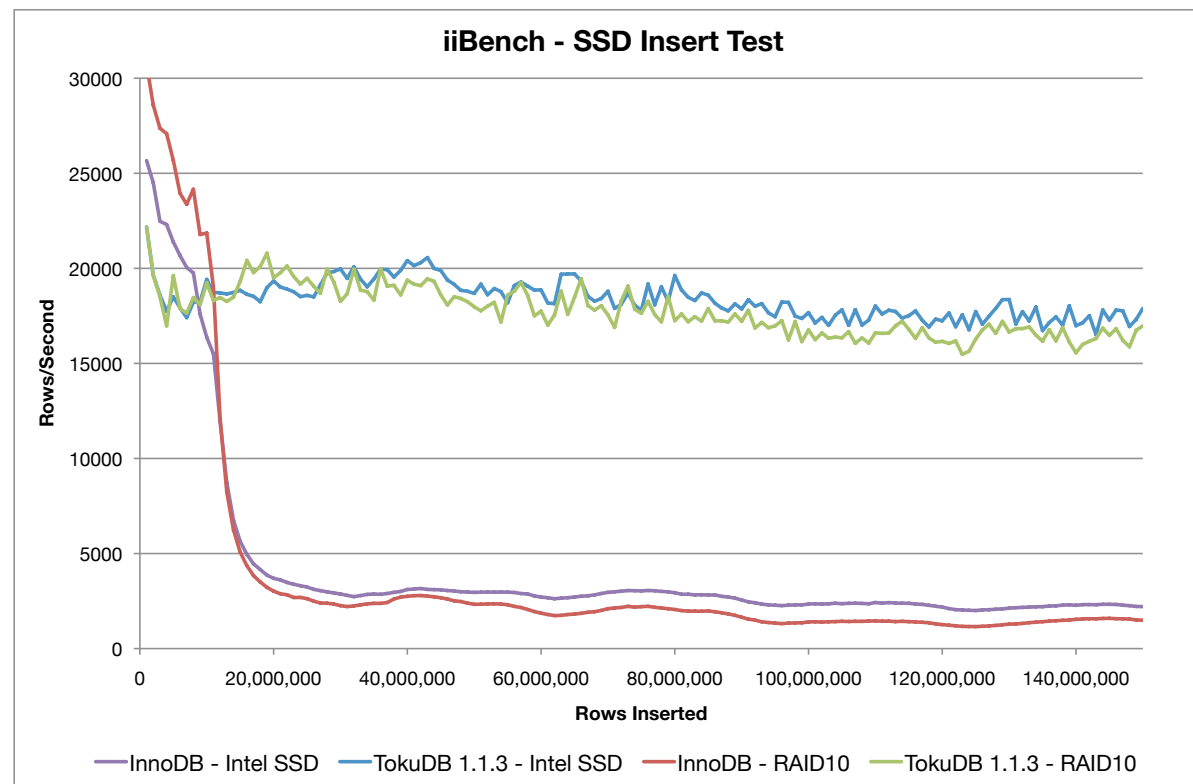


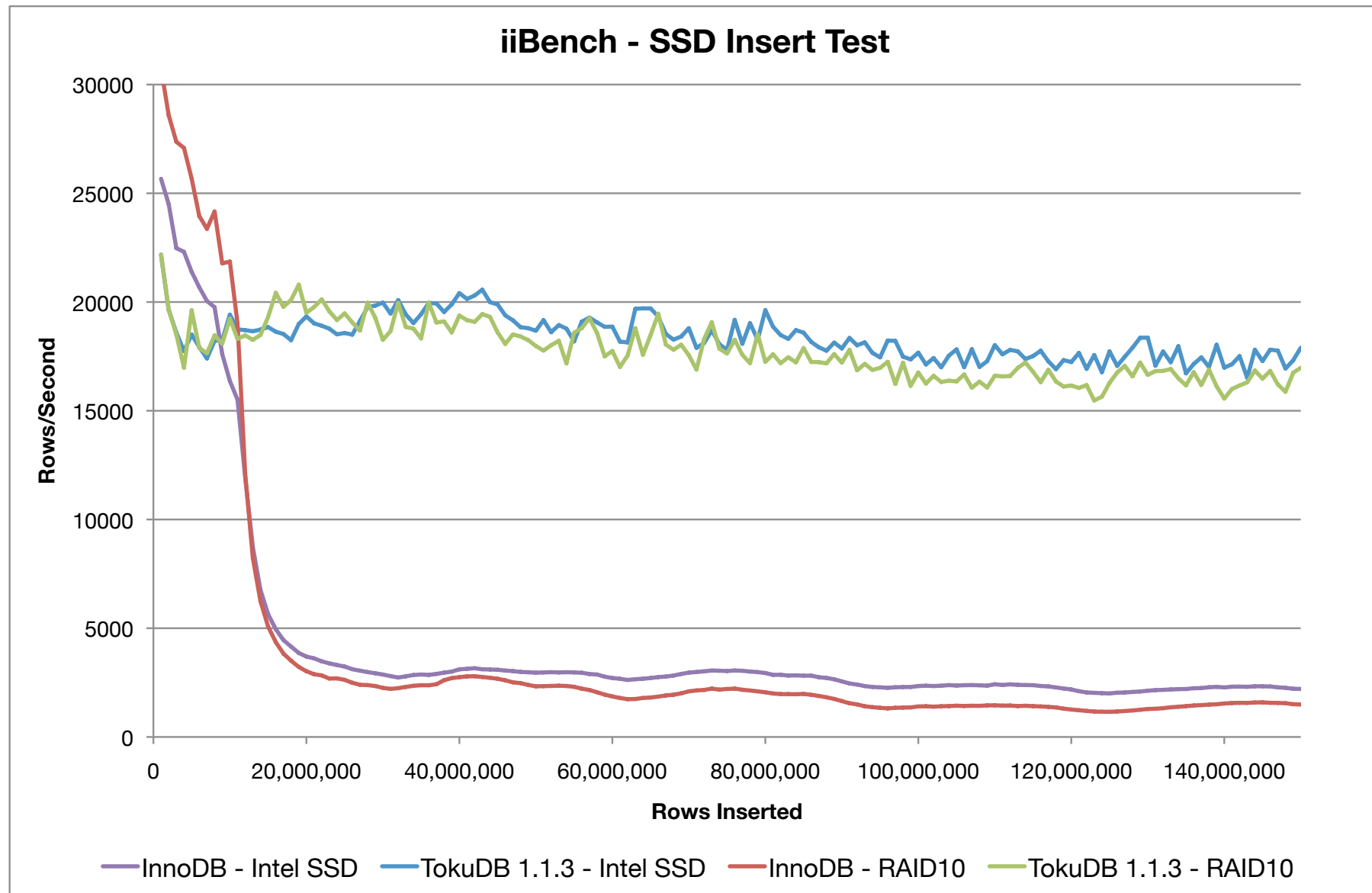
What Is a Performance Model for SSDs?

Bradley C. Kuszmaul
Tokutek & MIT



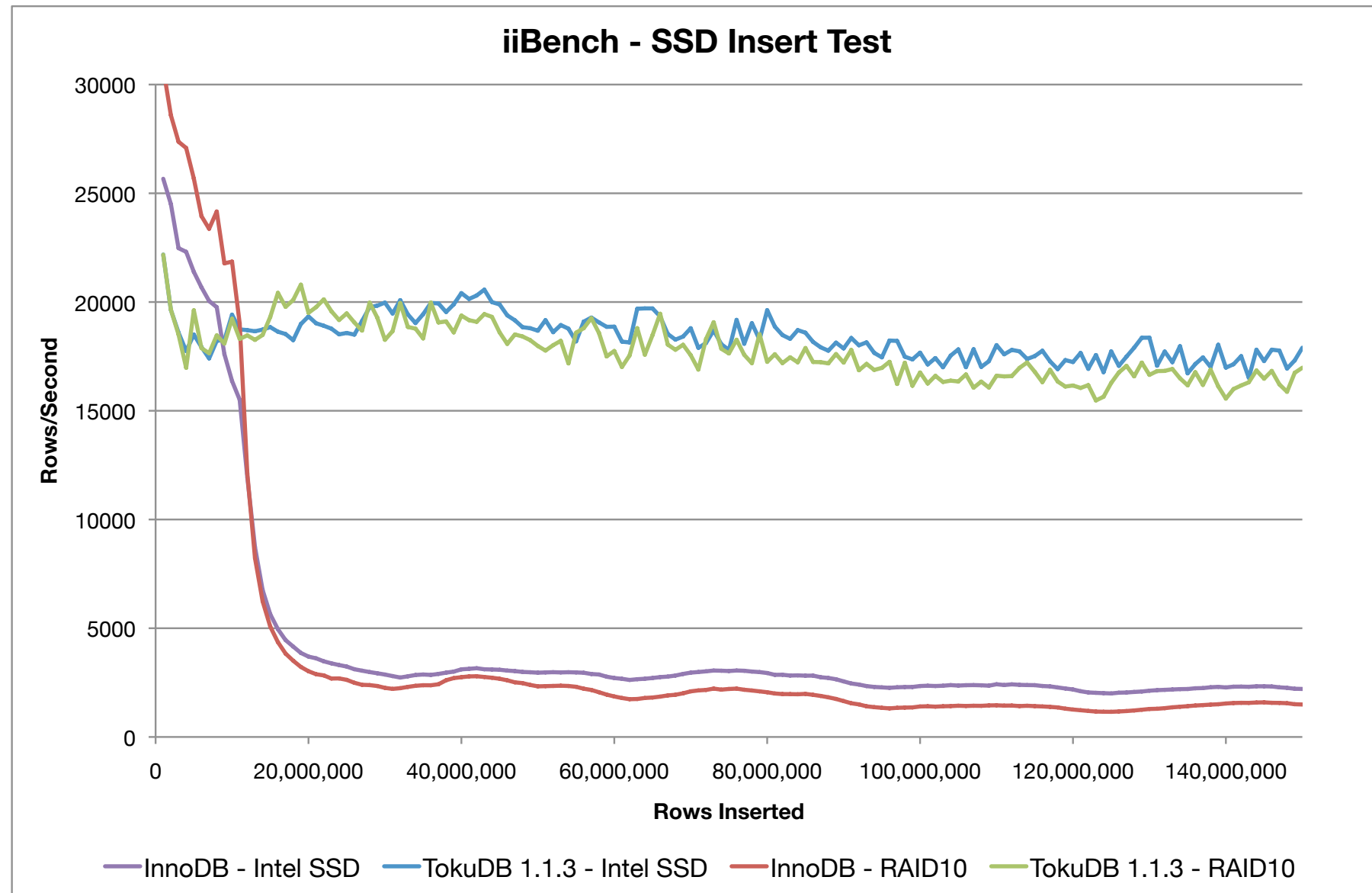
Motivation: Understand SSD performance to design fast data structures.

Poor MySQL B-Tree SSD Performance?



iiBench running on InnoDB and TokuDB, on rotating Disk and on Intel X25E. (Measured by Percona)

Poor MySQL B-Tree SSD Performance?



Surprisingly, disk is almost as good as SSD. Not CPU bound on InnoDB.

Intel X25E Specifications

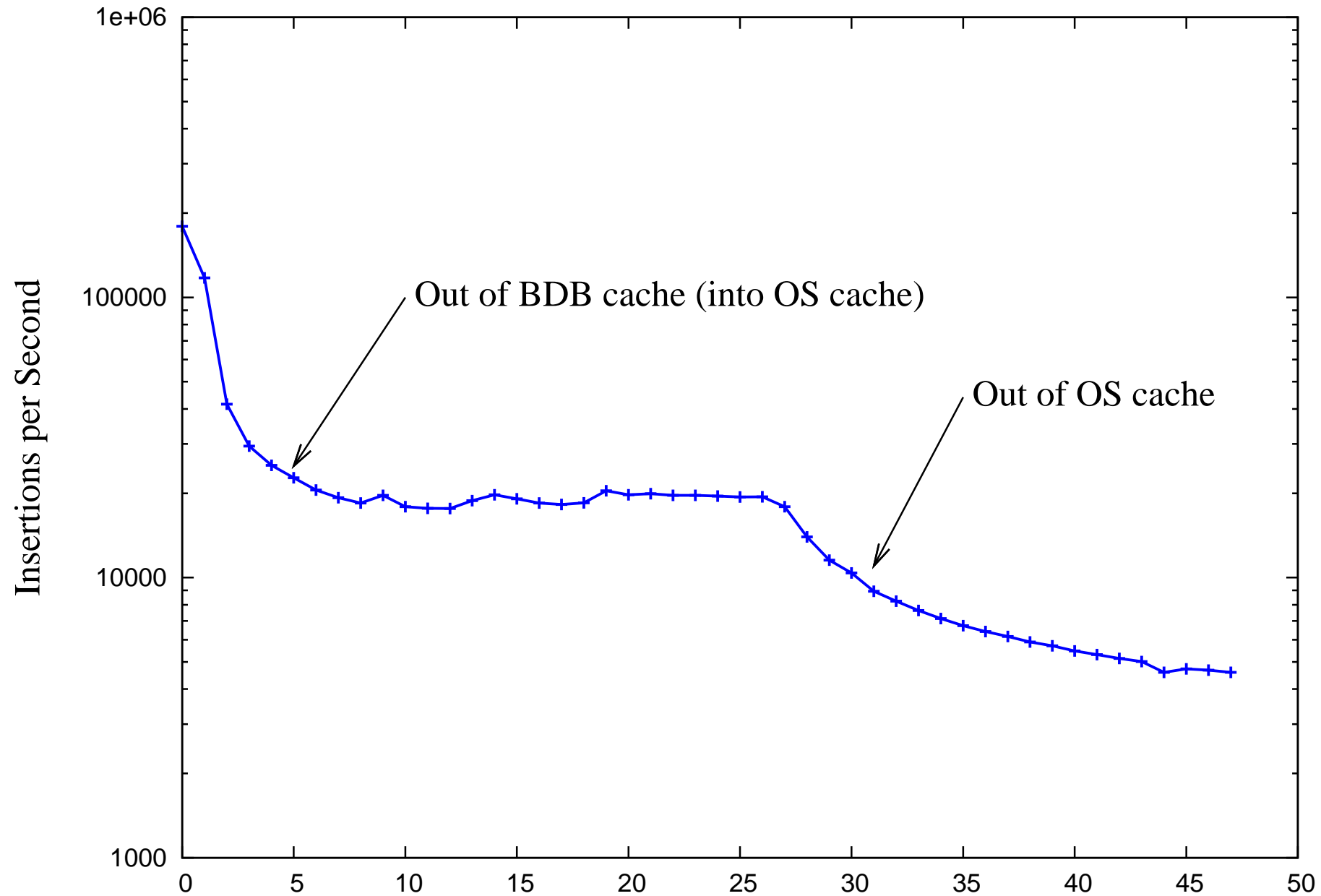
- Read bandwidth up to 250 MB/s.
- Write bandwidth up to 170 MB/s.
- Random 4KB read rate: 35 KIO/s.
- Random 4KB write rate: 3.3 KIO/s.

Disk bandwidth (5 disk RAID):

- Read/Write bandwidth about 400 MB/s.
- Random Read/Write rate: 600/s (with the wind at your back.)

So why isn't the SSD giving InnoDB a 6x performance boost?

MySQL too Complex \Rightarrow Berkeley DB



Trending to 4500 writes per second (still dropping...)

Berkeley DB too complex \Rightarrow Try File I/O

Method:

- Build a 12GB file on a machine with 3GB RAM.
- Perform random reads and writes of various sizes.
- Build a performance model.

A Performance Model

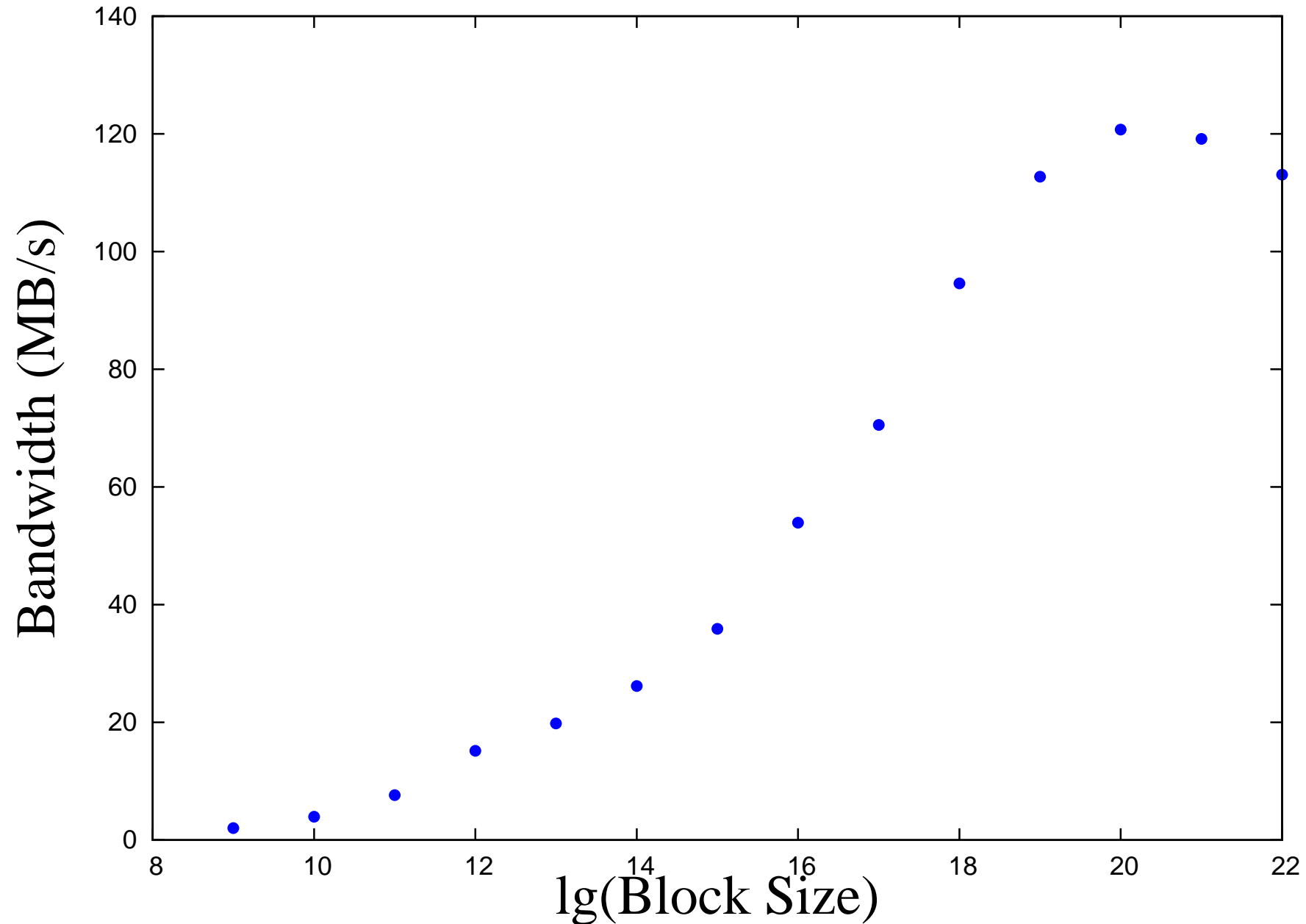
Can I make the following performance model work?
When reading a block of size B ,

- There is a startup cost, S , (“seek time”)
- There is bandwidth, W , (“transfer rate”).

The simple model is thus

$$T_R = S_R + B/W_R$$
$$T_W = S_W + B/W_W$$

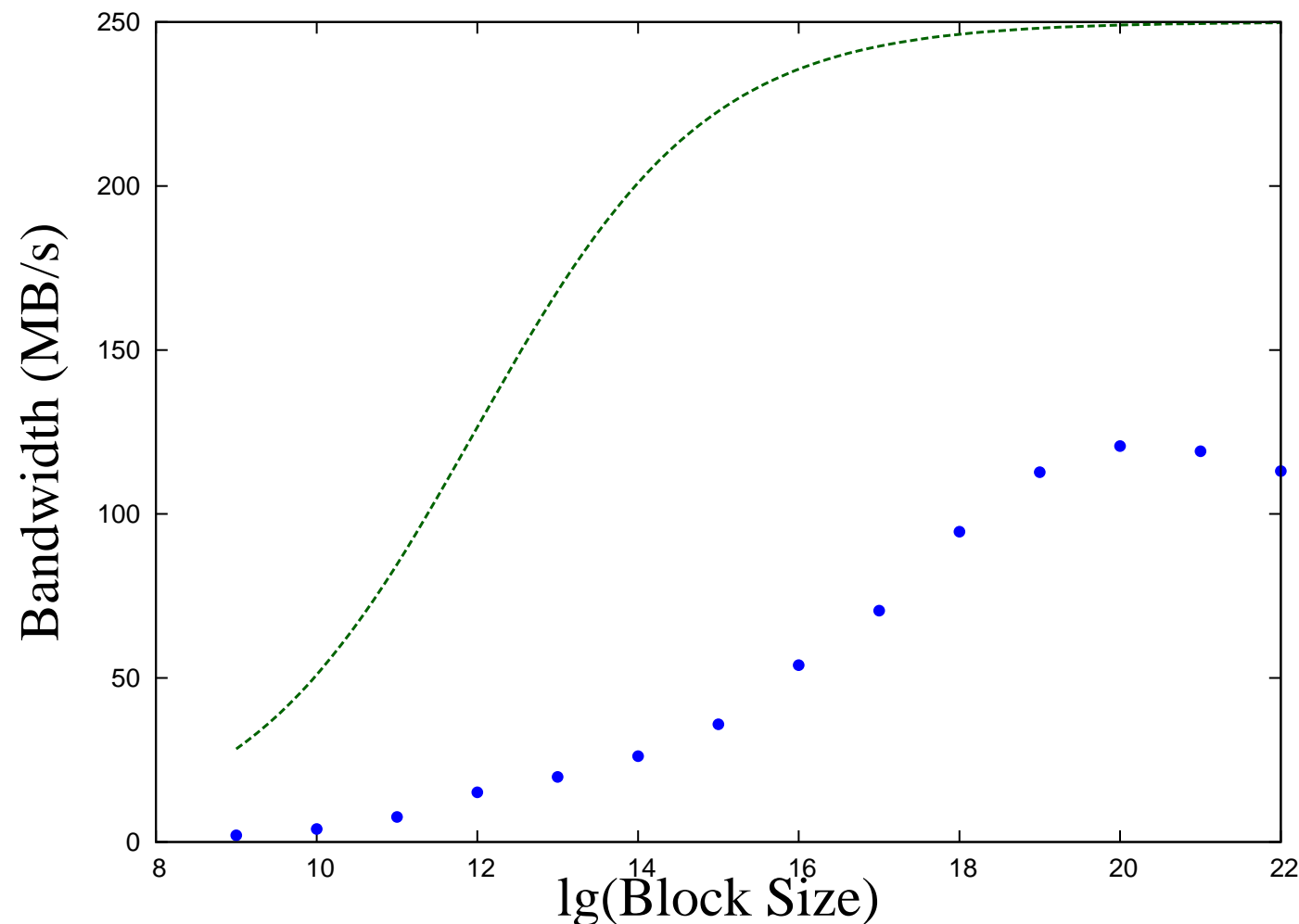
Read Performance as a Function of Block Size



A Model from the Data Sheet?

The Manufacturer's 35 KIO/s and 250 MB/s suggests this (poor) model:

$$T_R = 16\mu s + B / (250 \text{ MB/s}).$$

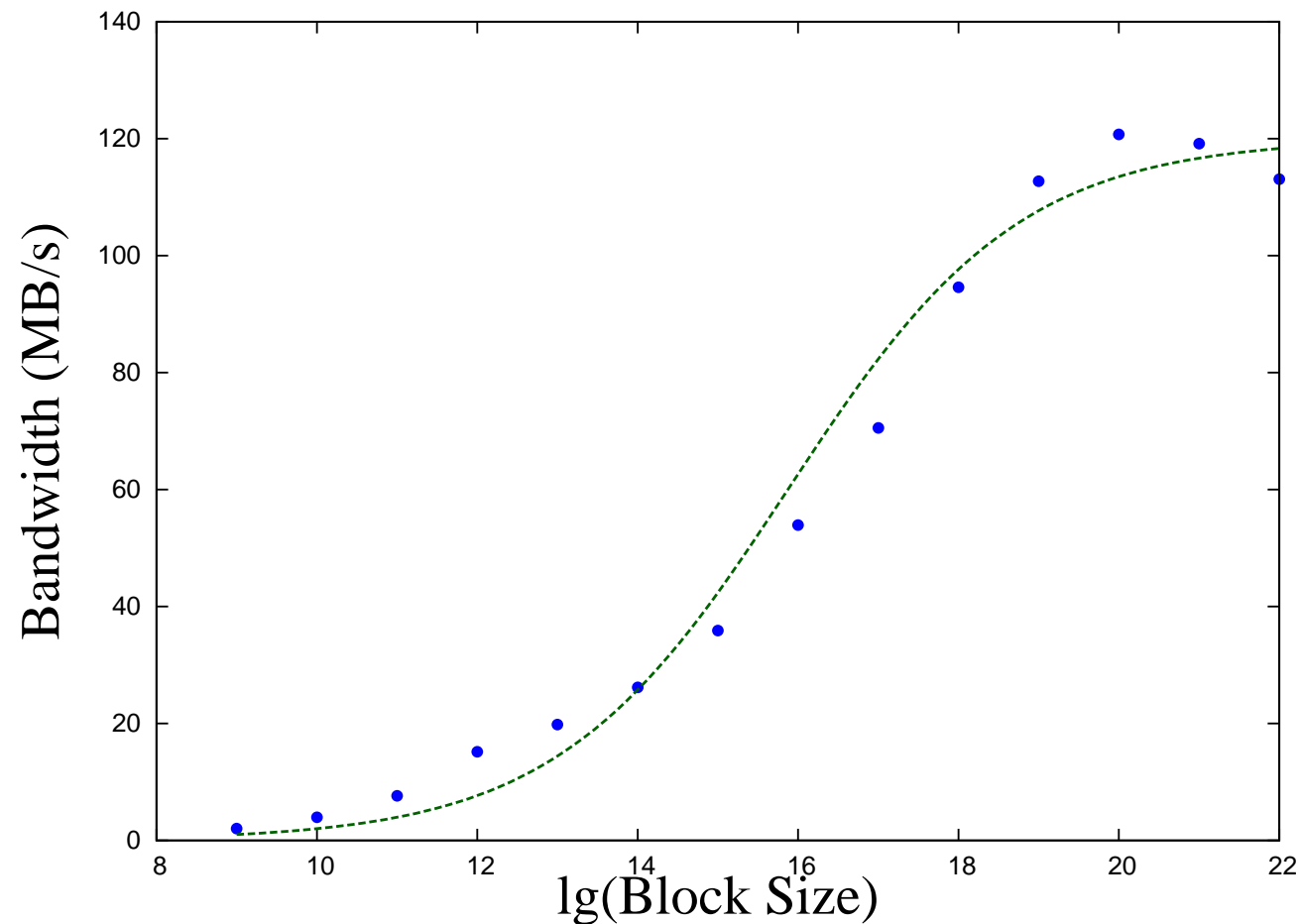


Actual Read Performance

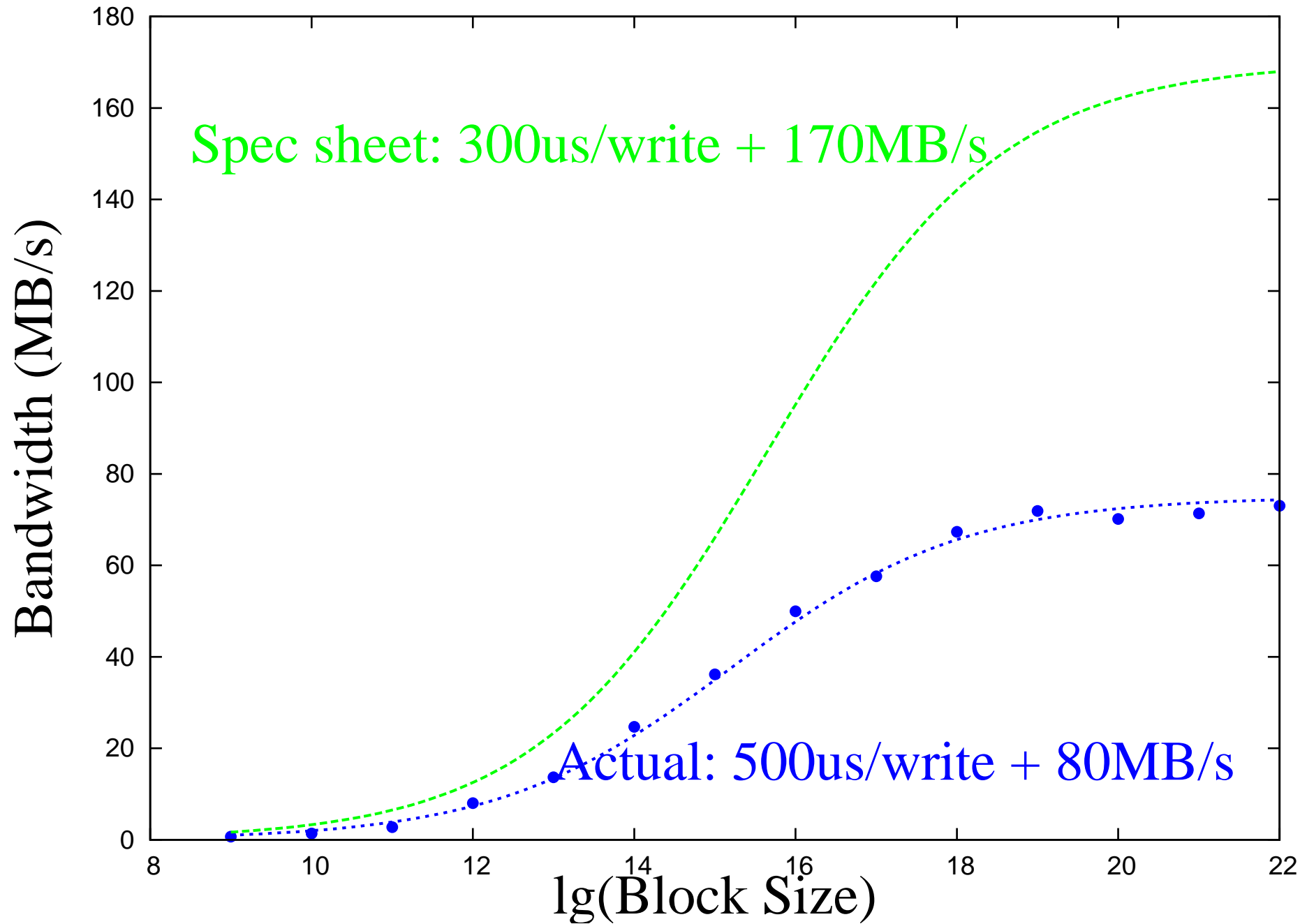
Actual read performance is 4K IO/s, not 35K IO/s.

$$T_R = 16\mu s + B / (250\text{MB/s}) \quad (\text{Datasheet})$$

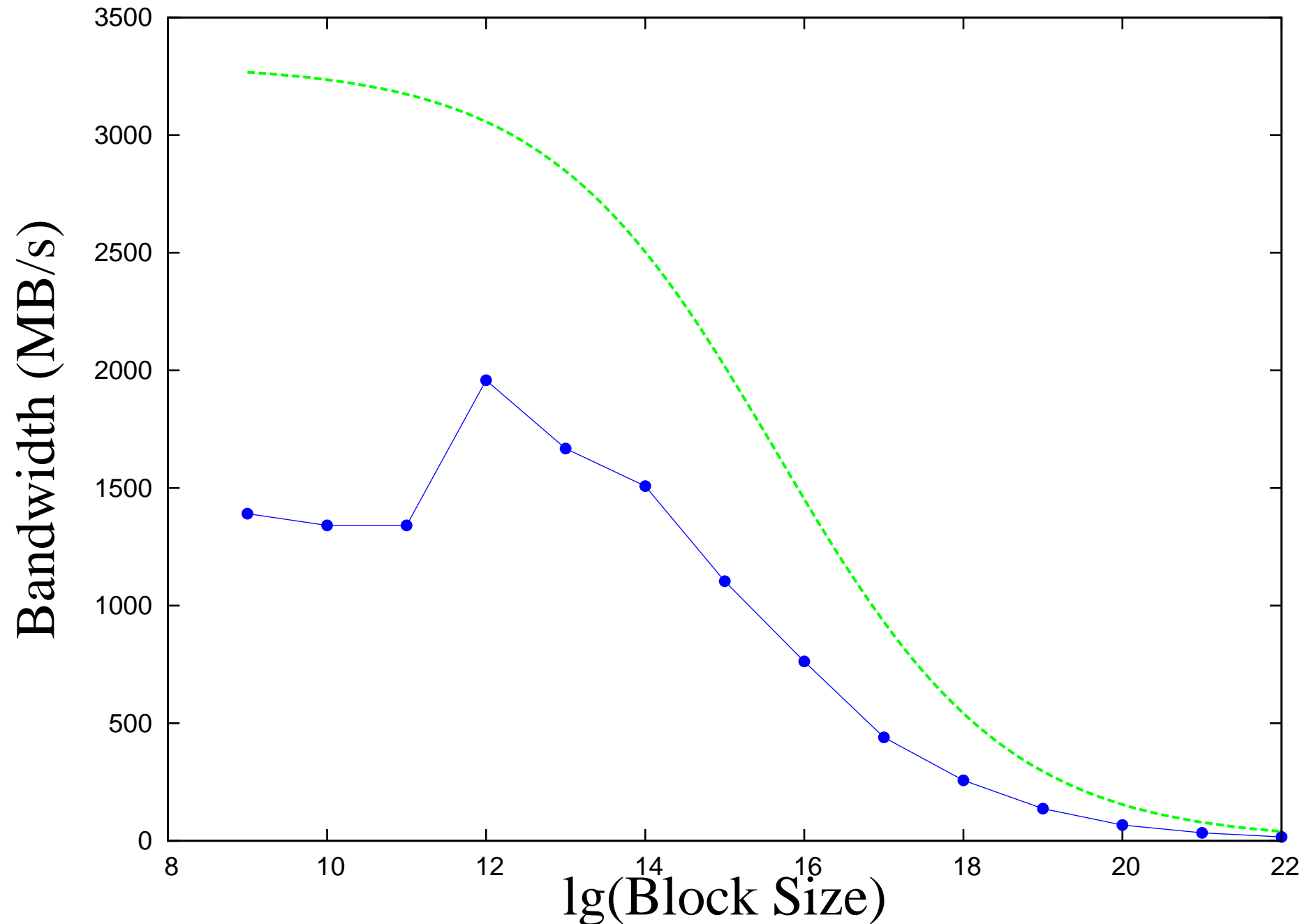
$$T_R = 500\mu s + B / (120\text{MB/s}) \quad (\text{Actual})$$



Write Performance

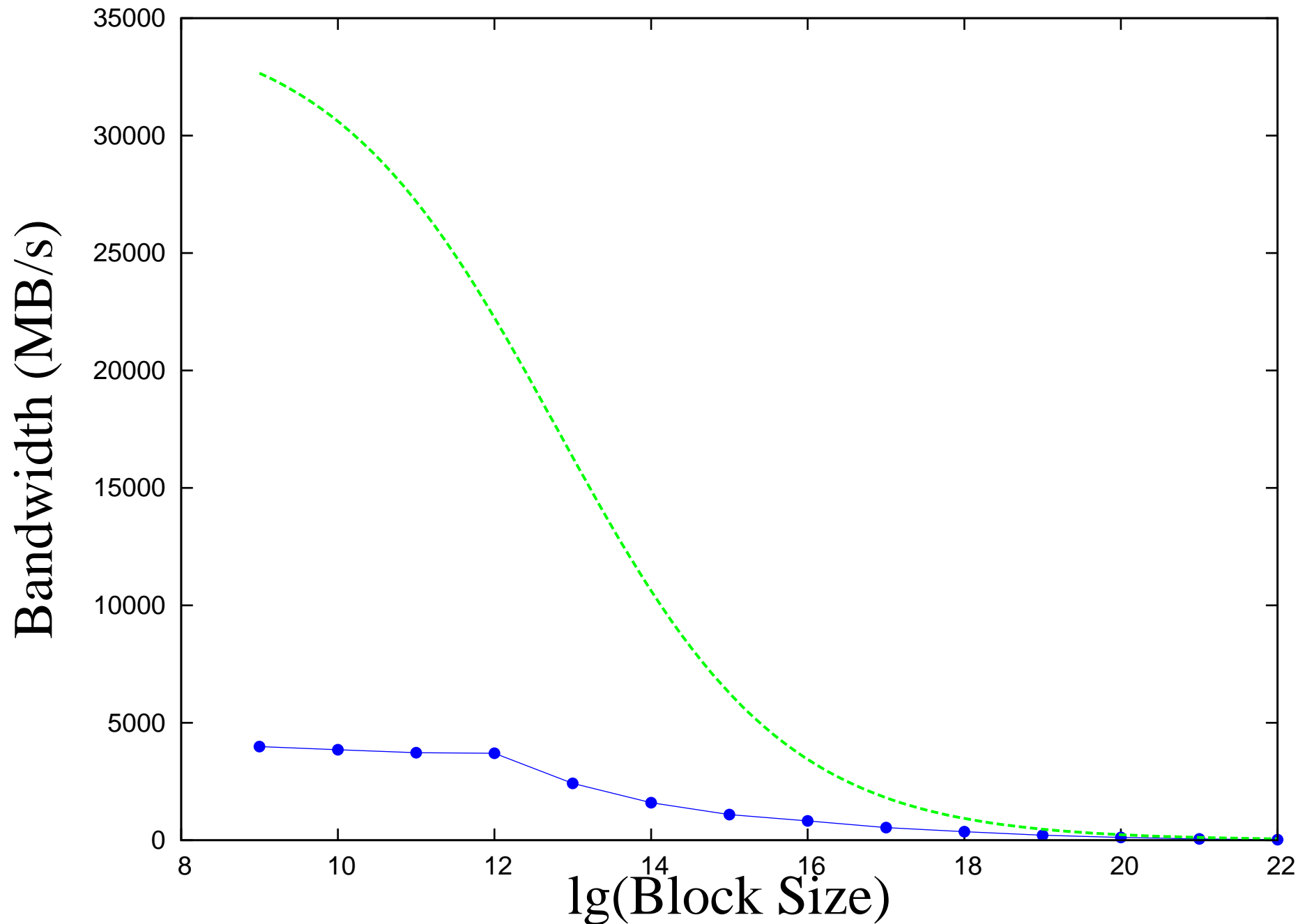


Write IO/s varying block size



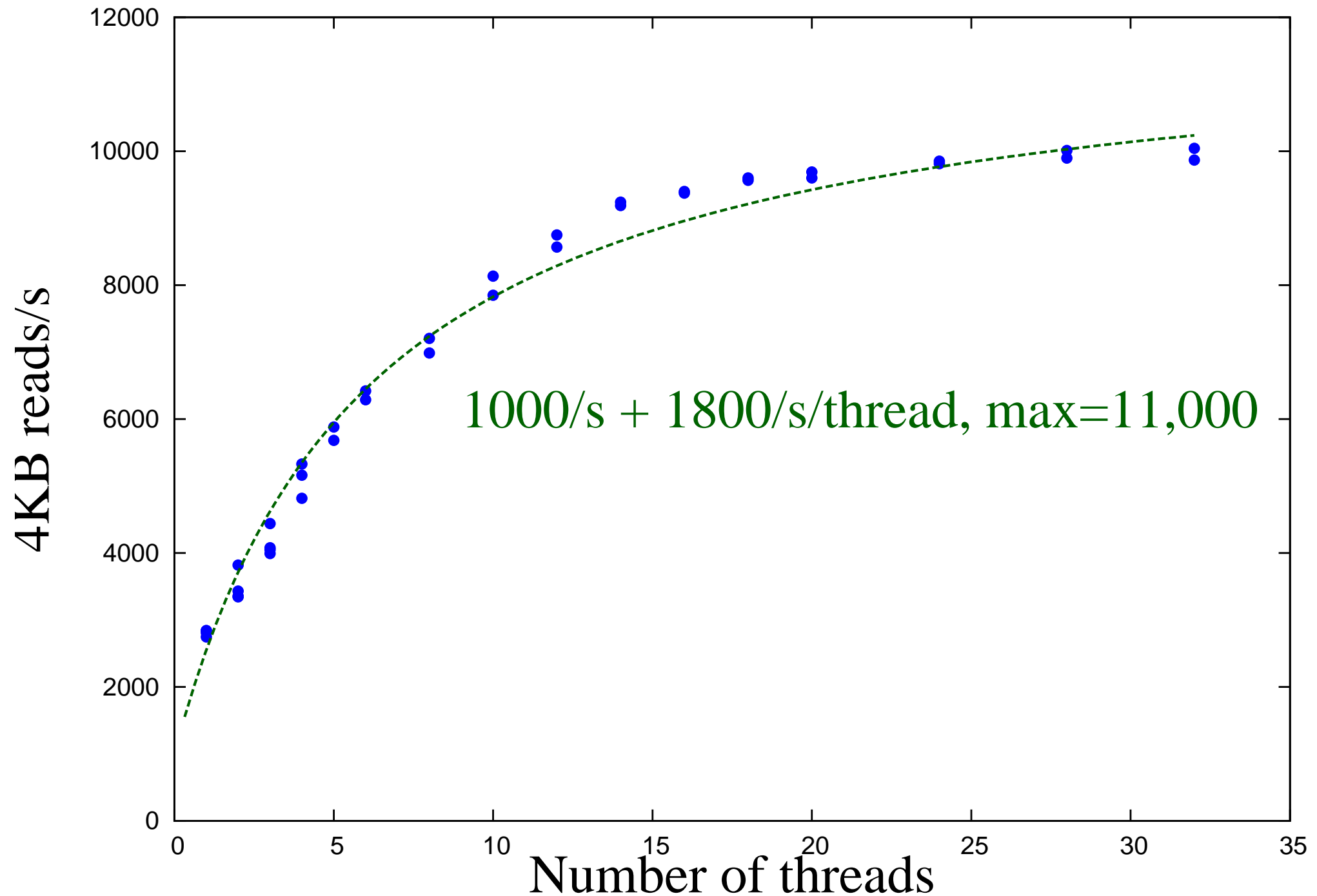
I get about 60% of what the spec sheet promises.

Read IO/s varying block size



I'm an order of magnitude off.

Up to 11,000 reads/s with Multithreading



What's wrong?

- The X25E may be aging. But it behaved about the same when it was fairly young.
- Tried writing to multiple files. Still not good.
- Haven't tried XFS.
- Haven't tried raw device.
- And InnoDB runs much slower than I would expect.

What Block Size To Use?

For point-queries, B-trees are insensitive to block size. As soon as you have any reasonable fanout you do well.

For range queries, the B-tree block size is important.

Tension:

- Large block sizes make range queries faster.
- Large block sizes make point queries slower.

Half-power point

Idea: Set block size so that half the time is accounted for the “seek time”, and half the time by “bandwidth”.

Half-power point

Idea: Set block size so that half the time is accounted for the “seek time”, and half the time by “bandwidth”.

“Half Power Point”

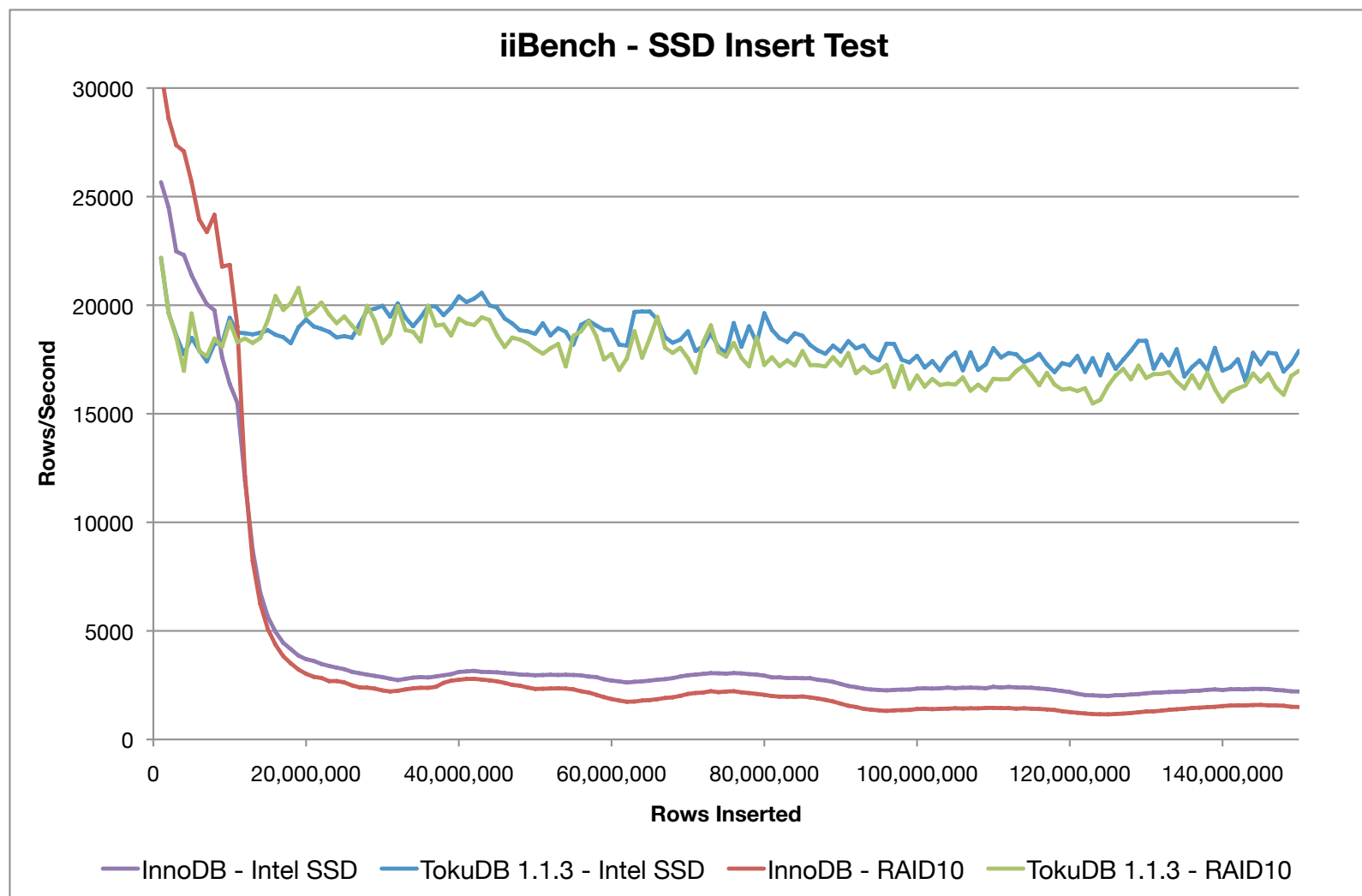
	SSD	Rotating Disk
read unthreaded	60KB	0.5MB–1MB
read threaded	30KB	
write	40KB	0.5MB–1MB

Recommendation: Use 40KB blocks, not 4KB blocks.

Cache-Oblivious Approach

- Use data structures that are fast for any block size.
- The key is to use asymptotically optimal data structures that are unaware of the block size.
- Hence the name “Cache Oblivious Data Structures”.

Fractal Trees are Cache Oblivious



Tokutek's MySQL storage engine uses these ideas to offer 10x or more speedups on insertions into a big database.