

# **ActiveData**

<https://activedata.allizom.org>

Kyle Lahnakoski

klahnakoski@mozilla.com

Product Integrity

December 2017

Another lightning talk on...

# ActiveData

<https://activedata.allizom.org>

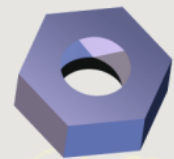
Kyle Lahnakoski

[klahnakoski@mozilla.com](mailto:klahnakoski@mozilla.com)

Product Integrity

December 2017

# What Data?



Buildbot

Jobs

Test Results

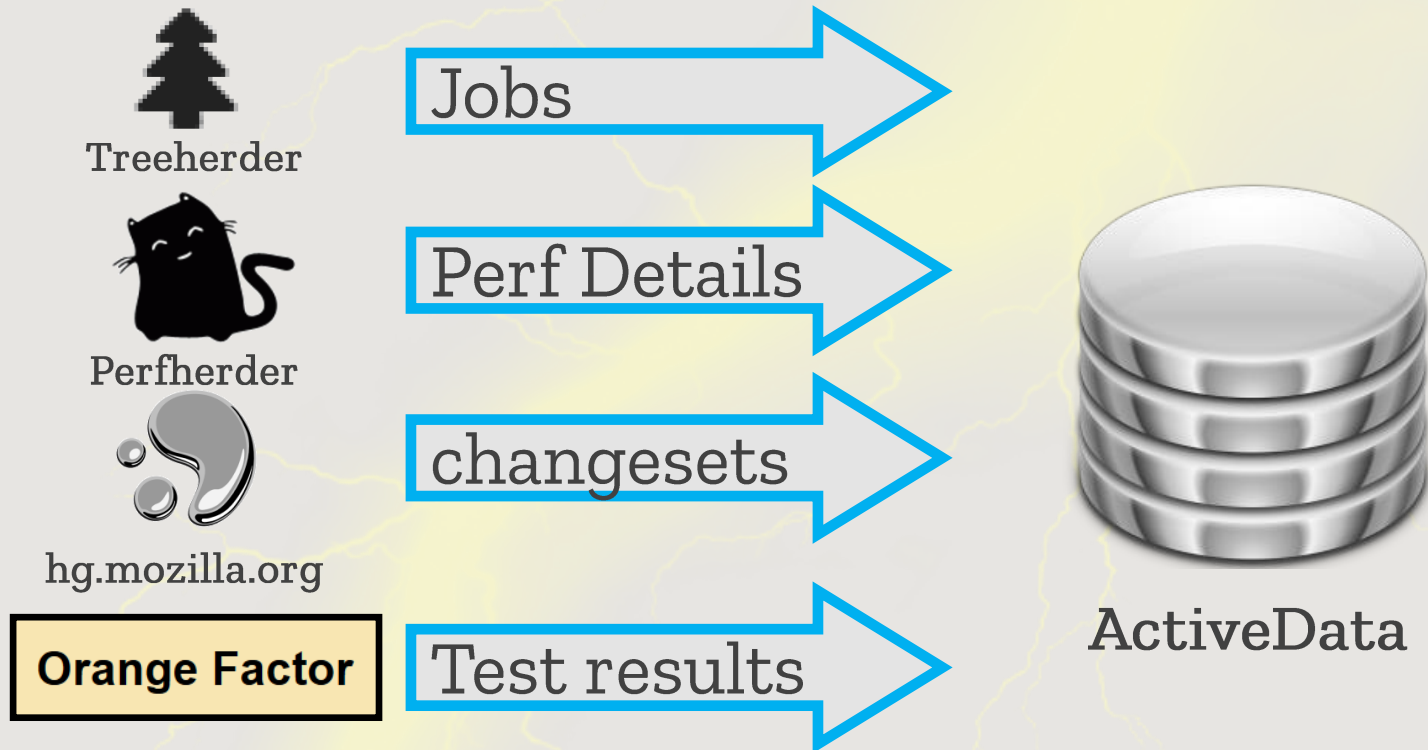


ActiveData

# What Data?



# What Data?



# Firefox Test Results

Flask app for presenting Firefox test results from [ActiveData](#).

Test Results [fx-test](#) [unittest](#)

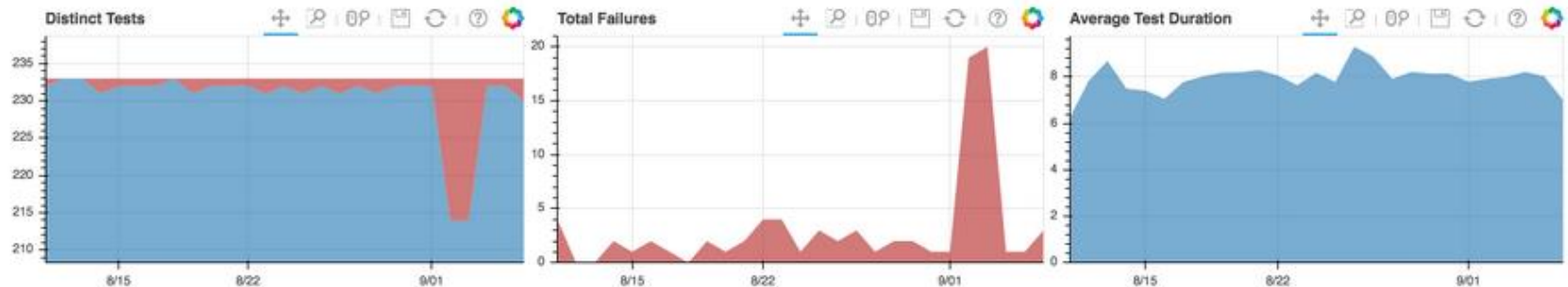
Branch  Path  Since  [Go](#)

Fork me on GitHub

## Summary

A total of **233** tests ran **504,853** times between **11-Aug-2017** and **08-Sep-2017**.

Query ran on 08 Sep 2017 at 14:55:02 and took 8.034 seconds.



## Slowest

42 seconds	dom/workers/test/serviceworkers/test_file_blob_response.html
33 seconds	dom/workers/test/serviceworkers/test_third_party_iframes.html
30 seconds	dom/workers/test/serviceworkers/test_fetch_event.html
29 seconds	dom/workers/test/test_referrer_header_worker.html

## Lowest pass rate

98%	dom/workers/test/test_fibonacci.html
99%	dom/workers/test/serviceworkers/test_openWindow.html
99%	dom/workers/test/serviceworkers/test_worker_reference_gc_timeout.html
99%	dom/workers/test/serviceworkers/test_service_worker_allowed.html

Data, so what!?

# Example: Find Artifacts

```
{
  "AUTHORS": [
    "mozilla.org",
    "Licensing"
  ],
  "accessible/aom/AccessibleNode.cpp": [
    "Core",
    "Disability Access APIs"
  ],
  "accessible/aom/AccessibleNode.h": [
    "Core",
    "Disability Access APIs"
  ],
  "accessible/aom/moz.build": [
    "Core",
    "Disability Access APIs"
  ],
}
```

**Bugzilla job generates components.json.gz**



# Example: Find Artifacts

```
{  
  "select": ["url", "action.start_time"],  
  "from": "task.task.artifacts",  
  "where": {"and": [  
    {"gte": {"action.start_time": {"date": "today"}}},  
    {"eq": {"name": "public/components.json.gz"}}  
  ]},  
  "sort": {"action.start_time": "desc"},  
  "limit": 1  
}
```

# Example: Find Artifacts

```
{  
  "select": ["url", "action.start_time"],  
  "from": "task.task.artifacts",  
  "where": {"and": [  
    {"gte": {"action.start_time": {"date": "today"}}},  
    {"eq": {"name": "public/components.json.gz"}},  
  ]},  
  "sort": {"action.start_time": "desc"},  
}
```

```
SELECT  
  url,  
  "action.start_time"  
FROM  
  "task.task.artifacts"  
WHERE  
  "action.start_time" >= TODAY() AND  
  name = 'public/components.json.gz'  
SORT BY  
  "action.start_time" DESC  
LIMIT  
  1
```

# Example: Find Artifacts

```
{  
  "select": ["url", "action.start_time"],  
  "from": "task.task.artifacts",  
  "where": {"and": [  
    {"gte": {"action.start_time": {"date": "today"}}},  
    {"eq": {"name": "public/components.json.gz"}}  
  ]},  
  "sort": {"action.start_time": "desc"}  
}
```

```
{ "data": {  
  "url": ["http://queue.taskcluster.net/v1/task/KSu  
Waly5QVmDi5u4040JyA/artifacts/public/components.json.gz  
"],  
  "action.start_time": [1513181228.52]  
}}
```

# Example: Find Artifacts

## ActiveData Query Tool

```
1 {  
2   "from": "task.task.artifacts",  
3   "select": ["url", "action.start_time"],  
4   "where": {"and": [  
5     {"gte": {"action.start_time": {"date": "today"}}},  
6     {"regex": {"name": "public/components.json.gz"}}  
7   ]},  
8   "sort": {"action.start_time": "desc"}  
9 }  
10  
11  
12  
13  
14
```

EXECUTE

Copy

Share

as Table

as Json

10 rows (up to 3000 shown)

rownum	url	action.start_time
0	http://queue.taskcluster.net/v1/task/DMmV1Cv7RBy6mJ4J30OOXA/artifacts/public/components.json.gz	13-Dec-2017 11:34:15
1	http://queue.taskcluster.net/v1/task/T-jPUSGYQie_6vyrbQACcg/artifacts/public/components.json.gz	13-Dec-2017 10:32:30

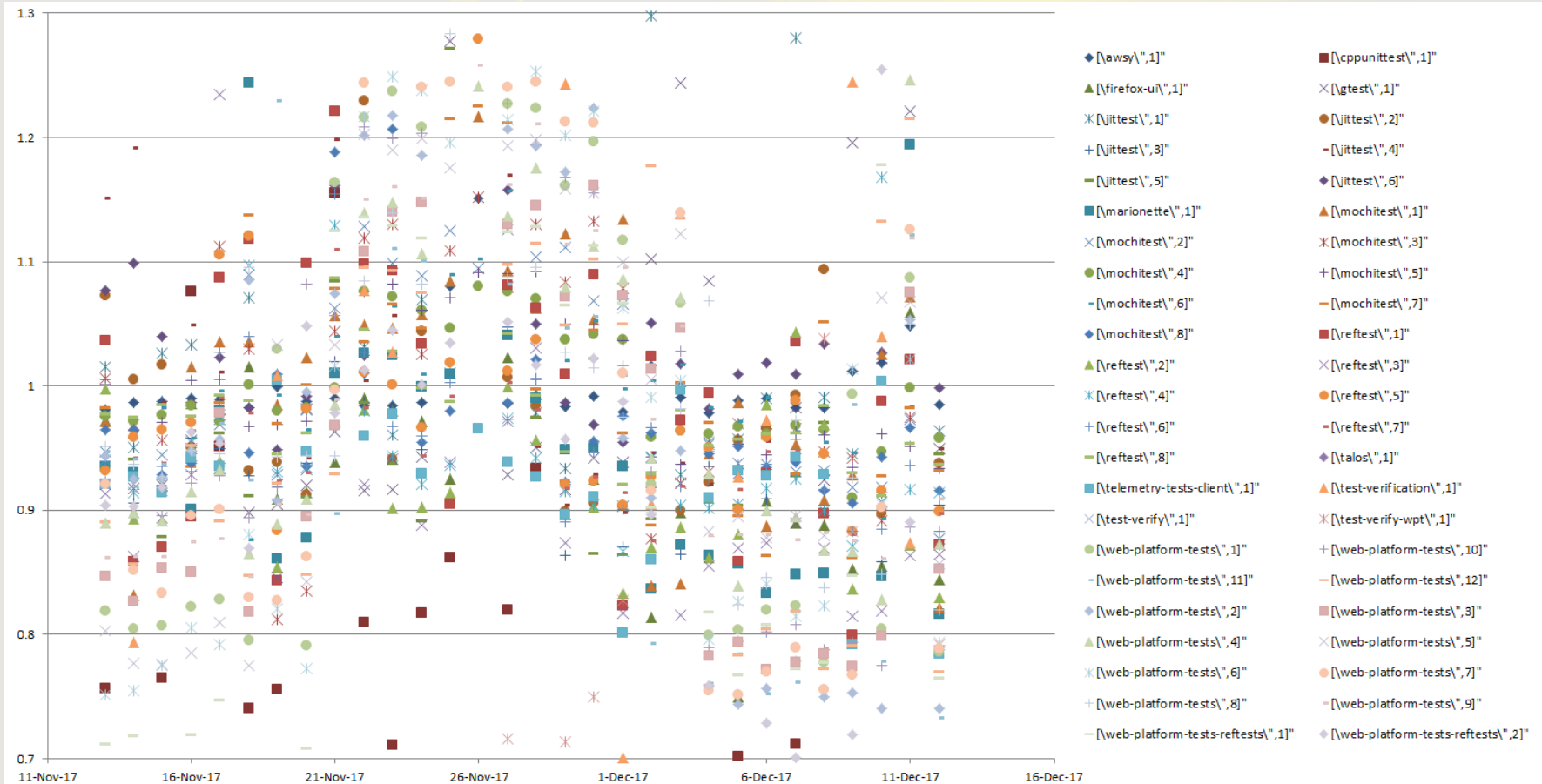
# Example: Test Task Times

```
{
  "from":"task",
  "select":[
    {"aggregate":"count",
     {
       "name":"median duration (seconds)",
       "value":"action.duration",
       "aggregate":"median"
     }
    ],
  "edges":[
    {"name":"chunk","value":["run.suite.name","run.chunk"]},
    {
      "value":"action.start_time",
      "domain":{"
        "type":"time",
        "min":"today-month",
        "max":"today",
        "interval":"day"
      }
    }
  ],
  "where":{"and":[
    {"eq":{"repo.branch.name":"mozilla-inbound"}},
    {"eq":{"treeherder.jobKind":"test"}},
    {"eq":{"build.type":"opt"}},
    {"eq":{"build.platform":"linux64"}},
    {"gte":{"action.start_time":{"date":"today-month"}}}
  ]},
  "limit":1000
}
```

# Example: Test Task Times

- Run the query
- Cut-and-paste to a spreadsheet
- Convert to pivot table
- Normalize the run times over time to 1
- Chart


# Example: Test Task Times



Well, that was painful!



# Recipes!

 This repository Search Pull requests Issues Marketplace Explore


mozilla / active-data-recipes Watch 1 Star 1 Fork 6

[Code](#) [Issues 0](#) [Pull requests 0](#) [Projects 0](#) [Wiki](#) [Insights](#)

A repository of various activedata queries and recipes

102 commits 2 branches 0 releases 3 contributors MPL-2.0

Branch: master New pull request Create new file Upload files Find file Clone or download

 ahal Merge pull request #8 from jmaher/slow Latest commit 2a31742 7 days ago

# Recipes!

- **Canned queries, with parameters**
- **Adds post-processing**
- **Allows cross-referencing**

# Recipes!

```
Microsoft Windows [Version 10.0.14393]  
(c) 2016 Microsoft Corporation. All rights reserved.
```

```
C:\Users\kyle>pip install active-data-recipes
```

```
Collecting active-data-recipes
```

```
...
```

```
C:\Users\kyle>adr --list
```

```
backout_rate
```

```
code_coverage
```

```
code_coverage_by_suite
```

```
files_with_coverage
```

```
hours_on_try
```

```
inspect
```

```
raw_coverage
```

```
task_durations
```

```
try_usage
```

```
try_users
```

# Recipes!

```
C:\Users\kyle>adr -v task_durations
```

Task	Count	Average Hours	Total Hours
test-linux64-nightly/opt-mochitest-valgrind-14	14	4.02	56
test-linux64-nightly/opt-mochitest-valgrind-6	14	4.02	56
test-linux64/opt-mochitest-valgrind-14	12	4.01	48
test-linux64/opt-mochitest-valgrind-6	12	4.01	48
test-linux64-nightly/opt-mochitest-valgrind-1	14	2.68	38
test-linux64/opt-mochitest-valgrind-1	12	2.5	30
spidermonkey-sm-compacting-win32/debug	97	2.3	223
test-linux64/opt-mochitest-valgrind-9	11	2.27	25
test-linux64-nightly/opt-mochitest-valgrind-40	14	2.25	32
spidermonkey-sm-plain-win32/debug	95	2.1	200
test-linux64/opt-mochitest-valgrind-40	12	2.08	25
test-windows10-64-ccov/debug-web-platform-tests-12	40	1.92	77
test-linux64-nightly/opt-mochitest-valgrind-9	13	1.88	24
test-linux64-nightly/opt-mochitest-valgrind-39	14	1.76	25
test-windows10-64-ccov/debug-web-platform-tests-11	41	1.71	70
test-windows10-64-nightly/opt-browser-instrumentation	14	1.69	24
build-win64-nightly/opt	15	1.67	25
test-windows10-64-ccov/debug-web-platform-tests-8	41	1.64	67
test-linux64/opt-mochitest-valgrind-39	12	1.61	19
build-win32-nightly/opt	16	1.56	25
build-win64/pgo	476	1.5	714
test-windows10-64-ccov/debug-web-platform-tests-10	41	1.48	61
test-windows10-64-ccov/debug-web-platform-tests-9	40	1.44	58
build-win32/pgo	471	1.41	664
test-linux64-ccov/opt-web-platform-tests-8	39	1.4	55

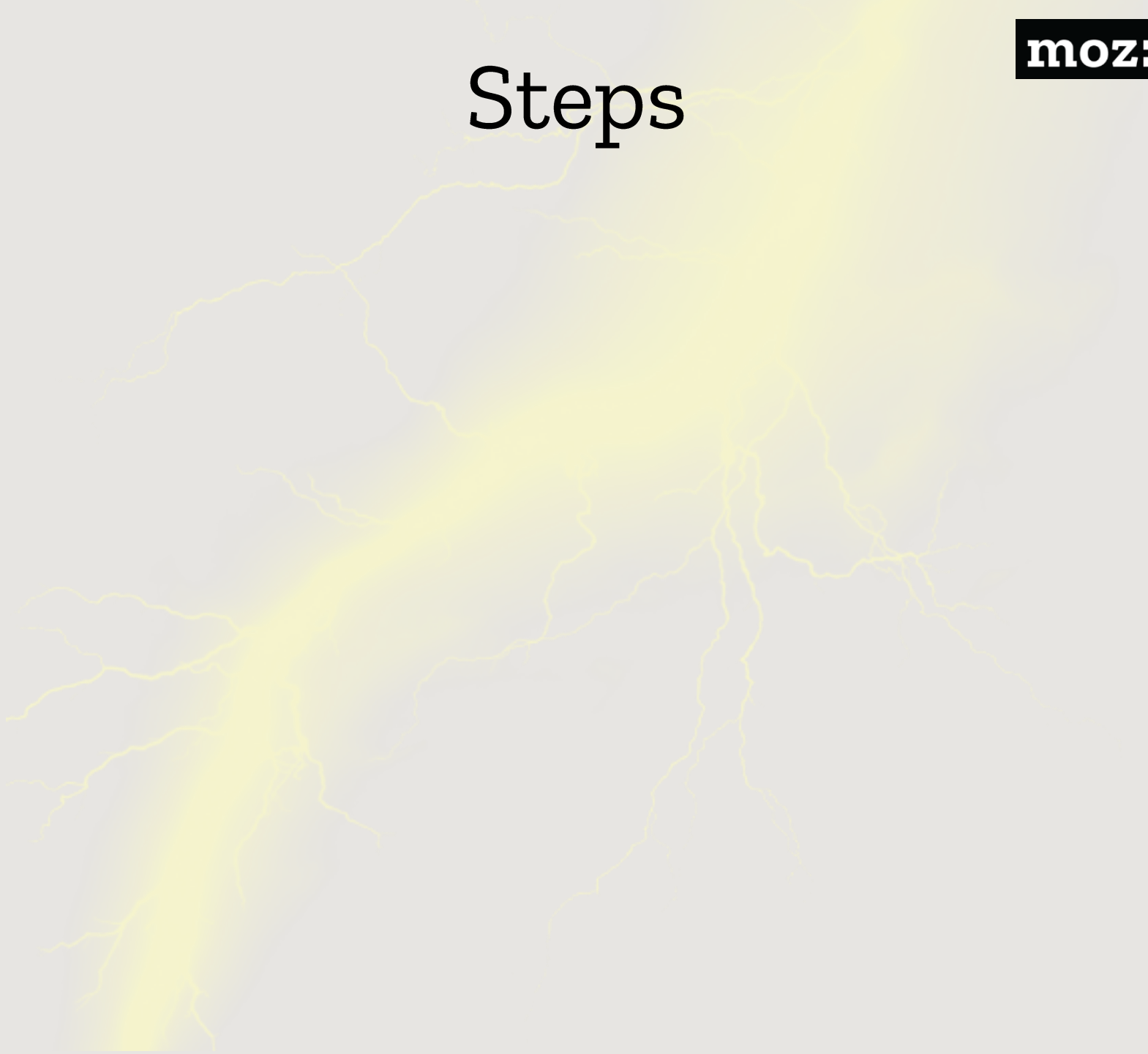
# Recipes!

Branch: master ▾

[active-data-recipes](#) / [adr](#) / [recipes](#) / [task\\_durations.py](#)

```
10 def run(args):
11     parser = RecipeParser('date')
12     parser.add_argument('-b', '--branches', default=['autoland', 'mozilla-inbound', 'mozilla-central'],
13                         help="Branches to gather backout rate on, can be specified "
14                             "multiple times.")
15     parser.add_argument('-p', '--platform', default='windows10-64',
16                         help="platform for results, default is windows10-64")
17     parser.add_argument('-c', '--build_type', default='opt',
18                         help="build configuration, default is 'opt'.")
19     parser.add_argument('--limit', type=int, default=20,
20                         help="Maximum number of jobs to return")
21     parser.add_argument('--sort-key', type=int, default=2,
22                         help="Key to sort on (int, 0-based index)")
23
```

# Steps



# Steps

- 1. Ask me, Kyle Lahnakoski**

# Steps

1. **Ask me, Kyle Lahnakoski**
2. **See #1**



**Done**

# ActiveData

<https://activedata.allizom.org>

Kyle Lahnakoski  
klahnakoski@mozilla.com  
Product Integrity  
December 2017