VampirTrace and Vampir

Andreas Knüpfer & Holger Brunst
{andreas.knuepfer|holger.brunst}@tu-dresden.de
March 5th 2008

Overview

- Introduction
- Event Trace Visualization
- Vampir & VampirServer
- Vampir Displays
  - Timeline
  - Process Timeline with Performance Counters
  - Summary Display
  - Message Statistics
- VampirTrace: Instrumentation & Run-Time Measurement
- Conclusions

Why bother with performance analysis?

- Well, why are you here after all?
- efficient usage of expensive and limited resources
- scalability to achieve next bigger simulation

Profiling and Tracing

- have an optimization phase
  - just like testing and debugging phase
- use tools!
- avoid *do-it-yourself-with-printf* solutions, really!

Program Instrumentation

- detect run-time events (points of interest)
- pass information to run-time measurement library

Profile Recording

- collect aggregated information (Time, Counts, …)
- about program and system entities
  - functions, loops, basic blocks
  - application, processes, threads, …

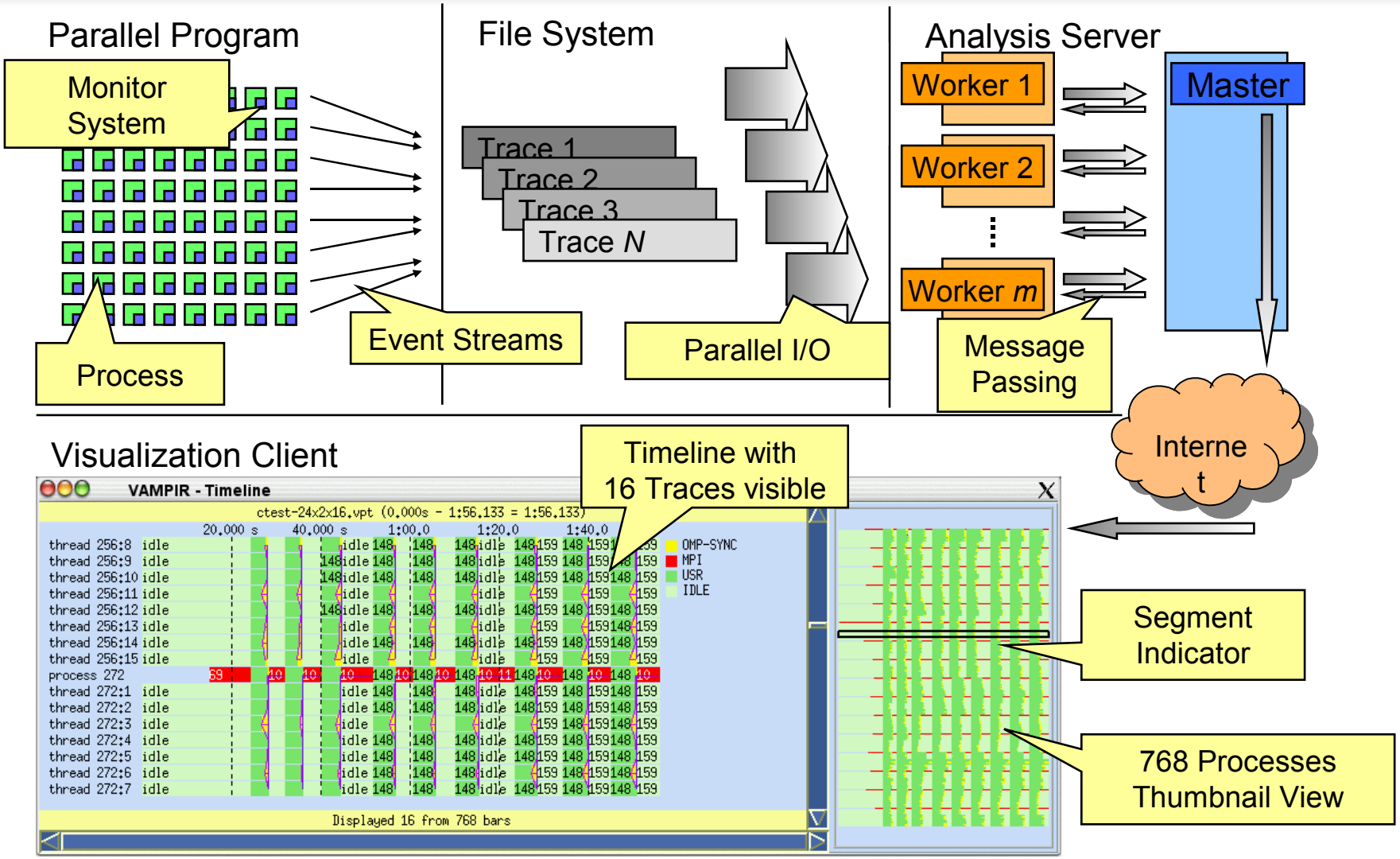
Trace Recording

- save individual event records
- with precise timestamp, process/thread ID, event type
- plus event specific information

Trace Visualization

- visualize trace data
- alternative + supplement to automatic analysis
- show dynamic run-time behavior graphically
- provide adaptive statistics and performance metrics
  - Timeline
  - Process Timeline, plus Performance Counters
  - Statistic Summary Display
  - Message Statistics
  - more
- interactive browsing, zooming, selecting

Vampir & VampirServer

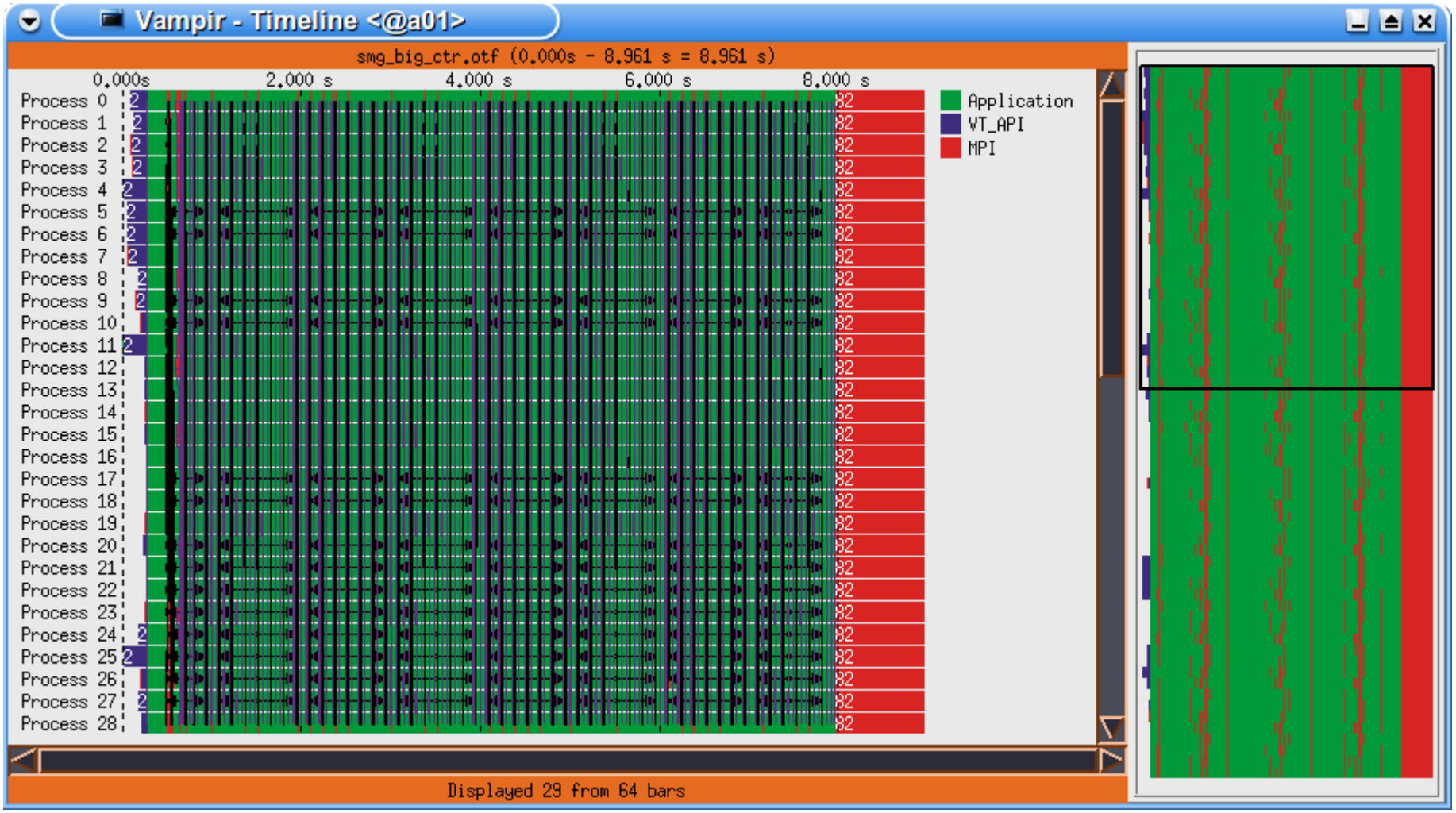


Vampir Displays

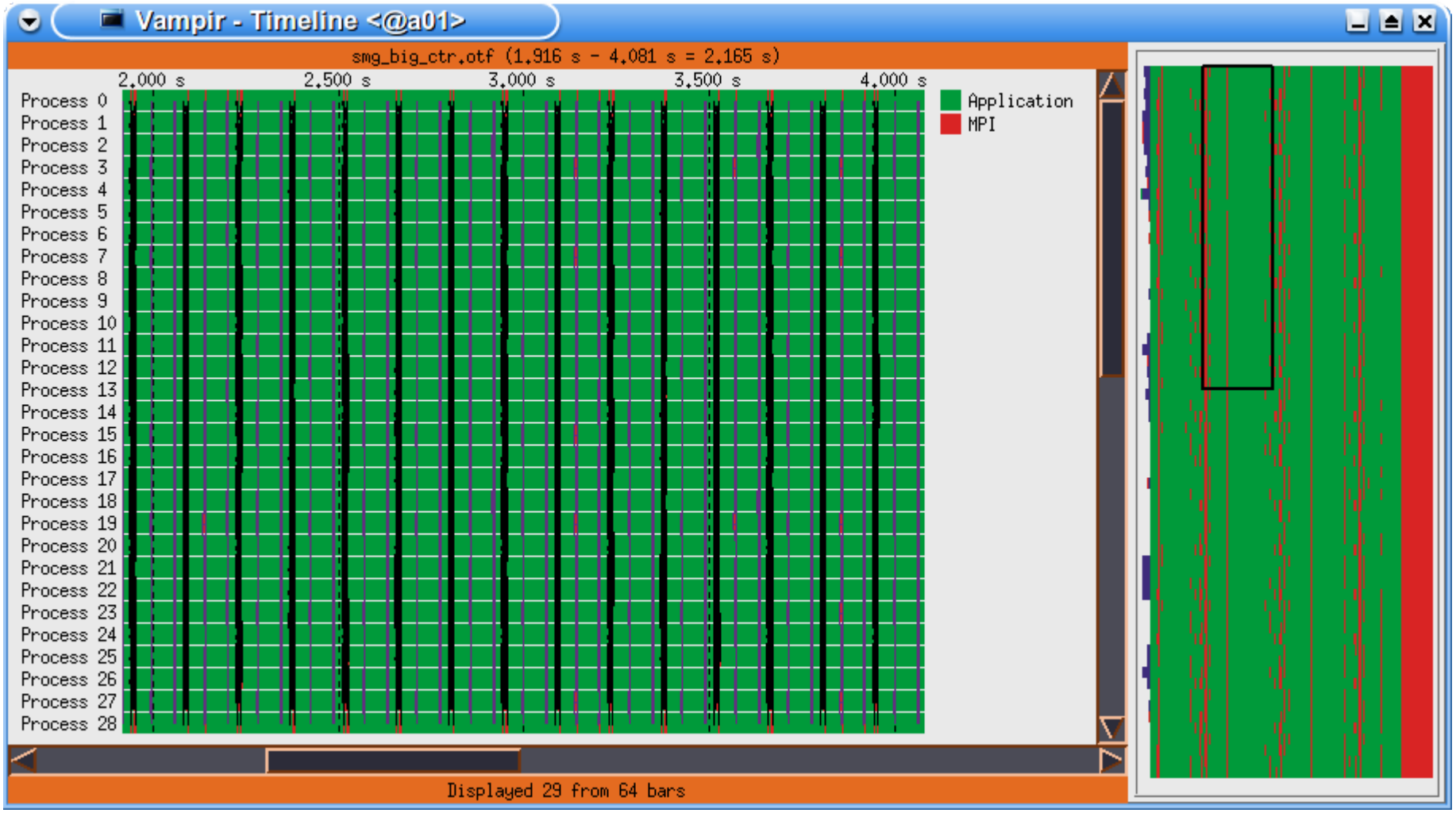
- Timeline
- Process Timeline
  - plus Performance Counters
- Statistic Summary
- Message Statistics

- more …

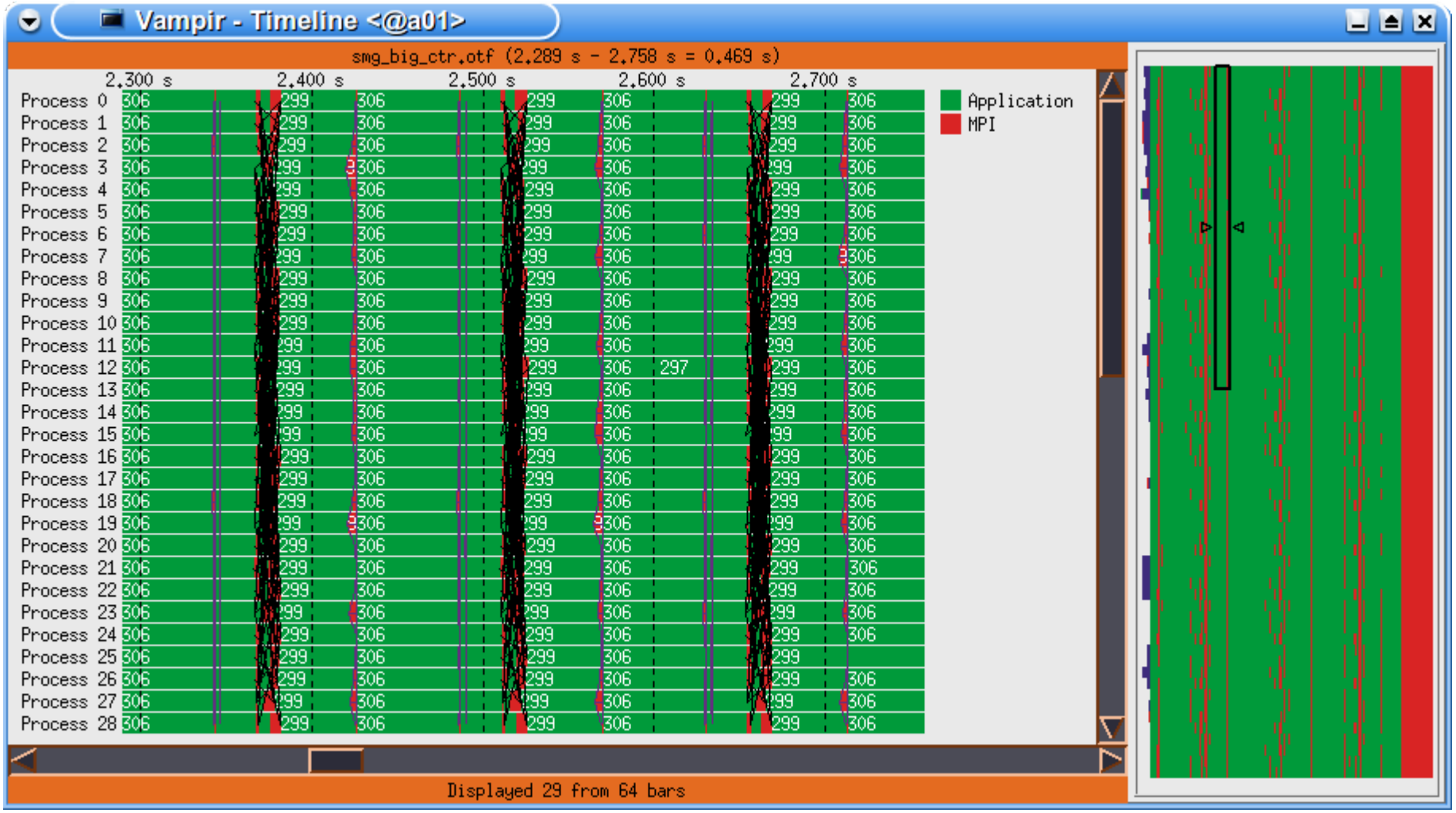
Timeline Display



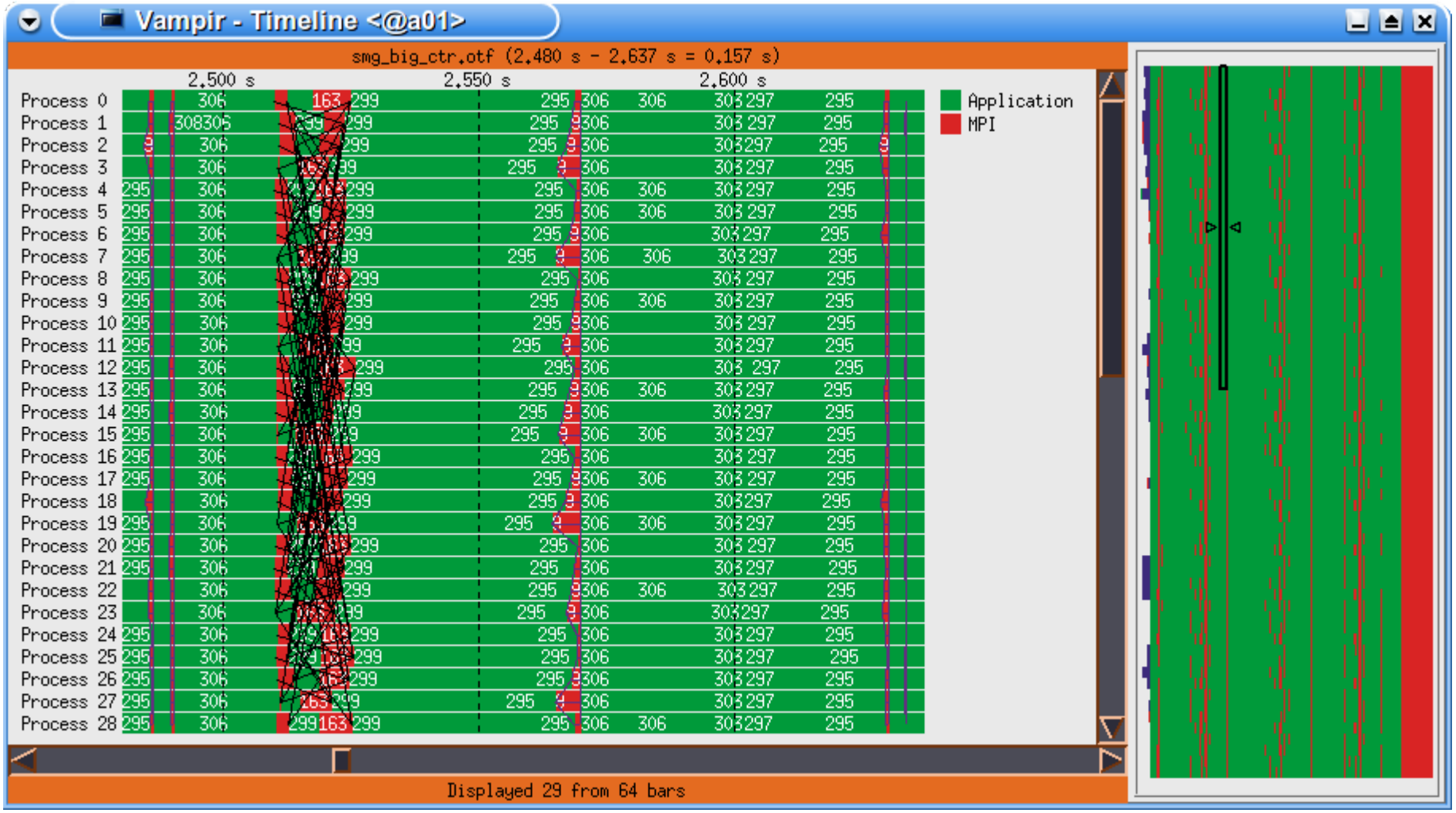
Timeline Display



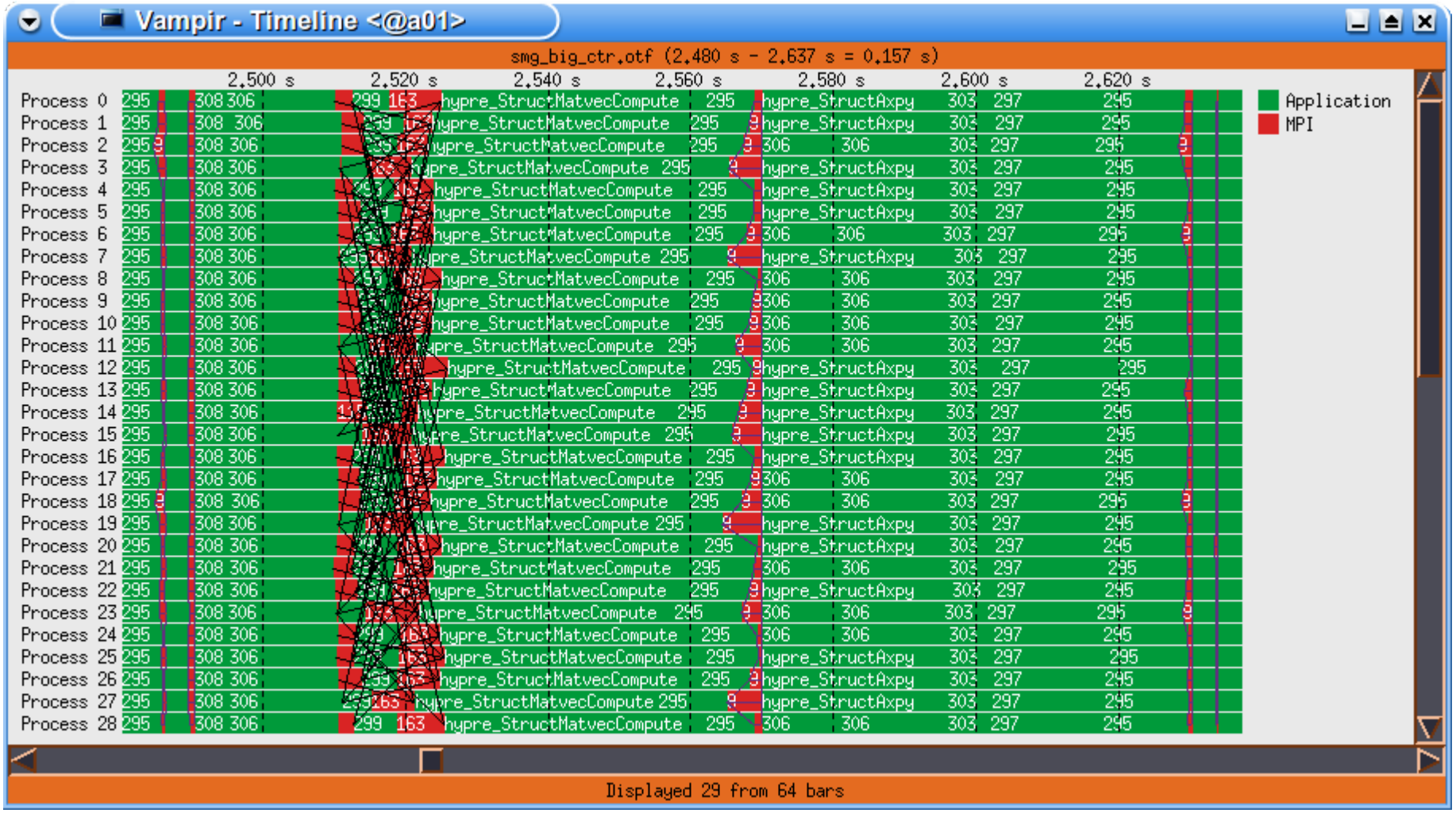
Timeline Display



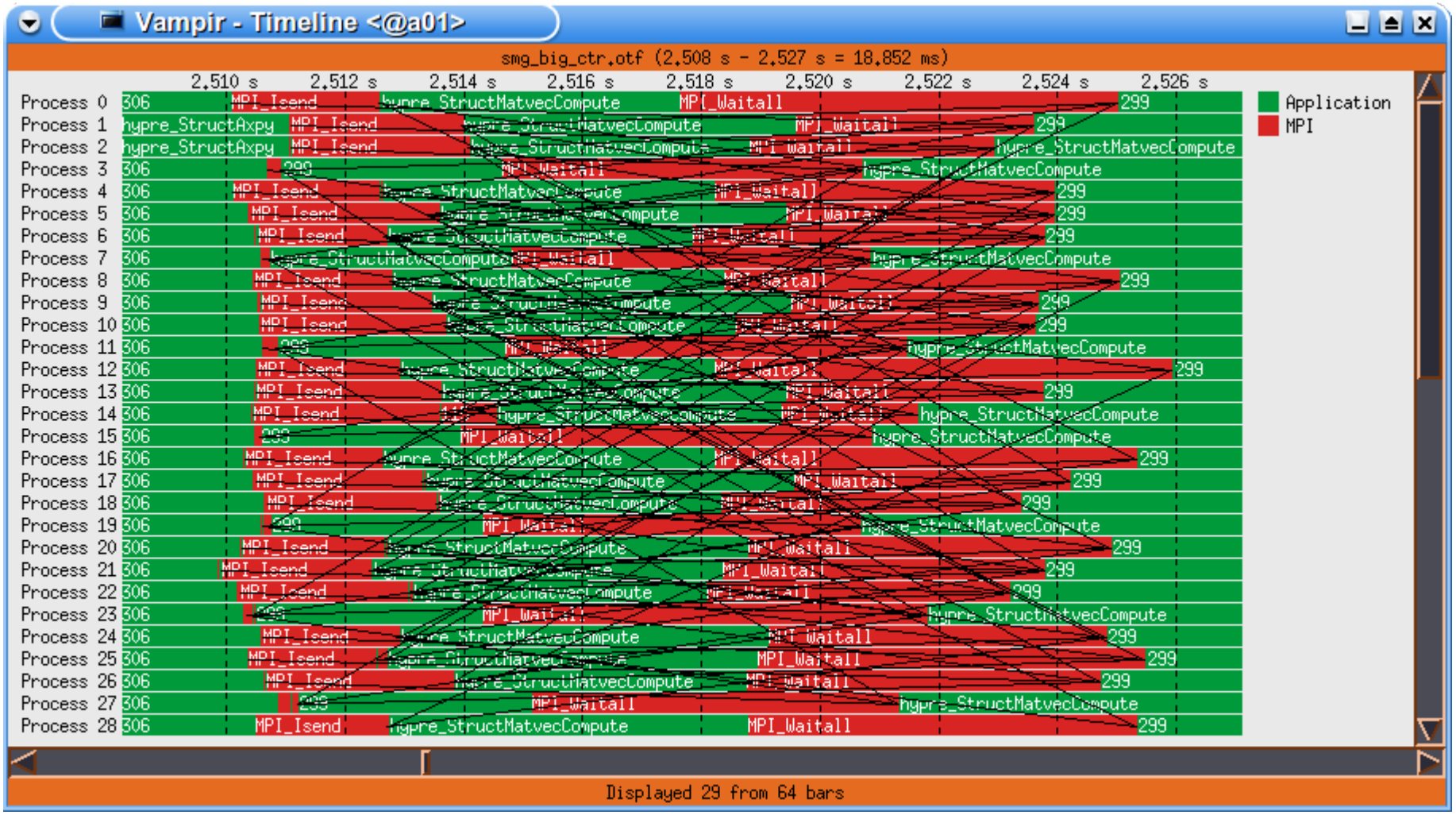
Timeline Display



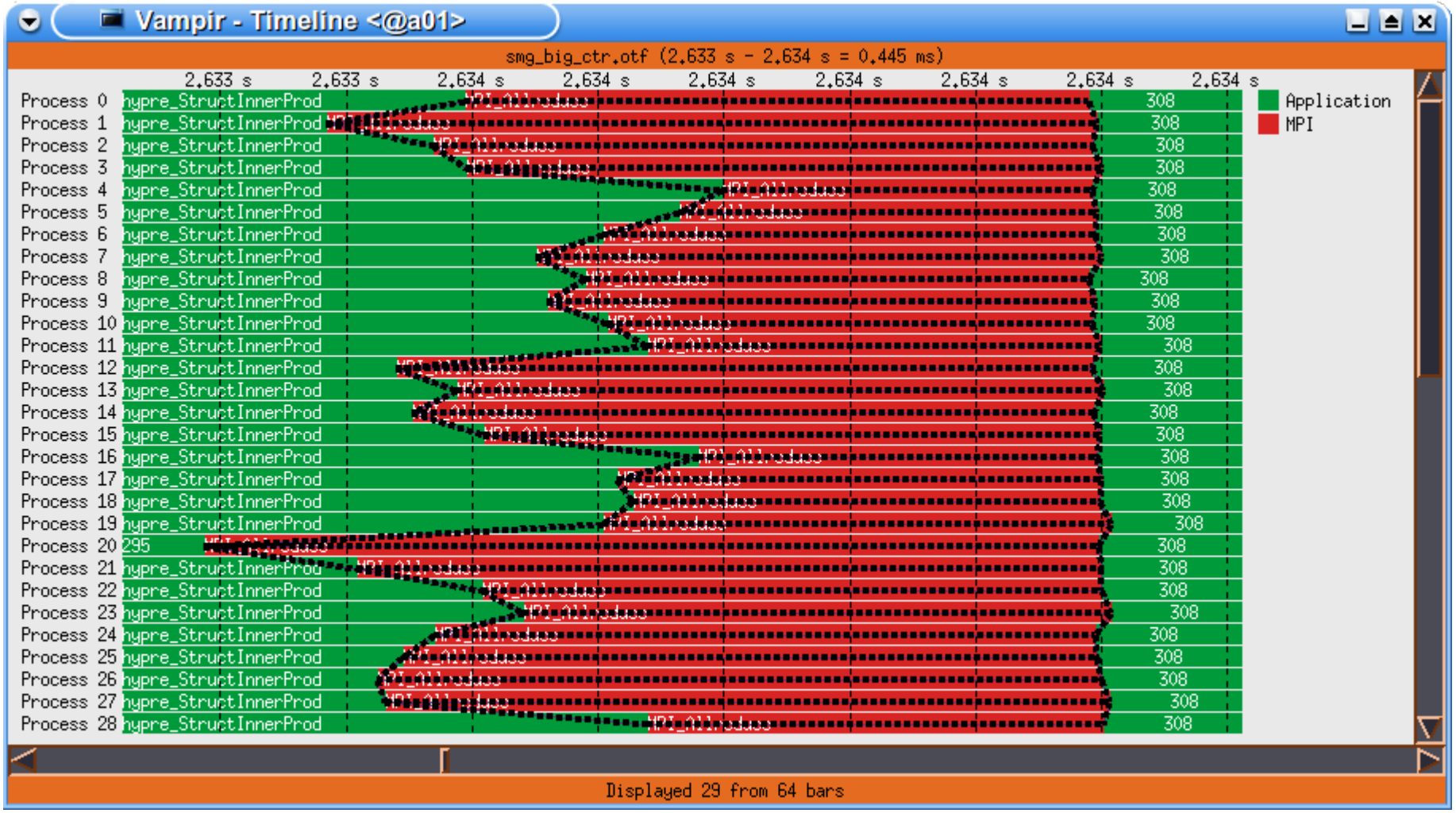
Timeline Display



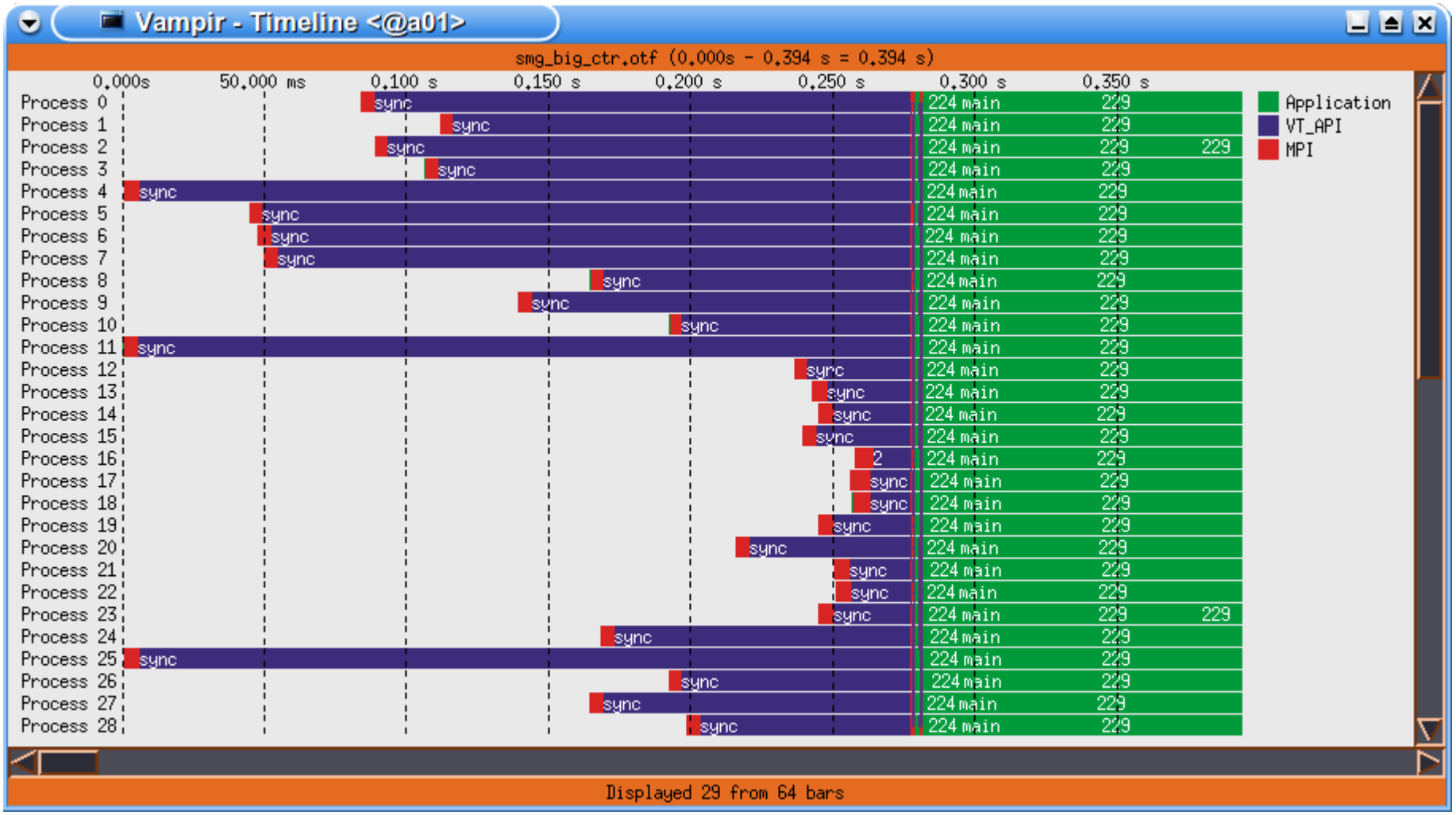
Timeline Display



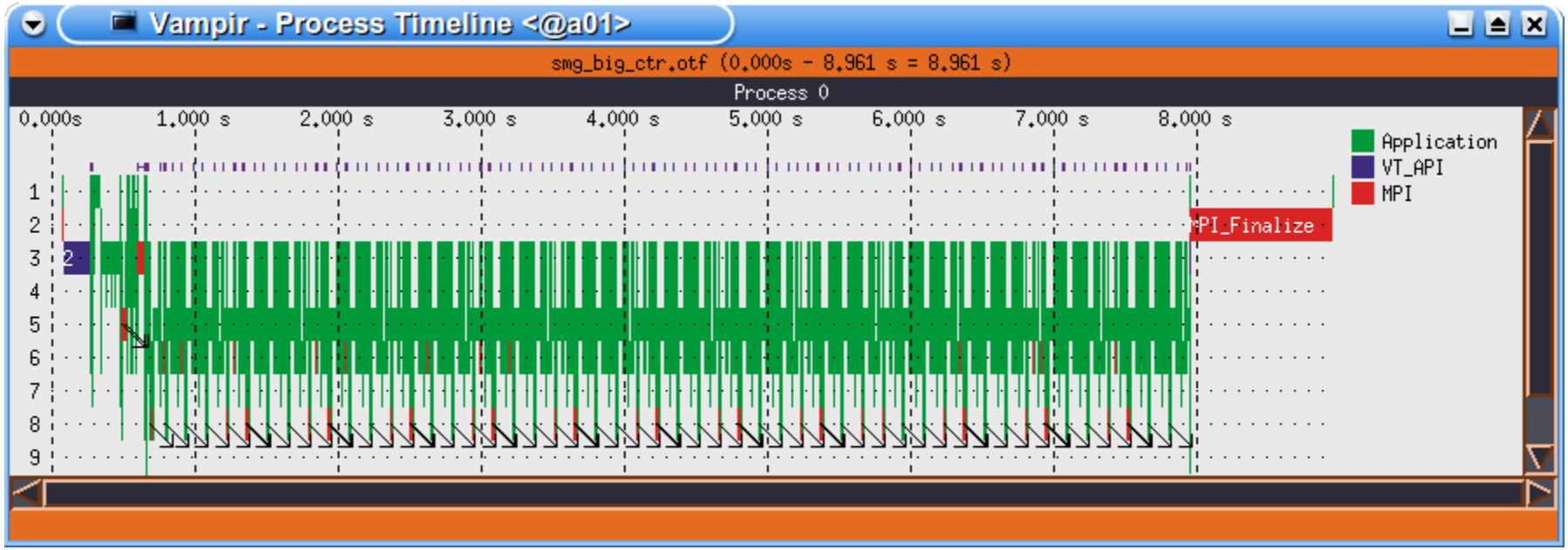
Timeline Display



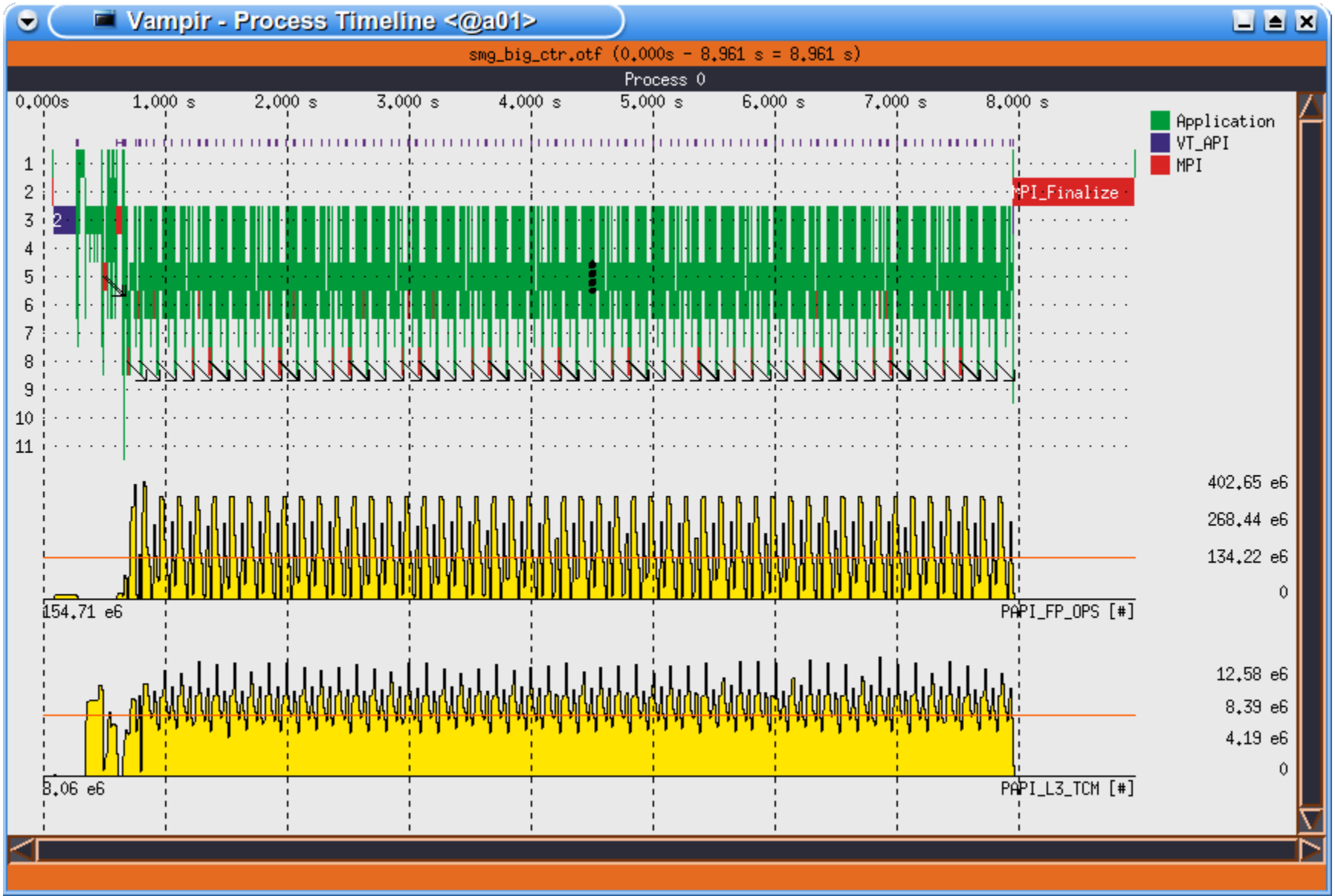
Timeline Display



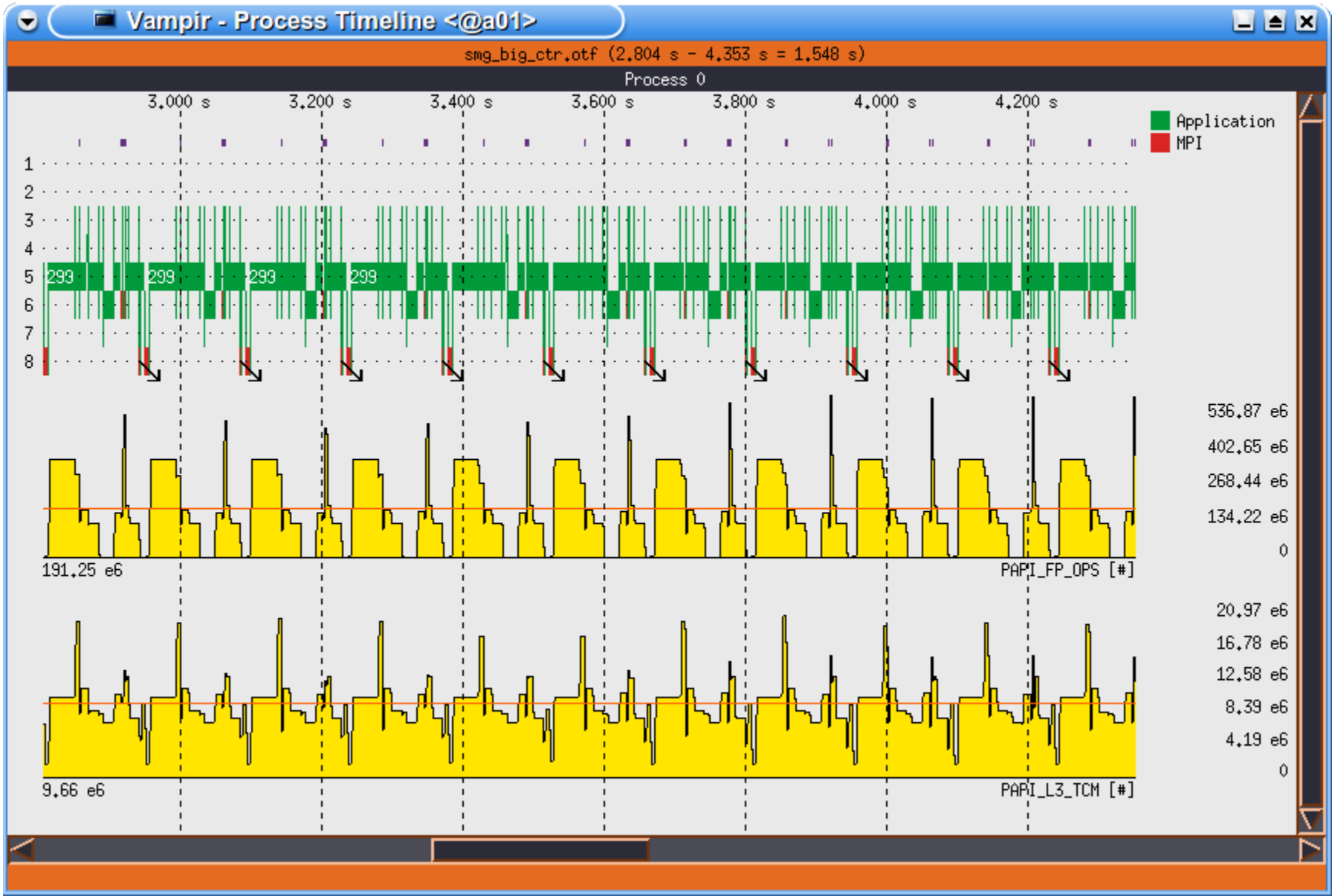
Process Timeline Display



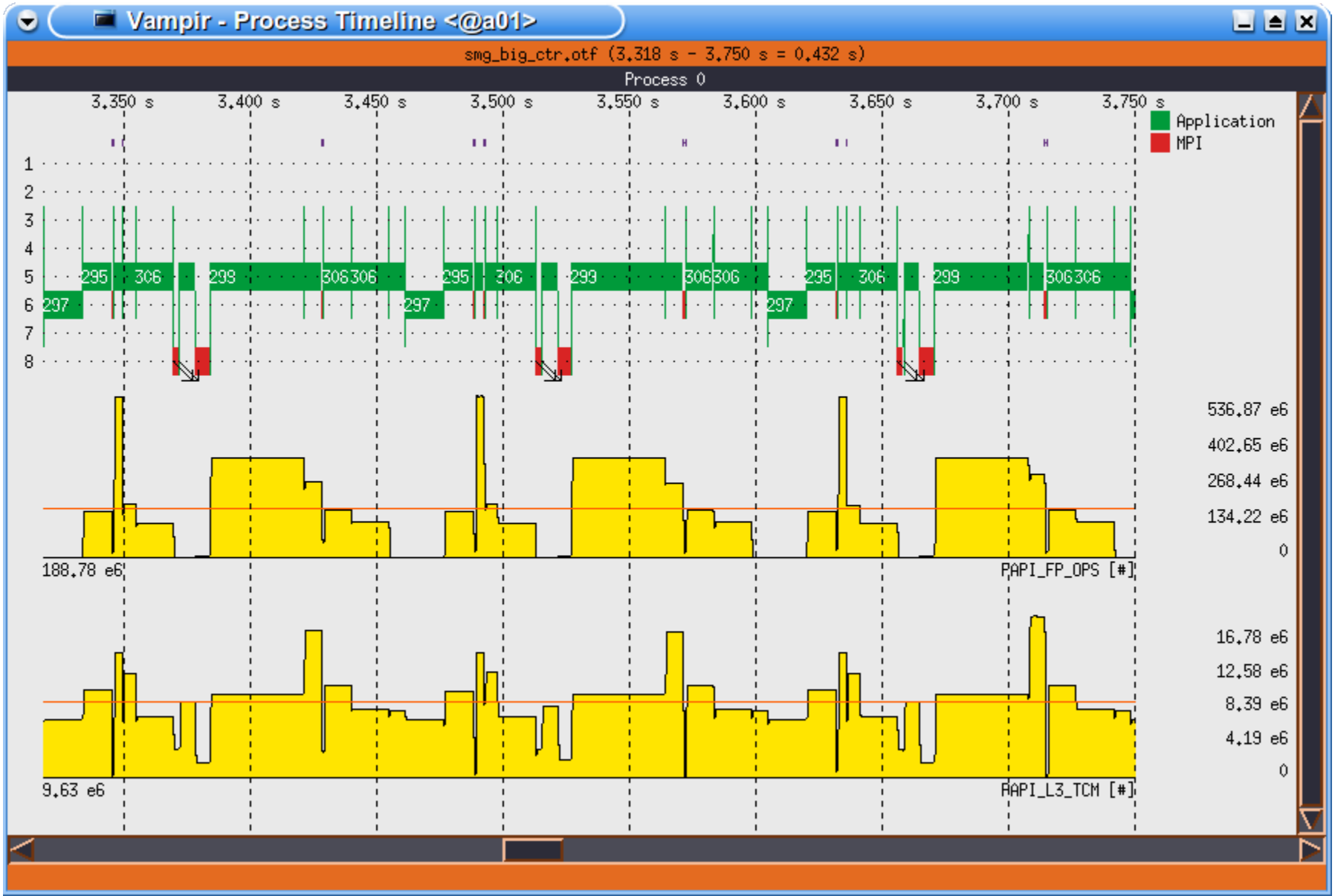
Process Timeline Display



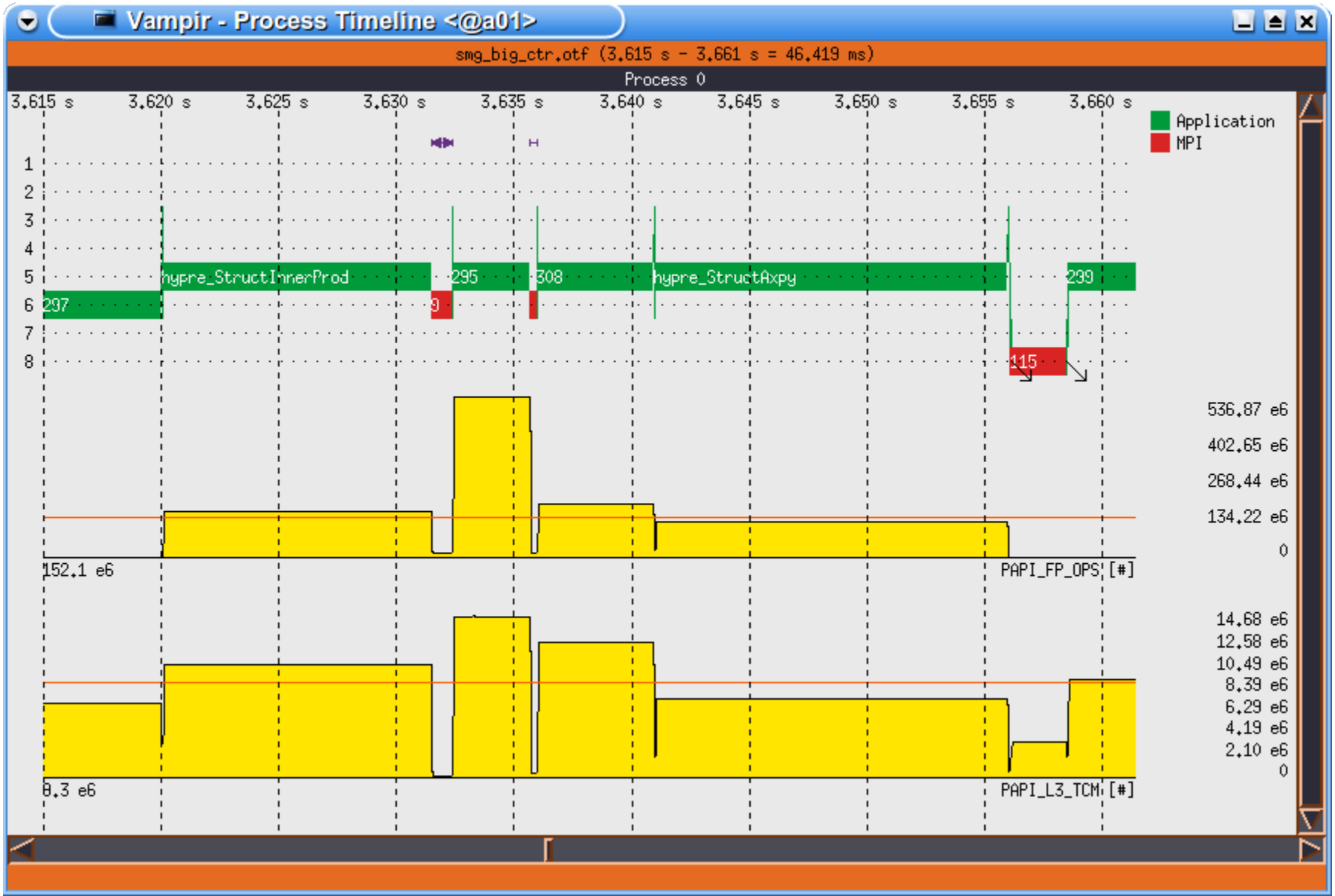
Process Timeline Display



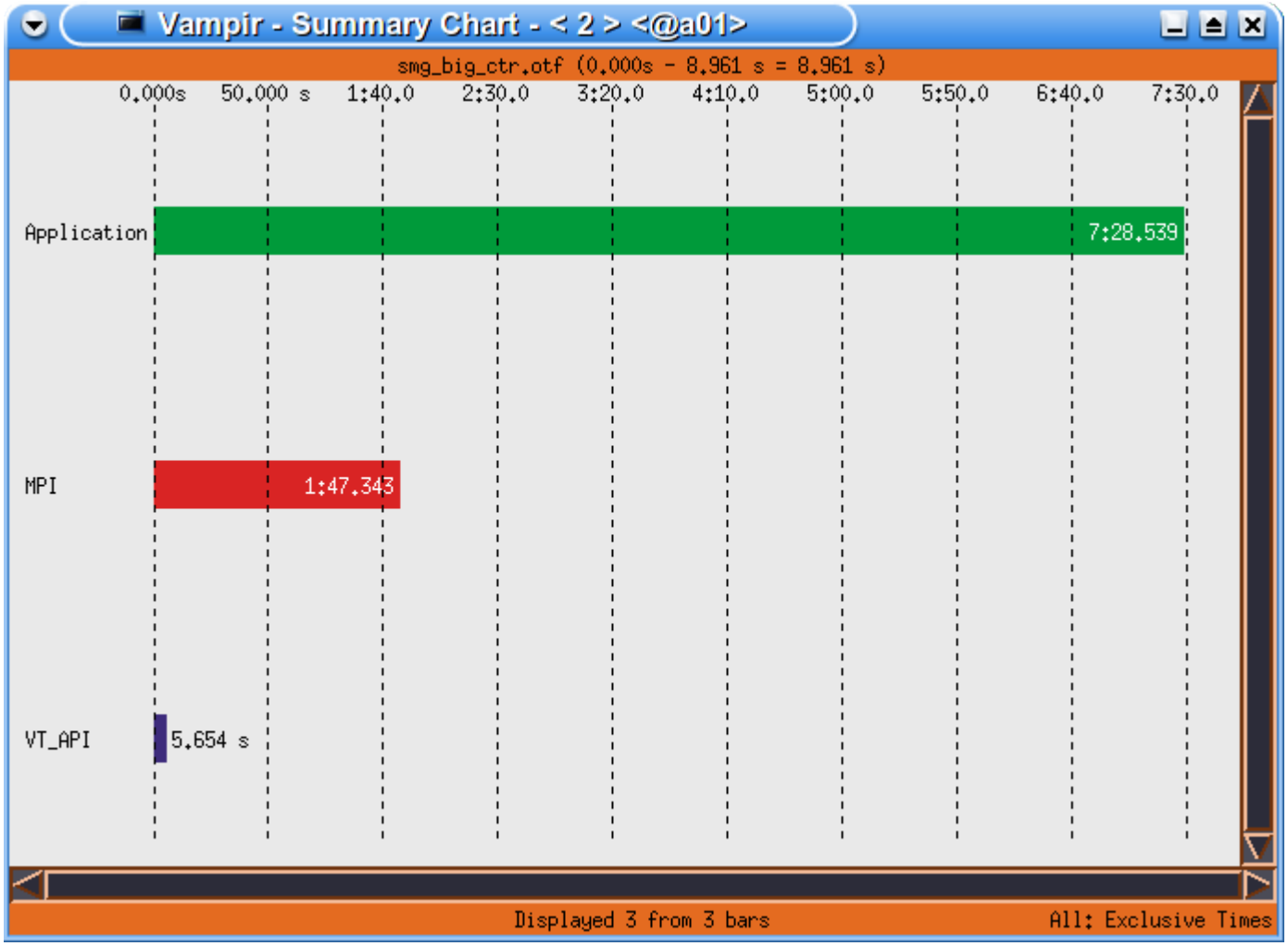
Process Timeline Display



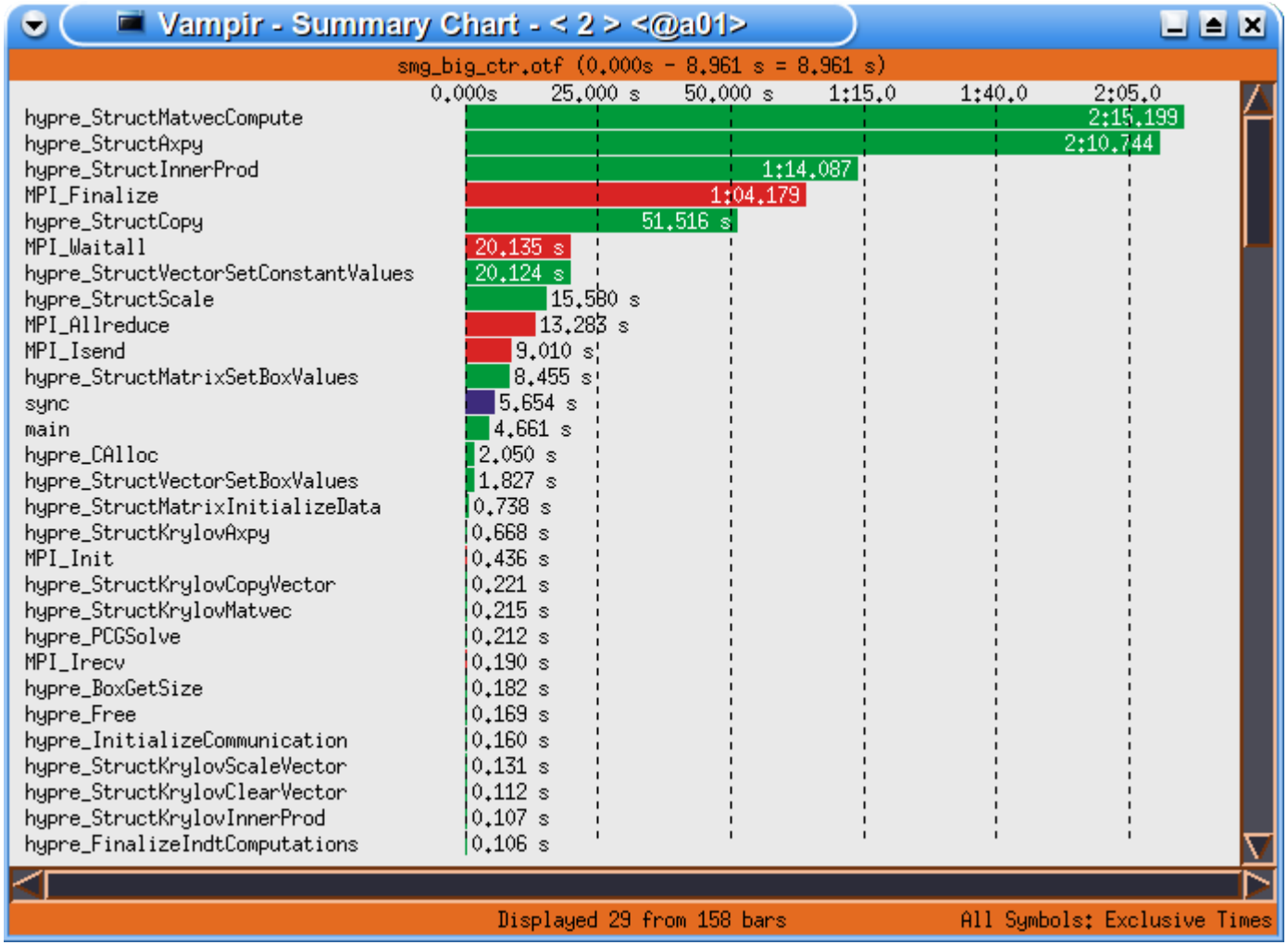
Process Timeline Display



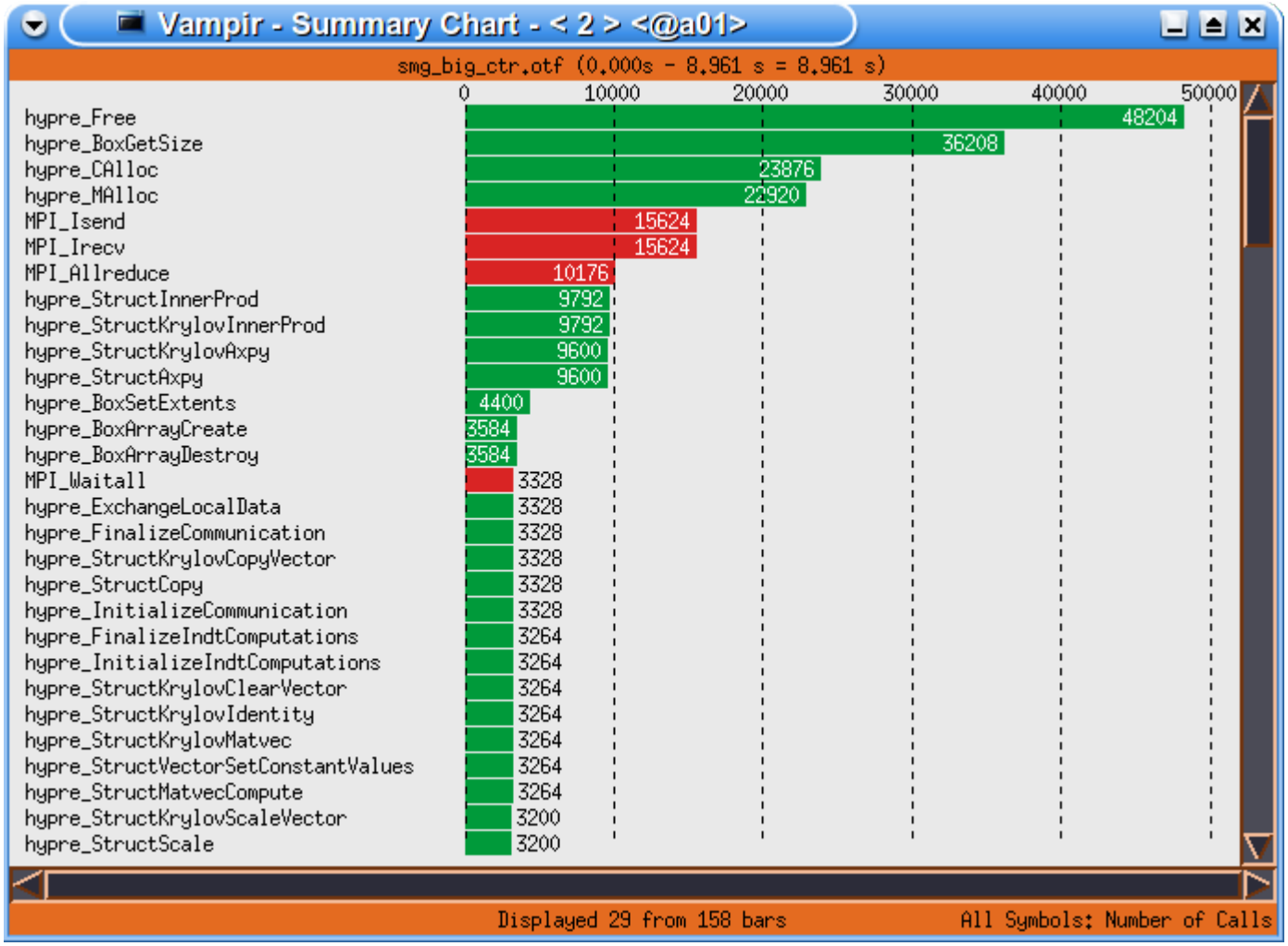
Statistic Summary Display



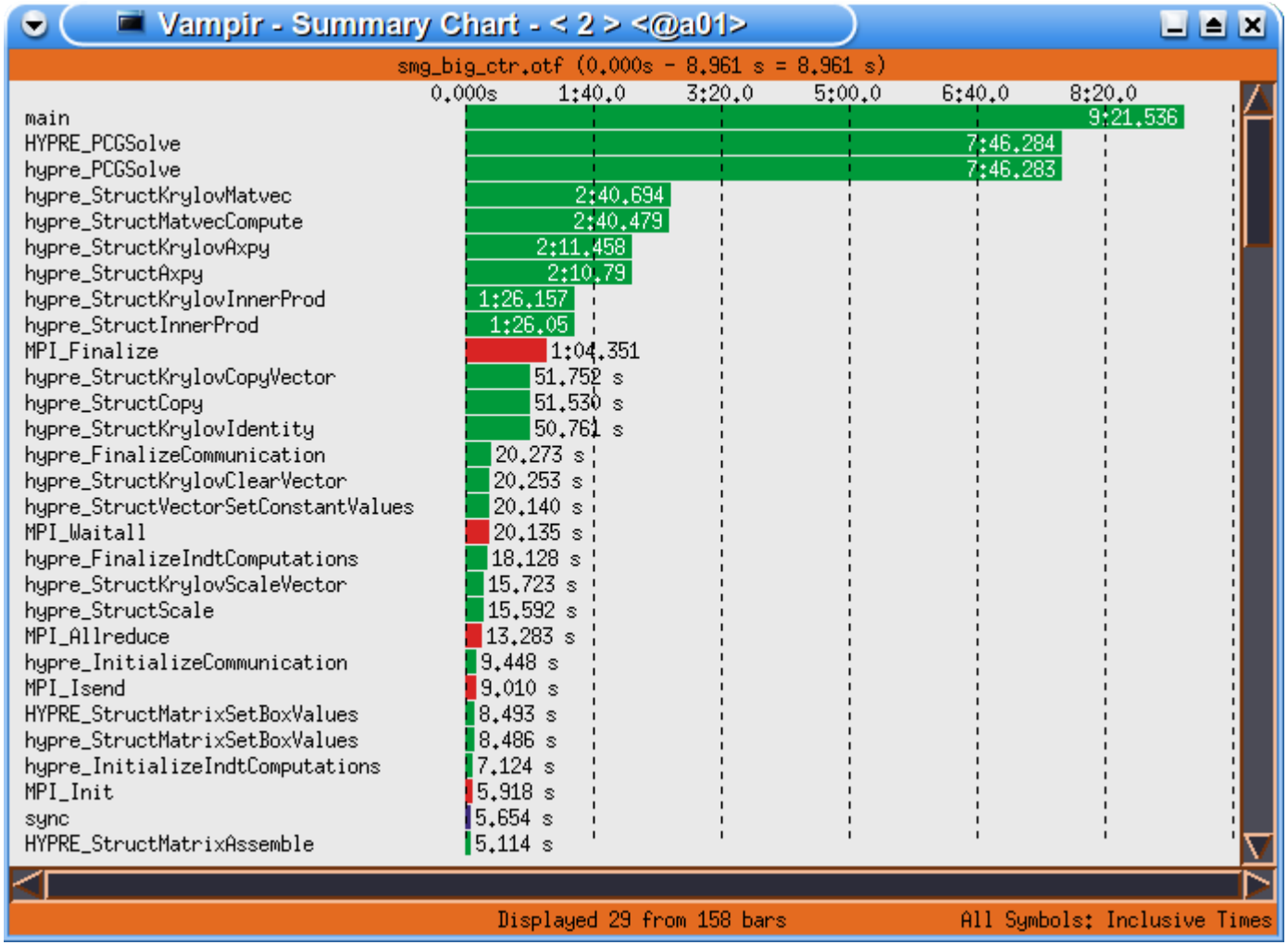
Statistic Summary Display



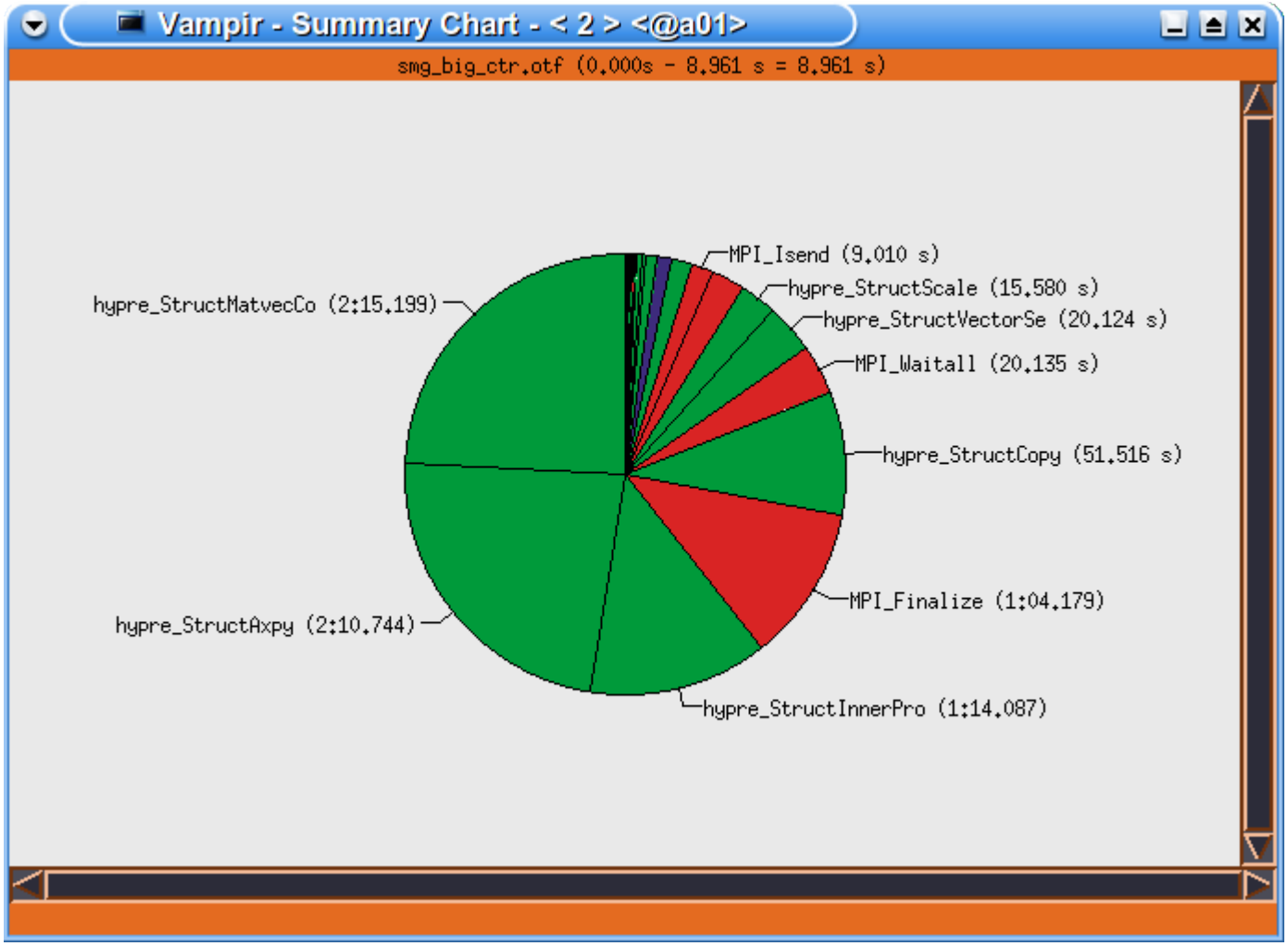
Statistic Summary Display



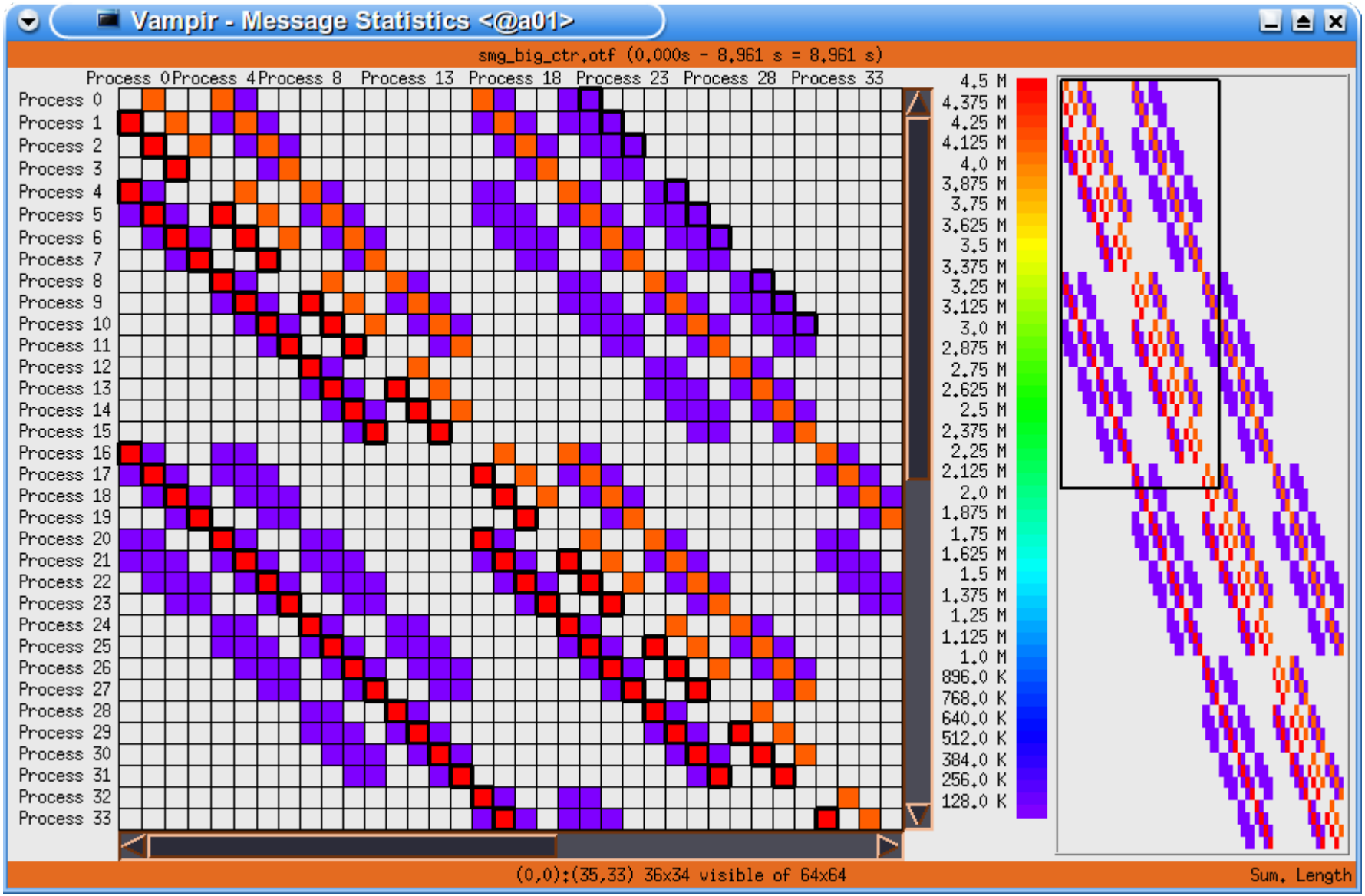
Statistic Summary Display



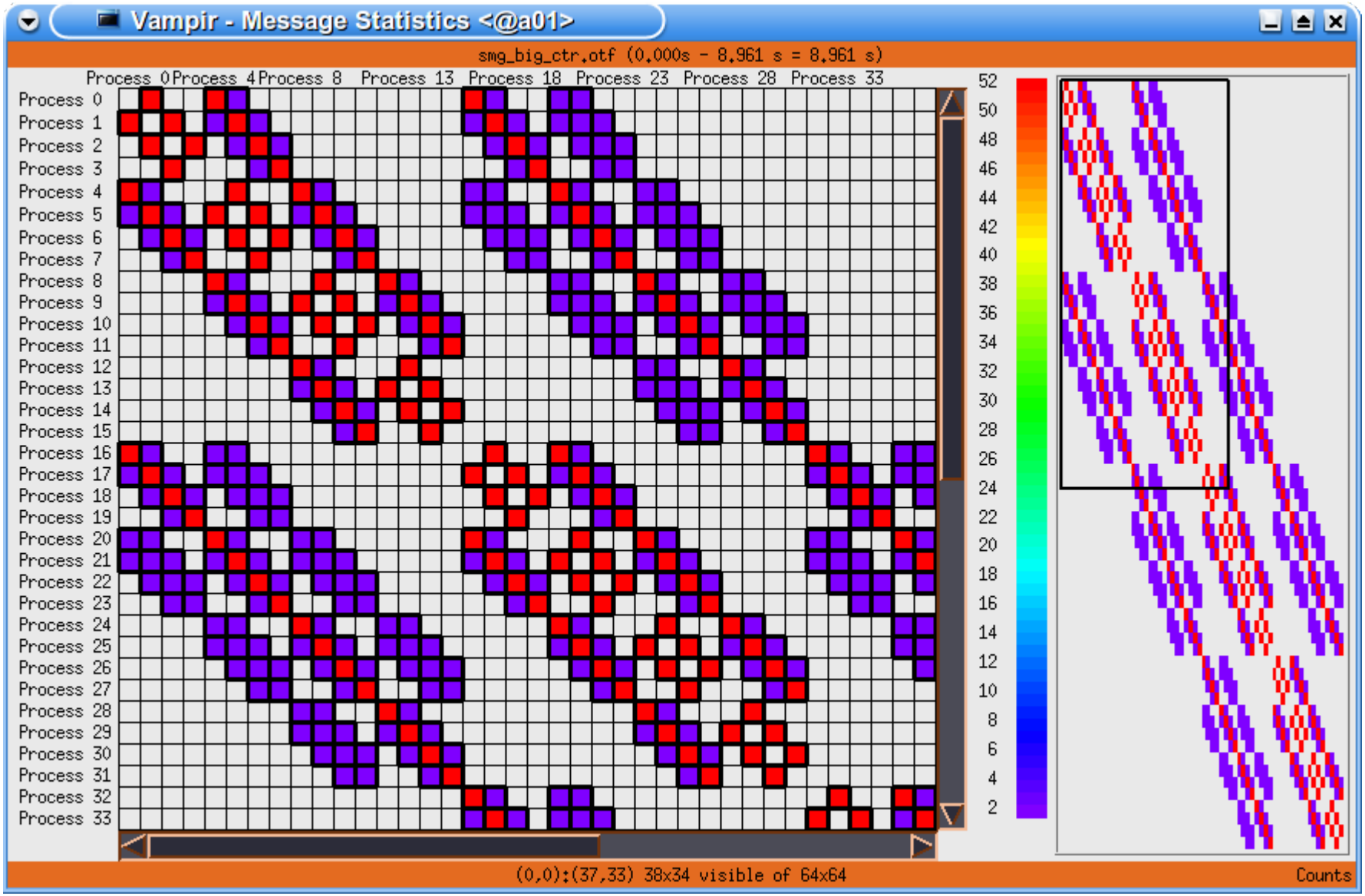
Statistic Summary Display



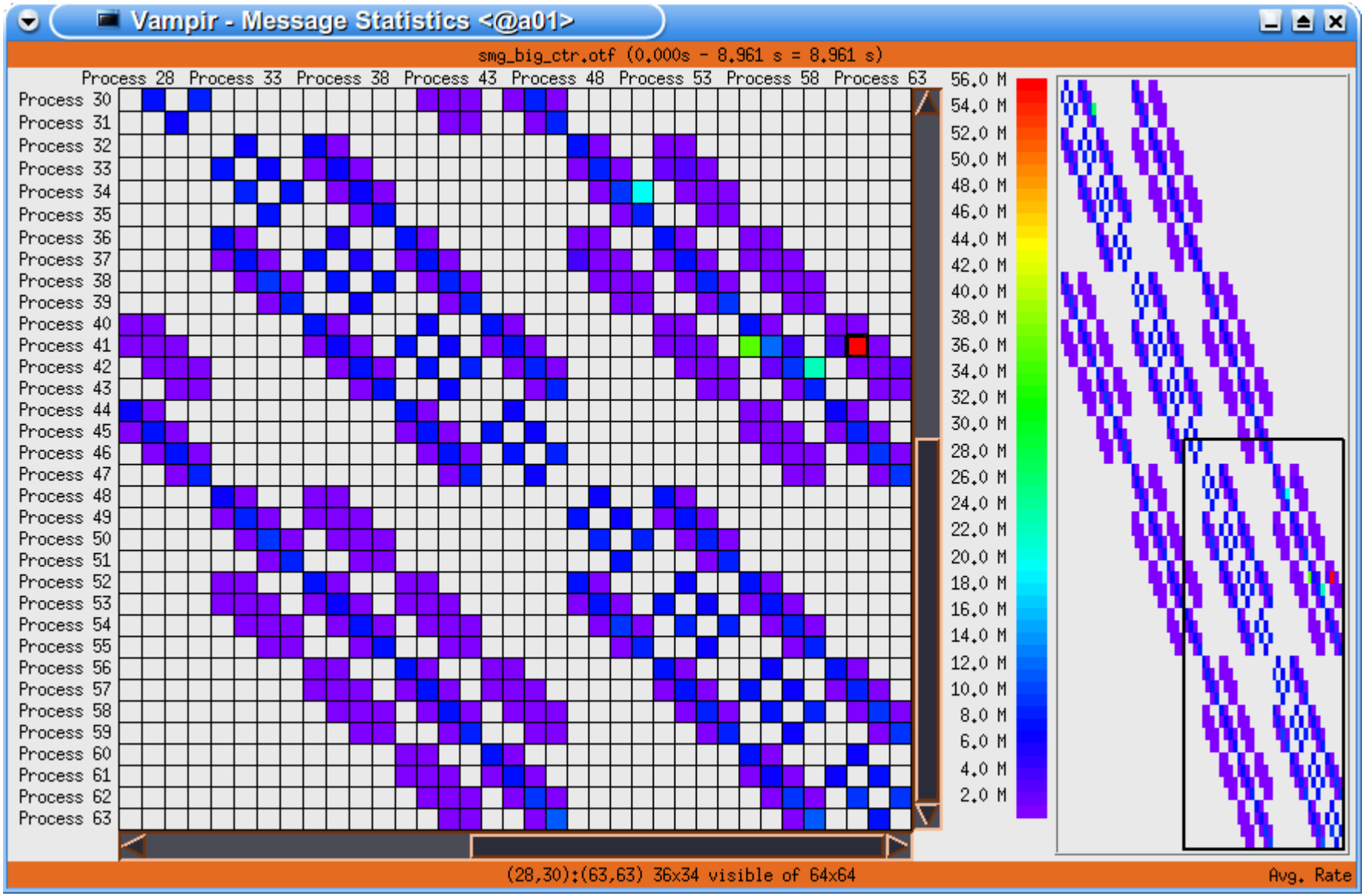
Message Statistics Display



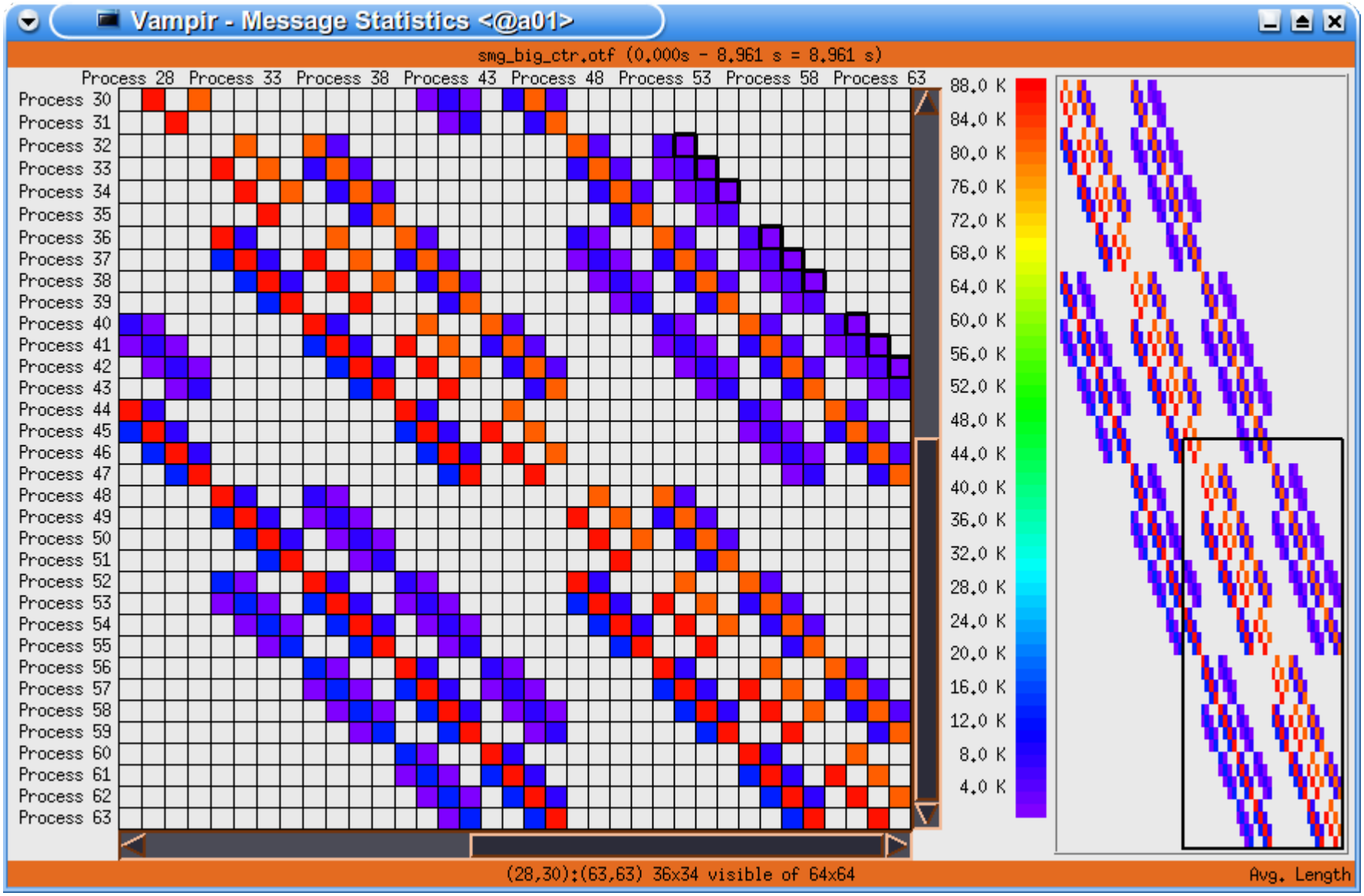
Message Statistics Display



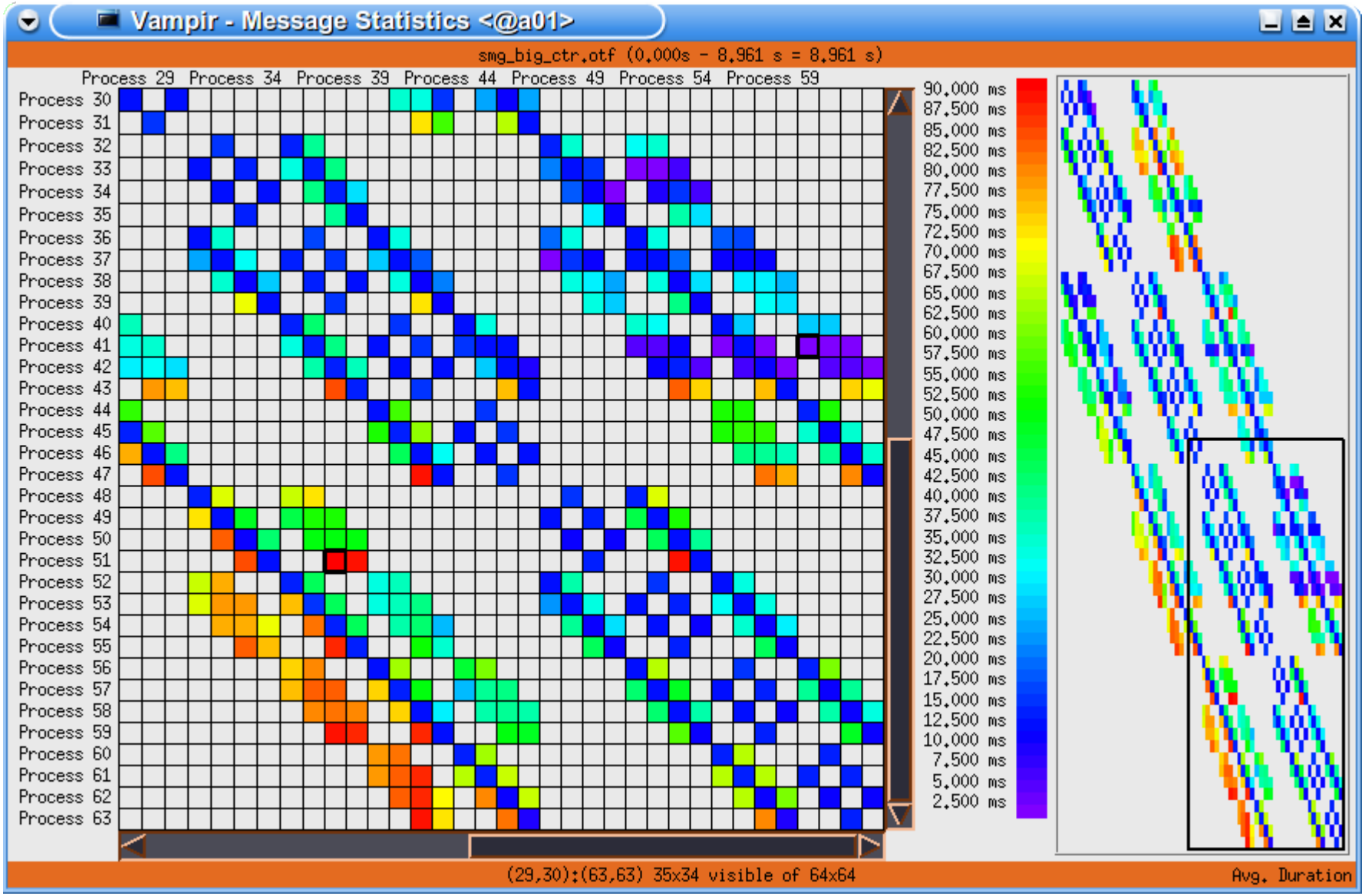
Message Statistics Display



Message Statistics Display



Message Statistics Display



Vampir Displays

- there are more displays
- plus special features for extended scalability

- there are more kinds of performance data

- see Holger's Vampir presentation

What do you need to do for it?

- Instrumentation
  - via compiler wrappers, specific for underlying compiler

```
icc hello.c -o hello

vtcc hello.c -o hello
```

```
mpicc hello.c -o hello

vtcc hello.c -o hello
```

```
CC=icc    CXX=icpc

F90=ifc   MPICC=mpicc
```

```
CC=vtcc    CXX=vtcxx

F90=vtf90  MPICC=vtcc
```

- Re-Compile & Re-Link
  - use compiler instrumentation, add measurement library
- Trace Run
  - run with appropriate test data set
- more details in my separate VampirTrace presentation

Conclusions

- VampirTrace
  - convenient instrumentation and measurement package
  - hides away complicated details
  - provides many options and switches for experts
- Vampir & VampirServer
  - interactive trace visualization and analysis
  - intuitive browsing and zooming
  - scalable to large trace data sizes (100GB)
  - scalable to many processes/threads in a trace (2000)