

FRASCAL HPC Day

Intel Trace Analyzer and Collector (ITAC)



Intel Trace Collector and Analyzer

Event-based tool that records

- user function calls
- MPI communication calls

Summary: xhpcg.stf

Total time: **1.88e+04** sec. Resources: **9** processes, **1** node.

Ratio

This section represents a ratio of all MPI calls to the rest of your code in the application.

Serial Code	1.86e+04 sec	99.1 %
MPI calls	160 sec	0.8 %

Top MPI functions

This section lists the most active MPI functions from all MPI calls.

MPI_Allreduce	854.298e-3 s
MPI_Wait	226e-6 s
MPI_Send	41e-6 s
MPI_Irecv	39.4382 s
MPI_Wtime	1.6031 s

Where to start with analysis

For deep analysis of the MPI-bound application click "Continue >" to open the tracefile View and leverage the **Intel® Trace Analyzer** functionality:

- *Performance Assistant* - to identify possible performance problems
- *Imbalance Diagram* - for detailed imbalance overview
- *Tagging/Filtering* - for thorough customizable analysis

To optimize node-level performance use:

Intel® VTune™ Amplifier XE for:

- algorithmic level tuning with hotspots and threading efficiency analysis;
- microarchitecture level tuning with general exploration and bandwidth analysis;

Intel® Advisor for:

- vectorization optimization and thread prototyping.

For more information, see documentation for the respective tool:

[Analyzing MPI applications with Intel® VTune™ Amplifier XE](#)

[Analyzing MPI applications with Intel® Advisor](#)

Show Summary Page when opening a tracefile

Performance Issue

Name	Duration (%)	Duration
Wait at Barrier	0.15%	100.472e-3 s
Late Sender	0.04%	29.609e-3 s
Late Receiver	0.00%	1.45e-3 s

Wait at Barrier

Description: Affected Processes: P1

Total Time [s] (Sender by Receiver)

	P0	P1	P2	P3	P4
MPI_Allreduce	18.43e-3	2.223e-3	43.023e-3	43.2e-3	43.2e-3
Sum	18.43e-3	2.223e-3	43.023e-3	33.6e-3	33.6e-3
Mean	18.43e-3	2.223e-3	43.023e-3	28.8e-3	28.8e-3
StdDev	0	0	0	24e-3	24e-3

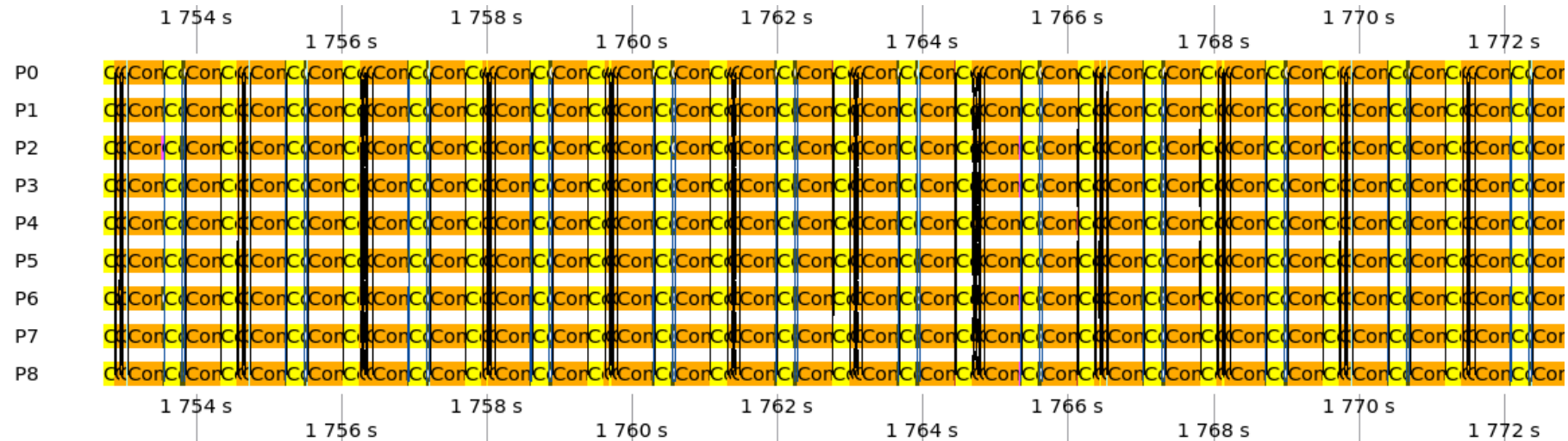
Total Time [s] (Collective Operation by Process)

	P0	P1	P2
MPI_Allreduce	18.43e-3	2.223e-3	43.023e-3
Sum	18.43e-3	2.223e-3	43.023e-3
Mean	18.43e-3	2.223e-3	43.023e-3
StdDev	0	0	0

Intel Trace Collector and Analyzer

Event timeline

- program structure
- event details
- functions
- messages
- collectives operations



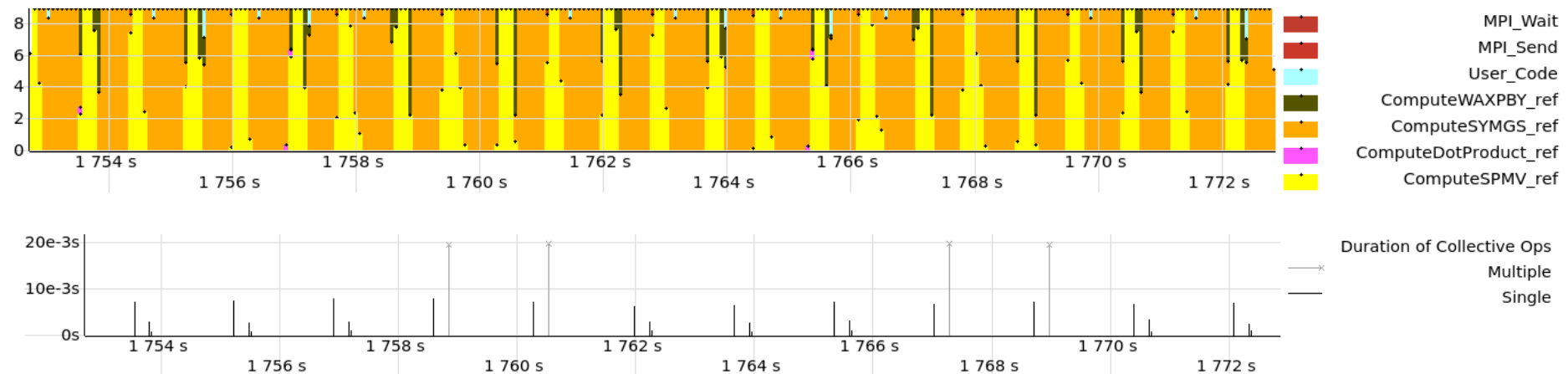
Intel Trace Collector and Analyzer

Event timeline

- program structure
- event details
 - functions
 - messages
- collectives operations

Quantitative + quantitative timeline

- transfer duration
- transfer volume
- activities measure



Intel Trace Collector and Analyzer

Event timeline

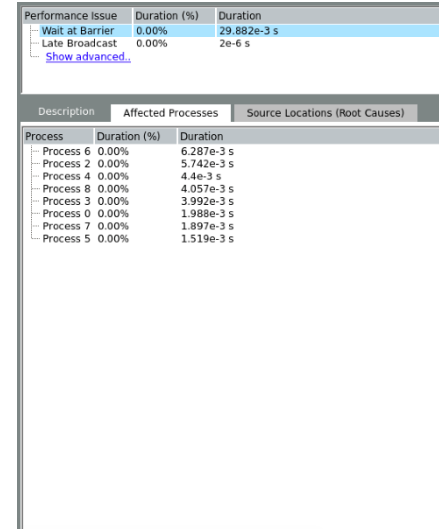
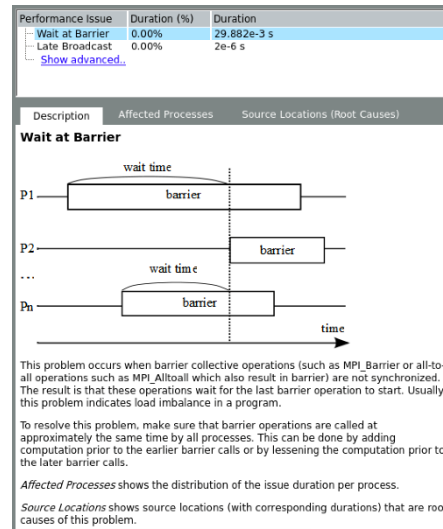
- program structure
- event details
 - functions
 - messages
 - collectives operations

Quantitative + quantitative timeline

- transfer duration
- transfer volume
- activities measure

Performance assistance

- identify performance issues with explanation
- tips on potential solutions



Intel Trace Collector and Analyzer

Event timeline

- program structure
- event details
 - functions
 - messages
- collectives operations

Quantitative + quantitative timeline

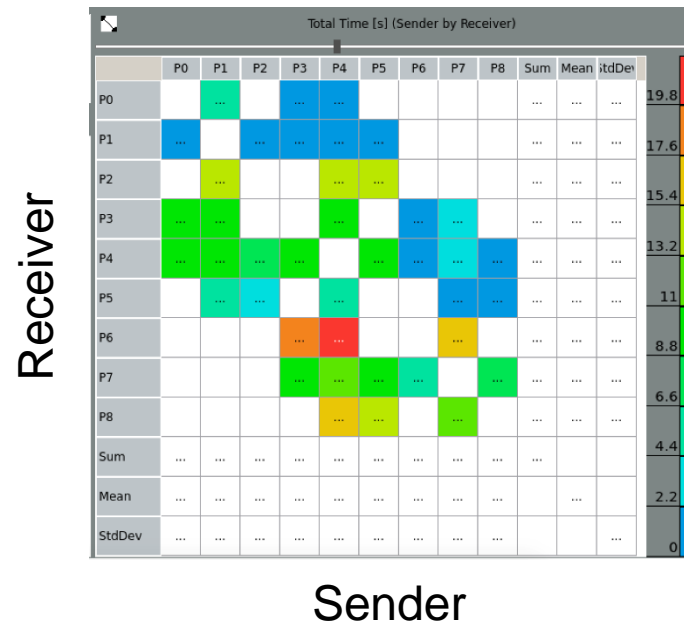
- transfer duration
- transfer volume
- activities measure

Performance assistance

- identify performance issues with explanation
- tips on potential solutions

Message profile

- message volume, time, count
- finding communication hotspots
- communication patterns



Intel Trace Collector and Analyzer

Hotspot

Event timeline

- program structure
- event details

Quantitative + quantitative timeline

- transfer duration
- transfer volume

Performance assistance

- identify performance issues with annotations on potential bottlenecks

Message profile

- message volume, time, count

Collective ops profile

- all collective operations for each process

Function profile

- functions statistics
- Call tree
- split to children of Group All_Processes
- filtered view with expansion into various MPI and Application
- Load balance
- Call graph

Name	TSelf	TSelf	TTotal	#Calls	TSelf /Call
All_Processes	5.46742e+3 s	5.46742e+3 s	5.5119e+3 s	44244	123.574e-3 s
ComputeSPMV_ref	5.46742e+3 s	5.46742e+3 s	5.5119e+3 s	44244	123.574e-3 s
ComputeWAXPBY	53.862e-3 s	53.862e-3 s	475.234 s	31995	1.68345e-6 s
ComputeDotProduct_ref	264.986 s	264.986 s	329.014 s	33408	7.93181e-3 s
ComputeSPMV	15.826e-3 s	15.826e-3 s	2.49945e+3 s	10863	1.45687e-6 s
ComputeSYMGs_ref	11.9892e+3 s	11.9892e+3 s	12.0259e+3 s	76608	156.5e-3 s
ComputeWAXPBY_ref	495.263 s	495.263 s	495.263 s	33345	14.8527e-3 s
ComputeDotProduct	31.118e-3 s	31.118e-3 s	315.682 s	32049	970.951e-9 s
User_Code	432.994 s	432.994 s	18.8104e+3 s	9	48.1104 s
MPI_Comm_size	132.739e-3 s	132.739e-3 s	132.739e-3 s	120861	1.09828e-6 s
MPI_Comm_rank	63.288e-3 s	63.288e-3 s	63.288e-3 s	120861	523.643e-9 s
MPI_Finalize	4.043e-3 s	4.043e-3 s	4.043e-3 s	9	449.222e-6 s
MPI_Bcast	60e-6 s	60e-6 s	60e-6 s	9	6.66667e-6 s
MPI_irecv	682.73e-3 s	682.73e-3 s	682.73e-3 s	537120	1.27109e-6 s
MPI_Wtime	140.191e-3 s	140.191e-3 s	140.191e-3 s	223353	627.666e-9 s
MPI_Send	7.83291 s	7.83291 s	7.83291 s	537120	14.5832e-6 s
MPI_Allreduce	75.8164 s	75.8164 s	75.8164 s	33534	2.26088e-3 s
MPI_Wait	75.7904 s	75.7904 s	75.7904 s	537120	141.105e-6 s

```

graph TD
    All_Processes --> User_Code
    All_Processes --> MPI_Bcast
    All_Processes --> MPI_Comm_rank
    All_Processes --> MPI_Comm_size
    All_Processes --> MPI_Wtime
    All_Processes --> MPI_Allreduce
    All_Processes --> ComputeSPMV_ref
    All_Processes --> MPI_Comm_size
    All_Processes --> MPI_Comm_rank
    All_Processes --> MPI_irecv
    All_Processes --> MPI_Send
    All_Processes --> MPI_Wait
    All_Processes --> ComputeSYMGs_ref
    All_Processes --> MPI_Comm_size
    All_Processes --> MPI_Comm_rank
    All_Processes --> MPI_irecv
    All_Processes --> MPI_Send
    All_Processes --> MPI_Wait
    All_Processes --> ComputeWAXPBY_ref
    All_Processes --> MPI_Wtime
    All_Processes --> MPI_Allreduce
    All_Processes --> ComputeSPMV
    All_Processes --> MPI_Comm_size
    All_Processes --> MPI_Comm_rank
    All_Processes --> MPI_irecv
    All_Processes --> MPI_Send
    All_Processes --> MPI_Wait
    All_Processes --> ComputeWAXPBY
  
```

Legend:

- ComputeSPMV_ref
- ComputeWAXPBY
- ComputeDotProduct_ref
- ComputeSPMV
- ComputeSYMGs_ref
- ComputeWAXPBY_ref
- ComputeDotProduct
- User_Code
- MPI_Comm_size
- MPI_Comm_rank
- MPI_Finalize
- MPI_Bcast
- MPI_irecv
- MPI_Wtime
- MPI_Send

```

graph TD
    subgraph Callers
    C1[ComputeSPMV_ref called by ComputeSPMV 2.49475e+3 s]
    C2[ComputeSPMV_ref called by User_Code 2.97267e+3 s]
    end
    subgraph Callees
    C3[ComputeSPMV_ref calling MPI_Comm_size]
    C4[ComputeSPMV_ref calling MPI_Comm_rank]
    C5[ComputeSPMV_ref calling MPI_irecv]
    C6[ComputeSPMV_ref calling MPI_Send]
    C7[ComputeSPMV_ref calling MPI_Wait]
    end
    C1 --- C3
    C1 --- C4
    C1 --- C5
    C1 --- C6
    C1 --- C7
    C2 --- C3
    C2 --- C4
    C2 --- C5
    C2 --- C6
    C2 --- C7
  
```


ITAC: Use with caution!



Compiler switches, API

Compiler switches

1. `-trace`
2. `-tcollect -trace`
function profile at a high price of maximum overhead
3. `-tcollect-filter=func.txt -tcollect -trace`
restrict tracing to certain functions

```
$ cat func.txt
      '.*' OFF
    '.*ComputeDotProduct.*' ON
    '.*ComputeSYMGS.*' ON
    '.*ComputeSPMV.*' ON
    '.*ComputeWAXPBY.*' ON
```

Variants

1. `-trace-imbalance`
trace only the MPI functions that cause application load imbalance (idle at some point of the application run)
2. `-trace-collectives`
trace only collectives
3. `-trace-pt2pt`
trace only point-to-point operations

Watch out

check pinning correctness

```
$ impi_info
$ export I_MPI_DEBUG=4
```

use filtering options for large problems
If not handled carefully,
generates a lot of unnecessary data

Starting the Analyzer app

```
$ traceanalyzer ${binary}.stf
```

VT API

Manually instrument the code to profile only interesting parts of an application or a subset of iterations

- run without "-trace"
- `#include <VT.h>`
- `-I${VT_ROOT}/include`
- inserting calls to `VT_traceoff()` and `VT_traceon()`
- `VT_begin(mark)`, `VT_end(mark)`

Environment variables

Environment variables	Default	Description
VT_FLUSH_PREFIX	/tmp	Location for temporary flush files
VT_LOGFILE_PREFIX	\$PWD	Location for physical trace information files
VT_LOGFILE_FORMAT	STF	SINGLESTF : rolls all trace files into one file (.single.stf)
VT_LOGFILE_NAME	\${binary}	name for the trace file
VT_MEM_BLOCKSIZE	64 KB	trace data in chunks of main memory
VT_MEM_FLUSHBLOCKS	1024	flushing is started when the number of blocks in memory exceeds this threshold
VT_MEM_MAXBLOCKS	1024	maximum number of blocks in main memory, if exceed the application is stopped until AUTOFLUSH/ MEM-OVERWRITE/ stop recording trace info
VT_CONFIG_RANK	0	control the process that reads and parses the configuration file

Environment variables: set up in the

1. corresponding environment variables
2. command line when running your application
3. configuration file

```
$ export VT_CONFIG=<config_file>
```

```
# enable all Application activities  
ACTIVITY Application ON
```

```
# disable all MPI activity  
ACTIVITY MPI OFF
```

```
# enable all bcasts, barrier, allreduce, recvs and sends  
SYMBOL MPI_WAITALL ON  
SYMBOL MPI_IRecv ON  
SYMBOL MPI_ISEND ON  
SYMBOL MPI_BARRIER ON  
SYMBOL MPI_ALLREDUCE ON
```

<https://www.intel.com/content/www/us/en/develop/documentation/itac-user-and-reference-guide/top/intel-trace-collector-reference/configuration-reference/configuration-options.html>