



DTRACE

Milan Juřík
Revenue Product Engineer
Sun Microsystems

Agenda

- What is dtrace?
- Why should I care?
- How to use it
 - > Some providers
- Q&A
- URLs

What is Dtrace?

- Dynamic tracing framework
- Observability tool
- Combination of kernel support and userspace utility
- Includes tracepoints to binary code dynamically and allows to gain data in these tracepoints

Why Should I Care?

- Allow to instrument everything in system
- Is used by administrators and developers on live production systems to examine the behavior of both user programs and of the operating system itself
- Can explore your system to understand how it works, track down performance problems across many layers of software, or locate the cause of abnormal behavior

How to Use it

```
dtrace [-32 | -64] [-aACeFGHhIqSvVwZ] [-b bufsz] [-c cmd]
  [-D name [=value]] [-l path] [-L path] [-o output]
  [-s script] [-U name] [-x arg [=val]]
  [-X a | c | s | t] [-p pid]
  [-P provider [[predicate] action]]
  [-m [provider:] module [[predicate] action]]
  [-f [[provider:] module:] function [[predicate] action]]
  [-n [[[provider:] module:] function:] name [[predicate] action]]
  [-i probe-id [[predicate] action]]
```

- man dtrace(1M)

How to Use it

- D language
 - > C/AWK syntax
 - > action based
 - > data aggregations
 - > ...
- Probes
 - > defines point of instrumentation
 - > dtrace -l

How to Use it - Probes

provider:module:function:name /predicate/ {action}

- Provider – which one should take care of it
- Module – which module (specific location)
- Function – which function we care about
- Name
 - > entry – fires when the function is entered
 - > return – fires when the function is ended
 - > ...
- Predicate – predicate if it should be done
- Action – what should be done

How To Use It – syscall provider

- What is app doing?
- `dtrace -n 'syscall:::entry /execname == "firefox-bin"/ { @syscalls[probefunc] = count();}'`
- Ctrl+C

<code>read</code>	<code>76</code>
<code>write</code>	<code>76</code>
<code>lwp_park</code>	<code>96</code>
<code>ioctl</code>	<code>220</code>
<code>pollsys</code>	<code>221</code>

How to Use it – syscall Provider

- `dtrace -n 'syscall::ioctl:entry
/execname == "firefox-bin"/
{ @fildes[arg0] = count();}'`

- Ctrl+C

4

4 7 4

How to Use it – syscall Provider

- `dtrace -n 'syscall::ioctl:return
/execname == "firefox-bin"/
{ printf("%d", arg1);}'`

CPU	ID	FUNCTION:NAME
0	799	ioctl:return 0

How to Use it – syscall Provider

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

How to Use it – Short Break

- `dtrace -n 'syscall::open64:entry /execname == "firefox-bin"/ { @files[copyinstr(arg0)] = count();}'`
- `dtrace -n 'syscall::open64:entry /execname == "opera"/ { @files[copyinstr(arg0)] = count();}'`

How to Use it – More Programming

```
dtrace:::BEGIN { i = 10;}
```

```
profile:::tick-1sec /i > 0/ { trace(i--); }
```

```
profile:::tick-1sec /i == 0/ {  
    trace("blastoff!");  
    exit(0);  
}
```

```
dtrace -s countdown.d
```

How to Use it – io Provider

```
#pragma D option quiet
```

```
io::start{  
    @[args[1]->dev_statname, execname, pid] = sum(args[0]-  
>b_bcount);  
}  
END{  
    printf("%10s %20s %10s %15s\n", "DEVICE", "APP",  
"PID", "BYTES");  
    printa("%10s %20s %10d %15@d\n", @);  
}
```

How to Use it – sched Provider

```
sched:::on-cpu
```

```
{  
    @counter[execname] = count();  
}
```

```
profile:::tick-20sec
```

```
{  
    exit(0);  
}
```

How to Use it – pid Provider

- Tracing per process
- Allows to trace every instruction
- No need to recompile application – it just works
- No effect on non-probed processes

How to Use it – Fancy Output

- What happens when `gethostbyname()` is called?

Tools

- Dtracetoolkit
- Chime
- Instruments (Mac OS X)

Q&A

Questions & answers

URLs

- <http://www.sun.com/bigadmin/content/dtrace/>
- <http://www.opensolaris.org/os/community/dtrace/>
- <http://wikis.sun.com/display/DTrace/Documentation>
- http://www.brendangregg.com/DTrace/dtrace_oneliners.txt



DTRACE

Milan Jurík

milan.jurik@sun.com