# MPI Performance Snapshot

## User's Guide

# Legal Information



| Optimization Notice |
| --- |
| Intel's compilers may or may not optimize to the same degree for non-Intel microprocessors for optimizations that are not unique to Intel microprocessors. These optimizations include SSE2, SSE3, and SSSE3 instruction sets and other optimizations. Intel does not guarantee the availability, functionality, or effectiveness of any optimization on microprocessors not manufactured by Intel. Microprocessor-dependent optimizations in this product are intended for use with Intel microprocessors. Certain optimizations not specific to Intel microarchitecture are reserved for Intel microprocessors. Please refer to the applicable product User and Reference Guides for more information regarding the specific instruction sets covered by this notice.<br>Notice revision #20110804 |

# Contents

# 1. Introduction

This document guides you through the main steps you need to perform when you first start using MPI Performance Snapshot. It also describes the main features of the tool and explains how to use them.

## 1.1. Introducing MPI Performance Snapshot

MPI Performance Snapshot is a scalable lightweight performance tool for MPI applications. It collects the MPI application statistics, such as communication, activity, load balance, and presents it in an easy-to-read format. The collected information can be used for in-depth analysis of the application scalability and performance.

Statistics collection is available only on Linux* OS, while the data analysis is available both on Linux* and Windows* operating systems.

## 1.2. What's New

This User's Guide documents the MPI Performance Snapshot 9.1 Update 2 release.

**MPI Performance Snapshot 9.1 Update 2:**

- Support for collection of disk usage statistics. The statistics is collected by default and displayed on the Summary and Counters diagrams.
- Output file postfixes for differentiating outputs from different runs. See Setting Unique Names for Output Files for details.
- Summary HTML report enhancements. See HTML Summary Report for details.

**MPI Performance Snapshot 9.1 Update 1:**

- Minor improvements and bug fixes.

**MPI Performance Snapshot 9.1:**

- Support for collection of CPI ad Memory Bound performance metrics (preview feature). See Collecting Performance Metrics with Intel® VTune™ Amplifier XE.
- Application summary HTML report.
- Added new command-line keys: `-g`, `-0`, `-F`, `-I`. See Command Line Reference.
- The `mps` tool for statistics analysis is now available for Windows* OS. See Analyzing the Collected Statistics.
- Bug fixes.

**MPI Performance Snapshot 9.0.3 Preview:**

- Support for Intel® Many Integrated Core Architecture (Intel® MIC Architecture).
- Improved accuracy of Intel® MPI Library native statistics.
- Statistics are now collected using the `-mps` option for `mpirun`. See Collecting Statistics.
- Reduced memory consumption.

**MPI Performance Snapshot 9.0.2 Preview:**

- Initial release.

# 1.3. System Requirements

## 1.3.1. Supported Operating Systems

- Red Hat* Enterprise Linux* 5, 6, 7

## 1.3.2. Required Software

- Intel® C++ Compiler 15.0.1 or newer
- Intel® MPI Library 5.0.2 or newer
- Intel® VTune™ Amplifier XE 2016 or newer

## 1.3.3. Optional Software

- PAPI library 5.3.0 or newer

***NOTE:*** The necessary libraries `libpapi.so` and `libpfm.so` are provided with the tool. However, you can install the PAPI library manually.

# 1.4. Known Issues and Limitations

- Intel® MPI Library native statistics and statistics collected by MPI Performance Snapshot may contain different values for the same metrics. This is caused by usage of different timers and sets of functions. See the list below.

**List of supported MPI functions**

| P2P | Collectives |
|---|---|
| Send | Allgather |
| Bsend | Allgatherv |
| Rsend | Allreduce |
| Ssend | Alltoall |
| Recv | Alltoallv |
| SendRecv | Alltoallw |
| Wait | Barrier |
| Init | Bcast |
| Finalize | Exscan |
| Probe | Gather |
| Test | Gatherv |
| Cart | Reduce_scatter |
| Buffer | Reduce |

| | |
|---|---|
| Start | Scan |
| Request | Scatter |
| | Scatterv |
| | Comm |

The internal functions `CSend` and `CSendRecv` are also supported but they are not included in the output by default. To view statistics for these functions in the charts, use the `-I` option.

- By default, MPI Performance Snapshot uses the PAPI libraries included in the package (`libpapi.so` and `libpfm.so`). If you have issues when using these files, consider installing PAPI version 5.3 or higher manually.
- The MPI Performance Snapshot collector (`limpiperf*.so`) does not work with the Intel® Trace Collector. Avoid running the collectors simultaneously.
- To reduce collected statistics to the zero rank, MPI Performance Snapshot uses the `MPI_Gather` function. This function may appear in the application analysis results, even if it is not used in your application.
- MPI Performance Snapshot supports Intel® C++ Compiler version 15.0.1 or higher. If you use an earlier version of the Intel C++ Compiler, or a different compiler, information about the OpenMP* regions and metrics will be unavailable.
- MPI Performance Snapshot supports Intel® MPI Library version 5.0.2 or higher. If you use an earlier version of the library, some functionality will not be available:
  - No time values for the MPI functions
  - No information about the MPI Imbalance metric
- On the Intel® Many Integrated Core Architecture (Intel® MIC Architecture) OpenMP* imbalance statistics may be inaccurate if you use Intel® C++ Compiler version lower than 16.0.
- Statistics collection is unavailable on Windows* OS, however you can process the collected data using the `mps.exe` tool.
- The results provided by MPI Performance Snapshot use internal counters which are reasonably accurate but due to limitations in their implementation are not 100% precise. As such, the GFLOP values provided by this tool are the best available approximations but not necessarily 100% accurate.
- Limitations related to performance metrics collection with Intel® VTune™ Amplifier XE:
  - The `perf` utility should be available on all nodes.
  - Metrics can be empty for short applications.
  - Intel® VTune™ Amplifier XE may produce additional output.
  - Intel® VTune™ Amplifier XE limitations (see the product documentation for details).
- Limitations related to collection of disk usage statistics:
  - To collect the statistics, MPI Performance Snapshot uses data provided by the OS. Wait time or idle time caused by waiting data from the I/O subsystem is measured with relatively low precision. This is why for short tasks I/O wait time may equal zero, even in case of active use of the I/O subsystem.
  - During a process run, if the system reassigns it to a different CPU core, the data read/written and I/O wait time will equal zero.

# 1.5. Conventions and Symbols

The following conventions are used in this document.

### Conventions and Symbols used in this Document

| | |
|---|---|
| `This type style` | Indicates an element of syntax, reserved word, keyword, filename, computer output, or part of a program example. The text appears in lowercase unless uppercase is significant. |
| This type style | Introduces new terms, denotation of terms, placeholders, or titles of manuals. |
| <this type style> | Placeholders for actual values. |
| [this type style] | Optional arguments. |

# 2. Before You Begin

## 2.1. Setting Up the Environment

Before you start using MPI Performance Snapshot on Linux* OS, make sure to install the necessary software and libraries and set up the environment for compilers, Intel® MPI Library and MPI Performance Snapshot using the appropriate scripts. No additional actions required for Windows* OS.

For example:

```
$ source <compiler_install_dir>/bin/compilervars.sh
$ source <IMPI_install_dir>/intel64/bin/mpivars.sh
$ source <ITAC_install_dir>/bin/mpsvars.sh
```

For performance metrics collection MPI Performance Snapshot uses Intel® VTune™ Amplifier XE (see here for details). Make sure to set the appropriate variables to collect these metrics:

```
$ source <VTune_install_dir>/amplxe-vars.sh
```

***NOTE:*** For limitations of using old versions of Intel® compilers and Intel® MPI Library, see the Known Issues and Limitations section.

## 2.2. Choosing Statistics to Collect

By default, the tool collects all kinds of statistics, except for hardware counter statistics (`--papi`) and performance metrics (`-w` or `--vtune`). Use `mpsvars.[c]sh` options to specify which statistics are to be collected, and to perform other actions:

| Option | Description |
|---|---|
| `-h [--help]` | Show help and exit. |
| `-v [--verbose]` | Enable verbose mode. |
| `-r [--reset]` | Reset the environment variables. |
| `--noimpistats` | Skip collection of the Intel® MPI Library native statistics. |
| `--nocollector` | Skip collection of the MPI Performance Snapshot statistics. |
| `--noopenmp` | Skip collection of the OpenMP* statistics. |
| `--papi` | Collect the hardware (PAPI) counter statistics. Disabled by default. |
| `--papidir=<dir>` | Set path to your PAPI installation folder. Use this option if you have installed PAPI manually. |
| `-w [--vtune]` | Collect performance metrics by means of Intel® VTune™ Amplifier XE (preview feature). This option is mutually exclusive with `--papi`. |

# 3. Collecting Statistics

MPI Performance Snapshot can collect the following data:

- Intel® MPI Library native statistics
- Memory usage statistics
- Disk usage statistics
- MPI library load balance statistics
- OpenMP* load balance statistics
- Hardware (PAPI) counter statistics
- Performance metrics by means of Intel® VTune™ Amplifier XE

***NOTE:*** Statistics collection is unavailable on Windows* OS.

## 3.1. Collecting in Standard Mode

In the standard mode the tool collects Intel® MPI Library native statistics, memory usage statistics, disk usage statistics, MPI library and OpenMP* load balance statistics.

To collect the statistics, do the following:

1. Set up the environment variables for compilers, Intel® MPI Library and MPI Performance Snapshot as described in Setting Up the Environment.
2. Run your application with the `-mps` option. For example:

```
$ mpirun -mps –n 4 ./myApp
```

MPI Performance Snapshot provides the statistics information for your application in the standard output and in the following files:

- `stats_<date>-<time>.txt` — contains the Intel® MPI Library native statistics
- `app_stat_<date>-<time>.txt` — contains the statistics provided by MPI Performance Snapshot

where <date> and <time> are placeholders for creation date and time, respectively.

***NOTE:*** The date and time postfixes are not appended to names of the output files, if you use Intel® MPI Library version earlier than 5.1.2. See Setting Unique Names for Output Files for details.

The statistics information in the standard output may look as follows:

```
=================== GENERAL STATISTICS ====================
Total time:           21.077 sec   (All ranks)
       MPI:            5.12%
   NON_MPI:           94.88%

WallClock :
      MIN :           10.538 sec   (rank 1)
      MAX :           10.539 sec   (rank 0)
```

```
================== DISK USAGE STATISTICS ==================
                 Read                  Written             I/O Wait time (sec)
All ranks:       61.6 MB               381.5 MB             17.020172
      MIN:       15.4 MB (rank 0)       95.4 MB (rank 3)     4.099998 (rank 3)
      MAX:       15.4 MB (rank 3)       95.4 MB (rank 0)     4.370003 (rank 1)
================== MEMORY USAGE STATISTICS ================
All  processes:      1.430MB
           MIN:      0.633MB  (rank 0)
           MAX:      0.797MB  (rank 1)

================ MPI IMBALANCE STATISTICS =================
MPI Imbalance:           10.123 sec           48.030% (All processes)
          MIN:            0.010 sec            0.095% (rank 1)
          MAX:           10.113 sec           95.959% (rank 0)

==================== OpenMP STATISTICS ====================
OpenMP Regions:           10.121 sec           48.021%               1 region(s) (All processes)
           MIN:            0.000 sec            0.000%               0 region(s) (rank 0)
           MAX:           10.121 sec           96.047%               1 region(s) (rank 1)

OpenMP Imbalance:            4.963 sec           23.548% (All processes)
             MIN:            0.000 sec            0.000% (rank 0)
             MAX:            4.963 sec           47.099% (rank 1)
```

In addition to this statistics, you can collect hardware counter statistics or performance metrics. See the instructions below.

# 3.2. Collecting Hardware Counter Statistics

***NOTE:*** Collecting this statistics is mutually exclusive with performance metrics collection described below.

To collect the hardware counter statistics, do the following:

1. Set up the environment variables for compilers and Intel® MPI Library as described in Setting Up the Environment.
2. Set up the environment variables for MPI Performance Snapshot by sourcing the `mpsvars.sh` script with the `--papi` option:

   ```
   $ source <ITAC_install_dir>/bin/mpsvars.sh --papi
   ```

   By default, MPI Performance Snapshot uses the PAPI libraries included in the package. You may point to the manually installed version of PAPI using the `--papidir` option.
3. Run your application with the `-mps` option. For example:

   ```
   $ mpirun -mps –n 4 ./myApp
   ```

4. The statistics will be written into the `app_stat.txt` file and printed in the standard output. For example:

```
================== HW COUNTERS STATISTICS ==================
GFlops:    0.038   MPI:  48.11%   NON_MPI:  51.89%

Floating-Point instructions:  13.63%
Vectorized  DP instructions:   0.00%
Memory  access instructions:  44.33%
```

# 3.3. Collecting Performance Metrics with Intel® VTune™ Amplifier XE (Preview)

***NOTE:*** Collecting performance metrics is mutually exclusive with hardware counter collection described above.

To collect the performance metrics, do the following:

1. Set up the environment variables for compilers, Intel® MPI Library and Intel® VTune™ Amplifier XE as described in Setting Up the Environment.
2. Set up the environment variables for MPI Performance Snapshot by sourcing the `mpsvars.sh` script with the `-w` or `--vtune` option:

```
$ source <ITAC_install_dir>/bin/mpsvars.sh --vtune
```

3. Run your application with the `-mps` option of `mpirun`. For example:

```
$ mpirun -mps -n 4 ./myApp
```

A new directory `_mps_<date>-<time>` will be created in the current directory, where `<date>` and `<time>` are placeholders for creation date and time, respectively. This directory contains the result files for each node, with the following naming scheme: `result.<hostname>`. Use this directory to analyze the data.

***NOTE:*** The date and time postfixes are not appended to the output folder name, if you use Intel® MPI Library version earlier than 5.1.2. See Setting Unique Names for Output Files for details.

# 3.4. Setting Unique Names for Output Files

To differentiate between outputs from different runs, MPI Performance Snapshot appends a postfix to all the files and folders generated: `stats.txt`, `app_stat.txt` and `_mps`.

***NOTE:*** This feature requires Intel® MPI Library 5.1.2 or newer installed on your system. Otherwise, no postfixes are added.

The default postfix pattern is:

`_%D-%T`

where:

- `%D` stands for creation date in the format: `yyyyMMdd`
- `%T` stands for creation time in the format: `hhmmss`

For example, after collecting statistics for your application you may get the following outputs:

`stats_20150101-235959.txt, app_stat_20150101-235959.txt, _mps_20150101-235959`

The postfix string is stored in the `MPS_FILE_POSTFIX` environment variable, which is set automatically by the `mpsvars` script. You can modify the postfix according to your needs by redefining this variable. For example, to include only creation time to the output names, you can set the variable to _%T:

```
$ export MPS_FILE_POSTFIX=_%T
```

You can also set the variable using the `-env/-genv` options of `mpirun`. In the example below, the `_test-n16` postfix will be added:

```
$ mpirun –mps –n 16 –genv MPS_FILE_POSTFIX=_test-n16 a.out
```

You will get the following outputs: `stats_test-n16.txt`, `app_stat_test-n16.txt`, `_mps_test-n16`.

If you do not want a postfix to be added to the output files, you can unset the `MPS_FILE_POSTFIX` variable in your user script.

# 4. Analyzing the Collected Statistics

## 4.1. Usage Instructions

To analyze the collected statistics, process the generated files `stats.txt` and `app_stat.txt`. Run the command:

```
$ mps ./stats.txt ./app_stat.txt
```

Optionally, you can process only one file. In this case, statistics information from the skipped file will not appear in the analysis results.

By default, the Summary Information is displayed.

You may also display the following diagrams using the appropriate keys (see Supported Keys):

- Counters and Memory Usage Statistics
- Function Summary for All Ranks
- MPI Time per Rank
- Collective Operations Time per Rank
- Message Sizes Summary for All Ranks
- Data Transfers per Rank-to-Rank Communication
- Data Transfers per Rank
- Data Transfers per Function
- Detailed Information for the Specified Rank

### 4.1.1. Filtering Output Data

For table diagrams, you can use filtering to display only information of interest. See the Filtering Capabilities section for details.

### 4.1.2. Displaying Statistics for Specific Rank

By default, all the diagrams show information for all ranks of the application. To display statistics only for specific rank, use the `-R` option. For example:

```
$ mps -R 1 ./stats.txt ./app_stat.txt
```

In this case, all the diagrams will display information only for rank 1.

### 4.1.3. Displaying Perormance Metrics

To display the performance metrics collected by Intel® VTune™ Amplifier XE (see here for instructions), process the result files from the `_mps` directory with `mps`:

```
$ mps _mps
```

MPI Performance Snapshot will parse the results and save the data in the `mps_counters.txt` file. The `_mps` directory will be deleted. For further analyses, use this text file:

```
$ mps ./mps_counters.txt
```

After processing input files, the Summary Information will be displayed containing the mean values of CPI and Memory Bound for your application.

# 4.2. HTML Summary Report

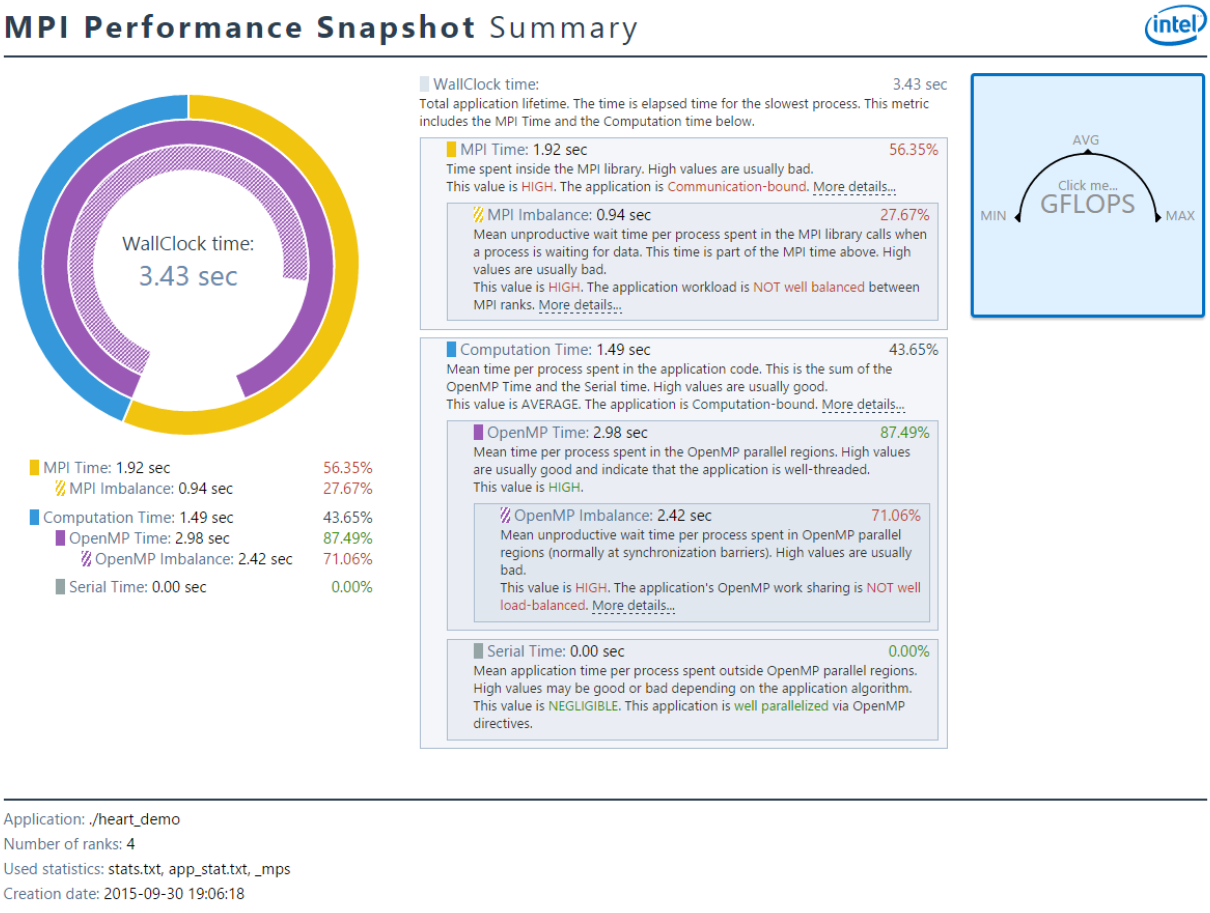
When analyzing statistics, MPI Performance Snapshot automatically generates an HTML Summary report containing the application summary. This information is the same as that printed in the Summary Information diagram.

The HTML report is generated automatically, if no options are specified. If you use any options, use the `-g` key to generate the report. Use the `-O` option to set the report filename. For example:

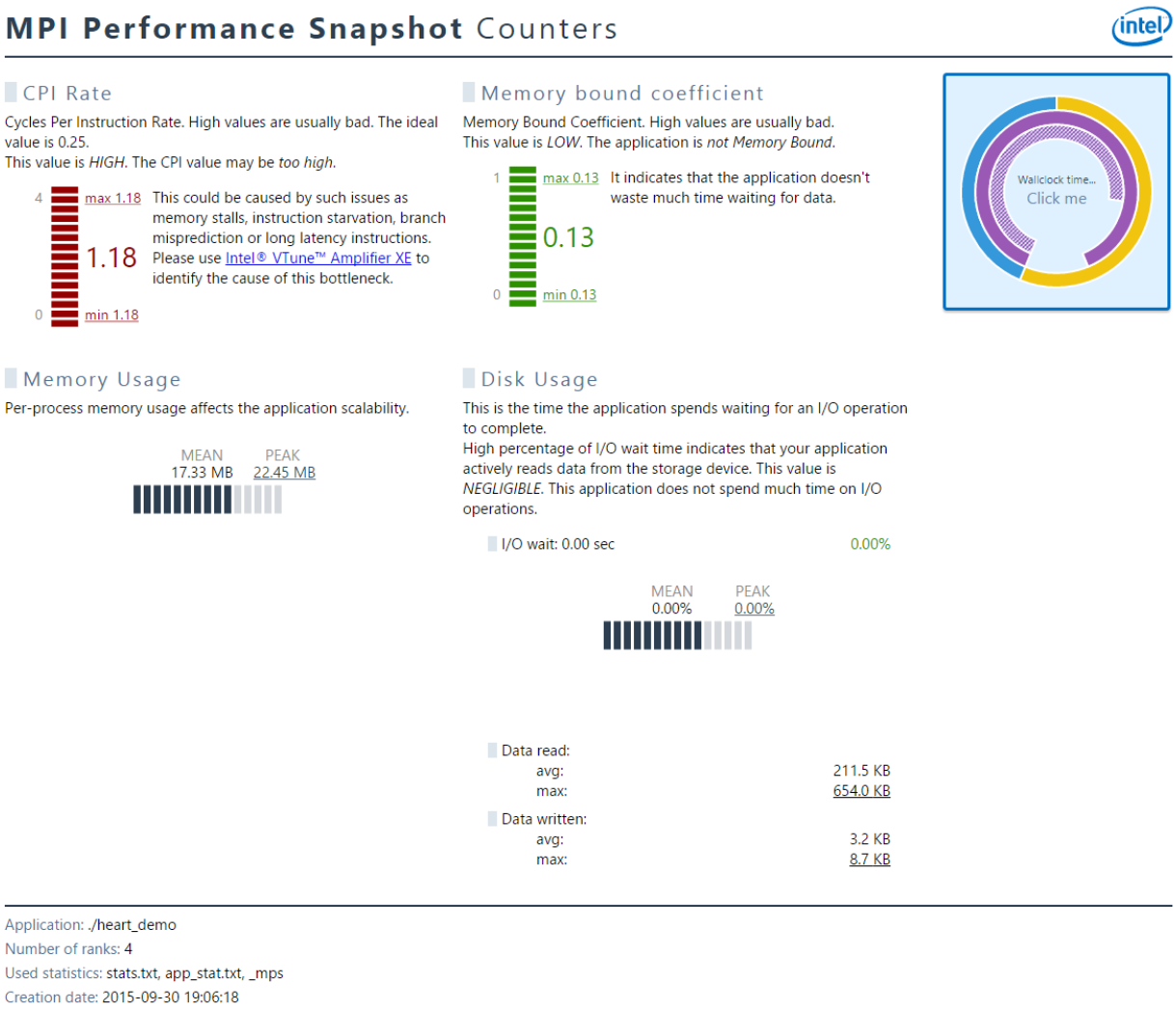
```
$ mps ./stats.txt ./app_stat.txt -g -O my_report.html
```

The generated file may look as shown below.

**Summary Page:**



**Counters Page:**




# 4.3. Diagrams

This section describes all the available diagrams for analyzing the application statistics.

***NOTE:*** Description of the diagrams below is given for the whole application. If you specify a rank with the `-R` option, information will be provided only for the specified rank.

## 4.3.1. Summary Information

### Key: -s [--summary] (default key)

Summary Information contains the essential information for novice users, which includes information about the application metrics with further recommendations and guidance.

Value for each metric related to MPI or OpenMP* is presented in the following ways:

- numeric format
- as percentage to the mean lifetime per-process
- value interpretation as NEGLIGIBLE, LOW, AVERAGE or HIGH. Depending on the metric or the application algorithm, high values may be good or bad.

***NOTE:*** If you use an Intel® compiler version lower than 15.0.1, or a different compiler, the OpenMP* metrics will not be available.

See the metrics description below.

## Wallclock Time

Total application lifetime, equal to the elapsed time for the slowest process in the application or the process specified with `-R`. This metric does not include the time spent in `MPI_Finalize`.

## MPI metrics

### MPI Time

Mean time per-process or for the process specified with `-R` spent in MPI calls. This metric does not include the time spent in `MPI_Finalize`.

**Thresholds**: NEGLIGIBLE < 2% ≤ LOW < 10% ≤ AVERAGE < 30% ≤ HIGH

### MPI Imbalance

Mean unproductive wait time per-process or for the process specified with `-R` spent in MPI calls. This is the time when a process is waiting for data or synchronizing.

**Thresholds**: NEGLIGIBLE < 1% ≤ LOW < 5% ≤ AVERAGE < 20% ≤ HIGH

### Computation Time

Mean time per-process or for the process specified with `-R` spent in the application code.

**Thresholds**: NEGLIGIBLE < 5% ≤ LOW < 20% ≤ AVERAGE < 80% ≤ HIGH

## OpenMP metrics

### OpenMP Time

Mean time per-process or for the process specified with `-R` spent in OpenMP* parallel regions.

**Thresholds**: NEGLIGIBLE < 2% ≤ LOW < 20% ≤ AVERAGE < 80% ≤ HIGH

### OpenMP Imbalance

Mean unproductive wait time per-process or for the process specified with `-R` spent in OpenMP* parallel regions (normally at synchronization barriers).

**Thresholds**: NEGLIGIBLE < 1% ≤ LOW < 5% ≤ AVERAGE < 20% ≤ HIGH

### Serial Time

Mean computation time per-process or for the process specified with `-R` spent outside OpenMP* parallel regions.

**Thresholds**: NEGLIGIBLE < 2% ≤ LOW < 10% ≤ AVERAGE < 40% ≤ HIGH

### Intel® VTune™ Amplifier XE metrics

#### GFLOPS

Billions of floating-point operations per second for all nodes.

#### CPI Rate

Cycles per instruction rate. The ideal value is `0.25`.

#### Memory Bound Coefficient

The memory bound coefficient indicates whether the application is memory bound.

### Disk Usage data

#### Data Read

The amount of data the application read from the storage device by all ranks, or by the rank specified with `-R`.

#### Data Written

The amount of data the application wrote onto the storage device by all ranks, or by the rank specified with `-R`.

#### I/O Wait Time

The time spent by the application, or by the rank specified with `-R`, waiting for I/O operations to complete.

### Memory consumption

Maximum and mean per-process memory (RAM) used by the application, or by the process specified with `-R`. Does not include memory used by the MPI library.

## 4.3.2. Counters and Memory Usage Statistics

### Key: -o [--counters]

This diagram contains the information about memory usage, hardware counters and performance metrics.

***NOTE:*** The diagram layout may vary depending on the data collected.

#### Example

```
| Counters and Memory usage statistics
|---------------------------------------------------------------------
GFlops     :            0.029
Computation:           51.00%
MPI        :           49.00%

Floating-Point instructions:          11.02%
```

```
Vectorized DP instructions :            0.00%
Memory access instructions :           43.95%

Total time                 :         119.98 sec (all ranks)

| WallClock per Rank:
            MIN:           30.00 sec (rank 0)
            MAX:           30.00 sec (rank 1)
            AVG:           30.00 sec

| Memory Usage per Rank:
          Total:            2.98 MB (all ranks)
            MIN:            0.62 MB (rank 0)
            MAX:            0.79 MB (rank 1)
            AVG:            0.75 MB

| Data Read per Rank:
          Total:           30.80 MB (all ranks)
            MIN:           15.40 MB (rank 0)
            MAX:           15.40 MB (rank 1)
            AVG:           15.40 MB

| Data Written per Rank:
          Total:          190.74 MB (all ranks)
            MIN:           95.37 MB (rank 1)
            MAX:           95.37 MB (rank 0)
            AVG:           95.37 MB

| I/O Wait Time per Rank:
          Total:            3.76 sec (all ranks)
            MIN:            1.88 sec (rank 1)
            MAX:            1.88 sec (rank 0)
            AVG:            1.88 sec
```

### Diagram Entries

| Entry | Description |
|---|---|
| GFlops | Billions of floating point operations performed by the application per second. |
| Computation | Percentage of non-MPI time in the application lifetime. |

| | |
|---|---|
| MPI | Percentage of MPI time in the application lifetime. |
| Floating-Point instructions | Percentage of floating point instructions in all instructions. |
| Vectorized DP instructions | Percentage of double precision vector instructions in all instructions. |
| Memory access instructions | Percentage of memory access instructions in all instructions. |
| Total time | Sum of the lifetime of all processes. |
| WallClock per Rank | Minimum, average and maximum process execution times. |
| Memory Usage per Rank | Information on the memory (RAM) used by the application (excluding MPI):<br>• Total memory used<br>• Minimum, average and maximum memory used by processes |
| Cycles per Instruction Rate | Minimum, average and maximum rate of cycles per instructions for each node. |
| Memory Bound Coefficient | Minimum, average and maximum memory bound values for each node. |
| Billions of Floating-point Operations Per Second | Minimum, average and maximum number of billions of floating-point operations per second for each node. |
| Data Read per Rank | Minimum, average, maximum and total data read from the storage device. |
| Data Written per Rank | Minimum, average, maximum and total data written onto the storage device. |
| I/O Wait Time per Rank | Minimum, average, maximum and total time spent by processes waiting for I/O operations to complete. |

## 4.3.3. Function Summary for All Ranks

### Key: -f [--functions]

This diagram contains information about all functions used in the application.

#### Example

```
| Function summary for all ranks
|-------------------------------------------------------------------------------------------------------
|           Function        Time(sec)          Time(%)       Volume(MB)        Volume(%)             Calls
|-------------------------------------------------------------------------------------------------------
           Allreduce          28.0045          13.5018          4109.99           5.7554            200928
          Allgatherv          23.8007           11.475          4109.99           5.7554            219248
           Alltoallv          21.3515          10.2942            12330          17.2662            219248
            Alltoall          20.6553          9.95852          4109.99           5.7554            219248
           Allgather          20.3144          9.79418          4109.99           5.7554            219248
              Reduce           19.691          9.49362          4109.95          5.75535            195248
      Reduce_scatter          16.2389          7.82925          1541.23          2.15826            195248
                Send           10.899          5.25473            16440          23.0216            877024
            Scatterv          10.2558          4.94464          4109.99           5.7554            219248
             Scatter           9.4317           4.5473          4109.99           5.7554            219248
              Gather          9.15275          4.41281             4110          5.75541            220668
             Gatherv          8.96437          4.32198          4109.99           5.7554            219248
               Bcast          8.12781          3.91865          4109.99          5.75541            220672
             Barrier        0.525528         0.253372                0                0             22062
```

```
|=============================================================================================
| TOTAL                    207.413          100          71411          100          3466586
```

### Diagram Entries

| Columns | |
|---|---|
| Function | Function name |
| Time(sec) | Total time of the function execution in all ranks (in seconds) |
| Time(%) | Percentage of the function execution time in the application lifetime |
| Volume(MB) | Amount of data transferred by the function in all ranks (in megabytes) |
| Volume(%) | Percentage of the data transferred by the function |
| Calls | Number of calls of this function in all ranks |
| **Rows** | |
| TOTAL | Total values for all columns |

## 4.3.4. MPI Time per Rank

### Key: -t [--mpi_time_per_rank]

This diagram shows the time each rank spent in MPI functions.

***NOTE:*** You should have Intel® MPI Library version 5.0.2 or higher installed on your system to view this diagram. Otherwise, this diagram will coincide with the Collective Operations Time per Rank diagram.

### Example

```
| MPI Time per Rank
|-----------------------------------------------------------
| Rank    LifeTime(sec)     MPITime(sec)       MPITime(%)
|-----------------------------------------------------------
  0001           482964           396396          82.0756
  0000           477183           387175          81.1378
|===========================================================
| TOTAL          960146           783571          81.6095
| AVG            480073           391785          81.6095
```

### Diagram Entries

| Columns | |
|---|---|
| Rank | Rank number |
| LifeTime(sec) | Total execution time of the rank (in seconds) |
| MPITime(sec) | Time spent in MPI functions (in seconds) |
| MPITime(%) | Percentage of the MPI time in the application lifetime |

| Rows | |
|---|---|
| TOTAL | Total values for the LifeTime(sec) and MPITime(sec) columns, average value for the MPITime(%) column |
| AVG | Average values for all columns |

## 4.3.5. Collective Operations Time per Rank

### Key: -c [--collop_time_per_rank]

This diagram shows the time each rank spent in MPI collective operations.

#### Example

```
| Collective Operations Time per Rank
|-----------------------------------------------------------
| Rank    LifeTime(sec)  CollOpTime(sec)    CollOpTime(%)
|-----------------------------------------------------------
  0062      1.09081e+07      4.86264e+06          44.5783
  0063      1.09081e+07      4.85122e+06          44.4736
  0055      1.09078e+07      4.84645e+06          44.4311
  0053      1.09065e+07        4.828e+06          44.2672
  0049      1.09081e+07      4.82586e+06          44.2411
| [skipped 24 lines]
  0044      1.08981e+07      3.97369e+06          36.4622
  0023       1.1117e+07      4.04246e+06          36.3627
  0008      1.11168e+07      4.02694e+06          36.2238
  0009       1.1117e+07      4.00637e+06          36.0382
  0011      1.11139e+07      4.00165e+06          36.0059
| [skipped 25 lines]
  0032      1.08979e+07      3.61322e+06          33.1551
  0030      1.08972e+07      3.57797e+06          32.8339
  0029       1.0898e+07      3.56967e+06          32.7553
  0025      1.08979e+07      3.50358e+06          32.1491
  0024      1.08912e+07      3.47305e+06          31.8885
|===========================================================
| TOTAL     7.02822e+08      2.63575e+08          37.5024
| AVG       1.09816e+07      4.11836e+06          37.5024
|
| Number of ranks = 64
```

### Diagram Entries

| Columns | |
|---|---|
| Rank | Rank number |
| LifeTime(sec) | Total execution time of the rank (in seconds) |
| CollOpTime(sec) | Time spent in MPI collective operations (in seconds) |
| CollOpTime(%) | Percentage of the MPI collective operations time in the application lifetime |
| **Rows** | |
| TOTAL | Total values for the LifeTime(sec) and CollOpTime(sec) columns, average value for the CollOpTime(%) column |
| AVG | Average values for all columns |

## 4.3.6. Message Sizes Summary for All Ranks

### Key: -m [--message_sizes]

This diagram contains summary for all message sizes. This information can be useful for the Intel® MPI Library internal thresholds tuning.

### Example

```
| Message Sizes summary for all ranks
|---------------------------------------------------------------------
| Message size(B)       Volume(MB)        Volume(%)        Transfers
|---------------------------------------------------------------------
            14000          627.691          30.1922            47013
          4194304              168          8.08087               42
          2097152              164          7.88847               82
          1048576              162          7.79227              162
           524288              161          7.74417              322
           262144            160.5          7.72012              642
           131072           160.25          7.70809             1282
            65536          160.125          7.70208             2562
            32768          125.062          6.01556             4002
            16384          62.5312          3.00778             4002
             8192          46.8984          2.25583             6003
             4096          23.4492          1.12792             6003
| [skipped 22 lines]
|=====================================================================
| TOTAL                    2078.98              100           143742
```

### Diagram Entries

| Columns | |
|---|---|
| Message size(B) | Message size (in bytes) |
| Volume(MB) | Amount of data transferred by messages of this size (in megabytes) |
| Volume(%) | Percentage of amount of data transferred by messages of this size |
| Transfers | Number of messages of this size |
| **Rows** | |
| TOTAL | Total values for all columns |

## 4.3.7. Data Transfers per Rank-to-Rank Communication

### Key: -x [--transfers_per_communication]

This diagram contains information about each rank-to-rank communication. This information can be useful for detecting imbalance in network usage.

### Example

```
| Data Transfers per Rank-to-Rank Communication
|-------------------------------------------------------------------
| Rank <-> Rank       Volume(MB)        Volume(%)        Transfers
|-------------------------------------------------------------------
  0000 <-> 0001            16440          34.7826           877039
  0001 <-> 0002          6164.98          13.0435           328872
  0002 <-> 0003          6164.98          13.0435           328872
  0000 <-> 0003          3082.49          6.52174           164483
  0000 <-> 0007          3082.49          6.52174           164483
  0003 <-> 0004          3082.49          6.52174           164436
  0004 <-> 0005          3082.49          6.52174           164436
  0005 <-> 0006          3082.49          6.52174           164436
  0006 <-> 0007          3082.49          6.52174           164436
| [skipped 4 lines]
|===================================================================
| TOTAL                  47264.9              100          2521681
| AVG                    1688.03          3.57143            90060
```

### Diagram Entries

| Columns | |
|---|---|
| Rank <-> Rank | Numbers of communicating ranks |
| Volume(MB) | Amount of data transferred in the communication (in megabytes) |

| | |
|---|---|
| Volume(%) | Percentage of amount of data transferred in the communication |
| Transfers | Number of messages in the communication |
| **Rows** | |
| TOTAL | Total values for all columns |
| AVG | Average values for all columns. The values are calculated as follows: `AVG = TOTAL / [(N*N + N)/2]`, where N is the number of ranks |

## 4.3.8. Data Transfers per Rank

### Key: -r [--transfers_per_rank]

This diagram shows the amount of data transferred by each MPI rank.

#### Example

```
| Data Transfers per Rank
|-----------------------------------------------------------
| Rank       Volume(MB)        Volume(%)        Transfers
|-----------------------------------------------------------
  0001          22604.9          47.8261          1205911
  0000          22604.9          47.8261          1206193
  0002            12330           26.087           657791
  0003            12330           26.087           657791
  0004          6164.98          13.0435           328919
  0005          6164.98          13.0435           328919
  0006          6164.98          13.0435           328919
  0007          6164.98          13.0435           328919
|===========================================================
| TOTAL         47264.9              100      2.52168e+06
| AVG           11816.2               25           630420
```

#### Diagram Entries

| **Columns** | |
|---|---|
| Rank | Rank number |
| Volume(MB) | Amount of data sent and received by the rank (in megabytes) |
| Volume(%) | Percentage of amount of data sent and received by the rank |
| Transfers | Number of messages sent and received by the rank |
| **Rows** | |
| TOTAL | Halved sum of all values for the Volume(MB) and Transfers columns (due to each byte being sent and received, it is counted twice), total value for the Volume(%) |

| | |
|---|---|
| | column |
| AVG | Average values for all columns |

## 4.3.9. Data Transfers per Function

### Key: -u [--transfers_per_function]

This diagram shows the amount of data transferred by each MPI function (for all ranks). This information can be useful for identifying functions used for data transfers.

#### Example

```
| Function summary for all ranks
|-------------------------------------------------------------------------
|           Function       Volume(MB)        Volume(%)        Transfers
|-------------------------------------------------------------------------
              Reduce          1385.98          66.6665            65128
               Csend          692.997          33.3335            78073
           Allreduce      0.000839233      4.03675e-05              132
              Gather      0.000335693       1.6147e-05               44
               Bcast      0.000312805      1.50461e-05               50
                Send       3.8147e-06      1.83489e-07                1
             Barrier                0                0              314
|=========================================================================
| TOTAL                       2078.98              100           143742
```

#### Diagram Entries

| Columns | |
|---|---|
| Function | Function name |
| Volume(MB) | Amount of data sent and received by the function (in megabytes) |
| Volume(%) | Percentage of amount of data sent and received by the function |
| Transfers | Number of function calls |
| **Rows** | |
| TOTAL | Total values for all columns |

## 4.3.10. Detailed Information for Specific Ranks

### Option: -D [--details]

This diagram shows details for processes with the minimum, mean, and maximum time values from the diagrams:

- MPI Time per Rank (use options: `-t -D`)
- Collective Operations Time per Rank (use options: `-c -D`)

Optionally, you can specify the rank to show details for. Use the `-R` option instead of `-D`. For example: `-t -R 1`.

The details include the process execution time and information about each function used by the process. Information about functions is presented in the following ways:

- Function summary
- Details for the messages transferred by each function

These two views share the table columns but their meanings are different for each view. See the diagram entries description below.

### Example

```
|############################################################
| RANK 3
|############################################################
| Life Time = 149.539 (sec)
| MPI Time  = 140.662 (sec) 94.0639%
```

**Function Summary**

| Function | Time(sec) | Time(%) | Volume(MB) | Volume(%) | Calls | AvrMsgSize(B) |
|---|---|---|---|---|---|---|
| Alltoallv | 23.9206 | 15.9962 | 12330 | 31.2704 | 109624 | 117939 |
| Alltoall | 23.8242 | 15.9317 | 2054.99 | 5.21173 | 109624 | 19656 |
| Barrier | 19.0241 | 12.7218 | 0 | 0 | 11070 | 0 |
| Allgatherv | 17.8779 | 11.9553 | 2054.99 | 5.21173 | 109624 | 19656 |
| Allgather | 16 | 10.6996 | 2054.99 | 5.21173 | 109624 | 19656 |
| Allreduce | 5.72988 | 3.83169 | 2054.99 | 5.21172 | 100272 | 21490 |
| Scatterv | 5.53121 | 3.69883 | 2054.99 | 5.21173 | 109624 | 19656 |
| Scatter | 5.40639 | 3.61536 | 2054.99 | 5.21173 | 109624 | 19656 |
| Gatherv | 5.21395 | 3.48668 | 2054.99 | 5.21173 | 109624 | 19656 |
| Gather | 5.20341 | 3.47963 | 2055 | 5.21175 | 110665 | 19472 |
| Reduce_scatter | 3.92054 | 2.62174 | 385.307 | 0.977187 | 97624 | 4139 |
| Send | 3.33028 | 2.22703 | 6164.98 | 15.6352 | 328919 | 19654 |
| Bcast | 2.95886 | 1.97865 | 2055 | 5.21173 | 110288 | 19538 |
| Reduce | 2.72119 | 1.81972 | 2054.98 | 5.21168 | 97624 | 22072 |
| TOTAL | 140.662 | 94.0639 | 39430.2 | 100 | 1623830 | 25462 |

**Details for Function Messages**

| Function | Time(sec) | Time(%) | Volume(MB) | Volume(%) | Calls | MsgSize(B) |
|---|---|---|---|---|---|---|
| Alltoallv | 23.9206 | 15.9962 | 12330 | 100 | 109624 | ALL |
| | 3.06902 | 2.05232 | 1440 | 11.6789 | 90 | 16777216 |
| | 3.04682 | 2.03747 | 1440 | 11.6789 | 180 | 8388608 |
| | 2.89341 | 1.93488 | 1440 | 11.6789 | 360 | 4194304 |
| | 2.83532 | 1.89604 | 1440 | 11.6789 | 2880 | 524288 |
| | 2.64094 | 1.76605 | 1440 | 11.6789 | 720 | 2097152 |
| | 2.53039 | 1.69212 | 1440 | 11.6789 | 1440 | 1048576 |
| | 2.47286 | 1.65365 | 1230 | 9.9757 | 4920 | 262144 |
| | 2.04191 | 1.36546 | 960 | 7.78591 | 30 | 33554432 |
| [skipped 17 lines] | | | | | | |
| Alltoall | 23.8242 | 15.9317 | 2054.99 | 100 | 109624 | ALL |
| | 3.14809 | 2.1052 | 240 | 11.6789 | 120 | 2097152 |
| | 2.96477 | 1.98261 | 240 | 11.6789 | 60 | 4194304 |
| | 2.9347 | 1.9625 | 240 | 11.6789 | 240 | 1048576 |
| | 2.77462 | 1.85545 | 240 | 11.6789 | 3840 | 65536 |

```
                          2.67651          1.78984              240          11.6789              480           524288
                          2.52501          1.68853              240          11.6789             1920           131072
                          2.50317          1.67392              240          11.6789              960           262144
                          2.20381          1.47373            187.5          9.12411             6000            32768
| [skipped 16 lines]
...
```

### Diagram Entries

| | |
|---|---|
| Life Time | Total execution time of the rank |
| MPI Time / Coll Op Time | Time spent by the rank in MPI calls and its percentage in the Life Time |
| **Function Summary Columns** | |
| Function | Function name |
| Time(sec) | Total time of the function execution in the rank (in seconds) |
| Time(%) | Percentage of the function execution time in the application lifetime |
| Volume(MB) | Amount of data transferred by the function within the rank (in megabytes) |
| Volume(%) | Percentage of the data transferred by the function |
| Calls | Number of calls of the function in the rank |
| AvrMsgSize(B) | Average message size transferred by the function within the rank (in bytes) |
| **Details for Messages Columns** | |
| Function | Function name |
| Time(sec) | Transfer time for the message of this size (in seconds) |
| Time(%) | Percentage of the message transfer time in the function time |
| Volume(MB) | Amount of data transferred by the message of this size (in megabytes) |
| Volume(%) | Percentage of the data transferred by the message in the data transferred by the function |
| Calls | Transfers of messages of this size |
| MsgSize(B) | Size of the message transferred by the function (in bytes) |
| **Rows** | |
| TOTAL | Total values for all columns |

# 4.4. Filtering Capabilities

MPI Performance Snapshot is oriented to highly scalable applications with thousands of MPI ranks running in parallel. For such applications the aggregated result tables may contain thousands of lines, which makes the results difficult to read.

The tool provides filtering capabilities to make the results tables shorter.

There are two independent filters in the tool, which can be applied to any table diagram:

- Filtering by key metric
- Filtering by number of lines

***NOTE:*** Filters do not skip one line, as the message about the skipped lines takes one line.

Filters are enabled by default, and the tool behavior is the same as with the key combination:

```
–V 1 –T 1 –N 5
```

To disable the filters, use the `-F` key, or the following key combination:

```
–V 0 –T 0 –N 0
```

For details, see the sections below.

## 4.4.1. Filtering by Key Metric

All diagrams provided by the tool are sorted by its key metric. The key metric for a diagram may be:

- TIME – controlled by the `-T` (`--time_threshold`) key
- VOLUME – controlled by the `-V` (`--volume_threshold`) key

If a diagram is sorted by TIME, the time threshold is applied for the table filtering. VOLUME threshold is not applied to this diagram. And vice versa: the volume threshold is applied to diagrams sorted by VOLUME and does not affect diagrams sorted by TIME.

You can use both thresholds only when the tool displays several diagrams sorted by different key metrics.

Default value for both key metric thresholds is 1%. This is equivalent to the following combination of command line options: `–T 1 –V 1`

Zero value `-V 0` disables filtering by VOLUME, and the `-T 0` filter disables filtering by TIME.

Maximum value for the thresholds is 100%.

The filter hides the lines with the key metric percentage less than the filter threshold. For example, lines with `Volume(%)` less than 1% are hidden in the table diagrams sorted by VOLUME.

***NOTE:*** By default, filter by number of lines is enabled. To see independent result of the metric filter usage, disable this filter. Use the key `-N 0`.

### Examples

In all examples, the Message Size diagram is used. The diagram is sorted by VOLUME, so the `-V` key is used for filtering. The `-T` key does not affect the output.

#### Example 1

To show all lines in the diagram, enter:

```
$ mps -V 0 -N 0 -m stats.txt
```

Output:

```
| Message Sizes summary for all ranks
|-----------------------------------------------------------------------
| Message size(B)       Volume(MB)        Volume(%)        Transfers
|-----------------------------------------------------------------------
           262144             8340          11.6789            33360
```

```
          524288              8340           11.6789            16680
         1048576              8340           11.6789             8340
         2097152              8280           11.5948             4140
         4194304              8160           11.4268             2040
          131072              8130           11.3848            65040
           65536              7650           10.7126           122400
           32768              5805             8.129           185760
           16384           2966.25           4.15377           189840
            8192              1500           2.10052           192000
         8388608              1440            2.0165              180
        16777216               960           1.34433               60
            4096               750           1.05026           192000
            2048               375          0.525129           192000
            1024             187.5          0.262565           192000
             512             93.75          0.131282           192000
             256            46.875         0.0656411           192000
             128           23.4375         0.0328206           192000
              64           11.7188         0.0164103           192000
              32           5.85938        0.00820514           192000
              16           2.92975        0.00410266           192004
               8           1.49738        0.00209684           196264
               4          0.749222        0.00104917           196404
               2          0.308998       0.000432704           162004
               1          0.143055       0.000200326           150004
              60       0.000114441      1.60257e-007                2
              12      2.28882e-005      3.20513e-008                2
               0                 0                 0           214062
|=================================================================
| TOTAL                     71411               100          3466586
```

## Example 2

To apply the default VALUE filter to the Message Size diagram, enter:

```
$ mps -N 0 -m stats.txt
```

Output:

```
| Message Sizes summary for all ranks
|-----------------------------------------------------------------
| Message size(B)       Volume(MB)        Volume(%)        Transfers
|-----------------------------------------------------------------
          262144              8340           11.6789            33360
          524288              8340           11.6789            16680
```

```
          1048576              8340          11.6789              8340
          2097152              8280          11.5948              4140
          4194304              8160          11.4268              2040
           131072              8130          11.3848             65040
            65536              7650          10.7126            122400
            32768              5805            8.129            185760
            16384           2966.25          4.15377            189840
             8192              1500          2.10052            192000
          8388608              1440           2.0165               180
         16777216               960          1.34433                60
             4096               750          1.05026            192000
| [skipped 15 lines]
|=====================================================================
| TOTAL                      71411              100           3466586
```

In this example, all lines with the `Volume(%)` value less than 1% are skipped.

### Example 3

To skip all lines with the `Volume(%)` value less than 10%, enter:

```
$ mps –V 10 -N 0 -m stats.txt
```

Output:

```
| Message Sizes summary for all ranks
|---------------------------------------------------------------------
| Message size(B)       Volume(MB)        Volume(%)        Transfers
|---------------------------------------------------------------------
           262144              8340          11.6789             33360
           524288              8340          11.6789             16680
          1048576              8340          11.6789              8340
          2097152              8280          11.5948              4140
          4194304              8160          11.4268              2040
           131072              8130          11.3848             65040
            65536              7650          10.7126            122400
| [skipped 21 lines]
|=====================================================================
| TOTAL                      71411              100           3466586
```

## 4.4.2. Filtering by Number of Lines

This filter can be applied to any table diagram sorted by any metric, since it works with the table lines.

This filter enables you to see only lines of interest. Usually, these lines are the top, the bottom, and the middle ones. The filter key `-N` defines the number of lines to be shown in each part.

The default value is five, therefore the tool displays 15 lines overall (five top lines, five middle, and 5 bottom lines).

Zero value disables the filter: `-N 0`. Maximum value for the filter is unlimited.

***NOTE:*** By default, filter by key metric is enabled. To see independent results of the filter by number of lines, disable this filter. Use the key combination `–T 0 –V 0`, or only one of the keys, depending on sorting filter of the diagram.

If you want to see only particular lines, use external text filtering tools. The MPI Performance Snapshot analyzer does not provide such functionality.

## Examples

In all examples, the Message Size diagram is used. The diagram is sorted by VOLUME, so the `-V` key is used for filtering. The `-T` key does not affect the output.

### Example 1

To show all lines in the diagram, enter:

```
$ mps -V 0 -N 0 -m stats.txt
```

For output, see Example 1 in the previous section.

### Example 2

If you set a number bigger than the table size, all lines will be displayed. For example, to display all lines of the table, enter the following command:

```
$ mps -V 0 -N 1000 -m stats.txt
```

For output, see Example 1 in the previous section.

### Example 3

To apply the default filter value (five lines), enter:

```
$ mps -V 0 -m stats.txt
```

Output:

```
| Message Sizes summary for all ranks
|---------------------------------------------------------------------
| Message size(B)       Volume(MB)        Volume(%)        Transfers
|---------------------------------------------------------------------
           262144             8340          11.6789            33360
           524288             8340          11.6789            16680
          1048576             8340          11.6789             8340
          2097152             8280          11.5948             4140
          4194304             8160          11.4268             2040
| [skipped 6 lines]
         16777216              960          1.34433               60
             4096              750          1.05026           192000
             2048              375         0.525129           192000
```

```
             1024            187.5         0.262565           192000
              512            93.75         0.131282           192000
| [skipped 7 lines]
                2         0.308998      0.000432704           162004
                1         0.143055      0.000200326           150004
               60      0.000114441     1.60257e-007                2
               12     2.28882e-005     3.20513e-008                2
                0                0                0           214062
|==================================================================
| TOTAL                      71411              100          3466586
```

### Example 4

To display two lines in the top, the middle and the bottom (six lines overall), enter:

```
$ mps -V 0 –N 2 -m stats.txt
```

Output:

```
| Message Sizes summary for all ranks
|-----------------------------------------------------------------
| Message size(B)       Volume(MB)        Volume(%)        Transfers
|-----------------------------------------------------------------
           262144             8340          11.6789            33360
           524288             8340          11.6789            16680
| [skipped 11 lines]
             2048              375         0.525129           192000
             1024            187.5         0.262565           192000
| [skipped 11 lines]
               12     2.28882e-005     3.20513e-008                2
                0                0                0           214062
|==================================================================
| TOTAL                      71411              100          3466586
```

## 4.4.3. Using Filter Combinations

By default, both filters described above are enabled. The default threshold for TIME and VOLUME is 1%, default value for the number of lines filter is five. Thus, the default tool behavior is the same as with the key combination `-V 1 -T 1 -N 5`.

***NOTE:*** Filter by key metric is applied first. Filtering by number of lines filters only the remaining lines.

### Examples

In all examples, the Message Size diagram containing 28 lines is used.

### Example 1

To use the default options, enter:

```
$ mps -m stats.txt
```

This command line is equal to this one:

```
$ mps -m –V 1 –T 1 –N 5 stats.txt
```

Output:

```
| Message Sizes summary for all ranks
|-----------------------------------------------------------------------
| Message size(B)       Volume(MB)        Volume(%)        Transfers
|-----------------------------------------------------------------------
           262144             8340          11.6789            33360
           524288             8340          11.6789            16680
          1048576             8340          11.6789             8340
          2097152             8280          11.5948             4140
          4194304             8160          11.4268             2040
           131072             8130          11.3848            65040
            65536             7650          10.7126           122400
            32768             5805            8.129           185760
            16384          2966.25          4.15377           189840
             8192             1500          2.10052           192000
          8388608             1440           2.0165              180
         16777216              960          1.34433               60
             4096              750          1.05026           192000
| [skipped 15 lines]
|=======================================================================
| TOTAL                       71411              100          3466586
```

The volume filter hides the bottom lines with `Volume(%)` less than 1% (15 lines). The remaining 13 lines are less than the number of lines to be displayed. Therefore, the filter by number of lines will not be applied.

## Example 2

To hide the lines with VOLUME less than 1% (default threshold), and display two top lines, two middle lines, and two bottom lines of the remaining lines, enter:

```
$ mps –N 2 -m stats.txt
```

or

```
$ mps –V 1 –N 2 -m stats.txt
```

Output:

```
| Message Sizes summary for all ranks
|-----------------------------------------------------------------------
| Message size(B)       Volume(MB)        Volume(%)        Transfers
|-----------------------------------------------------------------------
           262144             8340          11.6789            33360
```

```
           524288              8340          11.6789             16680
| [skipped 3 lines]
           131072              8130          11.3848             65040
            65536              7650          10.7126            122400
| [skipped 4 lines]
         16777216               960          1.34433                60
             4096               750          1.05026            192000
| [skipped 15 lines]
|=================================================================
| TOTAL                      71411              100           3466586
```

# 5. Command Line Reference

MPI Performance Snapshot has the following command-line syntax:

```
$ mps [keys] [options] <stat_files>
```

where:

- *[keys]* are the MPI Performance Snapshot keys. See Supported Keys.
- *[options]* are the MPI Performance Snapshot options. See Supported Options.
- *<stats_files>* are the paths to the statistics files:
  - `stats.txt` containing the Intel® MPI Library native statistics
  - `app_stat.txt` containing statistics collected by MPI Performance Snapshot
  - `mps_counters.txt` containing statistics collected by Intel® VTune™ Amplifier XE

  You can specify any of these files in any combination.

## 5.1. Supported Keys

MPI Performance Snapshot supports the following keys:

### General Keys

| Key | Description |
|---|---|
| -h [--help] | Show help and exit. |
| -v [--version] | Show the application version number and exit. |
| -a [--all] | Show all available diagrams except for Collective Operations Time per Rank |

### Specific Keys

| Key | Description | Filtered By | Input File |
|---|---|---|---|
| -s [--summary] | Show the Summary Information. Default key. | N/A | stats.txt<br>app_stat.txt<br>mps_counters.txt |
| -g [--html_summary] | Generate an HTML summary report. Set the file name with the -O option. Default key. | N/A | stats.txt<br>app_stat.txt<br>mps_counters.txt |
| -o [--counters] | Show the Counters and Memory usage statistics. | N/A | app_stat.txt<br>mps_counters.txt |
| -f [--functions] | Show the Function summary for all ranks. | time (-T) | stats.txt |

| | | | |
|---|---|---|---|
| -t [--mpi_time_per_rank] | Show the MPI Time per Rank diagram. | time (-T) | stats.txt |
| -c [--collop_time_per_rank] | Show the Collective Operations Timer per Rank diagram. | time (-T) | stats.txt |
| -m [--message_sizes] | Show the Message Sizes summary for all ranks. | volume (-V) | stats.txt |
| -x [--transfers_per_communication] | Show the Data Transfers per Rank-to-Rank Communication diagram. | volume (-V) | stats.txt |
| -r [--transfers_per_rank] | Show the Data Transfers per Rank diagram. | volume (-V) | stats.txt |
| -u [--transfers_per_function] | Show the Data Transfers per Function summary for all ranks. | volume (-V) | stats.txt |

# 5.2. Supported Options

MPI Performance Snapshot supports the following options:

| Option | Description |
|---|---|
| -R [--rank ] arg | Show detailed information for the specified rank. |
| -D [--details] | Show details for the ranks with minimum, mean, and maximum time values from the selected diagram. Applies to the keys -c and -t. Use the time filter, see the -T option. |
| -I [--show_internals] | Show additional information about the internal MPI functions: Csend, CSendRecv. By default, they are not displayed in the aggregated views. |
| -V [--volume_threshold] arg | Threshold for data volume in % of the total data volume transferred. Lines with volume less than the threshold will be skipped. Set it to 0 to disable the filter. Default: 1% |
| -T [--time-treshold] arg | Threshold for time in % of the total process lifetime. Lines with time less than the threshold will be skipped. Set it to 0 to disable the filter. Default: 1% |
| -N [--number_of_lines] arg | Number of lines to be shown in the table in the beginning, middle and ending part. All other lines will be skipped. Set it to 0 to disable the filter. Default: 5 for each part (15 in total). |
| -F [--no_filters] | Disable all filters. The option is equivalent to the string `-V 0 -T 0 -N 0`. |
| -O [--html_summary_file] arg | Set the name for the generated HTML summary report. Default: `mps_report.html` in the current folder. |