Visualization for Performance Debugging of Large-Scale Parallel Applications

Lucas M. Schnorr (CNRS), Guillaume Huard (UJF),
Benhur Stein (UFSM), Jean-Marc Vincent (UJF)

– SBAC-PAD 2009 Tutorial –
São Paulo, Brazil
October 31st, 2009

à la mémoire de Jacques Chassin de Kergommeaux

1. Introduction
   - Motivations
   - Examples
2. Trace Fundamentals
   - Fundamentals
   - Pajé
3. Performance Analysis
   - Three-Dimensional Model
   - Temporal & Spatial Aggregation Model
4. Synthesis
   - Research directions

Motivations

Scientific context

- Complex parallel/distributed programs
- Potentially large size parallel applications.
- Executing on large size parallel systems:
  - Distributed systems
  - Clusters and Grids
  - Desktop grids, P2P systems...

Keypoints

- Distributed heterogeneous resources
- Dynamicity of the architecture
- Scalability (huge amount of data)

General Objective

Help users find performance errors:

- Visualization of parallelism, identify synchronization overheads,
- Usage of resources, identify bottlenecks,
- Behavior analysis method.

General Objective

Help users find performance errors:

- Visualization of parallelism, identify synchronization overheads,
- Usage of resources, identify bottlenecks,
- Behavior analysis method.

Based on:

- Execution model : user events,
- Infrastructure model : Measurement environment
- Visualisation model : graphical objects.

Visualization of parallel program execution

Who ?

Program designer, Program certifier, ···

··· Parallel programs vendors

Visualization of parallel program execution

Who ?

Program designer, Program certifier, · · ·

· · · Parallel programs vendors

Why ?

- Program debugging,
- Quantitative debugging (performance evaluation),
- Dimensionning and performance tuning

Visualization of parallel program execution

Who ?

Program designer, Program certifier, · · ·

· · · Parallel programs vendors

Why ?

- Program debugging,
- Quantitative debugging (performance evaluation),
- Dimensionning and performance tuning

How ?

- Graphical representation of the parallel execution
- Interactive representation (exploration)
  - zoom in and out on time, infrastructure, on objects
  - compute statistics

Methodology

Execution model

- Abstraction of the parallel execution : state / event model
- Observability of states / Practical interest of states
- Quality of observation (interaction tracer/application)

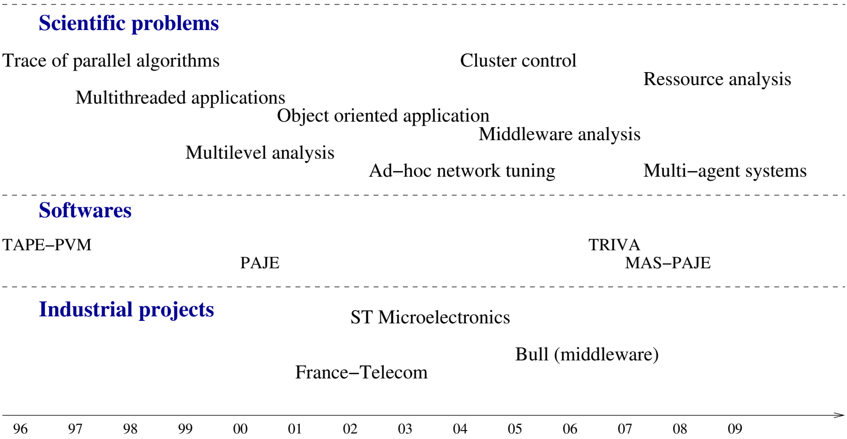
Environment model

- Structured set of resources (architecture)
- Model of time : Datation model

⇒ **Manipulation language of resources, states and events**

Collaboration (a not so short story)

UFSM, UFRGS, U. of Grenoble, INRIA



Scientific problems

Trace of parallel algorithms

Multithreaded applications

Multilevel analysis

Object oriented application

Ad−hoc network tuning

Cluster control

Middleware analysis

Ressource analysis

Multi−agent systems

Softwares

TAPE−PVM

PAJE

TRIVA

MAS−PAJE

Industrial projects

France−Telecom

ST Microelectronics

Bull (middleware)

Introduction - Existing Tools/Techniques

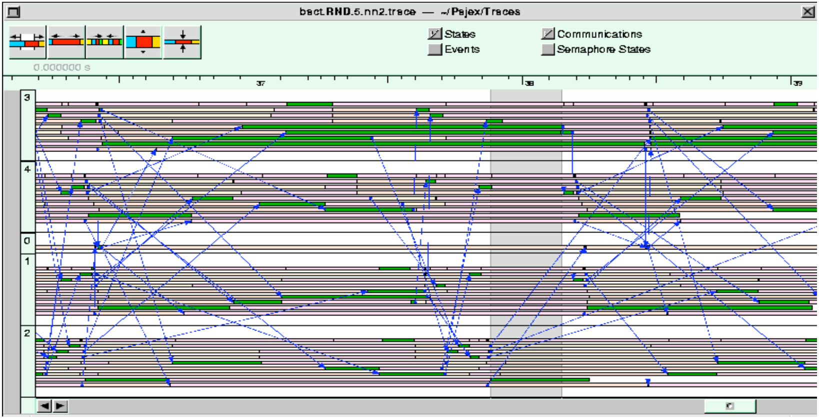
- Statistical Techniques
  - ParaGraph (1990) – bar charts, utilization Count
  - Pablo (1993) – bar charts + 3D scatter plot
  - Paradyn (1995) – histograms
- Behavioral Techniques
  - ParaGraph (1990) – Gantt-chart
  - Vampir (1996) – time-line system view
  - Jumpshot (1999), Pajé (2000) – space-time
  - Virtue (1999) – virtual reality to performance analysis
  - Kojak, ParaProf (2003) – Call Graph
- Structural Techniques
  - ParaGraph (1990) – network display / hypercube
  - Cray Apprentice (2007) – tree view of imbalances

Main difficulty

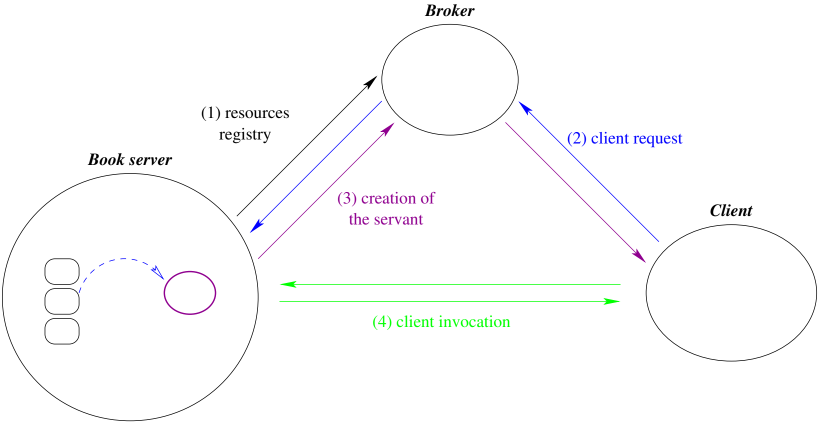
Large scale systems

- Large number of objects
- Complexity of views
- Level of abstraction
- Dynamicity of the observed infrastructure

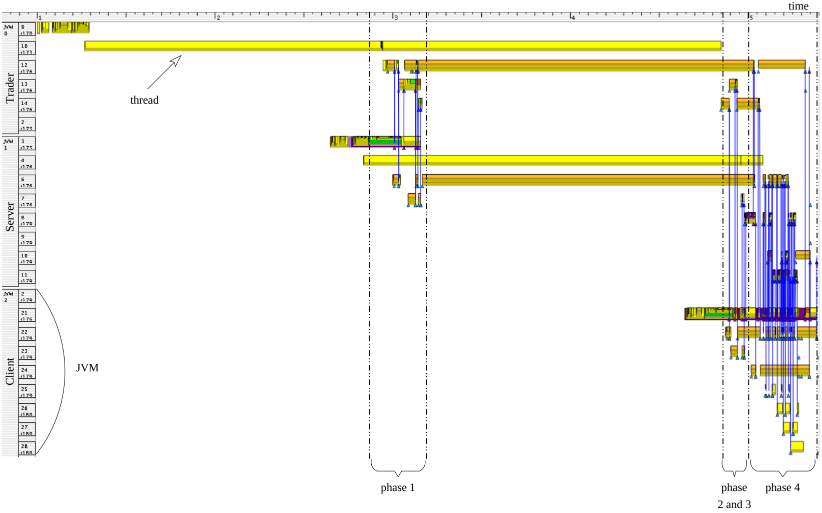
Multithreaded Applications (1999)



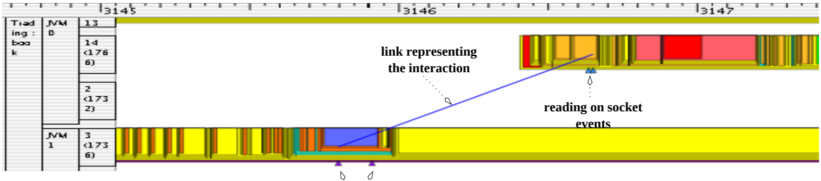
Distributed Middleware



Distributed Middleware (2)



Distributed Middleware (3)

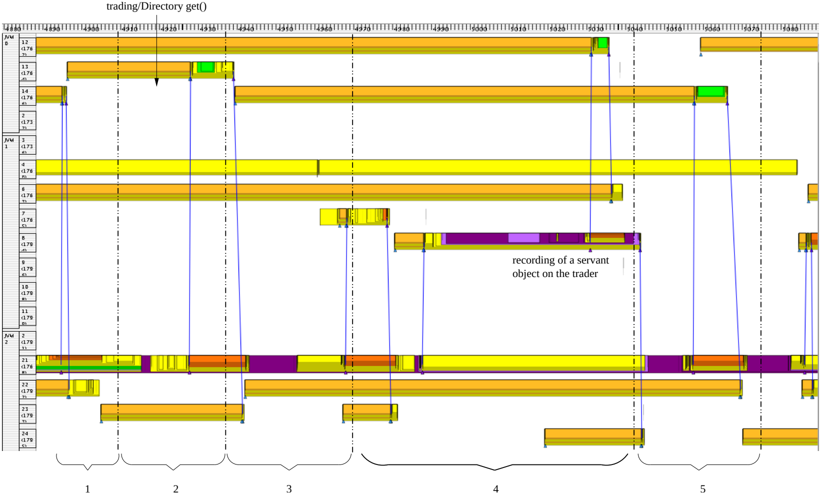


link representing the interaction

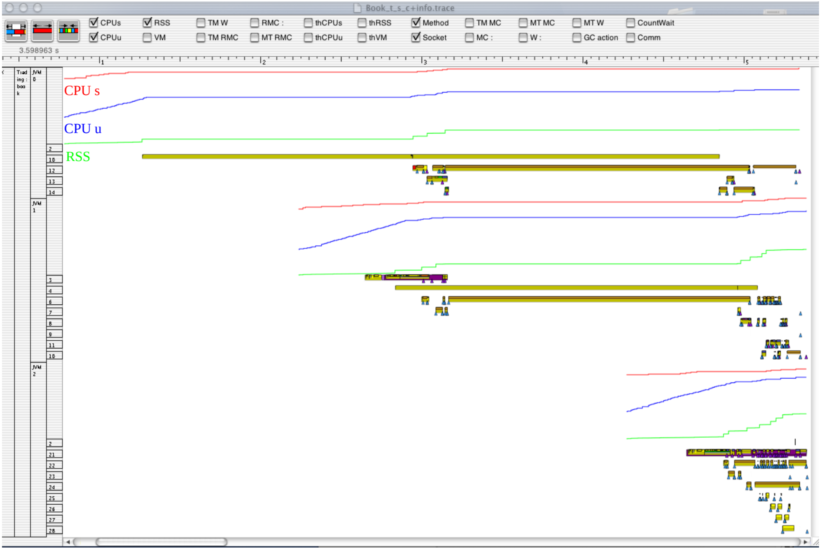
reading on socket events

sending through socket events

Distributed Middleware (4)



Distributed Middleware (5)



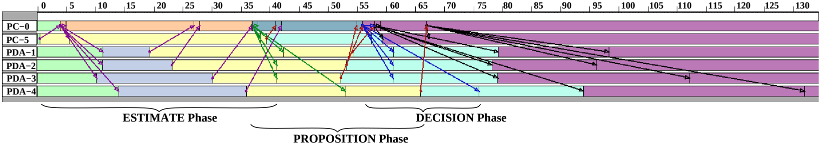
Book_t_s_c+info.trace

☑ CPUs ☑ RSS □ TM W □ RMC : □ thCPUs □ thRSS ☑ Method □ TM MC □ MT MC □ MT W □ CountWait

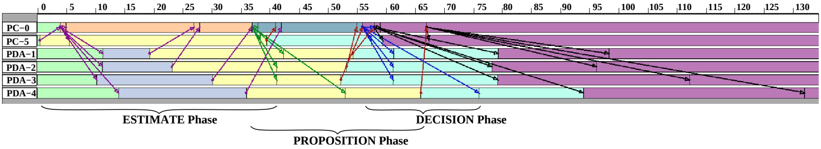
☑ CPUu □ VM □ TM RMC □ MT RMC □ thCPUu □ thVM ☑ Socket □ MC : □ W : □ GC action □ Comm

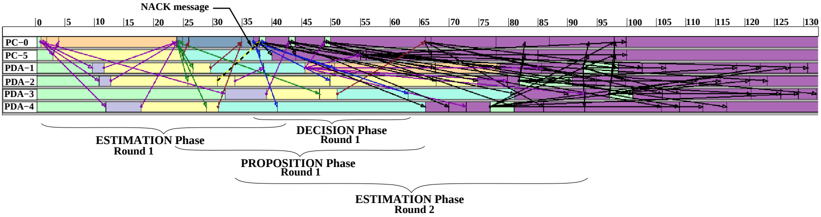
3.598963 s

Consensus in ad-hoc networks

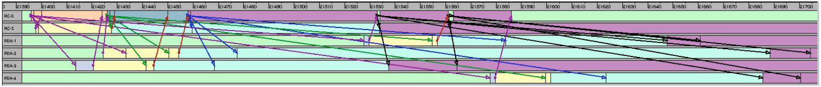


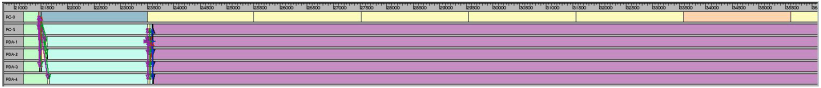
Consensus in ad-hoc networks

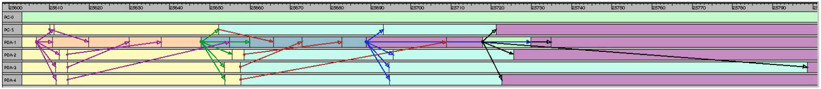




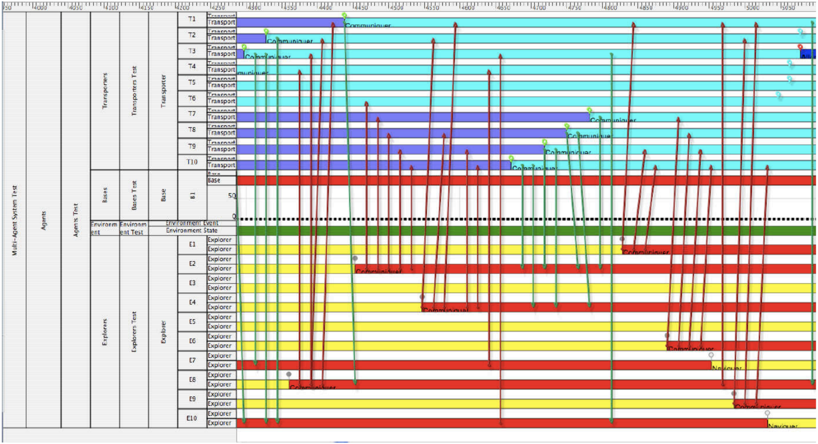
Coordinator Crashes







Multi-Agent Systems



1. Introduction
   - Motivations
   - Examples
2. Trace Fundamentals
   - Fundamentals
   - Pajé
3. Performance Analysis
   - Three-Dimensional Model
   - Temporal & Spatial Aggregation Model
4. Synthesis
   - Research directions

Performance Analysis

- Collect performance data
- Process collected data
- Visualize resulting data

Performance data collection

- Sampling

  let the system run, and from time to time, take a look at the state of the system

- Event-driven

  get informed of interesting changes in system state

Performance data collection

- Sampling

  let the system run, and from time to time, take a look at the state of the system
- Event-driven

  get informed of interesting changes in system state
  - Counting

    count number of times event happened
  - Timing

    accumulate time passed between pairs of events
  - Tracing

    register events for later processing

    usually also registers sampling data

Some tracing problems

- Clock synchronization
- Timer resolution
- Intrusion

  time / memory / I-O / influence in program behaviour
- Observability

  level of abstraction
- Matching independently captured events

  different machines or abstraction levels
- Amount of data
- Bufferization
- Trace file format

Trace data processing

- Merge / reorder
- Complement information
- Filter
- Reduce
- Prepare data for visualization

Pajé

- Generalize visualization tool, remove semantics
- Trace file contains
  - hierarchy of containers
  - each can contain combination of containers and visualizable entities
- Entities can contain extra data, used for filtering and reducing; user knows semantics
- Tool keeps original data and processed data, user chooses views

Pajé

Possible entity types

- **event** to represent events that happen at a certain instant
- **state** to represent that a given container was in a certain state during a certain period of time
- **link** to represent a relation between two containers that started at a certain instant and finished at a possibly different instant
- **variable** used to represent the evolution in time of a certain value associated to a container

1. Introduction
   - Motivations
   - Examples
2. Trace Fundamentals
   - Fundamentals
   - Pajé
3. Performance Analysis
   - Three-Dimensional Model
   - Temporal & Spatial Aggregation Model
4. Synthesis
   - Research directions

Outline

1 Introduction
- Motivations
- Examples

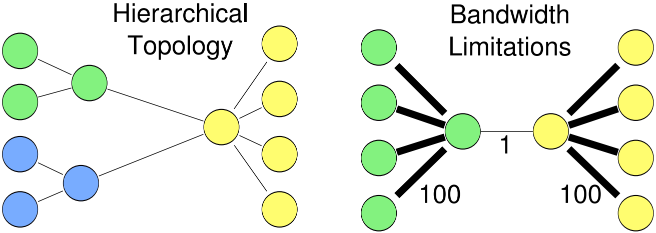
2 Trace Fundamentals
- Fundamentals
- Pajé

3 Performance Analysis
- Three-Dimensional Model
- Temporal & Spatial Aggregation Model

4 Synthesis
- Research directions

Performance Analysis

1. Analysis considering network topology



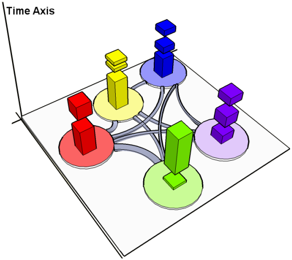
2. Large-scale analysis
   - How to analyze thousands of processes?
   - Temporal & Spatial Aggregation
   - Treemap representation

- Execution Platform: Grid'5000
  - Distributed resources in France
  - Highly hierarchical network organization
  - Limited heterogeneity – clusters

3D Model – Basics

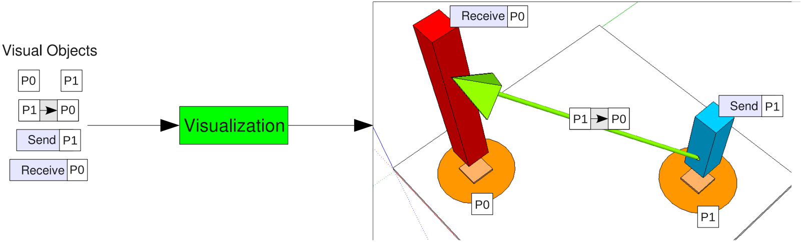
- Structural Representation – 2D
- Vertical dimension is time – 1D
  - Objects' Behavior Evolution
  - States and Links

- Interaction Techniques



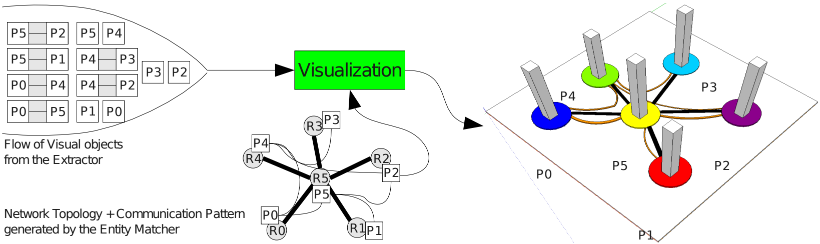
3D Model - Visualization

- How objects are represented in 3D



3D Model - Visualization

- How objects are represented in 3D
- Rendering the Network Topology + Comm. Pattern



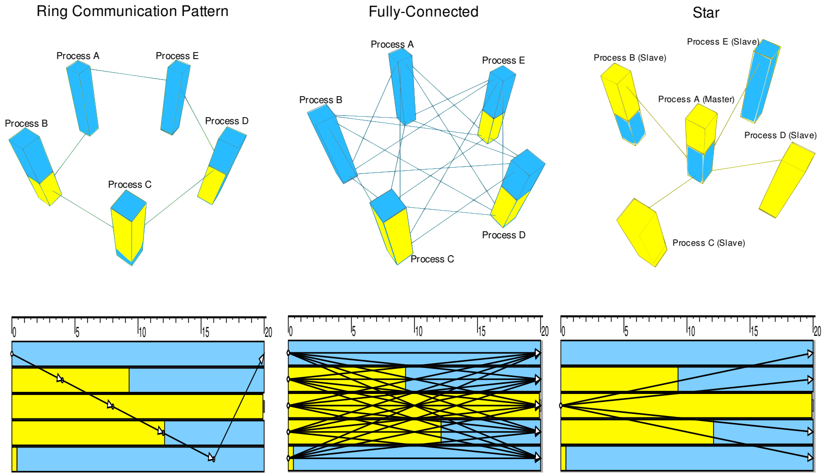
Flow of Visual objects from the Extractor

Visualization

Network Topology + Communication Pattern generated by the Entity Matcher

3D Visualization - Communication Patterns

- Differences from the space-time diagram



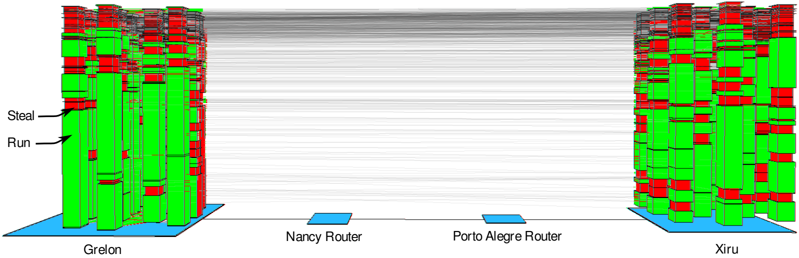
Ring Communication Pattern

Fully-Connected

Star

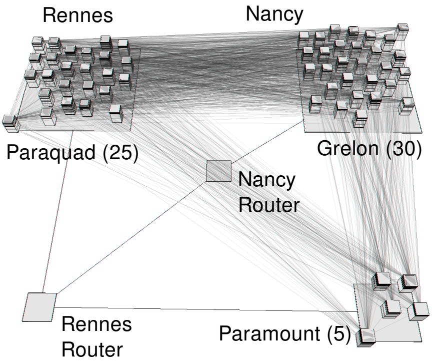
3D Visualization - KAAPI Trace

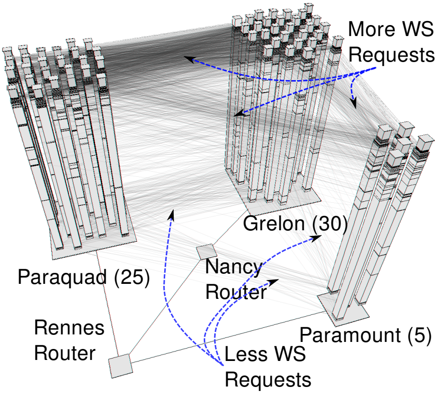
- Fibonacci Application
- 26 processes, two sites, two clusters
- Lines represent steal requests
- Different number of communication between clusters
  - beggining → big tasks, less communication
  - end → smaller tasks, more communication



3D Visualization - KAAPI Trace

- 60 processes, two sites, three clusters
- Total execution time of a KAAPI fibonacci application
- Observe number of requests in time

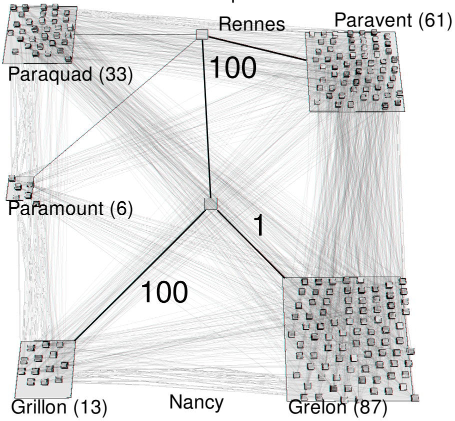




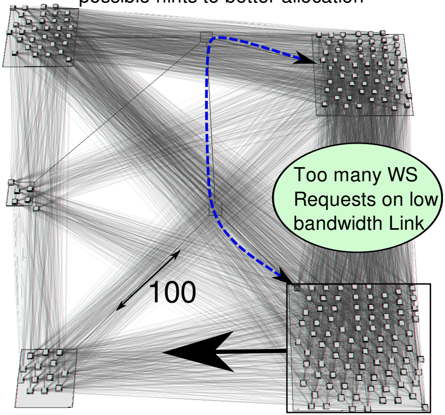
3D Visualization - KAAPI Trace

- 200 processes, 200 machines, two sites, five clusters
- Annotated manually with bandwidth limitations

Initial Execution of Application
with Link Properties

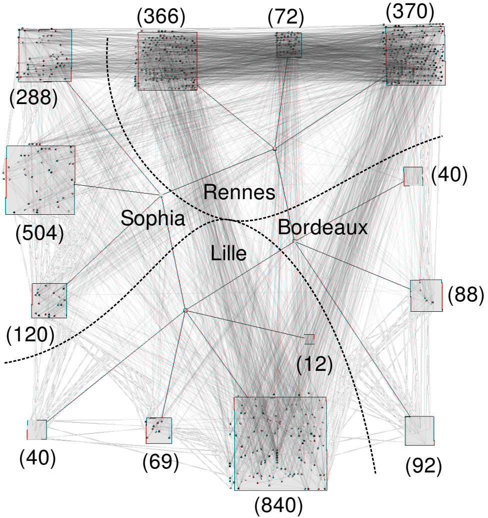


Interconnection becomes bottleneck,
possible hints to better allocation

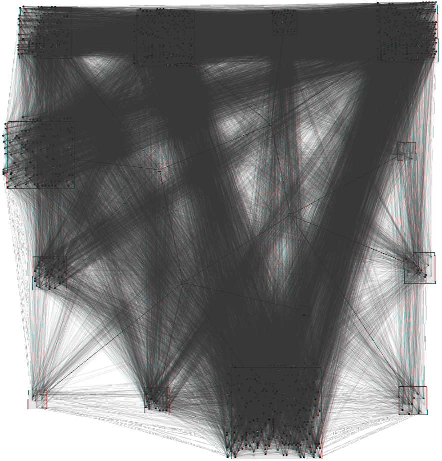


3D Visualization - KAAPI Trace

- 2900 processes, four sites, thirteen clusters



End of Execution



Outline

1 Introduction
- Motivations
- Examples

2 Trace Fundamentals
- Fundamentals
- Pajé

3 Performance Analysis
- Three-Dimensional Model
- Temporal & Spatial Aggregation Model

4 Synthesis
- Research directions

Temporal & Spatial Aggregation Model

- Enable large-scale trace analysis
- Visualy compare entities behavior
- Detect global and local characteristics

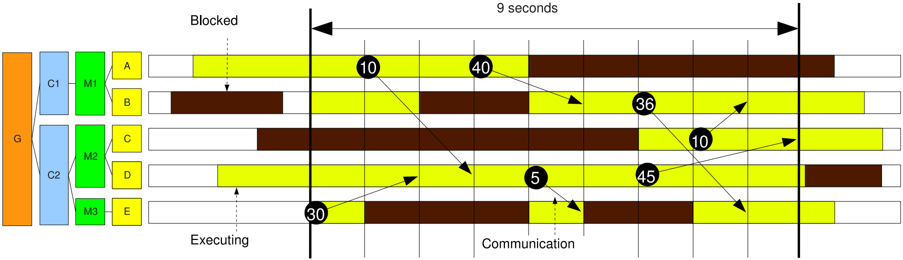
Steps of the Model

1. Hierarchical Monitoring Data
2. Temporal Aggregation
3. Spatial Aggregation
4. Treemap representation

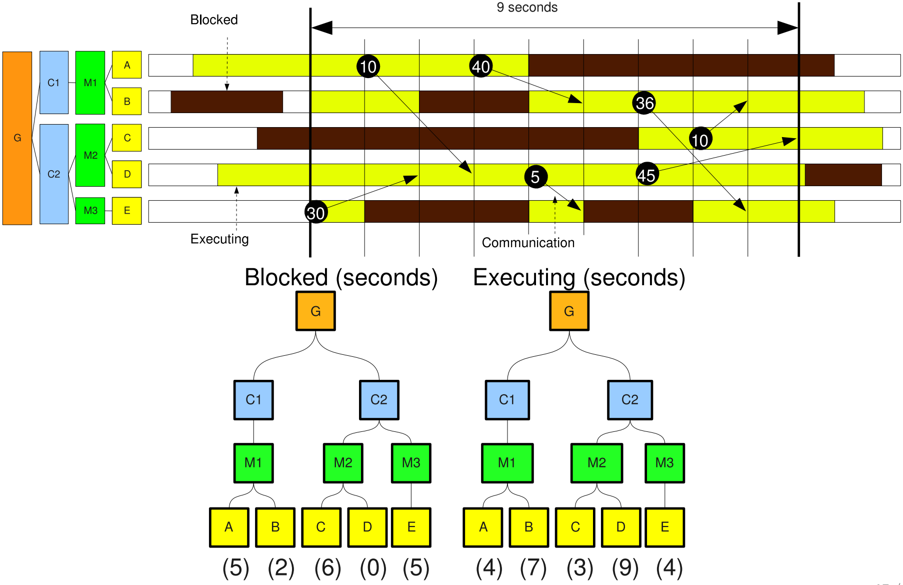
Temporal Aggregation - Basics

Objective: annotate leaves of the hierarchy

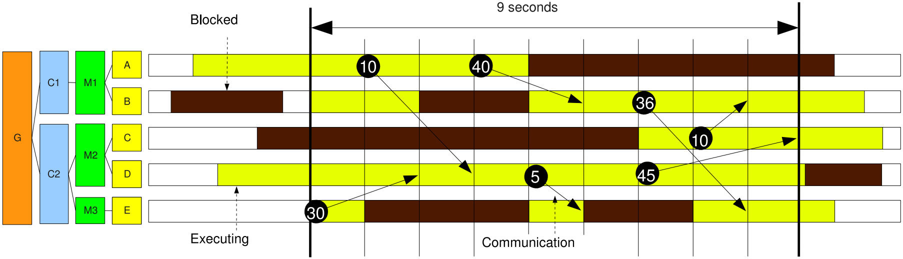
- Time-slice definition
- Summary of trace events on the interval
  - States, Variables, Links, Events, ...



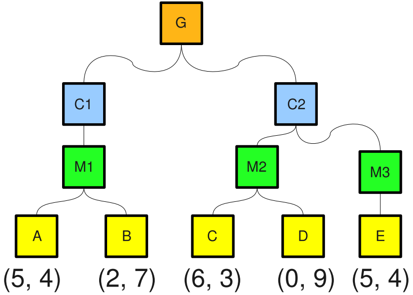
Temporal Aggregation - Example



Temporal Aggregation - Example



(Blocked, Executing)

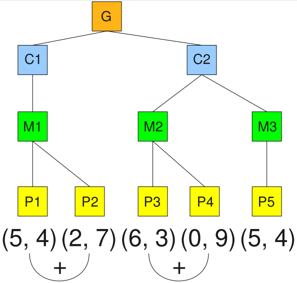


Spatial Aggregation

- Explore the hierarchical organization
- Create aggregated values at intermediary levels

Aggregation Functions

- add, subtract, multiply, divide, max, min, median, ...
- Depends on
    - what type of value the leaves have
    - the desired statistical result

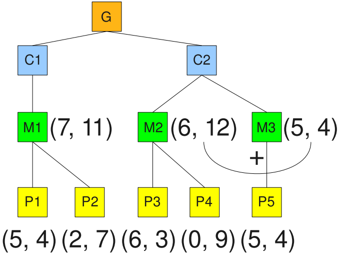


Spatial Aggregation

- Explore the hierarchical organization
- Create aggregated values at intermediary levels

Aggregation Functions

- add, subtract, multiply, divide, max, min, median, ...
- Depends on
  - what type of value the leaves have
  - the desired statistical result

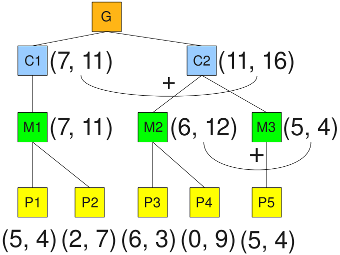


Spatial Aggregation

- Explore the hierarchical organization
- Create aggregated values at intermediary levels

Aggregation Functions

- add, subtract, multiply, divide, max, min, median, ...
- Depends on
  - what type of value the leaves have
  - the desired statistical result

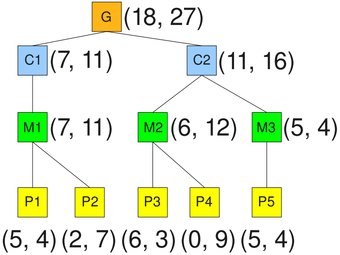


Spatial Aggregation

- Explore the hierarchical organization
- Create aggregated values at intermediary levels

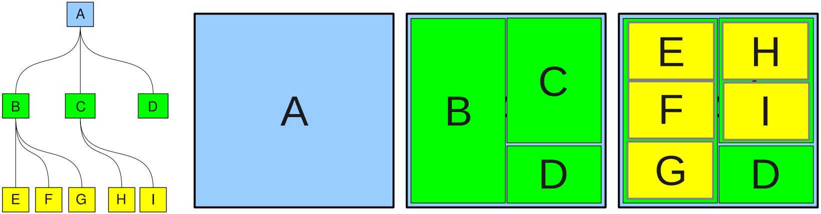
Aggregation Functions

- add, subtract, multiply, divide, max, min, median, ...
- Depends on
  - what type of value the leaves have
  - the desired statistical result



Visualization of the Approach - Treemaps

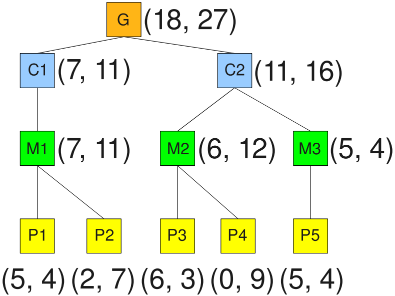
- Scalable hierarchical representation
- Top-down drawing algorithm
- For a given node, split screen space among children

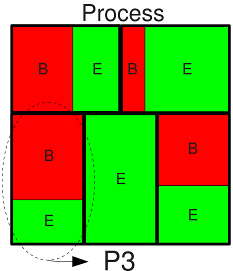


Original algorithm has several evolutions

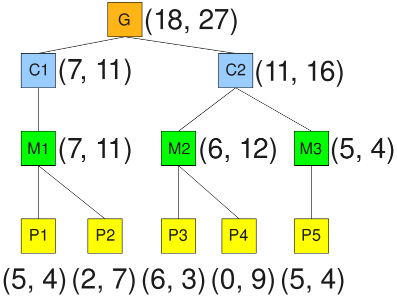
- Squarified treemap is used here
- Keeps rectangles as close to squares as possible

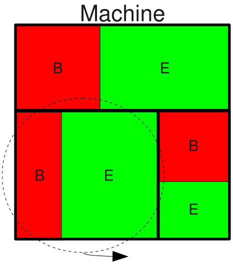
Treemap to view the Aggregated Hierarchy



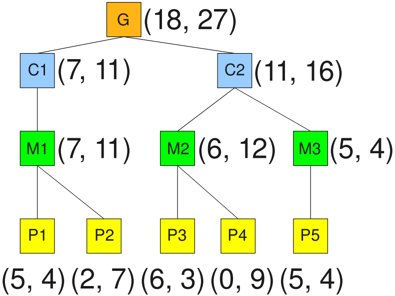


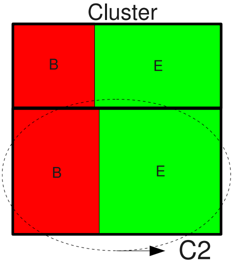
Treemap to view the Aggregated Hierarchy



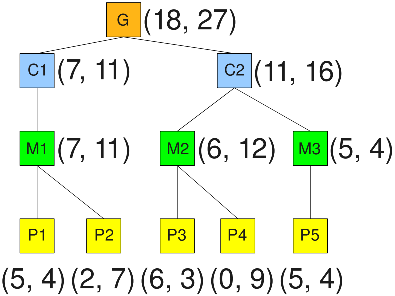


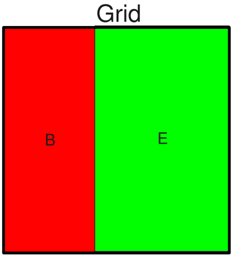
Treemap to view the Aggregated Hierarchy





Treemap to view the Aggregated Hierarchy

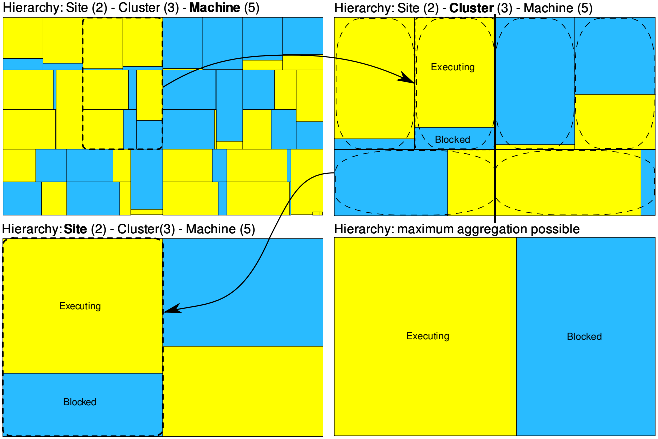




Treemap Visualization - Description

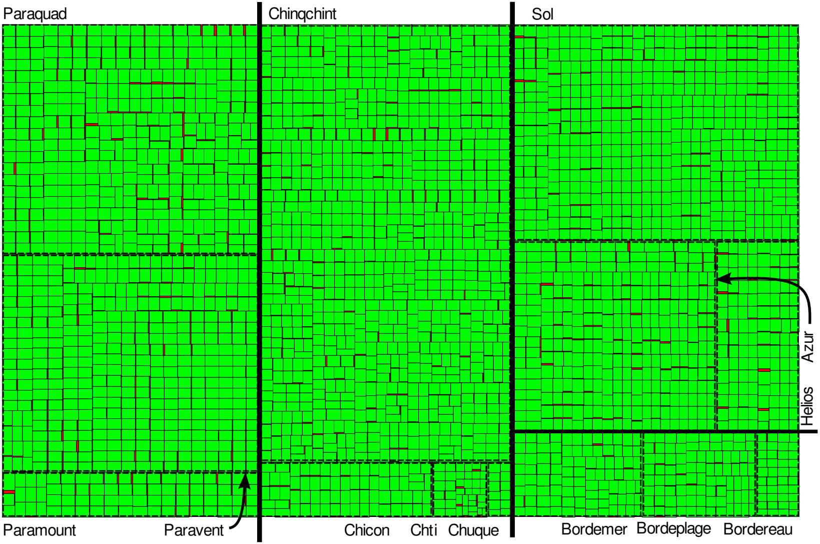
Time-Slice and Aggregated Hierarchies

- Interaction Techniques: mouse wheel, mouse over
- Detailed information is available in the status bar


Hierarchy: Site (2) - Cluster (3) - **Machine** (5)
Hierarchy: Site (2) - **Cluster** (3) - Machine (5)
Executing
Blocked
Hierarchy: **Site** (2) - Cluster(3) - Machine (5)
Executing
Blocked
Hierarchy: maximum aggregation possible
Executing
Blocked

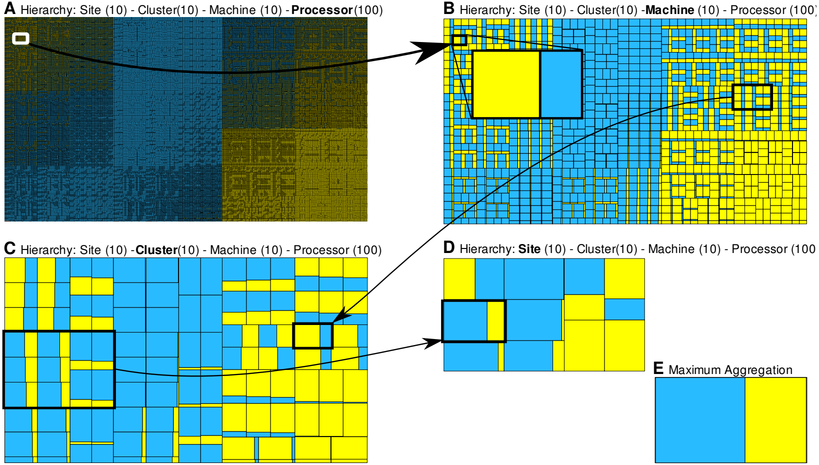
Treemap Visualization - KAAPI Trace

Run and **RSteal** states, 2900 processes, 310 processors



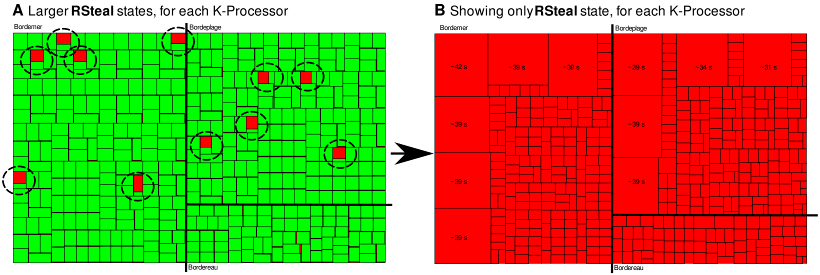
Treemap Visualization - Large-Scale

- Synthetic trace with 100 thousand processes
- Two states, four-level hierarchy



Treemap Visualization - KAAPI Trace

- 400 processes, 50 machines, one site
- 8 processes per machine
  - Overload of some machines with 2 CPUs
  - Unusual amount of time in Steal state
- Machines with 4 CPUs show normal behavior

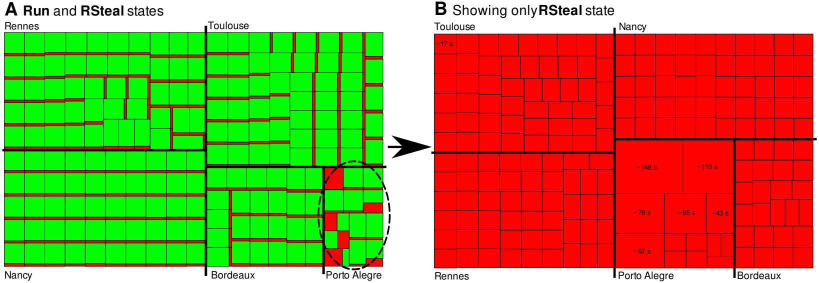


A Larger **RSteal** states, for each K-Processor

B Showing only **RSteal** state, for each K-Processor

Treemap Visualization - KAAPI Trace

- 188 processes, 188 machines, five sites
- Different behavior at Porto Alegre
- Probably due to the interconnection
  - Latency for Grid'5000 in France: ~10 ms
  - Latency between Porto Alegre and France: ~300 ms
- More time spent in work stealing functions

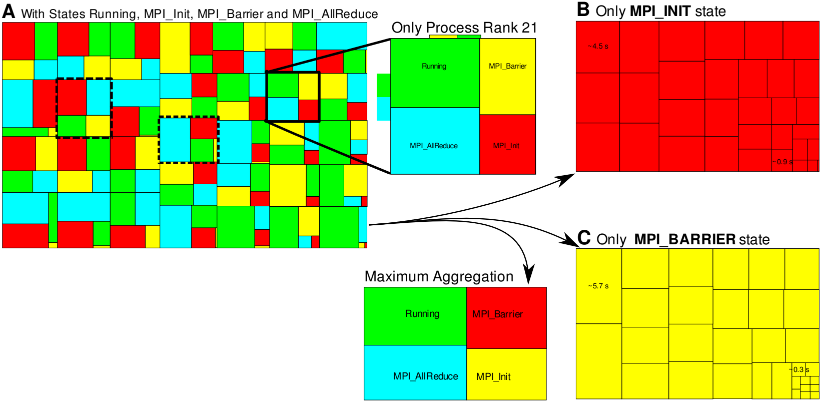


A **Run** and **RSteal** states

B Showing only **RSteal** state

Treemap Visualization - MPI Trace

- Traces from the EP application – NAS Benchmark
- 32 processes – time spent in each MPI operation
- Init and Barrier views indicate a linear implementation



A With States Running, MPI_Init, MPI_Barrier and MPI_AllReduce

Only Process Rank 21

B Only MPI_INIT state

C Only MPI_BARRIER state

Maximum Aggregation

1. Introduction
   - Motivations
   - Examples
2. Trace Fundamentals
   - Fundamentals
   - Pajé
3. Performance Analysis
   - Three-Dimensional Model
   - Temporal & Spatial Aggregation Model
4. Synthesis
   - Research directions

Conclusion

Concepts

- Trace of parallel/distributed applications
- Multi-level trace
- Structural informations

Conclusion

Concepts

- Trace of parallel/distributed applications
- Multi-level trace
- Structural informations

Algorithmic solutions

- Trace collection (quality of tracers, time estimation...)
- Simulation engine based on the state/event model
- Visualization engine (interactivity, extensibility, scalability)

Conclusion

Concepts

- Trace of parallel/distributed applications
- Multi-level trace
- Structural informations

Algorithmic solutions

- Trace collection (quality of tracers, time estimation...)
- Simulation engine based on the state/event model
- Visualization engine (interactivity, extensibility, scalability)

Case studies

- Parallel systems (MPI, Kaapi,...) Distributed middlewares, Wireless networks, Multi-agent systems,...
- Industrial application :Embedded systems, Jboss analysis, resilient protocols

Research directions

Scalability

- Aggregation : in time, space, structure (level, operators,...)
- Clustering : criteria of clustering

Research directions

Scalability

- Aggregation : in time, space, structure (level, operators,...)
- Clustering : criteria of clustering

User capabilities

- Observation environment : instrumentation, information synthesis
- Visualization environment : visual objects manipulation (time, objects, or structure selection), coherent multiple views

Research directions

Scalability

- Aggregation : in time, space, structure (level, operators,...)
- Clustering : criteria of clustering

User capabilities

- Observation environment : instrumentation, information synthesis
- Visualization environment : visual objects manipulation (time, objects, or structure selection), coherent multiple views

Global properties and trace mining

- Query language for traces (filtering/aggregation/selection)
- Automatic data mining in the trace (patterns, properties)

Bibliography

Main papers in the domain

- **Performance Measurement Intrusion and Perturbation Analysis**, Malony, A. D., Reed, A., and Wijshoff, H.A.G., IEEE TPDS 3(4) 1992
- **What to Draw? When to Draw? An Essay on Parallel Program Visualization**, Miller, B.P., JPDC, 18, 1993
- **SvPablo: A Multi-Language Architecture-Independent Performance Analysis System** DeRose, L. and Reed, D.A.,ICPP 1999
- **VAMPIR: Visualization and Analysis of MPI Resources** Nagel, W.E., Arnold, A. Weber, M., Hoppe, H-C., and Solchenbach, K. Supercomputer 1996

Some of our papers

- **Monitoring Parallel Programs for Performance Tuning in Cluster Environments** Chassin de Kergommeaux, J. and Maillet, E. and Vincent, J.-M., chap 6 in book Parallel Program Development for Cluster Computing: Methodology, Tools and Integrated Environments, Nova Science, 2001
- **Visualisation interactive et extensible de programmes parallèles base de processus lgers** Benhur de Oliveira Stein PhD 1999
- **Observations et analyses quantitatives multi-niveaux dapplications à objets réparties** Franois-Gael Ottogalli 2001
- **Some Visualization Model applied to the Analysis of Parallel Applications** Lucas Mello Schnorr 2009

Thanks for your attention
The slides of the tutorial will be at
`http://www.inf.ufrgs.br/~lmschnorr`

Pajé - `http://forge.ow2.org/projects/paje/`
Triva - `http://triva.gforge.inria.fr/`

Thanks to : Brazil/France collaboration projects, CAPES, CNPq, COFECUB, CNRS, INRIA, UFSM, UFRGS, UJF, Grenoble INP, and many colleagues providing nice ideas, improving the code and sharing drinks with us