Understanding the Performance-Temperature Interactions in Disk I/O of Server Workloads

Youngjae Kim† Sudhanva Gurumurthi‡ Anand Sivasubramaniam†

† Computer Systems Laboratory
Department of Computer Science and Engineering
The Pennsylvania State University, University Park, PA 16802
{youkim, anand}@cse.psu.edu

‡ Department of Computer Science
University of Virginia, Charlottesville, VA 22904
{gurumurthi}@cs.virginia.edu

Technical Report CSE-05-007
November, 2005

Abstract

Brisk growth in the data rate of disk drives has been instrumental in developing high performance storage systems for a diverse spectrum of applications. However, the pace of growth in the data rate is expected to significantly slow down in the near future, hindered mainly by the thermal design constraints of the drives. It is important to understand how real workloads exercise the disks, and how the resulting activities affect the drive temperature. Such an understanding can help us push the performance envelope by designing drives for average case behavior, with safeguards to control disk activities whenever we reach thermal emergencies. However, there is a critical void in toolsets today for integrated performance-thermal studies of disk drives with real workloads.

This paper describes the first infrastructure for integrated studies of the performance and thermal behavior of storage systems. Using microbenchmarks running on this infrastructure, we first gain insight into how I/O characteristics can affect the temperature of disk drives. We use this analysis to identify the most promising, yet simple, "knobs" for temperature optimization of high speed disks, which can be implemented on existing disks. We then analyze the thermal profiles of real workloads that use such disk drives in their storage systems, pointing out which knobs are most useful for dynamic thermal management when pushing the performance envelope.

Keywords: Storage System, Disk Drives, Power and Temperature Management.

1 Introduction

A steady growth in the data rate of disk drives has been instrumental in their successful deployment across a diverse range of environments. In addition to data-centric services such as file, web and media servers, transaction processing, etc., disk drive performance is becoming extremely critical for even consumer electronic products such as digital video recorders, personal entertainment and gaming devices. While parallelism using RAID [25] has been effectively employed in server environments for higher bandwidth, the growth in the raw data rate is still very important for single drive performance across all these applications.

The internal data rate (IDR) of the drive is dependent on the linear density, rotational speed, and the platter size. The IDR has been growing at an exponential rate of 40% per-annum over the last fifteen years, due to a combination of brisk growth in recording density in linear density and higher rotational speeds (expressed in Rotations-per-Minute or RPM). However, increasing the RPM leads to excessive heat being generated since the viscous dissipation is proportional to nearly the cubic power [7]. In order to ensure that the disk drives adhere to the thermal design constraints when increasing the RPM, the platter size (which is proportional in nearly the fifth power to heat) may need to be reduced. This provides a margin within which the target IDR can be achieved for the same amount of heat by merely shrinking the platters and then compensating for the smaller size by increasing the RPM appropriately.

Designing disks to operate within the thermal design envelope is critical for reliable operation [1]. High temperatures can cause a host of reliability problems, such as off-track writes due to the thermal tilt of the disk stack and actuators, which can lead to corruption of data, or even a complete failure of the device due to a head crash [16]. It may appear that a simple solution to this problem is to provision a more powerful cooling system, since that would facilitate the extraction of heat from the device, thereby reducing its operating temperature. However, such cooling systems are prohibitively expensive [34].

It has been shown that the pace of growth in the linear density is expected to slow in the future, requiring much more aggressive scaling of the RPM to sustain the IDR growth rate [14]. Furthermore, this study showed that such aggressive scaling of the RPM cannot be sustained within the thermal envelope even for very small platter sizes thereby leading to a significant slowdown in the IDR growth rate in the near future. The implication of this is that disks in the future would have to be designed for *average case* thermal behavior rather than the worst case situation, incorporating the characteristics needed for higher performance, such as a higher RPM. However, this design approach can cause the operating temperature to exceed the the thermal envelope at runtime, if we do not incorporate any safeguards. To avoid thermal emergencies, [14] suggested the use of *Dynamic Thermal Management (DTM)*, a philosophy that is being actively investigated in the context of microprocessor design as well [4, 30]. In DTM, the disk is allowed to serve I/O requests as usual. However, if there is an imminent danger of violating the thermal envelope, we dynamically modulate the drive activities to prevent such a situation from occurring.

Designing and optimizing DTM techniques requires a careful analysis of how different drive activities impact the temperature, using real workloads. For instance, if we have a disk operating at a given RPM, how do the seeks in the workload increase the temperature? How far apart do seeks need to be in order to remain within the thermal envelope? Within a seek, how do the different phases - acceleration, coast, deceleration - impact the temperature? Can we modulate the head scheduling or request service schemes for DTM? Given different DTM alternatives, how do we pick one over another for a given set of workload conditions and disk drive parameters?

Such a detailed understanding of the interaction between workload activities and disk drive parameters, and their impact on temperature, requires detailed toolsets that are currently unavailable. Though there are tools such as Disksim [10] which are widely used for performance studies, there is no tool available today to study the temperature of a drive running a real workload. The earlier work in thermal modeling of disk drives [7, 14] has been more intended to study the temperature of drives under steady state conditions for static configurations of different drive parameters, and have not really looked at the temperature during the dynamic execution of a workload.

With these motivations, this paper makes the following contributions for facilitating temperature-aware storage system design:

- We present the first integrated performance-thermal simulator to study the temperature of disk drives with

real workloads. This has been accomplished by extending the discrete-event Disksim performance simulator with a time-step simulation of the temperature, which sets up heat flow equations for the drive and solves them with a Finite Difference Method. The development has also involved detailed modeling of the components of the disk seek. The simulator is fairly modular with clear interfaces between the two components to facilitate DTM studies.

- Using this simulator, we study the impact of disk I/O on the drive temperature for a constant RPM disk drive. With detailed microbenchmarks we study the impact of the seek components - acceleration, coast and deceleration, together with the temporal separation between seeks, on the drive temperature. We point out that cooling can be achieved by one or more of the following (i) very short seeks (i.e. head scheduling algorithms such as SPTF may perform temperature control as well) so that the voice-coil motor (VCM) is not active for too long, (ii) very long seeks (i.e. a reverse of SPTF may sometimes be preferable) to amplify coast times when VCM is off, and (iii) increasing the temporal separation between seeks.

- We profile the thermal behavior of real server workloads and show how the temperature varies during the execution. We also show that the spatial locality (minimizing seek activity) and the temporal separation between the seeks is adequate in these workloads that we can automatically apply a 5000 RPM boost to their baseline disk configurations (which were designed to remain within the thermal envelope for worst case conditions) without exceeding the thermal envelope. This results in around 21-53% improvement in response times. Higher RPMs mandate more active DTM schemes. However, we show that the scope of such schemes is rather limited if we have to work under the constraints of a constant RPM drive.

The organization of the rest of this paper is as follows. Section 2 reviews the related work in this area. Section 3 describes our integrated thermal-performance framework, and the microbenchmark evaluations are given in Section 4. The evaluation with real workloads is conducted in section 5 for different drive RPMs. Section 6 explains a simple delay-based DTM scheme, and finally section 7 concludes this paper.

2 Related Work

There have been many prior studies on the power consumption of disk drives [19, 35] and its optimization in mobile/desktop systems. Prediction of idleness is used to spin down the disk to a low power mode during periods of inactivity [22, 8]. [24] uses a combination of prefetching and caching to increase such idleness for more effective power management.

More recently, there has been interest in reducing the disk power consumption in server systems [15, 5]. The problem is more challenging in these environments because the workloads may not have sufficient idleness, and may not tolerate degradation in performance. Further, server disks have quite different characteristics compared to their laptop/desktop counterparts [1], with much larger transition times to/from the low power modes. The solutions for server environments employ multi-speed/DRPM disks [12, 5], which can be used in conjunction with other techniques such as data clustering [26] or cache management [36, 37].

Another approach is to use flash memory (which consumes lower power and is also faster than a disk) to construct a large buffer, to increase disk idleness. In fact, Samsung recently announced a flash based disk that can provide over 16 GB of storage [27]. Such a disk can delay writes to the magnetic disk by accumulating them in the flash buffer and doing a bulk write. Although this solution is good for laptops and desktops, where I/O traffic is lower, it is not easily applicable for servers. First, the higher rate of I/O traffic in server systems can easily wear out the flash memory. It has been shown that under a typical disk usage scenario in a server and 100,000

write/erase cycles for the flash disk, wearout would occur in less than two years [23]. Secondly, since many server workloads (which have large working sets and may involve several compulsory misses) tend to have more disk reads than writes, many of the operations will still need to access the disk.

Temperature-aware design is becoming important in the context of microprocessors [30], interconnection networks [28], and storage systems [14] due to its strong correlation to the reliability of components and the high cost of cooling. [7] describes a model of the thermal behavior of a disk drive based on several parameters such as drive geometry, number of platters in the disk stack, RPM, and materials used for building the drive. However, this model [7], and the other closely related work in this area [14], are both studies of the thermal behavior of disk drives (based on different drive parameters) under static conditions, and the behavior has not been previously studied during the dynamic execution of real workloads. There has also been a study on modeling and designing disk arrays in a temperature-aware manner [20].

Dynamic Thermal Management (DTM) has been proposed as an architectural solution to manage the temperature [4, 31, 29] while pushing the performance envelope. The basic idea of any DTM scheme is to operate the component/system, which has been designed for an *average-case* behavior (rather than the worst-case as is conventionally done) and to modulate its activities at runtime to avoid exceeding the thermal envelope.

3 A Framework for Integrated Thermal-Performance Simulation

In order to analyze the thermal behavior of applications (and possibly control it dynamically), we need a framework that can relate activities in the storage system to their corresponding thermal phenomena as the workload execution is in progress. In a real system, this can be achieved by instrumenting the I/O operations and leveraging the thermal sensors [16] that are commonplace in most high-end disks drives today. However, since the objective of this study is to investigate the effect of disk configurations that are not yet available in the market today, using highly controlled experiments (without external perturbances), we use a simulation-based approach. In this section, we describe the simulation framework that we have developed to study performance and thermal behavior of storage systems in an integrated manner.

The simulator consists of two components, namely, a *performance model* and a *thermal model*. The performance model is used to simulate all activities in the storage system that could potentially affect workload performance, such as, interconnect latencies, cache accesses, multi-disk organizations (such as RAID), seek/rotational latencies, etc. The thermal model, on the other hand, considers the effect of parameters that can affect the temperature of the storage system components.

In our simulator, the performance model we use is Disksim [10], which models the performance aspects of the disk drives, controllers, caches, and interconnects in a fairly detailed manner. Disksim is an event-driven simulator, with the simulated time being updated at discrete events, e.g. arrival of request, completion of seek, etc. Disksim has been extensively used in different studies and has been widely validated with several disk models.

Our thermal simulation model is based on the one developed by Eibeck and Cohen [9]. The sources of heat within the drive include the power expended by the spindle motor (to rotate the platters) and the voice-coil motor (for moving the disk arms). The thermal model evaluates the temperature distribution within a disk drive from these two sources by setting up the heat flow equations for different components of the drive such as the internal air, the spindle and voice-coil motor assemblies, and the drive base and cover. It uses the finite difference method [21] to calculate the heat flow, and iteratively calculates the temperatures of these components at each time step until it converges to a steady state temperature. Such a simulation model is sometimes referred to as a time-step simulator.

There are some parameters/events of one model which may not affect the other. For instance, the performance

model does not really need to know the external air temperature, or the nature of the platter material. Similarly, the thermal model is relatively unaffected by cache hits (it is as though there was no incoming request). However, there are several shared parameters/state that need to flow from one to another. The thermal model needs to know when the seeks start and end (to account for VCM power), and also the operating RPM. If one is interested in modeling a Dynamic Thermal Management (DTM) technique, then information has to flow from the thermal model back to the performance model in order to control drive activities. Such information/state flow from one to the other needs to be carefully done, while ensuring the simulated time is reconciled between the two models - a discrete event model and a time-step model - for each disk.

Our integration of these two models is based on the observation that the only two governing factors from the performance model which affect the thermal model include the seek activity (particularly the VCM on and off events) and any RPM changes (if using a multi-speed/DRPM disk). At these points, the performance model invokes the thermal model to iteratively (time-steps) compute the heat flows until the simulated time of the thermal model reaches the simulated time of the next such point in the performance model. In other words, we normally run the performance model for the sequence of incoming I/O requests. Whenever this model incurs a VCM switch from its prior state (i.e. on from off, or vice-versa), it invokes the thermal model with the appropriate VCM state information so that the thermal model can catch up on its time to the time in the performance model, at which point control flows back to the performance model. In the case of a multi-speed disk, this invocation is also done at RPM change events.

Such an integration between the two models requires a careful tuning of the time-step in the thermal model, since it affects both the speed and accuracy of the simulation. A relatively large time-step, as can be expected, can give a faster simulator at the expense of lower accuracy, and a finer granularity would give high accuracy at a slower speed. To evaluate these trade-offs, we ran workloads comparing their temperature profile using different time-step granularities (varying between 100 to 2500 steps between successive I/O events in the performance simulator), with that of a high resolution thermal simulation (60,000 steps/minute). We found that a choice of 400 steps of the time-step thermal simulation between any two successive I/O requests of the performance simulator gives a thermal profile that is within 0.1% of the high resolution version, and we use this value in our experiments. Note that this translates to different granularities for a time-step during the course of a simulation, depending on how far temporally apart are the events in the performance simulator.

3.1 Modeling the Physical Behavior of Disk Seeks

When doing the thermal-performance simulation, one of the activities that needs to be modeled accurately is the dynamics of a physical seek operation. Although the time taken for a seek is already accounted for by the performance model, the mechanical work involved to effect the seek operation has a strong influence on temperature and needs to be accounted for.

The seek time depends on two factors, namely, the inertial power of the VCM assembly and the radial length of the data band being traversed on the platter [11]. The VCM, which is also sometimes referred to as the arm actuator, is used to move the disk arms across the surface of the platters. Physically, a seek involves an acceleration phase, when the VCM is powered, followed by a coast phase of constant velocity where the VCM is off, and then a deceleration phase to stop the arms near the desired track when the VCM is again turned on but the current is reversed to generate the braking effect. This is then followed by a head settling period. For very short seeks, the settle time dominates the overall seek time whereas for slightly longer (intermediate) seeks, the acceleration and deceleration phases dominate. Coasting is more significant for long seeks. We capture the physical behavior of seeks using a Bang-Bang Triangular model [18]. In this model, for any physical seek operation, the time taken

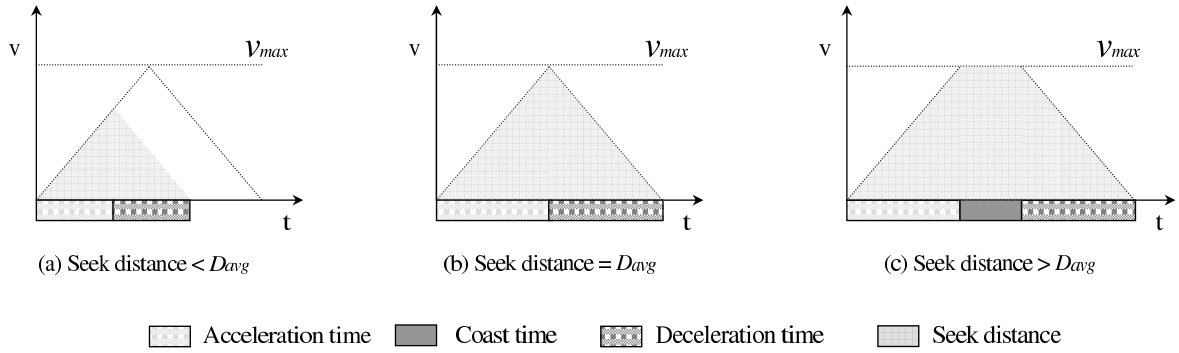


Figure 1: Different possibilities for a physical seek operation.

for acceleration and subsequent deceleration are equal. To calculate the acceleration/deceleration components, we make the following assumptions:

- The head settle time is approximated as the track-to-track seek time.
- Let V_{max} denote the maximum velocity that is permissible for the head, which is dictated by the characteristics of the VCM assembly and also by the bandwidth of the underlying servo system (needed to accurately position the head over the desired track). We use a V_{max} value of 120 inches/second, which reflects many modern disk drive implementations.
- The average seek distance (D_{avg}) for a large number of random seeks is equal to a seek across $\frac{1}{3}$ of the data zone [3].
- The coast time for an average seek (of this distance D_{avg}) is zero, since that would yield the lowest seek time on the average.

The last three assumptions are essentially used to fix/calculate the acceleration/deceleration of the VCM based on what is needed to bring the head assembly to a maximum velocity (V_{max}) immediately followed by the reverse braking/deceleration to give the lowest possible seek time when the average covered distance is D_{avg}.

Let the outer radius of the disk drive be denoted as r_o. We set the inner radius to be half that of the outer radius, i.e., $r_i = \frac{r_o}{2}$ [14]. Let D_{avg} and T_{avg} denote the distance of $\frac{1}{3}$ of the data zone and the corresponding (average) seek time. Since we are calculating the time only during the movement of the disk arm and not the settling period, T_{avg} is adjusted by subtracting the settle time of a head (i.e., the track-to-track seek time) from the average seek time. We can now calculate the time taken during the acceleration, coast, and deceleration phases of a physical disk seek operation (of distance d) as follows:

- Case $d = D_{avg}$: For a seek operation that needs to traverse a distance of D_{avg} as is shown in Figure 1 (b), the VCM accelerates the actuator from an initial speed of 0, to a maximum velocity V_{max}, and then immediately applies the reverse braking affect which takes the same amount of time as the acceleration, i.e. there is zero coast time and the VCM is on during the entire duration of the seek. To find this duration, we apply the Third Law of Motion to calculate the acceleration as follows:

  $Acc = \frac{V_{max}^2}{2 \times D_{avg}/2}$.

  We can then apply this acceleration in the First Law of Motion to calculate the total time for the seek (of distance D_{avg}) as:

$T_{Acc} = T_{Dec} = \frac{V_{max}}{Acc}$

So when the requested seek distance d is D_{avg}, the VCM is continuously on for this entire duration of $T_{Acc} + T_{Dec}$.

- Case $d > D_{avg}$: Since the actuator cannot move faster than V_{max}, once it reaches this velocity after the initial acceleration, there needs to be a coast phase (as depicted in Figure 1 (c)) before the deceleration. Note that the VCM is on during the T_{Acc} and T_{Dec} (whose values are the same as in the previous case) phases, with a coast time duration of $\frac{d-D_{avg}}{V_{max}}$ in between when the VCM is off.

- Case $d < D_{avg}$: The distance is lesser than what is needed to reach the maximum velocity for the calculated acceleration above. Consequently, we again only have an acceleration phase followed immediately by the deceleration phase. We can apply the Second Law of Motion to calculate the T_{Acc} and T_{Dec} in this case as follows:

  $T_{Acc} = T_{Dec} = \sqrt{\frac{2 \times \frac{d}{2}}{Acc}}$

The on/off states of the VCM are then communicated to the thermal model at the appropriate points as explained earlier.

Validation: In order to validate this model, we calculated the acceleration that is computed by our model, under all the stated assumptions for a Fujitsu AL-7LX disk drive, which is a 2.6" 15,000 RPM disk drive, and compared it to its measured mechanical seek characteristics [2]. Using the drive characteristics, we found the D_{avg} for this disk to be 0.22". The reported value for the acceleration to satisfy the seek time requirement is 220 G (2150 m/s^2), whereas our model calculates it using the D_{avg} to be 253.5 G (2488.1 m/s^2), which is within 15% of the reported value.

3.2 Simulation "Warm-up"

At the beginning of the simulation, all the disks are in a cold state, having the same temperature as that of the outside air. It takes roughly 50 minutes of simulated time before the temperature reaches a steady state. In order to prevent start-up effects from skewing our results, we perform the experiments only after the system has reached the steady state temperature. To simulate this physically warmed-up state, we use a method that is similar to "simulation-warm-up", which is common in computer architecture studies. We literally warm up the disk by running the stand-alone thermal model for the first 150 minutes of simulation assuming that the disks are idle (i.e., the disks are spinning but there are no arm movements). Simulation of the workload is started after this warm-up period.

4 Impact of I/O Activities on Disk Drive Thermal Behavior

In order to understand the thermal behavior of real workloads, we first analyze the impact of various types of I/O activities on the temperature of a disk drive. Let us first look at all the operations needed to service a disk request from the physical medium. First, a seek may need to be performed to take the heads to the destination track from where the data access needs to begin. As we have seen in Section 3, a seek is composed of an acceleration of the arms, a period of coasting, and then a deceleration. The VCM is active during the acceleration and deceleration phases, and not during the coast. The time that the arms coast may be zero based on the physical distance that

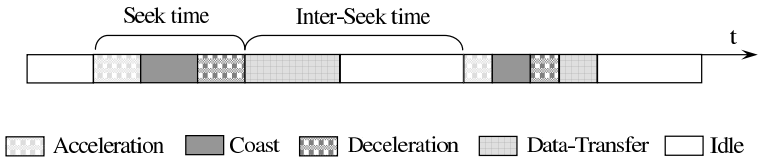


Figure 2: The different physical phases of an I/O operation to a disk.

needs to be traversed. The seek is then followed by a head settling period. After this, the actual data transfer (read/write) begins between the disk and the internal electronics of the drive. This is then followed by an idle period (potentially of zero duration) before the next request is serviced.

From the sequence of events shown in Figure 2 to serve disk requests, we see that the temperature variation of a disk operating at a given RPM depends on the *seek time*, *coast time*, and the *inter-seek time*, pictorially shown in Figure 2. Even though the coast is in turn accounted for in the seek times (i.e. a large coast does translate to a large seek time), we would like to identify this as a separate factor in our studies since its effect counter-acts the acceleration/deceleration effects (a long coast can possibly allow the disk to cool since the VCM is off). A seek operation that accelerates to the maximum velocity, V_{max} and subsequently decelerates without any coast time (i.e. the profile in Figure 1 (b)) generates the maximum heat for any given seek operation. Let us denote this type of seek operation as a *min-coast* seek. Note that the coast is zero even for those seeks with distances less than D_{avg} (Figure 1 (a)), and the term can be viewed to be somewhat of a misnomer, but we refer specifically to the profile in Figure 1 (b) as a min-coast seek.

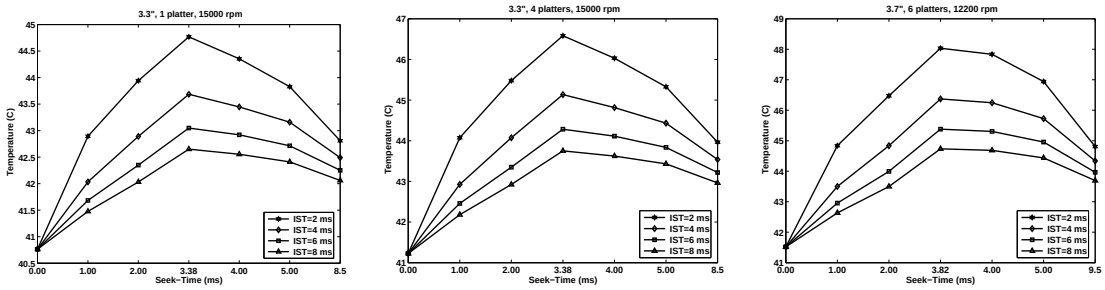
The inter-seek time is the time between the end of a seek operation and the beginning of another. If inter-seek times are short, then the dissipation of heat from inside the drive during the idle phase between any two seek operations is lower, thereby further increasing the temperature. Although a single seek operation might not create a significant change in the drive temperature, a sequence of such temporally close operations (burstiness) can have a more significant effect.

To summarize, a lower thermal profile can be achieved by one or more of the following:

- Low (possibly zero) seek times, where the acceleration/deceleration durations are low.
- Large coast times, which can possibly outweigh the effects of longer acceleration/deceleration phases.
- Large inter-seek times, allowing the disk to cool between successive accesses.

We next perform a microbenchmark study to investigate the impact of these factors on a disk's temperature. In these microbenchmarks, we vary the inter-seek time (IST) from 0 ms to 8 ms in increments of 2 ms. In addition, we also vary the total seek time by considering discrete values between 0 ms to 5 ms, in discrete steps of 1 ms. We also consider a seek time that corresponds to the min-coast value explained above (which turns out to be 3.38 ms and 3.82 ms for a 3.3" and 3.7" platter sizes respectively). For a given set of values in this two dimensional design space (inter-seek times and seek times), the microbenchmark introduces a large number of seeks (over a period of 150 minutes) of these specified parameters after initial warmup of the thermal state. We perform these experiments for specific disk configurations (platter size and number of platters) that were used in actual platforms to collect I/O traces of real workloads which will be described in section 5.

In Figure 3, we plot the temperature of some disk drive configurations for various seek-time values (x-axis) for a given inter-seek time. Note that the points in the middle of the seek-time range which usually yield the highest temperatures correspond to "min-coast". The extreme left points correspond to "zero-seek", and the ones on the extreme right correspond to "max-coast".



(a) (b) (c)

Figure 3: Relationship between seek-time and disk temperature. The highest point for each of the curves corresponds to a min-coast seek.

As expected, the temperature starts going up with non-zero seek times for a given inter-seek time. We see around 2-6 C increase in temperature when going from zero-seek to the min-coast value in these three disk configurations. The duration for which the VCM is active grows linearly with the seek time (until the min-coast value), contributing to the increase in temperature. Beyond the min-coast point, though the VCM is exercised as much in the seeks, the gap (coast) between the acceleration and deceleration phases allows the disk to cool a little, contributing to a reduction in temperature. Despite this cooling effect, the temperatures for even the full-stroke (the max-coast to the extreme right) seeks are still higher than not performing any seeks (the points to the extreme left), suggesting that seek time optimization plays an important role in thermal management as well (and not just for the traditional performance goals).

We find that the inter-seek time has an equally important effect on the thermal behavior. With temporally close (bursty) seeks, the disk does not have as much time to cool, yielding higher temperatures compared to a workload with seeks that are more temporally separated. Further, a smaller inter-seek time amplifies the effects of the individual seek activities. For instance, when we look at the curve for the 2 ms inter-seek time, in Figure 3(a), we see that if we reduce the seek-time by 1 ms from the 2 ms seek-time point, there is nearly a 1.05 C reduction in the temperature. Nearly the same reduction in temperature is also achievable by increasing the inter-seek time by 2 ms. On the other hand, when we see that for the curves with inter-seek times that are longer than 2 ms, the temperature variation becomes less sensitive to the inter-seek times but is affected more by the seek-time. For example, to reduce the temperature by 1 C for a seek-time of 2 ms and inter-seek time of 4 ms, we need to introduce an additional inter-seek time of 4 ms, whereas only about a 1 ms reduction in the seek-time is required to achieve nearly the same effect.

The rise in the temperature is faster for disks that have more platters (Figure 3(b)) or larger platters (Figure 3(c)), due to the increased viscous heating. This also makes the absolute temperature values in Figure 3(c) the highest due to the nearly fifth-power impact of platter size, and that in Figure 3(b) higher than the 1-platter configuration, since the number of platters has a linear effect on the viscous dissipation.

We have repeated these benchmarks across different disk/RPM configurations, particularly for those of interest in the latter portion of this paper. Rather than re-draw all the lines, we summarize the temperatures for the (i) zero seek, (ii) min-coast and (iii) max-coast (full-stroke seek), for the considered configurations in Figure 4. In addition, we also show the thermal envelope line (calculated to be 45.22 C using the same techniques described in [14]) below which we want the temperatures to remain. The second and third column of graphs shows the thermal profiles for successive increases in the RPM for the same platter size and number of platters shown in the first column.

When we look at the leftmost bar (where the VCM is always on) in each graph on the first column, we find

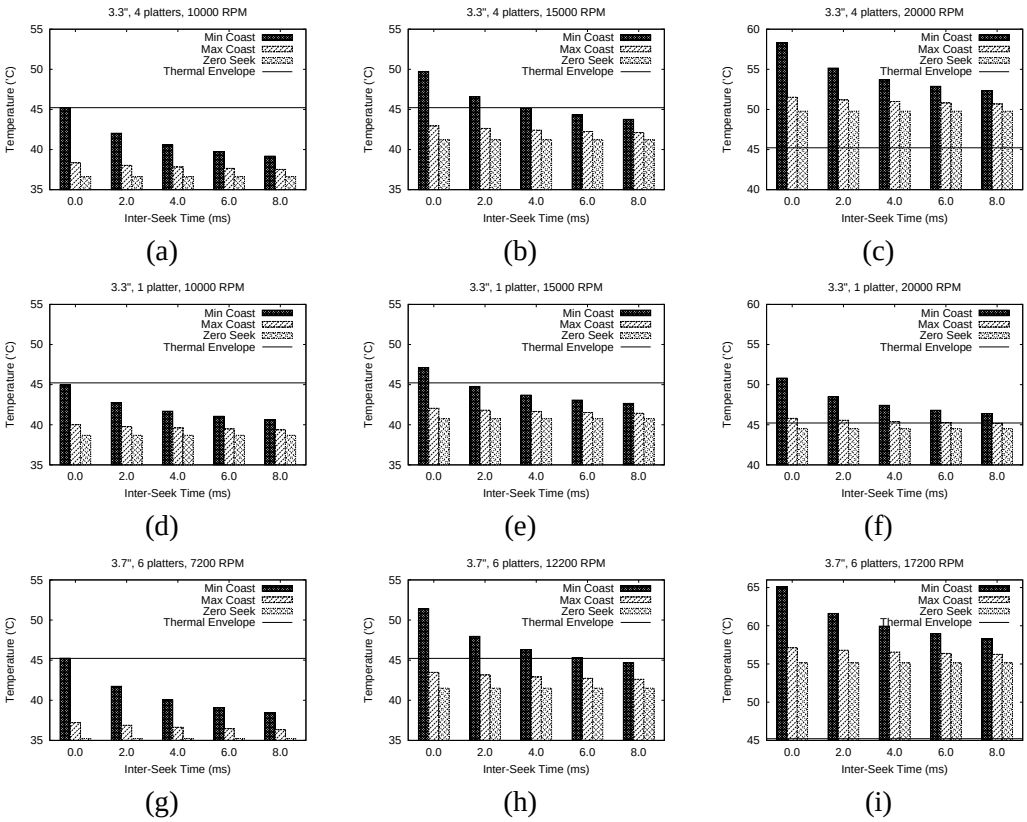


Figure 4: Results of Microbenchmark Study. Each row of graphs shows the steady-state temperature for a disk configuration for various RPMs (in increasing order from left to right). The horizontal line in each graph is the thermal envelope.

that the temperatures are very close to the thermal envelope, since the cooling system was provisioned to handle this workload scenario. These disk configurations correspond to the baseline case, i.e. they are actual product configurations of prior calendar years when the workload traces were collected. However, we observe that if there are long seek operations (higher coast time), the temperature of the disks are significantly lower than the worst-case. For instance, for the 3.3" 4-platter disk, when coast times are long, there is close to a 7 C drop in the temperature compared to the worst-case. This relative difference in the temperature is also observable for the higher RPMs.

We can also lower the temperature by having much shorter (or even zero) seeks, and the savings is much more pronounced when there is no movement of the arm at all. We see over 8 C temperature drop in the low inter-seek time experiments of the first column when we move from min-coast to zero-seek. We also find that there is a greater amount of temperature reduction for the 3.7" disk compared to the 3.3". This is because the power output of the VCM depends on the platter size and thus has a more significant impact on temperature for the 3.7" drive. When the disk seek-times are very small, increasing the inter-seek time lowers the temperature, although it has a lesser impact as observed earlier in Figure 3. When coast times are high, the inter-seek time has negligible impact on the temperature of the drives.

When we turn our attention to the second column of graphs, where the disk speeds are increased by 5000 RPM from their baseline counterparts in the first column, we find that the min-coast bars (where the VCM is always on) exceed the thermal envelope. As explained in [14], drive manufacturers assume worst-case operating conditions

(VCM always being on) when limiting their RPMs to make the disk operate within the thermal envelope. However, as we note in these graphs, there is a relatively large difference in temperature between the min-coast and the other two bars, especially at smaller inter-seek times. In fact, these two bars lie within the thermal envelope, suggesting that we can even operate at this higher RPM with an appropriate DTM scheme.

These results give us some initial observations on DTM schemes that could possibly be employed at this higher (5000 RPM boost) RPM level. As the bars reveal, three options exist for bringing the temperature down from the min-coast value: (a) bring down seek times close to zero, (b) increase coast time as much as possible, and (c) increase inter-seek times. Note that (a) and (b) are contrasting goals, and we need to keep in mind which one is more important. The above results for the 5000 RPM boost show that zero-seeks definitely give lower temperature than max-coast. Consequently, disk arm scheduling algorithms such as Shortest Positioning Time First (SPTF) can possibly serve to lower the temperature (and not just enhance performance for which it has been intended). However, if the waiting queue of requests is such that the seek distances are not necessarily that low (i.e. the thermal profile is heading more towards the min-coast region), then one may possibly opt for an inverse SPTF algorithm (i.e. Longest Positioning Time First Algorithm), since in this case we may be able to increase the coast times.

However, it is possible that we may reach points when changing the arm scheduling algorithm may not suffice to remain within the thermal envelope. The bars for the 0 ms inter-seek time in Figure 4 (f) give some evidence of this observation, where min-coast exceeds the thermal envelope as well and the zero-seek is fairly close to the envelope. Getting to the zero-seek value may not be achievable in a real workload, and in this case the DTM option may actually need to increase the inter-seek times (by introducing delays) sufficiently so that the disk may cool between successive requests. It is not clear if the performance benefits of a higher RPM become offset by these delays, unless the workload itself has temporally well-separated requests.

Finally, we notice that in the first and third rows of the last column of graphs, the 10,000 RPM increase from the baseline causes all bars to exceed the envelope. The greater amount of viscous dissipation due to the higher rotational speed (which has nearly a cubic impact) coupled with the nearly fifth-power relation to the platter size and the linear dependence of the number of platters, causes this behavior. Disk head scheduling and introducing delays are not sufficient to manage the temperature in these cases, and more aggressive techniques such as dynamic RPM modulation [12, 5] may need to be employed for DTM.

5 Thermal Behavior of Real Server Workloads

In the previous section, we identified the salient aspects of I/O behavior at the disk drive level that can affect temperature. Although this study helps us understand the relative importance of the various parameters on the drive temperature, it is important to analyze how real workloads use the disk drives within this broad space. Applications may also consist of different phases of execution, each of which might exhibit a different thermal behavior. Moreover, based on how applications utilize the disk drives, we can prioritize which physical I/O parameter needs to be the primary target of optimization for DTM.

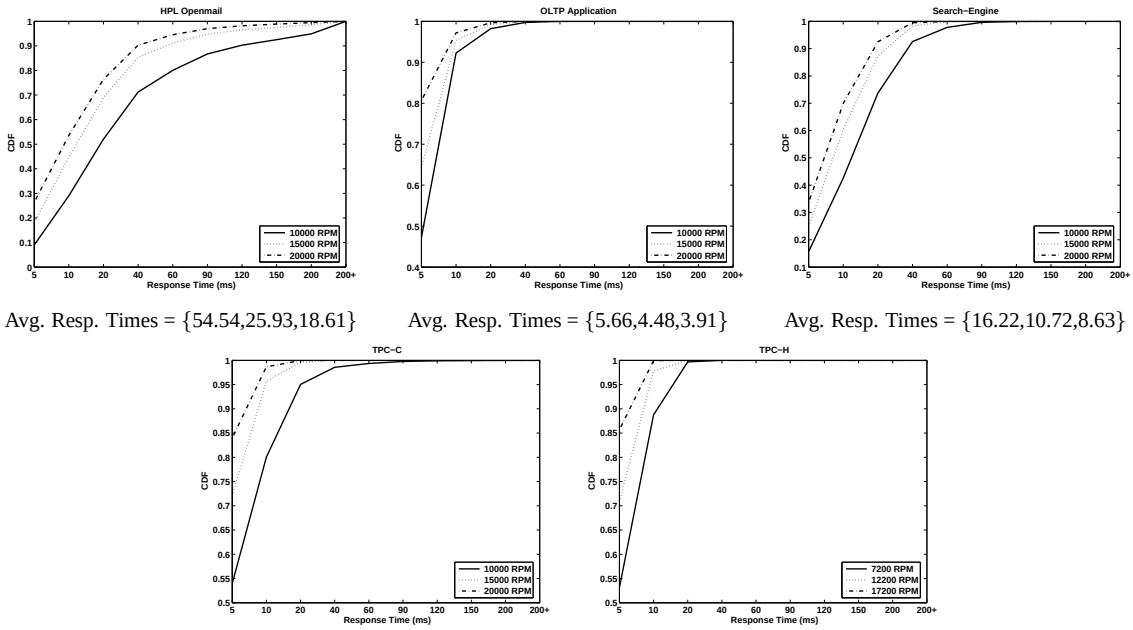
5.1 Workloads

In this paper, we use five commercial I/O traces, whose characteristics are given in Table 1, along with details of the storage system (from prior years) on which each trace was collected. Although the disks listed in the Table use platters that are larger and also lower RPMs than those used in drives today (eg. 2.6” and 15000 RPM), we tried to be as faithful as possible to the original storage system configurations used for these applications, so as not to

skew our observations. These workloads were also used in [14] to study the I/O performance benefits that higher RPMs could have provided for the disks used in their original storage systems, under the idealizing assumption that temperature is not a concern. We use these experiments as the starting point and consider the effects that temperature plays in the design and operation of such disk drives.

Workload	Year	# Requests	# Disks	Per-Disk Capacity (GB)	RPM	Platter Diameter (in)	Platters (#)	RAID ?
HPL Openmail [32]	2000	3,053,745	8	9.29	10000	3.3	1	Yes
OLTP Application [33]	1999	5,334,945	24	19.07	10000	3.3	4	No
Search-Engine [33]	1999	4,579,809	6	19.07	10000	3.3	4	No
TPC-C	2002	6,155,547	4	37.17	10000	3.3	4	Yes
TPC-H	2002	4,228,725	15	35.96	7200	3.7	6	No

Table 1: Description of workloads and storage system used.



Avg. Resp. Times = {54.54,25.93,18.61} Avg. Resp. Times = {5.66,4.48,3.91} Avg. Resp. Times = {16.22,10.72,8.63}

Avg. Resp. Times = {6.50,3.23,2.46} Avg. Resp. Times = {4.91,3.25,2.64}

Figure 5: Performance impact of faster disk drives for the server workloads. Each graph shows the CDF of the response times for each RPM. The solid curve for each graph shows the performance of the baseline system. The average response times are shown below each graph in the order of increasing RPMs.

In Figure 5 we quantify the performance for each of the workloads in their *baseline* and higher speed configurations by plotting the CDF of the response times when their respective storage systems employ the faster disks. However, we restricted the highest RPM value to 20,000 RPM, which has been shown to be feasible for reliable disk-drive operation [6]. We would like to employ DTM techniques that can bring us close to the response times of these higher RPM disks without exceeding the thermal envelope. Additionally, we also compare the performance improvement of the DTM techniques against their baseline configurations.

5.2 Thermal Profiles

Figure 6 shows the temperature of the higher RPM disks when running these workloads. For clarity, we look at the thermal profiles across three time granularities. The first column of graphs shows the profiles, for the disks of different RPMs, across the entire simulation of each workload. Again, in the interest of clarity and space, we

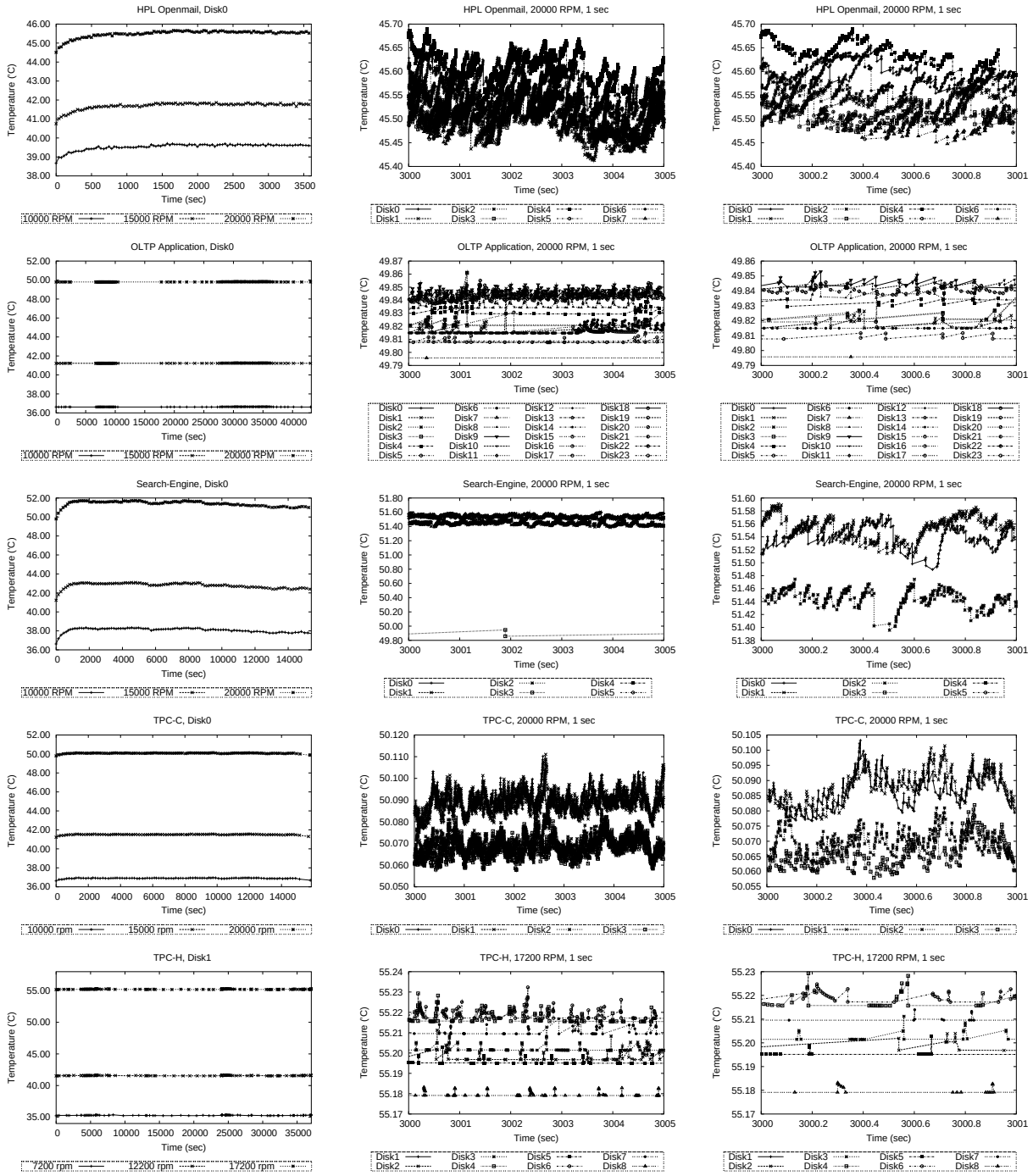


Figure 6: Thermal profiles of the workloads for three time ranges. The thermal envelope is 45.22 C. Note that the scale of the y-axes are different for each graph in order to make the temperature variations as detailed as possible.

Workload	First Most Frequent Seek-Time				Second Most Frequent Seek-Time			
	Frequency (%)	Seek-Time (ms)	Acceleration Time (ms)	Coast Time (ms)	Frequency (%)	Seek-Time (ms)	Acceleration Time (ms)	Coast Time (ms)
Openmail	46.7	1.0	0.5	0.0	18.6	2.0	1.0	0.0
OLTP	62.5	1.0	0.5	0.0	37.5	0.0	0.0	0.0
Search-Engine	29.4	1.0	0.5	0.0	19.6	2.0	1.0	0.0
TPC-C	58.4	0.0	0.0	0.0	41.6	1.0	0.5	0.0
TPC-H	56.5	1.0	0.5	0.0	43.2	0.0	0.0	0.0

Table 2: Seek-time breakdown of the applications using disks that are 5000 RPM faster than their baseline values. The deceleration time is not shown since its value is same as that for acceleration.

show the profiles only for one representative disk in the storage system for each workload. The middle column zooms into a 5 second time window starting from the fiftieth minute of the simulation and shows the temperature variation for all the disks that are employed for each storage system. The rightmost column goes for an even closer look by plotting just the first second of the 5 second period. These last two column of graphs show the profiles for all the disks that are used for each workload. All workloads are run after the system is warmed up using the methodology described in Section 3.

Let us look at the first column of graphs. We find that a 5000 RPM increase from the baseline RPM can be easily accommodated within the thermal envelope without having to increase the cooling requirements. The significance of this can be seen by looking at the performance plots in Figure 5, where a 5000 RPM increase can provide 21%-53% improvement in the response time from the baseline. The closest that we get to the thermal envelope value of 45.22 C is for the Search-Engine workload, where disk-0 experiences a maximum temperature of 43.15 C.

In order to better understand why we are still within the thermal envelope (despite boosting the disks by 5000 RPM more than what they were designed for), we first dissect the seek time of the workloads into the acceleration, coast, and deceleration components. We histogrammed these values for each workload, into bucket sizes of 1 ms granularities, and associated the value of each bucket with its upper interval. In Table 2 we show the results for the top two seek-time occurrences, since these really dominate the execution. In addition to this, we also show the probability density function (PDF) of the inter-seek times of disk-0 for the workloads in Figure 7. Each graph shows two sets of PDFs, one for the inter-seek times between any two disk seeks (denoted as “All”) and another only for the seeks that actually involve a movement of the disk arm (denoted as “Without 0-Seeks”). The latter is used to remove any bias towards the high occurrences of 0-seeks which do not increase the temperature despite coming temporally close to each other.

From the Table, we see that the bulk of the seeks, across all the workloads, have a duration of 0 ms or 1 ms. Only the Search-Engine and Openmail workloads show seeks that are of duration 2 ms and higher. As we saw in the previous section, if the time taken for a seek is around this 1 ms value, then the heat that is generated is much less than for a min-coast seek. However, the actual temperature of the disk also depends on the inter-seek time value, which is shown in Figure 7. For the OLTP and TPC-H applications, many of the inter-seek times are quite long, especially for the former, where they are in the order of several hundreds of milliseconds. However, between these workloads, we find that TPC-H runs a little cooler than OLTP, despite the latter experiencing about an order of magnitude larger inter-seek times for the majority of the seeks. This is due to the seek-time behavior of the two workloads (shown in Table 2). The vast majority of the seeks in both these workloads are of 0 ms and 1 ms in duration. However, TPC-H is composed of a larger proportion of zero-seek operations compared to OLTP (i.e. there is very good spatial locality in TPC-H). As we saw in Section 4, when inter-seek times are greater than 2 ms, the relative temperature differences for larger values of the inter-seek time become progressively smaller. However, the seek-time still has a strong impact on temperature, especially in the region that is less than the

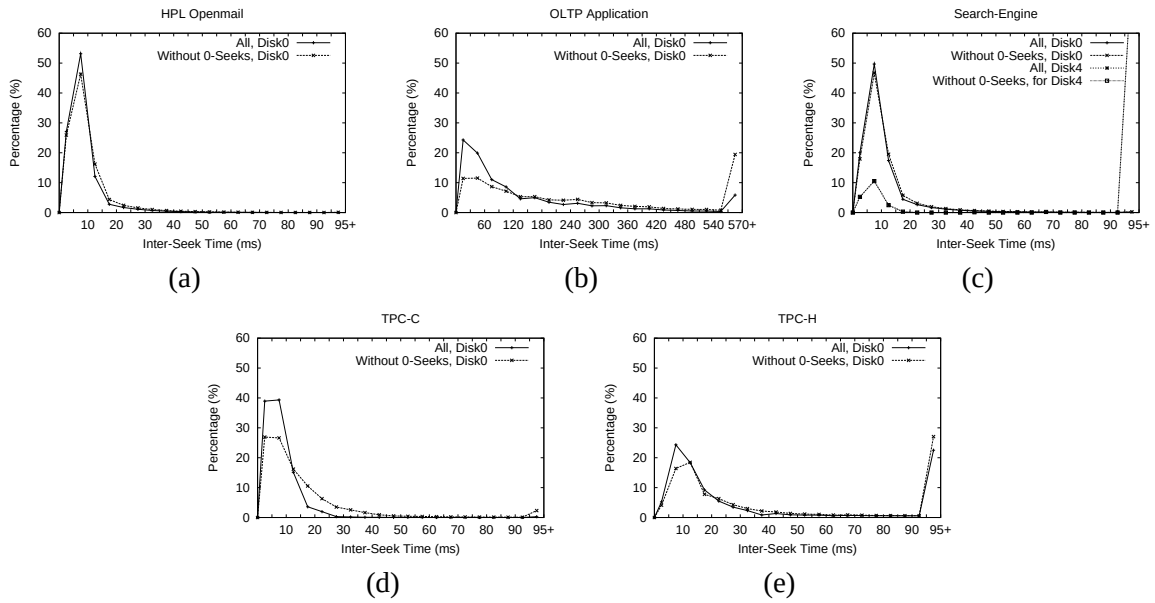


Figure 7: Probability Density Function (PDF) of the inter-seek times for the workloads. For each workload, the curve labeled "All" includes disk requests that do not involve a movement of the arm (zero-distance seeks) and "Without 0-Seeks" shows the inter-seek times only between seek operations that actually involve a movement of the arm.

min-coast value. As TPC-H has about 6% more 0 ms seeks than OLTP, its disks experience a lower temperature.

TPC-C shows a temperature profile that is somewhat similar to OLTP but exhibits a different set of characteristics. The bulk of the seek-times in this workload are of 0 ms in duration and the remaining being 1 ms. In particular, we can see from Table 2 that its seek-time distribution is quite comparable to the TPC-H workload. However, there are a significant number of inter-seek times (around 30%) that are less than 10 ms (the "Without 0-Seek" curve in Figure 7(d)). This would cause the temperature to be higher than TPC-H. However, as we have already seen, the differences in the inter-seek time do not play a very dominant role, except for very short values, making the temperature only slightly higher than TPC-H.

The Openmail and Search-Engine workloads exhibit a larger variation in seek-times. There is also significant variation in the inter-seek times between different disks for the Search-Engine workload (as shown in Figure 7(c) for disks 0 and 4). We find that 6.5% of the seeks in Openmail take between 2-3 ms and none between 3-4 ms. 4.7% of the seek operations in the Search-Engine workload have times between 2-3 ms and 14.8% of them are in the range of 3-4 ms. These two workloads also have half their inter-seek times in the 10 ms range. Recall that, as the seek-time increases (until we reach the min-coast point), the acceleration that is required increases as well, causing each seek operation to generate more heat. This phenomenon is observable for these two workloads, where the constituent disks in their respective storage systems experience the highest temperatures for 15,000 RPM. Between these two workloads, Search-Engine has a higher absolute temperature because it uses 4-platter disks (as shown in Table 1), which generates more heat (by a linear factor) than the 1-platter units used by Openmail.

Although we consider both 3.3" and 3.7" disks, we provision sufficient cooling such that all the drives satisfy the thermal envelope of 45.22 C in their baseline configurations. Search-Engine uses 3.3" 4-platter 10,000 RPM disks in its baseline configuration whereas TPC-H uses 3.7" 6-platter 7200 RPM drives. We found that these two configurations are not exactly equivalent from the thermal viewpoint in the sense that the latter can generate more

heat than the former, requiring the outside temperature to be slightly cooler. Therefore, if we increase the RPM by 5000 from their baseline configurations without altering the cooling system, we might expect the heating to be higher for the 3.7" disk by virtue of its larger platter size and number of platters. However, we have seen that the disks used by Search-Engine experience higher temperatures than TPC-H. In fact, the highest temperature that any disk in TPC-H reaches for 12,200 RPM is 41.86 C, compared to 43.15 C for Search-Engine. This is again due to the same factors outlined above, namely, very short seek-times and large idleness, both of which are application dependent.

When we increase the RPM by a further 5000, we find that all the curves are now above the thermal envelope. As we had seen in the microbenchmark evaluation, the highest temperatures are now experienced by the 3.7" disks used in TPC-H. This was observed in the microbenchmark result in Figure 4(i), where even with inter-seek times of 8 ms and maximum coasting, the temperature of a 3.7" disk is higher than those of the other two, across all the chosen values for the inter-seek time and coast-time. This change in the thermal behavior from lower speeds is because the RPM is now high enough such that it is now the most dominant determinant of the overall drive heat. Although there is still some variation in temperature with workload behavior, even the idle operating temperature is significantly (more than 10 C) above the thermal envelope. Similar trends are observable for the other workloads as well and most of them operate roughly 5 C above the thermal envelope. Since even a 5 C variation in temperature above the thermal envelope can significantly affect reliability [1], it is imperative to apply a DTM technique to manage its temperature.

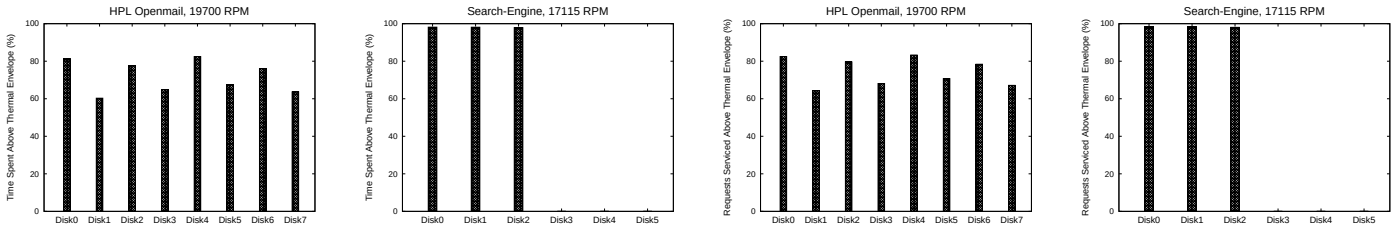
From the above results, we note the following important observations across the workloads:

- The seek-times are significantly lower than the min-coast value (with a significant number that do not need an arm movement at all), reducing the amount of time that the VCM needs to be on. Even if there are considerable short inter-seek times (over 50% lower than 10 ms in many workloads), the short (or zero-distance) seeks keep the disk cool enough even when there is a 5000 RPM boost from the baseline. This is achievable with neither an alteration of the cooling system nor by the use of any DTM technique. This is a rather powerful observation since we are pronouncing that the disk could have been provisioned with the 5000 RPM boost statically and we would have never exceeded the thermal envelope (and gained between 21-53% response time improvement).

- Going for another 5,000 RPM boost does cause the results to violate the thermal envelope especially in the disks with higher and larger platters, regardless of the workload.

- If we do decide to incorporate any DTM (say in the case of Openmail with a 10000 RPM boost from the baseline), our results also give insights on how we should go about it. With most of the seek-times falling less than 2 ms, SPTF is good enough for a thermal management strategy (we do not need to amplify the coast times). Rather, extending inter-seek times (by possibly introducing delays) can be more rewarding. This is investigated in more detail in the next section.

- Finally, at high enough RPMs, one can simply not sustain the workloads without exceeding the thermal envelope. In these cases, neither seek-time optimizations nor delays between requests may be viable DTM choices. One may need to opt for more extensive DTM techniques such as dynamic RPM modulation, and we leave it to future work to explore such issues.



(a) Time (b) Requests

Figure 8: Percentage of Workload Time and Requests Serviced above the thermal envelope

6 Feasibility of Delay-based DTM

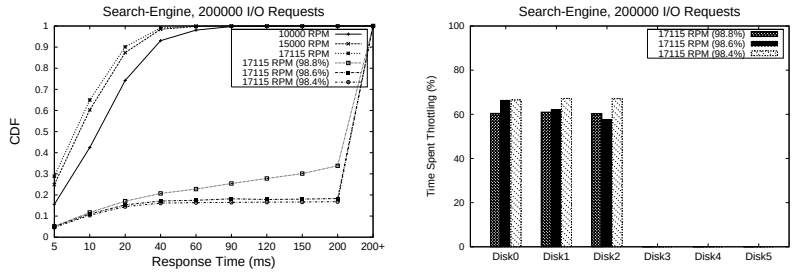
As we observed, the 5000 RPM boost to the baseline is still able to maintain the workload execution within the thermal envelope. At the other extreme of large RPMs, the temperature would exceed the envelope even without any seek activity. The question which we then try to answer is whether there is an intermediate RPM range where DTM to control I/O activities of a constant RPM disk can keep the temperature under control while enhancing performance. In this section, we examine a simple delay-based DTM technique for this region to extend inter-seek times.

We have already defined the notion of *thermal envelope*, which is the highest permissible operating temperature of the disk. We introduce two additional terminologies - *thermal emergency* and *thermal safety* - to describe our delay strategy. The thermal emergency is the temperature at which the DTM mechanism is invoked, when getting close to the thermal envelope. After this point, we do not serve any requests and they are queued up until the temperature falls to the *thermal safety* value, at which time normal operation resumes. It is to be noted that the strategy is non-preemptive in that any request being processed is serviced before the operations are temporarily suspended. There are several trade-offs for the choice of temperature values for the emergency and safety points, and we show results below with a value of 99% of the envelope for thermal emergency, and values of 98.8%, 98.6% and 98.4% of the envelope for thermal safety. We also chose an RPM that is high enough that the thermal profile of the execution hovers on both sides of the envelope. Further, in the interest of space, we focus mainly on the Openmail and Search-Engine workloads. The other workloads do not need DTM even for these higher RPMs, and are able to remain within the envelope because of their seek behaviors explained earlier .

In Figure 8 we quantify the percentage of the execution, in terms of time and the number of serviced requests, performed above the thermal envelope for these two workloads. We observe that the two workloads exhibit distinct thermal behaviors. In Openmail, the time spent above the thermal envelope is similar across the disks (since it is configured as RAID-5), while in Search-Engine we find the thermal envelope violations are only in half the disks. Load balancing across the disks and/or having heterogeneous (RPM) disks [13, 5] could be possible DTM methods for this workload. Further, we note that while the goal of energy optimization may favor clustering the data into a minimal set of disks, so that the rest can be powered down [26], the goals for temperature management may be contradictory.

Although we see that a substantial amount of time is spent above the thermal envelope, it is possible that a portion of this time is spent idling. It is therefore necessary to look at the fraction of I/O requests that are serviced above the thermal envelope. We observe a nearly one-to-one correspondence between the time and the number of requests, mandating the employment of a DTM technique for this execution.

We now apply the simple delay-based DTM technique which stops servicing requests upon reaching the thermal emergency, until the temperature drops to thermal safety, for the Search-Engine workload (results for



(a) (b)

Figure 9: Impact of the Delay DTM technique for the Search-Engine workload

Openmail are similar). Since these simulations take an excessive amount of time, we are showing the results for the first 200,000 requests. Figure 9(a) shows the CDF of the response times using this technique for the three values of thermal safety (98.8%, 98.6%, and 98.4% of the thermal envelope respectively). We also plot the response times of the baseline configuration which uses 10000 RPM disks, a configuration that uses disks whose speeds are 5000 RPM higher than the baseline but not needing DTM, and also disks of 17115 RPM which require DTM but we plot them anyway without applying DTM as a reference point to compare the degradation in performance. We can observe that even though in the idealized case, the 17115 RPM can give very good performance (if no DTM is applied), the application of the DTM severely degrades performance wherein a majority of the requests have response times in excess of 200 ms (compared to 40-60 ms even for the 10000 RPM disk). This is because the delay (throttling) is being done very frequently as is shown in Figure 9(b), which has a very disruptive effect on the normal activities of the disk drive. For instance, in disk-0, more than 60% of the time is spent in just throttling (delay) rather than serving requests.

The reason for the inapplicability of the delay based technique in these workloads can be explained as follows. The I/O requests in the workloads are coming in such a way that the temperature variances around the thermal envelope are rather small. This is because the load is itself not very different over time to cause much variation in the temperature. Consequently, whenever a delay is introduced, it affects some requests that are queued up. The moment we get back to the thermal safety point, we again have a burst of activity to service the queued up and newly arriving requests, causing the disk to heat up again leading to the next thermal emergency fairly soon. On the other hand, if the workload showed more thermal variances, then this technique would be more promising. This would require the load to be more temporally varying e.g. alternating phases of high and low activity. Other techniques include providing more disks, as either just a bunch of disks or as a RAID, than these baseline configurations, which can introduce longer idle periods when the throttling can be applied without significantly affecting requests.

It should be noted that there is only a narrow band of RPMs that can ever be applied, when even if this delay based request throttling is feasible. As was observed, at small RPM boosts the workload is still within the thermal envelope without requiring any DTM. At much higher RPMs, the idling, but spinning, disks would themselves exceed the envelope regardless of servicing requests. As our above results show, even in the narrow band where the delay technique may be applicable, the workload characteristics are not very conducive for applying this technique in isolation. All these observations further motivate the need for DTM techniques which can exploit multi-speed disks, where a lower RPM can be useful to reduce the temperature. Since RPM has nearly cubic relation to temperature, its modulation can cool the disk faster providing more scope for DTM. Using such DTM techniques on versions of multi-speed disks already available in the market (such as [17]) can thus delay fewer requests, while a more extensive full fledged DRPM disk [12] which can service requests at different RPM levels

would at best slow down the service times during these cool down periods. Exploring these issues is part of our future work.

7 Concluding Remarks

This paper has presented the first integrated performance-thermal simulator that can be used to study the temperature of disks executing real workloads. This infrastructure requires a careful modeling of the details of a seek, and we have shown how to account for the heat generated in the acceleration/deceleration phases, and the coast in-between when the disk can possibly cool down. The simulator integrates a discrete-event performance model (Disksim) with a time-step thermal model.

Using this simulator we have conducted detailed microbenchmark studies to understand the temperature relationship to disk level I/O activities. We point out several options for temperature management - reducing seek distances, amplifying coast times, and temporal spacing between seeks - which can be applied even on existing disks. With five real commercial traces, we show that one can obtain a 5000 RPM boost without having to resort to any explicit thermal management. Above this level, we need to employ DTM to stay below the thermal envelope.

As our results indicate, the DTM techniques at higher RPMs are somewhat limited by what they can achieve, if the underlying RPM is rigid. However, there could be a wide spectrum of opportunities to exploit if we had the luxury of multi-speed disks (whether it be similar to what is available in the market today [17] or a full-fledged dynamic RPM modulation disk as suggested in [12, 5]). We intend to investigate these possibilities in future work.

References

[1] D. Anderson, J. Dykes, and E. Riedel. More Than An Interface - SCSI vs. ATA. In *Proceedings of the Annual Conference on File and Storage Technology (FAST)*, March 2003.

[2] K. Aruga. 3.5-Inch High-Performance Disk Drives for Enterprise Applications: AL-7 Series. *Fujitsu Science and Technology Journal*, 37(2):126–139, December 2001.

[3] K.G. Ashar. *Magnetic Disk Drive Technology: Heads, Media, Channel, Interfaces, and Integration*. IEEE Press, 1997.

[4] D. Brooks and M. Martonosi. Dynamic Thermal Management for High-Performance Microprocessors. In *Proceedings of the International Symposium on High-Performance Computer Architecture (HPCA)*, pages 171–182, January 2001.

[5] E.V. Carrera, E. Pinheiro, and R. Bianchini. Conserving Disk Energy in Network Servers. In *Proceedings of the International Conference on Supercomputing (ICS)*, June 2003.

[6] S.X. Chen, Q.D. Zhang, H.C. Chong, T. Komatsu, and C.H. Kang. Some Design and Prototyping Issues on a 20000 RPM HDD Spindle Motor with a Ferro-Fluid Bearing System. *IEEE Transactions on Magnetics*, 37(2):805–809, March 2001.

[7] N.S. Clauss. A Computational Model of the Thermal Expansion Within a Fixed Disk Drive Storage System. Master's thesis, University of California, Berkeley, 1988.

[8] F. Douglis and P. Krishnan. Adaptive Disk Spin-Down Policies for Mobile Computers. *Computing Systems*, 8(4):381–413, 1995.

[9] P.A. Eibeck and D.J. Cohen. Modeling Thermal Characteristics of a Fixed Disk Drive. *IEEE Transactions on Components, Hybrids, and Manufacturing Technology*, 11(4):566–570, December 1988.

[10] G.R. Ganger, B.L. Worthington, and Y.N. Patt. *The DiskSim Simulation Environment Version 2.0 Reference Manual*. http://www.ece.cmu.edu/ ganger/disksim/.

[11] E. Grochowski and R.D. Halem. Technological Impact of Magnetic Hard Disk Drives on Storage Systems. *IBM Systems Journal*, 42(2):338–346, 2003.

[12] S. Gurumurthi, A. Sivasubramaniam, M. Kandemir, and H. Franke. DRPM: Dynamic Speed Control for Power Management in Server Class Disks. In *Proceedings of the International Symposium on Computer Architecture (ISCA)*, pages 169–179, June 2003.

[13] S. Gurumurthi, A. Sivasubramaniam, M. Kandemir, and H. Franke. Dynamic Speed Control for Server Class Disks. Technical Report CSE-03-007, The Pennsylvania State University, March 2003.

[14] S. Gurumurthi, A. Sivasubramaniam, and V. Natarajan. Disk Drive Roadmap from the Thermal Perspective: A Case for Dynamic Thermal Management. In *Proceedings of the International Symposium on Computer Architecture (ISCA)*, pages 38–49, June 2005.

[15] S. Gurumurthi, J. Zhang, A. Sivasubramaniam, M. Kandemir, H. Franke, N. Vijaykrishnan, and M.J. Irwin. Interplay of Energy and Performance for Disk Arrays Running Transaction Processing Workloads. In *Proceedings of the International Symposium on Performance Analysis of Systems and Software (ISPASS)*, pages 123–132, March 2003.

[16] G. Herbst. IBM's Drive Temperature Indicator Processor (Drive-TIP) Helps Ensure High Drive Reliability. In *IBM Whitepaper*, October 1997.

[17] Hitachi Power and Acoustic Management - Quietly Cool. In *Hitachi Whitepaper*, March 2004. http://www.hitachigst.com/tech/techlib.nsf/productfamilies/White_Papers.

[18] H. Ho. Fast Servo Bang-Bang Seek Control. *IEEE Transactions on Magnetics*, 33(6):4522–4527, November 1997.

[19] I. Hong and M. Potkonjak. Power Optimization in Disk-Based Real-Time Application Specific System s. In *Proceedings of the International Conference on Computer-Aided Desi gn (ICCAD)*, pages 634–637, November 1996.

[20] R.F. Huang and D.L. Chung. Thermal Design of a Disk-Array System. In *Proceedings of the InterSociety Conference on Thermal and Thermomechanical Phenomena in Electronic Systems*, pages 106–112, May 2002.

[21] H. Levy and F. Lessman. *Finite Difference Equations*. Dover Publications, 1992.

[22] K. Li, R. Kumpf, P. Horton, and T.E. Anderson. Quantitative Analysis of Disk Drive Power Management in Portable Computers. In *Proceedings of the USENIX Winter Conference*, pages 279–291, 1994.

[23] C. Nicholson. Improved Disk Drive Power Consumption Using Solid State Non-Volatile Memory. In *Proceedings of the Windows Hardware Engineering Conference (WinHEC)*, May 2004.

[24] A. E. Papathanasiou and M. L. Scott. Energy Efficient Prefetching and Caching. In *Proceedings of the USENIX Annual Technical Conference*, June 2004.

[25] D. Patterson, G. Gibson, and R. Katz. A Case for Redundant Arrays of Inexpensive Disks (RAID). In *Proceedings of ACM SIGMOD Conference on the Management of Data*, pages 109–116, June 1988.

[26] E. Pinheiro and R. Bianchini. Energy Conservation Techniques for Disk Array-Based Servers. In *Proceedings of the International Conference on Supercomputing (ICS)*, June 2004.

[27] Samsung Electronics Develops Solid State Disk Using NAND Flash Technology, May 2005. http://www.samsung.com/PressCenter/PressRelease/PressRelease.asp?seq=20050523_0000123980.

[28] L. Shang, L-S. Peh, A. Kumar, and N.K. Jha. Thermal Modeling, Characterization and Management of On-Chip Networks. In *Proceedings of the International Symposium on Microarchitecture (MICRO)*, pages 67–78, December 2004.

[29] R.K. Sharma, C.E. Bash, C.D. Patel, R.J. Friedrich, and J.S. Chase. Balance of Power: Dynamic Thermal Management for Internet Data Centers. *IEEE Internet Computing*, 9(1):42–49, January 2005.

[30] K. Skadron, M.R. Stan, W. Huang, S. Velusamy, K. Sankaranarayanan, and D. Tarjan. Temperature-Aware Microarchitecture. In *Proceedings of the International Symposium on Computer Architecture (ISCA)*, pages 1–13, June 2003.

[31] J. Srinivasan and S.V. Adve. Predictive Dynamic Thermal Management for Multimedia Applications. In *Proceedings of the International Conference on Supercomputing (ICS)*, pages 109–120, June 2003.

[32] The Openmail Trace. http://tesla.hpl.hp.com/private software/.

[33] UMass Trace Repository. http://traces.cs.umass.edu.

[34] R. Viswanath, V. Wakharkar, A. Watwe, and V. Lebonheur. Thermal Performance Challenges from Silicon to Systems. *Intel Technology Journal*, Q3 2000.

[35] J. Zedlewski, S. Sobti, N. Garg, F. Zheng, A. Krishnamurthy, and R. Wang. Modeling Hard-Disk Power Consumption. In *Proceedings of the Annual Conference on File and Storage Technology (FAST)*, March 2003.

[36] Q. Zhu, F.M. David, C. Devraj, Z. Li, Y. Zhou, and P. Cao. Reducing Energy Consumption of Disk Storage Using Power-Aware Cache Management. In *Proceedings of the International Symposium on High-Performance Computer Architecture (HPCA)*, Febuary 2004.

[37] Q. Zhu, A. Shankar, and Y. Zhou. PB-LRU: A Self-Tuning Power Aware Storage Cache Replacement Algorithm for Conserving Disk Energy. In *Proceedings of the International Conference on Supercomputing (ICS)*, June 2004.