Message Passing with MPI

PPCES 2015

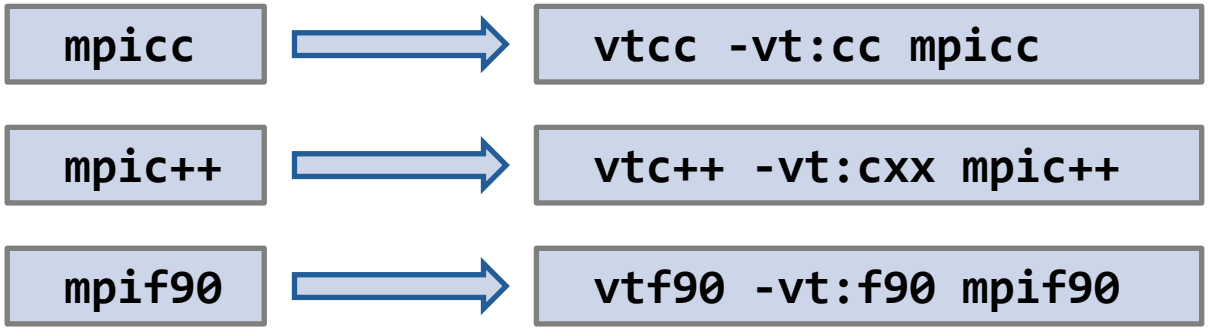
Hristo Iliev
IT Center / JARA-HPC

Vampir: The Suite

- **VampirTrace / Vampir tool suite:**
  - → **VampirTrace**
    - →instrumentation tool
    - →tracing library
    - →writes OTF trace files
    - →open-source code
  - → **Vampir**
    - →visualization and analysis GUI tool
    - →reads OTF files produced by VampirTrace
    - →commercial
- **On RWTH's cluster modules are located in the UNITE category.**

Vampir: Instrumentation with VampirTrace

- **Code first has to be instrumented using VampirTrace:**
  - → Recompile with instrumentation



  - → When run, the instrumented binary produces trace files in OTF format.

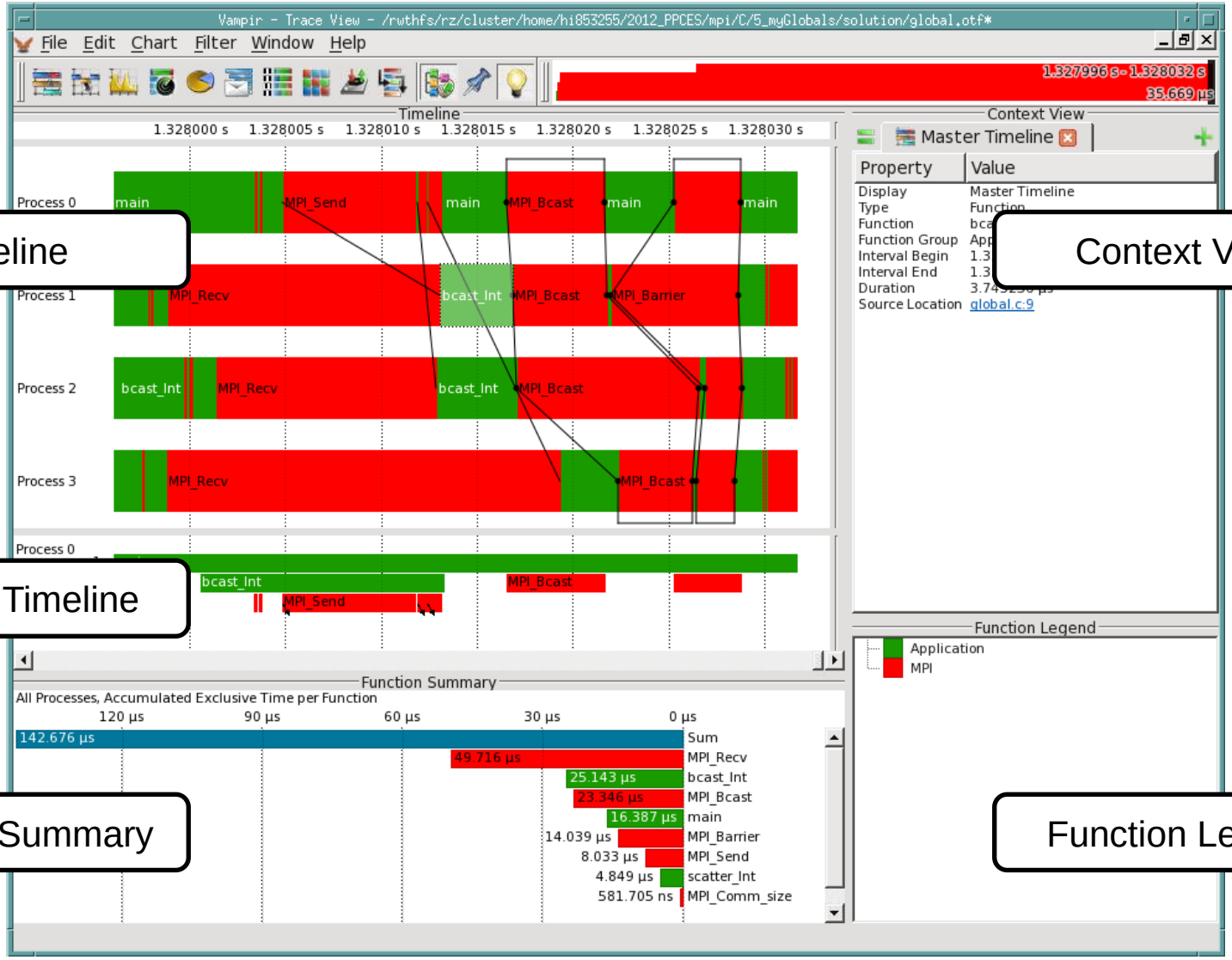
- **Instrumentation type selected via `-vt:inst <type>`**
  - → **`compinst`** compiler assisted instrumentation (default)
    all function calls traced; very detailed; huge trace files
  - → **`manual`** manual tracing using VampirTrace API
    traces only MPI events and user-specified events;
    significantly reduced trace file size

Vampir: Instrumentation with VampirTrace

- **VampirTrace is controlled by many environment variables**
  - → **`VT_BUFFER_SIZE`** internal trace buffer size; flushed to the disk when full (default: 32M)
  - → **`VT_FILE_PREFIX`** OTF file prefix (default: executable name)
  - → **`VT_MAX_FLUSHES`** number of trace buffer flushes before tracing is disabled (0 – no limit)
  - → **`VT_SYNC_FLUSH`** synchronised buffer flushes (default: no)

- **Things to be aware of:**
  - → By default buffers are flushed asynchronously and it takes time
  - → Significant skew in program's performance profile possible
  - → No trace written after abnormal program termination

Vampir: Trace View



Timeline

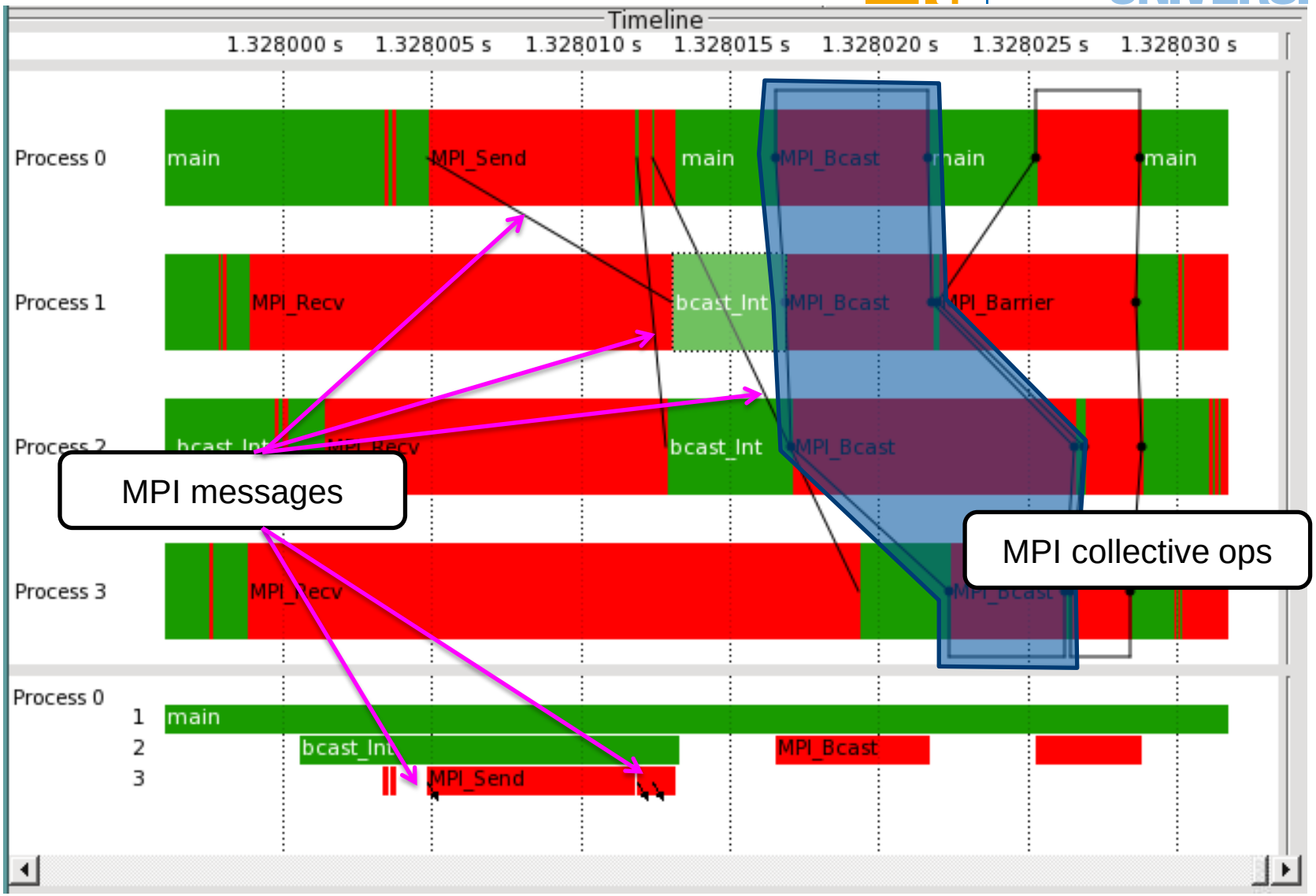
Context View

Process Timeline

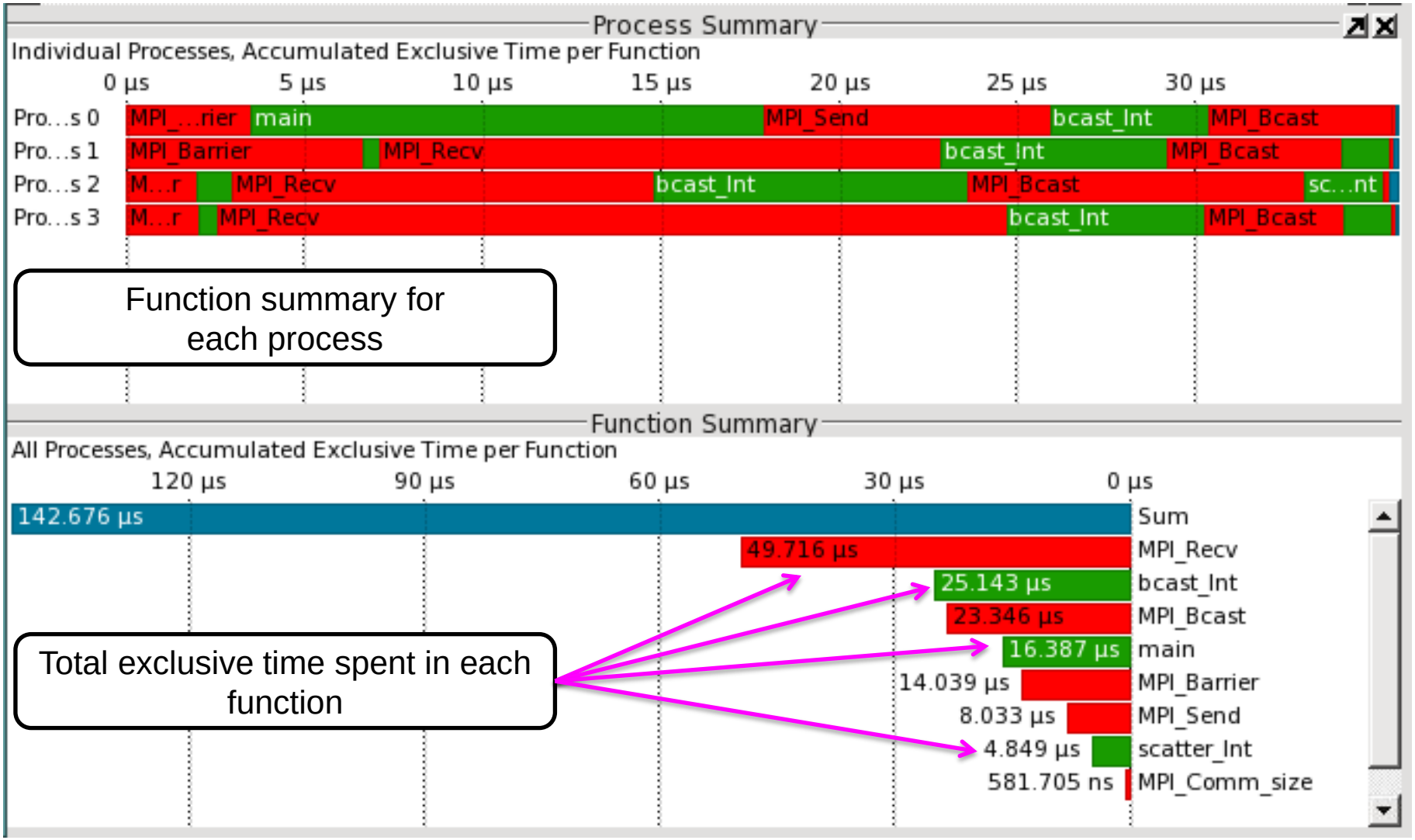
Function Summary

Function Legend

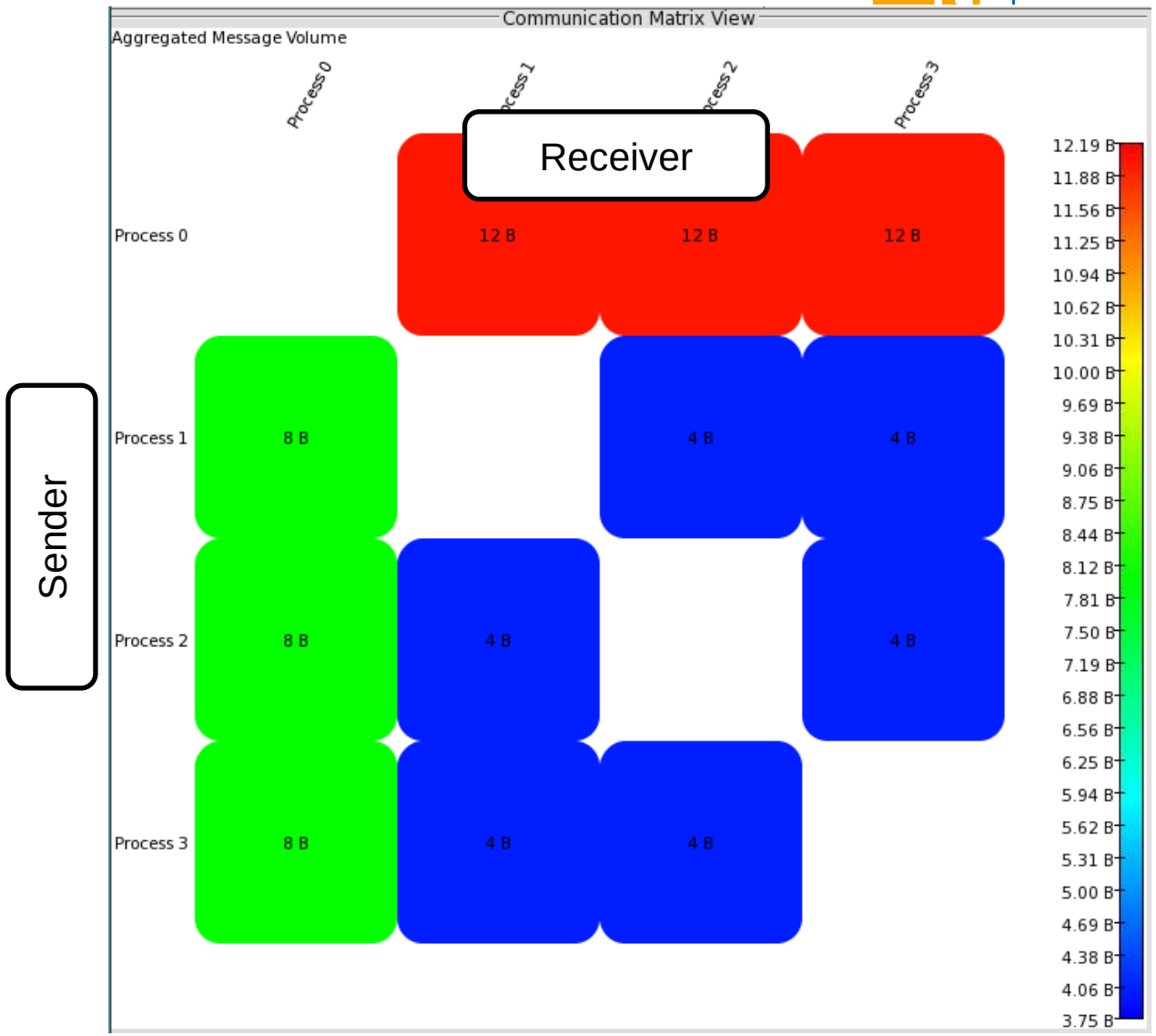
Vampir:
Timeline and Process Timeline



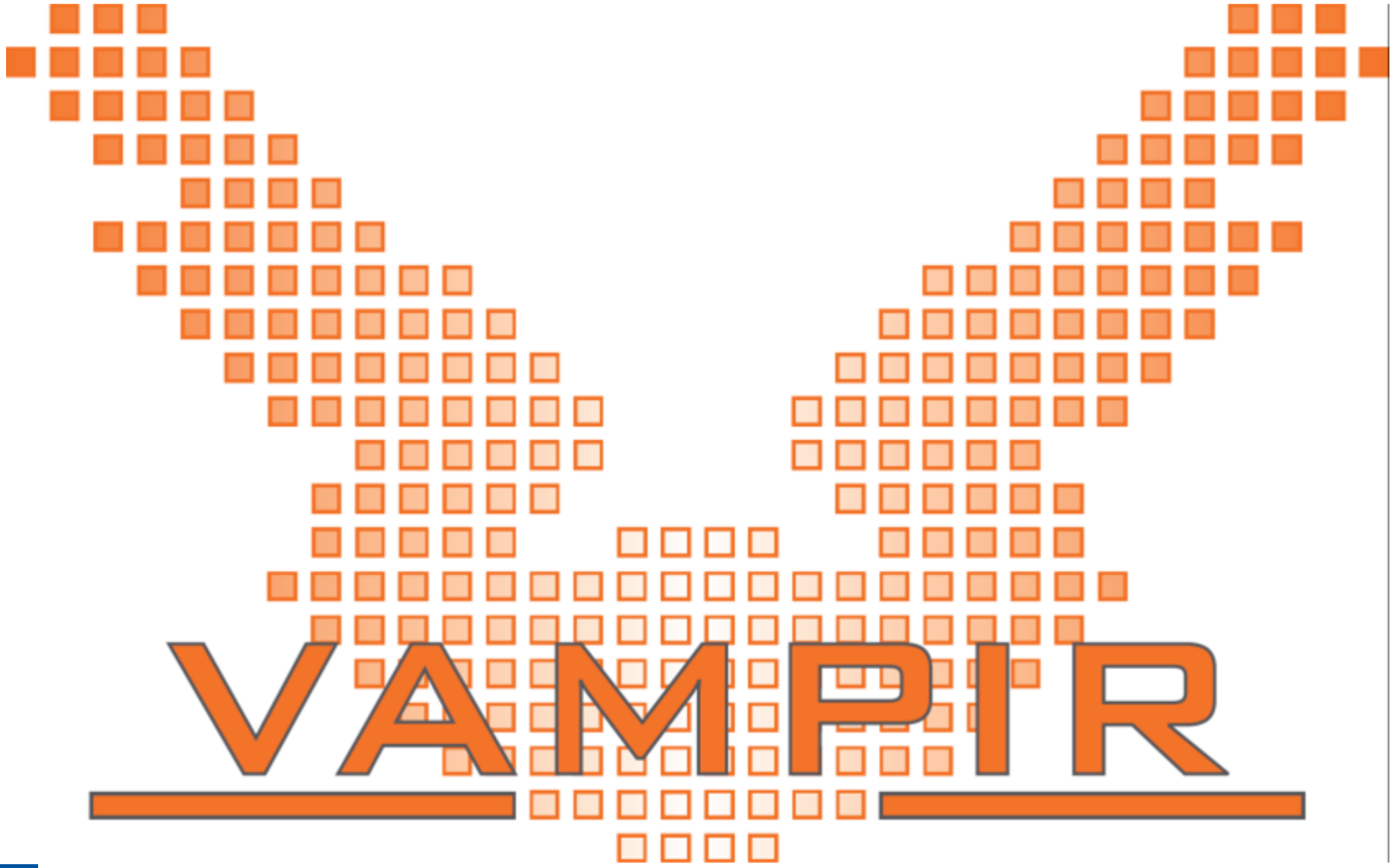
Vampir: Summaries



Vampir: Communication Matrix View



Vampir:
Live Demonstration



Thank you for your attention!