Distributed Financial Computations with ProActive

Grid@Work - Grid Users Group Conference
October 10, 2005

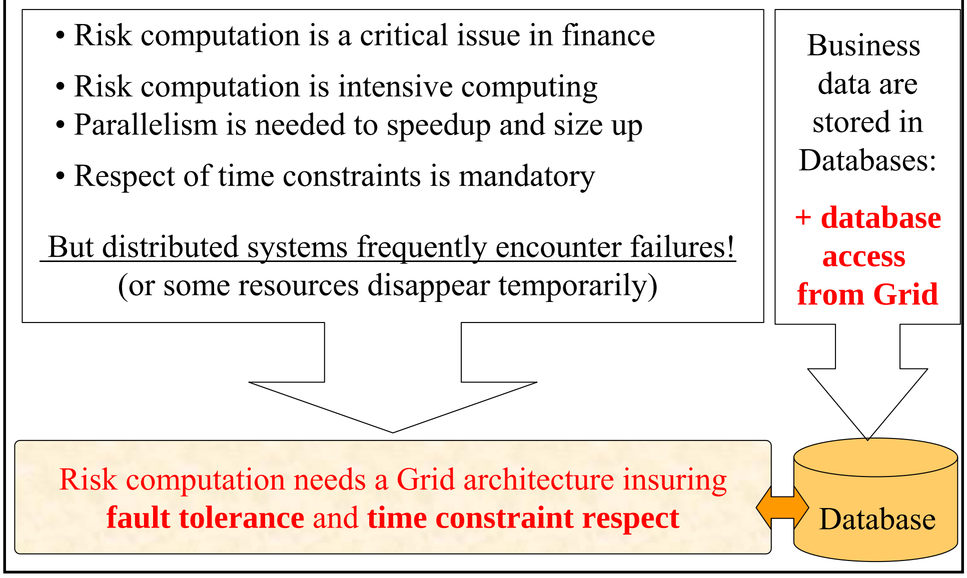
O. Pietquin, F. Sassolas, S. Vialle

SUPELEC
Olivier.Pietquin@supelec.fr
Frederic.Sassolas@supelec.fr
Stéphane.Vialle@supelec.fr

F. Baude, V. Legrand

INRIA, CNRS I3S, UNSA
Francoise.Baude@inria.fr
Virginie.Legrand@inria.fr

1 - Risk Analysis Requirements



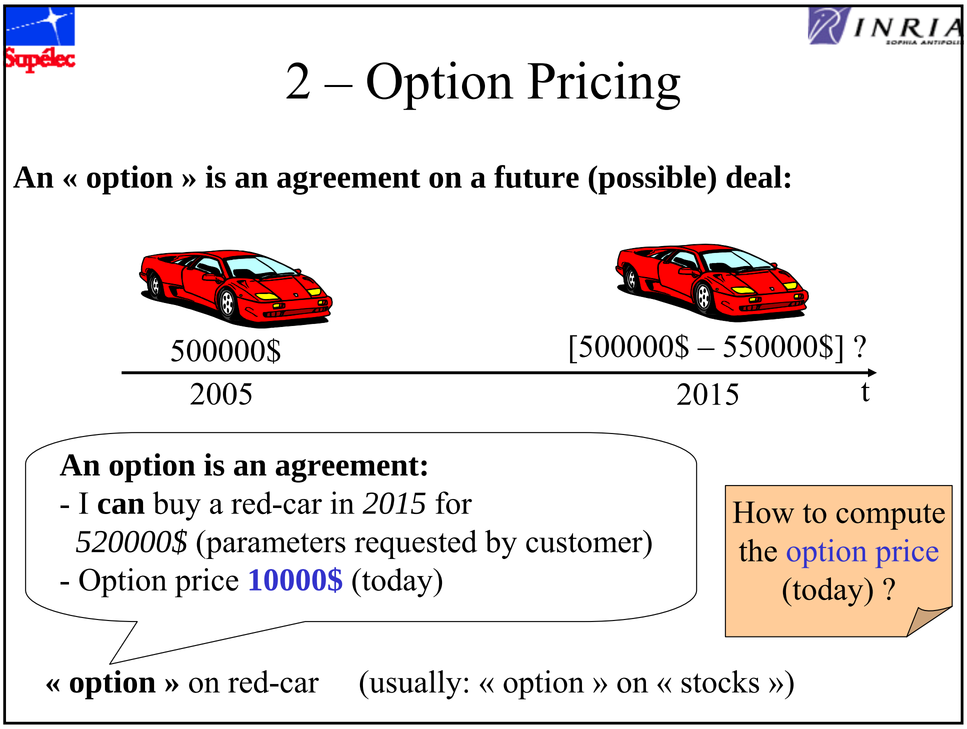
- Risk computation is a critical issue in finance
- Risk computation is intensive computing
- Parallelism is needed to speedup and size up
- Respect of time constraints is mandatory

But distributed systems frequently encounter failures!
(or some resources disappear temporarily)

Business data are stored in Databases:

+ database access from Grid

Risk computation needs a Grid architecture insuring **fault tolerance** and **time constraint respect**



2 – Option Pricing

An « option » is an agreement on a future (possible) deal:

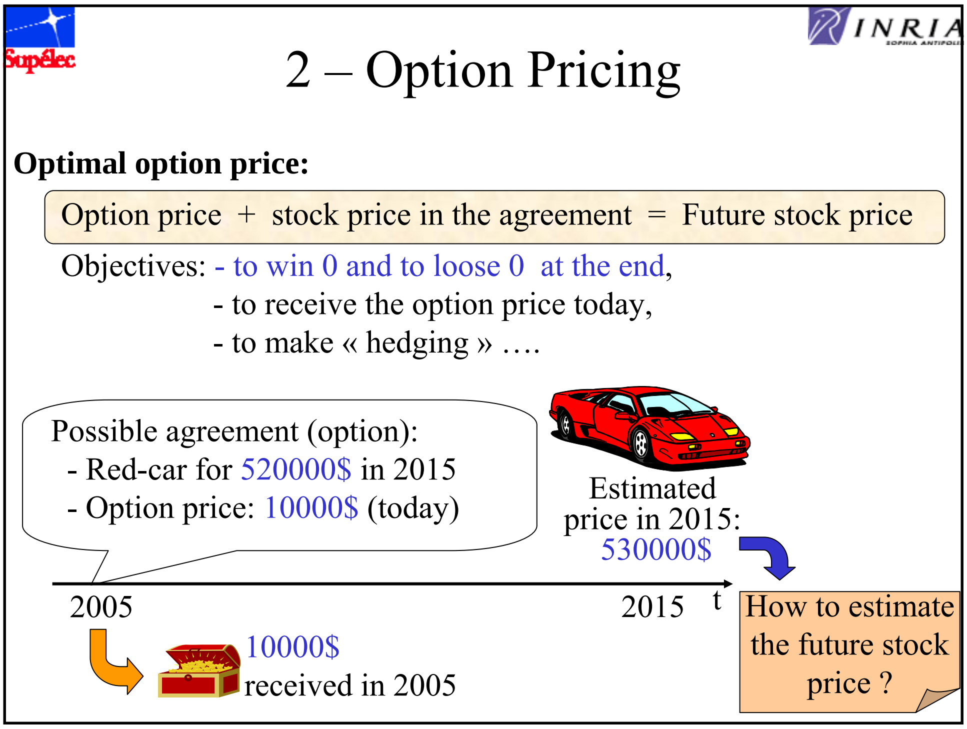
500000$ [500000$ – 550000$] ?

2005 2015 t

An option is an agreement:
- I **can** buy a red-car in *2015* for *520000$* (parameters requested by customer)
- Option price **10000$** (today)

How to compute the option price (today) ?

« option » on red-car (usually: « option » on « stocks »)



2 – Option Pricing

Optimal option price:

Option price + stock price in the agreement = Future stock price

Objectives: - to win 0 and to loose 0 at the end,
- to receive the option price today,
- to make « hedging » ….

Possible agreement (option):
- Red-car for 520000$ in 2015
- Option price: 10000$ (today)

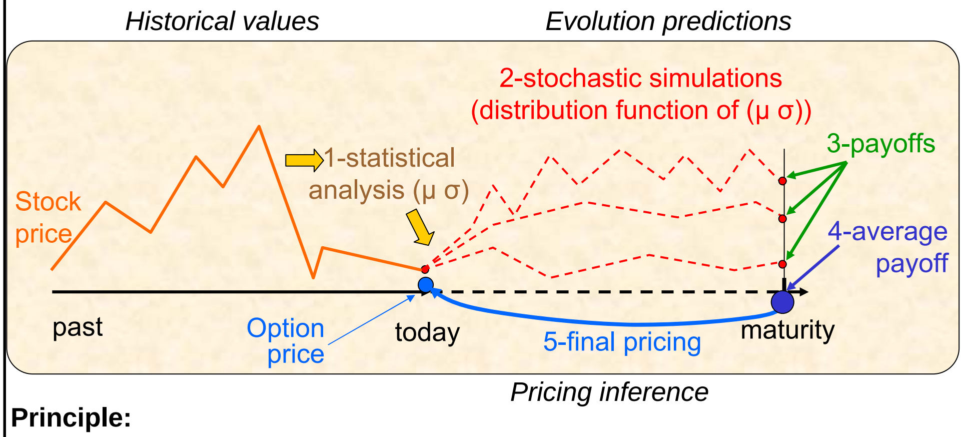
Estimated price in 2015: 530000$

2005 2015 t

10000$ received in 2005

How to estimate the future stock price ?

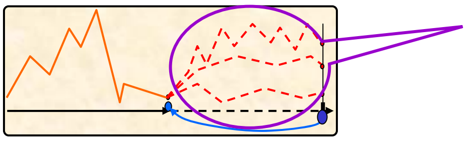
3 – Simulation of action evolution



Principle:

- Simulate many (all) possible evolutions of the stock price,
- Compute all possible payoffs at maturity date (considering the final stock price wished by the customer),
- Compute the average payoff,
- Infer a « good » (optimal) price of the option (today)

3 – Simulation of action evolution



Stochastic computations:

- **Monte-Carlo simulations**,
- Probabilistic trees,
- Other methods ….

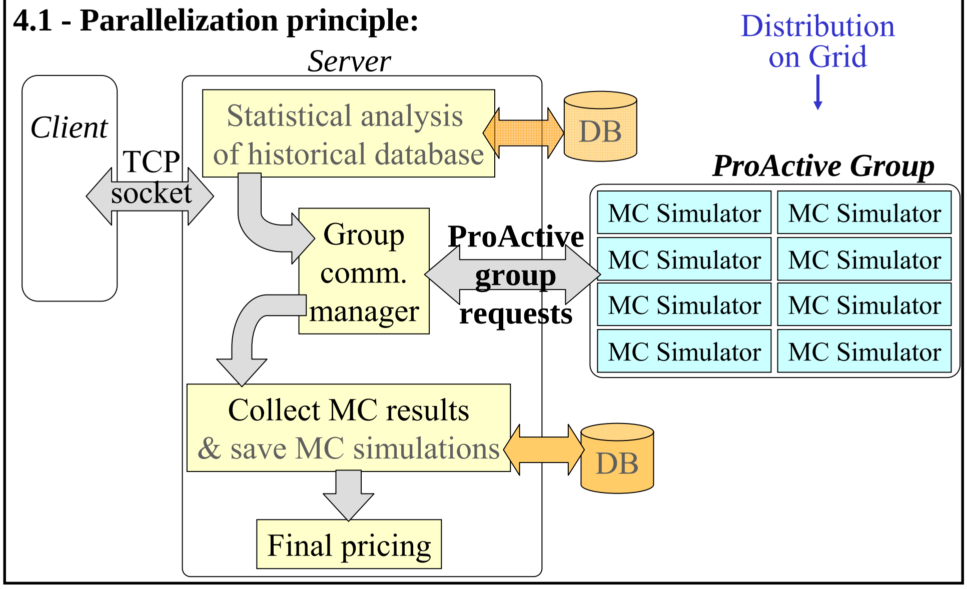
Final requirements:

➔ **Option pricing on Grid**

- Distribute MC simulations
- Include databases in the Grid architecture
- Achieve fault tolerance
- Achieve respect of time constraints
  ➔ reproducible execution time is mandatory

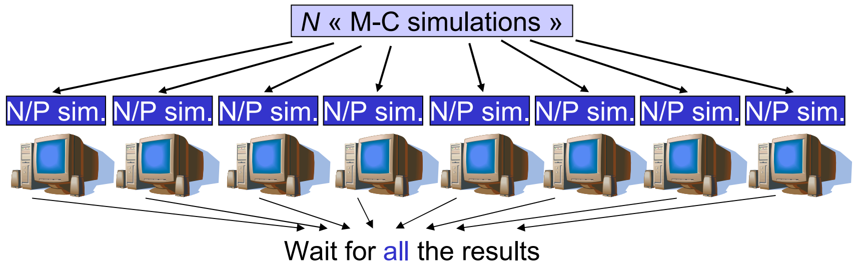
4 – Grid algorithmic strategy

4.1 - Parallelization principle:



4 – Grid algorithmic strategy

4.2 - Static and exact split of the simulation work:



Independent random number generation:

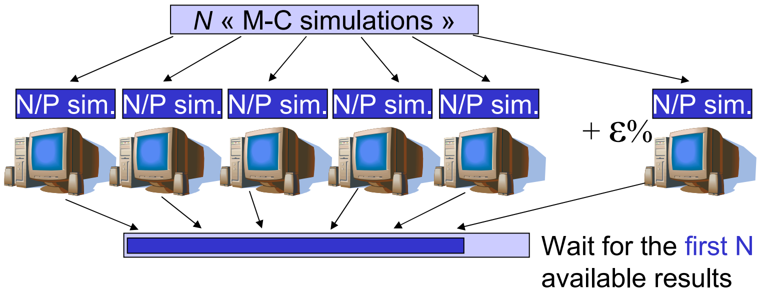
➔ Independent simulation sub-sets.

+ Easy-to-implement (straightforward)
+ Fast: no synchronization overhead…

- No fault-tolerance mechanism
- No load balancing: assumes identical nodes

4 – Grid algorithmic strategy

4.3 - Static and redundant split of the simulation work:



N « M-C simulations »

N/P sim. N/P sim. N/P sim. N/P sim. N/P sim. N/P sim.

+ ε%

Wait for the first N available results

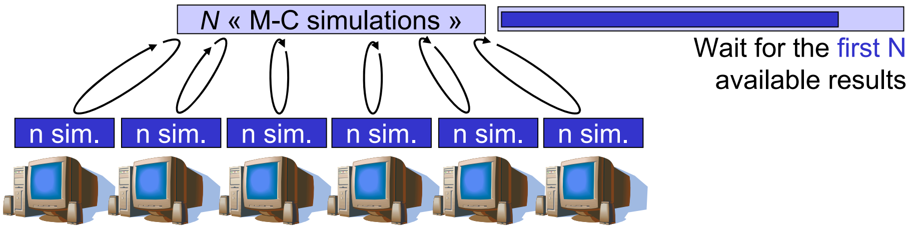
Independent simulation sub-sets.

+ Low synchronization overhead
+ Statistical fault-tolerance mechanism

- Use more resources than necessary
- No load balancing: assumes identical nodes

4 – Grid algorithmic strategy

4.4 - Dynamic split and load balancing of simulation sub-tasks:



N « M-C simulations »

Wait for the first N available results

n sim. n sim. n sim. n sim. n sim. n sim.

Independent simulation sub-sets.

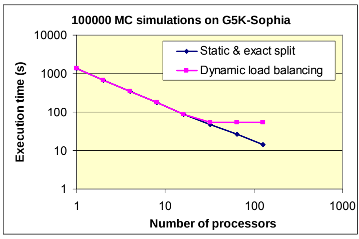
+ Dynamic load balancing
    + support heterogeneous resources
    + fault-tolerant system

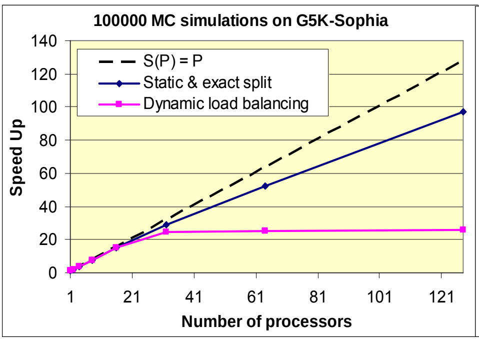
- Greater synchronization overhead
- Possible contention (high number of nodes)

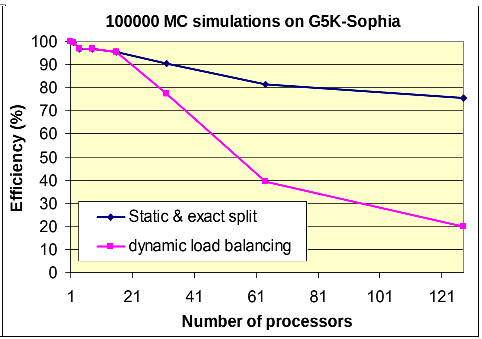
5 – First Experiments

128 processors of an homogeneous cluster:

- Static & exact load balancing is the most efficient,
- Would be different on an heterogeneous Grid.



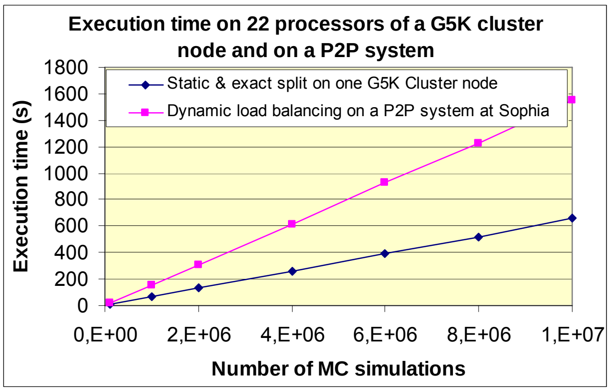


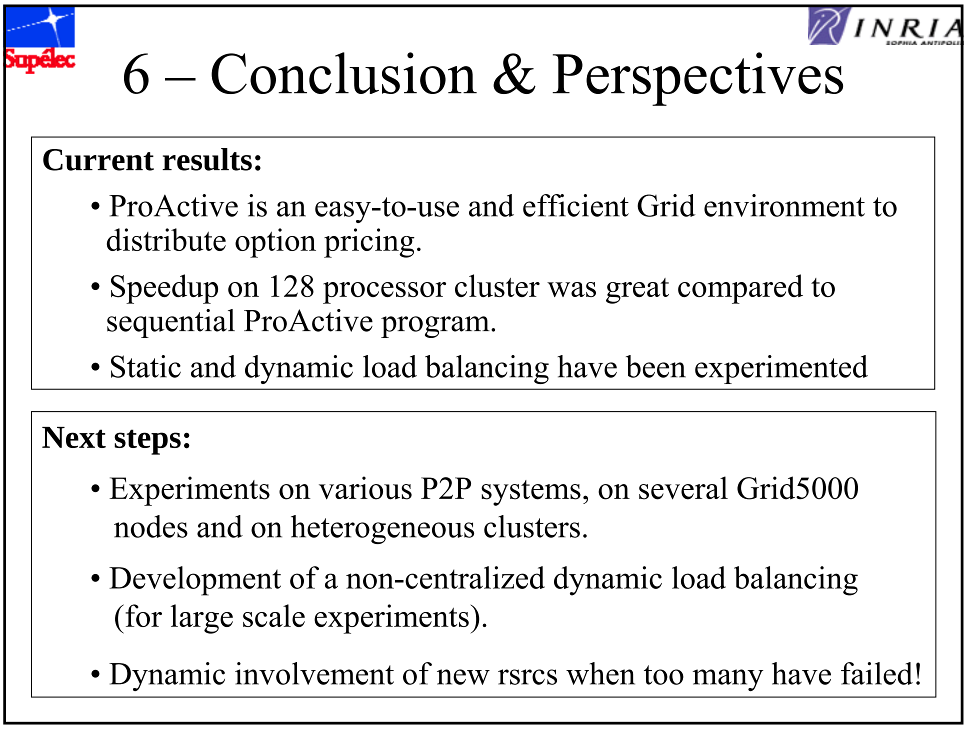


5 – First Experiments

Homogeneous Grid cluster node *vs* P2P system:

- Static & exact split on homogeneous Grid cluster node,
- Dynamic load balancing on heterogeneous P2P system
- Homogeneous cluster & static method is most efficient
- Should be different on an heterogeneous Grid, or on an old cluster !!





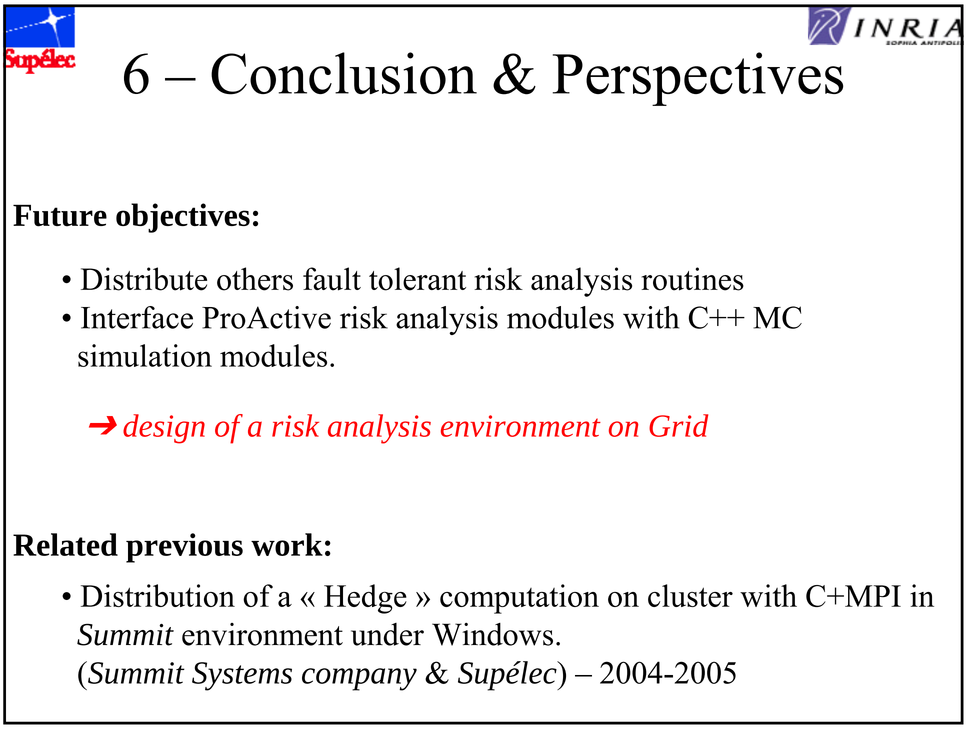
6 – Conclusion & Perspectives

Current results:

- ProActive is an easy-to-use and efficient Grid environment to distribute option pricing.
- Speedup on 128 processor cluster was great compared to sequential ProActive program.
- Static and dynamic load balancing have been experimented

Next steps:

- Experiments on various P2P systems, on several Grid5000 nodes and on heterogeneous clusters.
- Development of a non-centralized dynamic load balancing (for large scale experiments).
- Dynamic involvement of new rsrcs when too many have failed!



6 – Conclusion & Perspectives

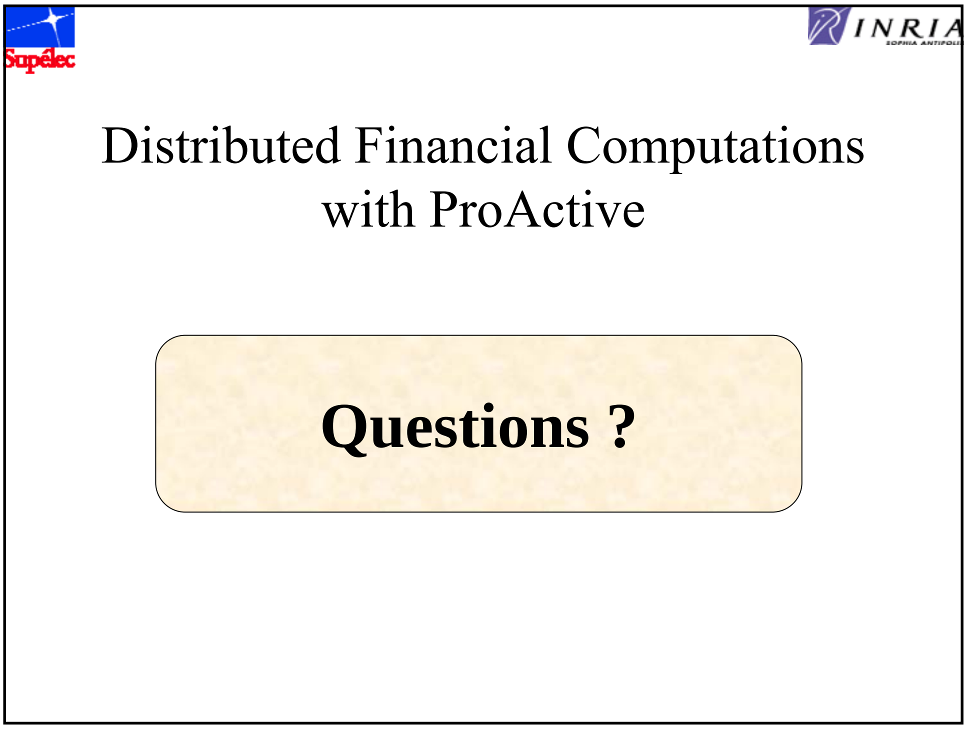
Future objectives:

- Distribute others fault tolerant risk analysis routines
- Interface ProActive risk analysis modules with C++ MC simulation modules.

➔ *design of a risk analysis environment on Grid*

Related previous work:

- Distribution of a « Hedge » computation on cluster with C+MPI in *Summit* environment under Windows.
  (*Summit Systems company & Supélec*) – 2004-2005



Distributed Financial Computations
with ProActive

Questions ?