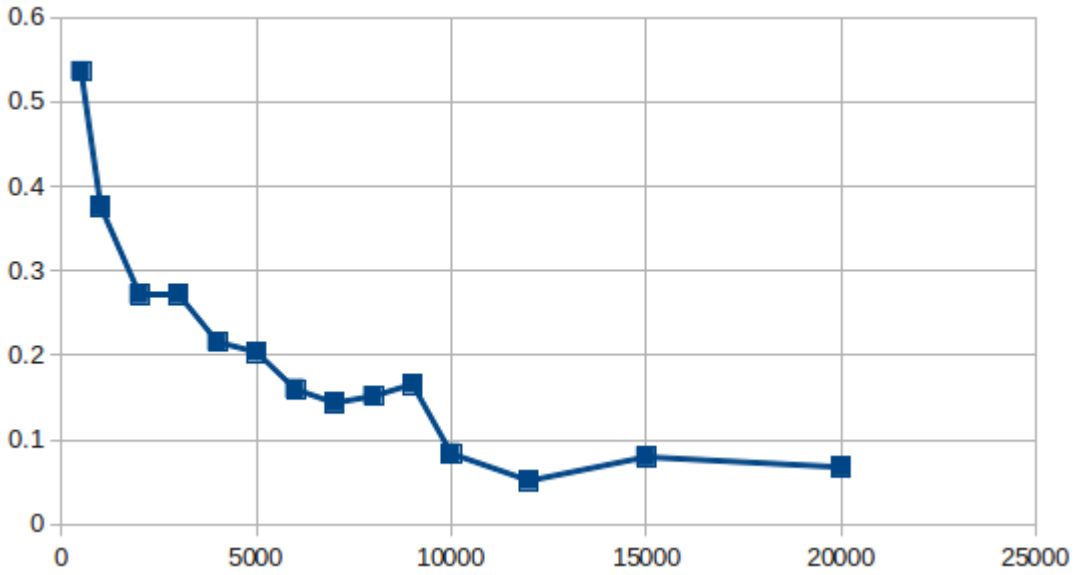
Performance Analysis

Problem 1 (copy)



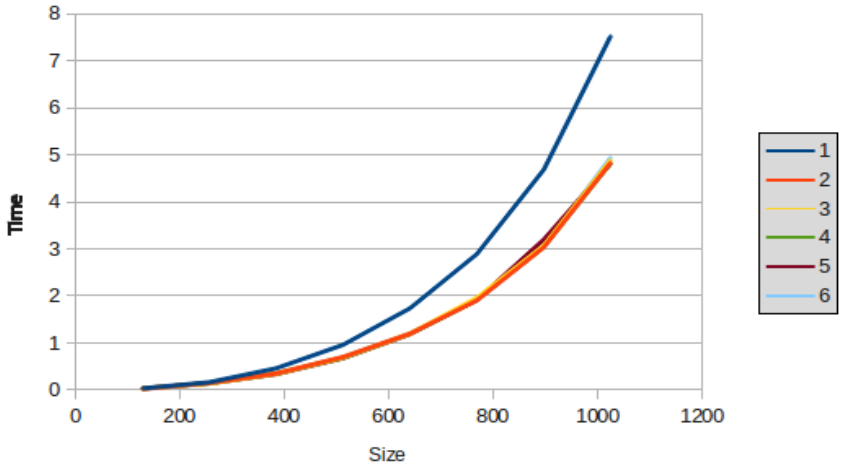
X axis: buffer size in bytes
Y axis: copy time in seconds

[Analysis]

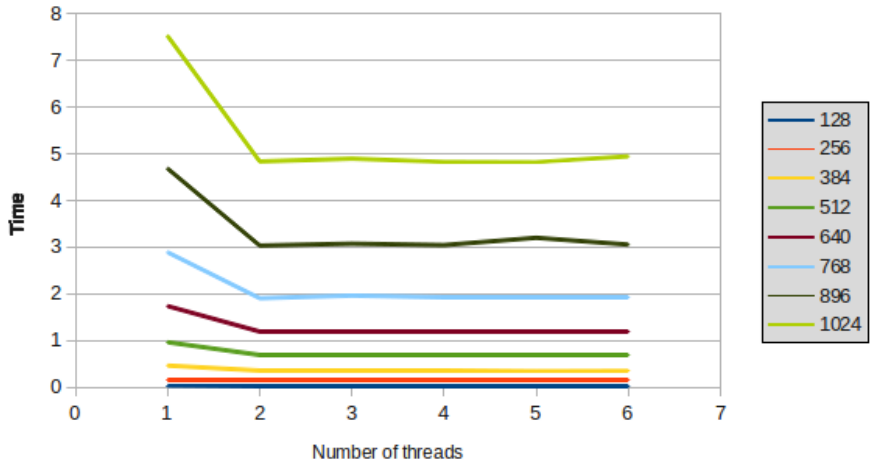
A single block of data on the hard disk can be fetched efficiently. The size of a block is 8192 bytes.

1. When the size of the buffer is no greater than the size of a block, the performance improves dramatically as the buffer size increases, because a single disk fetch doesn't fill a block and fetched ranges interleave block boundaries. Hence, there will be redundant and inefficiency.
2. When the size of the buffer is greater than the size of a block, the performances remains the same when the buffer size increases, because the utilization of a block fetching already reaches the limit.

Problem 3 (multi)



X axis: matrix size
Y axis: calculation time in seconds
Series: different number of threads



X axis: number of threads
Y axis: calculation time in seconds
Series: different matrix sizes

[Analysis]

1. We use a cubic polynomial time algorithm for matrix multiplication, so when the size of the matrix increases, the time grows polynomial cubically no matter how much threads are used.
2. The tests are run on a 2-core CPU, so when the number of threads increases from 1 to 2, the time used shrinks by roughly 2/5. But when the number of threads increases beyond 2, there will be no difference on the running time because there are only 2 cores.