I/O CHARACTERIZATION AND ATTRIBUTE CACHE DATA FOR ELEVEN MEASURED WORKLOADS

Kathy J. Richardson

Technical Report No. CSL-TR-94-656

Dec 1994

Supported by NASA under NAG2-248 and Digital Western Research Laboratory.

I/O CHARACTERIZATION AND ATTRIBUTE CACHE DATA FOR ELEVEN MEASURED WORKLOADS

by

Kathy J. Richardson

Technical Report No. CSL-TR-94-656

Dec 1994

Computer Systems Laboratory

Departments of Electrical Engineering and Computer Science

Stanford University

Stanford, California 94305-4055

Abstract

Workloads generate a variety of disk I/O requests to access file information, execute programs, and perform computation. Workload characterization is crucial to optimizing I/O system performance. This report contains detailed workload characterization data for eleven measured workloads. It includes numerous tables, and cache behavior plots for each workload.

The workload I/O traces, from which the characterization is derived, include both file system information and I/O system information, where previous traces only included one or the other. The additional information allows I/O characterization at the system level, and greatly increases the body of knowledge about the make-up and type of disk I/O requested. The new information shows that the I/O request stream contains statistically diverse components that can be separated. This allows the important features of the workload to be captured at the appropriate cache size, and increases the total cache utilization.

Note: This technical report is a companion report to the dissertation *I/O Characterization and Attribute Caches for Improved I/O System Performance* (CSL-TR-94-655). While the dissertation is self contained, this report is not; it presents data that is analyzed and discussed only in the dissertation.

Key Words and Phrases: I/O cache, I/O architecture, disk cache

Contents

List of Figures

List of Tables

Appendix A

Workload Descriptions

A DECstation 5000 running ULTRIX generated the traces, using a new version of the WRL tracing facilities. The original system was designed primarily to study processor memory issues in a multiprogramming environment [BKLW90, CBJ92]. A modified version of the ULTRIX 4.2 operating system stores trace information in a large buffer. When the buffer becomes sufficiently full, the kernel schedules a special process called the *analysis program* to read and process the buffer contents. The modified system logs system call information, rather than individual instructions, in the trace buffer. On an I/O system call, the call type, process ID, and call parameters are entered in the buffer. A separate entry in the buffer holds the return value, error status and information matching the appropriate call. The kernel traces all processes.

Most of the traces monitor several days of user activity. Such long traces are necessary to capture significant I/O activity and to show the interaction of the many large and small files that comprise a workload. For most workloads, I/O calls occur relatively infrequently; long trace periods are required to capture adequate I/O call events. For user driven workloads, the CPU is idle much of the time waiting for user commands which lengthens the time required for an I/O trace.

The traces cover a wide range of user applications and types of work. All traces were collected within a research laboratory with a broad range of activities. Individual traces cover activities of researchers involved in software and hardware development, as well as writing papers. Although not necessarily typical of heavy commercial use, such as large database systems, the traces should represent many engineering development and office environments. This Appendix describes the eleven traced workloads.

Kernel Build Workload

The Kernel Build trace is a configuration and compile of the ULTRIX kernel for a diskless DECstation 5000, with few extraneous devices [Cop]. Many modules are recompiled to make this kernel.

Workload Duration	119	minutes		
Datafiles				
Datafile Read	18077	Requests	67508	KBytes
Datafile Write	11904	Requests	11998	KBytes
Datafile References	12589	Uses	1079	Unique
Executables				
Executable Reads	778	Requests	93813	KBytes (approx.)
Executable References	778	Uses	52	Unique
Inodes and Directories				
Inode Read	75733	Requests		
Inode Writes	222	File system modifications		
Inode Update	75400	Requests		
Inode Referencs	1327	Unique		
Directory Read	40271	Requests		
Directory Write	129	Requests		
Directory Referencs	131	Unique		
Operations				
12449 Opens	140	Creates	334	Deletes

Ingres Workload

Ingres workload performs 10,000 banking debit and credit transactions on an Ingres database [Sto86]. The transactions are entirely random, and there are no complicated searches. Indexes exist for all of the lookups. The database environment consists of a server and client. Both run on the same machine. The trace includes starting the server and running the 10,000 transactions.

Workload Duration	68	minutes		
Datafiles				
Datafile Read	21012	Requests	49086	KBytes
Datafile Write	32291	Requests	179712	KBytes
Datafile References	3711	Uses	304	Unique
Executables				
Executable Reads	473	Requests	41061	KBytes (approx.)
Executable References	473	Uses	77	Unique
Inodes and Directories				
Inode Read	23000	Requests		
Inode Writes	271	File system modifications		
Inode Update	22535	Requests		
Inode Referencs	565	Unique		
Directory Read	11774	Requests		
Directory Write	193	Requests		
Directory Referencs	76	Unique		
Operations				
3518 Opens	193	Creates	137	Deletes

Application Data Anaysis Workload

Application Data Analysis workload manipulates a series of traces, simulates caches and displays cache results. A filter makes minor modifications to the traces using system utilities such as *fgrep* and *awk*. The cache simulations then use the traces, formatting the results into postscript plots.

Workload Duration	1591	minutes		
Datafiles				
Datafile Read	19428	Requests	97615	KBytes
Datafile Write	41573	Requests	26188	KBytes
Datafile References	16741	Uses	327	Unique
Executables				
Executable Reads	501	Requests	50489	KBytes (approx.)
Executable References	501	Uses	79	Unique
Inodes and Directories				
Inode Read	114689	Requests		
Inode Writes	220	File system modifications		
Inode Update	110899	Requests		
Inode Referencs	985	Unique		
Directory Read	64217	Requests		
Directory Write	95	Requests		
Directory Referencs	128	Unique		
Operations				
16621 Opens	120	Creates	86	Deletes

Document Preparation Workload

Document Preparation records six hours of intense work on a technical report. It includes text processing, editing, simulation, drawing, and data manipulation.

Workload Duration	277	minutes		
Datafiles				
Datafile Read	64947	Requests	383165	KBytes
Datafile Write	34326	Requests	20333	KBytes
Datafile References	17537	Uses	563	Unique
Executables				
Executable Reads	2420	Requests	172594	KBytes (approx.)
Executable References	2420	Uses	69	Unique
Inodes and Directories				
Inode Read	207866	Requests		
Inode Writes	465	File system modifications		
Inode Update	204708	Requests		
Inode Referencs	1168	Unique		
Directory Read	117814	Requests		
Directory Write	315	Requests		
Directory References	144	Unique		
Operations				
17190 Opens	374	Creates	111	Deletes

Software Development 1 Workload

Software Development 1 (8 to 5) develops and tests the ATOM simulation system [SE94]. ATOM is an instrumentation and simulation platform for understanding and debugging performance problems. It selectively instruments applications with an additional compiler pass, and incorporates libraries to count events or simulate performance of the program's execution. Although ATOM uses a standard set of libraries, the platform encourages architects to write custom applications and simulation libraries. The machine performing this activity only logged traces during the day while doing actual work. The primary system user turned tracing on when they arrived, and off when they left for the day.

Workload Duration	21330	minutes		
Datafiles				
Datafile Read	173013	Requests	316044	KBytes
Datafile Write	430215	Requests	232785	KBytes
Datafile References	318613	Uses	2923	Unique
Executables				
Executable Reads	7733	Requests	1177987	KBytes (approx.)
Executable References	7733	Uses	142	Unique
Inodes and Directories				
Inode Read	1993535	Requests		
Inode Writes	3614	File system modifications		
Inode Update	1901811	Requests		
Inode Referencs	9409	Unique		
Directory Read	1013153	Requests		
Directory Write	3059	Requests		
Directory Referencs	828	Unique		
Operations				
315687 Opens	2926	Creates	2897	Deletes

Software Development 2 Workload

Software Development 2 (24 hr.) is the same basic environment as **Software Development (8 to 5)**. The trace includes idle time at night and all the maintenance activity that goes on at night. The trace records 4 days of activity.

Workload Duration	7264	minutes		
Datafiles				
Datafile Read	76807	Requests	199503	KBytes
Datafile Write	345133	Requests	129952	KBytes
Datafile References	313157	Uses	1555	Unique
Executables				
Executable Reads	7788	Requests	1158650	KBytes (approx.)
Executable References	7788	Uses	143	Unique
Inodes and Directories				
Inode Read	2016482	Requests		
Inode Writes	3557	File system modifications		
Inode Update	1923672	Requests		
Inode Referencs	7148	Unique		
Directory Read	1031202	Requests		
Directory Write	2896	Requests		
Directory Referencs	559	Unique		
Operations				
310413 Opens	2744	Creates	2516	Deletes

Mecca Development Workload

Mecca Development (24 hr.) records the development and testing of a centralized e-mail system [Bor92]. Mecca uses an Ingres database with information about individuals to properly direct incoming mail to the proper recipient. This workload includes development and testing of the interface modules.

Workload Duration	4578	minutes		
Datafiles				
Datafile Read	64329	Requests	147224	KBytes
Datafile Write	115123	Requests	94100	KBytes
Datafile References	60486	Uses	1054	Unique
Executables				
Executable Reads	3721	Requests	469734	KBytes (approx.)
Executable References	3721	Uses	145	Unique
Inodes and Directories				
Inode Read	618809	Requests		
Inode Writes	1620	File system modifications		
Inode Update	563633	Requests		
Inode Referencs	13244	Unique		
Directory Read	302648	Requests		
Directory Write	1329	Requests		
Directory Referencs	836	Unique		
Operations				
59179 Opens	1307	Creates	1416	Deletes

Mecca Development with Server Workload

Mecca Development with Server (8 to 5) is the same basic environment as **Mecca Development (24 hr.)**, except that (1) the Ingres server resides on the traced machine, and (2) the machine recorded traces only during the day.

Workload Duration	5665	minutes		
Datafiles				
Datafile Read	180864	Requests	341168	KBytes
Datafile Write	30077	Requests	95255	KBytes
Datafile References	19259	Uses	1056	Unique
Executables				
Executable Reads	1846	Requests	212003	KBytes (approx.)
Executable References	1846	Uses	146	Unique
Inodes and Directories				
Inode Read	172184	Requests		
Inode Writes	1510	File system modifications		
Inode Update	156301	Requests		
Inode Referencs	3589	Unique		
Directory Read	65509	Requests		
Directory Write	936	Requests		
Directory Referencs	315	Unique		
Operations				
18282 Opens	977	Creates	1017	Deletes

Network Update Workload

Network Update (24 hr.) mostly consists of a large network gather-scatter operation. The operation gathered information from the whole DEC NET and then updated the whole net with new information. Additional work included the typical UNIX applications, and some special network configuration tools that run on top of the Ingres database.

Workload Duration	39659	minutes		
Datafiles				
Datafile Read	1938499	Requests	4052343	KBytes
Datafile Write	275788	Requests	272051	KBytes
Datafile References	111709	Uses	3638	Unique
Executables				
Executable Reads	5026	Requests	587524	KBytes (approx.)
Executable References	5026	Uses	235	Unique
Inodes and Directories				
Inode Read	1143705	Requests		
Inode Writes	3929	File system modifications		
Inode Update	1071904	Requests		
Inode Referencs	24080	Unique		
Directory Read	599337	Requests		
Directory Write	2639	Requests		
Directory Referencs	1643	Unique		
Operations				
109073 Opens	2636	Creates	3699	Deletes

CAD: Chip Build Workload

CAD: Chip Build traces the construction of a CPU layout from a high level description [DM92]. The high level description is in C, and much of the build involves standard compilation of the various chip pieces into a single program. The program then constructs the layout of the chip. This is not a complete layout because the chip is only partially finished. The workload ran design tests on the compiled chip description.

Workload Duration	4799	minutes		
Datafiles				
Datafile Read	138828	Requests	275841	KBytes
Datafile Write	409271	Requests	872561	KBytes
Datafile References	205194	Uses	1800	Unique
Executables				
Executable Reads	5143	Requests	648846	KBytes (approx.)
Executable References	5143	Uses	111	Unique
Inodes and Directories				
Inode Read	1746158	Requests		
Inode Writes	2005	File system modifications		
Inode Update	1655082	Requests		
Inode Referencs	17765	Unique		
Directory Read	1003335	Requests		
Directory Write	1652	Requests		
Directory Referencs	975	Unique		
Operations				
203706 Opens	1488	Creates	1969	Deletes

Compute Server Workload

Compute Server (24 hr.) consists of batch simulations that ran on a machine with an idle console. The simulations evaluated potential architectural features from a compiler technology standpoint.

Workload Duration	11091	minutes		
Datafiles				
Datafile Read	144000	Requests	334709	KBytes
Datafile Write	558053	Requests	239883	KBytes
Datafile References	477959	Uses	3017	Unique
Executables				
Executable Reads	10905	Requests	1374629	KBytes (approx.)
Executable References	10905	Uses	144	Unique
Inodes and Directories				
Inode Read	5081786	Requests		
Inode Writes	6269	File system modifications		
Inode Update	4773893	Requests		
Inode Referencs	39894	Unique		
Directory Read	3087764	Requests		
Directory Write	5239	Requests		
Directory Referencs	3177	Unique		
Operations				
472634 Opens	5325	Creates	4227	Deletes

Appendix B

Workload Characterization Data

Knowledge of the statistical distribution of I/O requests can help design systems that achieve the best price/performance ratio. Some techniques for improving performance rely heavily on the statistical nature of requests. Caches work because, statistically, programs and work environments exhibit locality. A better understanding of the statistical properties of the workloads allows cache designers to exploit the most prominent statistical properties of the workloads. The characterization shows the type of requests that are the most important to performance, and focuses attention on the most important part of the problem to be solved.

This appendix presents the characteristics of I/O with no cache effects.

1. Characteristics by type of data transferred.
2. Characteristics by transfer type (read or write).
3. Characteristics by file size.

Workload	Static Measure Objects/ Size	Static Break- down (percent)	Ratio Dynamic Counts to Static Cnts	Dynamic Measure	Dynamic Break- down (percent)
Kernel Build					
	Inode Objects	51.3	114.06	References	68.0
	Directory Objects	5.1	308.40	References	18.2
	Executable Objects	2.0	14.96	References	0.3
	Datafile Objects	41.7	27.82	References	13.5
	Total Objects	100.0	85.96	References	100.0
	Inode Size	0.4	114.06	Bytes Trans.	9.1
	Directory Size	0.1	308.40	Bytes Trans.	9.7
	Executable Size	14.1	14.29	Bytes Trans.	44.0
	Datafile Size	85.4	1.99	Bytes Trans.	37.3
	Total Size	100.0	4.57	Bytes Trans.	100.0
Ingres					
	Inode Objects	55.3	81.07	References	41.1
	Directory Objects	7.4	157.46	References	10.7
	Executable Objects	7.5	6.14	References	0.4
	Datafile Objects	29.7	175.42	References	47.8
	Total Objects	100.0	109.17	References	100.0
	Inode Size	0.1	81.07	Bytes Trans.	2.1
	Directory Size	0.1	157.46	Bytes Trans.	2.2
	Executable Size	40.6	1.82	Bytes Trans.	14.6
	Datafile Size	59.2	6.96	Bytes Trans.	81.2
	Total Size	100.0	5.08	Bytes Trans.	100.0
Application Analysis					
	Inode Objects	64.8	229.25	References	63.4
	Directory Objects	8.4	502.44	References	18.1
	Executable Objects	5.2	6.34	References	0.1
	Datafile Objects	21.5	199.63	References	18.3
	Total Objects	100.0	234.30	References	100.0
	Inode Size	0.1	229.25	Bytes Trans.	12.2
	Directory Size	0.1	502.44	Bytes Trans.	13.9
	Executable Size	10.7	5.02	Bytes Trans.	21.3
	Datafile Size	89.1	1.49	Bytes Trans.	52.6
	Total Size	100.0	2.52	Bytes Trans.	100.0

Table B.1: Relative Static and Dynamic benchmark measures - one

Workload	Static Measure Objects/ Size	Static Break- down (percent)	Ratio Dynamic Counts to Static Cnts	Dynamic Measure	Dynamic Break- down (percent)
Document Preparation					
	Inode Objects	60.1	353.63	References	65.3
	Directory Objects	7.4	820.34	References	18.7
	Executable Objects	3.5	35.07	References	0.4
	Datafile Objects	29.0	176.36	References	15.7
	Total Objects	100.0	325.56	References	100.0
	Inode Size	0.1	353.63	Bytes Trans.	7.7
	Directory Size	0.0	820.34	Bytes Trans.	8.8
	Executable Size	6.7	13.97	Bytes Trans.	25.0
	Datafile Size	93.1	2.36	Bytes Trans.	58.5
	Total Size	100.0	3.76	Bytes Trans.	100.0
Software Development 2					
	Inode Objects	76.0	551.72	References	72.8
	Directory Objects	5.9	1849.91	References	19.1
	Executable Objects	1.5	54.46	References	0.1
	Datafile Objects	16.5	278.90	References	8.0
	Total Objects	100.0	576.21	References	100.0
	Inode Size	1.6	551.72	Bytes Trans.	20.0
	Directory Size	0.5	1849.91	Bytes Trans.	21.0
	Executable Size	36.7	56.63	Bytes Trans.	45.9
	Datafile Size	61.1	9.67	Bytes Trans.	13.1
	Total Size	100.0	45.27	Bytes Trans.	100.0
MECCA Server					
	Inode Objects	70.3	91.95	References	54.0
	Directory Objects	6.2	210.94	References	10.9
	Executable Objects	2.9	12.64	References	0.3
	Datafile Objects	20.7	201.09	References	34.8
	Total Objects	100.0	119.59	References	100.0
	Inode Size	0.8	91.95	Bytes Trans.	5.8
	Directory Size	0.3	210.94	Bytes Trans.	4.7
	Executable Size	41.6	9.15	Bytes Trans.	29.3
	Datafile Size	57.3	13.68	Bytes Trans.	60.2
	Total Size	100.0	13.01	Bytes Trans.	100.0

Table B.2: Relative Static and Dynamic benchmark measures - two

Workload	Static Measure Objects/ Size	Static Break-down (percent)	Ratio Dynamic Counts to Static Cnts	Dynamic Measure	Dynamic Break-down (percent)
MECCA Development					
	Inode Objects	86.7	89.40	References	70.6
	Directory Objects	5.5	363.61	References	18.1
	Executable Objects	0.9	25.66	References	0.2
	Datafile Objects	6.9	176.17	References	11.1
	Total Objects	100.0	109.79	References	100.0
	Inode Size	2.7	89.40	Bytes Trans.	14.9
	Directory Size	0.7	363.61	Bytes Trans.	15.3
	Executable Size	35.8	20.83	Bytes Trans.	46.1
	Datafile Size	60.8	6.30	Bytes Trans.	23.7
	Total Size	100.0	16.16	Bytes Trans.	100.0
CAD Chip Build					
	Inode Objects	86.0	191.57	References	68.5
	Directory Objects	4.7	1030.76	References	20.2
	Executable Objects	0.5	46.33	References	0.1
	Datafile Objects	8.7	307.51	References	11.1
	Total Objects	100.0	240.52	References	100.0
	Inode Size	0.3	191.57	Bytes Trans.	16.0
	Directory Size	0.1	1030.76	Bytes Trans.	18.9
	Executable Size	1.7	47.63	Bytes Trans.	23.8
	Datafile Size	97.9	1.47	Bytes Trans.	41.2
	Total Size	100.0	3.48	Bytes Trans.	100.0
Network Update					
	Inode Objects	81.4	92.17	References	44.0
	Directory Objects	5.6	366.39	References	11.9
	Executable Objects	0.8	21.39	References	0.1
	Datafile Objects	12.3	610.15	References	44.0
	Total Objects	100.0	170.51	References	100.0
	Inode Size	2.0	92.17	Bytes Trans.	5.2
	Directory Size	0.6	366.39	Bytes Trans.	5.6
	Executable Size	39.3	9.88	Bytes Trans.	10.7
	Datafile Size	58.1	49.27	Bytes Trans.	78.6
	Total Size	100.0	36.42	Bytes Trans.	100.0

Table B.3: Relative Static and Dynamic benchmark measures - three

Workload	Static Measure Objects/ Size	Static Break-down (percent)	Ratio Dynamic Counts to Static Cnts	Dynamic Measure	Dynamic Break-down (percent)
Software Development 1					
	Inode Objects	70.7	414.39	References	70.4
	Directory Objects	6.2	1227.31	References	18.4
	Executable Objects	1.1	54.46	References	0.1
	Datafile Objects	22.0	209.59	References	11.1
	Total Objects	100.0	416.14	References	100.0
	Inode Size	1.2	414.39	Bytes Trans.	18.2
	Directory Size	0.4	1227.31	Bytes Trans.	19.0
	Executable Size	29.4	40.27	Bytes Trans.	42.9
	Datafile Size	69.0	7.98	Bytes Trans.	20.0
	Total Size	100.0	27.57	Bytes Trans.	100.0
CPU Server					
	Inode Objects	86.3	247.20	References	72.1
	Directory Objects	6.9	973.56	References	22.6
	Executable Objects	0.3	75.73	References	0.1
	Datafile Objects	6.5	236.91	References	5.2
	Total Objects	100.0	295.91	References	100.0
	Inode Size	5.7	247.20	Bytes Trans.	26.3
	Directory Size	1.8	973.56	Bytes Trans.	33.0
	Executable Size	25.4	60.52	Bytes Trans.	28.7
	Datafile Size	67.1	9.57	Bytes Trans.	12.0
	Total Size	100.0	53.57	Bytes Trans.	100.0

Table B.4: Relative Static and Dynamic benchmark measures - four

File type	Percent of requests	Breakdown for type (percent)	Percent Read requests	Percent Write requests
Kernel Build				
Inode	68.0	50.0 Read	34.0	
		49.9 writes		33.9
Directory	18.2	99.7 Read	18.1	
		0.3 Write		0.1
Executable	0.3	100.0 Read	0.3	
Datafile	13.5	60.3 Read	8.1	
		39.7 Write		5.3
Total RW Breakdown			60.6	39.4
Ingres				
Inode	41.1	50.2 Read	20.6	
		49.8 Write		20.4
Directory	10.7	98.4 Read	10.6	
		1.6 Write		0.2
Executable	0.4	100.0 Read	0.4	
Datafile	47.8	39.4 Read	18.9	
		60.6 Write		28.9
Total RW Breakdown			50.4	49.5
Application Analysis				
Inode	63.4	50.8 Read	32.2	
		49.2 Write		31.3
Directory	18.1	99.9 Read	18.0	
		0.1 Write		0.1
Executable	0.1	100.0 Read	0.1	
Datafile	18.3	30.9 Read	5.7	
		69.1 Write		12.7
Total RW Breakdown			56.1	43.9

Table B.5: Benchmark RW request measures - one

File type	Percent of requests	Breakdown for type (percent)	Percent Read requests	Percent Write requests
Document Preparation				
Inode	65.3	50.3 Read	32.8	
		49.7 Write		32.4
Directory	18.7	99.7 Read	18.6	
		0.3 Write		0.1
Executable	0.4	100.0 Read	0.4	
Datafile	15.7	65.4 Read	10.3	
		34.6 Write		5.4
Total RW Breakdown			62.1	37.9
Software Development 2				
Inode	72.8	51.1 Read	37.2	
		48.9 Write		35.6
Directory	19.1	99.7 Read	19.0	
		0.3 Write		0.1
Executable	0.1	100.0 Read	0.1	
Datafile	8.0	20.2 Read	1.6	
		79.8 Write		6.4
Total RW Breakdown			58.0	42.0
MECCA Server				
Inode	54.0	52.2 Read	28.2	
		47.9 Write		25.8
Directory	10.9	98.6 Read	10.7	
		1.4 Write		0.2
Executable	0.3	100.0 Read	0.3	
Datafile	34.8	85.7 Read	29.8	
		14.3 Write		5.0
Total RW Breakdown			69.0	31.0

Table B.6: Benchmark RW request measures - two

File type	Percent of requests	Breakdown for type (percent)	Percent Read requests	Percent Write requests
MECCA Development				
Inode	70.6	52.3 Read	36.9	
		47.7 Write		33.7
Directory	18.1	99.6 Read	18.0	
		0.4 Write		0.1
Executable	0.2	100.0 Read	0.2	
Datafile	11.1	37.8 Read	4.2	
		62.2 Write		6.9
Total RW Breakdown			59.3	40.7
CAD Chip Build				
Inode	68.5	51.3 Read	35.2	
		48.7 Write		33.3
Directory	20.2	99.8 Read	20.2	
		0.2 Write		0.1
Executable	0.1	100.0 Read	0.1	
Datafile	11.1	25.9 Read	2.9	
		74.1 Write		8.3
Total RW Breakdown			58.3	41.7
Network Update				
Inode	44.0	51.5 Read	22.7	
		48.5 Write		21.3
Directory	11.9	99.6 Read	11.9	
		0.4 Write		0.1
Executable	0.1	100.0 Read	0.1	
Datafile	44.0	87.5 Read	38.5	
		12.5 Write		5.5
Total RW Breakdown			73.1	26.8

Table B.7: Benchmark RW request measures - three

File type	Percent of requests	Breakdown for type (percent)	Percent Read requests	Percent Write requests
Software Development 1				
Inode	70.4	51.1 Read	36.0	
		48.9 Write		34.5
Directory	18.4	99.7 Read	18.3	
		0.3 Write		0.1
Executable	0.1	100.0 Read	0.1	
Datafile	11.1	29.5 Read	3.3	
		70.5 Write		7.8
Total RW Breakdown			57.7	42.3
CPU Server				
		48.5 Read		34.9
Directory	22.6	99.8 Read	22.6	
		0.2 Write		0.1
Executable	0.1	100.0 Read	0.1	
Datafile	5.2	21.7 Read	1.1	
		78.3 Write		4.1
Total RW Breakdown			60.9	39.1

Table B.8: Benchmark RW request measures - four

File type	Percent of bytes	Breakdown for type (percent)	Percent Read bytes	Percent Write bytes
Kernel Build				
Inode	9.1	50.0 Read	4.5	
		49.8 Write		4.5
Directory	9.7	99.7 Read	9.7	
		0.3 Write		0.0
Executable	44.0	100.0 Read	44.0	
Datafile	37.3	84.9 Read	31.6	
		15.1 Write		5.6
Total RW Breakdown			89.8	10.1
Ingres				
Inode	2.1	50.2 Read	1.0	
		49.2 Write		1.0
Directory	2.2	98.4 Read	2.1	
		1.6 Write		0.0
Executable	14.6	100.0 Read	14.6	
Datafile	81.2	21.5 Read	17.4	
		78.5 Write		63.8
Total RW Breakdown			35.2	64.8
Application Analysis				
Inode	12.2	50.8 Read	6.2	
		49.1 Write		6.0
Directory	13.9	99.9 Read	13.9	
		0.1 Write		0.0
Executable	21.3	100.0 Read	21.3	
Datafile	52.6	78.8 Read	41.4	
		21.2 Write		11.2
Total RW Breakdown			82.8	17.2

Table B.9: Benchmark RW byte measures - one

File type	Percent of bytes	Breakdown for type (percent)	Percent Read bytes	Percent Write bytes
Type	Percent	Breakdown RW-types	Read	Write
Document Preparation				
Inode	7.7	50.3 Read	3.9	
		49.6 Write		3.8
Directory	8.8	99.7 Read	8.7	
		0.3 Write		0.0
Executable	25.0	100.0 Read	25.0	
Datafile	58.5	95.0 Read	55.6	
		5.0 Write		2.9
Total RW Breakdown			93.2	6.7
Software Development 2				
Inode	20.0	51.1 Read	10.2	
		48.8 Write		9.8
Directory	21.0	99.7 Read	20.9	
		0.3 Write		0.1
Executable	45.9	100.0 Read	45.9	
Datafile	13.1	60.5 Read	7.9	
		39.5 Write		5.2
Total RW Breakdown			85.0	15.0
MECCA Server				
Inode	5.8	52.2 Read	3.0	
		47.4 Write		2.8
Directory	4.7	98.6 Read	4.6	
		1.4 Write		0.1
Executable	29.3	100.0 Read	29.3	
Datafile	60.2	78.2 Read	47.1	
		21.8 Write		13.1
Total RW Breakdown			84.0	16.0

Table B.10: Benchmark RW byte measures - two

File type	Percent of bytes	Breakdown for type (percent)	Percent Read bytes	Percent Write bytes
MECCA Development				
Inode	14.9	52.3 Read	7.8	
		47.6 Write		7.1
Directory	15.3	99.6 Read	15.2	
		0.4 Write		0.1
Executable	46.1	100.0 Read	46.1	
Datafile	23.7	61.0 Read	14.5	
		39.0 Write		9.2
Total RW Breakdown			83.6	16.4
CAD Chip Build				
Inode	16.0	51.3 Read	8.2	
		48.6 Write		7.8
Directory	18.9	99.8 Read	18.9	
		0.2 Write		0.0
Executable	23.8	100.0 Read	23.8	
Datafile	41.2	22.2 Read	9.2	
		77.8 Write		32.1
Total RW Breakdown			60.1	39.9
Network Update				
Inode	5.2	51.5 Read	2.7	
		48.3 Write		2.5
Directory	5.6	99.6 Read	5.6	
		0.4 Write		0.0
Executable	10.7	100.0 Read	10.7	
Datafile	78.6	93.7 Read	73.6	
		6.3 Write		5.0
Total RW Breakdown			92.5	7.5

Table B.11: Benchmark RW byte measures - three

File type	Percent of bytes	Breakdown for type (percent)	Percent Read bytes	Percent Write bytes
Software Development 1				
Inode	18.2	51.1 Read	9.3	
		48.8 Write		8.9
Directory	19.0	99.7 Read	18.9	
		0.3 Write		0.1
Executable	42.9	100.0 Read	42.9	
Datafile	20.0	57.3 Read	11.4	
		42.7 Write		8.5
Total RW Breakdown			82.5	17.5
CPU Server				
Inode	26.3	51.5 Read	13.6	
		48.4 Write		12.7
Directory	33.0	99.8 Read	33.0	
		0.2 Write		0.1
Executable	28.7	100.0 Read	28.7	
Datafile	12.0	58.3 Read	7.0	
		41.7 Write		5.0
Total RW Breakdown			82.2	17.8

Table B.12: Benchmark RW byte measures - four

	Number of Unique Files			Total Size of Unique Files (Bytes)		
	Read Only	Write Only	Read and Written	Read Only	Write Only	Read and Written
Kernel Build						
Inode	17	0	1310	2176	0	167680
Directory	79	0	51	60416	0	34816
Executable	52			6567235		
Datafile	521	89	469	27036605	1593250	11242638
Ingres						
Inode	62	0	503	7936	0	64384
Directory	53	0	22	31744	0	18432
Executable	77			22520447		
Datafile	82	150	72	8404721	1679101	22782347
Application Analysis						
Inode	402	0	583	51456	0	74624
Directory	89	0	38	36864	0	30720
Executable	79			10053175		
Datafile	140	106	81	63544690	14889988	5297367
Document Preparation						
Inode	354	0	814	45312	0	104192
Directory	95	0	48	44032	0	28672
Executable	69			12358509		
Datafile	230	106	227	149849477	7581563	13359422

Table B.13: Read, Write, and read-write file breakdown -one

	Number of Unique Files			Total Size of Unique Files (Bytes)		
	Read Only	Write Only	Read and Written	Read Only	Write Only	Read and Written
Software Development 2						
Inode	4928	0	2220	630784	0	284160
Directory	173	0	385	151552	0	332288
Executable	143			20458846		
Datafile	396	587	572	10879843	2801061	20370024
MECCA Server						
	Read	Write	Read-Write	RSize	WSize	RWsize
Inode	2016	0	1573	258048	0	201344
Directory	151	0	163	122880	0	77824
Executable	146			23159126		
Datafile	266	308	482	9146556	5259243	17492821
MECCA Development						
	Read	Write	Read-Write	RSize	WSize	RWsize
Inode	11327	0	1917	1449856	0	245376
Directory	287	0	548	160768	0	464896
Executable	145			22550814		
Datafile	228	323	503	10922535	6407136	20990240

Table B.14: Read, Write, and read-write file breakdown -two

	Number of Unique Files			Total Size of Unique Files (Bytes)		
	Read Only	Write Only	Read and Written	Read Only	Write Only	Read and Written
CAD Chip Build						
Inode	14891	0	2874	1906048	0	367872
Directory	318	0	656	284672	0	452096
Executable	111			13623139		
Datafile	644	530	626	30180755	691939161	42633967
Network Update						
Inode	18646	0	5434	2386688	0	695552
Directory	484	0	1158	413696	0	857197
Executable	235			59454820		
Datafile	1299	515	1824	31558030	7836531	48410862
Software Development 1						
Inode	5502	0	3907	704256	0	500096
Directory	278	0	549	248320	0	486408
Executable	142			29249815		
Datafile	1487	742	694	30930015	14111245	23674308
CPU Server						
Inode	33519	0	6375	4290432	0	816000
Directory	649	0	2527	479232	0	1582592
Executable	144			22712488		
Datafile	773	893	1351	17703701	7026312	35330725

Table B.15: Read, Write, and read-write file breakdown -three

Maximum Size Catagory	1k	4k	8k	16k	32k	64k	256k	512k	1M	4M	GT 4M
Kernel Build											
Objects	1	0	2	4	6	6	26	6	1	0	0
Uses	8	0	11	40	294	132	94	200	2	0	0
Reuse	8.00	-	5.50	10.00	49.00	22.00	3.62	33.33	2.00	-	-
Ingres											
Objects	1	5	3	11	11	3	30	8	1	4	0
Uses	12	122	36	48	178	30	90	25	1	4	0
Reuse	12.00	24.40	12.00	4.36	16.18	10.00	3.00	3.12	1.00	1.00	-
Application Analysis											
Objects	1	0	2	12	14	9	30	9	1	1	0
Uses	16	0	21	63	114	63	186	56	4	1	0
Reuse	16.00	-	10.50	5.25	8.14	7.00	6.20	6.22	4.00	1.00	-
Document Preparation											
Objects	1	2	2	8	15	7	23	6	2	3	0
Uses	30	68	11	181	618	923	567	57	9	5	0
Reuse	30.00	34.00	5.50	22.63	41.20	131.86	24.65	9.50	4.50	1.67	-
Software Development 2											
Objects	9	1	2	24	25	12	52	9	7	2	0
Uses	326	12	795	444	813	1924	2313	775	539	4	0
Reuse	36.22	12.00	397.50	18.50	32.52	160.33	44.48	86.11	77.00	2.00	-
MECCA Server											
Objects	6	6	3	31	21	10	49	9	8	3	0
Uses	62	126	139	135	257	322	694	149	49	4	0
Reuse	10.33	21.00	46.33	4.35	12.24	32.20	14.16	16.56	6.12	1.33	-
MECCA Development											
Objects	8	5	3	22	24	9	54	10	8	2	0
Uses	88	116	422	160	484	881	1134	406	135	2	0
Reuse	11.00	23.20	140.67	7.27	20.17	97.89	21.00	40.60	16.88	1.00	-
CAD Chip Build											
Objects	5	3	3	11	23	12	41	8	5	0	0
Uses	107	372	520	173	434	1494	1578	531	172	0	0
Reuse	21.40	124.00	173.33	15.73	18.87	124.50	38.49	66.38	34.40	-	-
Network Update											
Objects	8	11	4	42	28	11	89	22	7	13	0
Uses	260	142	462	531	867	803	1595	436	119	23	0
Reuse	32.50	12.91	115.50	12.64	30.96	73.00	17.92	19.82	17.00	1.77	-
Software Development 1											
Objects	9	0	3	14	22	12	59	12	6	5	0
Uses	384	0	809	315	600	1707	2675	980	321	46	0
Reuse	42.67	-	269.67	22.50	27.27	142.25	45.34	81.67	53.50	9.20	-
CPU Server											
Objects	17	2	3	13	27	12	49	13	4	4	0
Uses	444	30	1239	400	1025	3169	3195	1191	396	5	0
Reuse	26.12	15.00	413.00	30.77	37.96	264.08	65.20	91.62	99.00	1.25	-

Table B.16: Executable Files - Size and Usage by Size Distribution - one

Maximum Size Catagory	1k	4k	8k	16k	32k	64k	256k	512k	1M	4M	GT 4M
Kernel Build											
Objects	298	139	104	101	90	166	175	0	1	5	0
Uses	5833	2316	1438	1175	464	363	385	0	14	14	0
Reuse	19.57	16.66	13.83	11.63	5.16	2.19	2.20	-	14.00	2.80	-
Ingres											
Objects	243	26	6	2	7	8	3	0	2	5	2
Uses	2719	269	70	2	9	12	19	0	165	18	6
Reuse	11.19	10.35	11.67	1.00	1.29	1.50	6.33	-	82.50	3.60	3.00
Application Analysis											
Objects	147	45	17	39	25	18	17	1	2	11	5
Uses	15256	335	80	136	84	72	79	2	59	75	21
Reuse	103.78	7.44	4.71	3.49	3.36	4.00	4.65	2.00	29.50	6.82	4.20
Document Preparation											
Objects	235	75	26	97	28	16	67	7	4	7	1
Uses	10223	964	231	3578	128	186	271	45	79	63	2
Reuse	43.50	12.85	8.88	36.89	4.57	11.63	4.04	6.43	19.75	9.00	2.00
Software Development 2											
Objects	1062	220	52	46	23	9	126	7	8	2	0
Uses	279248	8240	794	936	132	42	7465	18	2372	26	0
Reuse	262.95	37.45	15.27	20.35	5.74	4.67	59.25	2.57	296.50	13.00	-
MECCA Server											
Objects	718	147	45	53	37	10	33	4	3	5	1
Uses	11526	2616	317	202	112	84	920	20	765	175	3
Reuse	16.05	17.80	7.04	3.81	3.03	8.40	27.88	5.00	255.00	35.00	3.00
MECCA Development											
Objects	802	73	36	36	9	14	74	1	2	6	1
Uses	44839	3857	626	209	25	56	3736	4	1506	107	1
Reuse	55.91	52.84	17.39	5.81	2.78	4.00	50.49	4.00	753.00	17.83	1.00
CAD Chip Build											
Objects	1154	271	59	55	54	13	133	26	13	18	4
Uses	181197	5627	1343	290	110	236	4831	236	1919	529	9
Reuse	157.02	20.76	22.76	5.27	2.04	18.15	36.32	9.08	147.62	29.39	2.25
Network Update											
Objects	1827	1064	238	125	120	97	134	12	10	9	2
Uses	53324	19673	2481	757	565	895	5388	46	14273	246	4
Reuse	29.19	18.49	10.42	6.06	4.71	9.23	40.21	3.83	1427.30	27.33	2.00
Software Development 1											
Objects	1570	532	236	175	116	63	199	13	10	8	1
Uses	282300	9257	1259	903	209	144	7489	24	2626	263	11
Reuse	179.81	17.40	5.33	5.16	1.80	2.29	37.63	1.85	262.60	32.88	11.00
CPU Server											
Objects	2113	328	148	82	56	45	230	6	2	7	0
Uses	422011	11648	724	514	77	82	10520	29	3827	451	0
Reuse	199.72	35.51	4.89	6.27	1.38	1.82	45.74	4.83	1913.50	64.43	

Table B.17: Datafile - Size and Usage by Size Distribution - one

B.1 Performance Bounds

These numbers can provide a minimum and maximum bounds for the I/O traffic of the workloads.

$$Minimum = Objects * overhead + \sum_{i=types} Min(size_i, bytes_i) * Rate$$

$$Maximum = References * overhead + \sum_{i=types} Max(size_i, bytes_i) * Rate$$

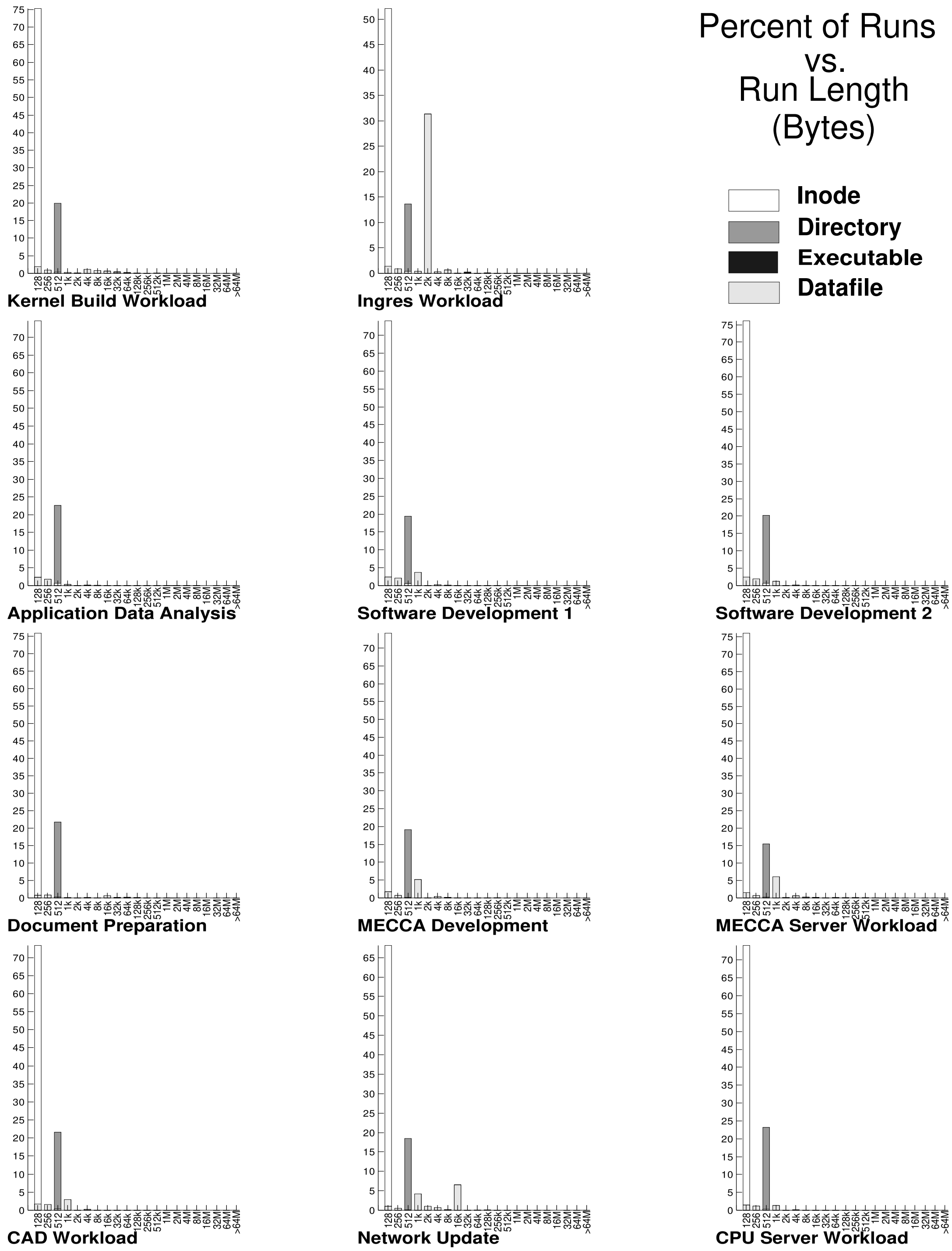

Figure B.1: Distribution of Run Lengths.

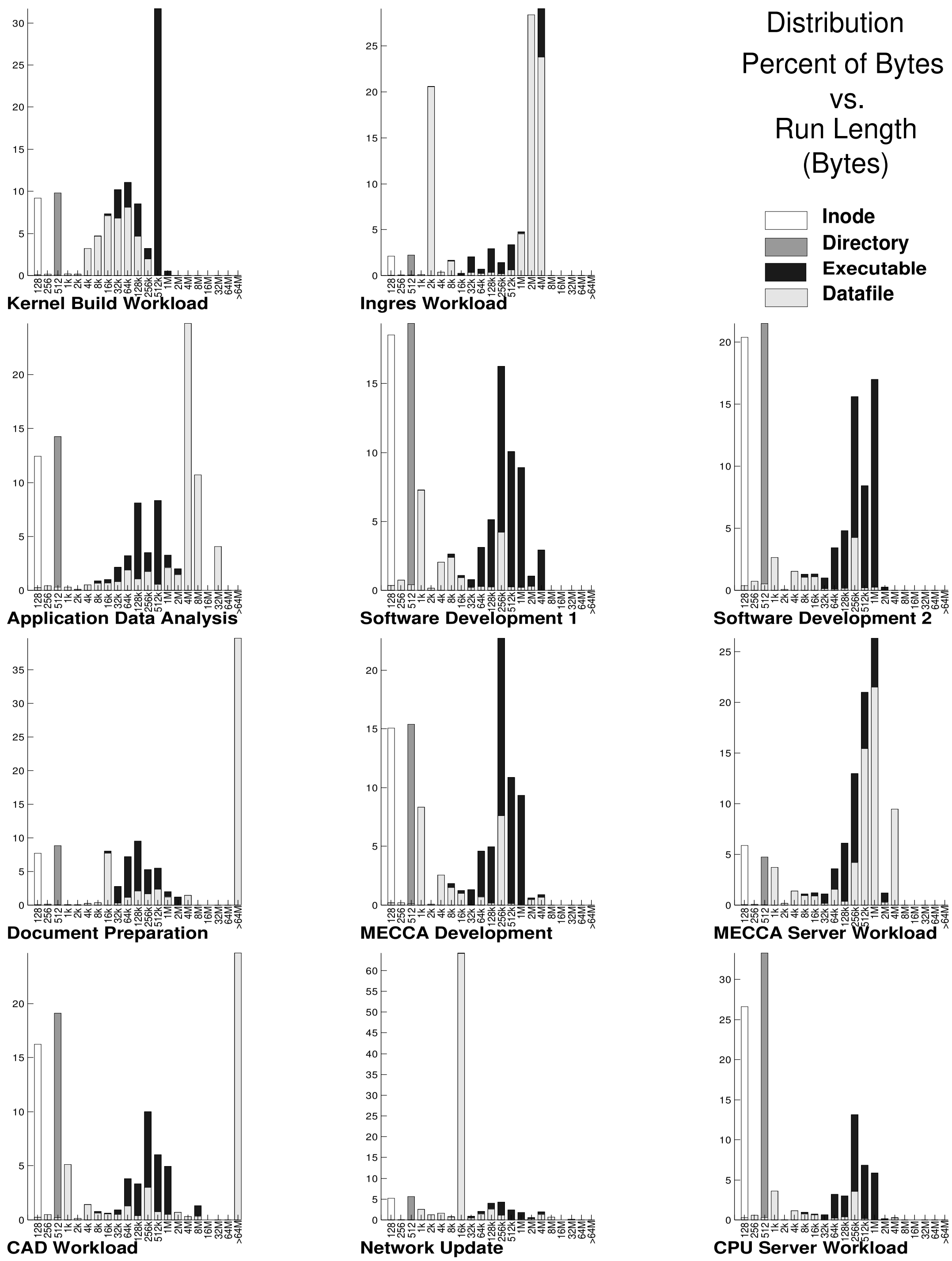

Figure B.2: Distribution of Bytes Transferred evaluated by Run Length.

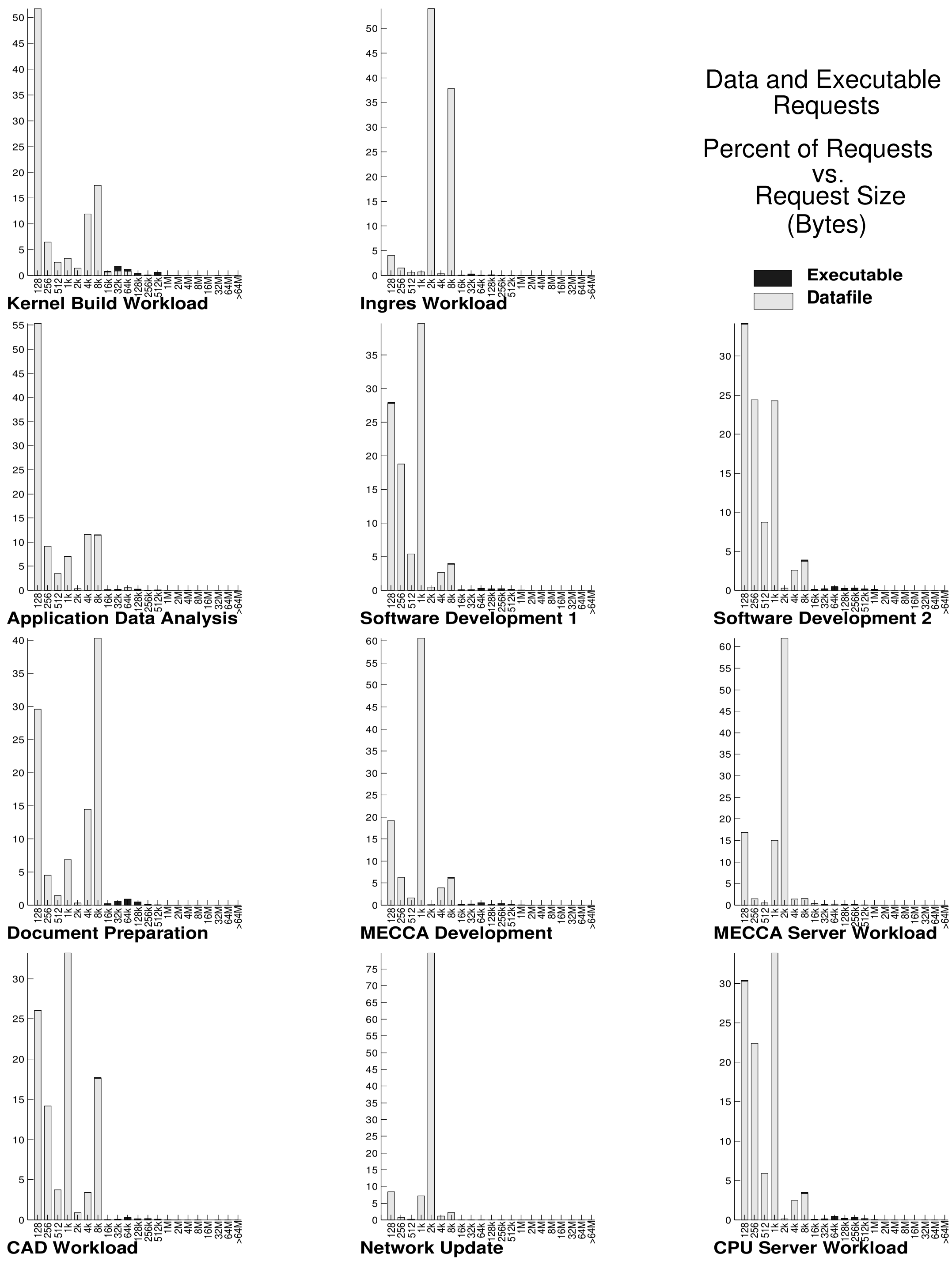

Figure B.3: Distribution of Request Sizes for Datafiles and Executables.

Appendix C

Workload Cache Behavior Data

Individually measuring the cache behavior of each of the four different file types—inode, directory, executable, and datafile—helps evaluate the usefulness of type information for an I/O cache. Differences can potentially be exploited to improve the cache behavior of the entire workload. The cache behavior of each type, along with the system utilization of each type, determine system I/O performance.

This appendix presents the cache behavior of the workload components in individual caches.

- Inode and Directory Cache Behavior.
- Executable Cache Behavior
- Datafile Cache Behavior
- Block Size Effect on Locality Capture
- Block Size Effect on Inode and Directory Locality Capture
- Block Size Impact on Traffic
- Separating Spatial and Temporal Locality
- Sequential Cache Properties
- Temporal Cache Properties

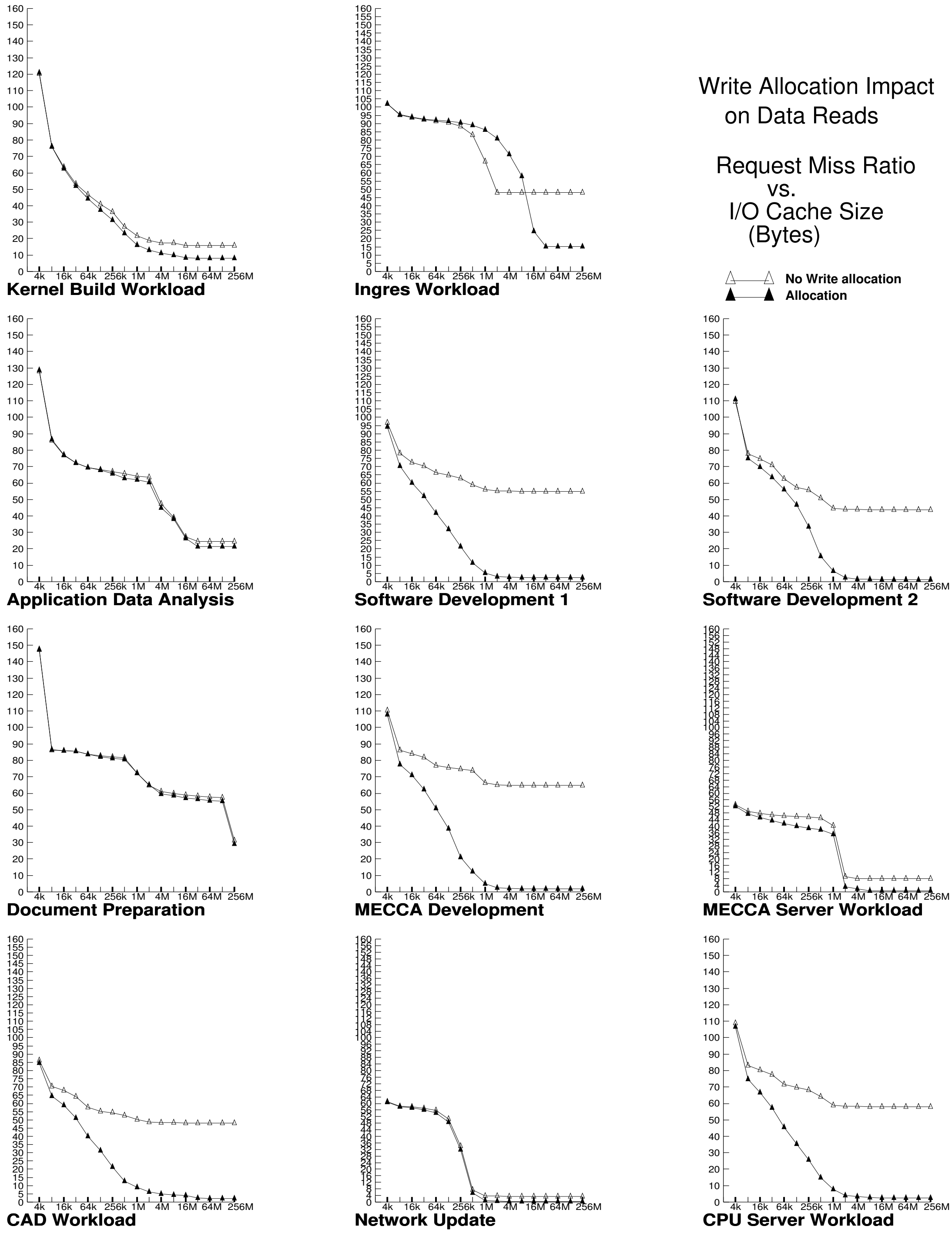

Figure C.1: Read Request Misses With and Without Write Allocation in the Cache.

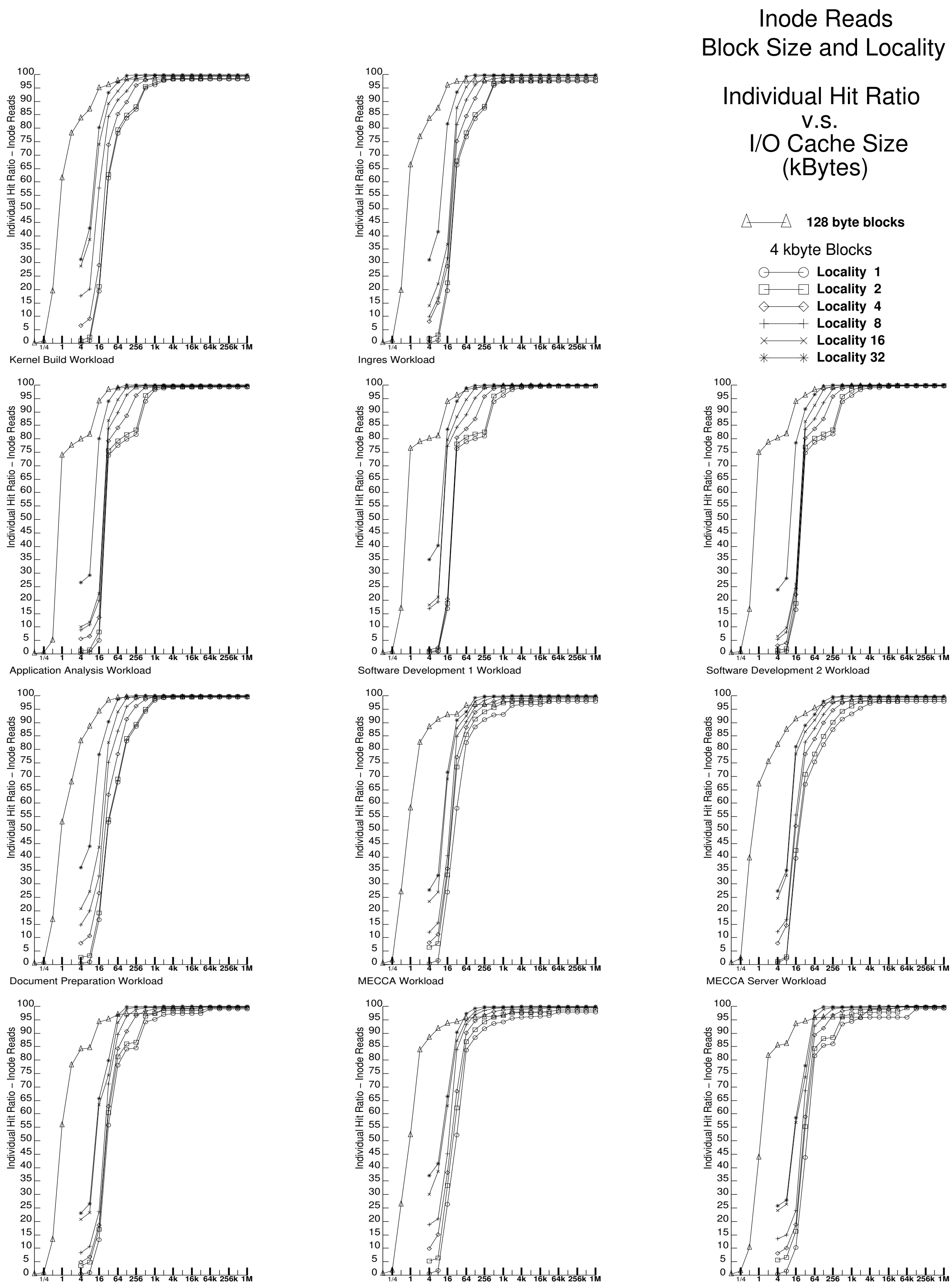

Figure C.2: Cache Hit Ratio for Inode Reads with Varying Amounts of Locality.

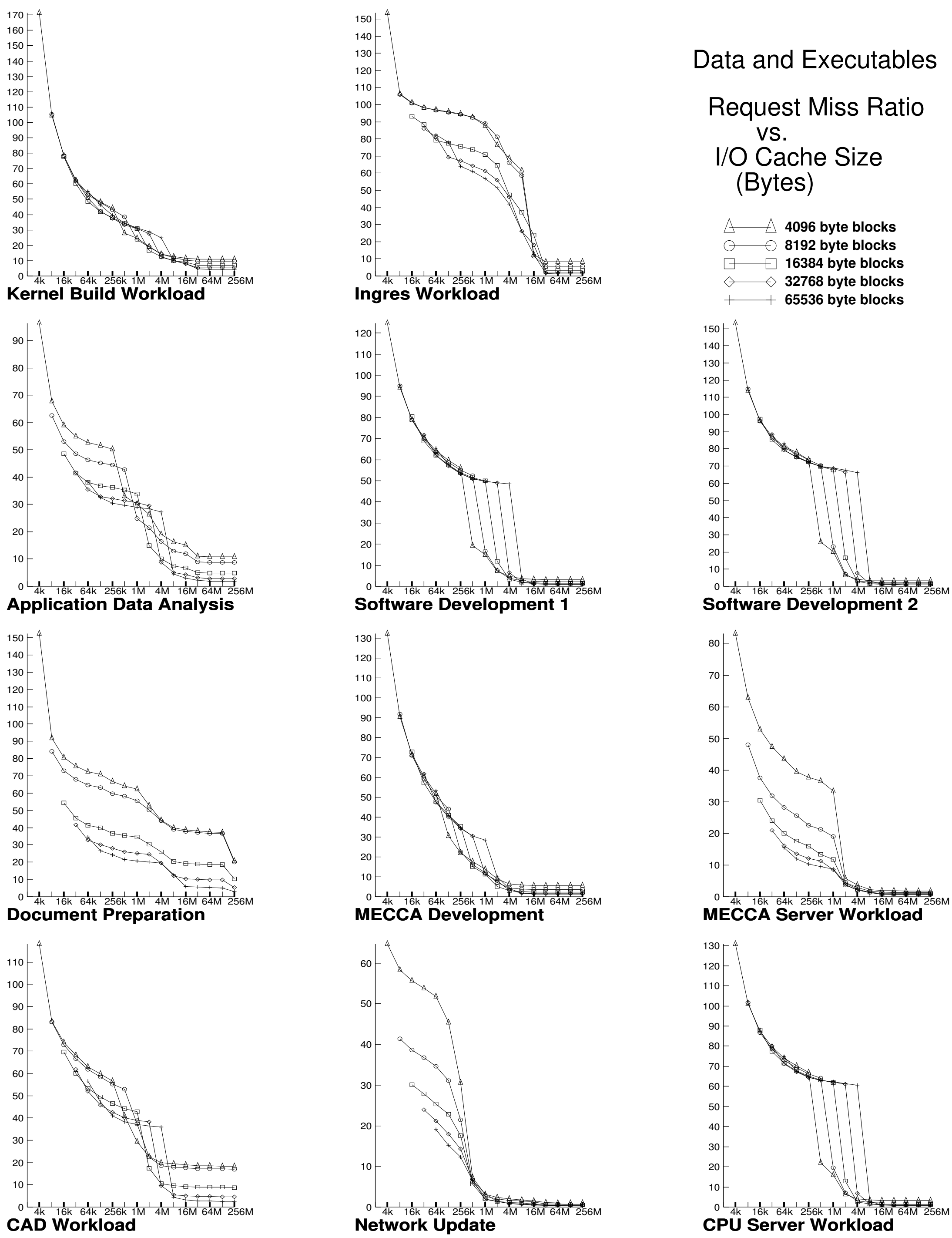

Figure C.3: Request Miss Ratio for various block sizes.

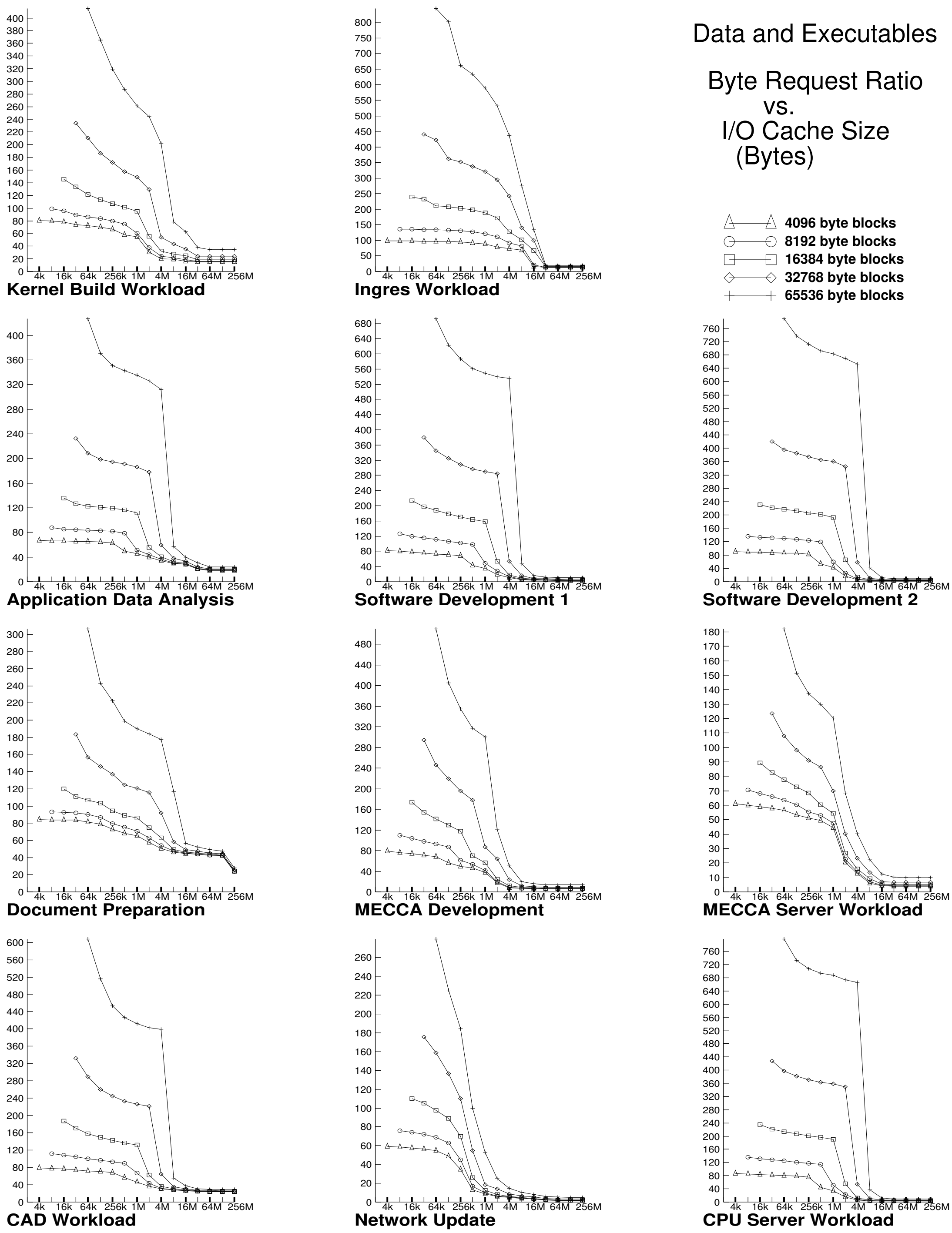

Figure C.4: Relative Traffic (based on blocks) for various block sizes.

Figure C.5: Request Miss Counts for various block sizes.

Figure C.6: Raw Traffic (based on blocks) for various block sizes.

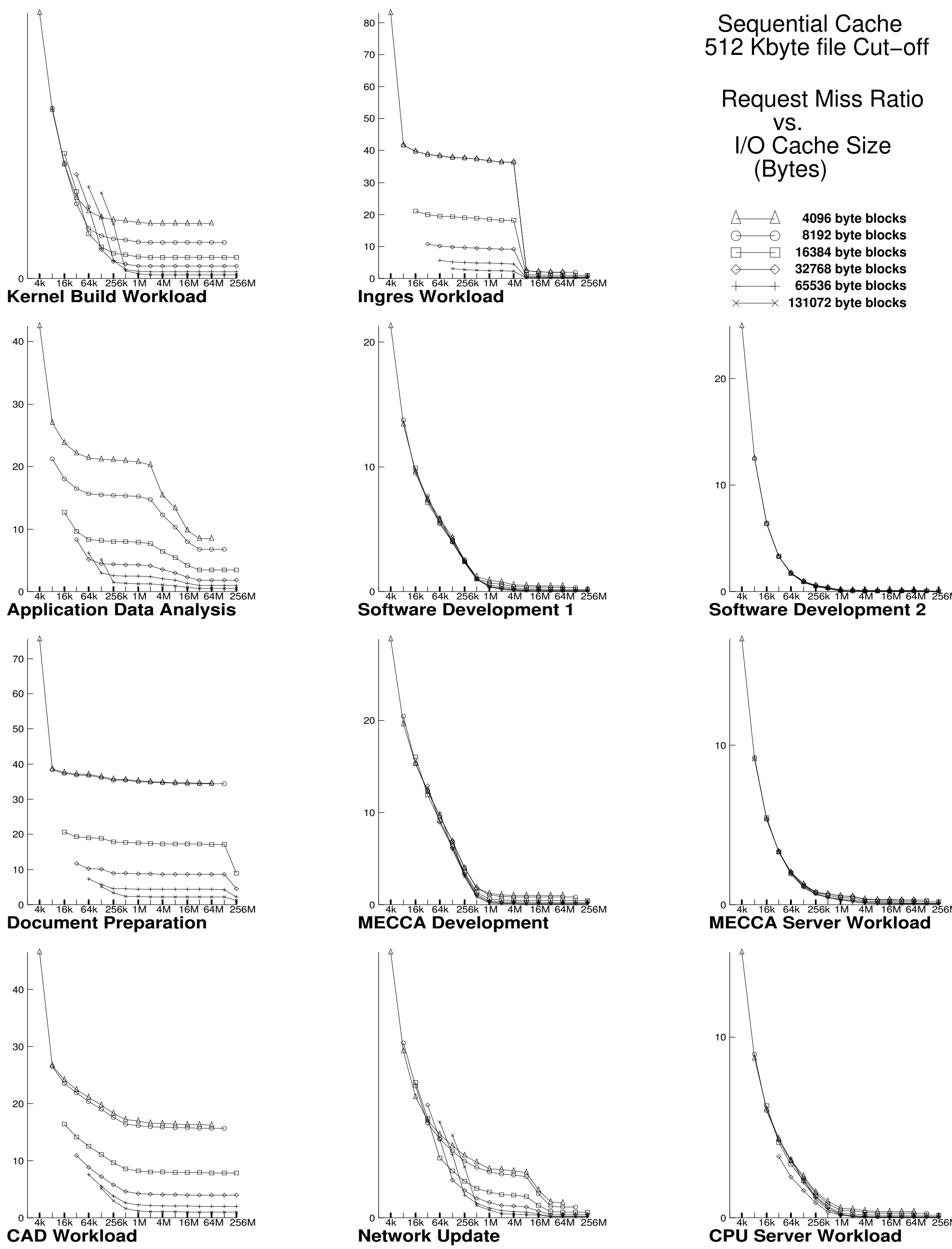

Figure C.7: Sequential Cache Request Misses for a File Cut-off of 512 Kb (rel.)

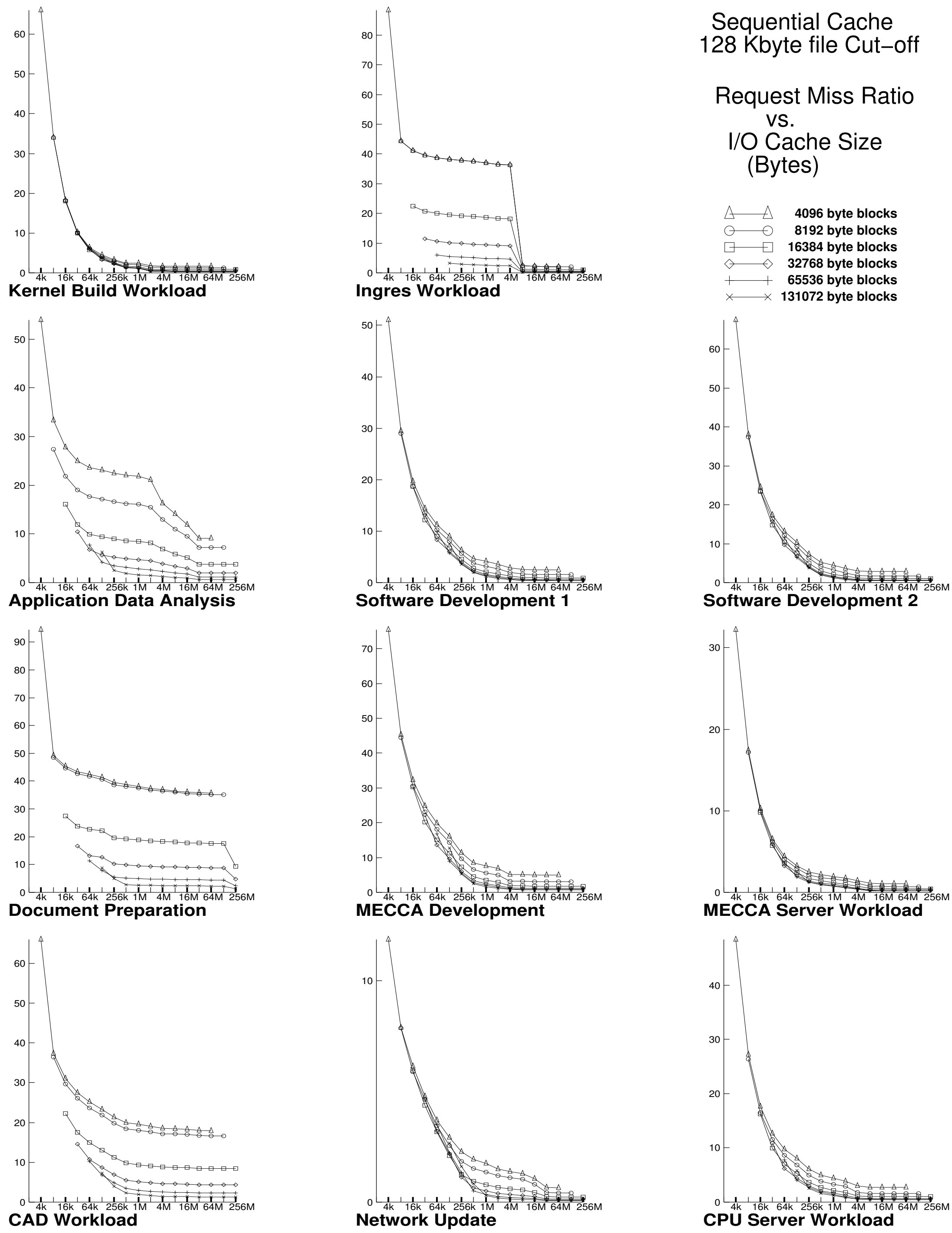

Figure C.8: Sequential Cache Request Misses for a File Cut-off of 128 Kb (rel.)

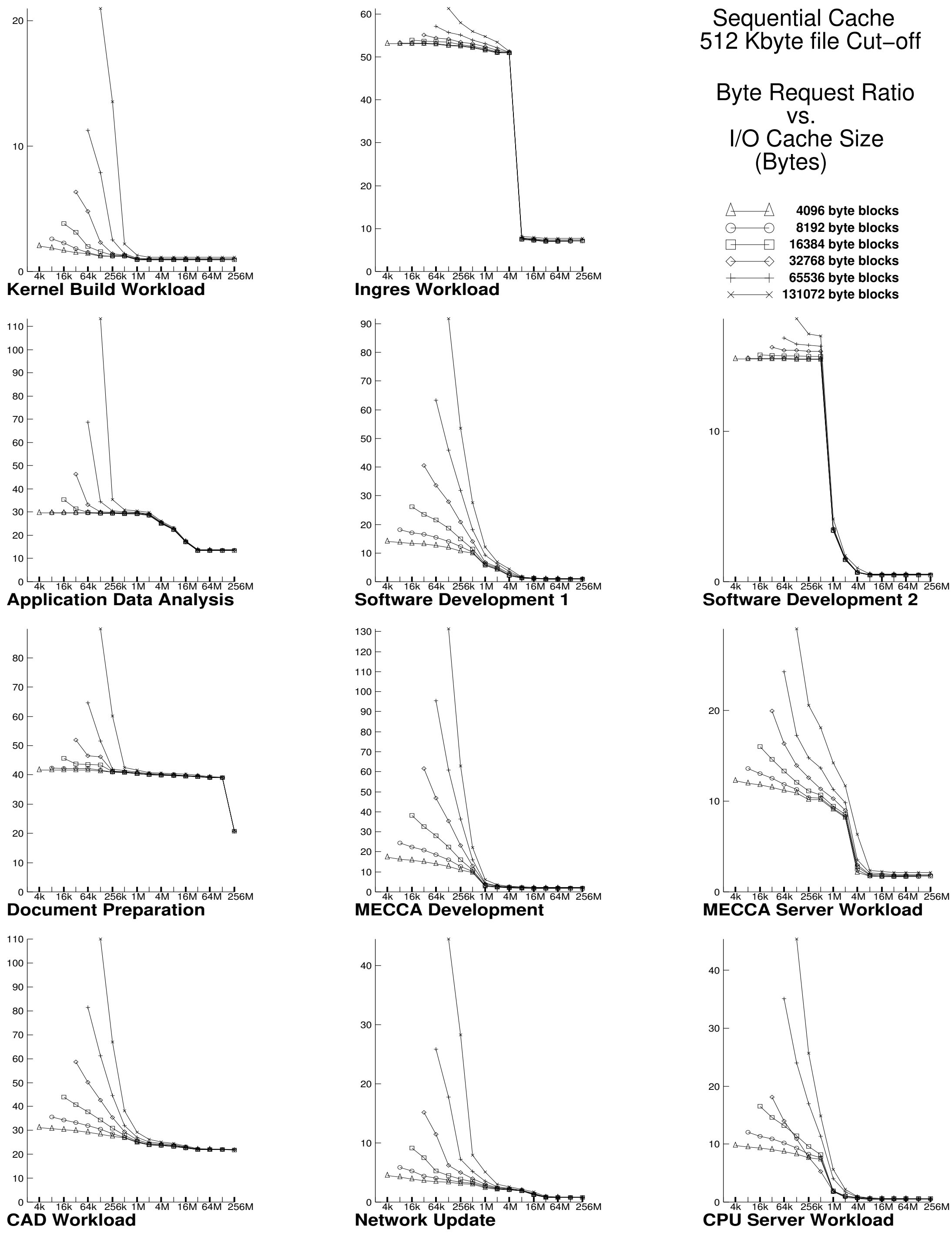

Figure C.9: Sequential Cache Traffic a File Cut-off of 512 Kb (relative)

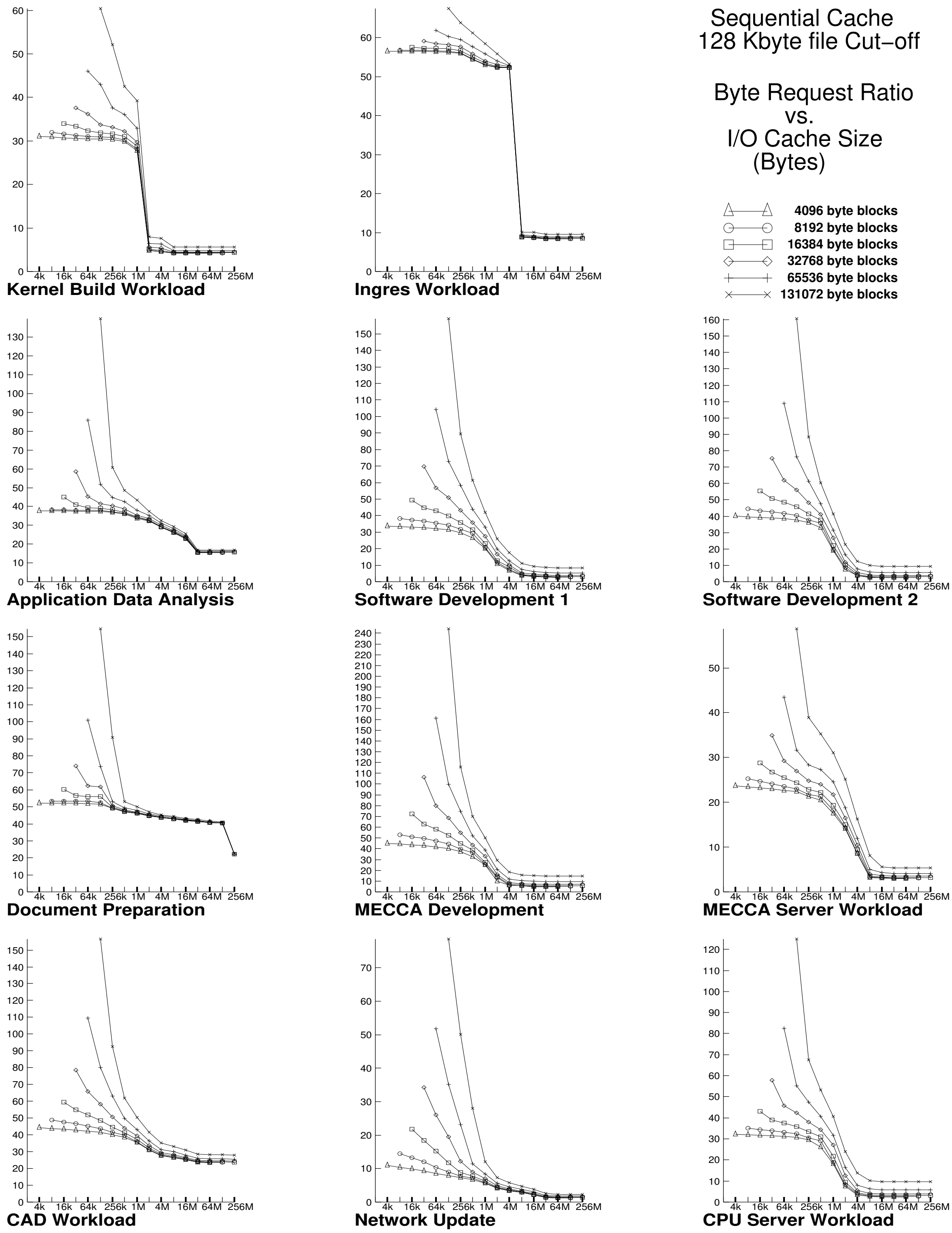

Figure C.10: Sequential Cache Traffic a File Cut-off of 128 Kb (relative)

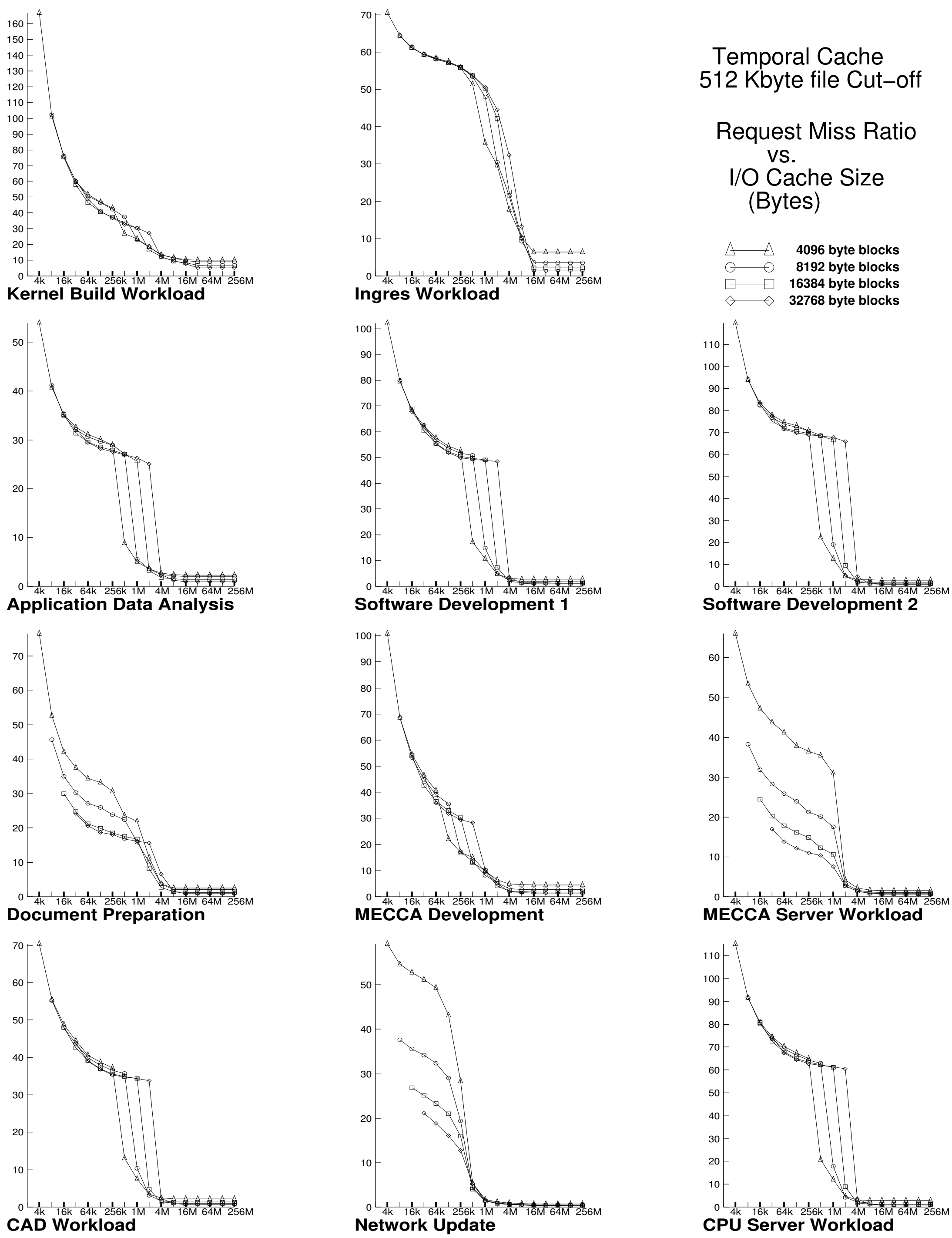

Figure C.11: Temporal Cache Request Misses for a File Cut-off of 512 Kb (relative)

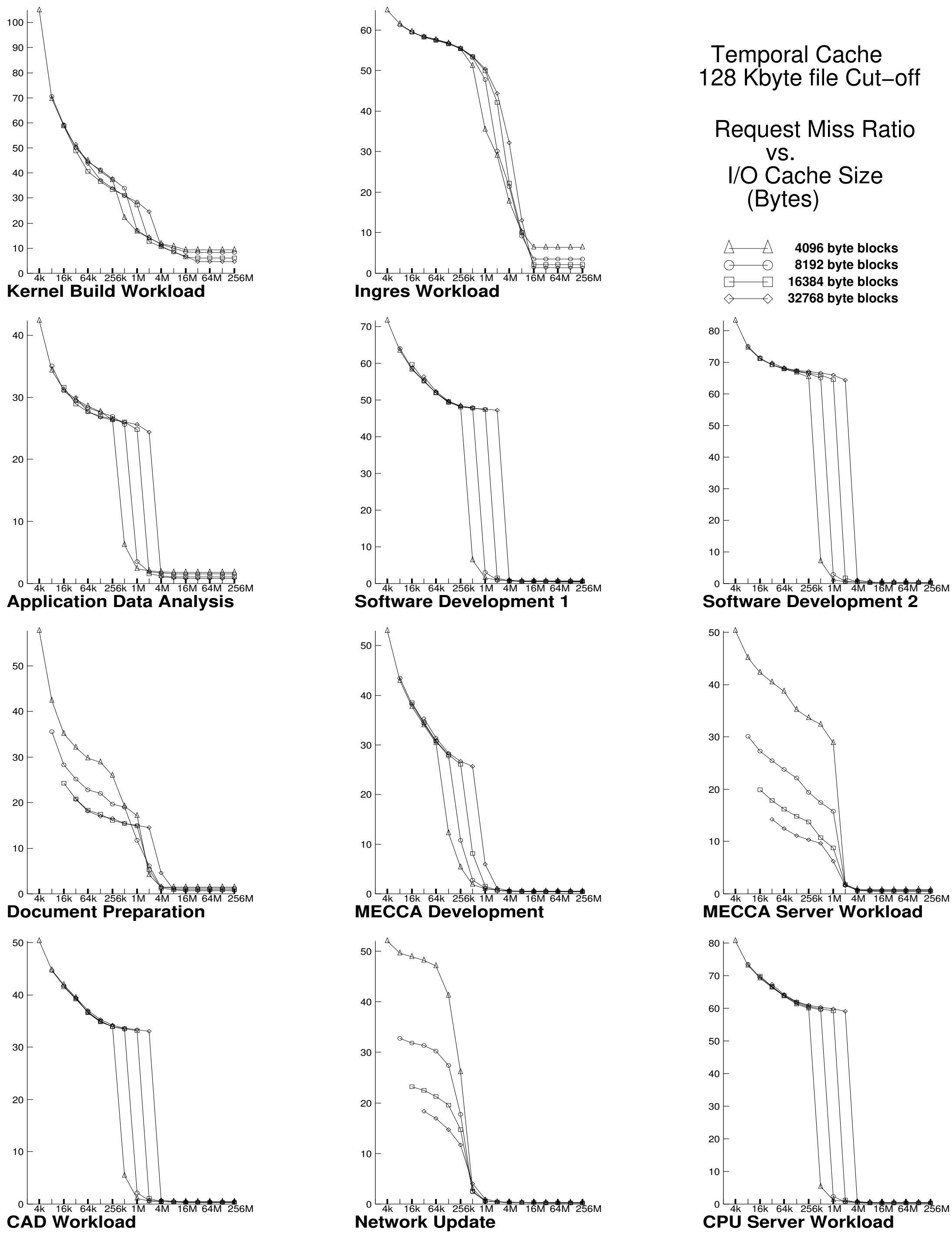

Figure C.12: Temporal Cache Request Misses for a File Cut-off of 128 Kb (relative)

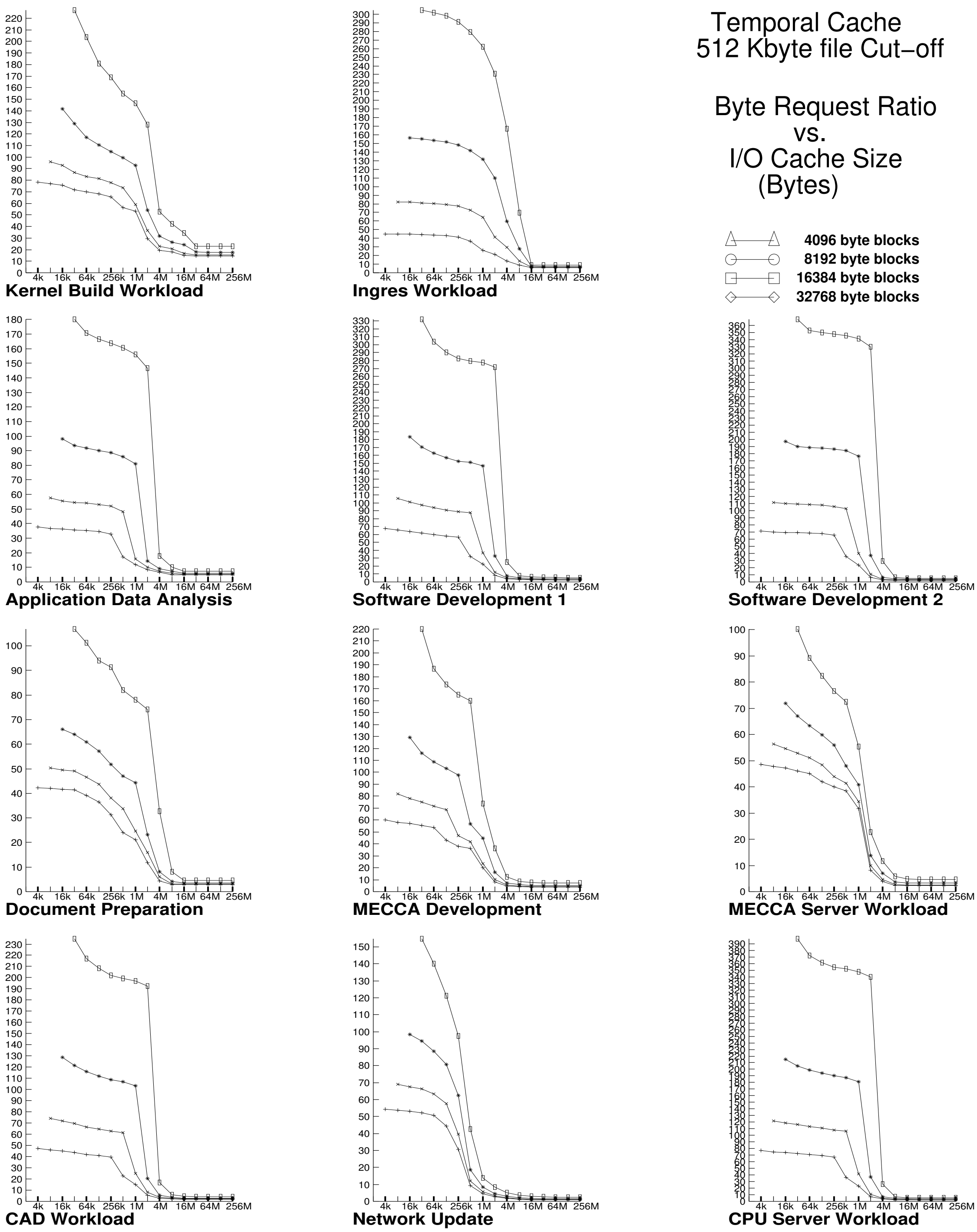

Figure C.13: Temporal Cache Traffic for a file cut-off of 512 Kbytes (relative).

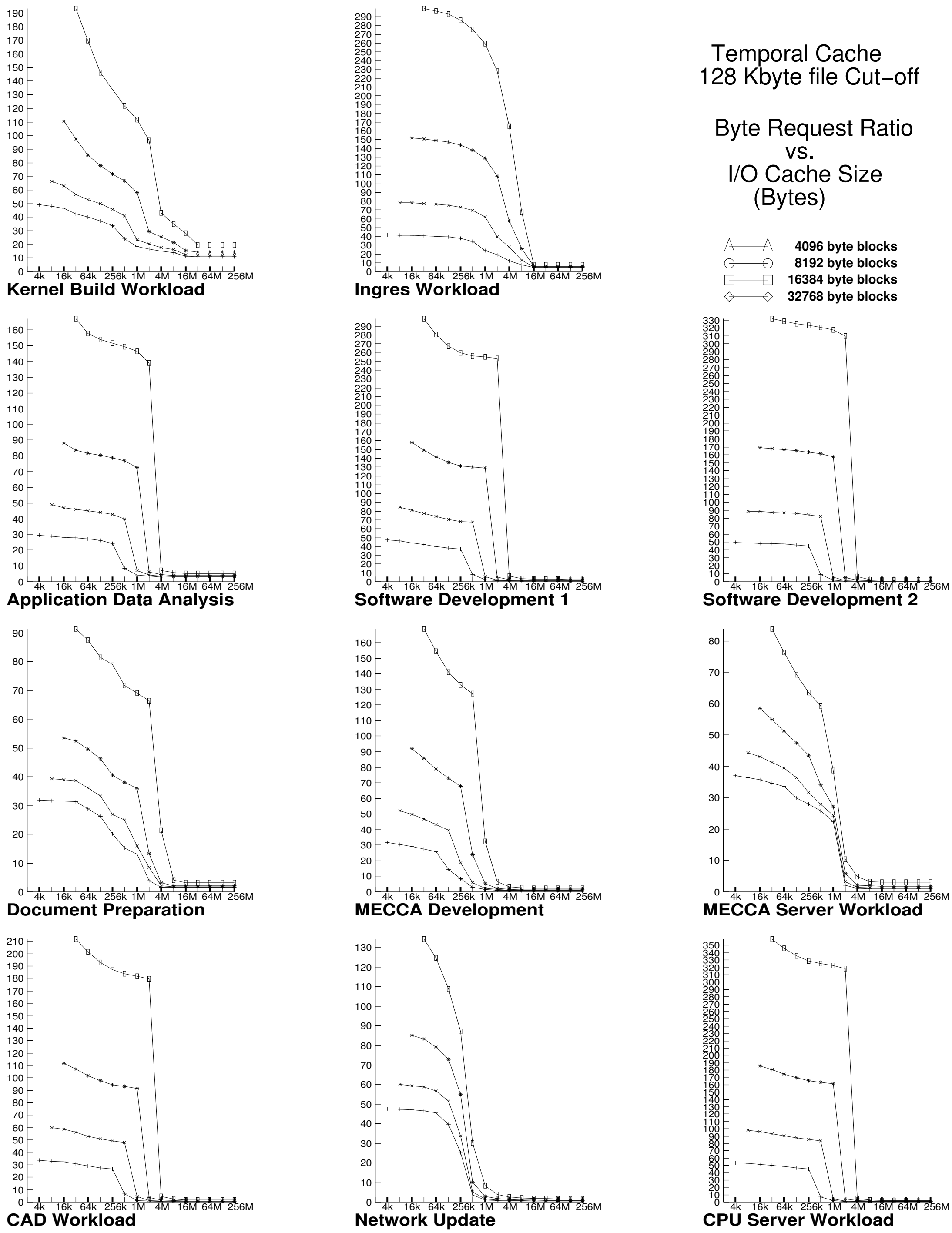

Figure C.14: Temporal Cache Traffic for a File Cut-off of 128 Kbytes (relative)

Appendix D

Variable Attribute Cache Data

This appendix presents attribute cache results used to develop an attribute cache scheme that improves the resource utilization of the cache and, in doing so, reduces the read request misses. The scheme was designed based on an evaluation of the overall workloads. It is neither an optimal solution given the cache partition requirement, nor an optimal solution given the set of experiments simulated. Many cache choices depend on the expected workload. The goal was to pick a simple scheme that works well over a broad range of workloads and for many potential disk or network systems.

D.1 Baseline for Comparison

A Unix style cache will be the baseline comparison for attribute caches. The Unix baseline allocates 32 Kbytes for inodes, and divides the rest of the cache into 4-Kbyte blocks. It is fully associative, and has LRU replacement. The cache allocates writes.

Figure D.1 shows the baseline read request miss ratio for all the workloads. The four sample workloads featured in the remainder of the chapter have solid symbols for highlighting. The data separates naturally into three regions of cache operation. Caches less than 64 or 128 Kbytes have high miss ratios, and increasing the cache size dramatically reduces the read request miss ratio. The region between 128 Kbytes and 8 Mbytes captures the data and executable working sets for most of the workloads. Extending the cache size beyond 8 Mbytes captures very large data sets, like that found in Ingres, and captures reuse of large runs. Each region has unique constraints and workload behavior, and will be evaluated individually. The boundaries between the three regions are not rigid, but for simplicity the regions are defined as follows:

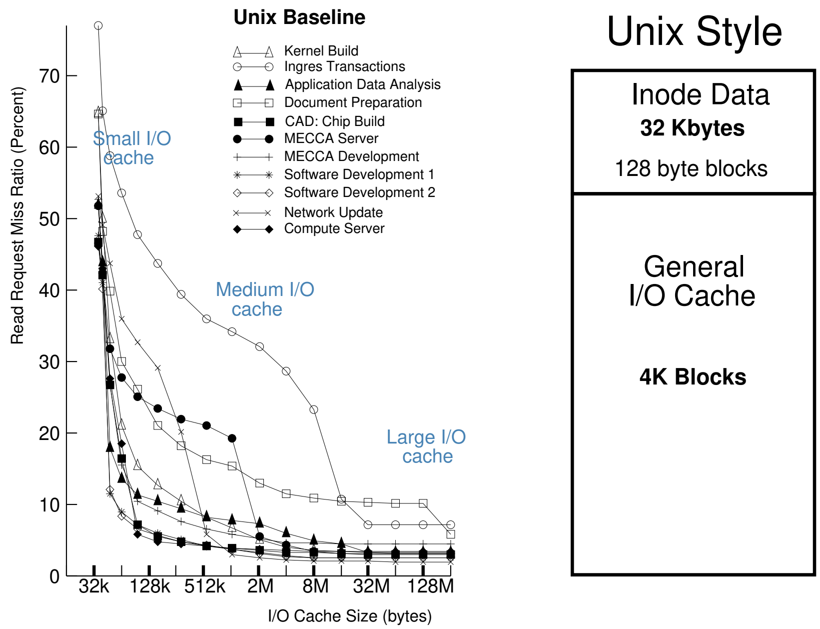

Figure D.1: Unix style cache read request miss ratio.

Small I/O Cache: 128 Kbytes or less.
Medium I/O Cache: 256 Kbytes to 4 Mbytes.
Large I/O Cache: 8 Mbytes or more.

D.2 Attribute Cache Design Parameters

Fixed-size subcaches allow the designer to evaluate many potential attribute cache configurations. The simulator handles each subcache independently, producing results for a range of cache sizes on each run. A post-simulation phase combines subcache results to determine the total cache read request misses. Figure D.2 shows the cache configurations used to demonstrate the usefulness of attributes and to determine how subcache splits should vary with total cache size. Four separate subcaches are used to construct two-category and three-category attribute caches.

The inode and directory subcache, also called the ID subcache, stores inodes and directories in small 128-byte blocks, packing many objects in a small space. The general I/O subcache is designed to capture both the temporal and sequential locality of non-ID requests. The temporal subcache caches the bulk of the datafile and executable references; its size determines whether the cache can capture the whole workload working set. The sequential cache captures large sequentially accessed objects or large objects with very low expected reuse. All the results shown in this chapter use a 512-Kbyte file cut-off to split file references between the temporal and sequential subcaches.

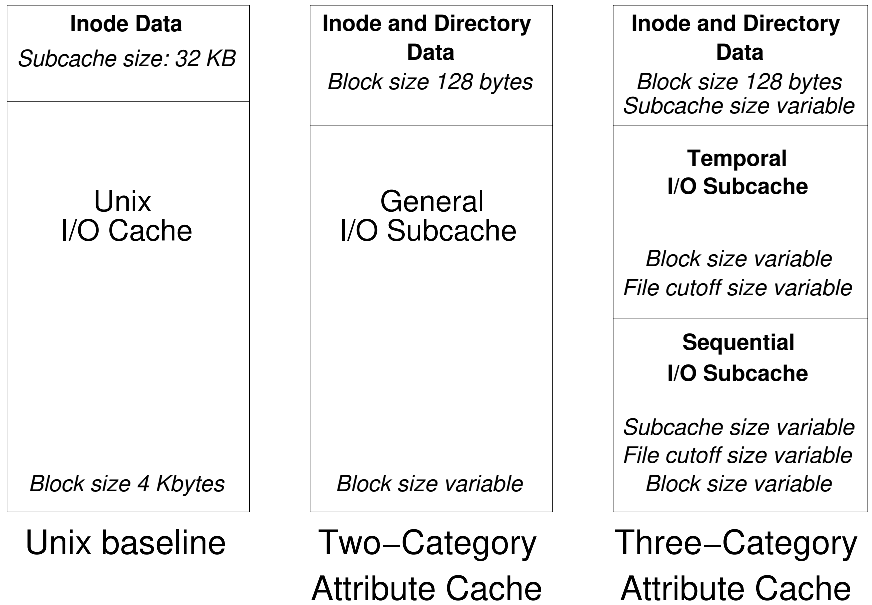

Figure D.2: Logical configurations for attribute I/O caches.

To concisely describe attribute cache organizations requires a notation convention. The important characteristics of each subcache are the cache size, the block size, and the cache category. Each subcache will be described as follows:

Cache_size/Block_size Cache Category

A string of subcache definitions describes a complete attribute cache configuration. For example, a two-category attribute cache with a 64K inode and directory cache and a general cache with 8-Kbyte blocks is described as follows:

64K/128 ID, x/8K General

The ID cache is fixed at 64-Kbytes and the General cache size determines the total cache size. A 128 Kbyte I/O cache would have a 64 Kbyte general cache, where as a 1 Mbyte cache would have a 960 Kbyte (1 MB less 64 Kbyte) general cache. From the size and blocks size it is easy to determine the number of cache blocks. The 64K/128 ID cache has 512 blocks.

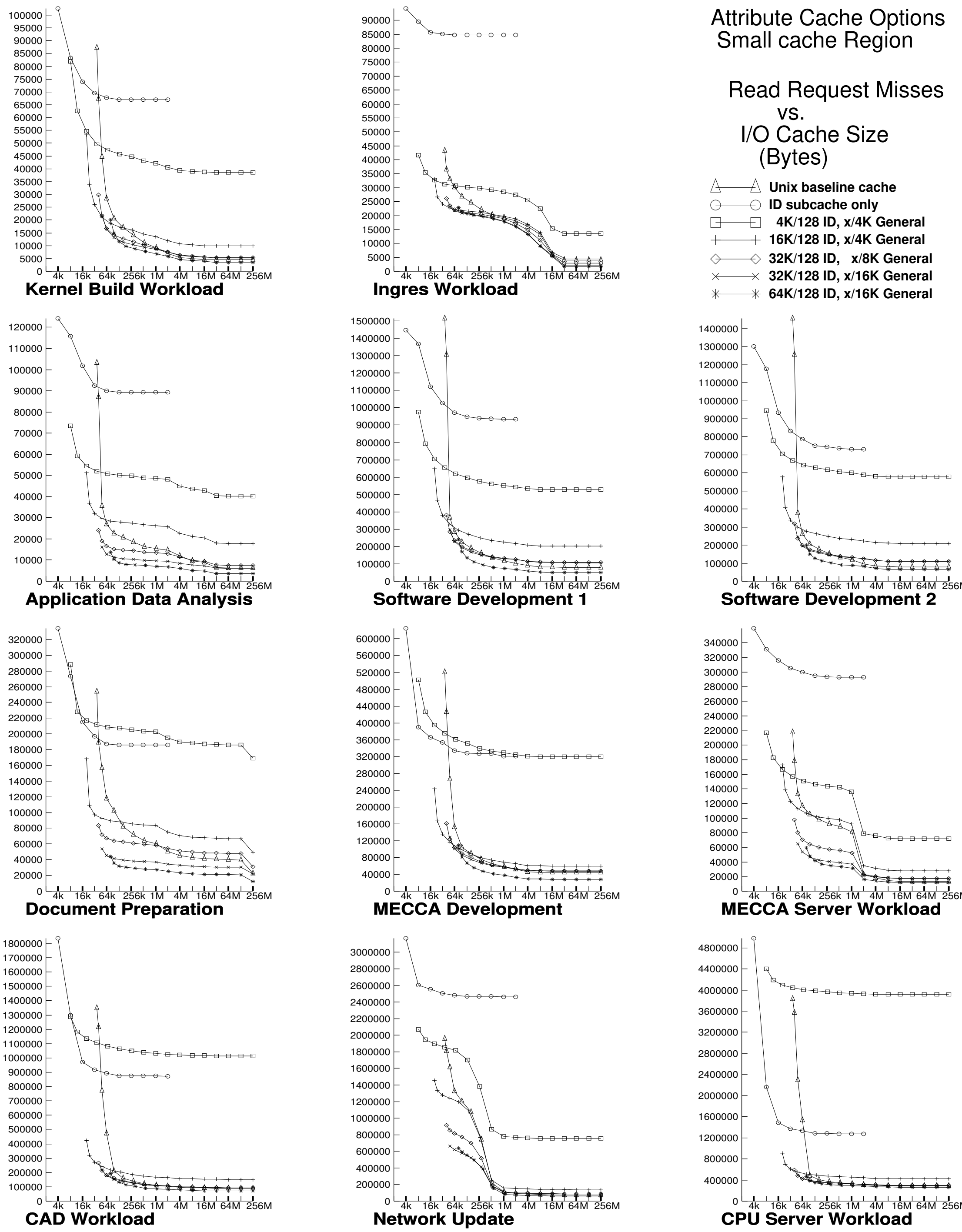

Figure D.3: Attribute cache option for small I/O caches.

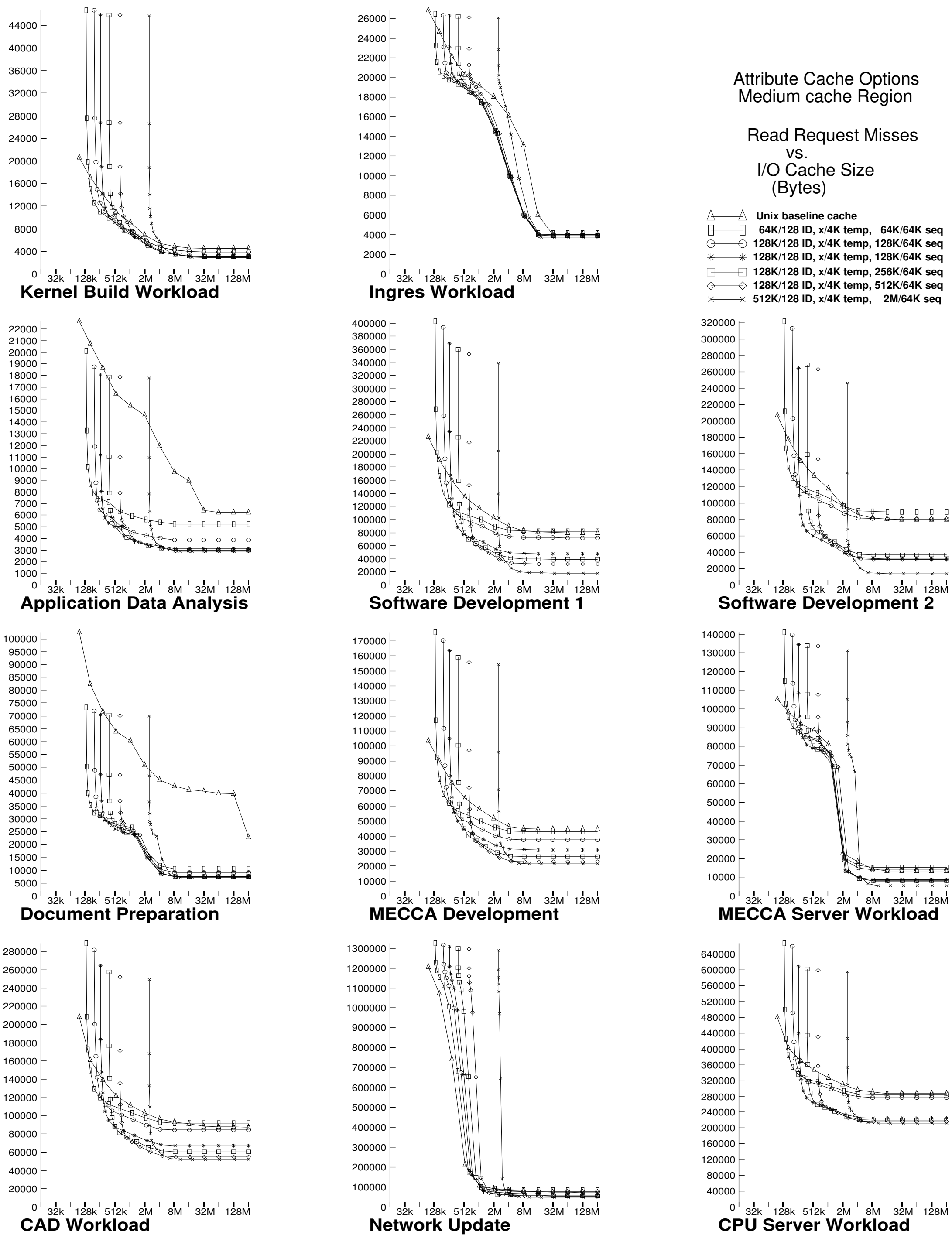

Figure D.4: Attribute cache option for medium I/O caches.

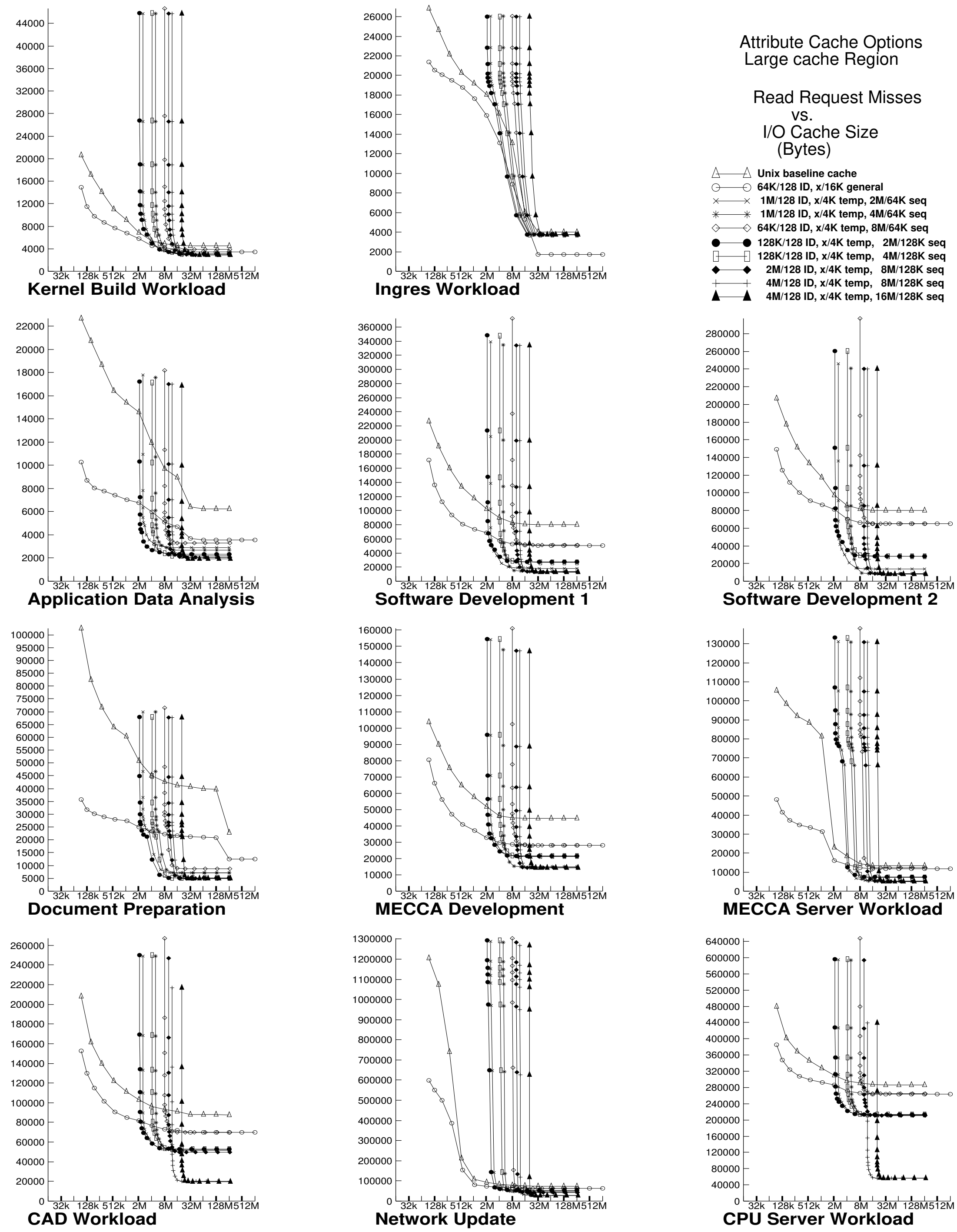

Figure D.5: Attribute cache option for large I/O caches.

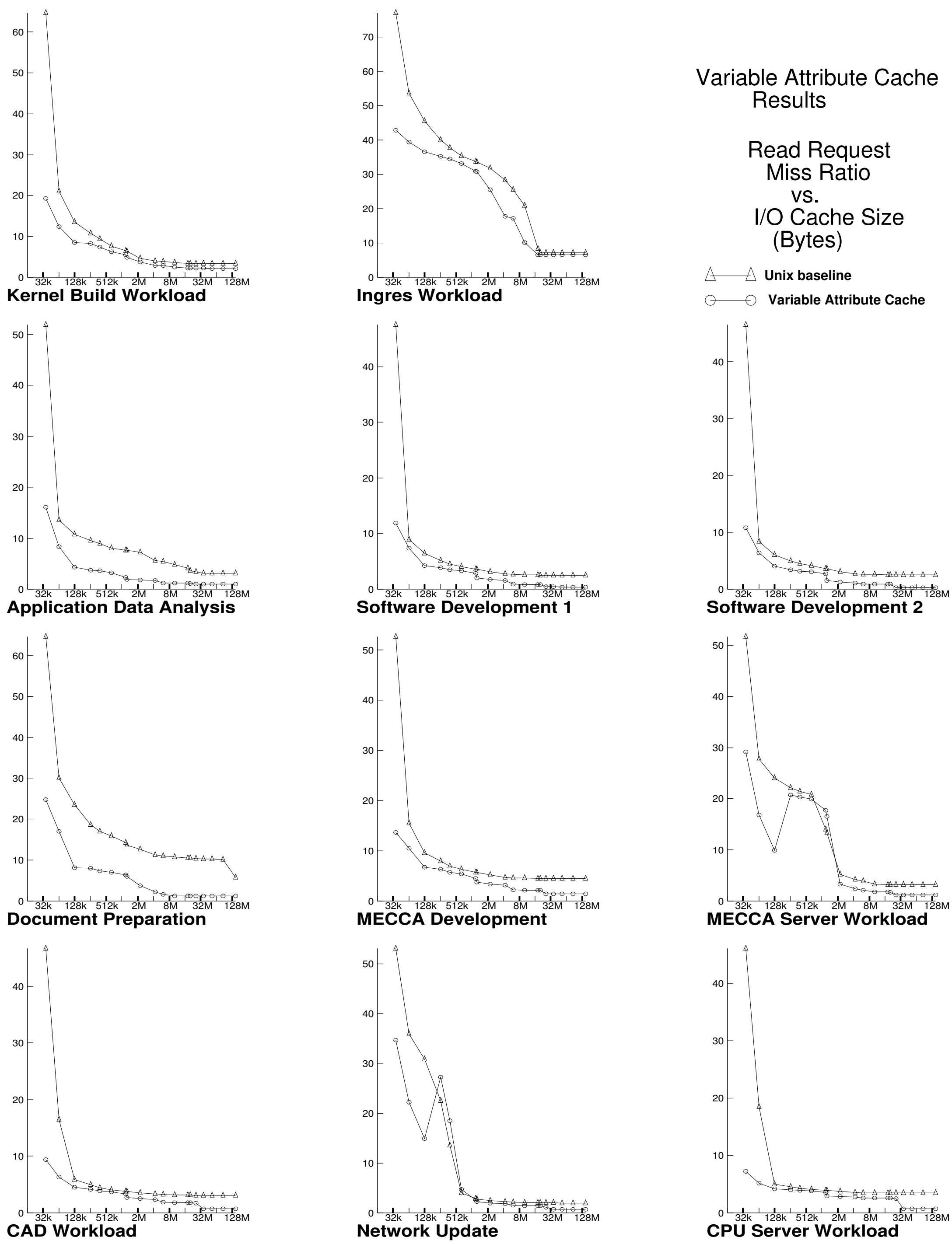

Figure D.6: Variable attribute cache read request miss ratio.

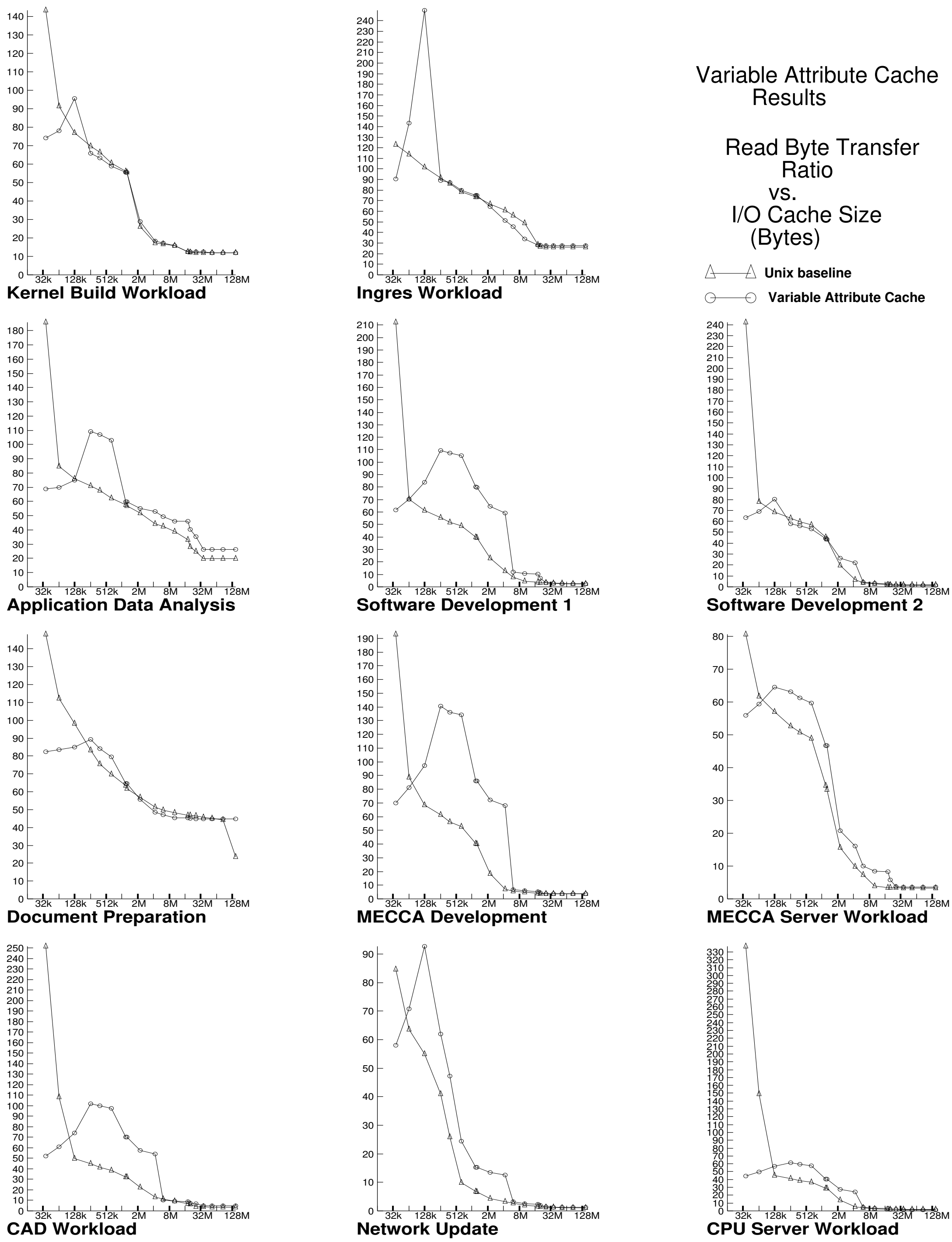

Figure D.7: Variable attribute cache read byte transfer ratio.

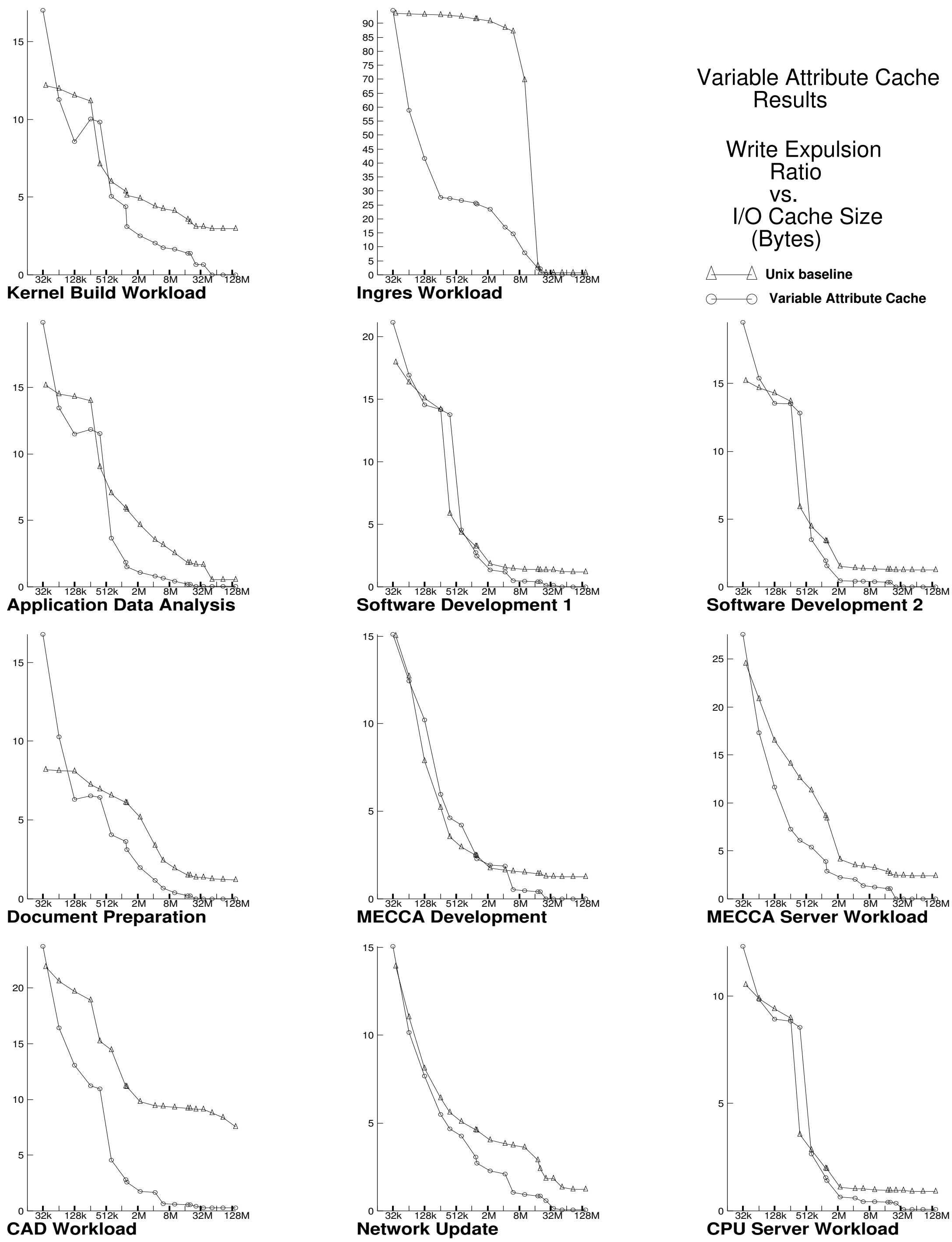

Figure D.8: Variable attribute cache write expulsion ratio.

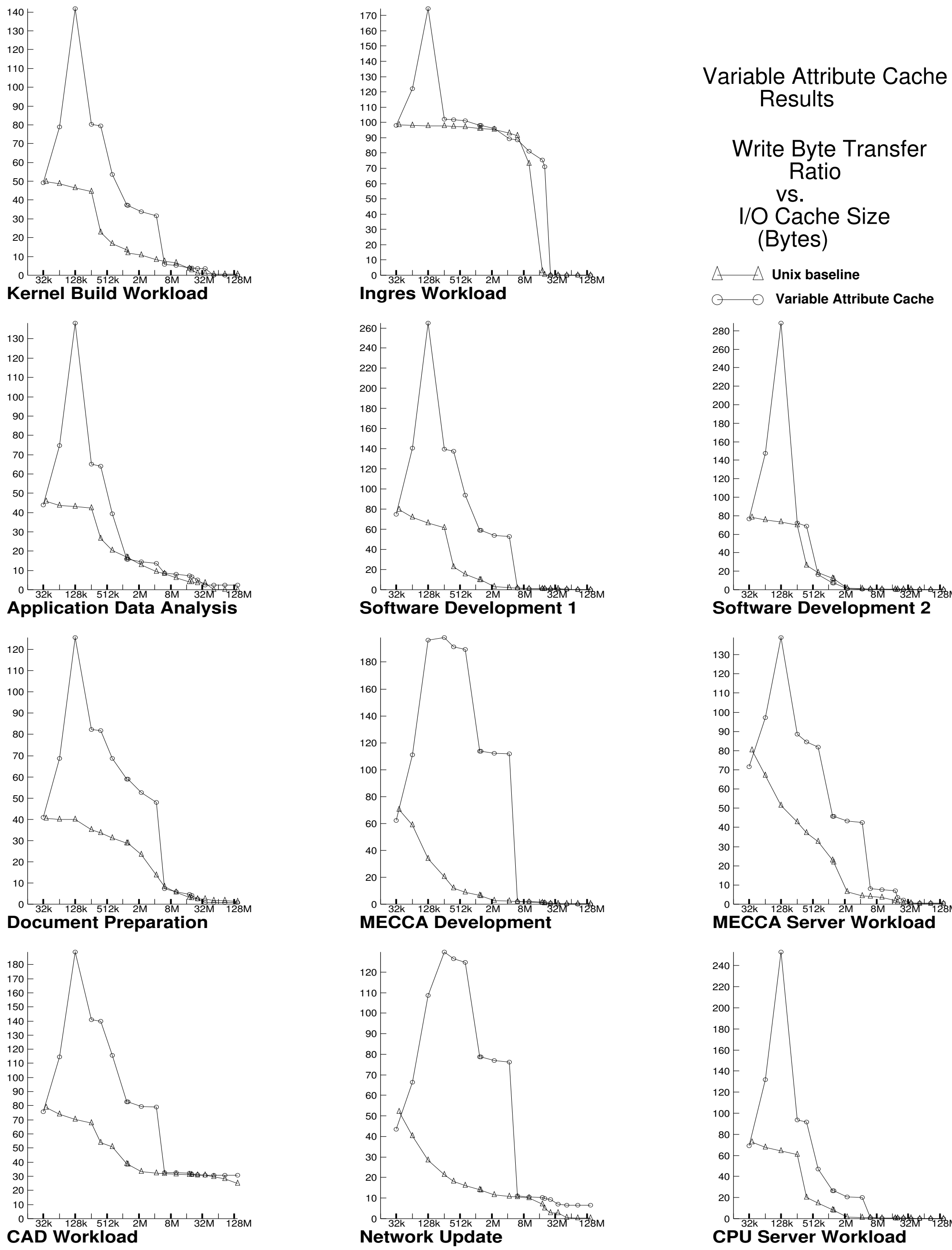

Figure D.9: Variable attribute cache write byte transfer ratio.

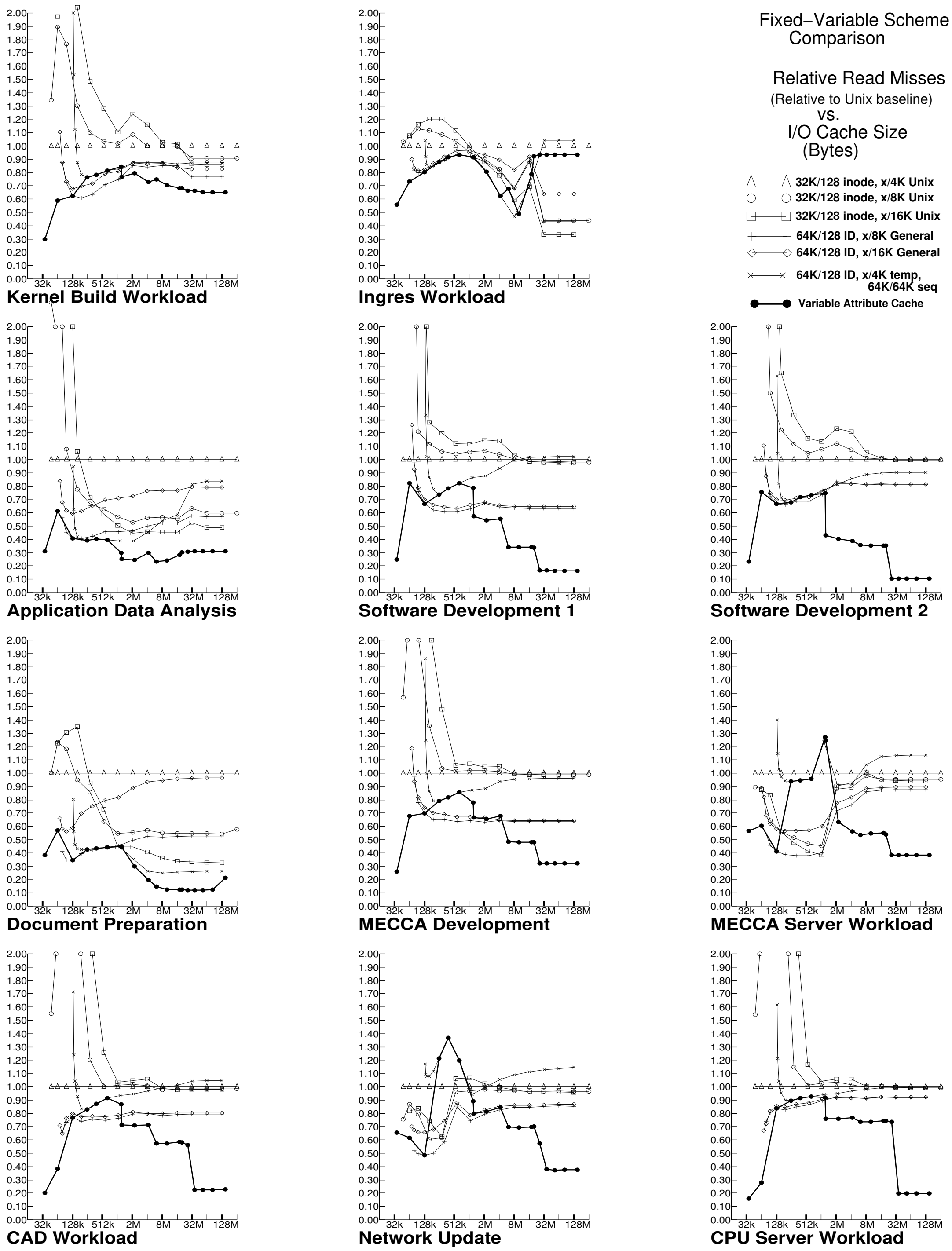

Figure D.10: Comparison between fixed schemes and the variable attribute cache.

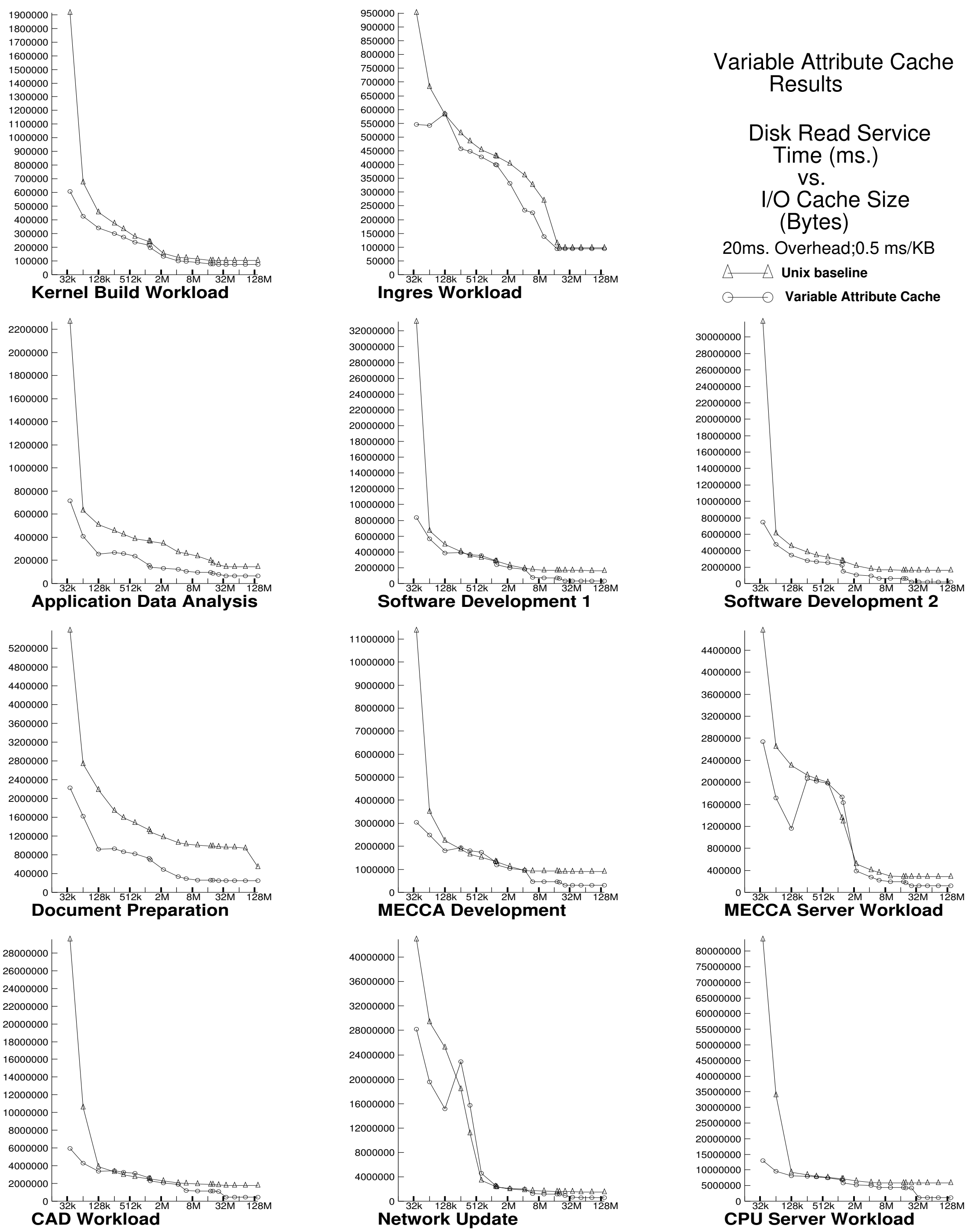

Figure D.11: Read service time.

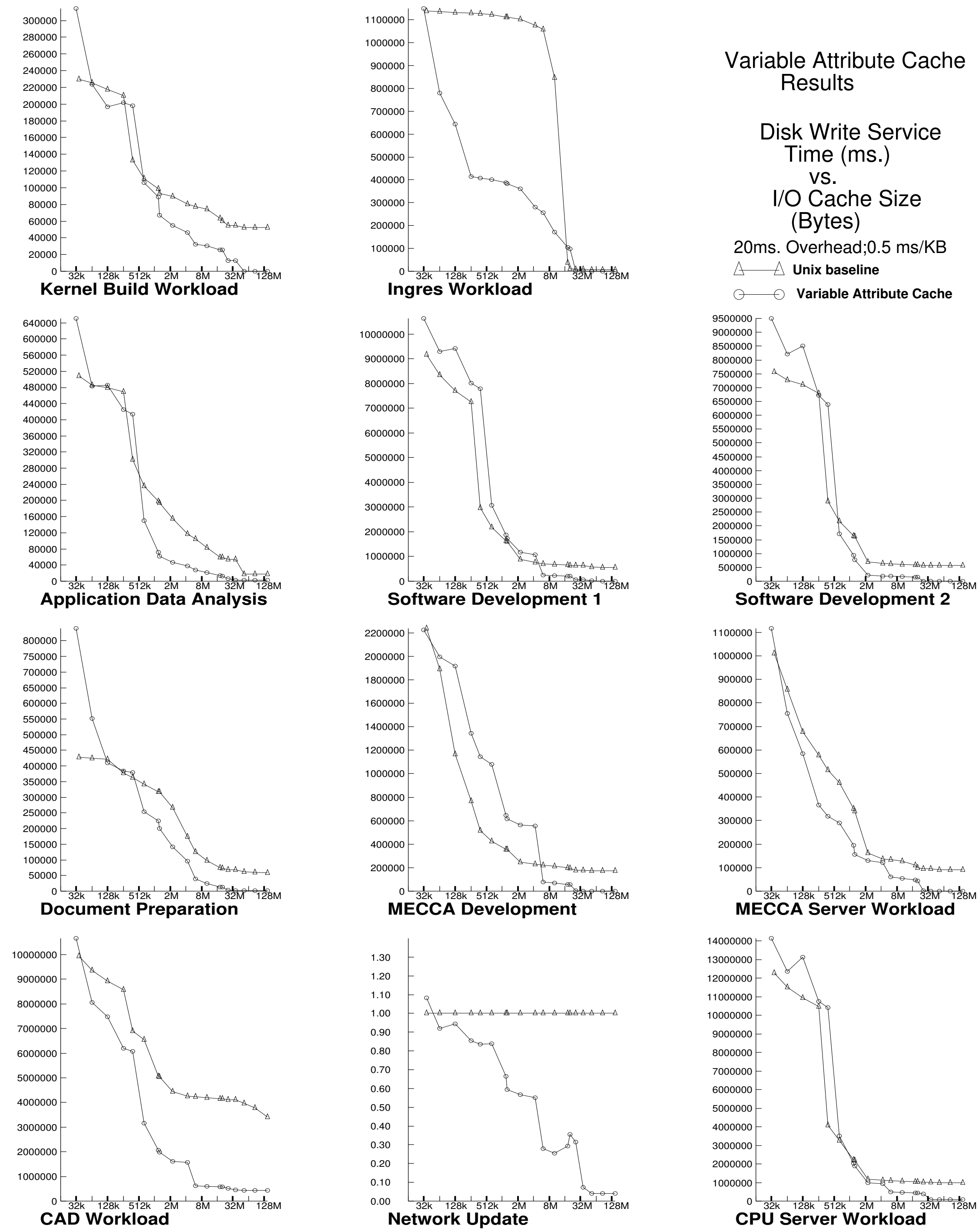

Figure D.12: Write service time.

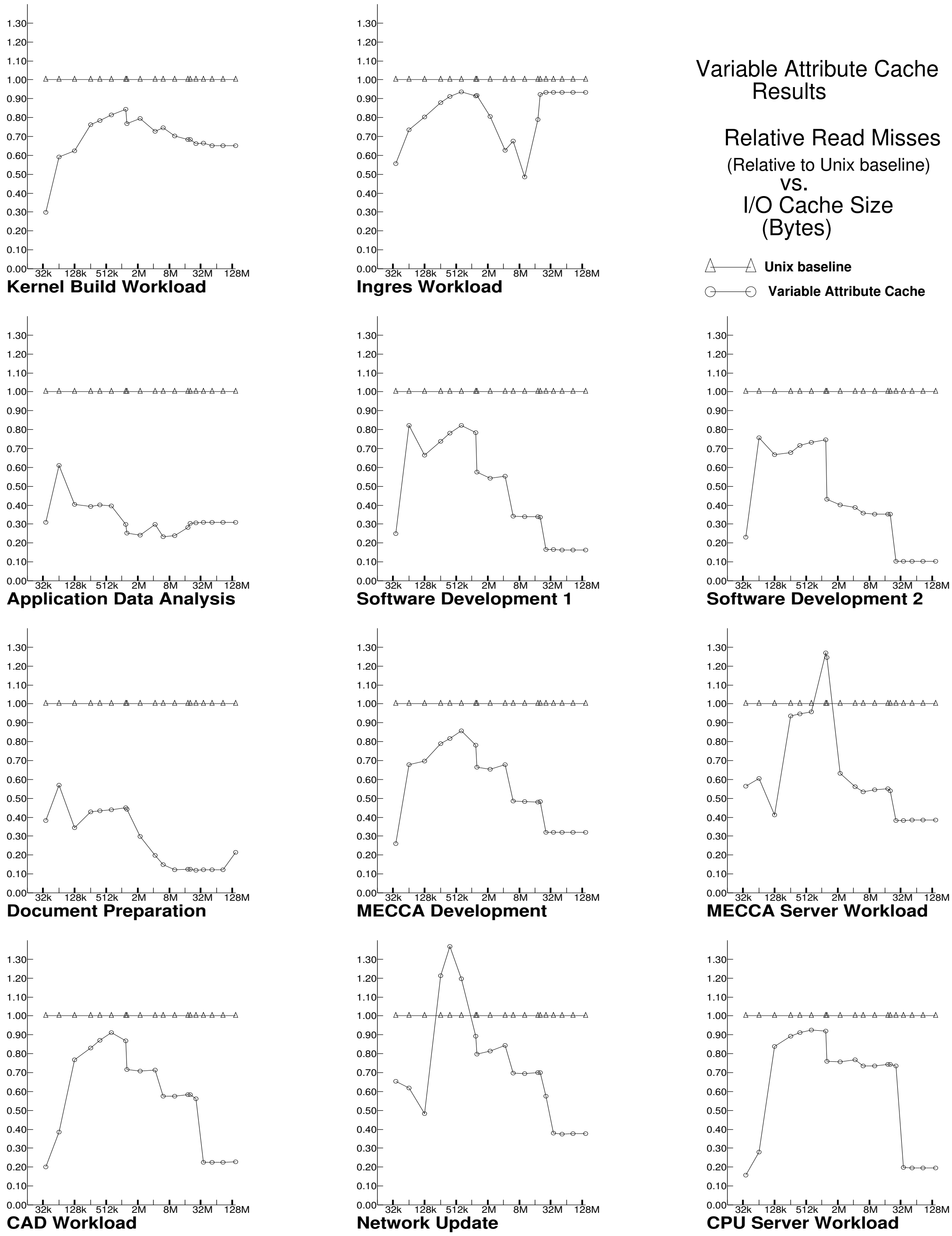

Figure D.13: Relative read requests misses.

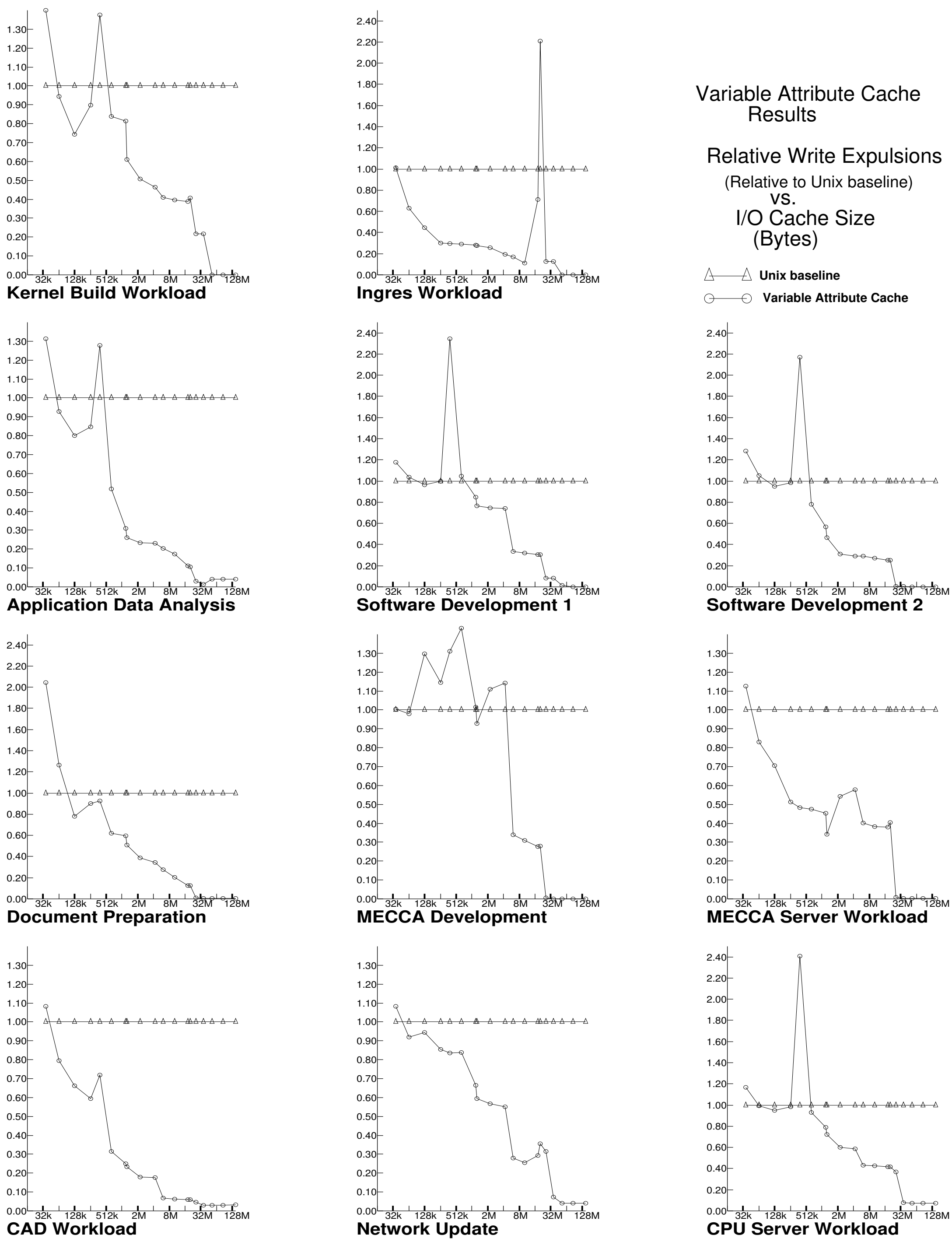

Figure D.14: Relative write expulsions.

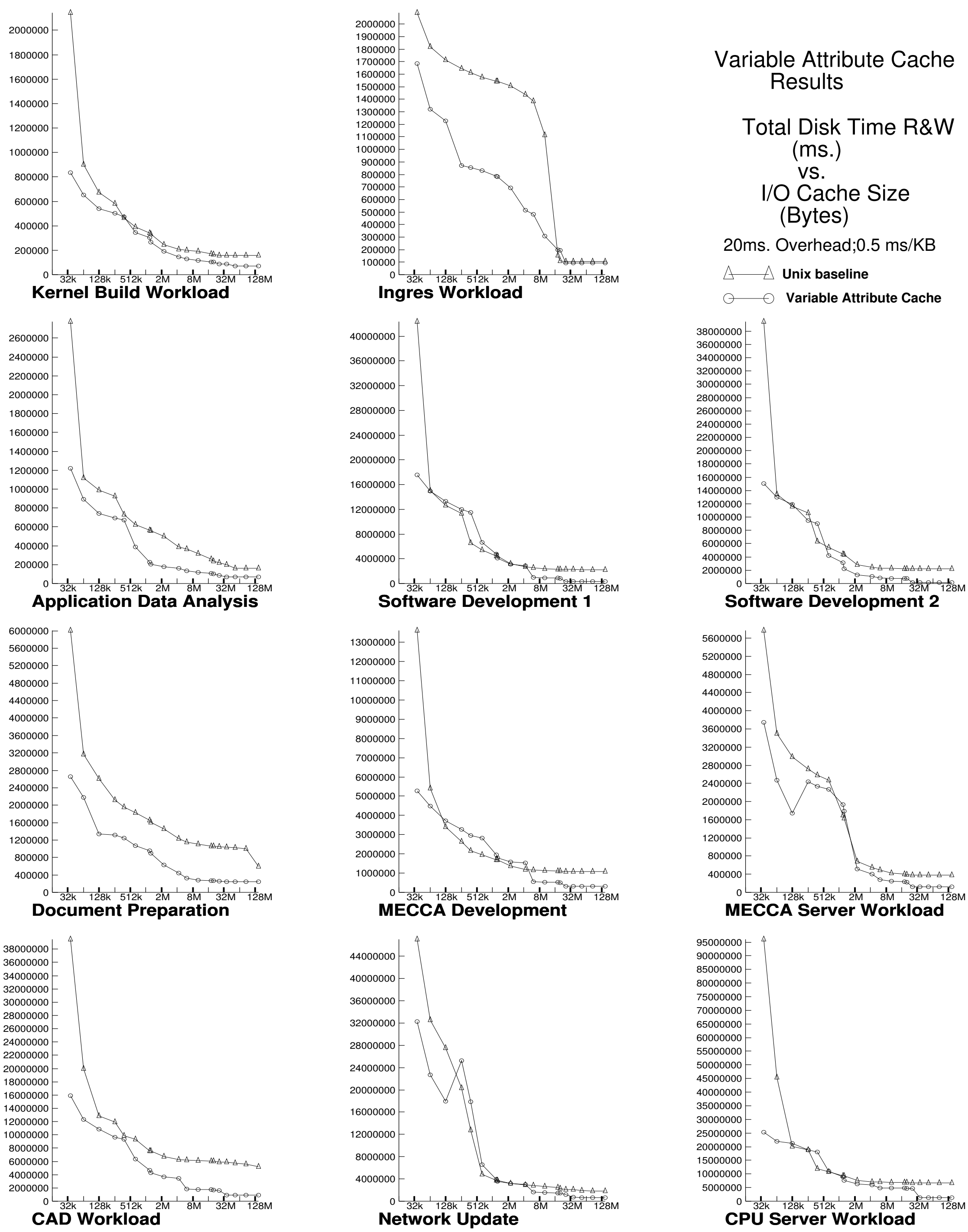

Figure D.15: Total disk service time.

Bibliography

[BKLW90] Anita Borg, R. E. Kessler, Georgia Lazana, and David Wall. Long address traces from RISC machines: Generation and analysis. In *The 17th Annual International Symposium on Computer Architecture*, pages 270–279. IEEE Computer Society Press, May 1990.

[Bor92] Anita Borg. MECCA: a message-enabled communication and information system. Technical Report Technical Note TN-10, Digital Network Systems Laboratory, November 1992.

[CBJ92] J. B. Chen, A. Borg, and N. P. Jouppi. A simulation based study of TLB performance. In *The 19th Annual International Symposium on Computer Architecture*, pages 114–123. IEEE Computer Society Press, May 1992.

[Cop] Digital Equipment Coproration. Configuration file maintenance. In *USENIX System and Network Management*, volume 2.

[DM92] Jeremy Dion and Louis Monier. Design tools for BIPS-0. Technical Report Technical Note TN-32, Digital Western Research Laboratory, December 1992.

[SE94] Amitabh Srivastava and Alan Eustace. ATOM: a system for building customized program analysis tools. Technical Report Research Report 94/2, Digital Western Research Laboratory, March 1994.

[Sto86] M. Stonebraker. *The Ingres Papers*. Addison-Wesley, 1986.