What is the Future of Disk Drives, Death or Rebirth?

YUHUI DENG
Department of Computer Science, Jinan University,
Guangzhou, 510632, P. R. China

Disk drives have experienced dramatic development to meet the performance requirements, since the IBM 1301 disk drive was announced in 1961. However, the performance gap between memory and disk drives has widened to 6 orders of magnitude and continues to widen by about 50% per year. Furthermore, energy efficiency has become one of the most important challenges in designing disk drive storage systems. The architectural design of disk drives has reached a turning point which should allow their performance to advance further, while still maintaining high reliability and energy efficiency. This paper explains how disk drives have evolved over five decades to meet challenging customer demands. First of all, it briefly introduces the development of disk drives, and deconstructs disk performance and power consumption. Secondly, it describes the design constraints and challenges that traditional disk drives are facing. Thirdly, it presents some innovative disk drive architectures discussed in the community. Fourthly, it introduces some new storage media types and the impacts they have on the architecture of the traditional disk drives. Finally, it discusses two of the important evolutions of the disk drives: hybrid disk and solid state disk. The paper highlights the challenges and opportunities facing these storage devices, and explores how we can expect them to affect the storage systems.

Categories and Subject Descriptors: C [Computer Systems Organization]: C. 0 [General] –System architectures

General Terms: Design

Additional Key Words and Phrases: Disk drive, flash memory, architecture, energy efficiency and thermal envelope

1. INTRODUCTION

Disk drives have become the most important persistent storage that offers high performance, large capacity, and high reliability. They are the major storage devices used in highly dynamic and ever changing computing environments. Since the IBM 1301 disk drive was announced in 1961, disk drives have experienced dramatic development to meet the capacity, performance, and other capability requirements.

The evolution of magnetic recording technology has experienced two important milestones: longitudinal recording and perpendicular recording. Over the last decade, the magnetic recording has successfully achieved 100% growth of Areal Density (AD) attributing to the traditional longitudinal recording technology, which results in 30% growth of Linear Density (LD), and 50% growth of Track Density (TD), respectively [Hitachi 2009]. However, the superparamagnetic effect poses a serious challenge for further increases of the AD. The reason is that each bit cell in a track is composed of multiple magnetic grains. The size or the number of magnetic grains in a bit cell has to be

Authors' addresses: Yuhui Deng, Department of Computer Science, Jinan University, Guangzhou, 510632, P. R. China; email: tyhdeng@jnu.edu.cn, or yuhuid@hotmail.com

decreased in order to increase the AD. Unfortunately, the grain size cannot be scaled much below a diameter of ten nanometres due to the superparamagnetic effect. Otherwise, because the signal energy stored in the grain can drop below the ambient thermal energy, the magnetic grains would become unstable. Using a recording medium which requires a stronger field to change the state of the bits is one way to overcome this, but this method challenges the design of disk write head. Using fewer magnetic grains in a bit cell results in lower signal to noise ratio, which requires more complicated error correcting codes[Gurumurthi et al. 2005]. Disk drives with longitudinal recording can obtain an estimated limit of 100 to 200 gigabits per square inch due to the superparamagnetic effect, though this estimate is constantly changing. Perpendicular recording is supposed to achieve much higher recording density. It is predicted that the recording density will continue to increase to 600 gigabits per square inch in 2009 to 2010, 1.2 terabits per square inch in the second half of 2011 to 2012, and 2.4 terabits per square inch in 2013 to 2014 [Nezu 2009; Perpendicular recording 2009]. Over the past two decades, the performance of disk drives has been experiencing 40% growth per year. The growth mainly depends on the improvement of Revolutions Per Minute (RPM), magnetic recording technology, the size of the on-board cache, along with some reductions in the seek time [Hitachi 2009].

The hierarchy of storage in current computer architectures is designed to take advantage of data access locality to improve overall performance. Each level of the hierarchy has higher speed, lower latency, and smaller size than lower levels. Although the performance of disk drives has achieved significant growth, the performance gap between the RAM and disk drives in the storage hierarchy has widened to 6 orders of magnitude in 2000 and continues to widen by about 50% per year [Schlosser et al. 2000]. In the past decades, many research efforts have gone into exploring how to optimize the disk drive performance to alleviate the gap. However, due to the highly complex and dynamic feature, the disk I/O subsystem has been repeatedly identified as a major bottleneck to system performance in many computing systems.

In the past 50 years, the disk drive architecture remained largely unchanged. Industry has been placing pressure on the evolution of disk drives for a few years. Recent industry and academic research suggests a shift in storage technology. Its goal is a technology that provides persistent storage with high performance, large capacity, high reliability, and competitive price. Trends of recent years imply that flash memory could be a good candidate for bridging the performance gap. Flash based Solid State Disk (SSD) has been widely used across mobile devices, computers, servers, and high-end storage systems [Lawton 2006; EMC 2009]. However, in contrast to the traditional disk drives, flash based storage devices are still relatively expensive, and their specific characteristics such as endurance cycles and erase before write are still challenging problems.

This paper provides a comprehensive survey on the evolution of storage devices from the traditional disk drives to hybrid disk and SSD. The remainder of the paper is organized as follows. Overview of the traditional disk drives is introduced in Section 2. Section 3 describes the design constraints and challenges of the next generation disk drives. Some innovative disk drive architectures are presented in Section 4. Some new storage media types and devices including hybrid disk drive and SSD are depicted in Section 5. The design challenges of hybrid disk and SSD are highlighted in Section 6. Section 7 concludes the paper with remarks on the contributions of the paper.

2. OVERVIEW OF DISK DRIVES

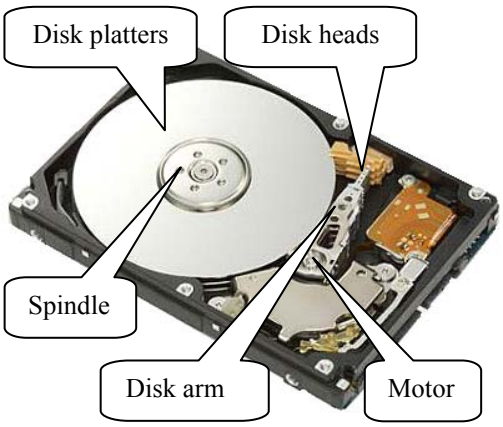

Fig. 1. Disk drive architecture

Fig.1 shows the traditional disk drive architecture. It mainly consists of platters, spindle, disk arm, disk head, motor, controller, etc. The platters spin at a constant rate called RPM. The data is recorded magnetically in concentric tracks on the platters. Some platters permit data to be recorded on both sides. The disk arm is driven by the motor to move the disk heads to a specific track. The disk controller performs mappings between the incoming logical addresses and the physical disk addresses that store the data, runs the track-following system, transfers data between the disk drive and its client (actually, the signals read by a disk head are converted by the disk controller, and then transmitted over the peripheral bus), and manages an embedded cache.

2.1 Performance overview

Disk access time (T_{access}) is mainly composed of seek time (T_{seek}), rotational latency (T_{rotate}) and data transfer time ($T_{transfer}$) [Deng 2009]. The seek time measures the time for the disk head to move to a specified track. When the disk head arrives at the required track, the time spent on rotating the required sector to appear underneath the disk head is called rotational latency. The data transfer time is the amount of data divided by the data transfer rate. T_{access} is expressed as follows:

$$T_{access} = T_{seek} + T_{rotate} + T_{transfer} \quad (1)$$

2.1.1 Seek time

The disk head is driven by a Voice-Coil Motor (VCM) to move over the recording surface to seek a target track. In order to reduce the heat dissipation of the VCM, the temperature of the coil in the VCM is controlled by selecting a fixed maximum current for the seek distances which exceed a threshold. The threshold is typically 35% of a full stroke [Seek distance 2009]. For a long seek distance which exceeds the threshold, the current in the coil reaches the maximum value when the disk head reaches a nominal maximum velocity. At the end of the acceleration period, the current is removed from the

coil, which incurs a coast period that maintains the nominal maximum velocity. Then, the fixed maximum current is applied to the coil in an opposite direction to decelerate the disk head. When the disk head reaches the target track, a procedure is triggered to verify the current position. The time cost for the disk head to settle at the end of a seek is called settling time. Therefore, a seek time is composed of an acceleration time, t_{acc}, a coast period, t_{coast}, a deceleration time, t_{dec}, and a head settling time, t_{settle}. The acceleration time and deceleration time are proportional to the square root of the seek distance. The cost period is linear in the seek distance. For a short seek (e.g. single cylinder seek), the disk arm accelerates and decelerates without reaching the nominal maximum velocity.

According to the above discussion, for the seek distances which are shorter than the threshold, the disk heads will never reach the nominal maximal velocity. This indicates that in this scenario there is no cost period no matter how fast a VCM is. The average seek time is generally taken to be the average time needed to seek between two random tracks on the disks which is normally called average seek distance $D_{average}$. The $D_{average}$ for a large number of random seeks is equal to a seek across 1/3 of the data zone which is shorter than the threshold. Therefore, the coast time of average seeks is zero. Consequently, we have an acceleration phase followed immediately by a deceleration phase [Kim et al. 2006]. This can be described as, $D_{average} = (1/2) \times a_{acc} \times t_{acc}^2 + (1/2) \times a_{dec} \times t_{dec}^2$, where a_{acc} is the acceleration which is equal to the deceleration (a_{dec}), and the acceleration time (t_{acc}) is equal to the deceleration time (t_{dec}). We assume that $a_{acc} = a_{dec} = a$, then we have:

$$t_{acc} = t_{dec} = \sqrt{\frac{D_{average}}{a}} \tag{2}$$

The average seek time T_{seek} is computed with the following equation:

$$T_{seek} = 2 \times \sqrt{\frac{D_{average}}{a}} + t_{settle} \tag{3}$$

For random small requests, seek time is a major component of disk access time, because the settling time dominates the overall short seeks and the settling time has remained largely constant [Akyürek and Salem 1995]. However, over the last decade, the areal density has achieved 100% growth. This has resulted in 50% growth of track density measured in Tracks Per Inch (TPI), and 30% growth of linear density measured in Bits Per Inch (BPI) [Hitachi 2009]. Due to the increased BPI, there are more sectors on a track, which means more sectors in a cylinder if the number of disk heads is not changed [Ng 1998]. Within a certain range of data, a bigger cylinder impacts seek time in two ways. First, it increases the probability of reducing the number of seeks. When dealing with a certain amount of data, having a bigger cylinder raises the probability that the next data request will be satisfied in the current cylinder, thus avoiding a seek completely. Secondly, it reduces the seek distance. If the size of each cylinder is increased, then an equal amount of data will occupy fewer cylinders compared with before. As a result, the seek distance is decreased. Both impacts result in shorter seek time in terms of equation (3). Though the seek time has been decreased significantly due to the increasing BPI, the

potential is being hindered by the settling time, because as seek distance decreases, the settling time (t_{settle}) becomes a relatively more important portion.

Lumb et al.[2000] investigated the impact of seek time, rotational latency and data transfer time that add up to 100% of the disk head utilization for five modern disk drives which were sold on the market from 1996 to 1999. The investigation indicated that the faster seek of the Cheetah 18LP (average seek time 5.2ms), relative to the Cheetah 9LP and Cheetah 4LP which have average seek time of 5.4ms and 7.7ms respectively, resulted in lower seek components. The results also showed that as the request size of the random workload increased, larger request size yielded larger media transfer component, and reduced the seek and rotational latency components by amortizing larger transfer over each positioning step.

2.1.2 Rotational latency

The rotational latency depends on the RPM and the number of sectors that must pass underneath the disk head. Traditionally, when the disk head arrives at the target track, it must wait for the disk platters to rotate until it reaches the first sector of the request before it begins to transfer data. The amount of time it takes for the required sector to appear underneath the disk head is called rotational latency. If the disk head settles above a sector which is one of the required sectors but not the first one, it will incur almost one revolution to reach the first sector. Average rotational latency is generally calculated as half the time it takes the disk to do one revolution. We have:

$$T_{rotate} = \frac{1}{2} \times \frac{60 \times 10^3}{RPM} \tag{4}$$

Zero-latency access, which is a new feature of modern disk drives, can start transferring data when the disk head is positioned above any of the sectors in a request. If multiple contiguous sectors are required to be read, the disk head can read the sectors from the media into its buffer in any order with zero-latency access support. The sectors in the buffer are assembled in ascending Logical Block Number (LBN) order and sent to the host. If exactly one track is required, the disk head can begin reading data as soon as the seek is completed. It involves no rotational latency because all sectors on the track are needed. The same concept applies to writes with a reverse procedure which moves the data from host memory to the disk cache before it can be written onto the media [Schindler et al. 2002]. Therefore, the rotational latency decreases with the growth of the useful blocks in a track.

2.1.3 Data transfer time

Data transfer time is the amount of data divided by data transfer rate. This consists of two parts. The first part is external data rate adopted to measure the transfer rate between memory and disk cache. The second part is employed to measure the transfer rate between disk cache and disk storage media, this part is called Internal Data Rate (IDR). Due to the mechanical components in disk drives, the IDR is much lower than the external data rate. Generally, the IDR is employed to measure the data transfer rate of disk drives because it is raw transfer rate. The IDR depends on the combination of BPI and RPM. The BPI indicates how many bits can be stored on a track, which in turn

determines the number of sectors on a track. The data transfer time can be calculated with the following equation:

$$T_{transfer} = \frac{N_{request}}{N_{track}} \times \frac{60}{RPM} \quad (5)$$

where N_{track} denotes the number of sectors on a track, and $N_{request}$ is the data length of a request measured in sectors.

Due to the geometric features, outer tracks on disk platters are much larger than the inner tracks. Modern disk drives employ a technique called ZBR, sometimes called Zoned Constant Angular Velocity (ZCAV). ZBR takes advantage of the geometric features to maximize disk capacity by varying the number of sectors per track within the distance from the spindle [Meter 1997]. This technique groups tracks into zones based on their distance from the spindle, and assigns each zone a different number of sectors per track. Outer zones are longer and contain more sectors than the shorter inner zones. The ratio of the sectors of the outmost zone to that of the innermost zone ranges from 1.43 to 1.58 according to the disk characteristics illustrated in [Lumb et al. 2000]. In terms of equation (4), for the same amount of data, the ZBR results in a much smaller data transfer time of the outer zones than that of the inner zones.

2.2 Disk controller

A disk controller is the circuit and the corresponding components that are responsible for controlling a disk drive. The disk controller is built around specially designed microprocessors, which often have digital signal processing capability. A disk controller mainly contains a storage interface, a disk sequencer, Error Correction Code (ECC), servo control, a microprocessor, a buffer controller, and disk cache [Ruemmler and Wilkes 1994; Jeppesen et al. 2001]. The storage interface offers a standard protocol (e.g. IDE, SCSI, FC, SATA, etc) for the disk drives to communicate with its client (e.g. a host system). The disk sequencer manages the data transfer between the storage interface and the data buffer. ECC is responsible for appending ECC symbols to the user data and also checking and correcting the data before it is sent through the storage interface. The servo control detects the current position of the disk head. Based on the position information, the VCM is controlled to allow for track following and seeking. The overall disk drive system is controlled by the microprocessor. The main function of the buffer controller is to provide arbitration and raw signal control to the bank of buffer memory. The disk cache is used as a temporary storage for read/write data from/to the disk drive.

2.2.1 Storage Interface

For the past decades, the most common storage interfaces (IDE, SCSI, FC, SATA, etc) which expose storage capacity as a linear array of fixed-size blocks to file systems have mainly consisted of simple read and write commands. Data access for read and write is specified by a LBN and a data block length. Disk controller is responsible for translating the LBN to physical addresses (Cylinder/Head/Sector, C/H/S). This high-level interface has enabled great portability, interoperability, and flexibility for storage devices and their vendors [Ganger 2001]. However, the narrow storage interface between file systems and storage hides details from both sides. Though both sides have made considerable advancement independently, the interface has limited opportunities for whole system performance improvement due to lacking effective cooperation.

A number of research efforts have recently been invested to augment the communication and cooperation between file systems and storage through the narrow storage interface. Object-based Storage Device (OSD) [Mesnier et al. 2003] offloads storage management from file system to storage device. Creating objects on an OSD is accomplished through a rich storage interface similar to a file system. And, because objects can grow and shrink dynamically, the storage device is responsible for all internal space management of the objects. Semantically-smart Disk System (SDS) [Arpaci-Dusseau et al. 2006] is designed to infer detailed knowledge of how the file system above is using the disk drive. The SDS exploits this knowledge to transparently improve performance or enhance functionality beneath a standard block storage interface. Jiri Schindler et al. [2002] proposed to utilize disk-specific knowledge to match access patterns. By allocating and accessing related data on disk track boundaries, a system can avoid most rotational latency and track crossing overheads to increase disk access efficiency by up to 50% for mid-sized requests (100–500 KB). They implemented two approaches including a general approach applicable to any disk interface supporting a read command and a specialized approach for SCSI disks to detect track boundaries. The Atropos logical volume manager [Schindler et al. 2004] stripes data in track-sized units and explicitly exposes the boundaries, allowing applications to maximize efficiency for sequential access patterns even when they share the array.

Riedel et al. [2001] indicated that the advance in magnetic storage density, mechanics, and electronics eliminated the hardware bottleneck and put pressure on interconnects and hosts to move data more efficiently. Modern disk drives have expanded computational power and disk cache capacity [Riedel et al. 2001; Carrera and Bianchini 2004; Seagate 2009]. For example, the Cheetah X15-36LP disk drive includes an ARM966E-S 32-bit RISC core clocked at 200 MHz and 8MB memory [Seagate 2009]. Therefore, they proposed using an active disk storage device which combines on-drive processing and memory with software download ability. This would allow disk drives to execute application level functions directly at the device. Gurumurthi [Gurumurthi 2007] showed that the bandwidth of disk drives is going to be increasingly difficult to optimize, due to power/thermal constraints. He suggested providing more computational capabilities that data intensive applications could leverage to boost performance. The above methods attempt to alleviate the requirements of IDR by processing the data at storage device level rather than transferring the data to the host memory, thus alleviating the impacts on the storage interface.

2.2.2 Disk cache

Disk cache works on the premise that the data in the cache will be reused often by temporarily holding data, thus reducing the number of physical accesses to the magnetic disk. To achieve this goal, caches exploit the principles of data locality to improve hit ratio. Compared with kinds of I/O optimizations that increase the efficiency of I/Os, reducing the number of physical disk I/Os by increasing the hit ratio of disk cache is the most effective method to improve disk performance. Almost all modern disk drives employ a small amount of on-board cache (RAM) to speed up access to data on a disk drive. Because accessing data from cache is much faster than from magnetic disk, the disk cache can significantly improve performance by avoiding slow mechanical latency, if the data accesses are satisfied from the disk cache (cache hit). The disk cache can also

reduce the heat dissipation because the power required to access disk cache is much smaller than that of magnetic disk.

Data locality can be further divided into spatial locality and temporal locality. The spatial locality implies that if a block is referenced, then nearby blocks will also soon be accessed. The temporal locality implies that a referenced block will tend to be referenced again in the near future. A salient feature of disk cache is that they have almost no temporal locality. This is because the capacity of host memory is normally orders of magnitude larger than the disk cache. Therefore, any data brought into host memory will be re-accessed there, not in the disk cache. Based on this, the hit ratio of disk cache should be very small [Carrera and Bianchini 2004]. Disk cache normally implements prefetch to take advantage of the spatial locality by anticipating future requests for data and bringing it into the cache. This helps increase the hit ratio of disk cache. However, a large prefetch can have a negative impact on small caches, because it can displace the data that would have been useful in the cache.

Disk cache is normally divided into independent segments that correspond to sequential streams of data. Effectively, each I/O stream is treated as having its own cache. When the controller detects that there are more streams than segments, segment replacement takes place to make room for the new streams [Carrera and Bianchini 2004]. There are several typical cache replacement algorithms including Random Replacement (RR), Least Frequently Used (LFU), and Least Recently Used (LRU) [Karedla et al.1994].

(1) The RR replaces cache lines by randomly selecting a cache line to evict. This policy is very fast, requires no extra storage, and is the easiest one to implement. However, it performs poorly because it does not take advantage of the spatial and temporal locality.

(2) The LFU is based on the access counts of the cache lines. The cache lines which have been used least frequently are evicted. Unfortunately, the recently active but currently cold cache lines tend to remain entrenched in the cache. Therefore, the inactive data increases the miss ratio and reduces the cache performance.

(3) The LRU evicts the cache lines used least in the recent past on the assumption that it will not be used in the near future. The LRU is simple to implement for small caches but becomes computationally expensive for large ones. Therefore, it is the most frequently used algorithm in disk cache.

Disk cache today can hold more data due to the increasing cache size (e.g. Ultrastar 15K has 16MB disk cache [Ultrastar 15K147]). This results in higher hit ratio. However, studies have indicated that increasing the cache beyond its optimal size has diminishing performance benefits. Cost is another factor in determining cache size, because the cache memory is still more expensive than the magnetic storage. To achieve an optimal cost-to-performance ratio, system designers generally believe that the size of a cache should be at least 0.1 to 0.3 percent of the backing store. Manufacturers typically offer caches between 0.1 and 1 percent of the backing store [Karedla et al. 1994]. Hsu and Smith [2004] reported that disk cache in megabyte range is sufficient, and for a very large disk cache, the hit ratio continues to slightly improve as the cache size is increased beyond a threshold. Therefore, the further increased cache size only achieves a limited contribution to the hit ratio.

2.2.3 Disk scheduler

Modern disk drives maintain a queue length by setting a queuing threshold. When the queue threshold is reached, the controller of the disk drives can queue the incoming requests until the queued requests are processed. Therefore, when a request is submitted to a disk drive through a file system, the response time consists of two parts including disk access time and queue time. Access time, which measures the time from start of an access to completion, denotes the time required to serve an I/O request. Queue time is the time spent in waiting for its turn to be served. As discussed in the previous sections, disk access time is mainly composed of seek time, rotational latency, and data transfer time. Therefore, the access time depends on the characteristics of disk drives, the current location of disk head, the request address, and the data length. Queue time depends on access time and the data access pattern such as the inter-arrival time of requests. Due to the decrease of access time, the corresponding queue time can be reduced.

Disk schedulers are designed to minimize the access time to the one single dimension of the rotating magnetic platters' logical address space. They can dynamically reorder or rearrange the pending requests in the queue to reduce the seek time and the rotational latency, thus reducing the access time. They do this by taking into account the various delays associated with the rotating media accesses, while providing reasonable response times for individual requests [Worthington et al. 1994]. Over the last four decades, a lot of scheduling algorithms have been proposed and implemented [Worthington et al. 1994; Riska et al. 2004; Hofri 1980; Geist and Daniel 1987].

(1) First Come First Served (FCFS) disk scheduling policy performs I/O requests in order and every request is served without any starvation. This scheduler is easy to implement and it is fair because the expected waiting time of a request is independent of its physical address C/H/S. However, the FCFS often results in suboptimal performance and high mean queue time.

(2) Shortest Seek Time First (SSTF) scheduler processes the pending request in the working queue which is the closest one to the current disk head position, regardless of the moving direction of the disk head. SSTF decreases the high queue time of FCFS over a wide range of workloads by reducing the total seek time. The problem is that the disk head could linger over a subset of the cylinders in an attempt to perform all requests close to that area, thus starving any requests outside of that space.

(3) SCAN algorithm serves requests in the path when the disk head shuttles from the outermost cylinder to the innermost cylinder and then back from the innermost to the outermost. Every request is performed during the scan of two directions. Requests to the middle cylinders achieve better performance because the disk head passes over the centre region at more regular intervals than the edges. Without sacrificing too much performance in the mean queue time, the SCAN algorithm reduces the variance of queue time, thus decreasing the probability of starvation in comparison with SSTF. The SCAN algorithm is sometimes called as Elevator algorithm because it serves requests in a way similar to the way that an elevator serves passengers.

(4) Many variations of the SCAN algorithm have been developed. Cyclical SCAN algorithm (C-SCAN) replaces the bidirectional scan with a single

direction of disk head travel. The C-SCAN treats each cylinder equally, rather than favouring the centre cylinders.

(5) LOOK algorithm is similar to the SCAN but it takes a reverse direction when it hits the last pending request in the current direction. C-LOOK moves the disk head inwards and serves requests in the path until there are no more pending requests in that direction, then it jumps to the outermost outstanding requests.

The reader is referred to [Geist and Daniel 1987] for a comprehensive understanding of the scheduling algorithms. According to the above analysis, a basic premise of disk schedulers is that the queue is long enough and the schedulers can take advantage of the queue to reorder the requests.

2.3 Reliability

Due to the explosive growth of digital information, a large-scale IT infrastructure can involve millions of components. For example, one of the significant advances in cluster networks over the past several years has been that it is now practical to connect tens of thousands of nodes with networks that have massively scalable capacity. However, with the growth of the system scale, hardware component failures are becoming a big challenge to deal with [Deng 2008]. Jiang et al. [2008] analyzed the storage logs collected from about 39,000 storage systems commercially deployed at various customer sites and reported that disk drive failures contribute to 20-55% of storage subsystem failures. The data set covers a period of 44 months and includes about 1,800,000 disks hosted in about 155,000 storage shelf enclosures. Schroeder and Gibson [2007] collected and analyzed seven data sets which vary in duration from one month to five years and cover in total a population of more than 100,000 disk drives from at least four different vendors. Their investigation shows that annual disk drive replacement rates typically exceed 1%, with 2-4% common and up to 13% observed on some systems.

Disk drives are highly complex and dynamic systems that consist of electronic and mechanical components. A disk drive consists of one or more platters rotating on a common spindle. A brushless DC spindle motor is adopted to spin the platters and maintain the RPM. A VCM is employed to drive the disk arm and move the Head Gimbal Assembly (HGA) (generally one disk head per surface) from track to track during seek operations and then hold the HGA on track during read and write operations. The involved electronics and mechanics are combined together to access the data stored on the rotating magnetic platters. They may fail due to various component failures (e.g. disk head, media, firmware, etc). The environmental factors including temperature, humidity, altitude, vibration, contact-start-stop frequency, and duty cycle, all have significant impacts on the failure of disk drives [Yang and Sun 1999].

Disk heads are all integrated into a ceramic slider, which includes an air-bearing surface (ABS) facing the magnetic media. Air entrained between the ABS and the magnetic media generates lift by taking advantage of the viscous properties of air being squeezed through the gap. The air flow is guided by the ABS to control the separation distance between the disk head and the magnetic media within a close tolerance. Since the separation distance directly impacts both signal strength and resolution, it is critical to the recording density of the magnetic media. As the magnetic recording densities increase, the separation distance must decrease correspondingly [Strom et al. 2007]. This separation distance has a significant impact on the reliability of disk drives. It has been proven that it is extremely challenging to simultaneously reduce the separation distance

while enhancing the reliability. Strom et al. [2007] discussed the impacts of disk factors, head factors, and environmental factors on the separation distance in details. They proposed to employ equation (6) to explore the impacts.

$$z = z0 + a \times \Delta T + b \times \Delta P + c \times \Delta P_w + f(S) \quad (6)$$

where T, P, and P_w are temperature, total air pressure, and water partial pressure, respectively. $z0$ denotes the initial separation distance. $f(S)$ is a function of the operation conditions. The coefficients a, b, and c, and $f(S)$ are all determined from controlled experiments. Any deviation in environment or operating condition from its initial state results in a new separation distance z.

Temperature is often the most important environmental factor which affects the reliability of disk drives, because the reliability of both the electronics and the mechanics degrades as temperature grows. Herbs [1997] discovered that high temperature can result in thermal tilt of the disk stack and actuator arms very quickly, thus causing off-track errors and corrupting data on adjacent cylinders. Due to the high temperature, outgassing of the lubricants in the spindle motor and VCM can be induced as well. This often leads to stiction failures and head crash. Herbs pointed out that a disk drive running for an extended period of time at five degrees above the recommended temperature can experience an increase in failure rate of 10% to 15%. Another investigation also shows that a fifteen degree temperature rise is expected to increase the failure rate of disk drives by a factor of two[Anderson et al. 2003]. A recent study reported that a typical server should maintain the air temperature at its front inlets in the range of 20 ^{O}C to 30 ^{O}C, every 10 ^{O}C increase over 21 ^{O}C decreases the reliability of electronics by 50% [Sullivan 2000].

Traditionally, the failure rates of disk drives follow a bathtub curve [Schroeder and Gibson 2007; Yang and Sun 1999]. It indicates that after the high failure rates in the first year (infant mortality), the failure rates are approximately in a steady state around 5-7 years, and then, wear-out phase starts. By investigating 100,000 disk drives, Schroeder and Gibson [2007] reported that disk drives wear out steadily without an infant mortality. They also reported that the disk-independent factors, such as operating conditions, affect the reliability of disk drives more than the component specific factors.

2.4 Energy consumption overview

2.4.1 Power state transition

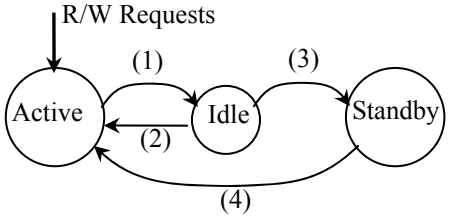

Fig. 2. Power state transition of disk drives

Disk drives have two components that contribute to their overall power demands. The first one is a 12V spindle motor used to spin the platters and drive the head motors. The second one is a 5V supply used to power the analog-to-digital converters, servo-control DSP's, and interface logic [Colarelli and Grunwald 2002]. Due to the mechanical nature, the hardware support for disk energy conservation has not been changed too much over the years. Most modern disk drives have at least three power states: active, idle, and standby. Fig.2 depicts the power state transition of disk drives labelled with a sequence number as defined in the following descriptions. Disk drives perform work while in an active state where the disk spins at full speed. (1) When a data access is completed and there is no succeeding request, the disk drive is transferred to the idle state where the disk platters are still spinning but the electronics may be partially unpowered, and the heads may be parked or unloaded. (2) If the disk drive receives a request when it is in an idle state, the disk drive will be transferred to the active state. (3) To conserve energy, the disk drive can be spun down to the standby state where the disk stops spinning and the head is moved off the disk. (4) To perform requests after entering the standby state, the disk drive must be transferred back from the standby state to the active state by spinning up [Papathanasiou and Scott 2004].

2.4.2 Energy conservation methods

Table I. The major characteristics of five different disk drives

Type		IBM 36Z15	IBM 73LZX	Western Digital WD2500JD	IBM 40GNX	Hitachi DK23DA
RPM		15,000	10,000	7,200	5,400	4,200
Average seek time		3.4ms	4.9ms	8.9ms	12ms	13ms
Average rotational latency		2ms	3ms	4.2ms	5.5ms	7.1ms
IDR(MB/sec)		55	53	93.5(max)	25	18.7~34.7
Power (Watt)	Active	13.5	9.5	13.25(Seek) 10.6(R/W)	3.0	2.0
	Idle	10.2	6.0	10.0	0.82	0.61
	Standby	2.5	1.4	1.8	0.25	0.15
Energy (Joule)	Spin Down	13.0	10.0	6.4	0.4	2.94
	Spin Up	135.0	97.9	148.5	8.7	5.0
Time (Sec)	Spin Down	1.5	1.7	4.0	0.5	2.3
	Spin Up	10.9	10.1	9.0	3.5	1.6

Table I summarizes the parameters of five disk drives from three different manufacturers, where IBM 36Z15 is a high performance server disk drive, IBM 73LZX is a low performance server disk drive, Western Digital WD2500JD is a desktop disk drive, IBM 40GNX and Hitachi DK23DA are mobile disk drives [Papathanasiou and Scott 2004; Carrera et al. 2003; Crk and Gniady 2008]. It shows that disk drives in the standby state or the sleep state use considerably less energy than disk drives in the active state. Many research efforts have gone into investigating the energy consumption of disk drives by taking advantage of this feature [Douglis et al. 1994; Helmbold et al. 2000; Li et al. 1994; Lu and Micheli 1999]. Generally, the existing approaches employed to save energy of disk drives can be classified into four categories [Douglis et al. 1994; Lu and Micheli 1999; Gniady 2006].

(1) The first one is a simple timeout strategy which has gained wide popularity and is currently implemented in many operating systems. Once a disk drive is idle for a specific period of time, which is longer than some

given timeout threshold, the disk is spun down in an effort to save energy. Upon the arrival of a new request, the disk is spun up to serve the request. The timeout strategy offers good accuracy, but it wastes energy when the disk is waiting for the timeout period to expire.

(2) The second one is a dynamic prediction which is based on the behaviours of applications. For example, a series of events that are likely to happen again in the future. The method shuts down the disk drive immediately to eliminate the waiting time of the timeout strategy. However, so far it is less accurate than the simple timeout mechanism.

(3) The third one is a stochastic mechanism. The problem is that the approach usually requires offline pre-processing and the prediction could be inaccurate due to the fluctuant data access pattern.

(4) The last one is an application-aware power management. This mechanism can have very accurate information of the data access pattern. However, it requires modifying the existing applications, which makes it impractical.

Recently, a few works explored how to further enhance the application-aware method for some specific applications without modifying the applications. Focusing on array-intensive scientific applications, Son et al. [2005] proposed a compiler-driven method for disk power management. The compiler analyzes the application code and extracts the disk access pattern. It then employs this information to insert explicit calls in the appropriate places in the code to trigger the power state transitions. It also can spin up/down the disk drive before it is actually required to eliminate the performance penalty incurred by the power state transition. They also developed a compiler directed code transformation approach based on the layout of the data on the disk subsystem to increase the disk inter-access times. Heath et al. [2004] proposed simple application transformations that increase device idle times and inform the operating system about the length of each upcoming period of idleness. These transformations can be either performed by a sophisticated compiler or be implemented by the programmer after a sample profiling run of the application. Deng and Pung [2010] designed a bucket method in virtual machine based environments by leveraging the bursty behaviour of data access pattern. The method divides the workloads issued from each virtual machine into buckets which are equal in time length, and predicts the number of the forthcoming requests in each bucket instead of the length of the idle periods. By doing so, the bucket method makes the converted workload more predictable. The method also squeezes the executing time of each request to the end of its respective bucket, thus extending the idle length. By deliberately reshaping the workloads such that the crests and troughs of each workload become aligned, the method can aggregate the peaks and the idle periods of the workloads. Due to the extended idle length caused by this aggregation, energy can be conserved. The above efforts have made important strides in taking advantage of the application information to effectively tackle the energy consumption of disk drives.

2.4.3 Impacts of power state transition

When switching the disk drives between different power states to save energy, following factors have to be considered:

(1) The power state transition can incur a significant energy cost and time penalty as the disk platters have to be spun up to full speed and the heads

have to be moved back before a request can be served, which requires servo calibration to accurately track the head as it moves over the drive (see table I). To justify this penalty, the energy saved by putting the disk in standby or sleep state has to be greater than the energy needed to spin it up again, and the disk has to stay in the low power state for a sufficiently long period of time to compensate for the energy overhead [Zhu et al. 2005].

(2) Although the increased time penalty may be tolerable, the effect on the reliability of disk drives is not. Frequently spinning down the disk drives has a significant impact on the reliability. Even though the reliability has been significantly improved by using load/unload technology to prevent head-to-disk interaction and start-up wear, the number of start/stop cycles a disk can tolerate during its service life time is still limited [Zhu et al. 2005]. Some disk specifications provide an expected lifetime value (e.g. the IBM Ultrastar 36Z15 can handle a minimum of 50,000 start/stop cycles). Greenawalt [1994] reported that if the disk drive operates continually, it can last approximately 17 years. Yet if it is spun down/up once an hour, its life time decreases to about 4.56 years. Therefore, each power state transition provides the same wear as 3.75 hours of continual operation. Disk drive manufacturers provide a duty cycle rating which is the number of times the disk platters can be spun down before the chances of failure increase to more than 50% on drive spin up. When controlling a disk's power state with a spin down/up algorithm, it results in an accelerated consumption of duty cycles [Bisson et al. 2007].

(3) The methods cannot be applied directly to server disk drives, since the spinning down and spinning up time of the server disk drives are much longer than that of the desktop and laptop. The reason is that server disk drives are physically different from the mobile disk drives. In order to reduce flexing under the stress of faster RPM and increased heat, the server disk drives use lower-capacity but heavy platters for continuous operation and higher vibration-tolerance while serving I/O requests. They also often use different bearing, airflow, and filter designs.

(4) Due to the intensive workload, it is also very difficult to find an idle interval which is long enough to spin down the server disk drives.

3. DESIGN CONSTRAINTS

Although the performance of disk drives has been experiencing 40% growth per year, a number of constraints pose challenges to continue the 40% growth rate [Gurumurthi et al. 2005; Gurumurthi 2007].

(1) The growth of AD results in decreased seek time and increased IDR. This leads to a significant growth of disk drive performance, while providing high storage capacity. A problem is that further density improvement requires a high error/noise tolerance and a very complex head design due to the involved superparamagnetic limit [Gurumurthi et al. 2005]. The perpendicular recording technology will also reach its limits soon, and new

technologies will be required [Perpendicular recording 2009].

(2) Disk cache can dramatically improve the performance of disk drives by avoiding slow mechanical latency, this is because accessing a byte of data in cache can be thousands of times faster than accessing a byte on the rotating magnetic disk media. However, studies have indicated that if the disk cache size grows beyond a threshold, the increased cache only achieves a limited contribution to the hit ratio [Karedla et al. 1994].

Gurumurthi et al. [2005] presented that one of the most fundamental factors affecting disk drive design is the heat generated by certain actions within the disk drive and its effect on reliable operation. This is because high temperature can result in off-track errors and head crashes. Since temperature is one of the most fundamental factors affecting the reliability of a disk drive, one of the requirements in disk drive design is to always keep the operating temperature below a particular threshold (the maximum operating temperature known as the thermal envelope) [Gurumurthi et al. 2005].Therefore, designing disk drives involves tradeoffs between capacity, performance, and power. The goal is achieving the maximum storage capacity and performance within a particular thermal envelope. Gurumurthi et al. [2005] proposed to use the following equation to calculate the power consumed by a disk drive:

$$Power = N_{platter} \times D_{platter}^{4.6} \times RPM^{2.8} \quad (7)$$

where $N_{platter}$ denotes the number of disk platters employed in disk drives, $D_{platter}$ indicates the diameter of the platters. Because the thermal envelope of disk drives has negligible variance over time， the $N_{platter}$, $D_{platter}$ and RPM have counteracting effects on the heat dissipation within a disk drive in terms of equation (7).

The evolution of magnetic recording technology has given the disk manufacturers opportunities to optimize cost (by reducing the number of platters $N_{platter}$ and employing more compact design using smaller diameter platters) and improve performance. For example, the disk diameter $D_{platter}$ has been decreased from 14 inches to 1.8 inches in the past decades. Smaller diameter platters and fewer disk platters are also advantageous in terms of reducing mechanical vibration, speeding up seek operations, and reducing power consumption, heat generation, and noise.

According to the performance overview of disk drives in Section 2, the disk access time mainly depends on platter diameter $D_{platter}$, settling time of disk head t_{settle} , RPM, and AD. The performance of disk drives can be improved by decreasing the t_{settle} and $D_{platter}$, or increasing the RPM and AD. The settling time has basically remained constant. It is also very difficult to further reduce the $D_{platter}$ smaller than 1.8 inches due to the involved mechanical components and heat dissipation in disk drives. Gurumurthi and Sivasubramaniam [2006] discussed this issue by defining a thermal slack. The thermal envelope is based on the temperature obtained with both the VCM and the spindle motor on. However, during idle periods, the VCM is off, this generates less heat. This implies that there is a thermal slack between the thermal envelope and the temperatures when the VCM is off. They reported that the amount of available thermal

slack decreases as the platter size is reduced, since the VCM power is lower for smaller platter sizes [Sri-Jayantha 1995]. This makes smaller slack to exploit in future designs with smaller platters. Therefore, the remaining choices are increasing the AD or RPM in order to improve the performance.

Increasing the RPM can improve disk drive performance significantly. Until now, drive manufacturers have continued to meet the 40% annual growth target of the IDR by increasing RPM and shrinking platter sizes. However, maintaining the current improvement rate poses some big challenges to the disk drive designers. Gurumurthi and Sivasubramaniam [2006] reported that within the thermal envelope, the response time of real workloads can be improved by 30–60% with a 10K increase of the RPM. Unfortunately, the disk drives rotating at speeds exceeding 20, 000 RPM have been researched but not commercialized due to heat generation, power consumption, noise, vibration and other problems in characteristics, and a lack of long term reliability [Thompson and Best 2000]. Therefore, it is a big challenge to design new disk drive architecture which could further advance disk performance.

4. NEW DISK DRIVE ARCHITECTURES

The architectural design of disk drives has reached a turning point which should enable the storage capacity and performance to advance further, while reducing power consumption and maintaining high reliability. As discussed in Section 2.4.3, switching disk drives between different power states is not applicable to the server disk drives. Dynamic Rotations Per Minute (DRPM) [Gurumurthi et al. 2003; Carrera et al. 2003] is proposed for power management in server disk arrays. The DRPM technique dynamically modulates the rotational speed of disk drives so that the disk can serve requests at different RPMs. This can provide large savings in power consumption with very little perturbation in delivered performance. Carrera et al. [2003] compared several techniques used for energy conservation. They discovered that the multi-speed disk approach is the only one that can really conserve energy on network servers. EED [Deng et al. 2008b] is an energy efficient disk drive architecture which integrates a relatively small-sized NAND flash memory into a traditional disk drive and moves the hot data from the disk drive to the flash memory, thus saving energy by extending the length of the idle intervals. A disk drive including two or more spindles each carrying one or more platters was introduced in [Hard disk drive with multiple spindles]. By using reduced diameter platters, multiple spindles' platters can be placed within the chassis of a disk drive to increase the capacity and/or performance of the drive. For example, two independent sets of heads accessing two independent sets of platters effectively doubles the rate at which data can be written to or read from the platters. Reducing the diameter of the platter correspondingly reduces the seek time and the heat dissipation as well. The disk drive architecture which has multiple disk actuators was proposed in early literature [Smith 1978]. Recently, some research efforts have begun to focus on the multiple disk actuators architecture again due to performance requirements. Zheng et al. [2005] used a circle-fit model testing method to identify the dual input and dual output frequency response model of the dual actuator plan. They discussed the decentralized control scheme of the dual actuator tracking servo as well. Chandy [2007] proposed a dual actuator logging disk architecture which adds a second actuator and set of disk heads to a disk drive. The second actuator is dedicated to reads, thus allowing the write head to remain in regions where there are more available free sectors. The architecture guarantees near-zero-access

writes regardless of the read behaviours while providing access times for reads equivalent to a traditional disk drive.

5. NEW STORAGE MEDIA AND STORAGE DEVICES

5.1 Flash memory

Flash memory is a non-volatile memory which can be electrically erased and reprogrammed. Its major advantages such as small physical size, no mechanical components, lower power consumption, non-volatility, and high performance have made it likely to replace disk drive in more and more systems (e.g. digital camera, MP3 player, mobile phone etc.) where either size and power or performance are important [Chang and Kuo 2005]. Flash memory technology has advanced considerably since its emergence more than 20 years ago. Samaung has delivered NAND flash memory with capacity ranging from 64MB to 4GB [Samsung 2009a; Samsung 2009b; Samsung 2009c]. A 32GB flash disk which integrates 16 2GB flash memory chips is also available on the market [Samsung 2009c]. Due to the increased capacity and decreased price, flash memory is expected to be widely used in consumer electronics, embedded systems, and mobile devices.

There are two major types of flash memory, which are available on the market, following different logic schemes: namely NOR, and NAND. The NOR flash memory, which employs a standard memory interface, is byte accessible and can be adopted as execute-in-place memory. It is mainly used for EEPROM replacement. Compared with the NOR flash memory, NAND flash memory has faster erasing and write times, along with higher data density. These features make NAND flash a better candidate for data storage.

Table II. Characteristics of typical NAND flash memories

Manufacturer	Samaung	Samaung	Intel	AMD	FUJITSU
Type	K9F6408U0A	K9NBG08U5A	JS29F16G08FANB1	Am30LV0064D	MBM30LV0128
Capacity	8M×8Bit	4G x 8 Bit	2G x 8 Bit	8M×8Bit	16M×8Bit
Page Size(Byte)	(512 + 16)	(2K + 64)	(2K + 64)	(512 + 16)	(512 + 16)
Block Size(Byte)	(8K + 256)	(128K + 4K)	(128K + 4K)	(8K + 256)	(16K + 512)
Random Read	10µs(Max)	25µs(Max)	25µs(Max)	N/A	10µs(Max)
Serial Read	50ns(Min)	50ns(Min)	25ns(Min)	<50ns	35ns(Min)
Program time	200µs (Typ.)	200µs(Typ.)	220µs(Typ.)	200µs	200µs (Typ.)
Erase Time	2ms(Typ.)	1.5ms(Typ.)	1.5ms(Typ.)	2 ms	2 ms
Endurance	1Million	100K	100K	10K	1Million
Voltage	2.7-3.6 V	2.7-3.6V	2.7-3.6V	2.7-3.6V	2.7-3.6V
Power(Active)	30 mW	75 mW	75 mW	30 mW	72 mW
Power (standby)	30 mW	60 mW	3 mW	0.03 mW	3.6 mW

Table II summarizes the parameters of five different NAND flash memories from four manufacturers [Samsung 2009a; Samsung 2009b; Intel 2009; AMD 2009; FUJITSU 2009]. The NAND flash memory is accessed much like block devices (e.g. disk drives) which require data to be read or written in larger units. NAND flash memory is composed of a fixed number of blocks, where each block consists of a number of pages, and each page has a fixed-size main data area and a spare data area. Data on a NAND flash

memory is read or written in a unit of one page, and the erasing is performed in a unit of one block. A page can be either writable or un-writable, and any page initially is writable. The writable pages are called free pages. A writable page becomes un-writable once it is written. A very important feature of NAND flash is that the pages cannot be rewritten. When a portion of data on a page is modified, the new version of data must be written to an available page somewhere. The page which stores the old version of the data is considered as dead, while the page which stores the newest version of data is considered live.

When the storage capacity becomes low, garbage collection has to be triggered to recycle the invalidated pages. Because erasing is performed in blocks, the valid pages in the recycled blocks have to be copied to somewhere before erasing the blocks. Therefore, the performance is normally very low when the system is performing the garbage collection. An optimal garbage collection algorithm can reduce the performance impact to a certain degree [Chang et al. 2004]. Another important feature of the NAND flash memory is the endurance cycles. A block will wear-out after a specified number of program/erase cycles ranging from 10,000 to 1,000,000. A poor garbage collection policy could quickly wear out a block and a flash memory chip. A wear levelling process attempts to evenly distribute the data between memory cells to guarantee that no one cell is overly burdened. Because when some blocks of flash memory were worn out, the whole flash memory chip would start to malfunction, a good wear-leveling scheme should be able to keep an even distribution of erase cycle counts across all the blocks.

The garbage collection and wear-leveling have two different objectives which could conflict with each other. The garbage collection scheme prefers to recycle the blocks which have a small number of valid pages. On the contrary, the wear-leveling policy normally recycles blocks which are not erased for a certain amount of time, in order to eliminate excessive writes to the same physical flash memory location. These blocks usually store much valid and read-only data. Because it is not necessary to perform garbage collection if there are already sufficient free pages in system. The garbage collection could be activated only when the number of free pages is less than a threshold value. A typical wear-leveling aware garbage collection policy might sometimes recycle the blocks that have the least number of erasing, regardless of how many free pages can be reclaimed. Therefore, it is always a challenge to balance between the garbage collection and wear-leveling. The reader is referred to [Chang and Kuo 2005] for a comprehensive understanding of flash memory.

NAND flash memory can play two roles in the existing computer system architecture: (1) As an extension to RAM, and a layer between RAM and the traditional disk drives. (2) Replacing the traditional disk drives as a new block storage media. We will discuss how the NAND flash memory plays the two roles in hybrid disk and SSD, respectively.

5.2 Promising storage media

Flash memory is a potential storage media. However, it is running into severe scalability challenges beyond the 40nm technology node since it relies on charge storage. For the charge storage, Gate Coupling Ratio (GCR) represents the fraction of voltage drop across the tunnel oxide and must be higher than 0.6 for the device to function during write and erase operations. High GCR is normally achieved by wrapping the control gate around the sidewalls of the floating gate. At below 40 nm, the spacing between two floating gates may become too narrow for the interpoly dielectric and control gate to wrap. Therefore, it is a challenge to maintain sufficiently high GCR [ITRS07].

Recently, the resurgence of a number of non-volatile storage technologies brings opportunities to the architectural design of disk drives and storage devices. Magnetic Random Access Memory (MRAM) combines a magnetic device with standard silicon based microelectronics to obtain the combined attributes of non-volatility, high performance, fast programming, and unlimited program endurance. This technology provides random access with no refresh. MRAM is expected to achieve the density of flash memory but at significantly faster write speeds and with unlimited endurance [Subramanian et al. 2004]. MicroElectroMechanical System (MEMS) is a very small-scale mechanical device which slides, bends, and deflects in response to electrostatic, electromagnetic, and external environmental forces. MEMS-based storage is a non-volatile storage technology that merges magnetic recording material with thousands of probe based recording heads to provide online storage [Schlosser et al. 2000]. However, both the MRAM and MEMS are still in their infant phase of development.

Memristors are a form of nano-scale resistor. They were first proposed in 1971 but have only recently been built by researchers at HP lab. The memristors can remember the amount of charge that has flowed through even when the power is off [What Are Memristors 2009]. This characteristic can be leveraged to build some kinds of non-volatile memory. It is normally believed that the growth trend of computing power begins to slow down as the components used in electronic circuits are shrunk to the size of just a few atoms. Memristors are supposed to extend Moore's Law beyond current physical limitations. They would provide greater performance at lower energy. Phase-change Random Access Memory (PRAM) is a much more mature technology than MRAM or MEMS. Samsung introduced a 512Mb working prototype of PRAM which is expected to be the main memory device and to replace the high density NOR flash within the next decade [Samsung 2009d]. Because the PRAM can rewrite data without having to first erase data previously accumulated, it is effectively 30 times faster than conventional flash memory. It is also expected to have at least 10 times the endurance of the conventional flash memory. However, as a product, both the memristors and PRAM still have a long way to go.

5.3 Hybrid disk

5.3.1 Performance gap within disk drives

The disk cache and magnetic media in a traditional disk drive construct a two layer architecture in terms of the memory hierarchy. Disk drives normally use conventional main memory (Synchronous Dynamic Random Access Memory, SDRAM) as disk cache. Today's SDRAM has access time ranging from 7 to 10 nanoseconds. We assume that 512 Byte data (one sector size of disk drive) need to be accessed in the SDRAM which has 64 bit chip configuration and 10 nanoseconds access time. The disk cache access time is about $T_{cache} = \left(\frac{512 \times 8}{64} \right) \times 10 \times 10^{-6} = 6.4 \times 10^{-4}$ milliseconds.

The latest Hitachi Ultrastar 15K147 [2009] has characteristics of 3.7 milliseconds average seek time T_{seek}, 15000 RPM and maximal 1129 Mbits/sec internal media transfer rate. Based on equation (4) and equation (5), it is very easy to calculate that the

average rotational latency is $T_{rotate}=\left(\frac{60\times10^3}{15000}\right)\times\frac{1}{2}$=2 milliseconds, and the internal transfer time of 512Byte is $T_{transfer}=\left(\frac{512\times8}{1129}\right)$=$3.63\times10^{-3}$ milliseconds, respectively. We have the average disk access time $T_{access}=T_{seek}+T_{rotate}+T_{transfer}$=3.7+2+$3.63\times10^{-3}$=5.70363 milliseconds.

According to the above analysis, it can be concluded that the average access time of disk cache is about $\frac{T_{access}}{T_{cache}}$=$5.70363/6.4\times10^{-4}\approx8.9\times10^3$ times faster than that of a modern disk drive. The large performance gap between disk cache and disk media leaves opportunities to explore new I/O optimizations or storage architectures to improve the disk I/O performance.

The access time of the traditional two-layer disk drive can be expressed with following formula:

$$T_{two-layer}=H_{cache}\times T_{cache}+(1-H_{cache})T_{access} \quad (8)$$

where T_{cache} is 6.4×10^{-4} milliseconds and T_{access} is 5.70363 milliseconds in terms of the above discussion. Because the characteristics of disk drives in this paper are based on the datasheet of Hitachi Ultrastar 15K147[2009], we have the access time $T_{two-layer}$=2 milliseconds from the datasheet. According to equation (8), we have:

$$H_{cache}=1-\frac{T_{two-layer}-T_{cache}}{T_{access}-T_{cache}} \quad (9)$$

It is easy to calculate the $H_{cache}=65\%$ in terms of equation (9). According to the discussions in Section 2.2.2, it seems that the hit ratio is too high. This is because the calculated disk access time T_{access} is based on the average seek time and average rotational latency which are higher than the real values. Equation (9) indicates that the H_{cache} increases with the growth of T_{access}.

5.3.2 Anatomy of hybrid disk

According to the above discussions, the traditional disk drives are millisecond devices, and DRAM are nanosecond devices. Table II demonstrates that the NAND chips are microsecond devices. It seems that NAND flash memory can play as an intermediate layer (e.g. a non-volatile cache) between the DRAM and the traditional disk drives in terms of the memory hierarchy.

A hybrid disk integrates NAND flash memory into a standard disk drive as a second level cache. Therefore, the hybrid disk consists of three layers: disk cache, NAND flash memory, and magnetic platters. By taking advantage of appropriate software, the hybrid disk can boot faster and save energy [Samsung 2006]. Windows Vista employs a new feature called ReadyDrive to leverage the flash memory embedded in the hybrid disk to reduce boot time. A computer with ReadyDrive and a hybrid disk copies files, which will

be required to start the computer again, to the flash memory of the hybrid disk to speed up the boot process when the computer is shut down. Another feature named SpuerFetch is adopted by Windows Vista to analyze data access pattern, and page the data that is likely to be needed into the system memory. Once the paging is done, the disk drive can be switched to a low power state to save energy since the Windows is supposed to read data from the main memory. The Windows writes data in the flash memory. In two cases, the disk drive has to be spun up. (1) When the flash memory is full and has to flush data to the disk drive. (2) When the Windows requires some data which has not been paged to the main memory. Intel's Turbo Memory (a. k. a. Robson) is a similar technology which puts a specialized flash memory cache on a notebook motherboard. It uses a PCI express mini card to hold the flash memory chips [Trainor 2007].

For the three layer hybrid disk drive, we have the access time as follows:

$$T_{three-layer} = H_{cache} \times T_{cache} + (1 - H_{cache}) \times (H_{flash} \times T_{flash} + (1 - H_{flash}) \times T_{access}) \quad (10)$$

where T_{cache} =6.4$\times 10^{-4}$ milliseconds, $H_{cache} = 65\%$, T_{access} =5.70363 milliseconds. According to the datasheet of Samsung NAND Flash Memory K9F6408U0A-TCB0 [Samsung 2009a], we take $T_{flash} = 10 \times 10^{-3}$ milliseconds and 200×10^{-3} milliseconds for read and write, respectively. The page size of K9F6408U0A-TCB0 is equal to the sector size of a disk drive. The hit ratio of flash memory is taken as $H_{flash} = 90\%$ based on the skew of 90/10 Rule [Staelin and Garcia-Molina 1990]. According to equation (10), we have $T_{three-layer}$ (read) =6.4×10^{-4}×65%+ (1-65%) × (10×10^{-3}+ (1-90%) ×5.70363) =0.2 milliseconds, and $T_{three-layer}$ (write) =6.4×10^{-4}×65%+ (1-65%) × (200×10^{-3}+ (1-90%) ×5.70363) =0.27 milliseconds. The analysis indicates that due to the integrated flash memory, the read and write performance of disk drives is speed up by 10 times and 7.4 times, respectively. This calculation is based on a relatively high disk cache hit ratio. However, we believe that the theoretical analysis can prove the key points of the hybrid disk architecture.

Hybrid disk architecture is unique. It brings some challenges to the disk scheduler, disk cache policy, spin up/down algorithm, etc, which are optimized for the traditional disk drive architecture. The main goal of the traditional disk schedulers is to dynamically order the pending requests in the queue and minimize the total positioning overhead. They do this by taking into account the various delays associated with the rotating media accesses, while providing reasonable response times for individual requests [Worthington et al. 1994]. However, with hybrid drives which integrate two different storage media into one closure, such a presumption may no longer be efficient. For example, two access times of a hybrid disk are available: one along the rotating media address space, and one along the flash media address space [Bisson and Brandt 2007]. Different storage media have completely different characteristics (e.g. the random write performance of flash memory is especially tricky in comparison to the magnetic media). Bisson and Brandt [2007] proposed a scheduler called flash-backed I/O requests for hybrid disk drive to reduce write latency. This method reduces write latency by redirecting write requests to flash media when it is more efficient to service an I/O request rather than the rotating platters. The redirected request is kept in the main memory until it can be written to

rotating media without a significant write penalty. Operating systems can also leverage hybrid disks to reduce power consumption by redirecting I/O to the flash memory, while the rotating media is in a low power state. This method can reduce the performance and power consumption penalty incurred by spinning up the disk platters. Four spin-down algorithm and I/O subsystem enhancements are proposed in [Bisson et al. 2007] to leverage the hybrid disk to achieve energy conservation.

5.4 Solid state disk

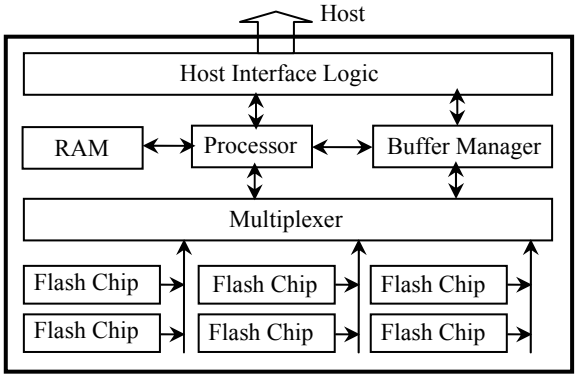

Fig. 3. SSD logic components

The traditional term SSD refers to semiconductor devices. Therefore, an SSD indicates the use of semiconductors to emulate a hard disk drive. SSD commonly consists of either DRAM volatile memory, or NAND flash non-volatile memory. The DRAM based SSD requires an internal battery and backup disk drive to guarantee data persistence. This is why most of the current SSDs employ non-volatile flash memory as the storage media (e.g. USB memory sticks). The advent of the NAND-flash based SSD represents a sea change in the architecture of computer storage subsystems. Agrawal et al. [2008] described a general block diagram for an SSD in a great detail. An SSD basically consists of host interface logic, a logical disk emulation, an internal buffer manager, a multiplexer, a processing engine, flash chips, etc. Fig.3 shows the logic components of a typical SSD. Please refer to [Agrawal et al. 2008] for more details. Table III illustrates the characteristics of four SSDs from four different manufacturers, where ZEUS 2GB FC SSD and RamSan-20 are high-end server-level SSDs, MCCOE64G5MPP and SP064GBSSD750S25 are low-end SSDs [ZEUS 2009; Texas Memory Systems 2009; Flash SSD Charts 2009].

Agrawal et al. [2008] discussed the potential issues which could significantly impact the SSD performance. These issues include data placement, parallelism, write ordering, and workload management. A high performance and NAND flash memory based storage system is proposed in [Kang et al. 2007]. The system consists of multiple independent channels, where each channel has multiple NAND flash memory chips. The system exploits and maximizes I/O parallelism from multiple channels and multiple NAND flash memory chips by leveraging striping, pipelining, and interleaving. Birrell et al. [2007] proposed to integrate sufficient RAM into flash disks to hold data structures describing a fine grain mapping between disk logical blocks and physical flash addresses. The method can significantly improve the performance of the flash disk. Compared with traditional

disk drives, NAND flash memory based SSDs exhibit much better performance for random read, a similar or better performance of sequential read and sequential write. However, SSDs exhibit worse performance for random writes due to the unique physical characteristics of NAND flash memory [Kim and Ahn 2008]. BPLRU [2008] is a new write buffer management scheme which significantly improves the random write performance of flash storage. BPLRU considers the common flash translation layer characteristics and attempts to establish a desirable write pattern with RAM buffering. A smart buffer cache is designed and integrated into a NAND flash memory package to enhance the spatial and temporal locality. This new flash memory package can achieve higher performance and lower power consumption compared with any conventional NAND-type flash memory module [Lee 2005]. Data access pattern also has significant impact on the flash based storage system. A highly efficient method for on-line hot data identification is proposed to reduce the impacts on the garbage collection, performance, and lifespan [Hsieh 2006].

Table III. Characteristics of typical SSDs

Type	ZEUS 2GB	RamSan-20	MCCOE64G5MPP	SP064GBSSD750S25
Manufacturers	STEC	Texas Memory	Samsung	Silicon Power
Capacity	18-146 GByte	450GByte	64GByte	64GByte
Interface	FC	PCI-e	SATA/300	SATA/300
Average access time	20 - 120 µsec	50 µsec	0.12ms	0.20 ms
Sustained read bandwidth	200 MB/sec	700 MB/sec	90.6 MB/sec	116.2 MB/sec
Sustained write bandwidth	100 MB/sec	500 MB/sec	83.7 MB/sec	33.5 MB/sec
Sustained IOPS	50,000 (random)	80,000 (R/W)	1852	247.5
Power (Idle) (Watt)	5.4	N/A	0.3	0.64
Power (Max) (Watt)	8.4/8.1(R/W)	15	0.77	1.47
Startup power (average)	14.1 Watt	N/A	N/A	N/A
Startup time (average)	30 sec	N/A	N/A	N/A

SSDs are expected to be the major storage media in the forthcoming 10 years. Currently, price and storage capacity are two major hurdles for the customers in widely adopting the SSDs. Samsung announced a 32GB SSD consisting of 16 2GB NAND flash chips. The product is supposed to replace the mini laptop hard drives. However, it costs around 960$ to purchase the 32GB SSD [Samsung 2009c]. Above discussions indicate that there are a lot of potential research topics involved in SSDs. For example, many rules of thumb extracted from the traditional storage systems may not be applicable to the SSDs due to their unique characteristics.

6. DISCUSSIONS

The traditional disk drives, hybrid disk, and SSD have been discussed in the previous sections. The challenges (e.g. endurance cycles, erase before write, etc) presented by NAND flash memory indicate that the flash memory of a hybrid disk could fail before the magnetic disk. Furthermore, if a hybrid disk is spun down and the operating system requires data that has not been cached in the flash memory or main memory, the magnetic platters would have to be spun up to serve the requests. As discussed in Section 2.4, spinning up the magnetic platters takes extra time and power. This incurs noticeable delay and power penalty. Spinning down/up the magnetic platters too often also has a significant impact on the reliability. Disk drive manufacturers provide a duty cycle rating which is the number of times the rotating platters can be spun down before the chances of failure increase to more than 50% on platters spin up. Using a spin down/up algorithm to control a disk's power state results in an accelerated consumption of duty cycles [Bisson et al. 2007]. Therefore, the hybrid disk must consider the energy saving against the decrease in the reliability, though it can strike a good balance among storage capacity, performance, and energy efficiency. We believe that the hybrid disk is a temporary method. In this section, we will only discuss the performance and power issues of the traditional disk drives and the SSD by summarizing the parameters in table I and table III.

Fig. 4 illustrates the bandwidth of five disk drives and four SSDs. Since ZEUS and RamSan-20 are high-end SSDs, they are capable of producing exceptional bandwidth in comparison to the traditional disk drives. Even the low-end SSDs provide higher bandwidth than the high-end magnetic disk drives. According to table II, the page size of large-capacity flash memory chip is 2KB. Therefore, we assume that 2KB data is required to be transferred from the magnetic media to the disk cache. Based on table I, the average access time can be calculated by using equation (1). Fig. 5 shows the average access time comparison. As expected, the SSDs are orders of magnitude faster than the traditional disk drives.

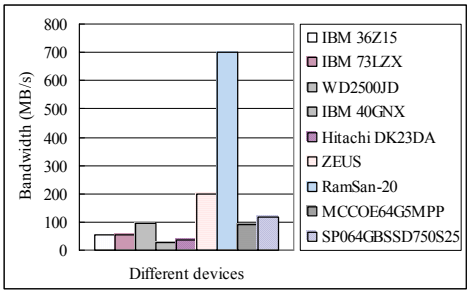

Fig. 4. Bandwidth comparison

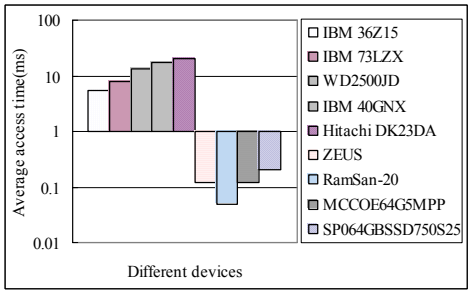

Fig. 5. Average access time comparison

To better understand the behavior of flash memory based SSDs, Polte et al. [2008] performed a thorough investigation about the performance of several high-end consumer and enterprise SSDs, and compared their performance to a few generally available disk drives. They reported that for sequential access pattern, the SSDs are up to 10 times faster for reads and up to 5 times faster for writes than the magnetic disk drives. For random reads, the SSDs provide up to 200 times performance advantage. For random writes, the SSDs offer up to 135 times performance advantage. They concluded that SSDs are approaching price per performance of the traditional disk drives for sequential access patterns workloads, and are superior technology to disk drives for random access patterns.

In contrast to the sophisticated controllers of the enterprise SSDs, they also discovered that the consumer SSDs perform even worse than the disk drives for small random writes. Their tests showed that the consumer SSDs achieve between 100 and 200 IOPS, and the disk drives achieved performance between 300 and 500 IOPS. Therefore, we believe that the intelligent algorithms (e.g. inherent log-structured pattern of writing, maintaining a pool of pre-erased blocks, coalescing writes to minimize rewriting data without changing it, servicing write requests in parallel, etc.), which can be integrated into the controller of SSDs, have significant impacts on the performance of SSDs. The reader is referred to [Polte et al. 2008] for details.

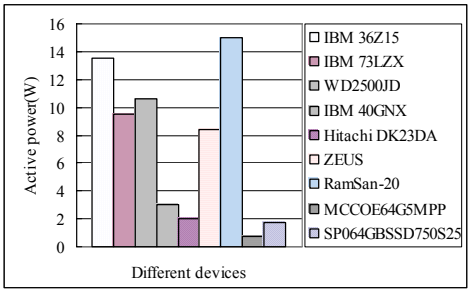

Fig. 6 Active power comparison

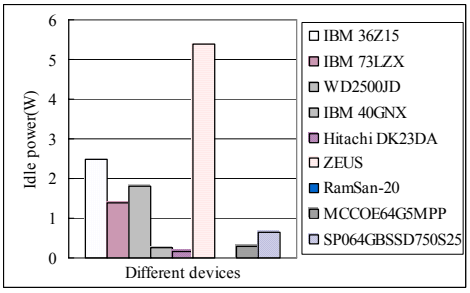

Fig. 7 Idle power comparison

Power consumption depends on the characteristics of different storage devices and data access patterns. This paper will discuss the power consumption when the devices are fully loaded (active state) and when the devices are in idle states. According to table II, it is easy to draw a conclusion that SSDs are more energy efficient than the traditional disk drives, since they do not involve mechanical components. However, this is arguable. Fig. 6 depicts the active power consumption of different disk drives and SSDs. It shows that the power consumption of high-end SSDs is comparable to the high-end disk drives, and the low-end SSDs take less power than the low-end disk drives. This is because the high-end SSDs employ more powerful processors, more complex circuits, bigger RAM than that of the low-end SSDs to offer much higher performance and reliability. Because the idle power of RamSan-20 is not available, Fig. 7 illustrates the idle power consumption of five disk drives and three SSDs. The idle power of the five disk drives is actually the standby power in table I, since the disk drives in the standby state take the lowest power. Fig. 7 indicates that the mobile disk drives take the lowest power across the disk drives and SSDs. It is interesting to observe that the high-end ZEUS consumes the highest power.

As discussed in Section 2.4, in some scenarios, we can achieve energy conservation by switching disk drives between different power states. A premise is that the disk has to stay in the low power state for a sufficiently long period of time, so that the saved energy can outweigh the energy needed to spin the disk up again. This technology can be applied to SSDs. However, it is more challenging if we want to make it practical. It takes 14.1×30=423 Joule to start the SSD in terms of table III, which is much bigger than the energy consumption of spinning up the high-performance server disk drive listed in table I (IBM 36Z15 takes 135.0 Joule). Unfortunately, due to the lack of publicly available parameters of other SSDs, this claim requires further confirm, but we believe that it may

be more applicable to switch some specific flash memory chips rather than the whole SSD.

In contrast to flash memory, a very important advantage of the traditional disk drives is the large capacity. As discussed in Section 1, the AD of magnetic recording has achieved 100% growth annually over the last decade. This trend is likely to continue for a few years. However, there are two reasons which make it very difficult to construct high-end storage systems by using the large-capacity disk drives. First, the reduction of seek time is hindered by the settling time of disk head, and the settling time has remained largely constant. Secondly, the sequential read rate grows linearly with the increase of LD whilst capacity goes up to the square, thus failing to keep up with the increased capacity. Due to the above two reasons, it takes about a few hours to read a one terabyte disk drive (7,200 RPM). Therefore, rebuilding a large RAID5 consisting of the large-capacity disk drives would take extremely long time. It also decreases the reliability of the storage system since disk failures are normally correlated with each other [Schroeder and Gibson 2007]. With the growth of the rebuilding time, it is more likely another disk drive in the storage system would fail. According to equation (7), decreasing the RPM can significantly reduce the power consumption of disk drives. However, lower RPM can further worsen the performance issue of the large-capacity disk drives. For example, replacing 7,200 RPM with 3,600 RPM will double the time required to read the whole disk drives. Based on the above discussions, it seems we are in a dilemma to strike a balance between capacity, performance, and power consumption of traditional disk drives. An innovative disk file system may be able to alleviate the dilemma by leveraging the available I/O bandwidth.

At the time of this writing, due to the explosive growth of magnetic recording technology, one terabyte disk drives (e.g. SATA Western Digital, WD RE2-GP) are available on the market, and 2.5 inch and 10,000RPM disk drives have been adopted by some high-end servers. The 2.5 inch disk drives take less power and space than the 3 inch disk drives, but provide comparable performance. These are all competitive selling points for the traditional disk drives in comparison to the SSDs. However, from a power standpoint, Table I shows that one disk drive is not a problem. Even the addition of several dozen disk drives would hardly be a concern. However, if hundreds or thousands of disk drives are put together (e.g. storage cluster [Deng 2008], storage grid [Deng et al. 2008a]), it will quickly become a big headache. SSDs provide a very good opportunity to build a cool storage system by leveraging their exceptional high performance. For example, using the SSDs as a kind of cache in a storage cluster to hold the frequently accessed data can significantly extend the idle length of the back-end disk drives, thus saving energy. As the cost of NAND flash memory continues to decline and the capacity continues to grow, the potential application space for SSDs will continue to increase.

Disk file system is designed to store and manage files on disk drives. It provides a transparent and easy way to locate and access files by hiding the details of disk drives. The disk file systems are optimized for the performance limitations and characteristics of disk drives (e.g. improving hit ratio of disk cache, reducing random accesses, minimizing seek time, avoiding fragmentation, etc). Therefore, a file system optimized for flash memory is required to handle the known weaknesses such as rewriting [YAFFS 2009]. SSD is much faster than the current most advanced mechanical disk drives. This indicates that the current disk shelves and controllers need to be redesigned from scratch for this very low latency environment. Placing the SSD into the existing disk shelves is a temporary and short-term approach.

7. CONCLUSIONS

Recently, it has been recognized that it is not necessary to instantly have a computer's maximum power available, as long as the Quality of Service (QoS) delivered satisfies a predetermined standard. Power consumption is now a metric equal to performance [SPEC-Power and Performance 2009]. In the past 50 years, the disk drive architecture has remained largely unchanged. The traditional disk drives are continually challenged by their competitors SSDs on the market at present. Fortunately, attributing to their advantages in the capacity and cost, they still have a few years to go. However, it has reached a turning point at which they have to be reborn, in order to further improve their performance and reduce the power consumption, while still maintaining high reliability. As explained before, hybrid disk is a temporary approach. Therefore, an architecture shift is required to achieve this goal. The dual actuator [Chandy 2007] and multiple spindles [Hard disk drive with multiple spindles] methods may be able to alleviate the pressure.

This paper presents comprehensive insights of the evolving storage devices from traditional disk drives, to hybrid disk, and to SSD. It identifies the design constraints that the traditional disk drives are facing. Where appropriate, challenges and opportunities are highlighted and discussed from both the performance and energy standpoints. The goal is to ferment future research in the community.

ACKNOWLEDGEMENTS

I would like to thank the anonymous reviewers whose constructive comments and suggestions have significantly enhanced this paper. Thanks also to Robin. The writing of this paper is inspired by his insightful comments in the community. In addition, I am grateful to Prof. Jose Martinez and the editors of ACM Computing Surveys for giving me the opportunity to clarify my thoughts. This work was funded in part by a startup research fund from Jinan University.

REFERENCES

AGRAWAL, N., PRABHAKARAN, V., WOBBER, T., DAVIS, J. D., MANASSE, M., PANIGRAHY, R. 2008. Design Tradeoffs for SSd Performance. In *Proceedings of the USENIX Technical Conference*.

AKYÜREK, S., AND SALEM, K. 1995. Adaptive Block Rearrangement. *ACM Transactions on Computer Systems*, 12(2), 89-121.

AMD. 2009. AMD NAND Flash Memory. Am30LV0064D datasheet.

ANDERSON, D., DYKES, J., AND RIEDEL, E. 2003. More Than an Interface -SCSI vs. ATA. In *Proceedings of the 2nd USENIX Conference on File and Storage Technologies (FAST03).*

ARPACI-DUSSEAU, A.C., ARPACI-DUSSEAU, R.H., ET AL. 2006. Semantically-Smart Disk Systems: Past, Present, and Future. *ACM SIGMETRICS Performance Evaluation Review*, 33 (4), 29-35.

BIRRELL, A., ISARD, M., THACKER, C., AND WOBBER, T. 2007. A Design for High-Performance Flash Disks. *ACM Operating Systems Review*, 41(2), 88-93.

BISSON, T., AND BRANDT, S. A. 2007. Reducing Hybrid Disk Write Latency with Flash-Backed I/O Requests. In *Proceedings of the 15th International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS)*.

BISSON, T., BRANDT, S. A., LONG, D.D.E. 2007. A Hybrid Disk-Aware Spin-Down Algorithm With I/O Subsystem Support. In *proceedings of IEEE International conference on Performance, Computing, and Communications Conference 2007 (IPCCC 2007)*, 236-245.

CARRERA, E. V., PINHEIRO, E., AND BIANCHINI, R. 2003. Conserving Disk Energy In Network Servers. In *Proceedings of the 17th International Conference on Supercomputing*, 86-97.

CARRERA, E. V., AND BIANCHINI, R. 2004. Improving Disk Throughput In Data-Intensive Servers. In *Proceedings of the 10th International Symposium on High-Performance Computer Architecture*.

CHANG, L., KUO, T. 2005. Efficient Management For Large-Scale Flash-Memory Storage Systems With Resource Conservation. *ACM Transactions on Storage*, 1(4), 381–418.

CHANDY, J.A. 2007. Dual Actuator Logging Disk Architecture And Modeling. *Journal of Systems Architecture*, 53(12), 913-926.

CHANG,L., KUO,T., LO,S. 2004. Real-Time Garbage Collection For Flash-Memory Storage Systems Of Real-Time Embedded Systems. *ACM Transactions on Embedded Computing Systems*, 3(4), 837-863.

COLARELLI, D., AND GRUNWALD, D. 2002. Massive Arrays Of Idle Disks For Storage Archives. In *Proceedings of the 2002 ACM/IEEE Conference on Supercomputing*. 1-11.

CRK, I., GNIADY, C. 2008. Context-Aware Mechanisms for Reducing Interactive Delays Of Energy Management in Disks. In *Proceedings of USENIX 2008 Annual Technical Conference on Annual Technical Conference*, 71-84.

DENG, Y. 2008. RISC: A Resilient Interconnection Network for Scalable Cluster Storage Systems. *Journal of Systems Architecture*, 54(1-2), 70-80.

DENG, Y., WANG, F., HELIAN,N., WU,S., LIAO,C. 2008a. Dynamic and Scalable Storage Management Architecture For Grid Oriented Storage Devices. *Parallel Computing*, 34(1), 17-31.

DENG, Y., WANG, F., HELIAn, N. 2008b. EED: Energy Efficient Disk Drive Architecture. *Information Sciences*,178(22), 4403-4417.

DENG, Y. 2009. Exploiting the Performance Gains of Modern Disk Drives By Enhancing Data Locality. *Information Sciences*, 179 (14), 2494-2511.

DENG, Y., PUNG, B. 2010. Conserving Disk Energy in Virtual Machine Based Environments by Amplifying Bursts. *Computing,* 87 (3).

DOUGLIS, F., KRISHNAN, P., AND MARSH, B. 1994. Thwarting the Energy-Hungry Disk. In *Proceedings of the Winter USENIX Conference*, 292-306.

EMC. 2009. EMC in Major Storage Performance Breakthrough; first with Enterprise-Ready Solid State Flash Drive Technology. http://www.emc.com/about/news/press/us/2008/011408-1.htm

FLASH SSD CHARTS. 2009. http://www.tomshardware.tw/charts/flash-ssd-charts-2008/benchmarks,28.html.

FUJITSU. 2009. FUJITSU NAND Flash Memory. MBM30LV0128 datasheet.

GANGER, G. 2001. Blurring The Line Between Oses and Storage Devices. Technical report, Carnegie Mellon University.

GEIST, R., AND DANIEL, S. 1987. A Continuum of Disk Scheduling Algorithms. *ACM Transactions on. Computer Systems*, 5(1), 77–92.

GNIADY, C., BUTT, A. R., HU, Y. C., AND LU, Y. 2006. Program Counter-Based Prediction Techniques For Dynamic Energy Management. *IEEE Transactions on Computers,* 55(6), 641-658.

GREENAWALT, P.M. 1994. Modeling Power Management For Hard Disks. In *Proceedings of the Conference on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems*, 62-66.

GURUMURTHI, S., SIVASUBRAMANIAM, A., KANDEMIR, M., FRANKE, H. 2003. Reducing Disk Power Consumption in Servers with DRPM. *IEEE Computer*, 36(12), 59-66.

GURUMURTHI, S., SIVASUBRAMANIAM, A., NATARAJAN, V. K. 2005. Disk Drive Roadmap From The Thermal Perspective: A Case For Dynamic Thermal Management. In *Proceedings of the 32nd Annual International Symposium on Computer Architecture (ISCA 2005)*, 38-49.

GURUMURTHI, S., SIVASUBRAMANIAM, A. 2006. Thermal Issues in Disk Drive Design: Challenges and Possible Solutions. *ACM Transactions on Storage*, 2(1), 41-73.

GURUMURTHI, S. 2007. Should Disks Be Speed Demons Or Brainiacs? *ACM SIGOPS Operating Systems Review*, 41(1), 33-36.

HARD DISK DRIVE WITH MULTIPLE SPINDLES. http://www.freepatentsonline.com/20060044663.html

HEATH, T., PINHEIRO, E., HOM, J., KREMER, U., AND BIANCHINI, R. 2004. Code Transformations For Energy-Efficient Device Management. *IEEE Transactions on Computers*, 53(8), 974-987.

HELMBOLD, D. P., LONG, D. D. E., SCONYERS, T. L., AND SHERROD, B. 2000. Adaptive Disk Spin-Down For Mobile Computers. Mobile Networks and Applications,15(4), 285-297.

HERBST, G. IBM's drive temperature indicator processor (Drive-TIP) helps ensure high drive reliability. IBM Whitepaper, 1997.

HITACHI. 2009. Hitachi Global Storage Technologies – HDD Technology Overview Charts. http://www.hitachigst.com/hdd/technolo/overview/storagetechchart.html.

HOFRI, M. 1980. Disk Scheduling: FCFS vs. SSTF Revisited. *Communications of the ACM*, 23(11), 645-653.

HSU, W.W., AND SMITH, A.J. 2004. The Performance Impact of I/O Optimizations and Disk Improvements. *IBM Journal of Research and Development*, 48(2), 255–289.

HSIEH, J., KUO,T., CHANG, L. 2006. Efficient Identification Of Hot Data For Flash Memory Storage Systems. *ACM Transactions on Storage*, 2(1), 22-40.

INTEL. 2009. Intel NAND Flash Memory. JS29F16G08FANB1 datasheet.

ITRS07. International Technology Roadmap for Semiconductors. 2007 Edition.

JEPPESEN, J., ALLEN,W., ANDERSON,S., PILSL,M. 2001. Hard Disk Controller: The Disk Drive's Brain And Body. In *Proceedings of the International Conference on Computer Design: VLSI in Computers & Processors*, 262-267.

JIANG, W., HU, C., ZHOU, Y., KANEVSKY, A. 2008. Are Disks The Dominant Contributor For Storage Failures? A Comprehensive Study of Storage Subsystem Failure Characteristics. In *Proceedings of the 6th USENIX Conference on File and Storage Technologies*, 2008.

KAREDLA, R., LOVE, J.S., WHERRY, V. K. 1994. Caching Strategies to Improve Disk System Performance. *Computer*, 27(3), 38 - 46.

KANG, J., KIM, J., PARK,C., PARK , H., LEE, J. 2007. A Multi-Channel Architecture for High-Performance NAND Flash-Based Storage System. *Journal of Systems Architecture*, 53(9), 644-658.

KIM, H., AND AHN, S. 2008. BPLRU: A Buffer Management Scheme For Improving Random Writes In Flash Storage. In *Proceedings of the 6th USENIX Conference on File and Storage Technologies (FAST2008).*

KIM, Y., GURUMURTHI, S., SIVASUBRAMANIAM, A. 2006. Understanding the Performance-Temperature Interactions In Disk I/O of Server Workloads. In *Proceedings of the 12th International Symposium on High-Performance Computer Architecture*, 176-186.

LAWTON, G. 2006. Improved Flash Memory Grows In Popularity. *IEEE Computer*, 39(1), 16-18.

LEE, J., PARK,G., KIM, S. 2005. A New NAND-Type Flash Memory Package with Smart Buffer System For Spatial And Temporal Localities. *Journal of Systems Architecture*, 51(2), 111-123.

LI, K., KUMPF, R., HORTON, P., AND ANDERSON, T. E. 1994. Quantitative Analysis of Disk Drive Energy Management in Portable Computers. In *Proceedings of the USENIX Winter Conference*, 279–291.

LU, Y., AND MICHELI, G.D. 1999. Adaptive Hard Disk Energy Management on Personal Computers. In *Proceedings of the IEEE Great Lakes Symposium*, 50-53.

LUMB, S. W., SCHINDLER, J., GANGER, G. R., NAGLE, D. F. 2000. Towards Higher Disk Head Utilization: Extracting Free Bandwidth From Busy Disk Drives. In *Proceedings of the Fourth Symposium on Operating Systems Design and Implementation (OSDI)*, 87–102.

MESNIER, M., GANGER, G.R., RIEDEL, E. 2003. Object-Based Storage. *IEEE Communications Magazine,* 41(8), 84 – 90.

METER, R. V. 1997. Observing the Effects of Multi-Zone Disks. In *Proceedings of the USENIX Annual Technical Conference*,19-30.

NEZU, T. 2009. HDD Surface Recording Density to Exceed 1Tb/inch2 in 2011. http://techon.nikkeibp.co.jp/english/NEWS_EN/20090604/171260/.

NG, S. W. 1998. Advances in Disk Technology: Performance Issues. *Computer*, 31(5), 75-81.

PAPATHANASIOU, A. E., AND SCOTT, M. L. 2004. Energy Efficient Prefetching and Caching. In *Proceedings of the USENIX Annual Technical Conference*.

PERPENDICULAR RECORDING. 2009. http://en.wikipedia.org/wiki/Perpendicular_recording.

POLTE, M., SIMSA, J., GIBSON, G. 2008. Comparing Performance Of Solid State Devices And Mechanical Disks. In *Proceedings of the 3rd Petascale Data Storage Workshop held in conjunction with Supercomputing'08.*

RIEDEL, E., FALOUTSOS, C., GIBSON, G A., NAGLE, D. 2001. Active Disks for Large-Scale Data Processing. *Computer*, 34(6), 68-74.

RISKA, A., RIEDEL,E., IREN,S. 2004. Adaptive Disk Scheduling For Overload Management. In *Proceedings of the 1st International Conference on The Quantitative Evaluation of Systems (QEST04)*,176-185.

RUEMMLER, C., AND WILKES, J. 1994. An Introduction to Disk Drive Modeling. *Computer*, 27(3), 17-28.

SAMSUNG. 2006. Samsung To Unveil First Commercial, Hybrid Hard Drive Prototype For Windows Vista At Winhec. http://www.samsung.com.

SAMSUNG. 2009a. Samsung NAND Flash Memory. K9F6408U0A-TCB0 Datasheet.

SAMSUNG. 2009b. Samsung NAND Flash Memory. K9NBG08U5A datasheet.

SAMSUNG. 2009c. Samsung Unveils 32GB Flash Hard Drive. http://news.zdnet.co.uk/hardware/

SAMSUNG. 2009d. Samsung Introduces Working Prototype Of PRAM. http://www.eetimes.com/news/semi/showArticle.jhtml?articleID=192700709

SCHROEDER, B., GIBSON,G.A. 2007. Disk Failures In The Real World: What Does An MTTF Of 1,000,000 Hours Mean To You? In *Proceedings of 5th USENIX Conference on File and Storage Technologies (FAST07).*

SCHINDLER,J., SCHLOSSER, S.W., SHAO,M., AILAMAKI, A., GANGER,G.R. 2004. Atropos: A Disk Array Volume Manager for Orchestrated Use of Disks. In *Proceedings of the 3rd UNENIX Conference on File and Storage Technologies*, 159-172.

SCHINDLER, J., GRIFFIN, J. L., LUMB, C. R., AND GANGER, G. R. 2002. Track Aligned Extents: Matching Access Patterns To Disk Drive Characteristics. In *Proceedings of Conf. on File and Storage Technologies (FAST02),* 259-274.

SCHLOSSER, S. W., GRIFFIN, J. L., NAGLE, D. F., GANGER, G. R. 2000. Designing Computer Systems With MEMS-Based Storage, In *Proceedings of the 9th International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS)*,1–12.

SEAGATE. 2009. Product Manual: Cheetah X15 36LP Disk Drive. 2001. http://www.seagate.com/.

SEEK DISTANCE. 2009. Seek Distance Dependent Variable Max VCM Seek Current To Control Thermal Rise In VCM's. http://www.patentstorm.us/patents/6724564-description.html.

SMITH, A. J. 1978. On The Effectiveness of Buffered and Multiple Arm Disks. In *Proceedings of the 5th annual symposium on Computer architecture*, 242-248.

SON, S. W., KANDEMIR, M., AND CHOUDHARY, A. 2005. Software-Directed Disk Power Management For Scientific Applications. In *Proceedings of the 19th IEEE International Parallel and Distributed Processing Symposium*.

SPEC-POWER AND PERFORMANCE. 2009. http://www.spec.org/power_ssj2008/.

SRI-JAYANTHA, M. 1995. Trends In Mobile Storage Design. In *Proceedings of the International Symposium on Low Power Electronics*, 54–57.

STAELIN, C., AND GARCIA-MOLINA, H. 1990. Clustering Active Disk Data to Improve Disk Performance. Tech. Rep. CS-TR-283-90. Dept. of Computer Science, Princeton University.

STROM, B.D., LEE, S., TYNDALL, G. W., AND KHURSHUDOV, A. 2007. Hard Disk Drive Reliability Modeling and Failure Prediction. *IEEE Transactions on Magnetics*, 43(9), 3676-3685.

SUBRAMANIAN, C. K., ANDRE, T. W., NAHAS, J. J., GARNI, B. J., LIN, H. S., OMAIR, A., AND MARTINO, J. W. L. 2004. Design Aspects of a 4 Mbit 0.18μm 1T1MTJ Toggle MRAM Memory. In *Proceedings of the IEEE International Conference on Integrated Circuit Design and Technology*,177-181.

SULLIVAN, R. F. 2000. Alternating Cold And Hot Aisles Provides More Reliable Cooling For Server Farms. In Uptime Institute.

TEXAS MEMORY SYSTEMS. 2009. RamSan-20. http://www.ramsan.com/products/ramsan-20.htm.

THOMPSON, D. A., AND BEST, J. S. 2000. The Future Of Magnetic Data Storage Technology. *IBM Journal of Research and Development*, 44(3), 311-322.

TRAINOR, M. 2007. Overcoming Disk Drive Access Bottlenecks with Intel Robson Technology. http://www.intel.com/technology/magazine/computing/robson-1206.htm.

ULTRASTAR 15K147. 2009. Ultrastar 15K147 Hard Disk Drives Specifications. http://www.hitachigst.com/hdd/support/15k147/15k147.htm

WHAT ARE MEMRISTORS? 2009. http://www.memristor.org/reference/13/what-are-memristors.

WORTHINGTON, B. L., GANGER, G. R., PATT, Y. N. 1994. Scheduling Algorithms For Modern Disk Drives. In *Proceedings of the 1994 ACM SIGMETRICS conference on Measurement and modeling of computer systems*, 241-251.

YANG, J., SUN, F. 1999. A Comprehensive Review of Hard-Disk Drive Reliability, In *Proceedings of Annual Reliability and Maintainability Symposium*, 403-409.

YAFFS. 2009. Yet Another Flash File System. http://www.yaffs.net/.

ZEUS. 2009. ZEUSIOPS 2GB Fiber Channel 3.5-inch Solid State Drive Product Manual.

ZHENG, J., GUO, G., WANG, Y. 2005. Identification and Decentralized Control of A Dual-Actuator Hard Disk Drive System. *IEEE Transactions on Magnetics*, 41(9), 2515-2521.

ZHU, Q., CHEN, Z., TAN, L., ZHOU, Y., KEETON, K., WILKES, J. 2005. Hibernator: Helping Disk Arrays Sleep Through The Winter. In *Proceedings of the 20th ACM Symposium on Operating Systems Principles 2005 (SOSP 2005)*, 177-190.

Received July 2009; accepted March 2010