



Contents

PART ONE		
Observability Methods		1
Chapter 1	Introduction to Observability Tools	3
	1.1 Observability Tools	4
	1.1.1 Kstat Tools	5
	1.1.2 Process Tools	5
	1.1.3 Forensic Tools	6
	1.2 Drill-down Analysis	7
	1.3 About This Section	8
	1.3.1 Chapter Layout	8
Chapter 2	CPUs	11
	2.1 Tools for CPU Analysis	11
	2.2 vmstat Tool	13
	2.3 CPU Utilization	14
	2.4 CPU Saturation	15
	2.5 psrinfo Command	15
	2.6 uptime Command	15
	2.7 sar Command	16

2.7.1	<code>sar</code> Default Output	17
2.7.2	<code>sar -q</code>	18
2.7.3	Capturing Historical Data	19
2.8	Clock Tick Woes	19
2.9	<code>mpstat</code> Command	21
2.10	Who Is Using the CPU?	23
2.11	CPU Run Queue Latency	24
2.12	CPU Statistics Internals	26
2.12.1	<code>usr, sys, idl</code> Times	26
2.12.2	Load Averages	28
2.12.3	<code>pr_pctcpu</code> Field	29
2.13	Using DTrace to Explain Events From Performance Tools	29
2.14	DTrace Versions of <code>runq-sz, %runocc</code>	31
2.15	DTrace Probes for CPU States	34
Chapter 3	Processes	37
3.1	Tools for Process Analysis	37
3.2	Process Statistics Summary: <code>prstat</code>	39
3.2.1	Thread Summary: <code>prstat -L</code>	40
3.2.2	Process Microstates: <code>prstat -m</code>	41
3.2.3	Sorting by a Key: <code>prstat -s</code>	42
3.2.4	User Summary: <code>prstat -t</code>	43
3.2.5	Project Summary: <code>prstat -J</code>	43
3.2.6	Zone Summary: <code>prstat -Z</code>	44
3.3	Process Status: <code>ps</code>	44
3.3.1	<code>/usr/bin/ps</code> Command	44
3.3.2	<code>/usr/ucb/ps</code>	47
3.4	Tools for Listing and Controlling Processes	48
3.4.1	Process tree: <code>ptree</code>	48
3.4.2	Grepping for Processes: <code>pgrep</code>	49
3.4.3	Killing Processes: <code>pkill</code>	49
3.4.4	Temporarily Stop a Process: <code>pstop</code>	50
3.4.5	Making a Process Runnable: <code>prun</code>	50
3.4.6	Wait for Process Completion: <code>pwait</code>	50
3.4.7	Reap a Zombie Process: <code>preap</code>	50
3.5	Process Introspection Commands	50

3.5.1	Process Stack: <code>pstack</code>	51
3.5.2	Process Memory Map: <code>pmap -x</code>	52
3.5.3	Process File Table: <code>pfiles</code>	53
3.5.4	Execution Time Statistics for a Process: <code>ptime</code>	54
3.5.5	Process Signal Disposition: <code>psig</code>	54
3.5.6	Process Libraries: <code>pldd</code>	55
3.5.7	Process Flags: <code>pflags</code>	55
3.5.8	Process Credentials: <code>pcred</code>	55
3.5.9	Process Arguments: <code>pargs</code>	56
3.5.10	Process Working Directory: <code>pwdx</code>	56
3.6	Examining User-level Locks in a Process	57
3.7	Tracing Processes	58
3.7.1	Using <code>truss</code> to Trace Processes	58
3.7.2	Using <code>apptrace</code> to Trace Processes	60
3.7.3	Using DTrace to Trace Process Functions	61
3.7.4	Using DTrace to Aggregate Process Functions	62
3.7.5	Using DTrace to Peer Inside Processes	63
3.7.6	Using DTrace to Sample Stack Backtraces	64
3.8	Java Processes	65
3.8.1	Process Stack on a Java Virtual Machine: <code>pstack</code>	66
3.8.2	JVM Profiling	66
3.8.3	Tuning Java Garbage Collection	68
3.8.4	Using DTrace on Java Applications	70
Chapter 4	Disk Behavior and Analysis	73
4.1	Terms for Disk Analysis	73
4.2	Random vs. Sequential I/O	75
4.2.1	Demonstration of Sequential Disk Activity	75
4.2.2	Demonstration of Random Disk Activity	76
4.3	Storage Arrays	76
4.4	Sector Zoning	77
4.5	Max I/O Size	78
4.6	<code>iostat</code> Utility	79
4.7	Disk Utilization	80
4.8	Disk Saturation	82
4.9	Disk Throughput	82

4.10	iostat Reference	82
4.10.1	iostat Default	84
4.10.2	iostat -D	85
4.10.3	iostat -l n	86
4.10.4	iostat -n	87
4.10.5	iostat -x	87
4.10.6	iostat -p, -P	88
4.10.7	iostat -e	88
4.10.8	iostat -E	89
4.11	Reading iostat	89
4.11.1	Event Size Ratio	89
4.11.2	Service Times	90
4.12	iostat Internals	92
4.13	sar -d	94
4.14	Trace Normal Form (TNF) Tracing for I/O	95
4.15	DTrace for I/O	95
4.15.1	io Probes	96
4.15.2	I/O Size One-Liners	97
4.15.3	A More Elaborate Example	98
4.15.4	I/O Size Aggregation	99
4.15.5	I/O Seek Aggregation	102
4.15.6	I/O File Names	104
4.16	Disk I/O Time	105
4.16.1	Simple Disk Event	105
4.16.2	Concurrent Disk Events	107
4.16.3	Adaptive Disk I/O Time Algorithm	108
4.16.4	Other Response Times	109
4.16.5	Time by Layer	109
4.17	DTraceToolkit Commands	110
4.17.1	iotop Script	110
4.17.2	iosnoop Script	113
4.17.3	Plotting Disk Activity	114
4.17.4	Plotting Concurrent Activity	116
4.18	DTraceTazTool	117
Chapter 5	File Systems	119

5.1	Layers of File System and I/O	119
5.2	Observing Physical I/O	121
5.3	File System Latency	123
5.4	Causes of Read/Write File System Latency	126
5.4.1	Disk I/O wait	127
5.4.2	Block or Metadata Cache Misses	127
5.4.3	I/O Breakup	127
5.4.4	Locking in the File System	128
5.4.5	Metadata Updates	129
5.5	Observing File System “Top End” Activity	129
5.6	File System Caches	130
5.6.1	Page Cache	132
5.6.2	Bypassing the Page Cache With Direct I/O	137
5.6.3	The Directory Name Lookup Cache	139
5.6.4	Block Buffer Cache	142
5.6.5	UFS Inode Cache	144
5.6.6	Monitoring UFS Caches With <code>fcachestat</code>	147
5.7	NFS Statistics	147
5.7.1	NFS Client Statistics: <code>nfsstat -c</code>	147
5.7.2	NFS Server Statistics: <code>nfsstat -s</code>	149
Chapter 6	Memory	151
6.1	Tools for Memory Analysis	151
6.2	<code>vmstat(1M)</code> Command	153
6.3	Types of Paging	154
6.3.1	File I/O Paging: “Good” Paging	155
6.3.2	Anonymous Memory Paging: “Bad” Paging	157
6.3.3	Per-Zone Paging Statistics	158
6.4	Physical Memory Allocation	158
6.4.1	Total Physical Memory	158
6.4.2	Free Memory	158
6.4.3	Using the <code>memstat</code> Command in <code>mdb</code>	159
6.5	Relieving Memory Pressure	161
6.6	Scan Rate As a Memory Health Indicator	162
6.6.1	Using <code>prstat</code> to Estimate Memory Slowdowns	164
6.7	Process Virtual and Resident Set Size	165

6.8	Using <code>pmap</code> to Inspect Process Memory Usage	166
6.9	Calculating Process Memory Usage with <code>ps</code> and <code>pmap</code>	166
6.10	Displaying Page Size Information with <code>pmap</code>	170
6.11	Using DTrace for Memory Analysis	171
6.11.1	Using DTrace to Estimate Memory Slowdowns	173
6.12	Obtaining Memory Kstats	174
6.13	Using the Perl <code>kstat</code> API to Look at Memory Statistics	175
6.14	System Memory Allocation Kstats	177
6.15	Kernel Memory With <code>kstat</code>	178
6.16	System Paging Kstats	179
6.17	Observing MMU Performance Impact With <code>trapstat</code>	181
6.18	Swap Space	182
6.18.1	Swap Allocation	183
6.18.2	Swap Statistics	185
6.18.3	Swap Summary: <code>swap -s</code>	185
6.18.4	Listing Physical Swap Devices: <code>swap -l</code>	186
6.18.5	Determining Swapped-out Threads	186
6.18.6	Monitoring Physical Swap Activity	187
6.18.7	MemTool <code>prtswap</code>	188
6.18.8	Display of Swap Reservations With <code>pmap</code>	189
Chapter 7	Networks	193
7.1	Terms for Network Analysis	193
7.2	Packets Are Not Bytes	195
7.3	Network Utilization	196
7.4	Network Saturation	197
7.5	Network Errors	197
7.6	Misconfigurations	197
7.7	Systemwide Statistics	198
7.7.1	<code>netstat</code> Command	198
7.7.2	<code>kstat</code> Command	200
7.7.3	<code>nx.se</code> Tool	201
7.7.4	<code>nicstat</code> Tool	201
7.7.5	SNMP	202
7.7.6	<code>checkcable</code> Tool	203
7.7.7	<code>ping</code> Tool	205

7.7.8	tracertool Tool	206
7.7.9	snoop Tool	206
7.7.10	TTCP	207
7.7.11	pathchar Tool	208
7.7.12	ntop Tool	208
7.7.13	NFS Client Statistics: <code>nfsstat -c</code>	209
7.7.14	NFS Server Statistics: <code>nfsstat -s</code>	210
7.8	Per-Process Network Statistics	210
7.8.1	tcptop Tool	211
7.8.2	tcpsnoop Tool	212
7.9	TCP Statistics	213
7.9.1	TCP Statistics Internals	213
7.9.2	TCP Statistics From Kstat	215
7.9.3	TCP Statistics Reference	215
7.9.4	TCP Statistics From DTrace	217
7.10	IP Statistics	217
7.10.1	IP Statistics Internals	218
7.10.2	IP Statistics From Kstat	218
7.10.3	IP Statistics Reference	218
7.10.4	IP Statistics From DTrace	220
7.11	ICMP Statistics	220
7.11.1	ICMP Statistics From Kstat	221
7.11.2	ICMP Statistics Reference	221
7.11.3	ICMP Statistics From DTrace	222
7.11.4	Tracing Raw Network Functions	222
Chapter 8	Performance Counters	225
8.1	Introducing CPU Caches	225
8.2	cpustat Command	228
8.2.1	Cache hit ratio, Cache Misses	228
8.2.2	Listing PICs and Events	231
8.2.3	PIC Examples: UltraSPARC IIi	231
8.2.4	PIC Examples: The UltraSPARC T1 Processor	233
8.2.5	Event Multiplexing	235
8.2.6	Using cpustat With Multiple CPUs	235
8.2.7	Cycles per Instruction	236

8.2.8	PIC Examples: UltraSPARC IV	237
8.3	<code>cputrack</code> Command	237
8.4	<code>busstat</code> Command	239
8.4.1	Listing Supported Buses	239
8.4.2	Listing Bus Events	240
8.4.3	Monitoring Bus Events	241
8.4.4	Event Multiplexing	242
8.4.5	Example: UltraSPARC-T1	243
Chapter 9	Kernel Monitoring	245
9.1	Tools for Kernel Monitoring	245
9.2	Profiling the Kernel and Drivers	246
9.2.1	Profiling the Kernel with <code>lockstat -I</code>	246
9.3	Analyzing Kernel Locks	248
9.3.1	Adaptive Locks	249
9.3.2	Spin Locks	249
9.3.3	Reader/Writer Locks	249
9.3.4	Thread Locks	250
9.3.5	Analyzing Locks With <code>lockstat</code>	250
9.4	DTrace <code>lockstat</code> Provider	253
9.5	DTrace Kernel Profiling	255
9.6	Interrupt Statistics: <code>vmstat -i</code>	256
9.7	Interrupt Analysis: <code>intrstat</code>	256
PART TWO		
Observability Infrastructure		259
Chapter 10	Dynamic Tracing	261
10.1	Introduction to DTrace	261
10.2	The Basics	262
10.2.1	D Program Structure	265
10.2.2	Providers and Probes	266
10.2.3	Aggregations	269
10.2.4	Variables	273
10.2.5	Probe Arguments	277
10.2.6	Mixing Providers	280

10.2.7	Accessing Global Kernel Data	282
10.2.8	Assorted Actions of Interest	284
10.3	Inspecting Java Applications with DTrace	286
10.3.1	Inspecting Applications with the DTrace jstack Action	292
10.3.2	Adding Probes to Pre-Mustang Releases	294
10.4	DTrace Architecture	296
10.5	Summary	302
10.6	Probe Reference	302
10.6.1	The IO Provider	302
10.6.1.1	bufinfo_t structure	304
10.6.1.2	devinfo_t	305
10.6.1.3	fileinfo_t	306
10.6.2	Virtual Memory Provider Probes	307
10.6.3	The Sched Provider	309
10.6.3.1	Arguments	309
10.6.4	DTrace Lockstat Provider	314
10.6.4.1	Adaptive Lock Probes	315
10.6.4.2	Spin Lock Probes	316
10.6.4.3	Thread Locks	316
10.6.4.4	Readers/Writer Lock Probes	317
10.6.5	The Java Virtual Machine Provider	318
10.6.5.1	VM Lifecycle Probes	318
10.6.5.2	Thread Lifecycle Probes	318
10.6.5.3	Classloading Probes	319
10.6.5.4	Garbage Collection Probes	320
10.6.5.5	Method Compilation Probes	321
10.6.5.6	Monitor Probes	323
10.6.5.7	Application Tracking Probes	325
10.6.5.8	The hotspot_jni Provider	325
10.7	MDB Reference	328
Chapter 11	Kernel Statistics	329
11.1	C-Level kstat Interface	329
11.1.1	Data Structure Overview	330
11.1.2	Getting Started	333
11.1.3	Data Types	333

11.1.3.1	KSTAT_TYPE_RAW	334
11.1.3.2	KSTAT_TYPE_NAMED	335
11.1.3.3	KSTAT_TYPE_TIMER	336
11.1.3.4	KSTAT_TYPE_INTR	336
11.1.3.5	KSTAT_TYPE_IO	337
11.1.4	kstat Names	339
11.1.5	Functions	340
11.1.6	Management of Chain Updates	340
11.1.7	Putting It All Together	341
11.2	Command-Line Interface	343
11.2.1	The <code>kstat</code> Command	344
11.2.2	Real-World Example That Uses <code>kstat</code> and <code>nawk</code>	345
11.3	Using Perl to Access <code>kstats</code>	348
11.3.1	The Tied-Hash Interface to the <code>kstat</code> Facility	348
11.3.2	The <code>update()</code> Method	349
11.3.3	64-Bit Values	349
11.3.4	Getting Started With Perl	350
11.3.5	<code>netstatMulti</code> Implemented in Perl	351
11.4	Snooping a Programs <code>kstat</code> use with <code>DTrace</code>	354
11.5	Adding Statistics to the Solaris Kernel	354
11.5.1	A <code>kstat</code> Provider Walkthrough	355
11.5.2	I/O Statistics	359
11.6	Additional Information	362
 PART THREE		
Debugging		363
Chapter 12	The Modular Debugger	365
12.1	Introduction to the Modular Debugger	365
12.1.1	MDB	366
12.1.2	MDB Features	366
12.1.3	MDB Features	367
12.1.4	Terms	368
12.2	MDB Debugger Concepts	368
12.2.1	Building Blocks	368
12.2.2	Targets	368

12.2.3	Debugger Commands	369
12.2.4	Walker	369
12.2.5	Debugger Modules	370
12.2.6	Macros	370
12.2.7	Modularity	370
Chapter 13	An MDB Tutorial	373
13.1	Invoking MDB	373
13.1.1	Logging Output to a File	374
13.2	MDB Command Syntax	375
13.2.1	Expressions	376
13.2.2	Symbols	378
13.2.3	Formatting Metacharacters	380
13.2.4	Formatting Characters	380
13.2.5	dcmds	384
13.2.6	Walkers	385
13.2.7	Macros	386
13.2.8	Pipelines	386
13.2.9	Piping to UNIX Commands	388
13.2.10	Obtaining Symbolic Type Information	388
13.2.11	Variables	390
13.2.12	Walkers, Variables, and Expressions Combined	393
13.3	Working With Debugging Targets	394
13.3.1	Displaying Stacks	396
13.3.2	Displaying Registers	396
13.3.3	Disassembling the Target	397
13.3.4	Setting Breakpoints	397
13.4	GDB-to-MDB Reference	398
13.5	dcmd and Walker Reference	400
13.5.1	Commands	400
13.5.2	Comments	401
13.5.3	Expressions	401
13.5.4	Symbols	401
13.5.5	dcmds	401
13.5.6	Variables	402
13.5.7	Read Formats	402

13.5.8	Write Formats	402
13.5.9	Search Formats	402
13.5.10	General dcmts	403
13.5.11	Target-Related dcmts	403
13.5.12	CTF-Related	404
13.5.13	Kernel: proc-Related	404
13.5.14	Kernel: Thread-Related	404
13.5.15	Kernel: Synchronization-Related	404
13.5.16	Kernel: CPU-Related	405
13.5.17	Kernel: Memory-Related	405
13.5.18	Kernel: kmem-Related	405
13.5.19	Process: Target-Related	406
13.5.20	Kernel: kmdb-Related	406
Chapter 14	Debugging Kernels	409
14.1	Working With Kernel Cores	409
14.1.1	Locating and Attaching the Target	409
14.1.2	Examining Kernel Core Summary Information	410
14.1.3	Examining the Message Buffer	412
14.1.4	Obtaining a Stack Trace of the Running Thread	412
14.1.5	Which Process?	413
14.1.6	Disassembling the Suspect Code	414
14.1.7	Displaying General-Purpose Registers	415
14.1.8	Navigating the Stack Backtrace	415
14.1.9	Looking at the Status of the CPUs	419
14.1.10	Traversing Stack Frames in SPARC Architectures	421
14.1.11	Listing Processes and Process Stacks	424
14.1.12	Global Memory Summary	425
14.1.13	Listing Network Connections	426
14.1.14	Listing All Kernel Threads	426
14.1.15	Other Notable Kernel dcmts	427
14.2	Examining User Process Stacks Within a Kernel Image	428
14.2.1	Enabling Process Pages in a Dump	428
14.2.2	Invoking MDB to Examine the Kernel Image	429
14.2.3	Locating the Target Process	429
14.2.4	Extracting the User-Mode Stack Frame Pointers	430

14.3	Switching MDB to Debug a Specific Process	431
14.3.1	Constructing the Process Stack	431
14.3.2	Examining the Process Memory	434
14.4	kldb, the Kernel Modular Debugger	434
14.4.1	Diagnosing With kldb and moddebug	435
14.4.1.1	Starting kldb From the Console	435
14.4.2	Booting With the Kernel Debugger	435
14.4.3	Configuring the a tty console on x86	436
14.4.4	Investigating Hangs	437
14.4.5	Collecting Information About Panics	438
14.4.6	Working With Debugging Targets	438
14.4.7	Setting Breakpoints	440
14.4.8	Forcing a Crash Dump With <code>halt -d</code>	441
14.4.9	Forcing a Dump With kldb	441
14.5	Kernel Built-in MDB <code>dcmds</code>	441
Appendix A	Tunables & Settings	445
Appendix B	DTrace OneLiners	451
Appendix C	Java DTrace Scripts	455
Appendix D	Sample Perl Kstat Utilities	461
Appendix E	Bibliography	477

