NHR FAU

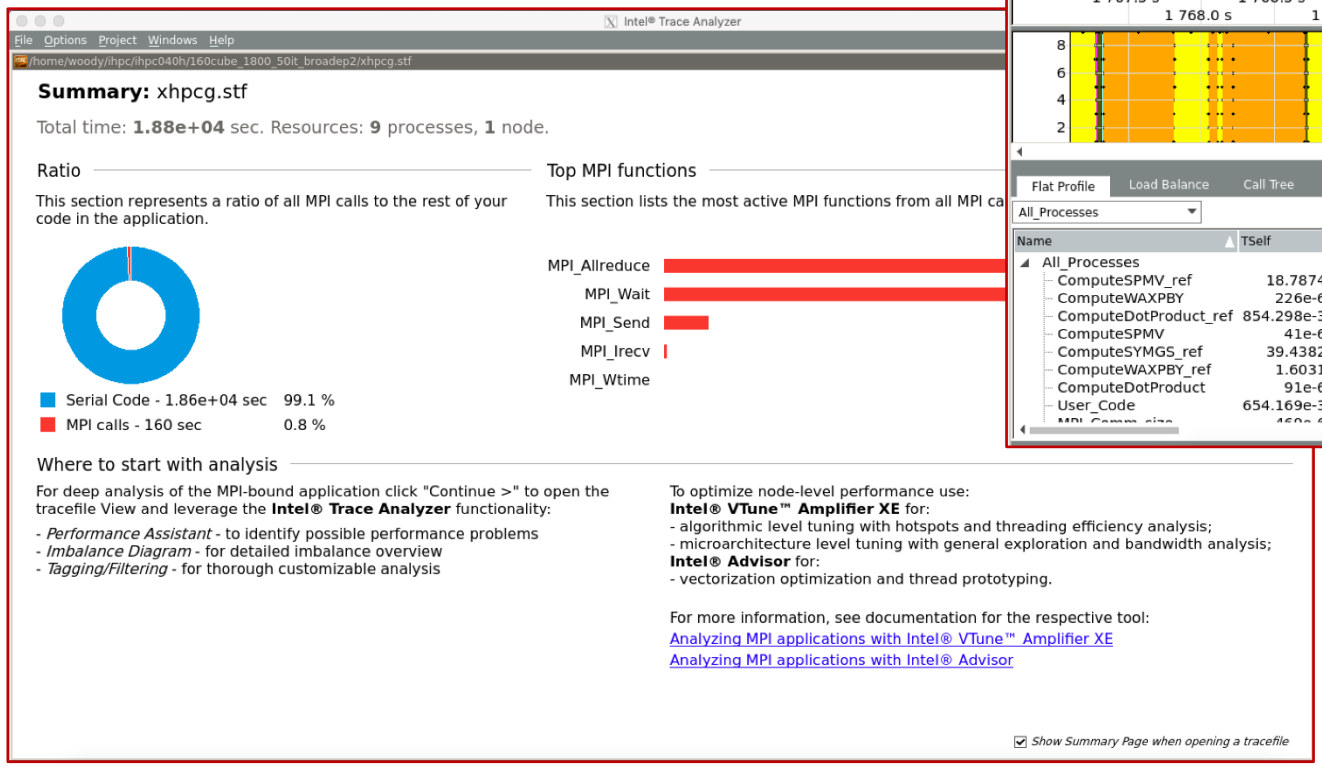
FAU Friedrich-Alexander-Universität Erlangen-Nürnberg

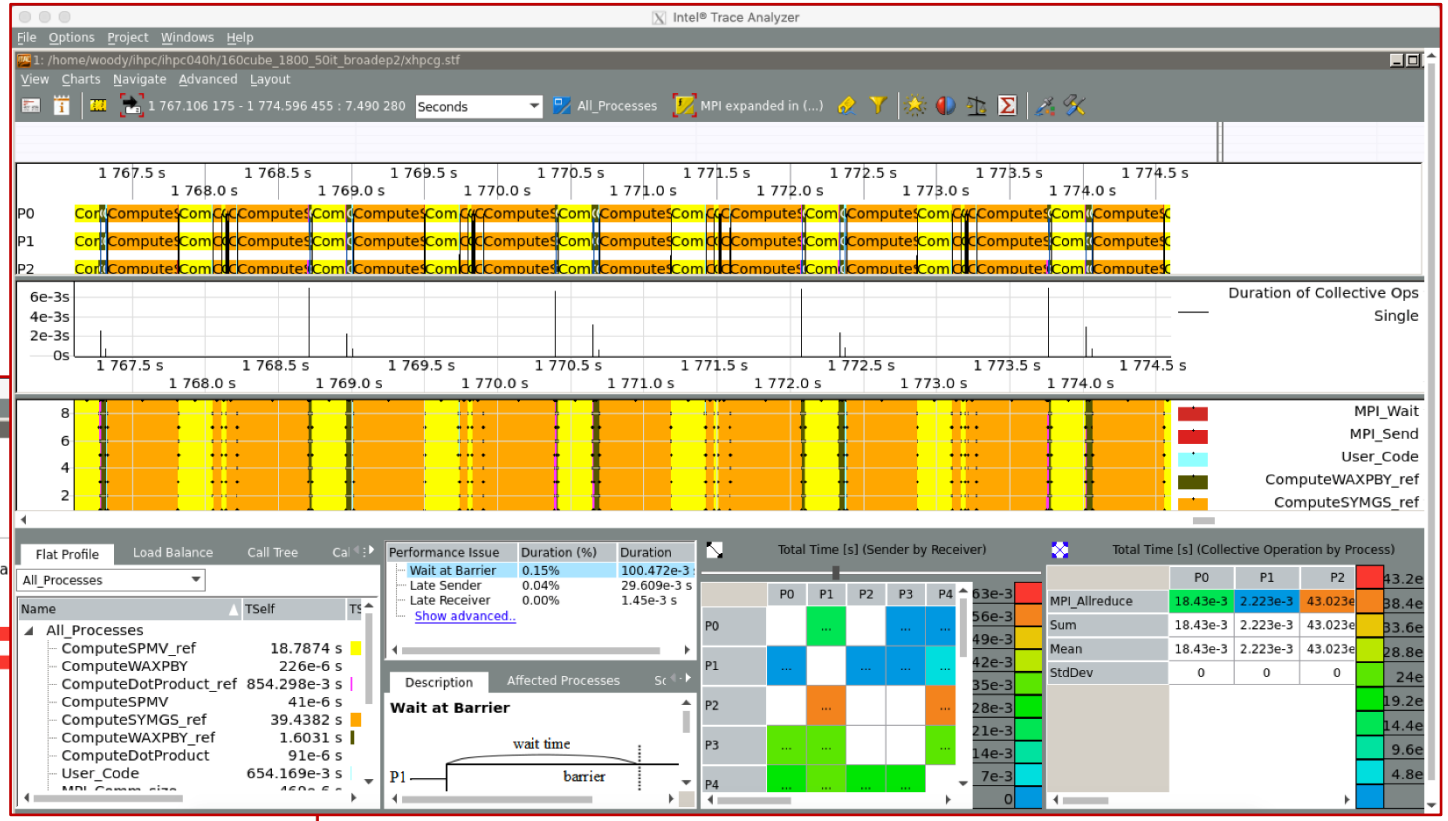
FRASCAL HPC Day

Intel Trace Analyzer and Collector (ITAC)

Intel Trace Collector and Analyzer

Event-based tool that records
- user function calls
- MPI communication calls

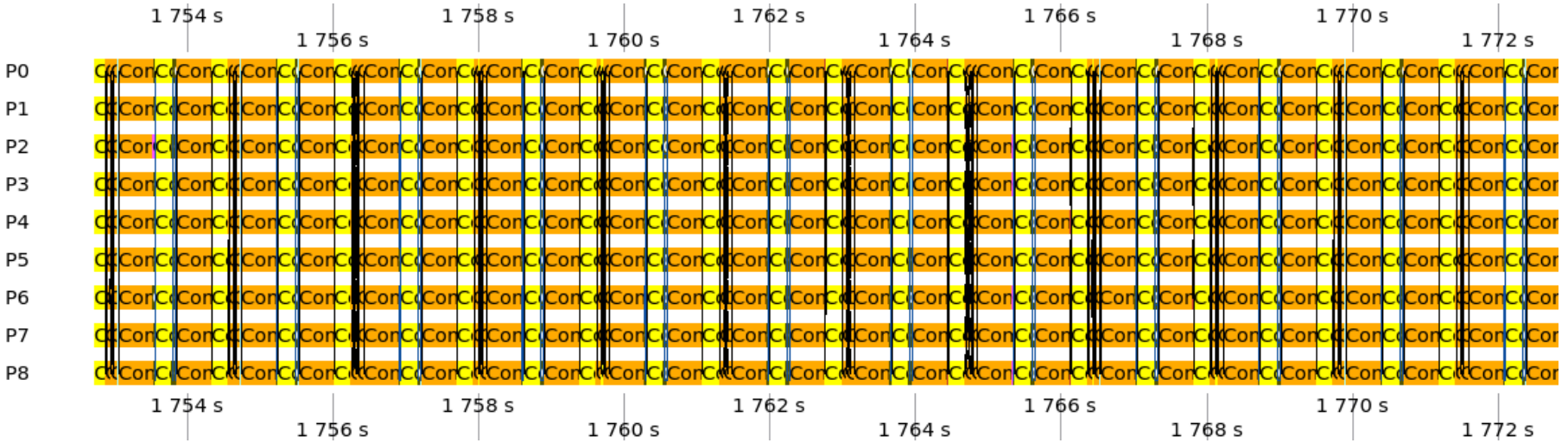




Intel Trace Collector and Analyzer

Event timeline

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



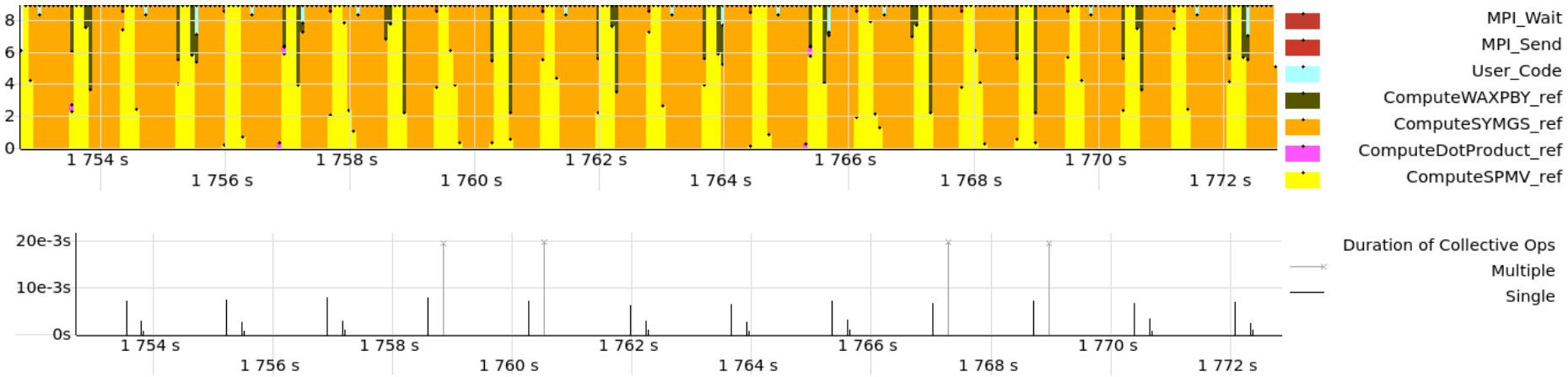
Intel Trace Collector and Analyzer

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

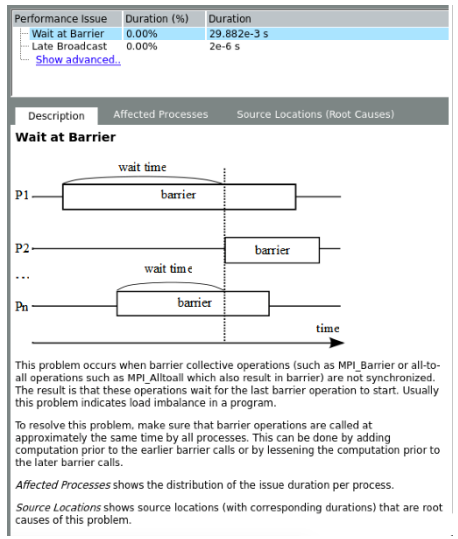
- program structure
- event details
  - functions
  - messages
- collectives operations

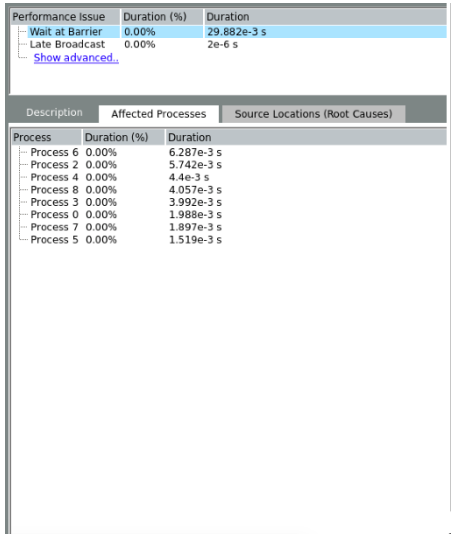
Quantitative + quantitative timeline

- 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

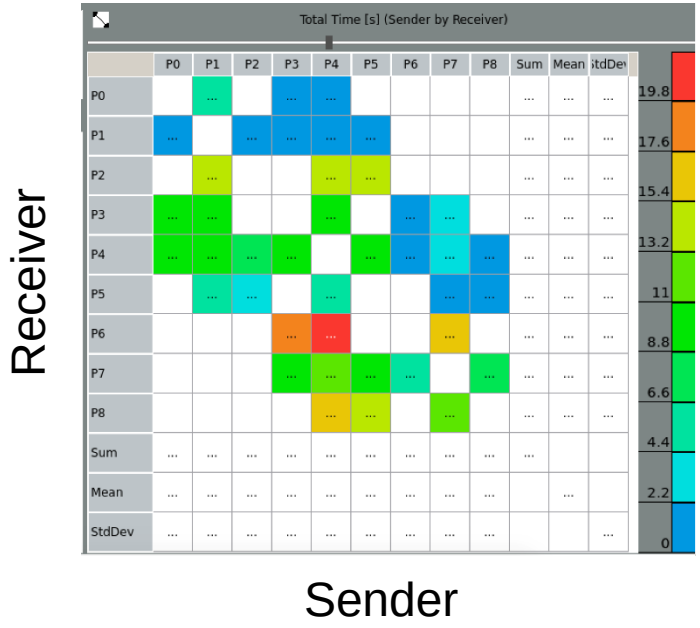
- transfer duration
- transfer volume
- activities measure

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

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

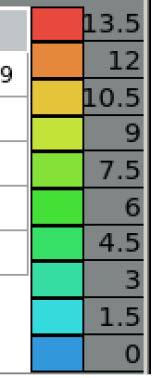
- message volume, time, count
- finding communication hotspots
- communication patterns

Collective ops profile

- all collective operations for each process

Total Time [s] (Collective Operation by Process)

	P0	P1	P2	P3	P4	P5	P6	P7	P8	Sum	Mean	StdDev
MPI_Bcast	5e-6	7e-6	7e-6	7e-6	7e-6	7e-6	6e-6	7e-6	7e-6	60e-6	6.66667e-6	666.667e-9
MPI_Allreduce	6.98827	2.41008	14.1332	9.46671	9.80818	2.28141	12.1689	7.89127	10.6684	75.8164	8.42405	3.81376
Sum	6.98828	2.41009	14.1332	9.46671	9.80818	2.28142	12.1689	7.89127	10.6684	75.8165		
Mean	3.49414	1.20504	7.06659	4.73336	4.90409	1.14071	6.08444	3.94564	5.33422		4.21203	
StdDev	3.49413	1.20504	7.06658	4.73335	4.90409	1.1407	6.08444	3.94563	5.33422			5.00135



Intel Trace Collector and Analyzer

Hotspot

Event timeline
- program structure
- event details

Quantitative + quantitative timeline
- transfer duration
- transfer

Performance assistance
- identify performance

Message profile
- message volume, time, count

Collective ops profile
- all collective operations for each process

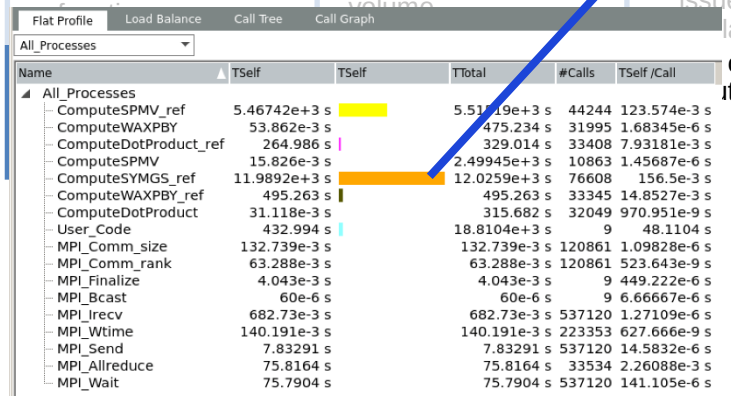
Function profile

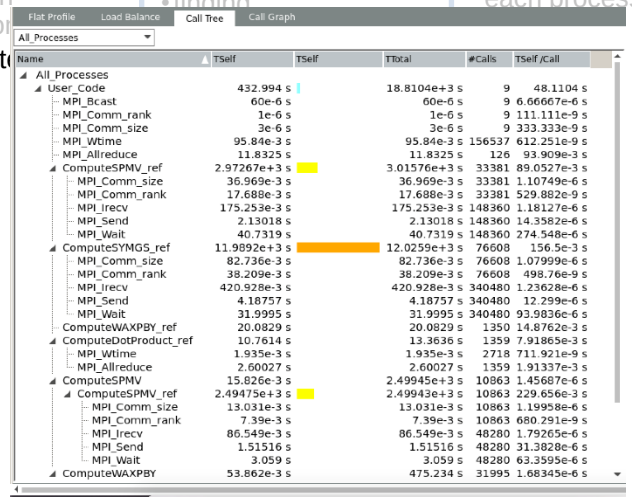
functions statistics

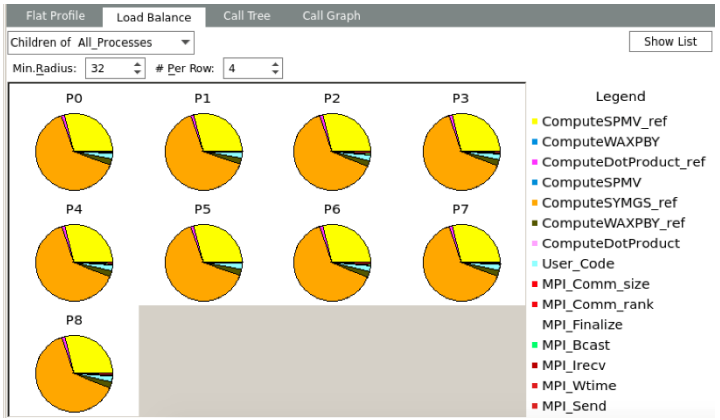
Call tree
- split to children of Group All_Processes
- filtered view with expansion into various MPI and Application

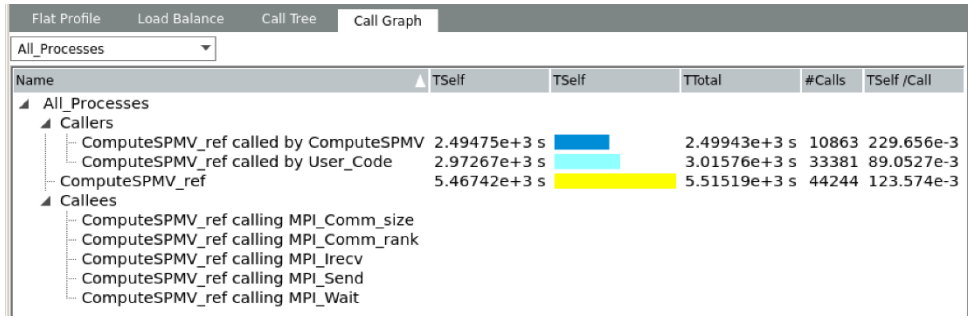
Load balance

Call graph









NHR FAU

FAU Friedrich-Alexander-Universität Erlangen-Nürnberg

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