

Count

100
50

Graphite at Scale

Scaling Graphite and giving back to open
source at the same time

Acquia background

- Focused on Drupal--the largest open source project in the world
- Fastest growing software company in America (500+ employees)
- Several engineering teams building various products (Cloud Services, Developer Tools, Personalization & Recommendation Engines, Drupal modules, etc.)
- Offices: Burlington MA, Portland OR, Reading UK



Acquia Cloud

- PaaS optimized for Drupal
- 8000+ AWS instances
- 6 AWS Regions
- 5 PB/mo data transfer
- 30 billion origin hits/mo
- 15 DevOps engineers
- 15 Ops engineers

Large traffic sites...



460M Hits to grammy.com
on night of award show

Large global sporting event...

37B TOTAL HITS
112K / SEC PEAK

Acquia

Lots of data / Lots of dashboards

Flypaper Cloud ▾ Site Factory ▾ ES ▾ Ops ▾ Network ▾ Search ▾

Flypaper

The Flypaper dashboard features several monitoring panels:

- Cloud Test Monitor
- Current ACE Perfmon Status
- Amazon Instance Totals
- Graphite Dashboard Proxy
- NN History - AC Prod
- NN Status - AC Prod
- NN Scenarios - AC Prod
- NN Coverage - AC Prod
- Search Farm health - AS Prod

A large, detailed dashboard titled "CloudAPI_Overall_Health" is displayed on the right side, containing multiple line graphs and charts showing system performance metrics over time.

Acquia Confidential

Why we love Graphite?

- Easy to use
 - Get data in
 - Get data out
- We can ask it questions like:
 - What is our average load across all ‘web’ nodes?
 - Which balancers are the most heavily loaded?
 - Is a spike in PHP errors correlated to customer doing a code deploy?
- Incredibly rich ecosystem

Things we don't love...

- Complicated to install
 - Many different pieces
 - Many different languages
- Hard to find metrics or dashboards without something like Grafana
- ...
- Hard to scale

Our Graphite problems...

- Lots of graphite data
 - ~25 Graphite servers globally
 - 650,000+ metrics today
- Lots of graphite problems
 - No way to elastically scale as we add metrics
 - No way to query across clusters
 - No way to scale/replicate globally
 - WhisperDB filesystem overhead
 - Vision for 10x as many metrics

Artur's solution will only go so far



So what's our history with Graphite?

Phase 1

One-off Graphite box

One-off Graphite boxes

```
#!/bin/bash
#
# Install and start statsd on a Lucid server.
# See https://i.acquia.com/wiki/everything-about-statsd for details.
#
# Ubuntu 10.04, us-east, ephemeral: ami-1db20274
# Ubuntu 10.04, us-east, EBS: ami-c7b202ae
#
# facct default
# ec2-run-instances -t c1.medium -k default -g ops AMI
# ec2-create-tags INSTANCEID --tag name=stats-1 --tag created_by=$USER
# scp statsd.sh ubuntu@$SERVERNAME:
# ssh $SERVERNAME -t ubuntu@server sudo -H ./statsd.sh
#
# For stats-1.acquia.com:
# ec2-associate-address 174.129.202.134 -i INSTANCEID

set -ex

echo "Enabling root login"
cp /home/ubuntu/.ssh/authorized_keys /root/.ssh/authorized_keys

echo "Install lots o' packages"
apt-get update
apt-get -y install build-essential apache2 libssl-dev git-core python-cairo python-django python-django-tagging
che libapache2-mod-wsgi libapache2-mod-python python-ldap bzr daemon git-core

echo "Download and prepare to install"
cd $HOME
wget -q http://launchpad.net/graphite/0.9/0.9.9/+download/whisper-0.9.9.tar.gz
wget -q http://launchpad.net/graphite/0.9/0.9.9/+download/carbon-0.9.9.tar.gz
wget -q http://launchpad.net/graphite/0.9/0.9.9/+download/graphite-web-0.9.9.tar.gz

export GRAPHITE_ROOT=/opt/graphite
export PYTHONPATH=$GRAPHITE_ROOT
if ! [ -d /mnt/graphite ]; then
    mkdir /mnt/graphite
    ln -s /mnt/graphite /opt/graphite
fi

echo "Install Whisper (fast)"
cd $HOME
tar xzf whisper-0.9.9.tar.gz
cd whisper-0.9.9
python setup.py build
python setup.py install
```

AWS Instance Stats



Phase 2

Supported ‘stats’

server type

 protochron 4 days ago CL-8289 | Closes #544: Statsd checks should reference the Upstart job

9 contributors



150 lines (127 sloc) | 3.918 kb

Raw Blame History

```
1  # Statsd class install statsd, whisper, graphite, carboncache
2  #
3  # @todo
4  #   This class does too much, each requirement of statsd should be another
5  #   class that is required by this class.
6  class stats::stats {
7    require nodejs::nodejs
8    include httpd::httpd_python
9
10   include acquia-util::eh
11
12   $statsd_version = '0.6.0-1'
13
14   file {"/vol/ebs1/gfs/graphite":
15     require => [ Package["httpd-packages"], Class["acquia-util::var_www_symlink"], Class["filesystem::local"] ],
16     ensure => directory,
17     mode => '0775',
18     group => 'www-data',
19     owner => 'www-data',
20   }
21
22   file {"/opt/graphite":
23     require => File["/vol/ebs1/gfs/graphite"],
24     ensure => link,
25     mode => '0775',
26     target => '/vol/ebs1/gfs/graphite',
27   }
28
29   # @todo Remove after this is hotfixed to 1.78
30   file {"statsd-upstart-link":
31     ensure => link,
32     name  => "/etc/init.d/statsd",
33     target => "/lib/init/upstart-job",
34   }
```

Supported 'stats' server type

Acquia

Why do we need so many stats servers?

Acquia Platform Health

Acquia Sites Add-ons Teams Subscriptions Help 192 Matt

ACCOUNTS.ACQUIA.COM ▾ Getting started 0 of 4 tasks

Insight Cloud Search Git URL ▾

Workflow Platform Health Domains Servers Databases Logs New Statistics Backups Drush and API Cron

Prod Last month ▾

Cache and load balancing

HTTP response codes

Count

HTTP 404 (blue line) HTTP 200 (green line)

Jul 23 Jul 30 Aug 6 Aug 13

CPU and memory usage

Memory (blue line) CPU (green line)

Percentage (%)

Jul 23 Jul 30 Aug 6 Aug 13

Application

CPU and memory usage

Memory (blue line) CPU (green line)

Percentage (%)

ACQUIA

EventHorizon



- What
 - A Ruby EventMachine based system to send system metrics to Graphite
- Why from scratch?
 - We evaluated lots of options (CloudWatch, Nagios, Diamond, etc.)
 - We wanted something easy for us to extend
 - We were very comfortable with Ruby

EventHorizon

- Plugins for various services
 - Easy to write new plugins
 - Plugins auto-enable themselves
- Varnish
 - lru_evictions
 - total hits / misses / etc.
- Nginx
 - Response codes by type
- Other
 - CPU % Idle
 - Volume IO and Usage
 - PHP error count
 - etc.

```
require 'stats/varnishstats'

module EventHorizon
  class Varnish < EH::Plugin
    name "varnish"

    def stats
      begin
        @stats = VarnishStats.fetch
      rescue VarnishStatsError
        return nil
      end

      result = []
      result <<-{:name => 'varnish.cache_hits', :value => @stats.cache_hits}
      result <<-{:name => 'varnish.cache_hitpasses', :value => @stats.cache_hitpasses}
      result <<-{:name => 'varnish.cache_misses', :value => @stats.cache_misses}
      result <<-{:name => 'varnish.lru_evictions', :value => @stats.lru_evictions}
      result <<-{:name => 'varnish.num_objects', :value => @stats.num_objects}
      result <<-{:name => 'varnish.memory_allocated', :value => @stats.mem_allocated}
    end
  end
end
```

Plugin list:

- cpu
- disk
- heartbeat
- memached
- memory
- mysql
- netstat
- network
- nginx
- php-errors
- process
- puppet
- stats-collector
- tomcat
- varnish

**But now we have 500k+
metrics and know we'll
need to scale to millions...**

Options for scaling Graphite

- Host it yourself
 - Get really, really big SSDs
 - or shard the data
 - or use one of the newer DBs like InfluxDB
- Build it yourself
 - Roll your own massive time series database
- Use a vendor
 - HostedGraphite.com
 - DataDog?
 - etc.
- Why outsource?
 - Free up our team to tackle other problems
 - Improve reliability
- Why not outsource?
 - Cost
 - Latency
 - Data sovereignty
 - Lock-in

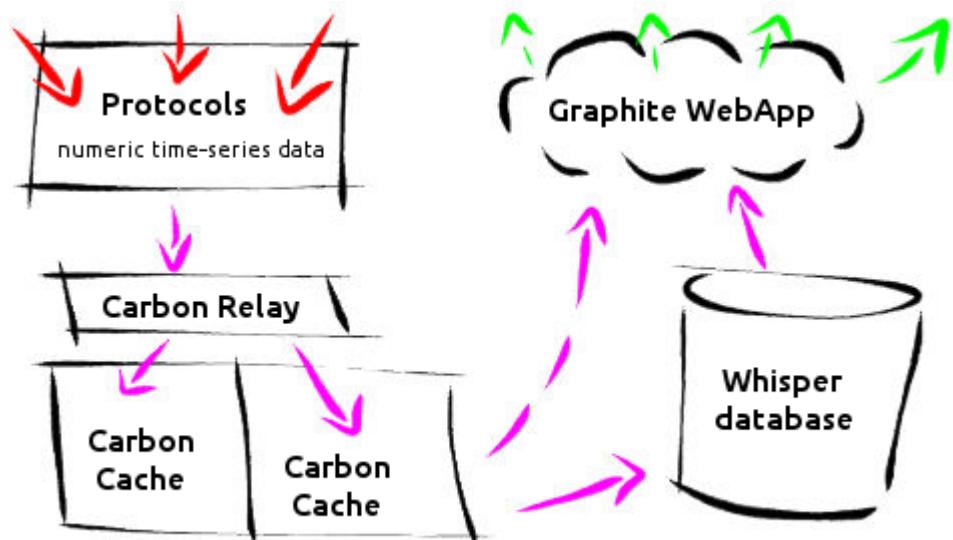


Phase 3

A scalable solution...

Normal Graphite components

- Graphite (Django web app)
- Carbon cache (cache Graphite uses)
- Carbon (daemon handling metrics)
- Carbon-relay (talks to backend carbon servers)
- Whisper - the underlying time series database
- Statsd (often used in combination with Graphite)



Thanks: <http://scalingup.eu/>

Our Solution

- **Cassandra** as a backend for open source Graphite
- Hackathon project - Acquia Build Week 2013
 - We had a Graphite plugin sending data to a Cassandra cluster
 - Great PoC
- Why Cassandra?
 - Optimized for writes
 - We're very familiar with Cassandra (our Mollom spam blocking service analyzes billions of spam messages in Cassandra yearly)
 - We believe in open source
 - Other options weren't ready yet (InfluxDB, etc.) or didn't solve our use case (Ceres)



Our Graphite Solution

- New Carbon cassandra plugin
 - <https://github.com/acquia/carbon-cassandra-plugin/>
- New Graphite cassandra plugin
 - <https://github.com/acquia/graphite-cassandra-plugin>
- **Nemesis** - automation for launching Cassandra & Graphite clusters
- How does it scale?
 - Benchmarked to 250k data points per minute, per Carbon-node (C* mostly idle)

- What it solves
 - Automatically creating Graphite & Cassandra clusters on AWS
 - CloudFormation API is a pain (hand-coding big JSON blobs to it)
 - Monitoring clusters via CloudWatch (JMX, etc.)
 - Backing up C* SSTables to S3
- How we're using it?
 - Spinning up CloudFormation autoscaling clusters for Cassandra and Graphite, Zookeeper, ...

Nemesis



mythindex.com

```

module Nemesiss::Aws::Templates
  # This template uses IAM Roles and S3 buckets created by the bootstrap template
  # and *Requires* that to be in place before launching this.
  class Cassandra < ServerBase
    attr_reader :_seeds, @_cluster_size, @_vpc, @_timeout, @_os

    def initialize(opts={})
      options = {
        :seeds => 1,
        :cluster_size => 5,
        :vpc => false,
        :timeout => 43200,
        :mappings => {
          :hypervisor => 'hvm',
          :storage => 'ebs',
        },
        :os => Nemesiss::DEFAULT_OS,
        :instance_type => "m3.xlarge",
      }.merge(opts)

      if options[:seeds] < 1
        raise Exception, "Must have at least 1 seed node!"
      end

      super(options)

      @description = "AWS CloudFormation template to spin up a Cassandra cluster"
      @_seeds = options[:seeds]
      @_cluster_size = options[:cluster_size]
      @_vpc = options[:vpc]
      @_timeout = options[:timeout]
      @_os = options[:os]

      # Parameters
      instance_type = Nemesiss::Aws::CloudFormation::Parameter.new("InstanceType", :default => options[:instance_type])

      cassandra_cluster_size = Nemesiss::Aws::CloudFormation::Parameter.new("CassandraClusterSize",
        :description => "Number of cassandra servers in the cluster.",
        :type => "Number",
        :default => @_cluster_size,
        :min_value => @_cluster_size)
    end
  end
end

```

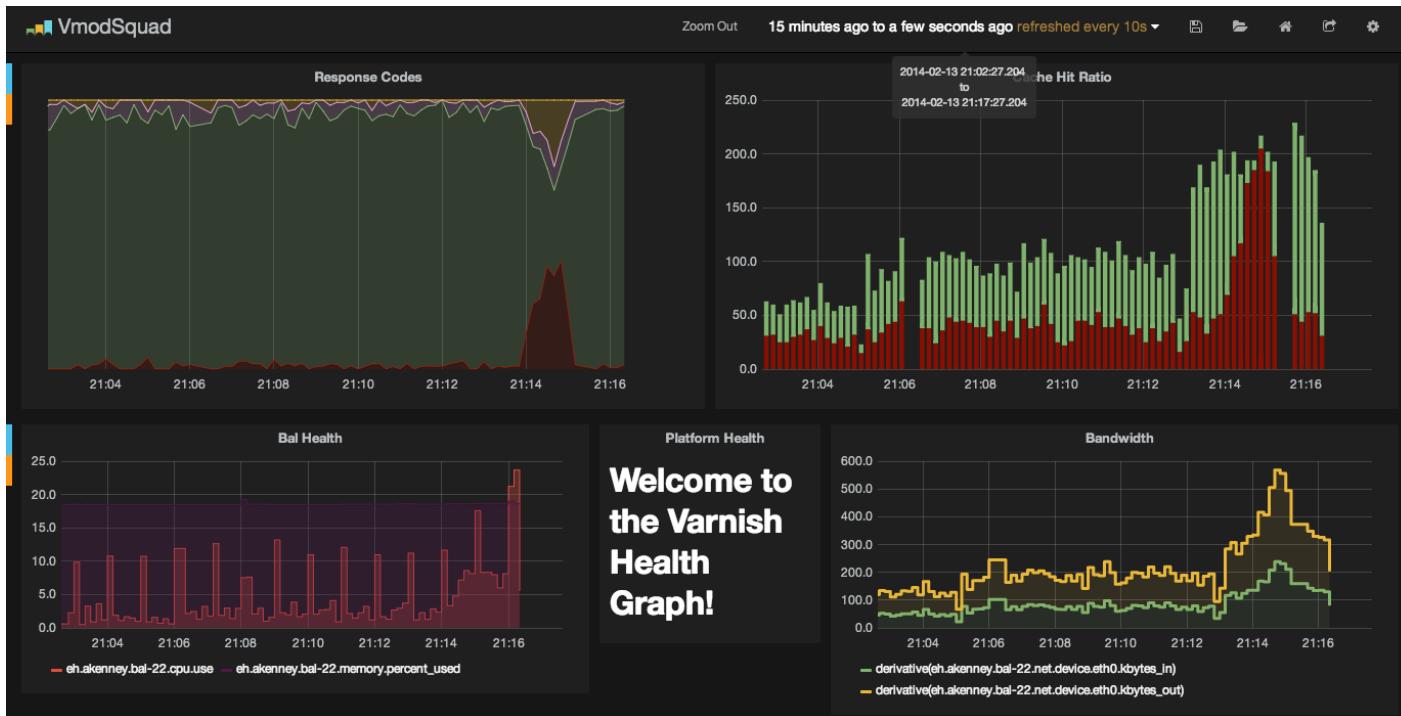
**But we still have
stats problems**

Varnish per-site-stats

- Big Data Problem
- 100,000+ Drupal websites
- Each website we want to track many stats
 - 200/300/400/500 response codes
 - hit / miss / pass
 - backend response time
- `stats.$varnish_box.$virtual_host.$response_code.$hit_miss_pass`
 - e.g. `stats.varnish-101.domain_com.500.pass`
 - Millions of possible metrics



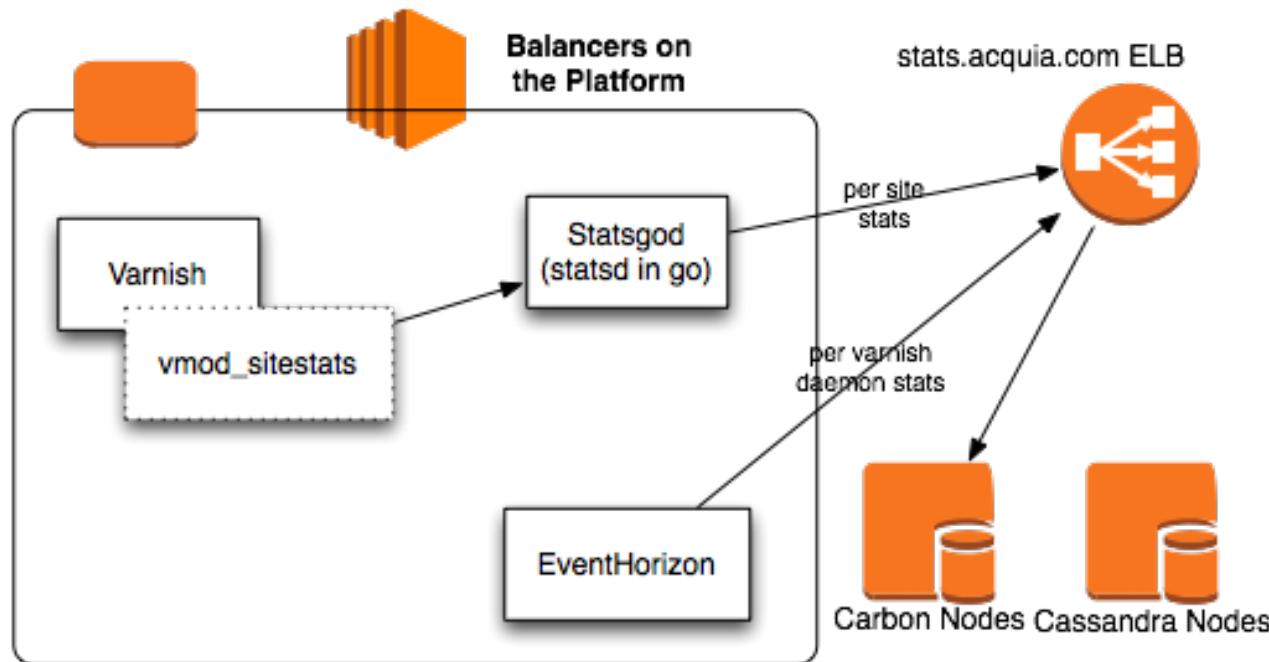
ACQUIA BUILD WEEK 2014



Hackathon Vmudsquad

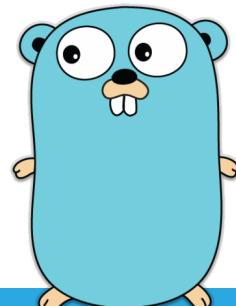
ACQUIA®

Varnish Stats Architecture



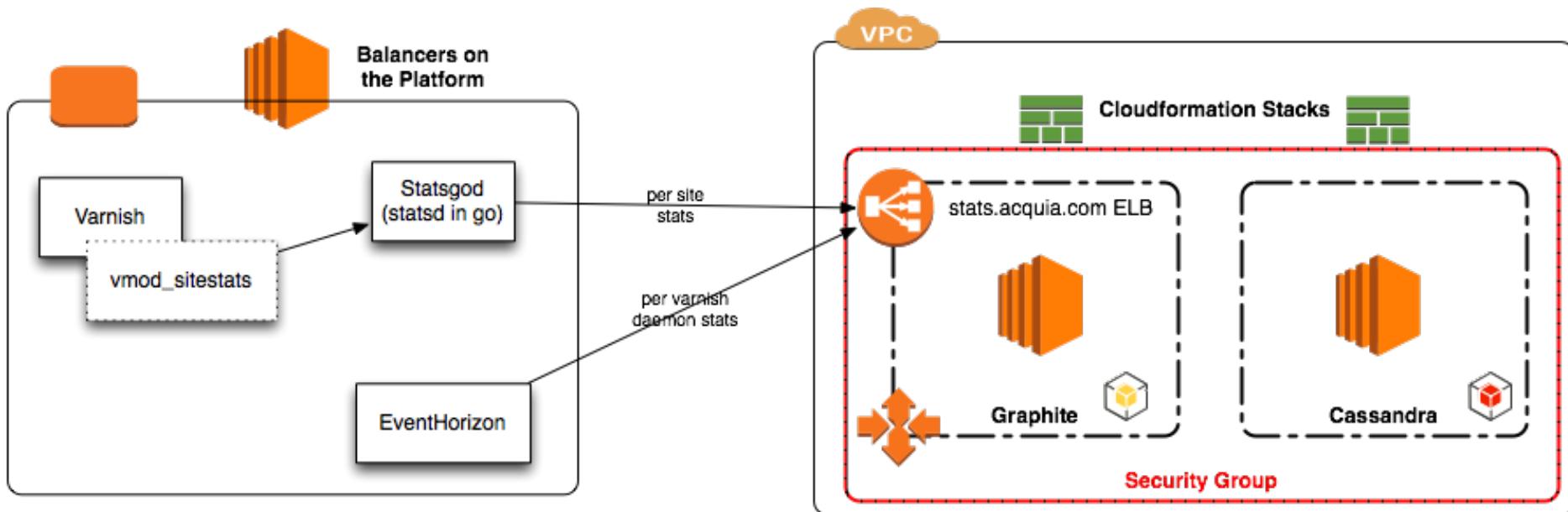
Statsgod

- What? Statsd in Go
- Why? Wanted to run Statsd locally
 - Don't want flood of packets out for every hit
 - Don't want to maintain complicated stats in Varnish memory
 - Worried about UDP packets being lost
 - Don't really want to run NodeJS everywhere
- Why Go in particular?
 - Great concurrency
 - Single binary
 - ... why not?



The future...

The ‘final’ architecture

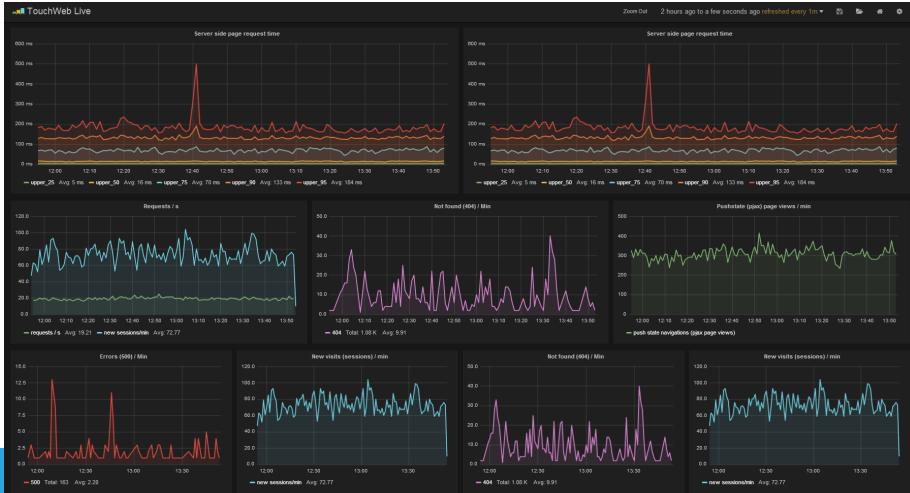


Long term plan...

- Scalable Graphite (megacarbon plugins, cassandra plugin, etc.)
 - PR open for ~11 months
 - <https://github.com/acquia/graphite-web/tree/db-plugin>
- Nemesis
 - Planning on open sourcing
- Statsgod - already open source <https://github.com/syrneus/statsgod>
- EventHorizon
 - open source?
 - pluggable backend?
 - replace EventMachine with Celluloid?
 - switch to Go? replace entirely with Diamond?

Long term plan part 2

- More monitoring/alerting on the underlying Graphite data
- Use more Graphite ecosystem projects
 - Etsy's Skyline, Kale, Oculus
 - More Grafana



The End...

Andrew Kenney

andrew.kenney@acquia.com

@syrneus

@AcquiaCloud



Acquia is hiring - bit.ly/acquiajobs

- DevOps engineers
- Distributed Systems engineers
- Cloud Operations
- VP IT
- Security Director
- ... Drupalists!