graphite & monitoring

monitoring solutions

- system monitoring
- application monitoring
- marketing
- anything

monitoring solutions

- system monitoring
    - nagios and forks (icinga, opsview…)
    - munin
    - ganglia
    - others...

monitoring solutions

- application monitoring
    - new relic
    - sentry (for exceptions)

monitoring solutions

- marketing (customer count, costs etc)
    - librato metrics
    - mixpanel

monitoring solutions

- monitoring everything (arbitrary metrics)
    - graphite
    - opentsbd (http://opentsdb.net/)
    - kairosdb (https://code.google.com/p/kairosdb/)
    - roll your own :)
        - rrdtool (http://oss.oetiker.ch/rrdtool/)
        - your own rrd

so wtf is rrd ?

- round robin database
- algorithm

so wtf is rrd ?

seconds

20140208-05:06:01: 12

20140208-05:06:02: 80

20140208-05:06:03: 9

….

20140208-05:07:00: 30

so wtf is rrd ?

minutes

20140208-05:06: 12

round(from seconds 05:06:00,05:06:59)

20140208-06:00: 90

so wtf is rrd ?

hours

20140208-05: 12

round(from minutes 05:06,05:06)

20140208-06: ...

graphite

- don’t re-invent the wheel
- monitor everything
- graphing something

echo “somewhere.somemetric 12 20140801” | nc graphiteserver 2003

graphite install

tips:
* DON’T install it in virtualenv
* use ubuntu

instructions:

    github...

demo

lets do something useful

$ curl -o /dev/null -m 5 -s -w "app.frontpage.time_total %{time_total}\n" http://localhost:9001

$ curl -o /dev/null -m 5 -s -w "app.frontpage.time_total %{time_total} `date +%s`\n" http://localhost:9001 | nc localhost 2003

$ while true; do curl -o /dev/null -m 5 -s -w "app.frontpage.time_total %{time_total} `date +%s`\n" http://localhost:9001 | nc localhost 2003; sleep 5; done

add another metric

$ while true; do

{ curl -o /dev/null -m 5 -s -w "app.frontpage.time_total %{time_total} `date +%s`\n" http://localhost:9001 &&

curl -o /dev/null -m 5 -s -w "app.articlepage.time_total %{time_total} `date +%s`\n" http://localhost:9001/show/foo_99999; } | nc localhost 2003; sleep 5; done

add more - logster

tails logs

$ sudo logster --dry-run --output=graphite --graphite-host=localhost:2003 MongodbParser /var/log/mongodb/mongodb.log

statsd - time it

install

- git clone ….
- npm i
- cp configExample.js config.js
- ./bin/statsd config.js
- pip install statsd (http://statsd.readthedocs.org/)

statsd - time your functions

```
import statsd
statsd = StatsClient(host='127.0.0.1',
                     port=8125)

@statsd.timer('index')

# or use timer at your own, (http://statsd.readthedocs.org/en/latest/timing.html)
```

statsd - counters

Counters are the most basic and default type. They are treated as a count of a type of event per second, and are, in Graphite, typically averaged over one minute. That is, when looking at a graph, you are usually seeing the average number of events per second during a one-minute period.

```
import statsd
statsd = StatsClient(host='127.0.0.1',
                     port=8125)


statsd.incr('index_pageview')
```

Questions ?