



Performance Statistics of JavaTM Applications Using DTrace and Chime

Steven Reynolds

Senior Product Manager
www.int.com

BOF-9472

You can get a previous version of this presentation on the web at

www.int.com/presentations/dtrace_chime

Agenda

Introduction to DTrace and Chime

DTrace

Chime

DTrace

- DTrace is a system wide tool that collects statistics from thousands of probes
- Is safe for use on production systems
- Text output
- No impact if probes are not enabled
- Programmable (D scripts)

Output from syscall.d

```
Satellite> dtrace -s syscall.d -Z -p 918  
dtrace: script 'syscall.d' matched 232 probes  
^C
```

close	1
lseek	1
open64	1
brk	2
fstat64	2
getpid	3
stat64	3
uname	3
write	54
read	212
pollsys	296
ioctl	304

Output from where.d

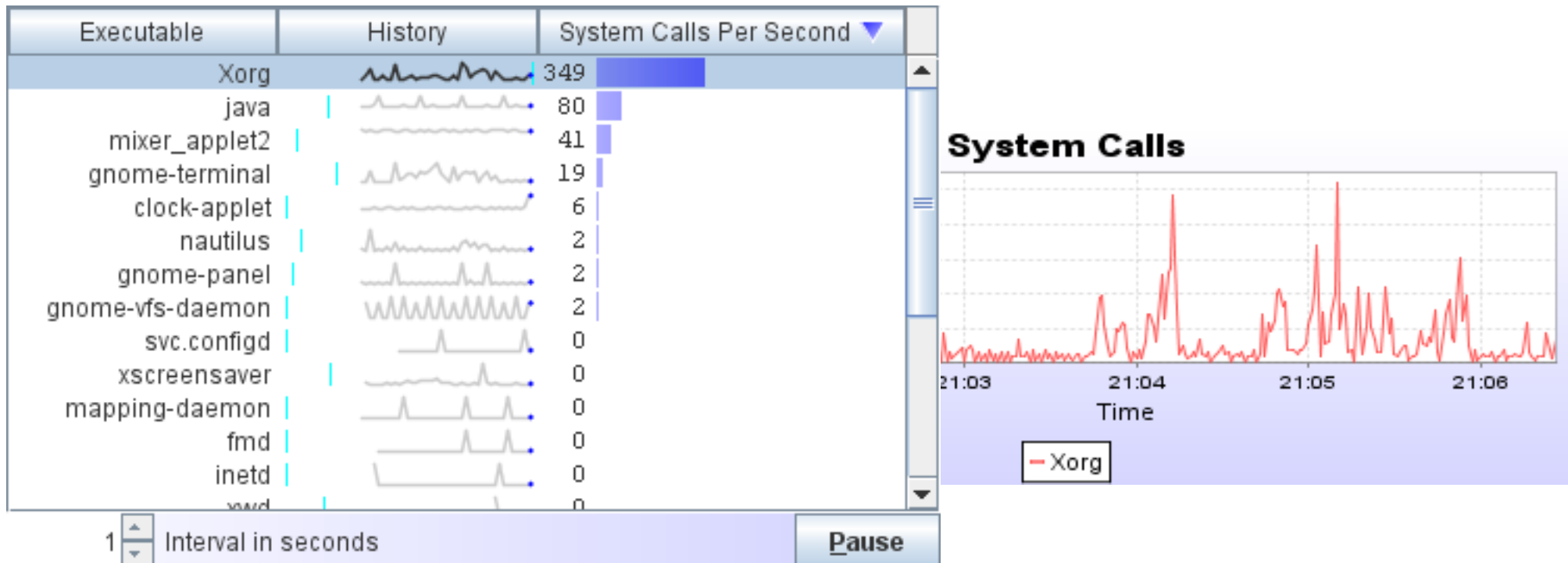
value	--- Distribution	---- count
1024		0
2048		10
4096	@@@@@@@@@@@@@@	370
8192	@@@@@@@@@@	232
16384	@@@@@@	148
32768	@	38
65536	@	15
131072		10
262144		7
524288		5
1048576		1
2097152		1
4194304		1
8388608	@@@@@@@@@@	273
16777216	@	28
33554432		0

DTrace

- Available on Solaris 10
- Coming soon to the Mac OS-X Leopard and also to FreeBSD
- Built in provider in JDK 6
- Downloadable provider available for JDK 5 and JDK 1.4

Chime

- Chime gives a GUI display of DTrace output
- Adds bar charts, line graphs, and sparkline graphs
- Easy control of sort order
- Can have drill-down support



Chime

Executable ▲ 1	Nanoseconds ▲ 2	Count
bash		
clock-applet		
dsdm	2,048	0
	4,096	3
	8,192	2
	16,384	0
gnome-panel	4,096	0
	8,192	11
	16,384	1
	32,768	0
gnome-terminal		
init		
java	2,048	0
	4,096	2
	8,192	103

- Displays defined by xml files
- Wizard to make new displays easily
- Can run remotely too
- Requires Solaris Nevada (OpenSolaris) build 35 or later.

Agenda

Introduction to DTrace and Chime

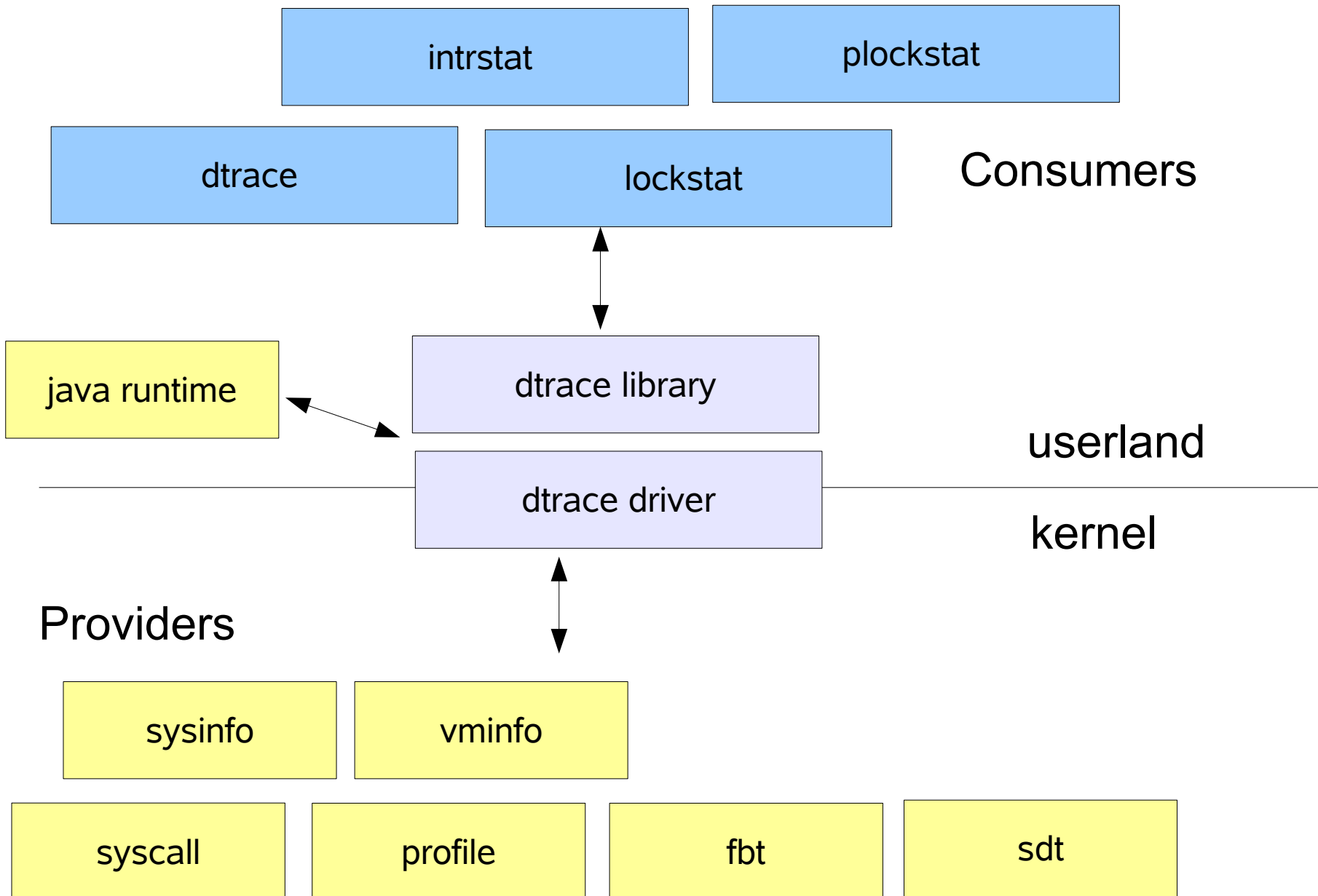
DTrace

Chime

DTrace Intro

- Read the DTrace Guide: [Dynamic Tracing Guide](#)
- Refer to the JDK 6 DTrace Probes
- Syntax is like c
- Also has macro variables: \$1, \$2,... and \$target
- Not Turing complete: no loops, no branches, no user defined functions
- Has a VM. The VM runs in the kernel.

DTrace Architecture



Java Probes

Hotspot Probes

- JVM Life-cycle
- Thread Life-cycle
- Classloading
- GC
- Method compilation
- Monitor
- Application
 - Method calls
 - Object allocations

Hotspot-JNI Probes

- JNI method calls
 - Entry
 - Return

Variable Scope

Clause Scope Variables

```
this->variable = 1;
```

Thread Scope Variables

```
self->somevariable = 10;
```

Global Variables - All aggregations are global

```
variable = 2;  
@something[key] = aggfunc(args);
```

D Script: Thread Table

Variable Declaration

```
this char *thread_name;
```

Probe Description

```
hotspot$target:::thread-start  
{
```

Action

```
/* args[0] thread name & args[1] length of the name */  
this->thread_name = (char*) copyin(arg0, arg1+1);  
this->thread_name[arg1] = '\0';  
  
/* args[3] native/OS thread ID */  
printf("%d\t%d\t%s\n", arg3,  
        stringof(this->thread_name));
```

```
}
```

D Script: Thread Table

Declare char array
thread_name in
clause context

```
this char *thread_name;
```

Fire action when a
Java thread starts

```
hotspot$target::thread-start  
{
```

Copy arg0 into
thread_name

```
/* args[0] thread name & args[1] length of the name */  
this->thread_name = (char*) copyin(arg0, arg1+1);  
this->thread_name[arg1] = '\\0';
```

```
/* args[3] native/OS thread ID */
```

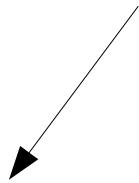
```
printf("%d\\t%d\\t%s\\n", arg3,  
        stringof(this->thread_name));
```

Print the thread ID
and thread name

```
}
```


Probe Definition

A field that is empty matches any probe.
Pattern match characters are used: * and ?



```
provider:module:function:name  
/ predicate /  
{  
    ...  
}
```

Using a Predicate in the Probe Definition

Built-in DTrace variable:
process ID passed in from
the command line

```
hotspot$target:::method-entry
```

```
/tid == $1/
```

```
{
```

```
...
```

```
}
```

Only fire the action if the thread ID is
the same as macro argument 1

Aggregations

Aggregations are a global variable that collects a value for a set of keys

```
syscall:::entry
/ execname == $$1 /
{
    @calls[tid,probefunc] = count();
}
```

Aggregation variable

Keys

Aggregating function

Aggregations are similar to a Java Hashtable

DTrace Script Tips

- Thread variables are initialized to zero (on **all** threads). Check for zero values, and don't use them.
- Keep thread specific calculations in the same thread:

use `self`

```
syscall::read:entry
{
    self->t = timestamp;
}
```

```
syscall::read:return
```

```
/self->t != 0/
```

```
{
    printf("%d/%d spent %d nsecs in read(2)\n",
           pid, tid, timestamp - self->t);
    self->t = 0;
}
```

DTrace Script Tips

- For user string data, use `copyin()` intrinsic to read it
- Dropouts: ran out of space, data was dropped
- Pragma options
 - Buffer size, buffering policy, aggregation rate
- Set thread scope variables to zero so they will be GC-ed: `self->t = 0;`
- Prefer Clause Scope variables

Tips for Running DTrace scripts

Most DTrace Scripts

```
dtrace -s script.d
```

Run script even when some probe definitions don't match anything

DTrace Scripts using Hotspot provider

```
dtrace -s script.d -Z -p <JVM_pid>
```

```
dtrace -s script.d -Z -c "java HelloWorld"
```

```
hotspot$target:::method-entry  
{  
    ...  
}
```

Sample scripts

DTrace scripts in `/usr/demo/dtrace`

JDK DTrace scripts under the JDK home in
`sample/dtrace`

Output from method_calls_stat.d

...

```
8235  java/lang/Object:<init>: ()V
12222  java/lang/StrictMath:floor: (D)D
13790  sun/awt/SunToolkit:awtLock: ()V
13790  sun/awt/SunToolkit:awtUnlock: ()V
```

=====

```
JAVA_CALLS:      745276
JNI_CALLS:       33454
SYS_CALLS:       43667
```

```
Run time:                29317339435
Syscall time:            11509
Java+JNI time:          29317327926
```


Extended Probes for Java Provider

- Use the JVM option: `-XX:+ExtendedDTraceProbes`
- Or use `jinfo`

```
jinfo -flag +ExtendedDTraceProbes <java_pid>
```

- This can slow down the application
- Needed for monitor, method entry & return, object allocation probes

No Data?

Your dtrace script gives no data?

Maybe you need to turn on `ExtendedDTraceProbes`?

You won't get any warnings!

How Do I use DTrace?

1. Start with the tools you know: prstat, mpstat, iostat...
2. See what functions are being called: dtrace
3. Look for lock contention: plockstat, dtrace

- Adam Leventhal

Solaris Internals and Solaris Performance and Tools
give lots of examples of drilling down like this

Agenda

Introduction to DTrace and Chime

DTrace

Chime

Chime

- Chime displays aggregations
- All aggregations must have the same keys
- Some new displays are available for Java, not part of the chime package yet.
- Chime at OpenSolaris site

<http://www.opensolaris.org/os/project/dtrace-chime/>

- Webcast by Tom Erickson

<http://frsun.downloads.edgesuite.net/sun/07C00941/>

Monitor Wait Times

Target Process ID: 953

Thread ID	Wait Time (ns) ▼	Count
15	65,536	0
	32,768	1
	16,384	0

1 Interval in seconds

Script for Monitor Display

When a monitor is contended,
save the timestamp of the start of contention

```
hotspot$target:::monitor-contended-enter  
{  
    self->ts = timestamp;  
}
```

```
hotspot$target:::monitor-contended-entered  
/ self->ts /  
{  
    @wait_times[tid] = quantize(timestamp - self->ts);  
    self->ts = 0;  
}
```

When the contended monitor is entered,
update a histogram of the lengths of contention (per thread)

Monitor Wait Times

Target Process ID: 953

Thread ID	Wait Time (ns) ▼	Count
15	65,536	0
	32,768	1
	16,384	0

1 Interval in seconds

Pause

Key: tid

@wait_times bucket values

@wait_times bucket frequencies

Chime I/O Statistics

Device	CPU	Interrupts Per Second	Percent Time ▼
uhci#1	cpu0	53	0.04%
ata#1	cpu0	1	0%
ehci	cpu0	1	0%
pcic	cpu0	2	0%
uhci	cpu0	2	0%
uhci#2	cpu0	1	0%
i8042	cpu0	0	0%
ata	cpu0	0	0%
rtls	cpu0	0	0%

9 devices

1 cpu

60

0.05%

1



Interval in seconds

Pause

Display Decomposition

Device	CPU	Interrupts Per Second	Percent Time ▼
uhci#1	cpu0	53	0.04%
ata#1	cpu0	1	0%
ehci	cpu0	1	0%
pcic	cpu0	2	0%
uhci	cpu0	2	0%
uhci#2	cpu0	1	0%
i8042	cpu0	0	0%
ata	cpu0	0	0%
rtls	cpu0	0	0%

9 devices 1 cpu 60 0.05%

1 Interval in seconds Pause

Keys for aggregation

First aggregation value

Second aggregation

Display Decomposition

Device	CPU	Interrupts Per Second	Percent Time ▼
uhci#1	cpu0	53	0.04%
ata#1	cpu0	1	0%
ehci	cpu0	1	0%
pcic	cpu0	2	0%
uhci	cpu0	2	0%
uhci#2	cpu0	1	0%
i8042	cpu0	0	0%
ata	cpu0	0	0%
rtls	cpu0	0	0%

Keys:

[stringof(`devnamesp[this->devi->devi_major].dn_name),
this->devi->devi_instance, cpu]

Display Decomposition

Device	CPU	Interrupts Per Second	Percent Time ▼
uhci#1	cpu0	53	0.04%
ata#1	cpu0	1	0%
ehci	cpu0	1	0%
pcic	cpu0	2	0%
uhci	cpu0	2	0%
uhci#2	cpu0	1	0%
i8042	cpu0	0	0%
ata	cpu0	0	0%
rtls	cpu0	0	0%

`@counts[...] = count();`

Display Decomposition

Device	CPU	Interrupts Per Second	Percent Time ▼
uhci#1	cpu0	53	0.04%
ata#1	cpu0	1	0%
ehci	cpu0	1	0%
pcic	cpu0	2	0%
uhci	cpu0	2	0%
uhci#2	cpu0	1	0%
i8042	cpu0	0	0%
ata	cpu0	0	0%
rtls	cpu0	0	0%

`@times[...] = sum(vtimestamp - self->ts);`

Tips for Chime

- Don't printa() the aggregation
- Be aware of dtrace options set by pragma directives

```
#pragma D option aggregate=100ms
```

Causes a failure – aggregation rate needs to be greater than 100ms

- Recommendation: Use chime to set these options, remove pragma directives from the dtrace script
- Use the zdefs chime option
- Limit the data rate

Limit the Data Rate

Easy to get scratch space overflow errors

Try chime option: `buffpolicy=ring`

Try `dtrace` function: `trunc(<aggregation>, <num_rows>)`

Try filtering on a thread

```
hotspot$target:::<probe_name>  
/tid=$1/
```

Set thread scope variables to zero

Getting Started with Chime

- Get it as a package from opensolaris site
<http://www.opensolaris.org/os/project/dtrace-chime/>
- It installs to /opt/OSOL0chime
 - The directory has two letter Oh's and a zero
- Requires Solaris Nevada build 35 or later. Get
 - Solaris Express from sun.com or
 - The OpenSolaris starter kit from opensolaris.org

Thanks

- DTrace team: Bryan Cantril, Mike Shapiro, Adam Leventhal
- Chime contributors: Tom Erickson, Bill Rushmore, Guillaume Bozon