elastic{ON}15

Elasticsearch in Anger

stories from the **GitHub** search clusters

Tim Pease

- Where have we come from
- What we are doing now
- Where we are going

A Whole New Code Search

January 23, 2013 TwP New Features Edit

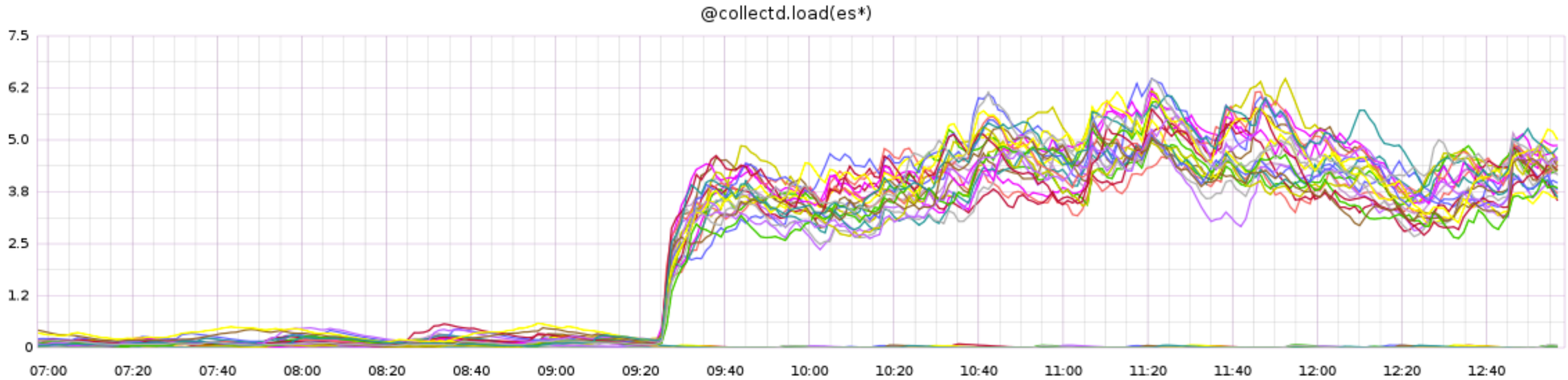
Finding great code on GitHub just got a whole lot easier. Today we're releasing several big improvements to code search.

New Technology

First, we are unveiling our new search infrastructure that will grow to support the immense amount of source code being pushed to GitHub each day.

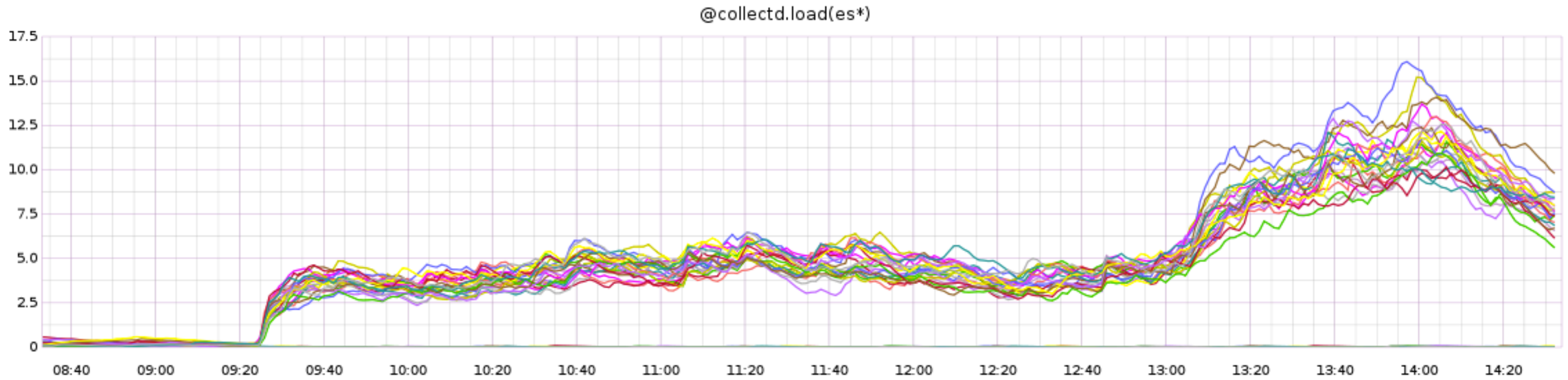
Under the hood is an ElasticSearch cluster that live-indexes your code as you push it up to GitHub. Search results will be returned from public and private repositories that you have access to.

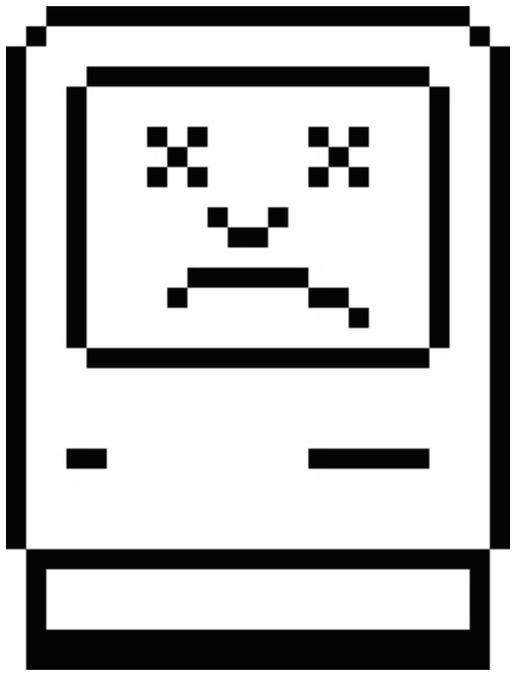
Code Search Load




“you can find **secret keys**
people have committed to their
public repositories”

Code Search Load



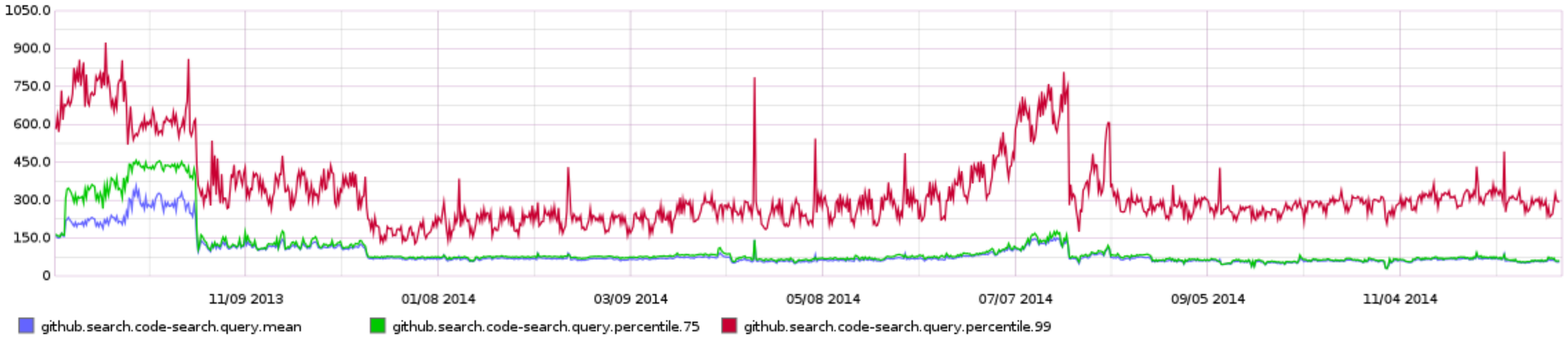


Drew Raines

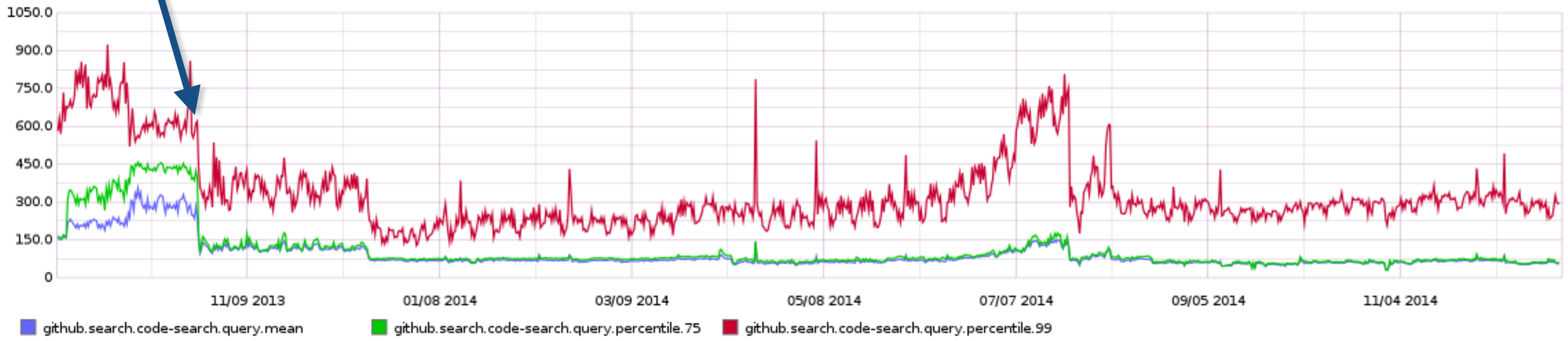
Code Search

- We performed inadequate load testing
- We had insufficient operations experience
- We need better tools and metrics

Code Search Query Performance



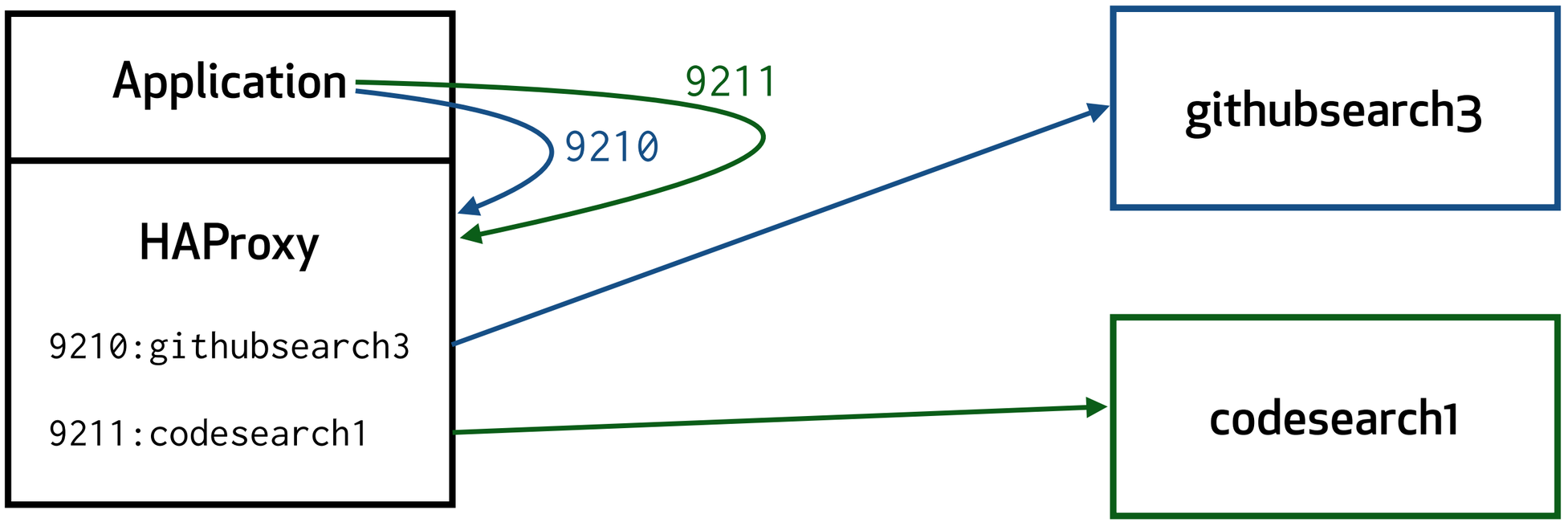
New Cluster


1050.0
900.0
750.0
600.0
450.0
300.0
150.0
0
11/09 2013
01/08 2014
03/09 2014
05/08 2014
07/07 2014
09/05 2014
11/04 2014
github.search.code-search.query.mean
github.search.code-search.query.percentile.75
github.search.code-search.query.percentile.99

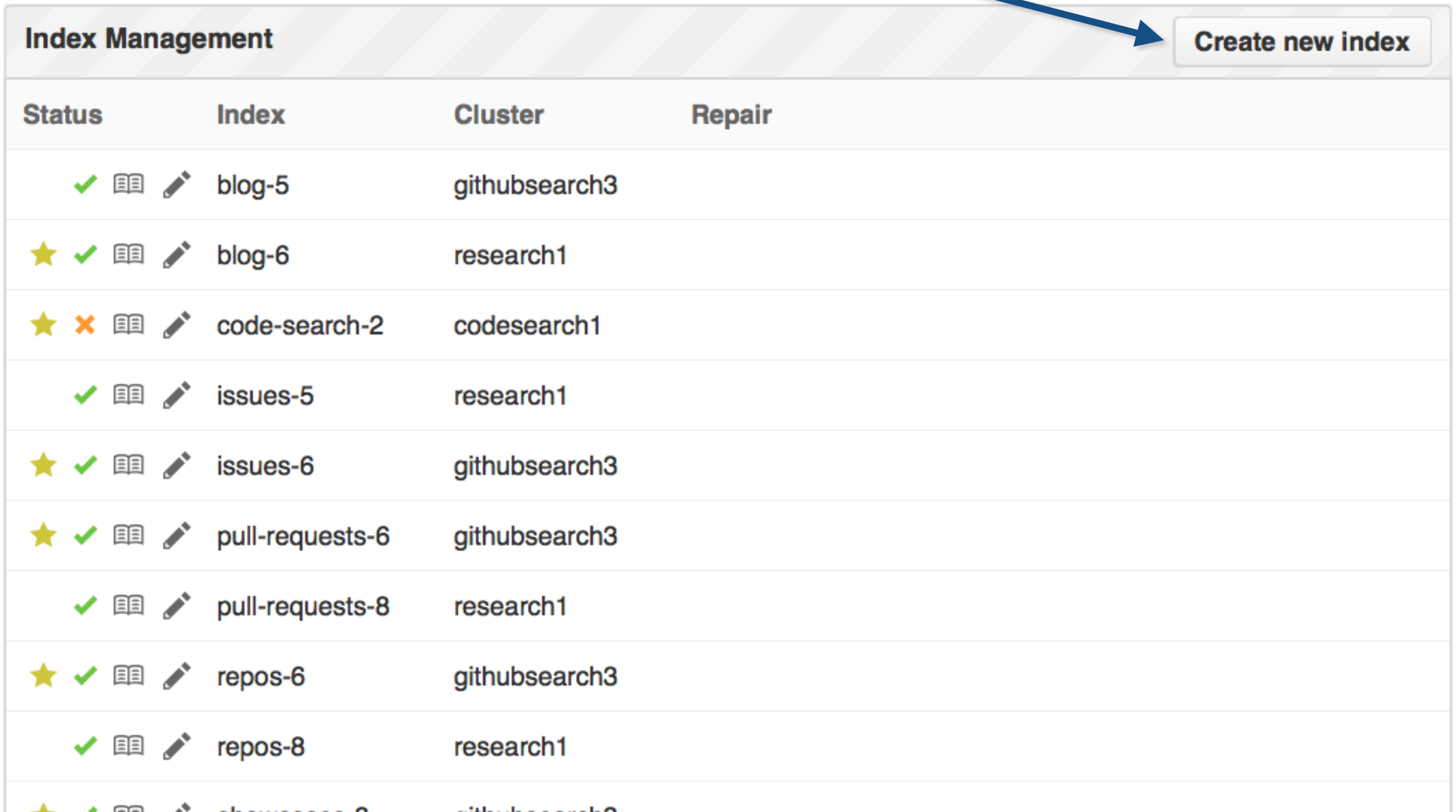
Index Management Create new index

Status	Index	Cluster	Repair
✔ 📖 ✎	blog-5	githubsearch3	
★ ✔ 📖 ✎	blog-6	research1	
★ ✖ 📖 ✎	code-search-2	codesearch1	
✔ 📖 ✎	issues-5	research1	
★ ✔ 📖 ✎	issues-6	githubsearch3	
★ ✔ 📖 ✎	pull-requests-6	githubsearch3	
✔ 📖 ✎	pull-requests-8	research1	
★ ✔ 📖 ✎	repos-6	githubsearch3	
✔ 📖 ✎	repos-8	research1	

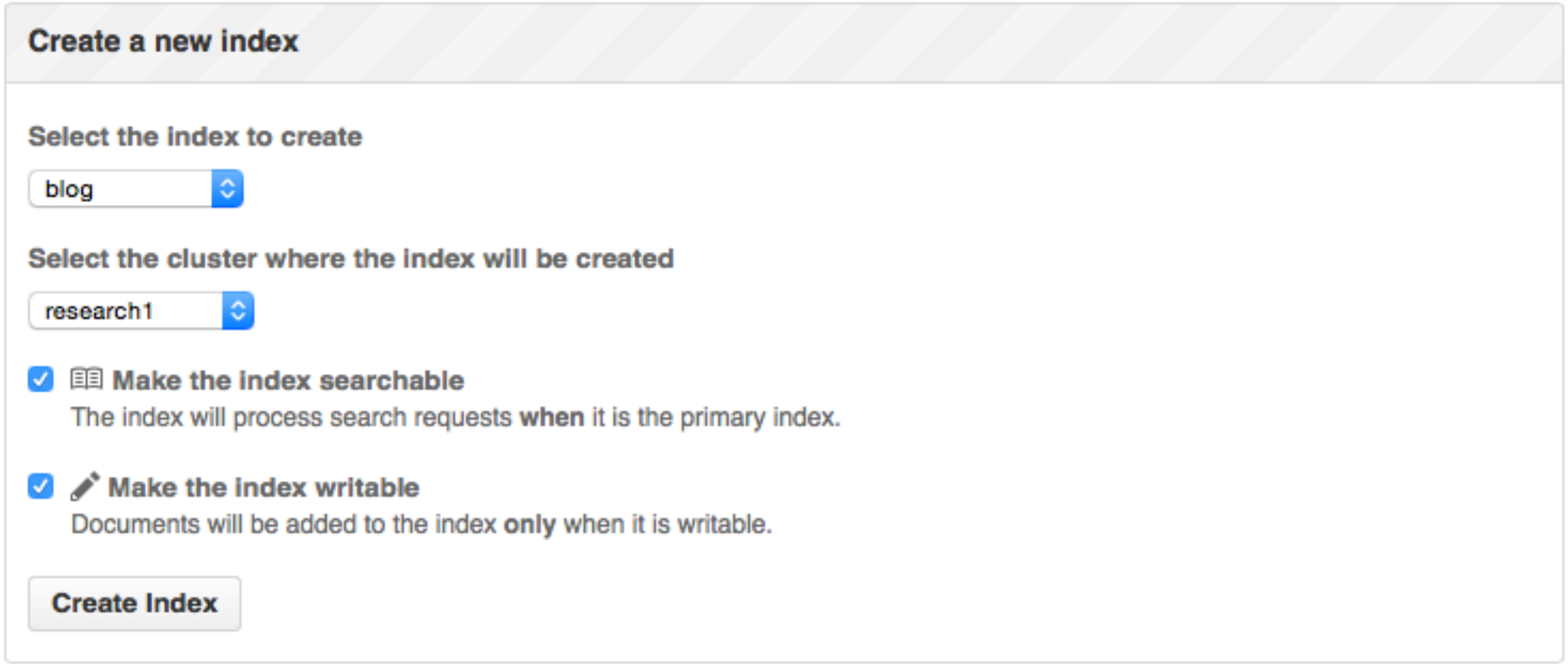
HAProxy for Multiple Clusters



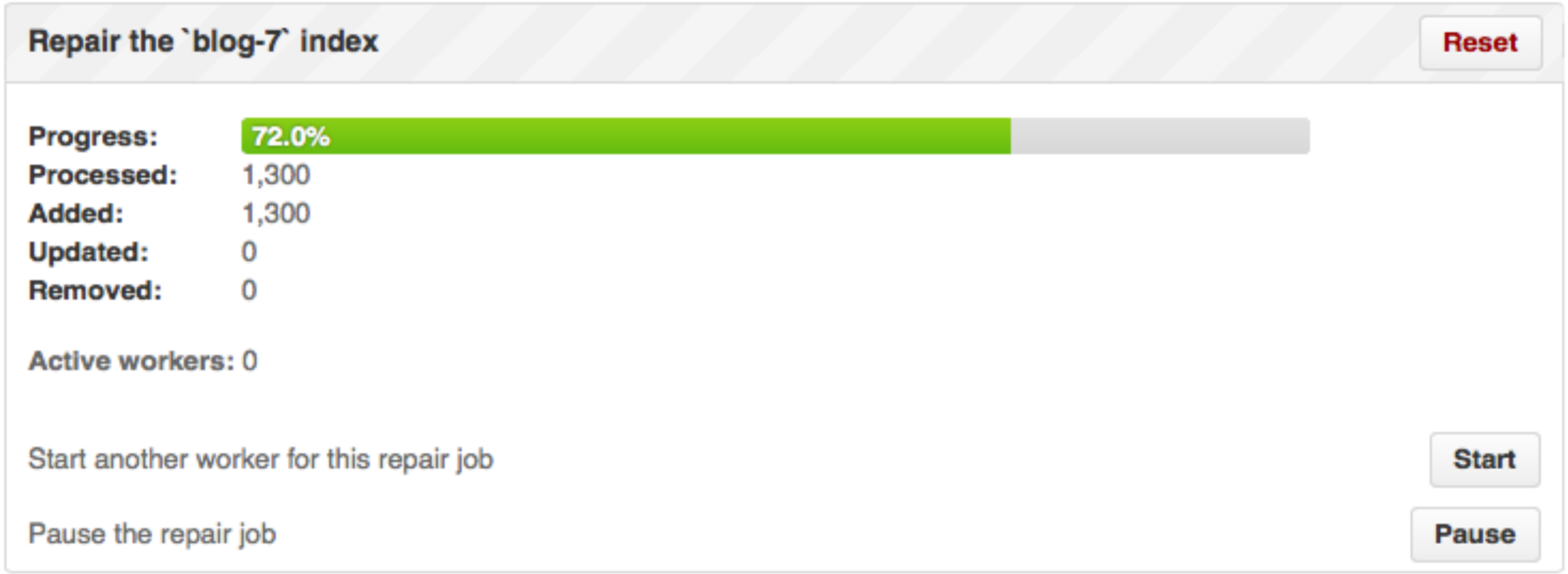
Push Button Index Creation



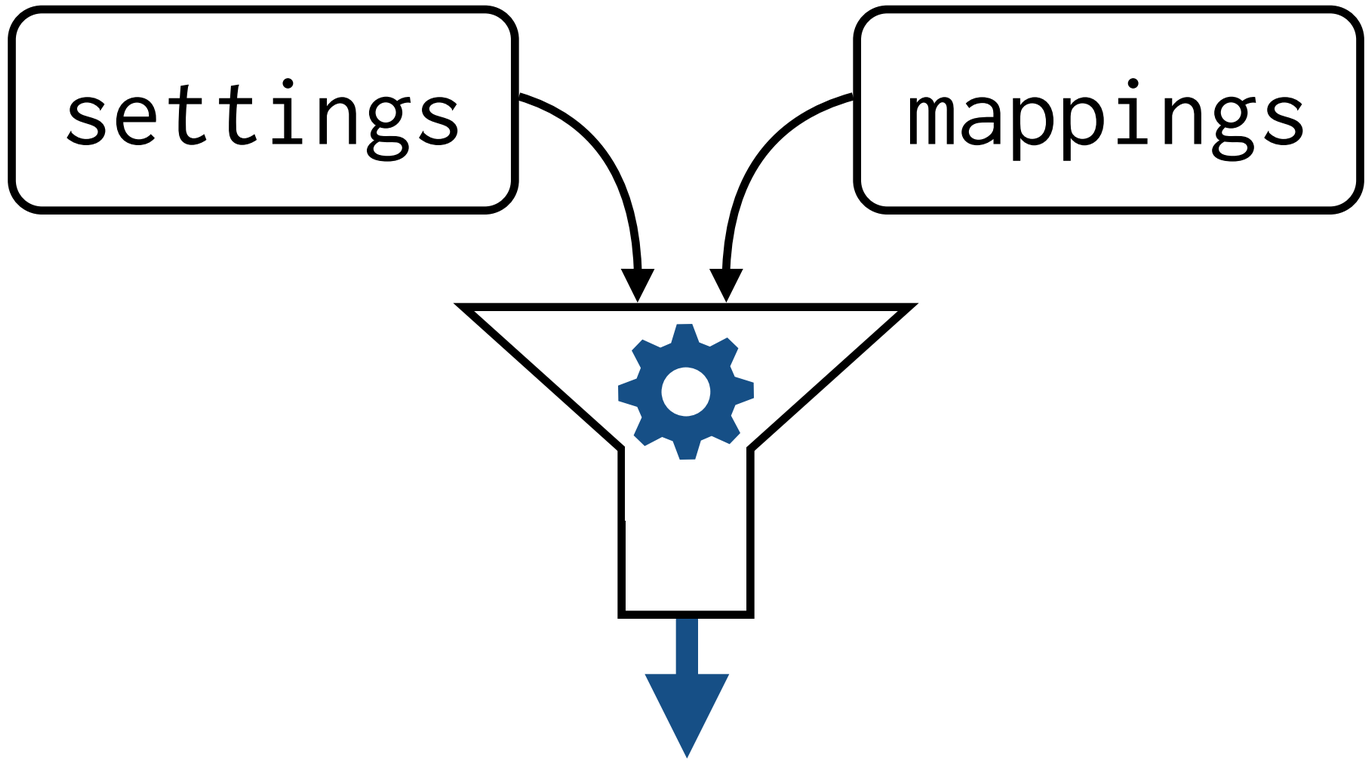
Create a New Index



Backfill Data



✔ Index Versions ✖



✔ Index Versions ✖

```
mappings: {
   index-meta: {
      _meta: {
         version: <SHA1 Hexdigest>
      }
   }
}
```

Because ...

GitHub Enterprise

Load Testing

Load Testing

Scientist

github.com/github/scientist

- control
- experiment
- throttling

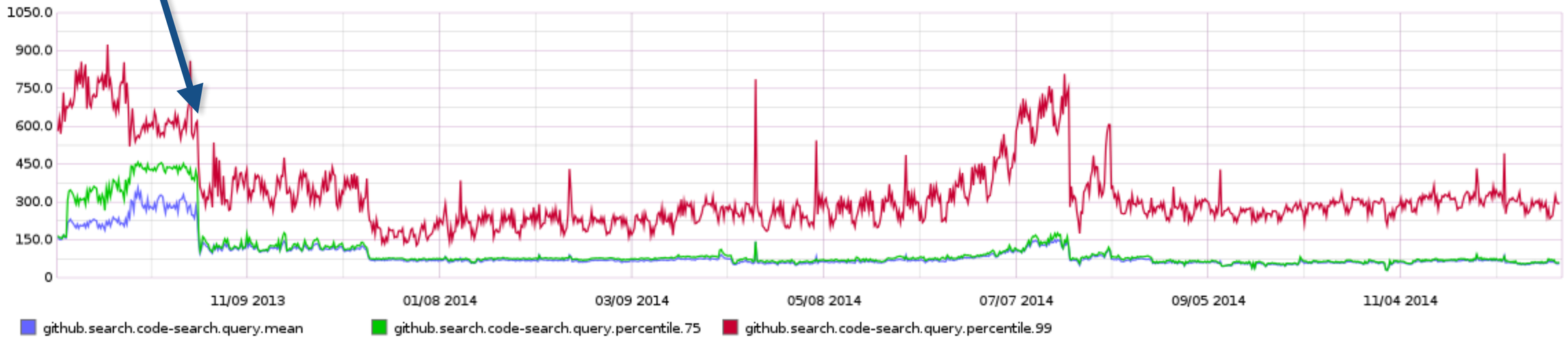
Load Testing

```
require "scientist"

def search(query)
  science "code-search-load-test" do |e|
    e.use { old_index.search(query) }
    e.try { new_index.search(query) }
  end
end
```

New Cluster

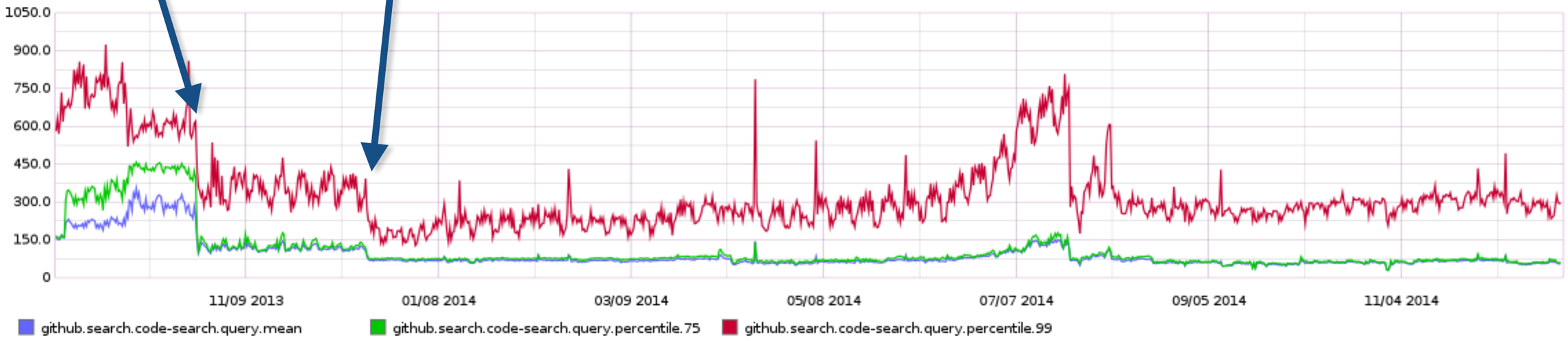
- We were outgrowing our old cluster
- We created migration tools
- We used production queries for load testing

New Cluster


1050.0
900.0
750.0
600.0
450.0
300.0
150.0
0
11/09 2013
01/08 2014
03/09 2014
05/08 2014
07/07 2014
09/05 2014
11/04 2014
github.search.code-search.query.mean
github.search.code-search.query.percentile.75
github.search.code-search.query.percentile.99

New Cluster

New Queries

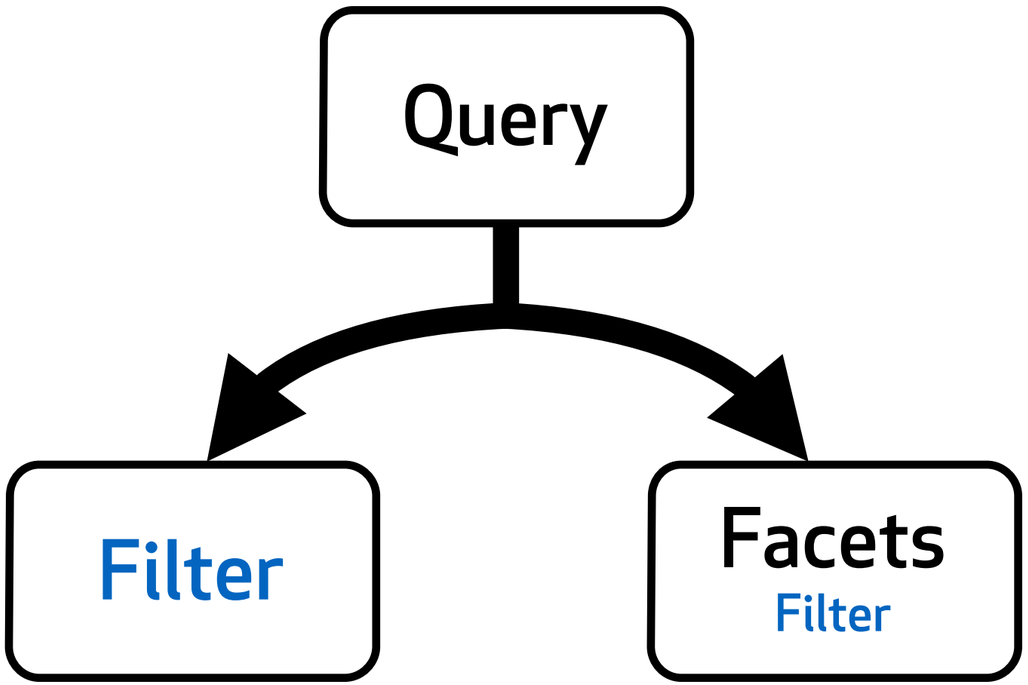


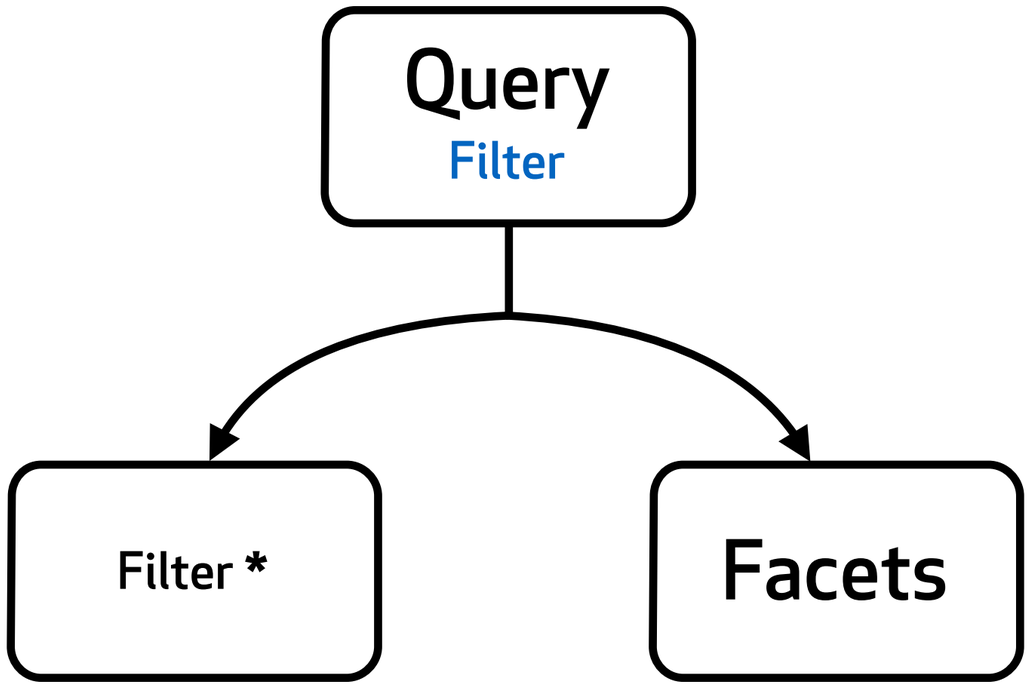
Elasticsearch Training

Tale of Two Queries

```
{
  query: {match_all: {}},
  filter: {
    term: {state: “open”}
  }
}
```

```
{
  query: {constant_score: {
    filter: {
      term: {state: “open”}
    }
  }}
}
```


Query
Filter
Facets
Filter

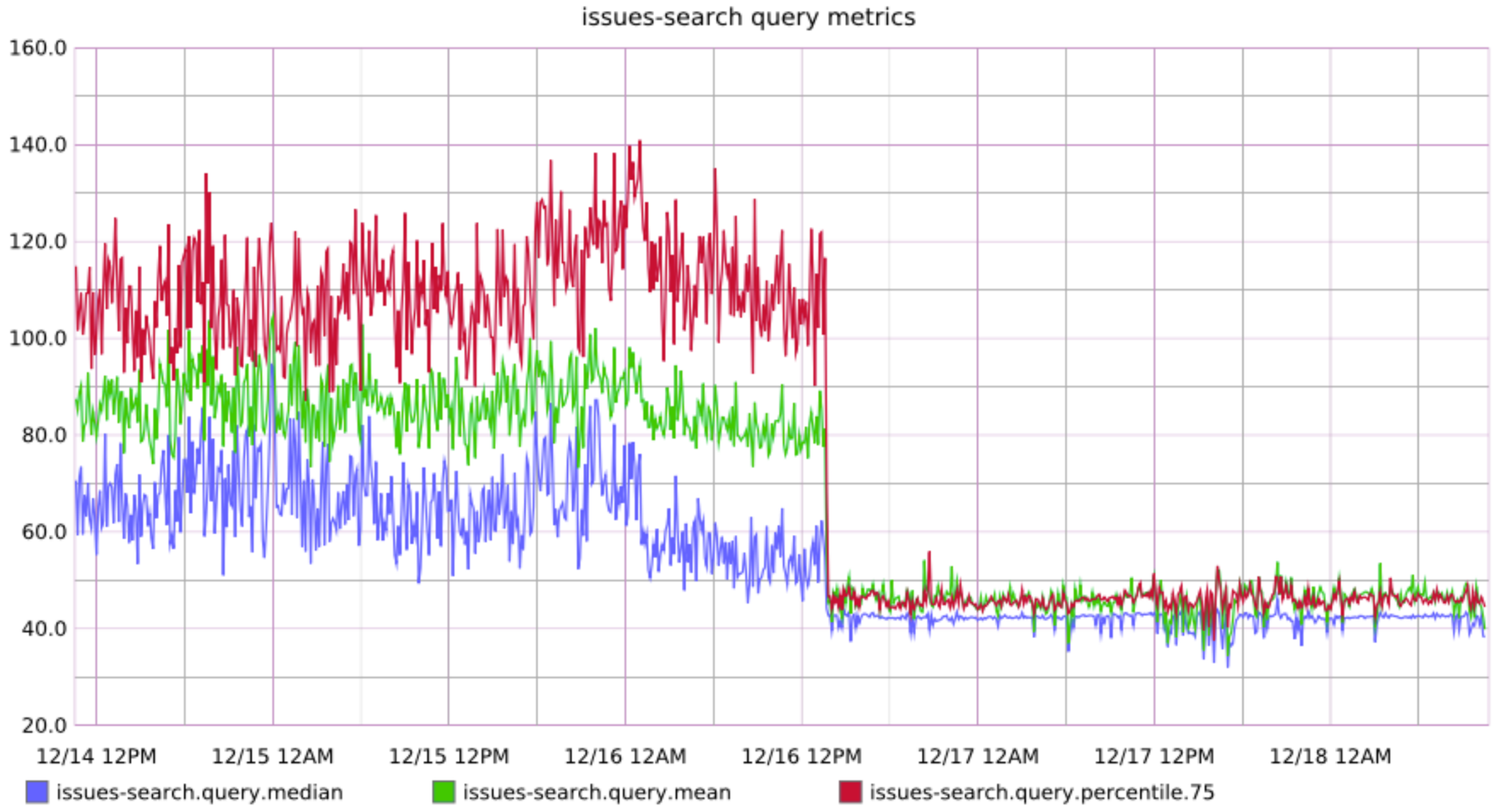

Query
Filter
Filter *
Facets

Common Issue Filters

```
author:grantr
assignee:kimchi
mentions:TwP
involves:drewr
label:bug
repo:elasticsearch/elasticsearch
created:<2015-01-01
updated:>2015-03-01
is:issue
is:open
```

Common Issue Filters

```
author:grantr
assignee:kimchi
mentions:TwP
involves:drewr
label:bug
repo:elasticsearch/elasticsearch
created:<2015-01-01
updated:>2015-03-01
is:issue
is:open
```

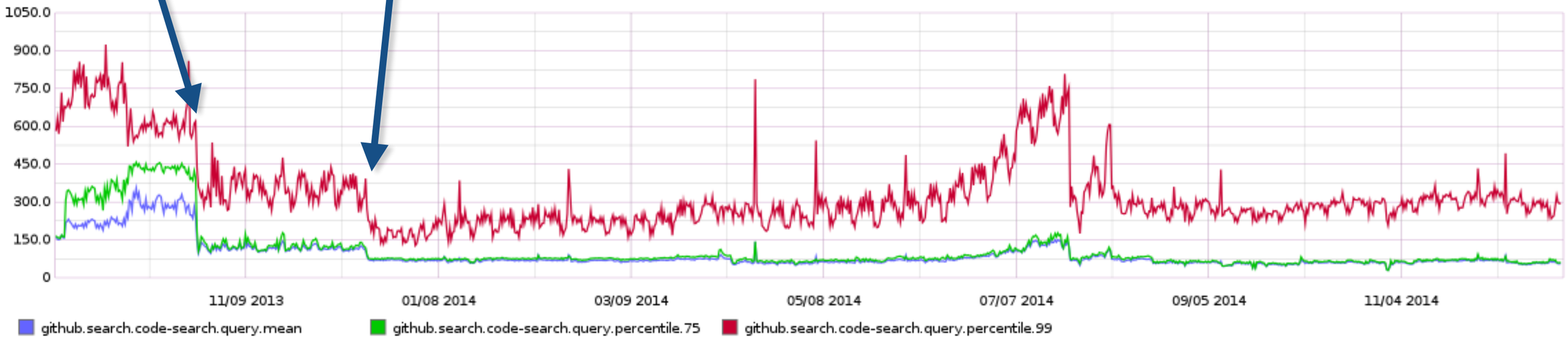

issues-search query metrics
160.0
140.0
120.0
100.0
80.0
60.0
40.0
20.0
12/14 12PM
12/15 12AM
12/15 12PM
12/16 12AM
12/16 12PM
12/17 12AM
12/17 12PM
12/18 12AM
issues-search.query.median
issues-search.query.mean
issues-search.query.percentile.75

New Queries

- We were able to look at query performance
- We got some education about filters
- We now enjoy efficient filtered queries

New Cluster

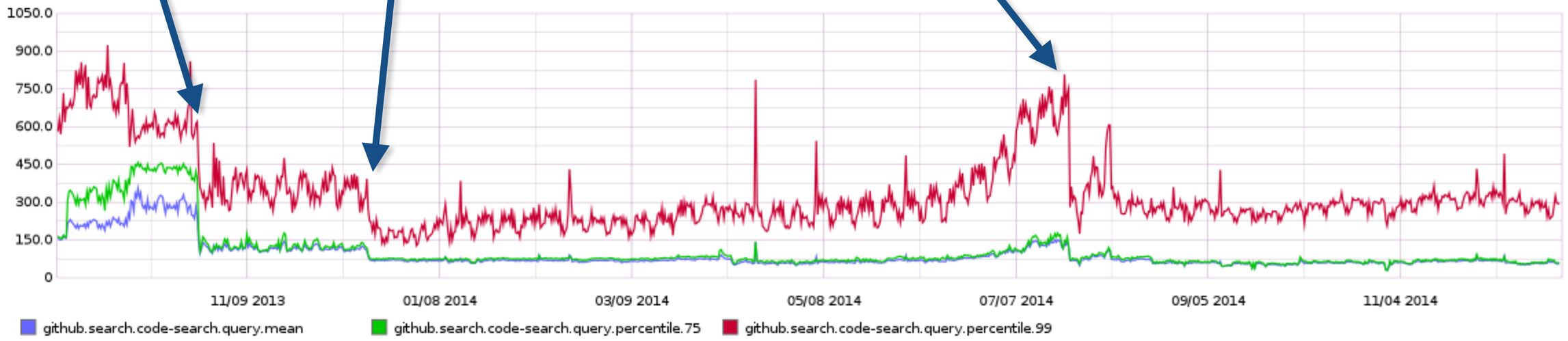
New Queries



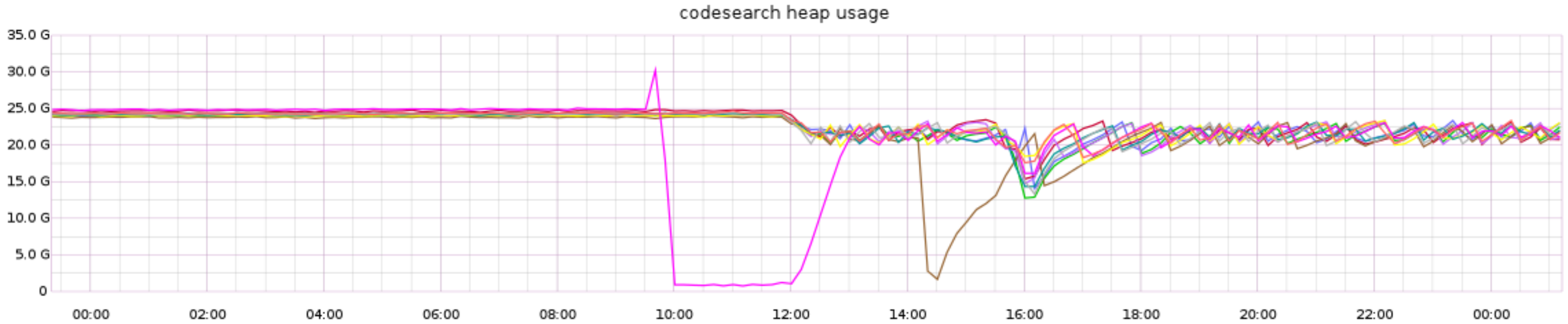
New Cluster

New Queries

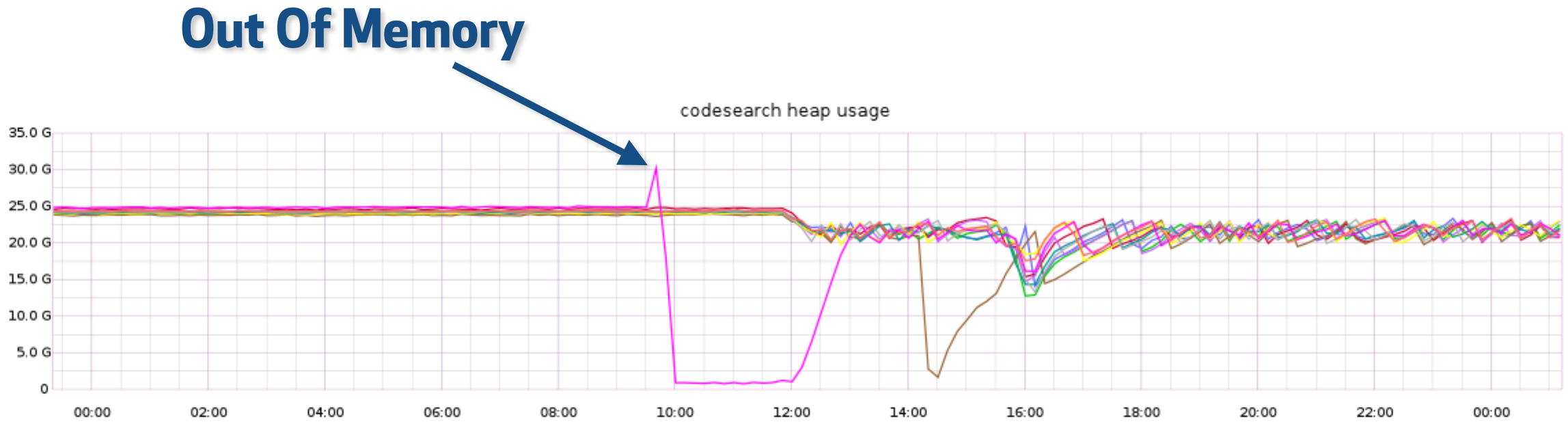
Heap Exhaustion



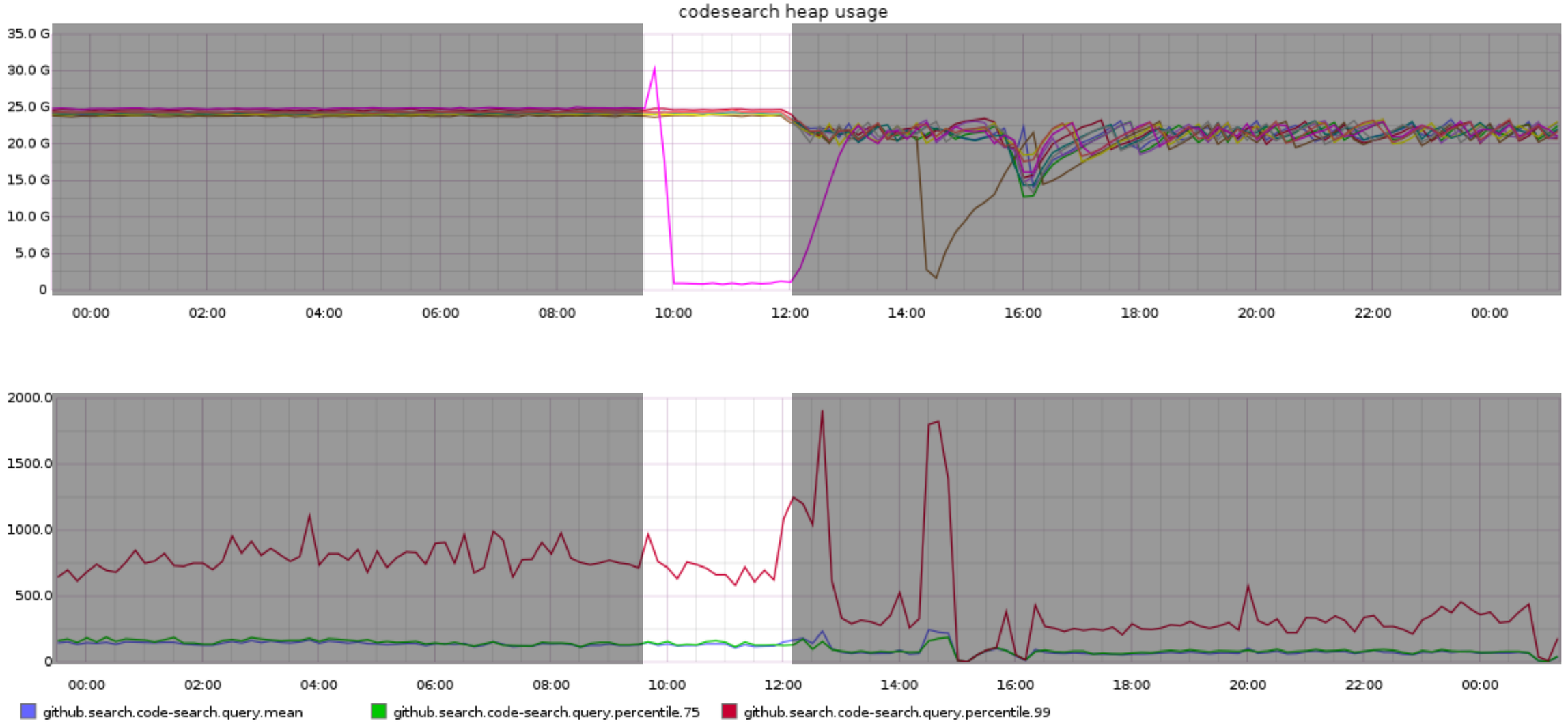
Code Search Heap Usage

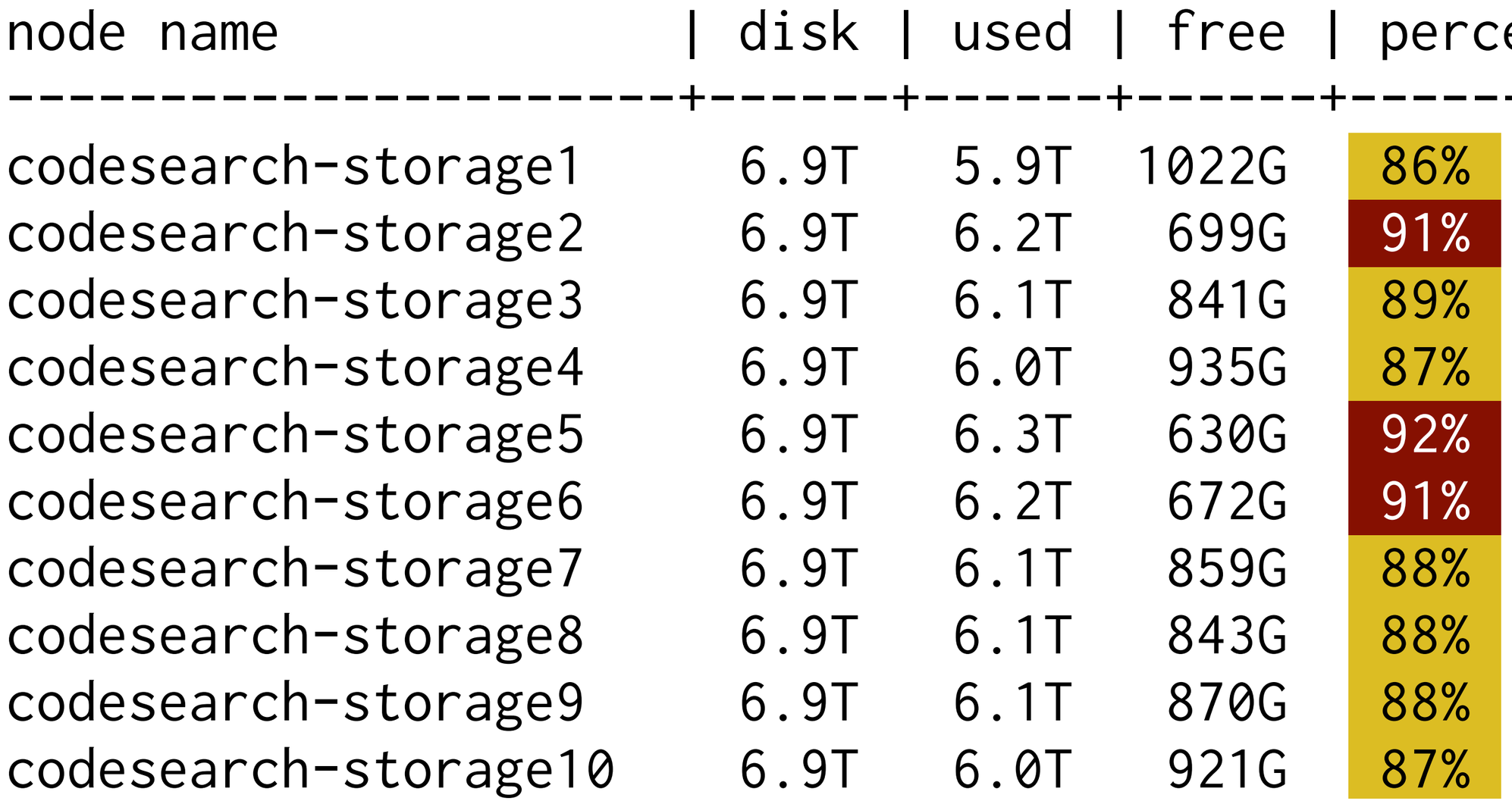


Code Search Heap Usage




codesearch heap usage
35.0 G
30.0 G
25.0 G
20.0 G
15.0 G
10.0 G
5.0 G
0
00:00
02:00
04:00
06:00
08:00
10:00
12:00
14:00
16:00
18:00
20:00
22:00
00:00
2000.0
1500.0
1000.0
500.0
0
00:00
02:00
04:00
06:00
08:00
10:00
12:00
14:00
16:00
18:00
20:00
22:00
00:00
github.search.code-search.query.mean
github.search.code-search.query.percentile.75
github.search.code-search.query.percentile.99


codesearch heap usage
35.0 G
30.0 G
25.0 G
20.0 G
15.0 G
10.0 G
5.0 G
0
00:00
02:00
04:00
06:00
08:00
10:00
12:00
14:00
16:00
18:00
20:00
22:00
00:00
2000.0
1500.0
1000.0
500.0
0
00:00
02:00
04:00
06:00
08:00
10:00
12:00
14:00
16:00
18:00
20:00
22:00
00:00
github.search.code-search.query.mean
github.search.code-search.query.percentile.75
github.search.code-search.query.percentile.99


node name | disk | used | free | percent
codesearch-storage1 6.9T 5.9T 1022G 86%
codesearch-storage2 6.9T 6.2T 699G 91%
codesearch-storage3 6.9T 6.1T 841G 89%
codesearch-storage4 6.9T 6.0T 935G 87%
codesearch-storage5 6.9T 6.3T 630G 92%
codesearch-storage6 6.9T 6.2T 672G 91%
codesearch-storage7 6.9T 6.1T 859G 88%
codesearch-storage8 6.9T 6.1T 843G 88%
codesearch-storage9 6.9T 6.1T 870G 88%
codesearch-storage10 6.9T 6.0T 921G 87%

Add Capacity

Add Capacity

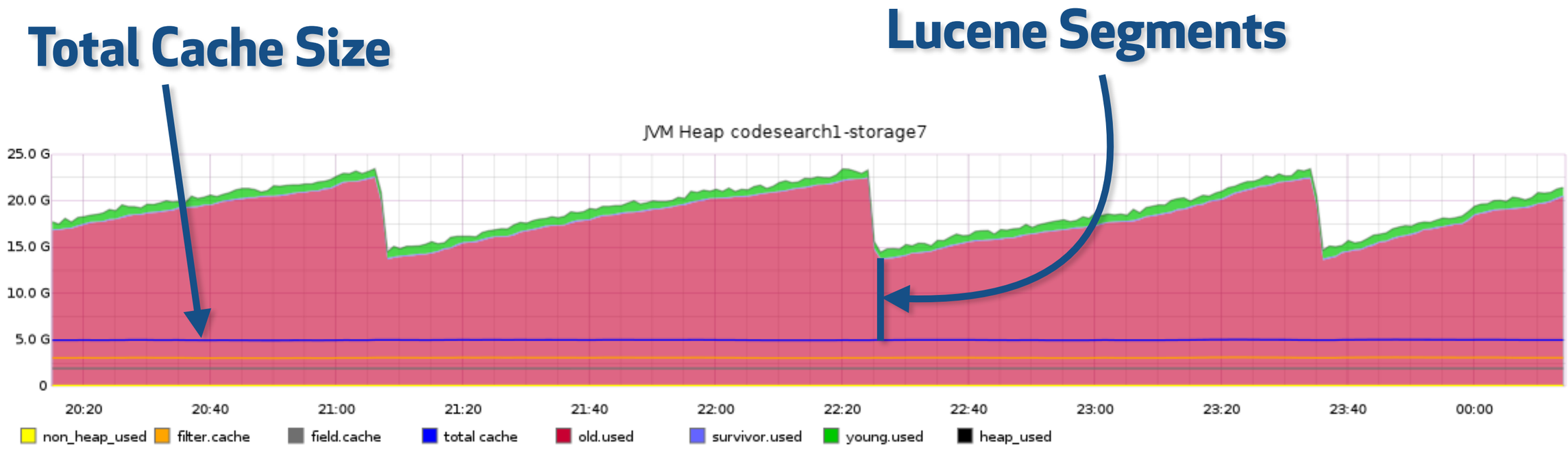
Logical
Volume
Manager


7.6 TB
6.9 TB

7.6 TB

How Did This Happen?

JVM Heap Usage



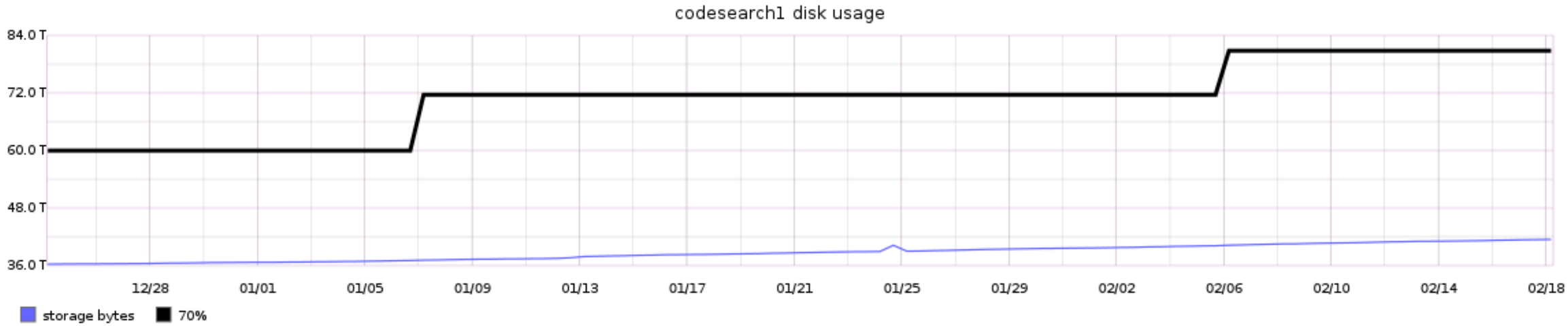
Prevention


Nagios


PAGERDUTY

/es forecast disk codesearch1

codesearch1 will reach 70% disk usage
in 302 days (2015-12-31) with 93% confidence



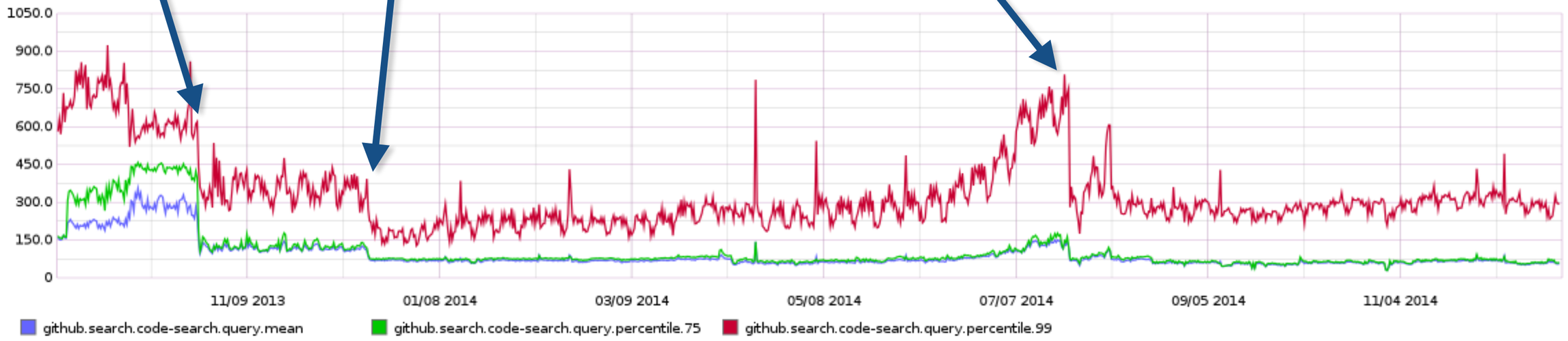
Heap Exhaustion

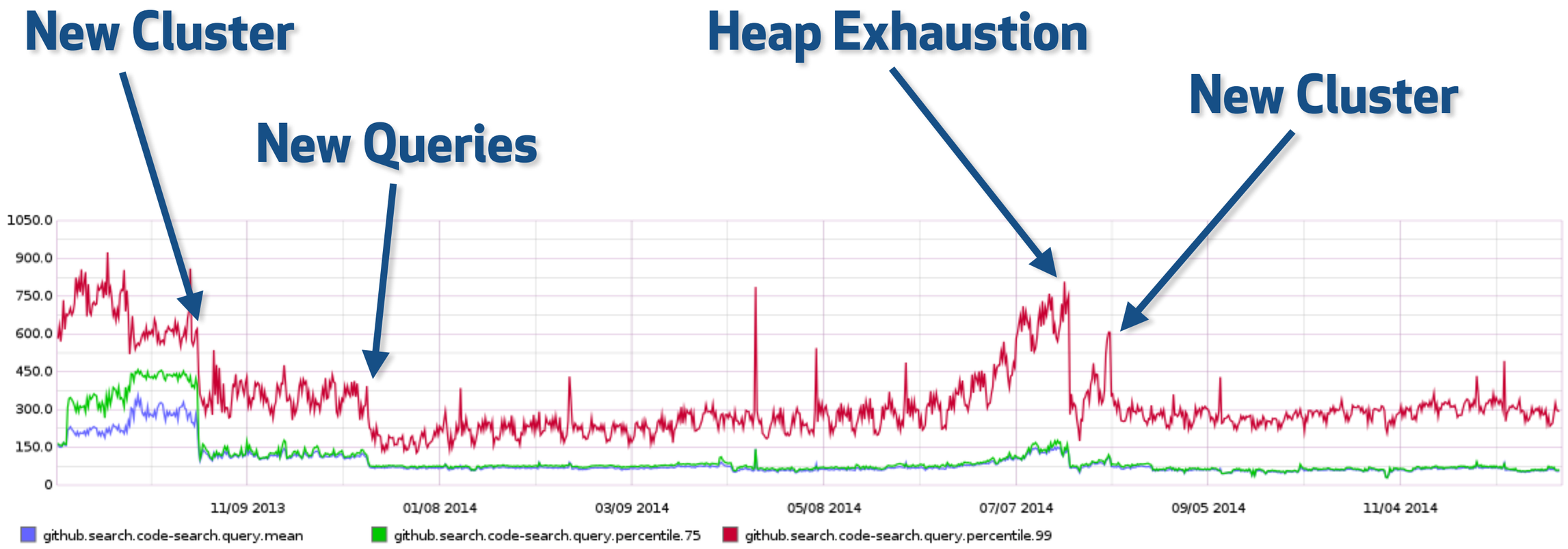
- We were ignoring key metrics
- We added alerts for key metrics
- We created tools to forecast growth

New Cluster

New Queries

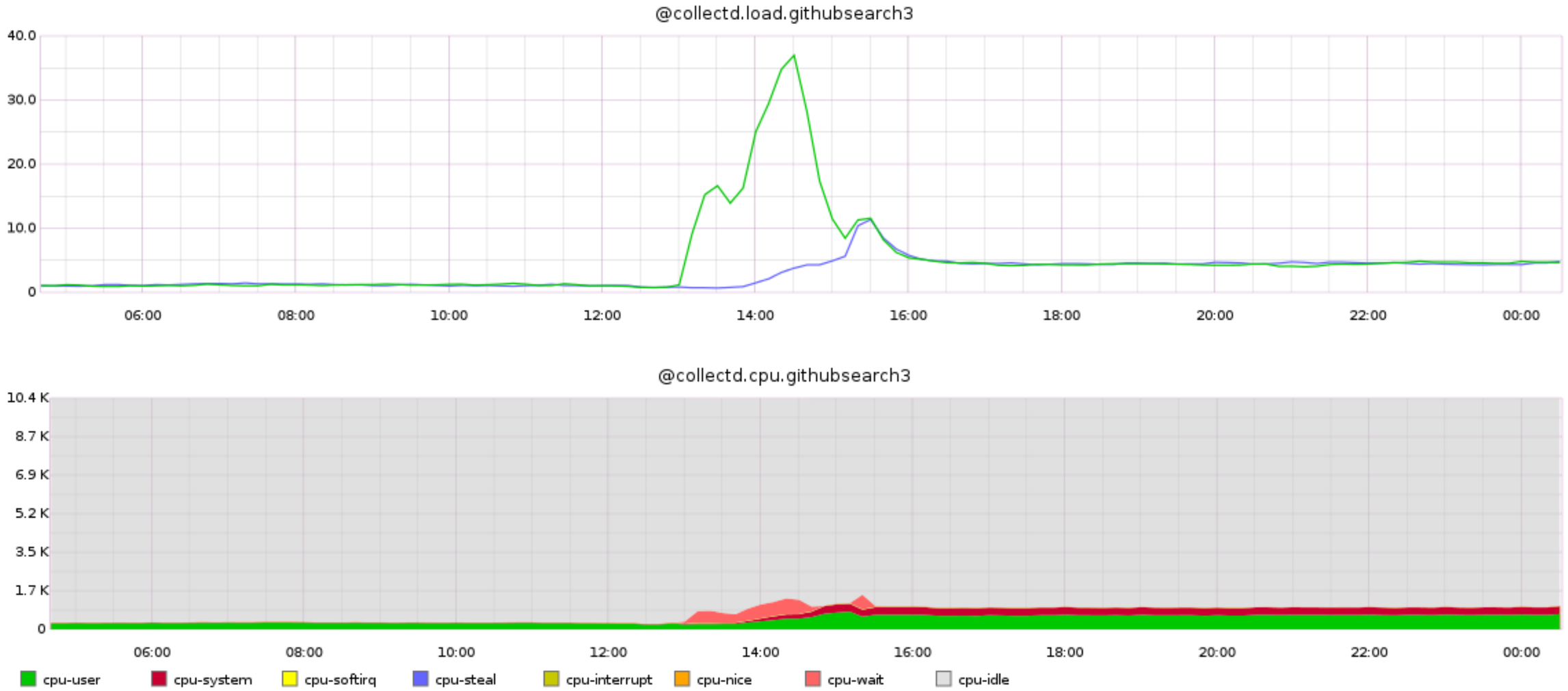
Heap Exhaustion



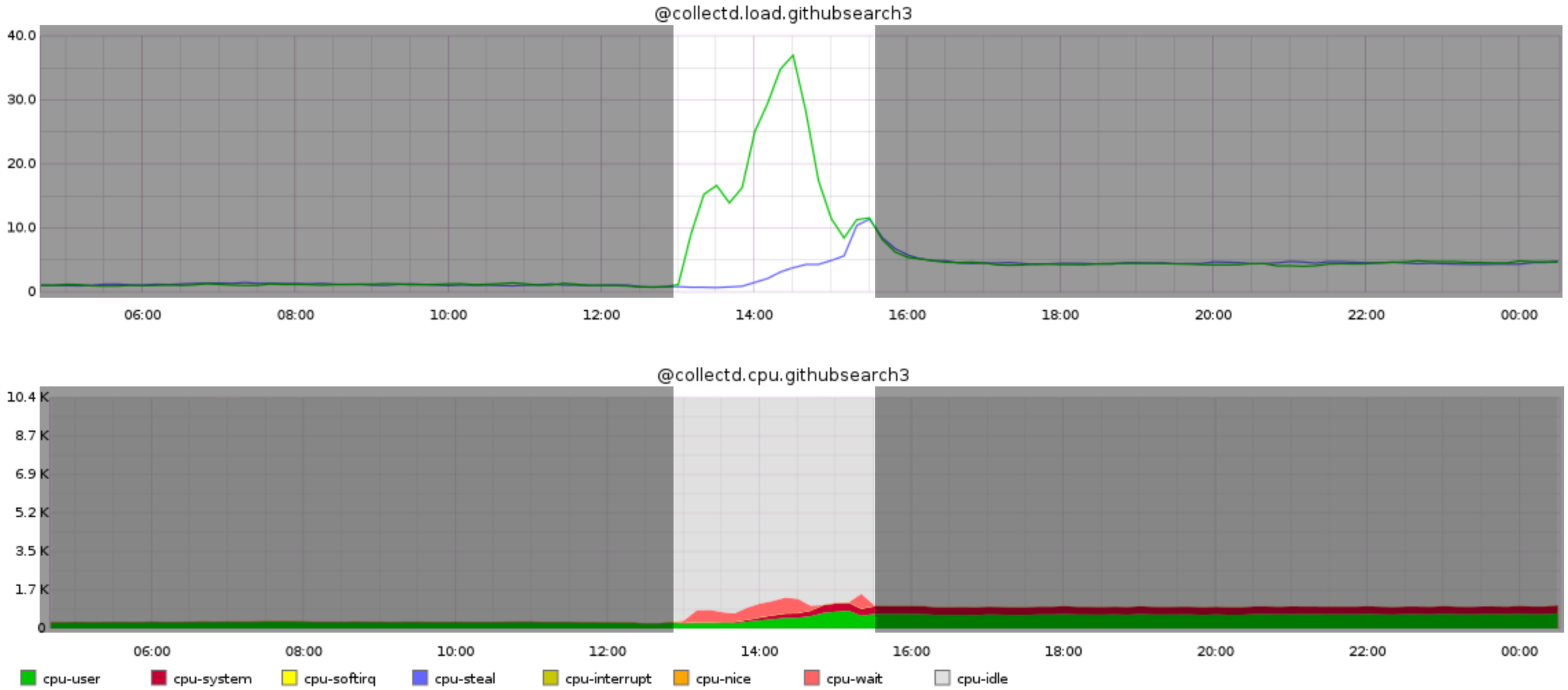

New Cluster
New Queries
Heap Exhaustion
New Cluster
1050.0
900.0
750.0
600.0
450.0
300.0
150.0
0
11/09 2013
01/08 2014
03/09 2014
05/08 2014
07/07 2014
09/05 2014
11/04 2014
github.search.code-search.query.mean
github.search.code-search.query.percentile.75
github.search.code-search.query.percentile.99

One More Story

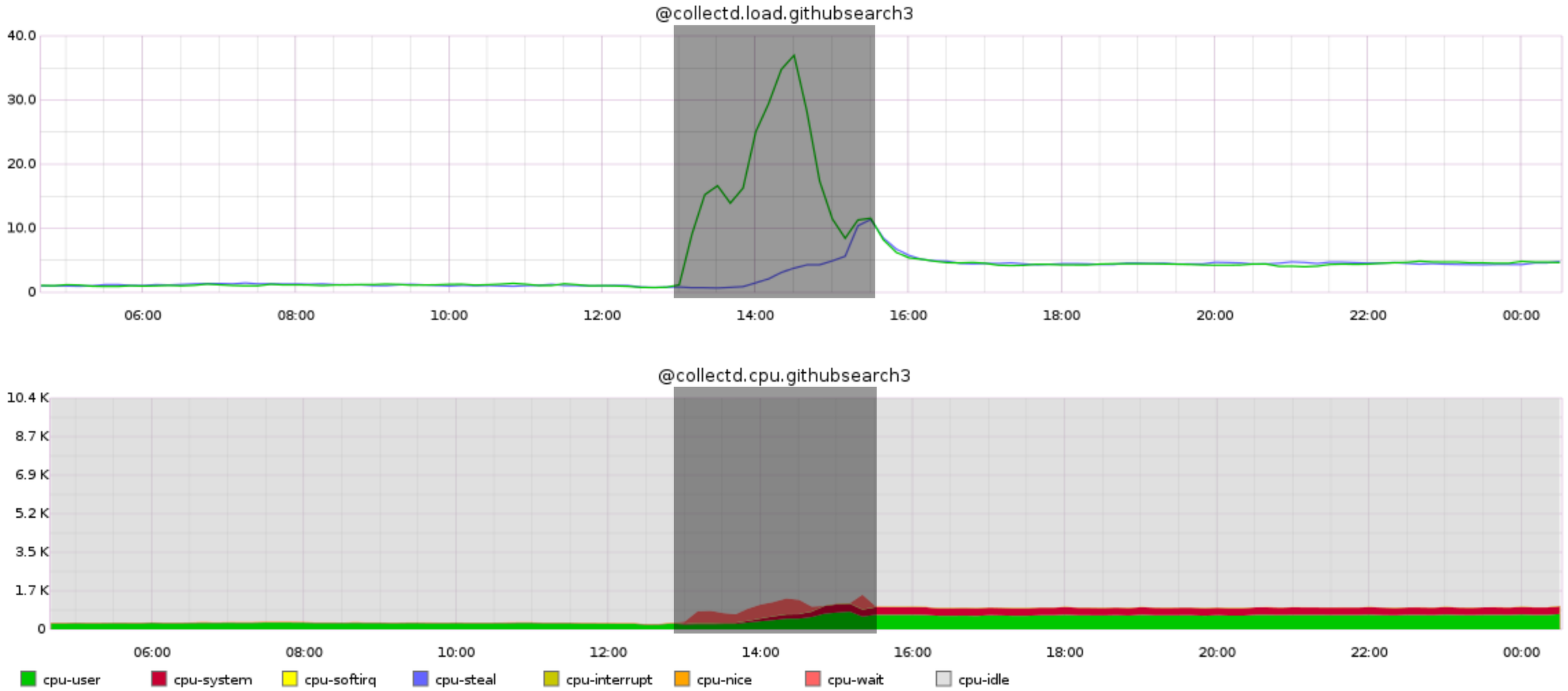
Upgrade to ES 1.4.2



Upgrade to ES 1.4.2



Upgrade to ES 1.4.2



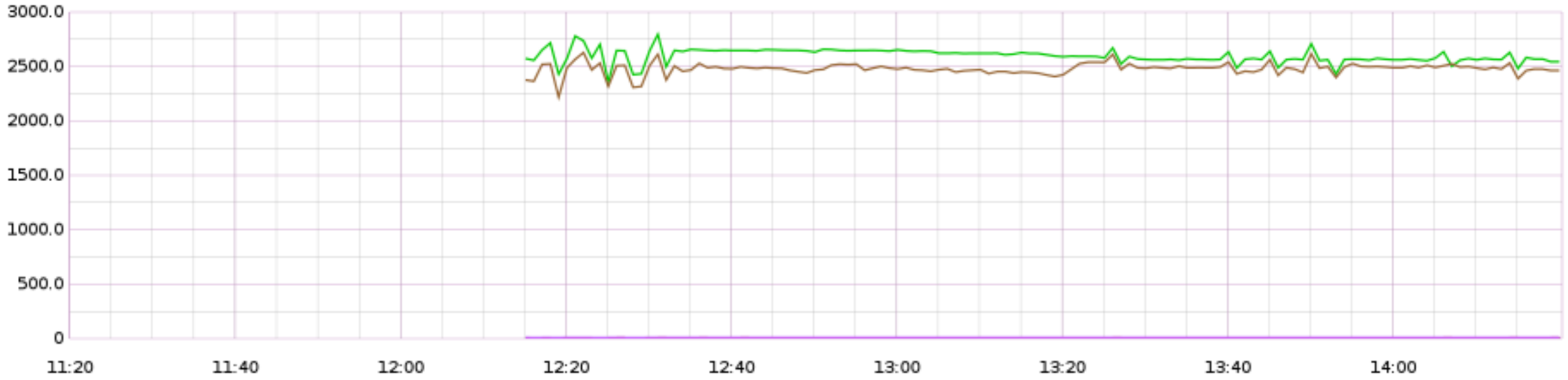
Hot Threads

```
97.4% (487.1ms out of 500ms) cpu usage by thread 'elasticsearch[githubsearch3-storage1-cp1-prd][management][T#2]'
 9/10 snapshots sharing following 9 elements
   org.elasticsearch.action.admin.indices.stats.ShardStats.<init>(ShardStats.java:49)

97.3% (486.3ms out of 500ms) cpu usage by thread 'elasticsearch[githubsearch3-storage1-cp1-prd][management][T#3]'
 2/10 snapshots sharing following 20 elements
   java.io.UnixFileSystem.getLength(Native Method)

96.4% (482.1ms out of 500ms) cpu usage by thread 'elasticsearch[githubsearch3-storage1-cp1-prd][management][T#4]'
 2/10 snapshots sharing following 19 elements
   org.apache.lucene.store.FSDirectory.listAll(FSDirectory.java:223)
```

Management Threads

Amen Sampler

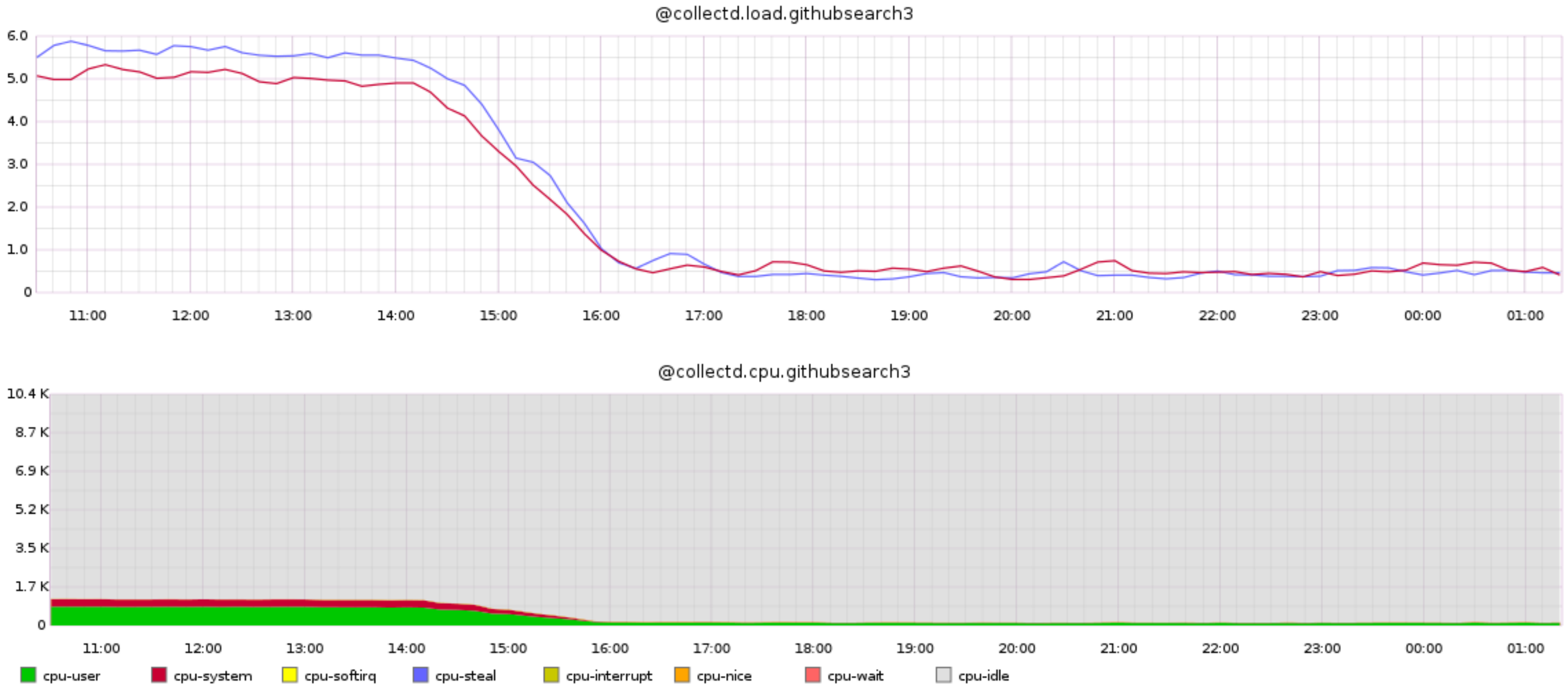
/_nodes/_local/stats



HAProxy Change

~~/_nodes/_local/stats~~ ➠ /

HAProxy Change

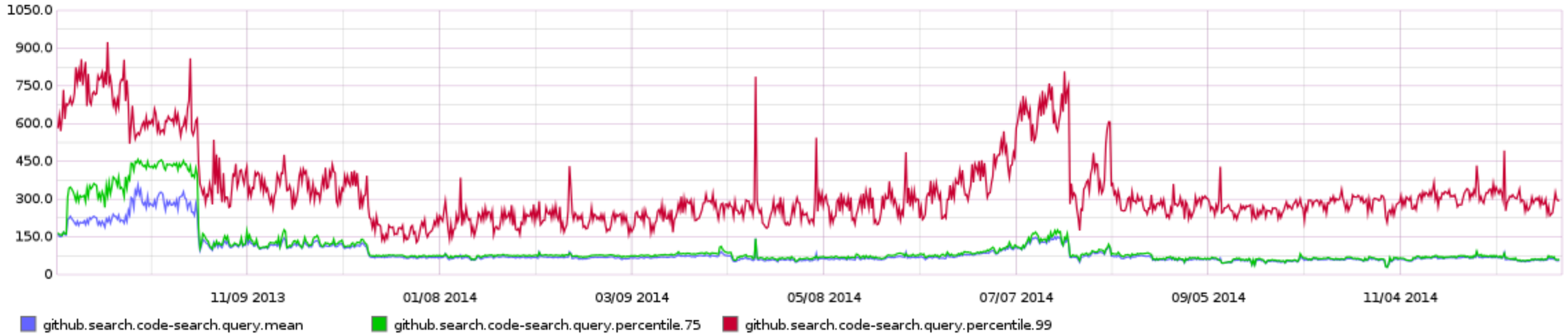


Upgrade to ES 1.4.2

- We were missing some important metrics
- We have an entire ecosystem
- We have confidence in ES 1.4.2

- Where have we come from
- What we are doing now
- Where we are going

Where have we come from

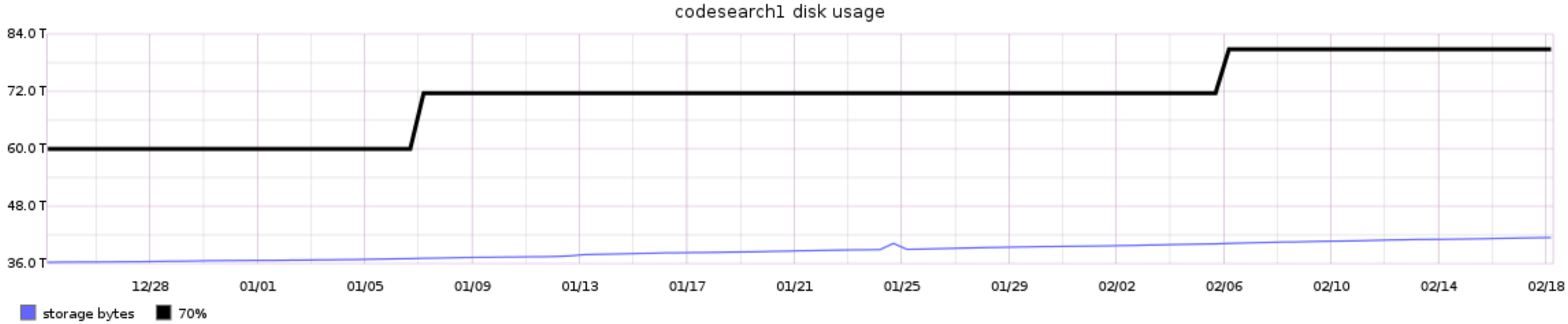


What are we doing now

Where are we going

/es forecast disk codesearch1

codesearch1 will reach 70% disk usage
in 302 days (2015-12-31) with 93% confidence



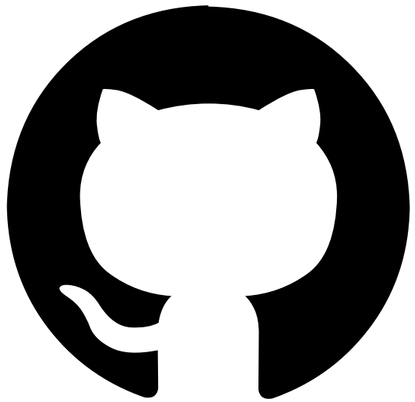
GitHub Search Team

Grant Rodgers

github.com/grantr

Tim Pease

github.com/TwP



The End