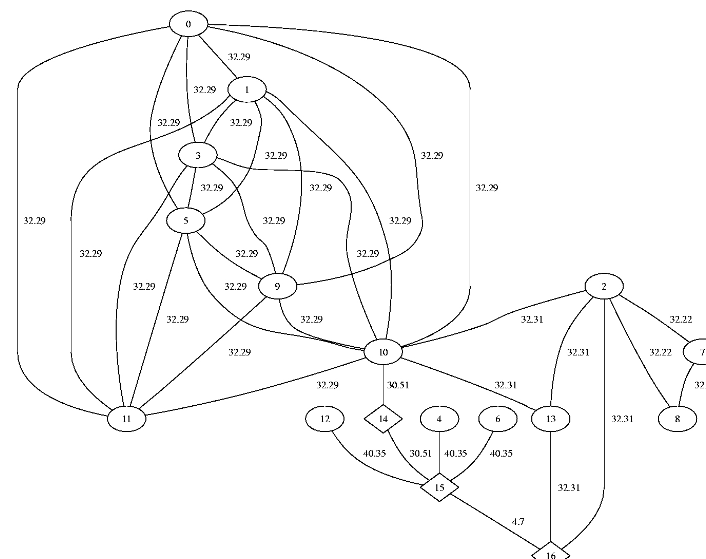
GRAAL

Activities around Client-Server Computing over the Grid

Jean-Yves L'Excellent
LIP ENS Lyon
INRIA Rhône-Alpes



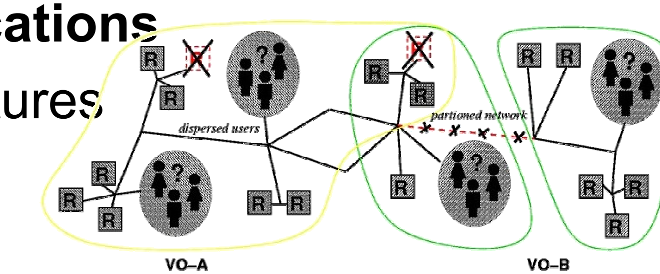
CONTEXT: CNRS / ENS Lyon / INRIA Project GRAAL

- GRAAL (previously ReMaP) = GRids And ALgorithms
- Project Leader: Frédéric Desprez (Frederic.Desprez@inria.fr)

- GOAL = concentrate on algorithmic problems
  - Algorithm Design and Scheduling Strategies (Y. Robert, F. Vivien)
  - Client-Server approach for distributed computing (E. Caron, F. Desprez)
  - Scheduling for solvers of sparse systems of equations (J.-Y. L'Excellent)

- Keywords:

  Design of algorithms + libraries + applications
  on heterogeneous and distributed architectures



Algorithm Design and Scheduling Strategies

Y. Robert, F. Vivien

Algorithm Design and Scheduling Strategies: Goals

- Study the impact of new architectural parameters:
  - heterogeneity,
  - volatility,
  - hierarchy.

- Need of a theoretical approach in spite of the difficulty of scheduling problems (minimisation of makespan)

- Inject static knowledge in an essentially dynamic environment

- Evaluate strategies: compare heuristics in the exact same experimental conditions with simulated realistic load:
  - Use NWS to get realistic load informations
  - SimGrid (developed in collaboration with UCSD) to simulate scheduling strategies

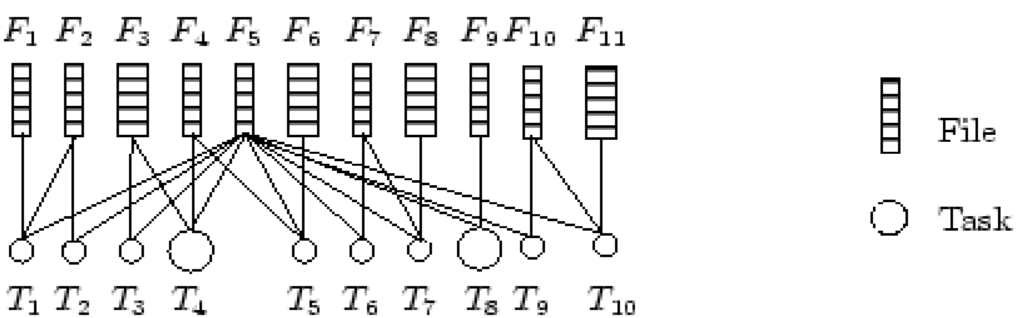
Algorithm Design and Scheduling Strategies: Steady-state Scheduling

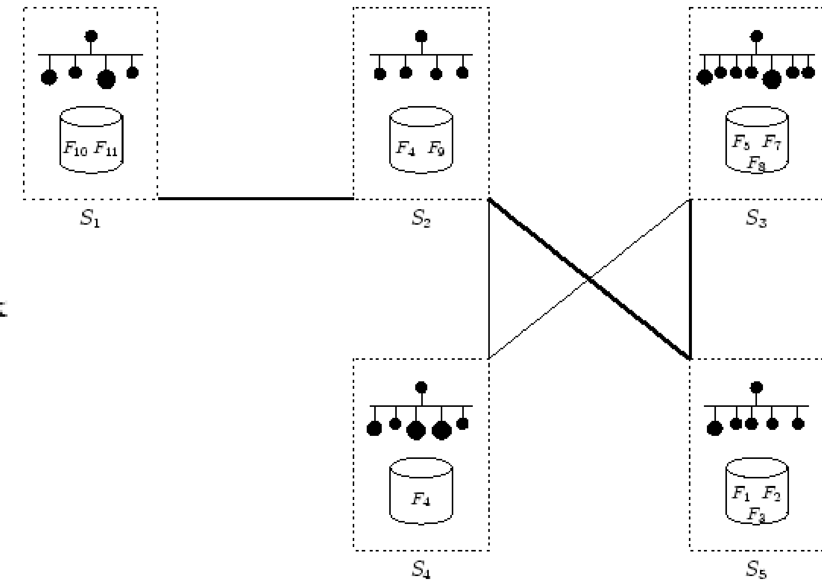
- Most scheduling problems are very difficult on heterogeneous platforms
- If you assume that the problem is very large and regular, you can solve some of these problems

→ **Asymptotic optimality for various problems:**

- Scheduling large number of identical task graphs on an heterogeneous platform.
- Divisible load scheduling.
- Collective communications (scatter/gather, broadcast, reduce,...)

Algorithm Design and Scheduling Strategies: Scheduling Tasks Sharing Files





- Set of tasks
  - Each task depends on several files
  - A file may be shared by several tasks
- Fully heterogeneous platform
- Files originally distributed on the different repositories

- Problem:
  - where to map the tasks? where to duplicate files?
- Solution:
  - (complexity results and) quick and efficient heuristics
- Possible application:
  - comparison of medical images hosted by different hospitals

Client-Server Approach for Distributed Computing

E. Caron, F. Desprez, J.-M. Nicod, L. Philippe

Goals

One long term idea for Grid computing: rent computational 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

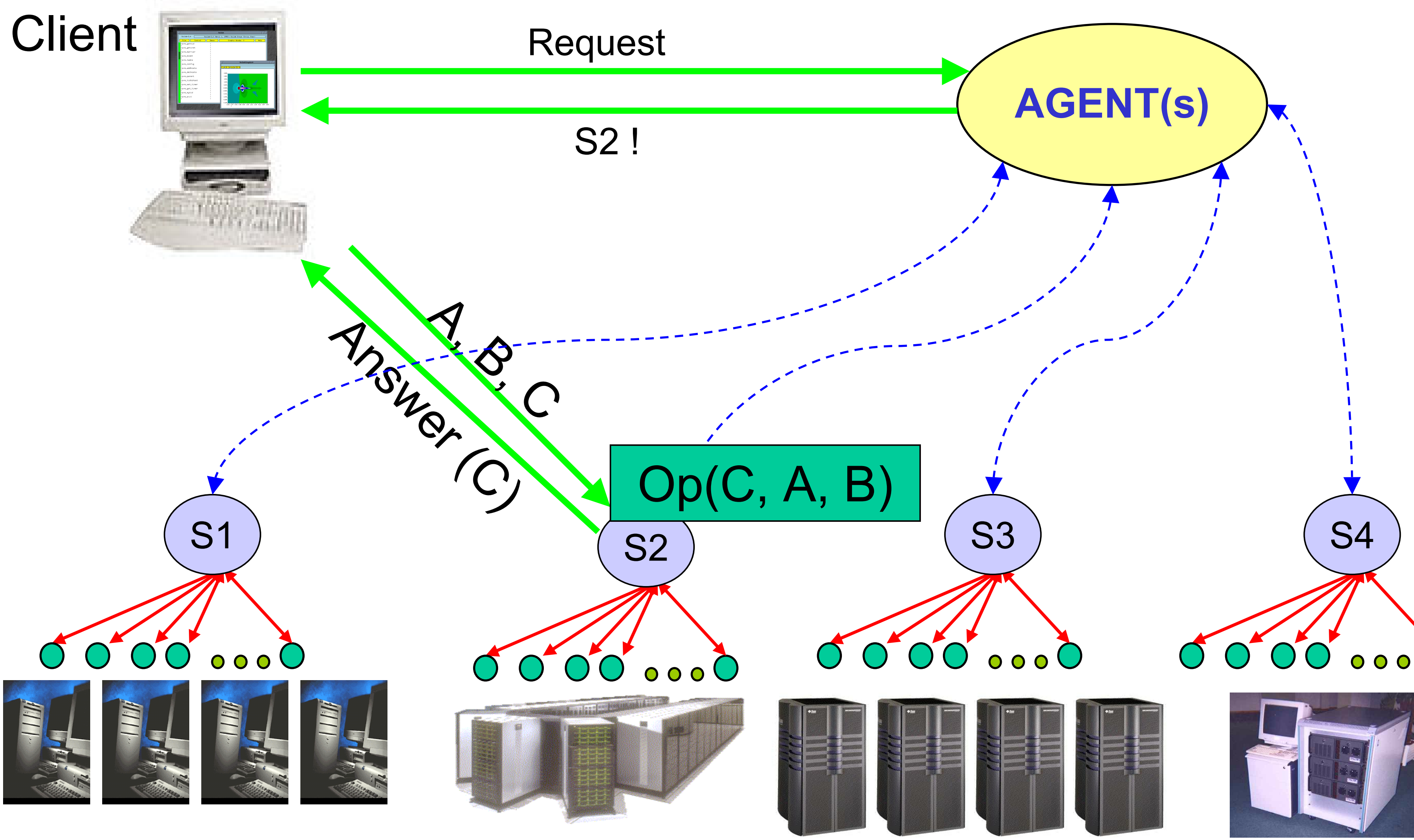
➨ **Use of computational servers accessible through a simple interface**

- But …
  - Still difficult to use for non-specialists
    - Almost no transparency
    - Security and accounting issues difficult to address
  - Often application-dependent PSEs
  - Lack of standards
    - (CORBA, JAVA/JINI, sockets, …) to build the computational servers

Goals

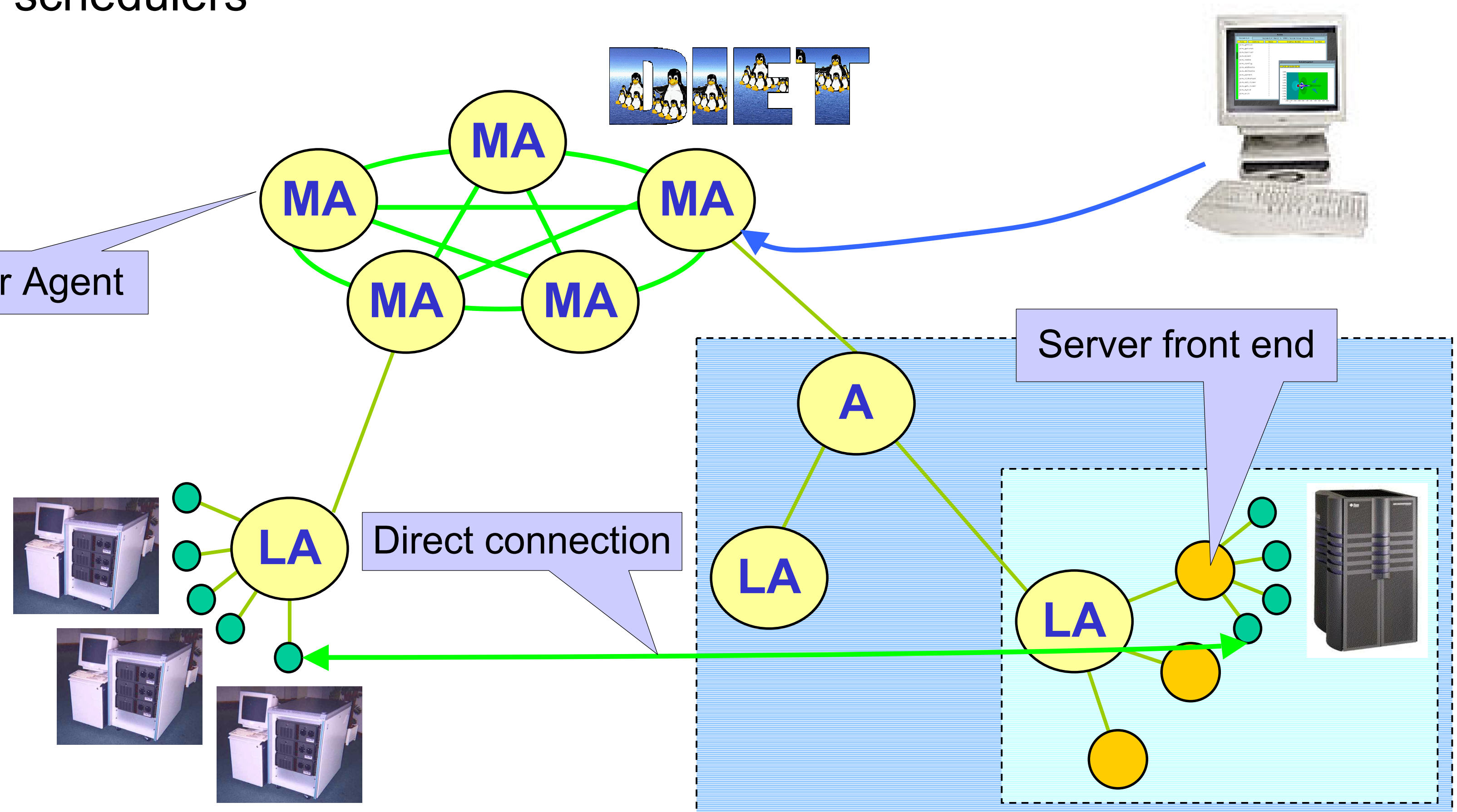
- Design of a toolbox for the deployment of environments using the *Application Service Provider* (ASP) paradigm (using CORBA)
- A simple idea
  - RPC programming model for the Grid
  - Use of distributed collections of heterogeneous platforms
  - Task parallelism programming model (synchronous/asynchronous) + data parallelism on servers ➡ mixed parallelism
- Functionalities required
  - Load balancing
    - resource discovery
    - performance evaluation
    - Scheduling
  - Fault tolerance
  - Data redistribution
  - Security
  - Interoperability, …

GridRPC



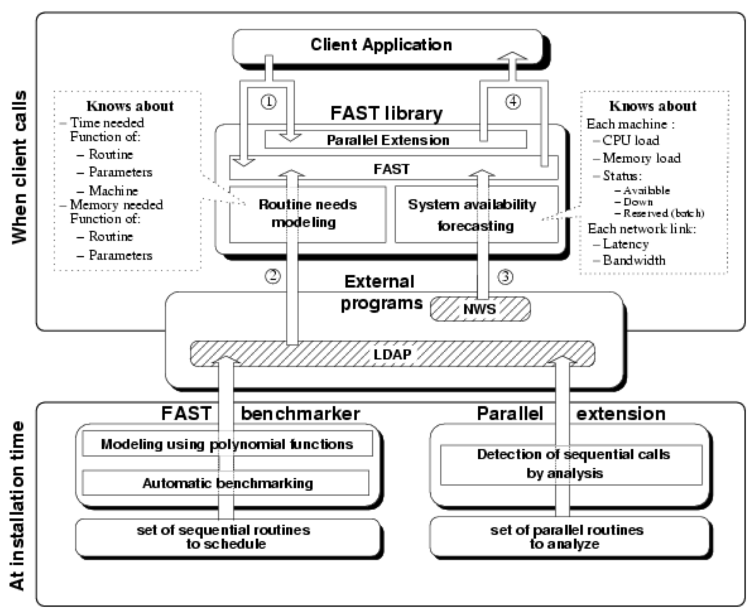
DIET - Distributed Interactive Engineering Toolbox -

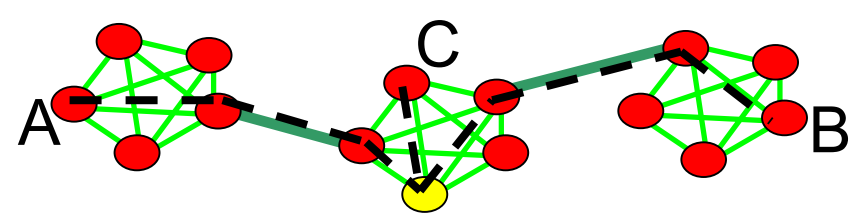
- **Hierarchical architecture** for an improved scalability
- Distributed information in the tree
- Plug-in schedulers



FAST - Fast Agent's System Timer -

- NWS-based (Network Weather Service, UCSB)
- Computational performance
  - Load, memory capacity, and performance of batch queues (dynamic)
  - Benchmarks and modeling of available libraries (static)
- Communication performance
  - To be able to guess the data redistribution cost between two servers (or clients to server) as a function of the network architecture and dynamic information
  - Bandwidth and latency (hierarchical)





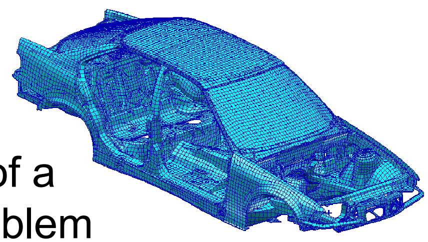
Things we are working on now

- Scheduling
  - Plugin schedulers,
  - Reservation of resources,
  - Hierarchical and distributed scheduling,
  - Mixed parallelism
- Performance evaluation
  - Automatic deployment of NWS,
  - Topology discovery (application point-of-view)
  - Modelization of parallel applications
- Data management
  - Data persistency
  - Replication of data
- Relations with Globus (OGSA)
- Applications !

Scheduling for solvers of sparse systems

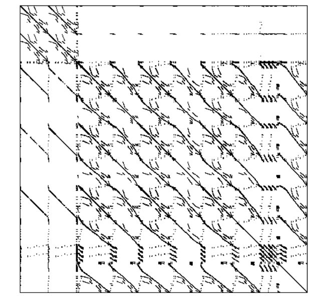
J.-Y. L'Excellent

Solvers for large sparse systems of equations

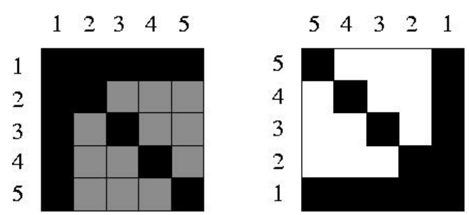


Simulation of a physical problem (eg, finite elements)

Solution of a sparse system A x = b



- Direct methods (e.g. : multifrontal method)
  - A = LU or LDL^T (Gauss)
  - Very robust if numerical pivoting (⇒ dynamic data structures)
- Reordering heuristics



Fill in

Initial matrix

  - AMD, AMF, SCOTCH (ScAlApplix), PORD (Univ of Paderborn), METIS (Univ of Minnesota)
  - Huge impact on the topology of the task dependency graphs

→ Study impact on memory / performance / parallelism

Solvers: current work

- Scheduling and Load Balancing issues
  - Distributed scheduling (dynamic approach + static information)
  - Adapt to various platforms (clusters of SMP, multi-user platforms, grid ?)
  - **Goal:** minimize **execution time**
    and/or **memory scalability**

- Numerical aspects
  - Combine direct and iterative methods
  - New functionalities for specific applications (optimization, eigenvalues, …)

- MUMPS (a MUltifrontal Massively Parallel Solver)
  - Competitive package (INRIA, ENSEEIHT-IRIT, CERFACS, PARALLAB)
  - Integrates recent research and is very general *(symmetric/unsymmetric sparse problems, element-entry, distributed matrix entry, partial factorization, Schur complement, real or complex arithmetic, scalings, backward error analysis, …)*
  - Available free of charge

Solvers: current work

- Sparse direct solvers in a client-server environment (DIET)
  - Provide remote access to the algorithms we develop (e.g. MUMPS)
  - Easy to use from a light client
  - Data persistency on the servers is crucial

- Application: an expertise site for sparse linear algebra: GRID TLSE (coordinated by ENSEEIHT-IRIT, Toulouse)
  - On a user's specific problem, compare execution time / accuracy / memory usage / … of various solvers:
    - public domain … as well as commercial,
    - sequential … as well as parallel
  - Find best parameter values / reordering heuristics on a given problem
  - Also bibliography, matrix collections, …

➢ All elementary requests executed on the/a GRID through DIET

➢ Must be highly evolutive (new solvers with new parameters, new scenarii)

Start a new expertise

Help about scenarii

Select solvers

- [x] MUMPS
- [ ] SUPERLU
- [x] UMFPACK

Choose metrics

- [ ] Estimated Flops
- [ ] Estimated Memory
- [ ] Effective Flops
- [x] Effective Memory
- [x] Total Time
- [ ] Residual

Choose an objective

- (•) Ordering Sensitivity
- ( ) Minimum Time
- ( ) Threshold Sensitivity
- ( ) Solve

File name : rdist1.rua

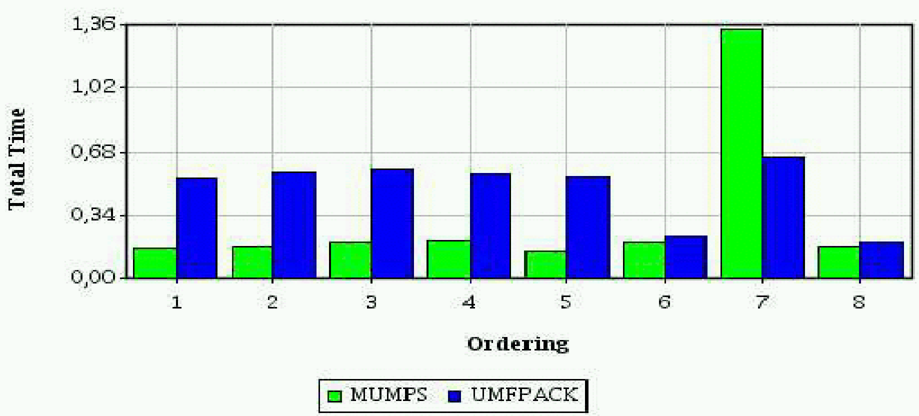
Continue / Search matrix | Reset

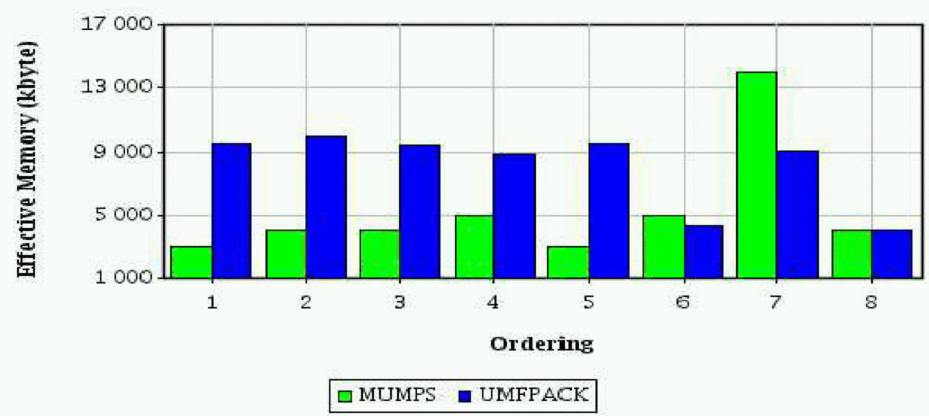
Experimental Results

New expertise

MATRIX NAME : rdist1.rua

	Ordering	Total Time	Effective Memory (kbyte)
MUMPS	(1) AMD	1,6E-1	3E3
	(2) AMF	1,7E-1	4E3
	(3) PORD	1,9E-1	4E3
	(4) METIS	2E-1	5E3
	(5) QAMD	1,4E-1	3E3
	(6) MMD x	1,9E-1	5E3
	(7) MMD +	1,34E0	1,4E4
	(8) COLAMD	1,7E-1	4E3
UMFPACK	(1) AMD	5,4E-1	9,528E3
	(2) AMF	5,7E-1	9,888E3
	(3) PORD	5,9E-1	9,456E3
	(4) METIS	5,6E-1	8,823E3
	(5) QAMD	5,5E-1	9,528E3
	(6) MMD x	2,2E-1	4,246E3
	(7) MMD +	6,5E-1	9,055E3
	(8) COLAMD	1,9E-1	4,009E3





Summary: Possible collaborations …

… on research themes of mutual interest:

- Algorithm design and scheduling strategies (contact: Y. Robert, F. Vivien)
- Parallel sparse direct solvers (contact: J.-Y. L'Excellent)
- Client-Server approaches over the grid (contact: E. Caron, F. Desprez)

… with teams interested in using the tools we work on:

- DIET (toolbox for client-server approach on the grid)
- SimGrid (simulation of distributed platforms)
- MUMPS (general sparse direct solver)
- GRID TLSE (expertise site for sparse linear algebra)