Microbenchmark-based Extraction of Local and Global Disk Characteristics

Nisha Talagala, Remzi H. Arpaci-Dusseau, and David Patterson

Computer Science Division
University of California, Berkeley

Abstract

Obtaining timely and accurate information about the low-level characteristics of disk drives presents a problem for system design and implementation alike. This paper presents a collection of three disk microbenchmarks which combine to empirically extract a relevant subset of disk geometry and performance parameters in an efficient and accurate manner, without requiring a priori *information of the drive being measured. Novel among the benchmarks is the utilization of* linearly-increased stride *to glean a spectrum of low-level details including head-switch and cylinder-switch times while factoring out rotational effects. A bandwidth benchmark extracts the zone profile of disks, revealing that the previously preferred linear model of zone bandwidth is less accurate than a quadratic model. A seek profile is also generated, completing the trio of benchmarks. Data is collected from a broad class of modern disks, including five SCSI, two IDE, and two simulated drives.*

1 Introduction

`Theories come and go, but fundamental data always remains.`
`Mary Leakey`

Sustained innovation in the hard-drive industry has spurred incredible advances in disk technology. Both performance and capacity have benefited – bandwidth is increasing at sixty percent per year, and capacity is growing at nearly the same rate. The disk drive industry moves quickly as well; a new drive appears on the market every nine to twelve months.

Due to this rapid evolution, clients of modern disks are left in a quandary: how can one obtain accurate, detailed information about the inner-workings of recently manufactured disks? For system implementors, knowledge of low-level performance characteristics can lead to much improved policy decisions, whereas for system researchers, simulations can be parameterized with the latest disk attributes, facilitating more timely and relevant research. Unfortunately, straight-forward methods for obtaining performance characteristics may not prove successful, as detailed specifications are not always available, complete, or accurate.

One solution put forth in the literature is to employ *microbenchmarks* to characterize hardware and software systems alike. Carefully crafted microbenchmarks have been utilized in a wide range of environments: to accurately describe the performance of uniprocessor and multiprocessor memory systems [1, 5, 14, 15], to discover the cost of communication mechanisms of parallel machines [3], to measure the performance of various operating system primitives [9], to evaluate file systems [2], to extract parameters from SCSI disk drives [19], and even to calculate the megahertz rating of processors [18].

Applying microbenchmarks to disk drives is a particularly vexing problem. Because of the complex drive mechanism involving several cooperating mechanical and electronic parts, many benchmarks that are adequate in other domains do not translate well to disk drives. The rotational factor often affects measurement results and renders the current position of the drive unpredictable. Thus, we seek to develop one or more microbenchmarks that are suitable for extracting performance parameters from modern disk drives. Ideally, our disk microbenchmarks would exhibit the following four properties:

- **General:** Runs across a vast array of systems; it is not specialized for any one specific kind. For disks in particular, the ideal benchmark requires no *a priori* information from the drive being measured.

- **Complete:** Extracts all relevant parameters, including disk geometry and performance parameters. Low-level parameters, including head and cylinder switch times, should not be overlooked.
- **Accurate:** Extracts those parameters with excellent precision.
- **Fast:** Runs quickly, giving useful information in seconds or minutes, not hours or days.

In this paper, we introduce three microbenchmarks designed to extract performance parameters from hard disk drives. In sum total, these microbenchmarks approach the ideal microbenchmark along all four axes: they are quite general, running on any SCSI or IDE drive via a raw-device interface; they run quickly, extracting most (though not all) parameters in a few seconds; they completely characterize the physical properties of a disk drive; and finally, they produce accurate drive geometry and performance parameters, within 3% percent of manufacturer-reported values.

The contributions of this paper are three-fold:

- A novel, but simple, method based on a *linearly increased step-size* for extracting many localized disk parameters, including platter count, sectors/track, rotational delay, head switch time, cylinder switch time, and minimum time to media. By slowly ramping up the step-size, we are able to factor out rotational effects, and thus unveil a host of drive performance characteristics.
- An empirical characterization of a large collection of modern disk drives, including both SCSI and IDE drives. Previous work has focused solely on SCSI-drive extraction [19].
- An update on the results of [10] on the zoned nature of modern disks, including a correction of the proposed linear model to a more accurate quadratic model.

We present results from executing the microbenchmarks on a diverse collection of modern drives, including five SCSI, two IDE, and even two simulated drives. In the course of our study, we have uncovered numerous interesting results. We discovered that the minimum overhead to write disk media widely varies between drives of the same generation. We also found that each family of drives from a particular manufacturer exhibited similar strengths and weaknesses; for example, Seagate drives tend to have excellent switching times. The multi-zoned nature of modern disks is pronounced, with outer tracks delivering 50% to 80% more bandwidth than inner tracks. Not surprisingly, SCSI disks have much different performance characteristics than the IDE disks that we have measured; whereas SCSI bandwidth and switching characteristics are better, the use of programmed I/O instead of DMA renders the IDE drive overhead lower.

The rest of this paper is organized as follows. In Section 2, we give a brief background on disk terminology, followed by related work in Section 3. Section 4 presents an overview of the collection of microbenchmarks. The results for the range of disks is presented in Section 5, and conclusions in Section 6.

2 Background

Before explaining the functionality of our disk characterization tool, we give a brief overview of modern disk drives. For in-depth and excellent summaries of modern disk drive behavior, see [13] and [16].

The basic internal structure of a disk drive is described as follows. There are several rotating disks coated on one or both sides with magnetic media. Each rotating disk is called a *platter*; each side of that disk is called a *recording surface*. Data is stored on each recording surface on concentric circles called *tracks*. Each track is divided into *sectors*; a sector is the minimum unit of data that can be accessed from the disk media. Typical modern disks have 512-byte sectors. The tracks from each surface that are equidistant from the center form a *cylinder*. Most disks use Zoned Bit Recording (ZBR), and thus outer tracks of the disk have a higher sectors/track ratio than the inner tracks.

The read/write heads of each surface are ganged together on the disk arm. The time to move the arm to the proper cylinder is called *seek time* and the time for the required sector to rotate under the head is referred to as *rotational latency*. The time to transfer the data from the media is called *transfer time*. In modern disks, only one head is active at any time. The sector after the last of any given track is the first of the next track in the same cylinder. When an access spans two tracks, the disk must complete the portion on the first track, switch heads, and continue on the second track. The sector mappings on consecutive tracks are skewed to allow for this *head switch time*. Switching heads requires a short repositioning time; the skew prevents a request that crosses track boundaries from missing the next logical block and having to wait a full rotation. Similarly, if an access spans two cylinders, the disk arm has to seek forward one cylinder. Consecutive cylinders are skewed to allow for this *cylinder switch time*.

3 Related Work

We were inspired by two separate works from the literature. The first, by Saavedra [15], presents a novel micro-benchmarking technique for memory systems. The second is a paper by Worthington *et. al.* [19] that describes how to extract performance and geometry parameters from SCSI disks. We seek to combine the simplicity and speed of the former with the accuracy of the latter.

Saavedra introduces a simple yet powerful method to extract performance characteristics from a multi-level memory hierarchy [15]. The benchmark repeatedly performs a basic loop of reading memory locations over a fixed-size array at a given stride. Surprisingly, almost all characteristics of the memory hierarchy, including the number of caches, their capacity, associativity, block size, and access times, can be extracted by simply changing the size of the array and the length of the stride.

Though this technique could be applied to the disk, it does not yield results as desired. Disk subsystems are much less regular than memory hierarchies, and the complex interaction of rotation and seek time leave the direct application of Saavedra to the disk infeasible.

The study by Worthington *et. al.* [19] describes partially automated tools for extracting parameters from SCSI disk drives. They used a twofold approach: interrogative and empirical extraction. Interrogative extraction uses a library of SCSI access functions to read the *mode pages* of the disk. The mode pages describe disk parameters such as the sectors/track ratio, prefetch buffer size, etc. The information extracted from the mode pages is used to construct test vectors for the empirical extraction process. They measure the minimum time between requests (MTBRC) of various kinds. By comparing the MTBRCs of different test vectors, they are able to calculate switching times and other parameters.

The main disadvantage of this approach is that the reliance upon interrogatively-acquired information. In particular, the user must be able to send low-level SCSI commands to the disks, which is highly non-portable, and requires the user to trust that the disk manufacturer has placed all of the necessary information therein. Also, each parameter extracted requires a separate group of test vectors, and the algorithms outlined by Worthington *et al.* can take between minutes to hours to extract those parameters. In contrast, our benchmarks require no low-level access to the disk interface, and in that sense are much closer to true *black box* microbenchmarks. Further, most drive parameters are extracted by the SKIPPY benchmark in a single, fast experiment.

4 Benchmarks

In this section, we present our collection of disk characterization tools. Table 1 summarizes the constituent benchmarks.

Microbenchmark	**What it does**	**What it extracts**
SKIPPY	Linearly increases step distance and writes sector-sized block	Platter count, sectors/track, rotational delay, head switch time, cylinder switch time minimum time to media (writes)
ZONED	Streams through entire disk reading large blocks	Bandwidth (as function of location)
SEEKER	Repeatedly writes sector-sized blocks at start of disk and various locations	Seek cost (as a function of distance)

Table 1: **Microbenchmarks.** *The table describes the collection of microbenchmarks described in this paper.* SKIPPY *is used to extract most parameters, and runs on a very small portion of the disk.* ZONED *produces a bandwidth versus location characterization, extracting a zone profile of the disk. Finally,* SEEKER *generates a seek profile as a function of distance using standard techniques.*

The most innovative component of the benchmark suite is SKIPPY. By utilizing the technique of linearly increasing the stride while writing to the disk, we are able to factor out rotational effects, and thereby extract a surprising amount of information from the disk, including the sectors/track ratio, rotation time, minimum time to access media, disk head positioning time, head switch time, cylinder switch time, and the number of recording surfaces. Also impressive is its run-time; the characterization completes in roughly one second.

SKIPPY alone does not completely characterize the behavior of a modern disk, as it is by nature a *local* benchmark. Two crucial pieces of *global* information are missing: the cost of seeks (as a function of distance), as well as the effect of zones (as a function of location). To derive these final two pieces of information, we utilize two further microbenchmarks, SEEKER and ZONED. Though we constructed these two benchmarks ourselves, they are quite similar to those found in [19] and [10]. Due to their global nature, these benchmarks are somewhat more time consuming than SKIPPY, each taking a few minutes to complete.

All three benchmarks rely on the raw device interface in order to bypass file system optimization activities such as caching, buffering, and read ahead. Without access to the raw interface, these benchmarks would be difficult if not impossible to construct.

4.1 SKIPPY

The SKIPPY microbenchmark implements a novel approach to disk measurement, using linearly increasing strides to counteract the disk's natural rotation. Figure 1 shows the pseudocode of the algorithm. The benchmark writes one sector to the disk, forwards the file pointer, and writes again. With each iteration, the distance (or $StepSize$) increases by a single sector.

The resulting latency versus step size curve has a distinctive sawtooth shape from which we extract the following parameters: sectors/track ratio, rotation time, minimum time to access media, disk head positioning time, head switch time, cylinder switch time and the number of recording surfaces.

We have also gathered results for the read variant of SKIPPYḢowever, for the sake of space, we only present the write version of the benchmark. The read results and analysis will be presented in the full version of the paper.

4.1.1 Intuition and Analytical Foundation

Traditionally, extracting parameters like the head switch time from a disk drive has been difficult since each request incurs an unpredictable rotational latency. The intuition behind the linearly increasing stride method is as follows. Between any two write accesses, the drive rotates some distance forward. Even though different requests will incur different latencies, the time between successive requests reaching the disk is roughly the same. Since the step size between requests is linearly increasing, it will eventually match the distance that the disk rotates between successive requests. This point is clearly observable since all requests prior to it will incur an extra rotation and all requests afterward will not. Basically, since the access pattern is designed to take advantage of the rotational mechanism, it is possible to separate the rotational latency of a request from all other contributing latencies. As a result, other disk characteristics, including head and cylinder switch times, are clearly observable.

To better describe how SKIPPY works, we use a simple analytical model. The model uses the following terms:

- $RotationTime$: Time for one full rotation. *Rotational Latency*, on the other hand, is the time that a given request spends waiting for the required sector to rotate under the head. The *Rotational Latency* can vary anywhere between zero and $RotationTime$.
- $TransferTime$: Time to transfer the data to the media (a per byte cost, not including overheads).
- MTM: Minimum Time to Media. This is the minimum time to access data on the disk surface. A disk request completes in $MTM + TransferTime$ when it incurs no rotational or seek latency.
- STM: Number of sectors that the disk rotates in time MTM. Equation 1 defines STM in terms of MTM, $TransferTime$, and $RotationTime$, as

$$STM = \frac{MTM \cdot Sectors/Track}{RotationTime} \tag{1}$$

Note that Equation 1 assumes a linear relationship between the latency and the number of sectors rotated. It is well known that seek time does not increase linearly with seek distance. However, as stated earlier, the step sizes used do not generate any arm movement; the delay is purely rotational. Since the disk rotates at a fixed speed, the delay increases linearly with the number of sectors rotated.

Figure 2 shows the expected sequence of events for two single sector writes, W1 and W2, on the same track. The figure shows five stages that each write goes through: $Start$, $atDisk$, $atSurface$, $underHead$, and End. The disk

```
fd = open("raw disk device");
for (i = 0; i < measurements; i++) {
    // time the following sequence, and output <i, time>
    lseek(fd, i * SINGLE_SECTOR, SEEK_CUR);
    write(fd, buffer, SINGLE_SECTOR);
}
close(fd);
```

Figure 1: SKIPPY **Algorithm.** *The basic algorithm skips through the disk, increasing the distance of the seek by one sector before every write, and outputs the distance and time for each write. The raw device interface must be used in order to avoid file system optimizations. SINGLE_SECTOR is the size of a single sector, in this case, 512 bytes. The SEEK_CUR argument to lseek moves the file pointer an amount relative to the current pointer.*

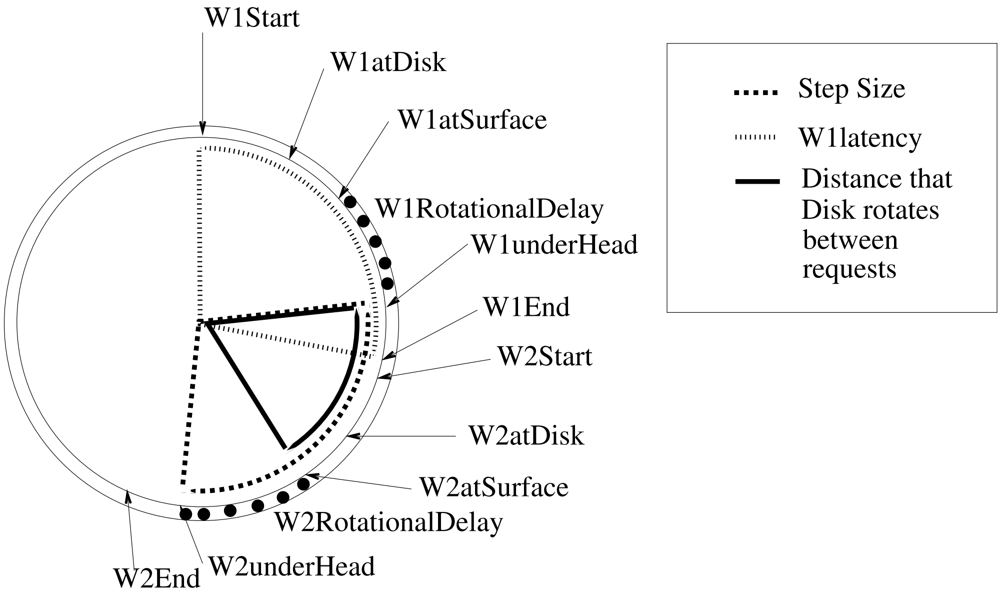

Figure 2: **Behavior.** *This figure shows the expected sequence of events for two 1 sector writes to the disk media. The writes are labeled W1 and W2.*

rotates a few sectors between each stage. The illustration does not show $TransferTime$; since we are focusing on single sector accesses, the transfer time is nearly negligible.

$W1$ starts at time $W1Start$. By time $W1atDisk$, the OS and SCSI subsystem have processed the request and the command has reached the disk. By time $W1atSurface$, the disk has positioned the head on the necessary track. By time $W1underHead$, the required sector is under the disk head. The difference between $W1atSurface$ and $W1underHead$ is the *Rotational Latency* for $W1$. By $W1End$, the write system call has returned. Some short time later, the second write begins. By $W2atSurface$, the disk has already rotated some distance forward. In the illustration, the step size is greater than this distance; the required sector is still up ahead and the second request can be served in the same rotation.

The time between $W1End$ and $W2Start$ is the time to start the next loop iteration and execute the `lseek` call; this time is negligible compared to the disk access times. In our system, it is 7 to 8 microseconds, on average, while the entire write takes between 2 and 10 ms. If we assume that the time between $W1End$ and $W2Start$ is negligible, we can make two interesting observations:

(i) As the rotational delay approaches zero, $W2End - W2Start$ becomes $MTM + TransferTime$.

(ii) If the rotational delay is eliminated from the figure, $W2atSurface - W1atSurface$ is also MTM. Therefore, the disk rotates for approximately MTM time, or over STM sectors, between any two requests. In other words, if $StepSize < STM$, $W2$ will need an extra rotation. If $StepSize > STM$, $W2$ can complete in the same rotation as

$W1$.

Using the above logic, we can model the latency of the second write request. If the access is on the same track as the prior access, (i.e $CurrentPosition + StepSize < Sectors/Track$), and $StepSize > STM$, the request can be satisfied in the current rotation and the latency is:

$$Latency = \frac{(StepSize - STM) \cdot RotationTime}{Sectors/Track} + MTM + TransferTime \quad (2)$$

This latency is the minimum time to media plus the time to rotate the remaining sectors. Substituting Equation 1 into 2 gives us a simpler term for the latency:

$$Latency = \frac{StepSize \cdot RotationTime}{Sectors/Track} + TransferTime \quad (3)$$

As the equation shows, the latency is a linear function of the step size. If $StepSize < STM$, the request is satisfied in the next rotation, and the latency is given by Equation 4:

$$Latency = \frac{(Sectors/Track + StepSize - STM) \cdot RotationTime}{Sectors/Track} + MTM + TransferTime \quad (4)$$

Equation 4 can also be simplified by substituting 1. The substitution gives Equation 5:

$$Latency = \frac{StepSize \cdot RotationTime}{Sectors/Track} + RotationTime + TransferTime \quad (5)$$

When $StepSize < STM$, the latency is still linear in Sectors/Trackwith an offset equal to the Rotational Latency. If the step puts the new request on a different track, then the request incurs an extra head switch delay. Since the tracks are skewed, a head switch does not cause the disk to have to wait a full rotation. In this case, the latencies can be calculated as in Equations 6 and 7, When $StepSize > STM$:

$$Latency = \frac{(StepSize - STM) \cdot RotationTime}{Sectors/Track} + MTM + HeadSwitchTime + TransferTime \quad (6)$$

When $StepSize < STM$:

$$Latency = \frac{(Sectors/Track + StepSize - STM) \cdot RotationTime}{Sectors/Track} + MTM + HeadSwitchTime + TransferTime \quad (7)$$

The equations for a cylinder switch are similar, with $CylinderSwitchTime$ in place of $HeadSwitchTime$.

Note that all these equations assume that there are no long distance seeks going on. This model and SKIPPY are not intended for step sizes that cause seeks greater than a single cylinder. After that point, there is significant arm movement and the latency does not scale linearly with step size.

Now we illustrate the expected graphical result using a mock disk. The mock disk is 7200 RPM, with 15 recording surfaces containing 150 sectors per track. The minimum time to access media is 2.0 ms, and the head and cylinder switch times are 0.7 ms and 2.1 ms respectively. Since the benchmark does not create long distance seeks, we do not specify a seek profile.

Figure 3 shows the expected graphical result; the accompanying illustrations, shown in Figures 4 through 7 reveal what happens at points (1) through (4) in the graph. Each illustration shows two writes; the second write shows the request pattern at the marked point in the graph. Each track is shown as two concentric circles; the rotational delay for $Write1$ is marked on the outer circle and the rotational delay for $Write2$ is marked on the inner circle.

As the $StepSize$ increases linearly, the latency follows a sawtooth pattern. At point (1), (Figure 4) $StepSize$ is zero, causing a large rotational delay for $Write2$ and making $W2Latency$ equal to the $RotationTime$. As $StepSize$ increases, the latency increases linearly as in Equation 5. When $StepSize$ approaches STM, Equation 5 shows that the latency approaches $MTM + RotationTime$. At point (2) (Figure 5), $StepSize$ is almost STM. By the time the disk head is lowered on the track, the required sector has just been missed and a full rotation takes place. The latency is therefore the $RotationTime$ plus MTM overhead.

A few steps later, we reach point (3) (Figure 6), where $StepSize$ is slightly larger than STM. In this case, the disk head is lowered just in time and there is no rotational latency. The latency therefore becomes MTM. From then on,

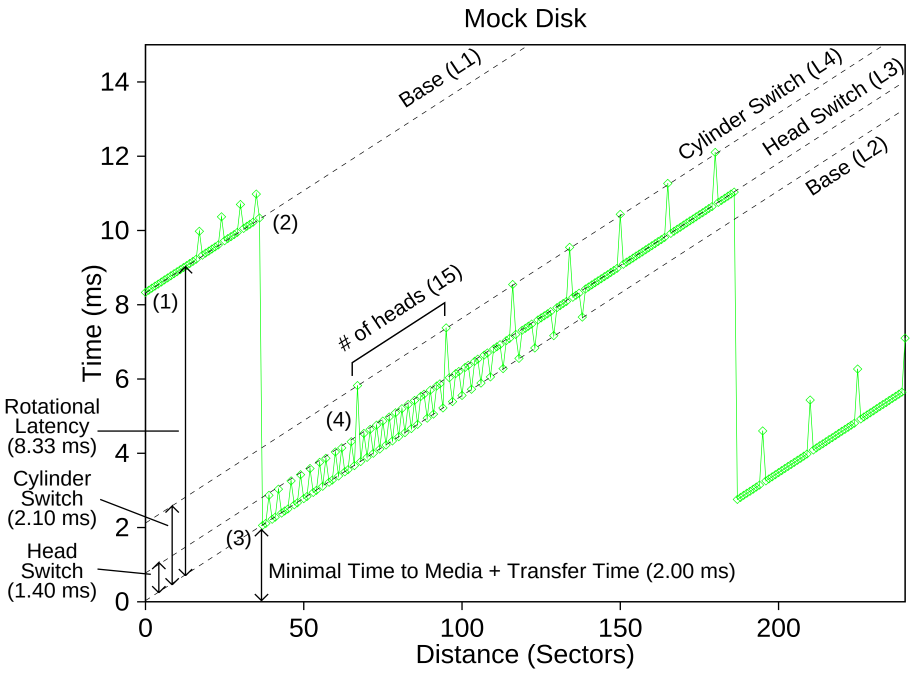

Figure 3: **Mock Disk.** *Example output from a mock disk is shown. The graph is constructed strictly from the models developed within the text, to illustrate what the output of the benchmark should look like.*

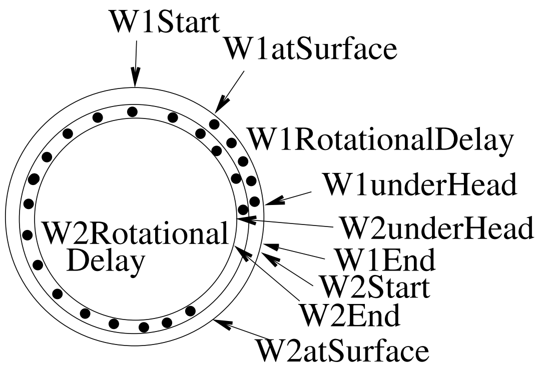

Figure 4: **Point (1).**

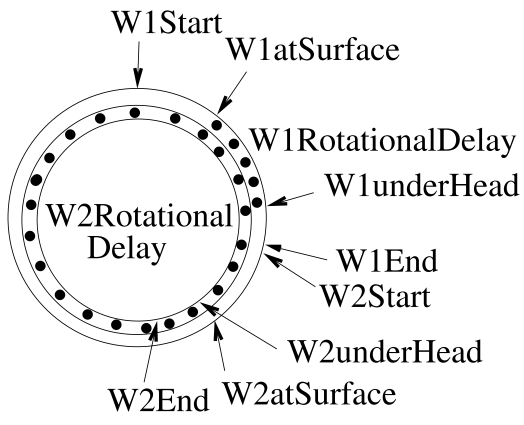

Figure 5: **Point (2).**

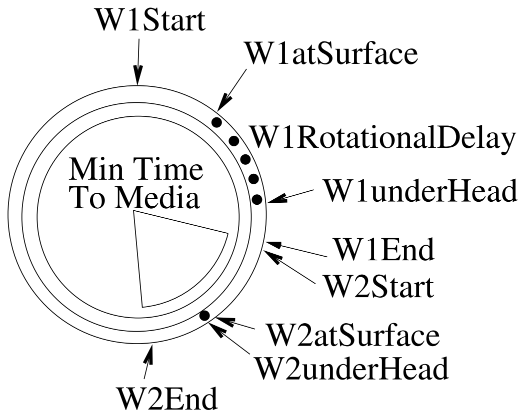

Figure 6: **Point (3).**

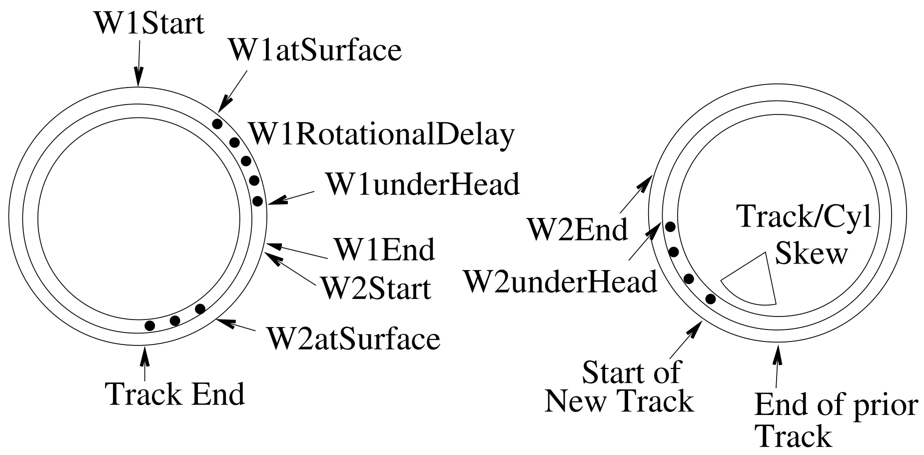

Figure 7: **Point (4).**

as $StepSize$ increases, the latency increases linearly as in Equation 3. The graph has a sawtooth shape; the transition happens at $StepSize = STM$.

The graph also shows a series of upward spikes that correspond to head and cylinder switches. Point (4) (Figure 7) illustrates a head switch. In this case, the rotational latency is increased by $HeadSwitchTime$ as specified by Equation 6.

4.1.2 Extracting the Parameters

Figure 3 exposes many useful disk details. The X coordinate of point (3) is STM and the Y coordinate is $MTM + TransferTime$. Since the transfer time is very small for a single sector, the Y coordinate of point (3) is a good estimate for MTM. MTM is also the difference between the Y coordinates of points (1) and (2). $RotationTime$ is the latency at step size 0 and the height of the transition at the MTM point.

Using this information, we can calculate the number of sectors per track. Since we know that the MTM point is reached when $StepSize = STM$, we can reverse Equation 1 to calculate the $Sectors/Track$.

Note that we can only calculate the sectors/track ratio for the region that was written. Most modern disks employ Zone Bit Recording (ZBR); outer cylinders can be packed with more sectors/track than inner cylinders, due to the circular nature of disks. To get the sectors/track ratio of other regions in the disk, one would need to run the benchmark on those regions as well.

As $StepSize$ increases, the latencies form three distinct lines with the same slope and different offsets. Figure 3 shows four lines labeled L1 through L4. L1 conforms to Equation 4 and L2 conforms to Equation 2. By taking the difference in offsets between these two lines, we can calculate $RotationTime$. The slope of each line is $\frac{RotationTime}{Sectors/Track}$. Once $RotationTime$ is known, we can extract $Sectors/Track$ from the slope value.

Each point on L3 represents a head switch and the latencies conform to Equation 6. Hence, the vertical offset between the L3 and the L2 is $HeadSwitchTime$. Each point on L4 corresponds to a cylinder switch; the vertical offset between the third line and L2 is $CylinderSwitchTime$.

Finally, while the step size is less than the number of sectors per track, we can get the number of recording surfaces by counting the number of head switches between two cylinder switches. As $StepSize$ gets larger, the number of steps between successive head and cylinder switches decreases. As Figure 3 shows, eventually, nearly every step results in a head switch.

4.1.3 A Sample Result

The prior section showed how all the mock disk's parameters can be extracted from Figure 3Ṅow we apply these techniques to the IBM UltraStar XP disk drive [6]. From the manufacturer specifications we learn that this disk is 7200 RPM (8.33 ms rotational latency), with 18 recording surfaces and 8 recording zones, the outermost of which has 184 sectors per track. The head and cylinder switch times are 0.85 ms and 2.17 ms respectively.

Figure 8 shows the result of running the benchmark on this disk; the figure is quite similar to the model result in Figure 3. The result follows the behavior predicted by Equations 1 through 6. Equation 7, however, does not completely explain the result of head and cylinder switches while $StepSize < STM$. In Figure 3, head switches always cause upward latency spikes consistent with 7; Figure 8 shows upward spikes for small $StepSize$ and also some downward spikes as $StepSize$ approaches STM. This variation does not affect our ability to extract the necessary parameters, but it does require a refinement of the analytical model. We refine the model in the next subsection; for now, we focus on extracting parameters from Figure 8. The error rates described below are calculated by comparing the extracted values to the manufacturer specified values.

Following the parameter extraction techniques described earlier, we get the following measured values. $RotationTime$ from the Y coordinate at point (1) is 8.39 ms; the actual latency is 8.33 ms and the error is 0.73%. If we use the height of the sawtooth wave to estimate $RotationTime$, we get 8.30 ms. In this case, the error is 0.43%. Both techniques yield extremely accurate results.

The X coordinate value of point (3) is 47 and the Y coordinate value is 2.20ms. As Equation 2 states, the offset of L2 is $TransferTime$; this value is 0.06 ms. Since we are writing only 512 bytes, the transfer time is very small. By subtracting $TransferTime$ from the Y coordinate value at point 3, we estimate MTM to be 2.1 ms. In fact, since the transfer time is so small, its effect on the MTM value is virtually indistinguishable from measurement noise and the Y coordinate value is itself a good estimate of MTM. On the other hand, if we estimate MTM as the difference

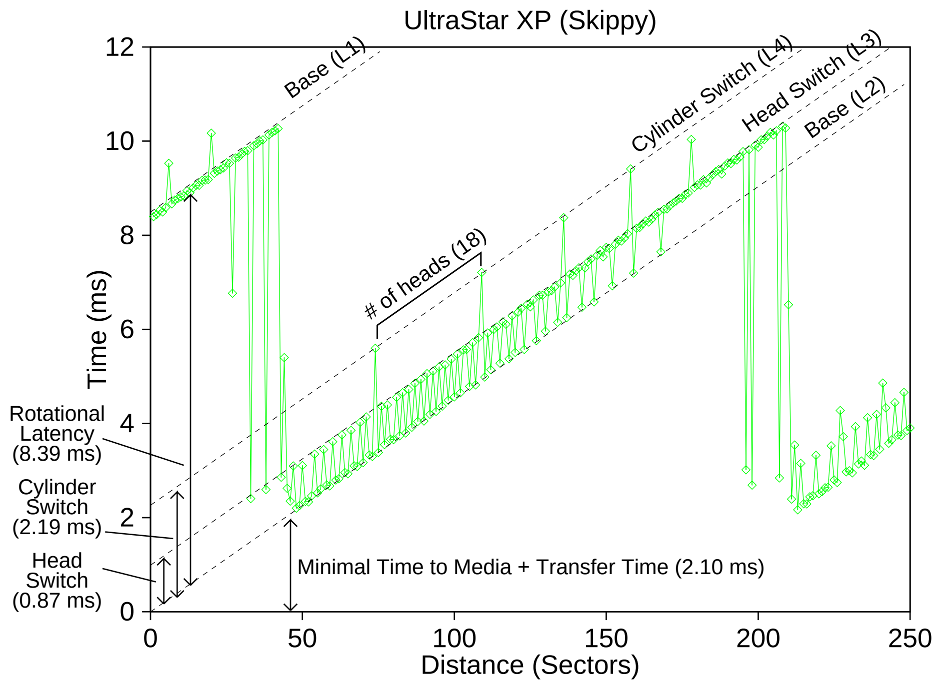

Figure 8: **IBM UltraStar XP.** *Sample write result from the IBM UltraStar XP disk drive.*

between the Y values at points (1) and (2), we get 1.87 ms. Although, MTM has no counterpart in the specification, it represents an important and useful estimate of system I/O overhead.

The $Sectors/Track$ ratio is 181.9; since the actual sectors per track is 184, the error is 1.1%. The measured $HeadSwitchTime$ is 0.87 ms, a 2.3% error compared to the specification. Similarly, the measured $CylinderSwitchTime$ is 2.19 ms, a 0.9% error compared to the specification. Finally, by counting the number of head switches between cylinder switches, we find that the disk has 18 recording surfaces. This value matches the disk specification. For this disk drive, the extracted values are very close to the actual values. In all cases, the error rate is less than 3%.

4.1.4 A Refined Analytical Model For Writes

Figure 8 showed that the simple model is inadequate for describing some parts of the benchmark behavior. In particular, the graph shows some downward spikes in the region $StepSize < STM$ that are not explained by Equation 7. In this section, we present a refinement to the initial model to explain these effects.

In Figure 8, the downward spikes near point (2) happen when a head switch occurs while $StepSize$ is close to STM. In normal circumstances, when there is no head switch, the mechanics of Equation 4 apply; there is not enough time to position the head and service the write in the same rotation as the prior write. When a head switch occurs, however, the track skew gives the disk head slightly extra time, enabling the disk to service some writes *without* waiting an extra rotation. These writes can complete with latencies close to MTM. Figure 8 shows that these downward spikes actually extend L3, the head switch line, to the left of the MTM point. This observation confirms our hypothesis that the spikes are caused by head switches.

By adjusting the model to account for these effects, we are able to create a model graph that looks identical to the sample result. Also, since the extra downward spikes reveal an interaction between the disk head positioning and a head switch, we can use this property to estimate the disk head positioning time. Again, the full paper will contain more details on this refinement.

4.1.5 Limitations

One limitation of the write SKIPPY technique is that it does not work on disks using delayed write optimizations (this limitation also applies to all microbenchmarks that attempt to measure write latencies). However, such disks are measurable using the read SKIPPY variant.

```
fd = open("raw disk device");
while (read(fd, buffer, LARGE_SIZE) == LARGE_SIZE) {
    transfer += LARGE_SIZE;
    if (transfer >= REPORT_SIZE) {
         // output location and bandwidth achieved over region
         transfer = 0;
    }
}
close(fd);
```

Figure 9: ZONED **Algorithm.** *The benchmark simply reads the disk sequentially, in blocks of size LARGE_SIZE. When a threshold amount has been read (REPORT_SIZE), the benchmark outputs the location as well as the bandwidth achieved over the region.*

As the full paper will show, the read result is slightly different from the write result since there is some interaction with the read-ahead mechanism at smaller cases. In almost all cases, however, all parameters are extractable with the read benchmark.

A slight variant on the basic read benchmark, a *backwards read* benchmark, strides across the disk in the reverse direction and measures the same parameters as the basic read and write versions, while avoiding some read-ahead optimizations which tend to obscure results. We plan to investigate its utility in future work.

We have also developed a tool to automatically extract the parameter values from the graphical result. The extraction algorithm utilizes work in the statistics and image-processing communities to process the latency versus $StepSize$ data and extract all the parameters listed in Table 3. More details will be available in the final version of the paper; the extraction tool will be made available online along with the benchmark tool set.

4.2 ZONED

This subsection briefly describes ZONED, a microbenchmark designed to extract a bandwidth profile across the different recording zones of the disk. The basic algorithm is depicted in Figure 9, and is quite straight-forward.

Figure 10 shows the algorithm's result on the UltraStar XP disk drive. From the manufacturer specification, we learn that the disk has eight recording zones, with Sectors/Track ranging from 184 at the outermost zone to 120 at the innermost zone. The graph clearly shows the recording zones. Earlier, we demonstrated how SKIPPY can extract Sectors/Track in the local area where it is run. By running SKIPPY in each zone defined by Figure 10, it is possible to extract Sectors/Track at each zone in the drive. Using this technique, we learn that the first and largest zone has on average 187.36 sectors per track. The Sectors/Track values for all subsequent zones are 179.85, 167.66, 155.82, 147.76, 142.10, 134.14, and 120.39, respectively. All values match the specifications in [6] to within 2%.

One also can observe the large difference in delivered bandwidth across the zones of the drive. In the outermost zone, bandwidth is roughly 9.68 MB/s, whereas the inner tracks deliver 6.29 MB/s, roughly a 54% increase from inner to outer tracks.

4.3 SEEKER

The second global disk characteristic missing is the seek profile. Fortunately, seek delays are based solely on the mechanical movements of the disk arm, and have been thoroughly explored in several prior studies. We limit our discussion of seeks, therefore, to the following. First, we present a variant of the SKIPPY technique that can be used to make seek experiments easier by factoring out the rotational latency component of the measured time. Second, we present seek curves as a function of sector distance, not cylinder distance.

Since SKIPPY is a local benchmark, it cannot be directly used to measure seek distances. However, we can utilize a slight variant, as described in Figure 11. Between each of the measurements, the algorithm writes to a fixed location at the beginning of the disk. This variant allows the same disk space to be reused, and creates a similar (although not identical) sawtooth wave whose minimum value can be to estimate the seek time if the rotational latency is zero.

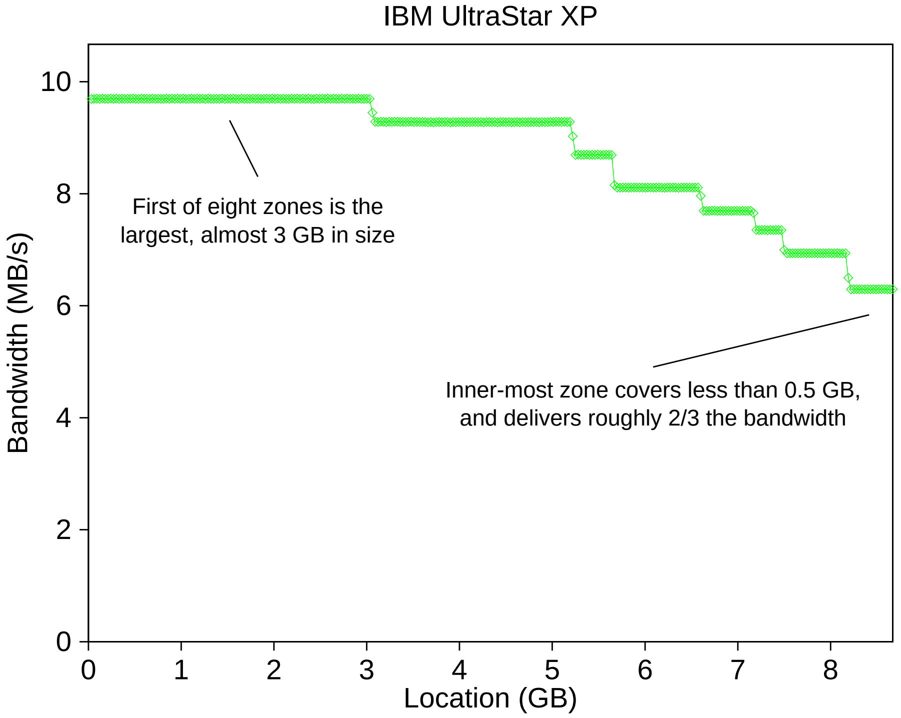

Figure 10: **IBM UltraStar XP.** ZONED *benchmark run on an IBM UltraStar XP.*

Figure 12 shows seek latency versus distance from sector 0 for the Seagate Barracuda. The shape of the curve is slightly different from most seek curves seen in papers and textbooks, since it is seek time versus sectors and not versus cylinders. Also note that the minimal time to media (MTM) is included in the values reported; the true seek values can be obtained by subtracting off the MTM value derived by the SKIPPY benchmark.

The basic algorithm described above suffers from two limitations. The first, similar to SKIPPY, is that it will not work if the drive does not immediately write the data through to disk. Second, since the same disk area is reused, a read version is not likely to work since most of the nearby disk sectors will be found in the buffer cache.

5 Results

This section presents our results for a range of modern SCSI and IDE drives, as well as for two simulated drives. Table 2 lists the drives that we measured. Each real drive was measured on a Pentium II with 256MB of memory, running FreeBSD version 2.2. The SCSI drives were connected via a Fast-Wide SCSI-2 bus.

Figures 14 to 21 show the results of SKIPPY, SEEKER, and ZONED, on each drive. Table 3 summarizes the extracted numbers from the benchmarks for each disk.

5.1 SKIPPY

5.1.1 SCSI Disk Drives

In all the SCSI disk cases, $RotationTime$ is clear from the height of the sawtooth wave. For the 5400 RPM Hawk, the wave is about 11.22 ms high; the error rate is 0.9%. For the 7200 RPM Barracuda disk, it is about 8.43 ms; the error rate is 1.2%. The estimated $RotationTime$ for the Micropolis disk is 8.41 ms; the error rate is 0.9%. Finally, the rotation time of the 10,000 RPM 9ZX is 6.06 ms, giving an error of 1.0%.

The measurements show that MTM can vary somewhat between disks of the same RPM and generation. The Hawk's average MTM is 1.9 ms, while the MTM for the 7200 RPM disks ranged from 1.8 ms to 3.8 ms. The lowest, 1.8 ms, was the Seagate Barracuda, while the highest, 3.8 ms, was the Micropolis drive. Finally, the latest disk, the 10000 RPM 9ZX, had the lowest MTM value of 1.4 ms. Since these disks were measured on the same testbed, we

```
fd = open("raw disk device");
for (base = 0; base < DISK_SIZE; base += LARGE_SIZE} {
    for (i = 0; i < measurements; i++) {
        lseek(fd, 0, SEEK_SET);
        write(fd, SINGLE_SECTOR);

        // time the following sequence, and output <location, time>
        lseek(fd, base + (i * SINGLE_SECTOR), SEEK_SET);
        write (fd, buffer, SINGLE_SECTOR);
    }
}
close(fd);
```

Figure 11: SEEKER **Algorithm.** *Pseudocode of seek algorithm is presented. The benchmark jumps betweeen the beginning of the disk and the target locale, writing a single sector each time. The time for the second write is timed. This is performed repeatedly for many parts of the disk, as shown by the loop over 'base'. The SEEK_SET argument moves the file pointer to the absolute (not relative) location specified by the call.*

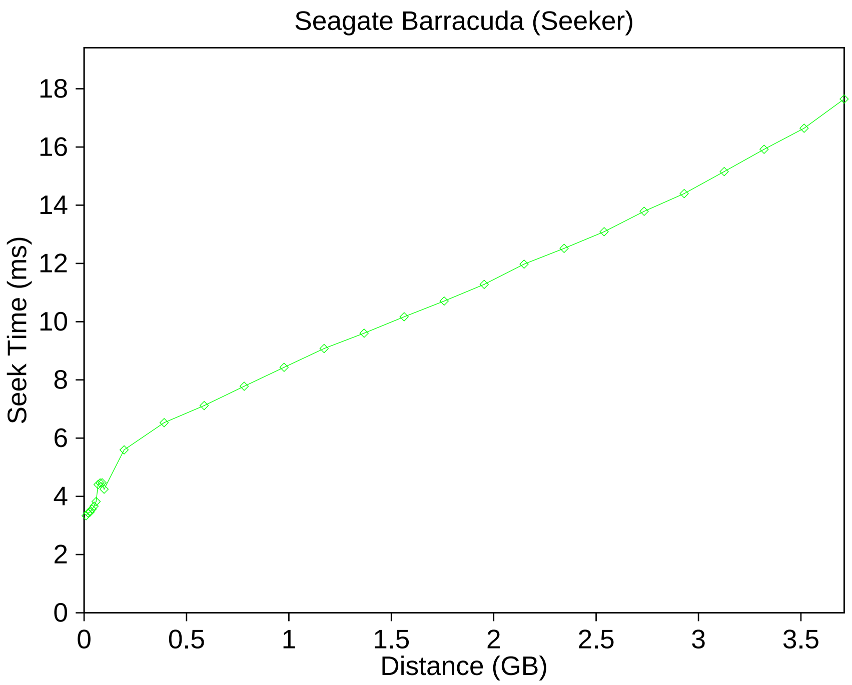

Figure 12: **Seagate Barracuda Seek Curve.** *Seek profile of the Seagate Barracuda. Note that seek time is non-linear under under small seeks, whereas a linear model is quite appropriate under long-distance seeks. Also note that the seek time reported includes the MTM.*

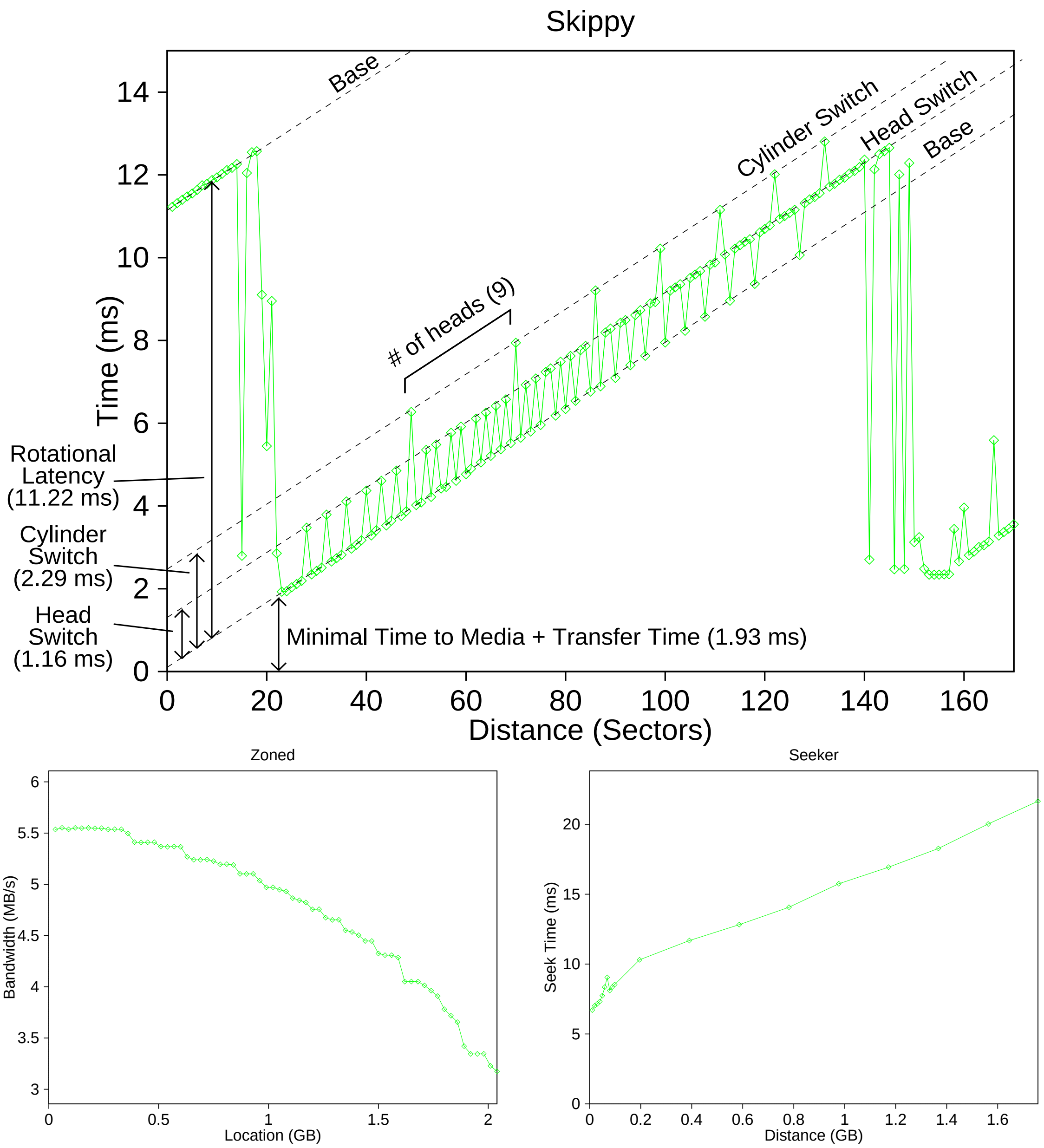

Figure 13: **Seagate Hawk.** *Results are presented for the Seagate ST32430W, referred to as the Hawk. Note that for a disk of an older generation, the head and cylinder switch times are quite good. The disk also has quite a large number of zones, as is typical in Seagate disks. Finally, the seek curve is quite standard.*

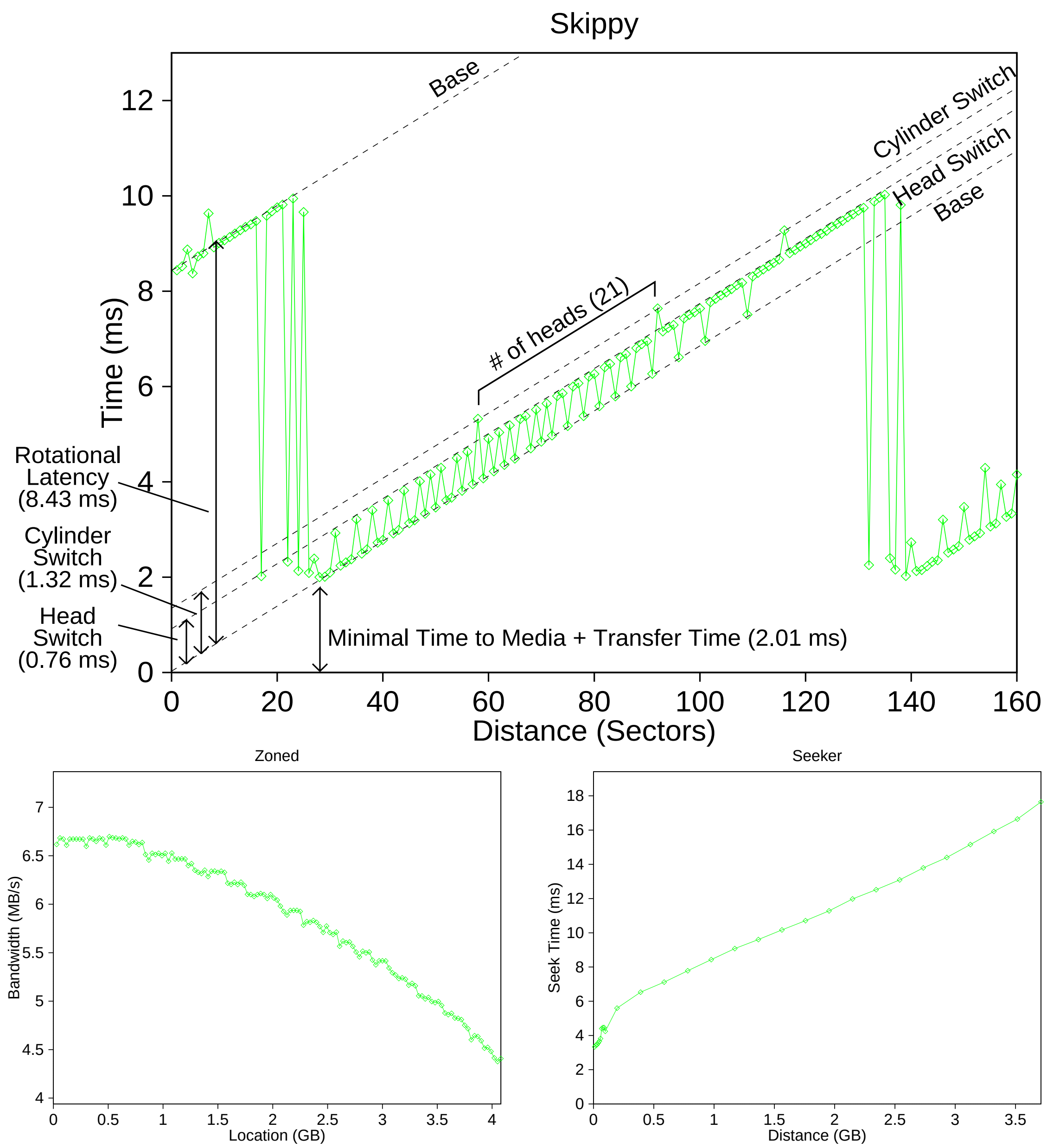

Figure 14: **Seagate Barracuda.** *Results are presented for the Seagate ST15150W, referred to as the Barracuda. Note the excellent head and cylinder switch times in the* SKIPPY *curve, as well as the large number of platters. Seagate devices seem to have an odd number of platters; our hypothesis is that the extra platter is used for position-sensing information. The* ZONED *curve shows a large number of zones, so small as to become indistinguishable. Finally, the* SEEKER *curve ranges from just above 3 ms all the way to about 18 ms.*

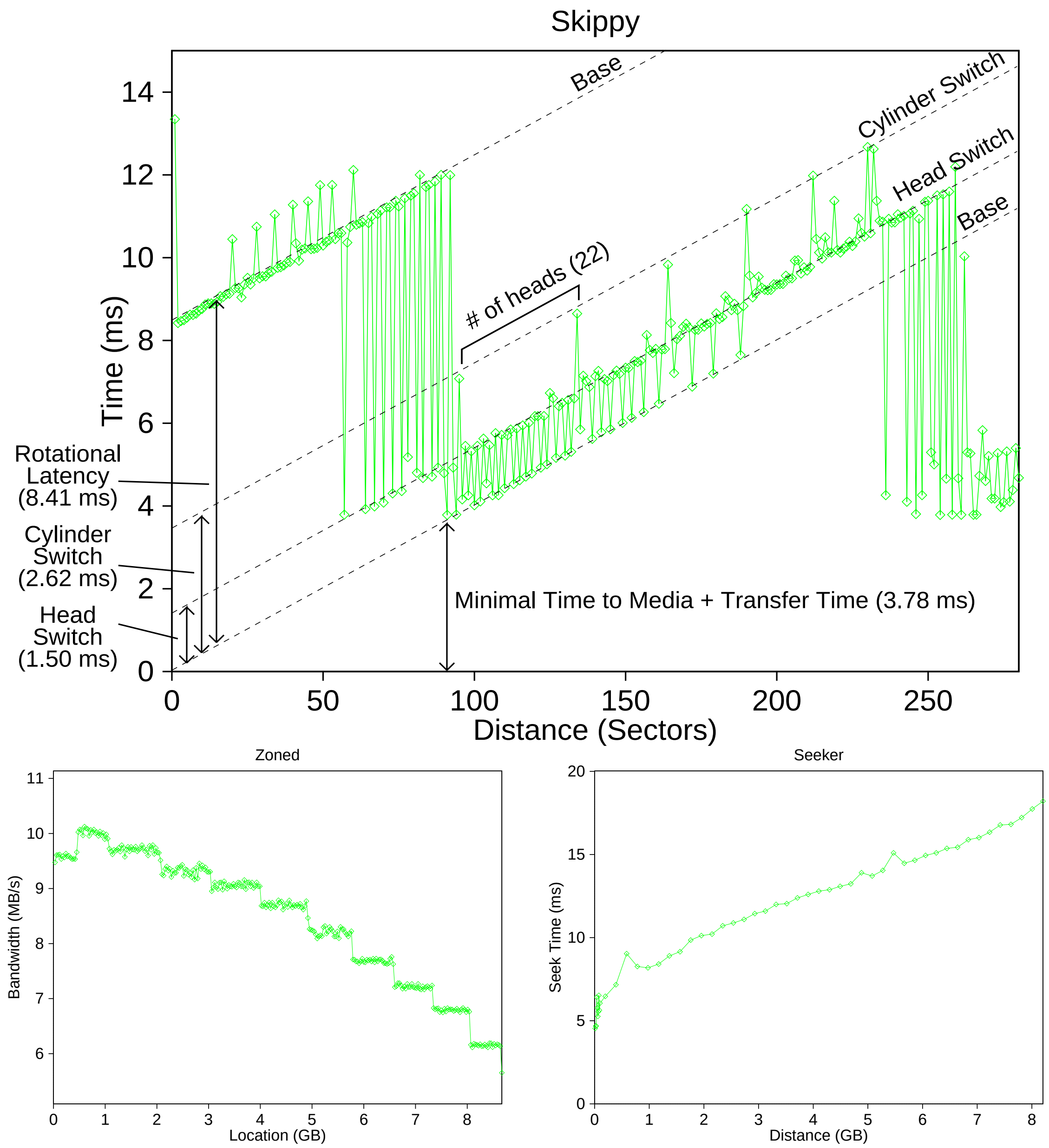

Figure 15: **Micropolis.** *The Micropolis disk is one of the worst performers in the SCSI class, with poor switch times and an exceptionally high MTM. The zone profile is somewhat odd, in that the second zone delivers notably higher performance than the first. We have no explanation for this behavior at this point.*

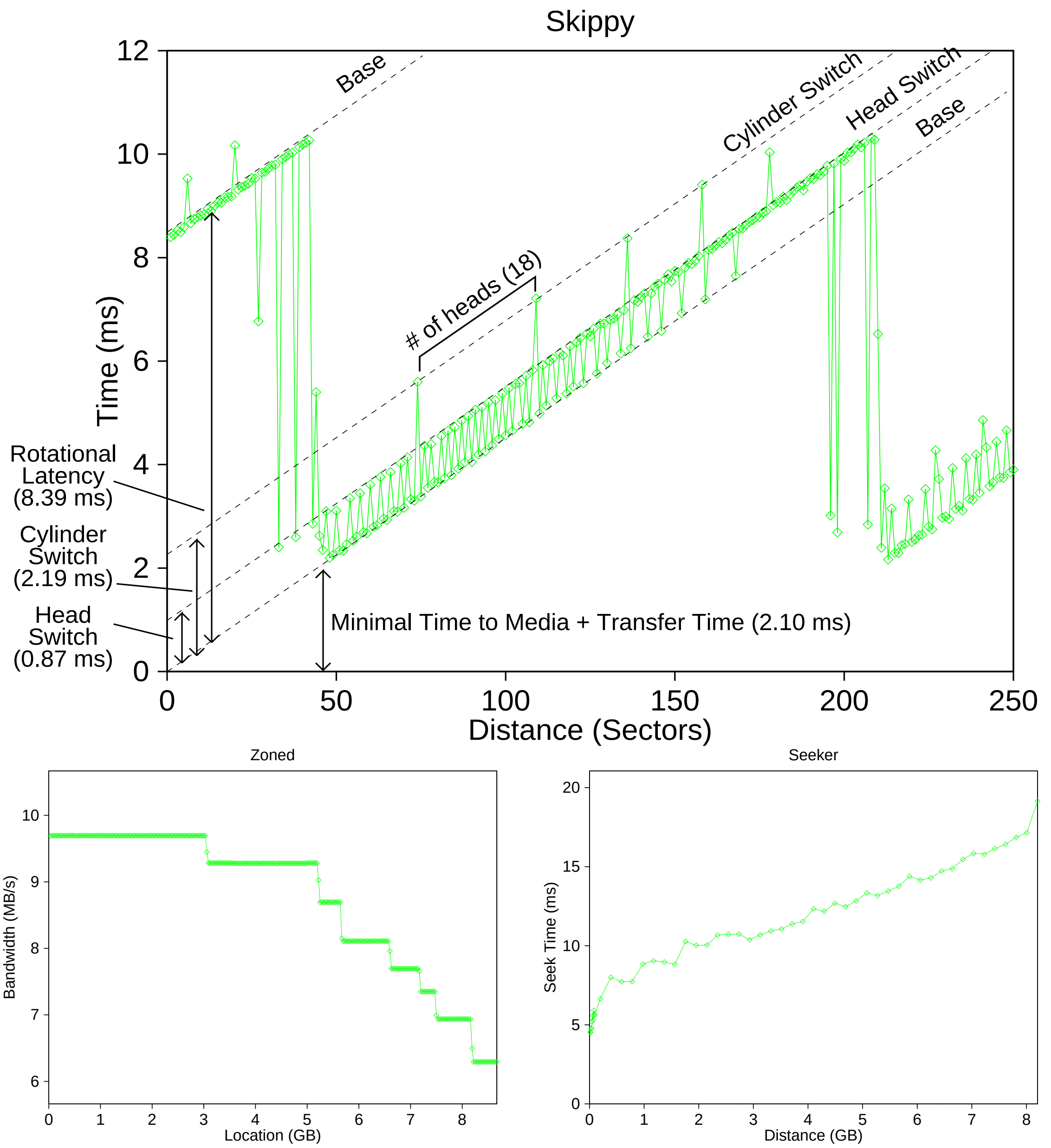

Figure 16: **IBM UltraStar XP.** *This disk presents us with a very typical output curve, with reasonable switch times and MTM. There are only 8 zones on the disk, and the outermost two zones occupy more than half of the disk. The seek numbers are quite noisy, even under repetition; we currently have no explanation as to this effect.*

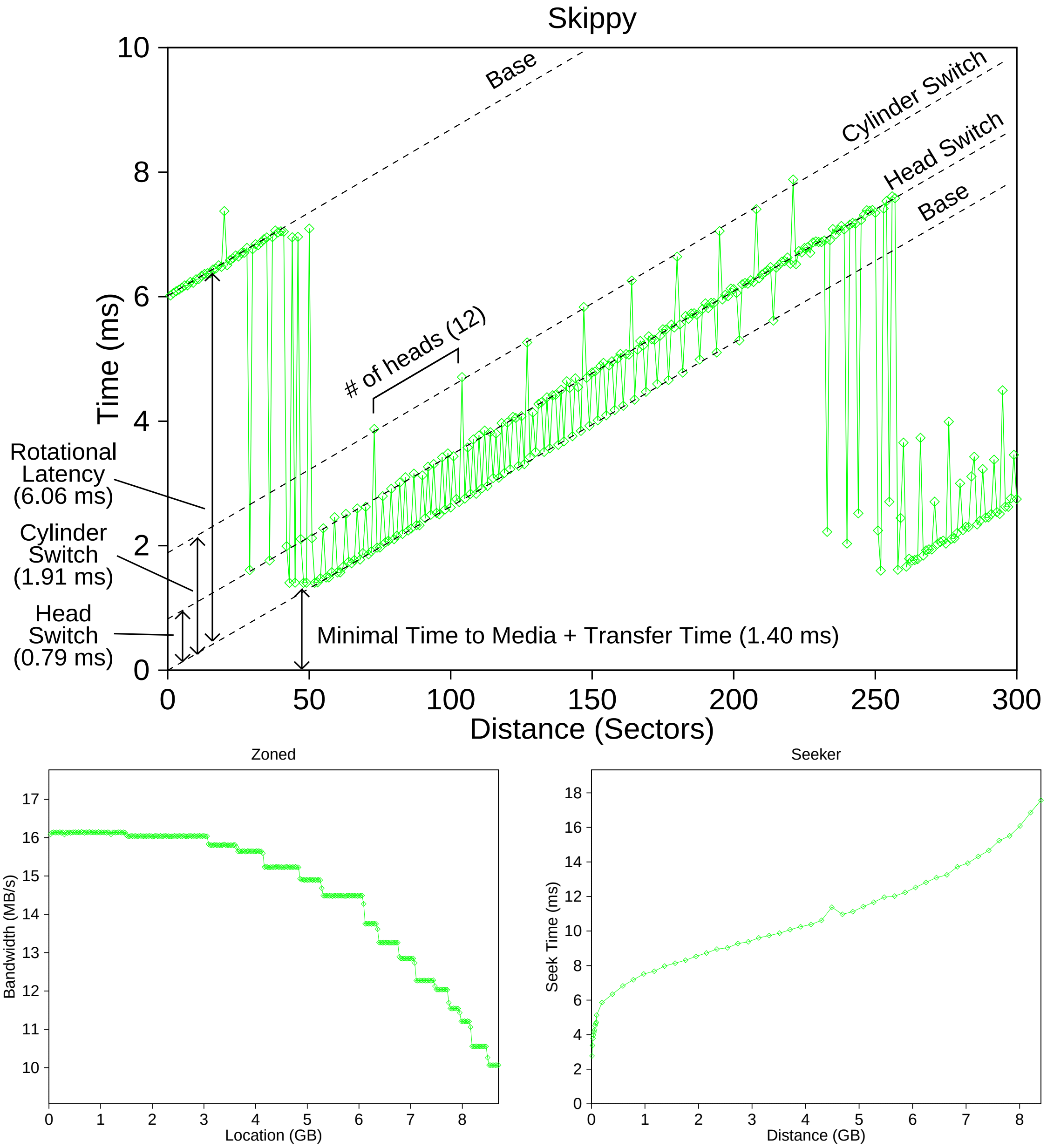

Figure 17: **IBM 9ZX.** *One of the more modern disks in the study, this is the fastest rotating disk, at 6 ms per revolution (10000 RPM). With such low rotational latencies as well as low small-seek costs, the head and cylinder switch times become much more prominent. The zoning of this and other IBM disks is more distinct than the zoning found in Seagate disks.*

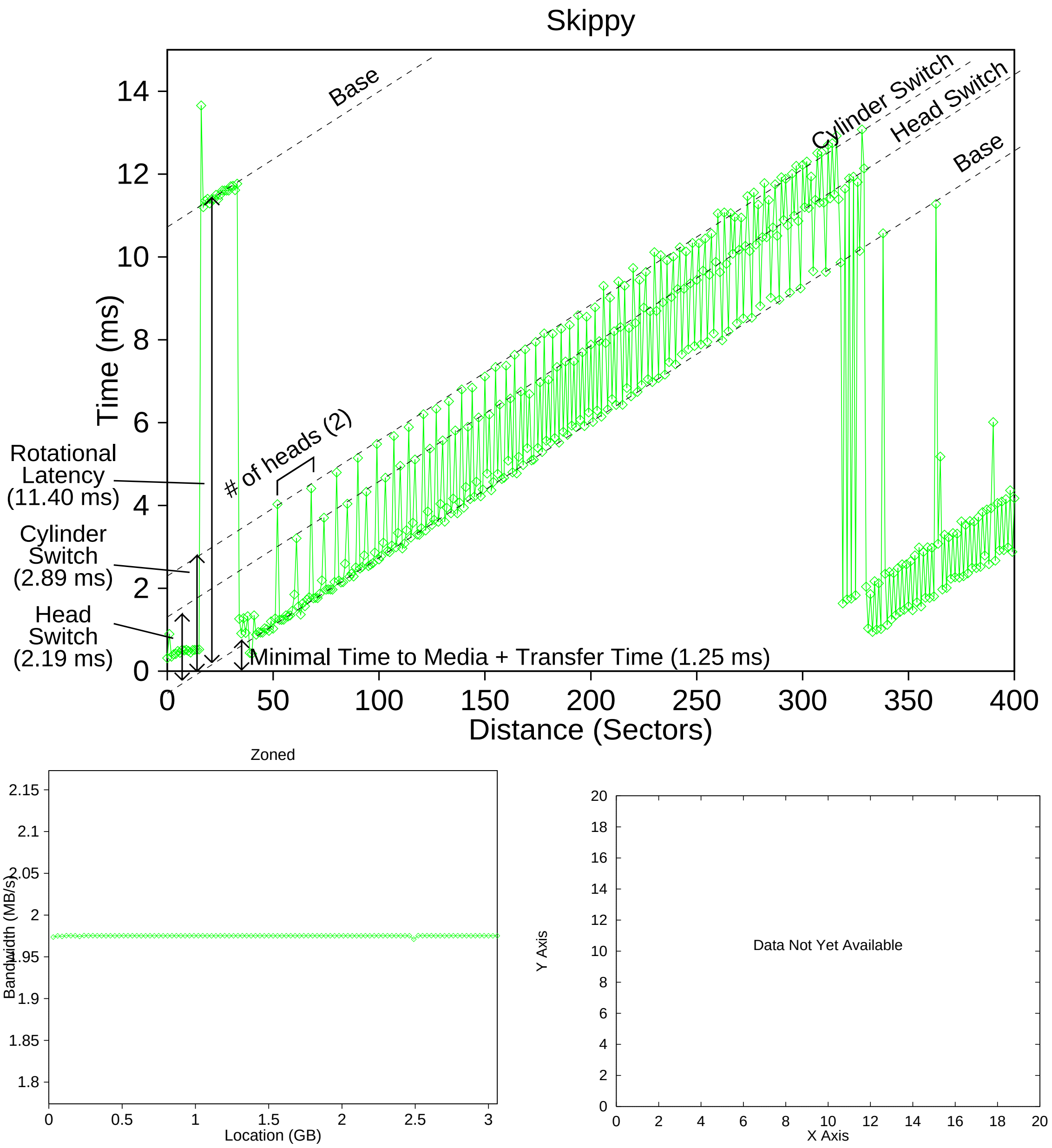

Figure 18: **Quantum Fireball (IDE).** *One of the two IDE disks in the study. This low profile disk has only two recording surfaces. Being one of the more recent drives, it also has a high Sectors/Track ratio. The single zone definition suggests that the drive manufacturers chose simplicity over performance. We were unable to generate the seek profile. This figure will be available in the full paper.*

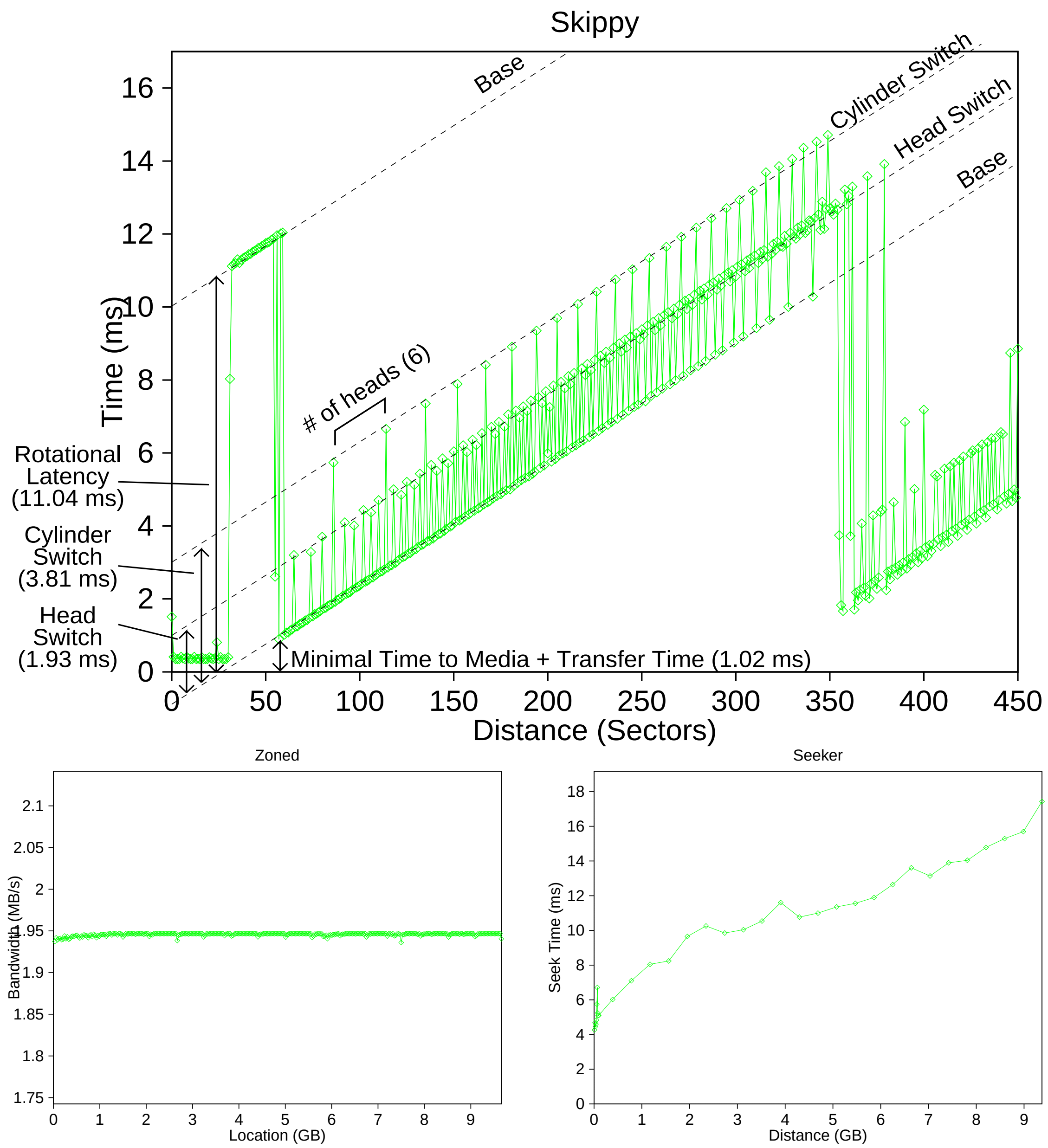

Figure 19: **IBM (IDE).** *The second IDE drive in the study. Although more recent than most of the SCSI drives, it has considerably lower bandwidth and higher switching times.*

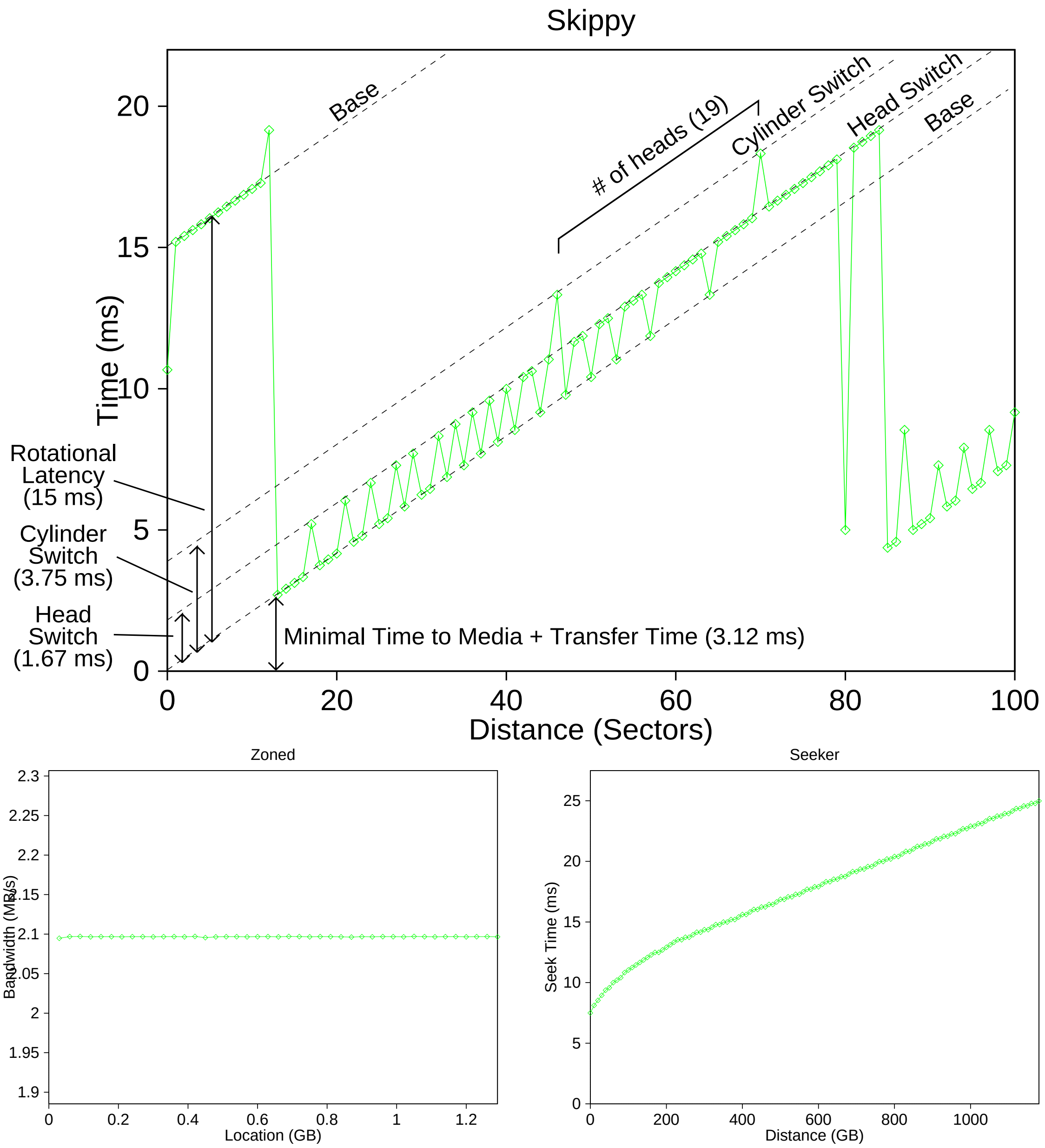

Figure 20: **HP 97560 (simulated).** *The Dartmouth simulator accurately simulates a disk from quite an old generation, as is indicated by the high rotational latency, switch times, and minimal time to media. As is true with many disks from that era, there is only one zone for the entire disk. Finally, the seek profile is quite regular, and matches the formula utilized by the authors exactly. Note that the data from the simulator is much cleaner than any of the real-world disks.*

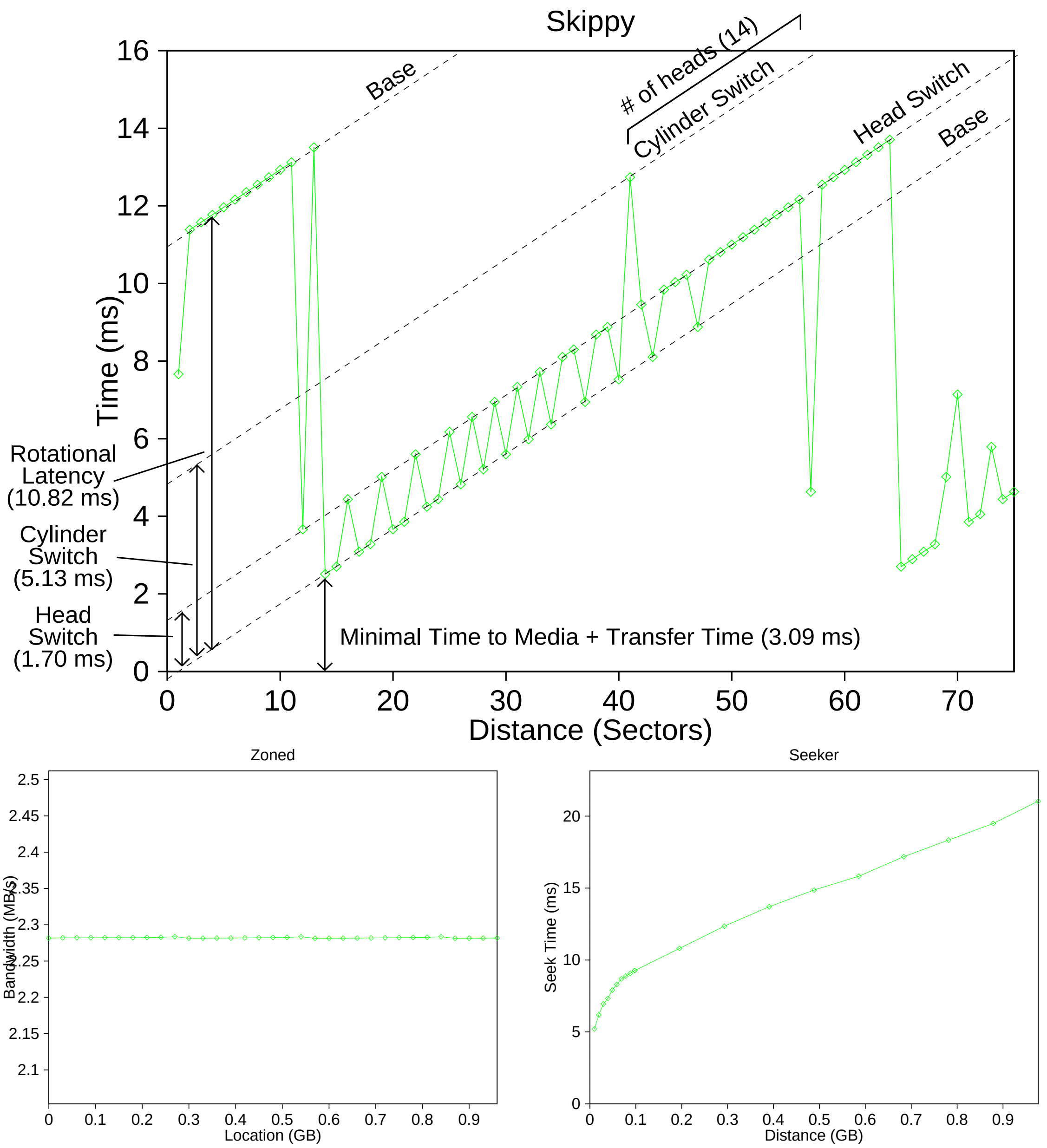

Figure 21: **DEC RZ26 (simulated).** *The data from* diskSim*, the Michigan disk simulator, is presented. This disk is somewhat more modern than the Dartmouth simulator, and does show some more realistic behavior, as described in the text. The zone profile is again quite uninteresting, and the seek curve is as expected.*

Model	Year	Interface	Capacity (GB)	Dimensions	RPM	Rotational Latency (ms)	Recording Surfaces
Seagate ST32430W "Hawk" [17]	1994	SCSI	2.05	3.5in, LP	5411	11.1	9
Seagate ST15150W "Barracuda" [17]	1995	SCSI	4.10	3.5in, HH	7200	8.3	21
Micropolis 3391 SS [11]	1996	SCSI	8.69	3.5in, HH	7200	8.3	22
IBM UltraStar XP [6]	1996	SCSI	8.69	3.5in, HH	7200	8.3	18
IBM 9ZX [7]	1998	SCSI	8.70	3.5in, HH	10020	6.0	12
Quantum Fireball EX 3.2A [12]	1998	IDE	3.08	3.5in, LP	5400	11.1	2
IBM-DTTA-351010 [7]	1998	IDE	9.6	3.5in, LP	5400	11.1	6
HP 97560 (simulated) [8]	-	-	1.26	5.25in	4002	15.0	19
DEC RZ26 (simulated) [4]	-	-	1.03	3.5in, HH	5400	11.1	14

Table 2: **Disks.** *These disks are between 1 and 5 years old and range from 5400 RPM to 10020 RPM. We only have detailed specifications for the IBM UltraStar XP disk drive. The table contains all the relevant information that we were able to gather for the other disk drives from their on-line specification sheets. All drives are (excepting the simulated disks) 3.5 inch. HH and LP mean "Half Height" and "Low Profile" respectively. The table also describes two simulated disk drives. The first, the HP 97560 drive, represents a trial of the benchmarks on the Dartmouth disk drive simulator. The second, the DEC RZ26, represents a trial on* diskSim*, the simulator developed at the University of Michigan.*

can assume that the operating system and SCSI overheads are similar. Therefore, the results show that the IBM drive has the lowest overhead to access media, with the Seagate Hawk and Barracuda drives not far behind. Interestingly, the Seagate Hawk, which is considerably older than the 7200 RPM drives, still has a better MTM than both the IBM Ultrastar and the Micropolis drive.

Since all of the measured drives employ ZBR, we extract the sectors/track ratio for the outermost zone of each drive. The Hawk has roughly 142 sectors per track, the Barracuda about 123, the UltraStar 186, the Micropolis 201, and the 9ZX about 224.

Finally, we compare the head and cylinder switch times. As the graphs show, the Seagate Barracuda drive has the lowest head and cylinder switch times. The Hawk's cylinder switch time is comparable to that of the UltraStar XP, even though the Hawk is an older drive.

By counting the number of head switches between cylinder, we learn that the Hawk has 9 recording surfaces, the Barracuda has 21, the Micropolis has 22, and the 9ZX has 12. All match the specification data in Table 2. Both Seagate drives use an odd number of recording surfaces, suggesting that they dedicate a surface for track following, as mentioned in [13].

5.1.2 IDE Disk Drives

Figures 18 and 19 show the write behavior for the Quantum and IBM IDE disks. These graphs show caching activity at the lower step sizes. In fact, it appears that the drives write to the buffer cache for several requests, and then empties the cache as each additional request is reached. This behavior causes the entire result graph to shift to the right.

Although the graphs are slightly shifted, we can measure the rotational latency as the height of the transition at the MTM point. The measured $RotationTime$ for the Quantum Fireball is 11.4 ms, a 3% error over the specification value of 11.1 ms. The drive has only two recording surfaces, consistent with the disk specifications in Table 2. The Quantum drive also has a head switch time of 2.19 ms and a cylinder switch time of 2.89 ms.

The measured rotation time of the IBM IDE disk is 11.04 ms, a 0.7% error compared to the specification. The sectors per track ratio is 330.01 and the disk has 6 recording surfaces. The head switch time is 1.93 ms and the cylinder switch time is 3.81 ms. This disk's MTM value is 1.02 ms.

The low MTM values are quite a bit better than most SCSI disks; perhaps this reflects the use of programmed

Disk	Rotation (ms)	MTM (ms)	Sectors per Track	Heads	Head Switch Time (ms)	Cylinder Switch Time (ms)	Bandwidth		Seek Time	
							Outer (MB/s)	Inner (MB/s)	Max (ms)	Min (ms)
Seagate Hawk	11.22	1.93	142.37	9	1.16	2.29	5.51	3.18	19.72	4.78
Seagate Barracuda	8.43	2.01	123.35	21	0.76	1.32	6.70	4.38	15.64	1.33
Micropolis	8.41	3.78	201.72	22	1.50	2.62	10.12	5.65	14.42	0.76
IBM UltraStar	8.39	2.10	181.90	18	0.87	2.19	9.68	6.29	16.95	2.33
IBM 9ZX	6.06	1.40	224.69	12	0.79	1.91	16.15	10.06	16.17	1.37
Quantum Fireball	11.40	1.25	356.35	2	2.19	2.89	1.98	1.98	?	?
IBM-DTTA	11.04	1.02	330.01	6	1.93	3.81	1.95	1.95	16.41	3.24
HP 97560 (sim)	14.78	3.12	70.99	19	1.67	3.75	2.10	2.10	21.87	4.38
DEC RZ26 (sim)	10.82	3.09	60.33	14	1.70	5.13	2.28	2.28	17.95	2.12

Table 3: **Extracted Values.** *The table lists extracted parameters from each disk drive, including ranges for the bandwidth and seek curves. Note that the values from the seek curve have been adjusted by the minimum time to media so as to reflect actual seek characteristics.*

I/O instead of DMA, which is common for IDE drives. Although this improves overhead, we will see the cost of this below when discussing the achieved bandwidth from IDE.

5.1.3 Simulated Drives

Figures 20 and 21 show the results on the Dartmouth disk simulator [8] and *diskSim* [4]. These experiments verify that the SKIPPY technique matches the way disks are expected to work by the two best known disk simulators. The values extracted from both measurements match the simulator disk specifications. The simulated results are noticeably cleaner than the measurement results. When comparing the two simulation results, we see that the *diskSim* result shows downward spikes before the sawtooth transition, much like the real disks. The Dartmouth result does not, suggesting that the newer *diskSim* simulates a drive more closely than the older Dartmouth simulator.

5.2 ZONED

Each drive's SKIPPY result is accompanied by its ZONED result. We can make several general observations from the ZONED results. First, the older simulated drives and the newer IDE drives show only one recording zone. For the IDE drives, this implies that drive manufacturers are willing to sacrifice performance for simplicity. Second, the achieved bandwidth from the IDE drives is quite low; this may reflect the use of programmed I/O instead of DMA. Third, among the SCSI drives, the Seagate drives are noticeably more finely zoned than the IBM and Micropolis drives. Finally, for the disks with multiple zones, the overall difference between outer-track and inner-track bandwidth ranges from 50% up to 80%.

The most recent, comprehensive, discussion of disk drive zoning behavior was in [10], which observed that the relationship between transfer rate and disk position was far better described with a linear function than a single value. After examining our zone results, we see that the curve is actually closer to parabolic than linear. A quadratic function, of the form $a \cdot x^2 + b$ is a much better fit for the zone graph than the linear function. In fact, by fitting both linear and quadratic functions to the data (using standard linear regression techniques), we learned that the quadratic function has between a factor of 2 to a factor of 10 better error than the simple linear fit. The linear fit explored in [10] had an extra advantage in that it only required the highest and lowest bandwidth values from the drive. However, we found that a quadratic fit using only these two values was still better (by a factor of 10 to 20!) than a linear fit using the same two values. In fact, in all cases but one, the quadratic fit with two values was better than a linear fit using all values, by a factor of 2 to 10.

Thus, if a model must be employed, we recommend usage of a quadratic fit. It is as simple to construct as the linear model (requiring only two data points) and matches the profiles better than the linear fit. For disks with only a few zones, the exact step function should be utilized; the *diskSim* simulator makes use of such an exact characterization.

5.3 SEEKER

The figures and table also show the seek latency from the start of the drive to other areas as a function of seek distance in sectors. For seeks over one tenth of the disk, the seek latency appears to increase linearly with sector distance (much like the seek latency increases with larger numbers of cylinders). Close examination of the data reveals that, for seeks reaching the innermost zones, the latency increase is higher than linear. This is most observable in the IBM 9ZX seek result. The seek time increases more rapidly because the Sectors/Track ratio decreases more rapidly in this area, requiring more arm movement for the same sector distance.

6 Conclusions

This paper presents three disk benchmarks, SKIPPY, ZONED, and SEEKER, that extract a range of parameters from modern disk drives. SKIPPY, in particular, illustrates a novel approach for measuring disks via linearly increasing stride patterns. This technique, and its extensions, can be used to filter out the rotational effect in all kinds of disk measurements. To our knowledge, we present the first benchmark that *utilizes* the disk's rotational mechanism in characterizing the disk, rather than trying to defeat it.

The benchmarks are run upon five SCSI drives, two IDE drives, and two disk simulators, revealing numerous results about modern disk drives. We find that the the minimum time to access drive media can vary widely, even between drives of the same generation. The Seagate drives show excellent switching time characteristics, whereas the IBM drives have better overall bandwidth. The results also show other similarities between drives made by the same manufacturer, such as the odd number of recording surfaces present in both Seagate drives. The SCSI drives, although older, show far better performance than the IDE drives both in switching times and bandwidth, whereas the overhead of IDE reads and writes is lower.

The improvements in linear and areal density are reflected in the sectors/track and number of recording surfaces of the measured drives; more modern drives have a higher sectors/track ratio than the older drives. The number of recording surfaces is concurrently decreasing.

As rotational latency and seek times decrease, head-switch and cylinder-switch times may become more important. For example, whereas the Hawk switch times are roughly 10% to 20% of the rotation time, the corresponding percentages for the IBM 9ZX are 13% and 32%. Simple models of disk behavior, which characterize disks only with seek and rotation time, will no longer be appropriate.

The full paper will provide more details of the SKIPPY read and write benchmarks. Our future work includes exploration of the *backwards read* variant that retains the benefits of the read benchmark while avoiding interaction with the read-ahead mechanism.

The benchmarks, extraction tool, and all measured data will be made available at a public website. Our hope is that others will run the benchmark and contribute their data to an active archive of disk characteristics.

References

[1] Remzi H. Arpaci, David E. Culler, Arvind Krishnamurthy, Steve G. Steinberg, and Kathy Yelick. Empirical Evaluation of the CRAY-T3D: A Compiler Perspective. In *The 22nd Annual International Symposium on Computer Architecture (ISCA-22)*, pages 320–331, June 1995.

[2] Peter M. Chen and David A. Patterson. A New Approach to I/O Performance Evaluation–Self-Scaling I/O Benchmarks, Predicted I/O Performance. In *Proceedings of the 1993 ACM SIGMETRICS Conference*, pages 1–12, May 1993.

[3] David E. Culler, Lok Tin Liu, Richard P. Martin, and Chad Owen Yoshikawa. LogP Performance Assessment of Fast Network Interfaces. *IEEE Micro*, 2/1996.

[4] Gregory R. Ganger, Bruce L. Worthington, and Yale N. Patt. The DiskSim Simulation Environment Version 1.0 Reference Manual. Technical Report CSE-TR-358-98, Department of Electrical Engineering and Computer Science, University of Michigan, February 1998.

[5] Cristina Hristea, Daniel Lenoski, and John Keen. Measuring Memory Hierarchy Performance of Cache-Coherent Multiprocessors Using Micro Benchmarks. In *Supercomputing '97*, San Jose, CA, November 1997.

[6] IBM. Ultrastar 2XP Hardware/Functional Specification: 4.55 GB and 8.22 GB Models, 7200 RPM, Version 5.03. Document Number AS05-0087-45 – IBM Storage Products Division, June 1996.

[7] IBM. IBM Disk Drive Specifications. http://www.storage.ibm.com/, 1999.

[8] David Kotz, Song Bac Toh, and Sriram Radhakrishnan. A Detailed Simulation Model of the HP 97560 Disk Drive. Technical Report PCS-TR94-220, Department of Computer Science, Dartmouth College, July 1994.

[9] Larry McVoy and Carl Staelin. lmbench: Portable Tools for Performance Analysis. In *Proceedings of the 1996 Winter USENIX*, January 1996.

[10] Rodney Van Meter. Observing the Effects of Multi-Zone Disks. In *Proceedings of the 1997 USENIX Conference*, January 1997.

[11] Micropolis. Micropolis Disk Drive Specifications. http://www.procom.com/homepage/tech/, 1999.

[12] Quantum. Quantum Disk Drive Specifications. http://www.quantum.com/, 1999.

[13] Chris Ruemmler and John Wilkes. An Introduction to Disk Drive Modeling. *IEEE Computer*, 27(3):17–28, March 1994.

[14] Rafael H. Saavedra, R. Stockton Gaines, and Michael J. Carlton. Characterizing the Performance Space of Shared-Memory Machines Using Micro-Benchmarks. In *Hot Interconnects '94*, San Jose, CA, August 1994.

[15] Rafael H. Saavedra-Barrera. *CPU Performance Evaluation and Execution Time Prediction Using Narrow Spectrum Benchmarking*. PhD thesis, U.C. Berkeley, Computer Science Division, February 1992.

[16] W. David Schwarderer and Andrew W. Wilson. *Understanding I/O Subsystems*. Adaptec Press, first edition, January 1996.

[17] Seagate. Seagate Disk Drive Specifications. http://www.seagate.com/, 1999.

[18] Carl Staelin and Larry McVoy. mhz: Anatomy of a micro-benchmark. In *Proceedings of the USENIX 1998 Annual Technical Conference*, pages 155–166, Berkeley, USA, June 15–19 1998. USENIX Association.

[19] Bruce L. Worthington, Greg R. Ganger, Yale N. Patt, and John Wilkes. On-Line Extraction of SCSI Disk Drive Parameters. In *Proceedings of the 1995 ACM SIGMETRICS Conference*, pages 146–156, May 1995.