



@MongoDB @asya999 #askAsya

Diagnostics and Debugging

Asya Kamsky

Principal Solutions Architect, MongoDB



#MongoDB @asya999 #askAsya

Diagnostics and Debugging

Asya Kamsky

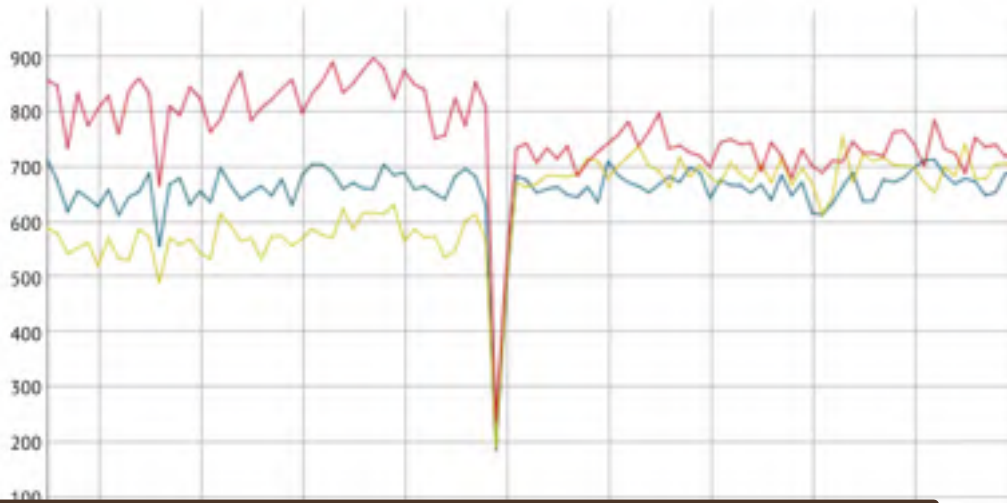
Principal Solutions Architect, MongoDB



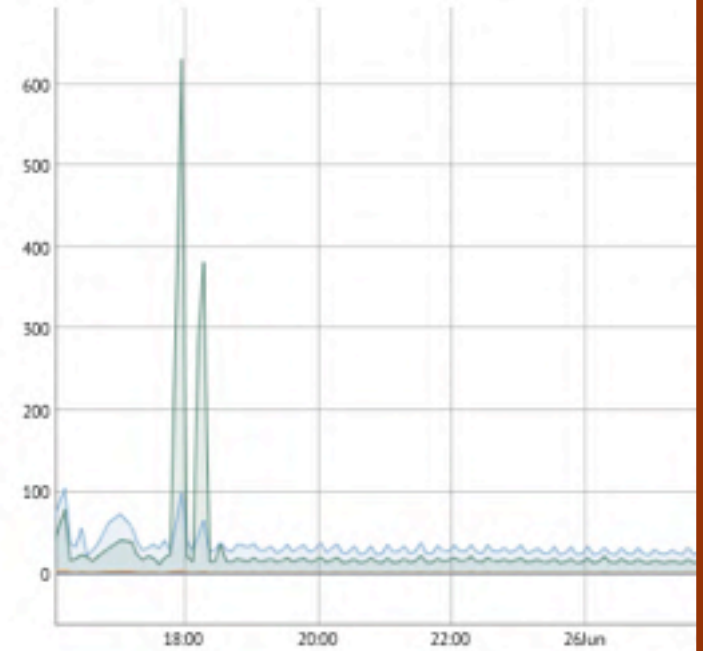


Shards Mongos

CHART insert



cpu time



Windows Internet Explorer

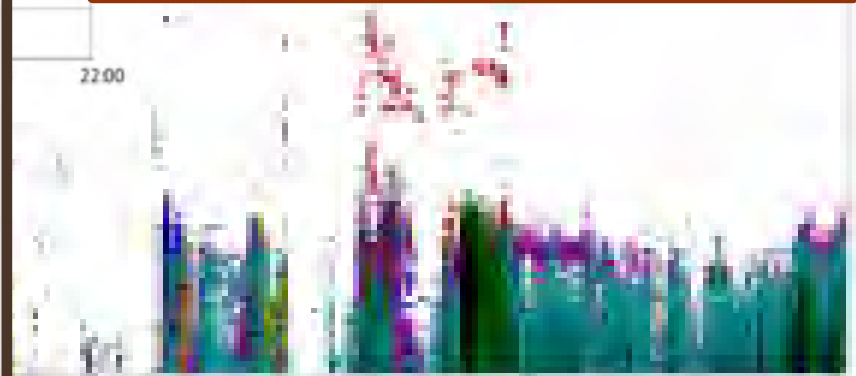


Stop running this script?

A script on this page is causing your web browser to run slowly. If it continues to run, your computer might become unresponsive.

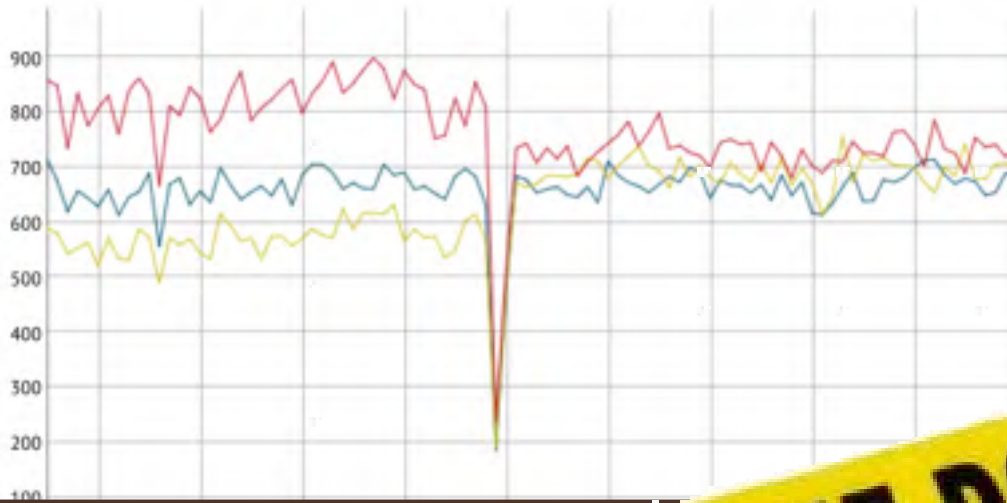
Yes

No



Shards Mongos

CHART insert



cpu time



CRIME SCENE DO NOT CROSS

Windows Internet Explorer

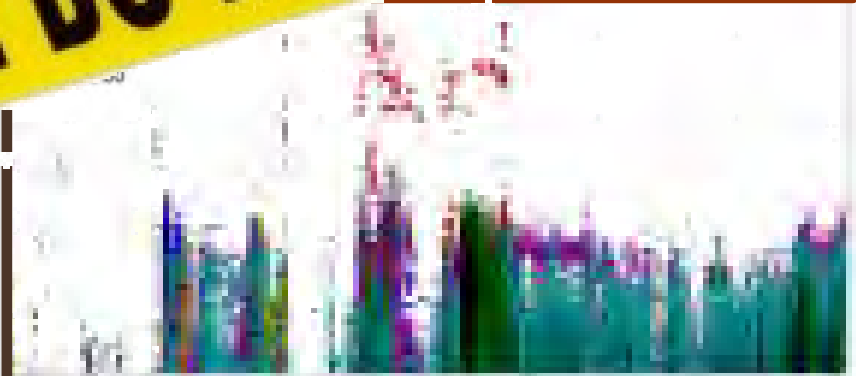


Str

...run slowly.
...become

Yes

No



Understanding Crime Scene

Understanding Crime Scene



“Data! Data! Data!

I can't make bricks without clay.”

-- Sherlock Holmes,

The Adventure of the Copper Beeches

Gathering Data



“It is a capital mistake to theorize before one has data. Insensibly one begins to twist facts to suit theories, instead of theories to suit facts.”

-- Sherlock Holmes, *A Scandal in Bohemia*



“There is nothing like first-hand evidence.”

-- Sherlock Holmes, *A Study in Scarlet*

Available Tools: OS

% iostat

% free

% blockdev --report

% dmesg

% ulimit -a

% ifconfig, ip <...>, iptables

% top, ...

Available Tools:

Available Tools: OS for MongoDB

% mongostat

% mongotop

Available Tools: MongoDB shell

% mongostat

% mongotop

> db.currentOp(), db.serverStatus()

> rs.status()

> sh.status()

Available Tools:

% mongostat

% mongotop

> db.currentOp(), db.serverStatus()

> rs.status()

> sh.status()



MongoDB Management Service (MMS)

Monitoring

Available Tools:



MongoDB Management Service (MMS)

Monitoring

Available Tools:



MongoDB Management Service (MMS) Monitoring

8 Shards

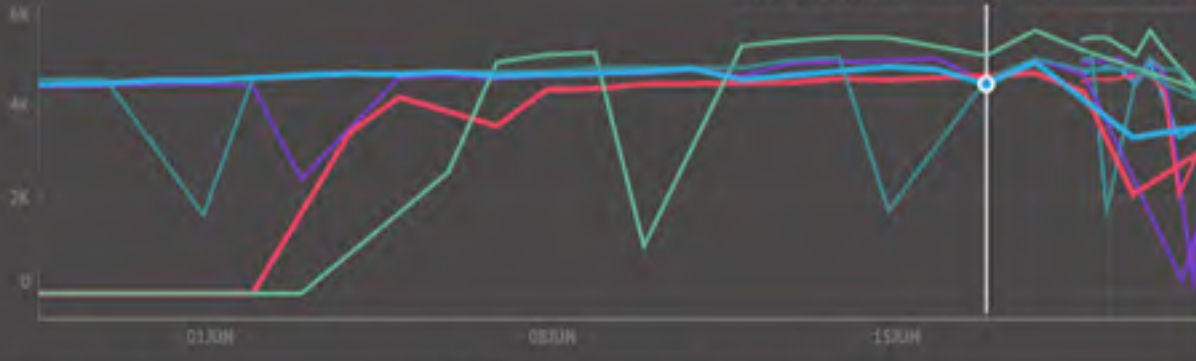
6 MongoS

3 Configs

1.32TB



2014/06/17 01:00: [mongo-08-2-7951982k-cc-27622-4.42K](#)



CHART

update

GRANULARITY

1 min 5 min 1 hr 1 day

ZOOM

1 wk 2 wk 1 mo 6 mo 1 yr

DISPLAY DATA

Individually Averaged Sum



SHARDS

SELECT

All None

ALERTS

DATA SIZE

MEMBERS

SELECT

Primaries Secondaries Both

Available Tools:



MongoDB Management Service (MMS) Monitoring



Available Tools:



MongoDB Management Service (MMS) Monitoring

8 Shards

6 MongoS

3 Configs

1.32TB



2014/06/22 05:48: 395-584 KiB, 3609722 27094: 4.05K



CHART

update

GRANULARITY

1 min 5 min 1 hr 1 day

ZOOM

1 hr 6 hr 12 hr 24 hr 48 hr

DISPLAY DATA

Individually Averaged Sum



SHARDS

SELECT

All

None

ALERTS

DATA SIZE

MEMBERS

SELECT

Primarys

Secondaries

Both

mongod logs

mongod logs

```
Tue Jul 30 19:15:27.898 [conn2] query
test.docs query: { query: { _id: { $gt:
ObjectId('51da44a8a4435e3d7174b696') }
}, orderby: { ts: 1.0 } } ntoreturn:10
ntoskip:0 nscanned:9235093
scanAndOrder:1 keyUpdates:0 numYields:5
locks(micros) r:6919962 nreturned:1
reslen:360 3578ms
```

mongod logs

```
Tue Jul 30 19:15:27.898 [conn2] query  
test.docs query: { query: { _id: { $gt:  
ObjectId('51da44a8a4435e3d7174b696') }  
}, orderby: { ts: 1.0 } } ntoreturn:10  
ntoskip:0 nscanned:9235093  
scanAndOrder:1 keyUpdates:0 numYields:5  
locks(micros) r:6919962 nreturned:10  
reslen:360 3578ms
```

mongod logs

date and time

thread

operation

namespace

Tue Jul 30 19:15:27.898 [conn2] query

test.docs query: { query: { _id: { \$gt: ObjectId('51da44a8a4435e3d7174b696') } }, orderby: { ts: 1.0 } } **ntoreturn:10**

n...
counters

ntoskip:0 nscanned:9235093

scanAndOrder:1 keyUpdates:0 **numYields:5**

locks (micros) r:6919962 nreturned:10

reslen:360 **3578ms**

number
of yields

lock
times

duration

"It is of the highest importance ... to be able to recognize, out of a number of facts, which are incidental and which vital. "

Sherlock Holmes, *The Reigate Puzzle*

How do you
find a **root cause**
in log files with
millions of lines?





<https://github.com/rueckstiess/mtools>

by Thomas Rückstieß

mtools

- mloginfo
- mlogfilter
- mplotqueries
- mlogvis
- mlaunch

mtools

- mloginfo
- mlogfilter
- mplotqueries

mloginfo

```
source: logs/mongodb.log.2014-05-04T15-52-59
  start: 2014 May 02 00:29:21
  end: 2014 May 04 15:52:59
date format: iso8601-local
length: 17091234
binary: mongod
version: 2.6.0
```

```
source: logs/mongodb.log.2014-05-04T15-53-06
  start: 2014 May 02 00:29:25
  end: 2014 May 04 15:53:06
date format: iso8601-local
length: 13696471
binary: mongod
version: 2.6.0
```

mplotqueries

```
$ mplotqueries --help  
usage: mplotqueries [OPTIONS] [logfile [logfile ...]]
```

A script to plot various information from logfiles. ...

mplotqueries

```
$ mplotqueries --help  
usage: mplotqueries [OPTIONS] [logfile [logfile ...]]
```

A script to plot various information from logfiles. ...

optional arguments:

```
--type {nscanned/n,rsstate,connchurn,durline,histogram,range,scatter,event}  
      type of plot (default=scatter with --yaxis duration)
```

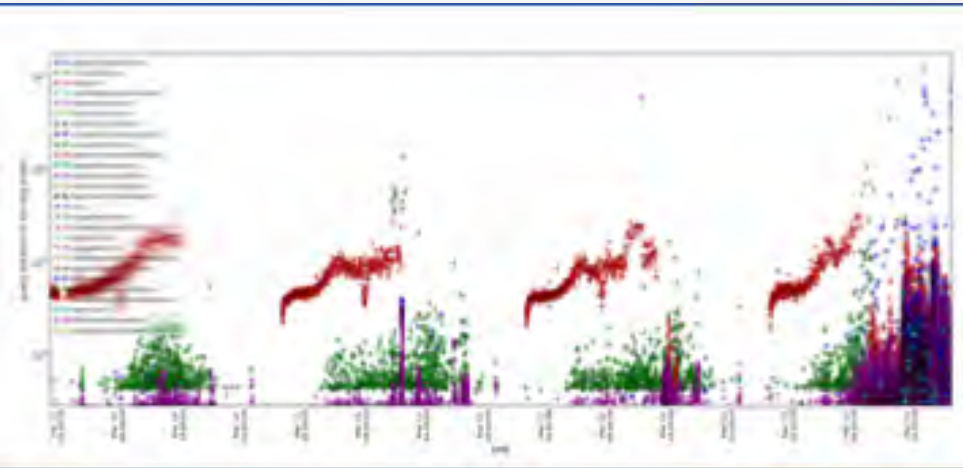
mplotqueries

```
$ mplotqueries --help
usage: mplotqueries [OPTIONS] [logfile [logfile ...]]
```

A script to plot various information from logfiles. ...

optional arguments:

```
--type {nscanned/n,rsstate,connchurn,durline,histogram,range,scatter,event}
      type of plot (default=scatter with --yaxis duration)
```



mplotqueries

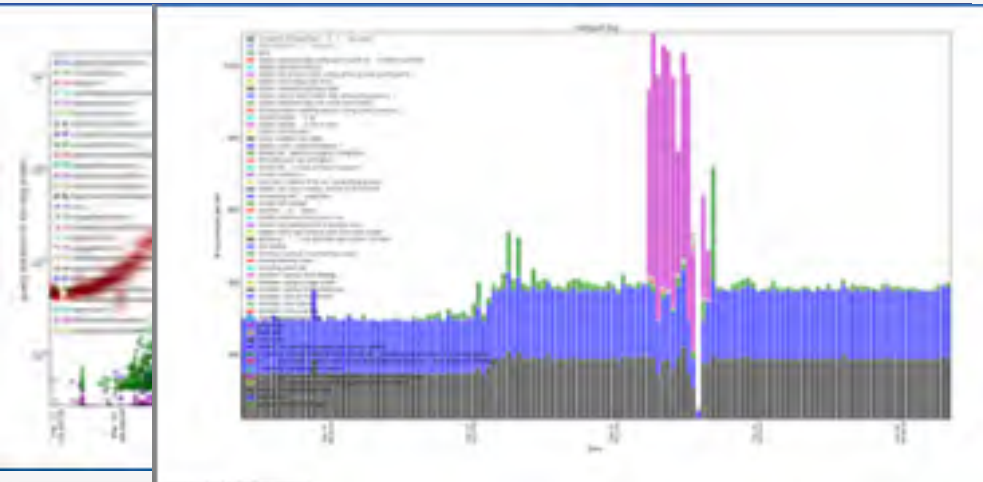
```
$ mplotqueries --help
```

```
usage: mplotqueries [OPTIONS] [logfile [logfile ...]]
```

A script to plot various information from logfiles. ...

optional arguments:

```
--type {nscanned/n,rsstate,connchurn,durline,histogram,range,scatter,event}  
      type of plot (default=scatter with --yaxis duration)
```



mplotqueries

```
$ mplotqueries --help
```

```
usage: mplotqueries [OPTIONS] [logfile [logfile ...]]
```

A script to plot various information from logfiles. ...

optional arguments:

```
--type {nscanned/n,rsstate,connchurn,durline,histogram,range,scatter,event}  
      type of plot (default=scatter with --yaxis duration)
```



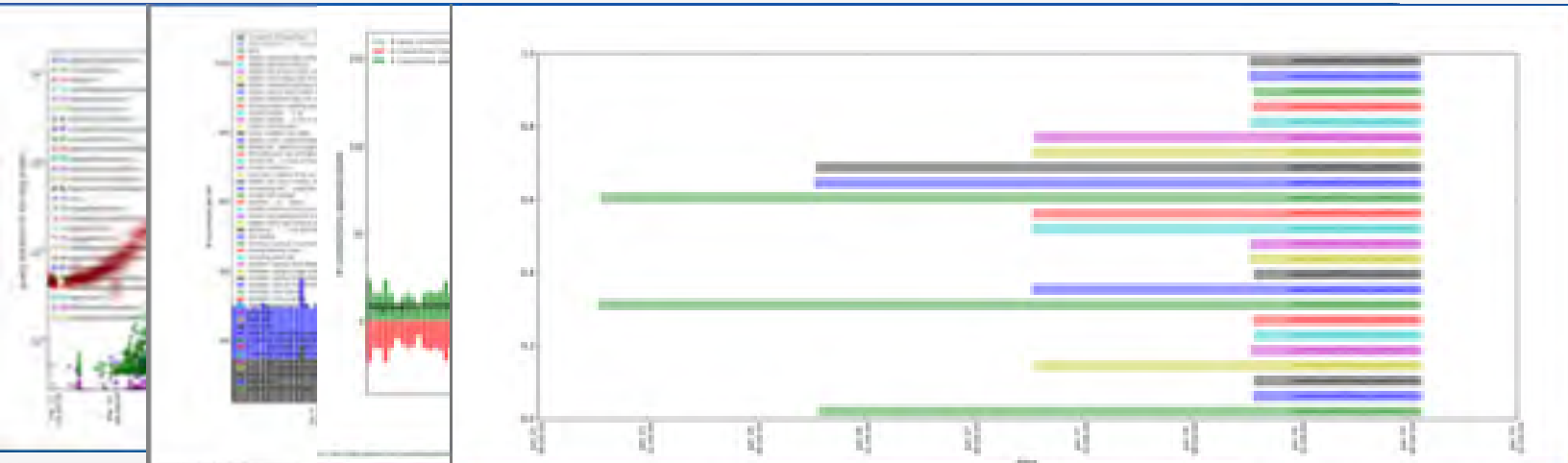
mplotqueries

```
$ mplotqueries --help
usage: mplotqueries [OPTIONS] [logfile [logfile ...]]
```

A script to plot various information from logfiles. ...

optional arguments:

```
--type {nscanned/n,rsstate,connchurn,durline,histogram,range,scatter,event}
      type of plot (default=scatter with --yaxis duration)
```



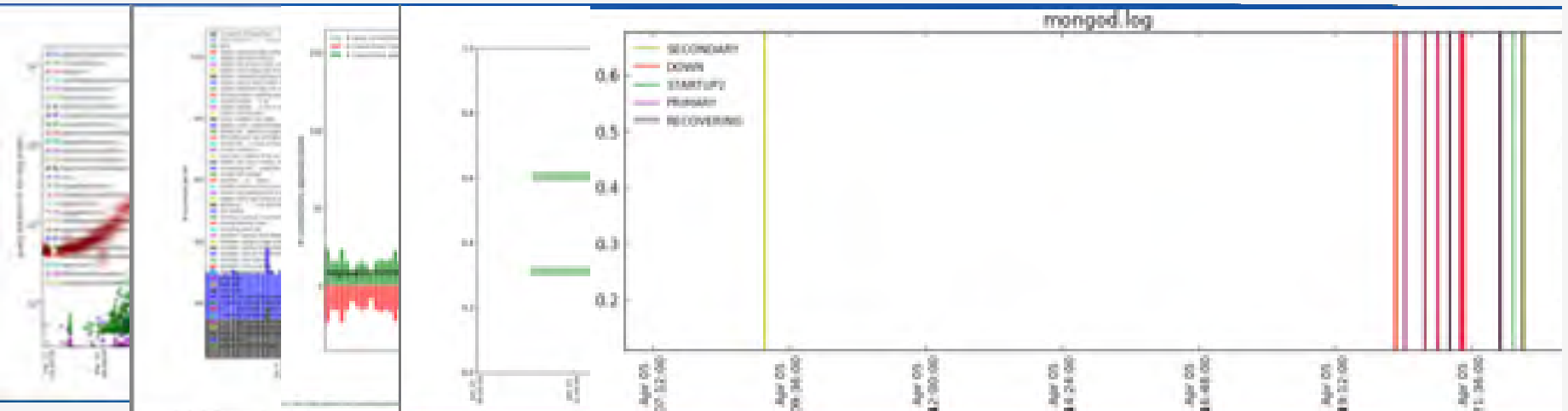
mplotqueries

```
$ mplotqueries --help
usage: mplotqueries [OPTIONS] [logfile [logfile ...]]
```

A script to plot various information from logfiles. ...

optional arguments:

```
--type {nscanned/n,rsstate,connchurn,durline,histogram,range,scatter,event}
      type of plot (default=scatter with --yaxis duration)
```



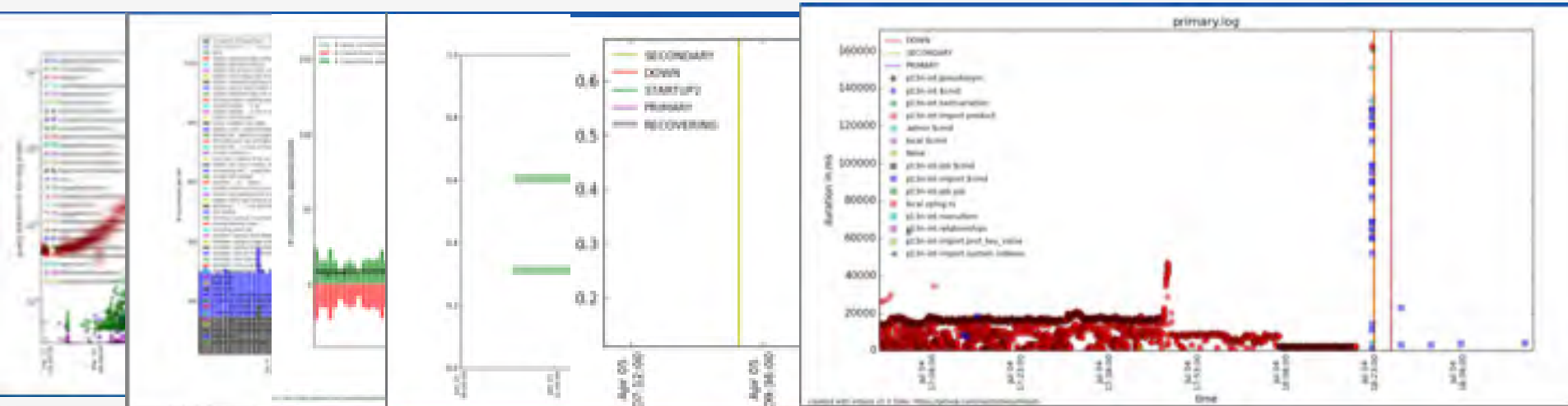
mplotqueries

```
$ mplotqueries --help
usage: mplotqueries [OPTIONS] [logfile [logfile ...]]
```

A script to plot various information from logfiles. ...

optional arguments:

```
--type {nscanned/n,rsstate,connchurn,durline,histogram,range,scatter,event}
      type of plot (default=scatter with --yaxis duration)
```



mplotqueries

```
$ mplotqueries --help
usage: mplotqueries [OPTIONS] [logfile [logfile ...]]
```

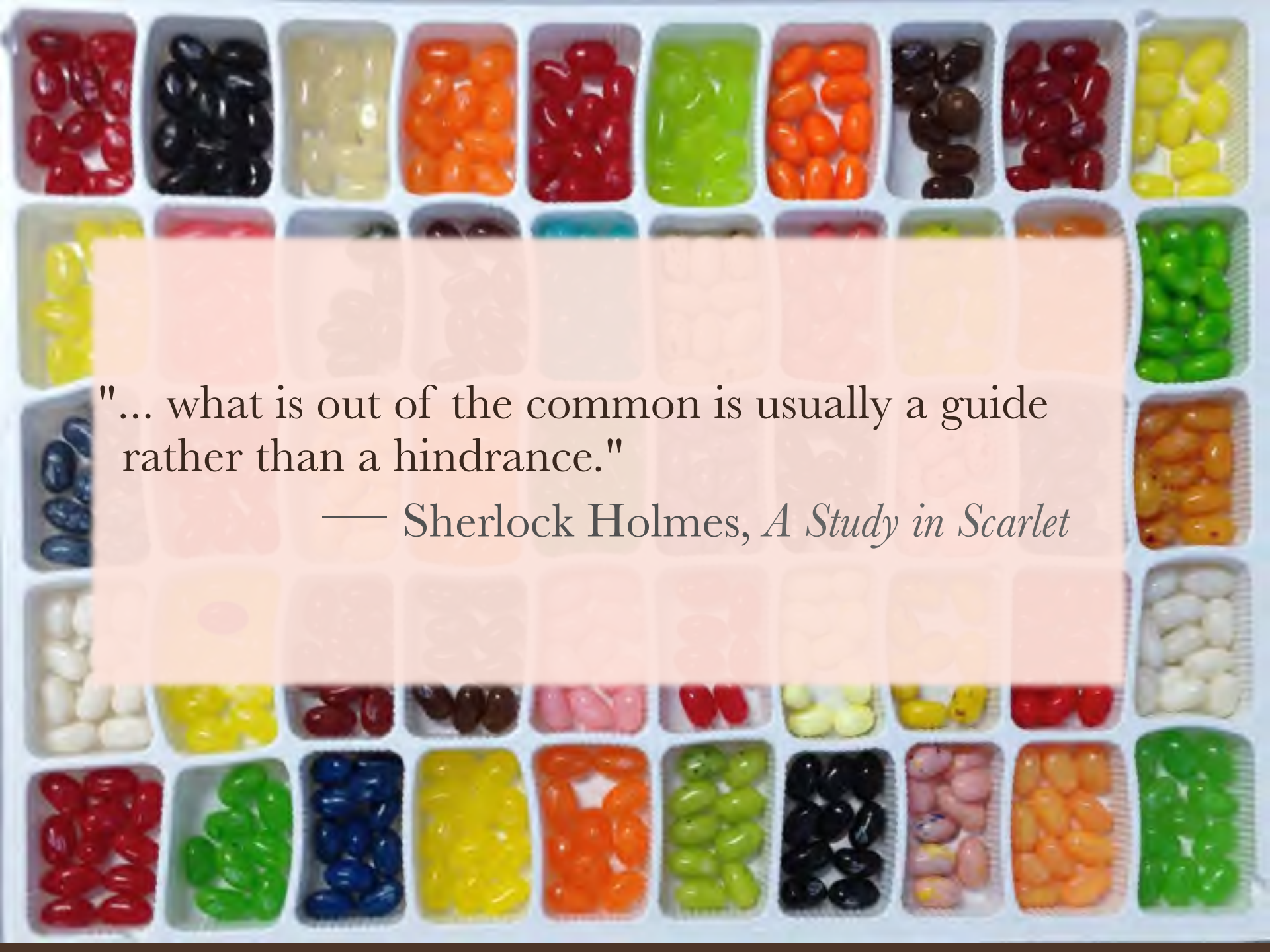
A script to plot various information from logfiles. ...

optional arguments:

--type {nscanned/n,rsstate,connchurn,durline,histogram,range,scatter,event}
type of plot (default=scatter with --yaxis duration)

--group GROUP specify value to group on.
All basic plot types can group on
'namespace', 'operation', 'thread', 'pattern' ...





"... what is out of the common is usually a guide rather than a hindrance."

— Sherlock Holmes, *A Study in Scarlet*

The Adventure of the Missing Three-Quarter

mloginfo

```
$ mloginfo firstmongo.log
  source: firstmongo.log
  start: 2014 Jun 10 11:19:54
  end: 2014 Jun 17 17:14:15
date format: iso8601-local
  length: 14170668
  binary: mongod
  version: 2.6.1
```

```
mloginfo firstmongo.log --queries
```

mloginfo

```
$ mloginfo firstmongo.log
  source: firstmongo.log
  start: 2014 Jun 10 11:19:54
  end: 2014 Jun 17 17:14:15
date format: iso8601-local
  length: 14170668
  binary: mongod
  version: 2.6.1
```

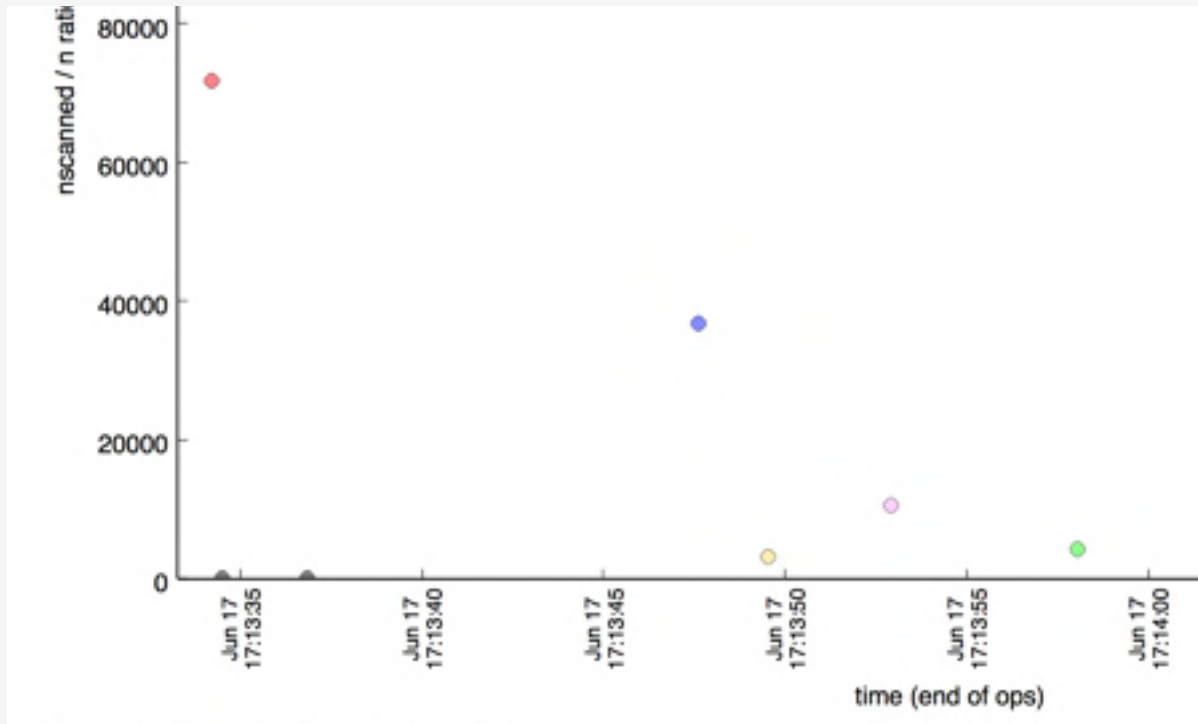
```
mloginfo firstmongo.log --queries
```

namespace	pattern	count	max (ms)	mean (ms)	95%-ile (ms)	sum (ms)
db1.request	{"requestSender": 1}	29	350161	93828	339441.2	2721018
db1.request	{"curRoute": 1,"isAdv": 1 }	169	9585	3942	6324.2	666302
db1.report	{"deliveryTime": 1}	9	111421	71536	106609.4	643829
db2.report	{"deliveryTime": 1}	10	85267	60363	81508.15	603634
db3.report	{"deliveryTime": 1}	10	72071	53062	70781.3	530624
db1.pending	{"sentTime": 1, "status": 1}	9	108734	38013	88470.4	342117

mplotqueries

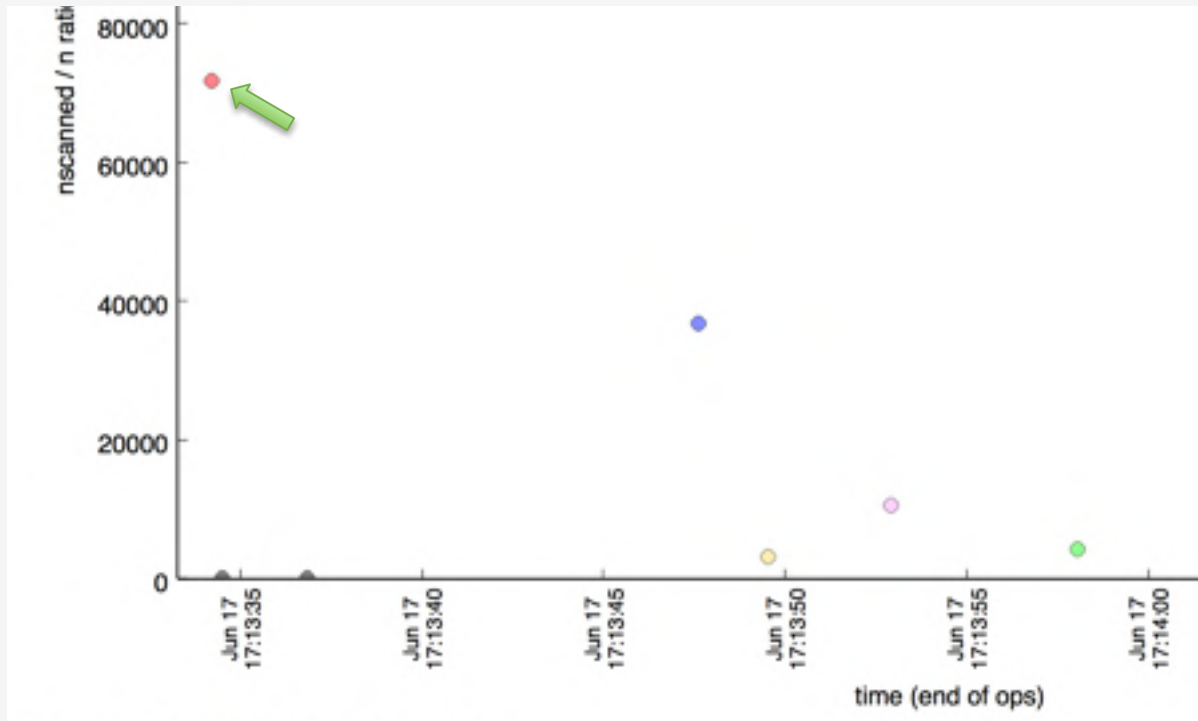
```
% mplotqueries firstmongo.log --type nscanned/n
```

mplotqueries



created with mtools v1.1.5: <https://github.com/ruckless/mtools>
% mplotqueries firstmongo.log --type nscanned/n

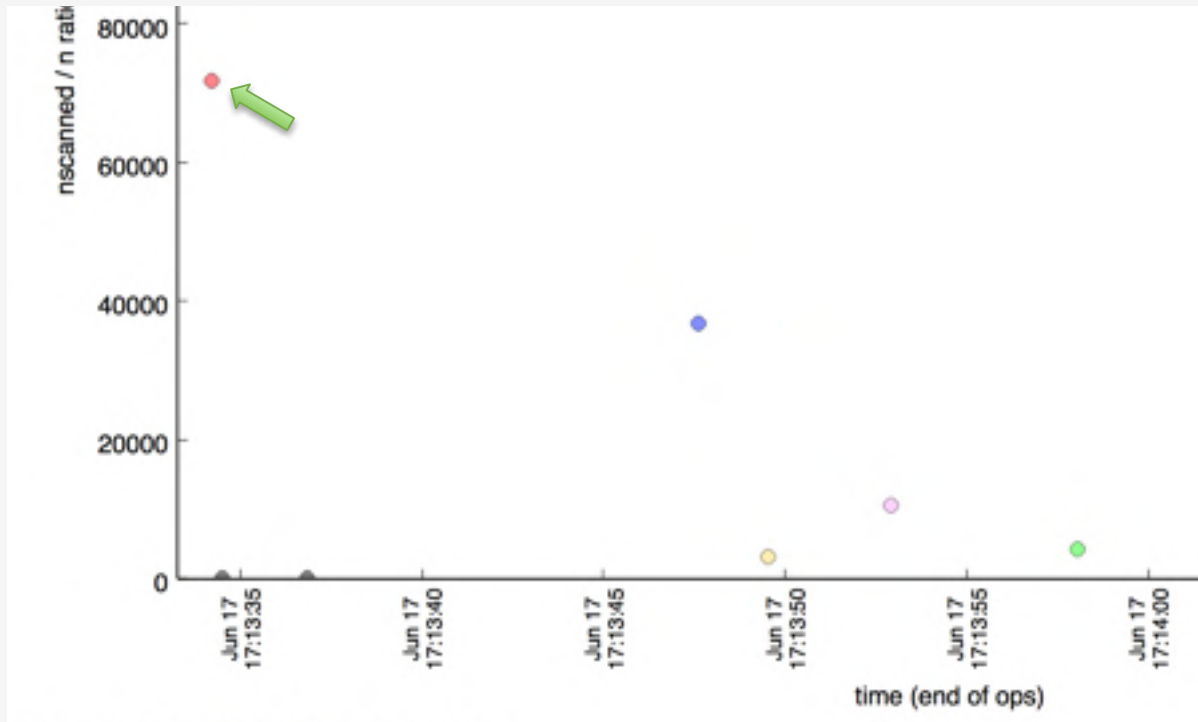
mplotqueries



`% mplotqueries firstmongo.log --type nscanned/n`

2014-06-17T17:13:34.235 [conn1569841] query db1.coll query: { time: { \$lt: "2014-06-17 17:13:31", \$gte: "2014-06-17 17:04:31" } } planSummary: COLLSCAN ntoreturn:0 ntoskip:0 **nscanned:5169727** nscannedObjects: 5169727 keyUpdates:0 numYields:12492 locks(micros) r:37736571 **nreturned:72** reslen:32707 30129ms

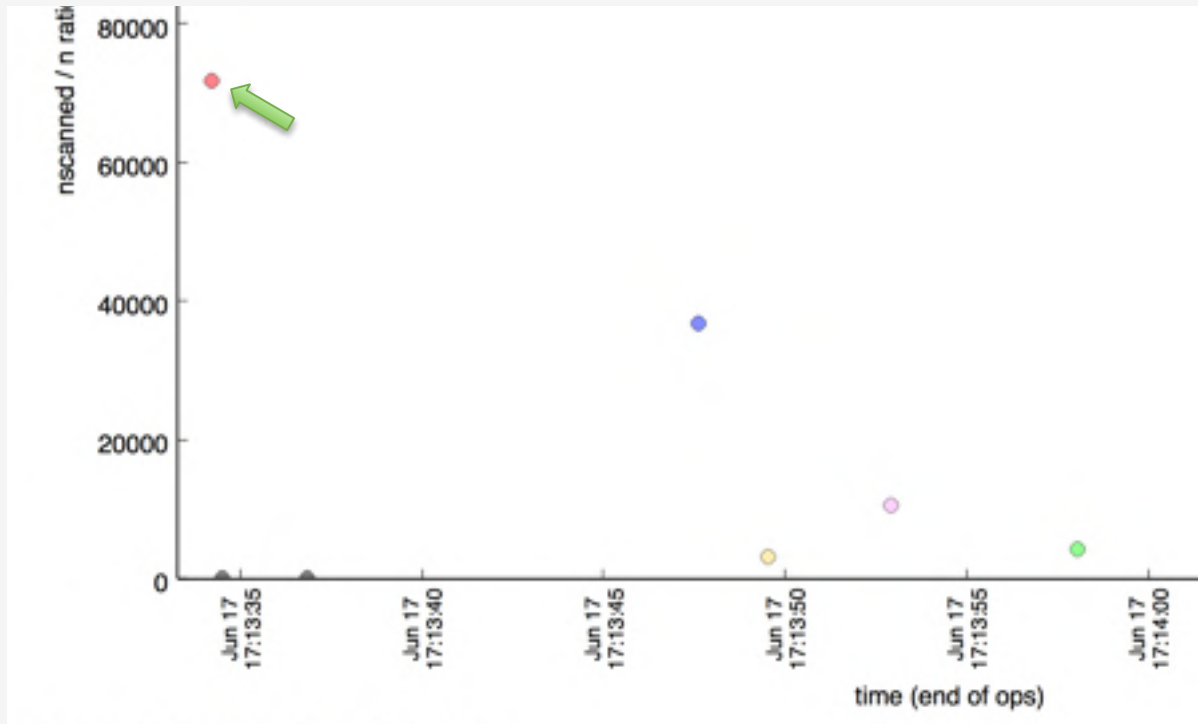
mplotqueries



`% mplotqueries firstmongo.log --type nscanned/n`

2014-06-17T17:13:34.235 [conn1569841] query db1.coll query: { time: { \$lt: "2014-06-17 17:13:31", \$gte: "2014-06-17 17:04:31" } } planSummary: COLLSCAN ntoreturn:0 ntoskip:0 **nscanned:5169727** nscannedObjects: 5169727 keyUpdates:0 numYields:12492 locks(micros) r:37736571 **nreturned:72** reslen:32707 resTime:129ms

mplotqueries

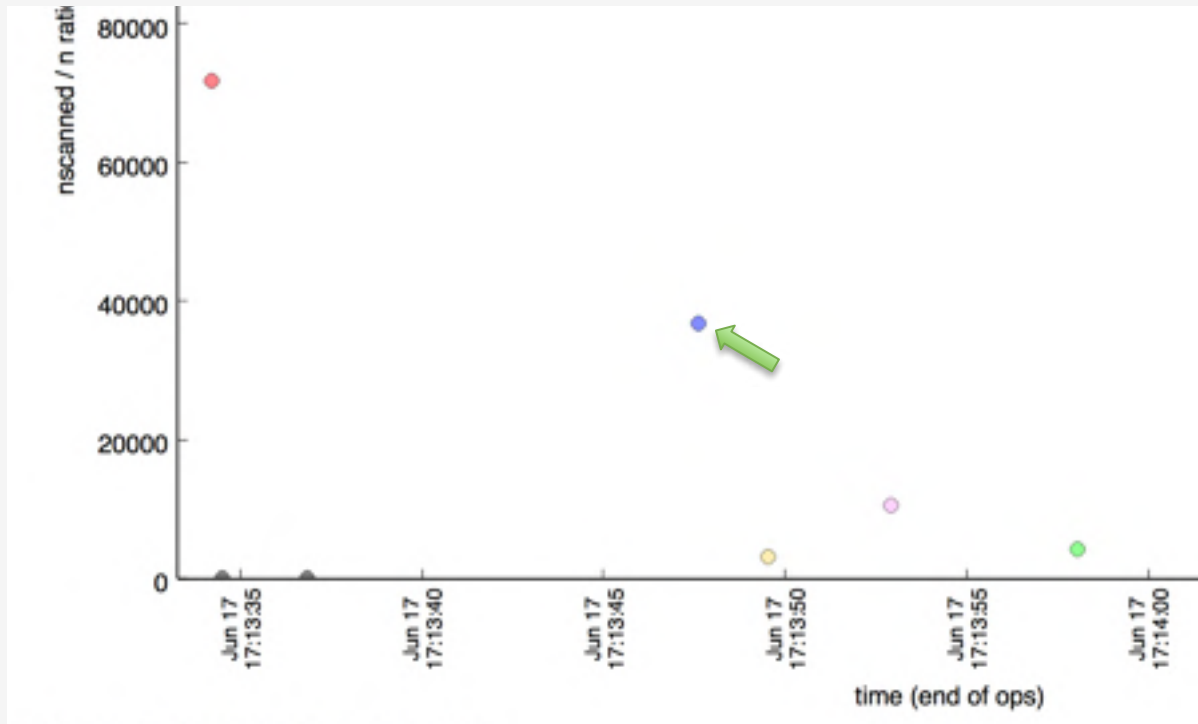


`% mplotqueries firstmongo.log --type nscanned/n`

2014-06-17T17:13:34.235 [conn1569841] query db1.coll query: { time: { \$lt: "2014-06-17 17:13:31", \$gte: "2014-06-17 17:04:31" } } planSummary: COLLSCAN ntoreturn:0 ntoskip:0 nscanned:5169727 nscannedObjects:5169727 keyUpdates:0 numYields:12492 locks(micros) r:37736571 nreturned:72 reslen:32707 30129ms



mplotqueries

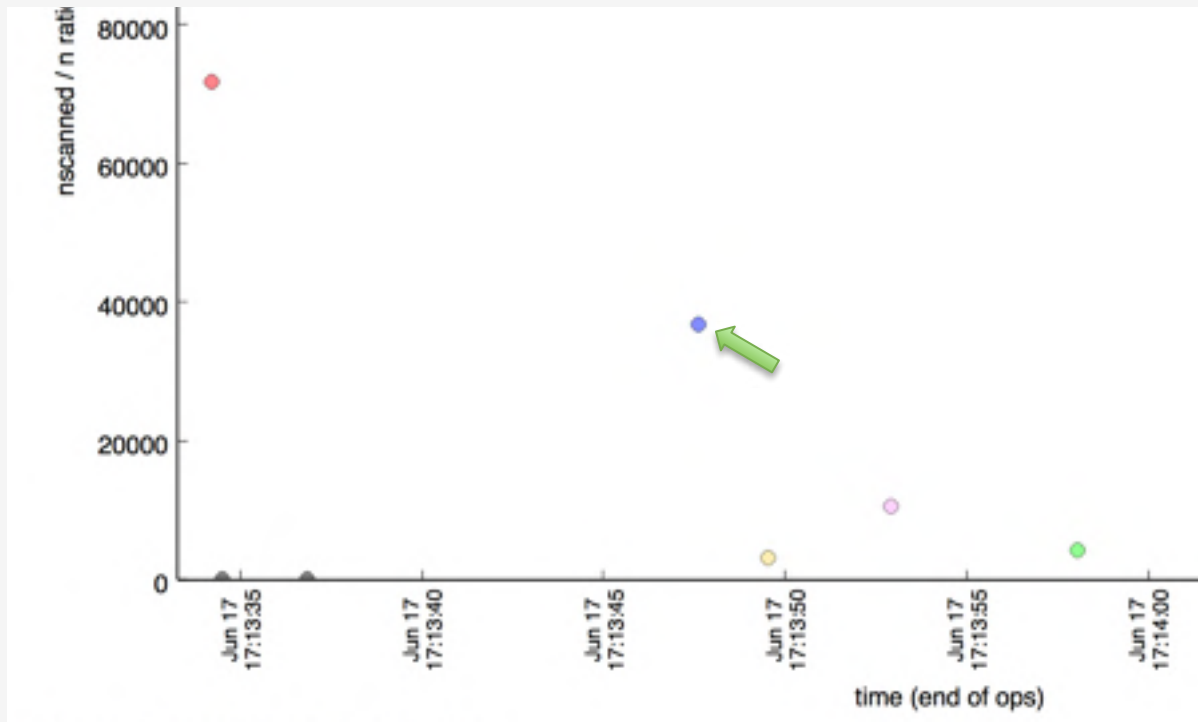


`% mplotqueries firstmongo.log --type nscanned/n`

2014-06-17T17:13:34.235 [conn1569841] query db1.coll query: { time: { \$lt: "2014-06-17 17:13:31", \$gte: "2014-06-17 17:04:31" } } planSummary: COLLSCAN ntoreturn:0 ntoskip:0 **nscanned:5169727** nscannedObjects: 5169727 keyUpdates:0 numYields:12492 locks(micros) r:37736571 **nreturned:72** reslen:32707 30129ms

2014-06-17T17:13:47.607 [conn1569990] query db2.coll query: { time: { \$lt: "2014-06-17 17:14:05", \$gte: "2014-06-17 17:05:05" }, status: 8 } planSummary: COLLSCAN ntoreturn:0 ntoskip:0 **nscanned:2057564** nscannedObjects:2057564 keyUpdates:0 numYields:5008 locks(micros) r:11557172 **nreturned:56** reslen:18745 13086ms

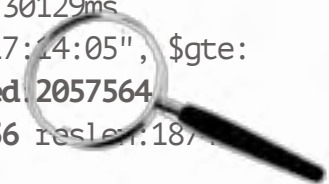
mplotqueries



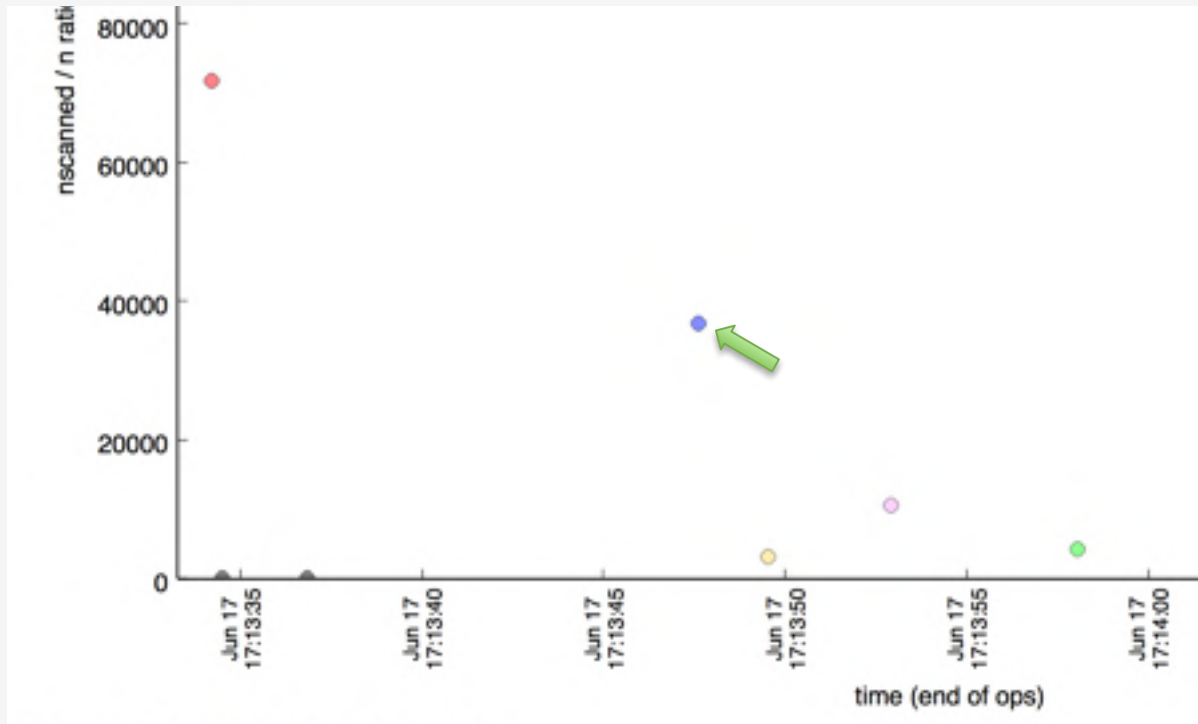
`% mplotqueries firstmongo.log --type nscanned/n`

2014-06-17T17:13:34.235 [conn1569841] query db1.coll query: { time: { \$lt: "2014-06-17 17:13:31", \$gte: "2014-06-17 17:04:31" } } planSummary: COLLSCAN ntoreturn:0 ntoskip:0 **nscanned:5169727** nscannedObjects:5169727 keyUpdates:0 numYields:12492 locks(micros) r:37736571 **nreturned:72** reslen:32707 30129ms

2014-06-17T17:13:47.607 [conn1569990] query db2.coll query: { time: { \$lt: "2014-06-17 17:14:05", \$gte: "2014-06-17 17:05:05" }, status: 8 } planSummary: COLLSCAN ntoreturn:0 ntoskip:0 **nscanned:2057564** nscannedObjects:2057564 keyUpdates:0 numYields:5008 locks(micros) r:11557172 **nreturned:56** reslen:187 13086ms



mplotqueries



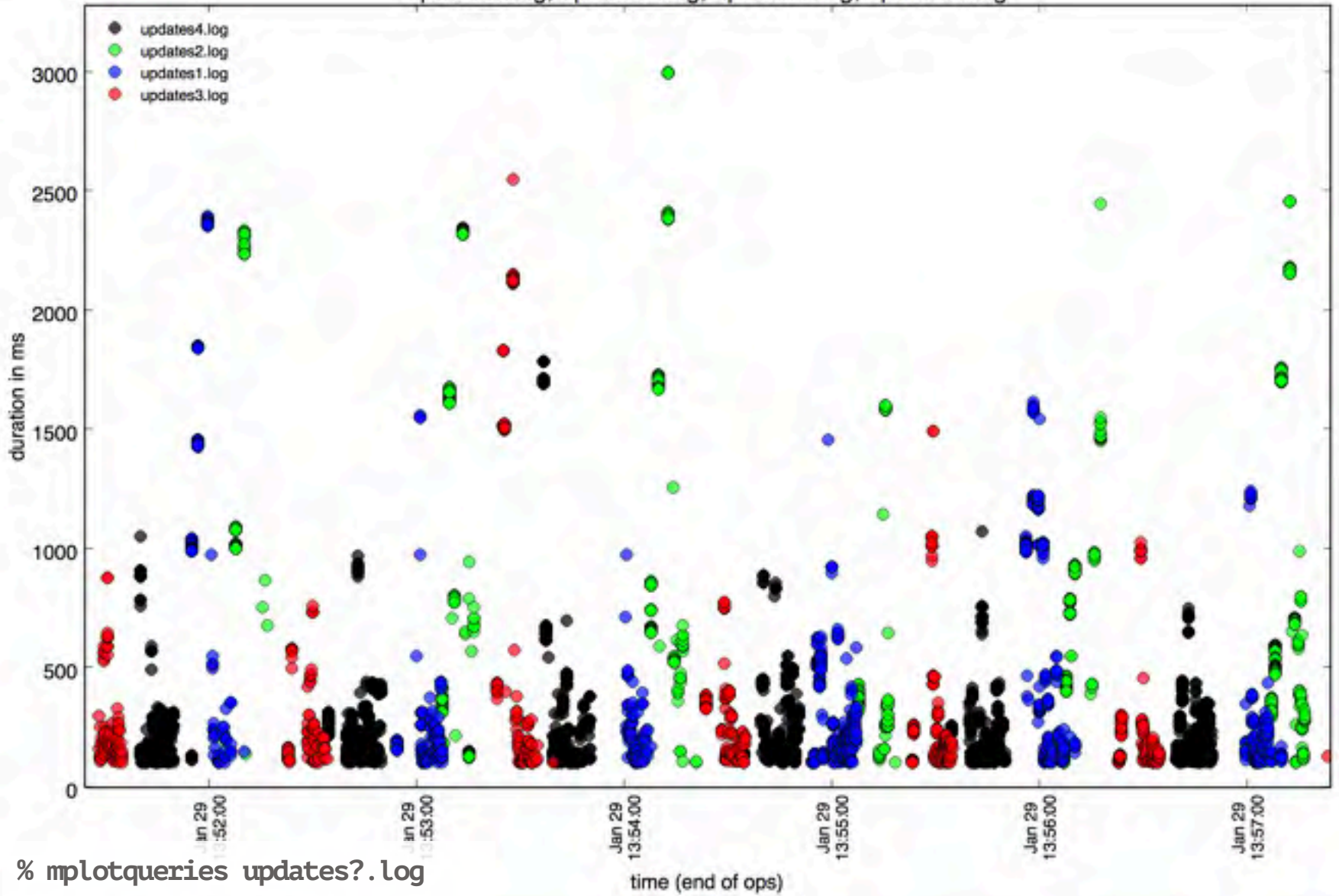
`% mplotqueries firstmongo.log --type nscanned/n`

2014-06-17T17:13:34.235 [conn1569841] query db1.coll query: { time: { \$lt: "2014-06-17 17:13:31", \$gte: "2014-06-17 17:04:31" } } planSummary: COLLSCAN ntoreturn:0 ntoskip:0 **nscanned:5169727** nscannedObjects: 5169727 keyUpdates:0 numYields:12492 locks(micros) r:37736571 **nreturned:72** reslen:32707 30129ms
2014-06-17T17:13:47.607 [conn1569990] query db2.coll query: { time: { \$lt: "2014-06-17 17:14:05", \$gte: "2014-06-17 17:05:05" }, status: 8 } planSummary: COLLSCAN ntoreturn:0 ntoskip:0 **nscanned:2057564** nscannedObjects:2057564 keyUpdates:0 numYields:5008 locks(micros) r:11557172 **nreturned:56** reslen:18745 13086ms

The Sign of Four



updates1.log, updates2.log, updates3.log, updates4.log

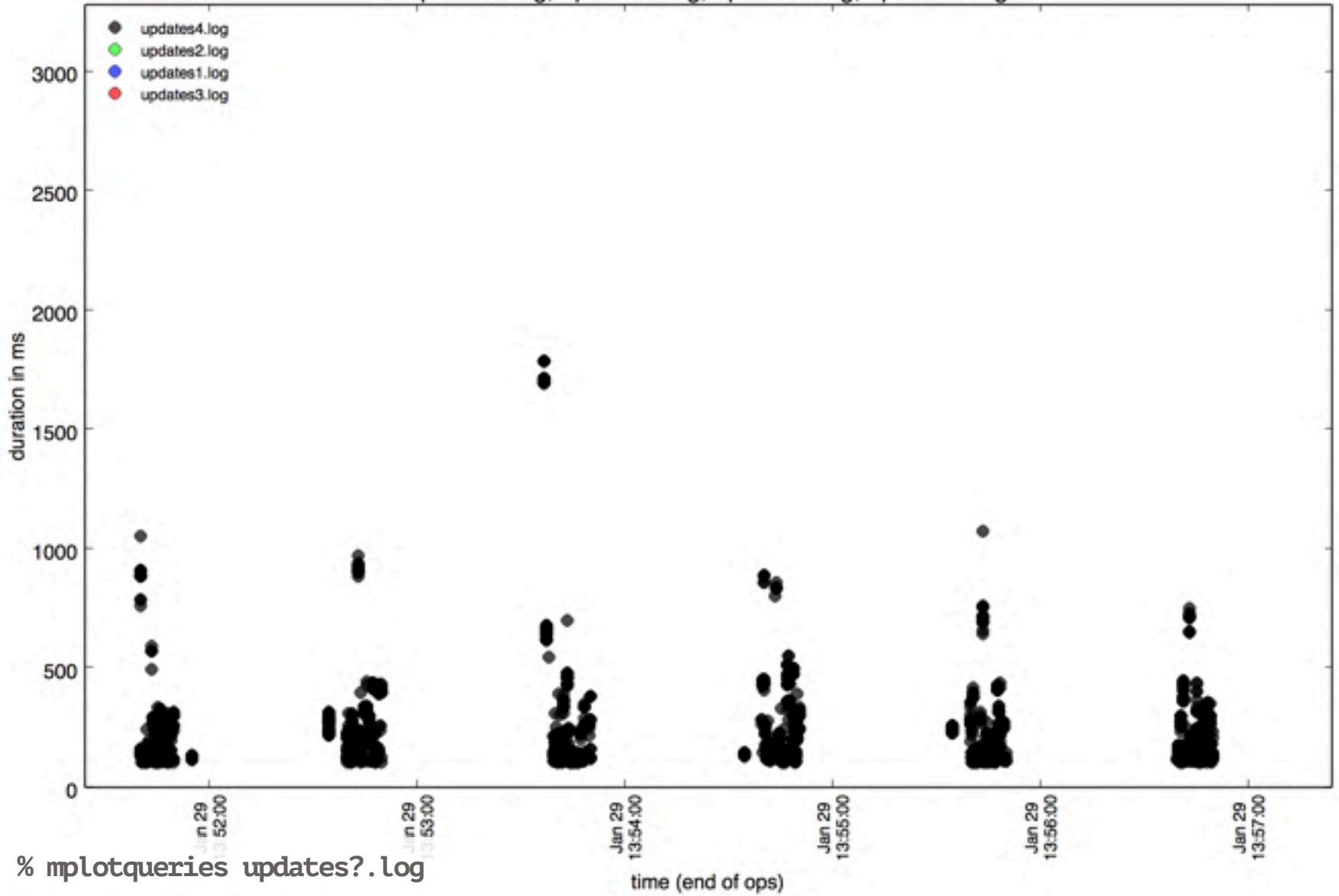


% mplotqueries updates?.log

time (end of ops)

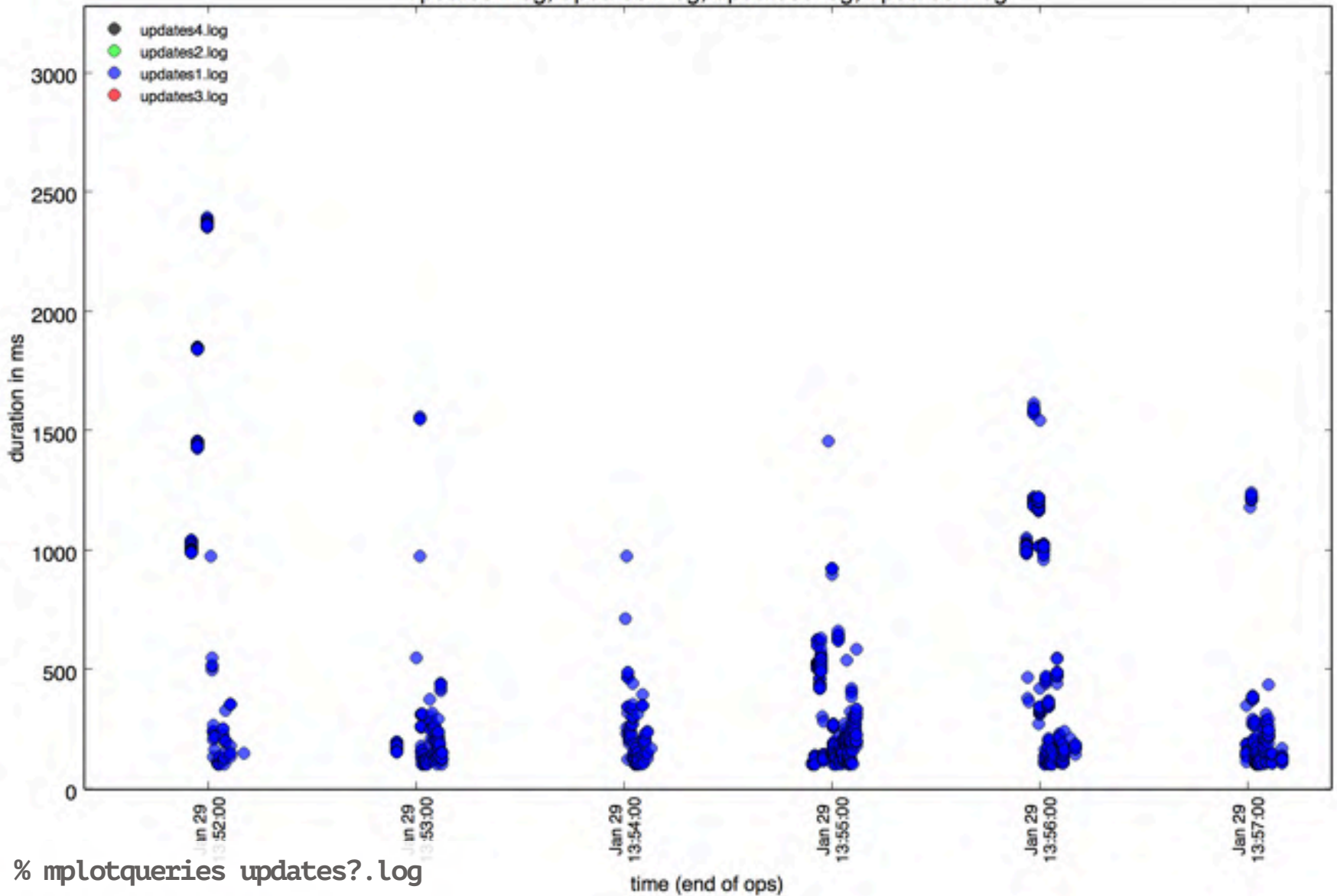
created with mtools v1.1.5: <https://github.com/hueckstless/mtools>

updates1.log, updates2.log, updates3.log, updates4.log



created with mtools v1.1.5: <https://github.com/hueckstless/mtools>

updates1.log, updates2.log, updates3.log, updates4.log

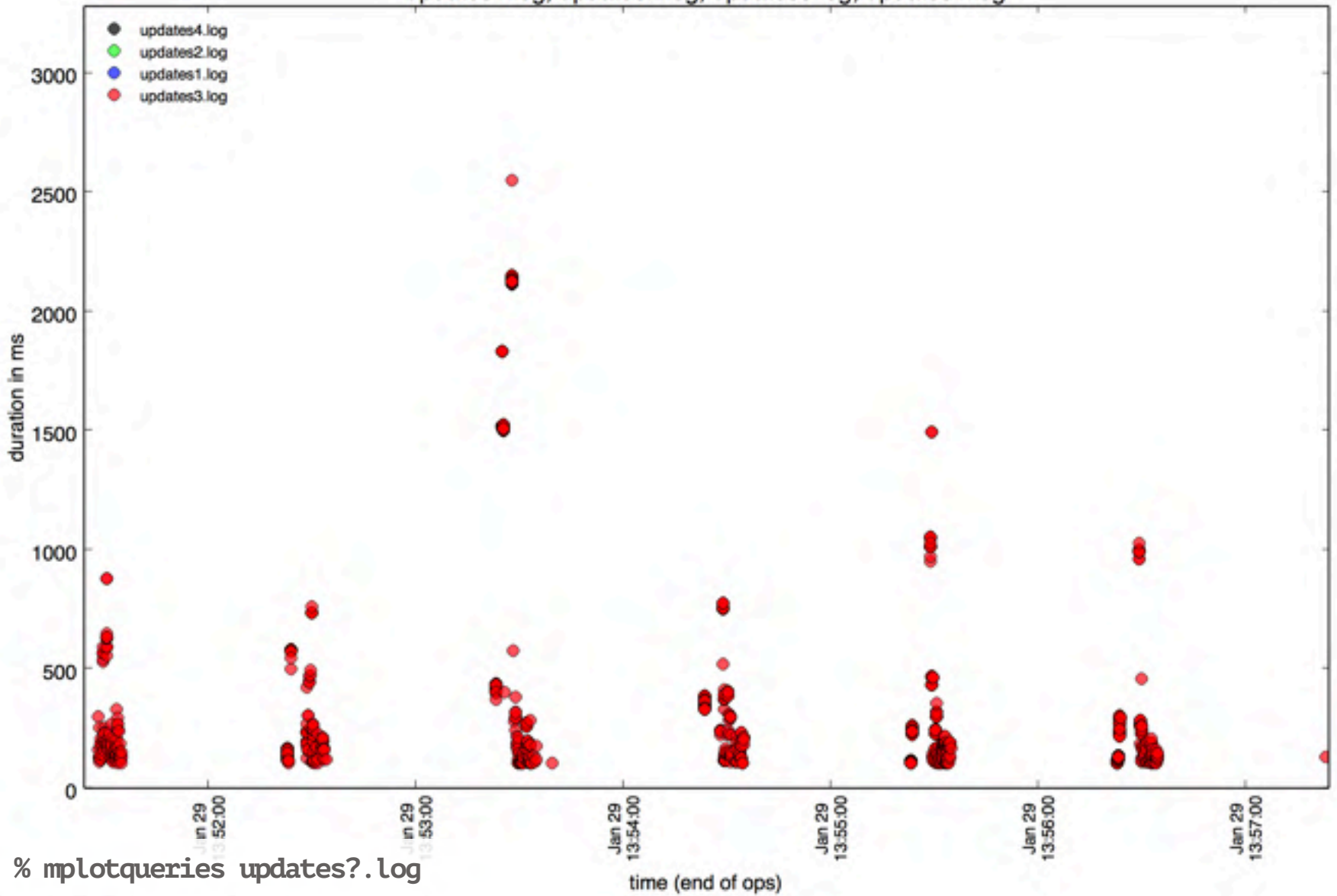


% mplotqueries updates?.log

time (end of ops)

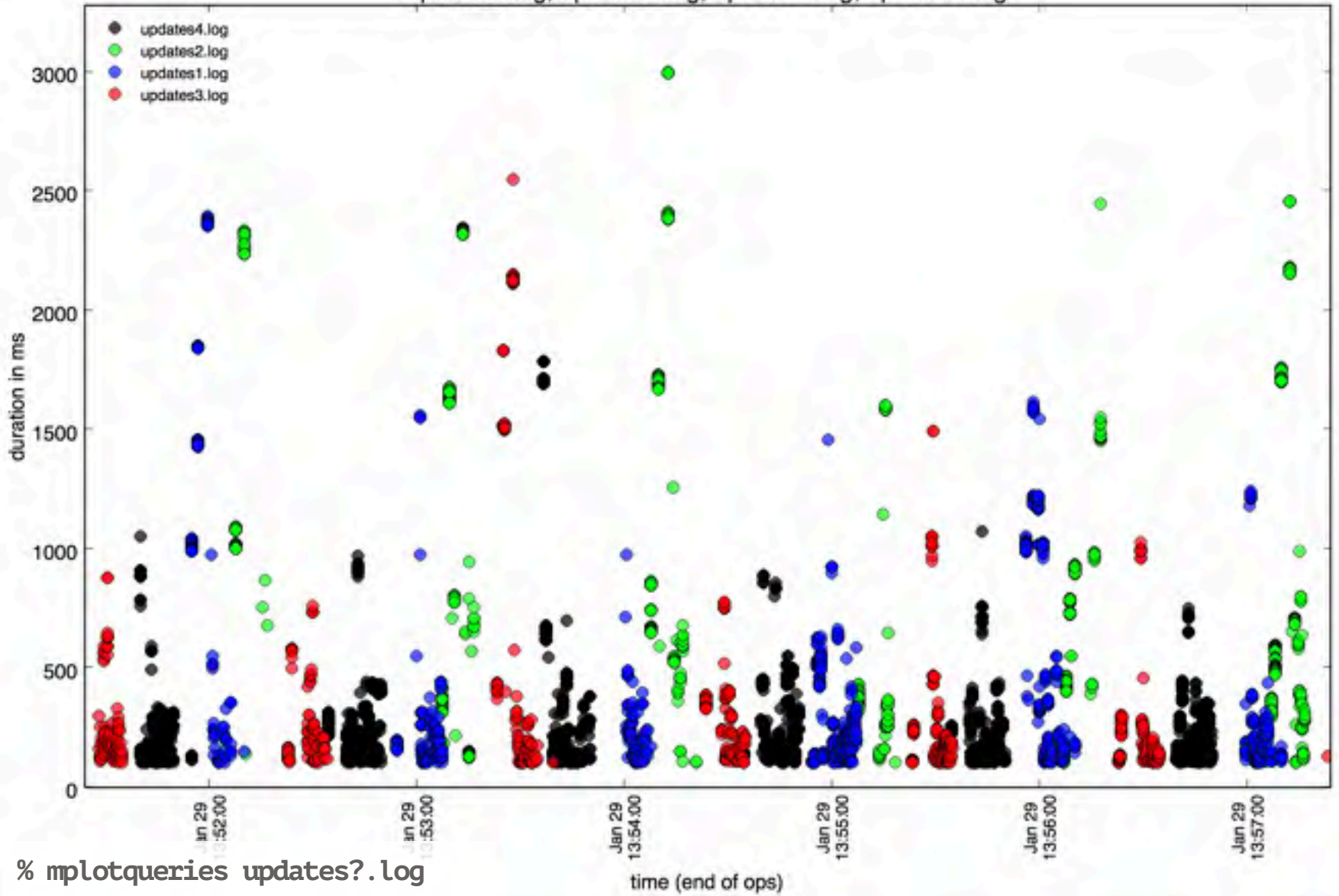
created with mtools v1.1.5: <https://github.com/huecksties/mtools>

updates1.log, updates2.log, updates3.log, updates4.log



% mplotqueries updates?.log

updates1.log, updates2.log, updates3.log, updates4.log



% mplotqueries updates?.log

time (end of ops)

created with mtools v1.1.5: <https://github.com/rueckstless/mtools>

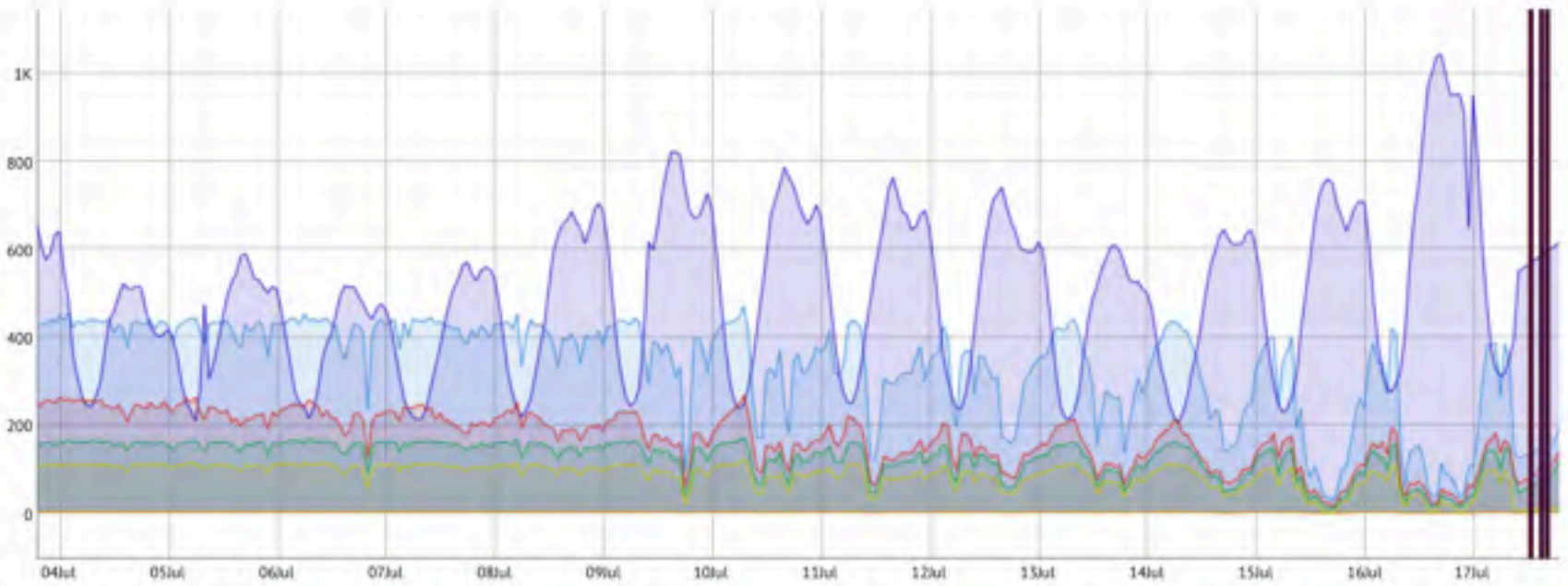
"Eliminate all other factors, and the one which remains must be the truth."

Sherlock Holmes - *The Sign of Four*

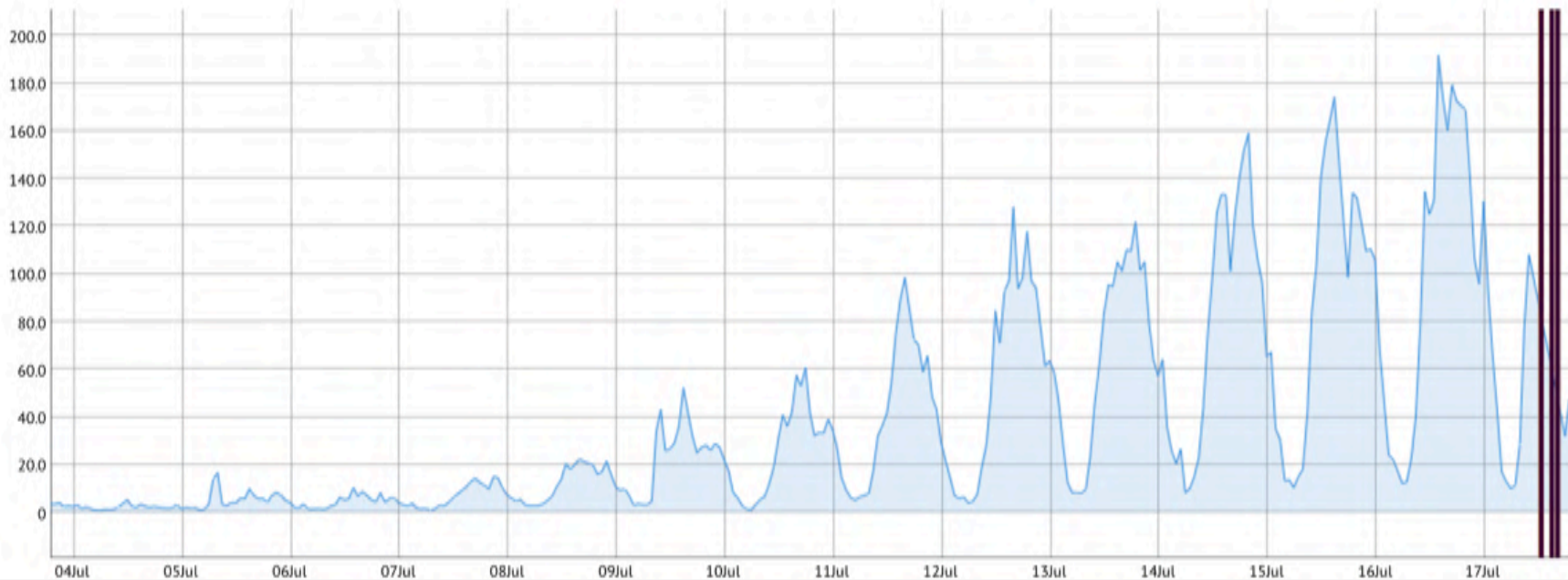
The Case of Identity



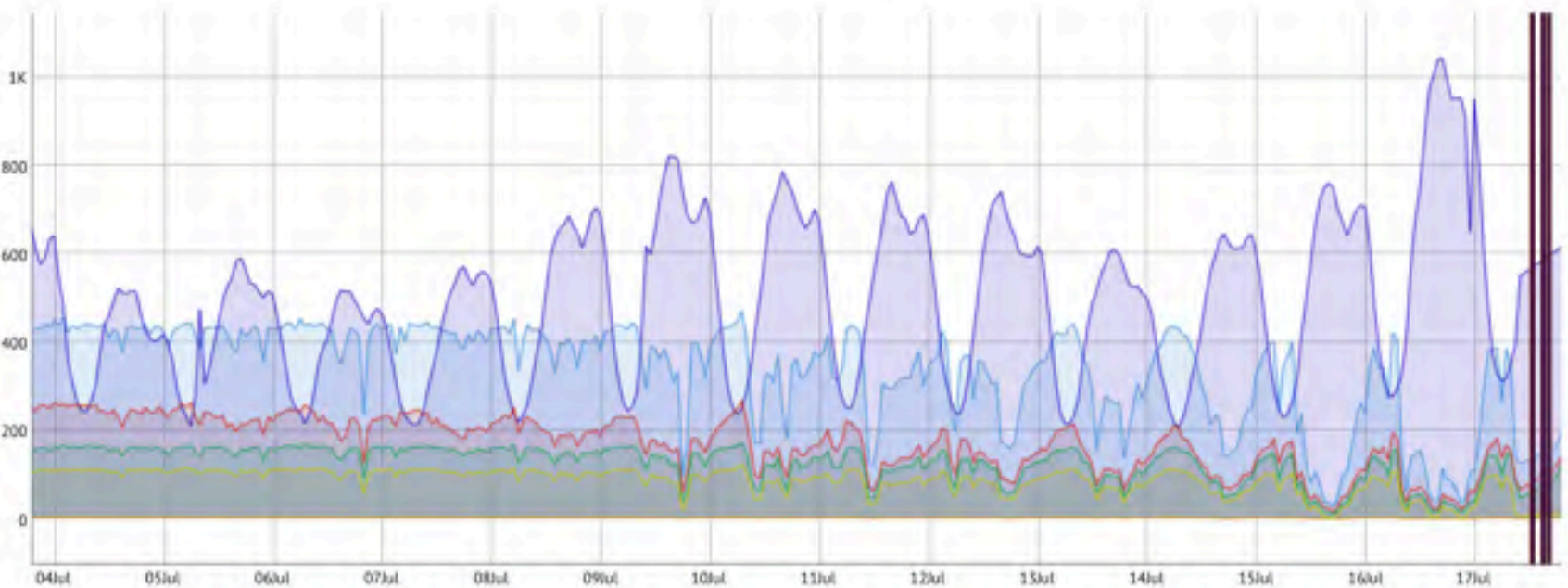
opcounters

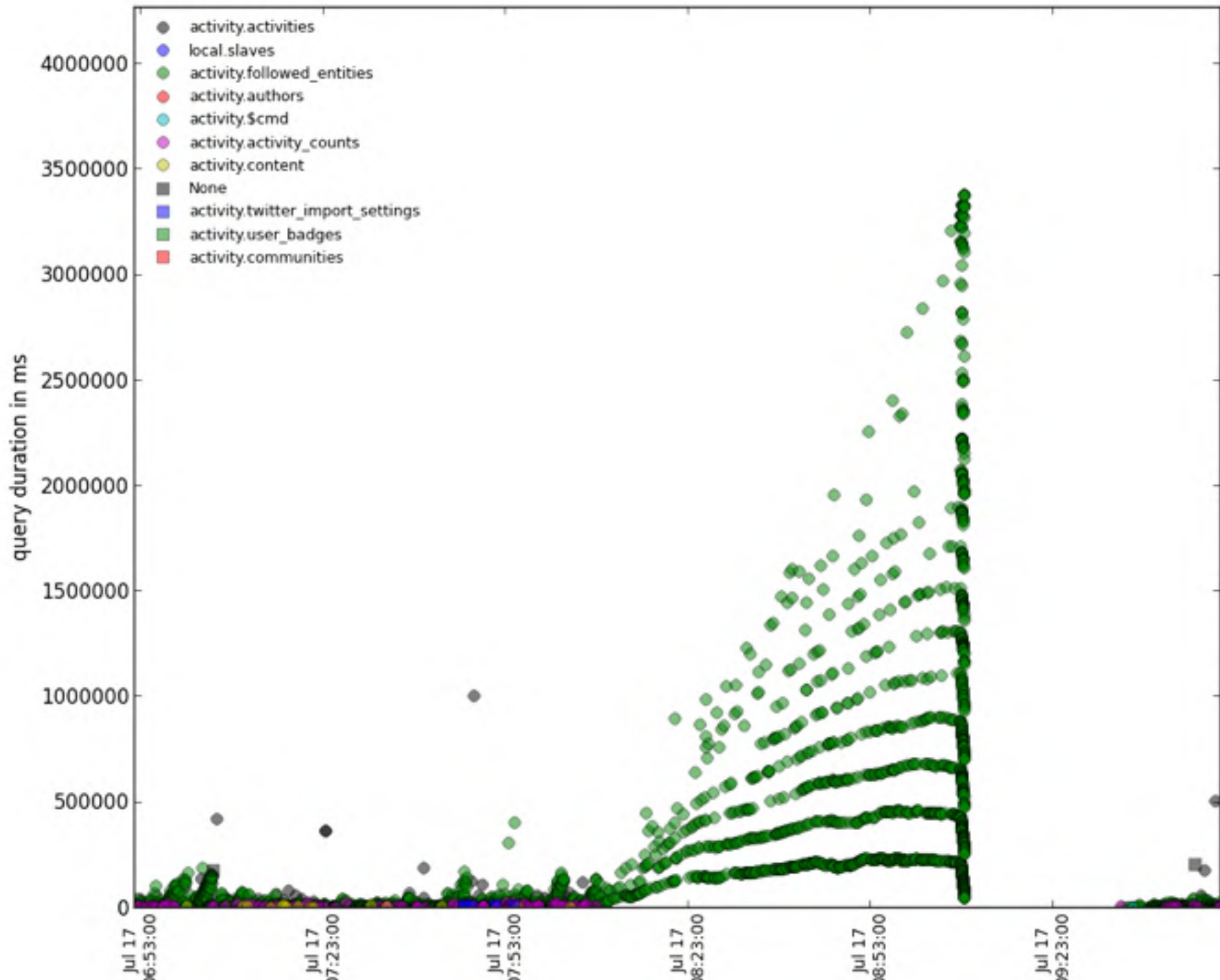


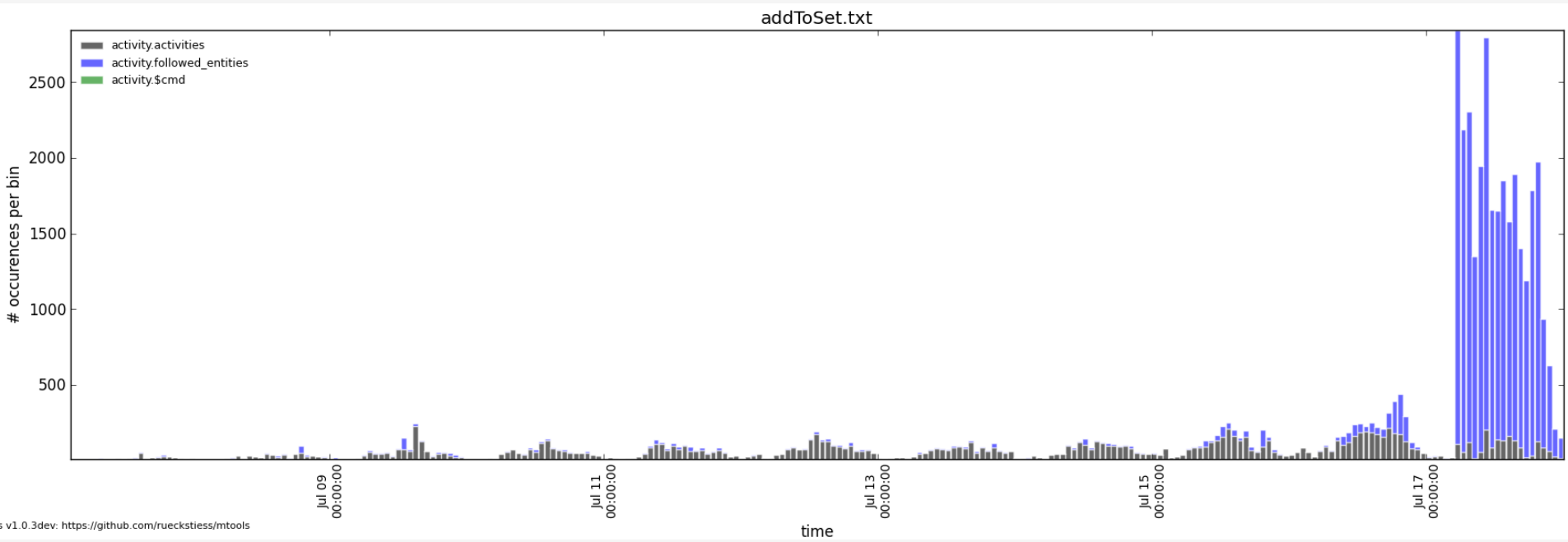
page faults



opcounters

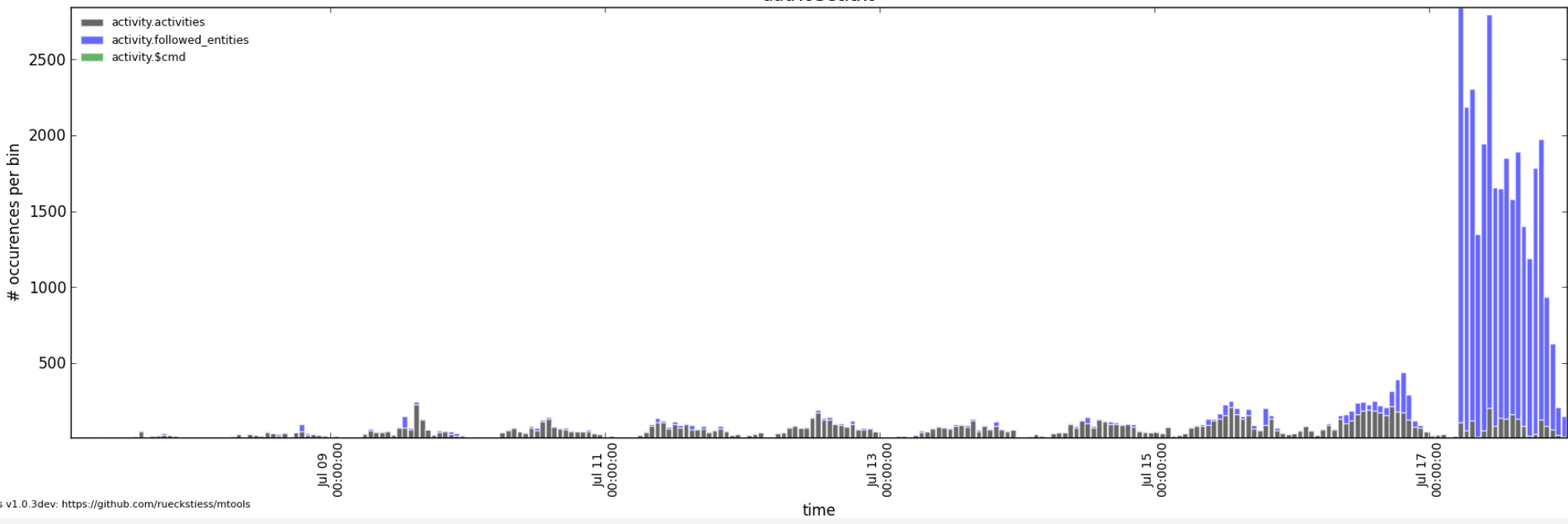




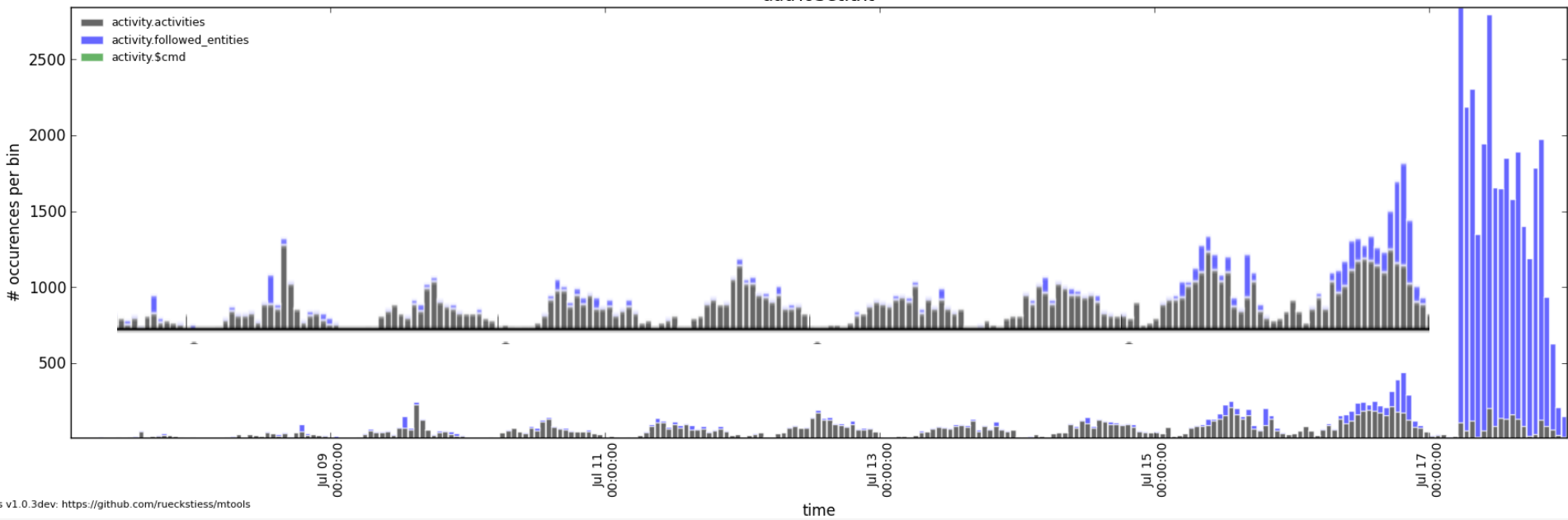


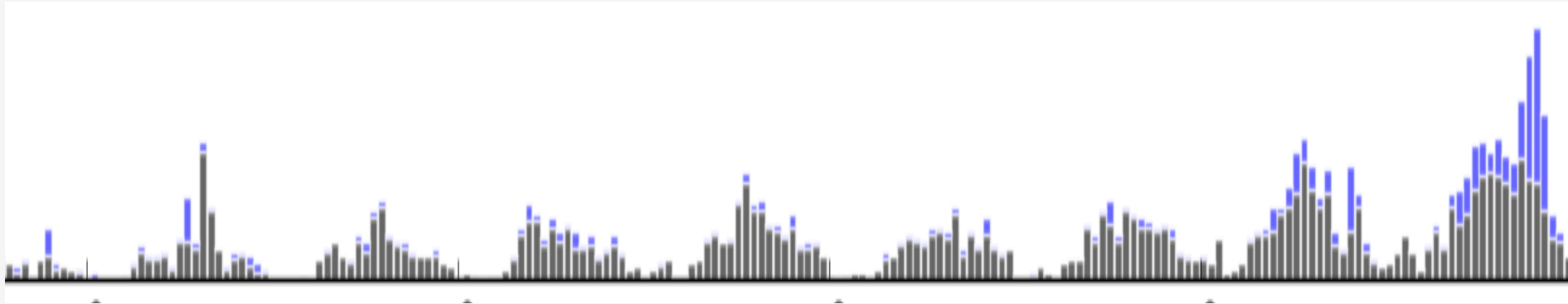
```
% mplotqueries -type histogram -group namespace -bucketSize 3600
```

addToSet.txt

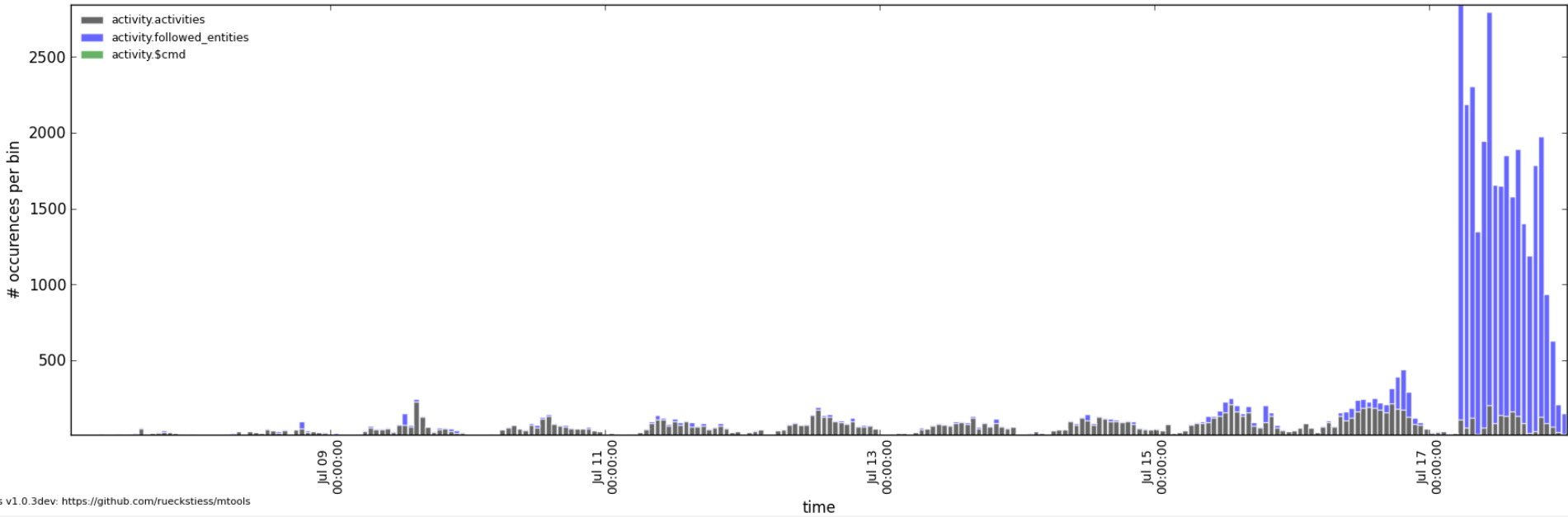


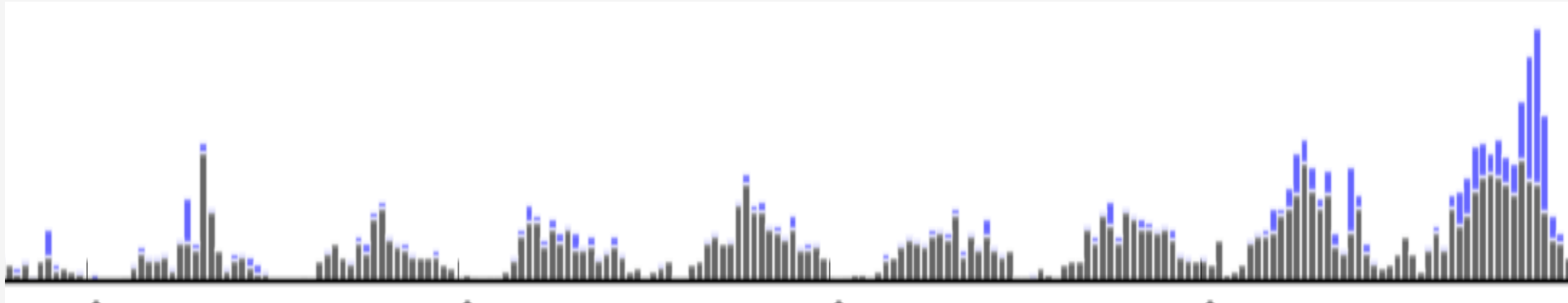
addToSet.txt

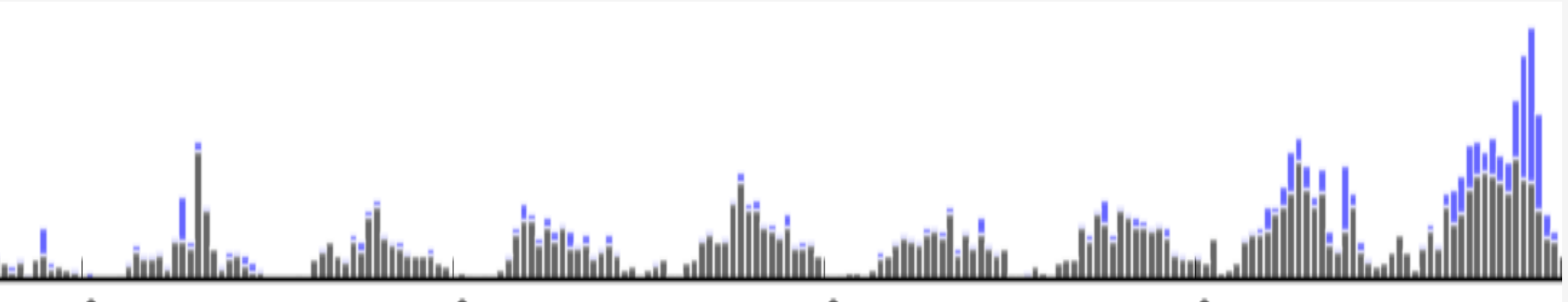


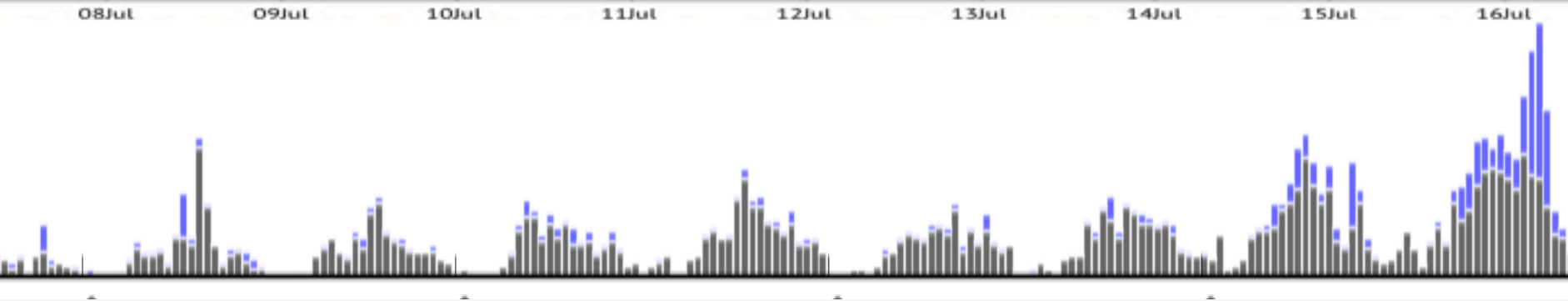
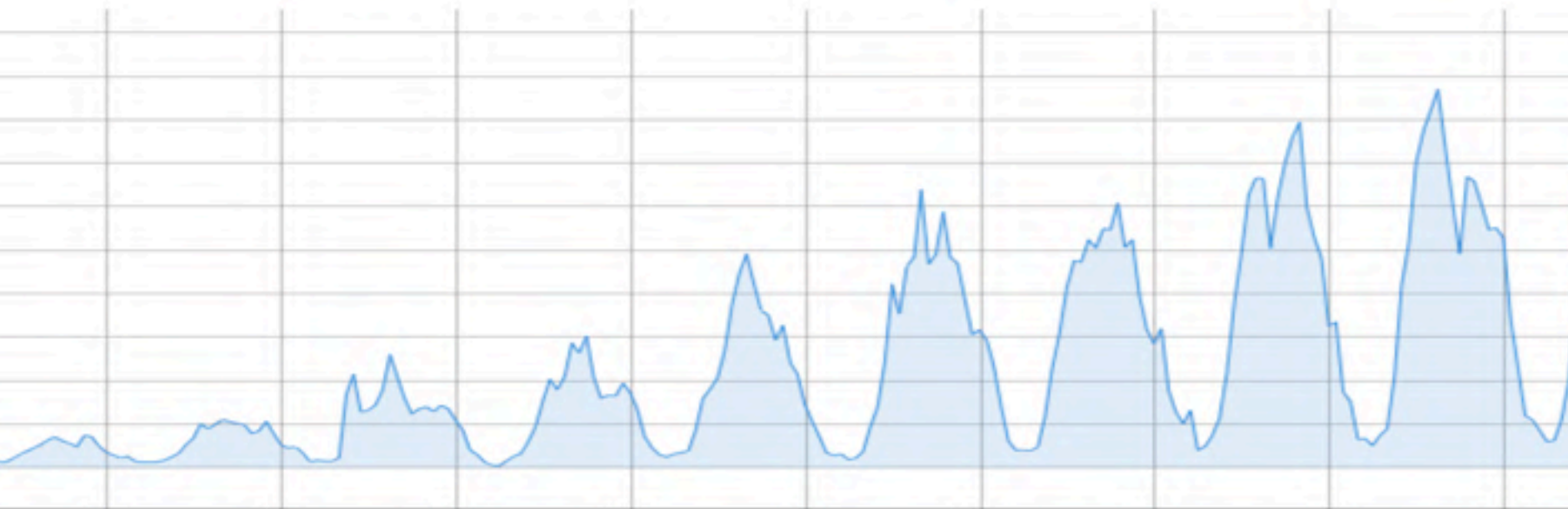


addToSet.txt





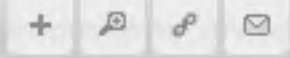




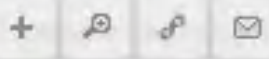
The Stockbroker's Clerk



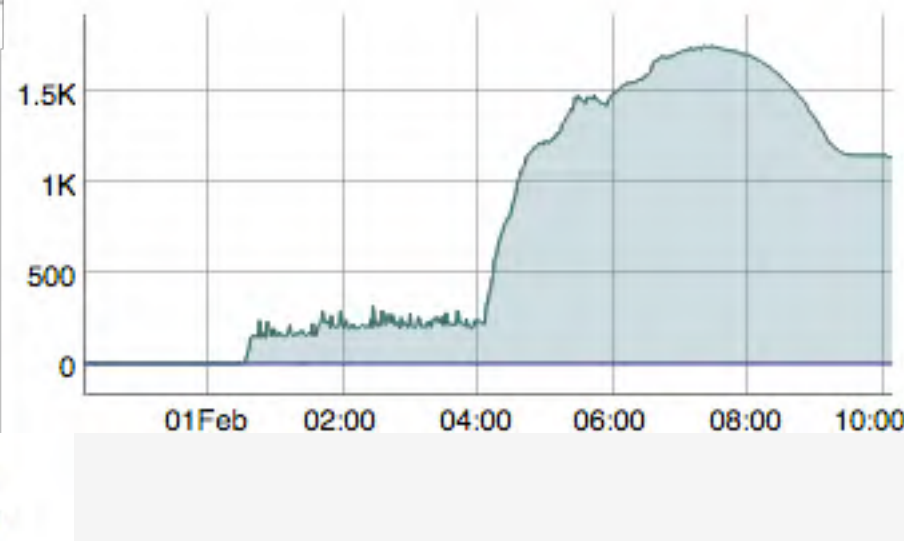
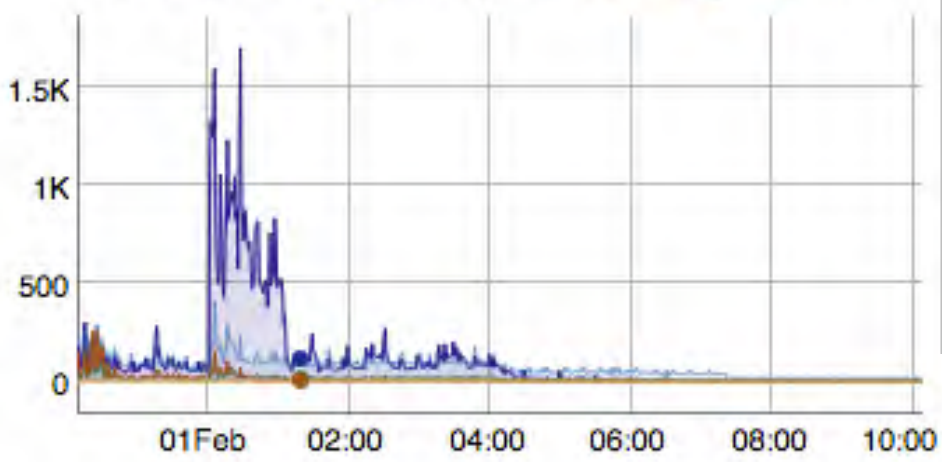
cursors



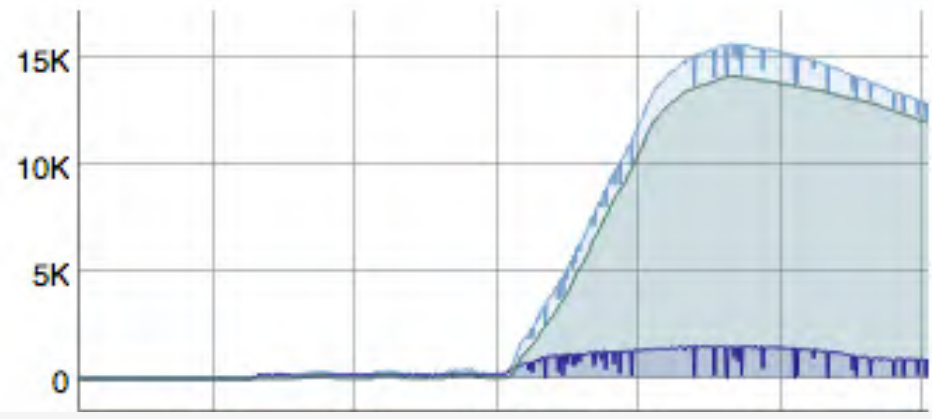
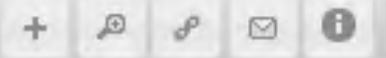
opcounters



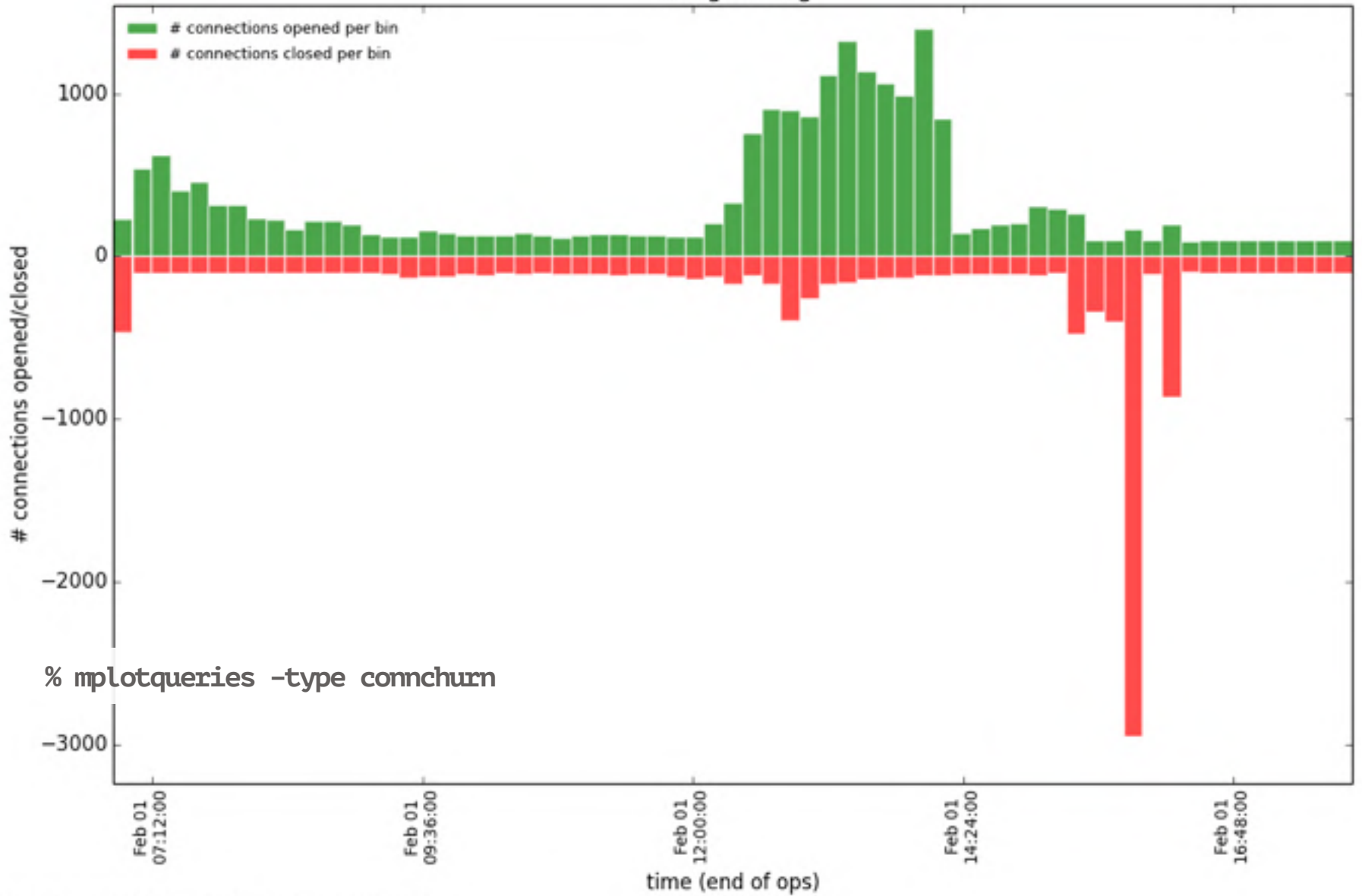
2013/02/01 01:21: query:115.0 update:11.8 delete:0 getmore:5.6



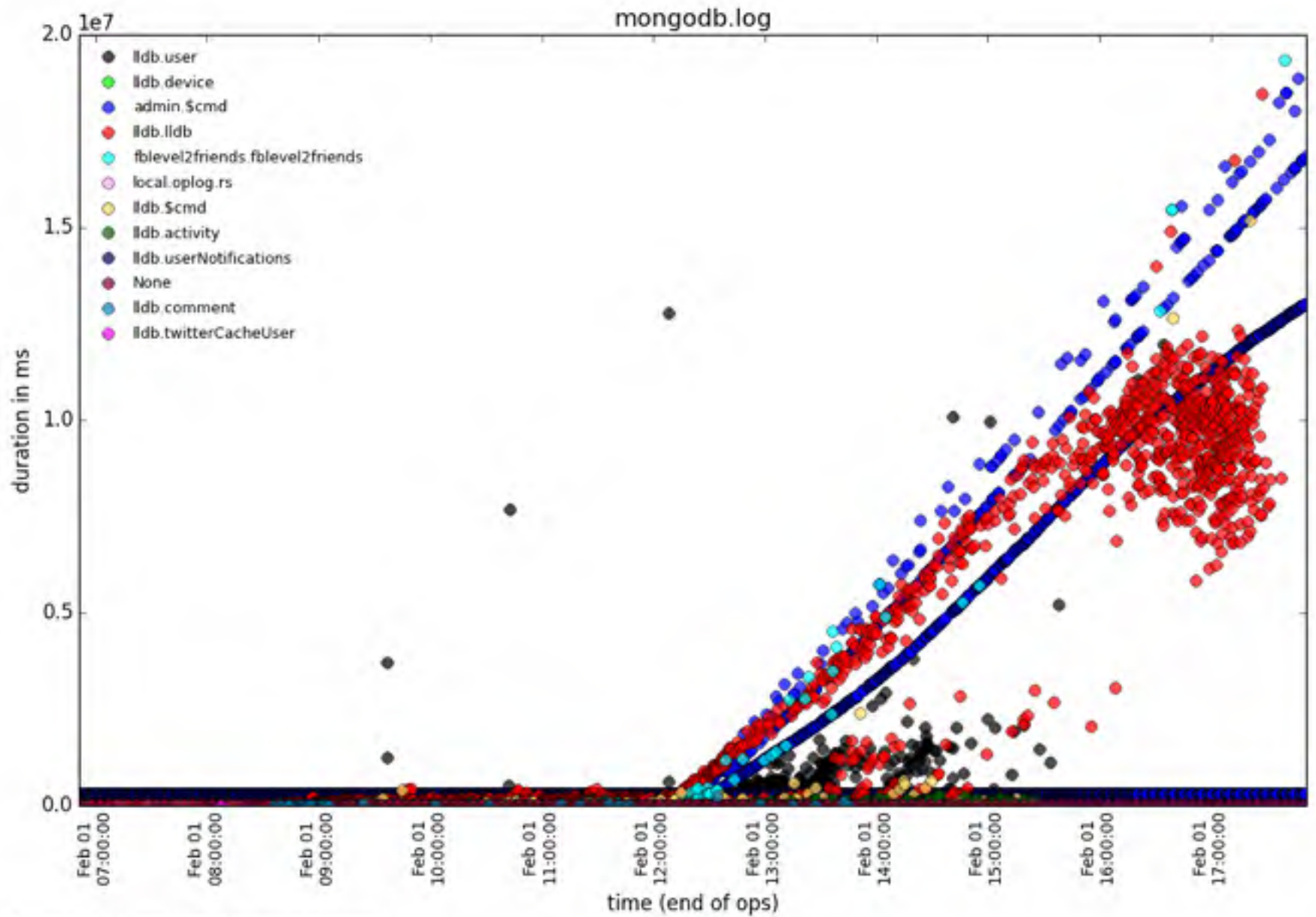
queues



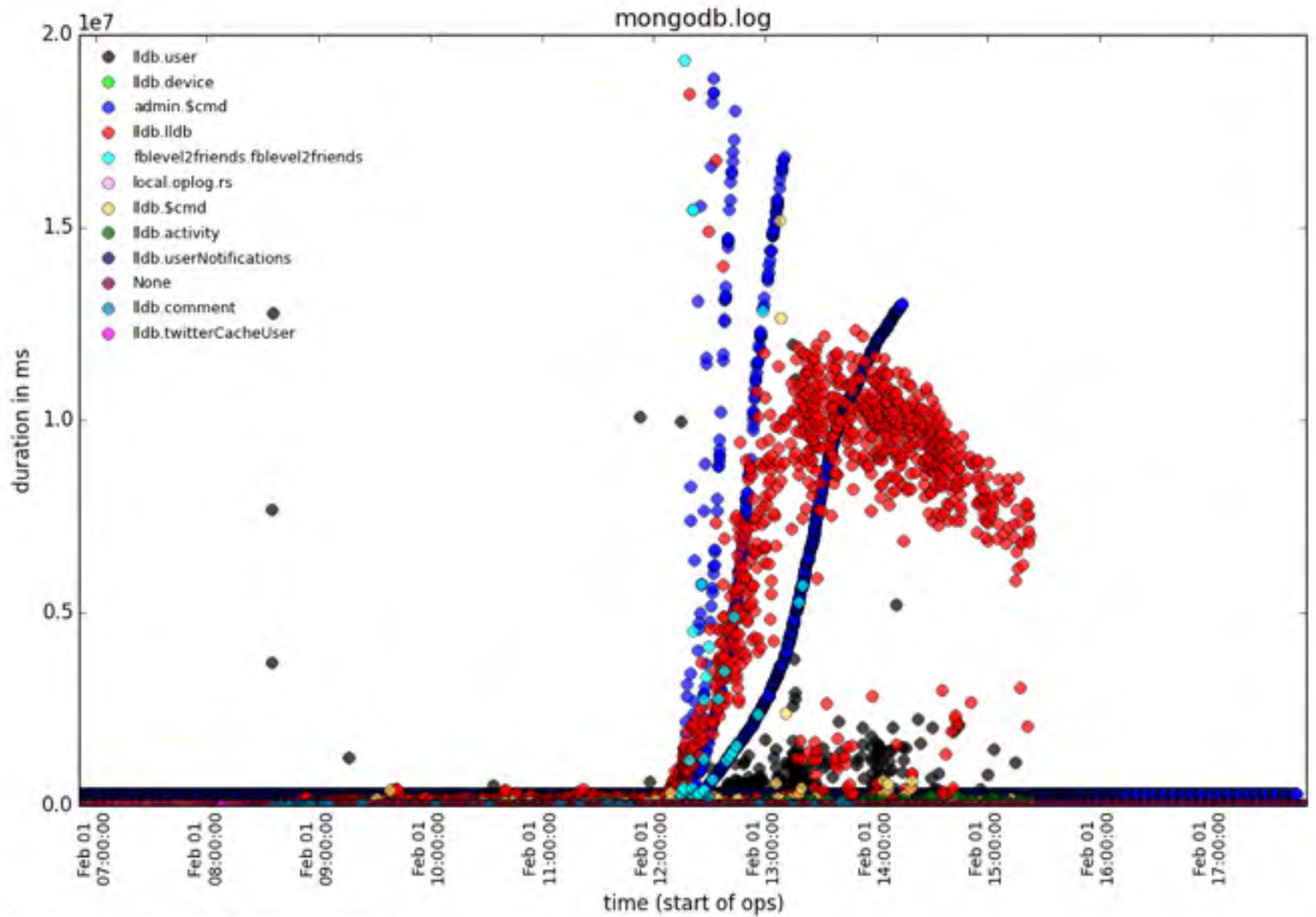
mongodb.log



created with mtools v1.1.5: <https://github.com/rueckstiess/mtools>

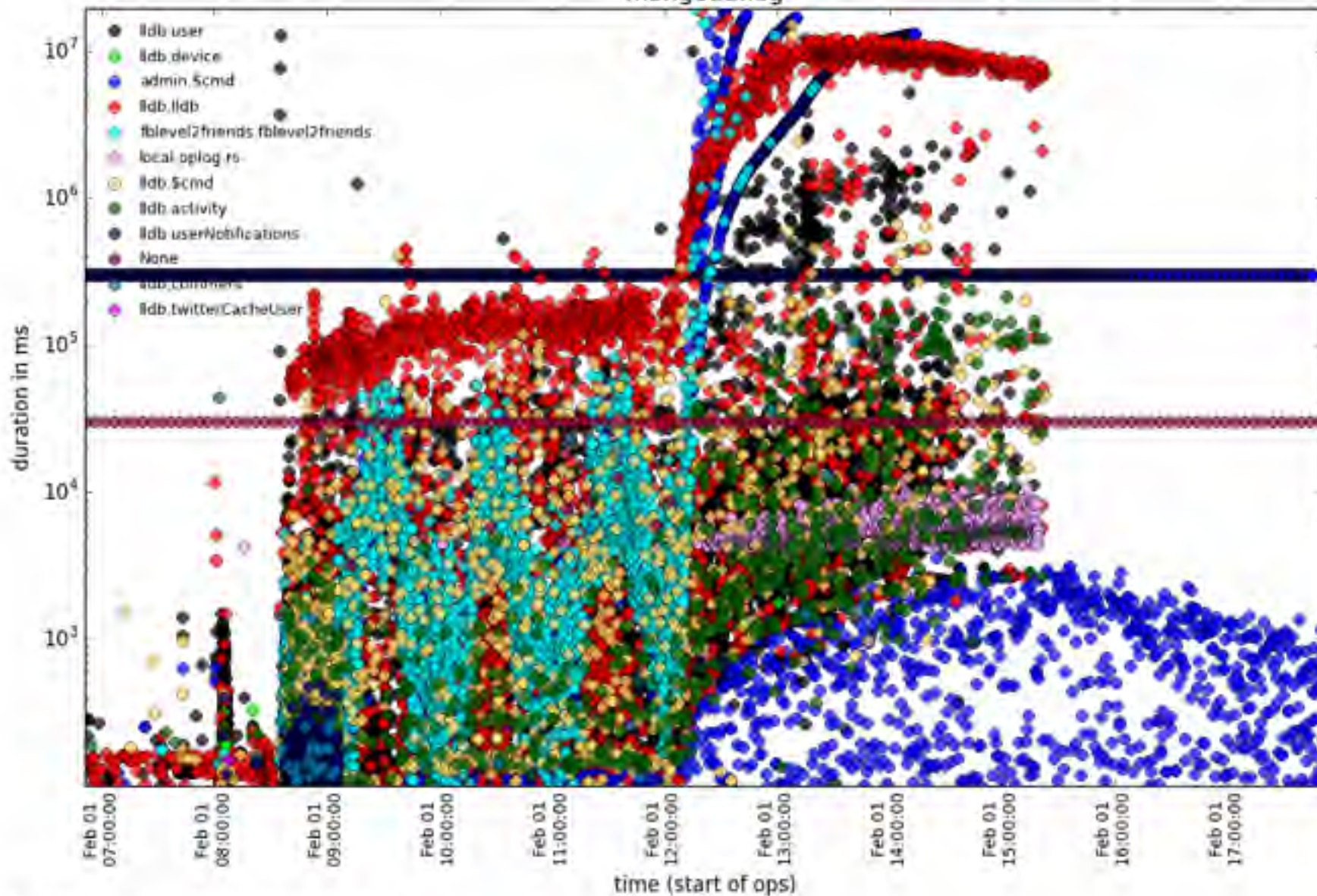


created with mtools v1.1.5: <https://github.com/rueckstiehs/mtools>



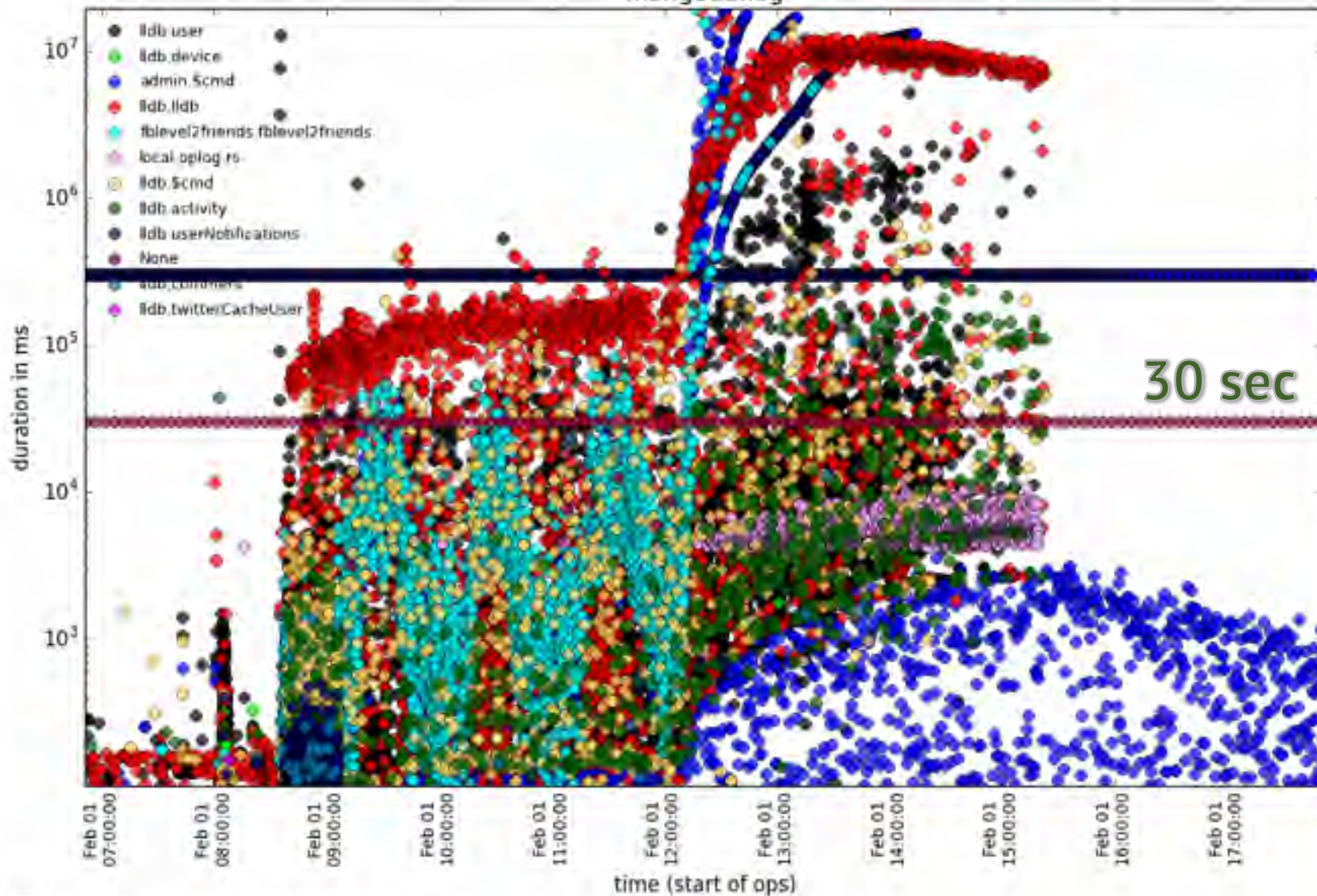
created with mtools v1.1.5: <https://github.com/ruecksties/mtools>

mongodb.log



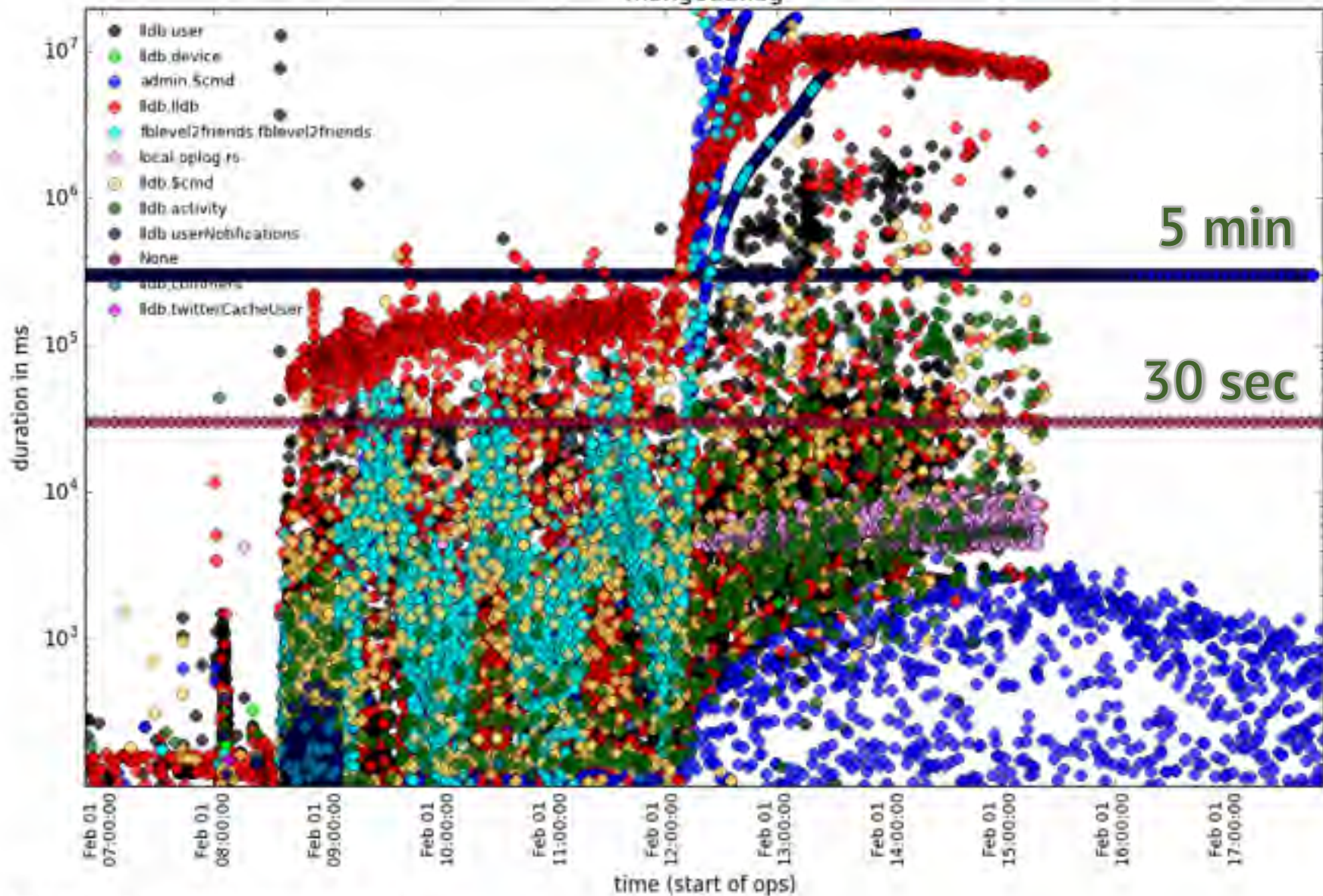
created with mtools V1.1.5 | <https://github.com/rueickstess/mtools>

mongodb.log

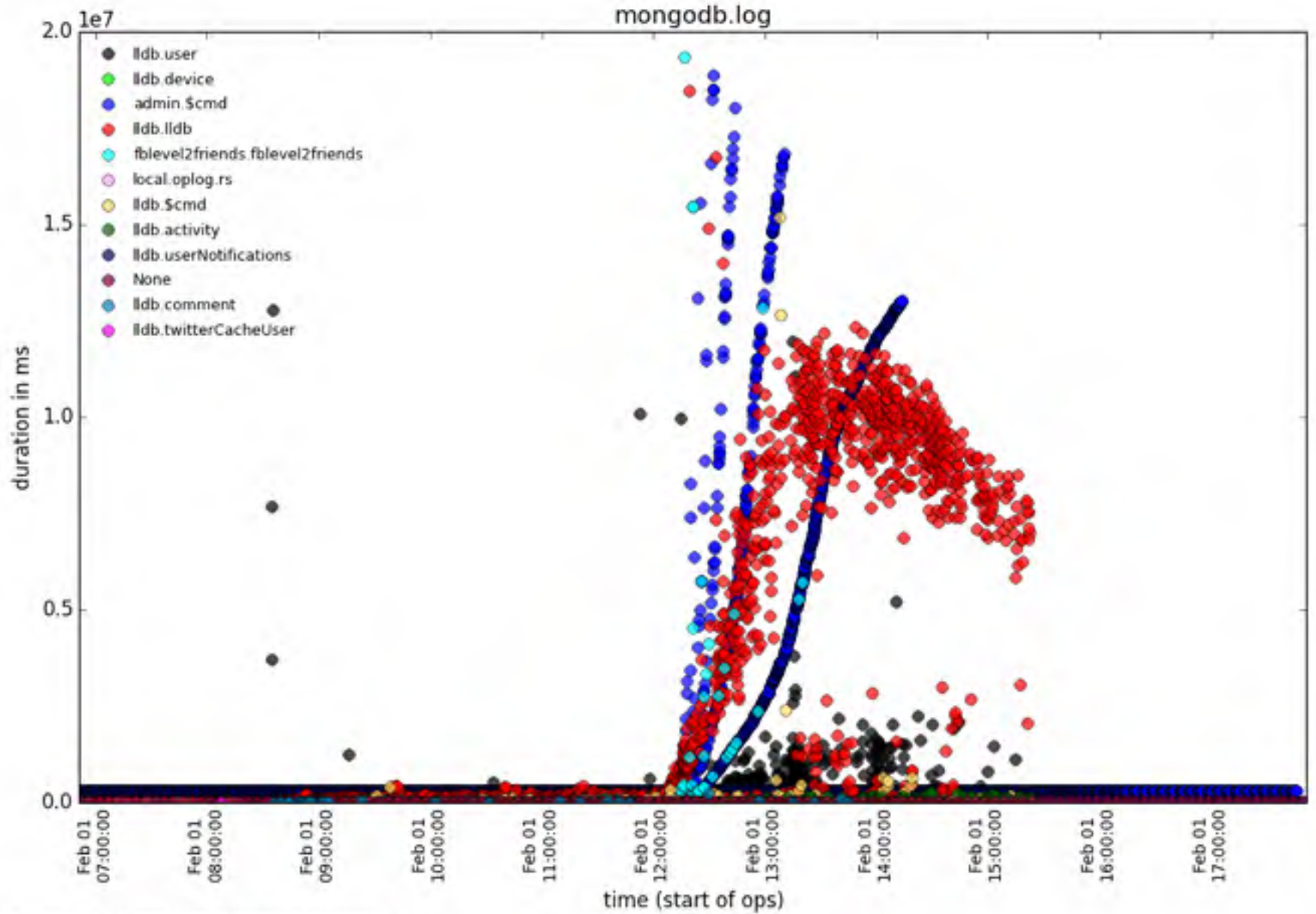


created with mtools V1.1.5 | <https://github.com/rueickstess/mtools>

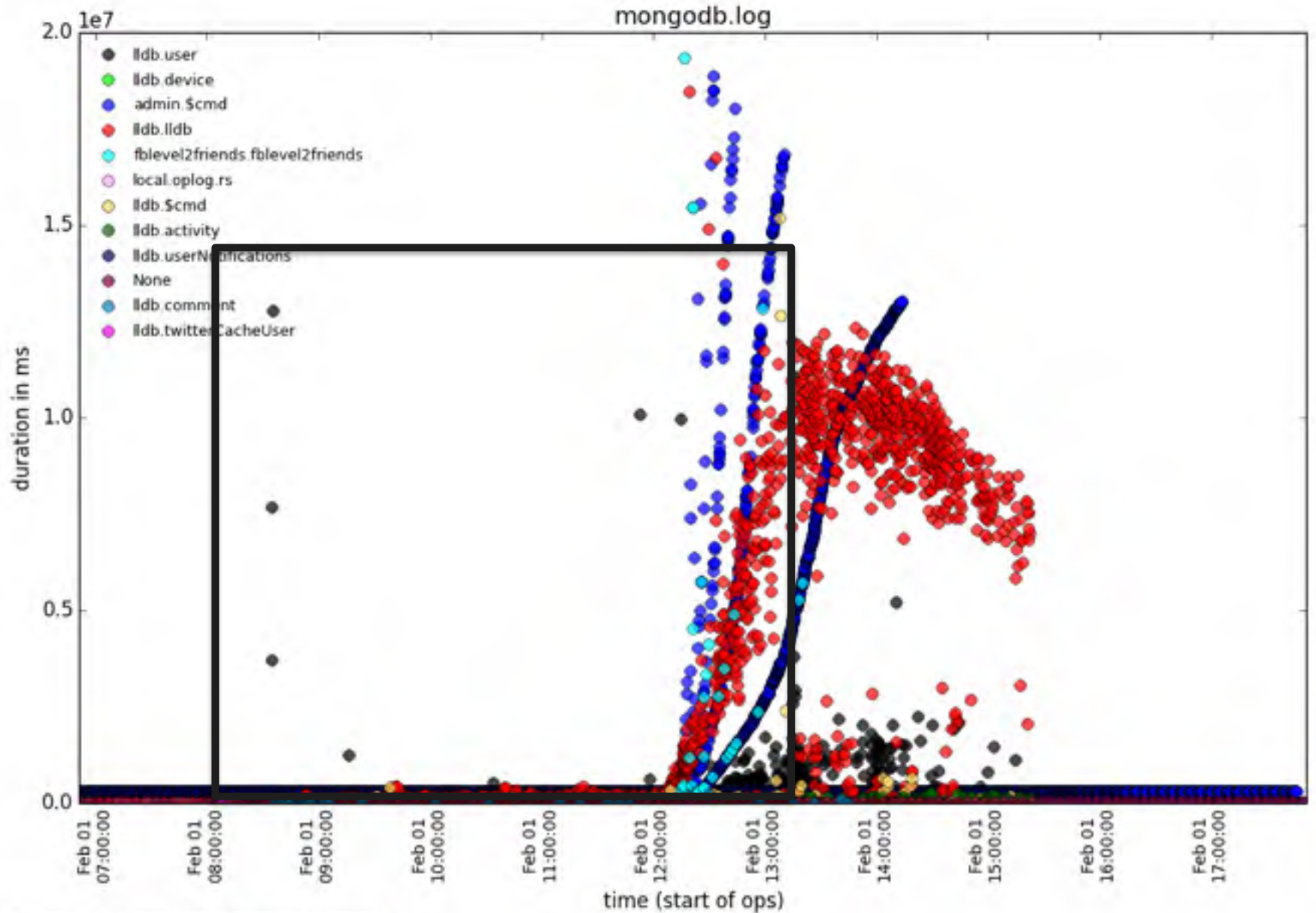
mongodb.log



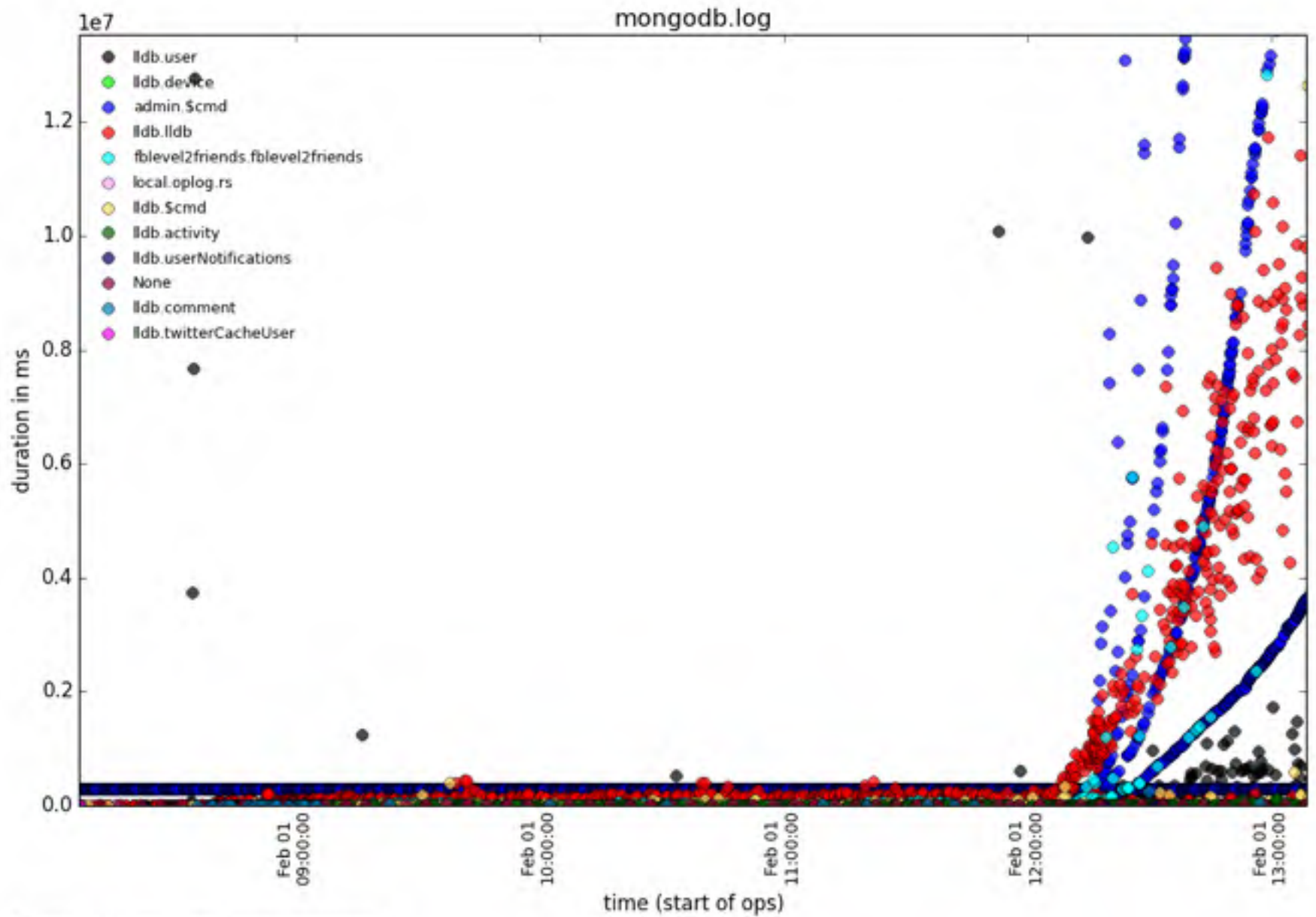
created with mtools V1.1.5 | <https://github.com/rueickstess/mtools>



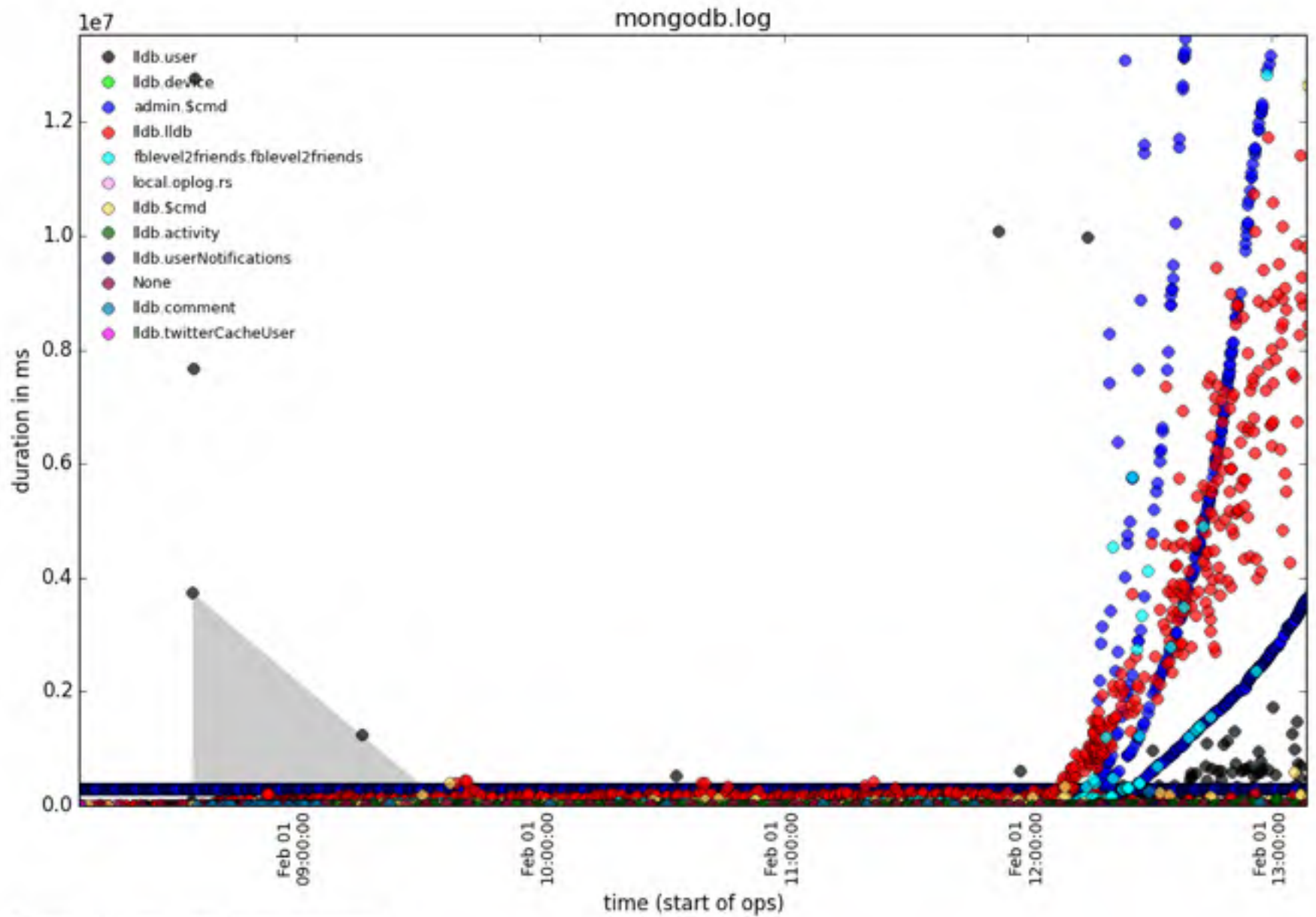
created with mtools v1.1.5: <https://github.com/rueckstiess/mtools>



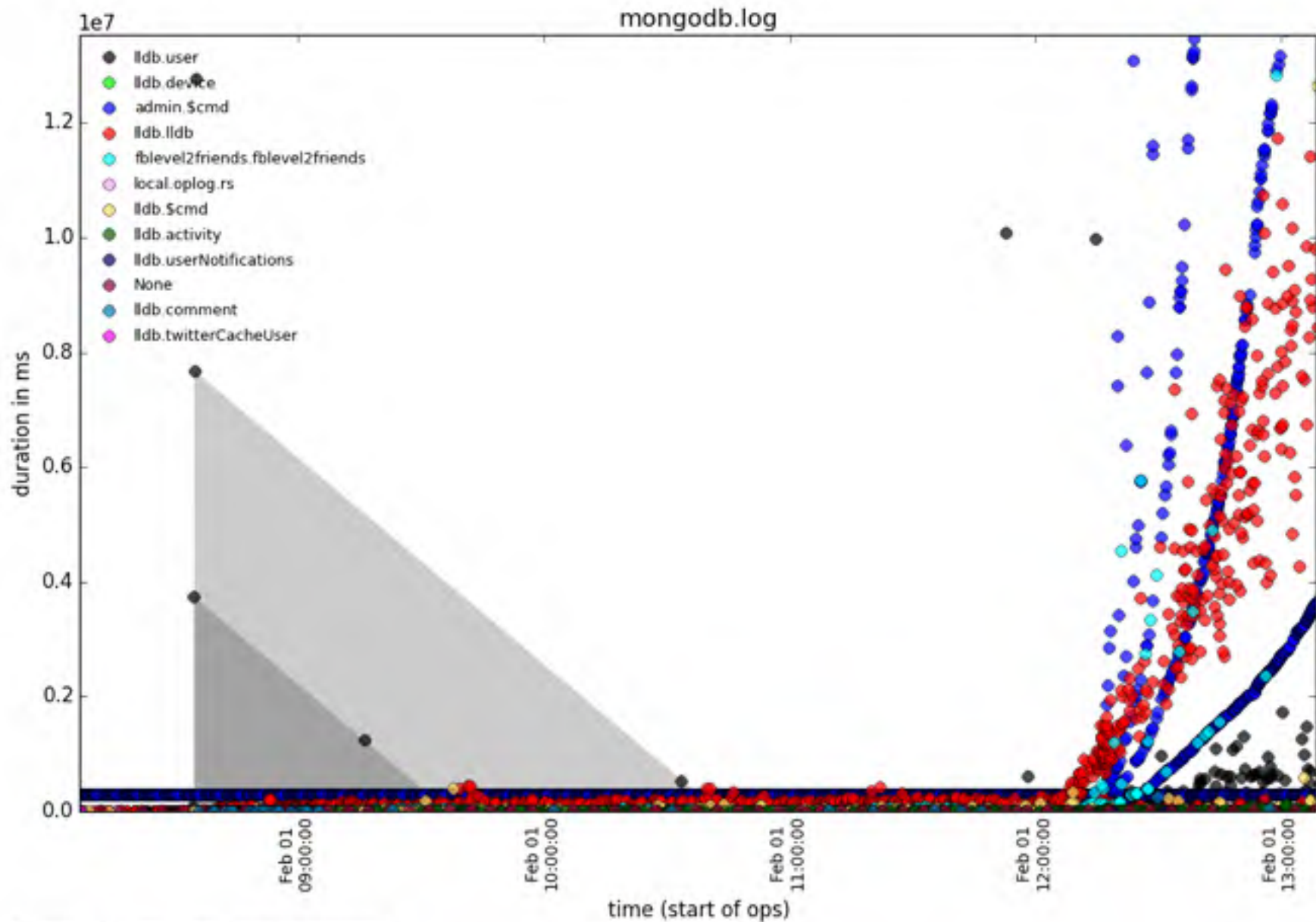
created with mtools v1.1.5: <https://github.com/rueckstiess/mtools>



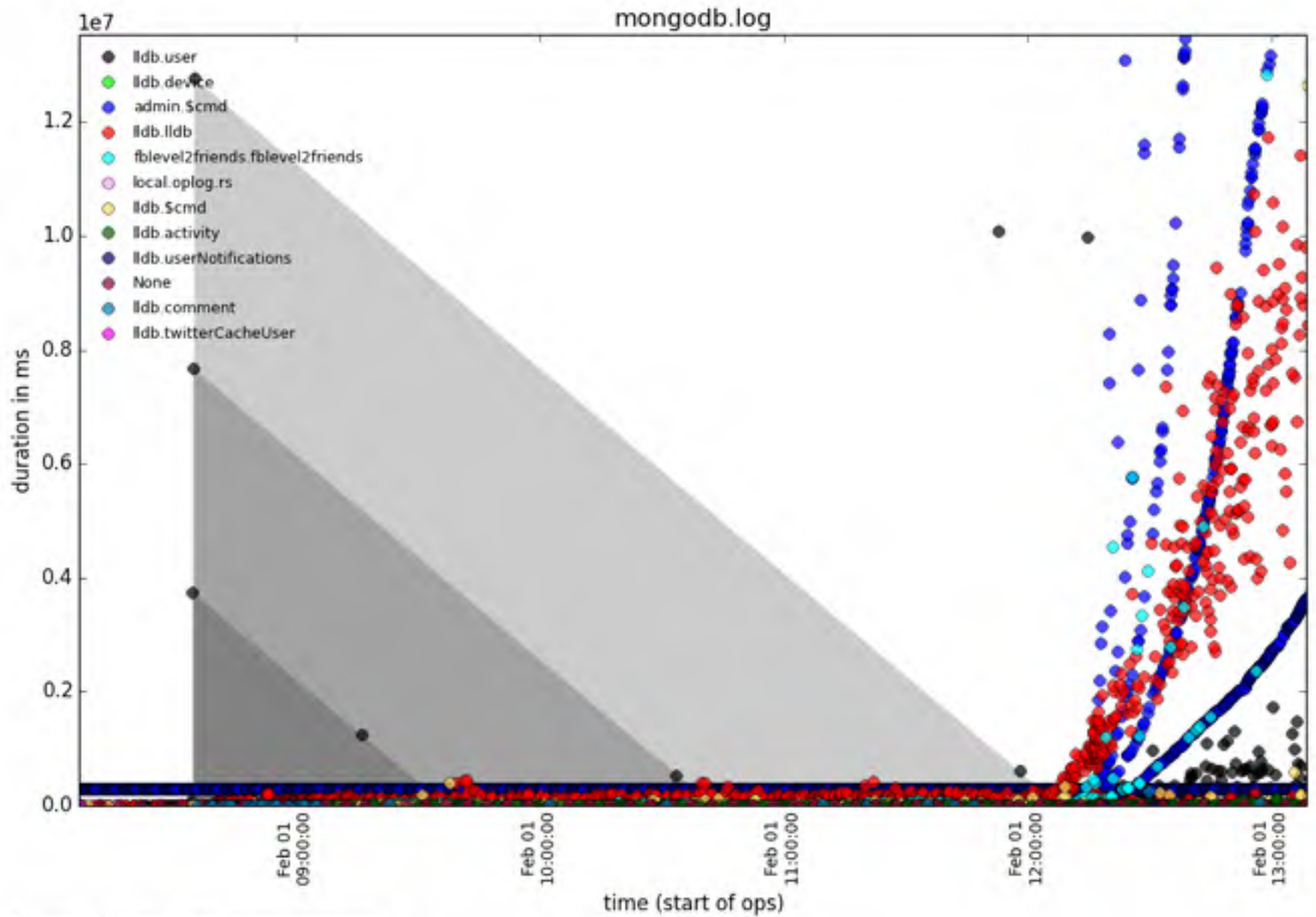
created with mtools v1.1.5: <https://github.com/rueckstiess/mtools>



created with mtools v1.1.5: <https://github.com/ruecksties/mtools>



created with mtools v1.1.5: <https://github.com/ruecksties/mtools>



created with mtools v1.1.5: <https://github.com/ruecksties/mtools>



"We balance probabilities and choose the most likely. It is the scientific use of the imagination."

Sherlock Holmes, *The Hound of the Baskervilles*

With profound thanks to:

Sir Arthur Conan Doyle (1859-1930)



#MongoDB @asya999 #askAsya

Thank You

Asya Kamsky

MongoDB, Inc.

<http://askasya.com/>

<http://www.kamsky.org/>