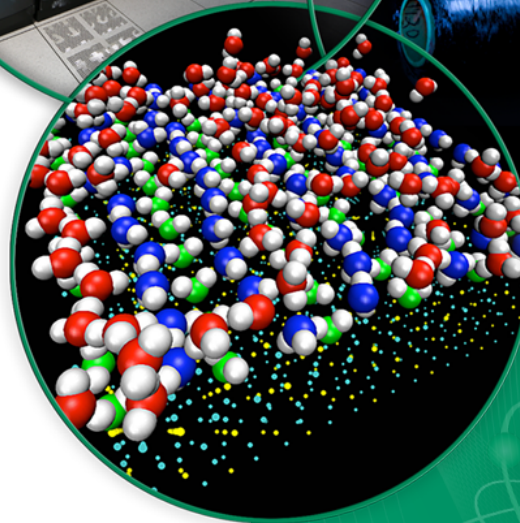
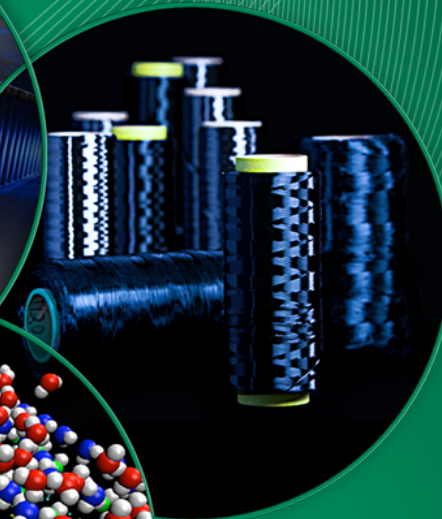


# Trace-based Performance Analysis using Vampir

Joseph Schuchart

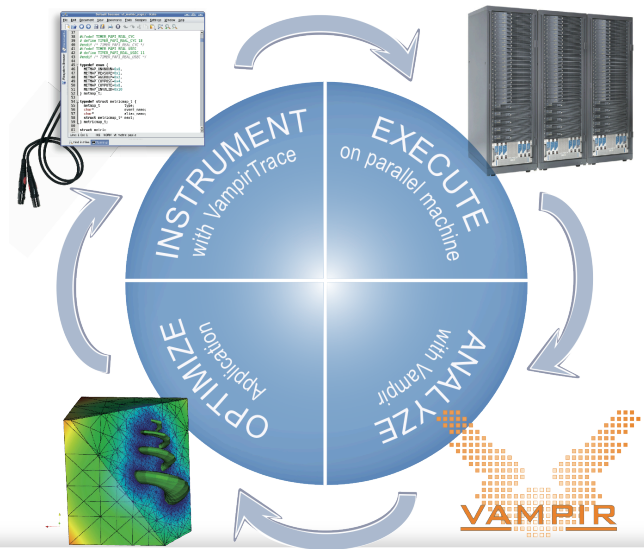
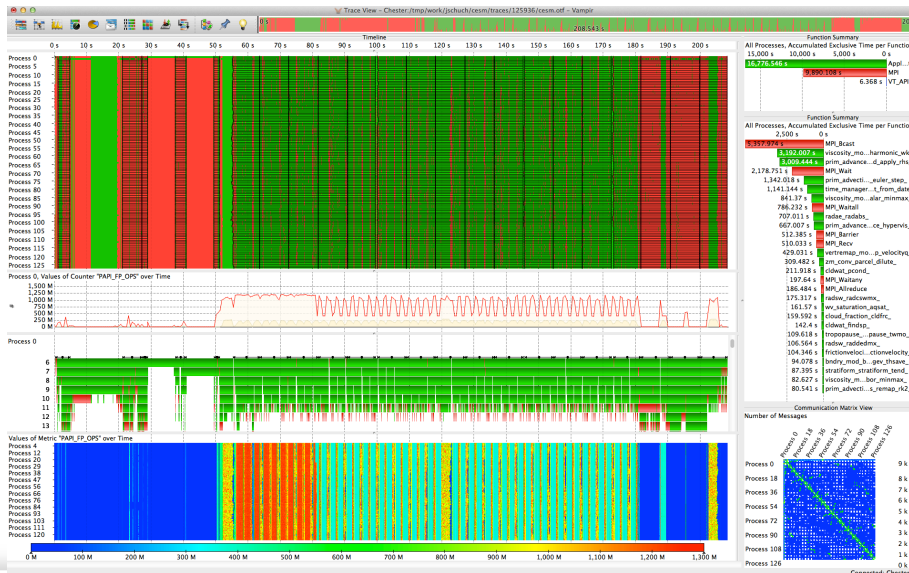


# Disclaimer

Performance tools will not automatically make you code run faster. They help you understand, what your code does and where to put in work.

# The Vampir Toolset

- Provides detailed insight into parallel applications
- No automatic analysis but sophisticated visualization
- Available modes:
  - Profiling: high-level summary of program behavior
  - Tracing: detailed insight, time-based relations of events

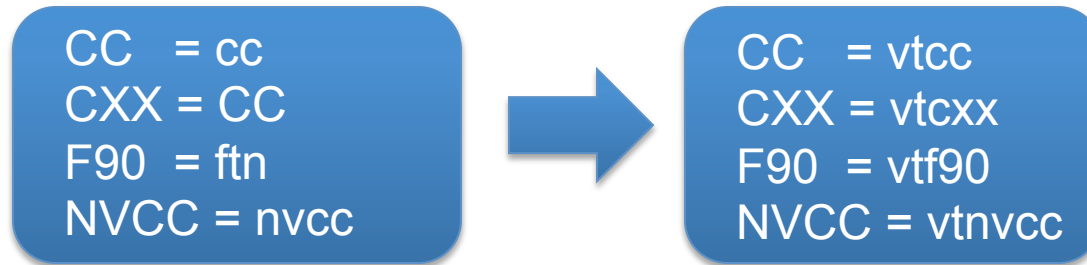


# Instrumentation: VampirTrace

- Scalable trace recording infrastructure
- Insert callbacks for events of interest via
  - Compiler instrumentation (C/C++, Fortran)
  - Binary instrumentation using Dyninst
  - Source rewriting using TAU/PDT
  - Manual instrumentation
  - Instrumented libraries (MPI, libc, Pthread)
- Supports also:
  - MPI, OpenMP, Pthreads
  - Performance metrics (PAPI, resource usage counter)
  - CUDA/CUPTI (also supports OpenACC)

# Compiler Instrumentation

- module load vampirtrace
- Use compiler wrapper



- Re-compile & re-link
- Options:
  - -vt:verbose
  - -vt:<seq,mt,hyb,mpi>
  - -vt:inst <compinst, tauinst, dyninst, manual>
  - -vt:help

# Execution

- Launch as usual
- Trace recording controlled via environment variables
  - Enable/Disable features
  - Set target directory for trace
  - Control buffer size and number of flushes
- Trace data is written to OTF files
  - One file per process/thread
  - For (really) large trace runs: use I/O forwarding

# Execution: Environment Variables

Environment Variable	Description	Default
VT_BUFFER_SIZE	Size of event buffer	32M
VT_NUM_FLUSHES	Maximum number of flushes	1
VT_PFORM_GDIR	Directory to hold trace data	

# Execution: vtsetup

The screenshot shows the VTSetup application window. The title bar is blue and contains the text 'VTSetup'. Below the title bar is a menu bar with 'File', 'Generate', 'View', and 'Help'. Below the menu bar is a toolbar with several icons. The main window is divided into two panes. The left pane is titled 'Configuration Steps:' and contains a list of steps: 'General Trace Settings' (selected with a red dot), 'Optional Trace Features', 'Counter', and 'Filtering and Grouping'. Below these steps are three checkboxes: 'Step 1 General Setup', 'Step 2 Group Setup', and 'Step 3 Filter Setup'. The right pane is titled 'General Trace Settings' and contains two sections: 'Trace File Settings' and 'Trace Buffer Settings'. The 'Trace File Settings' section has two fields: 'Executable:' and 'Trace filename:'. The 'Trace Buffer Settings' section has one field: 'Buffer size per process:'. The 'Buffer size per process:' field is set to '32' and has a unit dropdown menu set to 'MB'. There are also help icons (question marks) next to the 'Trace filename:' and 'Buffer size per process:' fields.

VTSetup

File Generate View Help

Configuration Steps:

- **General Trace Settings**
- Optional Trace Features
- Counter
- Filtering and Grouping

☐ Step 1 General Setup

☐ Step 2 Group Setup

☐ Step 3 Filter Setup

### General Trace Settings

- Trace File Settings -

Executable:

Trace filename:

- Trace Buffer Settings -

Buffer size per process:  MB

## Infobox:

### Buffer size per process:

Set the size of the internal event buffer, which is the place where the event records are stored before being written to a file.

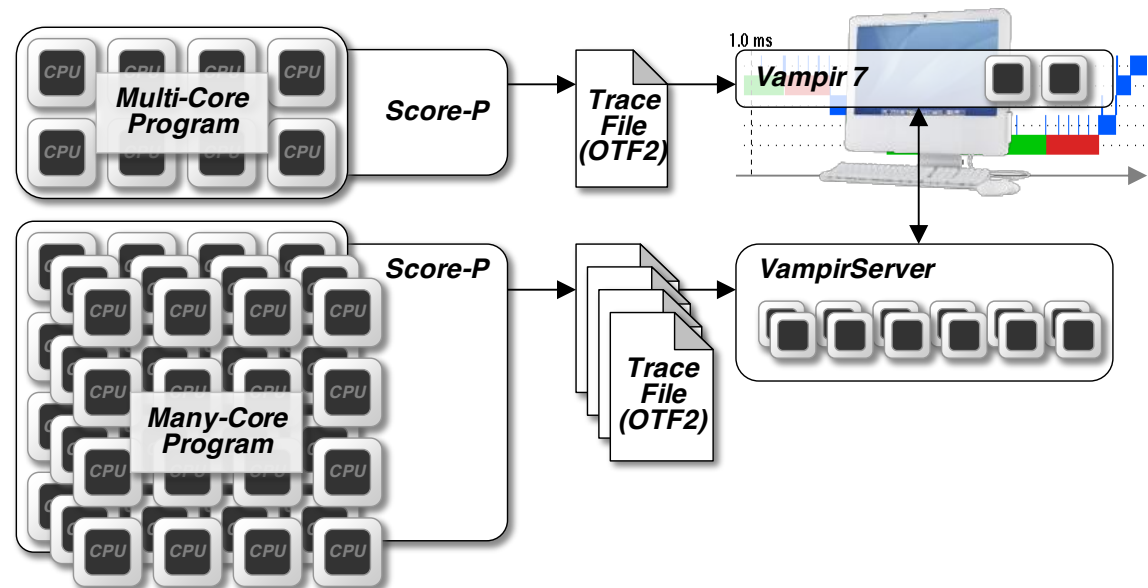
**Default:** 32 M

**Variable:** VT\_BUFFER\_SIZE



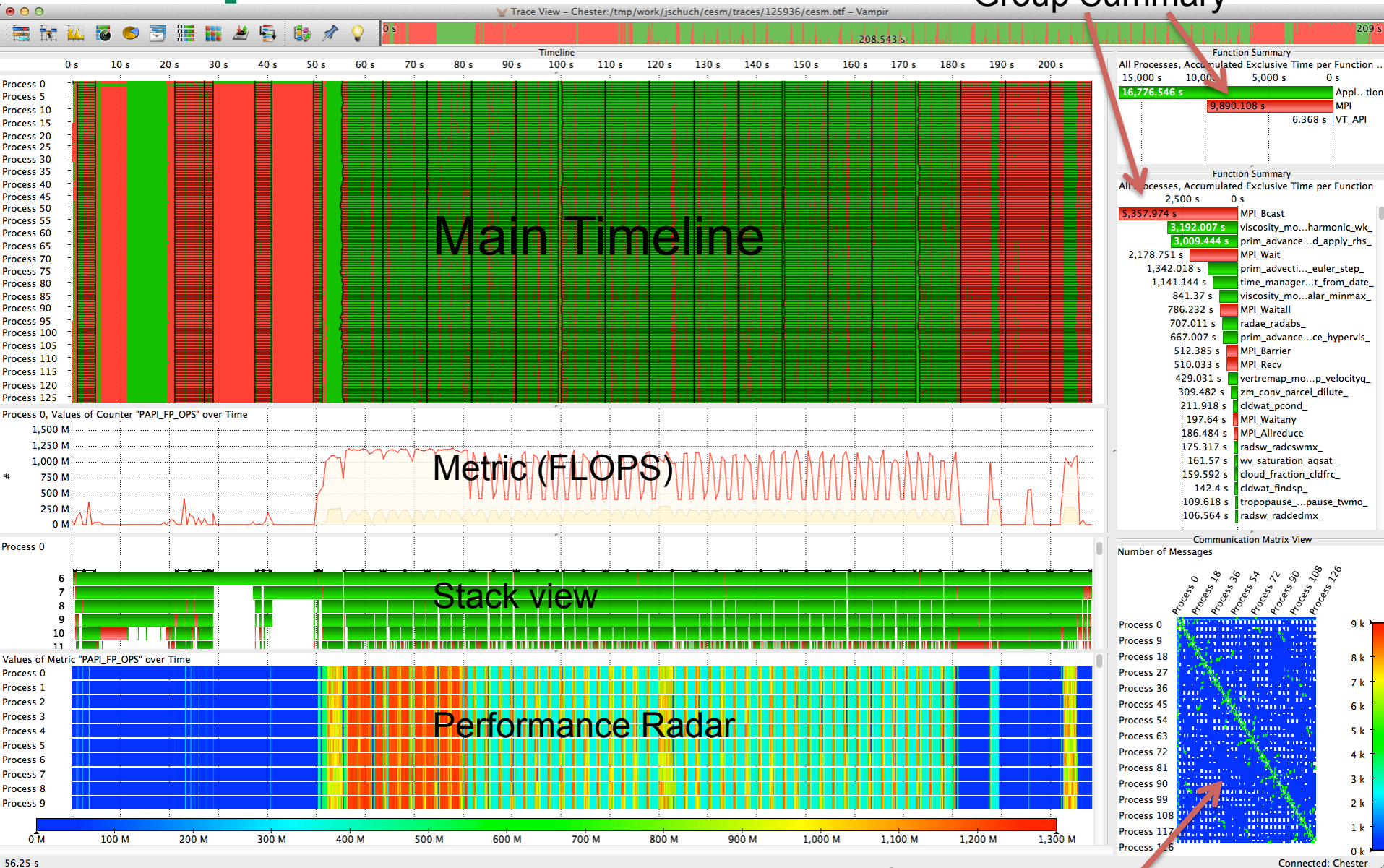
# Analysis: Vampir

- Scalable visualization infrastructure
- Arbitrary browsing, selection, and zooming in trace data
- Overview and detailed view of dynamic behavior
- Statistics and Metrics



# Example Trace: CESM

## Function and Function Group Summary



# Case Study: LAMMPS

- C++ Code
  - Lots of function calls
  - Buffers fill up fast

- Approach:

- Run with a filter

```
1 MPI_* -- -1
2 * -- 10000
```

- Select verbose/uninteresting functions
  - Exclude them from instrumentation

```
-finstrument-functions-exclude-file-list=/usr/include,include/g++,math_extra
-finstrument-functions-exclude-function-list=pack_3d,map,timing,Timer,operator
```

# Live Demo: LAMMPS trace

1. Connect with NX to home2.ccs.ornl.gov
2. `ssh -X titan.ccs.ornl.gov`
3. `module load vampir`
4. Start vampir
5. Load trace:  
`/tmp/proj/trn001/TitanDev/traces/lammps_lq_1/lmp_xk7.vt.otf`  
`/tmp/proj/trn001/TitanDev/traces/lammps_lq_2/lmp_xk7.vt.otf`

# Questions?

Contact us!

Joseph Schuchart ([schuchaj@ornl.gov](mailto:schuchaj@ornl.gov))

Bldg. 5700 Rm. B206