Studying and modeling the influence of the remapping technology on hard disk drives

Stanislav Semakin

May 22, 2006

University of Joensuu
Department of Computer Science
Master's Thesis

Abstract

Contemporary storage systems today are multifunctional and complicated, constantly storing an increasing amount of data. They consist of many physical disks which in turn are rapidly developing devices. With the growing complexity and increasing amount of data in disks, the requirements for self-monitoring and self-testing disk subsystems are becoming more strict. One of the failure-detecting subsystems is a remapping mechanism. It keeps track of defective sectors and remaps them to the specially allocated disk areas. The aim of this thesis is to study how remapped sectors impact the disk seek time.

This work considers basic methods of sector remapping since there is a lack of information on this topic. The paper describes how it is implemented on the hardware level and how it interacts with other components of disk devices. There is also an attempt taken to systematize the technology in order to draw a global view.

In order to determine the impact of remapped sectors on disk performance, an analytic model was built which predicts device behavior in terms of seek time within a relative error ranging from 2% to 18%. The model was validated on a real disk drive with synthetic workloads using a tool specially created for this purpose.

To find out how different remapping schemes impact on disk performance there are tests done in real conditions with three SCSI hard disks. The tests were performed with synthetic workloads and a remapping mechanism activated inside the devices.

Basic contributions of this work to the area of hard disk drives research include the following:

- The description, systematization and test results of the technology can be useful in future works on this subject.
- The model can be integrated with similar modeling systems to take the remapping technology into account.

Contents

List of Figures

Chapter 1

Introduction

The main subject of this work is the technology of sectors remapping for hard disk drives. A magnetic disk consists of double-sided platters, a platter's surface contains tracks, tracks are divided into sectors, one sector is a minimum portion of data which can be written or read. All the sectors of the disk are of the same size. This is usually equivalent to 512 bytes of user data and a few tens of the bytes for metadata (correction codes, checksums and etc.). For some of the disks it can be changed by expense of the metadata length. Each sector has its own unique number called Logical Block Address (LBA). There is also a read-write head moving over magnetic surfaces, one disk head serves one magnetic surface. Since hard drives consist of many constantly working mechanical units, they still remain one of the unreliable devices in current electronic machines. Defective sector is a sector which cannot be read or written in normal conditions (there are limits of a different nature e.g. time limit for command execution or maximum retry counter). Disk mechanics and electronics try to detect such sectors beforehand and assign a spare sector to the LBA which has been in use by the defective sector – this is called a process of remapping defective sectors. Defective sectors can appear as at once, normally it takes place because of a mechanical damage (e.g. read-write head 'falls' onto the disk surface), making up the whole defective areas; so one by one, it might take place on surfaces which are worn out due to time or extremely frequent read/write operations.

All the remapped sectors are recorded on the specially created lists. Generally, hard disks hold two lists of remapped sectors - Primary List (P-List) and Grown List (G-List). P-list is intended for manufacturers to fill in. It is done during the testing process of the

devices before the actual shipment. G-list is enlarged over the time of disk usage. Spare sectors can be allocated in different places of the platters. There can be several sectors per each track or per each cylinder or zone. Spare sectors can be scattered over the disk or concentrated in one particular place. Implementation of the mechanism varies depending on the manufacturer and not standardized.

In Chapters 2 and 3 the disk drive mechanisms and remapping technology are introduced. Chapter 2 contains a brief overview of the disk contents such as heads, cylinders, tracks and other disk internals. Chapter 3 gives a more detailed description of the defective sectors remapping technology. One of the contributions of this work is an attempt to systematize information regarding the subject. Since every manufacturer has its own technology and implementation, the information is not generally published neither in the technical books nor in the device manuals or technical papers. More technical information is provided on SCSI specifications regarding how to access P and G-Lists, do single reassigning or low-level format with both P-List and sector size changed, as well as choose access-level to the sectors in the SCSI hard disks [Com97, Com04, Com05c, Com05a, Com05b] (last versions of these papers and SCSI Standards Architecture can be found here [Lohb, Loha]). For a particular implementation of the remapping technology one needs to read SCSI disks manuals, such as [Fuj00, IBM00, Sea00]. Since there are no published standards for the remapping technology, some information can also be found on the Internet, on the unofficial forums and pages of the disk repairing shops. Some explanation on the technology can be found here [Gut96].

In this work I also try to determine how remapping technology affects disk performance and disk seek time in particular. Due to this reason I develop an analytic model which calculates disk seek time versus the number of remapped sectors. As an implementation of the remapping mechanism a generic scheme is used in which the only spare area for the remapped sectors is at the end of sector space placed in the last cylinders. Modeling is performed for the entire disk space. I also take tests on real hard disk drives and see how implementation of the mechanism affects sequential requests of various length. In Chapter 4 the basic concepts of disk performance are introduced, and the terms needed to understand the model, the tests and factors that influence it.

The model is divided into two parts. The first part deals with single sector requests distributed in a random uniform fashion over disk sector space. The second part is based on the first one and deals with chains of consecutive sectors which are requested in the same

manner as the single sector requests. The model shows how sector seek time behaves with a gradually growing number of remapped sectors. Remapped sectors are also randomly and uniformly distributed over the disk one by one. This model was validated on IBM DDYS-T36950 SCSI hard drive with a range of average relative error of 2.46% to 18.18% depending on the case. Please refer to Chapter 5 for more detailed results. The model is built upon the one created by Elizabeth Shriver in her PhD thesis [Shr97] since it provides an exhaustive explanation and good results. Therefore this work concentrates only on the subject of sectors remapping. Classical work on disk modeling is considered to be [RW94], please refer also to the following papers [TCG02, And01, WAA+04, Var00]. None of these works take into account remapping mechanism, therefore the model built in this thesis can be used for future modeling of disk systems.

Tests performed on real disks and described in Chapter 6 consider three hard drives where remapped sectors were created with a special SCSI command:

- IBM DDYS-T36950M;
- Fujitsu MAJ3182MC;
- COMPAQ BD009635C3.

As previously mentioned, the tests show how seek time reacts to the remapped sectors when reading sequential requests with lengths of 100 KB, 1 MB and 10 MB. Remapping was made with runs of lengths of 1, 5 and 20 sectors.

Chapter 2

Magnetic disks

Modern hard disk drives are complicated devices which have a long history. Today, these devices combine such sophisticated electronic and mechanical parts that it is difficult to name them all. In this chapter basic historical and technological aspects of disks are introduced. Section 2.1 provides insight on how recording started, and where it originated from, who made the first disk drives as we know them today and how it has grown. In Section 2.2 the basic inner organization and technologies of a disk are discussed.

2.1 History of magnetic recording

Magnetic recording was invented to record sound and by 1941 it was adopted to store data [Sta03]. The first machine successfully used to store digital data was ENIAC in 1947. At that time tapes served to store digital information. Since then and up to today magnetic tapes remain one of the ways to store data. The technologies, that allow to record data onto the tape, have changed as well as data capacity and record density but the idea of a tape remains the same along with its main disadvantage – sequential access to data. A few years later, in 1955 a group of scientists from IBM headed by Reynold B. Johnson created a first magnetic disk file, IBM 350 RAMAC [1] (Random Access Method of Accounting Control) – a relatively simple random access storage system. The disk file occupied about 8,5 cubic meters. IBM started its breath-taking leadership with ENIAC and RAMAC in the storage system industry.

[1] RAMAC-350 had 50 platters, 61 cm in diameter, and 5 MB of total capacity, with 1200 RPM and access time of 1 second.

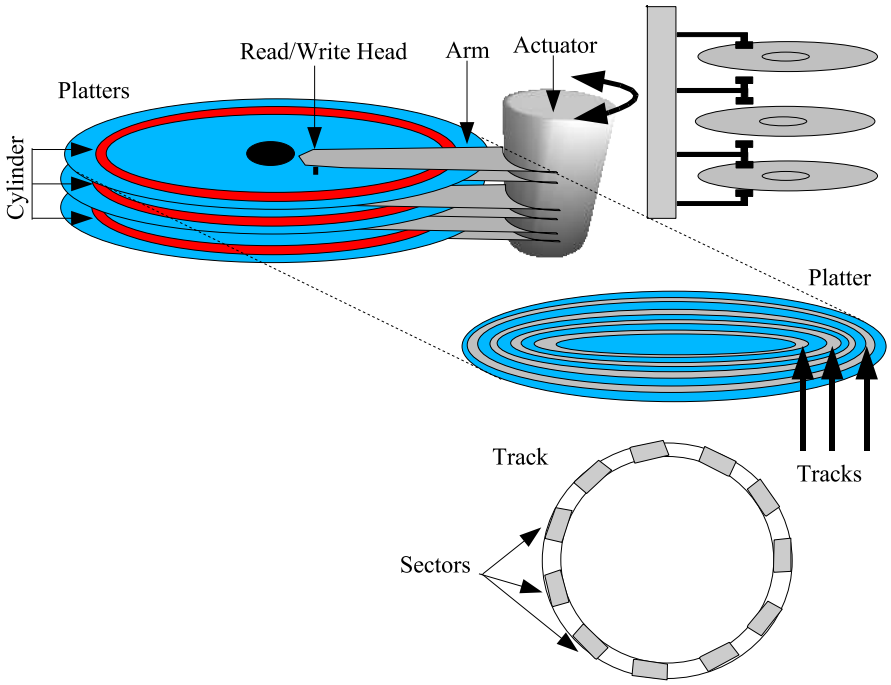


Figure 2.1: Disk is organized into platters, tracks, and sectors.

In fact the first and real breakthrough is considered to have emerged with air-bearing read/write heads in a disk. A head in such kind of disks held onto an air cushion created by the fast-moving disk surface. Such a construction allowed the head to follow all the imperfections on the disk surface and moreover be very close to the surface. Since then many more benefits of the construction have been discovered along with constantly improving precision. Nowadays heads move 0.05 to 0.07 micrometers above the surface, whereas in the RAMAC drive they were 25.4 micrometers away.

In 1970 the first floppy disk drive was introduced to but for PC it only became popular about 10 years later when it was originally used to hold microcode for the IBM 370 series. Around 1973 a disk with a so-called Winchester disk design appeared. It made the second breakthrough. Integrated circuits allowed to place the disk controllers and the electronics, which was controlling the disk arms, into a frame along with the rest mechanics what caused lowering the prices of the disk drives. This integration in turn removed the need to share the electronics making non-removable disks more economical. On the other hand Winchester disks benefited also from the system compacting since all

the components could be placed into a single case, this greatly improved the electronics controlling problems as well as increased areal density. The moniker "30-30"[2] for the disk was given to the first Winchester disk because of its two spindles, each with a capacity of 30 MB. By the mid-1980s, Winchester disks almost completely replaced removable disks.

A 3.5-inch drive is still the market leader, and there is a need for 2.5-inch drives for the laptop computers. These days personal video recorders (PVRs), photocameras, personal storage devices etc. require more density, more capacity. New technologies such as perpendicular recording are becoming more real which makes 1-inch 138 Gbit/square inch possible, but new kinds of RAM such as Flash started its competition and now it is almost impossible to predict what future is awaiting the magnetic disks.

2.2 Basic components and fundamental technologies of hard disk drives

A magnetic disk consist of a number of *platters*. These platters are made of metal or glass disks covered with magnetic recording material on both sides. The platters rotate at a certain speed, currently it varies from 3600 to 15000 revolutions per minute (RPM). While its diameters vary from 0.85 to 3.5 inches. Tendency for the disk capacity is observed to be as follows: the bigger disk the higher the performance, the smaller disk the lower its price.

Each magnetic surface of these platters is divided into circles called *tracks*. A set of these tracks, which are equally distant from the platters center and stacked up, makes up a *cylinder* (see Figure 2.1). A track consist of sectors – the smallest units to be written or read. Normally the size of a sector stores 512 bytes of user data but in contemporary disk systems this number can be changed. In addition there is also metadata written along with user data (sector number, error correction code, a gap size etc.) making the real sector size slightly bigger. A disk surface holds about $\approx$ 5000 to $\approx$ 80000 tracks, a track typically contains $\approx$ 100 to $\approx$ 1200 sectors. A *Read/write head* is fixed to a movable *arm* which slides over the disk surface, each surface has its own arm with a head. The name of the head indicates what it is intended for. All the arms are connected so that they access a cylinder at the same time.

[2]Popular sport rifle in America, the Winchester 94, "30-30" was given after the caliber of its cartridge.

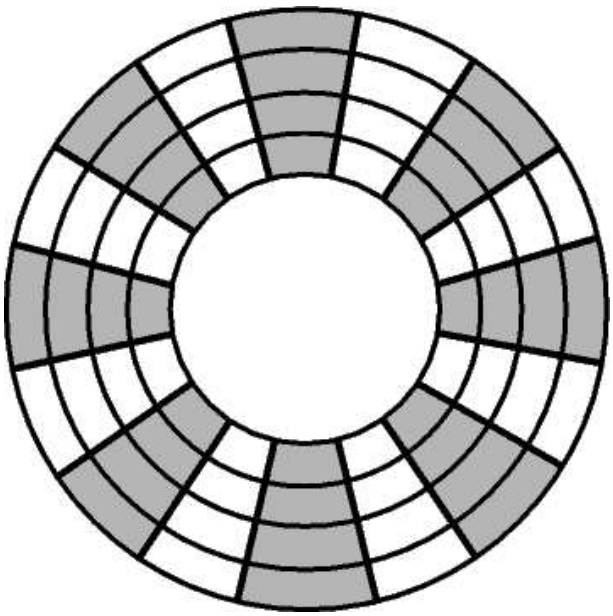


Figure 2.2: Old hard drives had the same number of sectors on the outer track as on the inner ones.

Hard disk drives of the past had the same number of sectors on the inner tracks (those which are nearer to the center) of the platters as on the outer ones. The velocity of the imaginary points increases as we move from the inner tracks of a platter to the outer tracks, therefore bits of the outer tracks slide faster past the read-write head. For the old disks it was a problem since the read-write head could access the data at a constant rate. In order to do so, the space between recorded segments was different for middle parts and side ones of the same platter – it was increasing along with increasing the distance from the middle. The information then could be read and written at the same rate by rotating the disk at a fixed speed called *constant angular velocity (CAV)* (the disk layout with technology of CAV is depicted on Figure 2.2). The advantage of this technology was the fact that individual blocks of data could be directly addressed by track and sector. The head movement from its current location to a specific address took only a short movement of the head to a specific track and a short wait for the proper sector to spin under the head. The disadvantage of CAV is based on the fact that the outer tracks are longer, but the amount of data stored on those is the same as on the inner tracks.

Such a disk drive is limited by the maximum recording *density* (which is measured in

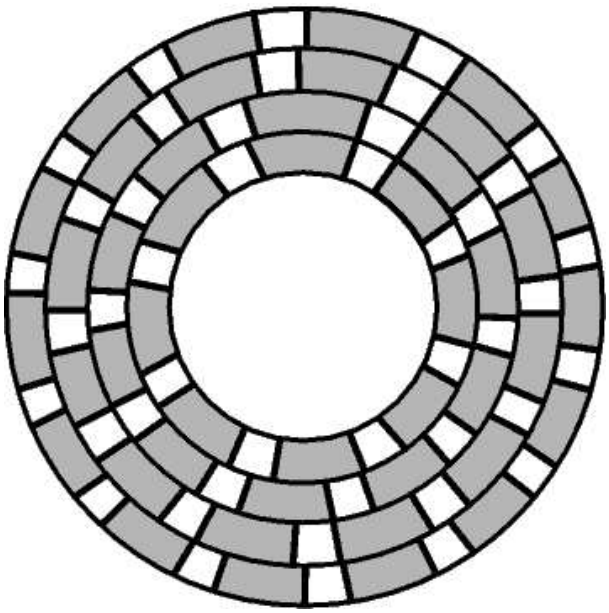


Figure 2.3: In current disks, the inner tracks and the outer tracks are recorded with different density.

bits per linear inch) of the innermost tracks since it increases from the outer tracks to the inner ones. The CAV was a real restriction from the perspective of the disk capacity. Modern hard disks use a different technology called *multiple zone recording*, which is intended for increasing the density of the outer tracks. The idea is to divide a disk platter into a number of zones (normally 16). The greater the distance of the zone from the center of the platter, the more bits and therefore sectors it can contain (see Figure 2.3 which depicts the idea). It gave a significant capacity growth for the disk platters with the expense of more complicated disk mechanics and electronics (e.g. when the disk head moves from one zone to another, a track bit length changes causing a change in time parameters for reading and writing). The technology for recording more sectors on the outer tracks than on the inner tracks is called *constant bit density*, which is the standard today. The idea behind this is based on the fact that the outer tracks are recorded at a much lower density than the inner ones, but these outer tracks are still longer.

2.3 Summary

So far a short introduction to the disks, their history and basic organization aspects have been discussed. At first glance, there seems nothing can fail or go broken but in fact these devices are rather fragile, and there is a sophisticated mechanism to prevent most of the disk failures inside. One of the parts of this mechanism is a system of remapping defective sectors, which this thesis concentrates on. In the next chapter, the technical aspects of this system will be discussed.

Chapter 3

Defects and technology of remapping

In this chapter, the basic conception of the defect and error management is discussed. Section 3.2 considers how disk space is divided and what data is stored in these areas. A low level track and sector formats are illustrated. Section 3.3 explains what defects are and how remapping is activated in disks. At the end of the chapter a systematized overview for the remapping technology is provided.

3.1 Notion of hard drive defective area

Hard drives remain one of the most unreliable and complicated computer components. Besides electronic components, it contains constantly working mechanicals units. Mechanical parts tend to wear out, defects of different nature emerge on the platters' surfaces. In other words there will be defective areas. There are many reasons why these areas appear but this paper only deals with such a type of disk fault. A *defective area* is a portion of disk surface where hardware errors occur while reading or writing data, what results in data storing impossible. In fact each hard drive contains defective zones from the very beginning before it is shipped. The technologies of HDD fabrications are still not so consummate to produce it completely clean. This is the reason why all the hard drives are immediately and accurately verified after the production process by the manufactures themselves. During the verification process bad sectors show up and are recorded on the specific tables called *defect lists*. The process of creating entries on the defect lists is called

remapping[1].

All the modern hard drives contain two main defect-lists: the first one is the list created by the manufacturer of the device and is called P-List (Primary); the second one is called G-List (Grown) and is filled up during the usage period of the storage as new bad sectors appear. Some hard drives also contain a defect-list of the system area. If a defective sector occurs then it is recorded on one of the lists and a spare sector is activated instead of a bad one. Different manufacturers provide different error recovery schemes and algorithms. The algorithms may involve different layers of recovery, from managing a single logical block where information is stored to a system which handles the whole cylinders. Each level may consist of multiple steps, where a step is defined as a recovery function involving a single re-read or re-write attempt.

3.2 Disk organization and units

In order to understand remapping and how it works, disk organization needs to be considered first. This section pays attention to a disk organizations: it introduces what areas a disk is divided into and what these areas are intended for. It also illustrates track and sector formats and how complicated their structure is.

Since the information dedicated to this topic is not highly published, as an example a Fujitsu SCSI hard drive specification [Fuj00] is analyzed. Fujitsu documentation provides a sufficiently exhaustive description of the low-level data and defect management and can be taken as a core in our explanation.

3.2.1 Disk Space

The entire disk space can be divided into the following three areas:

- User space: Storage area for user data;
- Internal test space: Reserved area for diagnostic purposes;
- System space: Area for an exclusive use of the disk drive itself.

In the following subsections, these items will be discussed in more detail in order to get a better understanding of the disk division into these areas.

[1] In the technical specifications there are also such definitions as *reallocation* and *reassigning*.

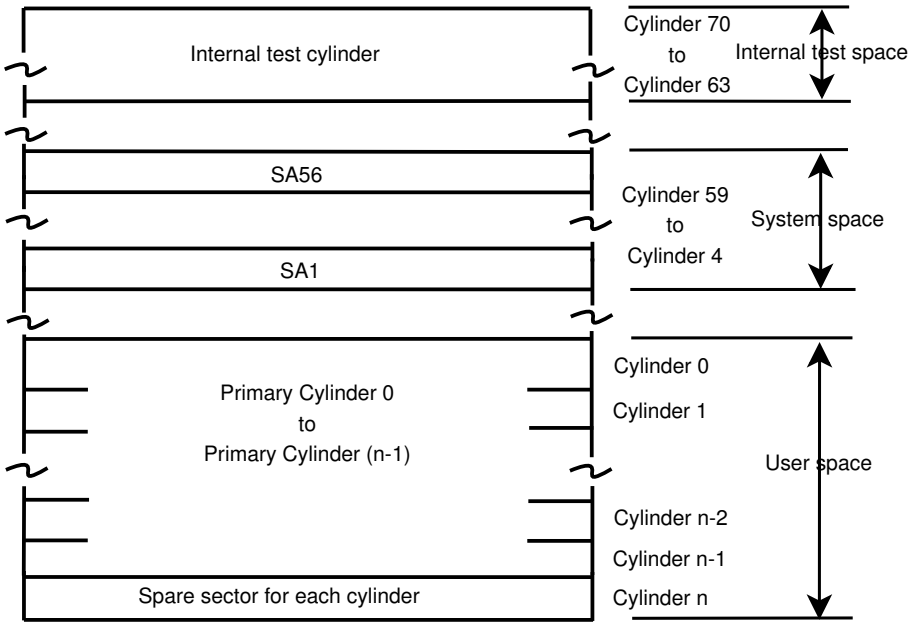


Figure 3.1: Cylinder map for the disk area.

User space

The user space serves to store user data. This space is organized into *logical data blocks* which are accessed using a logical data block addressing method. Each manufacturer may have its own physical medium structure but logical blocks addressing is a standardized way of accessing the data on a hard drive by a user application. During the low level formatting process, logical data block units are related to physical space (this process is unique for each manufacturer). Besides user space with directly accessible user logical blocks there is an alternate area which also belongs to the user space but cannot be accessed directly by a user application. Access to the sectors which are allocated as *alternate blocks* in the alternate space is made automatically by means of *sector slip treatment* or *alternate block treatment*[2].

Internal test space

The Internal test space serves to perform read/write self-diagnostic tests. There is no direct access for the test space since it is used only by a hard drive's mechanism. Data

[2]Those are technics for performing remapping of the sectors, they are discussed in Section 3.3.

block length in this area is predefined and cannot be changed by the user.

System space

System space is needed to store various system information for the hard drive to read it at power-on or during execution of a specific command. There is no direct user interface to access this area either. Data block length in this area is always 512 bytes. This space is used by the hard drive's electronics only and intended for storing the following information:

- Defect list (P and G lists);
- MODE SELECT parameter [Fuj00] (saved values);
- Statistical information [Fuj00] (log data);
- Controller control information.

For safety purposes this information is duplicated in several different locations. This area is also called SA space. Figure 3.1 depicts an example of cylinder allocation for the considered areas[3].

3.2.2 Track format

As mentioned earlier, a track consists of sectors, all the sectors are allocated one after another in a track. In Figure 3.2 the sector order is shown. The amount of bytes in a physical sector varies depending on the length of the data block and the number of sectors per track. The data block length affects the length of the unused area (G4 depicted in the figure).

On the physical level, tracks and sectors are described as electric pulses. This is how hard drive electronics recognize these units. The mechanical parts simply access the media with some frequency which is why those units are measured in pulses not in bytes. The interval for one sector pulse is decided by multiple of 20 MHz free running frequency. Since disks are divided into zones, this clock is not equal to the interval of the byte clock, therefore physical sector length cannot be described with a byte length.

[3]All the numerical data in this chapter is given to make up a complete picture and true for the hard drives described in [Fuj00] unless otherwise is said.

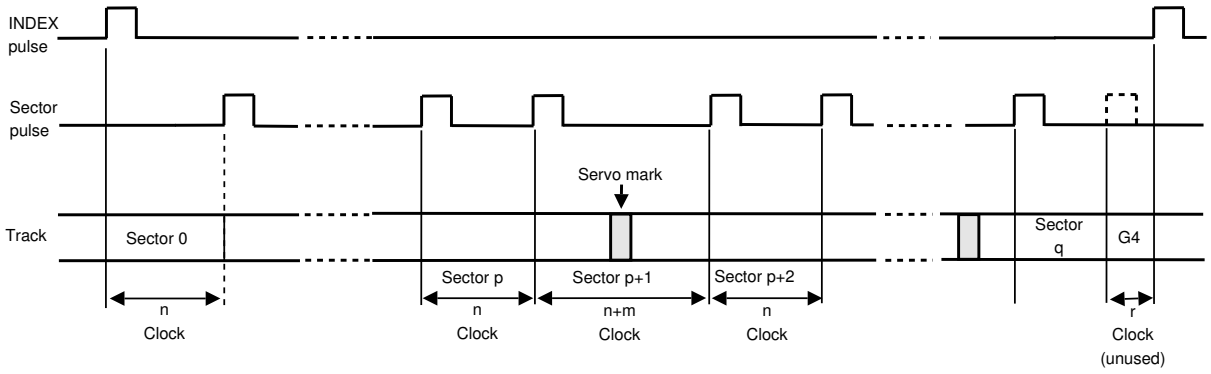


Figure 3.2: Track format.

3.2.3 Track skew and cylinder skew

Both track and head switching require a slice of time. In order to avoid an extra revolution, the first logical data block in each track is shifted by the number of sectors corresponding to the switching time. This number is called track skew in case of head switching and cylinder skew in case of cylinder switching. Figure 3.3 shows the allocation of the sectors in each track and tracks in a cylinder.

At the head switching location in a cylinder, the first logical data block in track *t+1* is allocated at the sector position which locates the track skew behind the sector position of the last logical data block sector in track *t*.

At the cylinder switching location, the first logical data block in a cylinder is allocated at the sector position which locates the cylinder skew behind the last logical sector position in the preceding cylinder. The last logical sector in the cylinder is allocated during formatting.

Since track skew and cylinder skew are managed for individual sectors, the logical data length affects the number of physical sectors (which is track and cylinder skew factors) needed to bring it together with switching time. This is up to the hard drive mechanism to automatically determine track skew factor and cylinder skew factor according to the specified logical data block length.

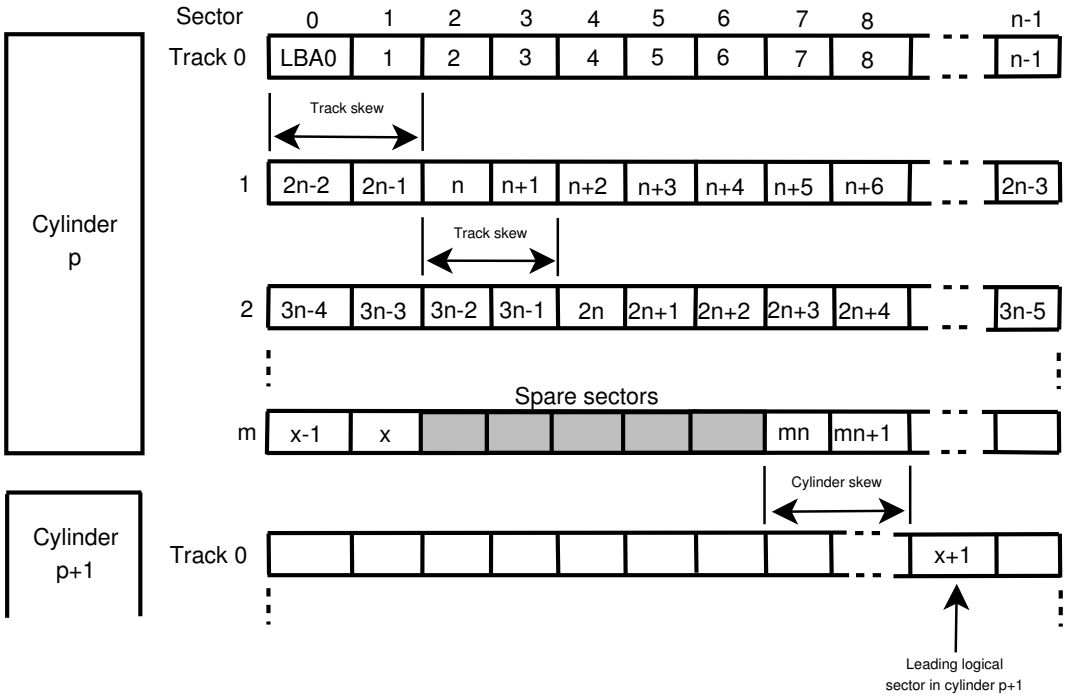


Figure 3.3: Track skew/cylinder skew. LBA - Logical data block address.

3.2.4 Sector format

Each sector on a track consists of an ID field, a data field and a gap field which separates them. Figure 3.2.4 depicts examples of a sector format.

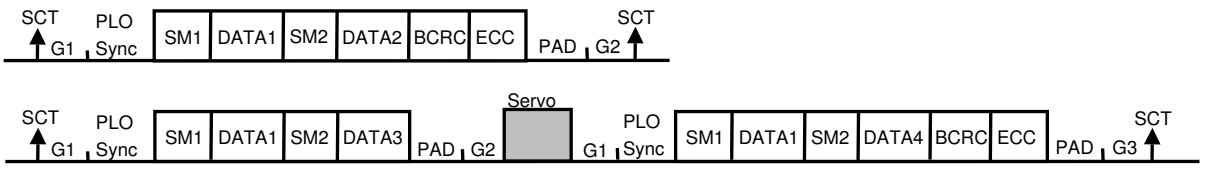
Let us now consider these fields in detail.

Gaps (G1, G2, G3). During first formatting (initializing) gap length is set to the values listed in Figure 3.4. These fields contain no pattern.

PLO Sync. This field contains a predefined pattern of '00' with the length listed in Figure 3.4.

Sync Mark (SM1, SM2). This field is intended for the indication of data field and contains a special pattern with the length listed in Figure 3.4.

Data (DATA1-DATA4). Data field contains user data. The length of the data field is determined as logical data block length which is specified with a parameter by



G1	4 bytes
G2	11 bytes
G3	8 bytes
PLO Sync	30 bytes
SM1	4 bytes
SM2	2 bytes
BCRC	4 bytes
ECC	40 bytes
PAD	6 bytes
DATA1	20 bytes
DATA2	492 bytes
DATA3	n bytes(0<= n<=472, n is a multiple of 4)
DATA4	(472-n) bytes

Figure 3.4: Sector format.

a special command. This parameter can be changed by the user. Even numbers between 512 and 528 bytes can only be specified as the length.

BCRC. There is a 4-byte error detection code stored in this field for the errors in the ID field. With this error code, a single burst error with a length of up to 32 bits for each logical block can be detected.

ECC. This is a 40-byte code. It allows to detect and correct errors on the fly in the data field. It is capable of correcting a single burst error up to 160 bit.

PAD. The '00' pattern is written in this field with the length shown in Figure 3.4. This field includes the variation by rotation and circuit delay untill reading/writing.

3.3 Defect management and remapping mechanism

By now the basic physical structure of hard drives was represented, technically describing disk areas, track and sector formats. In this section a better explanation for the defect lists is provided, what and how sectors are written on these lists, how and why remapping mechanism is activated.

3.3.1 Conditions for activating the remapping mechanism

Each time an error occurs while executing a hard drive operation, the disk performs error recovery procedures in order to attempt to restore the data. For instance, as already shown in Section 3.2.4, there is a sufficiently complicated sector format[4] – if there is an error in the read operation, disk controller tries to recover the data read using the aforementioned sector fields[5]. The error recovery procedures depend on the options previously set up in the error recovery parameters (some manufacturers allow the user to change the sector format length; number of recovery operations; time limits and etc.). If the error recovery procedure fails to restore the data after an unsuccessfully performed read or write operation, the accessed sector is remapped. Error recovery and defect management may involve the use of several disk firmware commands, which can be done as automatically so by a system administrator.

3.3.2 Defect lists

As previously said, all the information about reallocated sectors is stored on the special lists. A disk drive manages its defects only using these lists. If access to those ones is lost (e.g. the system area is damaged), then the disk behavior is unpredictable.

P list (Primary defect list). This list is recorded in the system space and contains information about defect areas of the disk which are found during initialization of the HDD. This list is placed on the disk before it is shipped by a manufacturer. All the information in this list is permanent.

[4]Pay attention to the fact that all the fields described are so-called 'published information'. Manufacturers have also their own ways to optimization the technologies.

[5]There are also other ways to recover information.

G list (Growth defect list). On this list there is information about defective areas that occurred in a hard drive after it has been shipped. This is the second list after the P list which is needed for the hard drive to manage its defects. The G list is stored in the system space. All the information is recorded on this list using the following:

- REASSIGN BLOCK command which specifies logical blocks to reassign directly by a user application;
- By means of an automatic alternate block allocation;
- Information specified as D list;
- Information specified as C list.

D list (Data defect list). There is a user command FORMAT UNIT needed for a user to manage the disk surface and the user area. The information supplied for this command containing the defect location is on this list. After the command has been performed the D list becomes a part of the G list.

C list (Logical unit certification list). This list contains defects detected by a hard drive during an optional certification process performed during FORMAT UNIT command. This list also becomes a part of the G list.

Normally, all of these 4 lists are implemented by a manufacturer but it is not mandatory, therefore some of those lists might not be found in some of the disk models. Generally, C list might be absent.

3.3.3 Alternate sector allocation

As long as there is no medium error, logical data blocks (or sectors) from the user space are accessed, and as we know, only after an error has occurred the process of remapping takes place. The alternate data block is allocated instead of a defective data block in a predefined area. Spare sectors to which alternate blocks are allocated can be in various areas of a hard drive depending on the remapping scheme. Some manufacturers allow the user to choose which area of the hard drive to use for the spare locations, and which remapping scheme to engage.

There exist two ways of treating the alternate block allocation:

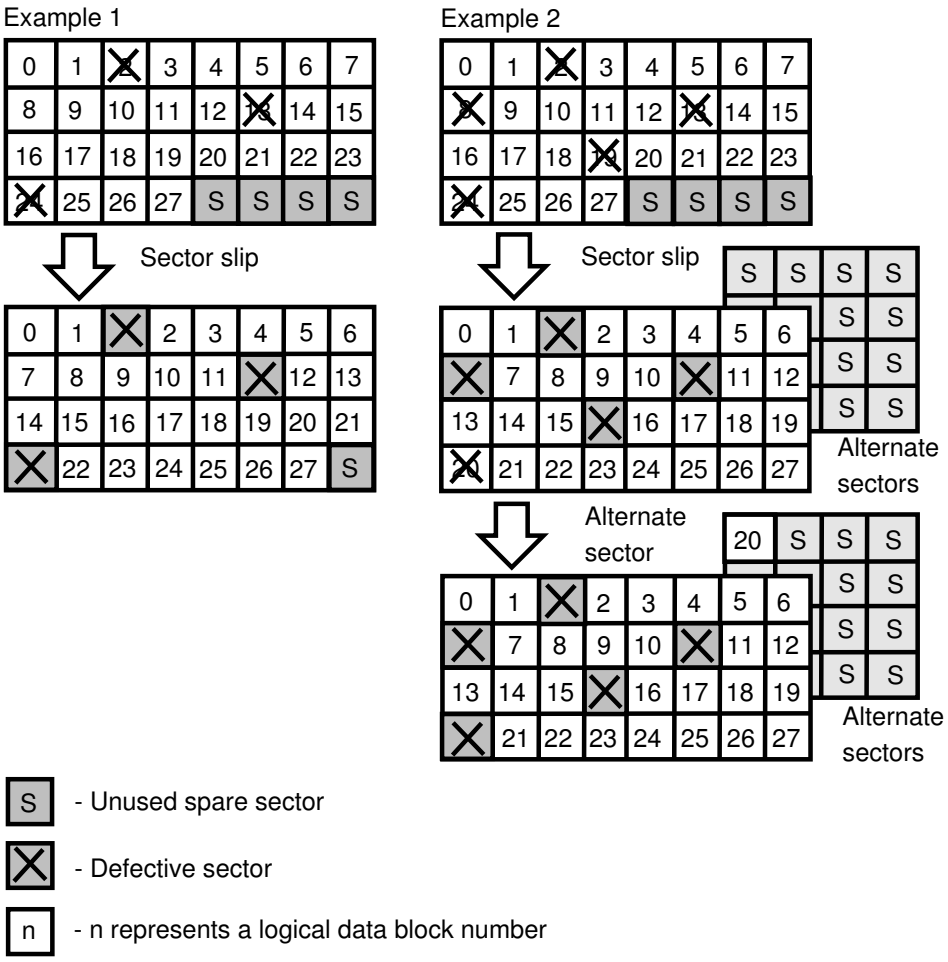


Figure 3.5: Defective sector treatment.

Sector slip treatment: Defective sectors are skipped and the logical data block corresponding to those sectors is allocated to the next physical sectors.

Alternate sector treatment: The logical data block corresponding to the defective sector is allocated to an unused spare sector in the alternate disk space.

In Figure 3.5, both methods are described. These two are used under certain conditions in different ways. In example 1, there are defective sectors which are slipped after reformatting the hard drive. Example 2 shows how reallocation takes place. Example 2 illustrates a hard drive where both methods are implemented – reallocation substitutes for the slipping process, if there is no more space for the slipping. Not all the manufactur-

ers support both methods on the fly, mainly reallocations occur for the reassigning, and slipping for the reformatting process.

There are several ways for a manufacturer to place spare area and treat defective sectors:

Method of a reserved sector: A spare sector is placed on each track of a hard drive. As soon as a defect is detected on a track it is reallocated to the spare area of the same track. The advantage of this method is that it is not actually reflected in efficiency loss. The disadvantage is that the general capacity of storage is prodigally used – a spare sector has to be on the track, even though a defective one is rare to appear. The second drawback of this method is inefficiency in case there is more than one defective sector on the track. There are different modifications of the method e.g. one spare sector per cylinder, but it does not increase efficiency significantly.

Method of a reserved track: Spare tracks are placed either in the beginning of the hard drive or at the outer area. After detecting the defect sector, the entire track containing it might be replaced with a reserved one or a single sector only. In this case, the heads of the device have to be constantly moved to the spare zone which is reflected in overall performance loss.

Method of track skip: The method of track skip also implies existence of the additional tracks in the outer work area. In this method the controller, after taking the real index of the track to work with, adds it to the number of the defects before this track (which are taken from the defect list) on the disk. In this case all the defective tracks are skipped, a shifting occurs to or from the center of the disk after adding a spare track to the work area. The obvious advantage of this method compared to the previous ones is the absence of the head movement to the spare zone. It results in the constant and stable speed.

The methods described above are only general ones. There are many more of them in various variations from different manufacturers. Since recently, the method of slipping the sector is becoming more and more widely used due to the evolution of technology allowing to do so on the fly. However, it is still complicated.

3.4 Summary of the remapping technology

In this chapter the technology of remapping was introduced. Generally, it discussed the basic points of hardware implementation. This section describes those more strictly in order to emphasize the main features. It helps in creating an analytical model of an HDD which will take the defective sectors expenses into account.

There are different areas for a hard drive to reserve spare space for the defective sectors. First, I characterize the remapping mechanisms by spare space. Placement options:

1. Spare sectors(a sector) can be placed in
   - a track/groups of tracks (for example N spare sectors for M user tracks);
   - a cylinder/group of cylinders (for example N spare sectors for M user cylinders);
   - a zone/group of zones/all zones[6];
2. Spare tracks(a track) can be placed in
   - a cylinder/group of cylinders;
   - a zone/group of zones/all zones;
3. Spare cylinders(a cylinder) can be placed in
   - a group of cylinders (for example N spare cylinders every M user cylinders);
   - a zone/group of zones/all zones;

Different manufacturers can use different areas depending on how they consider it to be more profitable regarding the capacity usage, performance and safety. The more the spares are allocated, the more reliability. On the contrary it strikes on the end user because of the redundant disk space usage and price. By remapping sectors it can be divided into:

1. Replacing a defective sector with a spare one by reassigning it[7].

[6]In fact, all zones means the whole disk space. It means each zone contains N spare sectors, where N can vary.

[7]Replacing one defective sector by one spare sector. Normally this is not the case.

2. Replacing a defective sector or a group of sectors by reassigning the entire track containing this sector/these sectors.

By remapping methods in can be divided into:

1. Reallocation of the defective area to a spare one.

2. Slipping of the defective area with shifting all the sectors/tracks.

Chapter 4

Disk Performance

In this work an attempt is made to unveil technology of remapping which is often out of consideration in papers dedicated to hard disk drives. This thesis also tries to look at the subject from the perspective of disk performance – one of the key points of modern computers. Chapter 4 considers general disk I/O performance issues as well as disk parameters which affect general disk performance. Section 4.1 explains basic conception of mass sotrage performance and lists most of its main features in order to give an overview how complicated it is. Section 4.2 introduces the key definitions of the topic. Section 4.3 illustrates how remapping technology is related to the disk performance.

4.1 Mass storage I/O performance

I/O performance is defined as a number of tasks completed per time unit, in other words it is *throughput*. One of the general purposes of computer systems is to store and retrieve data. Since accessing storage devices is slower than internal memory, the storage system is a bottleneck in many large applications [HP03]. If one thinks of a disk system and related components as a whole then many options are available to reduce the I/O bottleneck.

Let us briefly review general features which affect the I/O performance. These are divided into two categories – low-level hardware items and high-level software ones.

The most relevant hardware features affecting the I/O performance are the following:

Data transfer bus type [Dav05] : today several data transfer buses exist with the even greater number of standards. This is one of the crucial points about working

with data. The most common types are listed below.

- SCSI;
- ATA;
- Serial ATA;
- Fiber Channel;
- USB.

Block size to transfer in a burst: block length matters since extra resources are needed in order to transfer each data block. It also might significantly affect the total I/O performance and varies from a few bytes to gigabytes (e.g. 2,4,8 bytes; 131072 bytes; 4 GB).

Data cable length: the longer the data cable, the more efficiently the storage cases are used. However, the longer it is, the more electricity noise sensitive it is, and in turn more slow. It varies from tens of centimeters to thousands of kilometers (e.g. 1-3m; 100m; 10km).

Data storage type: the media type affects the access time to data. Which might be the most crucial point in I/O performance bottleneck.

- Optical media;
- Magnetic media;

Scheduling firmware algorithms provided by I/O queues [BG03, SMW98, TL02]: the firmware microprograms implement a variety of scheduling algorithms. The microcode resides inside the disk systems [1] and is responsible for the commands and data bursts order.

- FCFS (First Come First Service, all the requests are processed in the order they are received);
- SPTF (Shortest Positioning Time First, the requests are chosen in such an order that optimize positioning time, for both seek and rotational latency);

[1] Contemporary data managing adapters also contain similar microprograms.

- SCAN (For the SCAN policy all the requests in a queue are sorted by access order to the cylinders);
- SSTF (Shortest Seek Time First, next request to be chosen is the one which will give shortest seek time).

Disk cache: middle layer memory between the host and disk mechanism which serves as a buffer and stores frequently accessed data or data which is supposed to be accessed in the near future. This memory greatly improves disk performance since memory access occurs significantly faster than access to the disk surface with the following data transferring [ZH03, Shr97, SMW98]. There are various cache features that affect its efficiency:

- Cache size;
- Cache line size;
- Cache segmentation;
- Replacement algorithms (LRU, FIFO, LIFO, etc);
- Cache usage policies (Read-ahead strategies, write-to-cache strategies, etc).

Data accessed location: since angular velocity differs depending on the disk area – beginning of a drive, or its end – the final performance differs as well, depending on the accessing of the data located on a platters surface [ST].

Many of the low-level performance features are implemented also on the higher levels, for instance file system caching, or queuing requests to send via I/O system to the hardware level, etc. Software features affecting the I/O performance are the following:

Operation System Target type: there exist many types of operational systems, they can be represented as the middle layer between the user and user's data. From this point of view, they are constructed and optimized for a variety of user tasks – the range is difficult to enumerate – from everyday home routine to calculating astronomical distances and units.

- Network Appliance;
- Server;

- Workstation;
- Embedded;
- Home Desktop.

File system type: a file system might affect the performance dramatically, therefore there is large variety of them serving different purposes. We mark such parameters as minimal block size to operate, file system structure and attributes, self-monitoring features, scheduling policies and etc. which affect the performance.

- Disk file systems (e.g. NTFS, ext2-3, HFS, FAT);
- Network file systems (e.g. NFS, AFP, NCP);
- Database file systems (e.g. WinFS, BFS, GnomeVFS);
- Transactional file systems (Vista, PerDiS, ETFS);
- Special purpose file systems (e.g. 'swap' partitions or files, '/proc' file system in UNIX OS).

Data type to transfer/access: data can be structured in various ways with various block sizes which are transferred or accessed. The difference between data transfer speed versus request size can be seen in [ST].

4.2 Disk performance parameters

This section briefly defines disk performance parameters for the mechanical part of a hard drive which are usually involved in the common disk performance calculation.

Seek time: *Seek time* is the time required to move the disk's arm to the proper track. Seek time is a composition of the following [Sta03, HP03, Shr97]:

- the initial startup time or *speedup*, this operation lasts until the arm reaches either its maximum speed or half of the seek distance;
- *coast* for long seeks, time needed to traverse over the tracks to the required one at maximum velocity;
- *slowdown*, during this time the arm moves close to the desired track;

- *settle*, the disk controller adjusts the head to access the desired location.

Full stroke is the time it takes to pass over all the cylinders of a hard drive.

Rotational delay: *Rotational delay* or rotational latency is the time needed for the requested sector to be positioned under the head after it has reached the desired track.

Access time: *Access time* is a sum of the seek time and rotational latency. This is the time it takes to locate the arm to the position of write or read.

Transfer time: *Transfer time* is the time it takes to transfer data from or to a hard disk drive after placing the head in the read/write position.

Track switch time: One arm consists of several heads. Usually these heads share one data channel. After the controller has switched the channel from one head to another, the new head may need repositioning because of the misalignment of the tracks on the different platter surfaces. The time needed to do so is called *track switch time*.

Cylinder switch time: *Cylinder switch time* is time needed to do a seek of one cylinder.

4.3 Remapping technology and performance parameters

Technology of remapping was already introduced in Chapter 3. Subsection 3.3.3 gave a more detailed description of how and where spare sectors are allocated. Let us have a sequence of the defective sectors which have been reallocated to some other place on the platter or even on a different platter. Requesting the sectors which have been reallocated, there is an extra time needed to access their new location. The disk arm has to be moved from the continuous reading of a sector stream to the new location in order to access the disk space with the reallocated sectors. After all, this arm probably needs to move back to read the rest of the sectors in the sequence. If there is only one defective sector on a disk then the user unlikely notices any performance loss, but what would be the performance behavior if there is a set of these ones? In particular, how does it affect the seek time? In

a simple fashion [Sta03] *total average access time* can be expressed as:

$$T_a = T_s + \frac{1}{2r} + \frac{b}{rN}$$

where

- T_s – average seek time;
- b – number of bytes to be transfered;
- N – number of bytes on a track;
- r – rotation speed, revolutions per second.

One of the restrictions of this formula is modern multizone disks, where the number of bytes per track varies. This formula is sufficient as a rough approximation, but for disk system modeling it is unacceptable. PhD thesis of Elizabeth Shriver [Shr97], "Performance modeling for realistic storage devices", gives even more complicated and exhaustive formulas for disk mechanical parts, which take into consideration many real hard disk parameters. None of the works dedicated to disk models take the remapping mechanism parameters into account. Chapter 5 suggests an approach to model the seek time for a disk with defective sectors.

4.4 Summary

Disk drives have a variety of parameters, performance ones play a significant role amid them. In order to understand the performance of a disk device one must view which components affect this. In this chapter the general disk I/O performance issues were discussed. Section 4.1 explained the conception of disk performance, Section 4.2 represented the key definitions and relation of the performance to the remapping technology was illustrated in Section 4.3.

Chapter 5

An analytic model for a generic remapping scheme

This chapter introduces an analytic model of a disk mechanism with the implementation of generic remapping scheme for defective sectors substitution. The model is intended to calculate mean sector seek time, the main goal for that is to determine how remapping mechanism affects the seek time. Generic remapping scheme implies a scheme where disk cylinder space is divided into two parts, one part contains cylinders for storing user data and referred to as user space. The second part contains cylinders with spare sectors also referred to as spare space or spare cylinders. The model is based on Shriver's model created for storage devices and is described in her PhD work [Shr97] (additional information and sources regarding the model can be found here [Shra, Shrb]).

5.1 Types of models

There are five more or less generalized approaches to modeling storage systems. These will be briefly discussed in this section [Shr97].

5.1.1 Models with simple measures

This method mostly serves as a practical one for developing tools such as capacity planning, performance measurement, data reduction programs, performance prediction models, etc. As input data these models accept a workload and as an output, give a variety

of performance measures such as queue lengths and throughput [WAA+04, TCG02]. The measures are quite simple and do not support, for example, the determination of service time. This information is then used for different system configurations.

5.1.2 Simulation models

A *simulation model* is a software system written to represent the dynamic behavior of a system by reproducing its states and corresponding state transitions [LC87]. As an input this model receives either synthetic or traced workloads. The model's accuracy depends on level of details. These models are mostly used to analyze parts of the storage systems and for various system evaluations [Wil95, BG03].

One of the widely used methods is *trace-driven analysis*. It allows evaluation of part of a computer system by using event-driven simulations generated by real traces. The following steps are taken in order to build the model [Smi85]:

- tracing and recording the sequence of events and relevant parameters for a computer system;
- synthesis the traced data into an event-driven simulation;
- running the simulation for a variation of the computer system.

The result is an accurate prediction of the behavior of the variant system.

There is also a different method of *hybrid modeling* which is used in developing complex system simulation models. This method mainly uses analytic modeling for nodes of a complex model, and only if an analytic model cannot be built or it does not give the accuracy needed then simulation is applied. The more detailed the model, the more time is needed for the result, therefore complex simulations can be extremely time consuming.

5.1.3 Individual component models

This method is used to study individual physical components. Using analytic models of queues, caches or mechanical parts of the storage system, research can be done on a variety of algorithms such as scheduling algorithms, caching policies, reading ahead, etc. In this case the remainder of the storage system is either not modeled or simple assumptions are used. This model cannot be used for predicting the performance of the system as a whole.

This approach is used in order to study the impact of remapping on the seek time for a hard disk drive.

5.1.4 Composition models

This is an approach of composing component models into a single one. There are several known methods of composition [KB88] and decomposition [AL93] of the storage system into parts. For such models *queuing networks* and *layered queuing models* can also be applied.

5.1.5 Table-based models

This is a relatively simple approach for building a model based on generating the input points in a table which maps them to an output. It is used when there is no need to build a complex and huge model, where deep understanding of the background is necessary. It cannot be used for accurate calculation since it would require creating long tables even though interpolation can be used in some cases to reduce workload inputs and in these cases it gives an accurate result, which is enough for some tasks [And01].

5.2 The model

The sections above gave a brief overview of the approaches to disk storage modeling. The relevant information about disk models and also workload descriptions, validation approaches, as well as a very extensive description of Shriver's model itself can be found here [Shr97, Shra, Shrb]. Next sections of this paper consider the model developed with respective definitions and formulas as it follows. Subsection 5.2.1 describes and considers conditions of the model. Subsection 5.2.2 shows how to get physical parameters needed for a disk to be modeled. Subsections 5.2.3 and 5.2.4 contain explanation of the model. Thereafter, the results of model validation are discussed in Section 5.3.

5.2.1 Mean disk mechanism service time and model conditions

The value of mechanism service time (*MechanismServiceTime*) can be represented as a sum of two parts: time needed to position the disk head on the cylinder and sector

requested (PT) and the time taken to transfer data to the higher levels of the HDD ($TT(x)$ for transferring x bytes):

$$MechanismServiceTime = PT + TT[RequestSize].$$

Where PT is the sum of the seek time (ST) and rotational latency (RL)

$$PT = ST + RL.$$

The mean disk positioning time is then:

$$E(PT) = E(ST) + E(RL).$$

$E(ST)$ and $E(RL)$ are workload dependent. In order to make the difference significant the sequential and random requests are used. Workload with sequential ones accesses the sectors (in the case of rotational latency) and cylinders (in the case of seek time) one by one, hence there is not much disk head movement over the disk surface (in comparison with random requests). On the other hand, if the location of the data requested is arbitrarily distributed then the disk head constantly moves over the surface.

On the analogy of Shriver's dissertation, normalization of seek time is done in order to be independent of the number of cylinders by defining *seek fraction* (SF): this is a random variable representing the fraction of the total number of cylinders that need to be sought over. First, everything is considered from the perspective of SF, and after that the results are represented in terms of time.

The following remapping scheme is used: spare cylinders are at the end of the disk area, if a corruption of the disk surface occurs then sectors are allocated to the area of the spare cylinders. In this scheme, if a defective sector is detected, the disk head does not move to this sector but it starts to move to the reallocated one.

Since this thesis is after creating a model which would take into account the remapping mechanism but not an entire disk system, it does not concentrate on such things as disk zones effect [Met97], track and cylinder skews [SSP+05], transferring time, etc. as more profound research is required on combining those parts into a singe model.

In this work the rotational latency (RL) is also not taken into account since the analytic model is for a generic remapping scheme which reallocates sectors to different

cylinders in contrast to schemes (see Chapter 3) that reallocate sectors to a different part of the same track, in which case rotational latency would be remapped sectors dependent, but otherwise it is constant (see [Shr97]) which is no of interest.

5.2.2 Seeking

The disk head does seeking over the cylinders. There are several types of seeks while the mechanical part processes the requests. Very short seeks (2 to 4 cylinders) are dominated by the settle time (disk controller adjusts the head to access the desired location). Short seeks are dominated by the speedup (arm is accelerated until it reaches half of the seek distance or a fixed maximum velocity). Long seeks are dominated by the coast (the arm is moving at the maximum velocity). Thus, the seek function can be represented as [Shr97, TCG02]

$$SeekTime(dis) = \begin{cases} 0 & dis = 0 \\ a + b\sqrt{dis} & 0 < dis \leq e \\ c + d \cdot dis & dis > e \end{cases} \tag{5.1}$$

where a, b, c, d, e are device-specific parameters and dis is the number of cylinders to be traveled. The number of cylinders e is a point, where the disk head reaches its maximum velocity. Before this point, the head moves with velocity of $\left(\frac{2}{b^2}\right)(t-a)$ cylinders per second and acceleration of $\left(\frac{2}{b^2}\right)$ cylinders per squared second, t is the time. After the disk head has reached its maximum velocity, it starts moving with constant velocity of $\left(\frac{1}{d}\right)$ cylinders per second.

For the modeling, the parameters for an IBM DDYS-T36950 SCSI hard drive are used, these parameters are equal to $a = 0.9862, b = 0.06894, c = 4.082, d = 0.0003694, e = 5660$. In order to get these parameters, the time needed for the disk head to cover distance of $0-1$ cylinders, $0-2$, $0-3$, ..., $0-LastCylinder$ cylinders was measured (see Figure 5.1). In Figure 5.1, the time versus the number of cylinders passed is depicted as a solid line. Parameters for the approximation with a power function are the parameters for the tested disk. In Figure 5.1, the approximation function is depicted as a dashed line. Knowing these parameters, mean seek time for a single request can be calculated.

Let $P(SF \geq x)$ represent the probability that the seek fraction is greater than or

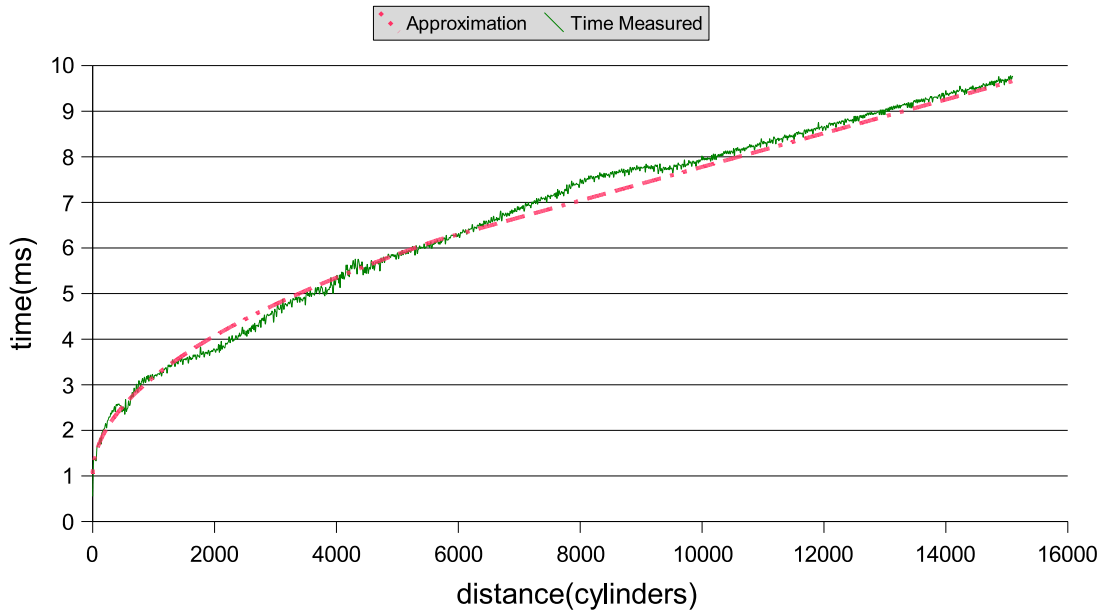


Figure 5.1: Graph displaying the measured-seek-time versus distance curve for IBM DDYS-T36950 SCSI hard drive.

equal to x, then it can be computed as:

$$E(SF) = \int_0^1 P(SF \geq x)dx. \tag{5.2}$$

Since the aforementioned seek curve is not linear, Formula 5.2 does not allow to compute mean seek time. Thus, the mean seek time can be calculated according to:

$$E(ST) = \int_0^\infty P(ST \geq x)dx. \tag{5.3}$$

The disk is assumed to be full of data. The seek fraction and seek time are analyzed as functions of the spatial locality of the workload – random uniform access across the cylinders and sequential access with uniformly distributed runs across the cylinders. There is also an assumption that remapped sectors are distributed uniformly over the disk area.

5.2.3 Random uniform access

Let m be the number of cylinders in user space, n is the number of spare cylinders in use and therefore a variable, $MaxCylinder$ can be defined as $MaxCylinder = m + n$. Let us consider seek fraction for the case. Request distribution can be approximated as a uniform cylinder address distribution. Thus any pair of the requests can be treated as a random sample of size two cylinders chosen from a population with uniform probability distribution between 0 and $MaxCylinder$. The starting point for any request is the ending cylinder for the previous request.

$P(SF \geq x)$ is directly related to $P(SD \geq z)$, where SD is an integer-valued seek distance, approximated as a continuous random variable and thus

$$SF = \frac{SD}{MaxCylinder}.$$

Let l_1 represent the cylinder number of the previous request (starting head movement point) and l_2 represent the cylinder number of the current request (ending head movement point).

$$SD = |l_1 - l_2|$$

Both variants $l_1 < l_2$ and $l_1 > l_2$ are equally likely, therefore the $P(SD \geq z)$ is computed taking into account only the first case and the result has to be multiplied by 2. The probability for any cylinder to be chosen is assumed to be equiprobable meaning that it is $1/MaxCylinder$, hence the probability for a particular pair is known to be $1/MaxCylinder^2$. Thus $P(SD \geq z)$ without consideration of the remapping mechanism is equal to:

$$P(SD \geq z) = \frac{2}{MaxCylinder^2} \int_0^{MaxCylinder-z} \int_{l_1+z}^{MaxCylinder} 1 dl_2 dl_1 =$$

$$= \left(\frac{MaxCylinder - z}{MaxCylinder}\right)^2 = \left(1 - \frac{z}{MaxCylinder}\right)^2. \quad (5.4)$$

If this is plugged into (5.2) and normalized $MaxCylinder$ to 1 then the result is:

$$E(SF_{ran}) = \int_0^1 (1-x)^2 dx = \frac{1}{3}$$

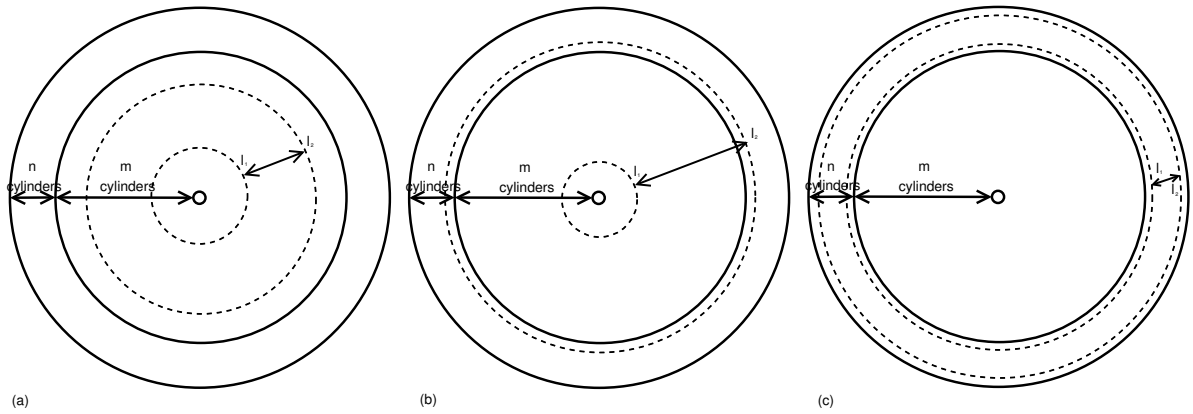


Figure 5.2: Allocations of the cylinders accessed on disk platters with random requests. (a) Both cylinders requested are in the main area. (b) One of the requested cylinders is in the spare area and the other cylinder is in the main area. (c) Both cylinders requested are in the spare area.

In this paper there are two different zones, which should be taken into account – the first one is the user area, the second one is the area with spare sectors, them both need to be considered. Therefore, $MaxCylinder$ for this case is the sum of the total number of the non-spare cylinders in use and spare ones which have already been taken from the reserved space. There can be four different cases for the model dependent on where requests are located (see Figure 5.2), according to this:

$$P_{ran}(SD \geq z) = P_1(SD \geq z) + P_2(SD \geq z) + 2P_3(SD \geq z). \tag{5.5}$$

$P_1(SD \geq z)$ is the probability that $SD \geq z$ when both cylinders requested are in the main area (see Figure 5.2.a). $P_2(SD \geq z)$ is the probability that $SD \geq z$ when both cylinders requested are in the spare area (see Figure 5.2.c). $P_3(SD \geq z)$ is the probability that $SD \geq z$ when one of the requests accesses the spare cylinder and the other accesses the cylinder from the main area (see Figure 5.2.b). The third case has to be considered more carefully since the number of cylinders from the main area is greater than the number of cylinders from the spare area and therefore the seek distance itself might be greater than or less than the number of spare cylinders. If $z \geq n$ then physically it means that the request from cylinder m might be greater than the number of $MaxCylinder$. By splitting $P_3(SD \geq z)$ into two parts, the seeks done with the starting cylinder x, $x \in [m...m+n-1]$,

to the seek distance of more than n cylinders are subtracted from the set of seeks which are probable in these circumstances.

$$P_3(SD \geq z) = \begin{cases} P_{31}(SD \geq z) & z < n \\ P_{32}(SD \geq z) & z \geq n \end{cases} \tag{5.6}$$

Since remapped sectors and requests are distributed uniformly over the disk area then the probability to hit a remapped sector is:

$$P_r = \frac{NumberOfRemappedSectorsInUse}{TotalNumberOfUserSectors}$$

As previously defined, m is the number of the total amount of user cylinders in a hard drive, n is the number of used spare cylinders in a hard drive. Formulas derived for the P_1, P_2, P_{31} and P_{32} are:

$$P_1(SD \geq z) = \frac{2(1-P_r)^2}{((1-P_r)m + P_r n)^2} \int_0^{m-z} \int_{l_1+z}^{m} 1 dl_2 dl_1 = \frac{(1-P_r)^2(m-z)^2}{((1-P_r)m + P_r n)^2}.$$

$$P_2(SD \geq z) = \frac{2P_r^2}{((1-P_r)m + P_r n)^2} \int_m^{m+n-z} \int_{l_1+z}^{m+n} 1 dl_2 dl_1 = \frac{P_r^2(n-z)^2}{((1-P_r)m + P_r n)^2}.$$

$$P_{31}(SD \geq z) = \frac{2P_r(1-P_r)}{((1-P_r)m + P_r n)^2} \left[\int_0^m \int_0^n 1 dl_2 dl_1 - \int_0^z l_3 dl_3 \right] = \frac{2P_r(1-P_r)(mn - \frac{1}{2}z^2)}{((1-P_r)m + P_r n)^2}.$$

$$P_{32}(SD \geq z) = \frac{2P_r(1-P_r)}{((1-P_r)m + P_r n)^2} \left[\int_0^m \int_0^n 1 dl_2 dl_1 - \int_0^z l_3 dl_3 + \int_0^{z-n} l_4 dl_4 \right] =$$
$$= \frac{2P_r(1-P_r)(mn - \frac{1}{2}n^2 - n(z-n))}{((1-P_r)m + P_r n)^2}.$$

Using a, b, c, d and e from (5.1) for the seek curve and taking into account that the

probability of seek time being greater than $SeekTime(MaxCylinder)$ is 0, applying (5.5):

$$E(ST_{ran}) = \int_0^{\infty} P(ST \geq x)dx = \int_0^{a} P(ST \geq x)dx + \int_a^{SeekTime(MaxCylinder)} P(ST \geq x)dx =$$

$$a + \int_a^{SeekTime(MaxCylinder)} P(SD \geq Cyl(x))dx =$$

$$= a + \int_a^{SeekTime(n)} [P_1(SD \geq Cyl(x)) + P_2(SD \geq Cyl(x)) + P_{31}(SD \geq Cyl(x))dx] +$$

$$+ \int_{SeekTime(n)}^{SeekTime(e)} [P_1(SD \geq Cyl(x)) + P_{32}(SD \geq Cyl(x))dx] +$$

$$+ \int_{SeekTime(e)}^{SeekTime(m)} [P_1(SD \geq Cyl(x)) + P_{32}(SD \geq Cyl(x))dx] .$$

Where

$$Cyl(x) = \begin{cases} \left(\frac{x-a}{b}\right)^2 & a < x < SeekTime(e) \\ \left(\frac{x-c}{d}\right) & SeekTime(e) \leq x < SeekTime(MaxCyl) \end{cases}$$

$$E(ST_{ran}) = a + \int_{a}^{SeekTime(n)} \left[\frac{(1-P_r)^2(m-(\frac{x-a}{b})^2)^2}{((1-P_r)m+P_rn)^2} + \frac{P_r^2(n-(\frac{x-a}{b})^2)^2}{((1-P_r)m+P_rn)^2} + \right.$$

$$\left. + \frac{4P_r(1-P_r)(mn-\frac{1}{2}(\frac{x-a}{b})^4)}{((1-P_r)m+P_rn)^2} \right] dx + \int_{SeekTime(n)}^{SeekTime(e)} \left[\frac{(1-P_r)^2(m-(\frac{x-a}{b})^2)^2}{((1-P_r)m+P_rn)^2} + \right.$$

$$\left. + \frac{4P_r(1-P_r)(mn-\frac{1}{2}n^2-n((\frac{x-a}{b})^2-n))}{((1-P_r)m+P_rn)^2} \right] dx + \int_{SeekTime(e)}^{SeekTime(m)} \left[\frac{(1-P_r)^2(m-(\frac{x-c}{d}))^2}{((1-P_r)m+P_rn)^2} + \right.$$

$$\left. + \frac{4P_r(1-P_r)(mn-\frac{1}{2}n^2-n((\frac{x-c}{d})-n))}{((1-P_r)m+P_rn)^2} \right] dx.$$

Letting $t = SeekTime(n)$, $g = SeekTime(e)$, $k = SeekTime(m)$, then:

$$
\begin{aligned}
E(ST_{ran}) = a + 1/5\,\frac{(1-p)^2\,(t^5-a^5)}{b^4\,((1-p)\,m+pn)^2} - \frac{(1-p)^2\,a\,(t^4-a^4)}{b^4\,((1-p)\,m+pn)^2} + \\
+ 1/3\;(1-p)^2\left(-2\left(m-\frac{a^2}{b^2}\right)b^{-2}+4\,\frac{a^2}{b^4}\right)\left(t^3-a^3\right)((1-p)\,m+pn)^{-2} + \\
+ 2\;(1-p)^2\left(m-\frac{a^2}{b^2}\right)a\left(t^2-a^2\right)b^{-2}\,((1-p)\,m+pn)^{-2} + \\
+ (1-p)^2\left(m-\frac{a^2}{b^2}\right)^2(t-a)\,((1-p)\,m+pn)^{-2} + 1/5\,\frac{p^2\,(t^5-a^5)}{b^4\,((1-p)\,m+pn)^2} - \\
- \frac{p^2a\,(t^4-a^4)}{b^4\,((1-p)\,m+pn)^2} + 1/3\,p^2\left(-2\left(n-\frac{a^2}{b^2}\right)b^{-2}4\,\frac{a^2}{b^4}\right)\left(t^3-a^3\right)((1-p)\,m+pn)^{-2} + \\
+ 2\,p^2\left(n-\frac{a^2}{b^2}\right)a\left(t^2-a^2\right)b^{-2}\,((1-p)\,m+pn)^{-2} + \\
+ p^2\left(n-\frac{a^2}{b^2}\right)^2(t-a)\,((1-p)\,m+pn)^{-2} - 2/5\,\frac{(1-p)\,p\,(t^5-a^5)}{b^4\,((1-p)\,m+pn)^2} + \\
+ 2\,\frac{(1-p)\,pa\,(t^4-a^4)}{b^4\,((1-p)\,m+pn)^2} - 4\,\frac{(1-p)\,pa^2\,(t^3-a^3)}{b^4\,((1-p)\,m+pn)^2} + 4\,\frac{(1-p)\,pa^3\,(t^2-a^2)}{b^4\,((1-p)\,m+pn)^2} + \\
+ 4\;(1-p)\,p\left(mn-1/2\,\frac{a^4}{b^4}\right)(t-a)\,((1-p)\,m+pn)^{-2} + 1/5\,\frac{(1-p)^2\,(g^5-t^5)}{b^4\,((1-p)\,m+pn)^2} + \\
+ 1/3\;(1-p)^2\left(-2\left(m-\frac{a^2}{b^2}\right)b^{-2}+4\,\frac{a^2}{b^4}\right)\left(g^3-t^3\right)((1-p)\,m+pn)^{-2} - \\
- \frac{(1-p)^2\,a\,(g^4-t^4)}{b^4\,((1-p)\,m+pn)^2} + 2\;(1-p)^2\left(m-\frac{a^2}{b^2}\right)a\left(g^2-t^2\right)b^{-2}\,((1-p)\,m+pn)^{-2} + \\
+ (1-p)^2\left(m-\frac{a^2}{b^2}\right)^2(g-t)\,((1-p)\,m+pn)^{-2} - 4/3\,\frac{(1-p)\,pn\,(g^3-t^3)}{b^2\,((1-p)\,m+pn)^2} + \\
+ 4\;(1-p)\,p\left(mn-1/2\,n^2-n\left(\frac{a^2}{b^2}-n\right)\right)(g-t)\,((1-p)\,m+pn)^{-2} + \\
+ 4\,\frac{(1-p)\,pna\,(g^2-t^2)}{b^2\,((1-p)\,m+pn)^2} - (1-p)^2\left(m+\frac{c}{d}\right)\left(k^2-g^2\right)d^{-1}\,((1-p)\,m+pn)^{-2} + \\
+ 1/3\,\frac{(1-p)^2\,(k^3-g^3)}{d^2\,((1-p)\,m+pn)^2} + (1-p)^2\left(m+\frac{c}{d}\right)^2(k-g)\,((1-p)\,m+pn)^{-2} - \\
- 2\,\frac{(1-p)\,pn\,(k^2-g^2)}{d\,((1-p)\,m+pn)^2} + \frac{4\;(1-p)\,p\left(mn-1/2\,n^2-n\left(-\frac{c}{d}-n\right)\right)}{(k-g)\,((1-p)\,m+pn)^2}. \qquad (5.7)
\end{aligned}
$$

Maple 6 the mathematical tool [Map06] was used in order to derive this formula. The result does not look highly optimized and probably might be slightly simplified for a final application but for the current research it is enough. The formula derived by Shriver for random access to the disk is given below.

$$E(ST_{ran}) = g - \frac{a^5}{5b^4f^2} + \frac{a^4g}{b^4f^2} - \frac{2a^3g^2}{b^4f^2} + \frac{2a^2g^3}{b^4f^2} - \frac{g^3}{3d^2f^2} - \frac{ag^4}{b^4f^2} + \frac{g^5}{5b^4f^2} + \frac{2a^3}{3b^2f} - \\ - \frac{2a^2g}{b^2f} + \frac{2ag^2}{b^2f} - \frac{2g^3}{3b^2f} - \frac{g^2(-c-df)}{d^2f^2} - \frac{g(c+df)^2}{d^2f^2} + \frac{(c+df)^2m}{d^2f^2} + \frac{(-c-df)m^2}{d^2f^2} + \frac{m^3}{3d^2f^2}. \quad (5.8)$$

If the parameters of spare sectors are set to 0 in Formula (5.7), exactly the same numerical result is received for the disk being modeled as Shriver's one calculates.

Now we have a closed form which can be used for the approximation of the seek time in case of random uniform access with the restrictions discussed earlier. Normally, disks perform more complicated tasks in comparison with the ones considered in this subsection. In the following subsection more realistic conditions for the disks will be discussed.

5.2.4 Sequential access

This part considers sequential access to the media. In case of random uniform access all the requests are single sectors[1] and are uniformly distributed across the cylinder space. Now a request consists of a number of runs. The accesses to the runs are distributed over the cylinders in the same way as accesses to the single sectors described in the previous subsection. A run in turn consists of a number of *run_length* sectors going one by one. The number of runs in one request is *run_num*. Defective sectors are uniformly distributed over the disk. For this case seek time is represented as:

$$E(ST_{seq}) = \frac{1}{run\_length}E(ST_{ran}) + (1 - \frac{1}{run\_length})(1 - P_{run})E(ST_{jump}). \quad (5.9)$$

Let us consider it more closely, the formula consists of two parts. The first part deals with jumps between runs in a request (see Figure 5.3, a, b, c) and is described in the first item.

[1] In fact, the seek time is calculated for cylinders only since the rotational latency is not considered (see Subsection 5.2.1), but because of the random uniform access and constant rotational latency, in this subsection it is easier to think of them as of single sectors.

The second part deals with jumps to the spare area within processing a run of sectors (see Figure 5.3.d). These are discussed further.

As previously mentioned, the runs are accessed in the same way as single sectors in the previous subsection, therefore, $E(ST_{ran})$ is used in Formula 5.9. The total number of seek operations in a request is $run\_num * run\_length$, the number of seek operations between the runs is run_num, then the probability that a jump between two sequential runs is being performed is equal to

$$\frac{run\_num}{run\_length * run\_num} = \frac{1}{run\_length}.$$

The second part deals with processing a run of the sectors and is described by the second item. The probability of being in a run is equal to

$$1 - \frac{1}{run\_length}.$$

The mean time spent for a disk head to move into the spare area accessing a run (see Figure 5.3.d) is equal to $(1 - P_{run})E(ST_{jump})$. Let us calculate the probability P_{run} that there is no defective sector (i.e. jumping to the spare cylinders area is not being performed) in a run being accessed:

$$P_{run} = \frac{(N_t - N_r)(N_t - N_r - 1)\cdots(N_t - N_r - run\_length)}{N_t(N_t - 1)\cdots(N_t - run\_length)} = \frac{(N_t - run\_length - 1)!(N_t - N_r)!}{N_t!(N_t - N_r - run\_length - 1)!}. \quad (5.10)$$

Where N_t is the total number of sectors in a disk, N_r is the number of remapped sectors. Consequentially, the probability that there is at least one defective sector is $(1 - P_{run})$.

$E(ST_{jump})$ is mean time spent jumping only into the spare area processing a run of sectors and is computed by analogy with $E(ST_{ran})$ in Subsection 5.2.3. The probability that $SD \geq z$ when the disk head moves to the area of spare cylinders accessing a run of sectors is $P_{jump}(SD \geq z)$. One of the cylinders accessed is in the main area and the other one is in the spare are, and so, the seek distance, again, can exceed the total number of

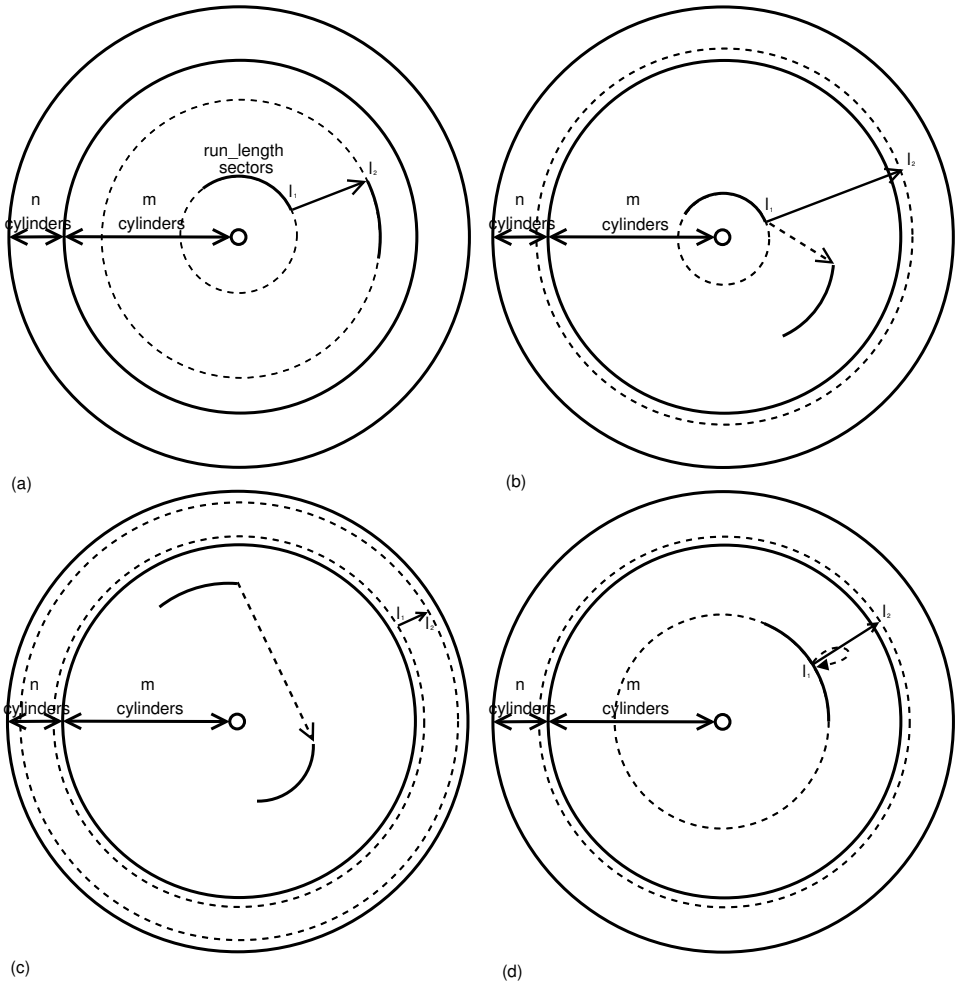


Figure 5.3: Allocations of the cylinders accessed on disk platters with sequential requests. Both runs requested are in the main area. (a) l_1 is the last sector accessed in the previous run, l_2 is the first sector accessed in the current run. Cylinders with the sectors are in the main area. (b) l_1 is the last sector accessed in the previous run, l_2 is the first sector accessed in the current run. One of the cylinders with the sector is in the spare area and the other cylinder is in the main area. (c) l_1 is the last sector accessed in the previous run, l_2 is the first sector accessed in the current run. Both cylinders with the sectors are in the spare area. (d). l_1 and l_2 are two adjacent sectors in the current run. One of the cylinders with the sector is in the spare area and the other cylinder is in the main area.

cylinders, therefore, two cases have to be considered (see Formula 5.6):

$$P_{jump}(SD \geq z) = \begin{cases} P_{jump1}(SD \geq z) & z < n \\ P_{jump2}(SD \geq z) & z \geq n \end{cases}$$

Thus

$$P_{jump1}(SD \geq z) = \frac{1}{((1-P_r)m + P_r n)^2}\left[\int_0^m \int_0^n 1 dl_2 dl_1 + \int_0^z l_3 dl_3\right] = \\ = \frac{nm + \frac{1}{2}z^2}{((1-P_r)m + P_r n)^2} \quad (5.11)$$

and

$$P_{jump2}(SD \geq z) = \frac{1}{((1-P_r)m + P_r n)^2}\left[\int_0^m \int_0^n 1 dl_2 dl_1 - \right. \\ \left. - \int_0^z l_3 dl_3 + \int_0^{z-n} l_4 dl_4\right] = \frac{nm - \frac{1}{2}z^2 + \frac{1}{2}(z-n)^2}{((1-P_r)m + P_r n)^2}. \quad (5.12)$$

Using a, b, c, d and e from (5.1) for the seek curve and taking into account that the probability of seek time being greater than $SeekTime(MaxCylinder)$ is 0, the final closed formula can be obtained for $E(ST_{jump})$.

$$E(ST_{jump}) = \int_0^{\infty} P_{jump}(ST \geq x)dx = \int_0^{a} P_{jump}(ST \geq x)dx +$$
$$+ \int_a^{SeekTime(MaxCyl)} P_{jump}(ST \geq x)dx = a + \int_a^{SeekTime(n)} P_{jump1}(SD \geq Cyl(x))dx +$$
$$+ \int_{SeekTime(n)}^{SeekTime(e)} P_{jump2}(SD \geq Cyl(x))dx + \int_{SeekTime(e)}^{SeekTime(m)} P_{jump2}(SD \geq Cyl(x))dx =$$
$$= a + \int_a^{SeekTime(n)} P_{jump1}(SD \geq Cyl(x))dx + \int_{SeekTime(n)}^{SeekTime(e)} P_{jump2}(SD \geq Cyl(x))dx +$$
$$+ \int_{SeekTime(e)}^{SeekTime(m)} P_{jump2}(SD \geq Cyl(x))dx. \tag{5.13}$$

Where

$$Cyl(x) = \begin{cases} \left(\frac{x-a}{b}\right)^2 & a < x < SeekTime(e) \\ \left(\frac{x-c}{d}\right) & SeekTime(e) \leq x < SeekTime(MaxCyl) \end{cases}$$

Plugging appropriate values into (5.13), $E(ST_{jump})$ now can be computed as follows::

$$E(ST_{jump}) = a + \int_a^t \frac{nm + \frac{\frac{1}{2}(x-a)^4}{b^4}}{((1-P_r)m + P_r n)^2}dx + \int_t^g \frac{nm - \frac{1}{2}\frac{(x-a)^4}{b^4} + \frac{1}{2}(\frac{(x-a)^2}{b^2} - n)^2}{((1-P_r)m + P_r n)^2}dx +$$
$$+ \int_g^k \frac{nm - \frac{1}{2}\frac{(x-c)^2}{d^2} + \frac{1}{2}(\frac{(x-c)}{d} - n)^2}{((1-P_r)m + P_r n)^2}dx = a + \frac{1}{10}\frac{t^5 - a^5}{b^4\left((1-p)\,m + pn\right)^2} -$$
$$- \frac{1}{2}\frac{a\left(t^4 - a^4\right)}{b^4\left((1-p)\,m + pn\right)^2} + \frac{a^2\left(t^3 - a^3\right)}{b^4\left((1-p)\,m + pn\right)^2} - \frac{a^3\left(t^2 - a^2\right)}{b^4\left((1-p)\,m + pn\right)^2} +$$
$$+ \frac{\left(nm + \frac{1}{2}\frac{a^4}{b^4}\right)(t-a)}{\left((1-p)\,m + pn\right)^2} + \frac{1}{3}\frac{\left(\left(\frac{a^2}{b^2} - n\right)b^{-2} - \frac{a^2}{b^4}\right)\left(g^3 - t^3\right)}{\left((1-p)\,m + pn\right)^2} +$$
$$+ \frac{1}{2}\frac{\left(-2\left(\frac{a^2}{b^2} - n\right)ab^{-2} + 2\frac{a^3}{b^4}\right)\left(g^2 - t^2\right)}{\left((1-p)\,m + pn\right)^2} + \frac{\left(nm - \frac{1}{2}\frac{a^4}{b^4} + \frac{1}{2}\left(\frac{a^2}{b^2} - n\right)^2\right)(g-t)}{\left((1-p)\,m + pn\right)^2} +$$
$$+ \frac{1}{2}\frac{\left(\frac{c}{d^2} + \left(-\frac{c}{d} - n\right)d^{-1}\right)\left(k^2 - g^2\right)}{\left((1-p)\,m + pn\right)^2} + \frac{\left(nm - \frac{1}{2}\frac{c^2}{d^2} + \frac{1}{2}\left(-\frac{c}{d} - n\right)^2\right)(k-g)}{\left((1-p)\,m + pn\right)^2}. \quad (5.14)$$

Where $t = SeekTime(n)$, $g = SeekTime(e)$, $k = SeekTime(m)$. Putting this value in (5.9), the final formula, which approximates seek time for sequential access to sectors, can be obtained.

5.3 Validation of the model

A software tool was built in order to validate the results, which does low-level[2] seek operation to single sectors. The SCSI disk, that is used, is the IBM DDYS-T36950, as mentioned before. There are two parts in validating the result formulas: firstly, the behavior of the random access formula is discussed compared to the tests done to the hard disk, and secondly, the results for sequential access are demonstrated.

The conditions for the hardware test are the following: there is a disk with 13000 user cylinders and 2000 spare cylinders. All the spare cylinders are placed at the end of the disk. It can also be considered as the beginning of the disk. The idea behind it is that all

[2]The low-level access implies a direct access to a device without using OS I/O subsystem.

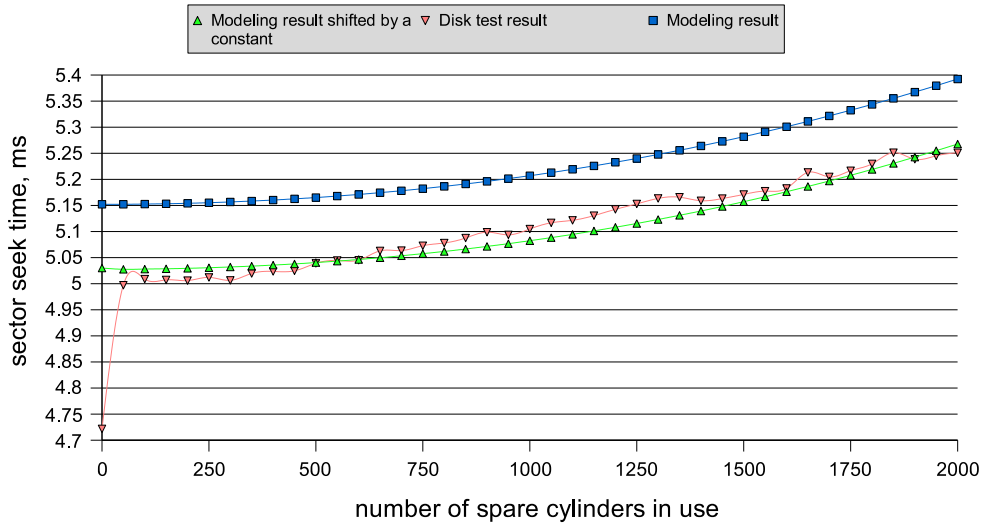


Figure 5.4: Graph displaying the measured-seek-time and modeled-seek-time versus the number of spare cylinders in use for IBM DDYS-T36950 SCSI disk in the case of randomly accessed sectors.

the spare sectors are gathered in one place, not distributed over the disk surface. If a sector becomes defective, then it is remapped to the next available sector in the area of spare cylinders. Since all the tests are made using the entire disk surface (13000 cylinders with an average number of 400 sectors per track, 12 heads, the sector size is equal to 512 bytes, the total capacity of such a disk is 30468 MB), then in order to catch the deterioration in performance the number of remapped sectors should be rather high. In order to comply with these conditions and to be efficient enough, the validation tool remaps sectors so that the entire spare cylinder is filled up with the "defective" sectors at once. Remapped sectors are randomly and uniformly distributed all over the disk.

5.3.1 Random access

In this case seek operations are done to the single sectors only. The requests are randomly and uniformly distributed over the disk. Three curves can be seen in Figure 5.4. The actual data to compare with is a hardware test result taken with the tool created and shown in the figure as $\bigtriangledown$. The second curve is the result of modeling using Formula (5.7), indicated by $\square$. Now it can be seen that the calculated result is rather precise. The result

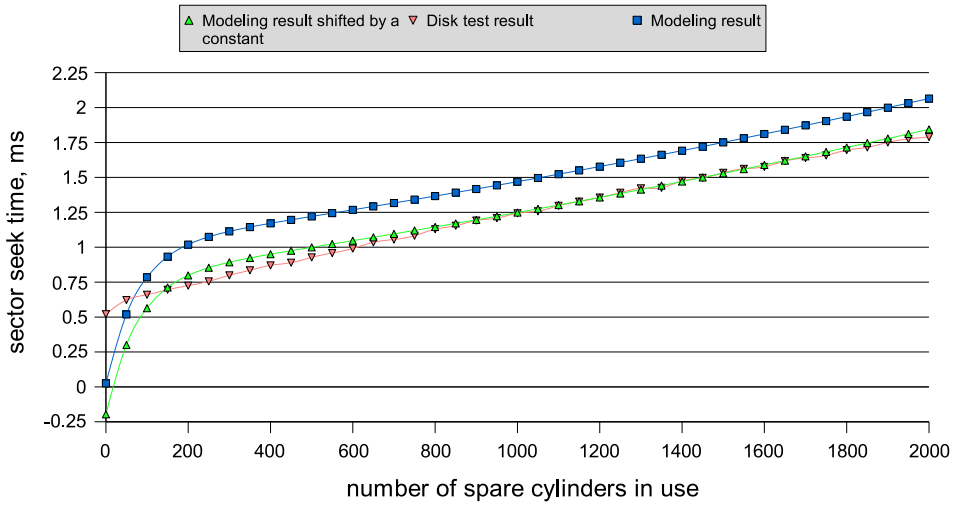


Figure 5.5: Graph displaying the measured-seek-time and modeled-seek-time versus the number of spare cylinders in use for IBM DDYS-T36950 SCSI disk in the case of random requests with the run size of 100 KB.

is not excellent in the case of low numbers of the spare cylinders allocated, but generally the curves grow with the same rate. The average relative error is 2.46%, which is also very good. Since there is a constant shift in the modeled data, which takes place because of the idealization of the using parameters, the shifting constant was found using the method of minimal squared error. It is equal to -0.124660. The third curve depicted and labeled as $\triangle$ gives us the possibility to visually evaluate the correspondence of the curves.

5.3.2 Sequential access

In this case the sequentially processed sectors make up a run, the runs are accessed in the random and uniform fashion. In Figures 5.5, 5.6, 5.7 the results can be seen for the run sizes of 100 KB, 1 MB, 10 MB, respectively.

Each figure contains the same three curves as described in the previous subsection. The hardware tests give the average seek times for one sector, changing the number of the remapped sectors shows the increase in seek time (the curves marked with $\bigtriangledown$). Again, the analytical model gives an imprecise time at low numbers of spare cylinders with remapped sectors but the general rate is rather good (these curves are marked with $\square$). The average

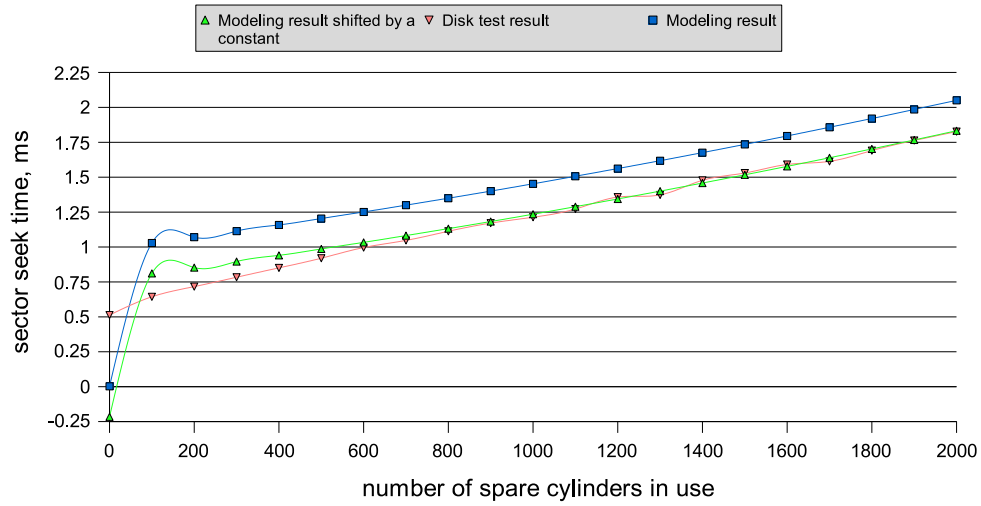


Figure 5.6: Graph displaying the measured-seek-time and modeled-seek-time versus the number of spare cylinders in use for IBM DDYS-T36950 SCSI disk in the case of random requests with the run size of 1 MB.

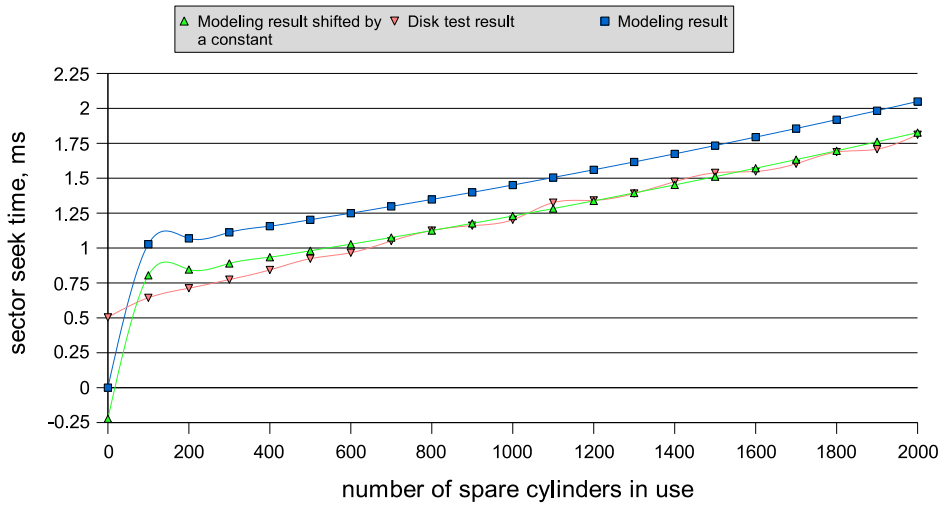


Figure 5.7: Graph displaying the measured-seek-time and modeled-seek-time versus the number of spare cylinders in use for IBM DDYS-T36950 SCSI disk in the case of random requests with the run size of 10 MB.

relative error for the run size of 100 KB is 17.95%, 17.80% for 1 MB and 18.18% is for 10 MB, which are satisfactory. For this case the best shifting constants are also calculated and they are found to be almost the same (as it has to be): -0.221110, -0.217910 and -0.222430 (in the figures marked with $\triangle$). Since all the tests are done independently with different parameters as well as modeling results got, the similarity of the constants proves indirectly validity of the analytical model once again.

5.4 Summary of the model

In this chapter, the basic approaches to modeling the disk systems were considered (Section 5.1). An analytic model was also introduced and considered in Section 5.2. Among all the parameters, the model receives number of remapped sectors as an input and calculates sector seek time for a disk mechanism. This is the first model which takes into account technology of remapping, it is split into two parts: the first one calculates the time for random uniform access to the sectors (see Subsection 5.2.3), the second calculates the seek time for random uniform access to the sequences of the sectors (see Subsection 5.2.4).

The model was validated with a specially created tool, which is capable of reproducing a generic remapping scheme, creating artificial defective sectors, accessing disk sectors and measuring the time. The Results of the validation were given in Section 5.3. The relative error for random access to the sectors is 2.46%, relative errors for random access to the sequentially placed sectors (runs) of size 100KB, 1MB and 10MB are 17.95%, 17.80% and 18.18%, respectively. Small values of the relative errors confirmed high precision of the model.

The modeling showed that for the access to single sector requests, the seek time grows rather slow, resulting in 1.3 times seek time loss comparing results of 2000 spare cylinders in use and 0 ones. The result for requests to the sequences of sectors was more dramatic, 3.5 times, revealing the threat to the performance of disk systems. In order to have this time, 2000 spare cylinders required in a 13000 cylinder-disk, which was the 6^{th} part of the total capacity. Normally, the number of the spare sectors for remapping to be done is much less. This issue will be considered in the next chapter. The modeling showed also that in general case, there is no dramatic hit on the disk performance, however, in some situations the performance may drop down significantly.

Chapter 6

Disk tests

This chapter considers results of the tests taken for three SCSI hard drives. The tests were done with the intent to study the influence of the remapped sectors on the disk drive performance. Section 6.1 contains the general description of the tests with the definition of the tests purpose and conditions. The actual test results are in the following three parts which represent results for the IBM disk drive (see Section 6.2), Fujitsu disk (see Section 6.3) and COMPAQ (see Section 6.4). An analysis of the results can be found in Section 6.5.

6.1 Test description

The goals of the tests are defined in the following two items:

1. There is a need to find out how remapped sectors influence the disk performance: in what fashion this technology impacts the sector seek time, what is the scale of this impact.

2. There is also a need to compare the efficiency of implementation for different remapping schemes.

6.1.1 What has been done

In order to achieve the goals, the following tests were performed:

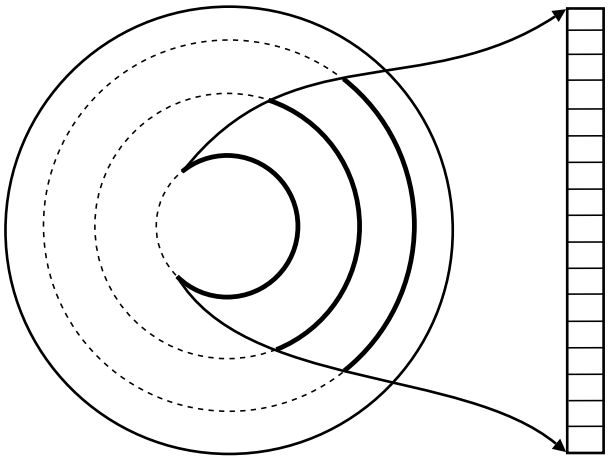


Figure 6.1: Reading a limited length request by sectors at the beginning/middle/end of a hard drive. The length of the request size is a fixed value. Remapped sectors are uniformly and randomly distributed over the request.

1. Reading a limited length request by sectors at the beginning/middle/end of a hard drive while increasing the number of the remapped sectors. Remapped sectors are uniformly and randomly distributed over the request length with run length of 1 sector (see Figure 6.1).

2. Reading a limited length request by sectors at the end of a hard drive while increasing the number of the remapped sectors for each request read operation. Remapped sectors are uniformly and randomly distributed over the request length with run length of 5 and 20 sectors (see Figure 6.2).

The request size is set to be 10 MB. The beginning of the disk implies first sectors which constitute the request size, the end of the disk implies last sectors, which constitute the request size, respectively. For the middle of the disk, the first sector for the request is taken by dividing the total number of the sectors for a disk drive by two.

A *run of remapped sectors* is a number of adjacent sectors[1] which have been remapped. The nature of the defects appearing on the disk surface can be different, but there are two main scenarios:

1. Adjacent defects arise at once. In this case all these sectors are remapped to the

[1]It means that LBA's for these sectors are consecutive numbers (see Chapter 3 for more information).

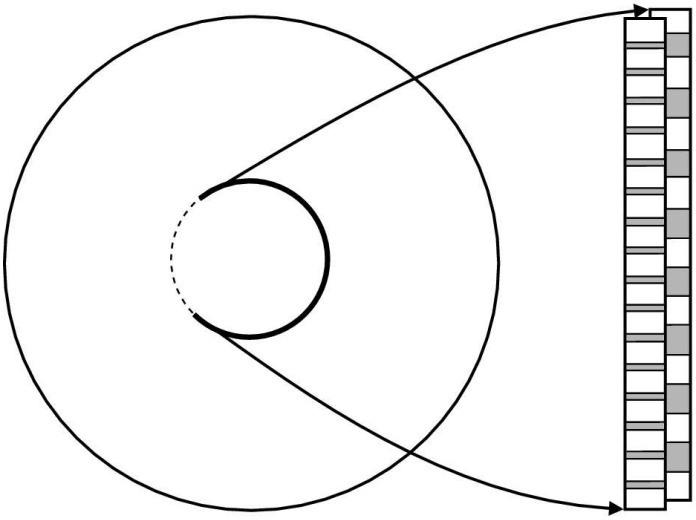


Figure 6.2: Reading a limited length request by sectors at the end of a hard drive. The length of request size is a fixed value. Remapped sectors are combined into runs. Tests performed for the runs of lengths 5 and 20 sectors.

same spare area meaning that if disk electronics is optimized enough, then disk head after having moved to the spare area reads all the adjacently remapped sectors without any further extra moves back, and only after that it returns to the sectors of interrupted request.

2. Adjacent defects arise as time goes by. In this case there is no a 'remapped chunk' which could be read at once. Theoretically it can happen so that some of the adjacent defects have been remapped to the same track, others to some neighboring disk zone and the remaining part found its place at the end of the cylinder space. This scenario will require much head movement.

The runs of the remapped sectors in the tests taken are produced at once.

It would be also useful to develop tests for measuring seek time with random requests across all the disk space but because of the REASSIGN BLOCKS command [Com97, Com04, Com05c] restriction[2], the tests would not be effective enough in achieving the

[2]This command lets us have only a small limited number of the remapped sectors while hard drives have millions of sectors. For instance, having 35,566,478 sectors in total and $\approx$ 1941 as a maximum number of the remapped sectors gives us little chance to measure remapped sectors influence on general performance of this disk.

goal. This restriction is set up by disk manufacturers and is not discussed in the SCSI specifications.

6.1.2 Test setup

Three types of hard drives were used for the tests of different capacity and manufacturer:

- IBM DDYS-T36950M [IBM00] (71,132,959 sectors; 36420 MB);
- Fujitsu MAJ3182MC [Fuj00] (35,566,478 sectors; 18210 MB);
- COMPAQ BD009635C3 (17,773,524 sectors; 9100 MB);

During the tests, real hardware sectors were remapped using the SCSI command REASSIGN BLOCKS. This command was used in order to create a G-LIST list of defective sectors. Once created, this list can be cleaned using FORMAT UNIT command[3][Com97, Com04, Com05c, Com05a, Com05b]. For actual tests, READ(10) SCSI command[4] was used [Com97] (this command can be used since the data transferring time itself is a constant but seeking the requested sector is what is measured and the READ(10) performs this seeking). The command transfers data from a disk device to the host. For the convenience and due to the established terminology *service time* of this command is referred hereafter to as *read time*.

In addition, "GSL - GNU Scientific Library" [GT05] was used since it gives much better support [GDT+04] for random numbers than the standard C/C++ library, the SG SCSI driver was used to access the disk devices [Gil05]. The tests were taken under Linux OS with kernel 2.6.11.

[3]This is one of the reasons why SCSI hard drives were used - ATA hard drives do not support this command. Therefore, there is no standard way to remove the list of remapped sectors what in turn does not allow us to perform different tests using the same disk (since created G-LIST cannot be removed).

[4]There is also SEEK(10) command that, according to [IBM00], is "The SEEK command requests the drive to seek the specified logical block address", or to [Com97, Com04, Com05c], it is "The SEEK(10) (see Table 28) command requests that the block device seek to the specified logical block address. This command is included for device types based on the MMC standard. This command allows the host to provide advanced notification that particular data may be requested in a subsequent command.", but by results of the tests it does nothing (at first, this command had been used). You may want to have a look at the result curves for the hard drives. For that, see Appendix A. This command cannot be used for hard drive performance measurements in case of the remapped sectors.

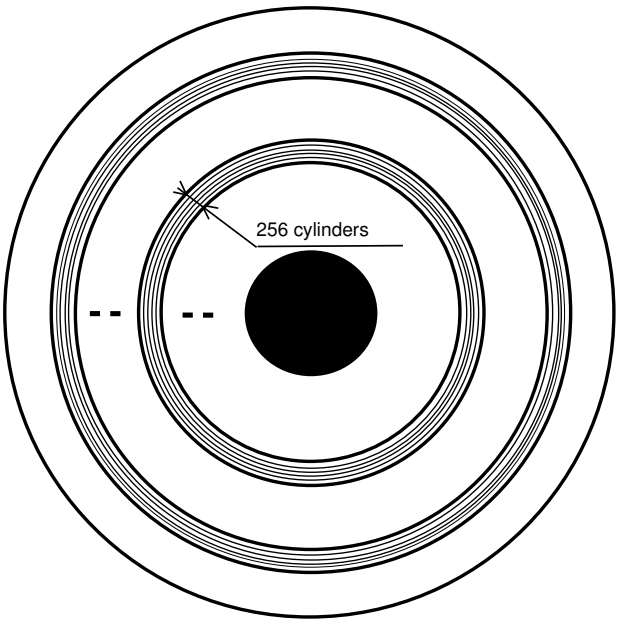


Figure 6.3: Location of the spare cylinders for the IBM DDYS-T36950M hard drive.

6.2 IBM hard drive

This section provides with the technical details of reassigning scheme and test results for the IBM DDYS-T36950M hard disk drive.

6.2.1 Remapping scheme

In this hard drive, there is one spare cylinder every 256 user cylinders (see Figure 6.3). Knowing the zone map – the number of cylinders for each zone and number of sectors per track in those cylinders for different zones, the number of spare sectors can also be calculated. For this hard drive, the total number of the spare sectors is ≈ 278760, which is 0.31% of the total capacity.

6.2.2 Test results

In Figure 6.4, test results can be found taken for three different parts of a platter (firstly, cylinders at the beginning of the hard drive were accessed, then in the middle and after that at the end of the disk). This test shows how the applied scheme reacts on the 'defects'

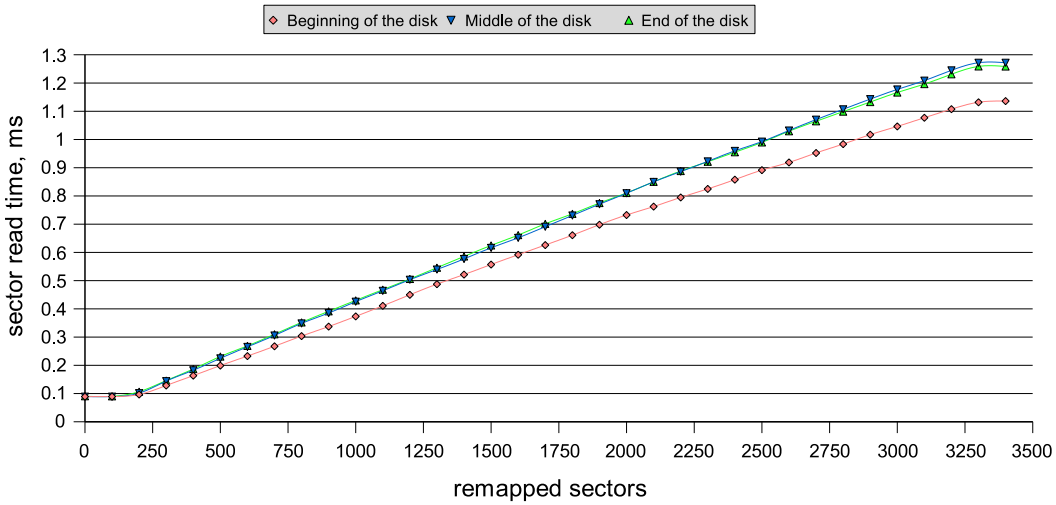


Figure 6.4: The mean time needed to read one sector in a 10 MB request versus the number of remapped sectors for the IBM DDYS-T36950M hard drive.

growing at the tested parts of the disk surface.

It should be noticed, that the read time stops growing after having reached border of ≈ 3250 remapped sectors – it happens because of the restriction of the G-LIST length by this number[5] (not because remapping stops affecting seek time). The same can be found on the remaining graphics.

The first thing here, which catches the eye, is that this scheme almost does not react to the growing remappings until this number exceeds the limit of 200 sectors. After linear growth, it stops at the number of ≈ 3250 remapped sectors resulting in more than 10 times increasing of the mean time to read one sector. Another thing is that graphics built for the middle part of the disk and for end are almost the same. I suppose the nature of this is due to the relatively large size of the hard disk (36420 MB) in comparison with length of the request (10 MB), but of course there are probably more reasons for that (location of the requested sectors for all the cases, reading head in use, etc.). Now let us consider numerical data in order to get a better idea of how the mean read time changes:

[5] It was already mentioned about the REASSIGN BLOCKS command restriction referring to the same problem. Unfortunately, it is not clarified either in SCSI specs or in disk manuals which one is more correct. If we think of that as of REASSIGN BLOCKS command restriction, then there are many more sectors remain (see calculations of total number of the spare sectors), on the other hand, extra disk space is needed to store reassigning table.

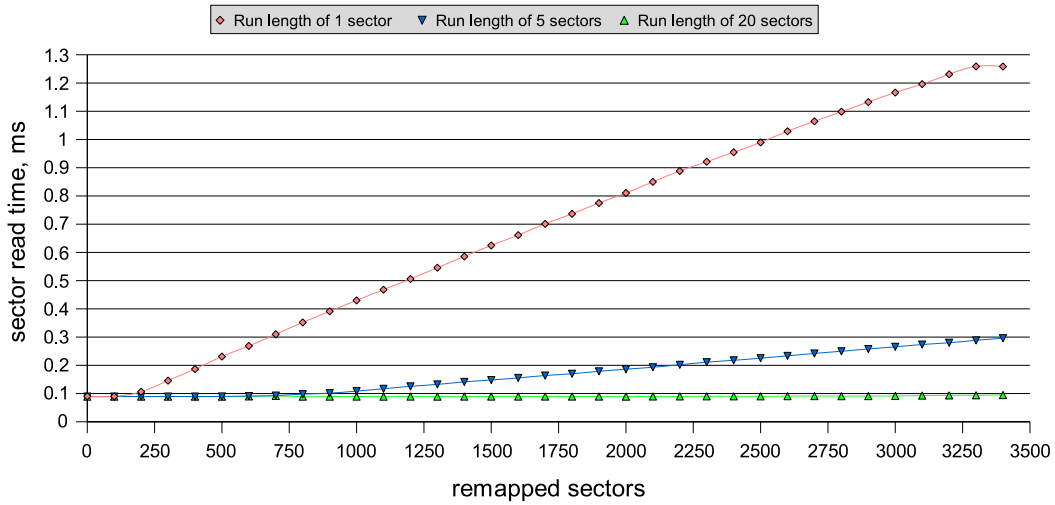


Figure 6.5: The mean time needed to read one sector in a 10 MB request versus the number of remapped sectors in runs for the IBM DDYS-T36950M hard drive.

- For the beginning of the disk – $1.132/0.089 = 12.7191$ times;
- For the middle of the disk – $1.271/0.089 = 14.2808$ times;
- For the end of the disk – $1.259/0.090 = 13.9888$ times;

In Figure 6.5 is shown how length of a run of remapped sectors affects the mean sector read time. Tests were done for the end of the disk. There is a large difference can be observed between runs of 1 sector and runs of 5 and 20 sectors. It is also worth paying attention to, that having a run length of 5 sectors, the read time starts growing only after ≈ 800 remapped sectors, and even more intriguing, it is to not growing at all with runs of 20 sectors. The reason for that is described in Section 6.1.1, in the case of runs of remapped sectors disk head produces much less movements, which results in better time.

Therefore:

- For the run of 5 sectors the mean sector read time increases $0.299/0.089 = 3.3595$ times;
- For the run of 20 sectors the mean sector read time changes $0.095/0.089 = 1.0674$ times;

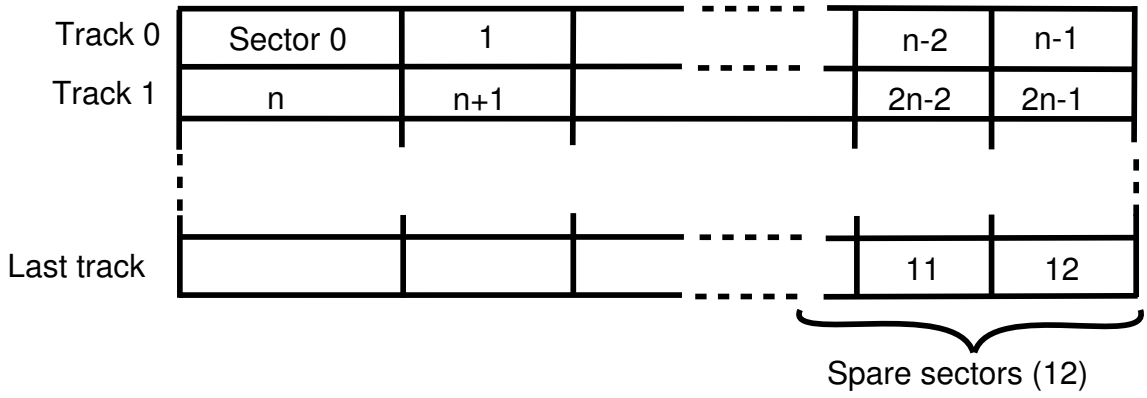


Figure 6.6: Location of the spare sectors for each cylinder in the Fujitsu MAJ3182MC hard drive.

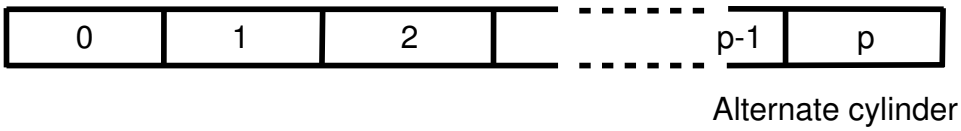


Figure 6.7: Location of the alternate cylinder in the cylinder space for the Fujitsu MAJ3182MC hard drive.

6.3 Fujitsu hard drive

This section provides with the technical details of reassigning scheme and test results for the Fujitsu MAJ3182MC hard disk drive.

6.3.1 Remapping scheme

There are 12 spare sectors in the last track of each cylinder (see Figure 6.6) for this model and one spare cylinder (Figure 6.7) at the very end of the cylinder space. An alternate cylinder is used when all 12 spare sectors are used up for a user's cylinder. Having the total number of the user cylinders (14807) and the number of the spare sectors per cylinder (12), the total number of the spare sectors can be calculated for this certain case (assuming, the number of the sectors in alternate cylinder equals the last one, i.e., 369(sectors per

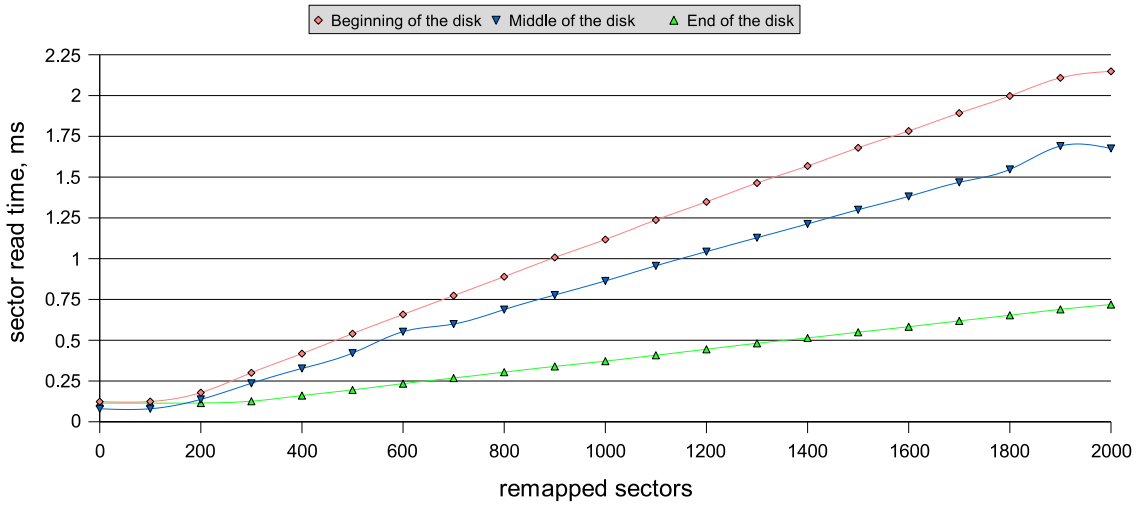


Figure 6.8: The mean time needed to read one sector in a 10 MB request versus the number of remapped sectors for the Fujitsu MAJ3182MC hard drive.

track) * 5(number of heads)):

$$14807 * 12 + 369 * 5 = 179529$$

This number is 0.5% of the total capacity of the disk. The G-LIST restriction for the number of the sectors to reassign with the REASSIGN BLOCKS command is $\approx$ 1900, which is 0.005% of the total capacity of the disk.

6.3.2 Test results

Figure 6.8 shows us wider intervals between the curves for the Fujitsu hard drive than for the IBM one. In the worst case (for the beginning of the disk[6]) the time increases $\approx$ 17 times, in contrast to the IBM disk, for the end of the disk time increases $\approx$ 6 times. The same manner can be seen for the indifference of the time increasing on the low number of the remapped sectors ($\approx$ 200). Limit for the G-LIST in this case is $\approx$ 1900 remapped sectors.

[6]Pay attention to the descending order of the curves for this disk by time – the worst case is for the beginning of the disk, the best for the end of the disk, whereas the IBM hard drive has it vice versa.

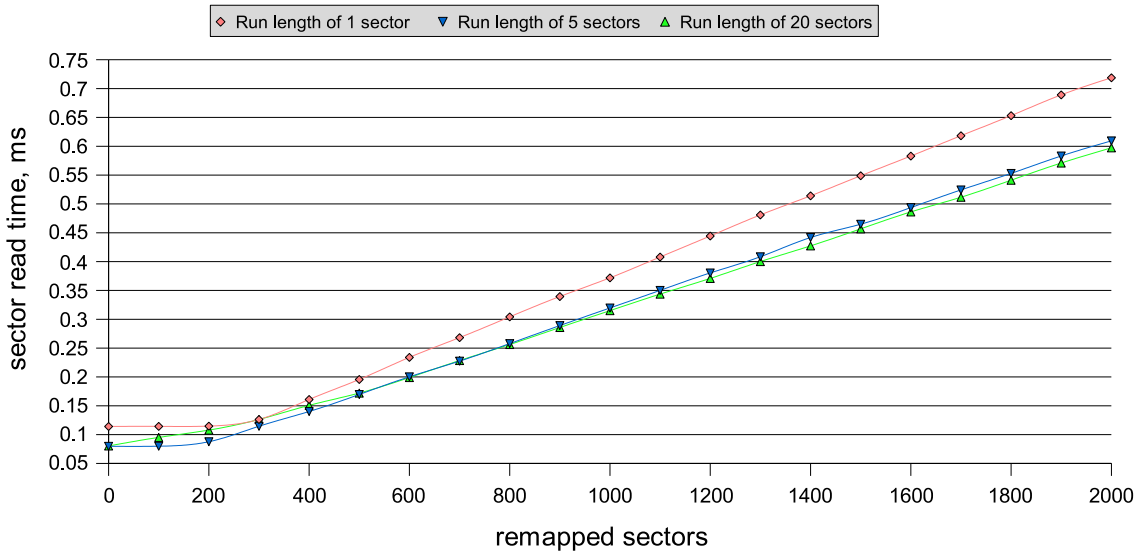


Figure 6.9: The mean time needed to read one sector in a 10 MB request versus the number of remapped sectors in runs for the Fujitsu MAJ3182MC hard drive.

Let us calculate how much the mean sector read time changes to be precise:

- For the beginning of the disk, there is an increase of $2.148/0.123 = 17.4634$ times;
- For the middle of the disk, there is an increase of $1.676/0.080 = 20.9500$ times;
- For the end of the disk, there is an increase of $0.719/0.114 = 6.3070$ times;

Figure 6.9 gives us a slightly different result than 6.5 does. The practical result is the same – the time with runs of 1 sector length has a worse mean read time than runs of 5 and 20 sectors, though the break between those is not as big as in Figure 6.5.

In Figure 6.9 the early growth of the curve with the run length of 20 sectors can be seen, unveiling a weak point of the long length runs in such a case of the remapping scheme (all 12 spare sectors in the last track are used up at once), even though the incline is the less rate than with lengths of 5 and 1 sectors, which start growing respectively. The time difference between the run length of 5 sectors and run length of 20 sectors is very small (in contrast to the previous test for runs).

For these tests the time results are the following:

- Having a run of 5 sectors, the mean read time changes $0.609/0.079 = 7.7088$ times;
- Having a run of 20 sectors, the mean read time changes $0.597/0.080 = 7.4625$ times;

6.4 COMPAQ hard drive

This section provides with the technical details of reassigning scheme and test results for the COMPAQ BD009635C3 hard disk drive.

6.4.1 Remapping scheme

This hard drive also has a compound reallocating scheme for defective sectors[7]. There are 12 spare sectors for each cylinder (in last track) and 75 spare cylinders at the end of the disk cylinder space (this scheme is similar to the one applied in the Fujitsu hard drive). After all the sectors in one cylinder are used up, sectors from the end of the disk are involved. For this hard drive, there are $12 * 13187 + 336 * 75 * 3 \approx 233844$ spare sectors, this is 1.31% of the total capacity of this disk.

6.4.2 Test results

The result, shown in Figure 6.10, is similar to the one depicted in Figure 6.8 – all three curves grow in their own fashion. The closer the requests to the end of the disk, the smaller the slope. It is interesting to notice, that for this scheme the closer the request to the end of the disk, the more remapped sectors are needed in order to affect the read time(see Figure 6.10) – for the beginning of the disk this number is ≈ 200 sectors, for the middle of the disk it is already ≈ 300 and for the end of the disk, the curve starts growing having ≈ 500 remapped sectors. Limit for the G-LIST in this case is ≈ 1200 remapped sectors.

The times calculated for this hard drive are:

- For the beginning of the disk – $1.196/0.173 = 6.9132$ times;
- For the middle of the disk – $0.972/0.173 = 5.6184$ times;

[7]Unfortunately, I was not able to find the technical specification for this hard drive. Therefore, I used technical information (number of the zones, cylinders, heads, zone map etc.) retrieved from the disk directly. In order to trace remapping scheme, I had the same LBAs remapped one, two, three and four times reading the physical addresses of these sectors.

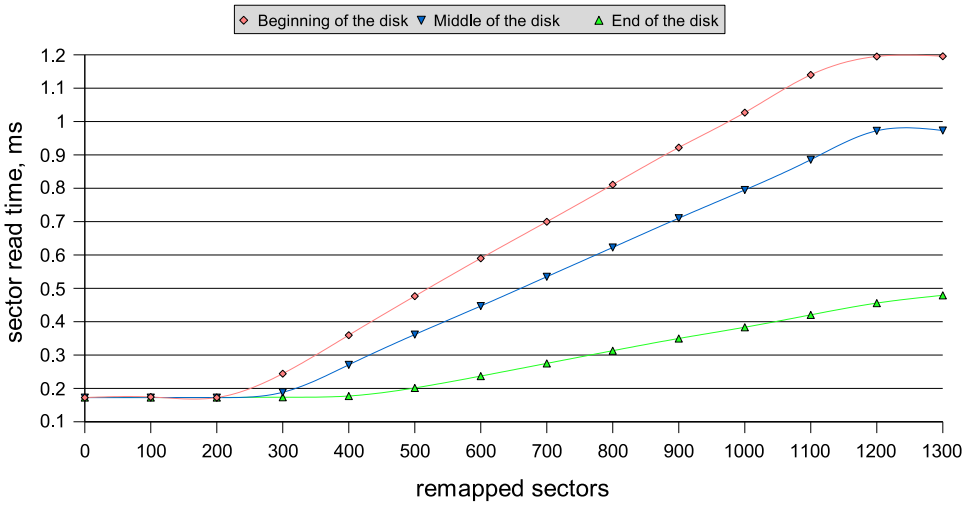


Figure 6.10: The mean time needed to read one sector in a 10 MB request versus the number of remapped sectors for the COMPAQ BD009635C3 hard drive.

- For the end of the disk – $0.479/0.173 = 2.7687$ times;

The curves shown in Figure 6.11 are also similar to the ones taken from the Fujitsu hard drive, including also the order in which these curves start growing (there is the same limit of 12 sectors in the last track for each cylinder), with one exception – they all start the growth after exceeding the border of $\approx$ 400 remapped sectors. The curves for the middle and end of the disk have almost the same tendency.

Therefore, the mean sector read time changes as it follows:

- For a run of 5 sectors – $0.398/0.173 = 2.3005$ times;
- For a run of 20 sectors – $0.389/0.173 = 2.2485$ times;

6.5 Comparing and analyzing the results

In this section, the results obtained from three different hard drives are discussed and analyzed. First of all, let us try to understand – why the mean read time for the Fujitsu and COMPAQ hard drives decreases, approaching the end of the disks, whereas it increases (but not significantly) for the IBM disk. The IBM disk is known to have a spare cylinder

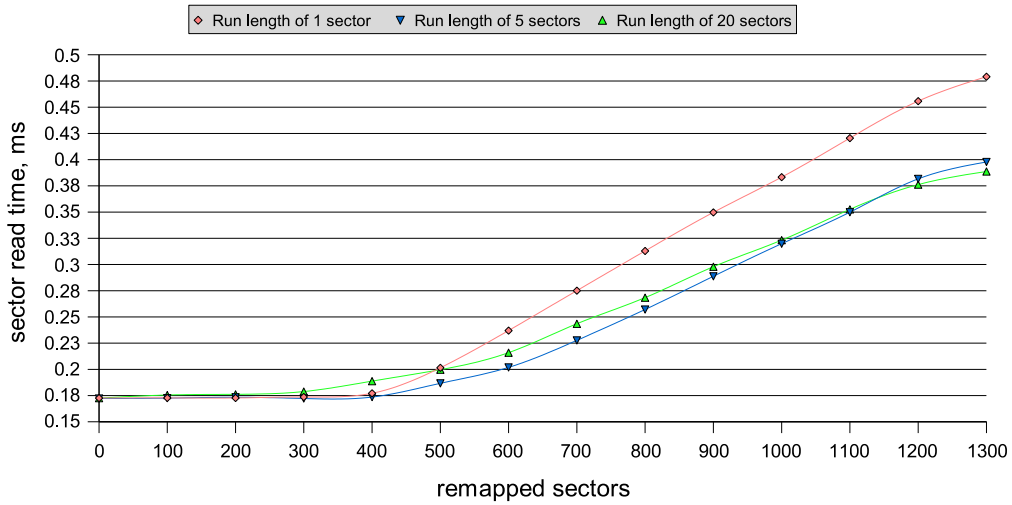


Figure 6.11: The mean time needed to read one sector in a 10 MB request versus the number of remapped sectors in runs for the COMPAQ BD009635C3 hard drive.

every 256 user cylinders. In the Fujitsu and COMPAQ disks, there are 12 spare sectors for each cylinder and an extra space at the end of the disk. In Chapter 4 it was already discussed, that the angular velocity is higher for the beginning of disk in comparison to its end. Therefore, the IBM disk gives us time results as they should be, the other two hard drives do not follow the common sense. Such the behavior is observed because of 12 spare sectors limit for one cylinder - after it is used up, the disk head starts moving to the end of the drives. Having requests in the very beginning of disk, the head has to carry out a long way accessing the reassigned sectors, which are at the very end of the disk. For such kinds of remapping schemes it is more advantageous to have requests at the end of disk in case of an increasing number of defective sectors – a tradeoff between angular velocity and the growing number of remapped sectors. Since the IBM disk has a different remapping scheme, the number of the defective sectors does not change its ordinary speed state. Looking at the curves in Figures 6.12, 6.13 and 6.14, it can be seen why the Fujitsu and COMPAQ disks have an advantage for a request stretching to the end of the disk. On the other hand, mean time for the IBM disk remains almost the same – its scheme provides more dispersed remapped sectors giving less loss in the time and more stability.

From the graphics, it can also be seen, and as previously mentioned, that both the Fujitsu and COMPAQ disks have almost the same remapping scheme but the closer the

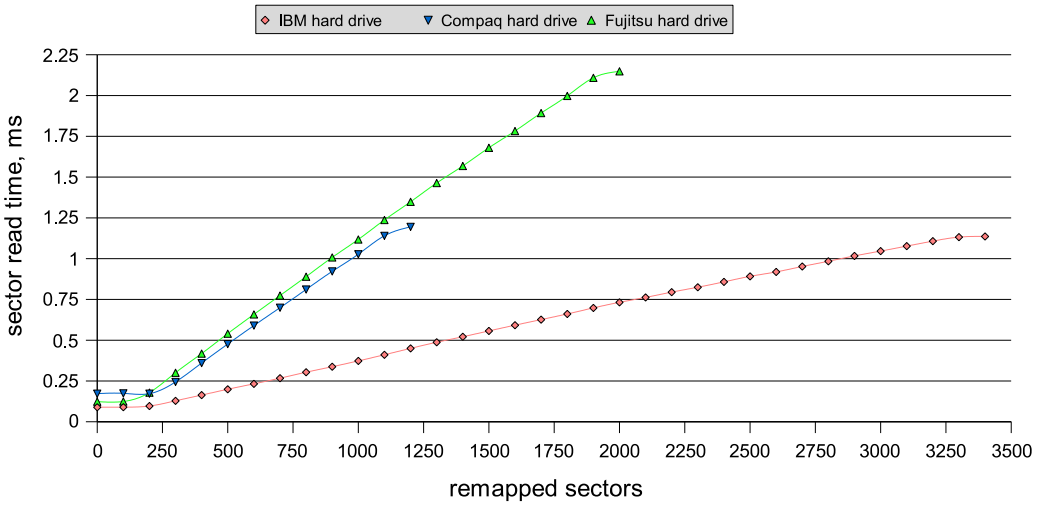


Figure 6.12: The mean time needed to read one sector in a 10 MB request accessed at the beginning of the disk space versus the number of remapped sectors for different hard drives.

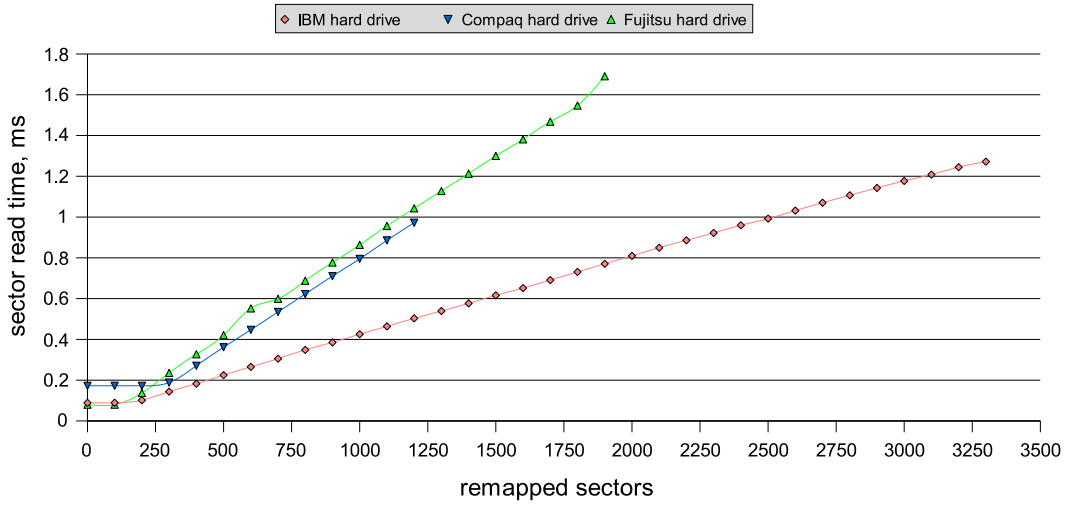


Figure 6.13: The mean time needed to read one sector in a 10 MB request accessed in the middle of the disk space versus the number of remapped sectors for different hard drives.

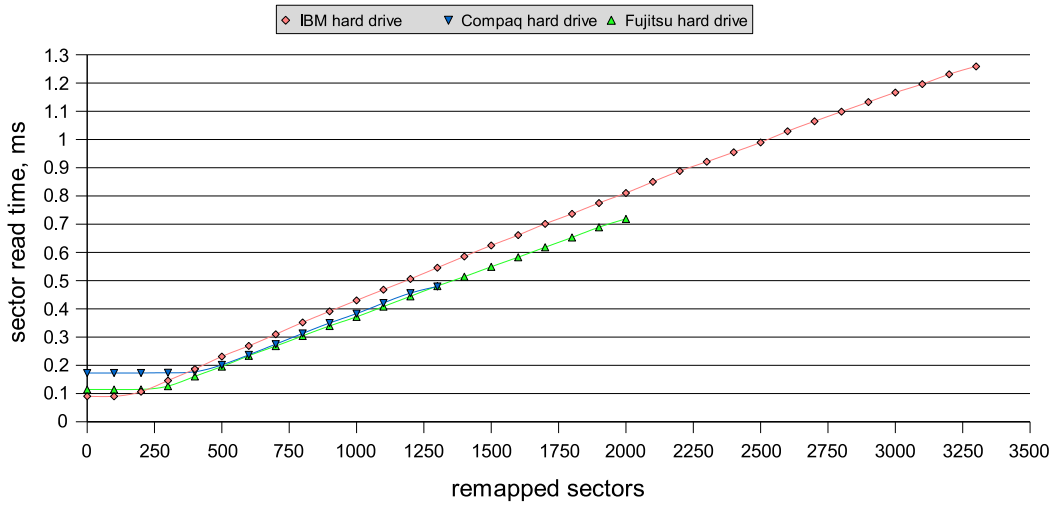


Figure 6.14: The mean time needed to read one sector in a 10 MB request accessed at the end of the disk space versus the number of remapped sectors for different hard drives.

request is done to the end of the cylinder space, the later the mean seek time starts growing. The reason for that is the number of the heads for these disks – 5 in case of the Fujitsu HDD [Fuj00] and 3 in case of the COMPAQ. Written on the disk's surface, the request takes more cylinders for the COMPAQ disk than for the Fujitsu one, therefore the limit of 12 sectors per cylinder takes more time to consume. For such scheme, the less platters and more cylinders are used in a hard drive, the better.

Figure 6.12 shows the similar slopes for the Fujitsu and COMPAQ disks' times and an impressively slow growing of the time for the IBM disk. In spite of a far larger number of remapped sectors (1.5 in comparison with Fujitsu disk and almost 3 times comparing to the COMPAQ drive), its time remains eventually lower than COMPAQ's curve and twice as low as Fujitsu's curve. It proves once again the sufficiency of the scheme used in this HDD.

In Figure 6.13, the IBM disk keeps the leader's status by having its time in the best position among the competitors. In Figure 6.14 all three curves have an almost identical angle. The Fujitsu and COMPAQ drives do the best time since it is end of the cylinder and disk space. Figure 6.15 and 6.16 depict how these disks behave with various lengths of the remapped sectors runs. The difference in the curves can be seen in the figures.

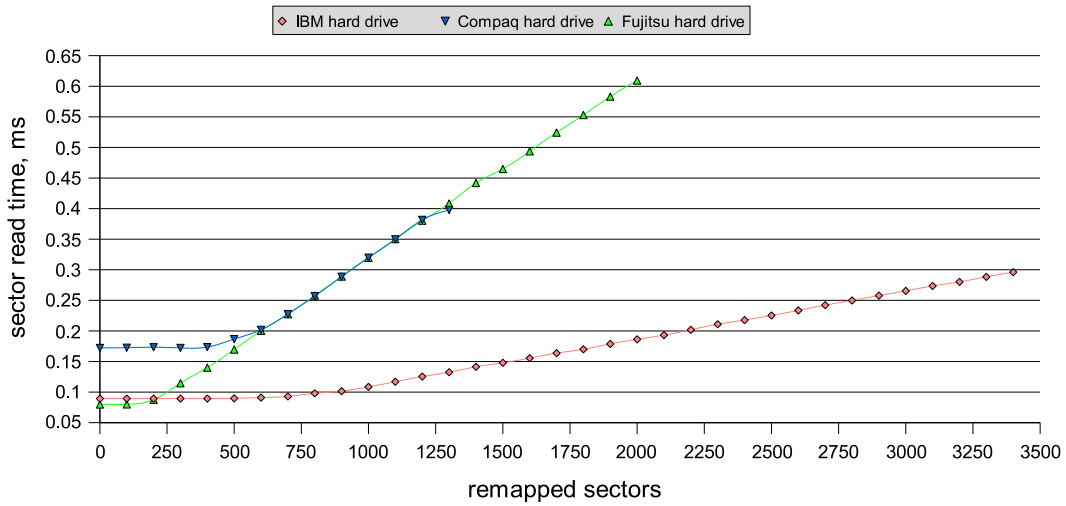


Figure 6.15: The mean time needed to read one sector in a 10 MB request accessed at the end of the disk space versus the number of remapped sectors in runs of length 5 sectors for different hard drives.

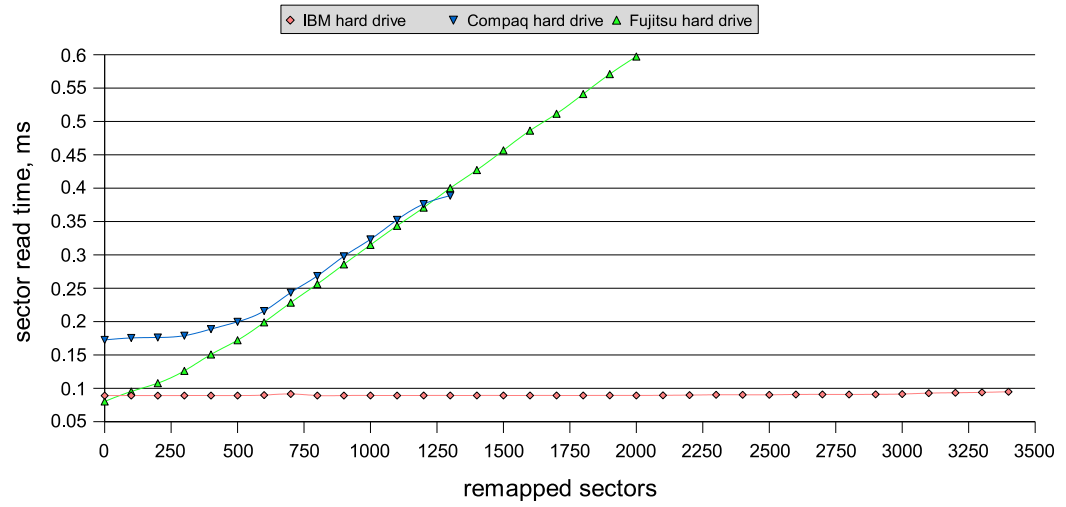


Figure 6.16: The mean time needed to read one sector in a 10 MB request accessed at the end of the disk space versus the number of remapped sectors in runs of length 20 sectors for different hard drives.

6.6 Summary of the tests

In this chapter, the results of the remapping scheme implementation tests were considered for three different brands of disk drives. One of the goals of the tests stated in Section 6.1, was to define how remapping influences on the disk performance and what is the degree of this influence. In Sections 6.2, 6.3 and 6.4 the mean sector seek time was established to vary for different parts of the IBM disk with different number of the remapped sectors from ≈ 13 to ≈ 14 times, for the Fujitsu disk this value varies from ≈ 6 to ≈ 20 times and in case of the COMPAQ disk, from ≈ 3 to ≈ 7. Having 5 and 20 sectors in runs of remapped sectors, the mean seek time varies for the IBM disk from 1 to 3 times, in case of the Fujitsu disk 7 times and 2 times for the COMPAQ drive. Even though the described tests were done for the restricted areas on the disks, the significant growth in time points to the importance of taking the problem of remapped sectors into account, modeling, simulating and planning disk systems, since the performance loss can be considerable. The other goal was to compare efficiency of the implementation for these three schemes, this was done in Section 6.5.

Chapter 7

Conclusions and future work

In this work the main subject was the mechanism of sectors remapping for hard disk drives. The main components of disks were considered, and a description was given on how the technology interacts with low-level disk structure as well as various implementations of the technology. Since every manufacturer has its own technology implementation, the information is not highly published neither in the technical books nor in the device manuals or technical papers. Therefore, an attempt was made to systematize information regarding the subject.

An analytic model was also developed which calculates the disk seek time versus the number of the remapped sectors. As an implementation of the remapping mechanism, a generic scheme was used in which the only spare area for the remapped sectors is at the end of sector space placed in the last cylinders. The modeling was performed for the entire disk space. Validation of the model showed that the relative error for random access to the sectors is 2.46%, relative errors for random access to the sequentially placed sectors of size 100KB, 1MB and 10MB are 17.95%, 17.80% and 18.18%, respectively. Small values of the relative errors confirm high precision of the model.

The tests were also performed on real hard disk drives in order to determine how the implementation of the remapping mechanism affects sequential requests of various lengths. The tests showed that for different parts of the IBM DDYS-T36950M disk, the mean sector seek time with a different number of the remapped sectors within a request size of 10 MB varies from $\approx$ 13 to $\approx$ 14 times, for the Fujitsu MAJ3182MC disk this value varies from $\approx$ 6 to $\approx$ 20 times and in case of the COMPAQ BD009635C3 disk from

$\approx$ 3 to $\approx$ 7. The modeling for the access done to a single sector requests also showed that the seek time grows slowly resulting in $\approx$ 1.3 times seek time loss comparing results of 2000 spare cylinders in use and 0 ones. Different result was obtained for requests to sequences of the sectors, $\approx$ 3.5 times. These numbers prove that there can be a dramatic hit on disk performance, even though the condition of having all the spare sectors used up is hard to see in reality, but if there is a frequently requested place on the disk surface and a growing number of the remapped sectors within this area then the access time is worsened significantly.

This thesis studied how the remapping mechanism affects disk performance for the whole disk area as well as for some parts of the disk and established how the seek time behaves. As a further improvement to the mechanism, I would suggest a special buffer, which could cache remapped sectors only. It could be a sort of volatile memory, which caches remapped sectors after power is switched on or a sort of flash memory, which could store them permanently. Knowing the access time to the different types of memories and access time to the sectors, further numerical calculations or modeling could be taken.

The model could be extended in order to create a more universal one, which would consider different remapping schemes as described in Chapter 4. Adding the feature of sequentially placed remapped sectors to the model would also make it closer to the real life disk drives. Studying distributions of the remapped sectors could also help in further work on the subject but it is almost an unfeasible task without support of the disk manufacturers who carry a variety of disks[1].

One more option to proceed the work is to use simulators. The DiskSim [BG03] was probed into and was found to be precise enough with some support of the remapping mechanism. The source code is available [BG05] and is under constant development. Supported remapping schemes are [BG03]:

- entire tracks of spare sectors are allocated at the "end" of some or all zones (sets of cylinders);
- spare sectors are allocated at the "end" of each cylinder;
- spare sectors are allocated at the "end" of each track;
- spare sectors are allocated at the "front" of each cylinder;

[1] I tried contacting Samsung, IBM, Seagate and a few others but did not succeed.

- spare sectors are allocated at the "end" of the disk;
- spare sectors are allocated at the "end" of each range of cylinders;
- spare sectors are allocated at the "end" of each zone;

Simulators could help, for example, with placing disks into the conditions which are hard to achieve in reality.

Appendix A

Tests done with SEEK command

SCSI and disks specifications [Com97, Com04, Com05c, IBM00, Fuj00] claim that

> The SEEK command requests the drive to seek the specified logical block address.

This command was used in the first tests but the results obtained did not seem to correspond to the papers. Here some of the results are given in order to prevent the further use of this command in such a scope of research. The tests that were taken are the same as described in Chapter 6 but SEEK SCSI command had been used instead of the READ command utilizing the same disks. Apparently, this effect takes place because of the SEEK command optimization, which makes the arm move to the sector (track/cylinder) requested only in the case of actual read/write operation for remapped sectors but this is not mentioned neither in SCSI nor in other disk specifications. The perturbation of the graphics in Figure A.1 can be explained in terms of sensitivity of the system.

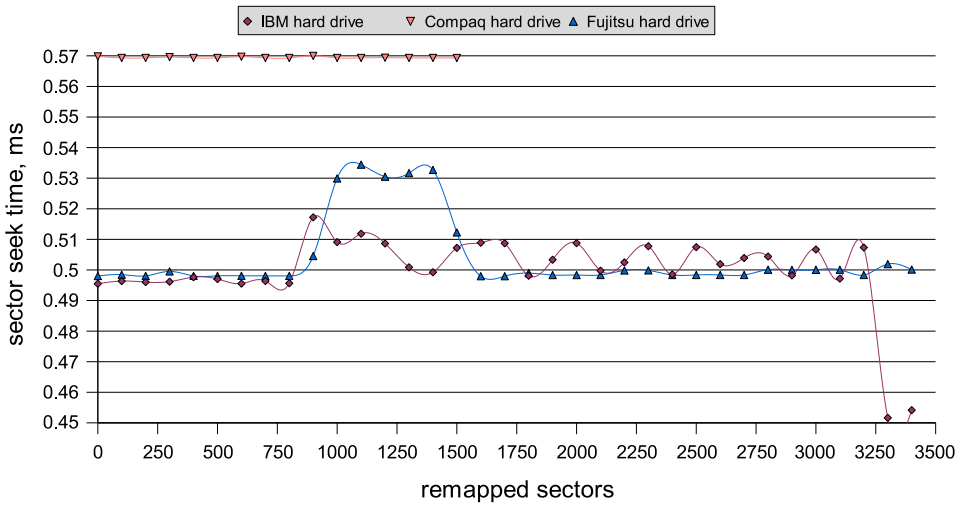


Figure A.1: Graphics showing the independence of the seek time for various hard drives from an increasing number of remapped sectors.

References

[AL93] Martin Abadi and Leslie Lamport. *Conjoining specifications.* Technical Report 118, Digital Equipment Corporation System Research Center, Palo Alto, CA, December 1993.

[And01] Eric Anderson. *Simple table-based modeling of storage devices.* Technical Report HPL-SSP-2001-4, Storage and Content Distribution Department, Hewlett-Packard Laboratories, Palo Alto, 2001.

[BG03] John S. Bucy and G. R. Ganger. *The DiskSim Simulation Environment Version 3.0 Reference Manual.* Technical Report CMU-CS-03-102, Carnegie Mellon University, School of Computer Science, January 2003.

[BG05] John Bucy and Greg Ganger. The DiskSim Simulation Environment (Version 3.0). Internet WWW-page, URL: `http://www.pdl.cmu.edu/DiskSim/`, June 2005.

[Com97] T10 Technical Committee. *Working Draft. Revision 8c. Information Technology - SCSI-3 Block Commands (SBC).* Technical Committee of Accredited Standards Committee NCITS, November 1997.

[Com04] T10 Technical Committee. *Working Draft. Revision 16. Information technology - SCSI Block Commands - 2 (SBC-2).* Technical Committee of Accredited Standards Committee INCITS, November 2004.

[Com05a] T10 Technical Committee. *Working Draft. Revision 2. Information technology - SCSI Primary Commands - 4 (SPC-4).* Technical Committee of Accredited Standards Committee INCITS, September 2005.

[Com05b] T10 Technical Committee. *Working Draft. Revision 23.Information technology - SCSI Primary Commands - 3 (SPC-3).* Technical Committee of Accredited Standards Committee INCITS, May 2005.

[Com05c] T10 Technical Committee. *Working Draft. Revision 3. Information technology - SCSI Block Commands - 3 (SBC-3).* Technical Committee of Accredited Standards Committee INCITS, November 2005.

[Dav05] Leroy Davis. Electronic Interface Buses. Internet WWW-page, URL: `http://www.interfacebus.com/Interface_Bus_Types.html`, September 2005.

[Fuj00] Fujitsu. *C141-E103-02EN: MAH3182MC/MP SERIES, MAH3091MC/MP SERIES, MAJ3364MC/MP SERIES, MAJ3182MC/MP SERIES, MAJ3091MC/MP SERIES DISK DRIVES PRODUCT/MAINTENANCE MANUAL.* Fujitsu, 2nd edition, December 2000.

[GDT+04] Mark Galassi, Jim Davies, James Theiler, Brian Gough, Gerard Jungman, Michael Booth, and Fabrice Rossi. *GNU Scientific Library. Reference Manual.* The GSL Team, www.gnu.org, 1.6th edition, December 2004.

[Gil05] Douglas Gilbert. The Linux SCSI Generic (sg) Driver. Internet WWW-page, URL: `http://sg.torque.net/sg/`, August 2005.

[GT05] Mark Galassi and Jim Theiler. GSL - GNU Scientific Library. Internet WWW-page, URL: `http://www.gnu.org/software/gsl/`, July 2005.

[Gut96] Peter Gutmann. Secure deletion of data from magnetic and solid-state memory. In *Sixth USENIX Security Symposium Proceedings.* Department of Computer Science, University of Auckland, 1996.

[HP03] John L. Hennessy and David A. Patterson. *Computer Architecture: A Quantitative Approach.* Morgan Kaufmann Publishers, Incorporated, San Mateo, CA, third edition, 2003.

[IBM00] IBM. *S31L-8989-06: Hard disk drive specifications, Ultrastar 36LZX, 3.5 inch SCSI hard disk drive, Models: DDYS-T36950, DDYS-T18350, DDYS-T09170.* IBM, 6th edition, June 2000.

[KB88] M. A. Ketabchi and V. Berzins. Mathematical model of composite objects and its application for organizing engineering databases. *IEEE Transactions on Software Engineering*, 14(1), January 1988.

[LC87] Shui F. Lam and K. Hung Chan. *Computer capacity planning: theory and practice*. Academic Press, 1987.

[Loha] John Lohmeyer. SCSI-3 Standards Architecture. Internet WWW-page, URL: `http://www.t10.org/scsi-3.htm`. Link is valid at August 2005.

[Lohb] John Lohmeyer. T10 Draft Standards and Technical Reports. Internet WWW-page, URL: `http://www.t10.org/drafts.htm`. Link is valid at August 2005.

[Map06] Maplesoft. MapleSoft. Internet WWW-page, URL: `http://www.maplesoft.com/products/maple/index.aspx`, January 2006.

[Met97] Rodney Van Meter. Observing the effects of multi-zone disks. In *USENIX Annual Technical Conference*, Monterey, CA, Jan 1997.

[RW94] Chris Ruemmler and John Wilkes. An introduction to disk drive modeling. *IEEE Computer*, 27(3):17–28, March 1994.

[Sea00] Seagate. *Barracuda 18XL Family: ST318436LW/LC/LWV/LCV, ST318426LW/LC, ST318416N/W, ST39236LW/LC/LWV/LCV, ST39226LW/LC, ST39216N/W, Product Manual, Volume 1.* Seagate, 1st edition, December 2000.

[Shra] Elizabeth Shriver. Ph.D. dissertation: Performance modeling for realistic storage devices. Internet WWW-page, URL: `http://www.cs.nyu.edu/~shriver/bell-labs/dissertation.html`. Link is valid at August 2005.

[Shrb] Elizabeth Shriver. Single disk and multiple disk models. Internet WWW-page, URL: `http://www.bell-labs.com/user/shriver/disk-c-model.html`. Link is valid at August 2005.

[Shr97] Elizabeth Shriver. *Performance modeling for realistic storage devices*. PhD Thesis, Department of Computer Science, New York University, New York, 1997.

[Smi85] Alan Jay Smith. Disk cache—miss ratio analysis and design considerations. *ACM Transactions on Computer Systems*, 3(3):161–203, August 1985.

[SMW98] Elizabeth A. M. Shriver, Arif Merchant, and John Wilkes. An analytic behavior model for disk drives with readahead caches and request reordering. In *Measurement and Modeling of Computer Systems*, pages 182–191, 1998.

[SSP+05] Steven W. Schlosser, Jiri Schindler, Stratos Papadomanolakis, Minglong Shao, Anastassia Ailamaki, Christos Faloutsos, and Gregory R. Ganger. On multidimensional data and modern disks. In *Proceedings of the 4th USENIX Conference on File and Storage Technology (FAST '05)*, pages 225–238, December 2005.

[ST] Sonnet Technologies. eSATA benchmarks. Internet WWW-page, URL: `http://www.sonnettech.com/publicfiles/pdfs/eSATA_benchmarks.pdf`. Link is valid at September 2005.

[Sta03] William Stallings. *Computer Organization and Architecture. Designing for Performance.* Prentice Hall, Upper Saddle River, New Jersey, sixth edition, 2003.

[TCG02] Peter Triantafillou, Stavros Christodoulakis, and Costas Georgiadis. A comprehensive analytical performance model for disk devices under random workloads. *Knowledge and Data Engineering*, 14(1):140–155, 2002.

[TL02] Alexander Thomasian and Chang Liu. Some new disk scheduling policies and their performance. In *SIGMETRICS*, pages 266–267. ACM, 2002.

[Var00] Elizabeth Varki. A performance model of disk array storage systems. In *Int. CMG Conference*, pages 635–644, 2000.

[WAA+04] Mengzhi Wang, Kinman Au, Anastassia Ailamaki, Anthony Brockwell, Christos Faloutsos, and Gregory R.Ganger. *Storage Device Performance Prediction with CART Models.* Technical Report CMU-PDL-04-103, Parallel Data Laboratory, Carnegie Mellon University, 2004.

[Wil95] J. Wilkes. *The Pantheon storage-system simulator*. Technical Report HPL–SSP–95–14. Storage Systems Program, Hewlett-Packard Laboratories, Palo Alto, CA, 1995.

[ZH03] Yingwu Zhu and Yiming Hu. Disk built-in caches: Evaluation on system performance. In *Modeling, Analysis and Simulation of Computer Telecommunications Systems, 2003. MASCOTS 2003. 11th IEEE/ACM International Symposium*, pages 306–313, 2003.