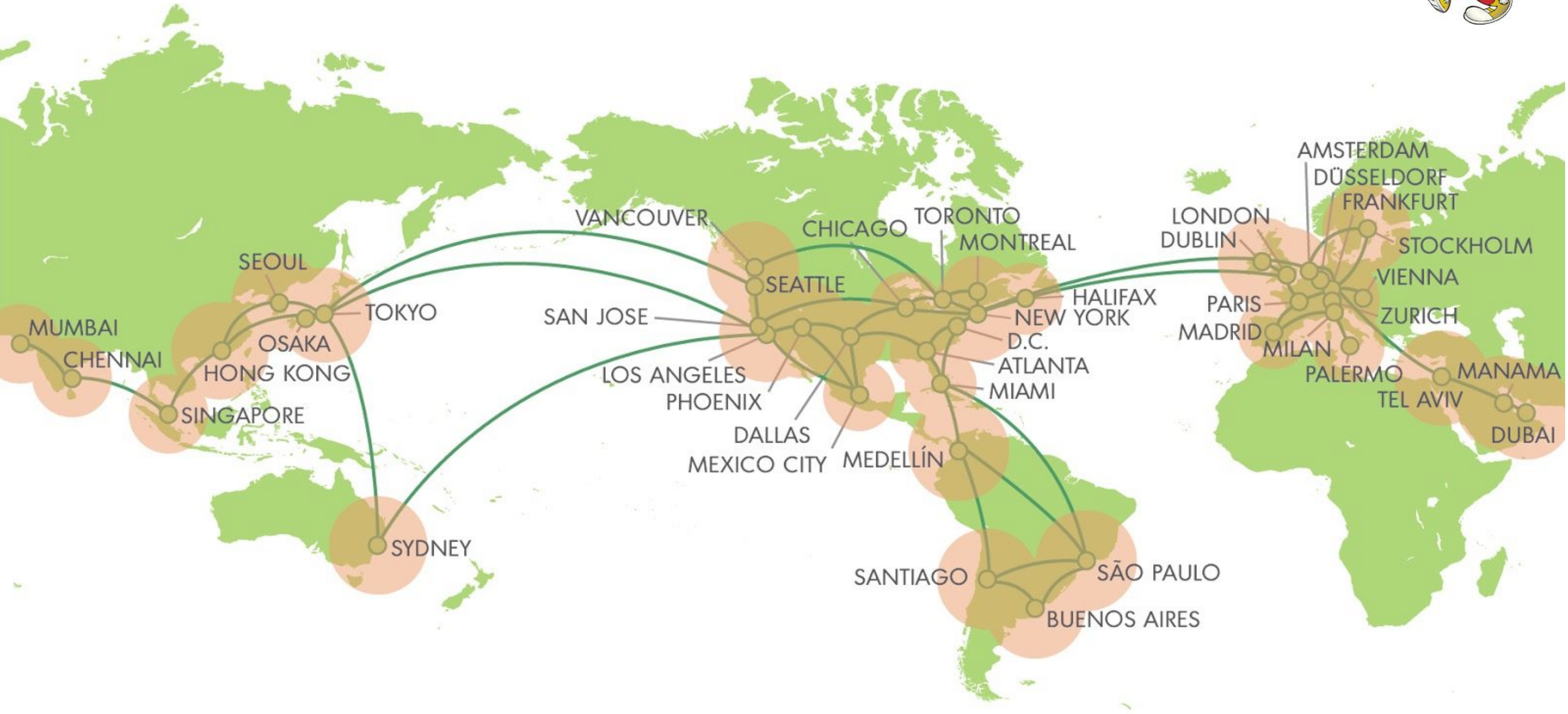
FreeBSD Operations at Limelight Networks



The densely architected Limelight CDN bypasses the public Internet via 80+ interconnected locations for truly unlimited global reach.

ASN 22822

Intro

Stay here to hear about

Scale out operations using FreeBSD

Limelighters at BSDCan 2015

Kevin Bowling - presenter

Sean Bruno (sbruno@freebsd.org)

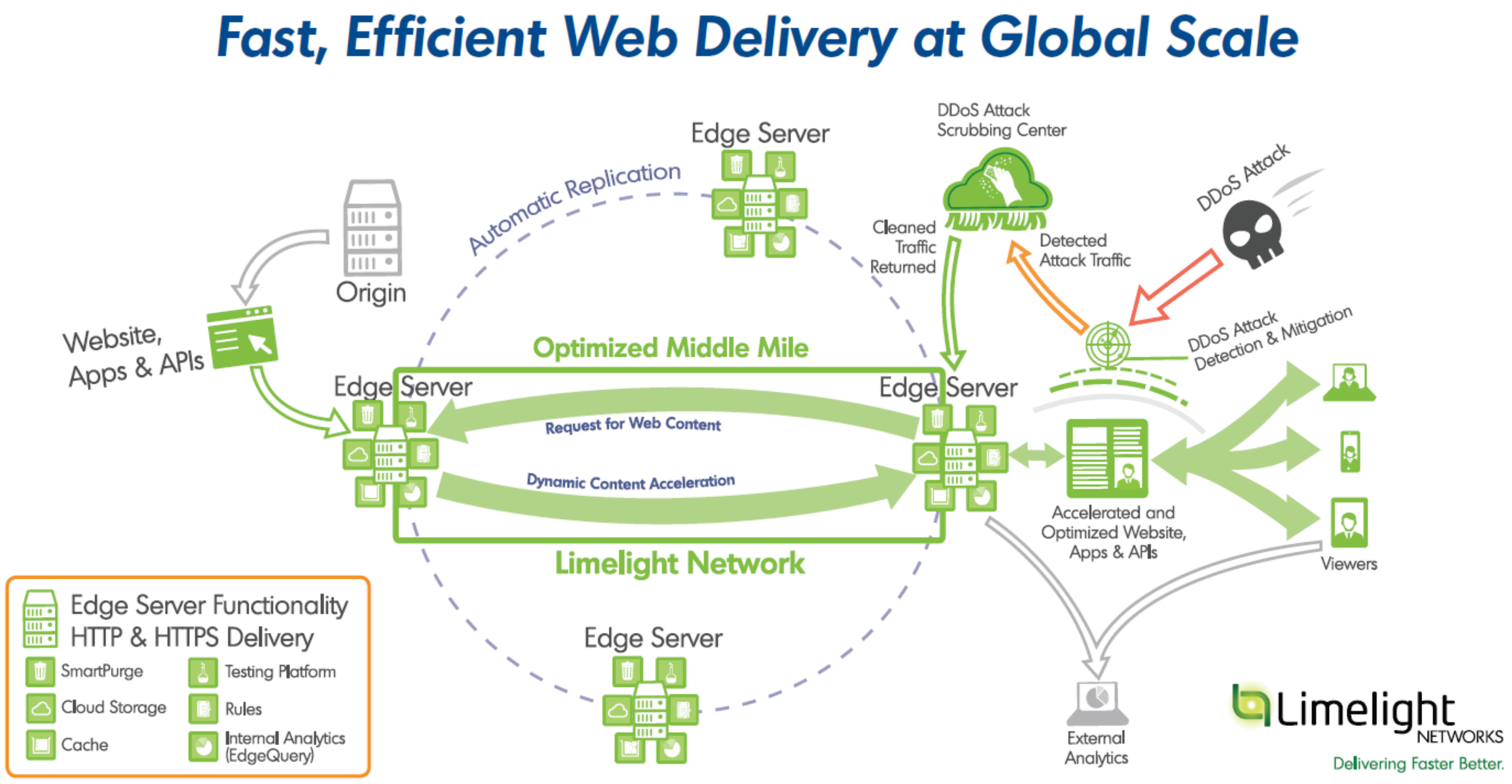
Hiren Panchasara (hiren@freebsd.org)

Jason Wolfe

Chris Christensen

Johannes Meixner (xmj@freebsd.org)

What does Limelight do?



POP Architecture

DWDM gear

Redundant Internet routers w/full routes

Large chassis switches or spine switching mesh

TOR switching

Lots of these:

Limelight NETWORKS

Let’s talk about Ops

There are plenty of folks talking about appliance, embedded, academic use. This talk was borne out of the desire to see more large “ops” installations talking about BSD.

Main difference.. systems are fluid - software and configuration are rolled out as standard operating procedure

Think: large web sites, API-centric companies, service providers

Workload almost exclusively consists of Internet facing services

My Entry Phase One: Analysis

My background: 10+ years professional Linux SysEng

"UNIX Aficionado" - but just a BSD observer - ran m0n0wall-> pfSense, dabbled with Net/Open/Free, AIX, IRIX, Solaris, etc

Start at LLNW - intrigued by BSD edge. *"How are so few BSD people doing so much?"*

Answer: BSD software and mindset. *"I need to get involved with THAT"*

Equal Answer: Observability trumps everything else

Ops: Monitoring

We use Zabbix and are generally happy with it

Getting it to run at scale took some doing, but it has been reliable

Key insight: use an API driven monitoring system.. monitoring should be configured by CM. Monitoring *must* be part of service entry into production. Monitoring *should* be part of testing/QA.

It's 2015, stop deploying nagios

ZABBIX

Limelight NETWORKS

Ops: Metrics

OpenTSDB

HBase clusterfsck but write-scalable, somewhat read scalable

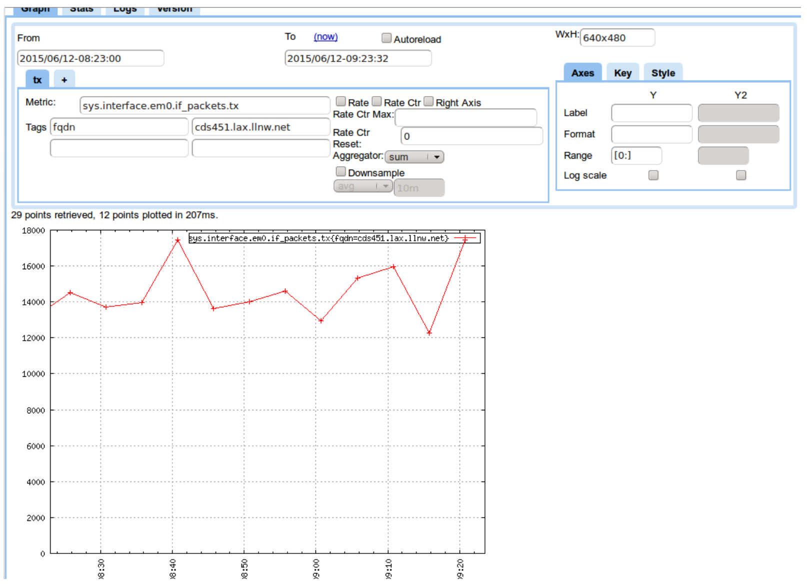
“So what you have a metrics dumping ground” Sean Chittenden @BSDCan - yeah more or less

I’m not so happy with this but it is sometimes better than nothing

Jut.io

Interesting data flow language startup, easy to aggregate data from multiple sources and APIs

Not quite metrics but mentioned here: Splunk, ELK

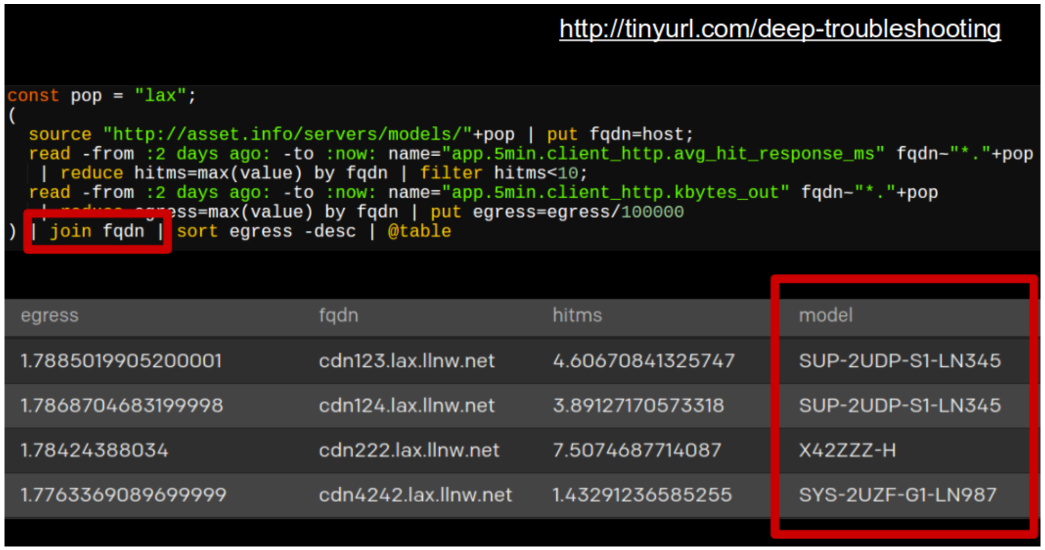


From 2015/06/12-08:23:00
To (now) 2015/06/12-09:23:32
Autoreload
WxH: 640x480

Metric: sys.interface.em0.if_packets.tx
Rate / Rate Ctr / Right Axis
Rate Ctr Max:
Tags fqdn cds451.lax.llnw.net
Rate Ctr Reset: 0
Aggregator: sum
Downsample avg 10m

	Y	Y2
Label		
Format		
Range	[0:]	
Log scale		

29 points retrieved, 12 points plotted in 207ms.



http://tinyurl.com/deep-troubleshooting

```
const pop = "lax";
(
  source "http://asset.info/servers/models/"+pop | put fqdn=host;
  read -from :2 days ago: -to :now: name="app.5min.client_http.avg_hit_response_ms" fqdn~"*."+pop
   | reduce hitms=max(value) by fqdn | filter hitms<10;
  read -from :2 days ago: -to :now: name="app.5min.client_http.kbytes_out" fqdn~"*."+pop
   | reduce egress=max(value) by fqdn | put egress=egress/100000
) | join fqdn | sort egress -desc | @table
```

egress	fqdn	hitms	model
1.7885019905200001	cdn123.lax.llnw.net	4.60670841325747	SUP-2UDP-S1-LN345
1.7868704683199998	cdn124.lax.llnw.net	3.89127170573318	SUP-2UDP-S1-LN345
1.78424388034	cdn222.lax.llnw.net	7.5074687714087	X42ZZZ-H
1.7763369089699999	cdn4242.lax.llnw.net	1.43291236585255	SYS-2UZF-G1-LN987

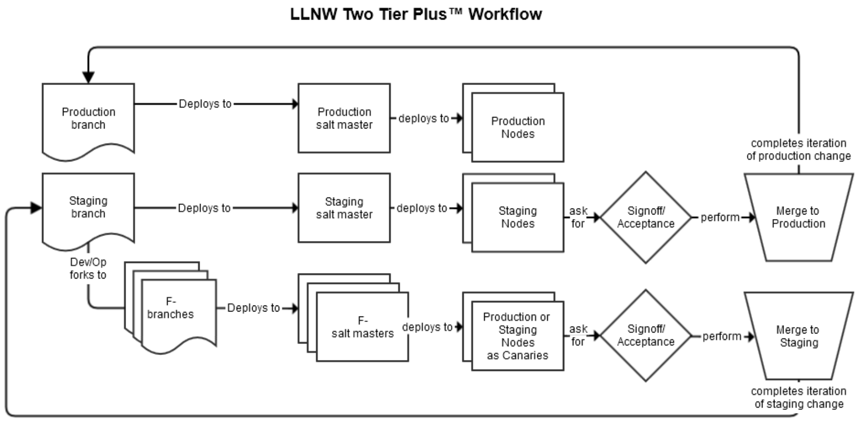
Ops: SaltStack

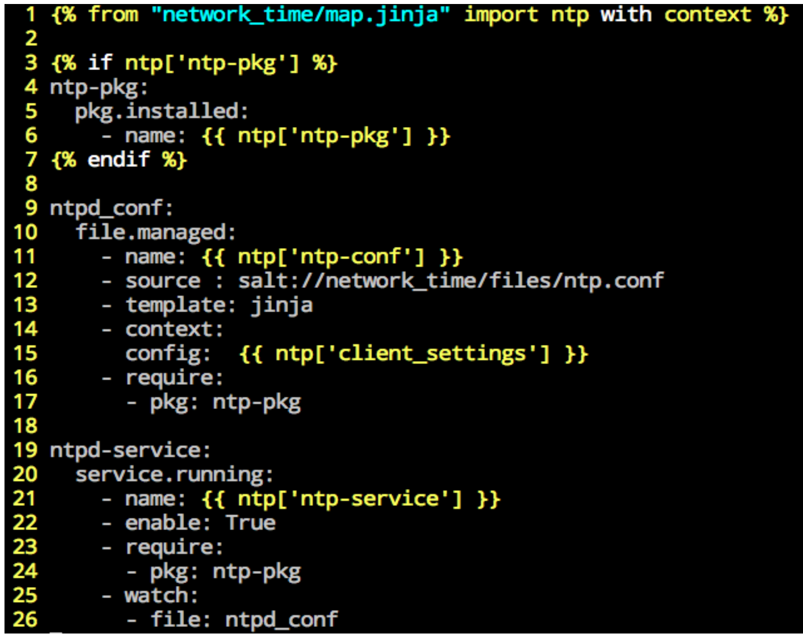
SaltStack is a Configuration Management tool built on an Orchestration bus. We think this is genius.

Limelight @ SaltConf15 https://www.youtube.com/watch?v=4lhVOhPJABQ

DTrace-ifying 2000 machines:

salt 'cds*' cmd.script salt://local/dsack.d | grep DSACK





```
{% from "network_time/map.jinja" import ntp with context %}

{% if ntp['ntp-pkg'] %}
ntp-pkg:
  pkg.installed:
    - name: {{ ntp['ntp-pkg'] }}
{% endif %}

ntpd_conf:
  file.managed:
    - name: {{ ntp['ntp-conf'] }}
    - source : salt://network_time/files/ntp.conf
    - template: jinja
    - context:
      config:   {{ ntp['client_settings'] }}
    - require:
      - pkg: ntp-pkg

ntpd-service:
  service.running:
    - name: {{ ntp['ntp-service'] }}
    - enable: True
    - require:
      - pkg: ntp-pkg
    - watch:
      - file: ntpd_conf
```

Simple state example

CM changes are a feedback loop

Changes to the CM system happen on the fly in containers

Ops: BSD RelEng Feedback Loop

We use git to maintain two branches of FreeBSD, which we call

llbsd-head - follows FreeBSD.org HEAD branch with LLNW patches

llbsd-stable - follows FreeBSD.org 10-STABLE branch with LLNW patches

buildotron - Jenkins jobs turn tags against these branches into built artifacts for deployment

Vagrant - offer developers and operators a production-like environment on their laptops

- helps greatly for new hires

Packer - boot ISO, add extra stuff, produce Vagrant Box

- we produce production Linux images with Packer.. much easier on FreeBSD because we can plug into build system

Configuration Management - extras for prod images and vagrant images

Limelight NETWORKS

My Entry Phase Two: Deliberate Use

Attract a src team

Upstream all the things

Use ports best practices

Make the system do what you want deliberately, not accidentally

Starting a src team

The more scaled out, the more dividends src influence pay

- FreeBSD 8 -> 10 while reducing custom patch stack
- Multiqueue em driver
- ipfw on inbound only
- PLMTUD implementation
- calloutng fixes
- TCP customization
- MFCs as needed

How do you do it?

Watch or offer on freebsd-jobs@freebsd.org

Recruit at conferences

Do cool stuff sensibly and people will come to you

Working **with** src

Develop against HEAD

MFC to -STABLE

Do internal RelEng

Deploy to prod

Monitor

Analyze

Change

Repeat

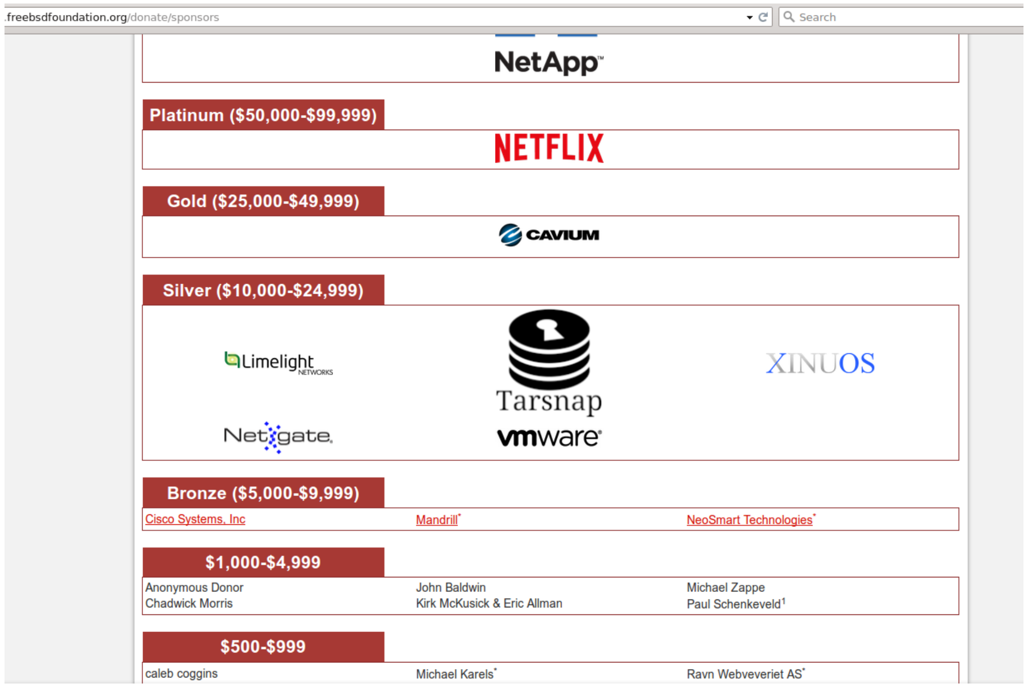
OODA loop or most simply a **feedback loop**

My Entry Phase Three: Now

- Identify and support key features and **community**
- Show company we are more **effective** and enlighten people that want to be the same
- Empower service owners and operators
  - Key technologies:
    - Base system building blocks
    - Poudreire + pkg
    - SaltStack
  - Elsewhere, perhaps?
    - ZFS
    - VIMAGE Jails (iocage)

FreeBSD Foundation

We are a proud sponsor of the FreeBSD Foundation and think they are doing an excellent job



NetApp

Platinum ($50,000-$99,999)

NETFLIX

Gold ($25,000-$49,999)

CAVIUM

Silver ($10,000-$24,999)

Limelight NETWORKS
Tarsnap
XINUOS
Netgate
vmware

Bronze ($5,000-$9,999)

Cisco Systems, Inc	Mandrill*	NeoSmart Technologies*

$1,000-$4,999

Anonymous Donor	John Baldwin	Michael Zappe
Chadwick Morris	Kirk McKusick & Eric Allman	Paul Schenkeveld[1]

$500-$999

caleb coggins	Michael Karels*	Ravn Webveveriet AS*

https://www.freebsdfoundation.org/donate/sponsors

Fin.

Keep Calm

Deploy *BSD To Prod

Thank you!

Backup: Intel em Multiqueue

Intel's em driver on FreeBSD and Linux only uses one tx and rx queue.

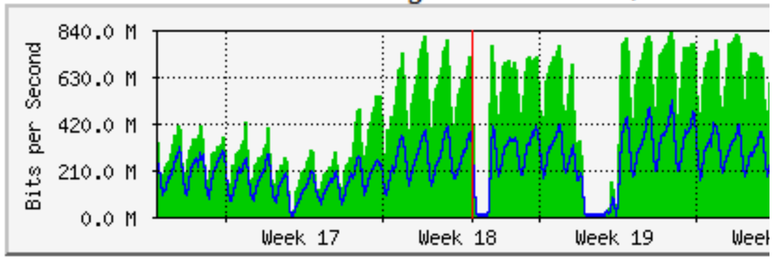
Sean found through some digging that the hardware is actually capable of 2 tx and rx queues and patched the driver to use them.

On a lagg cds box, this distributes network processing path over 4 cores.

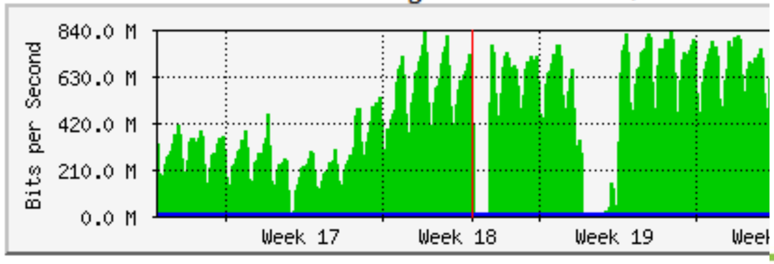
Previously ~1.2gbps with quality problems to ~1.9gbps with good quality (em FIFO seems too small so rare drops but no way to fix that in software)

We have 1000s of these boxes in production

cds345.lax-e0 swf1.lax6 GigabitEthernet1/33



cds345.lax-e1 swf1.lax6 GigabitEthernet1/34



Backup: pmcstat & dtrace

Increasing performance and efficiency requires understanding both the application and OS (kernel, base libs)

I am a poor stand in for Brendan Gregg, but his books and talks are a fantastic resource for companies developing or deploying any software