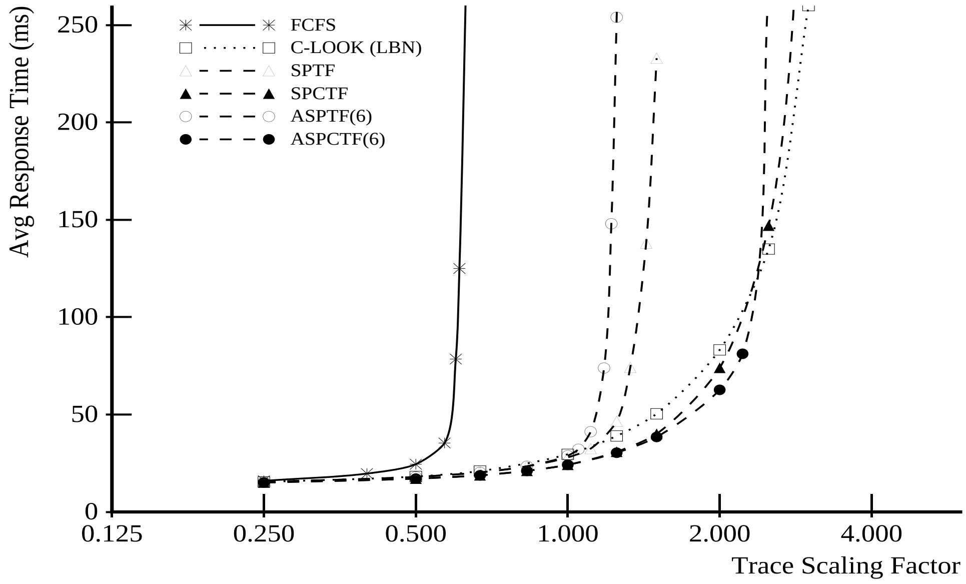
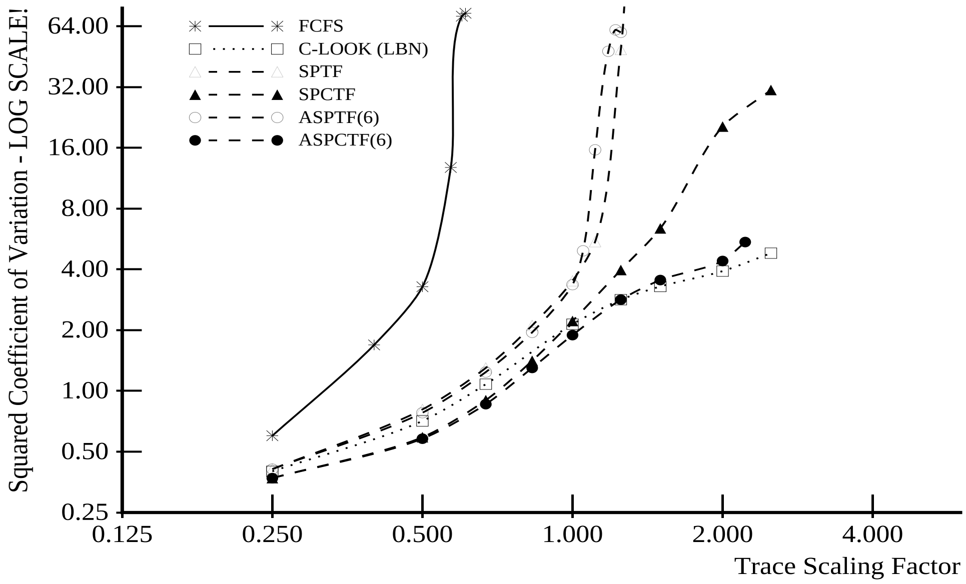
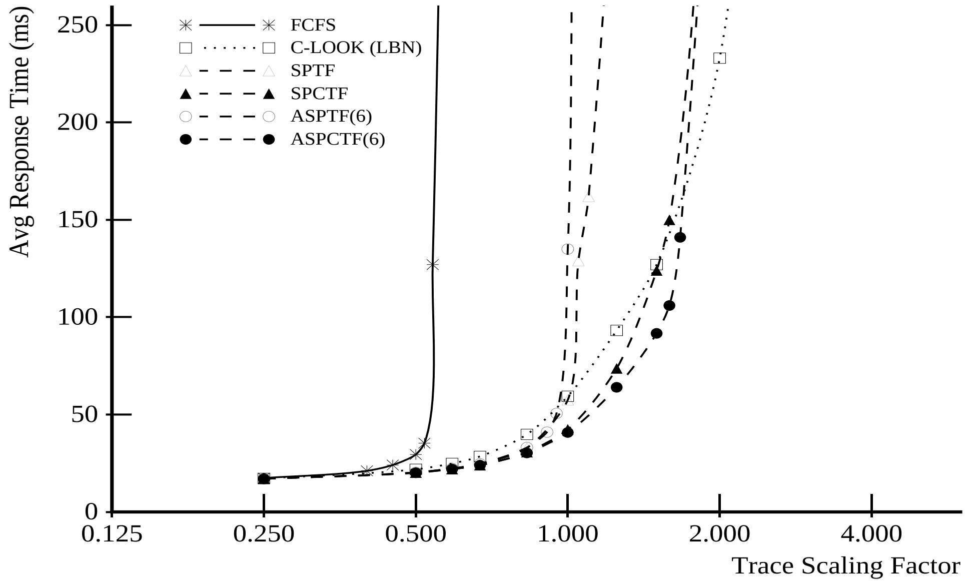


(a) Average Response Time

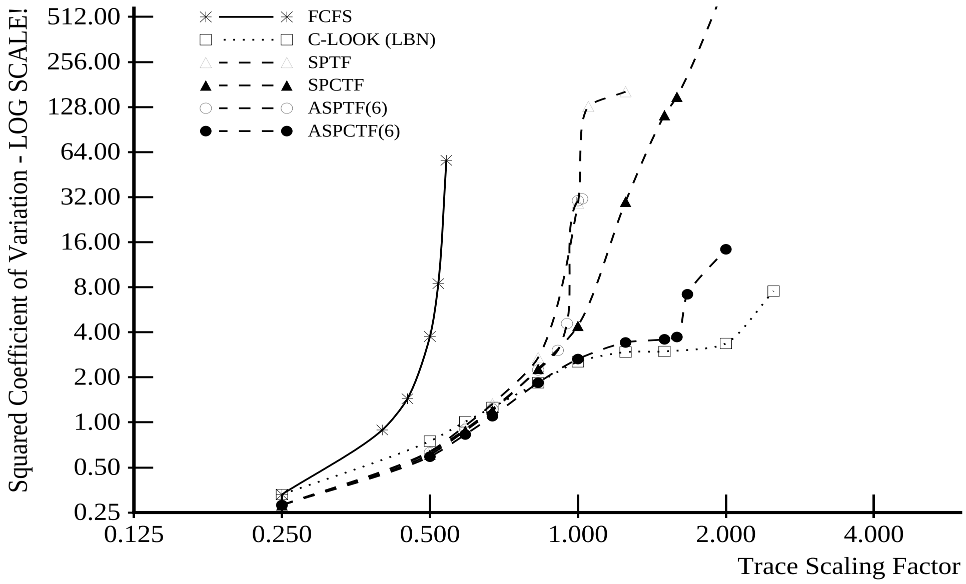


(b) Squared Coefficient of Variation

Figure 22: Full-Knowledge Algorithm Performance using the *Order* Trace



(a) Average Response Time



(b) Squared Coefficient of Variation

Figure 23: Full-Knowledge Algorithm Performance using the *Report* Trace

The SPTF-based algorithms have both a significant advantage and disadvantage over C-LOOK. The advantage is the obvious improvement in positioning delay prediction accuracy due to the inclusion of rotational latency information. A full rotation for an HP C2240 series disk takes approximately 11 ms, equivalent to a seek of over 550 cylinders or 27% of the maximum seek distance. For non-uniform workloads (with shorter average seek distances), the rotational latency delay represents an even larger fraction of the total positioning delay. The disadvantage is that these algorithms have no inherent bias towards servicing sequential reads in ascending order. Pending requests are selected based on their predicted positioning delay, not their LBN or cylinder. Even when cache knowledge is incorporated, only subsets of a sequential stream may be serviced "in order." As a result, the prefetching cache is less effectively utilized by SPTF-based algorithms. Table 5 lists the cache hit percentages for some sample points. Note that C-LOOK always exhibits a higher cache hit rate, as all sequential streams are processed in a logically ascending manner.

Trace	Scale Factor	C-LOOK (LBN)	SPCTF	ASPCTF(6)	SPTF	ASPTF(6)
Cello	1.0	11.5%	11.5%	11.5%	11.2%	11.2%
Snake	1.25	21.8%	21.2%	21.2%	17.4%	17.4%
Air-Rsv	2.5	15.0%	14.2%	14.1%	13.2%	13.1%
Sci-TS	2.5	26.9%	25.4%	25.3%	23.0%	22.5%
Order	1.0	17.6%	16.2%	16.0%	11.0%	10.9%
Report	1.0	14.5%	11.7%	11.6%	10.2%	9.6%

Table 5: On-board Disk Cache Hit Rates for SPTF-based Algorithms

This trade-off affects overall performance to different degrees for each of the six traces. For *Cello*, the SPTF-based algorithms are clearly superior to the seek-reducing algorithms studied in this report. Cache knowledge (i.e., SPCTF and ASPCTF(6)) increases performance an additional 5-15% under the largest scaling factors. In experiments using the other five traces, the comparison is not as clean. For these traces, SPTF and ASPTF(6) saturate more quickly than C-LOOK. However, they do provide superior performance for some sub-saturation workloads. For example, figure 23a shows that for *Report*, SPTF and ASPTF(6) are significantly (up to 18%) better than C-LOOK in the 0.4-0.9 range of scaling factors. For SPCTF and ASPCTF(6), saturation does not occur as quickly. Incorporating cache knowledge widens the window where SPTF-based algorithms are superior to C-LOOK. In the case of *Sci-TS*, both algorithms provide increased performance for all scaling factors studied.

Although ASPCTF(6) always has better starvation resistance than SPCTF, the relative performance (as measured by average response times) of these two algorithms is dependent on the individual workload. In some cases, the aging factor can improve both starvation resistance and performance by maintaining the ordering of sequential requests, provided they were issued "in order" over a relatively short period of time.

We believe that a modified version of ASPCTF which specifically schedules sequential requests in ascending logical order will increase performance beyond C-LOOK in all cases. This represents another item for future research.

7 Conclusions and Future Work

Modern disk drives have several features which can affect the ability of disk scheduling algorithms to produce good schedules. Using strongly-validated simulation, we have re-examined popular scheduling algorithms to determine the importance of some of these features. Synthetic workloads were used to allow comparisons with previous work. Extensive traces gathered from six different environments were used to represent more "realistic" workloads.

Features such as zoned recording, track/cylinder skew, and defect reallocation serve to complicate the translation of logical block numbers into physical media locations. Although full mapping knowledge is required for the most complex scheduling algorithms (e.g., SPTF), our results indicate that it produces little or no performance improvement (less than 2%) for seek-reducing algorithms. We have shown that an appropriate LBN-based algorithm can be chosen for each of the six production workloads examined.

C-LOOK, an algorithm which always schedules requests in logically ascending order, best exploits the prefetching cache for workloads with significant read sequentiality. For random workloads, C-LOOK has been shown (both in this work and in previous studies) to provide slightly inferior performance to other seek-reducing algorithms (e.g., SSTF and LOOK). For five of the six real-world traces, however, C-LOOK achieves the highest cache hit rates and lowest average response times. In addition, the LBN-based C-LOOK algorithm is straightforward and relatively simple to implement.

More powerful disk controllers can use variants of the Shortest Positioning Time First (SPTF) algorithm to achieve even higher performance. The use of such algorithms requires thorough knowledge of the disk's current state as well as the management schemes employed by the disk drive firmware. The computational cost (as indicated indirectly by our simulation times) can be very high. Also, our results indicate that it is critical for such an implementation to recognize the existence of on-board disk caches and optimize for them.

We are currently pursuing several avenues of research in scheduling. First, we are validating our results for other disk drives. Although we consider the HP C2247 to be an excellent example of its generation of disks, positioning times continue to decrease, bit densities continue to increase, and on-board processing power and memory continue to grow. There are also disk drive features which we have not yet explored. Most modern disks can maintain a queue of outstanding requests within their embedded controllers. Scheduling algorithms for such on-board queues may differ in nature from those outside the disk drive package. In addition, most drives can signal completion as soon as the on-board disk cache has received the write request data. The use of such *fast-writes* may require new scheduling techniques in order to maintain reliability guarantees and effectively utilize the available cache storage.

Finally, we believe that scheduling decisions should be based on both subsystem characteristics and system-level requirements. By incorporating knowledge of the system performance consequences of different disk requests, scheduling algorithms can be optimized for overall system performance [Gang93]. For example, a request that prevents a critical process from executing should be serviced before a low-priority background request. Such goals must be tempered with effective utilization of the storage resources, resulting in a complicated set of trade-offs which we plan to explore.

References

[Denn67] P. J. Denning, "Effects of scheduling on file memory operations", *AFIPS Spring Joint Computer Conference*, April 1967, pp. 9-21.

[Gang93] G. Ganger, Y. Patt, "The Process-Flow Model: Examining I/O Performance from the System's Point of View", *SIGMETRICS*, 1993, pp. 86-97.

[Geis87] R. Geist, S. Daniel, "A Continuum of Disk Scheduling Algorithms", *ACM Transactions on Computer Systems*, February 1987, pp. 77-92.

[Geis87a] R. Geist, R. Reynolds, E. Pittard, "Disk Scheduling in System V", *Performance Evaluation Review*, May 1987, pp. 59-68.

[HP91] Hewlett-Packard Company, "HP C2247 3.5-inch SCSI-2 Disk Drive, Technical Reference Manual", Draft Edition 1, December 1991.

[HP92] Hewlett-Packard Company, "HP C2240 Series 3.5-inch SCSI-2 Disk Drive, Technical Reference Manual", Part Number 5960-8346, Edition 2, April 1992.

[Jaco91] D. Jacobson, J. Wilkes, "Disk Scheduling Algorithms Based on Rotational Position", Hewlett-Packard Technical Report, HPL-CSP-91-7, Feb. 26, 1991.

[McNu86] B. McNutt, "An Empirical Study of Variations in DASD Volume Activity", *CMG*, 1986, pp. 274-283.

[Mert70] A. G. Merten, "Some quantitative techniques for file organization", Ph.D. Thesis, Technical Report No. 15, U. of Wisconsin Comput. Center, 1970.

[Rama92] K. Ramakrishnan, P. Biswas, R. Karedla, "Analysis of File I/O Traces in Commercial Computing Environments", *ACM SIGMETRICS*, 1992, pp. 78-90.

[Ruem93] C. Ruemmler, J. Wilkes, "UNIX Disk Access Patterns", *Winter USENIX*, 1993.

[Ruem94] C. Ruemmler, J. Wilkes, "Modelling Disks", *IEEE Computer*, March 1994.

[Seam66] P. H. Seaman, R. A. Lind, T. L. Wilson "An analysis of auxiliary-storage activity", *IBM System Journal*, Vol. 5, No. 3, 1966, pp. 158-170.

[Selt90] M. Seltzer, P. Chen, J. Ousterhout, "Disk Scheduling Revisited", *Winter USENIX*, 1990, pp. 313-324.

[Teor72] T. Teorey, T. Pinkerton, "A Comparative Analysis of Disk Scheduling Policies", *Communications of the ACM*, March 1972, pp. 177-184.

A Simulation Parameters

The following parameters were used to configure our simulator to model the HP C2247 disk drive. They were obtained from the manufacturer's specifications and from direct measurements.

Table A.1 contains both fixed and variable parameters used in configuring our C2247 disk model. Note that we use an NCR 3550 multiprocessor as the host system, and some of the values given contain combinations of disk and host delays. For example, the data transfer rates given in Table A.1 were limited by the host system rather than the C2247.

Table A.2 specifies the layout of the C2247's 8 zones. Most of this information is available in the disk drive specifications. All skew values are given in sectors. The Send/Receive Diagnostic SCSI commands were used to obtain full logical-to-physical mappings for each disk. This provided us with the the first logical sector in each zone (given in the table) as well as verifying other specifications. Note that the first 17 tracks are reserved for internal use by the disk, and all spare tracks are placed at the end of each zone. The C2247 uses track-based sparing for both slipped and dynamically reallocated defects.

Table A.3 presents the head switch, seek, and settling delays observed for the C2247. The resulting seek curve is given in figure A.1. These values were obtained via extensive repetition of predetermined seek distances and appropriate monitoring of SCSI activity using a logic analyzer. The seek values continue to be refined as additional experiments are performed. Seek distances greater than 10 cylinders are approximated by a two-piece curve. We cannot determine the exact overlap between command overhead and seek initiation, but separate command overhead and seek values have been determined which together account for the observed behavior.

Table A.4 lists various control and communication overheads measured from a C2247. In most cases, these values are sensitive to the activity of both the current request and the immediately previous request. Also, the cache contents affect the observed delays. In the case of two non-sequential write requests, the second write incurs an additional disconnect (immediately after receiving the command) and has different overhead delays than a write following a read request or a sequential write. Most of the "preparation" delays are caused by the host system rather than the disk drive.

After obtaining the above parameter values for the C2247, our model still deviated by a small amount from the observed behavior. By adding additional write overhead delays, we were able to significantly reduce this deviation. Although we cannot attribute these delays to any documented phenomena, the final model closely matches the observed activity of the HP C2247 disk drive. Table A.5 lists the additional overhead delays added to our model. In the last two table entries, the delay is dependent on the request size (B).

Figures A.2-A.5 compare the response time distributions for four typical validation runs using our model and the actual experimental system. The workloads consisted of 95% random reads, 95% random writes, 95% sequential reads and 95% sequential writes, respectively. The demerit figures (defined by [Ruem94] as the root mean square horizontal distance between the two distribution curves) are 0.43%, 0.52%, 1.58%, and 0.99% of the average response times for each run, respectively. The largest of these four corresponds to a workload of mostly sequential reads, which are usually satisfied by the prefetching cache and thus have very low average response times. (The root mean square difference between the two distributions was only 0.078 ms, the lowest of all four runs.) The demerit figure never exceeded 1.9% of the average response time for any validation run.

Constants	Value
Rotational Speed	5400 RPM
Cache Size	256 KB
Read Data Transfer	3.01 MB/s
Write Data Transfer	2.74 MB/s
Read Transfer Disconnect	No

Variables	Value
Cache Segments	2
Read Watermark	75 %
Write Watermark	75 %
Prefetch Minimum	0 KB
Prefetch Maximum	64 KB

Table A.1: Parameters for modeling the HP C2247 on the NCR 3550 system.

Zone	First Cyl	Last Cyl	Sectors per Track	First Logical Sector	Track Skew	Cyl Skew	Reserved Tracks	Spare Tracks
1	0	558	96	0	14	32	17	325
2	559	760	92	14	13	30	0	130
3	761	901	88	73	13	29	0	65
4	902	1051	84	80	12	28	0	78
5	1052	1193	80	52	12	27	0	52
6	1194	1519	72	0	11	24	0	104
7	1520	1793	64	32	10	22	0	78
8	1794	2050	56	30	9	19	0	65

Table A.2: Zone specifications for the HP C2247.

Seek Distance (cylinders)	Delay (ms)
0 (Head Switch)	0.89
1	2.69
2	3.48
3	3.61
4	3.78
5	4.07
6	4.45
7	4.32
8	4.45
9	4.62
10	4.79
$10 < D \leq 300$	$3.81 + 0.33\sqrt{D}$
$300 < D$	$7.75 + 0.0059D$
Write Settling	0.65

Table A.3: Seek parameters for modeling the HP C2247.

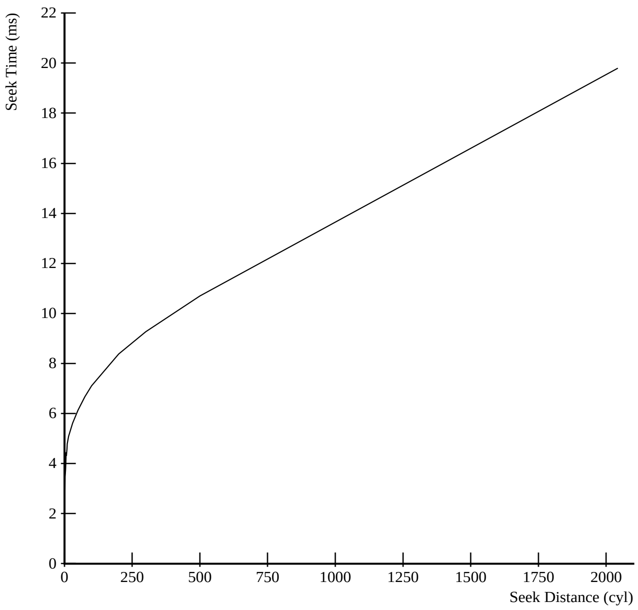


Figure A.1: HP C2247 Disk Drive Seek Curve

Overhead Description	Delay (ms)
Read Command (Hit)	0.953 ms
Read Command (Miss)	0.558 ms
Write Command (Seq or After Read)	0.824 ms
Write Command (Non-Seq Write)	0.642 ms
Read Disconnect Preparation (After Read)	0.023 ms
Read Disconnect Preparation (After Write)	0.046 ms
Write Disconnect Preparation (Seq or After Read)	0.166 ms
Write Disconnect Preparation (Non-Seq Write)	0.046 ms
Write Disconnect Duration (Non-Seq Write)	0.429 ms
Data Phase Preparation	0.025 ms
Read Completion	0.057 ms
Write Completion	0.050 ms
First Reselect	0.162 ms
Second Reselect (Non-Seq Writes Only)	0.129 ms

Table A.4: Overhead delays for modeling the HP C2247 on the NCR 3550 system.

Overhead Description	Value
Reconnect Preparation After Media Write Complete	0.540 ms
Minimum Time Before Media Write (After Command)	0.600 ms
Minimum Blocks Received Before Media Write ($B < 45$)	Min(10,B)
Minimum Blocks Received Before Media Write ($B \geq 45$)	15

Table A.5: Additional write overhead delays for modeling the HP C2247.

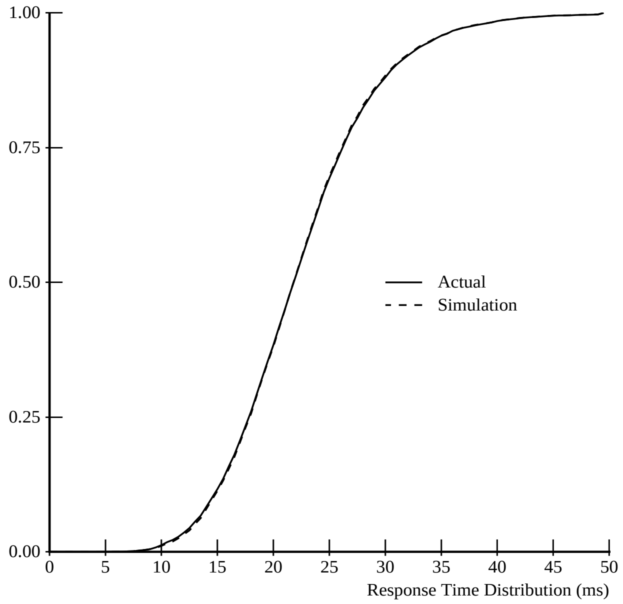


Figure A.2: Validation Workload Response Time Distributions (95% non-sequential reads, 8KB mean request size [exponential], 0-22 ms interarrival [uniform])

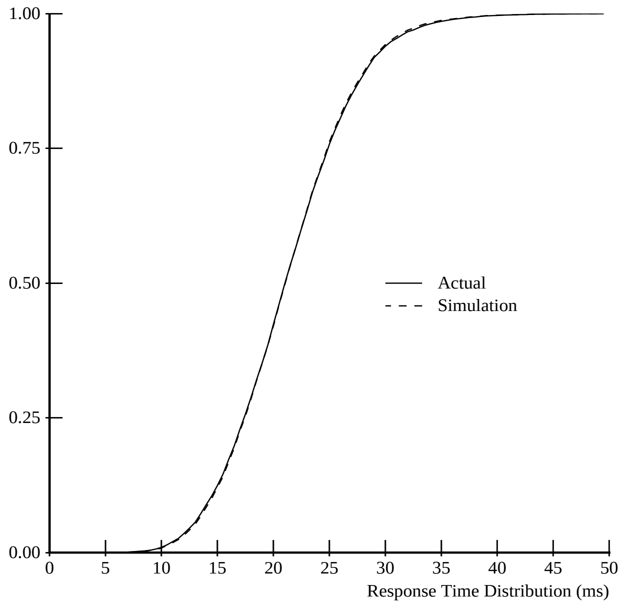


Figure A.3: Validation Workload Response Time Distributions (95% non-sequential writes, 8KB mean request size [exponential], 0-22 ms interarrival [uniform])

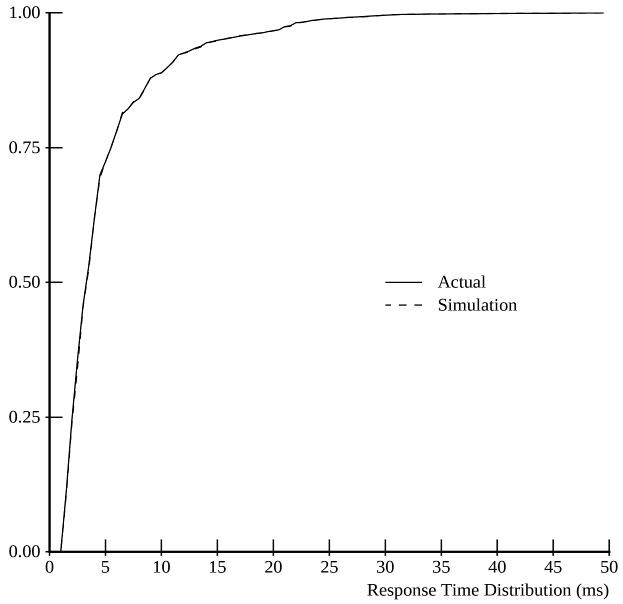


Figure A.4: Validation Workload Response Time Distributions (95% sequential reads, 8KB mean request size [exponential], 0-22 ms interarrival [uniform])

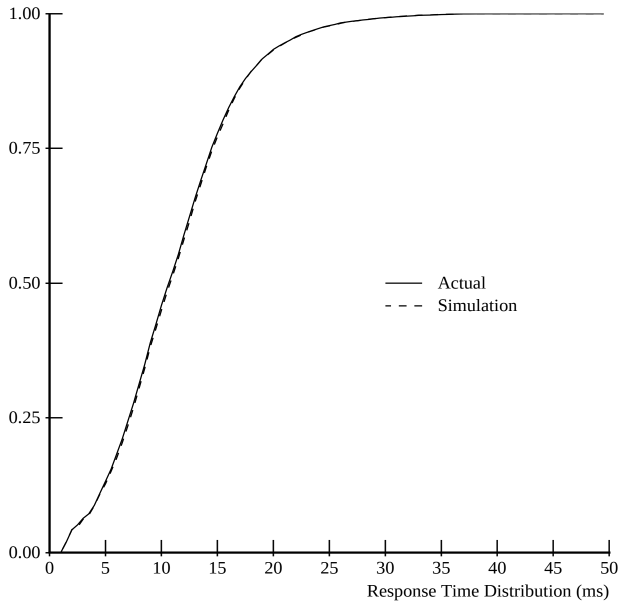


Figure A.5: Validation Workload Response Time Distributions (95% sequential writes, 8KB mean request size [exponential], 0-22 ms interarrival [uniform])