Trace Visualization

Lucas M. Schnorr
CNRS

Viz @ CERN's ATLAS DDM
Geneva, Switzerland – February 22nd, 2012

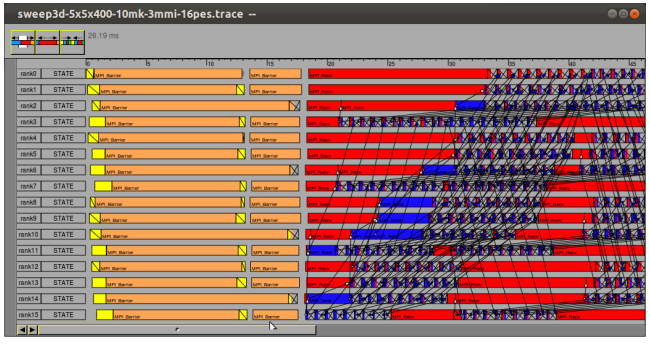
Introduction

- Analysis of large distributed systems
- Considers a set of assumptions
  - System achieves steady state (throughput)
  - System is network/cpu-bounded
  - System is fair (resource sharing)
- Verify them considering different workloads
- Detect anomalies, different behavior
- New assumptions based on the analysis

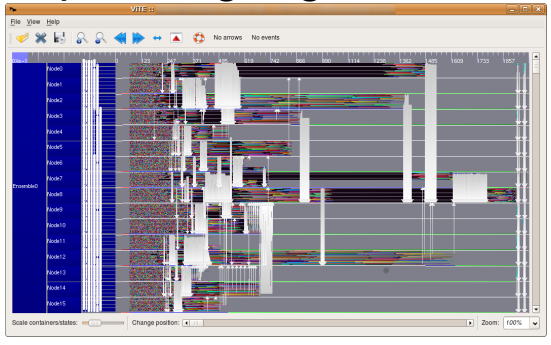
Review: Common techniques

- Plots, statistical charts
- Space/Time views (or Gantt-charts)
  - Space (vertical axis) *vs* Time (horizontal)
  - Detailed view of every single trace event
  - Gives the causal order among events
  - Several tools, widespread, useful

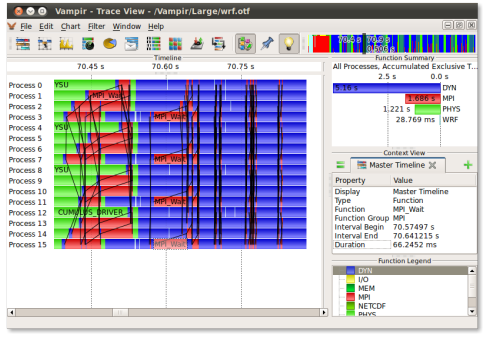
Paje
http://paje.sf.net



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



Vampir
http://vampir.eu



- Scalability issues, lacks network topology

Our approach

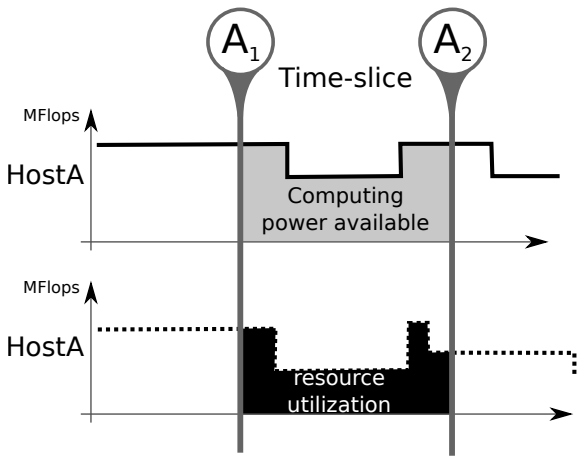
1. Register system behavior
   - timestamped data, per monitored entity
   - as much detailed as possible
2. Space/Time data aggregation
   - define space/time neighborhood
   - configure aggregation operators
3. Interactive visualization techniques
   - change data aggregation when needed
   - exploratory approach
   - different techniques

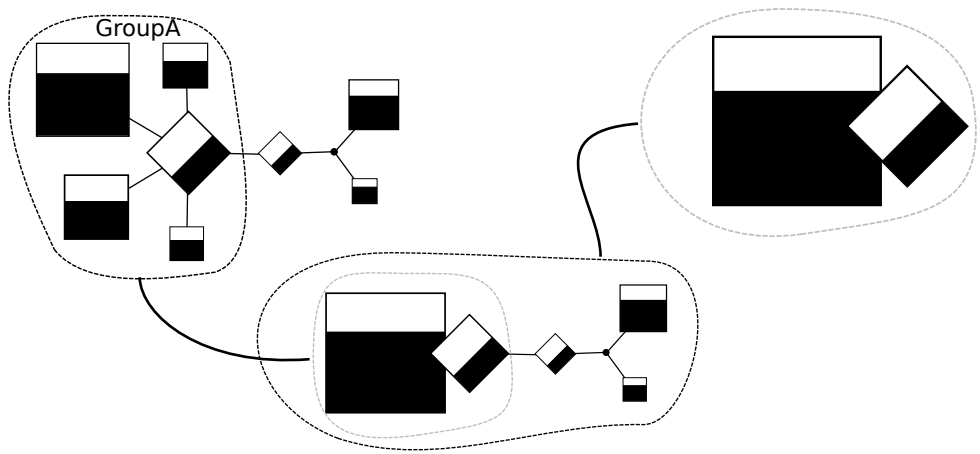
1. Register system behavior

- Real world
  - Distributed, large-scale
  - Different administrative domains
  - May be very hard to register system behavior
- Simulated world
  - Much easier, everything at hand, zero intrusion
  - Detailed resource capacity and utilization
  - Easily extensible (if more data is needed)
- **SimGrid** instrumentation
  - Categorized resource utilization tracing
  - Application-dependent user-defined variables
  - See `--help-tracing` for details

2. Space/Time data aggregation

- Analysis at small and large scales
- Reduce or highlight outliers
- See local or global behavior
- Interactive approach, analyst defines
  - Aggregation operation
  - Data neighborhood
    - Example: time slice





- Note: behavior attenuation

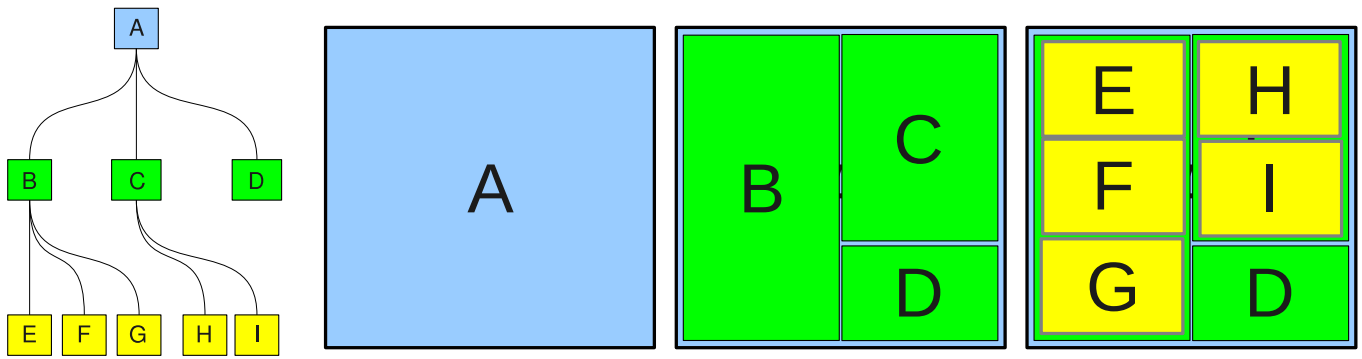
3. Visualization analysis

- Useful, based on case-studies
- Configurable according to user needs
- Interactive, exploratory

- Use aggregated data: in time and space
- Visualization techniques
  1. Squarified **treemap** analysis (the tree view)
  2. **Topological** analysis (the graph view)

Squarified treemap analysis (tree)

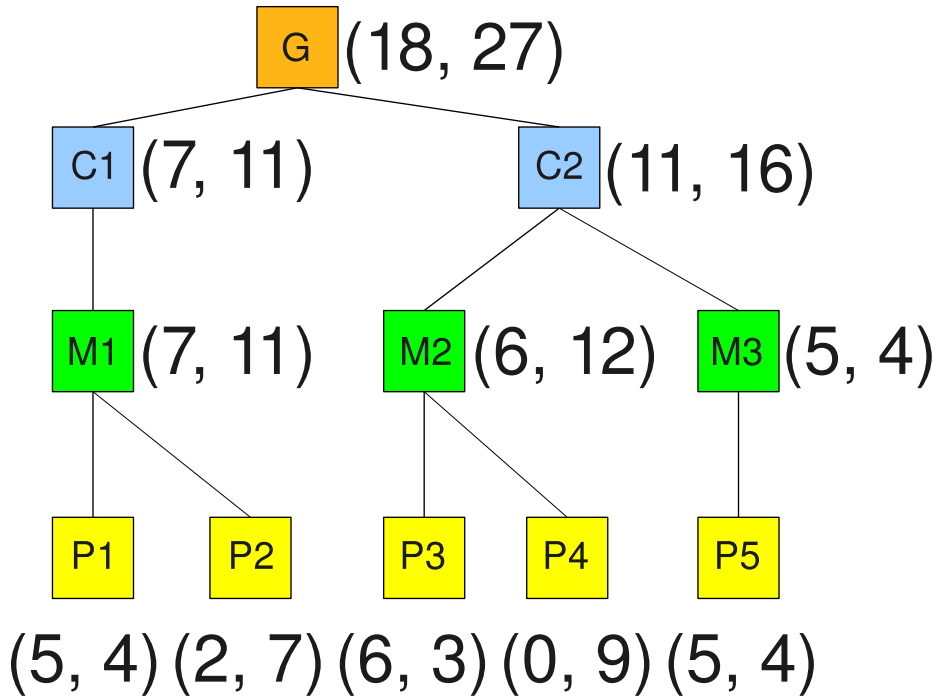
- Scalable hierarchical representation
  - Nodes are annotated with values
- Top-down recursive drawing
- Space-filling algorithm



- Squarified version
  - keep rectangles ratio close to 1

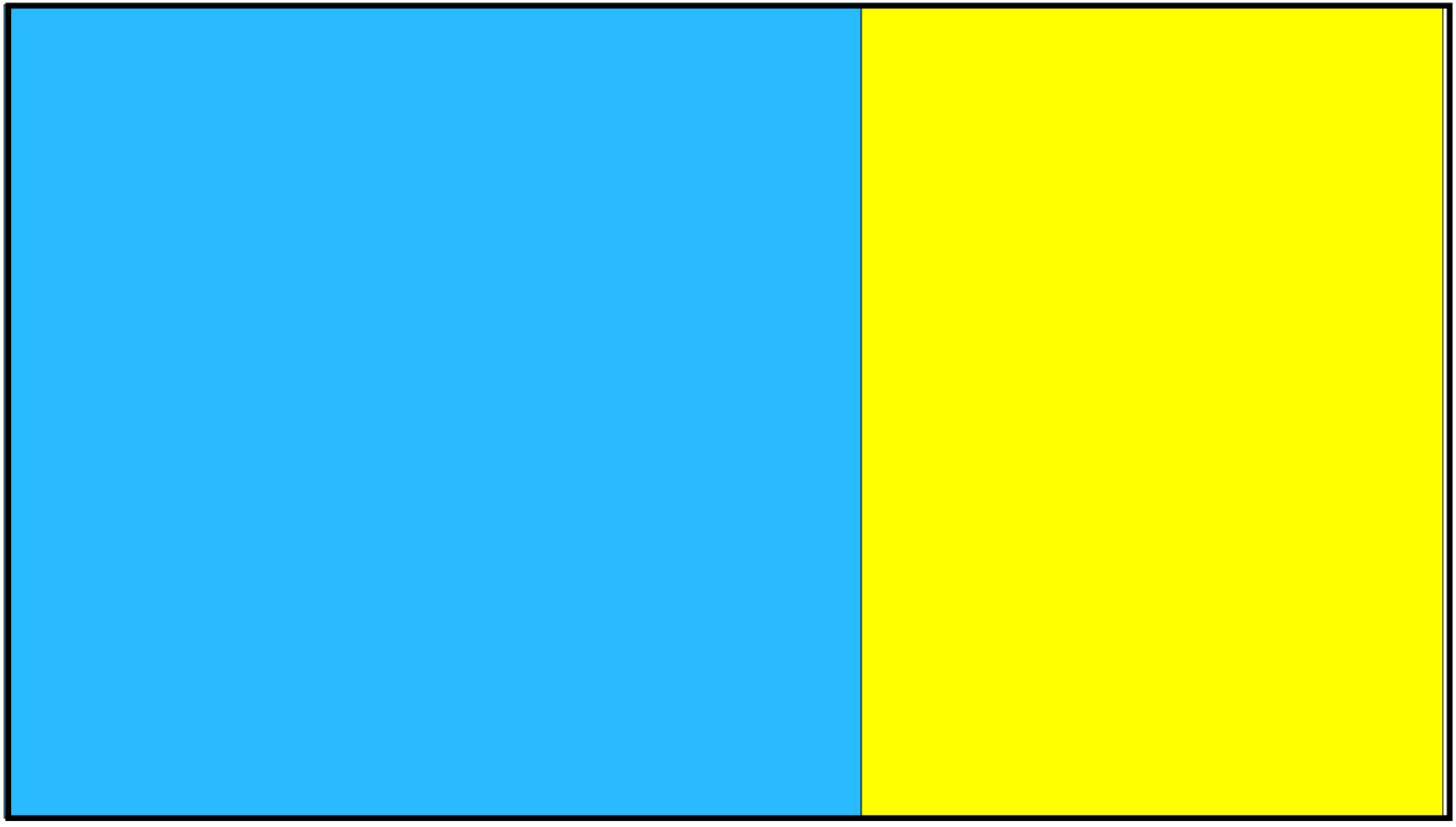
Treemap: our approach

- Monitored entities organized as a hierarchy
- Aggregated data is used in inner nodes



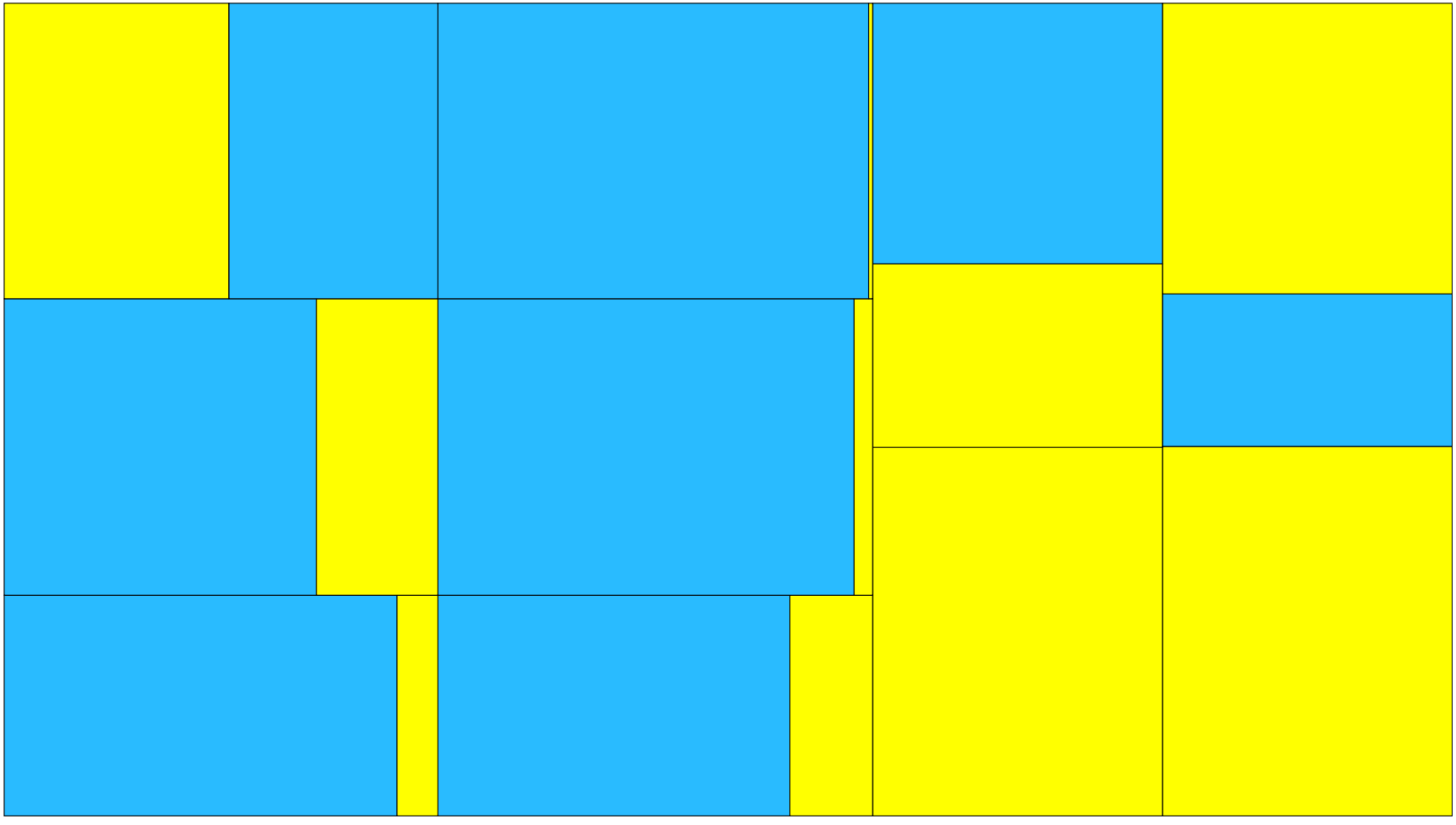
Treemap: a synthetic example

- Map trace metrics to screen space
- Full to less aggregated



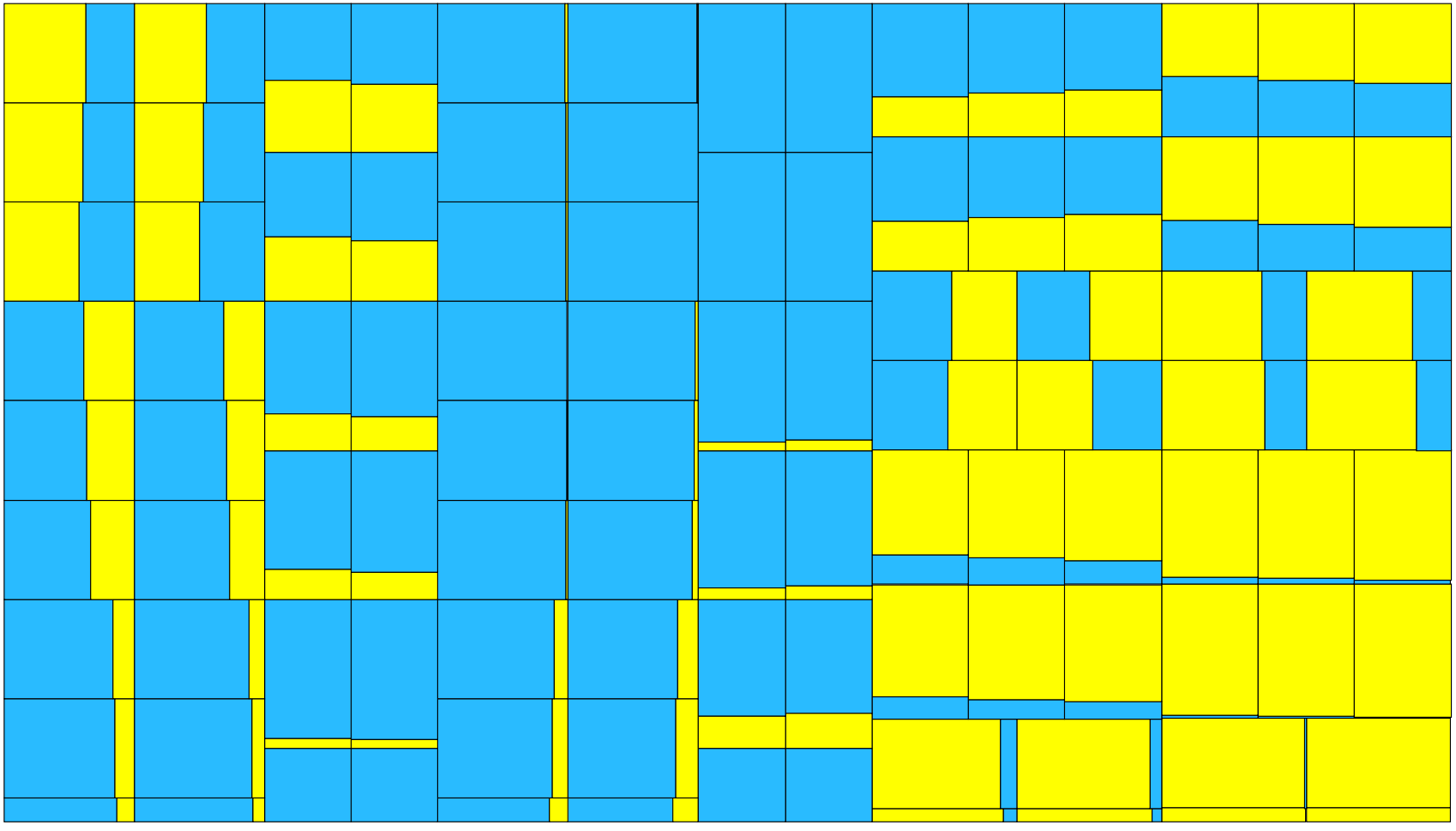
Treemap: a synthetic example

- Map trace metrics to screen space
- Full to less aggregated



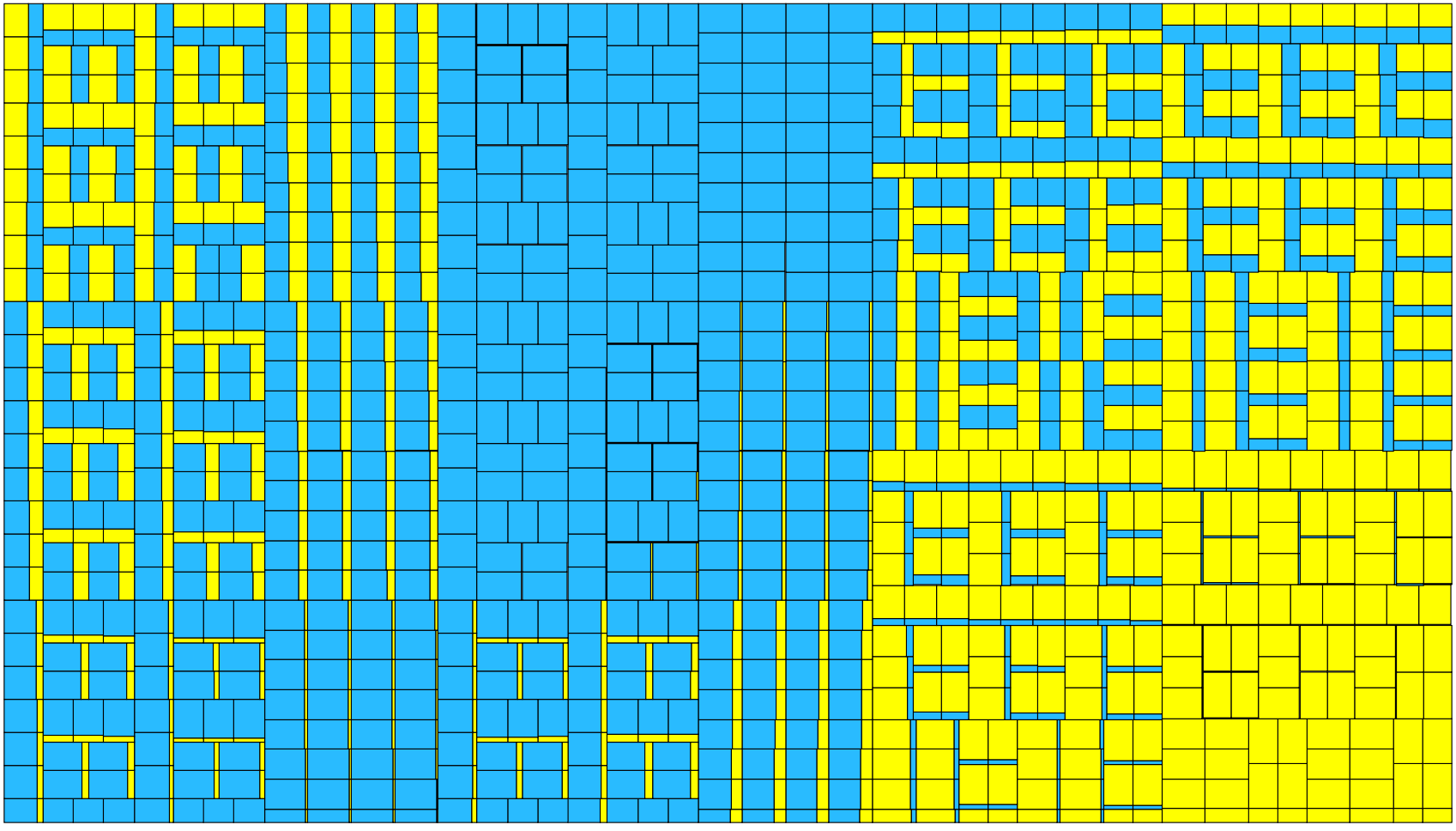
Treemap: a synthetic example

- Map trace metrics to screen space
- Full to less aggregated



Treemap: a synthetic example

- Map trace metrics to screen space
- Full to less aggregated

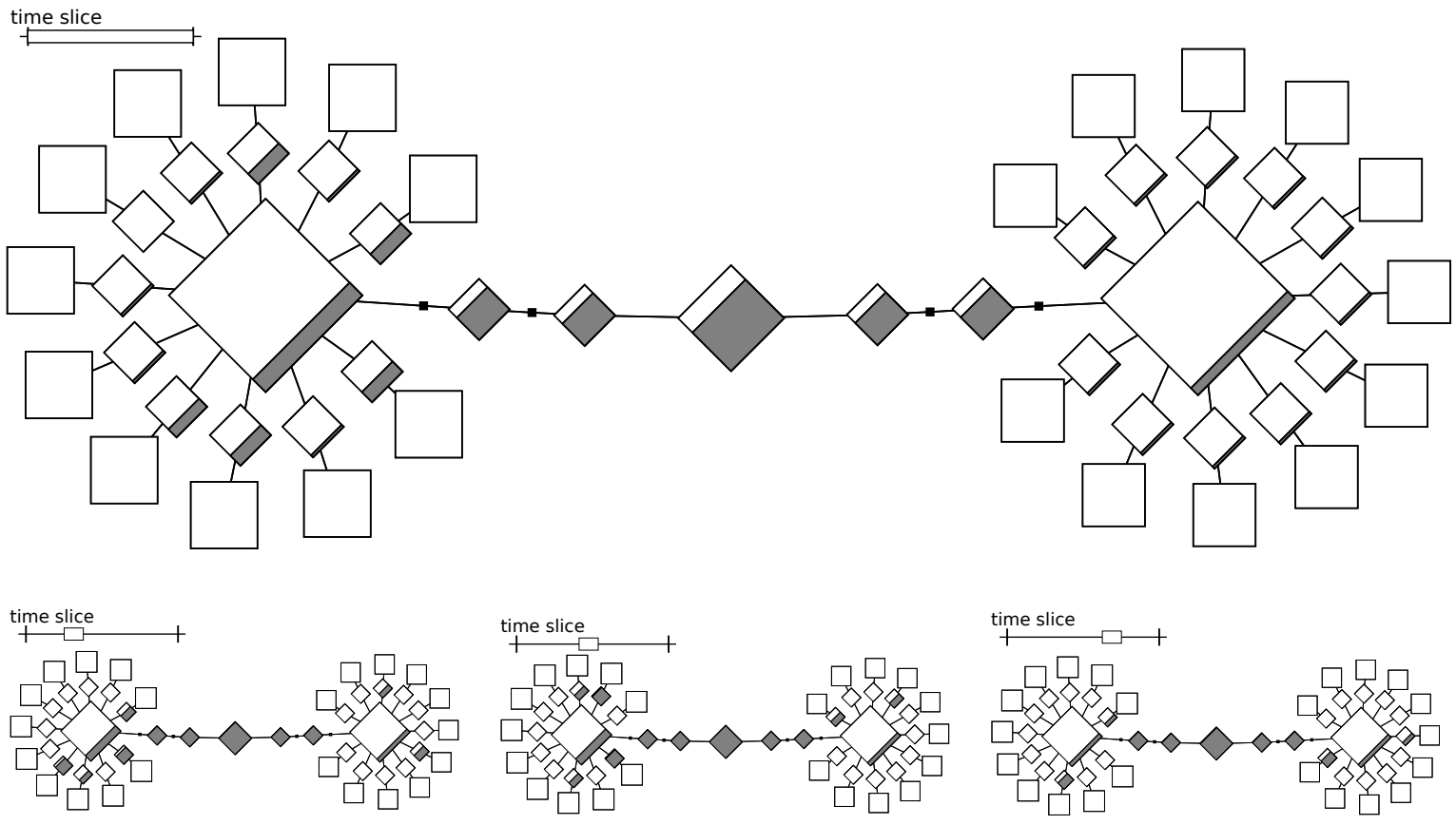


Topological analysis (graph)

- Mapping monitored resources to nodes
- Scalable topological representation
- Resource groups
  - Based on geo or logical location
  - Used to define a resource hierarchy
- Use aggregated data

Topological analysis: visualization

- Map trace metrics to geometrical attributes
- Network structure, application-level metrics
  - Multiple metrics on the same analysis



Roadmap

- Squarified treemap
  - Algorithm scales well: 20 thousands entities
  - Good interactivity on scale
  - More scenarios needed
- Network topology scalability
  - Today: limited to 150 entities (hosts + links)
  - Graph dynamic positioning (force-directed)
- **Case-studies dependent**

Tools

- Triva
  - Visualization tool for trace analysis
  - Based on GNUstep and Paje
  - http://triva.gforge.inria.fr
- SimGrid
  - Scalable simulation of distributed systems
  - Accuracy effort, fast implementation
  - http://simgrid.gforge.inria.fr
- More information:
  http://mescal.imag.fr/membres/lucas.schnorr/