

For more information about this document
contact the Reference Desk at Wendt Library
(askwendt@engr.wisc.edu) or 262-0696

Accelerating AIX

**Performance Tuning for
Programmers and System Administrators**

Rudy Chukran



ADDISON-WESLEY

An Imprint of Addison Wesley Longman, Inc.

Reading, Massachusetts • Harlow, England • Menlo Park, California
Berkeley, California • Don Mills, Ontario • Sydney
Bonn • Amsterdam • Tokyo • Mexico City

Contents

List of Figures	ix
List of Tables	xv
Preface	xvii
Chapter 1 Introduction to Tuning Concepts	1
1.1 Why Consider Tuning?	1
1.2 What Can I Tune?	2
1.3 What Are Your Performance Objectives?	3
1.4 Methodology	4
1.5 Roles of the Performance Analyst	4
Chapter 2 AIX System Design	7
2.1 General AIX Design Goals	8
2.2 Program Loading	9
2.3 Virtual Memory Overview	11
2.3.1 Page Faults	11
2.3.2 Late Page Space Allocation	12
2.3.3 Filesystem Caching	14
2.4 I/O Overview	16
2.4.1 Layers for File I/O	16
2.4.2 JFS Log	16
2.4.3 Sequential Read-Ahead and Write-Behind	20
2.4.4 Logical Volume Manager	20
2.5 CPU Scheduling Overview	24
2.5.1 Scheduling Threads	24

2.6	2.5.2	Memory Overcommitment	26	
2.7		Facts of Life about Disk Hardware	27	
		Summary	28	
Chapter 3		Tools for the System Administrator		31
3.1		Vmstat: CPU and Virtual Memory Utilization	32	
3.2		Iostat: Disk I/O Utilization	35	
3.3		Sar: System Activity Report	37	
3.4		Ps: Process Status	40	
3.5		AIX Monitor	41	
3.6		Tprof: Trace Profiler	43	
3.7		Filemon: File I/O Activity	44	
3.8		Svmon: Virtual Memory Report	50	
3.9		Rmss: Reduced Memory Simulator	52	
3.10		PDT: Performance Diagnostic Tool	53	
3.11		Performance Toolbox	55	
3.12		Summary	60	
Chapter 4		Remedies		61
4.1		CPU-Bound Remedies	62	
	4.1.1	Faster CPU	62	
	4.1.2	More CPUs	63	
	4.1.3	Scheduling Tasks at Other Times	63	
	4.1.4	Finding Runaway Processes	64	
	4.1.5	Tuning Scheduler Algorithm	64	
4.2		Memory-Bound Remedies	70	
	4.2.1	Adding More Real Memory	71	
	4.2.2	Tuning File Memory Allocation	71	
	4.2.3	Tuning Memory Overcommitment Algorithm	73	
	4.2.4	Page Space Configuration	74	
	4.2.5	Isolating Memory Leaks	75	
4.3		Disk-Bound Remedies	76	
	4.3.1	Importance of Data Placement on a Disk	77	
	4.3.2	Examples of Data Placement	78	
	4.3.3	Determining Your Disk Configuration	84	
	4.3.4	Specifying Where Your Data Goes	86	
	4.3.5	JFS Considerations	90	
	4.3.6	SCSI Bus Speed Limitations	97	
	4.3.7	Miscellaneous Tuning Parameters	98	
4.4		Summary	100	
Chapter 5		Local Area Network Tuning		101
5.1		Network Monitoring Tools	102	
	5.1.1	Netstat	102	
	5.1.2	Netpmn	105	
	5.1.3	Tcpdump	108	

5.2	Network Tuning Remedies	109	
5.2.1	Adapter	109	
5.2.2	TCP/IP Memory Options	110	
5.2.3	Network Interface	114	
5.2.4	TCP/IP Setup Errors	117	
5.2.5	NFS	117	
5.3	Summary	124	
Chapter 6	Programming Tools		127
6.1	Compiler	127	
6.1.1	XL Compiler Flags Common to All Languages	127	
6.1.2	XL FORTRAN Unique Flags	136	
6.1.3	XL C Unique Flags	137	
6.2	Linker	137	
6.2.1	Dynamically Linked Objects	139	
6.2.2	Statically Linked Objects	144	
6.3	Profilers	146	
6.3.1	Tprof Revisited	146	
6.3.2	Gprof	151	
6.3.3	Xgprof	153	
6.4	Debugging	156	
6.4.1	Trace	156	
6.4.2	Trace Postprocessing Tools	158	
6.4.3	Program Visualizer	160	
6.5	Summary	163	
Chapter 7	Remedies for Programmers		165
7.1	System Calls	165	
7.1.1	Read and Write	165	
7.1.2	Interprocess Communication	166	
7.1.3	Mapped File I/O	175	
7.1.4	Network Socket Options	176	
7.1.5	Dynamic Module Loading	177	
7.2	Hardware Type Inquiry	180	
7.3	Hardware Optimization	184	
7.3.1	Matrix Access Optimization	184	
7.3.2	Lock Access Optimization	187	
7.3.3	Word Alignment	188	
7.4	Summary	190	
Chapter 8	Putting It All Together		191
8.1	What Is the Problem—Really?	191	
8.2	Where Do I Start?	192	
8.3	Where Do I Go Next?	193	
8.4	Take Some Remedy	193	
8.5	When to Stop	194	
8.6	Summary of Tools	194	

Appendix		197
A.1	References	197
A.1.1	IBM Publications	197
A.1.2	External Publications	197
A.2	Sources for AIX Software	198
A.2.1	IBM Sources	198
A.3	Sources for Free Software for AIX on the Net	200
A.4	Performance Toolbox Metrics	200
A.5	Source Examples	203
A.5.1	Sieve of Erosthenes— <code>sieve.c</code>	203
Index		207

Each console is a collection of rectangular panes called instruments. Each instrument can show several values at once by using color coding to distinguish the values. Click on the bottom pane of the Local CPU Sample. This instrument will now have a dotted border around it to show it has been selected. From the Edit Console menu, select Modify Instrument, followed by Interval, as shown in Figure 3.28. We are going to slow down the sample interval from a feverish one second to a more leisurely ten seconds. You will see a dialog with a slider bar that you can move to increase the number of samples that are saved. Change this value to ten or to some value you like, and then press Proceed. Observe that the movement is ten times slower.

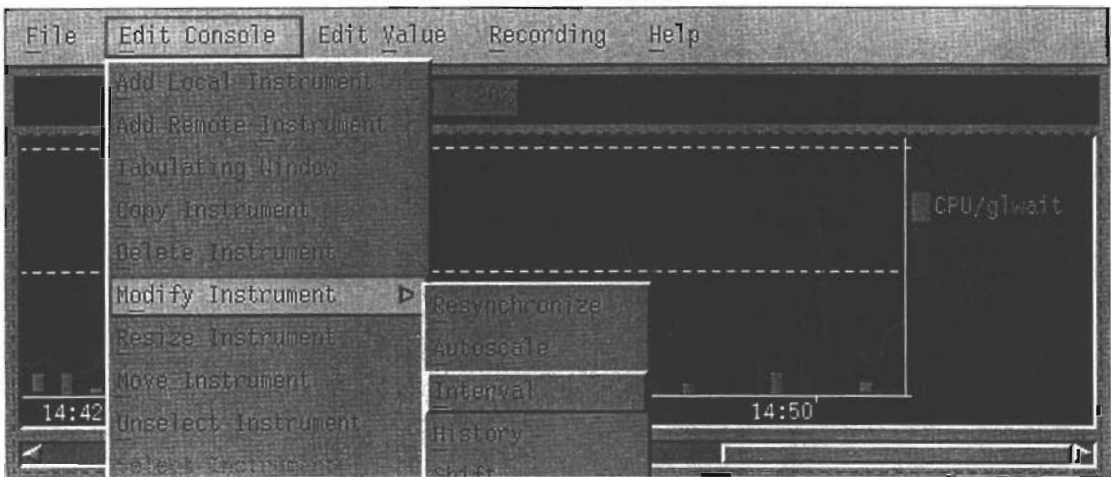


Figure 3.28 Xperf example customization menu

Sample Illustration

Now let's change some of the colors of the values in the instrument. Select the Edit value menu, and select Change Value. In the next dialog, select the `glkern` value, which is the same as the system-time statistic in `vmstat`. You are presented with a Changing Properties of a Value dialog, in which you select the color button. You now get a color palette with which to select another color, like red. Before hitting Proceed, look at the console and note that the color change has taken effect immediately. In this way you can easily punch a few color blocks to see how the colors meld. Before leaving the Properties dialog, key a more useful label into the Your label box. How about something like: CPU system %. Hit the Proceed button, and all the changes you made will take effect.

Next let's change a value to a different value in the same instrument. Select Edit Value and Delete; then select the `glidle` value to delete. Since the sum of system, user, wait, and idle always totals 100%, showing idle explicitly is redundant and clutters the graph. Let's add a value in its place. Select Edit Value and Add Value. You will get a mind-numbing list of system statistics from which to choose. Select Proc, and then select