

# MPI State Profiling

**CScADS Petascale Performance Workshop  
Snowbird, UT, July 21, 2008**

**Marty Itzkowitz**

**Sun Studio Performance Analyzer Team**

Sun Microsystems Inc.

**`marty.itzkowitz@sun.com`**

# Outline

- MPI Model, and Sun Studio **collect**
  - MPI version: Sun's CT 8, based on Open MPI
- MPI State Profiling
- Experiments and Data
  - Functions
  - Callers and Callees
  - Annotated Source
- Correlating the Data
- Pros and Cons

Work by Oleg Mazurov, Eugene Loh, Yukon Maruyama, Terry Dontje, and Marty Itzkowitz

# MPI User-model

- Multiple processes run simultaneously
  - Processes communicate by sending messages
  - May be hybrid: OpenMP and Open MPI
  - May be SIMD or MIMD
- Implementation
  - Processes launched with **mpirun**
    - Either on same node, or across cluster of nodes
  - MPI Data collected with **collect -M on mpirun ...**
    - Insert -- between **mpirun** and its target(s)
  - So-called “founder” experiment has message trace
    - Each process has a “subexperiment” inside founder experiment

# MPI State Profiling

- Three states:
  - Not-in-MPI, MPI-Work, MPI-Wait
  - More states could be added (*e.g.*, wait for specific events)
- API to read state
  - Thread-specific, asynchronous-signal-safe
- Called whenever clock profile signal is received
  - Solaris – no matter what thread is doing
  - Linux – only when thread is on CPU
- Callstack recorded with data

# MPI State Transitions

- Entry to library: Not-in-MPI  $\implies$  MPI-Work
- Start of waiting: MPI-Work  $\implies$  MPI-Wait
  - Whether busy waiting, or giving up the CPU
- End of waiting: MPI-Wait  $\implies$  MPI-Work
- Return from library: MPI-Work  $\implies$  Not-in-MPI

# MPI State Metrics

- If User-CPU and MPI-Work:
  - Increment MPI Work Time
- If not User-CPU, or User-CPU and MPI-Wait:
  - Increment MPI Wait Time
- Metrics attributed to frames in the callstack
  - Map to functions
  - Map to source lines

# MPI Tracing

- Trace calls to MPI library APIs
  - Open Source VampirTrace
  - Metrics
    - MPI-Sends, MPI-Recvs, MPI-Other
    - MPI-bytes-sent, MPI-bytes-recvd
    - MPI-time
  - Callstack recorded with each event
  - Also MPI Timeline, MPI Charts
    - Sends and receives matched in MPI Timeline

# MPI Example: bt .A. 25

- NAS Parallel Benchmark BT
  - Run 25 MPI processes
- **collect -M on mpirun -np 25 ...**
  - Collect MPI API traces from all 25 processes
    - Open Source VampirTrace
  - Collect MPI State data with clock profiles



# Function List

Sun Studio Analyzer [test.3.er, ...]

File View Timeline Help

Find Text: [ ]

MPI Timeline MPI Chart **Functions** Callers-Callees Source Disassembly Timeline Experiments

User CPU (sec.)	MPI (sec.)	MPI Wait (sec.)	MPI Work (sec.)	Name
1765.925	1834.497	1719.353	18.723	<Total>
1765.805	1834.497	1719.313	18.673	_start
1754.257	1834.497	1660.281	18.673	main
1754.197	1834.497	1660.271	18.673	MAIN
1717.001	1191.516	1169.548	17.502	adi_
930.291	1010.668	1002.241	10.027	MPI_WAIT
559.431	413.011	408.766	3.873	y_solve_
500.450	336.305	332.383	4.203	z_solve_
9.937	406.437	328.370	1.111	setup_mpi_
9.917	406.410	328.360	1.111	MPI_INIT
423.336	265.425	261.093	3.943	x_solve_
157.950	177.394	168.898	4.963	MPI_WAITALL
176.463	178.398	168.898	5.514	copy_faces_
6.304	222.322	148.184	0.010	MPI_FINALIZE
11.538	0.	59.031	0.	_exithandle
11.538	0.	59.031	0.	exit
11.528	0.	59.021	0.	vt_close
6.545	6.974	6.935	0.040	verify_
4.303	6.088	6.074	0.	MPI_BCAST
4.533	5.351	5.344	0.010	MPI_ALLREDUCE
3.743	4.494	4.483	0.	rhs norm

Summary Event MPI Chart Controls MPI Timel...

**Selected Object:**

Name: copy\_faces\_  
 PC Address: 2:0x0000B8C0  
 Size: 8880  
 Source File: BT/copy\_faces.f  
 Object File: bin/bt.A.25  
 Load Object: <bt.A.25>  
 Mangled Name:  
 Aliases:

**Metrics for Selected Object:**

	Exclusive	
User CPU:	16.081 ( 0.91%)	176.
Wall:	17.272 ( 0.66%)	198.
Total Thread:	17.272 ( 0.66%)	198.
System CPU:	0.020 ( 0.11%)	1.
Wait CPU:	0. ( 0. %)	0.
User Lock:	0. ( 0. %)	0.
Text Page Fault:	0. ( 0. %)	0.
Data Page Fault:	0. ( 0. %)	2.
Other Wait:	1.171 ( 0.14%)	17.
MPI:	0. ( 0. %)	178.

Sorted by MPI Wait time

# Caller-Callees

Sun Studio Analyzer [test.3.er, ...]

File View Timeline Help

Find Text:

MPI Timeline MPI Chart Functions Callers-Callees Source Disassembly Timeline Experiments

User CPU (sec.)	User CPU (sec.)	MPI (sec.)	MPI (sec.)	MPI Wait (sec.)	MPI Wait (sec.)	MPI Work (sec.)	MPI Work (sec.)	Name
174.782	1 717.001	176.775	1 191.516	167.307	1 169.548	5.484	17.502	adi_
1.681	6.545	1.623	6.974	1.591	6.935	0.030	0.040	verify_

User CPU	User CPU	MPI	MPI	MPI Wait	MPI Wait	MPI Work	MPI Work	Name
16.081	176.463	0.	178.398	0.	168.898	0.	5.514	copy_faces
157.950	157.950	177.394	177.394	168.898	168.898	4.963	4.963	MPI_WAITALL
1.341	8.166	0.625	3.467	0.	0.	0.360	1.751	MPI_ISEND
1.081	5.514	0.378	1.610	0.	0.	0.190	0.791	MPI_Irecv
0.010	0.190	0.	0.	0.	0.050	0.	0.050	do_exit_critical

Summary Event MPI Chart Controls MPI Timel...

**Selected Object:**

Name: copy\_faces\_  
 PC Address: 2:0x0000B8C0  
 Size: 8880  
 Source File: BT/copy\_faces.f  
 Object File: bin/bt.A.25  
 Load Object: <bt.A.25>  
 Mangled Name:  
 Aliases:

**Metrics for Selected Object:**

	Exclusive	
User CPU:	16.081 ( 0.91%)	176.
Wall:	17.272 ( 0.66%)	198.
Total Thread:	17.272 ( 0.66%)	198.
System CPU:	0.020 ( 0.11%)	1.
Wait CPU:	0. ( 0. %)	0.
User Lock:	0. ( 0. %)	0.
Text Page Fault:	0. ( 0. %)	0.
Data Page Fault:	0. ( 0. %)	2.
Other Wait:	1.171 ( 0.14%)	17.
MPI:	0. ( 0. %)	178.

Navigate with MPI Wait time to find delays

# Source Display

Sun Studio Analyzer [test.3.er, ...]

File View Timeline Help

Find Text:

MPI Timeline MPI Chart Functions Callers-Callees **Source** Disassembly Timeline Experiments

User CPU (sec.)	MPI (sec.)	MPI Wait (sec.)	MPI Work (sec.)	Source File: BT/copy_faces.f
0.210	0.081	0.	0.080	Function mpi_isend_ not inlined because the compiler has n 202. call mpi_isend(out_buffer(ss(5)), b_size(5), 203. > dp_type,predecessor(3), BOTTOM, 204. > comm_rhs,requests(11), error) 205. 206.
157.960	177.394	168.898	4.963	Function mpi_waitall_ not inlined because the compiler has 207. call mpi_waitall(12, requests, statuses, error 208. 209. c----- 210. c unpack the data that has just been received; 211. c----- 212. p0 = 0 213. p1 = 0 214. p2 = 0 215. p3 = 0 216. p4 = 0 217. p5 = 0

Summary Event MPI Chart Controls MPI Time...

**Selected Object:**

Name: line 4 in "copy\_faces.f"

PC Address: 2:0x0000B8C0

Size: 0

Source File: BT/copy\_faces.f

Object File: bin/bt.A.25

Load Object: <bt.A.25>

Mangled Name:

Aliases:

**Metrics for Selected Object:**

	Exclusive	
User CPU:	0. ( 0. %)	0.
Wall:	0. ( 0. %)	0.
Total Thread:	0. ( 0. %)	0.
System CPU:	0. ( 0. %)	0.
Wait CPU:	0. ( 0. %)	0.
User Lock:	0. ( 0. %)	0.
Text Page Fault:	0. ( 0. %)	0.
Data Page Fault:	0. ( 0. %)	0.
Other Wait:	0. ( 0. %)	0.

Find lines where MPI Wait time is high

# MPI Example: is.B.16

- NAS Parallel Benchmark IS
  - Run 16 MPI processes
  - **collect -M on mpirun -np 16 ...**
  - Collect MPI API traces from all sixteen processes
    - Open Source Vampir Trace
  - Collect MPI State data with clock profiles

# Function List

Sun Studio Analyzer [test.2.er, ...]

File View Timeline Help

Find Text:

MPI Timeline MPI Chart Functions Callers-Callees Source Disassembly Timeline Experiments

User CPU (sec.)	User CPU (sec.)	MPI Sends	MPI Receives	MPI Wait (sec.)	Name
87.952	87.952	15	15	303.793	<Total>
0.	87.932	15	15	303.793	_start
9.276	87.631	15	15	294.076	main
0.	6.174	0	0	222.956	MPI_Init
0.	2.462	0	0	46.763	MPI_Finalize
35.935	64.195	0	0	22.776	rank
0.010	12.309	0	0	12.609	MPI_Allreduce
0.	0.300	0	0	9.717	_exithandle
0.	0.300	0	0	9.717	exit
0.	0.290	0	0	9.707	vt_close
0.	14.940	0	0	9.116	MPI_Alltoallv
0.	1.011	0	0	1.051	MPI_Alltoall
0.	0.871	0	0	0.871	MPI_Reduce
0.	0.520	0	15	0.710	MPI_Wait
4.123	4.643	15	15	0.710	full_verify
0.	0.030	0	0	0.010	_ti_bind_clear
0.	0.010	0	0	0.010	atexit_fini
0.	0.010	0	0	0.010	call_fini
0.	0.030	0	0	0.010	do_exit_critical
0.030	0.030	0	0	0.010	take_deferred_signal
0.	0.	0	0	0.	MPI_Irecv

Summary Event MPI Chart Controls MPI Timel...

**Selected Object:**

Name: rank  
 PC Address: 2:0x000022C0  
 Size: 4600  
 Source File: IS/is.c  
 Object File: bin/is.B.16  
 Load Object: <is.B.16>  
 Mangled Name:  
 Aliases:

**Metrics for Selected Object:**

	Exclusive	Inclusive
User CPU:	35.935 (40.86%)	64.195 (72.73%)
Wall:	43.681 (8.74%)	82.738 (16.66%)
Total Thread:	43.681 (8.74%)	82.738 (16.66%)
System CPU:	0.370 (4.47%)	3.182 (38.54%)
Wait CPU:	0. (0.%)	0.280 (73.68%)
User Lock:	0. (0.%)	0. (0.%)
Text Page Fault:	0. (0.%)	0. (0.%)
Data Page Fault:	0.020 (0.61%)	3.112 (95.45%)
Other Wait:	7.355 (1.84%)	11.968 (29.42%)
MPI:	0. (0.%)	28.040 (70.00%)

Sorted by MPI Wait time – significant wait time in  
 MPI\_Alltoallv, MPI\_Alltoall, MPI\_Allreduce  
 Also, in MPI\_Init and MPI\_Finalize

# Caller-Callees

Sun Studio Analyzer [test.2.er, ...]

File View Timeline Help

Find Text:

MPI Timeline MPI Chart Functions Callers-Callees Source Disassembly Timeline Experiments

User CPU (sec.)	User CPU (sec.)	User CPU (sec.)	MPI Sends (sec.)	MPI Sends (sec.)	MPI Receives (sec.)	MPI Receives (sec.)	MPI Wait (sec.)	MPI Wait (sec.)	Name
64.195	9.276	87.631	0	15	0	15	22.776	294.076	main
35.935	35.935	64.195	0	0	0	0	22.776		rank
12.309	0.010	12.309	0	0	0	0	12.609	12.609	MPI_Allreduce
14.940	0.	14.940	0	0	0	0	9.116	9.116	MPI_Alltoallv
1.011	0.	1.011	0	0	0	0	1.051	1.051	MPI_Alltoall

Summary Event MPI Chart Controls MPI Timel...

**Name:** rank  
**PC Address:** 2:0x000022C0  
**Size:** 4600  
**Source File:** IS/is.c  
**Object File:** bin/is.B.16  
**Load Object:** <is.B.16>  
**Mangled Name:**  
**Aliases:**

**Metrics for Selected Object:**

	Exclusive	Inclusive
<b>User CPU:</b>	35.935 (40.86%)	64.195 (72.7%)
<b>Wait:</b>	43.681 (8.74%)	82.738 (16.4%)
<b>Total Thread:</b>	43.681 (8.74%)	82.738 (16.4%)
<b>System CPU:</b>	0.370 (4.47%)	3.182 (38.5%)
<b>Wait CPU:</b>	0. (0.%)	0.280 (73.3%)
<b>User Lock:</b>	0. (0.%)	0. (0.%)
<b>Text Page Fault:</b>	0. (0.%)	0. (0.%)
<b>Data Page Fault:</b>	0.020 (0.61%)	3.112 (95.3%)
<b>Other Wait:</b>	7.355 (1.84%)	11.968 (28.7%)
<b>MPI:</b>	0. (0.%)	38.949 (95.3%)
<b>MPI Bytes Sent:</b>	0 (0.%)	0 (0.%)

Navigate with MPI Wait time to find delays

# Source Display, I

Sun Studio Analyzer [test.2.er, ...]

File View Timeline Help

Find Text:

MPI Timeline MPI Chart Functions Callers-Callees **Source** Disassembly Timeline Experiments

User CPU (sec.)	User CPU (sec.)	MPI Sends	MPI Receives	MPI Wait (sec.)	Source File: IS/is.c
					Object File: bin/is.B.16
					Load Object: <is.B.16>
					565. )
					566.
					567. #ifdef TIMING_ENABLED
					568. timer_stop( 2 );
					569. timer_start( 3 );
					570. #endif
					571.
					572. /* Get the bucket size totals for the entire p
					573. will be used to determine the redistributio
					574. MPI_Allreduce( bucket_size,
					575. bucket_size_totals,
					576. NUM_BUCKETS+TEST_ARRAY_SIZE,
					577. MPI_INT,
					578. MPI_SUM,
					Function MPI_Allreduce not inlined because the compile
0.	12.309	0	0	12.609	579. MPI_COMM_WORLD );
					580.
					581. #ifdef TIMING_ENABLED
					582. timer_stop( 3 );

Summary Event MPI Chart Controls MPI Time...

**Name:** line 472 in "is.c"

**PC Address:** 2:0x0000207C

**Size:** 0

**Source File:** IS/is.c

**Object File:** bin/is.B.16

**Load Object:** <is.B.16>

**Mangled Name:**

**Aliases:**

**Metrics for Selected Object:**

	Exclusive	Inclusive
<b>User CPU:</b>	0. ( 0. %)	0.520 ( 0. %)
<b>Wall:</b>	0. ( 0. %)	0.721 ( 0. %)
<b>Total Thread:</b>	0. ( 0. %)	0.721 ( 0. %)
<b>System CPU:</b>	0. ( 0. %)	0.010 ( 0. %)
<b>Wait CPU:</b>	0. ( 0. %)	0. ( 0. %)
<b>User Lock:</b>	0. ( 0. %)	0. ( 0. %)
<b>Text Page Fault:</b>	0. ( 0. %)	0. ( 0. %)
<b>Data Page Fault:</b>	0. ( 0. %)	0. ( 0. %)
<b>Other Wait:</b>	0. ( 0. %)	0.190 ( 0. %)
<b>MPI:</b>	0. ( 0. %)	0.711 ( 0. %)
<b>MPI Bytes Sent:</b>	0 ( 0. %)	0 ( 0. %)

Find lines where MPI Wait time is high:  
I. MPI\_Allreduce()

# Source Display, II

Sun Studio Analyzer [test.2.er, ...]

File View Timeline Help

Find Text: Alltoally

MPI Timeline MPI Chart Functions Callers-Callees **Source** Disassembly Timeline Experiments

User CPU (sec.)	User CPU (sec.)	MPI Sends	MPI Receives	MPI Wait (sec.)	Source File: IS/is.c
					Object File: bin/is.B.16
					Load Object: <is.B.16>
					644. /* Now send the keys to respective processors
					645. MPI_Alltoallv( key_buff1,
					646. send_count,
					647. send_displ,
					648. MPI_INT,
					649. key_buff2,
					650. recv_count,
					651. recv_displ,
					652. MPI_INT,
0.	14.940	0	0	9.116	Function MPI_Alltoally not inlined because the compiler
					653. MPI_COMM_WORLD );
					654.
					655. #ifdef TIMING_ENABLED
					656. timer_stop( 3 );
					657. timer_start( 2 );
					658. #endif
					659.
					660. /* The starting and ending bucket numbers on e
					661. multiplied by the interval size of the buck

Summary Event MPI Chart Controls MPI Timel...

**Name:** line 508 in "is.c"

**PC Address:** 2:0x000022C0

**Size:** 0

**Source File:** IS/is.c

**Object File:** bin/is.B.16

**Load Object:** <is.B.16>

**Mangled Name:**

**Aliases:**

**Metrics for Selected Object:**

	Exclusive	Inclusive
<b>User CPU:</b>	0. ( 0. %)	0. ( 0. %)
<b>Wait:</b>	0. ( 0. %)	0. ( 0. %)
<b>Total Thread:</b>	0. ( 0. %)	0. ( 0. %)
<b>System CPU:</b>	0. ( 0. %)	0. ( 0. %)
<b>Wait CPU:</b>	0. ( 0. %)	0. ( 0. %)
<b>User Lock:</b>	0. ( 0. %)	0. ( 0. %)
<b>Text Page Fault:</b>	0. ( 0. %)	0. ( 0. %)
<b>Data Page Fault:</b>	0. ( 0. %)	0. ( 0. %)
<b>Other Wait:</b>	0. ( 0. %)	0. ( 0. %)
<b>MPI:</b>	0. ( 0. %)	0. ( 0. %)
<b>MPI Bytes Sent:</b>	0 ( 0. %)	0 ( 0. %)

Find lines where MPI Wait time is high:  
2. MPI\_Alltoallv()



# Correlating the Data

- All trace and profile records have callstacks
  - Suppress frames below API-layer of MPI
- Aggregate Messages into Transactions
  - Based on common send/recv callstacks
- Profile records with non-zero MPI-Wait / MPI Work time
  - Must be inside MPI
  - Will match a send or recv trace record callstack
- Attribute MPI-Wait / MPI-Work time to Transactions
  - Both sender and receiver times
  - Select expensive transactions to display

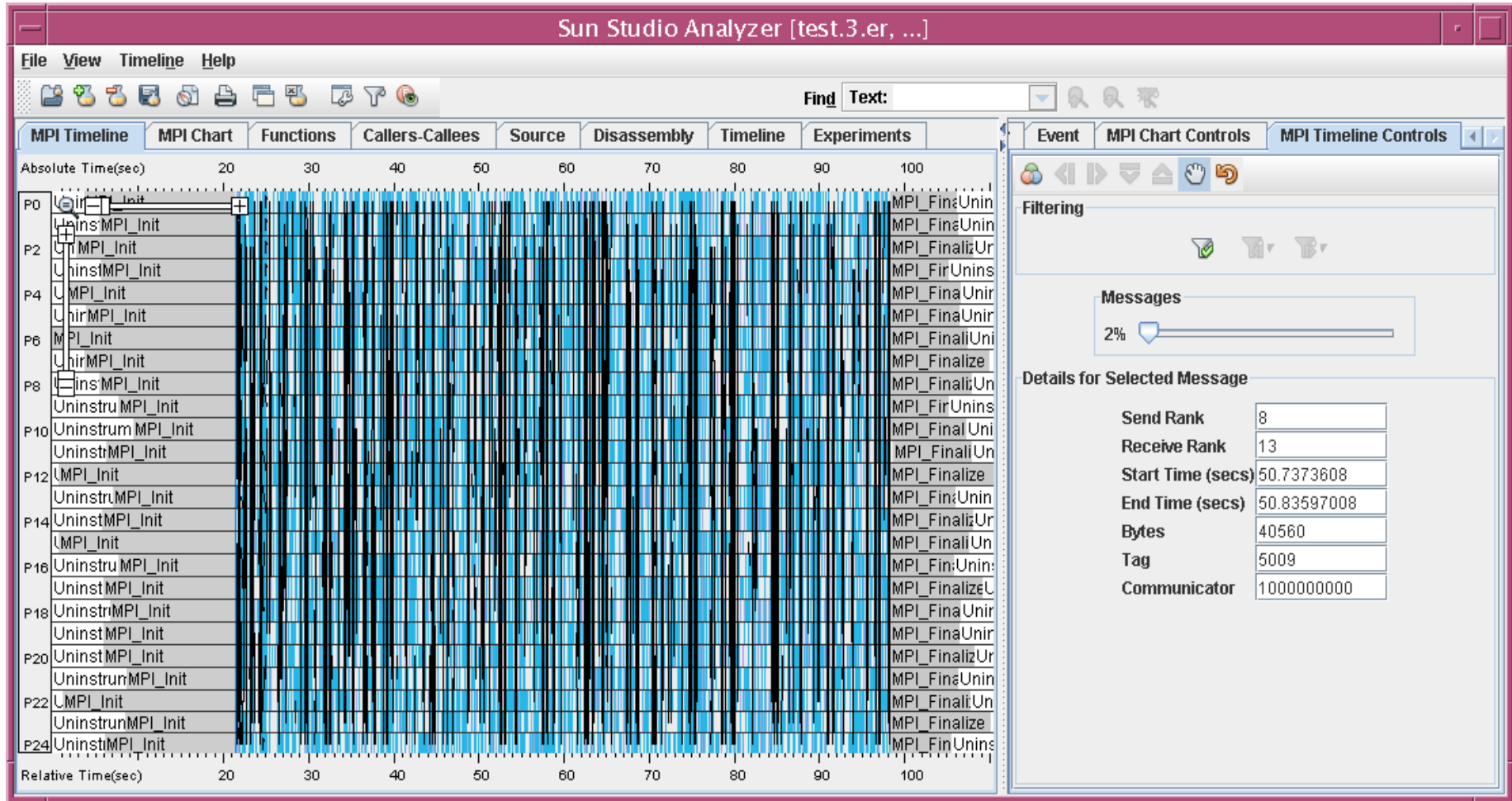
# Advantages and Disadvantages

- Advantages:
  - Easy implementation in Open MPI libraries
  - Scalability: profiling allows easy control of data volume
  - Direct navigation to source: no hunting through messages
- Disadvantages
  - Data is statistical, may miss anomalies
  - Dilation (in current prototype)
    - ~ 0.05 – 0.15  $\mu$ secs. in latency for small (0-, 8-byte) messages
- We will offer source to Open MPI Community

# MPI State Profiling

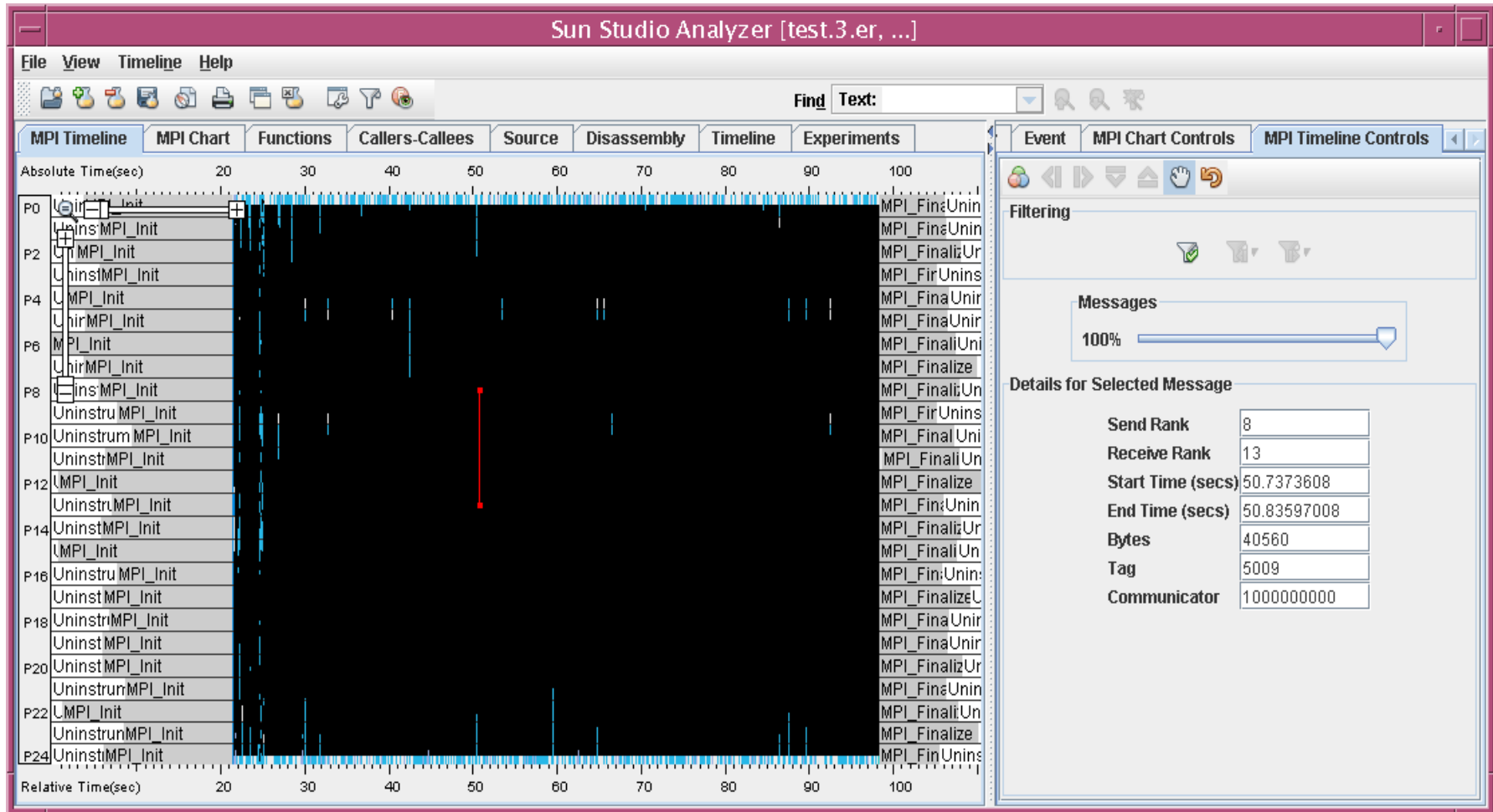
## Supplemental Information

# BT -- MPI-Timeline, I



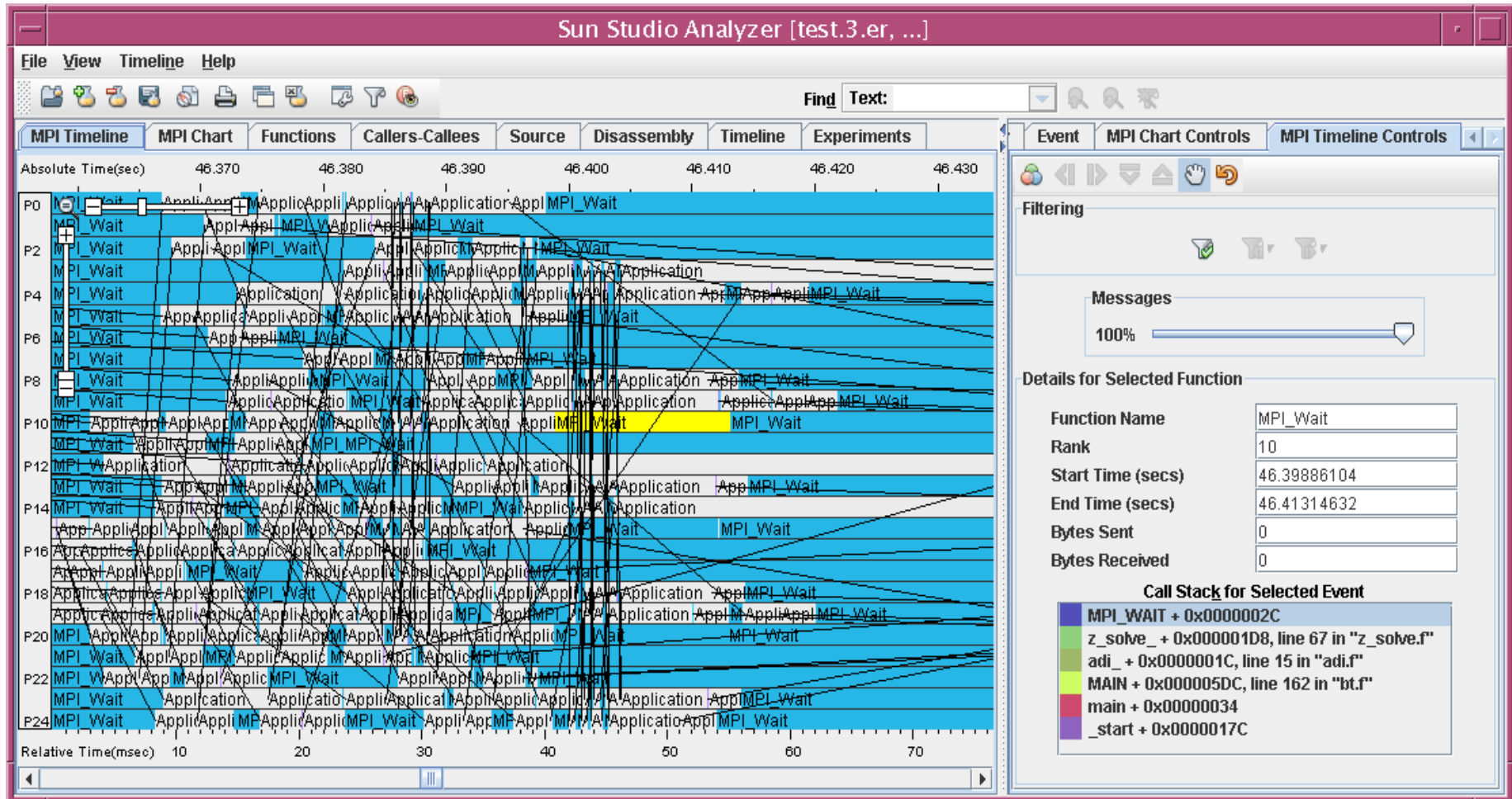
Only 2% of all messages shown

# BT -- MPI-Timeline, II



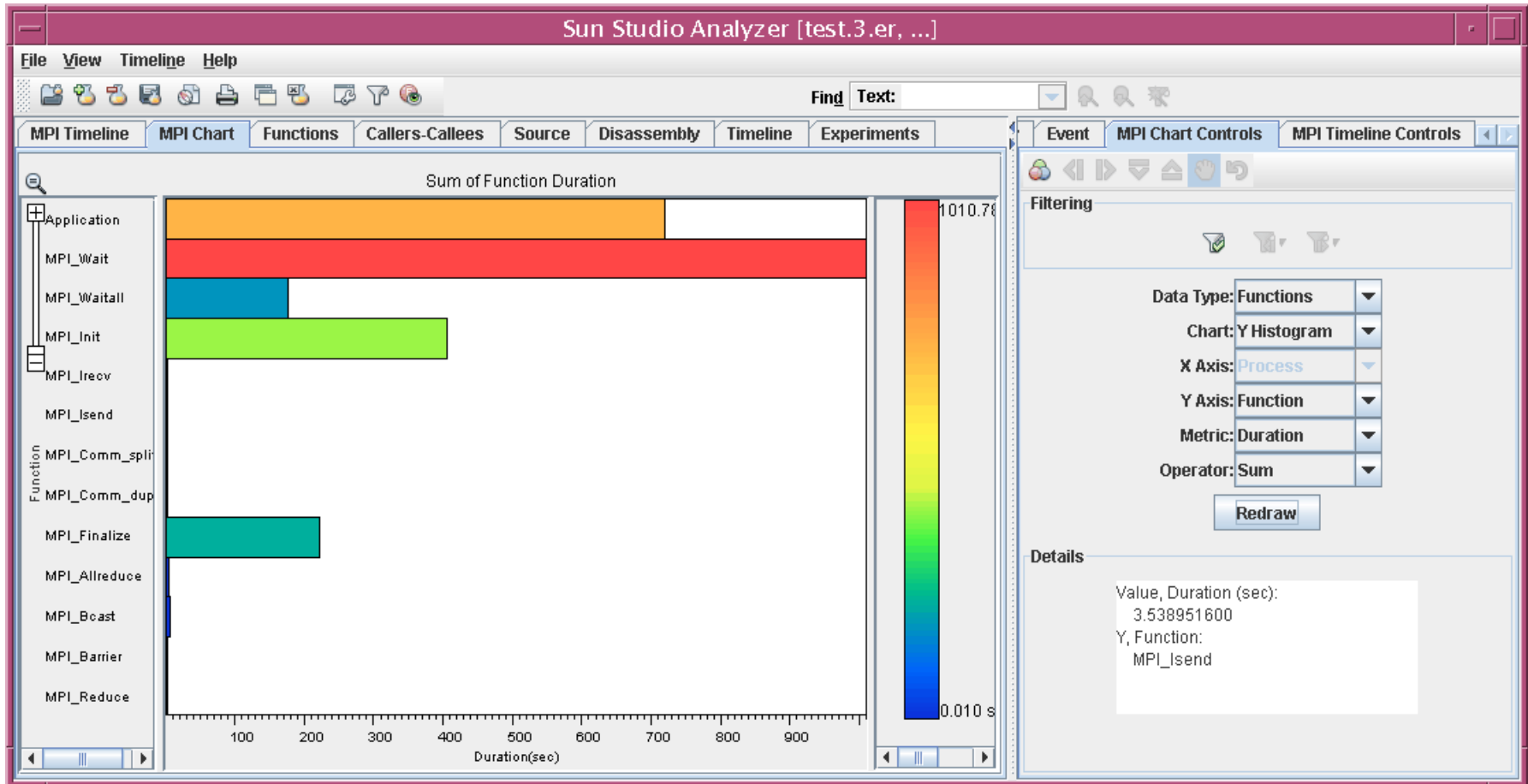
All messages shown

# BT -- MPI-Timeline, III



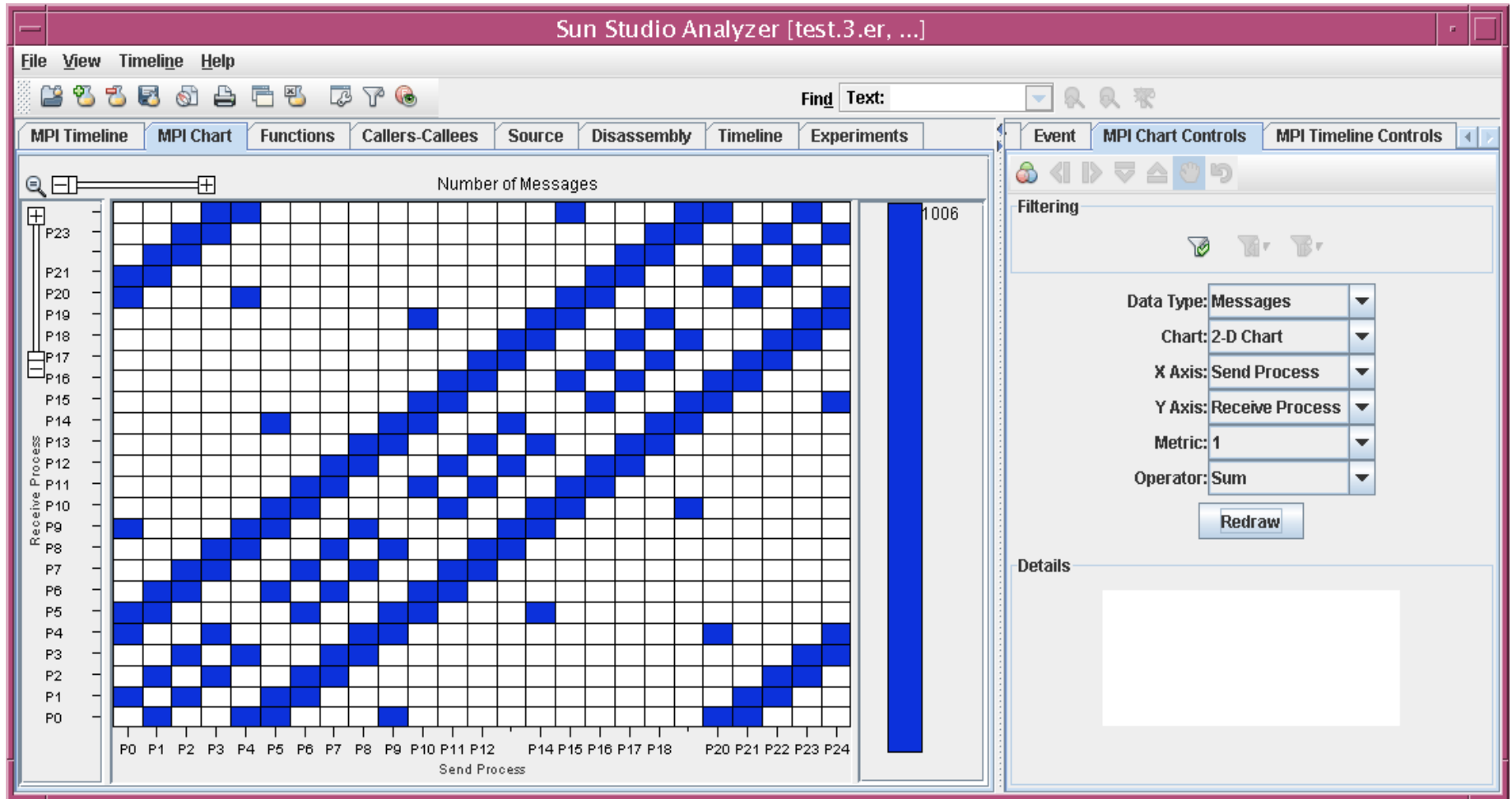
**Zoom in, to show individual messages**

# BT -- MPI Charts, I



Shows time in functions, aggregated over all ranks

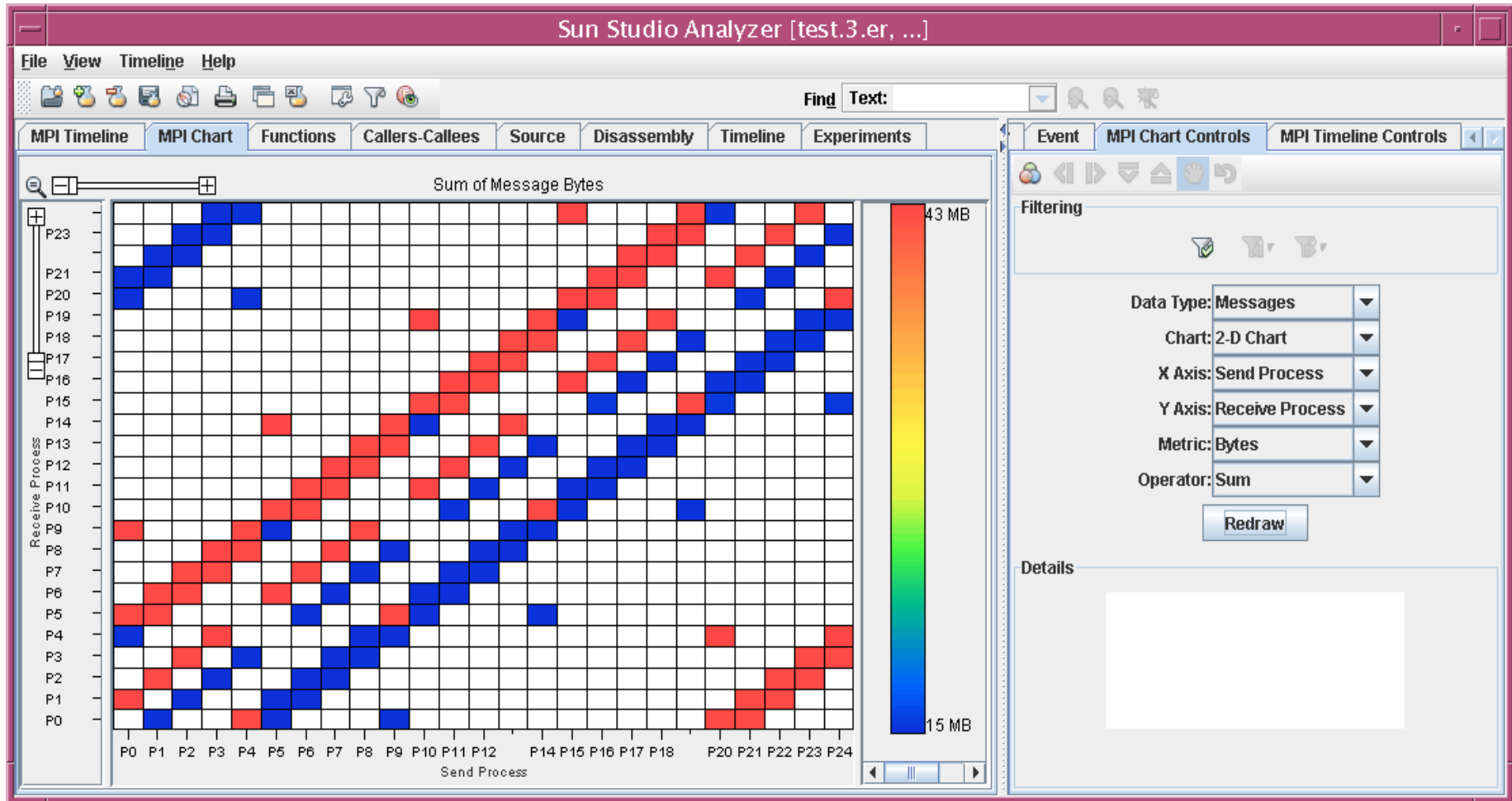
# BT -- MPI Charts, II



Shows count of messages between pairs of processes

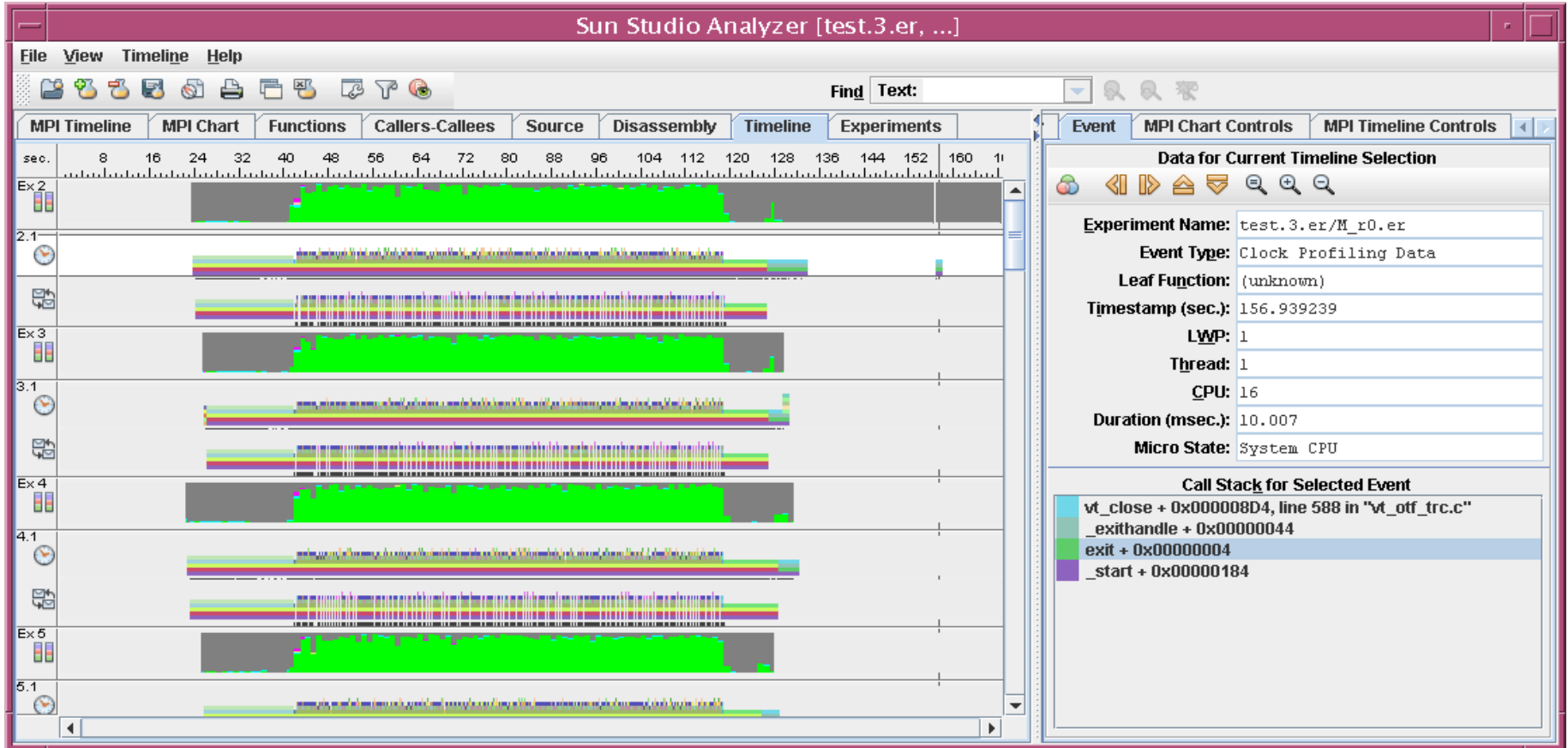


# BT -- MPI Charts, III



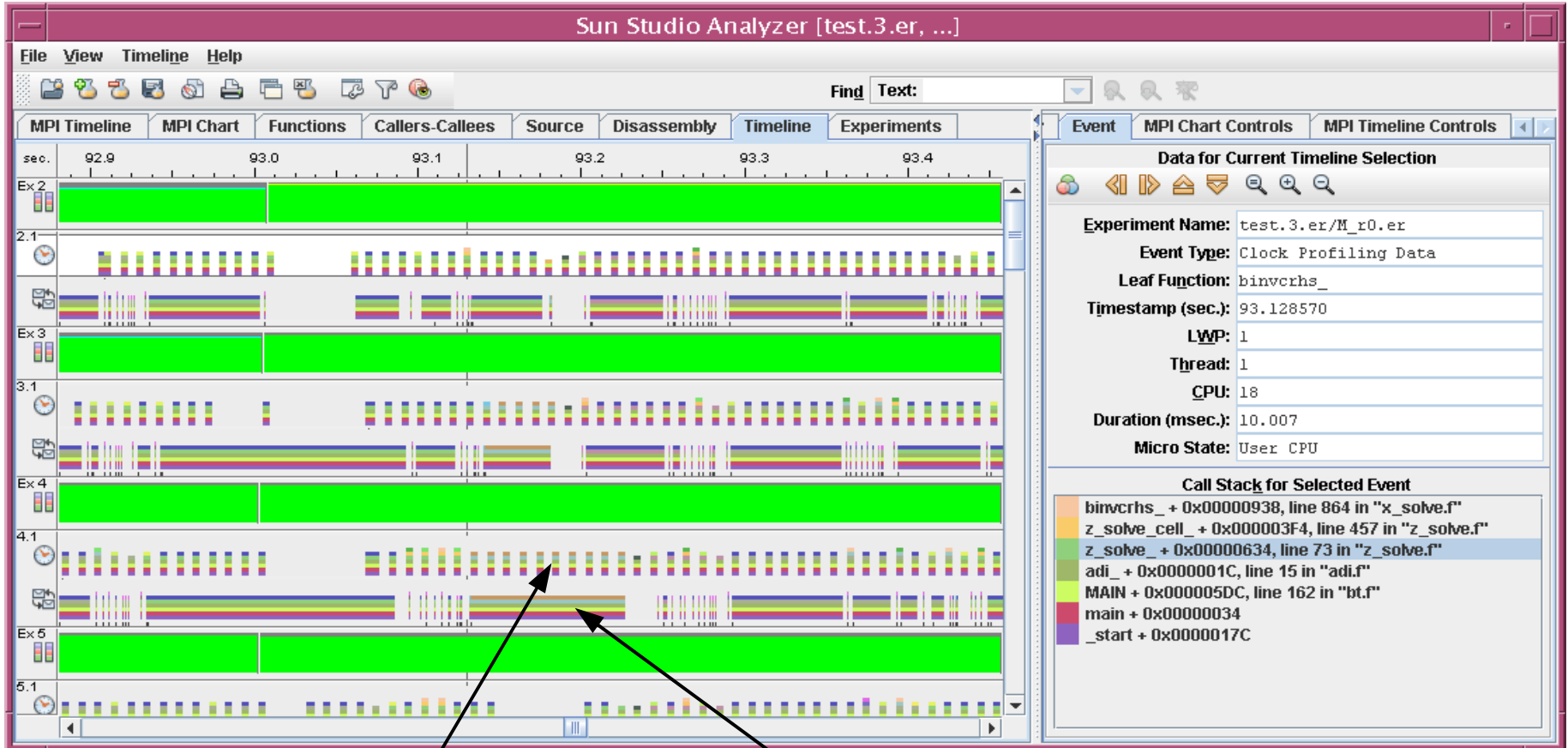
**Shows byte-count of messages between pairs of processes**

# BT – Traditional Timeline, I



Shows both profile and MPI Trace Records

# BT – Traditional Timeline, II

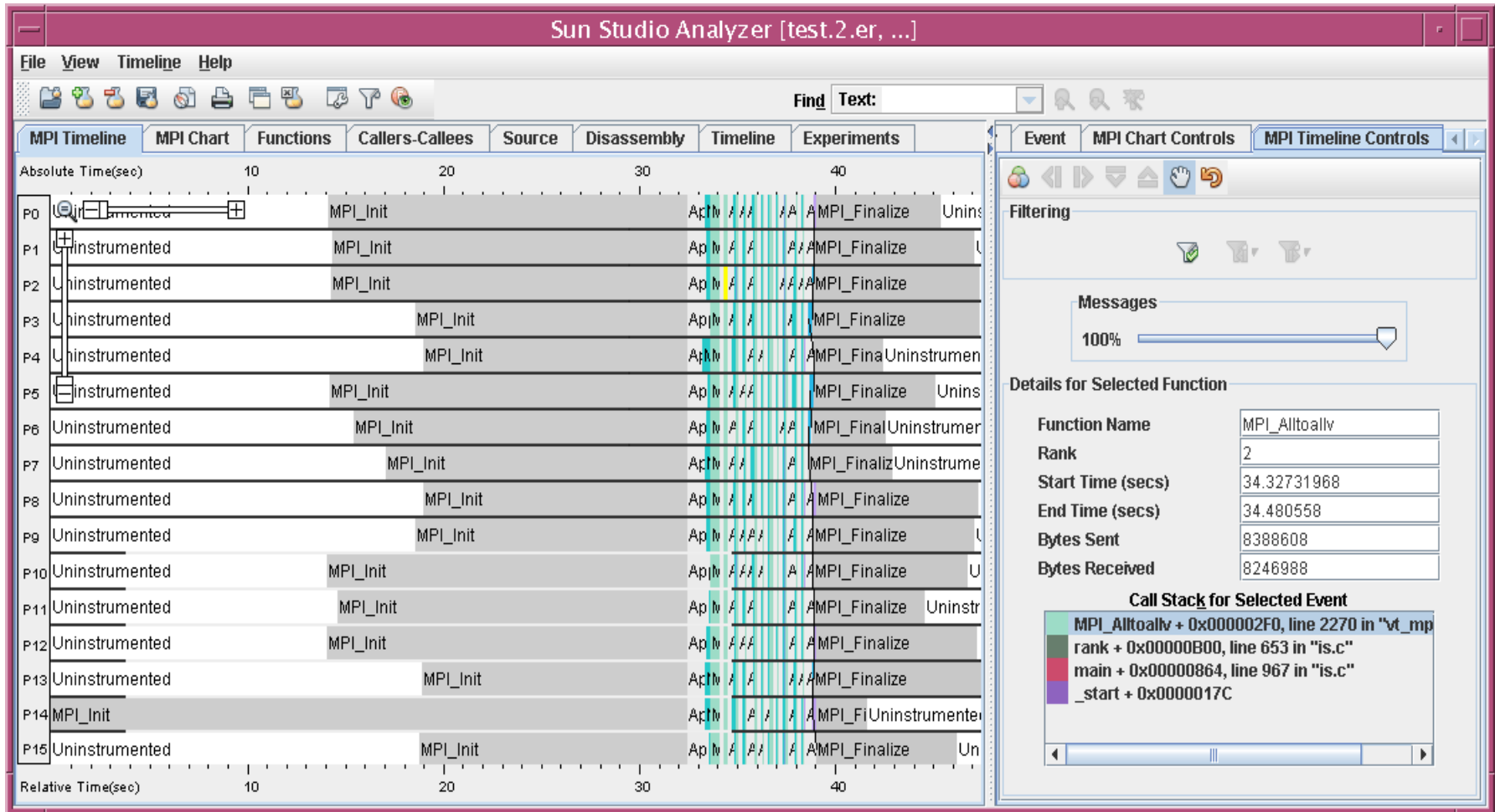


Profile records

MPI Trace records

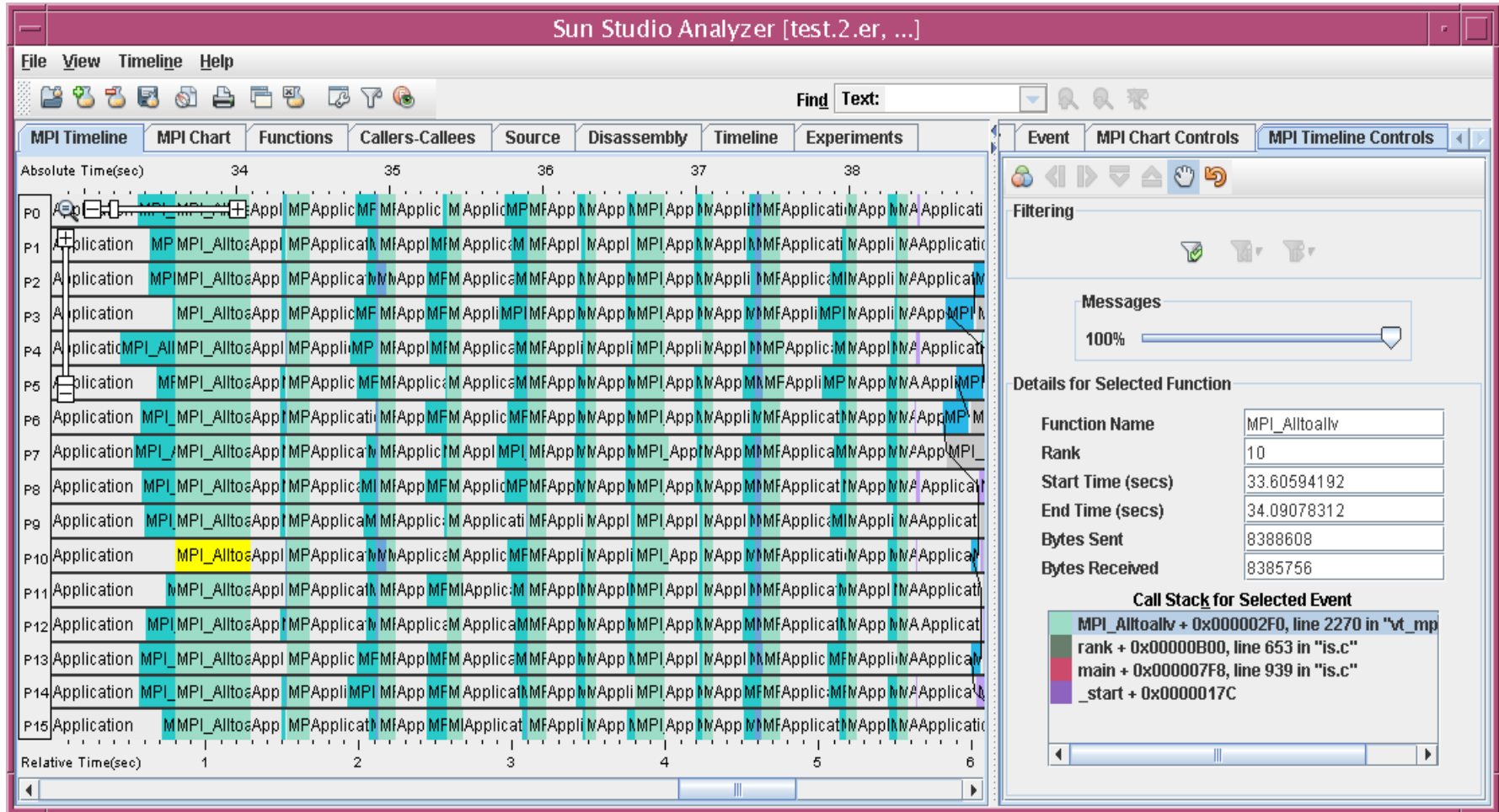
Matching callstacks when inside MPI call

# IS -- MPI-Timeline, I



Full scale, tiny run, dominated by MPI\_Init and MPI\_Finalize

# IS -- MPI-Timeline, II



**Zoomed in, to show pattern during computation**  
**Almost all MPI operations are collectives**

# MPI State Profiling

Marty Itzkowitz

Sun Studio Performance Analyzer Team

Sun Microsystems Inc.

**`marty.itzkowitz@sun.com`**