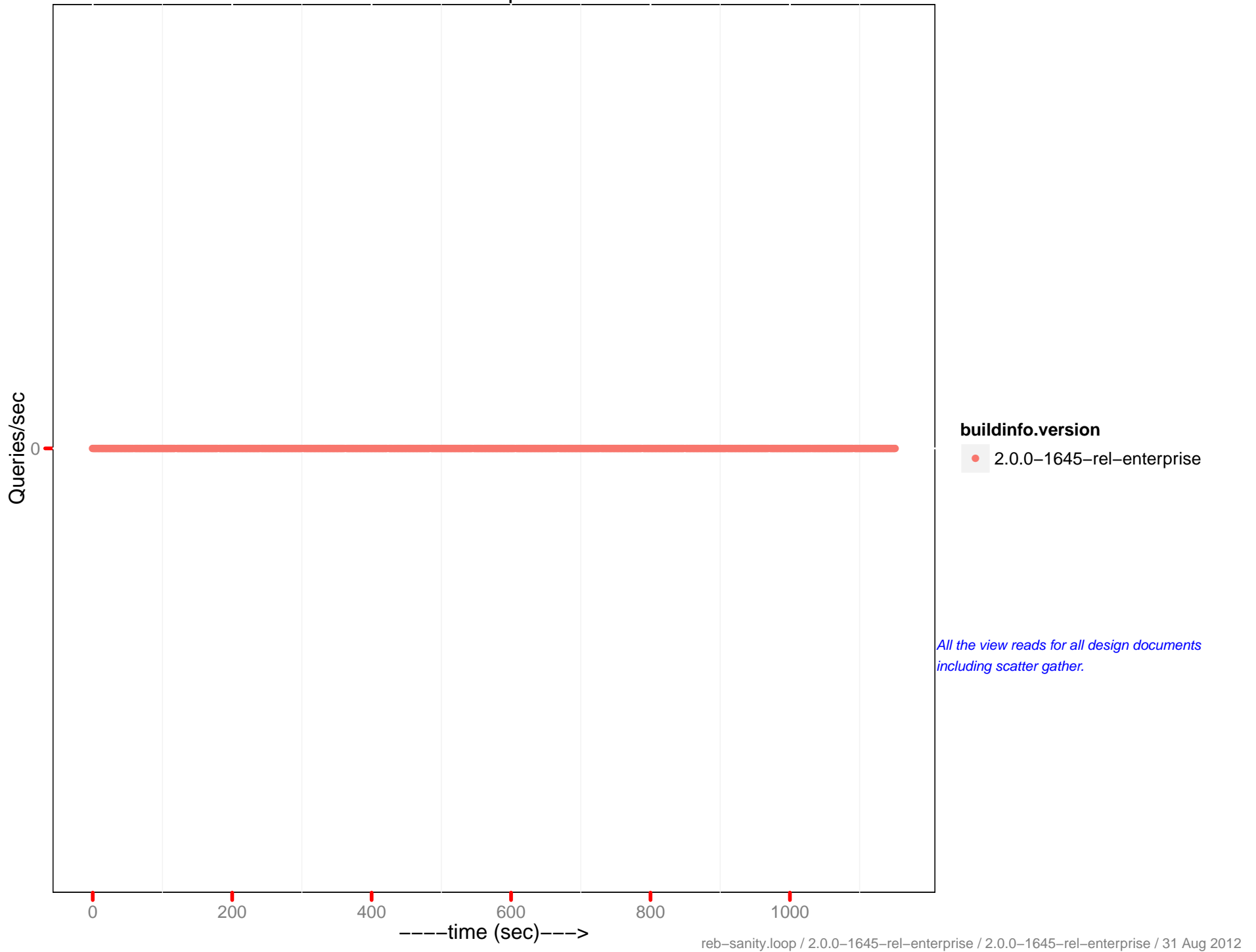
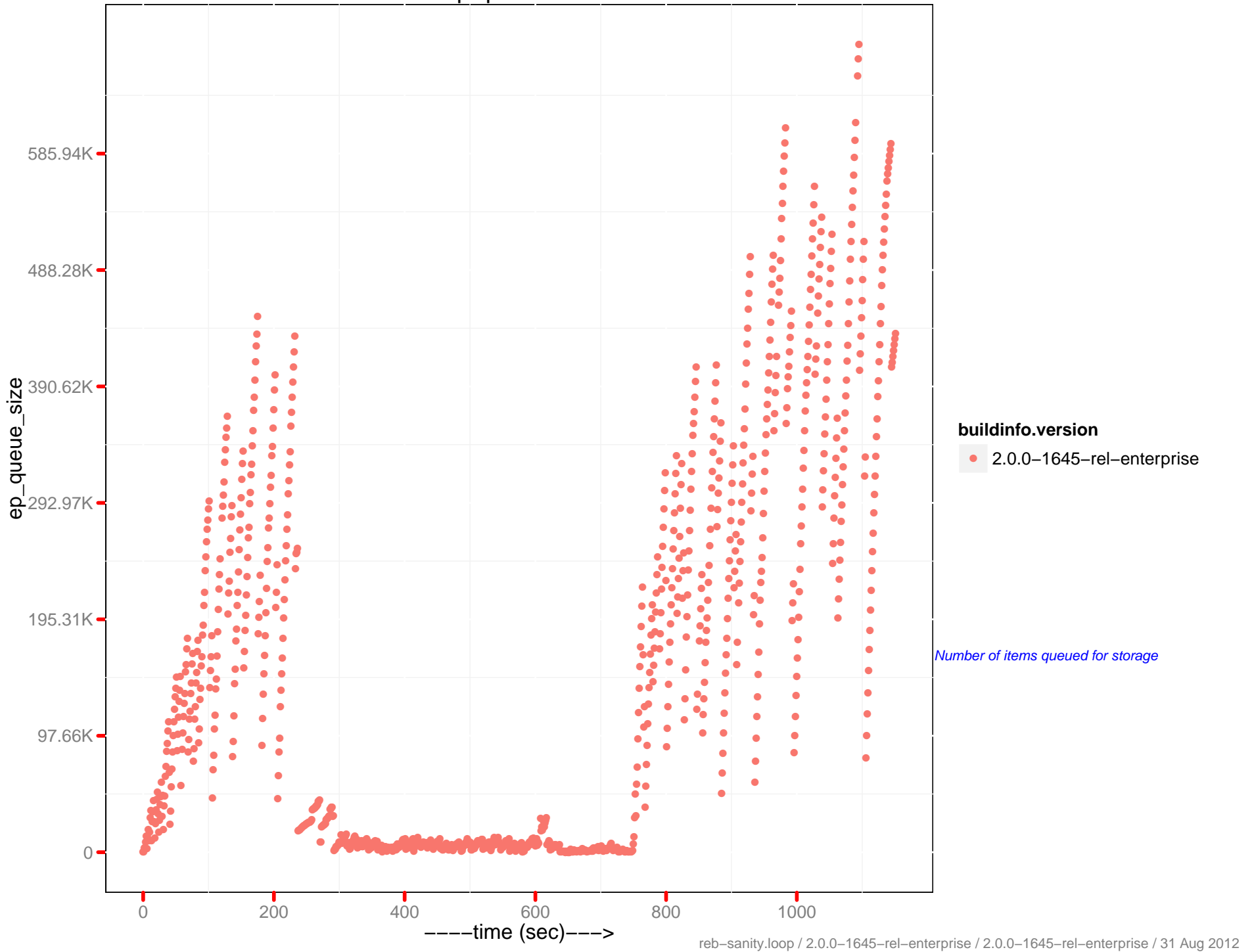


	2.0.0 – 1645	2.0.0 – 1645
<i>Runtime (in hr)</i>	0.32	NA
<i>Avg. Drain Rate</i>	9.48K	NANA
<i>Peak Disk (GB)</i>	6.33	NA
<i>Peak Memory (GB)</i>	2.71	NA
<i>Avg. OPS</i>	11.18K	NANA
<i>Avg. mem memcached (GB)</i>	1.65	NA
<i>Avg. mem beam.smp (MB)</i>	375.72	NA
<i>Avg. CPU rate (%)</i>	40.97	NA
<i>Latency-get (90th) (ms)</i>	0.88	NA
<i>Latency-get (95th) (ms)</i>	1.68	NA
<i>Latency-get (99th) (ms)</i>	4.46	NA
<i>Latency-set (90th) (ms)</i>	0.85	NA
<i>Latency-set (95th) (ms)</i>	1.65	NA
<i>Latency-set (99th) (ms)</i>	5.27	NA
<i>Latency-query (80th) (ms)</i>	NA	NA
<i>Latency-query (90th) (ms)</i>	NA	NA
<i>Latency-query (95th) (ms)</i>	NA	NA
<i>Latency-query (99th) (ms)</i>	NA	NA
<i>Latency-query (99.9th) (ms)</i>	NA	NA
<i>Avg. QPS</i>	0	NA
<i>Rebalance Time (sec)</i>	528.89	NA
<i>Testrunner Version</i>	94b0a6a	NA

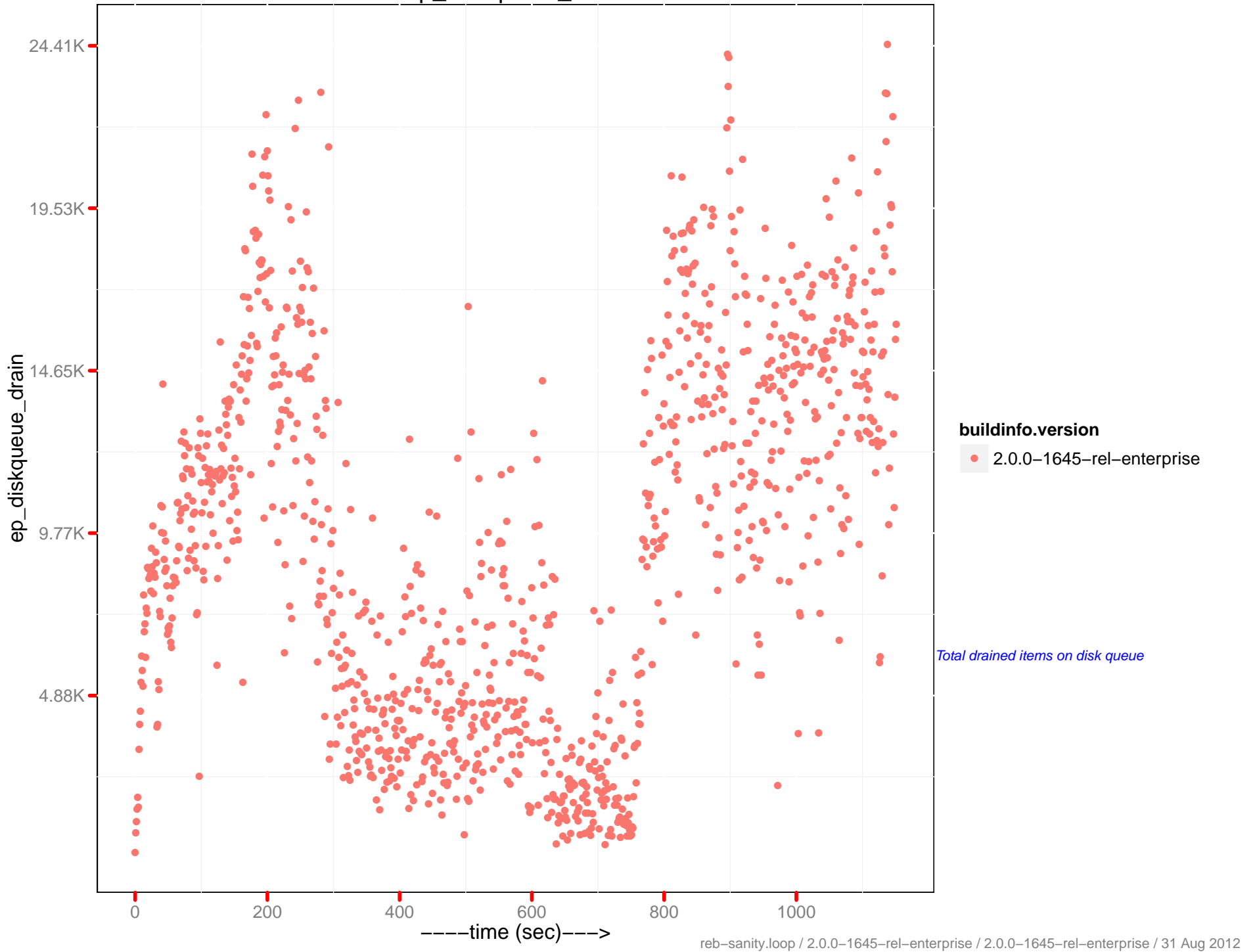
View read per sec.



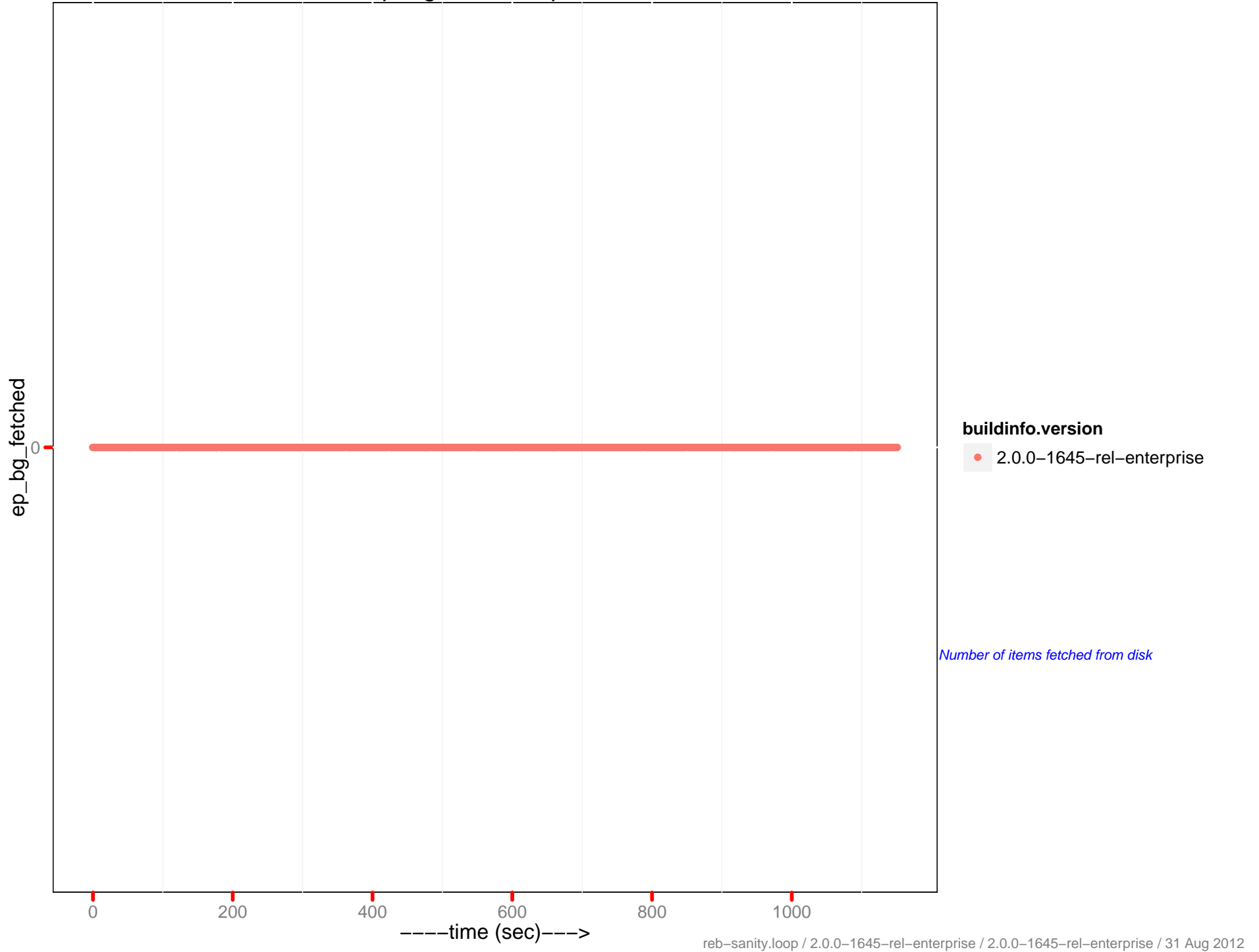
ep queue size



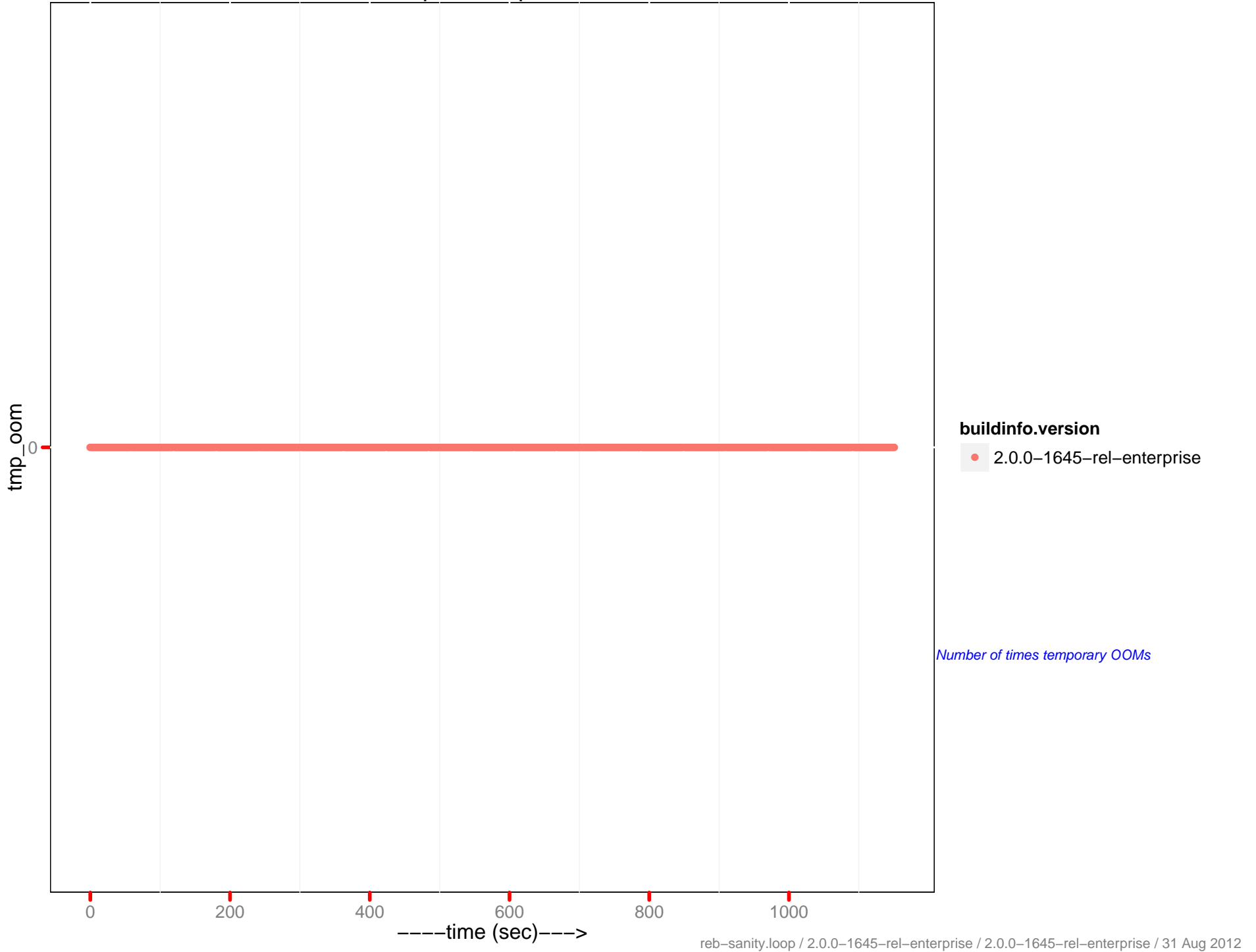
ep_diskqueue_drain



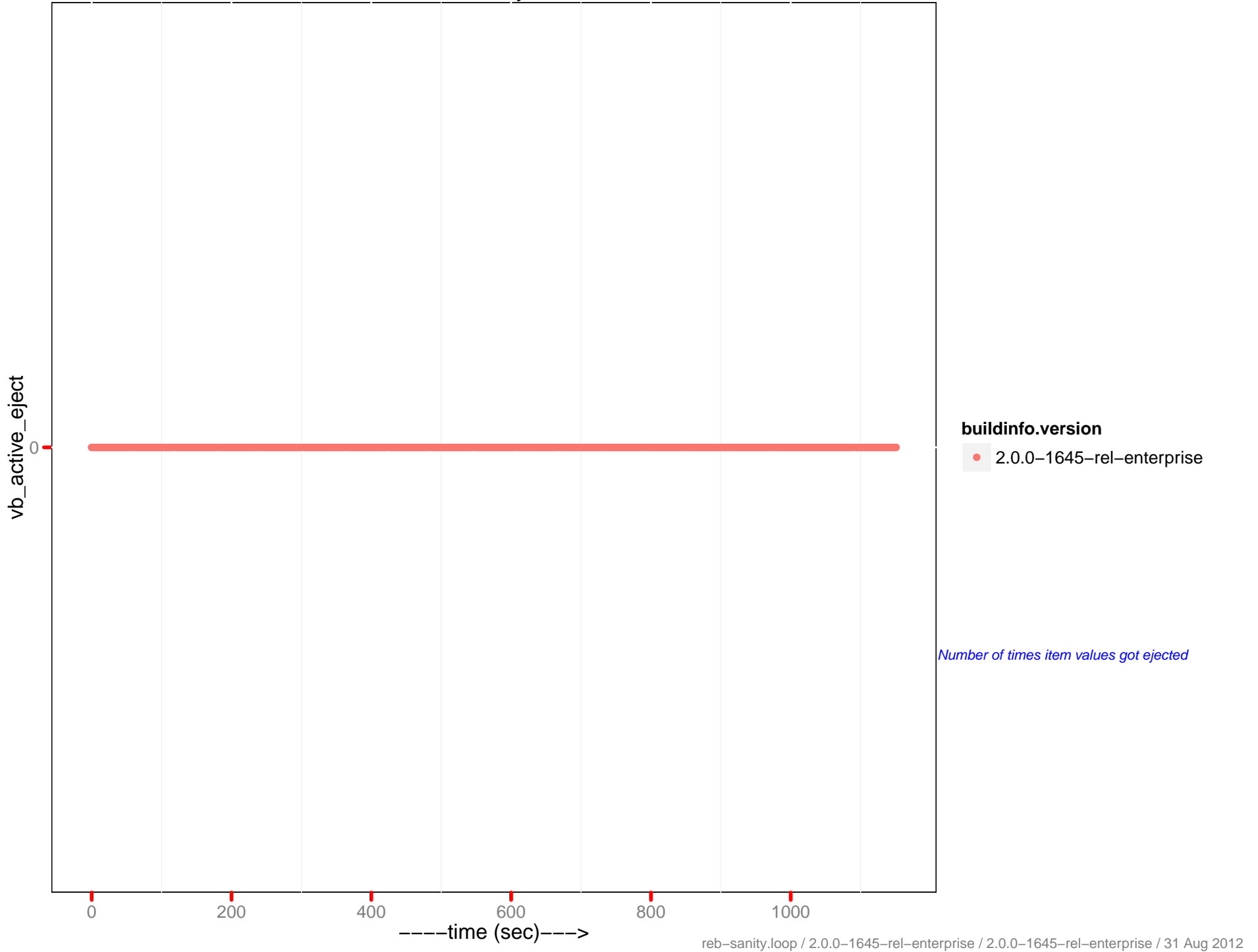
ep_bg_fetched ops/sec



tmp_oom ops/sec



vb_active_eject/sec

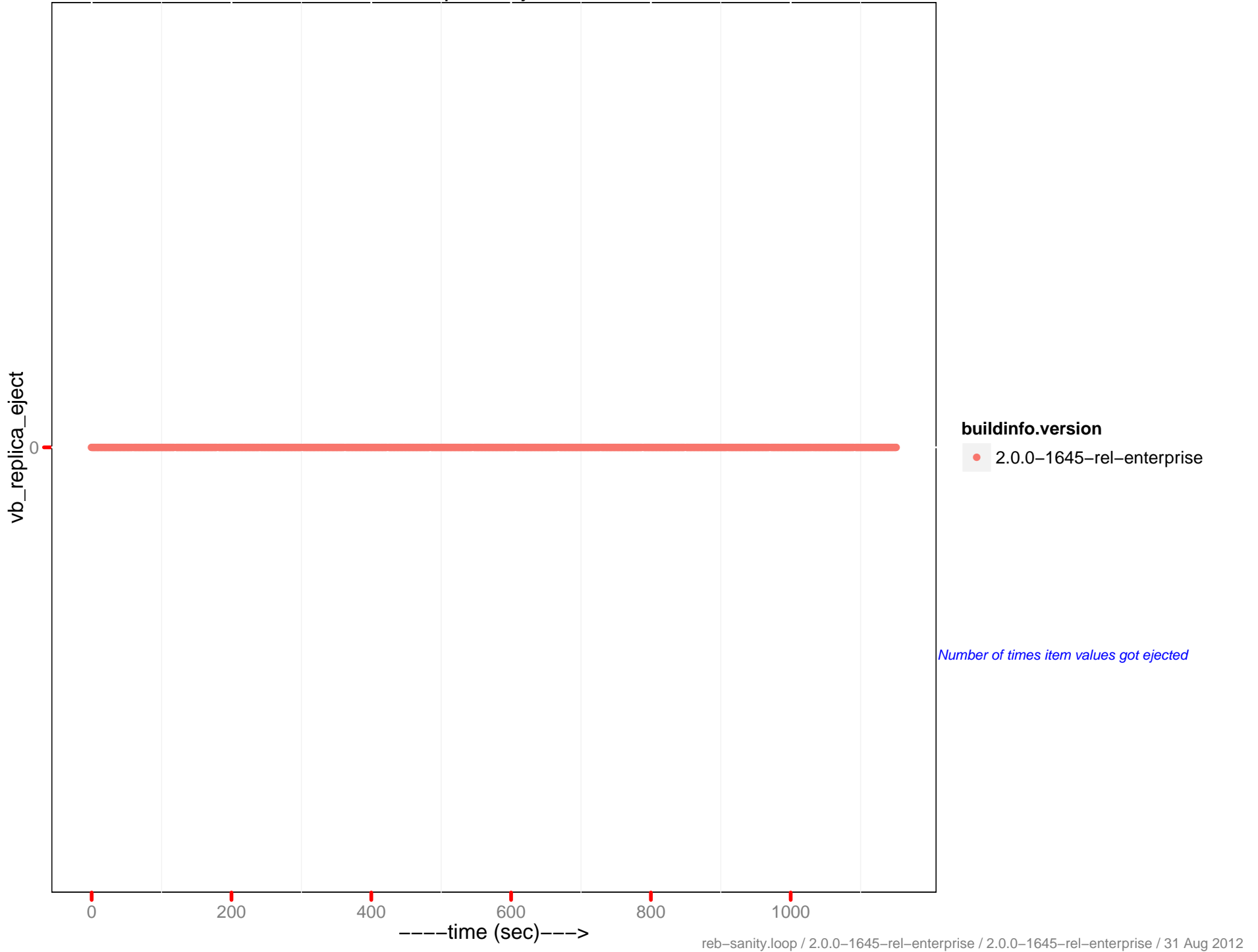


buildinfo.version

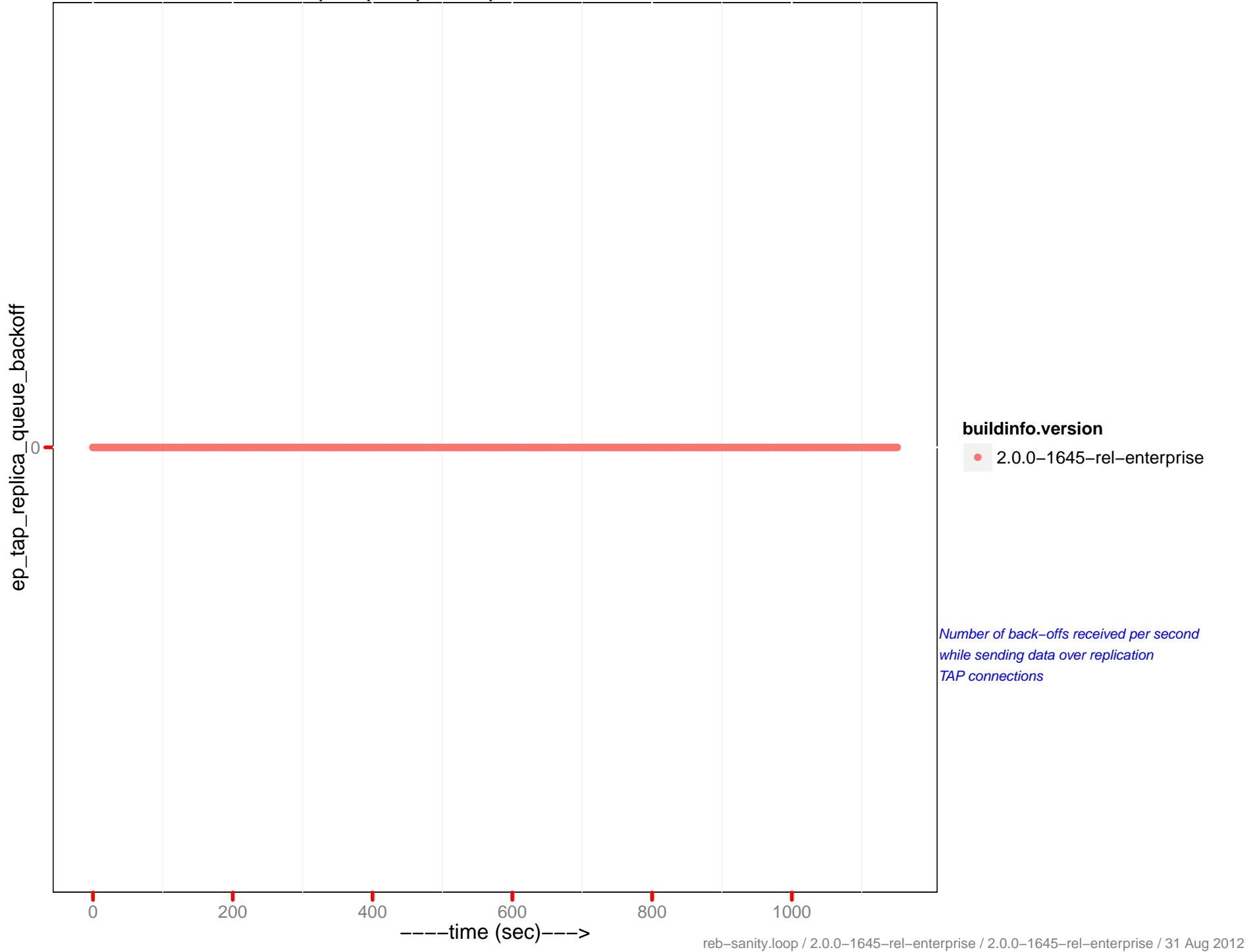
- 2.0.0-1645-rel-enterprise

Number of times item values got ejected

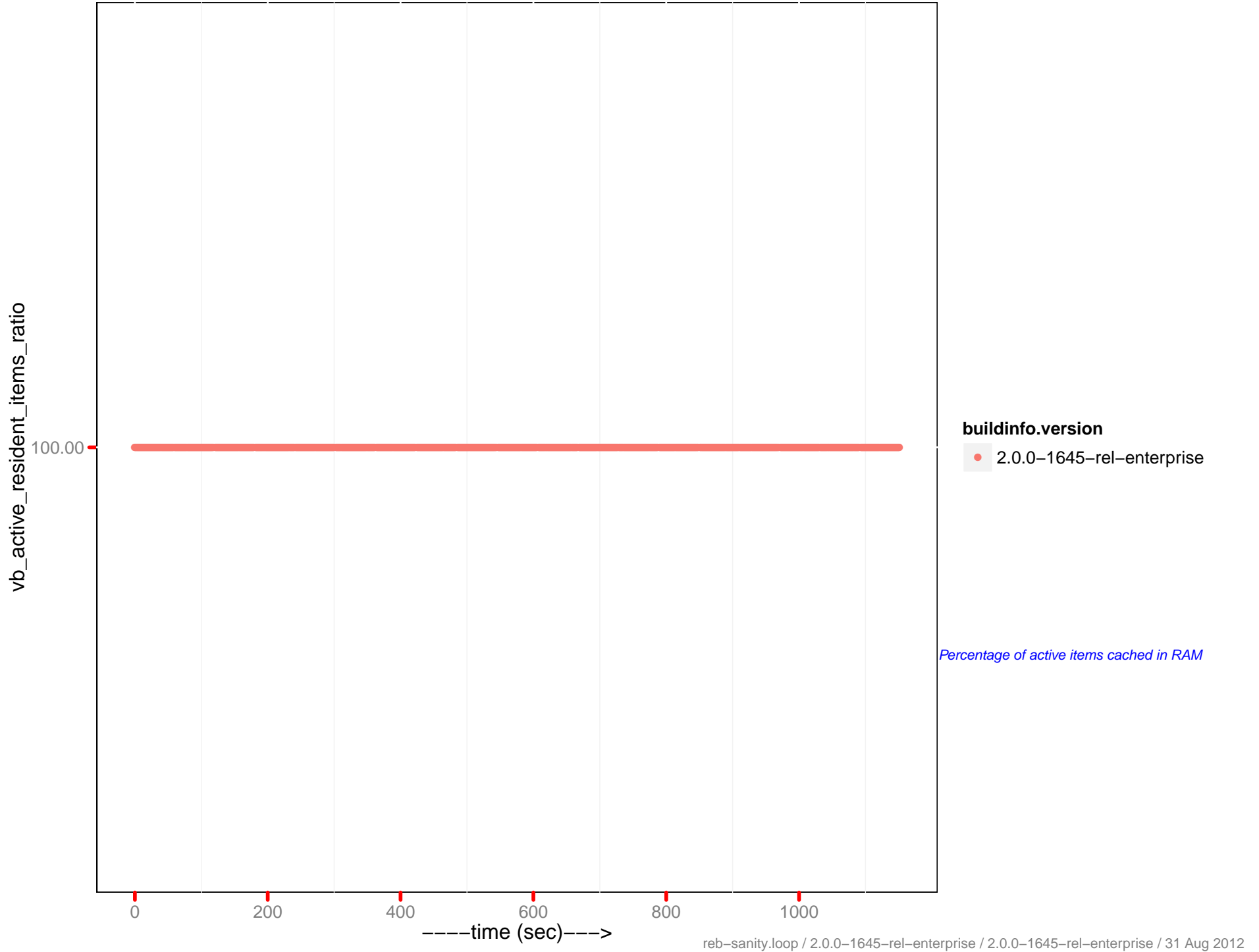
vb_replica_eject/sec



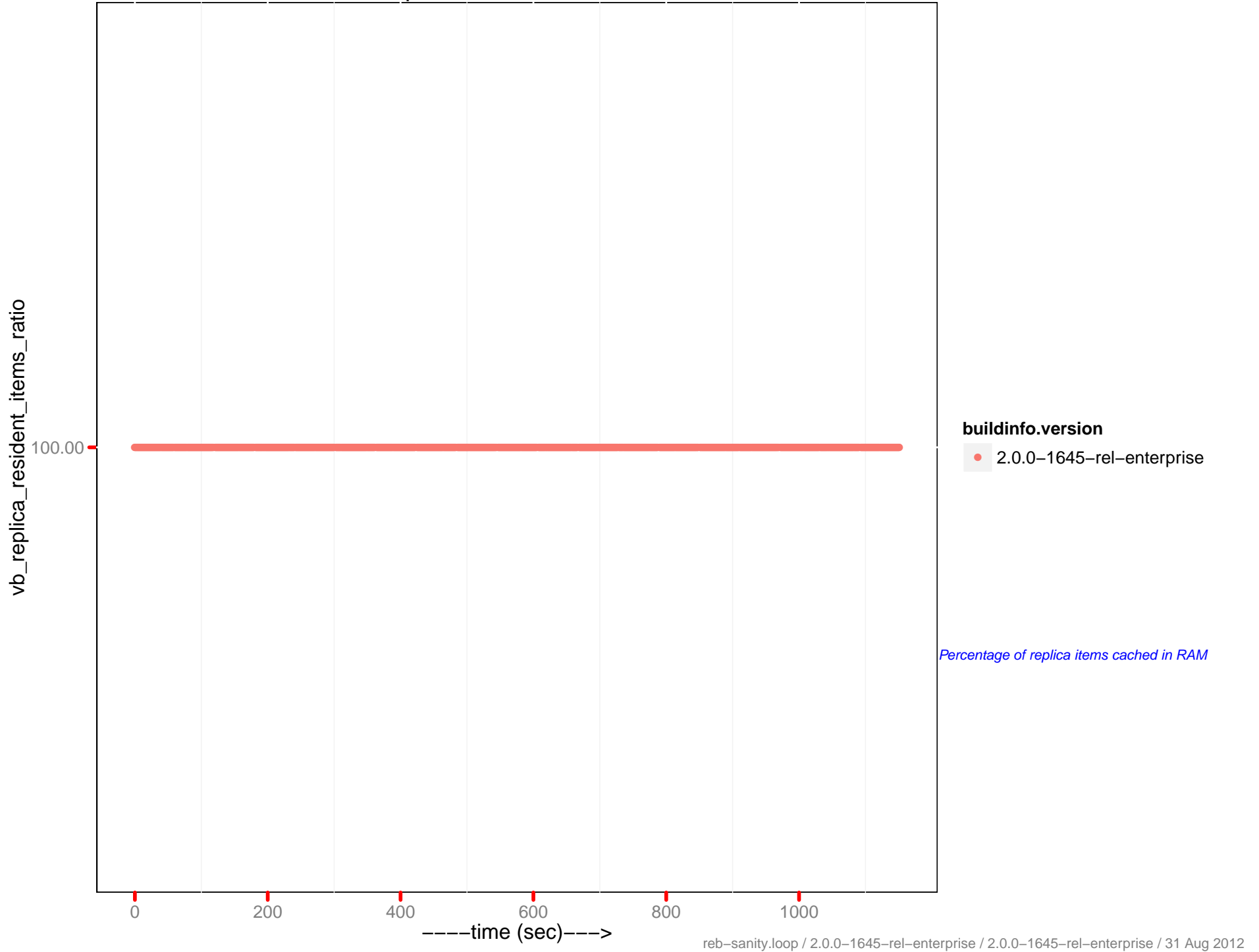
ep_tap_replica_queue_backoff/sec



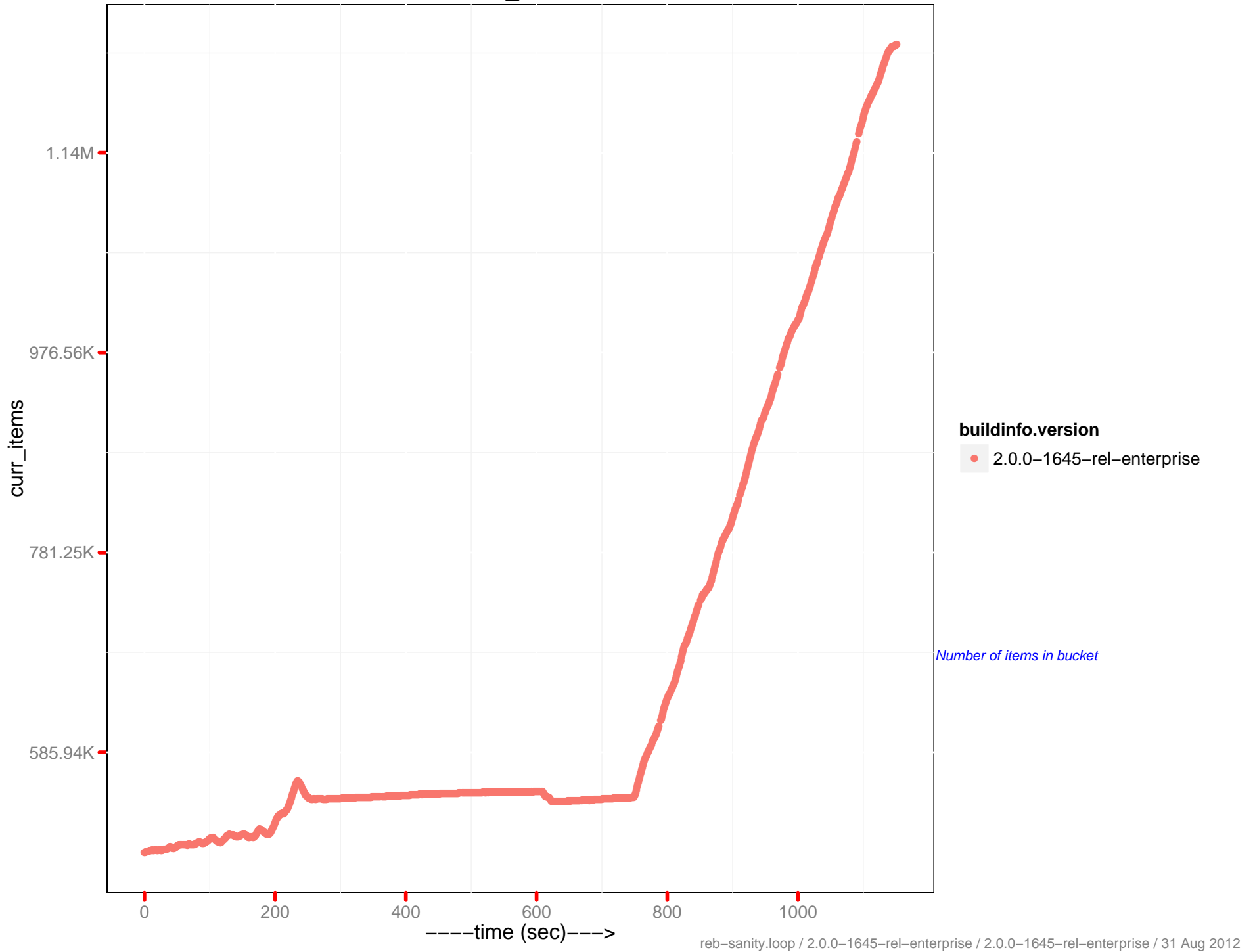
vb_active_resident_items_ratio



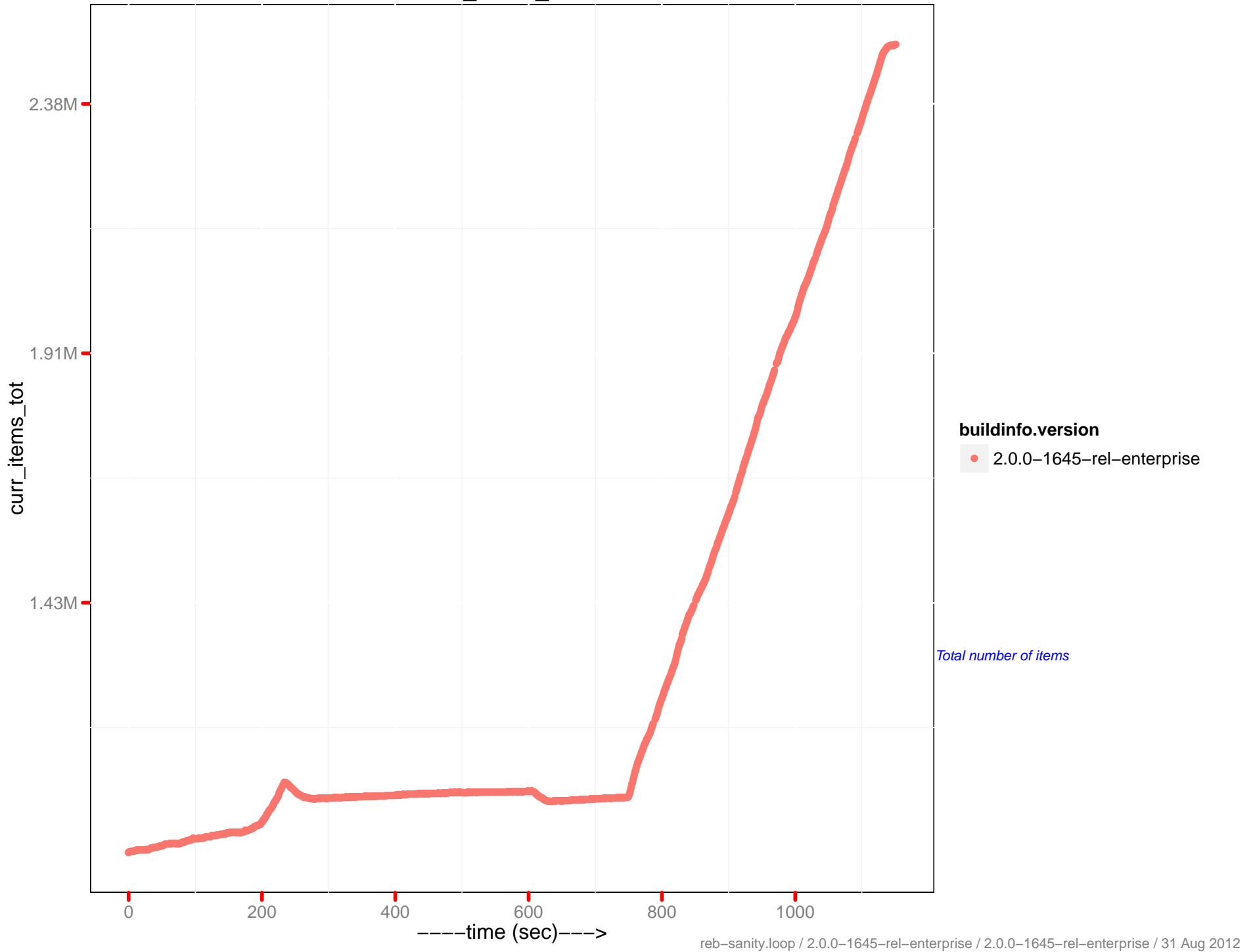
vb_replica_resident_items_ratio



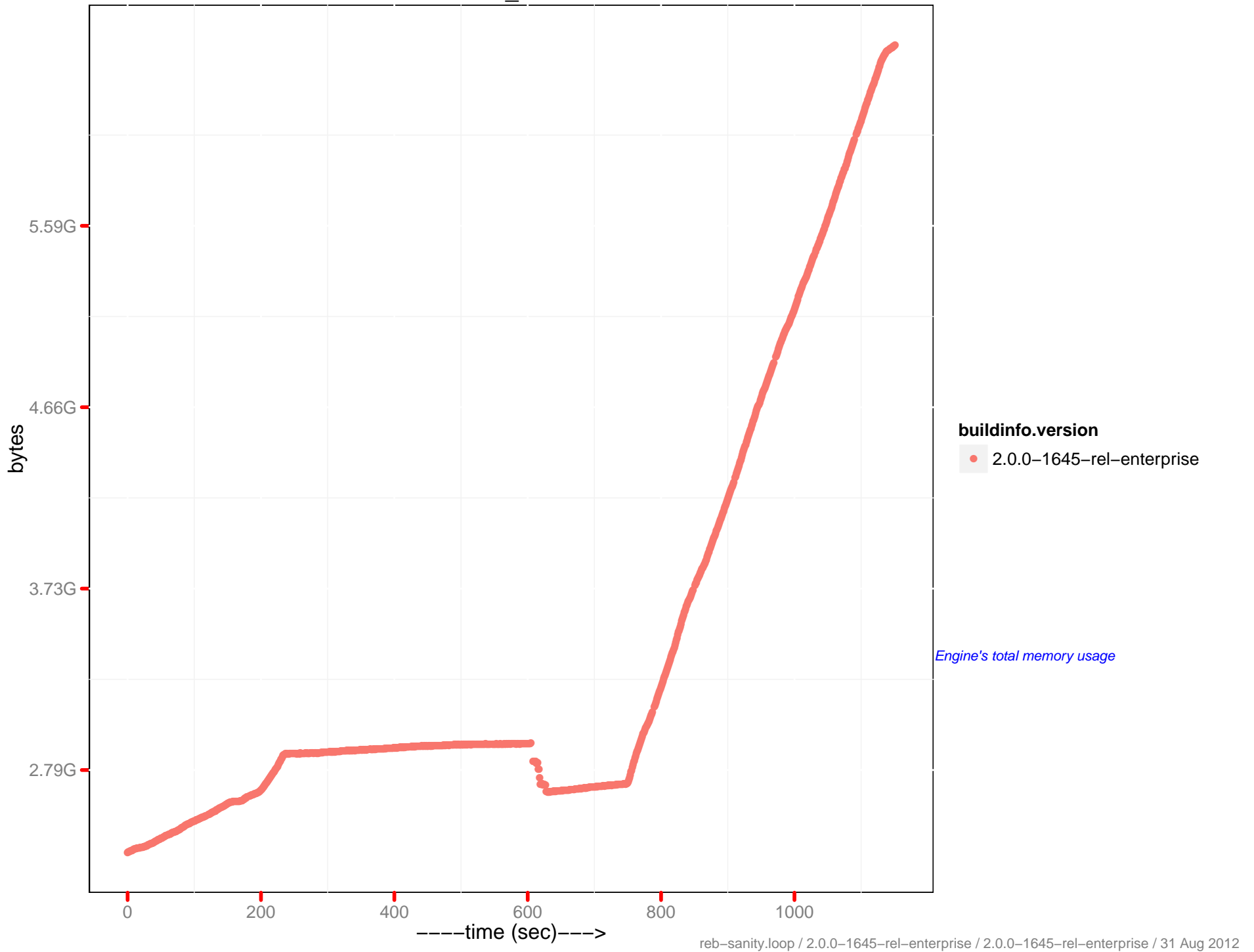
curr_items



cur_items_total

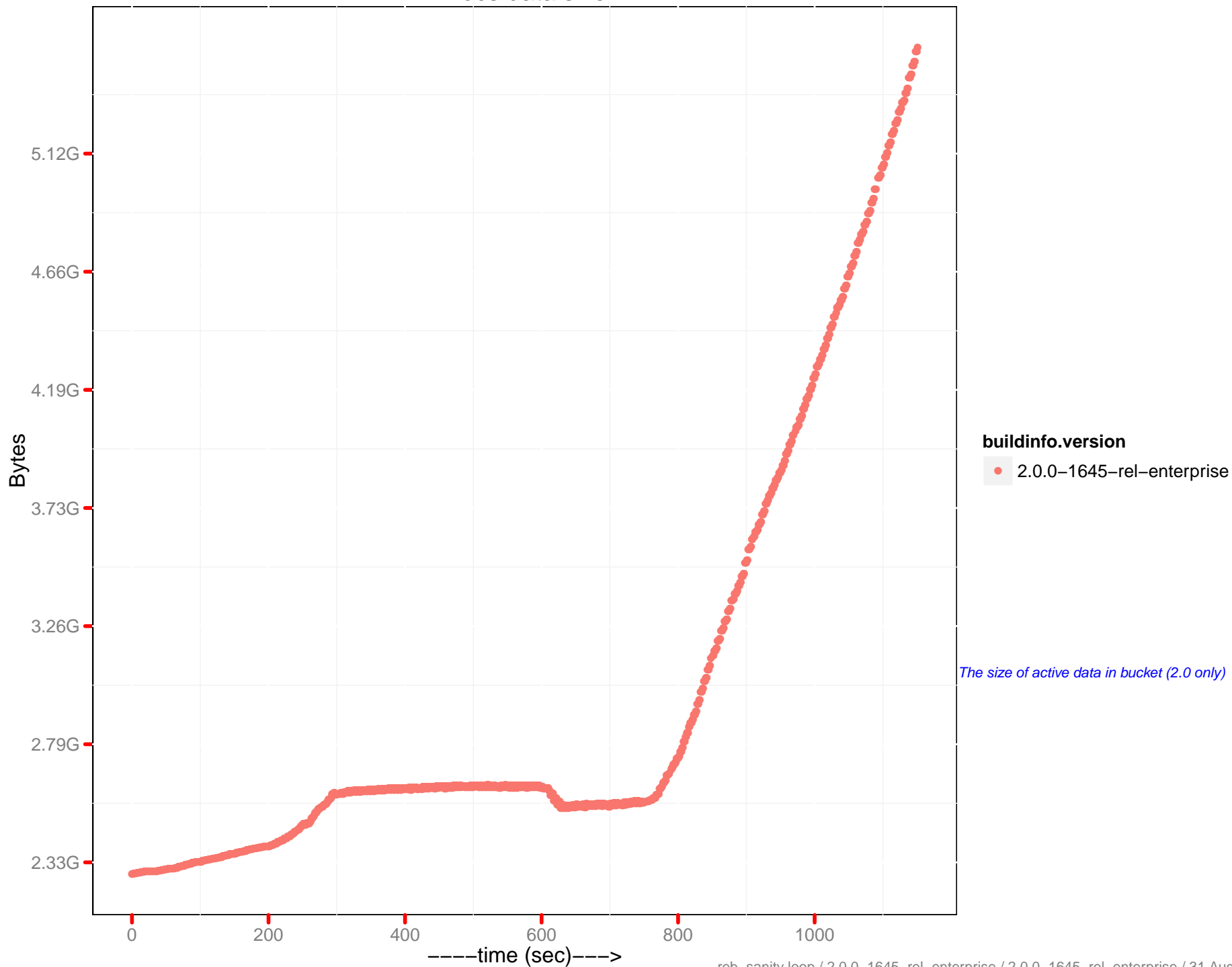


mem_used



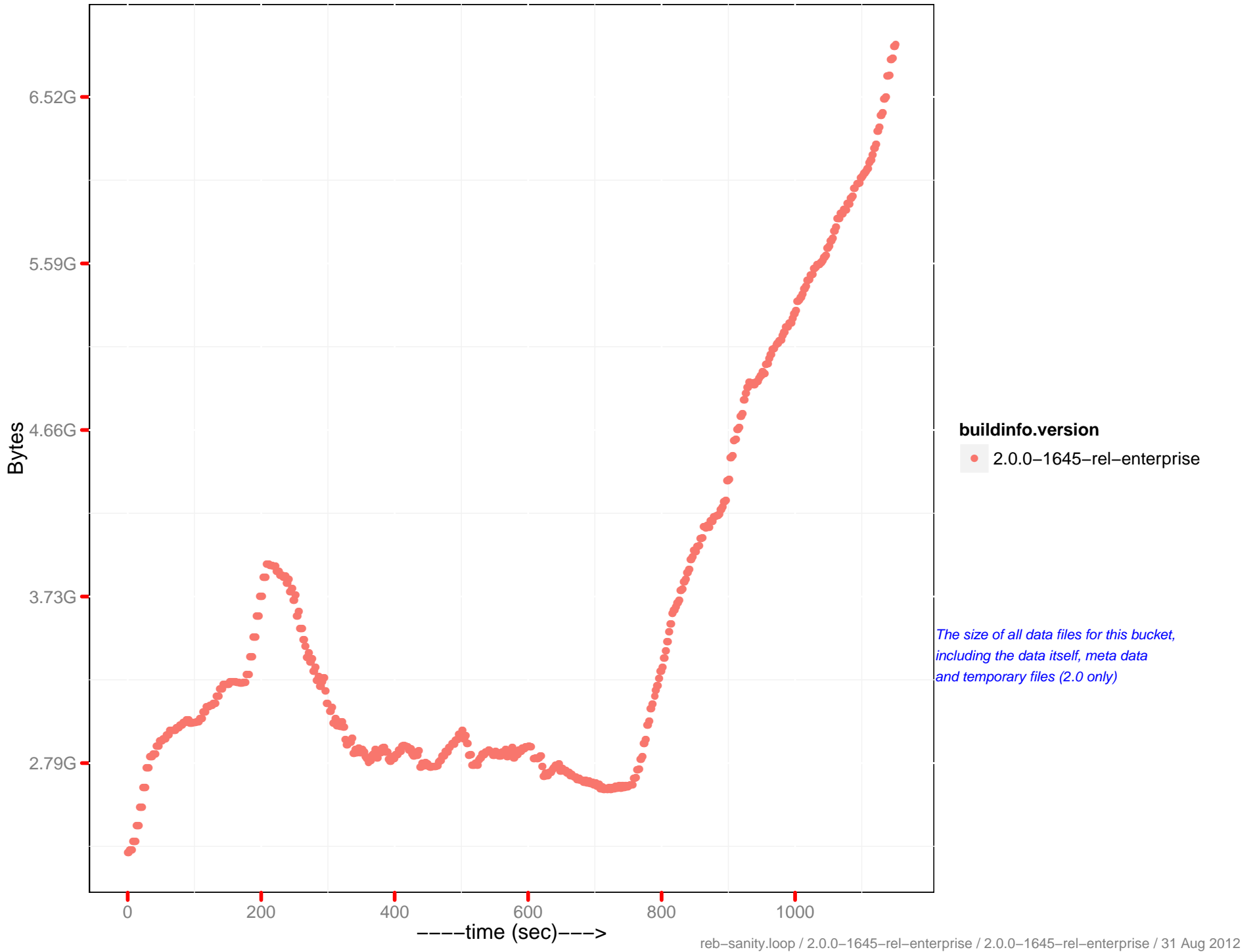
Engine's total memory usage

Docs data size

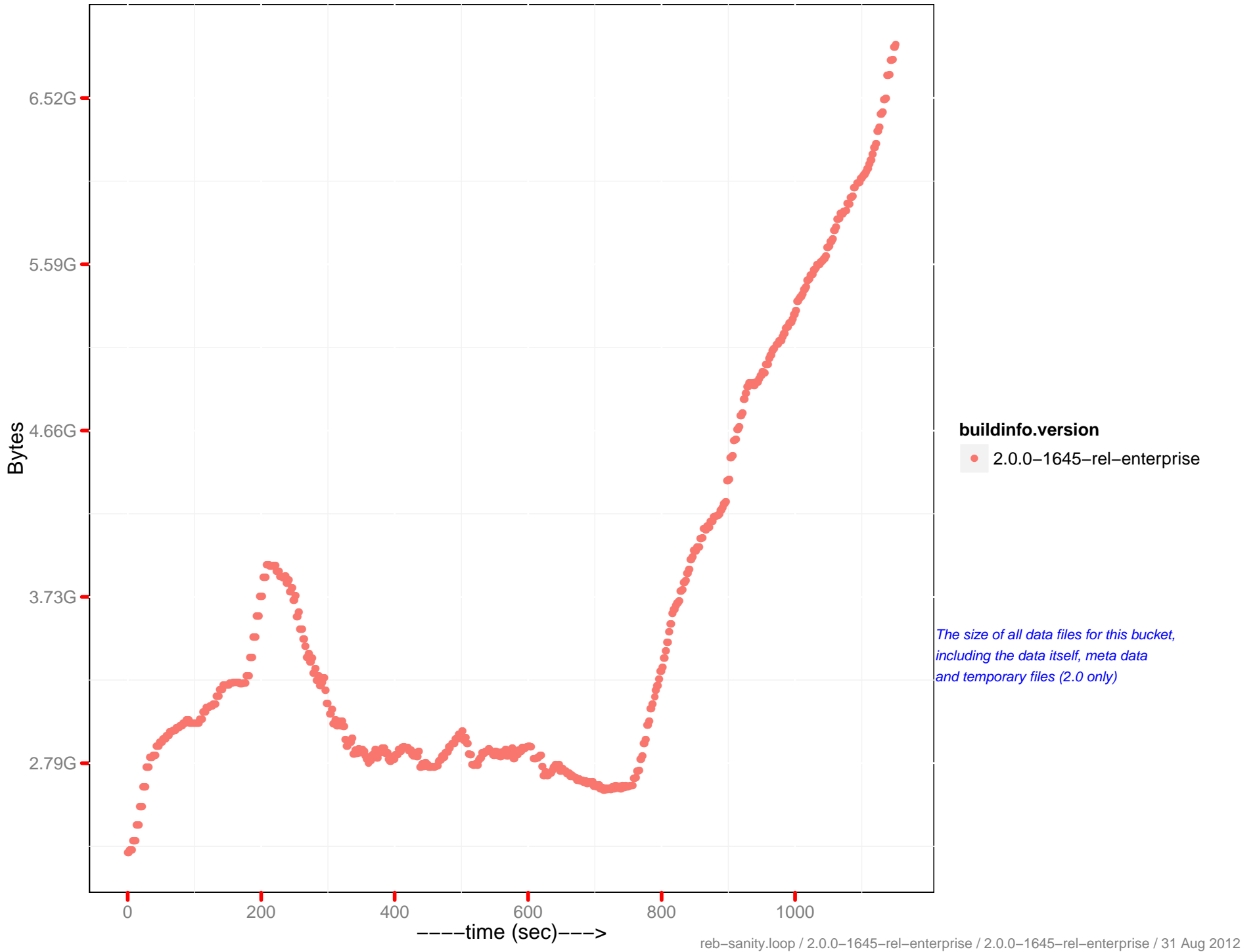


The size of active data in bucket (2.0 only)

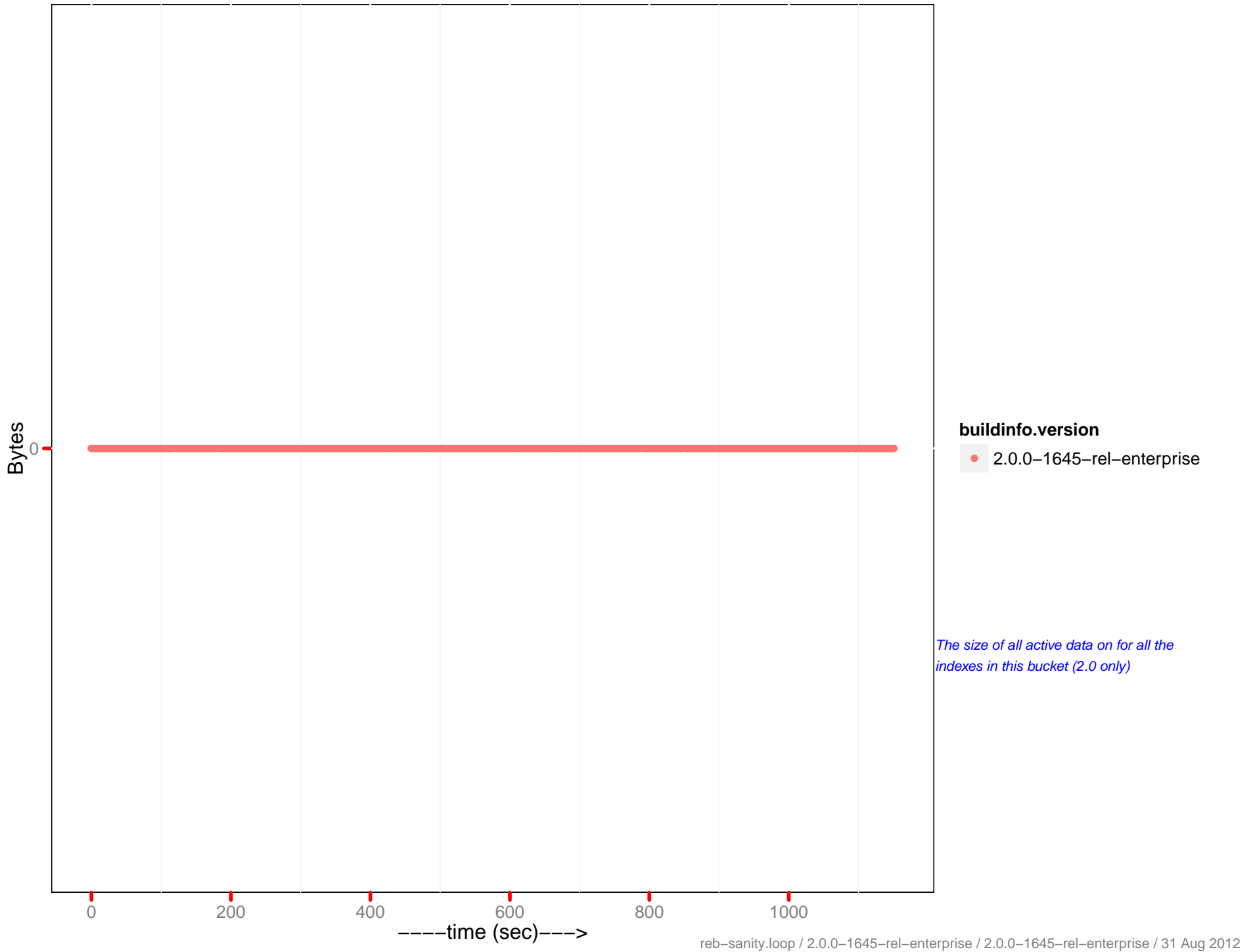
Docs disk size



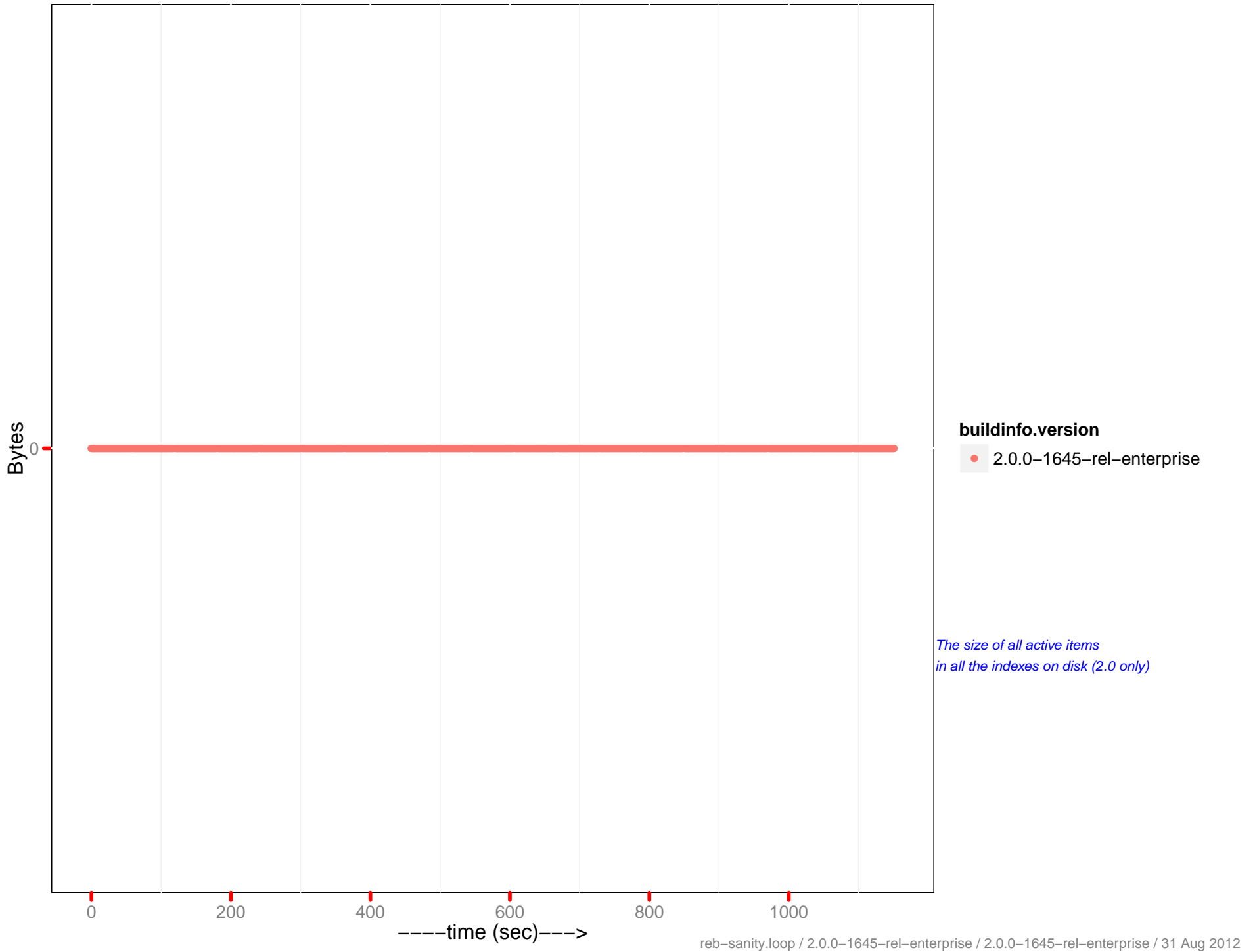
Docs actual disk size



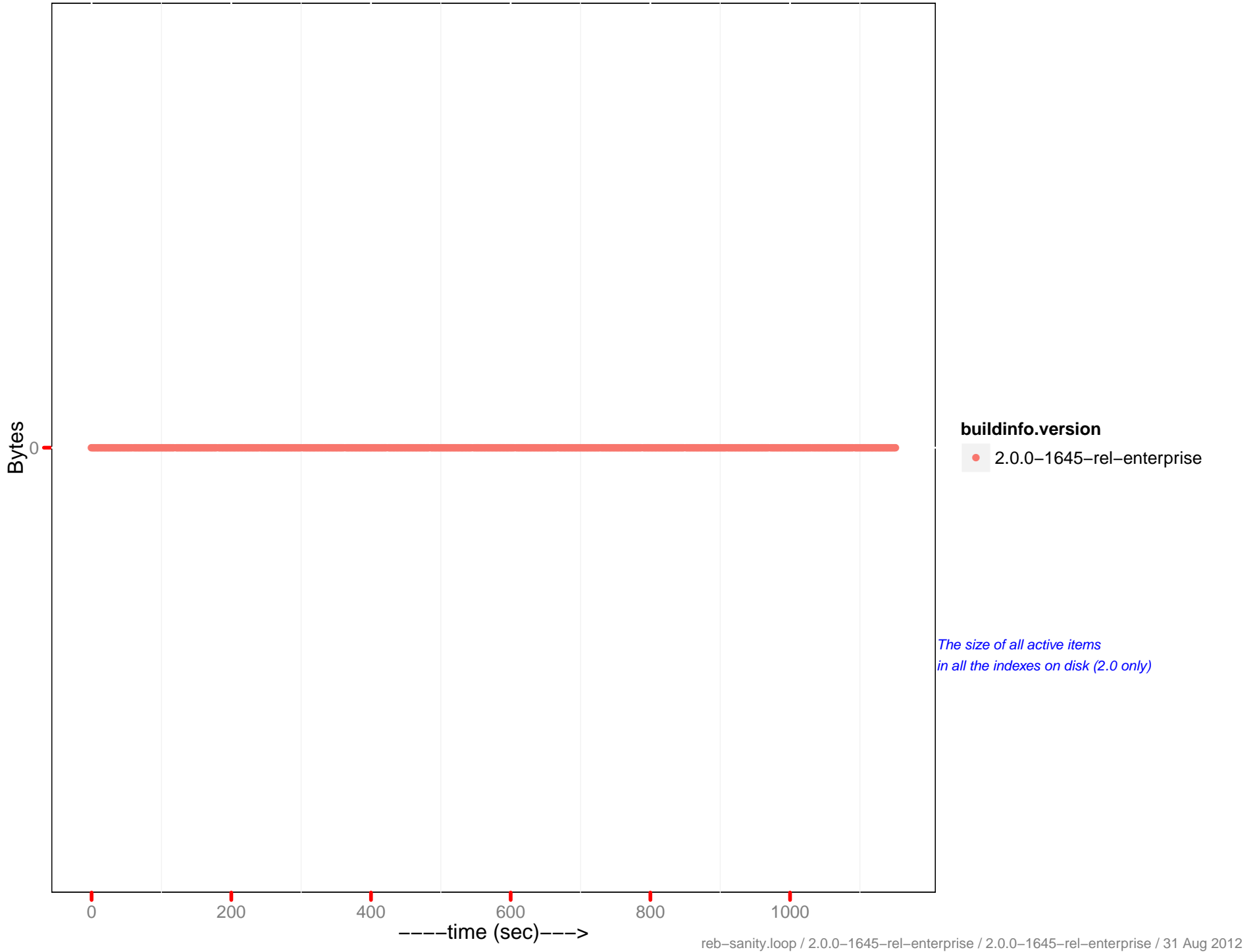
Views data size



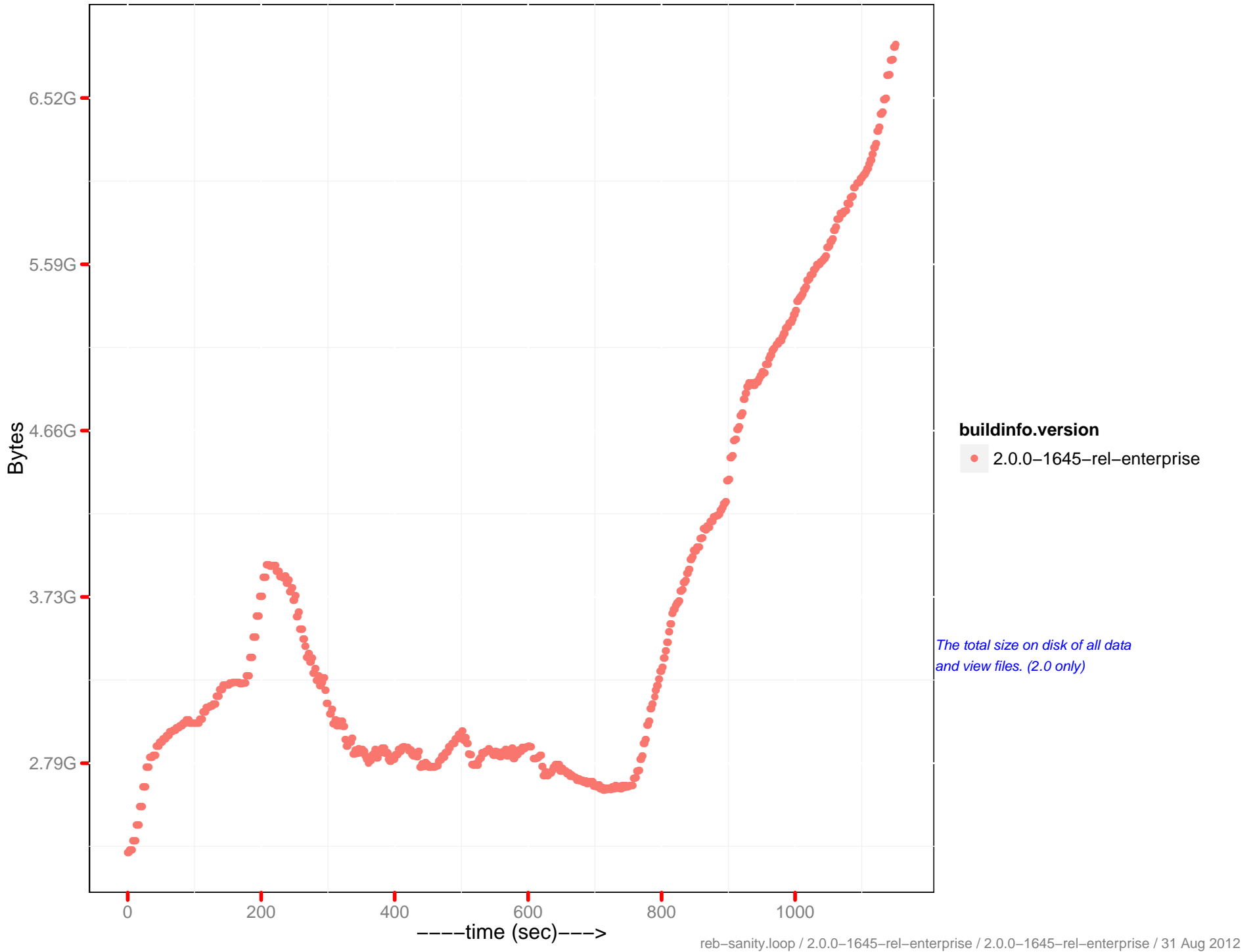
Views disk size



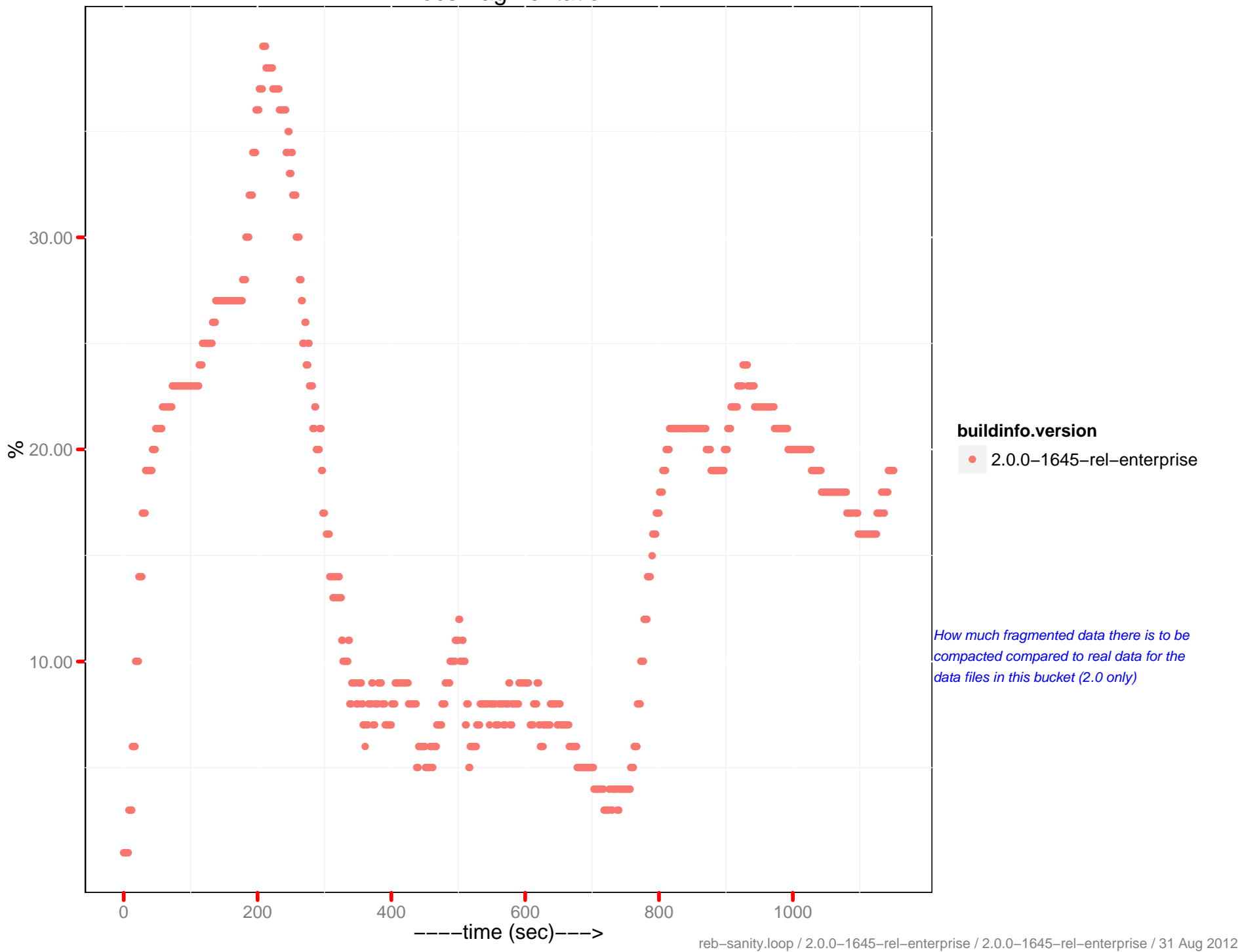
Views actual disk size



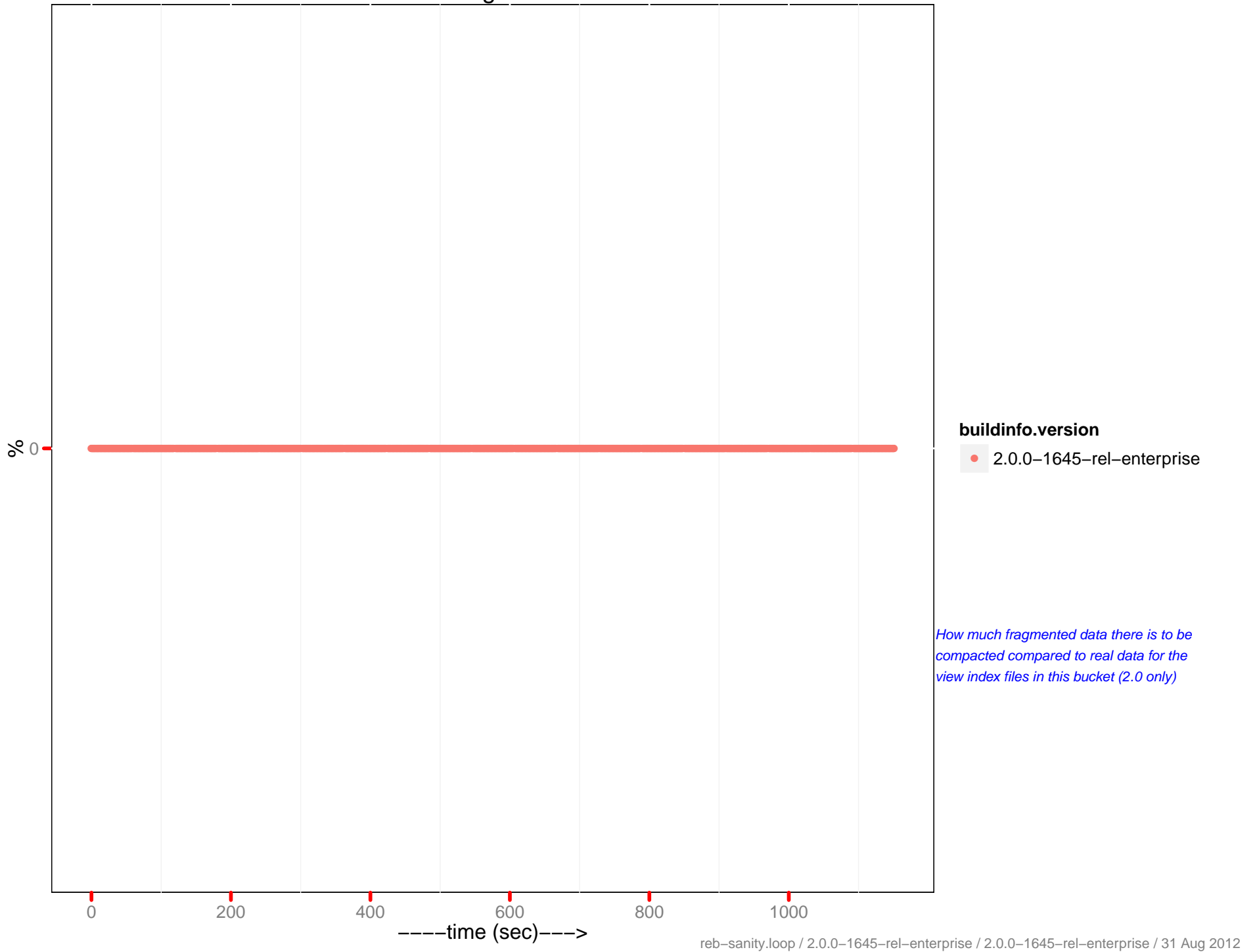
Total disk size



Docs fragmentation



Views fragmentation

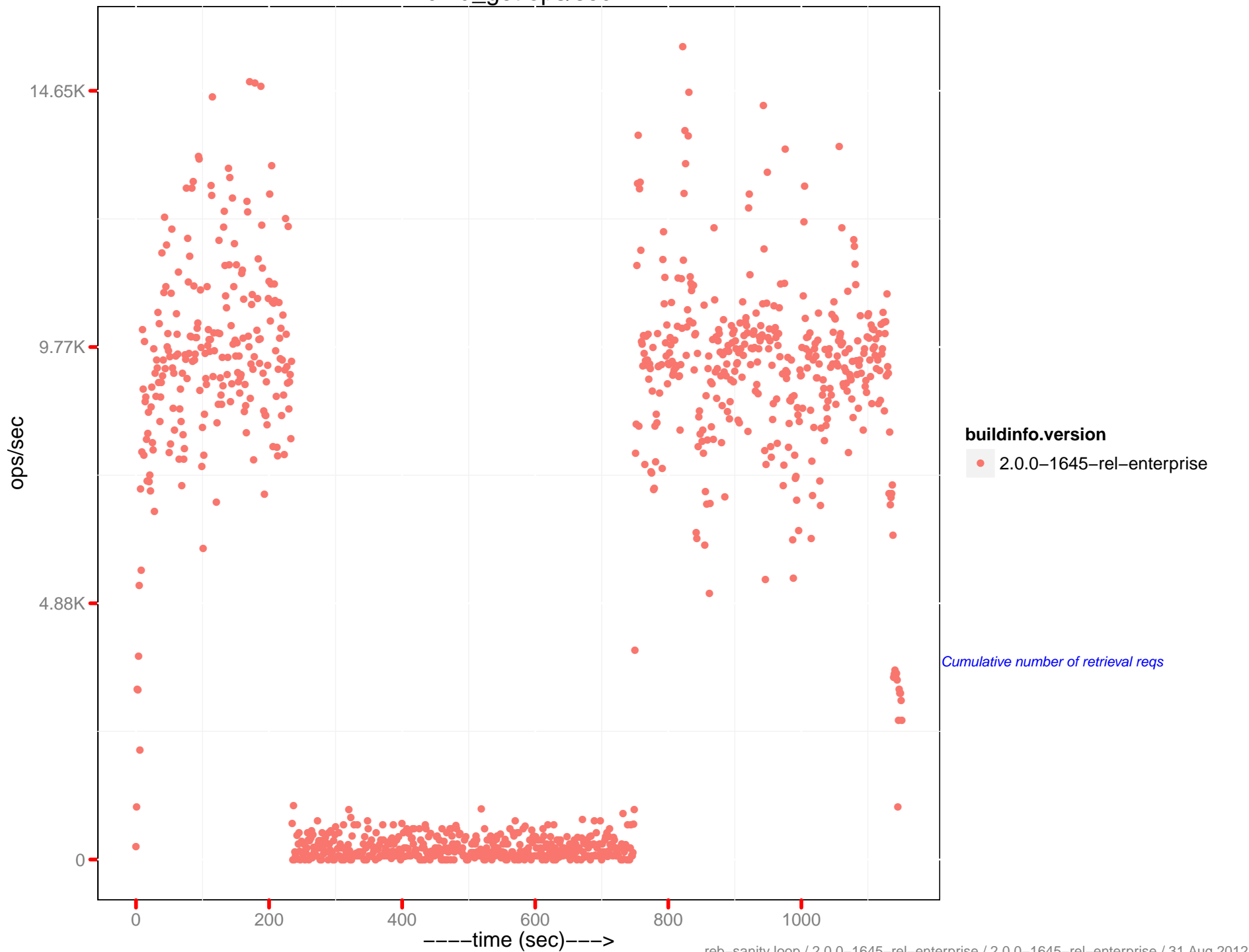


buildinfo.version

• 2.0.0-1645-rel-enterprise

How much fragmented data there is to be compacted compared to real data for the view index files in this bucket (2.0 only)

cmd_get ops/sec



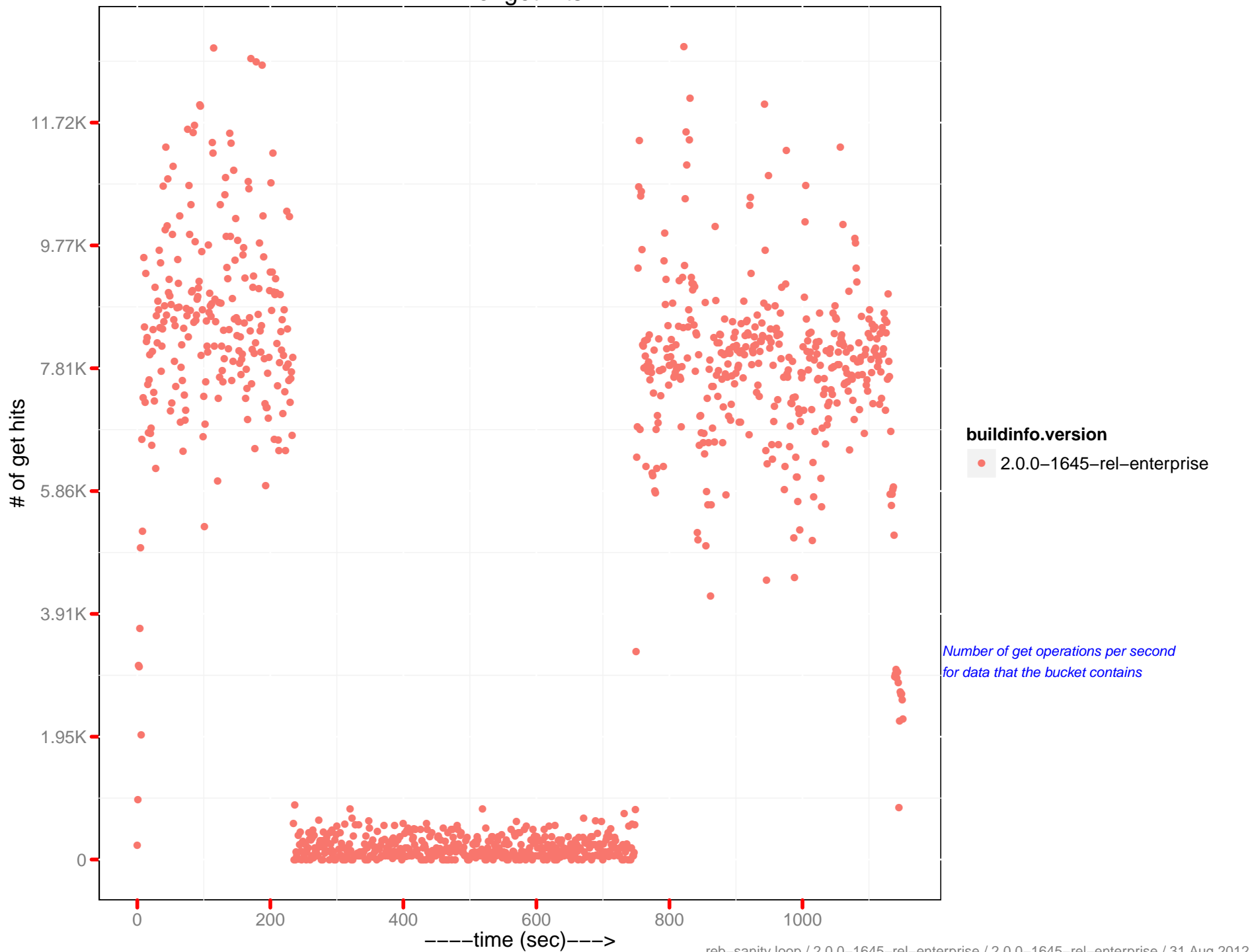
cmd_set ops/sec



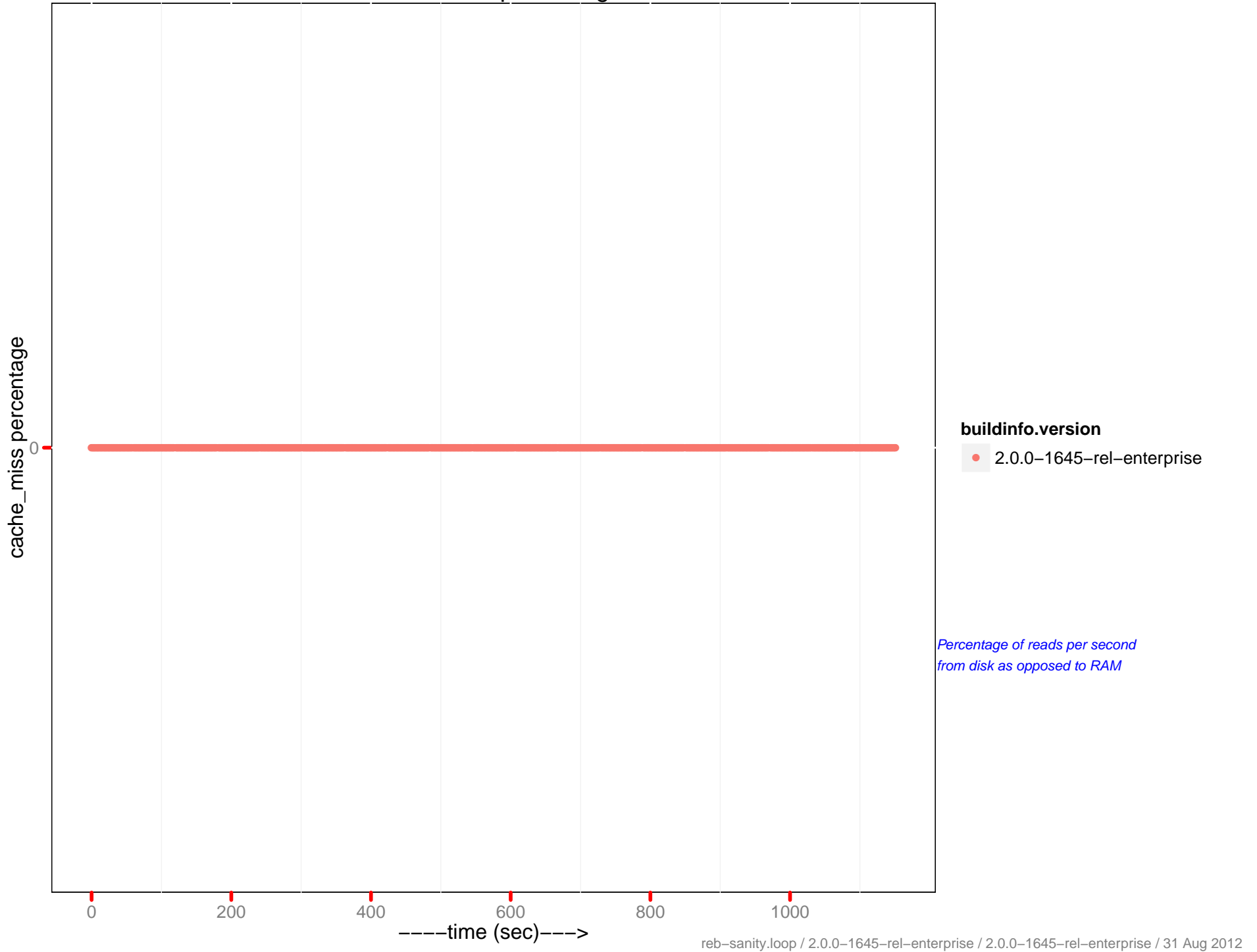
of get misses



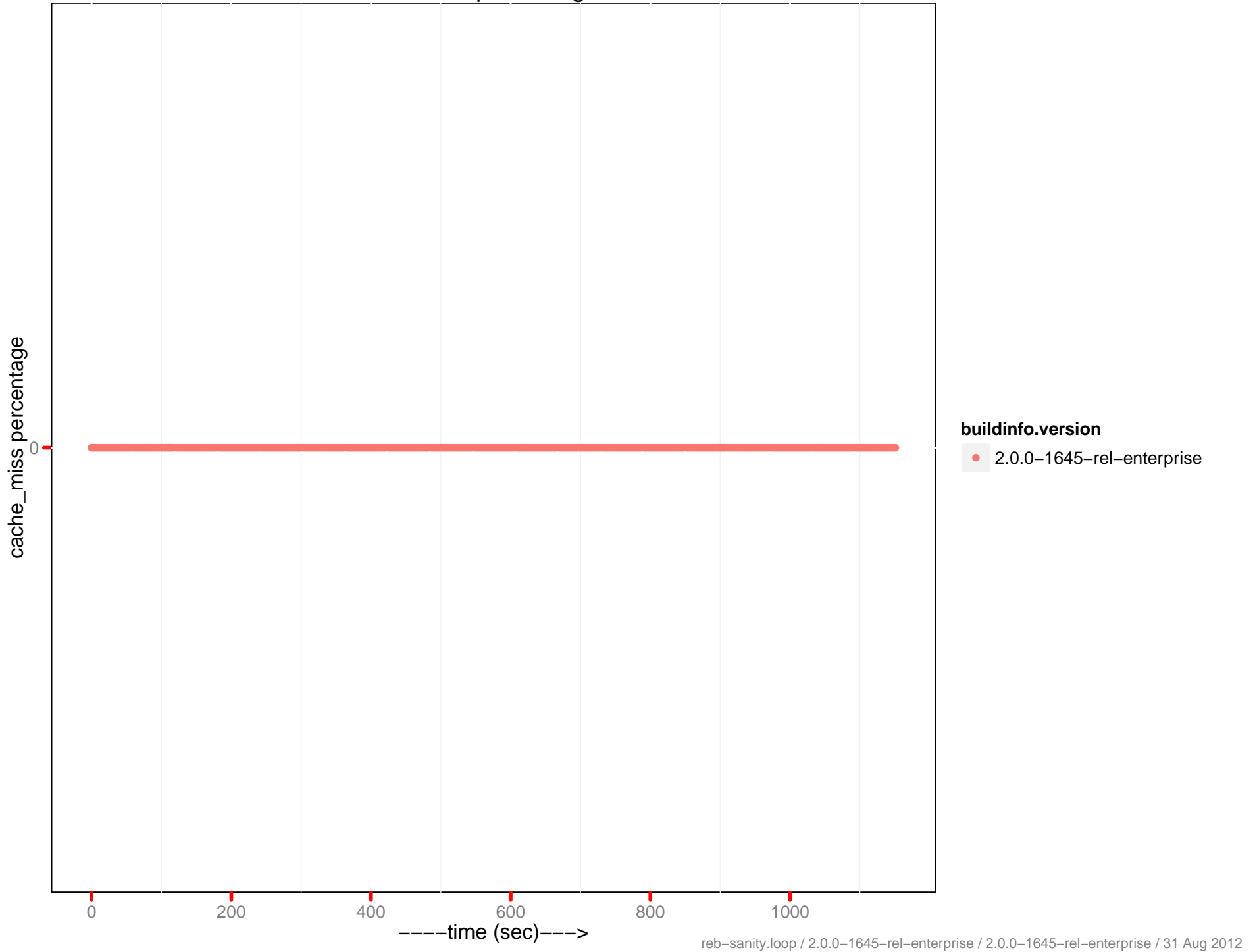
of get hits



cache_miss percentage



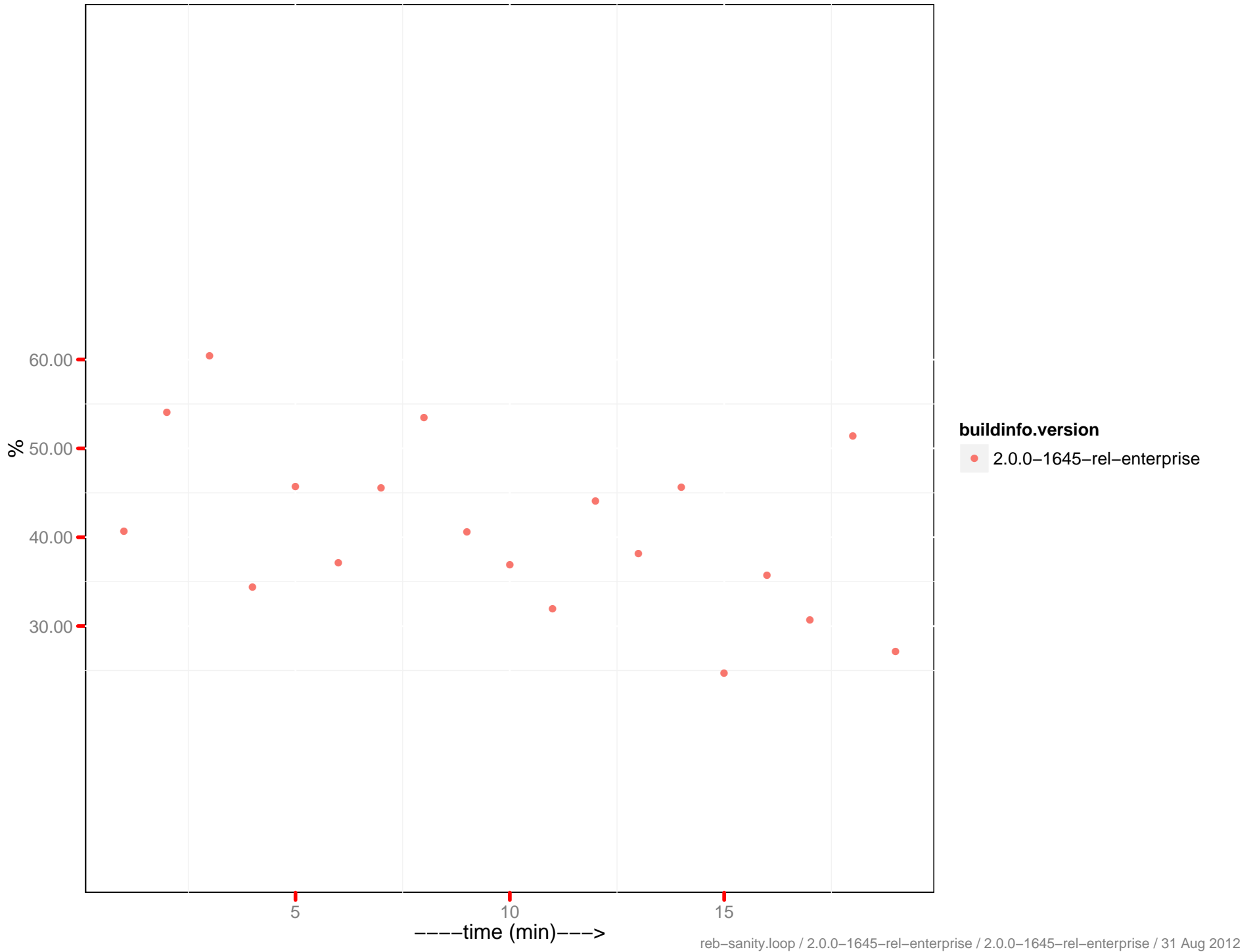
cache_miss percentage 0-5



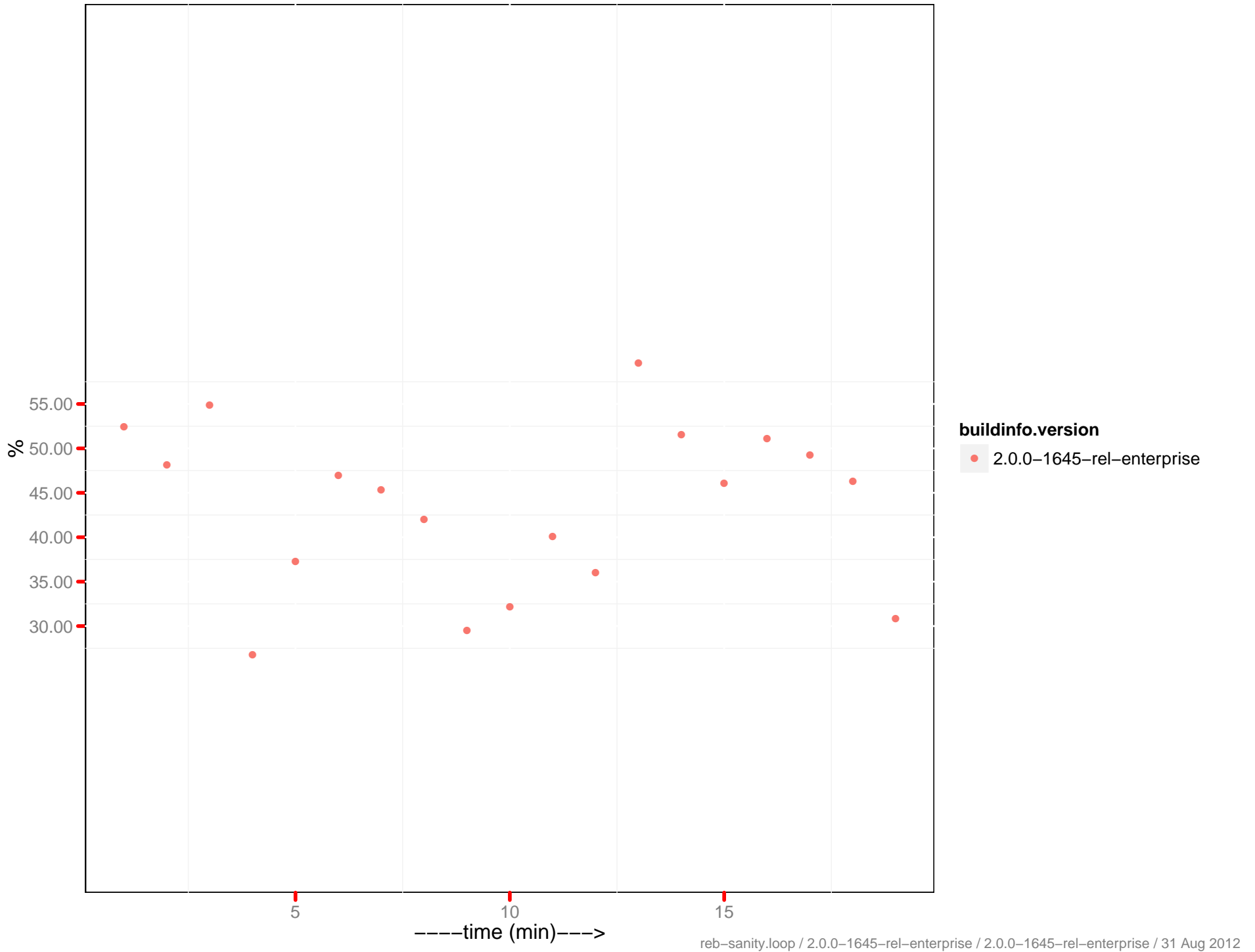
Number of connections



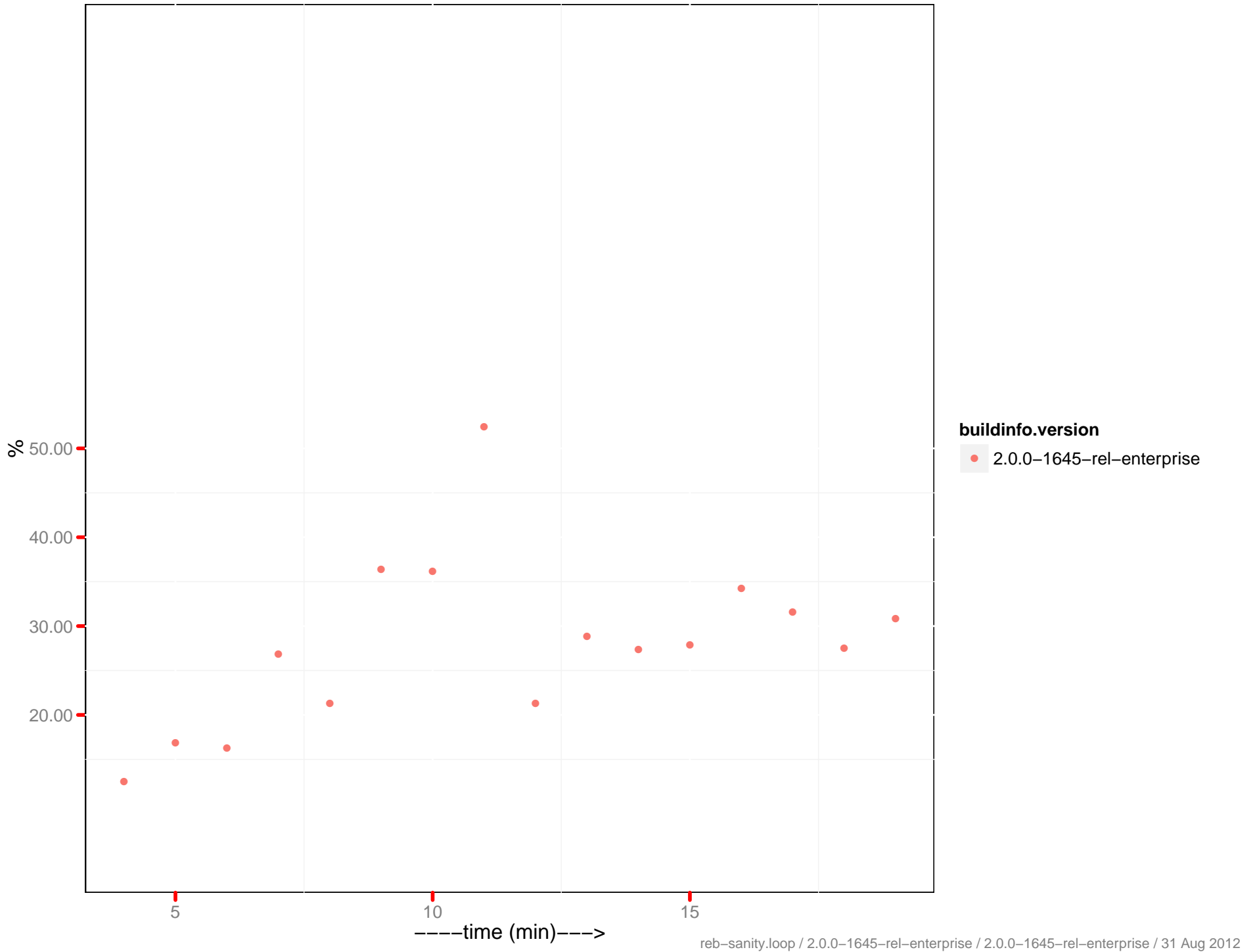
CPU utilization – 192.168.0.20:8091



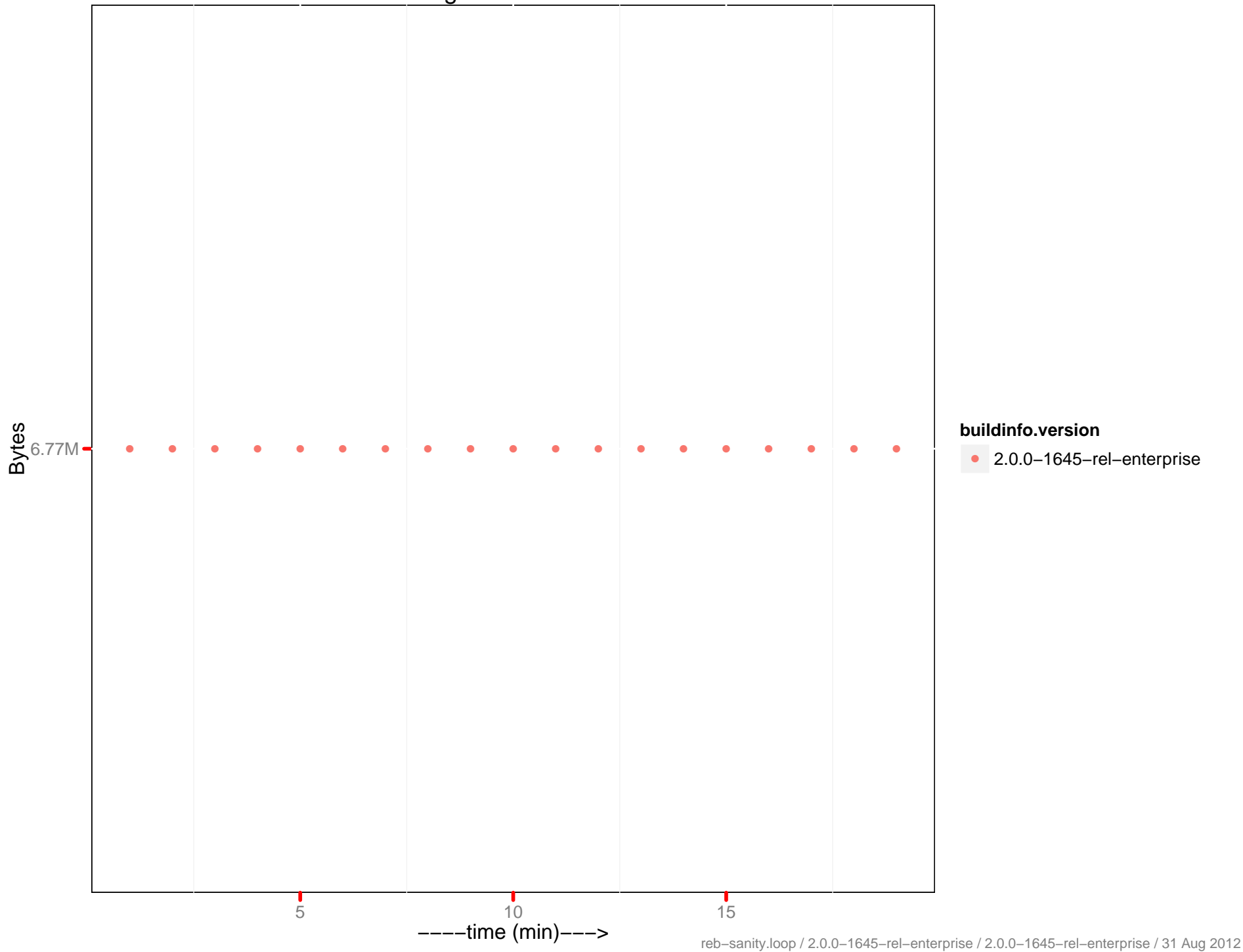
CPU utilization – 192.168.0.22:8091



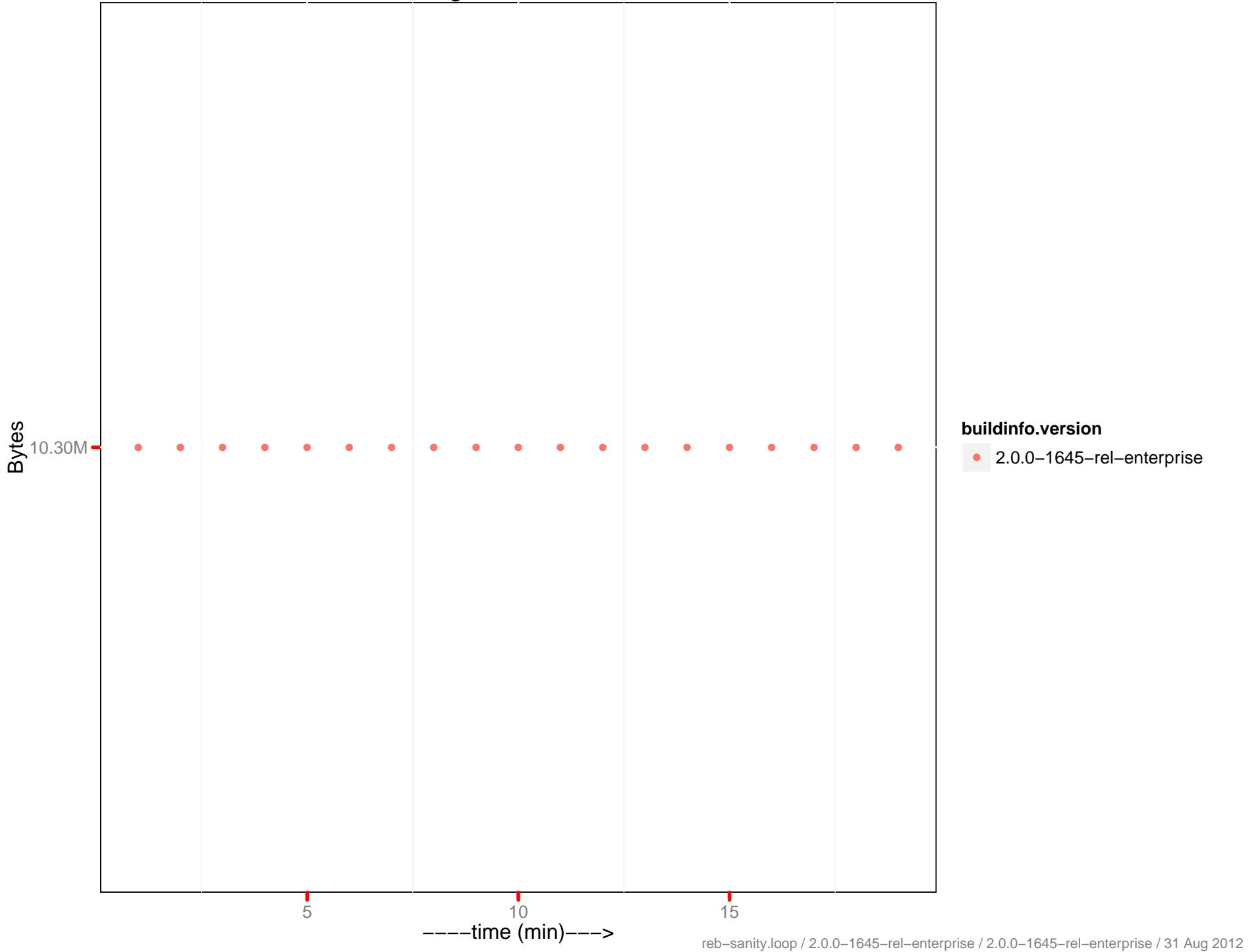
CPU utilization – 192.168.0.23:8091



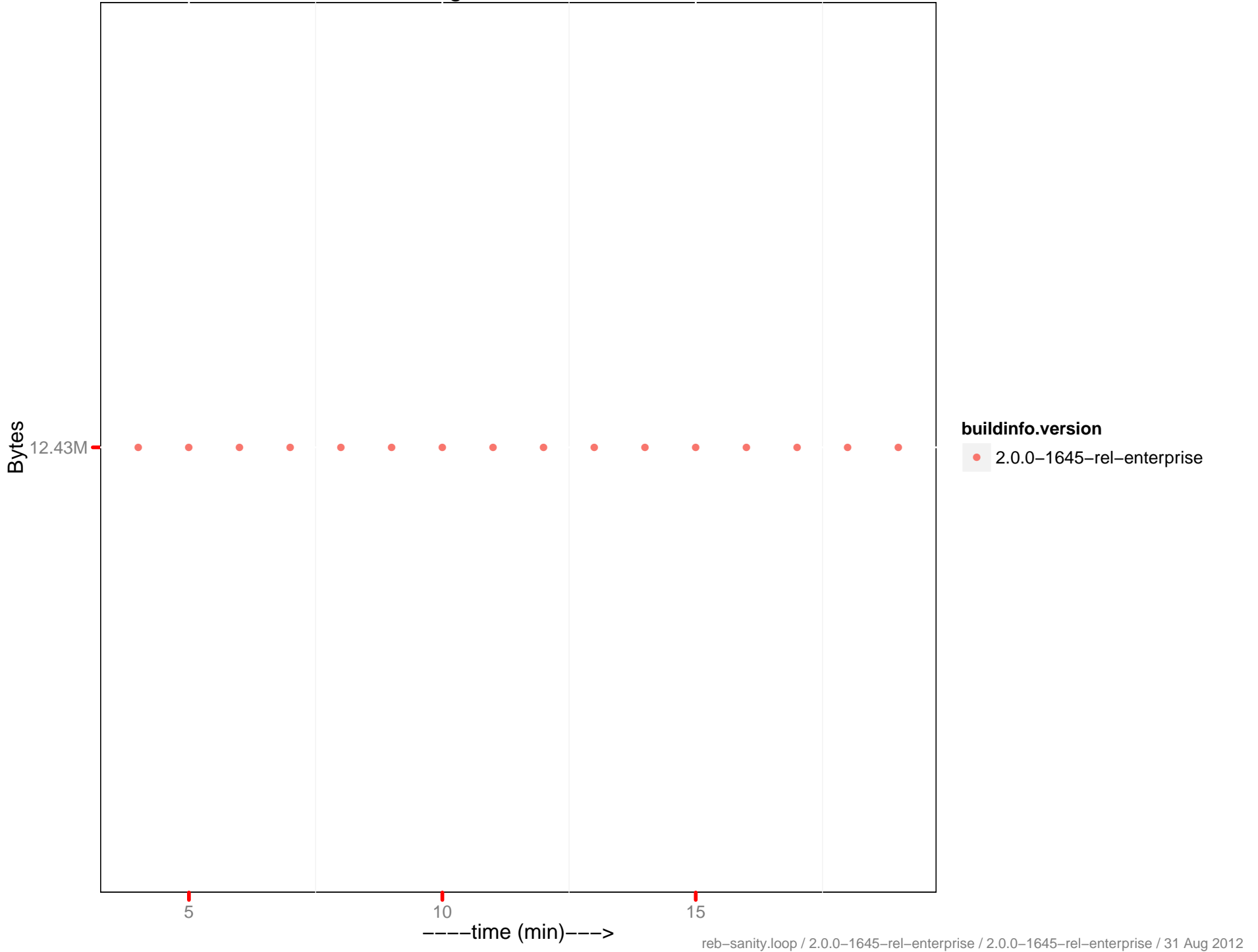
SWAP Usage – 192.168.0.20:8091



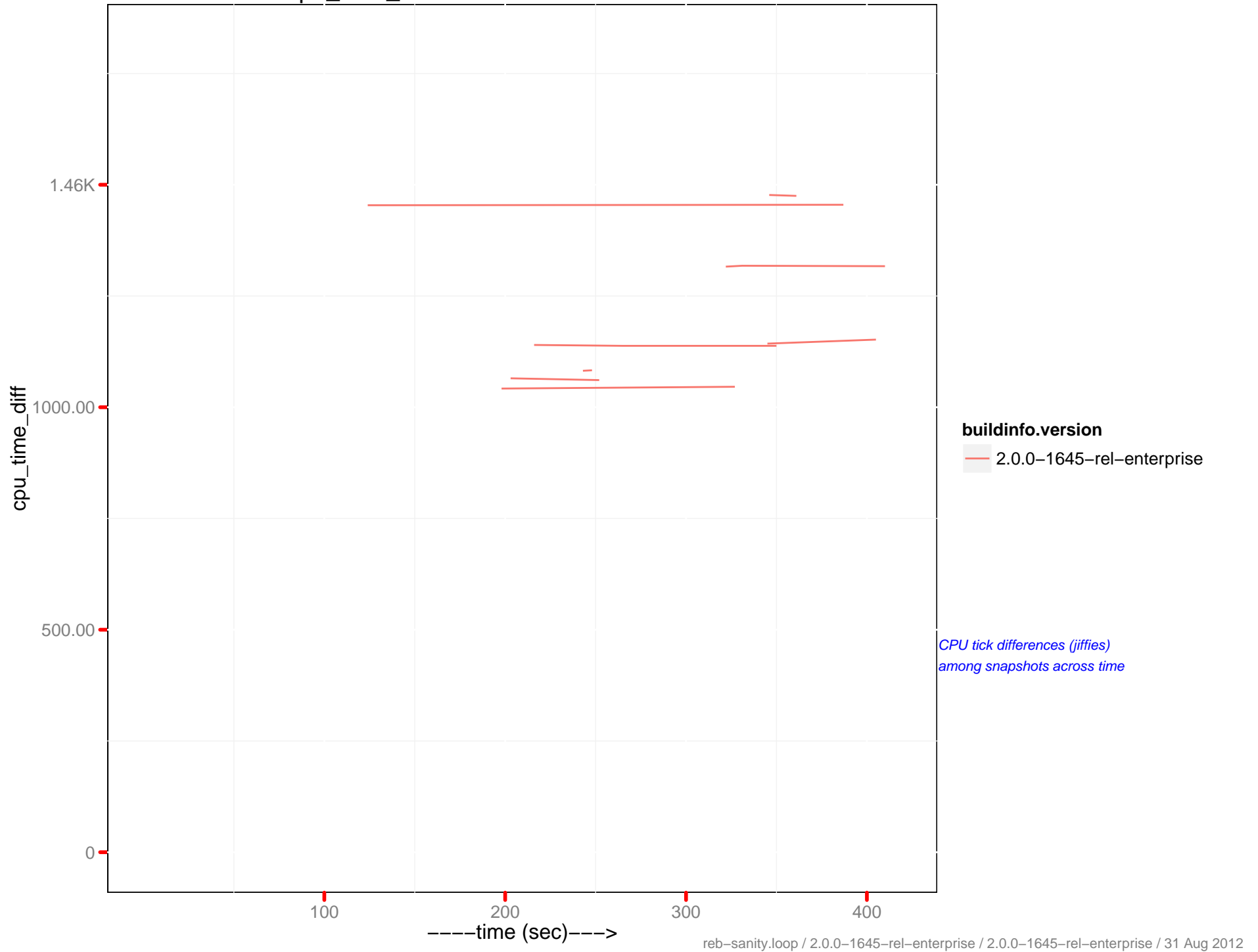
SWAP Usage – 192.168.0.22:8091



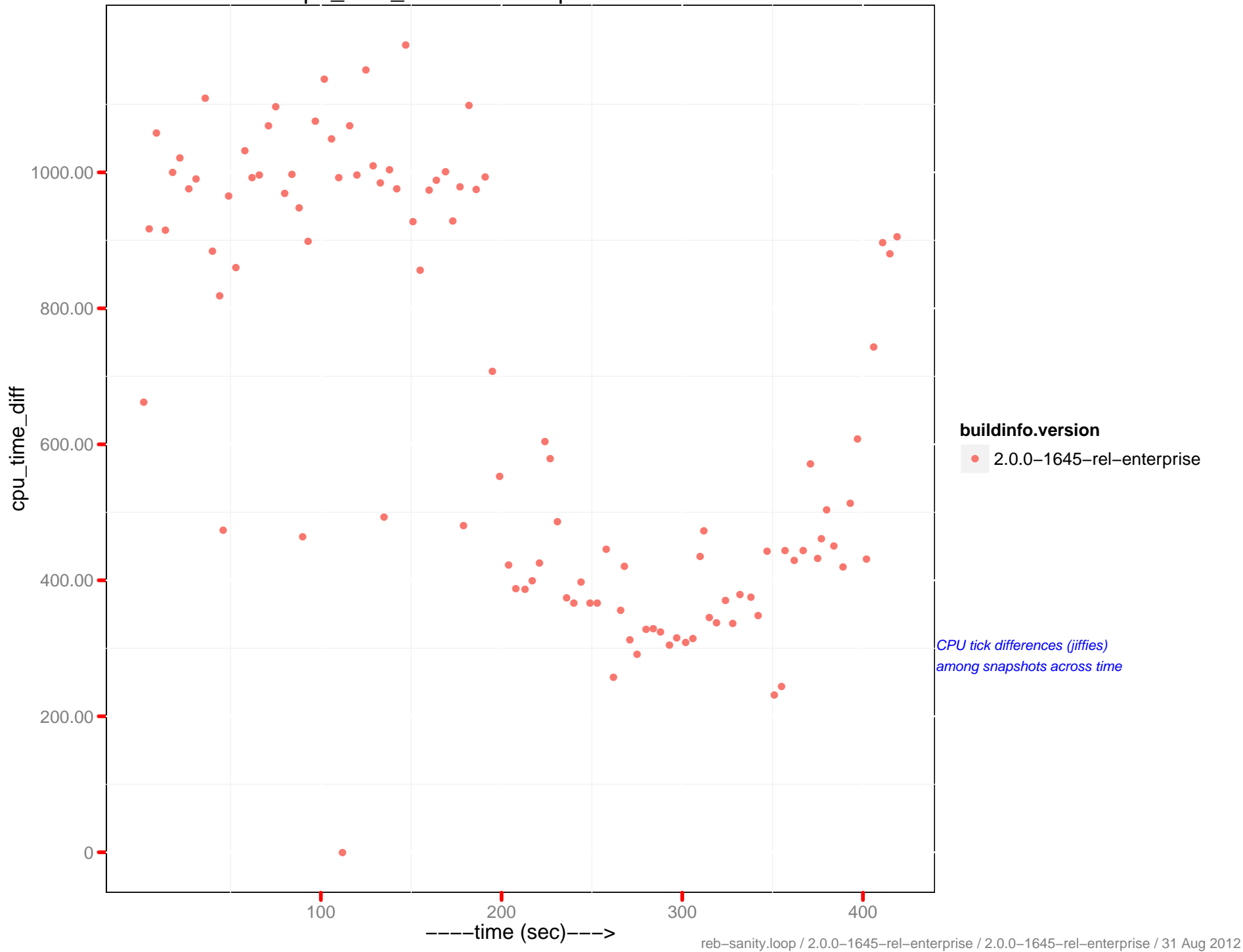
SWAP Usage – 192.168.0.23:8091



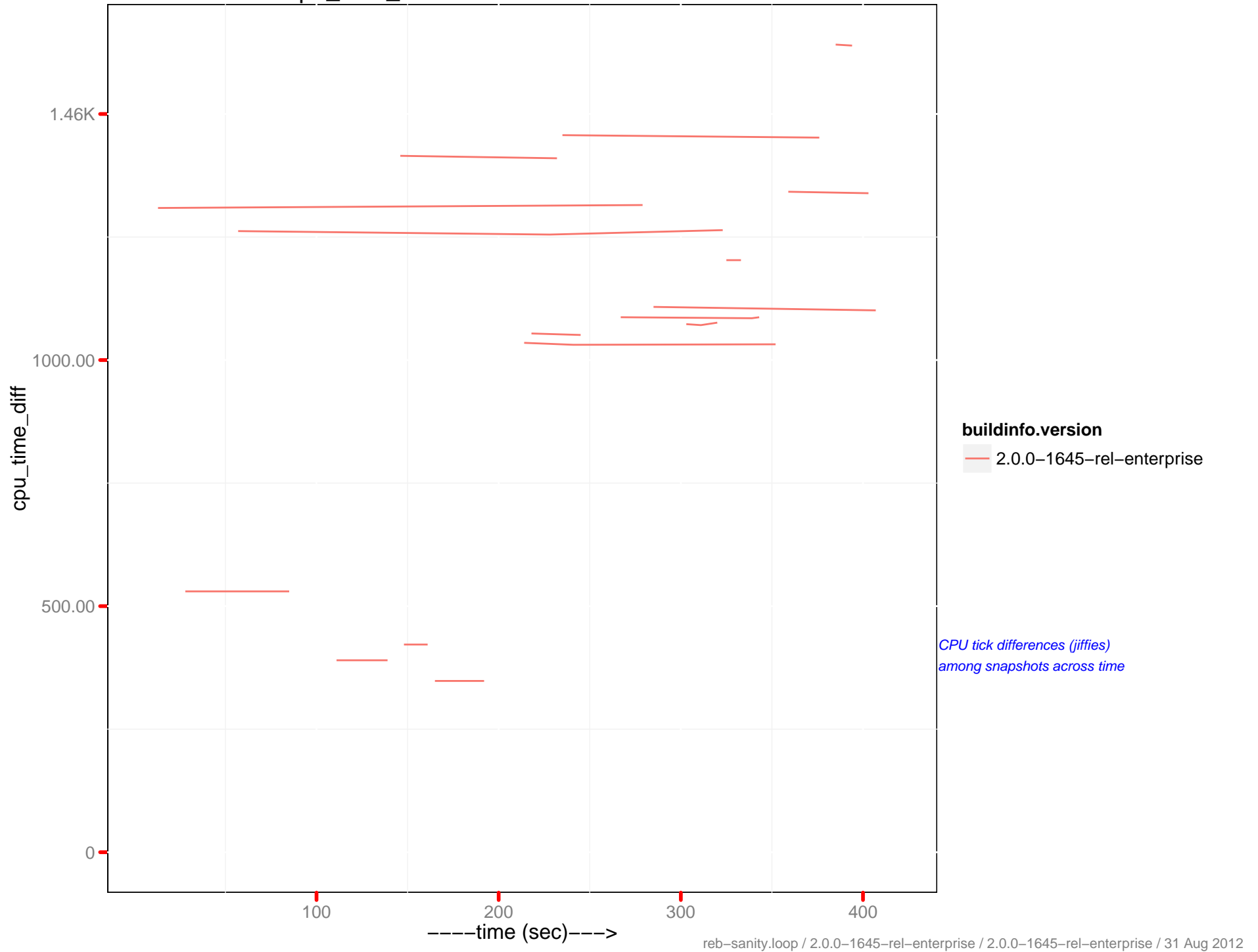
cpu_time_diff: memcached – 192.168.0.20



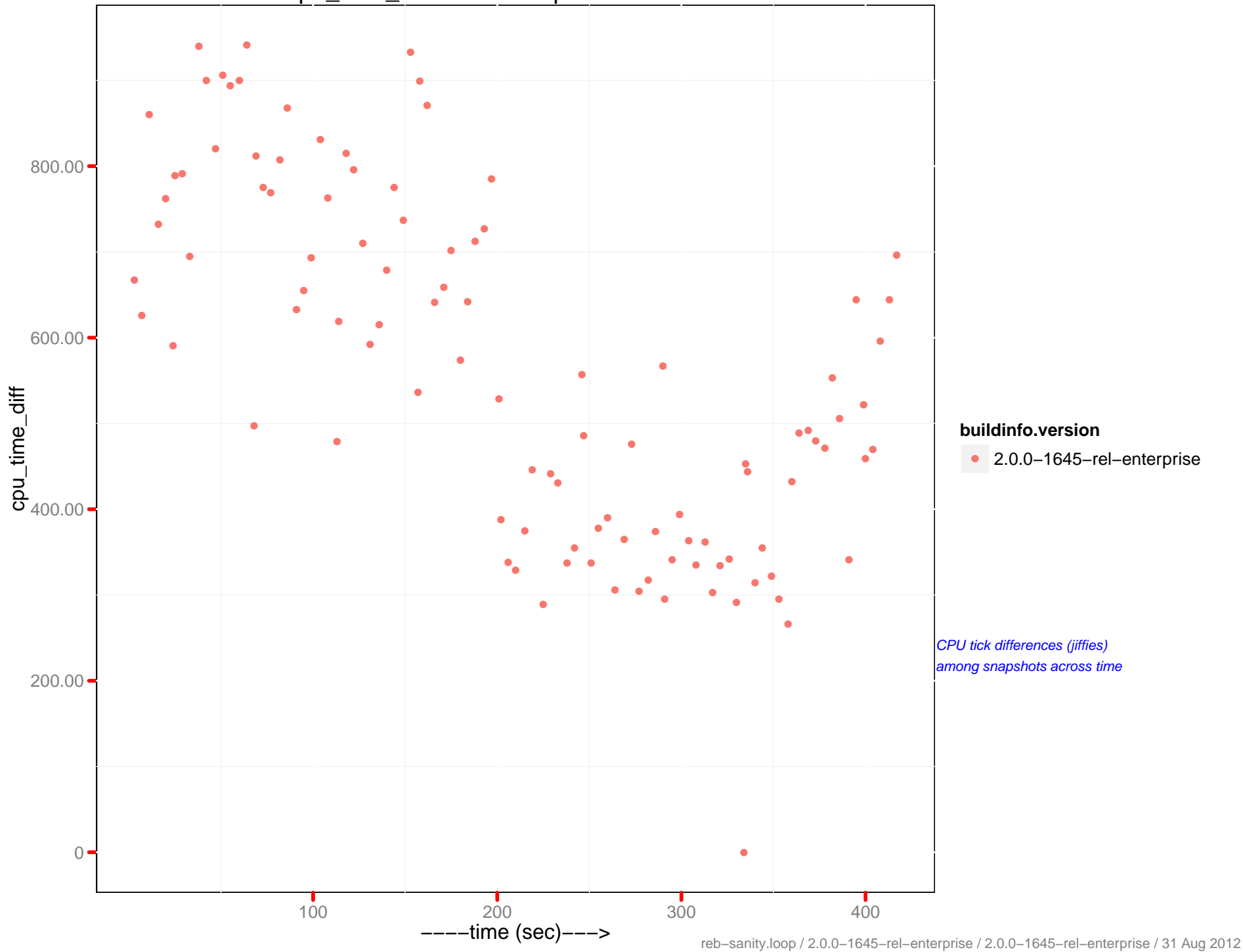
cpu_time_diff : beam.smp - 192.168.0.20



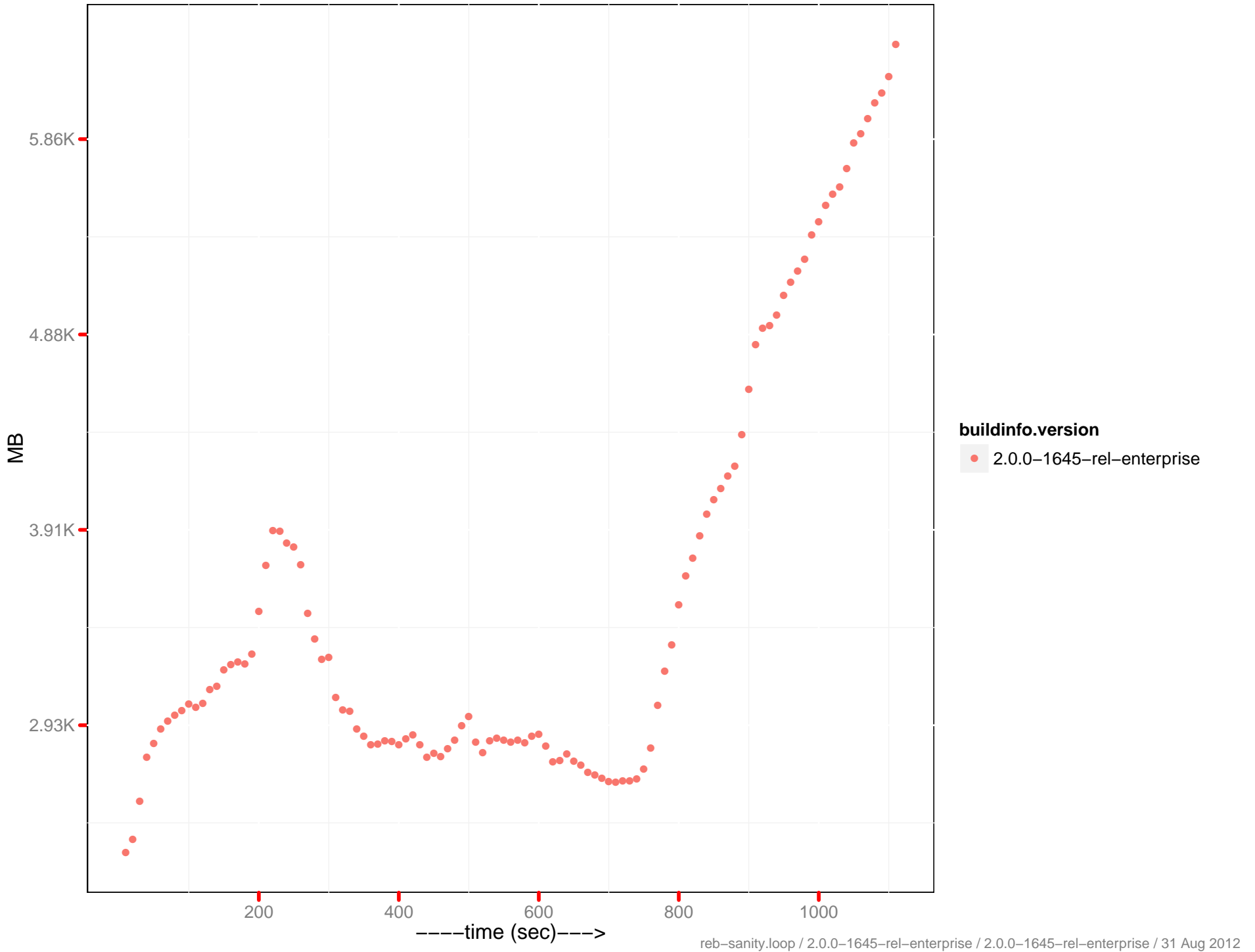
cpu_time_diff: memcached – 192.168.0.22



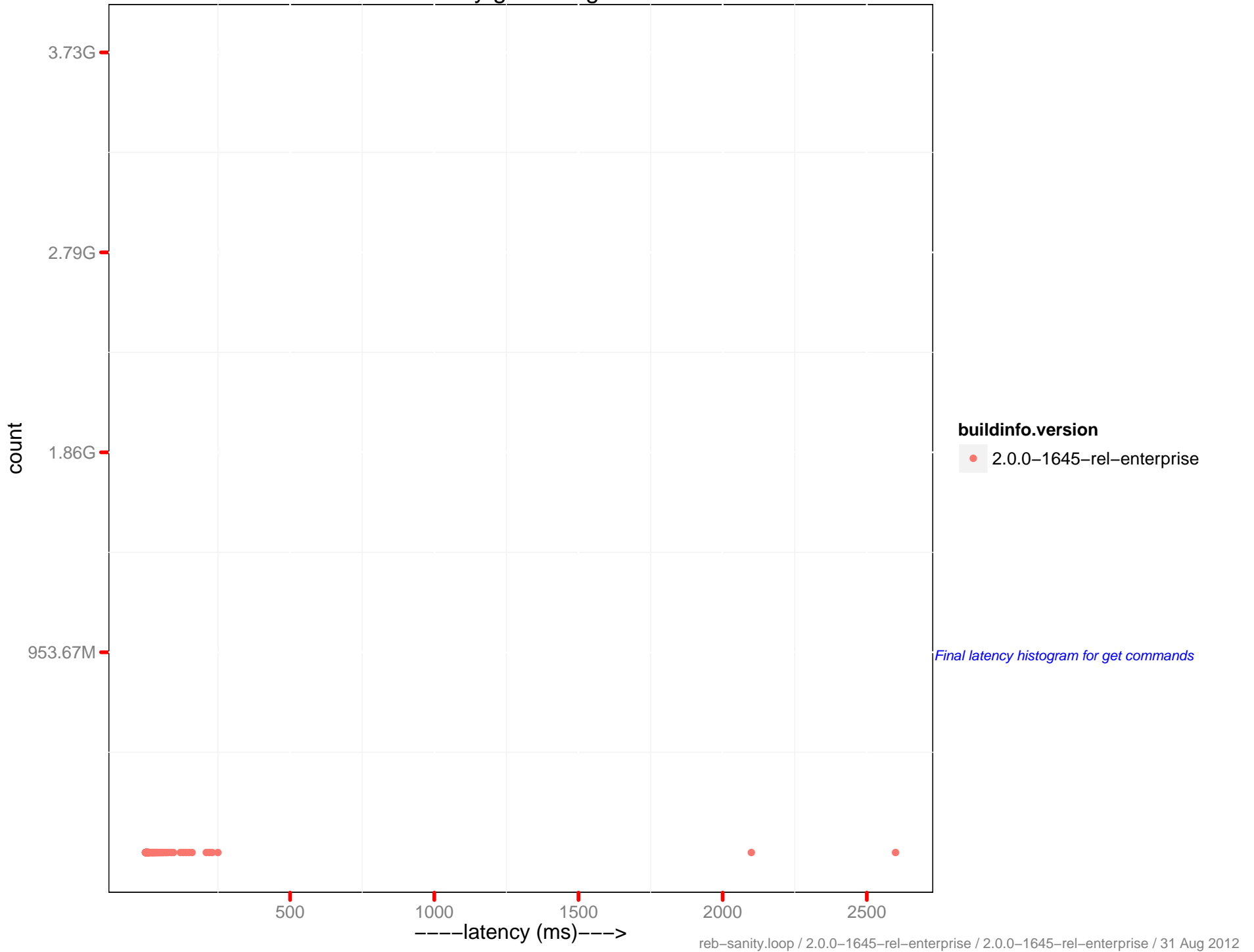
cpu_time_diff : beam.smp - 192.168.0.22



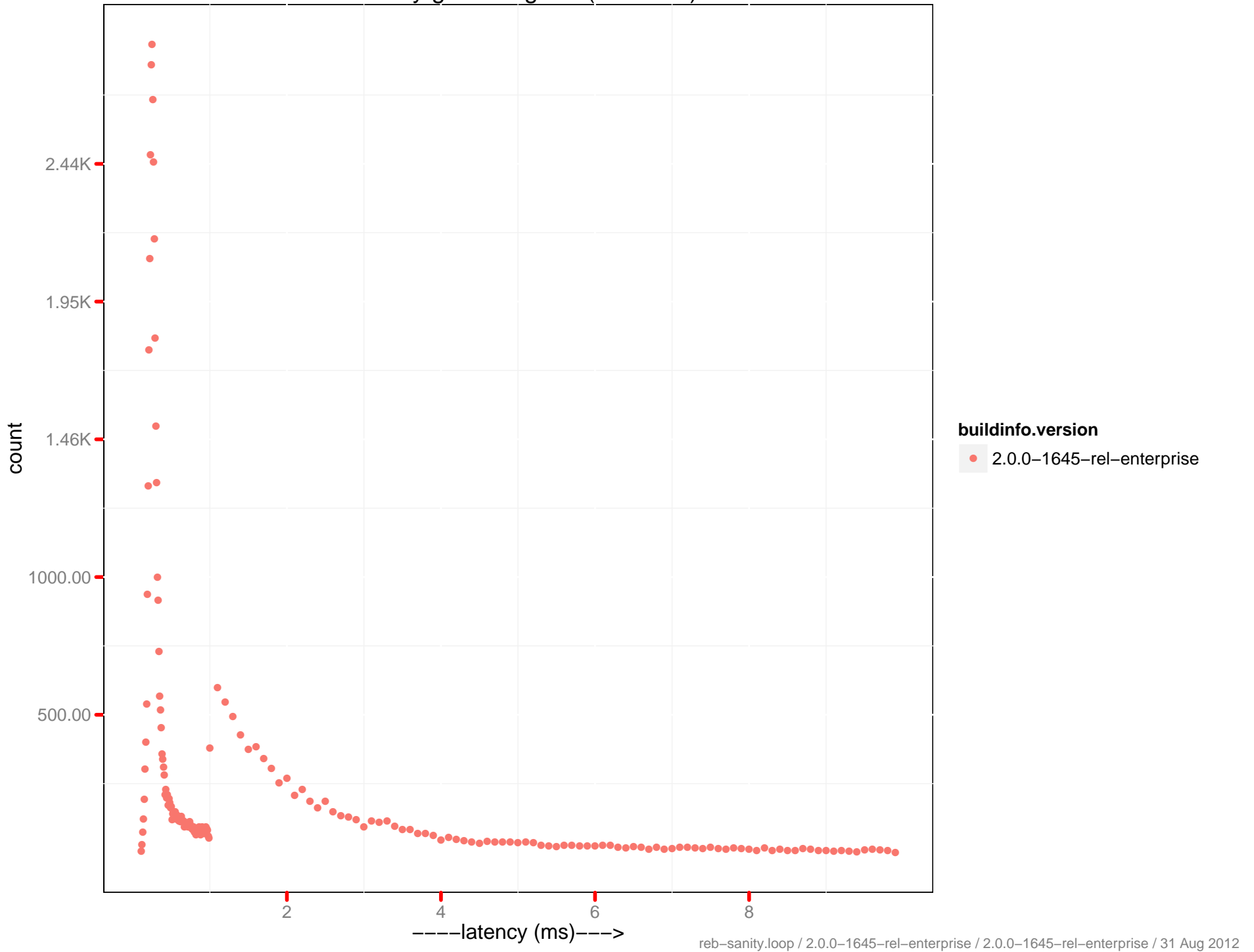
Data disk size



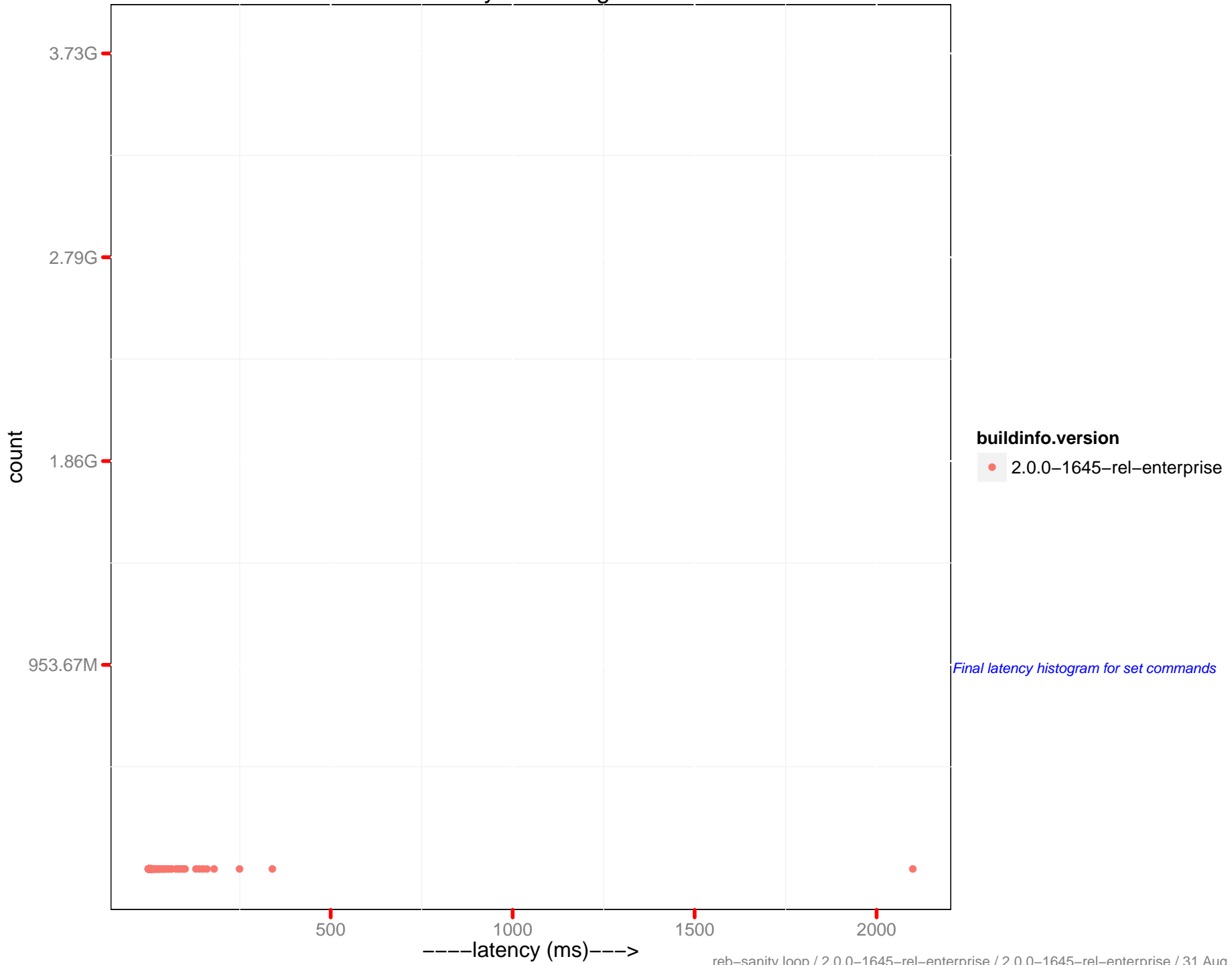
Latency get histogram



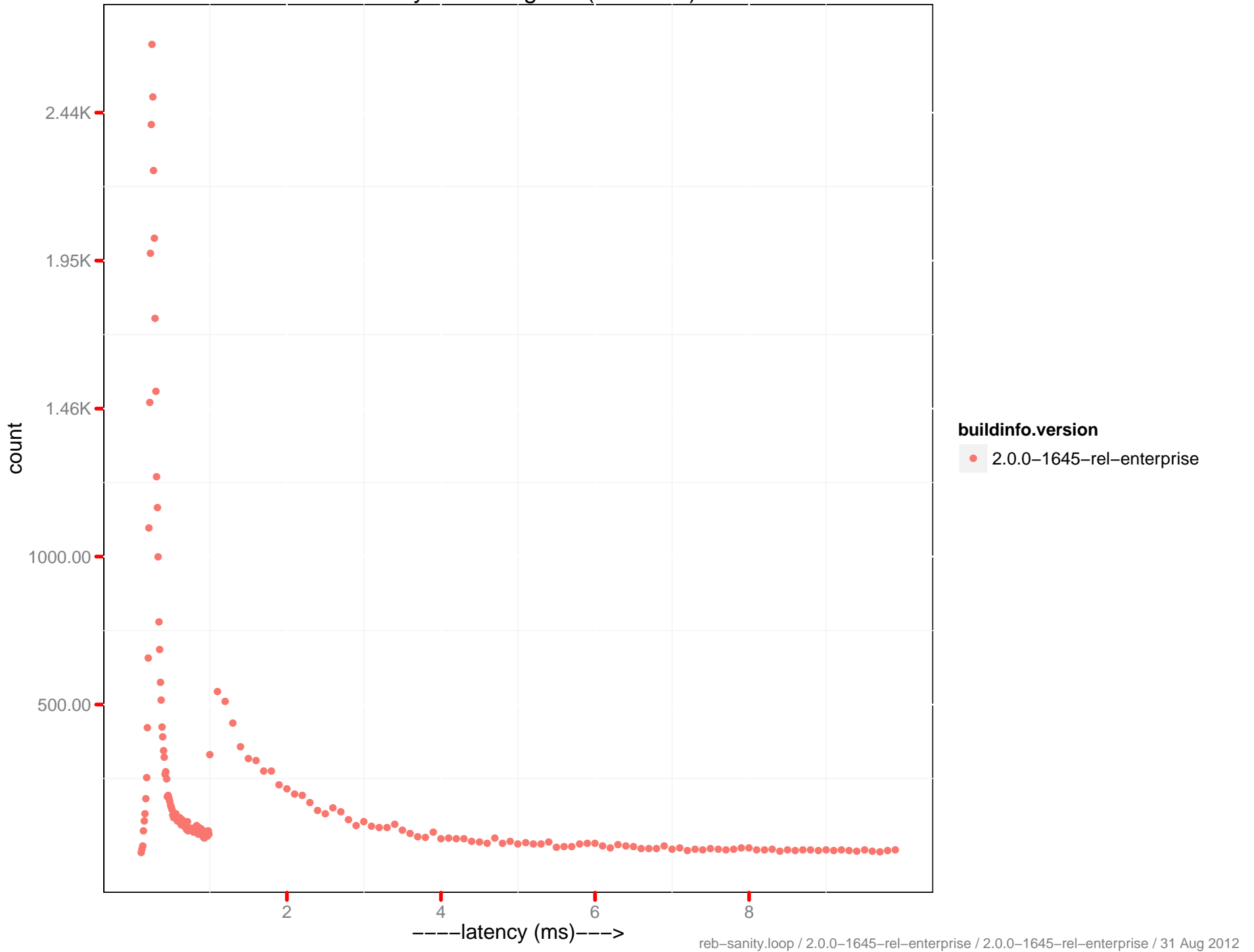
Latency get histogram (0-10 ms)



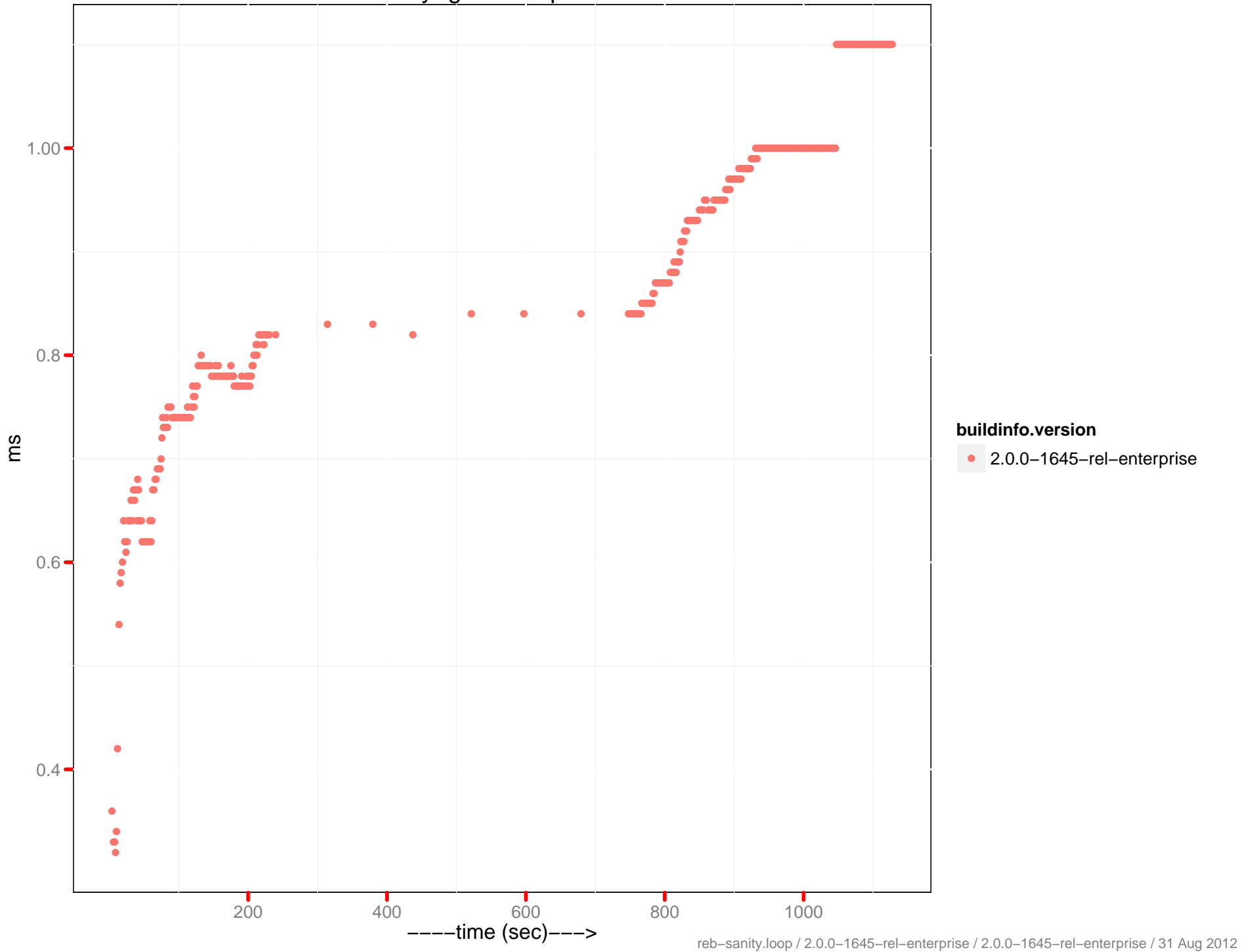
Latency set histogram



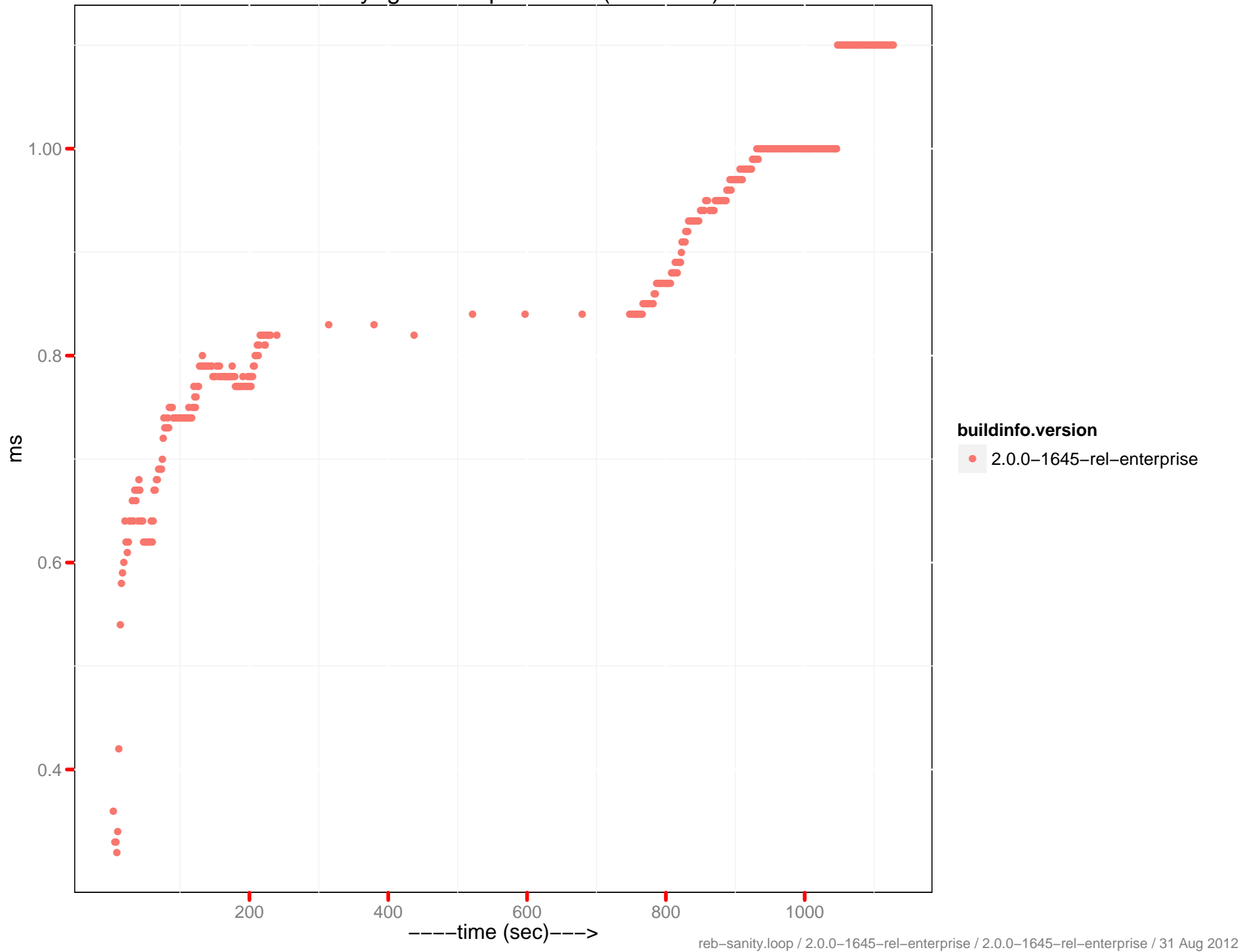
Latency set histogram (0–10 ms)



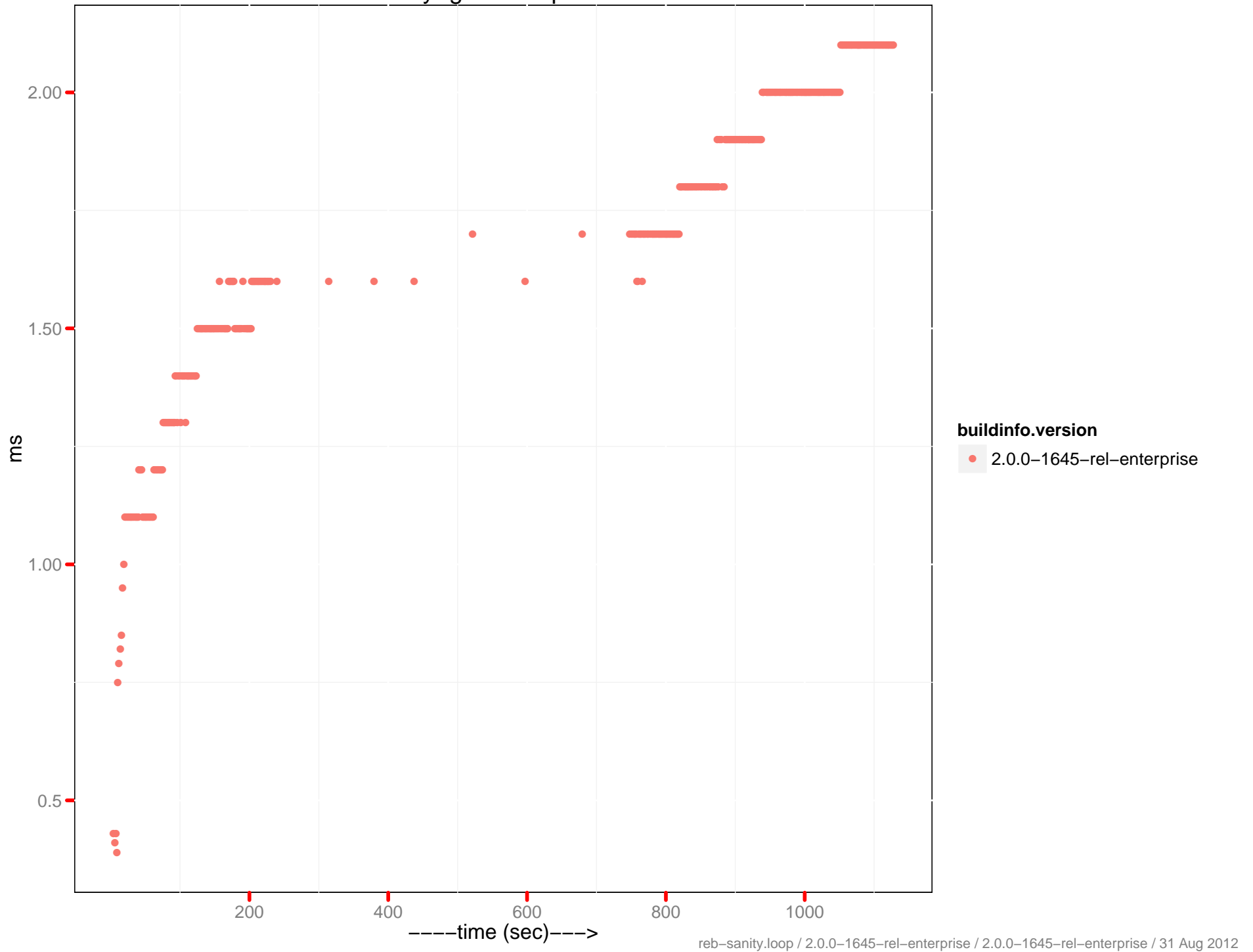
Latency-get 90th percentile



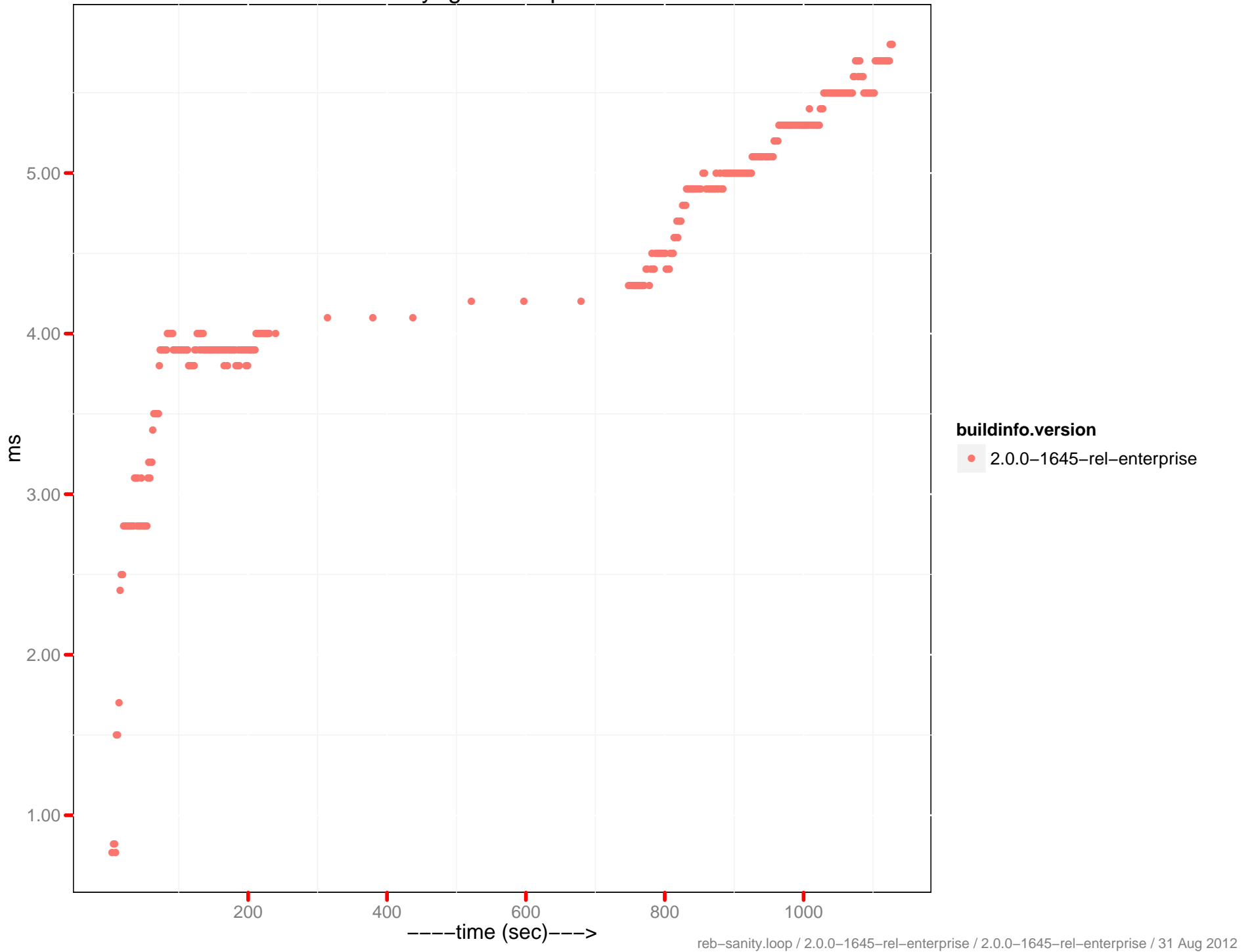
Latency-get 90th percentile (0 - 10ms)



Latency-get 95th percentile

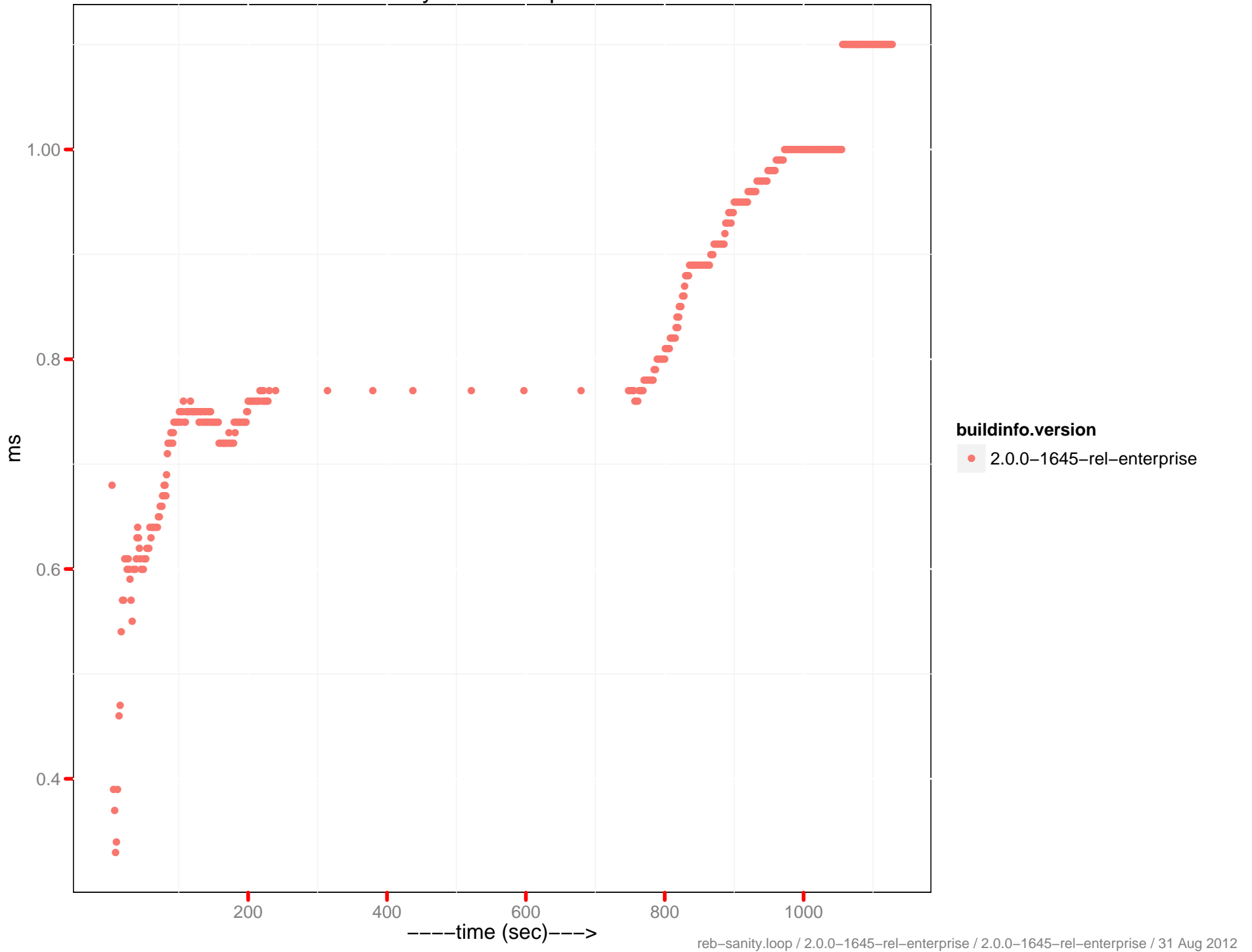


Latency-get 99th percentile

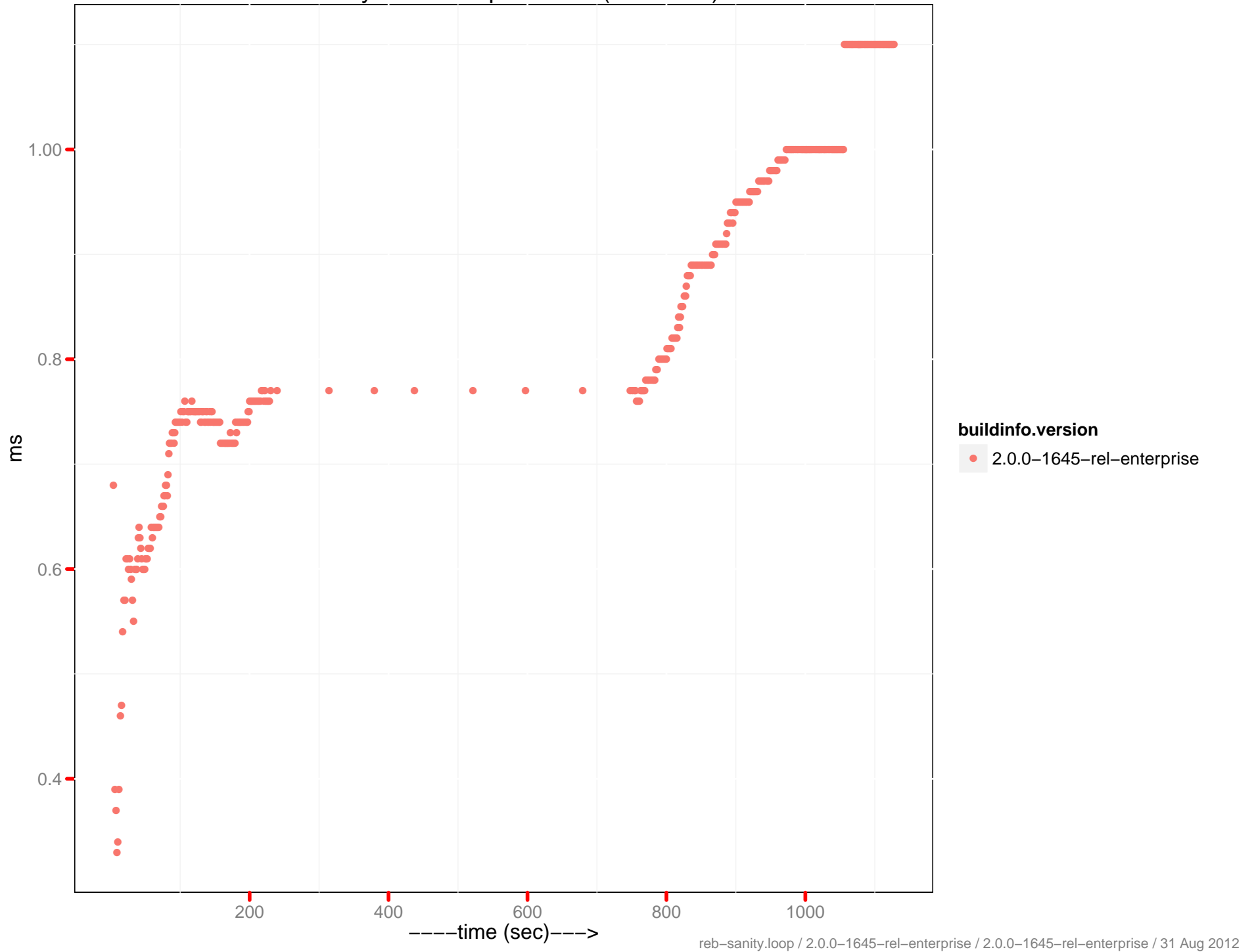


buildinfo.version
● 2.0.0-1645-rel-enterprise

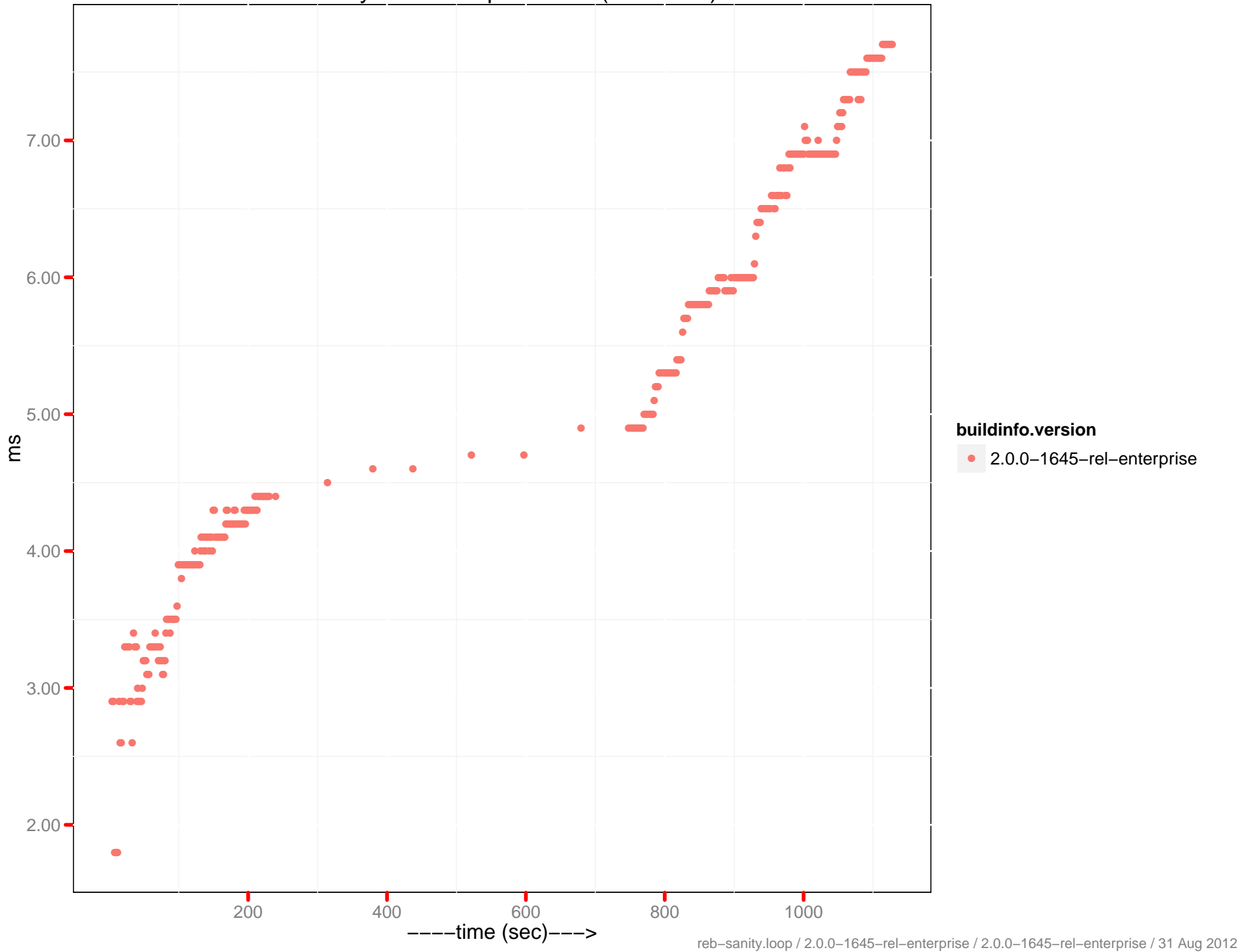
Latency-set 90th percentile



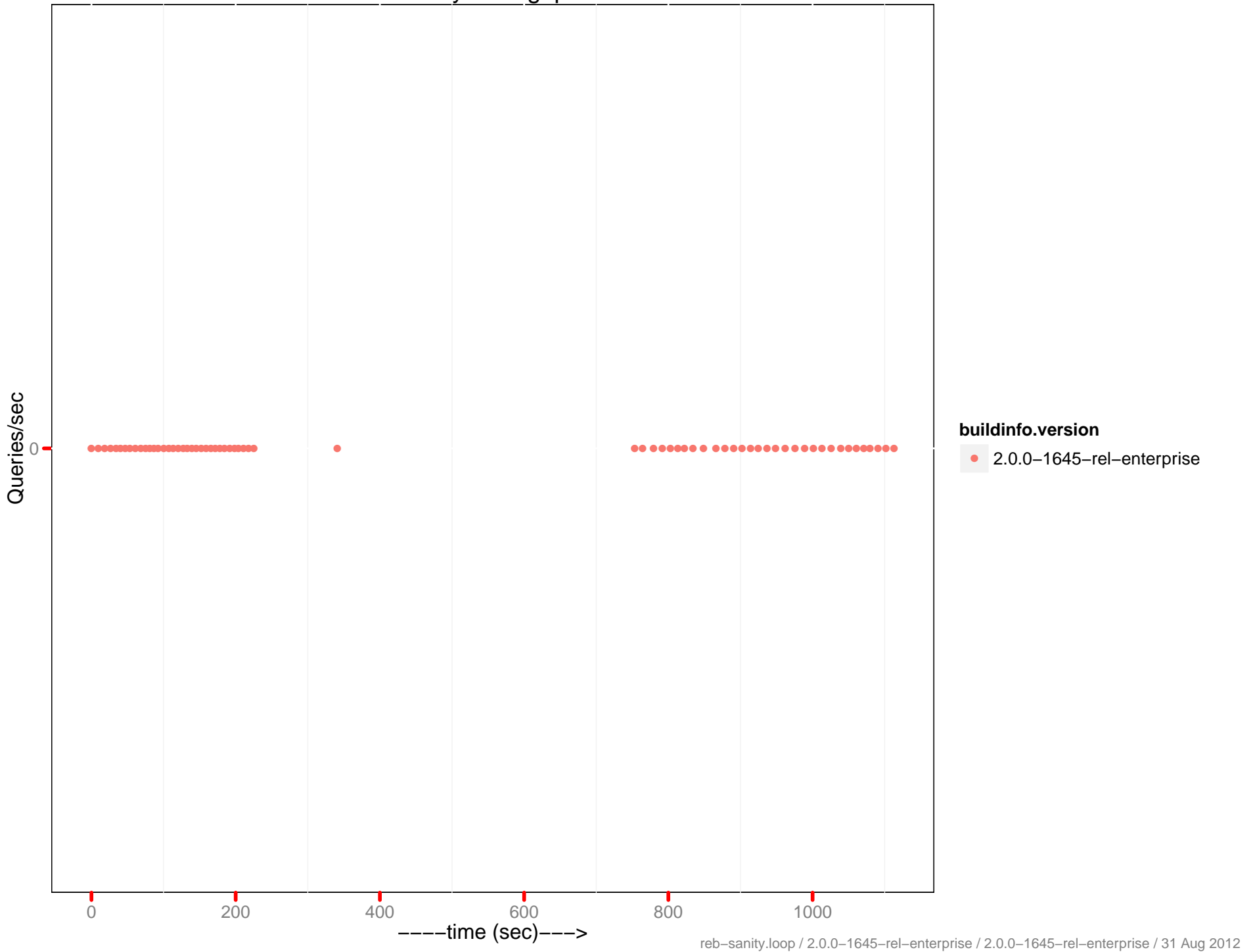
Latency-set 90th percentile (0 - 10ms)



Latency-set 99th percentile (0 - 10ms)



Query throughput



```
reb-sanity.conf
# mixed 0.5M load, 0.1M hot reload, 0.5M access creates, no draining
# rebalance after creating 0.2M items
#
# system memory: 7G per node

performance.eperf.EPerfClient.test_eperf_rebalance

params:

# general
batch=50
kind=nonjson
mem_quota=7000
spec=reb-sanity

# load phase
hot_init_items=100000
items=500000

# access phase
# Read:Insert:Update:Delete Ratio = 50:4:40:6.
ratio_sets=0.5
ratio_misses=0.05
ratio_creates=0.08
ratio_deletes=0.13
ratio_hot=0.05
ratio_hot_gets=0.099
ratio_hot_sets=0.099
ratio_expirations=0.03
max_creates=500000

# rebalance
rebalance_after=200000
num_nodes_after=2

# control (defaults: pytests/performance/perf_defaults.py)
load_wait_until_drained=1
loop_wait_until_drained=0
mcsoda_heartbeat=3
tear_down=1
tear_down_proxy=1
tear_down_bucket=0
tear_down_cluster=1
tear_down_on_setup=0
```

tahoe-dedicated.ini

[global]

username:root

password:couchbase

port:8091

data_path:/data

[servers]

1:192.168.0.20

2:192.168.0.21

3:192.168.0.22

4:192.168.0.23

[clients]

1:192.168.0.24

2:192.168.0.25

3:192.168.0.26

4:192.168.0.27

5:192.168.0.28

6:192.168.0.29

[membase]

rest_username:Administrator

rest_password:password

[dashboard]

1:dashboard.hq.couchbase.com:80