The magic 💩

when your node app miserably crashes!

Who am I?

@lucamaraschi

- Architect @ IceMobile during the day
- *Node.js* and *SmartOS* Italian lover (during the night!)
- International speaker

Basically…

I inspect your dumps!

Disclaimer

I am NOT a Joyent employee

But sorry...

This magic only works on SmartOS!

WHY?

Let's check the node core source...

Once upon a time…

but first...

your app/service is going to crash!

DON'T even think it's BUG FREE!

3AM and PagerDuty

tells you

PRODUCTION DOWN!

What the hell is going on?

Usual suspects

- Pathological performance
- Wrong behavior
- Memory problems (leaks, excessive usage)

Where shall I start from?

If you say console.log()

Change job!

Each lap is expensive
Each lap is disruptive

Production problems are often hard to

Worse, restarts often make the original problem go away

Dtrace

YC
CHELSEA
HOTEL
node

Node.js probes

Restify includes dtrace support

as first class thing

Now you know

why you MUST use restify!

But also bunyan

has some interesting dtrace in...

Imagine…

to change the log level LIVE

state of the app does not change!

Another reason to use restify!

dtrace

or die!

don't complain to me...I told you!

Runtime debugging

on CPU or off CPU

On CPU

running js code or doing GC

classical approach is to profile

Oh yeah…dtrace

Have you already taken your DUMP?

gcore

only on Solaris like OSs

Don't be a bad boy...

--abort-on-uncaught-exception

Then I can put my hands into your dump!

mdb

MDB is the illumos modular debugger, used to debug the live kernel, kernel crash dumps, processes, and process core dumps

So...

Resources

- mdb and node.js
- nshttpsnoop
- wrk