Blunk Microsystems

liteFS-NAND Benchmarks

1. Throughput

These benchmarks were performed using a 2KB page SLC NAND on Blunk's ARM9-based Flash Test System with HW ECC. The tests measured the following things:

- Read speed during a streaming read
- Best-case write speed: a streaming write to newly-erased volume
- Worst-case write speed: a streaming write to volume full of dirty sectors

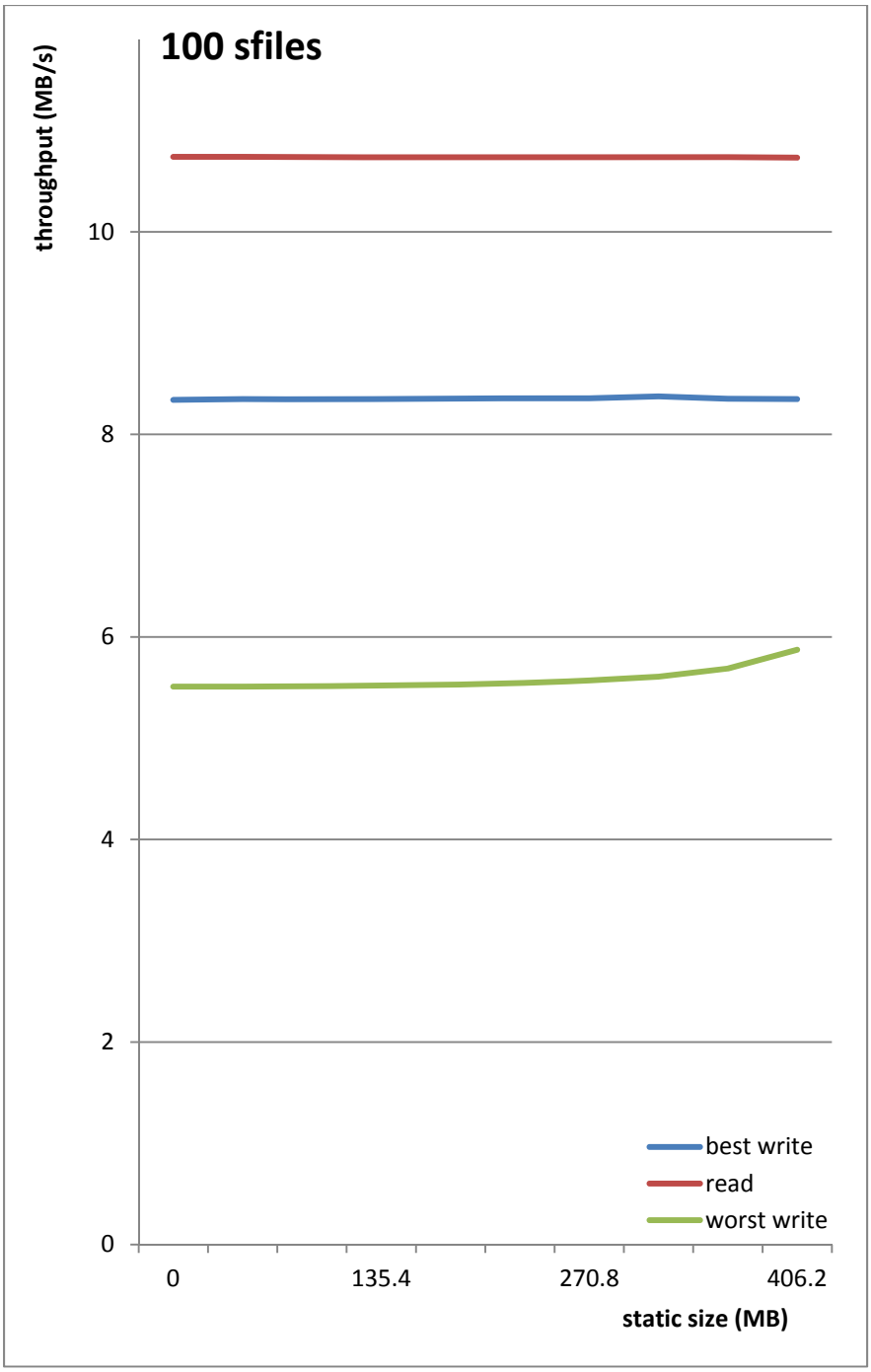
Write throughput was measured by writing to a single file until the volume was full. Read throughput was measured by reading the file back. The measurements were repeated while varying two things:

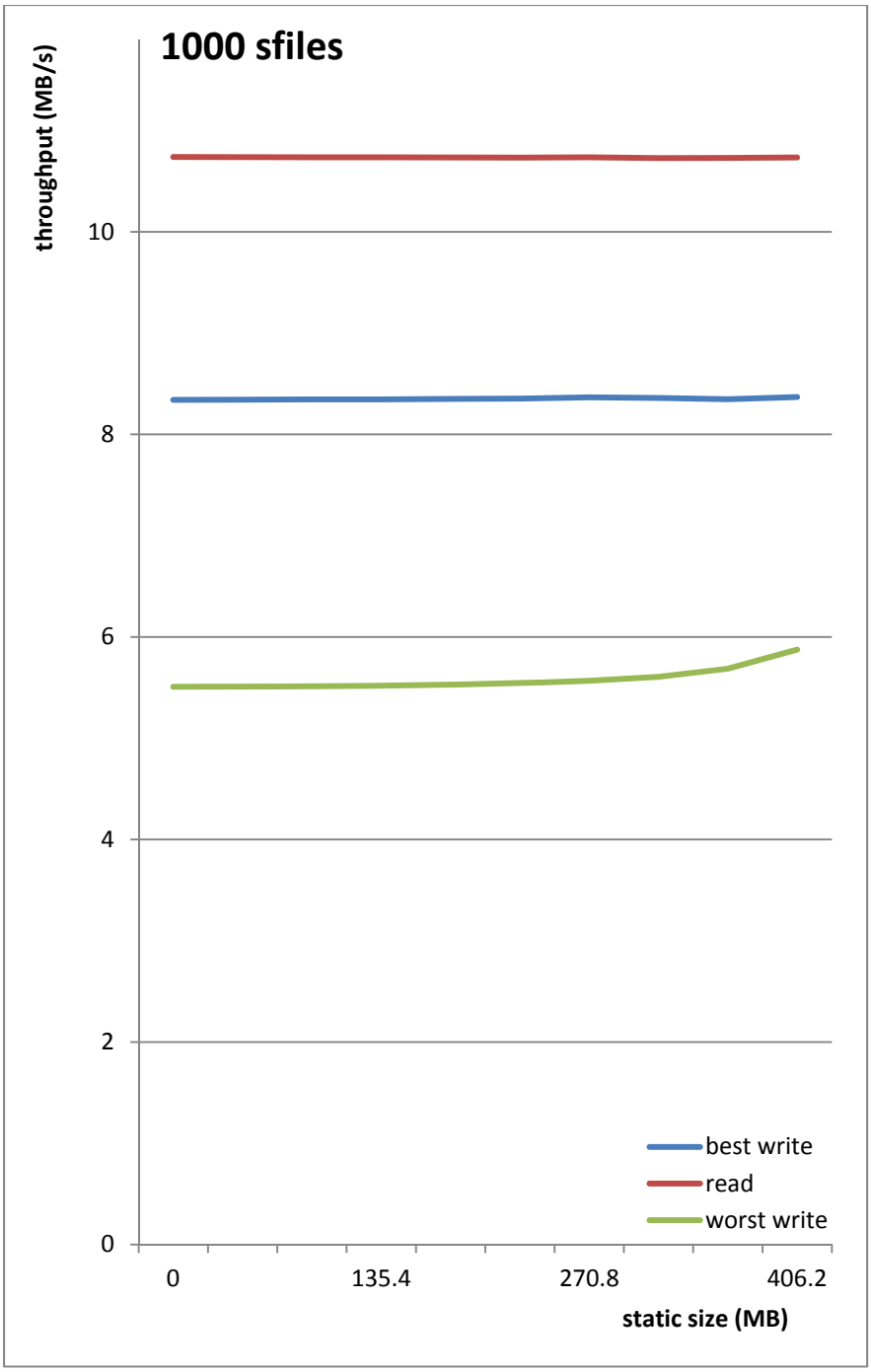
- The amount of initial data on the volume. For the first measurement, the volume was empty. The amount of initial data was increased for the following measurements. The volume was almost full at the beginning of the last write measurements.
- The number of other files present on the volume. For the first graph, the number of 'static' files was 100, for the next it was 1,000, and for the third graph there were 10,000 'static' files at the start of the measurement. This varied the amount of volume metadata.

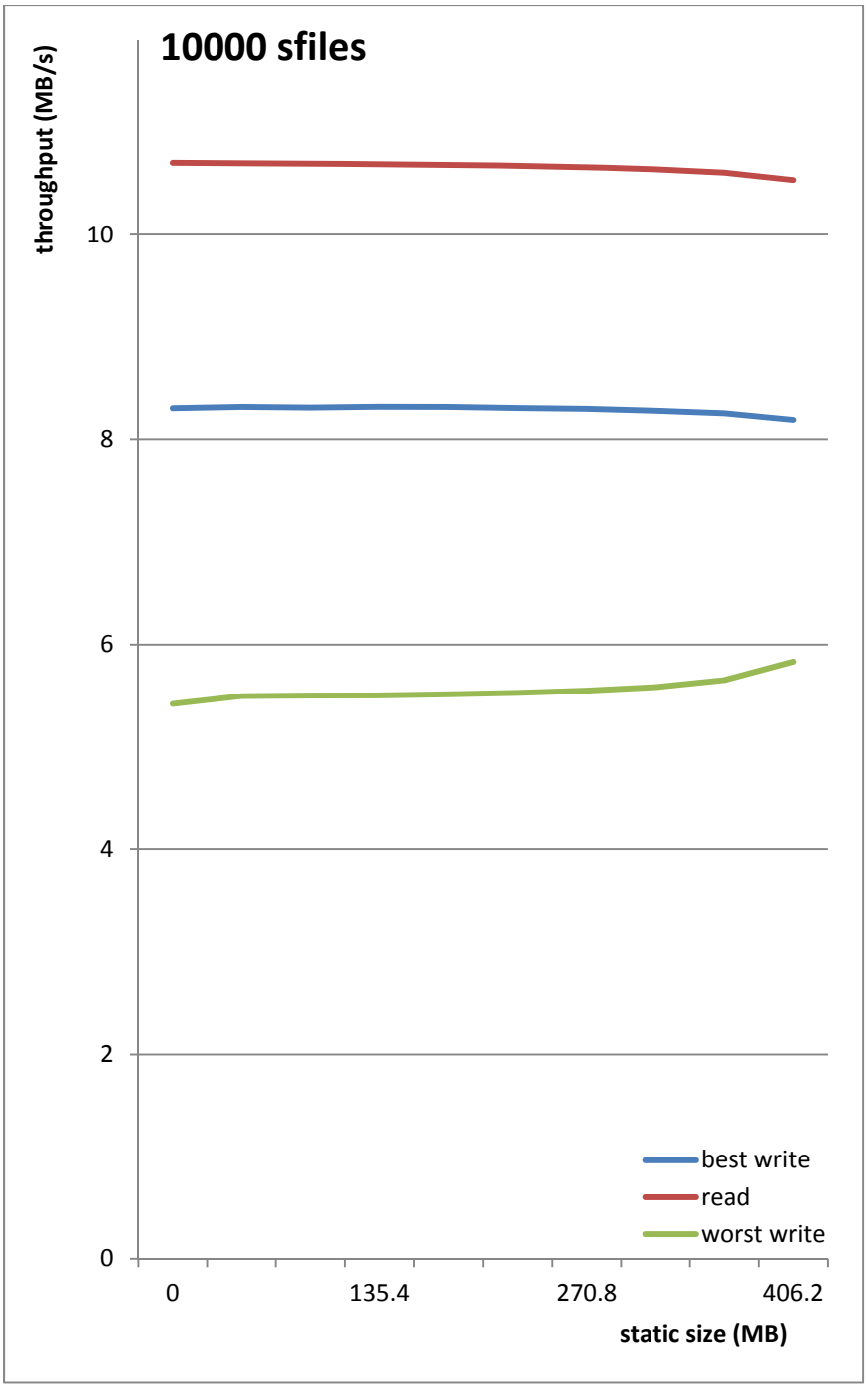
The benchmark performed the following loop, increasing the amount of initial data each pass:

- Add, format and mount the test volume
- Create the specified number of static files
- Write the specified amount of initial data
- Measure best-case write speed by filling volume with single file
- Measure read speed by reading the file back
- Delete the file (leaving volume filled with dirty sectors)
- Measure worst-case write speed by writing the file again
- Unmount and unformat volume

The results are shown in the following three graphs.


100 sfiles
throughput (MB/s)
10
8
6
4
2
0
0
135.4
270.8
406.2
static size (MB)
best write
read
worst write


1000 sfiles
throughput (MB/s)
10
8
6
4
2
0
0
135.4
270.8
406.2
static size (MB)
best write
read
worst write

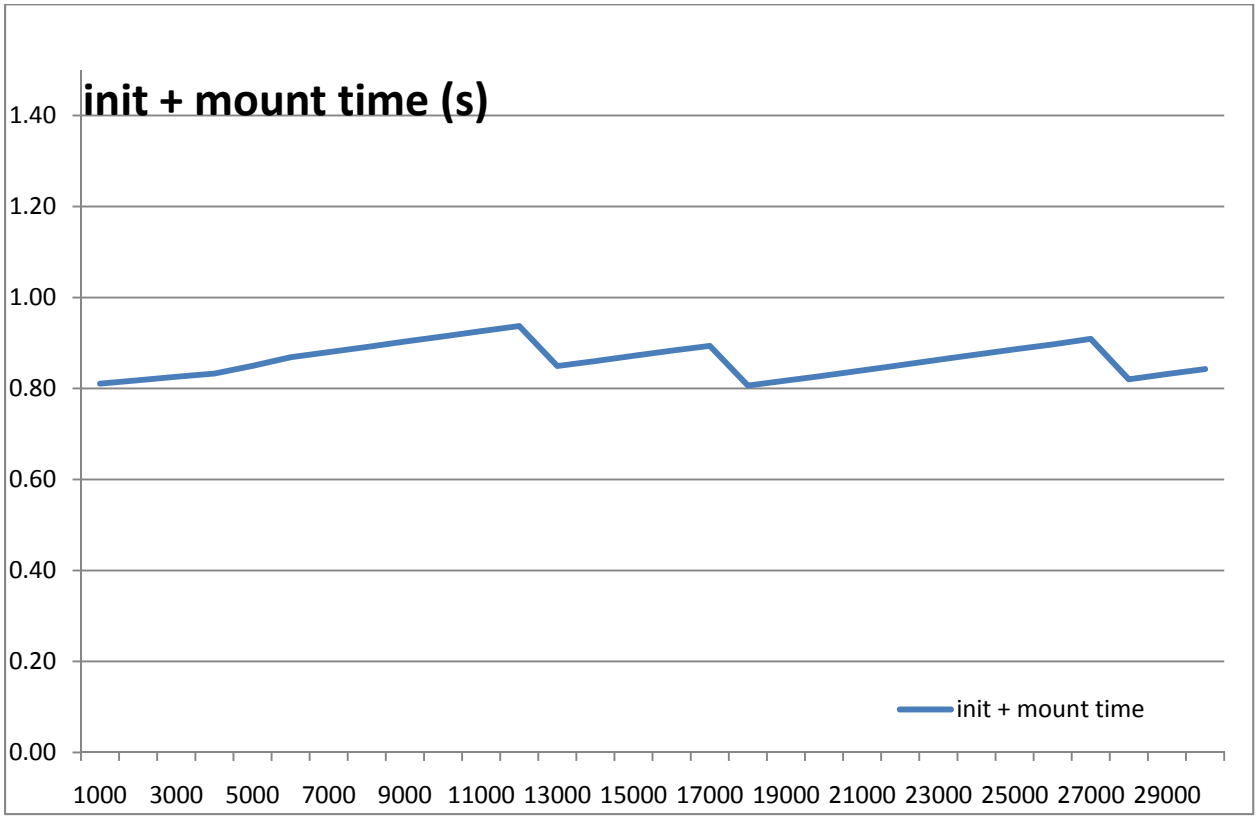


Flash file systems usually experience a falloff in write performance as the volume becomes full. These measurements show that liteFS-NAND does not suffer a significant decrease in write throughput as the amount of free space decreases. Also, they show that the amount of file system metadata has minimal impact on the throughput.

2. Mount/Initialization Benchmarks

During the throughput benchmark, the time to add and mount each volume was measured as well. The mount measurement was made at the end of the throughput measurement, when the volume was full. This is the worst case, since it maximizes the amount of volume metadata and mounting consists of finding and loading the most recent file system metadata.

The results are shown in the following graph. The time required to add the volume and mount it were added and shown in one plot, since those operations are normally done together.



Initialization and mount time versus number of files

3. Utilization Benchmark

Utilization is the percentage of user data held by the volume versus the total amount of flash memory assigned to it. Because file system metadata is also stored on the volume, a volume can never be filled completely with user data. For example, a volume made from 32MB of flash memory won't be able to hold 32M of user data. It can only hold something less than that.

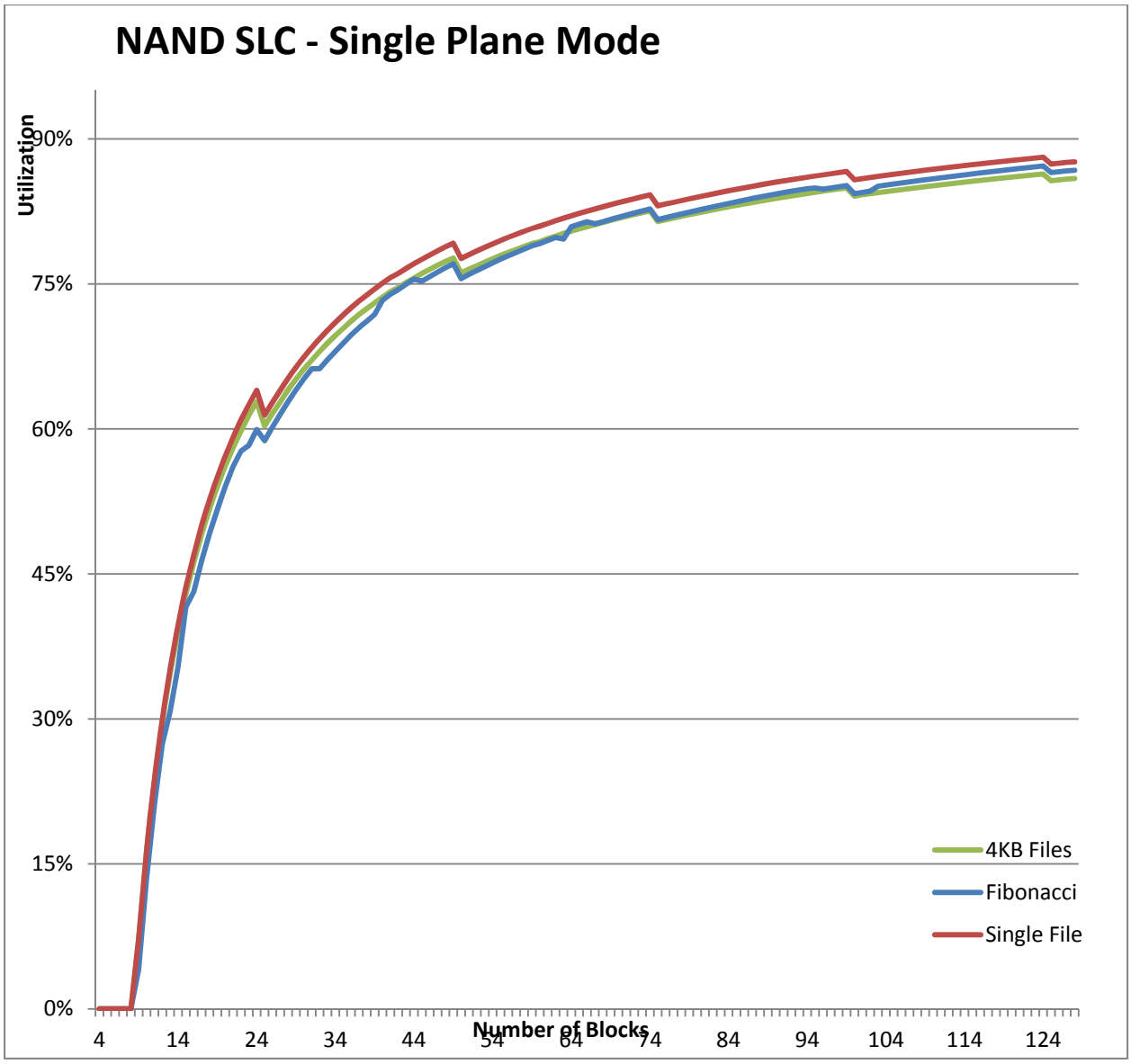
Utilization is affected by the file system type, the number of erasable blocks assigned to the volume, and the amount of volume metadata. Increasing the number of files increases the metadata, in order to hold the additional file names, modes, links, etc. and therefore decreases utilization.

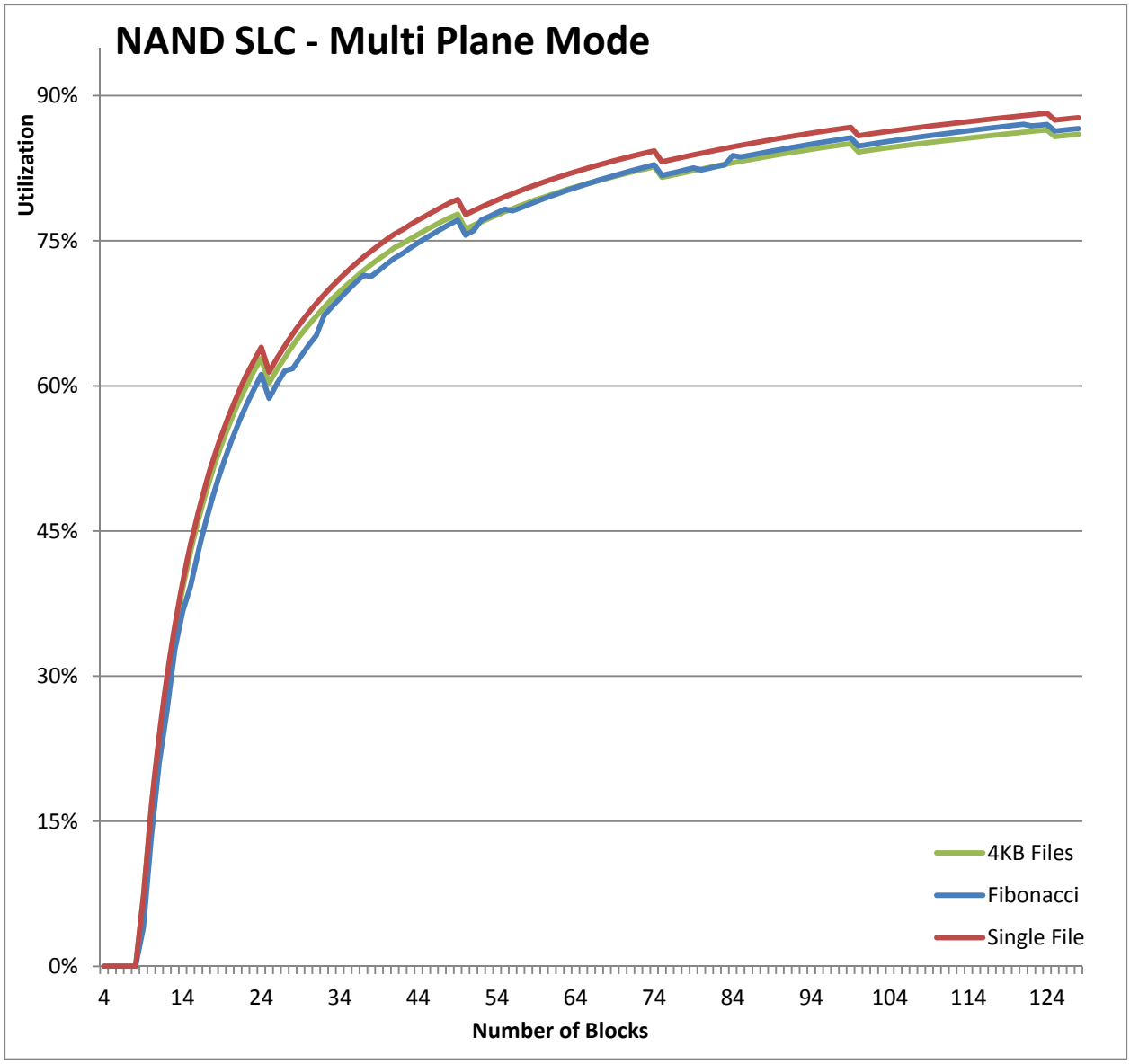
Flash file systems become more efficient as the number of erasable blocks is increased. Using the minimum number of erasable blocks gives very low utilization. liteFS-NAND volumes need to have at least seven blocks assigned, and for good utilization should have at least thirty-two.

This benchmark measured liteFS-NAND utilization versus the number of assigned blocks. Separate graphs were plotted for the following three methods of filling the volume:

4KB files: multiple 4KB files were created until the volume was filled.
Fibonacci: multiple files were created until the volume was filled, getting the size of each new file from the Fibonacci sequence (1, 1, 2, 3, 5, 8, ...), but restarting at 1 after reaching one tenth of the total flash size.
Single file: a single file was written to until it filled the volume.

The results are shown in the following two graphs, the first for single plane mode (128K blocks) and the second for multiplane mode (256K blocks):





The utilization is similar for both modes. However, since single plane mode uses a smaller page (2KB) than multiplane mode, single plane volumes will have better utilization if there are very many files less than 2KB in size.