# TACH BENCHMARK ENGINE REVISION 3.0
# 6/1/2004

## Sequential Read or Write Benchmark

This benchmark is intended to determine the maximum read or write speed on the device at various locations. Hard drives, for example, have faster read/write speeds at the inside edge (mapped to block 0) of the device, so it is useful to see the difference in speed. The maximum sustained speed (lowest speed on the device) is an important metric for real time sequential access applications. The maximum speed may be important to a power user that would like to partition the drive into a high speed partition and low speed partition.

Using the test parameters the engine will examine individual zones or the entire device and calculate a speed for each zone. The speed for the zone is calculated with one of two methods:

Read (Quick or Full), Write (Full Only)

Each block read or written will be timed separately, but only the median result is used (not the average). By accepting the median result we hope to isolate some of the unavoidable operating system interference and provide a more consistent test. Example: An 8 megabyte test size with 64kb blocks will consist of 128 separate read or write operations. After the 128 operations the results are sorted in ascending order and the value from the median (sorted item 64) is saved.

Write (Quick Only)

The test will time all writes over the test size range. After all blocks are written, a flush command is issued to the device and the end time is determined. The test result is the average speed over the entire range, including the flush. This is required on a quick test as the affect of the write cache in the device will be substantial on small zones. This test will not be as consistent as the flush command and loss of streaming to the device will be variable.

**PARAMETERS**

Write (Boolean)

If true, a write test is performed. If false, read testing is performed.

Quick (Boolean)

If true, a quick test covering a selection of areas (zones) from the hard drive is tested. If false, the drive is still divided into zones, but all zones on the drive are tested – this test can be very lengthy in execution as potentially every block on the drive is read.

Block Size (int32)

Specifies the block size the engine will use for read or write requests to the drive, in bytes. This should be a multiple of the media block size, for example, if the media block size is 512, the test block size should be 512, 1024, 2048, 4096, etc. There is no absolute upper limit, but the host operating system may potentially break larger blocks up into multiple reads, negating the affect of the larger block size and perhaps even producing lower performance due to the extra command overhead.

Max zones (int32)

This parameter is applicable to quick mode only. By default, quick mode will test one zone per 512 megabytes of device capacity, minimum two zones. However, on a 200 gigabyte drive this would test 400 zones, which may be a time consuming test. This parameter specifies a maximum number of zones to test, these zones will be equally spread out on the

device so testing results will be comparable to a larger test.

Pre-test size (int32)

The pre-test size specifies the number of bytes to read before starting the actual benchmark. This is primarily useful for quick tests as some drives respond differently to the seek required to switch zones. By allowing a small, non-clocked pre-test the benchmark results will be much more replicable and equivalent to a full test. The pre-test size must be a multiple of the test block size.

Test size (int32)

The test size specifies the number of bytes to read during the test. This test size must be a multiple of the test block size.

**OUTPUT**

The test will output the speed of each zone in bytes per second, and the position of the zone on the device in bytes. The average result is usually what you would present to a user as it's the average speed in that zone. The median result seems to create a useful stair step graph showing the different speed zones on the drive, but it will tend to be at exaggerated speeds compared to the average test.

## Interface Burst Speed Benchmark

This benchmark isolates the speed of the interface (IDE,SCSI,1394,USB,etc) that the device is attached to. The burst speed is the maximum speed data can be transferred from a device's internal cache memory to the CPU. Burst speeds tend to be important when running more than one device on a single interface – the more available burst speed, the better additional devices will perform. This test is also useful as a tool for determining if the drive and interface are operating as expected. If a ATA100 hard drive is attached to an ATA100 IDE controller and the burst result is low (under 66MB/s) it is an indicator that the configuration or compatibility of the devices is incorrect. This can not yield the true burst speed of the interface as there is software and command overhead that slows down the results. We generally achieve results with 10-12% of theoretical maximum – for instance, on an ATA100 bus, we get a burst speed of 90MB/s.

Starting with the maximum block size, the test will send a read request to the device. After the read request the benchmark will wait for a short time for the read cache of the device to fill with data and then send a read request to the next sequential sector. The block size will be divided in half (example: 512kb becomes 256kb) and the two reads are repeated. This cycle repeats until the block size is smaller than the minimum specified block size. The entire loop repeats for the repetition count specified. The single highest speed read is reported as the burst speed.

Planned enhancement: allow write burst speed testing.

**PARAMETERS**

Maximum Block Size (int32)

The maximum block size is specified in bytes. The benchmark uses this value as an upper limit for the read request sent during burst testing. As in the sequential read/write block size there is no absolute upper limit, but the OS may break larger (>64k) requests into more than one command. The maximum block size must be divisible by the device block size.

Minimum Block Size (int32)

Also specified in bytes, this is the minimum block size used during burst testing. The

minimum block size must be divisible by the device block size.

Repetitions (int32)

This specifies the number of times to execute the test in the various block sizes.

Pre-read Delay (int32)

The time, in milliseconds, to delay before doing the pre-read operation, after the 'seed' read. The 'seed' read is a one sector read performed before the burst test, it allows the drive to start filling the onboard read ahead cache with the next logical sectors.

Pre-read Block Size (int32)

Block size for the pre-read operation. The pre-read is a read performed before the measured burst read. Pre-read is used to help clear some small hardware caches in the drive controller.

Pre-read Count (int32)

Number of times to execute the a pre-read at the pre-read block size.

Write (Boolean)

If true a write burst operation is performed instead of a read burst.

**OUTPUT**

The test will output one data point per tested block size plus the ideal burst speed (point 0). The value (use maximum) is the burst speed in bytes per second. The position is the block size, in bytes, at which this speed was attained. Point 0, ideal burst point, is an interpolated number resulting from the highest burst speed and the next lowest, attempting to eliminate command overhead and pinpoint the data transfer alone. Point 0 is often confused by RAID controllers and other controllers with on-board cache, it is recommended that results are verified by looking at the other data points.

## Random Access Time Benchmark

This benchmark determines the random access speed of the device. Random access is the average time it takes to retrieve a randomly located sector from the device. Lower random access speeds result in better application and database performance. Random access times can also give insight into the efficiency of the interface a device is attached to by comparing a directly connected device (IDE hard drive) against the same device attached with an internal bus (1394).

To determine the random access time the benchmark will randomly read from a number of sectors on the disk. All reads are timed and the result is the average time.

Planned enhancement: allow random write time testing.

**PARAMETERS**

Read Count (int32)

Specifies the number of reads to perform.

**OUTPUT**

The value (use average result) is the random access time, in hundredths of milliseconds (milliseconds*100). The position is unused and will be zero.

## Flash Write Benchmark-Future Version of HD Tach

This benchmark determines the write speed of a flash device. Flash devices have unique characteristics including reserved areas on the device for spare block mapping of bad sectors. Because using a spare block is faster (no erase cycle required) a flash device must be written to on all sectors before testing. Benchmark applications implementing this test should provide an option to perform this.

The flash write benchmark will start with a series of writes that are not timed to the same starting sector. The goal of these writes is to further use up the spare area available to the device. After these writes are completed, a series of timed writes to the same starting sector are performed. This single highest write speed is the benchmark result.

**PARAMETERS**

Spare Percentage (int32)

A value of 0 to 100 that specifies what percentage spares the device uses. If in doubt, 5 is a good number. This value multiplied by the device size and divided by the block size will determine the number of pre writes done by the benchmark.

Block Size (int32)

Specifies the block size the engine will use for writes to the drive, in bytes. This should be a multiple of the media block size, for example, if the media block size is 512, the test block size should be 512, 1024, 2048, 4096, etc. There is no absolute upper limit, but the host operating system may potentially break larger blocks up into multiple reads, negating the affect of the larger block size and perhaps even producing lower performance due to the extra command overhead.

Repetitions (int32)

Specifies the number of writes to perform in the timed portion.

**OUTPUT**

The test will output one data point. The value is the maximum write speed, in bytes per second. The position is unused and will be zero.

## CPU Utilization

This benchmark determines the CPU load of the storage device. Lower CPU utilization is better. High CPU utilization (above 15%) generally indicates that a poor controller is being used, DMA needs to be enabled, or a driver update is required.

**PARAMETERS**

Idle test time (int32)

The number of milliseconds the system should be tested under idle load.

Load test time (int32)

The number of milliseconds the system should be test while reading at max speed from the storage device.

Block Size (int32)

Specifies the block size the engine will use for read requests to the device, in bytes. Larger block sizes should tend to have lower CPU utilization.

**OUTPUT**

All results are in tenths of a percent. Point 0 is the adjusted (load – idle) result, average would be the most useful result. Point 1 and Point 2 are the idle and loaded results, respectively.