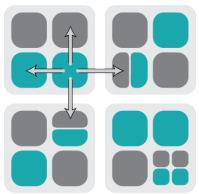
Performance Analysis and Projections

Ronak Buch

20 April 2016



Charm++

"We must not allow the clock and the calendar to blind us to the fact that each moment of life is a miracle and mystery."

H.G. Wells

Each moment of life may be a mystery, but each moment of our programs shouldn't be.

Measuring Performance

In the serial world:

- `clock_gettime()`, `gettimeofday()`
- `gprof`
- `valgrind`

Measuring Performance

In the serial world:

- `clock_gettime(), gettimeofday()`
- `gprof`
- `valgrind`

Parallel

These techniques don't really work for distributed parallel programs.

Problems in Parallel Analysis

- Different processors may do radically different work
- Performance problems may be on the network
- Load may not be balanced between processors

Parallel Analysis Tools

In general, tools must be used to do parallel performance analysis:

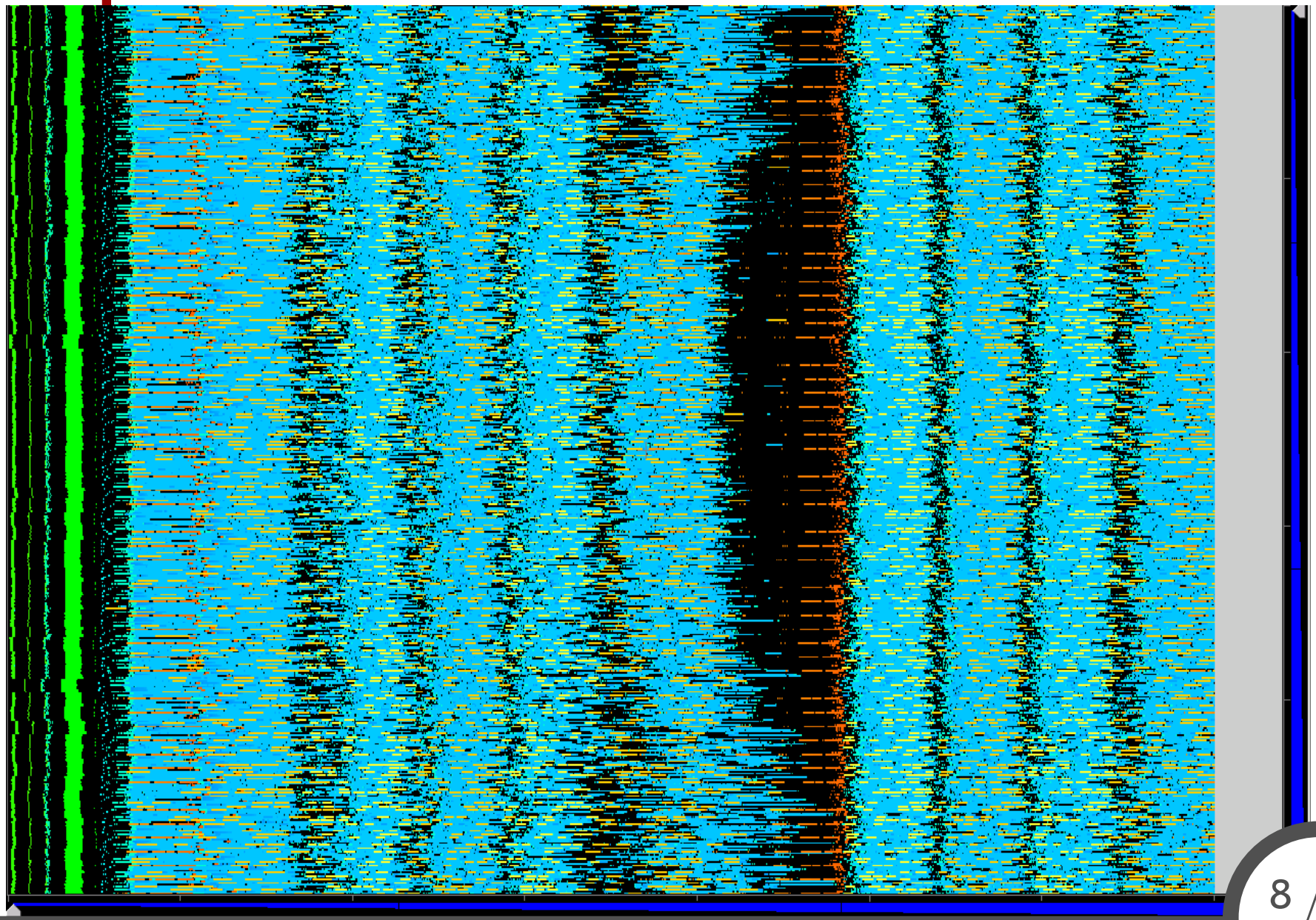
- mpiP
- Vampir
- HPCToolkit
- Projections

These tools provide details on communication, scaling, mapping, load imbalance, etc.

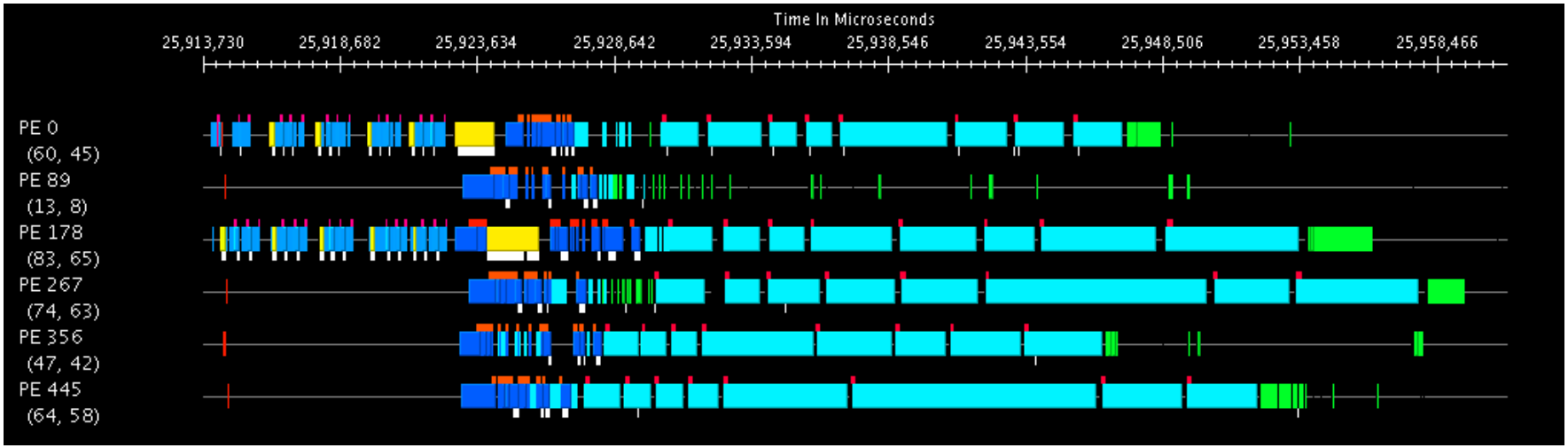
Projections

- Projections is a performance analysis tool for Charm++
- Provides tools for analyzing most of the discussed issues in parallel performance
- Runtime traces application executions, creates logs for local analysis

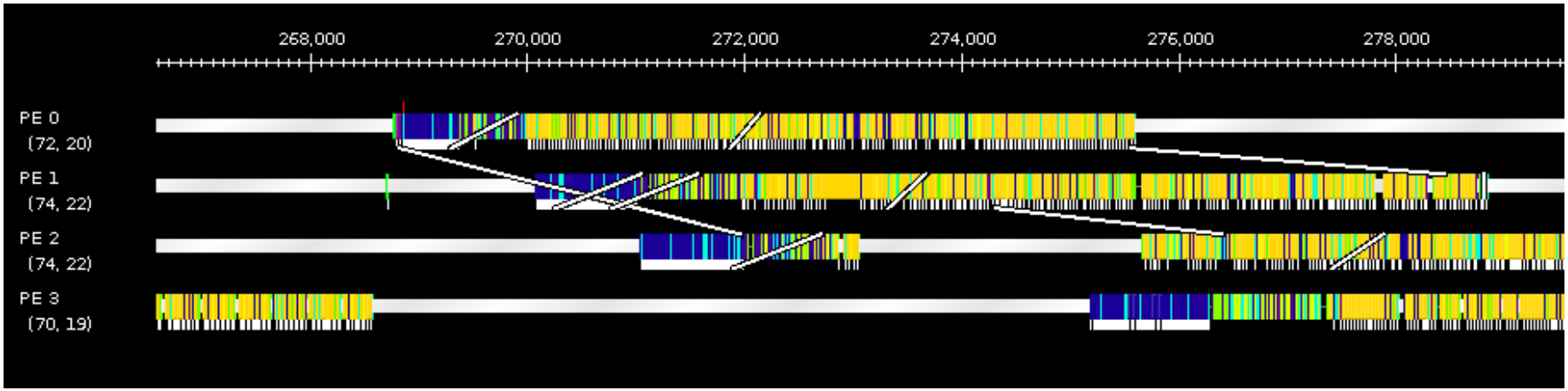
Projections Tools



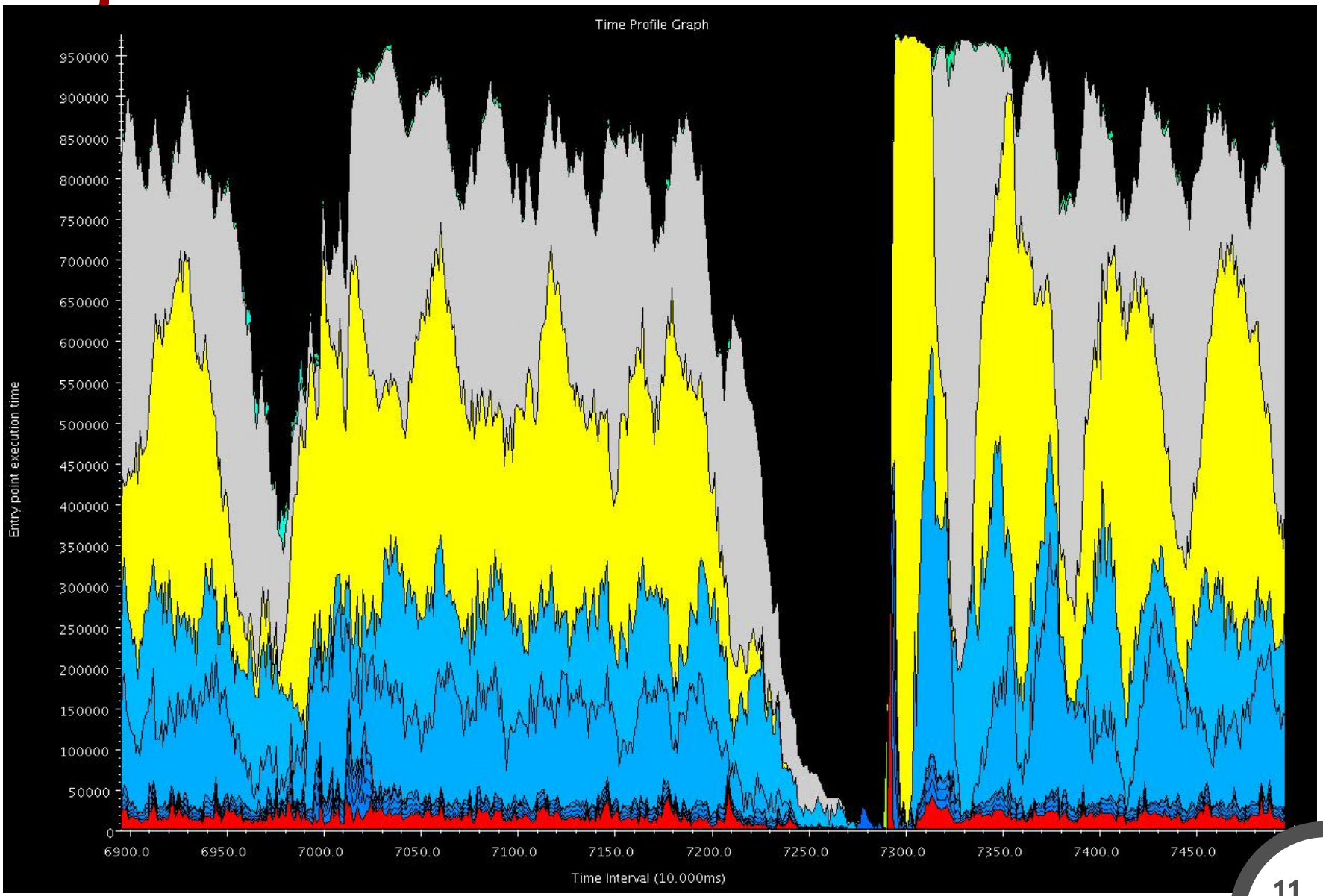
Projections Tools



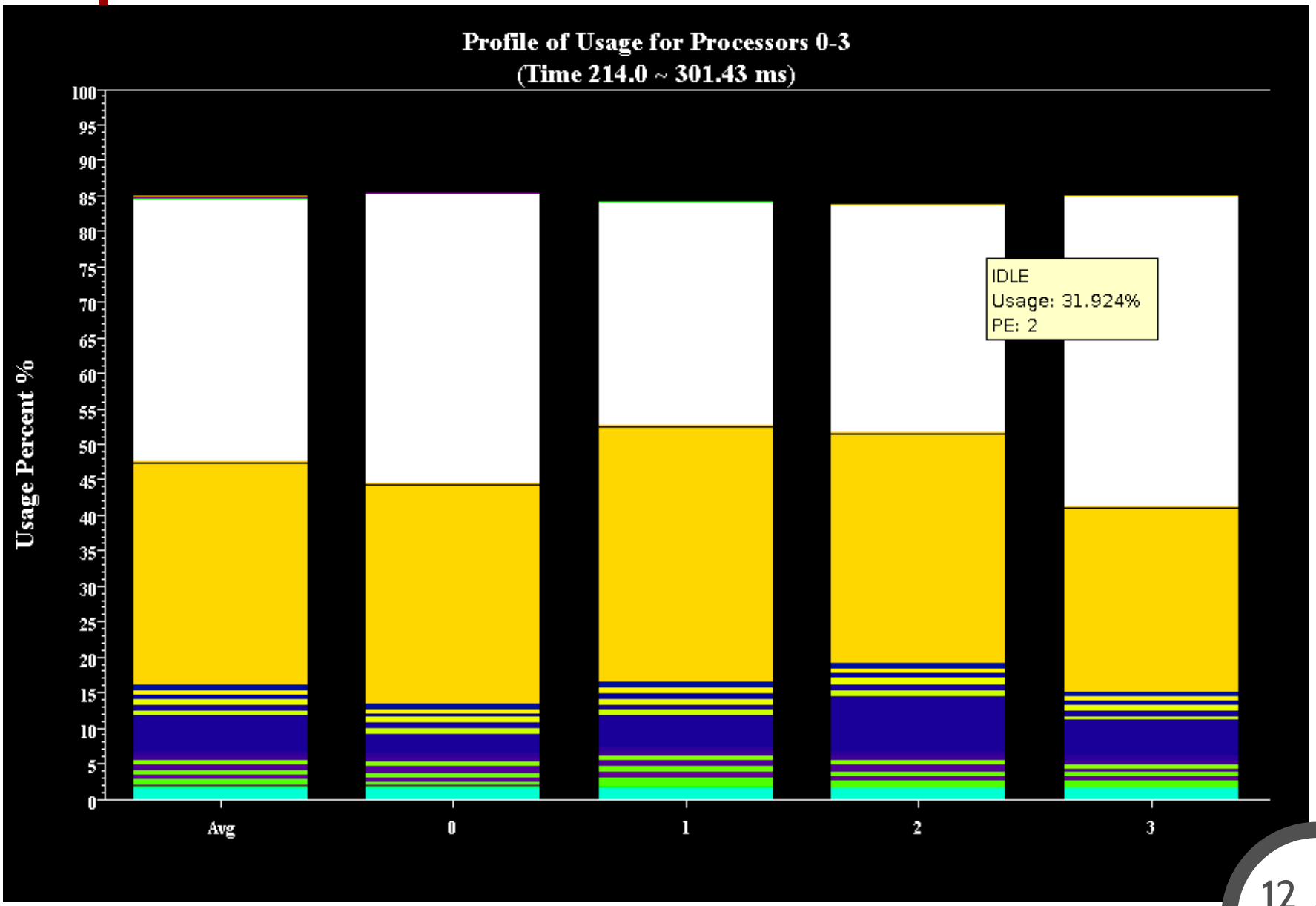
Projections Tools



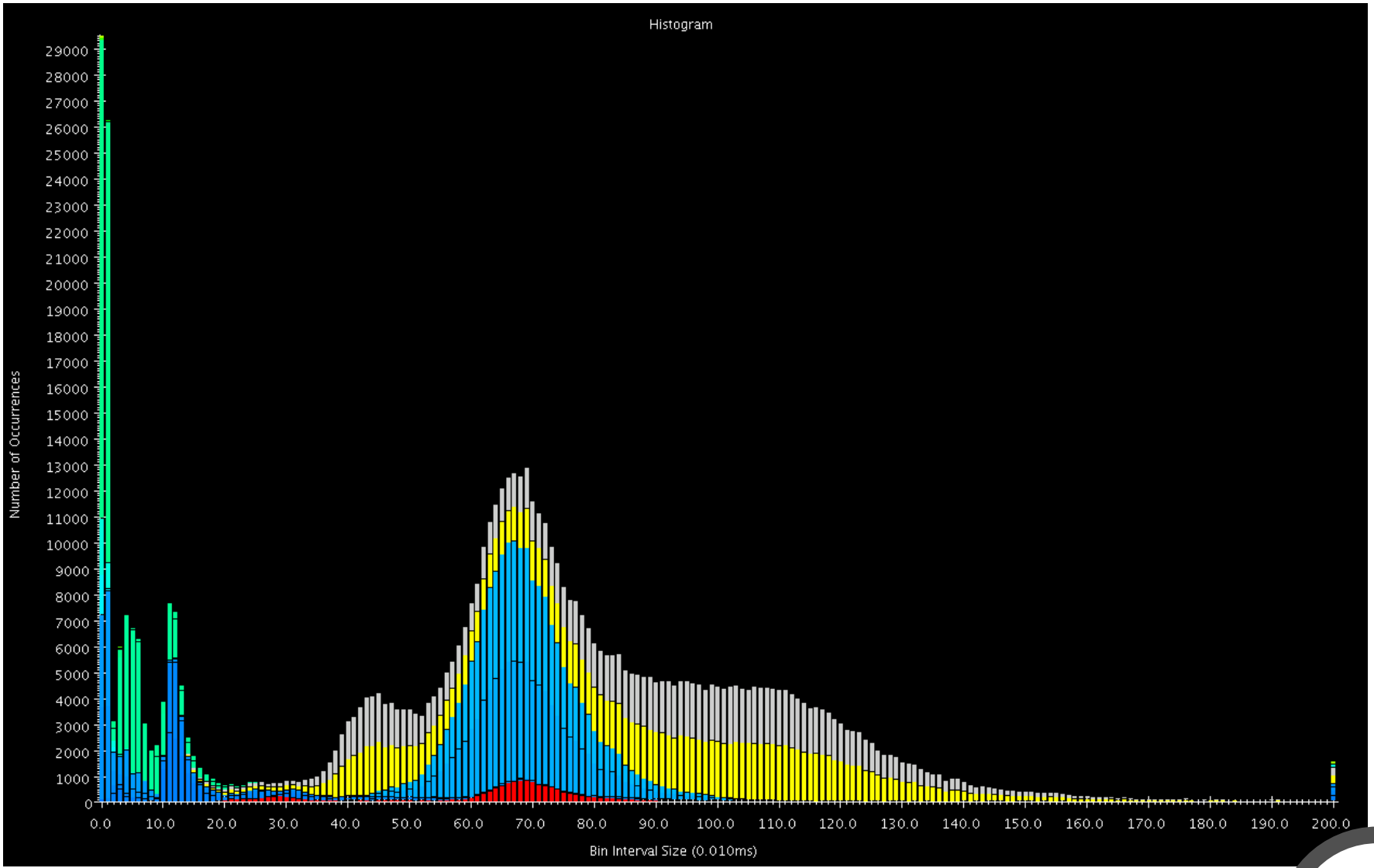
Projections Tools



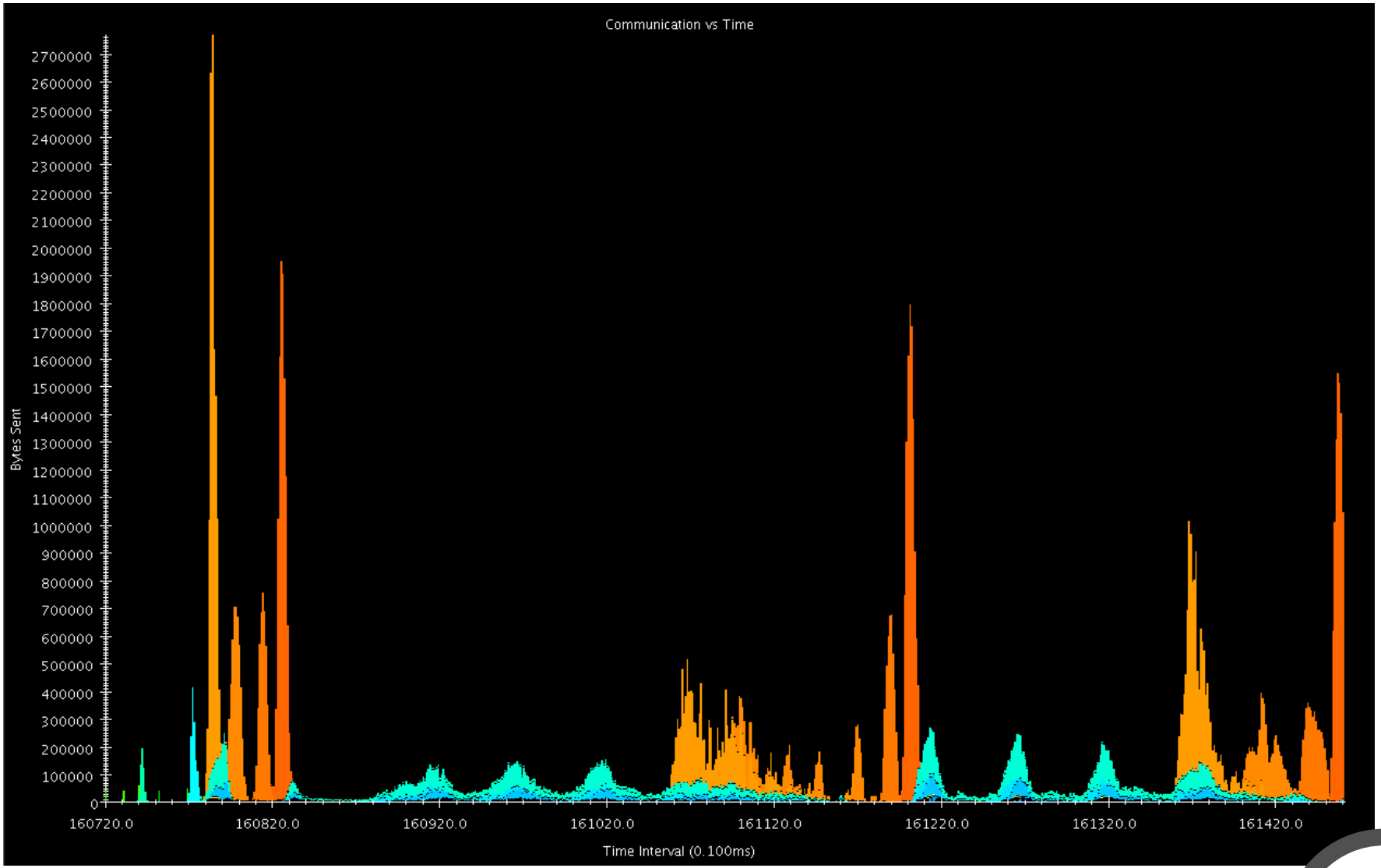
Projections Tools



Projections Tools



Projections Tools



Recent Advancements in Projections

Recently, we have added some advanced analysis tools to Projections:

- Cache latency measurement
- Communication thread tracing

Cache Latency Measurement

Using sampling techniques, we have developed a method to trace cache accesses and attribute them to specific lines.

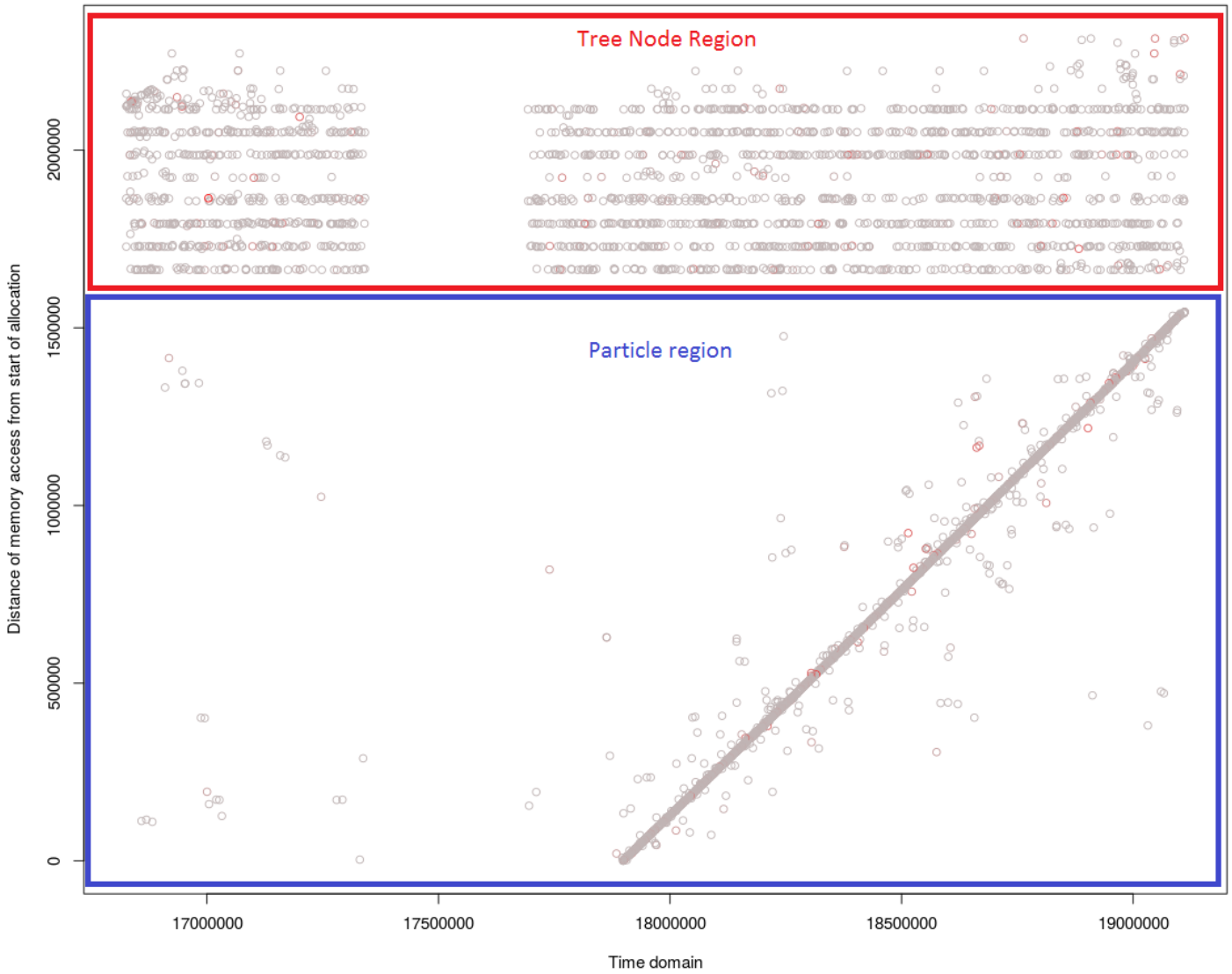
- Intel PEBS - Precise Event Based Sampling
- Provides instruction pointer with sample
- Provides latency of access, not mere hit/miss count

Cache Latency Measurement

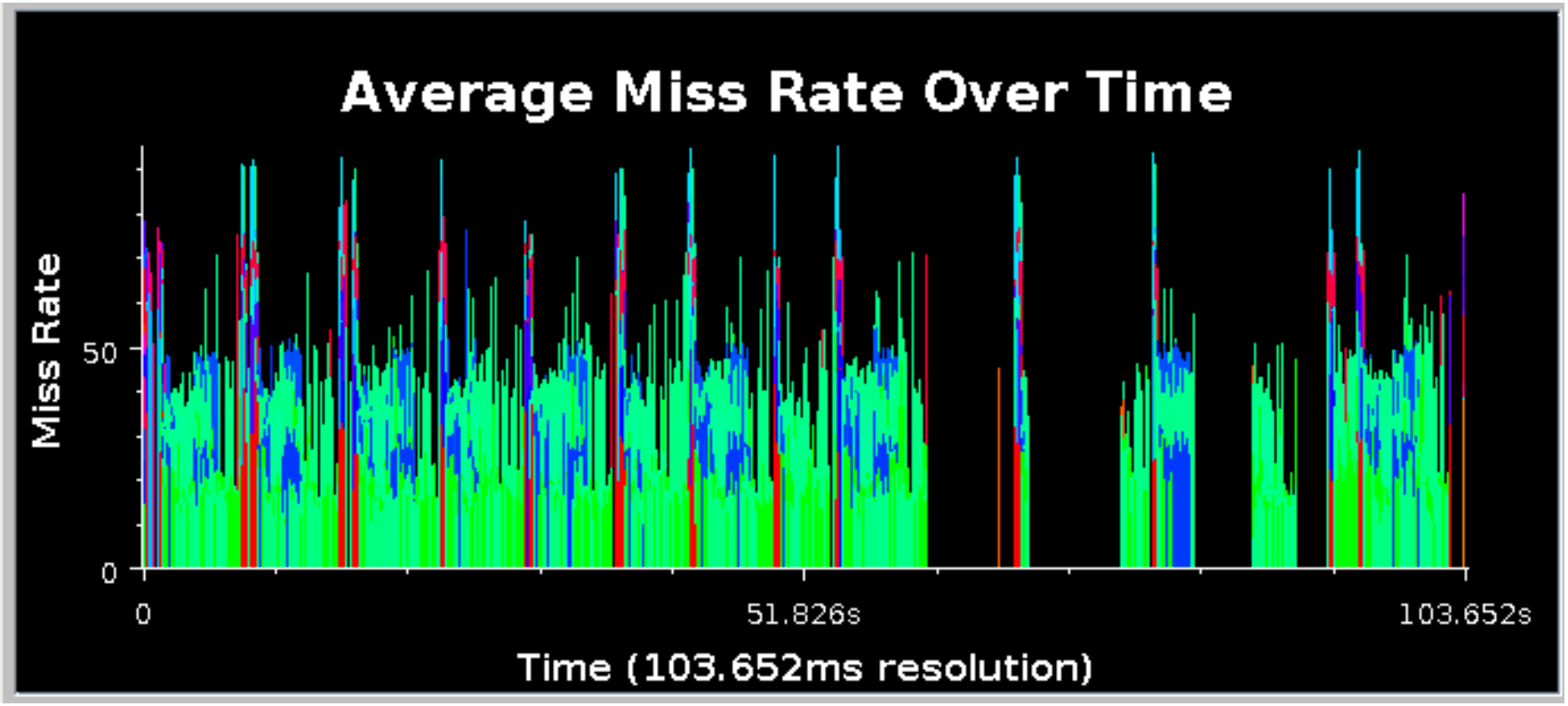
Collected data allows us to analyze:

- Spatiotemporal memory access pattern and latency
- Cache miss rate over time
- Cache latency per source line

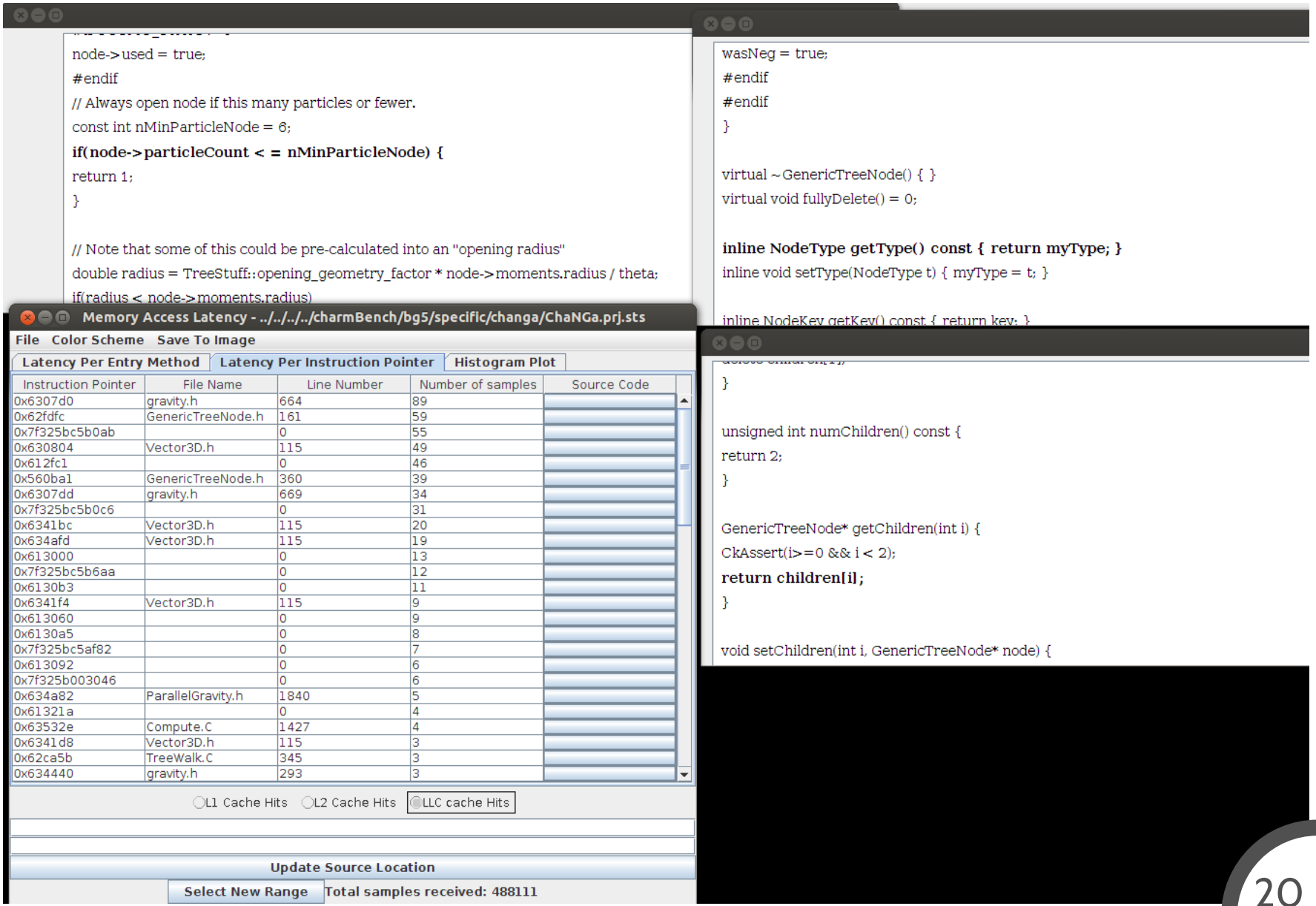
Cache Performance Tools



Cache Performance Tools



Cache Performance Tools

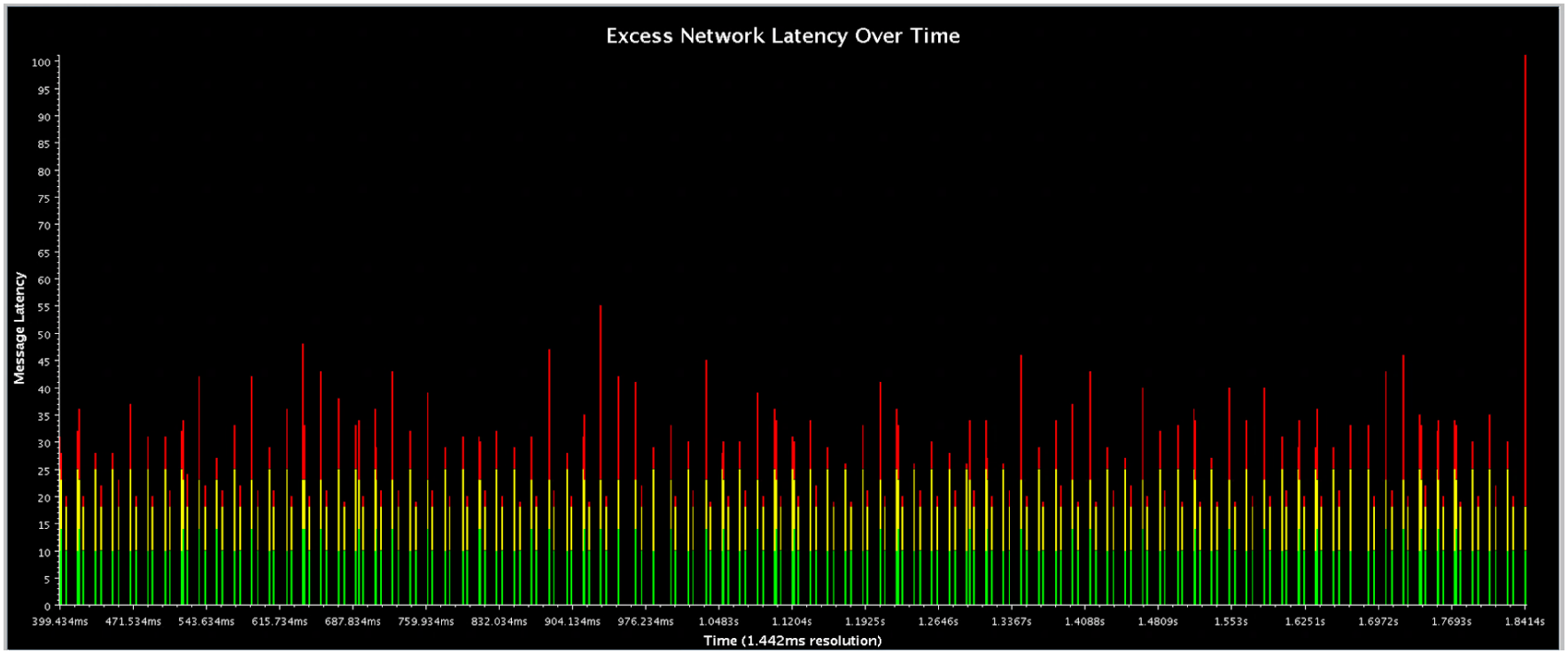


Communication Thread Tracing

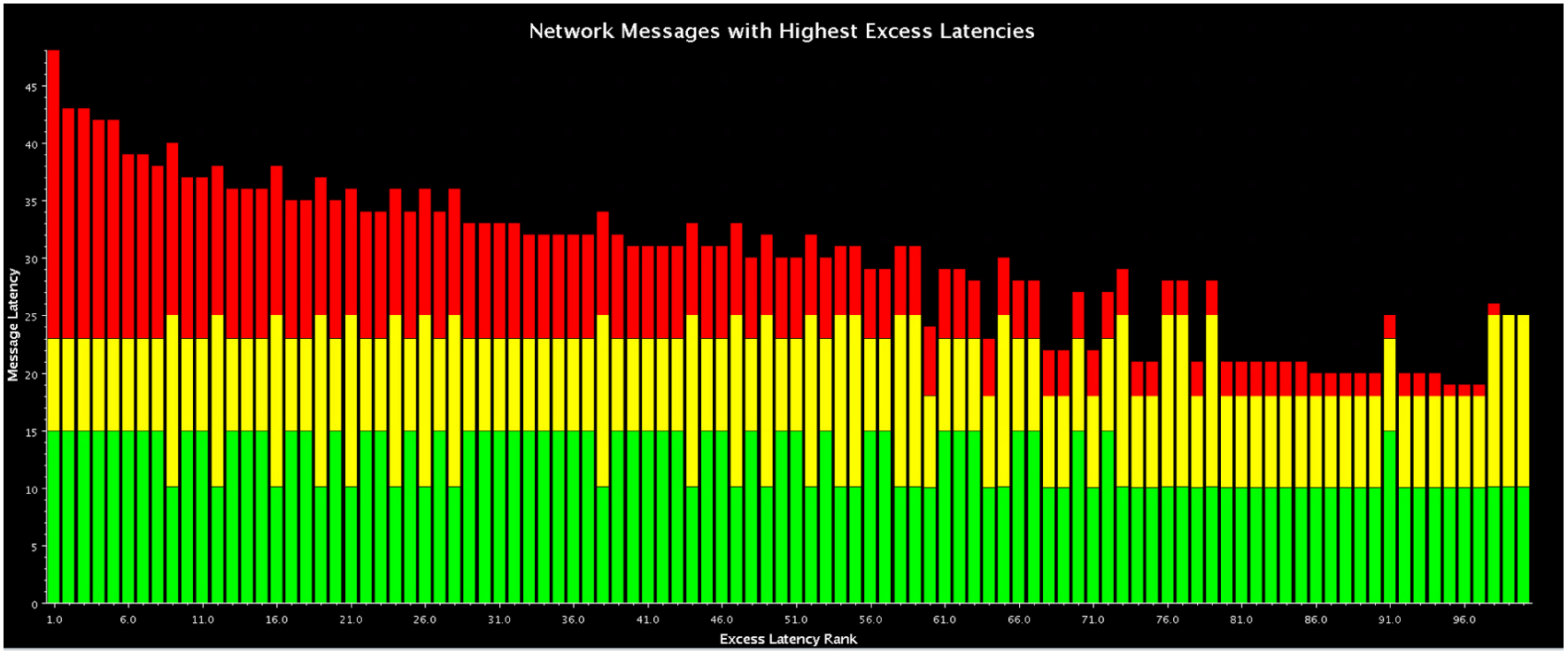
When Charm++ is used in SMP mode, every process has a communication thread. When built with `--enable-tracing-commthread`, the runtime will specifically trace this thread.

Using these logs, we can analyze network performance in detail.

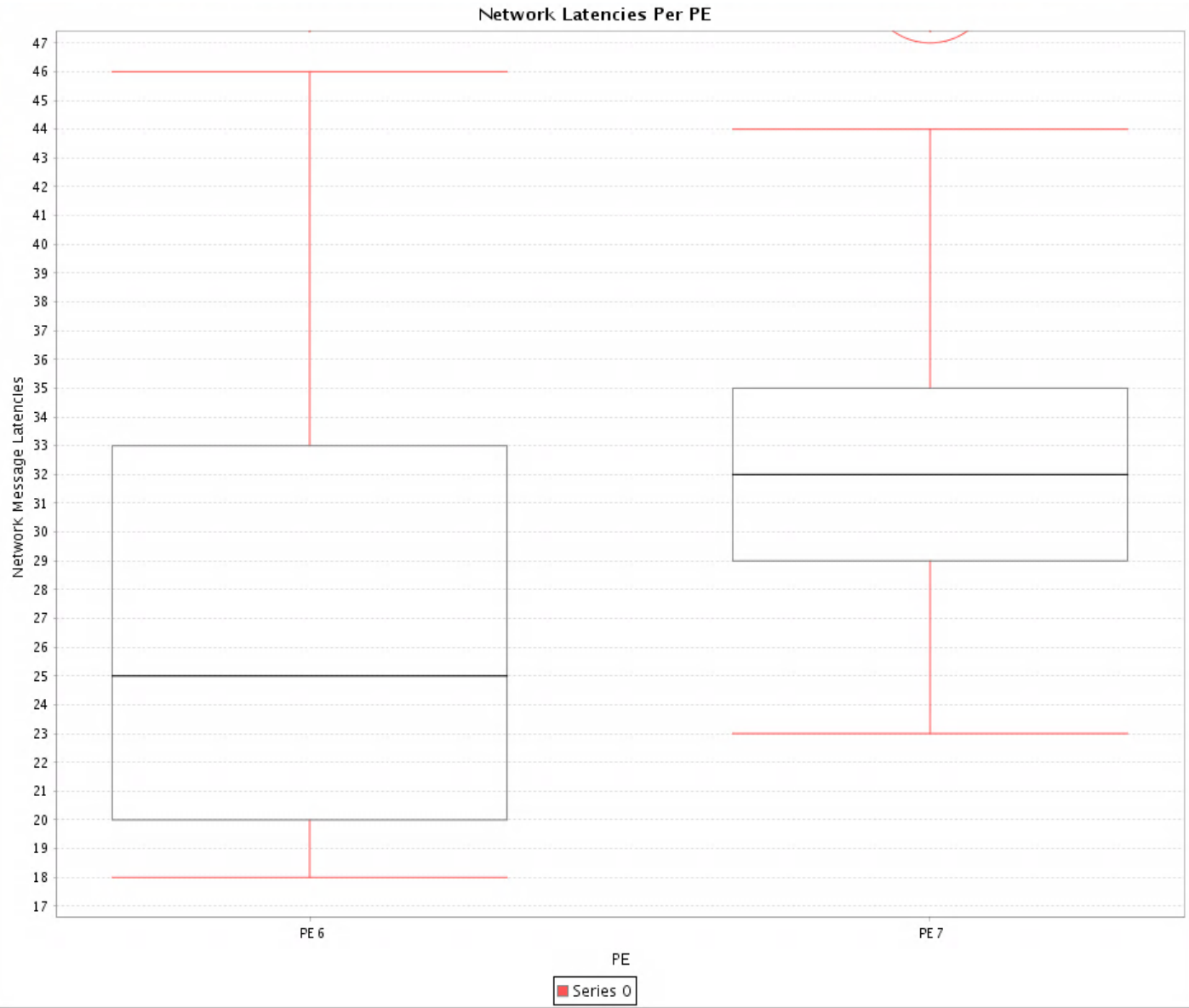
Communication Thread Tracing



Communication Thread Tracing



Communication Thread Tracing



Communication Thread Tracing

Future work:

- Identify communication thread oversubscription
- Estimate runtime given ideal network
- Use topology and routing information to identify network hotspots

Conclusion

- Performance analysis tools are *critical* for optimizing HPC applications
- Detailed metrics (performance counters, network delay, etc) can provide insight
- Make performance problems as obvious as possible
- Make it possible for users to write *ad hoc* analysis tools

Thanks