InfluxDB - a distributed time series, metrics, and events database

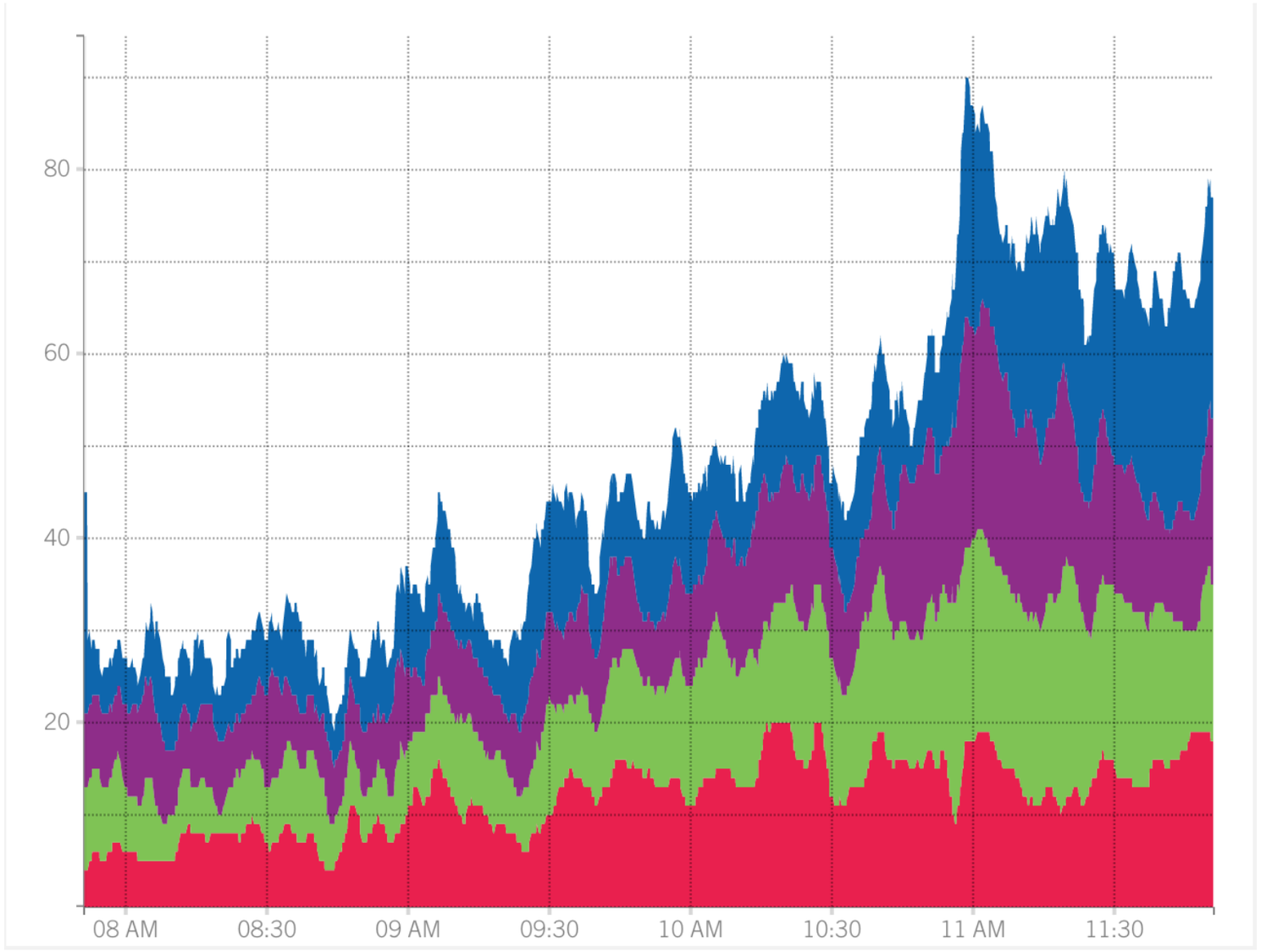
Paul Dix
paul@influxdb.com
@pauldix
@influxdb

YC (W13), 3 people full time:
Todd Persen
John Shahid
Paul Dix (me)

What it's for…

InfluxDB

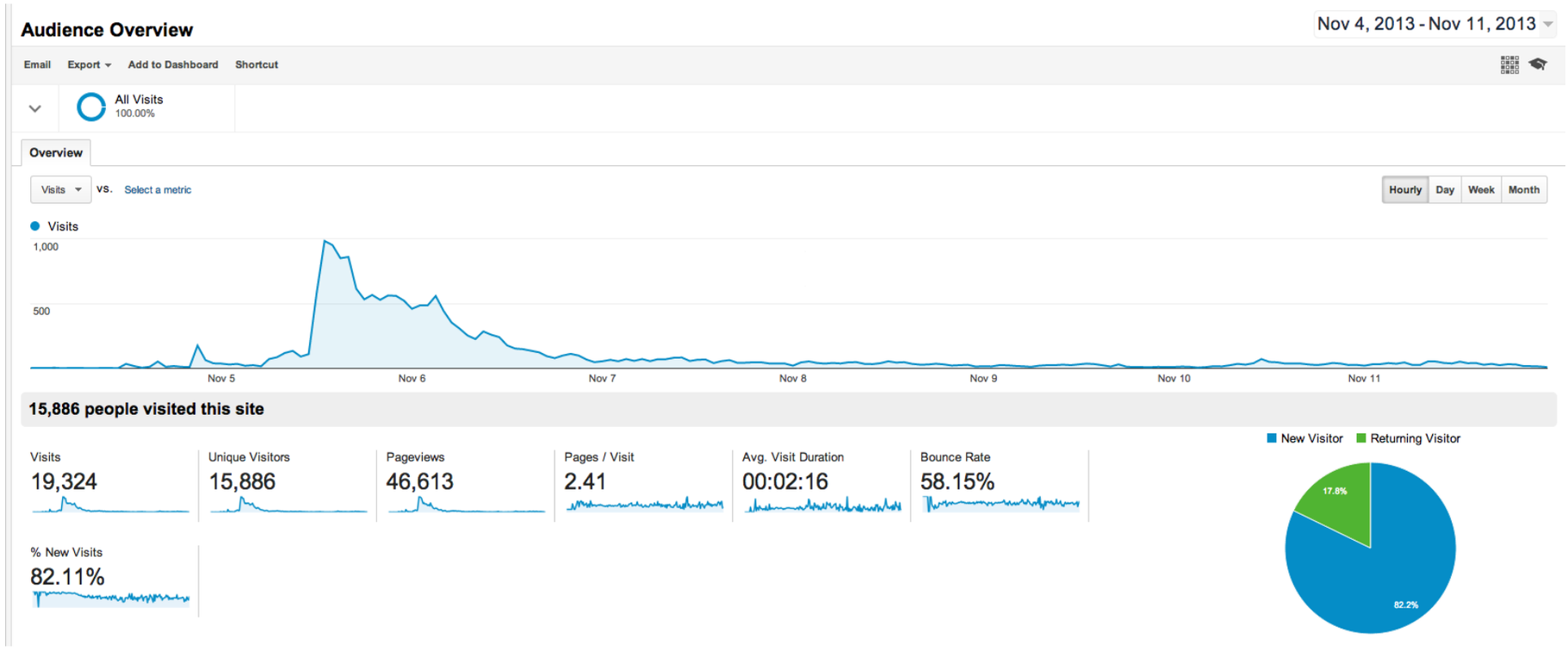
Metrics



Time Series


33.15 +0.08 (0.24%)
Oct 25 - Close
NYSE real-time data - Disclaimer
Currency in USD
Range 33.01 - 33.49
52 week 29.52 - 36.43
Open 33.21
Vol / Avg. 19.82M/22.63M
Mkt cap 154.01B
P/E 14.29
Div/yield 0.12/1.45
EPS 2.32
Shares 4.65B
Beta 1.08
Inst. own 61%
+1 199
Compare: Enter ticker here Add Dow Jones S&P 500 HPQ IBM MSFT CRM INTC CA SI
Zoom: 1d 5d 1m 3m 6m YTD 1y 5y 10y All
Oct 21, 2013 - Oct 25, 2013 +0.25 (0.76%)
33.50
33.25
33.00
32.75
Mon Oct 21
Tue Oct 22
Wed Oct 23
Thu Oct 24
Fri Oct 25
12 pm
2 pm
Volume (mil / 2min)
2.5
1.25
2012
2013
Settings | Technicals | Link to this view
Volume delayed by 15 mins.

Analytics


Audience Overview
Nov 4, 2013 - Nov 11, 2013
Email
Export
Add to Dashboard
Shortcut
All Visits
100.00%
Overview
Visits
vs.
Select a metric
Hourly
Day
Week
Month
Visits
1,000
500
Nov 5
Nov 6
Nov 7
Nov 8
Nov 9
Nov 10
Nov 11
15,886 people visited this site
Visits
19,324
Unique Visitors
15,886
Pageviews
46,613
Pages / Visit
2.41
Avg. Visit Duration
00:02:16
Bounce Rate
58.15%
% New Visits
82.11%
New Visitor
Returning Visitor
17.8%
82.2%

Events

```
64.242.88.10 - - [07/Mar/2004:16:05:49 -0800] "GET /twiki/bin/edit/Main/Double_bounce_sender?topicparent=Main.Configuration
64.242.88.10 - - [07/Mar/2004:16:06:51 -0800] "GET /twiki/bin/rdiff/TWiki/NewUserTemplate?rev1=1.3&rev2=1.2 HTTP/1.1" 200 4
64.242.88.10 - - [07/Mar/2004:16:10:02 -0800] "GET /mailman/listinfo/hsdivision HTTP/1.1" 200 6291
64.242.88.10 - - [07/Mar/2004:16:11:58 -0800] "GET /twiki/bin/view/TWiki/WikiSyntax HTTP/1.1" 200 7352
64.242.88.10 - - [07/Mar/2004:16:20:55 -0800] "GET /twiki/bin/view/Main/DCCAndPostFix HTTP/1.1" 200 5253
64.242.88.10 - - [07/Mar/2004:16:23:12 -0800] "GET /twiki/bin/oops/TWiki/AppendixFileSystem?template=oopsmore¶m1=1.12¶m2=1.
64.242.88.10 - - [07/Mar/2004:16:24:16 -0800] "GET /twiki/bin/view/Main/PeterThoeny HTTP/1.1" 200 4924
64.242.88.10 - - [07/Mar/2004:16:29:16 -0800] "GET /twiki/bin/edit/Main/Header_checks?topicparent=Main.ConfigurationVariabl
64.242.88.10 - - [07/Mar/2004:16:30:29 -0800] "GET /twiki/bin/attach/Main/OfficeLocations HTTP/1.1" 401 12851
64.242.88.10 - - [07/Mar/2004:16:31:48 -0800] "GET /twiki/bin/view/TWiki/WebTopicEditTemplate HTTP/1.1" 200 3732
64.242.88.10 - - [07/Mar/2004:16:32:50 -0800] "GET /twiki/bin/view/Main/WebChanges HTTP/1.1" 200 40520
64.242.88.10 - - [07/Mar/2004:16:33:53 -0800] "GET /twiki/bin/edit/Main/Smtpd_etrn_restrictions?topicparent=Main.Configurat
64.242.88.10 - - [07/Mar/2004:16:35:19 -0800] "GET /mailman/listinfo/business HTTP/1.1" 200 6379
64.242.88.10 - - [07/Mar/2004:16:36:22 -0800] "GET /twiki/bin/rdiff/Main/WebIndex?rev1=1.2&rev2=1.1 HTTP/1.1" 200 46373
64.242.88.10 - - [07/Mar/2004:16:37:27 -0800] "GET /twiki/bin/view/TWiki/DontNotify HTTP/1.1" 200 4140
64.242.88.10 - - [07/Mar/2004:16:39:24 -0800] "GET /twiki/bin/view/Main/TokyoOffice HTTP/1.1" 200 3853
64.242.88.10 - - [07/Mar/2004:16:43:54 -0800] "GET /twiki/bin/view/Main/MikeMannix HTTP/1.1" 200 3686
64.242.88.10 - - [07/Mar/2004:16:45:56 -0800] "GET /twiki/bin/attach/Main/PostfixCommands HTTP/1.1" 401 12846
64.242.88.10 - - [07/Mar/2004:16:47:12 -0800] "GET /robots.txt HTTP/1.1" 200 68
64.242.88.10 - - [07/Mar/2004:16:47:46 -0800] "GET /twiki/bin/rdiff/Know/ReadmeFirst?rev1=1.5&rev2=1.4 HTTP/1.1" 200 5724
64.242.88.10 - - [07/Mar/2004:16:49:04 -0800] "GET /twiki/bin/view/Main/TWikiGroups?rev=1.2 HTTP/1.1" 200 5162
64.242.88.10 - - [07/Mar/2004:16:50:54 -0800] "GET /twiki/bin/rdiff/Main/ConfigurationVariables HTTP/1.1" 200 59679
64.242.88.10 - - [07/Mar/2004:16:52:35 -0800] "GET /twiki/bin/edit/Main/Flush_service_name?topicparent=Main.ConfigurationVa
64.242.88.10 - - [07/Mar/2004:16:53:46 -0800] "GET /twiki/bin/rdiff/TWiki/TWikiRegistration HTTP/1.1" 200 34395
64.242.88.10 - - [07/Mar/2004:16:54:55 -0800] "GET /twiki/bin/rdiff/Main/NicholasLee HTTP/1.1" 200 7235
64.242.88.10 - - [07/Mar/2004:16:56:39 -0800] "GET /twiki/bin/view/Sandbox/WebHome?rev=1.6 HTTP/1.1" 200 8545
64.242.88.10 - - [07/Mar/2004:16:58:54 -0800] "GET /mailman/listinfo/administration HTTP/1.1" 200 6459
lordgun.org - - [07/Mar/2004:17:01:53 -0800] "GET /razor.html HTTP/1.1" 200 2869
64.242.88.10 - - [07/Mar/2004:17:09:01 -0800] "GET /twiki/bin/search/Main/SearchResult?scope=text®ex=on&search=Joris%20*Ber
64.242.88.10 - - [07/Mar/2004:17:10:20 -0800] "GET /twiki/bin/oops/TWiki/TextFormattingRules?template=oopsmore¶m1=1.37¶m2=
64.242.88.10 - - [07/Mar/2004:17:13:50 -0800] "GET /twiki/bin/edit/TWiki/DefaultPlugin?t=1078688936 HTTP/1.1" 401 12846
64.242.88.10 - - [07/Mar/2004:17:16:00 -0800] "GET /twiki/bin/search/Main/?scope=topic®ex=on&search=^g HTTP/1.1" 200 3675
64.242.88.10 - - [07/Mar/2004:17:17:27 -0800] "GET /twiki/bin/search/TWiki/?scope=topic®ex=on&search=^d HTTP/1.1" 200 5773
lj1036.inktomisearch.com - - [07/Mar/2004:17:18:36 -0800] "GET /robots.txt HTTP/1.0" 200 68
lj1090.inktomisearch.com - - [07/Mar/2004:17:18:41 -0800] "GET /twiki/bin/view/Main/LondonOffice HTTP/1.0" 200 3860
64.242.88.10 - - [07/Mar/2004:17:21:44 -0800] "GET /twiki/bin/attach/TWiki/TablePlugin HTTP/1.1" 401 12846
64.242.88.10 - - [07/Mar/2004:17:22:49 -0800] "GET /twiki/bin/view/TWiki/ManagingWebs?rev=1.22 HTTP/1.1" 200 9310
64.242.88.10 - - [07/Mar/2004:17:23:54 -0800] "GET /twiki/bin/statistics/Main HTTP/1.1" 200 808
64.242.88.10 - - [07/Mar/2004:17:26:30 -0800] "GET /twiki/bin/view/TWiki/WikiCulture HTTP/1.1" 200 5935
64.242.88.10 - - [07/Mar/2004:17:27:37 -0800] "GET /twiki/bin/edit/Main/WebSearch?t=1078669682 HTTP/1.1" 401 12846
64.242.88.10 - - [07/Mar/2004:17:28:45 -0800] "GET /twiki/bin/oops/TWiki/ResetPassword?template=oopsmore¶m1=1.4¶m2=1.4 HTTP
64.242.88.10 - - [07/Mar/2004:17:29:59 -0800] "GET /twiki/bin/view/TWiki/ManagingWebs?skin=print HTTP/1.1" 200 8806
64.242.88.10 - - [07/Mar/2004:17:31:39 -0800] "GET /twiki/bin/edit/Main/UvscanAndPostFix?topicparent=Main.WebHome HTTP/1.1"
64.242.88.10 - - [07/Mar/2004:17:35:35 -0800] "GET /twiki/bin/view/TWiki/KlausWriessnegger HTTP/1.1" 200 3848
```

Can’t you just use a regular DB?

InfluxDB

order by time?

Doesn’t Scale

Example from metrics:

100 measurements per host *
10 hosts *
8640 per day (once every 10s) *
365 days

= 3,153,600,000 records per year

Have fun with that table…

But wait, we'll just keep the summaries!

InfluxDB

1h averages =

8,760,000 per year

Lose Detail and

AdHoc Queryability

So let's use Cassandra, HBase, or Scaleasaurus!

InfluxDB

Too much application code and complexity

Application logic and scripts to compute summaries

Application level logic for balancing

No data locality for AdHoc queries

And then there's more…

Web services

Libraries for web services

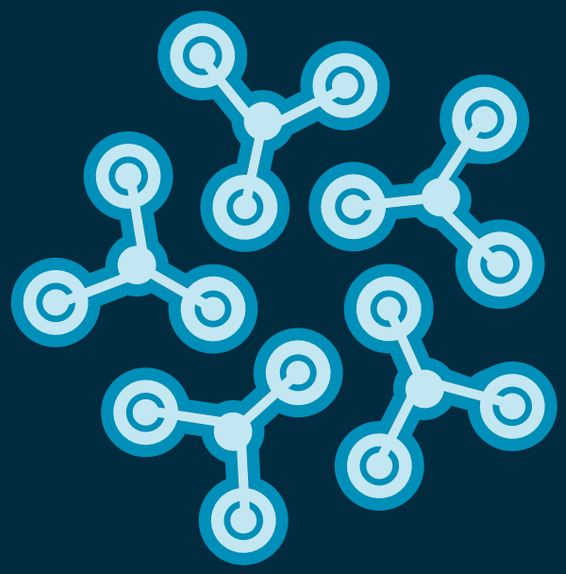
Data collection

Visualization

“Building an application with an analytics component today is like building a web application in 1998. You spend months building infrastructure before getting to the actual thing you want to build.”

–Paul Dix

Analytics should be about ***analyzing*** and ***interpreting*** data, not the infrastructure to store and process it.



InfluxDB

HTTP API

Web services built in

HTTP API (writes)

```
curl -X POST \
  'http://localhost:8086/db/mydb/series?u=paul&p=pass' \
-d '[{"name":"foo", "columns":["val"], "points": [[3]]}]'
```

Data (with timestamp)

```
[
   {
      "name": "cpu",
      "columns": ["time", "value", "host"],
      "points": [
         [1395168540, 56.7, "foo.influxdb.com"],
         [1395168540, 43.9, "bar.influxdb.com"]
      ]
   }
]
```

HTTP API (queries)

```
curl 'http://localhost:8086/db/mydb/series?u=paul&p=pass&q=.'
```

SQL-ish

```
select * from events
where time > now() - 1h
```

SQL-ish

```
select * from "series with weird chars ()*@#0982#$"
where time > now() - 1h
```

Where Regex

```
select line from application_logs
where line =~ /.*ERROR.*/ and
time > "2014-03-01" and time < "2014-03-03"
```

Only scans the time range

Series and time are the primary index

Work with many series…

Select from Regex

```
select * from /stats\.cpu\..*/
limit 1
```

Downsampling on the fly…

Aggregates

```
select percentile(90, value)
from response_times
group by time(10m)
where time > now() - 1d
```

Continuous

Downsampling…

Continuous queries (summaries)

```
select count(page_id) from events
group by time(1h), page_id
into events.[page_id]
```

Series per page id

```
select count from events.67
where time > now() - 7d
```

Continuous queries (regex downsampling)

```
select percentile(value, 90) as value
from /stats\.*/
group by time(5m)
into percentile.90.:series_name
```

Percentile series per host

```
select value
from percentile.90.stats.cpu.host1
where time > now() - 4h
```

Denormalization for performance

Range scans all user events for last hour

```
select * from events
where user_id = 3
and time > now() - 1h
```

Continuous queries (fan out)

```
select * from events
into events.[user_id]
```

Series per user id

```
select * from events.3
where time > now() - 1h
```

Distributed

Scale out, data locality, high availability

Raft for metadata

We owe Ben Johnson a beer or three…

Protobuf + TCP for queries, writes

Scalable

Have billions of points in 1 series* or a million different series

Libraries

Go, Ruby, Javascript, Python, Node.js, Clojure, Java,
Perl, Haskell, R, Scala, CLI (ruby and node)

Visualization

Built-in UI

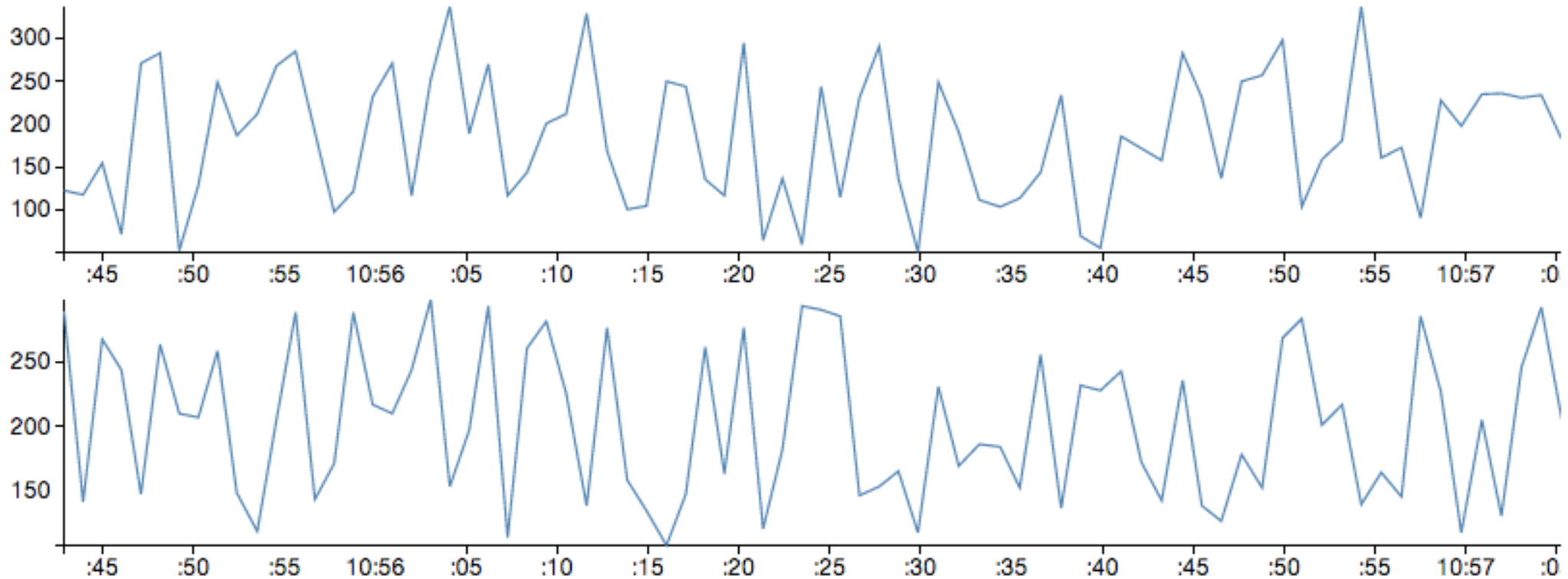
Read Points

Query

select foo, bar from data;

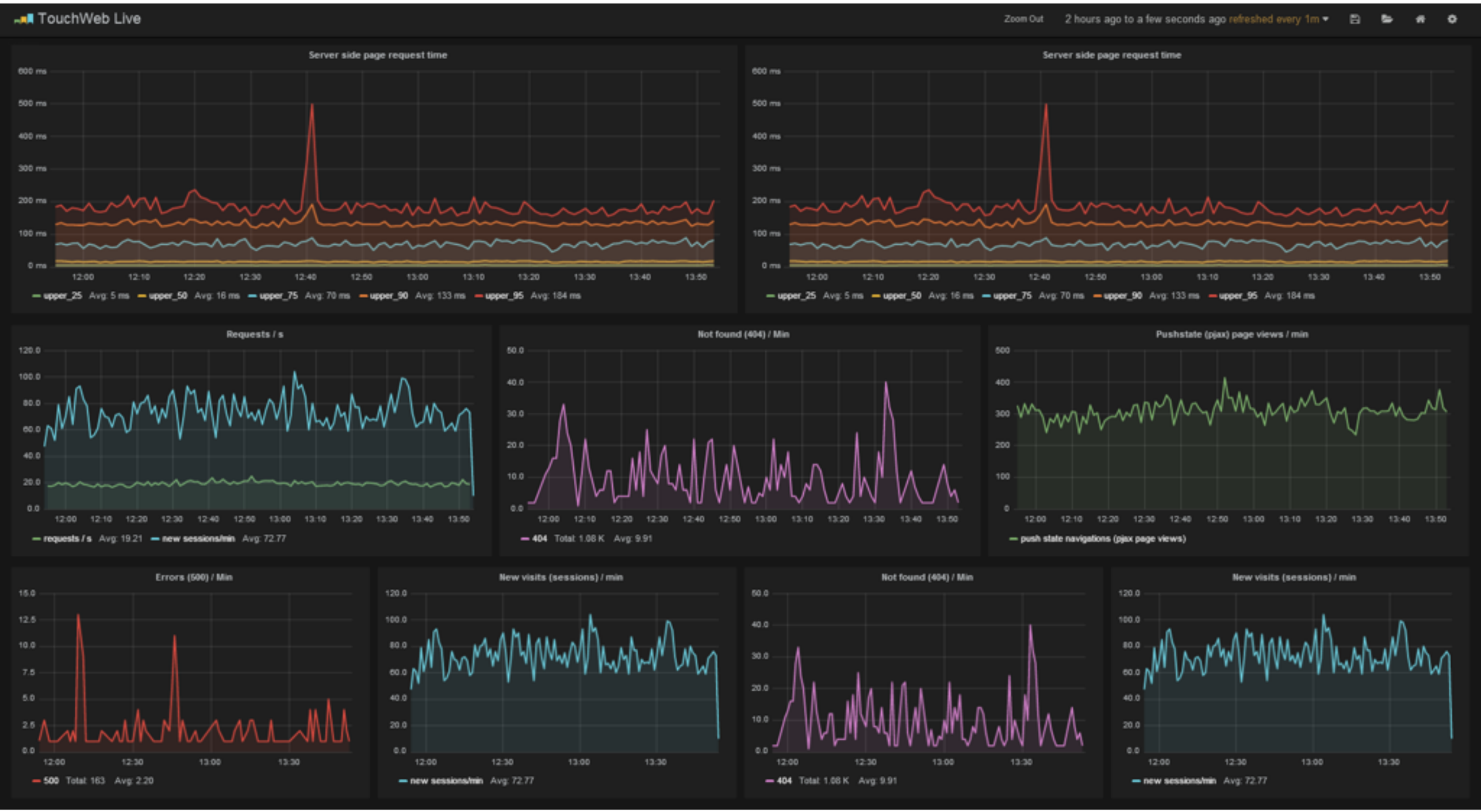
Execute Query

data



time	sequence_number	bar	foo
1384185425284	29696	182	205
1384185424149	29695	234	293
1384185423055	29694	231	246
1384185421955	29693	236	130
1384185420878	29692	235	205

Grafana


TouchWeb Live
Zoom Out
2 hours ago to a few seconds ago refreshed every 1m
Server side page request time
600 ms
500 ms
400 ms
300 ms
200 ms
100 ms
0 ms
12:00 12:10 12:20 12:30 12:40 12:50 13:00 13:10 13:20 13:30 13:40 13:50
upper_25 Avg: 5 ms upper_50 Avg: 16 ms upper_75 Avg: 70 ms upper_90 Avg: 133 ms upper_95 Avg: 184 ms
Server side page request time
600 ms
500 ms
400 ms
300 ms
200 ms
100 ms
0 ms
12:00 12:10 12:20 12:30 12:40 12:50 13:00 13:10 13:20 13:30 13:40 13:50
upper_25 Avg: 5 ms upper_50 Avg: 16 ms upper_75 Avg: 70 ms upper_90 Avg: 133 ms upper_95 Avg: 184 ms
Requests / s
120.0
100.0
80.0
60.0
40.0
20.0
0.0
12:00 12:10 12:20 12:30 12:40 12:50 13:00 13:10 13:20 13:30 13:40 13:50
requests / s Avg: 19.21 new sessions/min Avg: 72.77
Not found (404) / Min
50.0
40.0
30.0
20.0
10.0
0.0
12:00 12:10 12:20 12:30 12:40 12:50 13:00 13:10 13:20 13:30 13:40 13:50
404 Total: 1.08 K Avg: 9.91
Pushstate (pjax) page views / min
500
400
300
200
100
0
12:00 12:10 12:20 12:30 12:40 12:50 13:00 13:10 13:20 13:30 13:40 13:50
push state navigations (pjax page views)
Errors (500) / Min
15.0
12.5
10.0
7.5
5.0
2.5
0.0
12:00 12:30 13:00 13:30
500 Total: 163 Avg: 2.20
New visits (sessions) / min
120.0
100.0
80.0
60.0
40.0
20.0
0.0
12:00 12:30 13:00 13:30
new sessions/min Avg: 72.77
Not found (404) / Min
50.0
40.0
30.0
20.0
10.0
0.0
12:00 12:30 13:00 13:30
404 Total: 1.08 K Avg: 9.91
New visits (sessions) / min
120.0
100.0
80.0
60.0
40.0
20.0
0.0
12:00 12:30 13:00 13:30
new sessions/min Avg: 72.77

Javascript library + D3, HighCharts, Rickshaw, NVD3, etc.

Definitely more to do here!

Data Collection

CollectD Proxy, StatsD backend, Carbon ingestion,
OpenTSDB (soon)

Coming Soon

InfluxDB

ugh, Documentation

Series Metadata

Binary Protocol

Pubsub

```
select * from some_series
where host = “serverA”
into subscription()
```

```
select percentile(90, value) from some_series
group by time(1m)
into subscription()
```

Custom Functions

select myFunc(value) from some_series

Rack aware sharding and querying

Multi-datacenter replication

Push and bi-directional

Indexes?

Ponies?

Tell @jvshahid that you want your pony ;)

But it’s ready to go now.

Production deployments already running.

Need help?

support@influxdb.com

Thanks!

paul@influxdb.com

@pauldix