Analysis of distributed systems simulation through trace visualization

Lucas Mello Schnorr (CNRS)
LIG-MESCAL, Grenoble, France

INF-UFRGS – Porto Alegre, Brazil

August 9th, 2012

Introduction

- Distributed and parallel systems research
  - Best algorithm for parallel file systems?
    $\rightarrow$ considering reads, writes and multiple clients
  - Which scheduling algorithm works best? $\rightarrow$
    Publish-subscribe, centralized, distributed

Introduction

- Distributed and parallel systems research
  - Best algorithm for parallel file systems?
    $\rightarrow$ considering reads, writes and multiple clients
  - Which scheduling algorithm works best? $\rightarrow$
    Publish-subscribe, centralized, distributed
- Experimental research (no theoretical or simplistic models)
  - Write a prototype of the algorithm
  - Run experiments on selected platforms

Introduction

- Distributed and parallel systems research
  - Best algorithm for parallel file systems?
    $\rightarrow$ considering reads, writes and multiple clients
  - Which scheduling algorithm works best? $\rightarrow$
    Publish-subscribe, centralized, distributed
- Experimental research (no theoretical or simplistic models)
  - Write a prototype of the algorithm
  - Run experiments on selected platforms

Experiments analysis $\rightarrow$ results

1. Understand performance for a given platform
2. Get inspiration for changes, improvements
3. Test different hypothesis, assumption check

Introduction

Registering distributed system behavior

- Register behavior through timestamped events
- Very detailed in time $\rightarrow$ micro to nanoseconds
- Many observed entities (processes, threads)

Introduction

Registering distributed system behavior

- Register behavior through timestamped events
- Very detailed in time $\rightarrow$ micro to nanoseconds
- Many observed entities (processes, threads)

– Simulation –

Introduction

Registering distributed system behavior

- Register behavior through timestamped events
- Very detailed in time $\rightarrow$ micro to nanoseconds
- Many observed entities (processes, threads)

– Simulation –

Registering simulation behavior (of distributed systems)

- Enhanced traces with all kinds of behavioral data
  $\rightarrow$ Depends on the simulation models
  $\rightarrow$ Some data is hard to trace in real life

Introduction

Registering distributed system behavior

- Register behavior through timestamped events
- Very detailed in time $\rightarrow$ micro to nanoseconds
- Many observed entities (processes, threads)

– Simulation –

Registering simulation behavior (of distributed systems)

- Enhanced traces with all kinds of behavioral data
  $\rightarrow$ Depends on the simulation models
  $\rightarrow$ Some data is hard to trace in real life
- Zero intrusion (the world stands still while data gathering)

Approach

- Getting better traces (from the simulated world)
  $\rightarrow$ for an enhanced analysis
  - Register resource utilization by categories

- Exploratory trace visualization
  $\rightarrow$ scalable in time and space, interactive
  - Space/Time trace aggregation
  - Hierarchical Graph visualization

Outline

Getting better traces

Techniques to register behavior

Profilling

- Very useful for sequential programs
- Per application function summary
- However
  - Time-integrated data
  - No idea of when things happen

Techniques to register behavior

Profilling

- Very useful for sequential programs
- Per application function summary
- However
  - Time-integrated data
  - No idea of when things happen

Example

valgrind, gprof

Techniques to register behavior

Monitoring

- Coarse-grained monitoring data
- Usually system-centric, large-scale
- However
  - Sometimes space-integrated data
  - Hard to correlate with application

Techniques to register behavior

Monitoring

- Coarse-grained monitoring data
- Usually system-centric, large-scale
- However
  - Sometimes space-integrated data
  - Hard to correlate with application

Example

Ganglia, MonALISA, Nagios

Techniques to register behavior

Tracing

- Raw events $\rightarrow$ no aggregation
- Explicit event causality, correlation
- However
  - Hard to analyze (very detailed in time/space)
  - Which events to gather?

Techniques to register behavior

Tracing

- Raw events $\rightarrow$ no aggregation
- Explicit event causality, correlation
- However
  - Hard to analyze (very detailed in time/space)
  - Which events to gather?

Example

Akypuera – `https://github.com/schnorr/akypuera`, SimGrid

Registering behavior: summary

- Choice of technique dictates analysis possibilities
  $\rightarrow$ the analysis depends on available behavioral data

Registering behavior: summary

- Choice of technique dictates analysis possibilities
  $\rightarrow$ the analysis depends on available behavioral data

Example: resource utilization of a distributed application

- Tracing of one variable for each resource
  - Computing power in flops
  - Bandwidth consumption in bytes

Registering behavior: summary

- Choice of technique dictates analysis possibilities
  $\rightarrow$ the analysis depends on available behavioral data

Example: resource utilization of a distributed application

- Tracing of one variable for each resource
  - Computing power in flops
  - Bandwidth consumption in bytes
- Analysis
  - Bottleneck is on a specific network link
  - Not efficient in using full computing power

Registering behavior: summary

- Choice of technique dictates analysis possibilities
  $\rightarrow$ the analysis depends on available behavioral data

Example: resource utilization of a distributed application

- Tracing of one variable for each resource
  - Computing power in flops
  - Bandwidth consumption in bytes
- Analysis
  - Bottleneck is on a specific network link
  - Not efficient in using full computing power
- Unanswered questions
  - Which process is causing the bottleneck?
  - Is there a less efficient application phase?

Registering behavior: summary

- Choice of technique dictates analysis possibilities
  $\rightarrow$ the analysis depends on available behavioral data

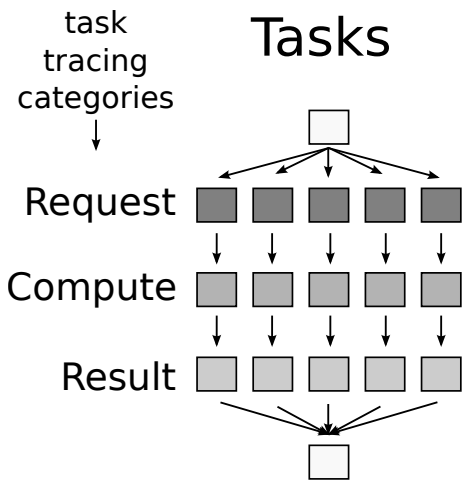
Example: resource utilization of a distributed application

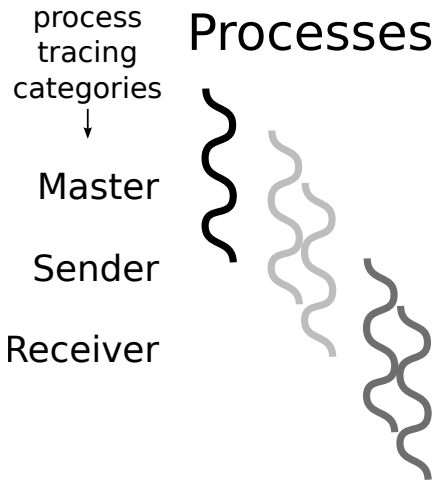
- Tracing of one variable for each resource
  - Computing power in flops
  - Bandwidth consumption in bytes
- Analysis
  - Bottleneck is on a specific network link
  - Not efficient in using full computing power
- Unanswered questions
  - Which process is causing the bottleneck?
  - Is there a less efficient application phase?

Maybe with categories?

Tracing resource utilization using categories

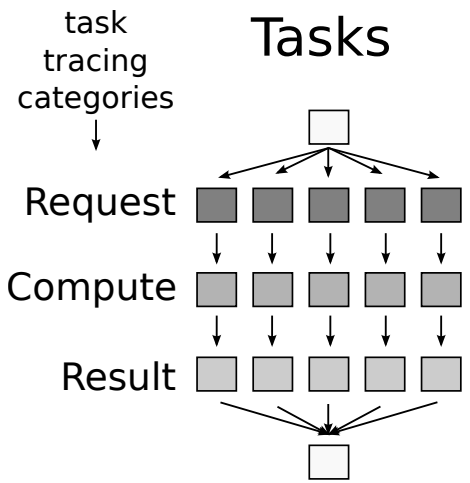
1. Define a set of application categories

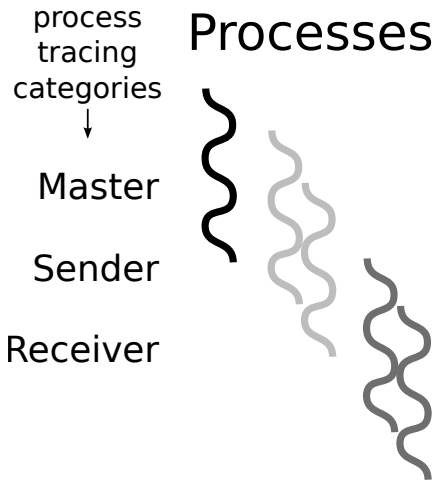




Tracing resource utilization using categories

1. Define a set of application categories

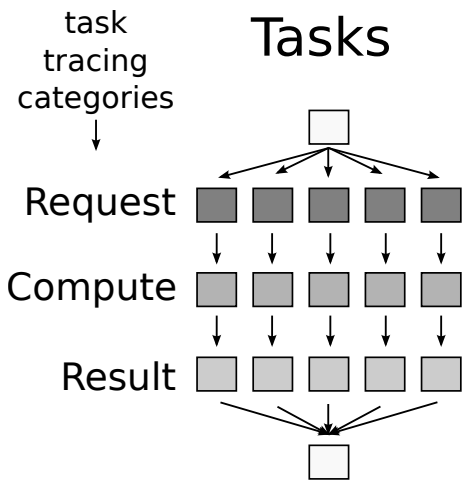


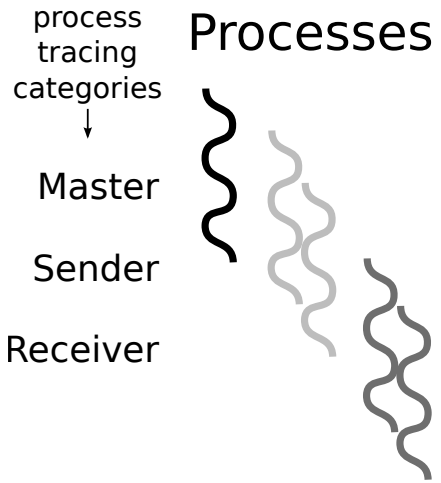


2. Trace resource utilization according to them

Tracing resource utilization using categories

1. Define a set of application categories





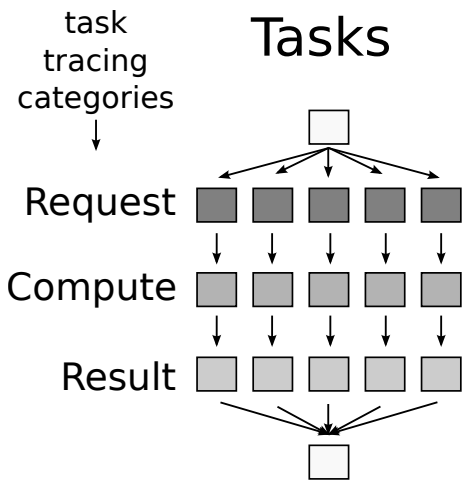
2. Trace resource utilization according to them

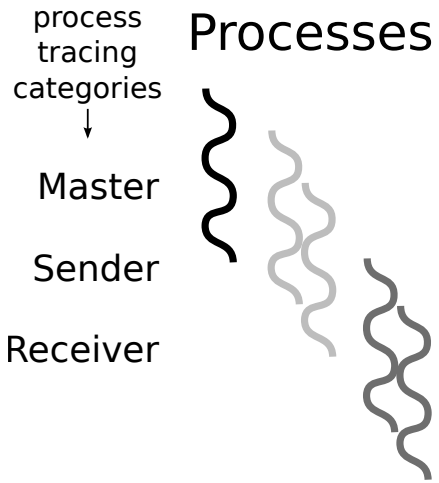
More analysis possibilities

- Temporal/Spatial correlation among categories
- Combine categories to test assumptions

Tracing resource utilization using categories

1. Define a set of application categories





2. Trace resource utilization according to them

More analysis possibilities

- Temporal/Spatial correlation among categories
- Combine categories to test assumptions

Very easy to trace this in a simulation

Exploratory trace visualization

Challenges

- Very large applications
  $\rightarrow$ Top500 has machines with 1.5 million cores
  $\rightarrow$ Easy to simulate millions of processes
- Low or Zero-intrusion tracing
  $\rightarrow$ Buffering, hardware support, simulation traces

Challenges

- Very large applications
  → Top500 has machines with 1.5 million cores
  → Easy to simulate millions of processes
- Low or Zero-intrusion tracing
  → Buffering, hardware support, simulation traces

Space/Time trace size explosion

- Very detailed in time, many entities in space
- Data representation without care
  → may deceive understanding

Challenges

- Very large applications
  → Top500 has machines with 1.5 million cores
  → Easy to simulate millions of processes
- Low or Zero-intrusion tracing
  → Buffering, hardware support, simulation traces

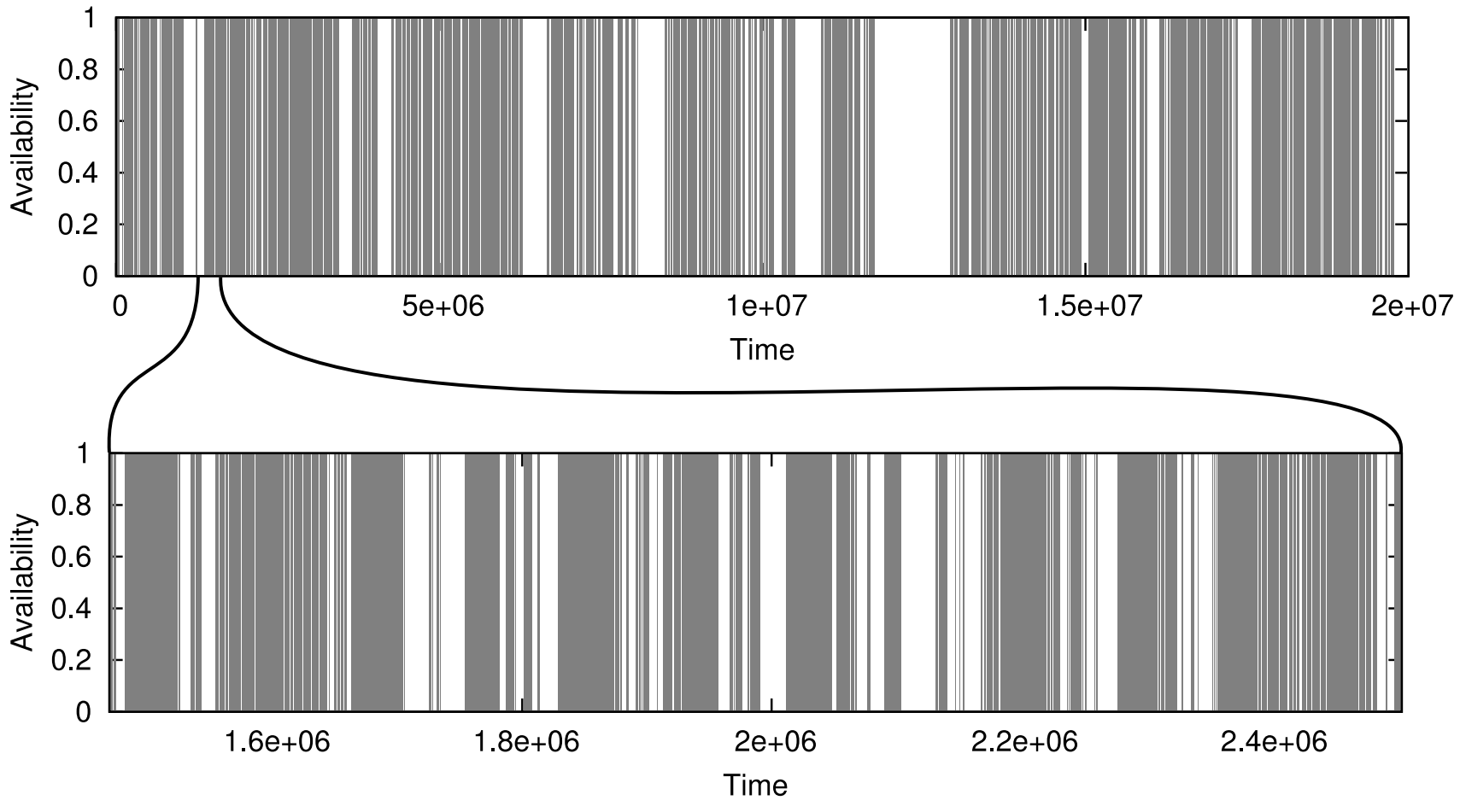
Space/Time trace size explosion

- Very detailed in time, many entities in space
- Data representation without care
  → may deceive understanding

- Real BOINC availability trace file
  - One volunteer, availability is either true or false
  - 8-month period, then 12-day zoom
- Plot with GNUPlot to a PDF (vector) file

Motivation (BOINC example)

- One volunteer client (top: 8-month, bottom: 12-day)
- Reasonable view, with a zoom for details



Motivation – trust the rendering?

Same vector file, two different views

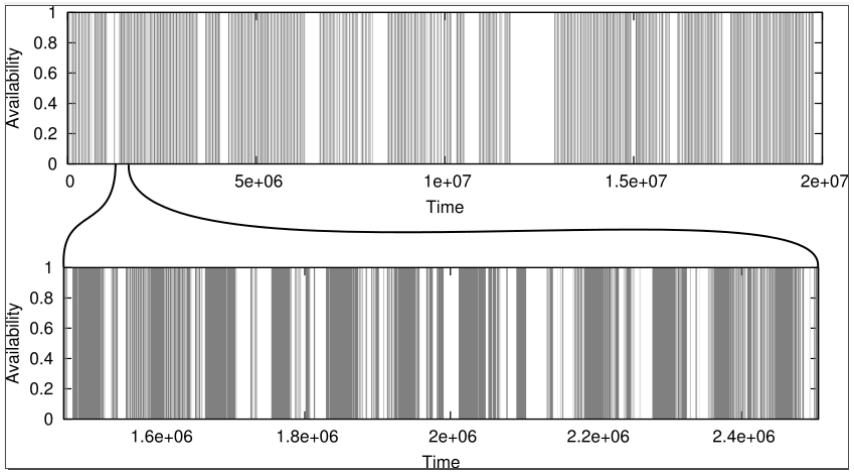
$\rightarrow$ Different interpretation depending on the viewer

Motivation – trust the rendering?

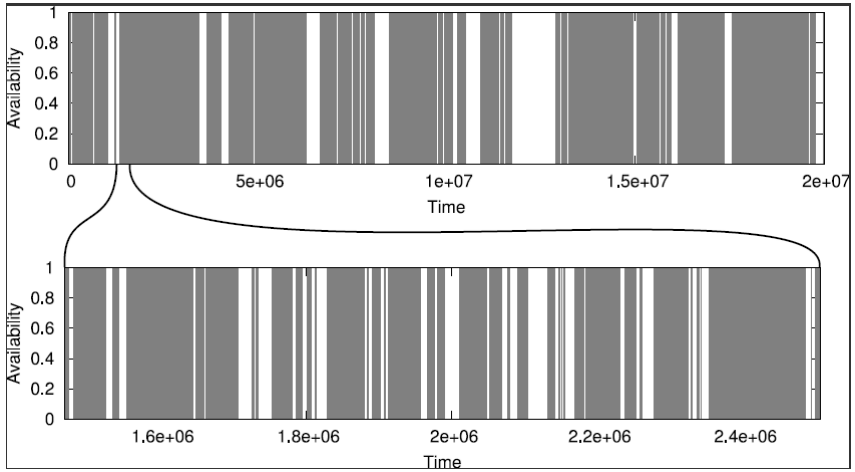
Same vector file, two different views

→ Different interpretation depending on the viewer

Evince



Acroread

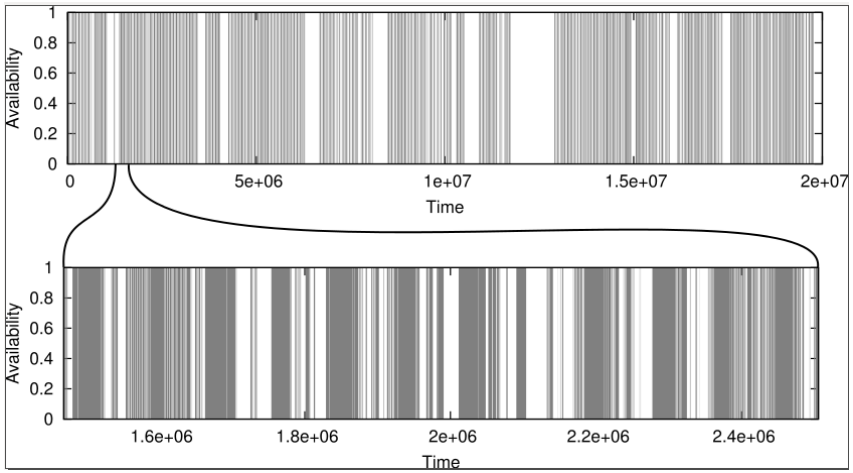


Motivation – trust the rendering?

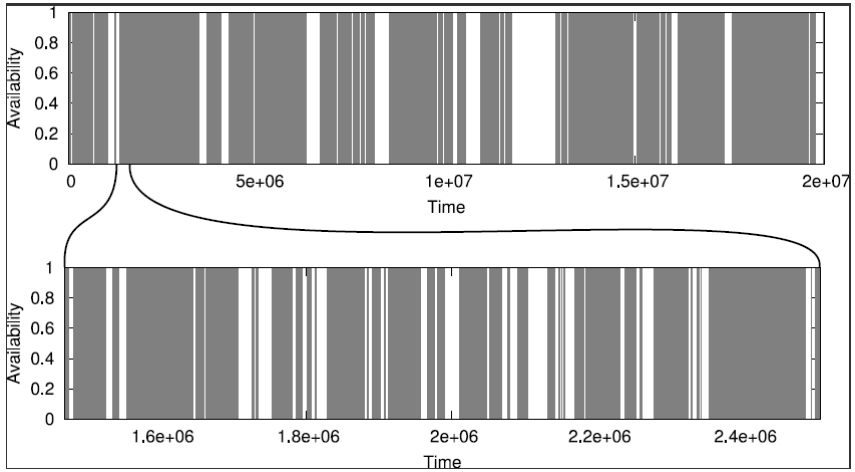
Same vector file, two different views

→ Different interpretation depending on the viewer

Evince



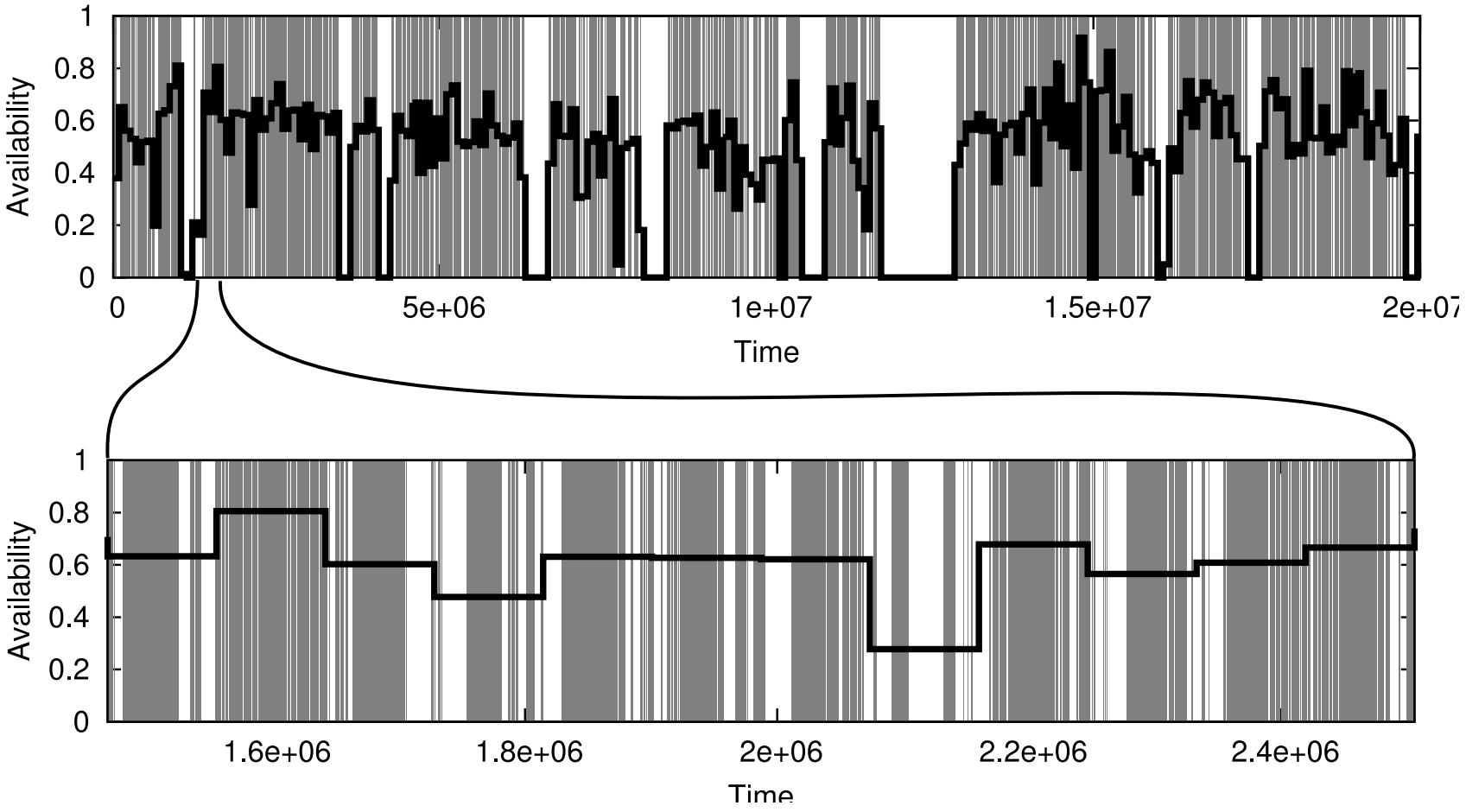
Acroread



- Should we trust the rendering ?
  - No!
  - We need to make choices before visualizing data

Motivation → data aggregation

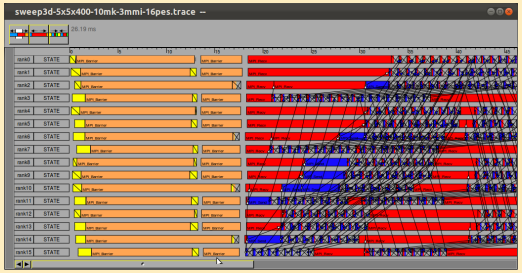
- 24-hour time integration



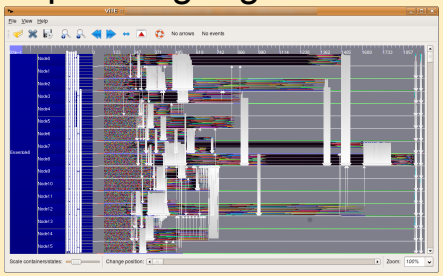
Trace analysis through visualization

Space/Time views

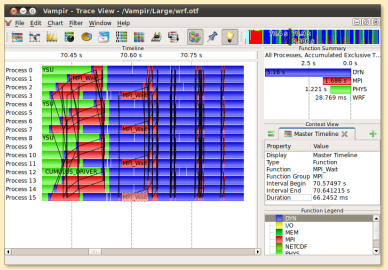
Paje
http://paje.sf.net



Vite
http://vite.gforge.inria.fr



Vampir
http://vampir.eu

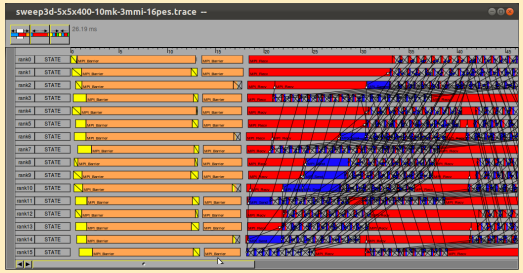


- Also impacted by ever larger trace sizes
- Limited visualization scalability

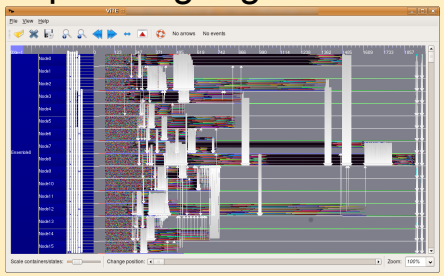
Trace analysis through visualization

Space/Time views

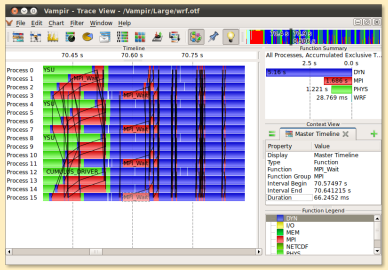
Paje
http://paje.sf.net



Vite
http://vite.gforge.inria.fr

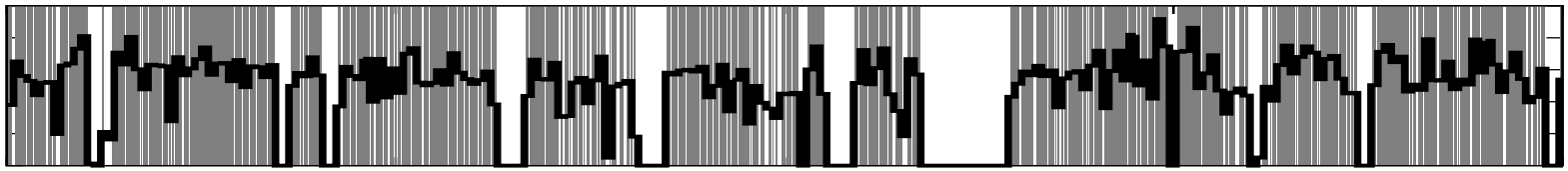


Vampir
http://vampir.eu



- Also impacted by ever larger trace sizes
- Limited visualization scalability

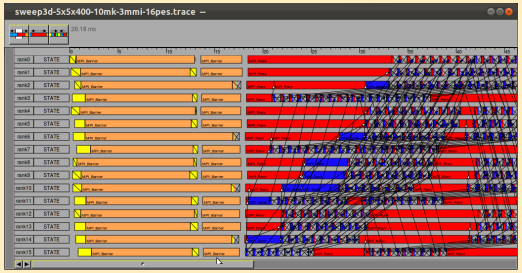
- Data aggregation is key for large-scale visualization
  $\rightarrow$ Avoid graphical aggregation rendering
- Aggregated data may be more representative



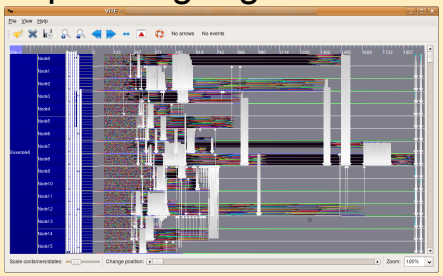
Trace analysis through visualization

Space/Time views

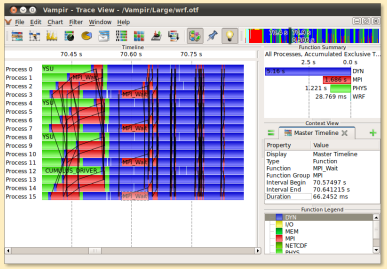
Paje
http://paje.sf.net



Vite
http://vite.gforge.inria.fr



Vampir
http://vampir.eu



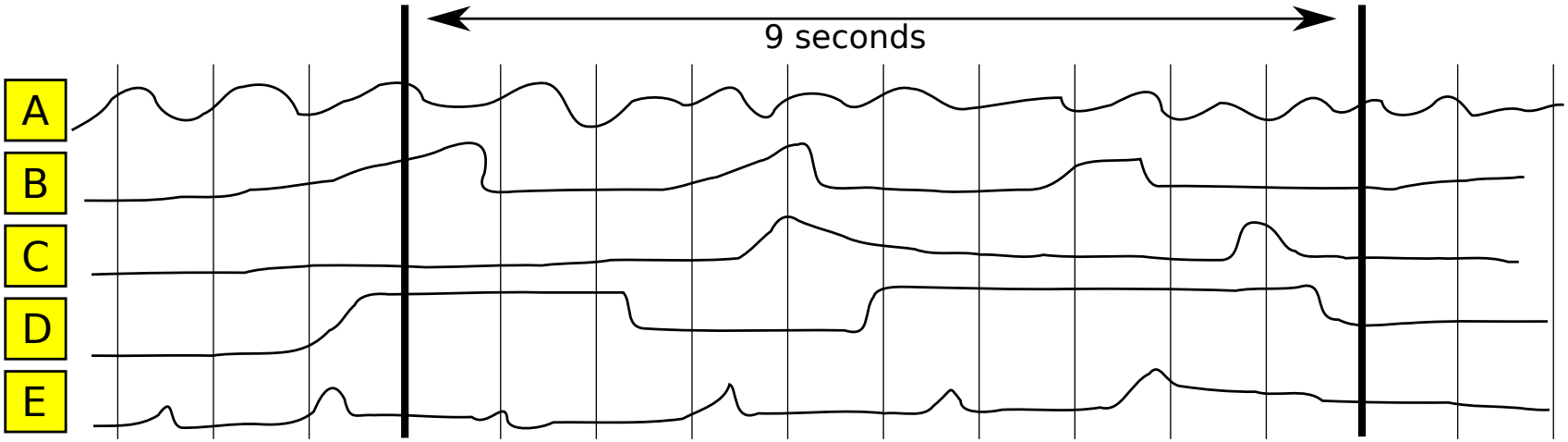
- Also impacted by ever larger trace sizes
- Limited visualization scalability

- Data aggregation is key for large-scale visualization
  $\rightarrow$ Avoid graphical aggregation rendering
- Aggregated data may be more representative
- Note: Concerns with behavior attenuation
  - Aggregation may remove important details
  - Flexible aggregation: operators & neighborhood

Space/Time Trace Aggregation

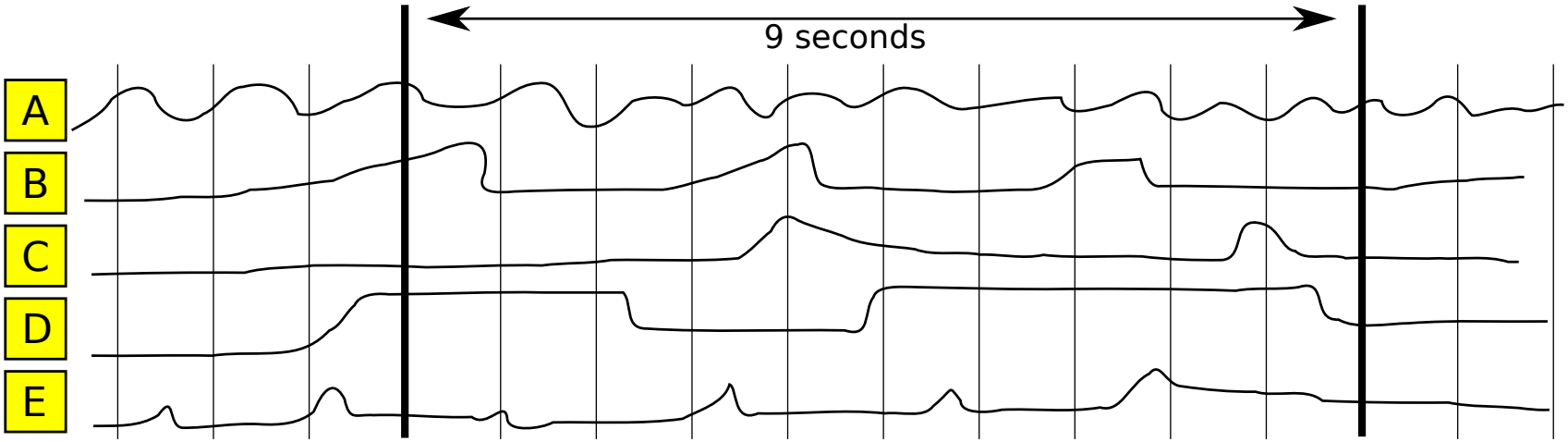
Space/Time Trace Aggregation

- Temporal integration
  1. Time interval defined during the analysis
  2. Summary of events for each monitored entity



Space/Time Trace Aggregation

- Temporal integration
  1. Time interval defined during the analysis
  2. Summary of events for each monitored entity



Time-integrated summary for processes

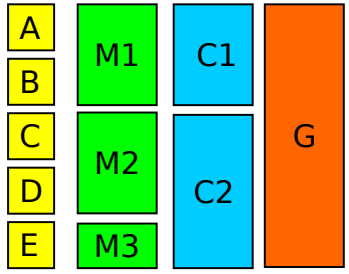
A=(44) B=(76) C=(36) D=(90) E=(45)

Space/Time Trace Aggregation

- Spatial integration
  1. Define a neighborhood for each monitored entity
  2. Apply an aggregating operator on the neighborhood

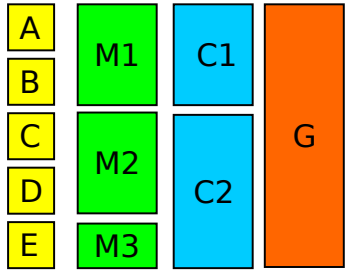
Space/Time Trace Aggregation

- Spatial integration
  1. Define a neighborhood for each monitored entity
  2. Apply an aggregating operator on the neighborhood
- Neighborhood as a hierarchy
  - Resource-based
  - Application groups



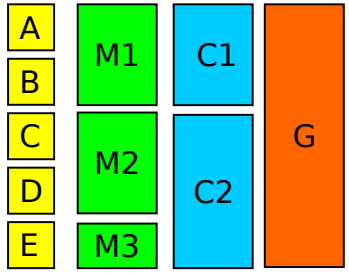
Space/Time Trace Aggregation

- Spatial integration
  1. Define a neighborhood for each monitored entity
  2. Apply an aggregating operator on the neighborhood
- Neighborhood as a hierarchy
  - Resource-based
  - Application groups
- Deeper the hierarchy $\rightarrow$ higher the quality



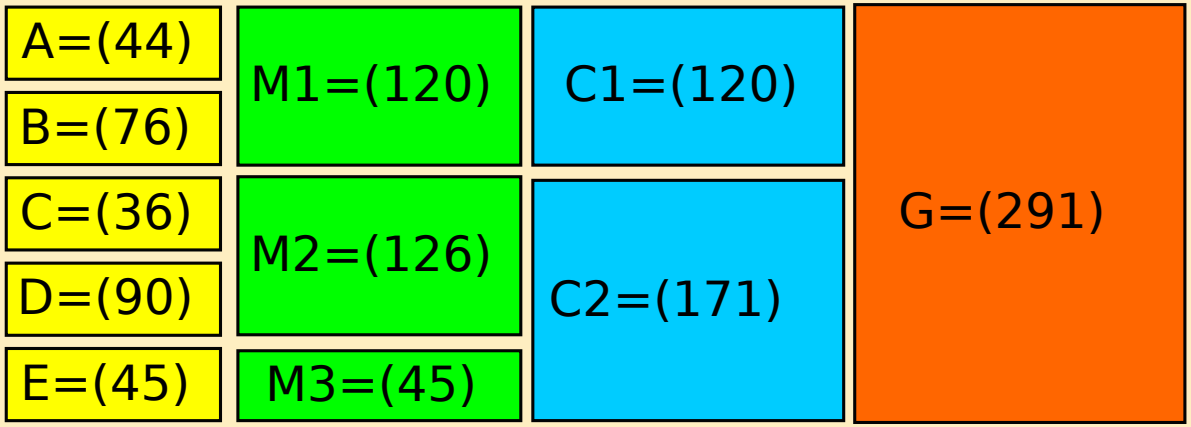
Space/Time Trace Aggregation

- Spatial integration
  1. Define a neighborhood for each monitored entity
  2. Apply an aggregating operator on the neighborhood
- Neighborhood as a hierarchy
  - Resource-based
  - Application groups
- Deeper the hierarchy $\rightarrow$ higher the quality



Space-integrated summary

Aggregating operator: addition

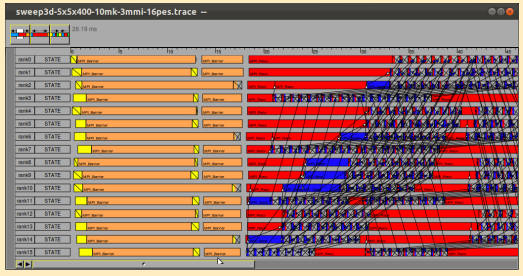


Hierarchical Graph View

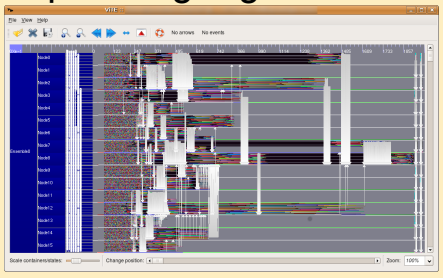
Hierarchical Graph View

Reminder: Space/Time views

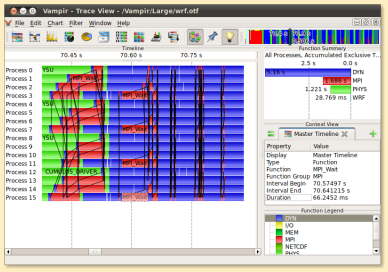
Paje
http://paje.sf.net



Vite
http://vite.gforge.inria.fr



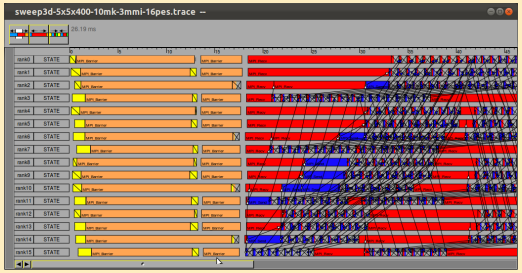
Vampir
http://vampir.eu



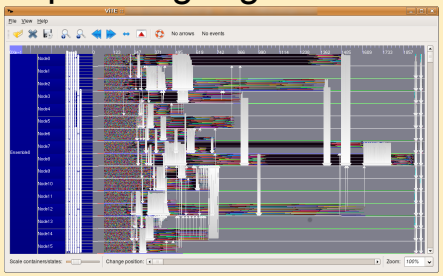
Hierarchical Graph View

Reminder: Space/Time views

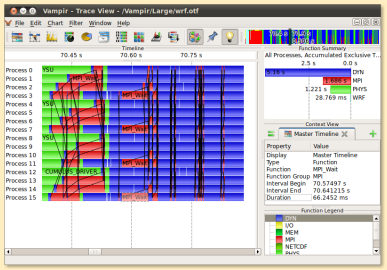
Paje
http://paje.sf.net



Vite
http://vite.gforge.inria.fr



Vampir
http://vampir.eu

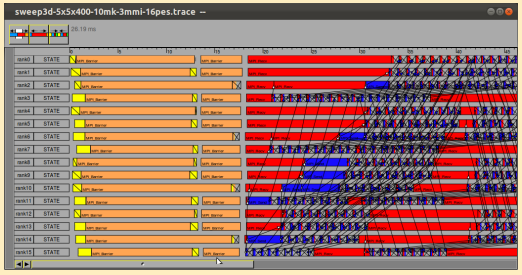


Problem $\rightarrow$ Lack of topological information

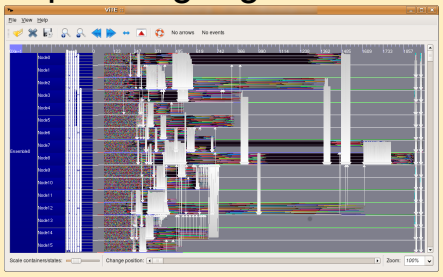
Hierarchical Graph View

Reminder: Space/Time views

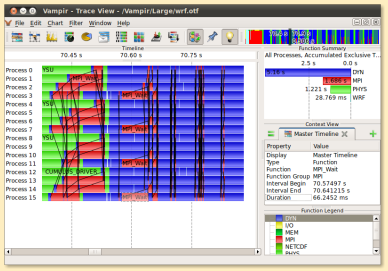
Paje
http://paje.sf.net



Vite
http://vite.gforge.inria.fr



Vampir
http://vampir.eu



Problem $\rightarrow$ Lack of topological information

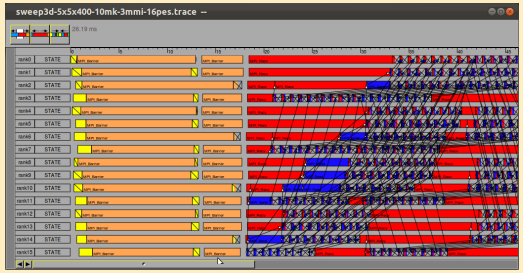
What we want?

- Pin-point resource contention
- Verify efficiency
- Explain anomalies

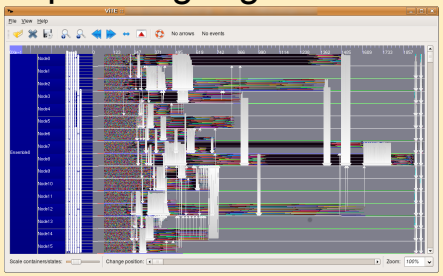
Hierarchical Graph View

Reminder: Space/Time views

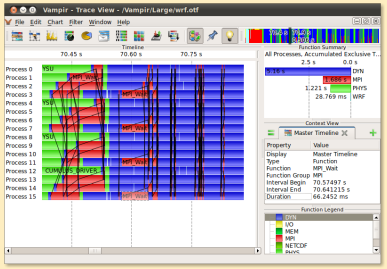
Paje
http://paje.sf.net



Vite
http://vite.gforge.inria.fr



Vampir
http://vampir.eu



Problem $\rightarrow$ Lack of topological information

What we want?

- Pin-point resource contention
- Verify efficiency
- Explain anomalies
- Correlate: application behavior $\rightarrow$ network topology

Hierarchical Graph View

- Graph visualization

Hierarchical Graph View

- Graph visualization

Scalable graph representation

- Topology, with application-level metrics
- Identify resource bottleneck in space and time

Hierarchical Graph View

- Graph visualization

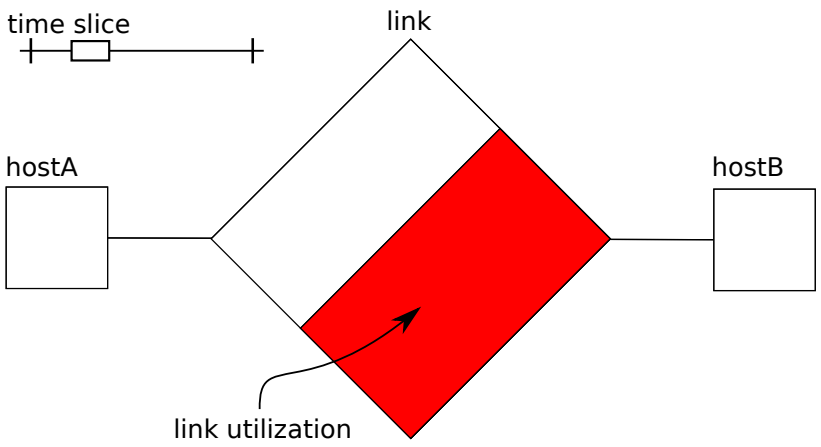
Scalable graph representation

- Topology, with application-level metrics
- Identify resource bottleneck in space and time

- Use spatial-temporal aggregated traces
- Interactive force-directed layout (Barnes-Hut algorithm)

Hierarchical Graph View - how it works

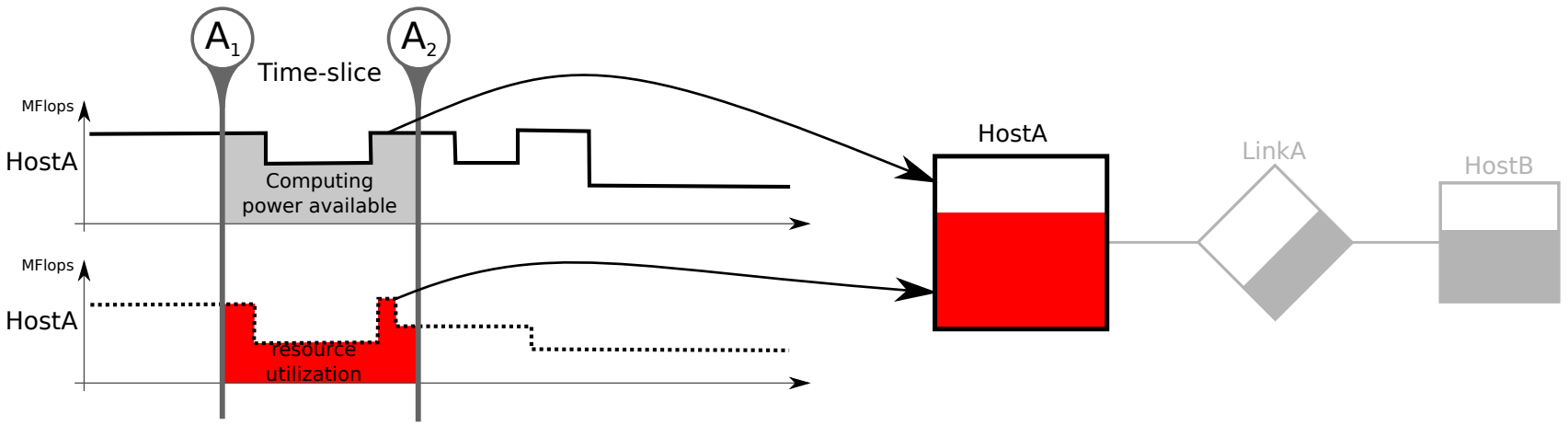
- Map trace metrics to geometrical attributes
  - → Size, shape, filling, colors
  - → Nodes: monitored entities
  - → Edges: relationship among entities



Hosts → squares
Links → diamonds

Hierarchical Graph View - Aggregated Data

- Considering temporal aggregation only



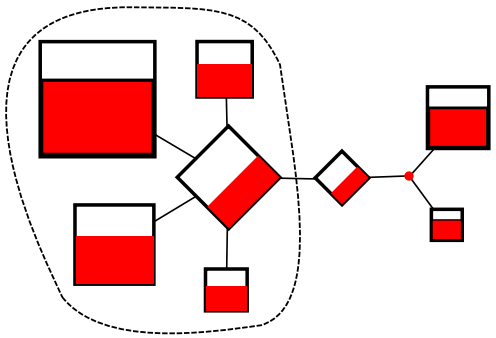
- Time-slice changes $\rightarrow$ new layout is rendered

Hierarchical Graph View - Aggregated Data

- Considering spatial-temporal aggregation
  - Explore trace hierarchy
    $\rightarrow$ cores, processors, machines, clusters, datacenters

Hierarchical Graph View - Aggregated Data

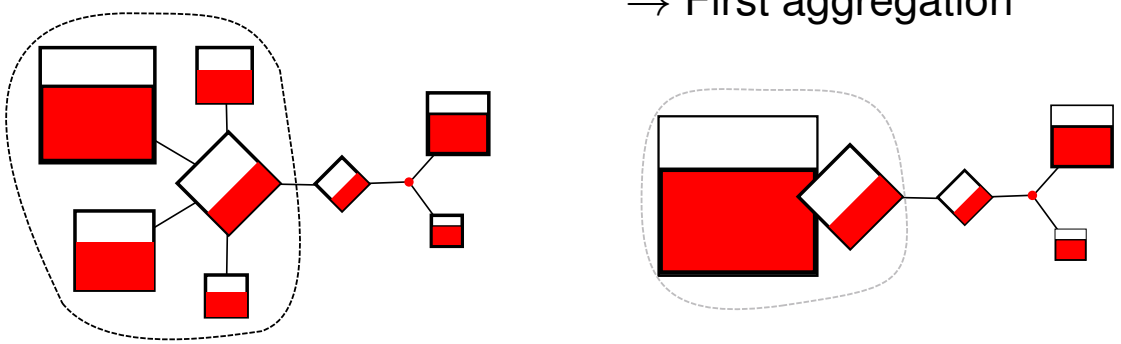
- Considering spatial-temporal aggregation
  - Explore trace hierarchy
    $\rightarrow$ cores, processors, machines, clusters, datacenters
- Aggregated representation and its steps



Hierarchical Graph View - Aggregated Data

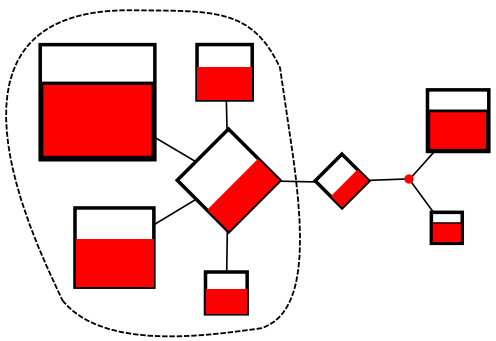
- Considering spatial-temporal aggregation
  - Explore trace hierarchy
    $\rightarrow$ cores, processors, machines, clusters, datacenters
- Aggregated representation and its steps

$\rightarrow$ First aggregation

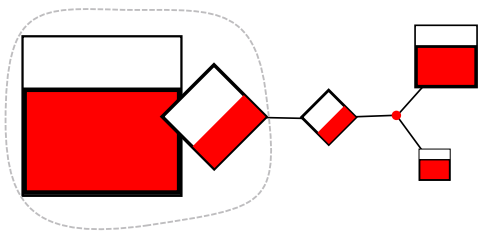


Hierarchical Graph View - Aggregated Data

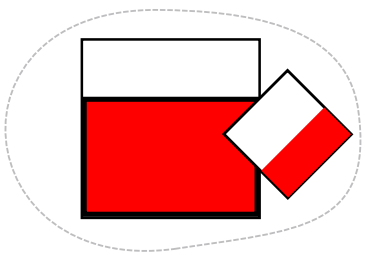
- Considering spatial-temporal aggregation
  - Explore trace hierarchy
    $\rightarrow$ cores, processors, machines, clusters, datacenters
- Aggregated representation and its steps



$\rightarrow$ First aggregation

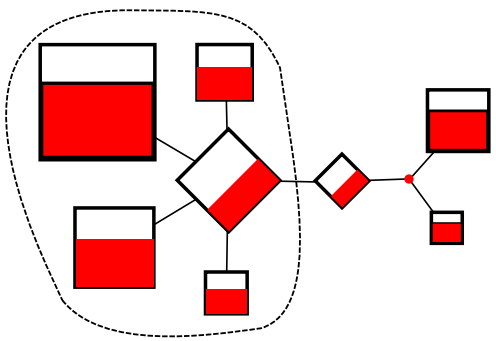


$\rightarrow$ Second aggregation

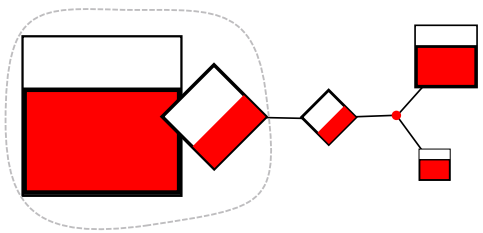


Hierarchical Graph View - Aggregated Data

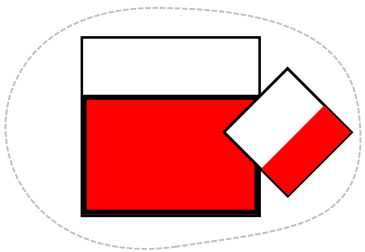
- Considering spatial-temporal aggregation
  - Explore trace hierarchy
    $\rightarrow$ cores, processors, machines, clusters, datacenters
- Aggregated representation and its steps



$\rightarrow$ First aggregation



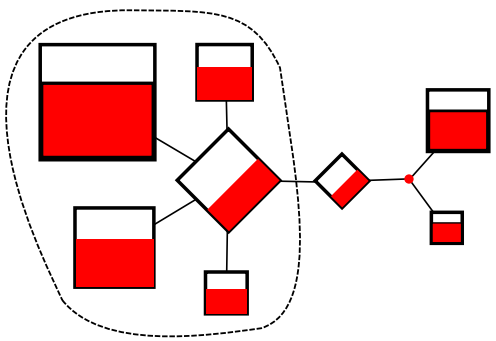
$\rightarrow$ Second aggregation



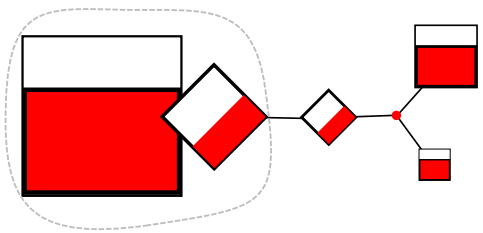
- Analyst decides
  - the cut on the hierarchy (defining a new graph)

Hierarchical Graph View - Aggregated Data

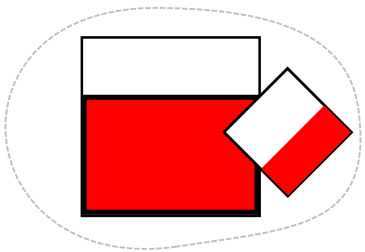
- Considering spatial-temporal aggregation
  - Explore trace hierarchy
    $\rightarrow$ cores, processors, machines, clusters, datacenters
- Aggregated representation and its steps



$\rightarrow$ First aggregation



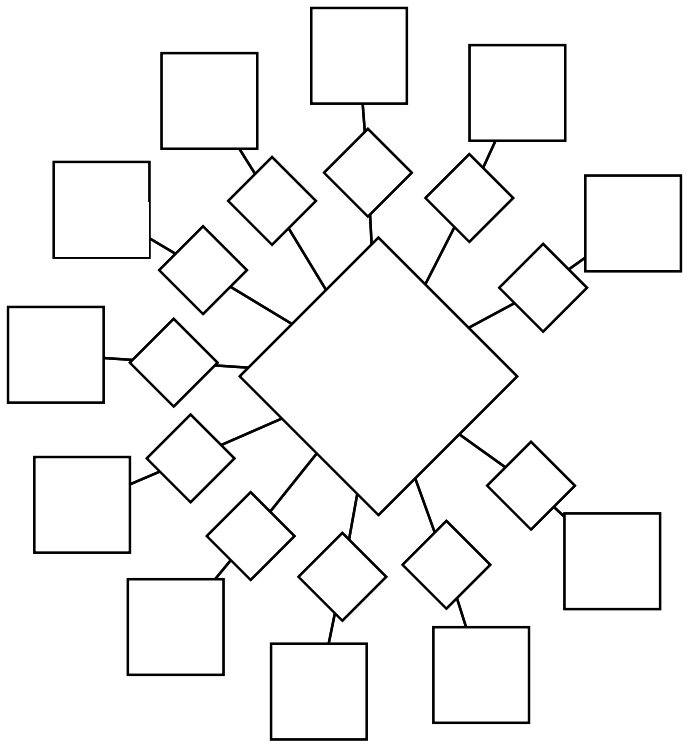
$\rightarrow$ Second aggregation



- Analyst decides
  - the cut on the hierarchy (defining a new graph)
- Graph changes $\rightarrow$ force-directed updates positions

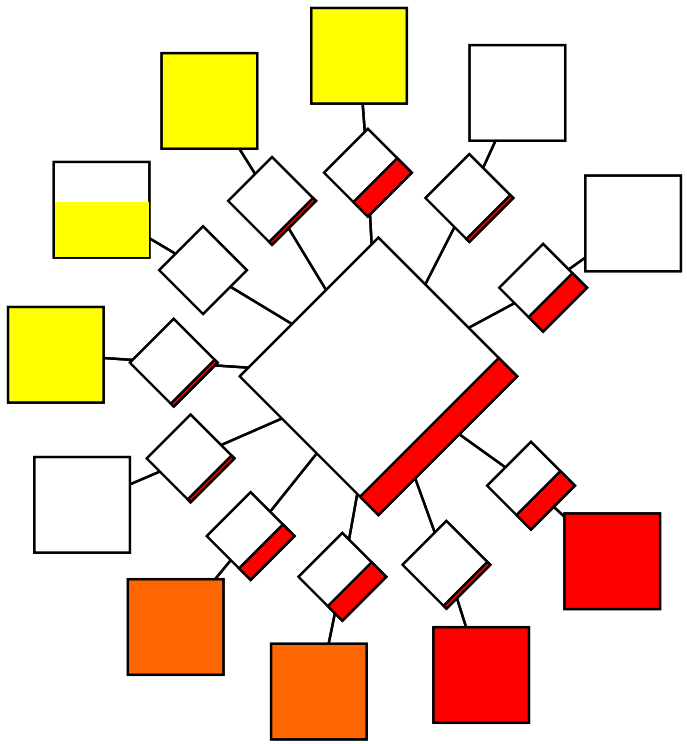
Hierarchical Graph View - an example

- Squares are hosts, diamonds are network links
  - Cluster backbone $\rightarrow$ larger bandwidth capacity
  - Hosts connected to the backbone by private links



Hierarchical Graph View - an example

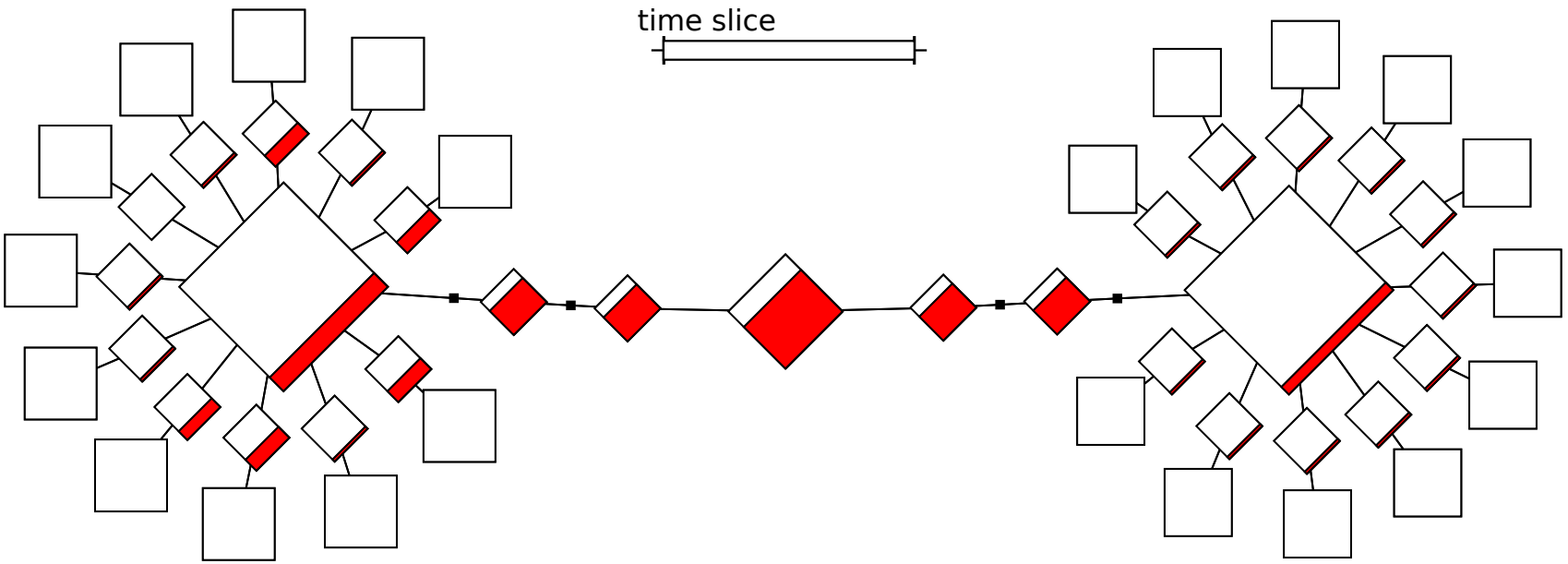
- Squares are hosts, diamonds are network links
  - Cluster backbone $\rightarrow$ larger bandwidth capacity
  - Hosts connected to the backbone by private links
- Colors represent different applications
- or parts of it (task type, phase)



Scenario and Demonstration

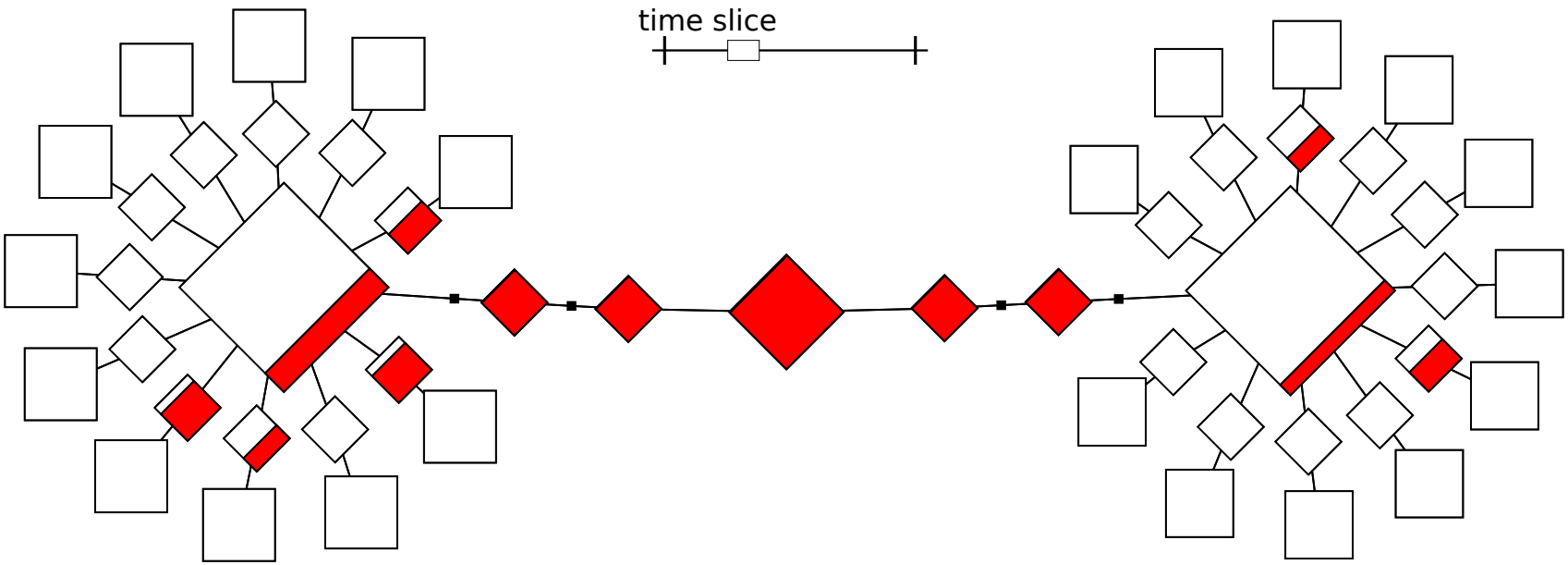
NAS-DT Class A WH

- NAS DT Class A White Hole algorithm
  $\rightarrow$ Traces from SMPI (Simulated MPI, part of SimGrid)
- Network topology – resource utilization by red filling
- Only temporal aggregation



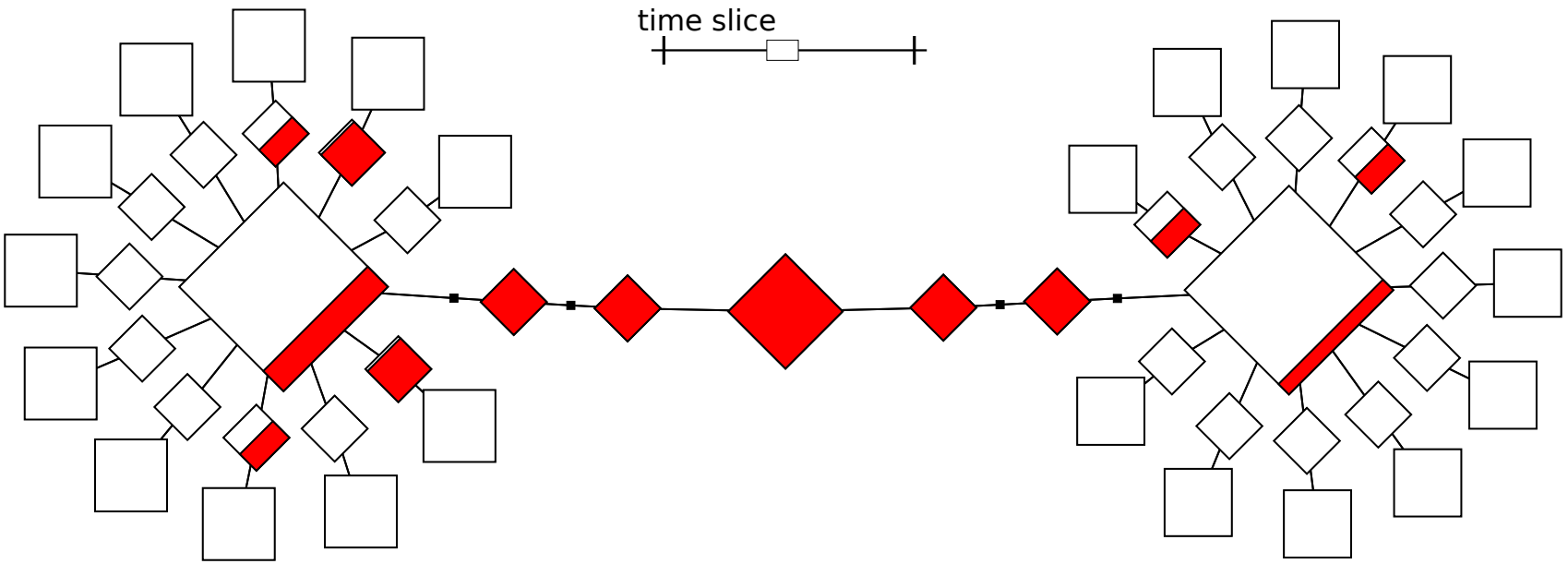
NAS-DT Class A WH

- NAS DT Class A White Hole algorithm
  $\rightarrow$ Traces from SMPI (Simulated MPI, part of SimGrid)
- Network topology – resource utilization by red filling
- Only temporal aggregation



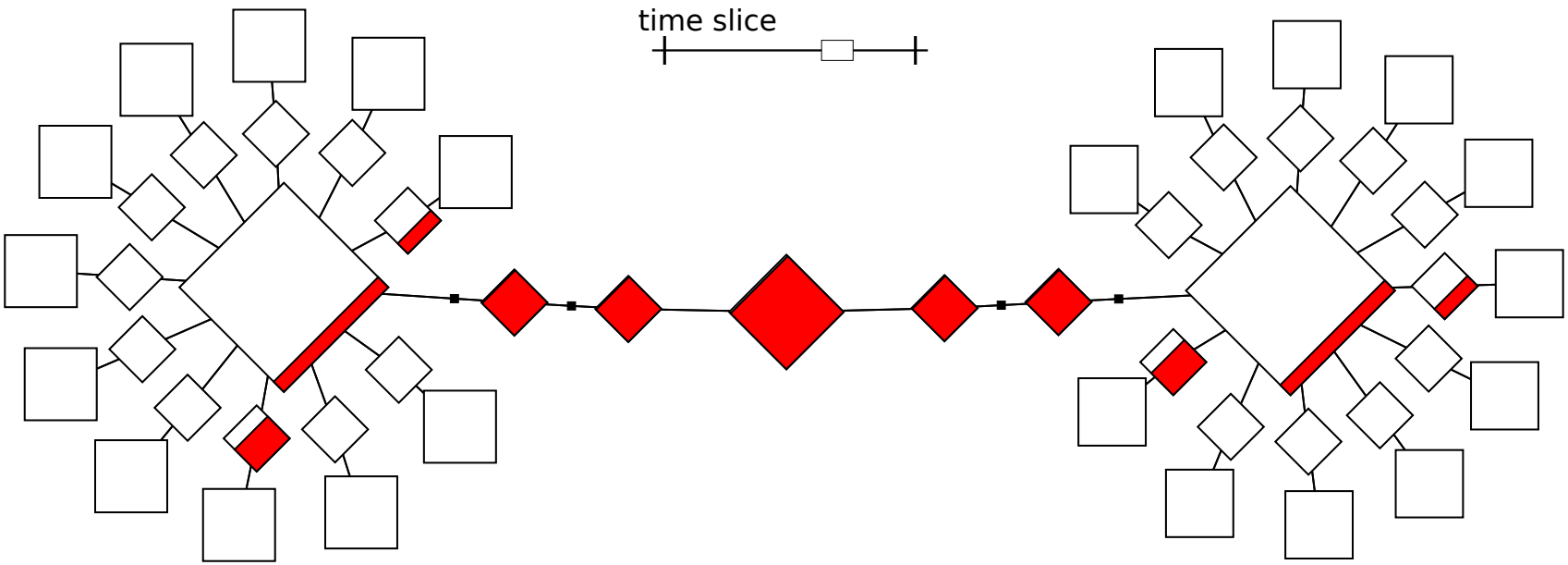
NAS-DT Class A WH

- NAS DT Class A White Hole algorithm
  $\rightarrow$ Traces from SMPI (Simulated MPI, part of SimGrid)
- Network topology – resource utilization by red filling
- Only temporal aggregation



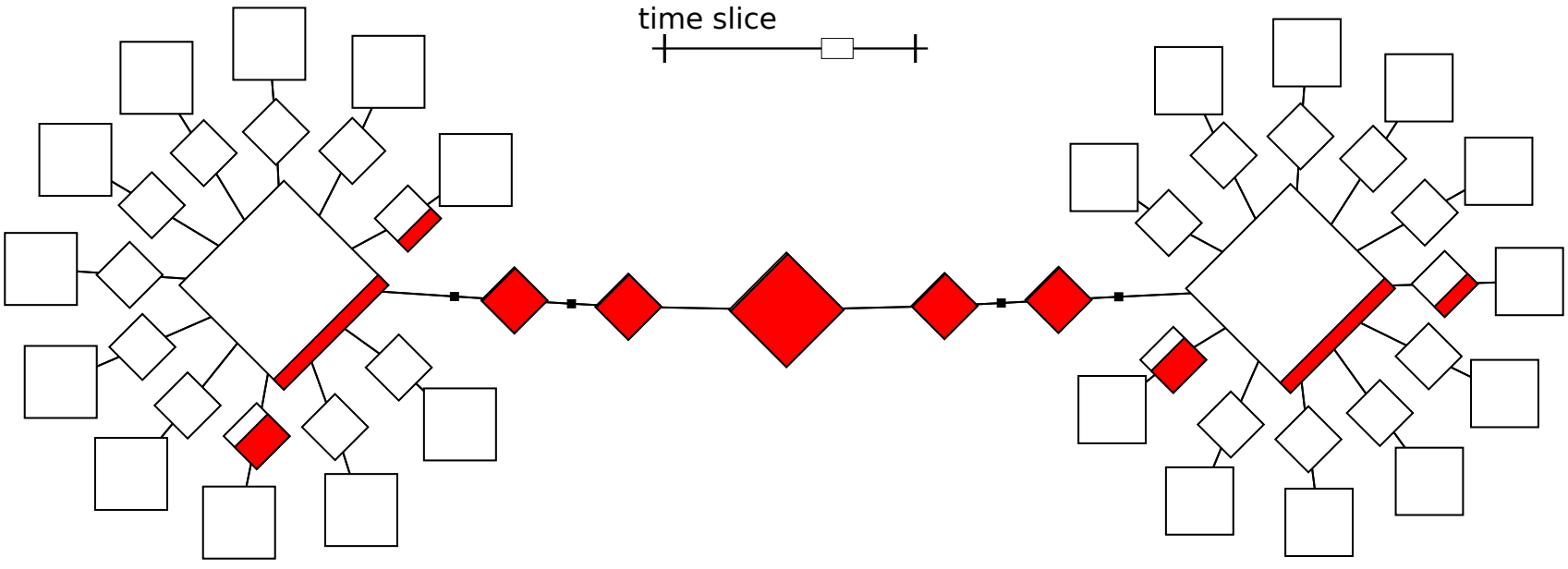
NAS-DT Class A WH

- NAS DT Class A White Hole algorithm
  $\rightarrow$ Traces from SMPI (Simulated MPI, part of SimGrid)
- Network topology – resource utilization by red filling
- Only temporal aggregation



NAS-DT Class A WH

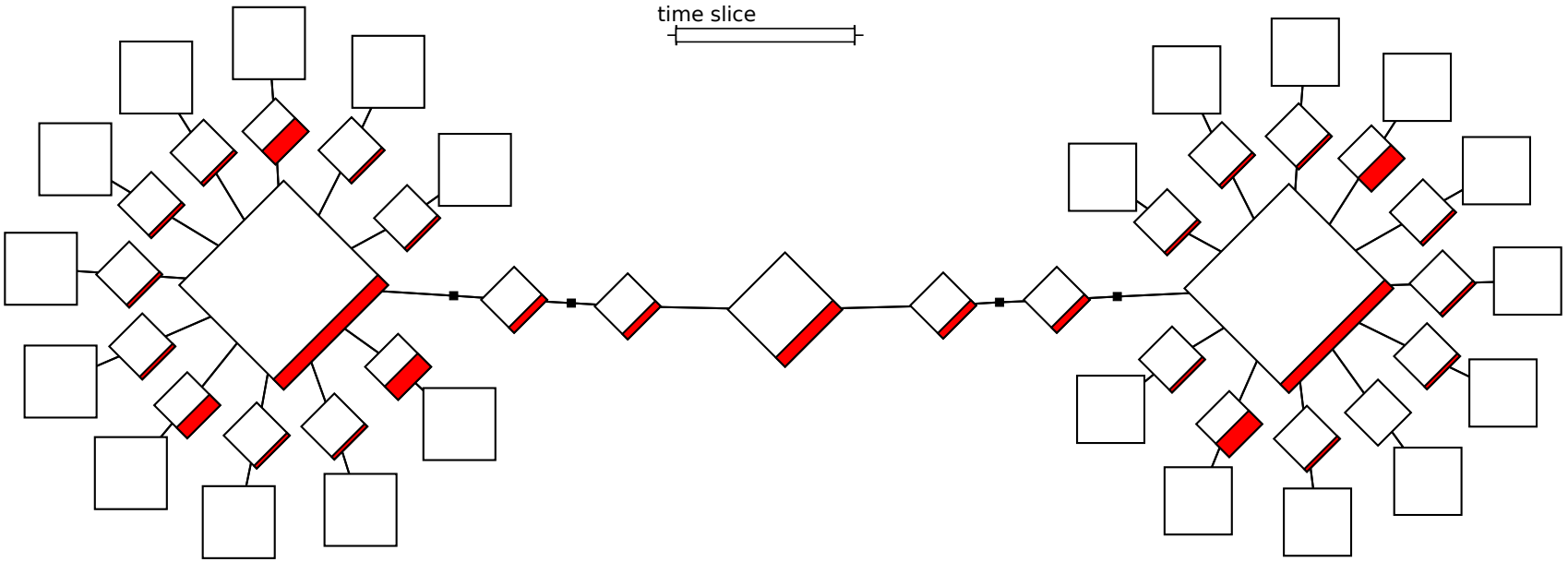
- NAS DT Class A White Hole algorithm
  $\rightarrow$ Traces from SMPI (Simulated MPI, part of SimGrid)
- Network topology – resource utilization by red filling
- Only temporal aggregation



- Analysis: interconnection backbone is the bottleneck

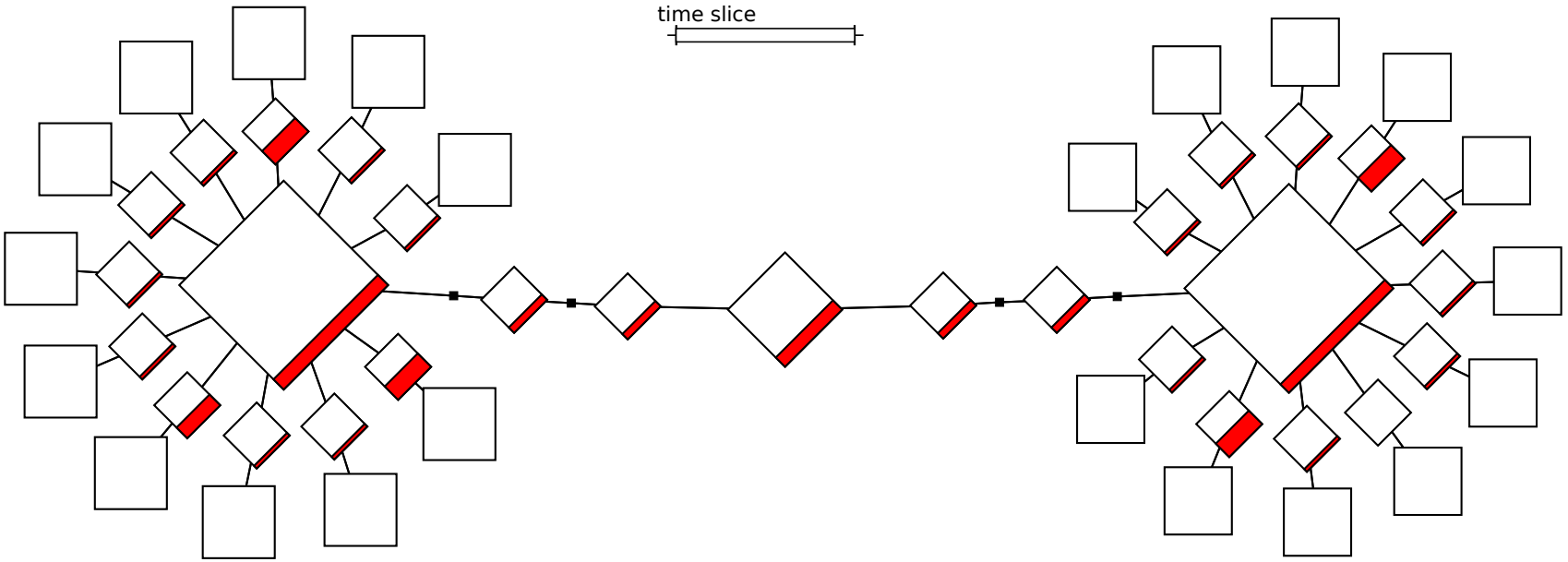
NAS-DT Class A WH (second try)

- Another deployment with a different mapping
  $\rightarrow$ by changing the order of machines in hostfile
- Explore communication locality



NAS-DT Class A WH (second try)

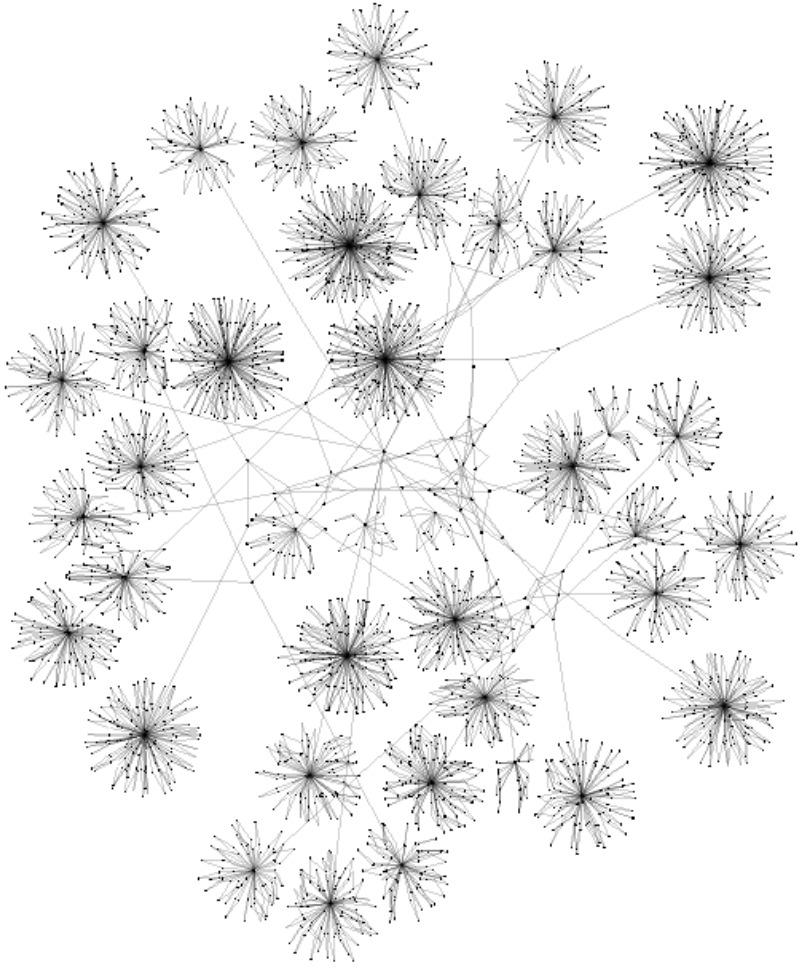
- Another deployment with a different mapping
  $\rightarrow$ by changing the order of machines in hostfile
- Explore communication locality



- Note: Small scale and easy scenario – but it is a start

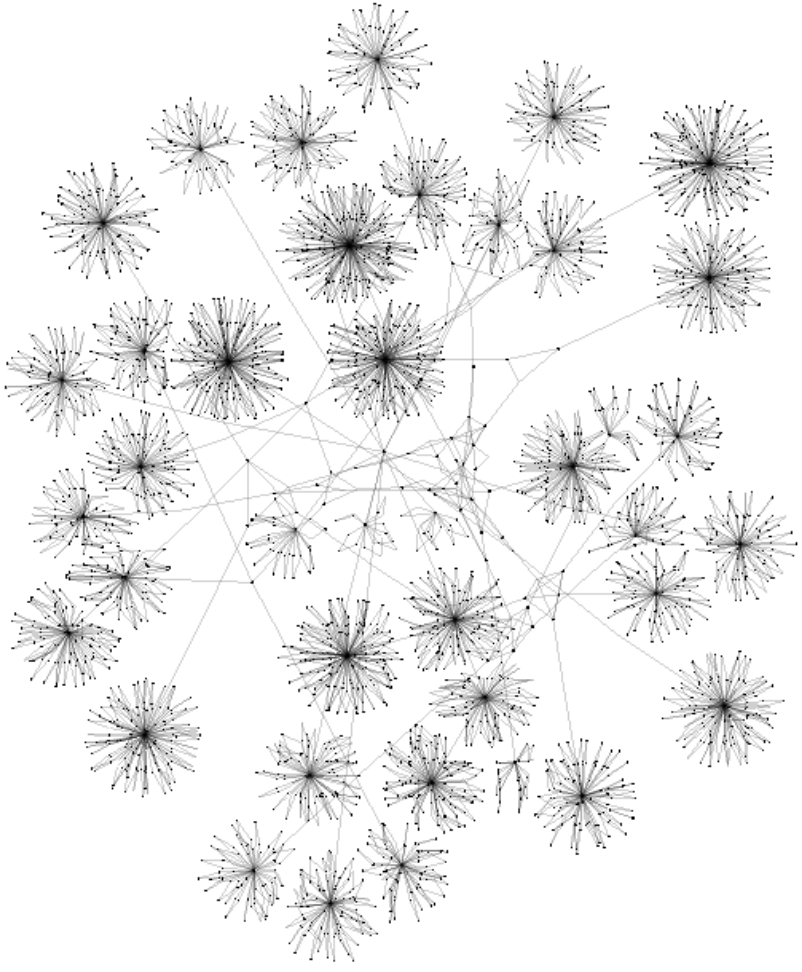
Live demonstration

- Using viva with Grid'5000 network topology
  $\rightarrow$ a graph with 4798 nodes
- Only resource capacity (power and bandwidth)



Live demonstration

- Using viva with Grid'5000 network topology
  $\rightarrow$ a graph with 4798 nodes
- Only resource capacity (power and bandwidth)



Really feasible to analyze them all, with all the details?

Conclusion

- Categorized resource tracing
  $\rightarrow$ Gives richer analysis scenarios
  $\rightarrow$ Should be used in real life as well

Conclusion

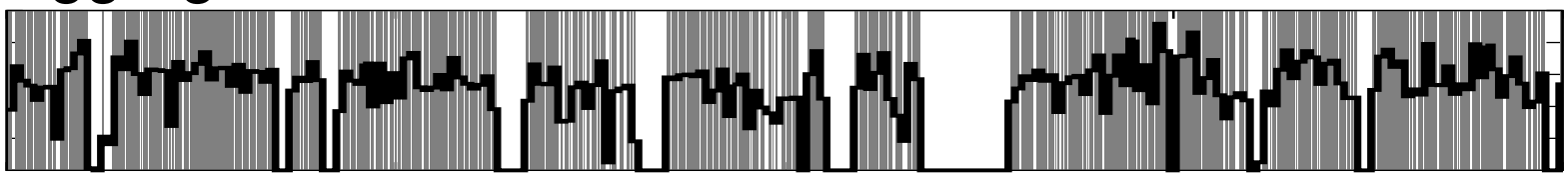
- Categorized resource tracing
  - $\rightarrow$ Gives richer analysis scenarios
  - $\rightarrow$ Should be used in real life as well
- Data aggregation
  - Key to scale data visualization for analysis
  - No pre-defined or fixed parameters
  - Fully configurable by the analyst
    - Time and space-slice, operators

Conclusion

- Categorized resource tracing
  $\rightarrow$ Gives richer analysis scenarios
  $\rightarrow$ Should be used in real life as well
- Data aggregation
  - Key to scale data visualization for analysis
  - No pre-defined or fixed parameters
  - Fully configurable by the analyst
    - Time and space-slice, operators
- Hierarchical graph view
  - Based upon aggregated data
  - Complementary to existing techniques
    - With larger data-sets, does it remain useful?
      $\rightarrow$ Is it relevant to see the behavior of everyone?

Conclusion

- Categorized resource tracing
  $\rightarrow$ Gives richer analysis scenarios
  $\rightarrow$ Should be used in real life as well
- Data aggregation
  - Key to scale data visualization for analysis
  - No pre-defined or fixed parameters
  - Fully configurable by the analyst
    - Time and space-slice, operators
- Hierarchical graph view
  - Based upon aggregated data
  - Complementary to existing techniques
    - With larger data-sets, does it remain useful?
      $\rightarrow$ Is it relevant to see the behavior of everyone?
- Aggregation $\rightarrow$ behavior attenuation



Open-source tools

- Paje (Space/Time views, pie-charts), LGPL

  `http://paje.sourceforge.net`

  - Since 2000, GNUstep-based, written in Objective-C
  - Not only a monolithic visualization tool
    - Component-based, graph of components
    - Framework for developing other tools
    - Paje Protocol
  - 30K SLOC, hard to maintain, hard to install GNUstep

Open-source tools

- Paje (Space/Time views, pie-charts), LGPL

  `http://paje.sourceforge.net`

  - Since 2000, GNUstep-based, written in Objective-C
  - Not only a monolithic visualization tool
    - Component-based, graph of components
    - Framework for developing other tools
    - Paje Protocol
  - 30K SLOC, hard to maintain, hard to install GNUstep

- Triva (Treemaps, Hierarchical graph), LGPL

  `http://triva.gforge.inria.fr`

  - Since 2007, GNUstep and Paje-based, also in Obj-C
    - Follows the Paje protocol
  - GNUstep runtime poses scalability problems

Open-source tools

- Paje (Space/Time views, pie-charts), LGPL
  `http://paje.sourceforge.net`
  - Since 2000, GNUstep-based, written in Objective-C
  - Not only a monolithic visualization tool
    - Component-based, graph of components
    - Framework for developing other tools
    - Paje Protocol
  - 30K SLOC, hard to maintain, hard to install GNUstep
- Triva (Treemaps, Hierarchical graph), LGPL
  `http://triva.gforge.inria.fr`
  - Since 2007, GNUstep and Paje-based, also in Obj-C
    - Follows the Paje protocol
  - GNUstep runtime poses scalability problems

Don't use these tools $\rightarrow$ see next slide

Future work

Technical

- Paje++ (or Paje2) – complete re-write in C++, Qt
- Viva – visualization tool (Treemap, Hierachical Graph)
  `https://github.com/schnorr/viva` (coming soon)
- For both, debian packaging

Future work

Technical

- Paje++ (or Paje2) – complete re-write in C++, Qt
- Viva – visualization tool (Treemap, Hierachical Graph)
  `https://github.com/schnorr/viva` (coming soon)
- For both, debian packaging

Research

- Evaluate data aggregation

Thank you for your attention

- Some references
  - Detection and Analysis of Resource Usage Anomalies in Large Distributed Systems Through Multi-scale Visualization. Lucas Mello Schnorr, Arnaud Legrand, Jean-Marc Vincent. Concurrency and Computation: Practice and Experience. Wiley. 2012.
  - Multi-scale Analysis of Large Distributed Computing Systems. Lucas Mello Schnorr, Arnaud Legrand, Jean-Marc Vincent. Third Workshop on Large-scale System and Application Performance (LSAP2011). The 20th International ACM Symposium on High-Performance Parallel and Distributed Computing (HPDC)
- More information

  $\rightarrow$ `http://mescal.imag.fr/membres/lucas.schnorr/`
- INFRA-SONGS Project (WP-7)

  Simulation of Next Generation Systems
  WP-7: Visualization and Analysis
  `http://infra-songs.gforge.inria.fr/`
- SimGrid toolkit

  `http://simgrid.gforge.inria.fr/`