Chapter 1

Introduction

1.1. Why I/O Performance?

In the last decade, innovations in technology have led to extraordinary advances in computer processing speed. These advances have led many of those who evaluate a computer's performance to focus their attention on measuring processor performance to the near exclusion of all other metrics; some have even equated a computer system's performance with how well its CPU performs. This viewpoint, which makes system-wide performance synonymous with CPU speed, is becoming less and less valid. One way to demonstrate this declining validity is illustrated in Figure 1.1, where IBM disk performances represented by the throughput of accessing a random 8 KB block of data, and IBM mainframe CPU performance [Patterson90]. For the sake of comparison, both CPU and disk performance are normalized to their 1971 levels. As can

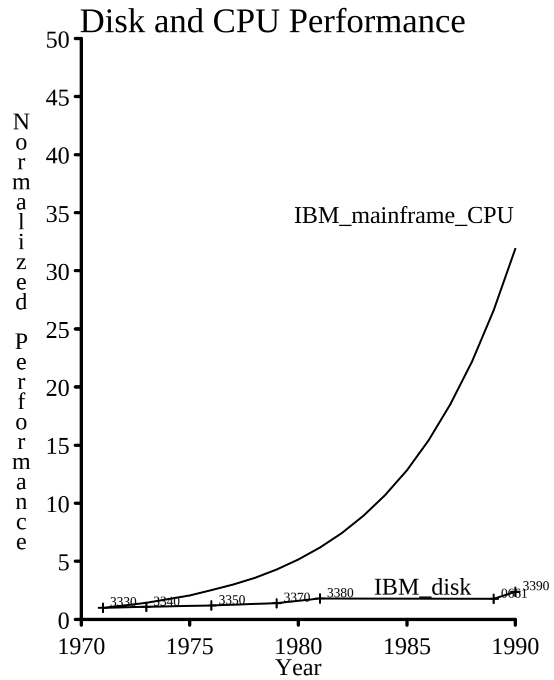

Figure 1.1: Contrasting trends of CPU and disk performance improvements. Over the past two decades, improvements in CPU performance have far outstripped those in disk performance. In this graph, CPU performance refers to IBM mainframe performance; disk performance refers to IBM disk (33x0 series) throughput on a random 8 KB access. Both are normalized to their 1971 levels. The data for IBM mainframe performance comes from Figure 1.1 on page 4 of [Patterson90].

readily be seen, over the past two decades, IBM mainframe CPU performance has increased more than 30-fold, while IBM disk performance has barely doubled. Microprocessor performance has increased even faster than mainframe performance [Myers86, Gelsinger89]. If CPU performance continues to improve at its current pace and disk performance continues to obtain more moderate improvements, eventually the performance of all applications that do any input or output (I/O) will be limited by that I/O component—further CPU performance improvements will be wasted [Amdahl67].

In light of this developing trend toward I/O-limited applications, I/O performance and architecture become increasingly more crucial to a system's overall performance. Researchers are rapidly developing new types of I/O architectures to match the improvements in processor performance; I contend that these new I/O architectures must be evaluated using new benchmarking techniques. In this dissertation, I propose a new approach to I/O benchmarks, which I've called *self-scaling evaluation and predicted performance,* that scales more comprehensively, provides more understanding of the system being measured, and estimates performance to within 10-15% over a wide range of I/O workloads and systems.

1.2. Overview of the Dissertation

The rest of this chapter presents background material on I/O performance evaluation. I discuss common metrics that researchers use to evaluate I/O systems. After this, I highlight current trends in I/O systems and how these trends affect I/O performance evaluation.

In Chapter 2, I survey and evaluate current I/O benchmarks, testing them on three Unix workstations. After doing so, I list desirable characteristics for I/O benchmarks and use these to critique current I/O benchmarks. I also summarize other research done in the modeling of I/O and file system performance.

In Chapter 3, I describe in detail the workload model that provides a framework for my new benchmarking approach. I discuss the process of developing the workload model and some alternatives. I then trace real applications and describe their I/O workload.

In Chapter 4, I propose the first part of my new approach to I/O evaluation—a *self-scaling benchmark.* This benchmark scales the workload used to evaluate a system based on that system's capabilities. I show that benchmarks must scale to be useful and that this method scales more effectively than any current benchmarking method. I also show how the self-scaling benchmark provides insight into a computer system by giving information on what workload might be appropriate to run on each system. I demonstrate the utility of the self-scaling benchmark by showing benchmark results gathered on a wide variety of I/O systems, ranging from a one disk, low-end workstation to a four disk mini-supercomputer.

In Chapter 5, I propose the second part of my new approach—*predicted performance*, which uses measurements from a small set of workloads to accurately estimate performance for arbitrary workloads. I show that predicted performance estimates performance very accurately, within 10-15%, on a wide range of I/O systems and investigate how error is correlated to each workload parameter. I compare my method of predicting performance against using an orthogonal sampling of many workloads and show how my method gives lower error while using fewer workload measurements. I end by applying predicted performance to predict the relative performance ratio between two systems.

In Chapter 6, I conclude with a summary of the contributions made by my thesis and I look at some future directions for this research.

1.3. Metrics

More than other areas of computer performance evaluation, I/O evaluation involves a great variety of metrics. This section contains an overview of some of the metrics commonly used today in choosing and evaluating I/O systems. The value of most metrics depend strongly on the workload used.

The most basic metric for I/O performance is *throughput*. Throughput is a measure of speed—the rate at which an I/O system delivers data, and it is measured in two ways: I/O rate, measured in *accesses/second*, and data rate, measured in *bytes/second* or *megabytes/second (MB/s)*. I/O rate is commonly used for applications where the size of each request is small, such as transaction processing [Anon85]; data rate is commonly used for applications where the size of each request is large, such as scientific applications [Miller91a].

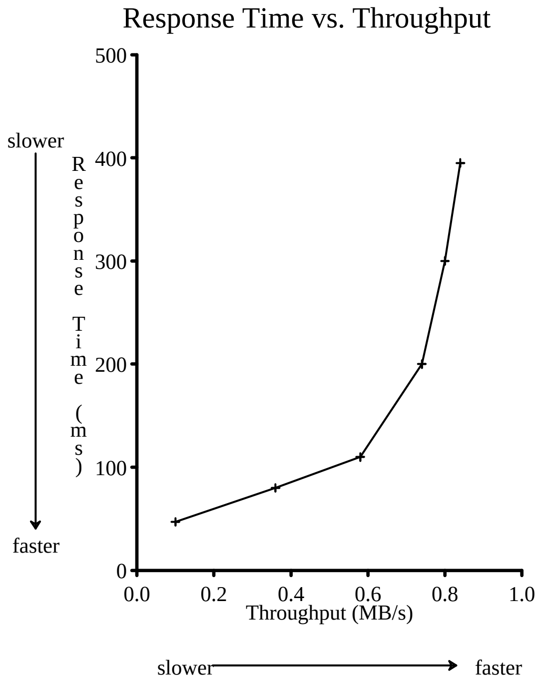

Figure 1.2: Example response time versus throughput graph. Increasing the utilization of a system usually leads to higher throughput but slower response time. This figure was adapted from [Chen90b].

Response time is the second basic performance metric for I/O systems, and it measures how long an I/O system takes to access data. This time can be measured in several ways. For example, one could measure I/O response time from the user's perspective, the operating system's perspective, or the disk controller's perspective, depending on what is considered the I/O system.

The usefulness of a I/O system not only includes *how fast* data can be accessed but also *how much* data can be stored. *Capacity* is not normally applied as a metric to non-storage components of a computer system, but it is an integral part of evaluating an I/O system. If capacity were ignored as a metric, tape and disk manufacturers would soon find their customers switching to solid-state (memory-based) storage systems, which offer much higher performance but less capacity per dollar.

Because users store valuable data on I/O systems, they demand a level of reliability much higher than for other parts of a computer. If a memory chip develops a parity error, for example, the system will (hopefully) crash and be restarted. On the other hand, if a storage device develops a parity error in a database of bank accounts, banks could unwittingly lose millions of dollars. Thus, *reliability* is a metric uniquely important to storage systems.

Cost, of course, applies to all components in computer systems, particularly to disk subsystems, which are often the most expensive component in a large computer installation [Bodega89]. Cost is usually expressed as a composite metric, such as cost per capacity (dollars per MB) or throughput cost (dollars per MB/s).

Table 1.1, adapted from [Gibson91], shows the values of the above metrics for two different disk systems.

These five metrics— throughput, response time, capacity, reliability, and cost—are commonly used in various combinations to evaluation I/O systems. One popular combination is a response time versus throughput graph (Figure 1.2). These graphs vary a parameter, such as the number of users on the system, to display the tradeoff between improving throughput and degrading response time. With more users, the system can often be utilized more efficiently, which increases throughput. On the other hand, higher utilization leads to slower response times. Because a single performance number is easier to use than a full graph, many evaluators combine throughput and response time by reporting throughput at a given response time [Anon85, Chen90b]. The TPC-B benchmark, for example, reports maximum throughput with 90% of all requests completed within 2 seconds [TPCB90] (see Section 2.4.1.2).

Another composite metric is *data temperature*, which is defined as I/O rate divided by capacity [Katz90]. Data temperature measures how many I/Os per second a I/O system can support for a fixed amount of storage. This is a valuable metric for users who are limited by I/O rate rather than capacity because it tells them that they should buy systems with high data temperature.

A general parameterizable composite metric can be formulated for any combination of the above metrics. For example, one could imagine a system administrator who wanted a system with the highest capacity per dollar, as long as it satisfied minimum demands for reliability, throughput, and response time.

1.4. Trends in I/O Systems

To understand how developments in I/O systems are straining the capabilities of current I/O benchmarks, I highlight some of the trends in I/O systems in this section. I discuss advances in magnetic disk technology, arrays of disks, file caching and solid state disks, magnetic tape, and log-structured file systems.

Magnetic disks have long been the mainstay of I/O systems, but since 1970, disk performance has improved only modestly. Table 1.2 compares two disks, the IBM 3330, introduced in 1971, and the IBM 0661, introduced in 1989. The average yearly improvement in performance has inched forward at a few percent a year. Cost per capacity, on the other hand, has improved at a much faster pace, averaging a 23% reduction per year from 1977 to 1986 [Gibson91]. Moreover, individual disks have also been gradually decreasing in physical size and

Type of Metric	Specific Measure	IBM 3390	Redundant Disk Array of IBM 0661 disks
Throughput	Max Read I/O Rate	609 I/O’s per second	3889 I/O’s per second
Throughput	Max Read Data Rate	15 MB per second	130 MB per second
Response Time	Min Response Time	20 ms	20 ms
Capacity	GB	23 GB	22 GB
Reliability	Mean Time to Data Loss	6-28 years	753 years
Cost	$ (estimated)	$156,000 - $260,000	$67,000 - ?

Table 1.1: Metrics for two I/O systems. Above are the differences in the values of several types of metrics for an IBM 3390 disk system and a redundant disk array made of IBM 0661 3.5" drives [IBM0661]. This table is adapted from [Gibson91].

cost. The most common diameter of a disk in the 1970's and 1980's was 14". Those disks are disappearing and are being replaced with 5.25" and 3.5" diameter disks. The performance of these smaller disks is comparable to that of their larger, more expensive predecessors.

The trend toward smaller, less expensive disks makes it possible to combine many of them into a parallel I/O system known as a *disk array*. Arrays of multiple disks have been used for many years for special purposes [Johnson84] but they are only now becoming popular for general use. The list of companies developing or marketing disk arrays is quite long, and it includes Array Technology, Auspex, Ciprico, Compaq, Cray, Datamax, Hewlett-Packard, IBM, Imprimis, Intel Scientific, Intellistor, Maximum Strategy, Pacstor, SF2, Storage Concepts, Storage Technology, and Thinking Machines. Some analysts have projected that the disk array market will expand to $8 billion by 1994 [Montgomery91].

The idea behind disk arrays is straightforward—combine many small disks and distribute data among them (Figure 1.3), which increases the aggregate throughput available to an application. An array of disks can service either many small accesses in parallel or cooperate to deliver a higher data rate to a single, large access [Patterson88, Gibson91, Livny87, Salem86]. Disk arrays compensate for the lower reliability inherent in using more disks by storing redundant, error-correcting information. Current research into disk arrays is focusing on 1) how to distribute (stripe) data across disks to get optimal performance [Chen90a, Lee91a, Lee91b, Weikum90] and 2) how to spread redundant information across disks to increase reliability and

Metric	IBM 3330	IBM 0661	Average Yearly Improvement
Year Introduced	1981	1989	
Average Seek Time	30 ms	12.5 ms	5%
Average Rotational Delay	8.3 ms	7 ms	1%
Transfer Rate	806 KB/s	1700 KB/s	4%

Table 1.2: Magnetic disk performance improvement over the past 20 years. This table shows the slow average improvement in disk performance over the past 20 years. The IBM 3330 was introduced in 1981 and is 14 inches in diameter; the IBM 0661 was introduced in 1989 and is 3.5 inches in diameter [Harker81, IBM0661].

minimize the effects of disk failures [Holland92, Gibson91, Muntz90].

Although disk arrays improve throughput by using more disks to service requests, requests that are serviced by a single disk still see the same response time. Techniques for improving response time include file caches, disk caches, and solid state disks, all of which use dynamic RAM (random access memory). Caches can be located in a variety of places in a system's memory hierarchy [Smith85]. Two common places are the disk controller, as in the IBM 3990 disk cache [Menon87], and main memory, as in the Sprite operating system's file cache [Ousterhout88, Nelson88]. Response times for writes are decreased by writing the data to

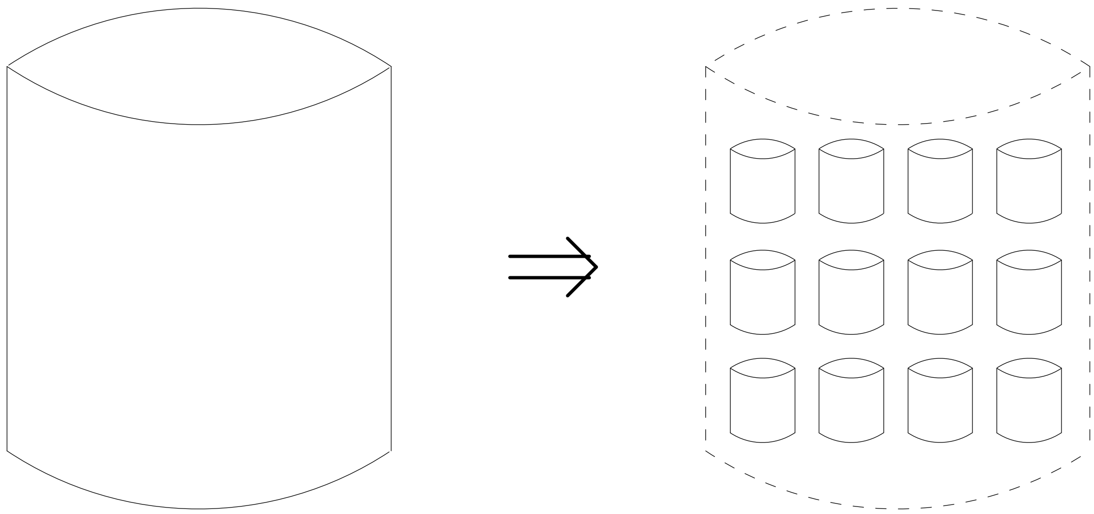

Figure 1.3: Combining multiple smaller disks to improve performance. Performance of single disks is not improving rapidly (Table 1.2); however, disks are rapidly becoming physically smaller and cheaper. Disk arrays take advantage of this downsizing to provide higher aggregate throughput by simultaneously operating many small disks.

RAM, acknowledging the request, then transferring the data to disk asynchronously. However, this technique, called *write-behind*, leaves the data in RAM more vulnerable to system failures until it is written to disk. Some systems, such as the IBM 3990, mitigate this problem of vulnerability by storing the cached data in non-volatile memory, which is immune to power failures [Menon87]. As with any cache, read response time is decreased if the requested data is found in cache RAM.

Solid state disks are similar to caches in that they improve response time by storing requests in RAM rather than on magnetic disks. The principal difference between solid state disks and caches is that solid state disks speed up all accesses while caches speed up access only to the most commonly requested data. Although solid state disks costs 50-100 times more per capacity than magnetic disks [Gibson91], they are dramatically faster. Response times for solid state disks are commonly less than 3 ms [Cassidy89, Jones89], while response times for magnetic disks are approximately 10-30 ms.

Two I/O metrics have been addressed, throughput and response time. Dramatic improvements to capacity per cost have occurred in magnetic tapes (Table 1.3). These improvements are due in part to ever-increasing bit densities on tapes and partly to the acceptance of helical scan technology. Helical scan is a method of writing and reading tapes which can increase the capacity of a single tape from 0.1-0.2 GB to 5-20 GB [Katz91, Tan89, Vermeulen89]. Tapes

Device	Total Capacity	Cost/Capacity	Media Capacity	Latency
Magnetic Disk	1 GB	$2,500/GB	1 GB	0.01 sec.
Dilog DAT Stacker	10 GB	$527/GB	1.3 GB	75 sec.
Exabyte 120 Tape Library	500 GB	$80/GB	5 GB	100 sec.
Metrum RSS-600 Tape Library	8700 GB	$62/GB	14.5 GB	50 sec.

Table 1.3: 1992 storage capacity. This table, adapted from [Fine92], shows the extraordinary capacity of today's tape systems.

are extremely slow, however, with response times ranging from 20 seconds to a few minutes; throughput for these devices is less dismaying, ranging from 0.1 to 2.0 MB/s. Current research related to tape devices addresses how to migrate data from tape to faster storage [Smith81, Thanhardt88, Hac89, Miller91b, Henderson89], how to increase tape throughput using striping [Katz91], and how to decrease response times by prefetching and caching [Gibson92, Fine92].

Reported disk reliability has improved dramatically over the past ten years, though actual reliability has improved more slowly. The most common metric used to gauge reliability, *mean-time-to-failure*, has increased from 30,000 hours to 150,000-200,000 hours. This jump in apparent reliability comes mostly from changing the method by which mean-time-to-failure is computed and is not expected to continue improving as quickly [Gibson91].

I/O innovation is also taking place in file systems. A good example of how file systems have improved I/O system performance is the Log-Structured File System (LFS) [Rosenblum91, Ousterhout89], which allocates data on disk in the same order that it is written. This leads to highly sequentialized disk writes and so improves the sustainable disk write throughput.

Although the raw performance of I/O technology has improved much more slowly than processor technology, innovation such as file caches, disk arrays, robot-driven tape systems, and new file systems have helped close the gap. At the same time, these innovations have created new challenges for I/O benchmarks. For instance, solid state disks, file caches, and disk caches all use dynamic RAM (DRAM), whose capacity has been quadrupling every three years [Myers86]. Due to this rapid growth, benchmarks that had, at one time, exercised the disk system will no longer do so.

1.5. References

[Amdahl67] G. M. Amdahl, ‘‘Validity of the single processor appraoch to achieving large scale computing capabilities’’, *Proceedings AFIPS 1967 Spring Joint Computer Conference 30* (April 1967), 483-485.

[Anon85] Anon and *et al.*, ‘‘A Measure of Transaction Processing Power’’, *Datamation, 31*, 7 (April 1, 1985), 112-118.

[Bodega89] *National Science Foundation Workshop on Next Generation Secondary Storage Architecture*, National Science Foundation, Bodega Bay, CA, May 1989.

[Cassidy89] C. Cassidy, ‘‘DEC’s ESE20 Boosts Performance’’, *DEC Professional*, May 1989, 102-110.

[Chen90a] P. M. Chen and D. A. Patterson, ‘‘Maximizing Performance in a Striped Disk Array’’, *Proceedings of the 1990 ACM SIGARCH Conference on Computer Architecture*, Seattle WA, May 1990, 322-331.

[Chen90b] P. M. Chen, G. Gibson, R. H. Katz and D. A. Patterson, ‘‘An Evaluation of Redundant Arrays of Disks Using an Amdahl 5890’’, *Proceedings of the 1990 ACM SIGMETRICS Conference on Measurement and Modeling of Computer Systems,* Boulder CO, May 1990.

[Fine92] J. A. Fine, T. E. Anderson, M. D. Dahlin, J. Frew, M. Olson and D. A. Patterson, ‘‘Abstracts: A Latency-Hiding Technique for High-Capacity Mass Storage Systems’’, Sequoia Technical Report 92/11, University of California at Berkeley, March 1992.

[Gelsinger89] P. P. Gelsinger, P. A. Gargini, G. H. Parker and A. Y. C. Yu, ‘‘Microprocessors Circa 2000’’, *IEEE Spectrum*, October 1989, 43-47.

[Gibson91] G. A. Gibson, ‘‘Redundant Disk Arrays: Reliable, Parallel Secondary Storage’’, UCB/Computer Science Dpt. 91/613, University of California at Berkeley, December 1991. also available from MIT Press, 1992.

[Gibson92] G. A. Gibson, R. H. Patterson and M. Satyanarayanan, ‘‘Disk Reads with DRAM Latency’’, *Third Workshop on Workstaion Operating Systems*, Key Biscayne, Florida, April 23-24, 1992.

[Hac89] A. Hac, ‘‘A Distributed Algorithm for Performance Improvement Through File Replication, File Migration, and Process Migration’’, *IEEE Transactions on Software Engineering 15*, 11 (November 1989), 1459-1470.

[Harker81] J. M. Harker, D. W. Brede, R. E. Pattison, G. R. Santana and L. G. Taft, ‘‘A Quarter Century of Disk File Innovation’’, *IBM Journal of Research and Development 25*, 5 (September 1981), 677-689.

[Henderson89] R. L. Henderson and A. Poston, ‘‘MSS II and RASH: A Mainframe UNIX Based Mass Storage System with a Rapid Access Storage Hierarchy File Management System’’, *Winter USENIX 1989*, January 1989, 65-83.

[Holland92] M. Holland and G. Gibson, ‘‘Parity Declustering for Continuous Operation in Redundant Disk Arrays’’, *Proceedings of the 5th International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS-V)*, October 12-15, 1992, 23-35.

[IBM0661] *IBM 0661 Disk Drive Product Description--Model 371*, IBM, July 11, 1989.

[Johnson84] O. G. Johnson, ‘‘Three-Dimensional Wave Equation Computations on Vector Computers’’, *Proceedings of the IEEE 72*, 1 (January 1984).

[Jones89] A. L. Jones, *SSD is Cheaper than DASD*, Storage Technology Corporation, October 1989.

[Katz90] R. H. Katz, D. W. Gordon and J. A. Tuttle, ‘‘Storage System Metrics for Evaluating Disk Array Organizations’’, UCB/Computer Science Dpt. 90/611, University of California at Berkeley, December 1990.

[Katz91] R. H. Katz, T. E. Anderson, J. K. Ousterhout and D. A. Patterson, ‘‘Robo-line Storage: Low Latency, High Capacity Storage Systems over Geographcially Distributed Networks’’, UCB/Computer Science Dpt. 91/651, University of California at Berkeley, September 1991.

[Kim86] M. Y. Kim, ‘‘Synchronized Disk Interleaving’’, *IEEE Transactions on Computers C-35*, 11 (November 1986), 978-988.

[Lee91a] E. K. Lee and R. H. Katz, ‘‘An Analytic Performance Model of Disk Arrays and its Applications’’, UCB/Computer Science Dpt. 91/660, University of California at Berkeley, 1991.

[Lee91b] E. K. Lee and R. H. Katz, ‘‘Performance Consequences of Parity Placement in Disk Arrays’’, *Proceedings of the 4rd International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS-IV),* April 1991, 190-199.

[Livny87] M. Livny, S. Khoshafian and H. Boral, ‘‘Multi-Disk Management Algorithms’’, *Performance Evaluation Review, Special Issue 15*, 1 (May 1987), 69-77. ACM SIGMETRICS 1987.

[Menon87] J. Menon and M. Hartung, ‘‘The IBM 3990 Model 3 Disk Cache’’, RJ 5994 (59593), IBM, December 9, 1987.

[Metrum91] *RSS-600 Rotary Storage System Product Information*, Metrum, 1991.

[Miller91a] E. L. Miller and R. H. Katz, ‘‘Input/Output Behavior of Supercomputing Applications’’, *Proceedings of Supercomputing ’91*, 1991, 567-576.

[Miller91b] E. L. Miller, ‘‘File Migration on the Cray Y-MP at the National Center for Atmospheric Research’’, UCB/Computer Science Dpt. 91/638, University of California at Berkeley, June 1991.

[Montgomery91] J. B. Jones(Jr.) and T. Liu, editors. ‘‘RAID: A Technology Poised for Explosive Growth’’, Report DJIA: 2902, Montgomery Securities, December 17, 1991.

[Muntz90] R. R. Muntz and J. C. S. Lui, ‘‘Performance Analysis of Disk Arrays under Failure’’, *Proceedings of the 16th Conference on Very Large Data Bases*, 1990. VLDB XVI.

[Myers86] G. J. Myers, A. Y. C. Yu and D. L. House, ‘‘Microprocessor Technology Trends’’, *Proceedings of the IEEE 74*, 12 (December 1986), 1605-1622.

[Nelson88] M. N. Nelson, B. B. Welch and J. K. Ousterhout, ‘‘Caching in the Sprite Network File System’’, *ACM Transactions on Computer Systems 6*, 1 (February 1988), 134-154.

[Ousterhout88] J. K. Ousterhout, A. Cherenson, F. Douglis and M. Nelson, ‘‘The Sprite Network Operating System’’, *IEEE Computer 21*, 2 (February 1988), 23-36.

[Ousterhout89] J. K. Ousterhout and F. Douglis, ‘‘Beating the I/O Bottleneck: A Case for Log-Structured File Systems’’, *SIGOPS 23*, 1 (January 1989), 11-28.

[Patterson88] D. A. Patterson, G. Gibson and R. H. Katz, ‘‘A Case for Redundant Arrays of Inexpensive Disks (RAID)’’, *International Conference on Management of Data (SIGMOD)*, June 1988, 109-116.

[Patterson90] D. A. Patterson and J. L. Hennessy, *Computer Architecture: A Quantitative Approach*, Morgan Kaufmann Publishers, Inc., San Mateo, CA, 1990.

[Rosenblum91] M. Rosenblum and J. K. Ousterhout, ‘‘The Design and Implementation of a Log-Structured File System’’, *Proceedings of the 13th ACM Symposium on Operating Systems Principles*, October 1991.

[Salem86] K. Salem and H. Garcia-Molina, ‘‘Disk Striping’’, *Proceedings of the Second International Conference on Data Engineering*, 1986, 336-342.

[Smith81] A. J. Smith, ‘‘Optimization of I/O Systems by Cache Disk and File Migration: A Summary’’, *Performance Evaluation 1*, 3 (November 1981), 249-262.

[Smith85] A. J. Smith, ‘‘Disk Cache-Miss Ratio Analysis and Design Considerations’’, *ACM Transactions on Computer Systems 3*, 3 (August 1985), 161-203.

[TPCB90] *TPC Benchmark B Standard Specification*, Transaction Processing Performance Council, August. 23, 1990.

[Tan89] E. Tan and B. Vermeulen, ‘‘Digital audio tape for data storage’’, *IEEE Spectrum*, October 1989, 34-38.

[Thanhardt88] E. Thanhardt and G. Harano, ‘‘File Migration in the NCAR Mass Storage System’’, *Proceedings of the Ninth IEEE Symposium on Mass Storage Systems*, October 1988.

[Vermeulen89] B. Vermeulen, ‘‘Helical Scan and DAT--a Revolution in Computer Tape Technology’’, *Systems Design and Networks Conference (SDNC)*, May 1989, 79-86.

[Weikum90] G. Weikum, P. Zabback and P. Scheuermann, *Dynamic File Allocation in Disk Arrays*, ETH Zurich, December 1990.