Research in the OASIS project-team:

Designing, programming, and verifying distributed systems

www.inria.fr/oasis

Eric Madelaine

The OASIS team (Apr. 2013)

Common team between:

INRIA centre de Sophia Antipolis Méditerranée

I3S Lab { **Université** de Nice Sophia Antipolis (**U.N.S.**)
CNRS

6 Permanent researchers:

Madelaine E.	**CR INRIA**	(Dr, Hdr, head)
Henrio L.	**CR CNRS**	(Dr, Hdr)
Baude F.	**Prof. U.N.S.**	(Dr, Hdr)
Dalle O.	**Ass. Pr. U.N.S.**	
Hermenier F.	**Ass. Pr. U.N.S.**	
Huet F.	**Ass. Pr. U.N.S.**	

Caromel D. **Prof. UNS** (ext. advisor)

10 (software) engineers

5 master interns

9 on-going PhDs

1 in 3rd year

FP7 PLAY	**France**

3 in 2nd year

BDO PACA / Oseo	**France**
CIFRE ActiveEon	**Portugal**
Cotutelle ECNU Shanghai	**China**

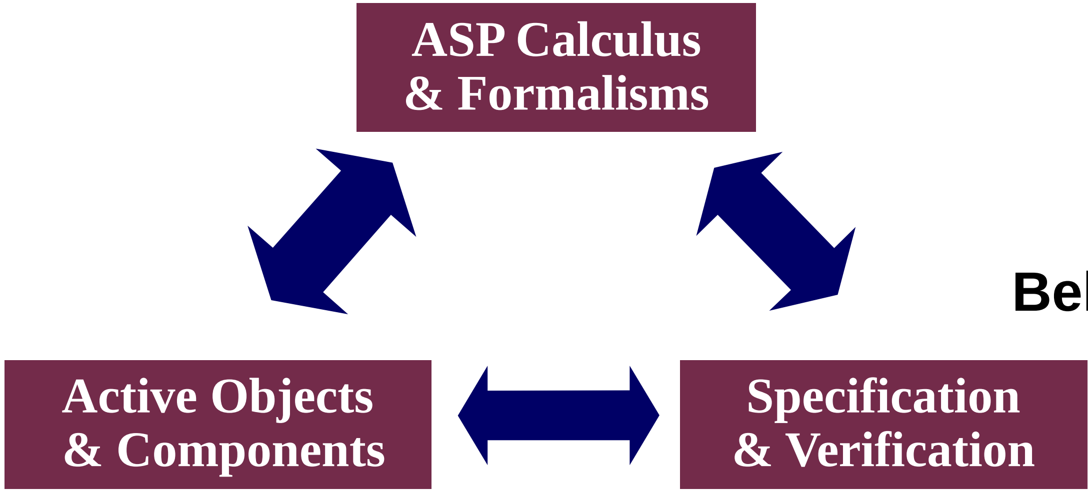
4 in 1st year

MENRT	**France**
FUI CloudForce	**Romania**
Cotutelle Carleton U.	**Canada**
Ecole Centrale Pekin	**China**

OASIS: Research Directions

Programming models

Formalisation



Middleware

Behavioral semantics

Programming and Runtime support

VerCors platform

A cut on Theory and Applications of Distributed Components

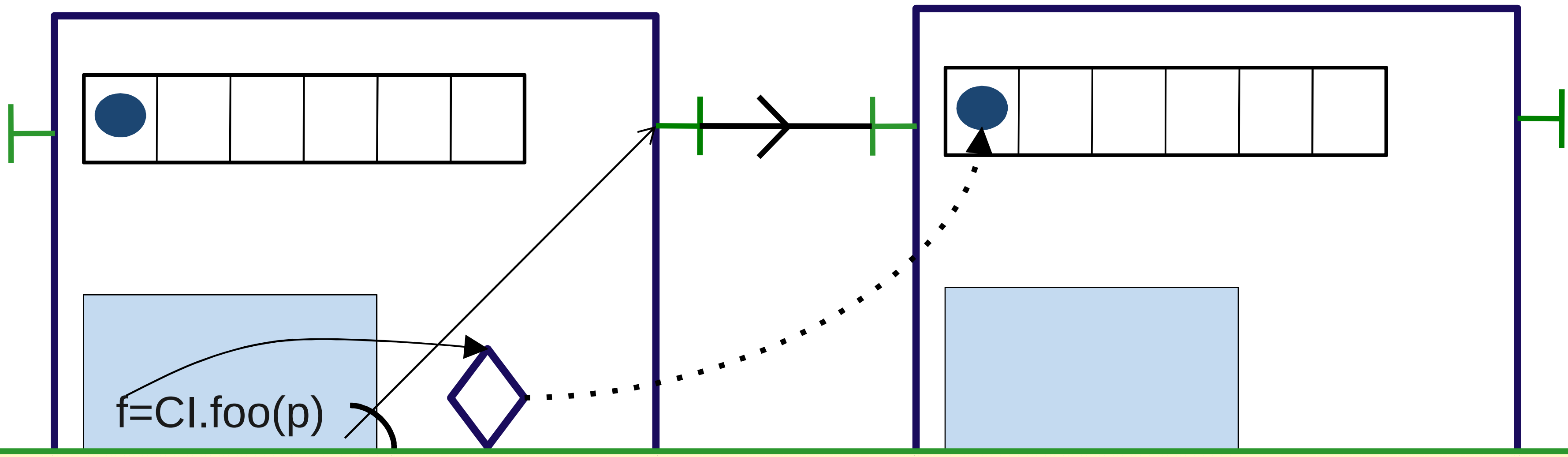
Grid Component Model (GCM):

definition within CoreGrid & GridComp EU (FP6) projects

standardization at ETSI (2008 – 2010)

- ➔ **GCM : execution model**
- ➔ **Using GCM for autonomicity and for high performance computing**
- ➔ **Formalization and Verification**

A Primitive GCM Component handled by a ProActive active object



➔ **Components abstract away distribution and *concurrency***

In ProActive reference implementation, components are mono-threaded
➔ **simplifies concurrency** but subject to more **deadlocks**

Multi-active Objects – preliminary ideas

Mono-threaded active objects have limitations:

- Deadlocks
- Inadapted to multicore architecture (no shared memory)

Solutions in the active-object community: JCoBox, JAC, X10

Our proposal, a programming model that mixes local parallelism (with limited shared memory) and distribution with high-level programming constructs

- Challenge:
  A programming model addressing highly heterogeneous resources, Multicores, GPUs, clusters, clouds…

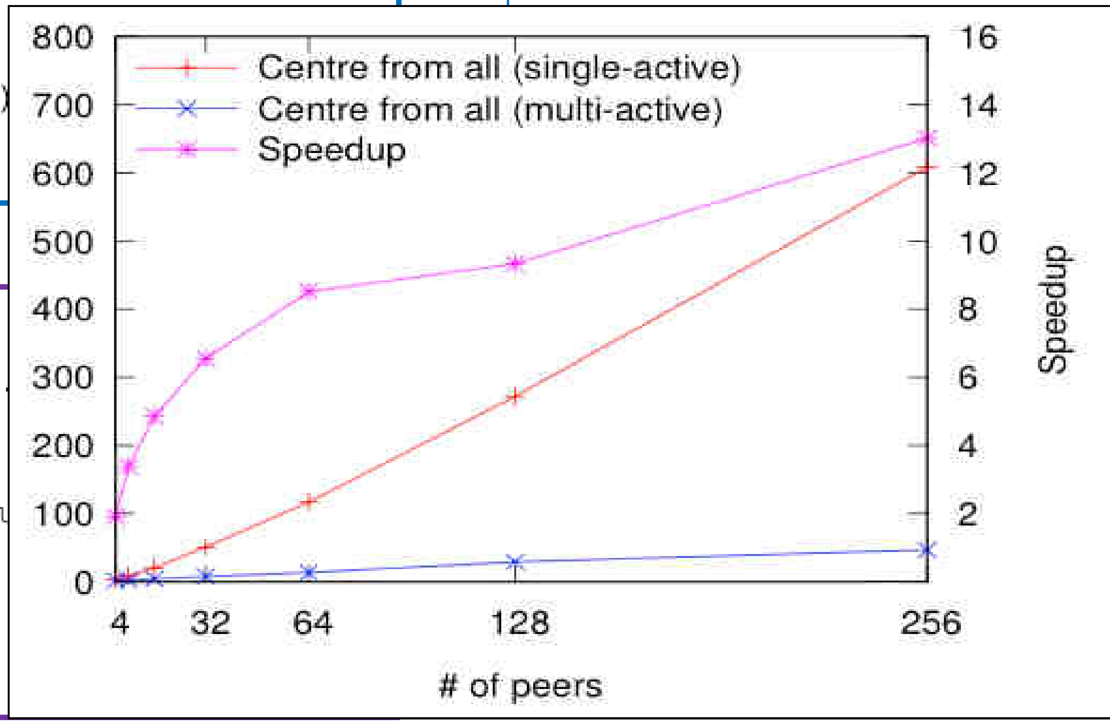
Ongoing work

Multi-active Objects – Annotation Example

```
@DefineGroups({
  @Group(name="join", selfCompatible=false)
  @Group(name="routing", selfCompatible=true)
})
```

Groups

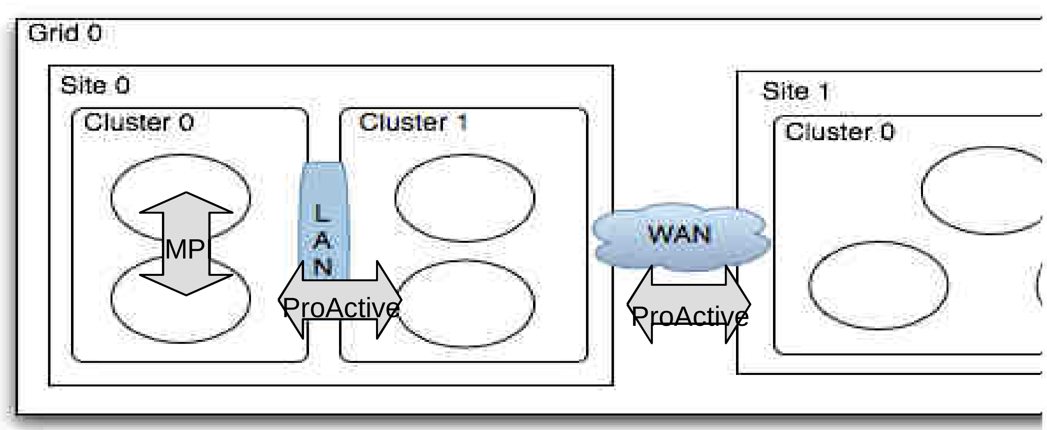
```
@DefineRules({
  @Compatible({"join", "monitoring"})
  @Compatible({"routing", "monitoring"})
})

public class Peer {
...
@MemberOf("join")
  public JoinResponse join(Peer other) { .

@MemberOf("routing")
  public void add(Key k, Serializable valu

@MemberOf("routing")
  public Serializable lookup(Key k) { ...
}
```

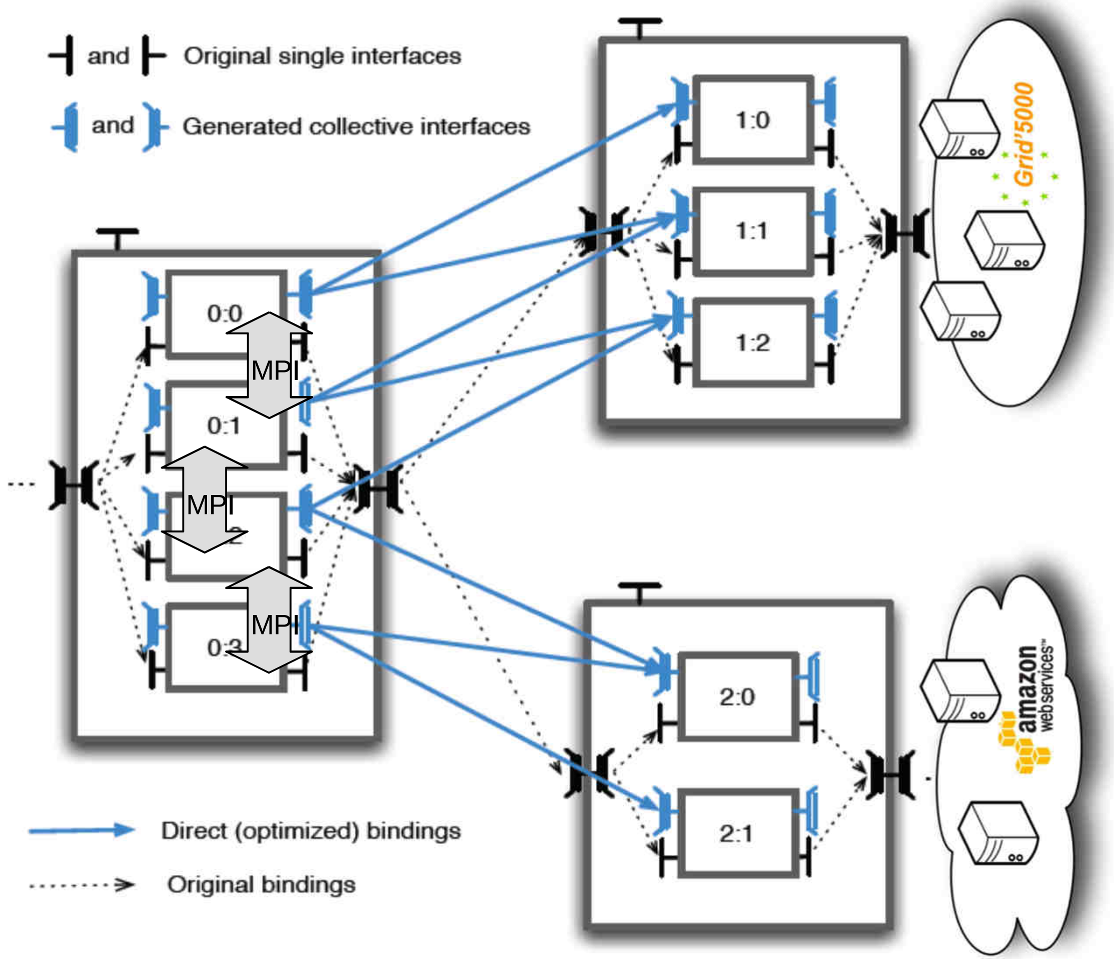


Adapting Active Objects to Multicore Architectures -
Ludovic Henrio, Fabrice Huet, Zsolt István, and Gheorghen Sebestyén - *ISPDC 2011*

Achievements

A component-based support for MPI-like hierarchical applications

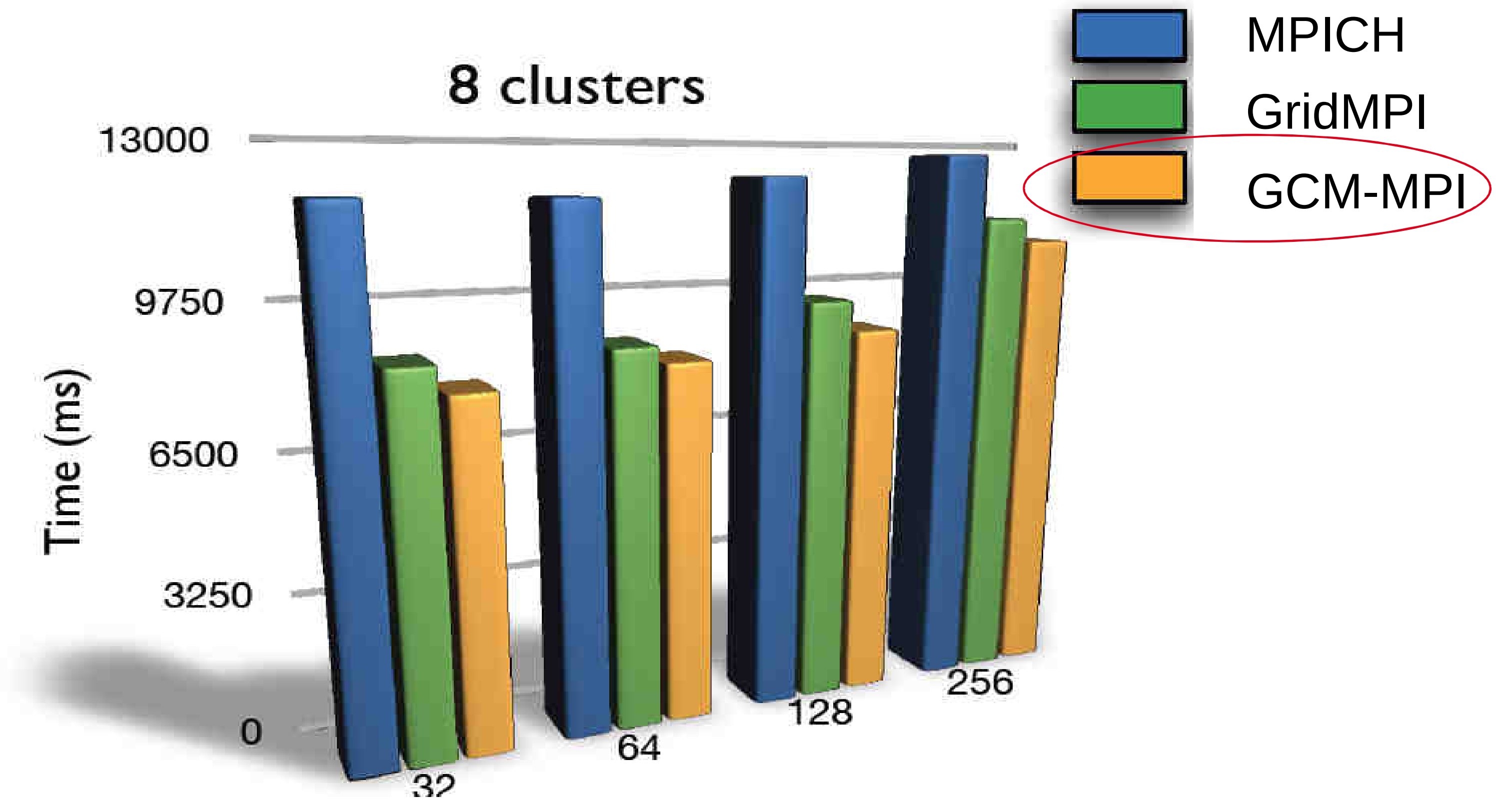




- Each MPI-like process is a primitive GCM cpt.
- Many to many GCM Bindings between composite cpts.
- Transparent Optimized inter-cluster Communication

E. Mathias, V. Cavé, S. Lanteri, F. Baude "**Grid-enabling SPMD Applications through Hierarchical Partitioning and a Component-Based Runtime** " – *EuroPAR 2009*

All-Reduce benchmark

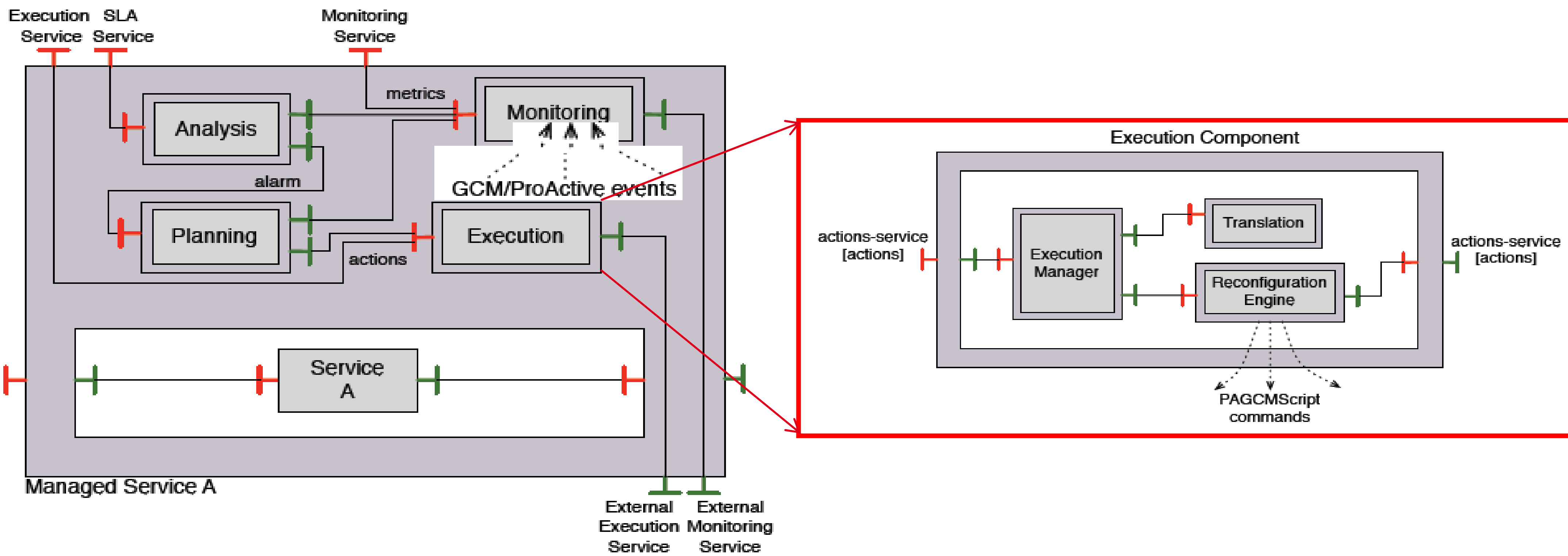


- **Competitive overall performances for a Complete SPMD (with collective operations) numerical simulation, compared to MPI on Grids.**
- **SPMD non-embarrassingly parallel applications on federated Grids & Clouds**

E. Mathias, F. Baude "**A Component-Based Middleware for Hybrid Grid/Cloud Computing Platforms** " – *Concurrency and Computation: Practice and Experience, 2012*

Autonomic computing support in GCM

- **Generic Framework** for a Monitoring Analysis Planning Execution (MAPE) control loop plugged to GCM components
- Possible personnalisation, e.g. using Jboss Drools for Analysis/Planning



C. Ruz, F. Baude, B. Sauvan, "**Using Components to Provide a Flexible Adaptation Loop to Component-based SOA Applications**" – *Int. Journal on Advances in Intelligent Systems, May 2012*

Formalisation and Verification

Machine-assisted Formalisation and Proofs:

Ex. of results for Future update strategies**:**

- Proved that all strategies are equivalent
- Formalised one strategy in Isabelle

Example of result:

- Complete registration: "All Future references are registered during reduction"

Provides generic properties of the models/languages

Strong guide for implementation

Model-checking approach**:**

Provide tools to prove Behavioural Properties: Deadlock freeness, progress, safety and liveness.

➢ **VerCors** platform

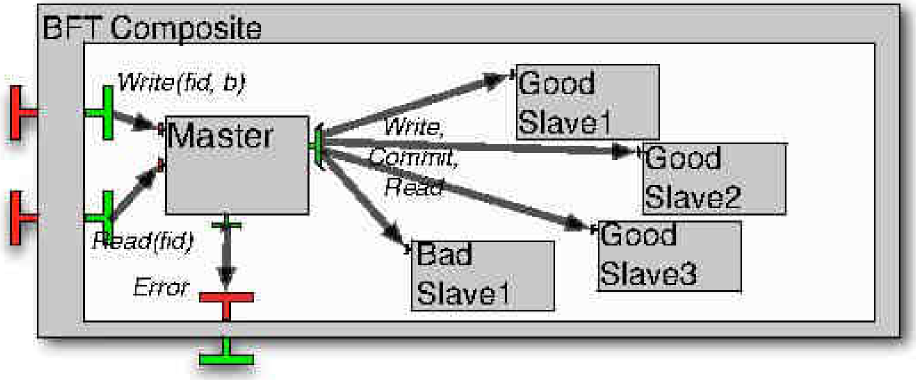
- MDE-oriented **specification formalisms**
- Abstract semantic models
- State-of-the-art **model-checking** tools

Specification & Verification environment

Application-specific properties

Example: BFT protocol proved safe

A distributed application: the Byzantine Fault Tolerant protocol, for ensuring replicated read/write consistency.



Properties:
read/write operations

Proved with 3f+1 proce
within a few hours us

Estimated brute force st

Difficulty:
Master state explosion by combination of:
Data abstraction;
Compositional & context dependent modelling;
Distributed model-checking;

R. Ameur-Boulifa , R. Halalai , L. Henrio , and E. Madelaine. "***Verifying Safety of Fault-Tolerant Distributed Components*** - *FACS, 2011*

Verification of parameterized and dynamic systems.

Architecture Language (ADL) for parameterized topologies; dynamic ADL

- More expressive and closer to the application logics
- Strongly needed for reconfigurable or autonomic applications
- Exploit directly the power of the pNet semantic model

➢ Challenge:

mix model-checking (MC), theorem-proving (TP), and run-time verification (RTV)

techniques to master the proof techniques for parameterized and dynamic models:

- Use TP to build higher-level (= smaller) models for MC
- Use TP to prove correctness of abstraction and of generalization
- Use RTV to prove preservation of model properties during system reconfiguration
- …

Softwares

Proactive (distributed by ActiveEon) (OW2)

- Programming
- Cloud portal: workflow studio, scheduler, resource manager (see http://proactive.inria.fr/pacagrid/)
- http://proactive.inria.fr/

VerCors

- Specification formalisms (Eclipse-based)
- Partial code generation : ADL and java class skeletons
- Bridges to CADP engines
- http://www-sop.inria.fr/oasis/index.php?page=vercors

OSA (INRIA ADT)

- Open Simulation Architecture
- Methodology and tools
- http://osa.gforge.inria.fr/

Collaborations

ANR: 4

European projects: 3

Industrial contracts: 3

Budget: ~800 Keuros/year

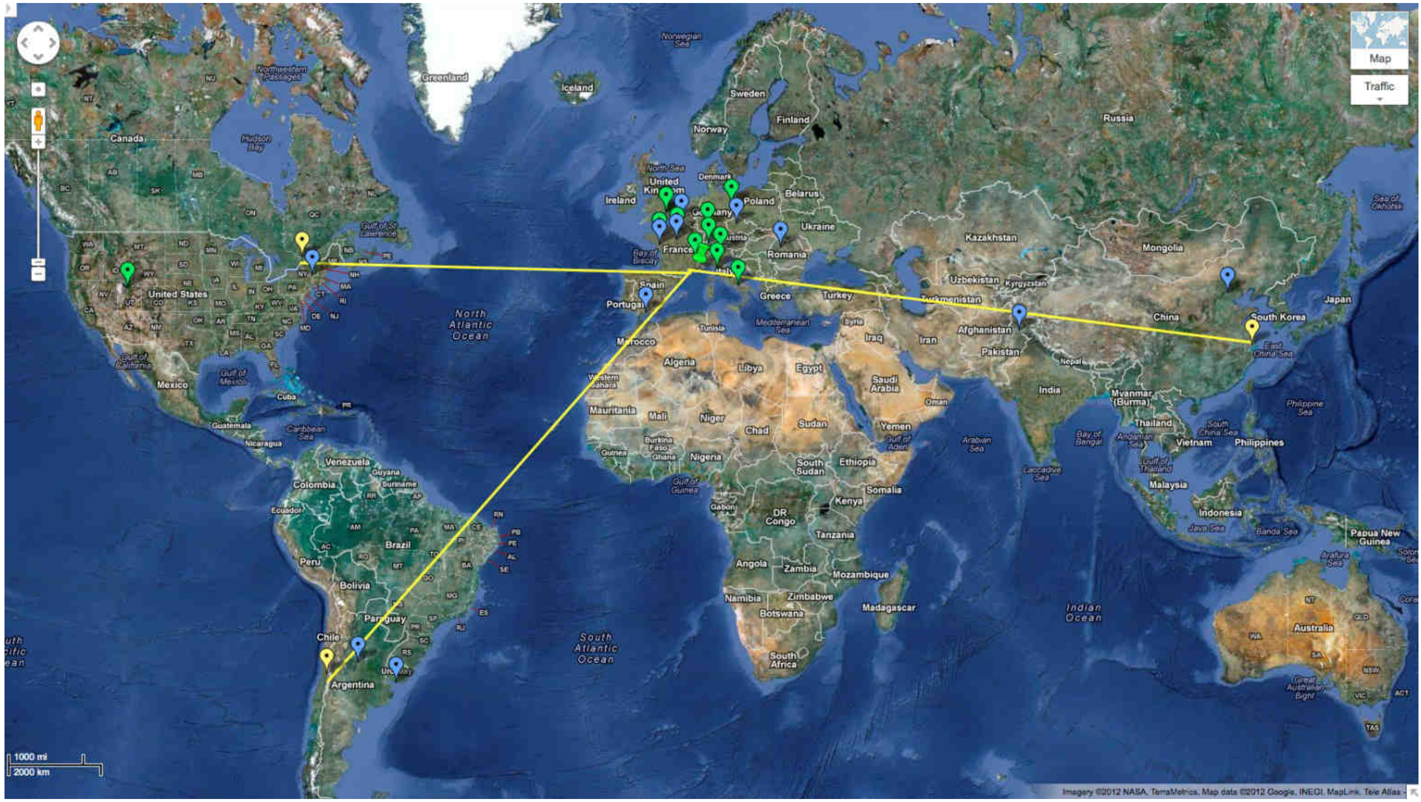
3 associated teams :

Shanghai, Santiago de Chili, Ottawa

Main (informal) collaborations:

INRIA: Adam, Sardes, Ascola, Convex, Tosca

others: U. Pisa, U. Sannio, T.U. Berlin, U. Middlesex, U. of Utah, Charles U. (Prague)



- Associated team
- Other common publication
- Other collaboration

Dissemination: Startup company

- **Current Status:**
  - ➔ Created November 5th 2007 by Pr. Denis Caromel
  - ➔ Now about 15 employees
  - ➔ Open Source Editor for ProActive Parallel Suite and Services around the technology
  - ➔ Technology Transfer and Collaboration contracts with INRIA/University of **Nice**

谢谢

Merci

Thank You

Eric Madelaine

- Eric.Madelaine@inria.fr
- http://www-sop.inria.fr/members/Eric.Madelaine

Project-team OASIS

- http://www-sop.inria.fr/oasis