

Real-time system analysis using tracing and sampling data

François Rajotte

Laboratoire DORSAL
Département de génie informatique

POLYTECHNIQUE
MONTRÉAL



AFFILIÉE À
L'UNIVERSITÉ DE MONTRÉAL



Outline

- Context
- Statistics improvement
- Oscilloscope view
- Future steps
- Conclusion



- Tracing and real-time applications
 - Low-overhead system observation
 - Provides detailed information
- Challenges
 - Extracting meaningful data
 - Statistics, abstraction
 - Facilitate user exploration
 - Tools, viewers



Current TMF statistics view

- Generic view
 - Only shows event count
- Special features
 - Usage of a state system
 - Cumulative event count
 - 2 queries per time range
- What about other statistics?
 - CPU time of processes

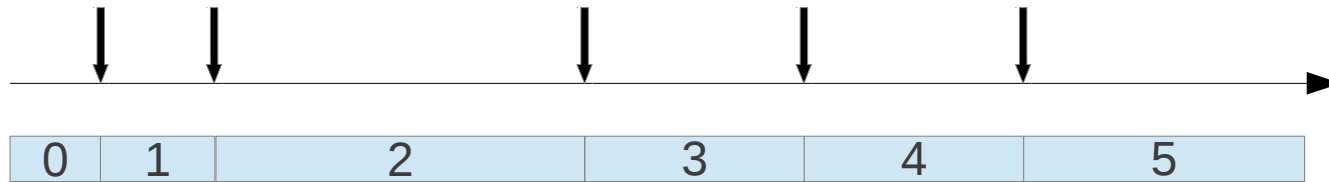


- Similarities
 - Cumulative time spent
- Dissimilarities
 - Tracking intervals instead of punctual events
 - Hierarchical structure



Tracking intervals statistics

Event statistics

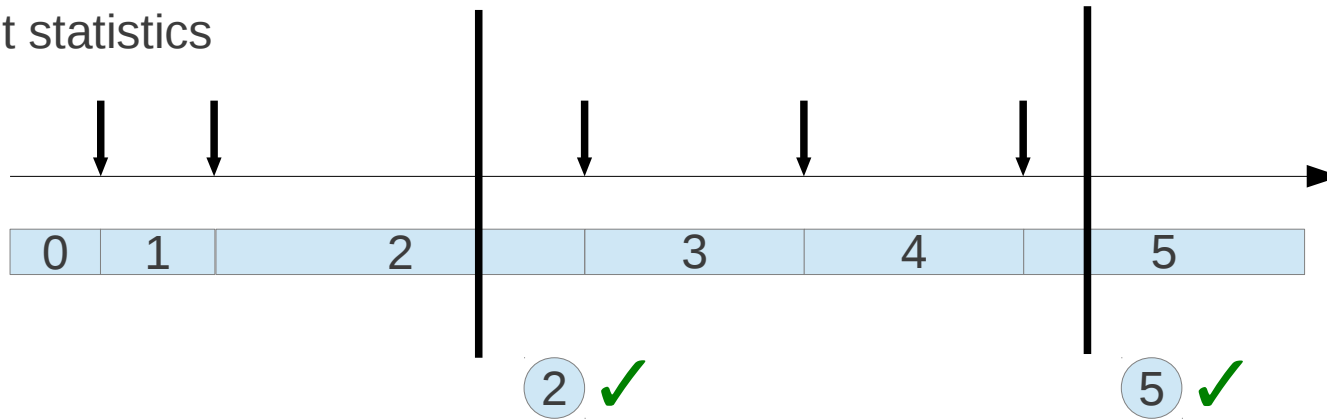


Interval statistics

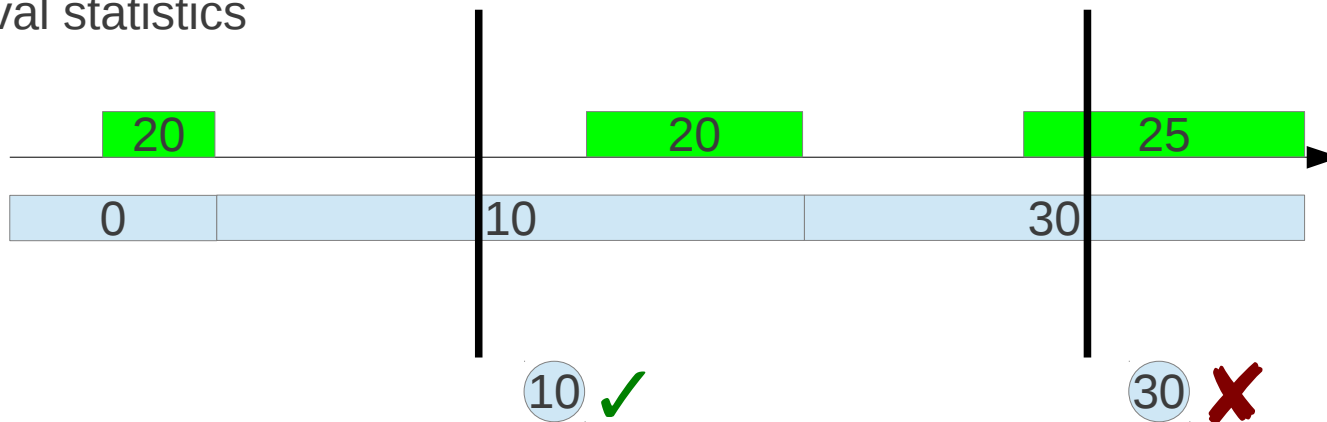


Tracking intervals statistics

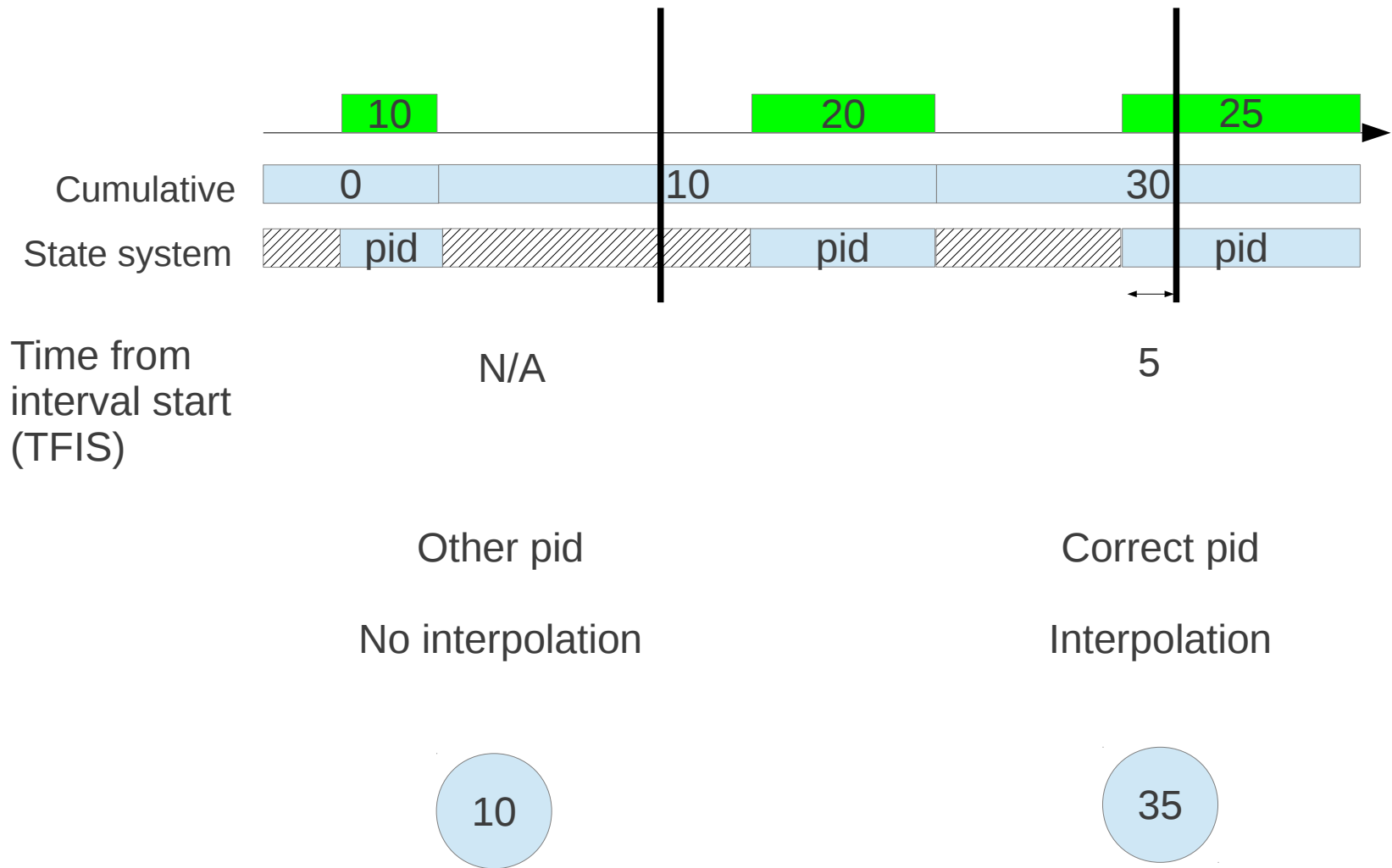
Event statistics



Interval statistics



Tracking intervals statistics



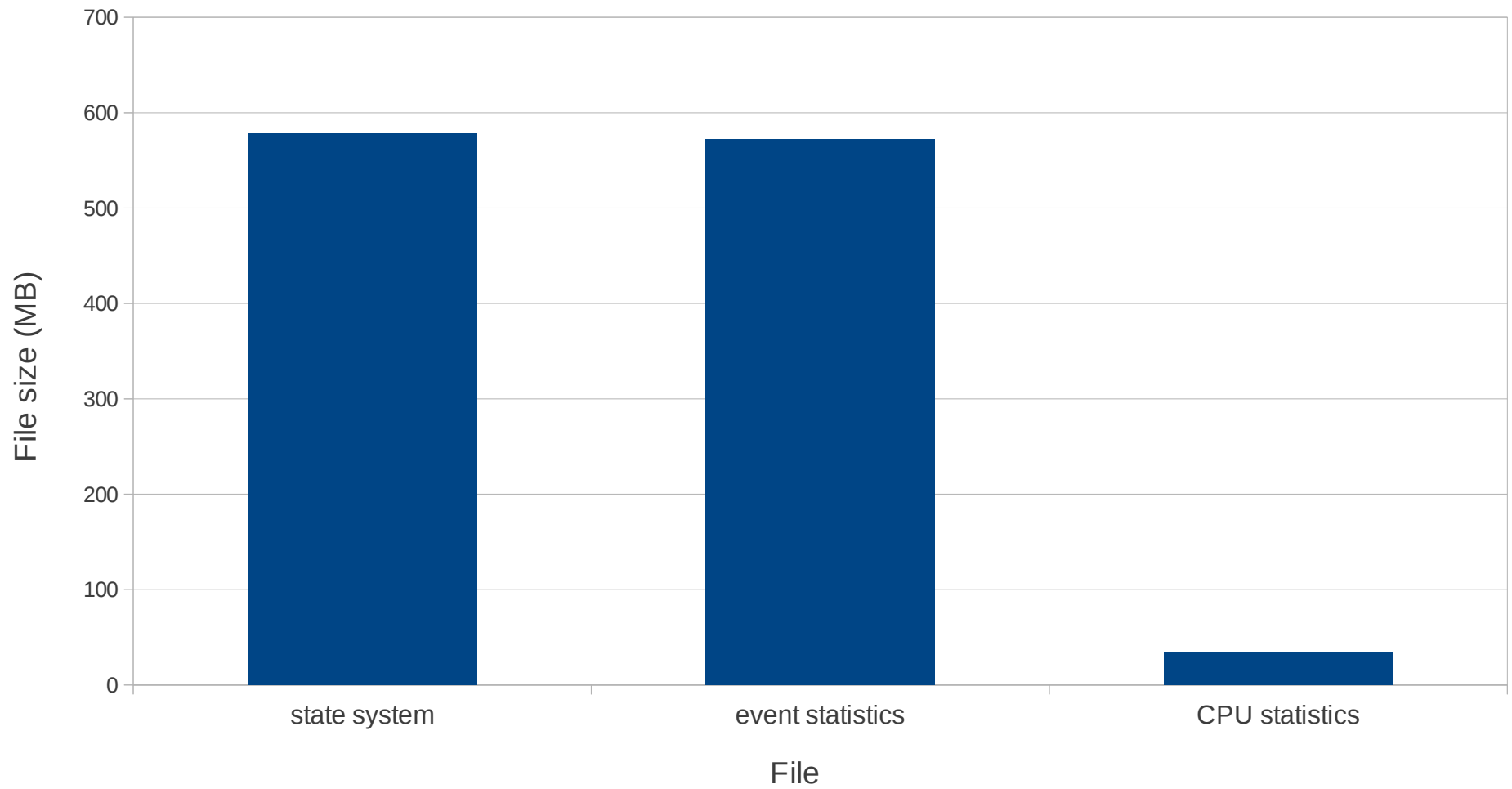
Analysis of approach

- New state value: 8-byte long
 - Required for long-running traces at nanosecond resolution
 - Uses string data location
 - Intervals are 33 bytes (vs 25 bytes for 4-byte int)
- Number of intervals stored equals number of sched_switch events



File size comparison

Trace: 286MB, 11M events, 1M sched_switch



- Motivation
 - Better understanding of real-time processes
 - Finding abnormal behaviour
- 2-step approach
 - Split the trace into repeating snippets
 - Display the snippets

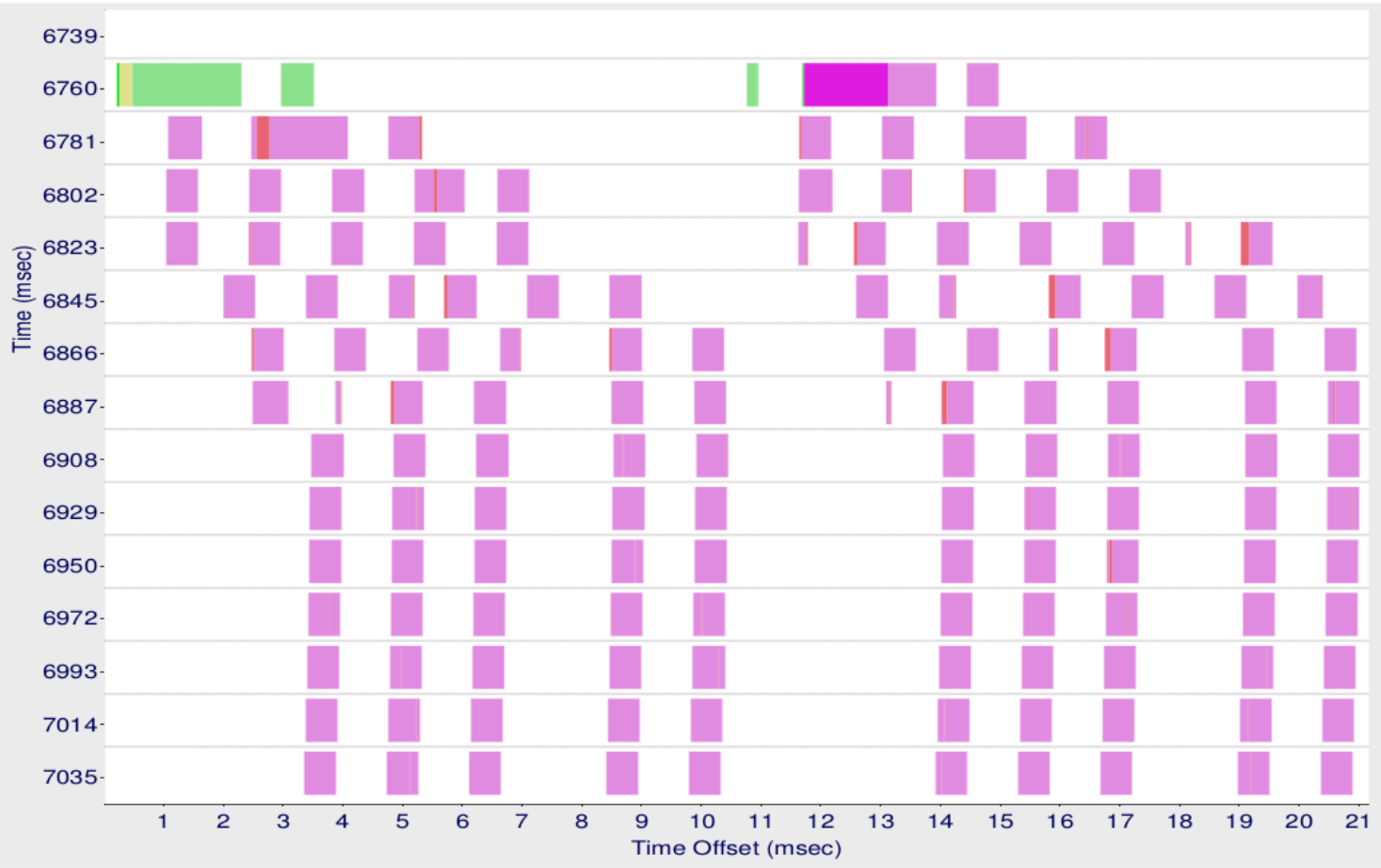


Similar approaches

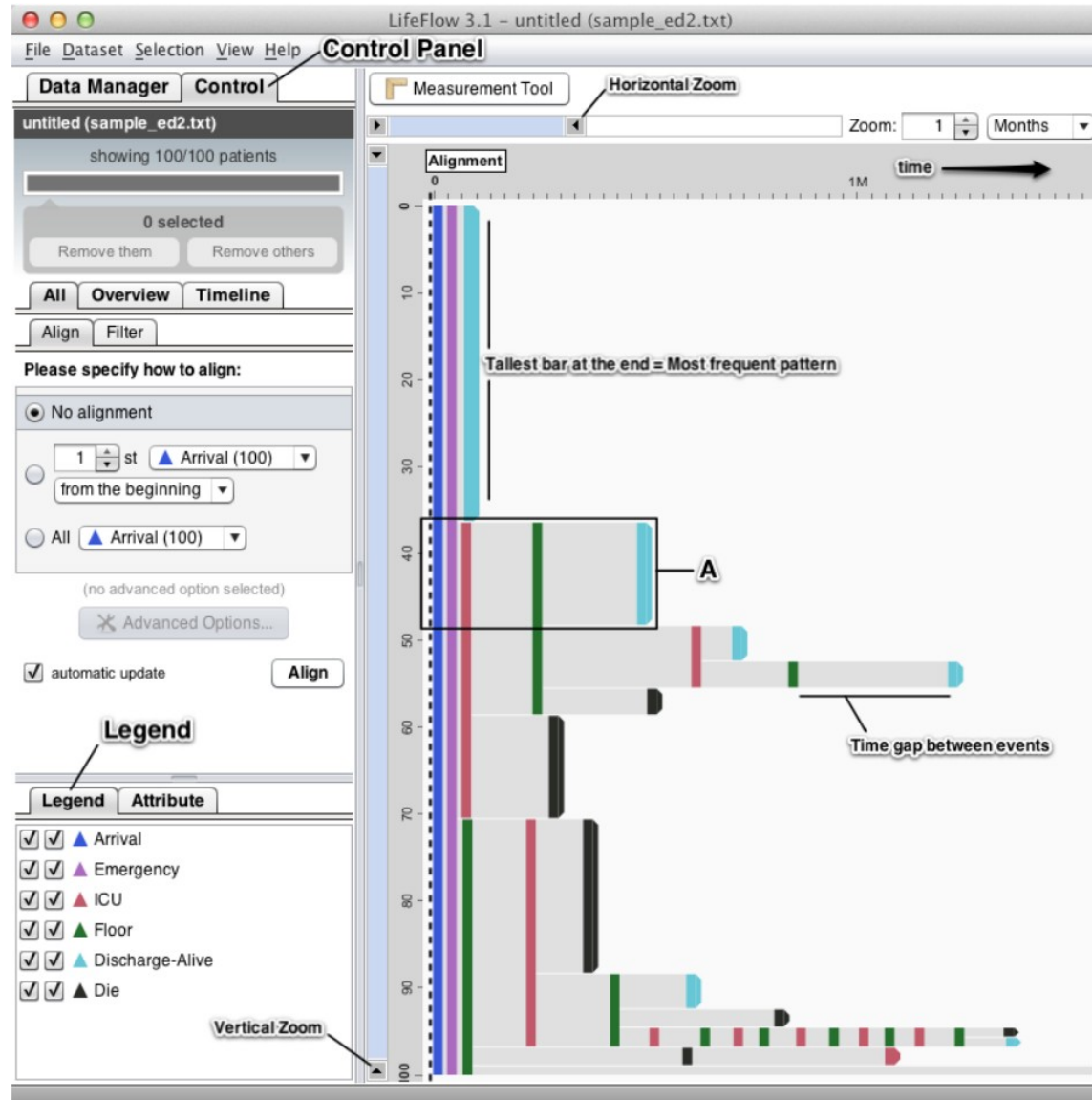
- TuningFork
 - Folding of a trace using a manually-given fixed period
- LifeFlow
 - Separate sequences of events
 - Grouping by similarity



TuningFork



Lifeflow

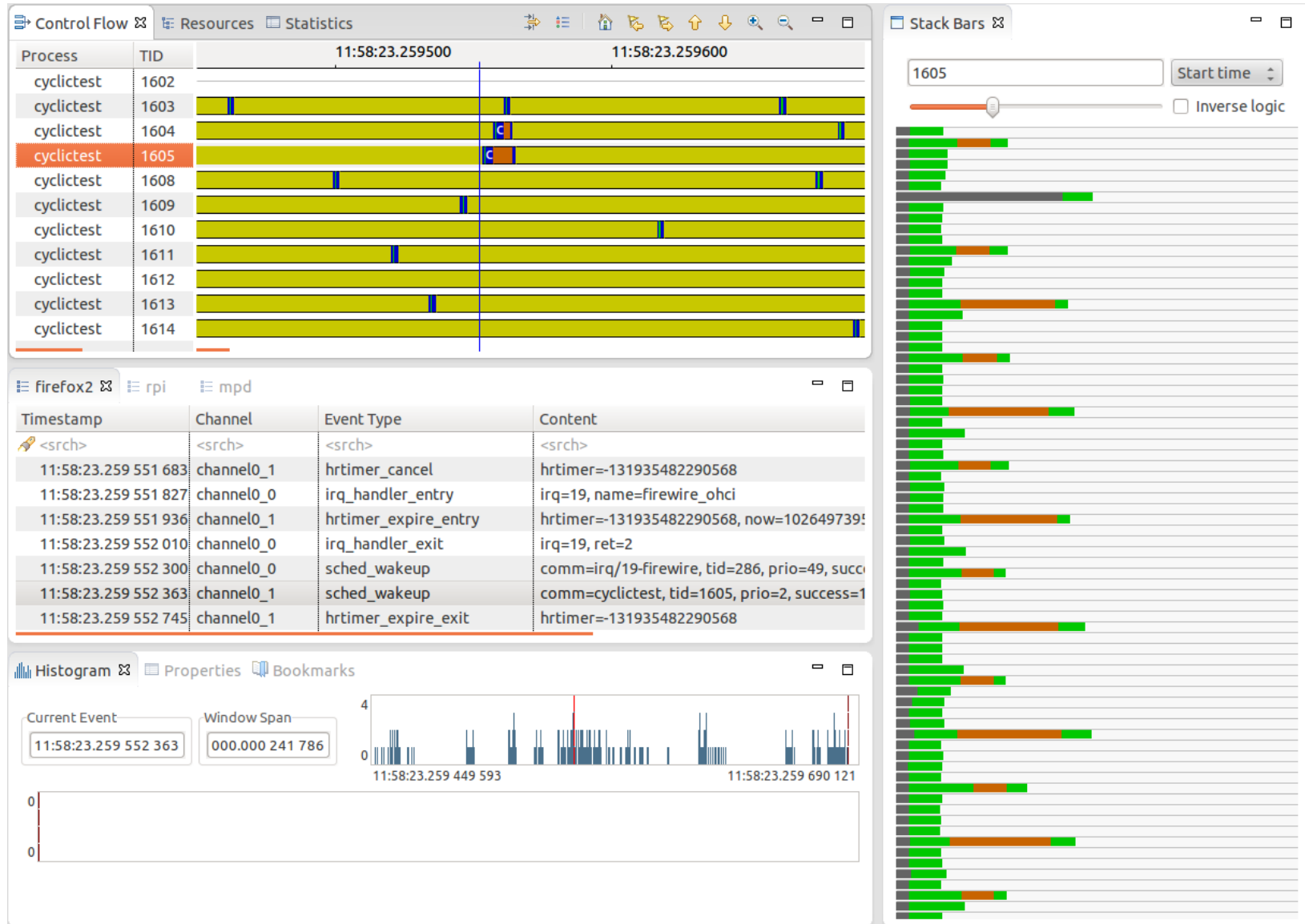


Proposed approach

- Splitting the trace into snippets
 - Using a specific event
 - Based on trace knowledge (e.g. timer or scheduler events), or
 - Based on a manually-defined event
 - Simple and fast
- Displaying the snippets
 - Using TMF infrastructure
 - Synchronisation with other views
 - Stacked snippets
 - Direct comparison



Example



Analysis of approach

- Different features highlighted using simple sorting
 - By chronological order
 - Repeating behaviours
 - By total snippet time
 - Abnormal snippets (e.g. long execution)
- Current prototype limitations
 - Storage in memory
 - Number of snippets displayed limited by screen size



Future steps

- Improve on the current oscilloscope view
 - Grouping of snippets
 - Using feedback from experts
- Integration with related work
 - i.e. Critical path analysis
- Performance analysis and benchmarks

