Revisiting HDD Rules of Thumb: 1/3 Is Not (Quite) the Average Seek Distance

Serkay Ölmez
MIT Lincoln Laboratory
Lexington, MA, US
serkay.olmez@ll.mit.edu

Yifan Dai
University of Wisconsin
Madison, WI, US
yifann@cs.wisc.edu

John Bent
Los Alamos National Laboratory
Los Alamos, NM, US
johnbent@lanl.gov

Remzi Arpaci-Dusseau
University of Wisconsin
Madison, WI, US
remzi@cs.wisc.edu

***Abstract*—Humans love rules of thumb: memory shortcuts enabling simple approximations to work in functionally equivalent manners to more precise, but more complex, realities. In this paper, we re-examine two classic rules of thumb in computer systems. First, that the average seek distance of a random hard drive access is 1/3rd of the maximum seek distance. Second, that the total latencies to access data on a hard drive is the sum of the seek, rotation, and transfer latencies. We first explain the derivation and intuition behind these rules of thumb. We then introduce rigorous mathematical models for seeks, rotations, and total access times to precisely compute their values. We show that the mean value of the seek time is $\sim 1/2$ of its maximum value. Furthermore, we include detailed studies of tail latencies in addition to mean values as tail latencies are of increasing importance in data centers.**

We verify our mathematical models with actual experimental measurement data and Monte Carlo simulations and study the precise inaccuracies of the rules of thumb. Using our more accurate models, we introduce new rules of thumb which are more accurate than the previous ones. We add a discussion of emerging multi-actuator drives (drives containing multiple independent seek units per surface) to provide insight into their latency metrics. Finally, we conclude with hypothetical real world problems which can result from using these inaccurate rules of thumb.

***Index Terms*—latency, HDD, performance**

I. Introduction

Despite significant advances in persistent storage technologies – including Flash-based solid state drives (SSDs) [1] and, more recently, various forms of persistent memories [2]– hard disk drives (HDDs) remain the dominant form of storage medium today. HDDs account for the overwhelming majority of bytes stored in cloud data centers [3], [4], as well as enterprise data centers and edge systems: "By the end of 2025, over 80% of the enterprise bytes shipped into the core and edge will continue to be HDD bytes when compared to SSDs and other non-volatile memory technologies. [5]"

The reasons for this continued dominance are manifold, but are largely driven by cost per byte; until other media can compete in this metric, most data will continue to be stored in HDDs. As such, designers of modern systems, whether backend distributed data centers, or near-edge server caches, must consider hard drives when planning future installations and upgrades to existing systems.

To create and configure such systems, designers often rely on a wide range of analytical techniques to determine important factors such as the total number of storage components to purchase, as well as the ratios between the different components in the system such as CPU, DRAM, SSD, network switches, and HDDs. Mistakes in these absolute values and ratios can result in either costly failures to deliver the requirements (performance, reliability, etc) or costly overprovisioning of the system.

The first of these analyses is often a rough "back of the envelope" set of calculations. As Bentley writes: "Early in the design of a system, rapid calculations can steer the designer away from dangerous waters into safe passages. [6]" As Dean more recently emphasized, an "Important skill [is the] ability to estimate performance of a system design – without actually having to build it! [7]"

To make such estimations, designers must employ a wide range of "well known" base numbers and *rules of thumb*. For example, Dean suggests numbers "everyone should know" to include cache and main-memory reference times, branch misprediction costs, time to send packets, and disk seek and access times, among other important metrics [7].

A. Our contribution

In this paper, we first examine classic rules of thumb that have pervaded the disk industry for decades, and show that, in some cases, the rules are inaccurate. We use analytical methods to derive closed form expressions for HDD latencies for drives. We validate the model with simulation techniques and experimental measurements done on modern, latest generation HDDs. We use our rigorous and accurate model to update these classic rules to be more precise. Specifically, we first show that the classic rule of thumb *"Average disk seek distance is one-third of full seek distance for random seeks"* [8] is (slightly) incorrect for modern drives. Furthermore, we develop a rigorous mathematical model to quantify another rule of thumb: "*A complete picture of I/O time: first a seek, then waiting for the rotational delay, and finally the transfer* [9]."

We then derive a broader set of drive rules of thumb, which can be useful in different types of calculations beyond simple average costs. For example, we show that the median seek distance is roughly one-fourth the full one, and that the 95 percentile seek distance is roughly three-quarters the full one. The latter numbers are especially critical in understanding increasingly important "tail latency" costs in systems [4],

[10]–[13]. We also show that in an extremely physically large disk, the average seek distance between random target sectors approaches $\frac{4}{15}$ of the maximum seek distance. More importantly, we convert the seek distance to the seek time and compute latencies associated with the radial seeks.

Finally, we show how to combine radial and rotational latencies in a mathematically meaningful (i.e., correct) manner. For example, when computing average positioning time, it suffices to simply sum average seek and average rotational latencies. However, when computing quantile positioning times (e.g., 95 percentile), simple addition of the corresponding quantile seek and quantile rotational latencies is inaccurate, often by a noticeable amount (i.e., up to 13% for current HDDs). As we develop the more accurate model to compute the statistics of the latencies, we extend the rules of thumbs as follows:

1) The median value is $\sim 1/4$ of the full seek distance,
2) The mean value is $\sim 1/3$ of the full seek distance,
3) 75% quantile is $\sim 1/2$ of the full seek distance,
4) 95% quantile is $\sim 3/4$ of the full seek distance.
5) The mean value of the seek time is $\sim 1/2$ of its maximum value.
6) The mean value of total latency is $\sim 1/2$ of the sum of the maximum seek time and rotation period.
7) The computation of the tail latencies, i.e., quantiles requires the complete model, see Eq. (16).

Item 6, in particular, is worth emphasizing, and it follows from the nonlinear relation between the seek distance and the seek time, which is not captured accurately with linearized models.

With our updated rules, designers can more accurately analyze new system designs and configuration changes, thus quickly understanding more accurate trade-offs in proposed systems. We thus show how to properly utilize such numbers, ensuring accurate calculations.

Note that the mathematical model we present is for conventional magnetic recording (CMR), and it is not intended to fully capture all characteristics of an HDD. For example, although we briefly touch on the topic, the model does not take larger queue depths into consideration and queue depth has been shown to be an important performance metric [14]. Instead, this paper presents an accurate model for the base primitive operations of a hard-drive: seek and rotation times. We believe this allows valuable, and actionable, analysis as we will show. We also believe this can (and should) serve as the foundation for more detailed models and HDD simulations .

B. Outline

The rest of this paper is organized as follows. In Section III, we provide a reminder for the textbook derivation of the 1/3rd rule and explain the reason for its inaccuracy. We show how it can be made more precise in Section IV where we derive the full statistics of the seek time including their tail latencies. In Section V, we discuss how to add the rotational latency to calculate the distribution for the total latency. In Section VI, we show that the results from the mathematical model match very well with the simulation and test results. We then apply this new calculation method to emerging HDDs with two heads per surface in Section VII. We briefly discuss the effects of command queuing in Section VIII. In Section IX, we discuss the potential costly implications of using inaccurate rules of thumb to design and provision data center storage systems. Finally, we offer parting thoughts in Section X.

II. Related work

HDDs have been studied in detail for decades with actual measurements, simulations, and theoretical models providing designers with valuable insights in configuring HDD based systems. Coffman et al. [8] establish the $1/3$ rule based on an approximated uniform track probability density as well as approximated constant (linear) head speed. Ruemmler& Wilkes [15] provide an excellent review of latency components using simulations and empirical data. Lumb et al. [16], [17] show that background operations can be intermixed with user IO operations without incurring extra delays. These studies further justify a careful modeling of the radial and rotational components enabling opportunities to serve background operations instead of staying idle as the next target sector rotates under the head. Linearized models have been further developed to study average access times [18]. Operating at fixed rotational speeds, HDDs implement zoned constant angular velocity (ZCAV) technique. The transfer times for ZCAV drives have been studied by calculating the amount of data at each zone (radii) and the corresponding time required to read/write sequentially [19]. Wilhelm [20] provides a model of a full system with several HDDs attached to a single I/O channel. Worthington et al. [21] provides a higher level study of different scheduling algorithms in various workload scenarios. Ng [22] provides a thorough review of existing models and systems. All three extrapolate from a low-level HDD model to make higher-level studies and observations.

Nomenclature

r_i, r_o	Inner and outer radii of disk
r_1, r_2	Radii of two randomly selected tracks
κ	The ratio of the outer radius to the inner radius
$\overline{s}$	Average radial separation between two random tracks
$f_R(r)$	Probability density of finding a sector at radius r
$f_S(s)$	Probability density of radial separation of two tracks
ω	The angular frequency of disk rotation
T_0	The time for one full disk rotation
T_R	The maximum (radial) seek time
ϕ	The angle of a random sector on a track
$f_T(t)$	Probability density of the total latency

III. Rules of thumb for seek and total access

An HDD internally has multiple spinning *disks* (often also alternatively referred to as platters). *Sectors* on each disk are read and written by read-write *heads*. The heads are positioned by a mechanical *actuator* (also referred to as an arm) to *seek* to concentric circles of sectors referred to as disk *tracks*. It takes a finite amount of time to move the actuator from one track to another, which is defined as the *seek time* [9]. Once the head is correctly positioned over the target track, some

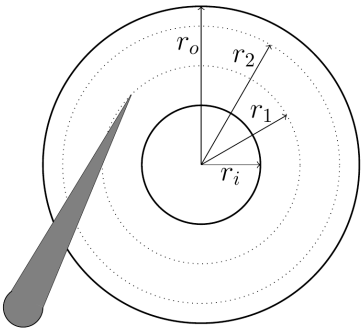

Fig. 1: A disk with innermost data track radius r_i and outermost data track radius r_o and two sample tracks located at radii r_1 and r_2. The actuator (and the arm), shown shaded in gray color, moves the read-write head to access tracks at different radii. Not shown: internally, HDD's have multiple of these disks stacked vertically, each disk has a top and a bottom surface, every surface is accessed by its own arm, and all arms are controlled by the single, shared actuator.

additional *rotational latency* may be incurred before the target sector rotates under the head. Finally, the last component of a disk access is the time to *transfer* the data on the target sector(s). Therefore, the classic rule of thumb for the latency of a disk access is the sum of the seek latency, the rotational latency, and the transfer latency [9].

In this paper, we focus on the seek and rotational latencies as the transfer latencies are fixed constants that are irrelevant for the analysis in this work. Please refer to Figure 1 which introduces some of the variables used throughout the remainder of this paper. Additionally, Table I shows typical values of current enterprise HDDs [23] which we will use for all calculations done throughout this paper.

Radius of innermost track (r_i)	0.73"
Radius of outermost track (r_o)	1.83"
A typical max seek time	15 ms
Rotational speed	7200 RPM

TABLE I: Typical values of key parameters for current enterprise HDDs. The maximum seek time is based on the test data, see Figure 9.

Given the task of locating many random sectors on a surface, the average time to position the head onto target tracks is the average seek time. This time is a function of the radial separation between the current track and the target track. The radial separation between two tracks at radii r_1 and r_2 where $r_i \leq r_{1,2} \leq r_o$ is $|r_2 - r_1| \equiv s$. Averaging over every possible position of r_1 and r_2 yields the average radial distance. Assuming that each r is equally likely to be a target track, the density will be $\frac{1}{r_o - r_i}$, and the average radial distance can be calculated as:

$$\overline{s} = \frac{1}{(r_o - r_i)^2} \int_{r_i}^{r_o} \int_{r_i}^{r_o} |r_2 - r_1| dr_2 dr_1 = \frac{r_o - r_i}{3}. \quad (1)$$

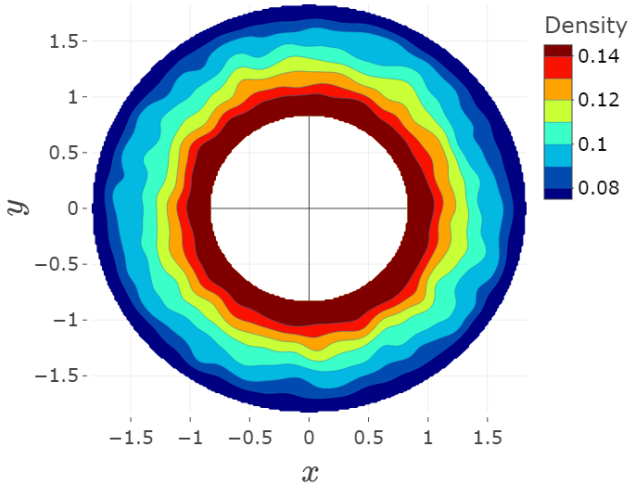

Fig. 2: The sector density when we uniformly sample radius values in the range $[r_i = 0.73", r_o = 1.83"]$, and the angle values in the range $[0, 2\pi]$. The steep gradient proves that the uniform track probability density results in a nonuniform sector density on the surface. This is in conflict with real sector density on a disk surface which is fairly uniform.

Taking the ratio of the mean distance to the maximum distance, $r_o - r_i$, yields the infamous result:

$$\frac{\overline{s}}{r_o - r_i} = \frac{1}{3}. \quad (2)$$

IV. A MORE PRECISE MODEL OF SEEK TIME

In the previous section, it was assumed that *"... each r is equally likely to be a target track"*, which implied that the density function is $\frac{1}{r_o - r_i}$. However, a closer inspection shows that this assumption is in conflict with real HDDs in which sectors are uniformly distributed across each disk surface [18]. To illustrate this conflict, we can do a simulation that selects tracks with radii from 0.73" to 1.83" from a uniform distribution and spreads the sectors uniformly around the tracks. The resulting sector density across the disk surface is shown in Figure 2, and it is clearly not uniform. Since it leads to a nonuniform sector distribution, the assumption of *"... each r is equally likely to happen"* used in the seek time rule of thumb is incorrect. This reflects the perhaps more intuitive observation that tracks at the outer diameter of each surface are physically longer than inner diameter tracks. Therefore, outer tracks typically contain more sectors, and as a result, they are more likely to be the target track for any random sector.

A. The correct probability density

The area of a small set of tracks from r to $r + \delta r$, with $\delta r \ll r$, is $2\pi r\, \delta r$, and it grows $\propto r$. Therefore, the probability of a target sector being in this set will similarly grow $\propto r$. In fact, with proper normalization, it will be

$$f_R(r) = \frac{2r}{r_o^2 - r_i^2}, \quad (3)$$

so that it yields 1 when integrated from r_i to r_o. The same density function can be derived more rigorously by starting from two uniformly distributed variables X and Y in the domain $r_i^2 \leq X^2 + Y^2 \leq r_o^2$. One can define polar coordinates in the standard way: $R = (X^2 + Y^2)^{1/2}$ and $\Phi = \text{sign}(Y) \arccos\left(\frac{X}{(X^2+Y^2)^{1/2}}\right)$, and show that the probability density for R is identical to the result above.

The track density as expressed in Eq. (3) should be intuitive given the previous explanation of HDDs: uniformly distributed data on the disk surface means that more data will be on the outer radii of each surface. Hence, the average seek time calculated using the proper density is:

$$\begin{aligned} \overline{s} &= \int_{r_i}^{r_0} \int_{r_i}^{r_0} f_R(r_1) f_R(r_2) |r_2 - r_1| dr_2 dr_1 \\ &= \frac{4}{15}(r_o - r_i)\left(1 + \frac{\kappa}{(1+\kappa)^2}\right), \end{aligned} \quad (4)$$

where $\kappa \equiv \frac{r_o}{r_i}$ is the ratio of the outermost radius to the innermost one. The ratio of the mean distance to the maximum distance becomes

$$\frac{\overline{s}}{r_o - r_i} = \frac{4}{15}\left(1 + \frac{\kappa}{(1+\kappa)^2}\right). \quad (5)$$

Figure 3 shows the average seek distance (as a fraction of the maximum seek distance) as a function of κ (the ratio between the diameters of the outermost and innermost tracks). Several various interesting values are annotated:

1) When κ is 1 (as it would be on a hypothetical disk with uniform track accesses), the average seek distance matches the rule of thumb of $\frac{1}{3}$.
2) The marked κ value of 2.51 corresponds to industry standard disk sizes of outer radius $r_o = 1.83$" and inner radius $r_i = 0.73$". Note that these values of r_o and r_i yield a max seek distance of 1.1", which is the value we will use later throughout the paper. This results in a more precise calculation of an average seek distance being 0.321 of the maximum seek distance.
3) The asymptote of $\frac{4}{15}$ would be the average seek distance (normalized to the maximum distance) on an infinitely large disk and/or very small inner radius.

For current typical disk geometries, the 1/3rd rule of thumb is accurate up to a $\sim$ 4% error, so it remains a useful approximation. However, if the HDD industry moves towards larger disks, the inaccuracy will increase. And depending on the precision required, the 1/3rd rule of thumb may not be accurate enough for practical calculations. It is also important to note that we assumed the data is uniformly distributed on the disk, and this assumption might be invalid in certain cases. For example, due to the skew angle of the read-write head, the areal density on a disk may vary radially [24]. In such cases, the density in Eq. (3) needs to be replaced with the measured areal density on the disk.

B. The complete statistics of seek distance

In real life applications, in addition to the average value, it is important to understand *tail latencies* of the seek times [25],

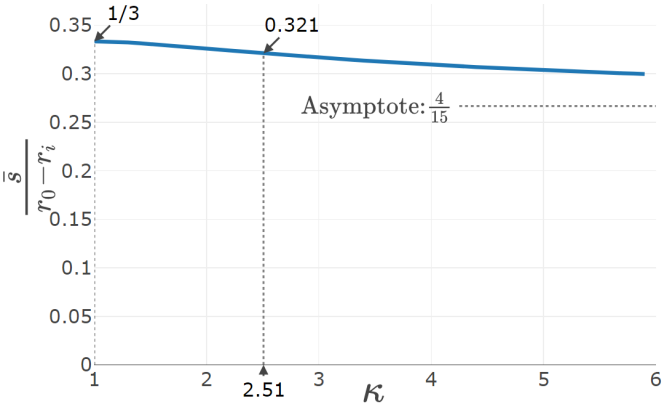

Fig. 3: This graph shows $\frac{\overline{s}}{r_o - r_i}$ (the average seek distance normalized to the max seek distance) as a function of κ (the ratio between the diameters of the outermost and innermost tracks).

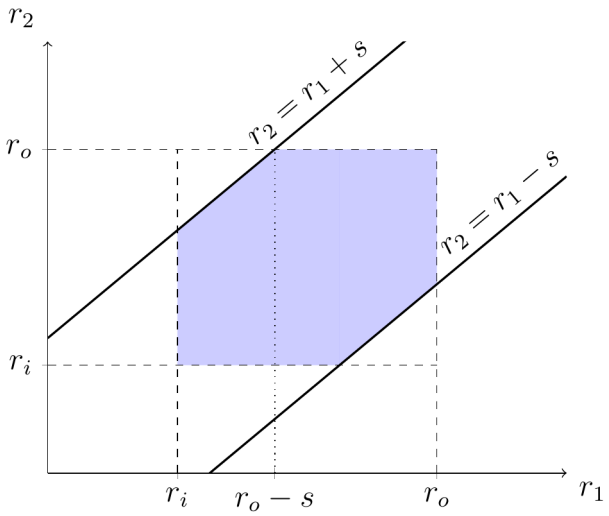

Fig. 4: The domain of interest for integration in Eq. (6). In the shaded area $|r_2 - r_1| < s$ is satisfied.

[26]. Although it was studied in the literature [18], [21], the explicit form of the seek time density function has not been given. In order to derive the exact form, the radial separation $S \equiv |R_1 - R_2|$ can be defined as a random variable with the following cumulative distribution:

$$F_S(s) = P(|r_1 - r_2| < s) = \iint_{|r_2 - r_1| < s} f_R(r_1) f_R(r_2) dr_1 dr_2. \quad (6)$$

The domain of the integration is the shaded area in Fig. 4. The cumulative probability function of the radial separation can be written as

$$\begin{aligned} F_S(s) &= \iint_{\text{shaded}} f_R(r_1) f_R(r_2) dr_1 dr_2 \\ &= 4\frac{2(r_o^3 - r_i^3)s - \frac{3}{2}(r_o^2 + r_i^2)s^2 + \frac{s^4}{4}}{3\left(r_o^2 - r_i^2\right)^2}, \end{aligned} \quad (7)$$

from which we can get the probability density by differentiating with respect to s:

$$f_S(s) = \frac{\partial F_S(s)}{\partial s} = 4\frac{2(r_o^3 - r_i^3) - 3(r_o^2 + r_i^2)s + s^3}{3\left(r_o^2 - r_i^2\right)^2}. \quad (8)$$

$f_S(s)$ completely defines the statistics of the radial distance from the head position to the target sector. It can be verified that the result in Eq. (5) can be reproduced by computing $\int_0^{r_o - r_i} s f_S(s) ds$. We can also use Eq. (8) to compute various important statistical parameters as summarized in Table II.

	Mean	S. D.	Med.	Q75	Q95	Max
abs.	0.35	0.25	0.31	0.53	0.84	1.1
norm.	0.32	0.23	0.28	0.48	0.76	1.00

TABLE II: Various key statistics for the radial seek distance. The first row is in the units of inches with the full seek distance of 1.1". The second row is normalized to this value.

Based on Table II, the rules of thumb for radial seek distance can be extended as follows:

1) The median value is $\sim 1/4$ of the full seek distance,
2) The mean value is $\sim 1/3$ of the full seek distance,
3) 75% quantile is $\sim 1/2$ of the full seek distance,
4) 95% quantile is $\sim 3/4$ of the full seek distance.

C. The seek time as a function of distance

The relation between the seek distance and the seek time is nontrivial due to the acceleration, cruising, deceleration, and the settling times of the actuator [15]. The actuator will be accelerating from its initial track until it reaches a maximum speed. It will start decelerating around the mid distance to the destination track. Furthermore there will be certain amount of settling time as an overhead. In order to derive analytical models, this relation is typically approximated as a linear one [15], [18]. However, the nonlinear part of the relation is associated with the short distance seeks, and as we have shown in Figure 8, the majority of seeks traverse short distances. Therefore, we model seek time including both the linear and non-linear regions. We will parameterize the acceleration duration as t_a, and estimate the functional form of the relation between the seek time and seek distance as

$$\mathcal{S}(t) = \begin{cases} 0, & \text{for } t < \beta \\ \frac{1}{2}\alpha(t-\beta)^2, & \text{for } \beta \le t \le t_a \\ v_a(t - t_a) + s_a, & \text{for } t > t_a, \end{cases} \quad (9)$$

where α is the acceleration factor and β is the overhead time. The free parameters in Eq. (9) are α, β, and t_a. Although the values of these parameters are dependent on the drive model, we observed only small variations for drives from the same model. The cruising speed, v_a, and the transition position, s_a, are fixed by imposing the continuity of the function and its derivative at $t = t_a$, which requires:

$$v_a = \alpha(t_a - \beta), \quad \text{and} \quad s_a = \frac{1}{2}\alpha(t_a - \beta)^2. \quad (10)$$

In Section VI, we will show that the proposed relation between seek distance and seek time fits well with the experimental data. We can now use the probability density of s is given in Eq. (8) to compute the probability density of the seek time as

$$f_{\text{seek}}(t) = \frac{d\mathcal{S}(t)}{dt} f_S\left(\mathcal{S}(t)\right) = \begin{cases} 0, & \text{for } t < \beta \\ \alpha(t-\beta) f_S\left(\frac{1}{2}\alpha(t-\beta)^2\right), & \text{for } \beta \le t \le t_a \\ v_a f_S\left((t-t_a)v_a + s_a\right), & \text{for } t > t_a, \end{cases} \quad (11)$$

where f_S, v_a and s_a are defined in Eqs. (8) and (10), respectively. For fit values of $\alpha = 0.014\frac{\text{inch}}{\text{ms}^2}$, $\beta = 1.18$ms, $t_a = 8.8$ms, see Section VI, we compute the statistics of the seek time and tabulate the results in Table III.

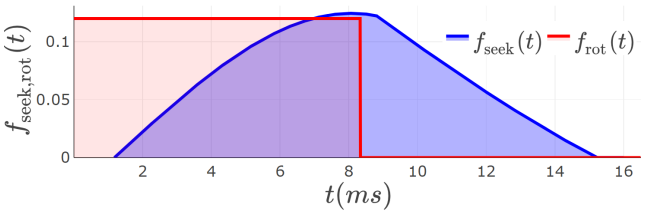

Fig. 5: The density functions for the seek ($f_{\text{seek}}(t)$) and rotational ($f_{\text{rot}}(t)$) latencies.

	Mean	S. D.	Q50	Q75	Q95	Max
abs	7.8	2.9	7.8	9.91	12.8	15.2
norm.	0.51	0.19	0.51	0.65	0.84	1.0

TABLE III: Various key statistics for the seek time. The first row is in the units of ms with the max seek time of 15.2ms. The second row is normalized to this value.

From Table III, we see that average seek time is close to $1/2$ of the maximum seek time. This is worth to emphasize because it is a common misconception to state that *since the average value of the seek distance is about* $1/3$ *of the maximum, the average value of the seek time is also* $1/3$ *of the maximum seek time*. In reality, due to the nonlinear mapping between the radial distance and the corresponding seek time, the average seek time is closer to the half of the maximum value. We can add this observation to the list of rules of thumb we started in Section IV-B:

5) The mean value of the seek time is $\sim 1/2$ of its maximum value.

V. Adding the Rotational Latency

In addition to the seek time, HDDs also need to wait for the target sector to rotate under the head and therefore both the seek time and the rotational time will contribute to the total time to access a target sector. As a reminder, all values used to compute latencies were shown earlier in Table I. Specifically, we consider 7200 RPM drives which results in $T_0 = 8.33$ms. A set of typical seek time ($f_{\text{seek}}(t)$) and rotational ($f_{\text{rot}}(t)$) latency densities are illustrated in Figure 5.

The simple rule of thumb to compute a total access latency is to combine the seek and rotation and transfer latencies. Ignoring the transfer latency (as it is a fixed constant irrelevant to our analysis) allows us to simplify and say that the

total *positioning* latency is the sum of the seek and rotation latencies. The key point we are making here is that this rule of thumb does work correctly to compute the mean positioning latency but *does not* work correctly to compute the quantiles. This is illustrated in Table IV where various key statistical parameters are tabulated.

	Mean	S. D.	Q50	Q75	Q95	Max
seek	7.8	2.9	7.8	9.91	12.8	15.2
rot.	4.2	2.4	4.2	6.3	7.9	8.3
total.	12.0	3.8	*11.9*	*16.2*	*20.7*	23.6

TABLE IV: Key statistics for the total latency in milliseconds with a 7200 rpm drive ($T_0 = 8.33$ms). The values are calculated using the distributions shown in Figure 5. The last row of the table is the rule of thumb estimate as computed by adding the first two rows. The values which are not correctly estimated by the rule of thumb are formatted as *X*.

Another subtle detail we need to address is that the seek of the head and the rotation of the disk happen at the same time, i.e., they are happening concurrently rather than sequentially. We also note that there are two typical ways to think about the rotational latency: one, the *total* rotational latency required at the start of the access and, two, the *remaining* rotational latency after the seek has completed. Throughout this paper, we always use the former. For some accesses, the seek time is shorter than the (total) rotational latency. Therefore, the head can be positioned to the target track sooner than the target sector rotates under the head. In such a case, the total latency will be equal to the rotational latency, and the seek time will be irrelevant. In fact, modern HDDs recognize, and optimize, this situation by reducing the seek speed such that the head arrives "just-in-time" (i.e., the seek finishes right as the target sector rotates into position) thereby reducing energy costs without reducing performance [27]. However, there will also be accesses in which the head *cannot* reach the target track before the sector rotates past, and it will have to wait for the next rotation. We will show that the average amount of time lost for these accesses is precisely equal to the average seek time.

In order to derive the full statistics of the total positioning latency, we define the rotational speed of the disk as ω, the radial separation from the head's current track to the target track as s, and the rotational separation of the initial head position and the target sector position as ϕ. If the seek time $t(s)$, Eq. (9), is smaller than the rotational latency (ϕ/ω) then the total positioning latency is simply the rotational latency. On the other hand, if the head cannot be rotated to the target sector in time (i.e., $t(s) > \frac{\phi}{\omega}$), it needs to wait an additional full rotation to access the target sector. In order to simplify the notation, let us define the condition described above as a piece-wise function:

$$q(s,\phi) \equiv \begin{cases} 0, & s < \mathcal{S}(t) \\ 1 & \text{otherwise} \end{cases}, \tag{12}$$

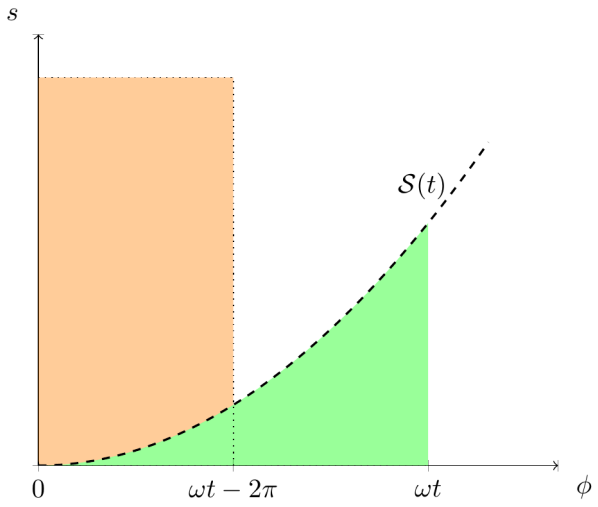

Fig. 6: The domain of interest for integration in Eq. (14). In the shaded areas $\frac{\phi}{\omega} + q(s,\phi)\frac{2\pi}{\omega} \leq t$ is satisfied.

and write the total latency, t_{total}, as follows:

$$t_{\text{total}}(s,\phi) = \frac{\phi}{\omega} + q(s,\phi)T_0, \tag{13}$$

where $T_0 = \frac{2\pi}{\omega}$ is the disk rotation period. We can treat t_{total} as a random variable and compute its cumulative distribution function as

$$F_T(t) = \iint\limits_{\frac{\phi}{\omega}+q(s,\phi)T_0 \leq t} f_\Phi(\phi) f_S(s) d\phi ds. \tag{14}$$

The domain for which $\frac{\phi}{\omega} + q(s,\phi)T_0 \leq t$ is satisfied in the shaded areas in Fig. 6. Therefore the integral in Eq. (14) can be evaluated as

$$F_T(t) = \begin{cases} \int_0^{\omega t}\int_0^{\mathcal{S}(t)} f_\Phi(\phi) f_S(s) d\phi ds, & t \leq T_0 \\ c_0 + \int_0^{\omega t - 2\pi}\int_{\mathcal{S}(t)}^{r_o - r_i} f_\Phi(\phi) f_S(s) d\phi ds, & t > T_0, \end{cases} \tag{15}$$

where $c_0 = F_S\left(\mathcal{S}(T_0)\right)$ ensures the continuity of the distribution. We can compute the density function by taking the derivative of Eq. (15) and combine the resulting terms as follows:

$$\begin{aligned} f_T(t) &= \frac{dF_T(t)}{dt} = \frac{1}{T_0}\int_{\mathcal{S}(t-T_0)}^{\mathcal{S}(t)} f_S(s) ds \\ &= \frac{1}{T_0}\left[F_S\left(\mathcal{S}(t)\right) - F_S\left(\mathcal{S}(t-T_0)\right)\right], \end{aligned} \tag{16}$$

where F_S and $\mathcal{S}$ are defined in Eqs. (7) and (9), respectively. Equation (9) defines the probability density of the total latency, and key statistical parameters are calculated in Table V.

Type	Mean	S. D.	Q50	Q75	Q95	Max
seek	7.8	2.9	7.8	9.91	12.8	15.2
rot.	4.2	2.4	4.2	6.3	7.9	8.3
total.	12.0	3.8	11.9	14.6	18.3	23.6

TABLE V: Precisely calculated key statistics for the total latency in milliseconds with a 7200 rpm drive ($T_0 = 8.33$ms).

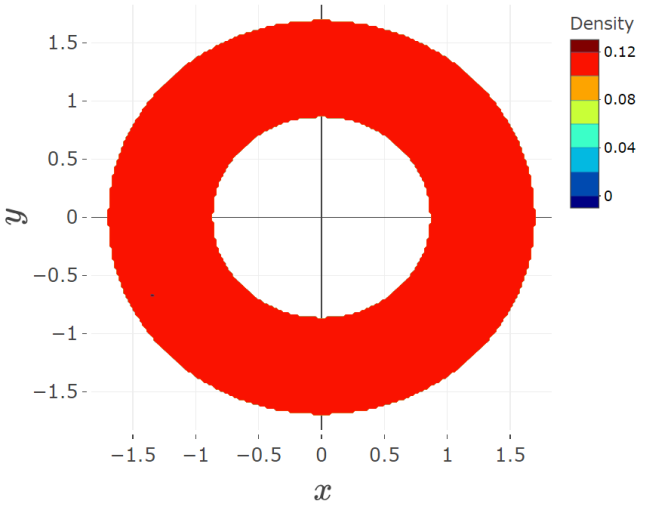

Fig. 7: The simulated probability density of points constructed from uniformly selected x and y values. The disk is uniformly covered. We use industry standard values of $r_o = 1.83$" and inner radius $r_i = 0.73$" in the simulation.

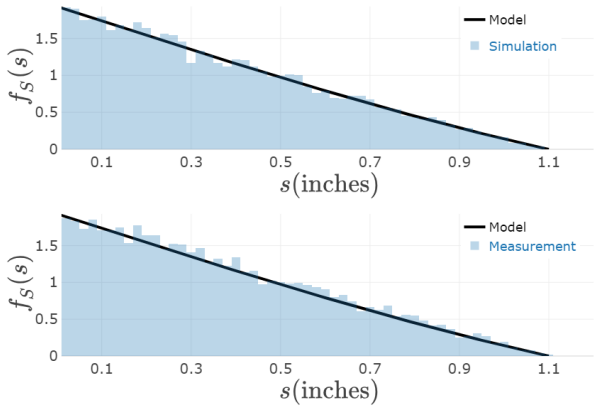

Fig. 8: The distribution of the seek distance for the simulated and measured data. The shaded bars show the simulated and measured distributions and the lines are predicted in Eq. (8).

Comparing Table V with Table IV, we conclude that although the mean values are estimated correctly, the quantiles are over estimated in the naive, rule of thumb approach. We can add these observations to the list of rules:

6) The mean value of total latency is $\sim 1/2$ of the sum of the maximum seek time and rotation period.
7) The computation of the tail latencies, i.e., quantiles require the complete model, see Eq. (16).

This completes the mathematical modeling of the total seek time and we move to the section where we verify the model with simulations and actual test results.

VI. Simulation and Experimental Test Results

In order to verify the predicted distributions, we first develop Monte Carlo simulations in which we select $100,000$ random numbers x and y in the range $[-r_o, r_o]$ from a uniform distribution, and reject all x, y pairs that are outside of the disk surface. Figure 7 shows that this selection method results in a uniform coverage of the disk. We then randomly select a pair of points $\vec{r}_1$ and $\vec{r}_2$, and build the histogram of $|r_2 - r_1|$.

The experimental data collection is done with several 7200 rpm 3.5" form factor HDDs from Seagate Exos X22 series. The data consist of about 1000 random seeks. Figure 8 shows the simulation and experimental data for the probability density of the radial seek distance. The shaded areas show the simulated and measured values whereas the curve is calculated using Eq. (8). As we argued in Section IV-C, the seek time as can be expressed a function of the seek distance as in Eq. (9), and this relation is confirmed with the experimental data in Figure 9. It is important to note that if we used a linear model, the acceleration domain, $t < 8.8$ms, would not have been captured properly, and furthermore, the line would have tilted causing an larger discrepancy in the fit for the tail statistics. Figures 10 and 11 show the radial seek time and total seek time respectively. The top subplots are based on simulation data and the bottom subplots are measurement data. All the figures are overlayed with the black lines which represent the predicted curves from the model. The model predictions, simulation and test data are all in very good agreement.

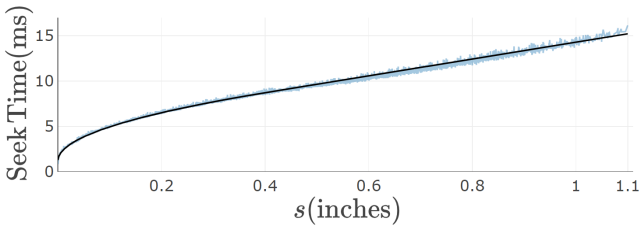

Fig. 9: Experimental data on seek time vs the radial distance collected with a 3.5" form factor HDD. The curve shows a three-parameter fit to the data, see Eq. (9) with parameters $\alpha = 0.014 \frac{\text{inch}}{\text{ms}^2}$, $\beta = 1.18$ms, and $t_a = 8.8$ms.

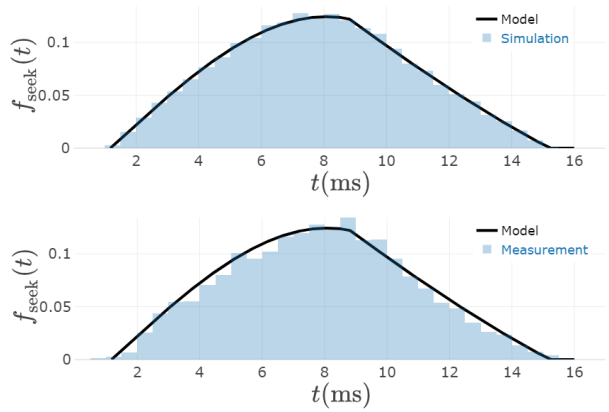

Fig. 10: The seek time distributions with a 3.5" form factor HDD for the simulated and measured data. The curves are calculated using Eq. (11) and they show good agreement with the simulation and experimental results shown in bars.

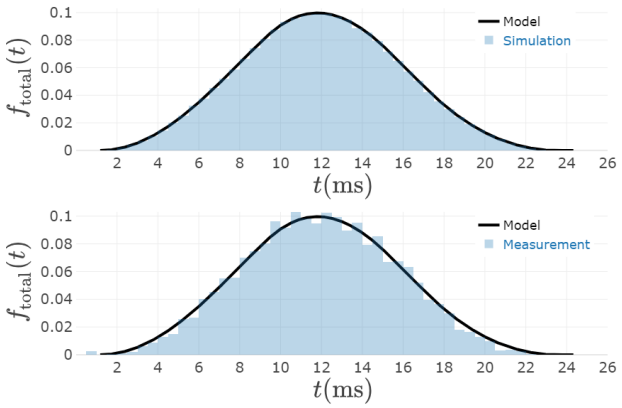

Fig. 11: The distribution of the total latency (seek and rotational latencies combined) for the simulated and measured data. The blue shaded bars show the simulation and test data densities and the curves are the density as predicted with Eq. (16).

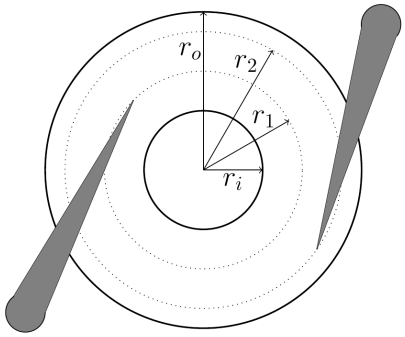

Fig. 12: Illustration of a drive with dual actuators positioned on the opposite sides of the disk.

VII. Modeling Latencies for Dual Actuator HDDs

Adding to the complexity of estimating and precisely computing HDD latencies is the recent emergence of a new type of HDD called *multi-actuator HDDs*. Although capacities of enterprise HDDs have been increasing over years, the I/O performance has been limited by the mechanical motion of the actuators. This creates challenges for modern datacenters because the ratio of performance to capacity has been steadily decreasing with each new generation of HDDs. This trend is responsible for the emergence of RAID6 [28] in the 1990s, the more recent emergence of wider erasure codes [29], the introduction of burst buffers in high-performance computing [30], and the identification of the importance of IOPS/GB in modern cloud datacenters [4].

To address this issue, HDDs with two actuators (instead of one) have been recently brought to the market [31]. These multi-actuator HDDs two actuators but each surface is only accessed by one of the two actuators. These HDDs have double the IOPS/GB and BW/GB than their single-actuator

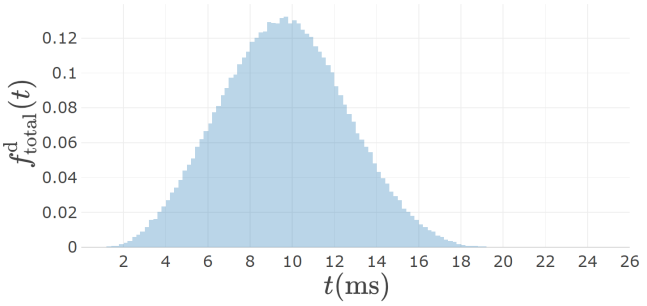

Fig. 13: The simulated distribution of the total positional latency that combines seek and rotational latencies for dual actuator HDDs in which two actuators access each surface. We consider an 7200 rpm drive with a maximum seek time of 14 ms.

counterparts but the latency calculations we have presented thus far are not affected.

However, future multi-actuator HDDs are envisioned that will have two actuators (and therefore two heads) accessing each surface, as illustrated in Figure 12. In this section, we study the latency effects of these potential future multi-actuator HDDs.

It is important to note that we cannot simply create two independent total latency metrics based on Eq. (15) and calculate the distribution for the smaller of the two. This is caused by a strong correlation between the rotational latencies for the two actuators due to their respective locations on opposite sides of the surface. More precisely, if the down-track angle of a random sector with respect to the first head is ϕ, then the angle with respect to the second read-write head is $\phi+\pi$ moduli 2π. In the worst case scenario, the first head will be in the innermost or outermost track, and will be unable to access the target sector if it rotates past the head more quickly than the maximum radial seek time, T_R. In this case, however, the target sector can be accessed more quickly by the other actuator, and the resulting worst-case total positioning time will be $T_R + T_0/2$.

We can derive an approximate form of density function for the total positional latency by replacing T_0 in Eq. (16) with $T_0/2$. Figure 13 shows that the our simulated results. We also show various statistics of the total positional latency for these dual actuator HDDs in Table VI.

mean	sigma	Q50	Q75	Q95	max
9.6	2.9	9.6	11.6	14.5	19.1

TABLE VI: Various key statistics for the total latency for dual actuator drives in milliseconds.

Comparing Tables V and VI, we see that this dual actuator HDD drive essentially halves the rotational latency and improves the tail statistics roughly by $T_0/2$ compared to HDDs with only one actuator accessing each surface.

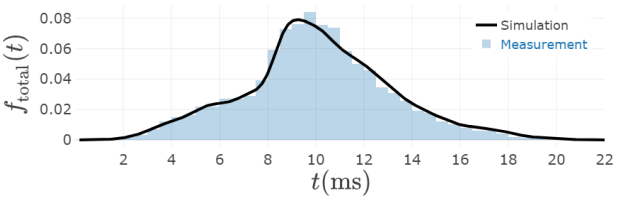

Fig. 14: The overlay of simulation data (black line) and the measured data (bars) for queue depth=2. The experimental data is collected with 7200 rpm 3.5" form factor Exos X22 series drives from Seagate.

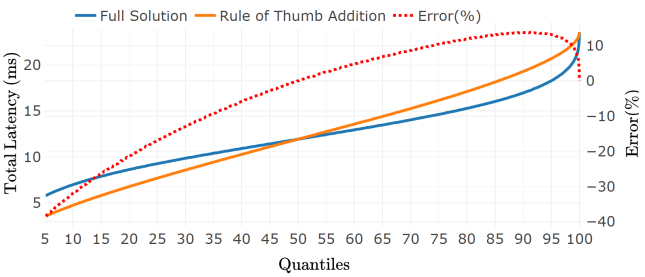

Fig. 15: The total latency quantiles calculated using the precise solution in blue, and the rule of thumb estimates in orange. The error, shown in dashed red line [right axis], made with the rule of thumb addition is as large as 13.3%.

VIII. Command Queuing

So far we have quantified the statistic of the latency under random workloads. In practical applications, an HDD is queried with a set of sectors and it is allowed to reorder outstanding commands to improve the latencies. The statistics of the latencies will depend on the ordering algorithm and it might be very complicated as the queue depth increases. Furthermore, such algorithms are considered as intellectual property and are not readily available publicly. In this section, we consider a queue depth of 2, which is simple enough to estimate what the HDD algorithm might be doing. We feed the HDD with a set of random seek commands with depth 2 and measure the latencies. For simulations, we take our latency model as in Eq. (16), and feed it with a set of random seeks. The simulation algorithm takes in a pair of incoming data sectors and selects the one with the shortest time from the current position of the read-write head. The process is repeated until all the commands are fulfilled. Figure 14 shows the measured and simulated data which are in good alignment.

IX. Real world implications

There are possible significant implications of using these inaccurate rules of thumbs. Figure 15 shows the full extent of possible error when using the rule of thumb to compute the quantiles as compared to using the precise values. As shown, the inaccuracy can be quite large. For example, when computed with the more precise statistical model, the 95% quantile of the positioning latency is about 18.3ms. Comparing to the rule of thumb estimate of 20.7ms reveals an overestimation of 13.2% error. This inaccuracy can result in significant system design errors with correspondingly large economic penalties.

For example, imagine a data center which has requirements for multiple petabytes of storage and wants to achieve a 95% tail latency no greater than 20 ms for random block IO. As we saw in the previous section, there is approximately 13.2% inaccuracy when using the basic rule of thumb to compute this latency with a resulting incorrect value of 20.7 ms. If the data center designer uses this inaccurate value to build a system meeting these requirements, they will make a costly mistake. For example, how might such a data center designer attempt to achieve this latency goal? Since the rotational speeds of nearline drives are fixed at 7200 rpm, the only mechanism to improve the latency is to reduce the seek time by decreasing the maximum seek distance.

There are at least two ways to do so. One would be to add a cache to store some number of either the innermost or the outermost tracks (typically one would cache the innermost tracks since the outermost tracks have a higher bandwidth). A second approach would be to simply not use those innermost tracks [32]. Both of these approaches would effectively decrease the ratio between r_i and r_o thereby decreasing the observed latencies.

In the specific example above, limiting the inner radius to 0.85", would result, using the inaccurate rule of thumb, in a newly computed 95% tail latency matching the target of 20 ms. However, the more accurate computation shows that the system already meets the requirements and can use the full radii from [0.73", 1.83"]. Using the inaccurate rule of thumb in this scenario results in either unnecessarily losing 7% of the capacity or unnecessarily adding a cache for 7% of the capacity.

X. Conclusions

Rules of thumb have long been useful approximations in compute systems design. In this paper, we examined long-standing rules of thumb for HDD latencies. First, we show that the average seek latency might differ from 1/3rd of the maximum seek latency depending on the geometry of the disks. Second, we provide a detailed model to combine seek time with the rotational latency to compute the total positioning latency. Our model accurately predicts the complete statistics of the total latency as shown by the simulations and actual drive test results.

Properly designing storage systems is a challenging endeavor in which any inaccurate inputs to the design can cause either costly missed requirements or costly overprovisioning. It is therefore crucial that critical performance characteristics of HDDs can be correctly considered during system design. We therefore offered mathematical models, verified with Monte Carlo simulation and experimental data, warning system providers that long-trusted rules of thumb can be significantly inaccurate. Our model can not only be elegantly expressed in a single closed form formula as in Eq. (16), but

also is accurate enough to be used to study the aspects of various scheduling methods without simulations.

It is our hope that our newly introduced models, as well as a few newly introduced rules of thumb, will aid future system designers to build more precisely provisioned data centers. To further aid future designers, we introduced, and verified, models for calculating latencies of emerging multi-actuator HDDs. Finally, we presented a hypothetical scenario in which using the imprecise historical rules of thumb can result in an unnecessary purchase of a cache equivalent to 7% of the total storage system (or a loss of 7% of the capacity).

XI. Future Work

Another storage technology is the shingled magnetic recording (SMR) [33] where tracks are heavily overlapped in bands. SMR enables higher track density, hence, higher capacity drives. However, any re-write of any sector will require the re-write of the entire zone degrading the write performance. A hybrid approach can be taken to optimize the capacity and the performance trade off by allocating inner radii to SMR and outer radii to conventional magnetic recording. Given the analytical nature of our model in terms of the inner and outer radii values, it can be leveraged to decide the critical value of the radius to switch from SMR to CMR depending on the expected read and rewrites of the user workload.

Acknowledgment

A portion of this work was completed when S.Ö., J. B., and Y. D. were at Seagate.

References

[1] Feng Chen, David A Koufaty, and Xiaodong Zhang. Understanding intrinsic characteristics and system implications of flash memory based solid state drives. *ACM SIGMETRICS Performance Evaluation Review*, 37(1):181–192, 2009.

[2] Richard F Freitas and Winfried W Wilcke. Storage-class memory: The next storage system technology. *IBM Journal of Research and Development*, 52(4.5):439–447, 2008.

[3] Andy Klein. Backblaze Drive Stats for Q2 2022. https://www.backblaze.com/blog/backblaze-drive-stats-for-q2-2022/, 2022.

[4] Eric Brewer, Lawrence Ying, Lawrence Greenfield, Robert Cypher, and Theodore T'so. Disks for data centers. Technical report, Google, 2016.

[5] Reinsel, D., Gantz, J., Rydning, J. The Digitization of the World From Edge to Core. https://www.seagate.com/files/www-content/our-story/trends/files/idc-seagate-dataage-whitepaper.pdf, 2018. Online; accessed 19 September 2022.

[6] Jon Bentley. The Back of the Envelope. Communications of the ACM, March 1984.

[7] Jeff Dean. Software Engineering Advice from Building Large-Scale Distributed Systems. http://static.googleusercontent.com/media/research.google.com/en/us/people/jeff/stanford-295-talk.pdf, 2003.

[8] E. G. Coffman, L. A. Klimko, and Barbara Ryan. Analysis of scanning policies for reducing disk seek times. *SIAM Journal on Computing*, 1(3):269–279, 1972.

[9] Remzi H. Arpaci-Dusseau and Arpaci-Dusseau Andrea C. *Operating Systems: Three Easy Pieces*. Arpaci-Dusseau Books, LLC, 1.00 edition, 2015. http://pages.cs.wisc.edu/~remzi/OSTEP/.

[10] Youmin Chen, Youyou Lu, Kedong Fang, Qing Wang, and Jiwu Shu. utree: a persistent b+-tree with low tail latency. *Proceedings of the VLDB Endowment*, 13(12):2634–2648, 2020.

[11] Christina Delimitrou and Christos Kozyrakis. Amdahl's law for tail latency. *Communications of the ACM*, 61(8):65–72, 2018.

[12] Kostis Kaffes, Timothy Chong, Jack Tigar Humphries, Adam Belay, David Mazières, and Christos Kozyrakis. Shinjuku: Preemptive scheduling for {μsecond-scale} tail latency. In *16th USENIX Symposium on Networked Systems Design and Implementation (NSDI 19)*, pages 345–360, 2019.

[13] Pulkit A Misra, María F Borge, Íñigo Goiri, Alvin R Lebeck, Willy Zwaenepoel, and Ricardo Bianchini. Managing tail latency in datacenter-scale file systems under production constraints. In *Proceedings of the Fourteenth EuroSys Conference 2019*, pages 1–15, 2019.

[14] Adam Manzanares, Filip Blagojevic, and Cyril Guyot. {IOPriority}: To the device and beyond. In *9th USENIX Workshop on Hot Topics in Storage and File Systems (HotStorage 17)*, 2017.

[15] C. Ruemmler and J. Wilkes. An introduction to disk drive modeling. *Computer*, 27(3):17–28, 1994.

[16] Christopher R. Lumb, Jiri Schindler, Gregory R. Ganger, Erik Riedel, and David F. Nagle. Towards higher disk head utilization: Extracting "free" bandwidth from busy disk drives. In *Fourth Symposium on Operating Systems Design and Implementation (OSDI 2000)*, San Diego, CA, October 2000. USENIX Association.

[17] Christopher R. Lumb, Schindler, and Gregory R. Ganger. Freeblock scheduling outside of disk firmware (cmu-cs-01-149). Jun 2018.

[18] Field Cady, Yi Zhuang, and Mor Harchol-Balter. A stochastic analysis of hard disk drives. *Hindawi Publishing Corporation International Journal of Stochastic Analysis Article ID*, 390548, 01 2011.

[19] Rodney Van Meter. Observing the effects of Multi-Zone disks. In *USENIX 1997 Annual Technical Conference (USENIX ATC 97)*, Anaheim, CA, January 1997. USENIX Association.

[20] Neil C. Wilhelm. A general model for the performance of disk systems. *J. ACM*, 24(1):14–31, jan 1977.

[21] Bruce L. Worthington, Gregory R. Ganger, and Yale N. Patt. Scheduling algorithms for modern disk drives. In *Proceedings of the 1994 ACM SIGMETRICS Conference on Measurement and Modeling of Computer Systems*, SIGMETRICS '94, page 241–251, New York, NY, USA, 1994. Association for Computing Machinery.

[22] S.W. Ng. Advances in disk technology: performance issues. *Computer*, 31(5):75–81, 1998.

[23] Seagate Technologies. private communication, Jan 2023.

[24] Michael A. Cordle, Drew M. Mader, Steven D. Granz, Alfredo S. Chu, Pu-Ling Lu, Frank Martens, Ying Qi, Tim Rausch, Jason W. Riddering, and Kaizhong Gao. Radius and skew effects in an hamr hard disk drive. *IEEE Transactions on Magnetics*, 52(2):1–7, 2016.

[25] Pulkit A. Misra, María F. Borge, Íñigo Goiri, Alvin R. Lebeck, Willy Zwaenepoel, and Ricardo Bianchini. Managing tail latency in datacenter-scale file systems under production constraints. In *Proceedings of the Fourteenth EuroSys Conference 2019*, EuroSys '19, New York, NY, USA, 2019. Association for Computing Machinery.

[26] Yin Li et al., Hao Wang, Xuebin Zhang, Ning Zheng, Shafa Dahandeh, and Tong Zhang. Facilitating magnetic recording technology scaling for data center hard disk drives through filesystem-level transparent local erasure coding. In *Proceedings of the 15th Usenix Conference on File and Storage Technologies*, FAST'17, page 135–148, USA, 2017. USENIX Association.

[27] Zoran Dimitrijevic and Raju Rangaswami. Design and implementation of semi-preemptible IO. In *2nd USENIX Conference on File and Storage Technologies (FAST 03)*. USENIX Association, March 2003.

[28] Walter A. Burkhard and Jai Menon. Disk array storage system reliability. *FTCS-23 The Twenty-Third International Symposium on Fault-Tolerant Computing*, pages 432–441, 1993.

[29] Adam Leventhal. Triple-parity raid and beyond: As hard-drive capacities continue to outpace their throughput, the time has come for a new level of raid. *Queue*, 7(11):30–39, 2009.

[30] Teng Wang, Sarp Oral, Yandong Wang, Brad Settlemyer, Scott Atchley, and Weikuan Yu. Burstmem: A high-performance burst buffer system for scientific applications. In *2014 IEEE International Conference on Big Data (Big Data)*, pages 71–79. IEEE, 2014.

[31] Feist, J. Multi Actuator Technology: A New Performance Breakthrough. https://blog.seagate.com/enterprises/mach2-and-hamr-breakthrough-ocp/, 2019. Online; accessed 19 September 2022.

[32] W. W. Hsu and A. J. Smith. The performance impact of i/o optimizations and disk improvements. *IBM Journal of Research and Development*, 48(2):255–289, 2004.

[33] Mary Dunn and Timoty Feldman. Shingled magnetic recording – smr models, standardization, and applications, Sep 2014.