

METRICS, METRICS EVERYWHERE

*MAKE BETTER DECISIONS USING
NUMBERS*

BETTER **MOBILE** MARKET KNOWLEDGE

18 B MARKET DATAPPOINTS

1.4 M APPS MONITORED

42 COUNTRIES

5+ APP STORES

EMPOWERING **MOBILE** ENTERPRISES

Autodesk®

PEARSON

intuit.®

CHASE 



Danfoss



 zendesk
LOVE YOUR HELP DESK

AS SOFTWARE ENGINEER
YOU'LL
BE WRITING
CODE

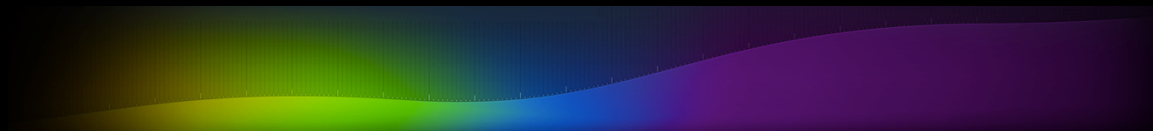
Adapted from Coda Hale



BUT YOU'LL GET PAYCHECK IF

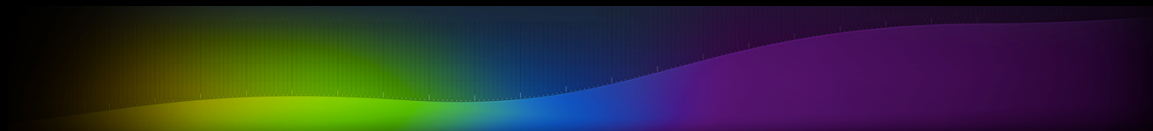
...

CODE → BUSINESS VALUE



WHAT THE HELL IS **BUSINESS VALUE?**

Hint: it's not a new scripting language

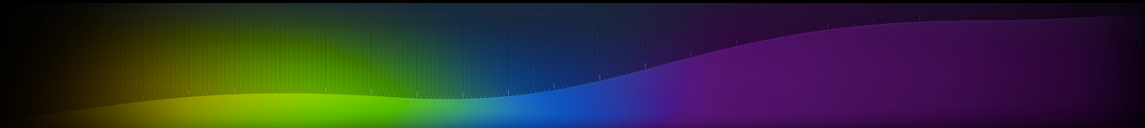
A decorative wavy line at the bottom of the slide, transitioning from yellow on the left to green in the middle, and then to blue and purple on the right.

A NEW FEATURE

AN IMPROVED EXISTING
FEATURE

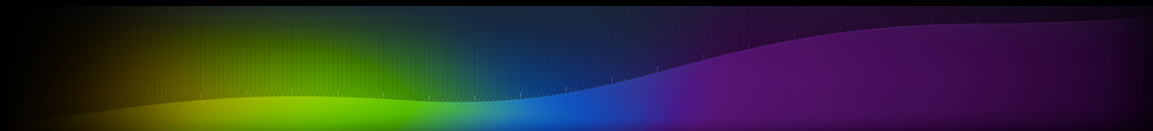
CODE WITH LESS BUGS

NOT PISSING OFF YOUR
USERS WITH A SLOW SITE

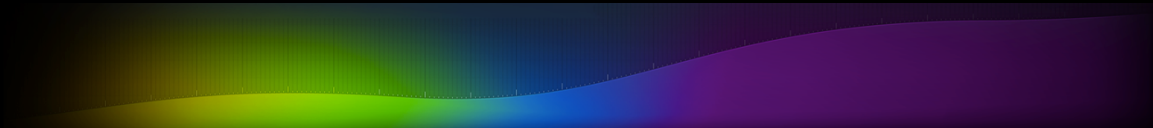


MAKING FUTURE CHANGES
EASIER

ADD A UNIT TEST **BEFORE**
FIXING THAT BUG



BUSINESS VALUE IS ANYTHING
THAT MAKES PEOPLE MORE
LIKELY TO GIVE US **MONEY**.



WE NEED TO DELIVER MORE
BUSINESS VALUE.

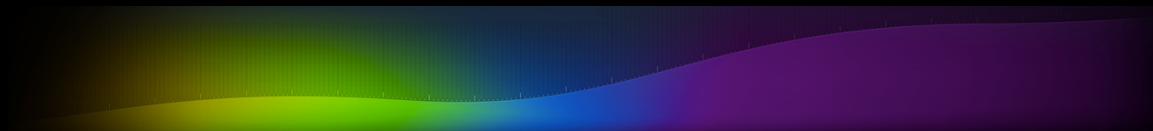
WE NEED TO MAKE **BETTER**
DECISION ABOUT OUR **CODE.**



OUR CODE GENERATES
BUSINESS VALUE WHEN...

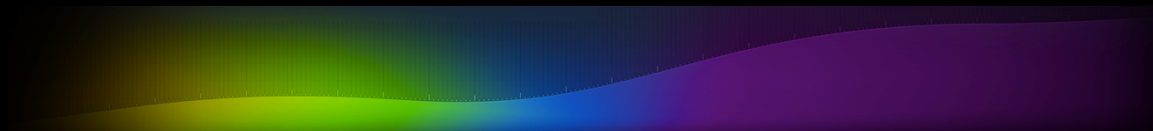
IT RUNS!

NOT WHEN WE WRITE IT!



WE NEED TO **KNOW** WHAT OUR
CODE DOES WHEN IT RUNS.

WE CAN'T DO IT UNLESS WE
MEASURE IT.

A decorative wavy line at the bottom of the slide, transitioning from green on the left to blue and then purple on the right.

WHY MEASURE IT?

WE WROTE THE CODE.
WE KNOW WHAT IT DOES!

RIGHT?!

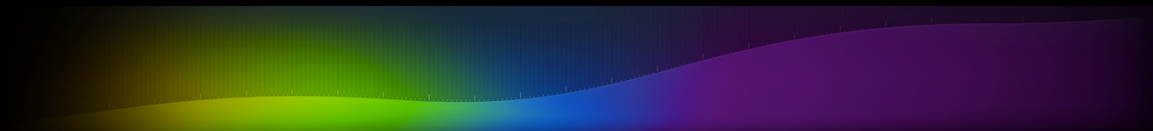
A decorative wavy line at the bottom of the slide, transitioning from green on the left to blue and then purple on the right.

WRONG!

ANALOGY:

MAP $\neq$ TERRITORY

PERCEPTION $\neq$ REALITY

A decorative wavy line at the bottom of the slide, transitioning from yellow-green on the left to blue and then purple on the right.

WE HAVE A **MENTAL MODEL** OF
WHAT THE CODE DOES.

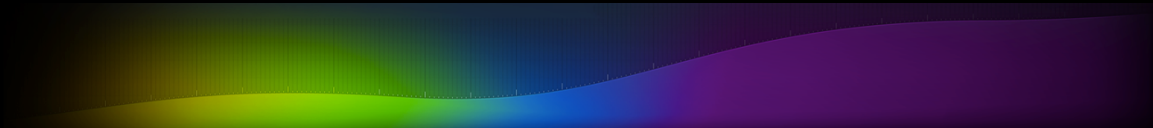
MENTAL MODEL $\neq$ CODE

A decorative wavy line at the bottom of the slide, transitioning from yellow on the left to green in the middle, and then to blue and purple on the right.

“THIS CODE CAN’T POSSIBLY FAIL”

(IT DOES!)

MENTAL MODEL IS OFTEN **WRONG**



WHICH CODE IS FASTER?

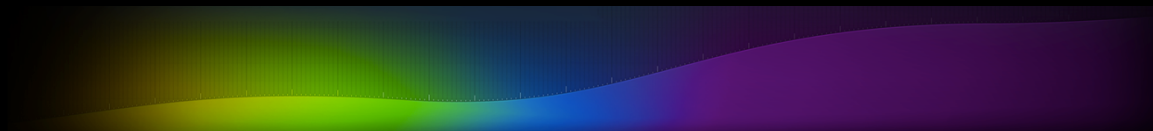
```
items.sort_by { |i| i.name }
```

```
items.sort_by { |a, b| a.name <=> b.name }
```

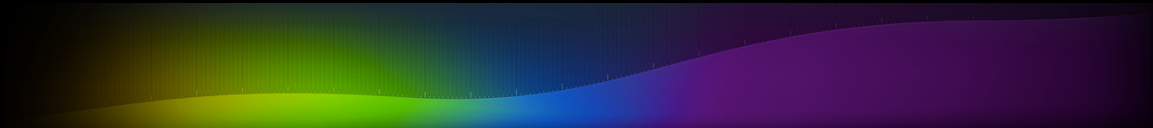
```
def sort_by(&blk)
  sleep(100) # FIXME: I AM POISON
  super(&blk)
end
```

WE DON'T KNOW

```
def sort_by(&blk)
  # TODO: make me not explode
  raise Exception.new("Haw Haw")
end
```



WE CAN'T KNOW UNTIL WE
MEASURE IT.



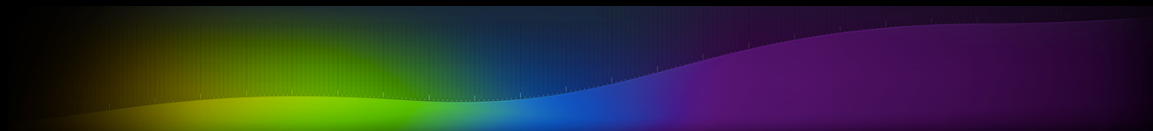
“THIS PAGE IS SLOW! IT TAKES
500ms TO LOAD. ***FIX IT!***”

Find the bottleneck

SQL Query

Template Rendering

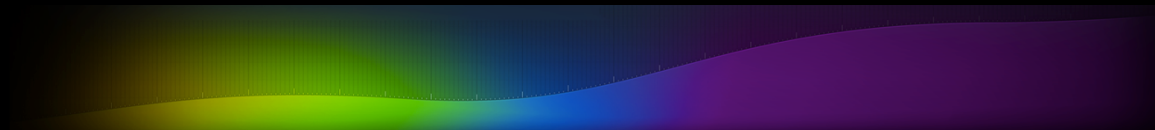
Session Storage



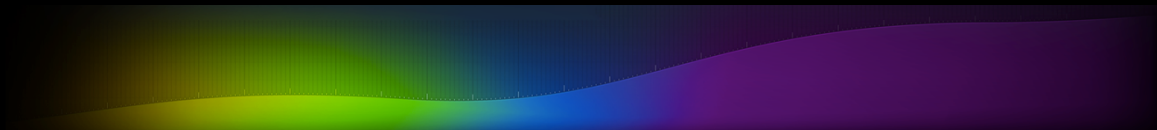
WE DON'T KNOW

Find the bottleneck 2.0!

<i>SQL Query</i>	53ms
<i>Template Rendering</i>	1ms
<i>Session Storage</i>	315ms

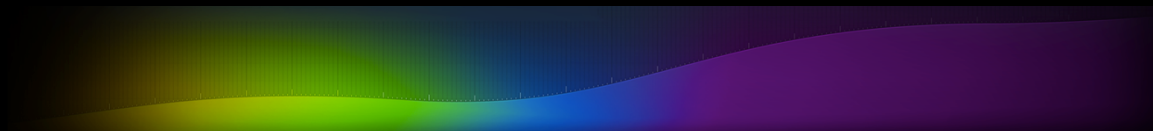


WE **IMPROVE** OUR MENTAL
MODEL BY **MEASURING** WHAT
OUR **CODE** DOES



A BETTER MENTAL MODEL
MAKES US **BETTER** IN
DECIDING WHAT TO DO.

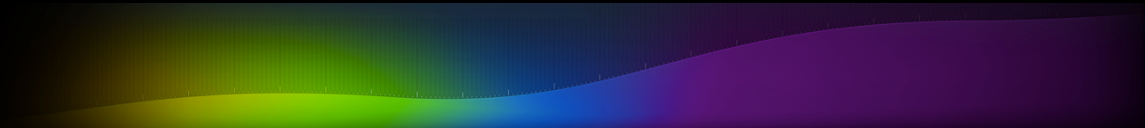
MAKES US **BETTER** IN
GENERATING **BUSINESS**
VALUE



MEASURING MAKES YOUR
DECISION BETTER

BUT ...

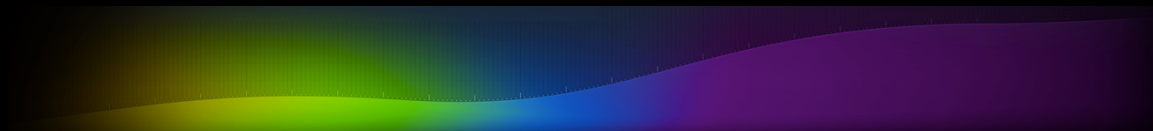
ONLY IF WE MEASURE THE
RIGHT THING!



WE NEED TO **MEASURE** THE
CODE OUT IN THE **WILD**.

WHILE GENERATING
BUSINESS VALUE

IN PRODUCTION



Autocomplete service for city names

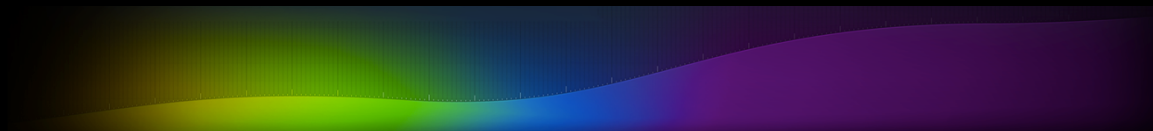
```
> GET /complete?q=San%20Fra
```

```
< HTTP/1.1 200 RAD
```

```
<
```

```
< ["San Francisco"]
```

What does this code do that
effects its **Business Value**?



And how can we measure that?

Gauges

The instantaneous value of something.

Counters

An incrementing and decrementing value.

Meters

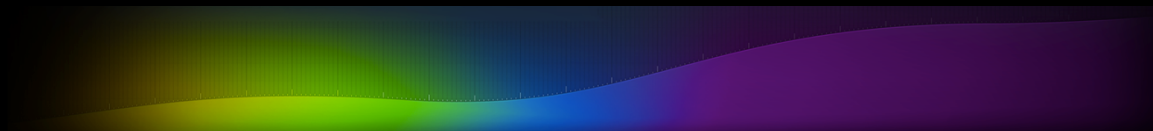
The average rate of events over a period of time

Histograms

The statistical distribution of values in a stream of data.

Timers

A histogram of durations and a meter of calls



Metric Code Examples

```
metrics.gauge("cities") { cities.size }
```

“The service has 589 cities registered.”

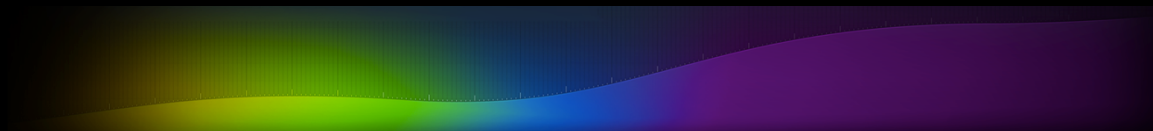
```
val counter =  
metrics.counter("connections")
```

“There are 594 active sessions on that server.”

```
counter.inc()  
counter.dec()
```

```
val meter = metrics.meter("requests",  
                           SECONDS)  
meter.mark()
```

“We went from 3,000 requests/sec to <500 a second.”



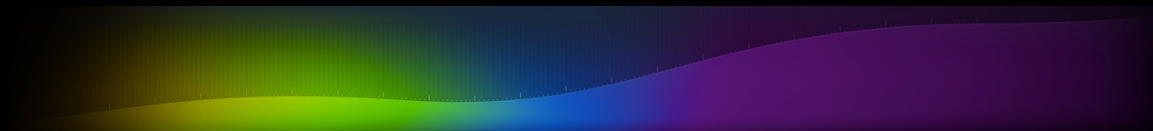
Metric Code Examples

```
val histogram =  
  metrics.histogram("response-sizes")  
  
histogram.update(response.cities.size)
```

“95% of autocomplete results return 3 cities or less.”

```
val timer = metrics.timer("requests",  
  MILLISECONDS,  
  SECONDS)  
  
timer.time { handle(req, resp) }
```

“At ~2,000 req/sec, our 99% latency jumps from 13ms to 453ms.”



Now What ?

Instrument it

If it could affect your code's business value, add a metric.

Collect it

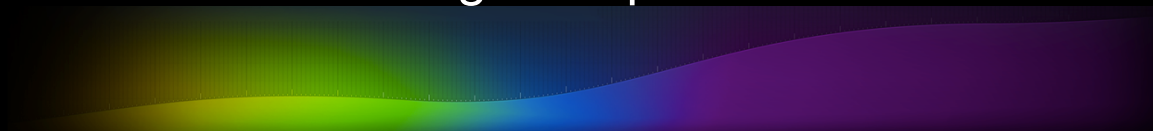
JSON via HTTP.
Every minute.

Monitor it

Nagios/Zabbix/Whatever
If it affects business value,
someone should get woken up.

Aggregate it.

Ganglia/Graphite/Cacti/Whatever
Place current values in historical
context.
See long-term patterns.



Observe

What is the 99% latency of our autocomplete service right now?

~500ms

Orient

How does this compare to other parts of our system, both currently and historically?

way slower

Decide

Should we make it faster?
Or should we add feature X?

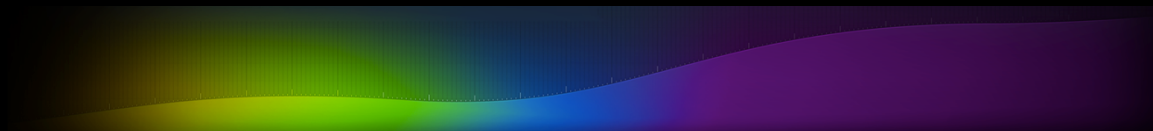
make it faster

Act

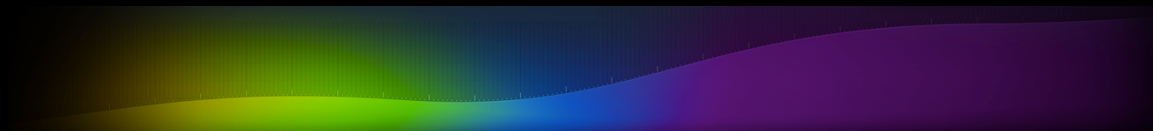
Write some code.

```
def sort_by(&blk)
  # sleep(100) # WTF DUDE!
  super(&blk)
end
```

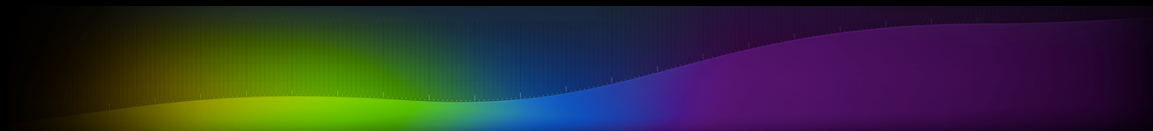
In order to know how well
our code is generating
business value, we need
metrics.

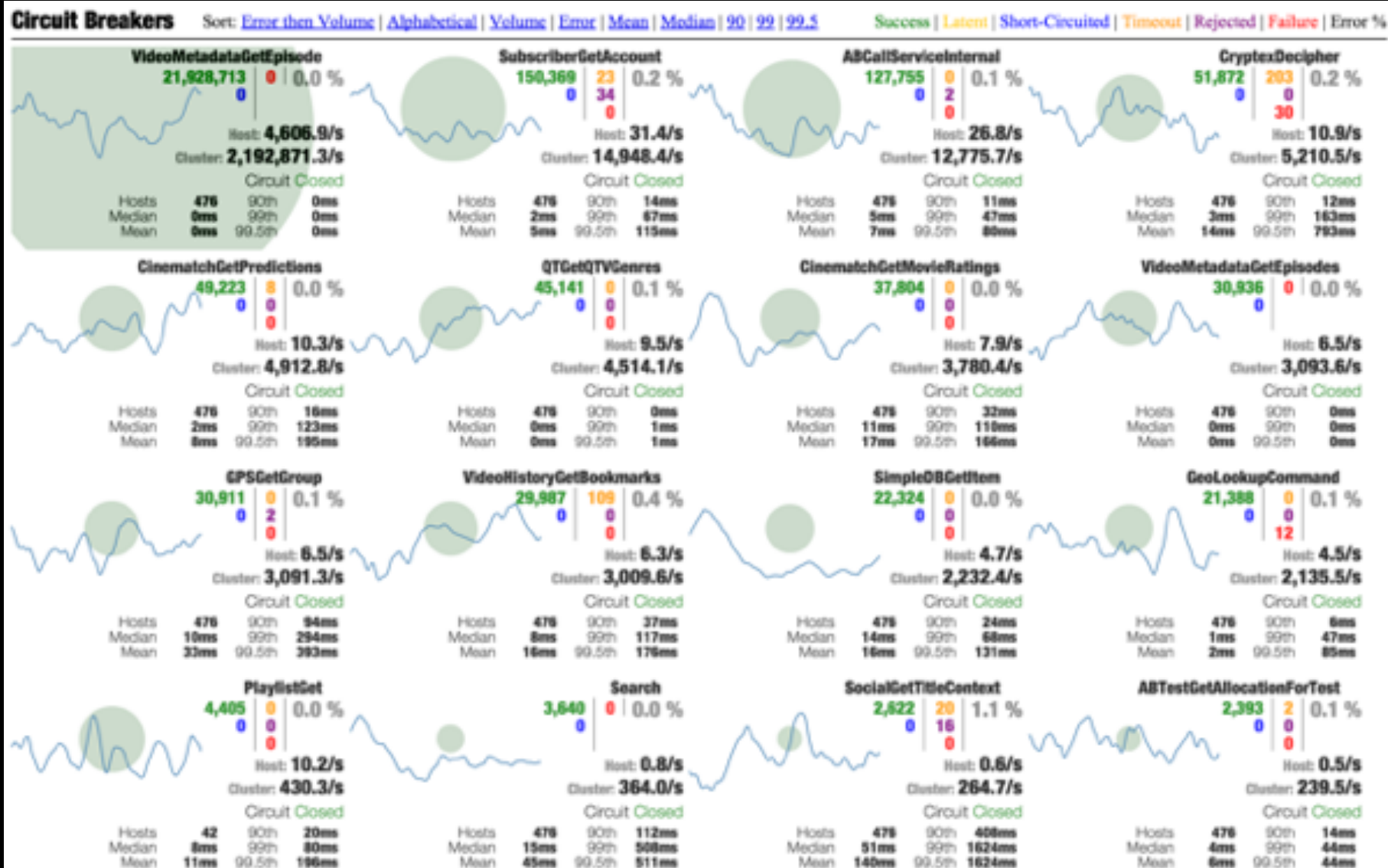


Make better **decisions**
by using **numbers**.



Make better **decisions**
by using **numbers**.





GECKOBOARD.COM



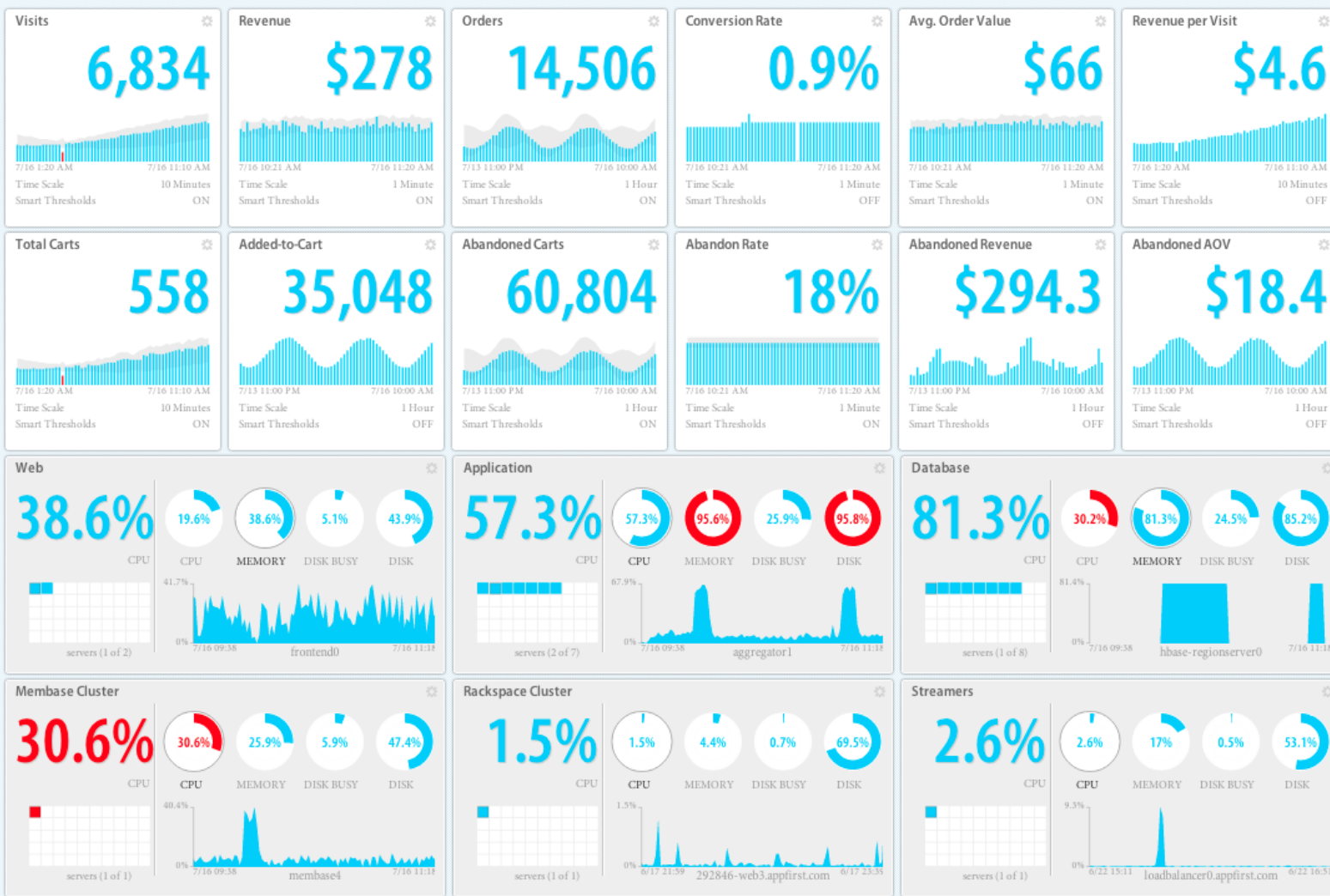
APPFIRST.COM



eCommerce Risks -

Hi Demo | logout

Mon Jul 16, 11:21 AM



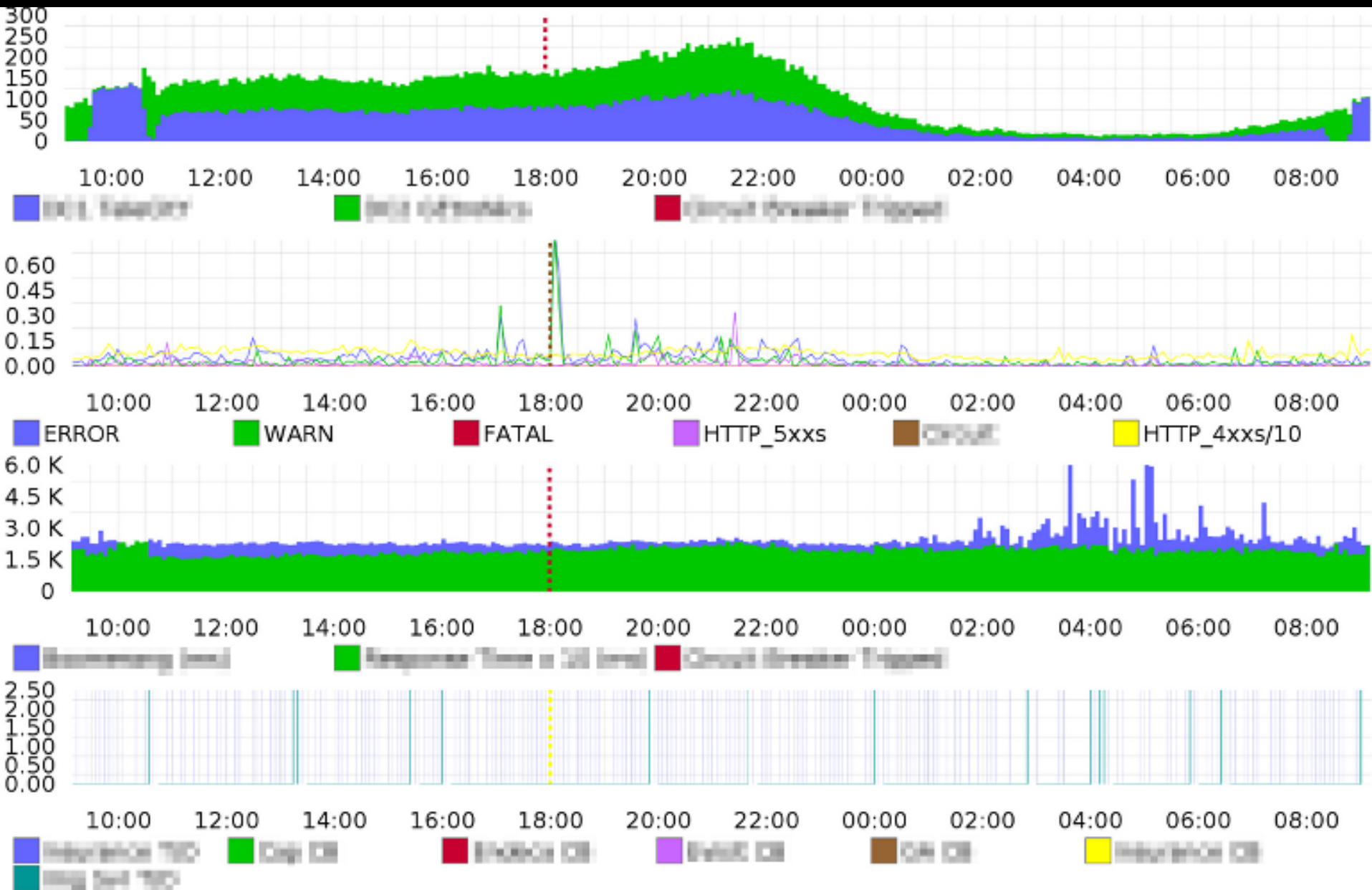
Type your message

Demo Admin
Jul 02, 2012 03:26pm
no worries memory on order

Demo Admin
Jul 02, 2012 09:22am
I see memory high

Demo Admin
Apr 19, 2012 03:21pm
High CPU on rackspace cluster.

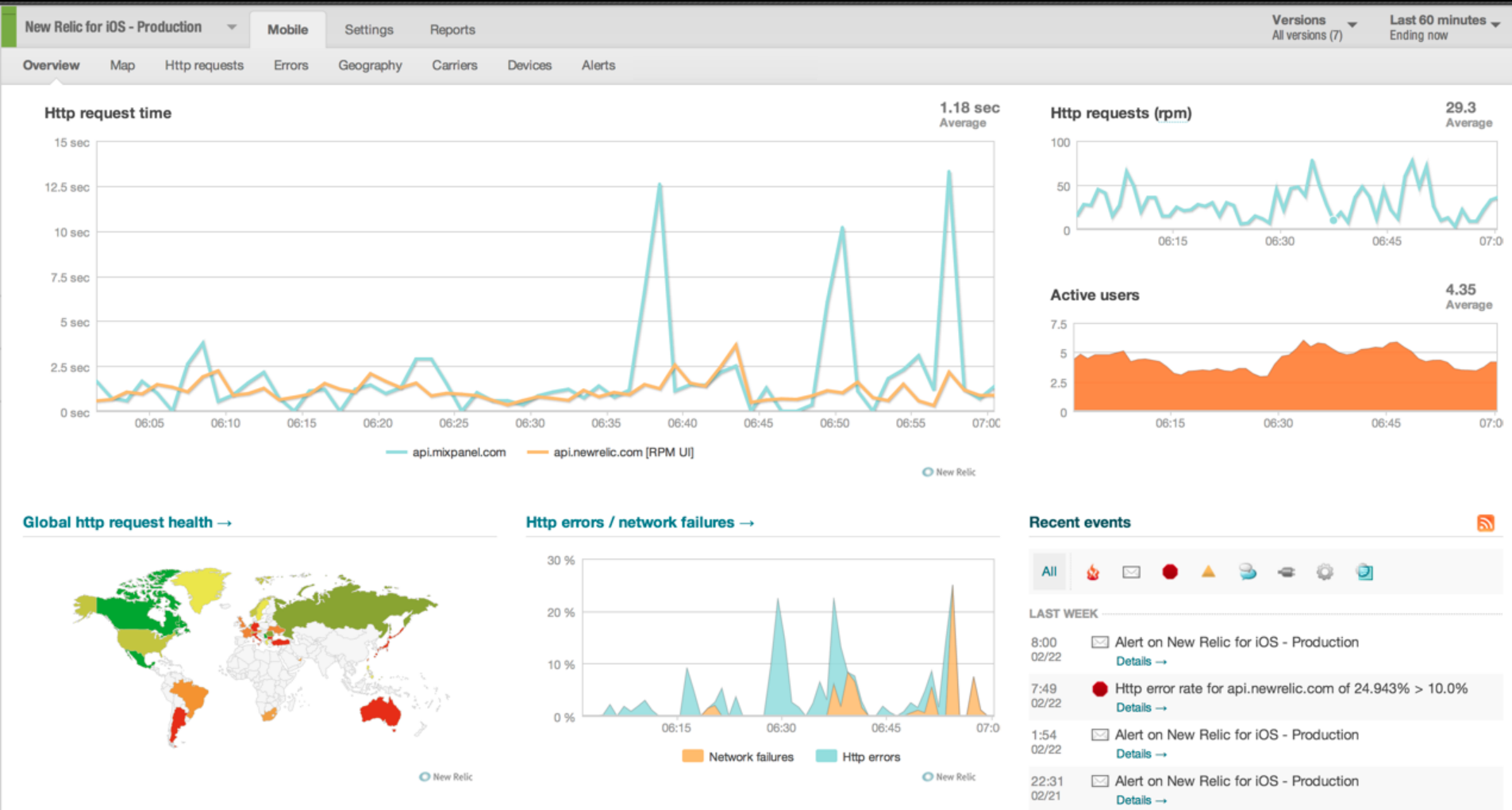
GRAFITE from ETSY.COM

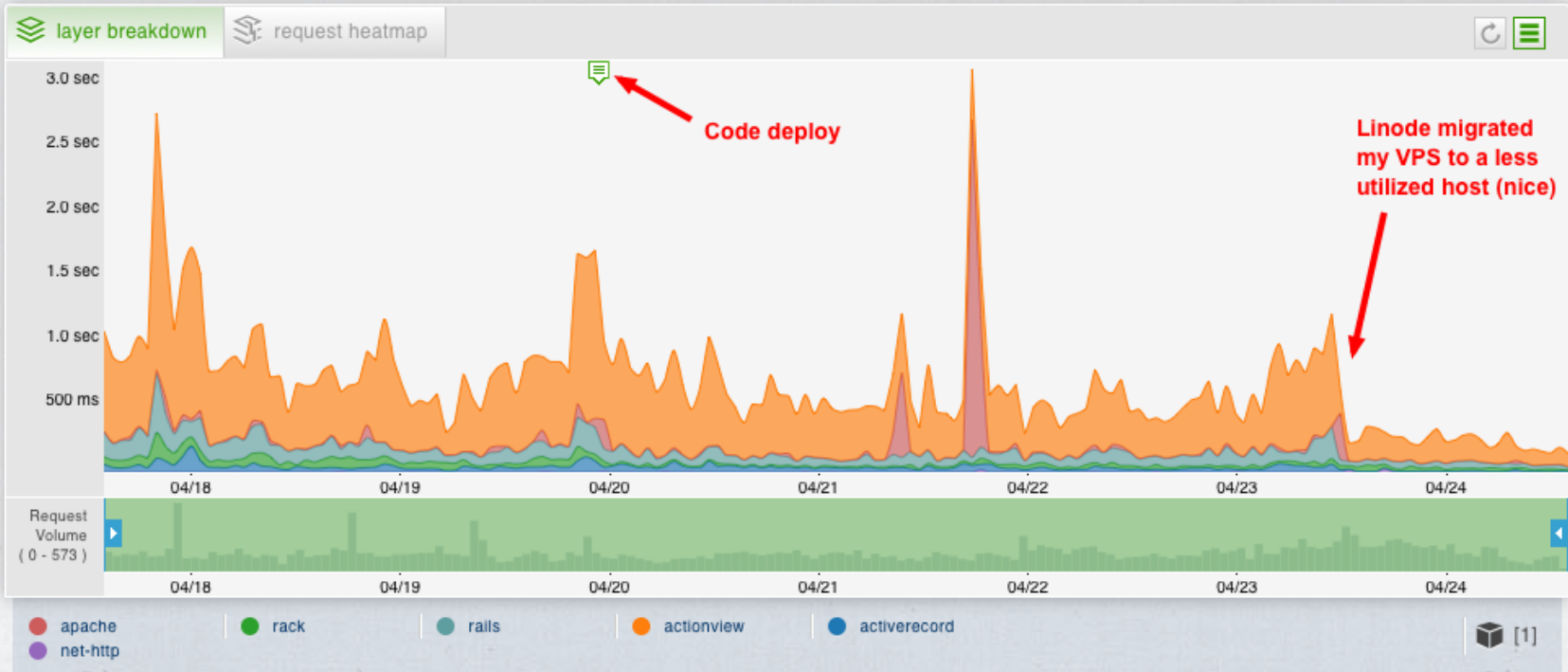


ETSY.COM



NEWRELIC.COM





Summary - 27,136 traces | 136 errors | 1 alerts

ActionView is consistently the worst performer (minimal view caching currently)

[04/17 02:00 PM - 04/24 02:00 PM]

Top URLs

Some of these actions need some TLC

sort by Freq x Avg. Duration

Domain	URL	Freq	Avg (ms)
gameface.in	/games	1272	1,287.41
gameface.in	/genre/action	1340	788.32
gameface.in	/recent/screenshots	336	2,132.95
gameface.in	/genre/simulation	433	835.34
gameface.in	/	427	821.43
gameface.in	/genre/role-playing	328	797.75
g.ameface.com	/	115	1,750.37
gameface.in	/genre/strategy	200	984.74
gameface.in	/genre/other	146	1,213.65
g.ameface.com	/profiles	41	2,757.88

show 10 20 40

Layers

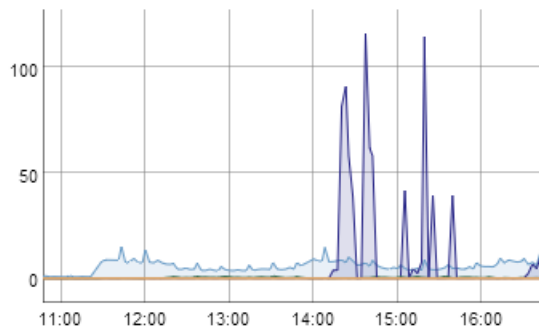
sort by Freq x Avg. Duration

Layer	Freq	Avg (ms)
actionview	26183	494.91
rails	26487	98.35
rack	26676	37.77
apache	27136	35.73
activerecord	26081	35.74
net-http	86	334.03

show 10 20 all

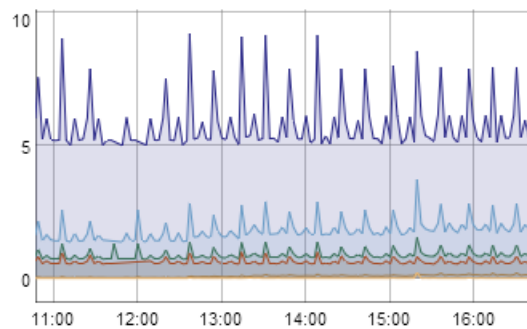
Cache Requests

opcounters



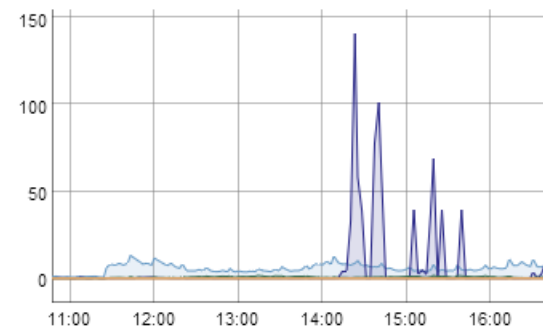
MopappMongoDB01:27018 • primary

opcounters



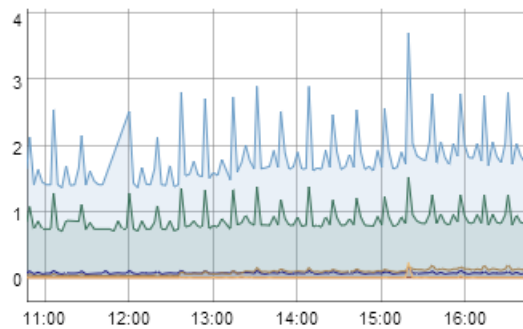
MopappMongoDB01:27019 • conf

opcounters



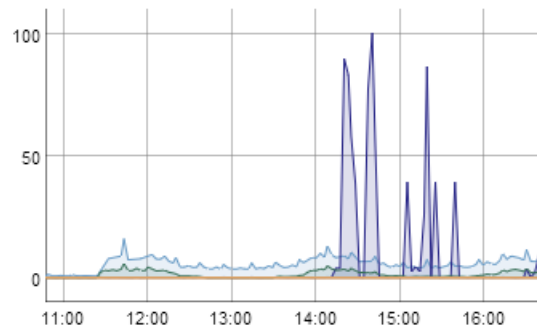
MopappMongoDB02:27018 • primary

opcounters



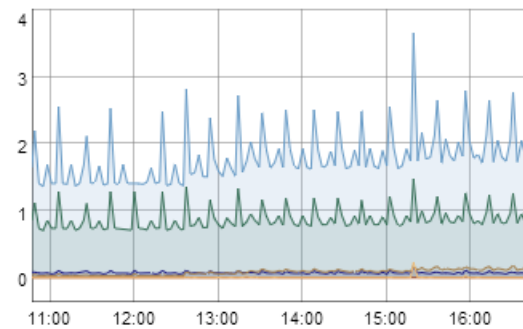
MopappMongoDB02:27019 • conf

opcounters



MopappMongoDB03:27018 • primary

opcounters



MopappMongoDB03:27019 • conf

REFERENCES:

<http://codeascraft.com> from ETSY

<http://codeascraft.com/2011/02/15/measure-anything-measure-everything/>

<http://metrics.codahale.com/>

<http://www.appneta.com/>

<http://newrelic.com/>

<http://www.appfirst.com/>

<http://www.geckoboard.com/>

