pallas

Competence in High Performance Computing

Linux tools for debugging and profiling MPI codes

Werner Krotz-Vogel, Pallas GmbH

<krotz@pallas.com>

MRCCS September 02000

Pallas GmbH
Hermülheimer Straße 10
D-50321 Brühl, Germany

info@pallas.com
www.pallas.com

Pallas Tools Software

State-of-the-art program development tools ...

... in detail:

- Vampir-2.5 (online-Demo), Vampirtrace-2.0, Dimemas
- Etnus TotalView 4.0 Multi-process Debugger

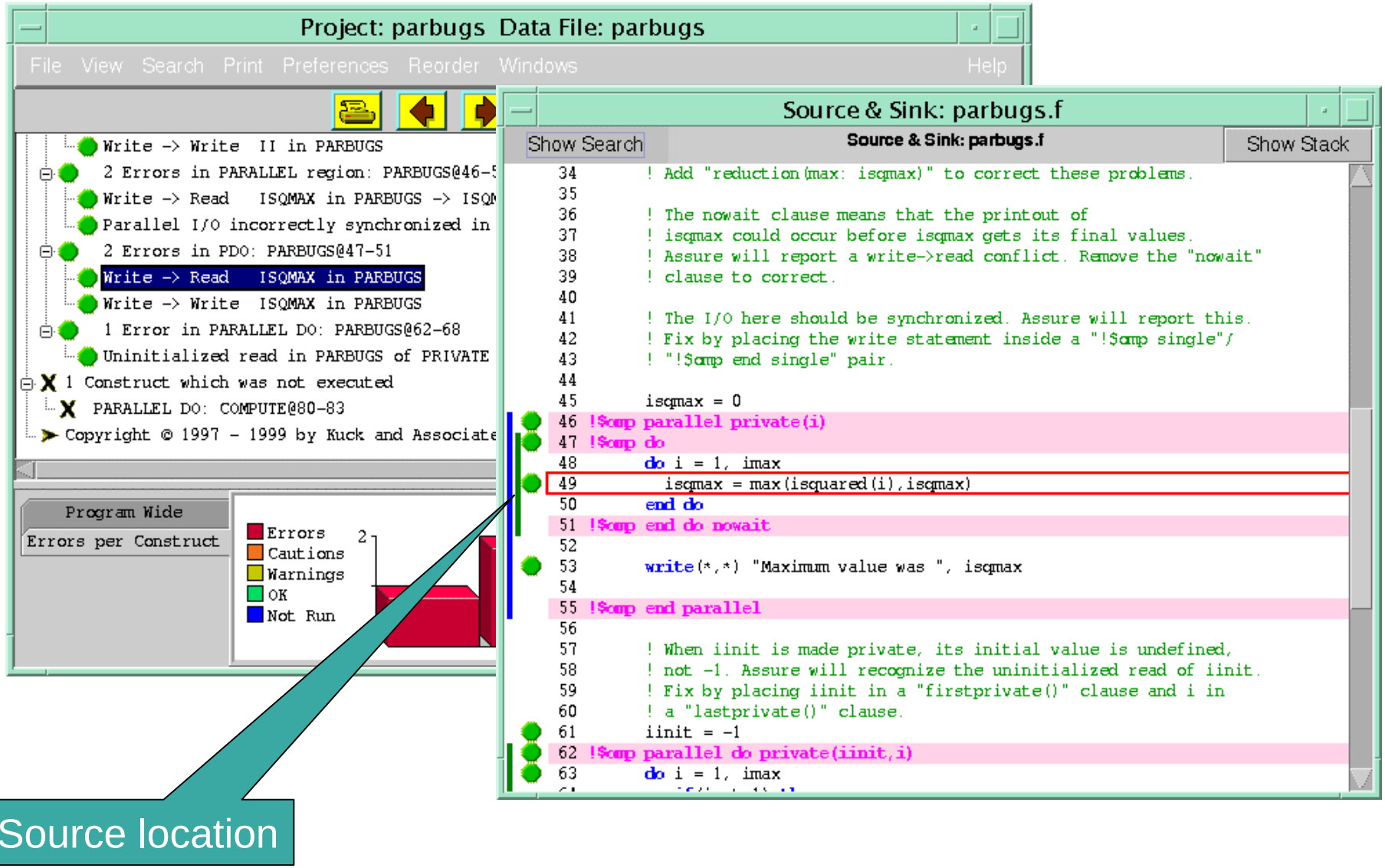
... briefly:

- KAP/Pro Toolset 3.8, OpenMP
- KAI C++ 3.4, ISO standard
- PGI 3.1 x86 Compilers, Cluster Development Kit (CDK)
- FORESYS - Fortran Restructuring Tool

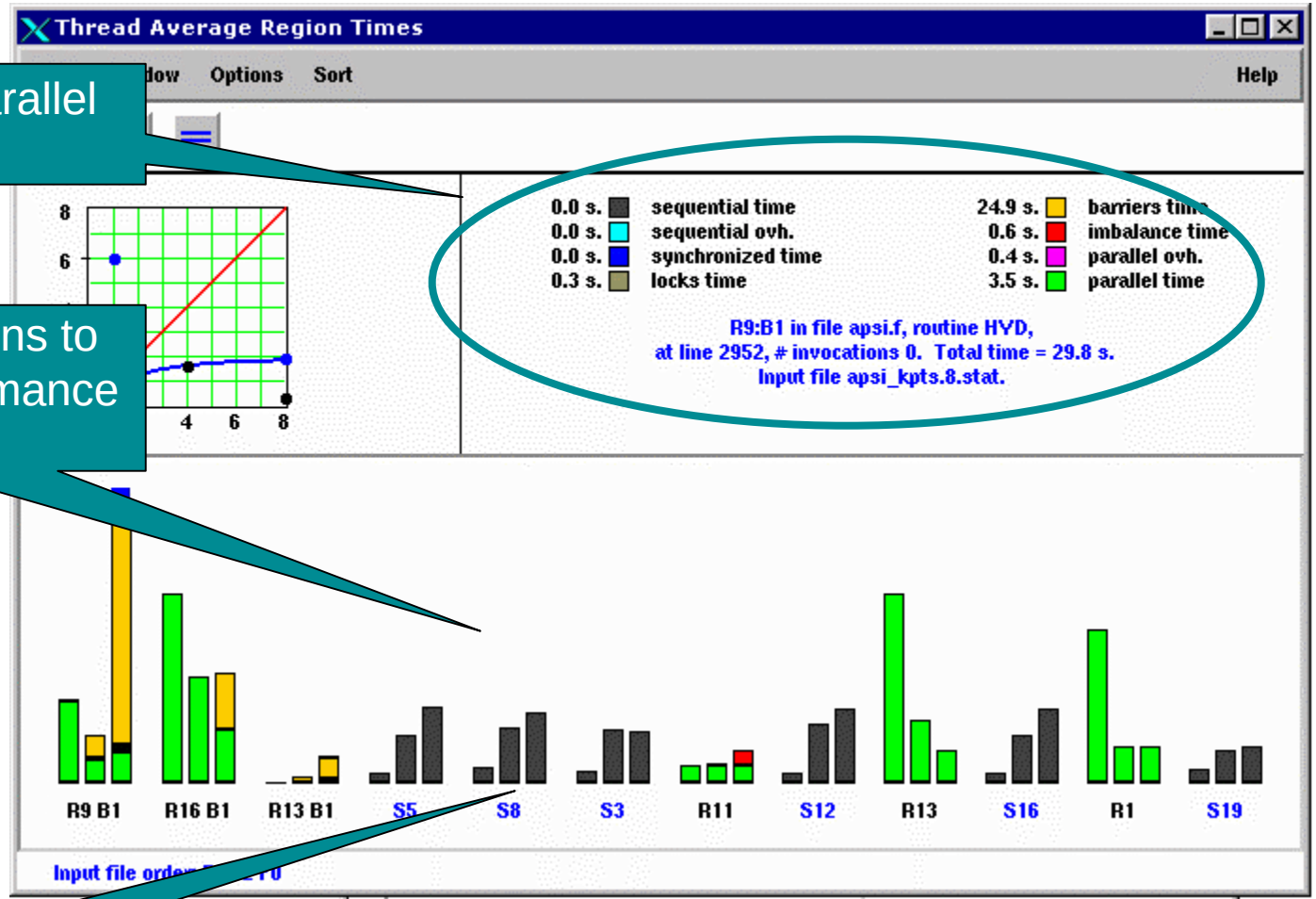
... free open source:

- PMB - Pallas MPI Benchmark Suite (incl. “effective Bandwith”)

KAP/Pro Toolset - Assure Example



KAP/Pro Toolset - GuideView Example



Analyse each parallel region

Sort or filter regions to navigate to performance hotspots

Identify serial regions that hurt scalability

KAI C++

The most modern, best performing, platform independant C++

- ISO C++ standard syntax, including exeptions and member templates
- ISO C++ standard class library
- multi-platform support
- meet C performance requirements
- thread safety (on most platforms)

PGI Cluster Development Kit (CDK)

Compilers & Tools ...

- PGI 3.1 x86 compilers , C, C++, F77, F90, HPF, pgrof, pgdbg
- SMP/OpenMP support for C, C++, F77, F90

... plus convenient add-on´s:

- parallel ScaLAPACK
- optimized BLAS, LAPACK
- MPI/mpich
- PVM
- PBS - Portable Batch System
- Tutorial, examples
- Cluster management utilities

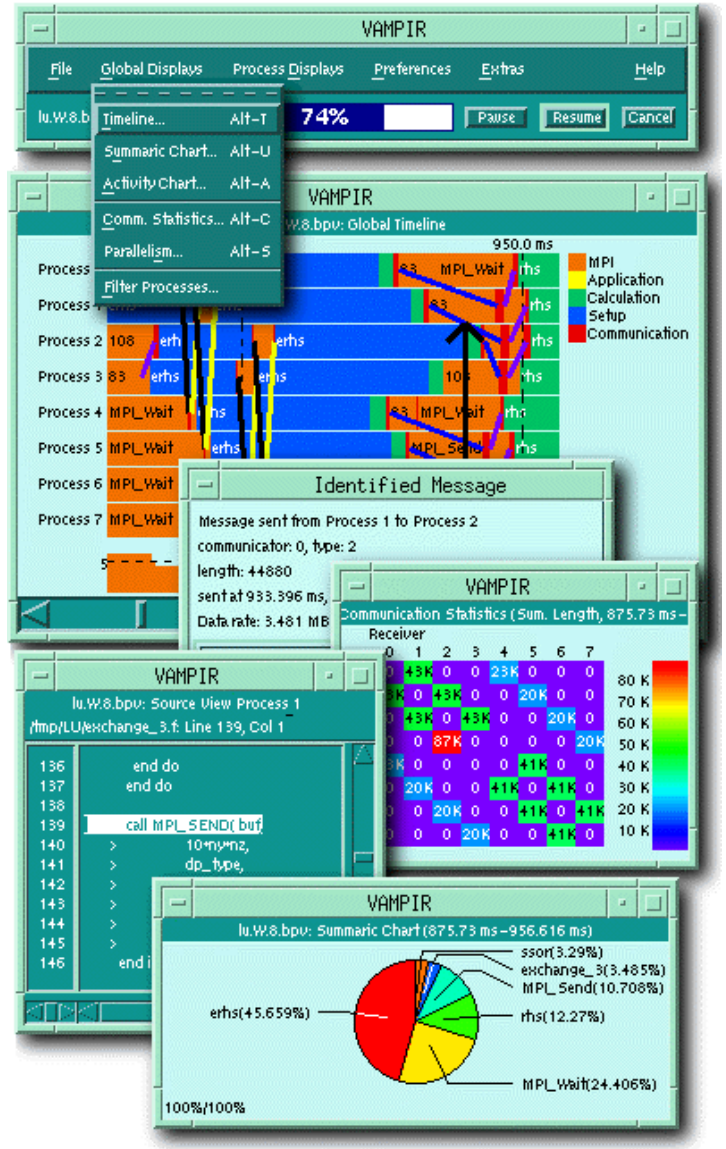
FORESYS

- Translates FORTRAN code (F77 - F95) into abstract syntax tree (ForLib)
- FORTRAN code consistency checks (definitions of functions, common blocks etc.)
- Interactive visualization & analysis of inconsistencies
- Upgrading from FORTRAN 77 to FORTRAN 90
- Interactive/batch analysis of parallelization possibilities
- Automatic code quality analysis/improvements

pallas

Vampir 2.5

Visualization and
Analysis of
MPI
Programs



Vampir

- Current version: Vampir 2.5

- Significant new features
  - support for collective MPI operations
  - trace comparison
  - tracefile re–write
  - message–length histogram
  - local and global calling trees
  - source–code reference
  - support for MPI–2 I/O operations

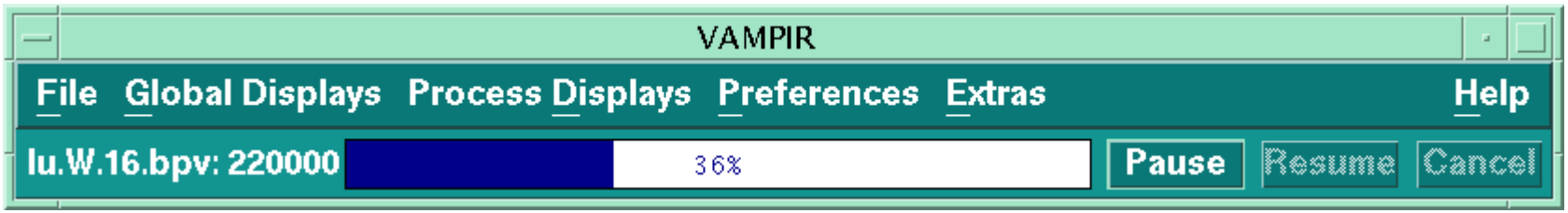
Vampir Features

- Offline trace analysis for MPI (and others ...)
- Traces generated by Vampirtrace tool (`ld ... -lVT -lpmpi -lmpi`)
- Convenient user–interface
- Scalability in time and processor–space
- Excellent zooming and filtering
- High–performance graphics
- Display and analysis of MPI and application events:
  - execution of MPI routines
  - point–to–point and collective communication
  - MPI–2 I/O operations
  - execution of application subroutines (optional)

- Easy customization

Vampir Main Window

Vampir 2.5 main window



- Tracefile loading can be interrupted at any time
- Tracefile loading can be resumed
- Tracefile can be loaded starting at a specified time offset

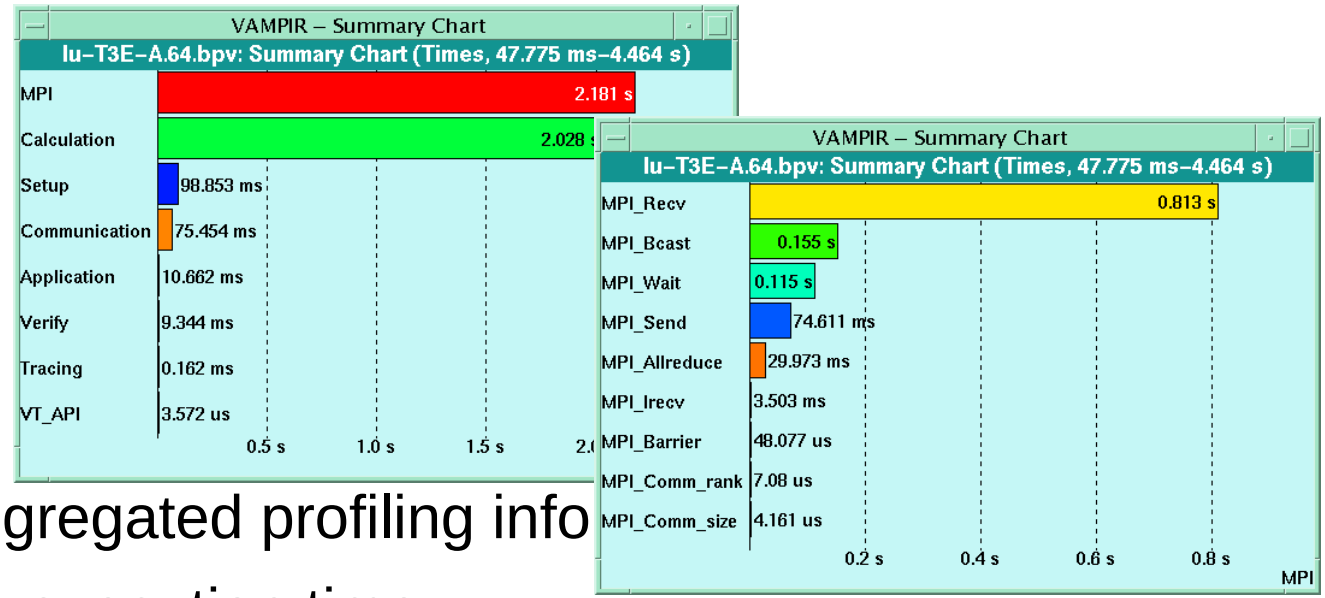
- Tracefile can be re–written (re–grouped symbols)

Vampir Displays

- Global displays show all selected processes
  - Summary Chart: aggregated profiling information
  - Activity Chart: presents per–process profiling information
  - Timeline: detailed application execution over time axis
  - Communication statistics: message statistics for each process pair
  - Global Comm. Statistics: collective operations statistics
  - I/O Statistics: MPI I/O operation statistics
  - Calling Tree: draws global or local dynamic calling trees

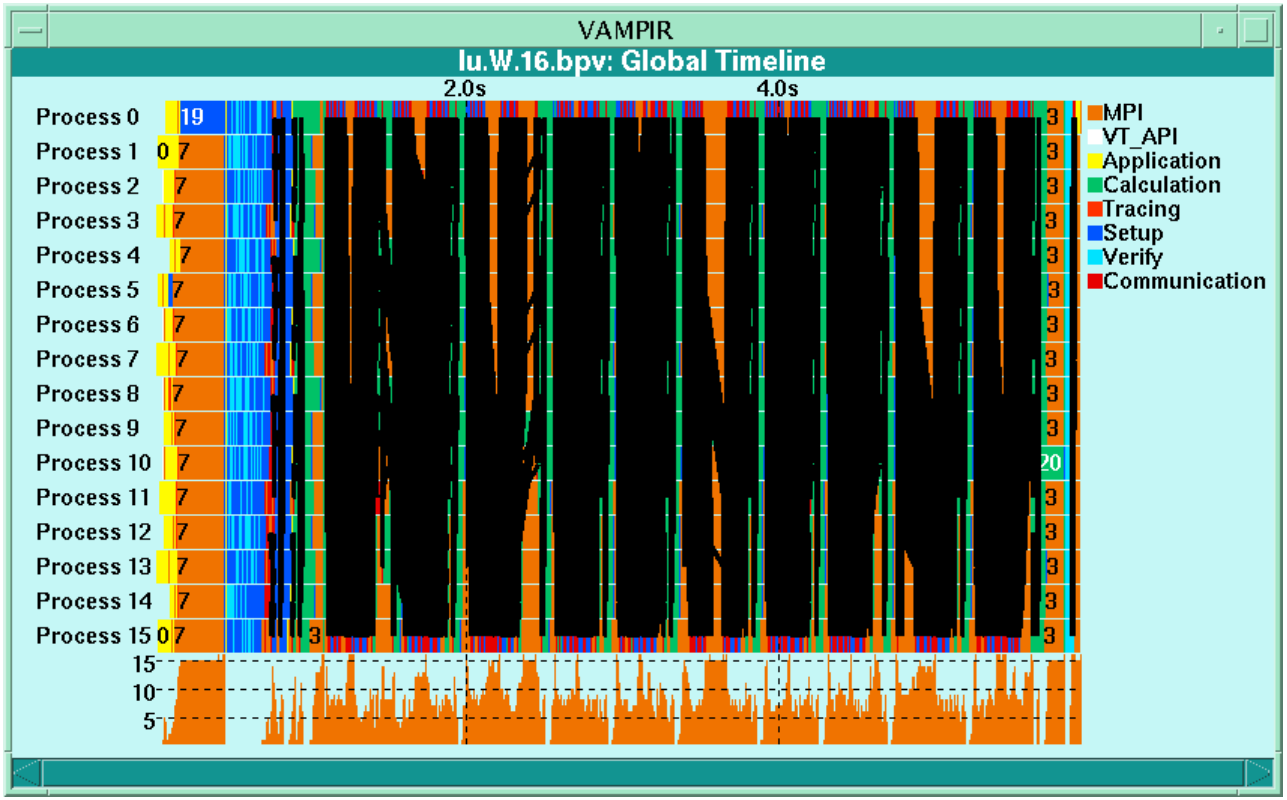
- Process displays show a single process per window
  - Activity Chart
  - Timeline
  - Calling Tree

Summary Chart



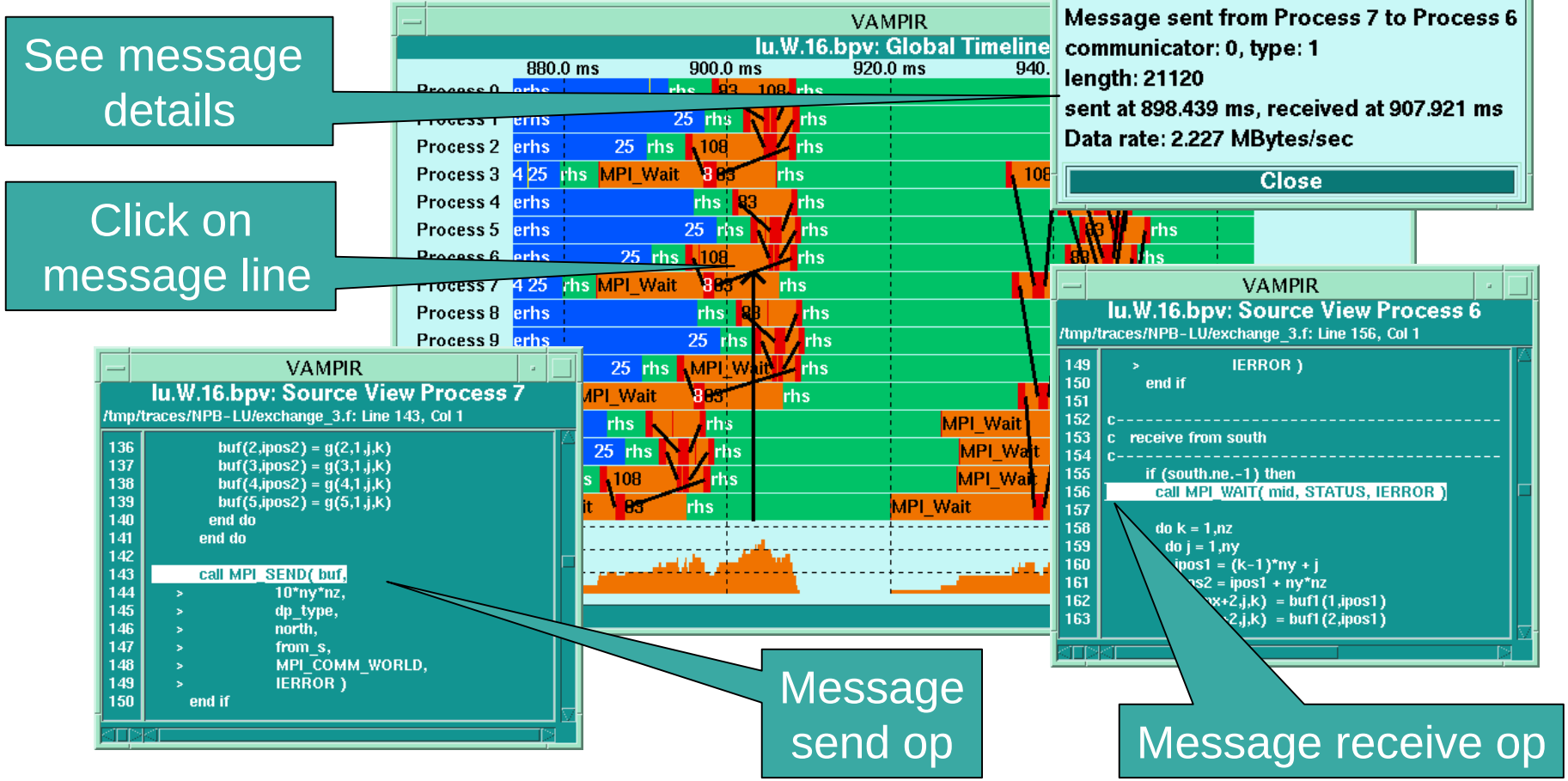
- Aggregated profiling info
  - execution time
  - number of calls
- Inclusive or exclusive of called routines
- Look at all/any category or all states
- Values can be exported/imported
- Tracefiles can be compared

Timeline Display



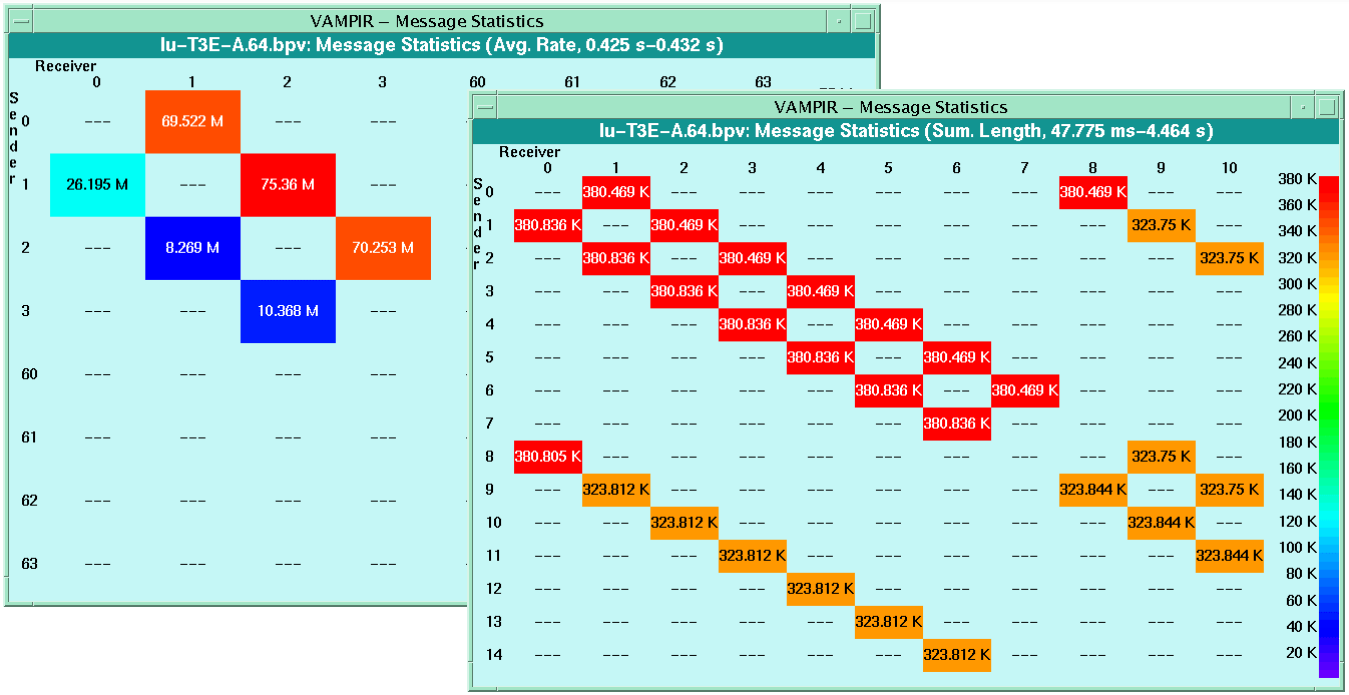
- Now displays MPI collective and I/O operations
- To zoom, draw rectangle with the mouse
- Also used to select sub–intervals for statistics

Timeline Display (Message Info)



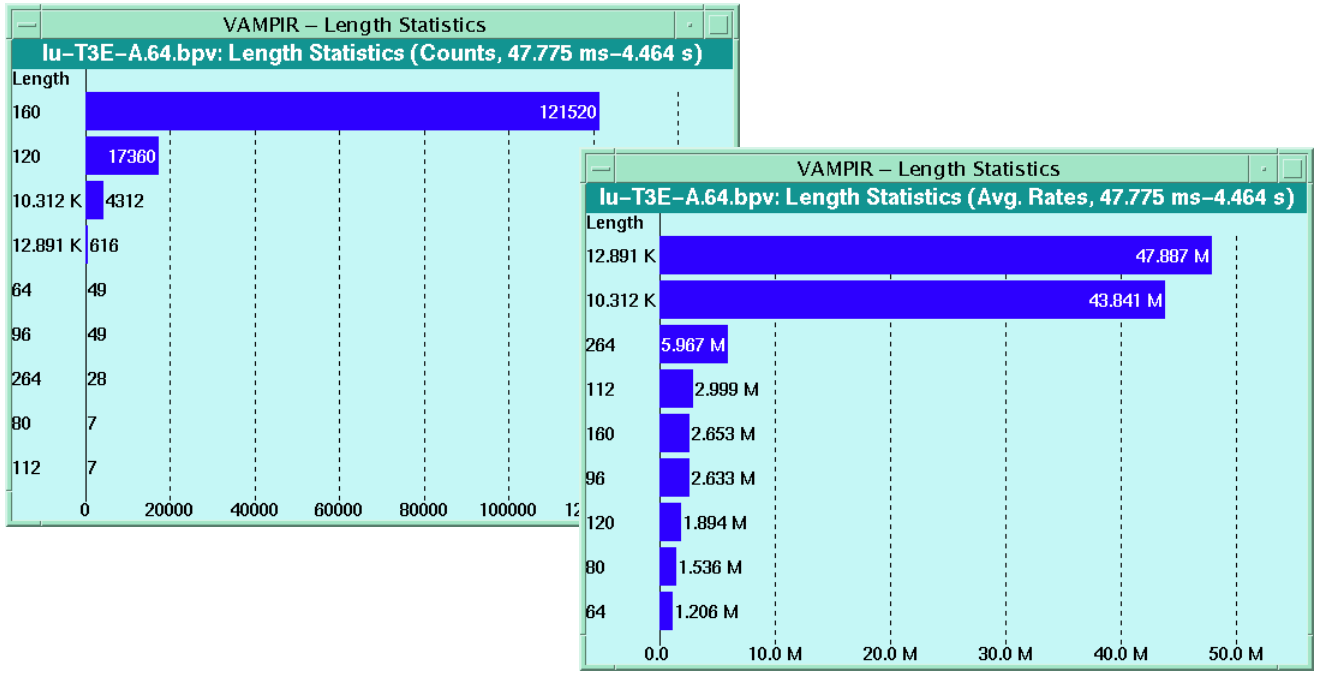
- Source–code references are displayed if recorded by Vampirtrace

Communication Statistics



- Message statistics for each process pair:
  - Byte and message count
  - min/max/avg message length
  - min/max/avg bandwidth
- Filter for message tags or communicators

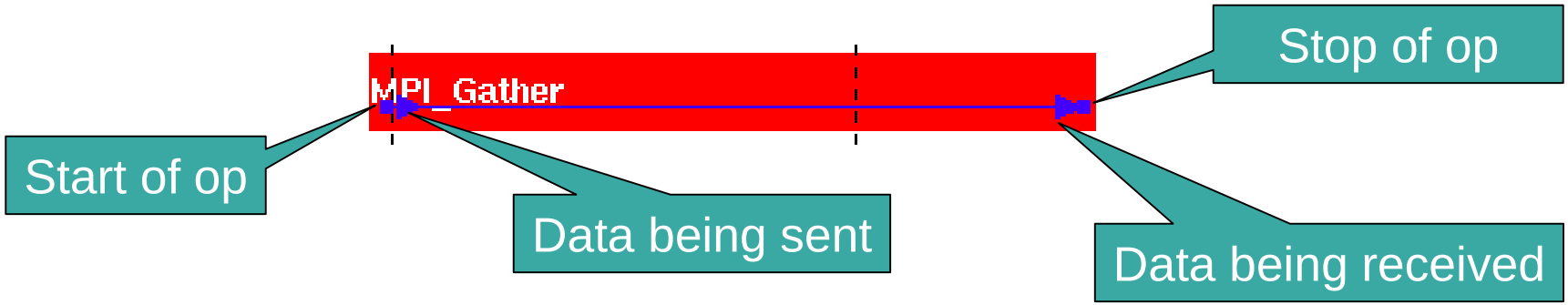
Message Histograms



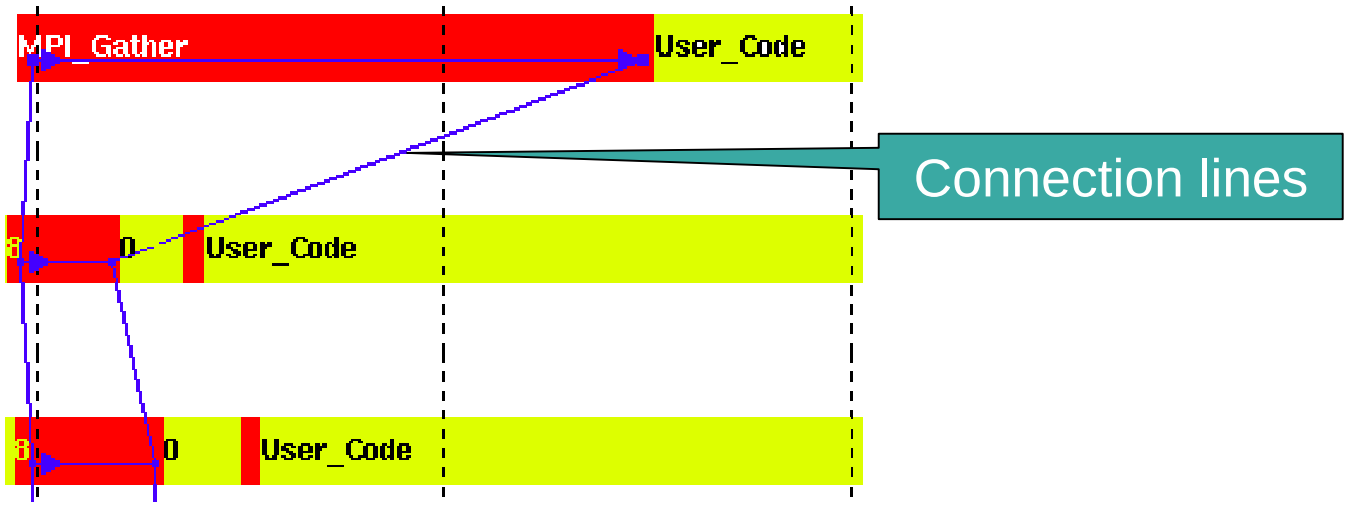
- Message statistics by length, tag or communicator
  - Byte and message count
  - min/max/avg bandwidth
- Filter for message tags or communicators

Collective Operations

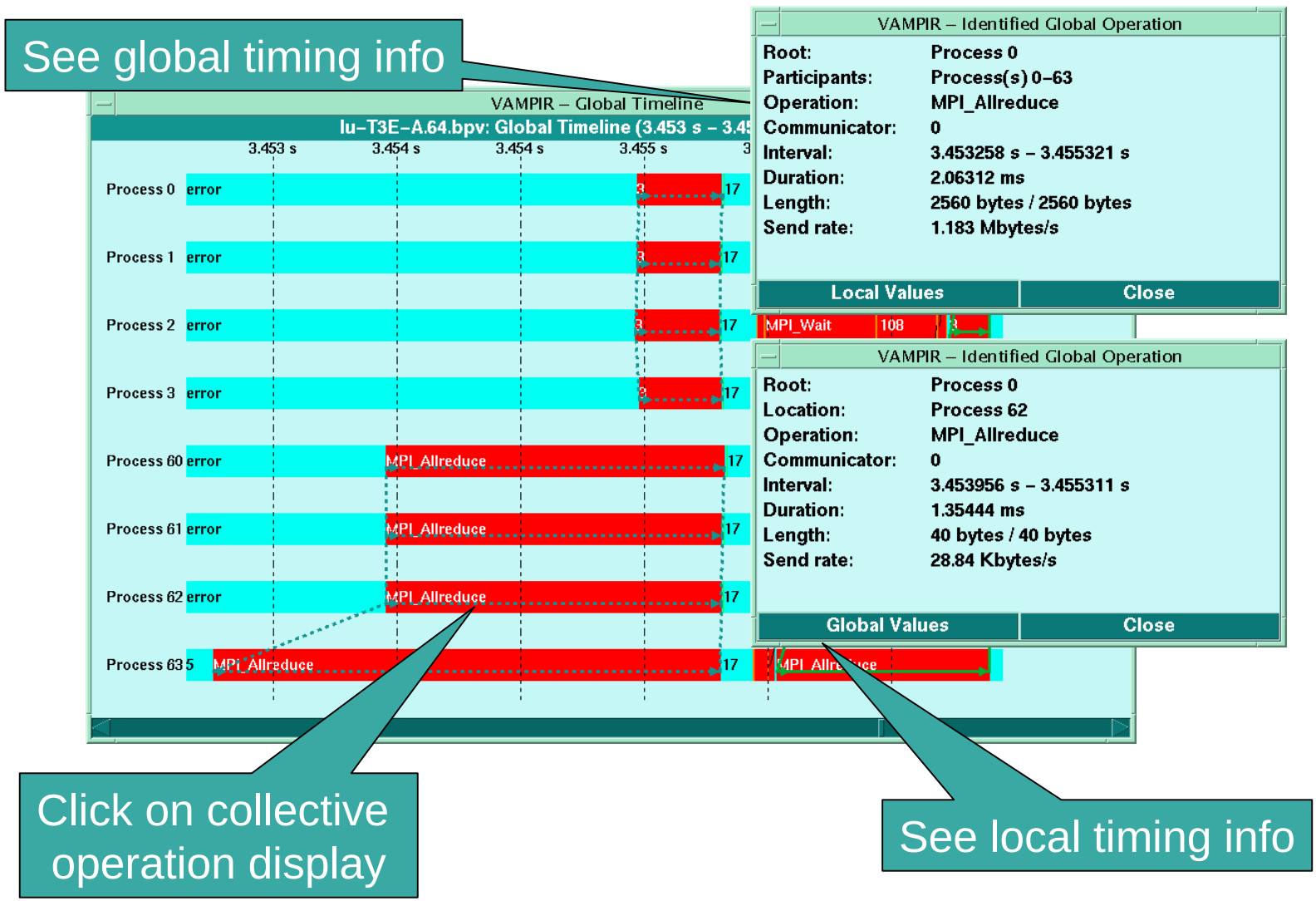
- For each process: mark operation locally



- Connect start/stop points by lines

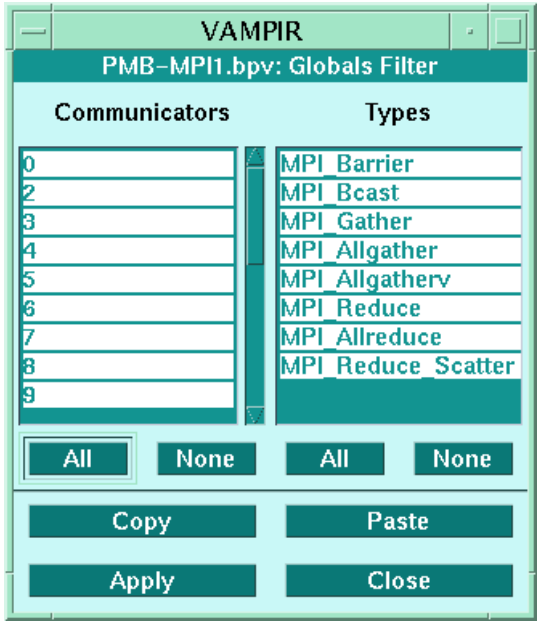


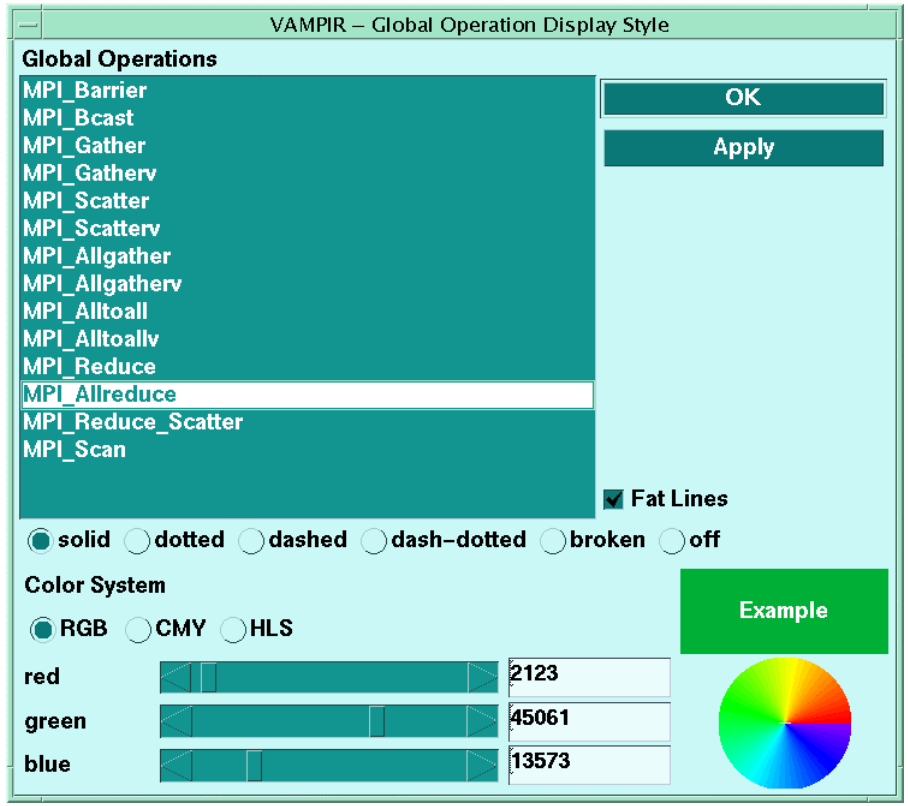
Collective Operations



Collective Operations

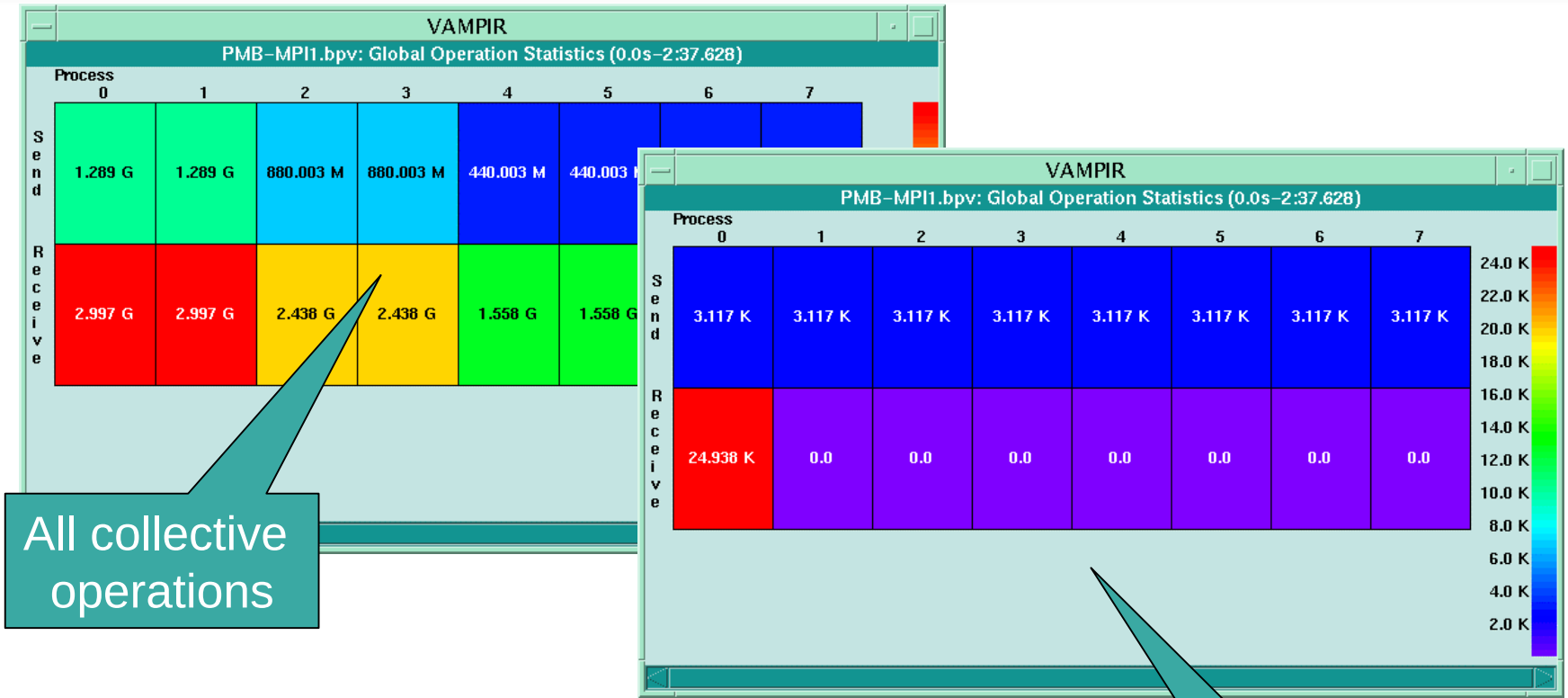
- Collective operations can be filtered





- The display style can be adapted for each collective operation

Global Communication Statistics

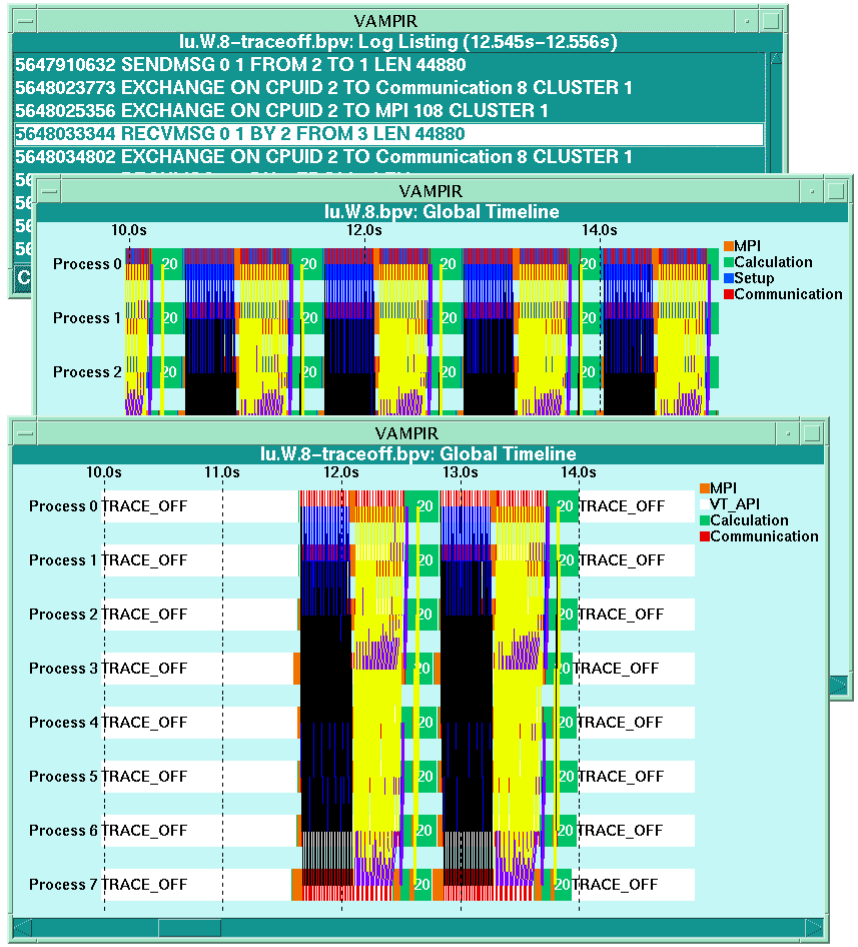


- Statistics for collective operations:
  - operation counts, Bytes sent/received
  - transmission rates
- Filter for collective operation

pallas

Vampirtrace

Tracing of MPI and Application Events

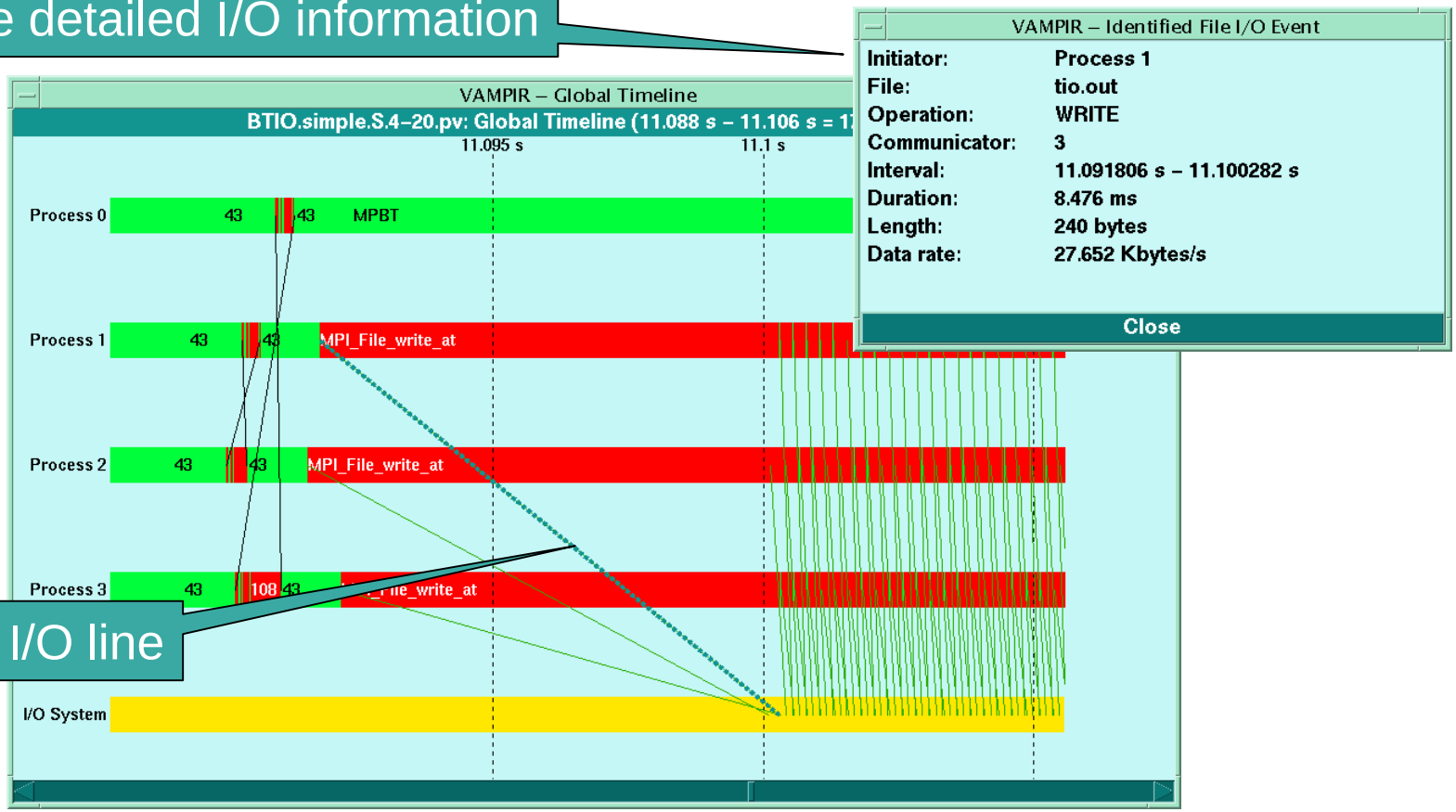


Vampirtrace

- New version: Vampirtrace 2.0
- Significant new features:
  - records collective communication
  - enhanced filter functions
  - extended API
  - records source–code information (selected platforms)
  - support for shmem (Cray T3E)
  - records MPI–2 I/O operations
- Available for all major MPI platforms

MPI–I/O Operations

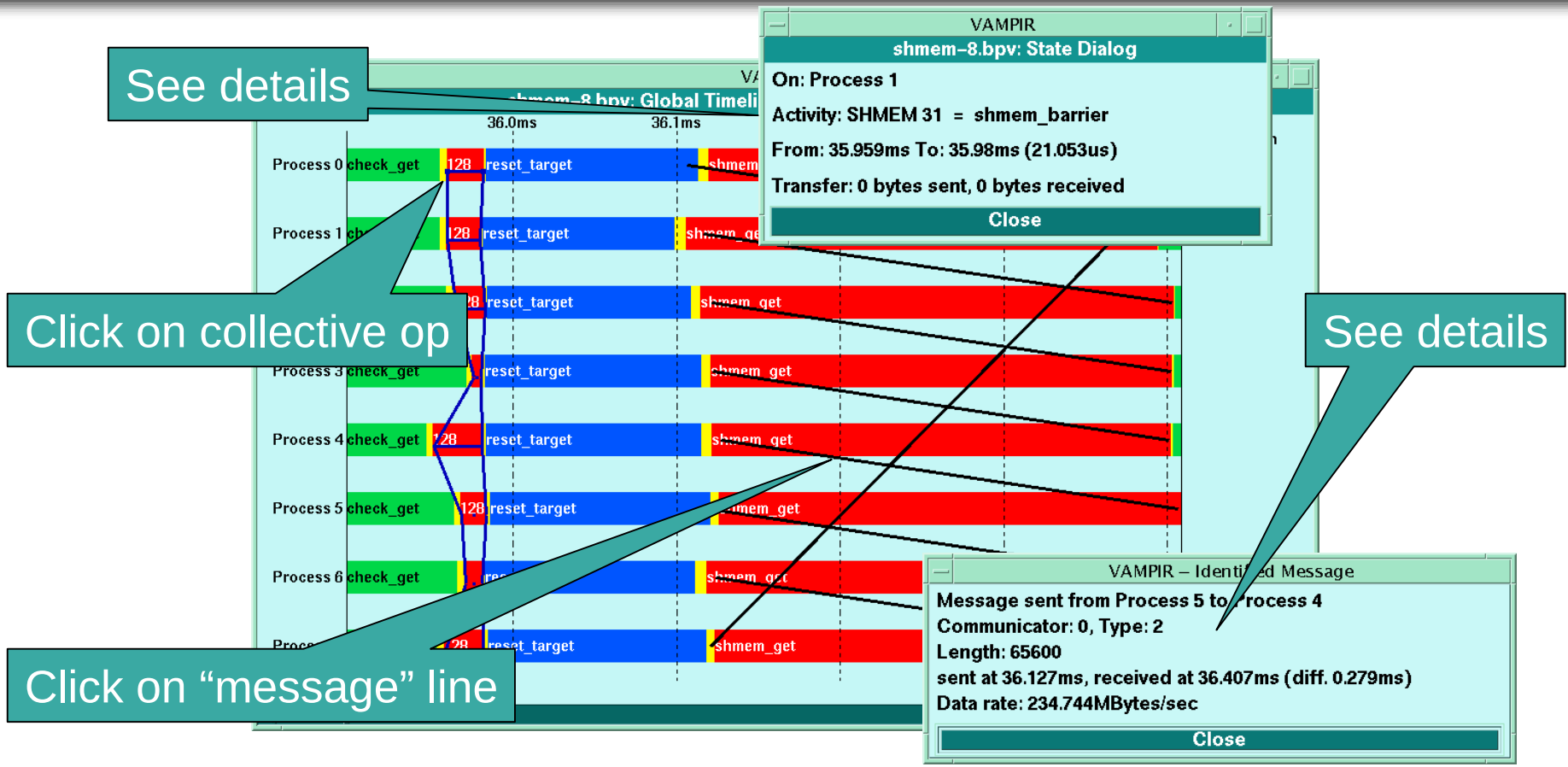
See detailed I/O information



Click on I/O line

- I/O transfers are shown as lines

shmem Operations



- Display one–sided transfers as messages
- Display shmem global operations

Vampir Track Record

- Reference customers: ARL, ARSC, CEWES, LANL, LLNL, MHPCC, NASA, NERSC, NSA, Cornell TC, Oregon Univ., CEA, DWD, ECMWF, GMD, HLRS, LRZ, PC^2, RUKA, ...

- URLs:
  - www.tc.cornell.edu/Edu/Tutor/Vampir
  - www.llnl.gov/sccd/lc/DEG/vampir/vampir.html
  - www.uni-karlsruhe.de/~Vampir
  - www.lrz-muenchen.de/services/software/parallel/vampir
  - www.hlrs.de/structure/support/parallel_computing/tools/performance/vampir.html

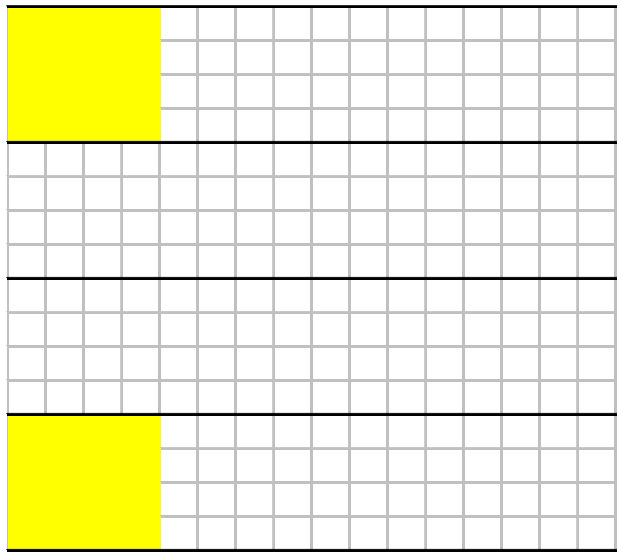
MPI+OpenMP - Example Hybrid Program

- Start simple: only single thread per process calls MPI
- move MPI call outside parallel region or remove OMP barrier, if possible, for optimization

```
...
! We are inside an MPI process
!$OMP PARALLEL
....
!$OMP BARRIER ! to ensure consistent memory
!$OMP MASTER ! cleanly separate OpenMP parallelism from MPI ...
CALL MPI_<some_action>( ... )
!$OMP END MASTER
!$OMP BARRIER ! to ensure consistent memory
....
```

MPI+OpenMP - Example Hybrid Program

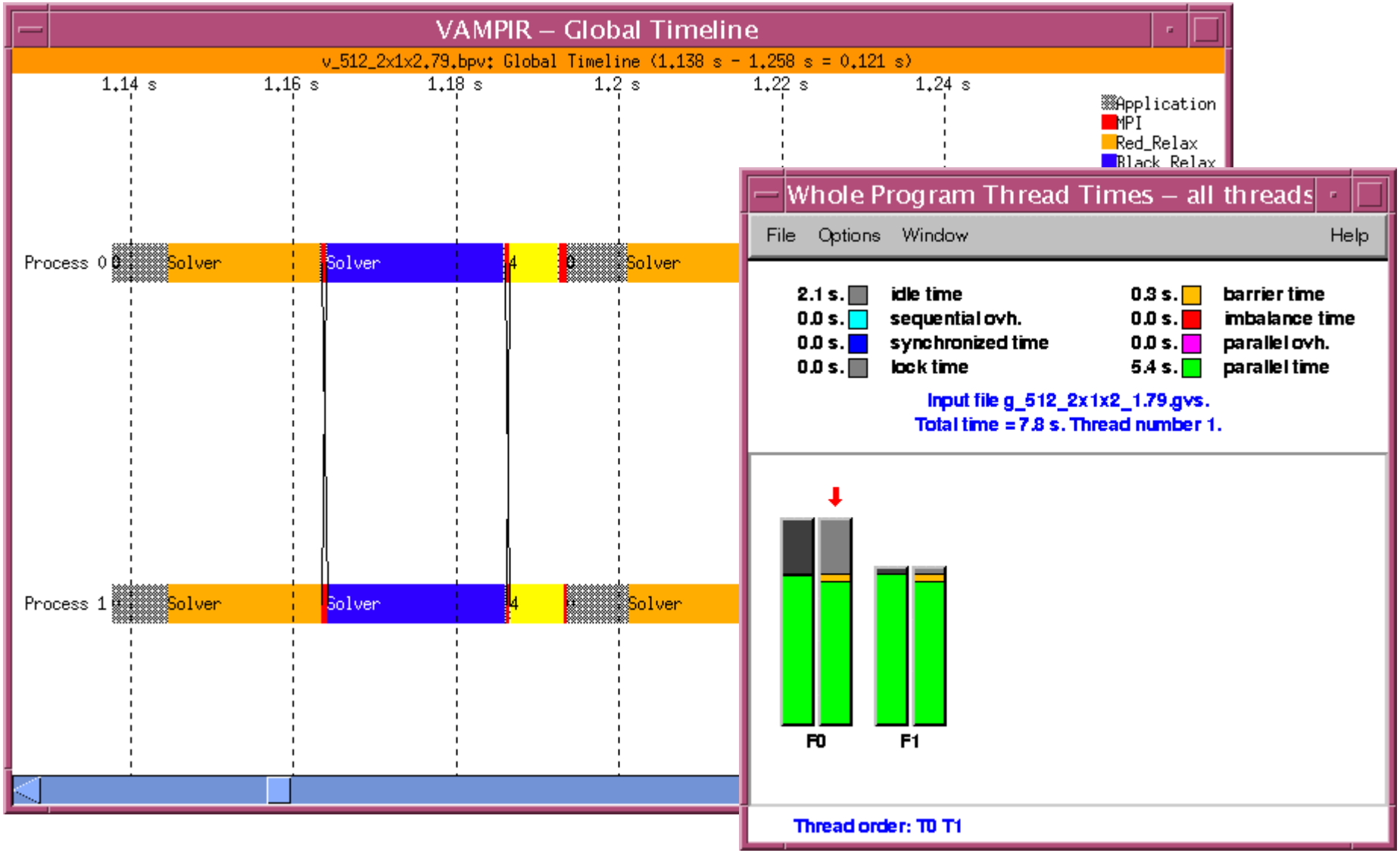
- Parallel Poisson solver in MPI and OpenMP
- certain sections of the grid with more work load
- MPI split not easy, except 2x1 horizontal
- 4 MPI processes run not balanced



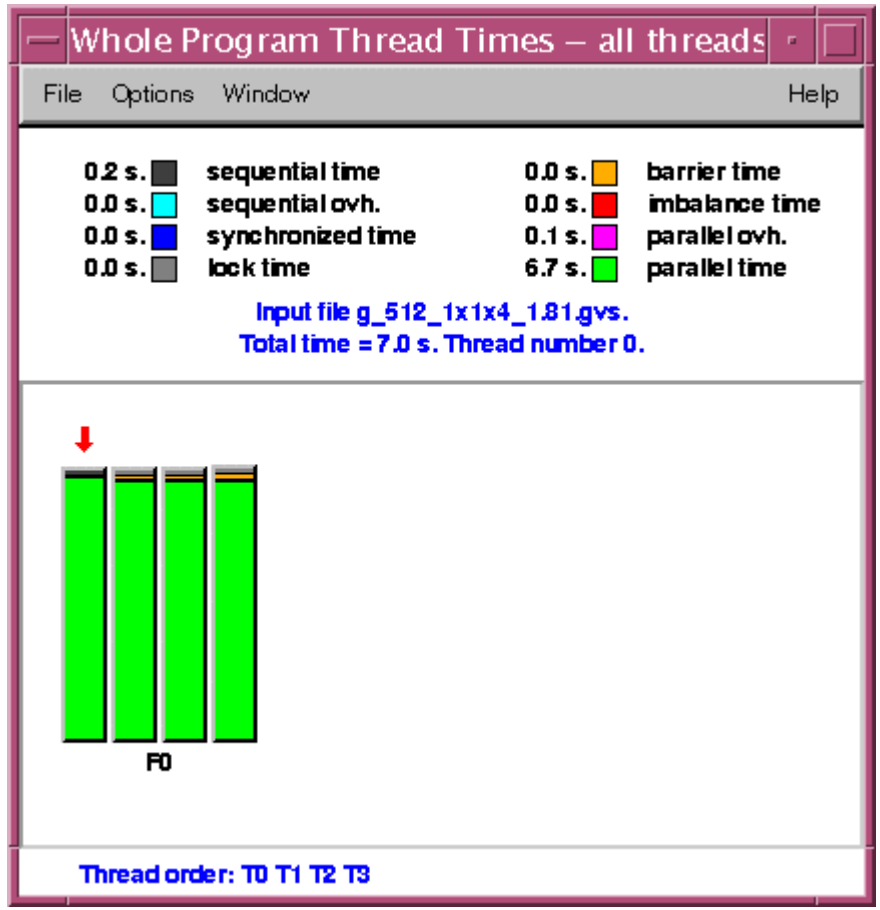
MPI+OpenMP - how to measure quality ?

- Judge quality of parallelization by means of tools:
  - Vampir, GuideView --> performance
  - Assure --> correctness
- Yes, the tools work for hybrid approaches. When running, e.g., a 2 by 2 hybrid model (2 MPI processes, 2 threads per process), one gets
  - one Vampir tracefile and,
  - for each (!) MPI process, an own GuideView profile. GuideView can combine these into one single graph. (--> setenv KMP_STATS_FILE guide_%I)

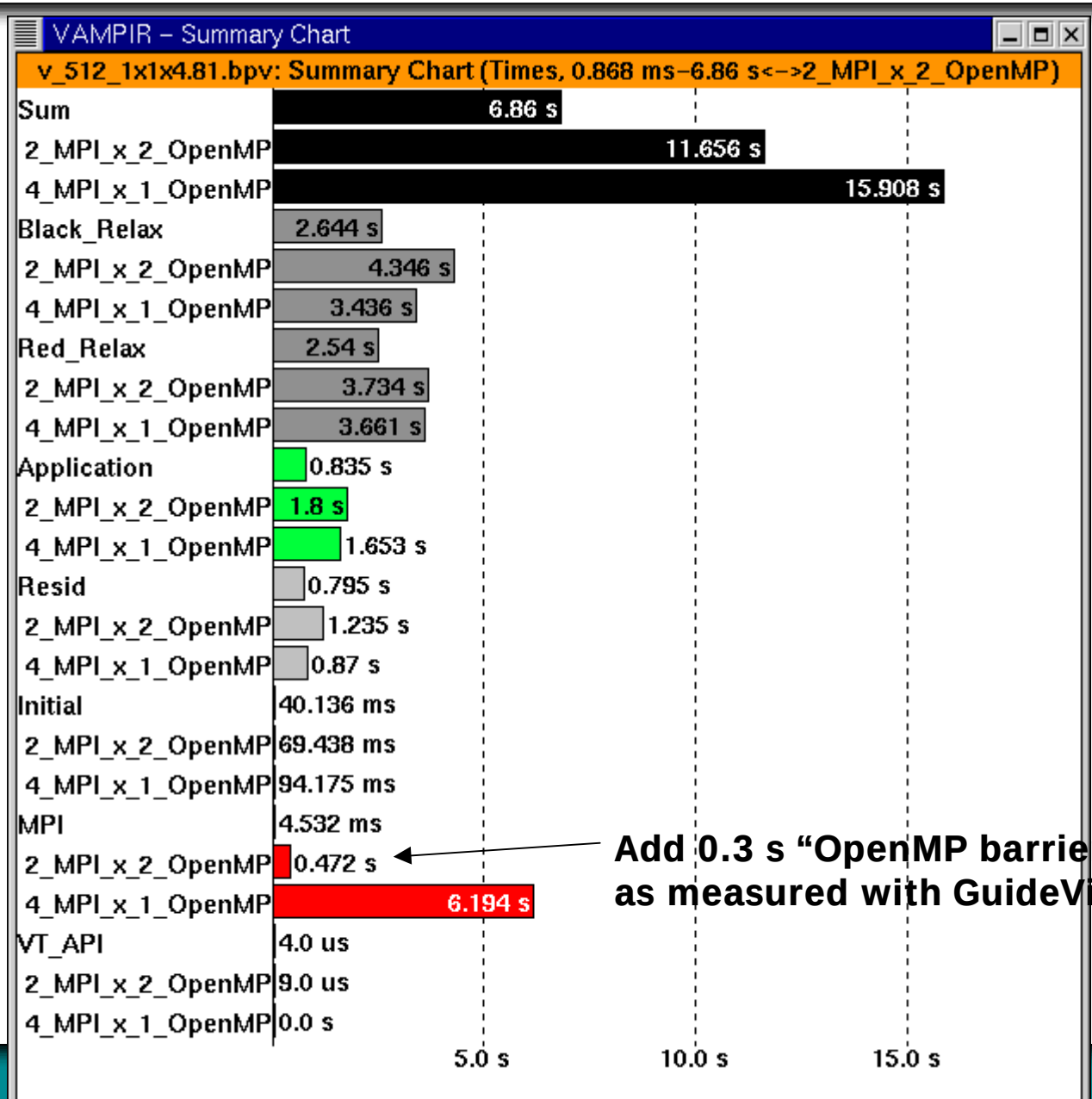
MPI+OpenMP Example - 2 MPI x 2 OpenMP



MPI+OpenMP Example - 1 MPI x 4 OpenMP



MPI+OpenMP Example - Vampir compares the runs



Add 0.3 s “OpenMP barrier-time“ as measured with GuideView

pallas

Future plans - Vampir

- Towards automatic performance analysis
  - improve user guidance in Vampir
  - add “assistant” module for inexperienced users
- Support for clustered shared–memory systems
  - support shared–memory programming models (threads, OpenMP)
  - expose cluster structure
  - aggregate information on SMP nodes
- Support for (very) large systems
  - new structured tracefile format
  - fine–grain interactive control over tracing
  - scalable displays
  - new Vampir structure (can exploit parallelism)

pallas

TotalView

- Symbolic debugging for C, C++, Fortran 77, 90, HPF
- Multi-process debugging for MPI, PVM, distributed processes, HPF, OpenMP, threads
- Fast, easy to learn, easy to use GUI
- Platforms;
  - Linux86, LinuxAlpha, Tru64Alpha, SGI, Sun SPARC, IBM RS6000, IBM SP2, Fujitsu VPP, Cray, NEC SX, NEC Cenju, Hitachi SR, Quadrics CS, Lynx Real-Time OS, CSPI (vxworks)
  - HP port in progress
- *TotalView is the undisputed market leader in multi-processor debugging*

Starting TotalView

- On a new process

  % totalview foo -a *arguments to foo*

- On a core file

  % totalview foo core

- To attach to a running process

  % totalview

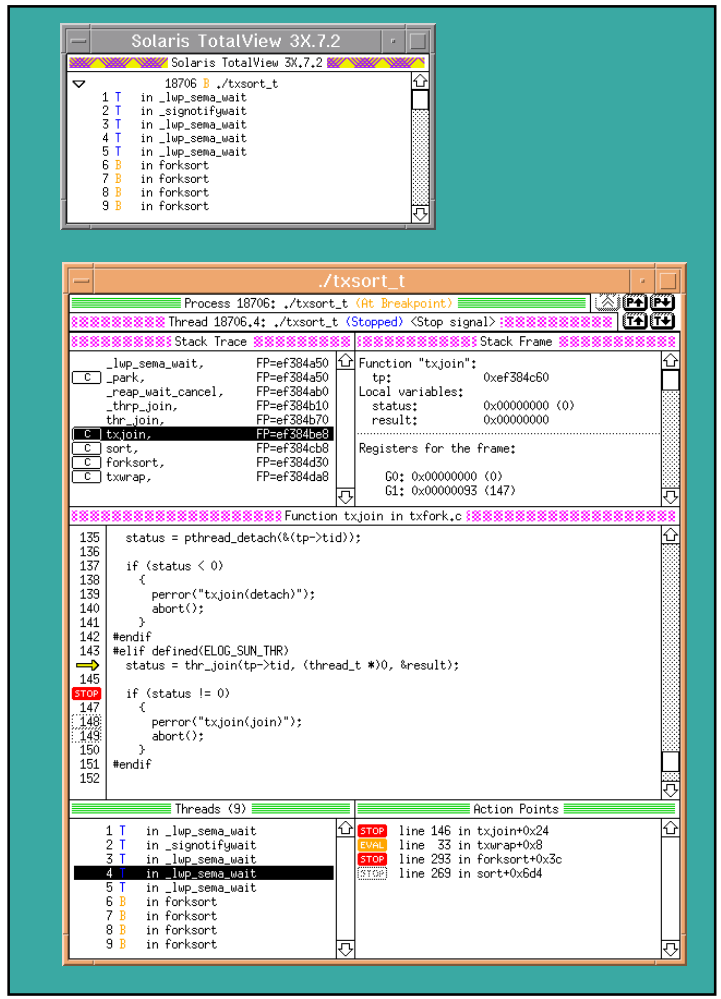
- To debug e.g. MPI/mpich programs

  % mpirun -np 3 -tv foo

  Totalview will start up on the first process and acquire all of the others automatically.

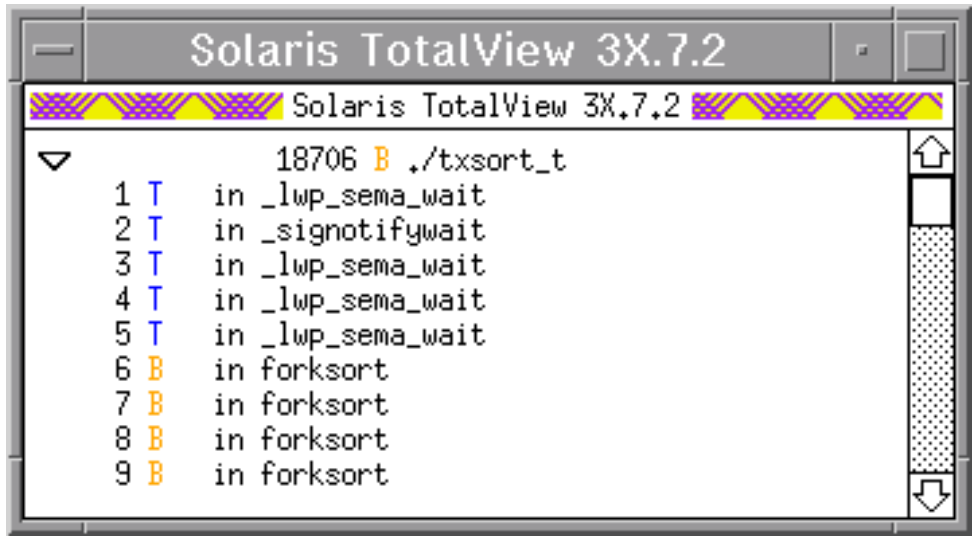
TotalView - Basic Display

- Root Window shows processes under TotalView's control
- Process Window shows information about a single process
- GUI is fast and very "active"
  - single click breakpoints
  - single click "dive"
  - single character accelerators

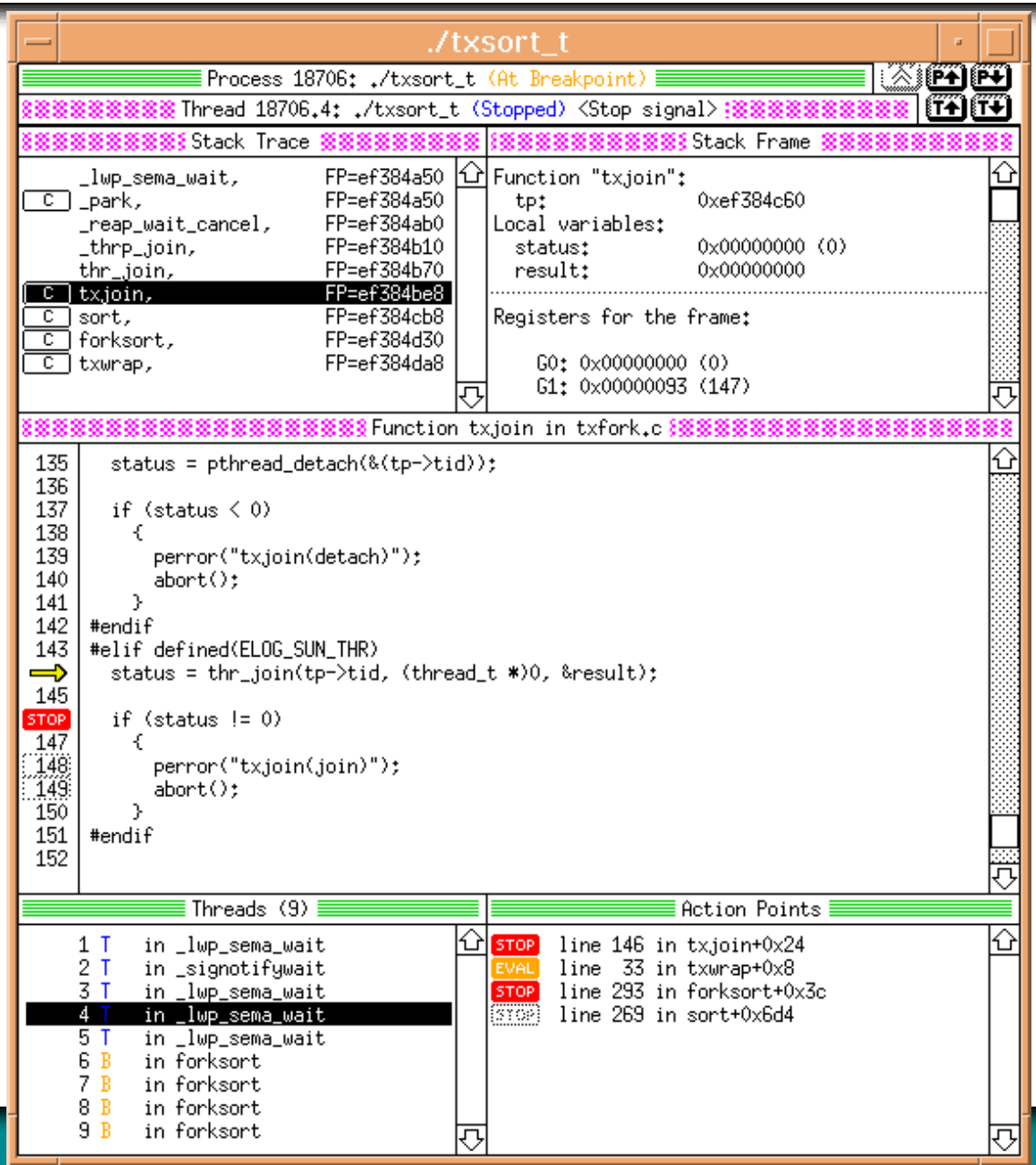


TotalView - Root Window

- Shows processes and threads under TotalView's control
- Toggle allows expansion / collapse of the thread list
- "Diving" on a process or thread opens a new process window
- "Selecting" a process or thread changes the focus of an existing process window
- Machine / node name is listed when processes are distributed
- Colored letters indicate state B=breakpoint, R=running, etc.

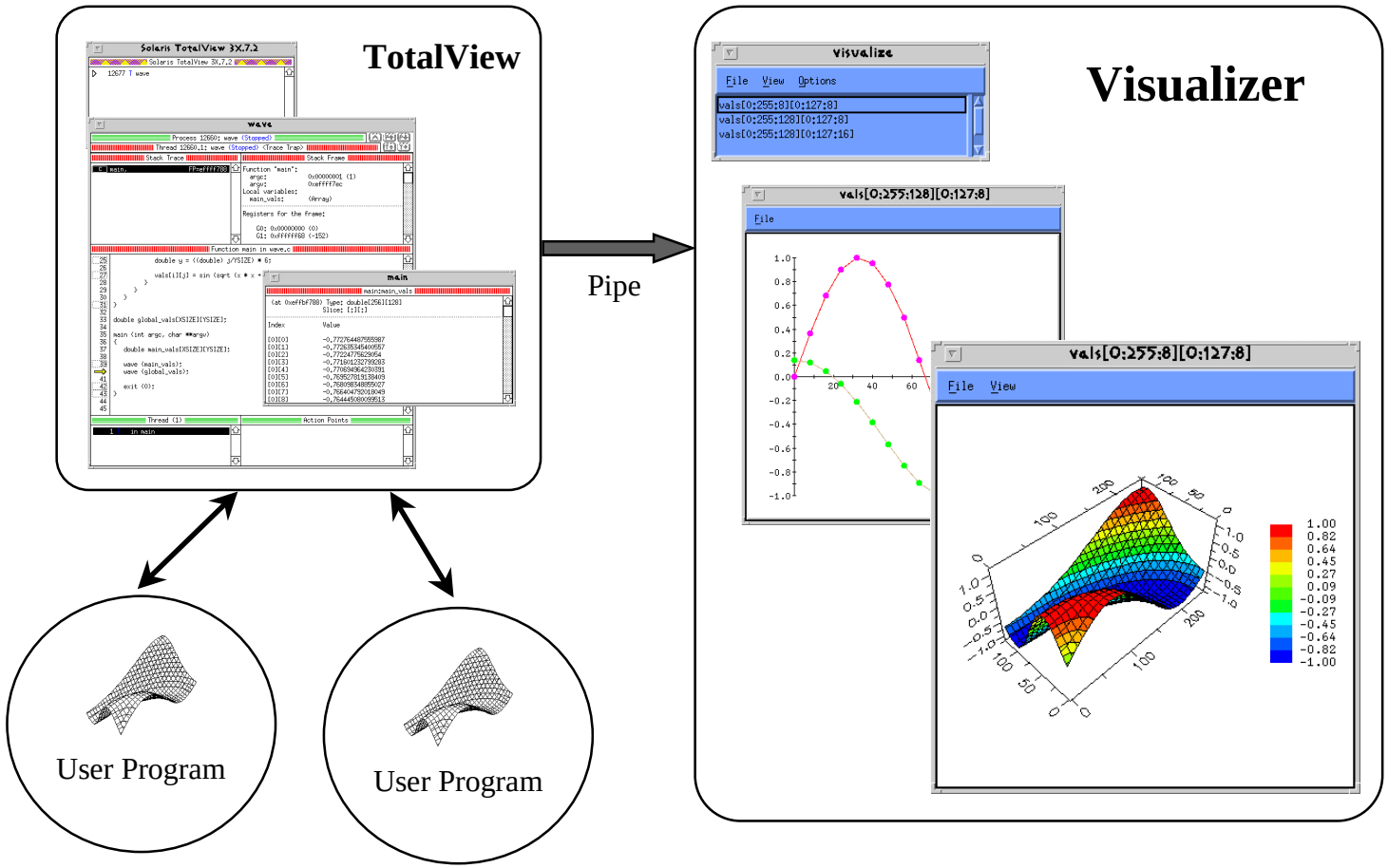


TotalView - Process Window



- Shows information about a single process / thread
  - Stack Trace
  - Registers and Locals
  - Source Code
  - Thread list
  - Action Points
- GUI is fast and very “active”
  - single click breakpoints
  - single click “dive”
  - single character accelerators
- Diving is a fast way to explore
  - variables and structures
  - function names
  - stack frames
  - threads
  - action points
- Values of registers and locals can be changed ‘on screen’

TotalView - Data Visualization



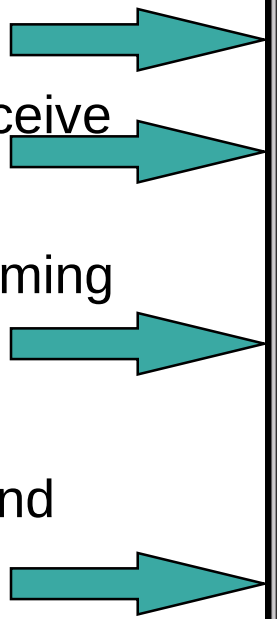
TotalView - MPI Message window

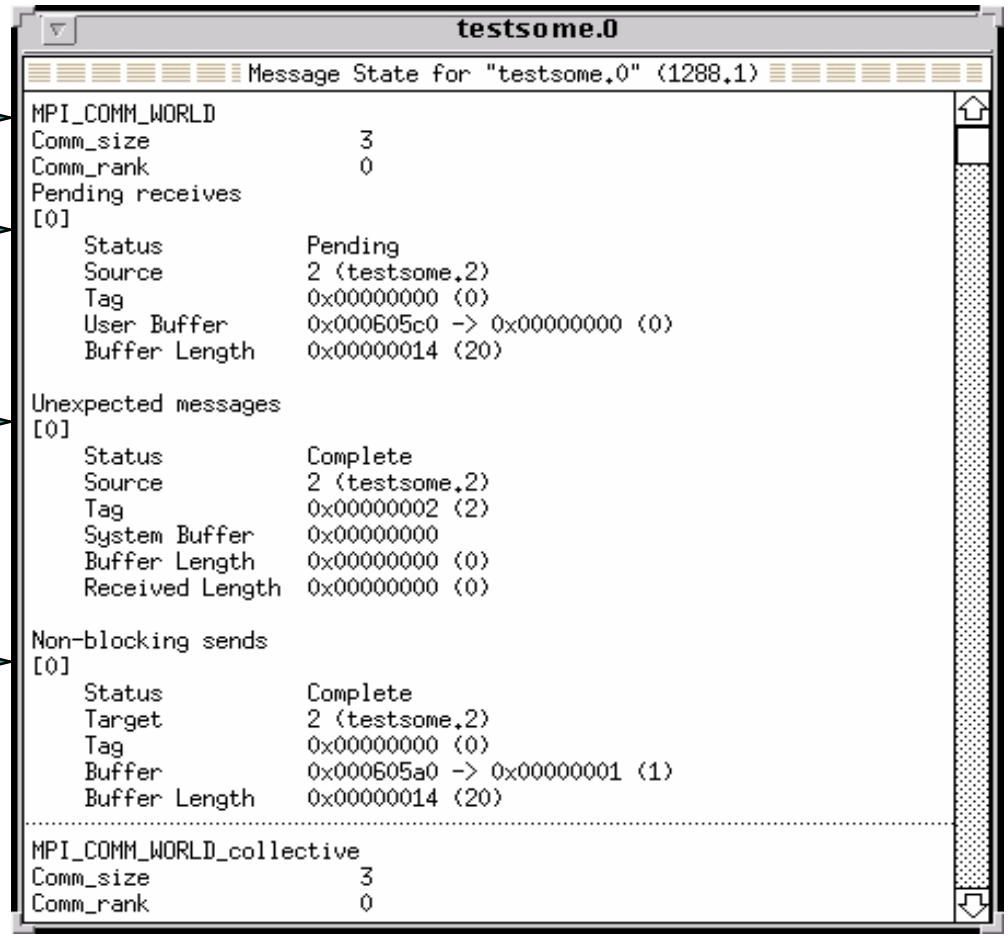
Communicator name and info

Non-blocking receive operations

Unmatched incoming messages

Non-blocking send operations





Dive on source or target to refocus process window.

Dive on buffer to see message contents

pallas

Access to Pallas Tools

Download free evaluation copies

http://www.pallas.com

Thanks for your attention!

Pallas GmbH
Hermülheimer Straße 10
D–50321 Brühl,
Germany

info@pallas.com
www.pallas.com