Some Visualization Models applied to the Analysis of Parallel Applications

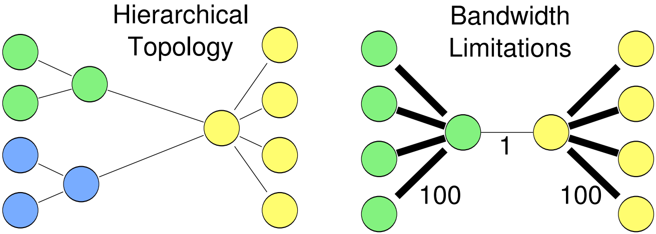
Lucas Mello Schnorr

Advisors:
Philippe O. A. Navaux & Denis Trystram & Guillaume Huard

Federal University of Rio Grande do Sul, Brazil
Grenoble Institute of Technology, France

Introduction - Context

- Distributed Systems → Grids
- Grid Interconnection and Scalability
  - Topology and Connectivity
  - Performance: bandwidth and latency
  - New resources can be added very easily

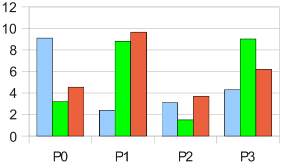


- Influence in the application execution
- Visualization – Performance Analysis

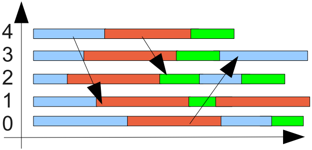
Introduction - Existing Tools/Techniques

- Statistical Techniques
  - ParaGraph (1990) – bar charts, utilization Count
  - Pablo (1993) – bar charts + 3D scatter plot
- Behavioral Techniques
  - Vampir (1996) – time-line system view
  - Jumpshot (1999), Pajé (2000) – space-time
- Structural Techniques
  - ParaGraph (1990) – network display / hypercube

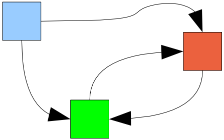
Statistical



Behavioral



Structural



Introduction - Problem Identification

- Lack of a network-aware analysis
  - Difficult to analyze using space-time views
  - Structural techniques undeveloped
- Problems of visualization scalability
  - Visualization techniques limitations reached
  - Analysis are limited to hundreds of entities

Desirable Characteristics for Application Analysis
$\rightarrow$ The Objectives

- Consider network properties
- Visualization scalability in the analysis

Introduction - The Thesis Approach

- Explore techniques from Information Visualization
- Context of parallel application analysis
  - Grid resources
  - Thread/Process parallel applications

Proposed Visualization Models

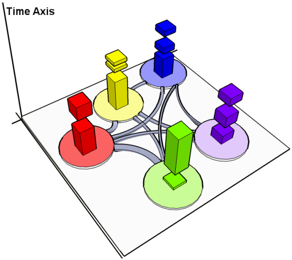
- Behavioral and Structural/Statistical (3D)
  - Communication Pattern
  - Network topology + Communication Pattern
  - Logical representation
- Visual Aggregation
  - Large-scale traces
  - Local and Global summaries

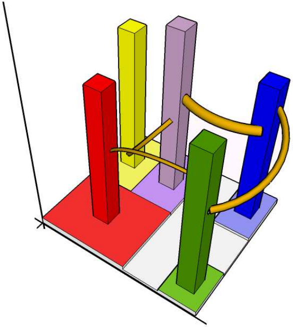
Outline

3D Model - Visual Conception

- Resources represented in 2D
  - Structural (e.g. a graph)
  - Statistical
- Vertical dimension is time
  - Objects' Behavior Evolution
  - States and Links

- Interaction Techniques
  - Notion of a Camera
  - Rotation
  - Translation
  - Objects Animation
  - Replay step-by-step





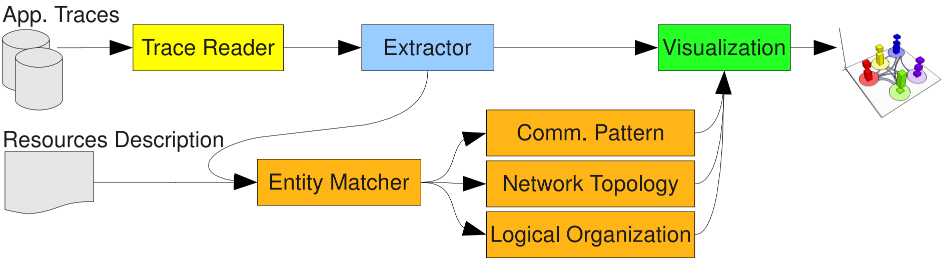
3D Model - Differences from existing tools

- 3D Statistical Representation
  - Pablo → 3D Scatter Plot
  - Paradyn → 3D Terrain
  - ParaProf → Triang Mesh, 3D Bar and 3D Scatter Plot
- 3D Behavioral Representation
  - ParaProf → 2 metrics and time
  - Virtue → the time-tunnel view

Our Approach

- Presence of a timeline to show objects' evolution
- Multiple Configurations in the visualization base

3D Model - Abstract Component Model



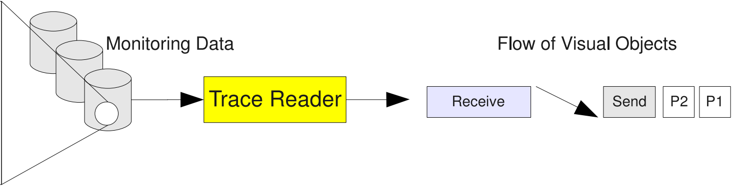
Input Data

- Application Traces
  - Timestamp-based events
  - Behavior registered
- Resources Description
  - Network topology: graph
  - Logical resource organization: tree

3D Model - The Trace Reader

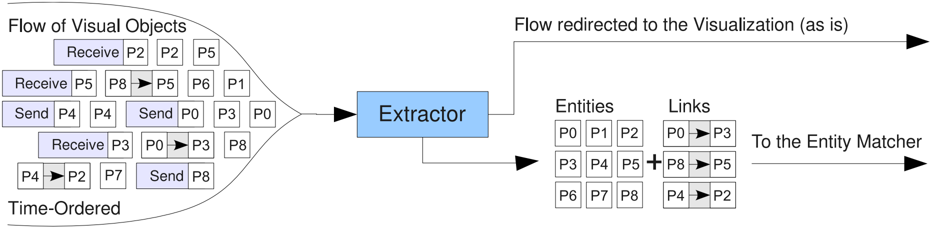
- Deals directly with application traces and events
- Only trace-dependent part of the model
- Transform events into high-level visual objects
  - Container → Entities
  - State/Variable/Event → Evolution
  - Link → Communications
- No semantics → Visualization is generic

Time	Process	OperationType
0.0	1	start
0.1	2	start
0.3	2	receive_start
0.5	1	send_start
0.7	1	msg_send
1.0	1	send_end
1.3	2	msg_receive
1.8	2	receive_end
...		



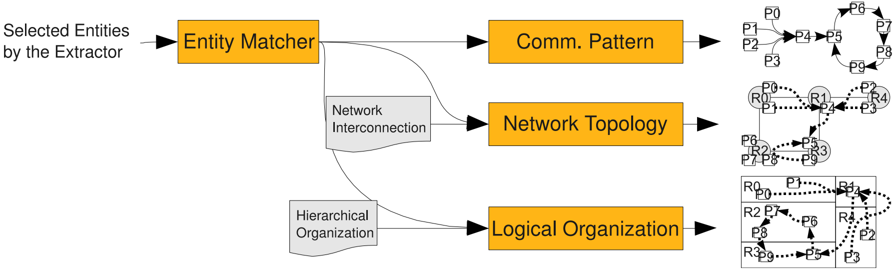
3D Model - The Extractor

- Supply entity matcher needs: links and entities
- Attribute entities with location data
  - where a process is executed
  - which process a thread belongs to
- Input is also redirected to the Visualization module



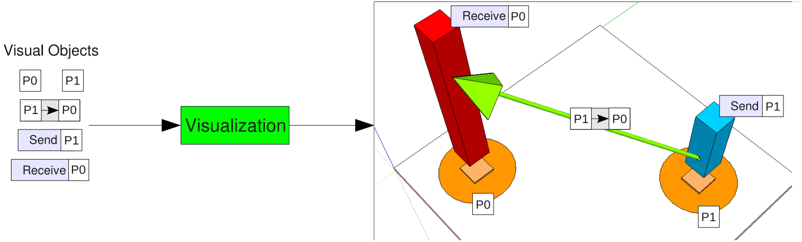
3D Model - The Entity Matcher

- Responsible for the Visualization Base layout
- Three possibilities of configuration are proposed
  - Communication Pattern (deadlocks, ...)
  - Network Topology (network utilization, routes, ..)
  - Logical Organization (load balancing, ...)



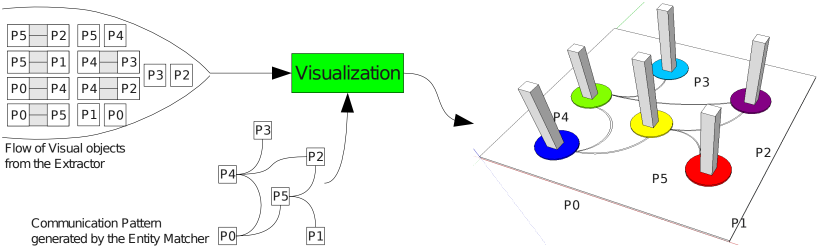
3D Model - Visualization

- How the visual objects are represented in 3D



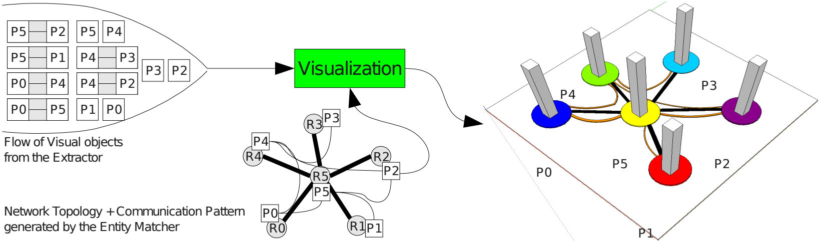
3D Model - Visualization

- How the visual objects are represented in 3D
- Rendering the visualization base
  - Application Communication Pattern



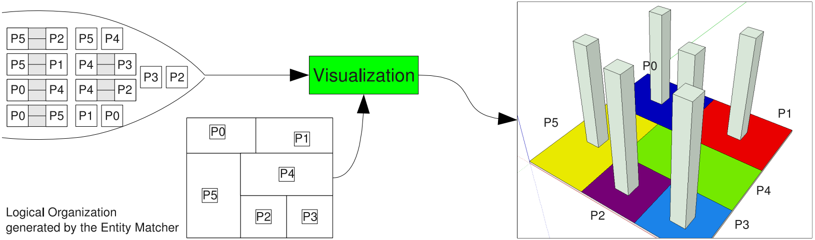
3D Model - Visualization

- How the visual objects are represented in 3D
- Rendering the visualization base
  - Application Communication Pattern
  - Network Topology + App. Communication Pattern



3D Model - Visualization

- How the visual objects are represented in 3D
- Rendering the visualization base
  - Application Communication Pattern
  - Network Topology + App. Communication Pattern
  - Logical Organization of Resources



Outline

Aggregation Model - Overview

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

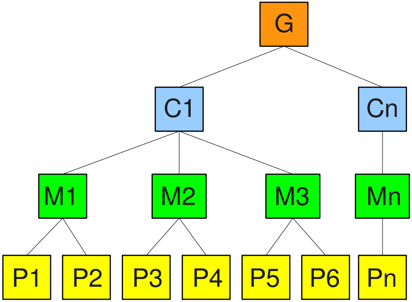
Steps of the Model

1. Hierarchical Monitoring Data
2. Time-Slice algorithm (temporal integration)
3. Aggregation model (spatial integration)
4. Treemap representation

- Visualization differences from existing tools
  - PlanetLab's CoVisualize → resources
  - Treemap for Workload Visualization [Stephen 2003]
  - Lack of configurable time intervals, aggregated data

Hierarchical Monitoring Data

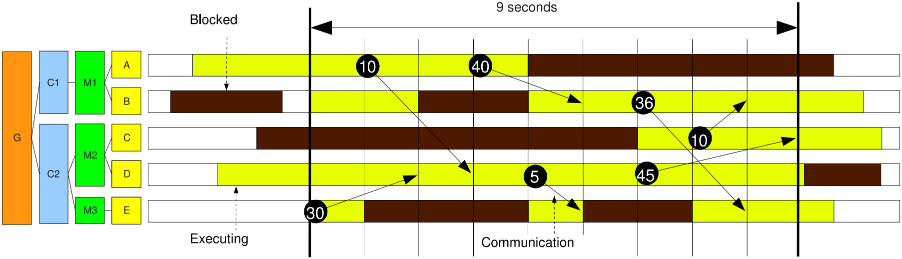
- Monitoring systems register entities behavior
- Entities can be processes and threads
- They can be organized as a hierarchy
  - Logical hierarchy
  - Geographical Location hierarchy
  - Other possibilities: libraries, components
- Grid'5000 example



Time-Slice Algorithm - Basics

Objective: annotate leaf nodes of the hierarchy

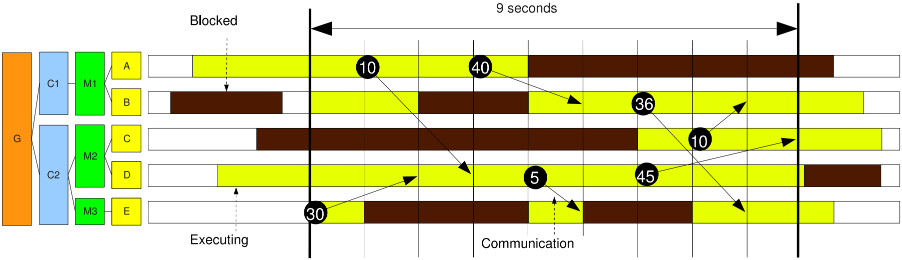
- Time-slice definition
- Summary of trace events on the interval
  - States, Variables, Links, Events, ...

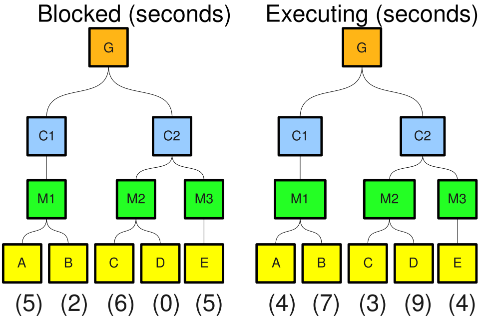


Output of the Algorithm

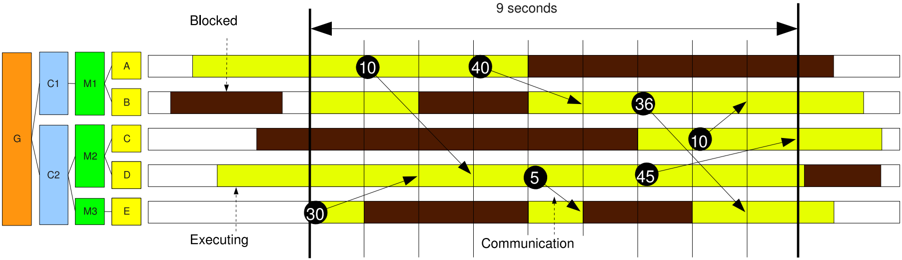
- Hierarchy of input + computed values on leaves

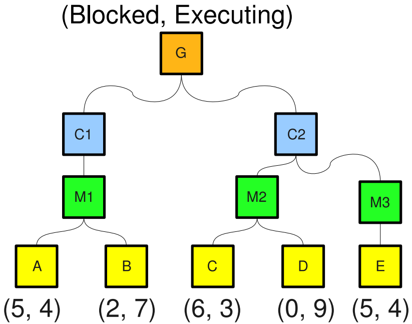
Time-Slice Algorithm - Example





Time-Slice Algorithm - Example



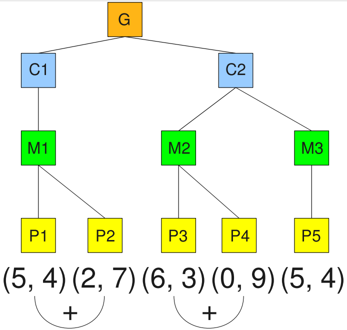


Aggregation Model

- Objective: 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

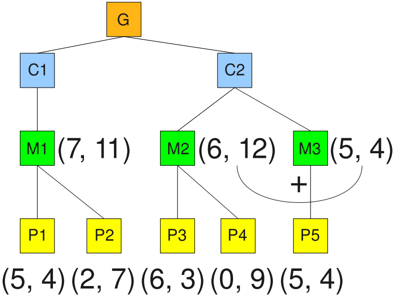


Aggregation Model

- Objective: 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

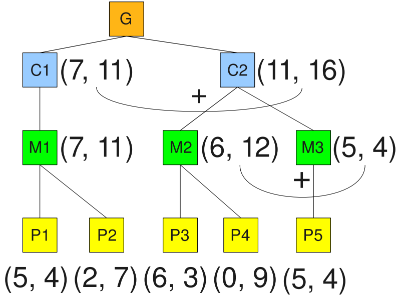


Aggregation Model

- Objective: 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

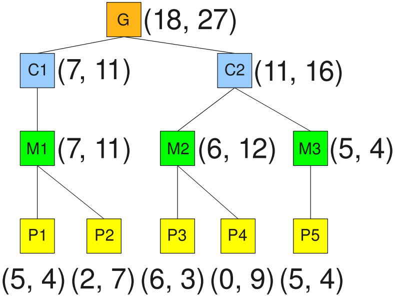


Aggregation Model

- Objective: aggregated values at intermediary levels

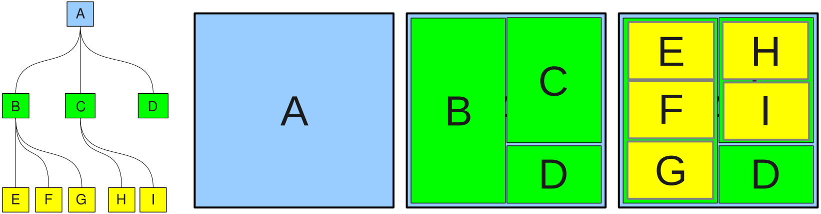
Aggregation Functions

- 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

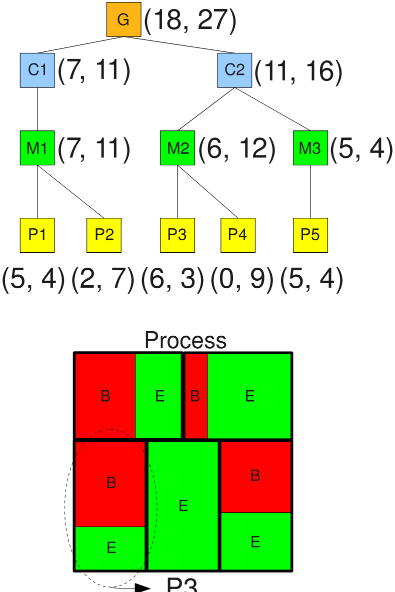
- Technique created in 1991
- Scalable hierarchical representation
- Algorithm
  - Top-down drawing
  - For a given node, split screen space among children



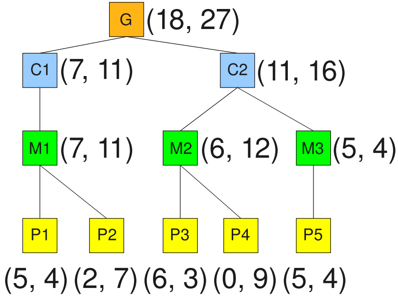
Original algorithm has several evolutions

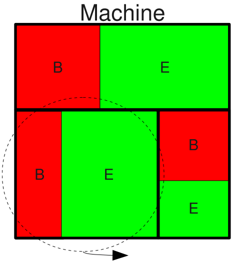
- Squarified treemap is used here
  $\rightarrow$ Keeps rectangles as close to squares as possible

Treemap to view the Aggregated Hierarchy

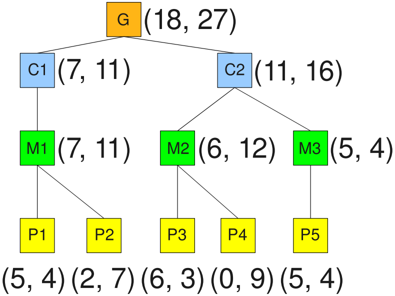


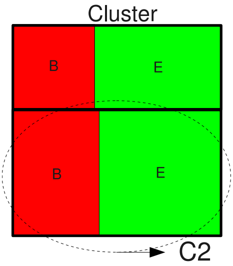
Treemap to view the Aggregated Hierarchy



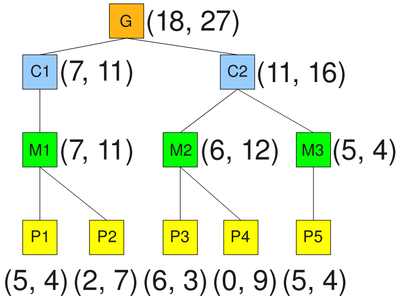


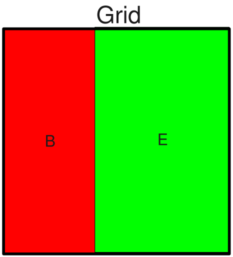
Treemap to view the Aggregated Hierarchy





Treemap to view the Aggregated Hierarchy

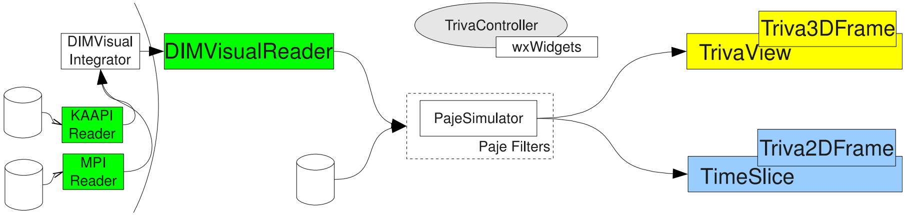




Outline

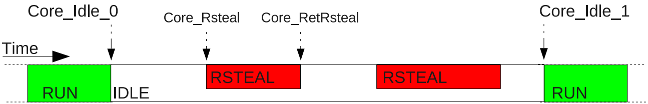
Triva Prototype Implementation

- Developed in Objective-C and C++
- Combine several existing tools
  - DIMVisual library
  - Pajé Components (the Simulator)
  - Graphviz, Ogre3D, wxWidgets
- Performance evaluation of Pajé
  - Able to handle large-scale traces
  - Small response-time
  - Memory limitations

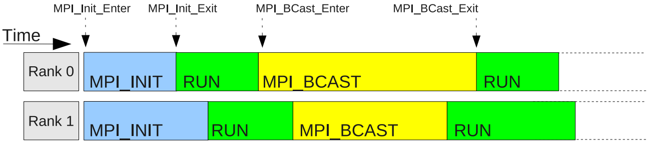


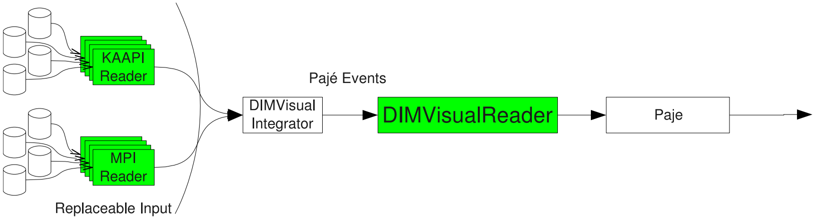
DIMVisualReader - Trace Reader

- Built-in instrumentation of KAAPI library



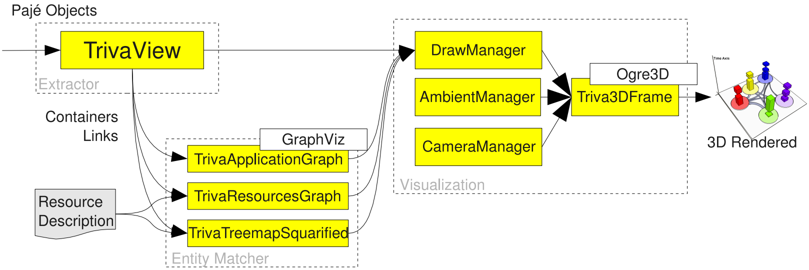
- MPIRastro wrapper for MPI applications





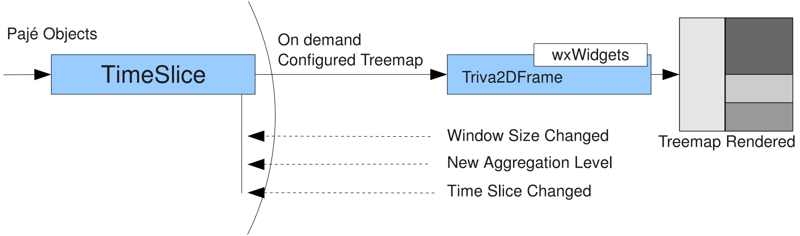
TrivaView - The 3D Approach

- Model: Extractor, Entity Matcher & Visualization
- Interaction Techniques (Ambient, CameraManager)
- Base configuration
  - Application Comm. Pattern created with GraphViz
  - Network Topology description (dot format)
  - Logical Organization (plist format)
- Placement on the Visualization Base
- Rendering the 3D Timestamped Pajé Objects



TimeSliceView - The Aggregation Model

- Only two components
  - TimeSlice Filter
  - Triva2DFrame
- Time-Slice Algorithm and Aggregation Model
- Implementation of the Squarified Treemap Algorithm
- Drawing the rectangles with the wxWidgets



Outline

Results

- Different application traces are used as input
- Results are composed of screenshots of the prototype

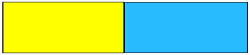
Objective

- Check if 3D visualizations enable a better understanding of traces with the network topology
- Check if large-scale analysis are possible with the aggregation model

- Traces Description
- 3D Visualization
- Treemap Visualization

Results - Trace Description

- Synthetic traces
  - Large-scale hierarchies (up to 100 thousand)
  - Typical Communication Patterns



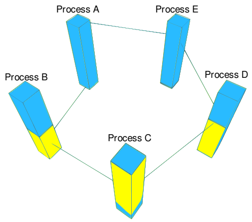
- Real traces
  - KAAPI Traces

  - MPI Traces
- Grid’5000 platform in France
- Xiru Cluster at Porto Alegre

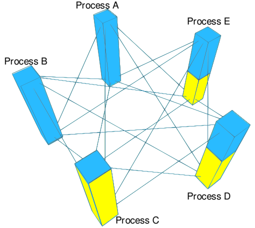
3D Visualization - Communication Patterns

- Differences from the space-time diagram
  → 3D enables Graph-like representations
  → with time evolution

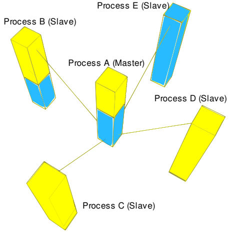
Ring Communication Pattern

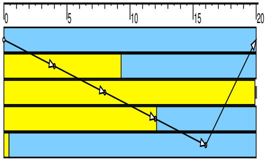


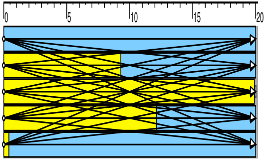
Fully-Connected

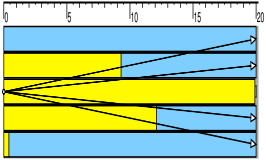


Star



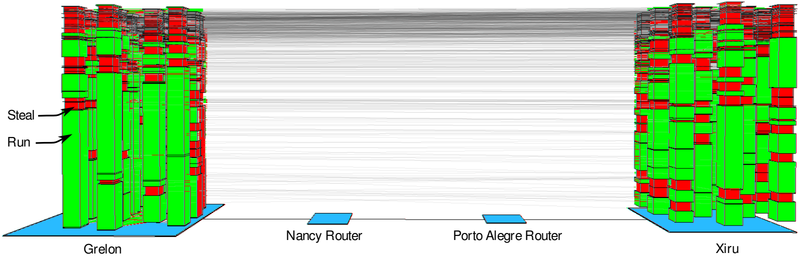






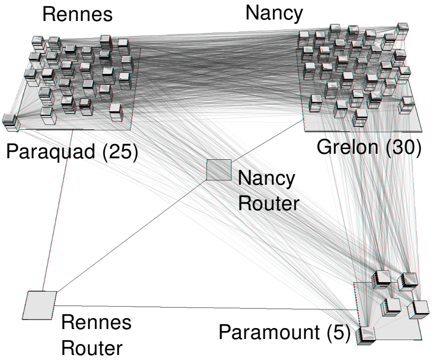
3D Visualization - KAAPI Trace

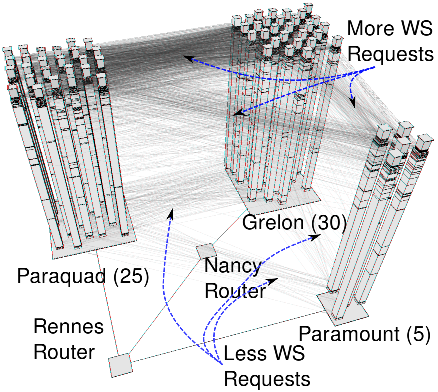
- Fibonacci Application
- 26 processes, two sites, two clusters
- Lines represent steal requests
- Different number of communication between clusters
  - beggining → big tasks, less communication
  - end → smaller tasks, more communication



3D Visualization - KAAPI Trace

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

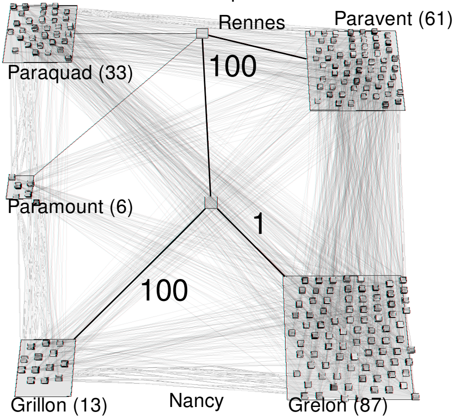




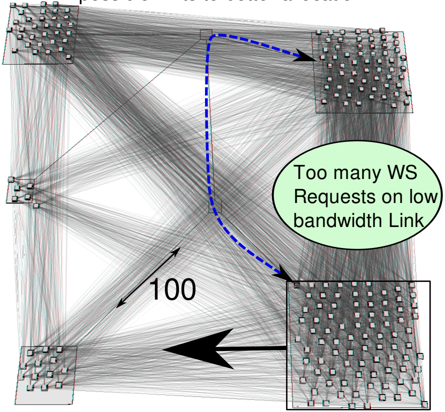
3D Visualization - KAAPI Trace

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

Initial Execution of Application
with Link Properties

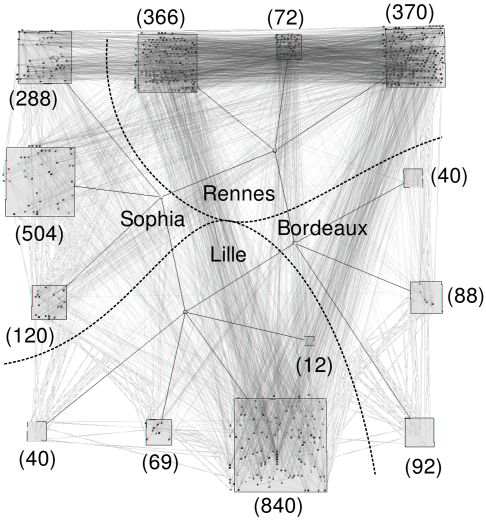


Interconnection becomes bottleneck,
possible hints to better allocation

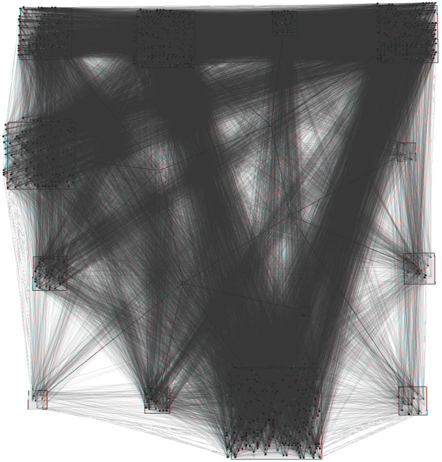


3D Visualization - KAAPI Trace

- 2900 processes, four sites, thirteen clusters



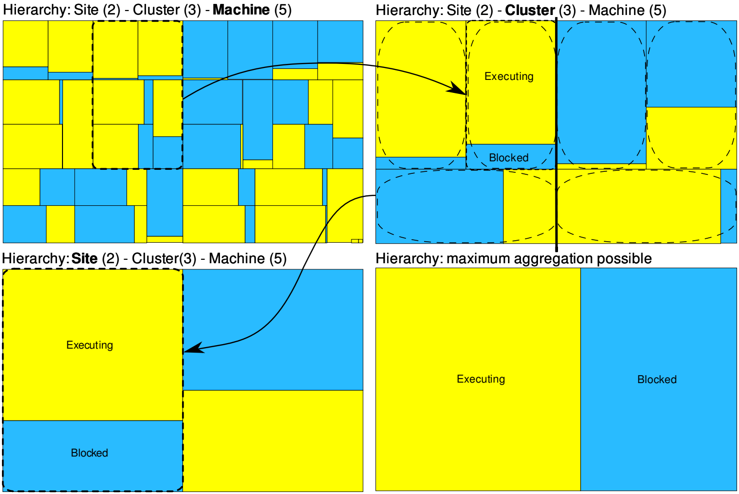
End of Execution



Treemap Visualization - Description

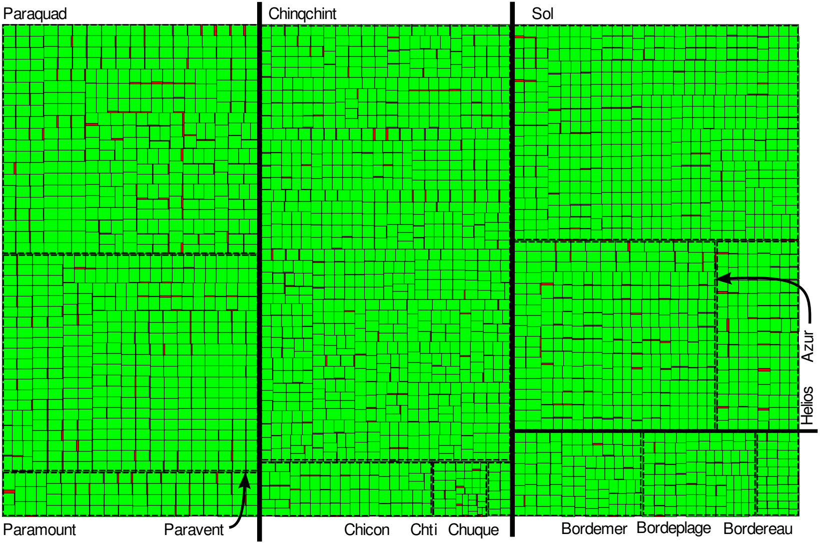
Time-Slice and Aggregated Hierarchies

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



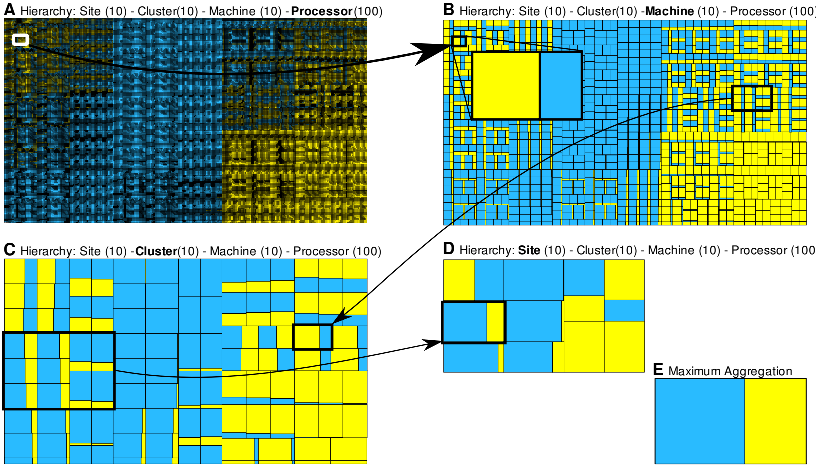
Treemap Visualization - KAAPI Trace

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



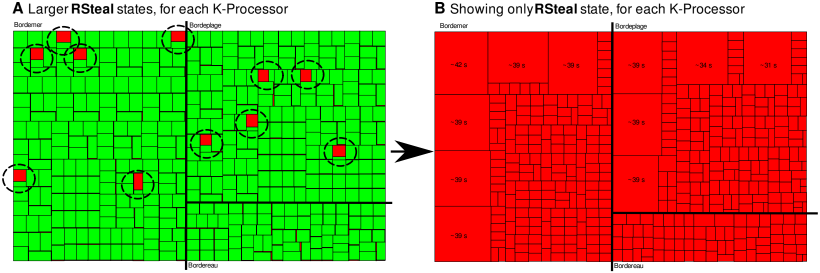
Treemap Visualization - Large-Scale

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



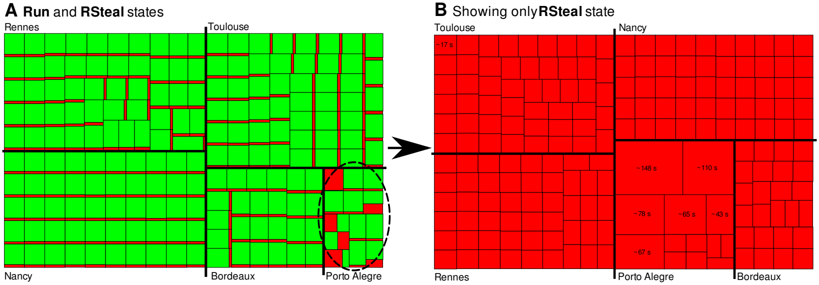
Treemap Visualization - KAAPI Trace

- 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



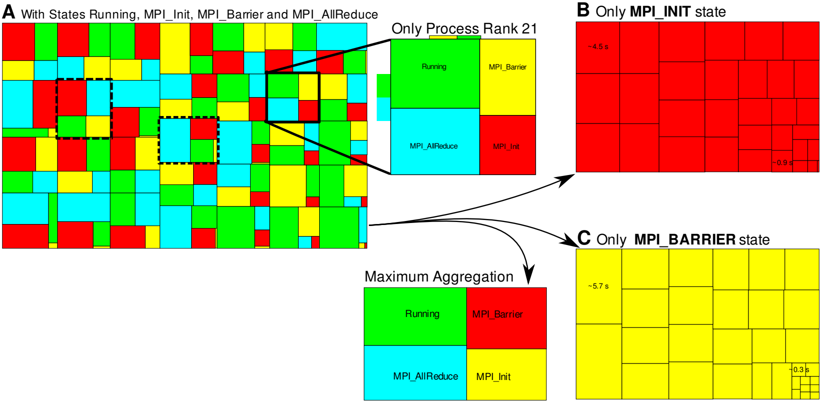
Treemap Visualization - KAAPI Trace

- 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



Treemap Visualization - MPI Trace

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



Conclusions

The problem identified in the Thesis

- Lack of structural visualization analysis
- Visualization scalability

Main Achievements

- Behavioral with Structural/Statistical Model (3D)
  - Analysis considering network structure
  - Experiments using Grid'5000 platform
  - Identification of behavior in KAAPI work stealing
- Time-Slice Technique & Aggregation Model
  - Validated with real-scenario with 2900 processes
  - Tested with synthetic traces up to 100K processes
  - Load-balance efficiency / global and local summaries

Perspectives and Implications

- Perspectives
  - Show aggregated objects in the 3D visualization
  - Other types of information for the time-slice technique
  - Use of other aggregation functions
  - Aggregation model to merge communication patterns
  - ...
- Implications
  - Better understanding of parallel applications
    $\rightarrow$ consider execution environment details
    $\rightarrow$ large-scale visual analysis
  - Re-thinking behavioral visualization
    $\rightarrow$ Do we need a timeline in representations?
    $\rightarrow$ Aggregated data
  - Use of information visualization techniques

Publications

Future Generation Computer Systems Journal

Sbac 2009

Grid 2008

CCGrid 2009

Submit to Journal of Grid Computing

Acknowledgements

- Thesis financed with scholarships by
  - CAPES and CNPq
  - CAPES/Cofecub – Project 4602/06-4
- Thanks to
  - Advisors: Navaux, Guillaume and Denis
  - Family and friends