

From Microservices to Monoliths

BSDCan 2017 v0.19

Dave Cottlehuber

@dch



<https://home.apache.org/~dch>



iwantmyname

This Talk

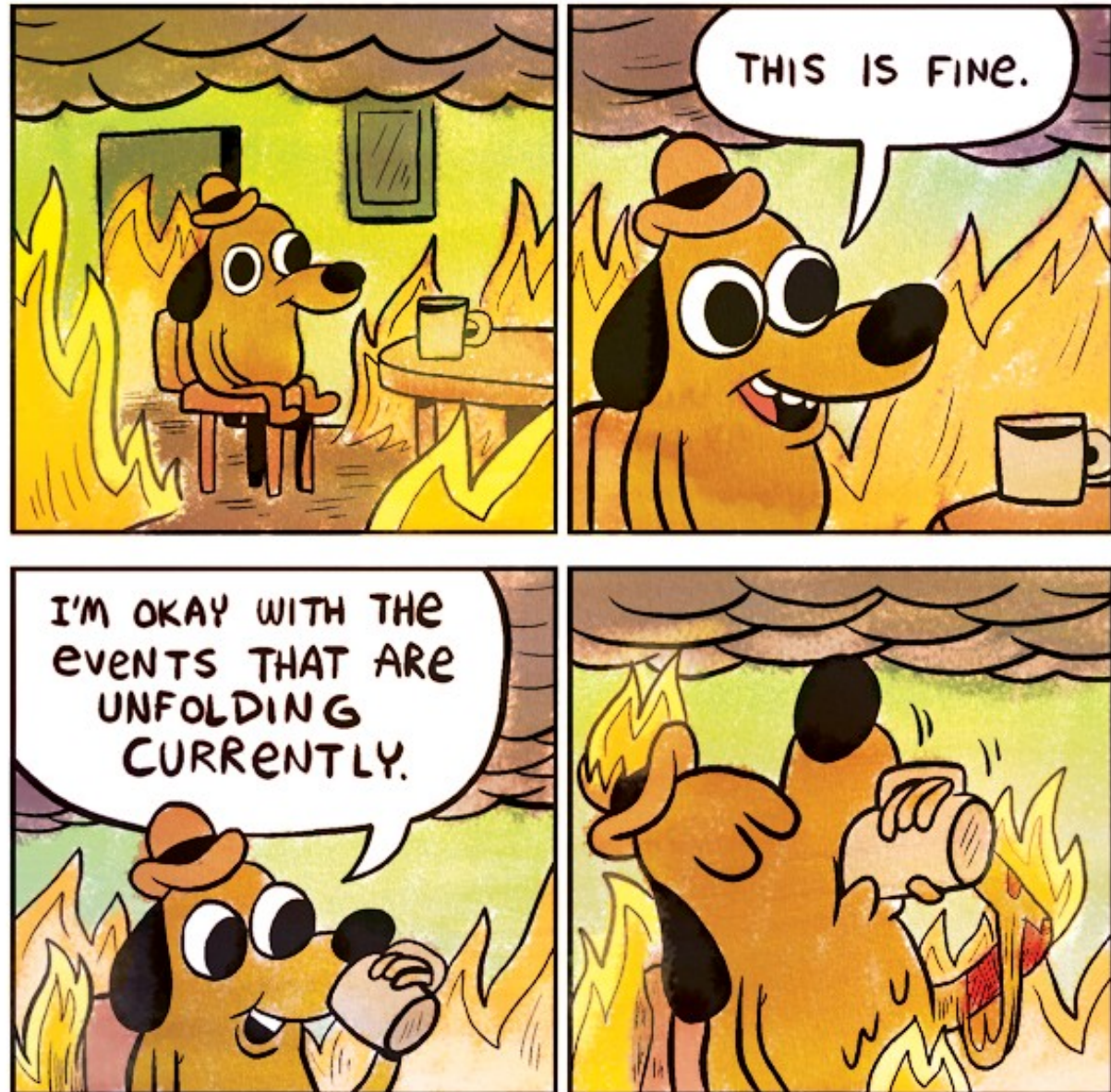
- about
- why we migrated & to what
- maintaining & building our source
- architecture
- moving the back end
- interlude 1
- the batbelt – vault & ansible
- cluster/jail internals
- interlude 2
- metrics & monitoring
- interlude 3
- thoughts Q&A

iwantmyname

- Hand-Crafted Artisanal Domain Resellers
 - Domains for Real People
 - Ethical business
 - Simple interface
 - NZ + global distributed team
- 
- A photograph of a wooden tray filled with small, dark, rectangular pieces of wood, possibly charcoal or kindling. A metal tool, possibly a trowel or a similar implement, is resting on top of the tray. The tray is divided into several compartments by wooden dividers.
- Perl Catalyst front end
 - Erlang backend: RabbitMQ, Apache CouchDB, Search App
 - Kyoto Tycoon Session Store
 - 20 Debian Unstable VMs

Why Migrate?

- Microservices-induced Latency
- VM-induced network instability
- Debian-induced base package variability –
OpenSSL, Net::HTTP
- TCP ephemeral port exhaustion
- Upgrade Hell – no turning back



Choose Wisely



The Holy Grail

- reliable std & custom packages
- easy rollbacks for Dbs & upgrades
- ansible for lower non-sysadmin learning curve
- robust transparent failover for app & backend
- spiped tunnels for resilience
- stable API for containers – no fuss, no latency
- 3 physical machines (A + B cluster nodes, C sysadmin/backup)
- a few VMs during migration for eventual consolidation



Poudriere – Floating Ports

- custom git fork of github:freebsd-ports
- custom ports + private packages
- best of both worlds
- bump every 2-3 months when it suits
- pull in security fixes only when needed
- no rush for upstream committers ;-)
- guaranteed consistency across dev/prod

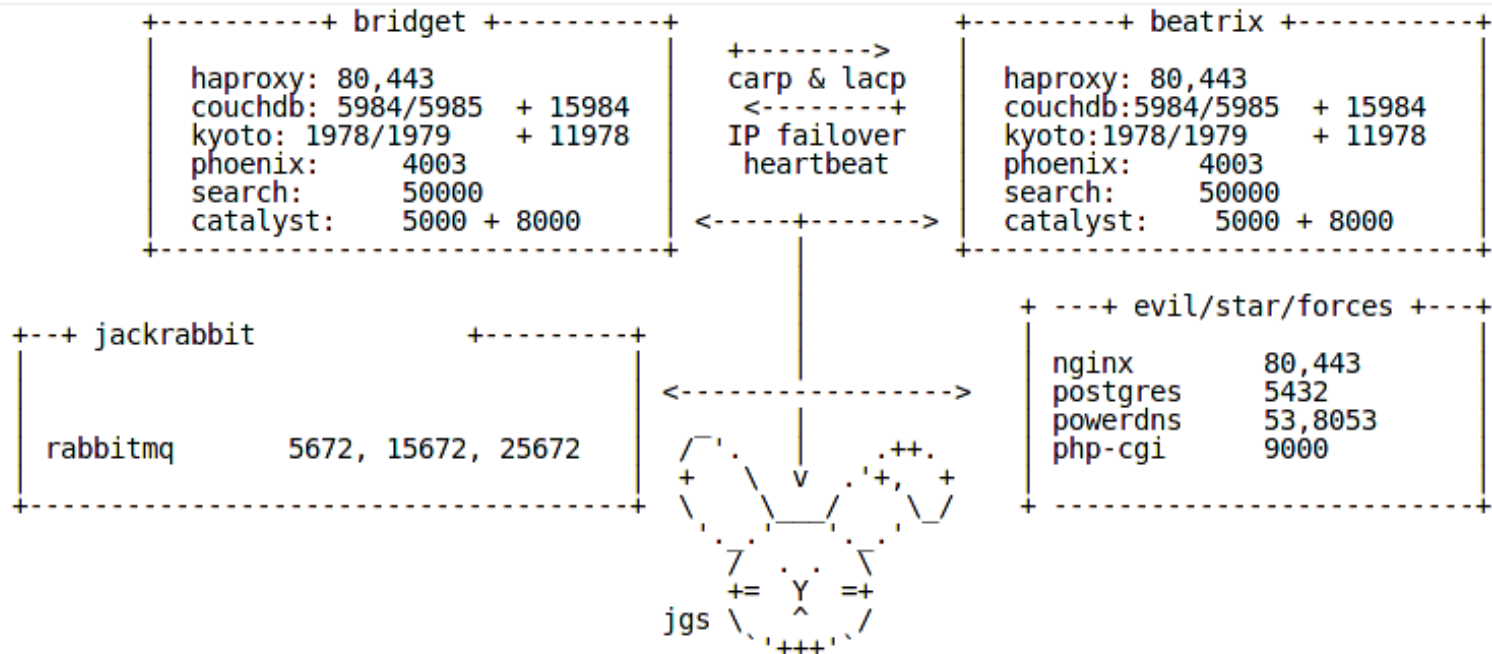
The screenshot shows a GitHub repository page for 'ideegeo / ports', which is a fork of 'freebsd/freebsd-ports'. The repository is on the 'master' branch. The commit history is displayed, with several commits circled in red. The circled commits are:

- net/haproxy: import from upstream** (buildbot committed 10 days ago)
- net-mgmt/riemann: update 0.2.12 to 0.2.13** (buildbot committed 10 days ago)
- databases/kyototycoon: import patch that finally** (buildbot committed 17 days ago)

Other visible commits include:

- security/libressl: import upstream 2.5.4 security** (buildbot committed 15 days ago)
- net-mgmt/riemann-c-client: pull upstream patches** (buildbot committed 16 days ago)
- *IDs: bump from master to get kyototycoon changes** (buildbot committed 17 days ago)
- sysutils/ansible: add py-jmespath missing dependency** (buildbot committed 22 days ago)
- sysutils/ansible: update to 2.3.0.0** (buildbot committed 22 days ago)
- devel/jenkins: bump 2.47 to 2.61** (buildbot committed 23 days ago)
- sysutils/spined: update 1.5.0 -> 1.6.0 for unlimited** (buildbot committed 23 days ago)

Architecture

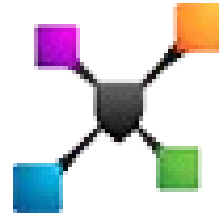


RabbitMQ, couchdb, kyototycoon, rsyslog, collectd, graphite all via spiped tunnels

+-----+ mrbrown (monitoring logging) +-----+	
h2o	443 pkg.....com
riemann	5555-5559
rsyslog	514
vault	8200
graylog	9000,9200,9300
jenkins	8180
couchdb	5986 *backup no views*
carbon	2003
h2o	80 acme/letsencrypt

Backwards First

Multiple nodes all behind



HAPROXY

All distributed, reliable, boring tech:

- Session: Kyoto Tycoon
- Store: Apache CouchDB – Erlang/OTP JSON doc DB
- Queue: RabbitMQ – Erlang/OTP

Interlude the First

- tested load through spiped tunnels based on 2x loading
- started DB migration to FreeBSD cluster
- on Debian systems we switched over spiped tunnels
- perfect until the last system
- roll back and reviewed logs
- tweaked FreeBSD network stack
- tried again → looking good → go to watch Rogue One
- no cell reception in the theatre ...

Interlude the First

> - [72288] sonewconn: pcb 0xffff80d68996cb0: Listen queue overflow: 16

> already in queue awaiting acceptance (64492 occurrences)

>

> 1. Am I using spiped incorrectly by having it handle many short-lived

> connections? There are usually about 100, sometimes up round 200,

> concurrent HTTP connections most of the time, spread across 6 main

> "client" servers that all connect to the same tunnel endpoint.

“... spiped has a **default limit of 100 connections**. You can increase this up to 500 via the -n option; if you need more than that, we'll need to fix the networking code to use poll instead of select (entirely doable; it just hasn't happened yet).”

– cperciva@

Behold, The Sacred Beast of SysAdmins: The Hardy Yak



“They Who Must Be Shaved”

SysAdmin Batbelt

- vault – secrets management
- ansible – config management
- rsyslog* – centralised log shipping
- graylog* – log aggregation & search
- collectd – host-based metrics agent
- riemann – centralised event correlation & realtime dashboard
- graphite & grafana – pretty pictures and trends
- poudriere & jenkins – custom ports and applications

Hashicorp Vault

- key value store specifically for storing secrets
- Shamir's Secret Sharing algorithm prevents theft
- github auth yields time-limited admin tokens
- mostly pushed out via ansible
- no secrets in git
- no redundancy just S3 backend
- future plans

Hashicorp Vault

```
$ export VAULT_ADDR=https://vault.foo.com:8200/  
$ vault auth -method=github token=<your_token>
```

Successfully authenticated! You are now logged in.
The token below is already saved in the session. You do not

need to "vault auth" again with the token.
token: 7271d370-aa45-0629-b3b4-e3a7f150974b
token_duration: 7775999
token_policies: [admins, default]

```
$ cat ~/.vault-token  
7271d370-aa45-0629-b3b4-e3a7f150974b
```

Blackadder

```
$ vault write secret/blackadder scarlet_pimpernel="we do not know"
```

Success! Data written to: secret/blackadder

```
$ vault write secret/blackadder scarlet_pimpernel="comte de frou frou"
```

Success! Data written to: secret/blackadder

```
$ vault read -format=json secret/blackadder
```

```
{  
  "request_id": "36592ad4-b8e0-8a72-e4a5-540bcc6d3a4d",  
  "lease_id": "",  
  "lease_duration": 7776000,  
  "renewable": false,  
  "data": {  
    "scarlet_pimpernel": "comte de frou frou"  
  },  
  "warnings": null  
}
```

```
$ vault read -format=yaml secret/blackadder
```

data:

```
  scarlet_pimpernel: comte de frou frou  
lease_duration: 7776000  
lease_id: ""  
renewable: false  
request_id: 901a598f-3b6b-e0a9-9fbd-eba625f2aa3f  
warnings: null
```

```
$ vault read -field=scarlet_pimpernel  
secret/blackadder
```

comte de frou frou

```
$ vault delete secret/blackadder
```

Success! Deleted 'secret/blackadder' if it existed.

Ansible

- 50+ roles: pf, zfs, spiped, iocell jail creation, patch & reboot
- simpler on-ramp for non-sysadmins
- oops: native jail support is *local only*

- proper jail & vault support via plugins
- thanks xmj@ for both moral support and fractal cells

Ansible

```
# jail definition
```

```
---
```

```
config:
```

```
  iocell:
```

```
    instances:
```

```
      iwmbase:
```

```
        ipv4: "{{ net.private.ip.iwmbase }}"
```

```
        packages: |
```

```
          lang/python2
```

```
          security/sudo
```

```
          devel/gmake
```

```
          devel/git
```

```
          devel/p5-App-cpanminus
```

```
          www/p5-LWP-Protocol-https
```

```
          dns/p5-Net-LibIDN
```

```
          textproc/p5-XML-SAX-Expat
```

```
        properties:
```

```
          allow_raw_sockets: 1
```

```
# jail app config
```

```
---
```

```
config:
```

```
  iwmbase:
```

```
    catalyst:
```

```
      primary_port: 5000
```

```
      root_path:    /usr/local/d8o/iwmbase/...
```

```
      debug:        0          # 0|1 starman ...
```

```
      testmode:     0          # catalyst test ...
```

```
      app:          debug     # (app|debug).psgi ...
```

```
    log:
```

```
      debug:        0          # enable debug
```

```
      level:        info       # debug only in dev
```

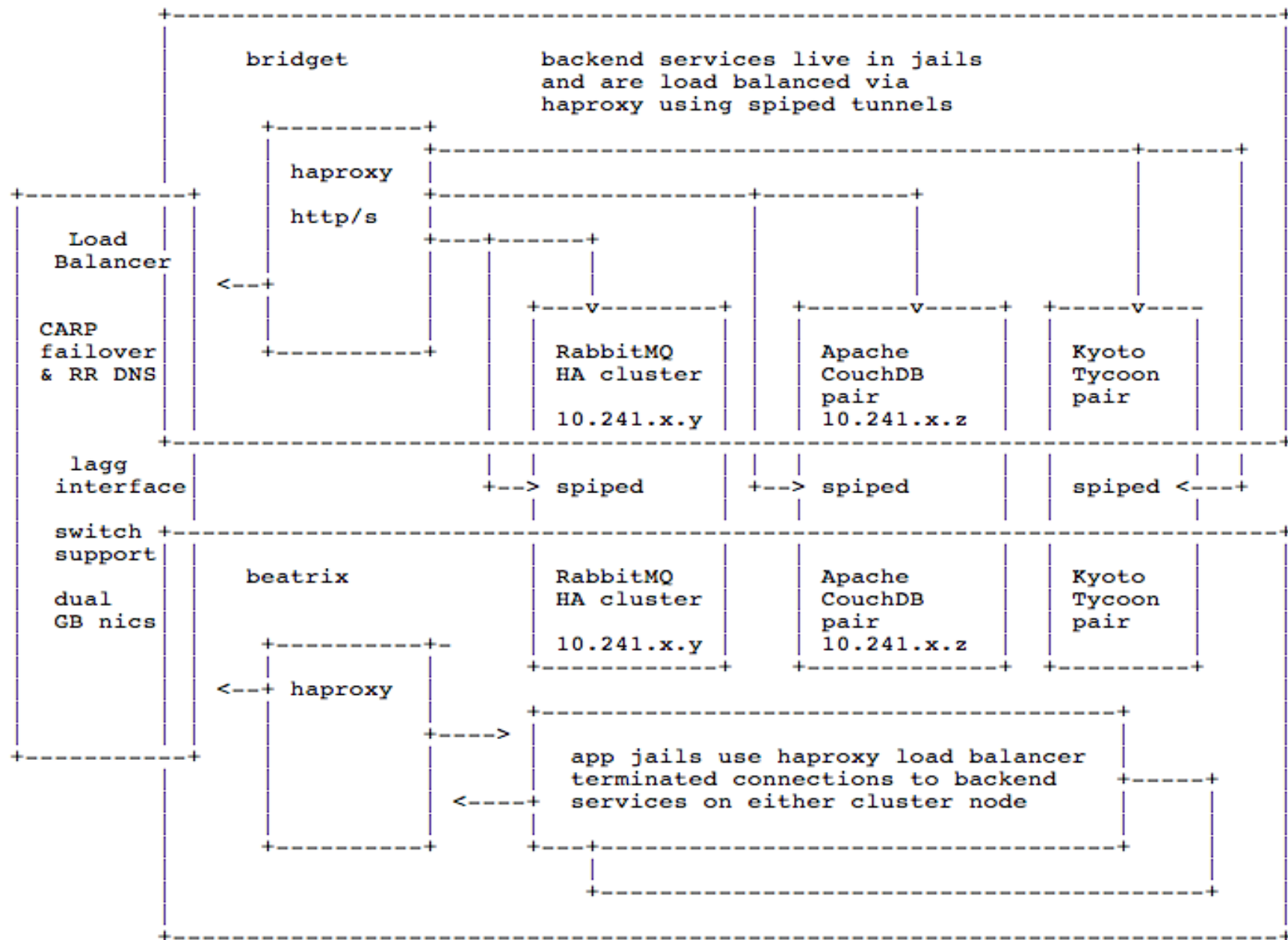
```
    nginx:
```

```
      root_path:    /usr/local/d8o/iwmn-frontend/templates
```

```
    hipchat:
```

```
      token:        "{{ lookup('vault',  
                               'secret/iwmbase',  
                               'hipchat_token') }}"
```

Cluster



Interlude the Second

- DBs migrated & first apps behind haproxy
- one morning, *everything* wedged completely
- different everything - networks, h/w vs VMs, service providers
- 3 other people reported same event, same time
- FreeBSD 11.x only, not 10.x, different haproxy versions
- gremlins? Aliens? Port scanners? Ping'o'death?

Interlude the Second

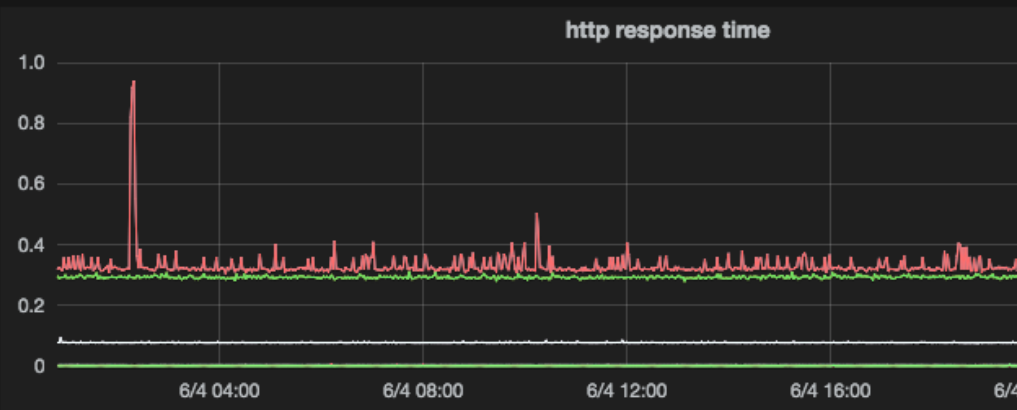
“Haproxy relies on **signed integer wraparound** on overflow, however this is really an undefined behavior, so **the C compiler is allowed to do whatever it wants**, and clang does exactly that, and that causes problems when the timer goes from $\leq \text{INT_MAX}$ to $> \text{INT_MAX}$, and explains the various hangs reported on **FreeBSD every 49.7 days**. To make sure we get the intended behavior, use **-fwrapv** for now. [...]

Many thanks to David King, Mark S, Dave Cottlehuber, Slawa Olhovchenkov, Piotr Pawel Stefaniak, and any other I may have forgotten for reporting that and investigating.”

Pro Ops Yak Herder Jedi



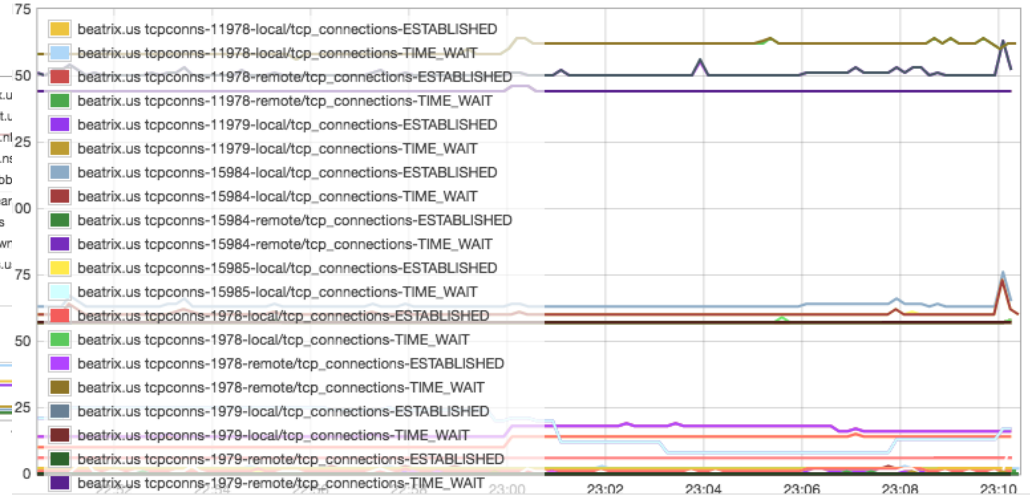
Riemann - Real Time



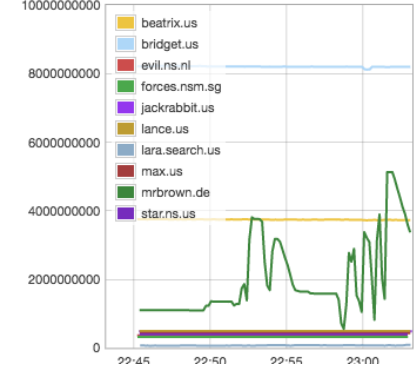
bridget



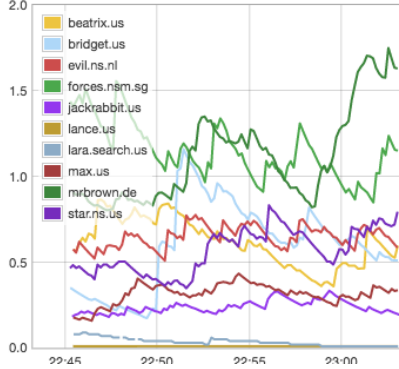
beatrx



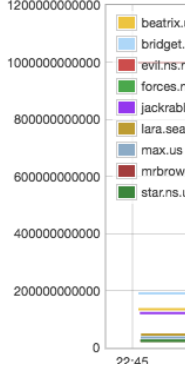
memory



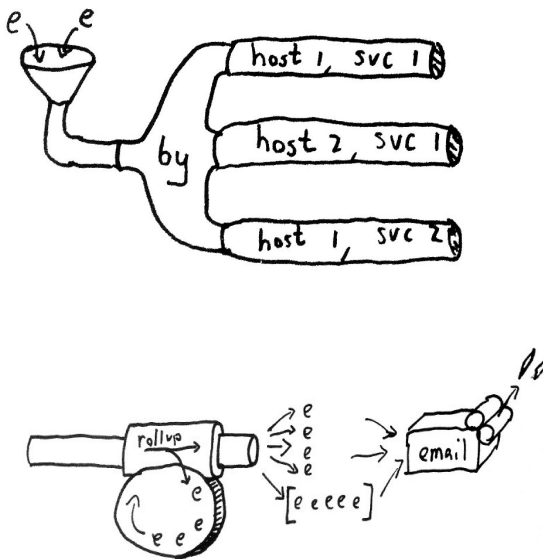
load



disk



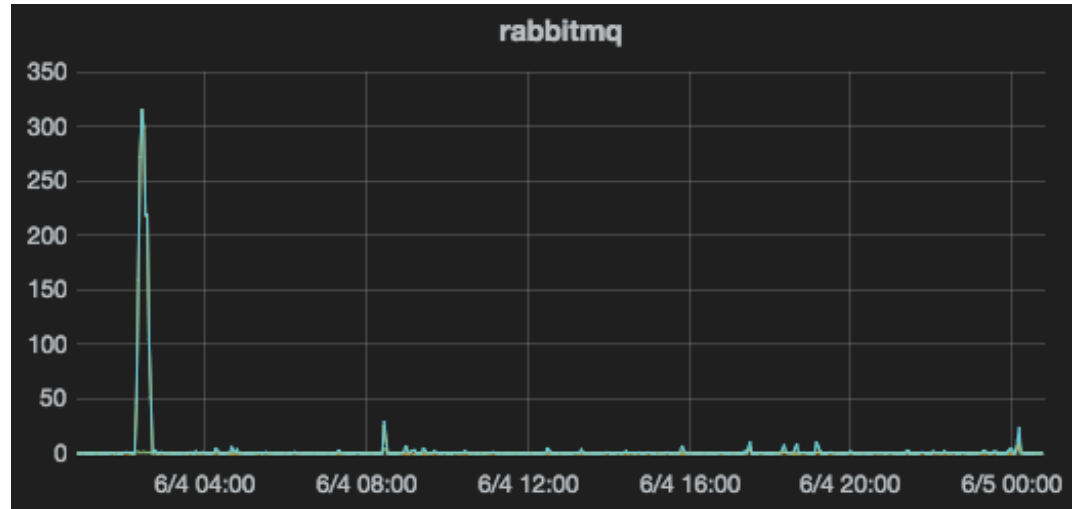
Riemann - Streams



```
(streams
  (default {:ttl      300
            :state    "ok"
            :service   "undefined"
            :host      "riemann.{{ config.domain }}"

            ; index all events immediately before any further transformation
            index
            ; events explicitly tagged for pagerduty or pushover always get sent
            (where (tagged "push") push)
            (where (tagged "page") pager)
            ; flip state if queue backlog grows unreasonably
            (where (service "curl_json-rabbitmq/gauge-queue_totals-messages")
              (with {:ttl      120
                    :service   "rabbitmq backlog"
                    :tags      ["notify"]})
                (where (or (> metric 300) (state "expired"))
                  (with :state "critical" reinject)
                  (else
                    (with :state "ok" reinject)))
                )))
            (where (service "rabbitmq backlog") pager)
            ; send to local graphite instance
            graph
            ; Calculate an overall rate of events and insert into index
            ; clear tags and description first
            (with {:metric      1
                  :service     "riemann.events.second"
                  :host        "riemann.{{ config.domain }}"
                  :tags        "riemann"
                  :description nil}
              (rate 5 index))
```

Riemann - App Events



`jackrabbit.us` rabbitmq monitoring critical

Mon Jun 05 2017 01:06:44 GMT+0200 (CEST) ttl 120 tags: rabbitmq

`jackrabbit.us` rabbitmq monitoring ok

Mon Jun 05 2017 01:03:08 GMT+0200 (CEST) ttl 120 tags: rabbitmq

Monitoring operational

rabbit

	jackrabbit.us	beatrix.us	bridget.us
rabbitmq monitoring	ok		
rabbitmq.object_totals.channels	0		
rabbitmq.object_totals.connections	0		
rabbitmq.object_totals.exchanges	7		
rabbitmq.object_totals.queues	116		
rabbitmq.queue_totals.messages	1		
rabbitmq.queue_totals.messages_ready	0		
rabbitmq.queue_totals.messages_unacknowledged	1		
		curl-kyoto-primary/gauge-connections	3
		curl-kyoto-primary/gauge-records	50958
		curl-kyoto-primary/gauge-repl_delay	0
		curl-kyoto-primary/gauge-repl_interval	0
		curl-kyoto-primary/gauge-sys_mem_peak	428040192
		curl-kyoto-primary/gauge-sys_mem_rss	428040192
		curl-kyoto-primary/gauge-sys_mem_size	428040192
			411619328

Interlude the Third

- databases, message queues, catalyst app all migrated
- lets move the job queue processors
- I plan 2 weeks holiday in Sardinia starting Saturday
- major backend problems – but site is still usable**
- some connections to rabbitmq are decidedly flakey
- tweak all the things
- un-tweak all the things
- check spiped tunnels
- disable pf everywhere
- enable pf everywhere again
- slow panic sets in
- I finally work it out after tcpdump/ngrep at 23h00 Friday

My Bad

- haproxy connections are tuned for HTTP keep-alive
- aggressive timeouts on connections to avoid spiped blowout
- rabbitmq reverse proxied TCP connections inherit these defaults
- queues with low or no message volume have their TCP sessions dropped
- perl is not fast enough on restarting workers to keep pm
- enable heartbeats in all rabbit libraries that support them
- perl ofc doesnt but erlang / elixir do
- change haproxy idle connection time to



Thoughts & Wants

- FreeBSD is interstellar ninja hipster tech: but
- noobie networking is unclear:
 - what are epairs, tun,tap, clone_if?
 - how to make typical jail networks?
 - is VIMAGE really really safe?
- want jail.conf + pf.conf + openbgp with zfs properties
- libxo everywhere all the time
- streaming libxo API of “system events”
- can I have @ in GeCOS fields?
- systemd-like service declaration, restart/mgmt
- mheilhy/jobd ?
- dtrace & tcpdump/ngrep are your friends
- tcp support for base syslogd
- “zfs send” for freebsd-update

Thanks

- adrian
- amdmi3
- az
- bap
- brnr
- calomel.org
- cperciva
- cooltrainer.org
- demon
- dumbbell
- emaste
- eviljg
- feld
- gblach
- junovitch
- kev009
- koobs
- krion
- marcel
- mat
- mmacy
- matthew
- mdexter
- miwi
- mwluca
- olgeni
- pi
- rakuo
- swills
- sunpoet
- wblock
- wen
- will
- xmj

- #bsdports on IRC
- mmacy for drm-next
- everybody@ports
- everybody@re
- everybody@docs
- BSDNow & Techsnap 'casters
- iwantmyname team

Thanks!

Questions?

@dch



<https://home.apache.org/~dch>



iwantmyname