Two Roads to Extreme Scale Event Trace Analysis?

Andreas Knüpfer, Thomas Ilsche, Robert Schöne, Bert Wesarg

VI-HPS Workshop on Extreme Scale Programming Tools

Denver, 2013-11-18

Overview

- Evolutionary
- Revolutionary
  - Semantic compression for event traces with C3G
  - Event trace recording based on sampling
  - Combination
- Conclusions and Outlook

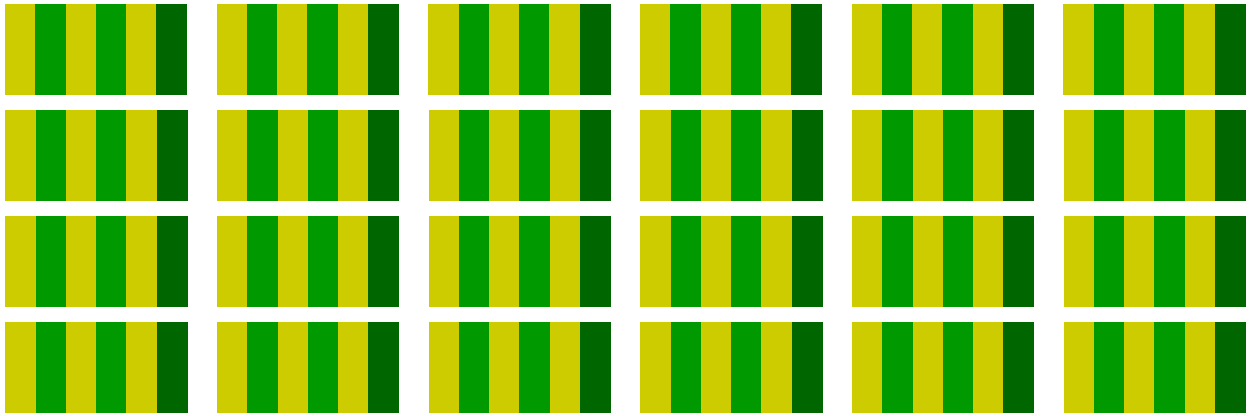
Motivation

- Performance Analysis and Tools
  - Vital prerequisite, operational at scale
  - Combine alternative approaches and methods
- Evolutionary
  - Scale up the established state-of-the-art tools
    … event trace recording and analysis in our case
  - Fixed percentage of the resources of the target application (5-15%)
  - Sensible and balanced overhead does not hurt (5-15%)
- Revolutionary …

Semantic Compression for Event Traces

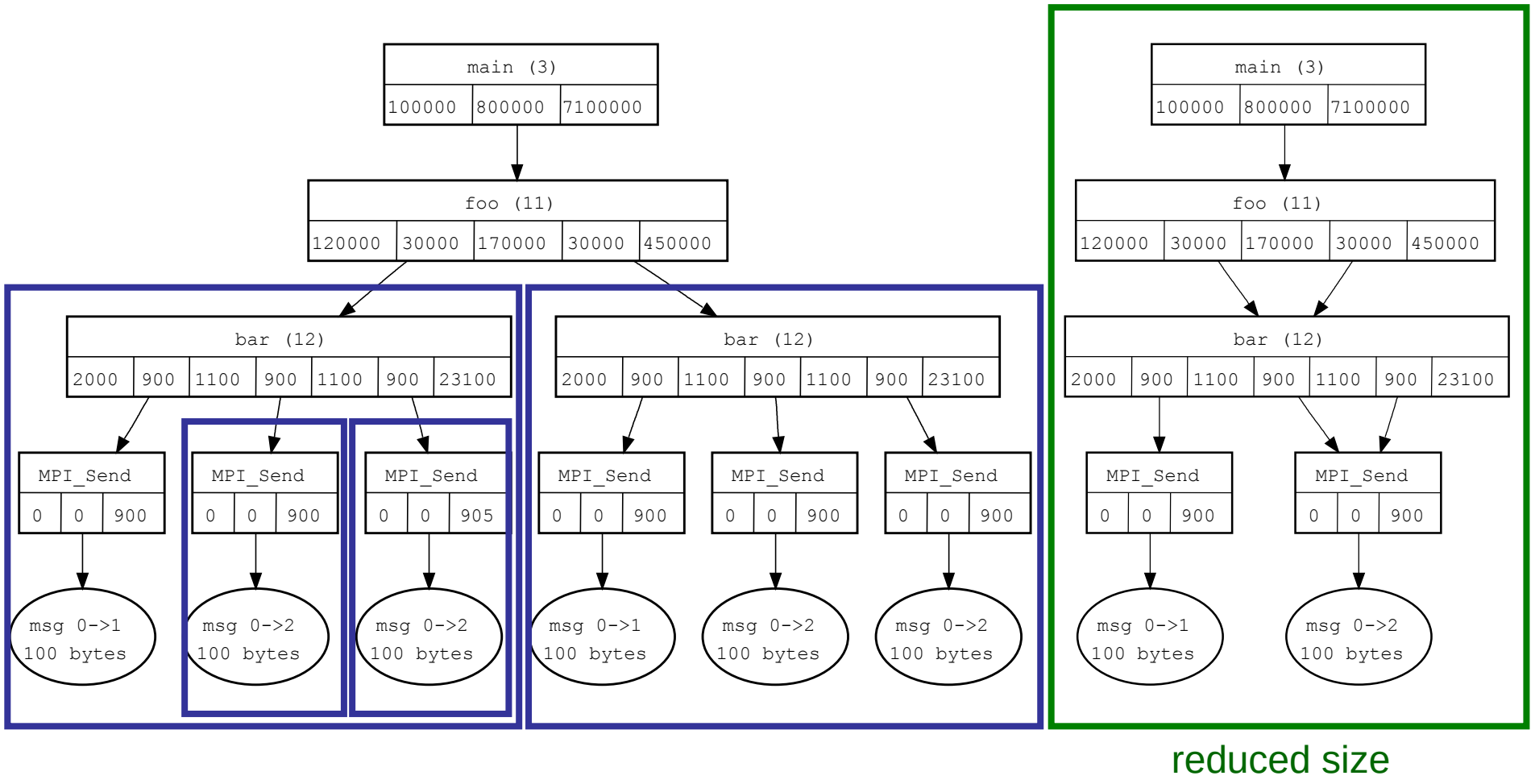
Compressed Complete Call Graph

- Redundancy in program executions and event traces:
  - Temporal repetitions due to iteration (and recursion)
  - Spatial repetitions due to SPMD parallelism



- Capture identical and similar repetitions – define “similar”
- Continuous compression, without large temporary data

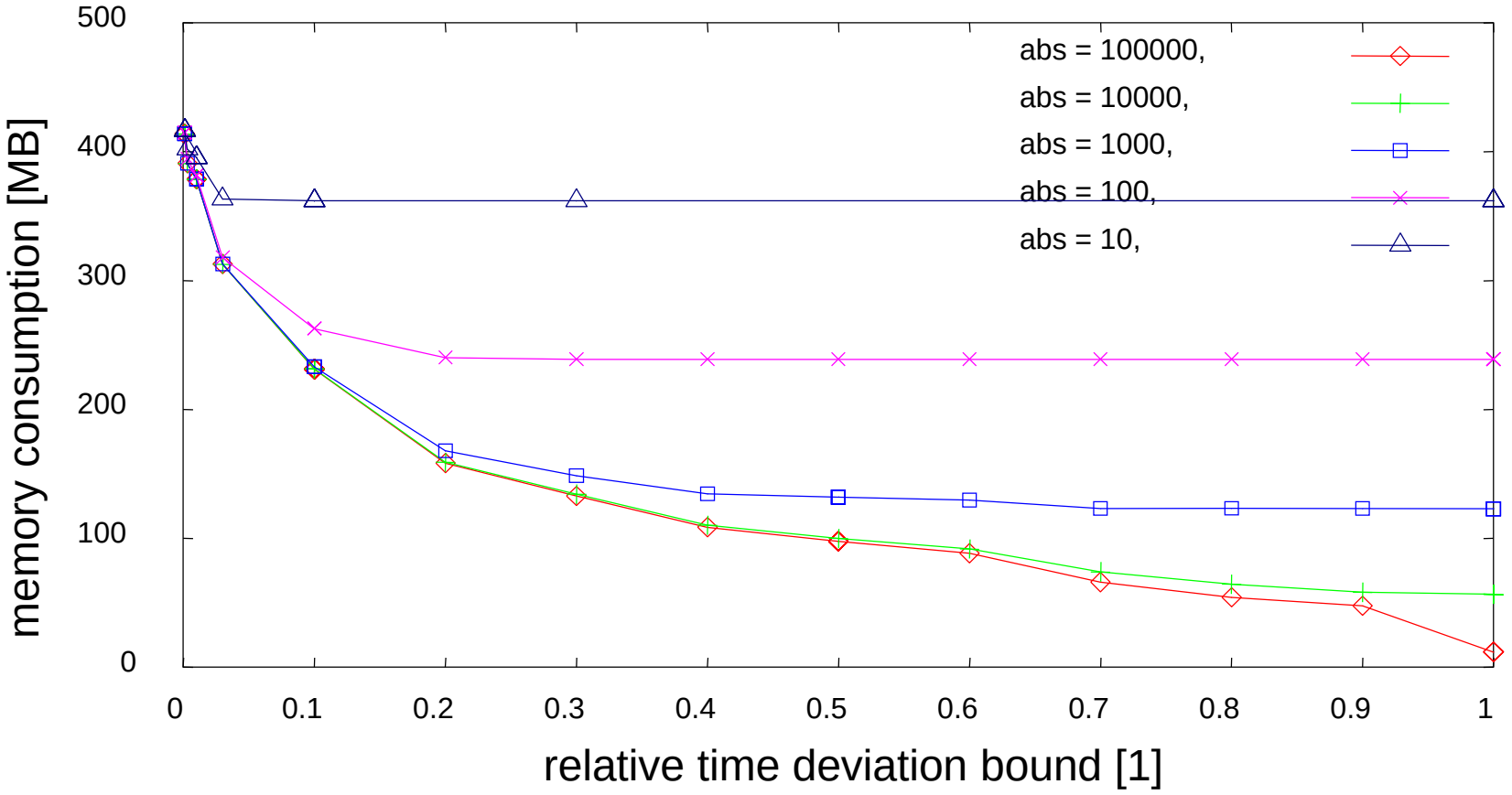
C3G Principle



Generalized tree data structure: Compressed Complete Call Graph

Compressed Complete Call Graph

- Define accuracy bounds for timing (absolute, relative)
- Trade timing accuracy for compression
- No explicit decompression necessary



Sampling-based Event Trace Recording

Sampling and Event Trace Recording

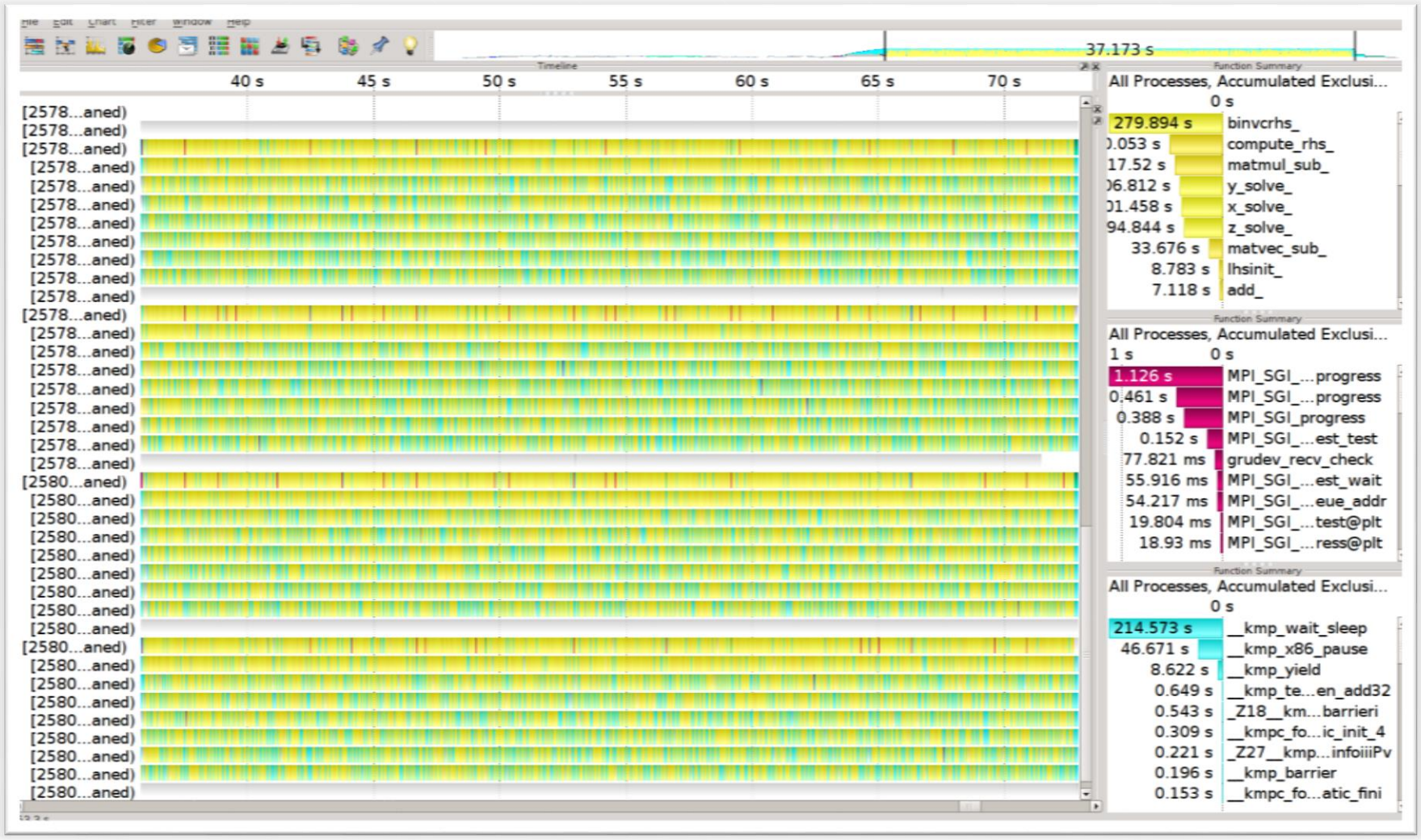
- Event tracing based on instrumentation
  - Indeed challenging data volumes sometimes
  - Filtering provides some control
- Sampling classically associated with profile generation
  - Call path profiles, phase profiles, …
- Now, record call paths of all samples over time
  - Proportional control over data size: runtime, sample frequency

Sampling based on Perf

- Linux perf infrastructure
  - Kernel support and user space tools, since kernel 2.6.31 (2009)
  - No recompilation or re-linking
  - Stack walk including libraries calls
  - In default HPC production environments, no root access needed

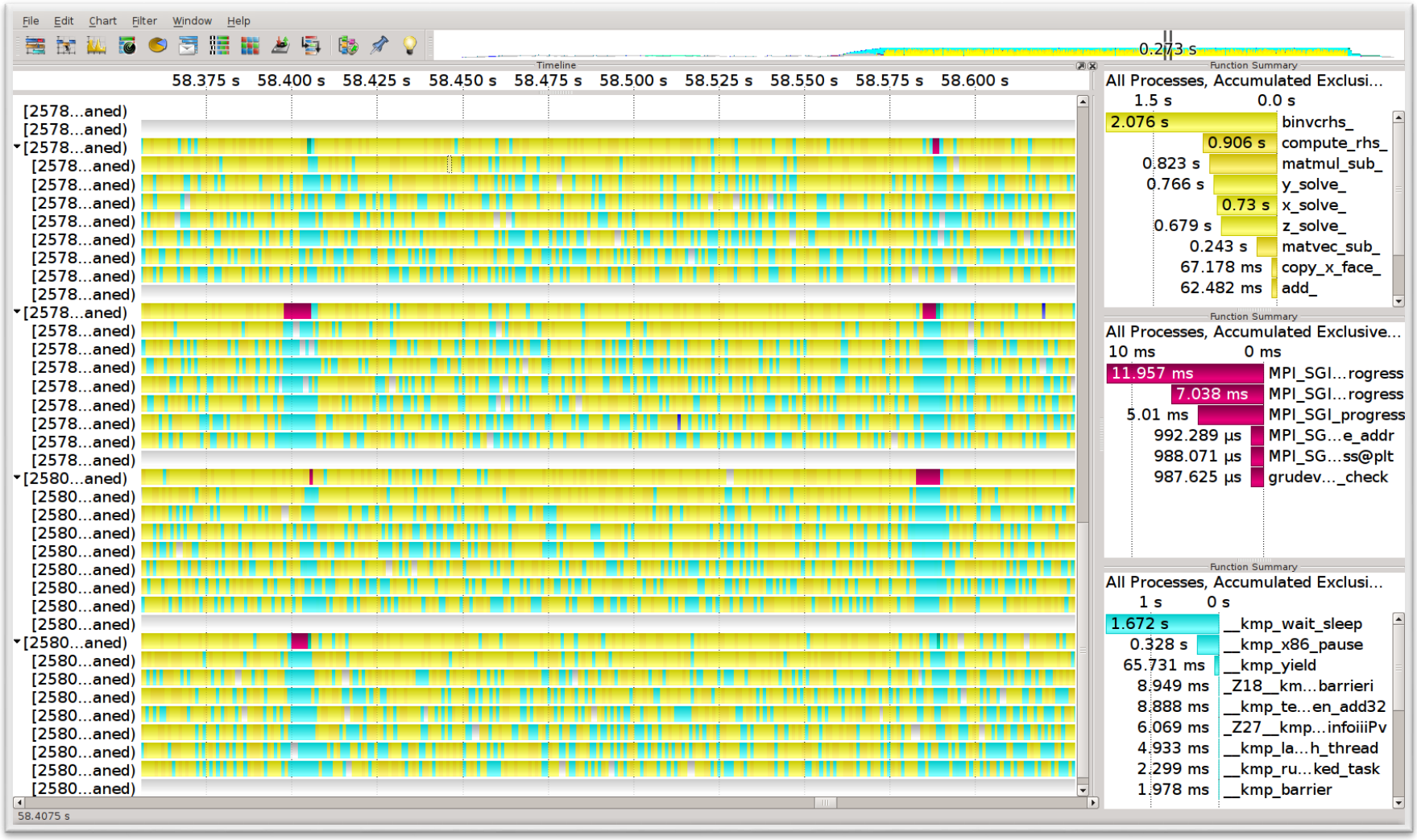
- Prototype implementation
  - Samples triggered by hardware counter (cycle counter)
  - Defined frequency (1kHz)
  - Convert perf recording data to event traces

Sampling instead of Instrumentation



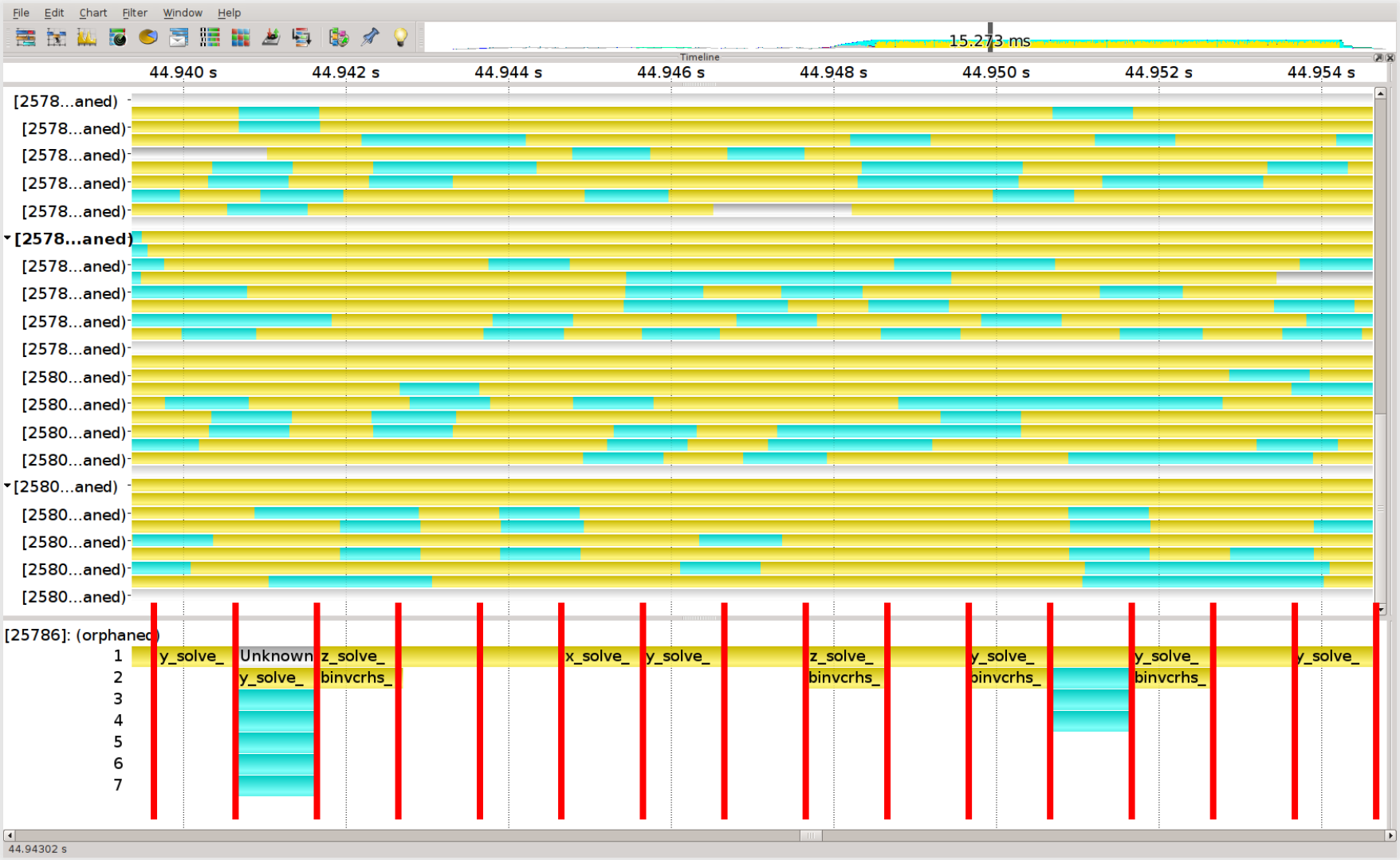
Sampling-based trace from NPB BT, MPI+OpenMP 4x8

Sampling instead of Instrumentation



NPB BT MPI+OpenMP 4x8, zoomed, with load imbalances

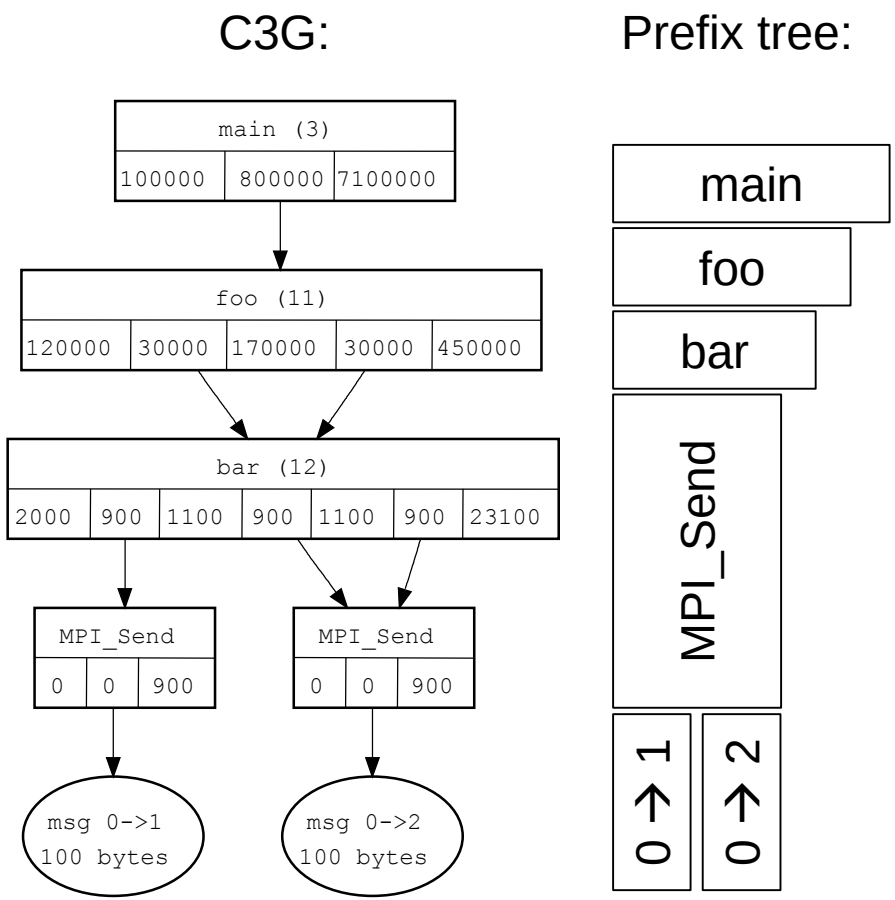
Sampling instead of Instrumentation



Fixed sampling interval

Combination

- Combine C3G and sampling-based trace recording
  - C3G paths vs. call stack samples
  - Similar representation in data structures and file formats
  - Amend with MPI information based on instrumentation
- Expected effects:
  - Sampling picks random call stack situations
  - Regular sampling intervals yield better compression
- Equivalent to prefix tree representation of call stack samples



Conclusion & Outlook

- C3G as Vampir's storage engine
  - Still fixed percentage of resources for performance analysis
  - Yet lower percentage due to removal of redundant parts
  - Implementation in product version underway
- Sampling-based trace recording in Score-P
  - Truly proportional data volumes over time
  - Still sufficient resolution for typical performance flaws
  - Planned for 2014
- Combine the "two roads" again
  - Added value: regular timing due to constant sampling interval leads to better and faster compression