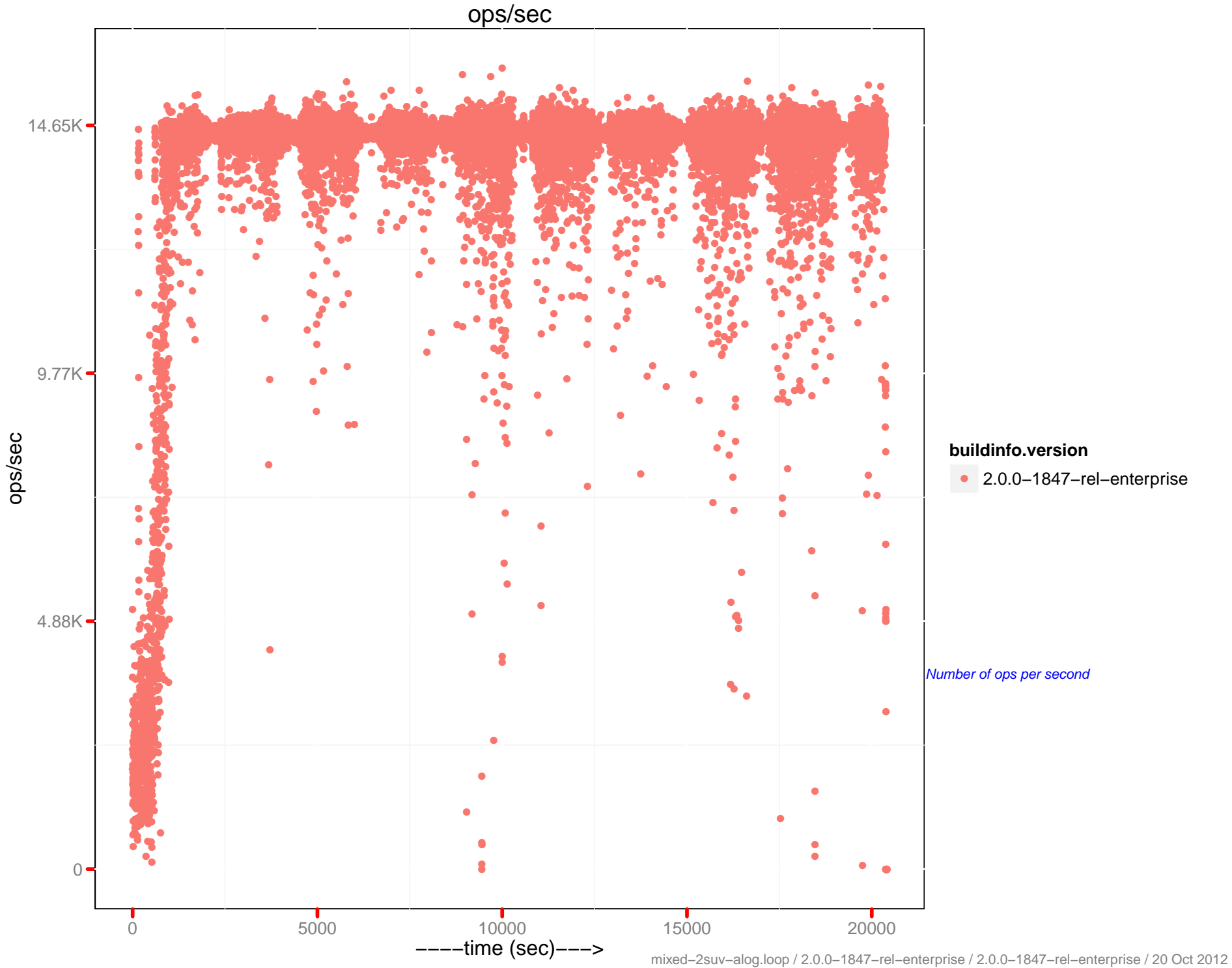
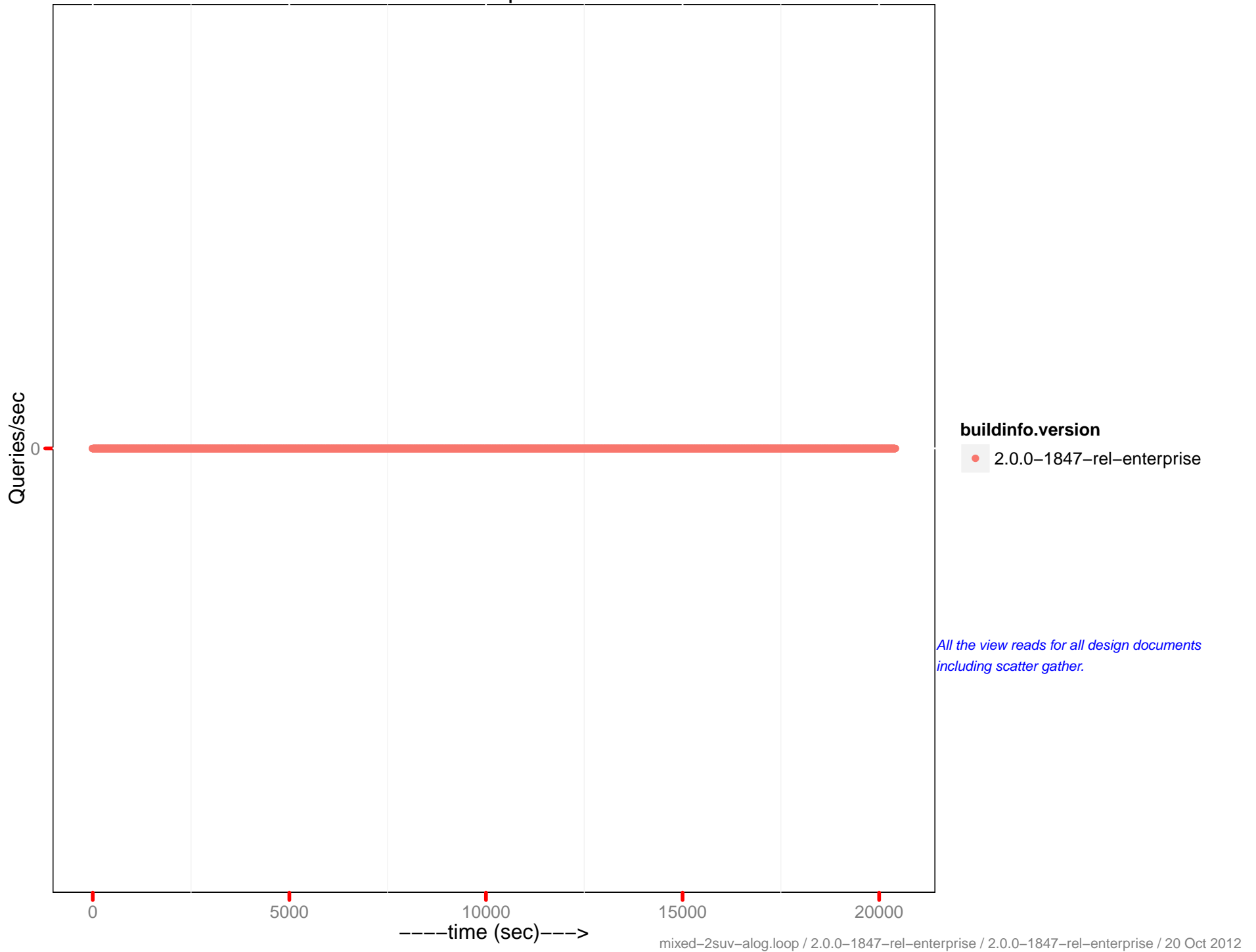


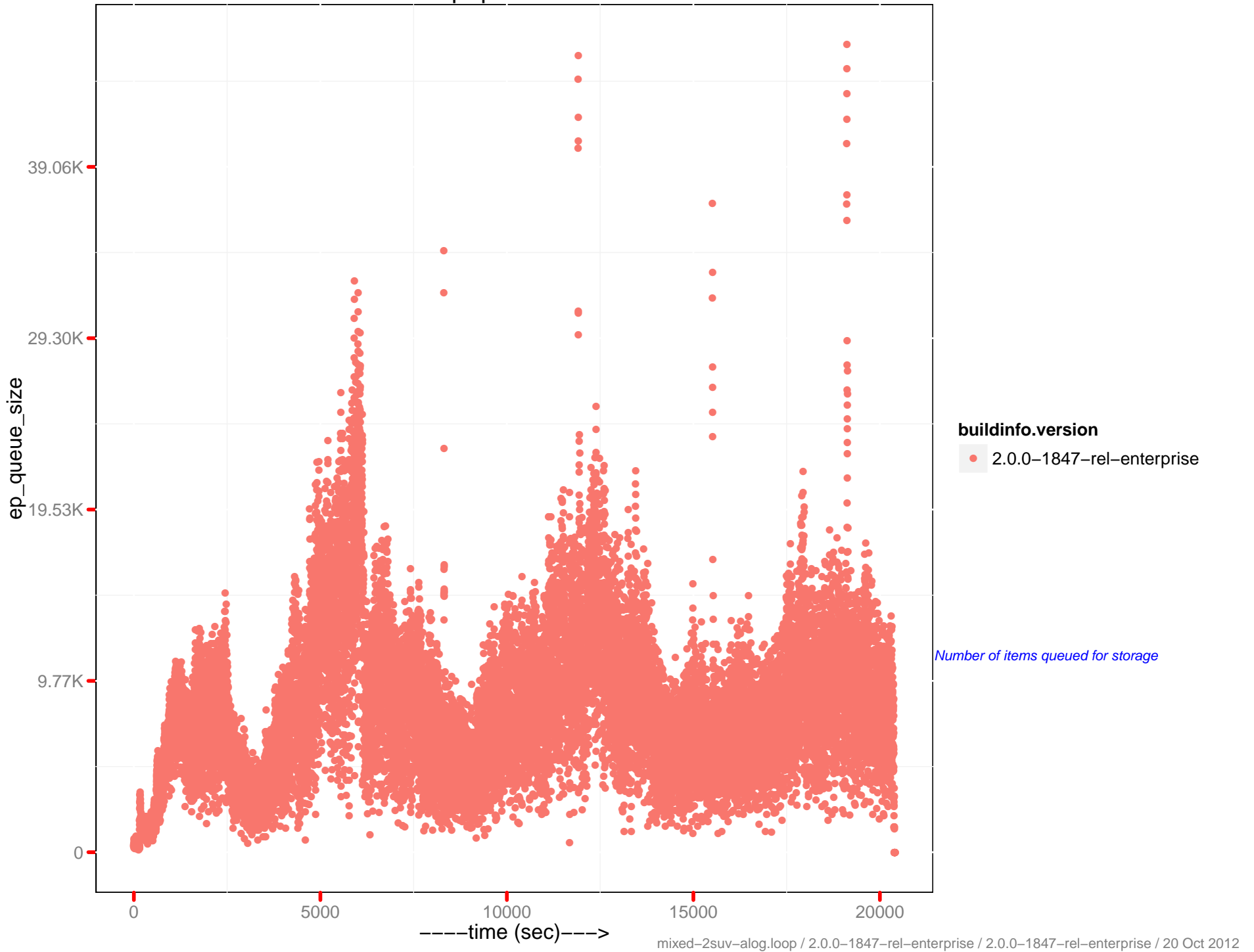
	2.0.0 – 1847	2.0.0 – 1847
<i>Runtime (in hr)</i>	5.67	NA
<i>Avg. Drain Rate</i>	1.35K	NANA
<i>Peak Disk (GB)</i>	123.61	NA
<i>Peak Memory (GB)</i>	16.55	NA
<i>Avg. OPS</i>	13.89K	NANA
<i>Avg. mem memcached (GB)</i>	16.21	NA
<i>Avg. mem beam.smp (MB)</i>	300.77	NA
<i>Avg. CPU rate (%)</i>	11.22	NA
<i>Latency-get (90th) (ms)</i>	0.97	NA
<i>Latency-get (95th) (ms)</i>	2.23	NA
<i>Latency-get (99th) (ms)</i>	21.31	NA
<i>Latency-set (90th) (ms)</i>	1.13	NA
<i>Latency-set (95th) (ms)</i>	1.85	NA
<i>Latency-set (99th) (ms)</i>	4.2	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>Avg. XDC ops/sec</i>	NaN	NA
<i>Avg. XDC queue</i>	NaN	NA
<i>Rebalance Time (sec)</i>	0	NA
<i>Testrunner Version</i>	f199912	NA



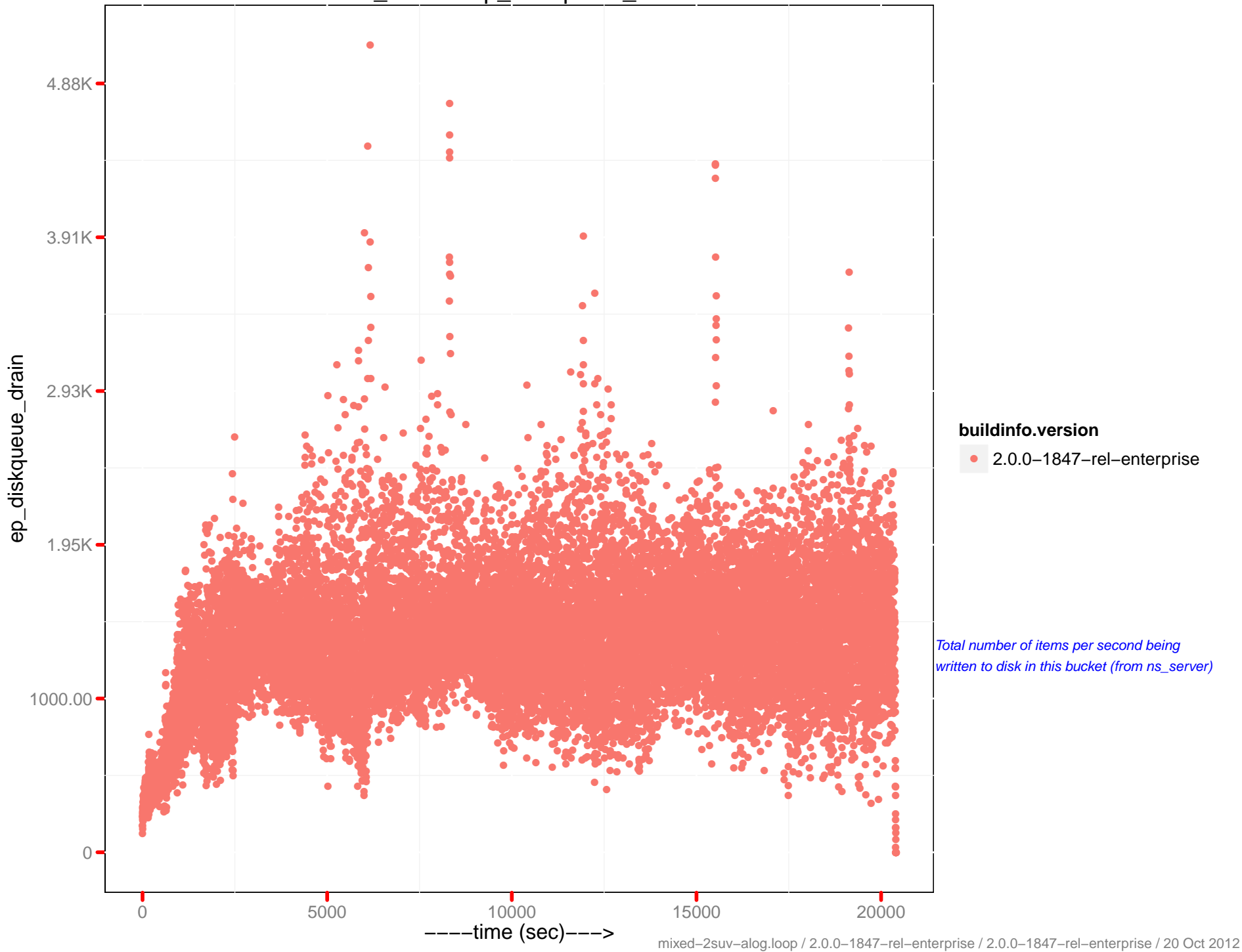
# View read per sec.



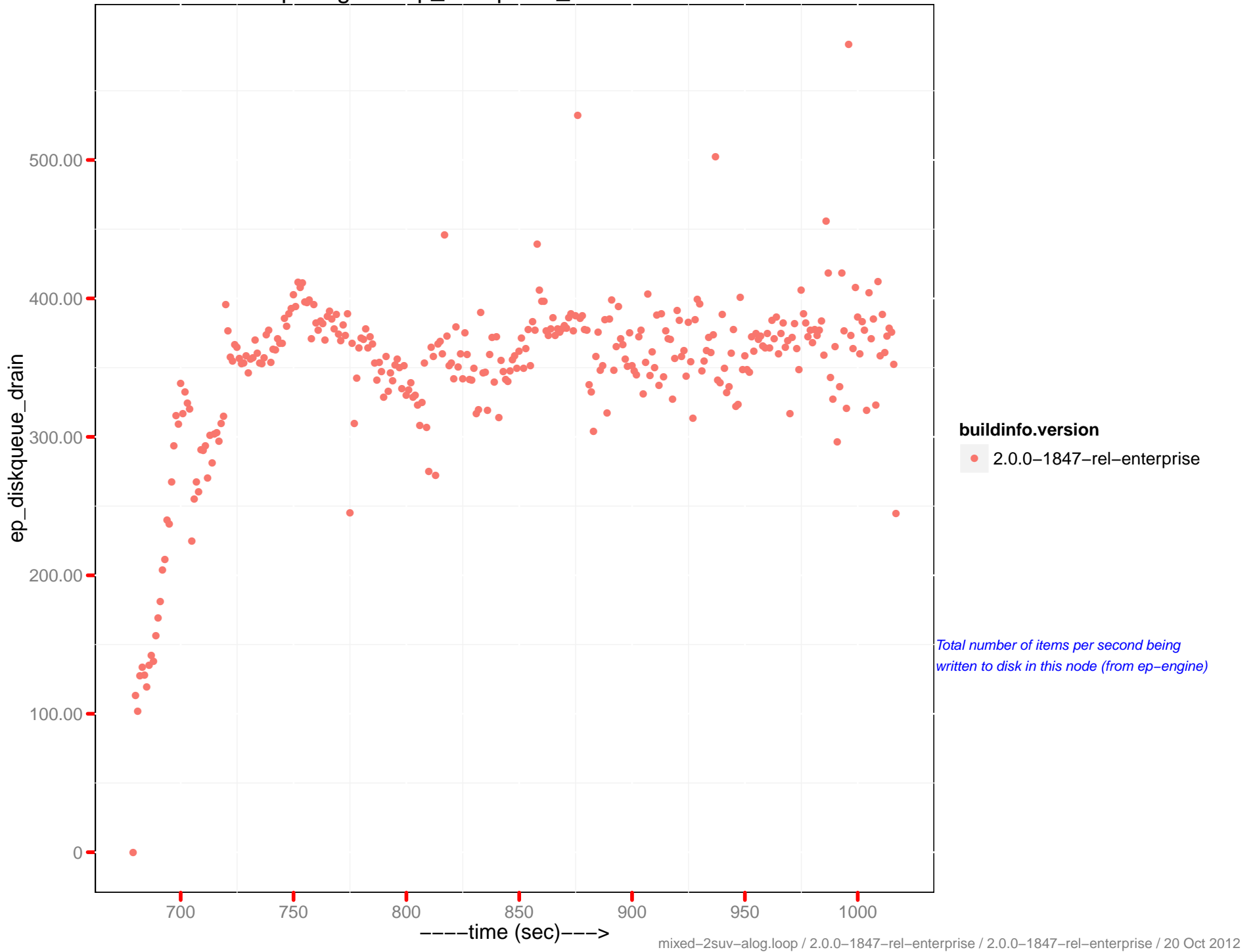
# ep queue size



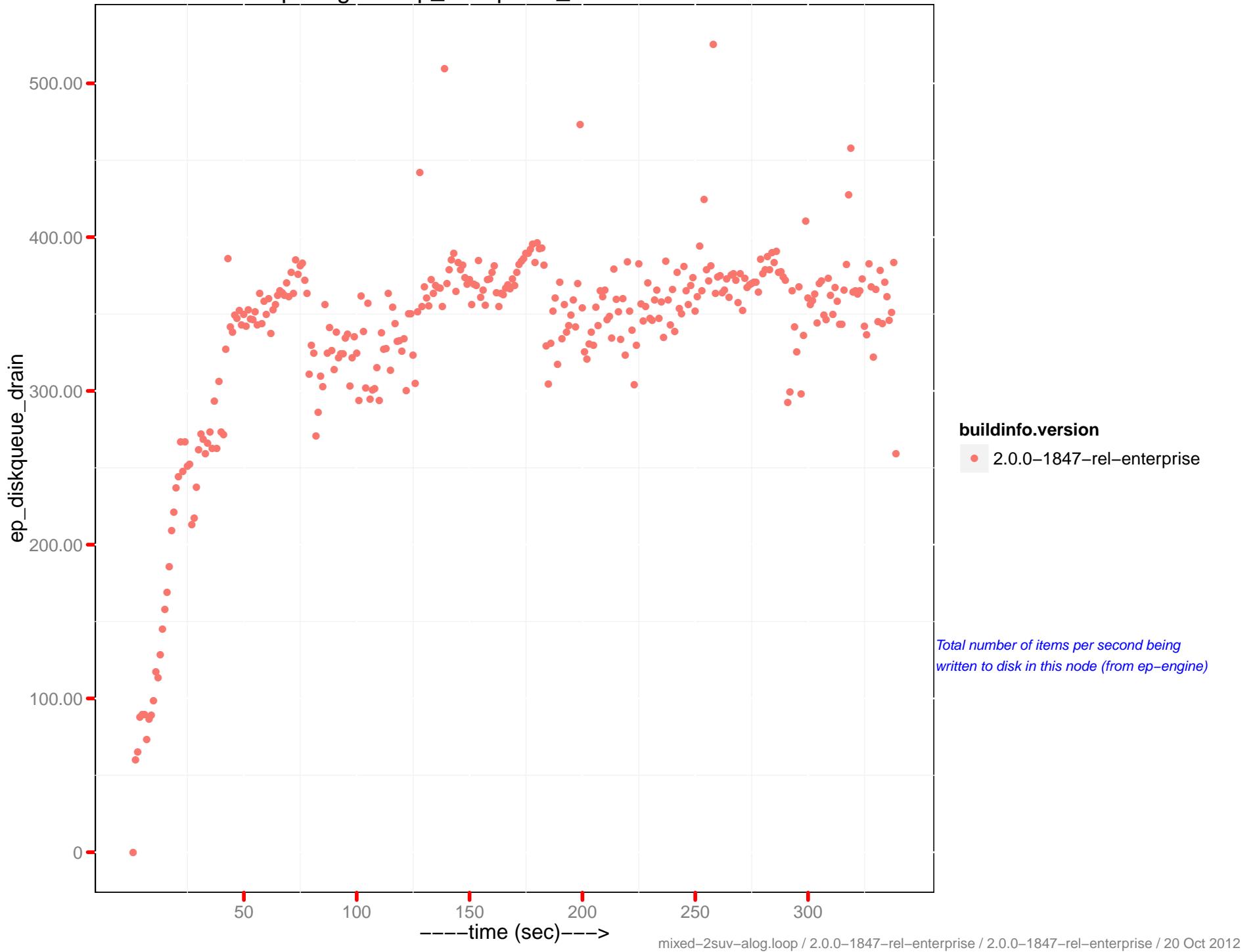
# ns\_server: ep\_diskqueue\_drain



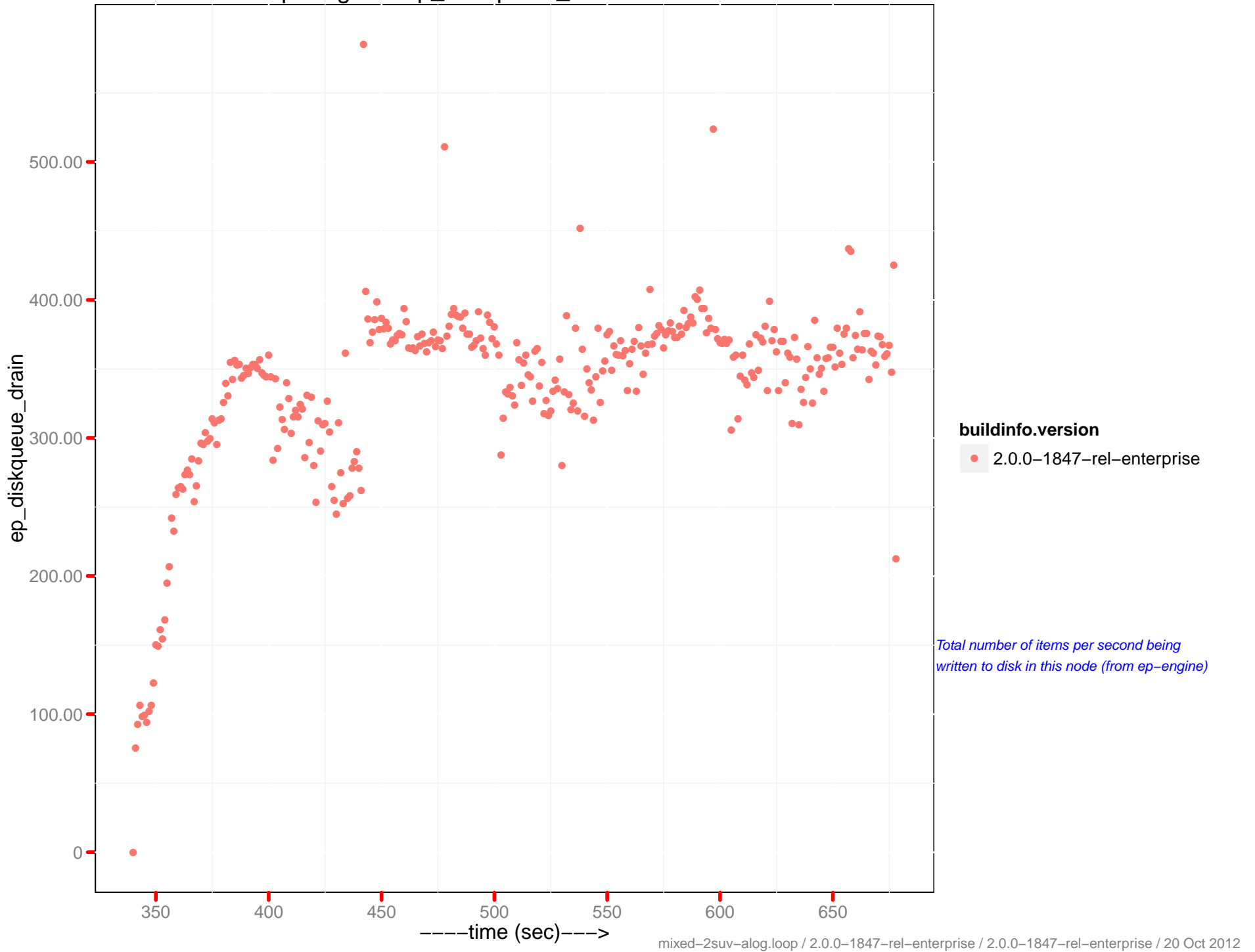
ep-engine : ep\_diskqueue\_drain - 10.2.1.58



ep-engine : ep\_diskqueue\_drain - 10.2.1.61

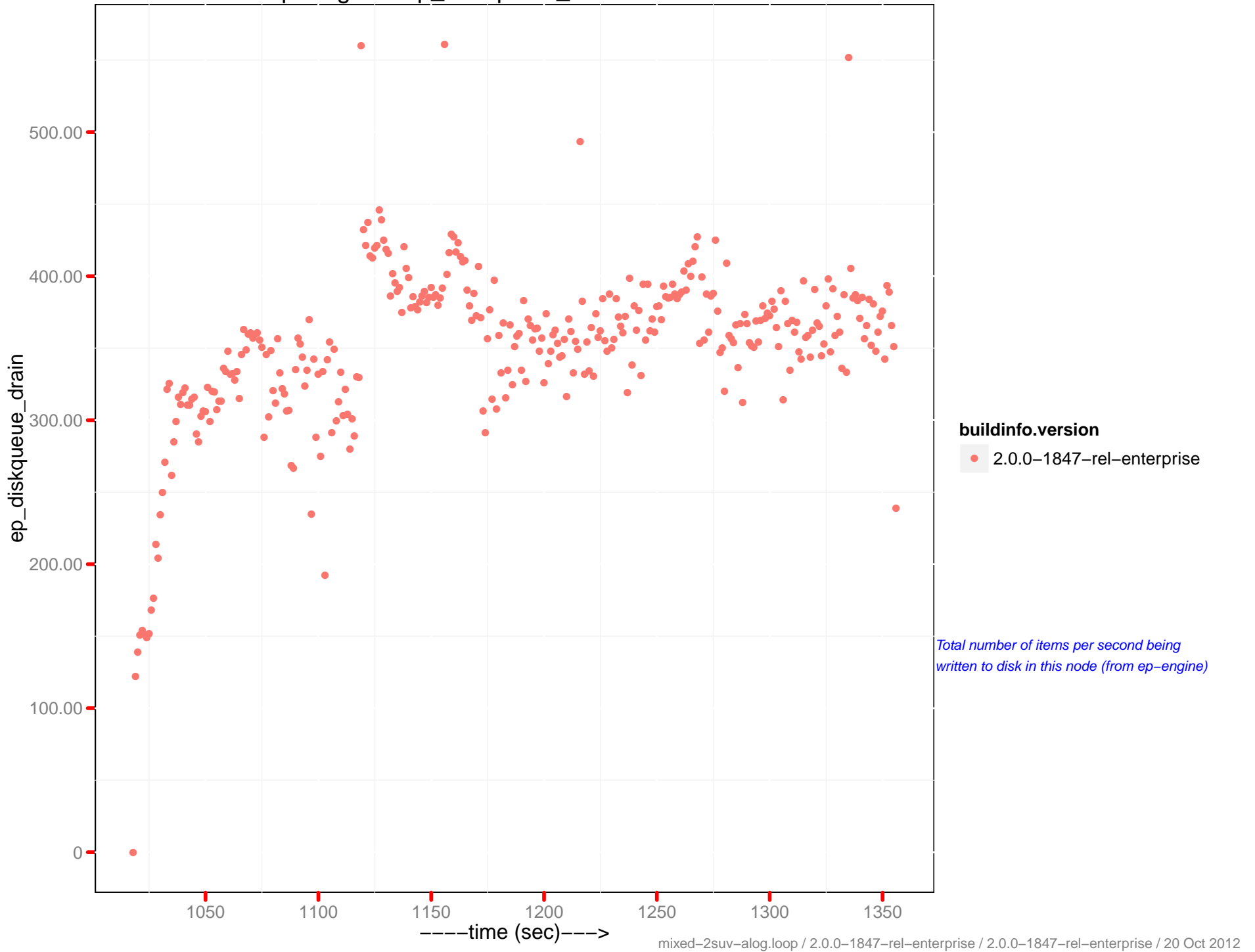


ep-engine : ep\_diskqueue\_drain - 10.2.1.63

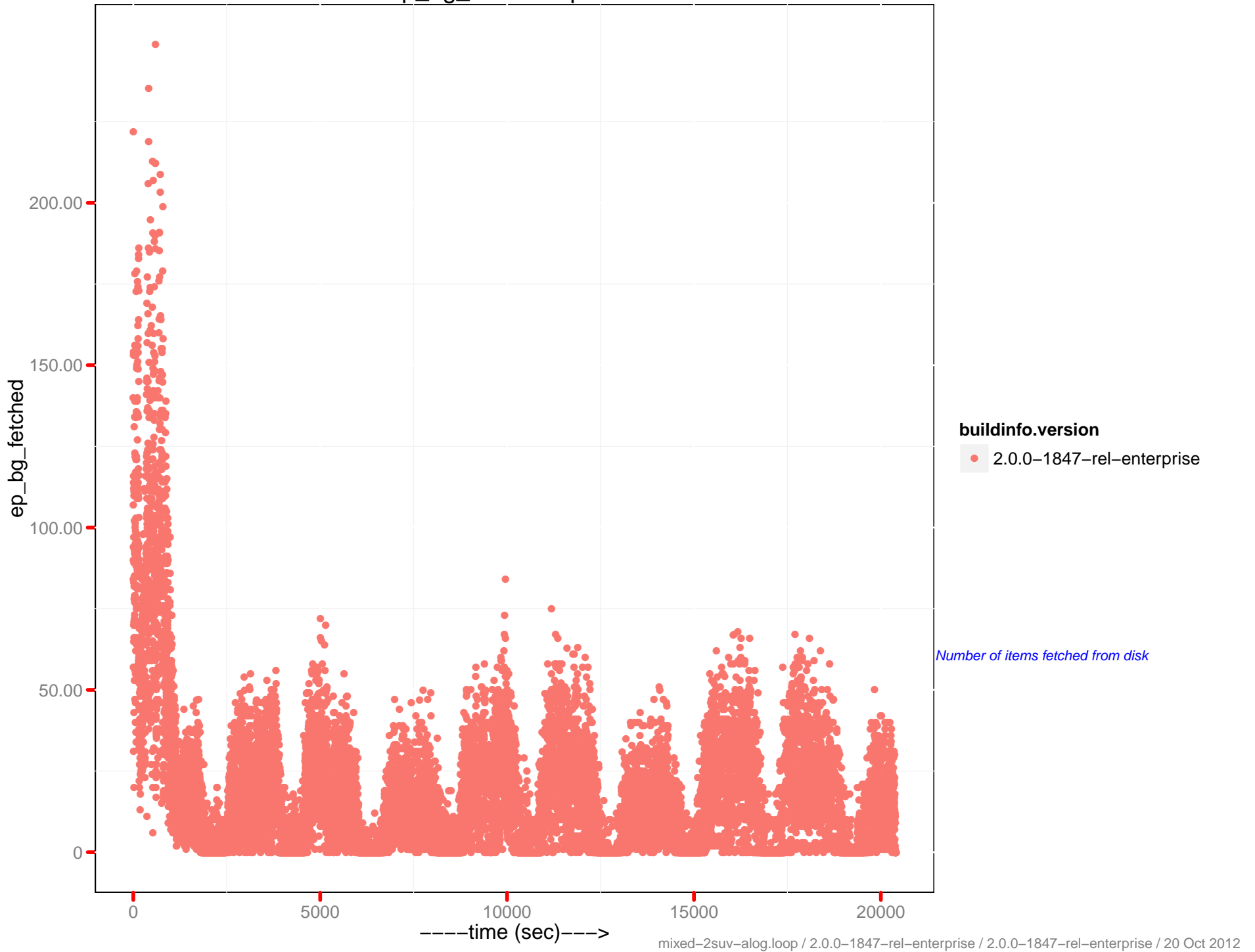




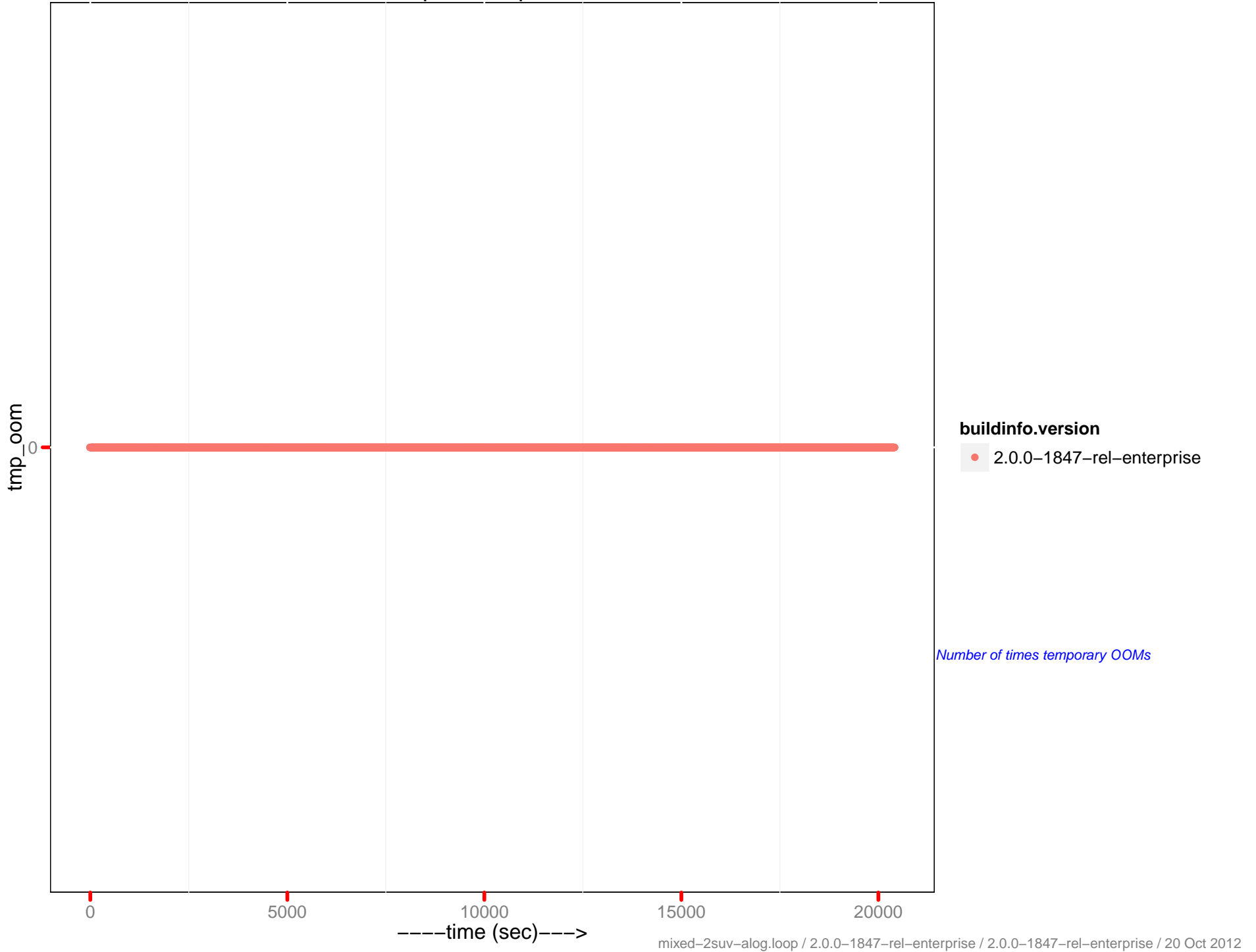
ep-engine : ep\_diskqueue\_drain - 10.2.1.64



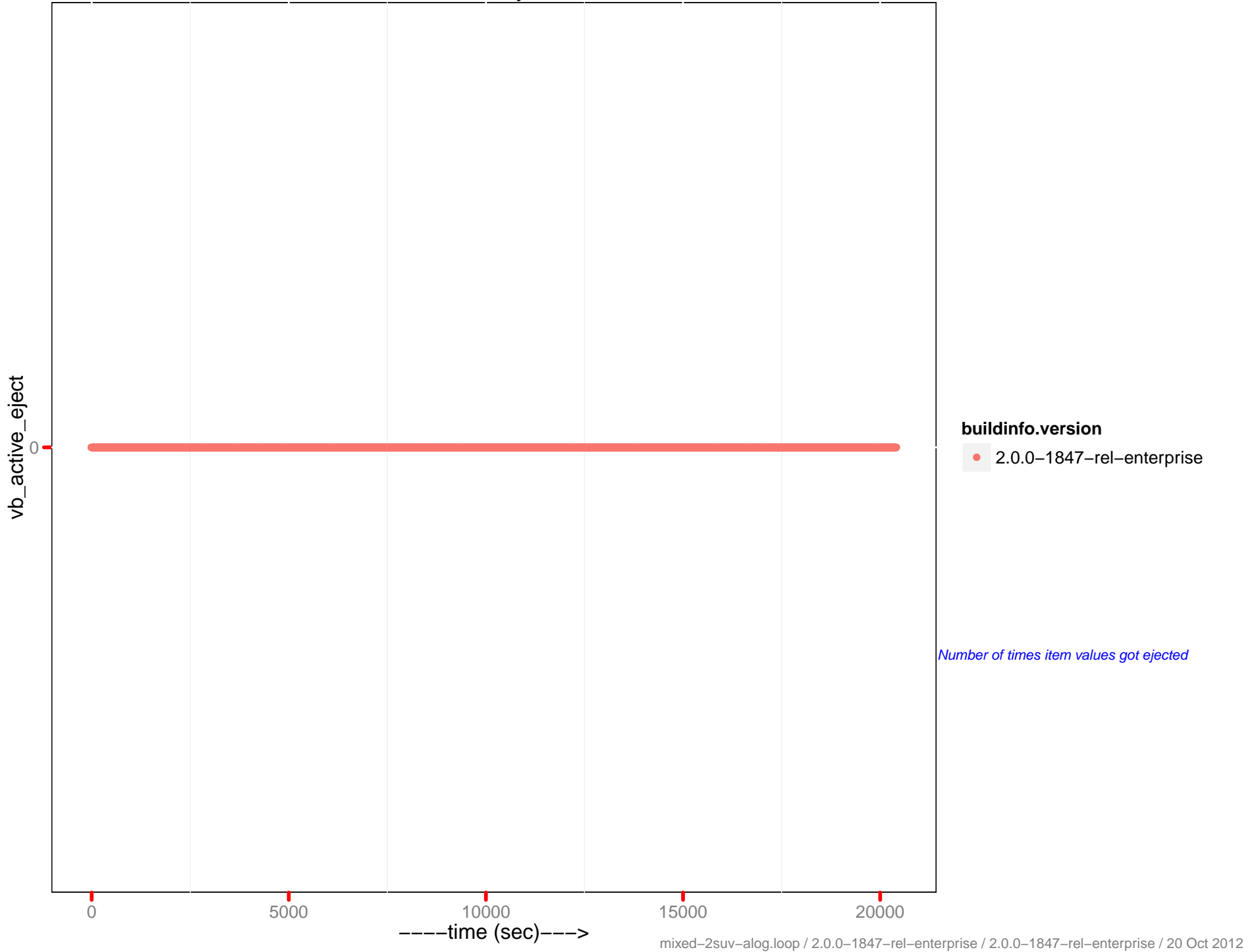
# ep\_bg\_fetched ops/sec



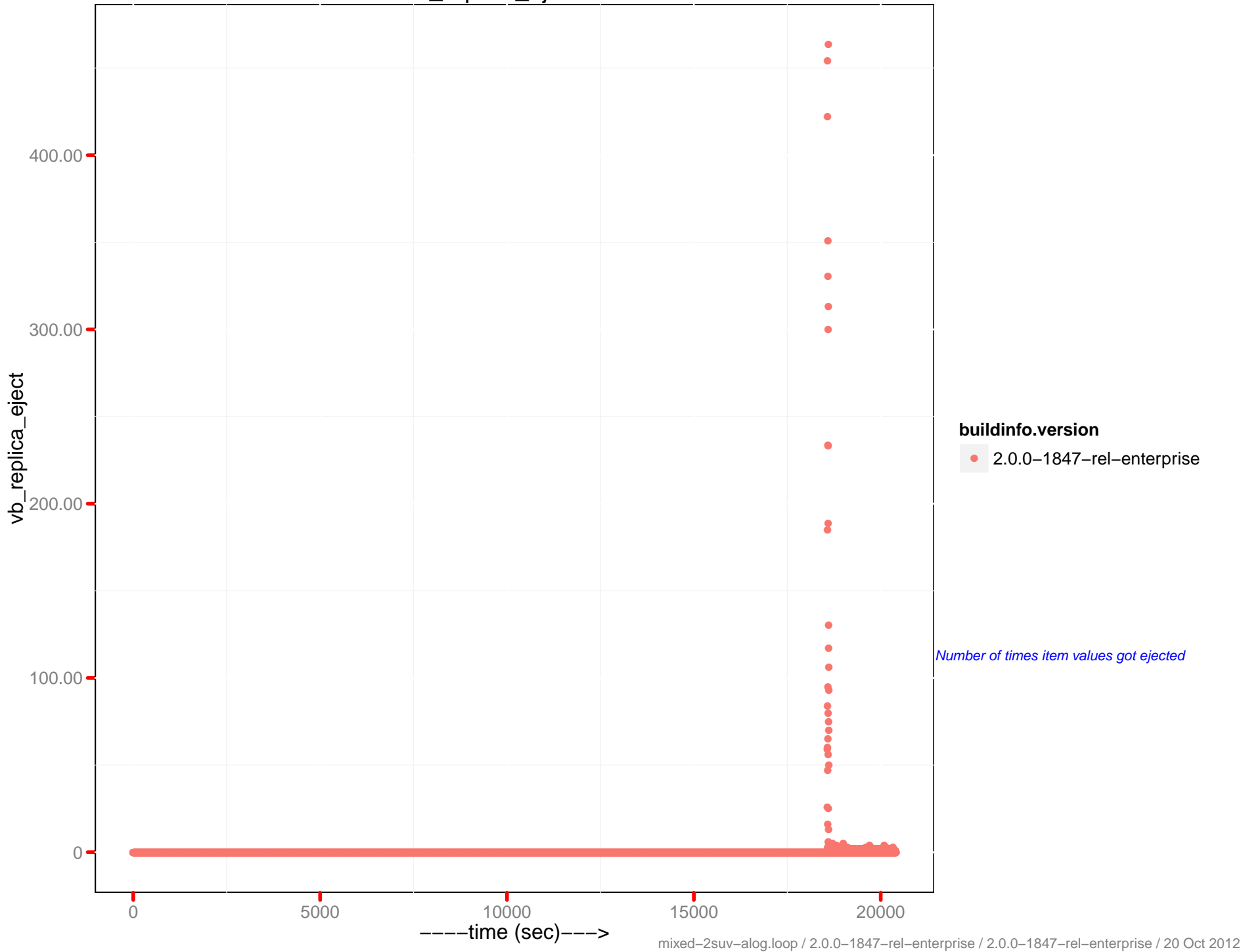
# tmp\_oom ops/sec



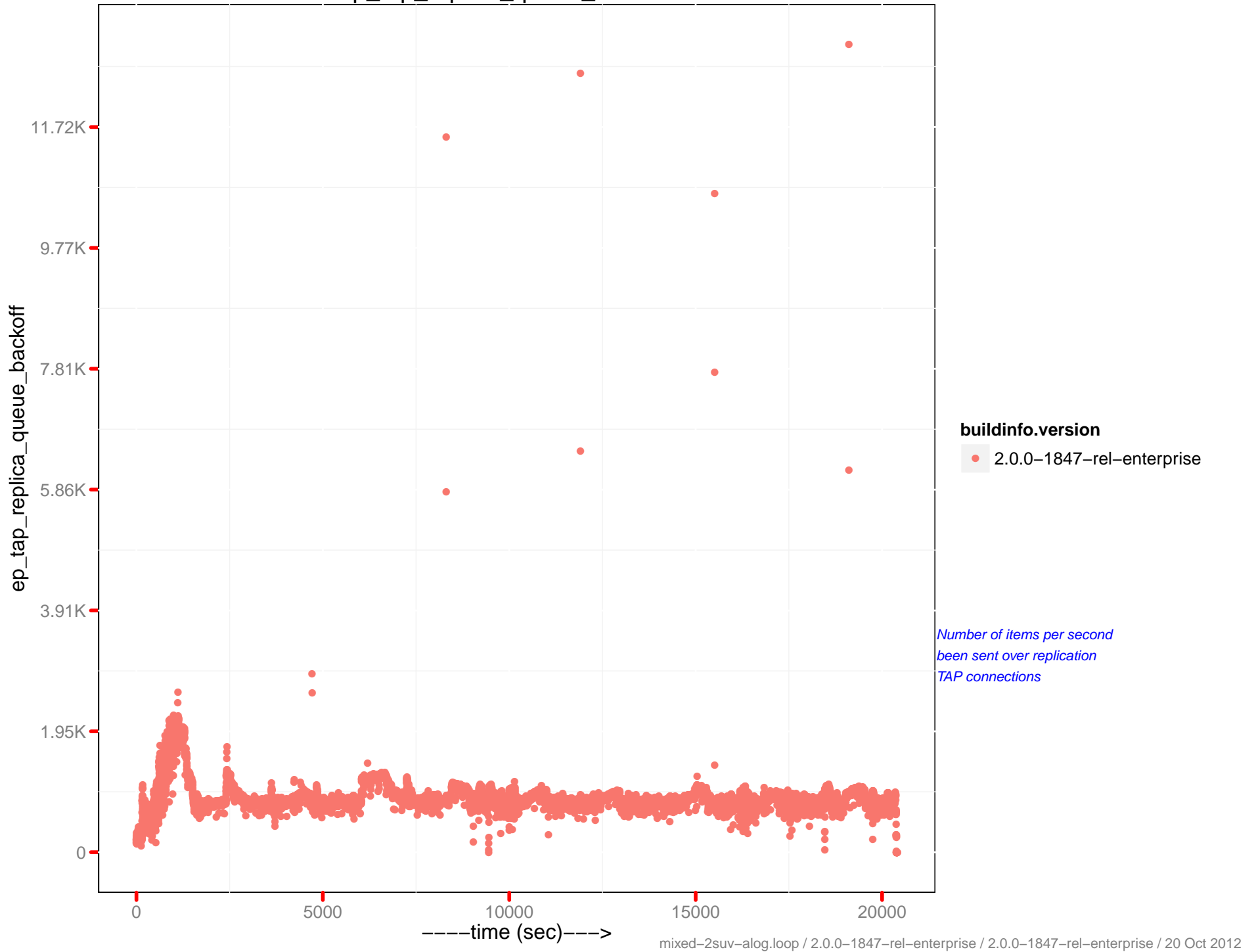
# vb\_active\_eject/sec



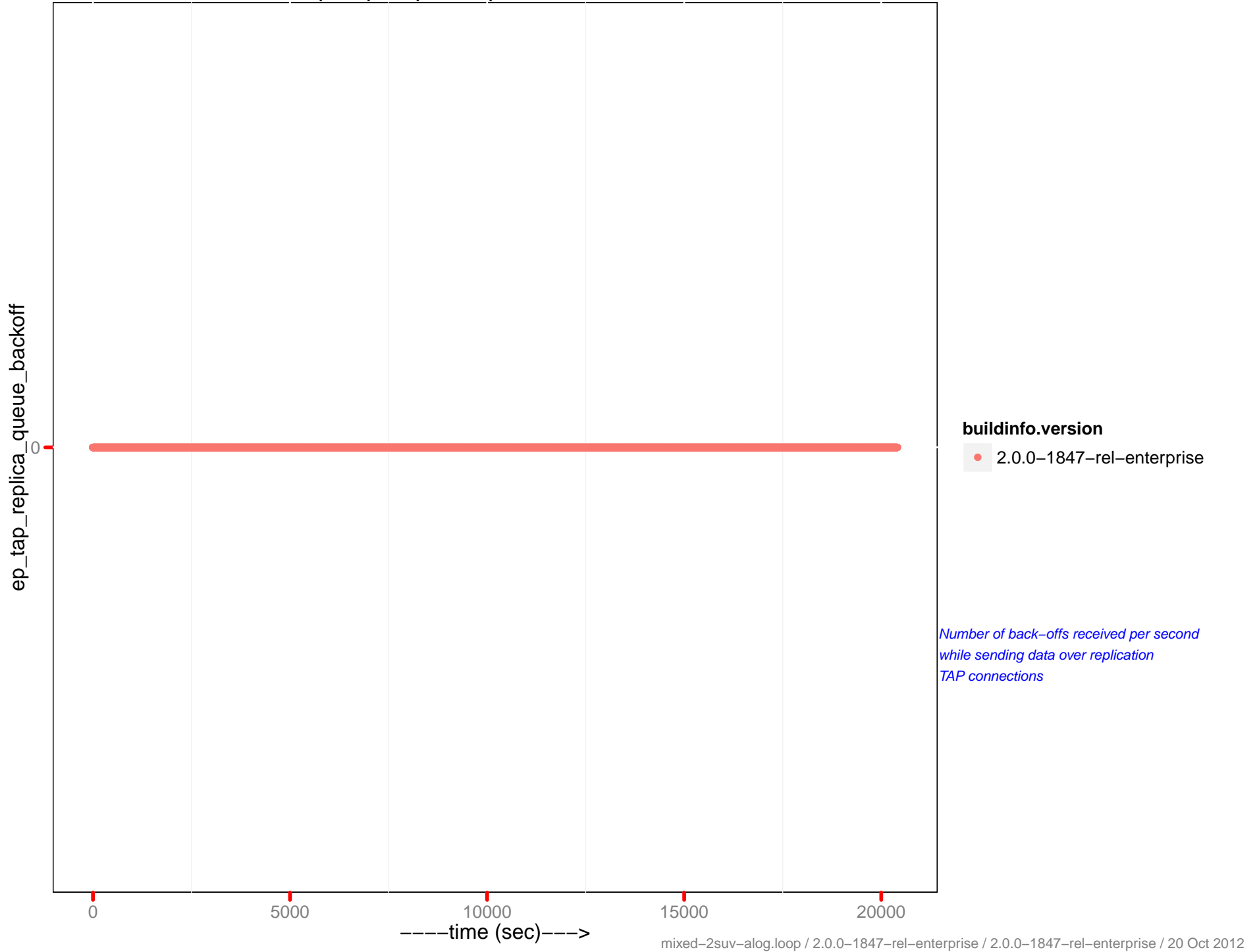
vb\_replica\_eject/sec



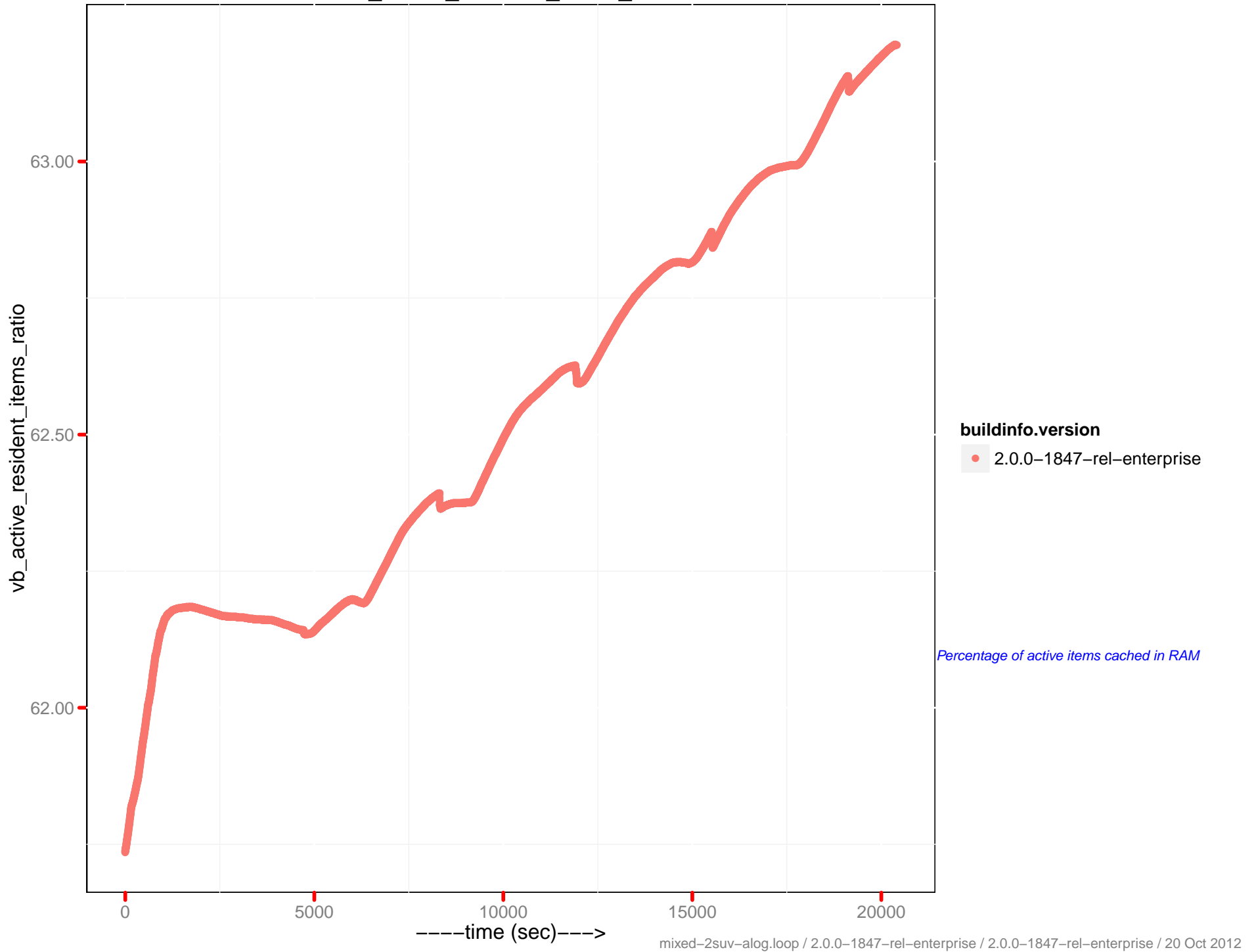
# ep\_tap\_replica\_queue\_drain/sec



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

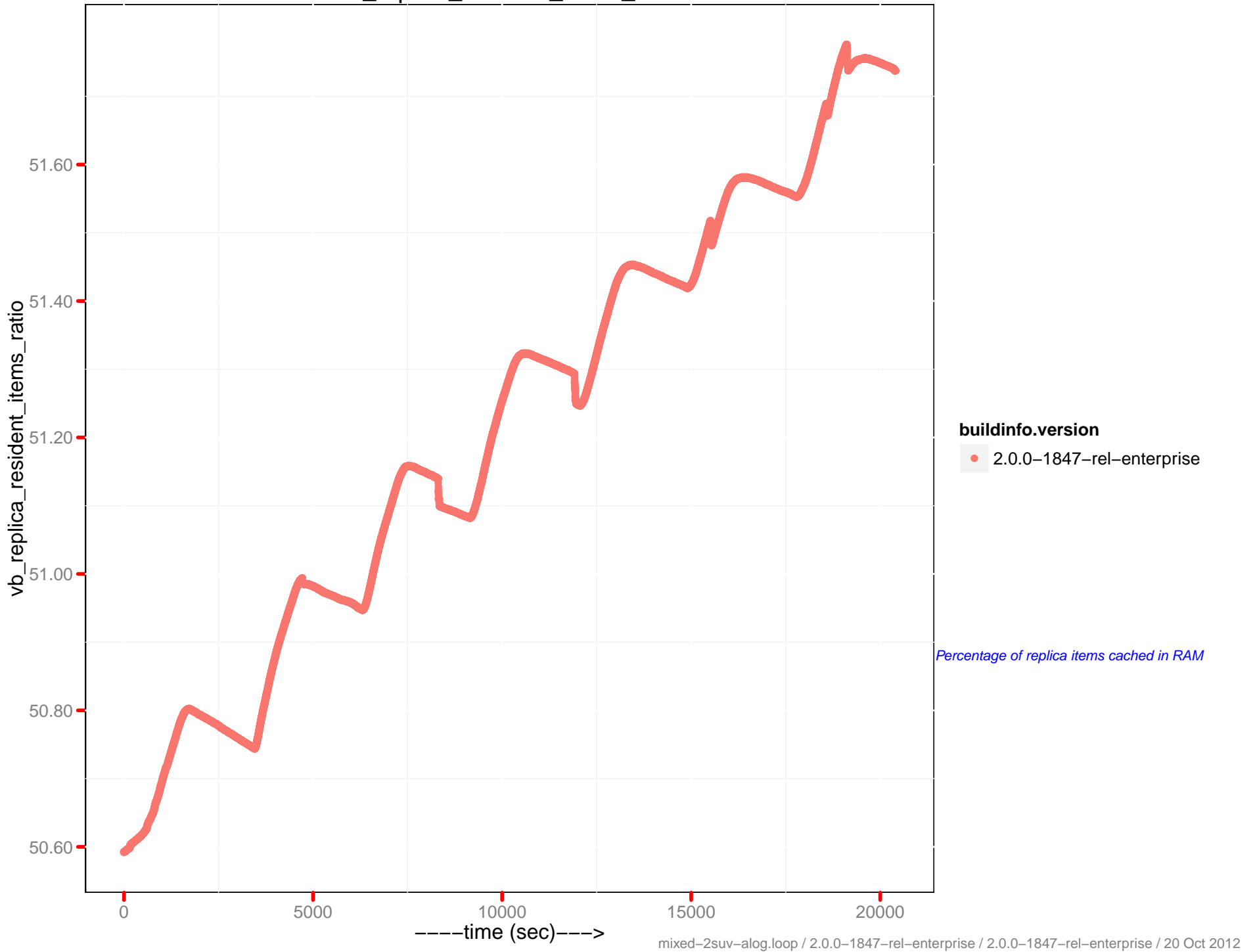


vb\_active\_resident\_items\_ratio

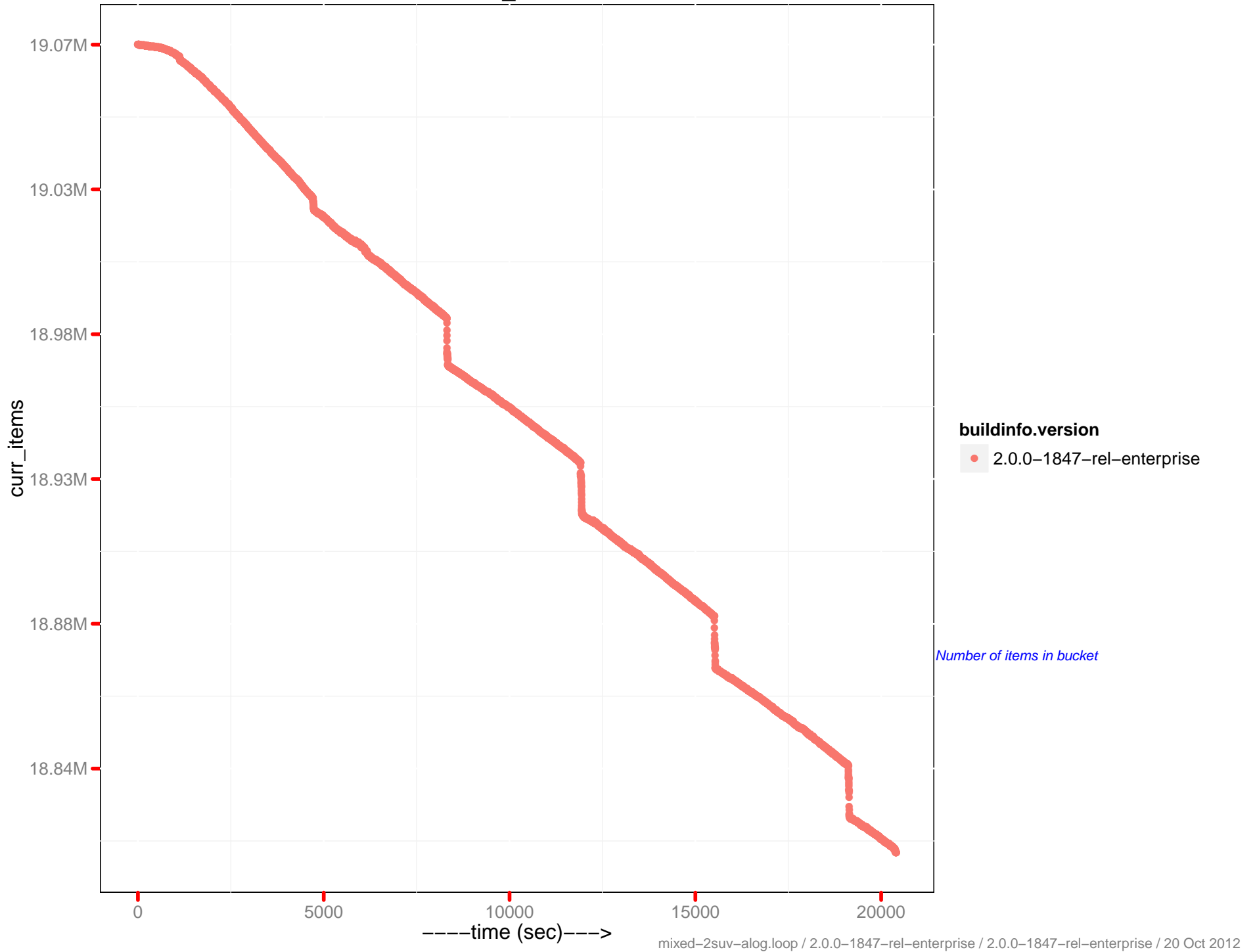




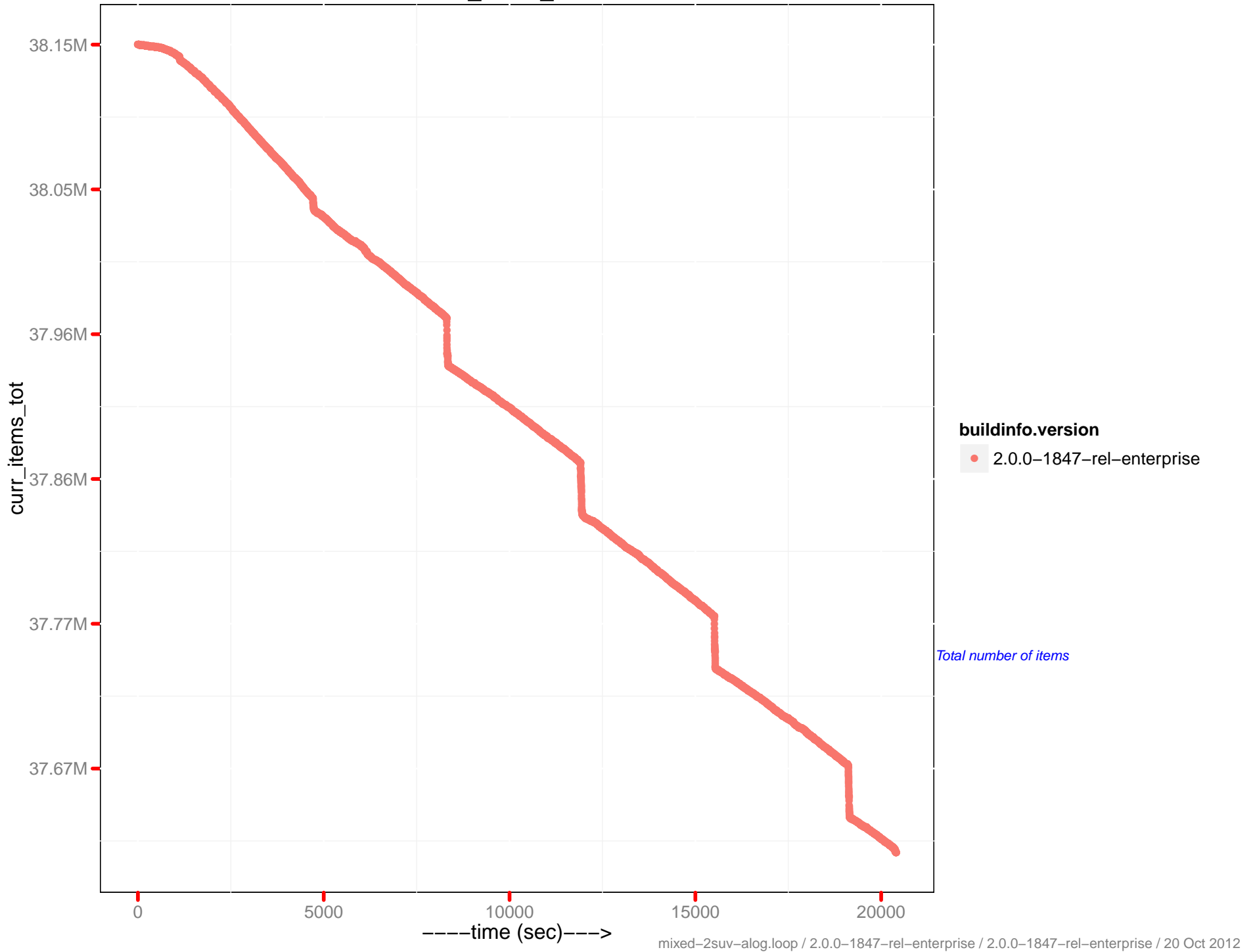
# vb\_replica\_resident\_items\_ratio



# curr\_items



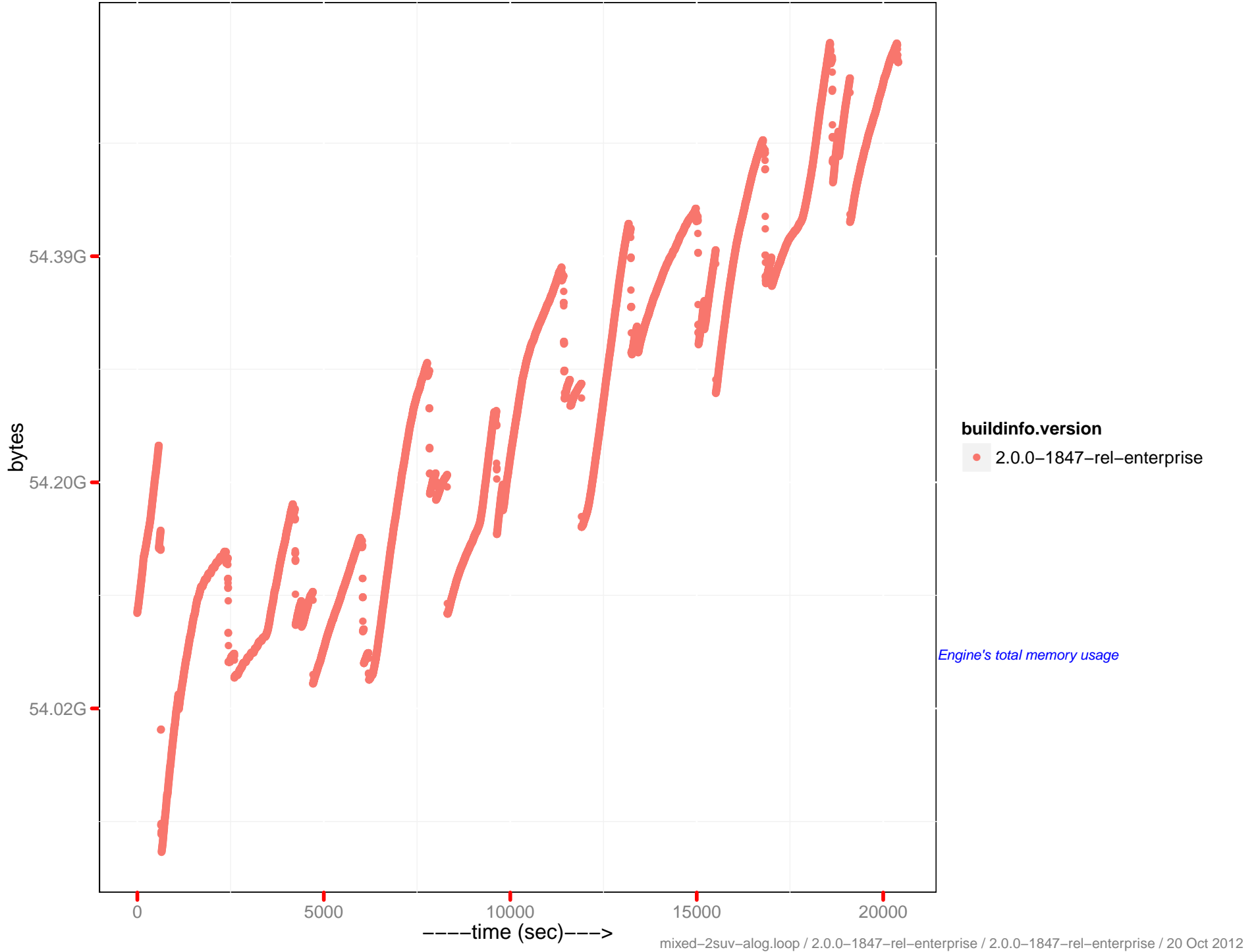
# cur\_items\_total



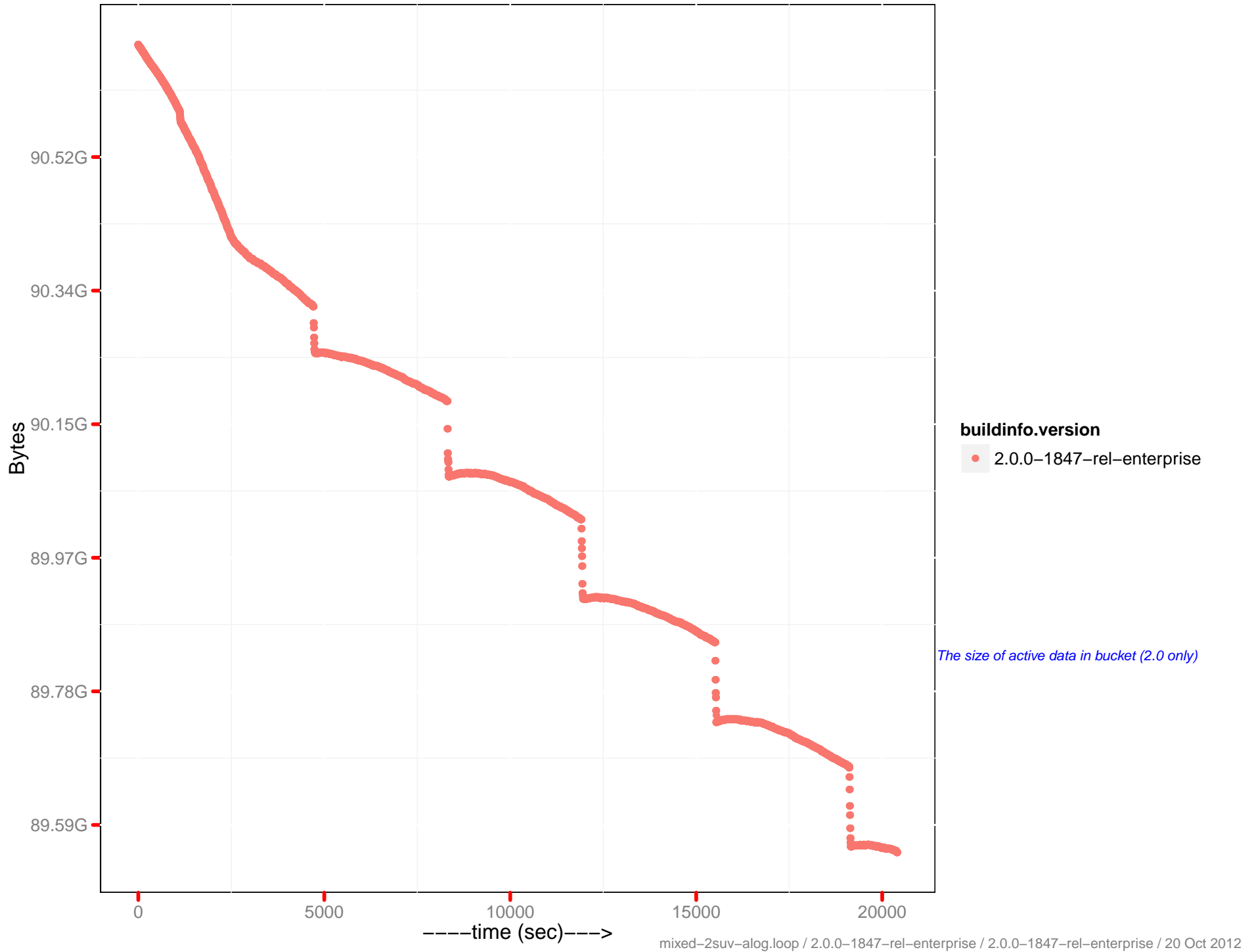
**buildinfo.version**  
● 2.0.0-1847-rel-enterprise

*Total number of items*

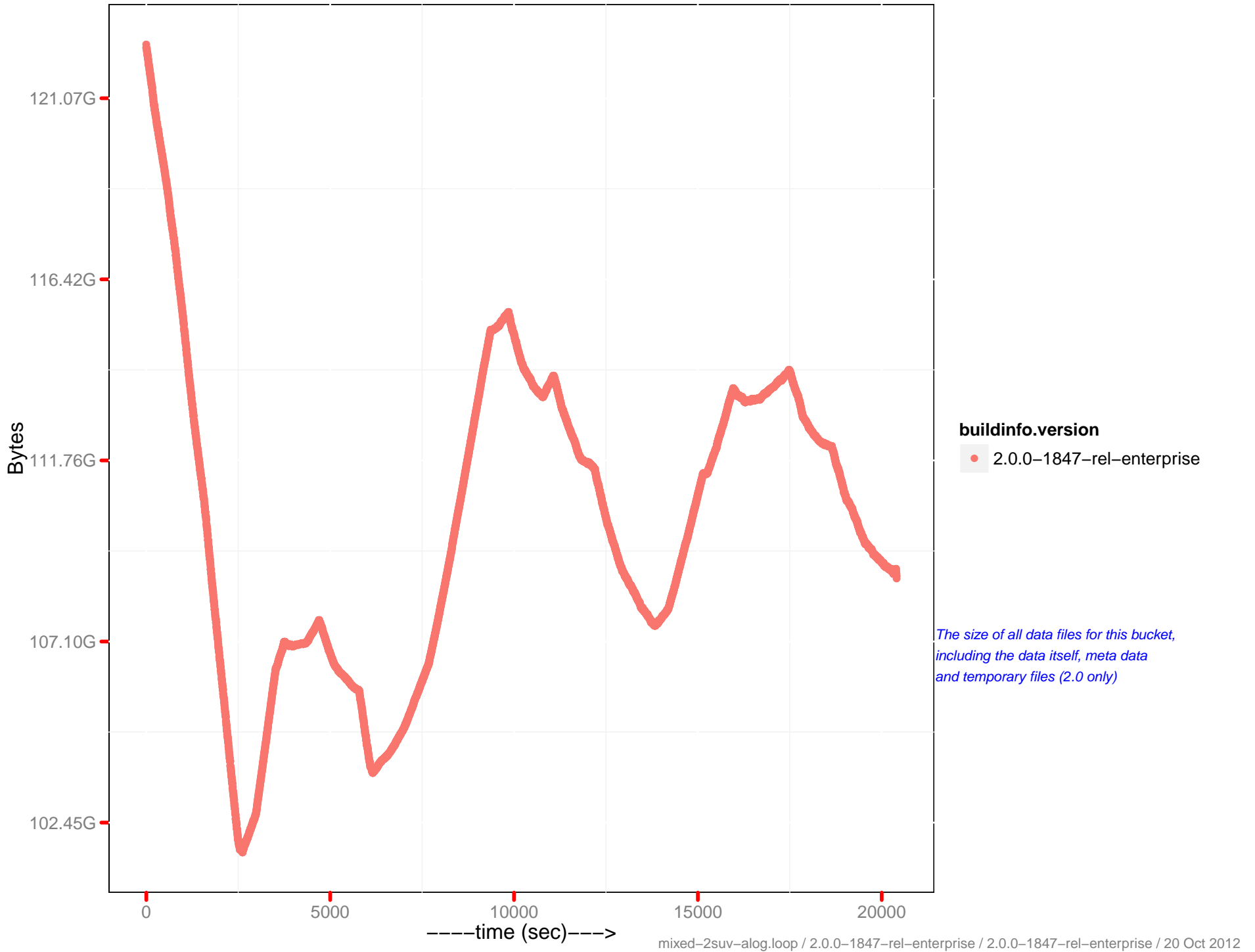
# mem\_used



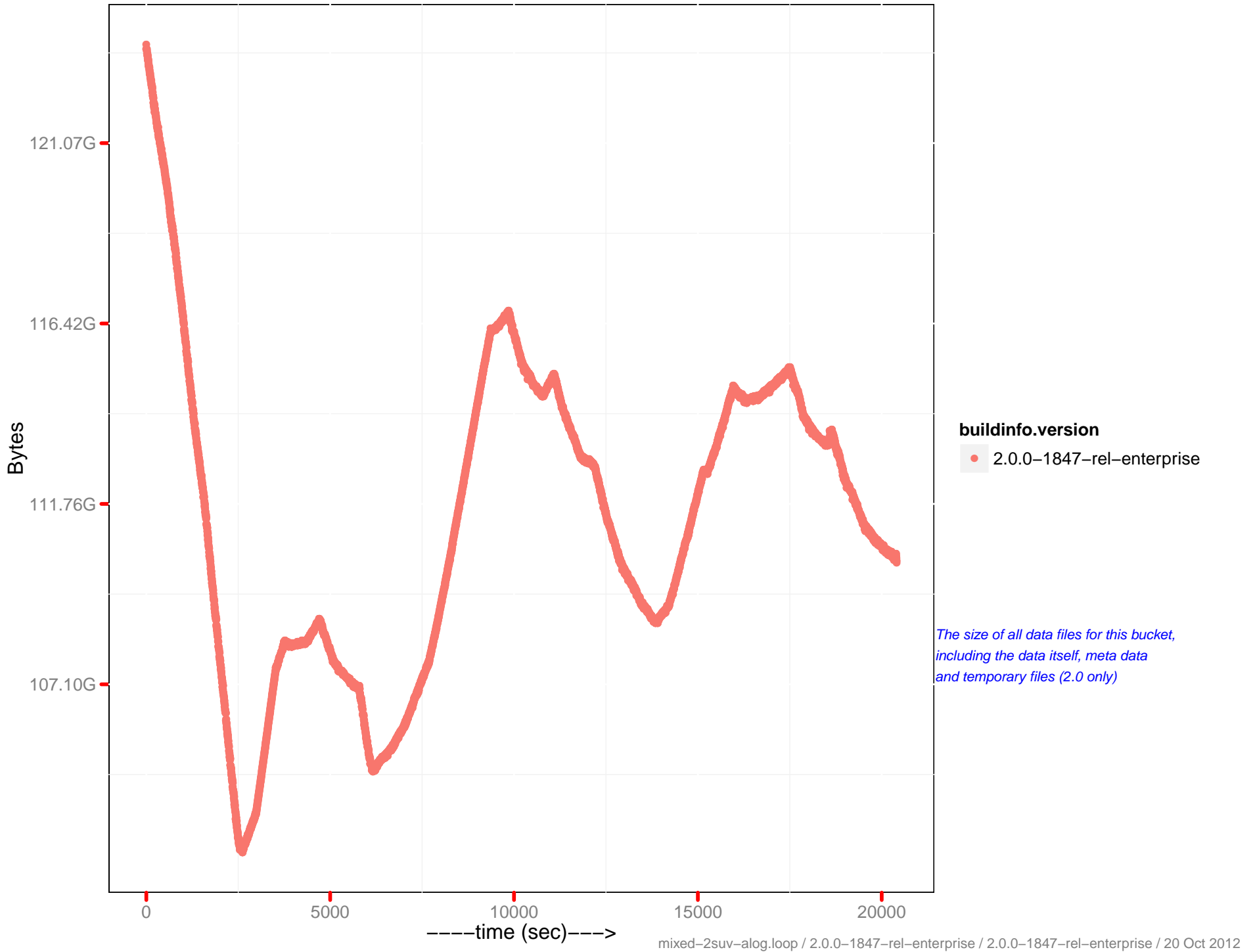
# Docs data size



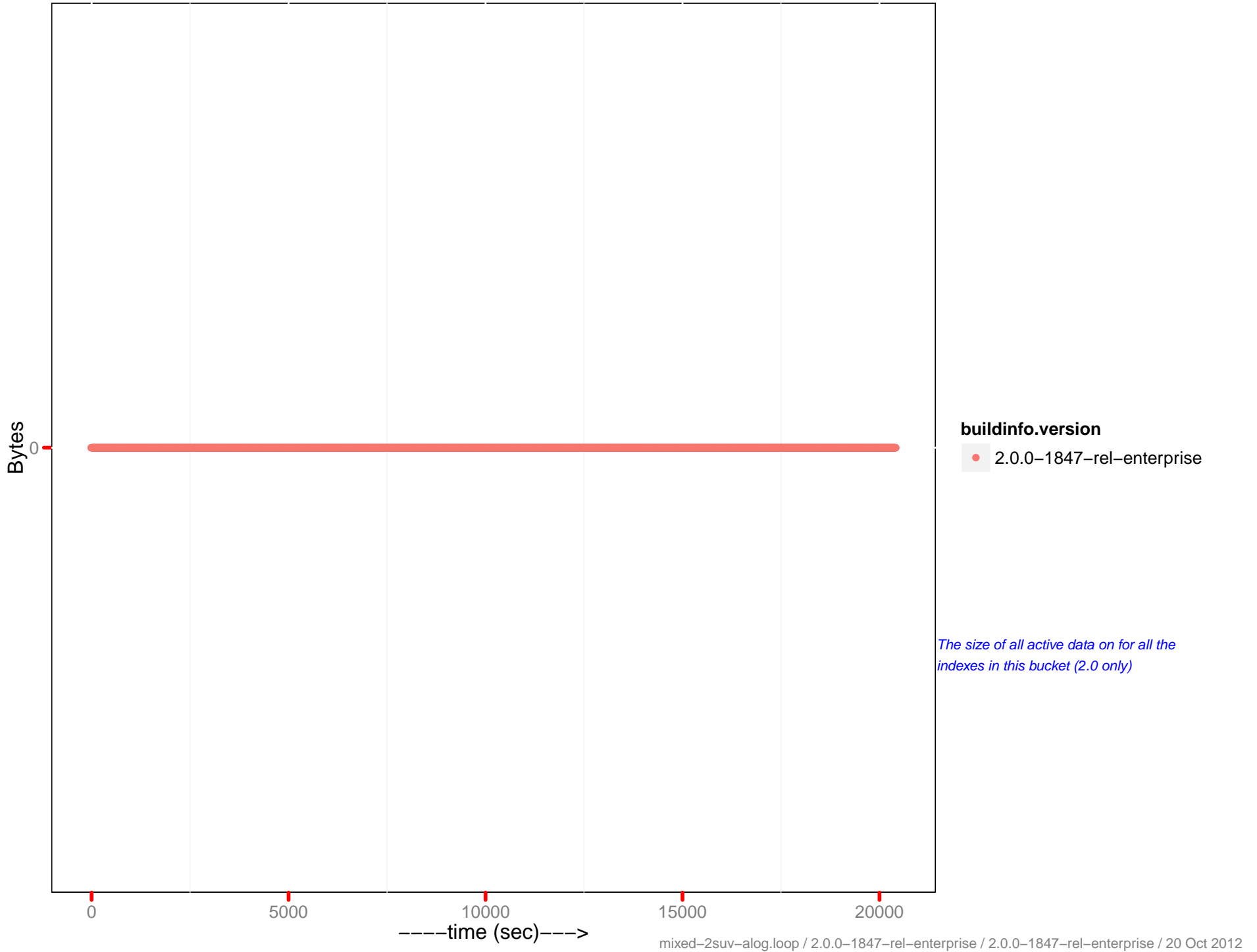
# Docs disk size



# Docs actual disk size

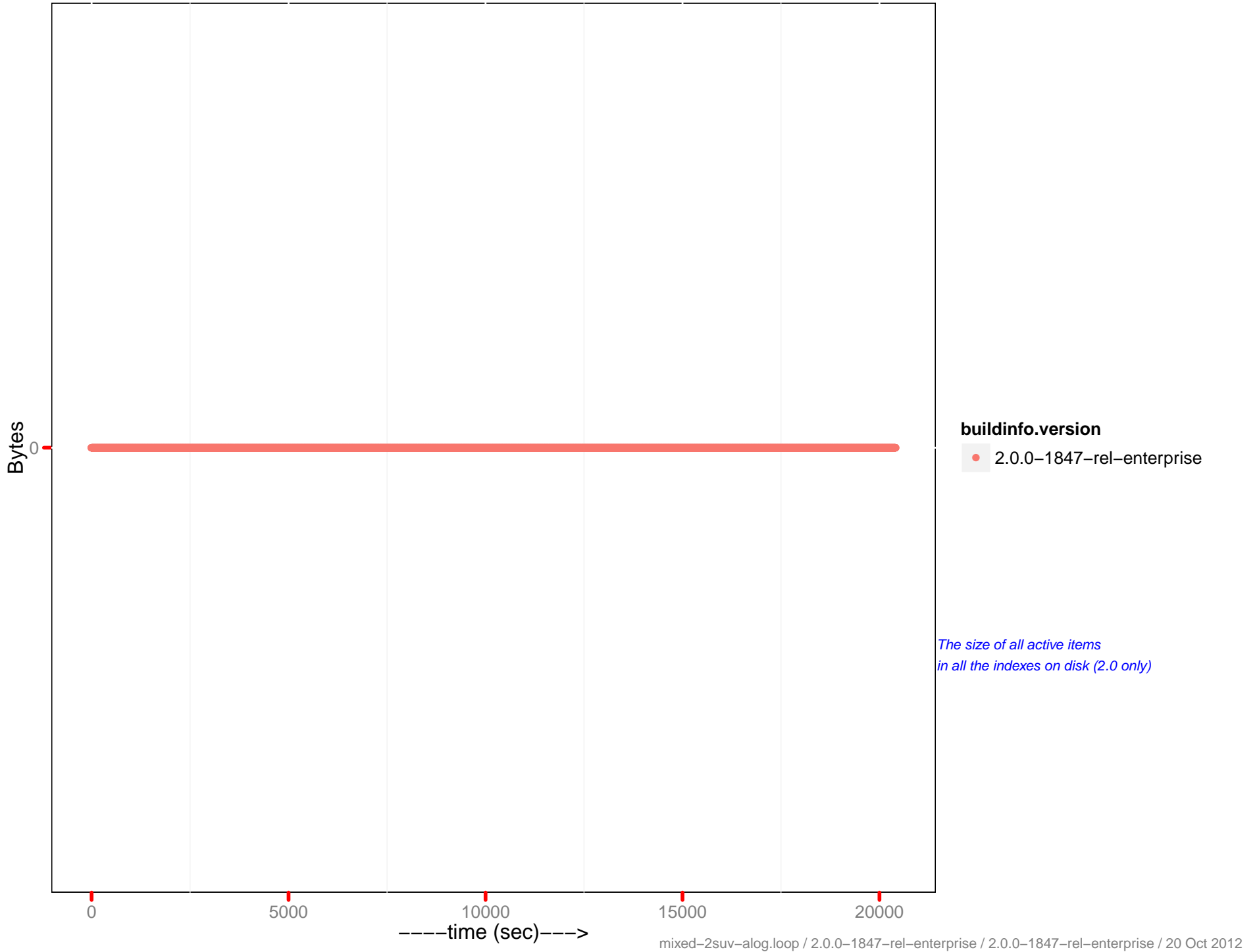


# Views data size

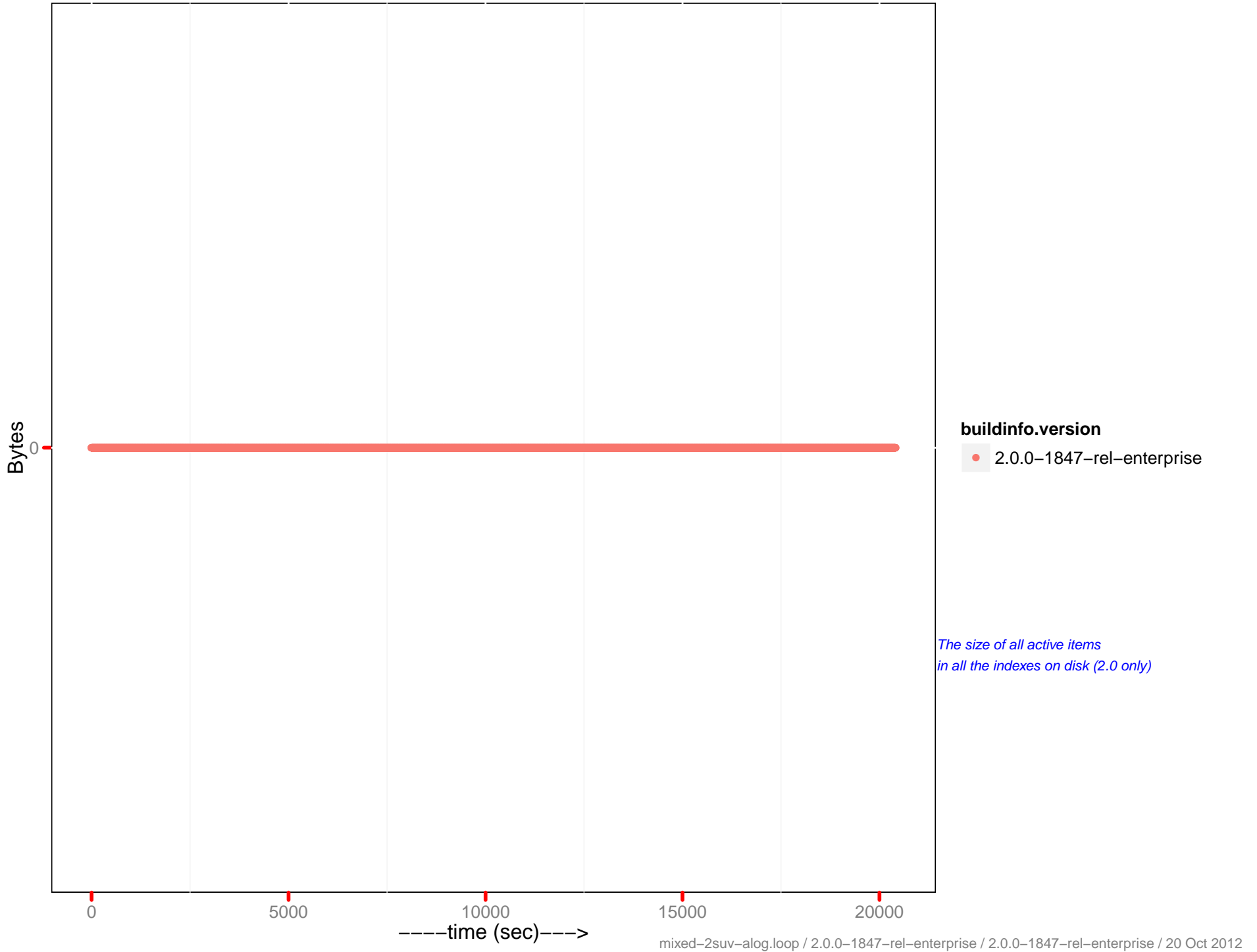




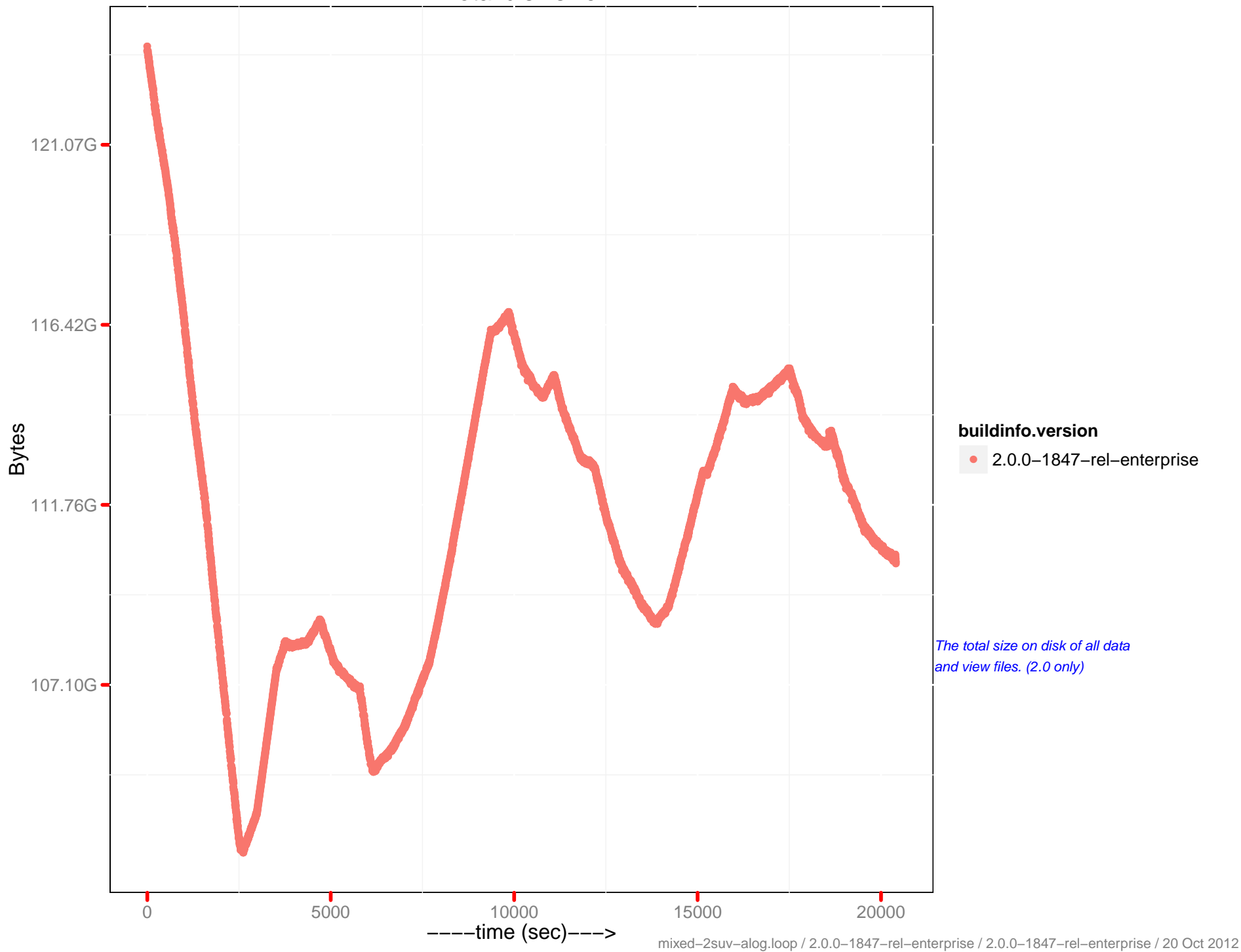
# Views disk size



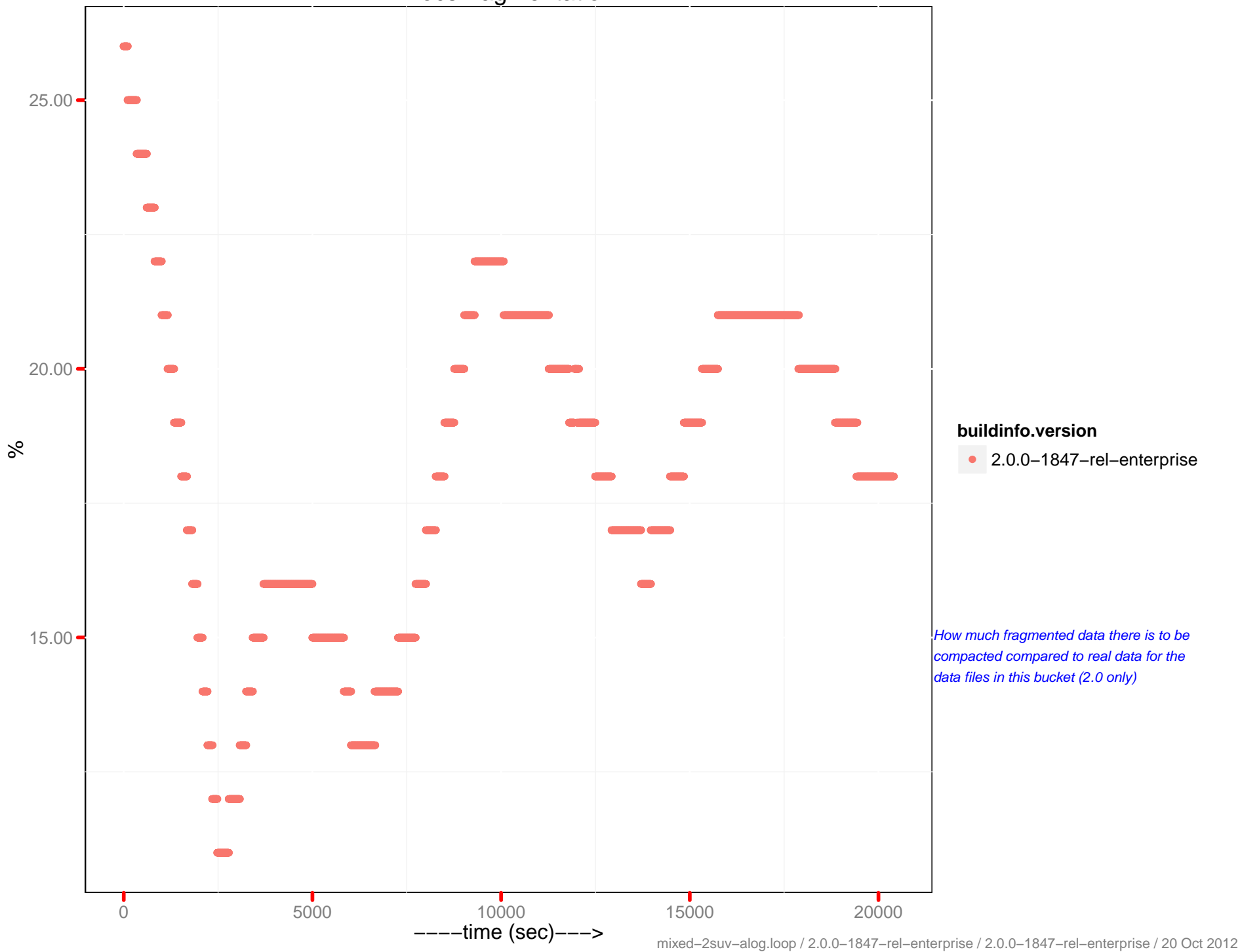
# Views actual disk size



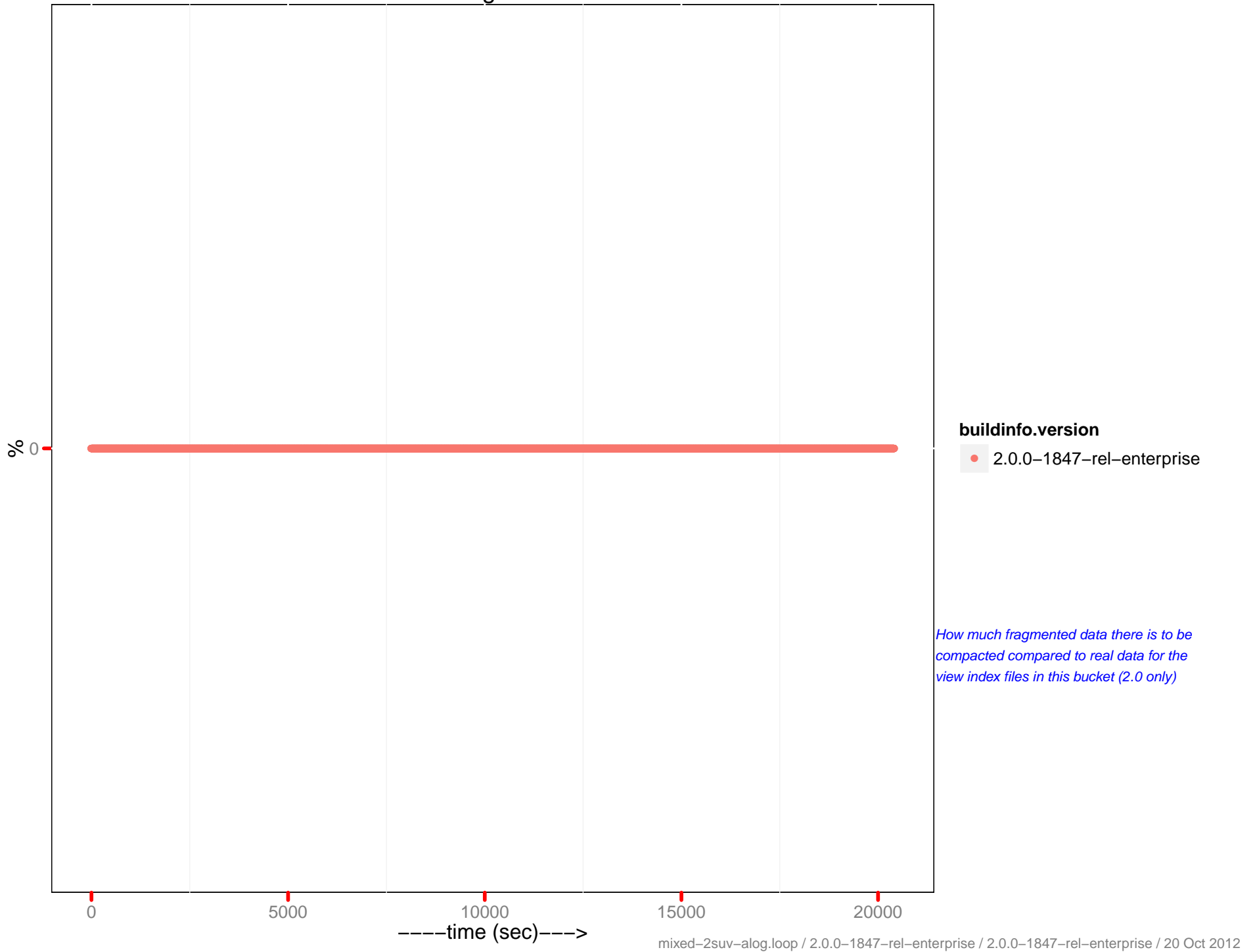
# Total disk size



# Docs fragmentation



# Views fragmentation

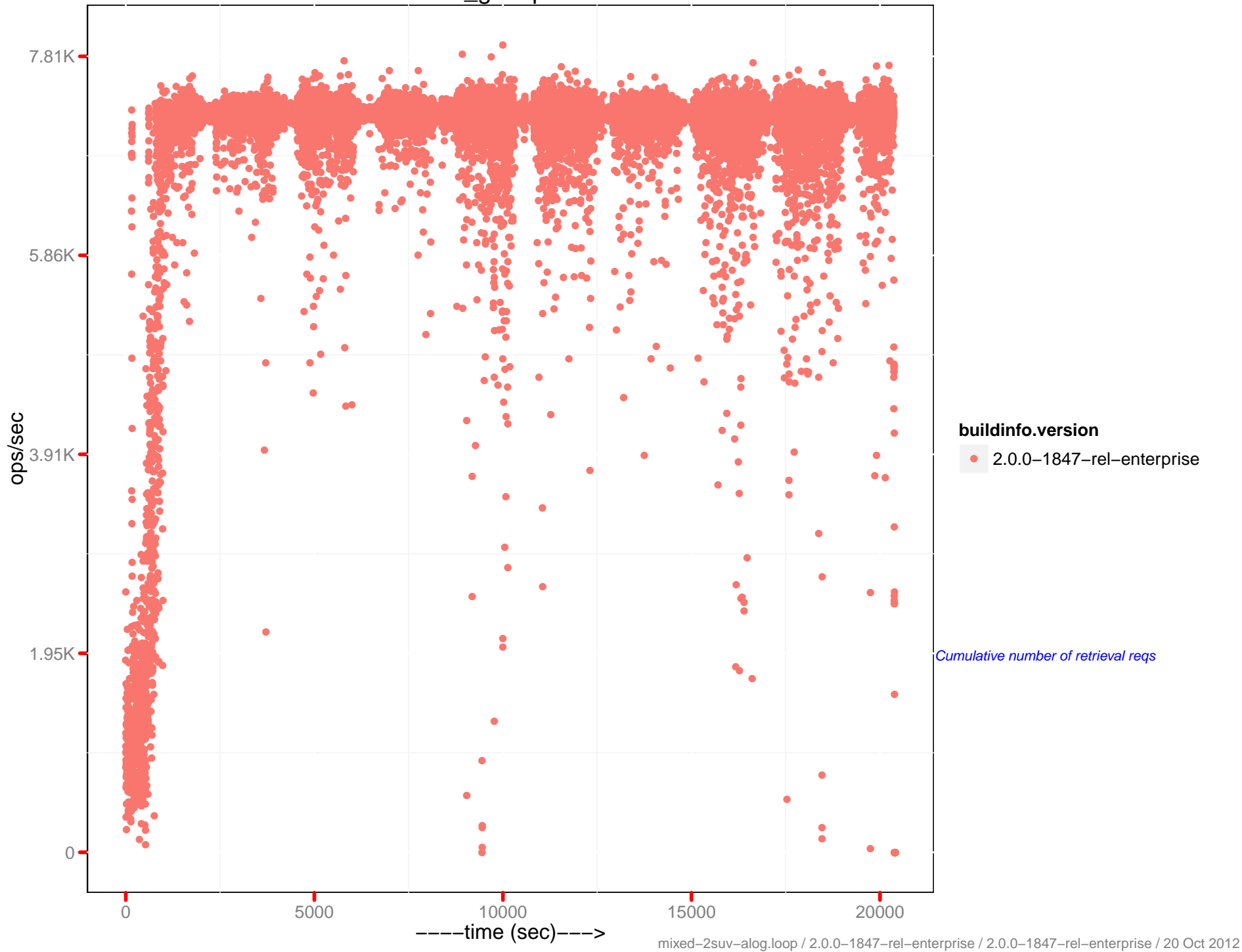


**buildinfo.version**

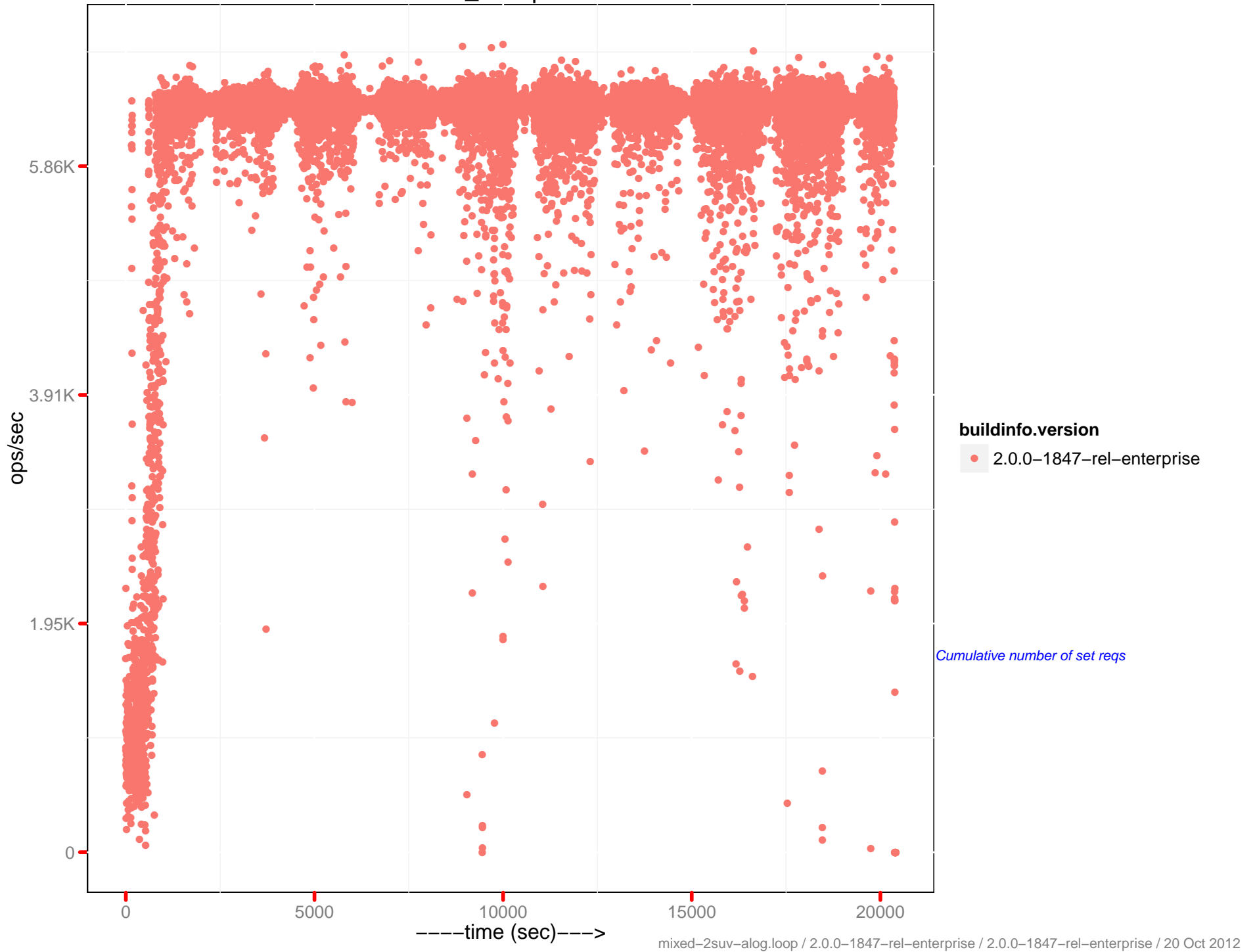
• 2.0.0-1847-rel-enterprise

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

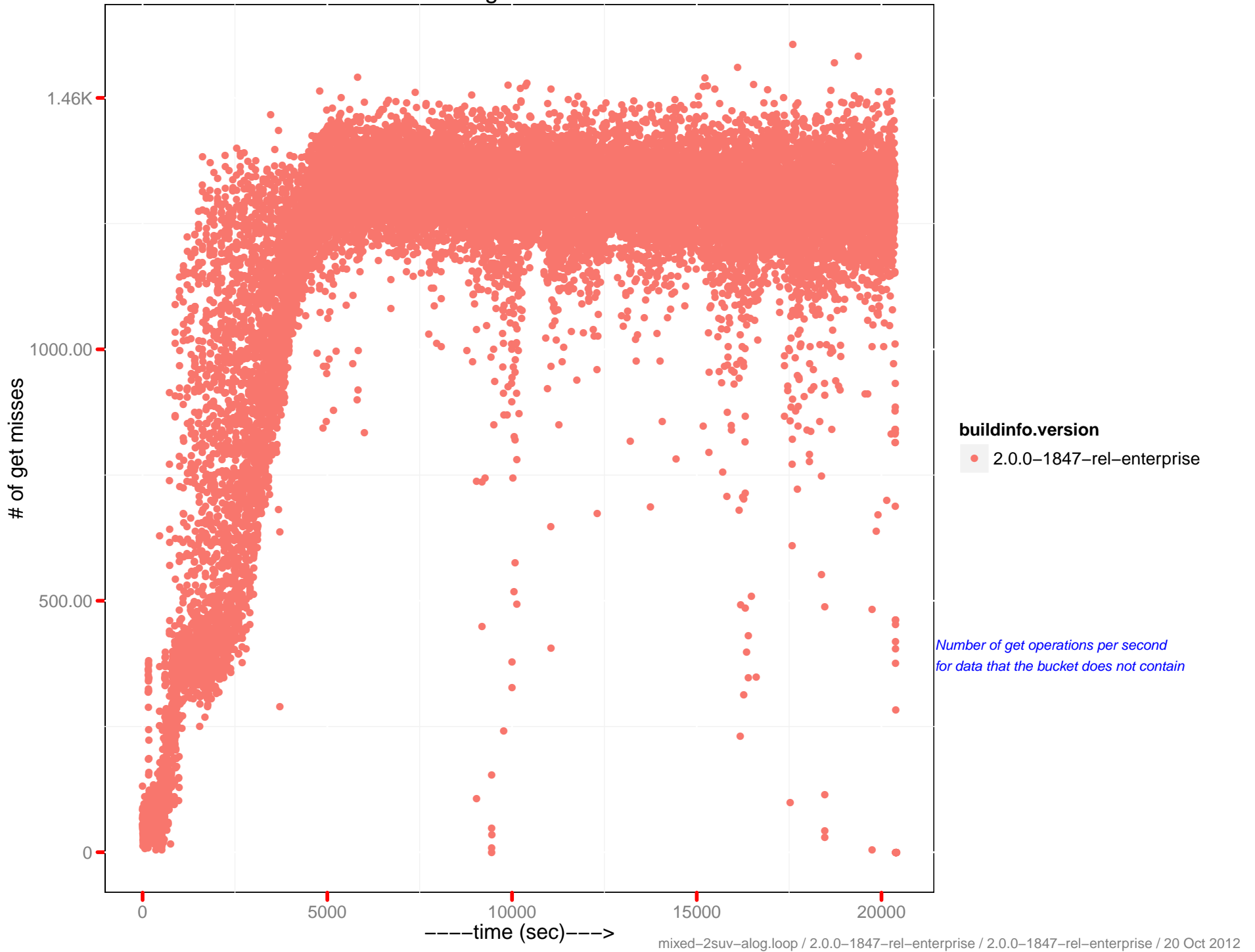
# cmd\_get ops/sec



# cmd\_set ops/sec

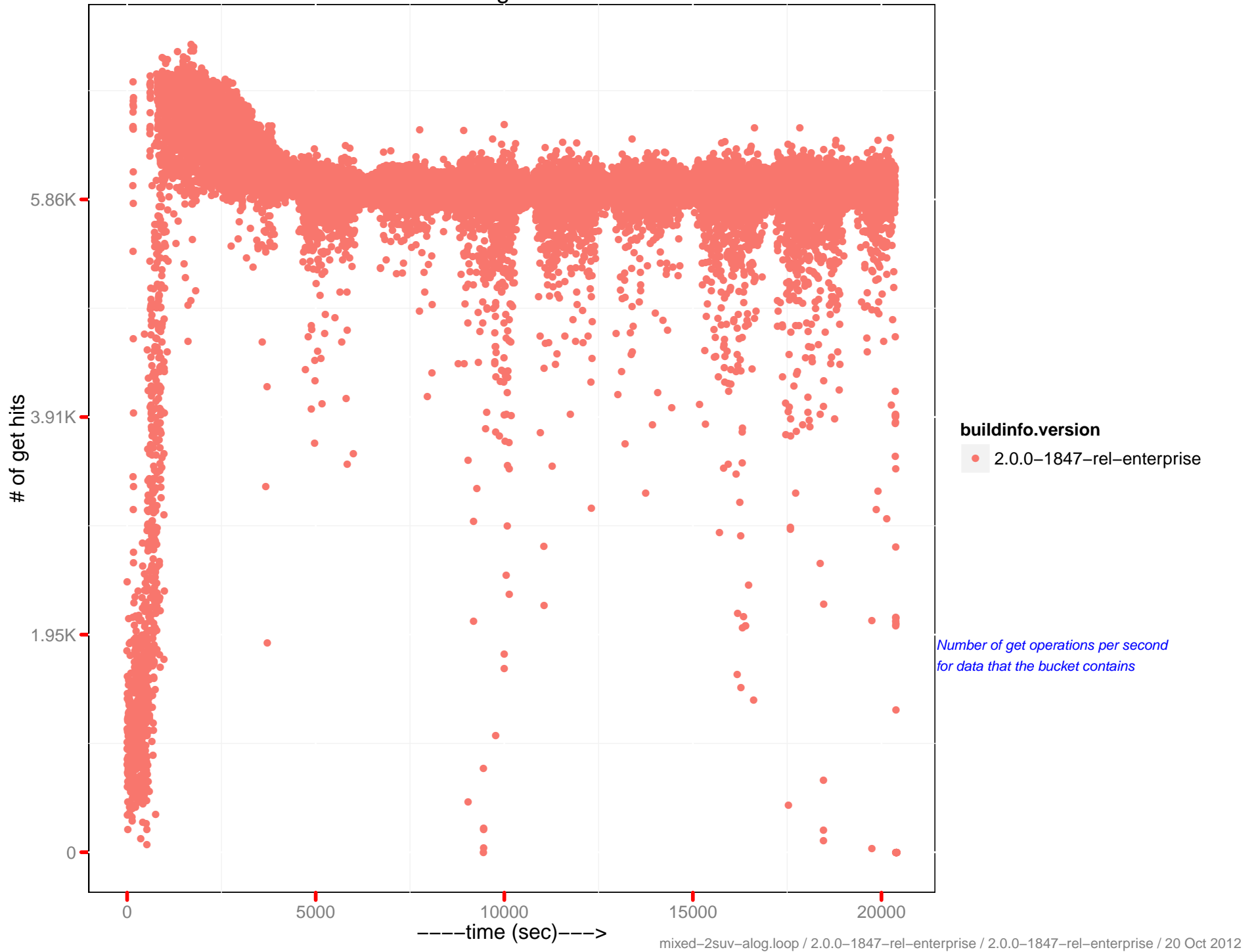


# # of get misses

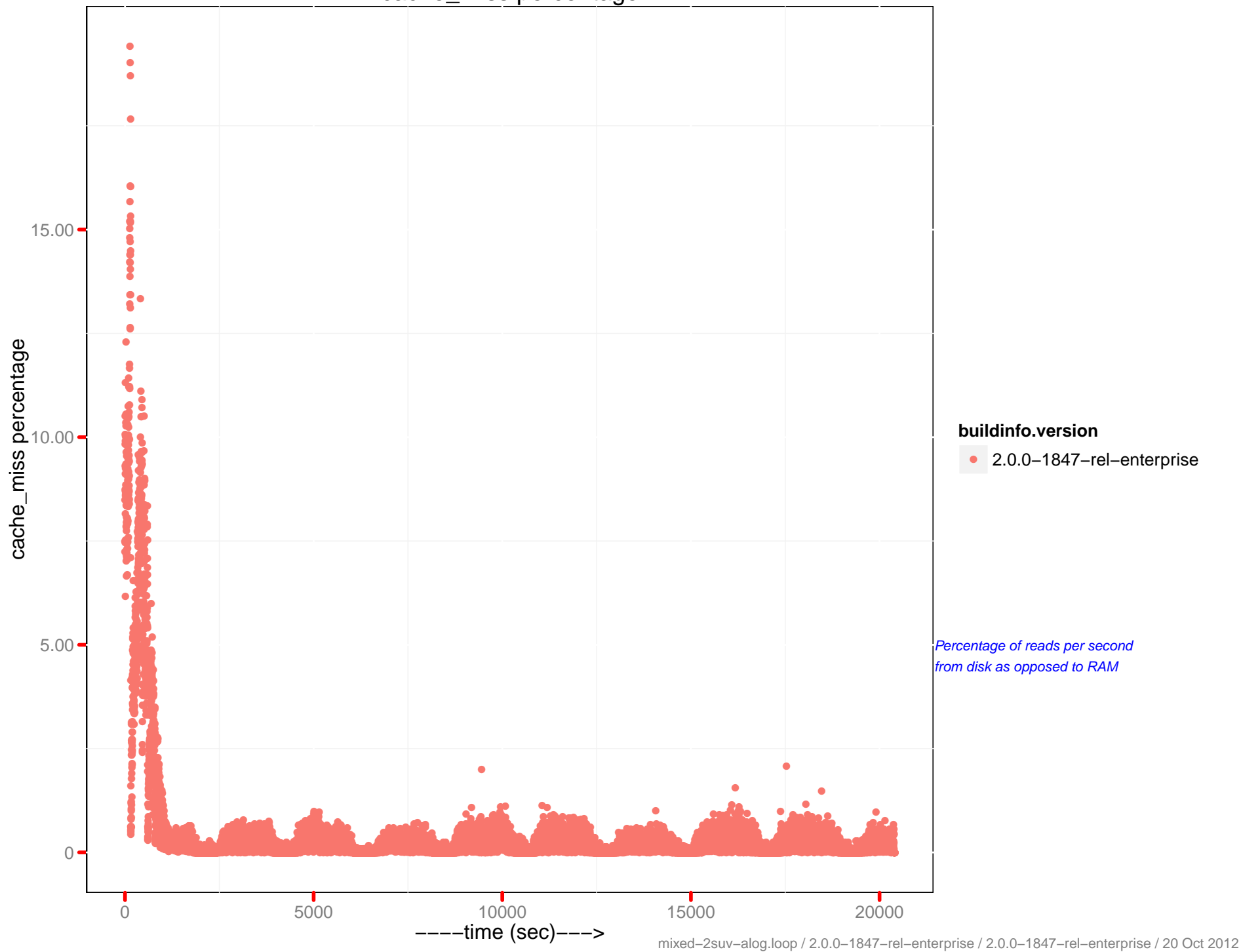




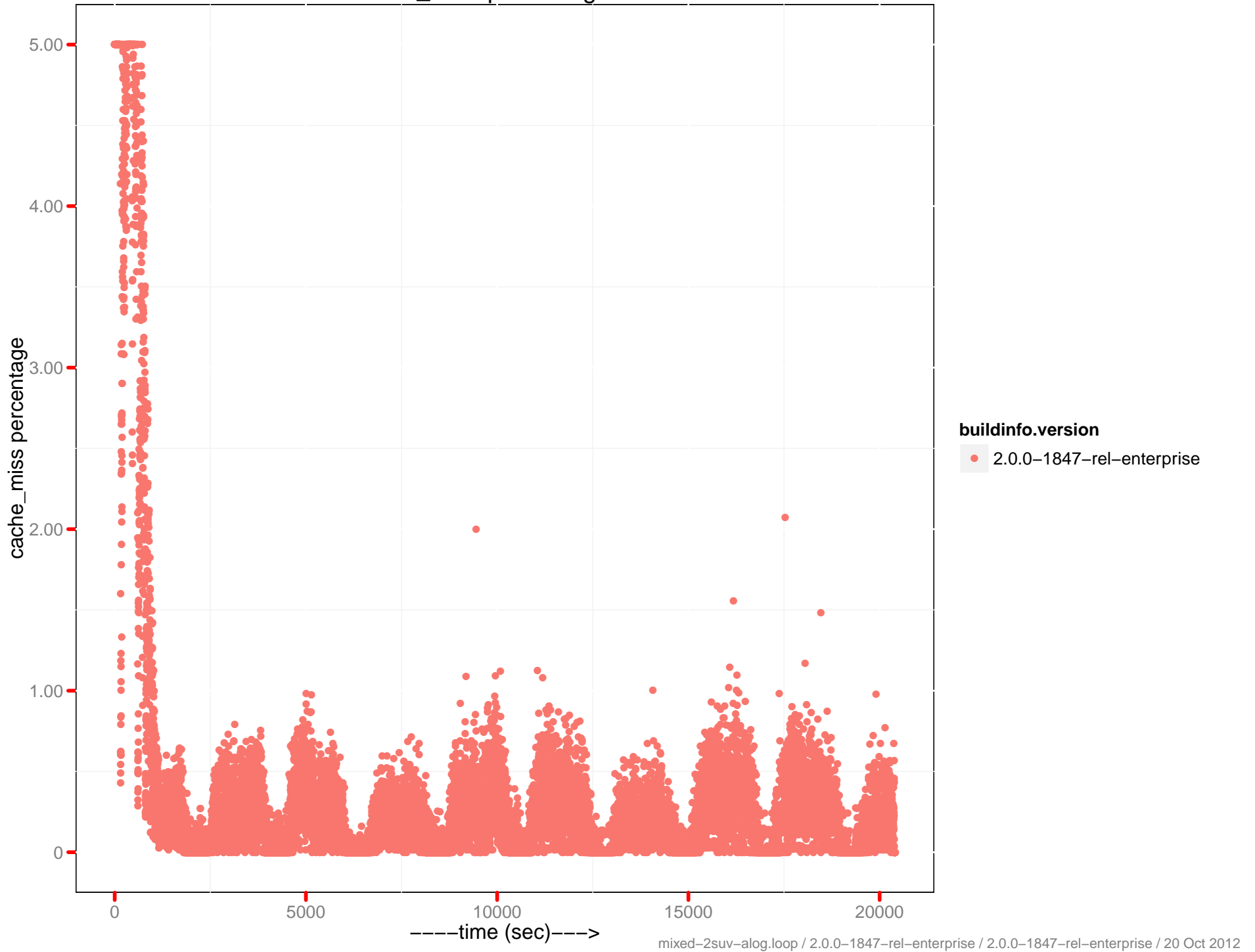
# # of get hits



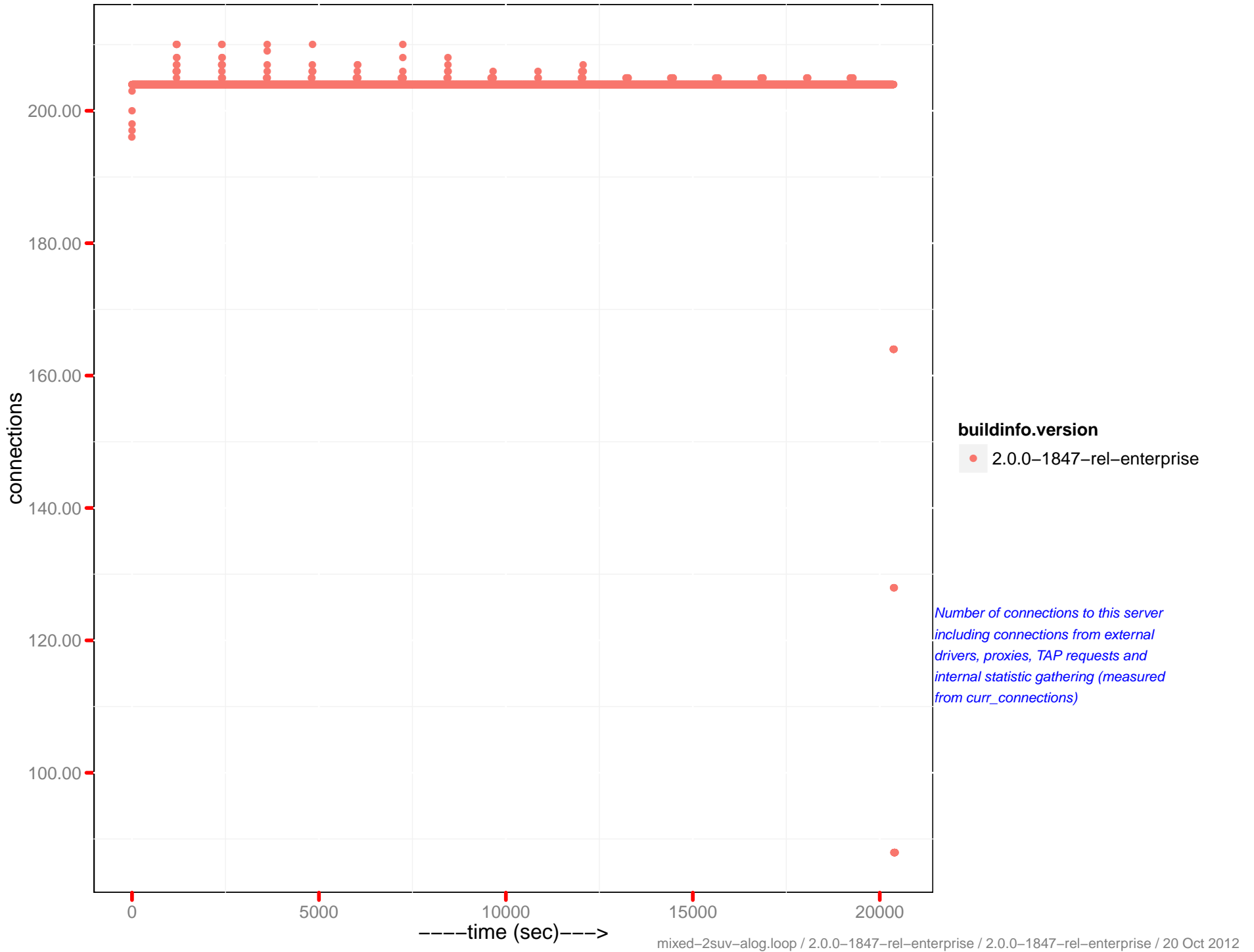
# cache\_miss percentage



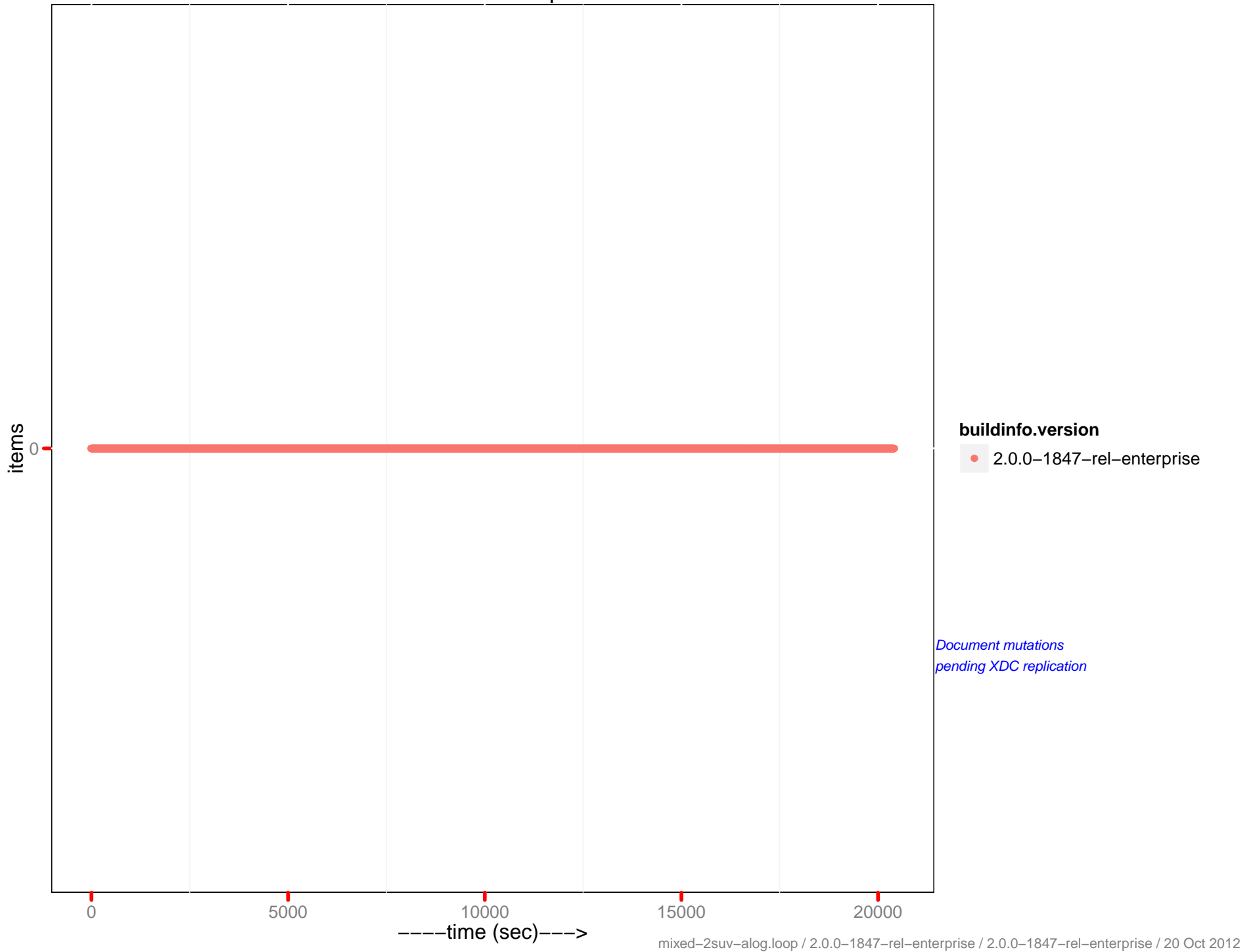
cache\_miss percentage 0-5



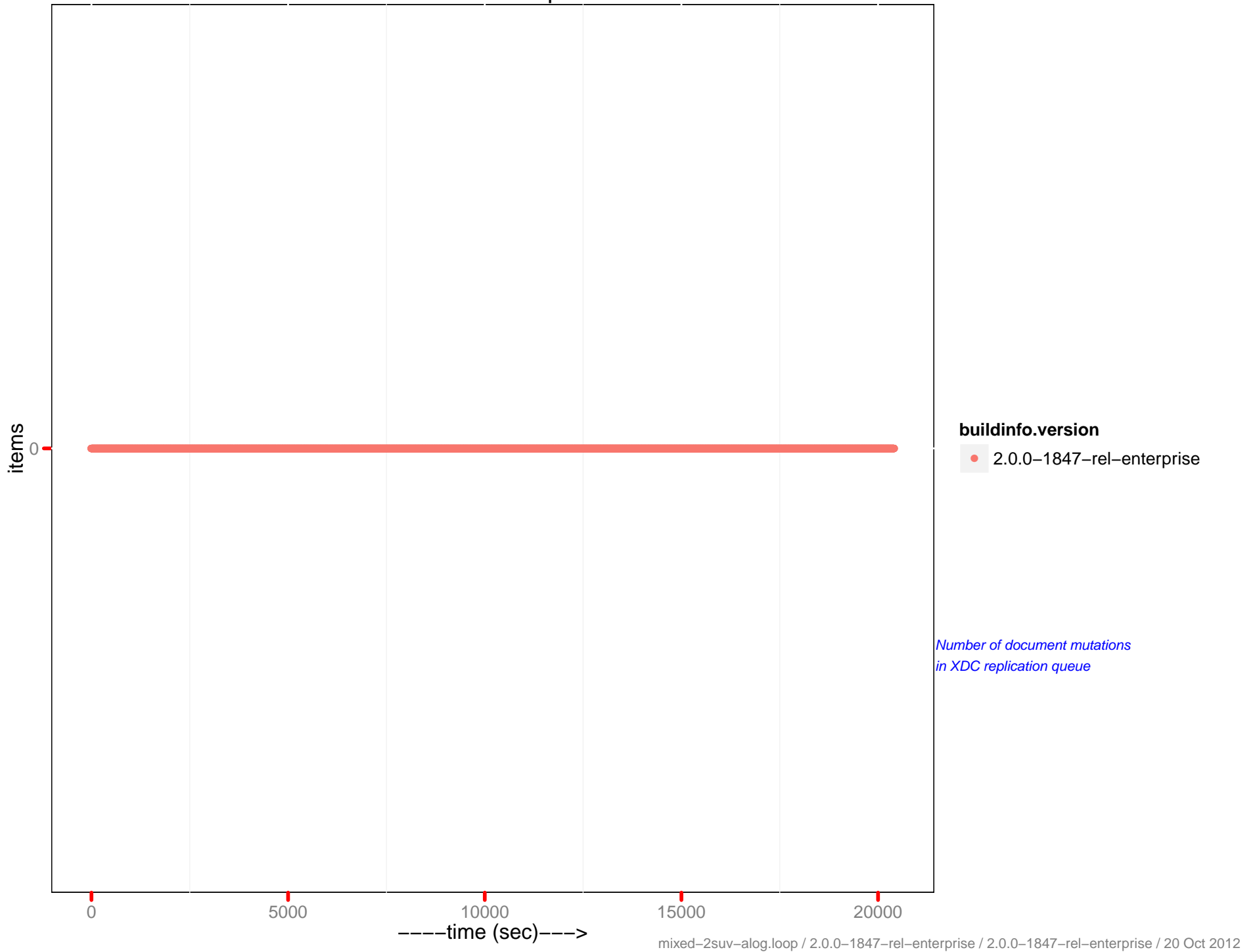
# Number of connections



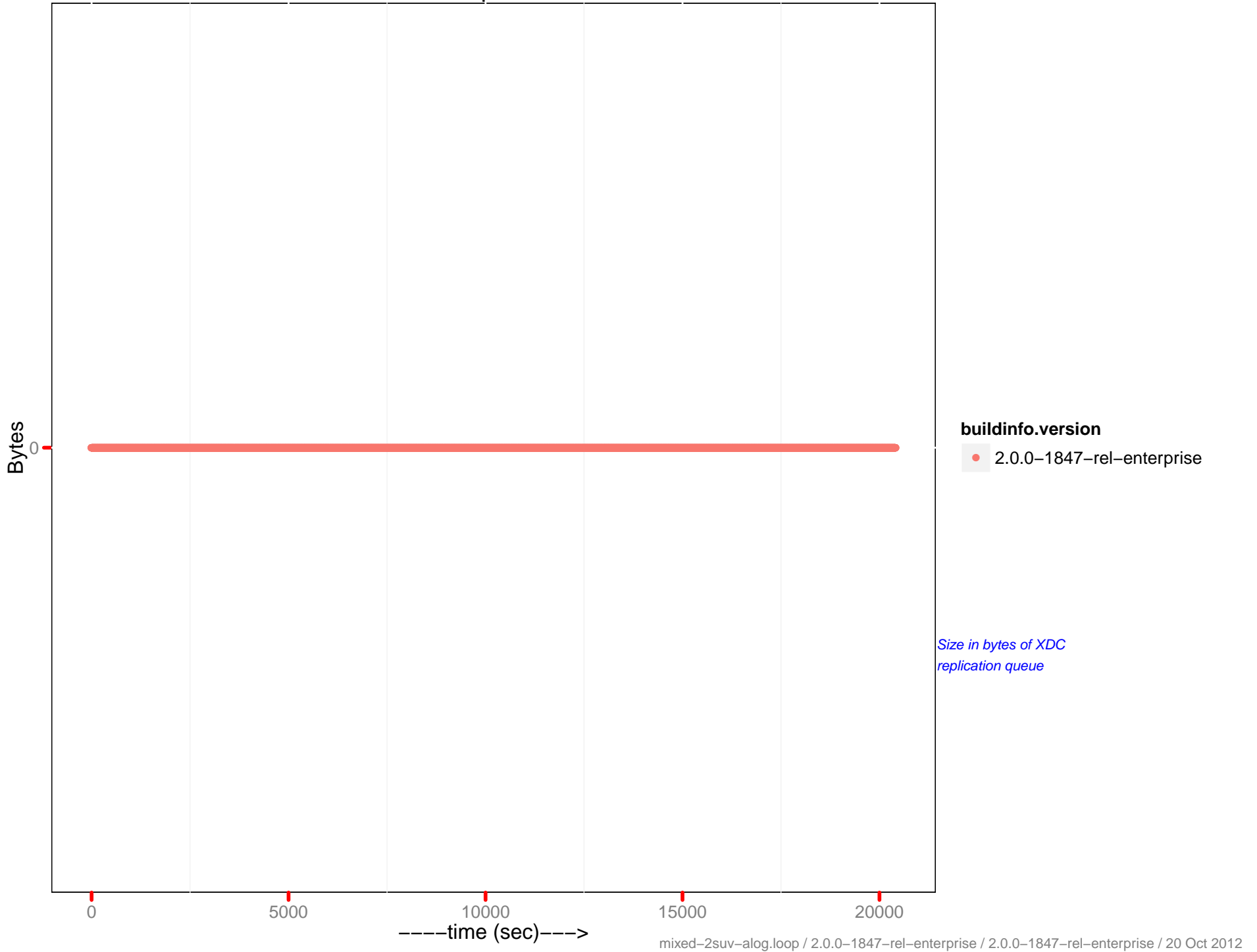
# XDCR docs to replicate



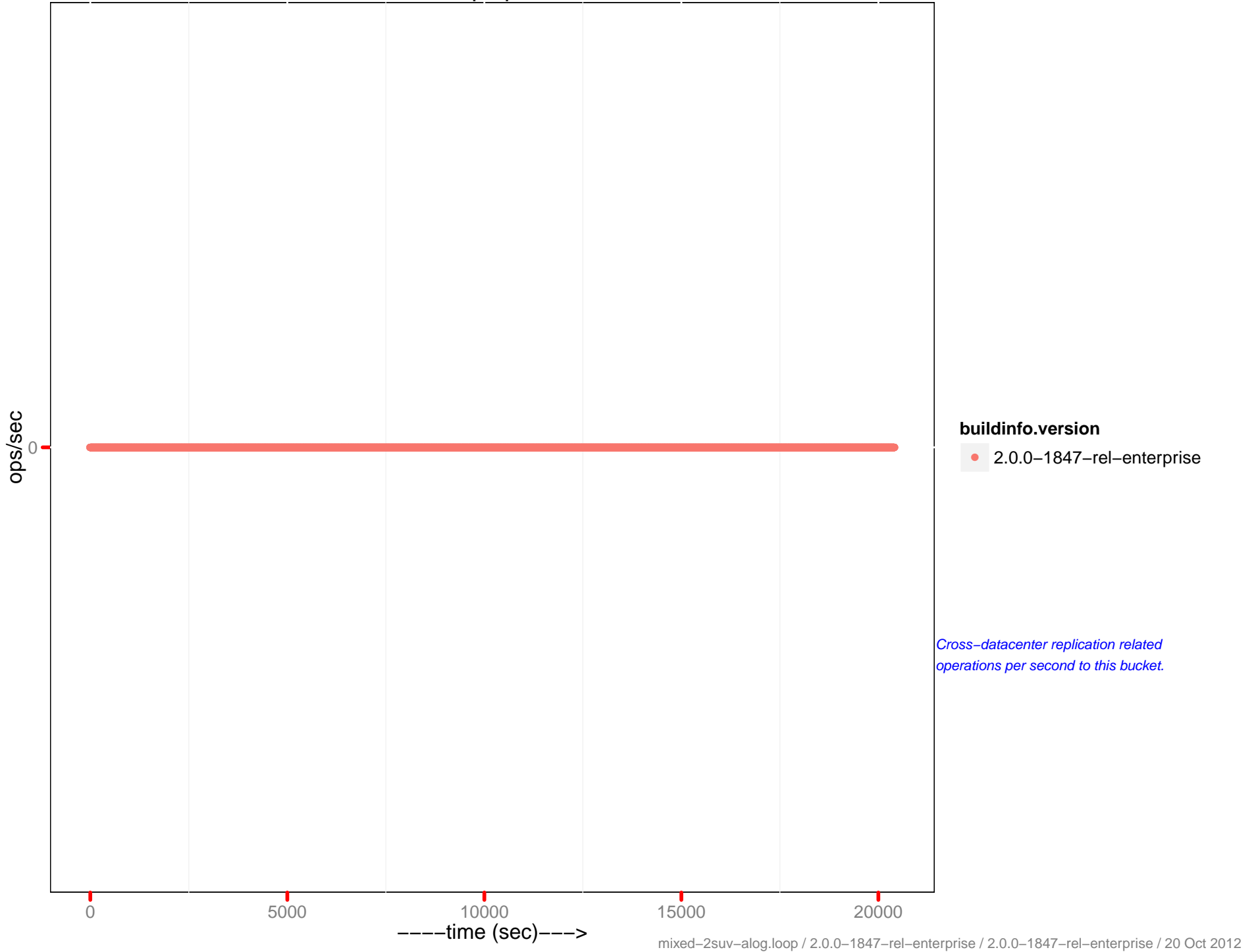
# XDCR docs in queue



# XDCR queue size

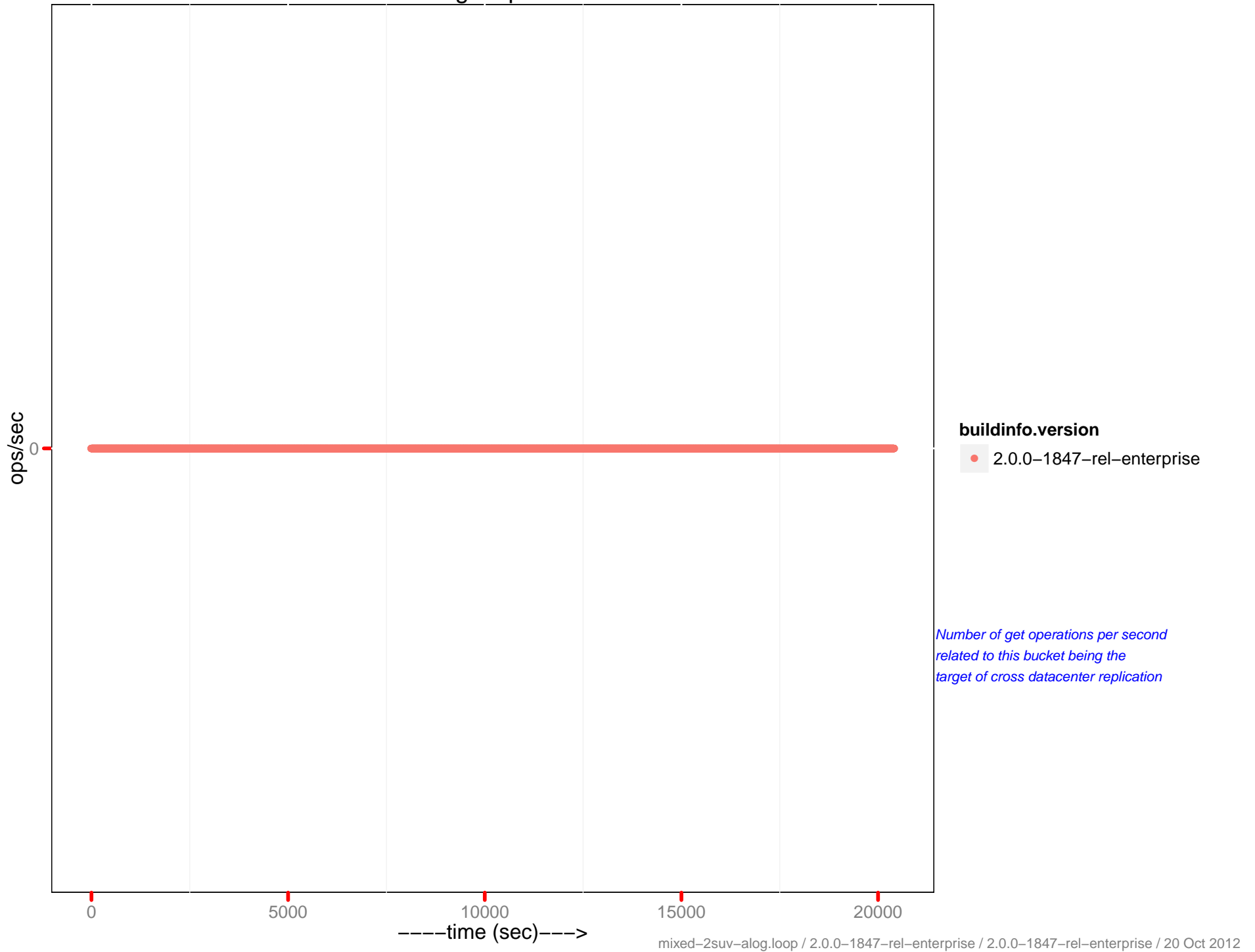


# XDC ops per sec

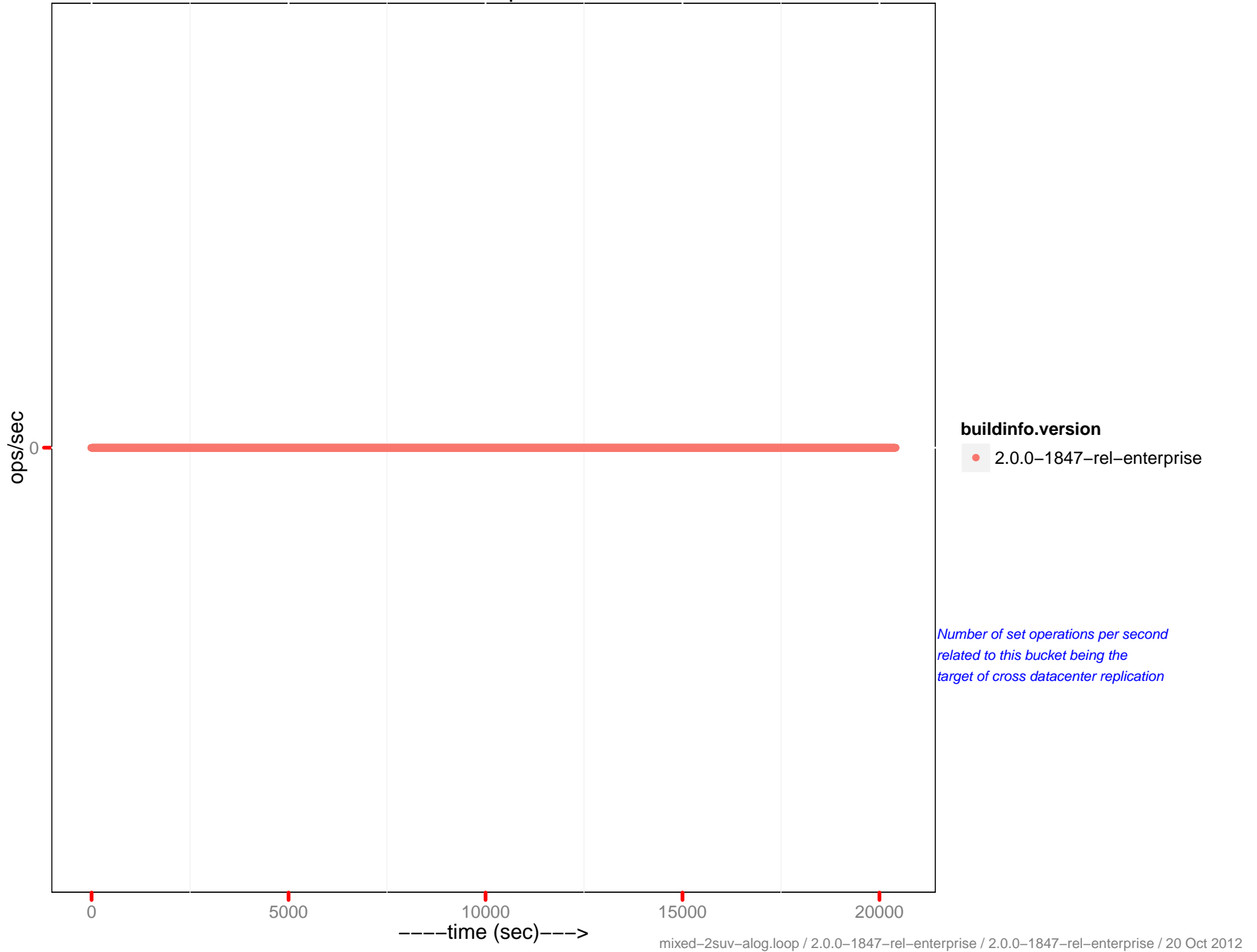




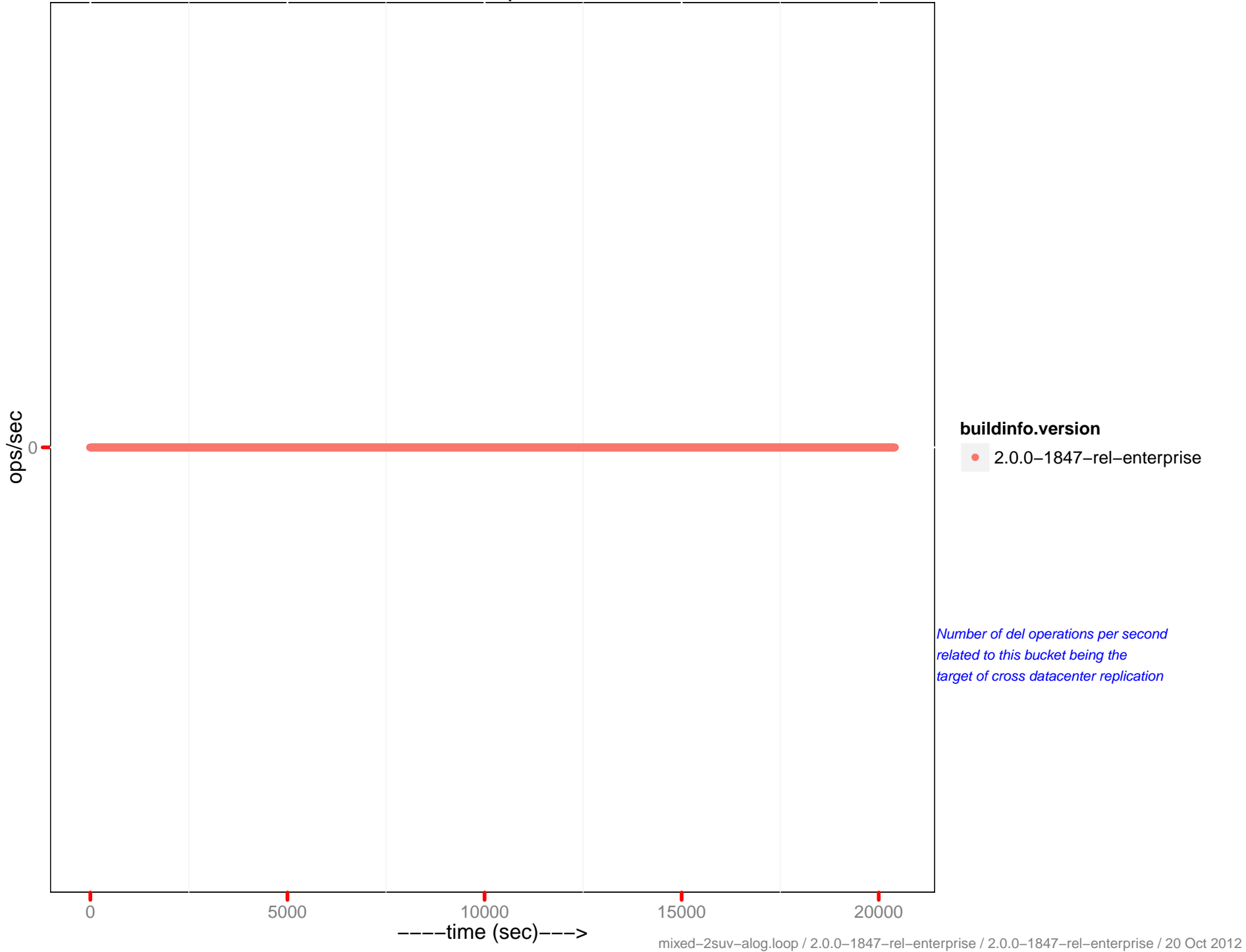
# XDC gets per sec



# XDC sets per sec



# XDC dels per sec

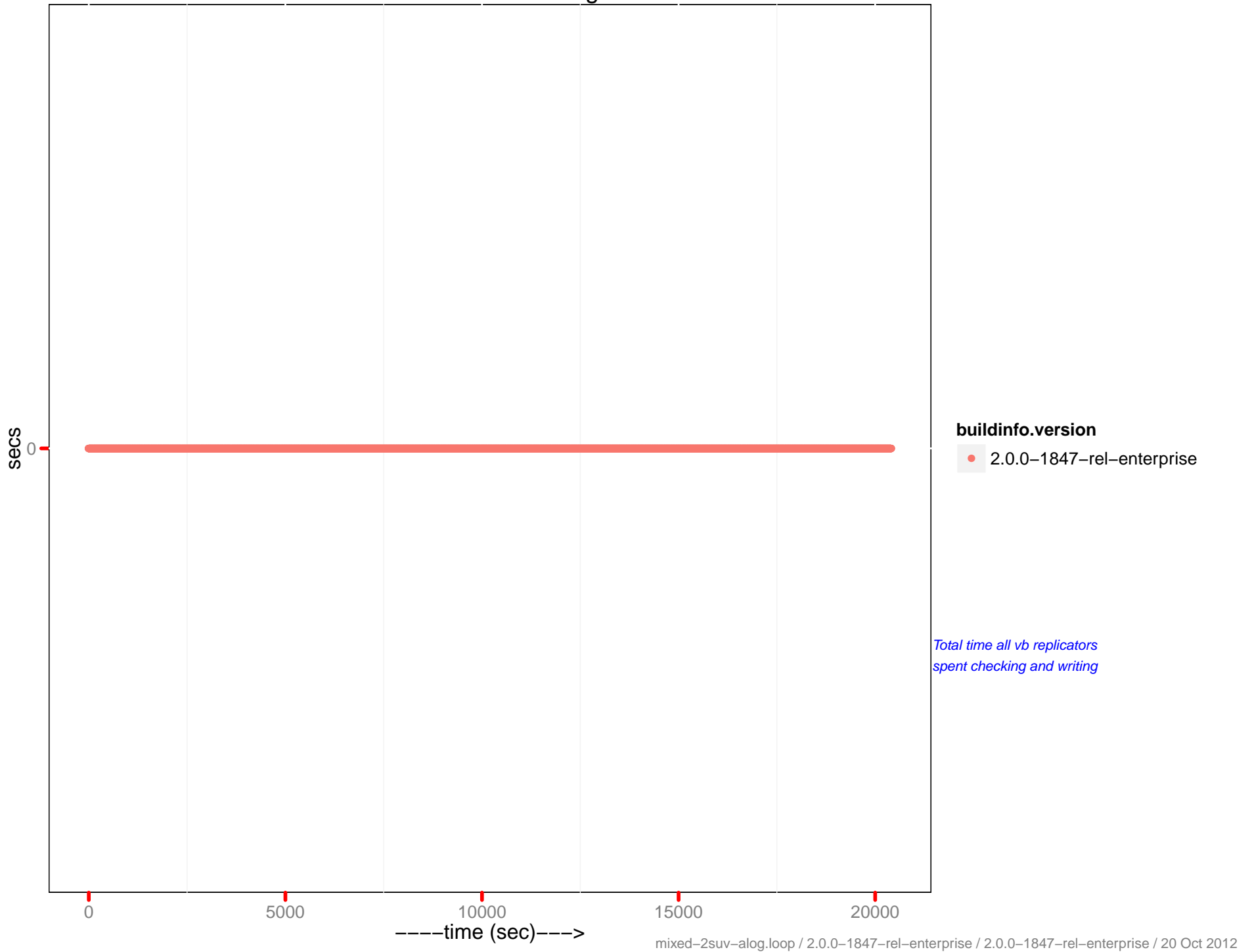


**buildinfo.version**

• 2.0.0-1847-rel-enterprise

*Number of del operations per second related to this bucket being the target of cross datacenter replication*

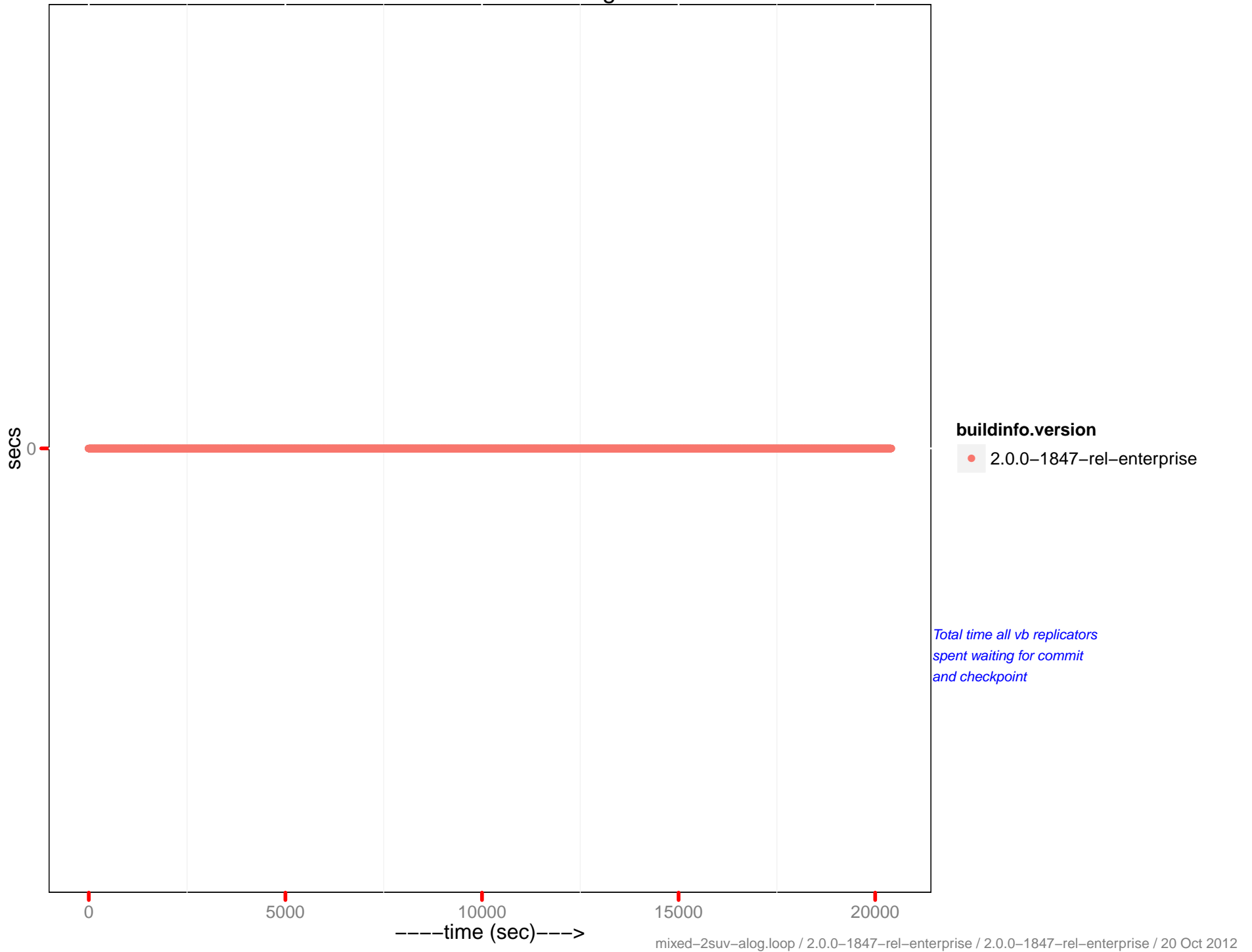
# XDCR secs working



**buildinfo.version**  
● 2.0.0-1847-rel-enterprise

*Total time all vb replicators  
spent checking and writing*

# XDCR secs committing

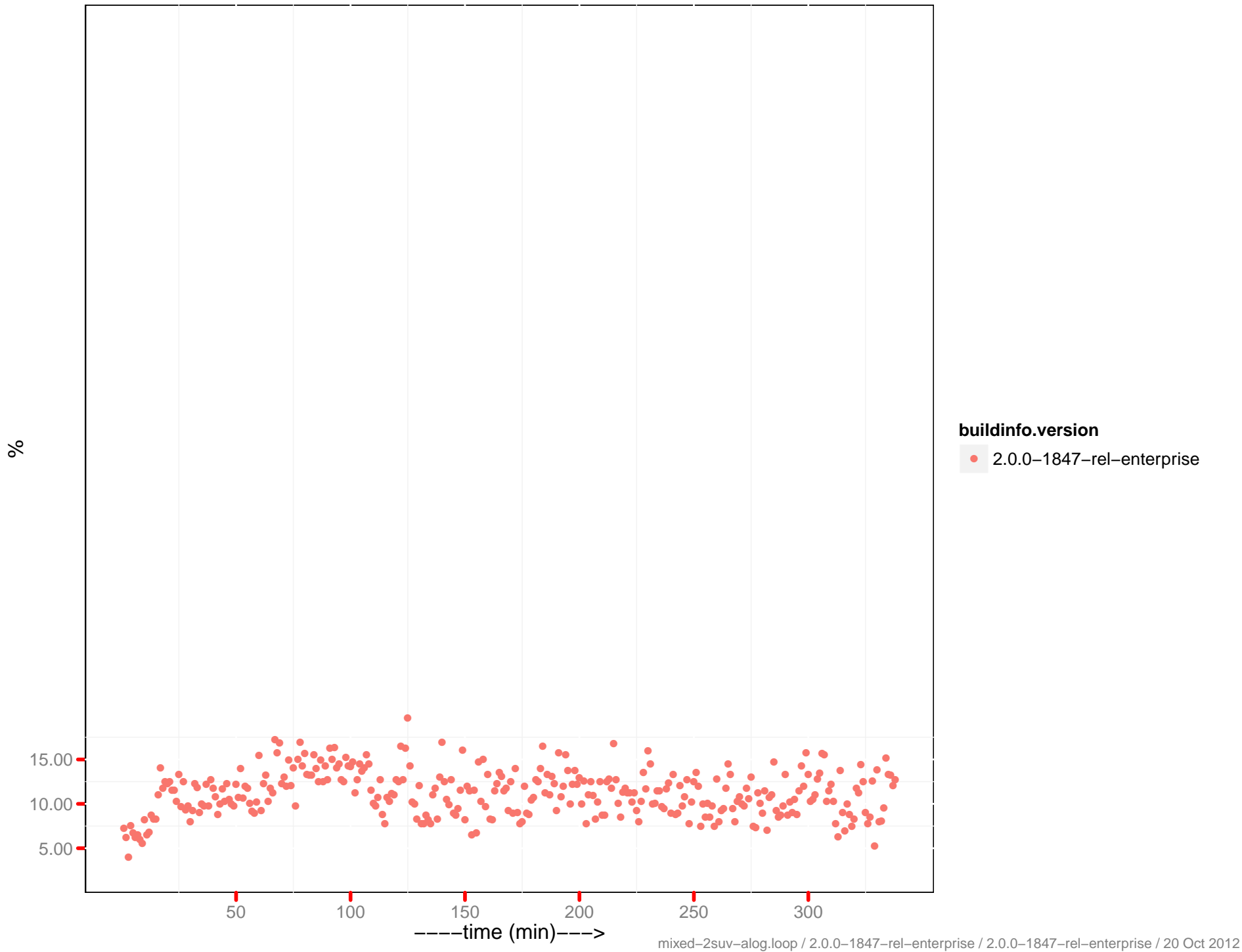


**buildinfo.version**

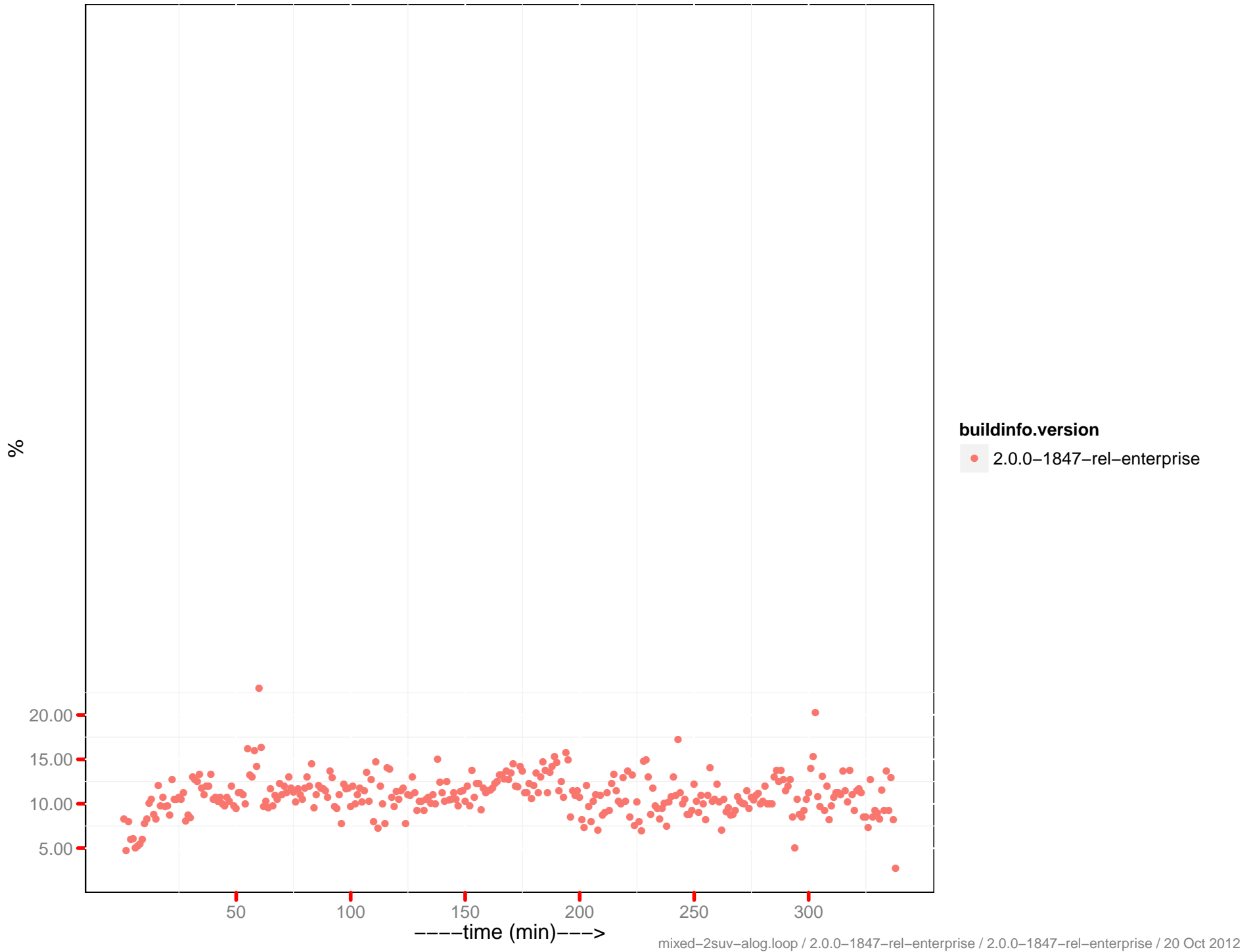
• 2.0.0-1847-rel-enterprise

*Total time all vb replicators  
spent waiting for commit  
and checkpoint*

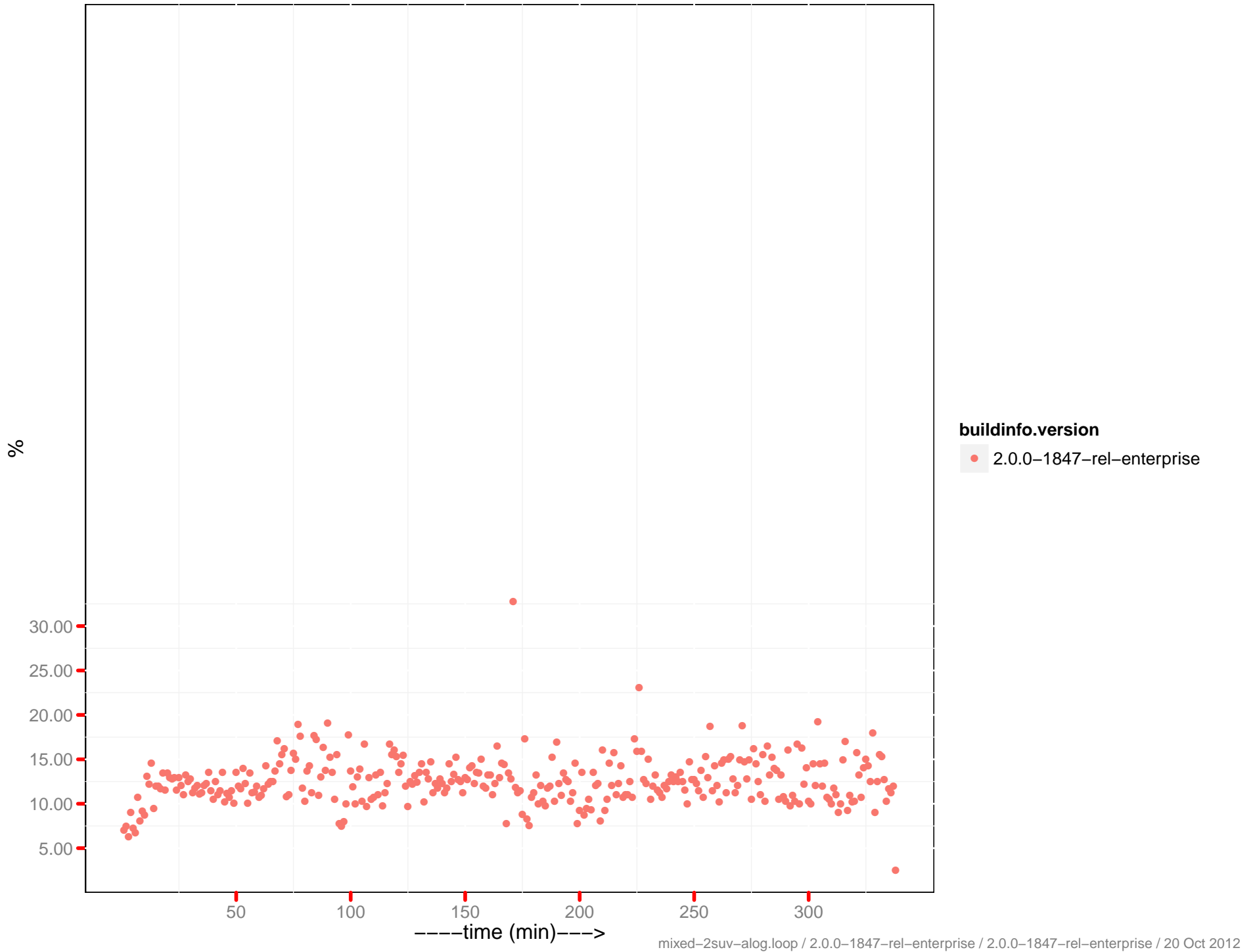
# CPU utilization – 10.2.1.58:8091



# CPU utilization – 10.2.1.61:8091

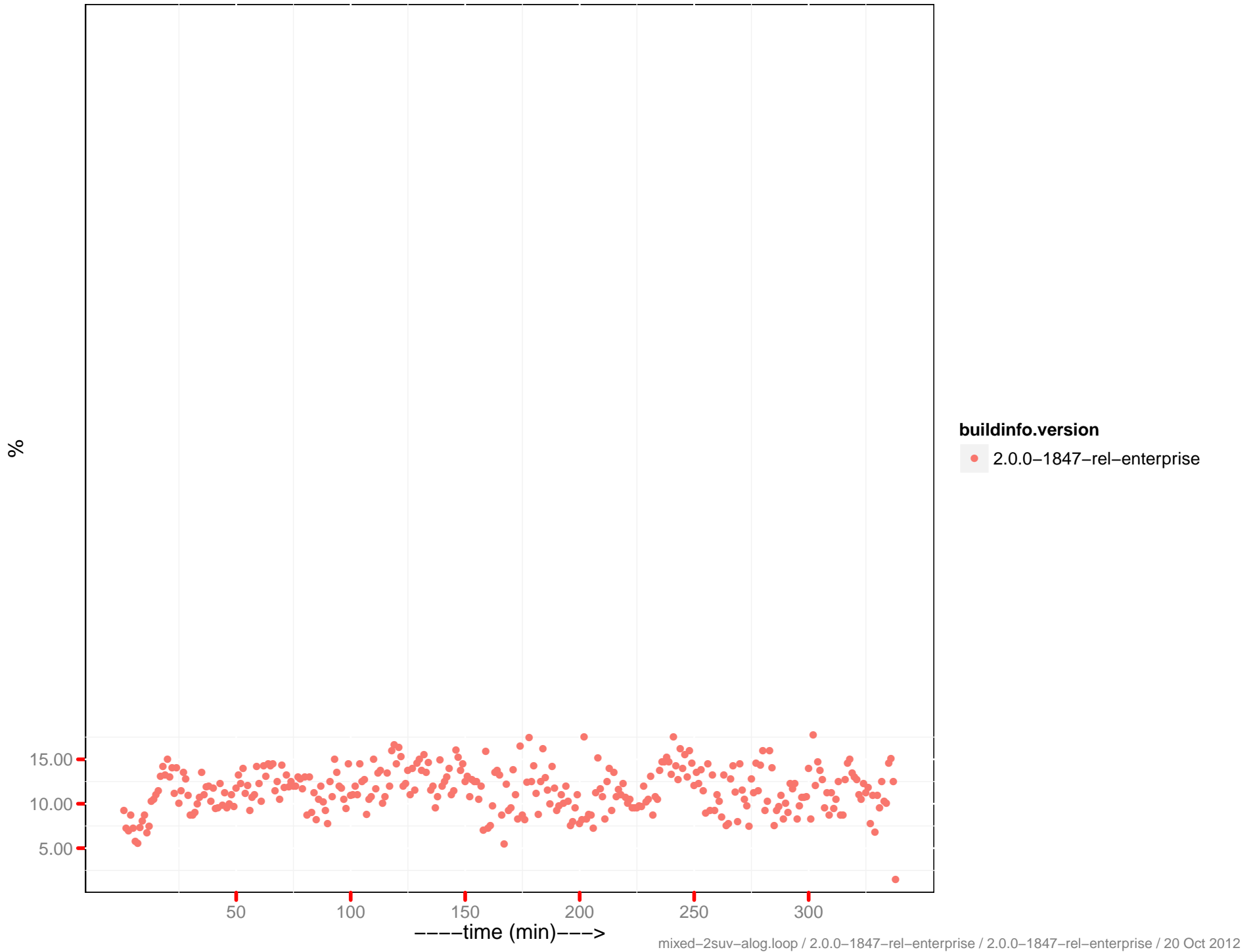


# CPU utilization – 10.2.1.63:8091

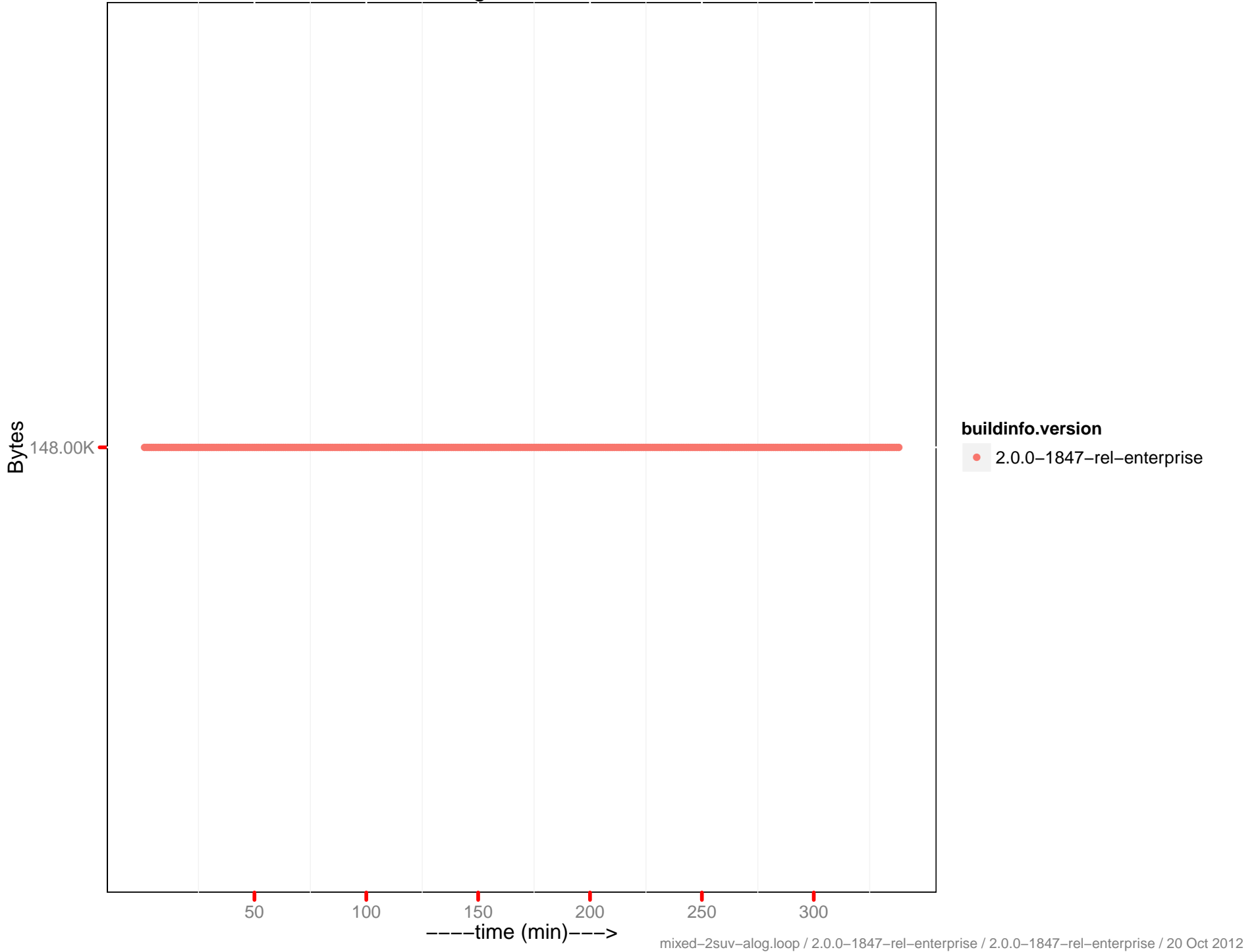




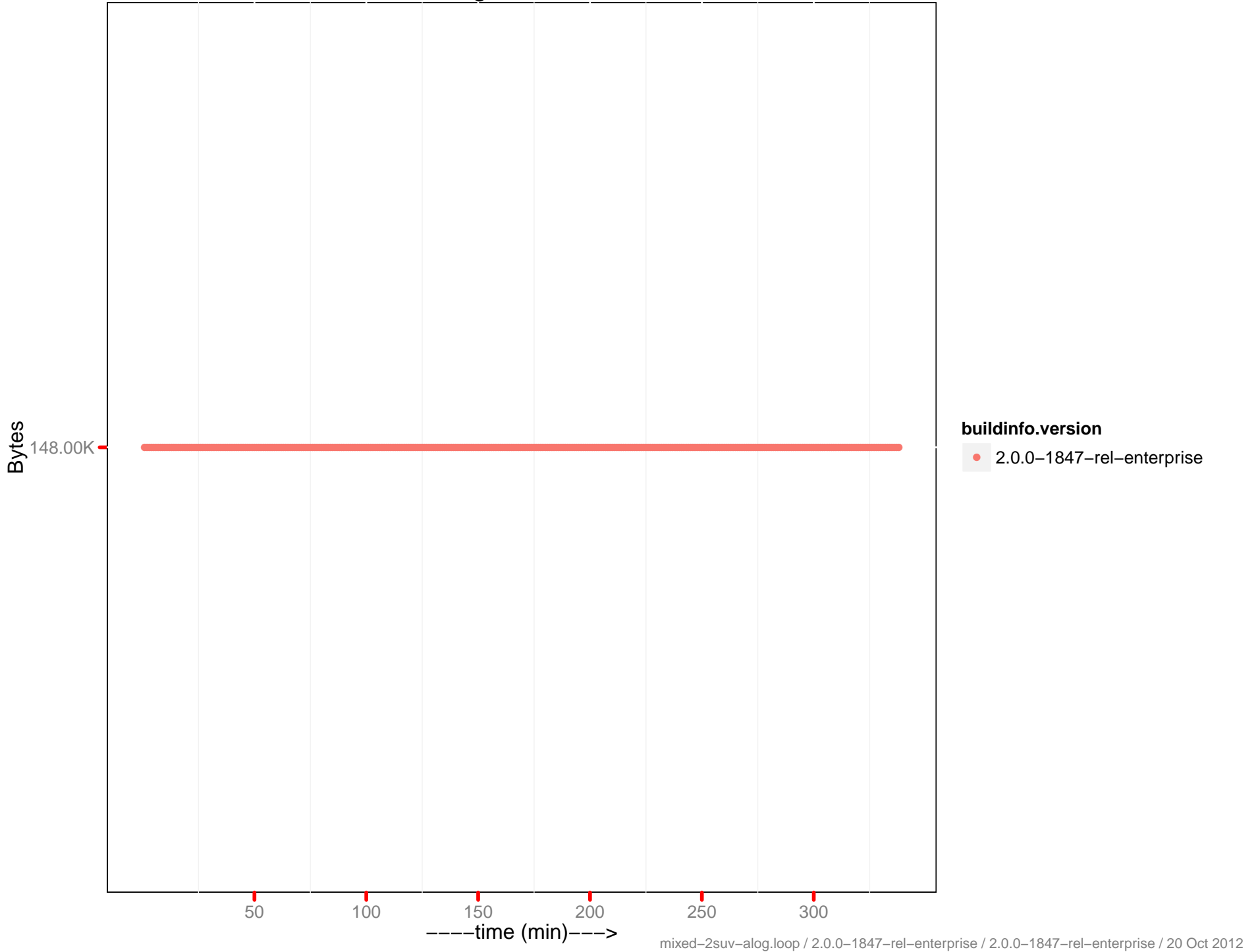
# CPU utilization – 10.2.1.64:8091



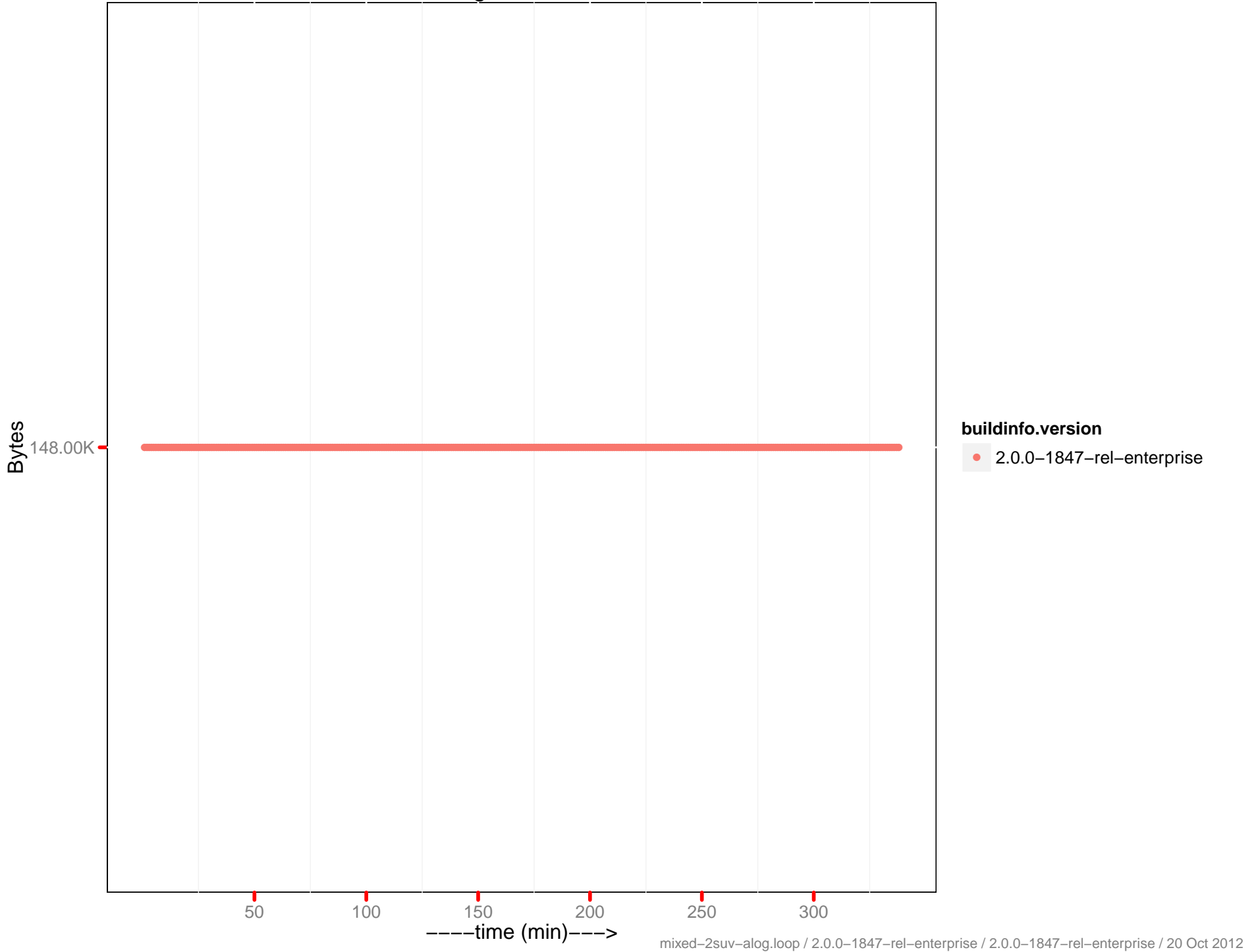
# SWAP Usage - 10.2.1.58:8091



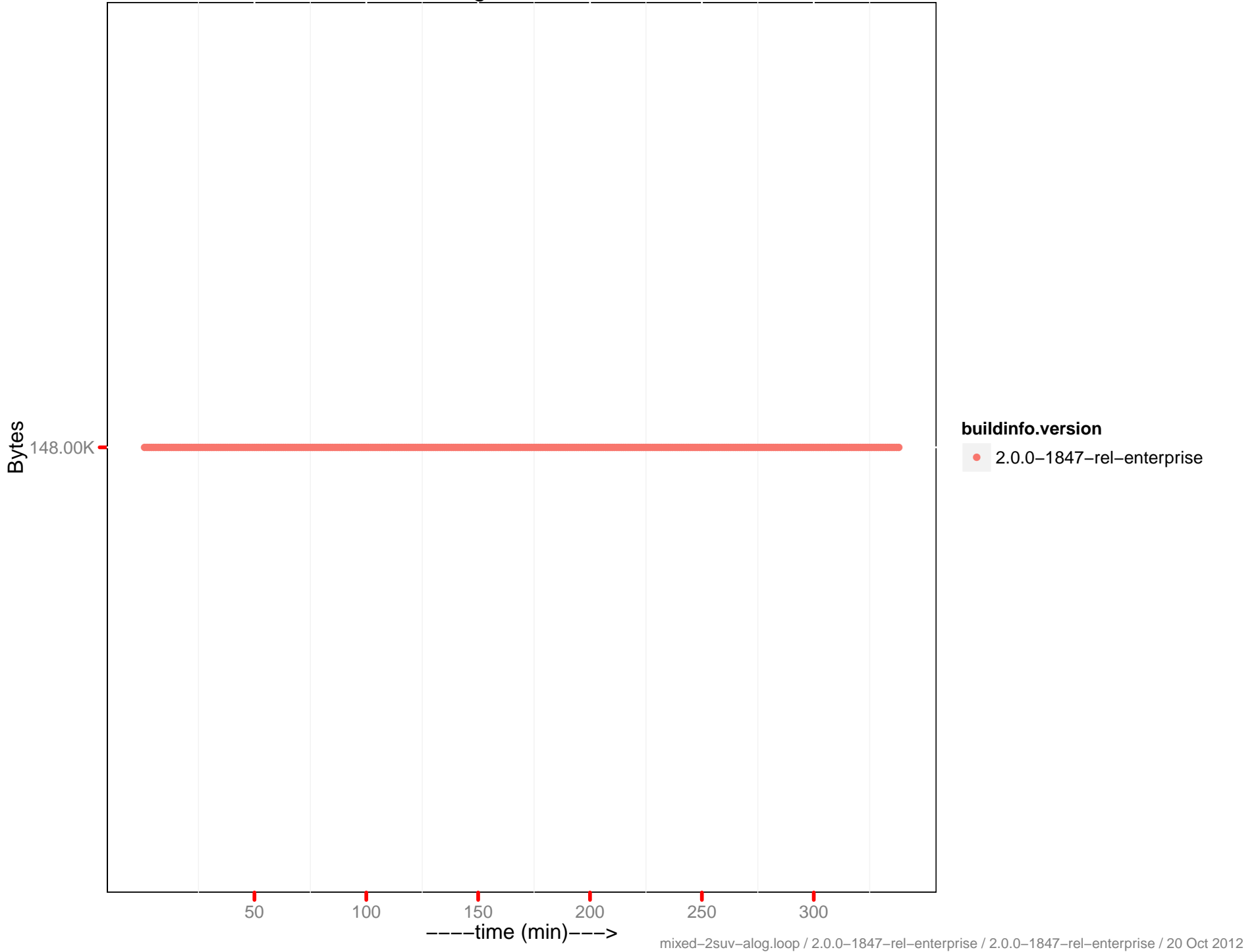
# SWAP Usage - 10.2.1.61:8091



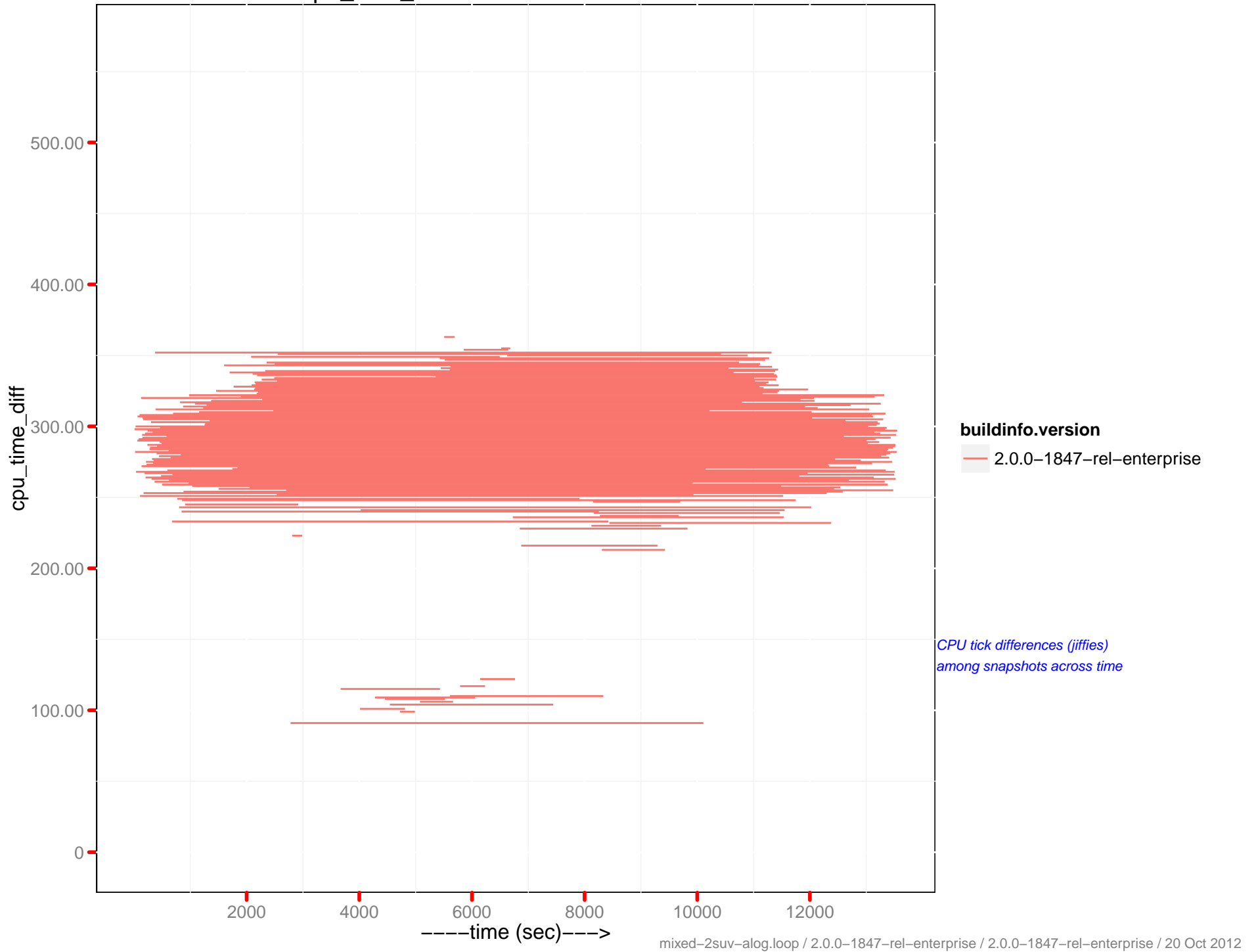
# SWAP Usage - 10.2.1.63:8091



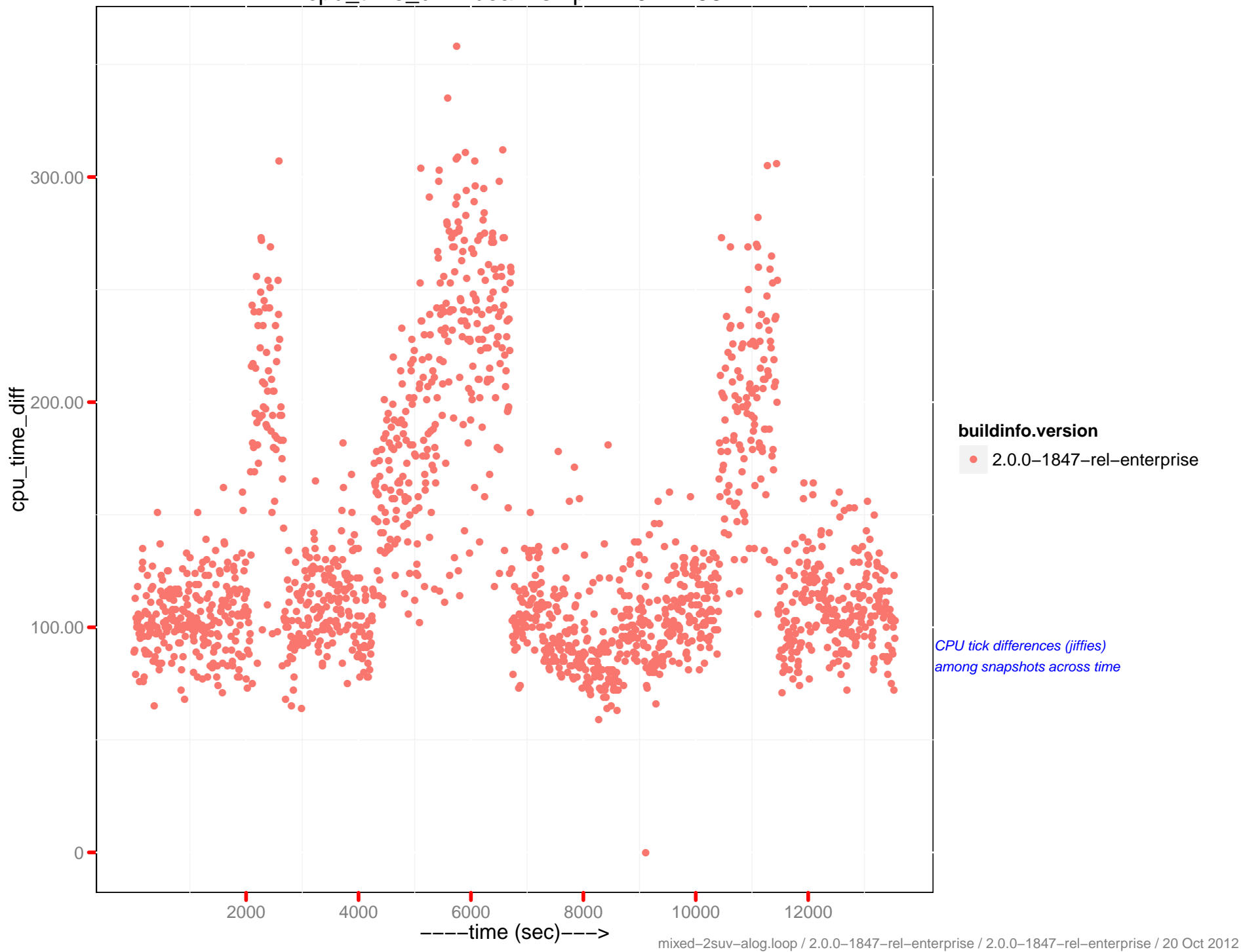
# SWAP Usage - 10.2.1.64:8091



# cpu\_time\_diff: memcached - 10.2.1.58



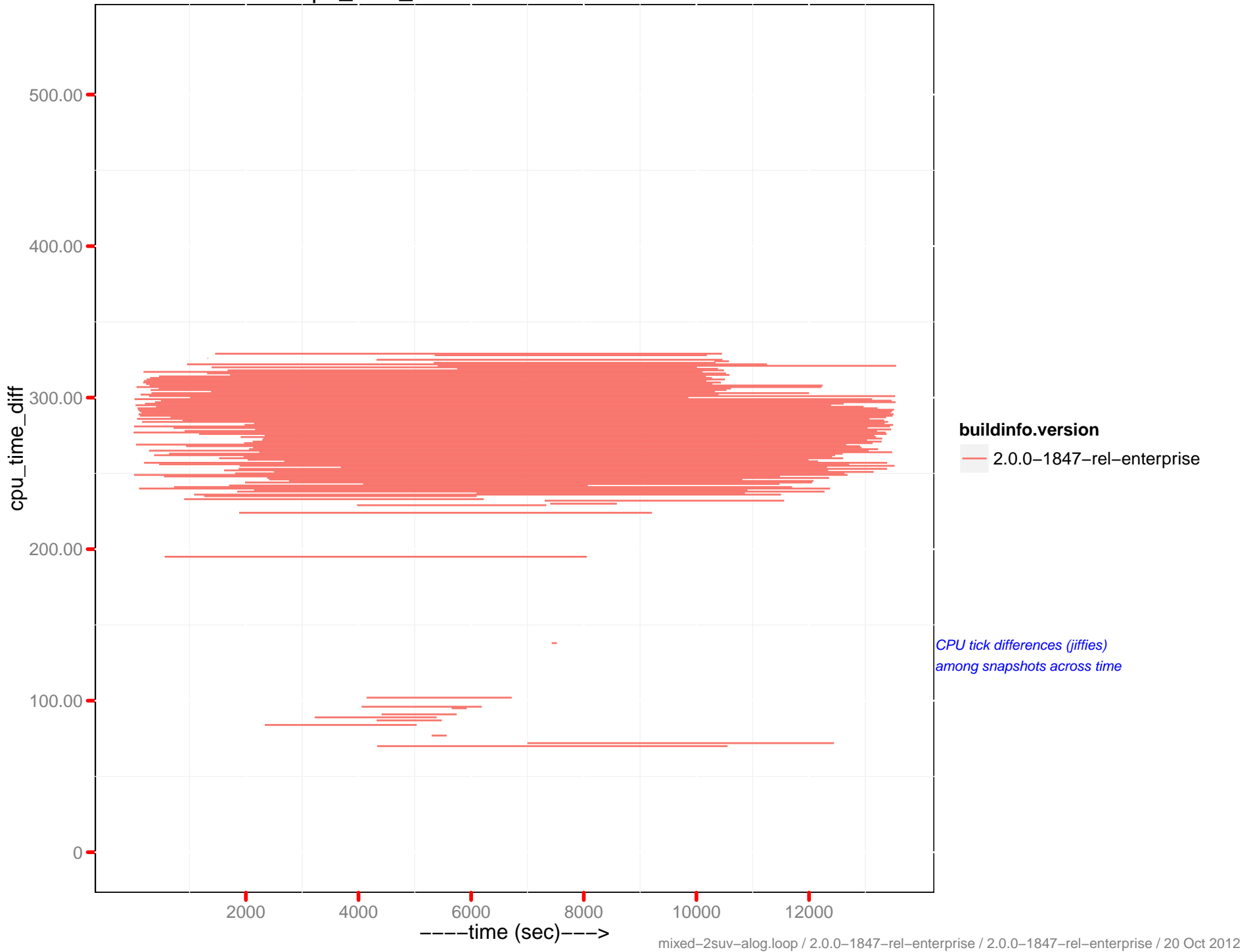
cpu\_time\_diff : beam.smp - 10.2.1.58



**buildinfo.version**  
● 2.0.0-1847-rel-enterprise

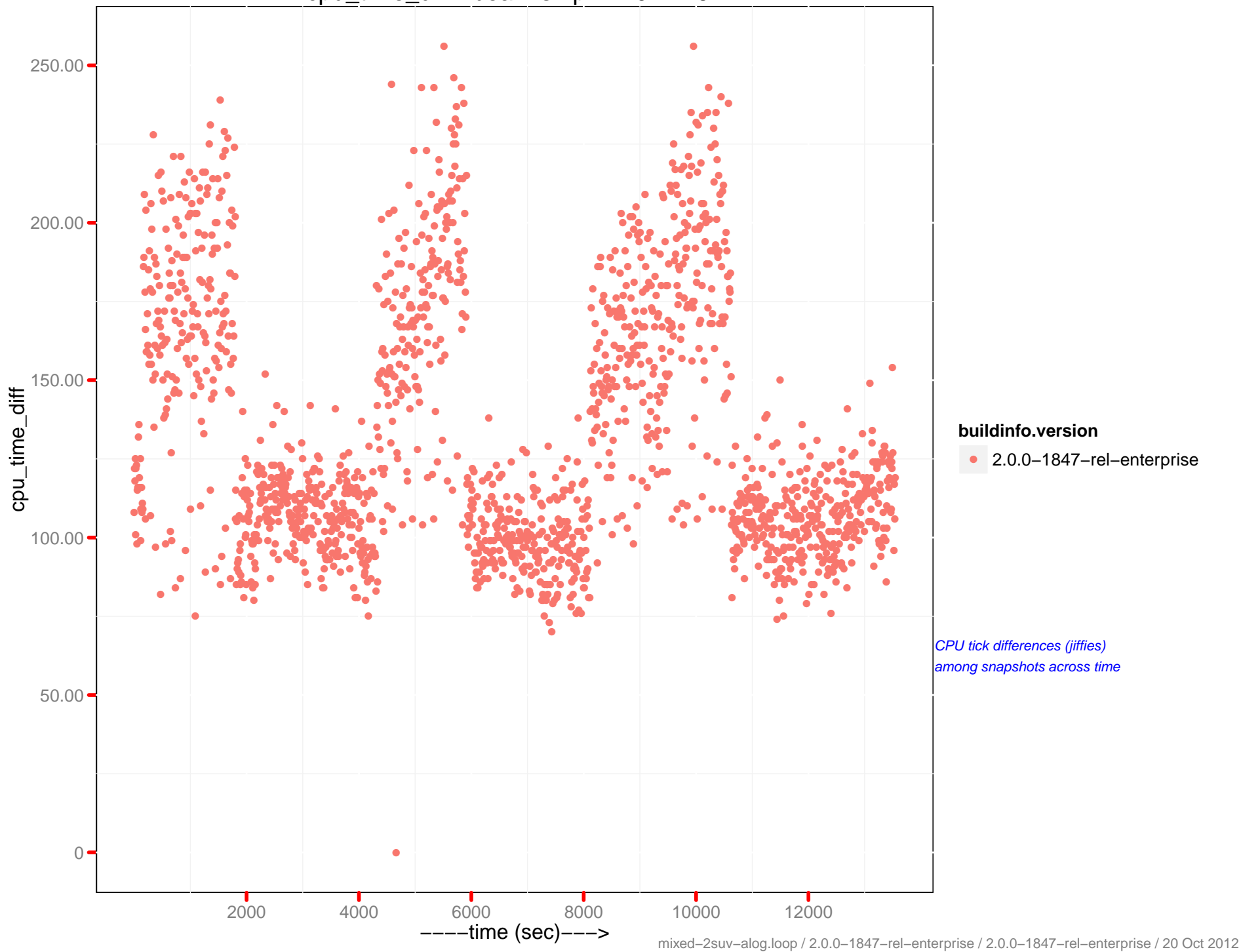
*CPU tick differences (jiffies)  
among snapshots across time*

# cpu\_time\_diff: memcached - 10.2.1.61

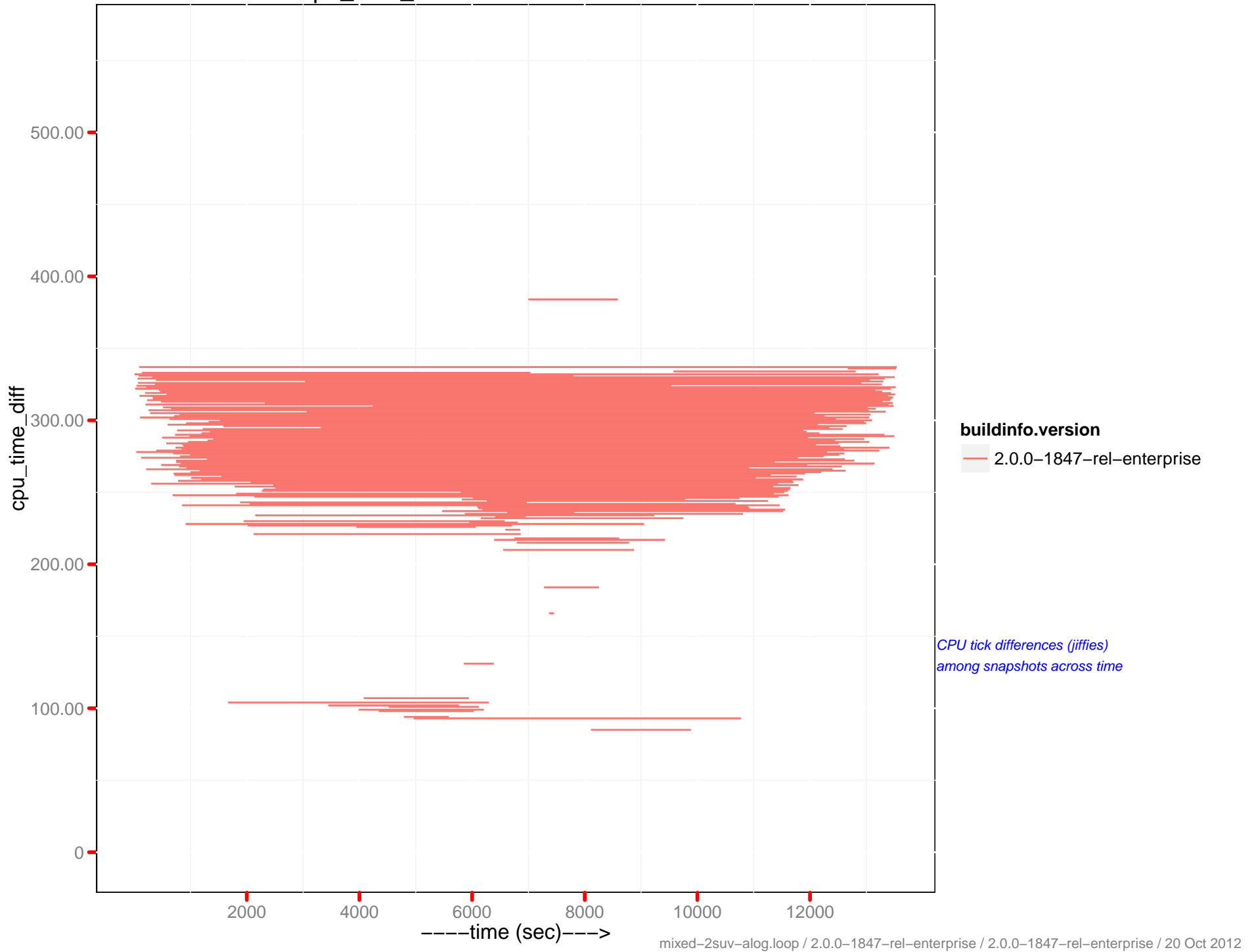




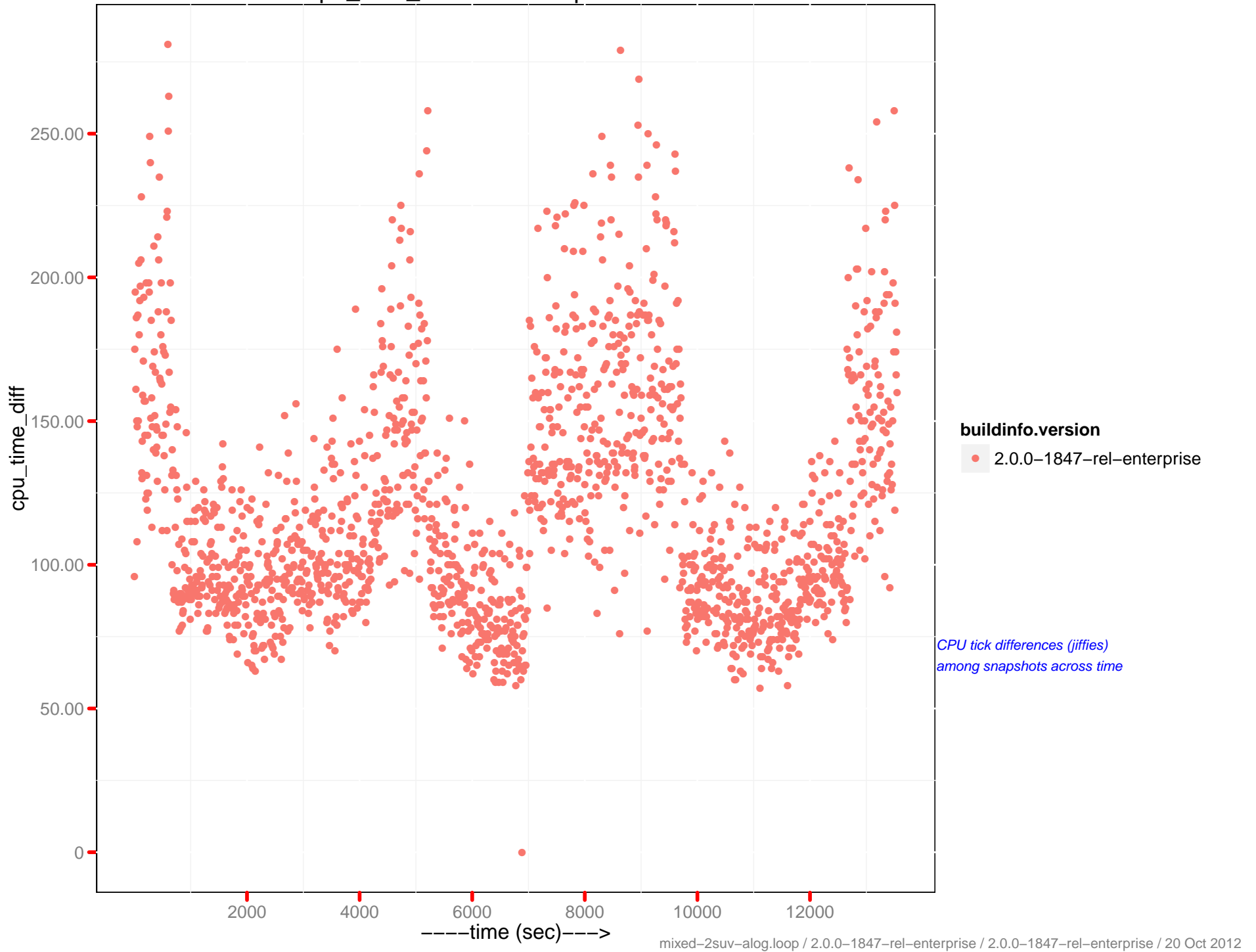
cpu\_time\_diff : beam.smp - 10.2.1.61



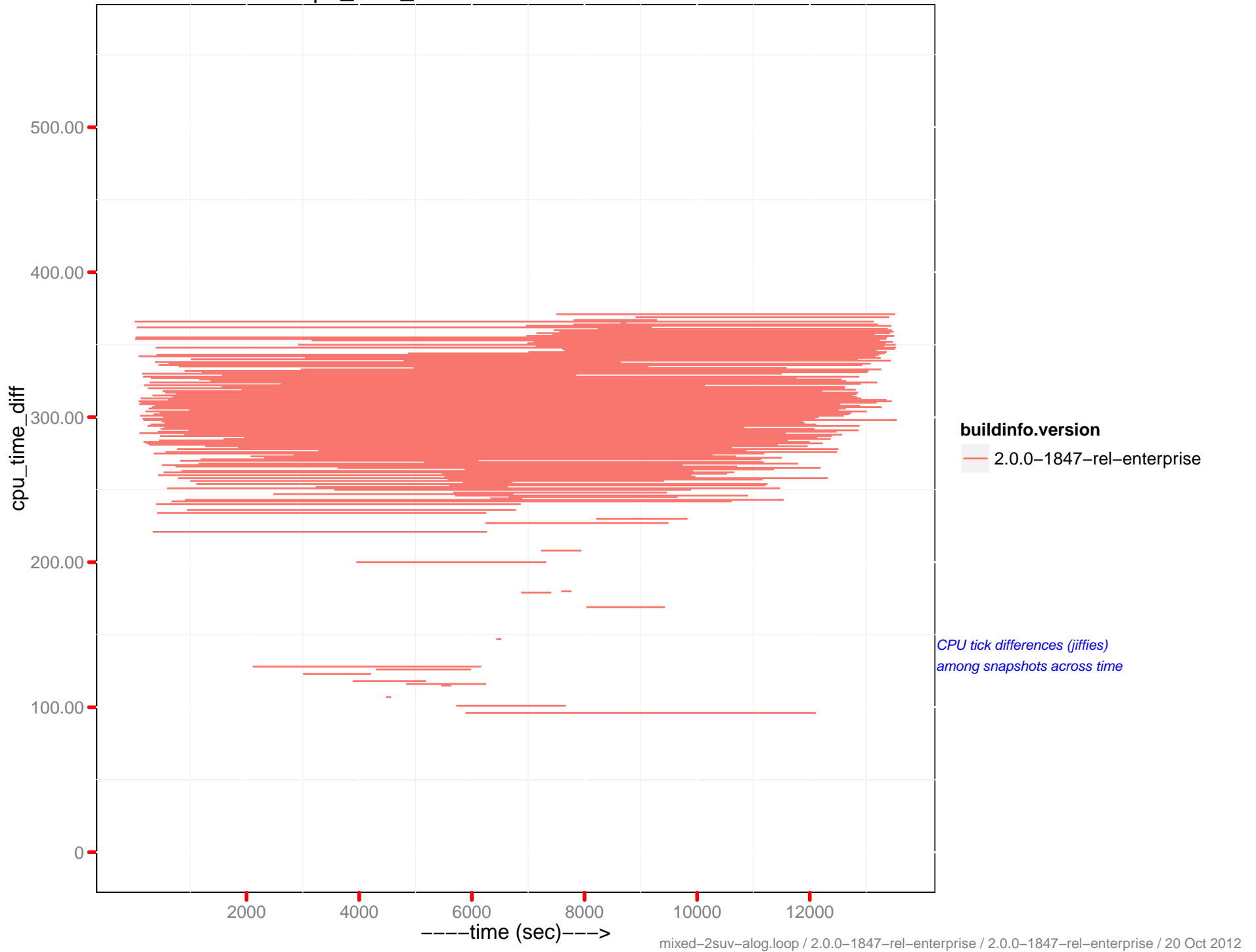
# cpu\_time\_diff: memcached – 10.2.1.63



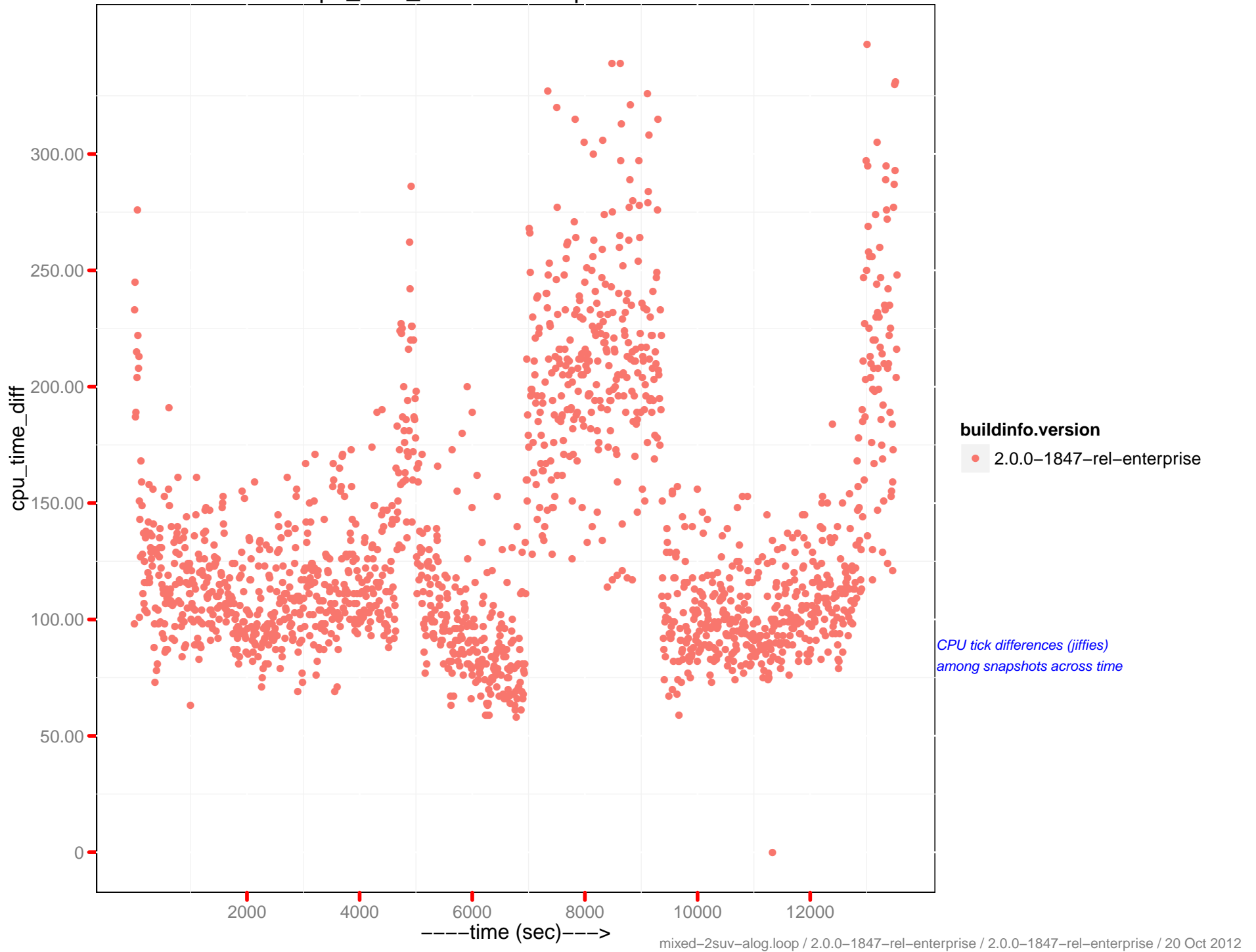
cpu\_time\_diff : beam.smp - 10.2.1.63



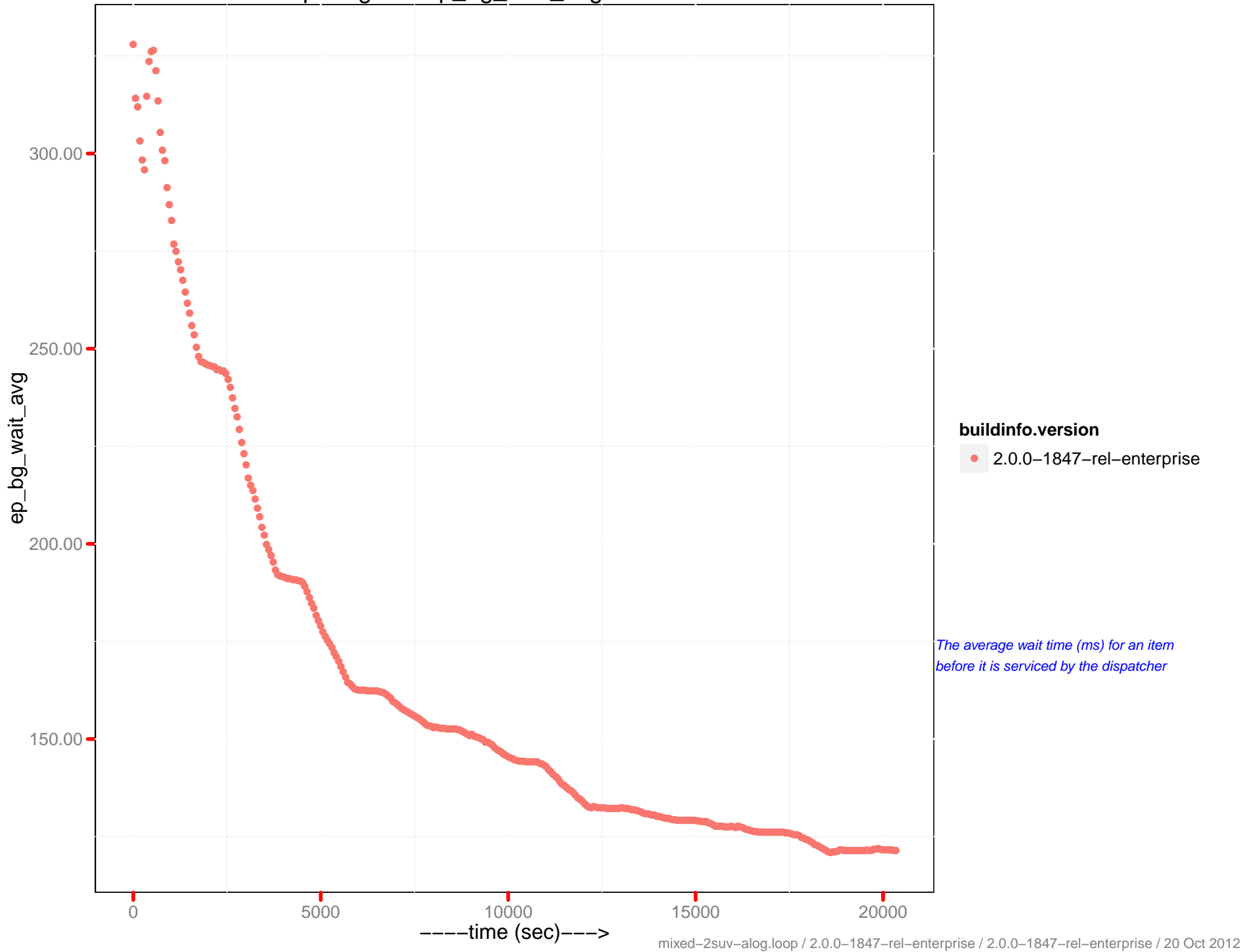
# cpu\_time\_diff: memcached – 10.2.1.64



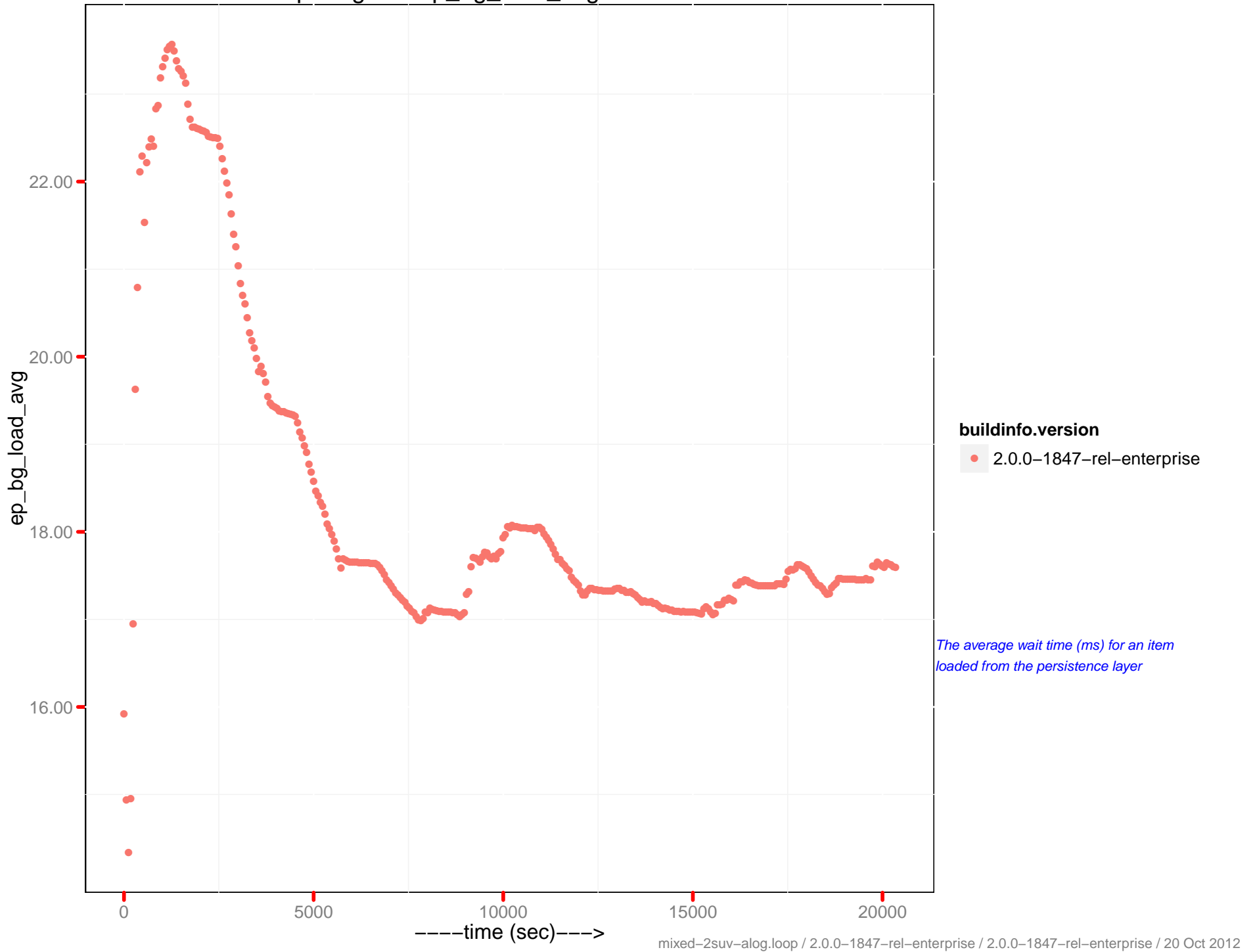
# cpu\_time\_diff : beam.smp - 10.2.1.64



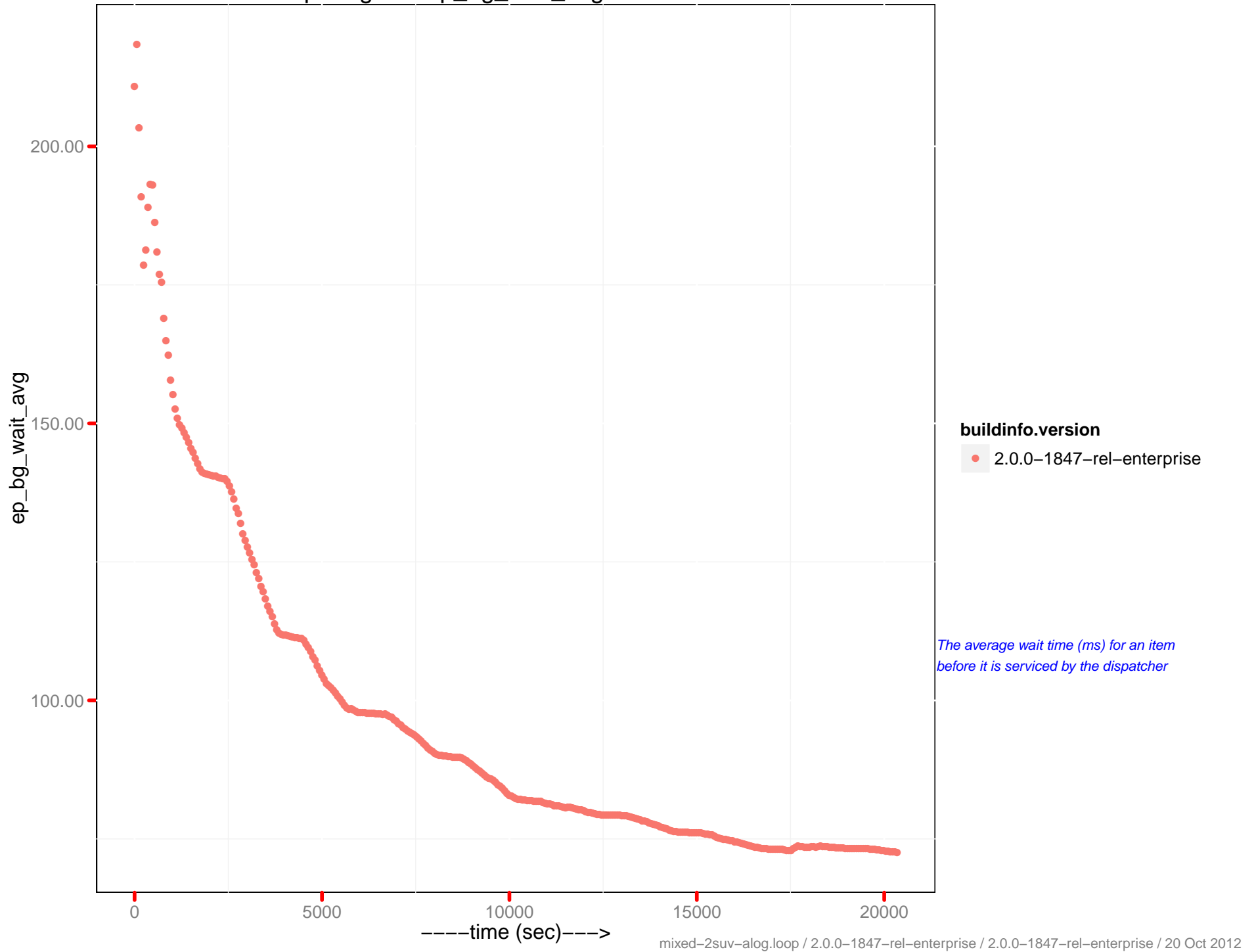
# ep-engine : ep\_bg\_wait\_avg - 10.2.1.58



# ep-engine : ep\_bg\_load\_avg - 10.2.1.58

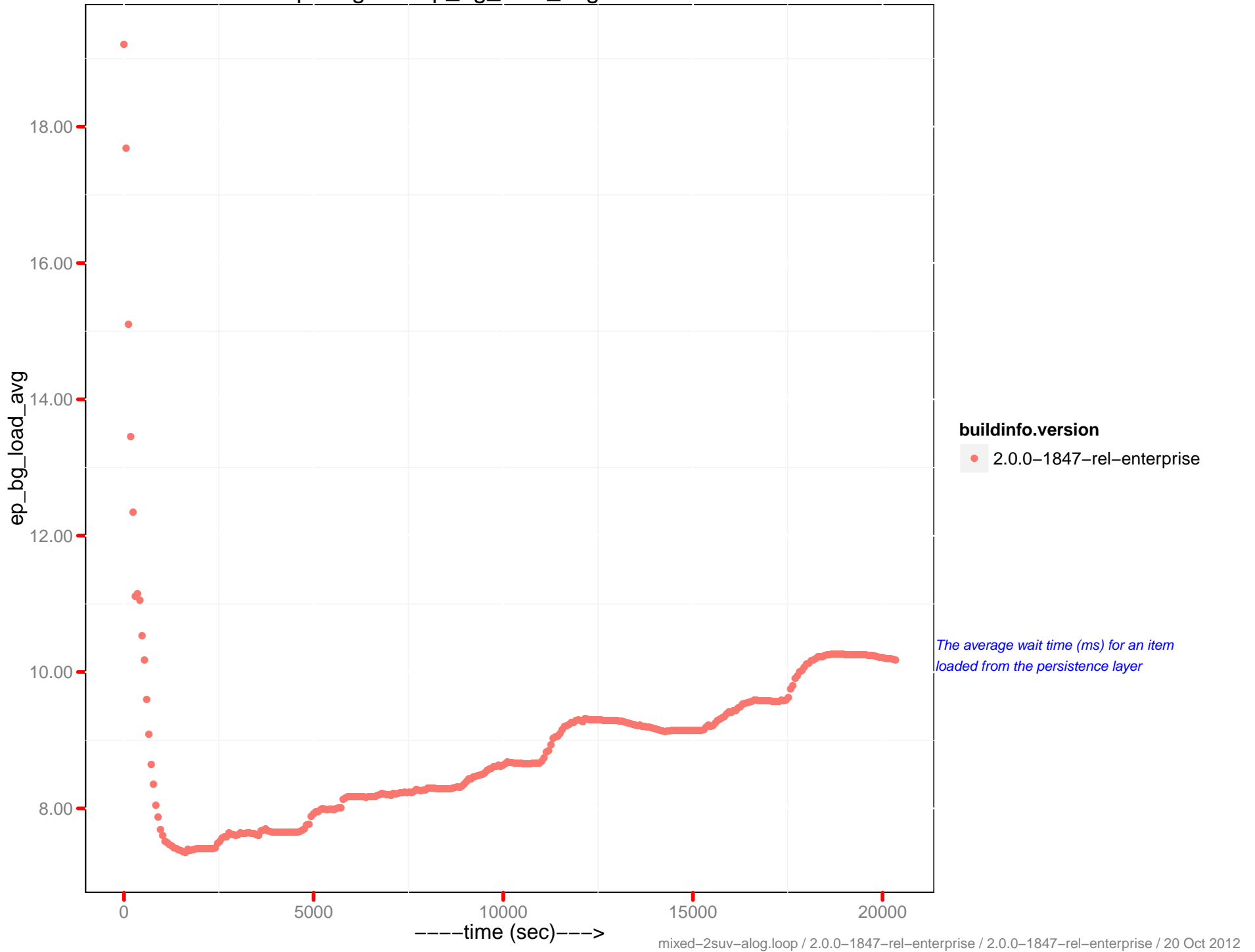


# ep-engine : ep\_bg\_wait\_avg - 10.2.1.61

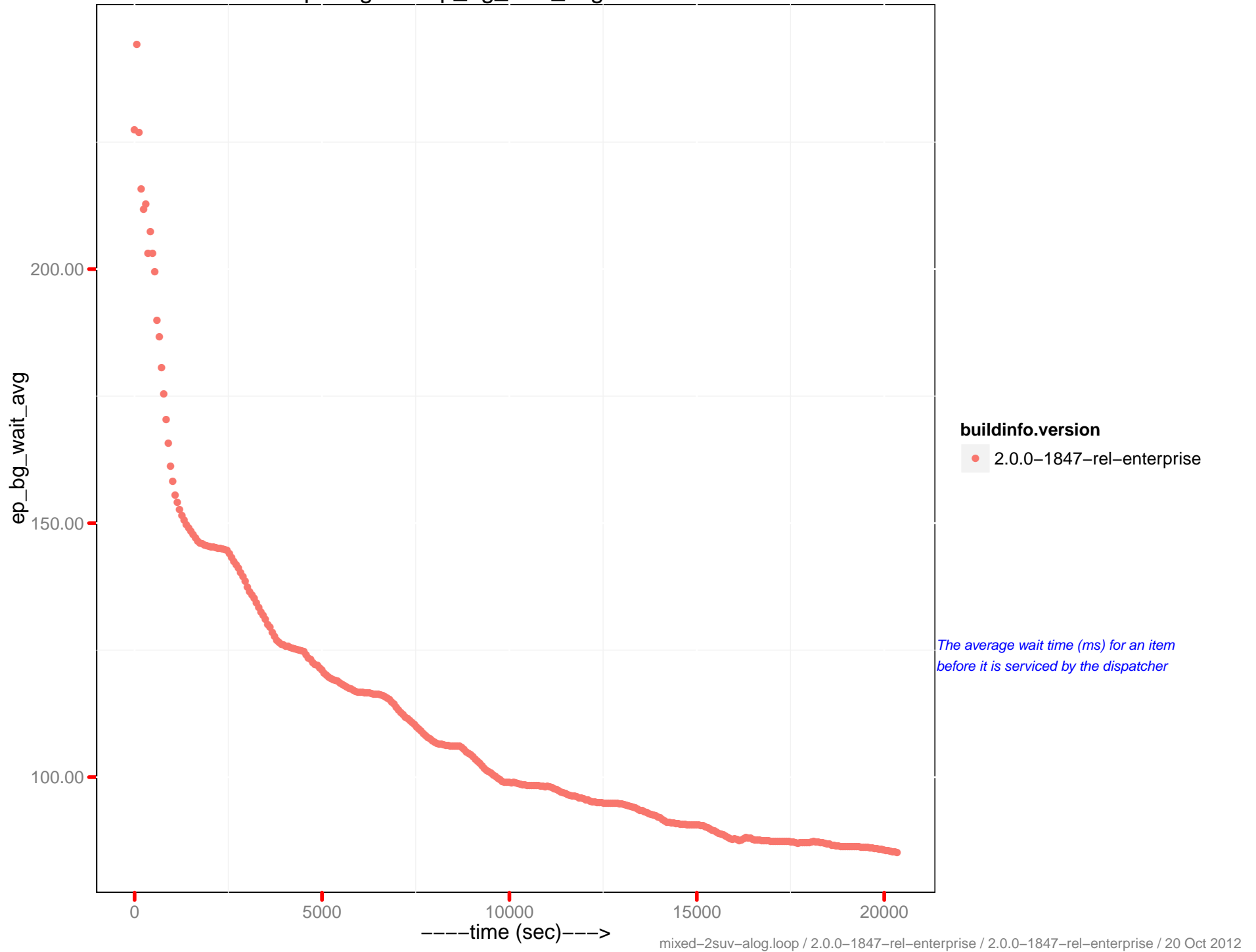




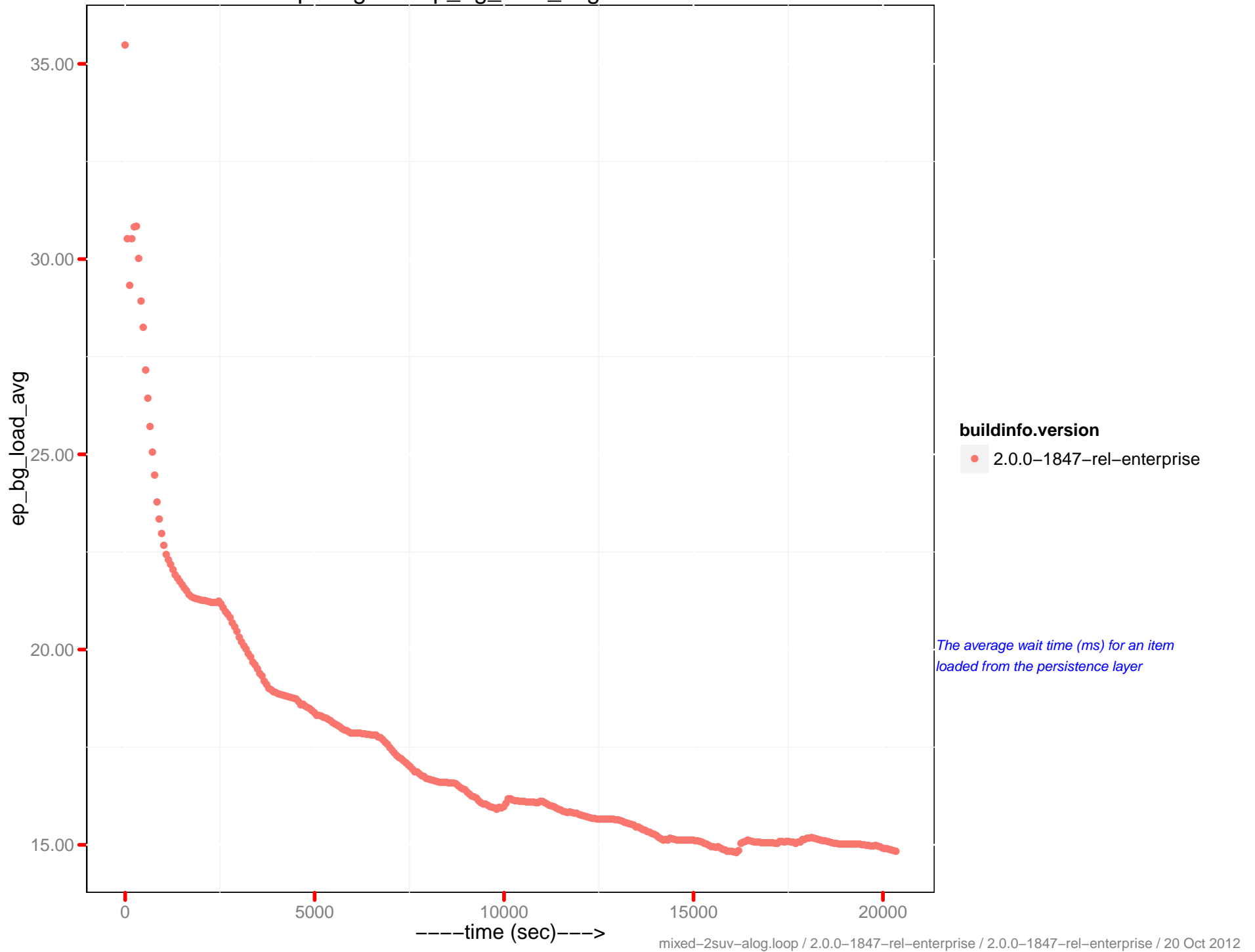
# ep-engine : ep\_bg\_load\_avg - 10.2.1.61



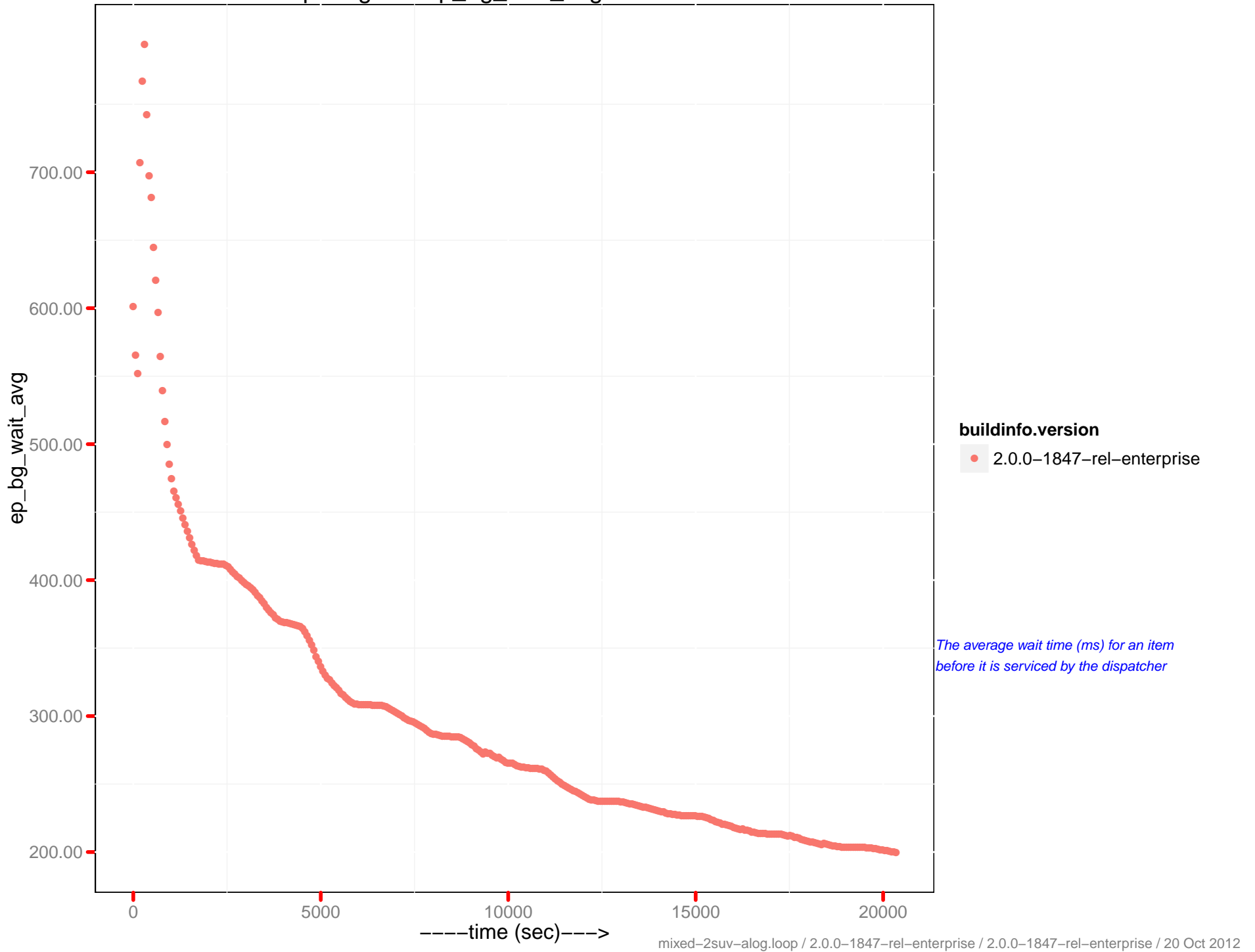
# ep-engine : ep\_bg\_wait\_avg - 10.2.1.63



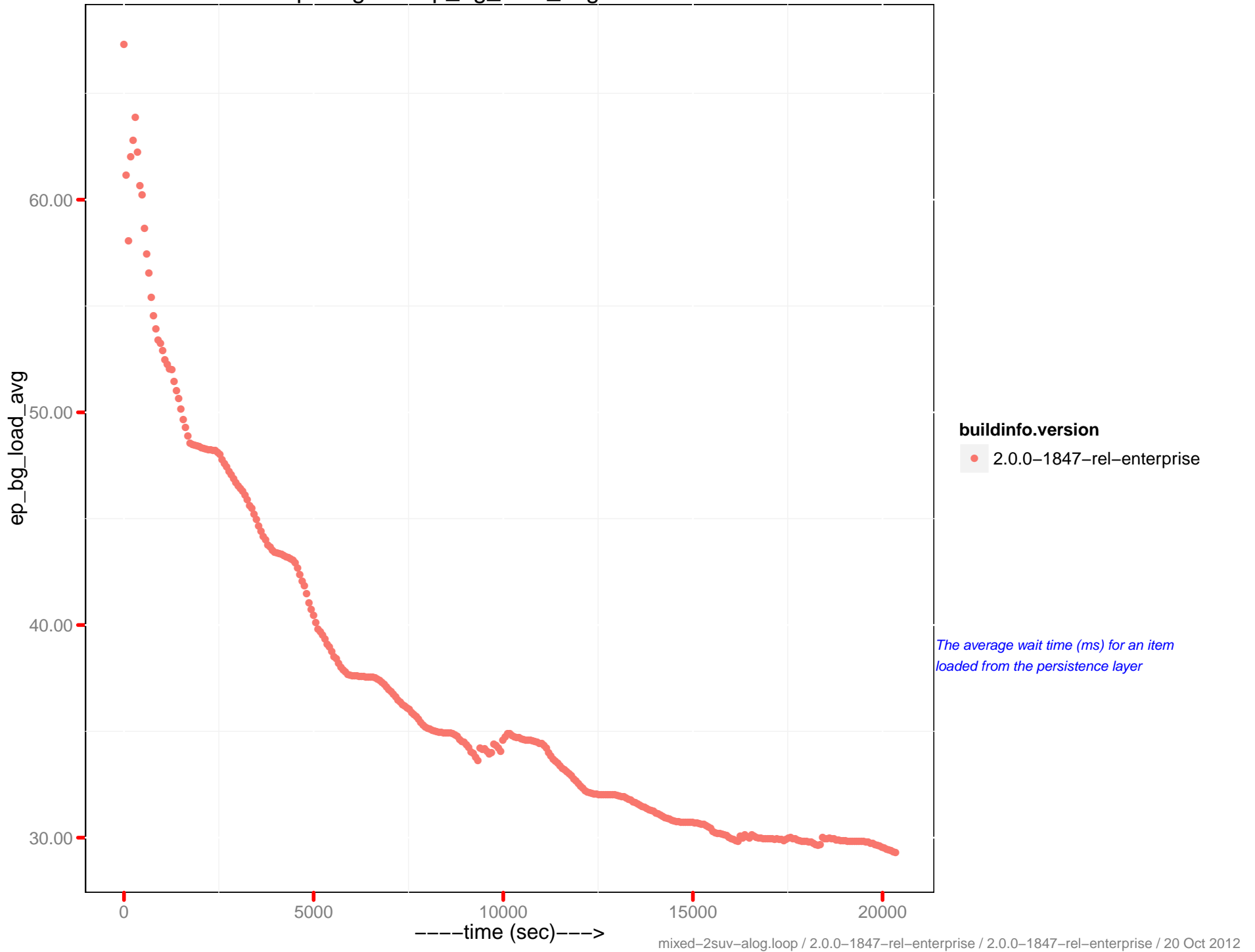
# ep-engine : ep\_bg\_load\_avg - 10.2.1.63



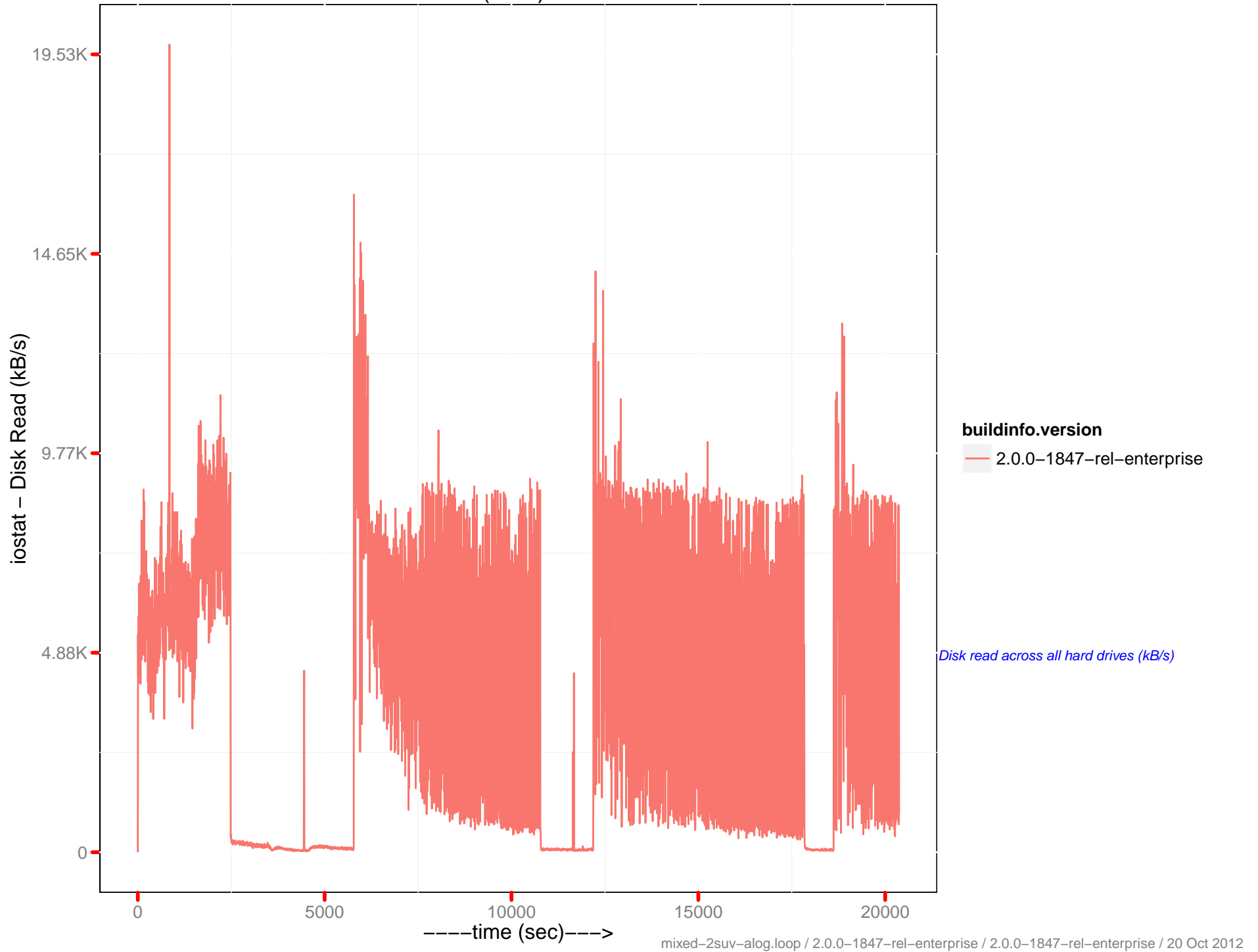
# ep-engine : ep\_bg\_wait\_avg - 10.2.1.64



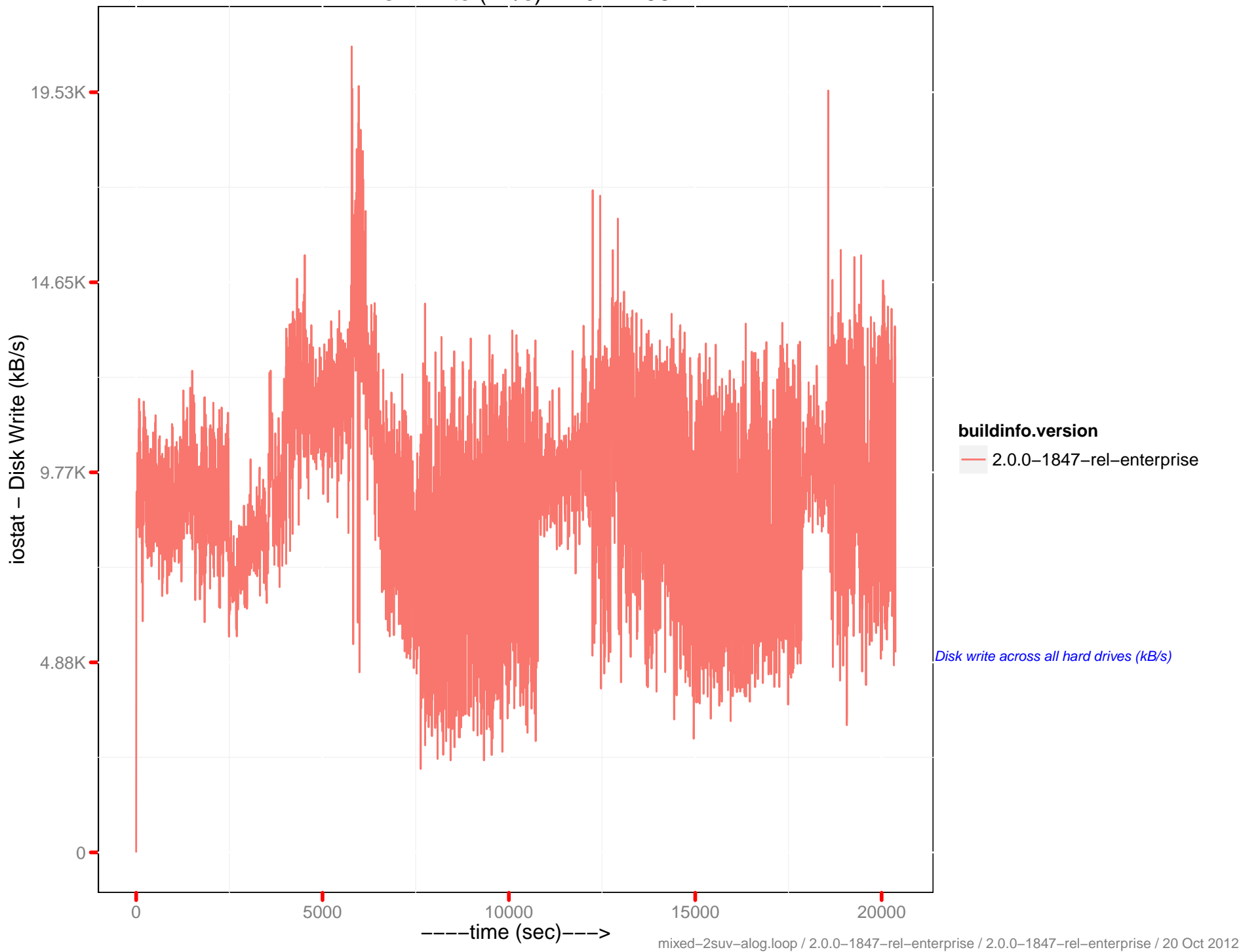
# ep-engine : ep\_bg\_load\_avg - 10.2.1.64



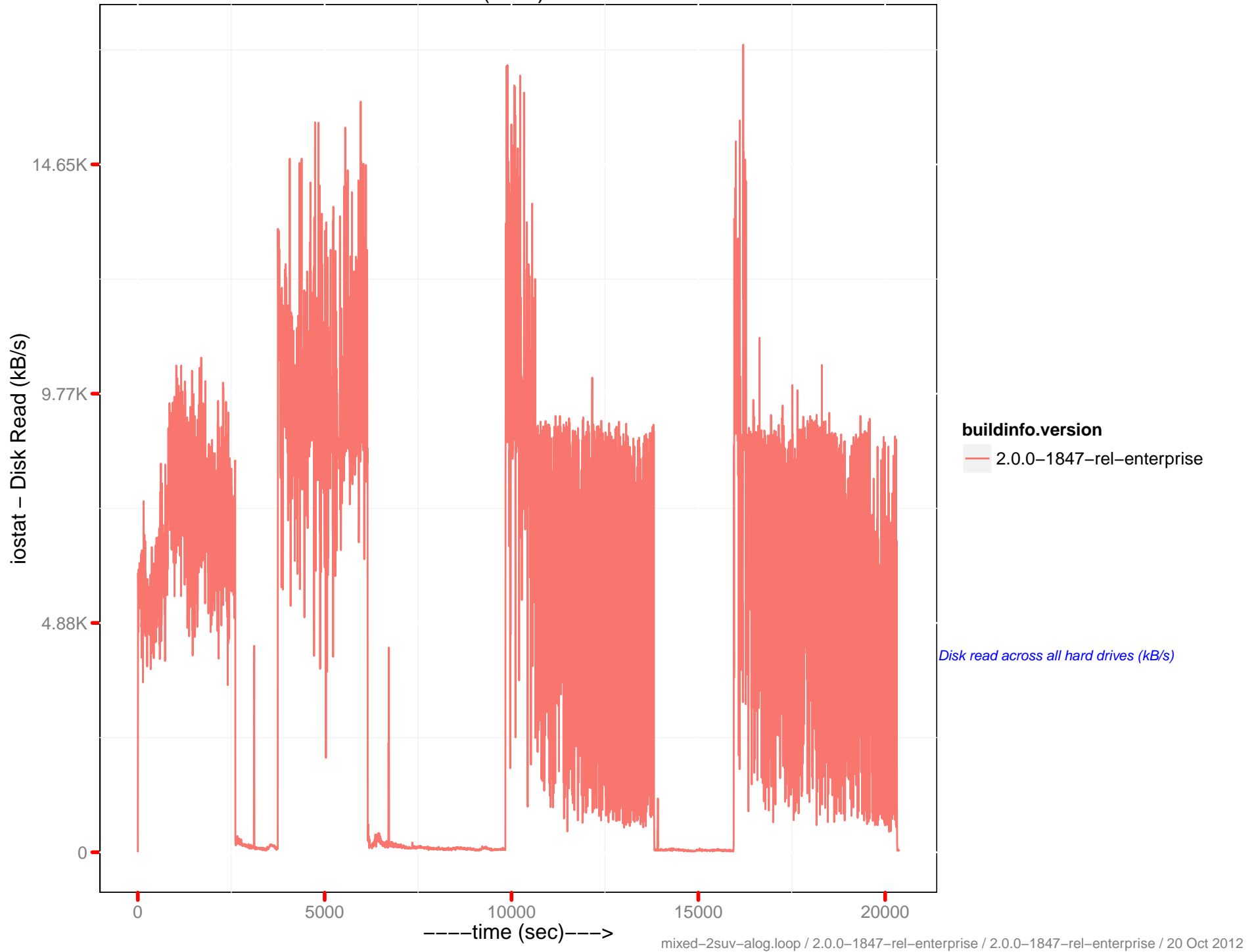
# Disk Read (kB/s) : 10.2.1.58



# Disk Write (kB/s) : 10.2.1.58

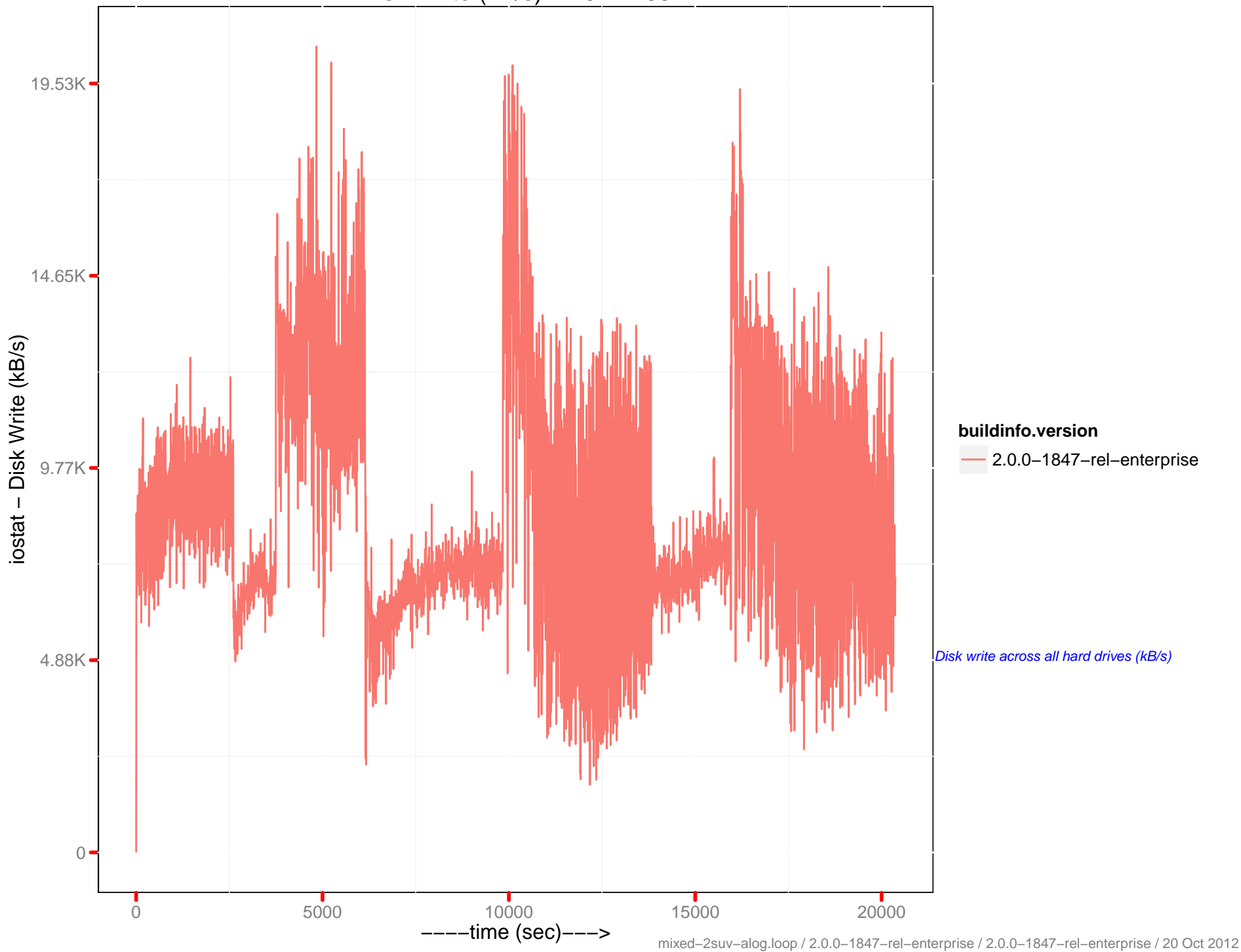


# Disk Read (kB/s) : 10.2.1.63

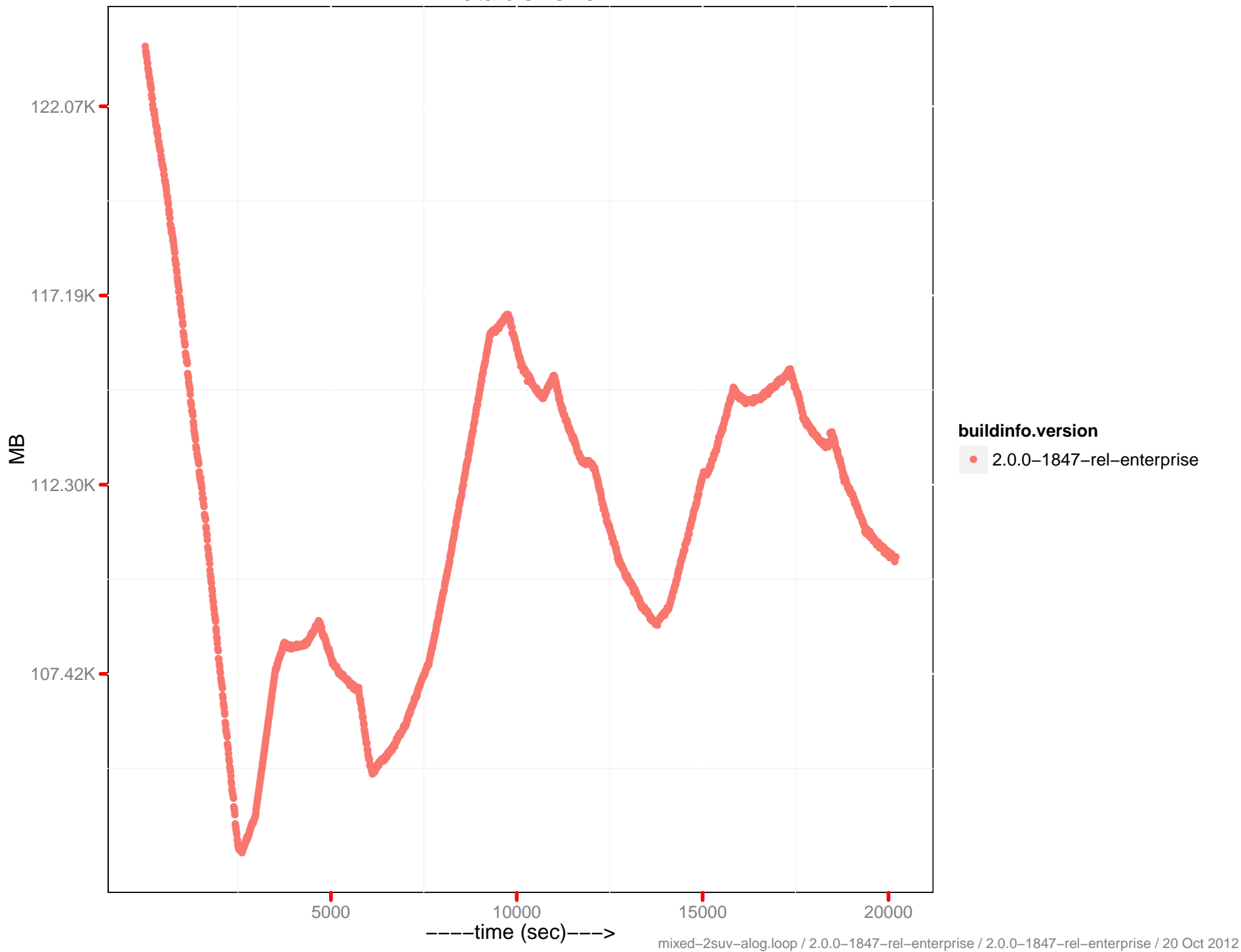




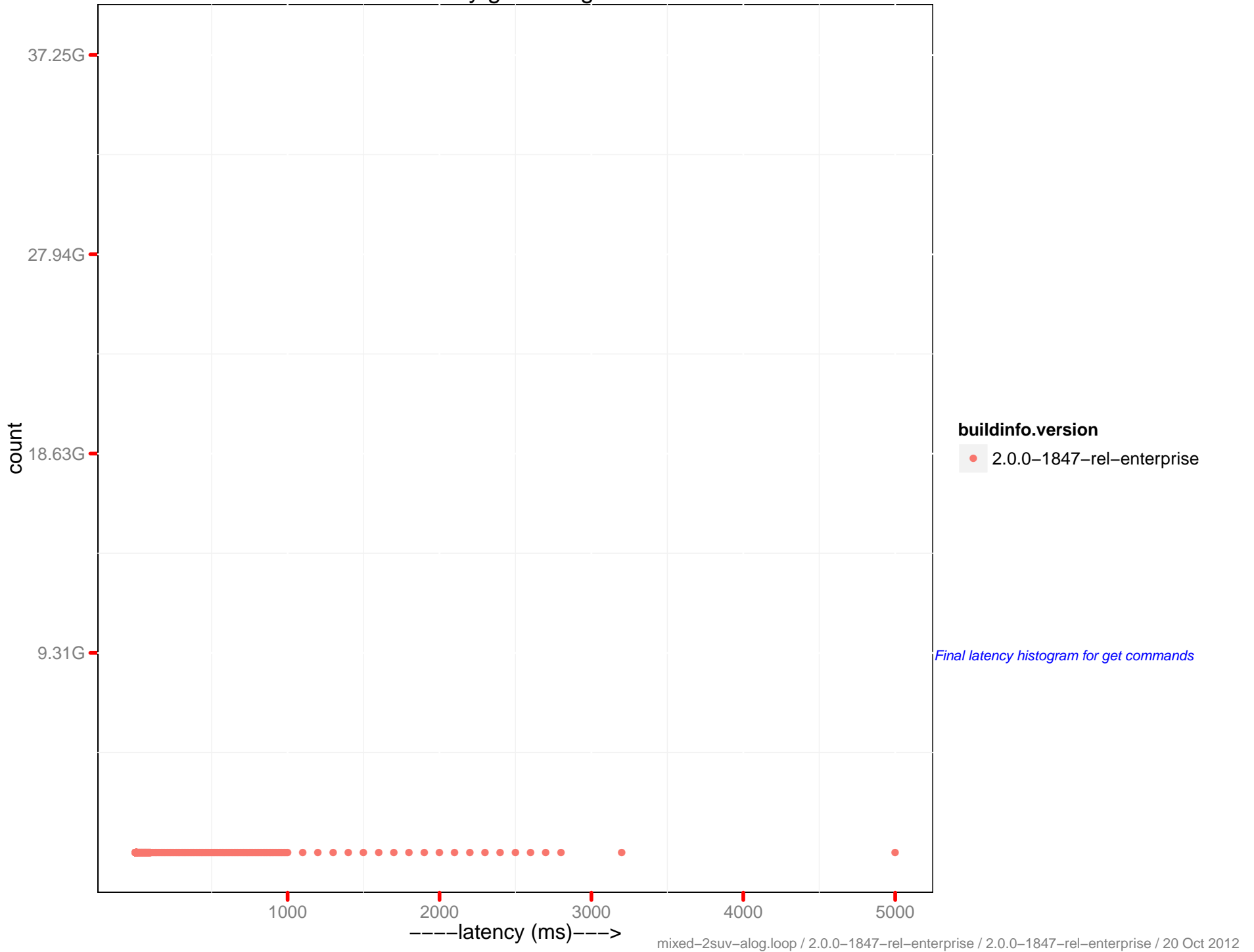
# Disk Write (kB/s) : 10.2.1.63



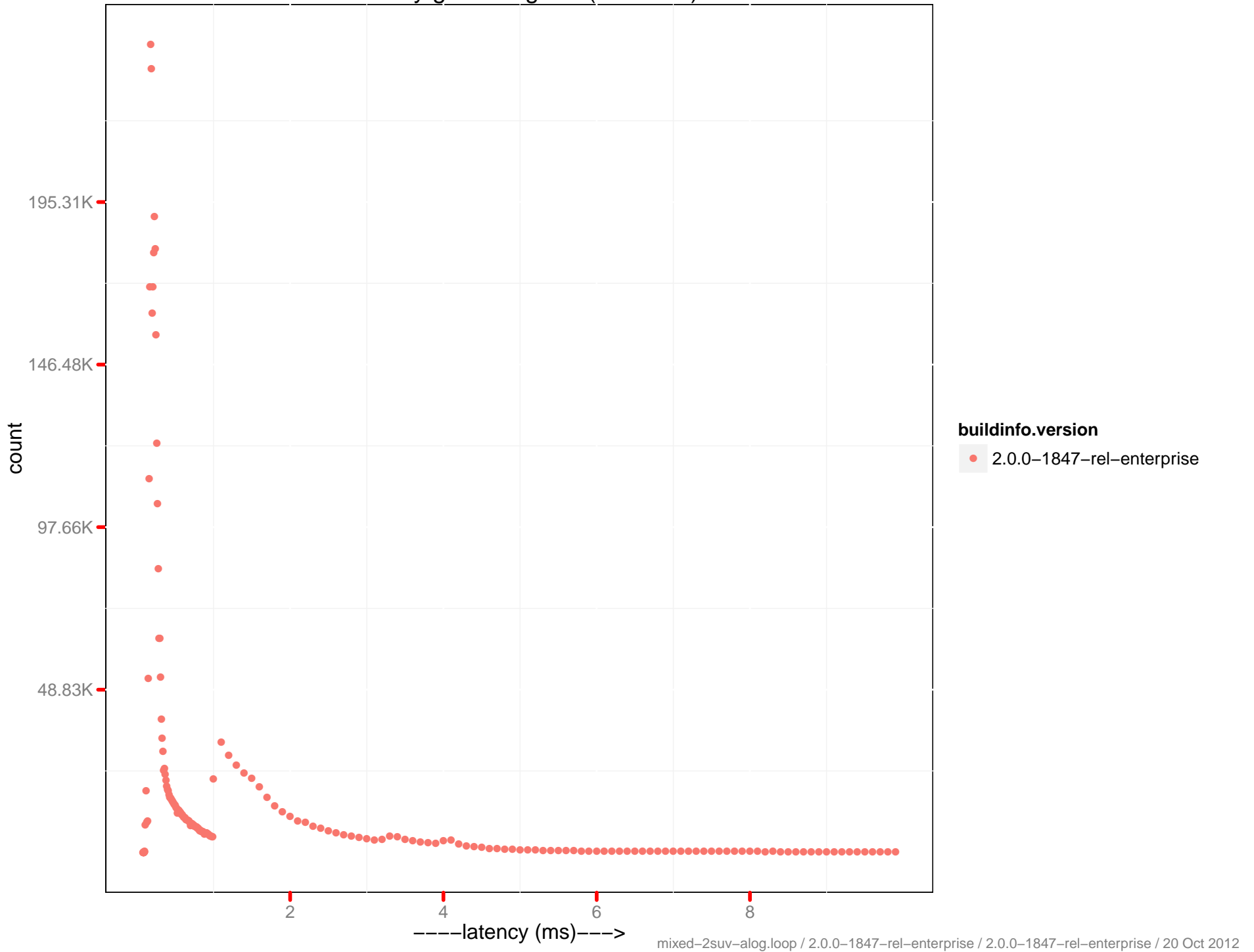
# Data disk size



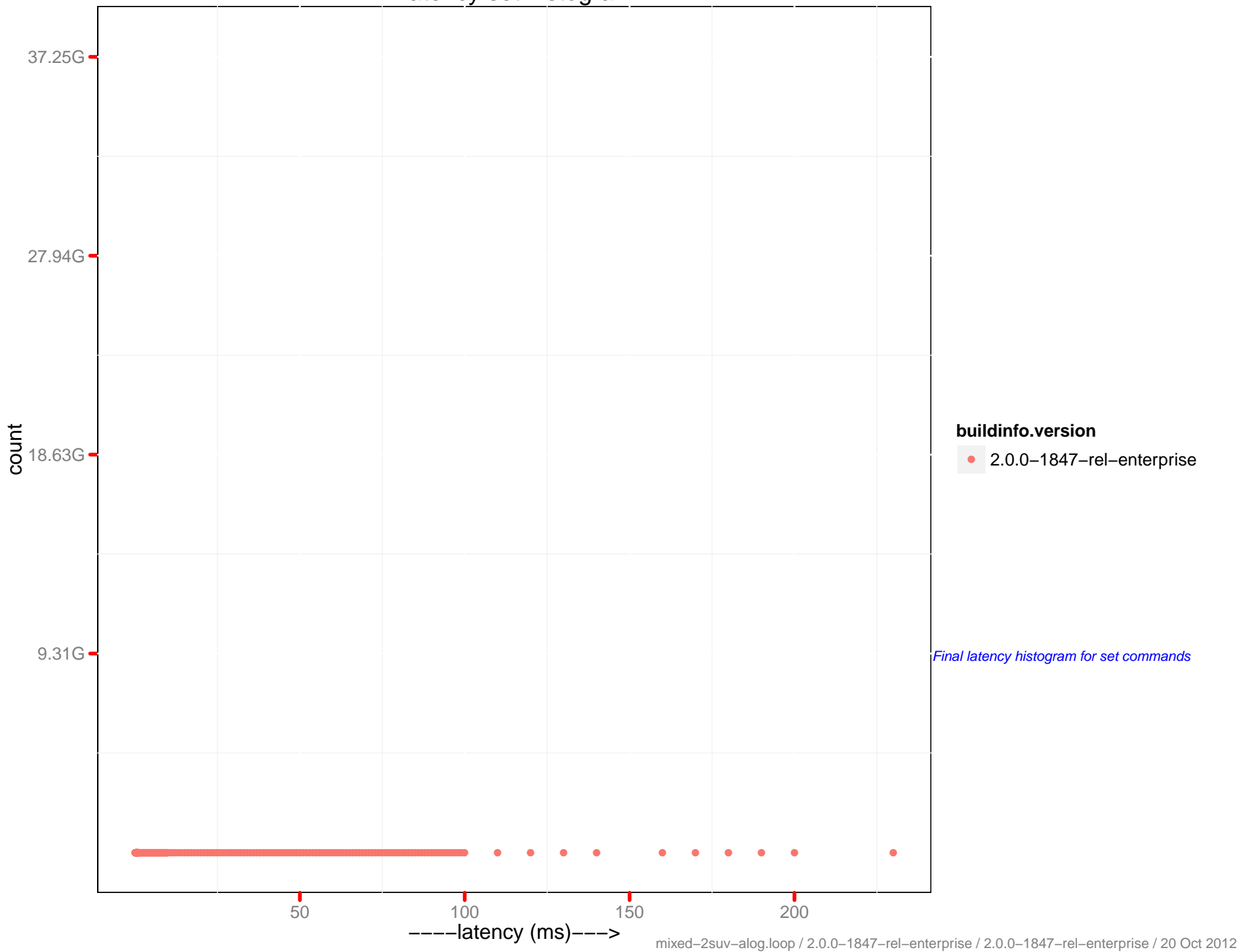
# Latency get histogram



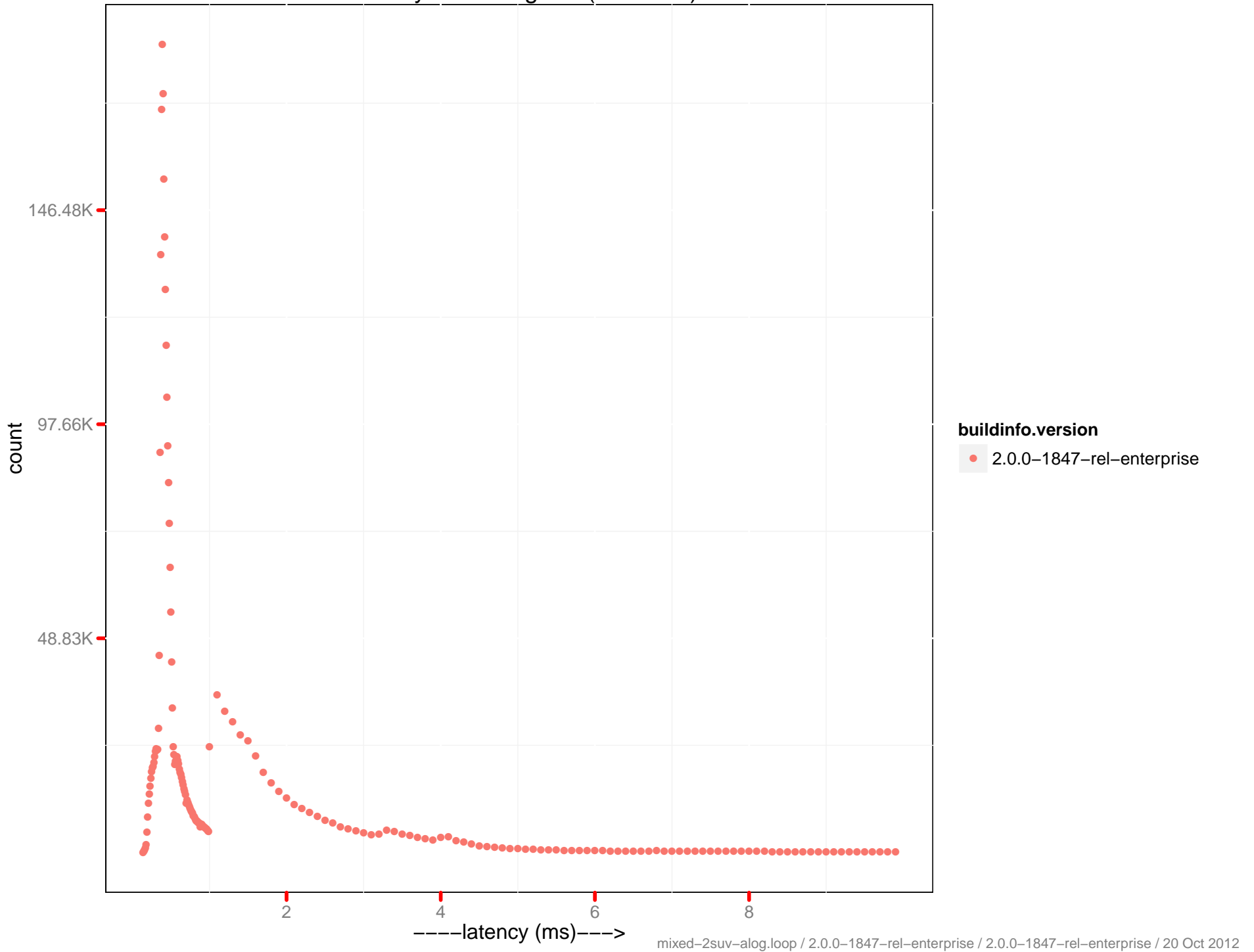
Latency get histogram (0–10 ms)



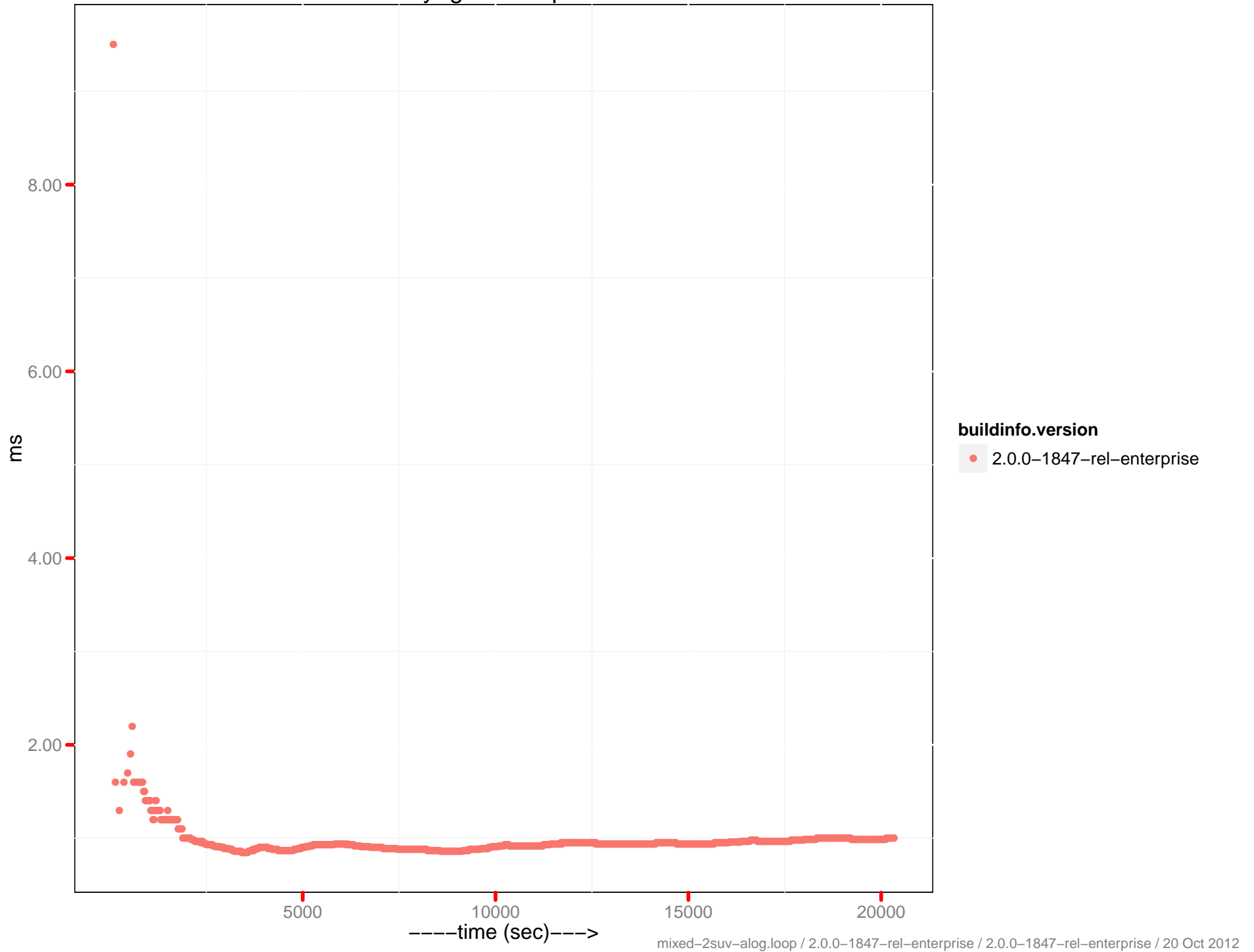
# Latency set histogram



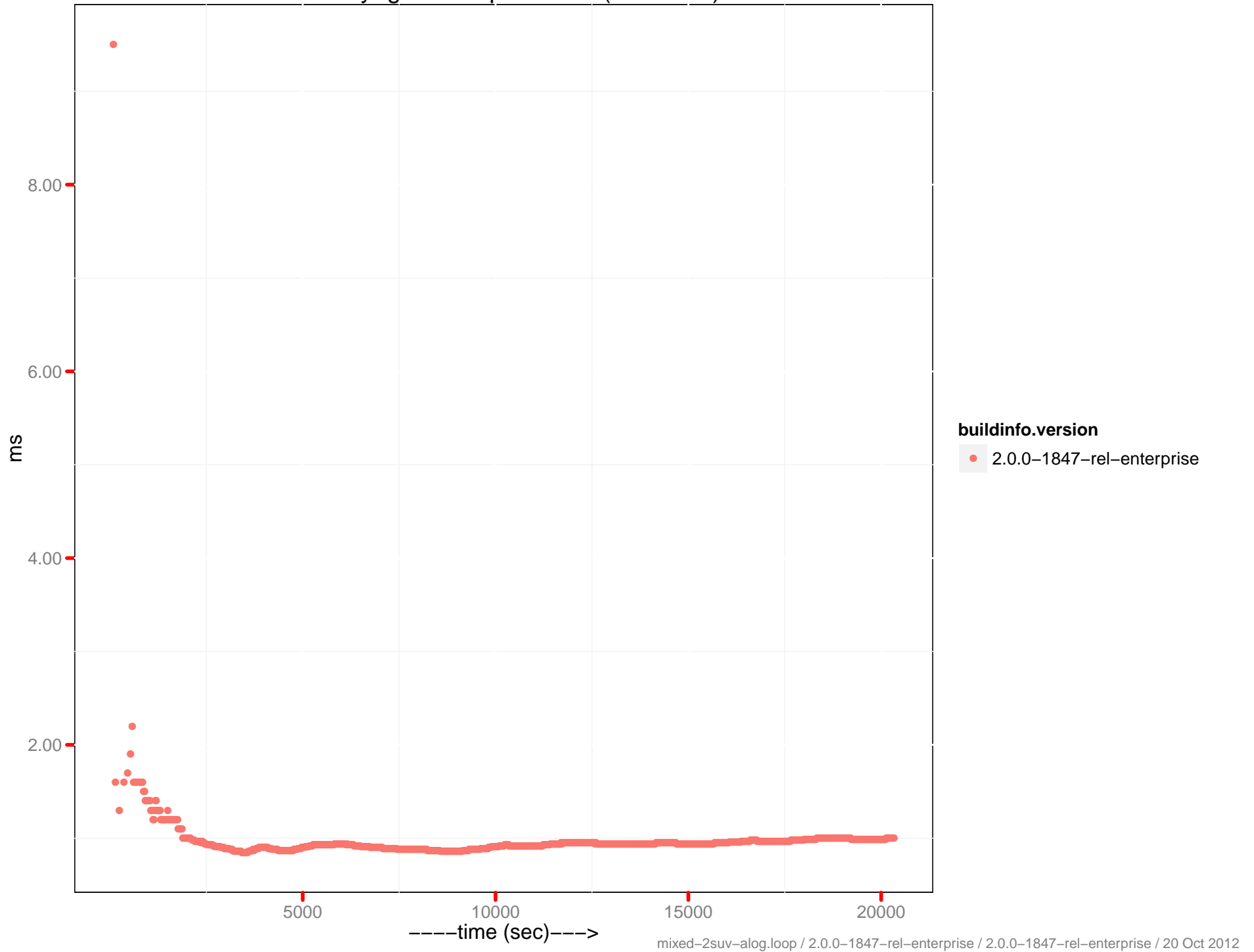
Latency set histogram (0–10 ms)



# Latency-get 90th percentile

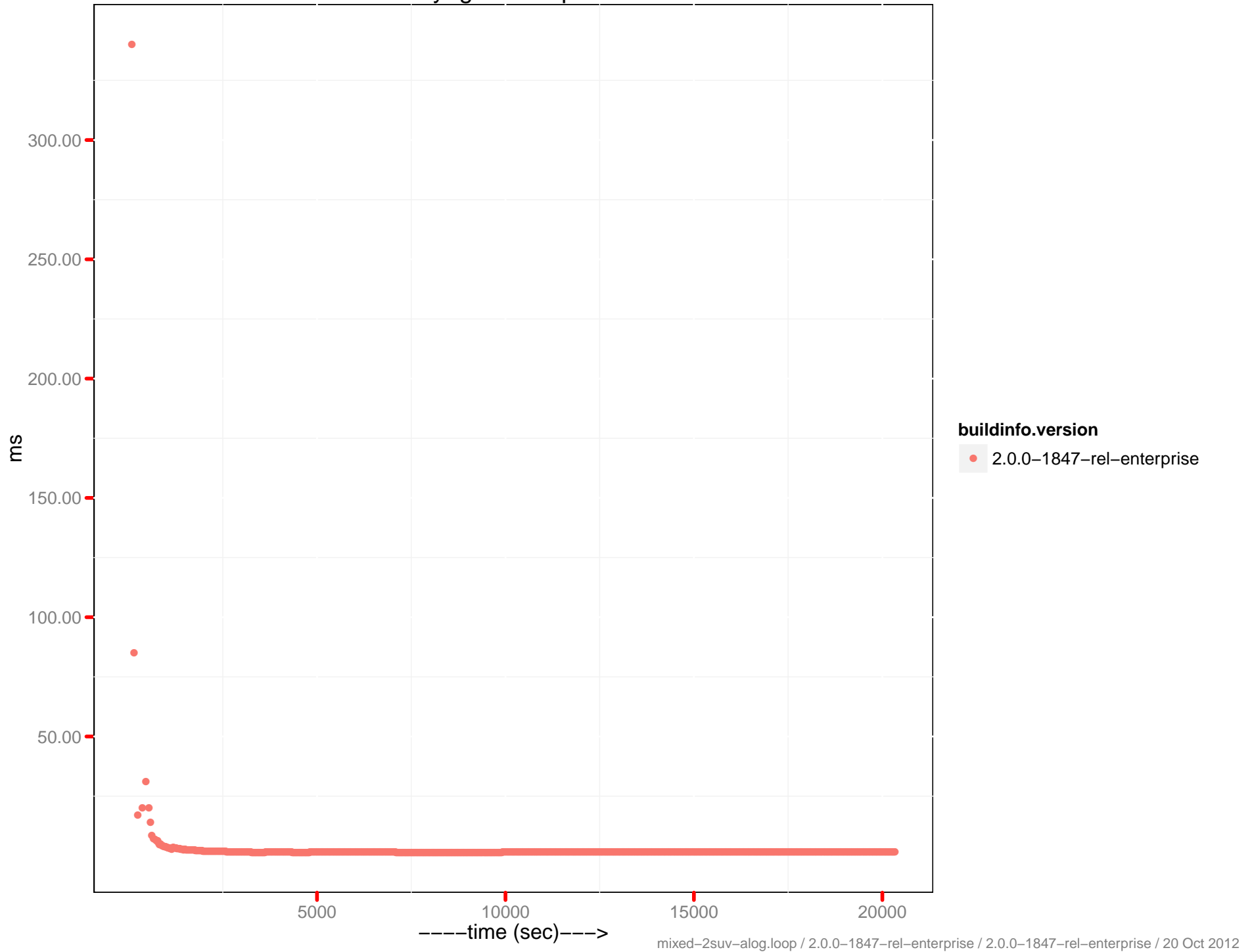


Latency-get 90th percentile (0 - 10ms)

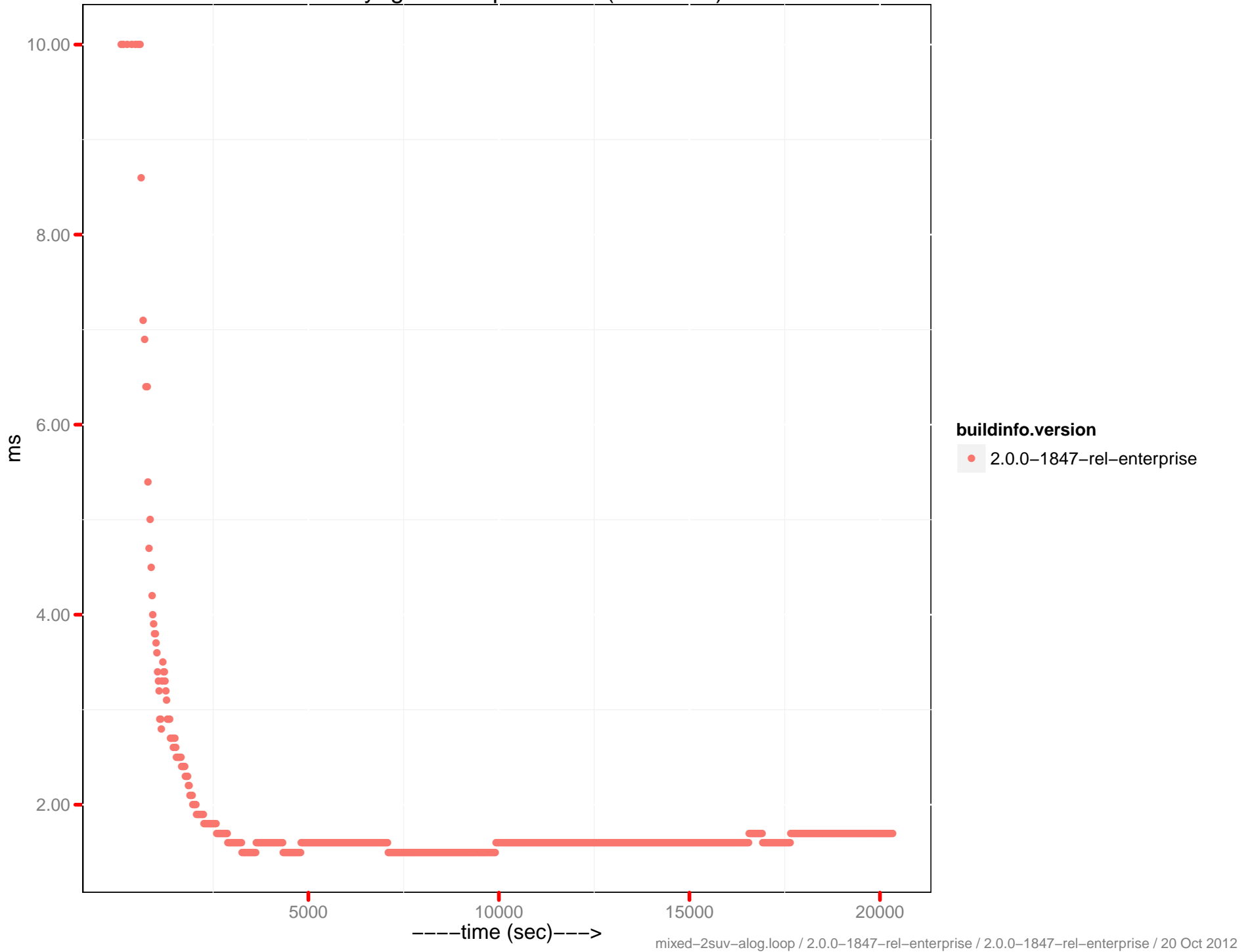




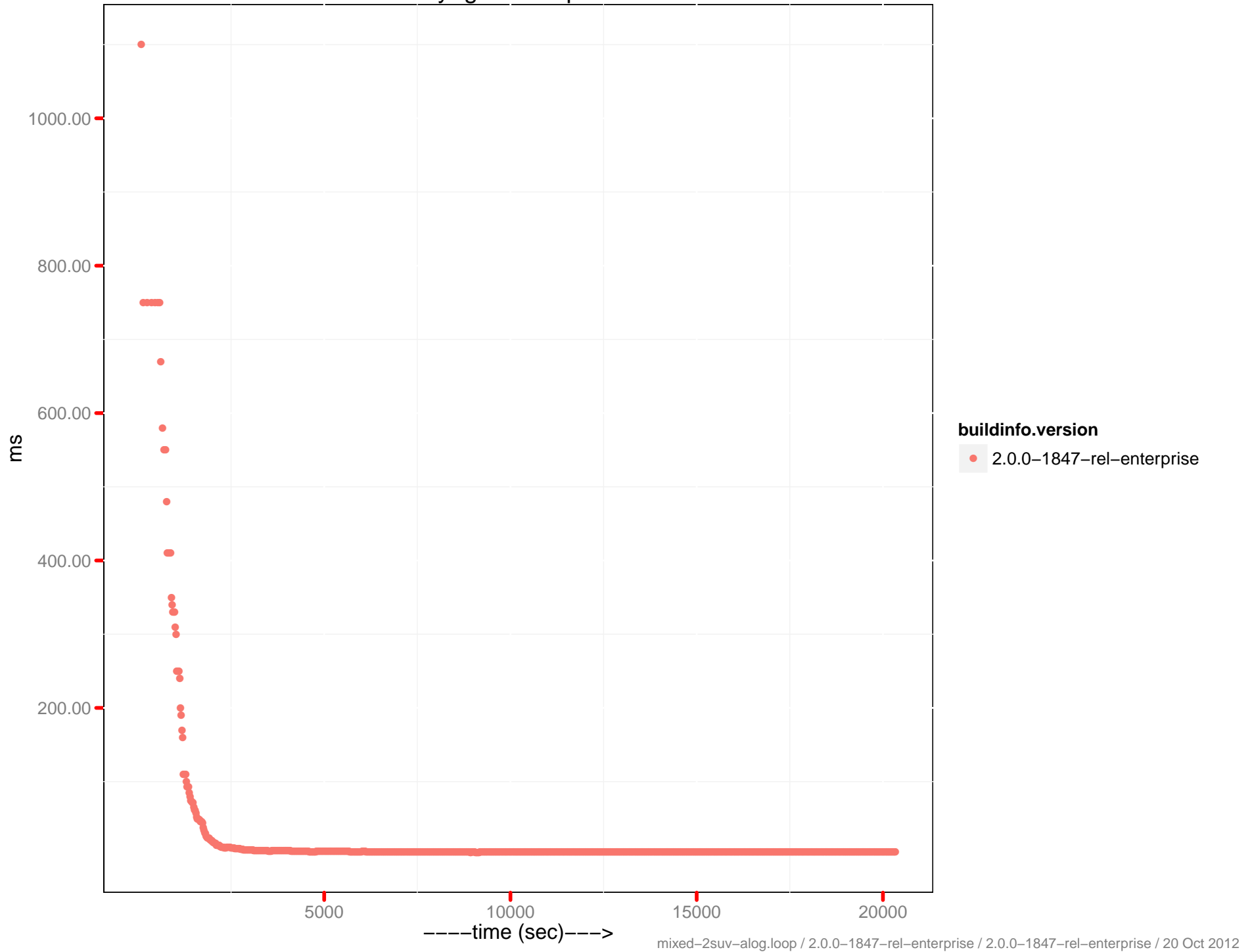
Latency-get 95th percentile



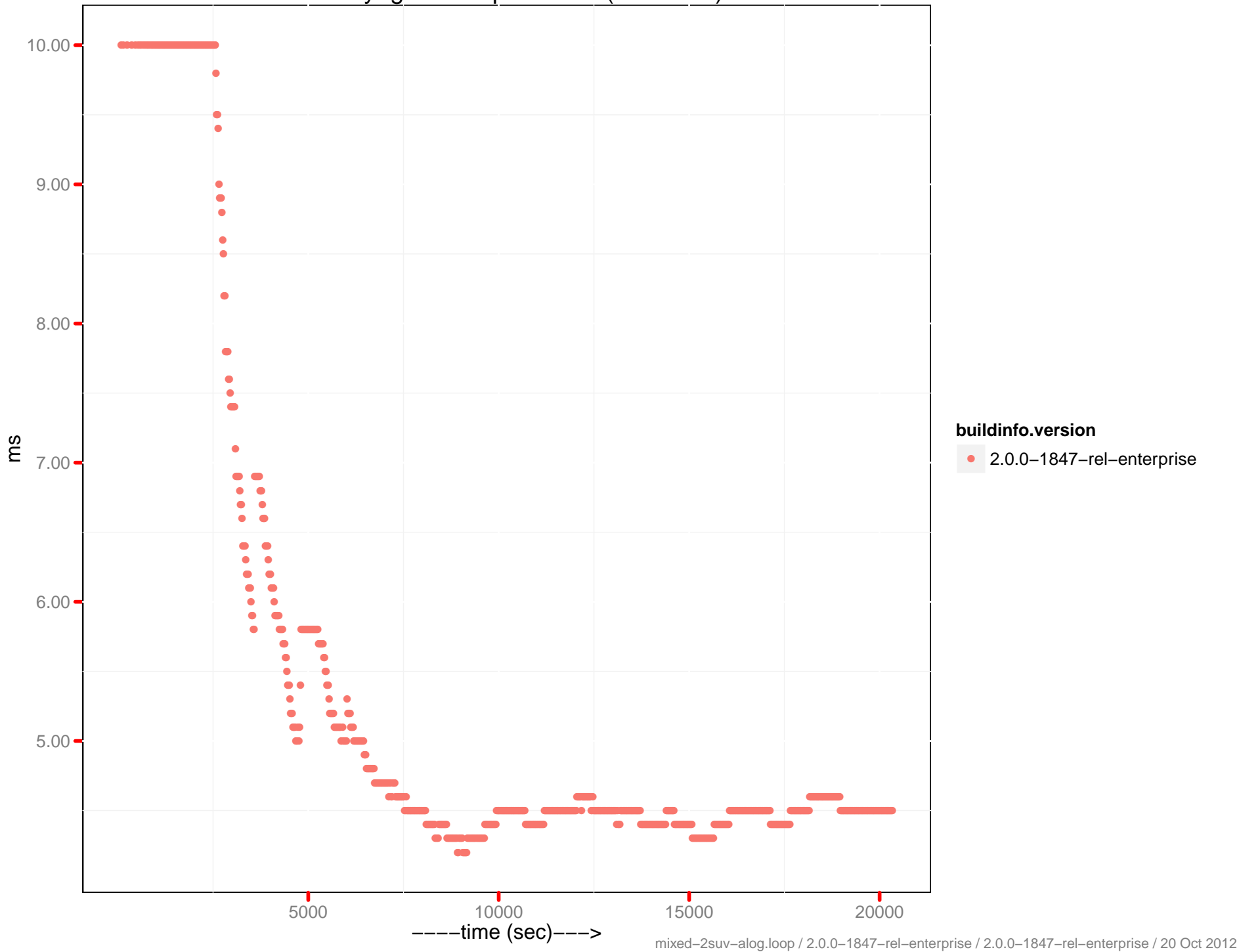
Latency-get 95th percentile (0 - 10ms)



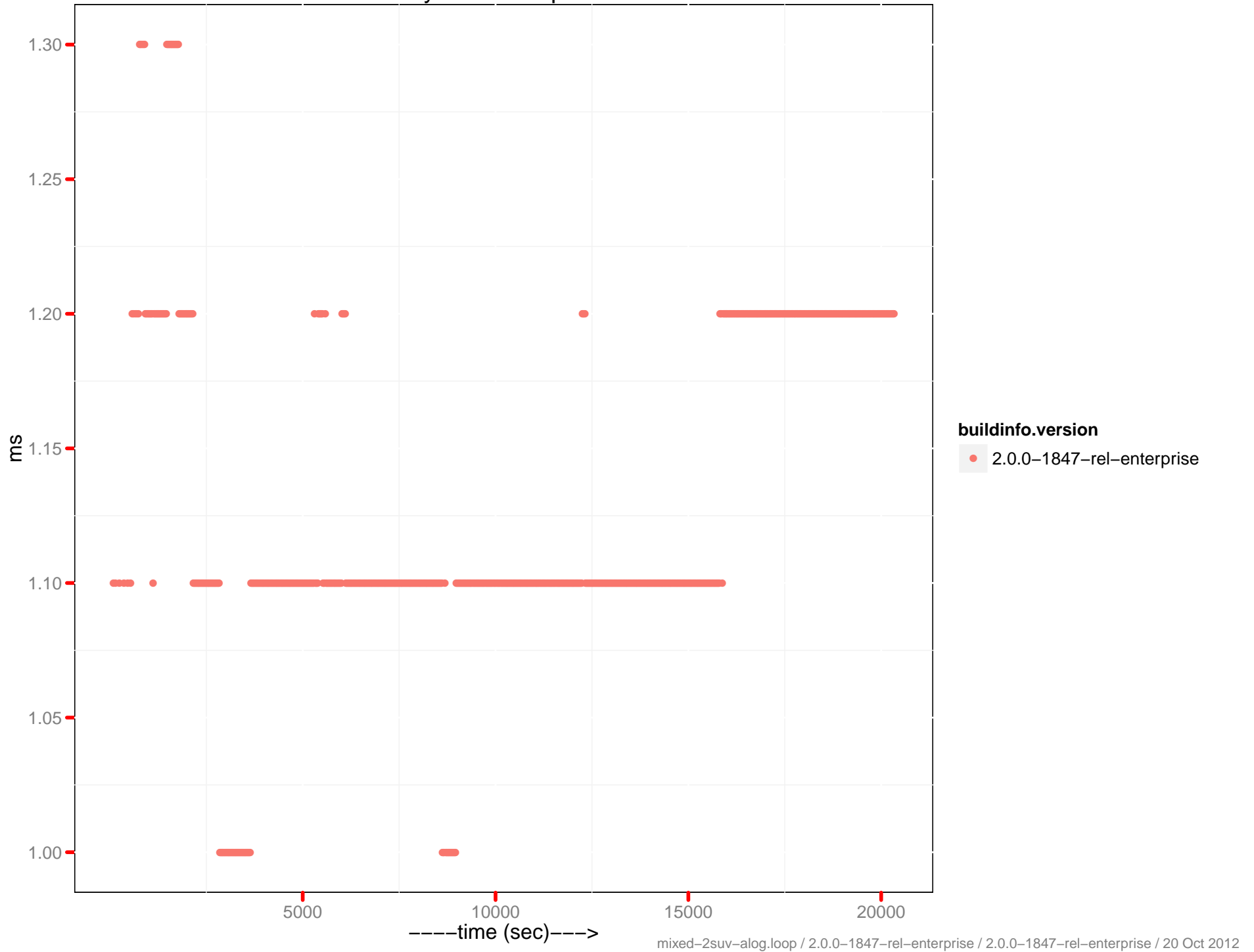
# Latency-get 99th percentile



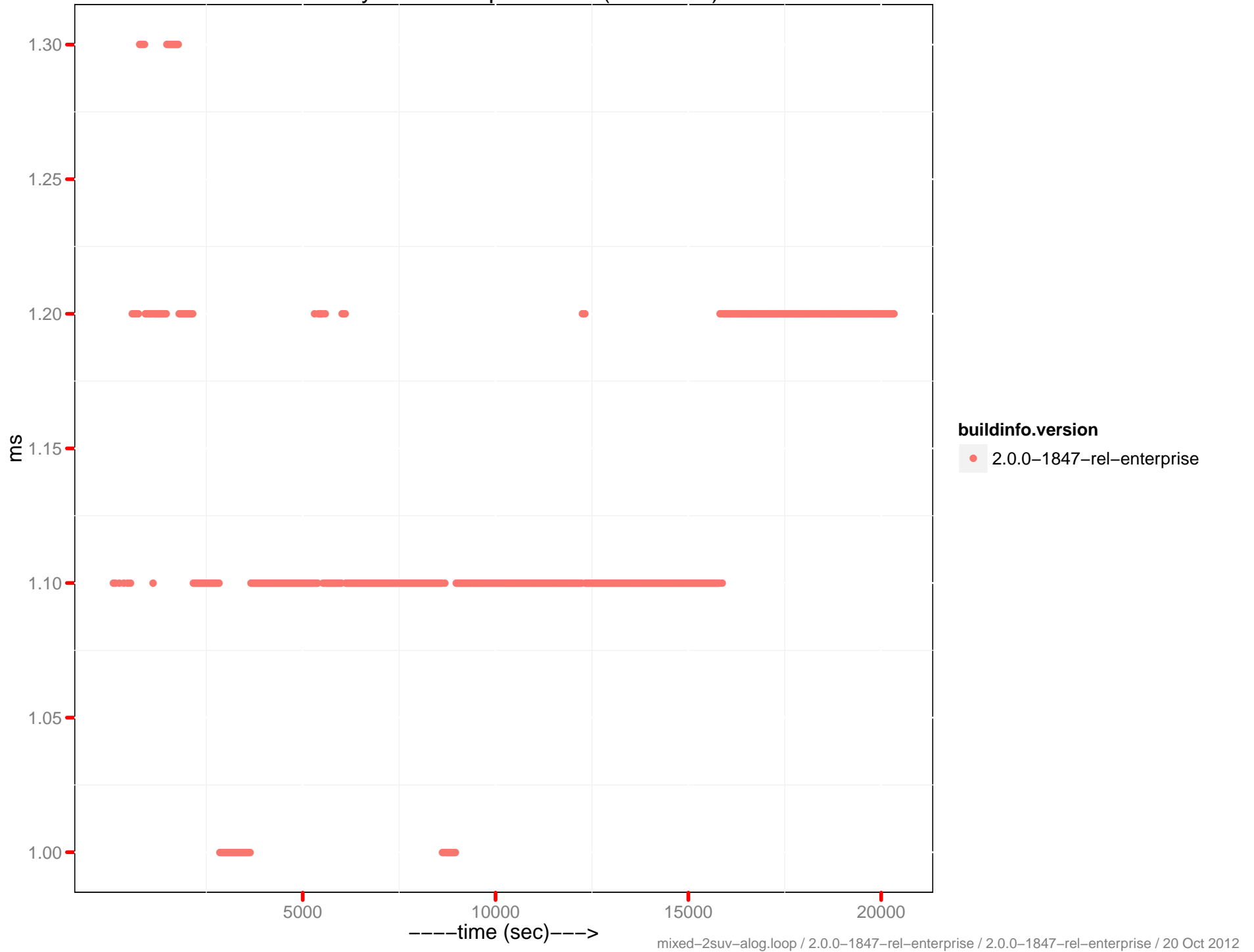
Latency-get 99th percentile (0 - 10ms)



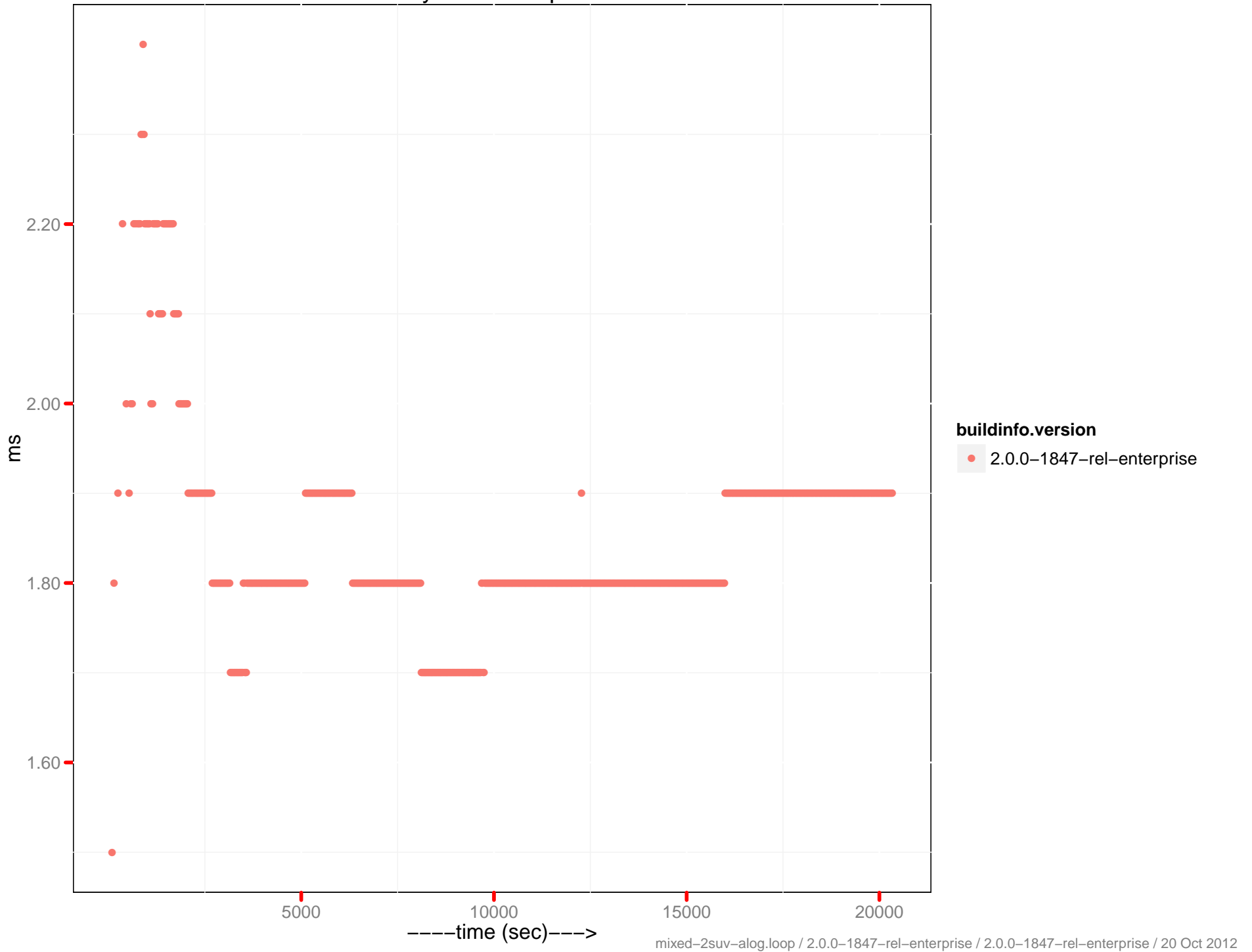
# Latency-set 90th percentile



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



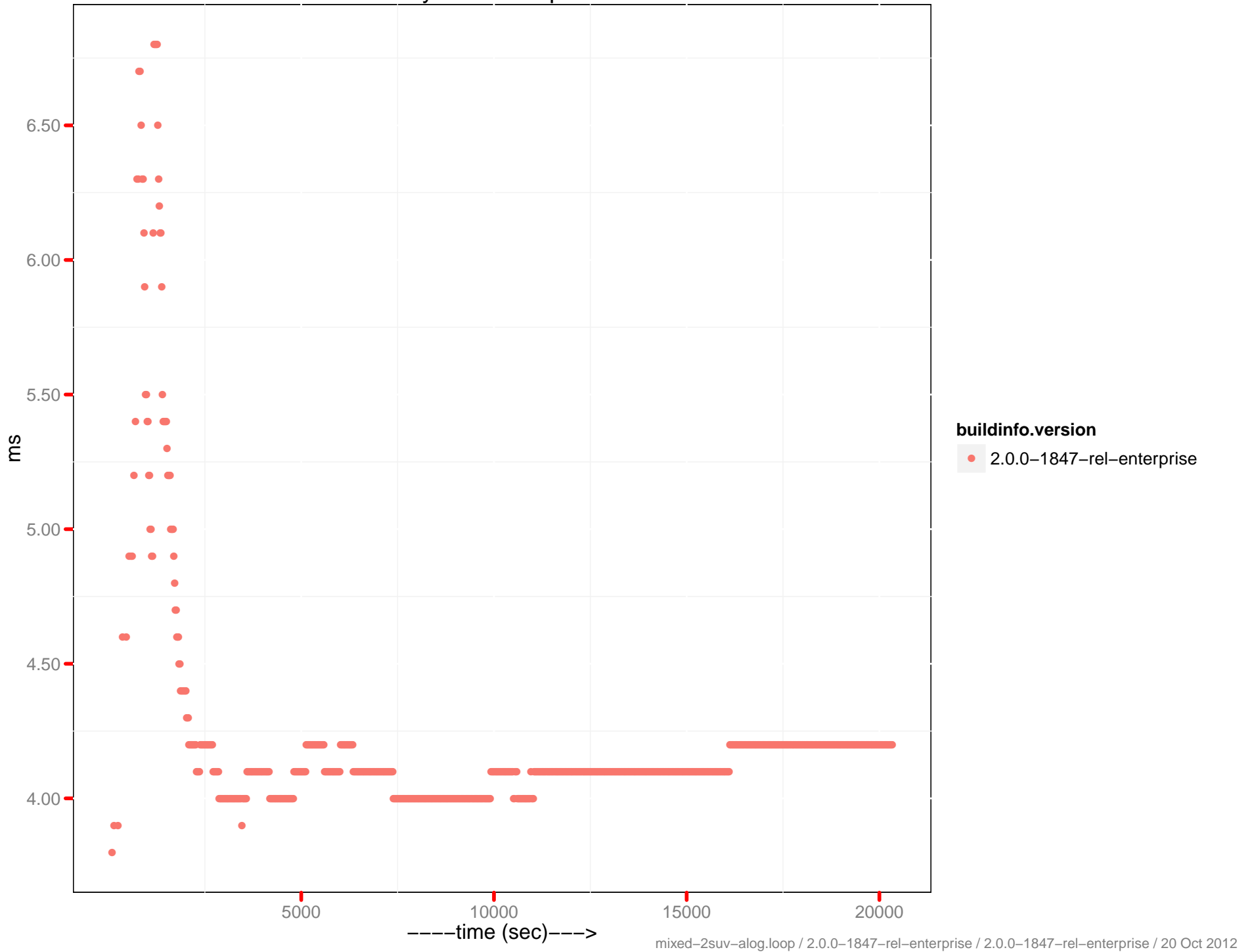
# Latency-set 95th percentile



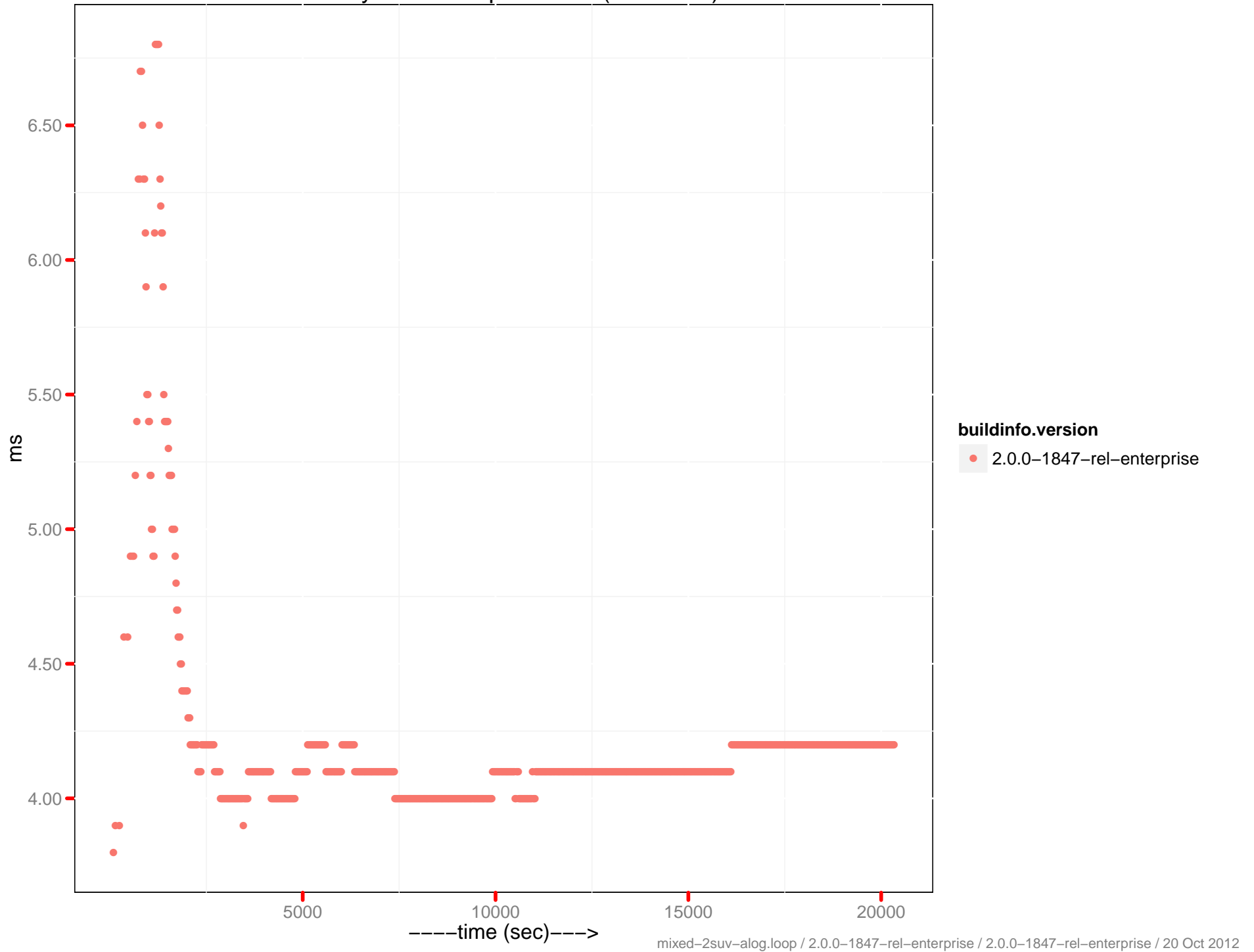




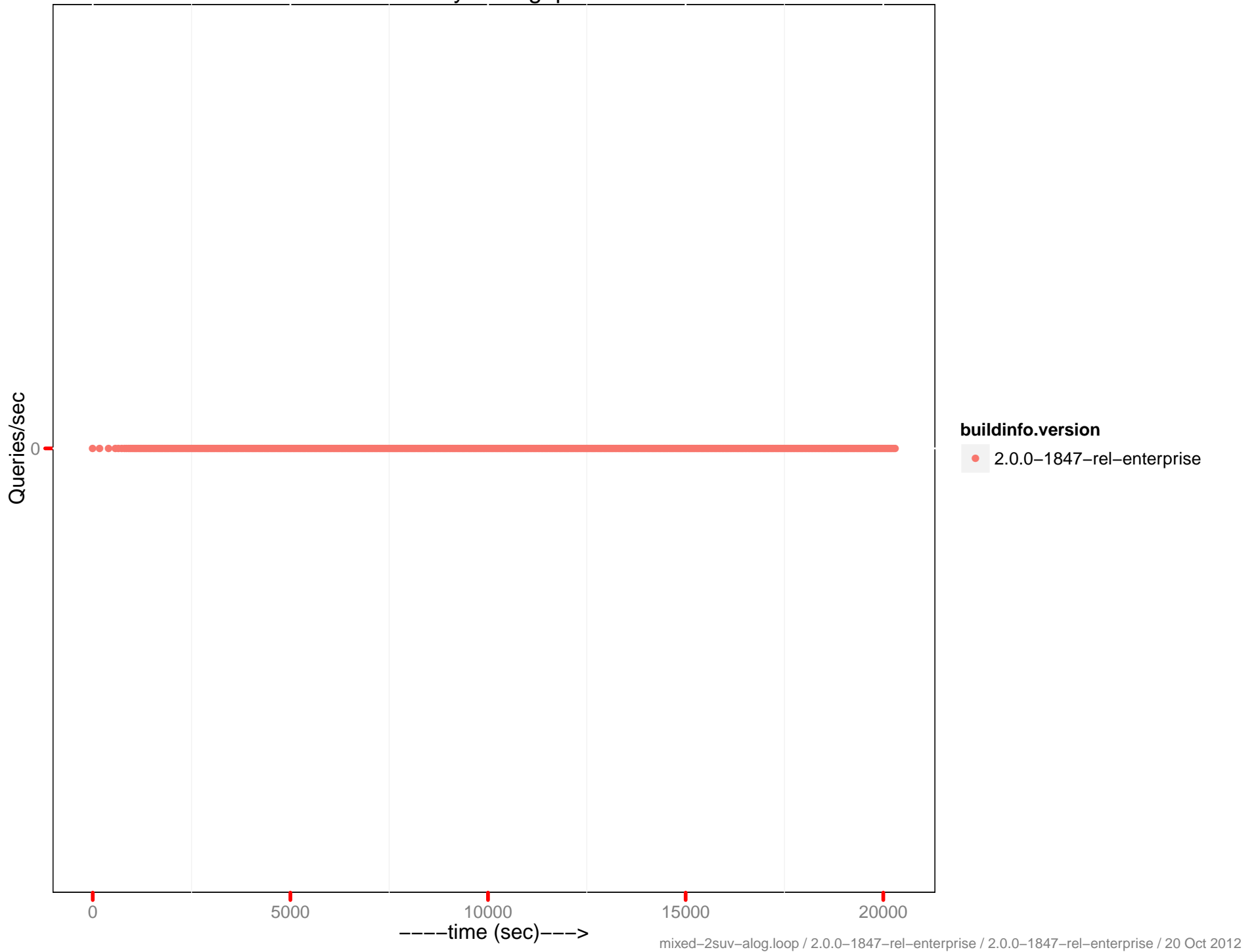
Latency-set 99th percentile



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



# Query throughput



```
mixed-2suv-alog.conf
# mixed suv 20M load, 2M hot reload, 12M access creates
# speed limit = 0.5k per client
# num clients = 30
# DGM
#
performance.eperf.EPerfClient.test_eperf_mixed

params:

# general
batch=50
kind=nonjson
mem_quota=20000

# cbstats collector
cb_stats=1

# load phase
hot_init_items=2000000
alog_hot_load=1
items=20000000

# 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=12000000

# control (defaults: pytests/performance/perf_defaults.py)
mcsoda_heartbeat=3
mcsoda_max_ops_sec=500
tear_down=1
tear_down_proxy=1
tear_down_bucket=0
tear_down_cluster=1
tear_down_on_setup=0
```

```
terra.ini
[global]
username:root
password:couchbase
port:8091
data_path:/data2
index_path:/data

[servers]
1:10.2.1.61
2:10.2.1.58
3:10.2.1.63
4:10.2.1.64

[clients]
1:10.2.1.59

[membase]
rest_username:Administrator
rest_password:password

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