



The Science and Art of Performance Analysis/Tuning

Jim Fiori

Principal Field Technologist

Sun Microsystems Federal, Inc



Agenda

- Performance – Does it matter?
- How did I get here?
- Important Concepts
- Background of a Good Performance Engineer
- Recommended Coursework
- Performance Disciplines, Tools, Examples
- Approaches to Solving Performance Problems
- Trends in Performance Analysis
- Demos

Performance – Does it Matter?

- Lost Art?
- Hardware is cheap – just throw more at it?
 - Faster CPU's?
 - More CPU's?
 - More memory?
 - More systems?
- Big Iron is dead?
 - Not quite
- Customers still care deeply about costs!
 - All about price-performance

How Did I Get Here?

- College
 - Bits and bytes
- Early job assignments
 - Limited resources (cpu, memory)
 - Networking protocols
- Problem solving
- Performance engineering at Sun
 - Why is it enjoyable?
 - Stories

Important Concepts

- Heisenberg Uncertainty Principle
- Amdahl's Law
- Keep your data close!
 - On-chip (L1) cache 1X (nanosecond)
 - L2 cache 10X
 - Memory 100X
 - File system cache 1000X
 - Disk Controller 1,000,000X
 - Disk 10,000,000X
 - Difference between L1 and disk – (1 sec vs 115 days)
- Scaling to today's processors

Background of a Good Performance Engineer

- Software Development
 - Multi-lingual
 - Multi-threading concepts
- Understanding of computer architectures
 - Pipelining, caches, memory
 - Memory concepts (caches, bandwidth, latency, UMA vs NUMA)
- Assembly
- Operating Systems Concepts
 - Scheduling, interrupts, pre-emption
 - File systems
 - Virtual memory
 - Mutual exclusion
 - Distributed OS concepts

Background of a Good Performance Engineer (cont.)

- Disk Technologies
 - Mechanics
 - Controllers, RAID
- Networking Technologies
 - Protocols, TCP/UDP/IP (RFC's)
 - Socket programming

Recommended Coursework

- Operating Systems
- Computer Architecture
- Queuing Theory
- Networking
- Database

Performance Disciplines, Tools, Examples

- Operating Systems
 - User time vs system time
 - Lock Contention
 - Tools:
 - vmstat, top, prstat, mpstat, lockstat
 - Profiling tools – Dtrace/SystemTap
 - lsof, pmap, ptree
- Application
 - truss/strace for observing system calls
 - prof/gprof
 - SunStudio Performance Analyzer
 - DTrace/SystemTap

Performance Disciplines, Tools, Examples

- Java
 - jmap/jconsole
 - -Xprof, -Xrunhprof:cpu=samples
 - Garbage Collection - -verbosegc, -XX:+PrintGCDetails, jheap
 - SunStudio/Netbeans
 - DTrace (Java 6)
- Application
 - truss/strace for observing system calls
 - prof/gprof
 - SunStudio Performance Analyzer
 - DTrace/SystemTap

Performance Disciplines, Tools, Examples

- Network
 - snoop/tcpdump, Wireshark
 - Netstat, nicstat
 - Load generators – ttcp, iperf, uperf
- Disk/File Systems
 - iostat
 - Load generators – dd, vdbench
 - SWAT for monitoring
- Database
 - Oracle – STATSPACK/AWR
 - MySQL – mysql “show profiles”, MySQL Enterprise Monitor
- Capacity Planning
 - TeamQuest, Quest Foglight

Trends

- Virtualization
 - Xen/VMWare/Hyper-V/LDOM's
 - Consolidation
- Chip Multi-processing, chip multi-threading
- Solid State Disks
- 10gigE Networking
- Database
- Dynamic Tracing

Approaches to Solving Performance Problems

- Clear Problem Definition
 - “It's Slow”
 - Express in terms of response times, throughput
- Remove potential pieces to the puzzle
- De-compose the problem
 - eg. isolate network or disk component
 - Reproduce problem *outside* of application
- Measure, tune, repeat

What is DTrace?

- Solaris Dynamic Tracing allows safe instrumentation of production systems with no impact to performance when not enabled
- 10's of thousands of “probes” in the kernel and your applications
- DTrace “consumers” send commands to “providers” to enable probes
- DTrace providers instrument the kernel and applications, record custom data, and pass the data back to the consumers
- Designed to drastically reduce debugging cycle
- DTrace provides concise answers to arbitrary questions
- 'D' scripting language allows for arbitrary actions and predicates
- Demos

*“ [expletive deleted] It's like they saw
inside my head and gave me The One True Tool.”*
- A Slashdotter, in a post referring to DTrace

Demos...

DTrace Probes for Scripting Languages

- PHP, Ruby
 - > <http://blogs.sun.com/bmc>
- Perl (PUSHSUB/POPSUB)
 - > http://blogs.sun.com/roller/page/alanbur?entry=dtrace_and_perl
- Python
 - > http://blogs.sun.com/roller/page/smg?entry=beer_python_and_stuff
- Apache
 - > <http://www.opensolaris.org/jive/thread.jspa?threadID=3788&tstart=60>

```
$ dapachetop 10
```

```
2005 Nov 28 02:01:00, load: 0.00, Requests: 4, Bytes: 8 Kb
```

```
Requests : GET: 4 POST: 0 HEAD: 0 TRACE: 0
```

```
Requests Request/Sec Bytes Sent URI
2          .20          4 Kb /foo
1          .10          2 kb /bar
1          .10          2 kb /bling
```

DTrace and Databases

- PostgreSQL

- > <http://www.postgresql.org/docs/8.2/static/dynamic-trace.html>

```
transaction__start(int);  
transaction__commit(int);  
transaction__abort(int);  
lwlock__acquire(int, int);  
lwlock__release(int);  
lwlock__startwait(int, int);  
lwlock__endwait(int, int);  
lwlock__condacquire(int, int);  
lwlock__condacquire__fail(int, int);  
lock__startwait(int, int);  
lock__endwait(int, int);
```

DTrace and Databases

- PostgreSQL

```
#!/usr/sbin/dtrace -Zqs
postgresql*:::transaction-start
{
    self->ts=timestamp;
    @cnt[pid]=count();
}
postgresql*:::transaction-commit
{
    @avg[pid]=avg(timestamp - self->ts);
}
```

DTrace and Databases

- MySQL
 - > 6.0.4 will have DTrace probes

```
DELETE_START (arg0)
DELETE_END (arg0)
EXTERNAL_LOCK (arg0, arg1)
FILESORT_START (arg0)
FILESORT_END (arg0)
INSERT_ROW_START (arg0)
INSERT_ROW_END (arg0)
INSERT_START (arg0)
INSERT_END (arg0)
SELECT_START (arg0)
SELECT_END (arg0)
UPDATE_START (arg0)
UPDATE_END (arg0)
```

DTrace and Java

- Current VM agents have limitations
- Fixed in Mustang (Java 6)
- Native DTrace probes:
 - > method-entry/exit
 - > gc-start/finish
 - > thread-start/end
 - > object-allocation
 - > and more
- See blog:

http://blogs.sun.com/kamg/#Dtrace_Probes_in_Mustang_Hotspot

DTrace Toolkit V 0.99

- Dtrace community at opensolaris.org
- 230 scripts!
- /Examples directory
- /Code directory
- Scripts for:
 - > Java 6 (j_XXX)
 - > Java Script (js_XXX)
 - > Ruby (rb_XXX)
 - > Python (py_XXX)
 - > Perl (pl_XXX)
 - > Tcl (tcl_XXX)
 - > Bourne Shell (sh_XXX)
 - > PHP (php_XXX)

DTrace and GUI's

- Chime
 - > DTrace visualization tool
 - > view aggregations over time
 - > downloaded from opensolaris.org,
 - > runs on Solaris 8/07
- D-Light
 - > plug-in to Sun Studio 12 (express)
 - > profile code using DTrace
 - > allows for more of a systemic view
- Netbeans plug-in
 - > allows DTrace scripts to be run from SS12, NetBeans IDE's
- Sun's OpenStorage

Scenarios – How to Apply tools

- Start with the “Big 5” XXstat tools:

```
% vmstat 3
```

kthr			memory			page			disk				faults			cpu					
r	b	w	swap	free	re	mf	pi	po	fr	de	sr	cd	s0	s1	--	in	sy	cs	us	sy	id
0	0	0	1108996	652152	9	33	19	0	0	0	16	2	0	0	0	398	1026	509	5	2	92
0	0	0	1002760	513164	0	45	15	0	0	0	0	2	0	0	0	402	792	528	2	4	94

```
% mpstat 3
```

CPU	minf	mjf	xcal	intr	ithr	csw	icsw	migr	smtx	srw	syscl	usr	sys	wt	idl
0	0	0	2	16	11	14	0	2	0	0	10	0	0	0	100
1	4	0	2	221	117	101	0	2	2	0	6	0	0	0	100
CPU	minf	mjf	xcal	intr	ithr	csw	icsw	migr	smtx	srw	syscl	usr	sys	wt	idl
0	0	0	2	15	9	18	0	4	1	0	12	0	0	0	100
1	4	0	2	226	122	107	0	4	1	0	13	0	0	0	99

```
% iostat -dxnz 3
```

extended device statistics

r/s	w/s	kr/s	kw/s	wait	actv	wsvc_t	asvc_t	%w	%b	device
0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.4	0	0	c0t0d0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	3.2	0	0	c1t1d0
0.5	0.1	10.4	48.9	0.0	0.0	0.0	10.4	0	0	c1t0d0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	3.4	0	0	c2t2d0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	5.4	0	0	c2t5d0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	7.1	0	0	c2t8d0
0.1	0.0	48.6	7.0	0.0	0.0	0.0	54.8	0	0	c2t11d0

Scenarios – How to Apply tools

- Start with the “Big 4” XXstat tools (cont.):

```
% prstat -mLcn 10 3
```

PID	USERNAME	USR	SYS	TRP	TFL	DFL	LCK	SLP	LAT	VCX	ICX	SCL	SIG	PROCESS/LWPID
1186	jdf	0.3	2.2	0.7	0.0	0.0	0.0	97	0.2	355	2	1K	0	find/1
4418	jdf	0.1	0.3	0.0	0.0	0.0	0.0	100	0.0	13	0	224	0	prstat/1
119	root	0.0	0.0	0.0	0.0	0.0	0.0	100	0.0	3	0	17	0	picld/4
116	root	0.0	0.0	0.0	0.0	0.0	0.0	100	0.0	10	0	31	0	nscd/2
398	root	0.0	0.0	0.0	0.0	0.0	0.0	100	0.0	3	0	10	0	sendmail/1
7	root	0.0	0.0	0.0	0.0	0.0	100	0.0	0.0	0	0	0	0	svc.startd/4
7	root	0.0	0.0	0.0	0.0	0.0	100	0.0	0.0	0	0	0	0	svc.startd/3
7	root	0.0	0.0	0.0	0.0	0.0	0.0	100	0.0	0	0	0	0	svc.startd/2
7	root	0.0	0.0	0.0	0.0	0.0	0.0	100	0.0	0	0	0	0	svc.startd/1
119	root	0.0	0.0	0.0	0.0	0.0	100	0.0	0.0	0	0	0	0	picld/5

```
% netstat -i 1 5
```

input		e1000g			output		input (Total)		output		
packets	errs	packets	errs	colls	packets	errs	packets	errs	packets	errs	colls
5691	0	13859	0	0	8672	0	16840	0	0	0	0
2302	0	6778	0	0	2302	0	6778	0	0	0	0
1764	0	7862	0	0	1764	0	7862	0	0	0	0
2252	0	9433	0	0	2252	0	9433	0	0	0	0
2059	0	9142	0	0	2059	0	9142	0	0	0	0

Key Resources

- Open Solaris Performance
 - > solarisinternals.com
 - > Solaris Performance and Tools
- DTrace Toolkit, Chime
 - <http://opensolaris.org/os/community/dtrace/>
- D-Light
 - http://blogs.sun.com/solarisdev/entry/project_d_light_tutorial
- DTrace Manual now a wiki!
 - <http://wikis.sun.com/display/DTrace/Documentation>
- Solaris 10 - DTrace - An Introduction (webcast)
 - http://www.snpnet.com/sun_DTrace/dtrace_flash.html

Key Resources (cont.)

- Java VM Agents
 - <http://solaris10-dtrace-vm-agents.dev.java.net/>
 - http://blogs.sun.com/ahl/entry/dtrace_presentation_at_javaone
- DTrace FAQ
 - http://www.genunix.org/wiki/index.php/DTrace_FAQ
- Advanced: Tips, Tricks, and Gotchas
 - http://blogs.sun.com/bmc/resource/dtrace_tips.pdf
- Subject Matter Expert Contact Information
 - dtrace-discuss@opensolaris.org
- Books: Solaris Performance and Tools - Mauro/McDougall/Gregg
- Recommended Training: Solaris Dynamic Tracing Guide
- Solaris Performance Wiki: solarisinternals.com
- Next Steps – download scripts, practice, engage



THANK YOU!

Jim Fiori

jim.fiori@sun.com

