

DTrace + Ruby

LRUG

8th October 2012

Andhapp **Dutta**

@andhapp

Ruby Freelancer

I've met Matz



DTrace

2005

Sun OS

Mac OSX
Pre-installed

FreeBSD
~~Pre-installed~~

Linux
~~Pre-installed~~

sudo

Static and Dynamic Instrumentation Framework



Static and Dynamic Instrumentation Framework

Static and Dynamic Instrumentation Framework

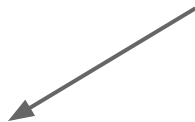
Static and Dynamic Instrumentation Framework

Example

```
sudo dtrace  
-n 'syscall::read:entry  
/pid == 12345/  
{ @num[execname] = count(); }'
```


Example

Consumer



sudo **dtrace**

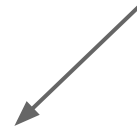
-n 'syscall::read:entry

/pid == 12345/

{ @num[execname] = count(); }'

Example

Probe



sudo dtrace

-n '**syscall::read:entry**

/pid == 12345/

{ @num[execname] = count(); }'

Example

sudo dtrace

-n 'syscall::read:entry

/pid == 12345/

← **Clause**

{ @num[execname] = count(); }'

Example

sudo dtrace

-n 'syscall::read:entry

/pid == 12345/

{ @num[execname] = count(); }



Predicate

**Available Probes
varies across OS**

Use cases

Enough DTrace

We ♥ Ruby

Ruby trunk

function-entry

object-create

array-create

hash-create

and so on...

I want my own
PROBES

ruby-usdt

Activesupport:: Notifications

rails-dtrace

Use cases

Unstable

Future





Thanks

Questions?