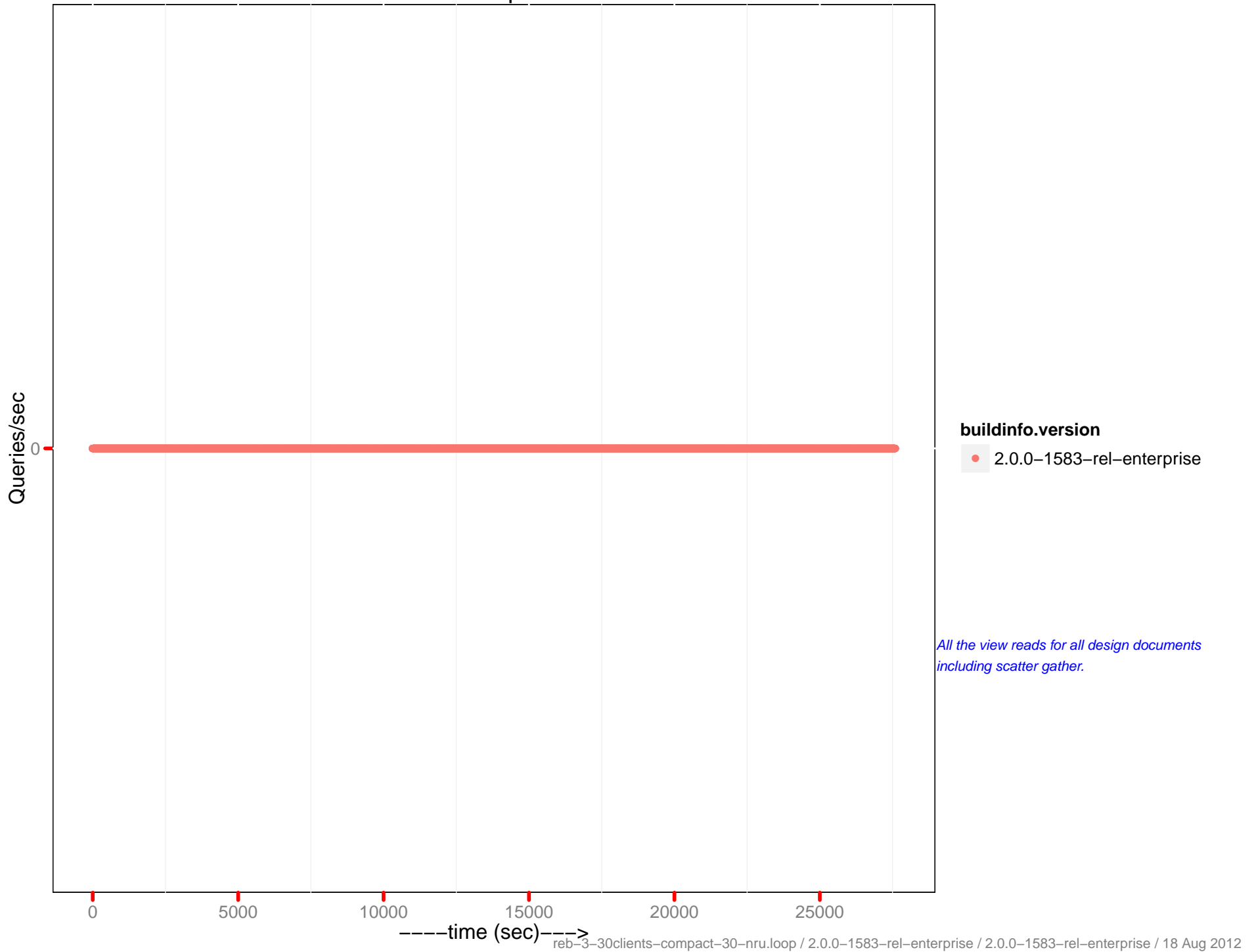
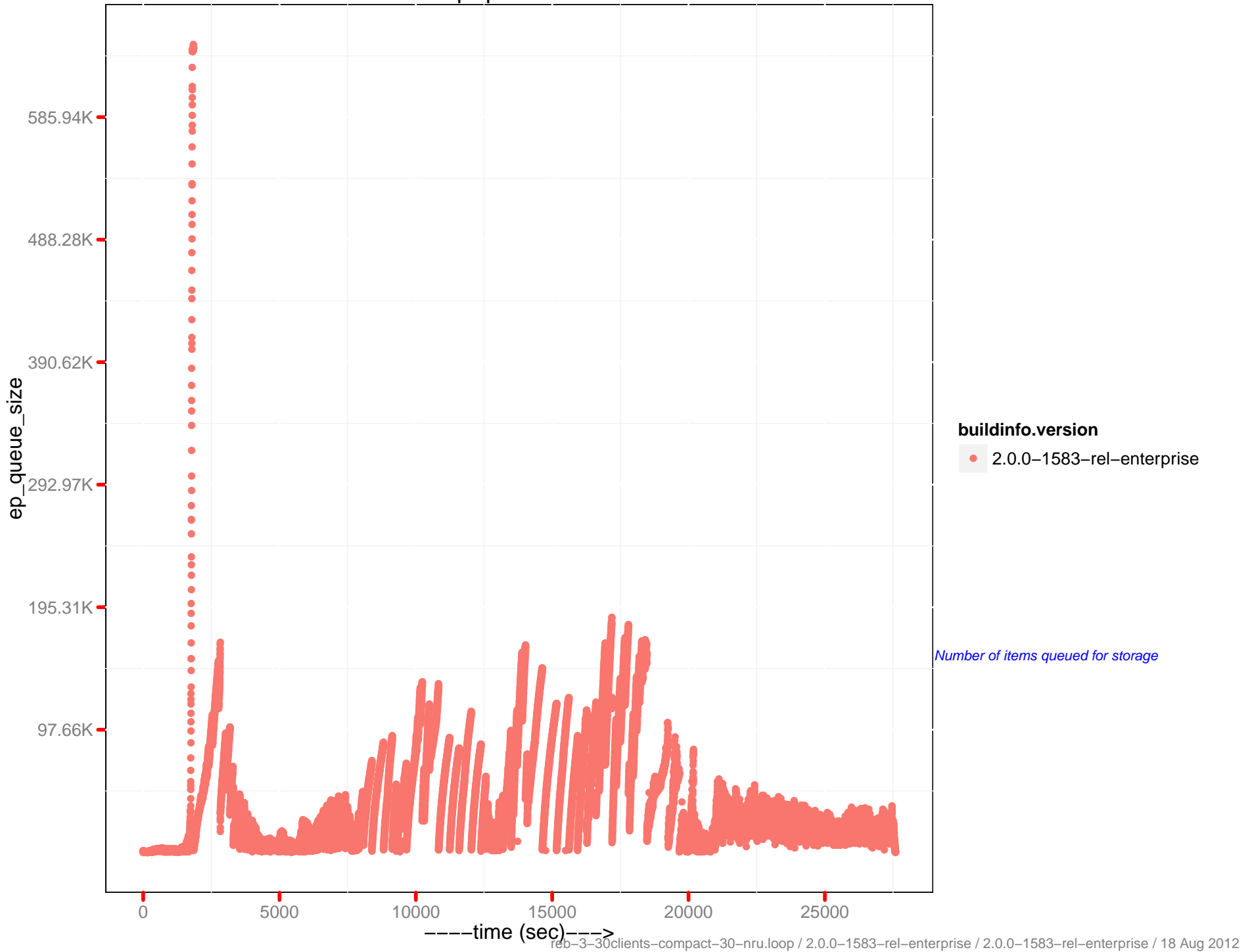


	2.0.0 – 1583	2.0.0 – 1583
<i>Runtime (in hr)</i>	7.68	NA
<i>Avg. Drain Rate</i>	1.99K	NANA
<i>Peak Disk (GB)</i>	61.3	NA
<i>Peak Memory (GB)</i>	16.86	NA
<i>Avg. OPS</i>	8.97K	NANA
<i>Avg. mem memcached (GB)</i>	16.42	NA
<i>Avg. mem beam.smp (MB)</i>	368.84	NA
<i>Latency-get (90th) (ms)</i>	1.64	NA
<i>Latency-get (95th) (ms)</i>	3.1	NA
<i>Latency-get (99th) (ms)</i>	14.35	NA
<i>Latency-set (90th) (ms)</i>	1.68	NA
<i>Latency-set (95th) (ms)</i>	3.07	NA
<i>Latency-set (99th) (ms)</i>	6.74	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>	2731.88	NA
<i>Testrunner Version</i>	4e95644	NA

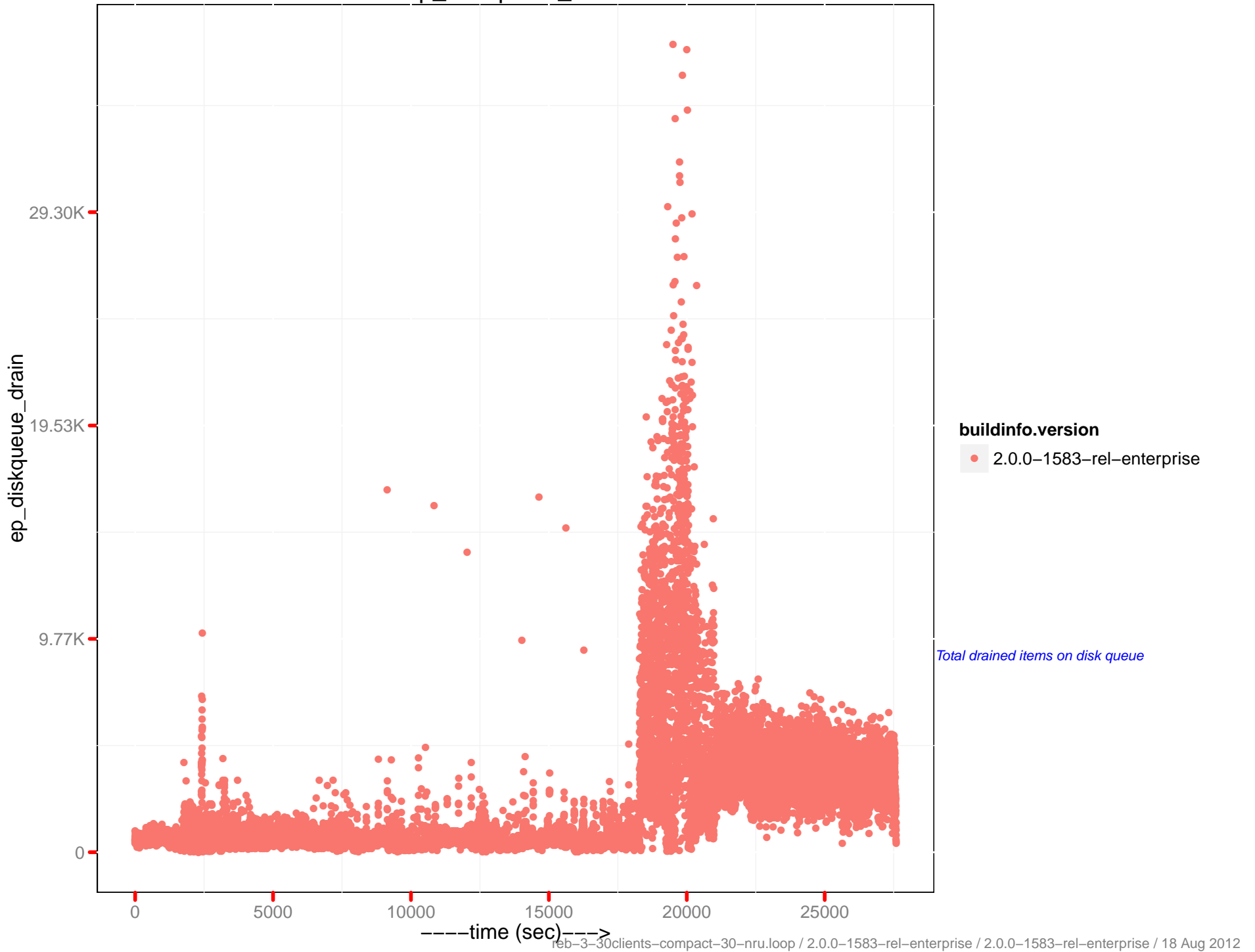
View read per sec.



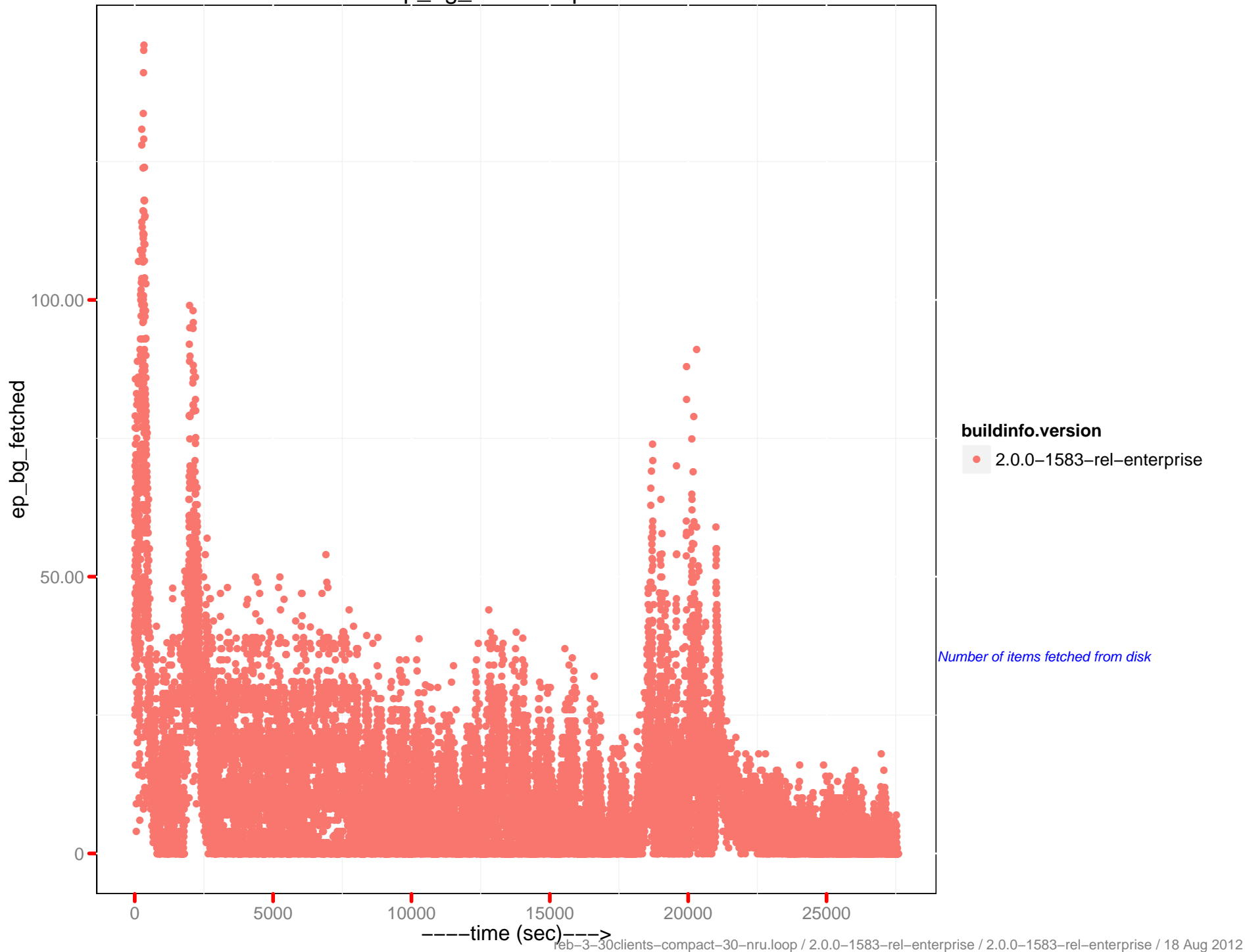
ep queue size



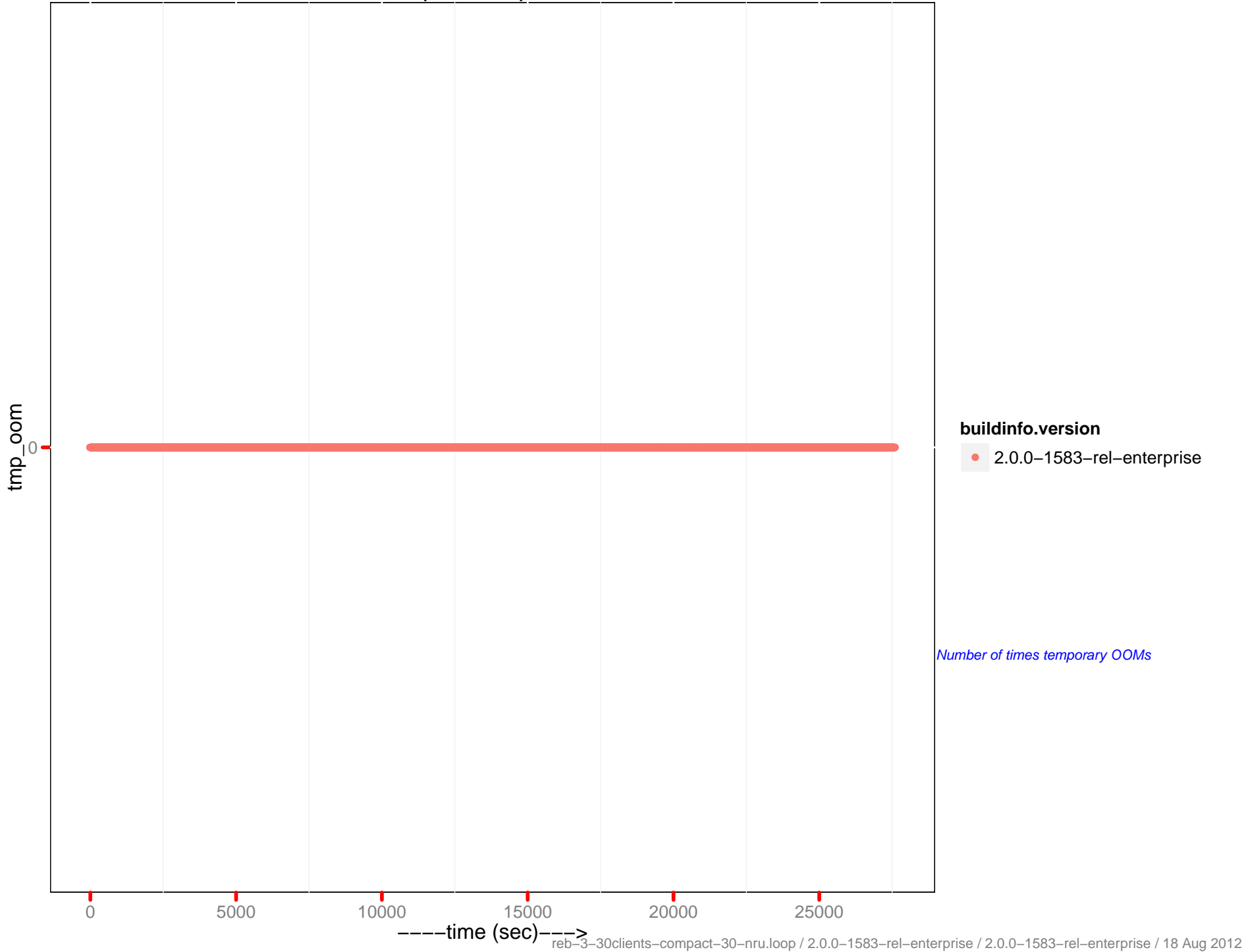
ep_diskqueue_drain



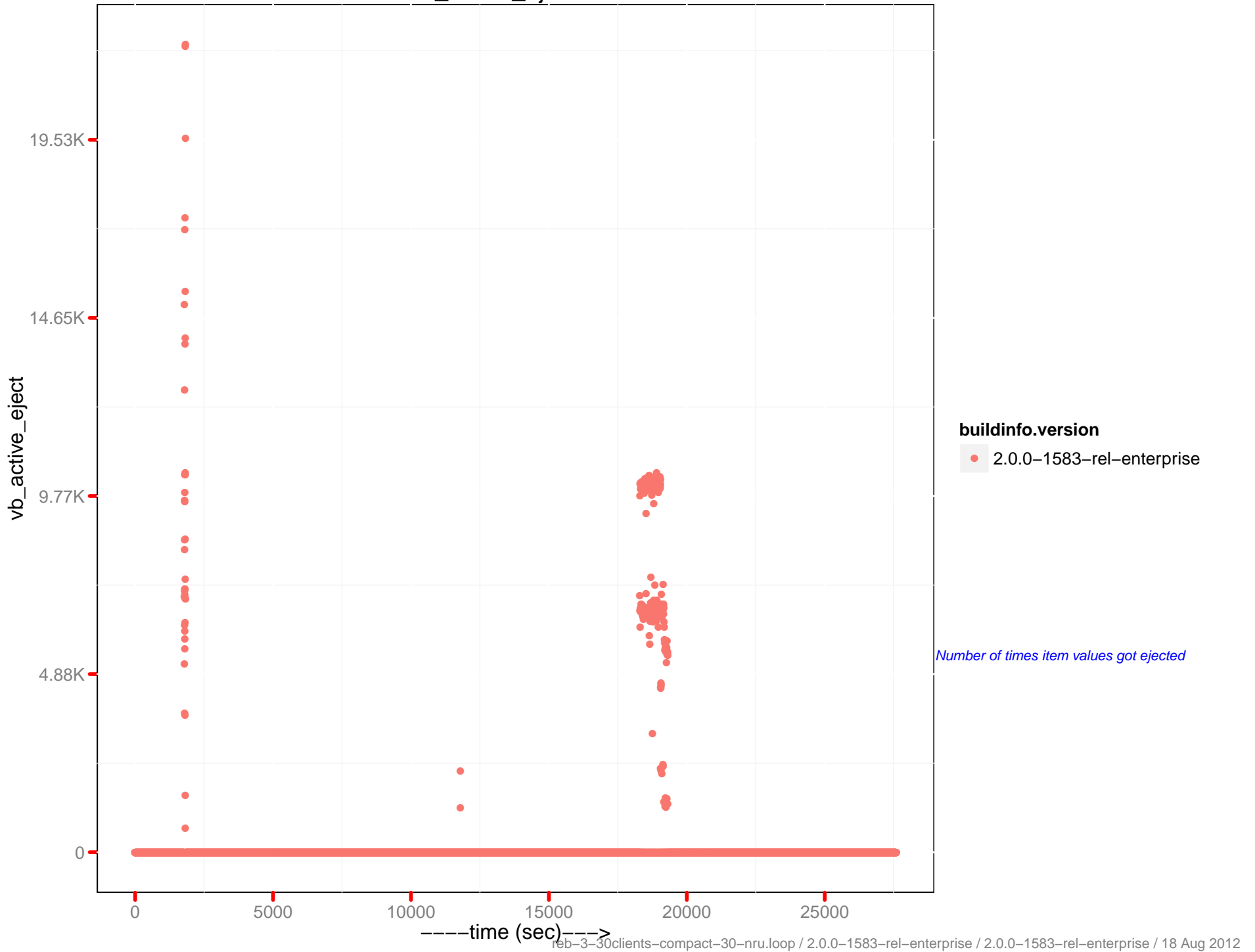
ep_bg_fetched ops/sec



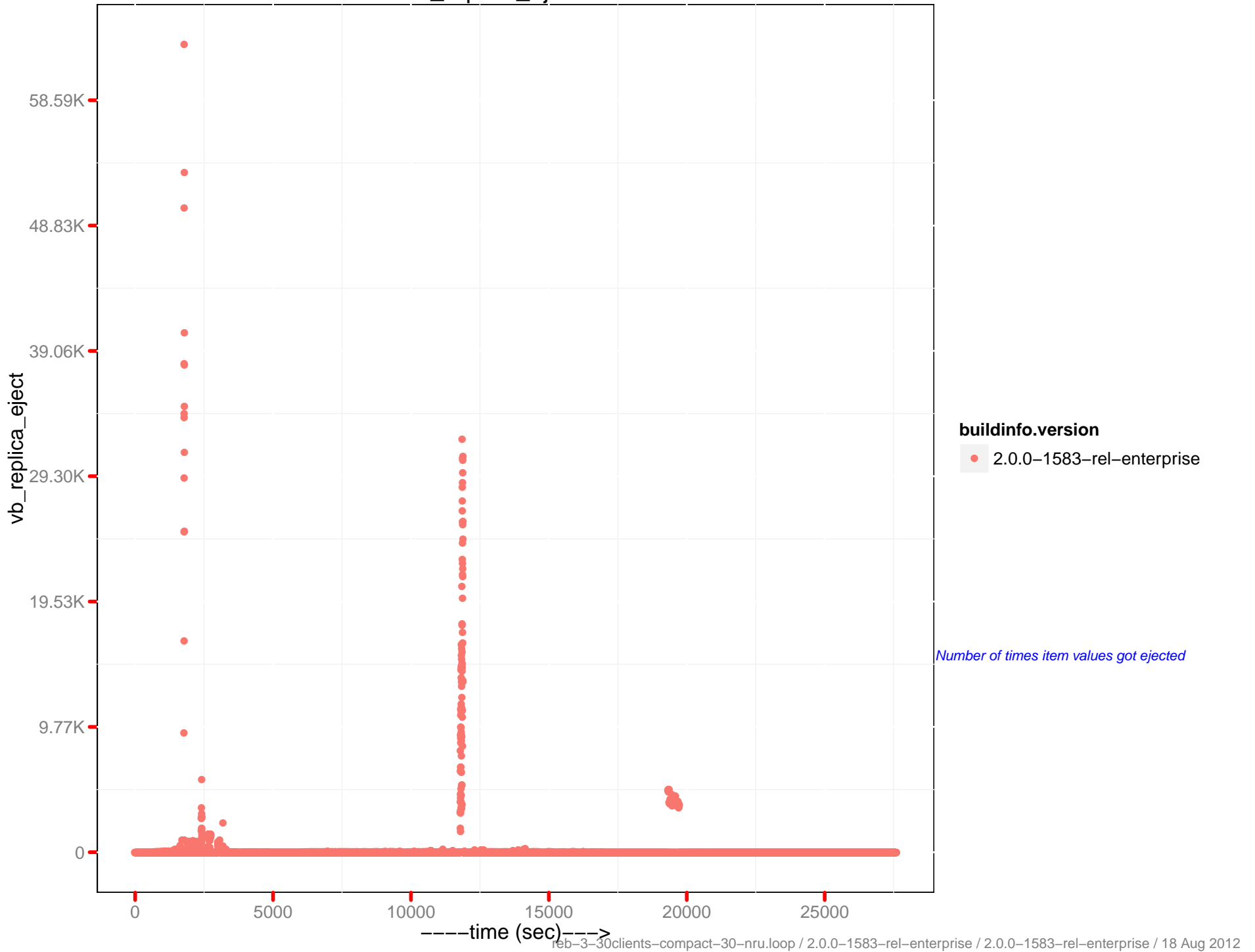
tmp_oom ops/sec



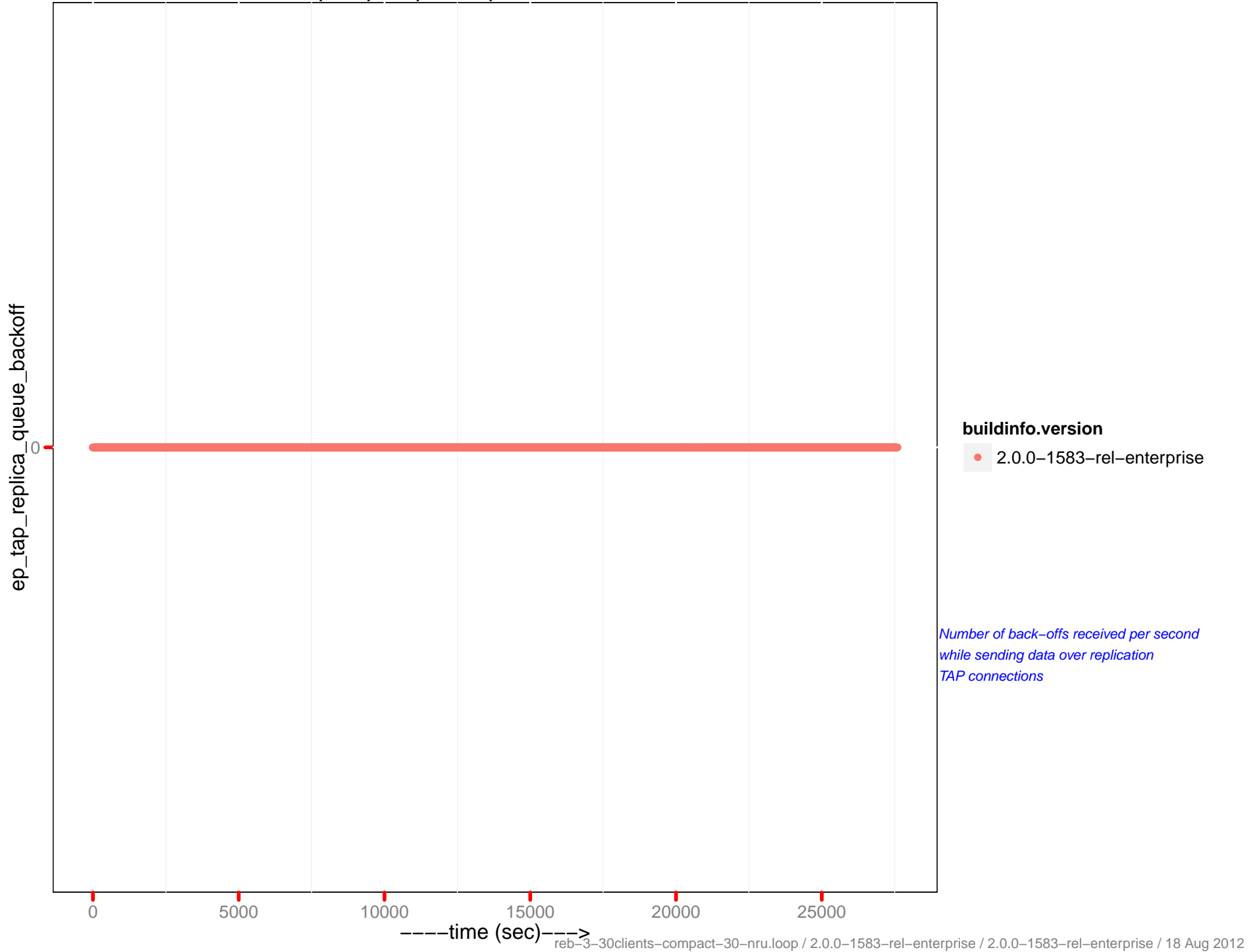
vb_active_eject/sec



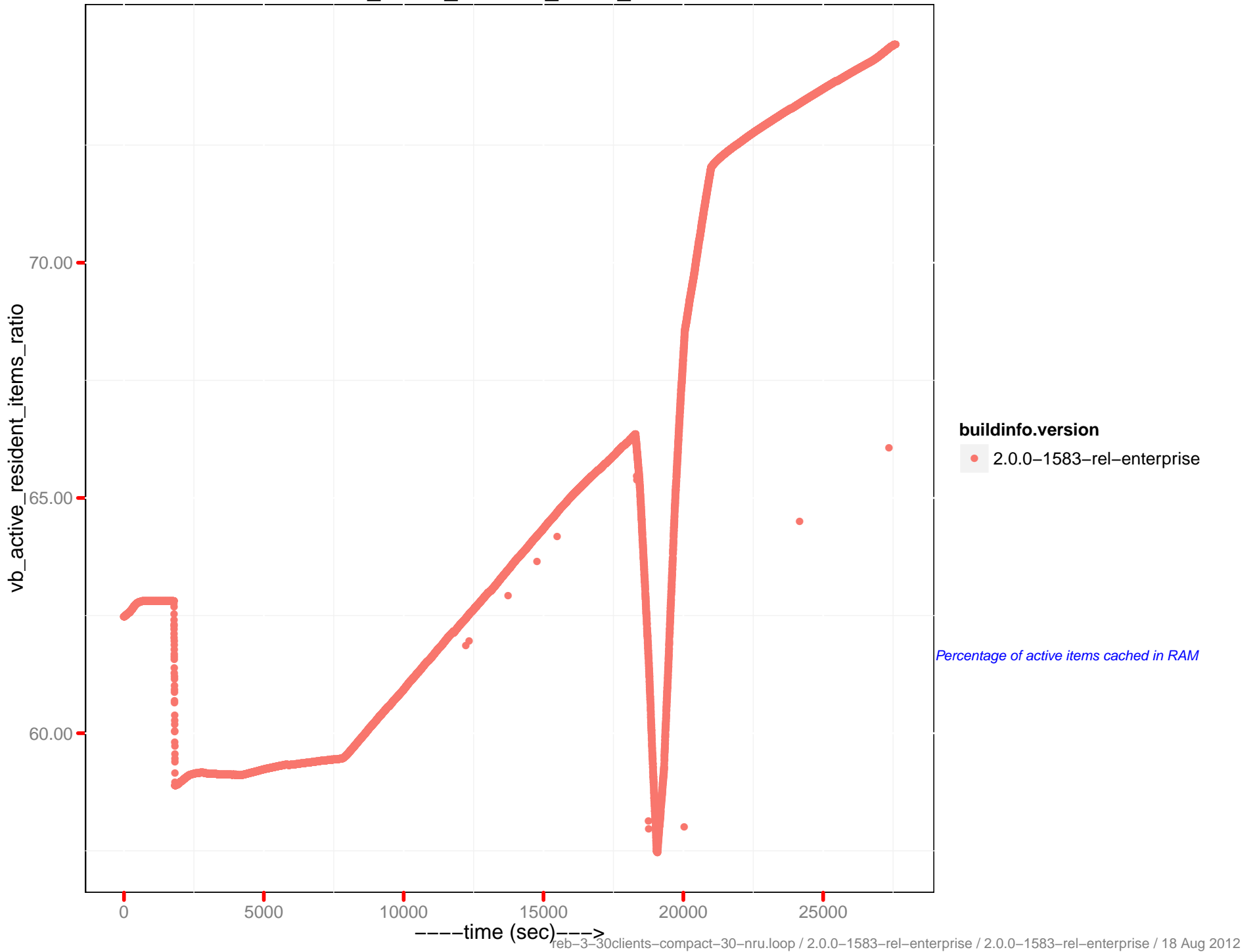
vb_replica_eject/sec



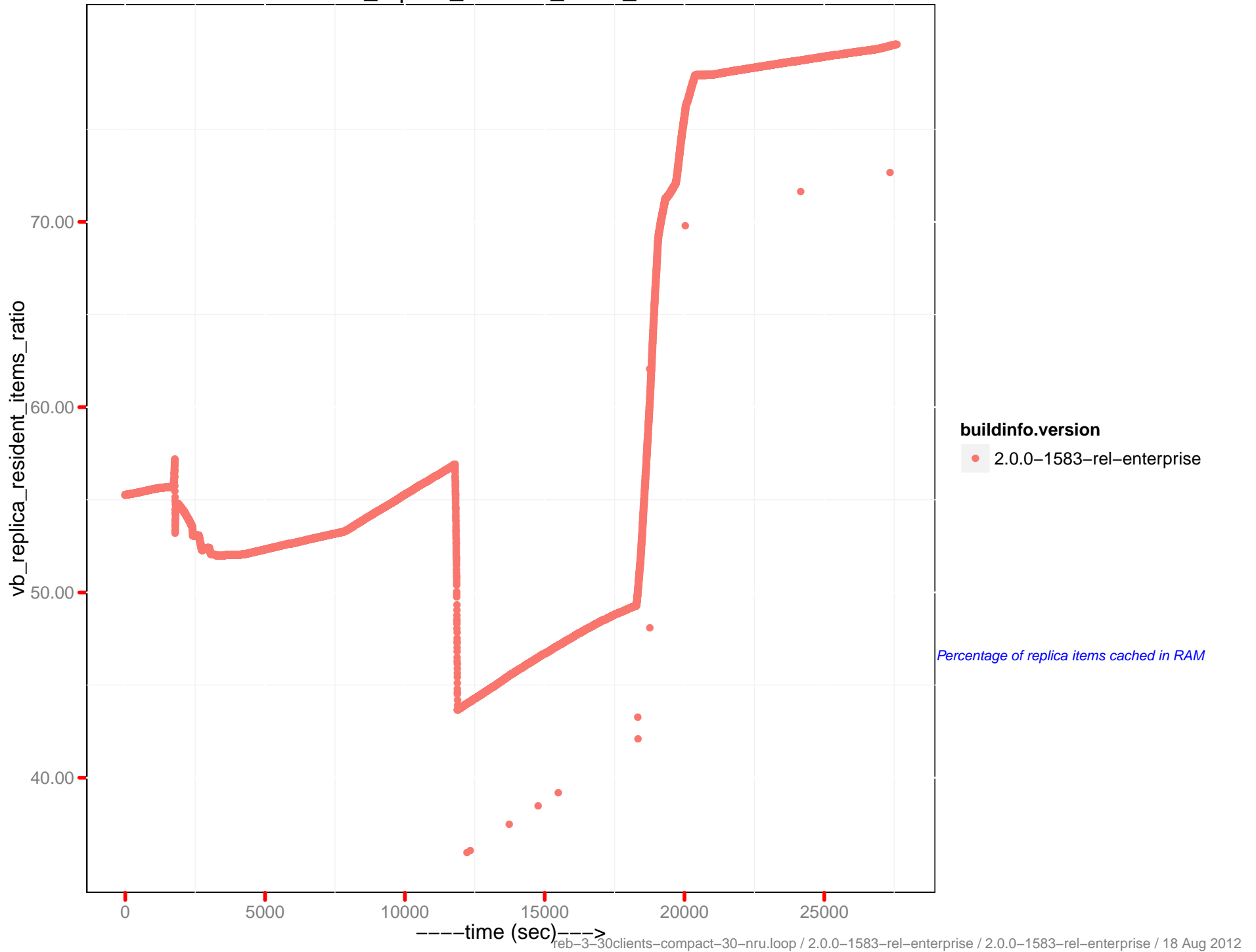
ep_tap_replica_queue_backoff/sec



vb_active_resident_items_ratio



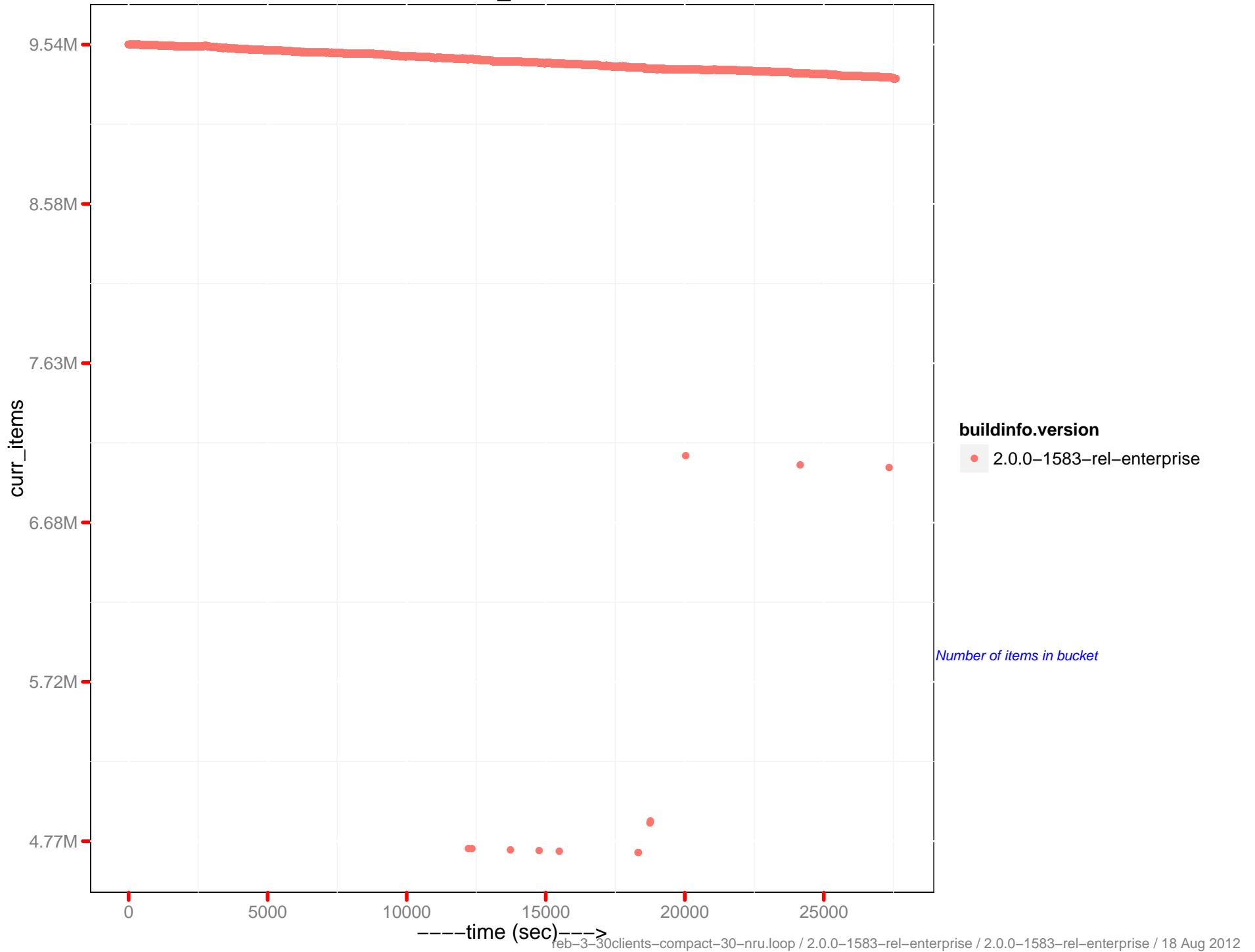
vb_replica_resident_items_ratio



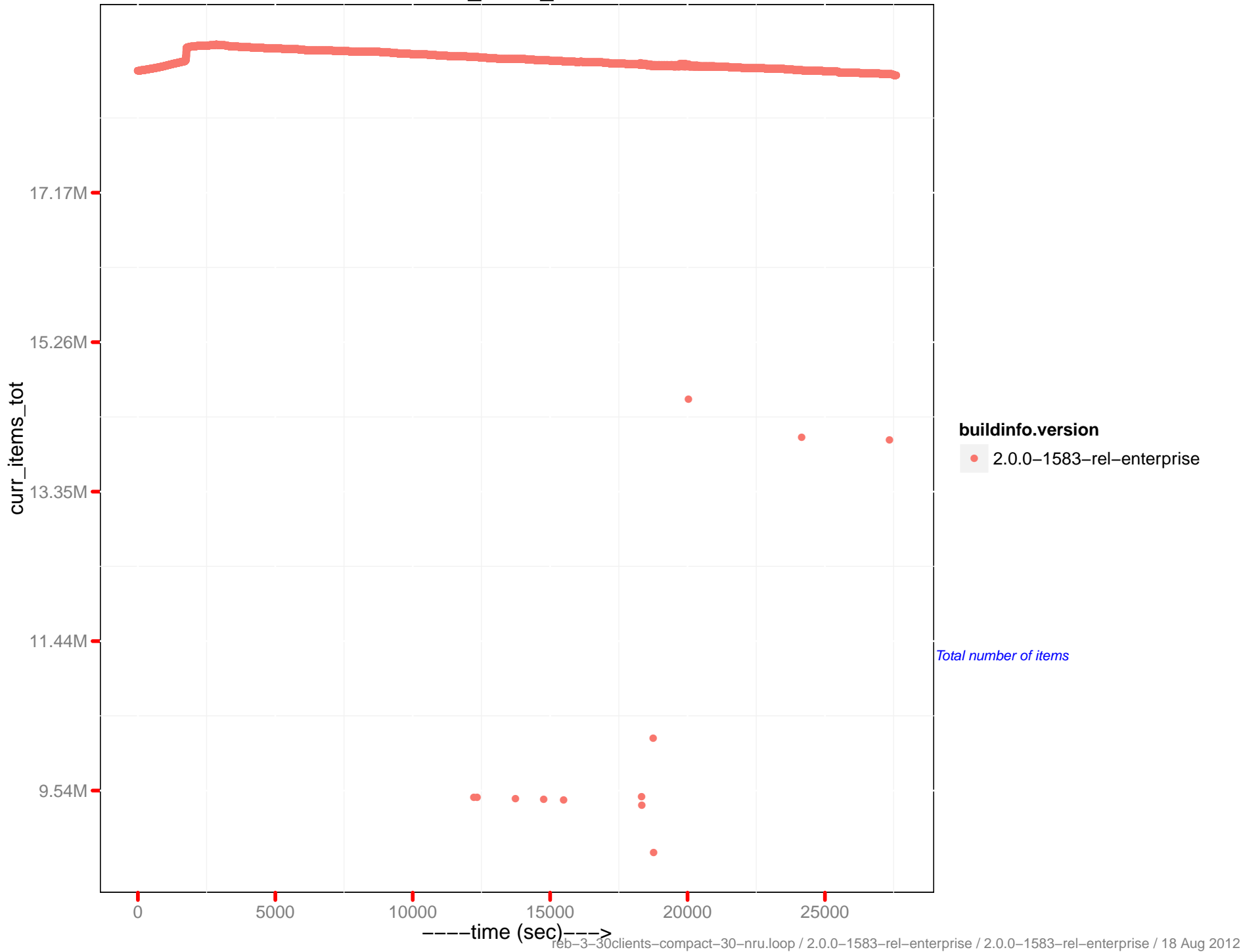
buildinfo.version
● 2.0.0-1583-rel-enterprise

Percentage of replica items cached in RAM

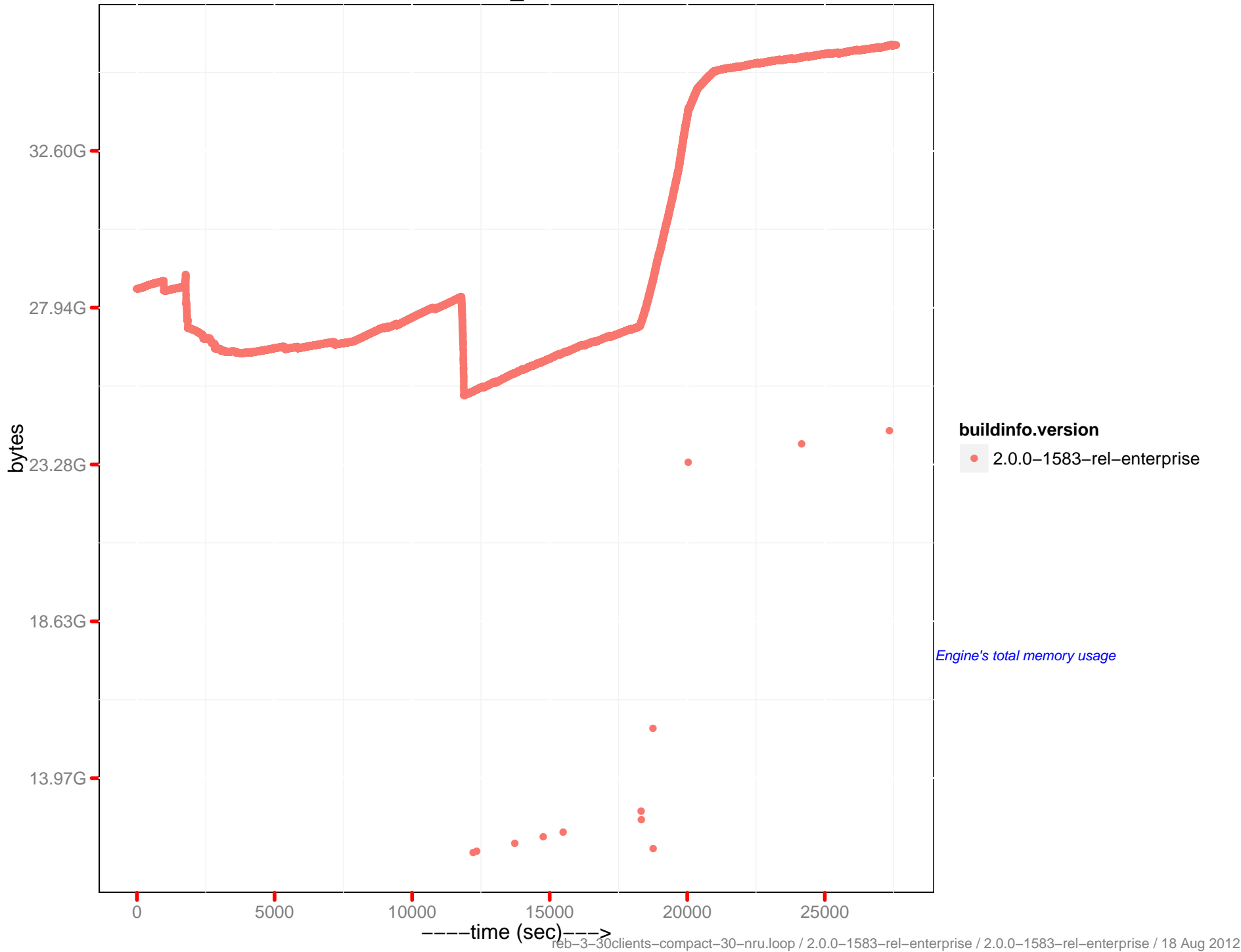
curr_items



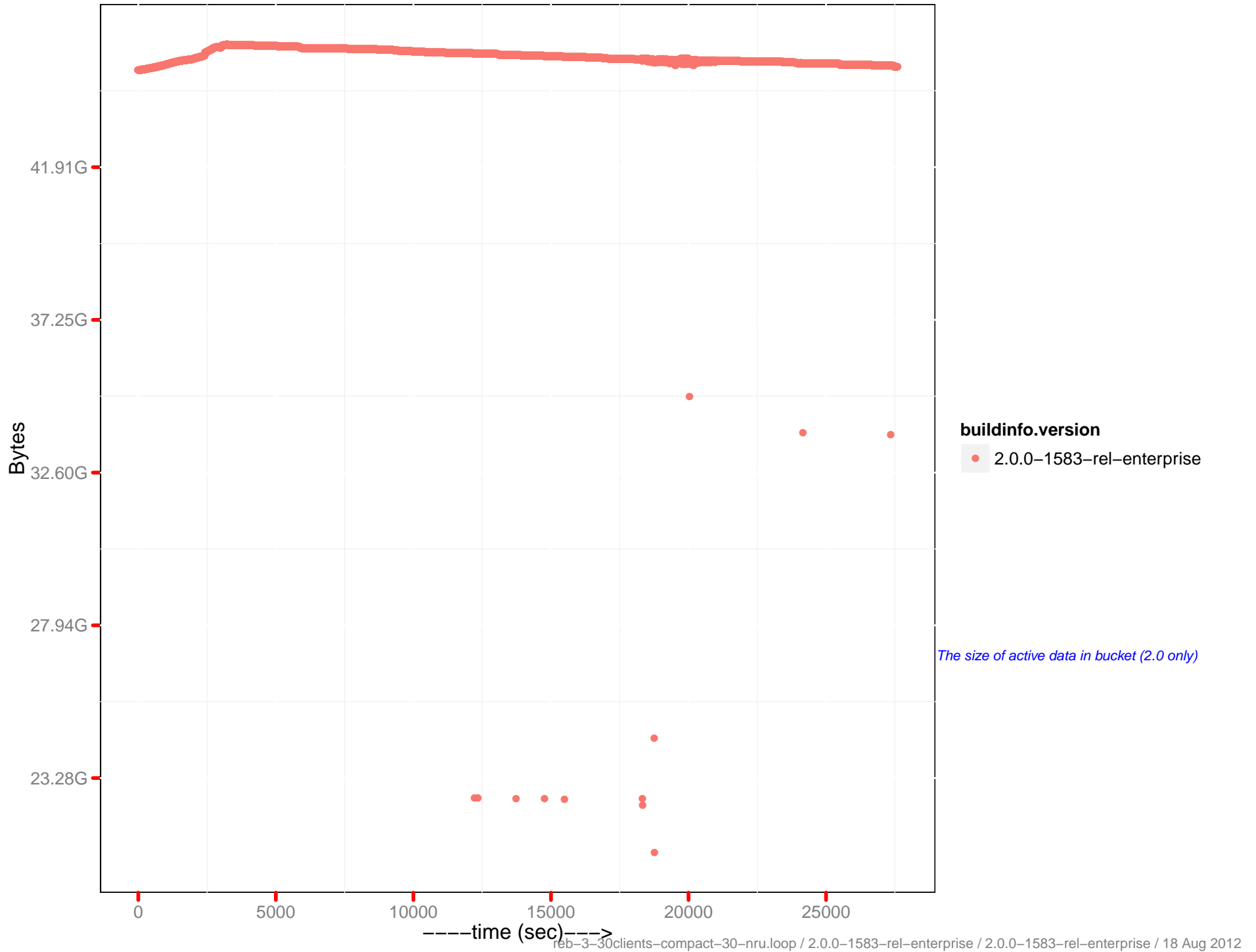
cur_items_total



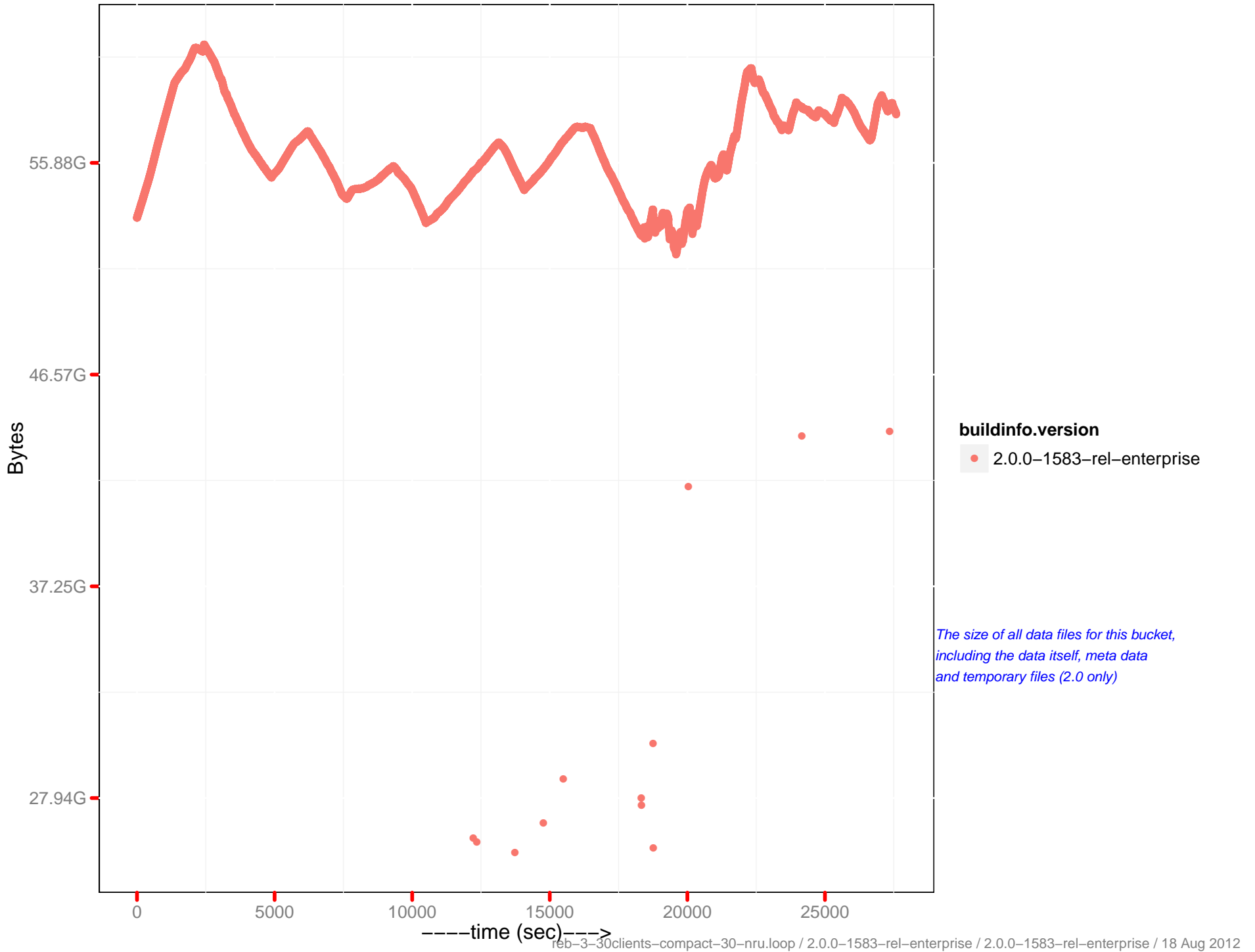
mem_used



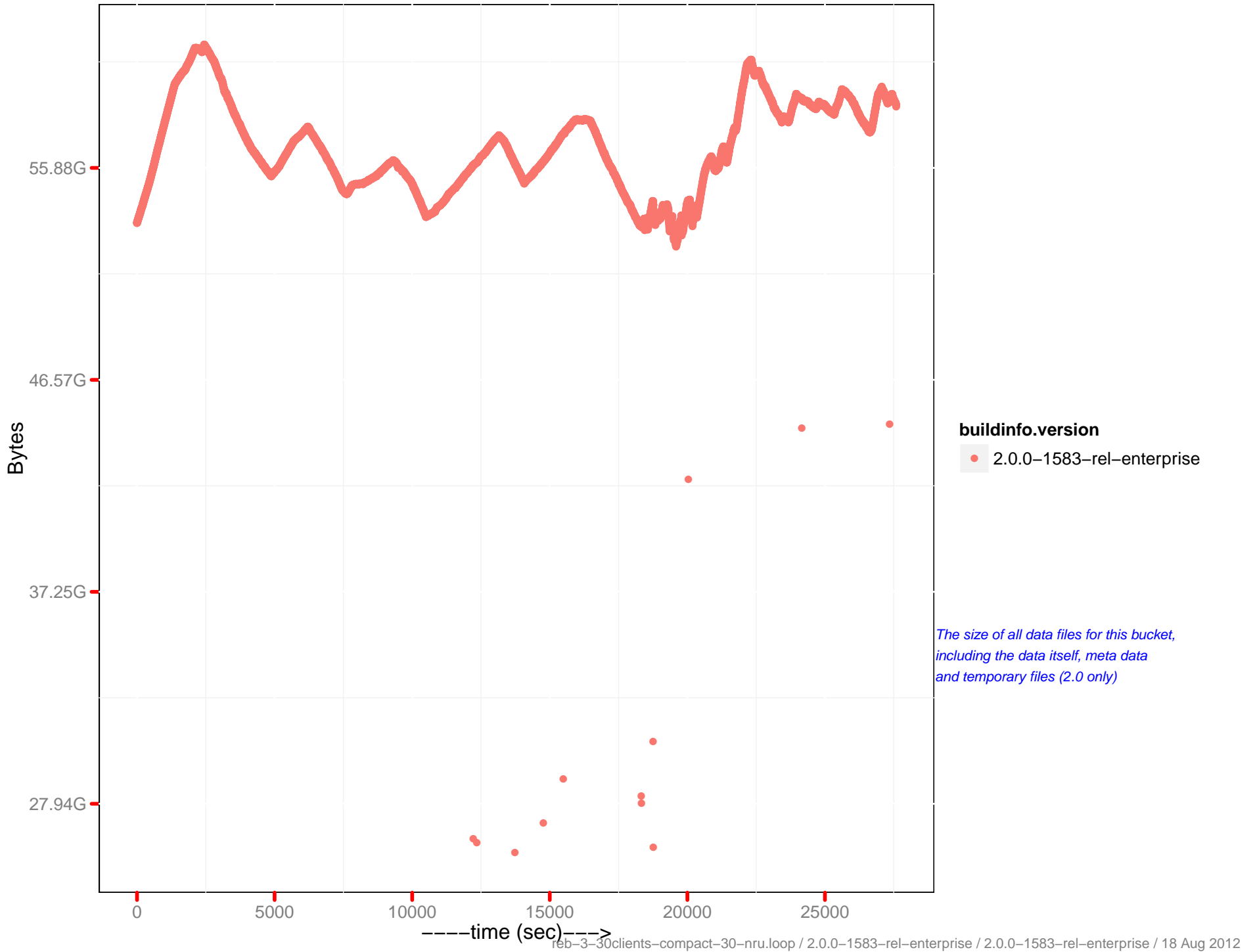
Docs data size



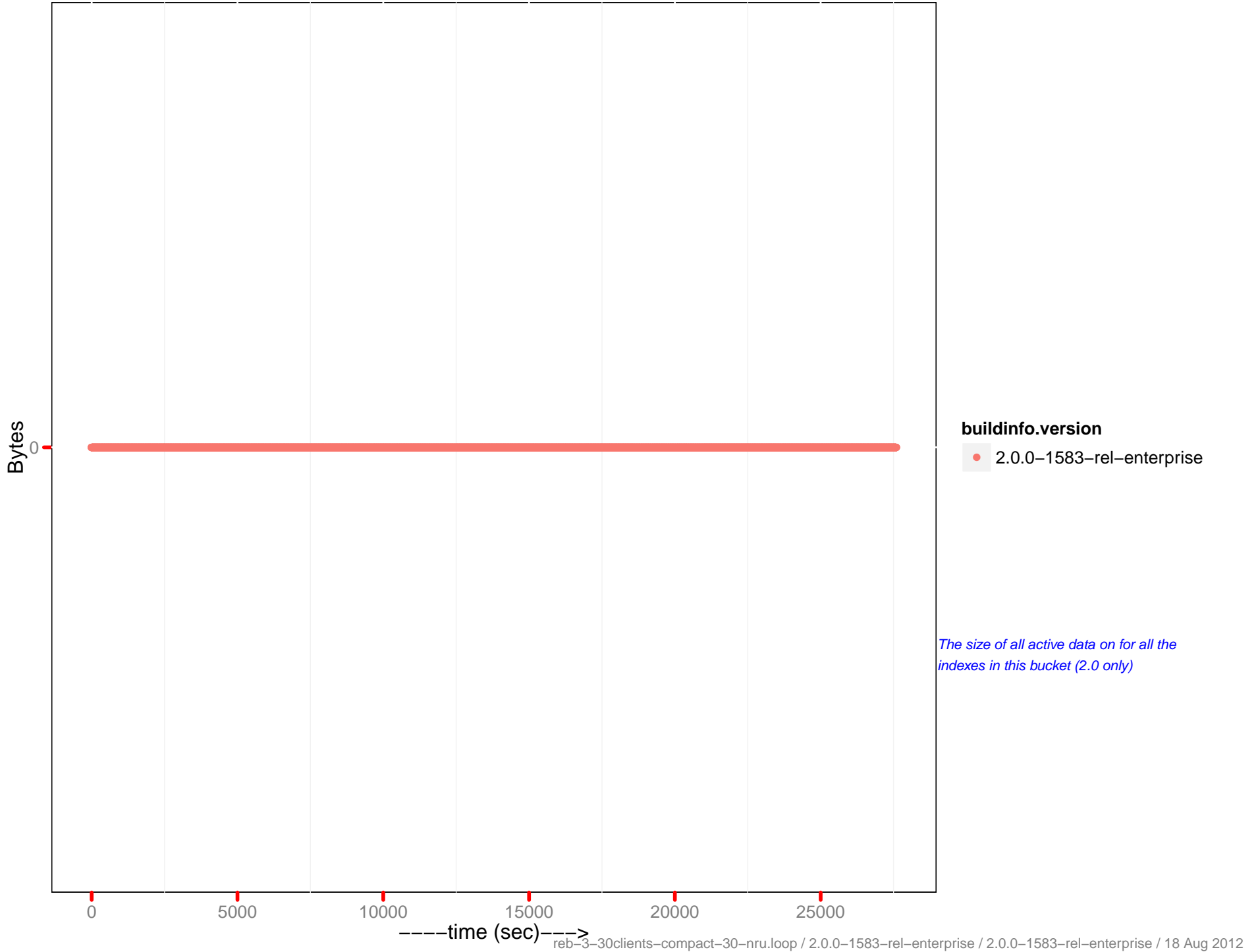
Docs disk size



Docs actual disk size



Views data size

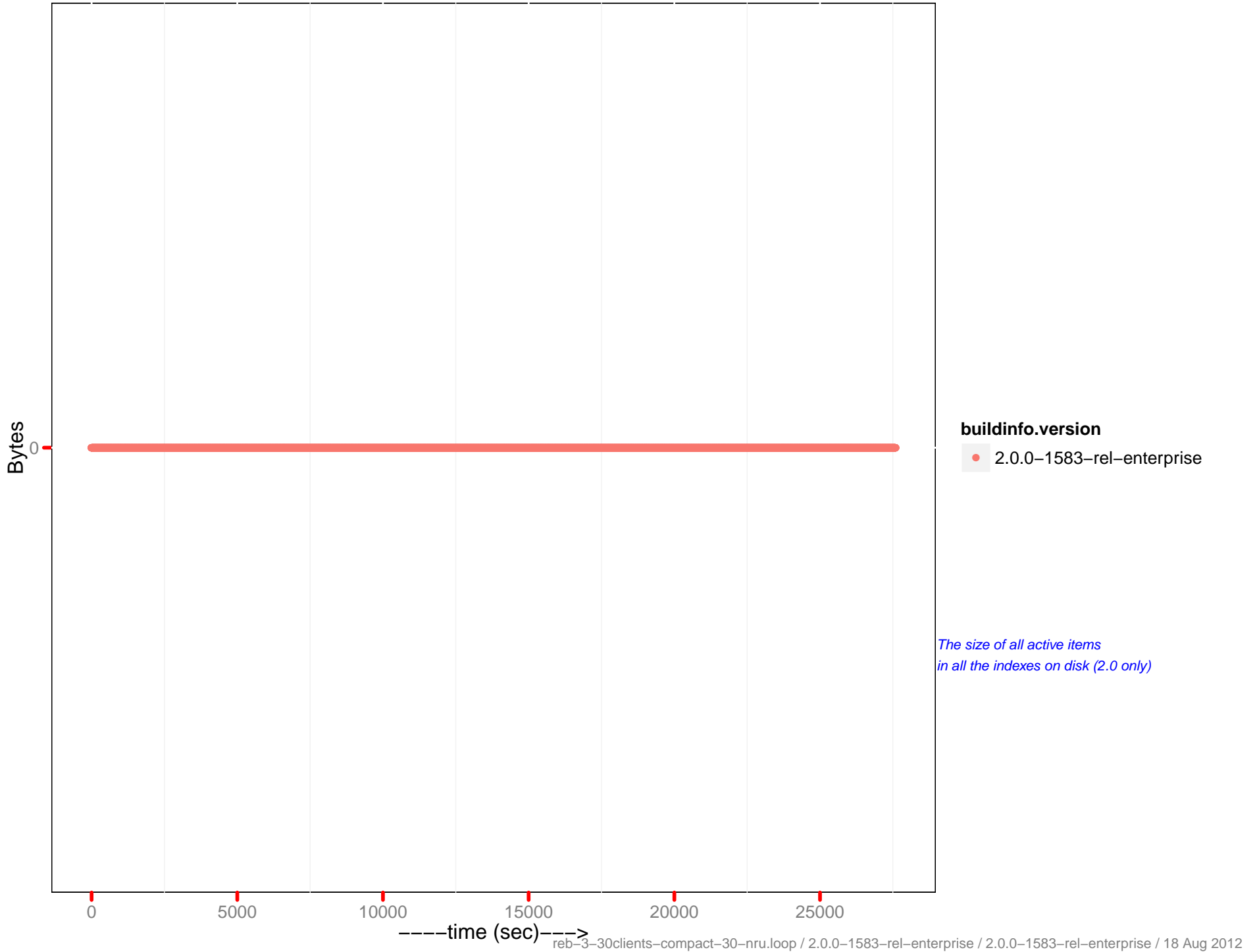


buildinfo.version

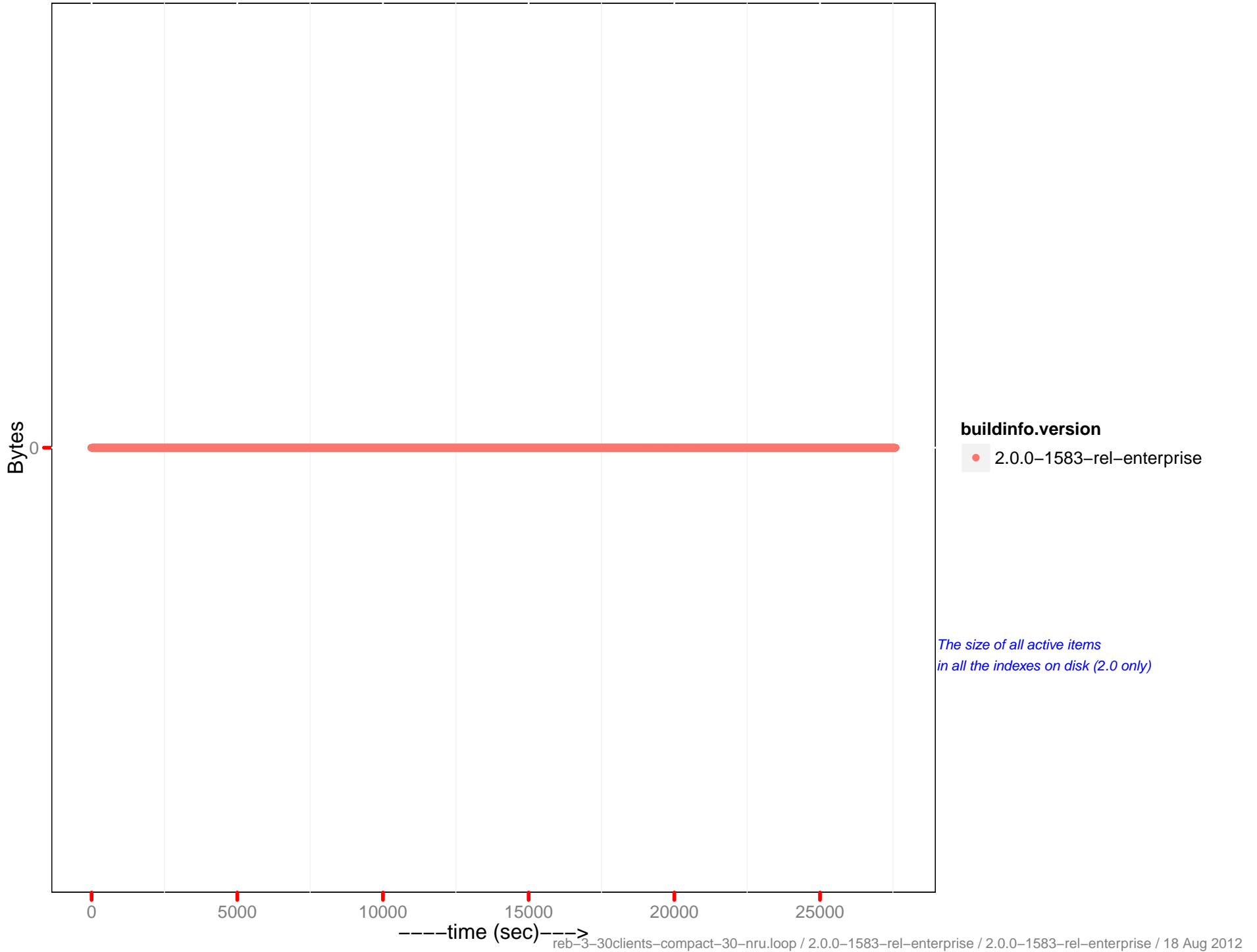
• 2.0.0-1583-rel-enterprise

The size of all active data on for all the indexes in this bucket (2.0 only)

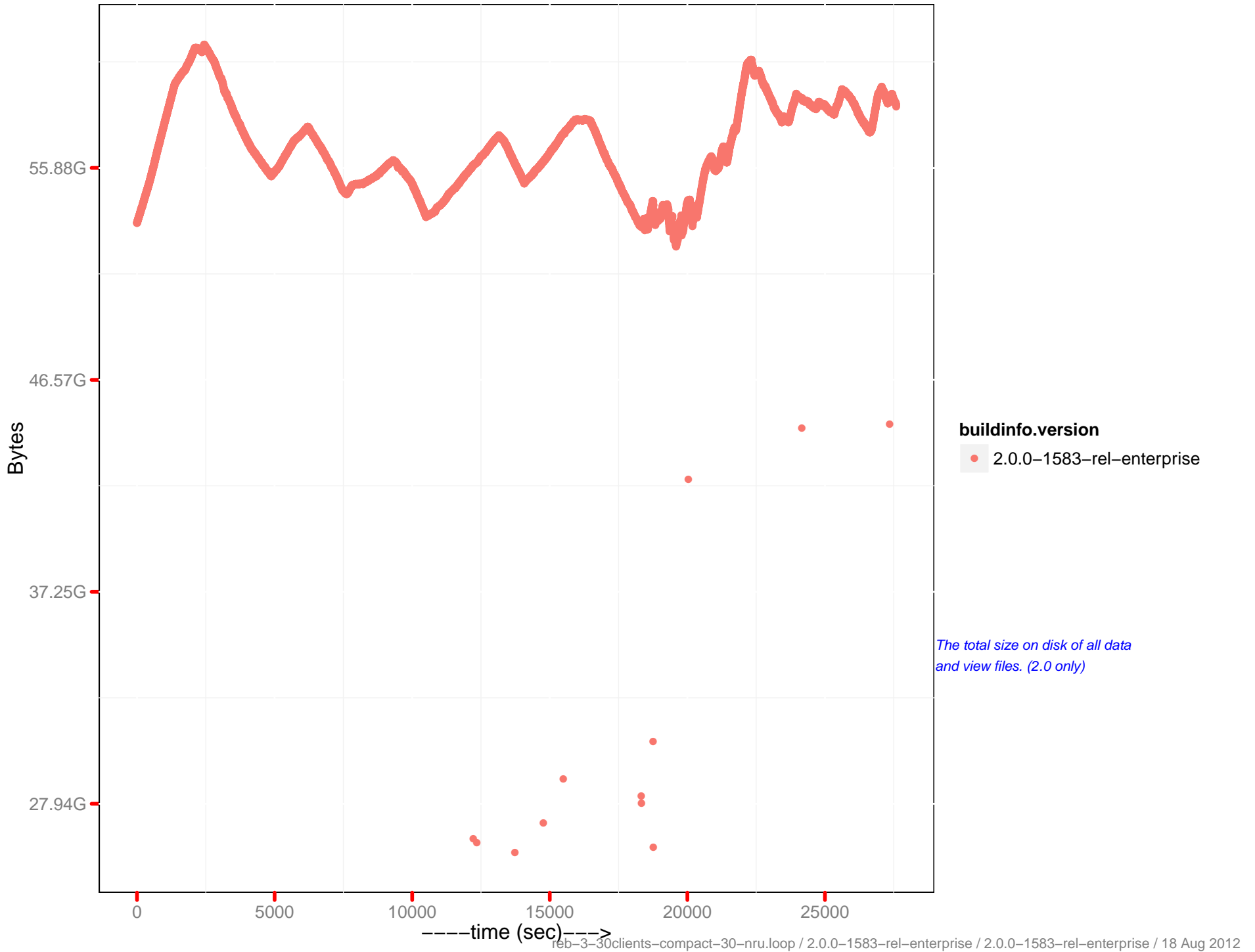
Views disk size



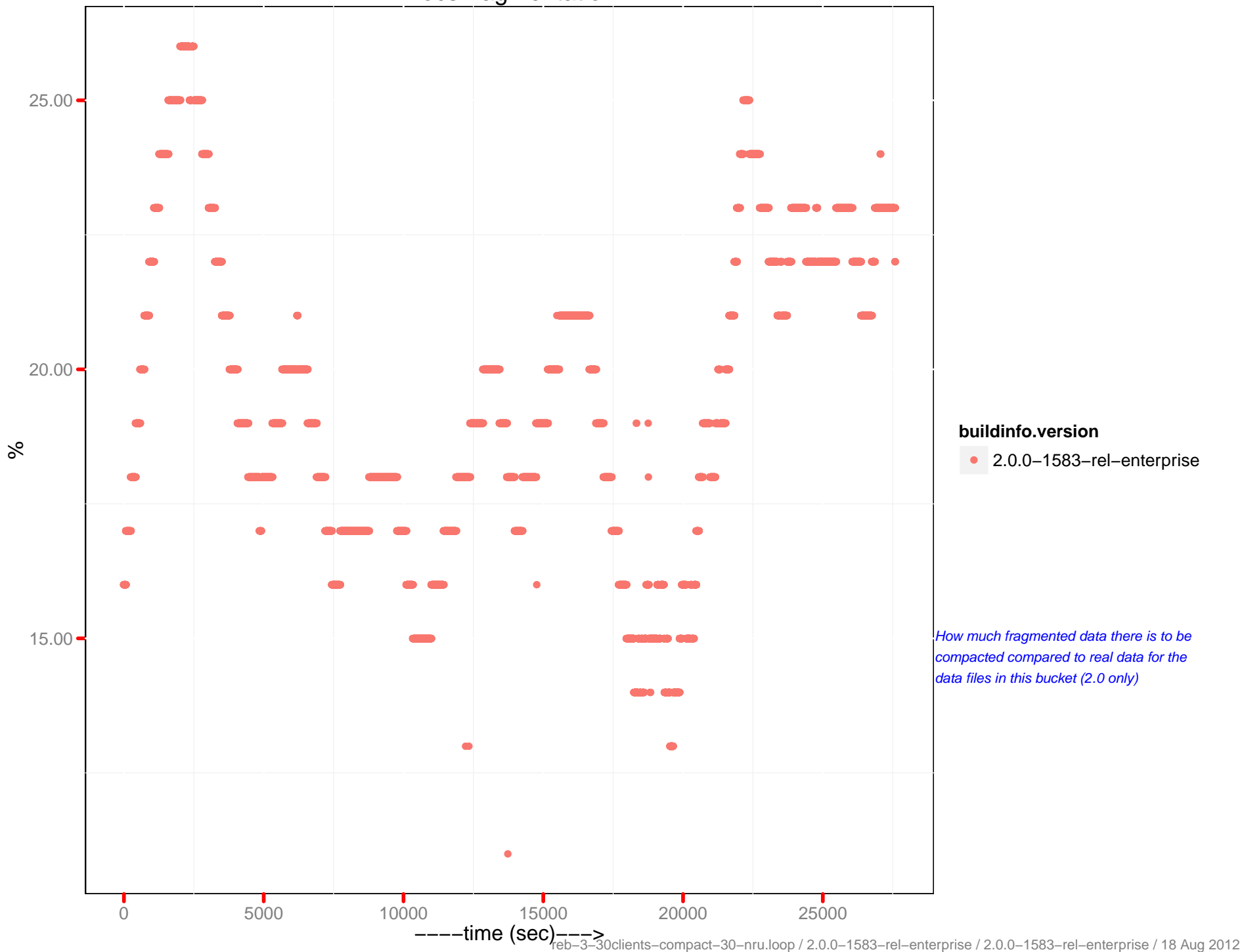
Views actual disk size



Total disk size

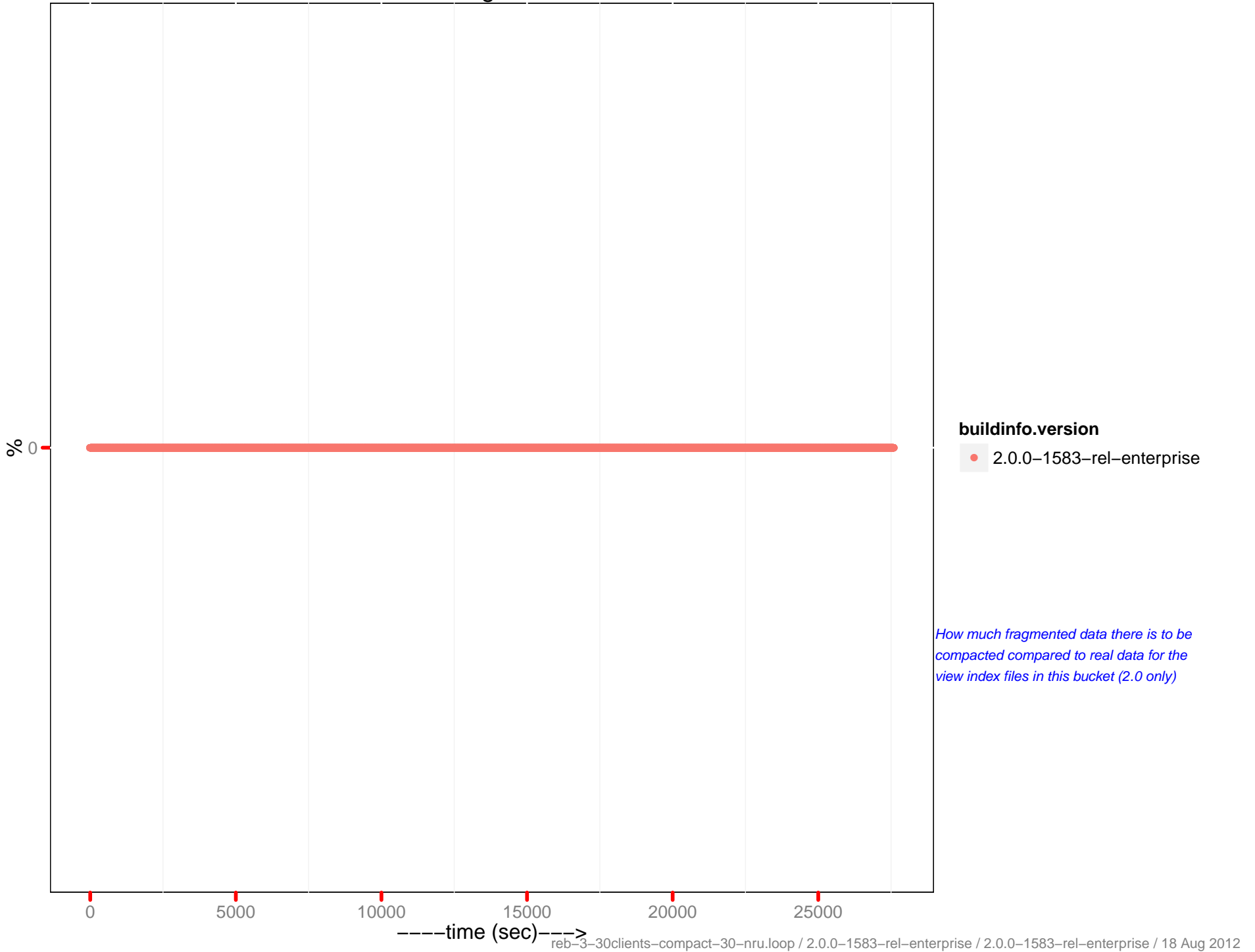


Docs fragmentation



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

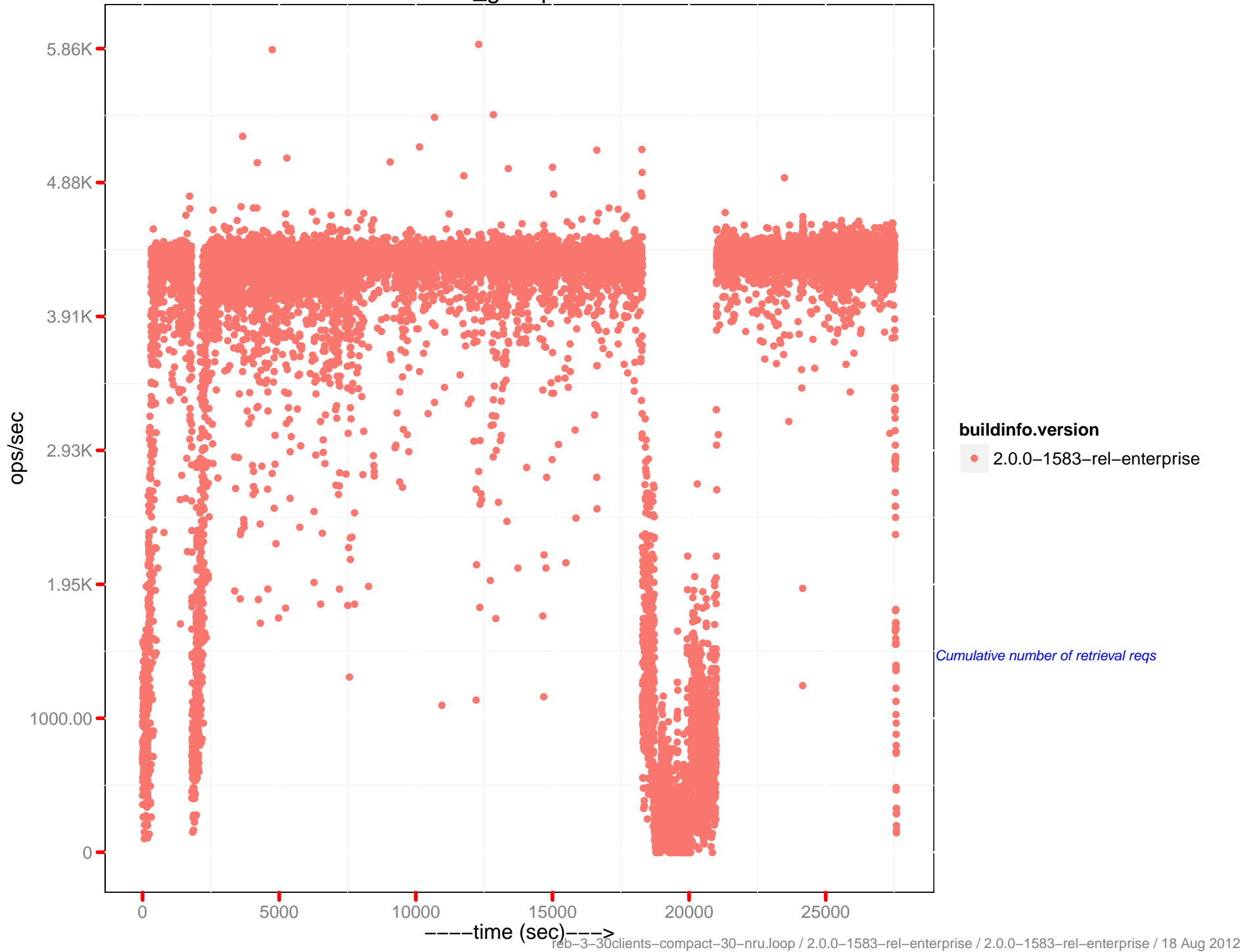
Views fragmentation



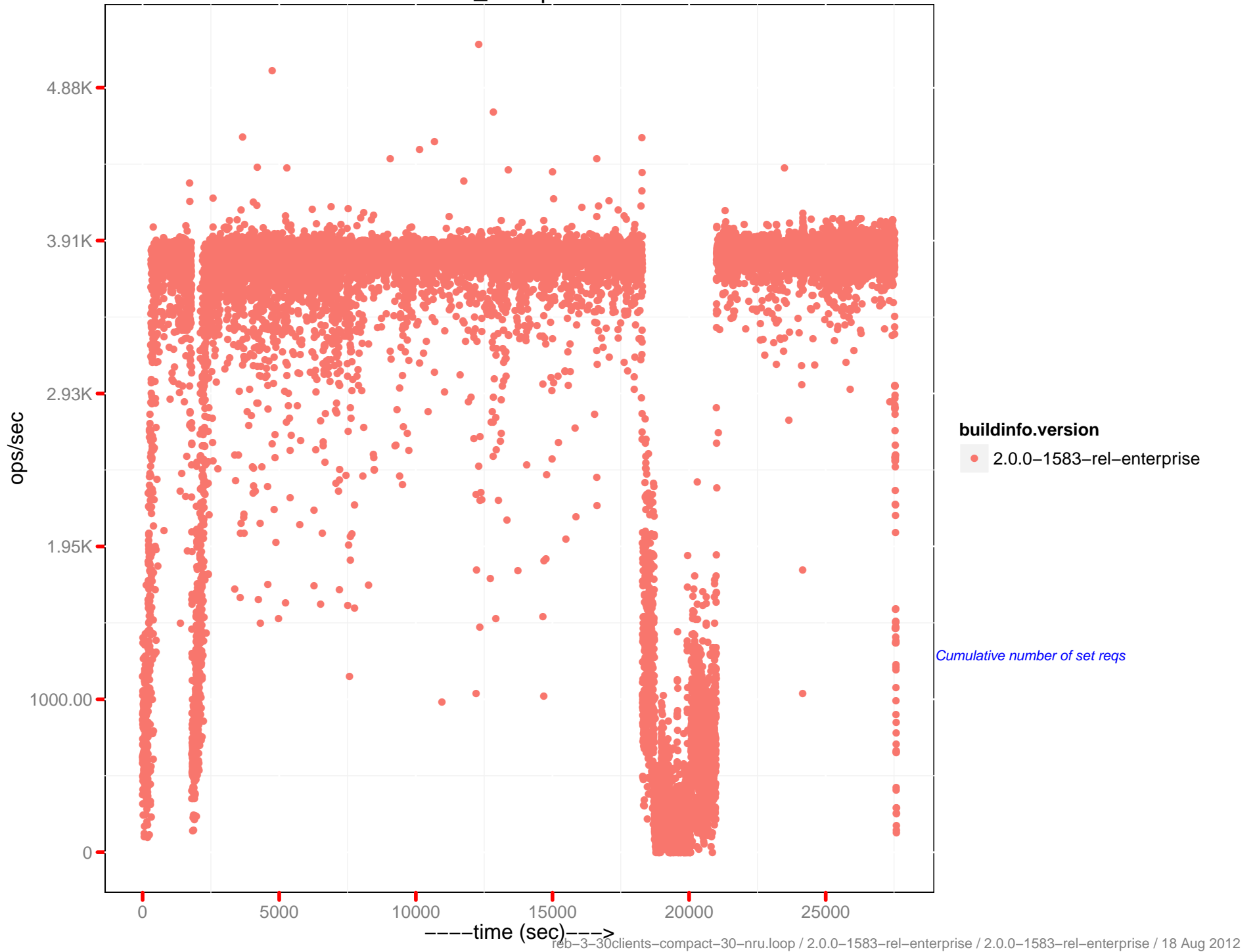
buildinfo.version
● 2.0.0-1583-rel-enterprise

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

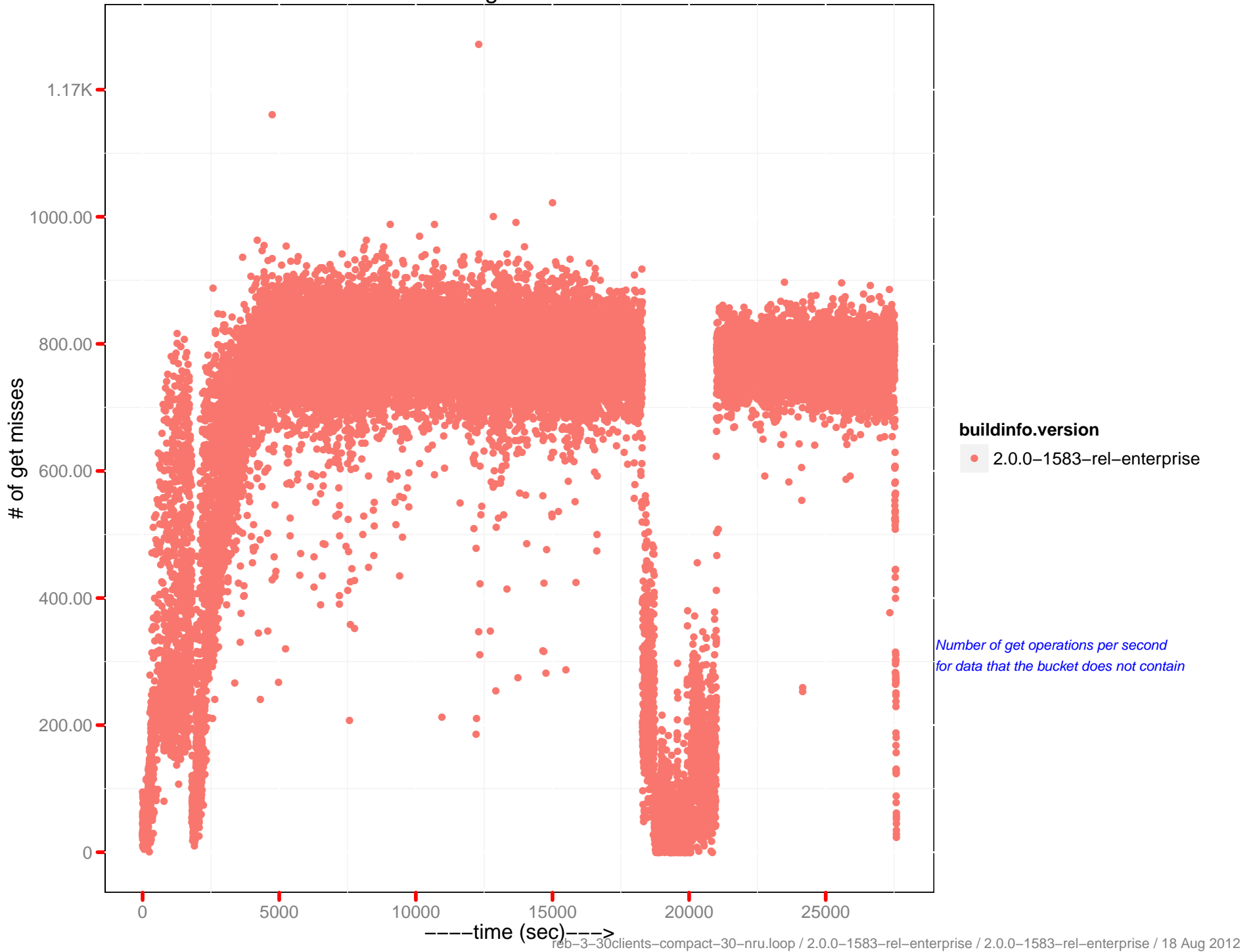
cmd_get ops/sec



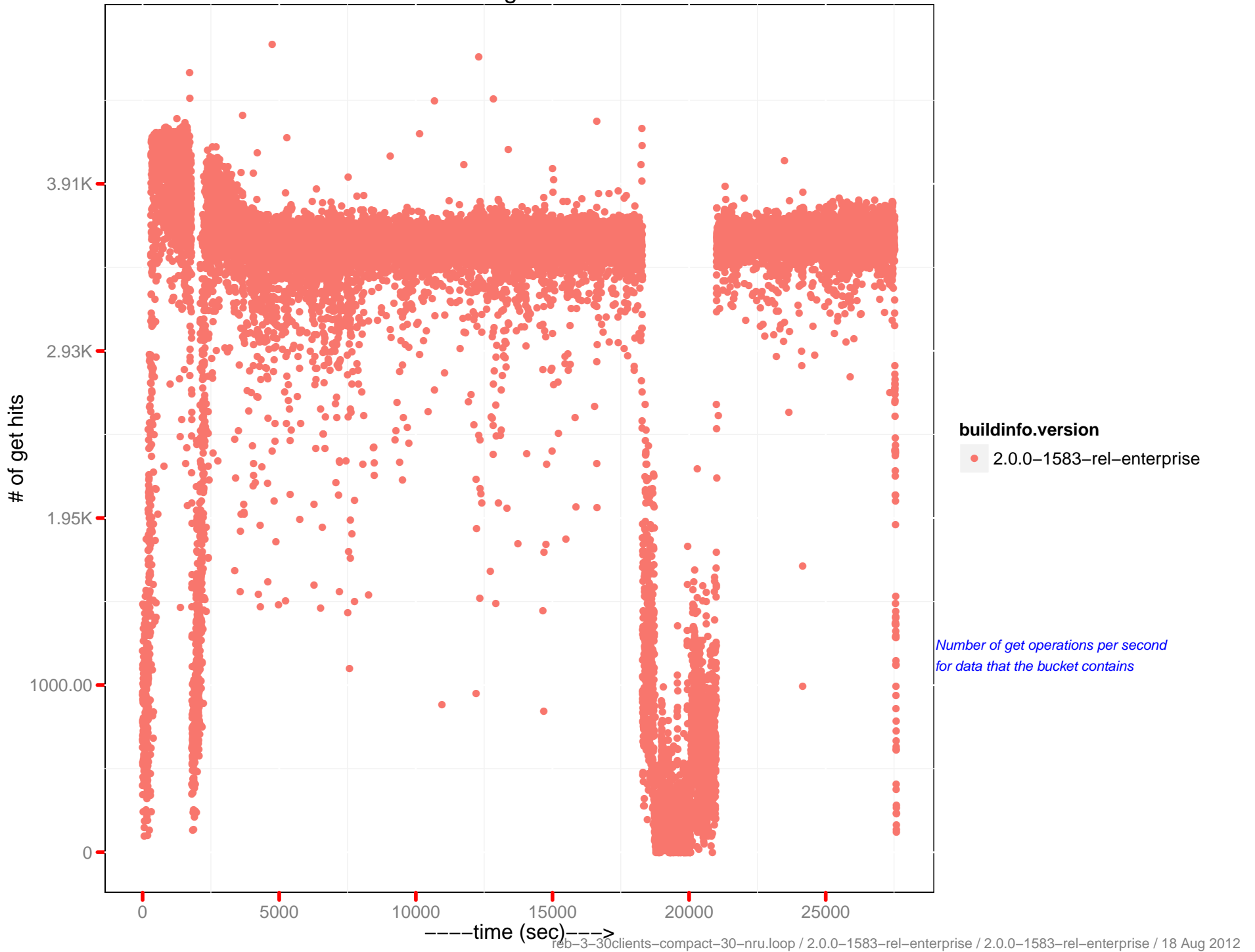
cmd_set ops/sec



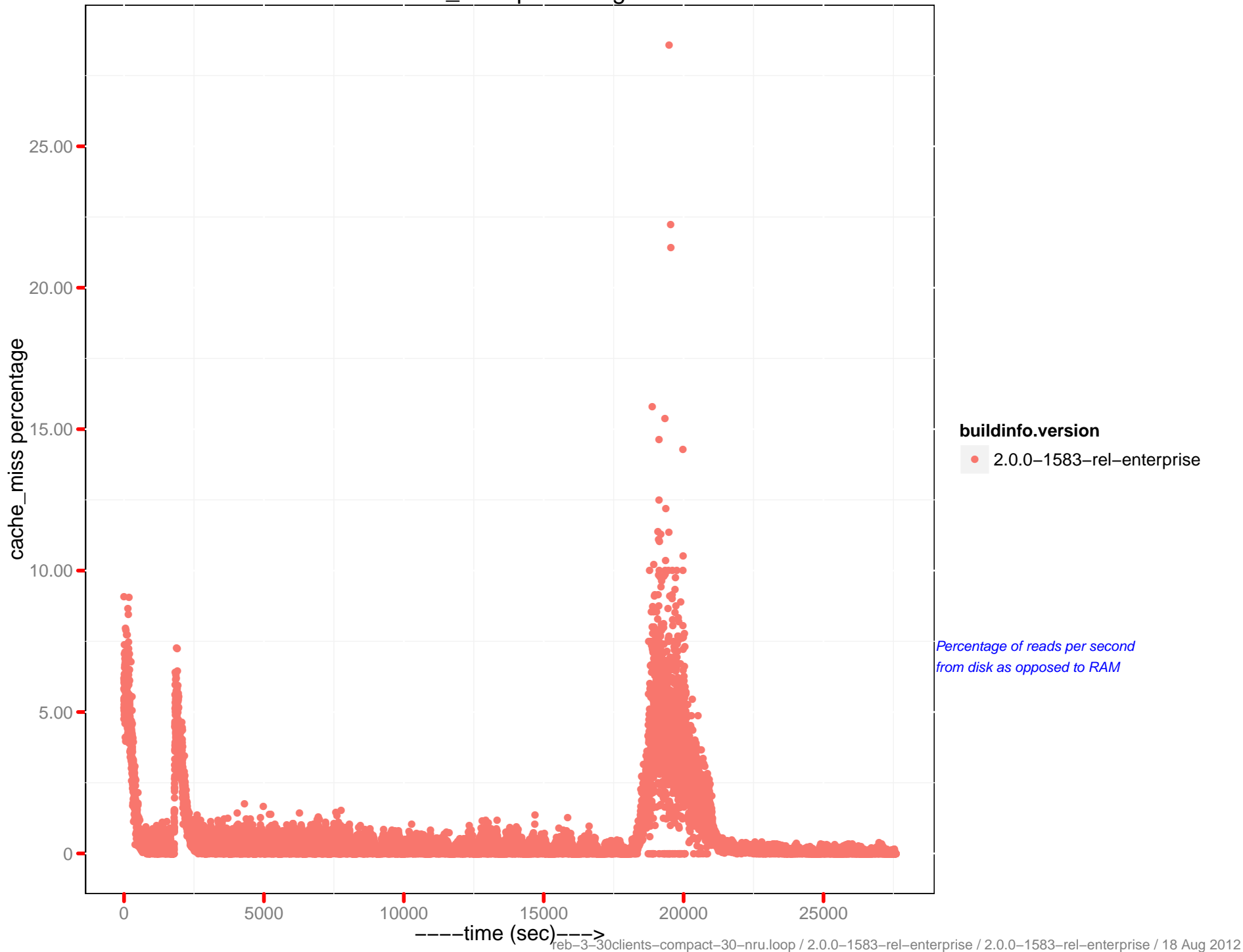
of get misses



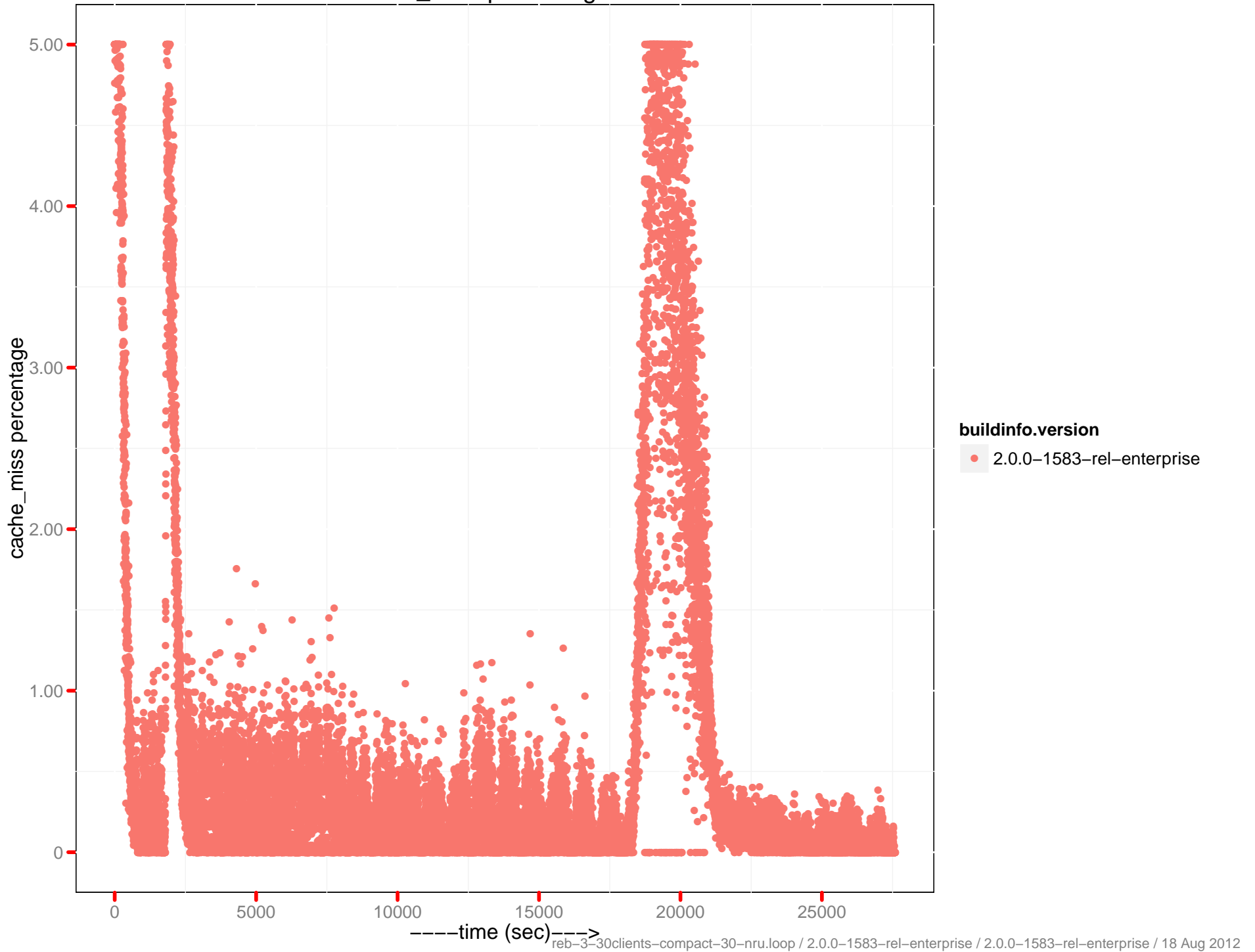
of get hits



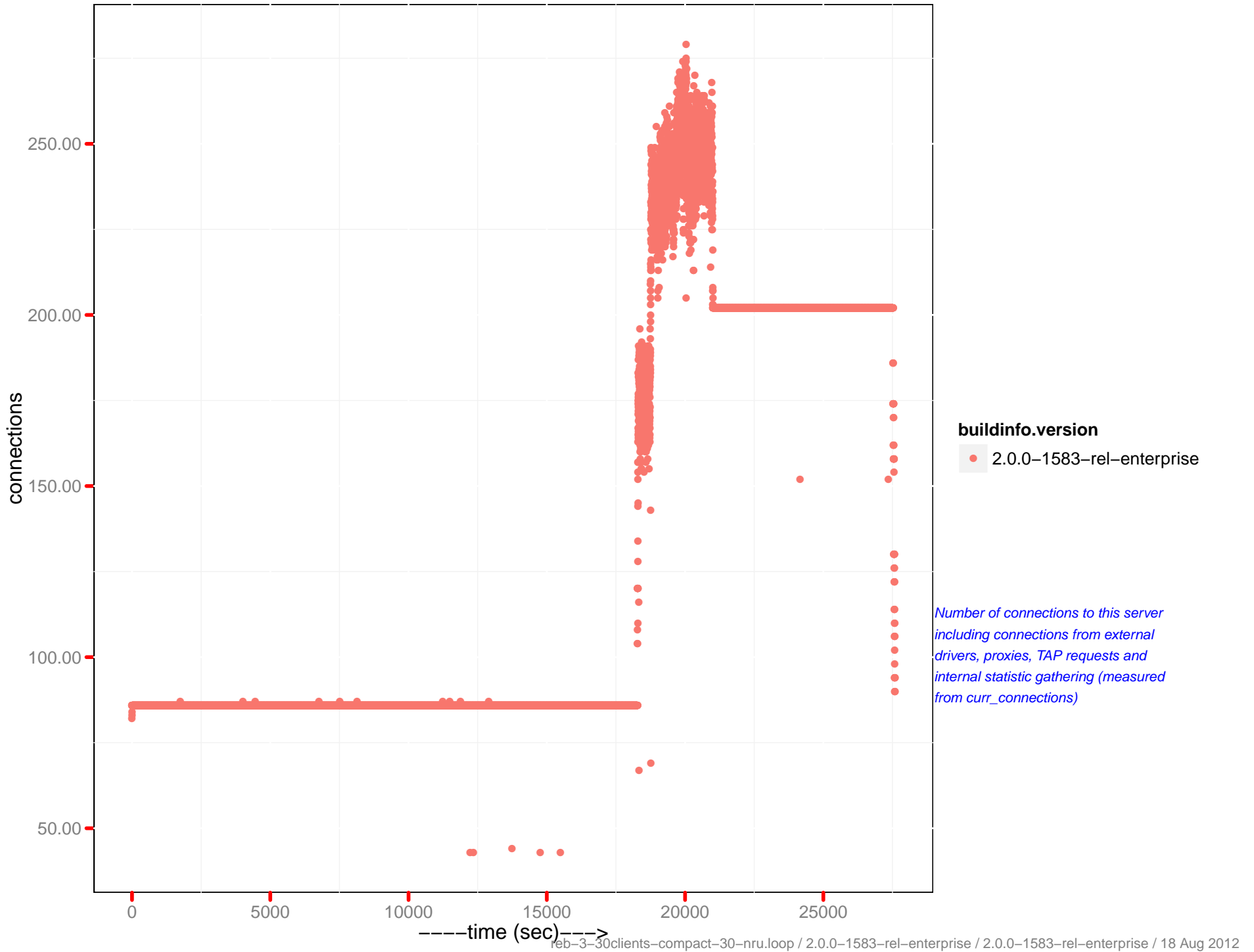
cache_miss percentage



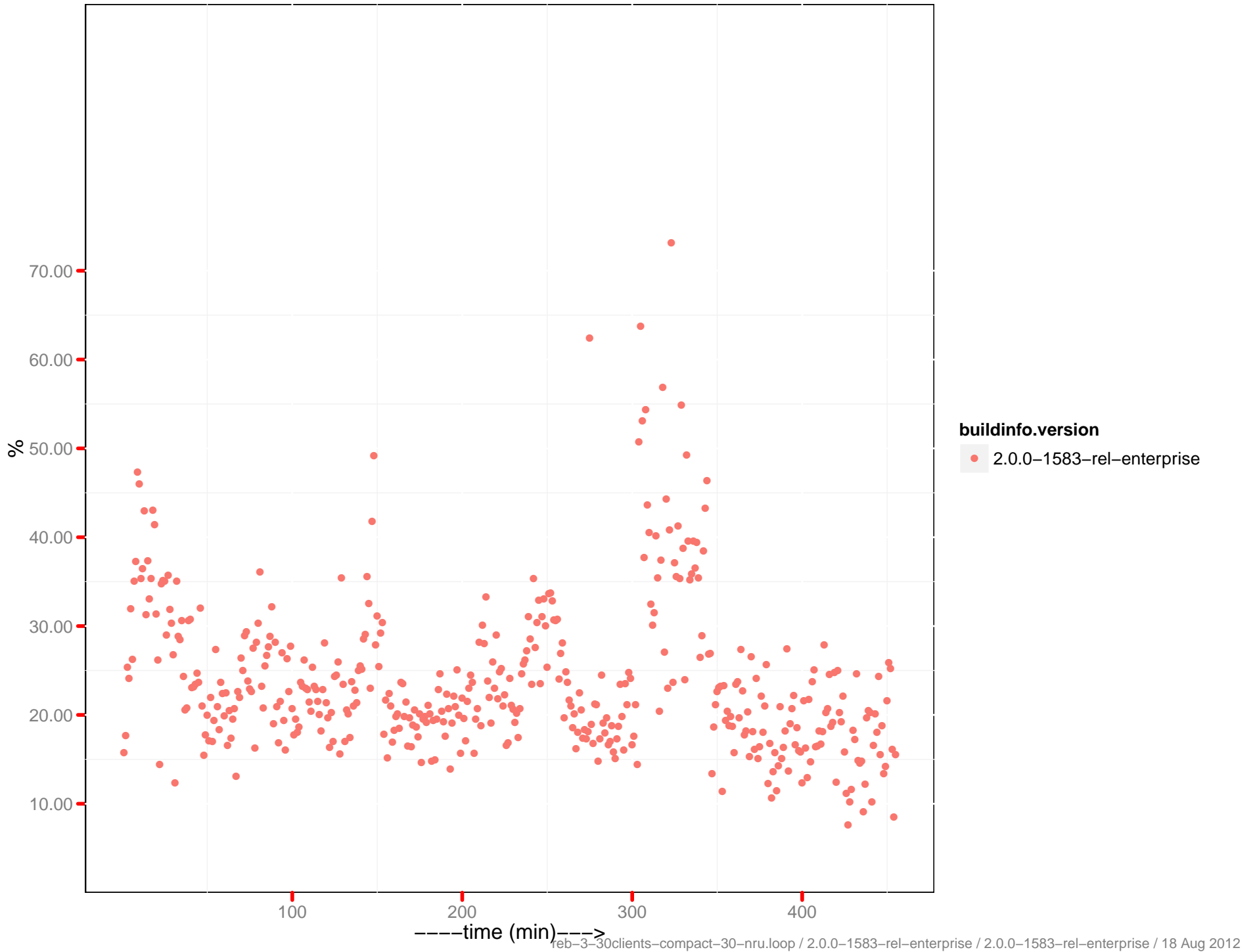
cache_miss percentage 0-5



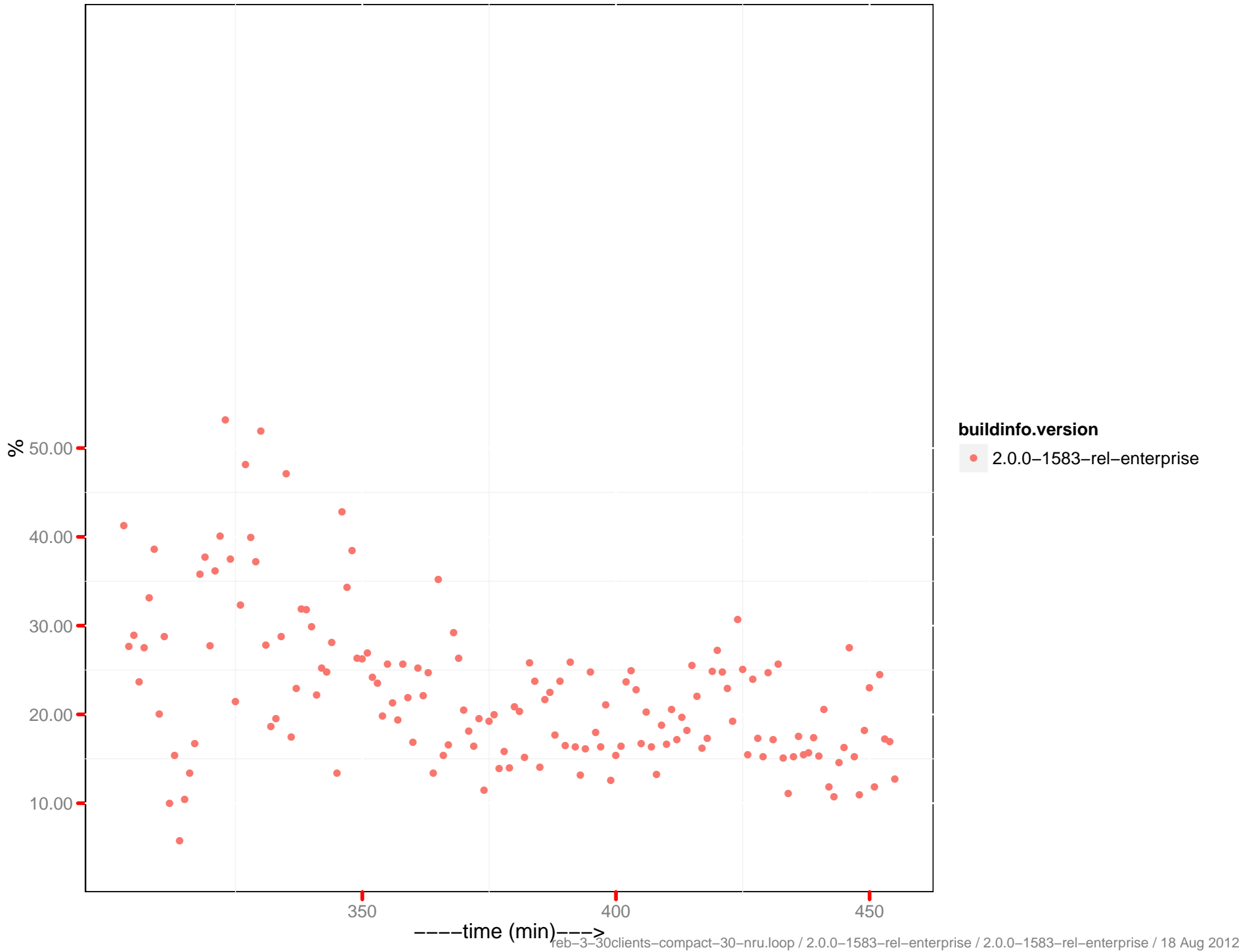
Number of connections



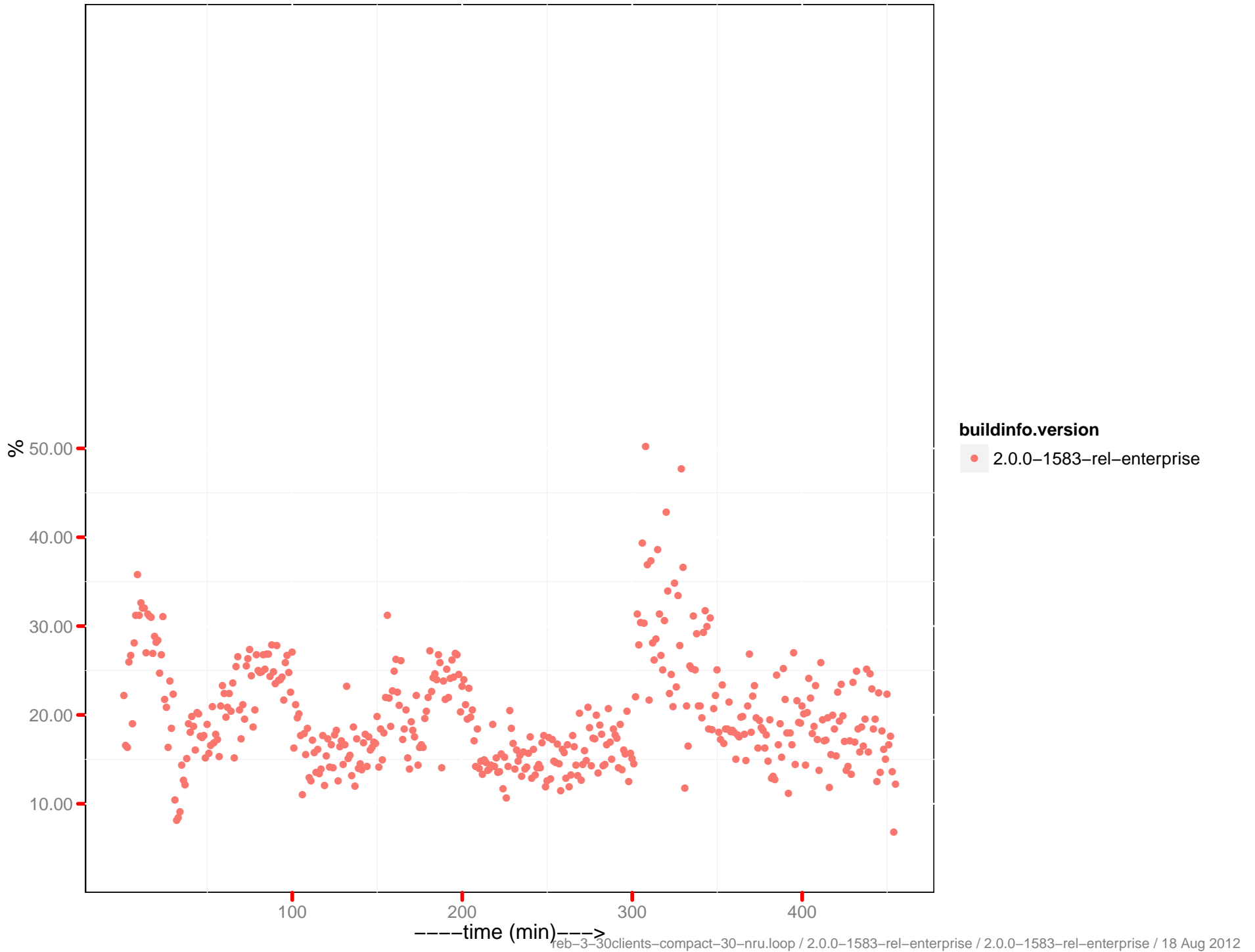
CPU utilization – 192.168.162.20:8091



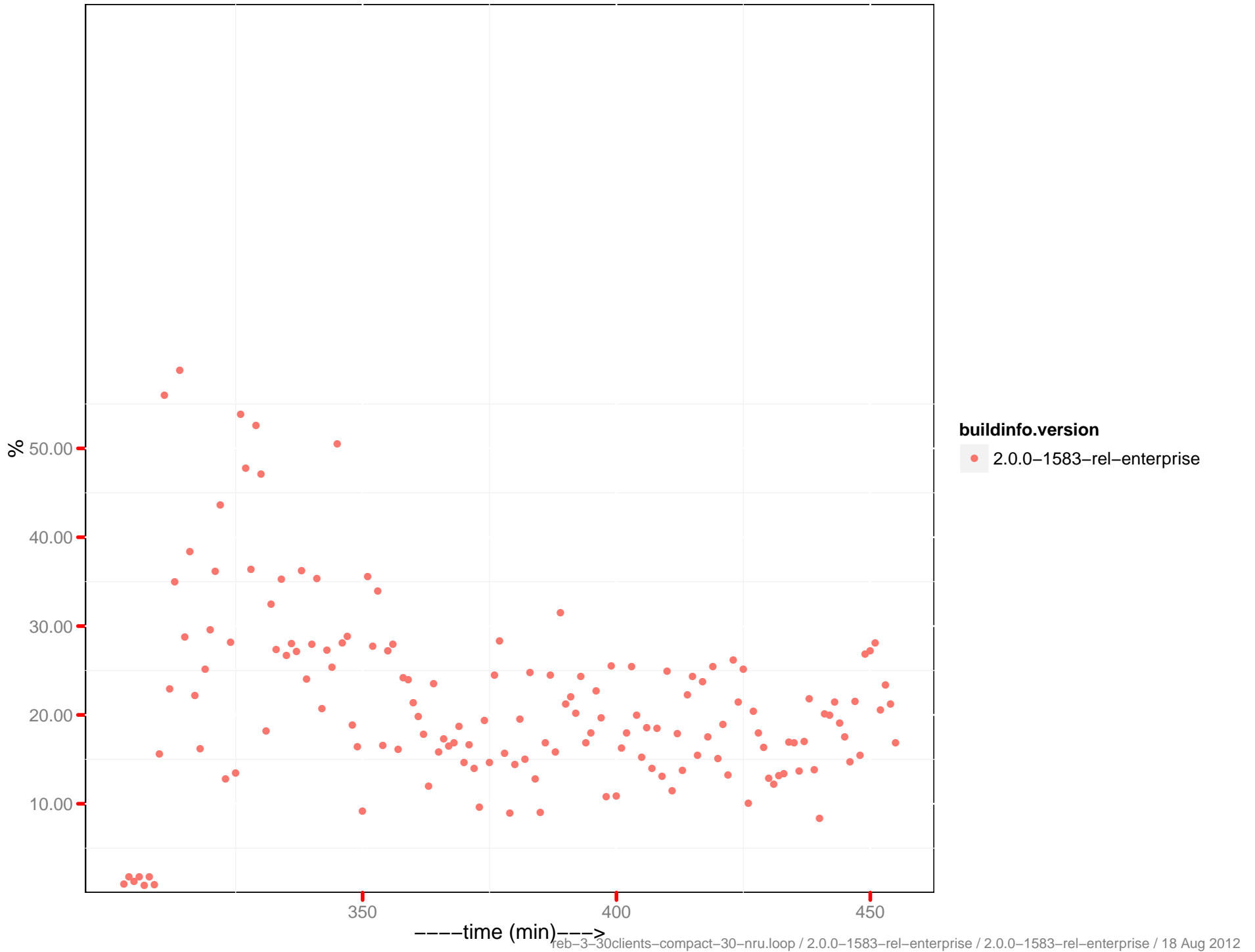
CPU utilization – 192.168.162.21:8091



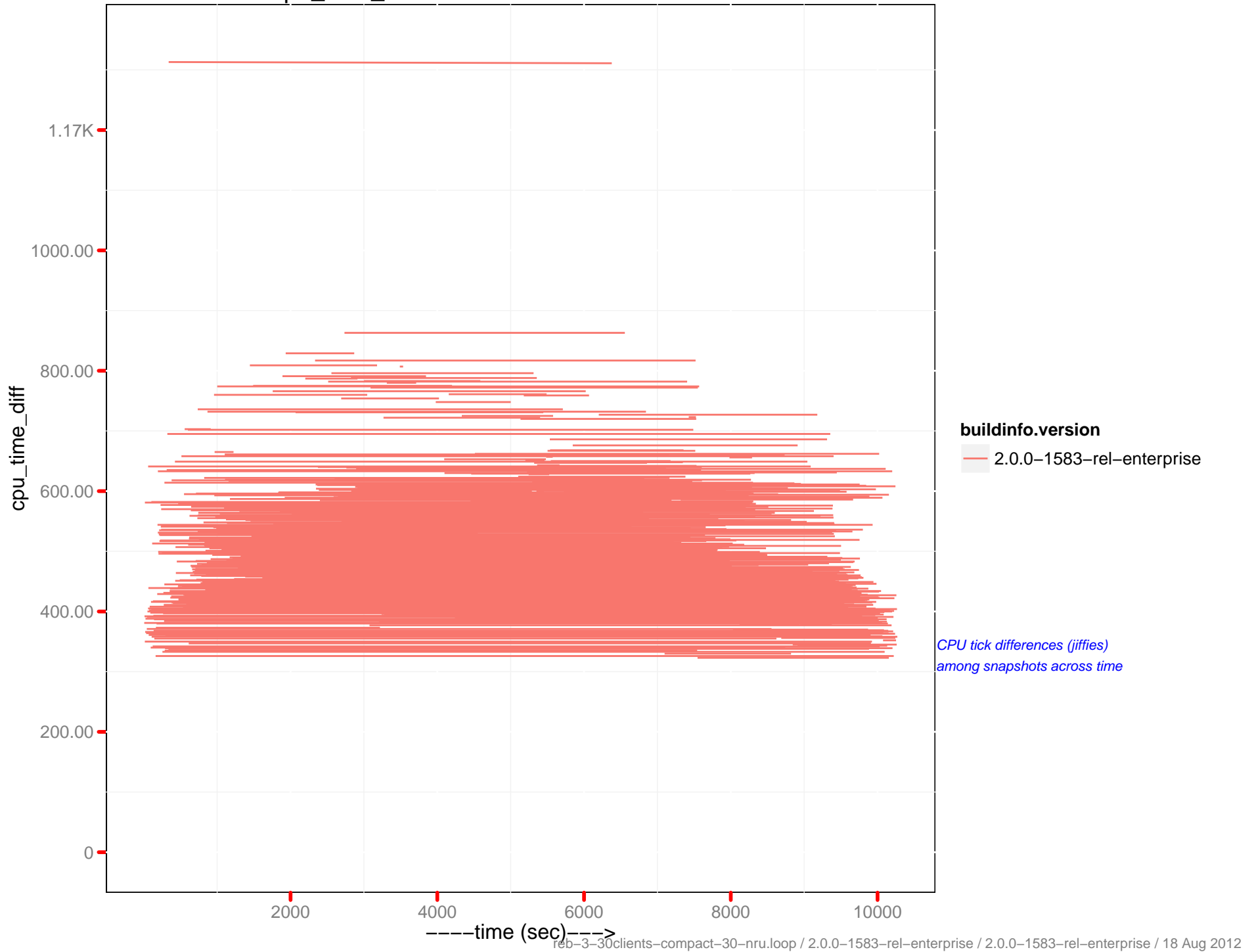
CPU utilization – 192.168.162.22:8091



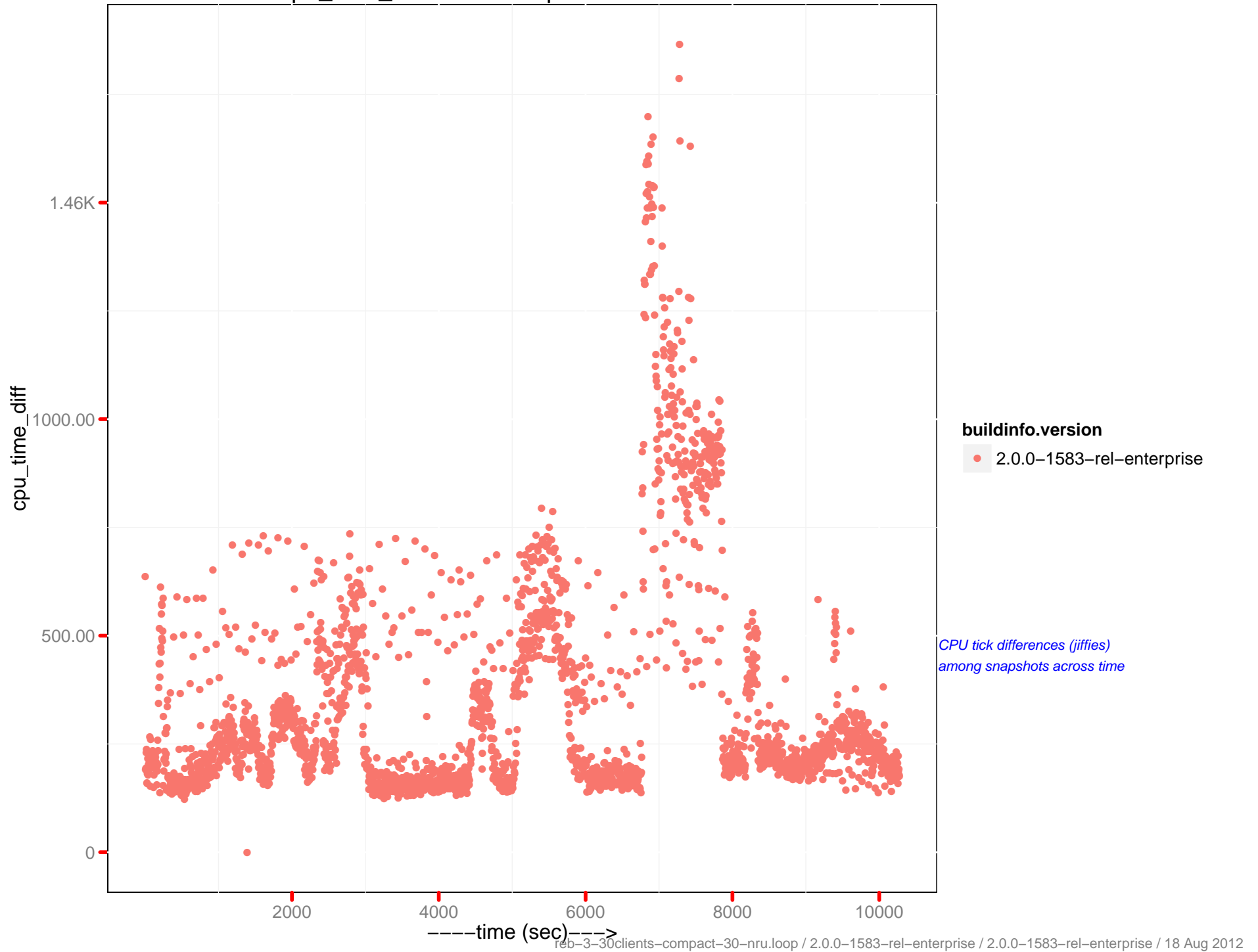
CPU utilization – 192.168.162.23:8091



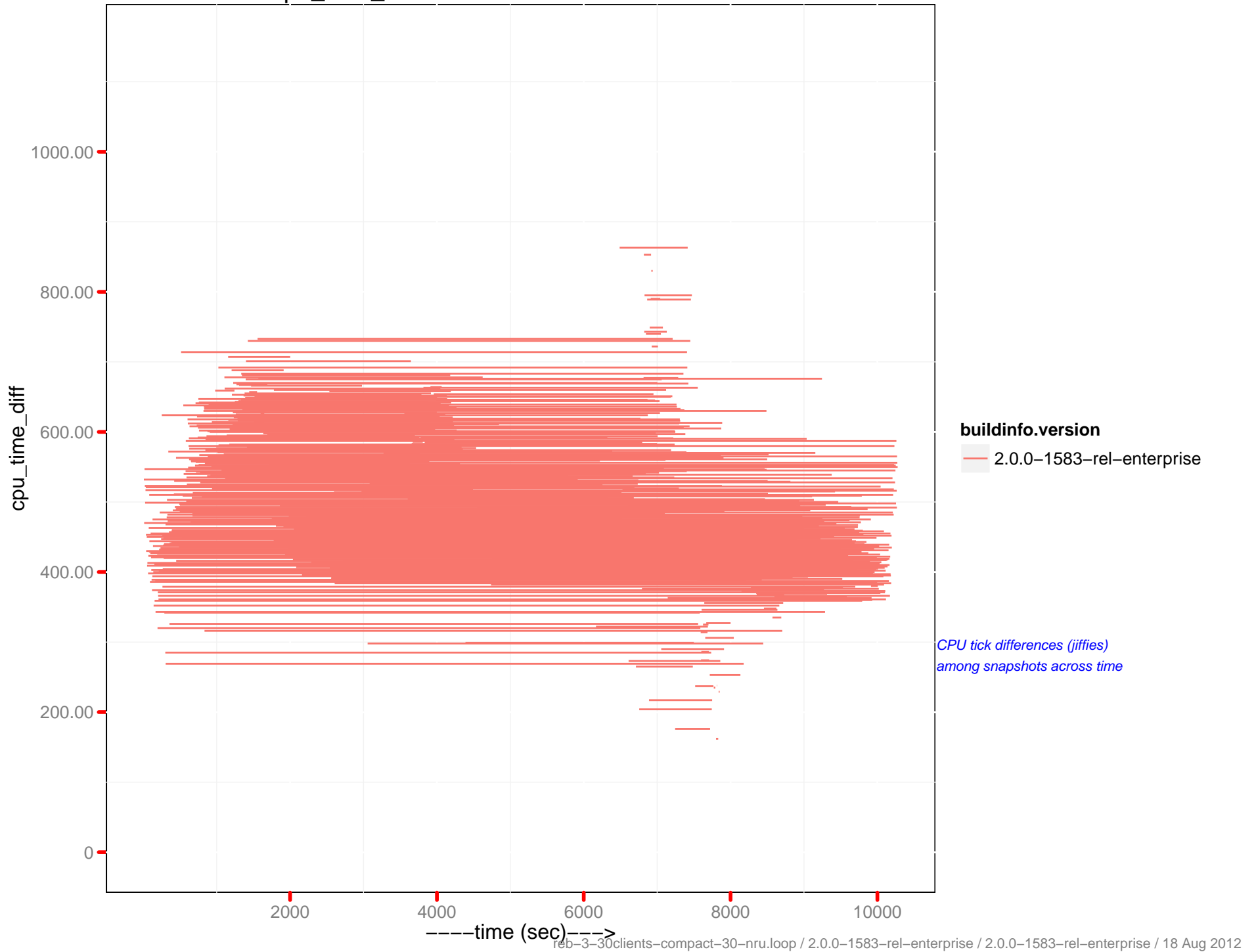
cpu_time_diff: memcached - 192.168.162.20



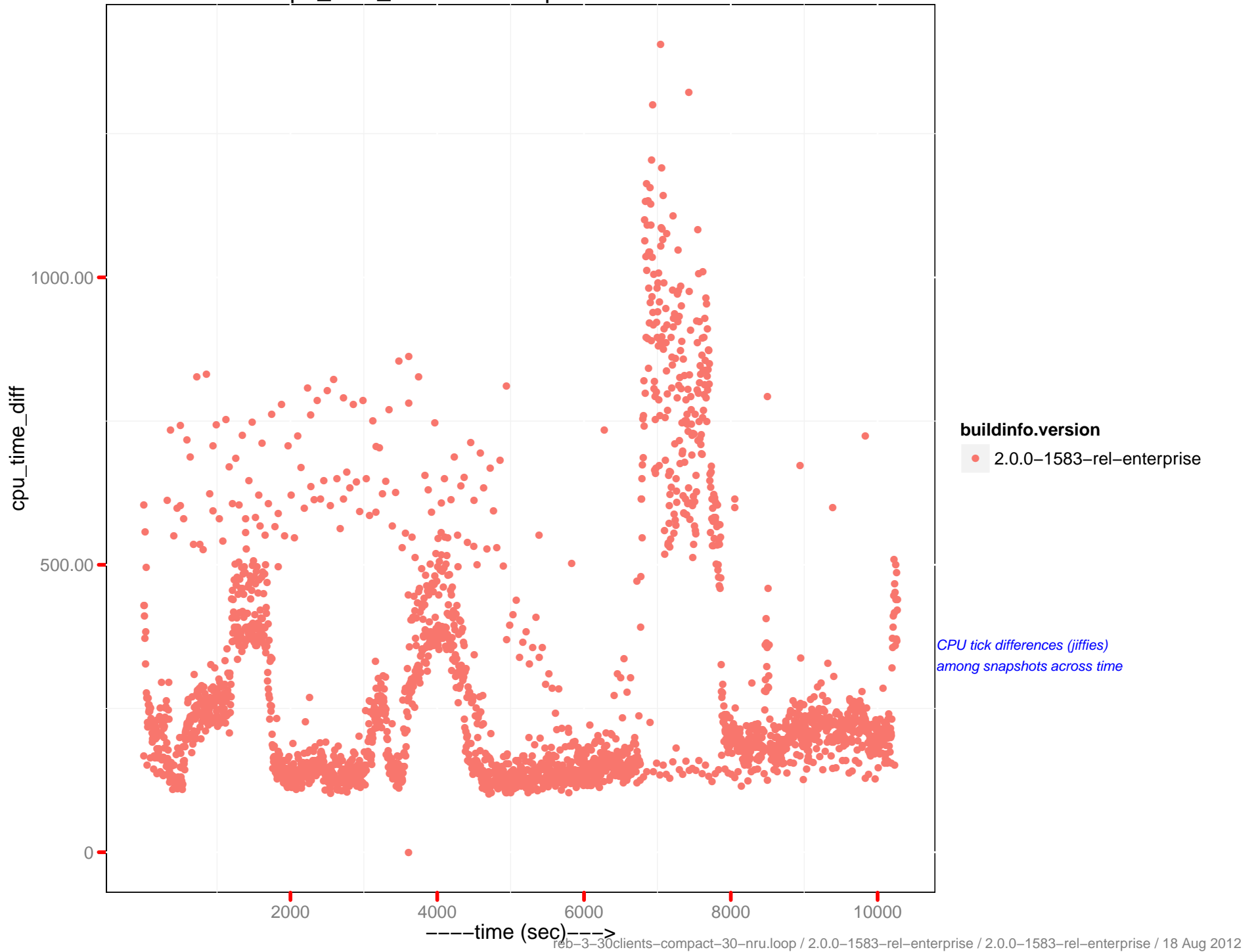
cpu_time_diff : beam.smp - 192.168.162.20



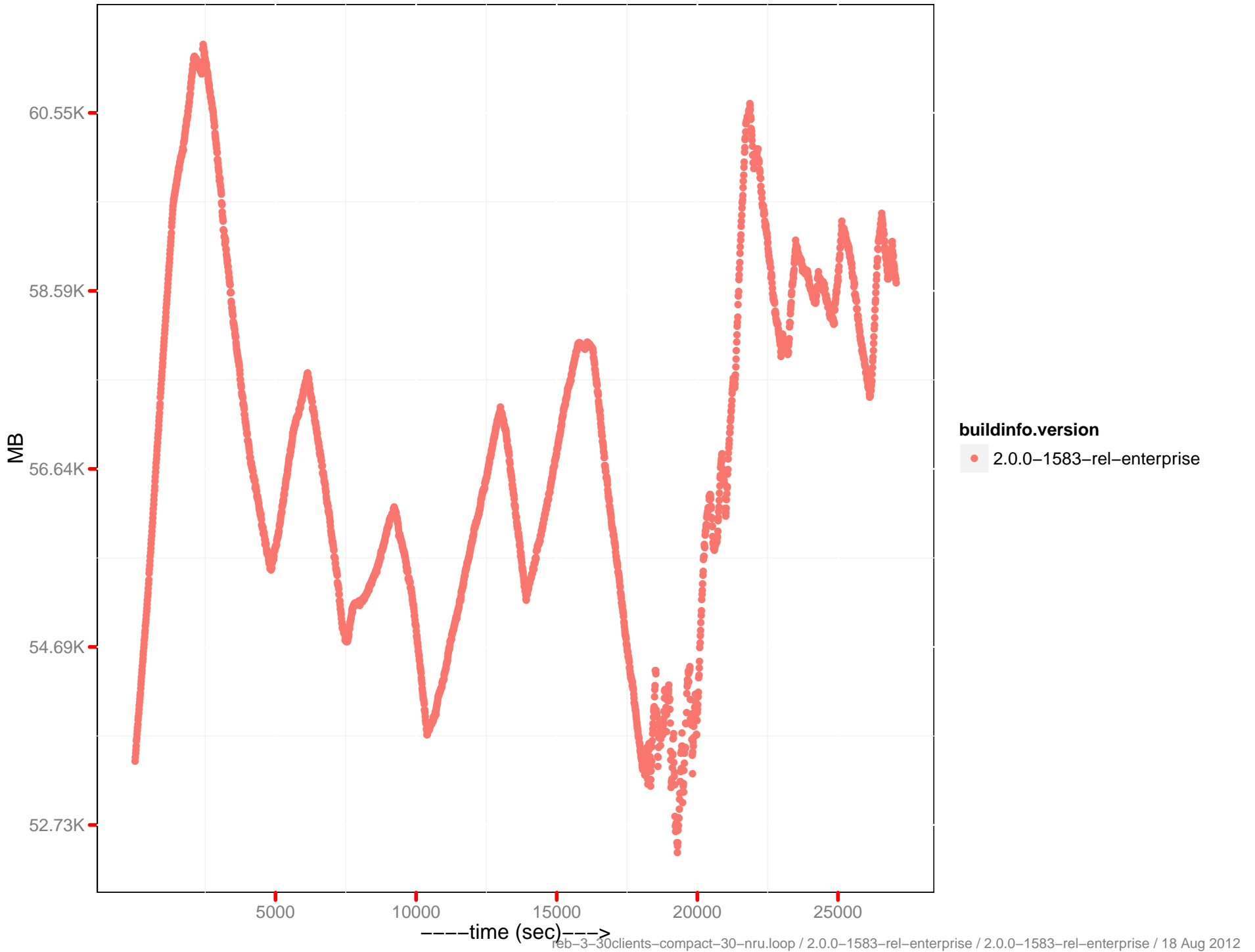
cpu_time_diff: memcached - 192.168.162.22



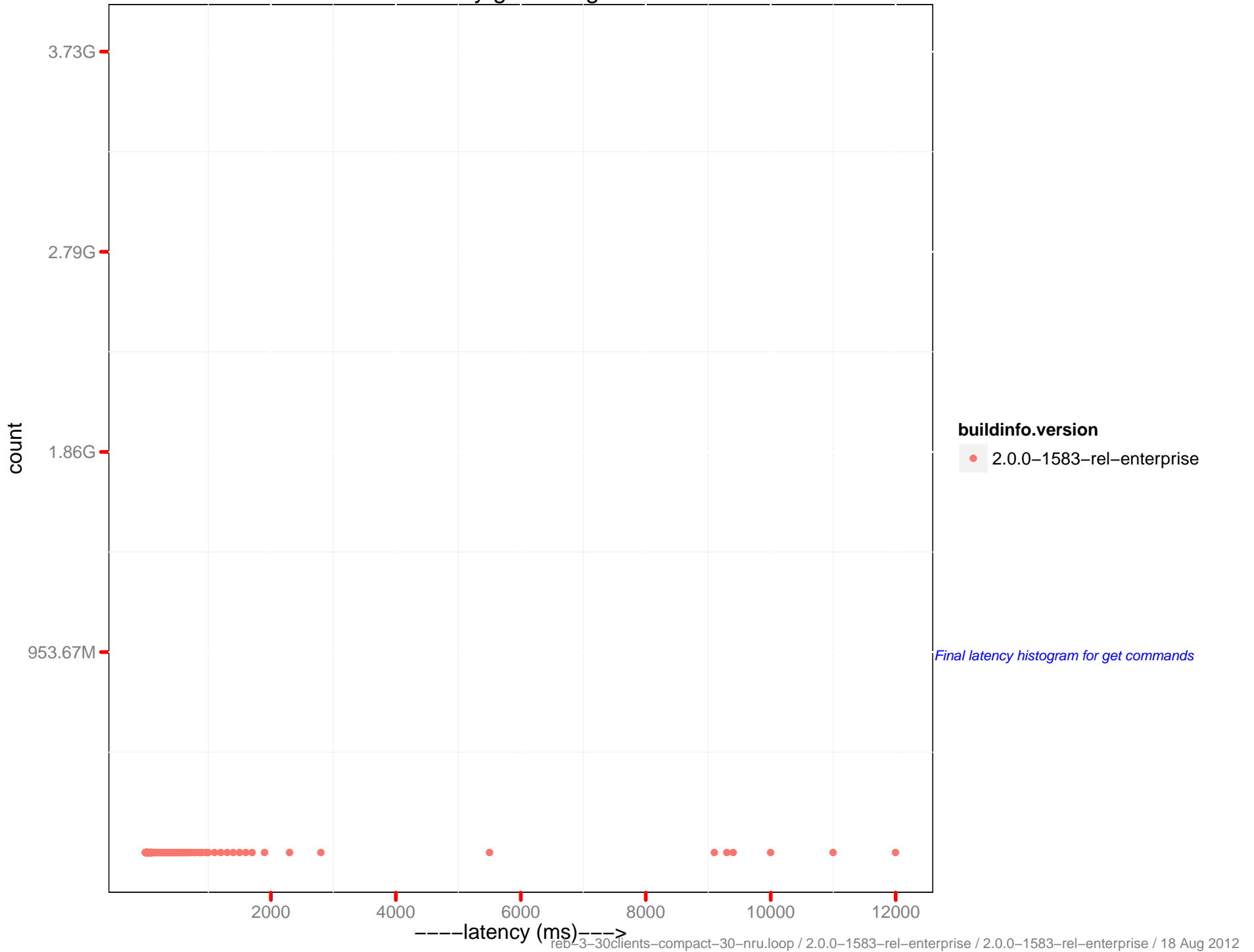
cpu_time_diff : beam.smp - 192.168.162.22



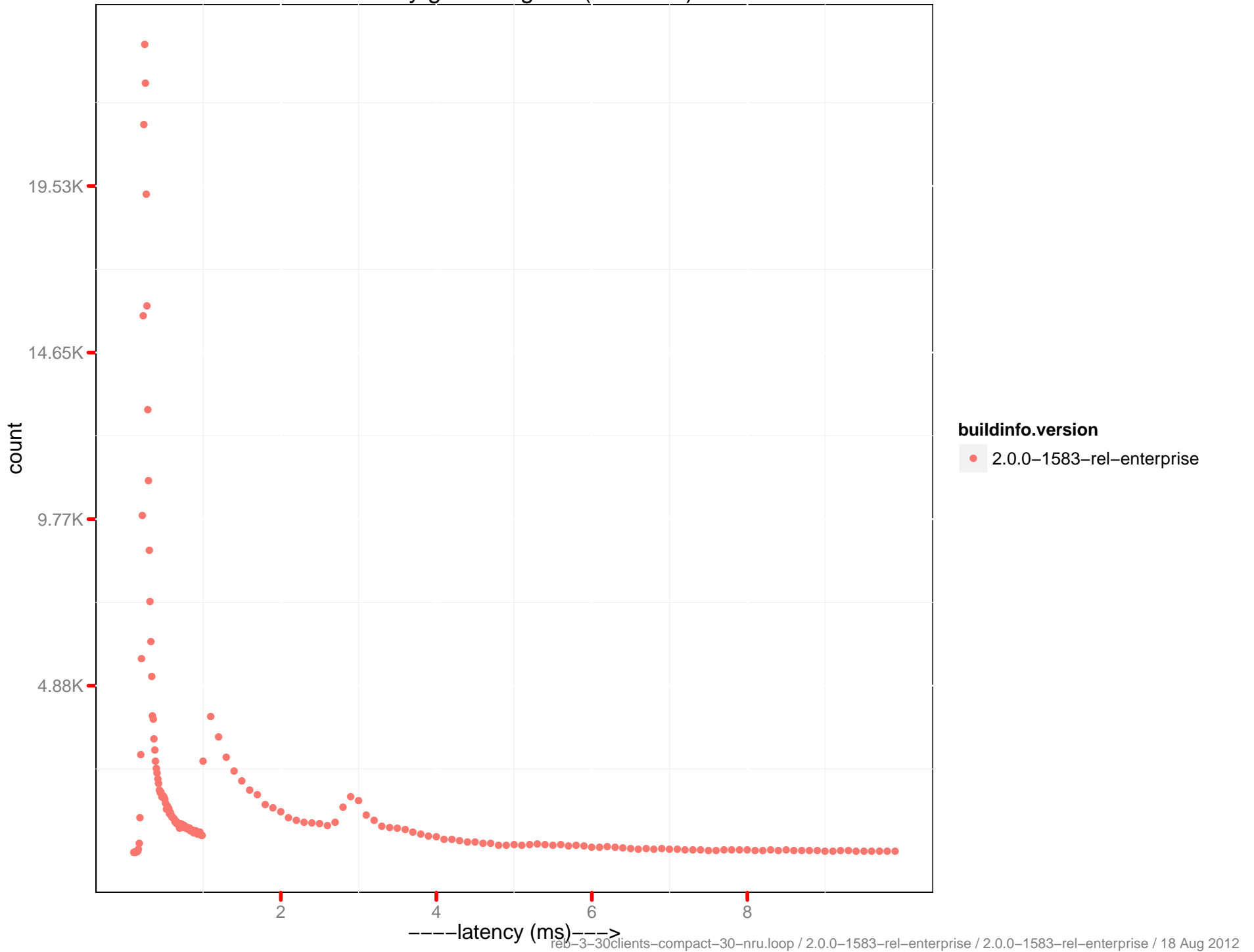
Data disk size



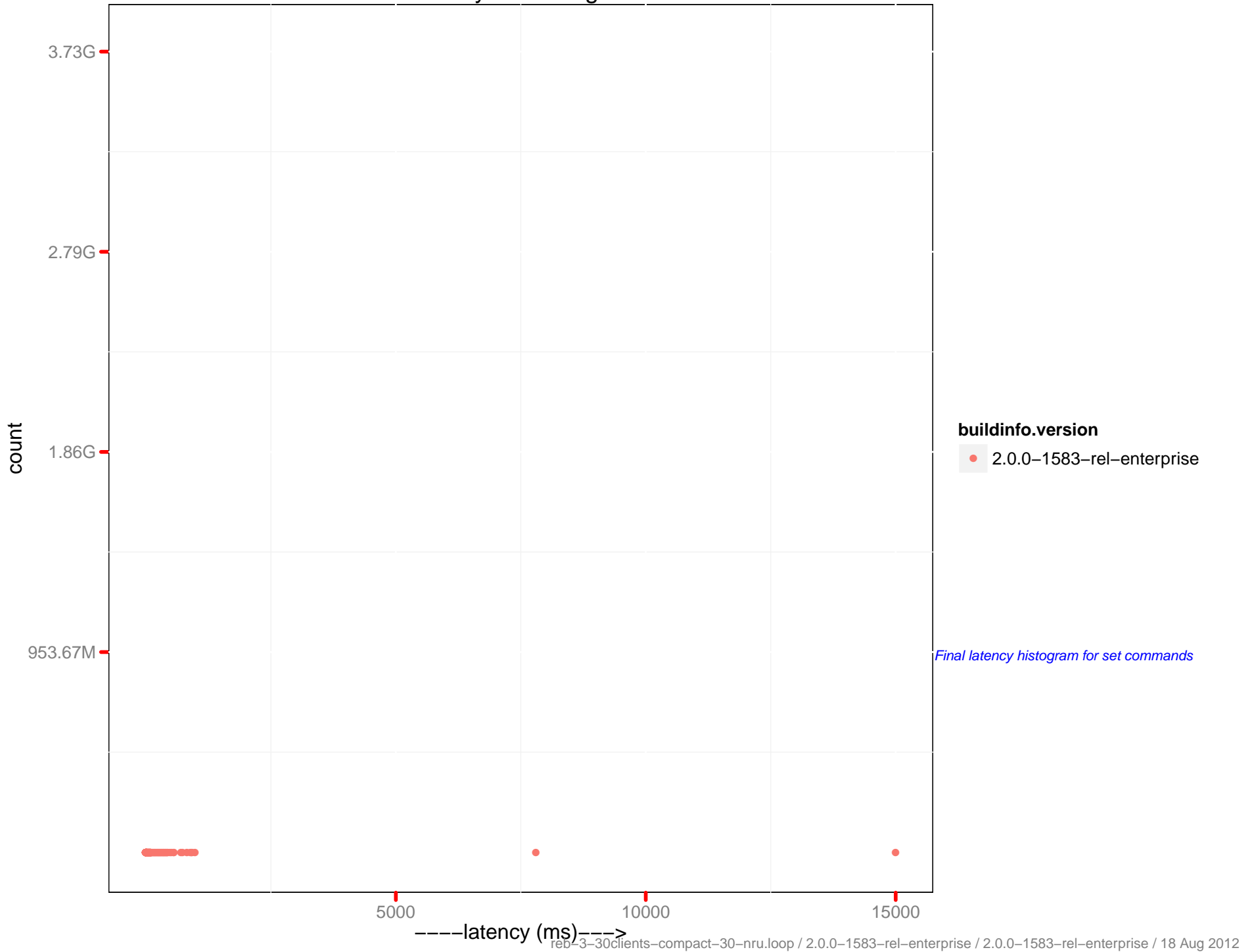
Latency get histogram



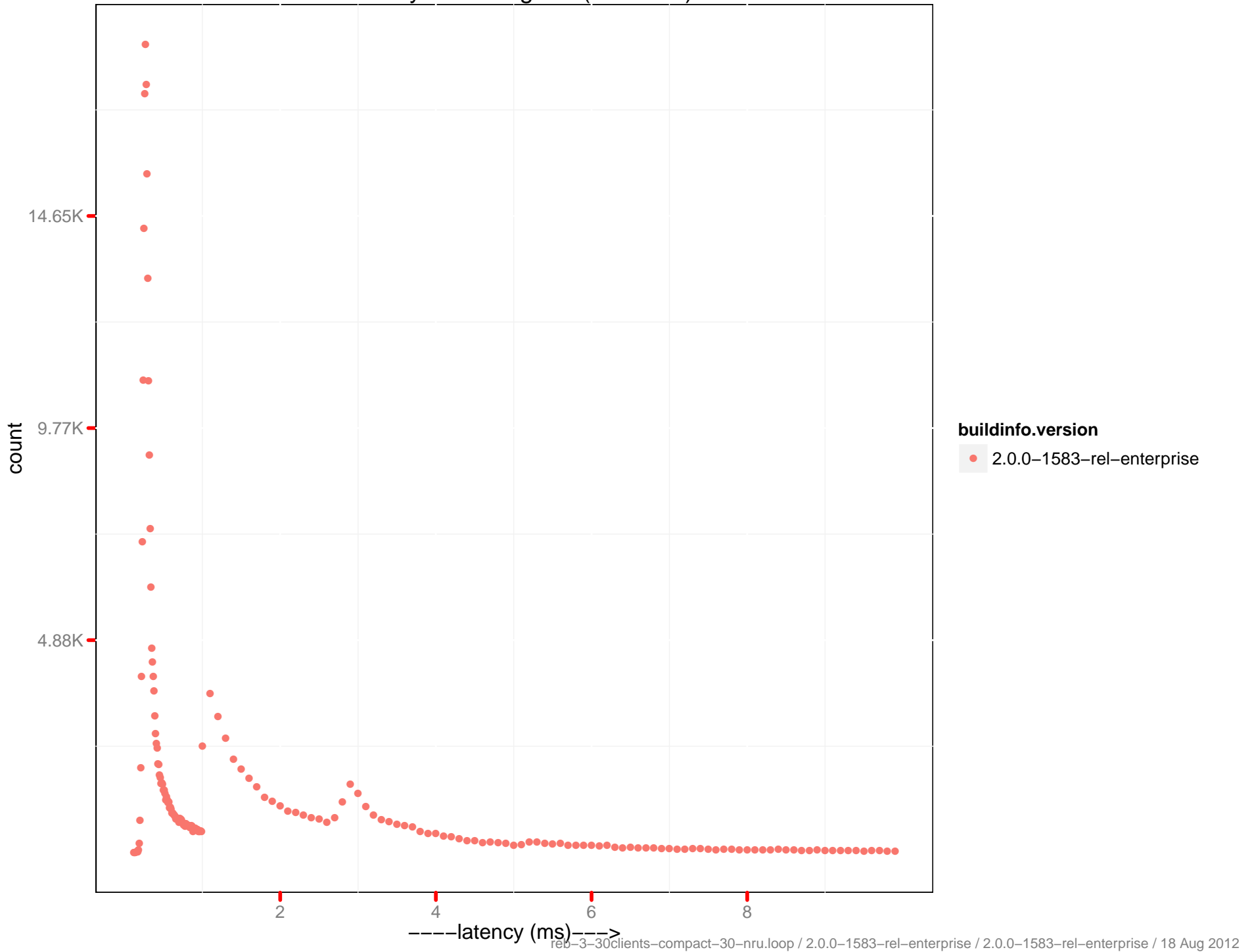
Latency get histogram (0-10 ms)



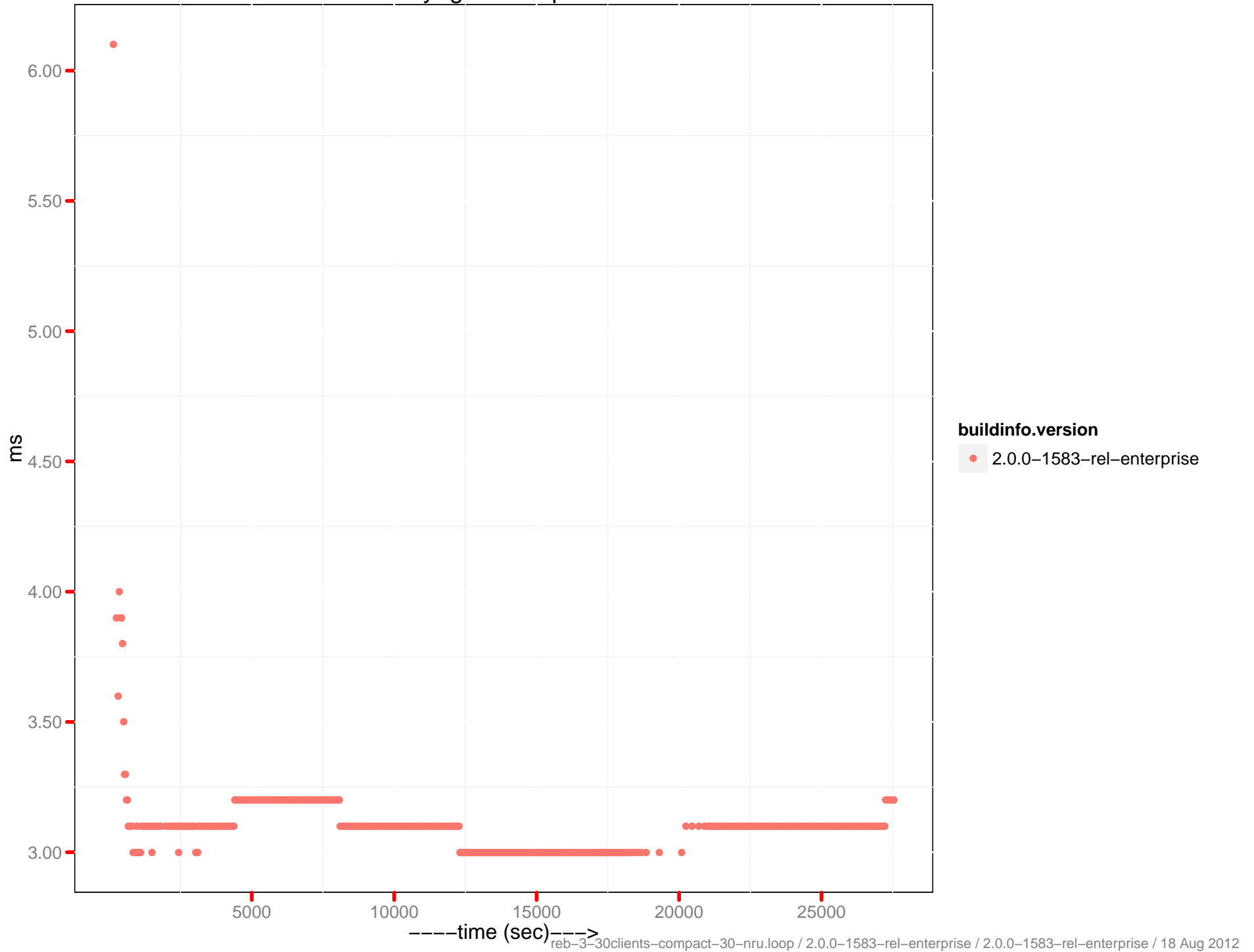
Latency set histogram



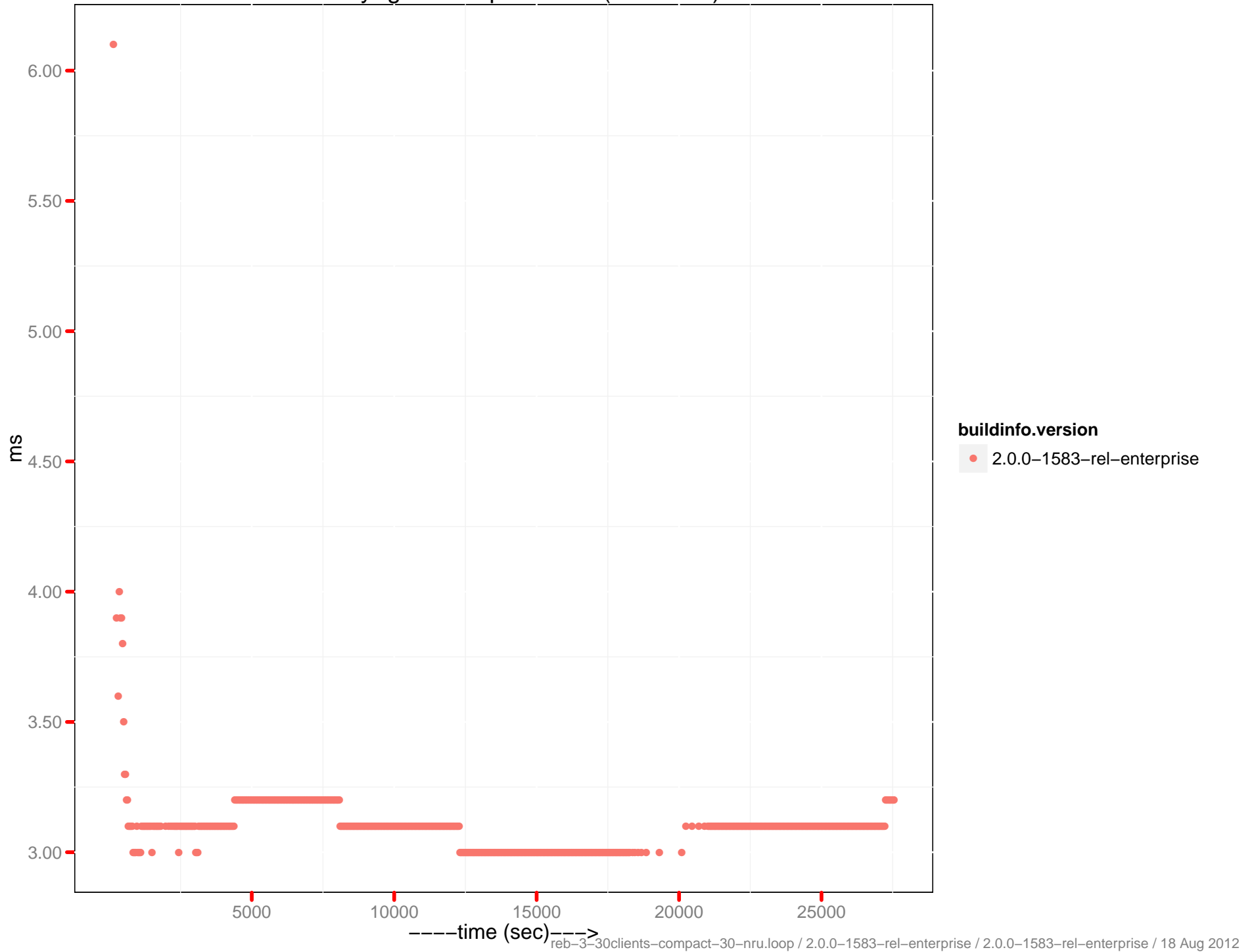
Latency set histogram (0–10 ms)



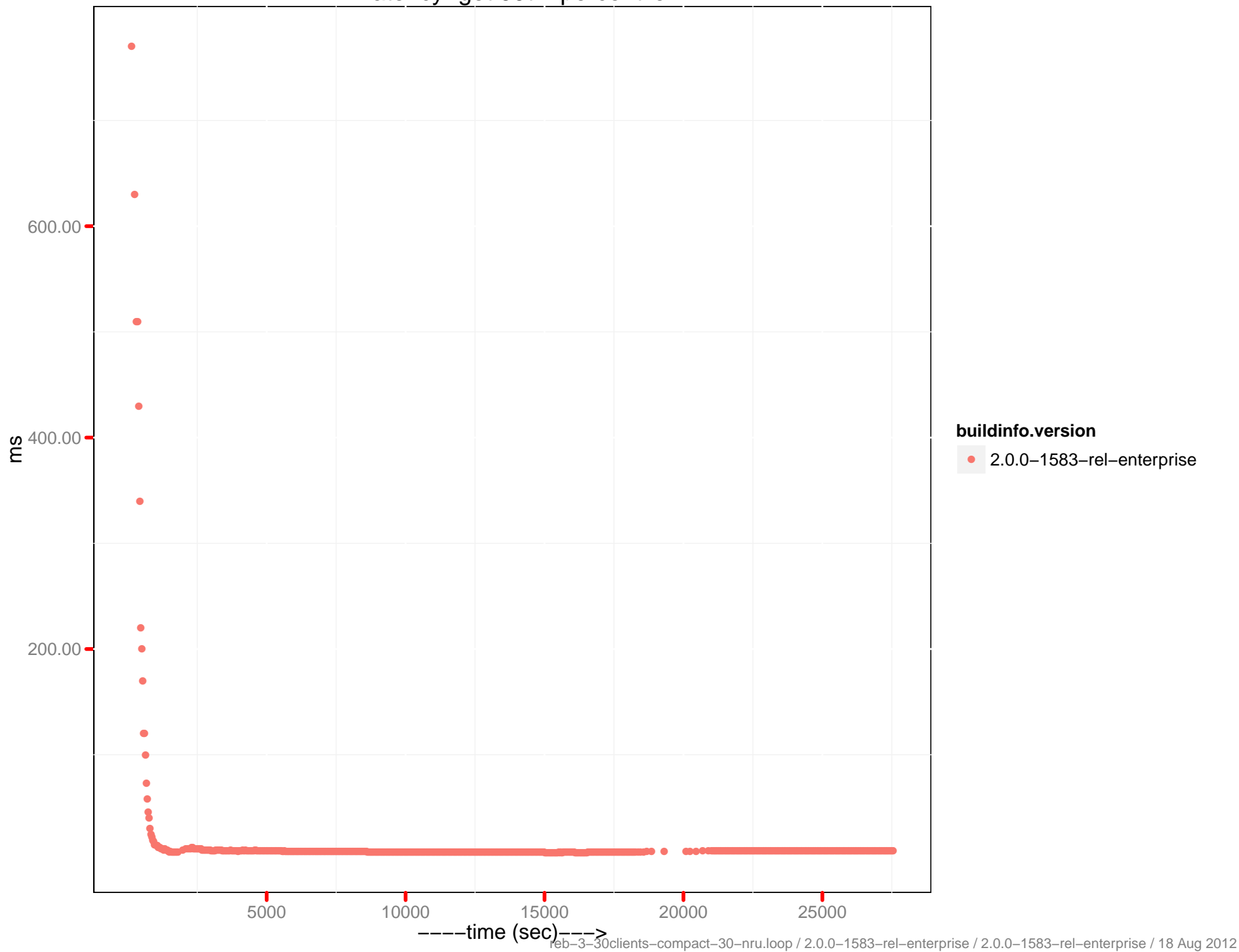
Latency-get 95th percentile



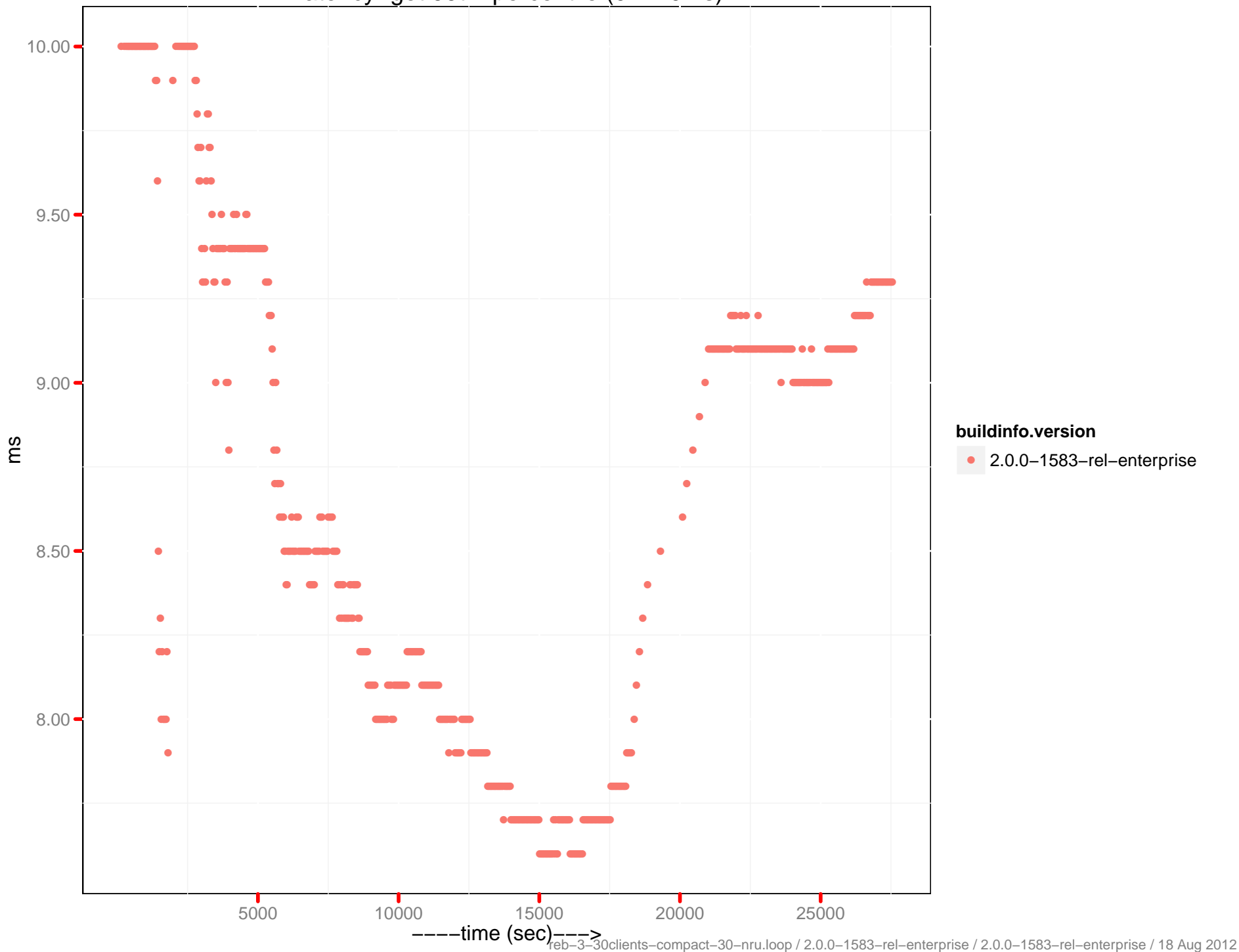
Latency-get 95th percentile (0 - 10ms)



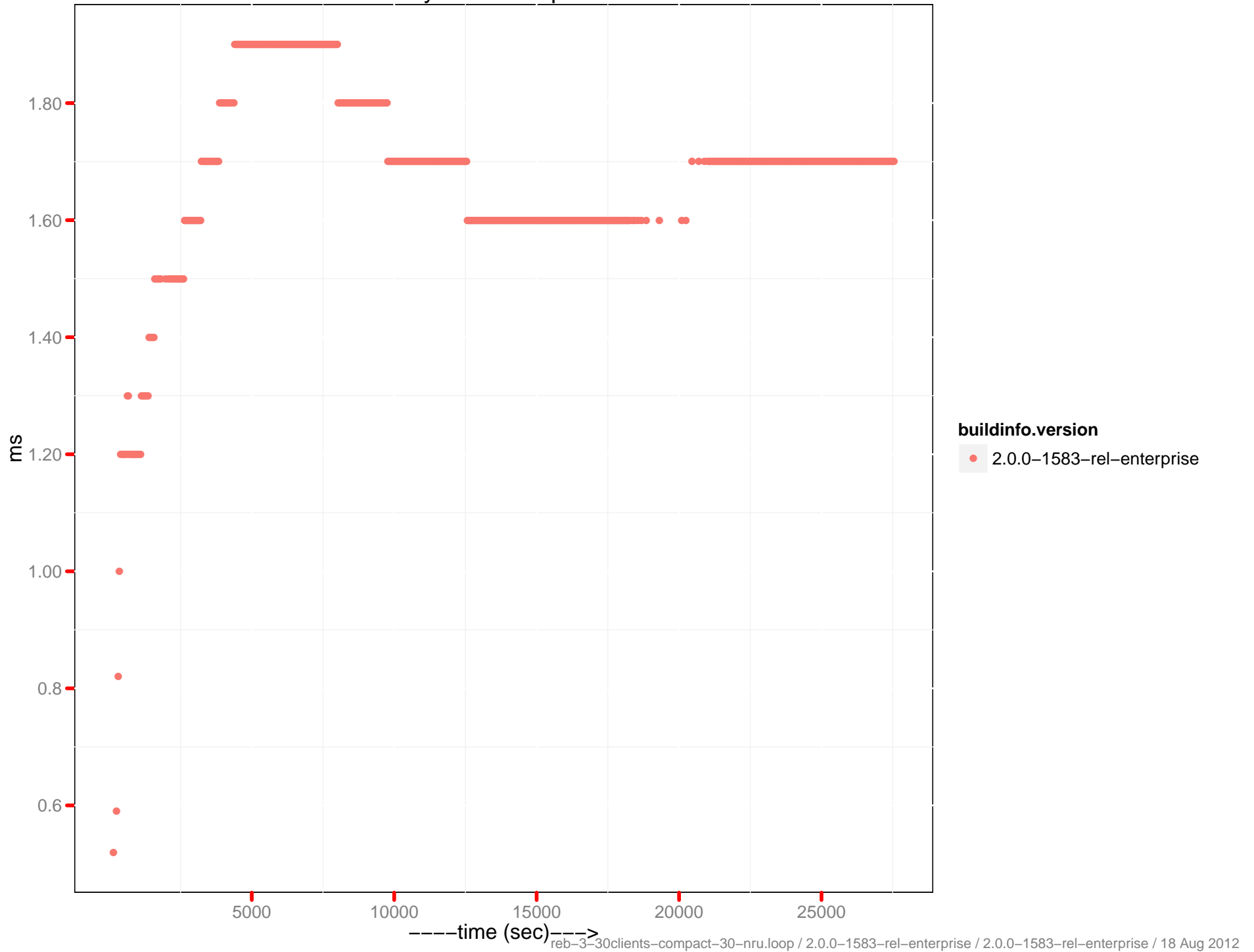
Latency-get 99th percentile



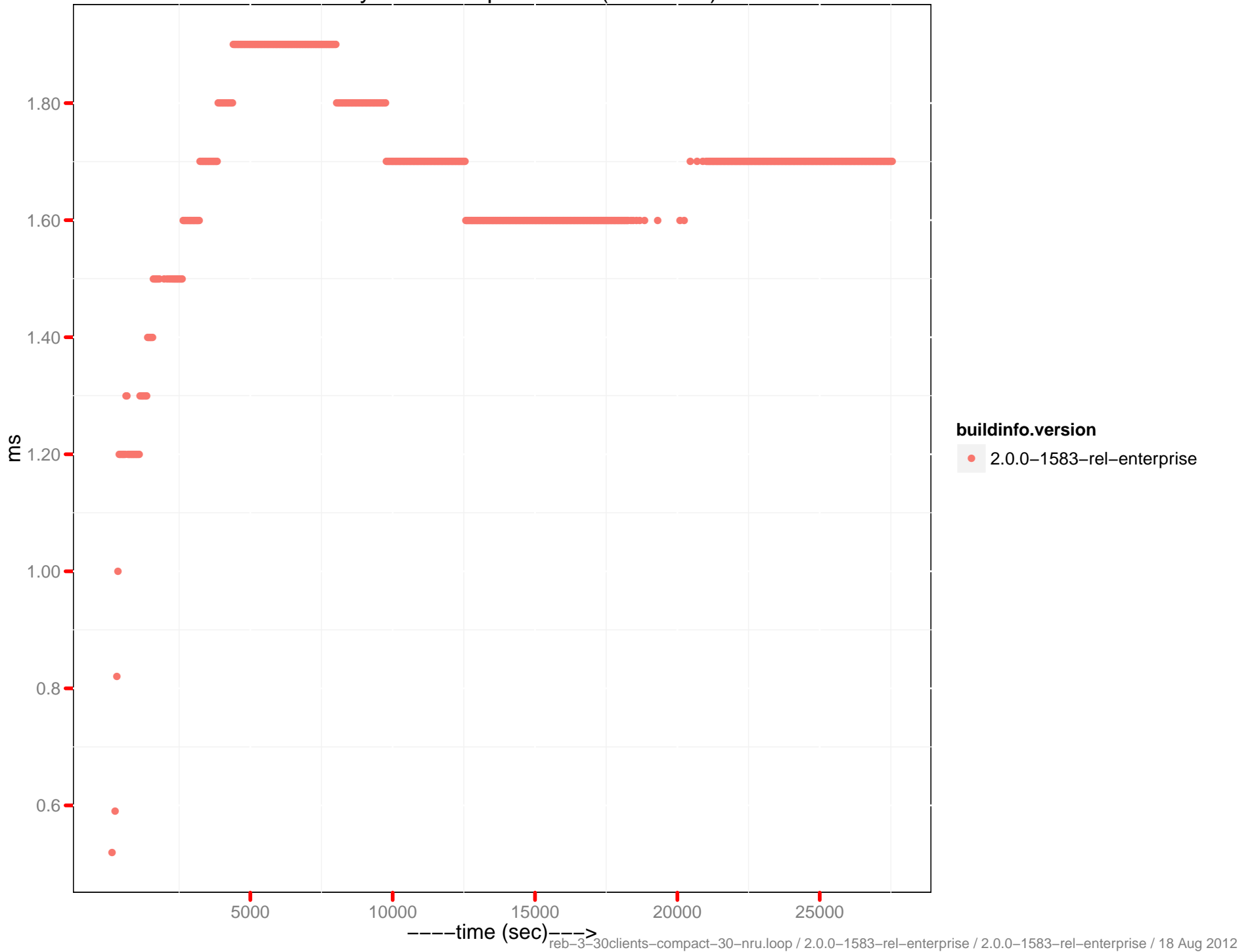
Latency-get 99th percentile (0 - 10ms)



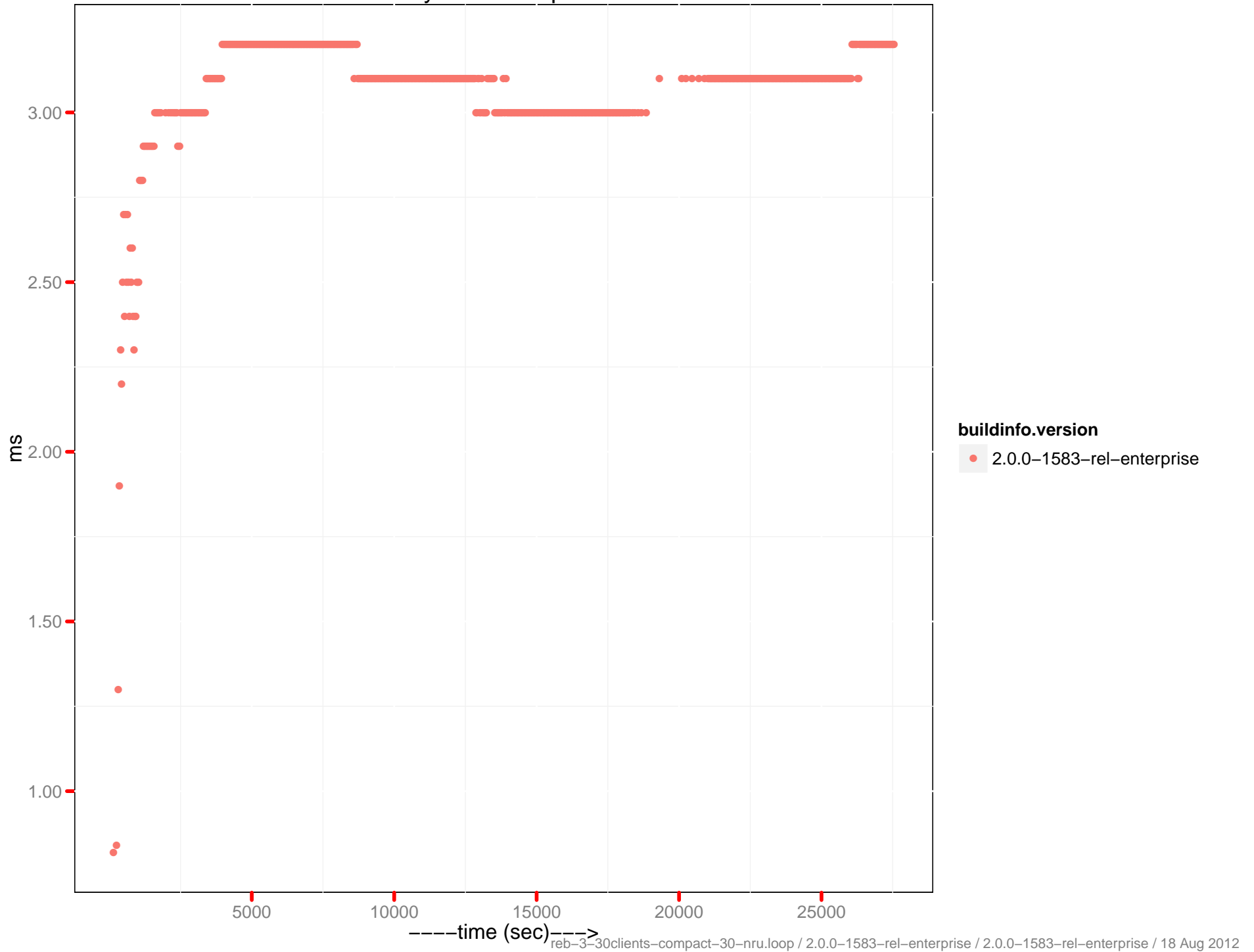
Latency-set 90th percentile



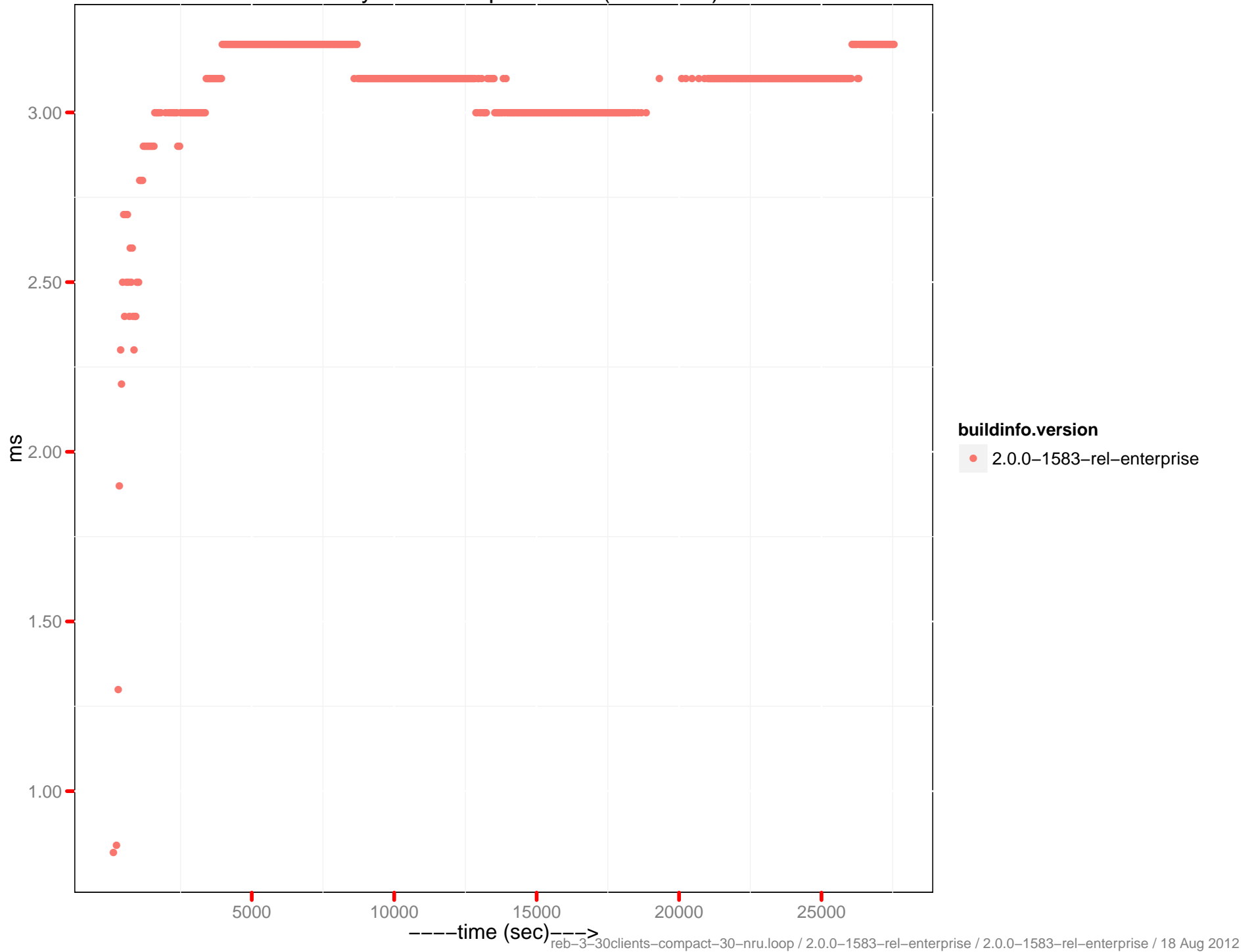
Latency-set 90th percentile (0 - 10ms)



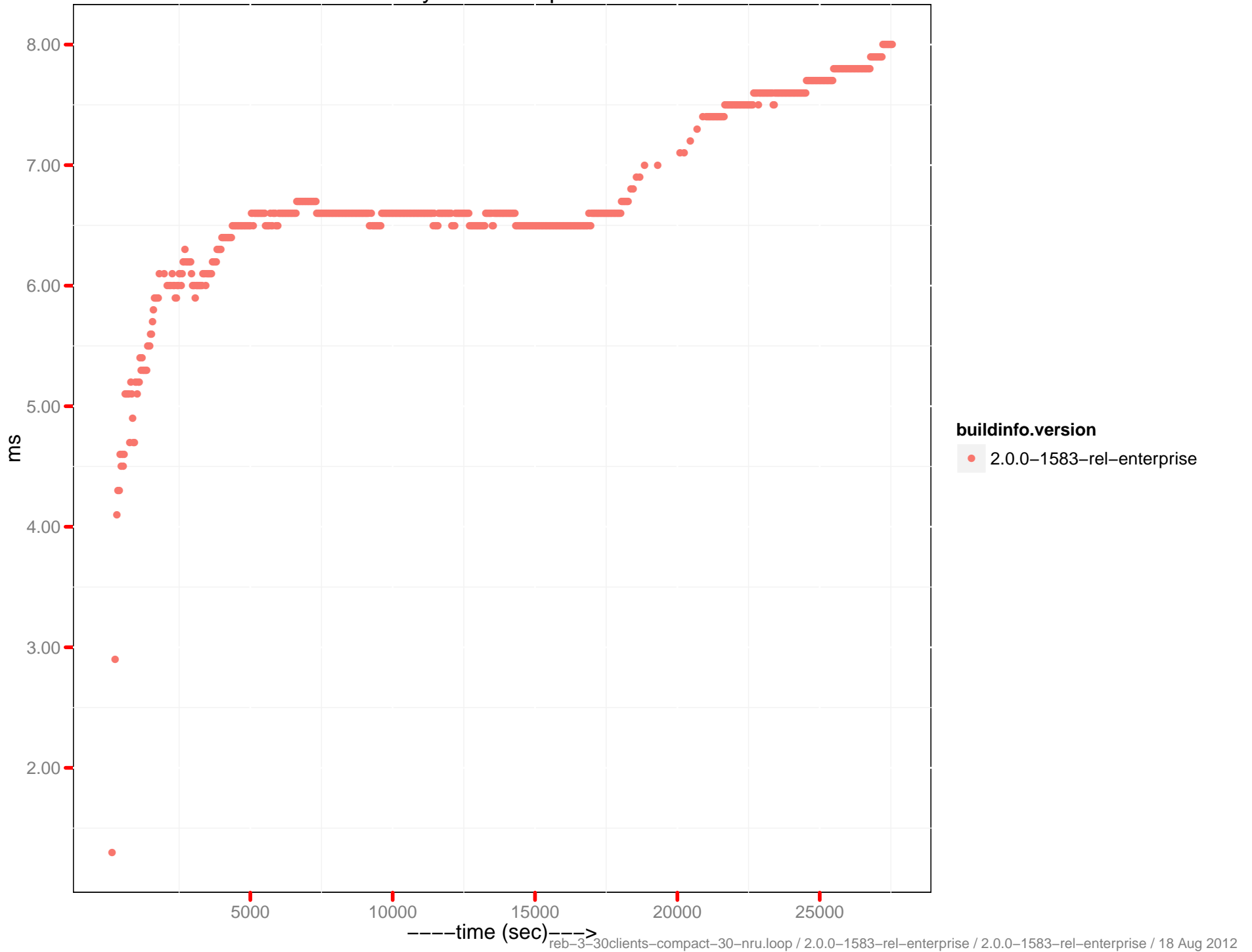
Latency-set 95th percentile



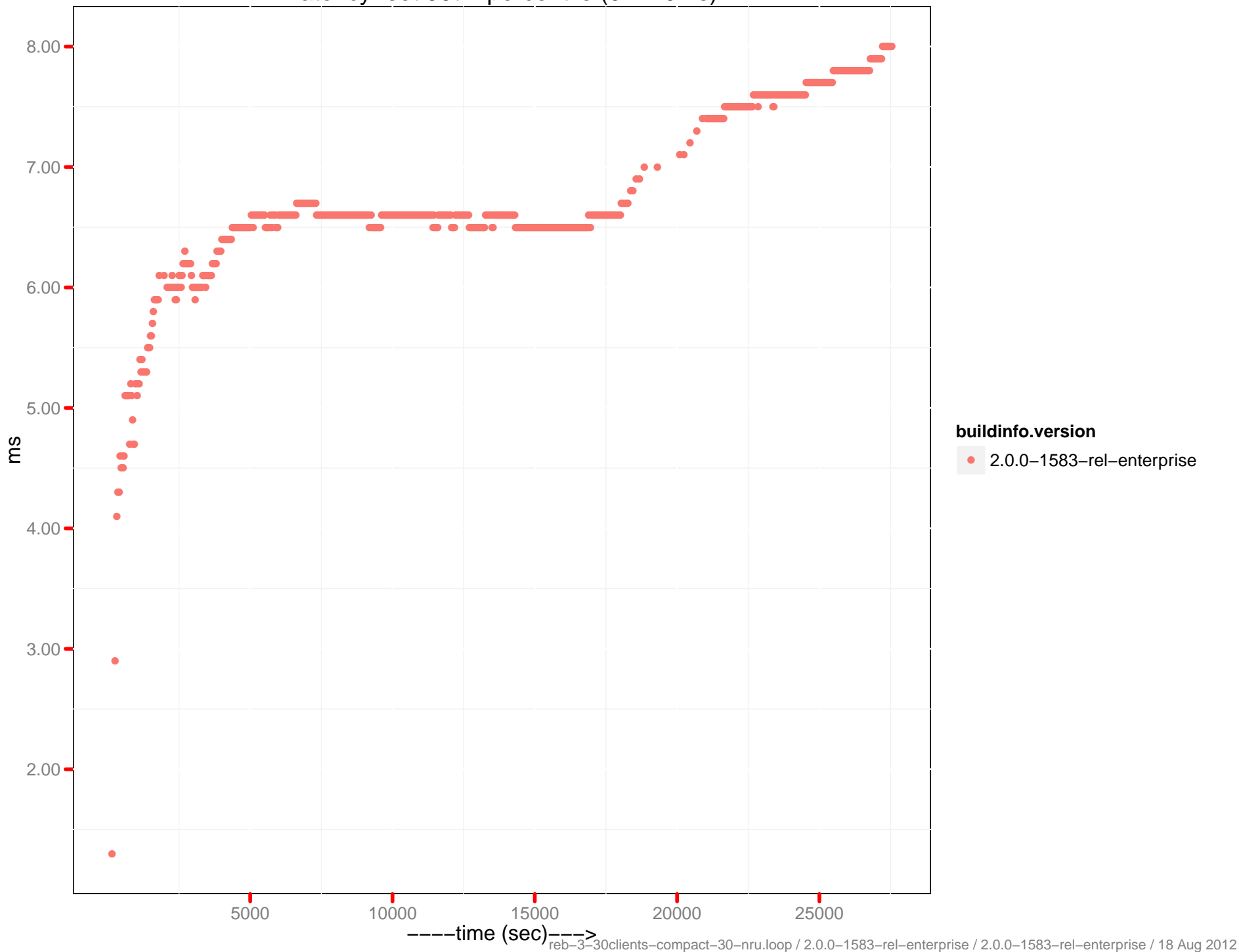
Latency-set 95th percentile (0 - 10ms)



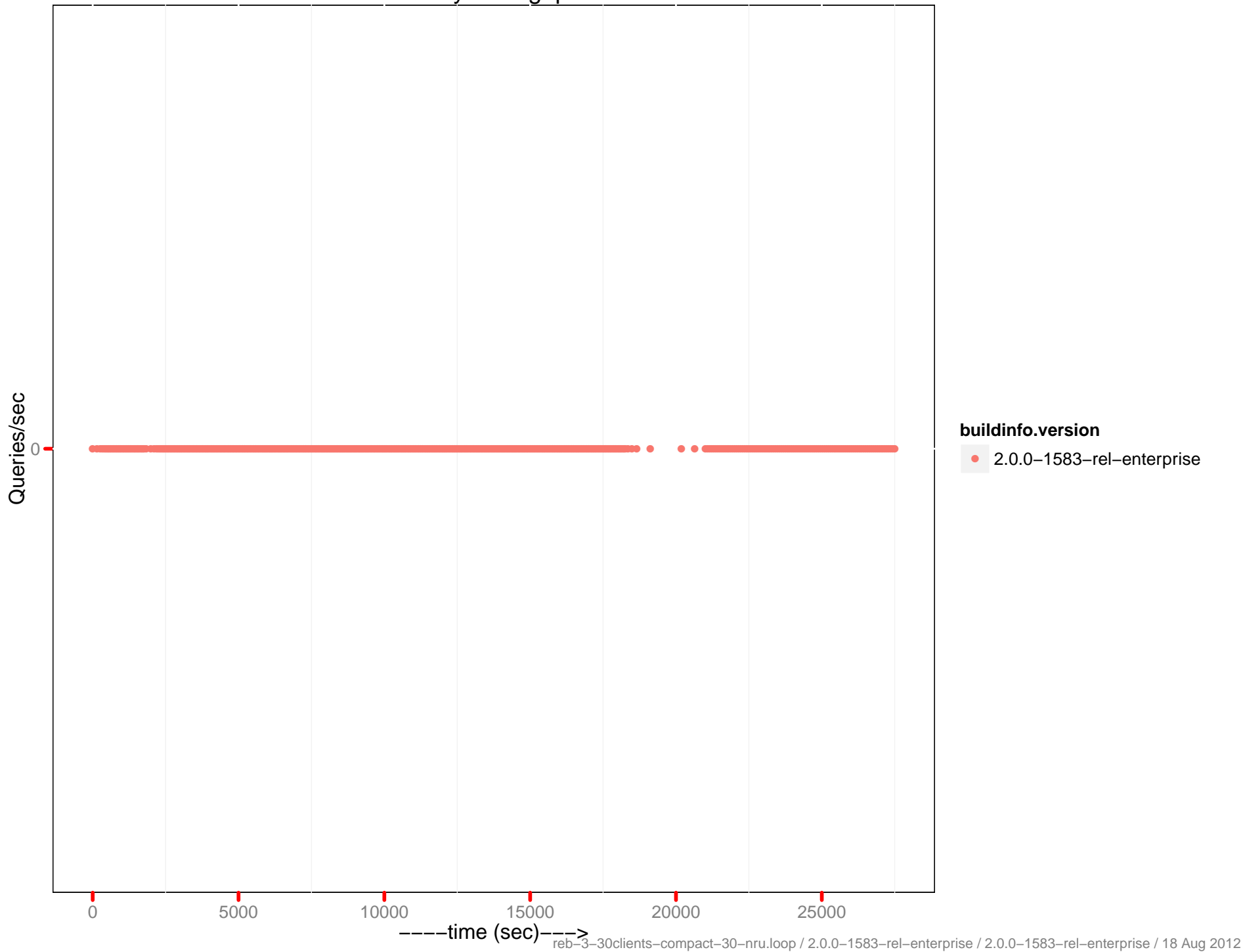
Latency-set 99th percentile



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



Query throughput



```
reb-3-30clients-compact-30-nru.conf
# rebalance mixed 10M load, 1M hot reload, 6M access creates
#
performance.epperf.EPerfClient.test_eperf_rebalance

params:

# general
batch=50
kind=nonjson
mem_quota=20000
db_compaction=30

# load phase
hot_init_items=1000000
items=10000000

# 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=0.99
ratio_hot_sets=0.99
ratio_expirations=0.03
max_creates=9000000

# rebalance
nru_task=1
nru_reb_delay=3600
reb_max_retries=5
num_nodes_after=3

# control (defaults: pytests/performance/perf_defaults.py)
load_wait_until_drained=1
loop_wait_until_drained=0
mcsoda_heartbeat=3
mcsoda_max_ops_sec=300
tear_down=1
tear_down_proxy=1
tear_down_bucket=0
tear_down_cluster=1
tear_down_on_setup=0
```

hummer-dedicated.ini

[global]

username:root

password:couchbase

port:8091

data_path:/data

[servers]

1:192.168.162.20

2:192.168.162.21

3:192.168.162.22

4:192.168.162.23

[clients]

1:192.168.162.24

2:192.168.162.25

3:192.168.162.26

4:192.168.162.27

5:192.168.162.28

6:192.168.162.29

[membase]

rest_username:Administrator

rest_password:password

[dashboard]

1:dashboard.hq.couchbase.com:80