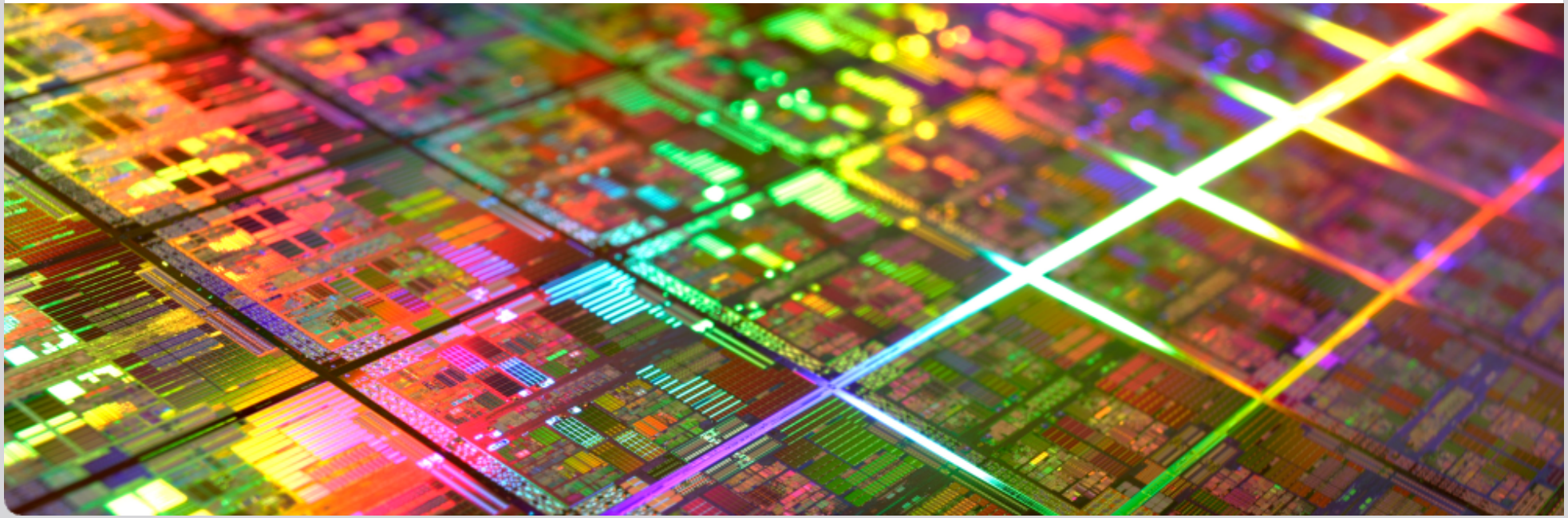
Karlsruher Institut für Technologie

Methods and Strategies to Rate and Optimize Transactional Memory Applications

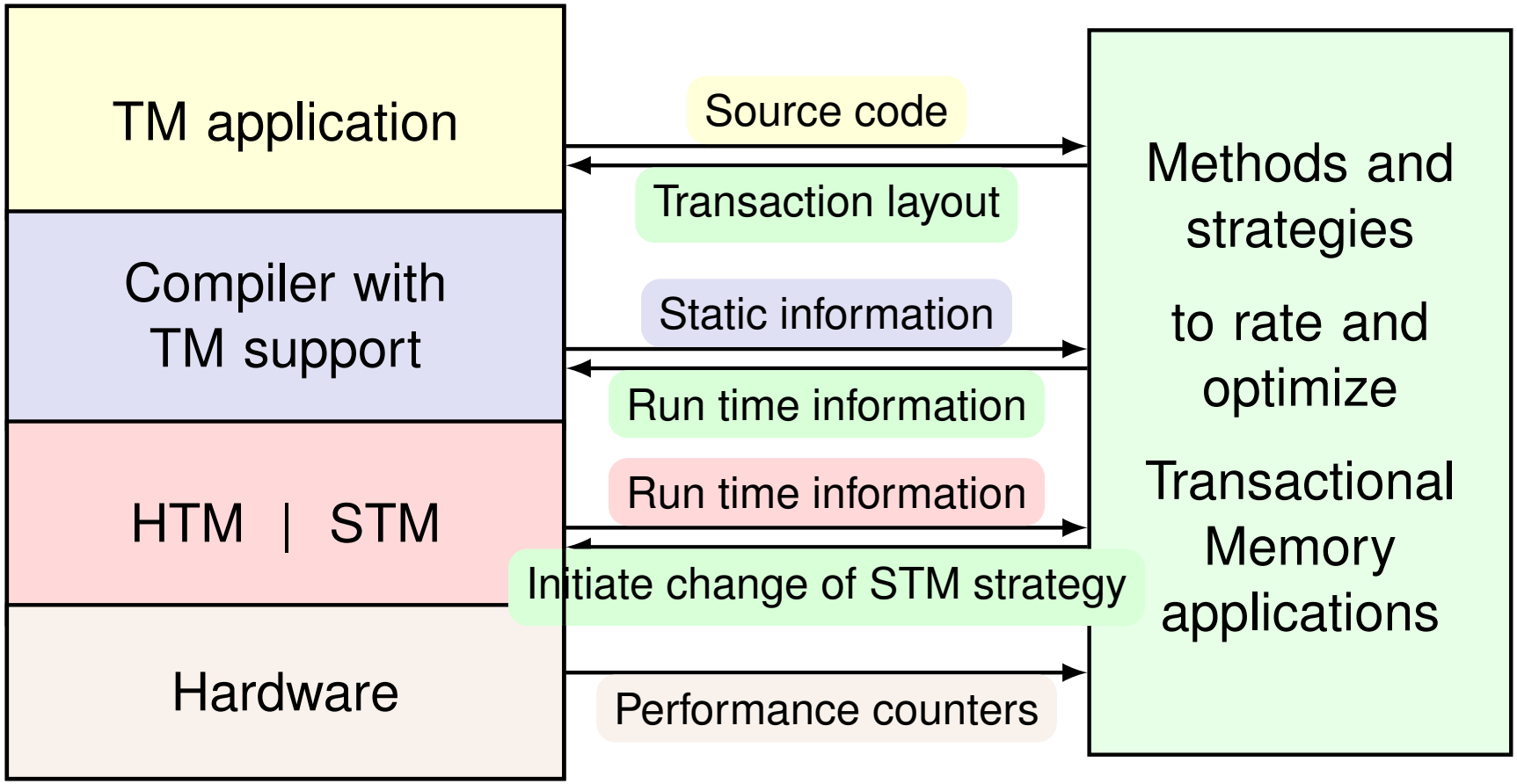
Martin Schindewolf, Wolfgang Karl

Chair for Computer Architecture and Parallel Processing

20th May, 2011



TM-Opt: Overview



TM-Opt

Karlsruher Institut für Technologie

Goal

- Support and aid the development and optimization of TM applications
- Aggregate and analyse information from all layers of the TM system
- Guide application developer when optimizing

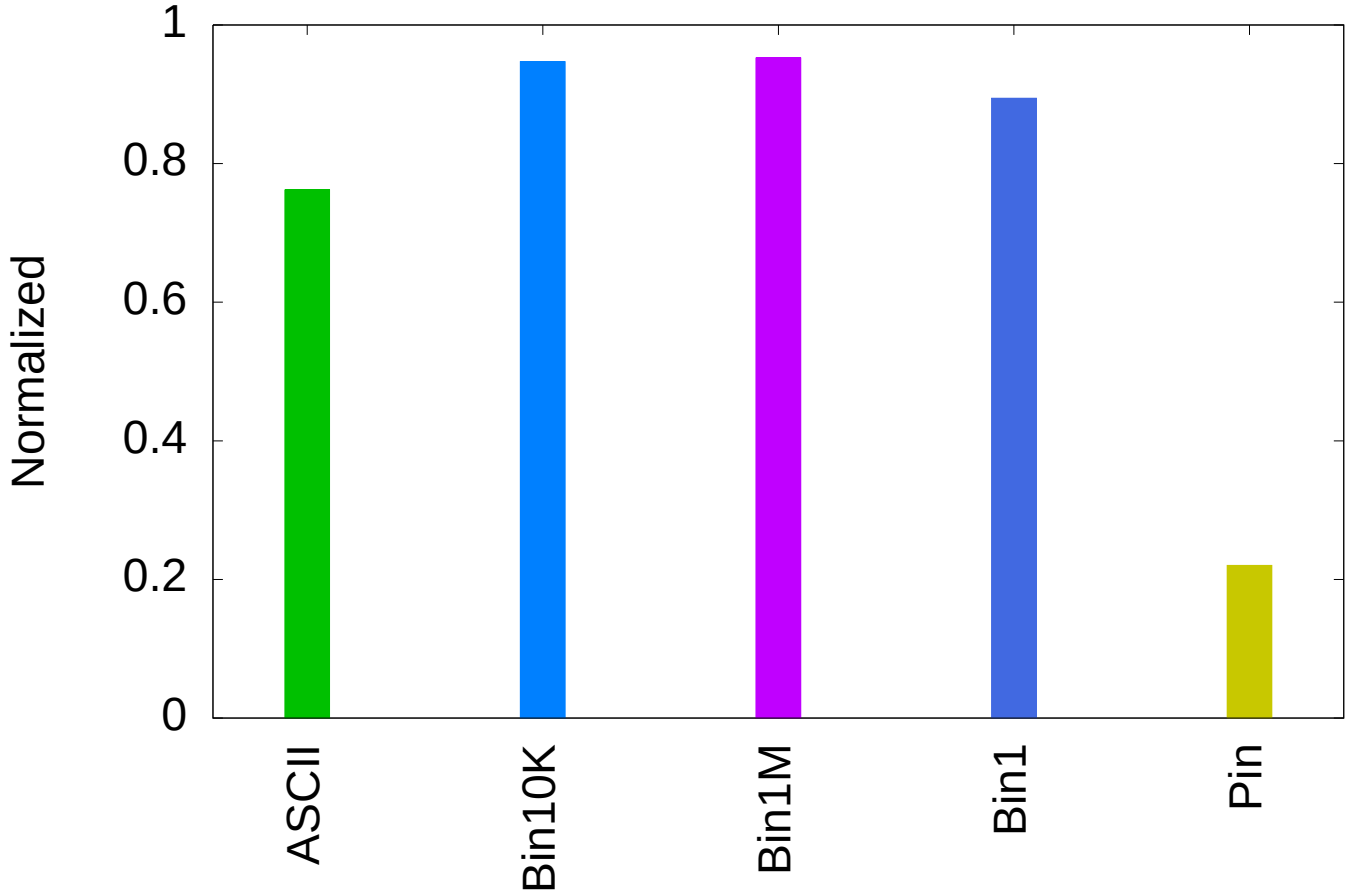
TM-Opt

Karlsruher Institut für Technologie

Information retrieval in STM systems

- TracingTinySTM logs STM events (start, commit, abort, load, store) in trace files
- Compresses buffers with zlib or lzo (multi-threaded trace compression!)
- Comparison with Pin:
  - Smaller influence on application behavior
  - Higher throughput (Txn/s)
- Evaluation with STAMP benchmarks
  - Run times of different tracing variants
  - Size of log files

TracingTinySTM – Bank TM application



- Normalized to execution w/o tracing
- Former throughput [Txn/s]

TM-Opt: ARCS 2011 – MAPT

Karlsruher Institut für Technologie

Exploiting and analyzing static information (LLVM)

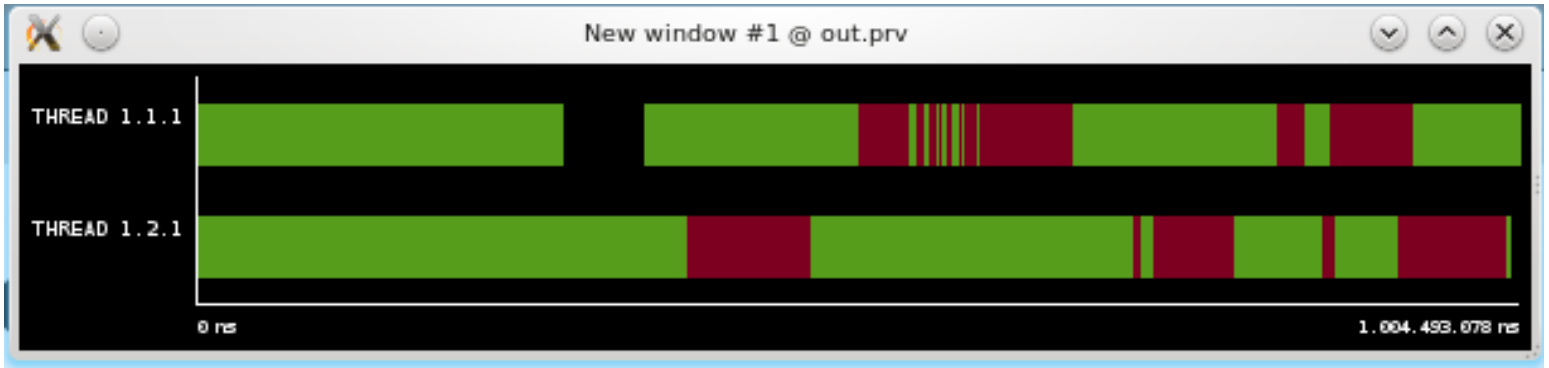
- Analysis of Memory Access Patterns in Transactions: MAPT
- Granularity of conflict detection influences run time
- Conflicts are detected falsely (False Positives)
- Problem is analog to the false sharing in caches

→ Exploit compiler information

- Extract and analyze the memory accesses

→ Compiler proposes the superior conflict detection granularity: between cache line and word-based conflict detection

Visualization of TM applications

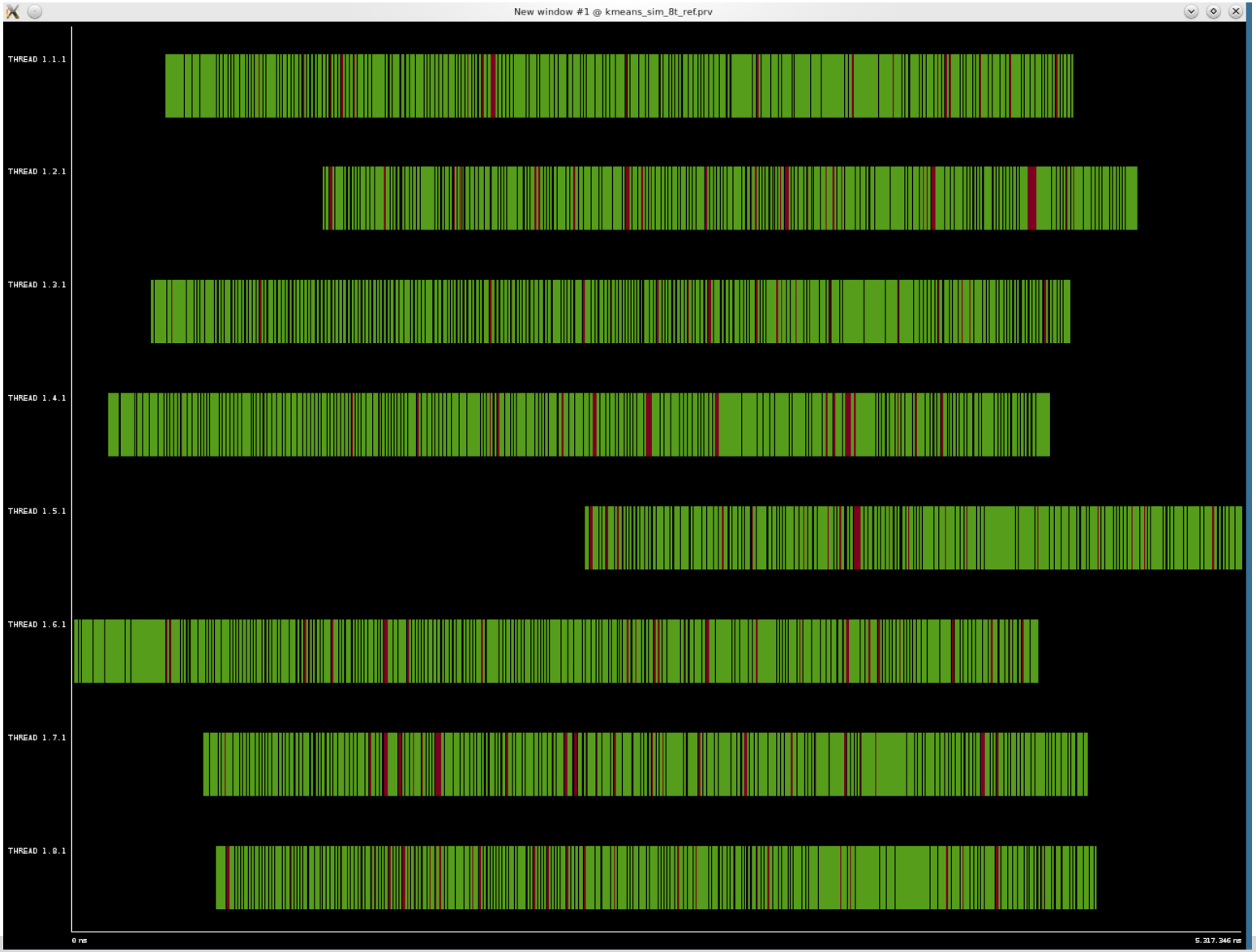


Paraver (UPC/BSC)

- Abort/Commit coded as states
- Loads/Stores as events
- Time transformation

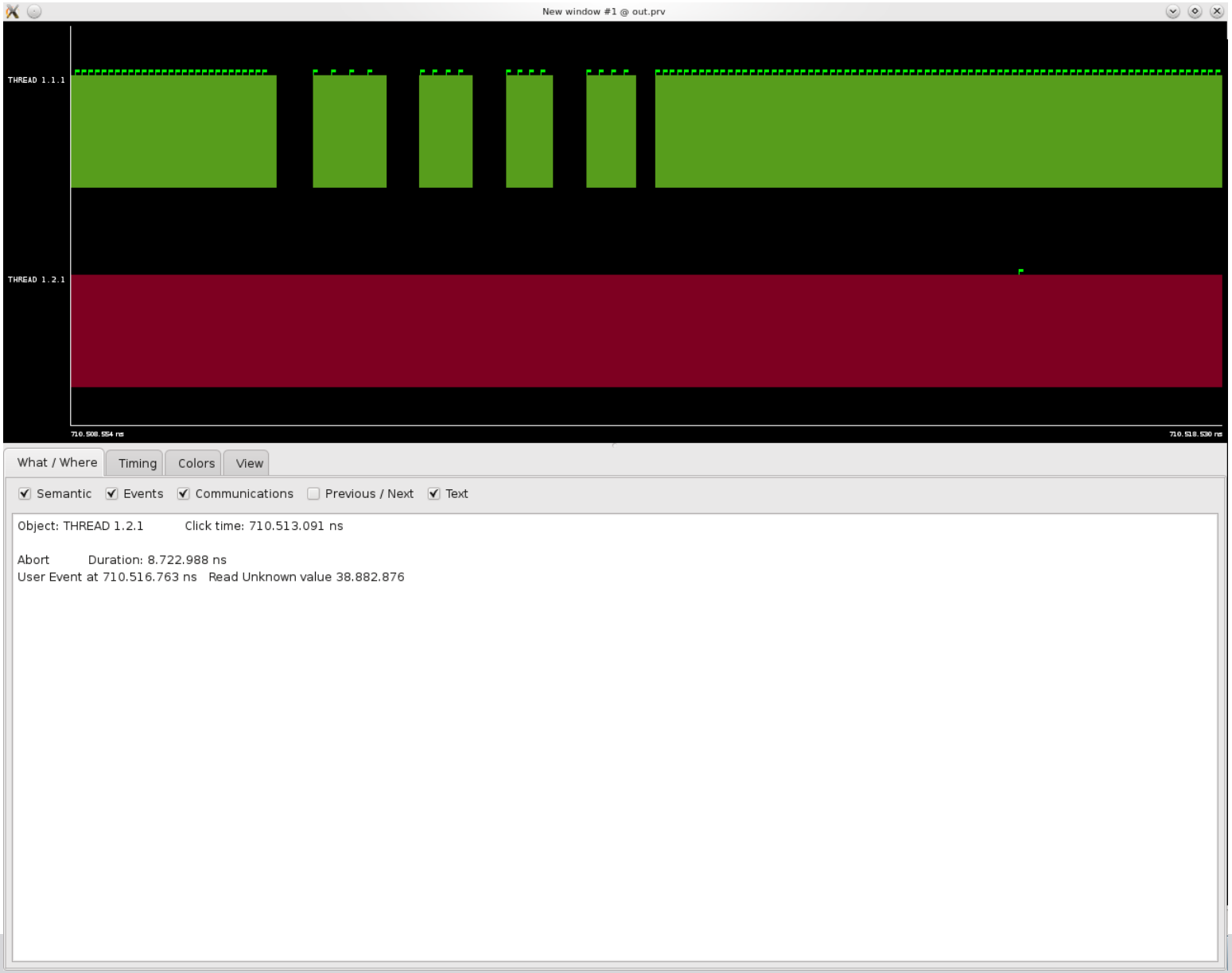
Visualization of kmeans (from STAMP)

Karlsruher Institut für Technologie



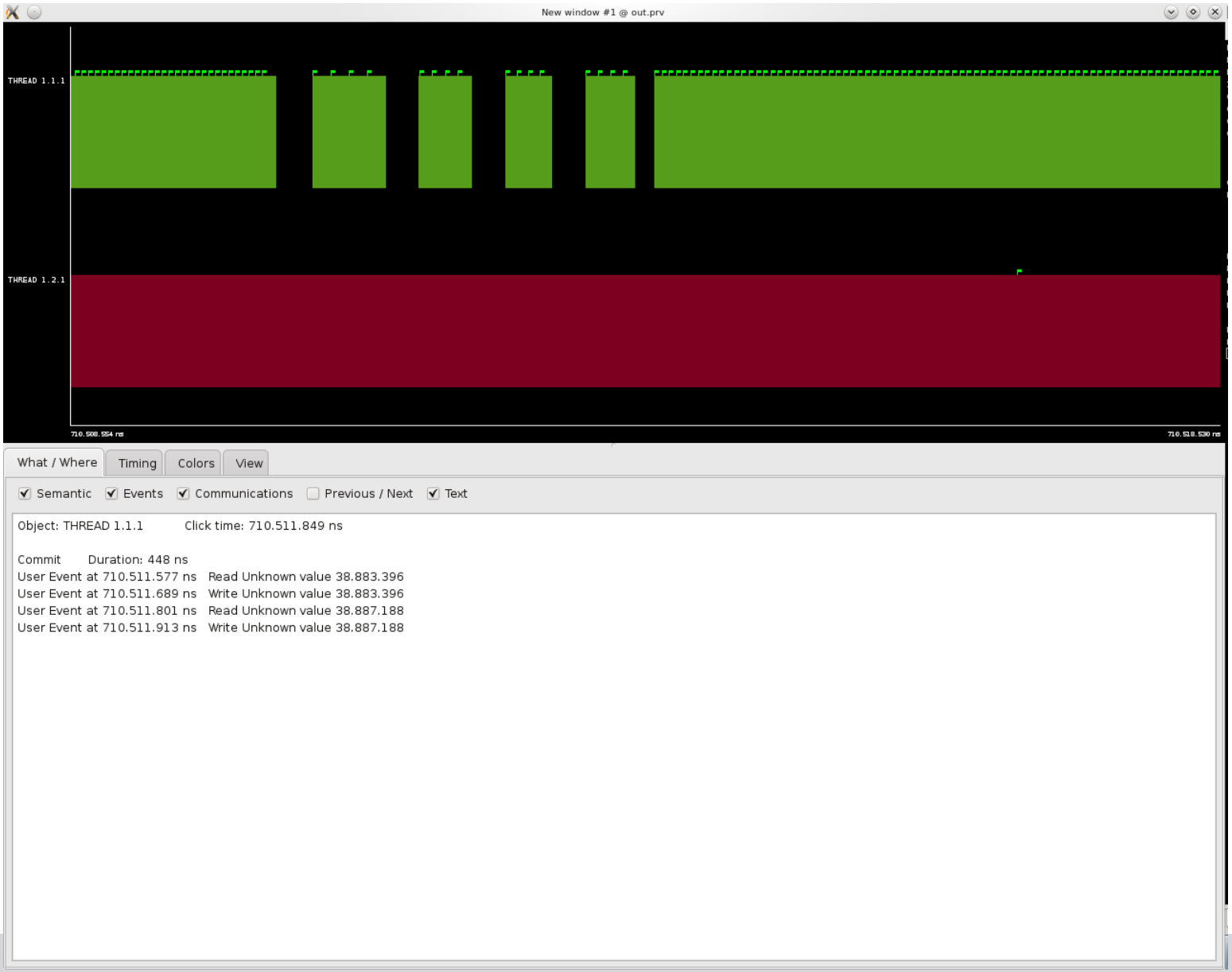
Visualization of TM applications III

Karlsruher Institut für Technologie

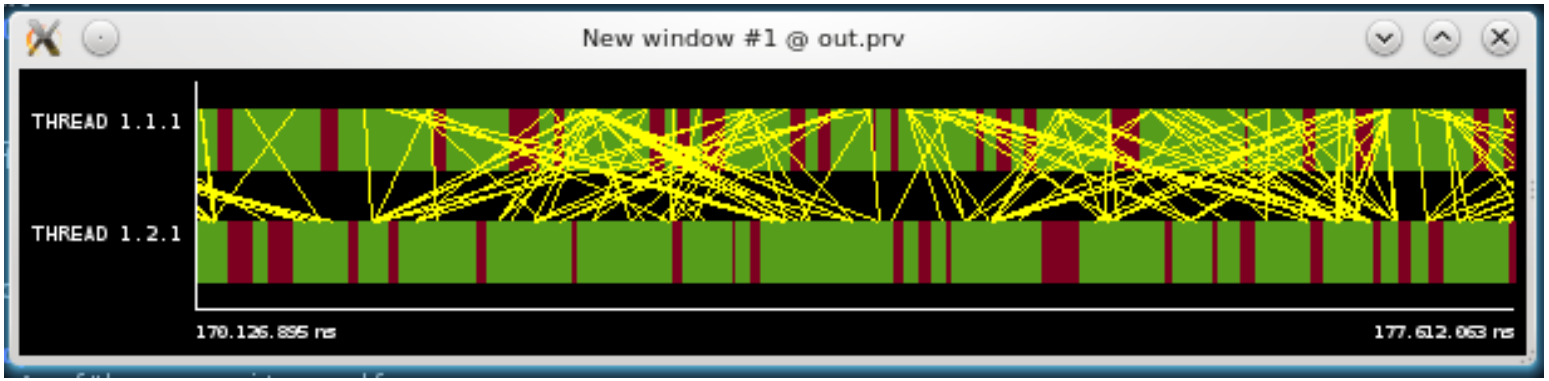


Visualization of TM applications IV

Karlsruher Institut für Technologie



Visualization of TM applications V



Visualize dependencies between Txns

- Analysis identifies dependencies
- Code as separate event (cf. MPI communication)
- Visualize in Paraver (zoomable)

→ Find source of conflicts

Optimization of TM applications

Karlsruher Institut für Technologie

Optimizing the layout/size of transactions

- Transaction size is critical:
  - Overhead for setup/destroy of transaction must amortize through parallelism (longer runs are better)
  - Long Txns waste a lot of work when rolled back
- Identify transactions with low/high conflict potential
  - Compare complete Read/Write Sets

  → Avoid chasing conflicts one after the other
  - Optimize Txns:
    Fuse Txns with low conflict potential
    Split Txns with high conflict potential
- Determine beneficial size of Txn experimentally
  Goal: Automatically detect cases where optimization is promising