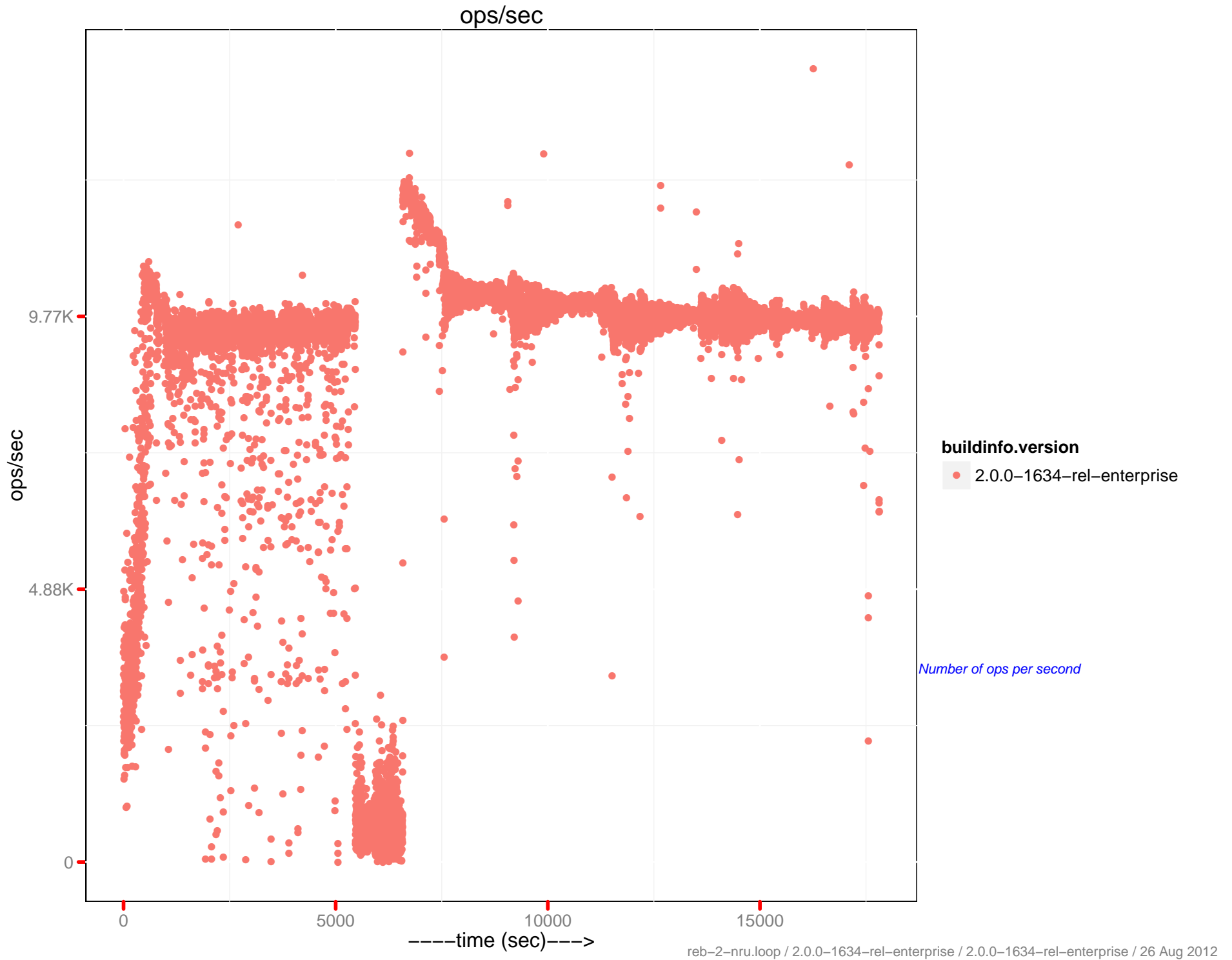
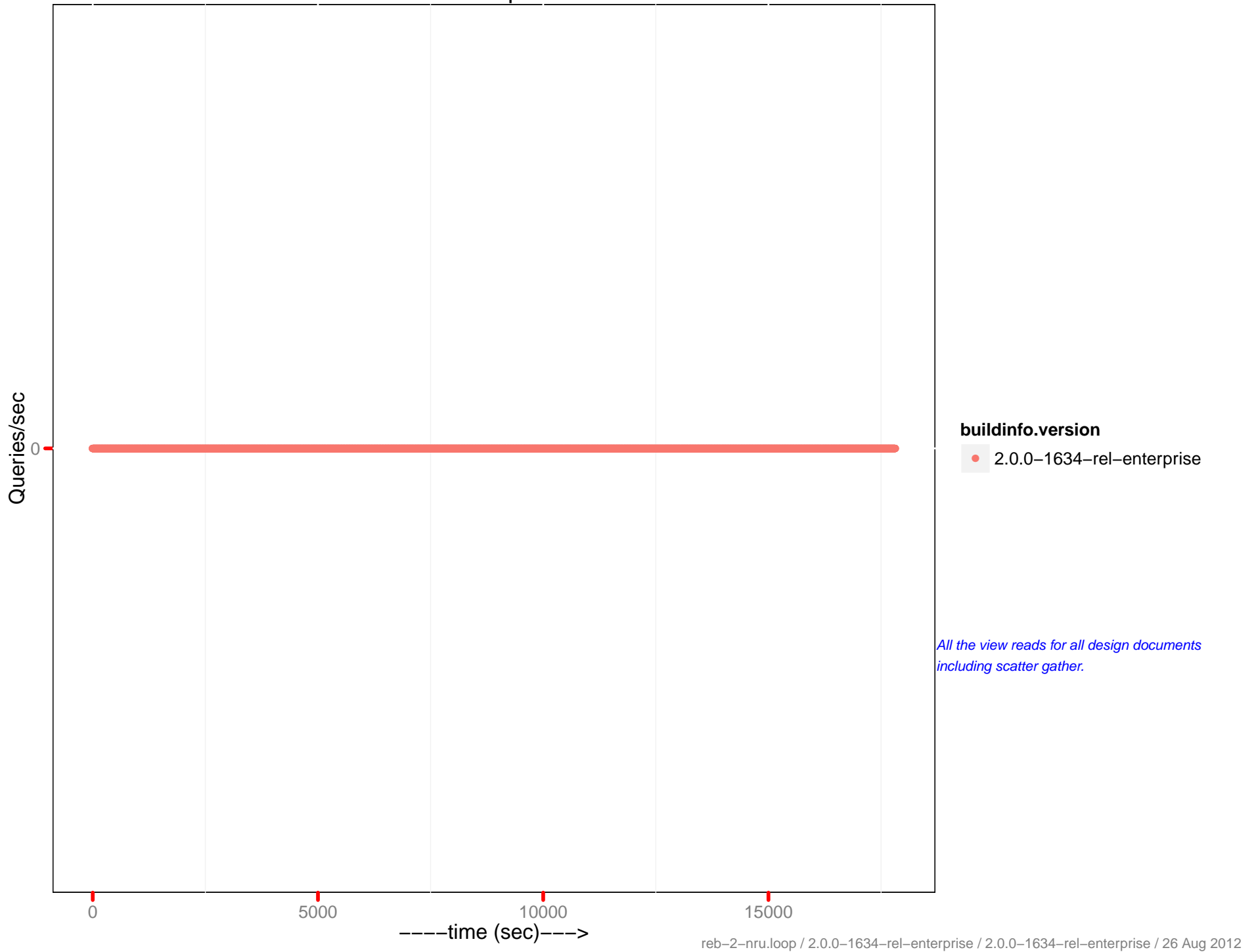


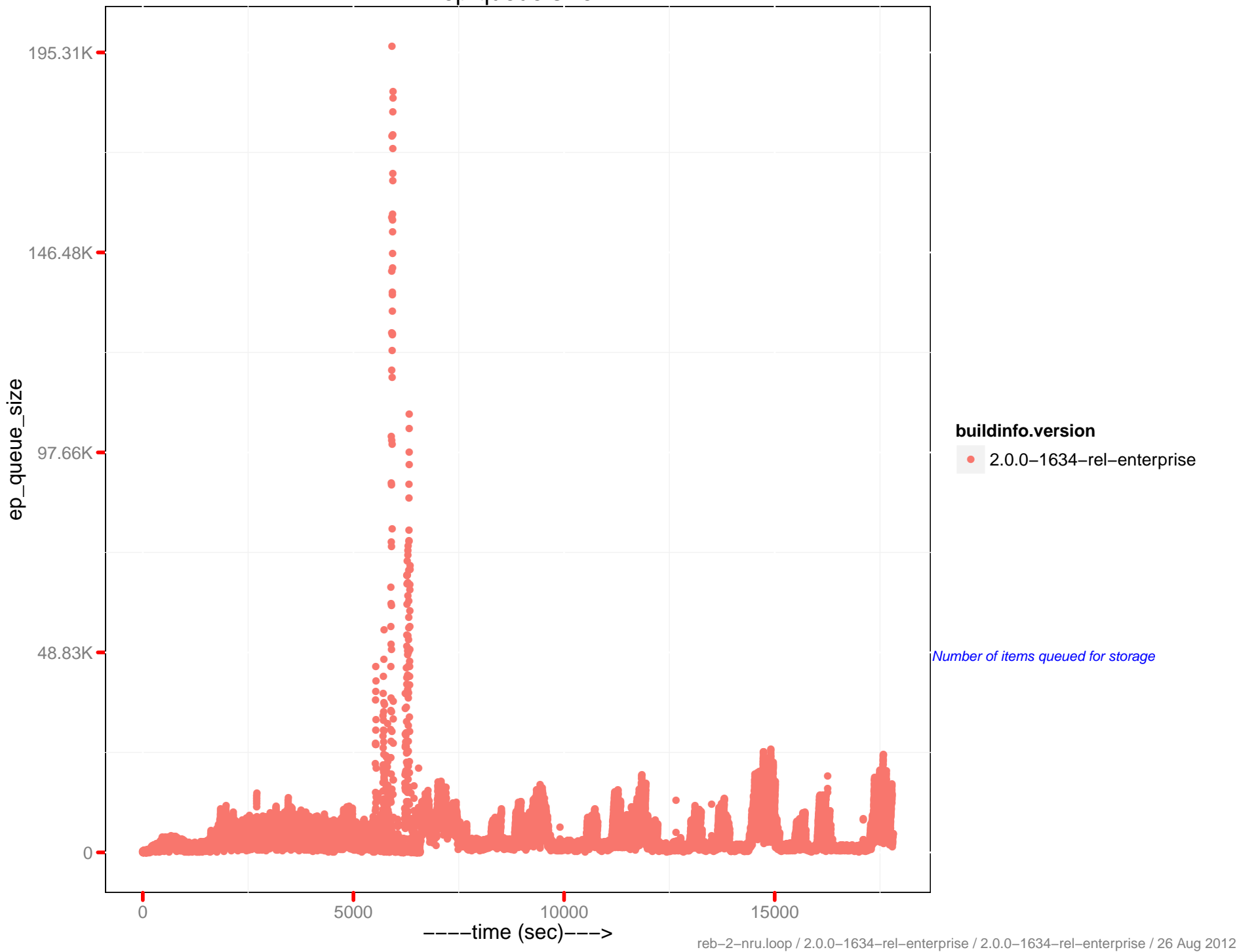
	2.0.0 – 1634	2.0.0 – 1634
<i>Runtime (in hr)</i>	4.96	NA
<i>Avg. Drain Rate</i>	2.00K	NANA
<i>Peak Disk (GB)</i>	63.73	NA
<i>Peak Memory (GB)</i>	17.05	NA
<i>Avg. OPS</i>	9.05K	NANA
<i>Avg. mem memcached (GB)</i>	16.51	NA
<i>Avg. mem beam.smp (MB)</i>	474.5	NA
<i>Avg. CPU rate (%)</i>	15.22	NA
<i>Latency-get (90th) (ms)</i>	1.05	NA
<i>Latency-get (95th) (ms)</i>	1.97	NA
<i>Latency-get (99th) (ms)</i>	21.55	NA
<i>Latency-set (90th) (ms)</i>	1.21	NA
<i>Latency-set (95th) (ms)</i>	2.07	NA
<i>Latency-set (99th) (ms)</i>	6.03	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>	1129.74	NA
<i>Testrunner Version</i>	5fecf69	NA



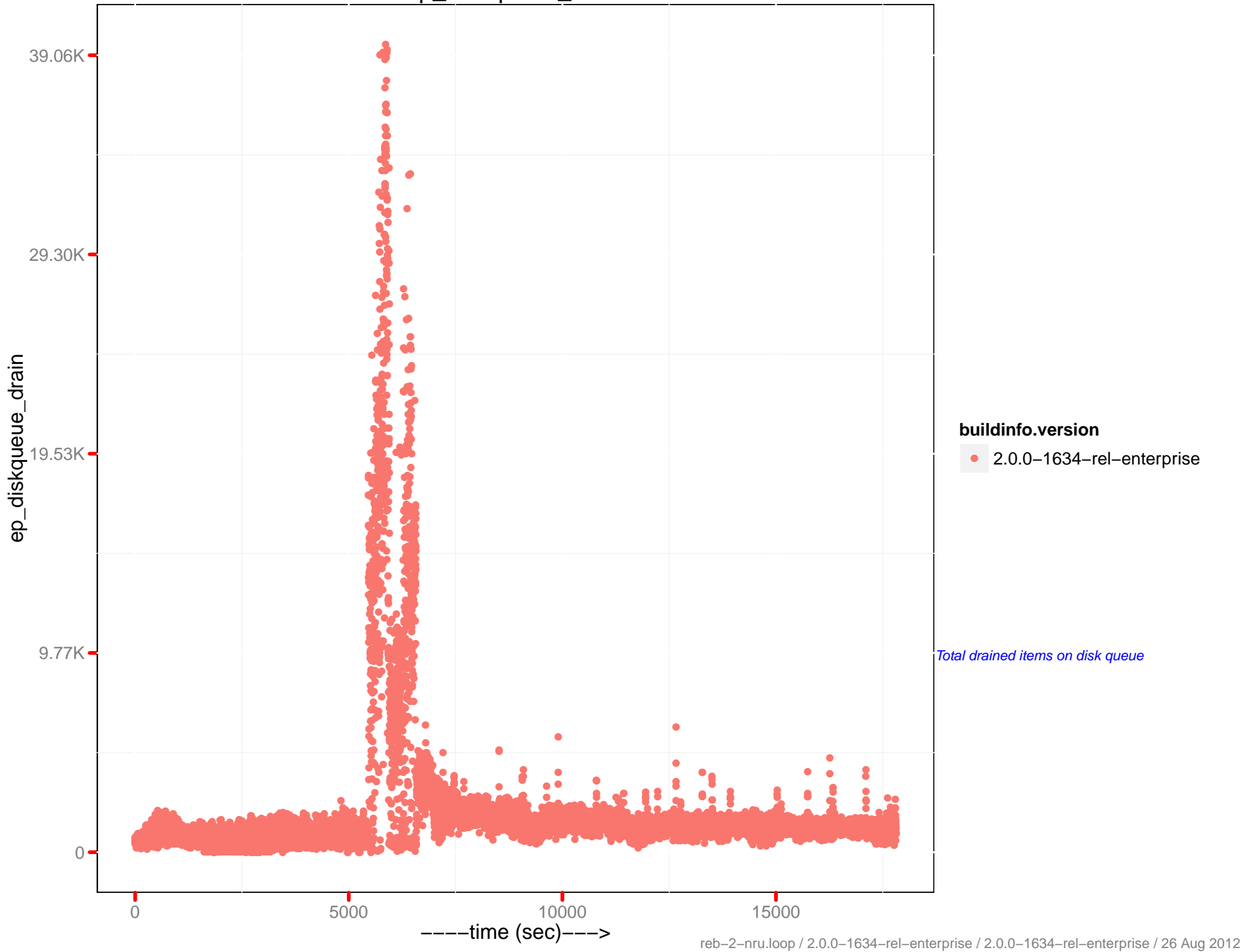
# View read per sec.



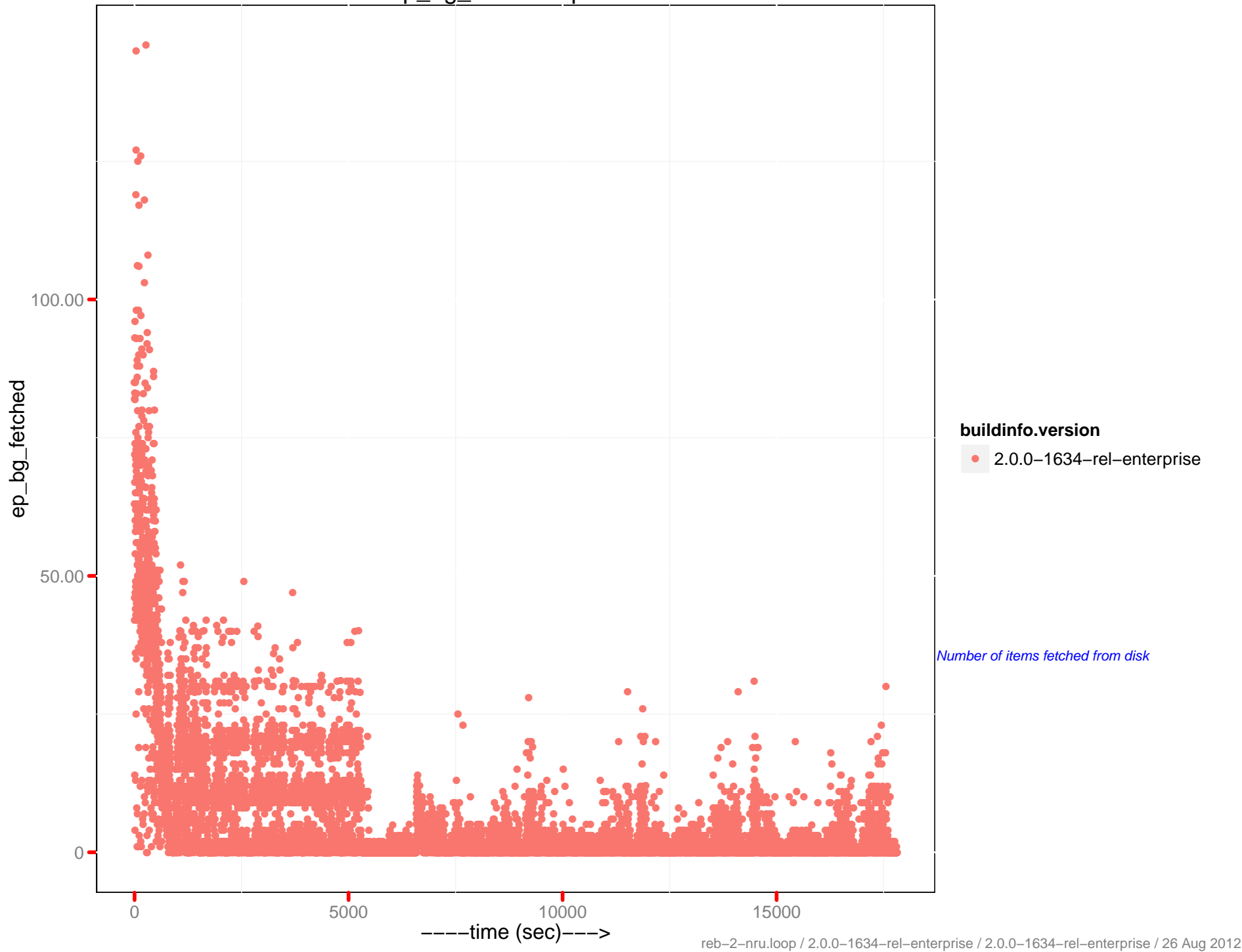
# ep queue size



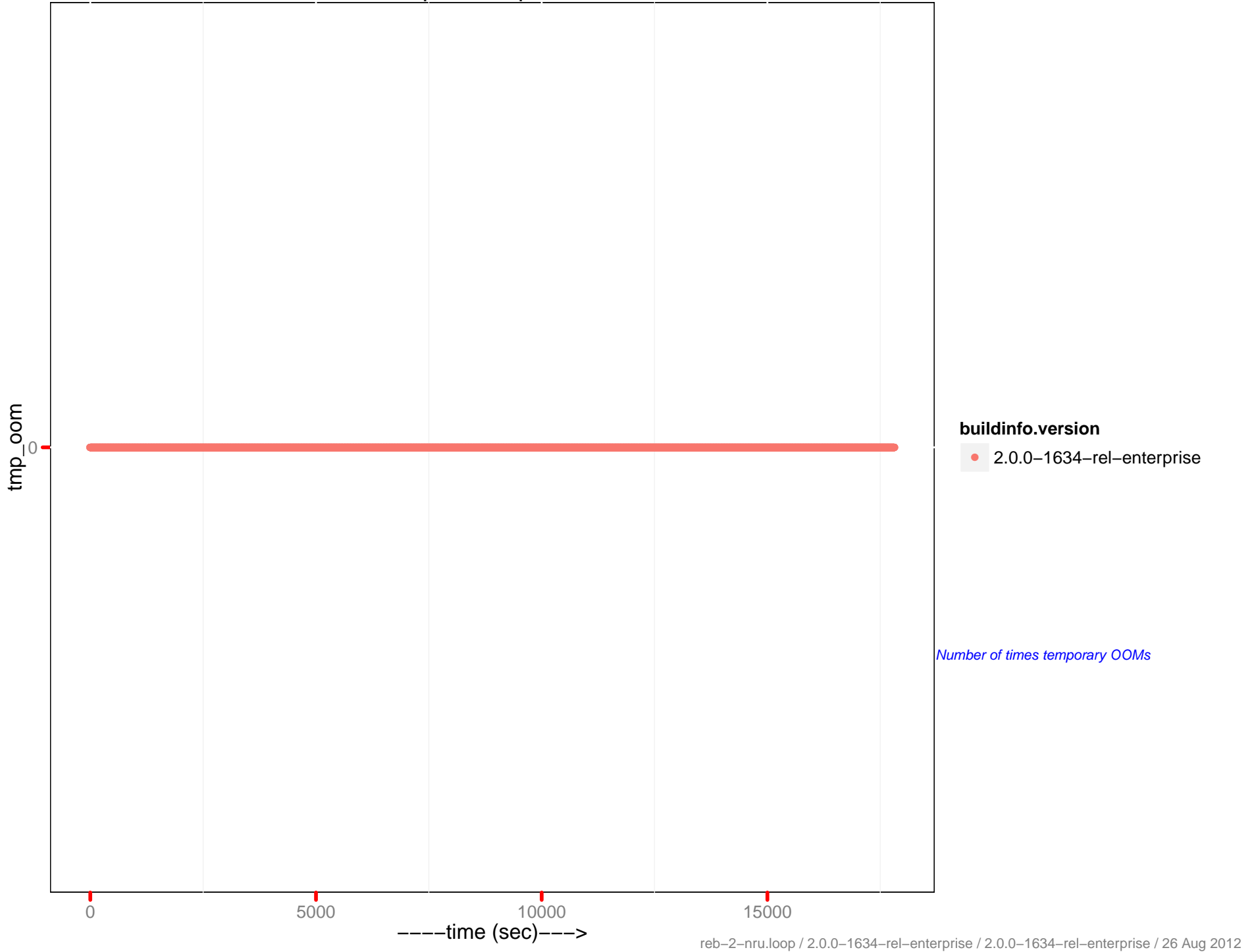
# ep\_diskqueue\_drain



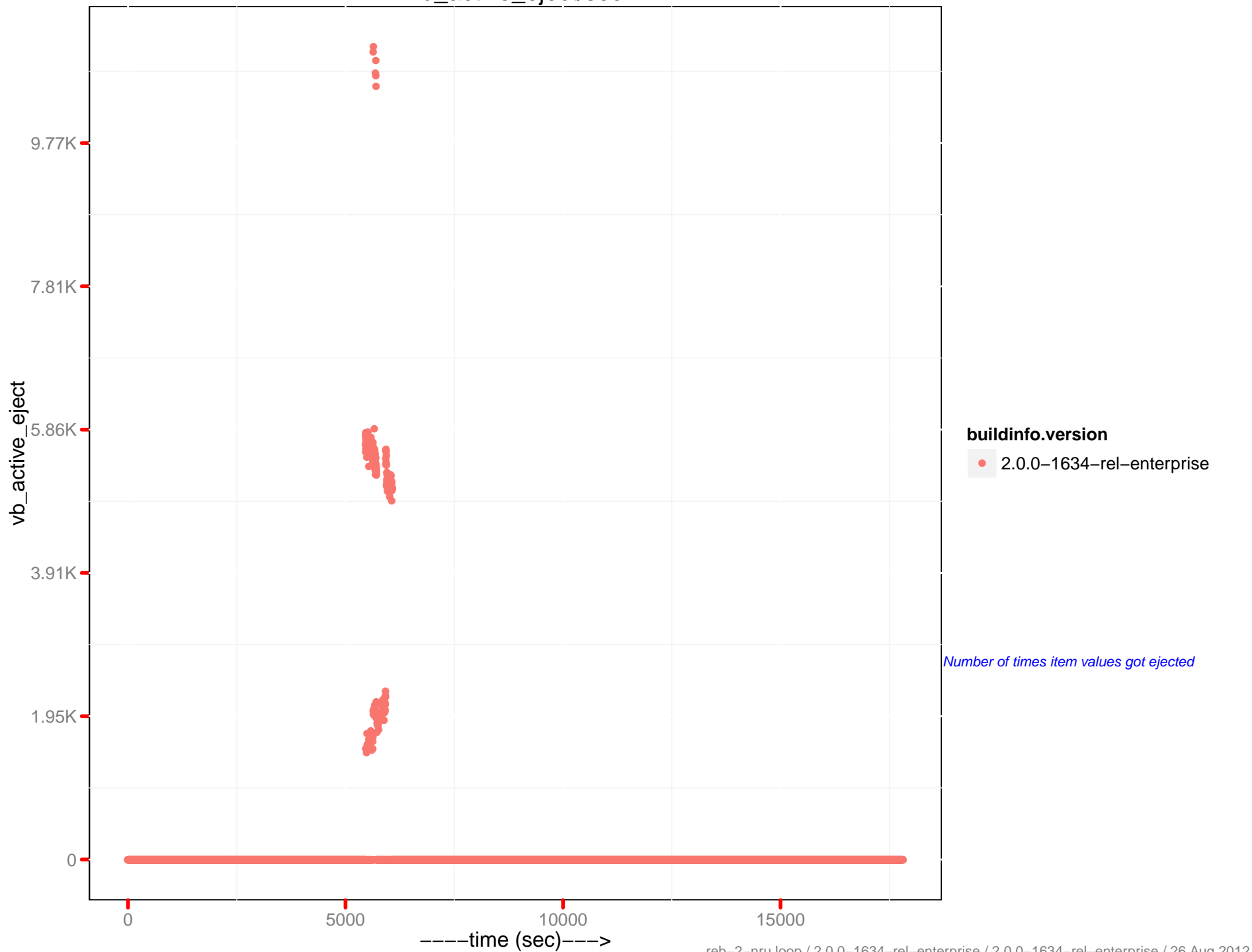
# ep\_bg\_fetched ops/sec



# tmp\_oom ops/sec

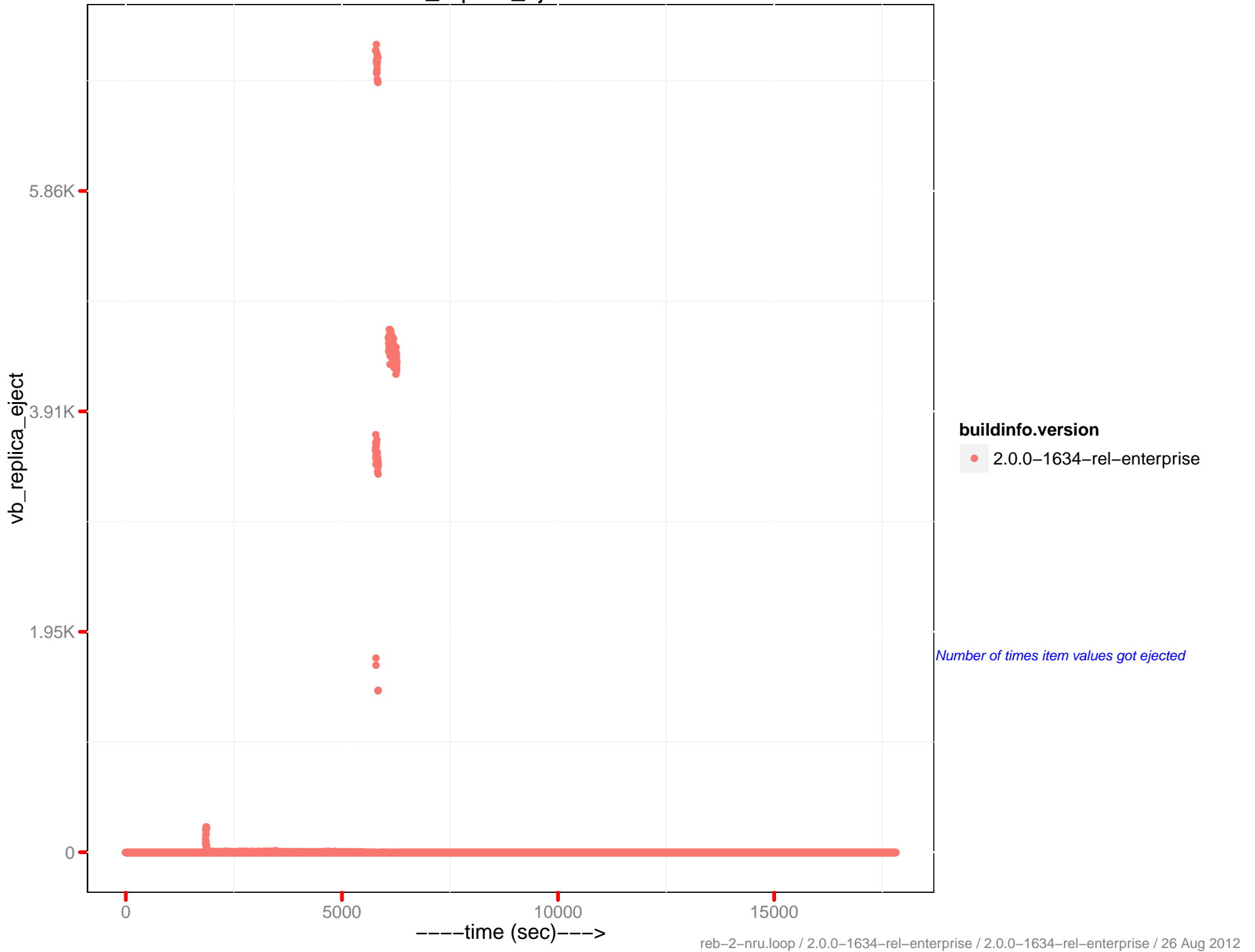


# vb\_active\_eject/sec

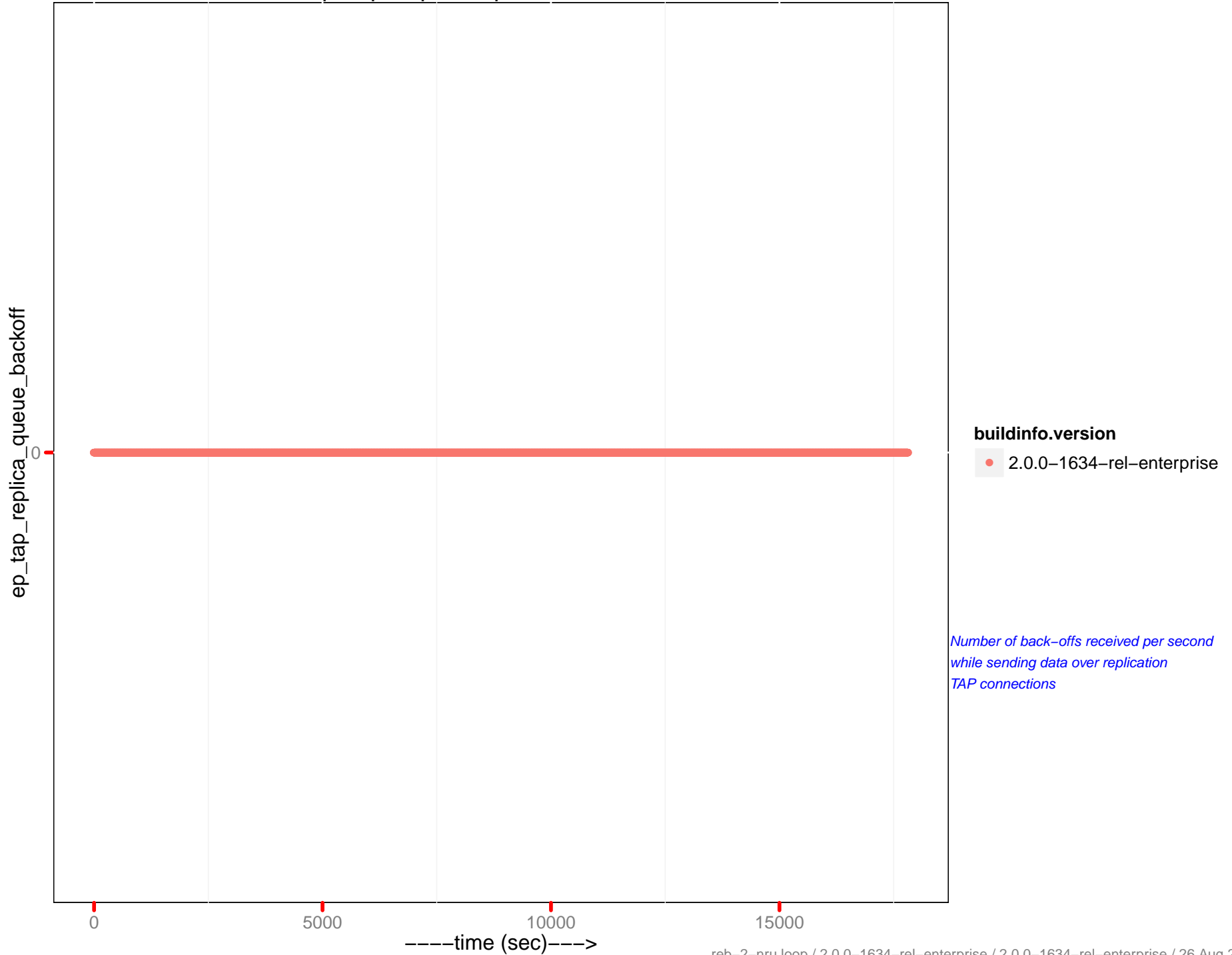




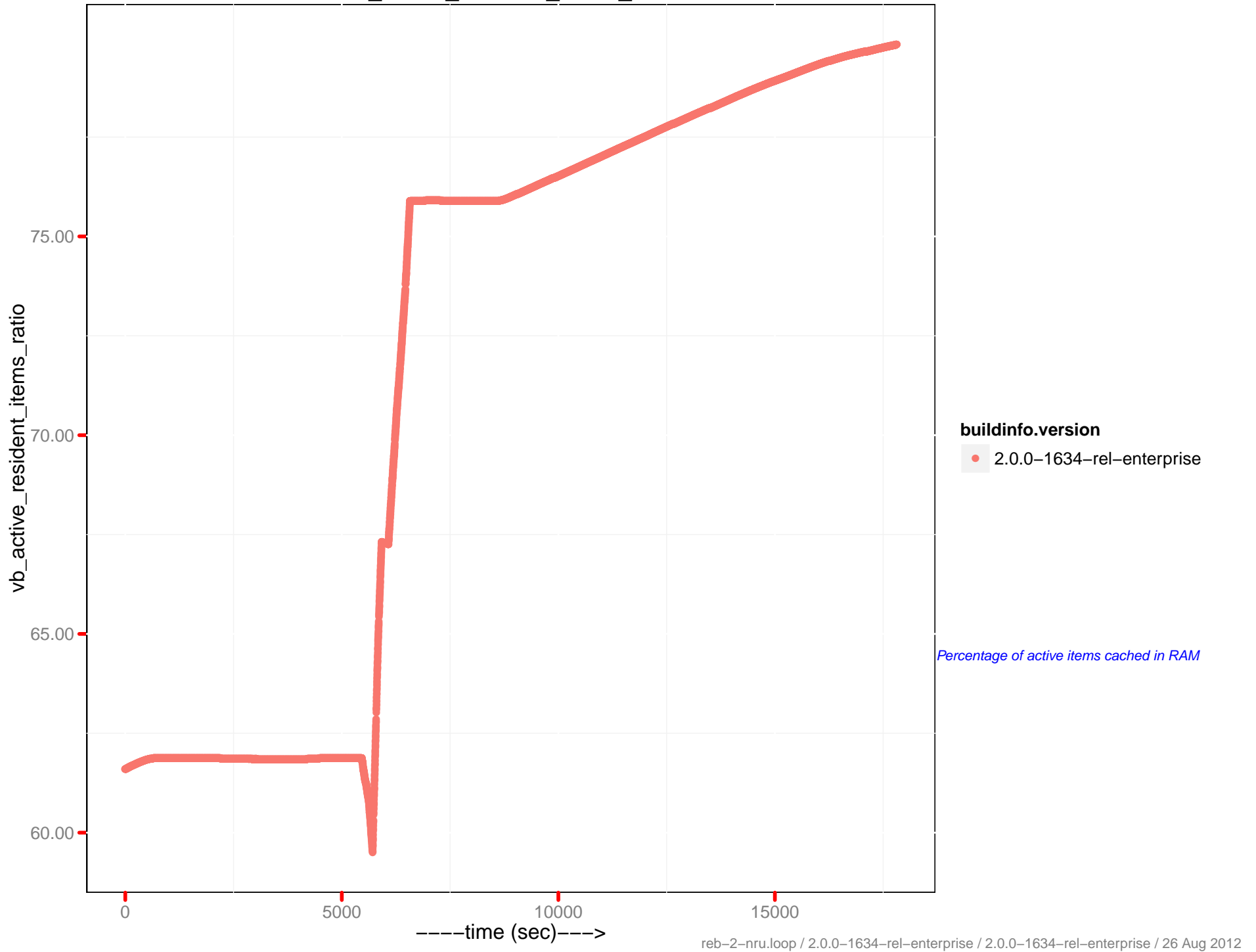
# vb\_replica\_eject/sec



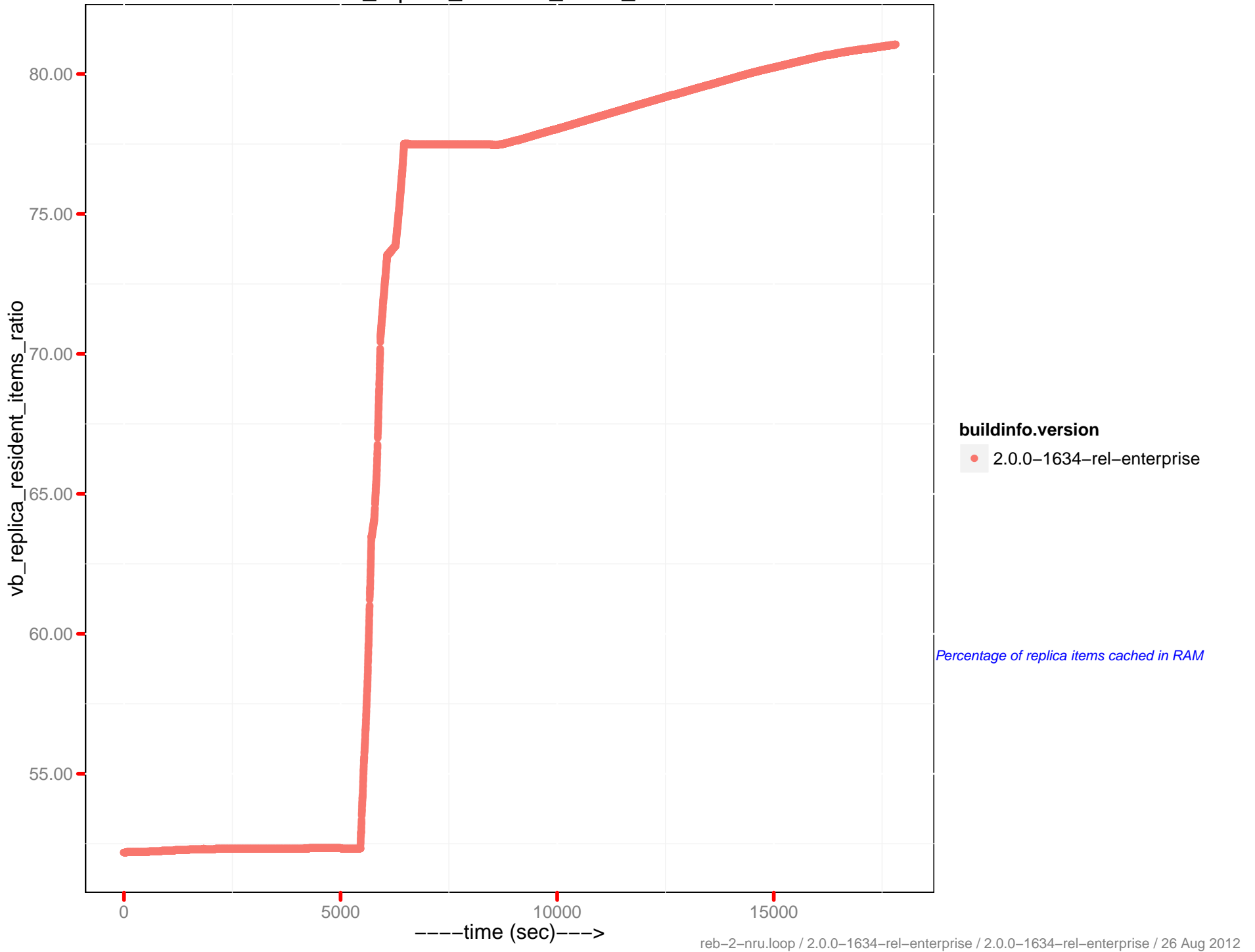
# ep\_tap\_replica\_queue\_backoff/sec



# vb\_active\_resident\_items\_ratio



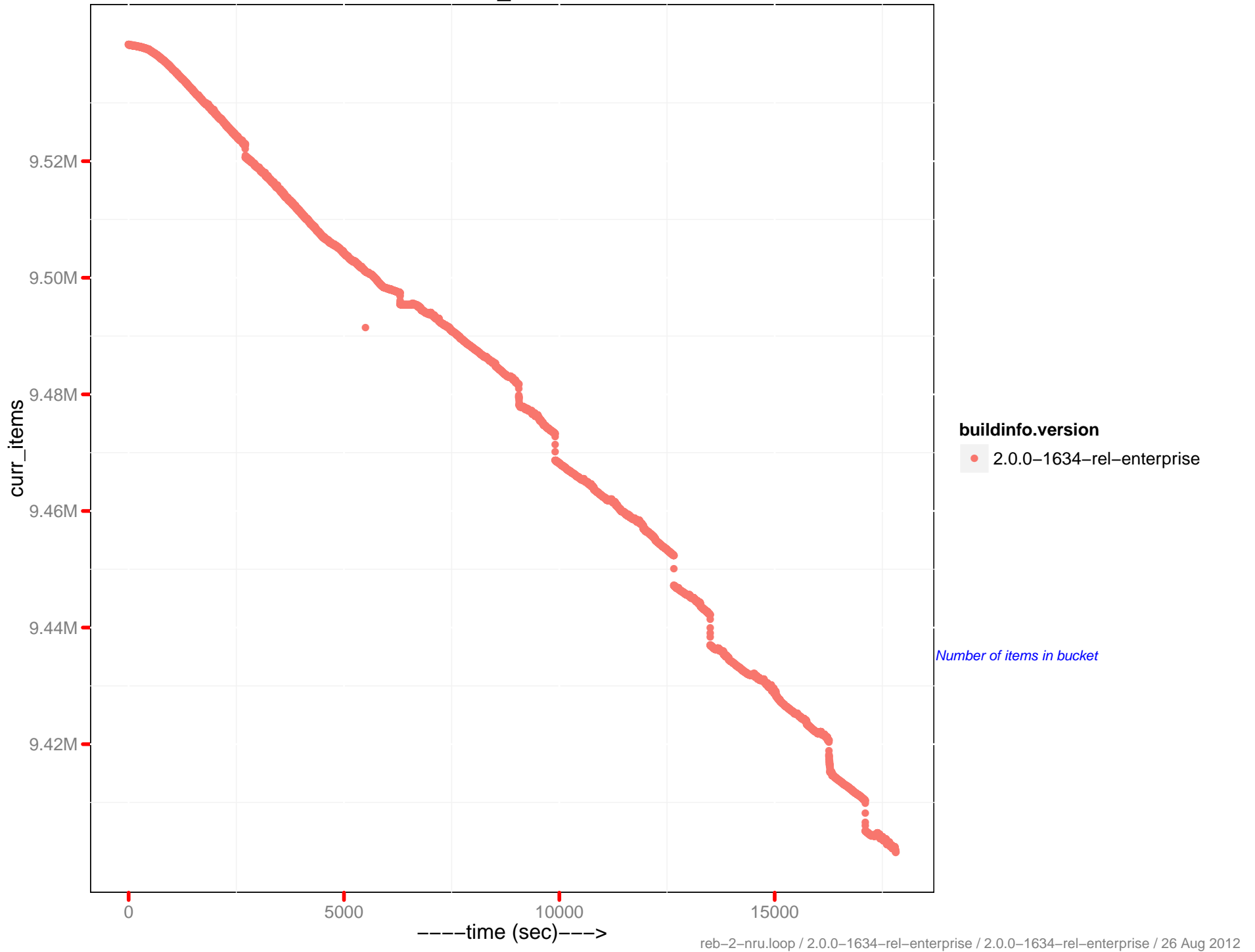
# vb\_replica\_resident\_items\_ratio



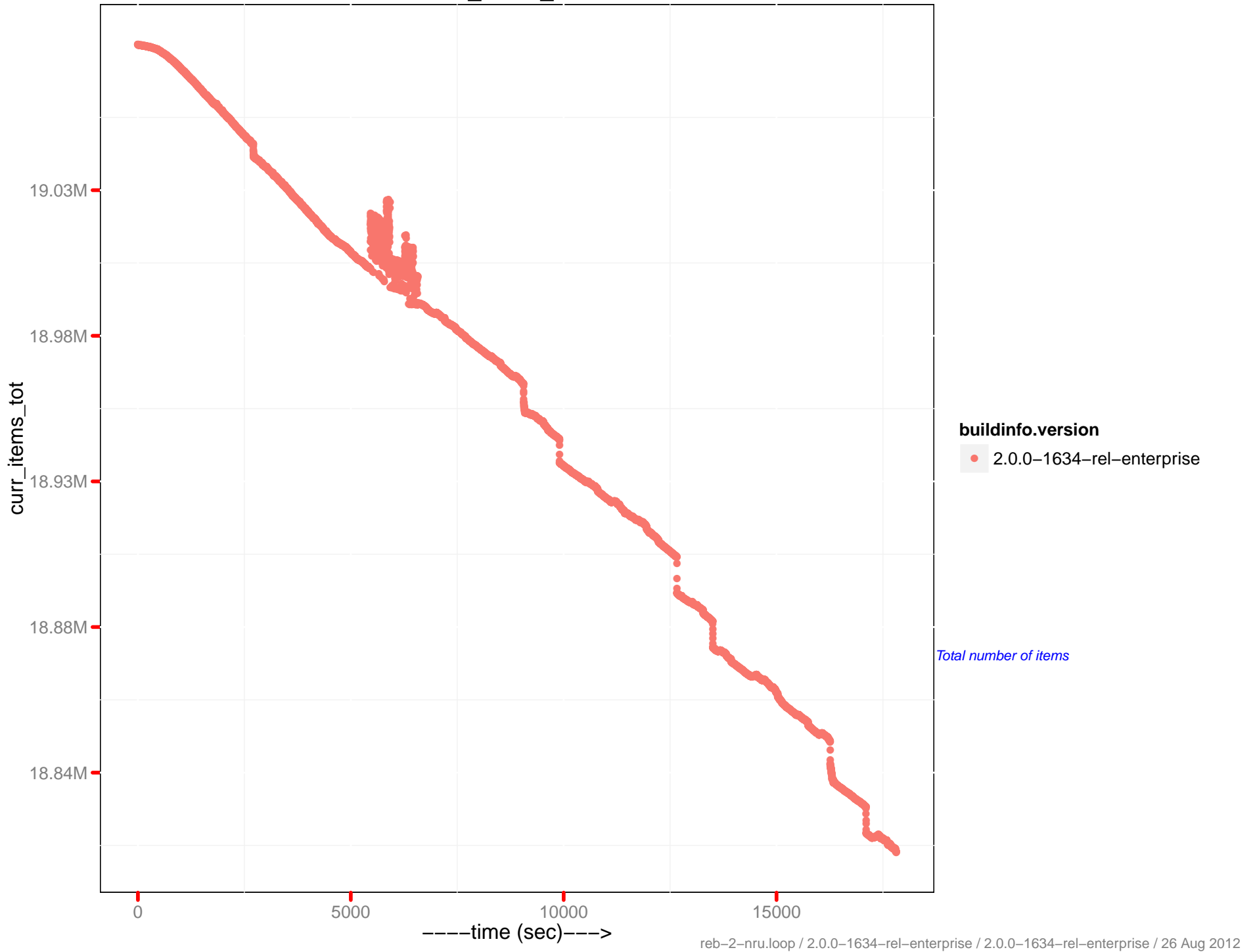
**buildinfo.version**  
● 2.0.0-1634-rel-enterprise

*Percentage of replica items cached in RAM*

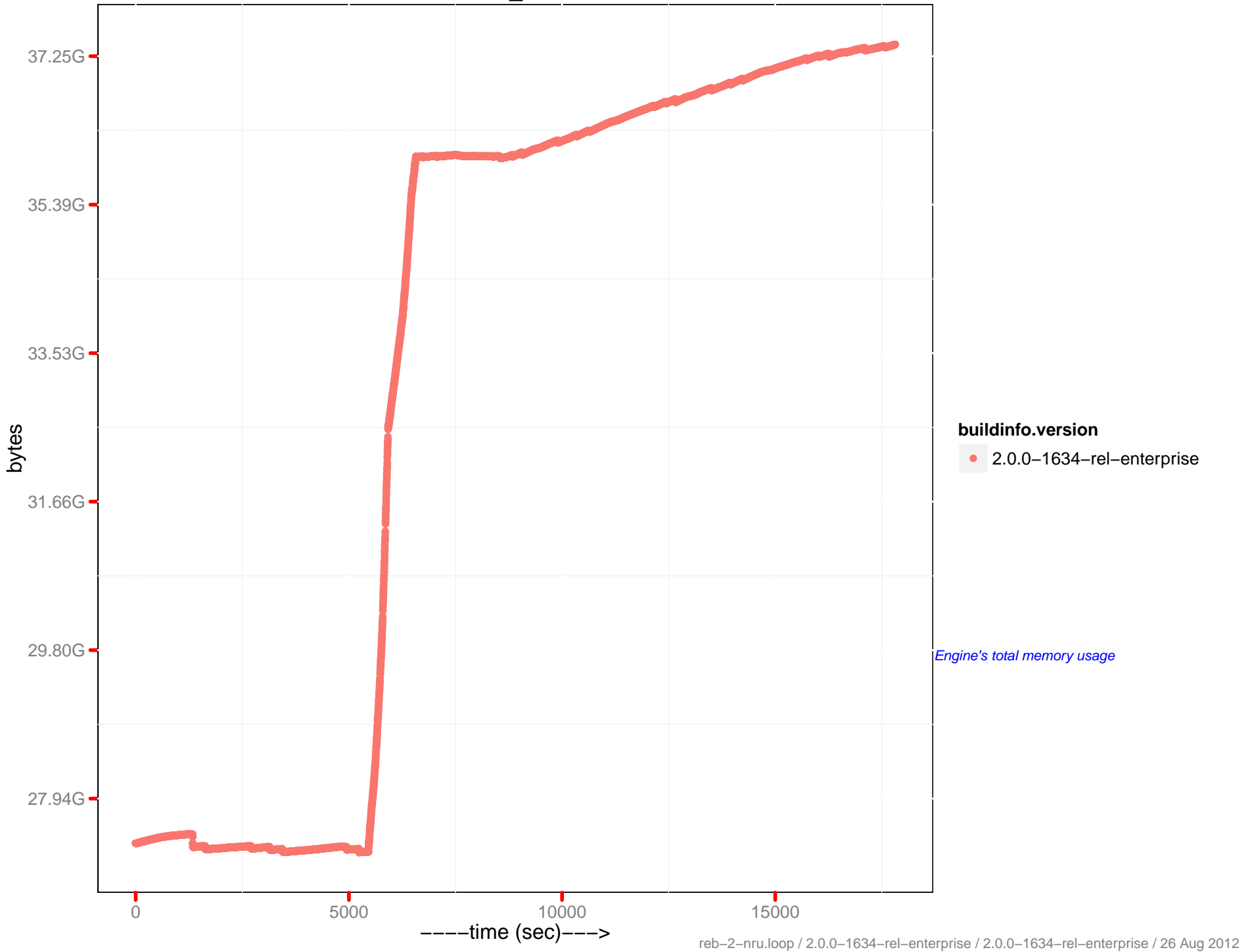
# curr\_items



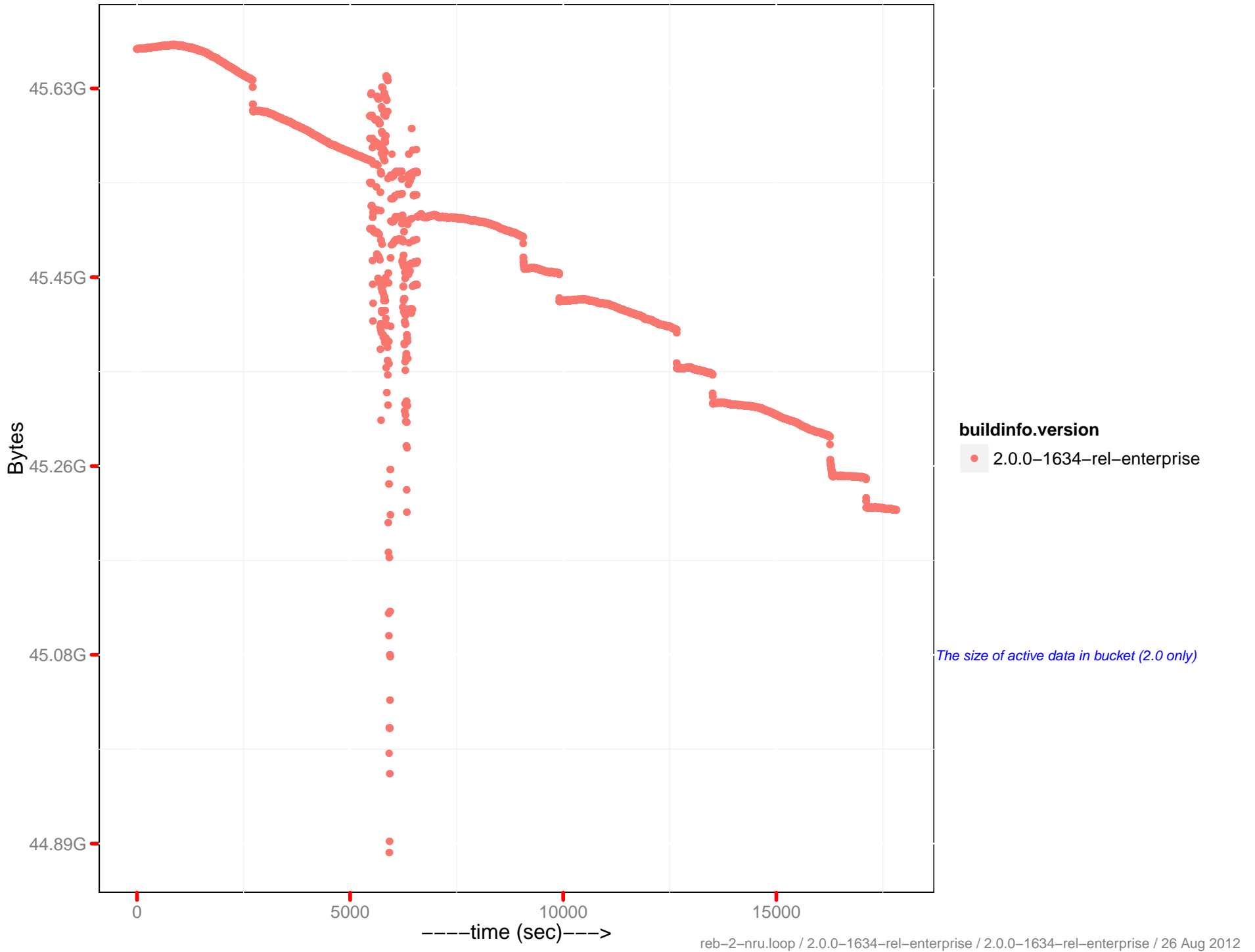
cur\_items\_total



mem\_used

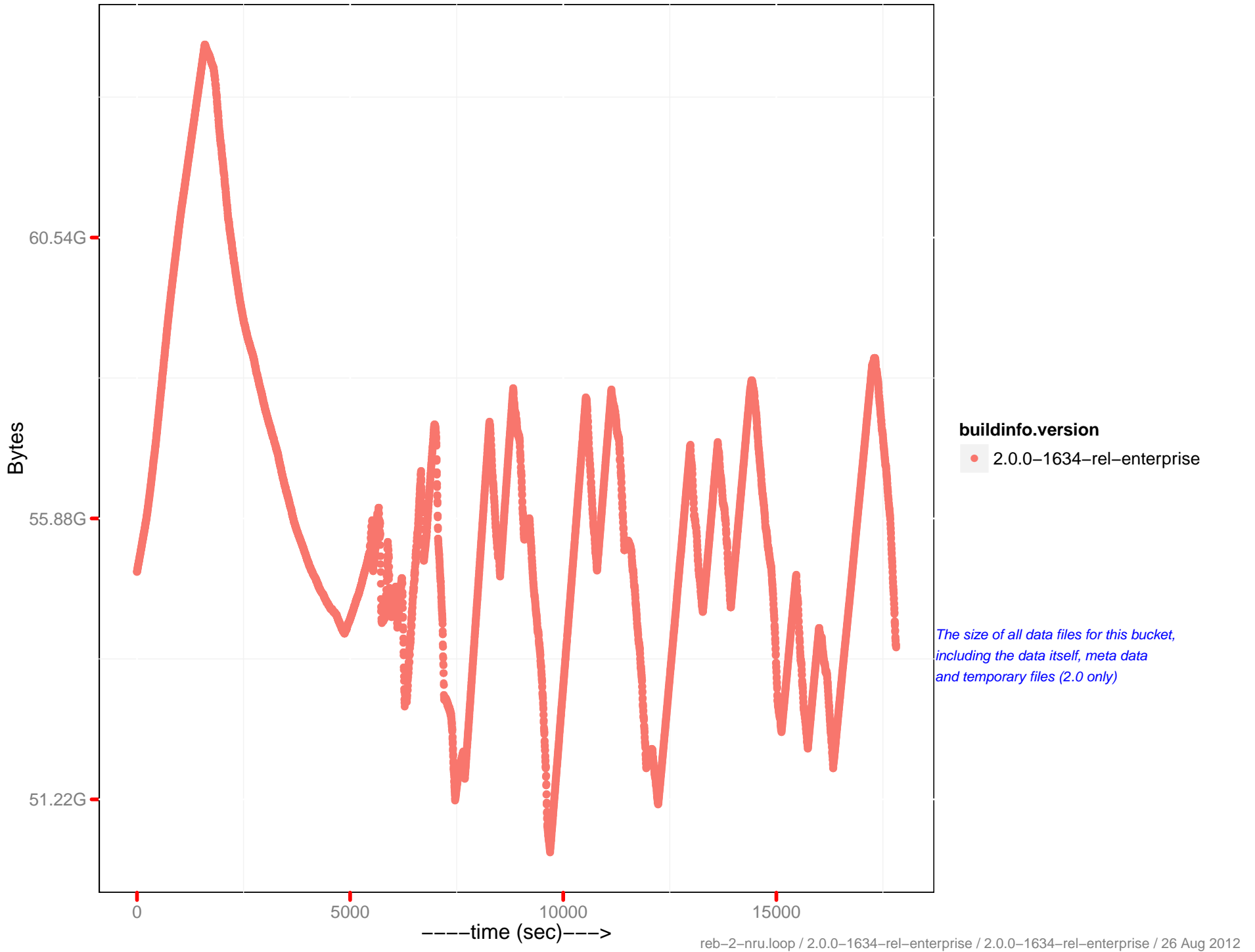


# Docs data size

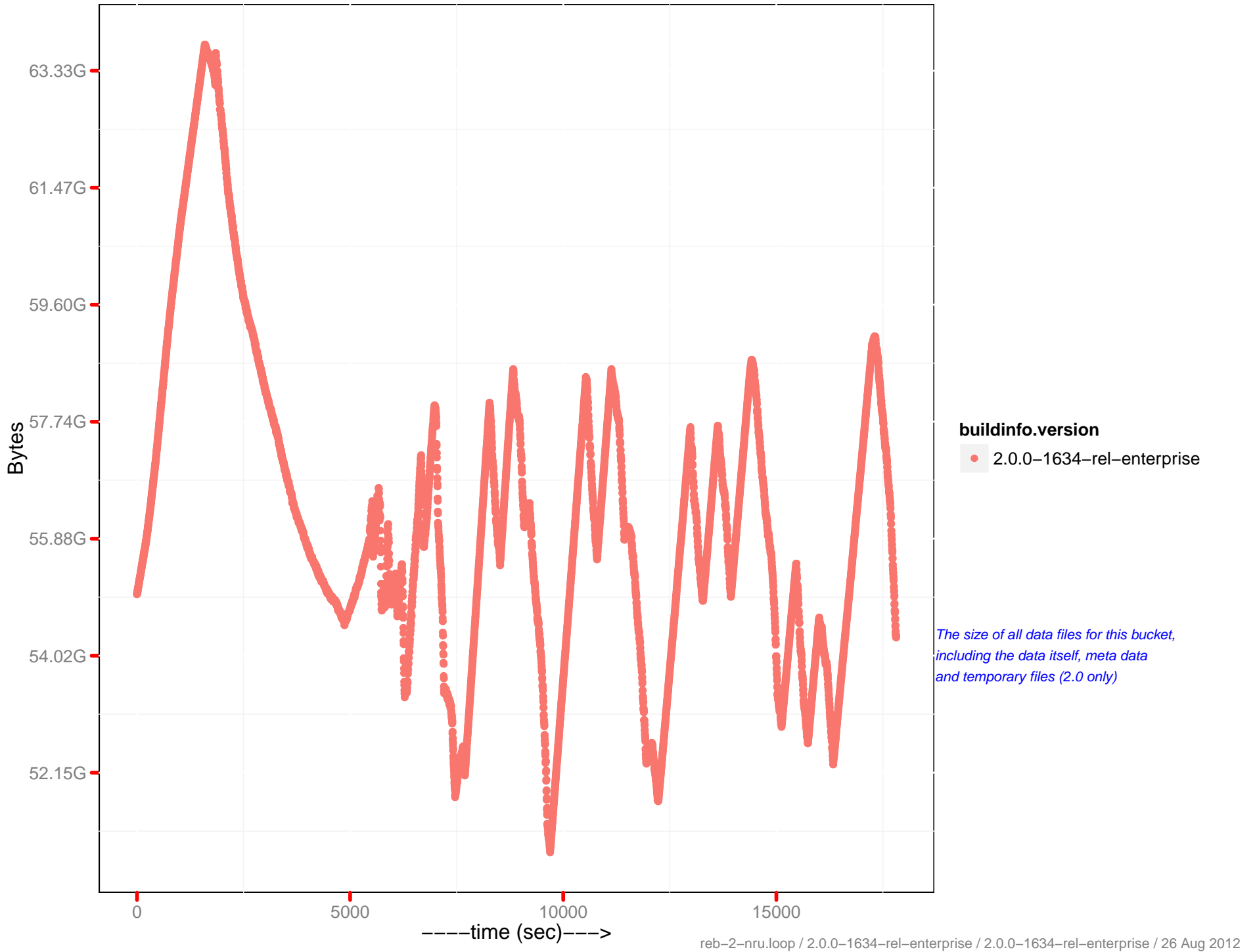




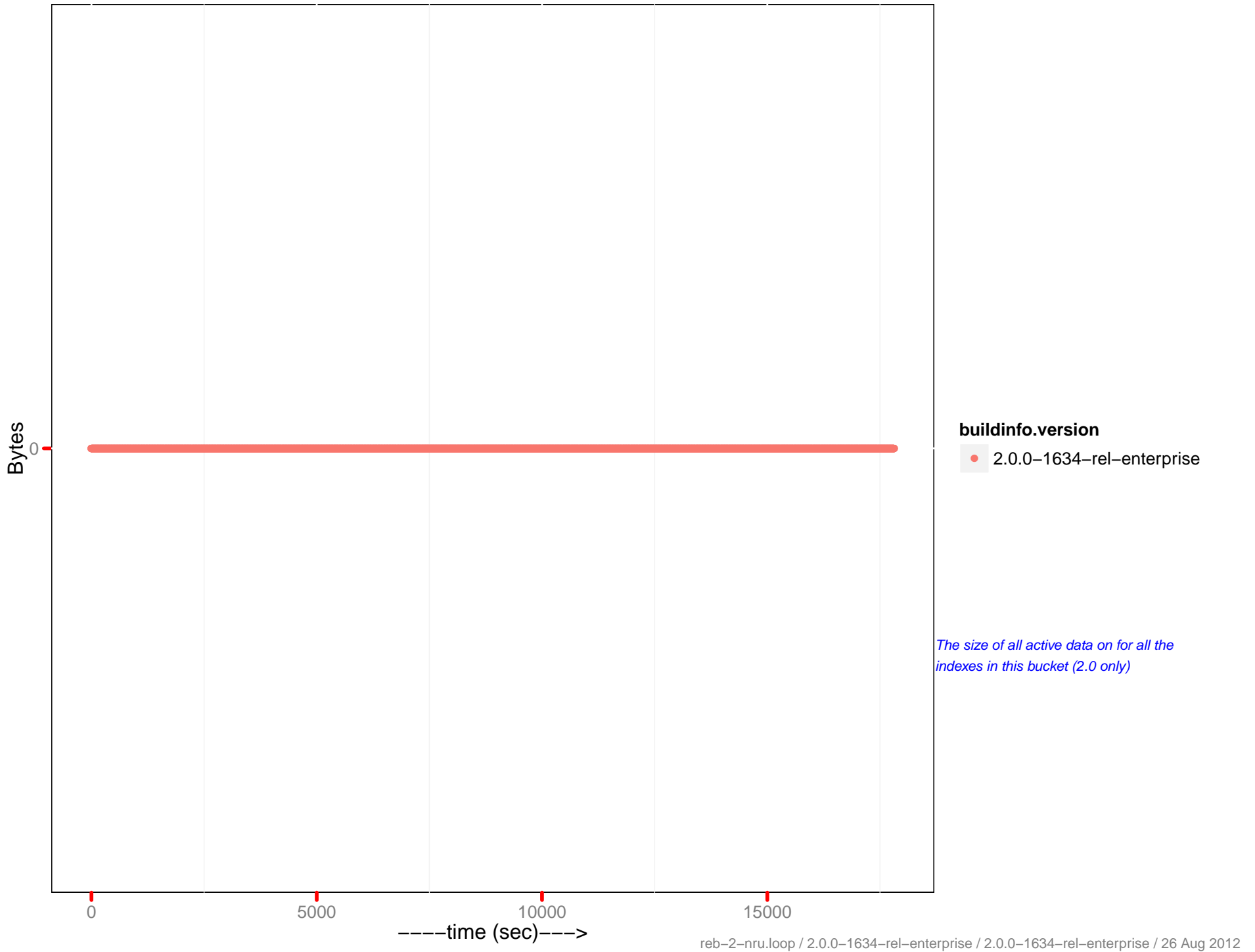
# Docs disk size



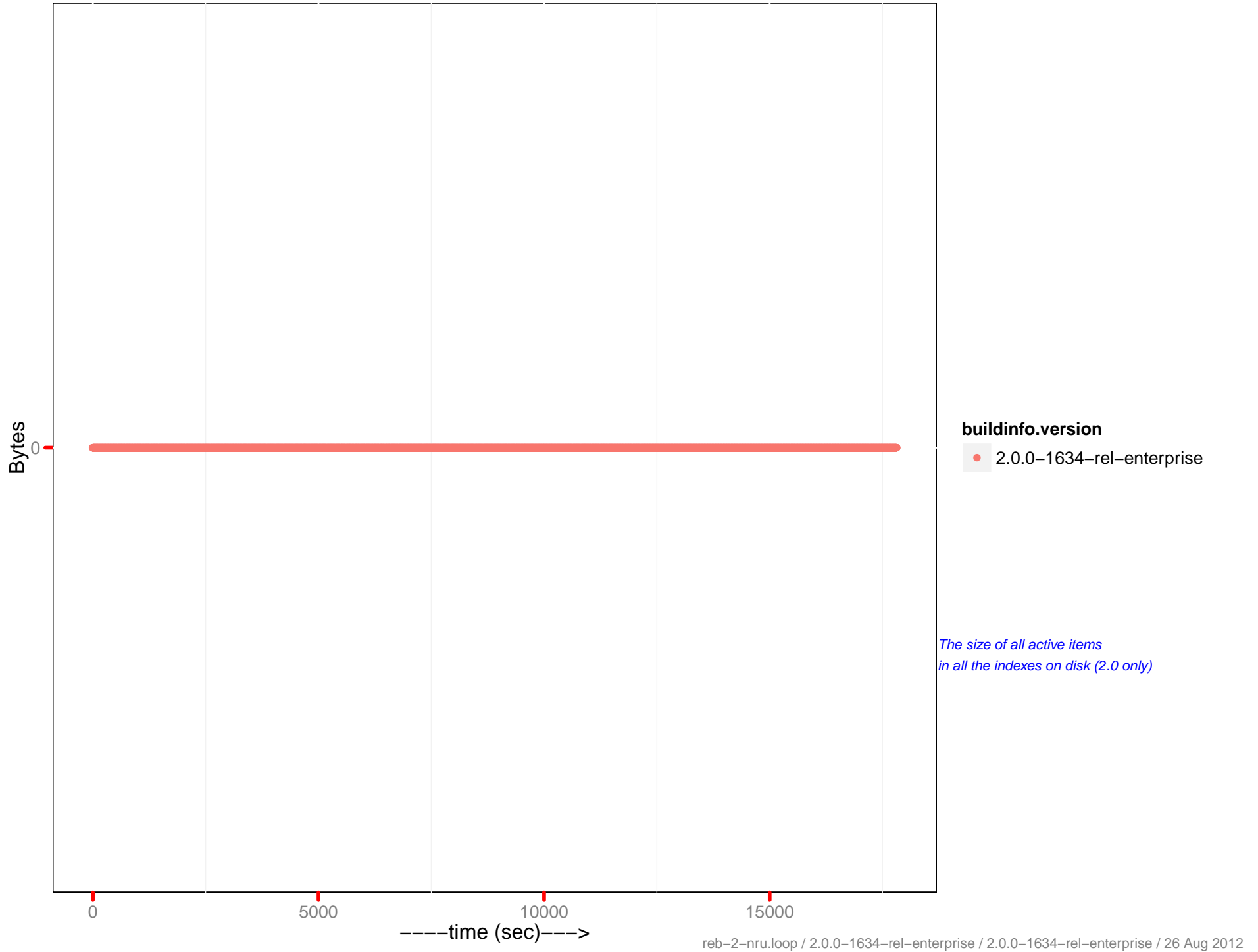
# Docs actual disk size



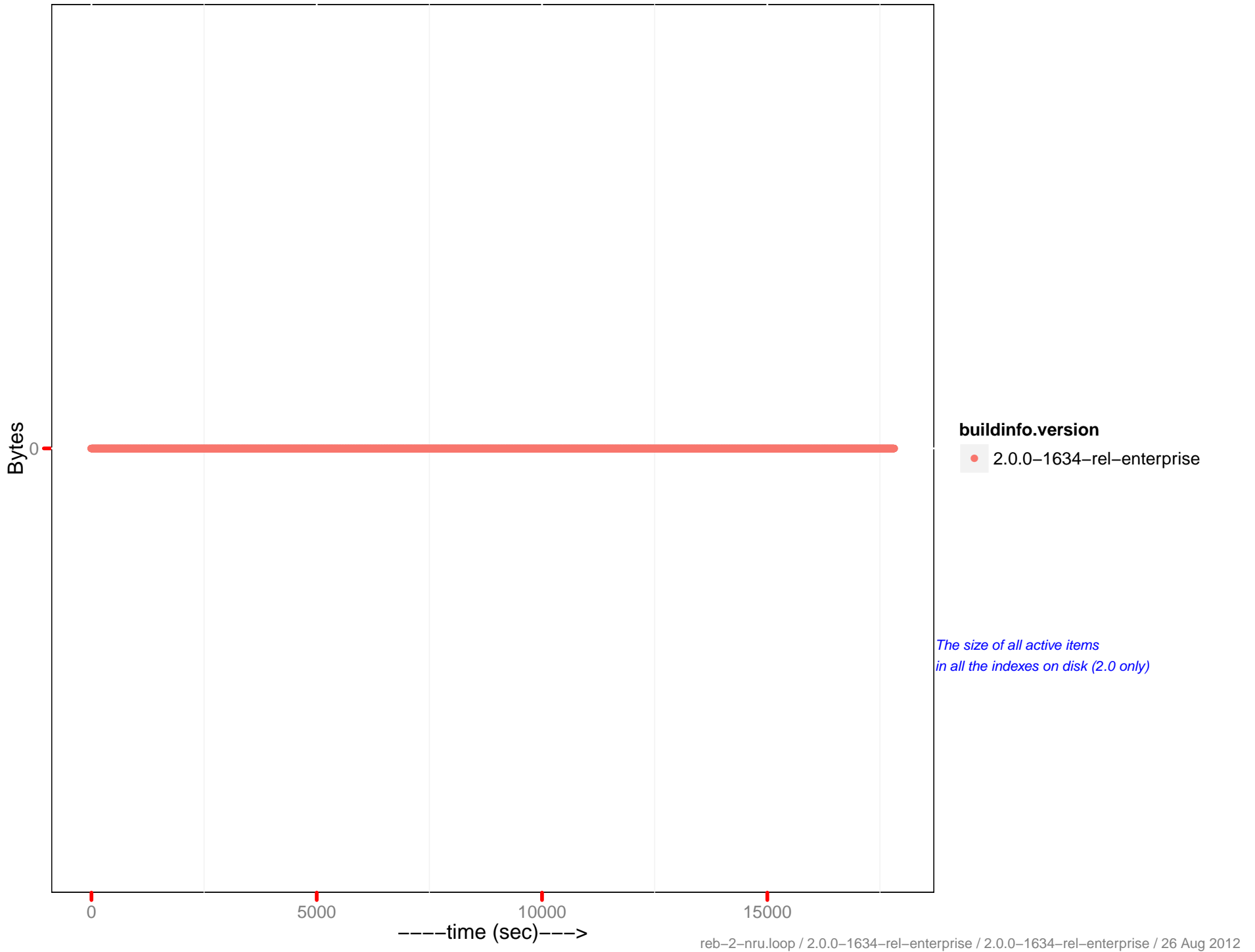
# Views data size



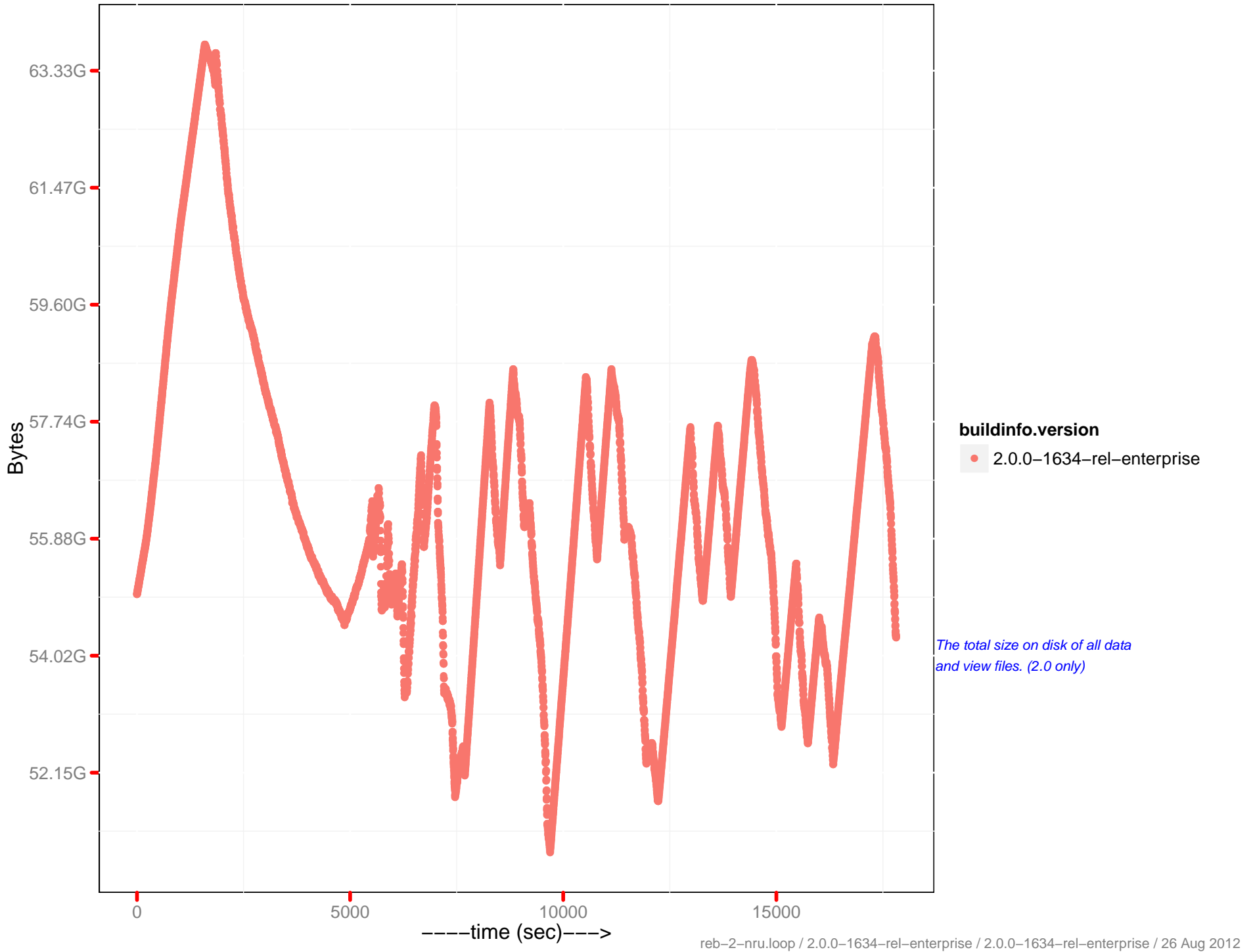
# Views disk size



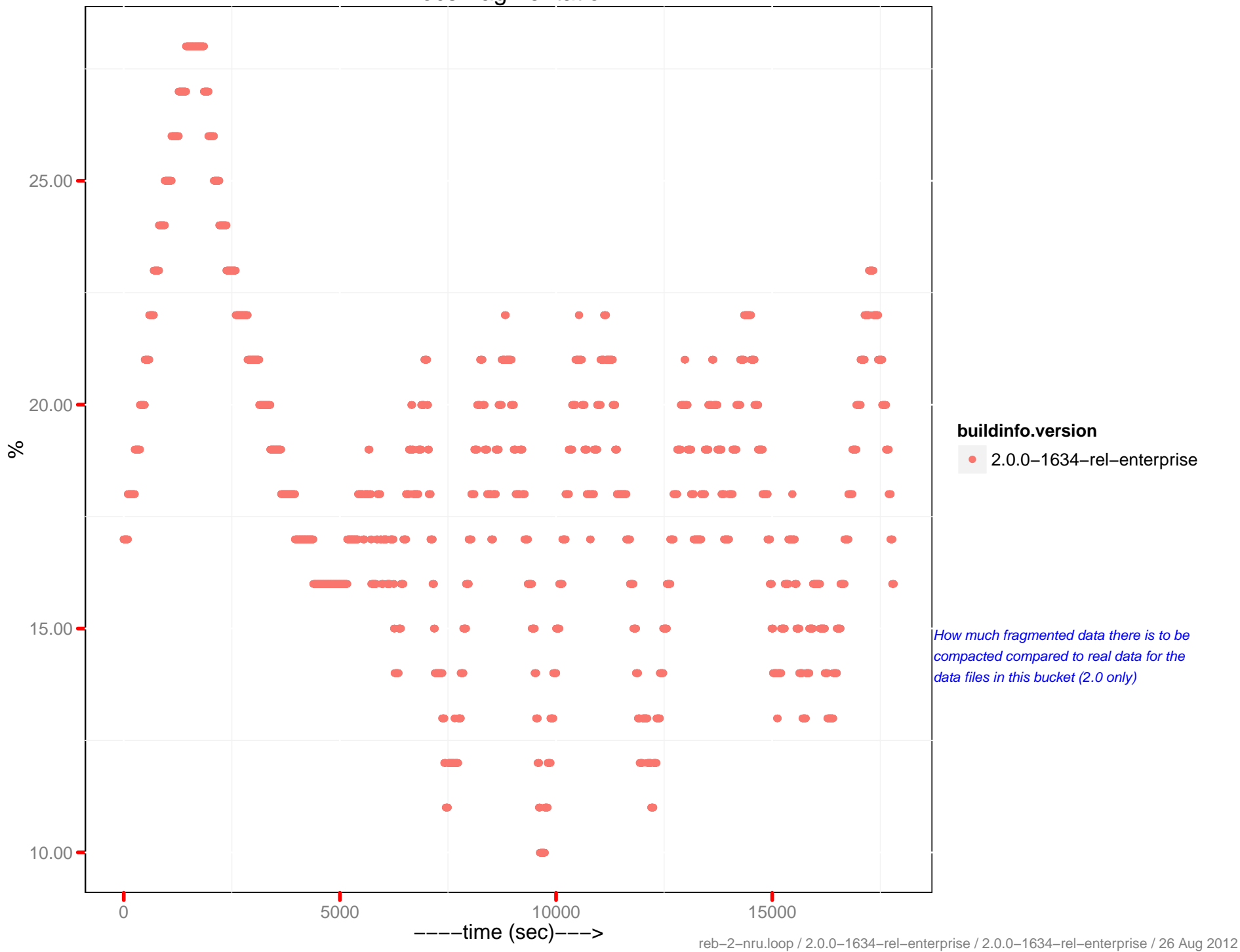
# Views actual disk size



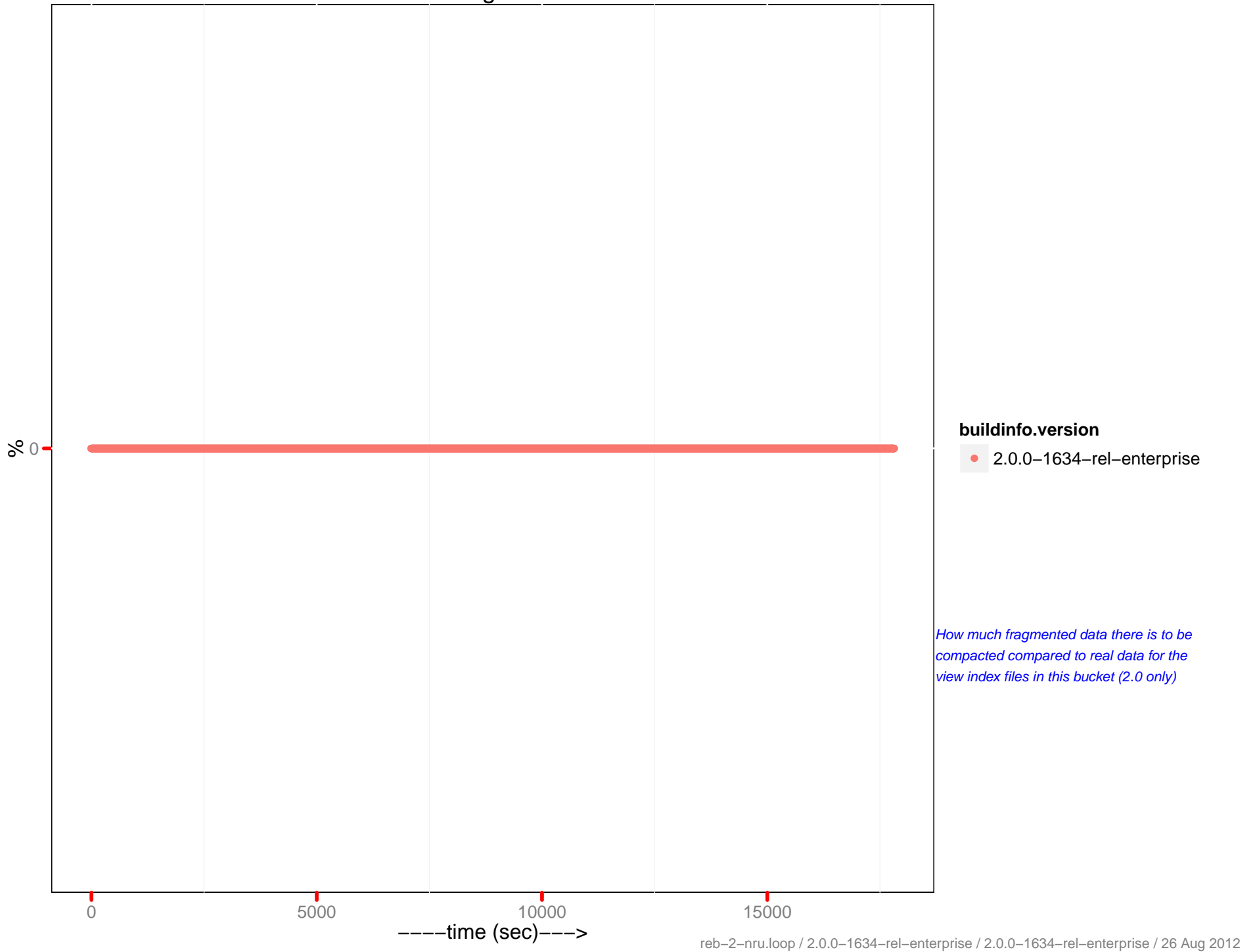
# Total disk size



# Docs fragmentation



# Views fragmentation



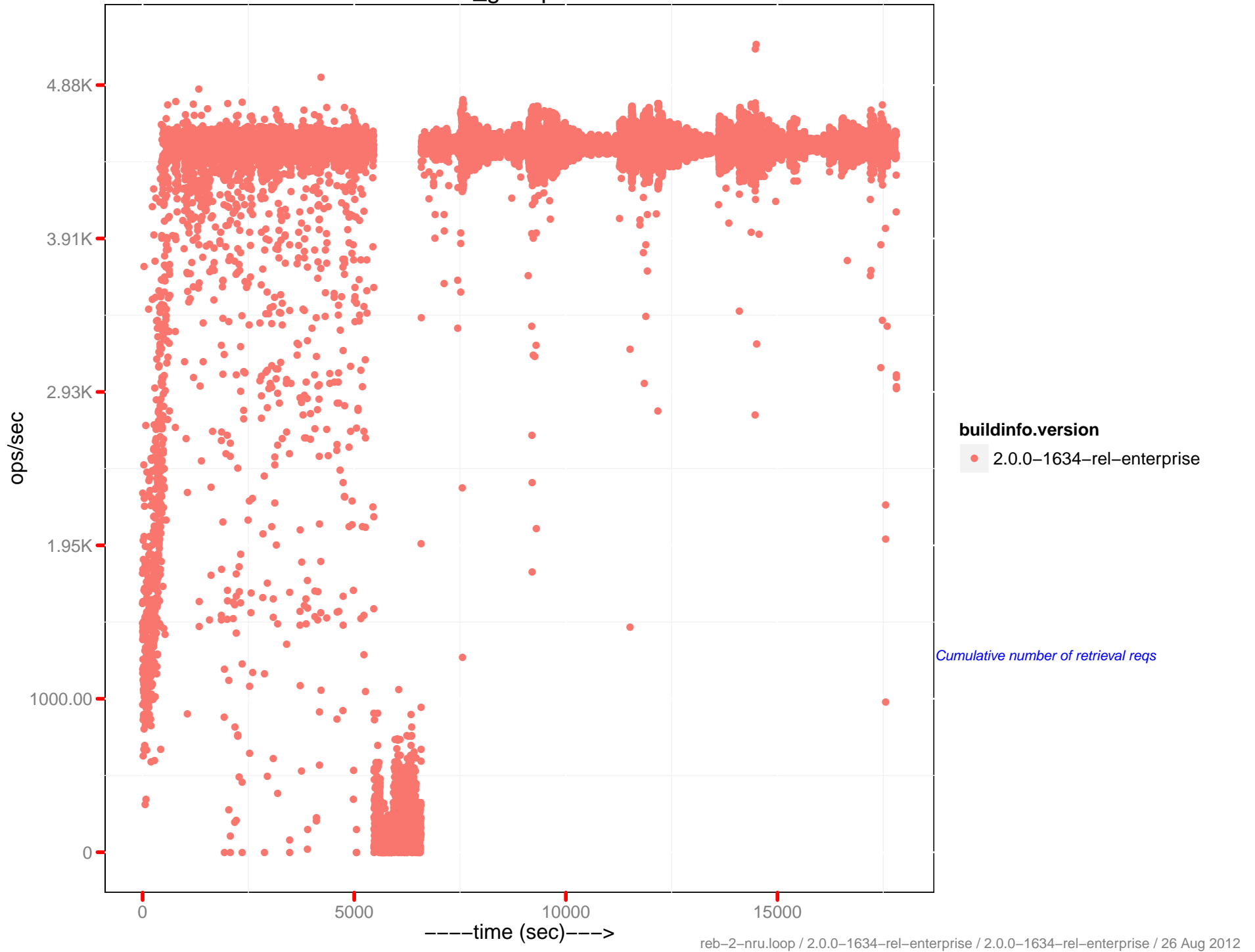
**buildinfo.version**

• 2.0.0-1634-rel-enterprise

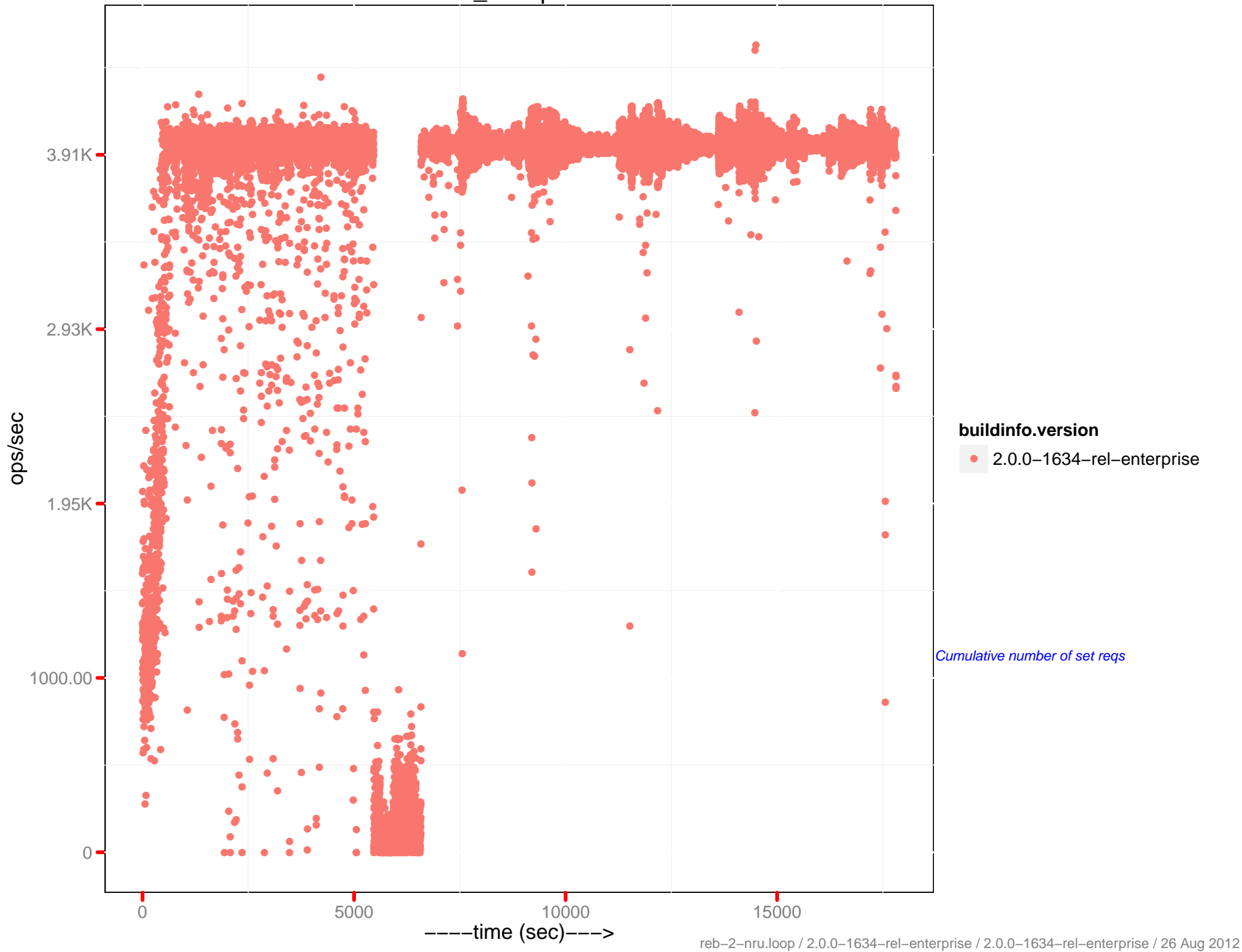
*How much fragmented data there is to be compacted compared to real data for the view index files in this bucket (2.0 only)*



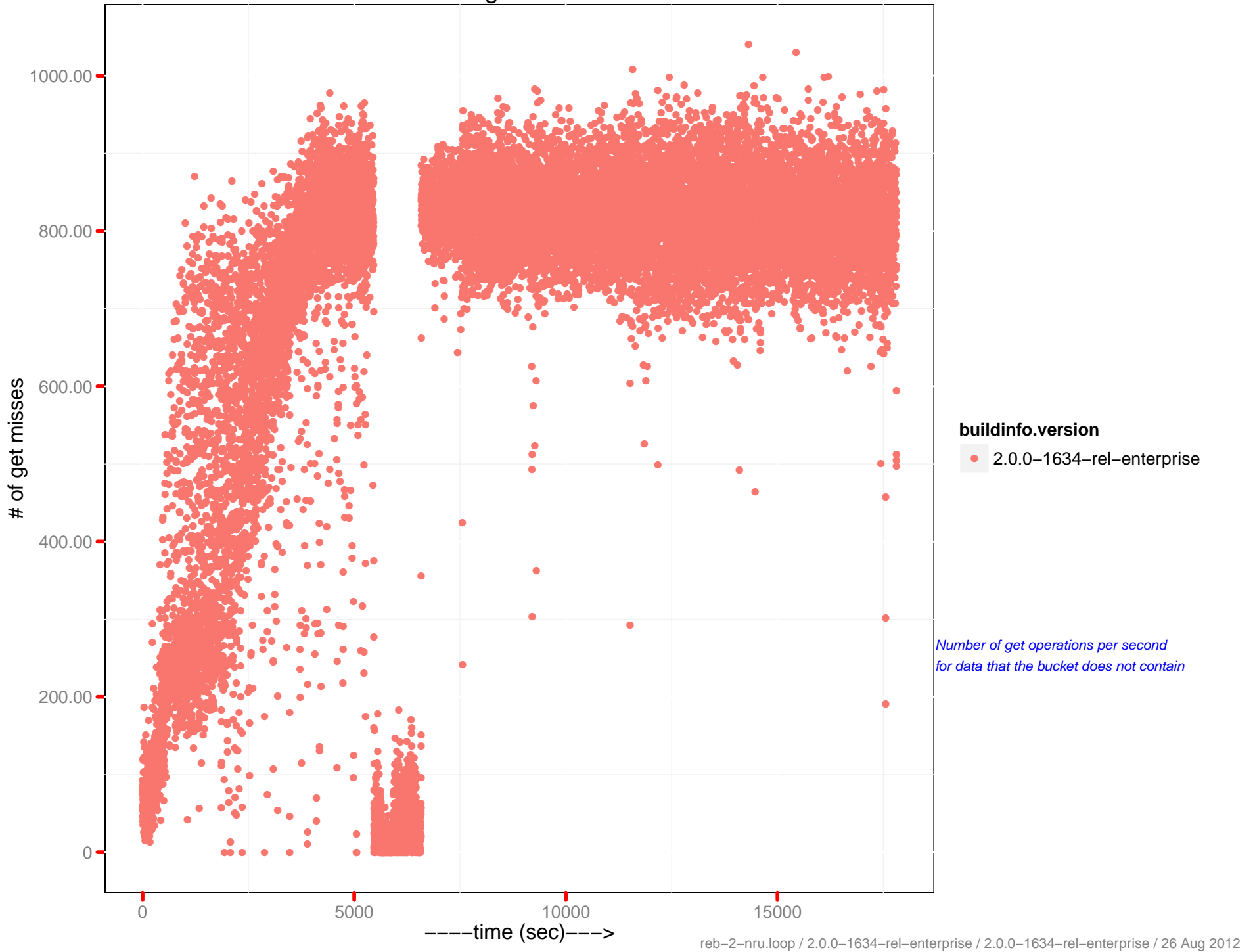
# cmd\_get ops/sec



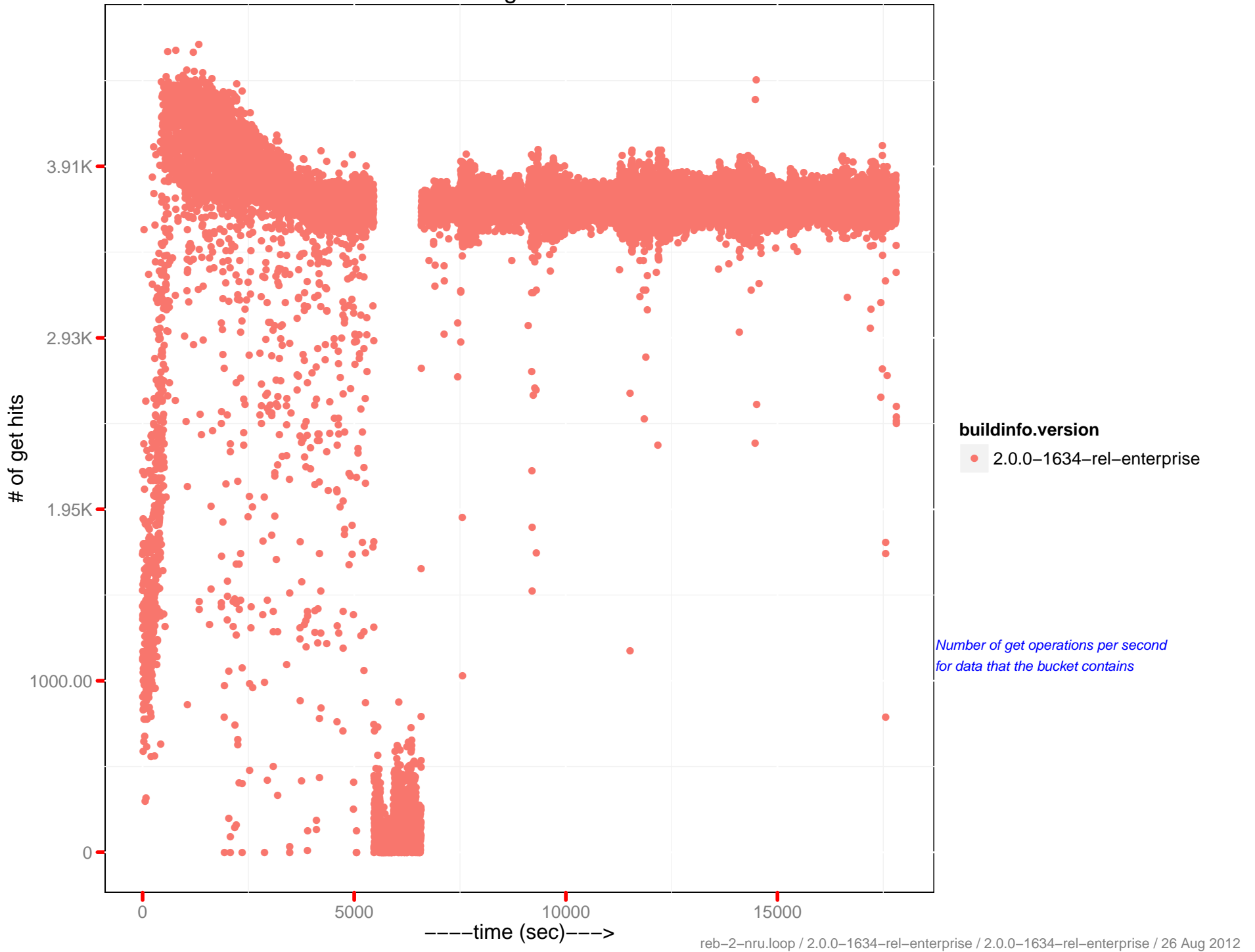
cmd\_set ops/sec



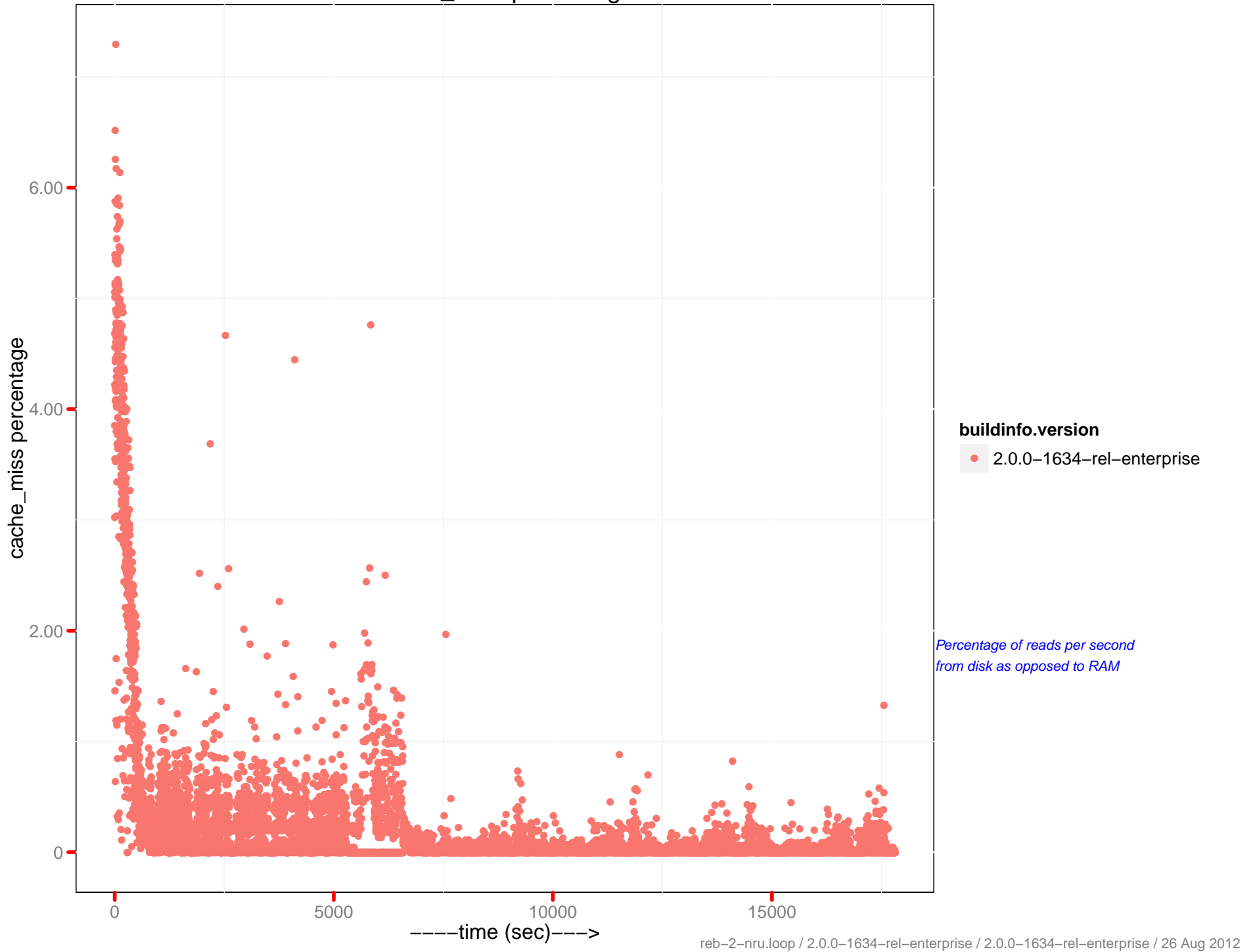
# # of get misses



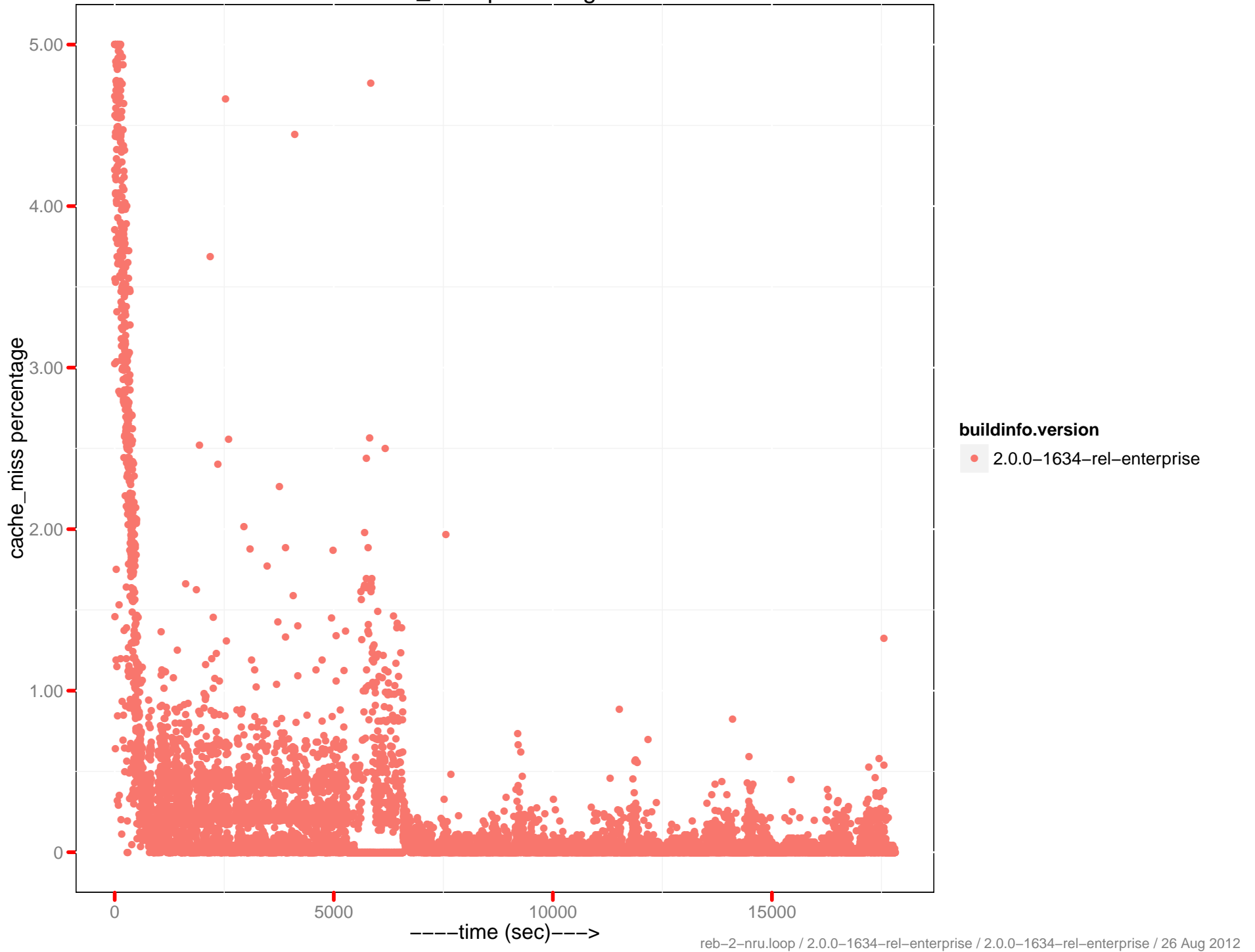
# # of get hits



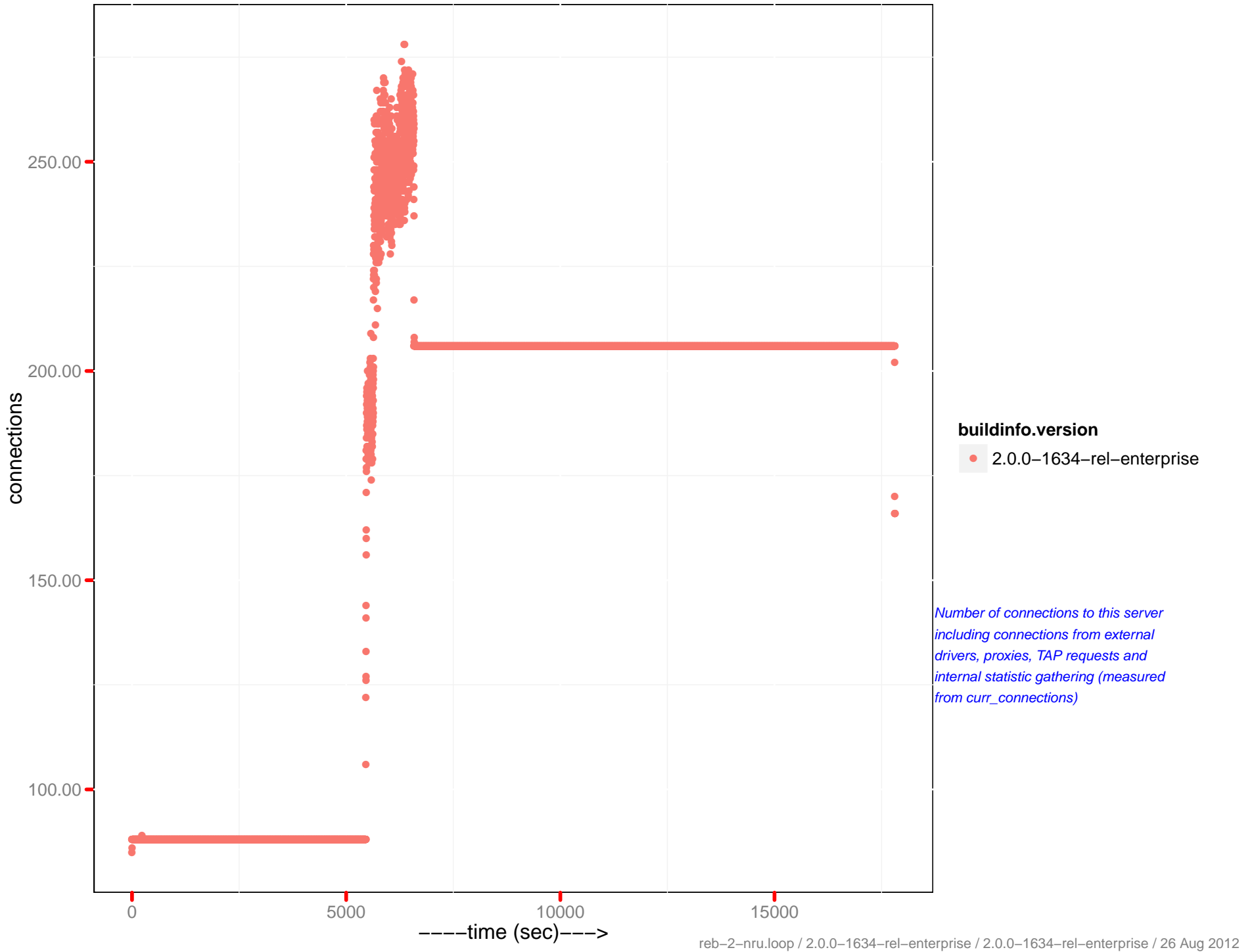
# cache\_miss percentage



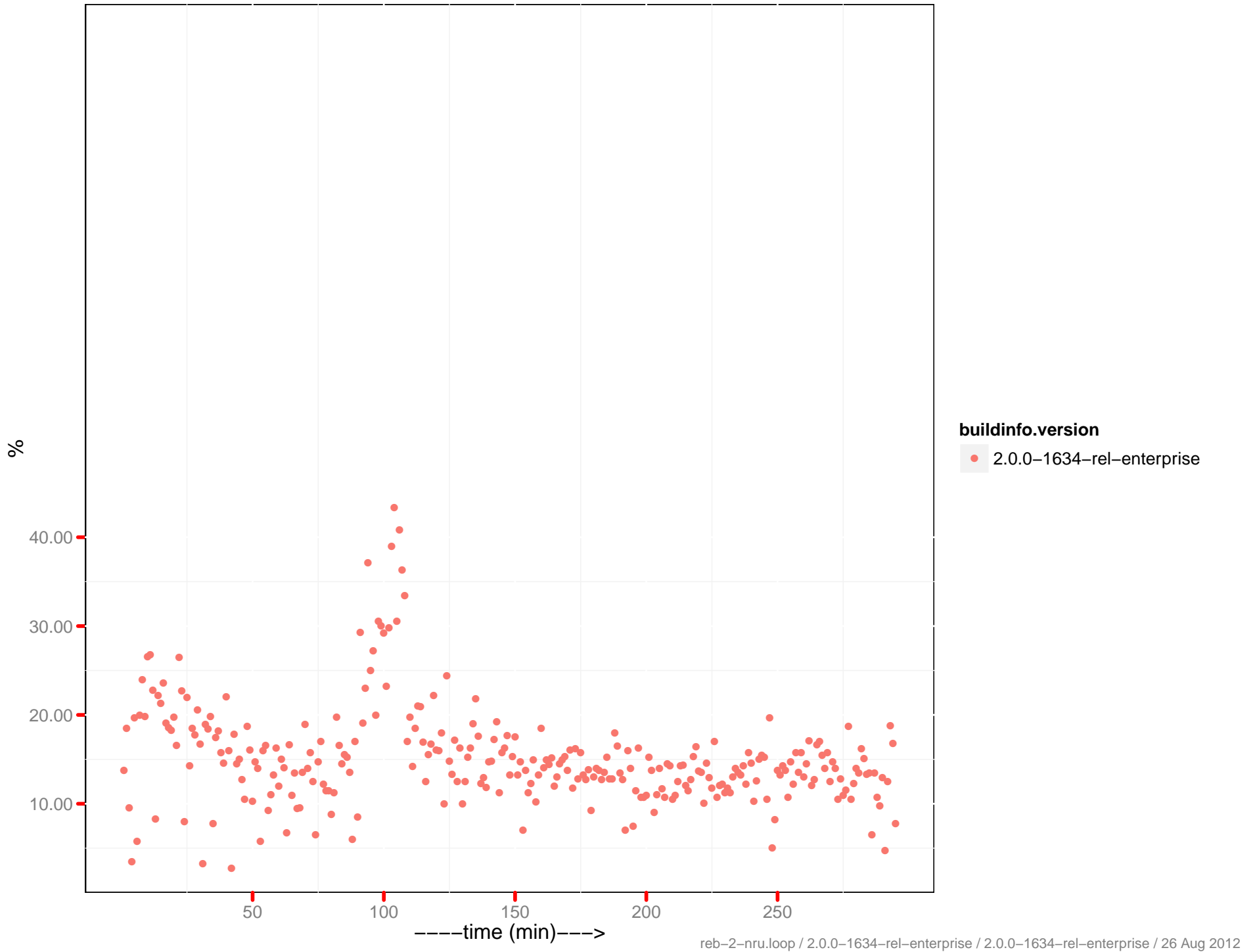
cache\_miss percentage 0-5



# Number of connections

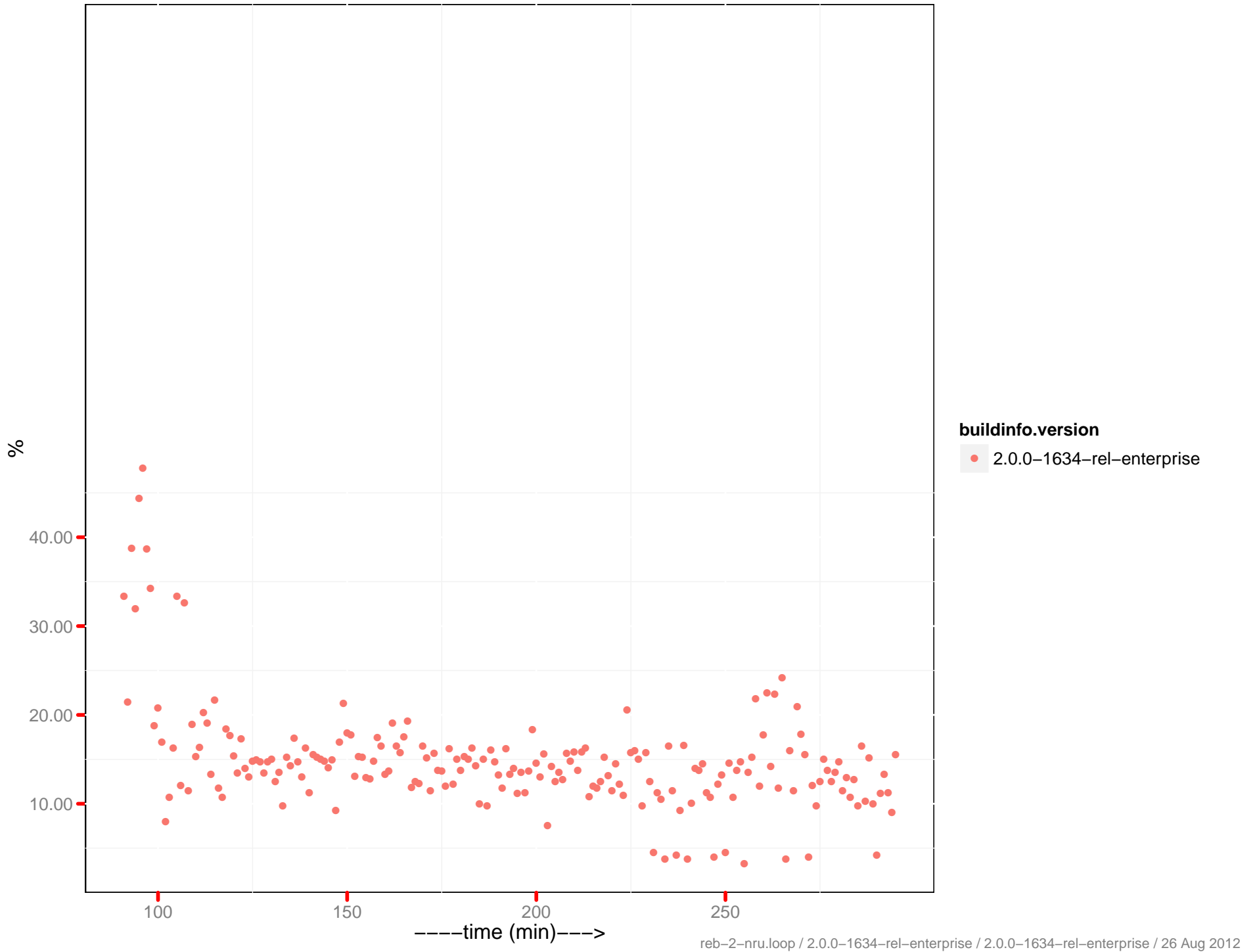


# CPU utilization – 10.2.1.65:8091

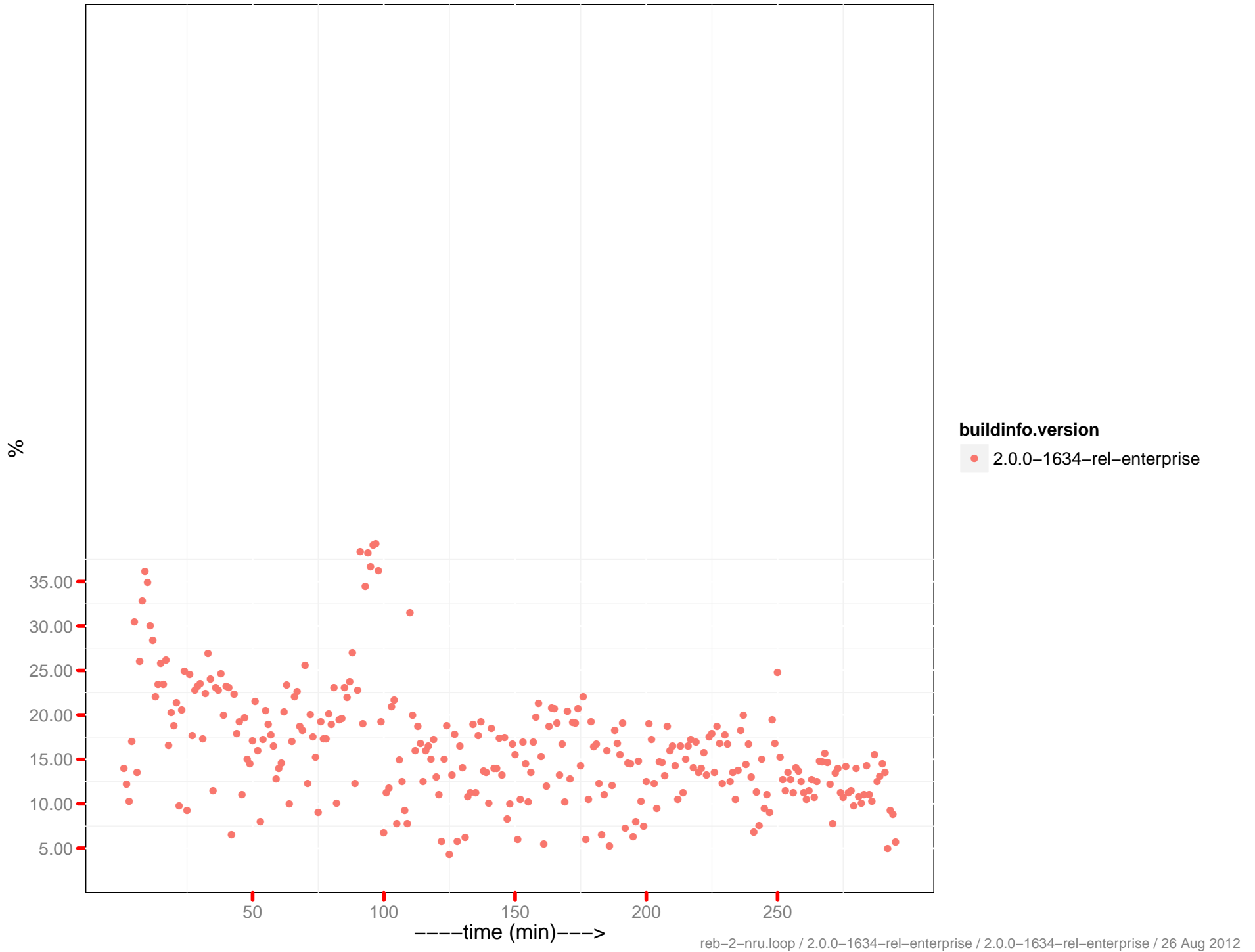




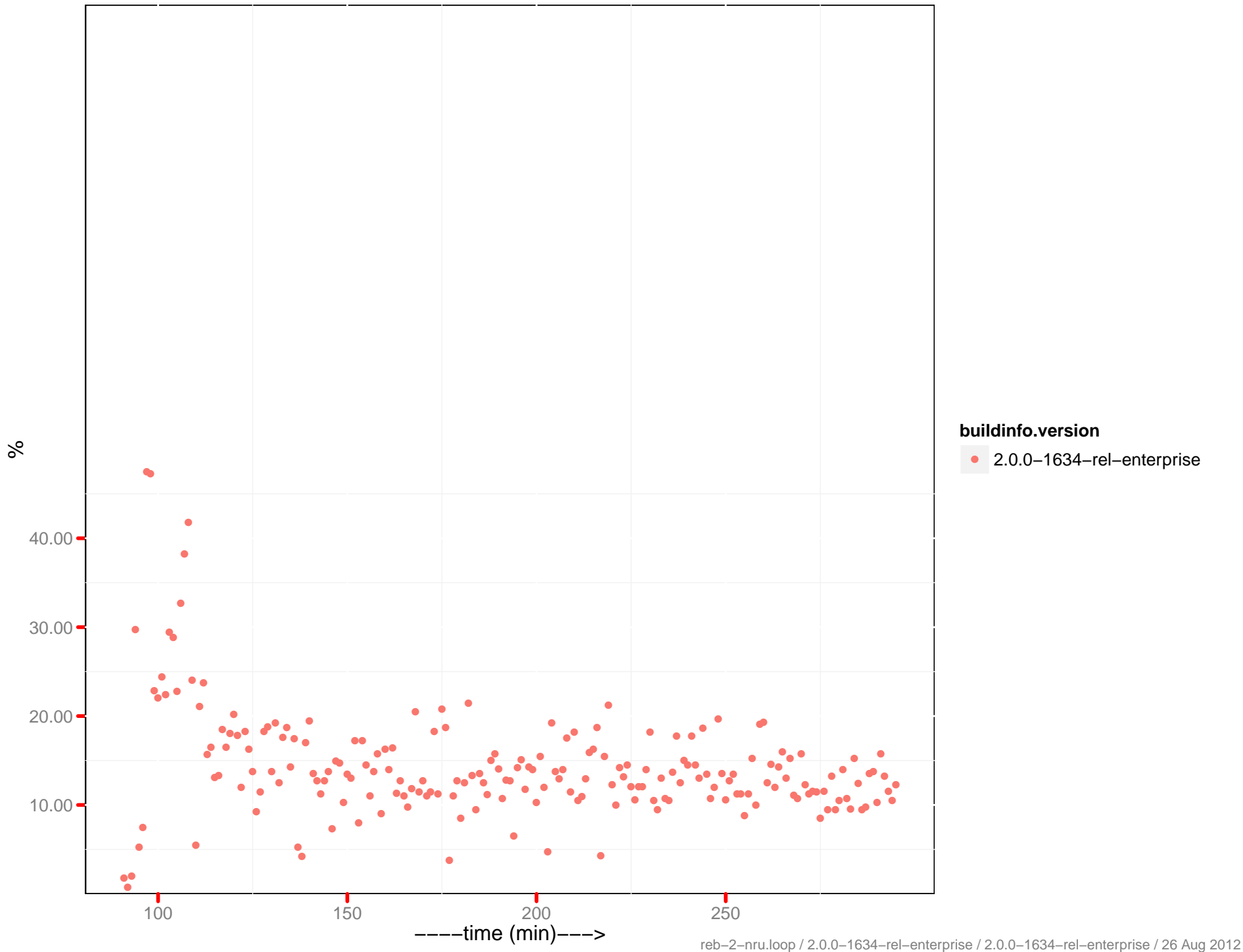
# CPU utilization – 10.2.1.66:8091



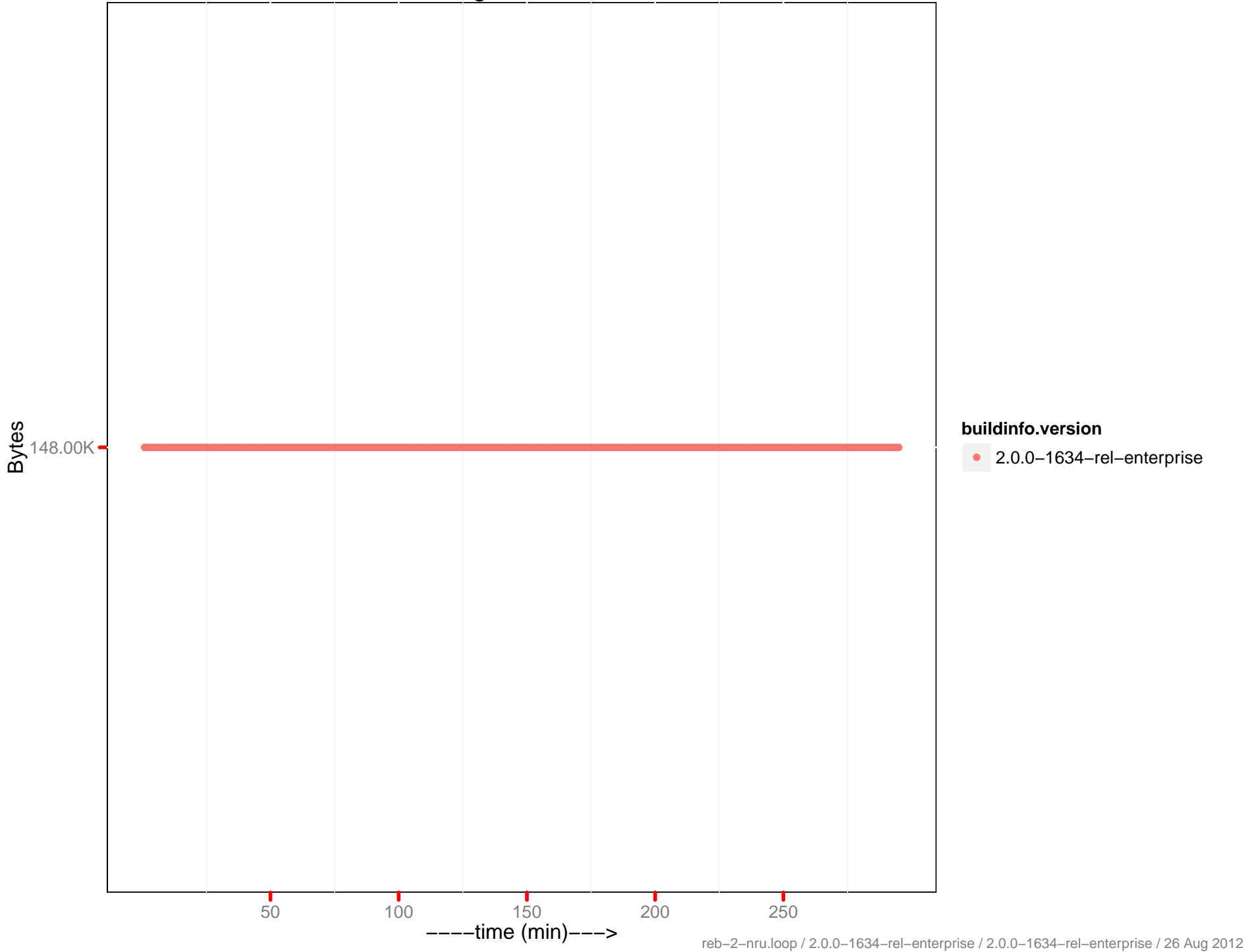
# CPU utilization – 10.2.1.67:8091



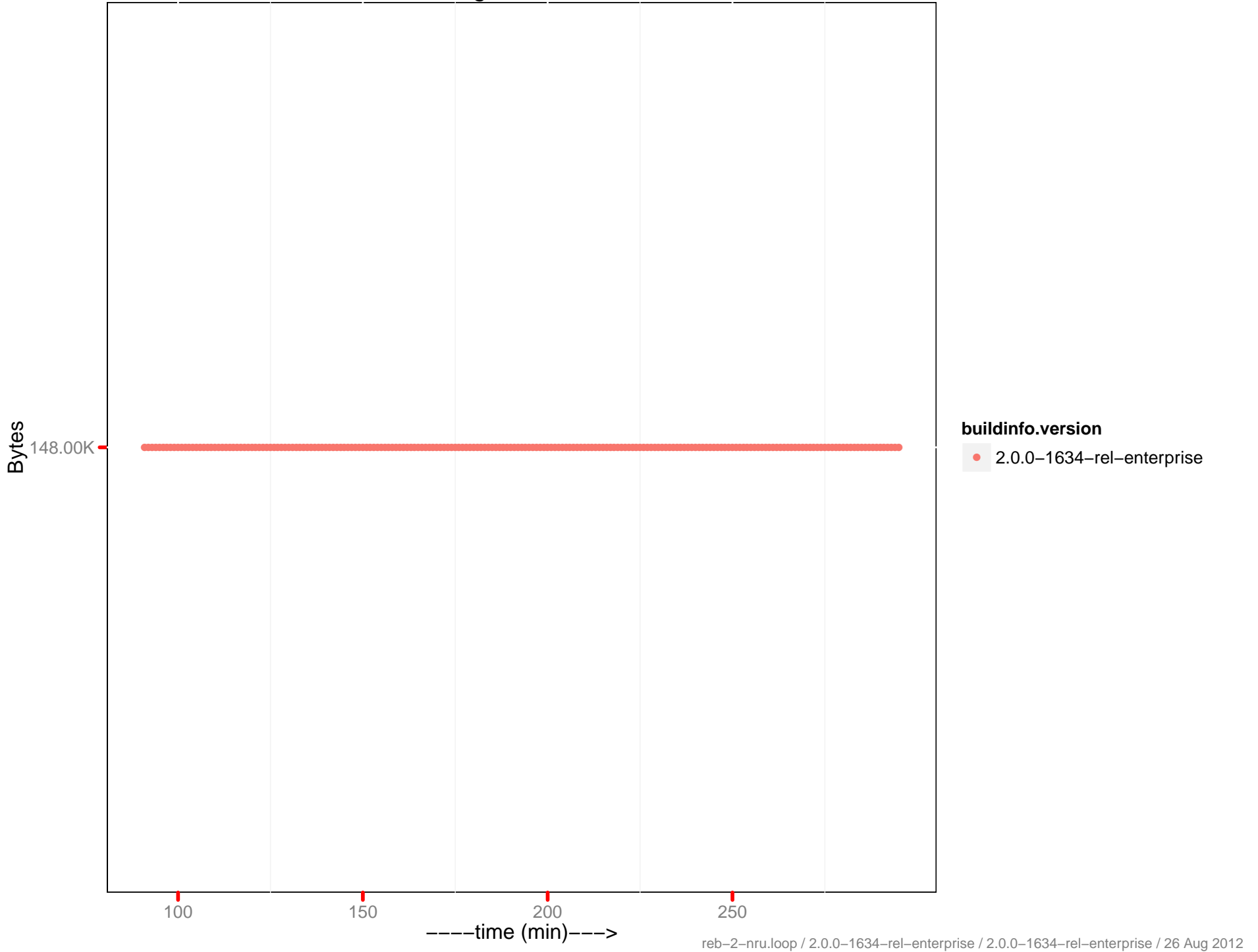
# CPU utilization – 10.2.1.68:8091



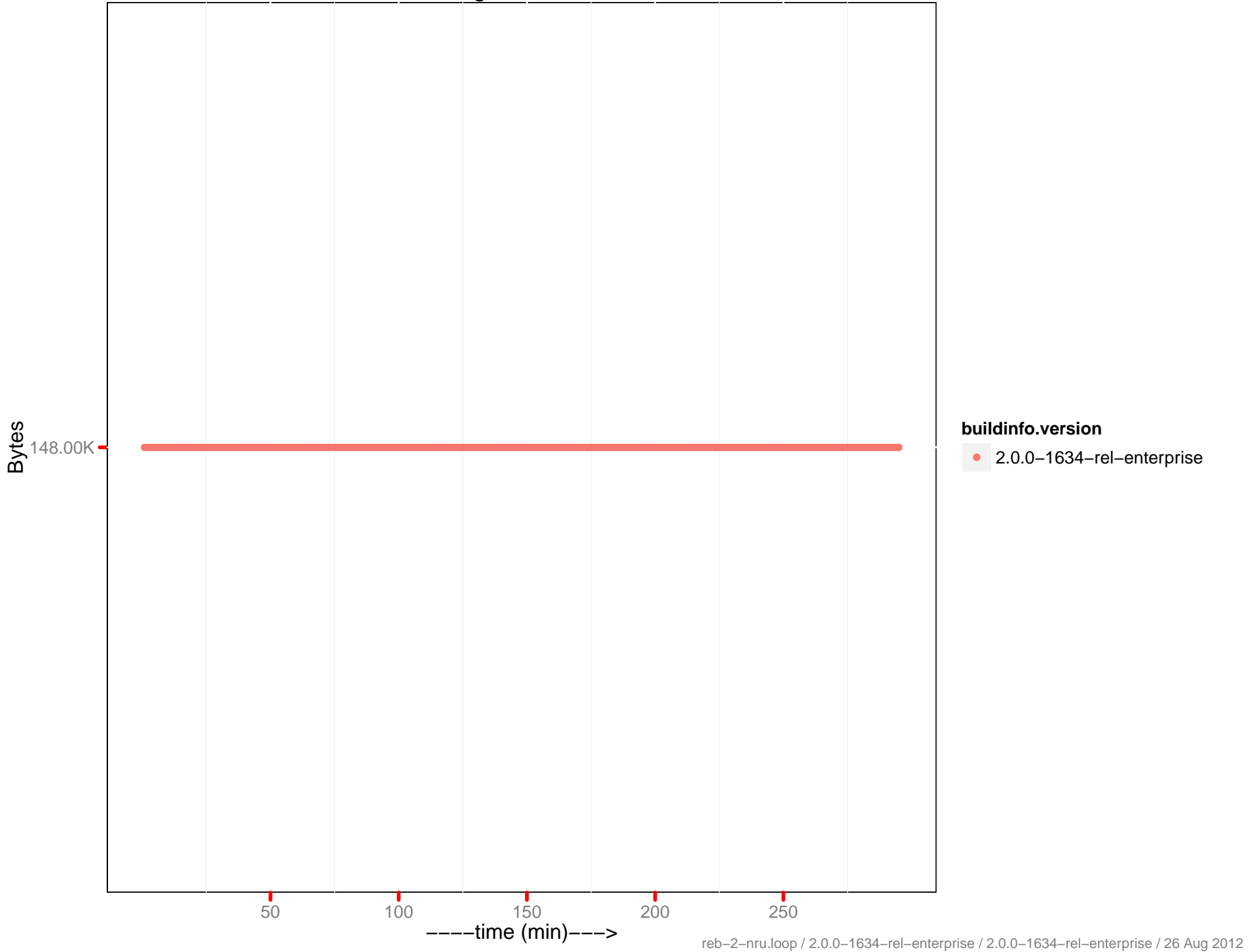
# SWAP Usage - 10.2.1.65:8091



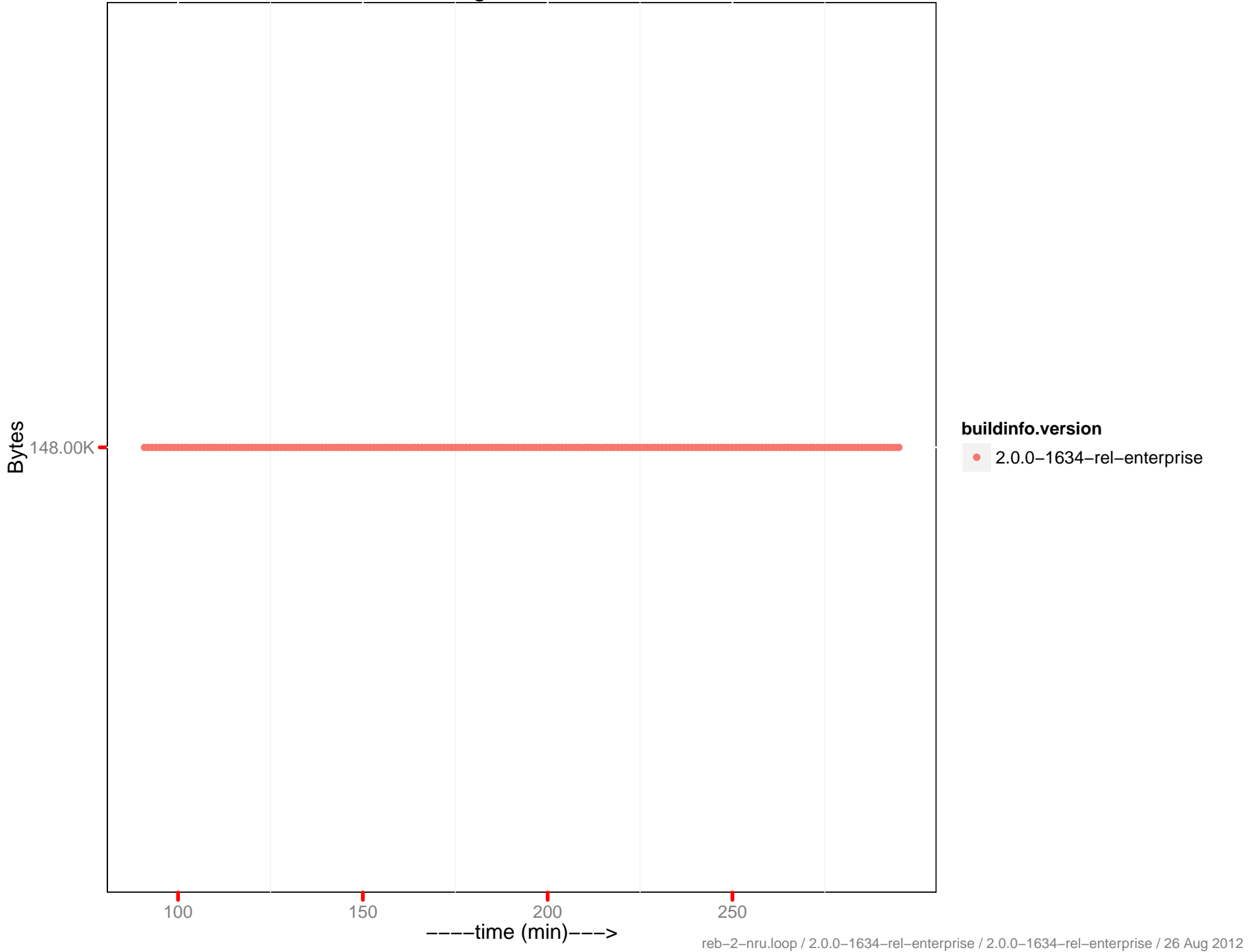
# SWAP Usage - 10.2.1.66:8091



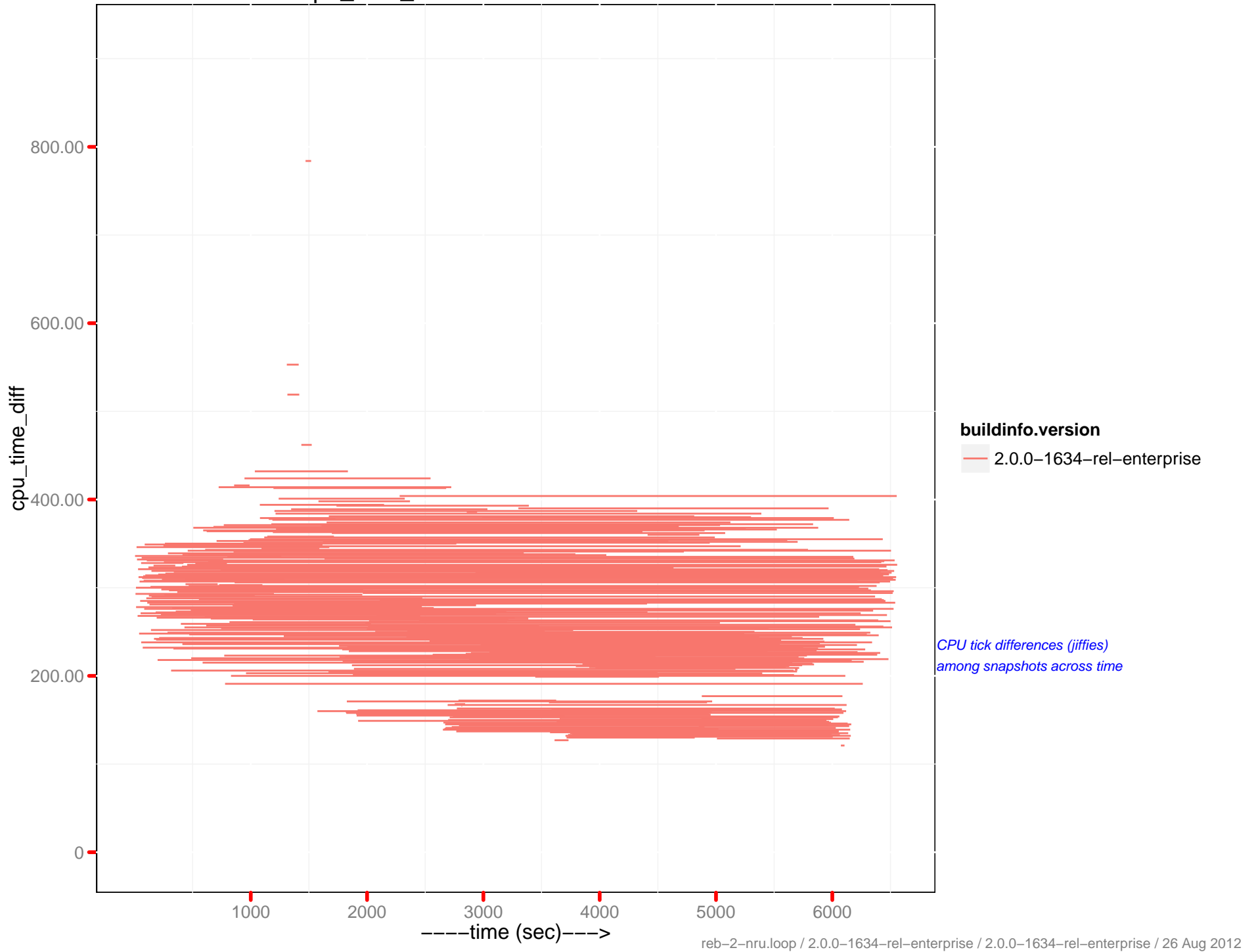
# SWAP Usage - 10.2.1.67:8091



# SWAP Usage - 10.2.1.68:8091

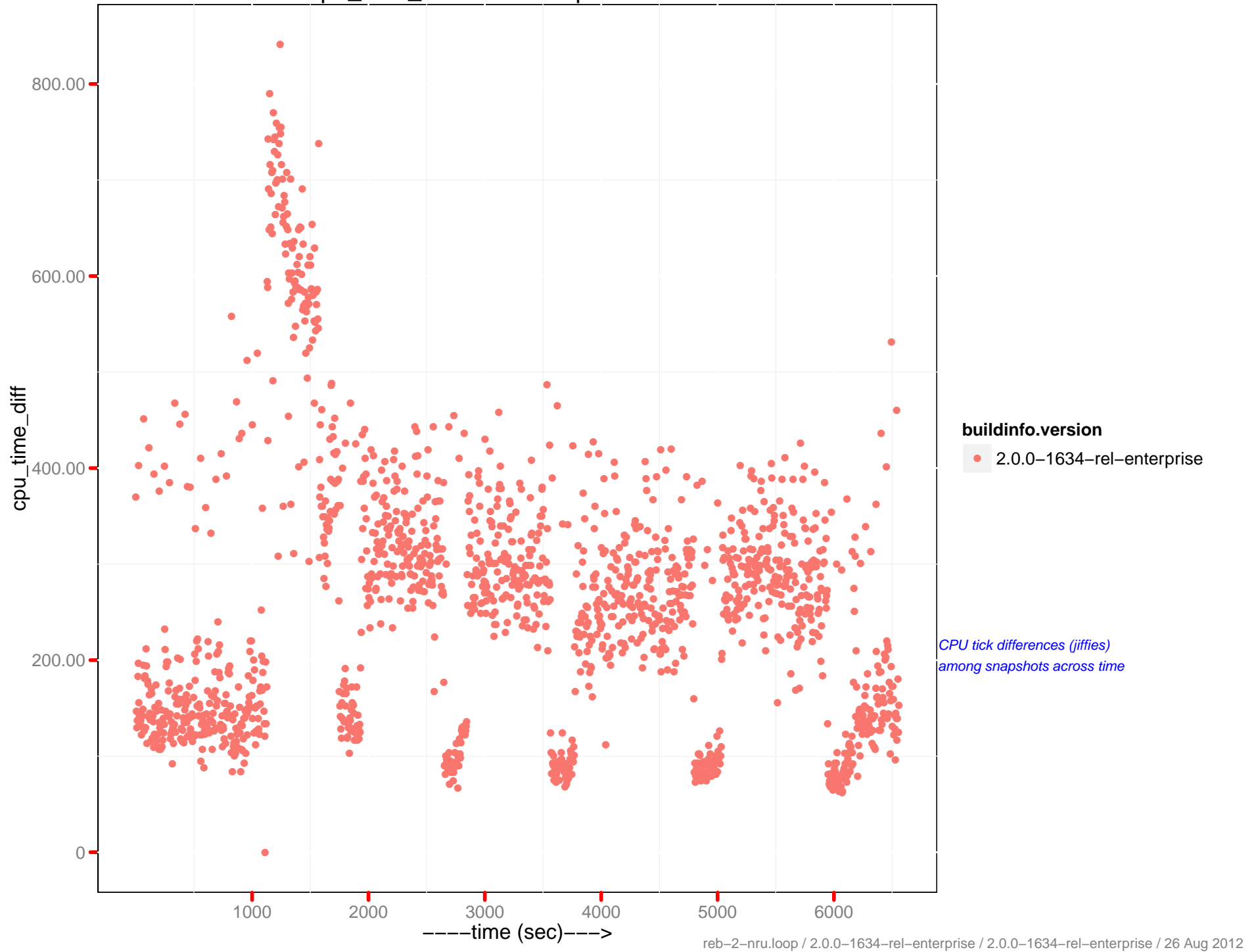


# cpu\_time\_diff: memcached - 10.2.1.65

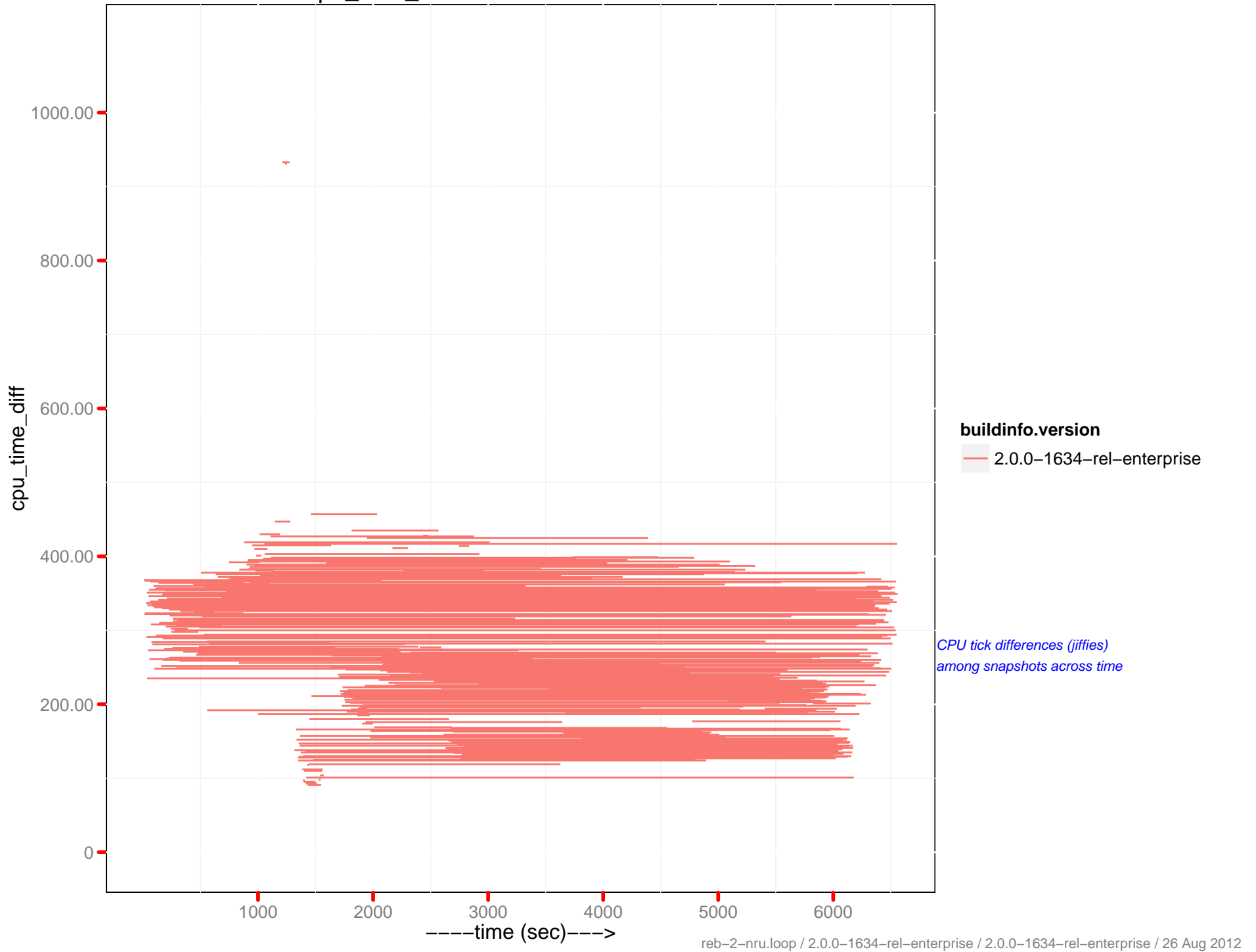




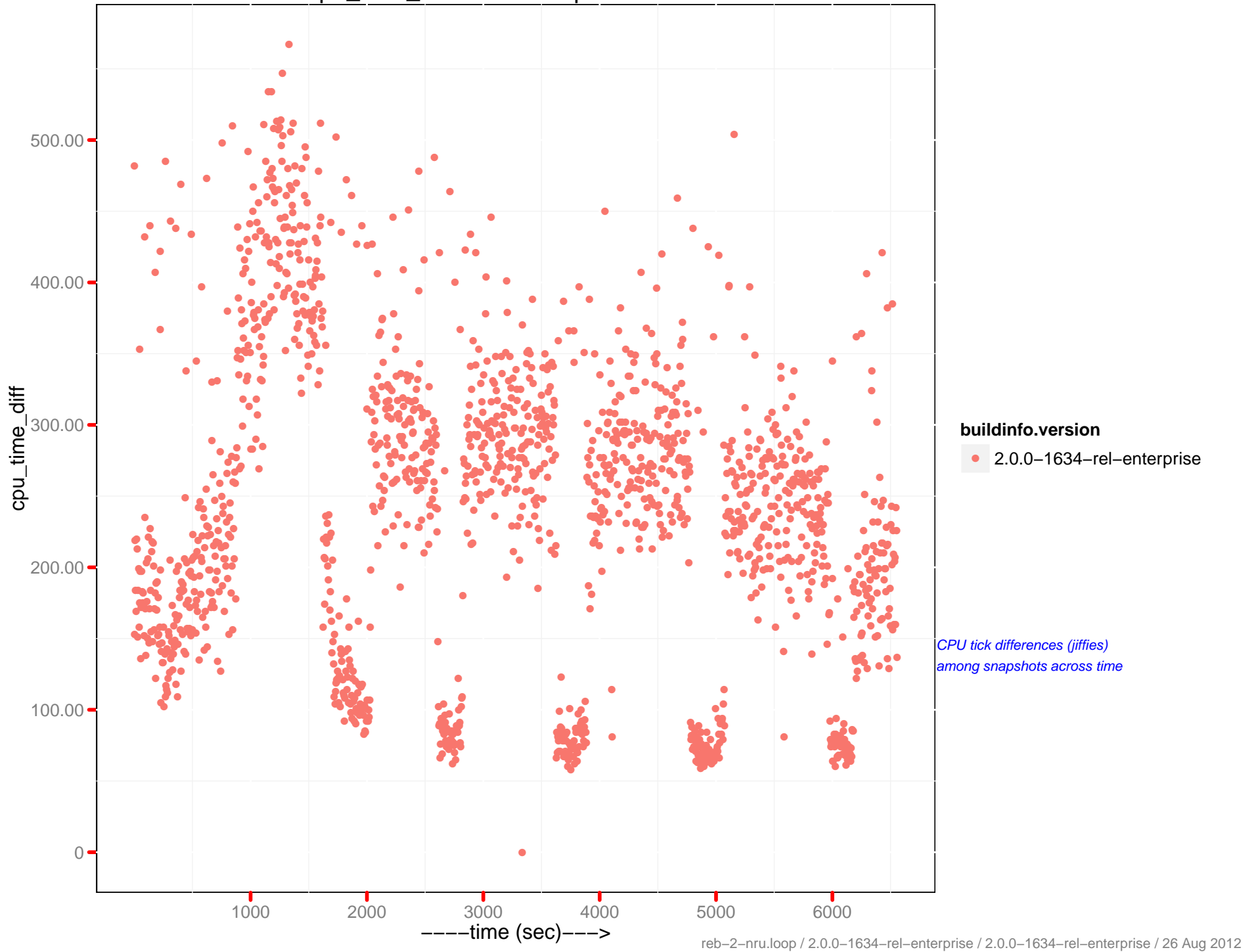
# cpu\_time\_diff : beam.smp - 10.2.1.65



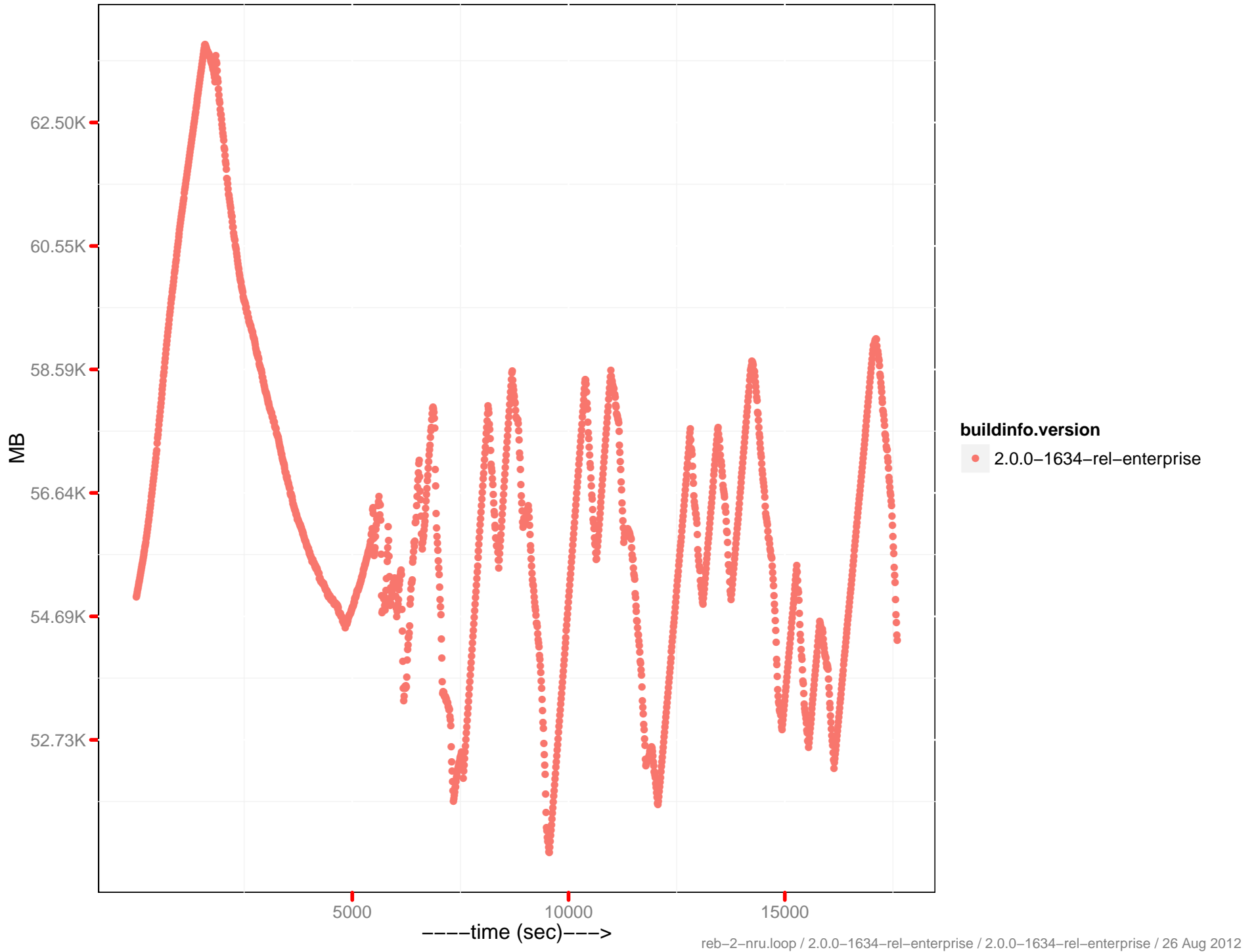
# cpu\_time\_diff: memcached – 10.2.1.67



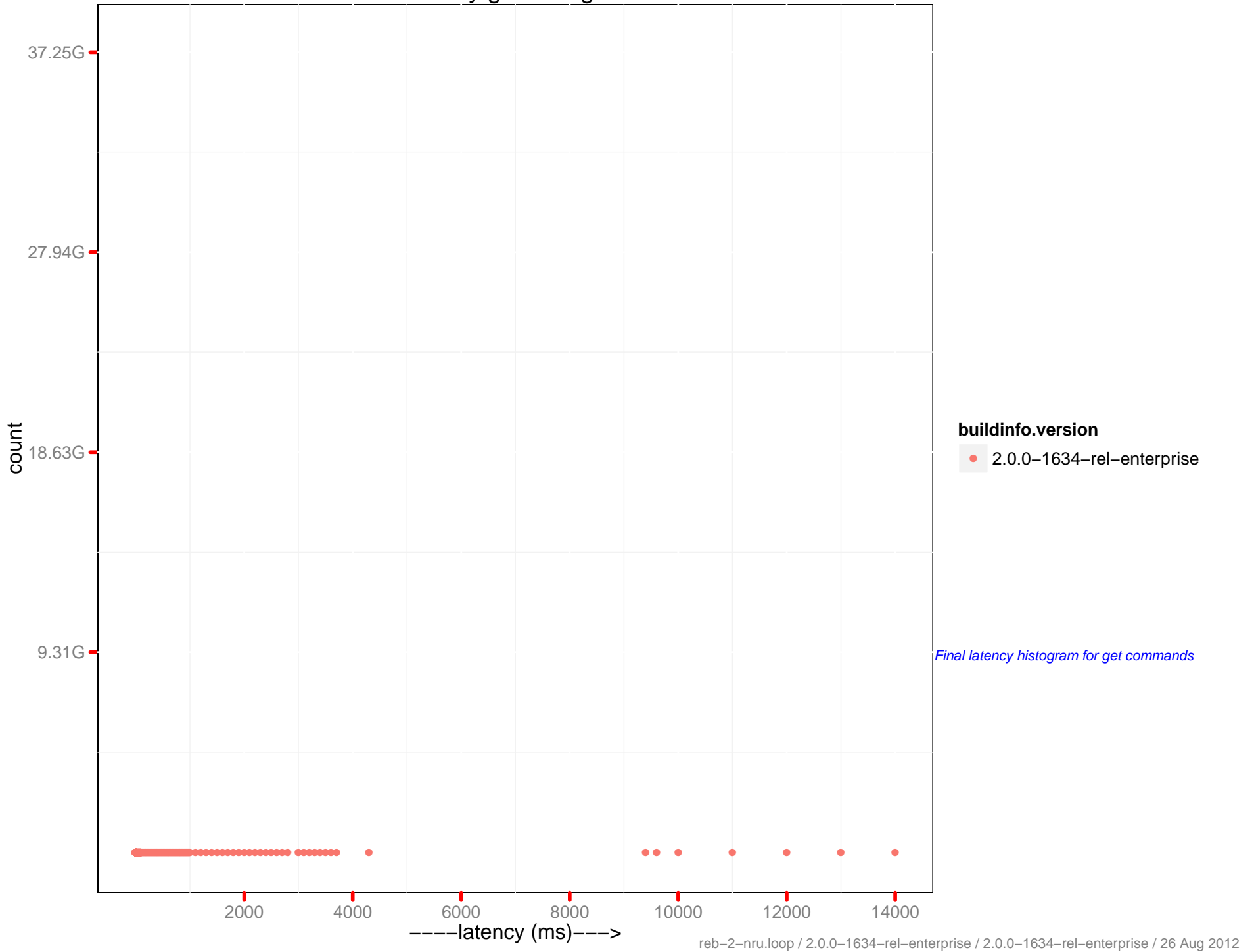
# cpu\_time\_diff : beam.smp - 10.2.1.67



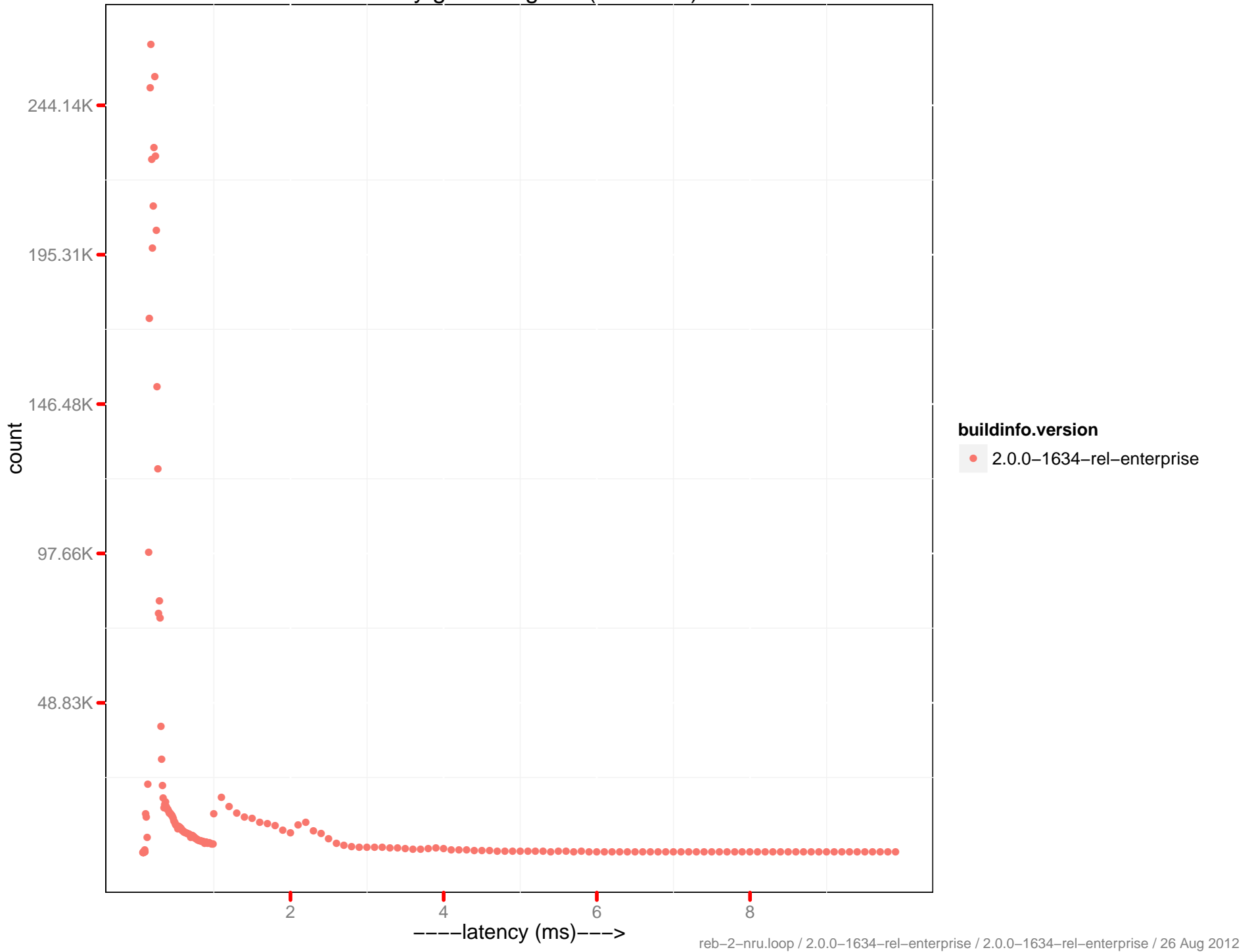
# Data disk size



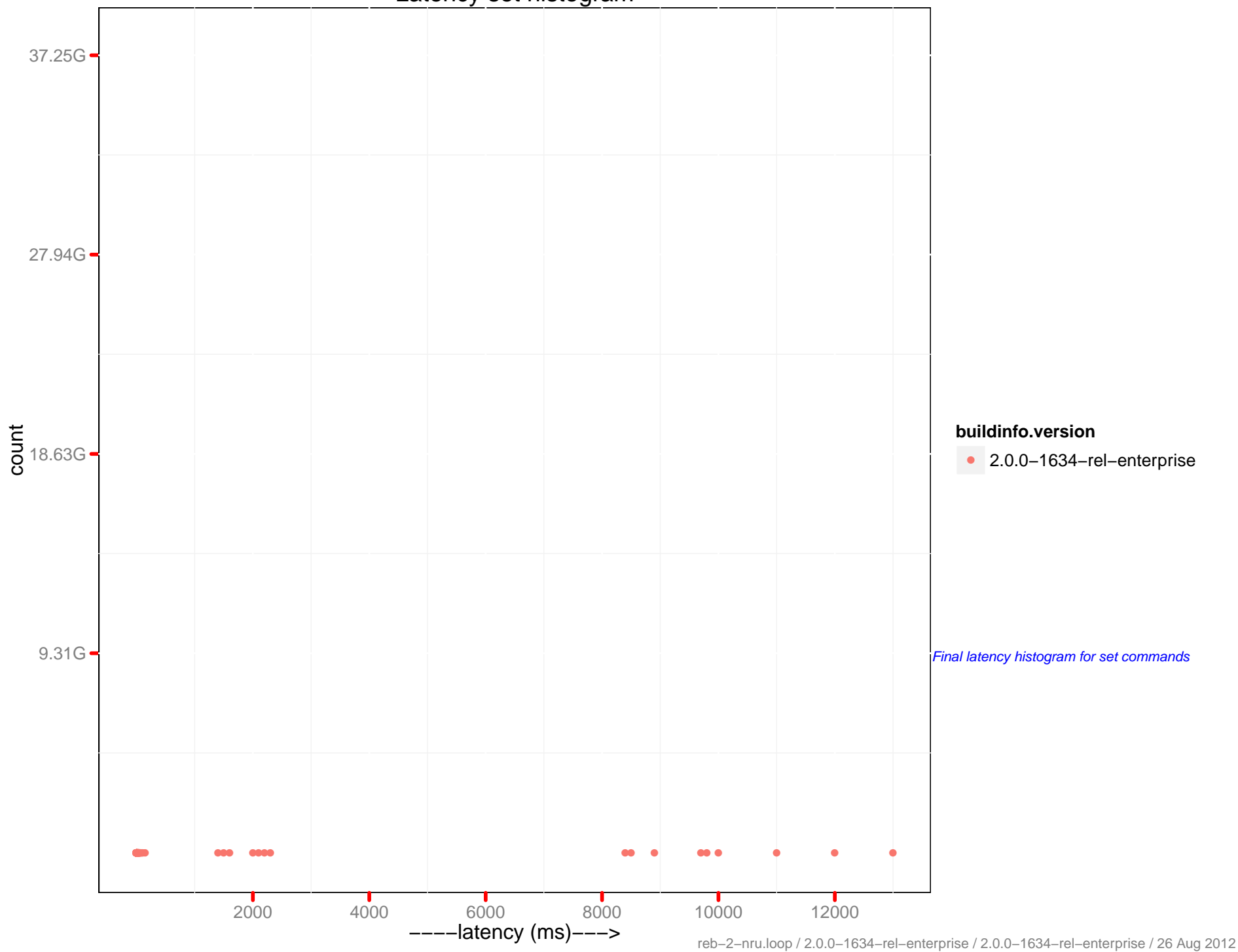
# Latency get histogram



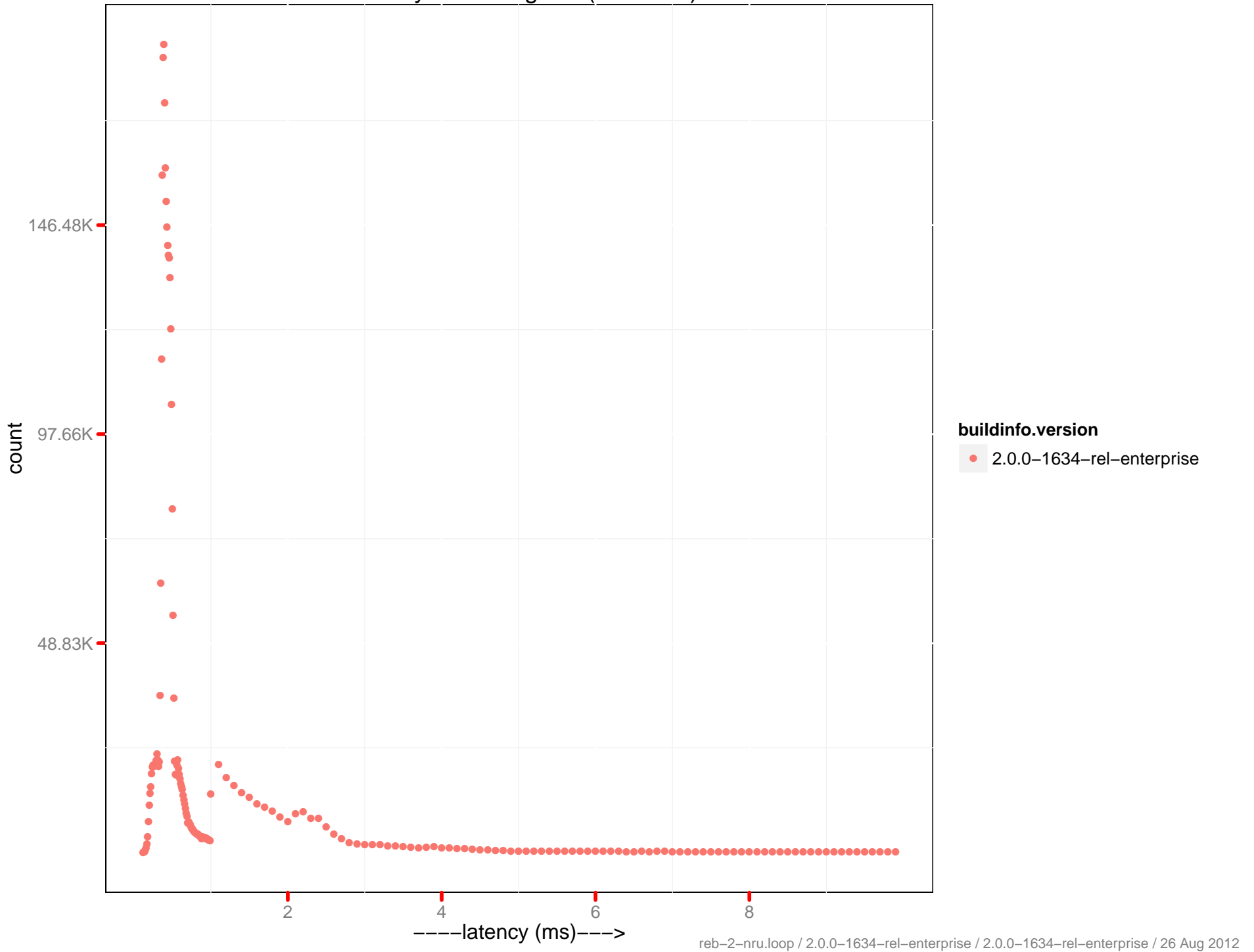
Latency get histogram (0-10 ms)



# Latency set histogram

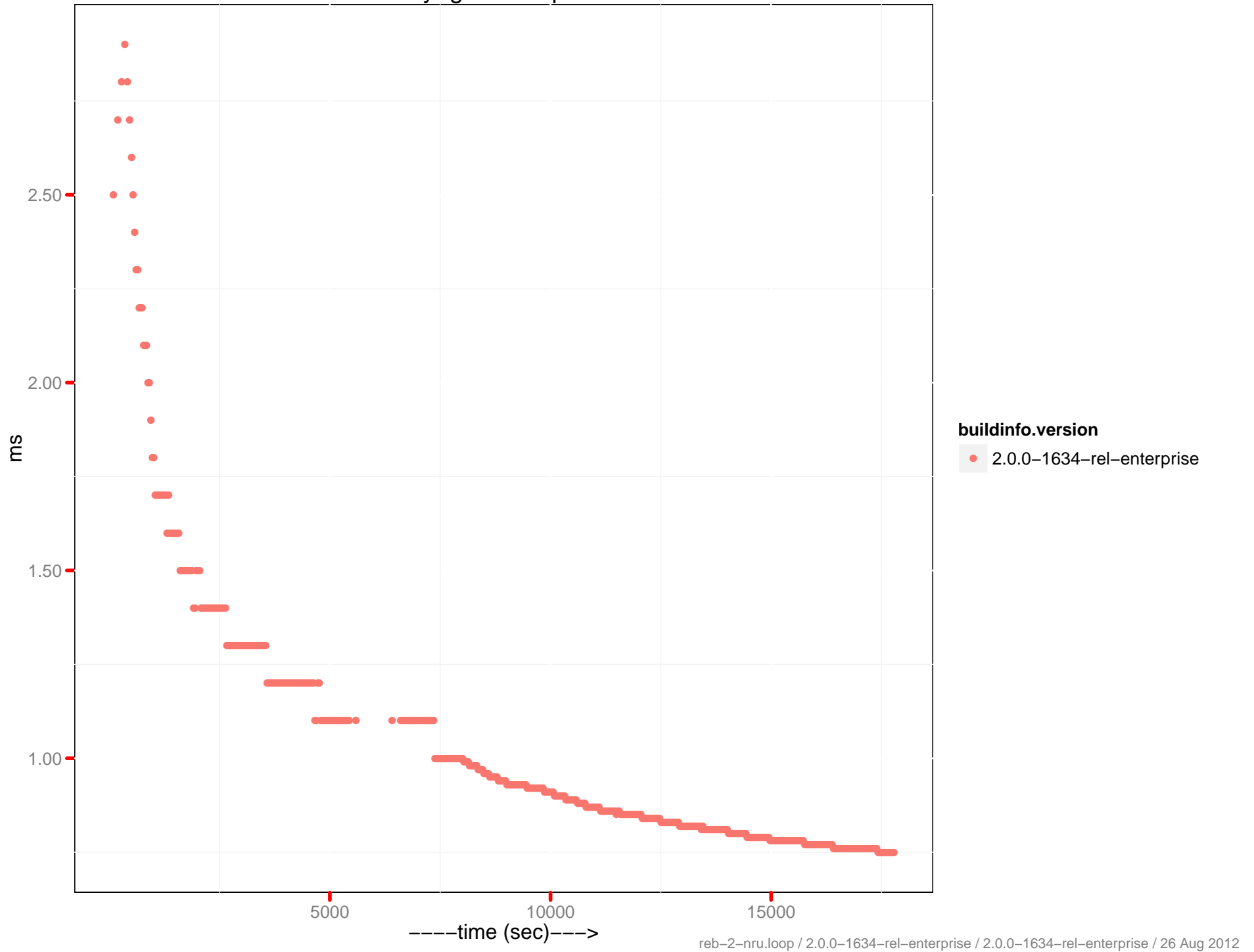


Latency set histogram (0–10 ms)

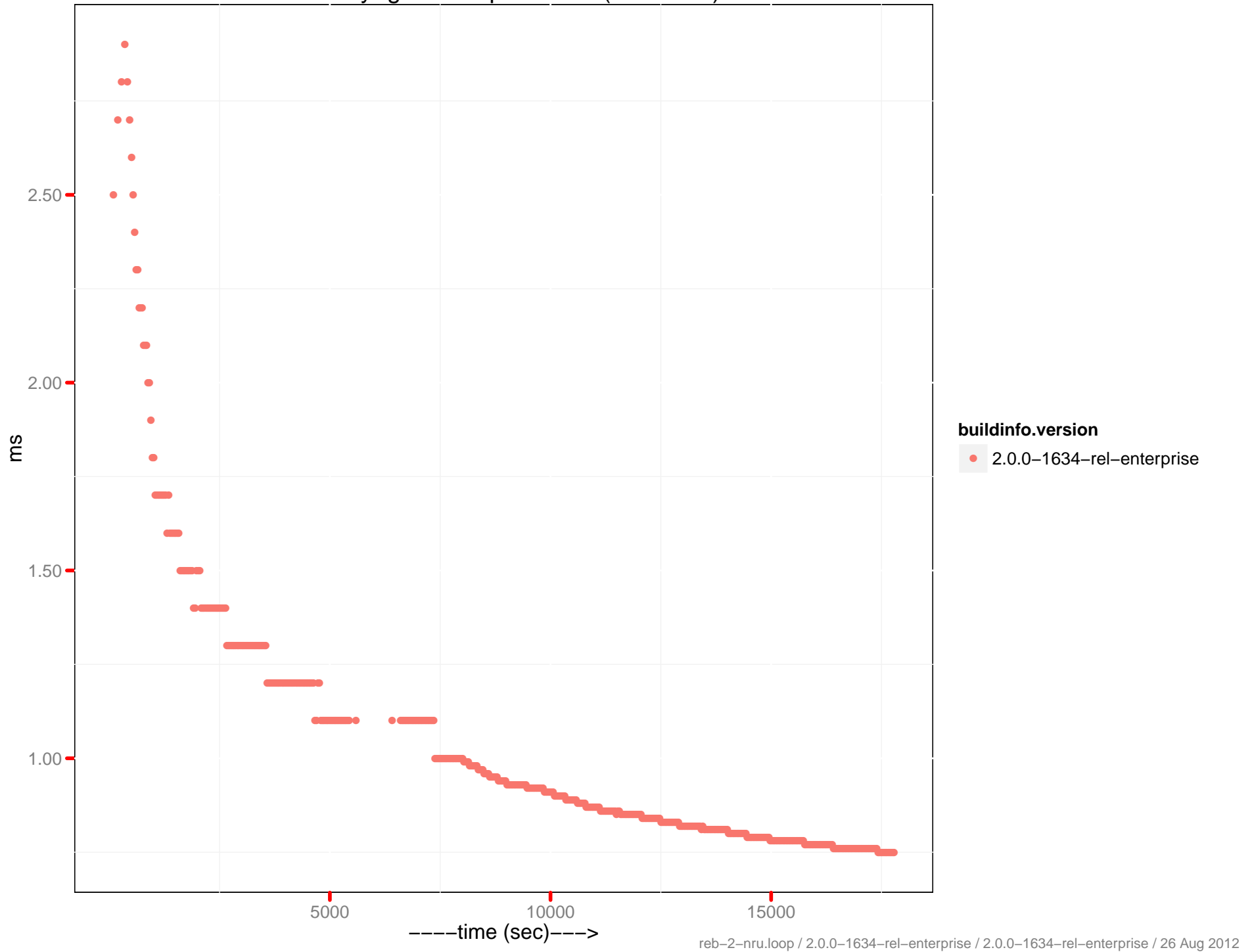




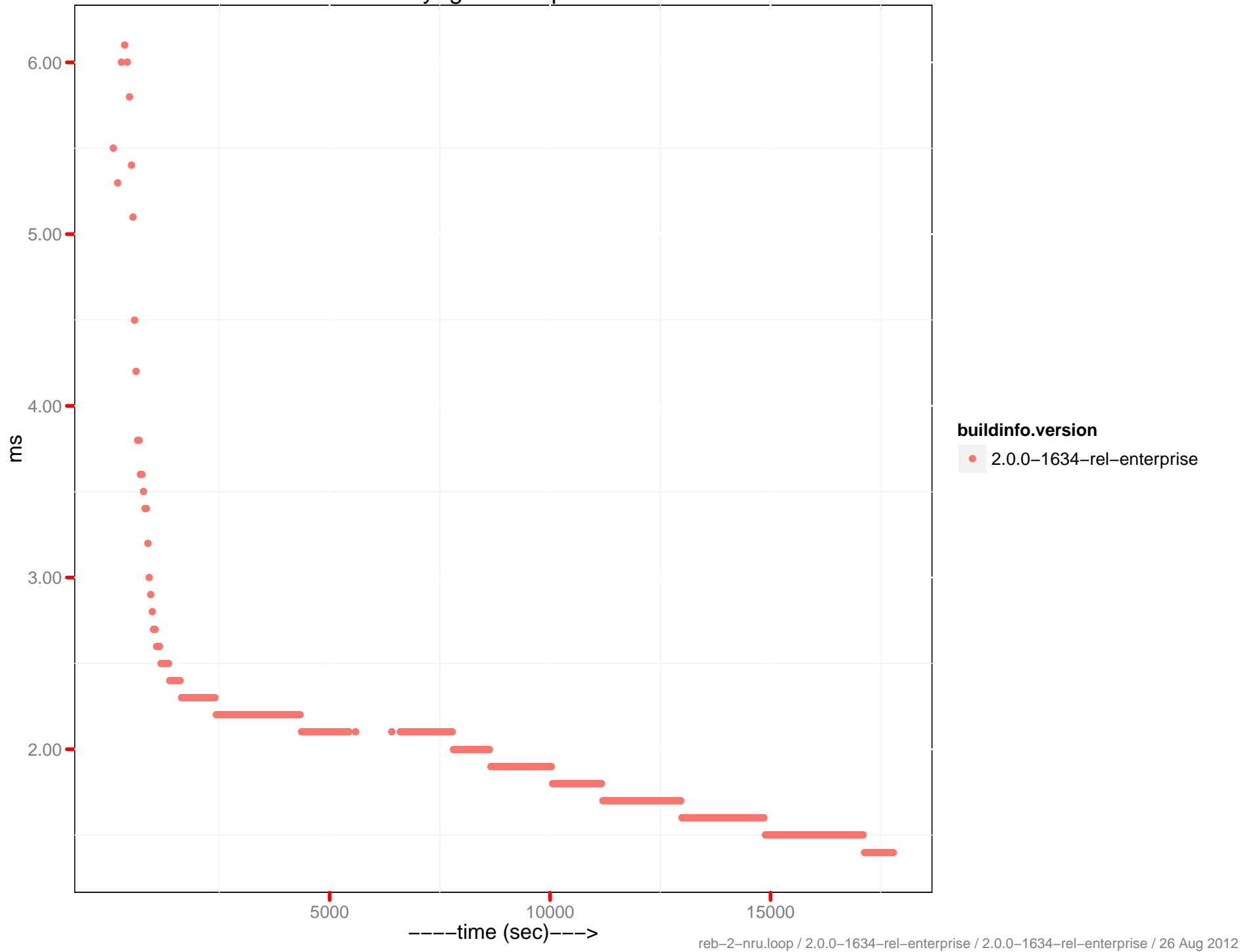
# Latency-get 90th percentile



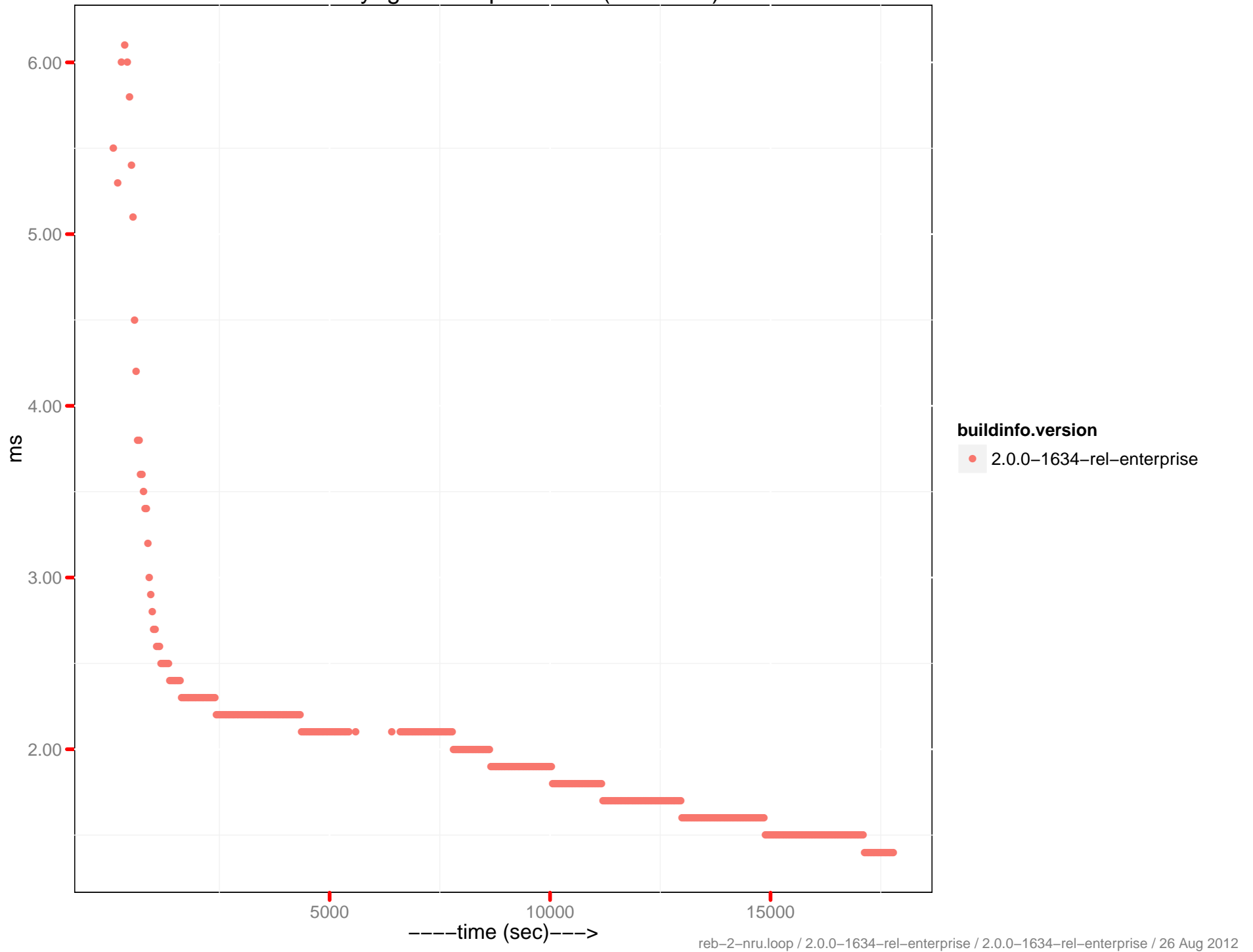
Latency-get 90th percentile (0 - 10ms)



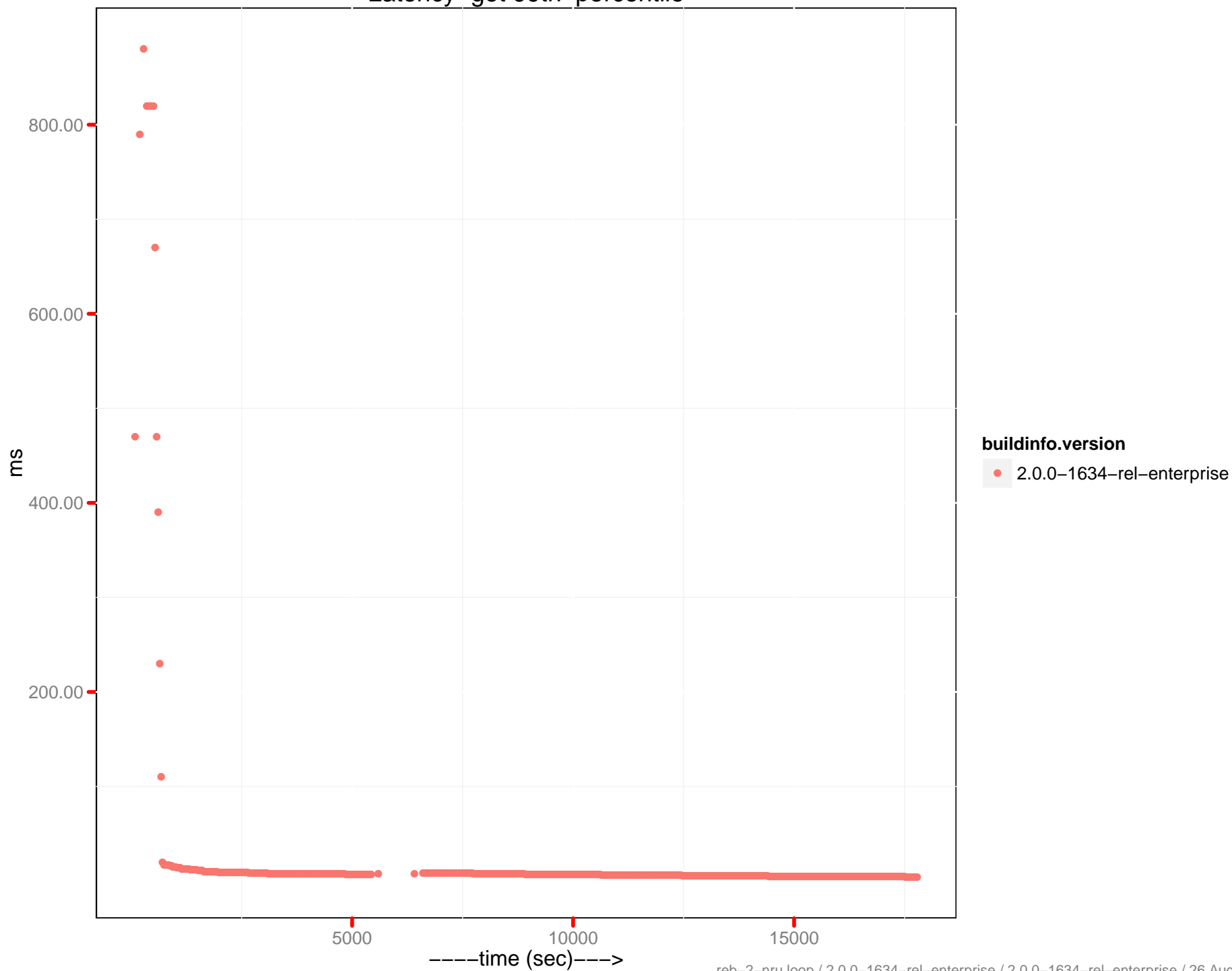
# Latency-get 95th percentile



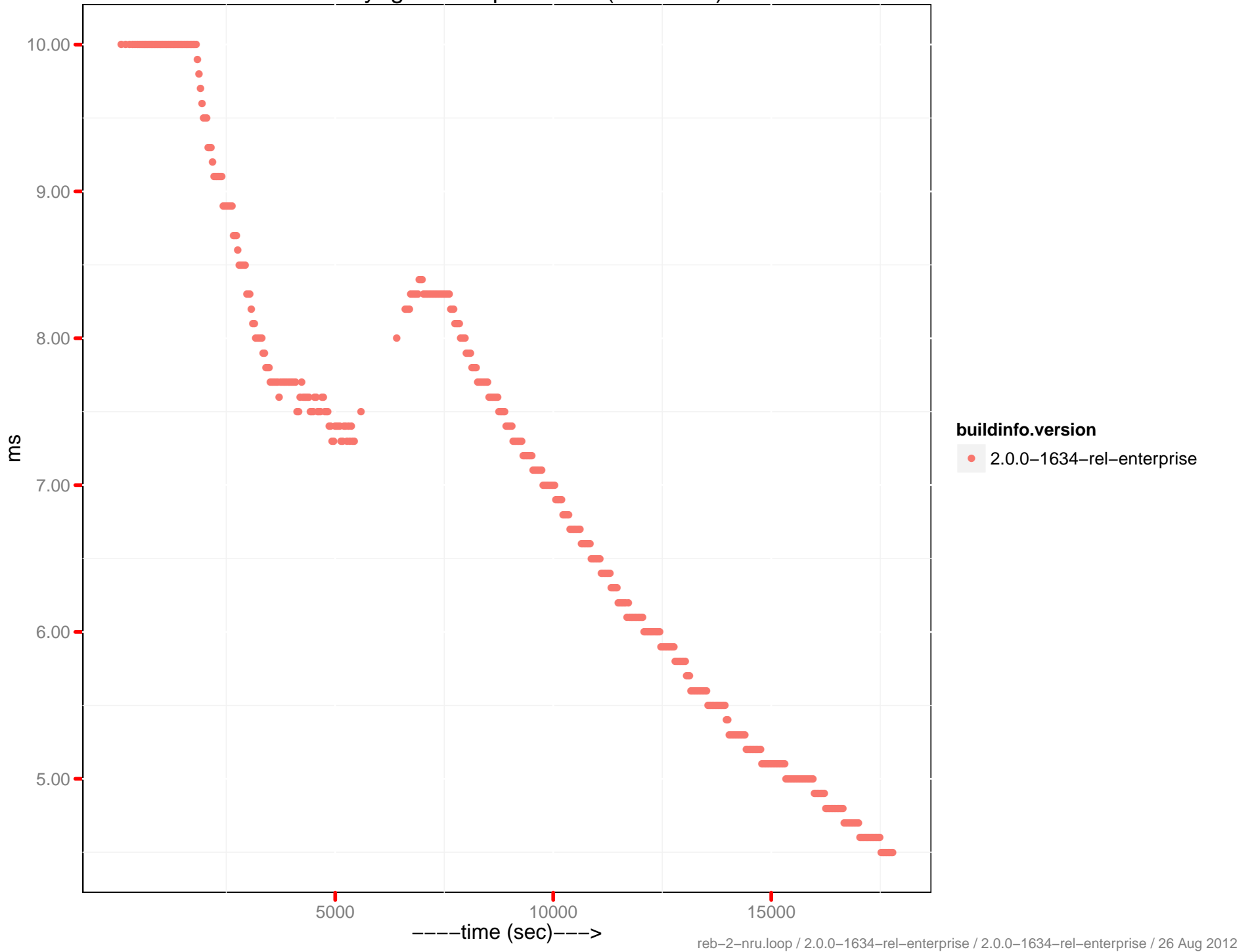
Latency-get 95th percentile (0 - 10ms)



# Latency-get 99th percentile



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

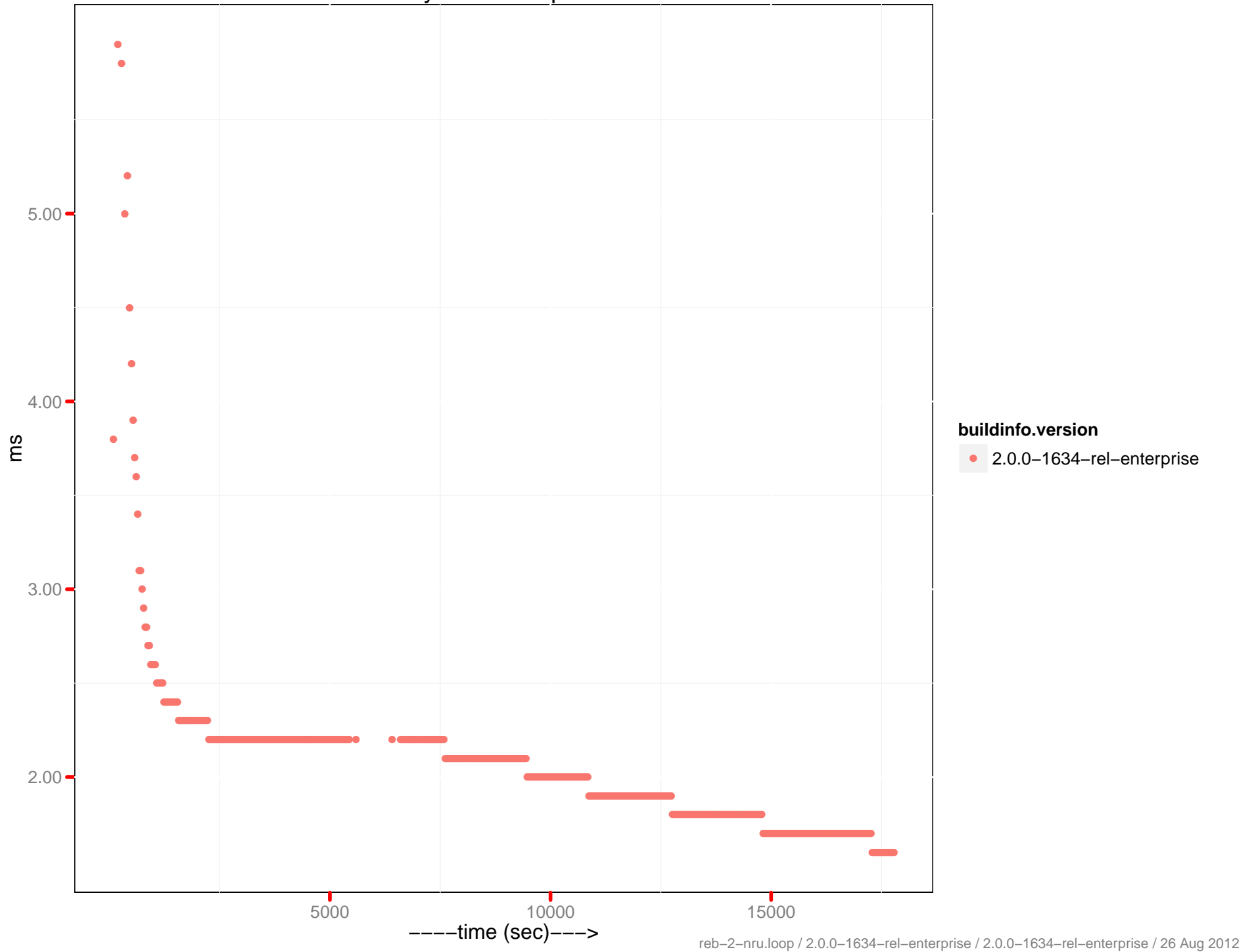




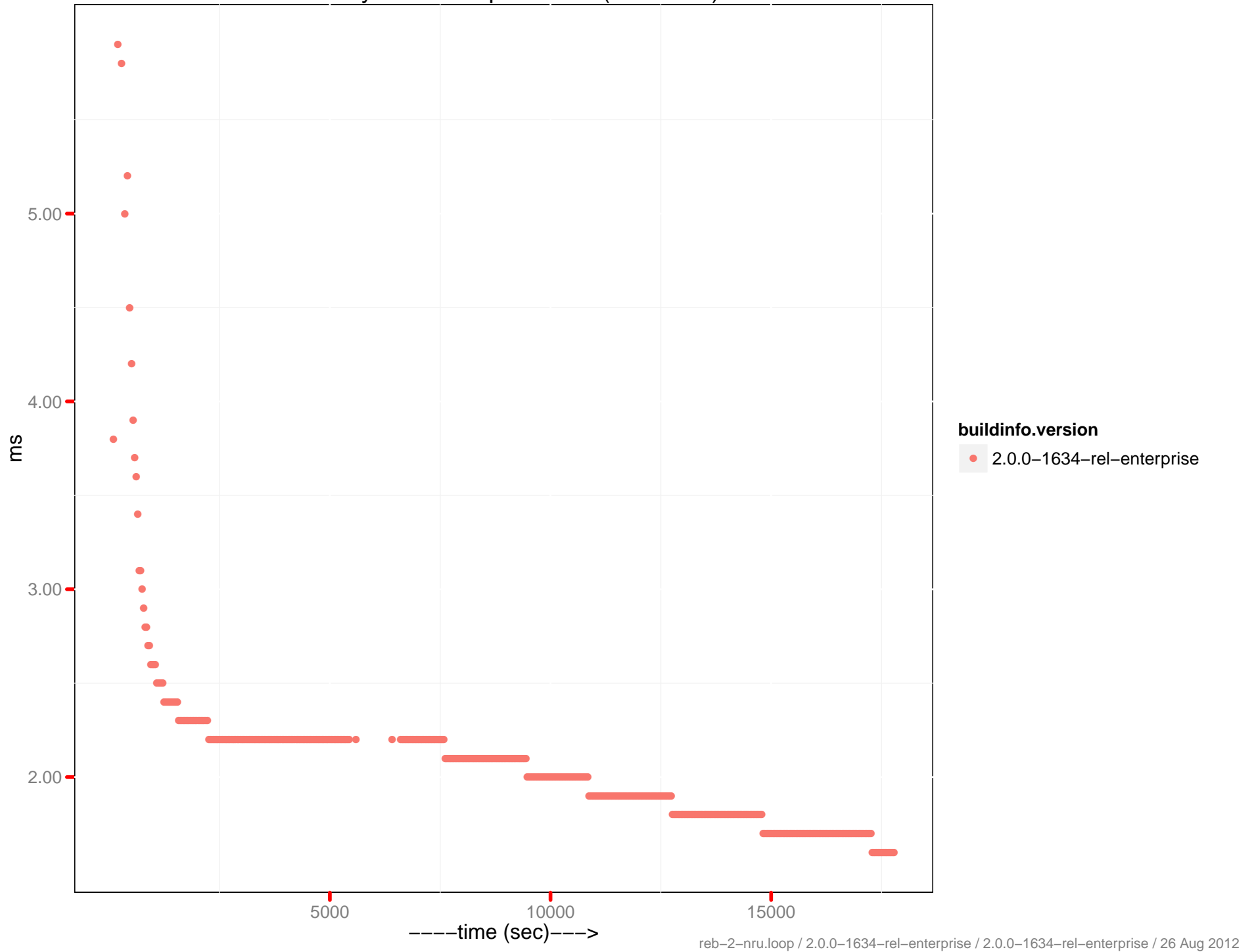




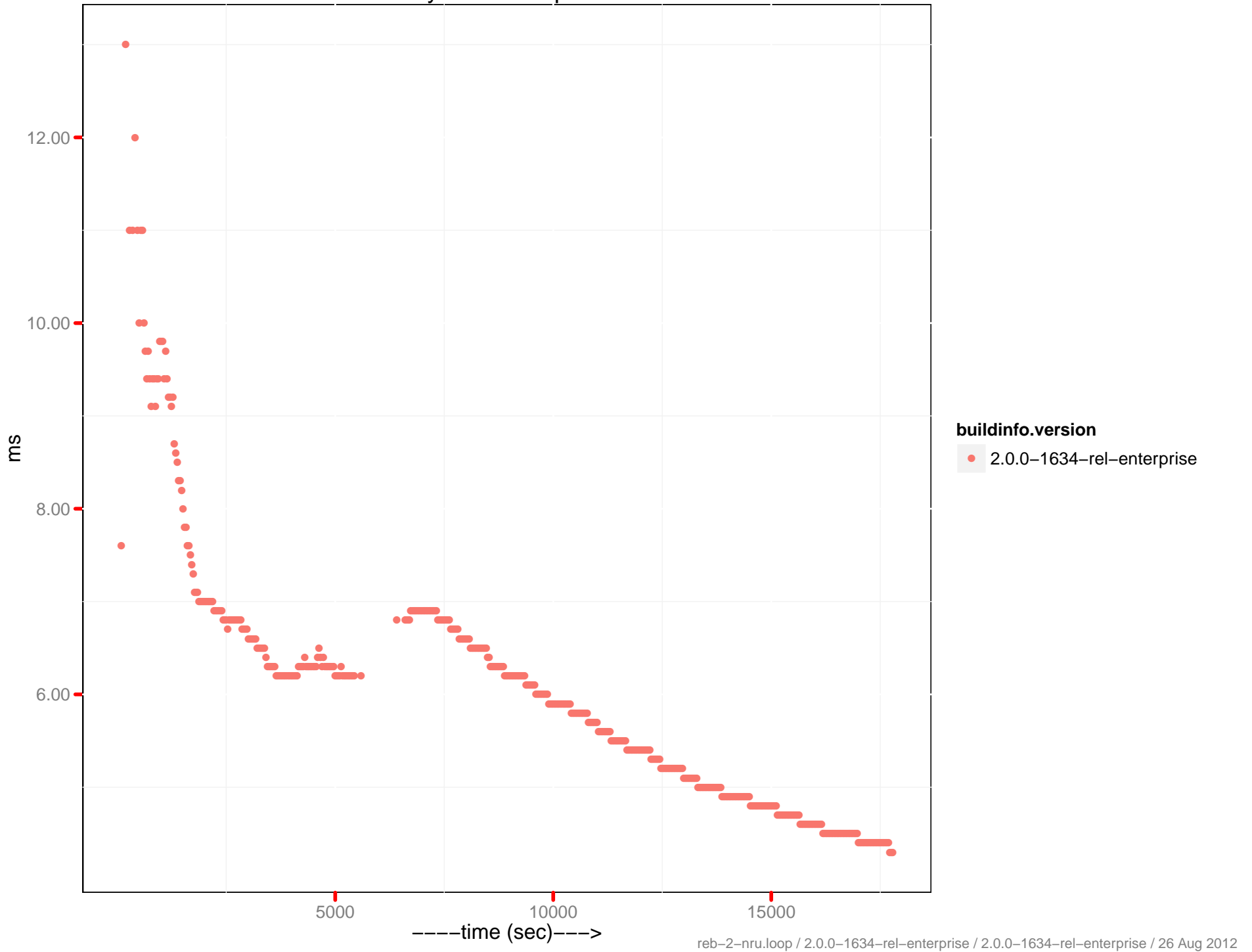
Latency-set 95th percentile



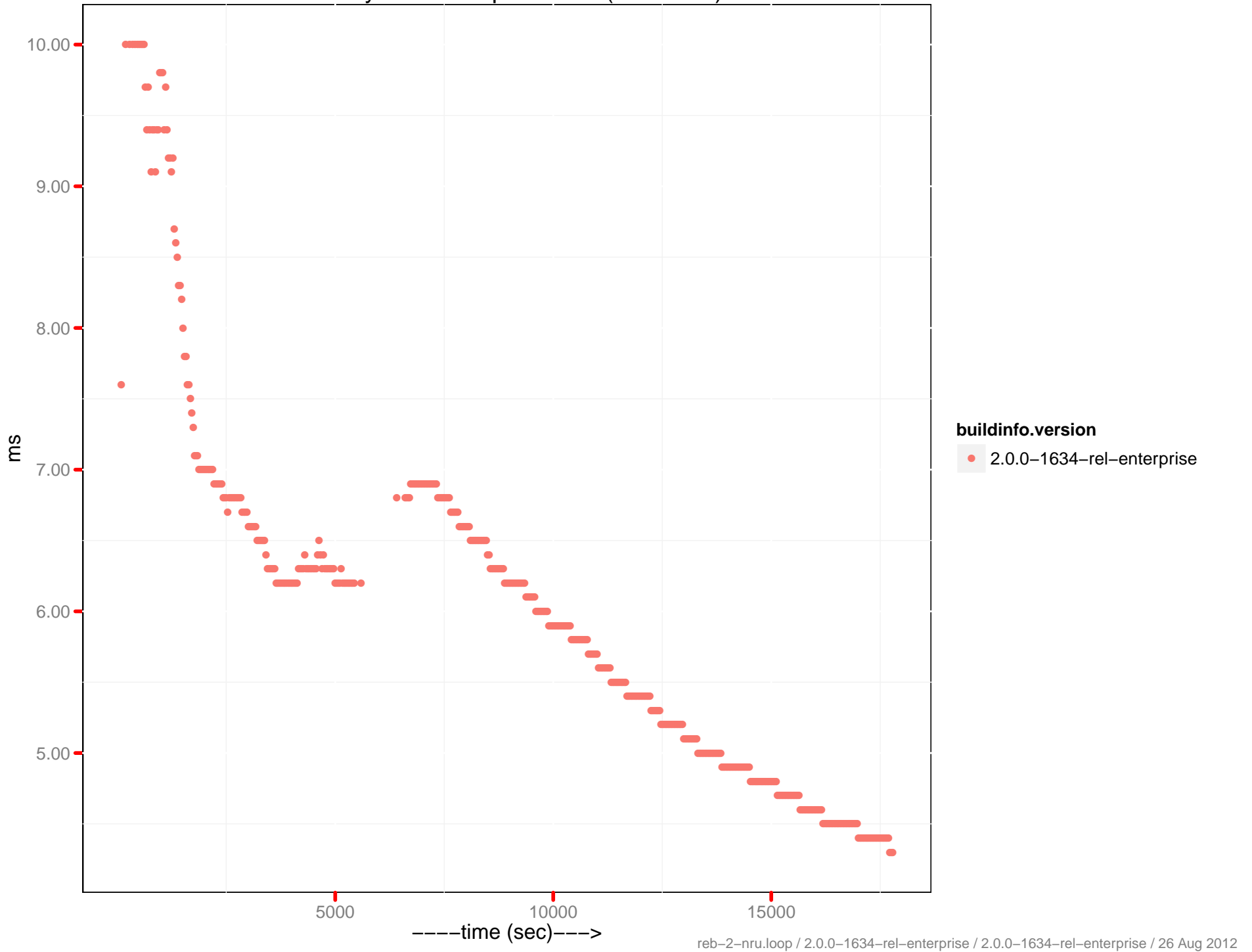
Latency-set 95th percentile (0 - 10ms)



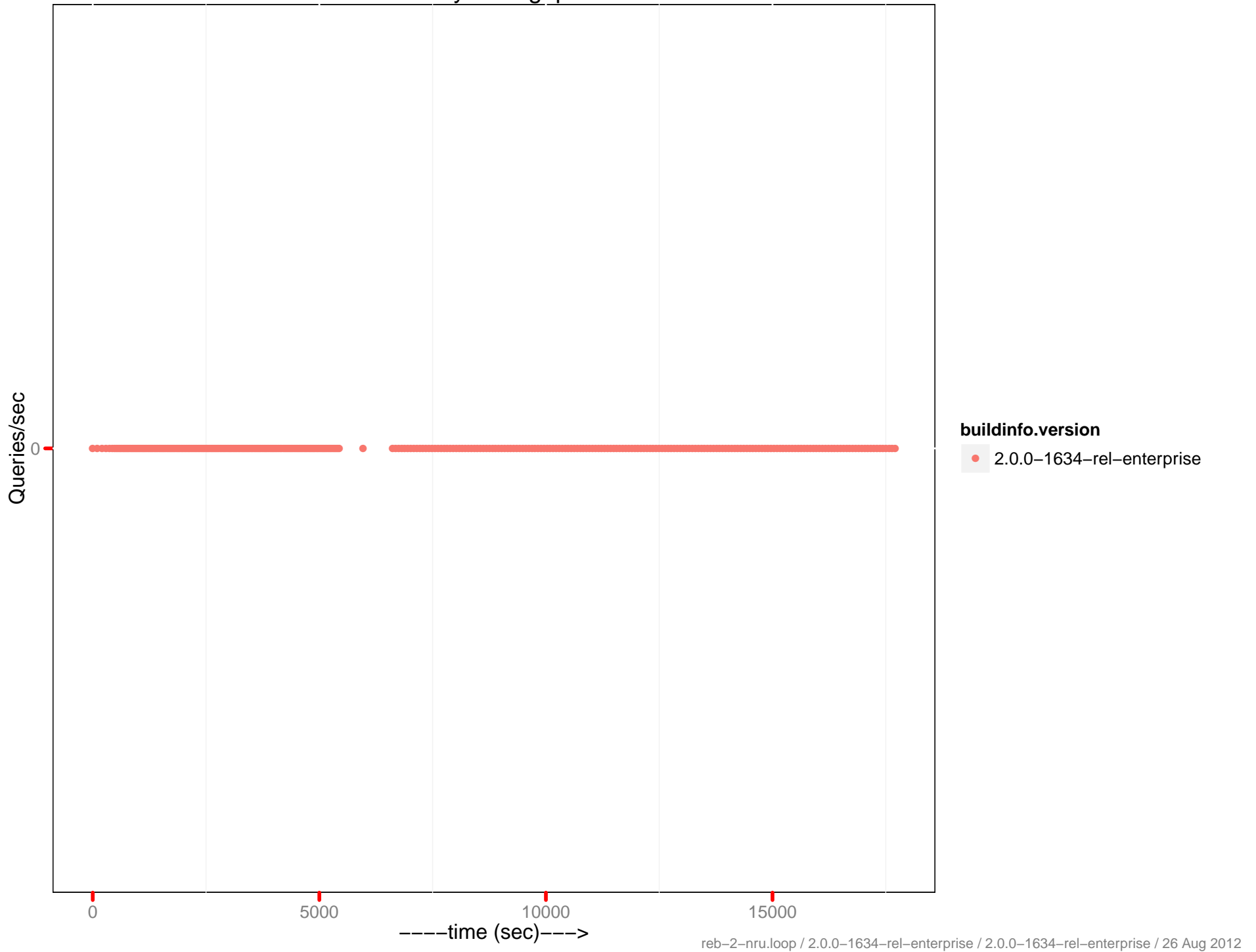
Latency-set 99th percentile



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



# Query throughput



```
reb-2-nru.conf
# rebalance mixed 10M load, 1M hot reload, 6M access creates
#
performance.eperf.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=6000000

# 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
load_wait_until_repl=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
```

```
vesta.ini
[global]
username:root
password:couchbase
port:8091
data_path:/data

[servers]
1:10.2.1.65
2:10.2.1.66
3:10.2.1.67
4:10.2.1.68

[clients]
1:10.2.1.60

[membase]
rest_username:Administrator
rest_password:password

[dashboard]
1:dashboard.hq.couchbase.com:80
```