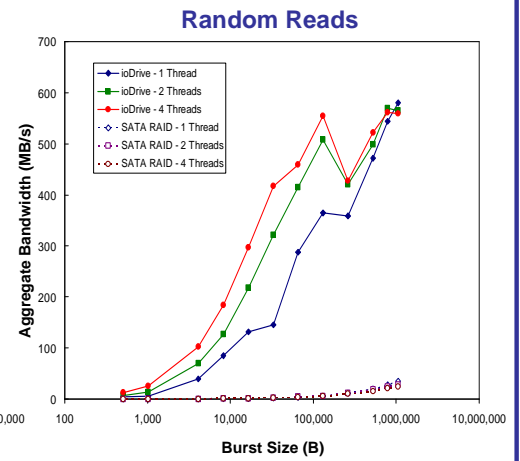
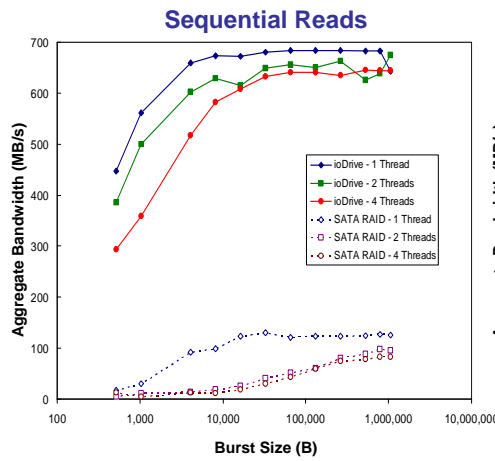


Block Transfer

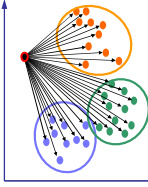
- Task**
 - Access different blocks of data on disk
 - Sequential and Random, Reads and Writes
- Caveats**
 - Access patterns picked to minimize caching
- Approach**
 - Multiple threads issue requests as fast as possible
 - Bandwidth reported as an aggregate
- Observations**
 - Sequential reads work best with one thread for both devices
 - Threading improves ioDrive's random reads
 - ioDrive's dip at 256KB is due to its internal block size
- Speed comparison (Peak Performance)**

Test	SATA RAID	ioDrive	Speedup
SeqRead	125 MB/s	683 MB/s	5x
SeqWrite	139 MB/s	661 MB/s	4x
RandRead	34 MB/s	580 MB/s	17x
RandWrite	46 MB/s	658 MB/s	14x

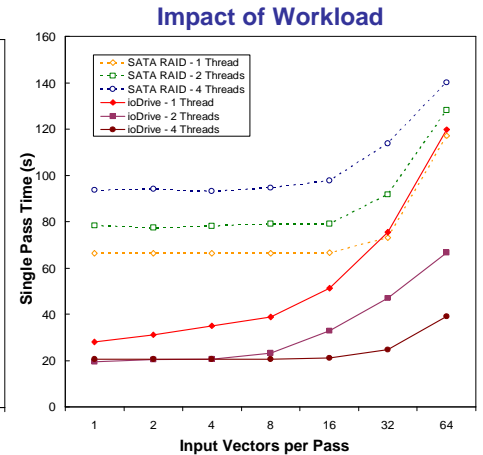
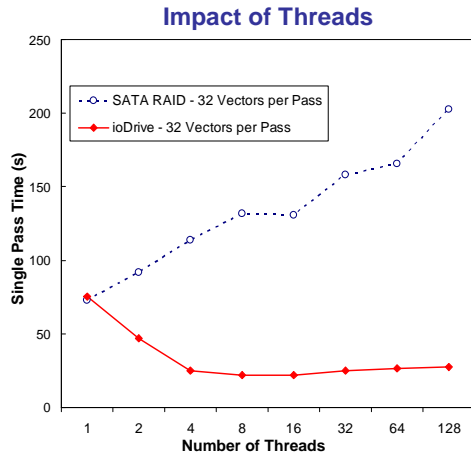


k-Nearest Neighbors

- Task**
 - Find k training vectors in a large file that are the most similar to an input vector
- Caveats**
 - Must read all training vectors
- Approach**
 - Use multiple threads that operate on different portions of training vector file
 - Combine threads' results after all complete
 - Process multiple input vectors at a time
- Observations**
 - Purely streaming I/O behavior
 - Threading helps ioDrive but hurts SATA RAID
 - Transitions from I/O-bound to Compute-bound workload
- Speed comparison (Peak Performance)**

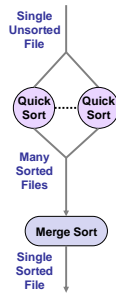


Test	SATA RAID	ioDrive	Speedup
16-inputs per pass	66 s	21 s	3x
32-inputs per pass	73 s	22 s	3x

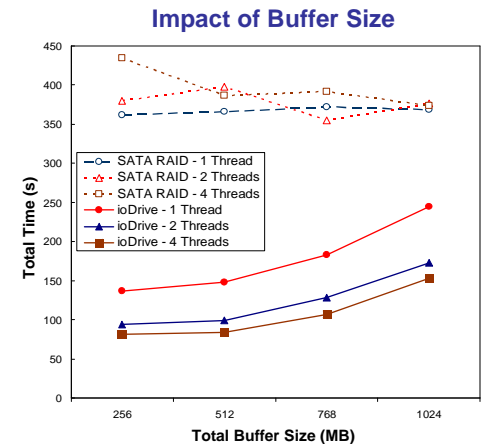
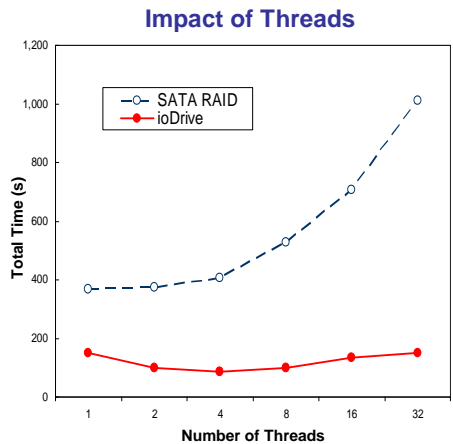


External Sort

- Task**
 - Sort a large file of vectors
- Caveats**
 - File is larger than main memory
 - Fixed amount of buffer space available
- Approach**
 - Many threads quick sort different regions
 - Intermediate results written back to disk
 - Single thread merges all files together
- Observations**
 - Threading degraded SATA performance
 - ioDrive had best performance with
 - Small number of threads (four)
 - Small buffer size (256MB)
- Speed comparison (Peak Performance)**

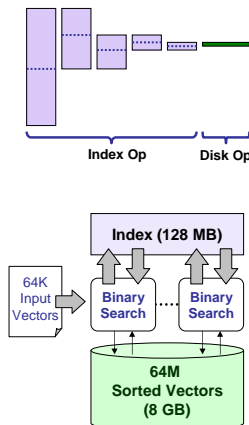


Test	SATA RAID	ioDrive	Speedup
Total Time	361 s	81 s	4x

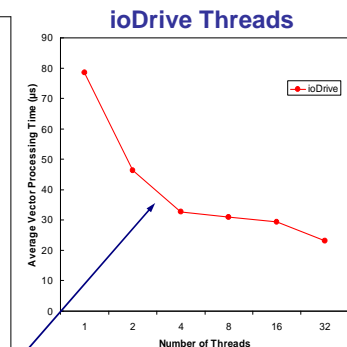
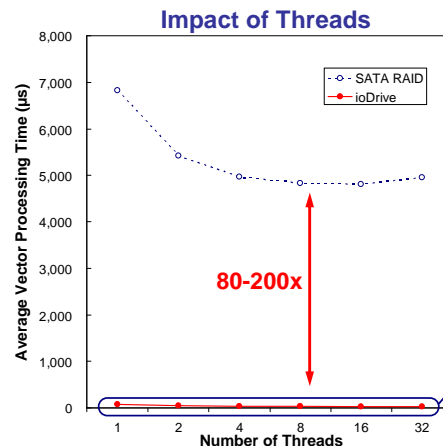


Binary Search

- Task**
 - Locate input vectors in a large, sorted file
- Caveats**
 - Sorted file is larger than main memory
 - Requires log(n) vector comparisons
- Approach**
 - Index the sorted file at start time
 - Use index to minimize disk reads
 - Use threads to process multiple inputs at a time
- Observations**
 - Purely random I/O behavior
 - Threading helps both SATA RAID and ioDrive
 - Performance depends on disk access time
- Speed comparison (Peak Performance)**



Test	SATA RAID	ioDrive	Speedup
Create 128 MB index	387 s	38 s	10x
Process 64k inputs	315 s	1.5 s	> 200x



A small number of threads improves performance.