

# Active Data

<http://activedata.allizom.org>

Kyle Lahnakoski  
Engineering Productivity



Lightning talk on...

# Active Data

<http://activedata.allizom.org>

Kyle Lahnakoski  
Engineering Productivity



# What is ActiveData?

- A public web service...

```
$ curl http://activedata.allizom.org/query  
-X POST  
-d "{\"from\":\"unittest\"}"
```



# What is ActiveData?

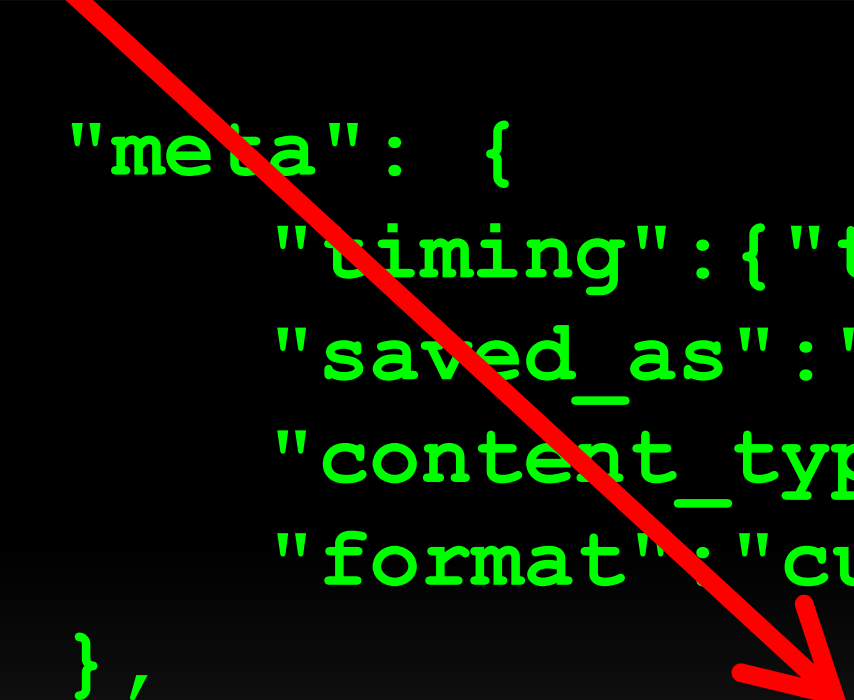
- A public web service...
- providing a query interface...

```
{  
  "select": {  
    "aggregate": "count"  
  },  
  "from": "unittest"  
}
```



# What is ActiveData?

- A public web service...
- providing a query interface...
- to over a billion test results...



```
{
  "meta": {
    "timing": {"total": 0.35},
    "saved_as": "3iK9T6sG",
    "content_type": "application/json",
    "format": "cube"
  },
  "data": {"count": 2898920602}
}
```

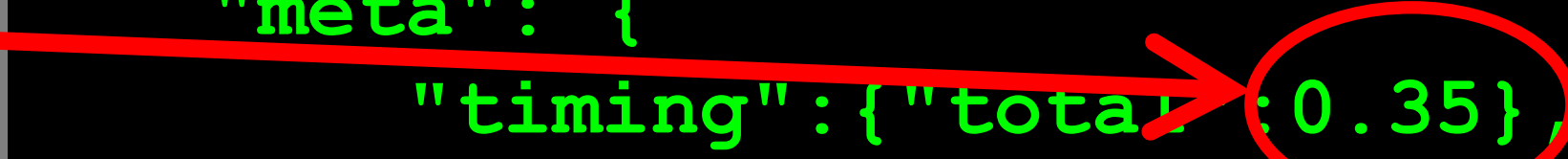


# What is ActiveData?

- A public web service...
- providing a query interface...
- to over a billion test results...

• Quickly!

```
{
  "meta": {
    "timing": {"total": 0.35},
    "saved_as": "3iK9T6sG",
    "content_type": "application/json",
    "format": "cube"
  },
  "data": {"count": 2898920602}
}
```





# What is ActiveData?

ActiveData stores JSON documents

```
{
  "run": {
    "timestamp": 1448535906
  },
  "result": {
    "test": "test_getAll.js",
    "duration": 3.016
  }
}
```

```
{
  "run.timestamp": 1448535906,
  "result.test": "test_getAll.js",
  "result.duration": 3.016
}
```

ElasticSearch does the heavy lifting as an in-memory columnar data store



# What is ActiveData?

| result.<br>duration | result.<br>test | run.<br>timestamp |
|---------------------|-----------------|-------------------|
| 3.016               | test_getAll.js  | 1448535906        |
| 1.601               | test_660156.js  | 1448536787        |
| 1.718               | test_getAll.js  | 1448536787        |
| 1.780               | test_660156.js  | 1448536787        |
| 2.127               | test_getAll.js  | 1448536787        |
| 3.188               | test_getAll.js  | 1448536787        |
| 2.272               | test_660156.js  | 1448536787        |
| 2.247               | test_660156.js  | 1448536787        |
| 1.730               | test_660156.js  | 1448499043        |
| 2.045               | test_getAll.js  | 1448499043        |
| 1.571               | test_660156.js  | 1448499043        |

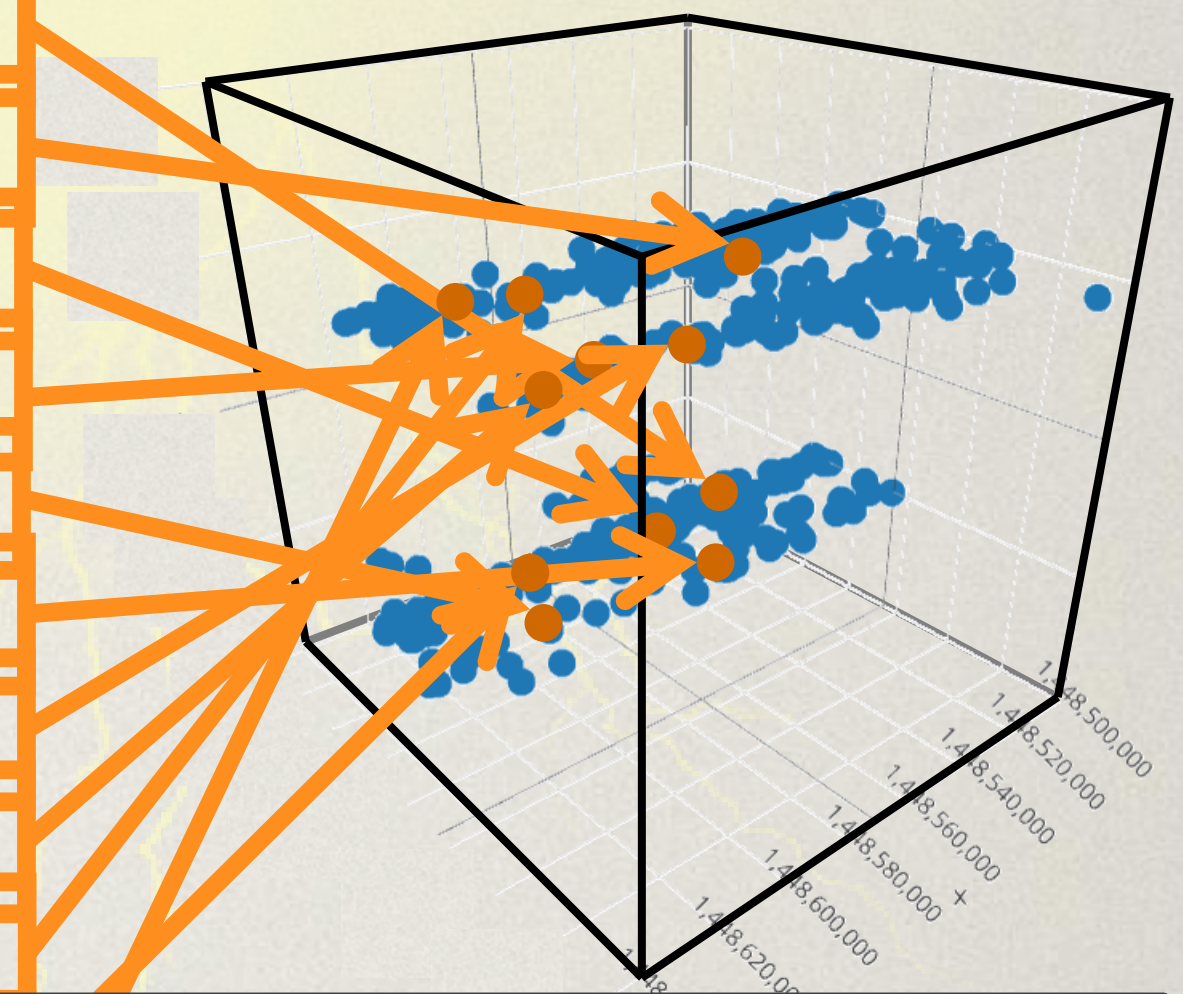
```
{  
  "run.timestamp": 1448535906,  
  "result.test": "test_getAll.js",  
  "result.duration": 3.016  
}
```

ActiveData queries documents as if they are records in a table



# What is ActiveData?

| result.<br>duration | result.<br>test | run.<br>timestamp |
|---------------------|-----------------|-------------------|
| 3.016               | test_getAll.js  | 1448535906        |
| 1.601               | test_660156.js  | 1448536787        |
| 1.718               | test_getAll.js  | 1448536787        |
| 1.780               | test_660156.js  | 1448639419        |
| 2.127               | test_getAll.js  | 1448639419        |
| 3.188               | test_getAll.js  | 1448637767        |
| 2.272               | test_660156.js  | 1448643888        |
| 2.247               | test_660156.js  | 1448545374        |
| 1.730               | test_660156.js  | 1448499043        |
| 2.045               | test_getAll.js  |                   |
| 1.571               | test_660156.js  |                   |

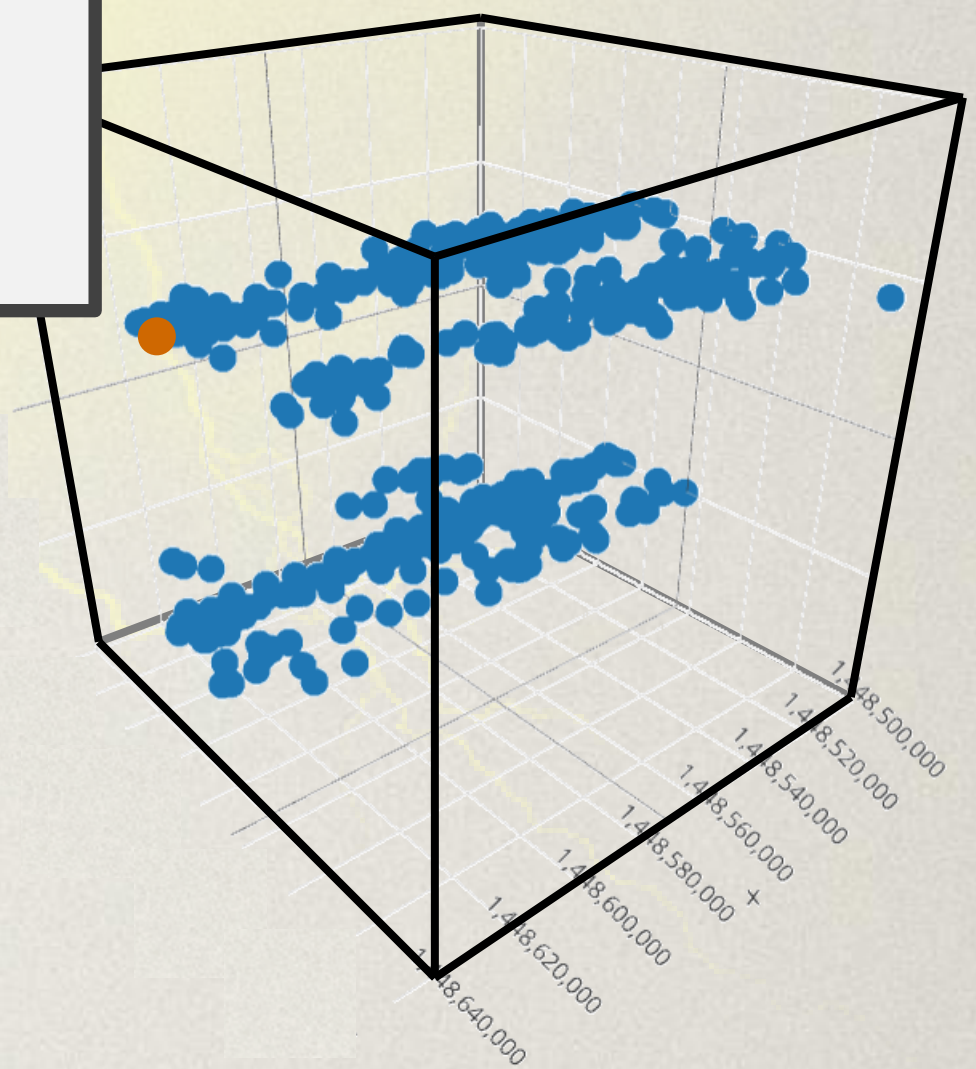
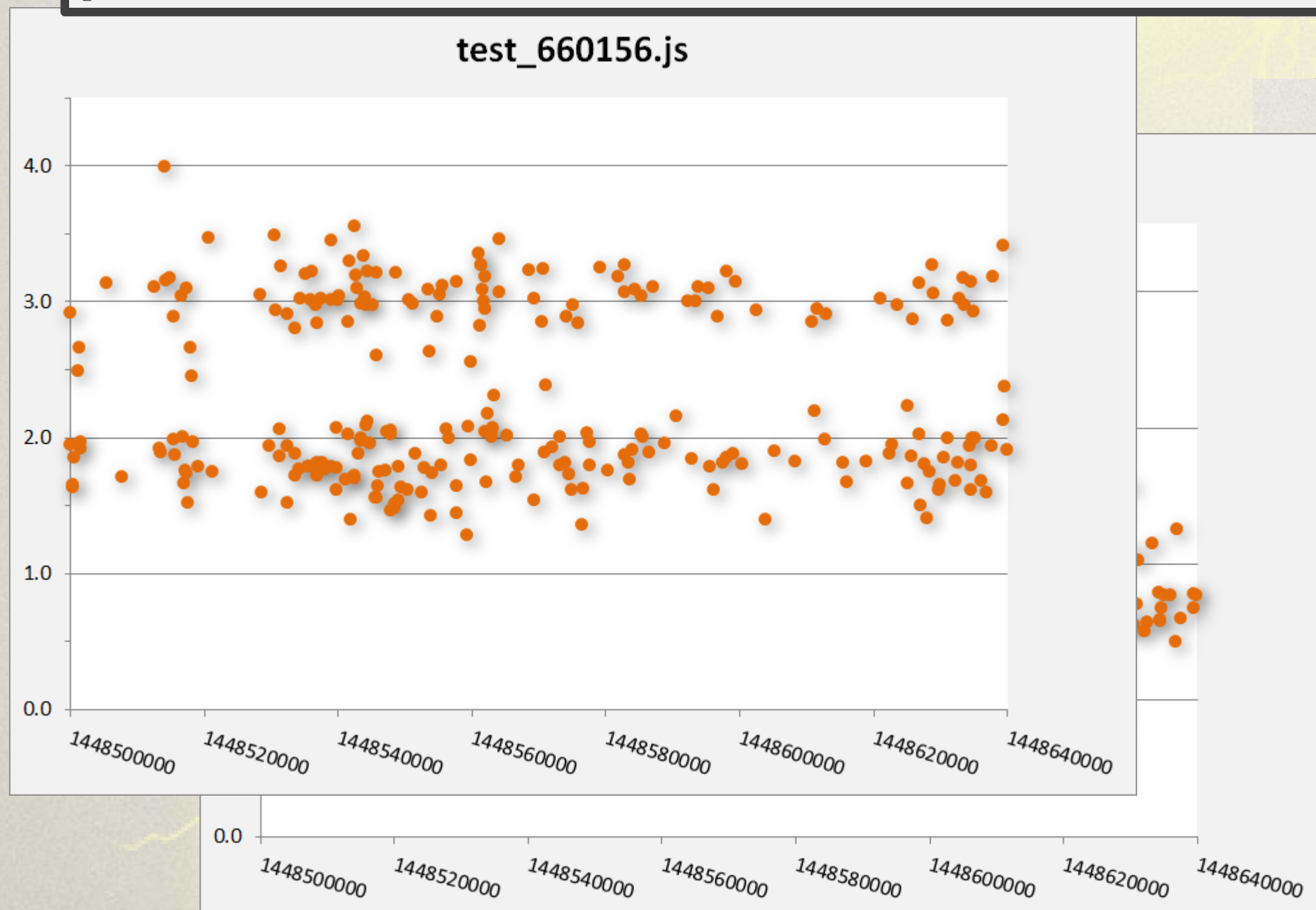


Even better, every column can be queried like an edge in a data cube; each row is a point in the cube



# What is ActiveData?

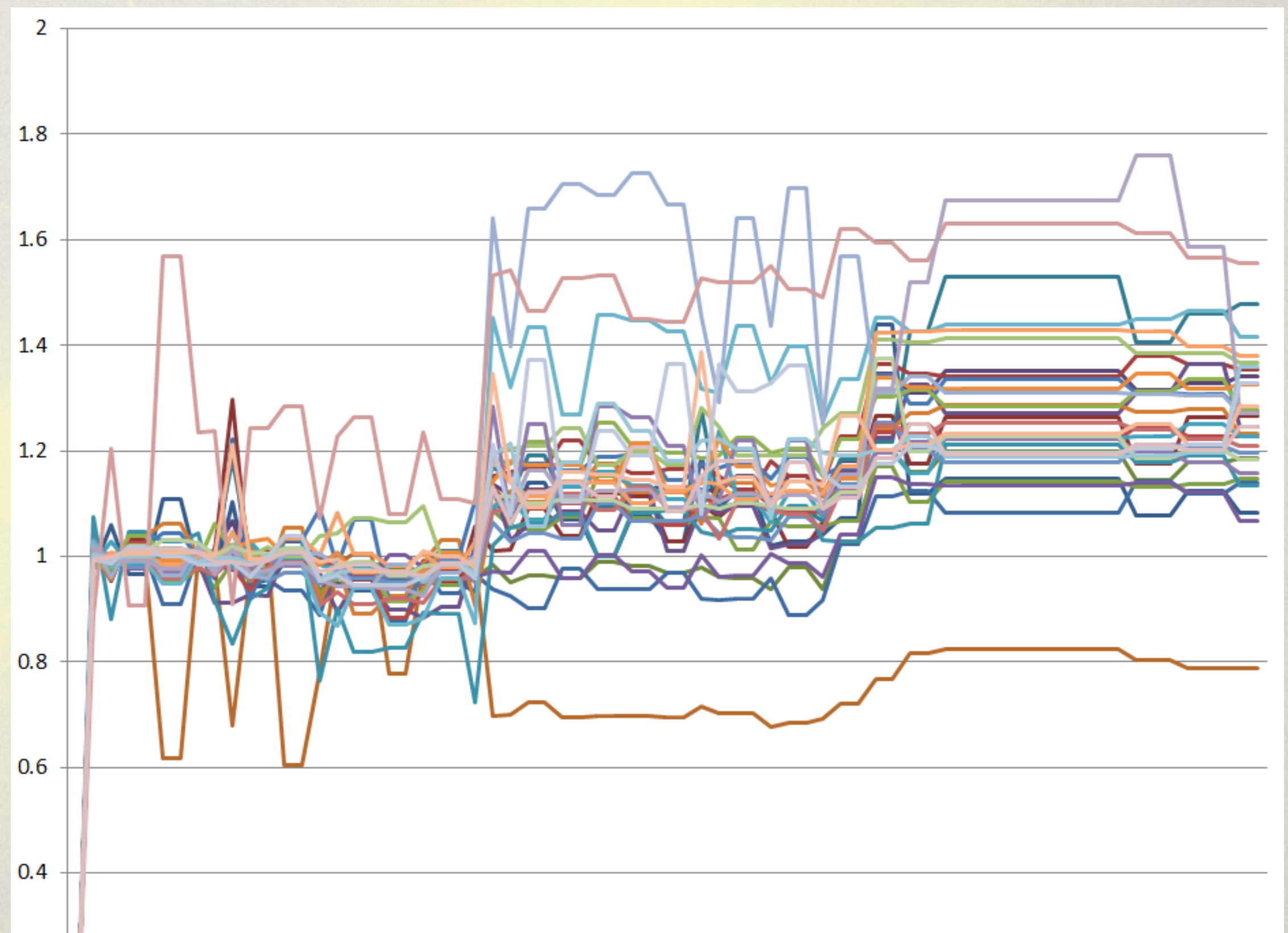
Slice the cube along the dimensions for humane presentation





# Examples

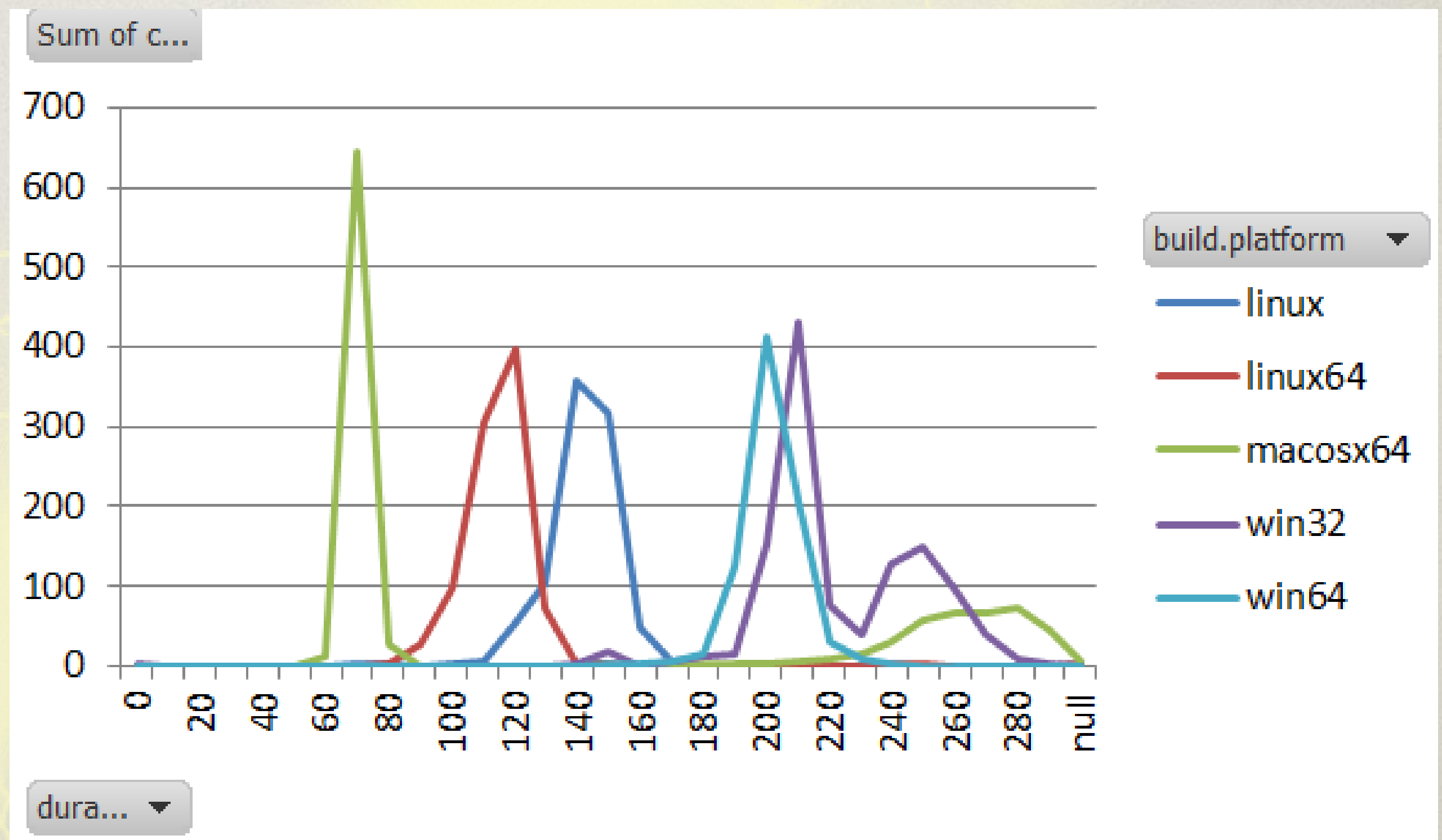
- Identify change in test times





# Examples

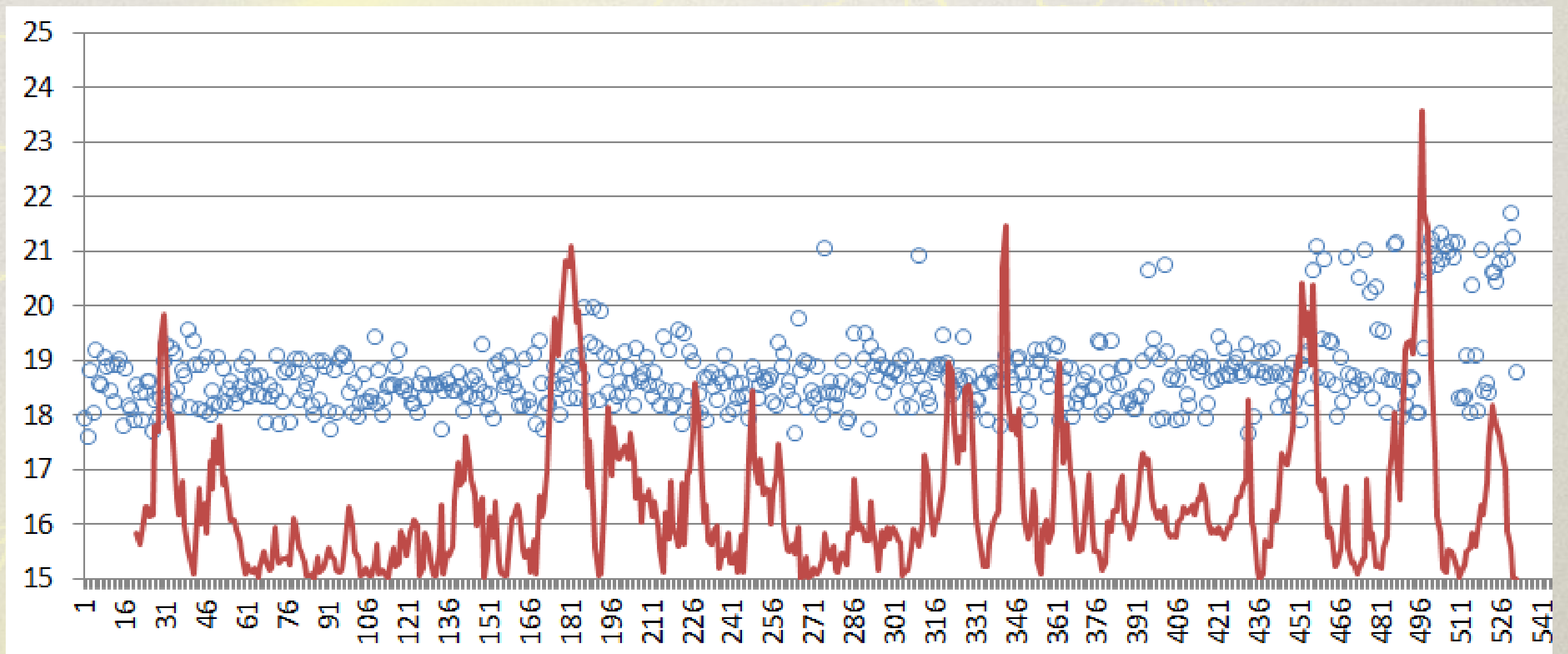
- Identify change in test times
- Test-time distributions





# Examples

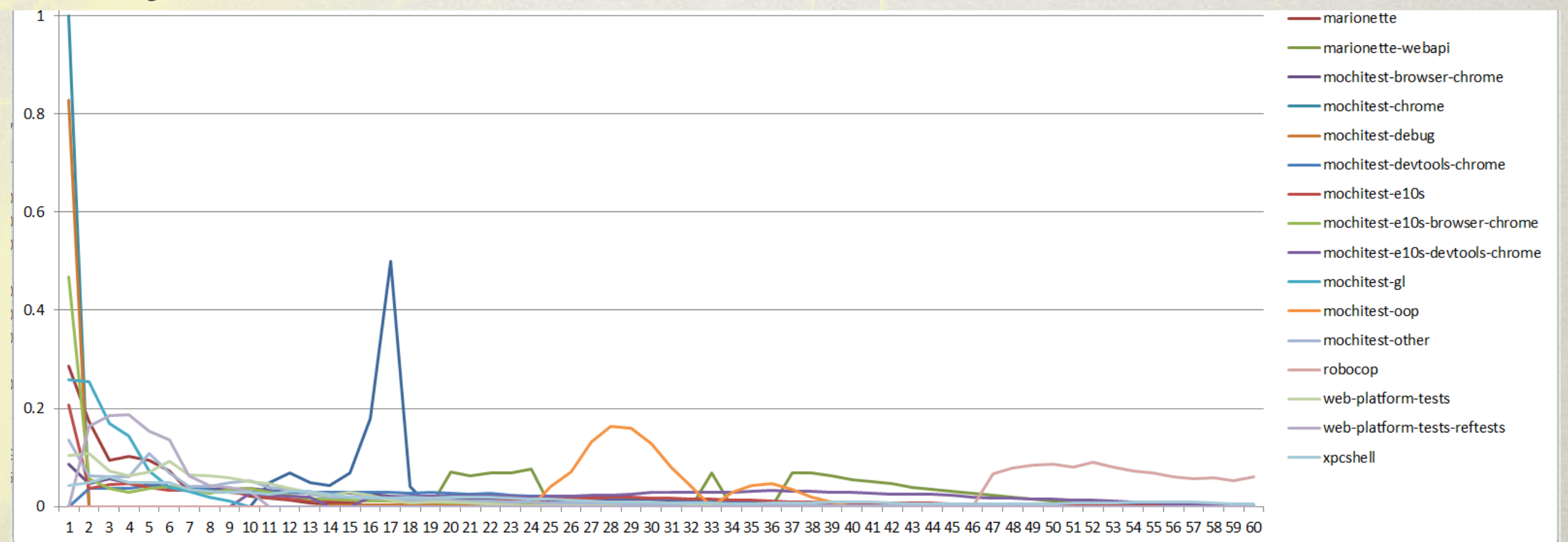
- Identify change in test times
- Test time distributions
- Visualize perf regression





# Examples

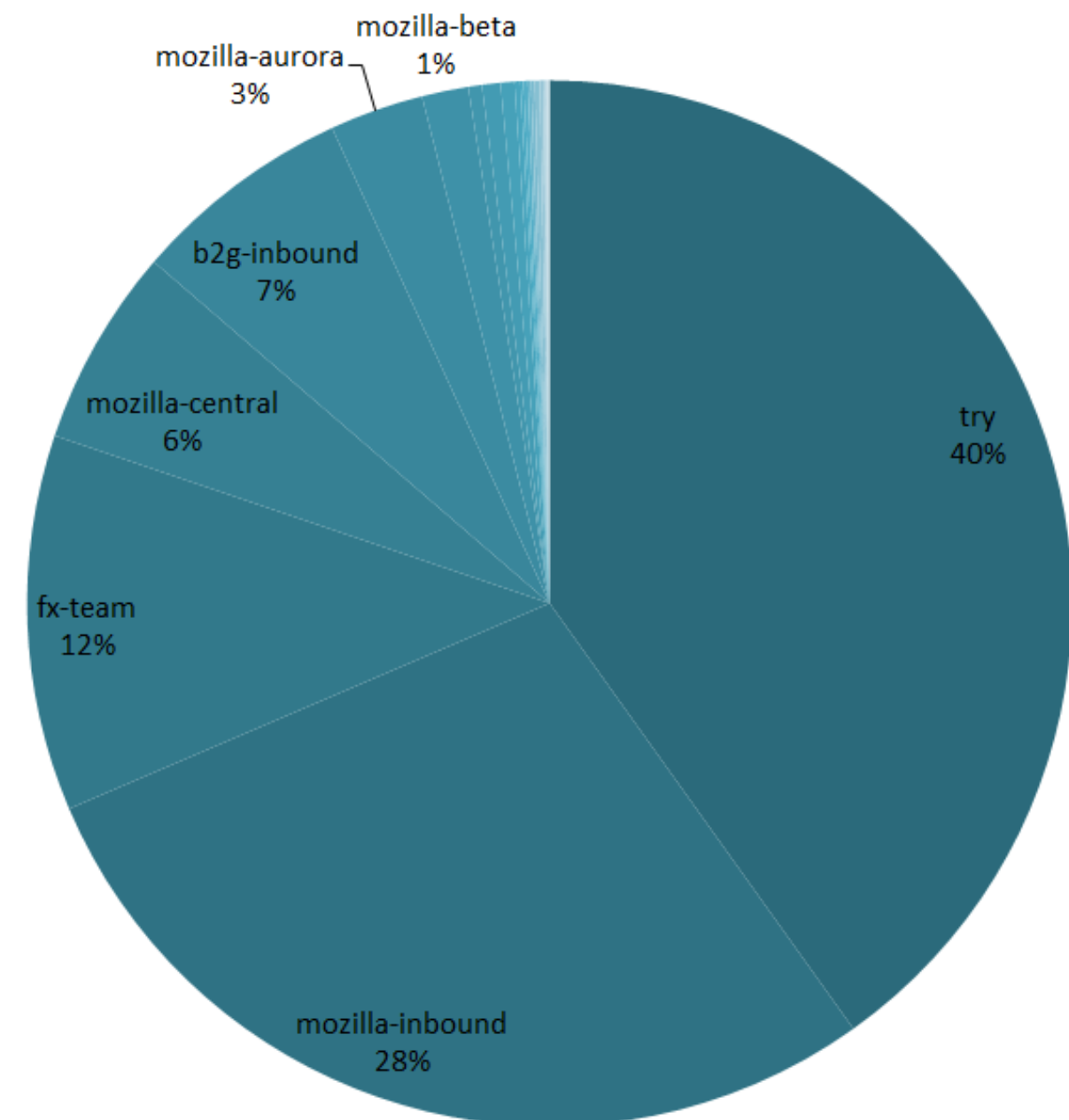
- Identify change in test times
- Test time distributions
- Visualize perf regression
- Fail rate by time into suite





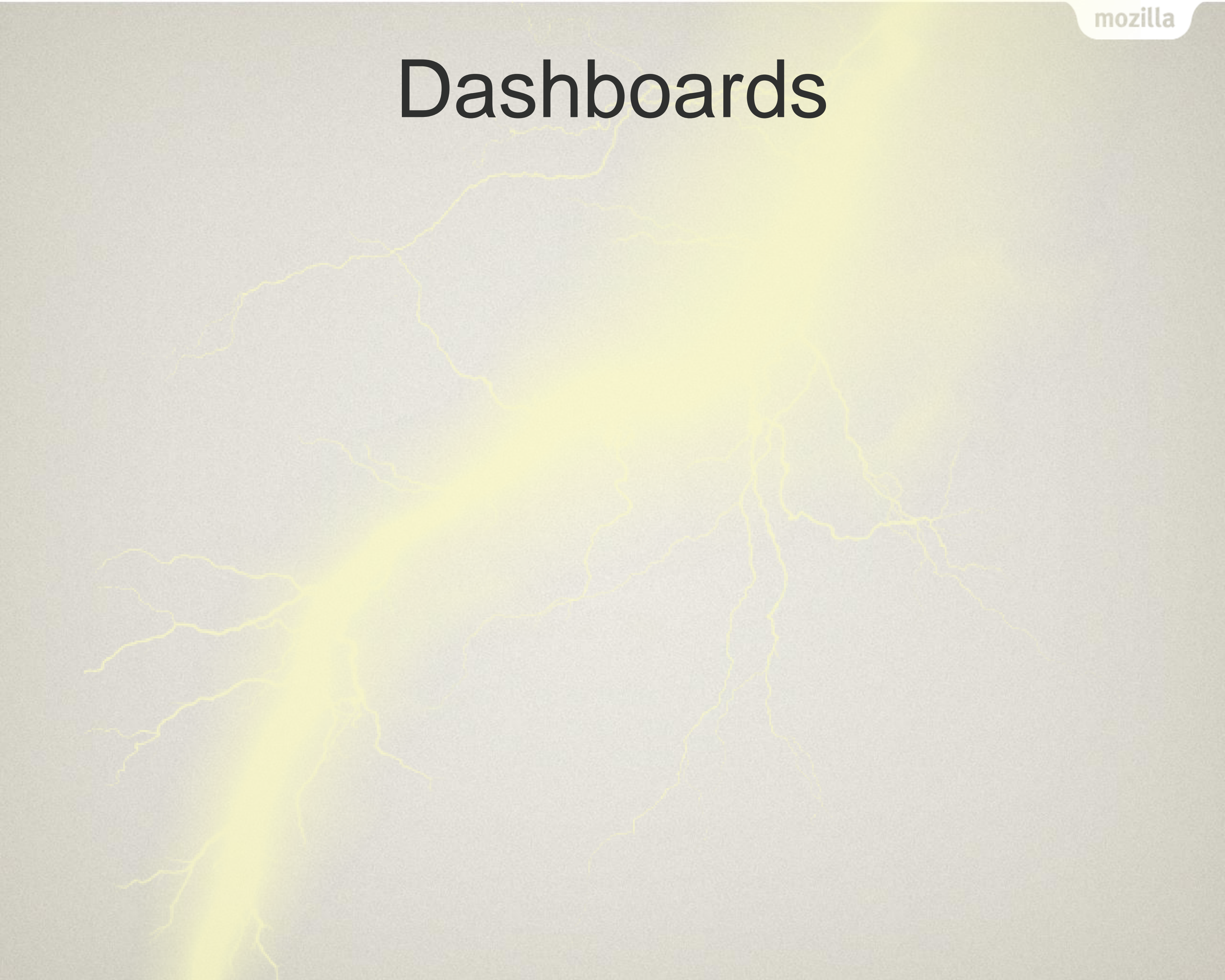
# Examples

- Identify change in test times
- Test time distributions
- Visualize perf regressions
- Fail rate by time into success
- Wall time by Branch





# Dashboards



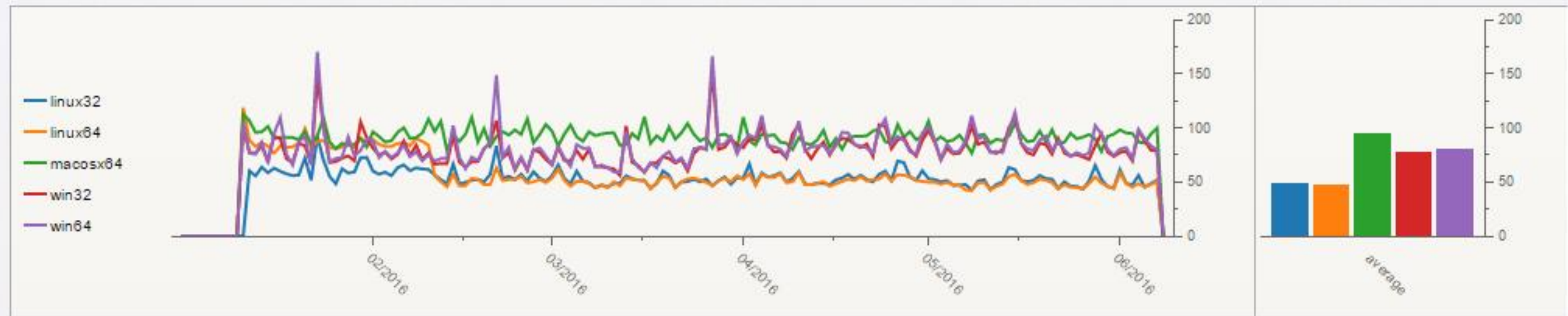


# Dashboards

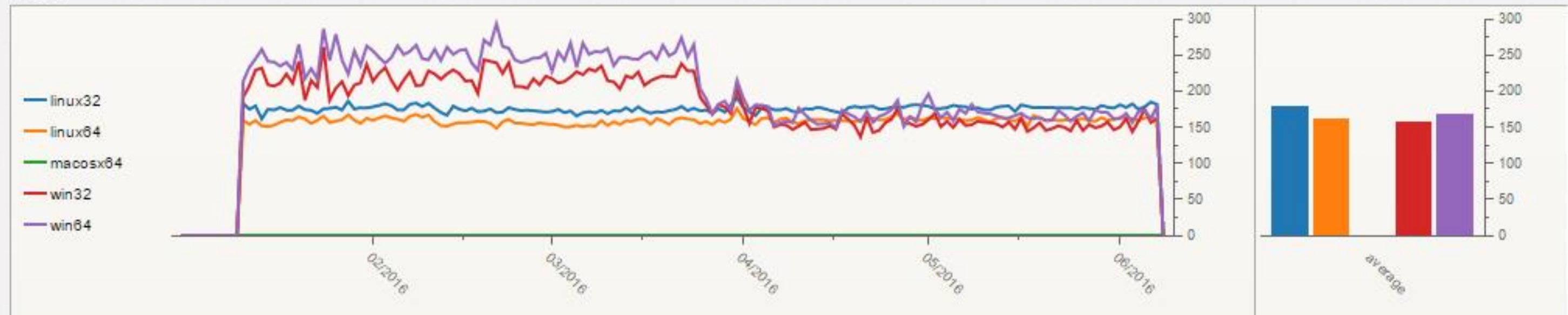
## Firefox Build Times

Done

Standard



PGO





# Dashboards

## Reduced Build Times

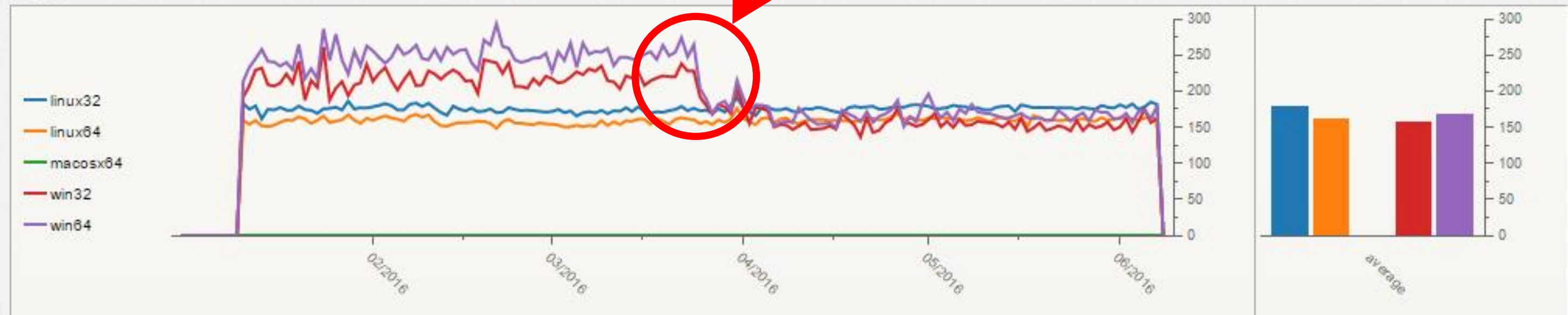
### Firefox Build Times

Done

Standard



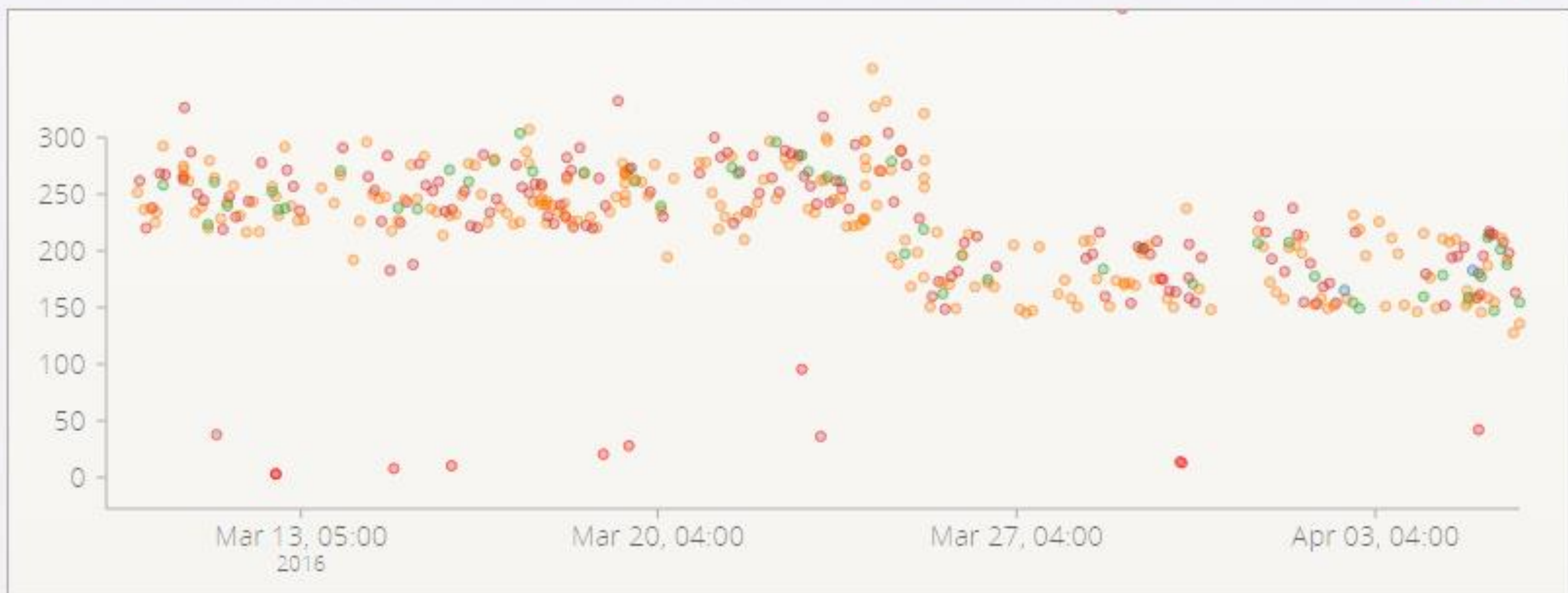
PGO





# Dashboards

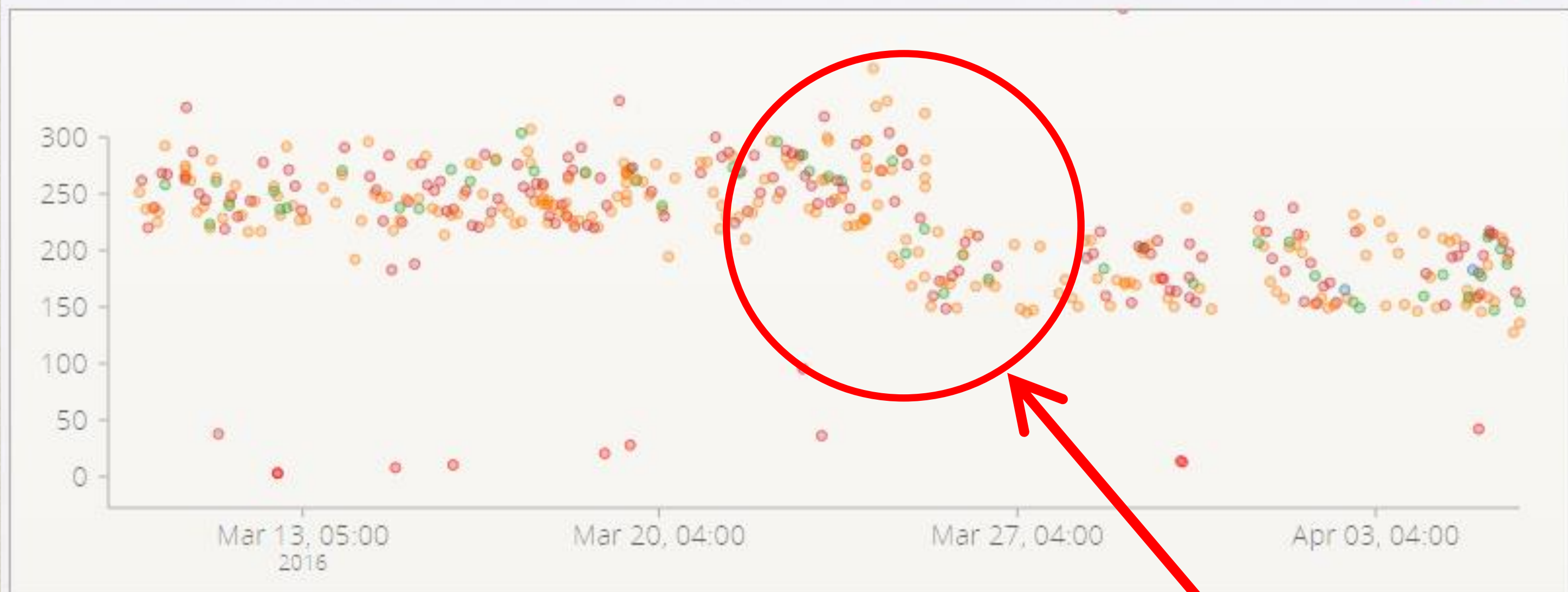
## PGO Firefox Windows64 Build Times Done





# Dashboards

## PGO Firefox Windows64 Build Times Done



**WIN!**



# Dashboards

Done

View average **test** timings for each step in BuildBot, or MozHarness. Filter by pool, platform or suite.

Only steps executed over 10% of the time are shown here. It is important you select a suite to get a reasonable estimate of the steps involved. Looking at steps across multiple suites may neglect showing steps unique to particular suites.

ETL for the builds is not complete, so data will appear to be missing. Would you like to see it? Email me [klahnakoski@mozilla.com](mailto:klahnakoski@mozilla.com)!

Start Date

End Date

Interval

## Selection Summary

**Pool: All**

**Platform: win64**

**Product: firefox**

**Suite: mochitest-browser-chrome**

**Hold CTRL while clicking to multi-select and deselect from the lists below.**

## Buildbot Test Timings (1 week average)

step

downloading to buildprops.json

wget archiver\_client.py

'rm -rf ...'

archiver\_client.py

downloading to oauth.txt

set props

desktop\_unittest.py

desktop\_unittest.py::clobber

desktop\_unittest.py::read-buildbot-config

desktop\_unittest.py::download-and-extract

desktop\_unittest.py::create-virtualenv

desktop\_unittest.py::install

desktop\_unittest.py::run-tests

'rm -f ...'

reboot slave lost

duration

0

2

4

2

0

0

0

98

0

65

117

4

776

0

1

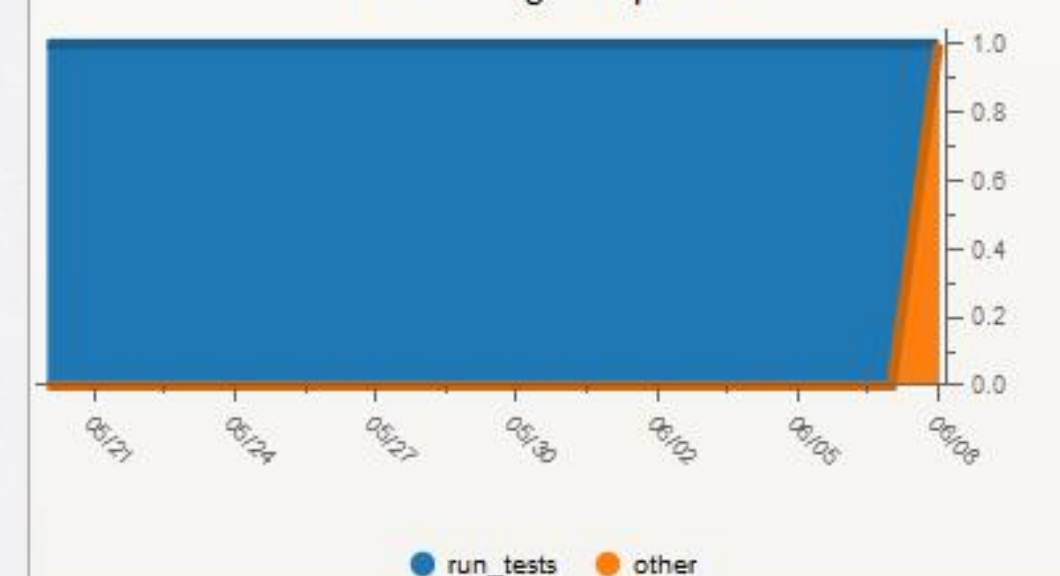
Number of 'run-tests'



Average Step Time



Percentage Step Time





# Dashboards

Done

View average **test** timings for each step in BuildBot, or MozHarness. Filter by pool, platform or suite.

Only steps executed over 10% of the time are shown here. It is important you select a suite to get a reasonable estimate of the steps involved. Looking at steps across multiple suites may neglect showing steps unique to particular suites.

ETL for the builds is not complete, so data will appear to be missing. Would you like to see it? Email me [klahnakoski@mozilla.com](mailto:klahnakoski@mozilla.com)!

Start Date

End Date

Interval

## Selection Summary

Pool: All

Platform: win64

Product: firefox

Suite: mochitest-browser-chrome

Hold CTRL while clicking to multi-select and deselect from the lists below.

## Buildbot Test Timings (1 week average)

step

downloading to buildprops.json

wget archiver\_client.py

'rm -rf ...'

archiver\_client.py

downloading to oauth.txt

set props

desktop\_unittest.py

desktop\_unittest.py::clobber

desktop\_unittest.py::read-buildbot-config

desktop\_unittest.py::download-and-extract

desktop\_unittest.py::create-virtualenv

desktop\_unittest.py::install

desktop\_unittest.py::run-tests

'rm -f ...'

reboot slave lost

duration

0

2

4

2

0

0

0

98

0

65

117

4

776

0

1

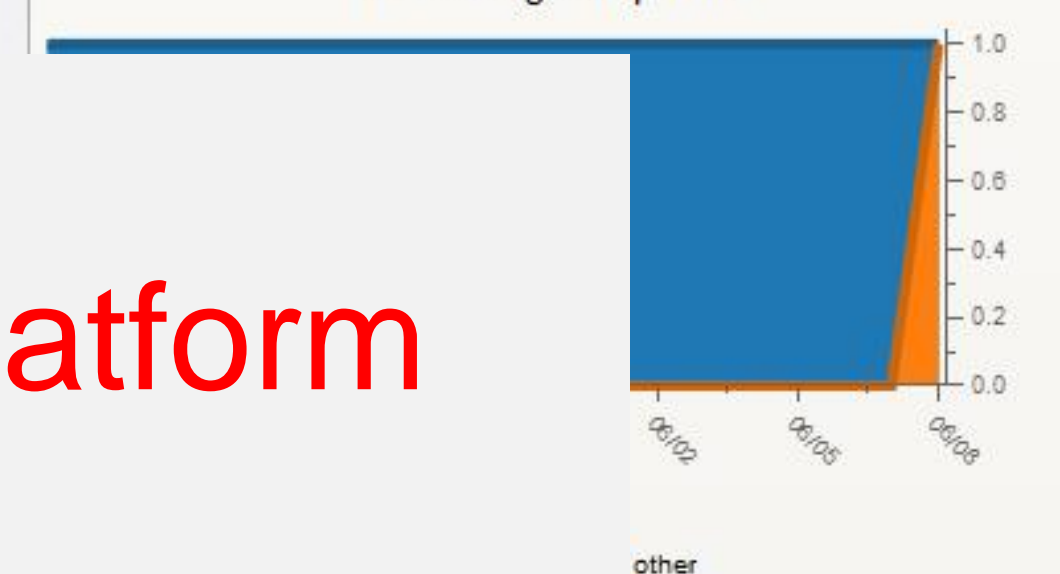
Number of 'run-tests'



Average Step Time



Percentage Step Time



← suite and platform



# Dashboards

Done

View average **test** timings for each step in BuildBot, or MozHarness. Filter by pool, platform or suite.

Only steps executed over 10% of the time are shown here. It is important you select a suite to get a reasonable estimate of the steps involved. Looking at steps across multiple suites may neglect showing steps unique to particular suites.

ETL for the builds is not complete, so data will appear to be missing. Would you like to see it? Email me [klahnakoski@mozilla.com](mailto:klahnakoski@mozilla.com)!

Start Date

End Date

Interval

## Selection Summary

**Pool:** All

**Platform:** win64

**Product:** firefox

**Suite:** mochitest-browser-chrome

Hold CTRL while clicking to multi-select and deselect from the lists below.

## Buildbot Test Timings (1 week average)

step

downloading to buildprops.json

wget archiver\_client.py

'rm -rf ...'

archiver\_client.py

downloading to oauth.txt

set props

desktop\_unittest.py

desktop\_unittest.py::clobber

desktop\_unittest.py::read-buildbot-config

desktop\_unittest.py::download-and-extract

desktop\_unittest.py::create-virtualenv

desktop\_unittest.py::install

desktop\_unittest.py::run-tests

'rm -f ...'

reboot slave lost

duration

0

2

4

2

0

0

0

98

0

65

117

4

776

0

1

Number of 'run-tests'



Average Step Time



Percentage Step Time



2min for virtualenv



# Dashboards

Done

View average **test** timings for each step in BuildBot, or MozHarness. Filter by pool, platform or suite.

Only steps executed over 10% of the time are shown here. It is important you select a suite to get a reasonable estimate of the steps involved. Looking at steps across multiple suites may neglect showing steps unique to particular suites.

ETL for the builds is not complete, so data will appear to be missing. Would you like to see it? Email me [klahnakoski@mozilla.com](mailto:klahnakoski@mozilla.com)!

Start Date

End Date

Interval

## Selection Summary

**Pool:** All

**Platform:** win64

**Product:** firefox

**Suite:** mochitest-browser-chrome

Hold CTRL while clicking to multi-select and deselect from the lists below.

## Buildbot Test Timings (1 week average)

step

downloading to buildprops.json

wget archiver\_client.py

'rm -rf ...'

archiver\_client.py

downloading to oauth.txt

set props

desktop\_unittest.py

desktop\_unittest.py::clobber

desktop\_unittest.py::read-buildbot-config

desktop\_unittest.py::download-and-extract

desktop\_unittest.py::create-virtualenv

desktop\_unittest.py::install

desktop\_unittest.py::run-tests

'rm -f ...'

reboot slave lost

duration

0

2

4

2

0

0

0

98

0

65

117

4

776

0

1

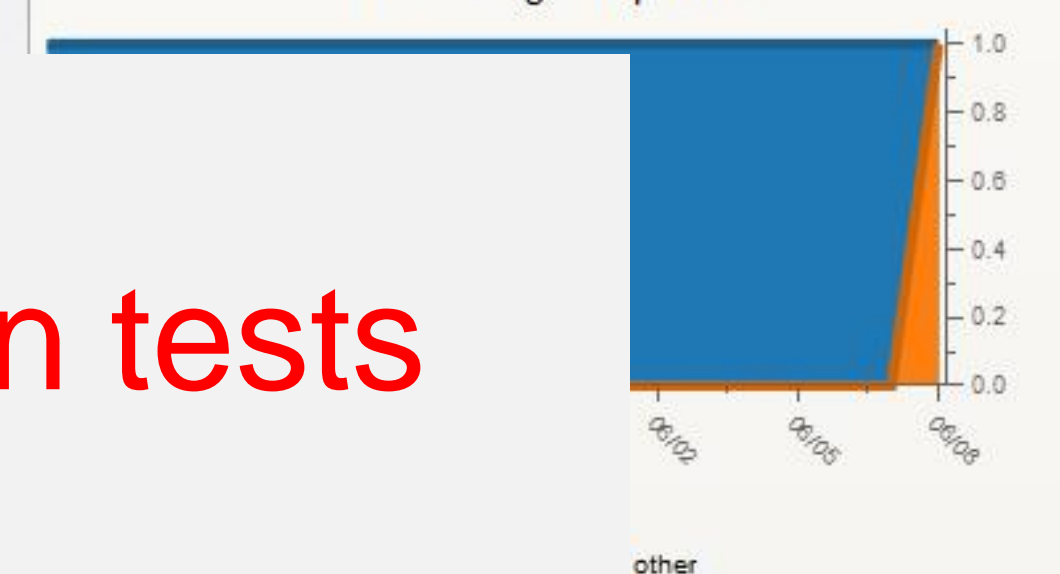
Number of 'run-tests'



Average Step Time



Percentage Step Time



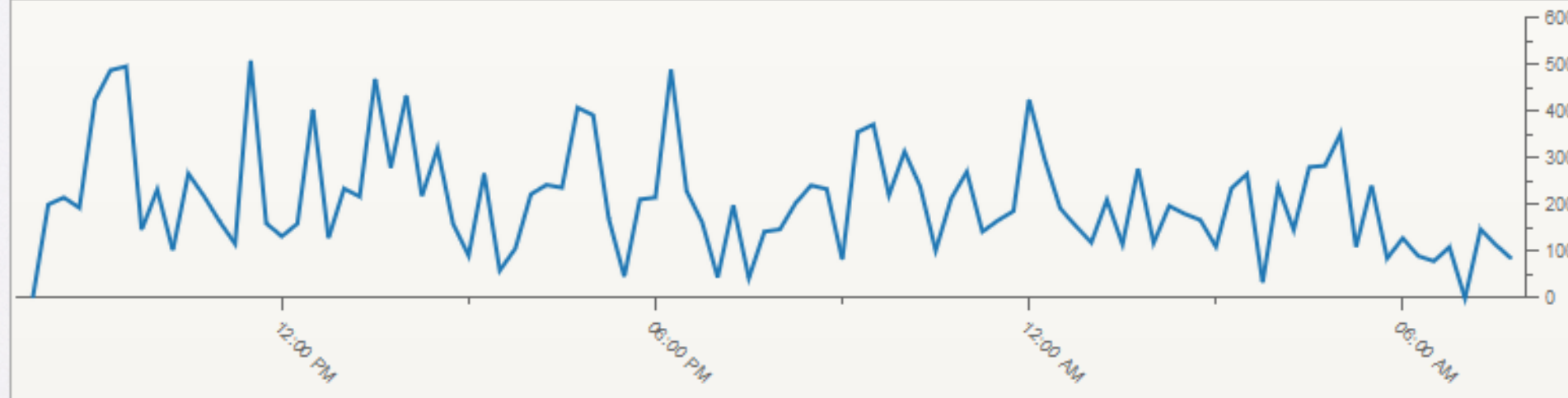
13min to run tests



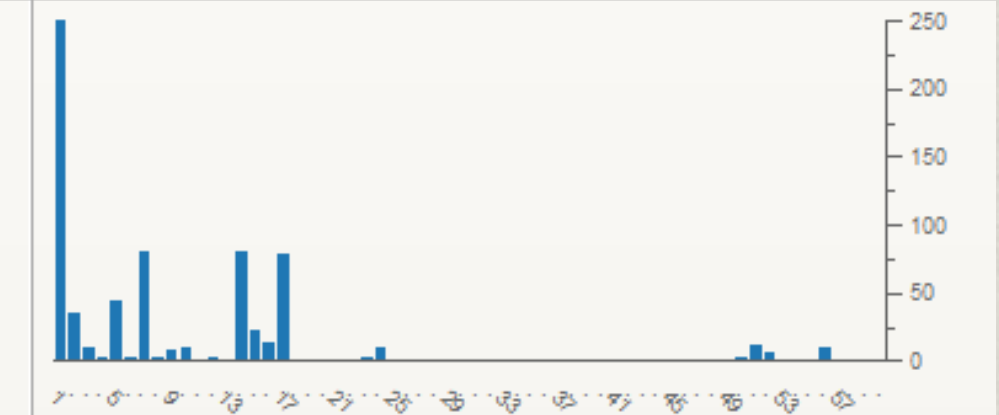
# Dashboards

## Buildbot Simulator (t-yosemite-r7, Monday, Jun 06)

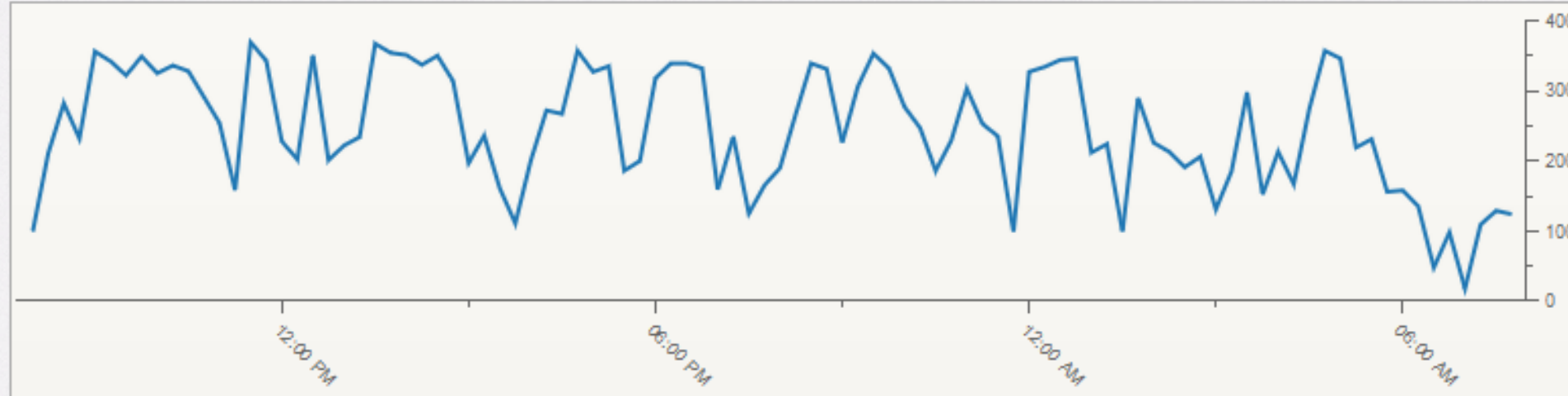
New Requests



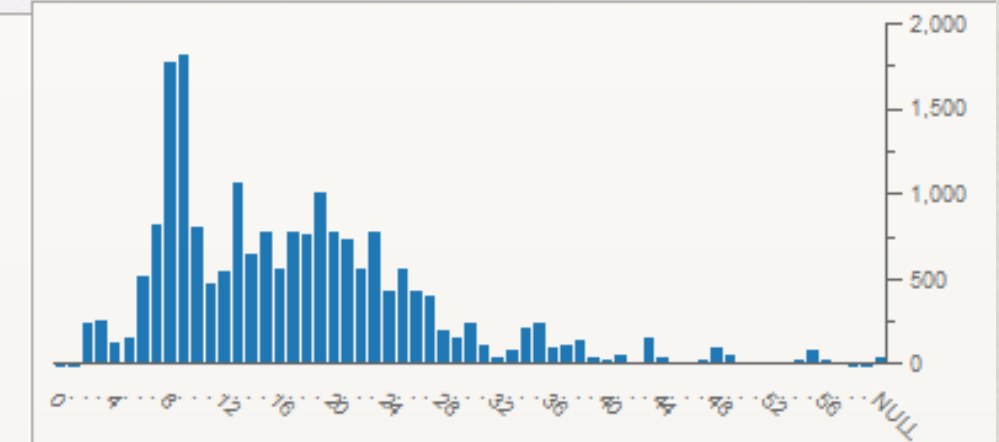
Request Block Size



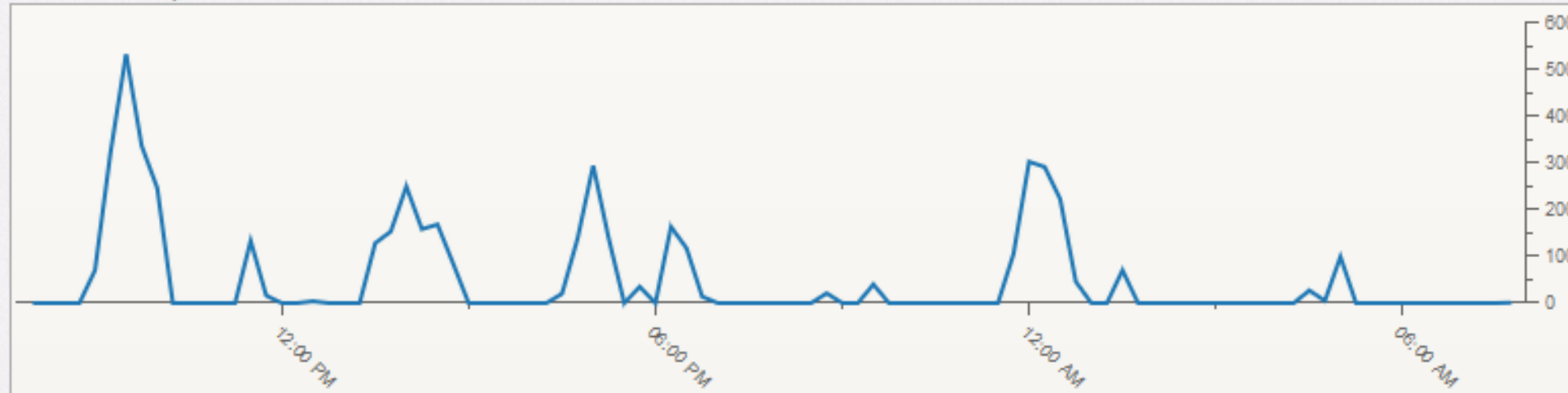
Historical Active Machines



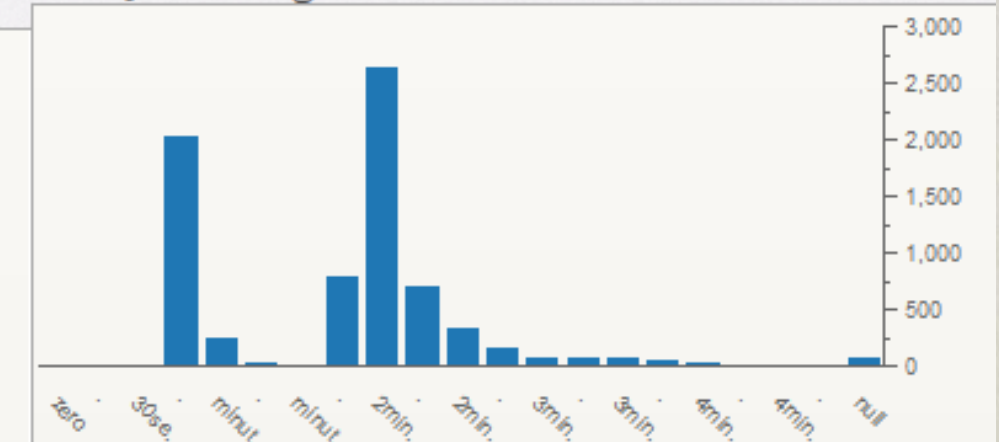
Job Duration



Historical Queue Size



Inter-Job Timing

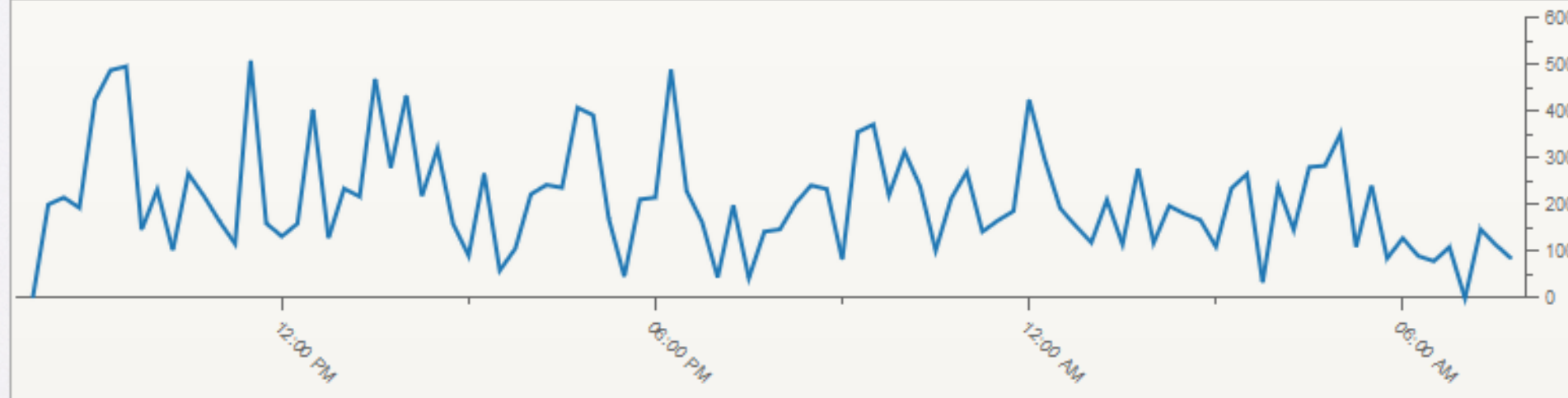




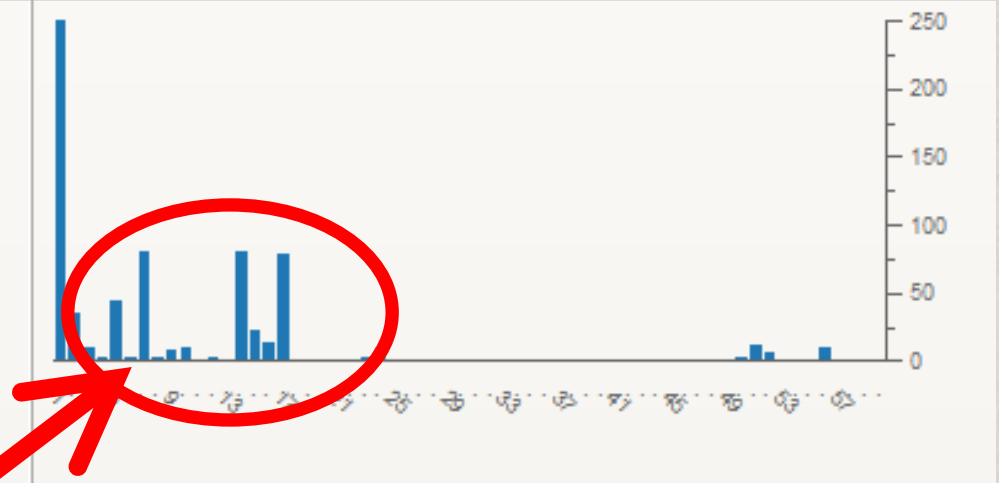
# Dashboards

## Buildbot Simulator (t-yosemite-r7, Monday, Jun 06)

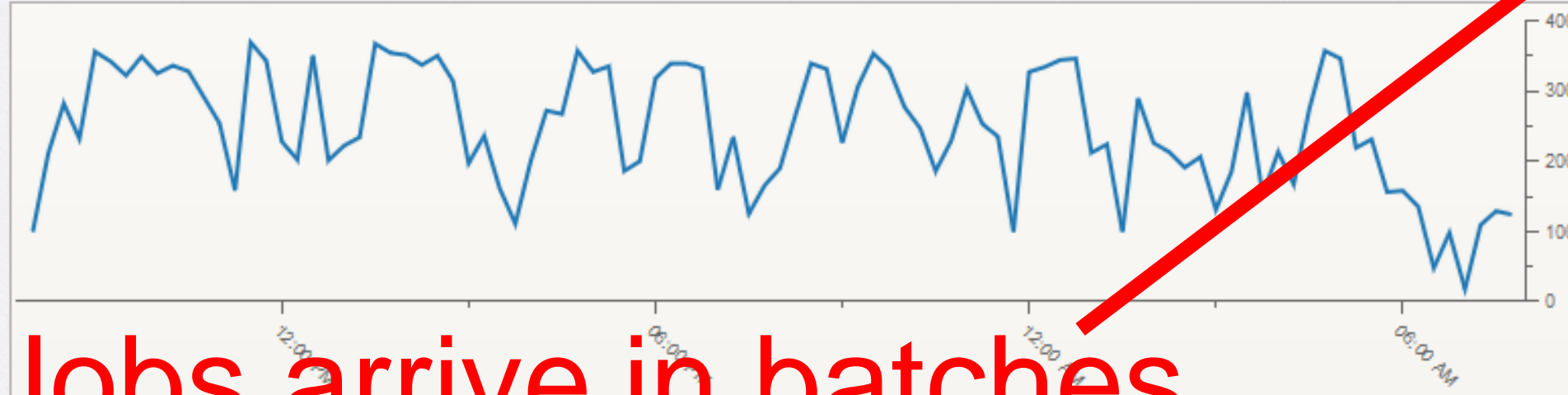
New Requests



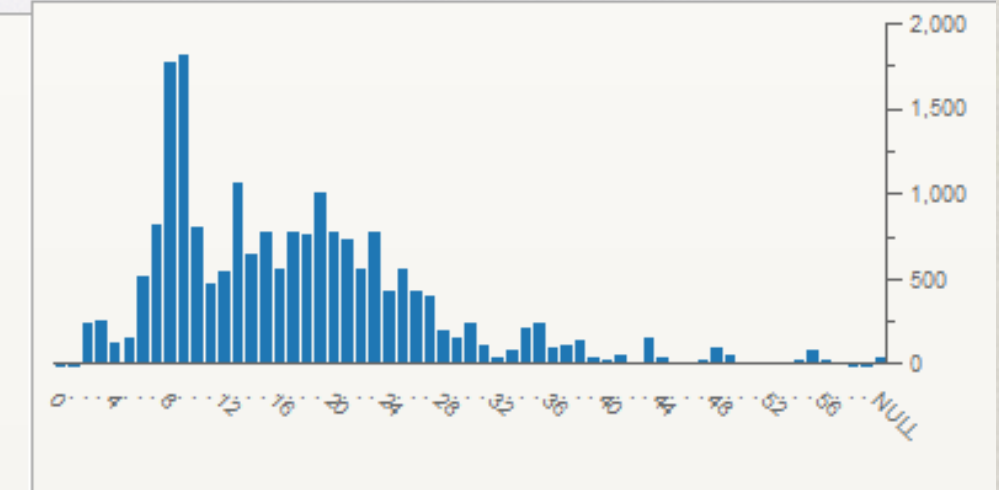
Request Block Size



Historical Active Machines

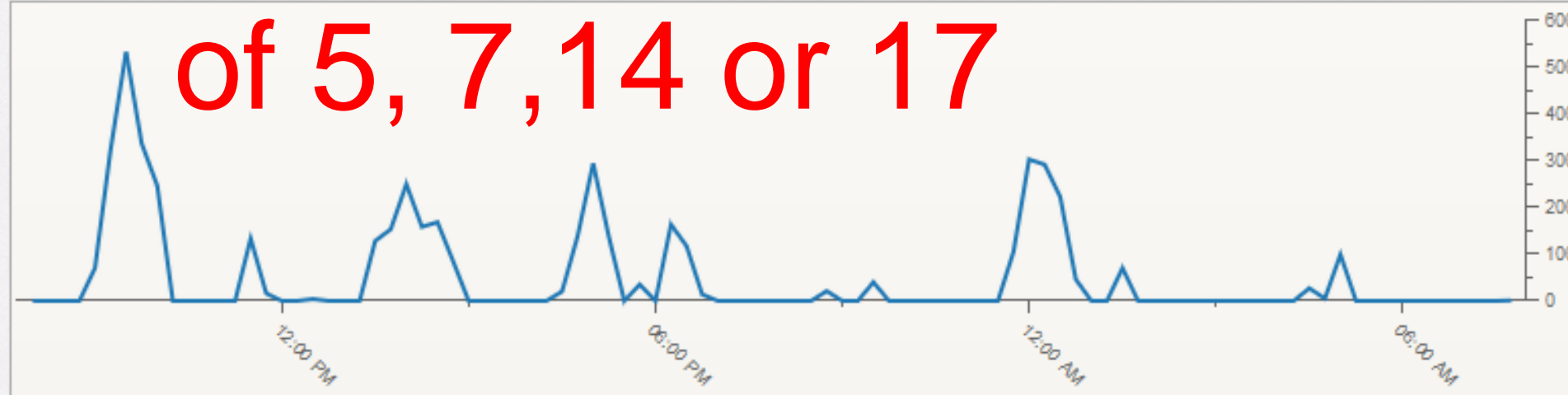


Job Duration

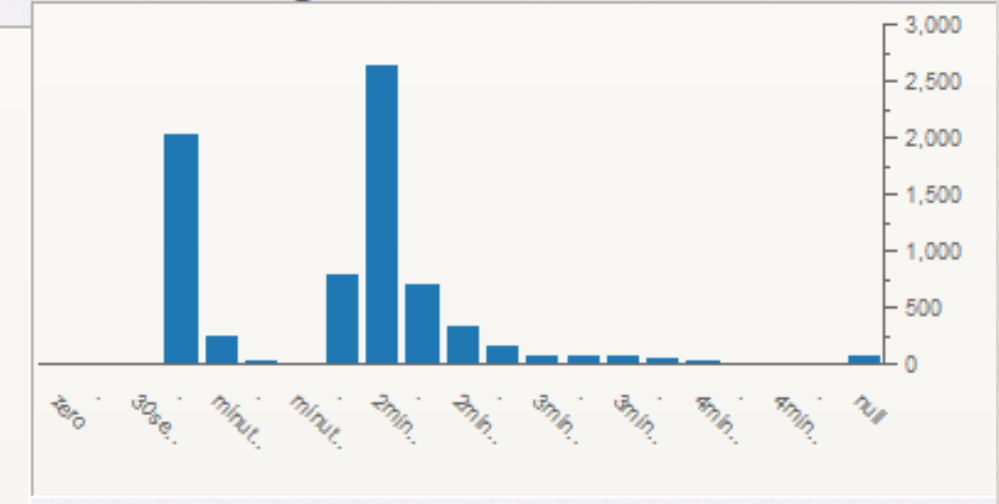


**Jobs arrive in batches  
of 5, 7, 14 or 17**

Historical Queue Size



Inter-Job Timing

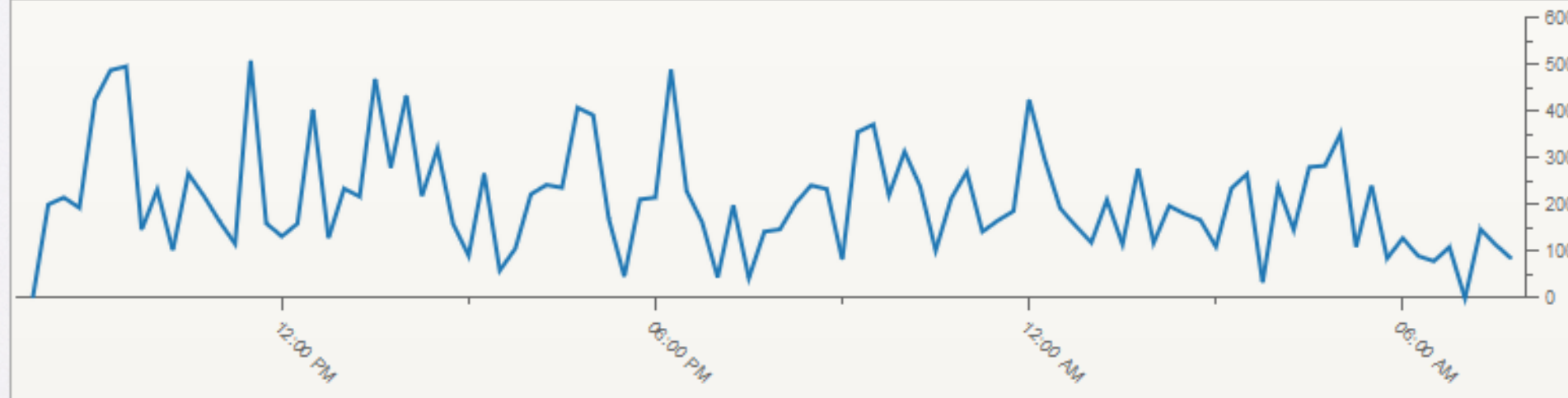




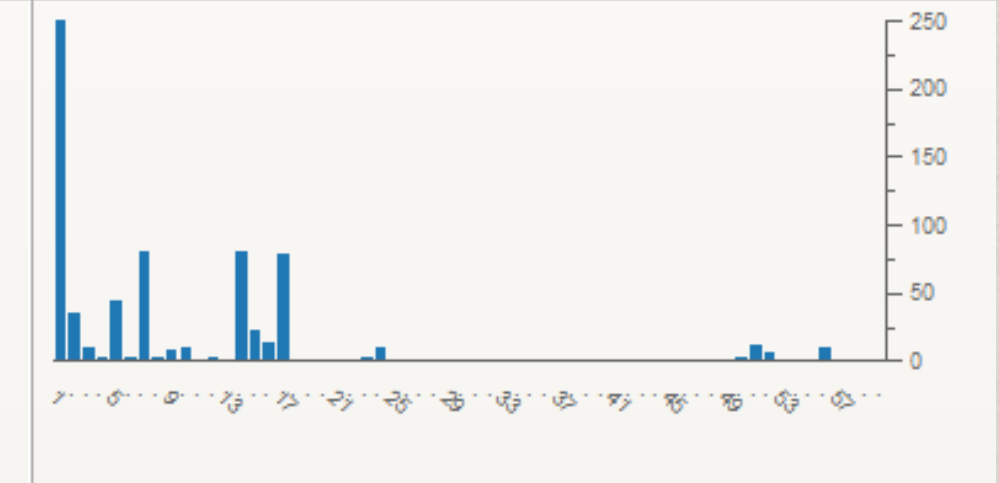
# Dashboards

## Buildbot Simulator (t-yosemite-r7, Monday, Jun 06)

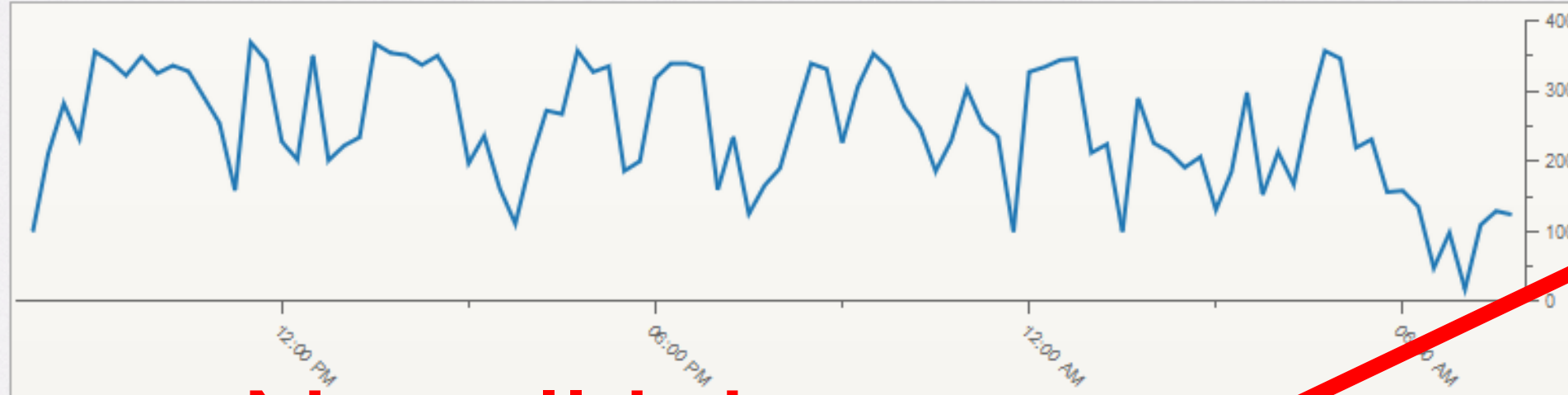
New Requests



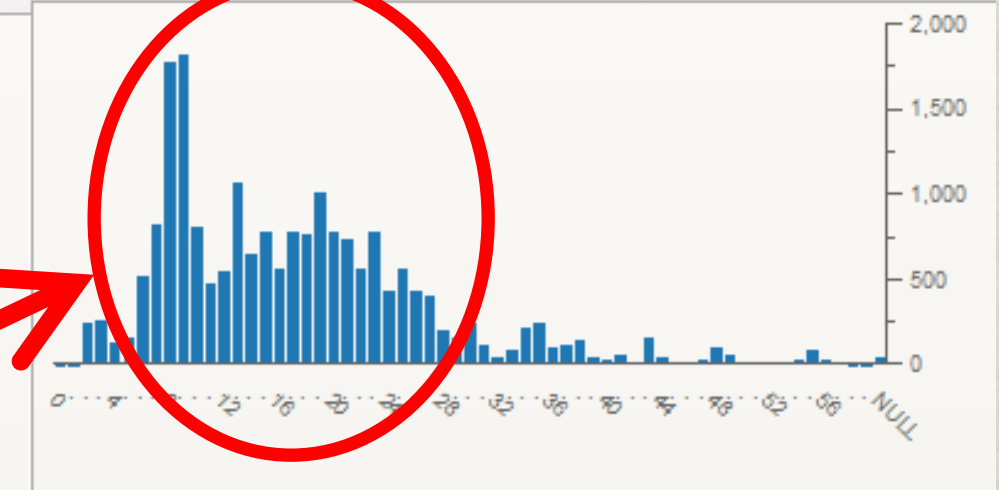
Request Block Size



Historical Active Machines



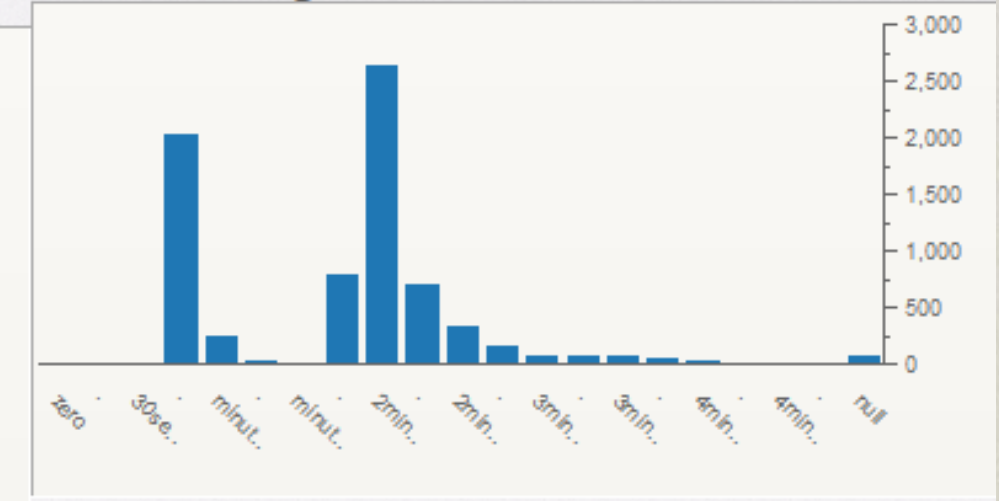
Job Duration



Historical Queue Size



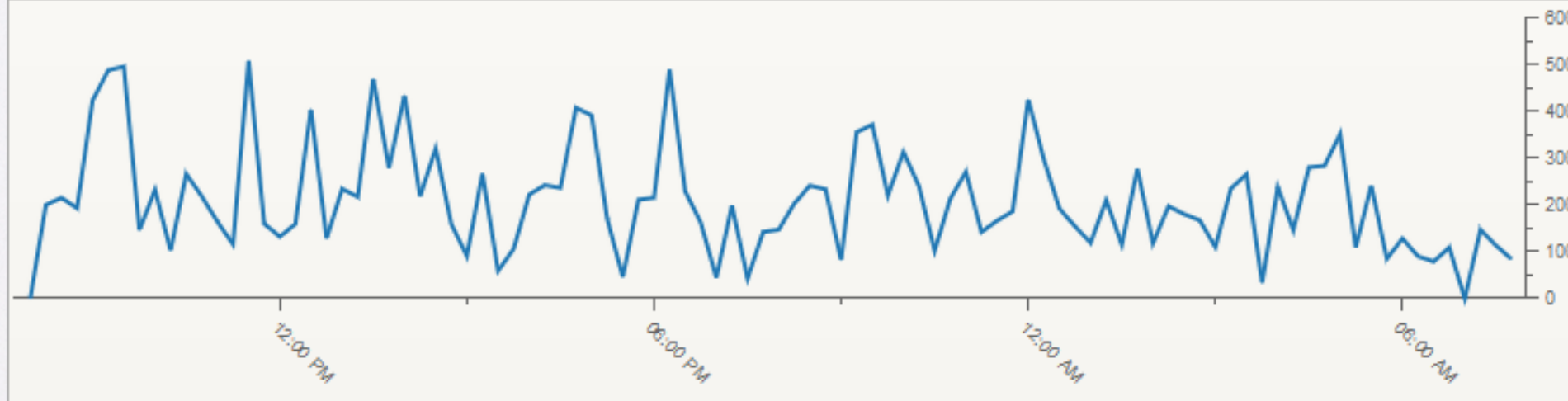
Inter-Job Timing



Not all jobs are  
the same



## New Requests



| Year | Number of Publications |
|------|------------------------|
| 1970 | 240                    |
| 1971 | 60                     |
| 1972 | 20                     |
| 1973 | 10                     |
| 1974 | 70                     |
| 1975 | 10                     |
| 1976 | 100                    |
| 1977 | 10                     |
| 1978 | 20                     |
| 1979 | 10                     |
| 1980 | 10                     |
| 1981 | 10                     |
| 1982 | 10                     |
| 1983 | 10                     |
| 1984 | 10                     |
| 1985 | 10                     |
| 1986 | 10                     |
| 1987 | 10                     |
| 1988 | 10                     |
| 1989 | 10                     |
| 1990 | 10                     |
| 1991 | 10                     |
| 1992 | 10                     |
| 1993 | 10                     |
| 1994 | 10                     |
| 1995 | 10                     |
| 1996 | 10                     |
| 1997 | 10                     |
| 1998 | 10                     |
| 1999 | 10                     |
| 2000 | 10                     |
| 2001 | 10                     |
| 2002 | 10                     |
| 2003 | 10                     |
| 2004 | 10                     |
| 2005 | 10                     |
| 2006 | 10                     |
| 2007 | 10                     |
| 2008 | 10                     |
| 2009 | 100                    |
| 2010 | 10                     |

2min between jobs

A bar chart showing the frequency of the number of children per family. The x-axis is labeled with the number of children (0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10) and the y-axis is labeled with the frequency (0, 500, 1,000, 1,500, 2,000). The distribution is unimodal and slightly right-skewed, peaking at 2 children with a frequency of approximately 1,900.

sometimes longer

A bar chart showing the frequency of different time intervals. The x-axis categories are zero, 30se., minut., minut., 2m., 2min., 3min., 3min., 4min., 4min., and null. The y-axis represents frequency from 0 to 3,000. A red oval highlights the '2m.' category, which has the highest frequency, around 2,800. A red arrow points to the 'minut.' category, which has a frequency of approximately 2,200.

| Category | Frequency |
|----------|-----------|
| zero     | 0         |
| 30se.    | 0         |
| minut.   | ~2,200    |
| minut.   | ~100      |
| 2m.      | ~2,800    |
| 2min.    | ~1,000    |
| 3min.    | ~400      |
| 3min.    | ~200      |
| 3min.    | ~200      |
| 3min.    | ~200      |
| 4min.    | ~100      |
| 4min.    | ~100      |
| null     | ~200      |

# 2min between jobs, sometimes longer



# Dashboards

| High Failing Tests  |           |       | Test Failures   |           |       | Recent Failing Tests |           |       |
|---|-----------|-------|---|-----------|-------|----------------------|-----------|-------|
| Test Name   | Fail Rate | Count | Test Name   | Fail Rate | Count | Test Name            | Fail Rate | Count |
| Main app process exited normally  | 100%      | 66    | toolkit/components/passwordmgr/test/browser/browser_capture_doorhanger.js | 16%       | 80    |                      |           |       |
| automation.py   | 100%      | 24    | browser/base/content/test/urlbar/browser_action_keyword.js                | 16%       | 36    |                      |           |       |
| browser/base/content/test/popupNotifications/browser_popupNotification_4.js | 43%       | 273   | browser/base/content/test/social/browser_social_chatwindowfocus.js        | 15%       | 31    |                      |           |       |
| toolkit/components/passwordmgr/test/browser/browser_capture_doorhanger.js   | 15%       | 233   |   |           |       |                      |           |       |
| browser/base/content/test/social/browser_social_chatwindowfocus.js          | 15%       | 267   |   |           |       |                      |           |       |
| devtools/client/framework/test/browser_toolbox_window_title_frame_select.js | 11%       | 126   |   |           |       |                      |           |       |



# Dashboards

### High Failing Tests

| Test Name   | Fail Rate | Count |
|---|-----------|-------|
| Main app process exited normally  | 100%      | 66    |
| automation.py   | 100%      | 24    |
| browser/base/content/test/popupNotifications/browser_popupNotification_4.js | 43%       | 273   |

### Test Failures

| Test Name   | Fail Rate | Count |
|---|-----------|-------|
| toolkit/components/passwordmgr/test/browser/browser_capture_doorhanger.js | 16%       | 80    |
| browser/base/content/test/urlbar/browser_action_keyword.js                | 10%       | 36    |
| browser/base/content/test/social/browser_social_chatwindowfocus.js        | 15%       | 31    |

### Recent Failing Tests

| Test Name   | Fail Rate | Count |
|---|-----------|-------|
| toolkit/components/passwordmgr/test/browser/browser_capture_doorhanger.js | 16%       | 80    |
| browser/base/content/test/urlbar/browser_action_keyword.js                | 10%       | 36    |
| browser/base/content/test/social/browser_social_chatwindowfocus.js        | 15%       | 31    |

### High Failing Tests

| Test Name   | Fail Rate | Count |
|---|-----------|-------|
| Main app process exited normally  | 100%      | 66    |
| automation.py   | 100%      | 24    |
| browser/base/content/test/popupNotifications/browser_popupNotification_4.js | 43%       | 273   |
| toolkit/components/passwordmgr/test/browser/browser_capture_doorhanger.js   | 15%       | 233   |
| browser/base/content/test/social/browser_social_chatwindowfocus.js          | 15%       | 267   |
| devtools/client/framework/test/browser_toolbox_window_title_frame_select.js | 11%       | 126   |



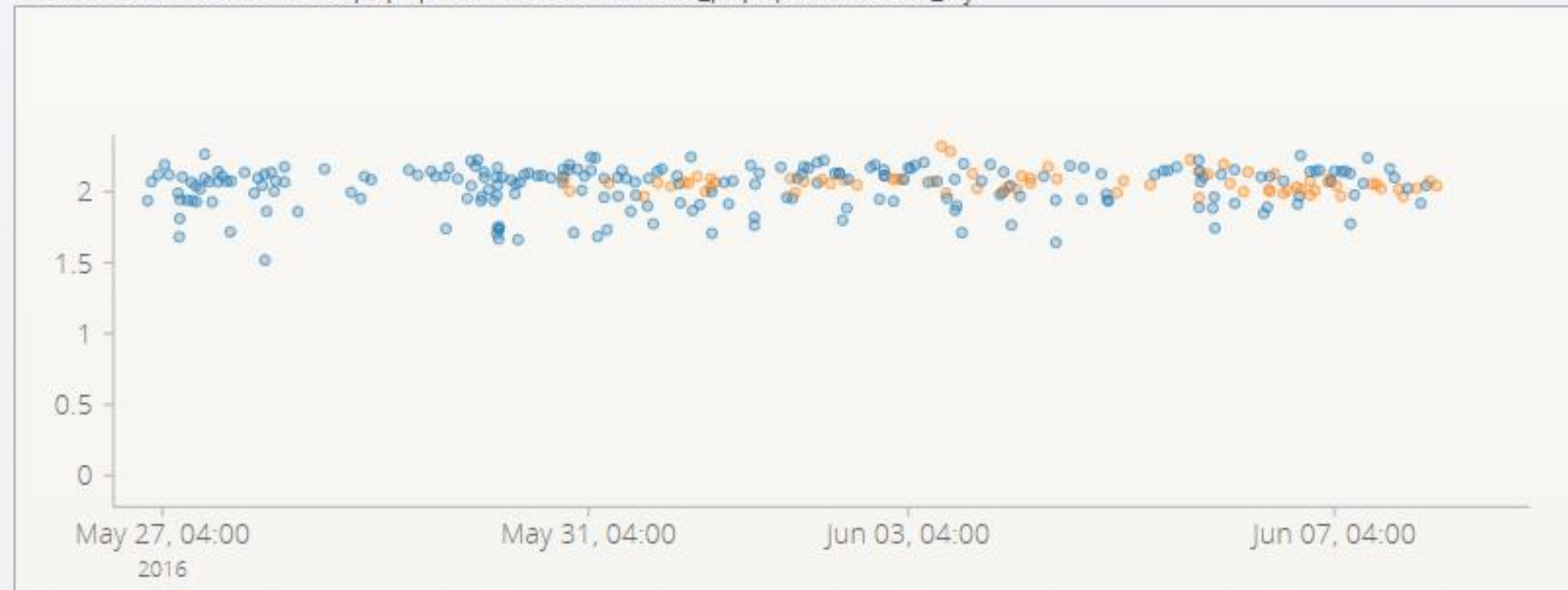
# Dashboards

Search for test: `browser/base/content/test/popupNotifications/browser_popupNotifi`

Done

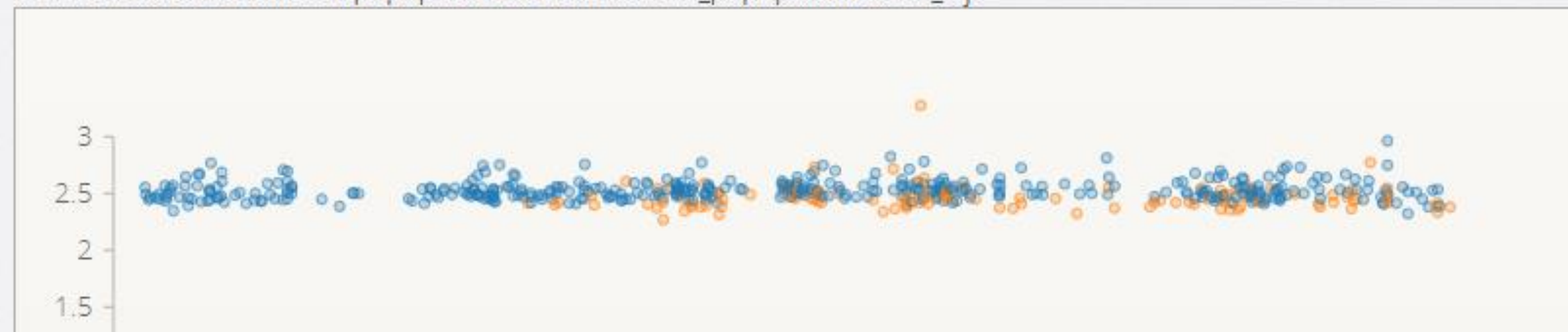
FIREFOX - win64 (pgo) mochitest-e10s-browser-chrome (e10s) (35% failures)

`browser/base/content/test/popupNotifications/browser_popupNotification_4.js`



FIREFOX - win64 (opt) mochitest-e10s-browser-chrome (e10s) (29% failures)

`browser/base/content/test/popupNotifications/browser_popupNotification_4.js`





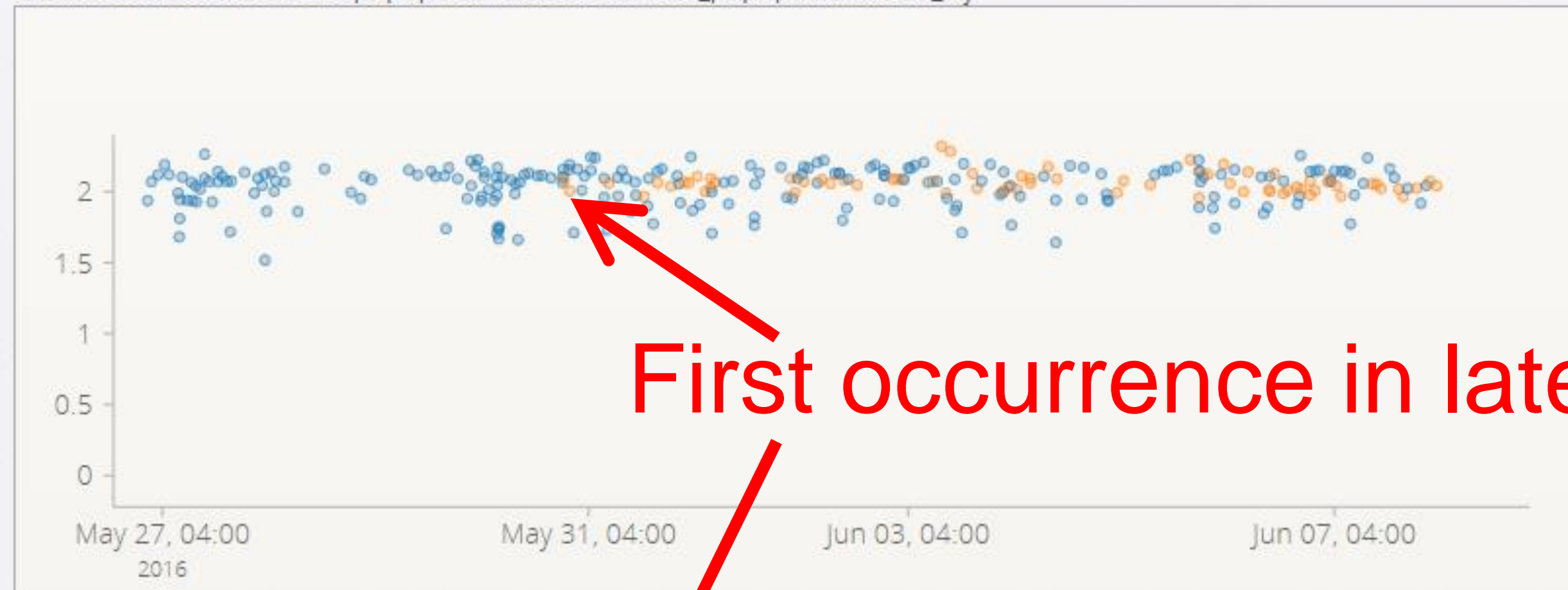
# Dashboards

Search for test: `browser/base/content/test/popupNotifications/browser_popupNotifi`

Done

FIREFOX - win64 (pgo) mochitest-e10s-browser-chrome (e10s) (35% failures)

`browser/base/content/test/popupNotifications/browser_popupNotification_4.js`



First occurrence in late May

FIREFOX - win64 (opt) mochitest-e10s-browser-chrome (e10s) (29% failures)

`browser/base/content/test/popupNotifications/browser_popupNotification_4.js`





# Active Data

Don't use static files!  
Use an Active Database!

Kyle Lahnakoski  
Engineering Productivity



# Active Data

<http://activedata.allizom.org>

Kyle Lahnakoski  
Engineering Productivity