

# h5perf

[Expand all](#) [Collapse all](#)

- [Jump to ...](#)
- [Summary](#)
- [Description](#)
- [Example](#)
- [Switch language ...](#)
- [C](#)
- [C++](#)
- [FORTRAN](#)
- [JAVA](#)

[Summary](#)  
[Description](#)  
[Example](#)  
[JAVA](#)  
[FORTRAN](#)  
[C++](#)  
[C](#)

## h5perf

Tests Parallel HDF5 performance.

### Syntax:

```
h5perf [-h | --help]  
h5perf [options]
```

### Description:

`h5perf` is a tool for testing the performance of the Parallel HDF5 library. The tool can perform testing with 1-dimensional and 2-dimensional buffers and datasets. For details regarding data organization and access, see the "[h5perf User Guide](#)."

The following environment variables have the following effects on `h5perf` behavior:

HDF5_NOCLEANUP	If set, <code>h5perf</code> does not remove data files. ( <i>Default:</i> Data files are removed.)
HDF5_MPI_INFO	Must be set to a string containing a list of semi-colon separated key=value pairs for the MPI <code>INFO</code> object. Example:
HDF5_PARAPREFIX	Sets the prefix for parallel output data files.

### Options and Parameters:

These terms are used as follows in this section:	
<i>file</i>	A filename

<i>size</i>	A size specifier, expressed as an integer greater than or equal to 0 (zero) followed by a size indicator: K for kilobytes (1024 bytes) M for megabytes (1048576 bytes) G for gigabytes (1073741824 bytes) Example: 37M specifies 37 megabytes or 38797312 bytes.						
<i>N</i>	An integer greater than or equal to 0 (zero)						
<b>-h, --help</b>							
	Prints a usage message and exits.						
<b>-a <i>size</i>, --align=<i>size</i></b>							
	Specifies the alignment of objects in the HDF5 file. (Default: 1)						
<b>-A <i>api_list</i>, --api=<i>api_list</i></b>							
	Specifies which APIs to test. <i>api_list</i> is a comma-separated list with the following valid values:						
	<table border="1"> <tr> <td>phdf5</td> <td>Parallel HDF5</td> </tr> <tr> <td>mpio</td> <td>MPI-I/O</td> </tr> <tr> <td>posix</td> <td>POSIX</td> </tr> </table>	phdf5	Parallel HDF5	mpio	MPI-I/O	posix	POSIX
phdf5	Parallel HDF5						
mpio	MPI-I/O						
posix	POSIX						
	(Default: All APIs) Example, --api=mpio,phdf5 specifies that the MPI I/O and Parallel HDF5 APIs are to be monitored.						
<b>-B <i>size</i>, --block-size=<i>size</i></b>							
	Controls the block size within the transfer buffer. (Default: Half the number of bytes per process per dataset)  Block size versus transfer buffer size: The <i>transfer buffer size</i> is the size of a buffer in memory. The data in that buffer is broken into <i>block size</i> pieces and written to the file.  Transfer buffer size is discussed below with the -x (or --min-xfer-size) and -X (or --max-xfer-size) options.  The pattern in which the blocks are written to the file is described in the discussion of the -I (or -interleaved) option.						
<b>-c, --chunk</b>							
	Creates HDF5 datasets in chunked layout. (Default: Off)						
<b>-C, --collective</b>							
	Use collective I/O for the MPI I/O and Parallel HDF5 APIs. (Default: Off, i.e., independent I/O)  If this option is set and the MPI-I/O and PHDF5 APIs are in use, all the blocks of every process will be written at once with an MPI derived type.						
<b>-d <i>N</i>, --num-dsets<math>N</math></b>							
	Sets the number of datasets per file. (Default: 1)						
<b>-D <i>debug_flags</i>, --debug=<i>debug_flags</i></b>							

Sets the debugging level. *debug\_flags* is a comma-separated list of debugging flags with the following valid values:

1	Minimal debugging
2	Moderate debugging (“not quite everything”)
3	Extensive debugging (“everything”)
4	All possible debugging (“the kitchen sink”)
r	Raw data I/O throughput information
t	Times, in additions to throughputs
v	Verify data correctness

(Default: No debugging)

Example: `--debug=2,r,t` specifies to run a moderate level of debugging while collecting raw data I/O throughput information and verifying the correctness of the data.

Throughput values are computed by dividing the total amount of transferred data (excluding metadata) over the time spent by the slowest process. Several time counters are defined to measure the data transfer time and the total elapsed time; the latter includes the time spent during file open and close operations. A number of iterations can be specified with the option `-i` (or `--num-iterations`) to create the desired population of measurements from which maximum, minimum, and average values can be obtained. The timing scheme is the following:

```
for each iteration
    initialize elapsed time counter
    initialize data transfer time counter
    for each file
        start and accumulate elapsed time
counter
        file open
        start and accumulate data transfer
time counter
        access entire file
        stop data transfer time counter
        file close
        stop elapsed time counter
    end file
    save elapsed time counter
    save data transfer time counter
end iteration
```

The reported write throughput is based on the accumulated data transfer time, while the write open-close throughput uses the accumulated elapsed time.

`-e size, --num-bytes=size`

Specifies the number of bytes per process per dataset.  
(Default: 256K for 1D, 8K for 2D)

Depending on the selected geometry, each test dataset can be a linear array of size *bytes-per-process \* num-processes* or a square array of size *(bytes-per-process \* num-processes) x (bytes-per-process \* num-processes)*. The number of processes is set by the `-p` (or `--min-num-processes`) and `-P` (or `--max-num-processes`) options.

-F <i>N</i> , --num-files= <i>N</i>	Specifies the number of files. (Default: 1)
-g, --geometry	Selects 2D geometry for testing. (Default: Off, i.e., 1D geometry)
-i <i>N</i> , --num-iterations= <i>N</i>	Sets the number of iterations to perform. (Default: 1)
-I, --interleaved	<p>Sets interleaved block I/O. (Default: Contiguous block I/O)</p> <p>Interleaved and contiguous patterns in 1D geometry: When a contiguous access pattern is chosen, the dataset is evenly divided into <i>num-processes</i> regions and each process writes data to its assigned region. When interleaved blocks are written to a dataset, space for the first block of the first process is allocated in the dataset, then space is allocated for the first block of the second process, etc., until space is allocated for the first block of each process, then space is allocated for the second block of the first process, the second block of the second process, etc.</p> <p>For example, with a three process run, 512KB bytes-per-process, 256KB transfer buffer size, and 64KB block size, each process must issue two transfer requests to complete access to the dataset.</p> <p>Contiguous blocks of the first transfer request are written as follows: 1111----2222----3333----</p> <p>Interleaved blocks of the first transfer request are written as follows: 123123123123-----</p> <p>The actual number of I/O operations involved in a transfer request depends on the access pattern and communication mode. When using independent I/O with an interleaved access pattern, each process performs four small non-contiguous I/O operations per transfer request. If collective I/O is turned on, the combined content of the buffers of the three processes will be written using one collective I/O operation per transfer request.</p> <p>For details regarding the impact of performance and access patterns in 2D, see the <a href="#">"h5perf User Guide."</a></p>
-m, --mpi-posix	This option is no longer available.
-n, --no-fill	Specifies to not write fill values to HDF5 datasets. This option is supported only in HDF5 Release v1.6 or later. (Default: Off, i.e., write fill values)
-o <i>file</i> , --output= <i>file</i>	Sets the output file for raw data to <i>file</i> . (Default: None)
-p <i>N</i> , --min-num-processes= <i>N</i>	Sets the minimum number of processes to be used. (Default: 1)
-P <i>N</i> , --max-num-processes= <i>N</i>	Sets the maximum number of processes to be used. (Default: All MPI_COMM_WORLD processes)
-T <i>size</i> , --threshold= <i>size</i>	Sets the threshold for alignment of objects in the HDF5 file. (Default: 1)
-w, --write-only	Performs only write tests, not read tests. (Default: Read and write tests)

<p><code>-x size, --min-xfer-size=size</code></p>	<p>Sets the minimum transfer buffer size. (<i>Default:</i> Half the number of bytes per processor per dataset)</p> <p>This option and the <code>-x size</code> option (or <code>--max-xfer-size=size</code>) control <i>transfer-buffer-size</i>, the size of the transfer buffer in memory. In 1D geometry, the transfer buffer is a linear array of size <i>transfer-buffer-size</i>. In 2D geometry, the transfer buffer is a rectangular array of size <i>block-size</i> × <i>transfer-buffer-size</i>, or <i>transfer-buffer-size</i> × <i>block-size</i> if the interleaved access pattern is selected.</p>
<p><code>-X size, --max-xfer-size=size</code></p>	<p>Sets the maximum transfer buffer size. (<i>Default:</i> The number of bytes per processor per dataset)</p>

**Exit Status:**

<p>0</p>	<p>Succeeded.</p>
<p>&gt; 0</p>	<p>An error occurred.</p>

**History:**

Release	Change
<p>1.6.0</p>	<p>Tool introduced in this release.</p>
<p>1.6.8 and 1.8.0</p>	<p>Option <code>-g, --geometry</code> introduced in this release.</p>