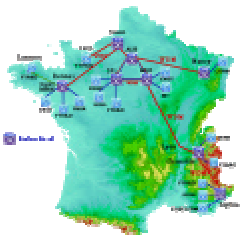


# Computational Servers on a High Speed WAN

E. Fleury, E. Jeannot
**LORIA, INRIA Lorraine**
**Nancy, France**

E. Caron, F. Desprez, M. Quinson, F. Suter
**INRIA Rhône-Alpes**
**LIP ENS Lyon, France**

S. Contassot, F. Lombard, J.-M. Nicod, L. Philippe
**LIFC**
**Besançon, France**

## INTRODUCTION

One long term idea for Metacomputing: renting computation power and memory capacity over the Internet

☺ Very high potential
- Need of Problem Solving Environments (PSEs)
- Applications need more and more memory capacity and computational power
- Some proprietary libraries or environments need to stay in place
- Some confidential data must not circulate over the net

➔ Use of computational servers accessible through a simple interface

☹ Still difficult to use for non-specialists
- ☹ Almost no transparency
- ☹ Security and accounting issues usually not addressed

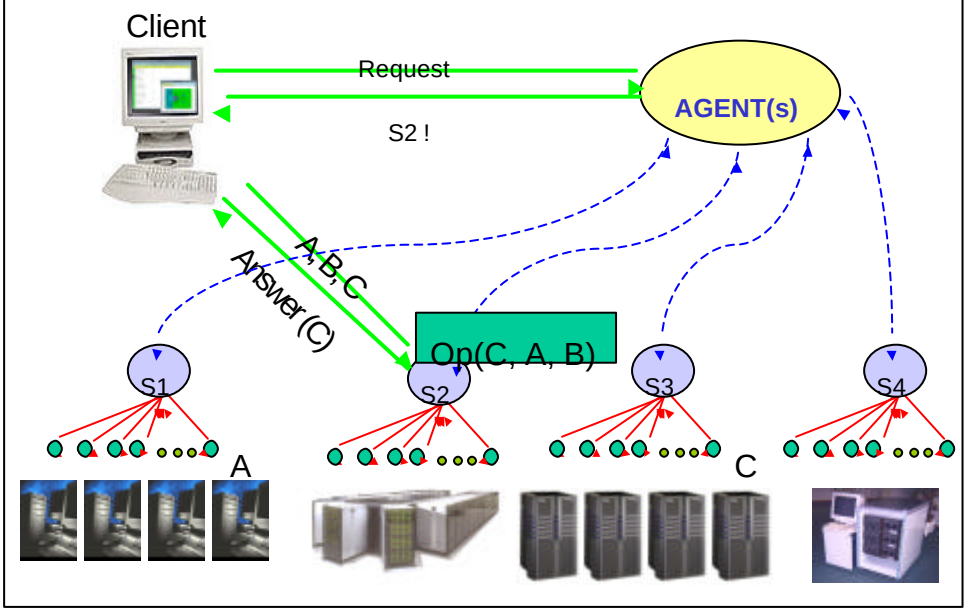
☹ Often application dependent PSEs

☹ Lack of standards (CORBA, JAVA/JINI, sockets, …) to build the computational servers

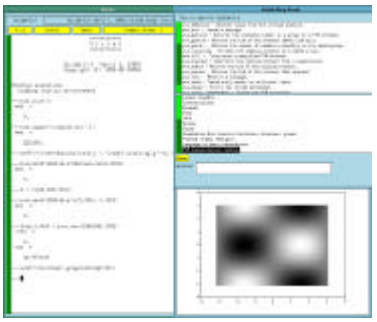
## Related projects

- Network enabled solvers
  - Netsolve (University of Tennessee, USA)
  - NINF (Electrotechnical Lab, Umezono, JP)
  - Remote Computation System (ETH Zürich, CH)
  - Meta-NEOS (Argonne National Lab.)
- Problem Solving Environments
  - Information Power Grid (NASA)
  - Parallel Simulation User Environment (Univ. of Wales, UK)
- Web access to computational servers
  - Virtual Distributed Computing Environment (Syracuse Univ.)
  - WebFLOW (Syracuse Univ.)
- Many other projects (distributed visualization, collaborative working, distance education, application dependent simulation environment, …)
- Related software efforts
  - GrADS, Globus, Legion, APPLEs, Nimrod-G, DISCWorld, CoG toolkit

## Big picture

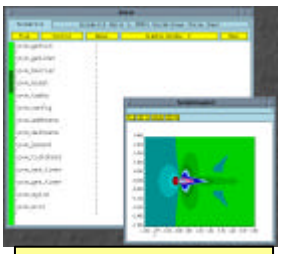


## Scilab to Scilab//



- **Scilab**:
  - free Matlab-like tool
  - simple syntax
  - many mathematical functions
  - graphical interface
  - high level language
- **To develop tools for big applications in Scilab.**
  - parallel version of Scilab
  - dynamic load-balancing
  - parallel sparse solver libraries (PaStiX) with a parallel graph partitioning tool (Scotch), data visualization tool
  - computational servers with parallel libraries (in-core and OOC)
  - in a metacomputing environment

## Several approaches



Message passing between Scilab tasks (PVM, MPI),

Parallel libraries interfaces, (ScaLAPACK),

Out-of-core parallelism (ScaLAPACK extension)

Netsolve Interface (Univ. of Tennessee),

CORBA approach (management of servers)

Computational servers (SLIM, FAST)

Mixed parallelism management (task/data-parallelism)

Mixed compilation/run-time support

Multithreaded Scilab scheduling & migration

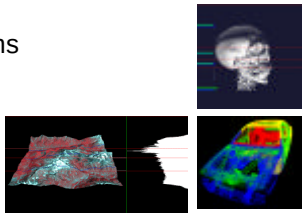
Sparse matrix solvers in heterogeneous environments

## Our view of computational servers

- **Ideas**
  - Scilab as a (first) target application
  - To avoid multiple interpretation steps
  - To ease the insertion of new libraries
  - To benefit of existing developments around metacomputing
  - To develop a toolkit for computational servers
- **First prototype developed from existing software**
  - Netsolve (University of Tennessee)
  - NWS (UCSD, UTK & UCSB) for the dynamic evaluation of performances
  - Our developments on libraries (redistribution routines, sparse solver, out-of-core routines)
  - LDAP for the software database and CORBA for the management of servers

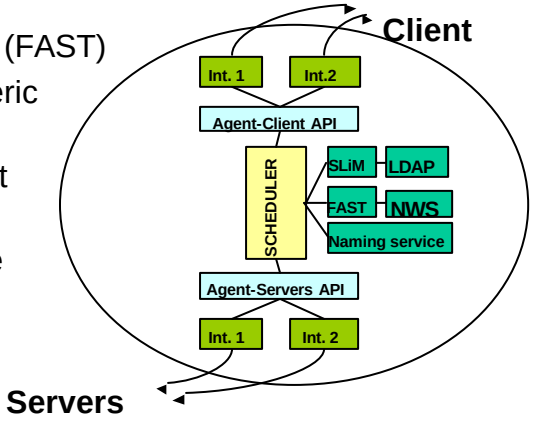
## Our first goals

- To add some features to Netsolve for our own application
  - Data persistence on servers
  - Redistribution and parallelism between servers
  - Better accuracy of the evaluation of pairs [routine, machine] for small grain computations
  - Improving the scheduler (testing on-line scheduling heuristics)
  - Portable database for the available libraries (LDAP)
- To have a test and experimentation platform for our developments
  - Mixed parallelism approach (data-parallelism/task-parallelism)
  - Scheduling heuristics for data-parallel tasks
  - Parallel algorithms on heterogeneous platforms
  - Performance evaluation
  - CORBA management of servers
- NSF/INRIA with J. Dongarra's team



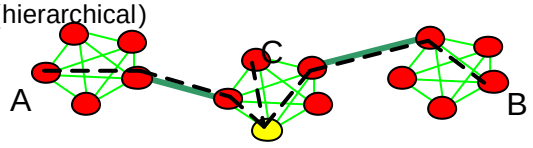
## Improving the agent(s)

- Scheduler
  - application dependent and general heuristics
- Software resource localization (SLiM)
  - based on LDAP
- Performance evaluation (FAST)
- Naming service for generic data
- Different API for different interfaces
- Hierarchical architecture for scalability

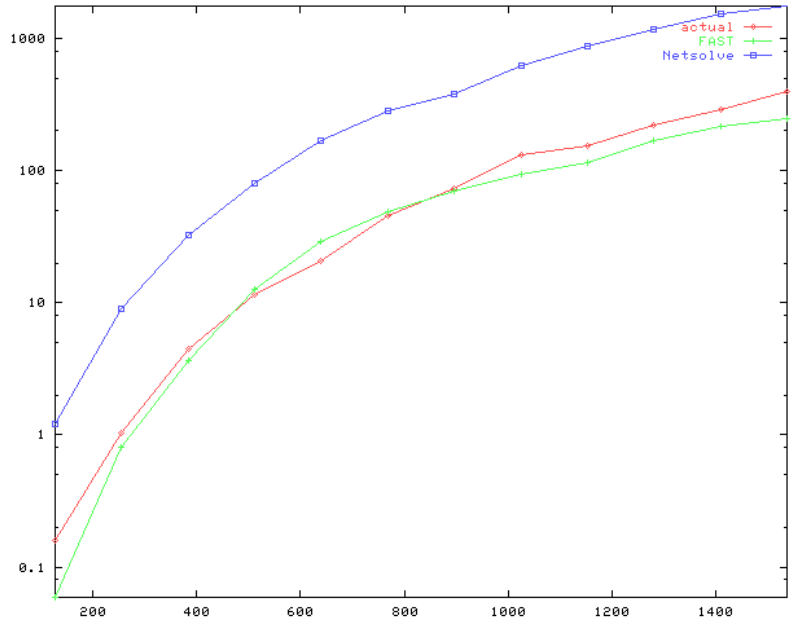



## FAST: Fast Agent's System Timer

- Performance evaluation of the platform to be able to find a efficient server (redistribution and computation costs) without testing every configuration ➔ performance database for the scheduler
- Based on NWS (*Network Weather Service*)
- **Computation performances**
  - machine load, memory capacity, and batch queues performances (dynamic)
  - extensive testing of several libraries (static)
- **Communication performances**
  - To be able to guess the cost of the redistribution of data between two servers as a function of the network architecture and dynamic informations
  - Latency and bandwidth (hierarchical)
- **Hierarchical set of agents**
  - Scalability issues



## Performance results




## Adding data persistence to Netsolve

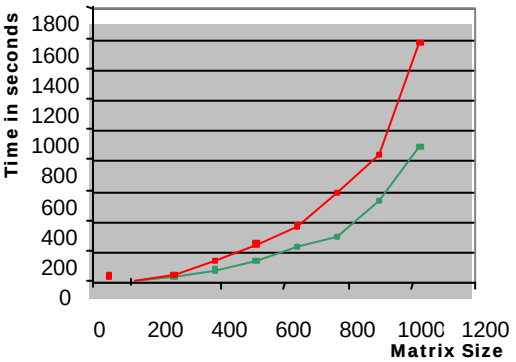
- Netsolve sends data back and forth between client and servers
- **Our goals**:
  - To leave data on servers after computations
  - To enable communication between servers
  - Standard clients should continue to execute as usual

  - **Execution example**
    - Complex matrix multiplication

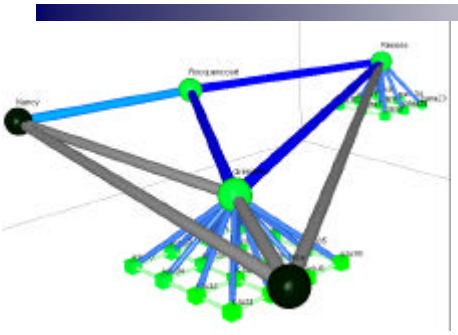
      C=A*B

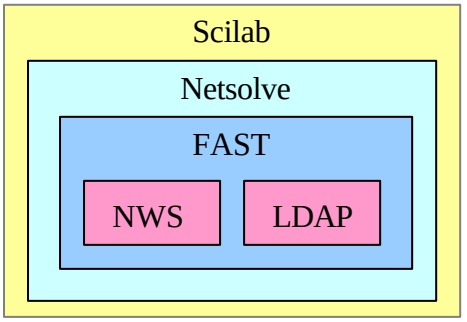
      Cr=Ar*Br-Ai*Bi

      Ci=Ar*Bi+Ai*Br
    - Clients and servers in 2 differents labs in France



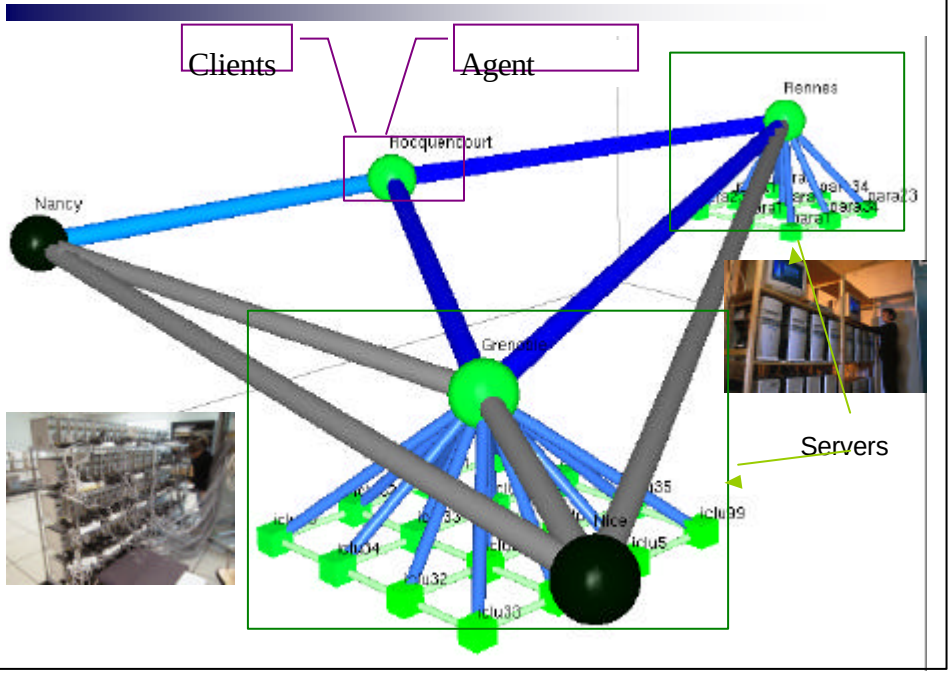
## Netsolve over VTHD



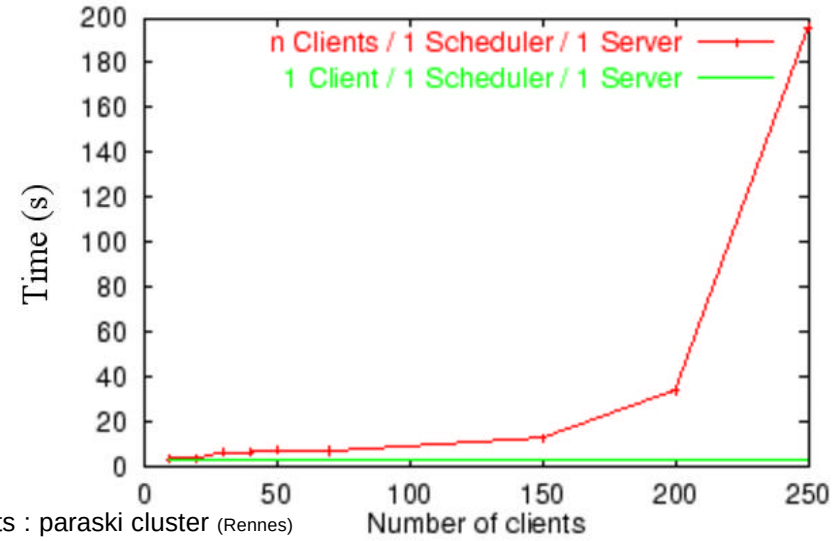



- Application: Strassen.
- Client: INRIA Rocquencourt
- Servers: Rennes and Grenoble
- Visualization Tool: Cichlid

## Netsolve over VTHD



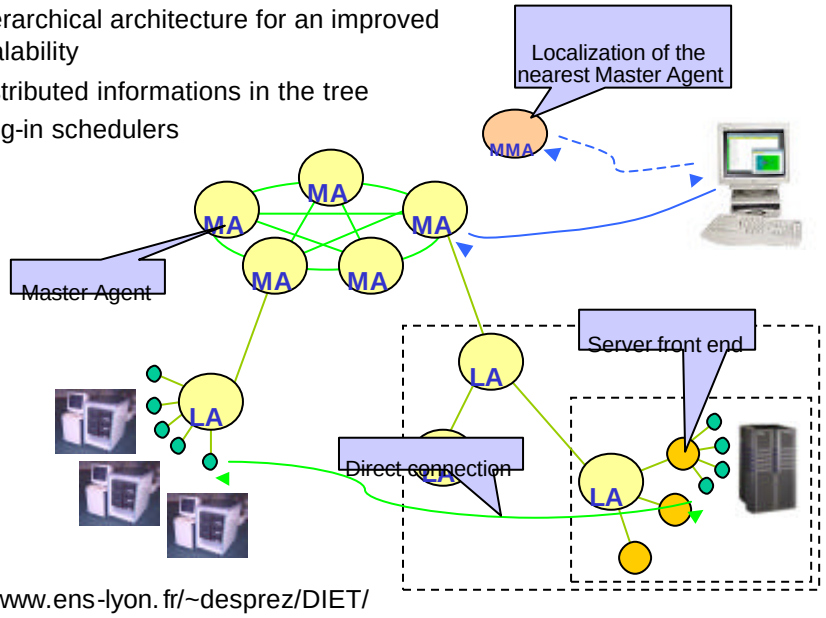
## Netsolve Agent behavior




- Clients : paraski cluster (Rennes)
- Scheduler : Agent Netsolve (Rocquencourt)
- Server : paraski26 (Rennes)

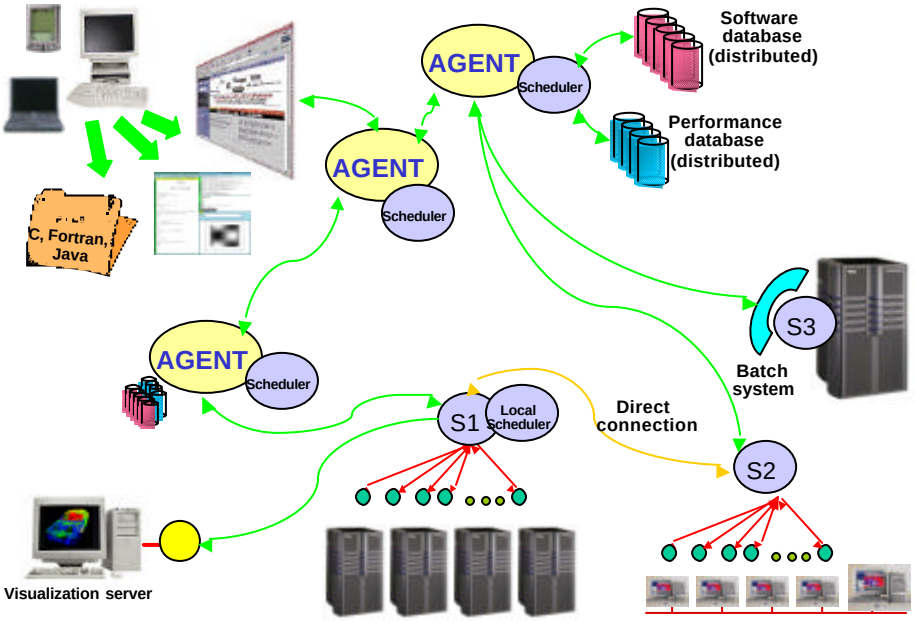
## Hierarchical architecture for VTHD (DIET)

- Hierarchical architecture for an improved scalability
- Distributed informations in the tree
- Plug-in schedulers




http://www.ens-lyon.fr/~desprez/DIET/

## Improved big picture




## Conclusions and Future Work about DIET

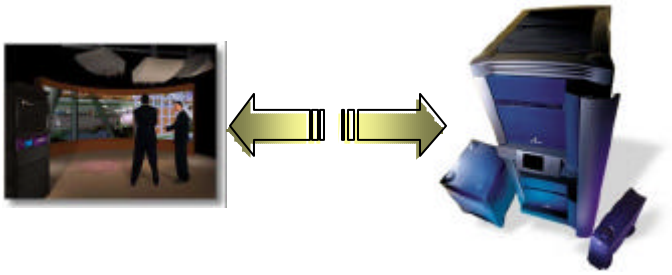
- Building a portable set of tools for computational servers
- Focus on issues like resource localization, scheduling, performance analysis, mobile clients and applications
- Extension to sparse matrix solvers
  - Needs problem analysis (LSA)
  - Interoperability of libraries (data redistribution between servers)
- Study the integration of other applications (not only numerical ones)
- Discussions with Paris, NINF, NWS, AppLeS, and Netsolve teams (at least) !

DIET
Distributed Interactive Engineering Toolbox

http://www.ens-lyon.fr/~desprez/DIET/

## Other metacomputing applications on VTHD

- **SIMBIO**
  - Molecular simulation and code coupling
- **CAST**
  - Code coupling environment
- **Madeleine**
  - Multi-protocol communication library
- **Virtual reality** (connection of SGI O2K and CAVEs)



## Conclusions and Future Work

- VTHD is a good testbed for our developments
- Not so easy !!!
- **Connection of several clusters** (10 to 225 processors)
  - Heterogeneity (processors, networks)
  - Administration issues
- **Network issues**
  - Application heterogeneity
  - QoS
  - Performance prediction
- Several other projects will benefit of the results
  - RNTL, ACI Grid, RNRT, IST, …
  - Connection to other networks ?

# Hiring ! *On embauche !*

- Stages DESS, DEA
- Postdocs ou ingénieurs de recherche sur des projets RNTL ACI Grid ou RNRT


"Stop whining — most people do their best work under pressure!"