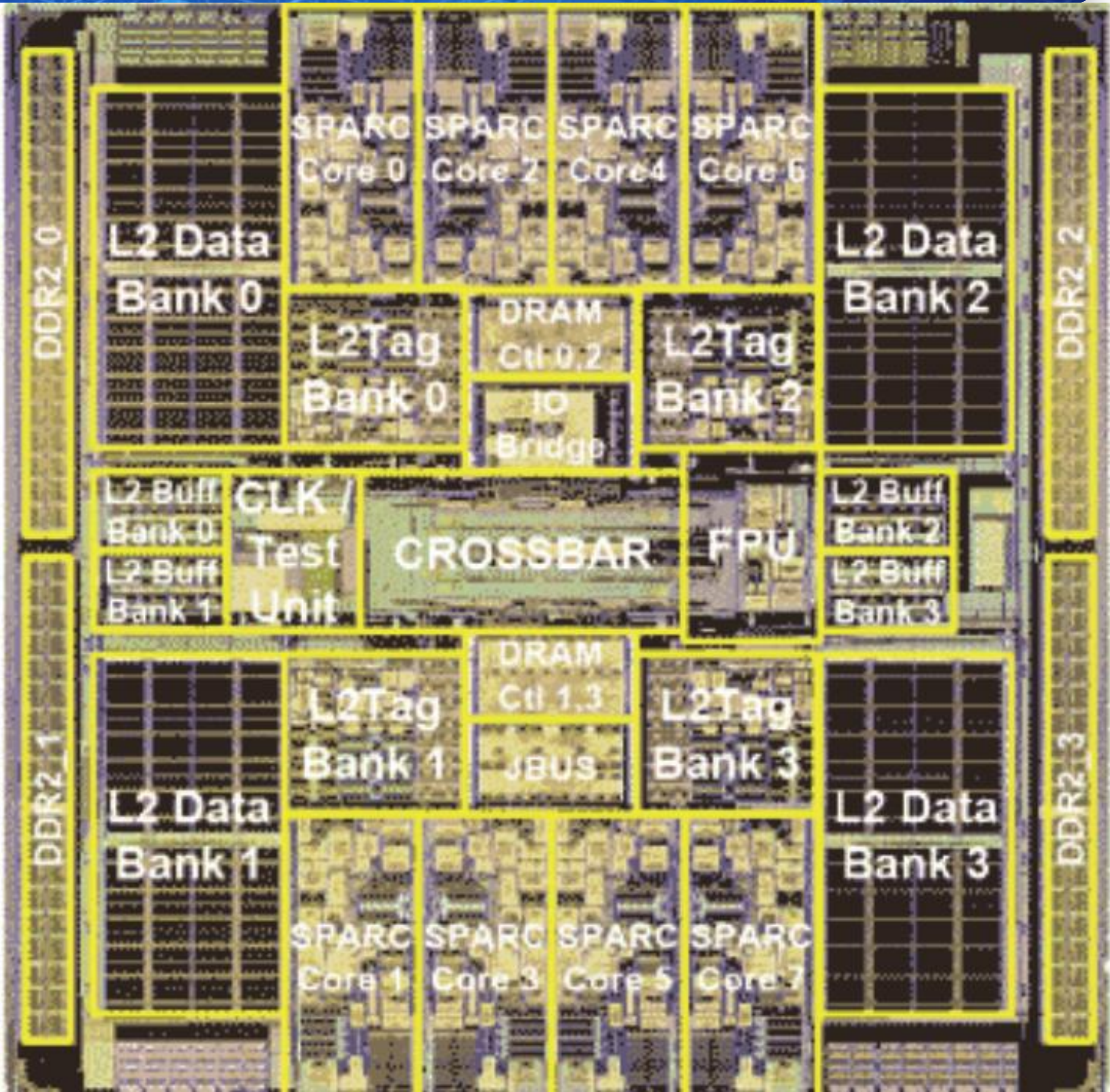
Some Partners:

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

Multi-Cores

Symetrical Multi-Core: 8-ways Niagara II

- 8 cores
- 4 Native threads per core
- Linux see 32 cores!



Multi-Cores
A Few Key Points

- Not Shared Memory (NUMA)
- Moore's Law rephrased:

  Nb. of Cores double every 18 to 24 months
- Key expected Milestones: Cores per Chips (OTS)
  - 2010: 32 to 64
  - 2012: 64 to 128
  - 2014: 128 to 256

  1 Million Cores Parallel Machines in 2012

  100 M cores coming in 2020
- Multi-Cores are NUMA, and turning Heterogeneous (GPU)

  They are turning into SoC with NoC: NOT SMP!

2. Overview

ProActive Parallel Suite

ProActive Parallel Suite

Parallel Acceleration Toolkit in Java:

- Java Parallel Programming + Legacy-Code + Wrapping and Control
- Taskflow Scheduling
- Resource Manager

Multi-Core + Distributed

Open Source Used in production by industry

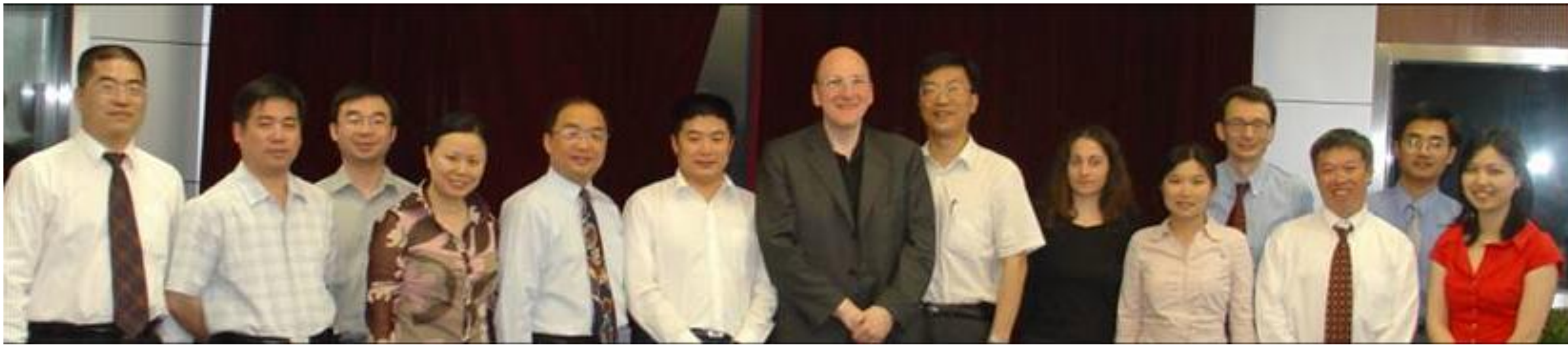
OW2: Object Web + Orient Ware

Leading Open Source Middleware

OW2: Object Web + Orient Ware

ObjectWeb Open Source Middleware

四方国件 ORIENTWARE



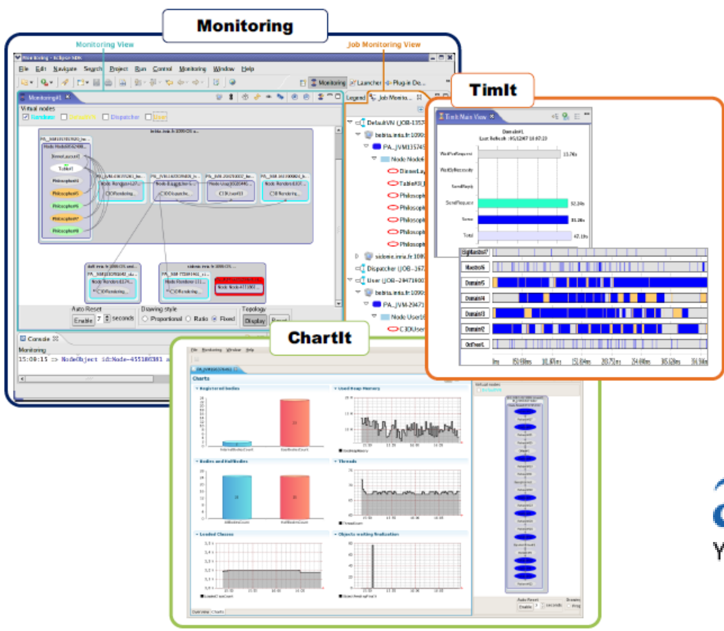
LIU Jiangning (CVIC SE), Prof. MA Dianfu (BEIHANG UNIVERSITY), Prof. WEI Jun (ISCAS), Prof. JIA Yan (NUDT), Prof. WANG Huaiming (NUDT), Mr. YUCHI Jan (MOST), Jean-Pierre Laisné (BULL), Prof. HUAI Jinpeng (BEIHANG UNIVERSITY), Julie Marguerite (INRIA), ZHOU Minghui (PEKING UNIVERSITY), Stephane Grumbach (French Embassy), Hongbo XU (GMRC), ZHOU Bin (NUDT), Than Ha Ngo (French Embassy).

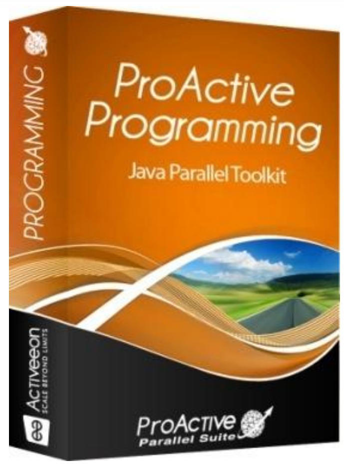
OW2 Consortium

Leading Open Source Middleware

Product: ProActive Parallel Suite

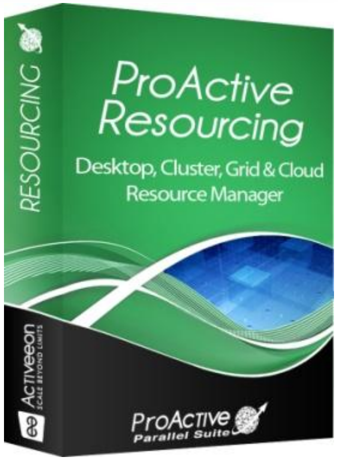
ProActive Parallel Suite





Java Parallel Toolkit

Multi-Platform Job Scheduler



Resource Manager

AMADEUS Your technology partner

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

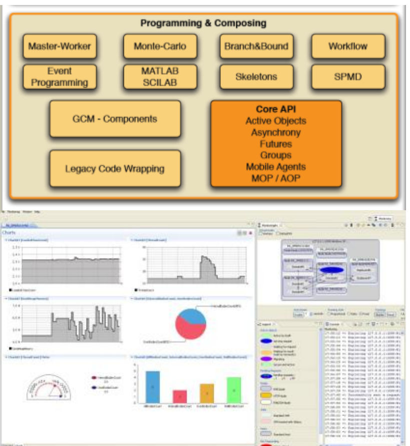
Strong Differentiation:

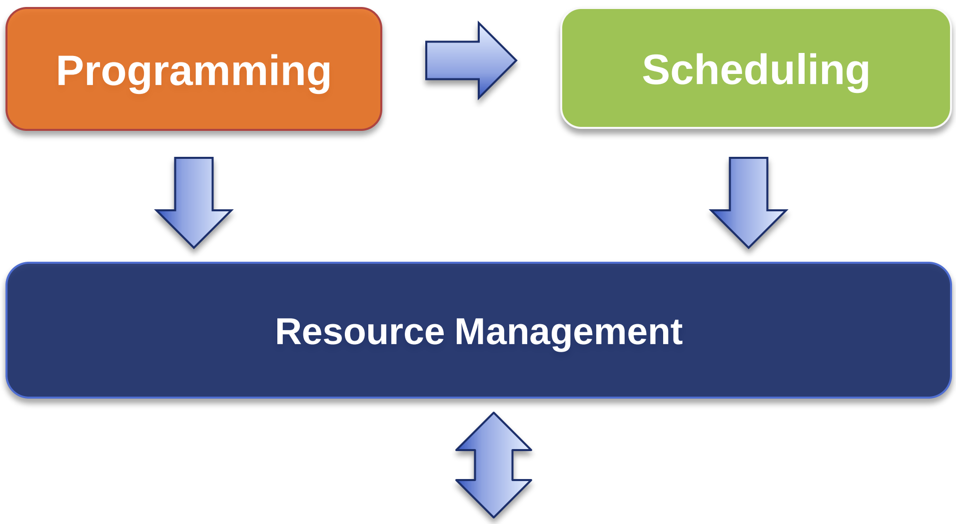
- ❑Java Parallel Programming + Integration +
- ❑Portability: Linux, Windows, Mac +
- ❑Versatility: Desktops, Cluster, Grid, Clouds = Perfect Flexibility

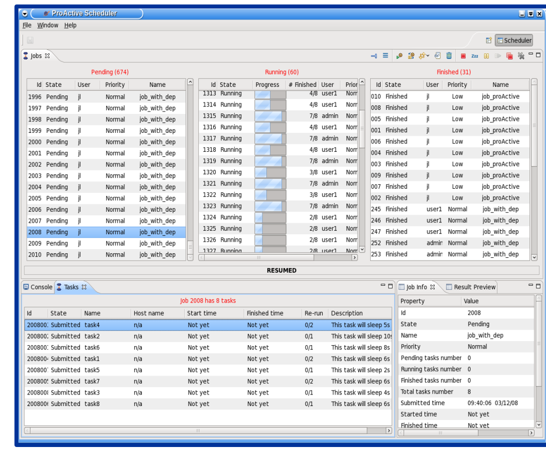
ActiveEon SCALE BEYOND LIMITS · ProActive Parallel Suite · OW2 Consortium

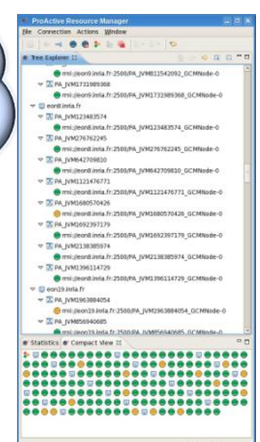
ProActive Parallel Suite

- ❑ Three fully compatible modules:









ProActive Contributors

Abhijeet Gaikwad (Option Pricing)
Abhishek-Rajeev Gupta
Antonio Cansado
Baptiste De Stefano
Bastien Sauvan
Brian Amedro (SPMD)
Cédric Dalmasso (Component)
Clement Mathieu (Core, GCM Deployment)
Elaine Isnard
Elton Mathias
Eric Madelaine
Etienne Vallette-De-Osia
Fabien Viale (Matlab, Scilab)
Fabrice Huet (Mobility, P2P)
Florin Bratu
Franca Perrina
Francoise Baude
Germain Sigety (Scheduling)
Guillaume Laurent
Guilherme Perretti Pezzi
Imen Filiali
Jean-Luc Scheefer (Scheduling)

Jean-Michael Guillamume
Johann Fradj (Scheduling)
Jonathan Martin
Julian Krzeminski
Kamran Qadir
Khan Muhammad
Laurent Vanni
Ludovic Henrio
Marcela Rivera
Mario Leyton (Skeleton)
Maxime Menant
Nicolas Dodelin
Olivier Helin
Paul Naoumenko
Regis Gascon
Tomasz Dobek
Vasile Jureschi (Technical Writer)
Viet Dong Doan
Vincent Cave (Legacy Wrapping)
Virginie Contes (OSGi, WS)
Yu Feng
Yulai Yuan
Zhihui Dai

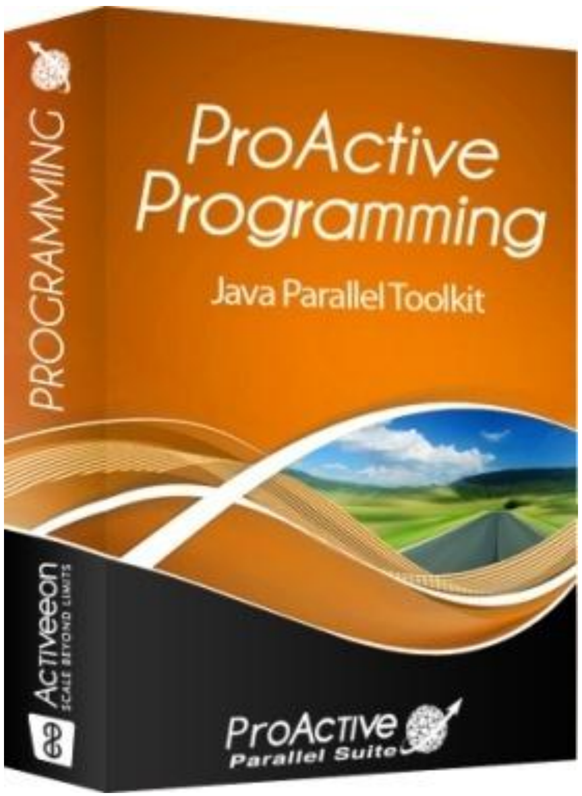
Arnaud Contes (Core, Security, Debugging)
Christian Delbé (FT, Scheduling)
Emil Salageanu (IC2D, Agent)
Vladimir Bodnartchouk (IC2D, TimIt)

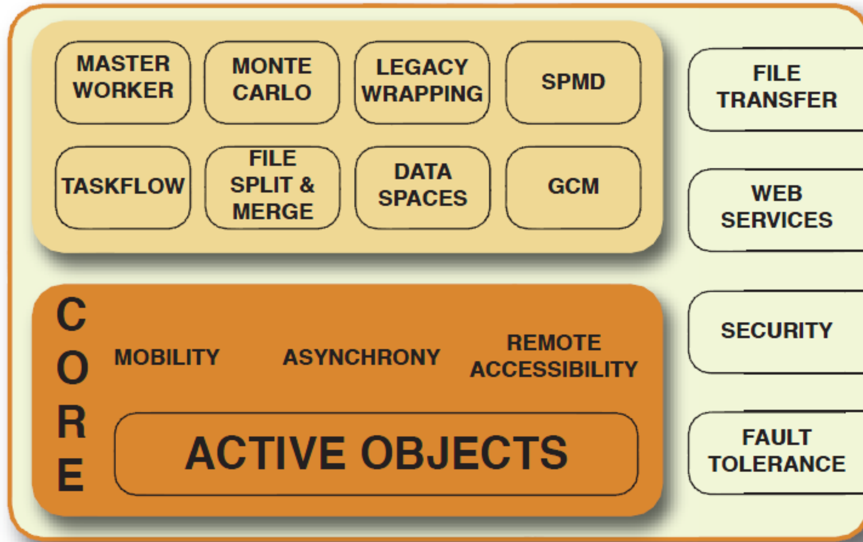
Alexandre di Costanzo (P2P, B&B)
Boutheina Bennour
Guillaume Chazarain (DGC)
Julien Vayssiere (MOP, Active Objects)

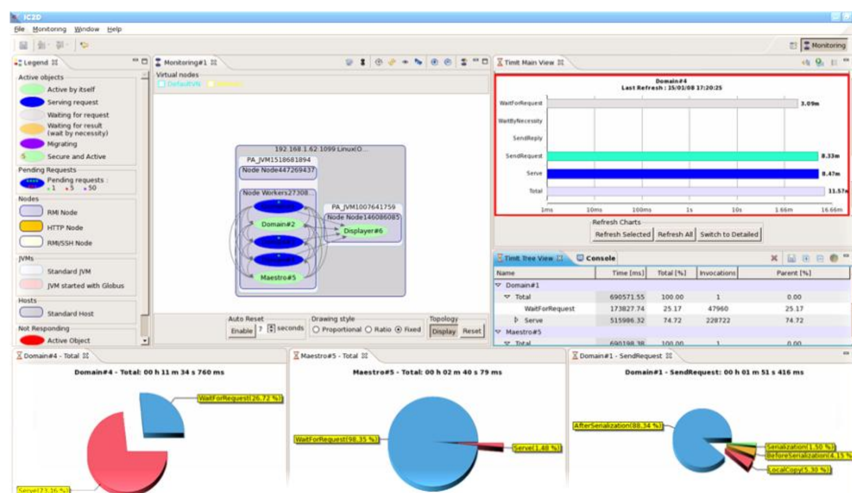
Lionel Mestre
Laurent Baduel (Group Communications)
Matthieu Morel (Initial Component Work)
Nadia Ranaldo (Core, Deployment)
Romain Quilici

ProActive Programming: Active Objects

ProActive Programming







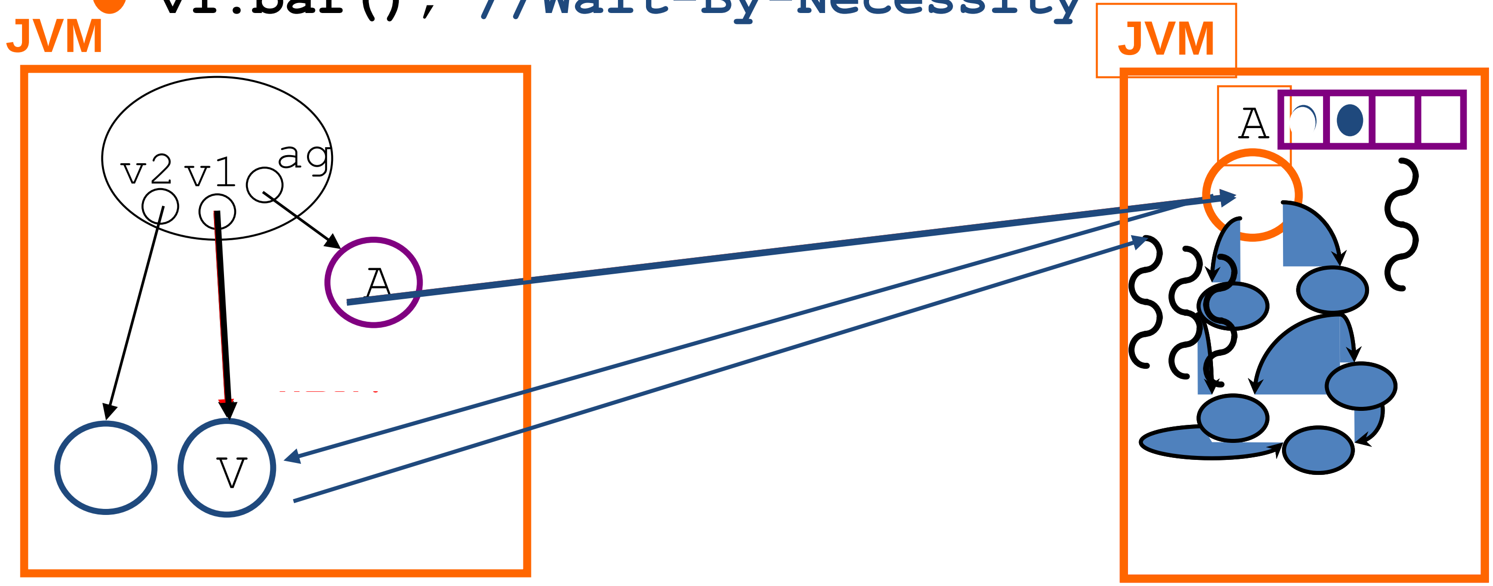
ActiveEon SCALE BEYOND LIMITS

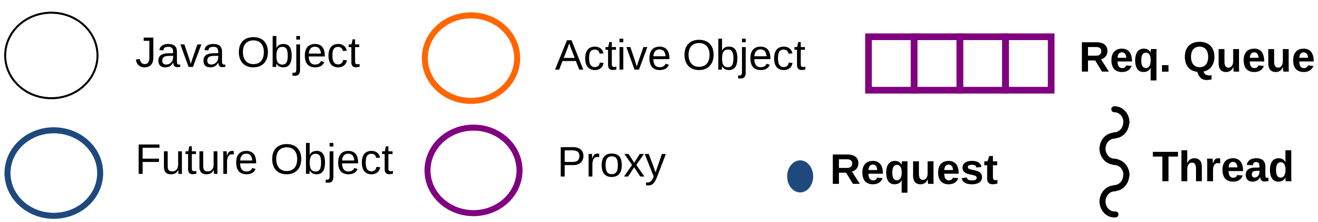
ProActive Parallel Suite

OW2 Consortium

ProActive : Active objects

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



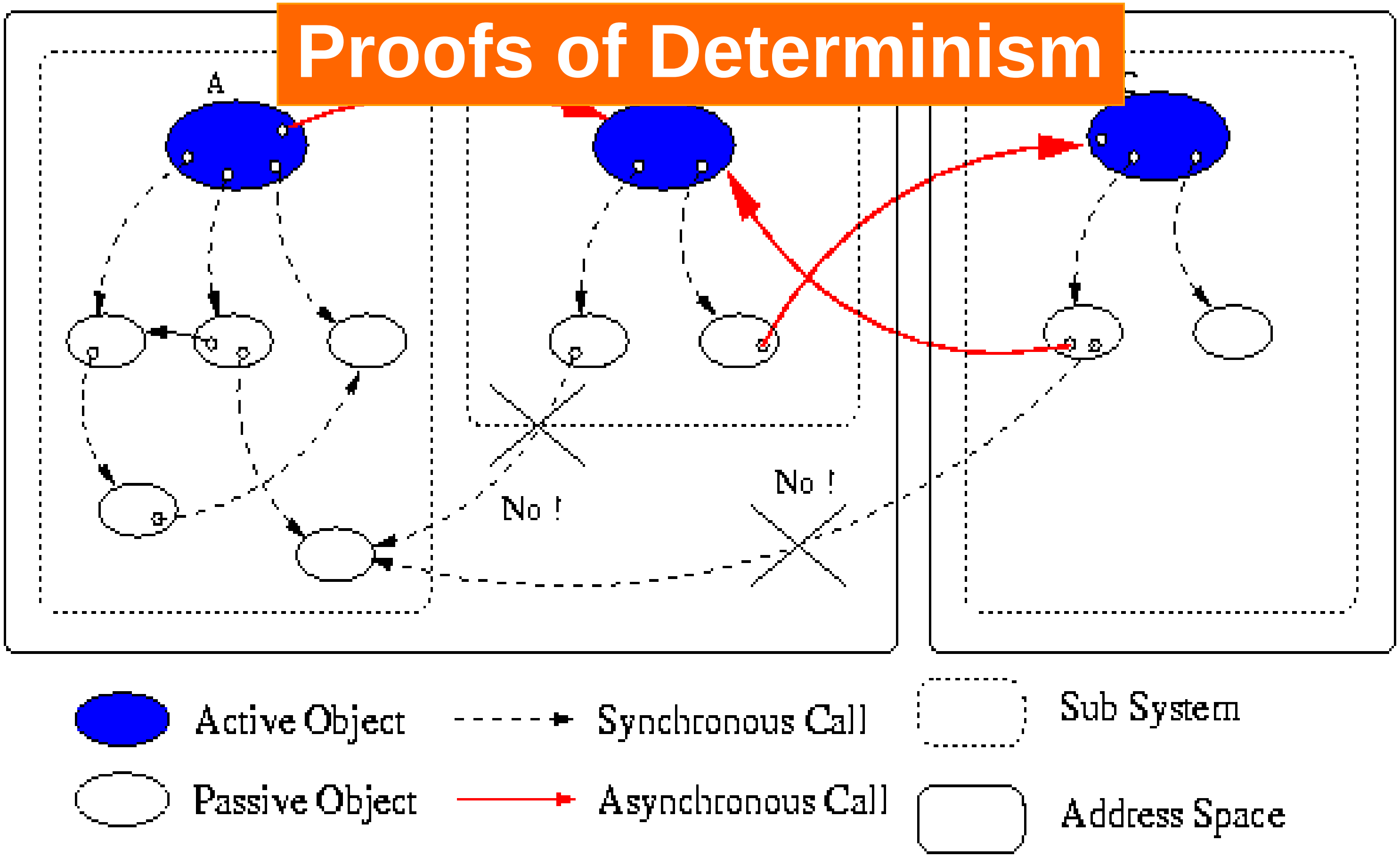


Wait-By-Necessity is a Dataflow Synchronization

Standard system at Runtime: No Sharing

NoC: Network On Chip

Proofs of Determinism



(2) ASP: Asynchronous Sequential Processes

Local

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

Creating an Activity

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

Sending a Request

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

Service

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

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

Sending a Reply

$$\frac{\sigma_\alpha(\iota) = fut(f_i^{\gamma\rightarrow\beta}) \quad F_\beta(f_i^{\gamma\rightarrow\beta}) = \iota_f \quad \sigma'_\alpha = Copy\&Merge(\sigma_\beta,\iota_f \ ; \ \sigma_\alpha,\iota)}{\alpha[a_\alpha;\sigma_\alpha;\iota_\alpha;F_\alpha;R_\alpha;f_\alpha] \parallel \beta[a_\beta;\sigma_\beta;\iota_\beta;F_\beta;R_\beta;f_\beta] \parallel P \longrightarrow \alpha[a_\alpha;\sigma'_\alpha;\iota_\alpha;F_\alpha;R_\alpha;f_\alpha] \parallel \beta[a_\beta;\sigma_\beta;\iota_\beta;F_\beta;R_\beta;f_\beta] \parallel P} \ (\text{REPLY})$$

Key Point:
Locality will more than ever be Fundamental

- ❑Let the programmer control it
- ❑No global shared memory

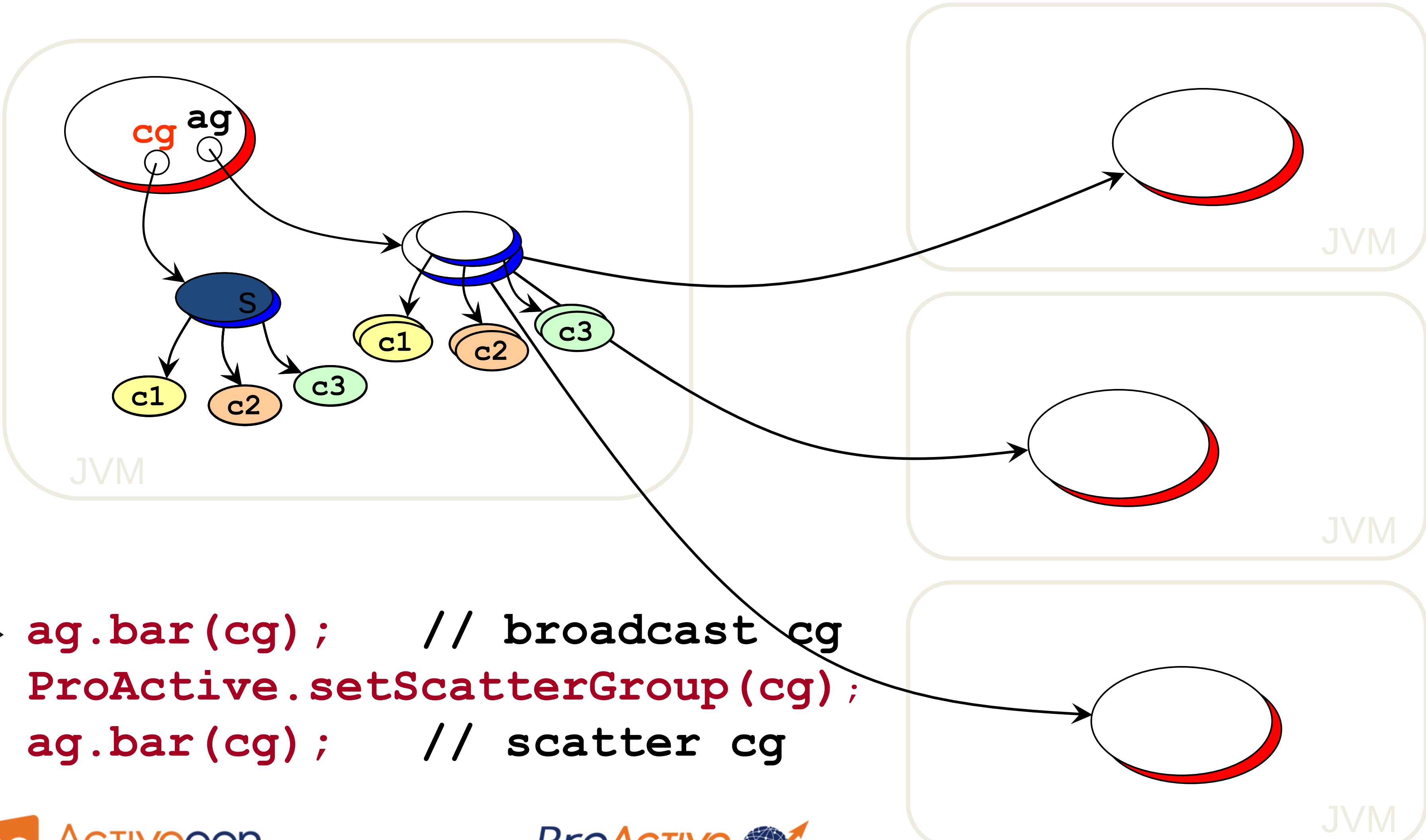
At user choice ***PGAS:*** *Partitioned Global Address Space*

TYPED ASYNCHRONOUS GROUPS

Broadcast and Scatter

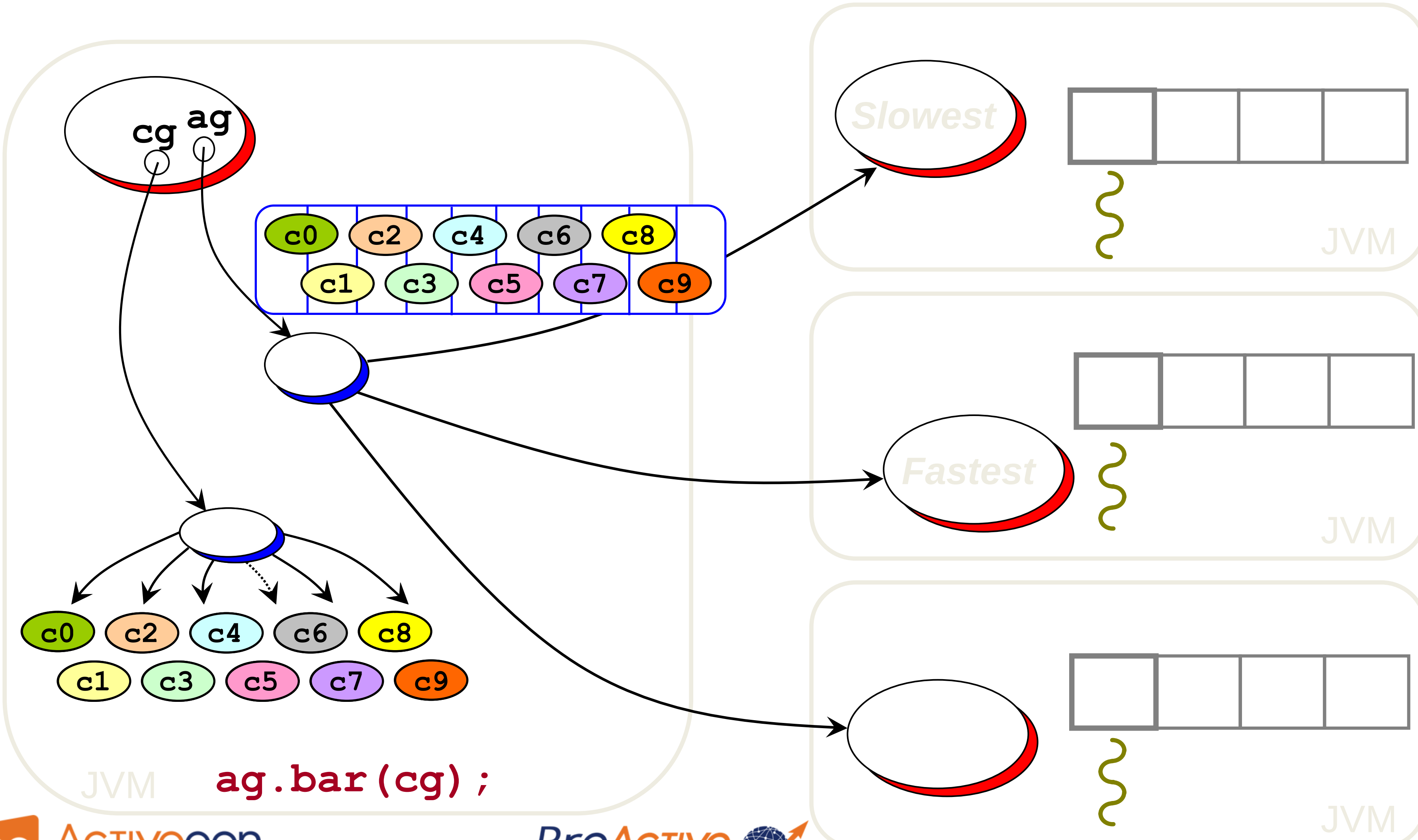
Broadcast is the default behavior

Use a group as parameter, Scattered depends on rankings




```
ag.bar(cg);   // broadcast cg
ProActive.setScatterGroup(cg);
ag.bar(cg);   // scatter cg
```

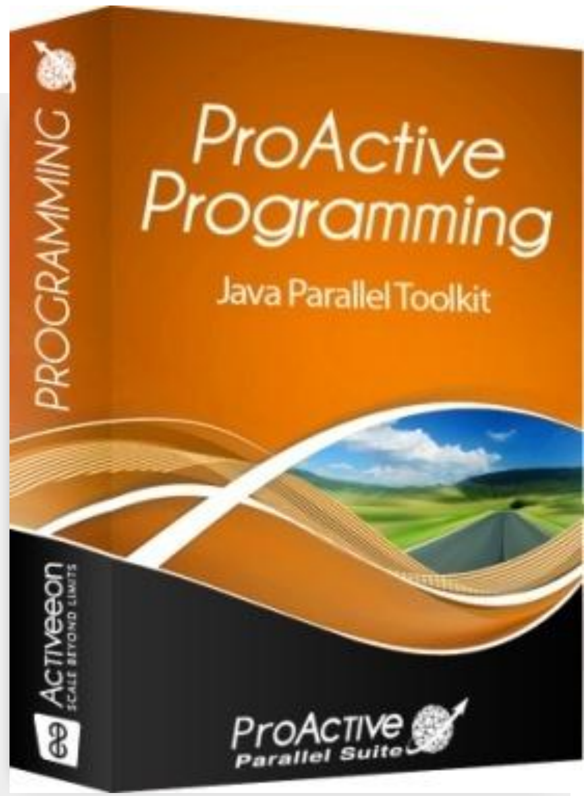
Dynamic Dispatch Group



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

Components: GCM Standard

GridCOMP Partners

European Research Consortium for Informatics and Mathematics ERCIM

INRIA

Université Nice Sophia Antipolis

CNRS Centre National de la Recherche Scientifique

In Supremæ Dignitatis 1343

University of Westminster

ISTI Istituto di Scienza e Tecnologie dell'Informazione "A. Faedo"

GridSystems Simplify Complexity

IBM

Atos Origin Worldwide IT Partner

清華大學 Tsinghua University

Postera Crescam Laude
The University of Melbourne

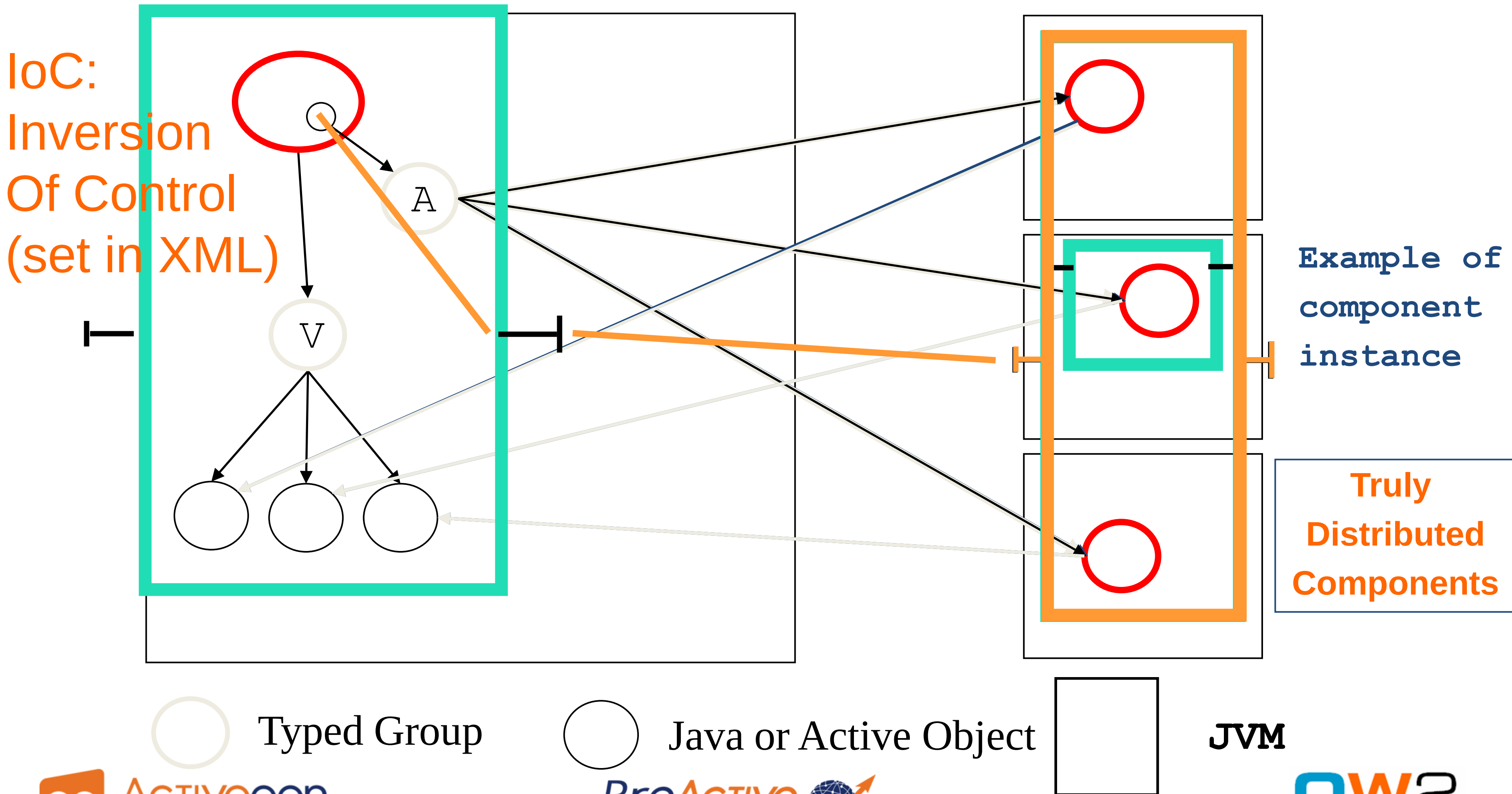
Universidad de Chile

Activeeon Scale Beyond Limits

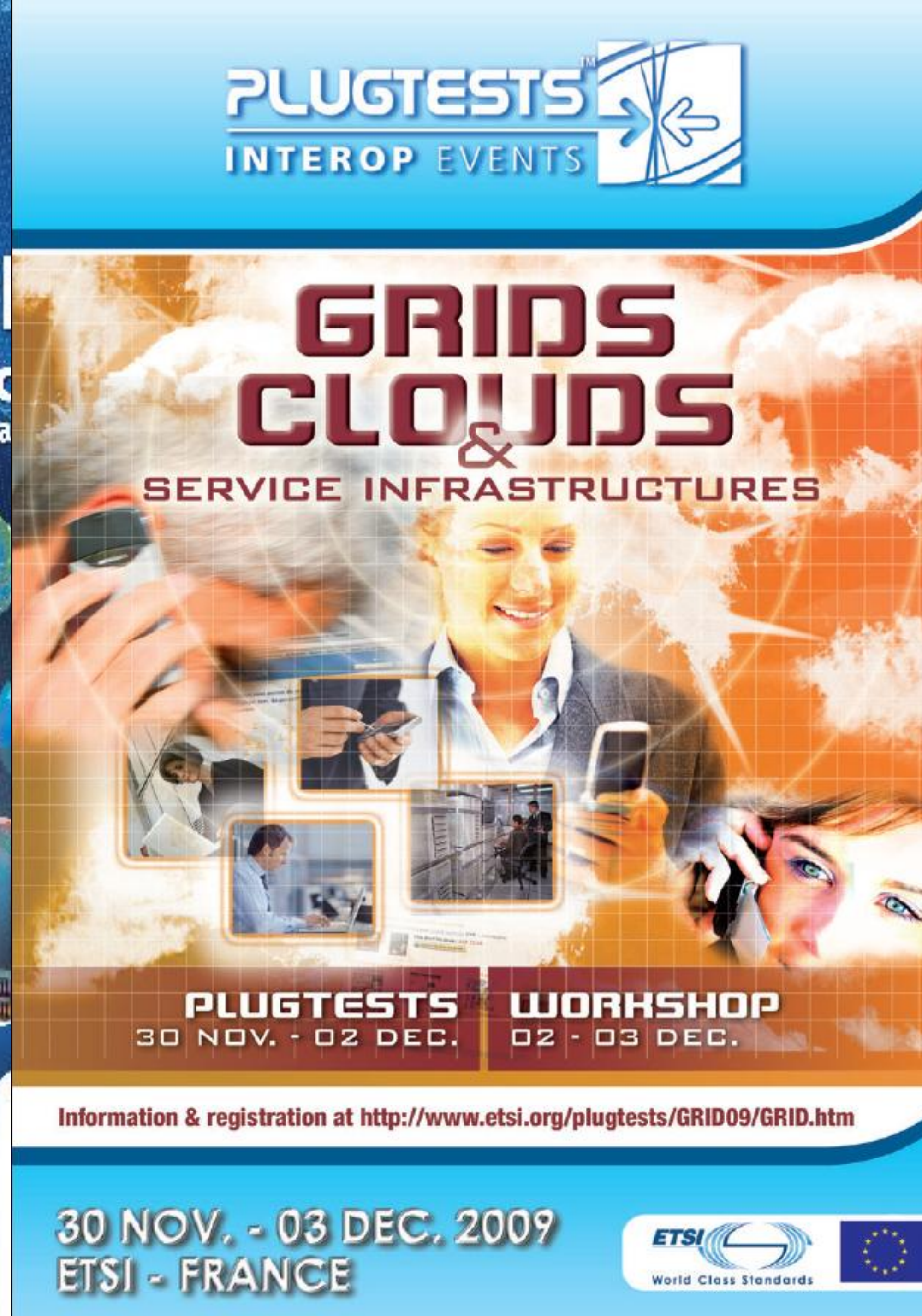
ProActive Parallel Suite

OW2 Consortium

Objects to Distributed Components




PLUGTESTS
INTEROP EVENTS
GRIDS for Finance
& Telecommunications
V GRI
20-24 Oc
INRIA – Sophia
Information & Registration: www.etsi.org/plugtests/GRID2008/GRID.htm
With the support of
European partners
GridCOMP
ProActive
OW2
FMCO
Industrial sponsors
Microsoft
ActiveEon


PLUGTESTS
INTEROP EVENTS
GRIDS
& CLOUDS
SERVICE INFRASTRUCTURES
PLUGTESTS
30 NOV. - 02 DEC.
WORKSHOP
02 - 03 DEC.
Information & registration at http://www.etsi.org/plugtests/GRID09/GRID.htm
30 NOV. - 03 DEC. 2009
ETSI - FRANCE
ETSI
World Class Standards

From 2004 to 2008:

- **2004 Grid Plugtests:**

  Winner: Univ CHILE

  Deployed 560 Workers all over the world

  on a very heterogeneous infrastructure (no VO)

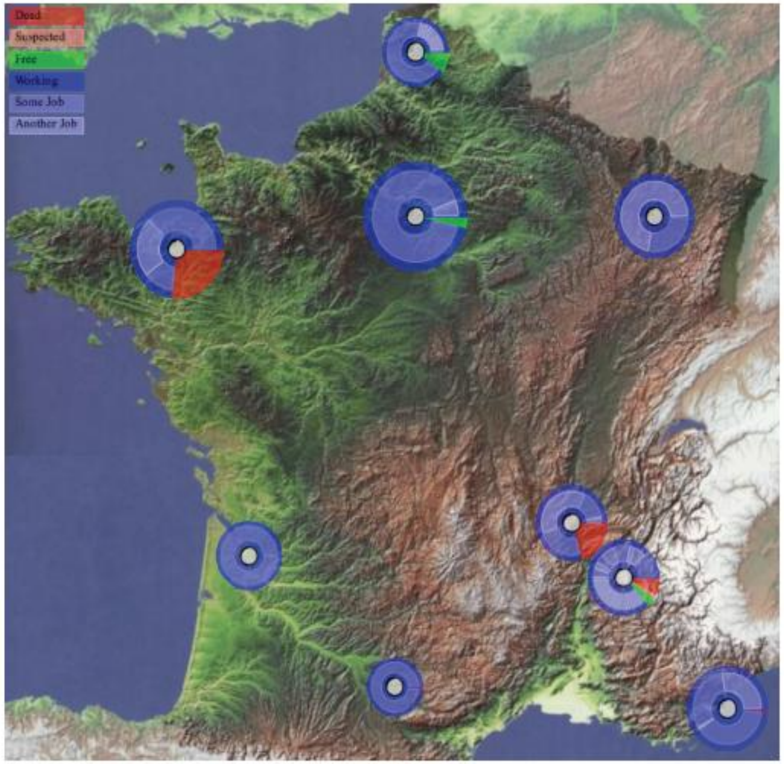
- **2008 Grid Plugtests:**

  KAAPI, MOAIS Grenoble: <u>3609 Nodes</u>

  ACT, China: Beihang University, Beijing, China:

  <u>4329 Nodes</u>

Grid'5000*



Lille:
500 (198)
Orsay
1000 (684)
Nancy:
500 (334)
Lyon
500 (252)
Grenoble
500 (270)
Rennes
522 (522)
Toulouse
500 (116)
Bordeaux
500 (198)
Sophia Antipoli
500 (434)

ministère éducation nationale enseignement supérieur recherche
ministère délégué à la recherche
ACI GRID
INRIA
Renater
CNRS CENTRE NATIONAL DE LA RECHERCHE SCIENTIFIQUE

ObjectWeb
Open Source Middleware

ProActive

Programming, Composing, Deploying on the Grid

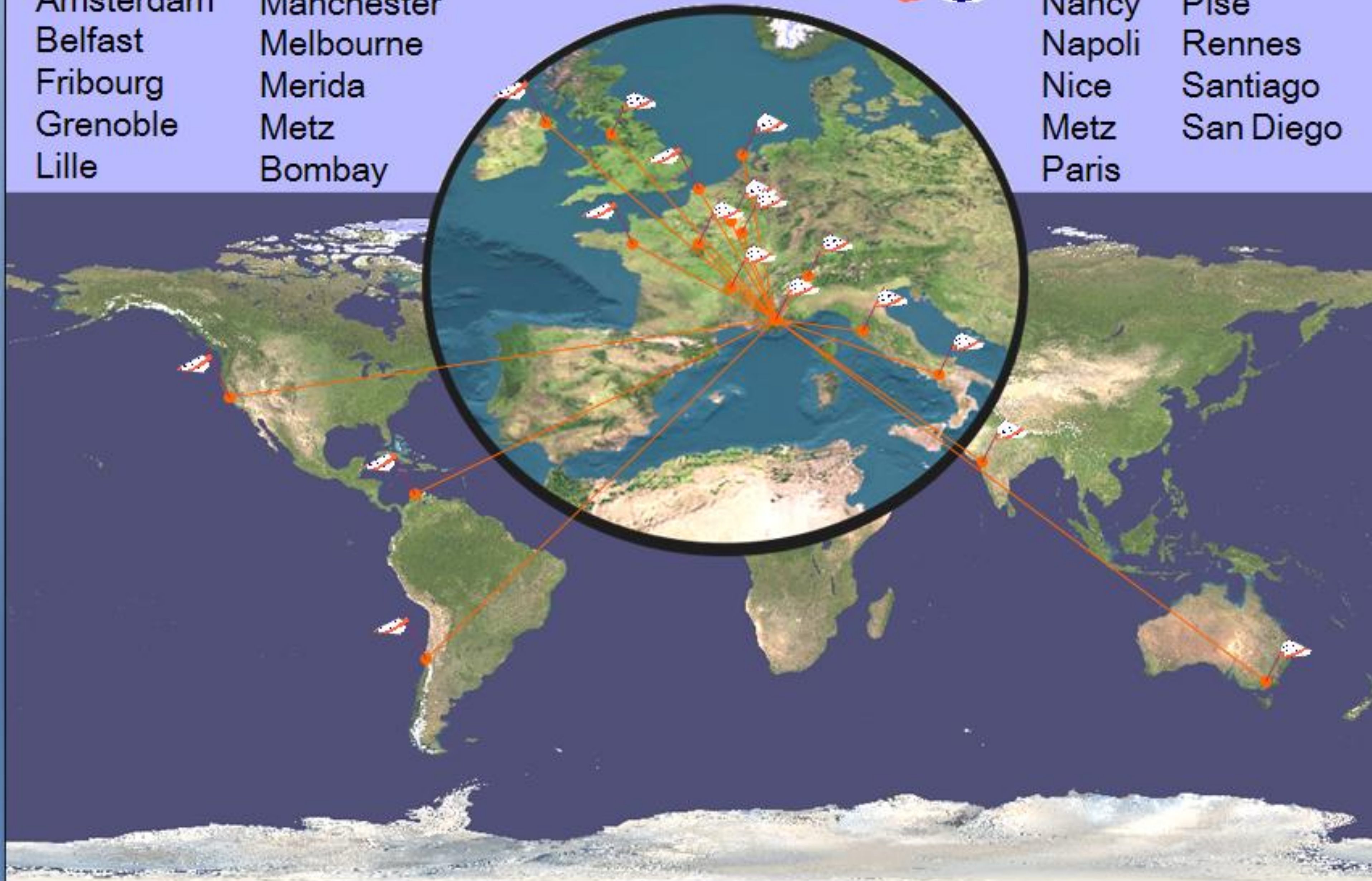
PLUGTESTS
THE INTEROPERABILITY SERVICE

Amsterdam
Belfast
Fribourg
Grenoble
Lille

Manchester
Melbourne
Merida
Metz
Bombay

Nancy
Napoli
Nice
Metz
Paris

Pise
Rennes
Santiago
San Diego



Chinese Collaborations on Grid PlugTests

- Professor Chi
- Prof. Baoping Yan
- Hosted the IV Grid Plugtests Grid@works 2007
- CNIC: Computer and Network Information Center
- SCC AS: Super Computing Center of AS

- Prof. Ji Wang
- In EchoGrid, Chinese Leader of OW2
- NUDT: National Univ. of Defense Technology
- PDL: Laboratory of Parallel & Distributed Processing

Infrastructure tested in Plugtests and in GCM Deployment Standard

- **Protocols:**
  - Rsh, ssh
  - Oarsh, Gsissh
- **Scheduler, and Grids:**
  - GroupSSH, GroupRSH, GroupOARSH
  - ARC (NorduGrid), CGSP China Grid, EEGE gLITE,
  - Fura/InnerGrid (GridSystem Inc.)
  - GLOBUS, GridBus
  - IBM Load Leveler, LSF, Microsoft CCS (Windows HPC Server 2008)
  - Sun Grid Engine, OAR, PBS / Torque, PRUN
- **Clouds:**
  - Amazon EC2

GCM Official Standardization

Grid Component Model

World Class Standards

Overall, the standardization is supported by industrials:

BT, FT-Orange, Nokia-Siemens, NEC, Telefonica, Alcatel-Lucent, Huawei …

Infrastructure tested in Plugtests and in GCM Deployment Standard

Interoperability:
Cloud will start with existing IT infrastructure,
Build Non Intrusive Cloud with ProActive

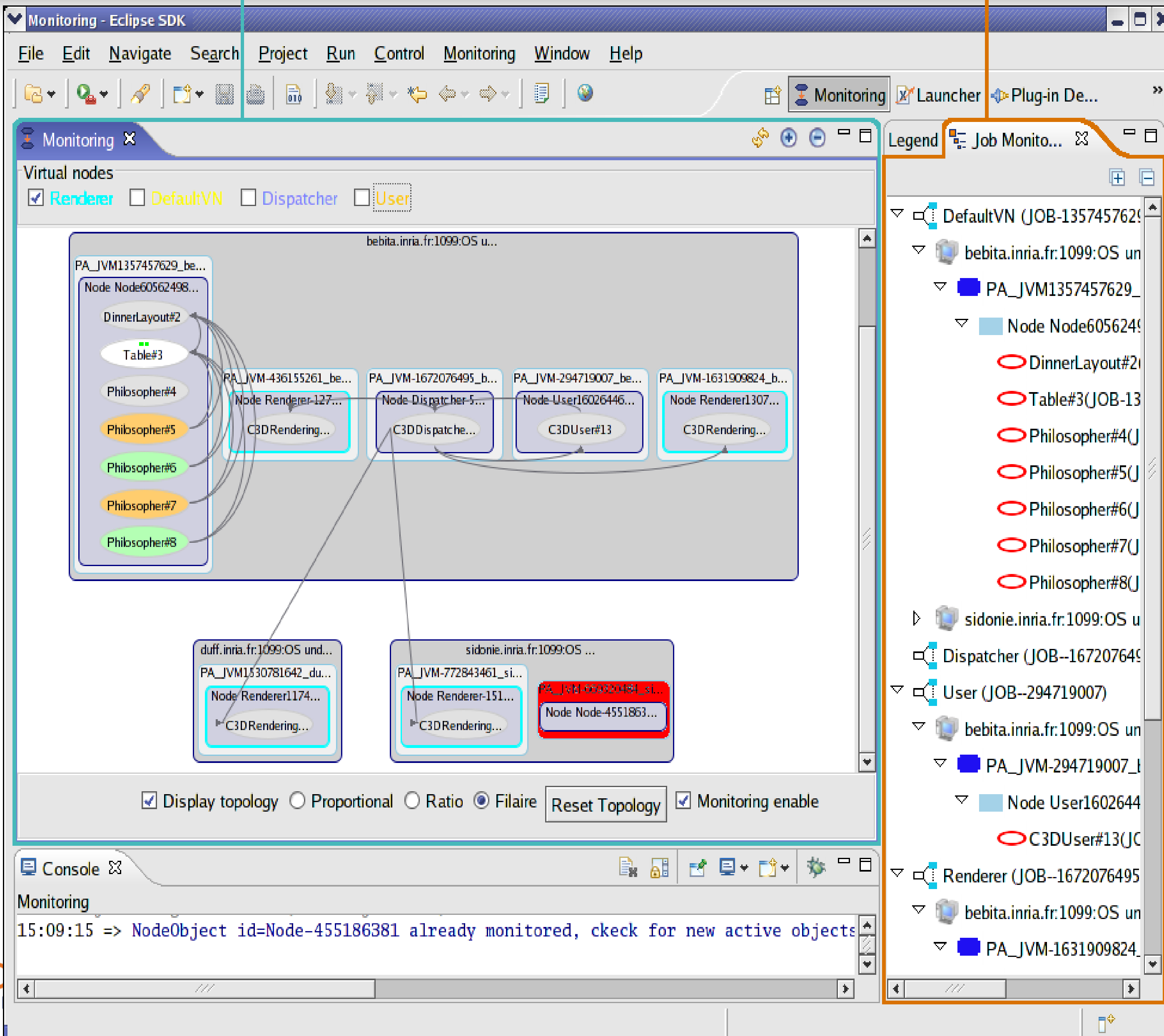
- **Protocols:**
  - Rsh, ssh
  - Oarsh, Gsissh
- **Scheduler, and Grids:**
  - GroupSSH, GroupRSH, GroupOARSH
  - ARC (NorduGrid), CGSP China Grid, EEGE gLITE,
  - Fura/InnerGrid (GridSystem Inc.)
  - GLOBUS, GridBus
  - IBM Load Leveler, LSF, Microsoft CCS (Windows HPC Server 2008)
  - Sun Grid Engine, OAR, PBS / Torque, PRUN
- **Clouds:**
  - Amazon EC2

IC2D: Optimizing

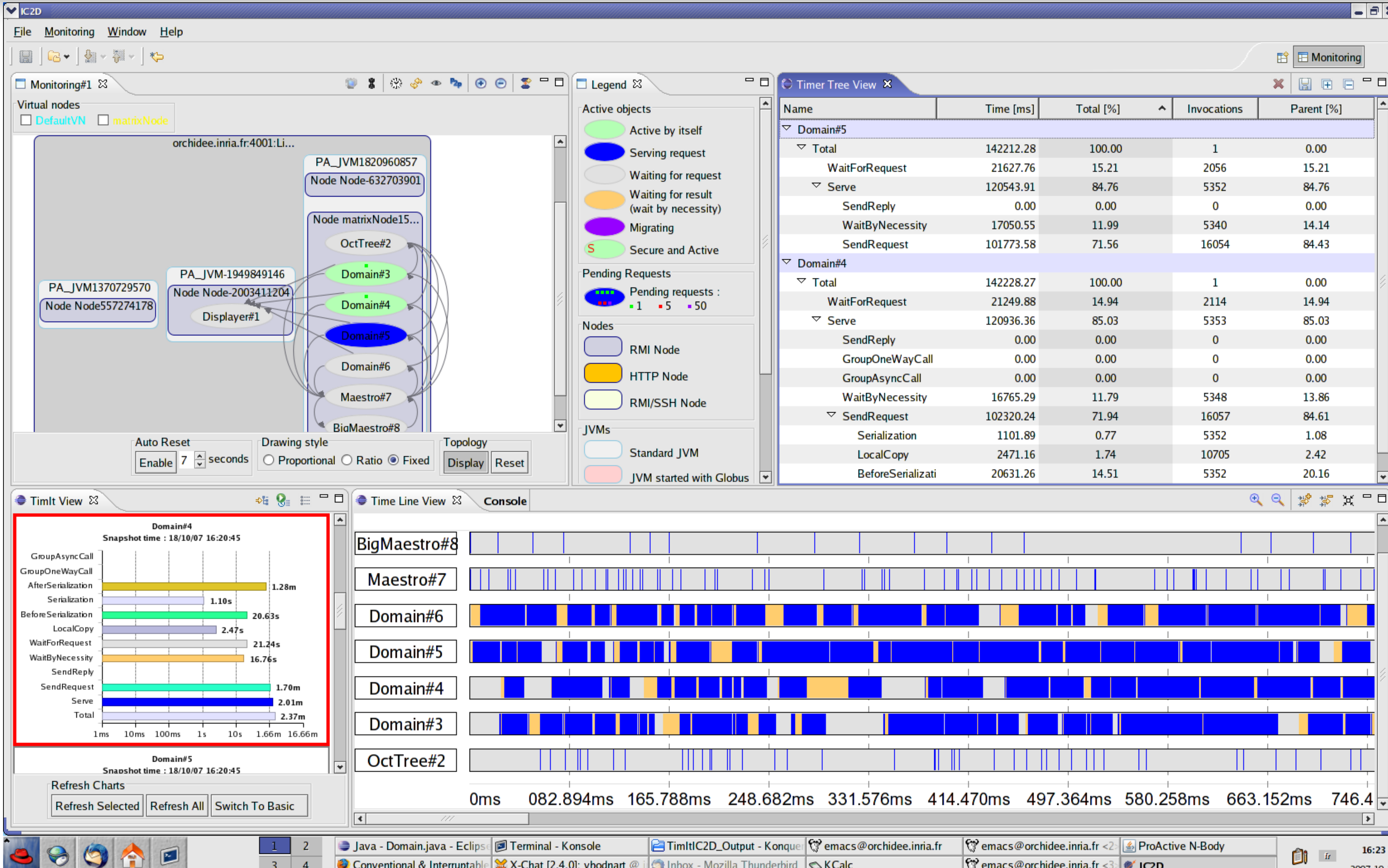
IC2D

Monitoring View

Job Monitoring View



IC2D



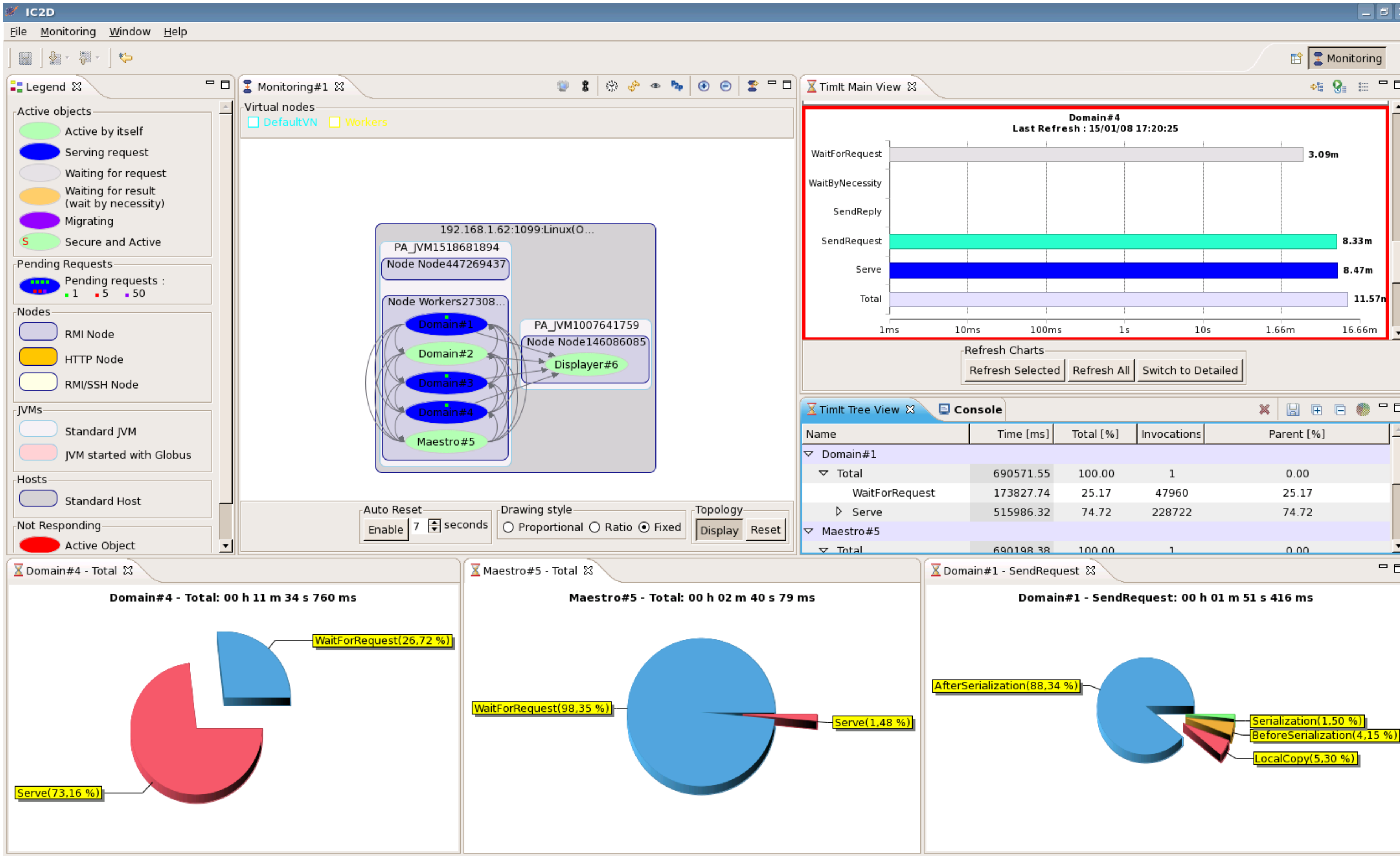
ActiveEon SCALE BEYOND LIMITS

ProActive Parallel Suite

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

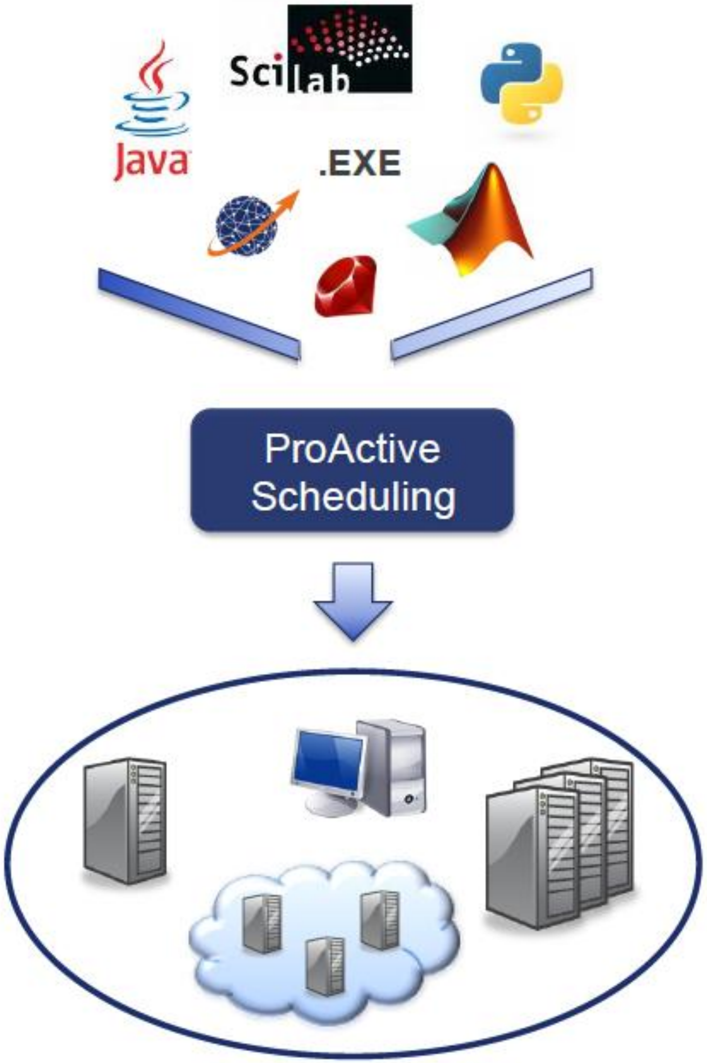
ActiveEon SCALE BEYOND LIMITS

ProActive Parallel Suite

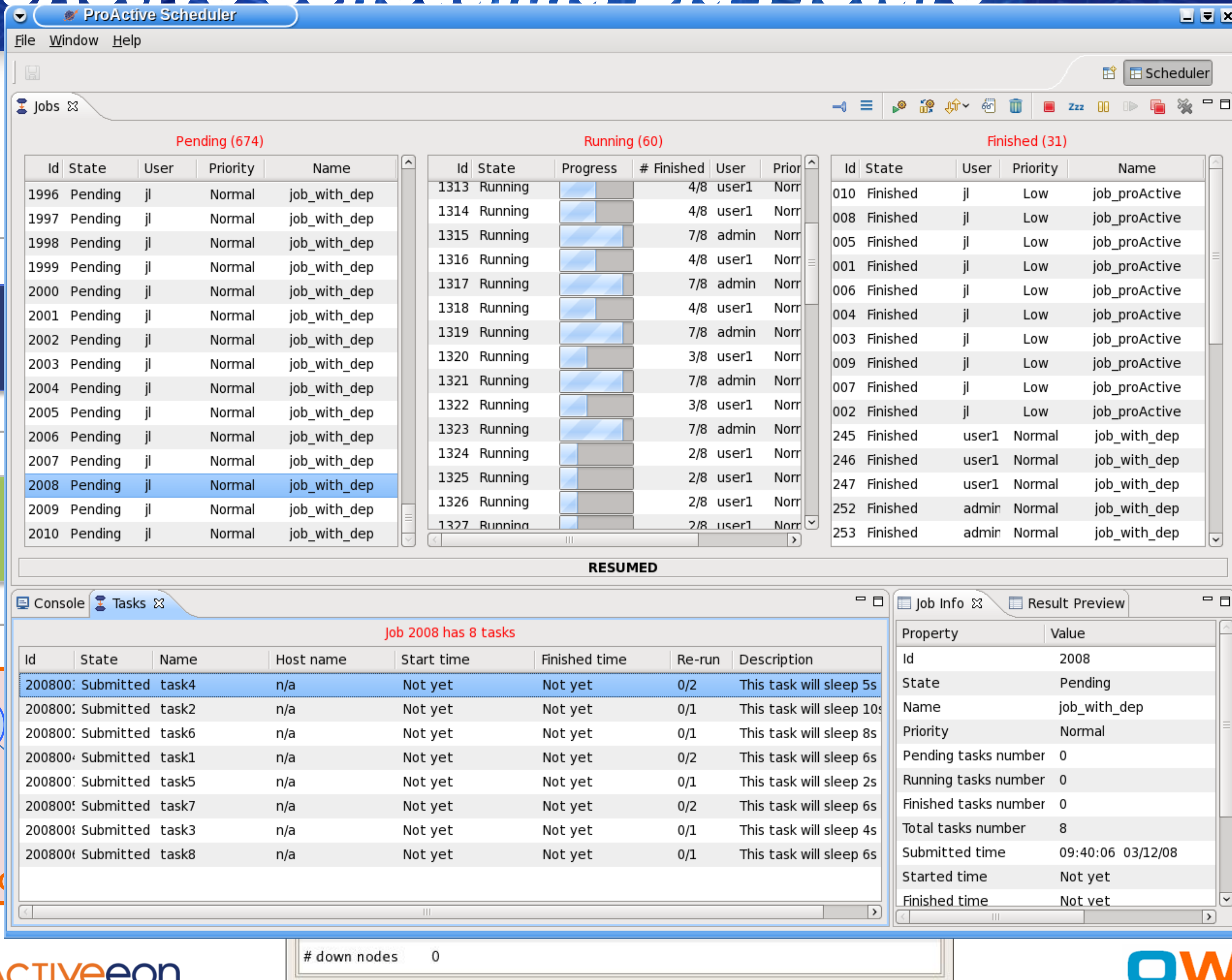
OW2 Consortium

Scheduling & Resourcing

ProActive Scheduling



ProActive Scheduling Big Picture



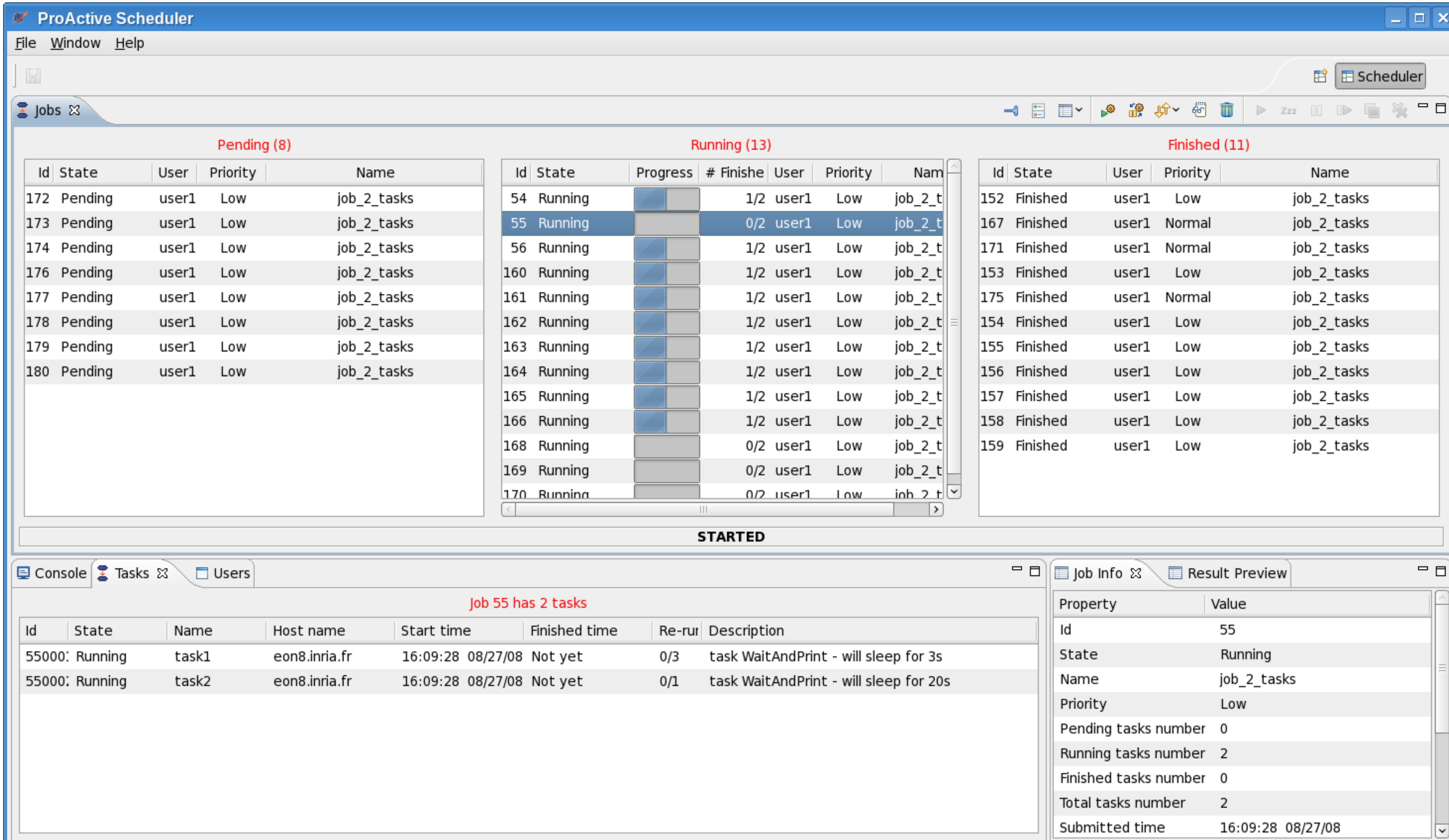
RESO

Activeeon SCALE BEYOND LIMITS

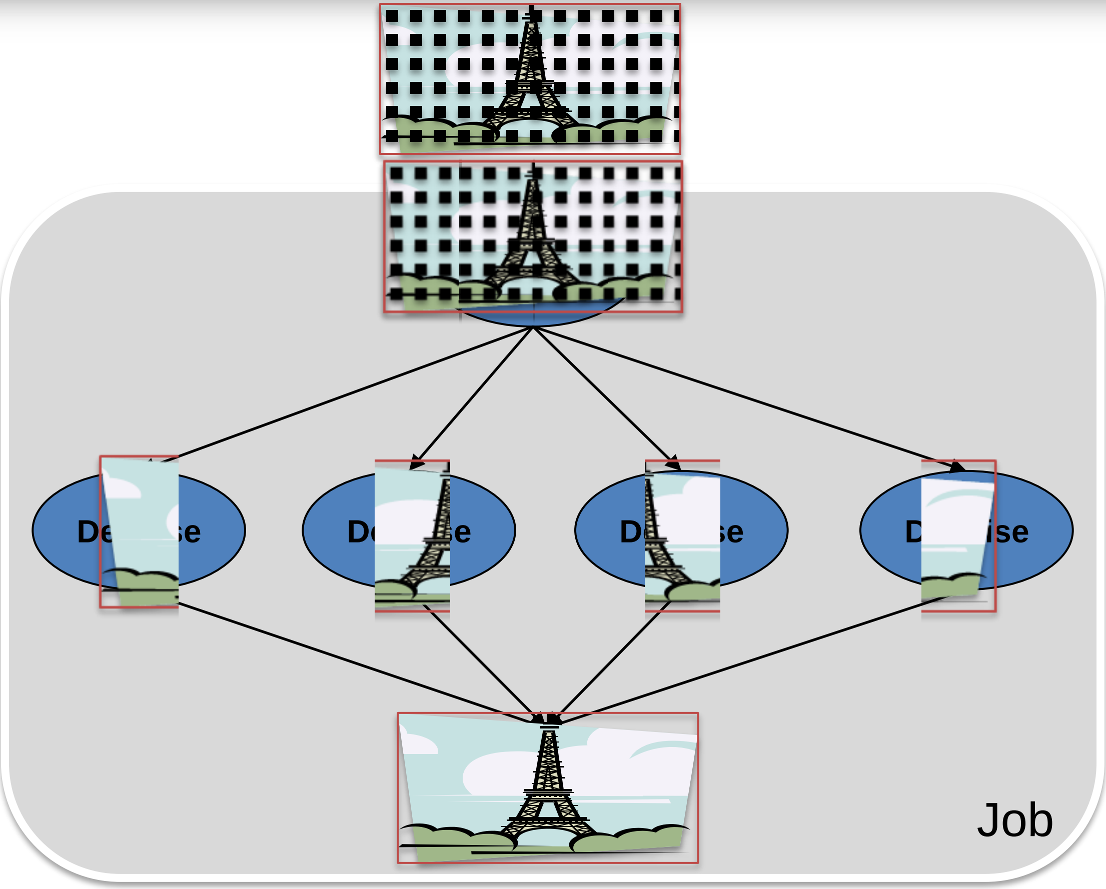
Parallel Suite

OW2 Consortium

Scheduler: User Interface



Another Example : Picture Denoising



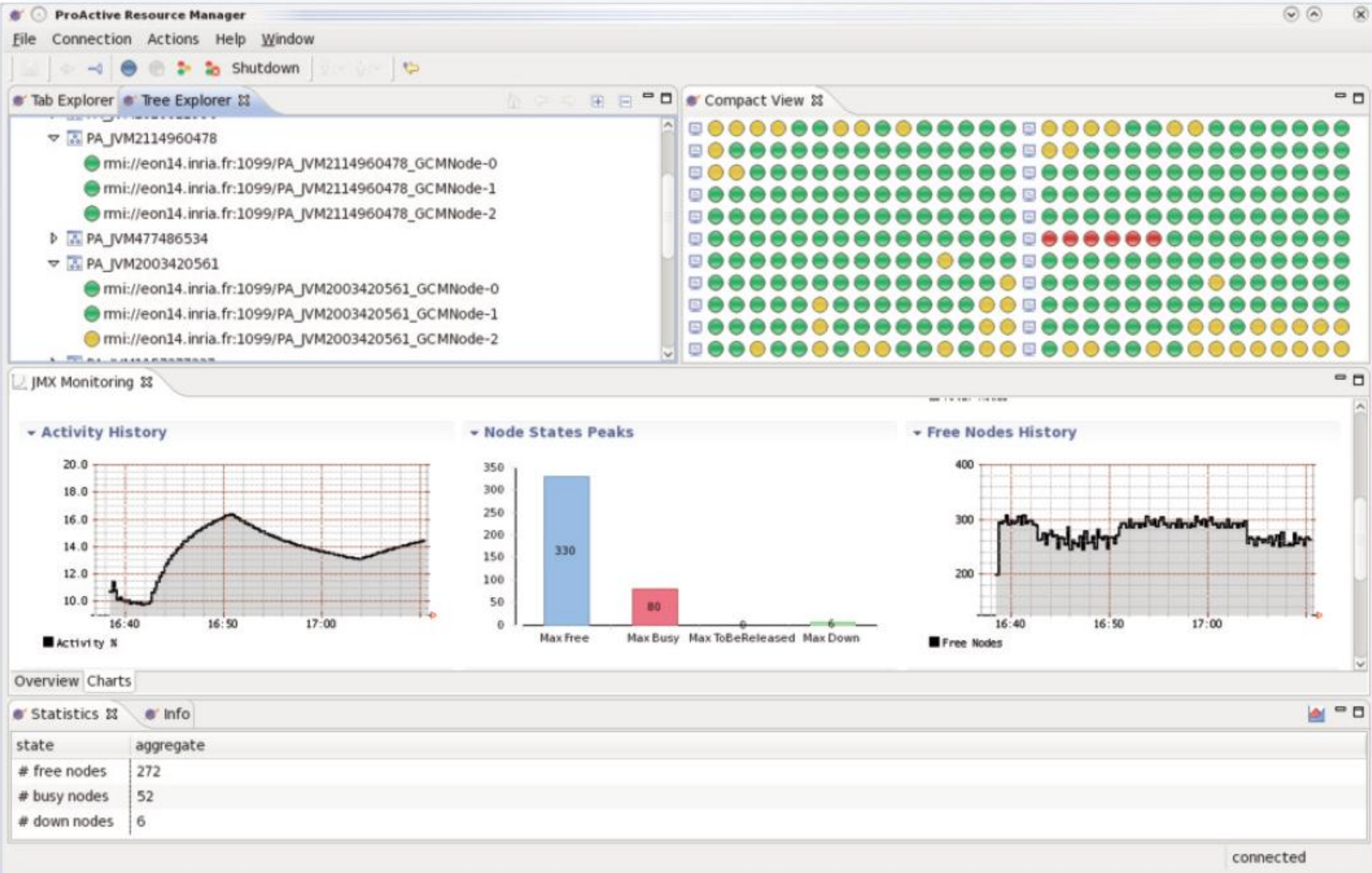
•with selection on native executable availability (ImageMagik, GREYstoration)

• Multi-platform selection and command generation

•with file transfer in pre/post scripts

ProActive Resourcing

RESOURCING User Interface



Video 2:
Scheduler, Resource Manager

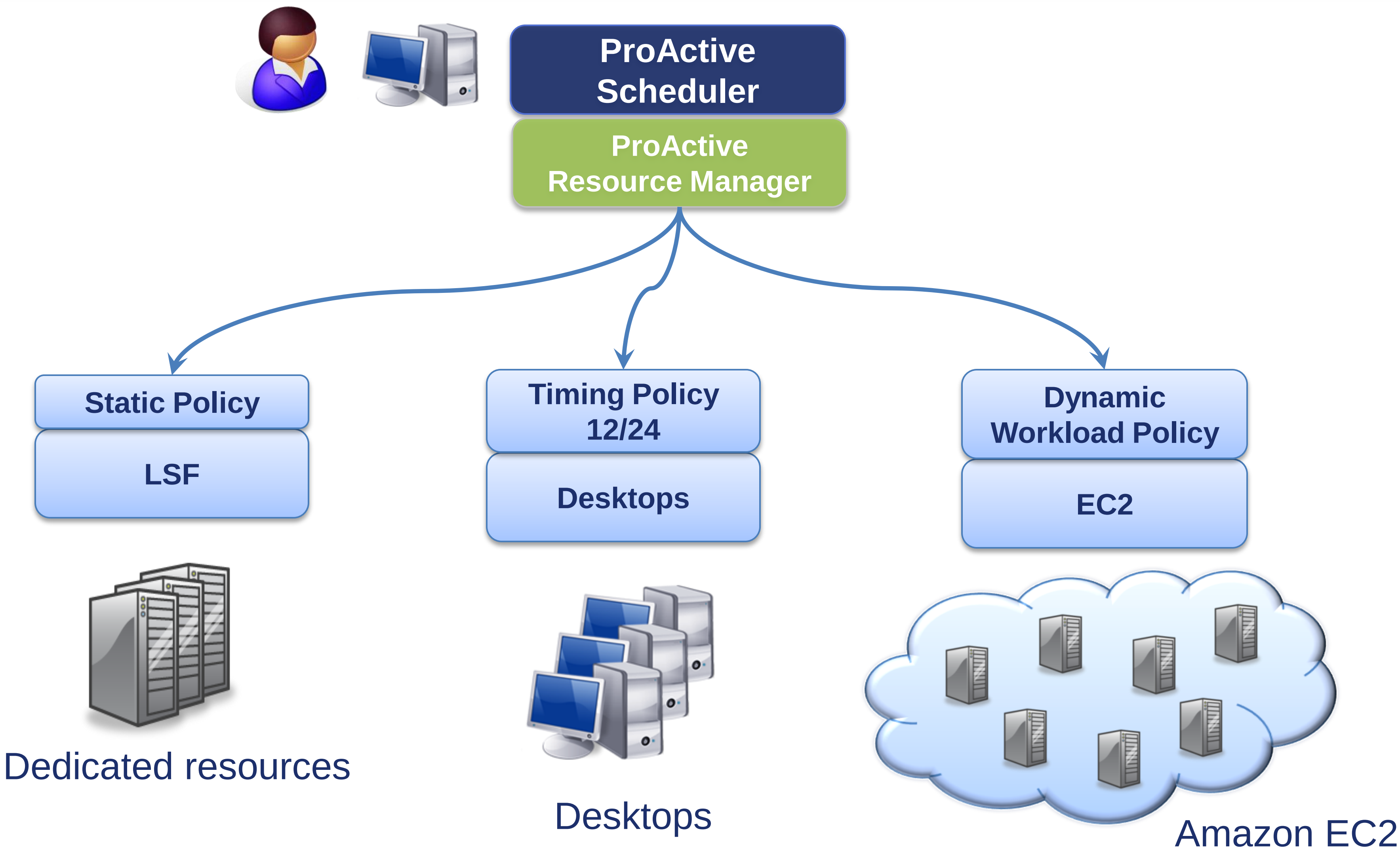
Activeeon SCALE BEYOND LIMITS

ProActive Parallel Suite

OW2 Consortium

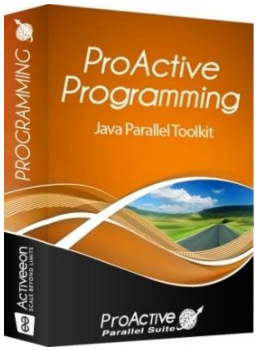
Clusters to Grids to Clouds: e.g. on Amazon EC2

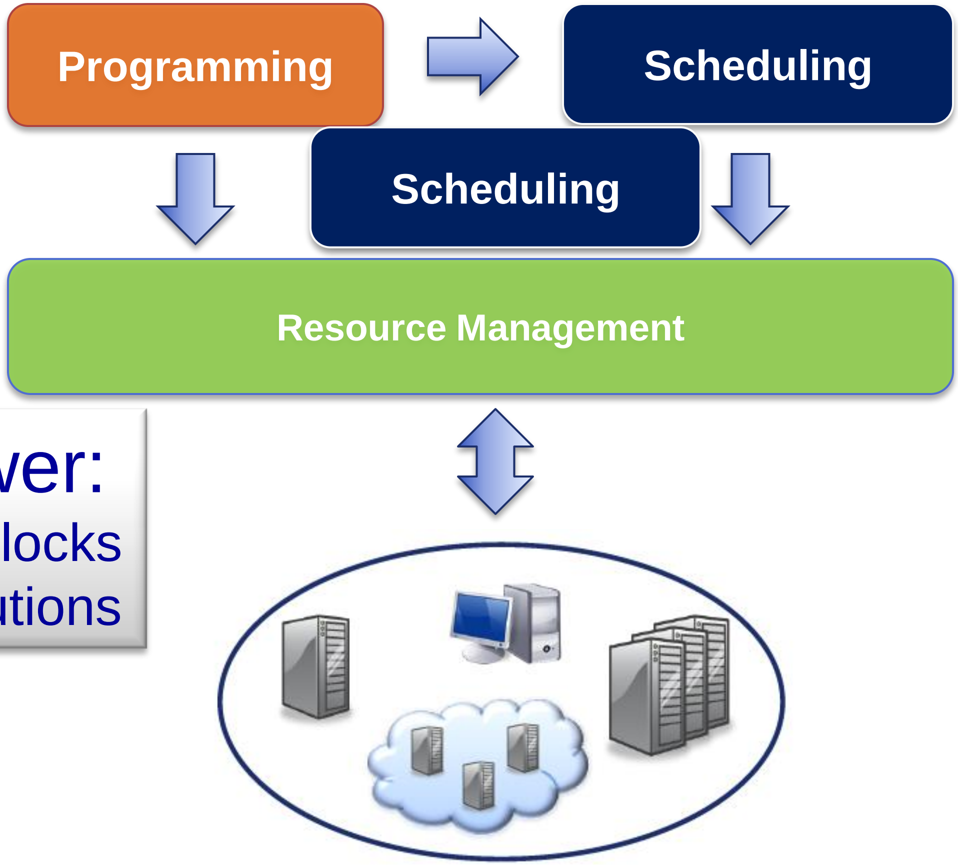
Node source Usecase : Configuration for external cloud with EC2



ProActive Parallel Suite

Three fully compatible modules





Clutch Power:
Solid Building Blocks for Flexible Solutions

ActiveEon SCALE BEYOND LIMITS

ProActive Parallel Suite

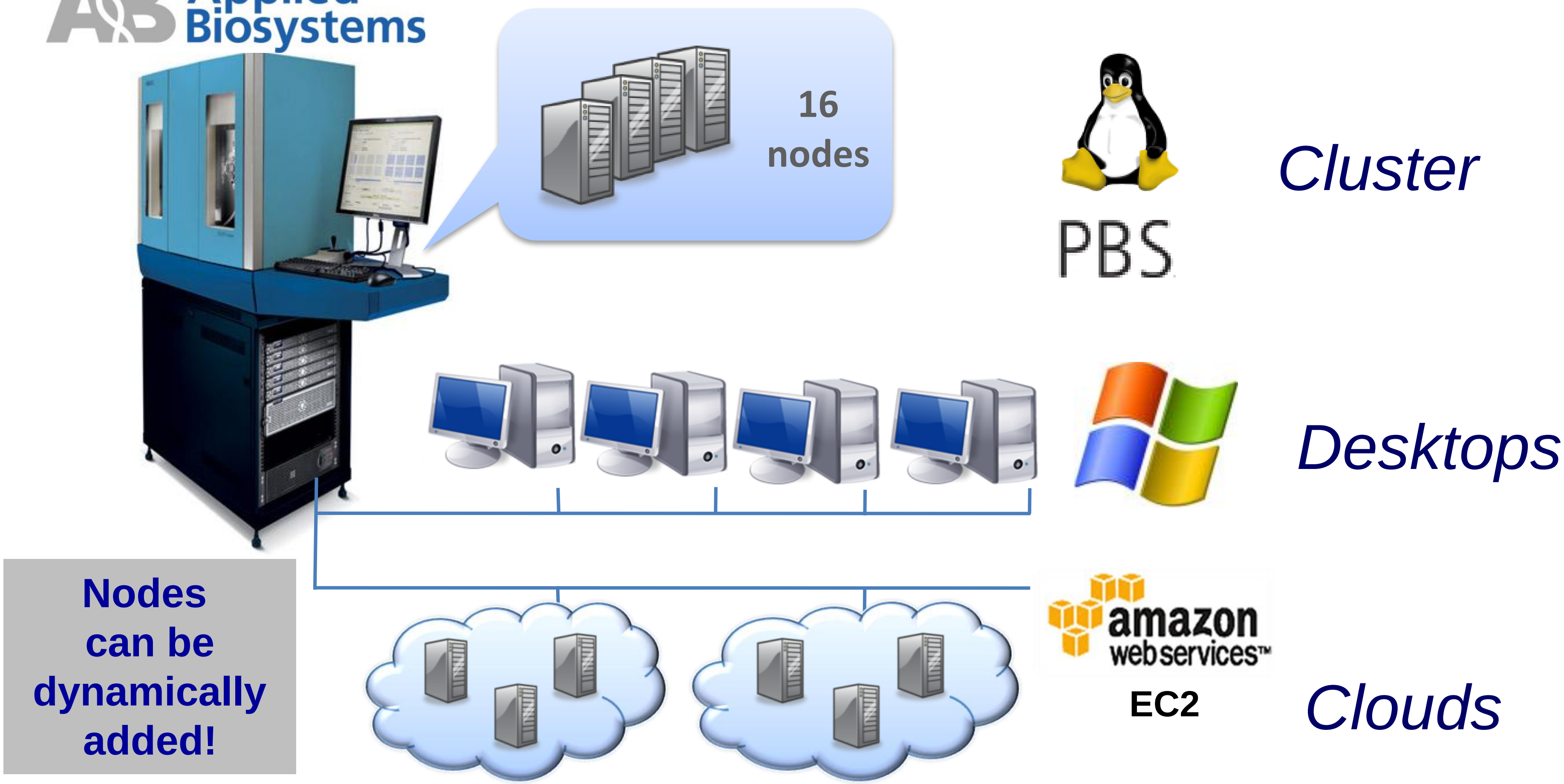
3. Use Case: Genomics

SOLiD and ProActive

- SOLiD Transcriptom Pipeline:
  - Genomic Sequencing Solution
  - Including Multi-language tools, partially ported on Windows
  - Pipelined with Java wrappers
- SOLiD Platform:
  Hardware provided with preconfigured Linux solution (based on Torque)
- Up to 20 days Long Computation !
  ➔ *Need for extra computational power to reduce computation time*
- Many Windows Desktops are Available
  ➔ *Need for a dynamic and multi-OS solution*

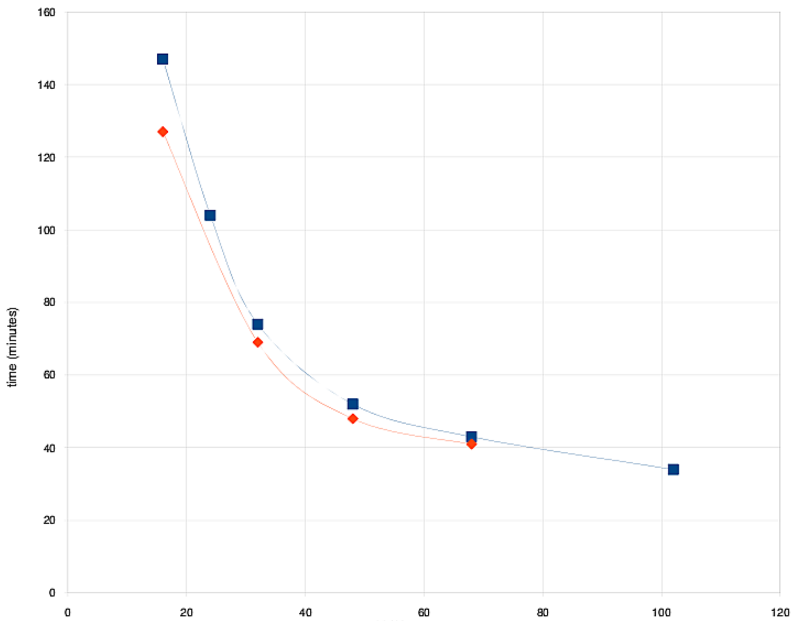
Resources set up

SOLID machine from Applied Biosystems



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)

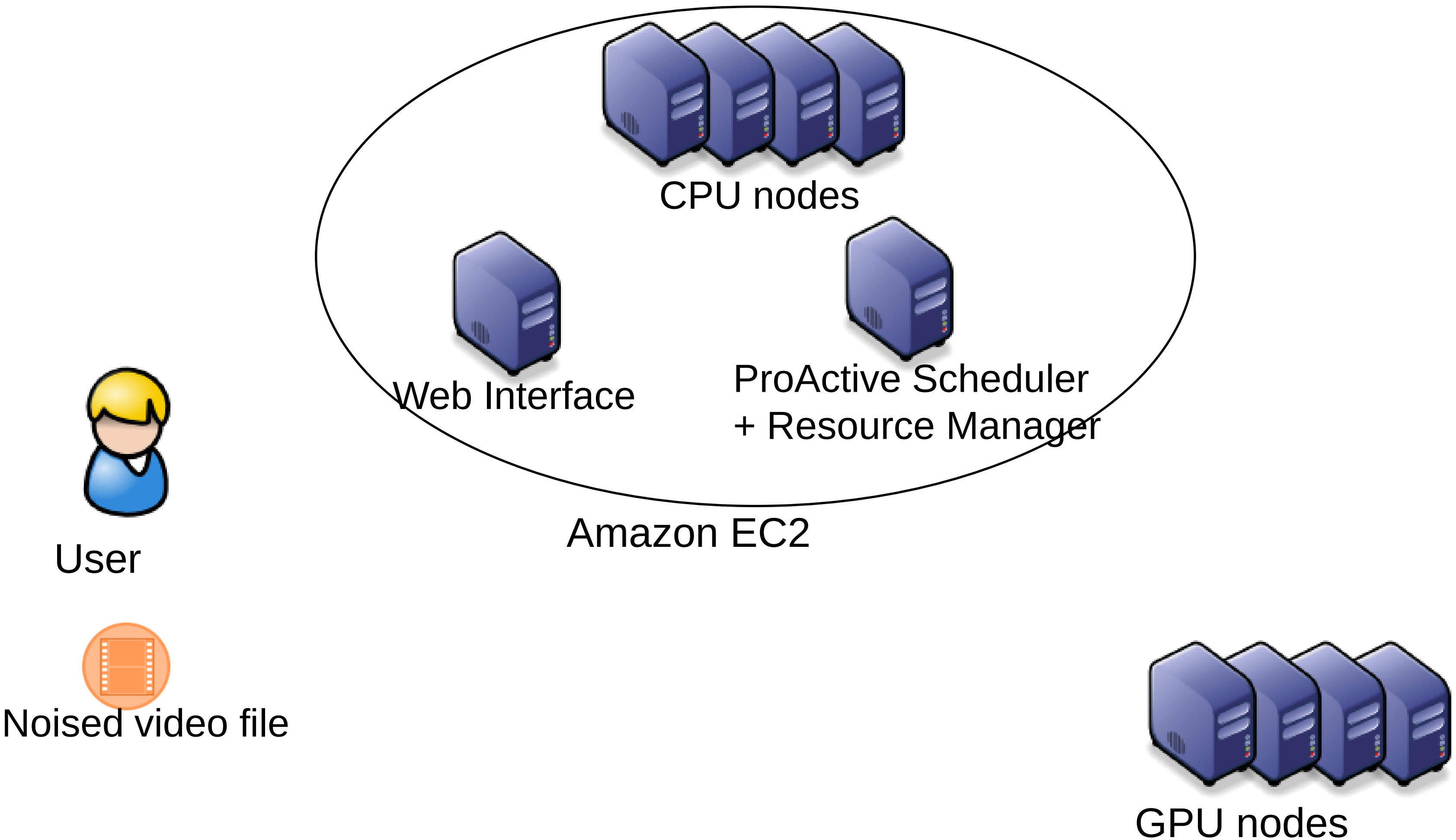
4. Cloud Seeding

Cloud Seeding with ProActive

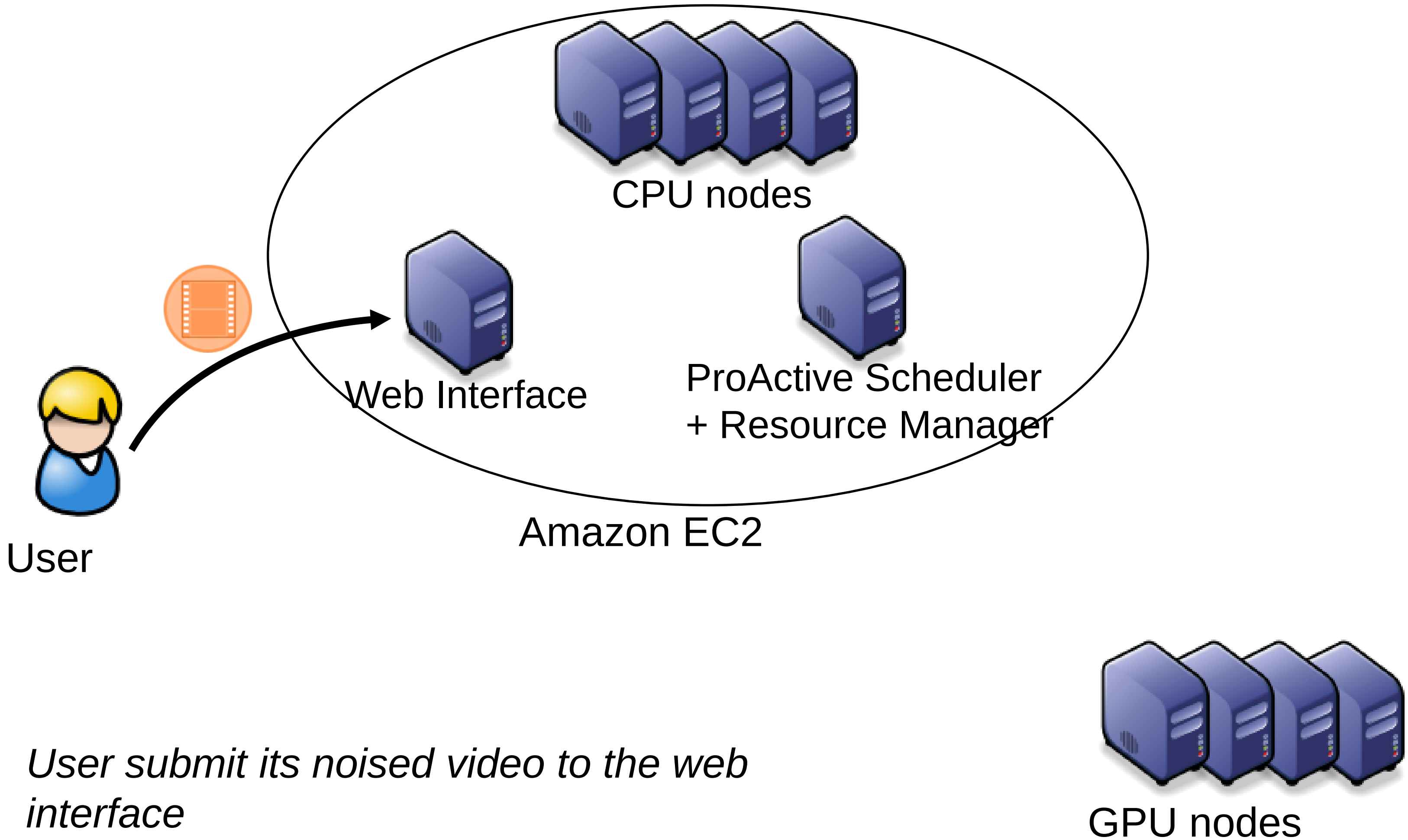
- Amazon EC2 Execution
- *Cloud Seeding* strategy to mix heterogeneous computing resources :
  - External <u>GPU resources</u>

ActiveEon SCALE BEYOND LIMITS · ProActive Parallel Suite · OW2 Consortium

Cloud Seeding with ProActive

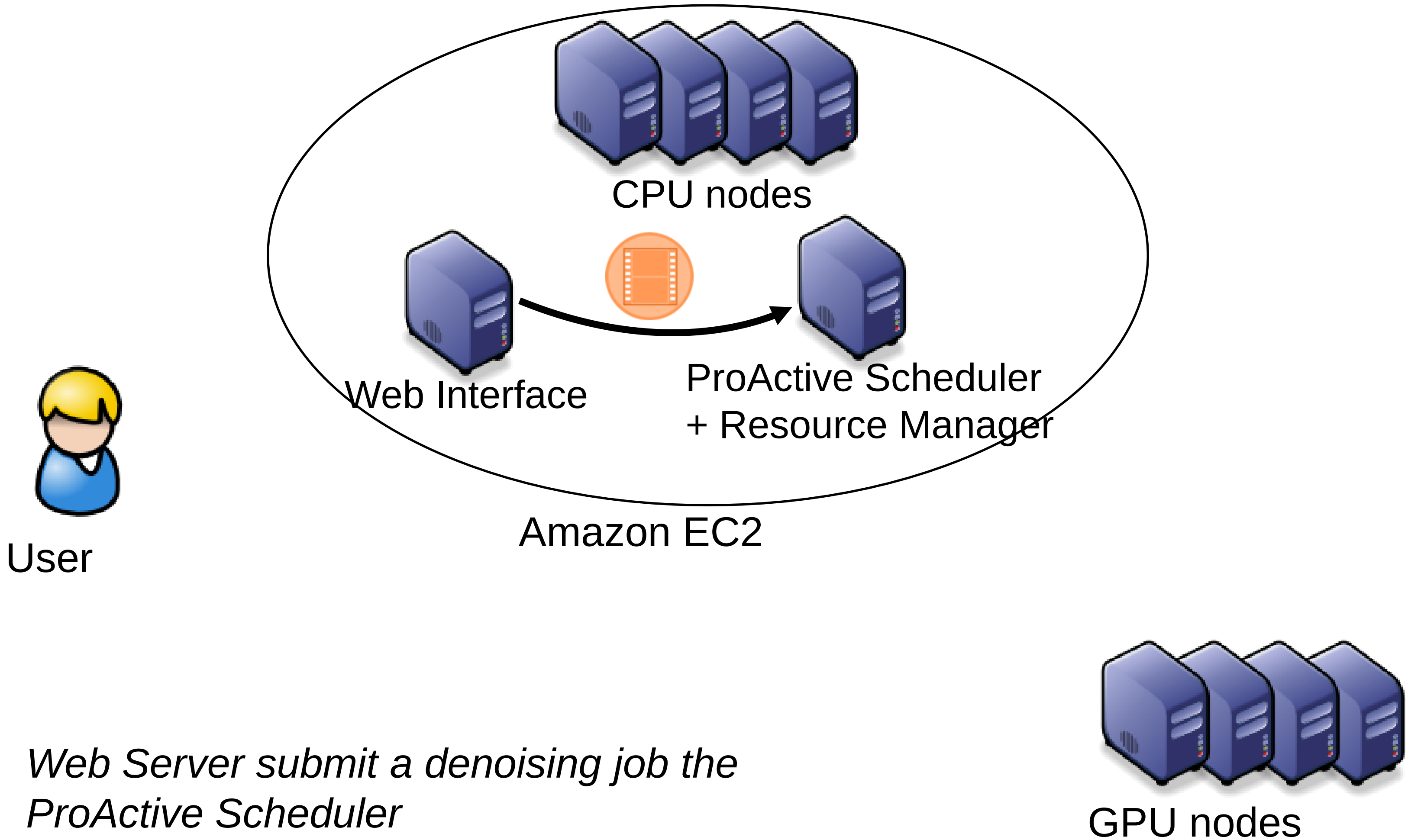


Cloud Seeding with ProActive



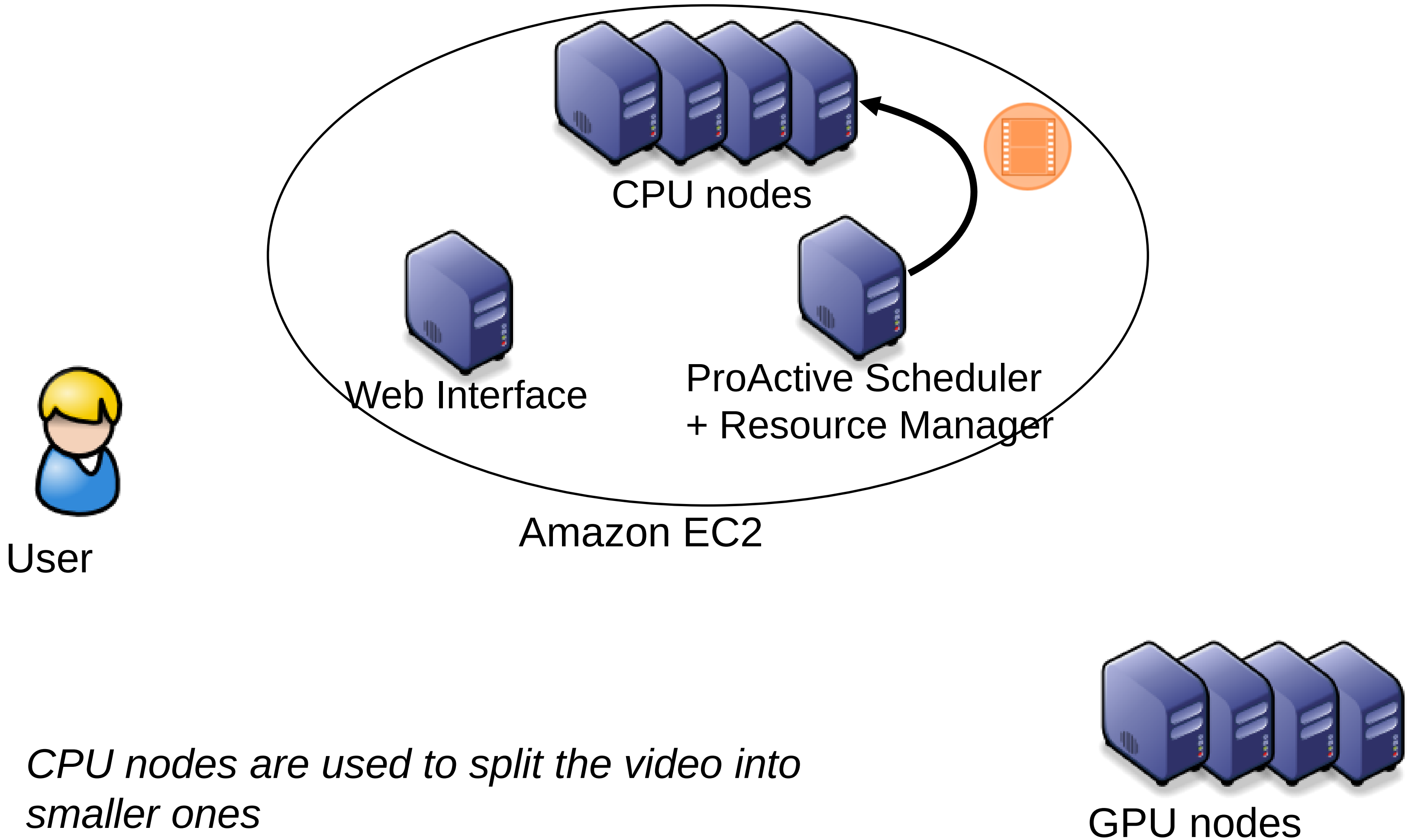
User submit its noised video to the web interface

Cloud Seeding with ProActive



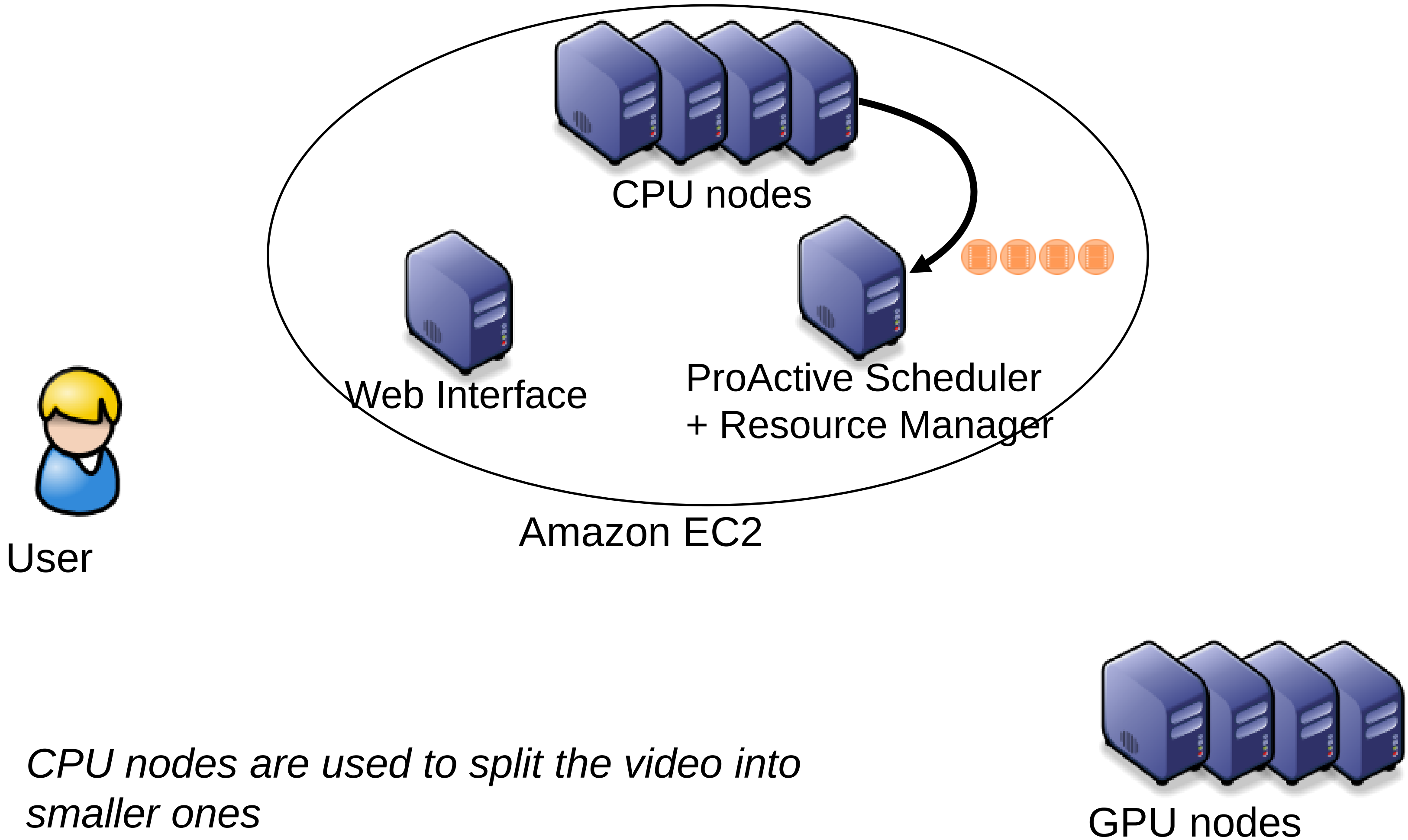
Web Server submit a denoising job the ProActive Scheduler

Cloud Seeding with ProActive



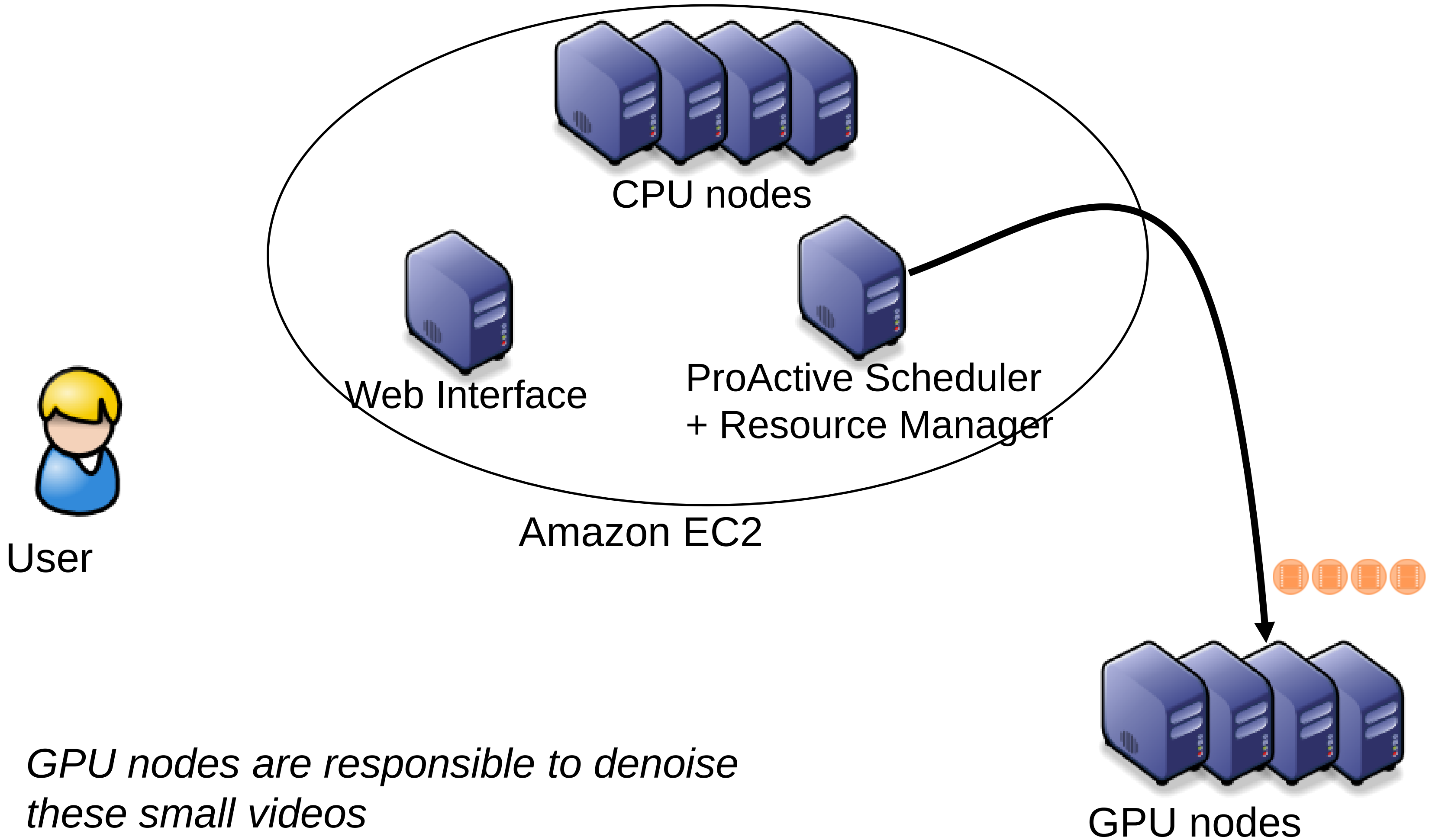
CPU nodes are used to split the video into smaller ones

Cloud Seeding with ProActive



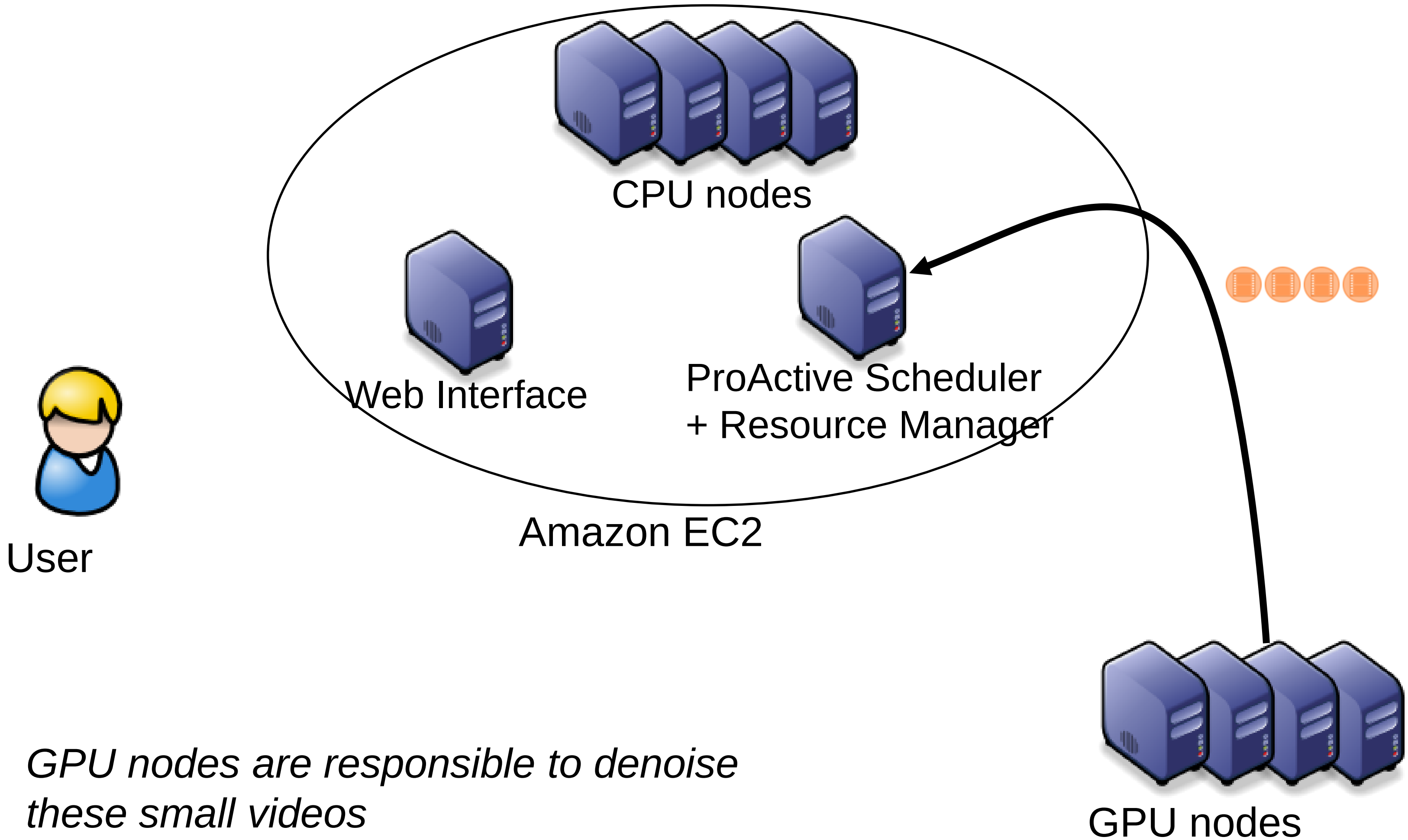
CPU nodes are used to split the video into smaller ones

Cloud Seeding with ProActive



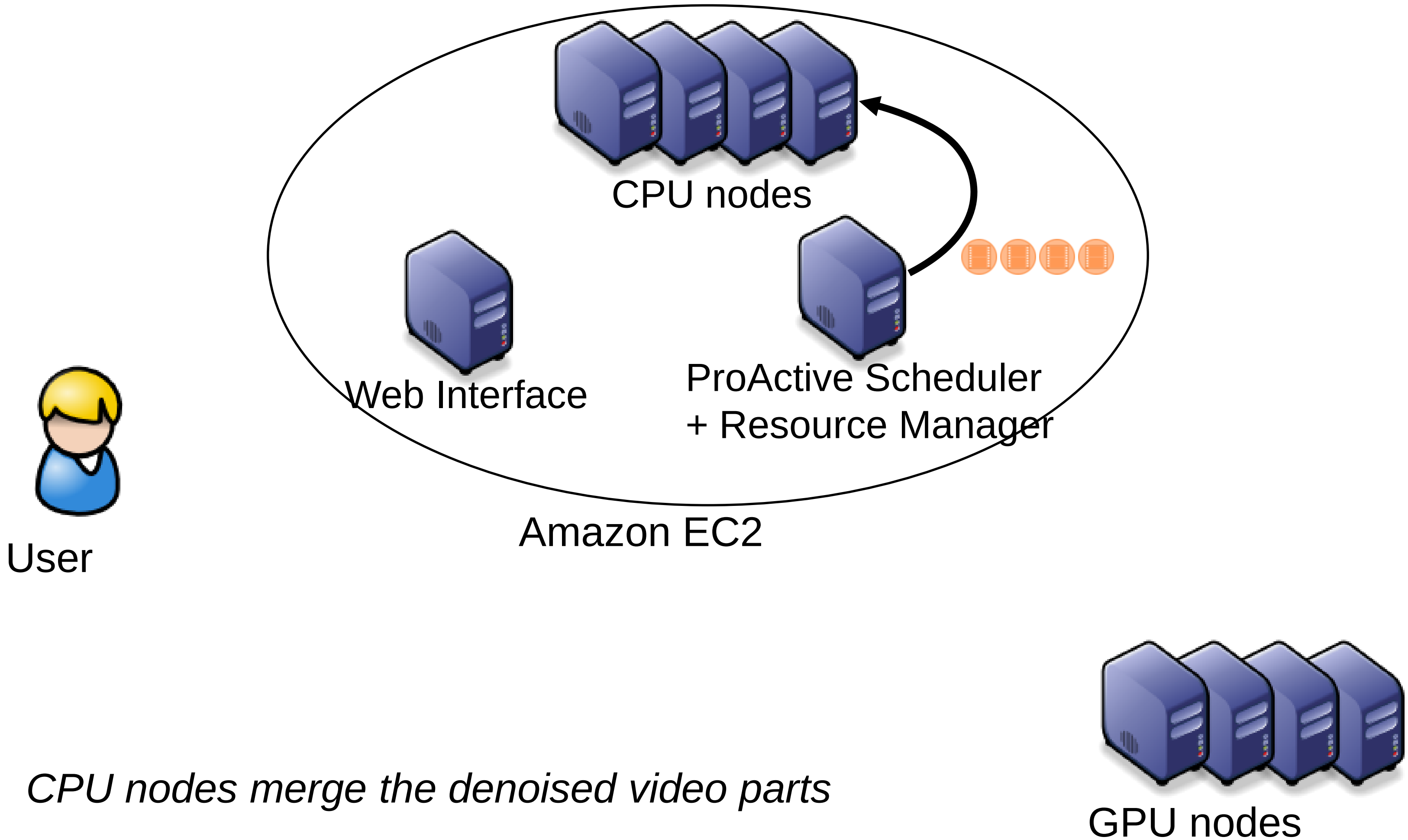
GPU nodes are responsible to denoise these small videos

Cloud Seeding with ProActive



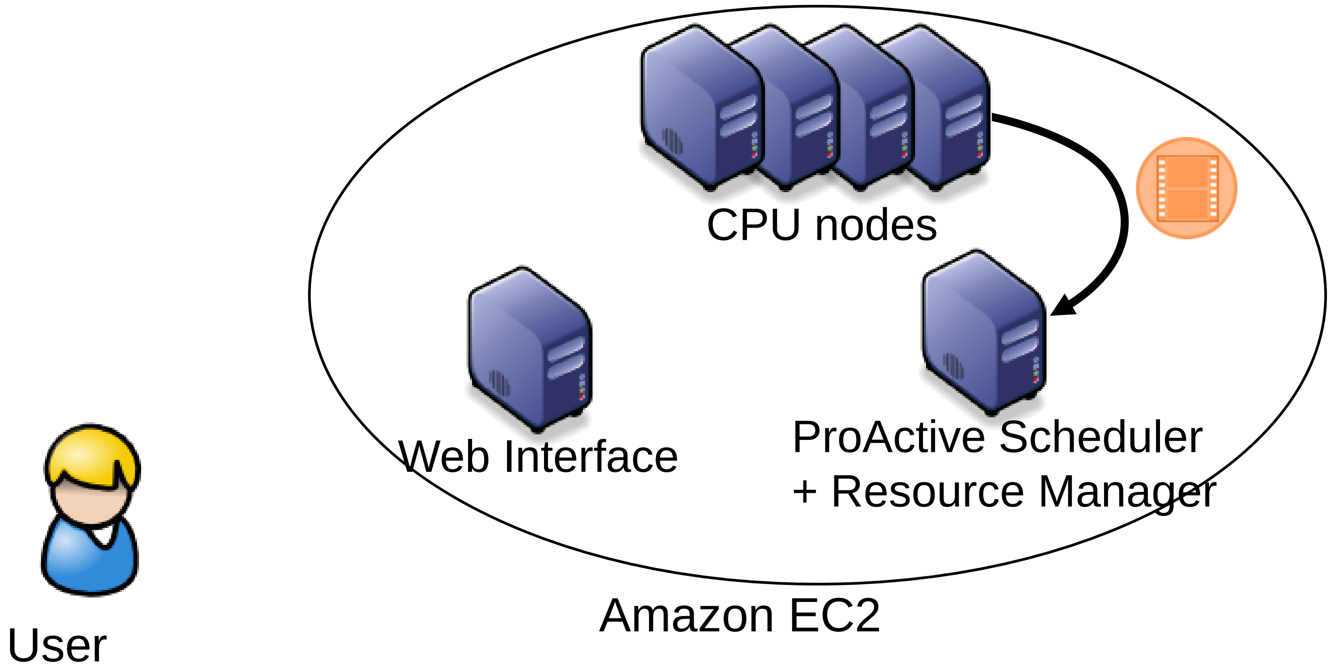
GPU nodes are responsible to denoise these small videos

Cloud Seeding with ProActive

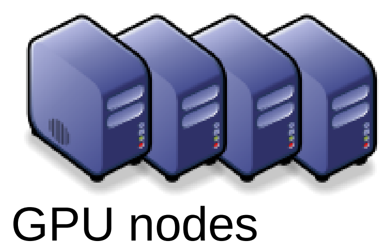


CPU nodes merge the denoised video parts

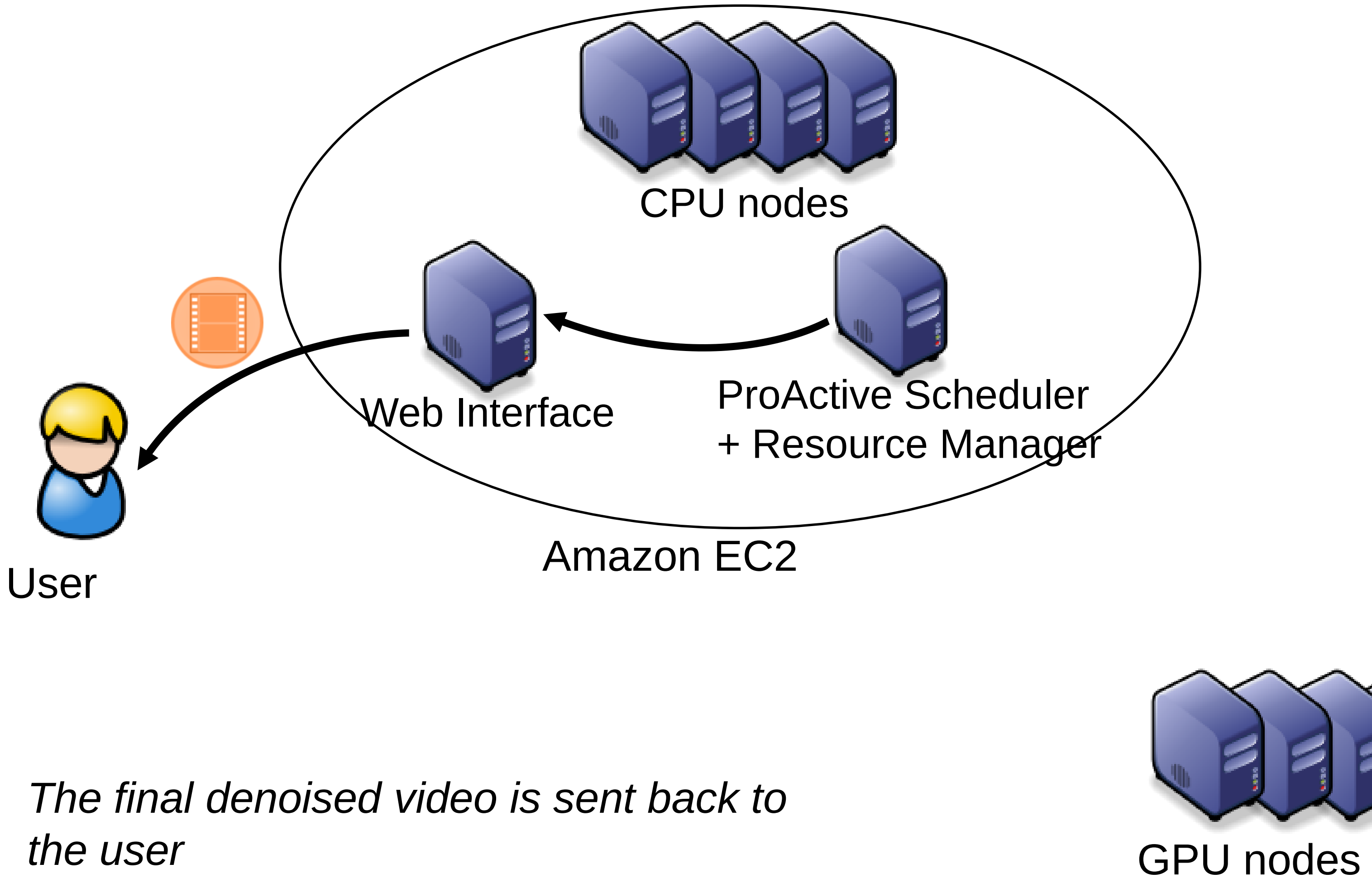
Cloud Seeding with ProActive



CPU nodes merge the denoised video parts



Cloud Seeding with ProActive



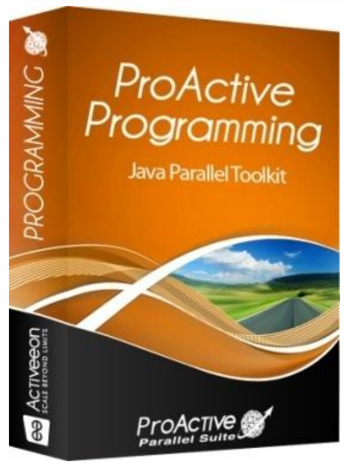
The final denoised video is sent back to the user

Conclusion

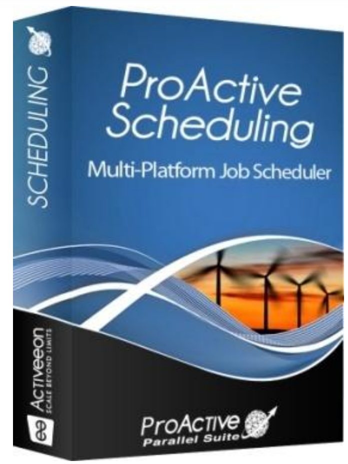
Conclusion

ProActive Parallel Suite

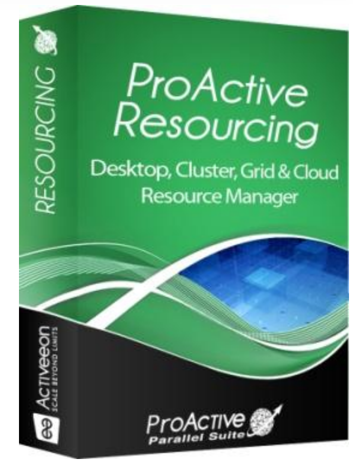
- **Flexibility**
  Clutch Power
- **Portability:**
  Windows, Linux, Mac
- **Versatility:**
  Desktops, Grids, Clouds



Java Parallel Toolkit



Multi-Platform Job Scheduler



Resource Manager

Free Professional Open Source Software

ProActive.inria.fr

Multi-Core: No sharing Parallel Programming Model

Cloud: Smooth transition needed (Interop)

We removed VO, but we Hype the same dreams!!

Danger: same KO than experienced with Grid

Lets be pragmatic!

ActiveEon SCALE BEYOND LIMITS

ProActive Parallel Suite

74

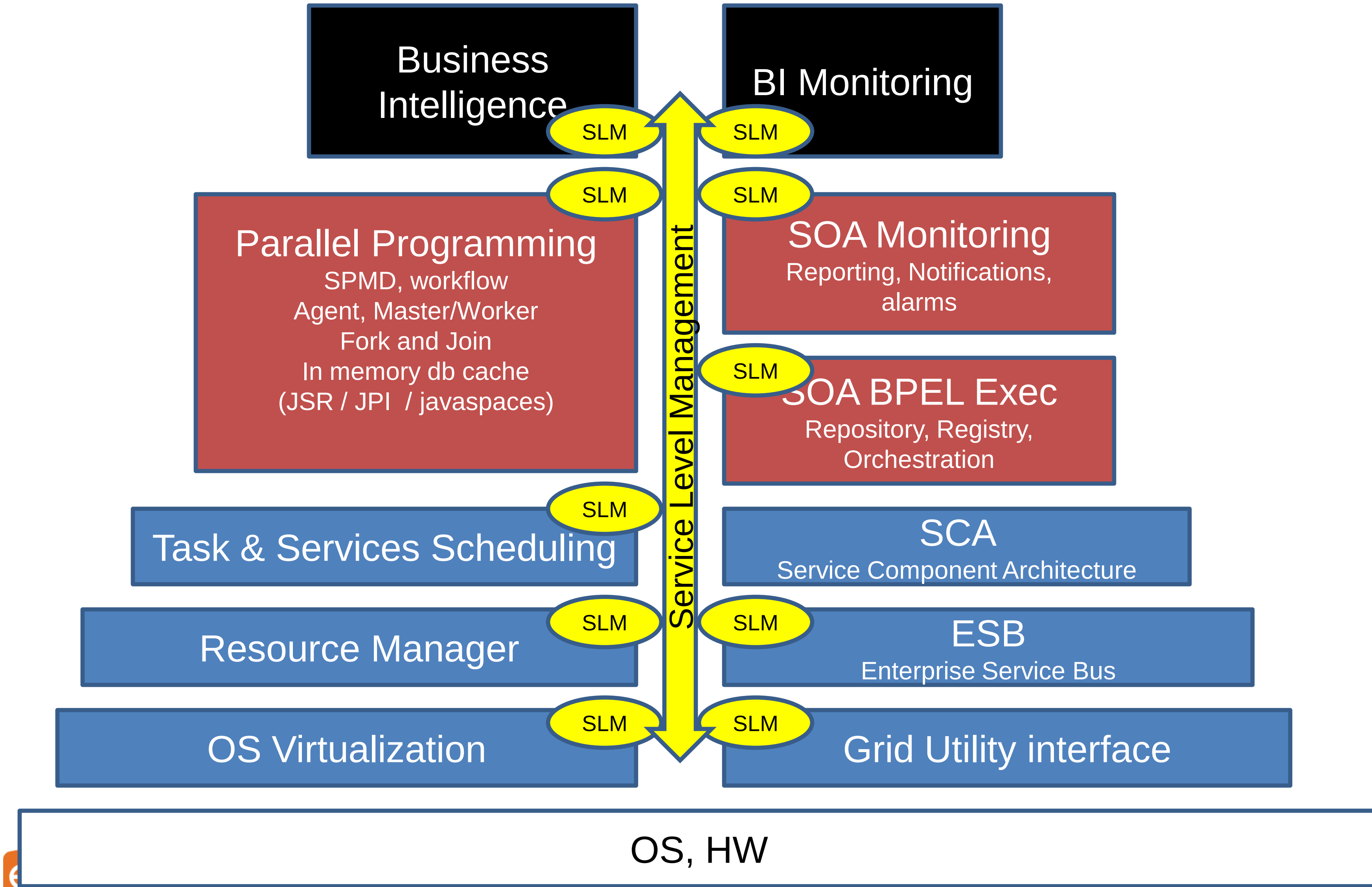
AGOS: Grid Architecture for SOA

Building a Platform for Agile SOA with Grid

- AGOS Solutions

In Open Source with Professional Support

AGOS Generic Architecture for Autonomic SOA with GRIDs & Clouds



SCALE BEYOND LIMITS

Parallel Suite

Key Point: Software Evolution

- ❑ Distributed To Multicores
- ❑ Multi-Cores: 32 (2010) to 64 to 128 to 256 (2014)

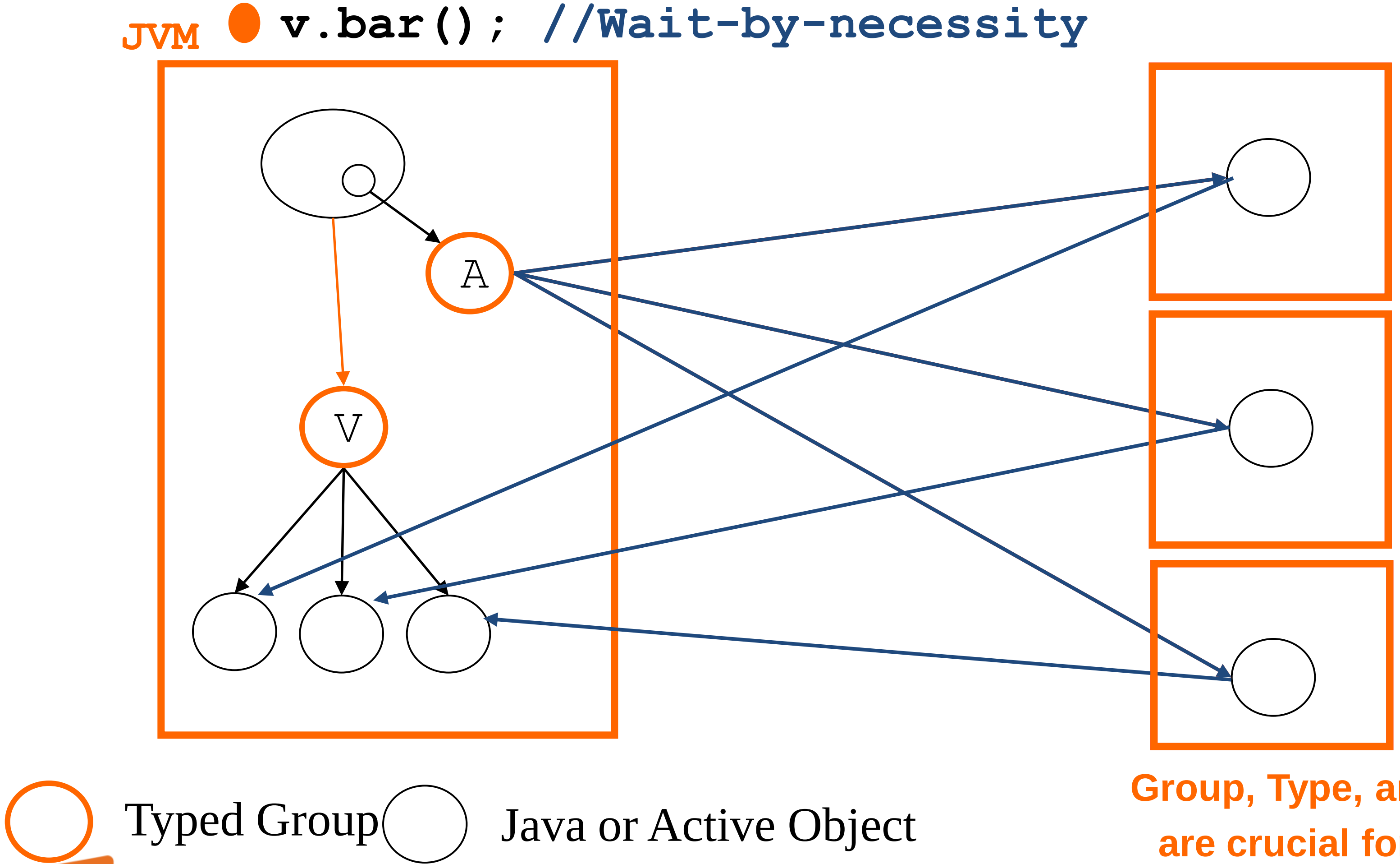
Shift the execution from several multi-cores executing the same application simultaneously to a single, larger multi-core chip.

An application requiring 128 cores to correctly execute, can be executed in 2012 on four 32 cores, and seamlessly executed in 2016 on a single 128-core chips

➔ Smooth evolutivity of applications:
Distributed and Multi-core Platforms

Creating AO and Groups

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



Group, Type, and Asynchrony are crucial for Composition

GCM Standardization

Fractal Based Grid Component Model

World Class Standards

4 Standards:

1. GCM Interoperability Deployment
2. GCM Application Description
3. GCM Fractal ADL
4. GCM Management API

Key Points about Parallel Components

- Parallelism is <u>*captured*</u> at the Module <u>*interface*</u>
  Identical to Typing for functional aspects
- <u>*Composition*</u>, parallel word, becomes possible
- <u>*Configuration*</u> of the Parallel *aspects*