ORACLE®

Solaris Dynamic Tracing - DTrace

Jim Mauro
Oracle Corporation

Sun

ORACLE®

Acknowledgements

- Some of this material represents an aggregation and consolidation of existing material, including the original Usenix paper*. A significant amount of the material contained in these slides was inserted from the slide sets, blogs, emails, letters, post cards, faxes, telegrams and sticky notes of others:
- Stefan Parvu
- Brendan Gregg
- Bryan Cantrill
- Mike Shapiro
- Adam Leventhal
- Jon Haslam
- Bart Smaalders
- Jarod Jenson
- Chad Mynhier
- Jim Fiori
- Jonathan Adams
- John Birrell
- Simon Ritter
- Angelo Rajadurai
- Bob Netherton
- Peter Karlsson
- Roch Bourbonnais
- Richard McDougall
- Keith McGuigan
- John Levon
- Chip Bennett
- Phil Harman

*http://www.sun.com/bigadmin/content/dtrace/dtrace_usenix.pdf

Solaris Dynamic Tracing - DTrace

" [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

" With DTrace, I can walk into a room of hardened technologists and get them giggling"

- Bryan Cantrill, Inventor of DTrace

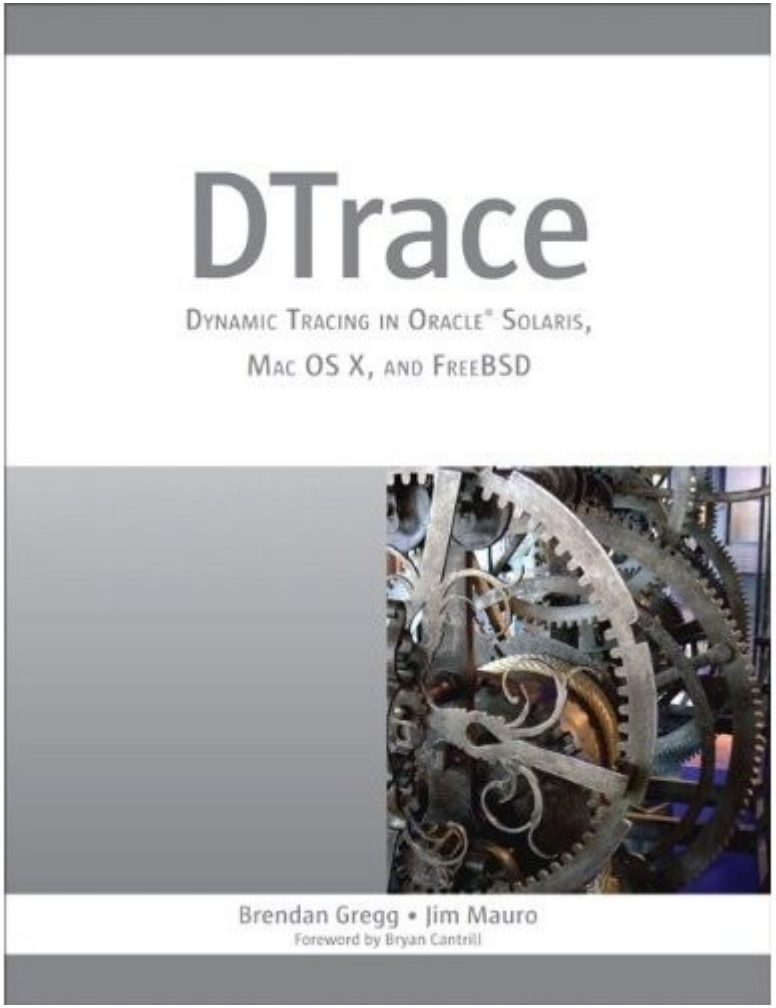
Resources!

- DTrace documentation
  http://wikis.oracle.com/display/DTrace/Documentation
- DTrace tutorials, scripts, etc
  http://www.solarisinternals.com/wiki/index.php/DTrace_Topics
- DTrace community (lots 'o stuff)
  http://www.opensolaris.org/os/community/dtrace/
- DTrace ToolKit
  http://www.brendangregg.com/dtrace.html#DTraceToolkit
- Blogs, blogs, blogs
  - Oracle, Joyent
  - http://www.dtrace.org
  - http://www.dtracebook.com

Shameless Plug

- DTrace cookbook
- Over 230 D Scripts
- Over 270 one-liners
- Shows the entire software stack, by example
- Topics include;

  CPU, Memory, Network, Disks, File Systems, Applications, the Kernel, etc...



Scripts!

- Some of the commands and scripts we'll be looking at today

  /usr/demo/dtrace/*.d (on Solaris 10/Solaris 11)

  http://www.solarisinternals.com/wiki/index.php/DTrace_Topics_One_Liners

  http://www.nbl.fi/~nbl97/solaris/dtrace/index.html

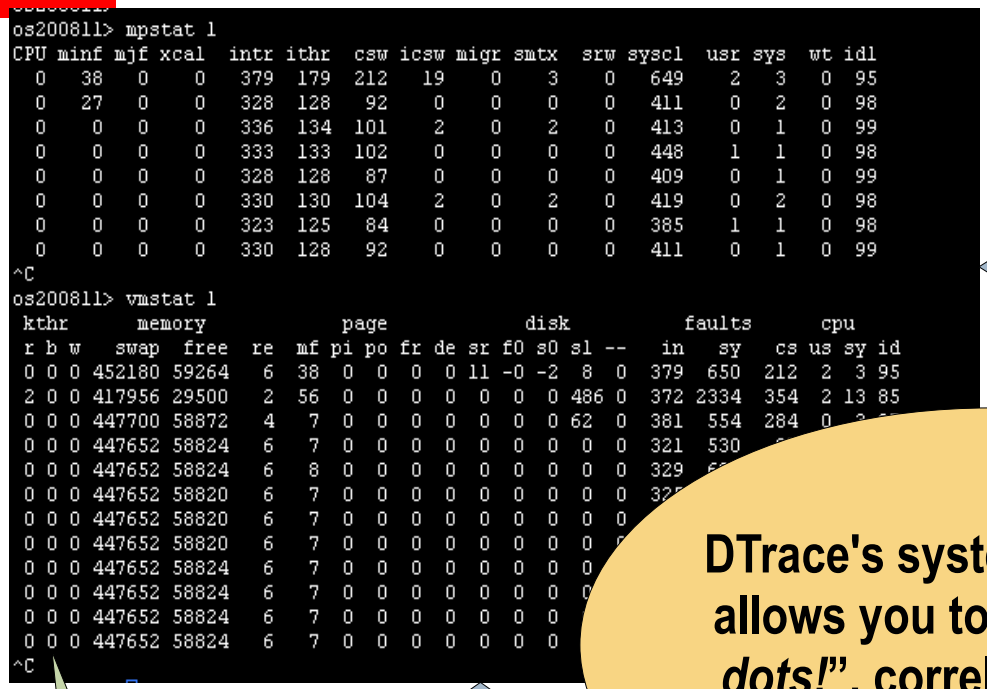
  http://www.solarisinternals.com/si/dtrace/index.php

  http://www.opensolaris.org/os/community/dtrace/dtracetoolkit/

  http://www.dtracebook.com

What is DTrace

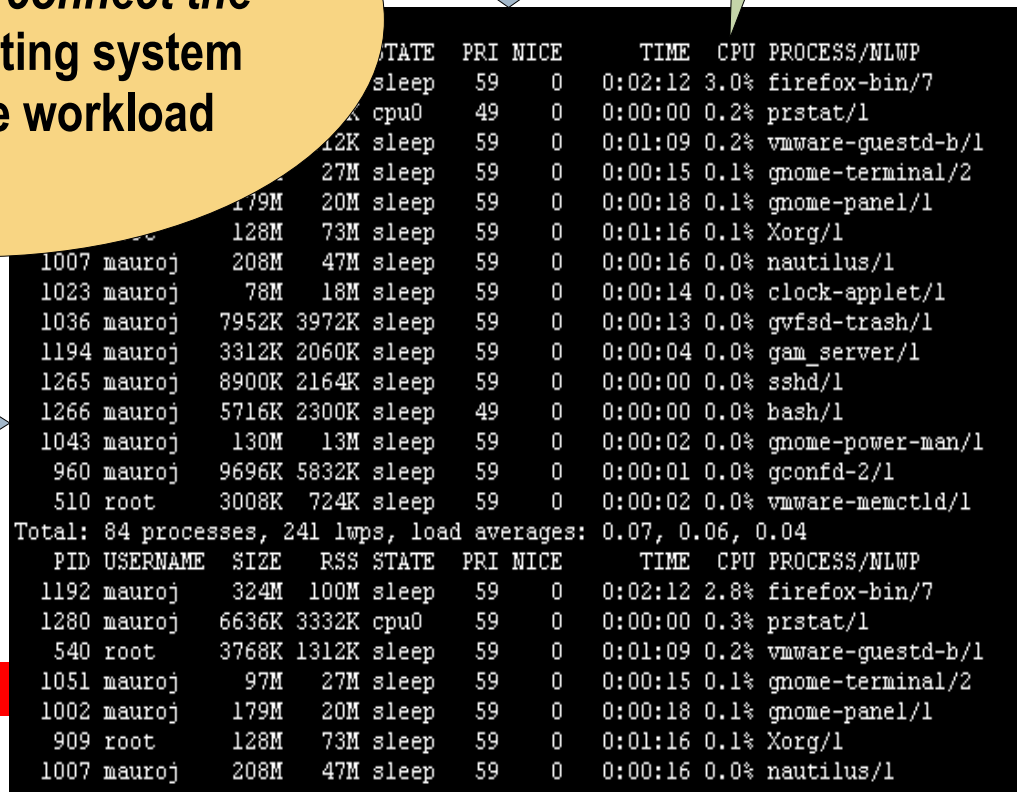
- DTrace is a facility for the dynamic instrumentation of production systems, for the purpose of troubleshooting and analysis.
  - First introduced in Solaris 10 (3/05)
  - Ported to Mac OS X and FreeBSD
- DTrace is many things, in particular:
  - An instrumentation framework
  - A programming language
- DTrace provides observability across the entire software stack from one tool. This allows you to examine software execution like never before.
  - Instrument kernel and user software in a unified fashion



DTrace's system-wide view allows you to "*connect the dots!*", correlating system activity to the workload

"what are the processes doing?"

"what's the system doing?"



DTrace

An Observability Revolution

- Ease-of-use and *instant gratification* engenders serious *hypothesis testing*
- Instrumentation directed by high-level control language (not unlike AWK or C) for easy scripting and command line use
  - Build your DTrace toolbox
- Comprehensive probe coverage and powerful data management allow for *concise* answers to *arbitrary* questions
  - *What are these system calls, and who's executing them?*

DTrace

- Safe and comprehensive: tens-of-thousands of data monitoring points
  - Inspect kernel and user space
- Reduced costs: problems usually found in minutes or hours, not days or weeks
- Flexibility: DTrace lets you create your own custom programs to dynamically instrument the system
- No need to instrument your applications via source code modifications; no need to stop or restart them

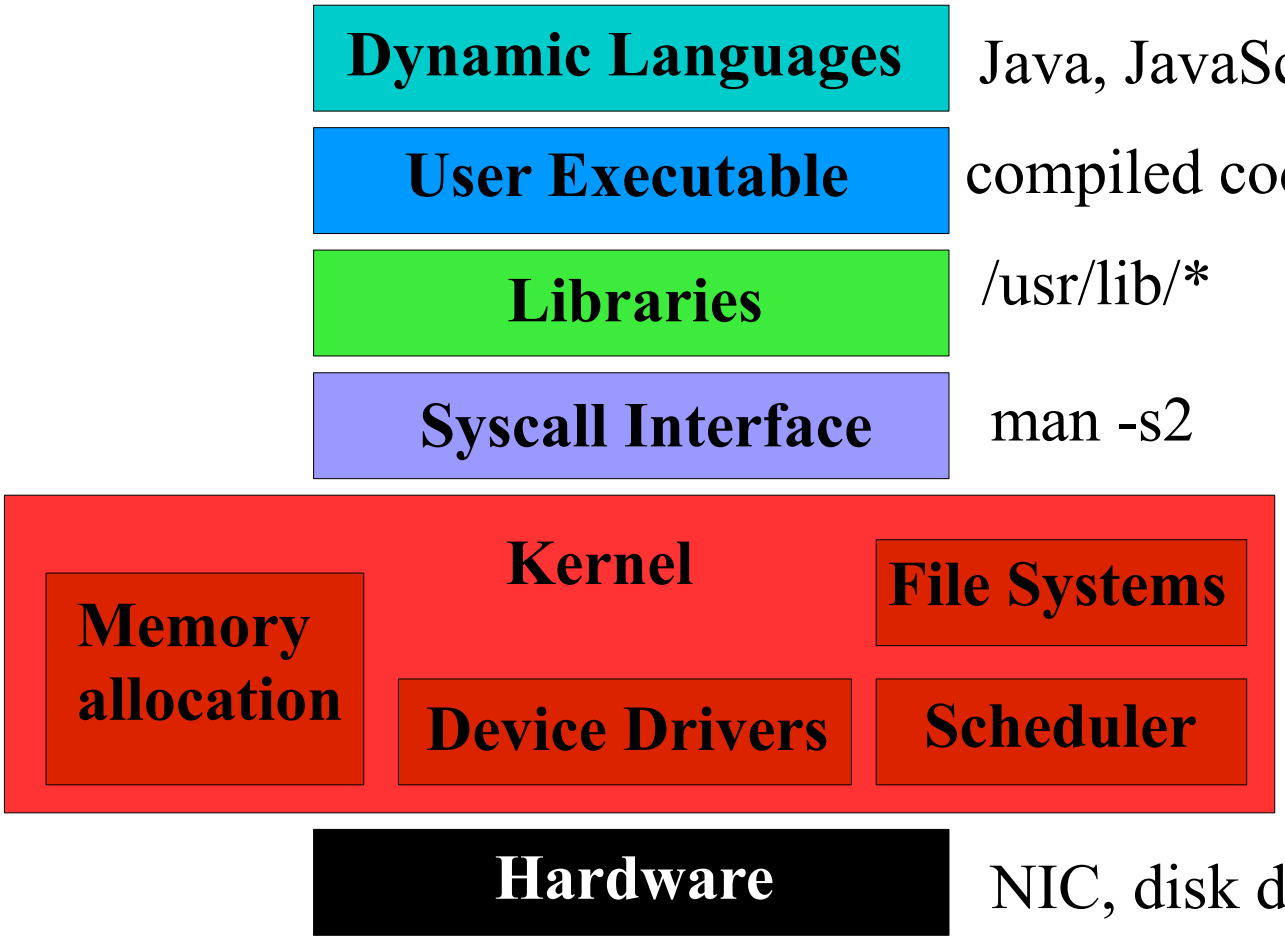
DTrace Features*

- Dynamic Instrumentation
- Unified Instrumentation
- Arbitrary-context kernel instrumentation
- Data integrity
- Arbitrary actions
- Predicates
- High-level control language
- Scalable data aggregation
- Speculative tracing
- Scalable architecture
- Virtualized consumers

*http://www.sun.com/bigadmin/content/dtrace/dtrace_usenix.pdf

The Entire Software Stack

- How did you analyze these?



Examples:

Java, JavaScript, ...

compiled code, /usr/bin/*

/usr/lib/*

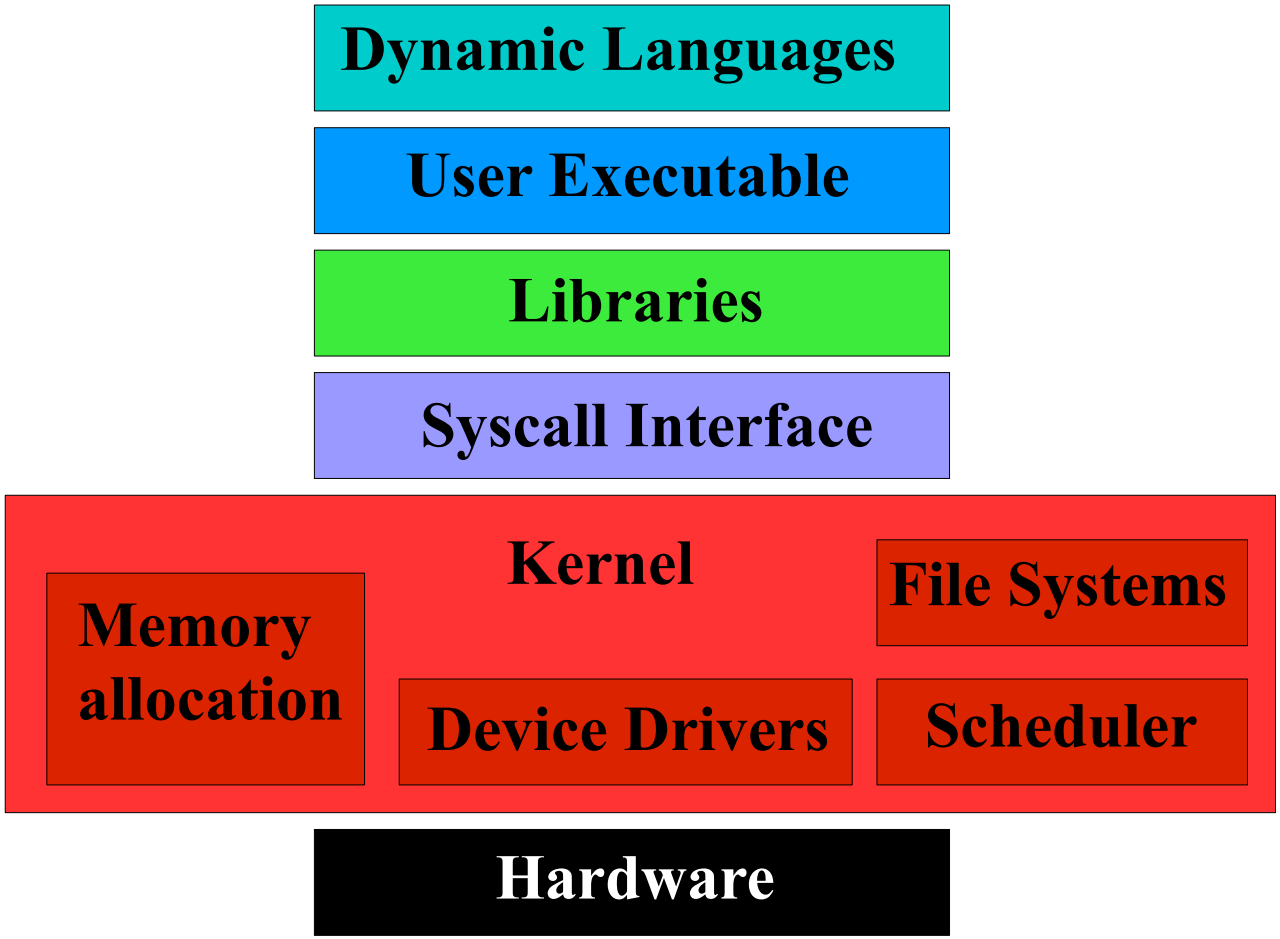
man -s2

VFS, DNLC, UFS,
ZFS, TCP, IP, ...
sd, st, hme, eri, ...

NIC, disk data controller, CPU

The Entire Software Stack

- It was possible, but difficult.



Previously:

debuggers

truss -ua.out

apptrace, sotruss

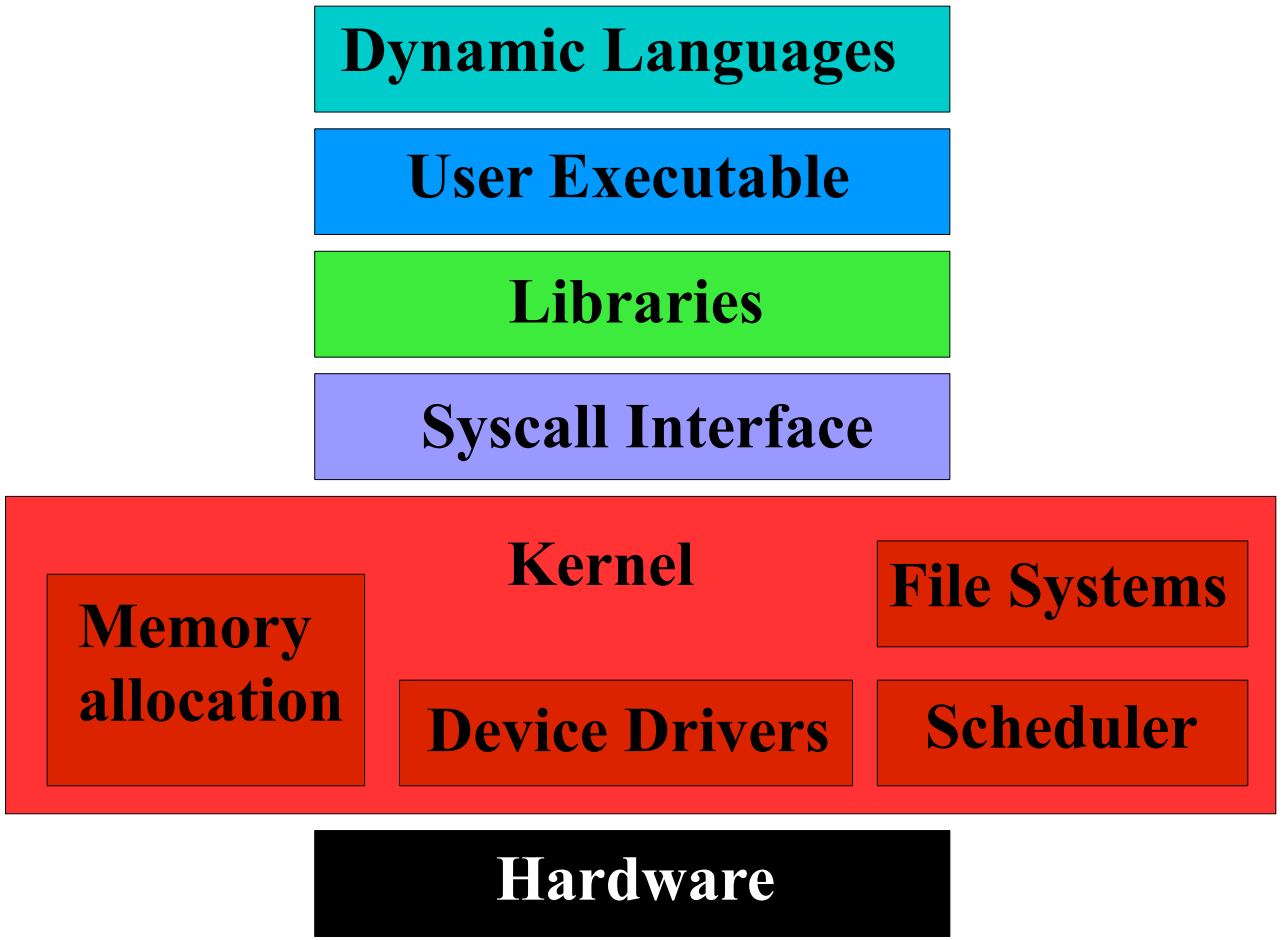
truss

prex; tnf*
lockstat
mdb

kstat, PICs, guesswork

The Entire Software Stack

- DTrace is all seeing:



DTrace visibility:

Yes, with providers

Yes

Yes

Yes

Yes

Yes (to some extent, very recent)

Syscall Example

- Using truss,

Only examine 1 process

```
$ truss date
execve("/usr/bin/date", 0x08047C9C, 0x08047CA4)  argc = 1
resolvepath("/usr/lib/ld.so.1", "/lib/ld.so.1", 1023) = 12
resolvepath("/usr/bin/date", "/usr/bin/date", 1023) = 13
xstat(2, "/usr/bin/date", 0x08047A58)              = 0
open("/var/ld/ld.config", O_RDONLY)                = 3
fxstat(2, 3, 0x08047988)                           = 0
mmap(0x00000000, 152, PROT_READ, MAP_SHARED, 3, 0) = 0xFEFB0000
close(3)                                           = 0
mmap(0x00000000, 4096, PROT_READ|PROT_WRITE|PROT_EXEC, MAP_PRIVATE|MAP_ANON, -1
sysconfig(_CONFIG_PAGESIZE)                        = 4096
[...]
```

Output is limited to provided options

truss *slows down* the target (probe effect)

Syscall Example

- Using DTrace

You can select which syscall(s)

You choose the output

```
# dtrace -n 'syscall:::entry { printf("%16s %x %x", execname, arg0, arg1); }'
dtrace: description 'syscall:::entry ' matched 233 probes
CPU     ID                    FUNCTION:NAME
  1  75943                       read:entry             Xorg f 8047130
  1  76211                  setitimer:entry             Xorg 0 8047610
  1  76143                     writev:entry             Xorg 22 80477f8
  1  76255                   pollsys:entry              Xorg 8046da0 1a
  1  75943                       read:entry             Xorg 22 85121b0
  1  76035                      ioctl:entry      soffice.bin 6 5301
  1  76035                      ioctl:entry      soffice.bin 6 5301
  1  76255                   pollsys:entry      soffice.bin 8047530 2
[...]
```

Minimum performance cost

Watch every process

What is DTrace For?

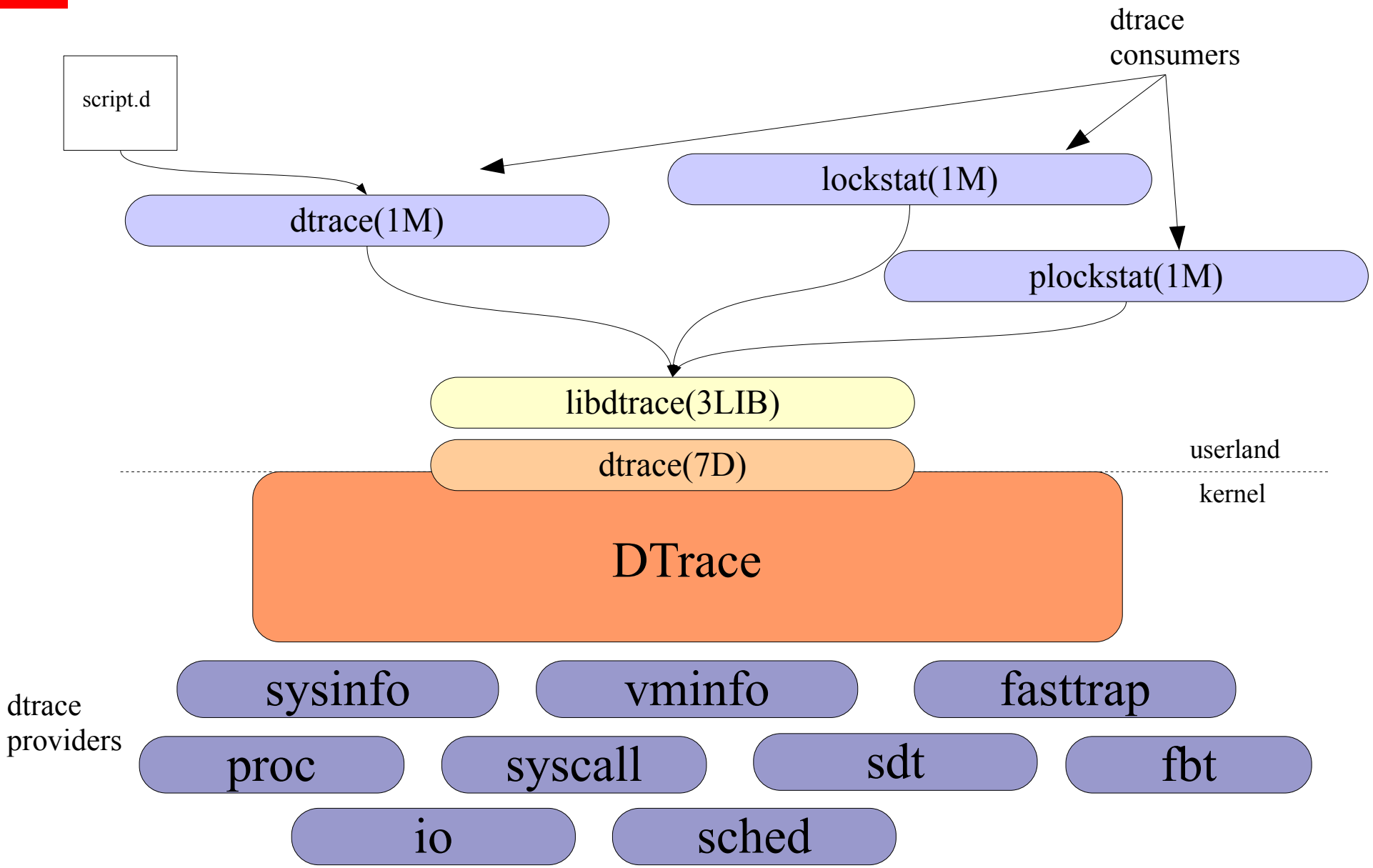
- Troubleshooting performance problems
  - Profile applications and the kernel
  - latency measurements
  - Looking for areas for improvement even when performance is acceptable
- Troubleshooting software bugs
  - Proving what the problem is, and isn't.
  - Measuring the magnitude of the problem.
- Detailed observability
  - Observing the kernel
  - Observing devices, such as disk or network activity.
  - Observing applications, whether they are from Sun, 3rd party, or in-house.

DTrace Components

- Probes
  - A point of instrumentation and data generation
  - Typically in a specific area of the code
- Providers
  - A major component of DTrace, Providers manage probes of specific types, and for a specific area of the system
    - syscall, io, sched, proc, vminfo, etc
- Consumers
  - Users of the framework

    `dtrace(1), lockstat(1), plockstat(1), intrstat(1)`

DTrace – The Big Picture



DTrace User Components

- Predicates
  - User-defined conditional statements evaluated when probes fire
  - Provides a control flow mechanism for your D programs – data pruning at the source
- Actions
  - What to do when the probe fires
    - Data to gather
    - Timestamps for profiling
    - Many other actions supported

A DTrace Program

- DTrace on the command line...

```
# dtrace -n 'syscall:::entry / execname == “soffice” /
                                { @sc[probefunc] = count(); }'
```

- A DTrace script

```
#!/usr/sbin/dtrace -s

#pragma D option quiet

syscall:::entry
/ execname == “soffice” /
{
   @sc[probefunc] = count();
}
```

DTrace – What Happens

- dtrace command compiles the D language Script.
- Intermediate code checked for safety (like java).
- The compiled code is executed in the kernel by DTrace.
- DTrace instructs the provider to enable the probes
- As soon as the D program exits all instrumentation removed
- No limit (except system resources) on number of D scripts that can be run simultaneously
- Different users can debug the system simultaneously without causing data corruption or collision issues.

DTrace Probes

- A point of instrumentation, made available by a provider
- A four-tuple name uniquely identifies every probe;
  provider:module:function:name
  - *provider*: the DTrace provider that manages the probe (DTrace kernel module)
  - *module*: kernel module or user library where the probe is located
  - *function*: kernel or user function containing the probe
  - *name*: represents an entry point in that function (e.g. *entry* or *return*), or has a meaningful name (e.g. io:::start, proc:::exec)

Probes

- Anchored Probes
  - Instrument a specific point in code, e.g.

    ```
    fbt:ufs:ufs_read:entry
    io:::start
    ip:::receive
    ```
- Unanchored Probes
  - Are not associated with a specific location in code
  - Do not have a module or function component to their name
  - profile and tick

    ```
    profile-997hz, tick-10sec
    ```

Probes

- List probes
  - Use dtrace(1M) with the '-l' option
  - For each probe the four-tuple name will be displayed, probe components are ':' separated
  - List all probes:

```
$ dtrace -l
```

  - List all probes offered by syscall provider:

```
$ dtrace -lP syscall
```

  - List all probes offered by the ufs module:

```
$ dtrace -lm ufs
```

  - List all providers:

```
$ dtrace -l | awk '{print $2}' | sort -u
```

Probes

- Enabling probes
  - Activate a probe by not using '-l' option
  - Default action with enabled probes- the CPU, the probe number and name are displayed whenever the probe fires
  - Enable all probes from nfs and ufs module:

```
$ dtrace -m nfs,ufs
```

  - Enable all read function probes:

```
$ dtrace -f read
```

  - Enable all probes from io provider:

```
$ dtrace -P io
```

Probes – Where Did I Fire?

- Built-in DTrace variables provide probe name components in your probe clauses
  - This are all string type variables

probeprov - Provider

probemod - Module

probefunc - Function

probename – Name

Probes and DTrace Built-in Variables

- Among the many built-in variables provided by DTrace, there are probe-specific variables available when a probe fires
  - Function call argument list. Arguments passed to a function instrumentable through an “entry” probe are available as either:

    int64_t arg0, arg1, ..., arg9 – args available as raw 64 bit integers

    args[0], args[1], ..., args[9] – typed args, corresponding to the specific data types of the arg list
- Probes in some providers have a specific arg list made available by the provider
  - e.g. the IO provider arg list of pointers to a buf structure, devinfo structure and fileinfo structure when IO provider probes fire
- You need to RTFM to determine what args are available for a given provider
  - For function entry arg lists, you need man pages, kernel source, or just mdb(1) on a running Solaris system

Providers

- A methodology for instrumenting the system
- Providers offer all probes to the DTrace framework
- DTrace framework confirms to providers when a probe is activated
- Providers pass the control to DTrace when a probe is enabled
- Example of providers: syscall, lockstat, fbt, io, mib

Providers

- Providers do a couple interesting things for us...
  - Manage probes
  - Abstract a complex subsystem with intuitive probes, enabling and enhancing observability and analysis

    sched:::oncpu

    io:::start

    etc...
  - You can use DTrace effectively to track application and kernel activity in areas of the kernel that you may not be familiar with

Provider Documentation

- Additional documentation may be found here,

Target	Provider	Additional Docs
syscalls	`syscall`	man(2)
libraries	`pid:lib*`	man(3C)
app code	`pid:a.out`	source code, ISV, developers
raw kernel	`fbt`	Solaris Internals 2nd Ed, http://cvs.opensolaris.org

Providers

```
nv98> dtrace -l | awk '{ print $2}' | sort -u
PROVIDER
Xserver767
dtrace
fbt
fsinfo
io
ip
lockstat
lx-syscall
mib
proc
profile
sched
sdt
syscall
sysevent
sysinfo
vminfo
```

```
macosx> dtrace -l | awk '{ print $2}' | sort -u
PROVIDER
dslockstat87530
dtrace
fbt
io
lockstat
mach_trap
mds66
plockstat16190
plockstat16191
plockstat16192
proc
profile
syscall
vminfo
macosx>
```

Providers

- DTrace refers to most providers as “Stable” providers
  - The probes and args will not change across releases
  - Provides for building a toolbox that will work indefinitely
  - io, sched, proc, vminfo, sysinfo, fpuinfo, mib, etc, are all stable providers
  - fbt is not, since fbt by definition instruments the kernel functions entry and return points.
  - It is generally recommended to stick with stable providers, at least while you're getting started
  - Check the documentation for the specific stability level of a provider
  - New providers under development!

Providers, cont.

- Examples
  - **proc:::exec**
  - **sched:::oncpu**
  - **fbt:ufs:ufs_read:entry**
  - **syscall::**read:entry{ printf("Process %d", pid); }
  - **syscall::**write:entry/execname=="firefox-bin"/ { @[probefunc] = count(); }
  - **sysinfo:::**readch{ trace(execname); exit(0); }
  - **sysinfo:::**writech
  - **io:::**start

The D language

- A simple dynamically interpreted language used by dtrace(1M)
- Similar to C language and awk(1):
  - Supports ANSI C operators and has support for strings
  - Supports several variable types, including built-in variables: pid, execname, timestamp, curthread, etc
- No control-flow constructs: loops, if statements
- Arithmetic may only be performed on integers in D programs, floating-point arithmetic is not permitted in D
  -

A DTrace D Program

```
probe
/ optional predicate /
{
    clause
    what to do when the probe(s) fire, and the predicate,
    if present, evaluates true
}
```

Example;

```
syscall::read:entry
/ execname == “java” /
{
    @reads[pid, fds[arg0].fi_pathname] = count();
}
```

Or, via the command line;

```
#dtrace -n 'syscall::read:entry / execname == “java” /
    { @reads[pid, fds[arg0].fi_pathname] = count(); }'
```

The D Language

- Supports a full range of native data types
  - integers, floating point, strings, pointers
- Supports a full range of operators
  - arithmetic, logical, relational, assignment
- Variables
  - Scalar variables
  - Variable scope – thread local, clause local and global
  - associative arrays
- Built-in variables
  - pid, execname, timestamp, etc...
- Actions and Subroutines
  - Taken when the probe fires - enclosed in { }
  - Data recording, speculations, various subroutines...

The D language, cont.

- Scripting in D
- Easy to create D scripts to hold one or more probe clauses
- All D scripts end in dot d (script_name.d)
- Add the interpreter as the first line in the script

```
#!/usr/sbin/dtrace -s
```

  - Or create the script and run as;

```
#dtrace -s ./script.d
```

Predicates

- D expressions that define a conditional test
- Allow actions to only be taken when certain conditions are met.
  - */ predicate /*
- The actions will be activated only if the value of the predicate expression is true
- Used to filter and meet certain conditions: look only for a process which has the pid = 1203, match a process which has the name firefox-bin

Aggregations

- Used to aggregate data and look for trends
- Simple to generate reports about: total system calls used by a process or an application, the total number of read or writes by process...
- Has the general form:

  @name[keys] = aggfunc(args)

- There is no need to use other tools like: awk(1), perl(1)
- The general definition of aggregating function:

  $f(f(x_0) \cup f(x_1) \cup ... \cup f(x_n)) = f(x_0 \cup x_1 \cup ... \cup x_n)$

Aggregations

- Aggregating functions
  - count() : the number of times called, used to count for instance the total number of reads or system calls
  - sum() : the total value of the specified expressions
  - avg() : the arithmetic average of the specified expression
  - min() : the smallest value of the specified expression
  - max(): the largest value of the specified expression
  - quantize() : a power-of-two frequency distribution, simple to use to draw distributions
  - lquantize(): a linear distribution (scalar, min, step, max)
- Non-aggregating functions
  - mode and median

Aggregations, cont.

- What's going on with my system ?

```
dtrace -n syscall:::entry
```

- Difficult to read, start aggregating...

```
dtrace -n 'syscall:::entry{@[execname] = count();}'
```

- Filter on read system call

```
dtrace -n
  'syscall::read*:entry{@[execname]=count();}'
```

- Add the file descriptor information

```
dtrace -n
  'syscall::read*:entry{@[execname,arg0]=count();}'
```

Aggregations, cont.

- Drill-down and get a distribution of each read by application name

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


syscall::read*:return
/self -> ts/
{
 @time[execname] = quantize(timestamp - self->ts);
 self->ts = 0;
}
```

Aggregations, cont.

- Data normalization
  - used to aggregate over a specific constant reference: e.g.: system calls per second
  - normalize()
  - denormalize()
- Truncate
  - used to minimize the aggregation results, keep certain top results
  - trunc(aggregation, trunc value)

Output formatting

- Special routines to format the output: trace(), printf() or printa()
- For specific output format use built-in printf()
  - `printf("execname is %s", execname);`
  - `printf("%d spent %d secs in read\n", pid, timestamp - t);`
- For aggregations use printa()
  - `printa("Aggregation is:", @a);`
  - `printa(@count);`
- Basic trace()
  - `trace(execname);`

DTrace Tunable Parameters

- DTrace implements several variables that set sizes (e.g. buffers), rates (buffer rotation), etc...
- The default values are good 90% of the time
  - Sometimes it's obvious when you need to change them (DTrace reported messages)
- DTrace enables setting some tunable parameter values on the command line,in D scripts, or system-wide in /etc/system
  - It is recommended that system-wide settings be avoided...per consumer tweaks as needed are the way to

Option Name	Type	dtrace(1M)	Default	Description
aggrate	time		1hz	Rate of aggregation reading
aggsize	size			Aggregation buffer size
bufresize	auto or manual			Buffer resizing policy
bufsize	size	-b		Principal buffer size
cleanrate	time		101hz	Cleaning rate
cpu	scalar	-c		CPU on which to enable tracing
defaultargs	-			Allow references to unspecified macro arguments
destructive	-	-w		Allow destructive actions
dynvarsize	size			Dynamic variable space size
flowindent	-	-F		Indent function entry and prefix
grabanon	-			Claim anonymous state
jstackframes	scalar		50	Number of default stack frames jstack
jstackstrsize	scalar		512	Default string space size for jstack
nspec	scalar			Number of speculations
quiet	-	-q		Output only explicitly traced data
rawbytes	-			Always print tracemem output in hexadecimal
specsize	size		32k	Speculation buffer size
strsize	size		256	String size
stackframes	scalar		20	Number of stack frames

http://wikis.sun.com/display/DTrace/Options+and+Tunables

Modifying DTrace Tunables

- In D scripts using the #pragma D compiler directive

```
#pragma D option bufsize=512k
```

- On the command line using -x

```
#dtrace -x bufsize=512k -n 'syscall:::entry … '
```

- On the command line with supported dtrace(1M) flags

```
#dtrace -b 512k -n 'syscall:::entry …'
```

Using DTrace

DTrace One Liners

- System Calls Count by Application

```
$ dtrace -n 'syscall:::entry{@[execname] =
  count();}'
```

- System Calls Count by Application and PID

```
$ dtrace -n 'syscall:::entry{@[execname,pid]
  = count();}'
```

- How many times a file has been opened

```
$ dtrace -n
  'syscall::open:entry{@[copyinstr(arg0)] =
  count();}'
```

DTrace – syscall example output

```
# dtrace -n 'syscall:::entry { @sc[execname,probefunc] = count() }'
dtrace: description 'syscall:::entry ' matched 235 probes
^C
  . . .
  dtgreet           pollsys                                       20
  java              stat64                                        45
  java              pollsys                                       88
  syslogd           getmsg                                       160
  syslogd           pollsys                                      160
  dtrace            p_online                                     256
  syslogd           lwp_park                                     640
  dtrace            ioctl                                       1599
```

DTrace One Liners

- Files Opened by process

```
$ dtrace -qn
  'syscall::open*:entry{ printf("%s
  %s\n",execname,copyinstr(arg0)); }'
```

- Read Bytes by process

```
$ dtrace -n 'sysinfo:::readch{ @[execname] =
  sum(arg0);}'
```

- Write Bytes by process

```
$ dtrace -n 'sysinfo:::writech{ @[execname]
  = sum(arg0);}'
```

DTrace One Liners, cont.

- Read sizes by process

```
$ dtrace -n 'syscall::read:entry{@[execname]
  = quantize(arg2);}'
```

- Writes sizes by process

```
$ dtrace -n
  'syscall::write:entry{@[execname] =
  quantize(arg2);}'
```

- Disk size by process

```
$ dtrace -qn 'io:::start{printf("%d %s
  %d\n",pid,execname,args[0]->b_bcount); }'
```

DTrace One Liners, cont.

- High system time

```
$ dtrace -n profile-501'/ arg0 / {@[stack()]
  = count()}END{trunc(@, 25)}'
```

- What processes are using fork

```
$ dtrace -n 'syscall::fork*:entry{printf("%s
  %d",execname,pid);}'
```

Discovery with DTrace

- Which process is reading which files?

```
 > dtrace -n 'syscall::read:entry / execname != "dtrace" /
                  { @reads[execname, fds[arg0].fi_pathname] = count(); }'
dtrace: description 'syscall::read:entry ' matched 1 probe
^C
bash             /proc/1709/psinfo                                       1
loader           /zp/space/f2                                            1
nscd             /etc/user_attr                                          1
bash             /export/home/mauroj/.bash_history                       2
loader           /zp/space/f3                                            2
nscd             /etc/group                                              2
su               /etc/default/su                                         8
su               /devices/pseudo/sy@0:tty                                9
bash             /dev/pts/5                                             66
Xorg             /devices/pseudo/conskbd@0:kbd                         152
gnome-terminal   /devices/pseudo/clone@0:ptm                           254
```

Discovery with DTrace

- What does a process or command actually do?

```
opensolaris>dtrace -n 'exec-success { trace(curpsinfo->pr_psargs); }'
dtrace: description 'exec-success ' matched 1 probe
CPU     ID                    FUNCTION:NAME
  0  24953         exec_common:exec-success   man ls
  0  24953         exec_common:exec-success   neqn /usr/share/lib/pub/eqnchar -
  0  24953         exec_common:exec-success   col -x
  0  24953         exec_common:exec-success   sh -c less -siM /tmp/mp1RaGim
  0  24953         exec_common:exec-success   less -siM /tmp/mp1RaGim
  1  24953         exec_common:exec-success   sh -c cd /usr/man; tbl /usr/man/man1/ls.1
                                               |neqn /usr/share/lib/pub/eqnchar - |n
  1  24953         exec_common:exec-success   tbl /usr/man/man1/ls.1
  1  24953         exec_common:exec-success   nroff -u0 -Tlp -man -
  1  24953         exec_common:exec-success   sh -c trap '' 1 15;
                    /usr/bin/mv -f /tmp/mp1RaGim /usr/man/cat1/ls.1 2> /dev/null
  1  24953         exec_common:exec-success   /usr/bin/mv -f /tmp/mp1RaGim
                                               /usr/man/cat1/ls.1
```

Kernel Mutex Locks

- mpstat smtx column

```
opensolaris> dtrace -n 'sysinfo:::mutex_adenters { @s[stack(2)] = count(); }'
         . . .
              ip`squeue_enter+0x44
              ip`tcp_wput+0xc5
             3236

              ip`squeue_drain+0x1d6
              ip`squeue_enter_chain+0x394
             3273

              e1000g`e1000g_rxfree_func+0xaa
              genunix`dblk_lastfree_desb+0x26
            12933

opensolaris> dtrace -n 'sysinfo:::mutex_adenters { @s[execname] = count(); }'
dtrace: description 'sysinfo:::mutex_adenters ' matched 1 probe
^C

  java                                                              2
  syslogd                                                           8
  sched                                                          6484
  uperf                                                         17906
```

A Walk-Through Example

System View

```
#mpstat 1
. . .
CPU minf mjf xcal  intr ithr  csw icsw migr smtx  srw syscl  usr sys  wt idl
  0    0   0  953  3046  293 5630   66  263 2863    0  3768   10  31   0  59
  1    0   0 4722  4881 3816 2635   59  184 2819    0  2763    1  30   0  69
  2    0   0  792  2768  134 5537   72  200 2883    0  3762    6  35   0  59
  3    0   0  818   853    0 2233   44  166 2850    0  2758    0  27   0  73
  4    0   0 3667  5901 3579 6592   44  167 2630    0  2734    5  31   0  64
  5    0   0  756   991  128 2115   43  130 2545    0  3792    1  30   0  69
  6    0   0 3897  7016 3963 7953   55  199 2534    0  2497    2  31   0  67
  7    0   0  661  1503  131 3028   48  132 2813    0  2808    1  27   0  72
 40    0   0  186   638    0 1002   31   86  198    0   430   75   3   0  22
 41    0   0  183   268    0  540   17   42  216    0   470   79   2   0  19
 42    0   0  214  1753    0 2433   61  265  564    0   685   65   9   0  26
 43    0   0  111   463    0  903   32   70  184    0   433   72   2   0  26
 44    0   0  242  1572    0 2748   38  140  351    0   583   55   9   0  36
 45    0   0  189   246    0  497   24   40  186    0   424   72   2   0  26
 46    0   0  171  1705   10 3186   44  228  388    0   984   54   8   0  38
 47    0   0 4344  4609 4090 1577   26   93  642    0  1037   44  12   0  45
```

System Calls

```
opensolaris> cat sc.d

#!/usr/sbin/dtrace -s

#pragma D option quiet

syscall:::entry
/ execname != "dtrace" /
{
        @sc[execname,tid,probefunc] = count();
}
tick-1sec
{
        trunc(@sc, 20);
        printf("%-16s %-8s %-16s %-8s\n","EXEC","TID","SCALL","COUNT");
        printa("%-16s %-8d %-16s %-@8d\n",@sc);
        trunc(@sc);

        printf("\n");
}
```

System Calls

```
#./sc.d

EXEC                TID       SCALL               COUNT
sshd                1         pollsys             17
sshd                1         lwp_sigmask         34
mpstat              1         ioctl               49
ldr                 8         lwp_park            476
ldr                 14        lwp_park            493
ldr                 10        lwp_park            512
mpstat              1         p_online            512
ldr                 2         lwp_park            513
ldr                 4         lwp_park            513
ldr                 6         lwp_park            530
ldr                 12        lwp_park            593
ldr                 16        lwp_park            734
ldr                 11        write               1544
ldr                 9         write               2128
ldr                 3         write               2134
ldr                 13        write               2145
ldr                 17        write               2159
ldr                 15        write               2183
ldr                 7         write               2225
ldr                 5         write               2335
. . .
```

System Calls – where's write(2) coming from?

```
# dtrace -n 'syscall::write:entry / execname == "ldr" / { @[ustack()] = count(); }'
dtrace: description 'syscall::write:entry ' matched 1 probe
^C

              libc.so.1`_write+0x8
              libc.so.1`_ndoprnt+0x309c
              libc.so.1`printf+0x100
              ldr`timer+0x38
              libc.so.1`__sighndlr+0xc
              libc.so.1`call_user_handler+0x3e0
              libc.so.1`lwp_wait+0x60
              libc.so.1`_thrp_join+0x38
              ldr`main+0x628
              ldr`_start+0x7c
                7

              libc.so.1`_write+0x8
              ldr`write_test+0x288
              ldr`iotd+0x1c0
              libc.so.1`_lwp_start
           198879
```

System Calls – where's lwp_park coming from?

```
# dtrace -n 'syscall::lwp_park:entry / execname == "ldr" / { @[ustack()] = count(); }'
dtrace: description 'syscall::lwp_park:entry ' matched 1 probe
^C

              libc.so.1`__lwp_park+0x10
              libc.so.1`lmutex_lock+0xd4
              libc.so.1`lrand48+0x24
              ldr`pt1+0x144
              libc.so.1`_lwp_start
             6536

              libc.so.1`__lwp_park+0x10
              libc.so.1`lmutex_lock+0xd4
              libc.so.1`lrand48+0x24
              ldr`pt1+0x114
              libc.so.1`_lwp_start
             6643

              libc.so.1`__lwp_unpark+0xc
              libc.so.1`lrand48+0x3c
              ldr`pt1+0x114
              libc.so.1`_lwp_start
             6647
```

System Calls – what about those writes?

```
# cat write.d
#!/usr/sbin/dtrace -s

#pragma D option quiet

syscall::write:entry
/ execname == "ldr" /
{
        @w[fds[arg0].fi_pathname, arg2, fds[arg0].fi_offset] = count();
}
tick-1sec
{
        trunc(@w, 20);

        printf("%-16s %-8s %-16s %-8s\n","PATHNAME","SIZE","OFFSET","COUNT");
        printa("%-16s %-8d %-16d %-@8d\n",@w);
        trunc(@w);

        printf("\n");
}
```

System Calls – what about those writes?

```
#./write.d
PATHNAME              SIZE     OFFSET           COUNT
/md20/f9              8192     369393664        1
/md20/f9              8192     369401856        1
/md20/f9              8192     369410048        1
/md20/f9              8192     369418240        1
/md20/f9              8192     369426432        1
. . .
/md20/f9              8192     369500160        1
/md20/f9              8192     369508352        1
/md20/f9              8192     369516544        1
/md20/f9              8192     369524736        1
/md20/f9              8192     369532928        1
/md20/f9              8192     369541120        1
/md20/f9              8192     369549312        1

PATHNAME              SIZE     OFFSET           COUNT
/md20/f9              8192     451354624        1
/md20/f9              8192     451362816        1
/md20/f9              8192     452763648        1
/md20/f9              8192     452771840        1
. . .
/md20/f9              8192     452861952        1
/md20/f9              8192     452870144        1
/md20/f9              8192     452878336        1
/md20/f9              8192     452886528        1
/md20/f9              8192     452894720        1
/md20/f9              8192     452902912        1
```

System Calls – write latency

```
# cat write_t.d
#!/usr/sbin/dtrace -s

#pragma D option quiet

syscall::write:entry
/ execname == "ldr" /
{
        self->flag = 1;
        self->fpath = fds[arg0].fi_pathname;
        self->st[self->fpath] = timestamp;
}
syscall::write:return
/ self->flag /
{
        @wt[self->fpath] = quantize(timestamp - self->st[self->fpath]);
        self->flag = 0;
}

END
{
        normalize(@wt, 1000);
        printa(@wt);
}
```

System Calls – write latency

```
/md20/f9
           value  ------------- Distribution ------------- count
            8192 |                                         0
           16384 |@@@                                      2
           32768 |@@@@@@@@@@@@@@@@@@@@                     19
           65536 |@@@@@@@@@@@@                             11
          131072 |@@                                       2
          262144 |@                                        0
          524288 |@@                                       1
         1048576 |                                         0

  /md20/f5
           value  ------------- Distribution ------------- count
            8192 |                                         0
           16384 |                                         0
           32768 |@@@@@@@@@@@@@@@@@@                       13
           65536 |@@@@@@@@@@@@@@@@@@@@                     14
          131072 |@@@                                      2
          262144 |                                         0
          524288 |@@                                       1
         1048576 |@                                        0
         2097152 |                                         0

  /md20/f7
           value  ------------- Distribution ------------- count
            8192 |                                         0
           16384 |@@@                                      2
           32768 |@@@@@@@@@@@@@@@@@@@@                     18
           65536 |@@@@@@@@@@@@                             11
          131072 |@@                                       2
          262144 |@                                        0
          524288 |@@                                       1
         1048576 |                                         0
         2097152 |                                         0
```

Threads on-cpu time.

```
# cat oncpu.d
#!/usr/sbin/dtrace -s

#pragma D option quiet

sched:::on-cpu
/ execname == "ldr" /
{
        self->st[tid] = timestamp;
}
sched:::off-cpu
/ self->st[tid] /
{
        @oncpu[tid] = avg(timestamp - self->st[tid]);
        self->st[tid] = 0;
}
tick-1sec
{
        printa(@oncpu);
        trunc(@oncpu);
}
```

Threads on-cpu time.

```
# ./oncpu.d

         1            131400
        17            193277
         9            203073
        13            207760
        11            218620
         7            222036
         5            266111
        15            297220
         3            399937
         6           1118369
        10           1252954
         2           1542556

         1             16600
        15            213874
        17            219744
        11            230908
         7            241097
        13            255832
         9            406652
         5            455303
         3            487639
         2           1511637
         6           1764175
        10           1798527
```

Where are the threads spending %usr time?

```
# dtrace -n 'profile-997hz / arg1 && execname == "ldr" /
                                  { @[tid, ufunc(arg1)] = count(); }'
dtrace: description 'profile-997hz ' matched 1 probe
^C
. . .
         2  libc.so.1`mutex_unlock_queue                          1053
         6  libc.so.1`mutex_unlock_queue                          1234
        12  libc.so.1`mutex_unlock_queue                          1247
         4  libc.so.1`mutex_unlock_queue                          1271
        16  libc.so.1`mutex_unlock_queue                          1273
         8  libc.so.1`mutex_unlock_queue                          1286
        10  libc.so.1`mutex_unlock_queue                          1300
        14  libc.so.1`mutex_unlock_queue                          1317
         2  libc.so.1`mutex_trylock_adaptive                      4972
         4  libc.so.1`mutex_trylock_adaptive                      5664
         6  libc.so.1`mutex_trylock_adaptive                      7293
         8  libc.so.1`mutex_trylock_adaptive                      7409
        16  libc.so.1`mutex_trylock_adaptive                      7747
        12  libc.so.1`mutex_trylock_adaptive                      7762
        14  libc.so.1`mutex_trylock_adaptive                      7822
        10  libc.so.1`mutex_trylock_adaptive                      7843
```

Where's the lock call coming from?

```
# dtrace -n 'pid$target:libc:mutex_trylock_adaptive:entry
                    { @[tid, ustack()] = count(); }' -p `pgrep ldr`
dtrace: description 'pid$target:libc:mutex_trylock_adaptive:entry ' matched 1 pro
^C
. . .
         8
               libc.so.1`mutex_trylock_adaptive
               libc.so.1`lmutex_lock+0xb8
               libc.so.1`lrand48+0x24
               ldr`pt1+0x114
               libc.so.1`_lwp_start
            334990
         8
               libc.so.1`mutex_trylock_adaptive
               libc.so.1`lmutex_lock+0xb8
               libc.so.1`lrand48+0x24
               ldr`pt1+0x144
               libc.so.1`_lwp_start
            335136
```

Where's the lock call coming from?

```
# dtrace -n 'pid$target:libc.so:mutex_trylock_adaptive:entry
                    { @[ufunc(ucaller)] = count();  }' -p `pgrep ldr`
dtrace: description 'pid$target:libc.so:mutex_trylock_adaptive:entry '
matched 1 probe
^C

  libc.so.1`lmutex_lock                                          4014141

# dtrace -n 'pid$target:libc.so:lmutex_lock:entry
                    { @[ufunc(ucaller)] = count();  }' -p `pgrep ldr`
dtrace: description 'pid$target:libc.so:lmutex_lock:entry '
matched 1 probe
^C

  libc.so.1`call_user_handler                                          7
  libc.so.1`lrand48                                              3786675
```

Another view of the lock time

```
#prstat -Lmp `pgrep ldr`

PID USERNAME USR SYS TRP TFL DFL LCK SLP LAT VCX ICX SCL SIG PROCESS/LWPID
  1326 root   73 1.0 0.0 0.0 0.0  25 0.0 0.3  3K 391  7K   0 ldr/2
  1326 root   72 1.0 0.0 0.0 0.0  26 0.0 0.3  3K 325  7K   0 ldr/8
  1326 root   72 1.0 0.0 0.0 0.0  26 0.0 0.3  3K 348  7K   0 ldr/14
  1326 root   71 0.9 0.1 0.0 0.0  27 0.0 0.3  3K 415  7K   0 ldr/10
  1326 root   71 1.0 0.0 0.0 0.0  28 0.0 0.3  3K 370  7K   0 ldr/6
  1326 root   69 0.9 0.0 0.0 0.0  30 0.0 0.3  3K 394  7K   0 ldr/12
  1326 root   60 0.8 0.1 0.0 0.0  38 0.0 0.4  3K 338  6K   0 ldr/4
  1326 root   57 0.8 0.1 0.0 0.0  42 0.0 0.4  3K 339  6K   0 ldr/16
  1326 root  0.2  33 0.0 0.0 0.0 0.0  67 0.4  3K 315 14K   0 ldr/3
  1326 root  0.2  32 0.0 0.0 0.0 0.0  68 0.4  3K 264 14K   0 ldr/15
  1326 root  0.3  31 0.0 0.0 0.0 0.0  68 0.4  4K 298 17K   0 ldr/5
  1326 root  0.2  31 0.0 0.0 0.0 0.0  68 0.4  3K 341 14K   0 ldr/17
  1326 root  0.3  30 0.0 0.0 0.0 0.0  69 0.5  4K 279 17K   0 ldr/13
  1326 root  0.3  30 0.0 0.0 0.0 0.0  69 0.4  4K 277 17K   0 ldr/9
  1326 root  0.3  29 0.0 0.0 0.0 0.0  70 0.4  3K 268 16K   0 ldr/11
  1326 root  0.2  28 0.0 0.0 0.0 0.0  71 0.5  4K 308 15K   0 ldr/7
  1326 root  0.0 0.0 0.0 0.0 0.0 0.0 100 0.0   5   0  20   5 ldr/1
```

Another view of the lock time

```
# plockstat -A -p `pgrep ldr`
^C
Mutex hold

Count     nsec Lock                         Caller
-------------------------------------------------------------------------------
103062     4261 libc.so.1`seed_lock          ldr`pt1+0x114
103036     4261 libc.so.1`seed_lock          ldr`pt1+0x144
  436    10177 libc.so.1`seed_lock          ldr`pt1+0x144
  411     9979 libc.so.1`seed_lock          ldr`pt1+0x114
    4    76125 libc.so.1`_iob+0xa8          ldr`timer+0x38
    7    10414 libc.so.1`libc_malloc_lock   ldr`funcb+0x20
    7     9814 libc.so.1`libc_malloc_lock   ldr`funcb+0xa0
    7     5785 libc.so.1`libc_malloc_lock   ldr`pt1+0x1bc
    7     4885 libc.so.1`libc_malloc_lock   ldr`pt1+0x420
    3     4133 libc.so.1`_uberdata+0x500    libc.so.1`do_exit_critical+0xcc
    1    12300 libc.so.1`_uberdata+0x500    ldr`write_test+0x288

Mutex block

Count     nsec Lock                         Caller
-------------------------------------------------------------------------------
 1736  5855034 libc.so.1`seed_lock          ldr`pt1+0x114
 1805  5487603 libc.so.1`seed_lock          ldr`pt1+0x144
 1682   788333 libc.so.1`seed_lock          ldr`pt1+0x114
 1650   752148 libc.so.1`seed_lock          ldr`pt1+0x144
```

Where are the threads spending %usr time?
Code change – removed lrand48()

```
# dtrace -n 'profile-997hz / arg1 && execname == "ldr" /
                                { @[tid, ufunc(arg1)] = count(); }'
dtrace: description 'profile-997hz ' matched 1 probe
^C
. . .

        12  libc.so.1`gethrtime                                  2434
        14  ldr`pt1                                              3246
        10  ldr`pt1                                              3254
         6  ldr`pt1                                              3369
         4  ldr`pt1                                              3406
        16  ldr`pt1                                              3559
         8  ldr`pt1                                              3590
        12  ldr`pt1                                              3712
         2  ldr`pt1                                              3794
        14  ldr`funcb                                           14135
         2  ldr`funcb                                           14161
        12  ldr`funcb                                           14184
        16  ldr`funcb                                           14561
         8  ldr`funcb                                           14573
         4  ldr`funcb                                           14694
         6  ldr`funcb                                           14817
        10  ldr`funcb                                           15152
```

Another view of the lock time
After code change

```
#prstat -Lmp `pgrep ldr`

   PID USERNAME USR SYS TRP TFL DFL LCK SLP LAT VCX ICX SCL SIG PROCESS/LWPID
  1567 root     100 0.0 0.0 0.0 0.0 0.0 0.0 0.0   0  35   0   0 ldr/4
  1567 root     100 0.0 0.0 0.0 0.0 0.0 0.0 0.0   0  26   1   0 ldr/6
  1567 root     100 0.0 0.0 0.0 0.0 0.0 0.0 0.0   0  25   0   0 ldr/16
  1567 root     100 0.0 0.0 0.0 0.0 0.0 0.0 0.0   0  27   0   0 ldr/10
  1567 root     100 0.0 0.0 0.0 0.0 0.0 0.0 0.0   1  36   2   0 ldr/8
  1567 root     100 0.0 0.0 0.0 0.0 0.0 0.0 0.0   0  36   0   0 ldr/2
  1567 root     100 0.0 0.0 0.0 0.0 0.0 0.0 0.0   1  34   1   0 ldr/12
  1567 root     100 0.0 0.0 0.0 0.0 0.0 0.0 0.0   0  35  12   4 ldr/14
  1567 root     0.2  23 0.0 0.0 0.0 0.0  76 0.0  27  91 16K   0 ldr/11
  1567 root     0.2  23 0.0 0.0 0.0 0.0  77 0.0  41  87 16K   0 ldr/3
  1567 root     0.2  22 0.0 0.0 0.0 0.0  78 0.0  33  98 15K   0 ldr/17
  1567 root     0.2  21 0.0 0.0 0.0 0.0  78 0.0  28  86 16K   0 ldr/9
  1567 root     0.2  21 0.0 0.0 0.0 0.0  78 0.0  21  93 14K   0 ldr/7
  1567 root     0.2  21 0.0 0.0 0.0 0.0  79 0.0  30  72 15K   0 ldr/15
  1567 root     0.2  20 0.0 0.0 0.0 0.0  79 0.0  24  91 14K   0 ldr/13
  1567 root     0.2  18 0.0 0.0 0.0 0.0  81 0.0  17  85 14K   0 ldr/5
  1567 root     0.0 0.0 0.0 0.0 0.0 0.0 100 0.0   5   0   4   1 ldr/1
```

%sys – where's the kernel spending time?

```
# dtrace -n 'profile-997hz / arg0 && curthread->t_pri != -1 /
                { @k[stack()] = count(); } END { trunc(@k,10); printa(@k); }'
dtrace: description 'profile-997hz ' matched 2 probes
^C
CPU     ID                    FUNCTION:NAME
 43      2                                 :END
              px`px_dvma_map_fast+0xc4
              px`px_dma_bindhdl+0x130
              genunix`ddi_dma_buf_bind_handle+0x118
              fcp`ssfcp_prepare_pkt+0x234
              fcp`ssfcp_scsi_start+0x148
              emcpsf`pp_filter_transport+0x624
              ssd`ssd_start_cmds+0x4e8
  . . . . . . . . . . .
              emcp`power_strategy+0xdc
              md`mdstrategy+0xd4
              ufs`lufs_write_strategy+0x11c
              ufs`ufs_putapage+0x308
              ufs`ufs_putpages+0x2a4
              genunix`fop_putpage+0x1c
              652

              FJSV,SPARC64-VII`copyin_more+0x1ac
              genunix`uiomove+0xa8
              ufs`wrip+0x610
              ufs`ufs_write+0x580
              genunix`fop_write+0x20
              genunix`write+0x268
              unix`syscall_trap+0xac
             1900
```

%sys – where's the kernel spending time?

```
# cat kprof.d
#!/usr/sbin/dtrace -s

#pragma D option quiet

profile-997hz
/ arg0 && curthread->t_pri != -1 /
{
        @k[func(caller), func(arg0)] = count();

}
END
{
        trunc(@k, 16);
        printf("%-32s %-32s %-8s\n","CALLER","FUNCTION","COUNT");
        printa("%-32a %-32a %-@8d\n",@k);
}
```

%sys – where's the kernel spending time?

```
# ./kprof.d
^C
CALLER                                  FUNCTION                                COUNT
unix`page_trylock                       unix`mutex_enter                          448
unix`page_list_add                      unix`page_add_common                      463
genunix`swap_getconpage                 unix`page_lookup_create                   465
ufs`wrip                                genunix`segmap_getmapflt                  534
genunix`segmap_release                  genunix`segmap_smapadd                    541
genunix`segmap_getmapflt                genunix`get_free_smp                      650
unix`sfmmu_mlspl_enter                  unix`mutex_enter                          832
genunix`segmap_pagefree                 unix`page_lookup_nowait                   884
px`px_dma_bindhdl                       px`px_dvma_map_fast                       925
ufs`ufs_getpage                         unix`page_lookup_create                   926
genunix`grab_smp                        genunix`segmap_hashout                   1075
unix`mutex_vector_enter                 platmod`plat_lock_delay                  1076
ufs`rdip                                genunix`segmap_getmapflt                 1512
genunix`uiomove                         FJSV,SPARC64-VII`copyin_more             1750
unix`pagezero                           FJSV,SPARC64-VII`hwblkclr                6208
genunix`uiomove                         FJSV,SPARC64-VII`copyout_more            6938
```

cross calls - xcalls

```
# dtrace -l | grep xcall
  354    sysinfo                 unix                                xc_all xcalls
  355    sysinfo                 unix                               xc_some xcalls
  683    sysinfo  FJSV,SPARC64-VII                    send_one_mondo xcalls
  684    sysinfo  FJSV,SPARC64-VII                    send_mondo_set xcalls
 2287        fbt                 unix                         cbe_xcall entry
 2288        fbt                 unix                         cbe_xcall return
 2299        fbt                 unix                cbe_xcall_handler entry
 2300        fbt                 unix                cbe_xcall_handler return
 6580        fbt              genunix              cyclic_expand_xcall entry
 6581        fbt              genunix              cyclic_expand_xcall return
 6586        fbt              genunix                 cyclic_add_xcall entry
 6587        fbt              genunix                 cyclic_add_xcall return
 6590        fbt              genunix              cyclic_remove_xcall entry
 6591        fbt              genunix              cyclic_remove_xcall return
 6612        fbt              genunix            cyclic_suspend_xcall entry
 6613        fbt              genunix            cyclic_suspend_xcall return
 6614        fbt              genunix              cyclic_resume_xcall entry
 6615        fbt              genunix              cyclic_resume_xcall return

# dtrace -n 'sysinfo:::xcalls { @[stack()]=count(); }'
```

xcalls

```
# dtrace -n 'sysinfo:::xcalls { @[stack()]=count(); }'
dtrace: description 'sysinfo:::xcalls ' matched 4 probes
^C
              FJSV,SPARC64-VII`send_one_mondo+0x20
              unix`xt_one_unchecked+0xc8
              unix`setbackdq+0x314
              genunix`sema_v+0x88
              ufs`alloccg+0x1f0
              ufs`hashalloc+0x24
              ufs`alloc+0x128
              ufs`bmap_write+0xc40
              ufs`wrip+0x4b8
              ufs`ufs_write+0x580
              genunix`fop_write+0x20
              genunix`write+0x268
              unix`syscall_trap+0xac
            11575

              FJSV,SPARC64-VII`send_one_mondo+0x20
              unix`xt_one_unchecked+0xc8
              genunix`sleepq_wakeone_chan+0x70
              genunix`cv_signal+0x58
              genunix`lwp_unpark+0x44
              genunix`syslwp_park+0x64
              unix`syscall_trap+0xac
            38024
```

xcalls (cont)

```
              FJSV,SPARC64-VII`send_one_mondo+0x20
              unix`xt_one_unchecked+0xc8
              genunix`sema_v+0x88
              ufs`alloccg+0x1f0
              ufs`hashalloc+0x24
              ufs`alloc+0x128
              ufs`bmap_write+0xc40
              ufs`wrip+0x4b8
              ufs`ufs_write+0x580
              genunix`fop_write+0x20
              genunix`write+0x268
              unix`syscall_trap+0xac
             87403

              FJSV,SPARC64-VII`send_one_mondo+0x20
              unix`xt_one_unchecked+0xc8
              genunix`sleepq_wakeone_chan+0x70
              genunix`cv_signal+0x58
              emlxs`emlxs_thread_trigger2+0x130
              emlxs`emlxs_handle_ring_event+0x998
              emlxs`emlxs_proc_attention+0x168
              emlxs`emlxs_msi_intr+0x1f8
              px`px_msiq_intr+0x1e8
              unix`intr_thread+0x168
            224009
```

Interrupts (intr)

```
# dtrace -n '*intr*:entry { @i[probefunc] = count(); }'
dtrace: description '*intr*:entry ' matched 311 probes
^C

  so_lock_read_intr                                 1
  scf_softintr                                      4
  scsi_watch_request_intr                           5
  pcmu_intr_wrapper                                12
  scf_dscp_intr                                    12
  scf_intr                                         12
  sdintr                                           26
  mpt_intr                                         31
  mpt_process_intr                                 31
  ddi_intr_trigger_softint                         36
  ddi_trigger_softintr                             36
  intr_passivate                                   73
  resume_from_intr                                 73
  bge_intr                                         82
  px_intx_intr                                     82
  cpu_intr_swtch_enter                            146
  ssdintr                                      271904
  emlxs_msi_intr                               281711
  px_msiq_intr                                 281743
  hvio_intr_setstate                           281824
  px_lib_intr_setstate                         281824
  poke_cpu_intr                                367456
```

Misc Examples

Chasing user process memory allocations

Disk IO Latency

Disk IO Latency

sd112 (227,7168), us:

Using the fds[] variable

```
# cat fds.d

#!/usr/sbin/dtrace -s

#pragma D option quiet

syscall::write:entry
/fds[arg0].fi_oflags & O_APPEND/
{
        printf("%s appending file %s at offset %d\n",
            execname, fds[arg0].fi_pathname, fds[0].fi_offset);
}

# ./fds.d
ksh appending file /.sh_history at offset 349345
ksh appending file /.sh_history at offset 349378
```

dtrace -lv – probe args

```
# dtrace -lv -n fbt::ufs_write:entry
   ID   PROVIDER            MODULE                          FUNCTION NAME
27473        fbt               ufs                        ufs_write entry

        Probe Description Attributes
                Identifier Names: Private
                Data Semantics:   Private
                Dependency Class: Unknown

        Argument Attributes
                Identifier Names: Private
                Data Semantics:   Private
                Dependency Class: ISA

        Argument Types
                args[0]: struct vnode *
                args[1]: struct uio *
                args[2]: int
                args[3]: cred_t *
                args[4]: caller_context_t *
```

network address functions - inet_ntoa()/inet_ntop()

```
#cat inet.d

fbt::ip_tcp_input:entry
{
    @[inet_ntoa(&args[1]->ipha_src),
      inet_ntoa(&args[1]->ipha_dst)] = count();
}
END
{
    printa("%30s-> %-30s %@d\n",@);
}

#./inet.d
^c
    212.58.227.137-> 129.156.38.34    3
    212.58.226.20-> 129.156.38.34     59
    212.58.228.8-> 129.156.38.34      99
```

Network one-liners

```
# dtrace -n 'tcp:::receive /args[2]->tcp_dport == 80/ {
        @pkts[args[1]->ip_daddr] = count();
}'
dtrace: description 'tcp:::receive' matched 1 probe
^C

  192.168.1.8                                                     9
  fe80::214:4fff:fe3b:76c8                                       12
  192.168.1.51                                                   32
  10.1.70.16                                                     83
  192.168.7.3                                                   121
  192.168.101.101                                               192

# dtrace -n 'tcp:::accept-established {
                        @[args[2]->tcp_dport] = count(); }'
dtrace: description 'tcp:::accept-established' matched 1 probe
^C

       79                2
       22               14
       80              327
```

Network TCP Connection Latency

```
#!/usr/sbin/dtrace -s

tcp:::connect-request
{
          start[arg1] = timestamp;
}

tcp:::connect-established
/start[arg1]/
{
       @latency["Connect Latency (ns)", args[2]->ip_daddr] =
             quantize(timestamp - start[arg1]);
       start[arg1] = 0;
}
```

Network TCP Connection Latency

```
dtrace: script './tcpconnlat.d' matched 2 probes
^C

  Connect Latency (ns)                                  192.168.1.109
           value  ~-------------- Distribution ~-------------- count
           65536 |                                               0
          131072 |@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@ 3
          262144 |@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@ 2
          524288 |                                               0

  Connect Latency (ns)                                  72.5.124.61
           value  ~-------------- Distribution ~-------------- count
         4194304 |                                               0
         8388608 |@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@ 3
        16777216 |@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@ 1
        33554432 |                                               0
```

Network – socket accepts and IOs

```
# dtrace -n 'syscall::accept*:entry { @[execname] = count(); }'
dtrace: description 'syscall::accept*:entry ' matched 1 probe
^C

  sshd                                                              1
  inetd                                                             2
  httpd                                                            15
```

```
# dtrace -n 'syscall::read*:entry /fds[arg0].fi_fs == "sockfs"/
                                  { @[probefunc] = count(); }'
dtrace: description 'syscall::read*:entry ' matched 3 probes
^C

  readv                                                             2
  read                                                           2450
```

Network – socket IOs by Process

```
# dtrace -n 'syscall::read*:entry /fds[arg0].fi_fs == "sockfs"/
                    { @[execname] = count(); }'
dtrace: description 'syscall::read*:entry ' matched 3 probes
^C

  FvwmButtons                                                     2
  FvwmIconMan                                                     2
  finger                                                          2
  xbiff                                                           2
  xclock                                                          2
  xload                                                           8
  gconfd-2                                                       10
  opera                                                          16
  firefox-bin                                                    44
  soffice.bin                                                    89
  ssh                                                            94
  FvwmPager                                                     123
  gnome-terminal                                                762
  fvwm2                                                        1898
  realplay.bin                                                 2493
  Xorg                                                         3785
```

Network – socket read bytes by Process

```
 # dtrace -n 'syscall::read:entry /fds[arg0].fi_fs == "sockfs"/
             { self->ok = 1; } syscall::read:return /self->ok/
             { @[execname] = sum(arg0); self->ok = 0; }'
dtrace: description 'syscall::read:entry ' matched 2 probes
^C

  FvwmAnimate                                             124
  xload                                                   128
  soffice.bin                                             288
  opera                                                   384
  FvwmPager                                              1231
  ssh                                                    1312
  gnome-terminal                                         5236
  fvwm2                                                 22206
  realplay.bin                                         360157
  firefox-bin                                         1049057
  Xorg                                                1097685
```

Network – TCP event statistics

```
# dtrace -n 'mib:::tcp* /(int)arg0 > 0/ { @[probename] = sum(arg0); }'
dtrace: description 'mib:::tcp* ' matched 94 probes
^C

  tcpActiveOpens                                                  1
  tcpInDupAck                                                     1
  tcpTimRetrans                                                   1
  tcpOutControl                                                   2
  tcpOutAckDelayed                                               11
  tcpOutAck                                                      24
  tcpInDataInorderSegs                                           33
  tcpInAckSegs                                                 5003
  tcpRttUpdate                                                 5003
  tcpOutDataSegs                                              10002
  tcpInDataInorderBytes                                       24851
  tcpInAckBytes                                            10240157
  tcpOutDataBytes                                          14411804
```

Network – UDP event statistics

```
# dtrace -n 'mib:::udp* { @[probename] = sum(arg0); }'
dtrace: description 'mib:::udp* ' matched 20 probes
^C

  udpIfStatsNoPorts                                             2
  udpHCOutDatagrams                                            46
  udpHCInDatagrams                                             50
```

Network – System-wide socket IO

```
#!/usr/sbin/dtrace -s

#pragma D option quiet

dtrace:::BEGIN
{
        printf("Tracing Socket I/O... Hit Ctrl-C to end.\n");
}

syscall::read*:entry,
syscall::write*:entry,
syscall::send*:entry,
syscall::recv*:entry
/fds[arg0].fi_fs == "sockfs"/
{
      @[execname, pid, probefunc] = count();
}

dtrace:::END
{
      printf("  %-16s %-8s %-16s %10s\n", "PROCESS", "PID", "SYSCALL",
          "COUNT");
      printa("  %-16s %-8d %-16s %@10d\n", @);
}
```

Network – System-wide socket IO

```
# socketio.d
Tracing Socket I/O... Hit Ctrl-C to end.
^C
  PROCESS           PID      SYSCALL                   COUNT
  ssh               864116   write                         1
  sshd              942634   read                          1
  FvwmPager         701861   write                         2
  ssh               864116   read                          4
  xclock            785004   write                         4
  FvwmIconMan       701860   write                         5
  FvwmPager         701865   write                         5
  FvwmPager         701865   read                          7
  sshd              942634   write                         7
  firefox-bin       272642   write                         8
  soffice.bin       453667   read                          8
  soffice.bin       453667   write                         8
  fvwm2             701854   write                        25
  gnome-terminal    701876   write                        37
  firefox-bin       272642   read                         40
  fvwm2             701854   read                         41
  gnome-terminal    701876   read                         49
  Xorg              614773   writev                      100
  Xorg              614773   read                        207
  java              440474   send                      10000
```

DTrace & Java

DTrace and Java

- DTrace can be used to debug and observe Java applications
- Easy to start: use *jstack()*, to display the Java activity as a stack backtrace. *jstack()* based on *ustack()*
- Useful to understand the I/O and scheduling caused by your Java application
- Java 5: VM agents, shared libraries which are dynamically loaded when the VM starts
- Java 6, Mustang, introduces two new providers: hotspot and hotspot_jni

DTrace and Java, cont.

- *jstack()*
- The simplest form to record a stack trace from a Java application
  - Not jstackstrsize default of 512 may need to be increased

- Delivered already with DTrace framework:

```
$ dtrace -n 'syscall:::entry/pid==xxx/
  {jstack(40);}'
```

```
$ dtrace -n 'syscall:::entry/pid==xxx/
  {@[jstack(40)] = count();}'
```

jstack() Action

- jstack action prints mixed mode stack trace
- Both java frames and native (C/C++) frames are shown
- Only JVM versions 5.0_01 and later are supported
- jstack shows hex numbers for JVM versions before 5.0_01

```
#!/usr/sbin/dtrace -s
syscall::pollsys:entry
/ pid == $1 /   {
    jstack(50,8192);
}
```

- first optional argument limits the number of frames shown
- second optional argument changes the string size
- jstackstrsize pragma / -x to increase buffer for all jstack()'s

jstack()

```
libc.so.1`__pollsys+0x7
libc.so.1`pselect+0x19e
libc.so.1`select+0x69
libXt.so.4`IoWait+0x36
libXt.so.4`_XtWaitForSomething+0x1a9
libXt.so.4`XtAppPending+0x188
libmawt.so`0xd43d3928
libmawt.so`0xd43d37d6
libmawt.so`Java_sun_awt_motif_MToolkit_run+0x34
sun/awt/motif/MToolkit.run
java/lang/Thread.run
StubRoutines (1)
libjvm.so`__1cJJavaCallsLcall_helper6FpnJJavaValue_pnMmethodHandle_pnRJavaCallArguments_pnGThread__v_+0x187
libjvm.so`__1cCosUos_exception_wrapper6FpFpnJJavaValue_pnMmethodHandle_pnRJavaCallArguments_pnGThread__v2468_v_+0x14
libjvm.so`__1cJJavaCallsEcall6FpnJJavaValue_nMmethodHandle_pnRJavaCallArguments_pnGThread__v_+0x28
libjvm.so`__1cJJavaCallsMcall_virtual6FpnJJavaValue_nLKlassHandle_nMsymbolHandle_4pnRJavaCallArguments_pnGThread__v_+0xb
libjvm.so`__1cJJavaCallsMcall_virtual6FpnJJavaValue_nGHandle_nLKlassHandle_nMsymbolHandle_5pnGThread__v_+0x6d
libjvm.so`__1cMthread_entry6FpnKJavaThread_pnGThread__v_+0xd0
libjvm.so`__1cKJavaThreadRthread_main_inner6M_v_+0x51
libjvm.so`__1cKJavaThreadDrun6M_v_+0x105
libjvm.so`__1cG_start6Fpv_0_+0xd2
libc.so.1`_thr_setup+0x51
libc.so.1`_lwp_start
397

libc.so.1`stat64+0x7
java/io/UnixFileSystem.getBooleanAttributes0*
0x20245c8b
932
```

The dvm Provider

- java.net project to add DTrace support in
  - 1.4.2 (libdvmpi.so)
  - 1.5 (libdvmti.so)
  - https://solaris10-dtrace-vm-agents.dev.java.net/
- Download shared libs
- Add location of libs to LD_LIBRARY_PATH variable
- Set JAVA_TOOL_OPTIONS to -Xrundvmti:all
- Name of provider - “dvm”

The dvm Provider: Probes

- dvm probes and their signatures

  vm-init(), vm-death()

  thread-start(char *thread_name), thread-end()

  class-load(char *class_name)

  class-unload(char *class_name)

  gc-start(), gc-finish()

  gc-stats(long used_objects, long used_object_space)

  object-alloc(char *class_name, long size)

  object-free(char *class_name)

  method-entry(char *class_name, char *method_name, char *method_signature)

  method__return(char *class_name, char *method_name, char *method_signature)

The dvm Provider: alloc and free

- Object allocation/deallocation

```
#!/usr/sbin/dtrace -qs
dvm$target:::object-alloc
{
  printf("%s allocated %d size objects\n”,
    copyinstr(arg0), arg1);
}

dvm$target:::object-free
{
  printf("%s freed %d size objects\n",
    copyinstr(arg0), arg1);
}

# ./java_alloc.d -p `pgrep -n java`
```

The dvm Provider: Methods

- Count methods called

```
#!/usr/sbin/dtrace -s

dvm$target:::method-entry
{
  @[copyinstr(arg0),copyinstr(arg1)] = count();
}

# ./java_method_count.d -p `pgrep -n java`
```

The dvm provider: Time Spent

- Time spent in methods

```
#!/usr/sbin/dtrace -s
dvm$target:::method-entry
{
  self->ts[copyinstr(arg0),copyinstr(arg1)] =
    vtimestamp;
}

dvm$target:::method-return
{
  @ts[copyinstr(arg0),copyinstr(arg1)] =
    sum(vtimestamp - self->ts[copyinstr(arg0),
      copyinstr(arg1)]);
}

# ./java_method.d -p `pgrep -n java`
```

DTrace and Java, cont.

- VM Agents
  - Some probes have a significant probe effect, and require enabling when the JVM is started

    -XX:+ExtendedDtraceProbes

    jinfo -XX:+ExtendedDtraceProbes

DTrace and Java, cont.

- Java 6, Mustang
  - Added two new providers: hotspot and hotspot_jni
  - Using these providers it is now possible to collect data from your Java applications
  - Hotspot_jni: probes related with Java Native Interface
  - Hotspot provider:
    - VM Probes: Initialization and Shutdown
    - Thread statistics Probes
    - Class loading and unloading Probes
    - Garbage Collection Probes
    - Method Compilation Probes

DTrace in JDK 6

- hotspot provider implements all dvm probes plus extensions:
  - Method compilation (method-compile-begin/end)
  - Compiled method load/unload(compiled-method-load/unload)
  - JNI method probes.
    - DTrace probes as entry and return from each JNI method.
- Strings are now unterminated UTF-8 data. Always use associated length value with copyinstr().

Method Compilation Probes

```
hotspot$1:::method-compile-begin {
    self->str = (char*) copyin(arg2, arg3+1);
    self->str[arg3] = '\0';
    self->classname = (string)self->str;
    self->str = (char*) copyin(arg4, arg5+1);
    self->str[arg5] = '\0';
    self->methodname = (string)self->str;
    printf("Compile begin %s.%s\n",
      self->classname, self->methodname);
}
```

Exception Stack Trace

```
hotspot$1:::method-entry {
    self->ptr = (char*)copyin(arg1, arg2+1);
    self->ptr[arg2] = '\0';
    self->classname = (string)self->ptr;
    self->ptr = (char*)copyin(arg3, arg4+1);
    self->ptr[arg4] = '\0';
    self->methodname = (string)self->ptr;
}
hotspot$1:::method-entry
/self->classname == "java/lang/Throwable" &&
   self->methodname == "<init>"/
{
   jstack();
}
```

JDK 6 DTrace Usage

- Certain probes are expensive
  - Turned off by default
  - object-alloc
  - method-entry, method-return
  - monitor probes
    - monitor-wait, monitor-contended-enter, etc
- Requires you to start your application with the flag

  -XX:+ExtendedDTraceProbes
- Use -XX:+DTrace{Alloc,Method,Monitor}Probes if possible

JDK6 hotspot_jni Provider

- Probes for Java Native Interface (JNI)
- Located at entry/return points of all JNI functions
- Probe arguments are same as corresponding JNI function arguments (for _entry probes)
- For XXX_return probes, probe argument is return value
- Examples:

    hotspot_jni$1:::GetPrimitiveArrayCritical_entry

    hotspot_jni$1:::GetPrimitiveArrayCritical_return

JDK 1.6 and DTrace

- Check out

DTrace Community

- Build around OpenSolaris community
- Available under www.opensolaris.org
  - The main page:
    http://www.opensolaris.org/os/community/dtrace/

    IRC on irc.freenode.net channels: #opensolaris, #dtrace
- The leaders:
  - Bryan M. Cantrill
  - Adam H. Leventhal
  - Mike Shapiro
  - Brendan Gregg
- Working with other communities

<Insert Picture Here>

ORACLE®

Jim Mauro
james.mauro@sun.com

Thank you!

"open" artwork and icons by chandan:
http://blogs.sun.com/chandan

Supplemental Material

DTraceToolkit

- Introduction
- Installation and Setup
- Toolkit Elements
- Categories
- Free your mind
- **Real Examples**

1.High System Calls

- A case where *vmstat 1* reports a high number of system calls
- What to do ?
- Count the total number of system calls
- Use a simple DTrace aggregation to find out what application are responsible for that
- Think to enhance the aggregation for a better reporting or better...
- Use DTT utilities to find out what is going on, getting as well a nice report

1.High System Calls, cont.

kthr			memory			page					disk					faults		cpu		
r	b	w	swap	free	re	mf	pi	po	fr	de	sr	cd	cd	cd	f0	in	sy	cs	us	sy	id
0	0	0	2883592	1077152	28	281	0	0	0	0	0	0	0	0	0	1563	2570	1591	2	1	97
0	0	0	2883592	1077152	28	278	0	0	0	0	0	0	0	0	0	1535	2537	1546	2	1	97
0	0	0	2883592	1077152	28	281	0	0	0	0	0	0	0	0	0	1852	3655	2168	2	2	96
0	0	0	2883592	1077152	28	296	0	0	0	0	0	0	0	0	0	1902	4950	2421	4	2	94
0	0	0	2883592	1077152	28	304	0	0	0	0	0	48	0	0	0	2175	6404	2979	9	2	89
0	0	0	2883584	1077144	28	278	0	0	0	0	0	2	0	0	0	1903	5431	2568	6	2	91
0	0	0	2883584	1077144	28	279	0	0	0	0	0	0	0	0	0	2017	6956	2830	7	2	91
0	0	0	2883584	1077144	28	278	0	0	0	0	0	0	0	0	0	1901	6855	2650	6	2	91
0	0	0	2883584	1077144	28	279	0	0	0	0	0	0	0	0	0	2114	9656	3387	8	3	89
0	0	0	2883584	1077144	28	282	0	0	0	0	0	0	0	0	0	1774	5176	2292	4	2	94
0	0	0	2883584	1077144	27	276	0	0	0	0	0	0	0	0	0	1651	2964	1742	2	2	96
0	0	0	2883584	1077144	28	282	0	0	0	0	0	0	0	0	0	1546	2696	1552	2	1	97
0	0	0	2883584	1077144	28	278	0	0	0	0	0	0	0	0	0	1900	4065	2287	3	2	95
0	0	0	2883584	1077144	28	279	0	0	0	0	0	0	0	0	0	1644	3741	1883	4	1	95
0	0	0	2883584	1077144	28	279	0	0	0	0	0	0	0	0	0	1982	5698	2650	11	2	87
0	0	0	2883584	1077144	28	279	0	0	0	0	0	0	0	0	0	1844	3867	2223	3	1	95
0	0	0	2883584	1077144	28	281	0	0	0	0	0	0	0	0	0	1555	2506	1542	1	1	97
0	0	0	2883584	1077144	28	278	0	0	0	0	0	0	0	0	0	1475	2497	1495	2	1	96
0	0	0	2883584	1077144	28	279	0	0	0	0	0	0	0	0	0	1501	2527	1542	2	1	97
kthr			memory			page					disk					faults		cpu		
r	b	w	swap	free	re	mf	pi	po	fr	de	sr	cd	cd	cd	f0	in	sy	cs	us	sy	id
0	0	0	2883584	1077144	28	279	0	0	0	0	0	0	0	0	0	1517	2531	1539	2	1	97
0	0	0	2883584	1077144	28	279	0	0	0	0	0	0	0	0	0	1494	2510	1464	2	1	97
0	0	0	2883584	1077144	28	279	0	0	0	0	0	0	0	0	0	1517	2576	1585	2	1	97
0	0	0	2883584	1077144	28	281	0	0	0	0	0	0	0	0	0	1813	3656	2180	2	2	96
0	0	0	2883584	1077144	29	281	0	0	0	0	0	0	0	0	0	1472	2475	1482	2	1	97
0	0	0	2883584	1077144	28	278	0	0	0	0	0	0	0	0	0	1465	2508	1468	2	1	96
0	0	0	2883584	1077144	28	279	0	0	0	0	0	0	0	0	0	1500	2491	1564	2	1	97
0	0	0	2883584	1077144	28	279	0	0	0	0	0	0	0	0	0	1510	2526	1541	2	1	96
0	0	0	2883584	1077144	28	279	0	0	0	0	0	0	0	0	0	1480	2534	1477	2	1	97

1.High System Calls, cont.

- Start a simple aggregation:

```
$ dtrace -n 'syscall:::entry{@[execname] =
 count();}'
```

- Select the top consumer and start aggregating again:

```
$ dtrace -n
 'syscall:::entry/execname==”your-app”/
 {@[probefunc] = count();}'
```

- Count the number of system calls globally:

```
$ dtrace -n 'syscall:::entry{@[probefunc]
 = count();}'
```

- Better run *topsysproc* from Proc Category

1.High System Calls, cont.

```
2006 May 25 15:20:43, load average: 0.15, 0.12, 0.10   syscalls: 2552

  PROCESS                          COUNT
  httpd                                3
  xscreensaver                         9
  mixer_applet2                       10
  nscd                                10
  gnome-netstatus-                    12
  intrd                               15
  java                                18
  gnome-panel                         31
  webservd                            43
  tput                                49
  vmstat                              51
  gnome-terminal                      62
  dtrace                              65
  soffice.bin                         69
  at-spi-registryd                    72
  sh                                 122
  clear                              136
  Xorg                               151
  firefox-bin                        151
  realplay.bin                      1473
```

1.High System Calls, cont.

- **Conclusions:**
  - Not able to see who does all those system calls using basic utilities: *vmstat*, *iostat*, *prstat*
  - Easy to detect and get the report about the top system calls consumers using DTT utility: *topsysproc*

2.High CPU Utilization

- There is a high CPU utilisation under the system without any sign who is generating that
- What to do ?
- Does it help to run: *prstat*, *mpstat*, *vmstat*, *iostat* ?
- Solve the problem by using: *topsysproc*, and *execsnoop* from DTT

2.High CPU Utilisation, cont.

- The output from *vmstat 1*:

```
kthr      memory            page            disk          faults      cpu
r b w   swap  free  re  mf pi po fr de sr cd cd cd f0   in   sy   cs us sy id
0 0 0 2791884 983816 13 169 2  0  0  0  2  0  0  0  0  903 1550  805  2  1 98
0 0 0 2762448 973096 5510 74407 0 0 0 0 0  0  0  0  0 2847 51987 5458 14 44 43
0 0 0 2762448 973124 5429 73284 0 0 0 0 0  0  0  0  0 2741 51068 5333 15 43 42
0 0 0 2762548 973096 5445 73504 0 0 0 0 0  1  0  0  0 2710 51428 5335 13 45 42
0 0 0 2762432 973084 5446 73548 0 0 0 0 0  0  0  0  0 2758 51364 5343 14 43 43
0 0 0 2762476 973044 5454 73573 0 0 0 0 0  0  0  0  0 2791 51366 5433 14 43 42
0 0 0 2762576 973128 5459 73745 0 0 0 0 0  0  0  0  0 2776 51501 5408 14 44 42
0 0 0 2762576 973128 5514 74416 0 0 0 0 0  0  0  0  0 2821 51881 5429 14 43 44
0 0 0 2762468 973032 5419 73135 0 0 0 0 0  0  0  0  0 2774 51382 5331 15 43 42
0 0 0 2762476 973040 5485 74017 0 0 0 0 0  0  0  0  0 2806 51692 5438 13 43 43
0 0 0 2762540 973092 5431 73348 0 0 0 0 0  0  0  0  0 2757 51242 5332 14 43 42
0 0 0 2762504 973080 5493 74114 0 0 0 0 0  0  0  0  0 2771 51682 5407 14 43 43
0 0 0 2762440 973100 5431 73367 0 0 0 0 0  3  0  0  0 2784 51210 5365 14 43 42
1 0 0 2762576 973128 5446 73504 0 0 0 0 0  0  0  0  0 2765 51299 5336 14 43 43
0 0 0 2762448 973128 5438 73422 0 0 0 0 0  0  0  0  0 2863 51713 5629 14 44 43
0 0 0 2762564 973116 5441 73401 0 0 0 0 0  0  0  0  0 2835 52062 5700 15 43 42
0 0 0 2762432 973084 5428 73341 0 0 0 0 0  0  0  0  0 2850 51972 5662 14 44 42
0 0 0 2762500 973064 4656 63220 0 0 0 0 0  0  0  0  0 2644 52488 6327 28 41 31
```

2.High CPU Utilisation, cont.

- The output from *mpstat 1*:

```
CPU minf mjf xcal  intr ithr  csw icsw migr smtx  srw syscl  usr sys  wt idl
  0 51776   0    0  1446  559 2109  130  612  574    0 32116   19  56   0  25
  1 21034   0    0  1165    9 3126  129  352  285    0 18848   14  30   0  56
CPU minf mjf xcal  intr ithr  csw icsw migr smtx  srw syscl  usr sys  wt idl
  0 58151   0    0  1374  546 1975  107  623  648    0 35360   19  62   0  20
  1 14682   0    0  1230   10 3236   88  282  245    0 15526   13  24   0  63
CPU minf mjf xcal  intr ithr  csw icsw migr smtx  srw syscl  usr sys  wt idl
  0 53541   0    0  1324  552 1828  139  608  589    0 32613   26  56   0  18
  1 18246   0    1  1163   17 3291  135  286  238    0 18093   15  29   0  56
CPU minf mjf xcal  intr ithr  csw icsw migr smtx  srw syscl  usr sys  wt idl
  0 45416   0    0  1516  551 2257  142  548  572    0 29019   19  50   0  31
  1 28010   0    0  1168   10 3081  121  397  349    0 22440   12  36   0  52
```

2.High CPU Utilisation, cont.

- The output from *prstat -a*:

```
   PID USERNAME  SIZE   RSS STATE   PRI NICE      TIME  CPU PROCESS/NLWP
  8120 sparvu   1204K  716K run       0    4   0:01:51 5.9% ksh/1
  2961 root      197M  174M sleep    59    0   0:46:03 3.6% Xorg/1
  3169 sparvu     70M   32M sleep    59    0   0:05:40 2.5% gnome-terminal/2
  6971 sparvu     63M   14M sleep    59    0   0:04:11 1.3% realplay.bin/1
  6765 sparvu    129M   77M sleep    59    0   0:04:26 0.3% firefox-bin/3
  1922 root     5752K 4544K sleep    59    0   0:20:49 0.2% intrd/1
  7068 sparvu    249M  134M sleep    49    0   0:02:43 0.1% soffice.bin/5
  3291 sparvu     44M 7836K sleep    59    0   0:01:53 0.1% at-spi-registry/1
  3154 sparvu     50M   12M sleep    59    0   0:01:32 0.0% gnome-netstatus/1
  1967 root      102M   36M sleep    29   10   0:01:14 0.0% webservd/31
  1984 webservd  142M   54M sleep    59    0   0:01:11 0.0% webservd/76
 23319 sparvu   3416K 2872K cpu0     59    0   0:00:00 0.0% prstat/1
  1810 noaccess  177M   65M sleep    59    0   0:00:56 0.0% java/28
  3133 sparvu     49M   14M sleep    59    0   0:00:46 0.0% metacity/1
  3152 sparvu     51M   14M sleep    59    0   0:00:31 0.0% wnck-applet/1
 27399 sparvu     71M   36M sleep    37    4   0:00:13 0.0% gimp-2.0/1
  3156 sparvu     48M   11M sleep    59    0   0:00:25 0.0% mixer_applet2/1
  6983 sparvu   1204K  908K sleep    59    0   0:00:00 0.0% ksh/1
  3137 sparvu     56M   19M sleep    59    0   0:00:31 0.0% gnome-panel/1
  3111 sparvu   6052K 3036K sleep    59    0   0:00:06 0.0% xscreensaver/1
  3116 sparvu   8148K 3740K sleep    59    0   0:00:05 0.0% gnome-smproxy/1
   360 root     4660K 1848K sleep    59    0   0:00:00 0.0% automountd/2
   480 root     1736K  544K sleep    59    0   0:00:00 0.0% smcboot/1
   478 root     1740K  944K sleep    59    0   0:00:00 0.0% smcboot/1
NPROC USERNAME  SIZE   RSS MEMORY      TIME  CPU
   57 sparvu   1491M  543M     27%   0:26:49  10%
   39 root      560M  307M     15%   1:08:55 3.8%
    1 webservd  142M   54M    2.6%   0:01:11 0.0%
    1 noaccess  177M   65M    3.2%   0:00:56 0.0%
    1 smmsp    6872K 1480K    0.1%   0:00:00 0.0%
    1 lp       2936K 1084K    0.1%   0:00:00 0.0%
    4 daemon     11M 6004K    0.3%   0:00:00 0.0%
Total: 104 processes, 371 lwps, load averages: 1.36, 1.30, 0.96
```

ORACLE

2.High CPU Utilisation, cont.

- Run *topsysproc*:

```
2006 May 28 17:43:08, load average: 0.56, 0.22, 0.12   syscalls: 46333

  PROCESS                          COUNT
  gnome-vfs-daemon                     3
  httpd                                3
  mixer_applet2                        8
  xscreensaver                         9
  gnome-netstatus-                    10
  intrd                               15
  java                                20
  gnome-panel                         31
  mpstat                              35
  tput                                49
  webservd                            49
  dtrace                              62
  firefox-bin                         84
  soffice.bin                        108
  sh                                 122
  clear                              136
  Xorg                               628
  gnome-terminal                    2455
  ksh                               9727
  date                             32778
```

2.High CPU Utilisation, cont.

- Run *execsnoop*:

```
sparvu@earth>./execsnoop
  UID    PID   PPID ARGS
  100  13575   2540 date
  100  13576   2540 date
  100  13577   2540 date
  100  13578   2540 date
  100  13579   2540 date
  100  13580   2540 date
  100  13581   2540 date
  100  13582   2540 date
  100  13583   2540 date
  100  13584   2540 date
  100  13585   2540 date
  100  13586   2540 date
  100  13587   2540 date
  100  13588   2540 date
  100  13589   2540 date
  100  13590   2540 date
  100  13591   2540 date
  100  13592   2540 date
  100  13593   2540 date
  100  13594   2540 date
  100  13595   2540 date
  100  13596   2540 date
  100  13597   2540 date
```

2.High CPU Utilisation, cont.

- **Conclusions:**
  - A high CPU utilisation was detected by *vmstat* and *prstat*. However the CPU consumption was not easy related to any process on the system
  - Using DTT utilities: *topsysproc* and *execsnoop* the real problem was very easily found and the process/owner generating all the load was easy identified

3.High Cross-Calls

- It has been detected on a multiprocessor server a high number of inter-processor cross-calls per second. This was discovered using *mpstat*
- Inter-processor cross-calls is a number indicating how often CPUs are sending the work from one to another. A clear indication of overhead
- Investigate using *mpstat* and see if it is easy to find out who generates all these cross-calls
- Solve the problem by using: *xcallsbypid.d* from DTT Cpu category

3.High Cross-Calls, cont.

- *mpstat* reports:

CPU	minf	mjf	xcal	intr	ithr	csw	icsw	migr	smtx	srw	syscl	usr	sys	wt	idl
0	0	0	0	494	371	260	1	36	1	0	307	1	1	0	98
1	0	0	0	125	3	325	8	48	2	0	552	1	0	0	99
CPU	minf	mjf	xcal	intr	ithr	csw	icsw	migr	smtx	srw	syscl	usr	sys	wt	idl
0	0	0	0	517	380	371	2	76	9	0	839	2	0	0	98
1	0	0	0	152	5	406	4	69	7	0	817	2	1	0	97
CPU	minf	mjf	xcal	intr	ithr	csw	icsw	migr	smtx	srw	syscl	usr	sys	wt	idl
0	0	0	4	506	384	279	6	52	4	0	306	1	0	0	99
1	0	0	1	154	10	312	6	50	1	0	272	0	0	0	100
CPU	minf	mjf	xcal	intr	ithr	csw	icsw	migr	smtx	srw	syscl	usr	sys	wt	idl
0	0	0	0	684	443	431	13	60	13	0	702	10	1	0	89
1	0	0	0	288	7	714	19	63	2	0	906	5	1	0	94
CPU	minf	mjf	xcal	intr	ithr	csw	icsw	migr	smtx	srw	syscl	usr	sys	wt	idl
0	171	93	5915	4832	736	2227	318	117	392	0	143341	13	37	0	50
1	573	62	3507	7098	5	4971	648	128	247	0	54178	23	19	0	58
CPU	minf	mjf	xcal	intr	ithr	csw	icsw	migr	smtx	srw	syscl	usr	sys	wt	idl
0	0	2	3089	4004	410	3263	468	126	3364	0	79532	16	47	0	38
1	0	4	2715	4010	9	3296	541	121	3500	0	83183	16	49	0	36
CPU	minf	mjf	xcal	intr	ithr	csw	icsw	migr	smtx	srw	syscl	usr	sys	wt	idl
0	0	0	3274	5373	391	2660	377	148	1904	0	67229	16	37	0	47
1	2	0	4236	4169	5	3172	683	142	2076	0	88053	18	53	0	29

3.High Cross-Calls, cont.

- Run *xcallsbypid.d* from Cpu category:

```
Tracing... Hit Ctrl-C to end.
^C
   PID CMD                          XCALLS
 11257 ksh                              1
 11258 ksh                              1
 11259 ksh                              1
 11260 ksh                              1
 11255 ksh                              2
 11256 ksh                              2
  2540 ksh                              3
  2163 gnome-panel                      7
 11254 dtrace                          15
  1922 intrd                           27
  2372 mpstat                          27
     0 sched                           46
 11255 find                         27329
```

3.High Cross-Calls, cont.

- **Conclusions:**
  - Solaris's *mpstat* was used to identify the high xcalls, however *mpstat* was not reporting on who was generating that big number
  - Very easy to identify the process/application which was generating lots of cross calls directly using DTT utility: *xcallsbypid.d*

4.Network Connections

- The network status utility *netstat* displays a status of all network connections on a system
- With the current tools there is no easy way to find out and co-relate a network connection with a process or the owner of it
- Extra tools like *lsof* can list what connections were made and by who
- What about incoming connections ?
- Solve the problem by using: *tcptop*, *tcpsnoop* and *connections* utilities from DTT

4.Network Connections, cont.

- Under Net category execute: *tcpsnoop*

```
UID    PID LADDR          LPORT DR RADDR          RPORT  SIZE CMD
100  11336 192.168.1.5    42931 -> 212.58.224.163   554    54 realplay.bin
100  11336 192.168.1.5    42931 -> 212.58.224.163   554   470 realplay.bin
100  11336 192.168.1.5    42931 <- 212.58.224.163   554    66 realplay.bin
100  11336 192.168.1.5    42931 -> 212.58.224.163   554    54 realplay.bin
100  11336 192.168.1.5    42931 <- 212.58.224.163   554    54 realplay.bin
100  11336 192.168.1.5    42931 <- 212.58.224.163   554   381 realplay.bin
100  11336 192.168.1.5    42931 -> 212.58.224.163   554    54 realplay.bin
100  11336 192.168.1.5    42931 -> 212.58.224.163   554   511 realplay.bin
100  11336 192.168.1.5    42931 <- 212.58.224.163   554    54 realplay.bin
100  11336 192.168.1.5    42931 <- 212.58.224.163   554  1514 realplay.bin
100  11336 192.168.1.5    42931 -> 212.58.224.163   554    54 realplay.bin
100  11336 192.168.1.5    42931 <- 212.58.224.163   554  1514 realplay.bin
100  11336 192.168.1.5    42931 <- 212.58.224.163   554   632 realplay.bin
100  11336 192.168.1.5    42931 -> 212.58.224.163   554    54 realplay.bin
100  11336 192.168.1.5    42931 -> 212.58.224.163   554   624 realplay.bin
100  11336 192.168.1.5    42931 -> 212.58.224.163   554   226 realplay.bin
100  11336 192.168.1.5    42931 <- 212.58.224.163   554   307 realplay.bin
100  11336 192.168.1.5    42931 <- 212.58.224.163   554   137 realplay.bin
100  11336 192.168.1.5    42931 -> 212.58.224.163   554   271 realplay.bin
100  11336 192.168.1.5    42931 <- 212.58.224.163   554   236 realplay.bin
100  11336 192.168.1.5    42931 -> 212.58.224.163   554    54 realplay.bin
100  11336 192.168.1.5    42931 -> 212.58.224.163   554   231 realplay.bin
100  11336 192.168.1.5    42931 <- 212.58.224.163   554   137 realplay.bin
100  11336 192.168.1.5    42931 -> 212.58.224.163   554    54 realplay.bin
100   2287 192.168.1.5    52043 -> 72.5.124.61       80    54 firefox-bin
100   2287 192.168.1.5    52043 -> 72.5.124.61       80   706 firefox-bin
100  11336 192.168.1.5    42931 -> 212.58.224.163   554   231 realplay.bin
100   2287 192.168.1.5    52043 <- 72.5.124.61       80    66 firefox-bin
100   2287 192.168.1.5    52043 -> 72.5.124.61       80    54 firefox-bin
100  11336 192.168.1.5    42931 <- 212.58.224.163   554   137 realplay.bin
100  11336 192.168.1.5    42931 -> 212.58.224.163   554    54 realplay.bin
100   2287 192.168.1.5    52043 <- 72.5.124.61       80    54 firefox-bin
```

4.Network Connections, cont.

- To display top network packets run *tcptop*:

```
2006 May 28 18:31:34,  load: 0.28,  TCPin:    104 KB,  TCPout:     20 KB

 UID    PID LADDR           LPORT RADDR           RPORT      SIZE NAME
 100   2287 192.168.1.5     52155 65.205.8.181       80      1078 firefox-bin
 100  11359 192.168.1.5     43839 212.58.227.71      80      1331 realplay.bin
 100  11359 192.168.1.5     59306 212.58.224.54     554      1672 realplay.bin
 100   2287 192.168.1.5     36402 72.5.124.59        80      2730 firefox-bin
 100   2287 192.168.1.5     58374 216.52.17.7        80      2983 firefox-bin
 100   2287 192.168.1.5     39219 72.5.124.59        80      4420 firefox-bin
 100   2287 192.168.1.5     44541 72.5.124.61        80      8753 firefox-bin
 100   2287 192.168.1.5     48599 72.5.124.61        80     19620 firefox-bin
 100   2287 192.168.1.5     64240 212.58.227.71      80     24082 firefox-bin
 100   2287 192.168.1.5     47685 72.5.124.61        80     47258 firefox-bin
 100   2287 192.168.1.5     56155 212.58.227.71      80     49685 firefox-bin
```

4.Network Connections, cont.

- To monitor and check the incoming connections run *connections*:

```
UID   PID CMD          TYPE  PORT IP_SOURCE
  0   266 inetd         tcp    23 192.168.1.3
  0   422 sshd          tcp    22 192.168.1.3
 80  1984 webservd      tcp    80 192.168.1.3
  0   422 sshd          tcp    22 192.168.1.3
  0   422 sshd          tcp    22 192.168.1.3
  0   266 inetd         tcp    21 192.168.1.3
```

4.Network Connections, cont.

- **Conclusions:**
  - Not very easy to relate network connections to processes on the system or list the top of connections
  - Net category has a lot of scripts which can easily help like: *tcpsnoop*, *tcptop* and *connections*

5.Disk Utilization

- Disk utilisation can be monitored using *iostat* – but to co-relate the utilisation with a process is a hard mission
- There are tools to check CPU usage by process but there are no tools to check disk I/O by process
- The old good friend: *iostat -xnmp*
- I/O type: reading *iostat* data a SysAdmin can describe if the I/O is sequential or random

5.Disk Utilization, cont.

- It is important to know what type of I/O there is: sequential or random
- How can you list what processes are generating I/O, or list disk events or how much a process is using the disk (size of the disk event or the service time of the disk events) ?
- Easily use the following DTT scripts: *iotop*, *iosnoop* from DTT root directory

5.Disk Utilization, cont.

- One Liner says:

```
sparvu@earth>dtrace -n 'io:::start{printf("%d %s %d",pid,execname,args[0]->b_bcount);}'
dtrace: description 'io:::start' matched 6 probes
^C
CPU     ID                    FUNCTION:NAME
  0     71              bdev_strategy:start 5637 bart 8192
  0     71              bdev_strategy:start 5637 bart 4096
  0     71              bdev_strategy:start 5637 bart 3072
  0     71              bdev_strategy:start 5637 bart 8192
  0     71              bdev_strategy:start 5637 bart 8192
  0     71              bdev_strategy:start 5637 bart 12288
  0     71              bdev_strategy:start 5637 bart 4096
  0     71              bdev_strategy:start 5637 bart 4096
  0     71              bdev_strategy:start 5637 bart 20480
  0     71              bdev_strategy:start 5637 bart 12288
  0     71              bdev_strategy:start 5637 bart 4096
  0     71              bdev_strategy:start 5640 find 3072
  0     71              bdev_strategy:start 5640 find 1024
  0     71              bdev_strategy:start 5640 find 1024
  0     71              bdev_strategy:start 5640 find 1024
  0     71              bdev_strategy:start 5640 find 1024
  0     71              bdev_strategy:start 5637 bart 32768
  0     71              bdev_strategy:start 5640 find 2048
  0     71              bdev_strategy:start 5637 bart 8192
  0     71              bdev_strategy:start 5640 find 2048
  0     71              bdev_strategy:start 5637 bart 24576
  0     71              bdev_strategy:start 5637 bart 3072
  1     71              bdev_strategy:start 5640 find 1024
```

5.Disk Utilization, cont.

- Run *iotop*:

```
2006 Jun  4 14:40:33,  load: 0.27,  disk_r:  10416 KB,  disk_w:      8 KB

  UID    PID   PPID CMD              DEVICE  MAJ MIN D   %I/O
  100   1968      1 gconfd-2         cmdk0   102   7 W      0
    0    121      1 nscd             cmdk0   102   0 R      1
  100   5568      1 gnome-panel-scre cmdk0   102   1 R      1
  100   1968      1 gconfd-2         cmdk0   102   7 R      1
    0    392    386 Xorg             cmdk0   102   0 R      2
  100   5555   4816 bart             cmdk0   102   7 R     16
  100   5568      1 gnome-panel-scre cmdk0   102   7 R     21
  100   5568      1 gnome-panel-scre cmdk0   102   0 R     54
```

5.Disk Utilization, cont.

- Run now *iosnoop*:

```
UID   PID D    BLOCK   SIZE       COMM PATHNAME
100  5603 R  3475216   8192       bart /opt/openoffice.org2.0/program/libres680si.so
100  5603 R  3475232   8192       bart /opt/openoffice.org2.0/program/libres680si.so
100  5603 R  3475248  16384       bart /opt/openoffice.org2.0/program/libres680si.so
100  5603 R  3462668   2048       bart /opt/openoffice.org2.0/program/libres680si.so
100  5603 R 56037520   8192       bart <none>
100  5603 R 56038128   8192       bart /opt/openoffice.org2.0/program/libsb680si.so
100  5603 R 56038168   8192       bart /opt/openoffice.org2.0/program/libsb680si.so
100  5603 R 56038192   4096       bart /opt/openoffice.org2.0/program/libsb680si.so
100  5603 R 56038272   8192       bart /opt/openoffice.org2.0/program/libsb680si.so
100  5603 R 56038296   4096       bart /opt/openoffice.org2.0/program/libsb680si.so
100  5603 R 56038336  20480       bart /opt/openoffice.org2.0/program/libsb680si.so
100  5603 R 56038392   8192       bart /opt/openoffice.org2.0/program/libsb680si.so
100  5603 R 56038416  16384       bart /opt/openoffice.org2.0/program/libsb680si.so
100  5603 R 56038528  45056       bart /opt/openoffice.org2.0/program/libsb680si.so
100  5603 R 56038688  36864       bart /opt/openoffice.org2.0/program/libsb680si.so
100  5603 R 56038792  53248       bart /opt/openoffice.org2.0/program/libsb680si.so
100  5603 R 56038952   4096       bart /opt/openoffice.org2.0/program/libsb680si.so
100  5603 R 56038968   4096       bart /opt/openoffice.org2.0/program/libsb680si.so
100  5603 R 56038984   4096       bart /opt/openoffice.org2.0/program/libsb680si.so
100  5603 R 56039040  57344       bart /opt/openoffice.org2.0/program/libsb680si.so
100  5603 R 56039152   4096       bart /opt/openoffice.org2.0/program/libsb680si.so
100  5603 R 56039224   4096       bart /opt/openoffice.org2.0/program/libsb680si.so
100  5603 R 56039288   8192       bart /opt/openoffice.org2.0/program/libsb680si.so
```

5.Disk Utilization, cont.

- How much the process reads...use *bitesize.d*:

```
  PID  CMD
 5602  find /opt\0

       value  ------------- Distribution ------------- count
         512 |                                         0
        1024 |@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@ 21
        2048 |@@@                                      2
        4096 |                                         0
        8192 |@@@@@                                    3
       16384 |                                         0

 5611  find /\0

       value  ------------- Distribution ------------- count
         512 |                                         0
        1024 |@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@ 886
        2048 |@@                                       71
        4096 |@                                        21
        8192 |@@@@@@@                                  208
       16384 |                                         0

 5603  bart create -I\0

       value  ------------- Distribution ------------- count
         512 |                                         0
        1024 |@@@@                                     127
        2048 |@@                                       64
        4096 |@@                                       70
        8192 |@@@@@@@@@@                               334
       16384 |@@                                       83
       32768 |@@@@@@@@@@@@@@@@@@@@@@                   669
       65536 |                                         0
```

5.Disk Utilization, cont.

- Look for seek distance of the disk events. Run *seeksize.d* to understand if the I/O is sequential or not:

```
5603  bart create -I\0

           value  ------------- Distribution ------------- count
              -1 |                                         0
               0 |@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@ 914
               1 |                                         0
               2 |                                         18
               4 |@                                        26
               8 |@                                        26
              16 |@                                        26
              32 |                                         16
              64 |@                                        31
             128 |@                                        33
             256 |@                                        25
             512 |@                                        19
            1024 |                                         15
            2048 |@                                        28
            4096 |@                                        51
            8192 |@@@@@                                    170
           16384 |                                         6
           32768 |                                         0
           65536 |                                         1
          131072 |                                         1
          262144 |                                         0
          524288 |                                         0
         1048576 |                                         2
         2097152 |                                         3
         4194304 |                                         2
         8388608 |                                         2
        16777216 |@                                        21
        33554432 |@                                        26
        67108864 |                                         0
```

5.Disk Utilization, cont.

- Other important DTT utilities used to measure and analyse disk I/O events
- *rwsnoop*: snoops the read/write operations
- *rwtop*: used to display the top read/write operations by process id
- *opensnoop*: used to snoop what files are being open and by who. Very easy to discover what processes are opening what files

5.Disk Utilization, cont.

- *rwtop* and *opensnoop*:

```
2006 Jun  4 15:38:03,  load: 0.33,  app_r:   2883 KB,  app_w:   2842 KB

  UID    PID   PPID CMD              D            BYTES
  100   1954   1952 gnome-session    R               16
  100   2194   2193 BitchX-1.1-final R               59
  100   5411   5405 firefox-bin      R               63
  100   5411   5405 firefox-bin      W               63
  100   5650   4816 gimp-2.0         R               64
  100   5454   5443 soffice.bin      W               80
  100   2101      1 nautilus         W              216
  100   1982      1 xscreensaver     W              248
  100   2125      1 clock-applet     W              320
  100   5454   5443 soffice.bin      R              320
  100   2129      1 gnome-netstatus- W              416
  100   2099      1 gnome-panel      R              552
  100   2101      1 nautilus         R              640
  100   2194   2193 BitchX-1.1-final W              681
    0      1      0 init             W              824
  100   1968      1 gconfd-2         R              832
  100   2099      1 gnome-panel      W              920
```

```
UID    PID COMM         FD PATH
  0    252 utmpd         5 /var/adm/utmpx
  0    252 utmpd         6 /var/adm/utmpx
  0    252 utmpd         7 /proc/1840/psinfo
  0    252 utmpd         7 /proc/2150/psinfo
  0    252 utmpd         7 /proc/2150/psinfo
  0    252 utmpd         7 /proc/2150/psinfo
  0    252 utmpd         7 /proc/2150/psinfo
  0    252 utmpd         7 /proc/2150/psinfo
  0    252 utmpd         7 /proc/2150/psinfo
  0    252 utmpd         7 /proc/2150/psinfo
  0    252 utmpd         7 /proc/2150/psinfo
  0    252 utmpd         7 /proc/2150/psinfo
100   5685 find         -1 /var/ld/ld.config
100   5685 find          3 /lib/libsec.so.1
100   5685 find          3 /lib/libc.so.1
100   5685 find          3 /lib/libavl.so.1
100   1968 gconfd-2     44 /export/home/sparvu/.gconfd/saved_state.tmp
100   5687 bart         -1 /var/ld/ld.config
100   5687 bart          3 /lib/libsec.so.1
100   5687 bart          3 /lib/libmd5.so.1
100   5687 bart          3 /lib/libc.so.1
100   5687 bart          3 /lib/libavl.so.1
100   5686 find         -1 /var/ld/ld.config
100   5686 find          3 /lib/libsec.so.1
100   5686 find          3 /lib/libc.so.1
100   5686 find          3 /lib/libavl.so.1
100   5688 bart          3 /etc/default/init
100   5688 bart          3 /usr/share/lib/zoneinfo/Europe/Helsinki
100   5688 sort         -1 /var/ld/ld.config
100   5688 sort          3 /lib/libc.so.1
100   5688 sort          3 /proc/self/auxv
100   5688 sort         -1 /var/ld/64/ld.config
100   5688 sort          3 /lib/64/libc.so.1
100   5688 sort          3 /dev/null
100   5687 bart          3 /opt/sfw/lib/firefox/LICENSE
100   5687 bart          3 /opt/sfw/lib/firefox/README.txt
100   5687 bart          3 /opt/sfw/lib/firefox/browserconfig.properties
```

DTrace – Life After Solaris 10 3/05

Contributed by Jon Haslam
http://blogs.sun.com/jonh

Multiple aggregation printa()
Release: 08/07-30

- multiple aggregations in single printa()
- aggregations must have same type signature
- output is effectively joined by key
- 0 printed when no value present for a key
- default behavior is to sort by first aggregation value (ties broken by key order)

Multiple aggregation printa()

Multiple aggregation printa()

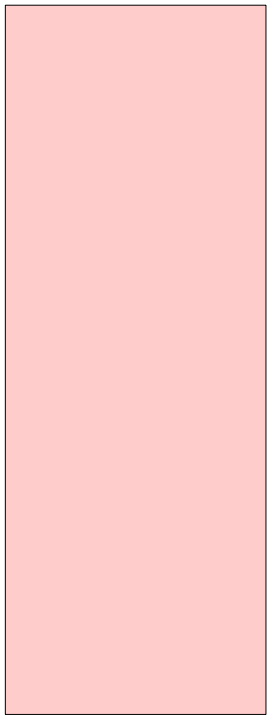
Aggregation sorting options
Release: 08/07-30

- aggregations sorted by value by default
- options allow change of behaviour
  - aggsortkey - sort by key order, ties broken by value
  - aggsortrev - reverse sort
  - aggsortpos - position of the aggregation to use as sort primary sort key with mutiple aggs
  - aggsortkeypos - position of key to use as primary sort key when with multiple aggs
- Use the above in combination

Aggregation sorting options

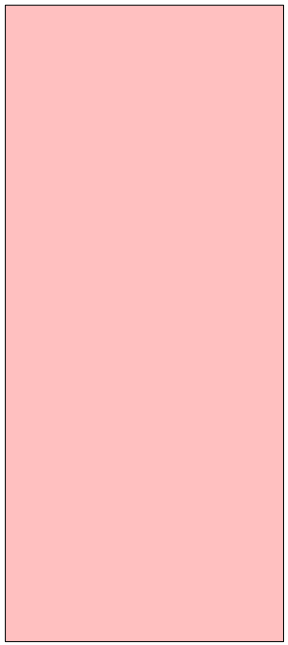
```
/* aggsort.d */
syscall::read:entry
{
        @avg[execname, pid] = avg(arg2);
        @max[execname, pid] = max(arg2);
        @min[execname, pid] = min(arg2);
        @cnt[execname, pid] = count();
}

END
{
        printf("%20s %10s %10s %10s %10s %10s\n", "EXECNAME", "PID",
"COUNT", "MIN", "MAX", "AVG");
        printa("%20s %10d %@10d %@10d %@10d %@10d\n", @cnt, @min,
@max, @avg);
}
```

Aggregation sorting options



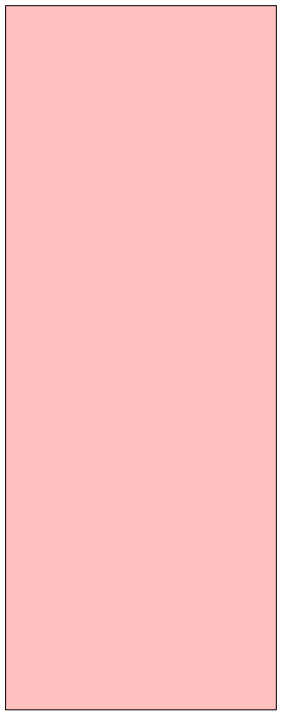
Sort by value of first aggregation (default)

aggregation sorting options



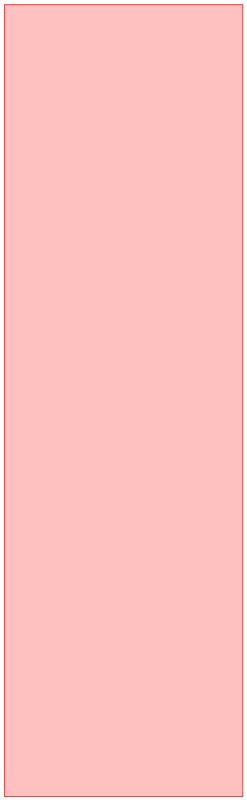
Reverse sort using value of first aggregation

Aggregation sorting options



Reverse sort by key in second position

Aggregation sorting options



Reverse sorted by value of third aggregation

(u)mod/(u)func/(u)sym
Release: 08/07-23

- Profiling often requires post-processing when using %a/%A to print arg0/arg1 symbolically
- Samples in format [module]'[func]+[offset]
- Want to first get high level view and then drill down
- (u)mod(%pc) - module name
- (u)func(%pc) - function name
- (u)sym(%pc) - symbol name

(u)mod/(u)func/(u)sym

Example uses prototype cpc provider to show TLB misses on a global basis broken down by module.

(u)mod/(u)func/(u)sym

ucaller variable
Release: 08/07-23

String parsing subroutines Release: 01/06-15

- The usual suspects
  - strtok() - tokenise a string
  - strstr() - find first occurrence of str2 in str1
  - strchr() - find first occurrence of char 'c' in str
  - strrchr() - find last occurrence of char 'c' in str
  - substr() - return substr of str starting at pos p for length l
  - index() - return position of first char 'c' in str
  - rindex() - return position of last char 'c' in str

(Don't forget about strjoin() and strlen()!)

String example

```
pid$target:libc:putenv:entry
{
        this->str = copyinstr(arg0);
        this->var = strtok(this->str, "=");
        this->val = substr(this->str, index(this->str, "=") + 1);
        printf("env[%s] = %s\n", this->var, this->val);
}

# dtrace  -s ./putenv.d -c "date -u"
dtrace: script './putenv.d' matched 1 probe
Mon Sep 17 18:48:37 GMT 2007
CPU     ID                    FUNCTION:NAME
  0  74554                     putenv:entry env[TZ] = GMT0
```

The fds[] variable
Release: 01/06-16

- array of fileinfo_t's indexed by integer (fd)
- inlines expanded to accommodate associative arrays for this.
- Definition in /usr/lib/dtrace/io.d
- fileinfo_t gets new fi_oflags member

The fds[] variable

```
#pragma D option quiet

syscall::write:entry
/fds[arg0].fi_oflags & O_APPEND/
{
        printf("%s appending file %s at offset %d\n",
            execname, fds[arg0].fi_pathname, fds[0].fi_offset);
}

# ./fds.d
ksh appending file /.sh_history at offset 349345
ksh appending file /.sh_history at offset 349378
```

dtrace -v
Release: 01/06-11

- '-v' option added to dtrace(1M) to display probe arguments and stability attributes
- use in combination with '-l' or '-e'
- use with mdb(1) for examining kernel type info
- use docs for D type info (or /usr/lib/dtrace)

dtrace -v

inet_ntoa*()/inet_ntop() (-66)

- inet_ntoa() - converts IPv4 address into human readable string
- inet_notp() - convert both IPv4 and IPv6 addresses into human readable string
- inet_ntoa6() - IPv6 version of inet_ntoa()

inet_ntoa*()/inet_ntop()

```
fbt::ip_tcp_input:entry
{
        @[inet_ntoa(&args[1]->ipha_src), inet_ntoa(&args[1]->ipha_dst)] = count();
}
END
{
        printa("%30s -> %-30s %@d\n", @);
}

# ./inet_ntoa.d
^C
                212.58.227.137 -> 129.156.38.34        3
                 212.58.226.20 -> 129.156.38.34       59
                  212.58.226.8 -> 129.156.38.34       99

# dig -x 212.58.226.8 +short
newslb14.thdo.bbc.co.uk.
```

Misc bits

- ustackdepth variable (11/06-20)
- ucaller variable (08/07-23)
- %k printf/printa modifier for printing stacks (08/07-23)
- hton[s,l,ll], ntoh[s,l,ll], htohll (/50)
- Test suite integration (08/07-48)
- DTrace support in Zones (08/07-37)
- lockstat/plockstat -x option (01/06-9)
- DTrace JNI binding (08/07-35)

Available Providers

- fsinfo (u2/38) - Filesystem information provider
- sysevent (/58) - System event channel provider
- Xserver (/??) - Xorg
- python (/65) - John Levon's python beauty
- iscsi (/69) -
- xpv (/75) - x86 paravirtualized provider