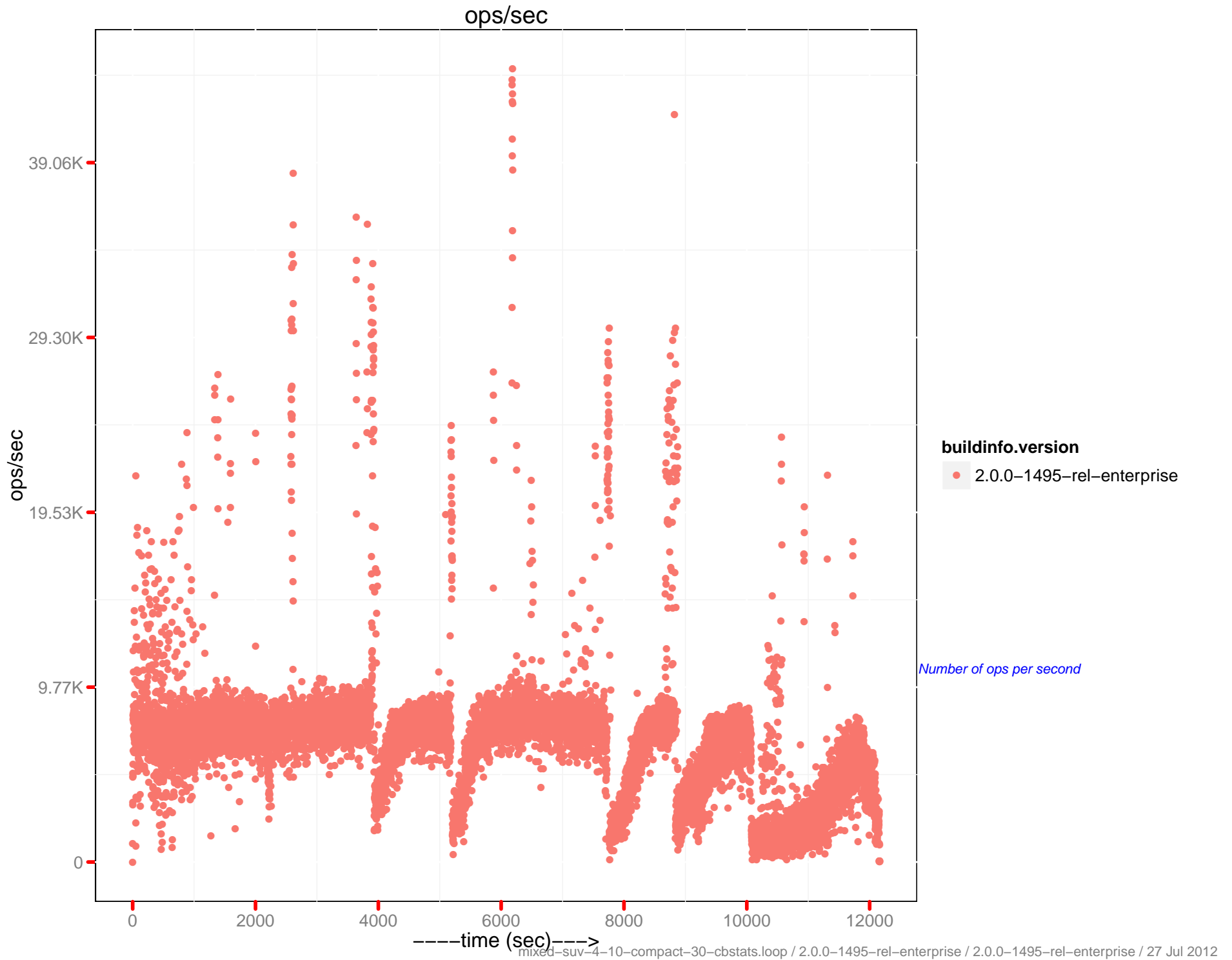
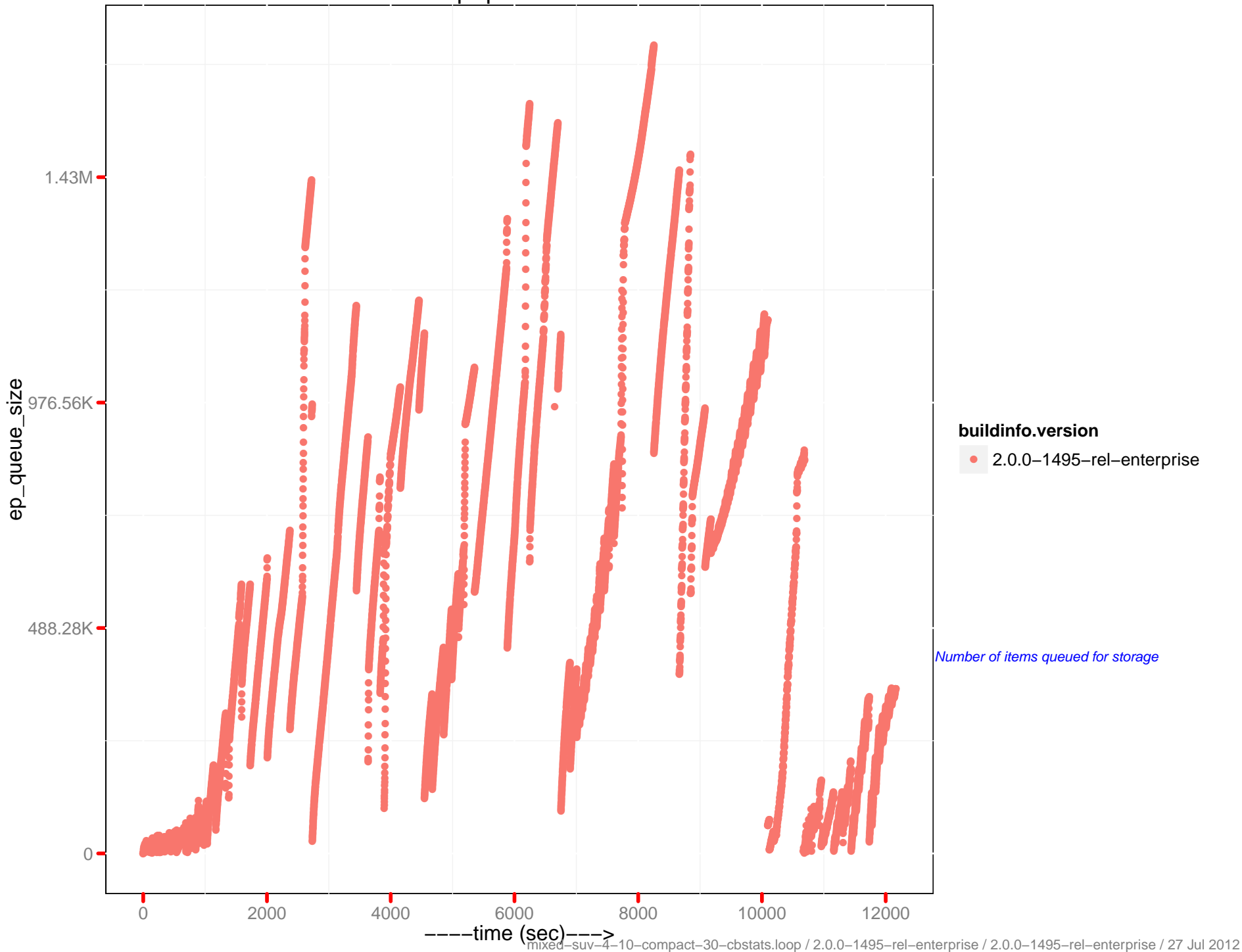


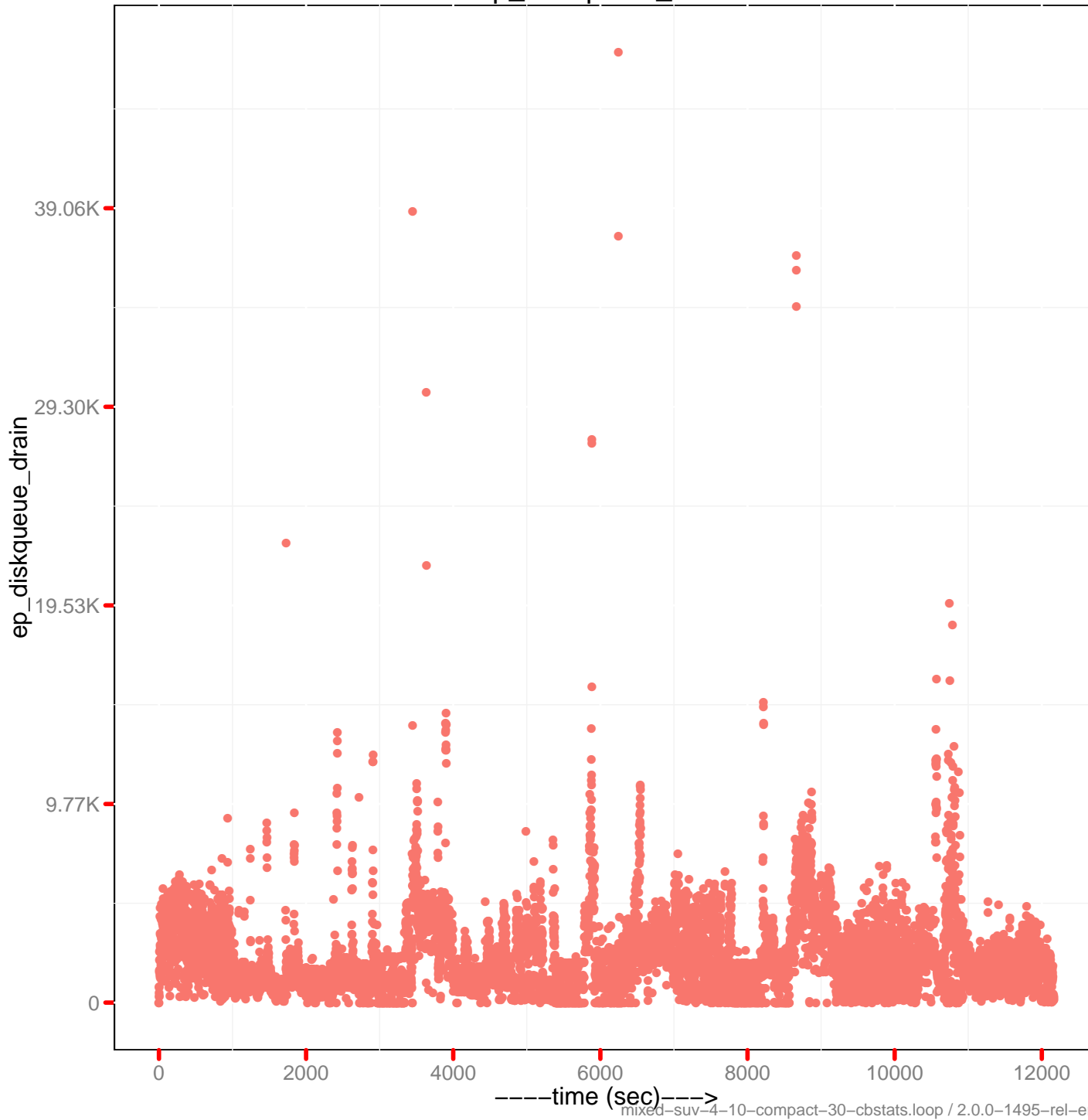
	2.0.0 – 1495	2.0.0 – 1495
<i>Runtime (in hr)</i>	3.39	NA
<i>Avg. Drain Rate</i>	2.27K	NANA
<i>Peak Disk (GB)</i>	48.76	NA
<i>Peak Memory (GB)</i>	16.83	NA
<i>Avg. OPS</i>	6.76K	NANA
<i>Avg. mem memcached (GB)</i>	16.34	NA
<i>Avg. mem beam.smp (MB)</i>	333	NA
<i>Latency-get (90th) (ms)</i>	1.03	NA
<i>Latency-get (95th) (ms)</i>	1.52	NA
<i>Latency-get (99th) (ms)</i>	4.38	NA
<i>Latency-set (90th) (ms)</i>	1.04	NA
<i>Latency-set (95th) (ms)</i>	1.55	NA
<i>Latency-set (99th) (ms)</i>	4.22	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>	0	NA
<i>Testrunner Version</i>	5f2a3cf	NA



# ep queue size



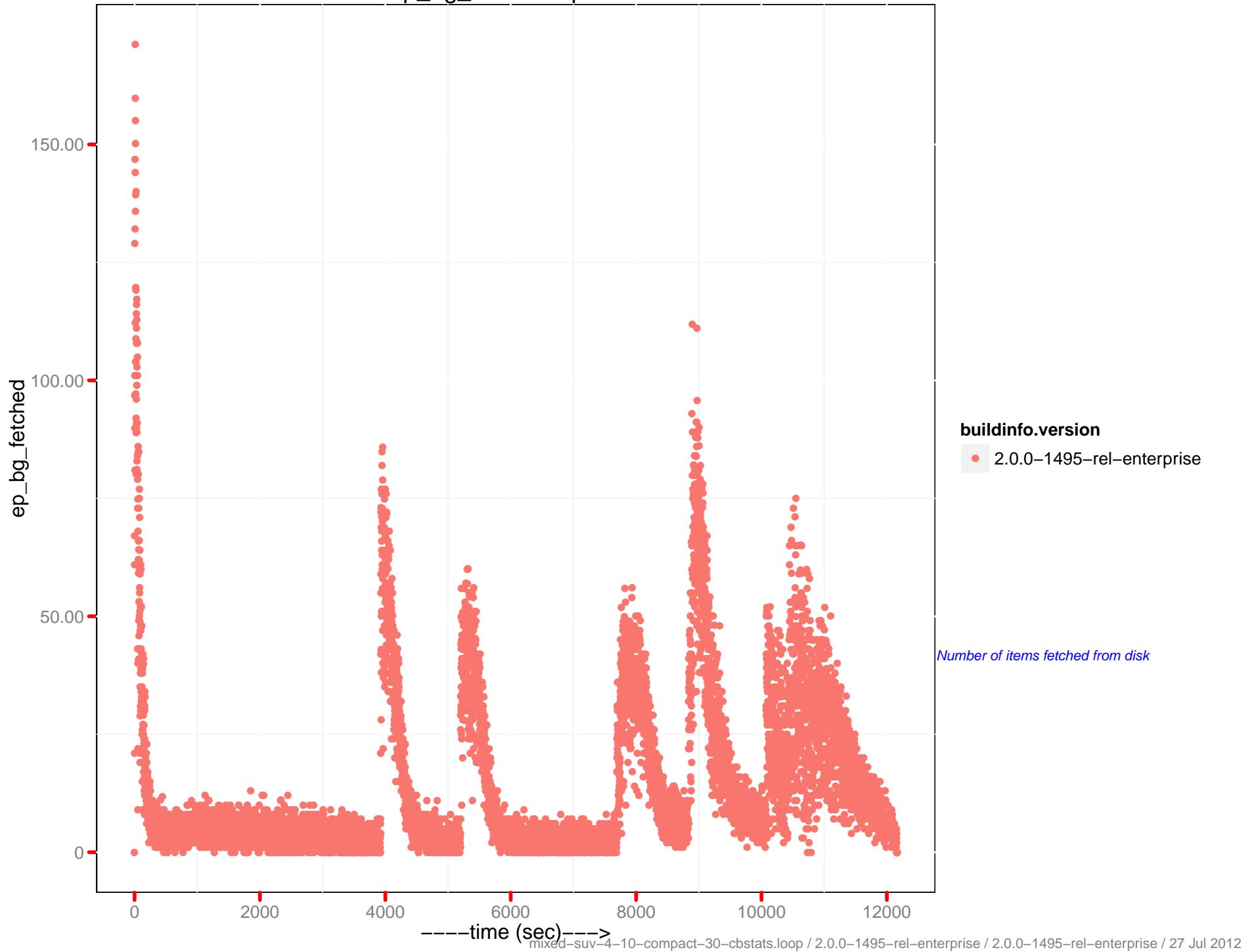
# ep\_diskqueue\_drain



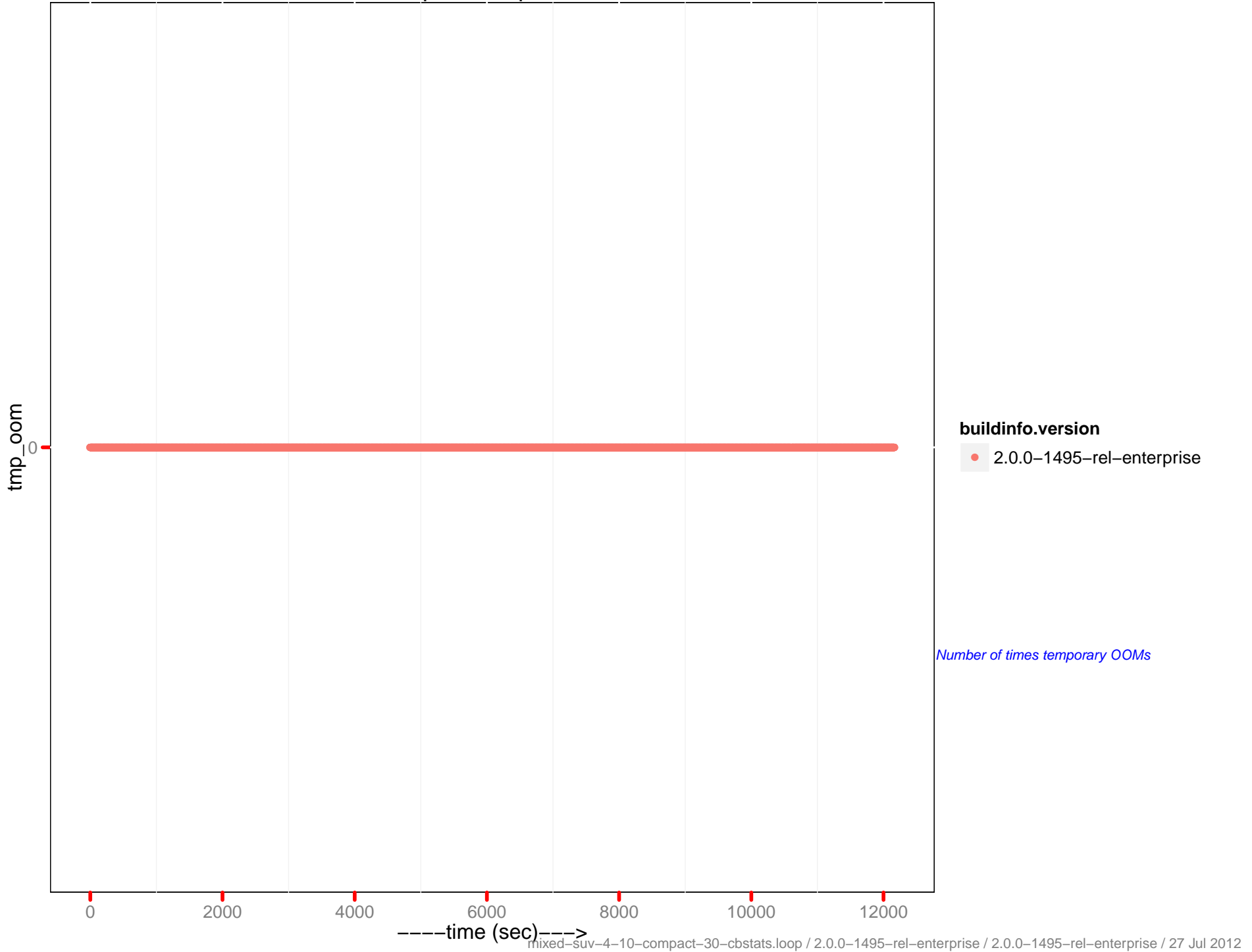
**buildinfo.version**  
● 2.0.0-1495-rel-enterprise

*Total drained items on disk queue*

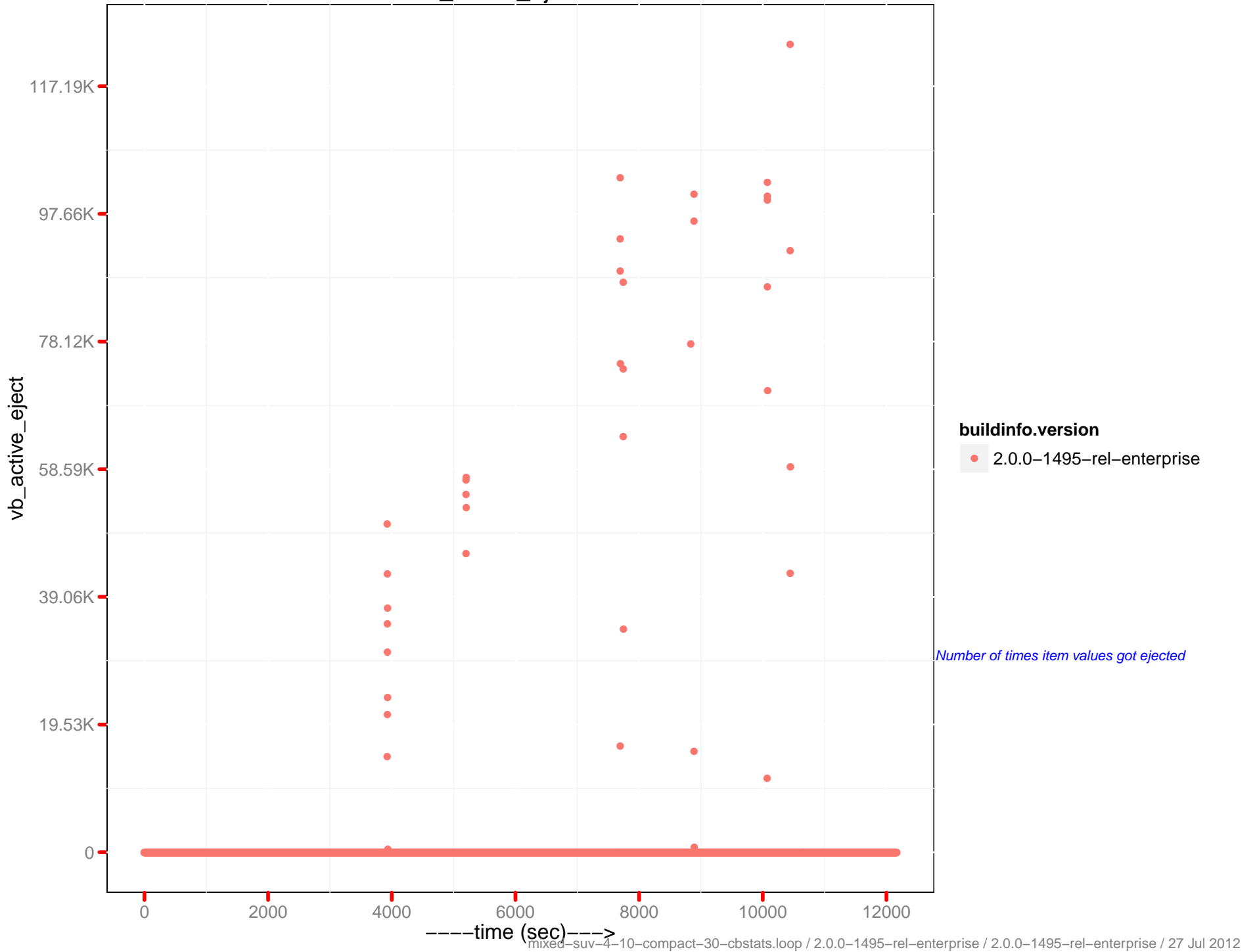
# ep\_bg\_fetched ops/sec



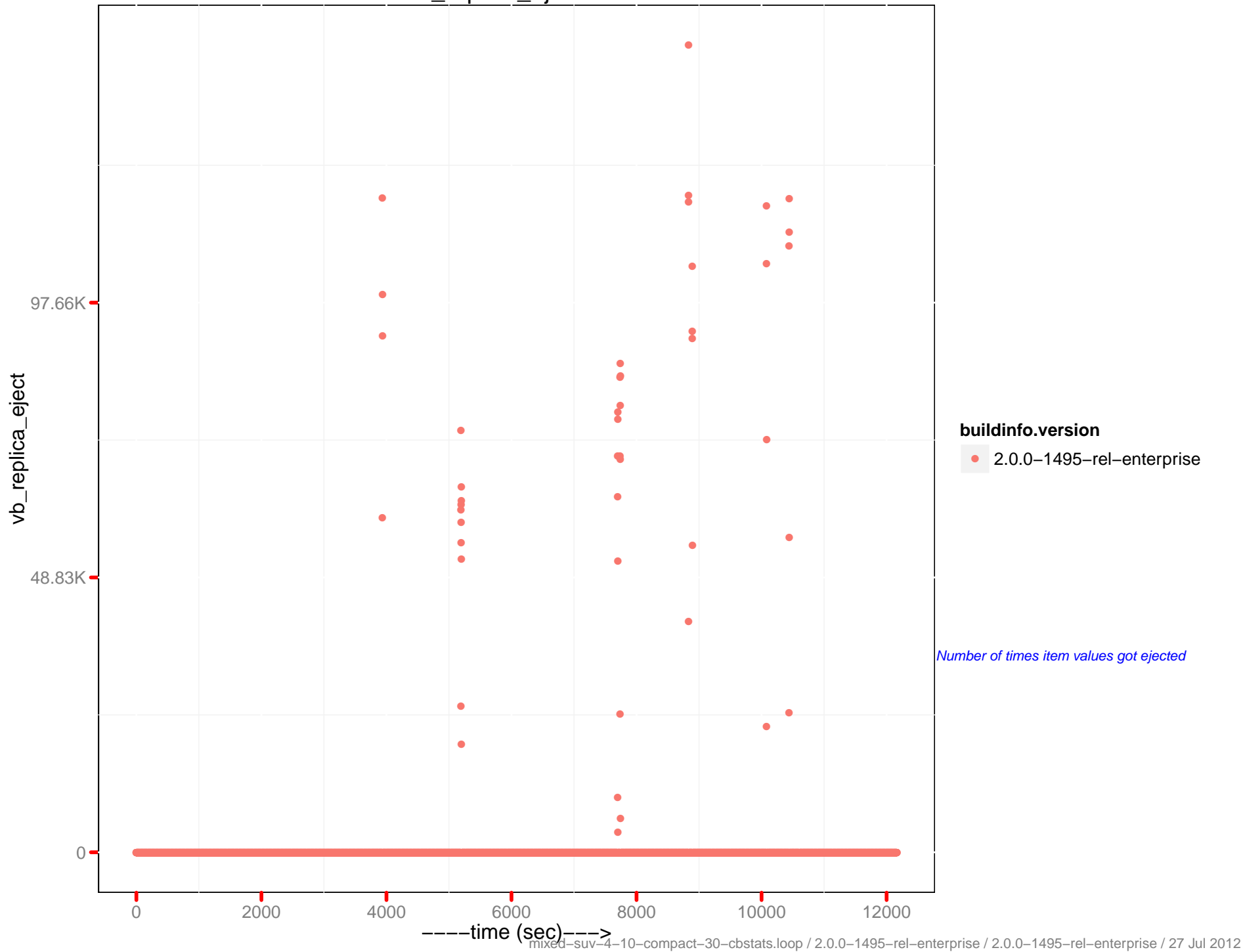
# tmp\_oom ops/sec



# vb\_active\_eject/sec

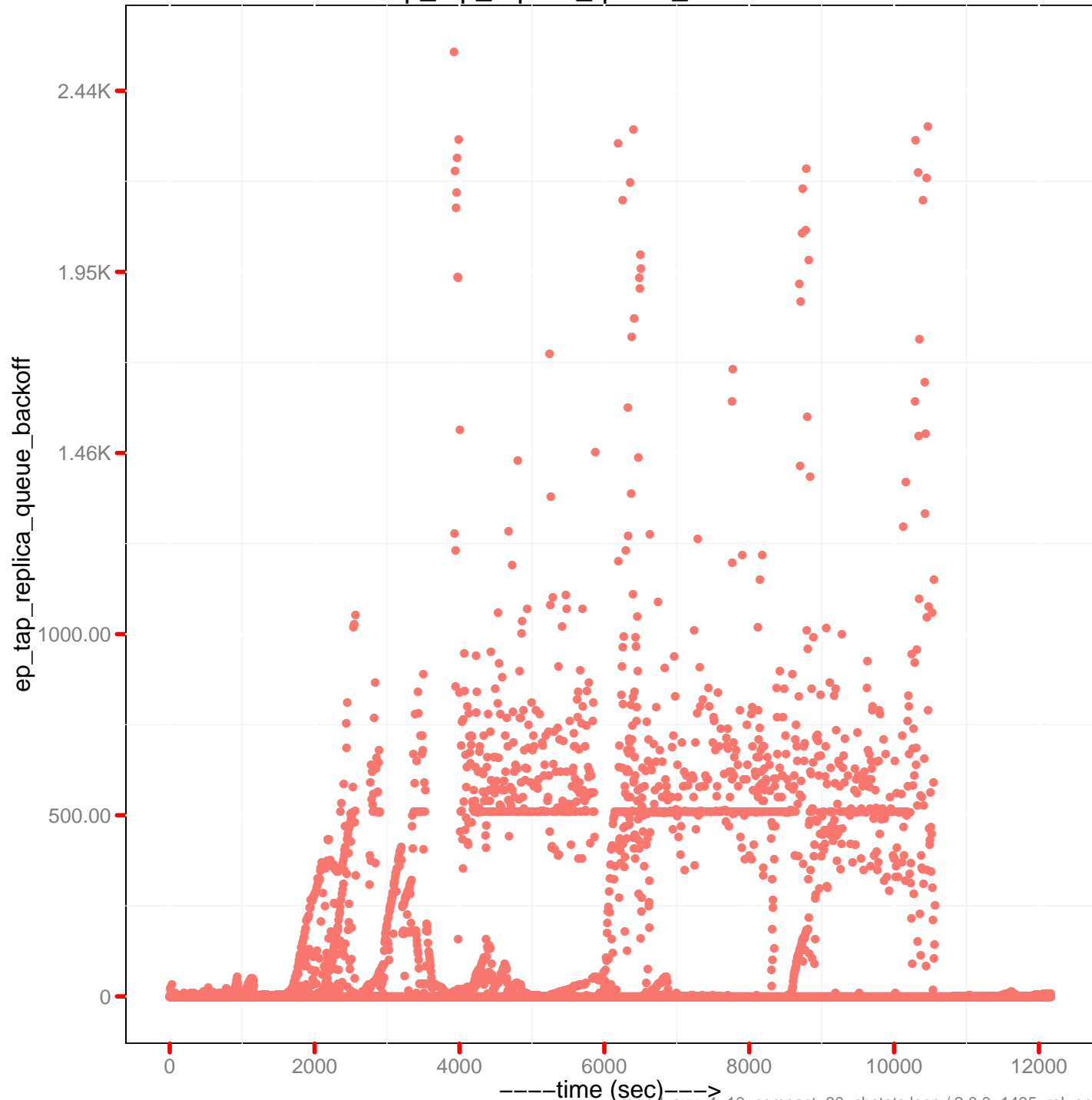


# vb\_replica\_eject/sec





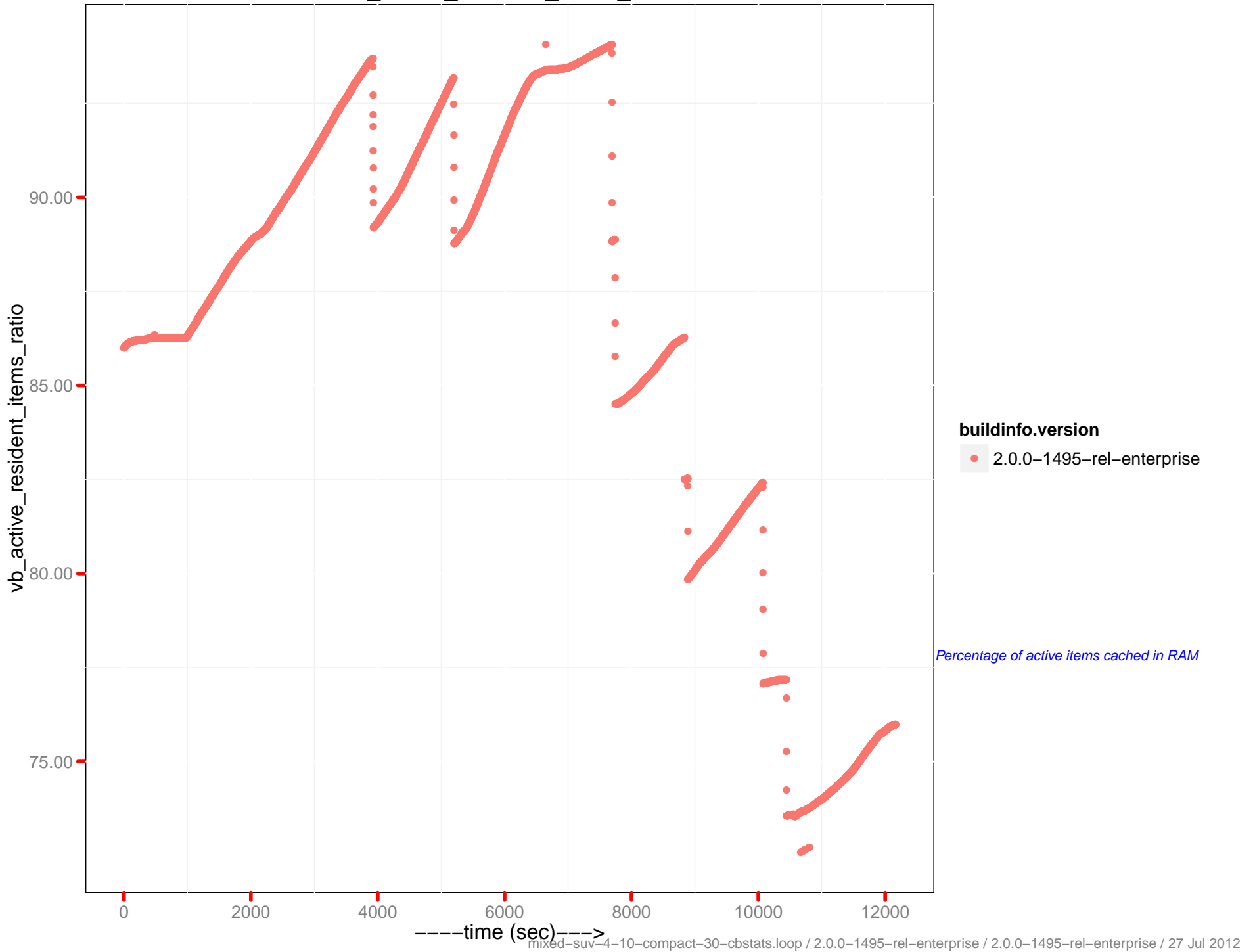
# ep\_tap\_replica\_queue\_backoff/sec



**buildinfo.version**  
● 2.0.0-1495-rel-enterprise

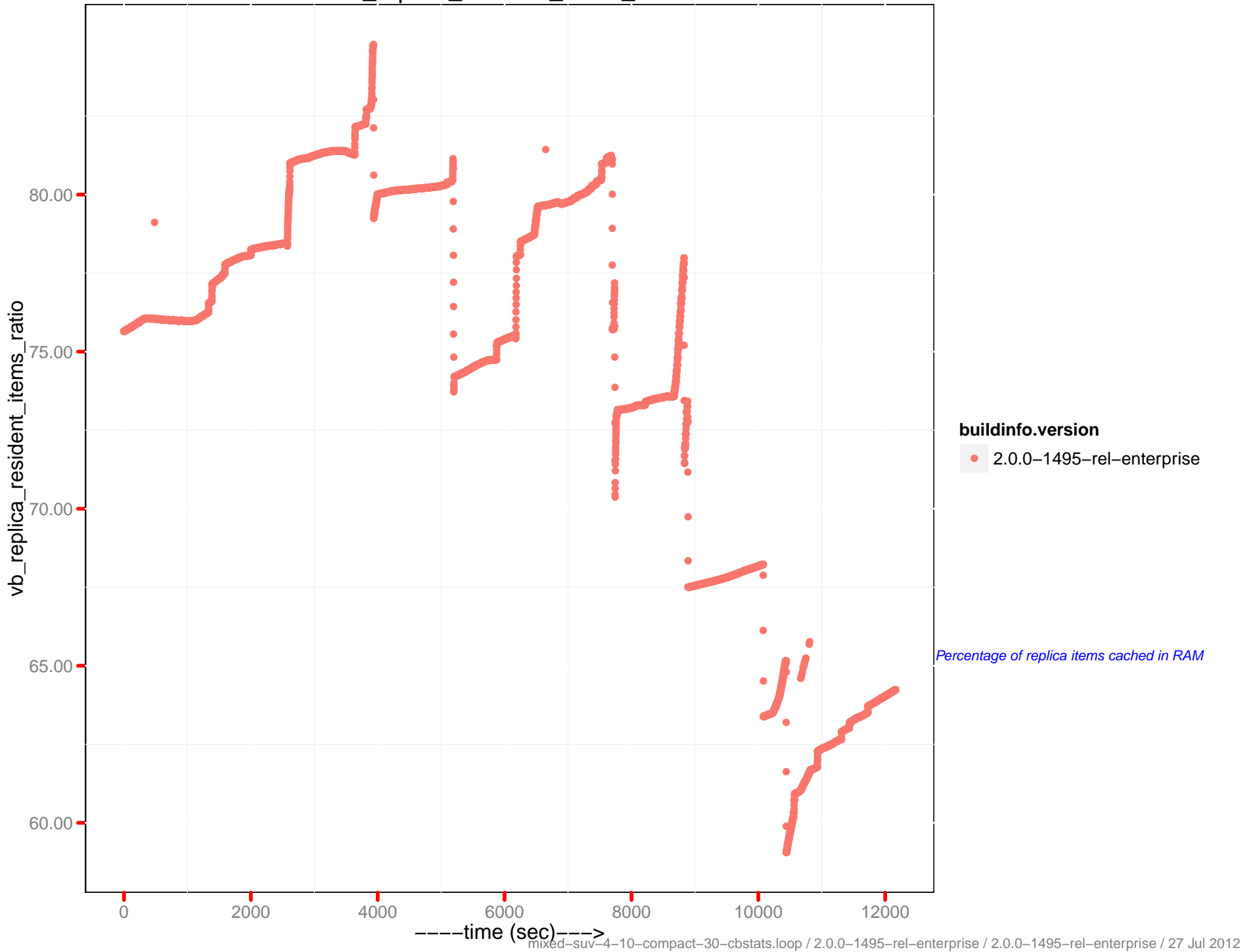
*Number of back-offs received per second  
while sending data over replication  
TAP connections*

# vb\_active\_resident\_items\_ratio

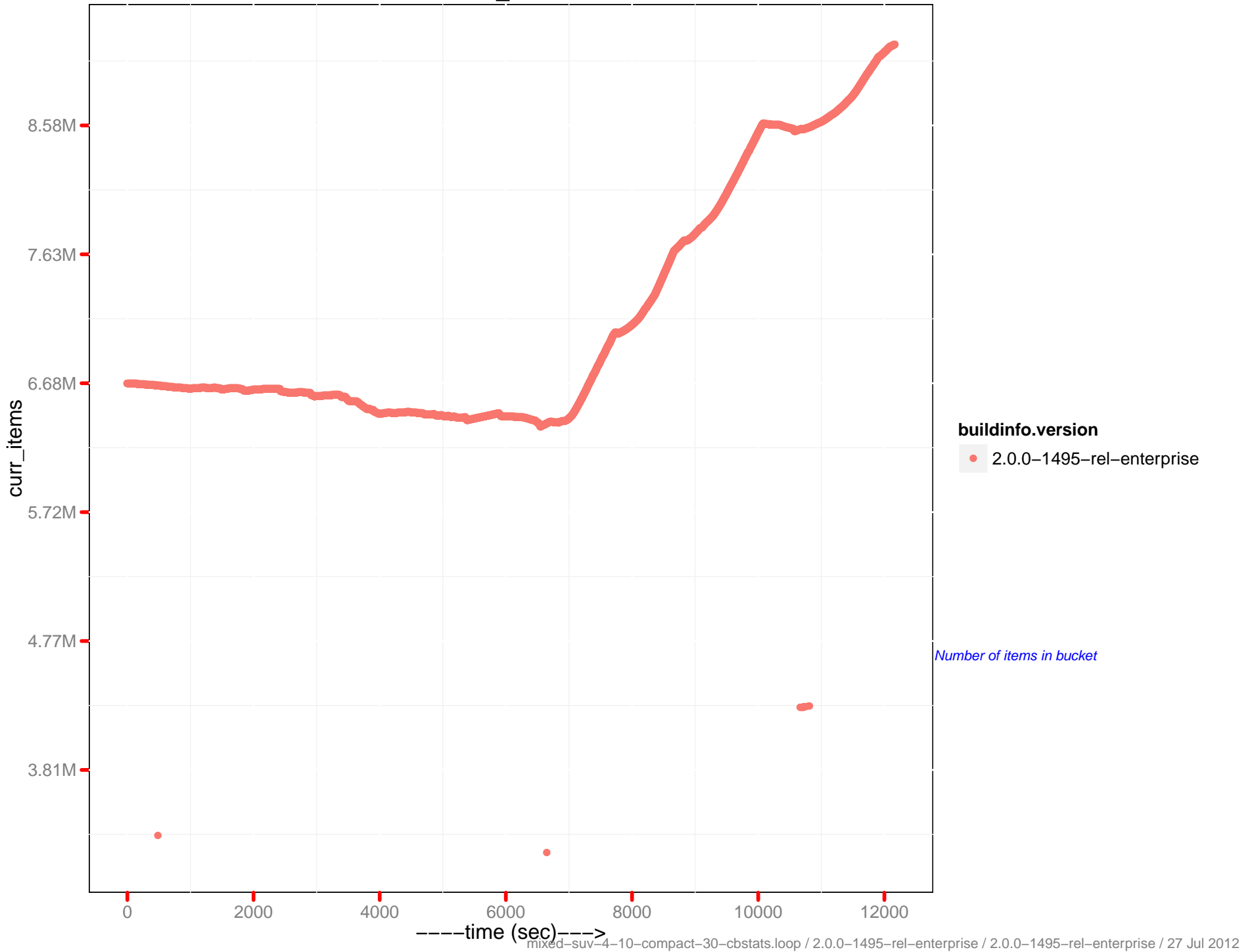


Percentage of active items cached in RAM

# vb\_replica\_resident\_items\_ratio



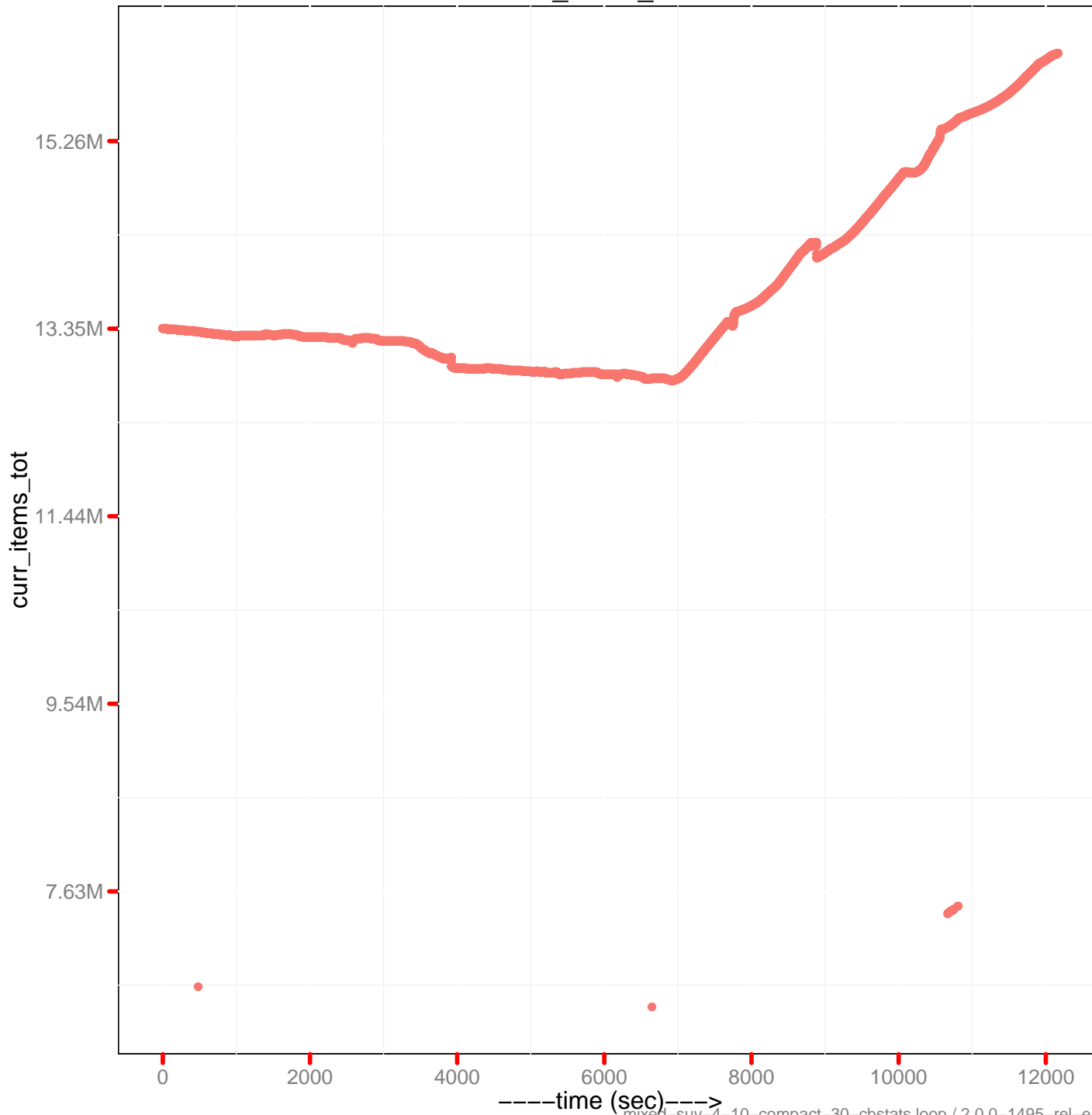
# curr\_items



**buildinfo.version**  
● 2.0.0-1495-rel-enterprise

*Number of items in bucket*

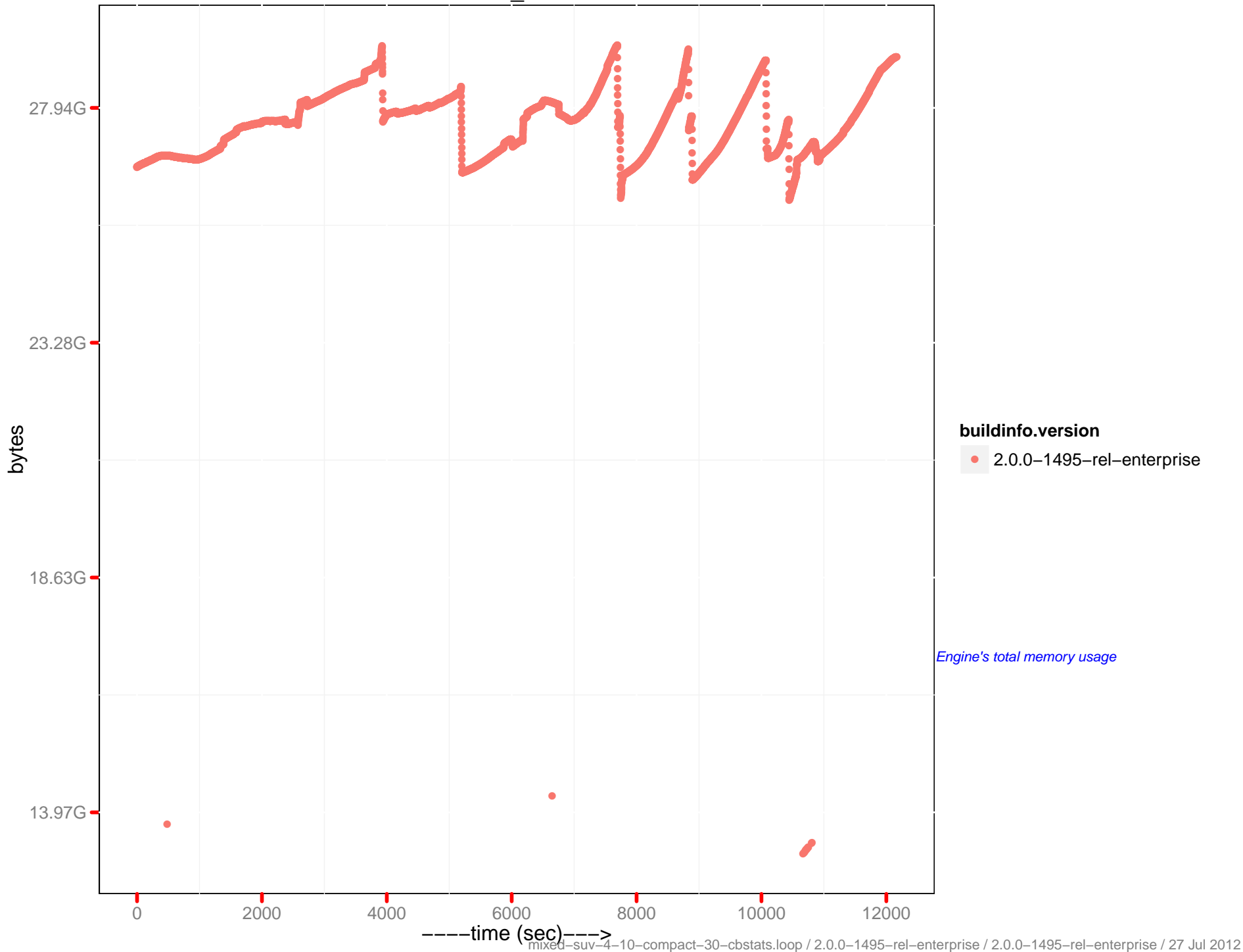
# cur\_items\_total



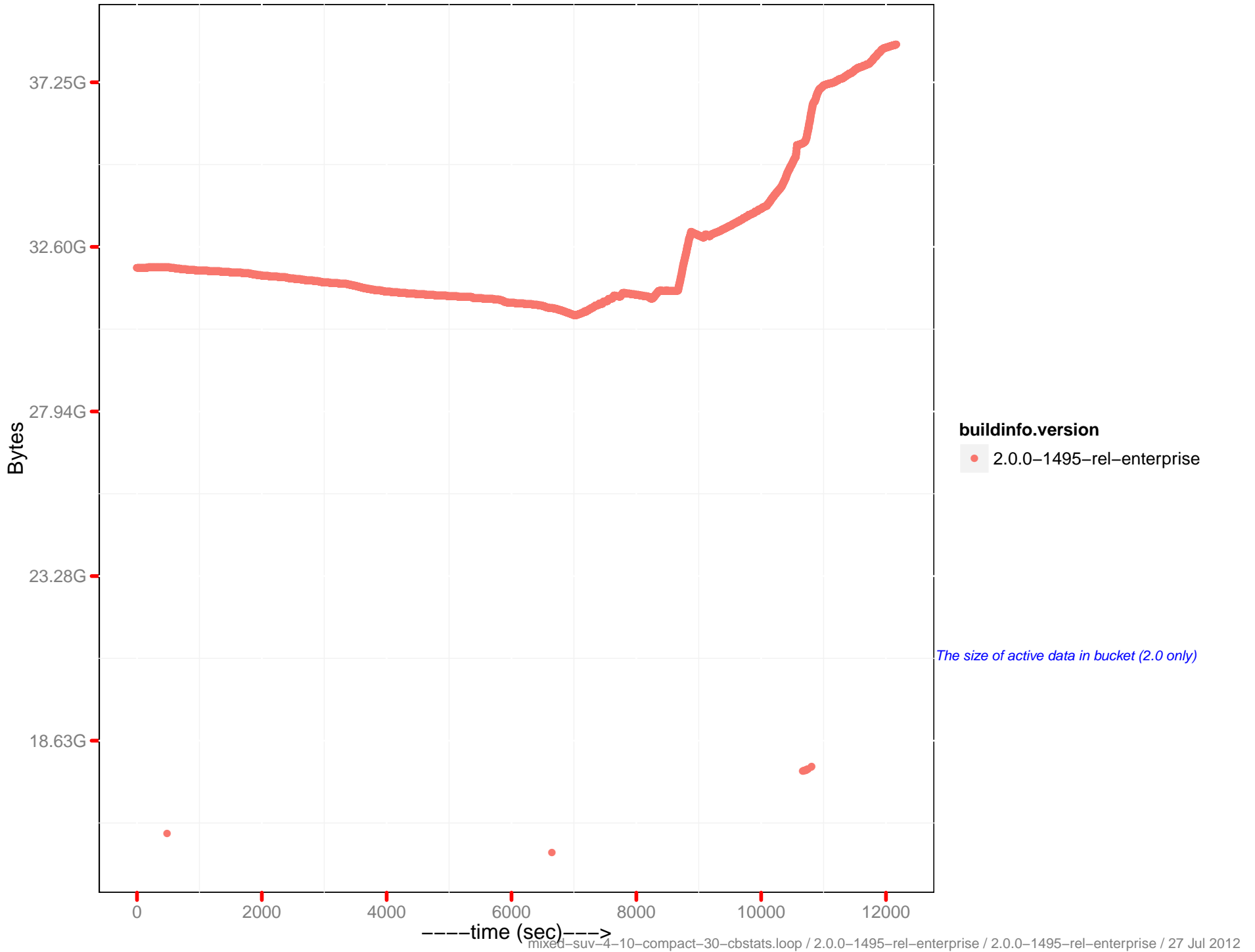
**buildinfo.version**  
● 2.0.0-1495-rel-enterprise

Total number of items

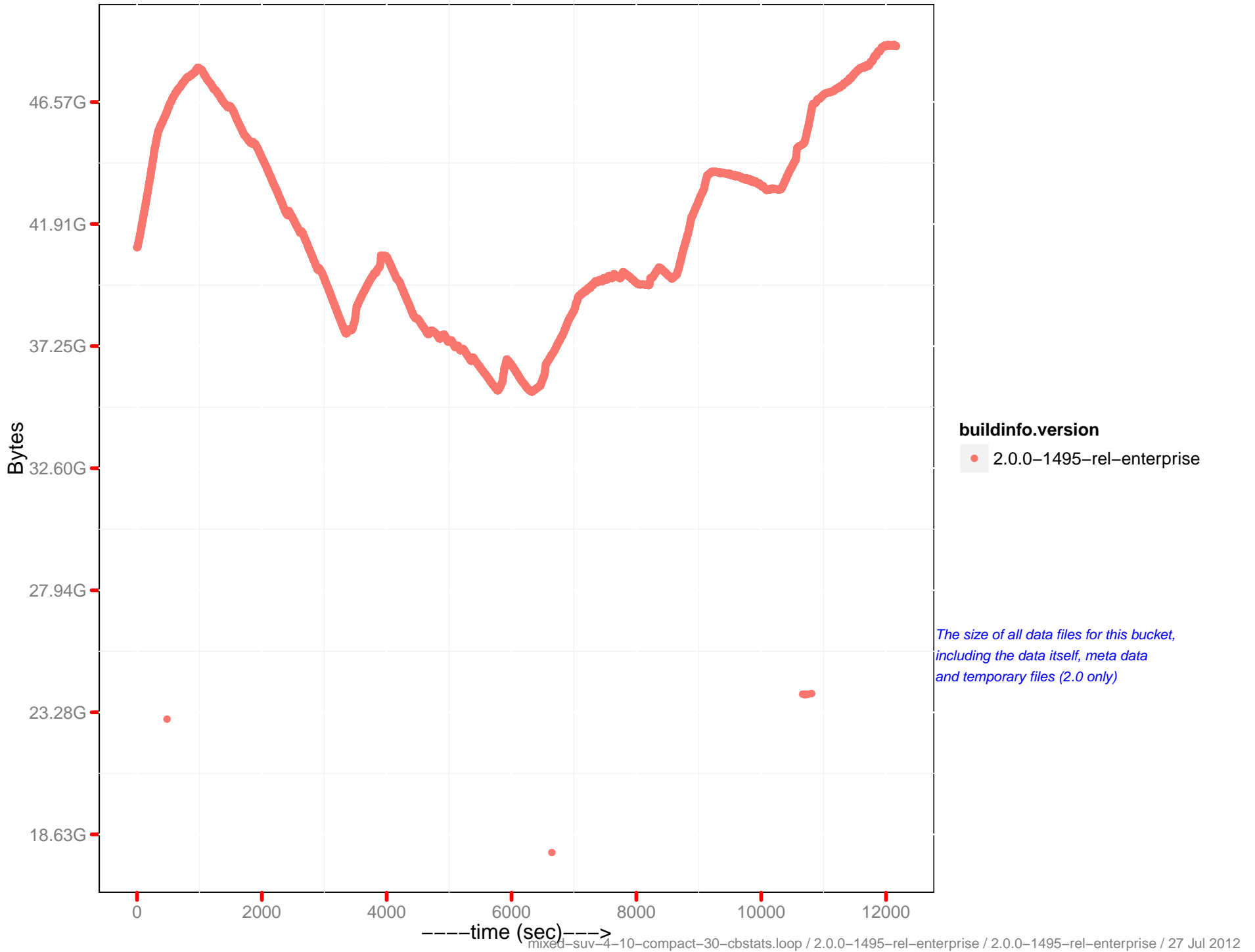
mem\_used



# Docs data size

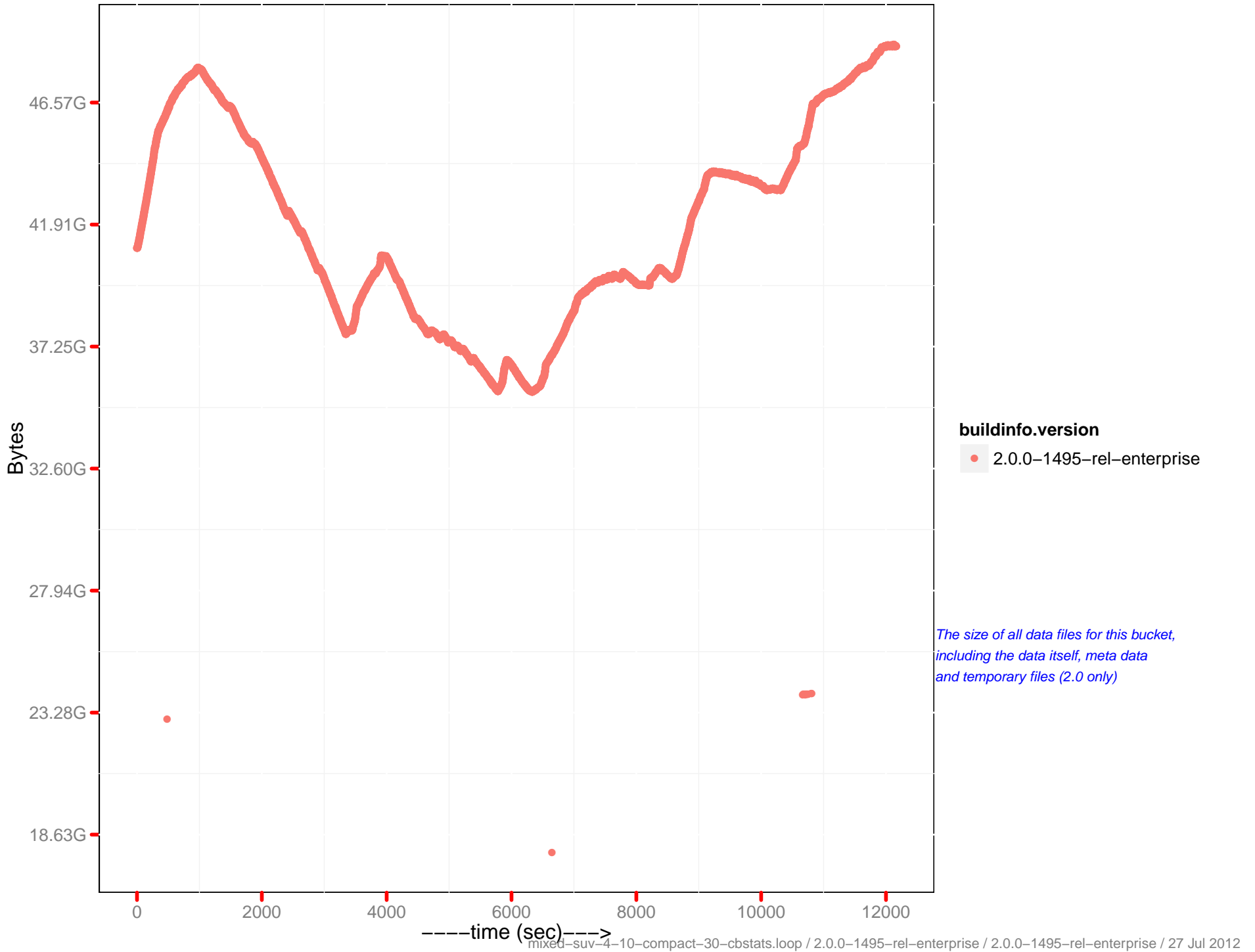


# Docs disk size

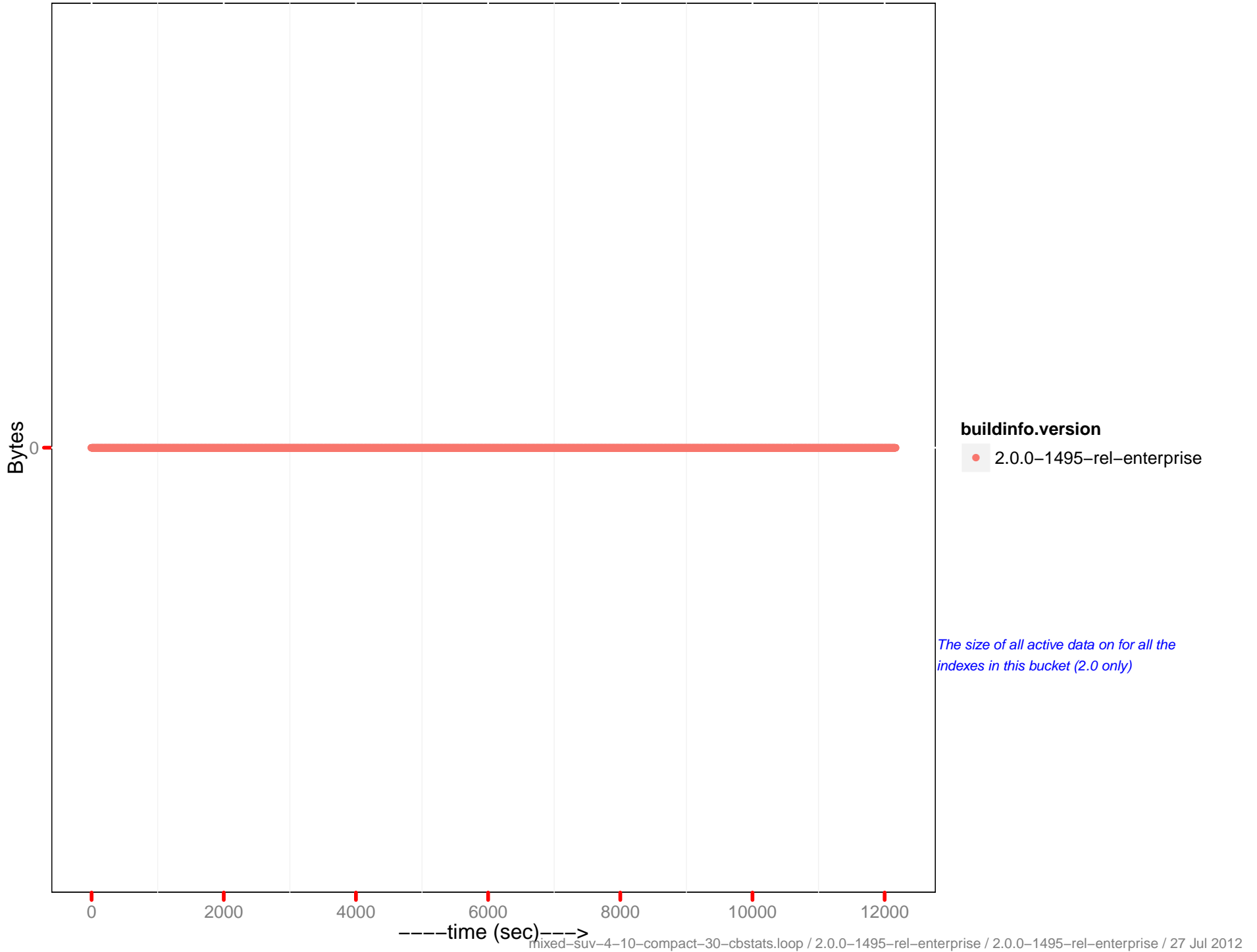




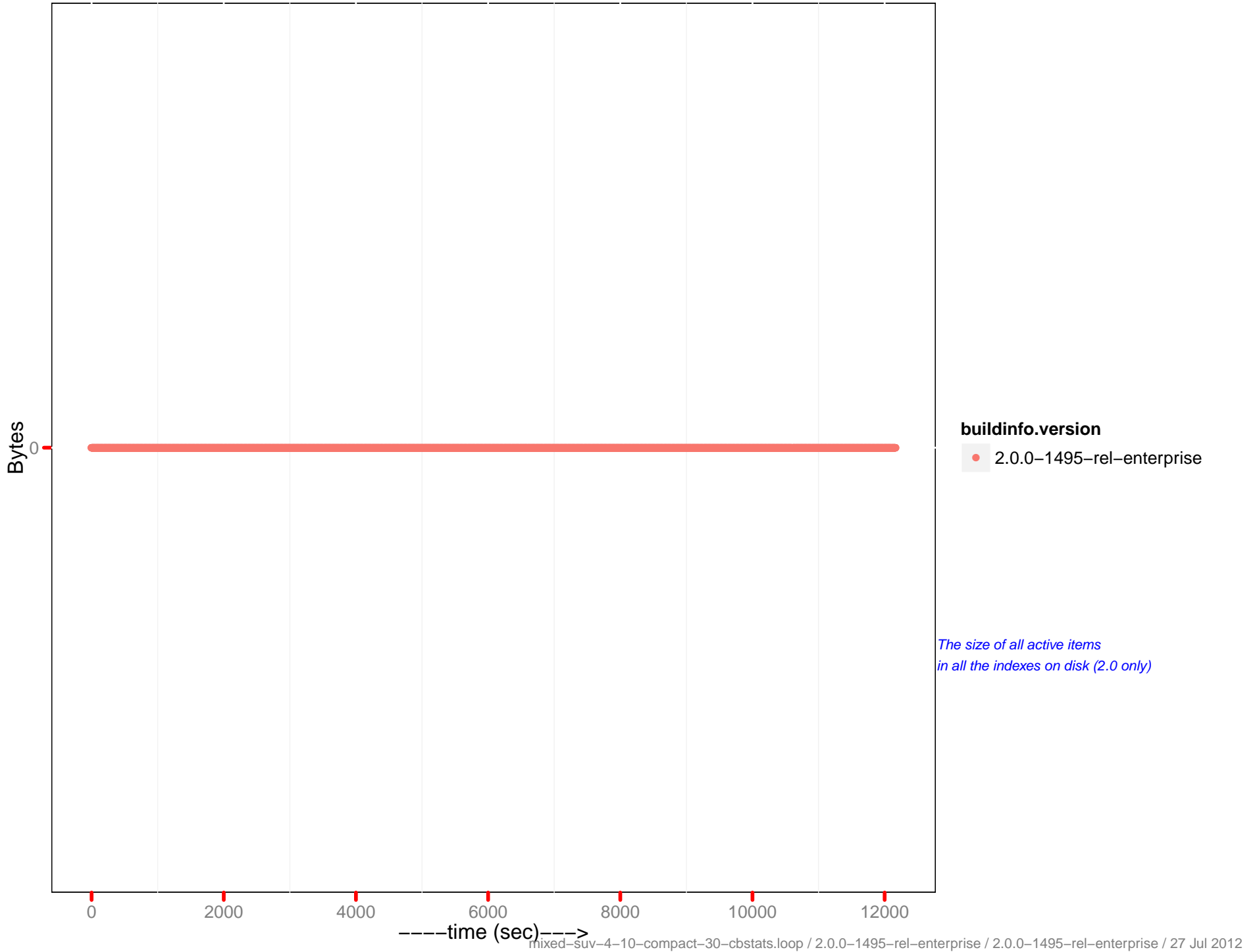
# Docs actual disk size



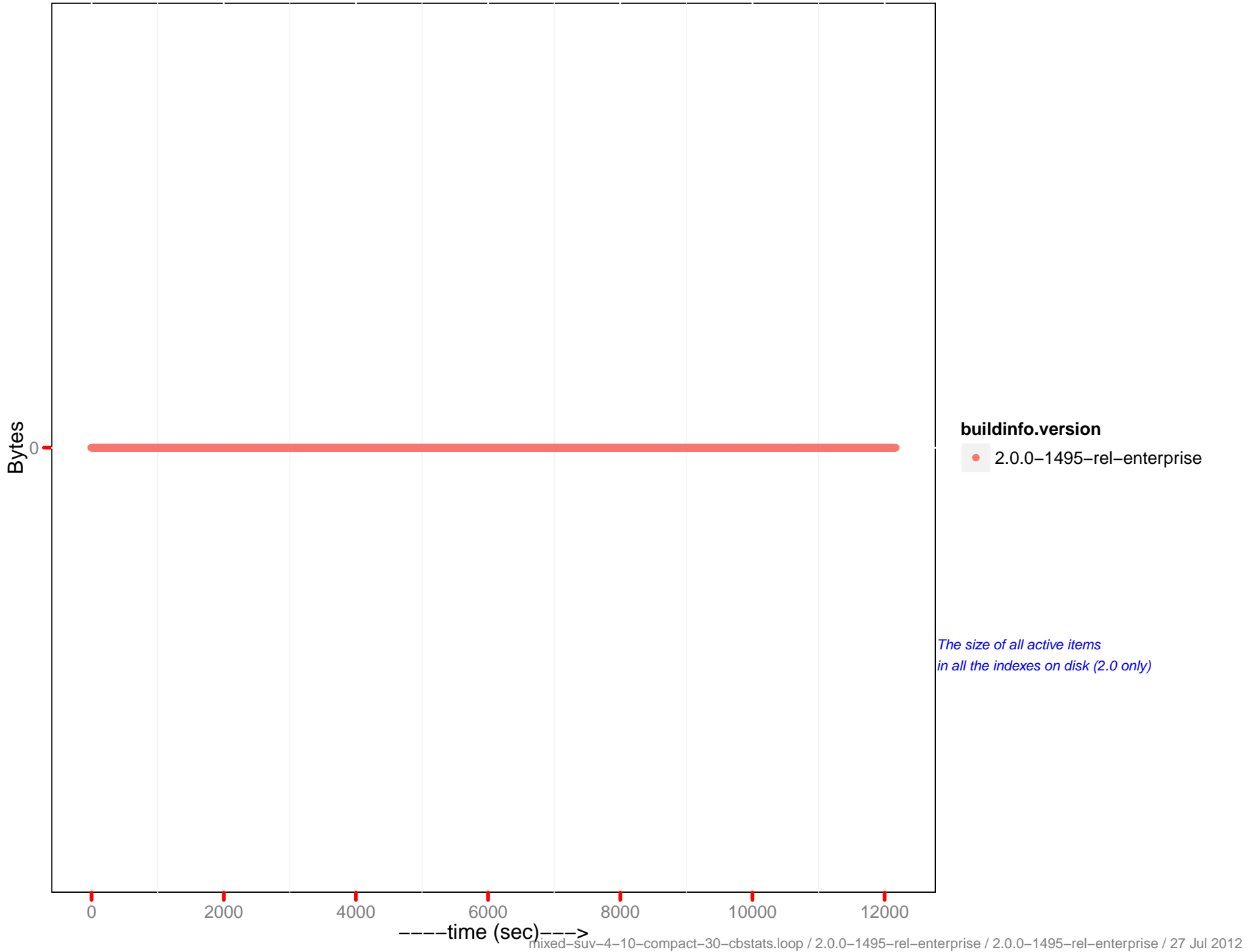
# Views data size



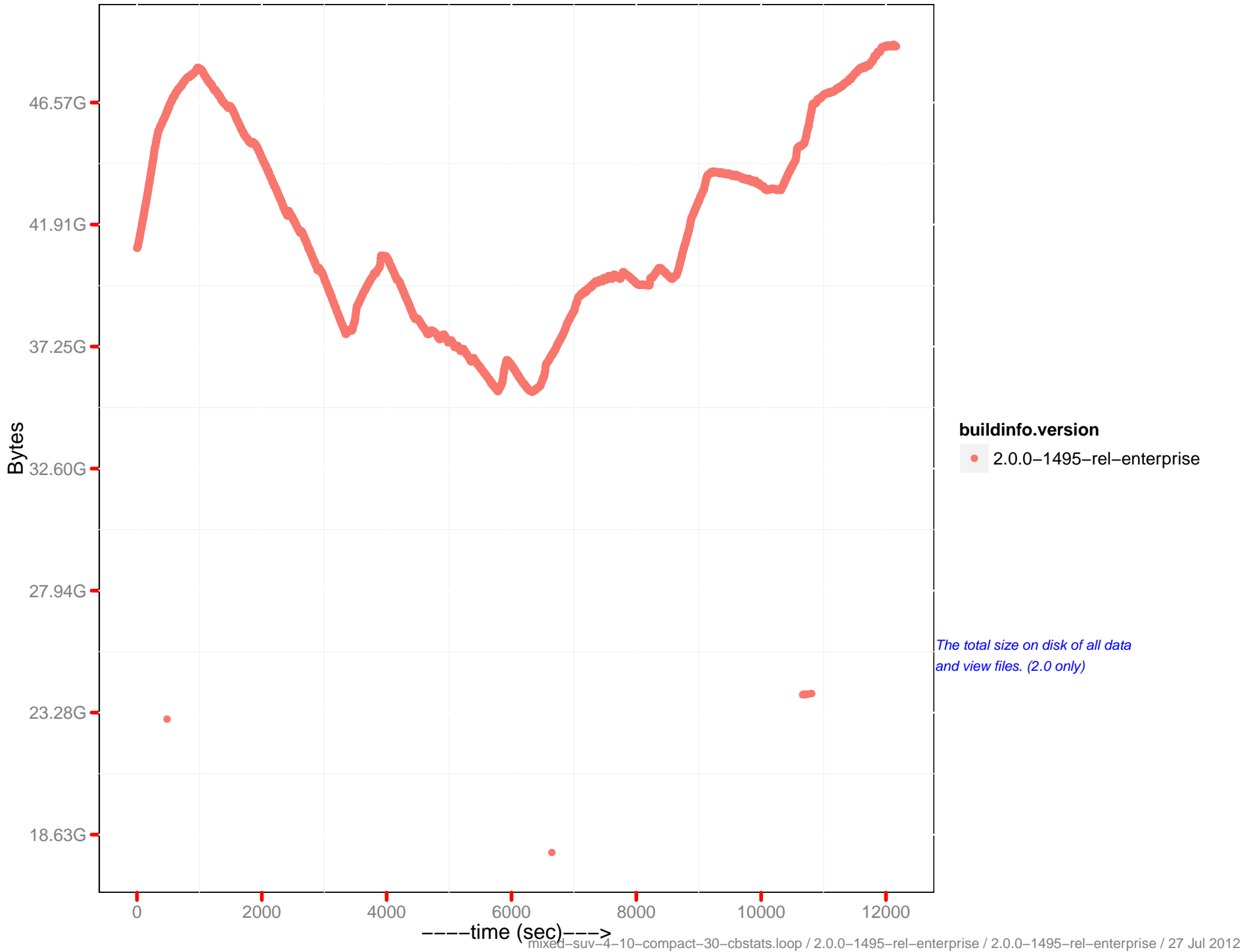
# Views disk size



# Views actual disk size



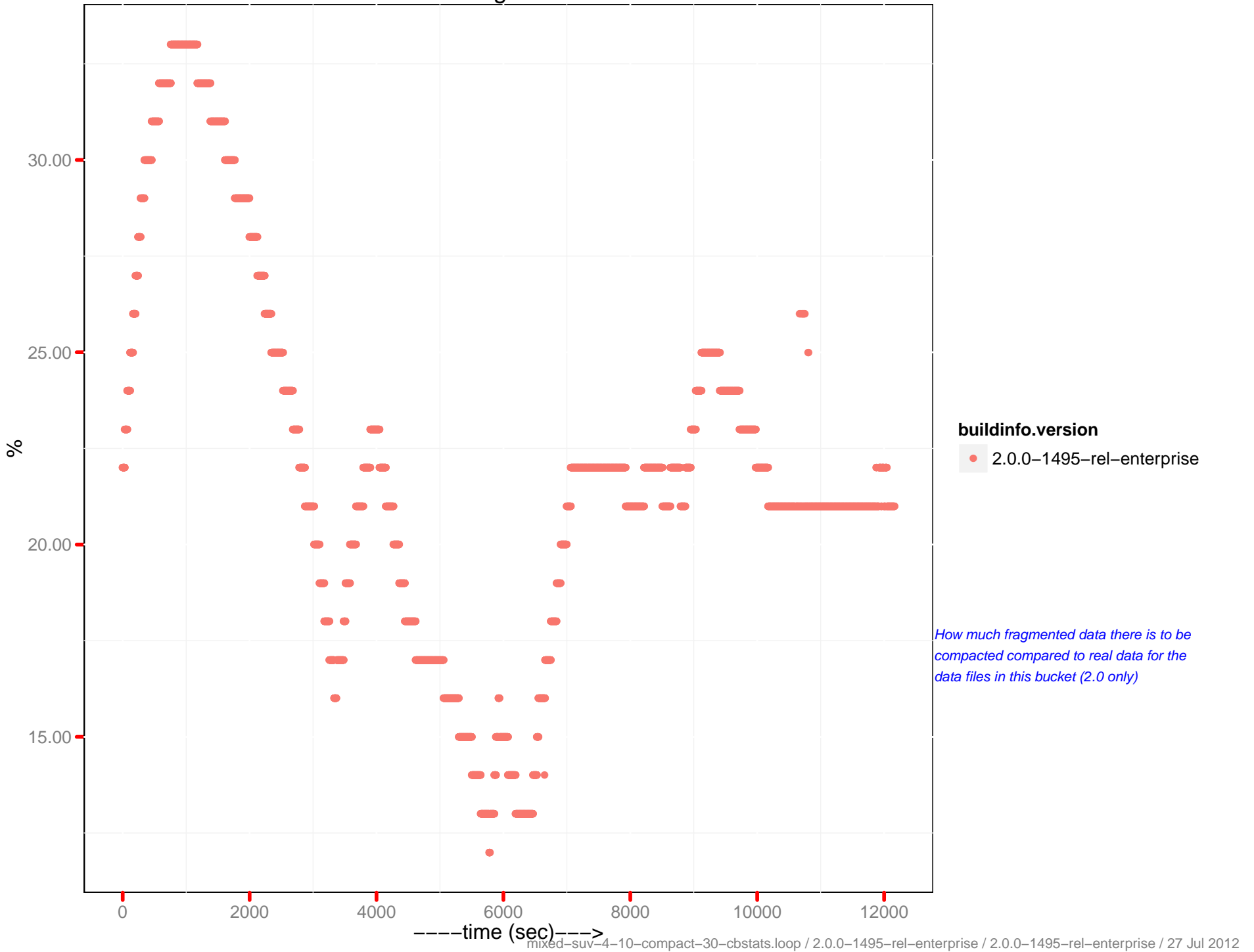
# Total disk size



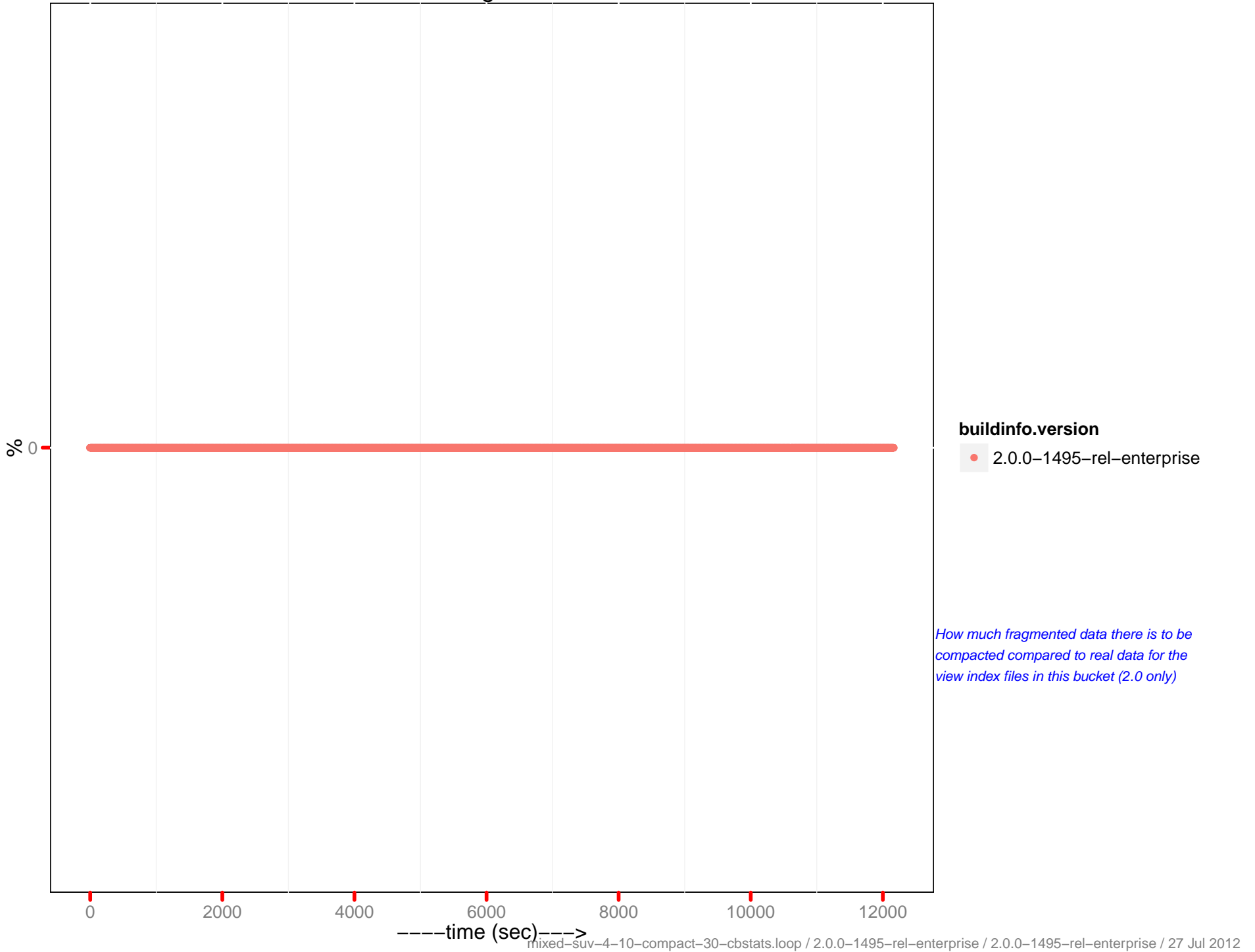
**buildinfo.version**  
● 2.0.0-1495-rel-enterprise

*The total size on disk of all data and view files. (2.0 only)*

# Docs fragmentation



# Views fragmentation

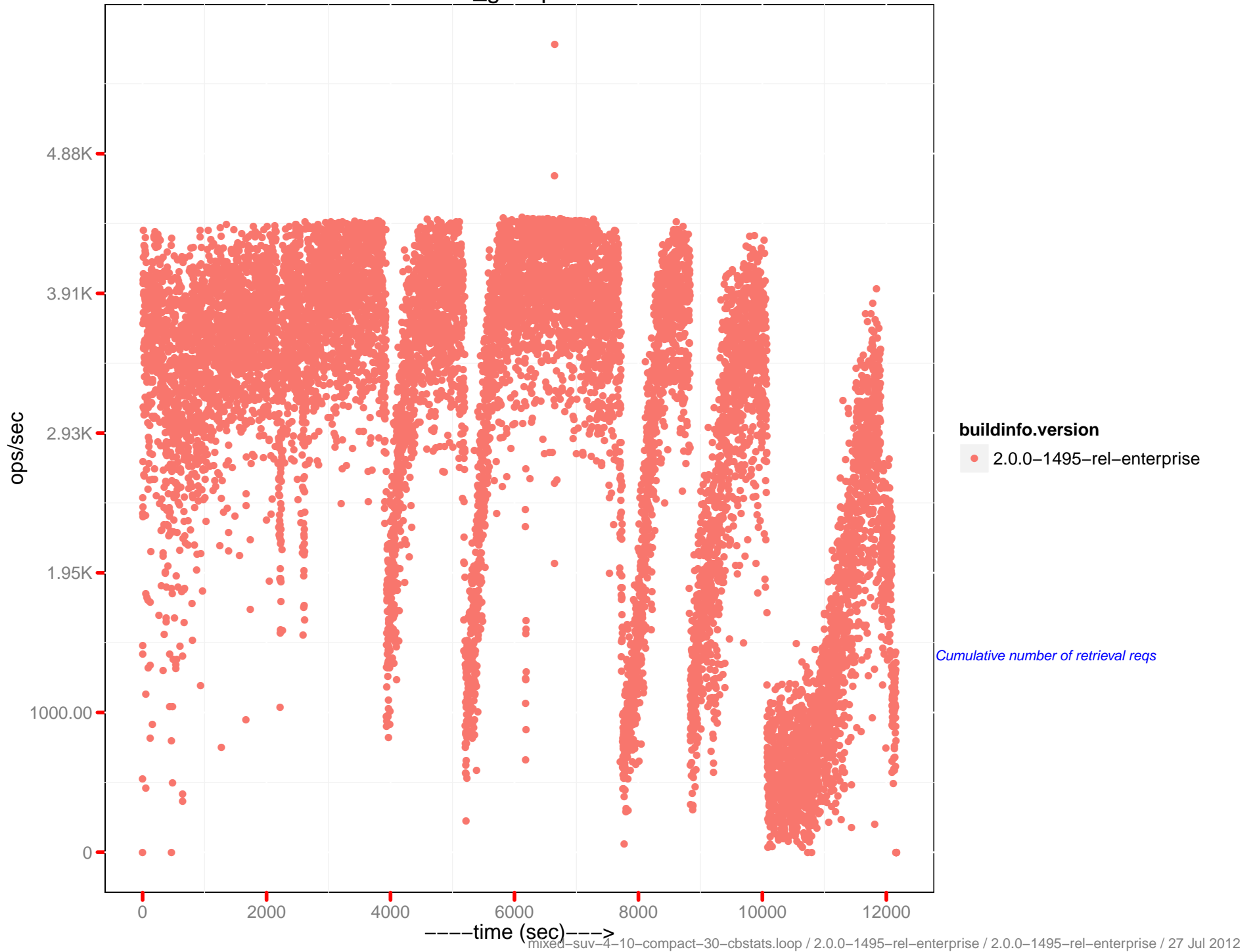


**buildinfo.version**

• 2.0.0-1495-rel-enterprise

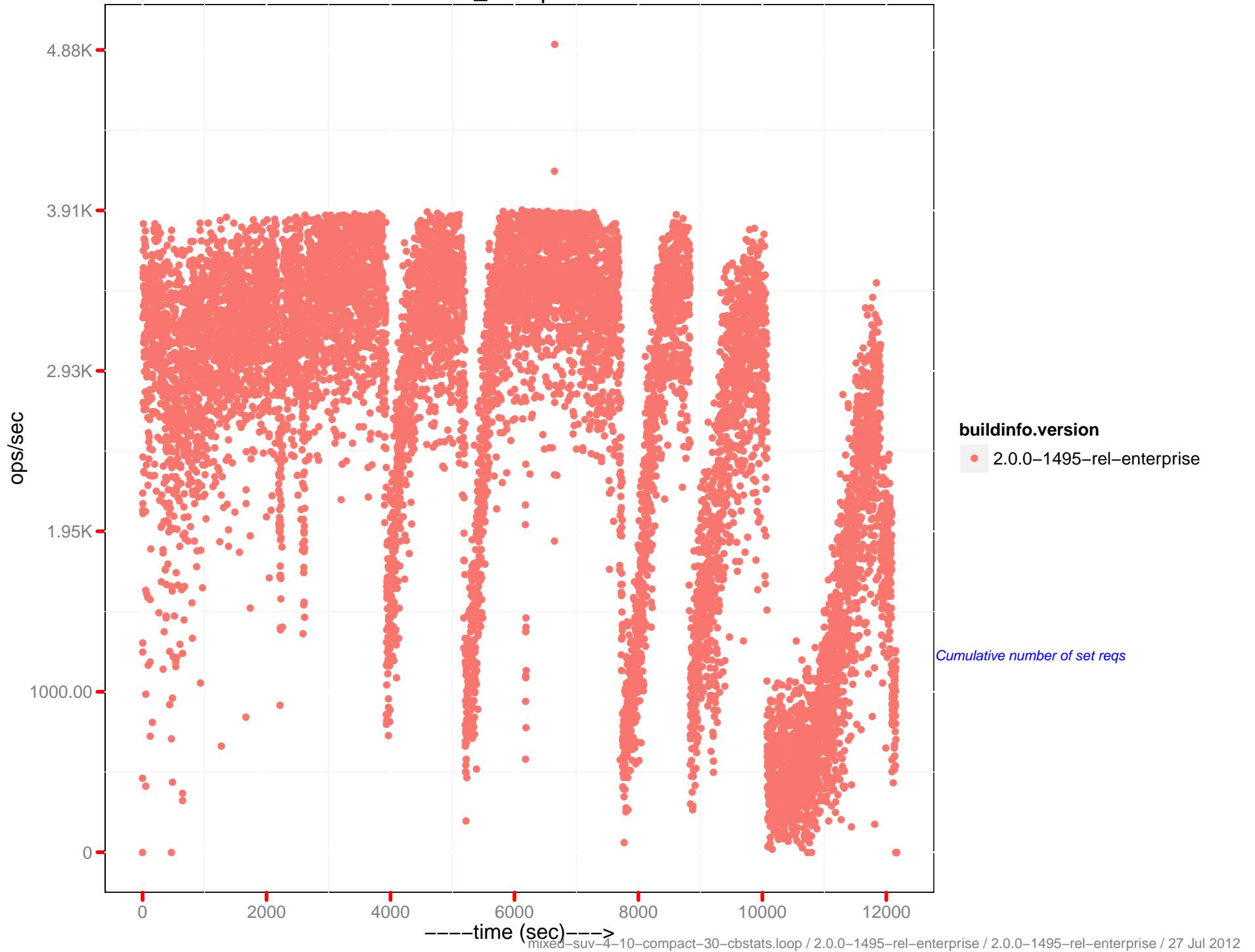
*How much fragmented data there is to be compacted compared to real data for the view index files in this bucket (2.0 only)*

# cmd\_get ops/sec

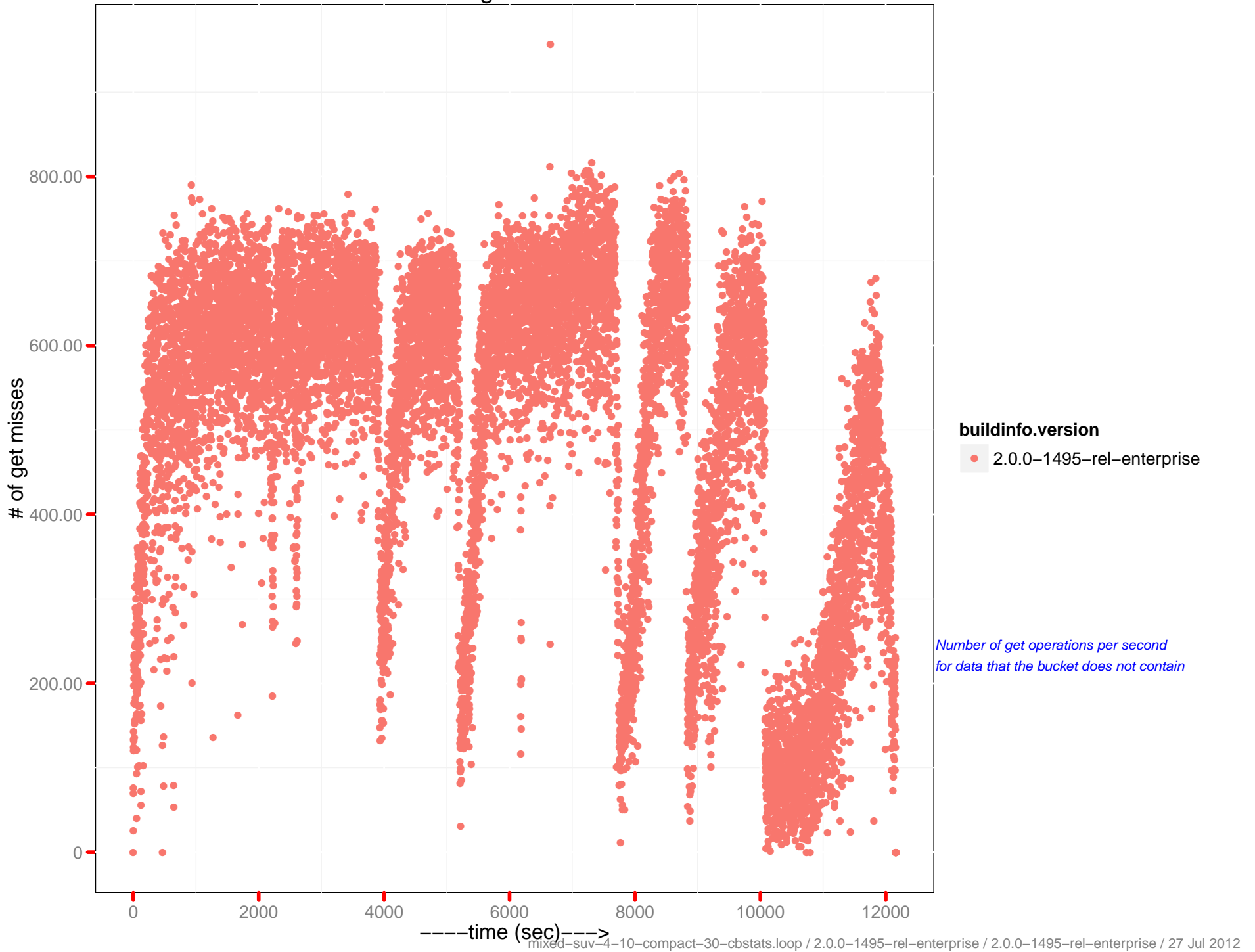




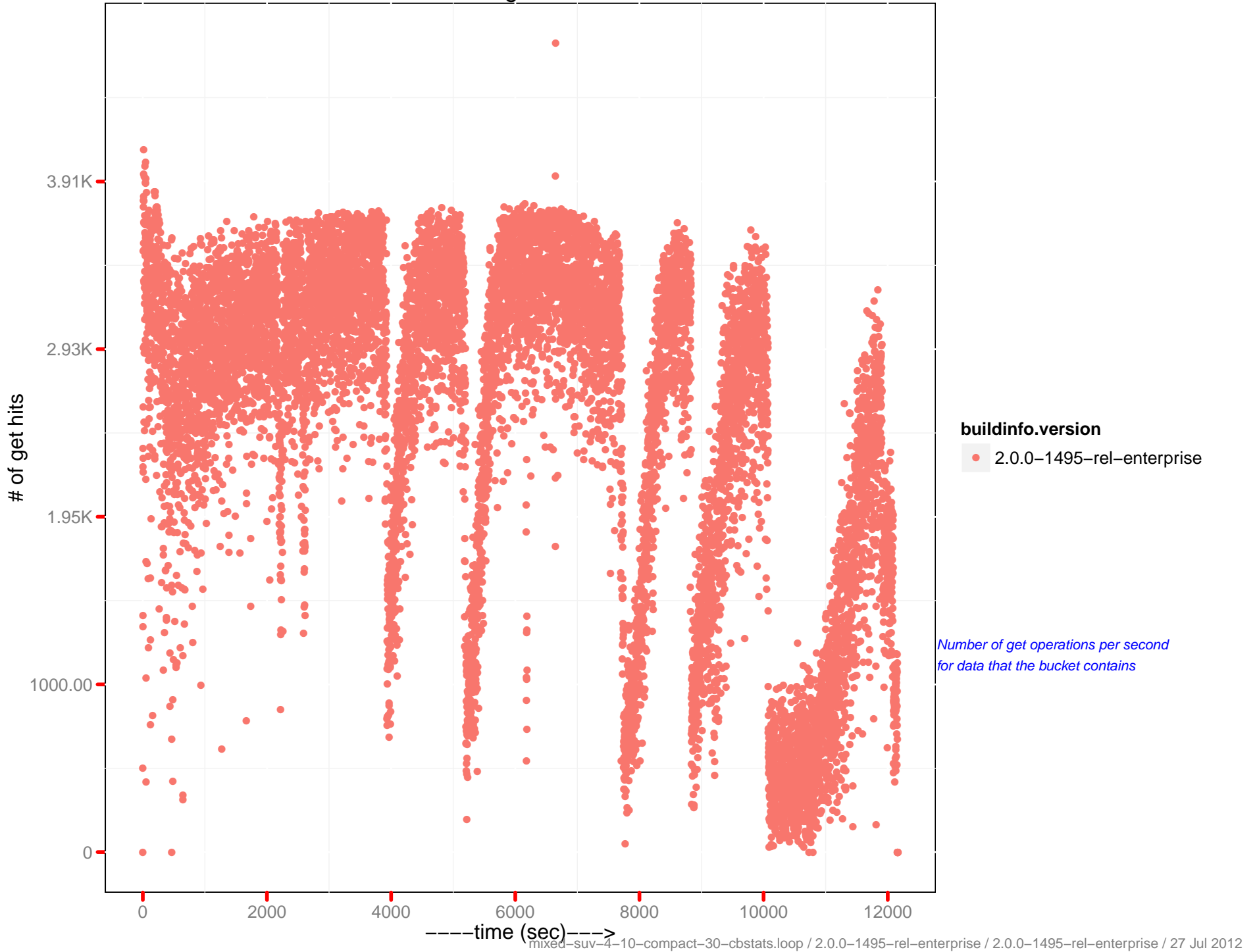
# cmd\_set ops/sec



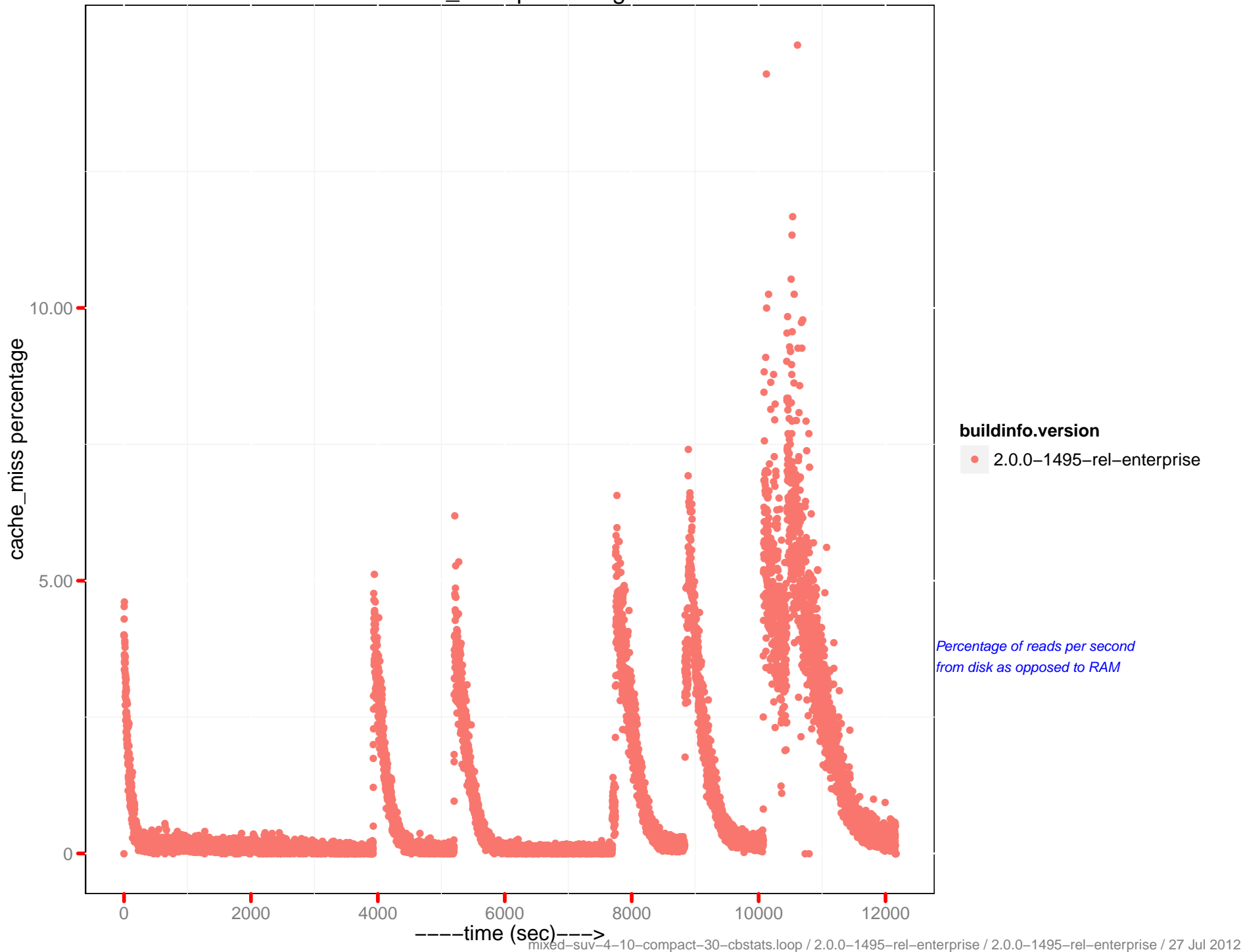
# # of get misses



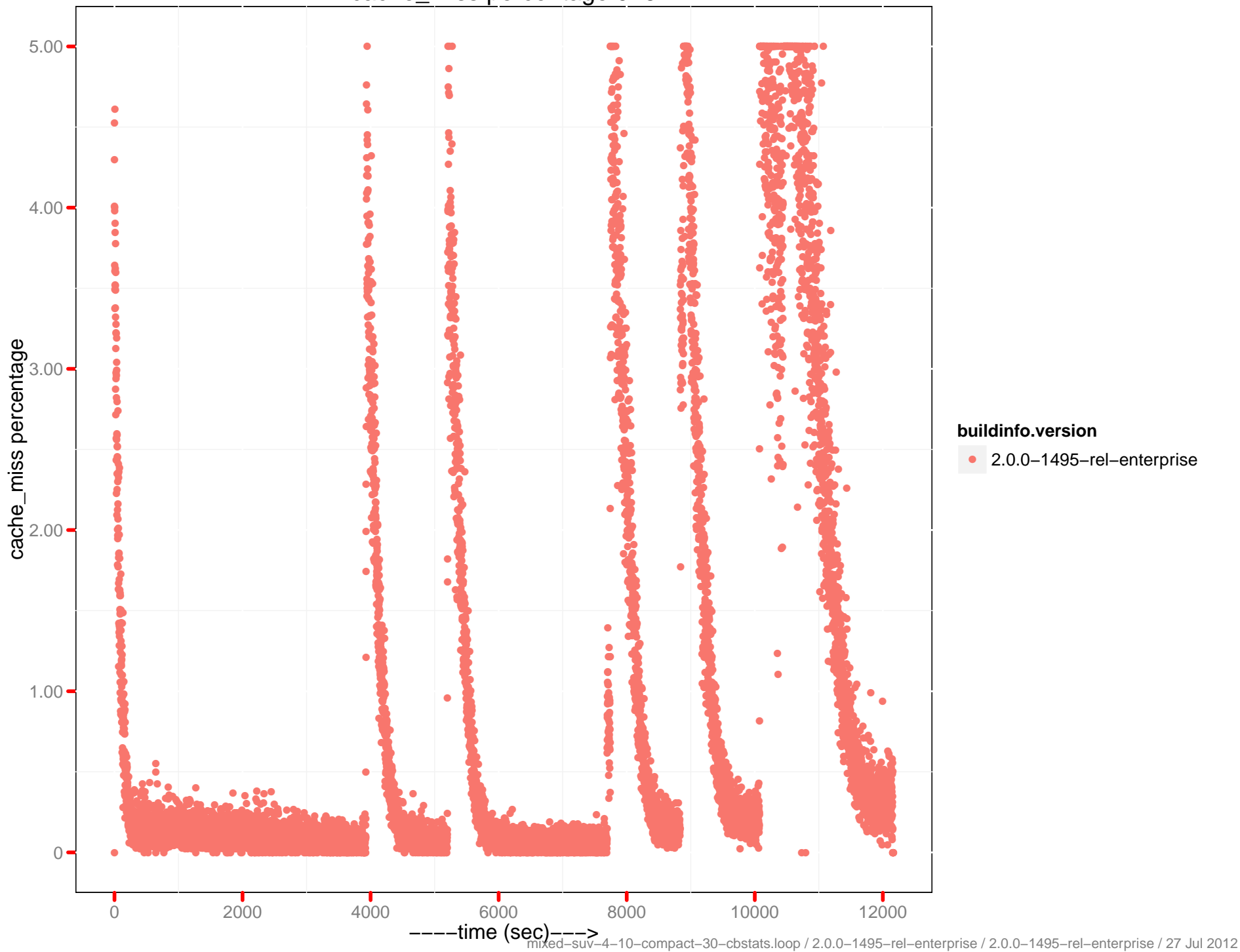
# # of get hits



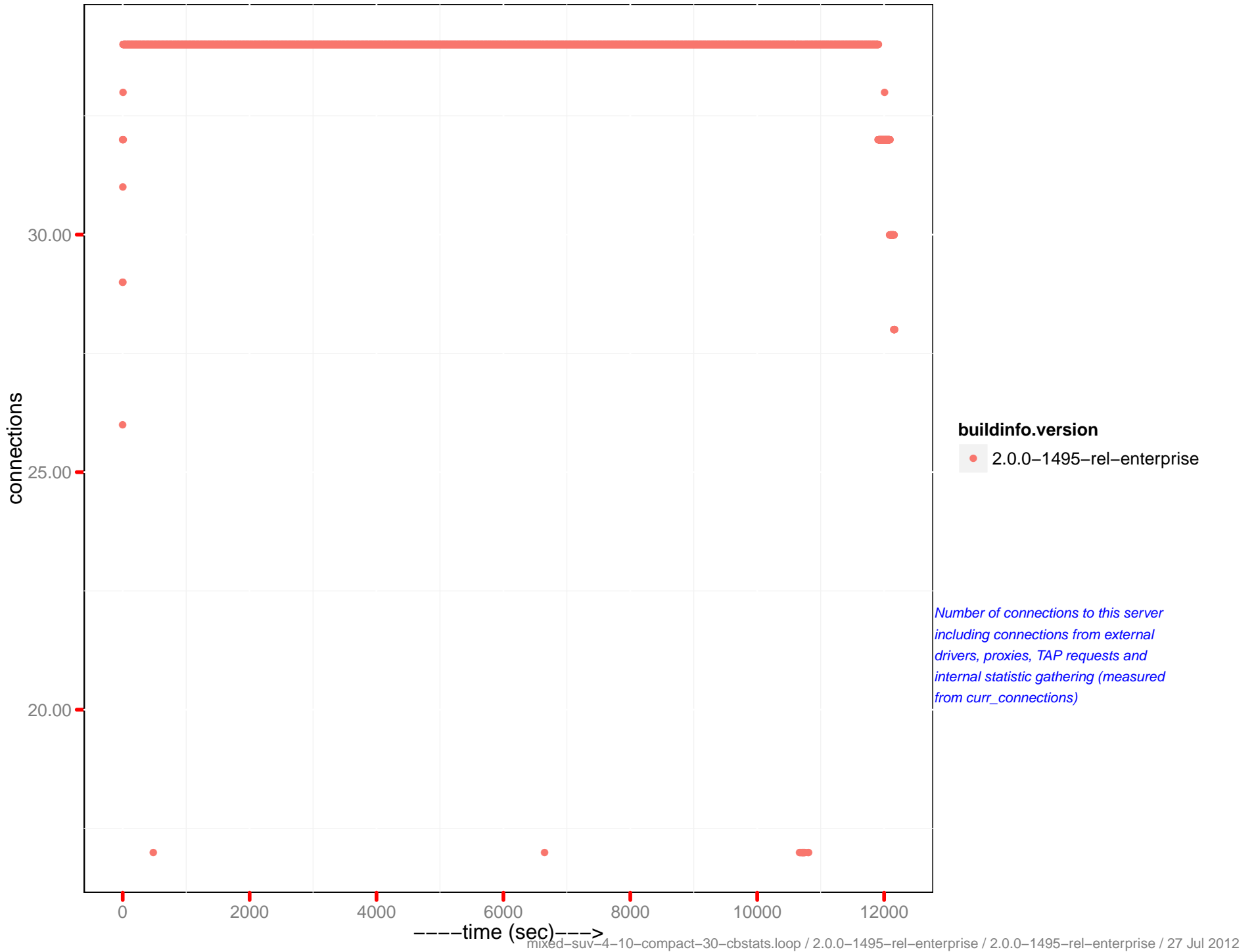
# cache\_miss percentage



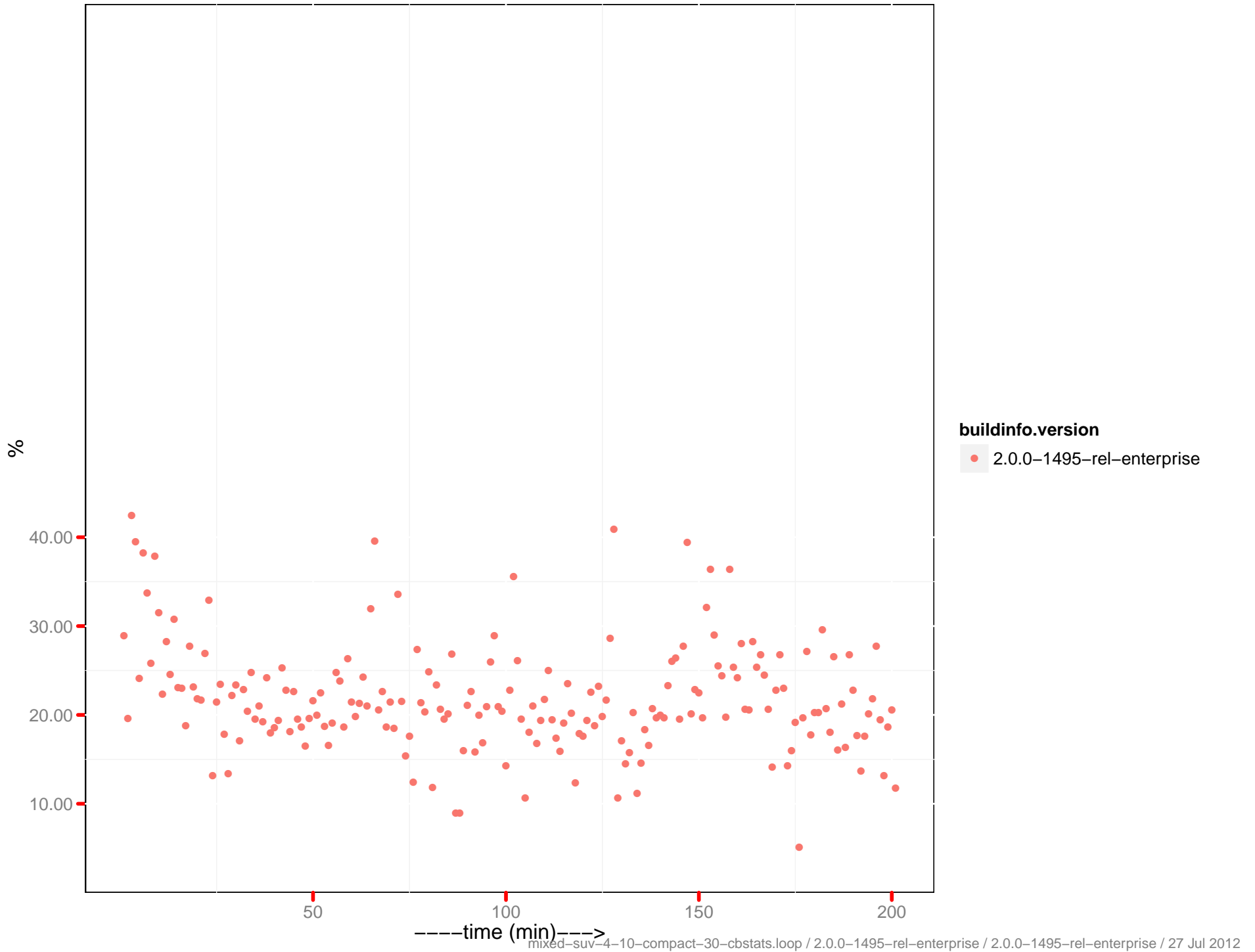
cache\_miss percentage 0-5



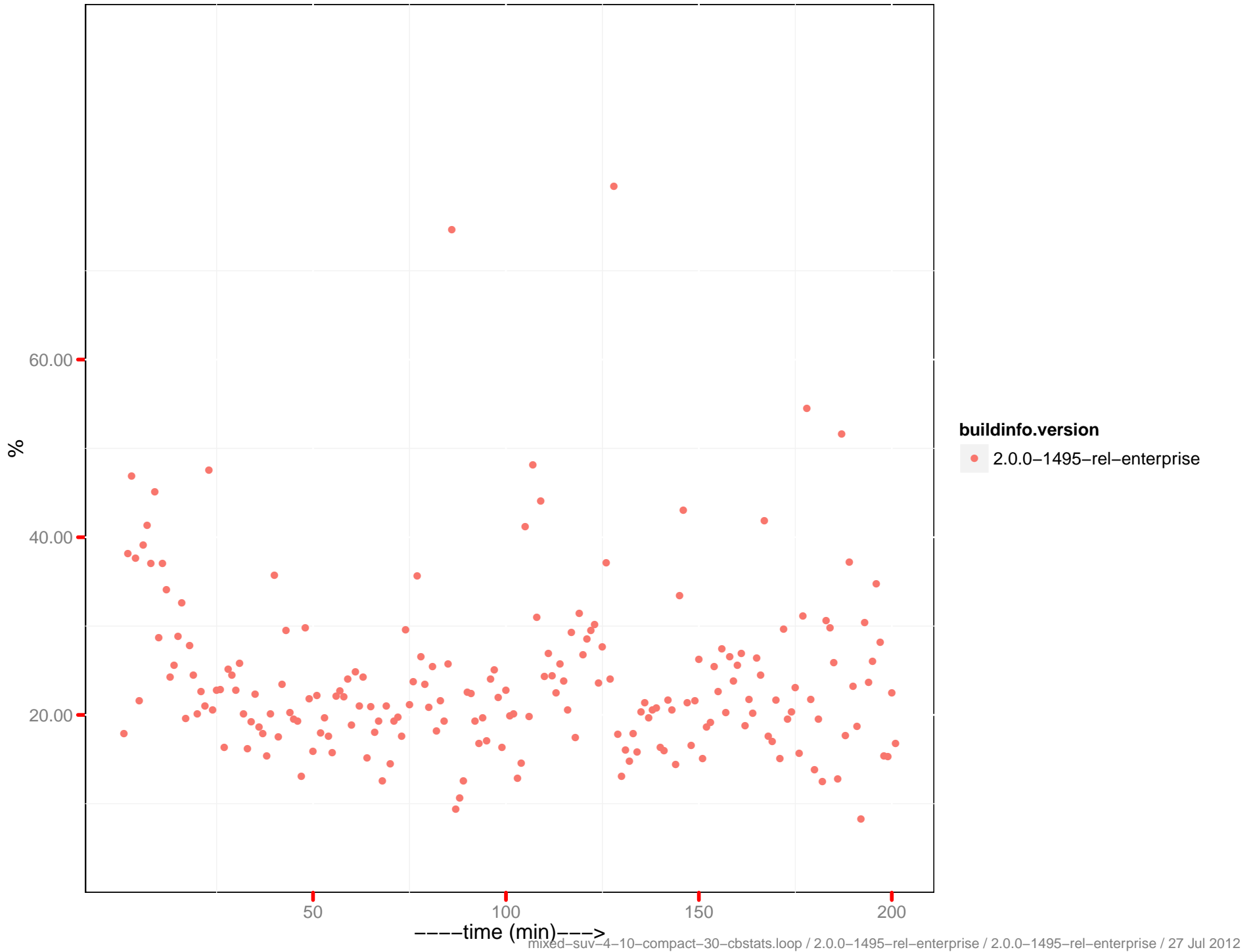
# Number of connections



# CPU utilization – 192.168.162.20:8091

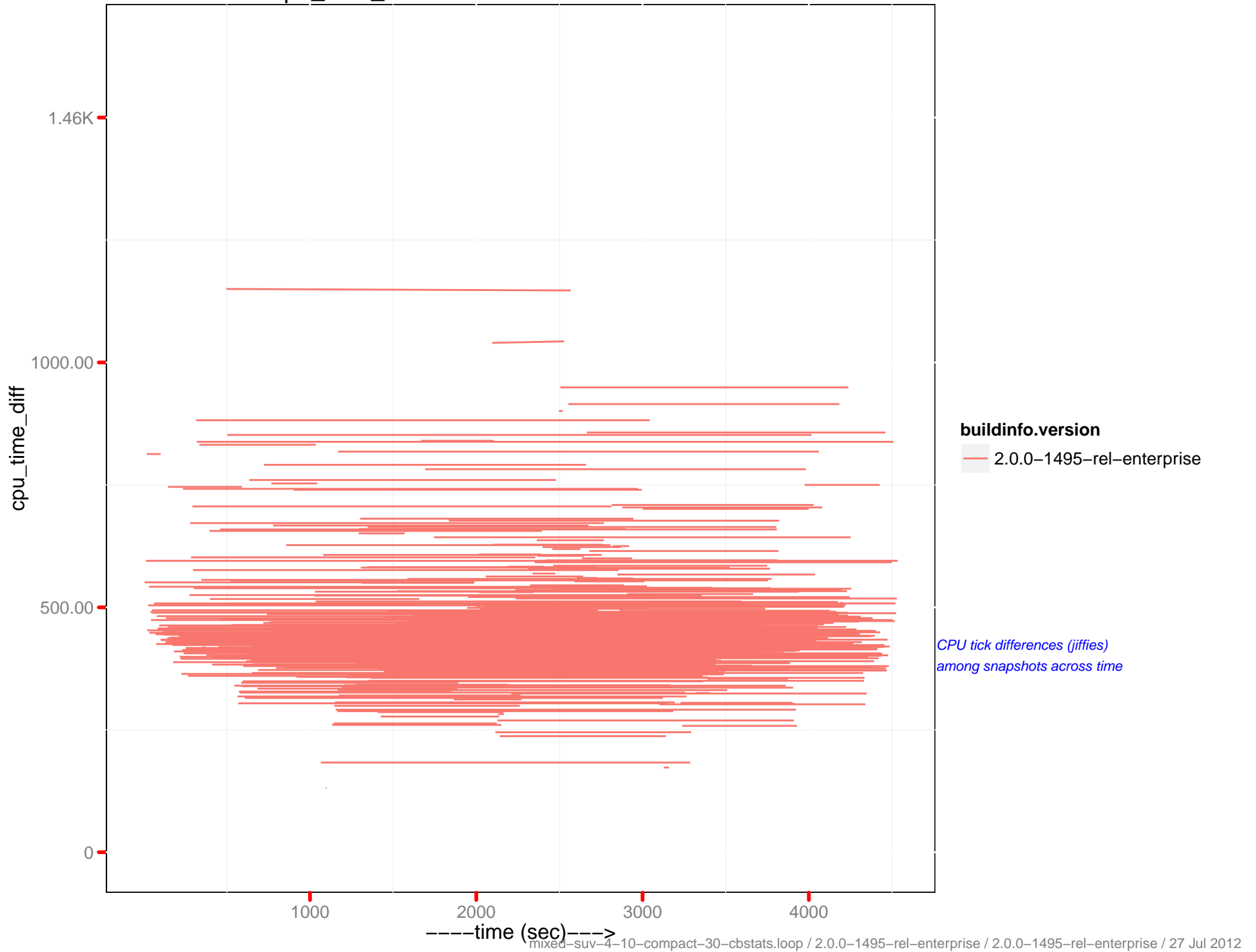


# CPU utilization – 192.168.162.21:8091

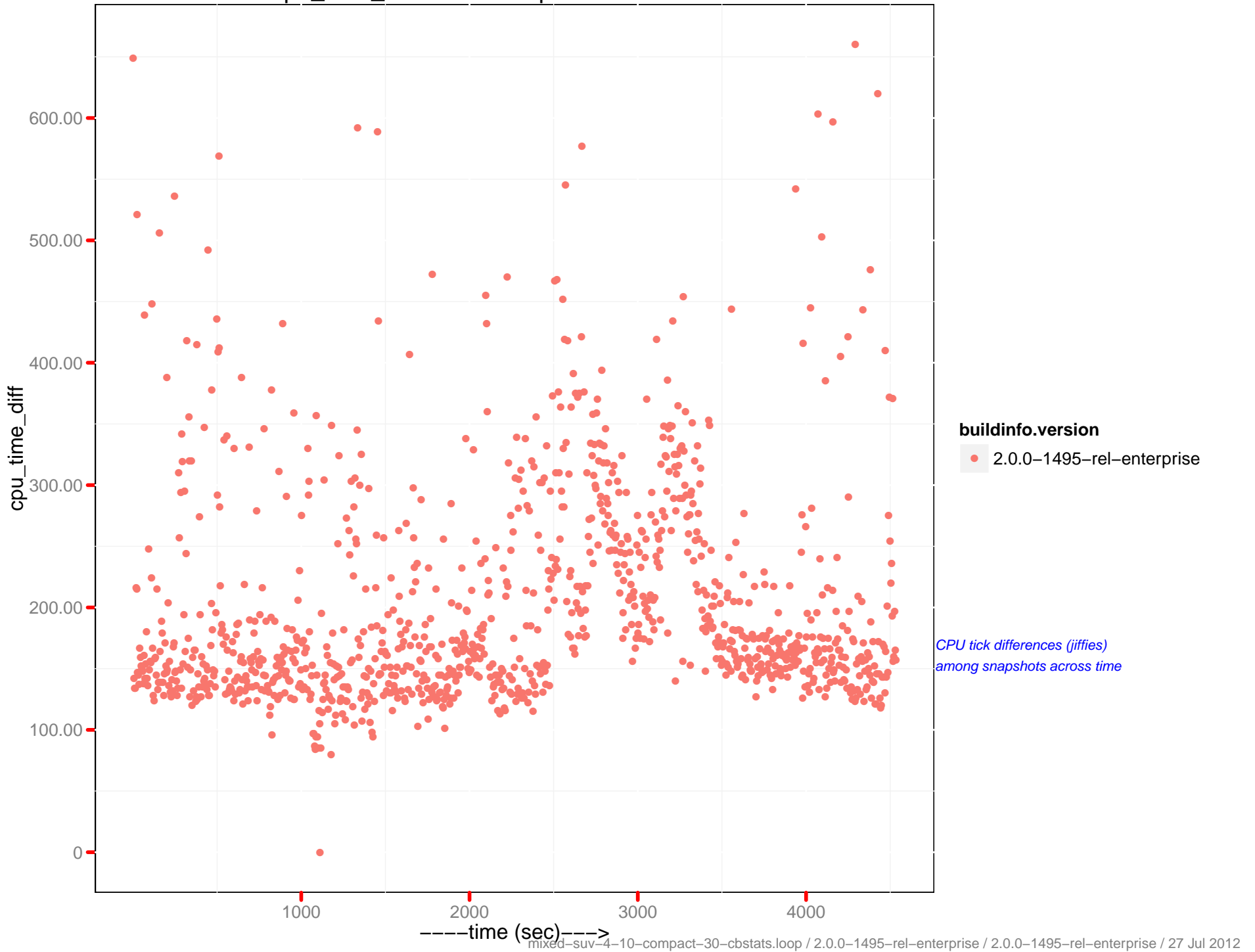




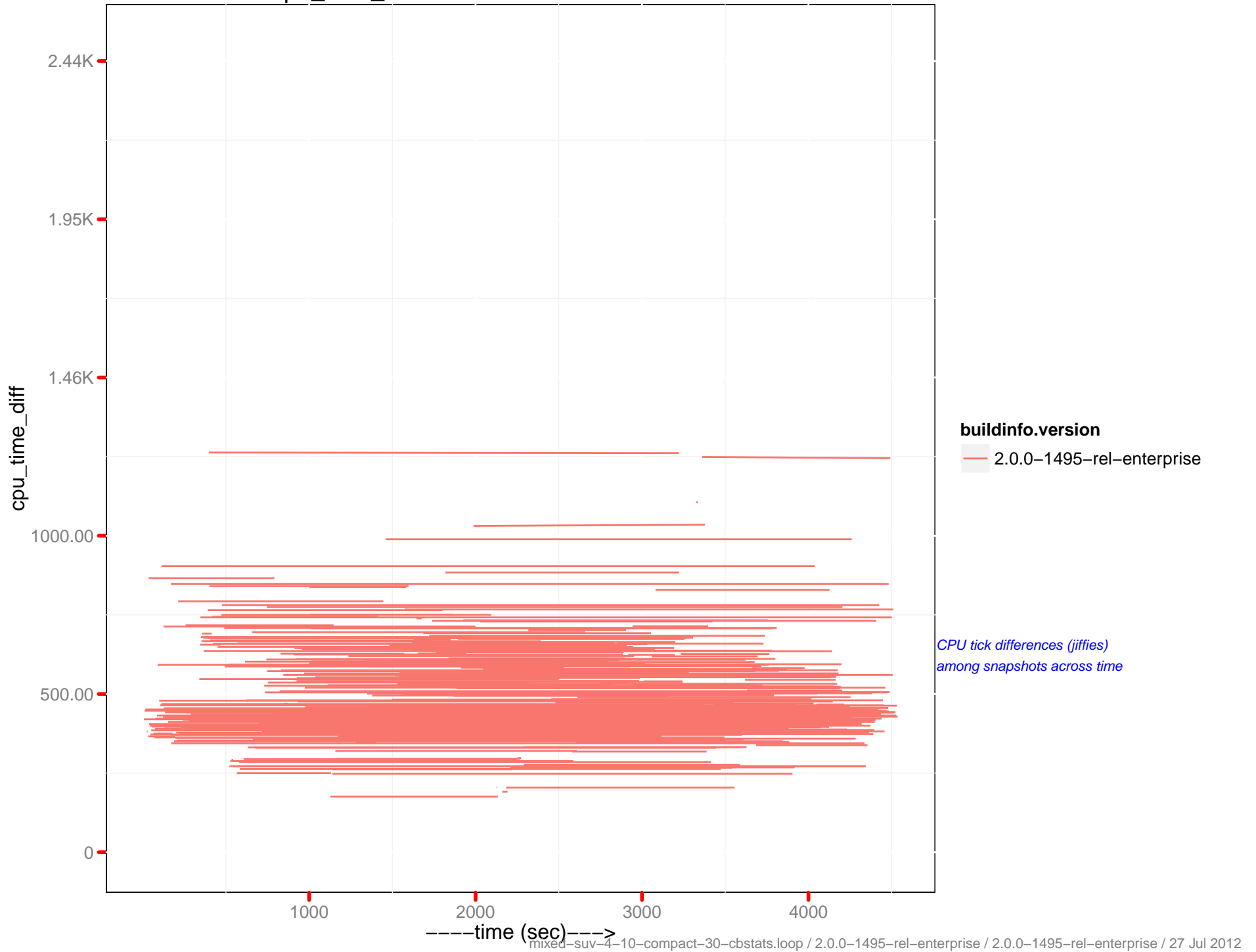
# cpu\_time\_diff: memcached - 192.168.162.20



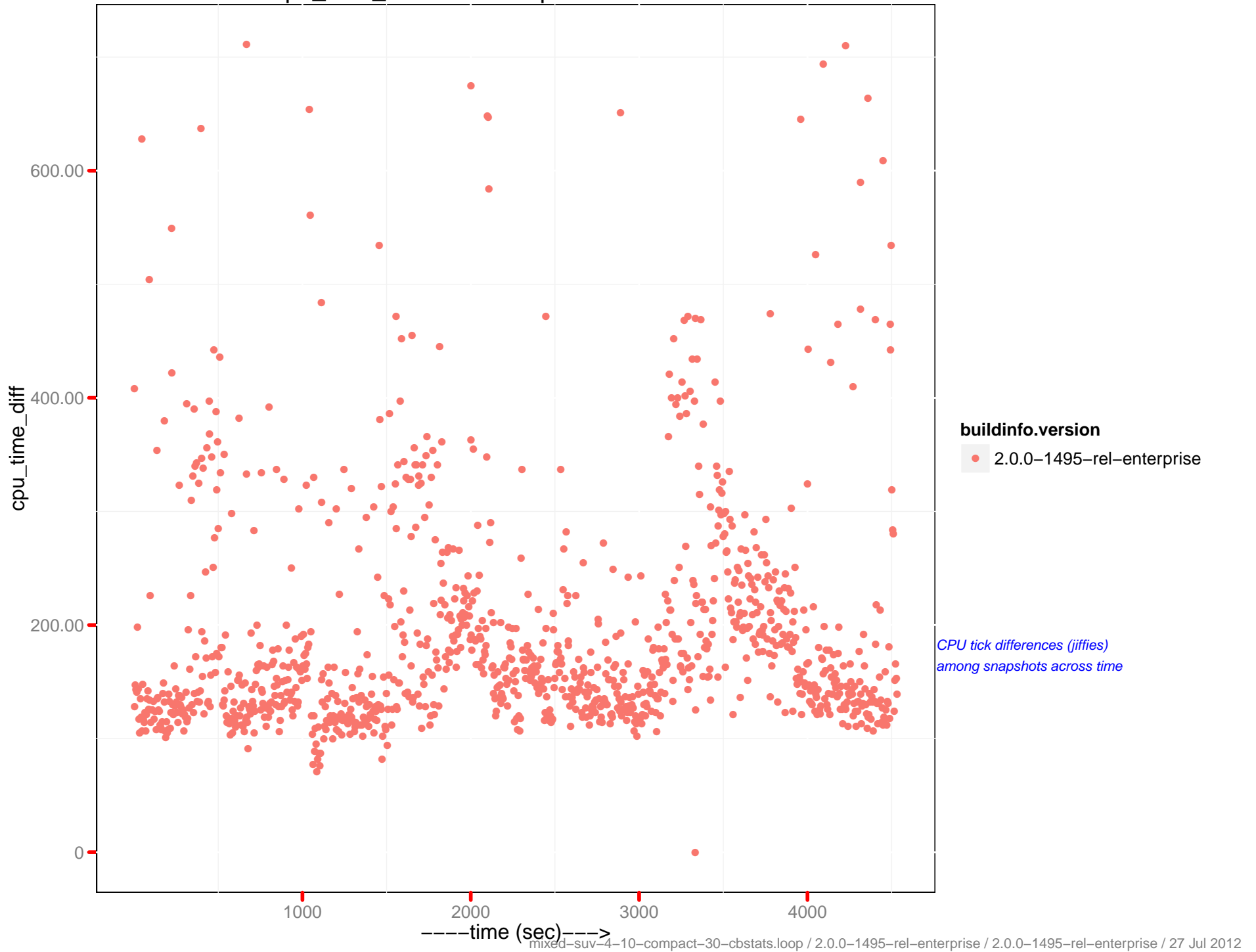
cpu\_time\_diff : beam.smp - 192.168.162.20



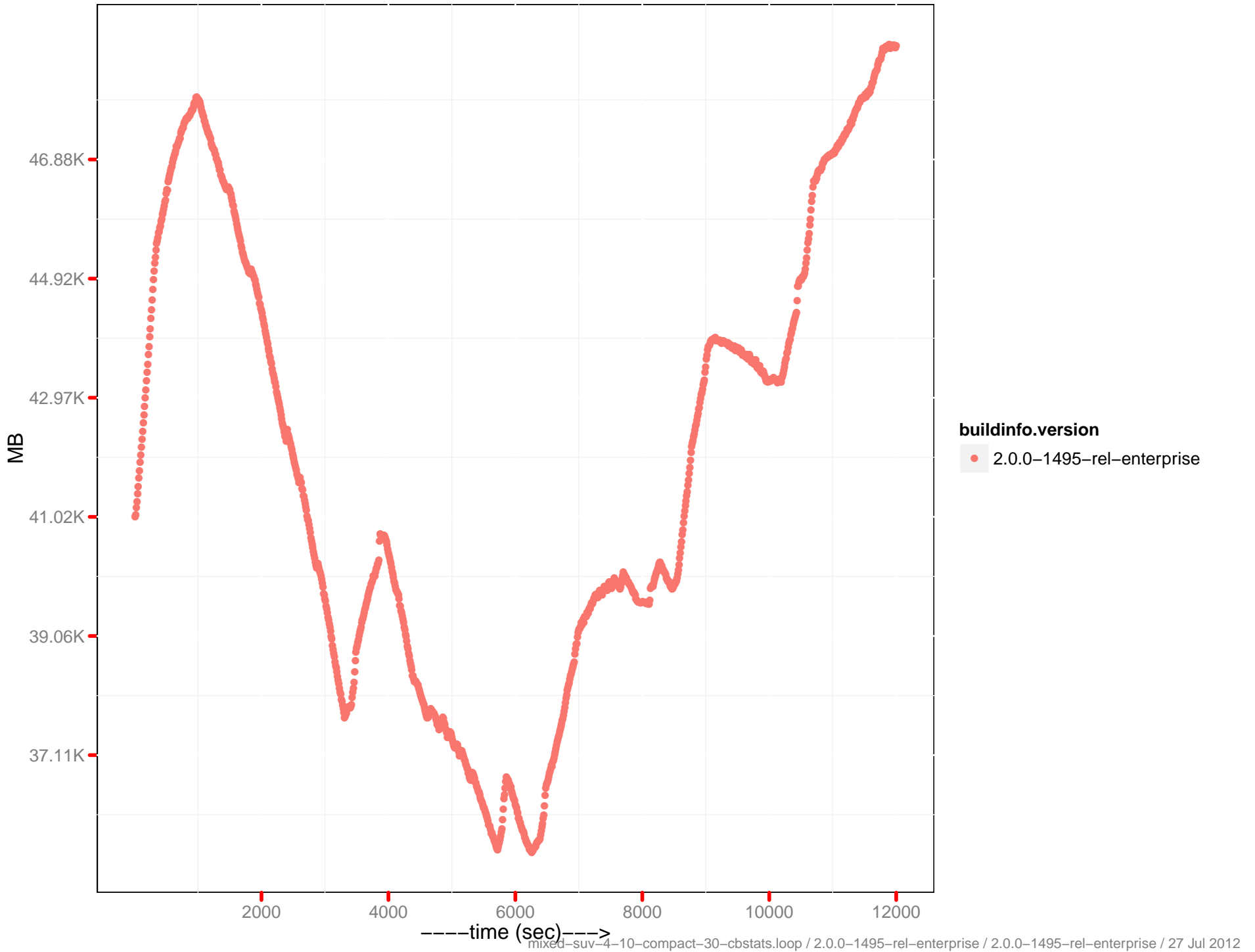
# cpu\_time\_diff: memcached - 192.168.162.21



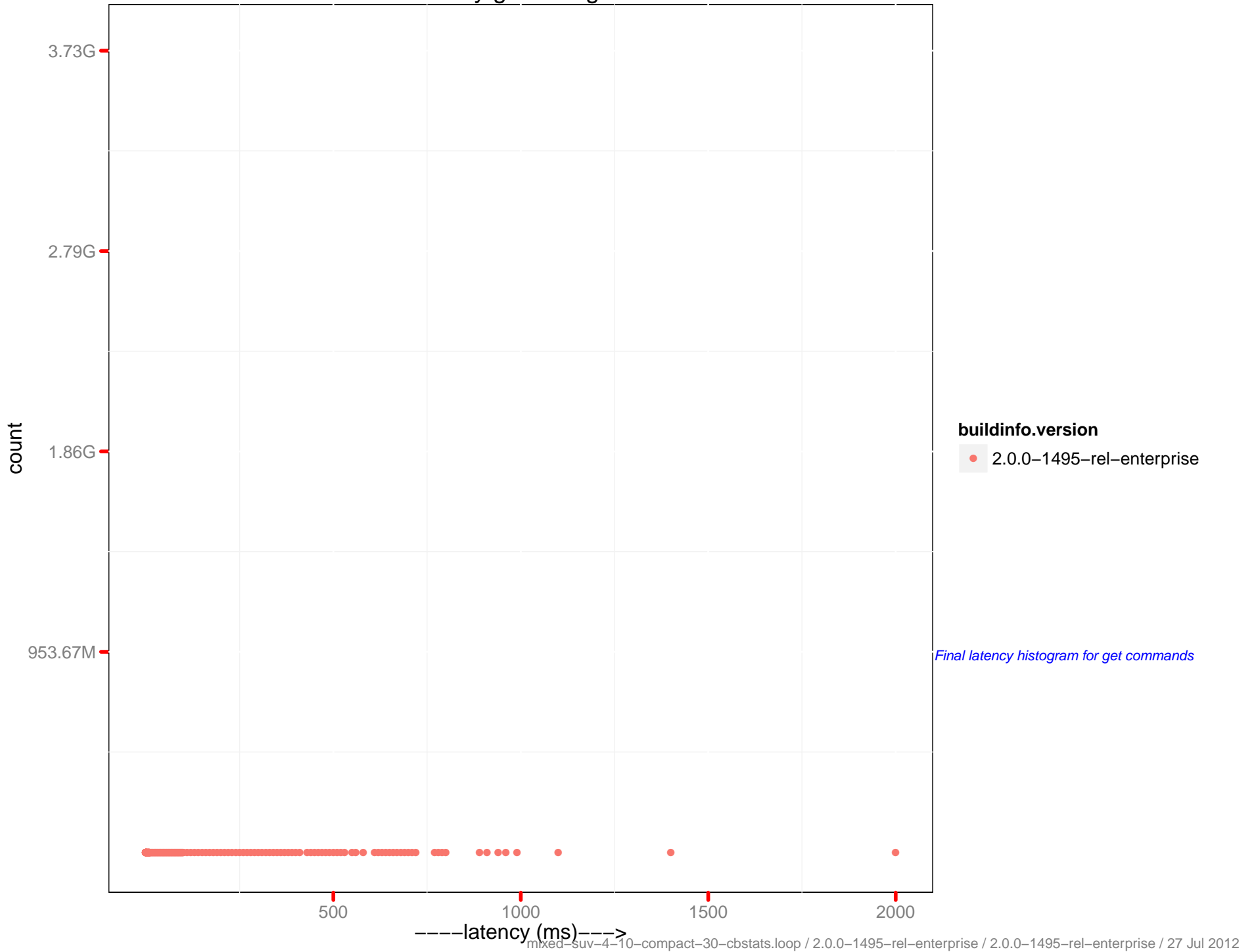
cpu\_time\_diff : beam.smp - 192.168.162.21



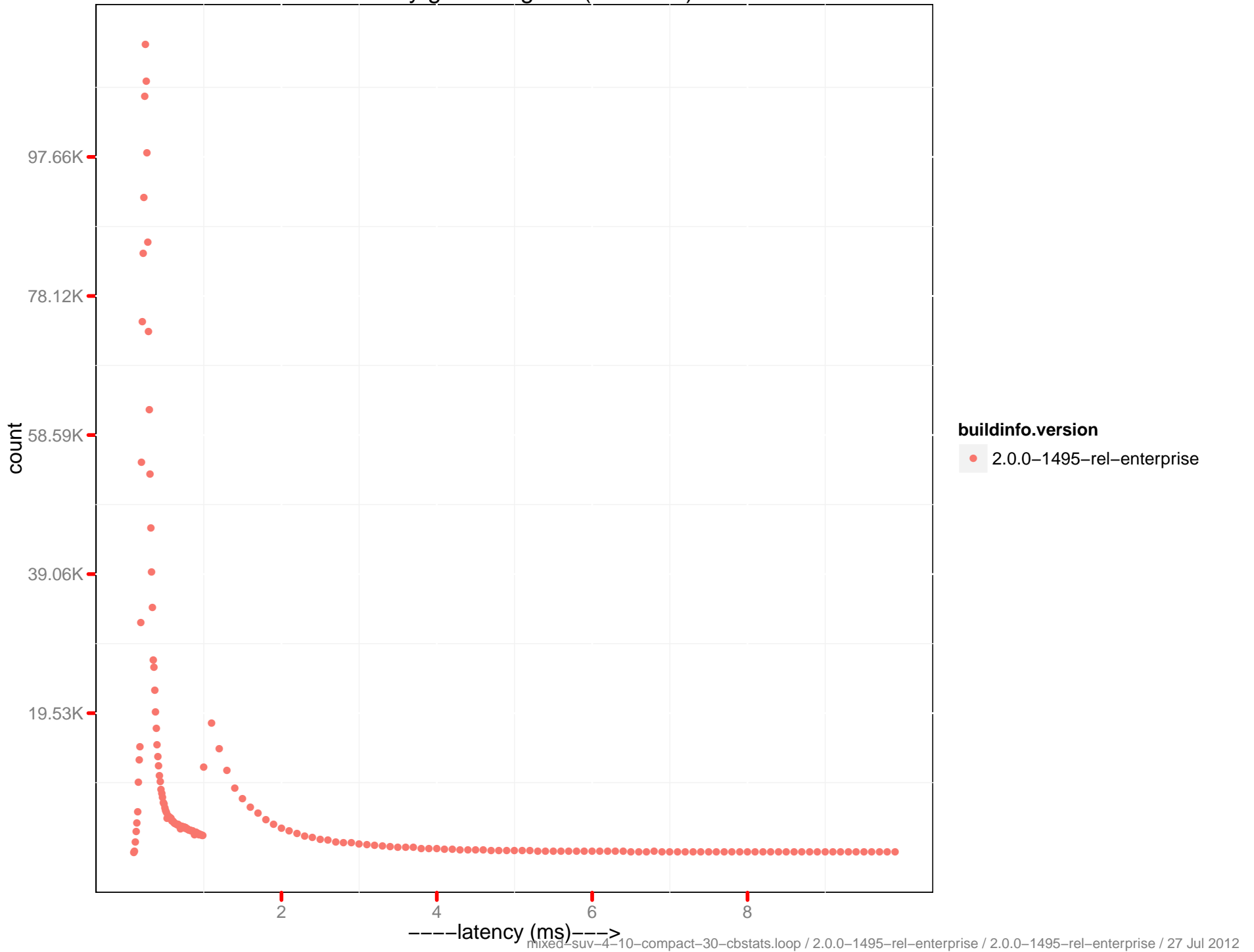
Data disk size



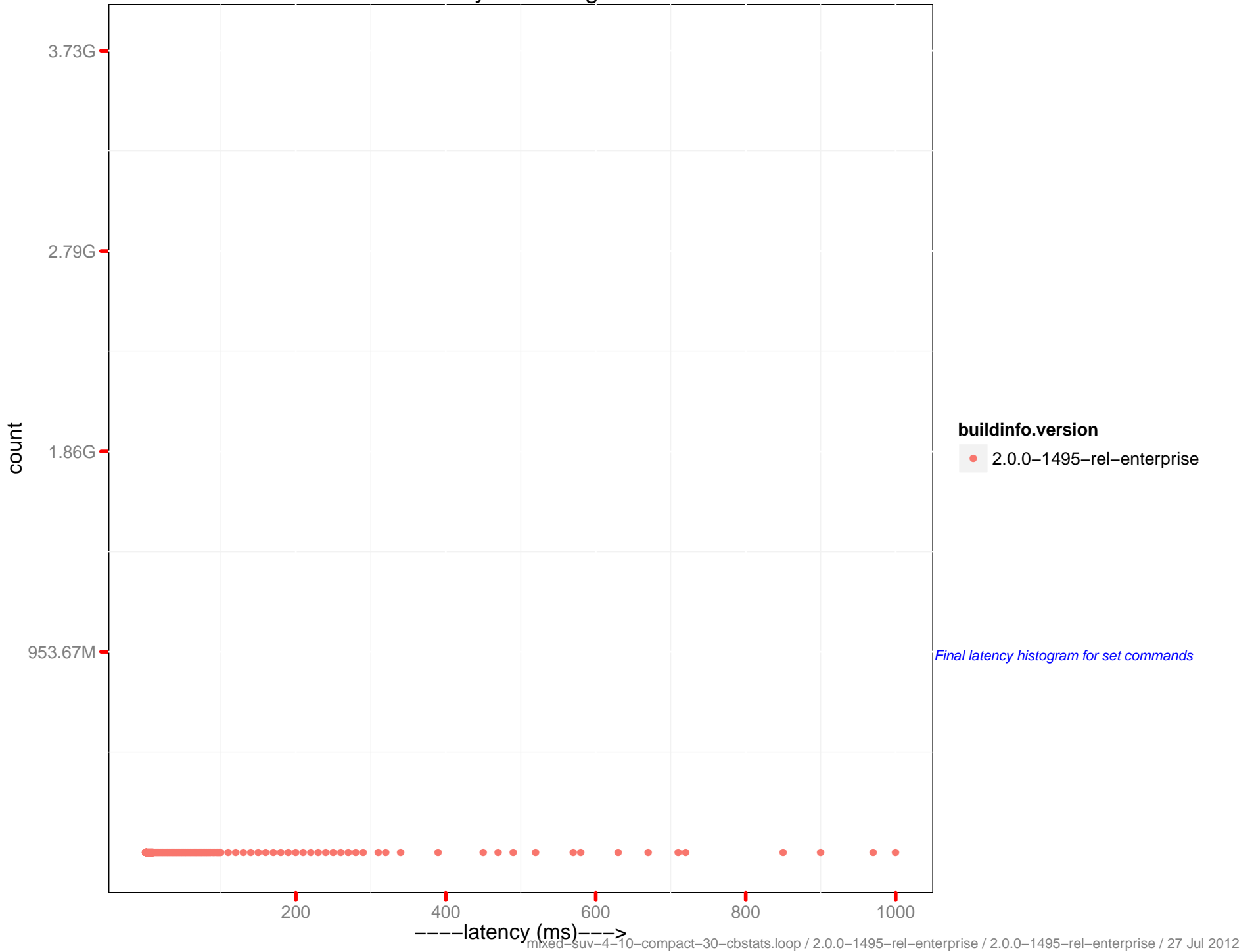
# Latency get histogram



Latency get histogram (0–10 ms)

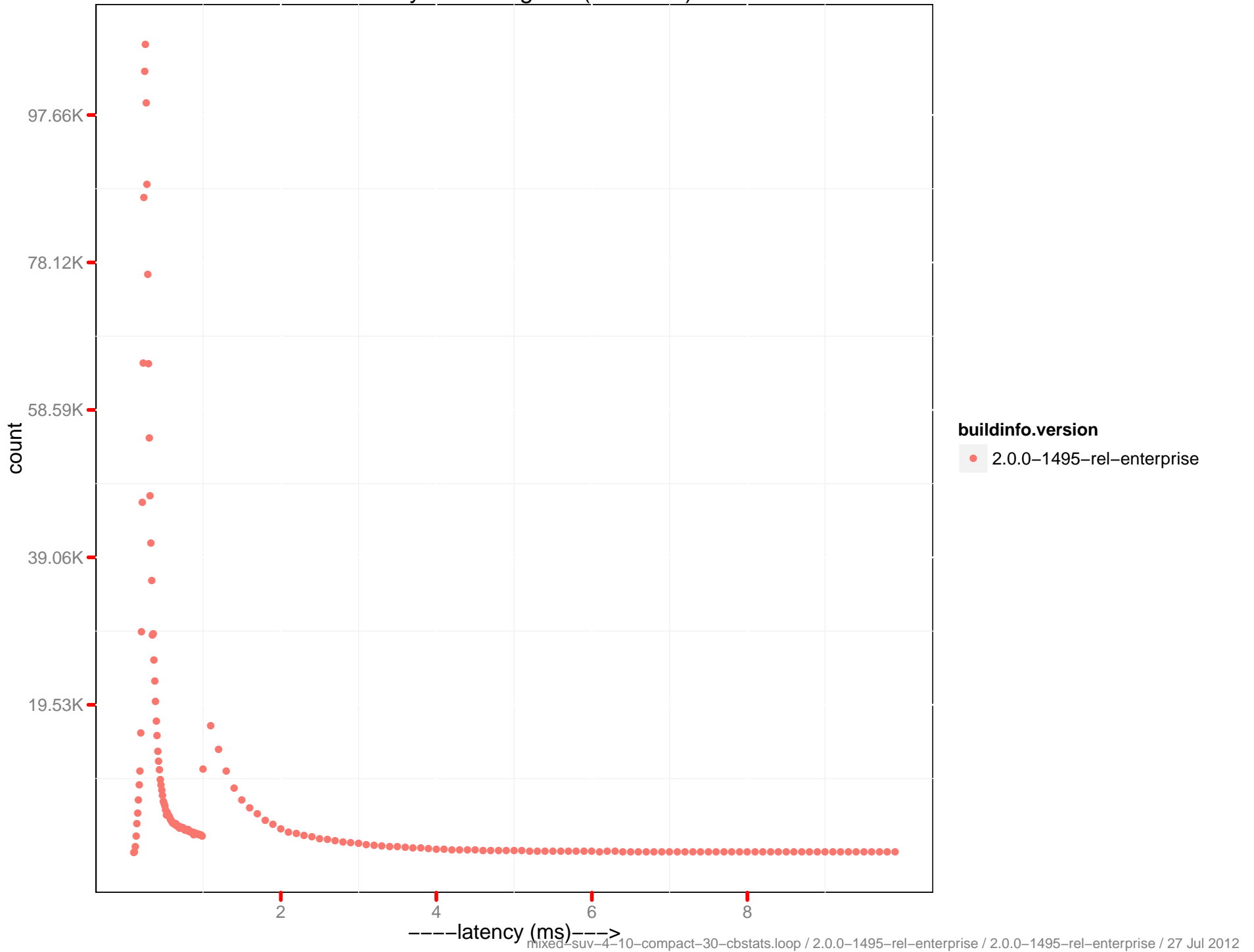


# Latency set histogram



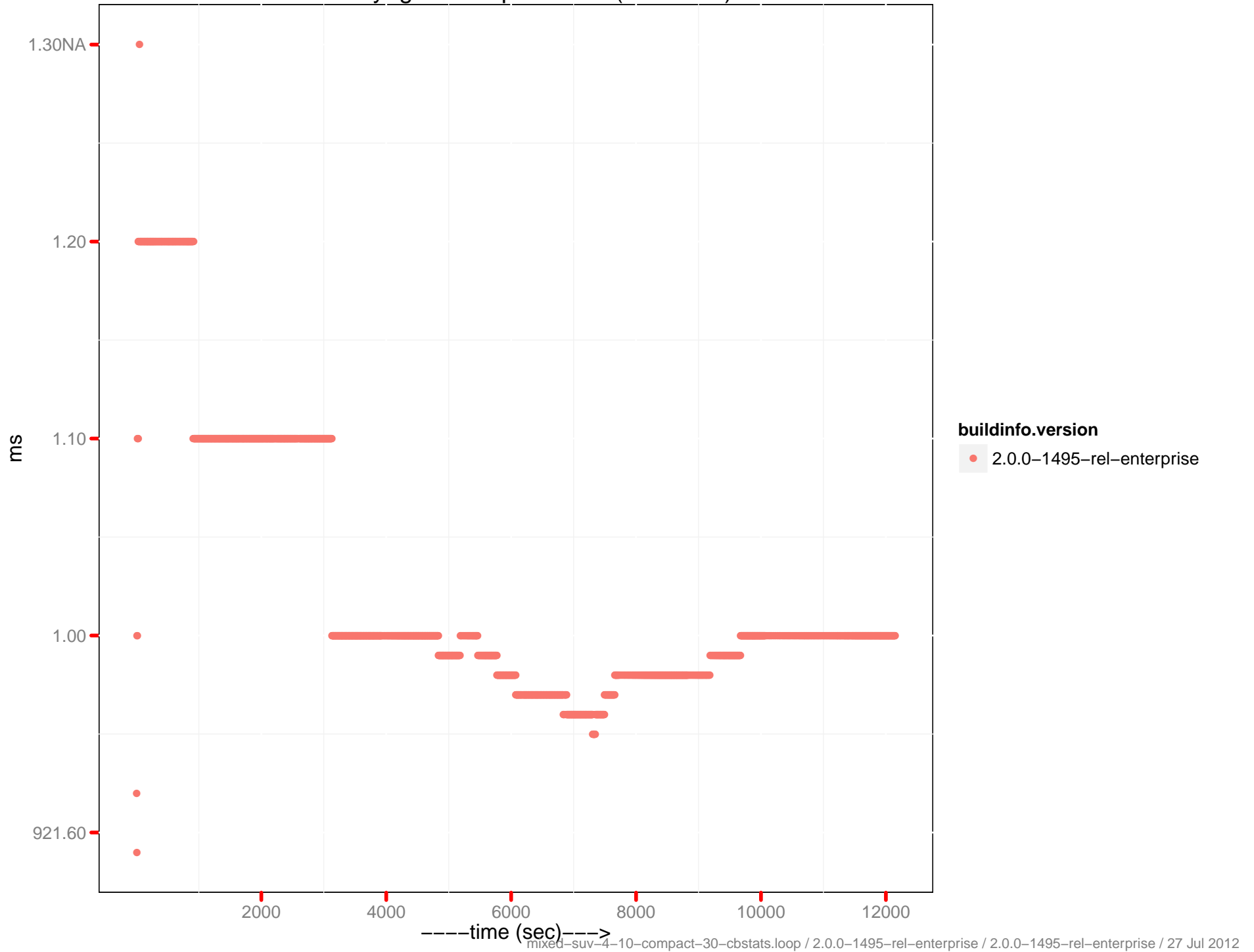


Latency set histogram (0–10 ms)

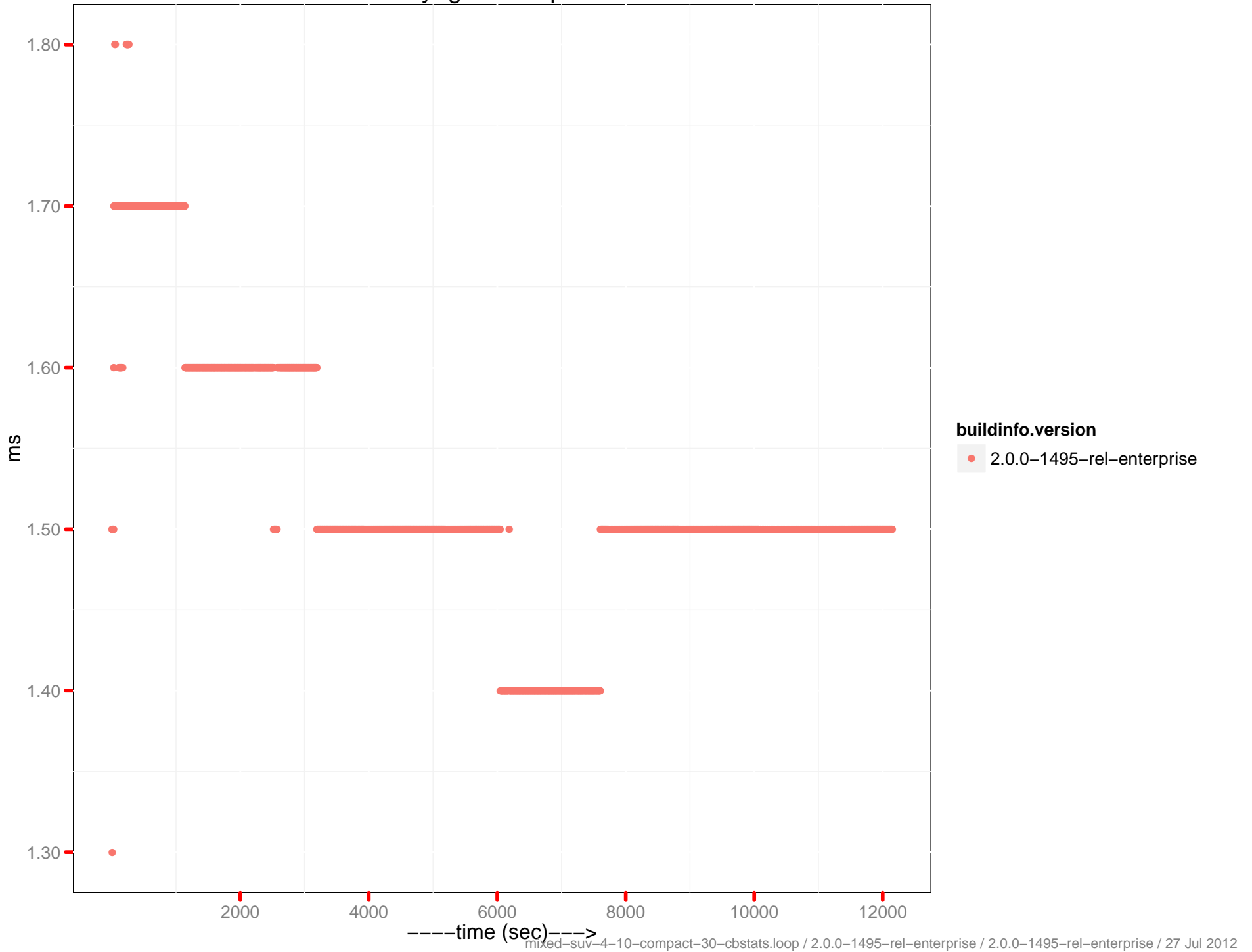




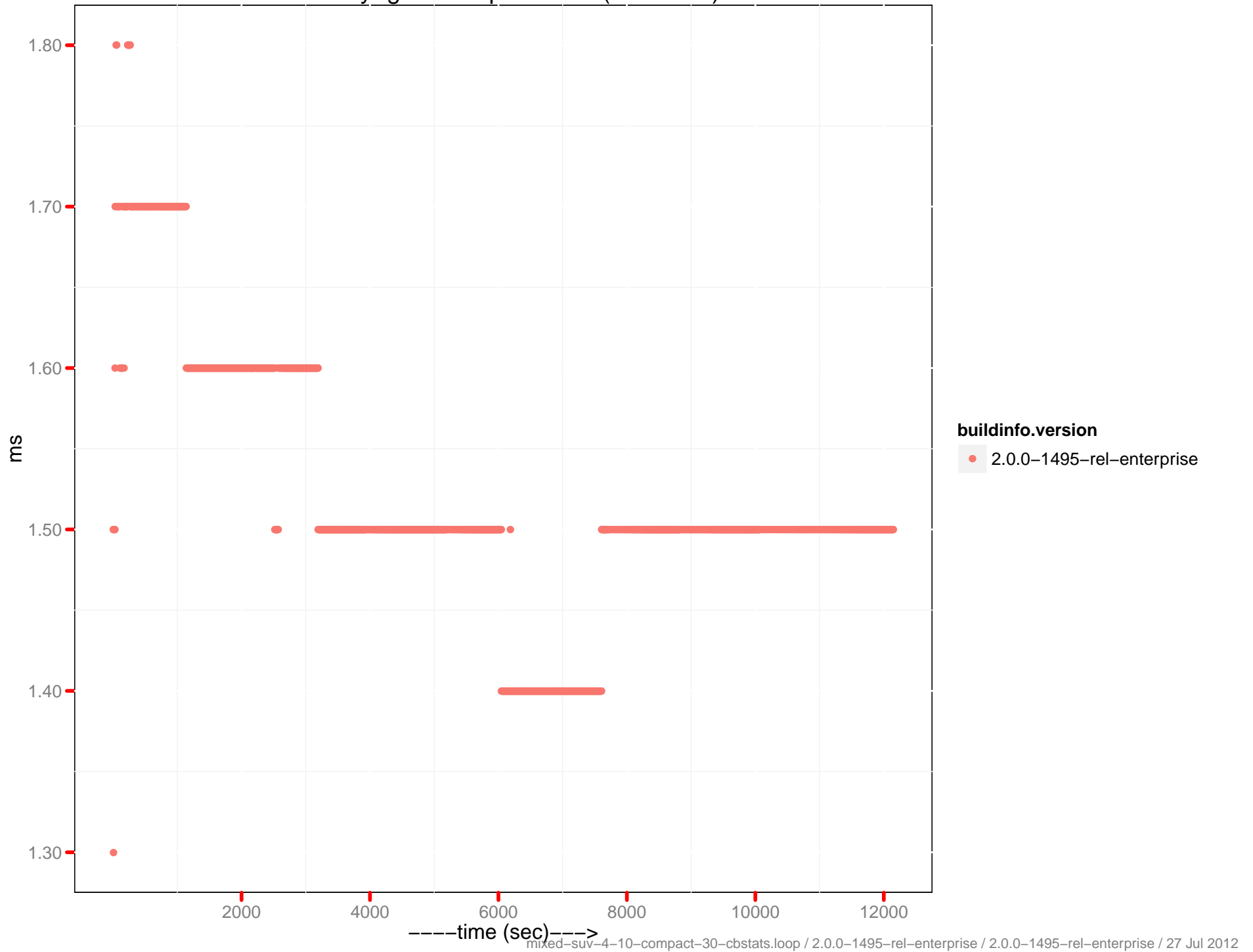
# Latency-get 90th percentile (0 - 10ms)



# Latency-get 95th percentile



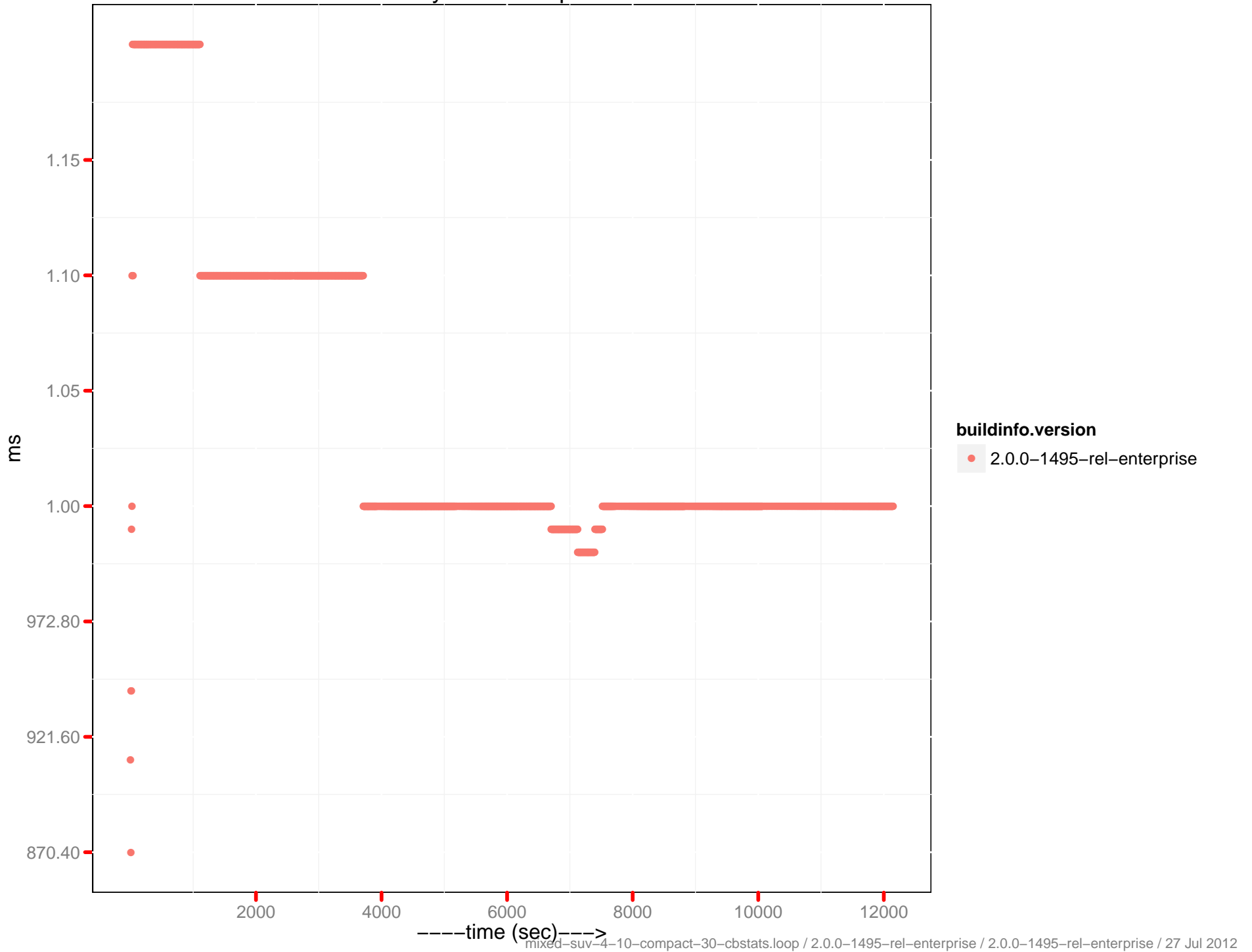
Latency-get 95th percentile (0 - 10ms)





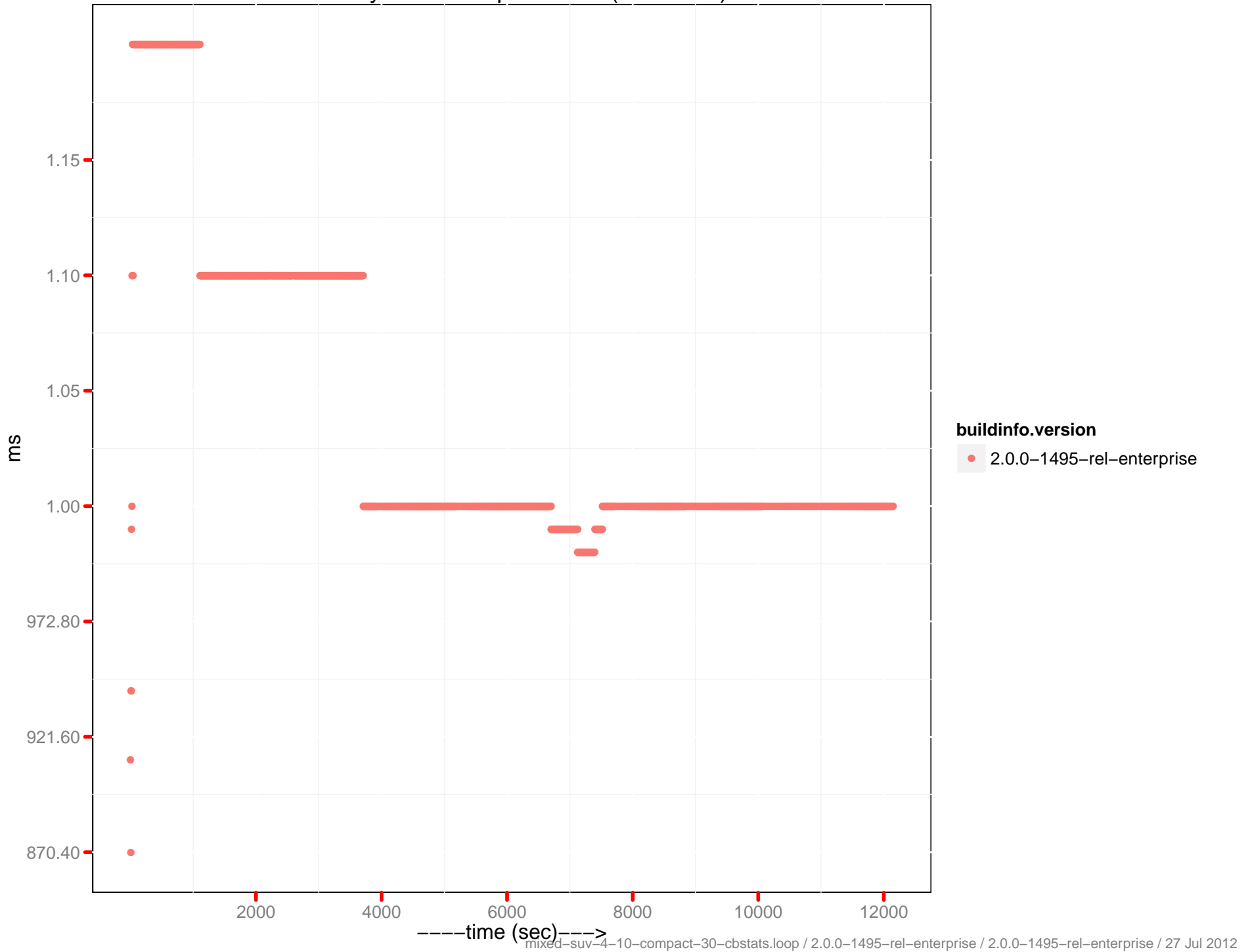


# Latency-set 90th percentile

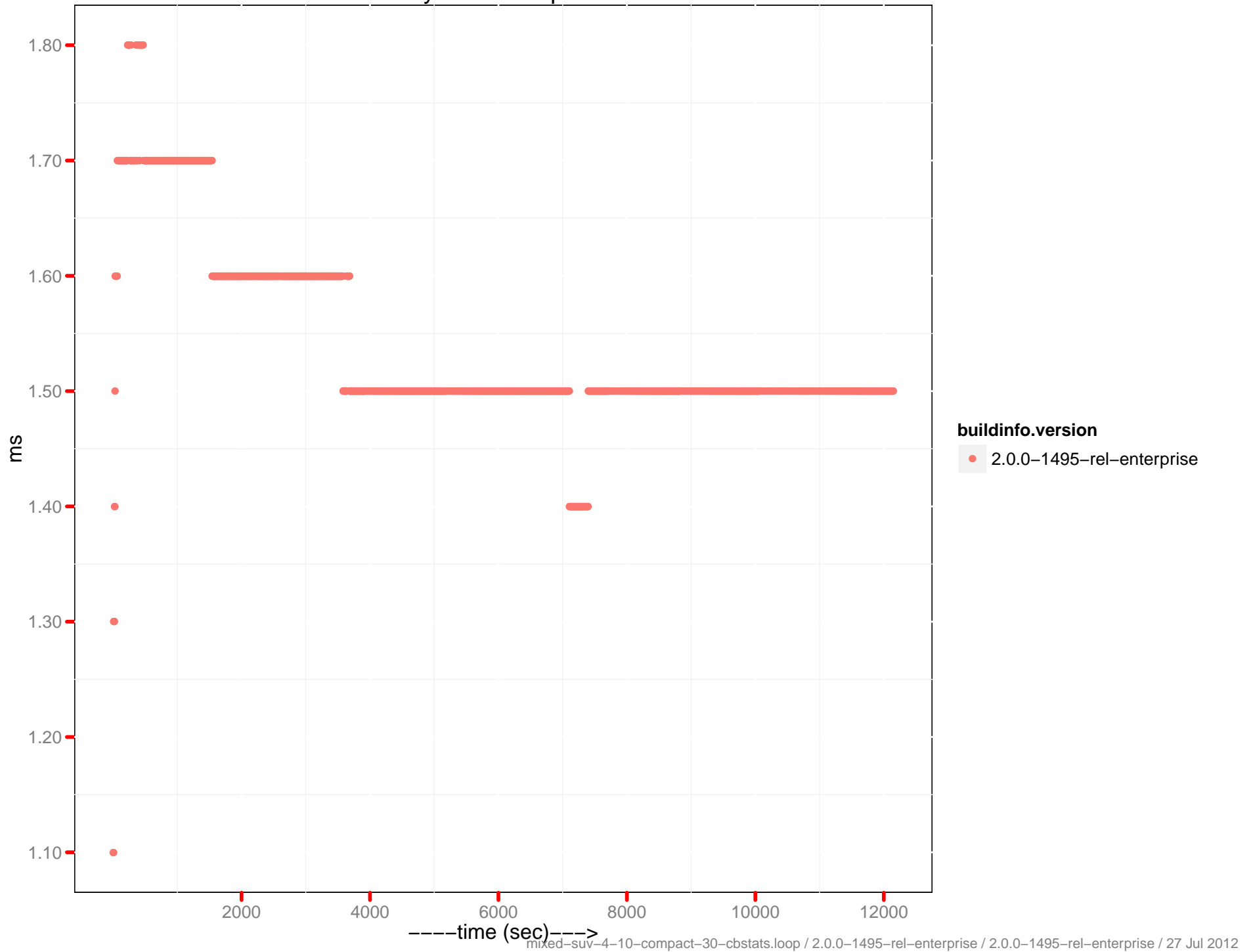




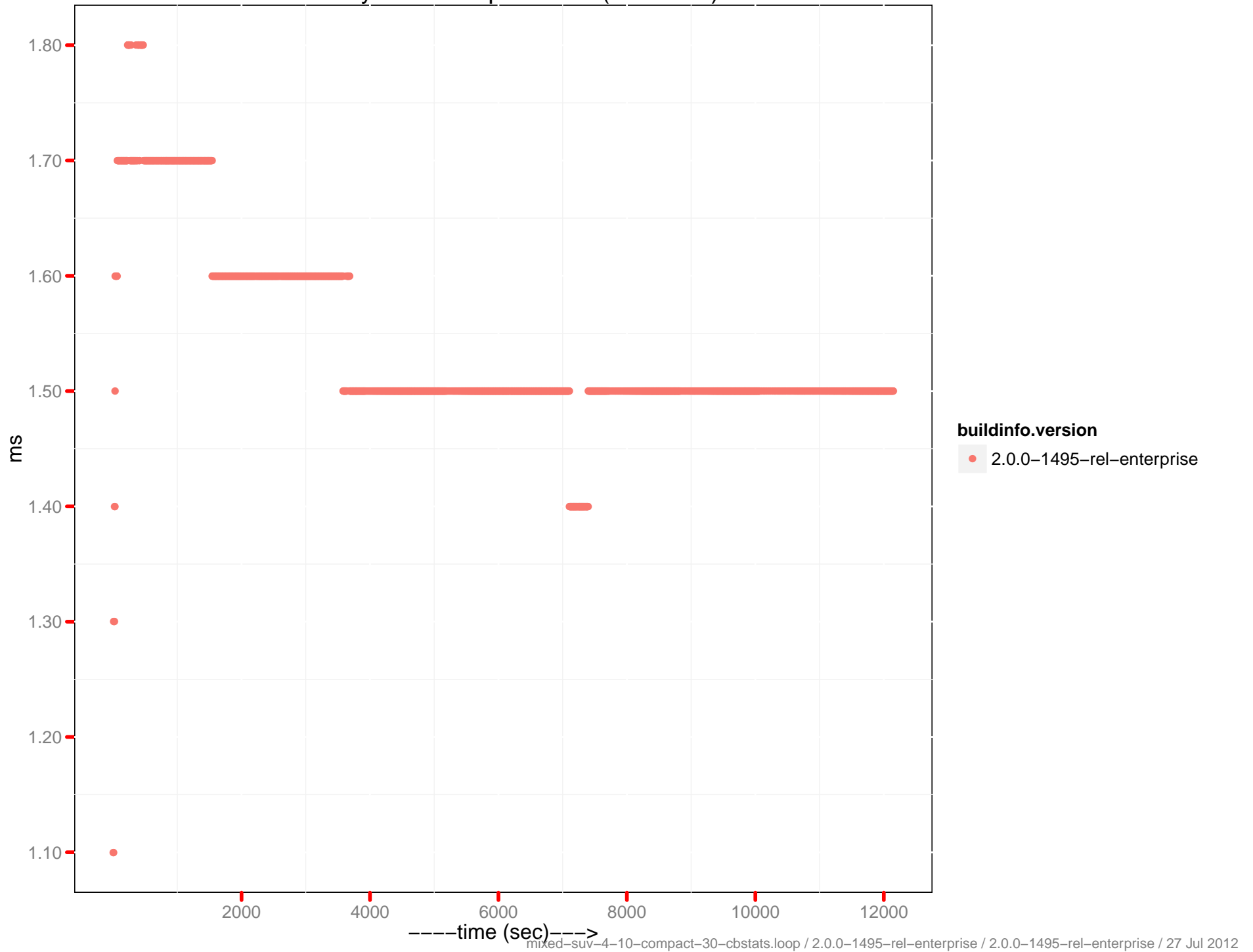
Latency-set 90th percentile (0 - 10ms)



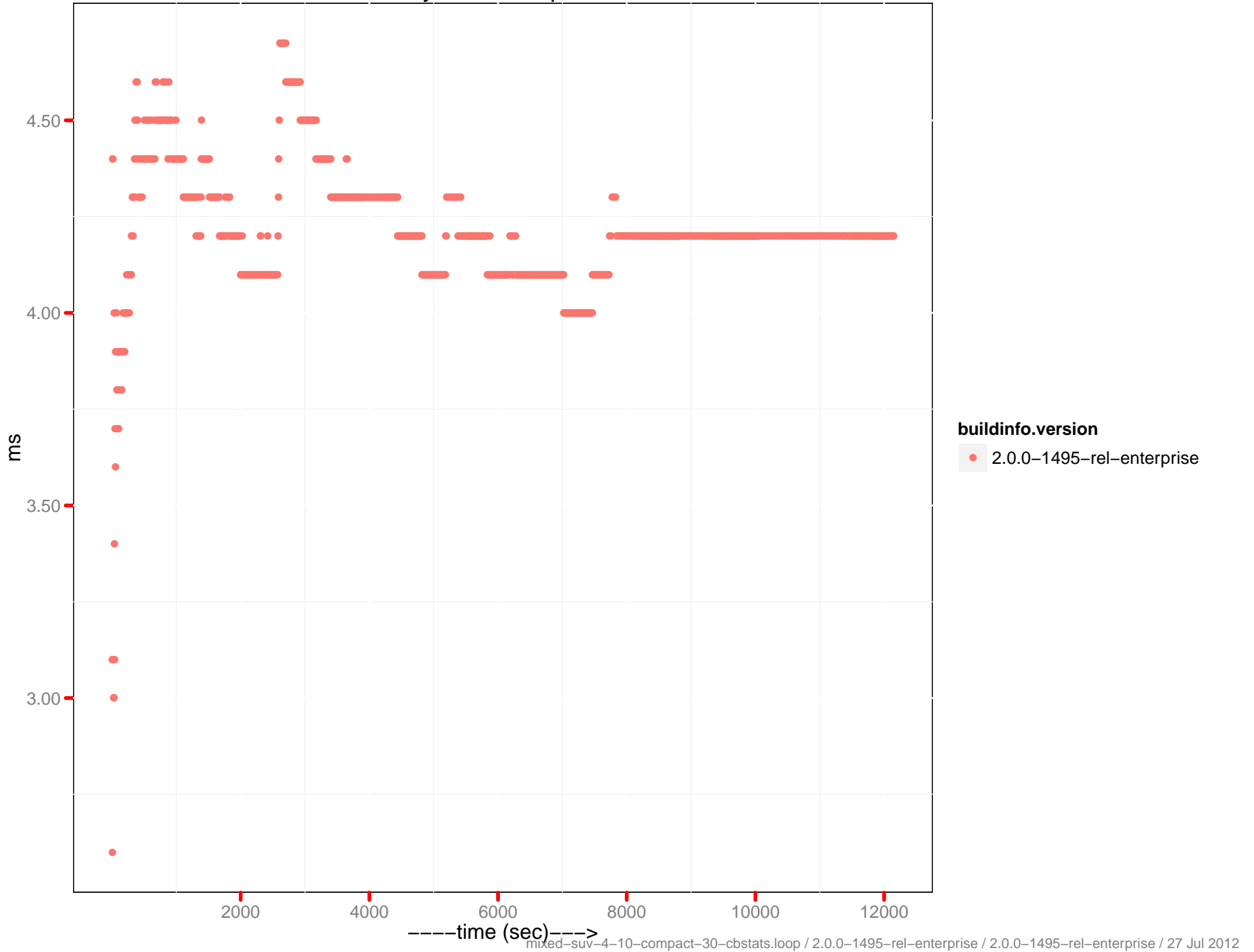
# Latency-set 95th percentile



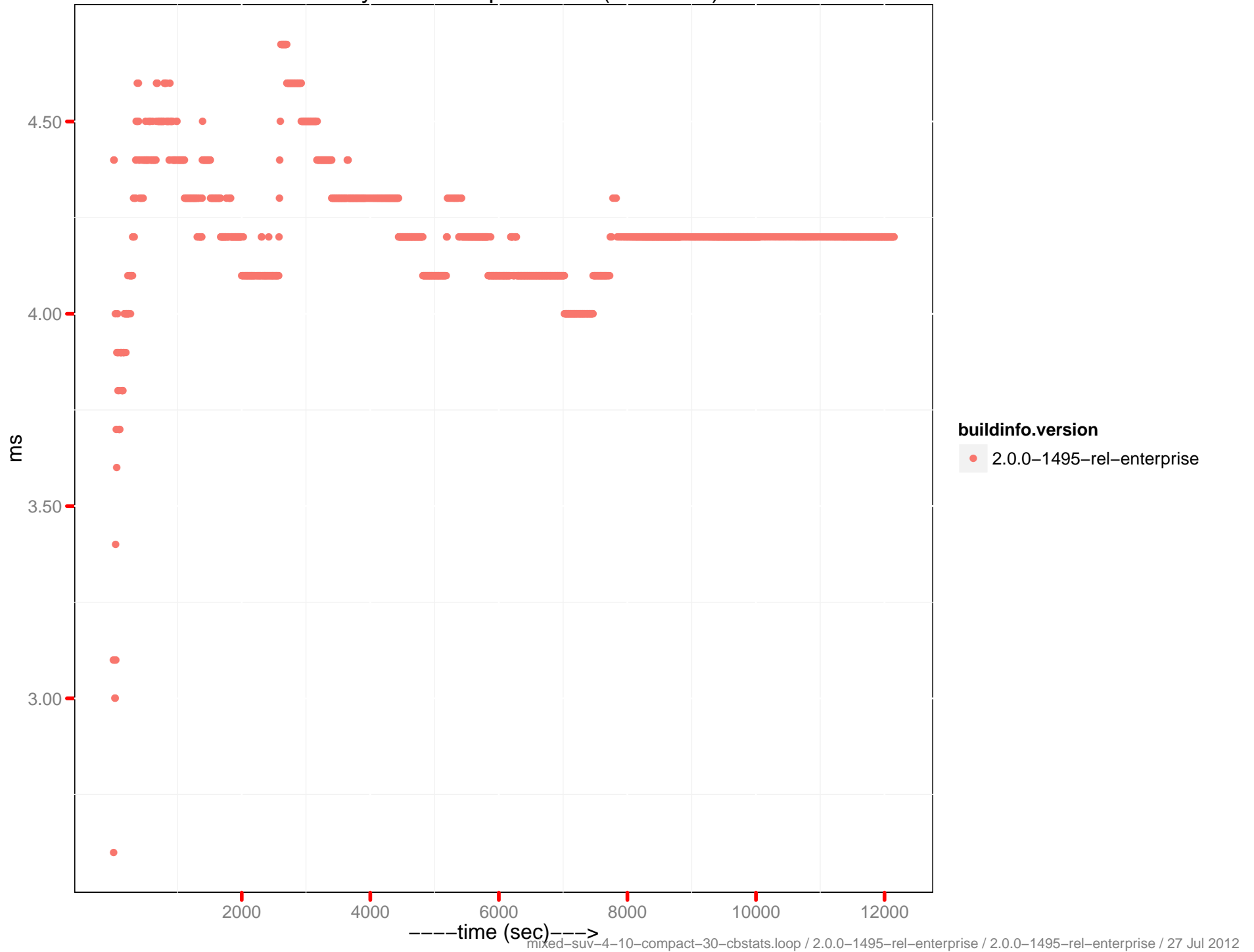
Latency-set 95th percentile (0 - 10ms)



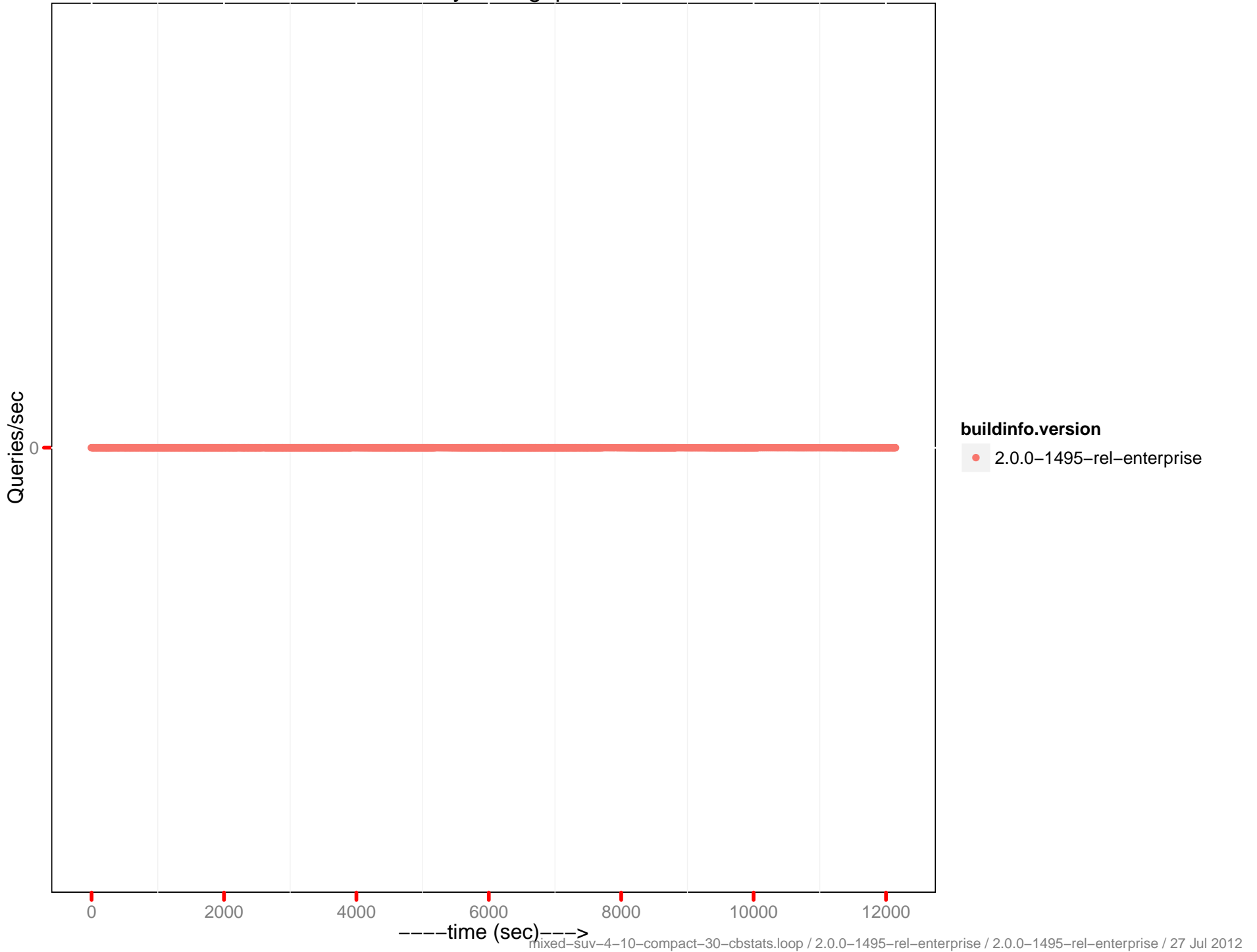
# Latency-set 99th percentile



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



# Query throughput



```
mixed-suv-4-10-compact-30-cbstats.conf
# mixed 7M load, 1M hot reload, 3M access creates
# speed limit = 3k
#
performance.eperf.EPerfClient.test_eperf_mixed

params:

# general
batch=50
kind=nonjson
mem_quota=20000
spec=mixed-suv-4-10-compact-30-cbstats
db_compaction=30

# cbstats collector
cb_stats=1

# load phase
hot_init_items=1000000
items=7000000

# 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=3000000

# control (defaults: pytests/performance/perf_defaults.py)
load_wait_until_drained=1
loop_wait_until_drained=0
mcsoda_heartbeat=3
mcsoda_max_ops_sec=3000
tear_down=1
tear_down_proxy=1
tear_down_bucket=0
tear_down_cluster=1
tear_down_on_setup=0
```

hummer-dedicated-1.ini

[global]

username:root

password:couchbase

port:8091

data\_path:/data

[servers]

1:192.168.162.20

2:192.168.162.21

[clients]

1:192.168.162.24

2:192.168.162.25

3:192.168.162.26

[membase]

rest\_username:Administrator

rest\_password:password

[dashboard]

1:dashboard.hq.couchbase.com:80