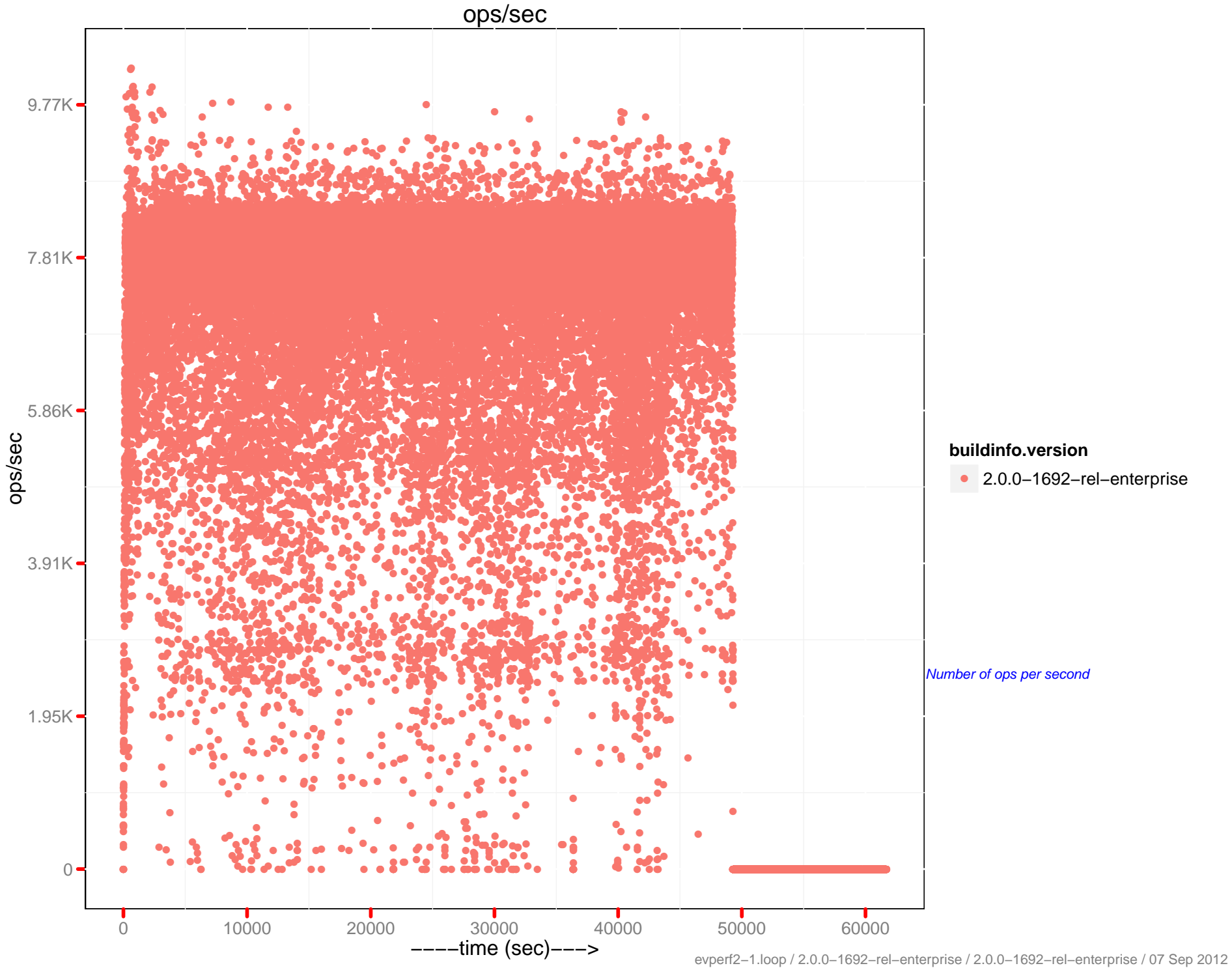
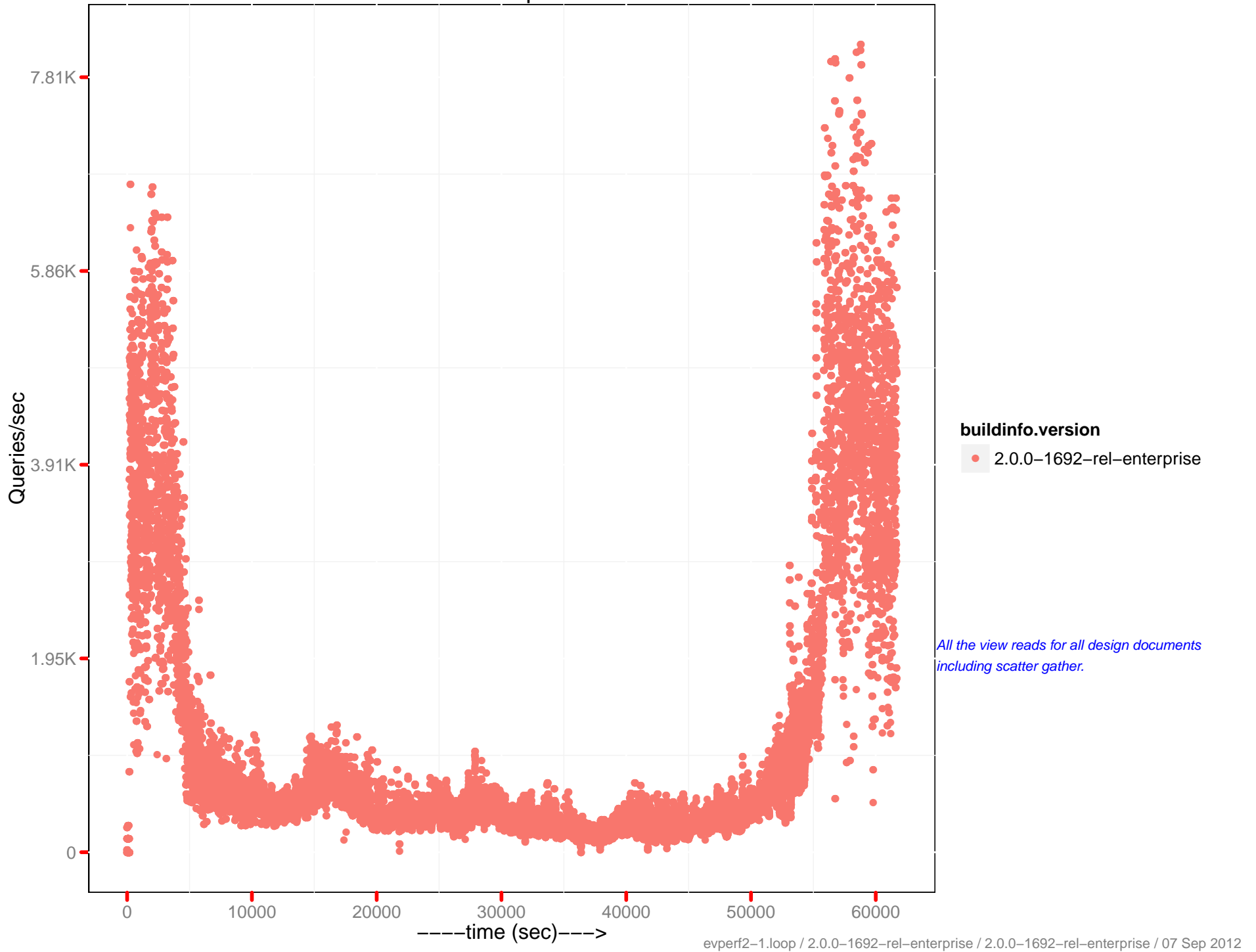


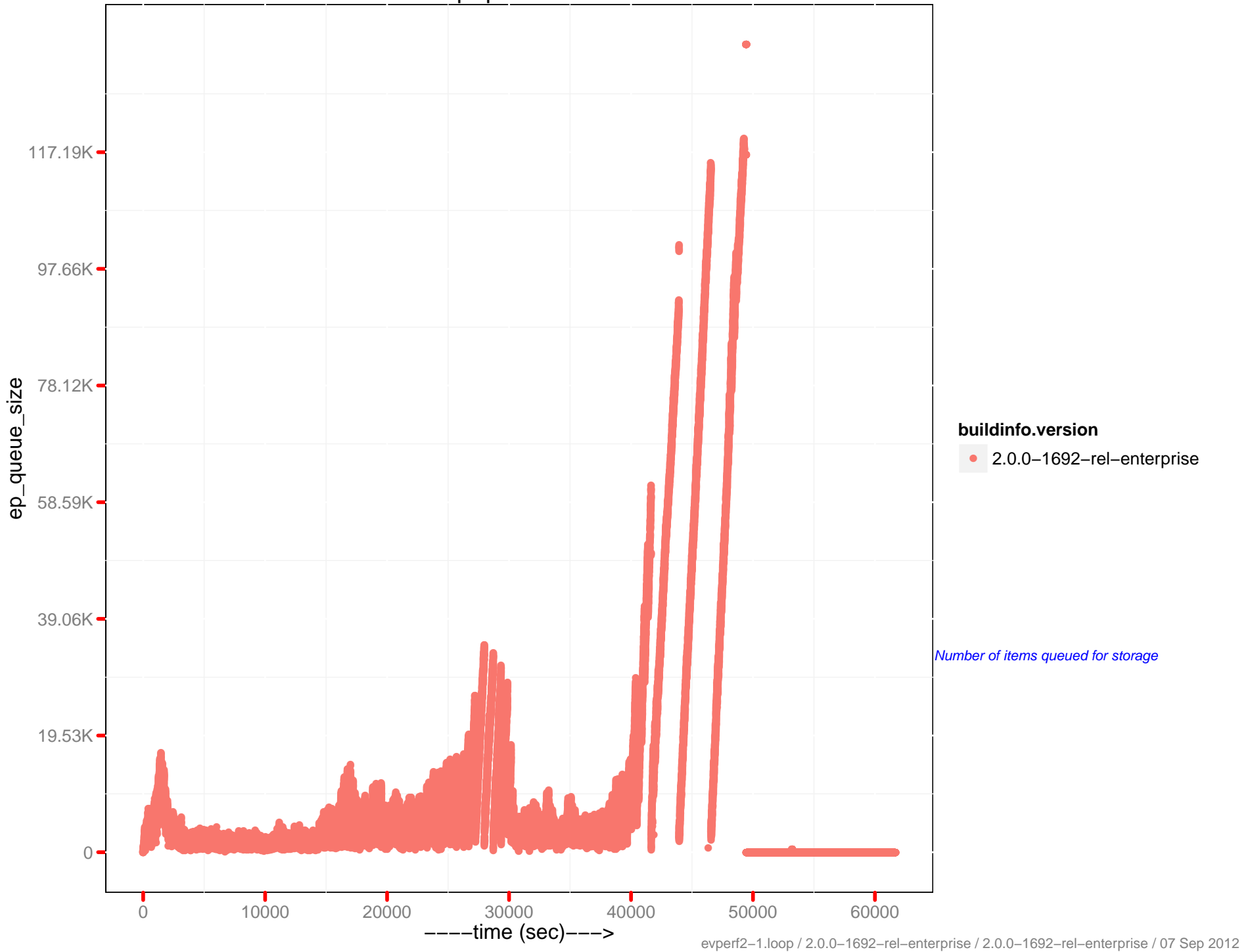
	2.0.0 – 1692	2.0.0 – 1692
<i>Runtime (in hr)</i>	17.14	NA
<i>Avg. Drain Rate</i>	264.12	NANA
<i>Peak Disk (GB)</i>	225.28	NA
<i>Peak Memory (GB)</i>	20.16	NA
<i>Avg. OPS</i>	7.43K	NANA
<i>Avg. mem memcached (GB)</i>	19.69	NA
<i>Avg. mem beam.smp (MB)</i>	354.36	NA
<i>Avg. CPU rate (%)</i>	26.3	NA
<i>Latency-get (90th) (ms)</i>	1.29	NA
<i>Latency-get (95th) (ms)</i>	2.38	NA
<i>Latency-get (99th) (ms)</i>	7.68	NA
<i>Latency-set (90th) (ms)</i>	1.39	NA
<i>Latency-set (95th) (ms)</i>	2.41	NA
<i>Latency-set (99th) (ms)</i>	5.38	NA
<i>Latency-query (80th) (ms)</i>	183.09	NA
<i>Latency-query (90th) (ms)</i>	334.41	NA
<i>Latency-query (95th) (ms)</i>	478.58	NA
<i>Latency-query (99th) (ms)</i>	1059.87	NA
<i>Latency-query (99.9th) (ms)</i>	2101.25	NA
<i>Avg. QPS</i>	185.8	NA
<i>XDC ops/sec</i>	NaN	NA
<i>Rebalance Time (sec)</i>	0	NA
<i>Testrunner Version</i>	f3ae21f	NA



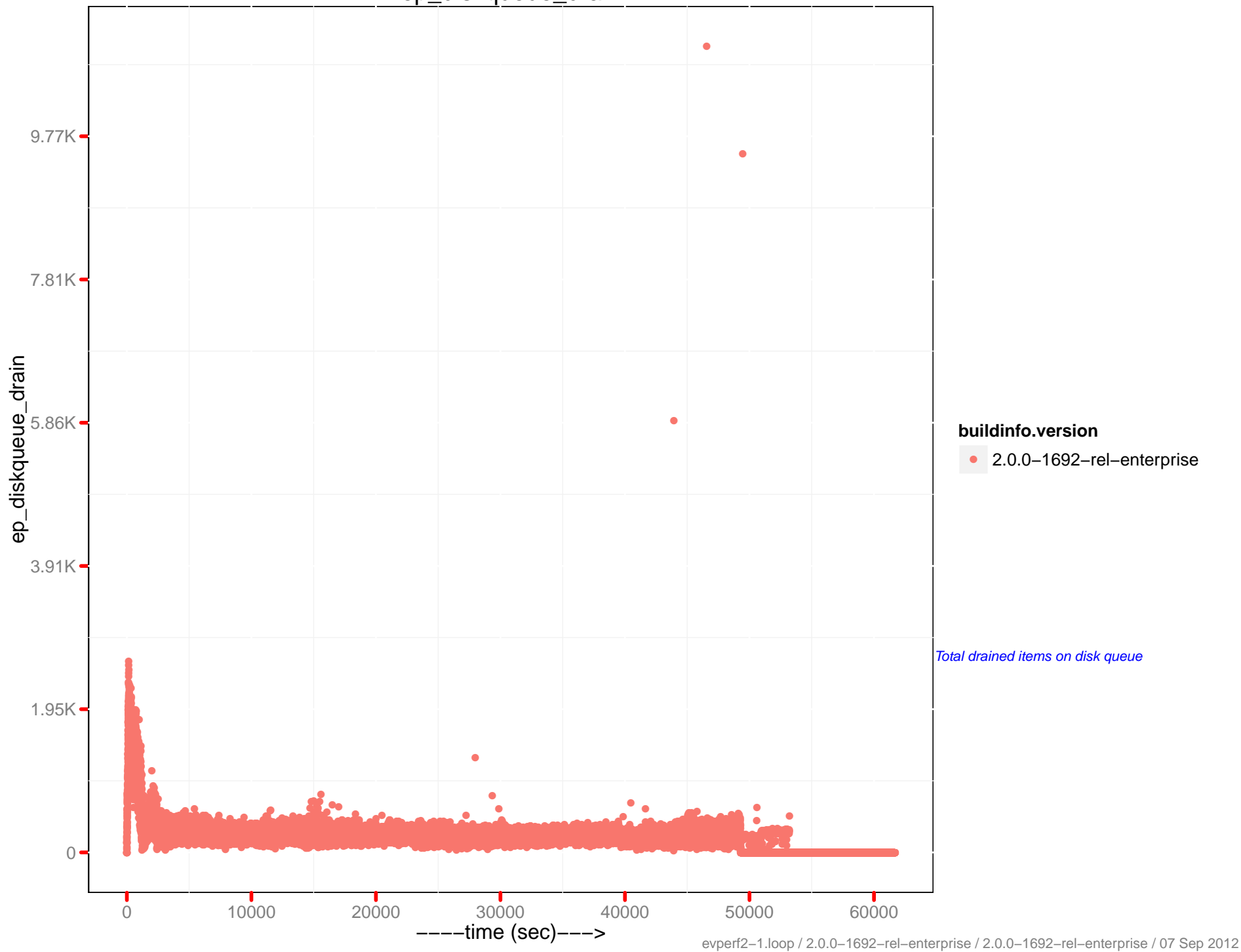
# View read per sec.



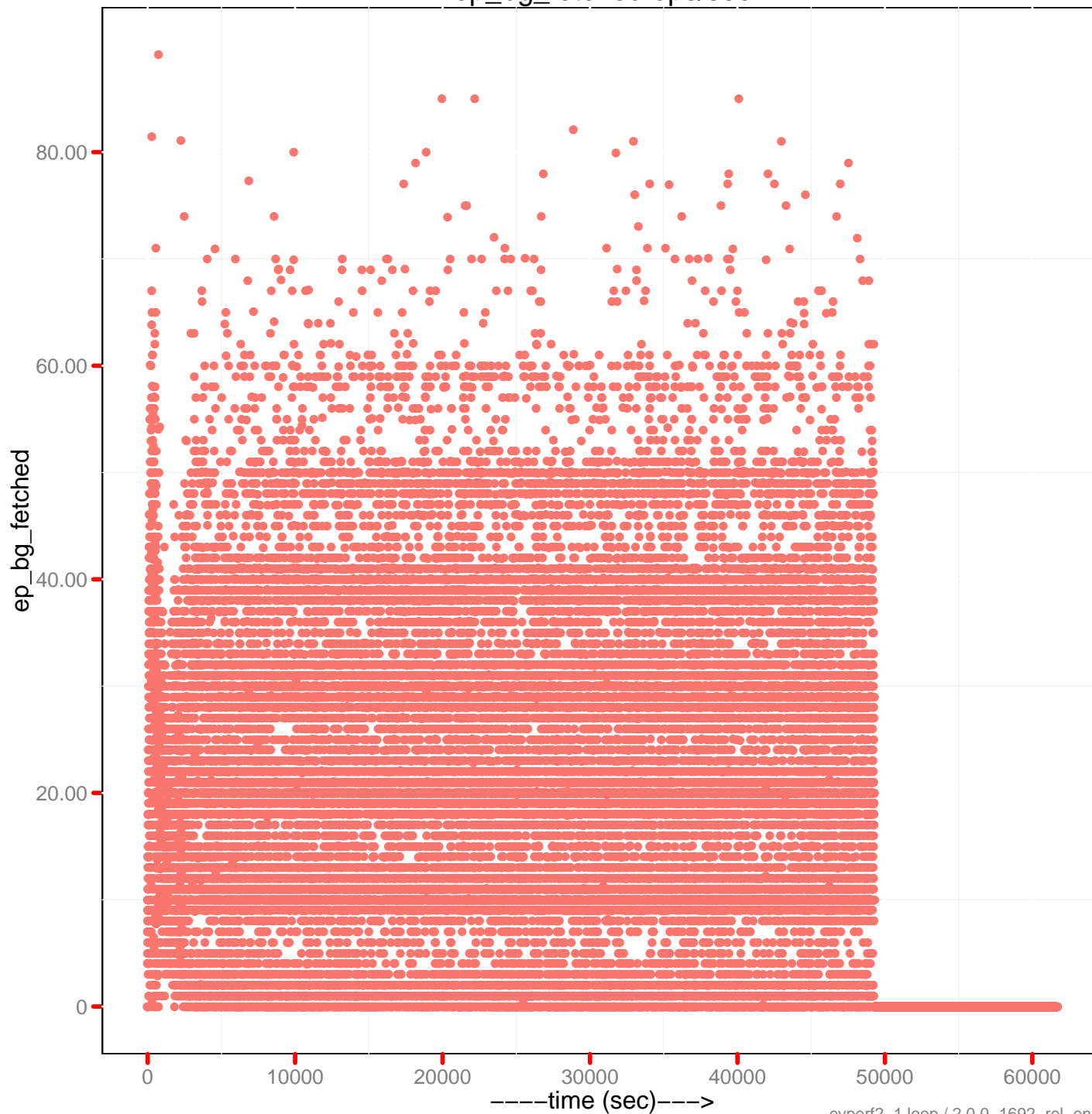
ep queue size



# ep\_diskqueue\_drain



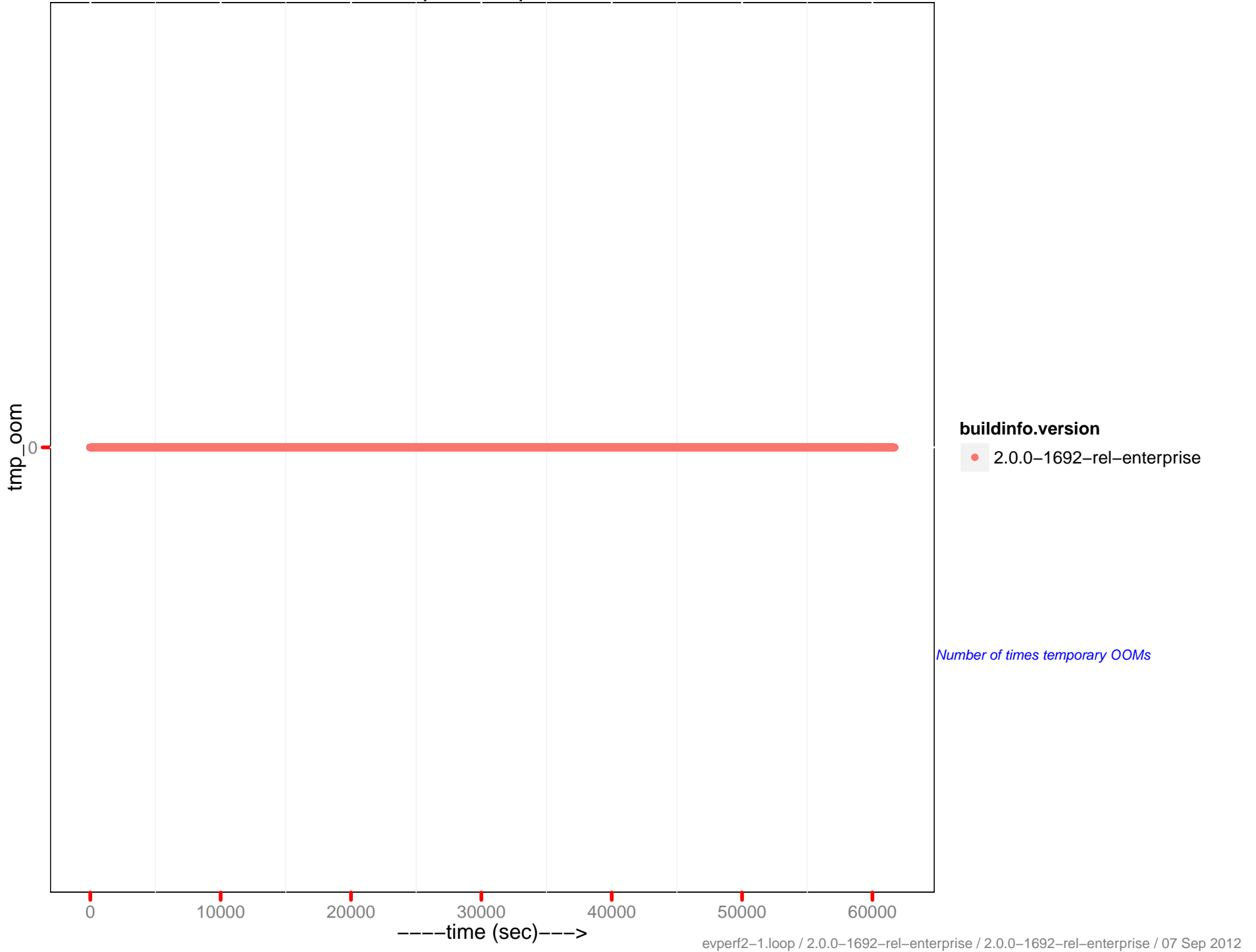
# ep\_bg\_fetched ops/sec



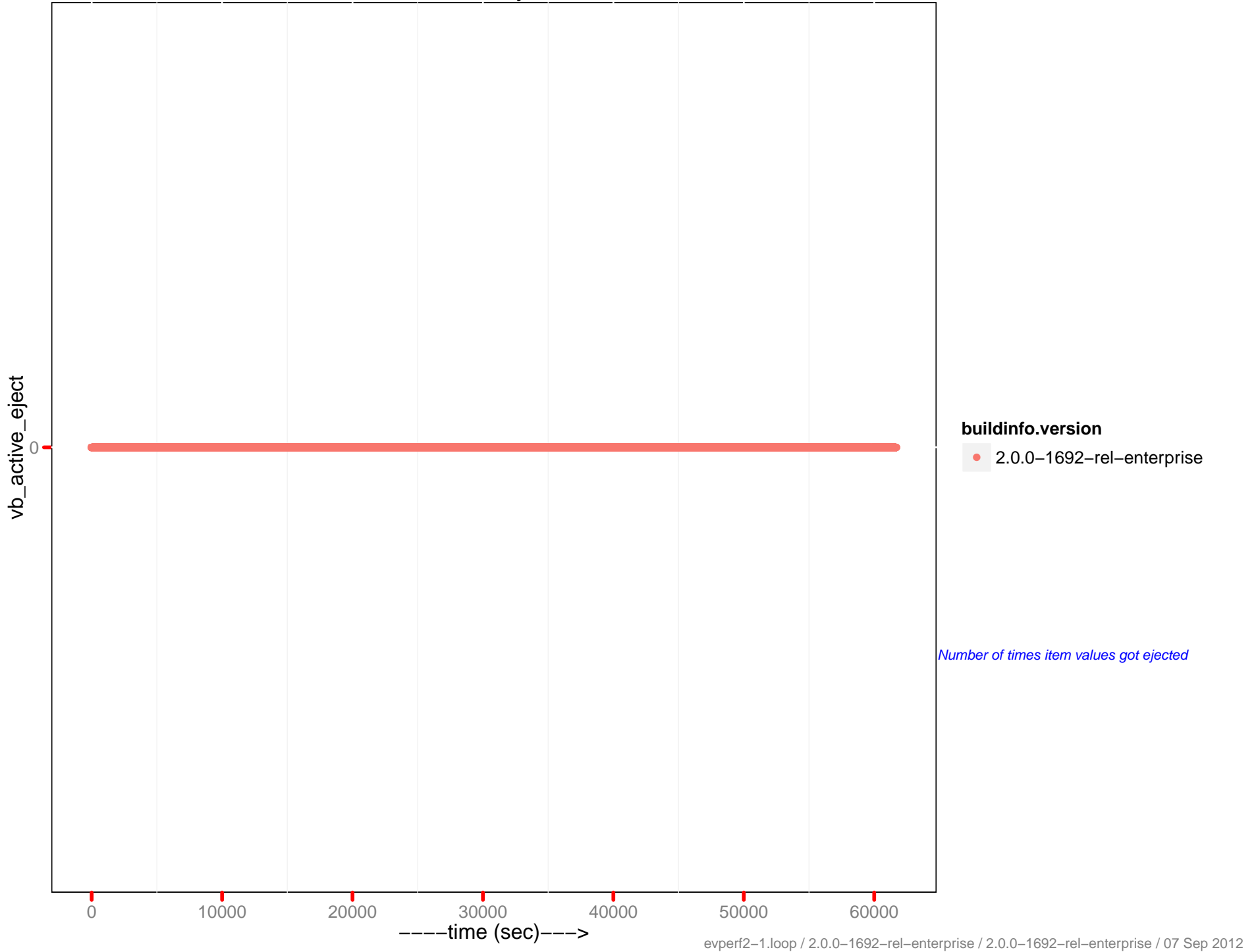
**buildinfo.version**  
● 2.0.0-1692-rel-enterprise

*Number of items fetched from disk*

# tmp\_oom ops/sec



# vb\_active\_eject/sec



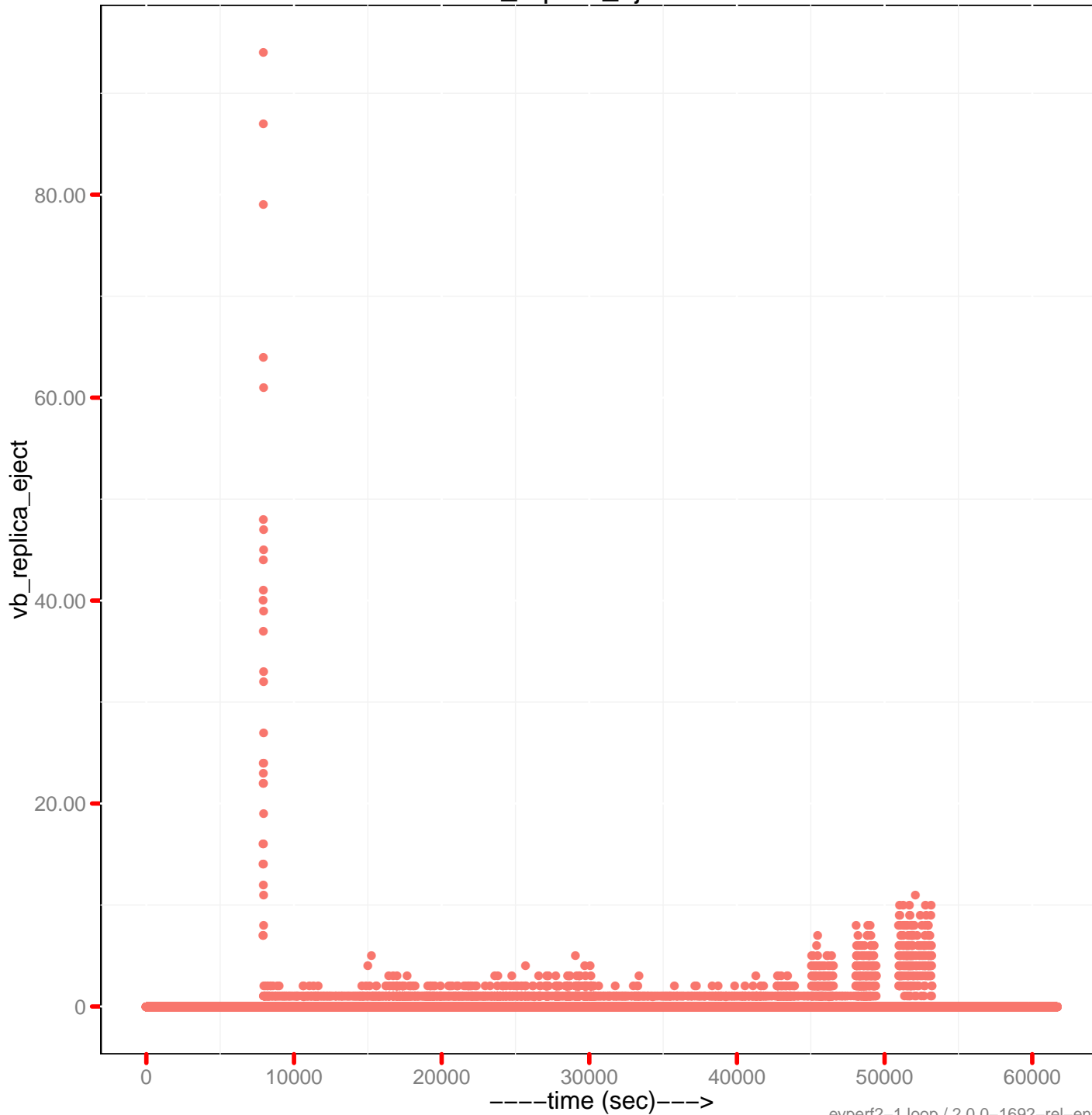
## buildinfo.version

- 2.0.0-1692-rel-enterprise

*Number of times item values got ejected*



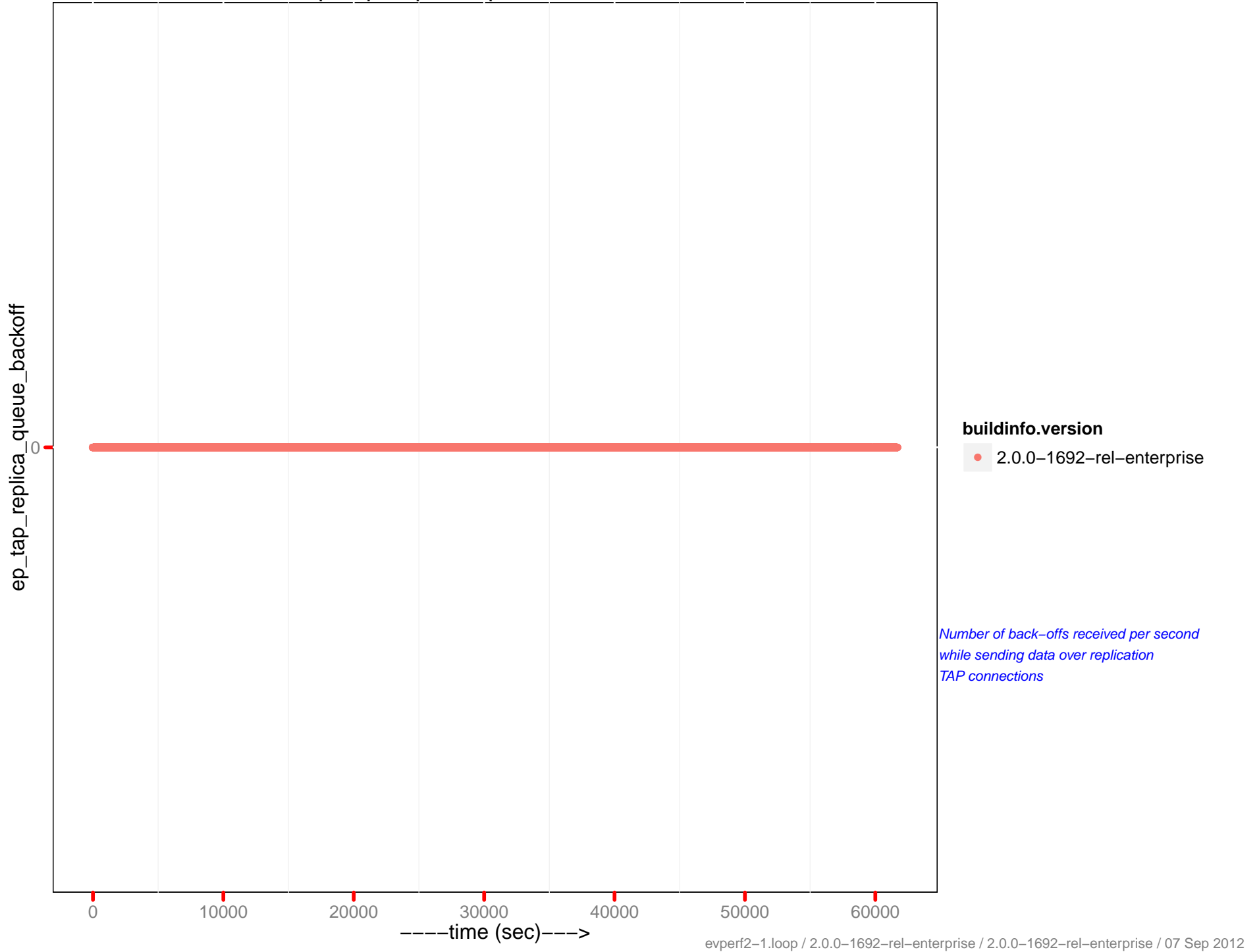
# vb\_replica\_eject/sec



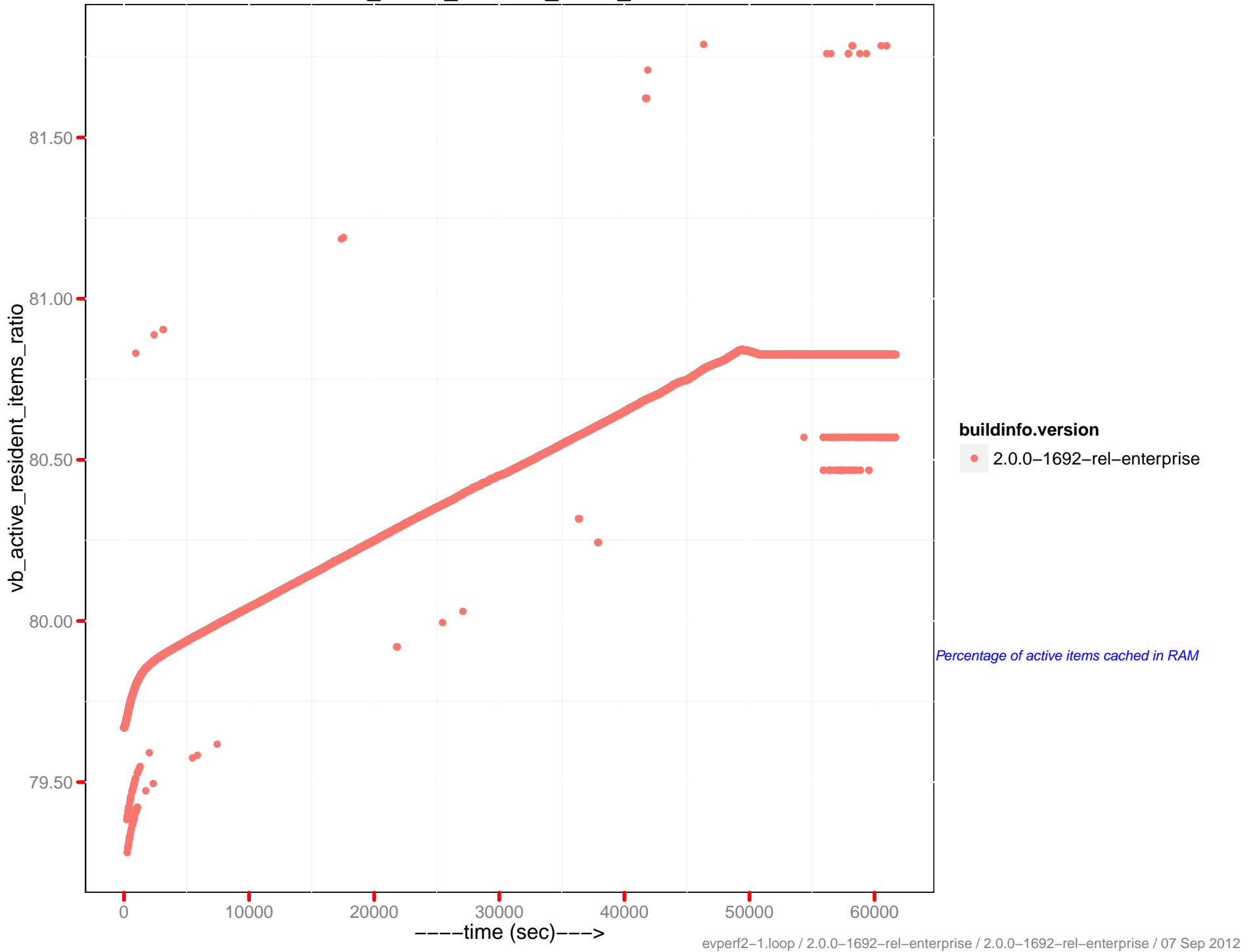
**buildinfo.version**  
● 2.0.0-1692-rel-enterprise

*Number of times item values got ejected*

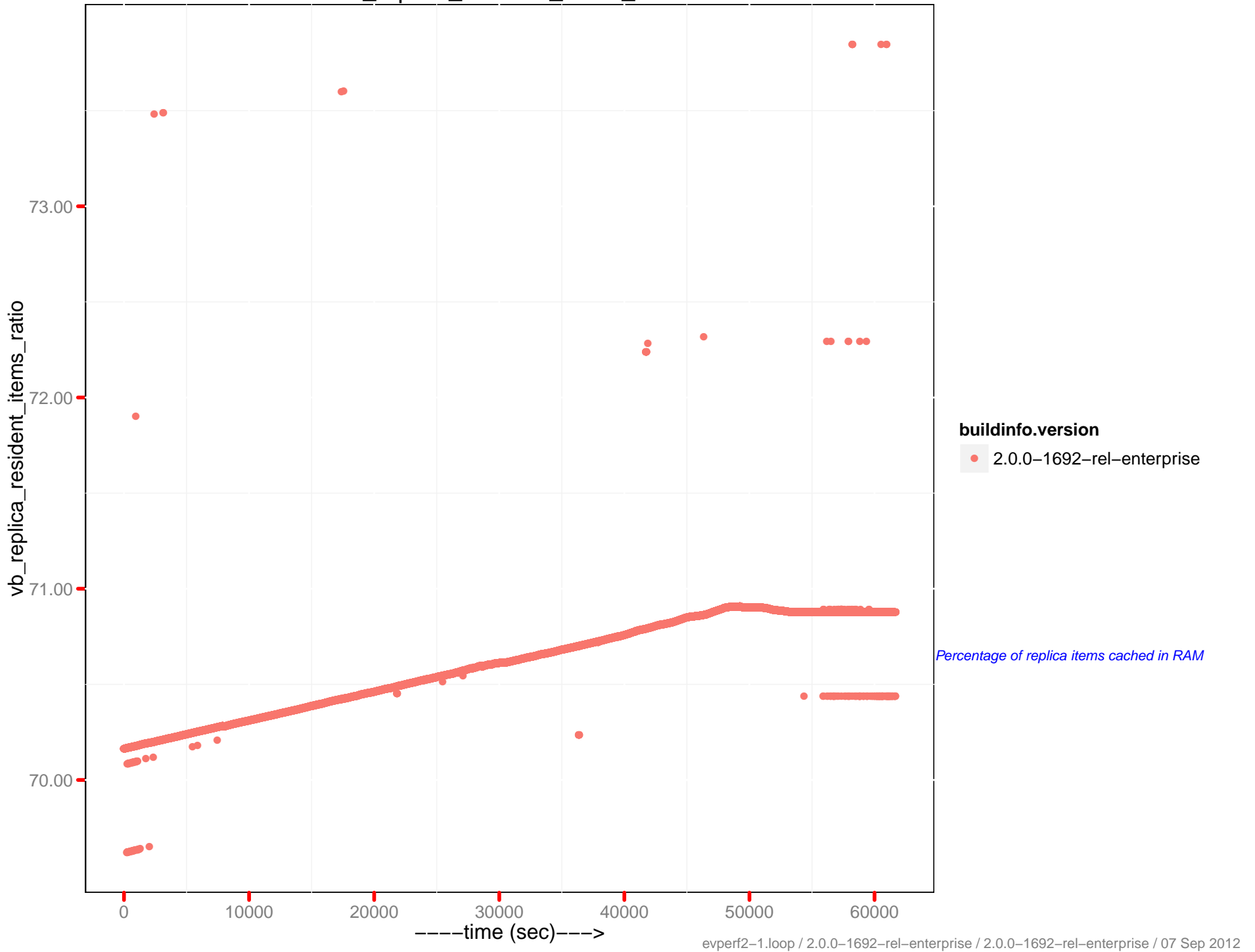
# ep\_tap\_replica\_queue\_backoff/sec



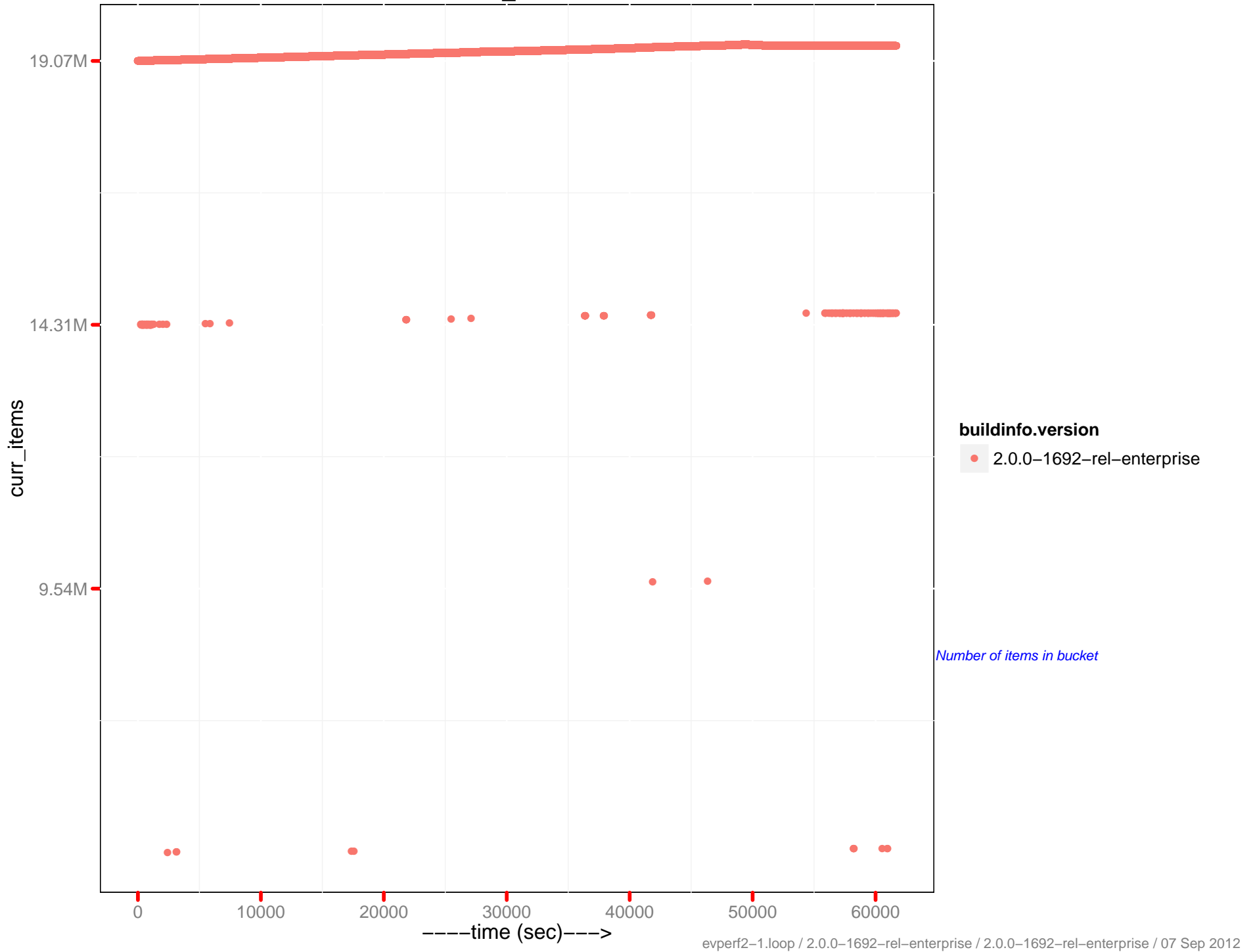
# vb\_active\_resident\_items\_ratio



# vb\_replica\_resident\_items\_ratio



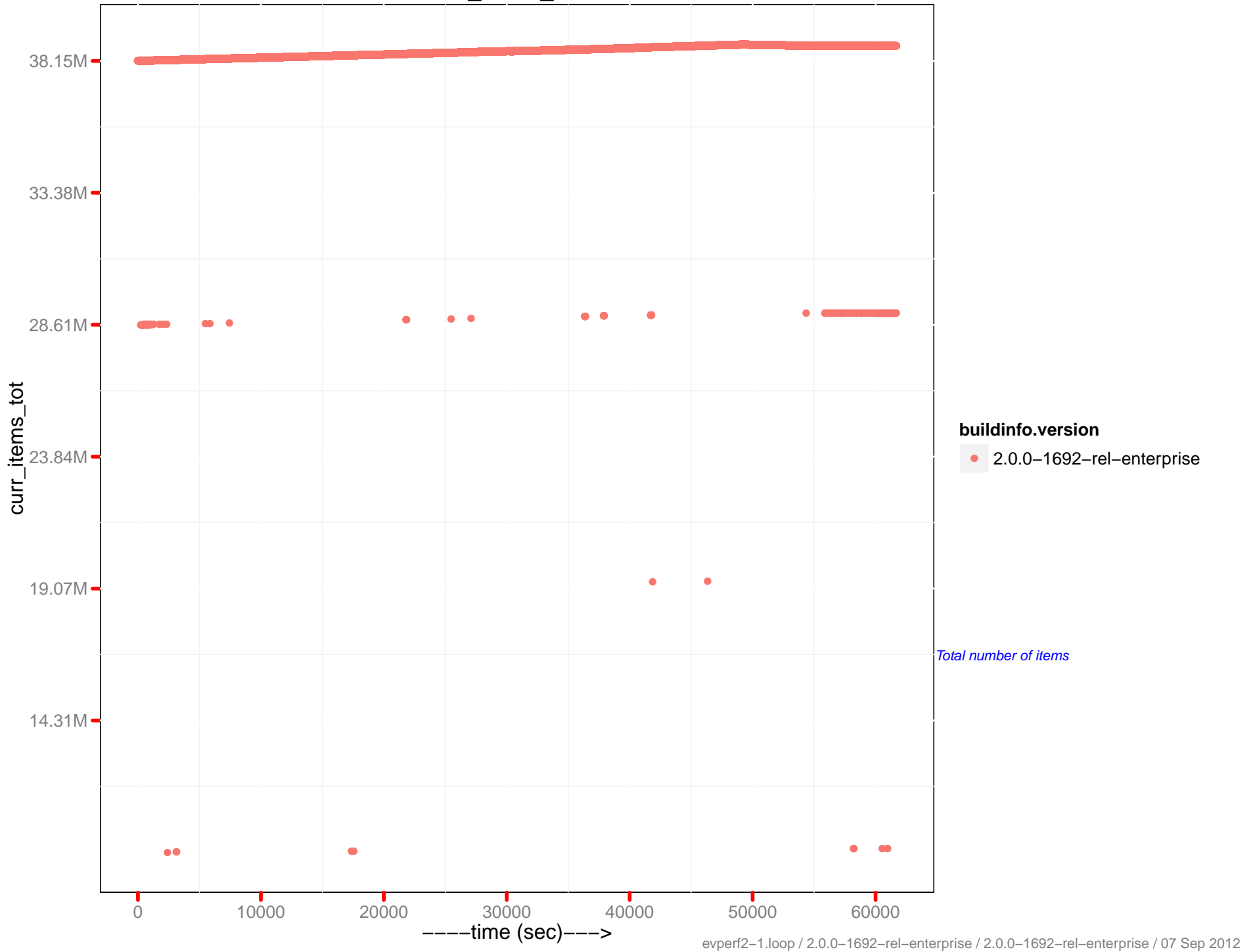
# curr\_items



**buildinfo.version**  
● 2.0.0-1692-rel-enterprise

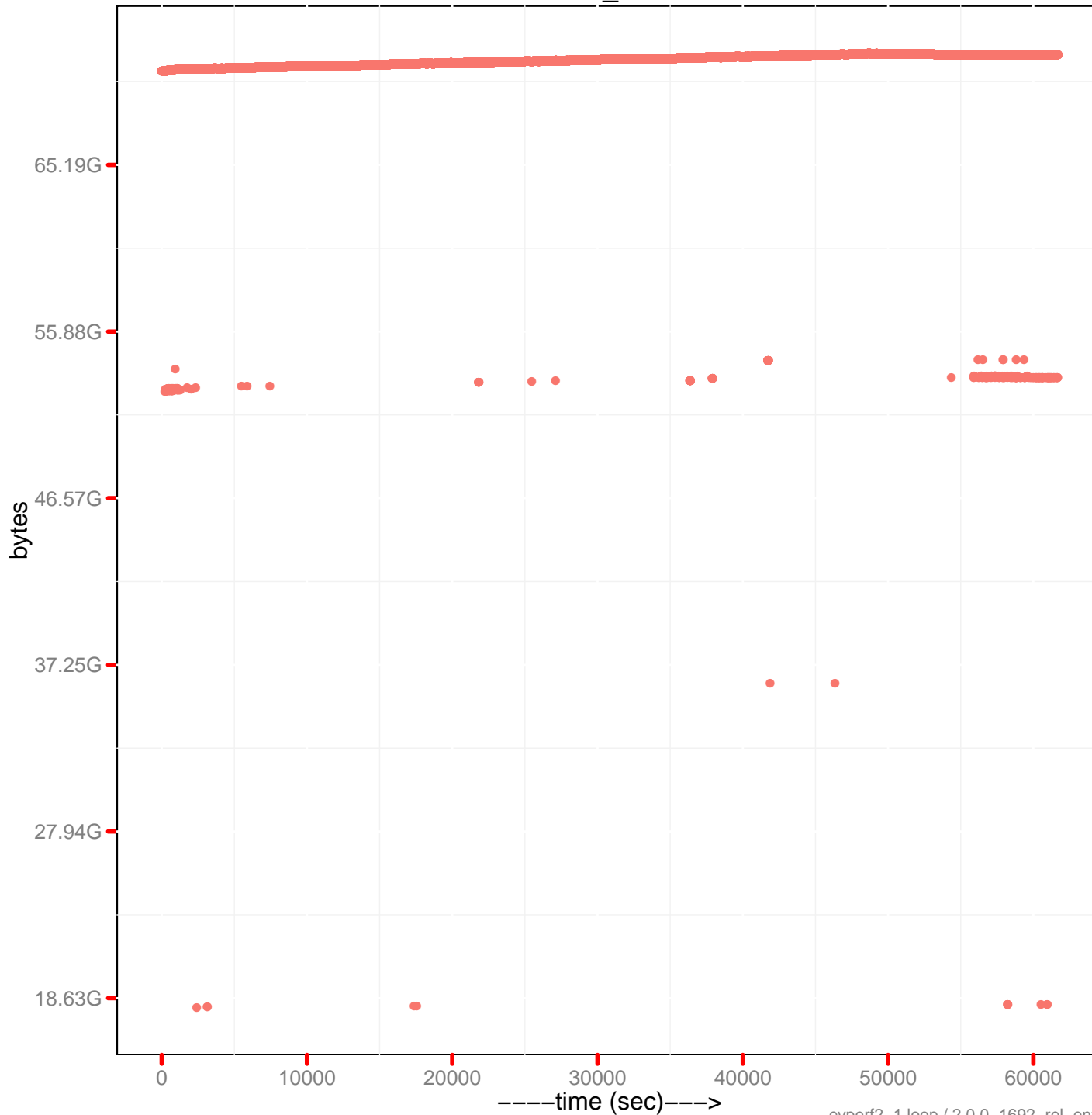
*Number of items in bucket*

# cur\_items\_total



Total number of items

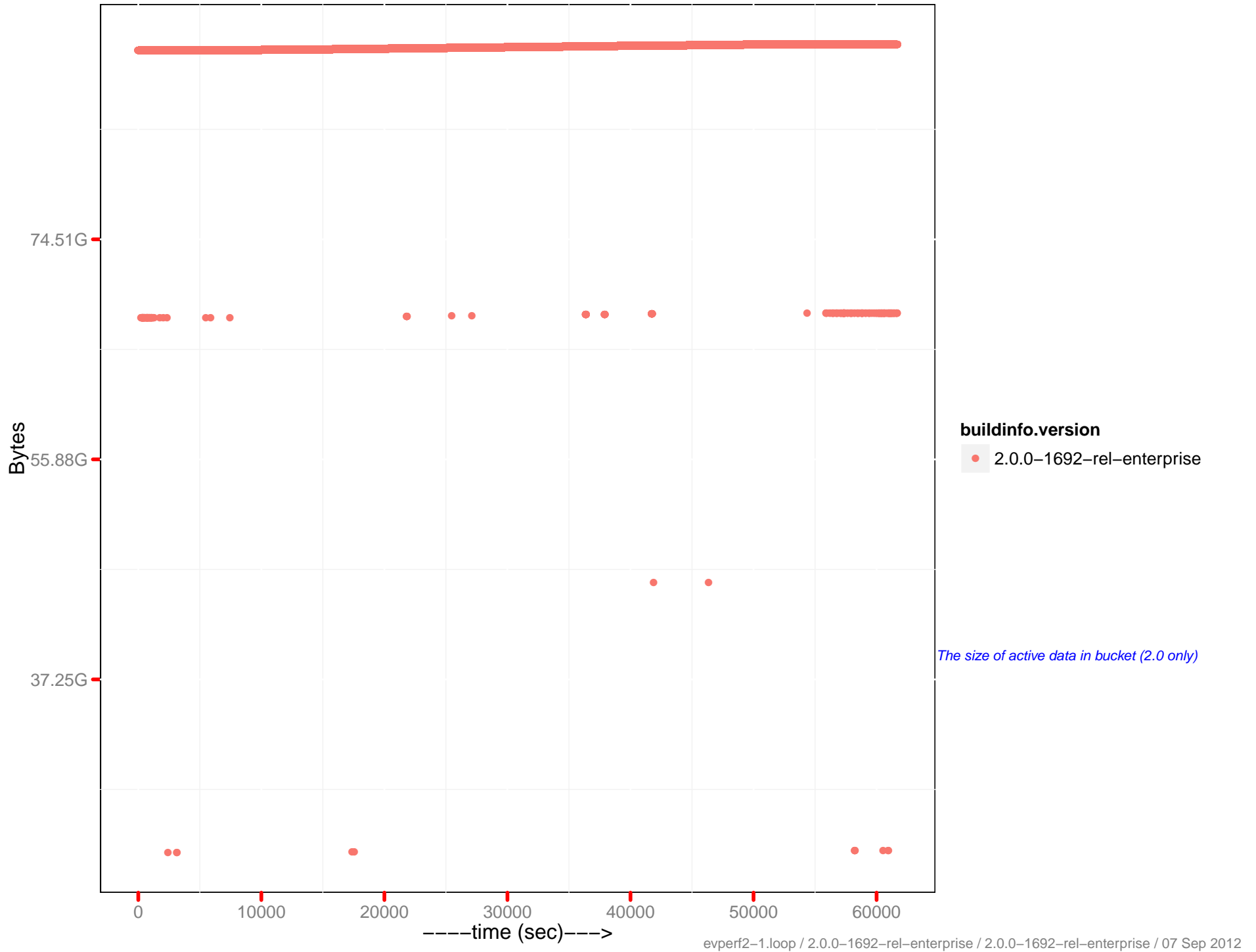
# mem\_used



**buildinfo.version**  
● 2.0.0-1692-rel-enterprise

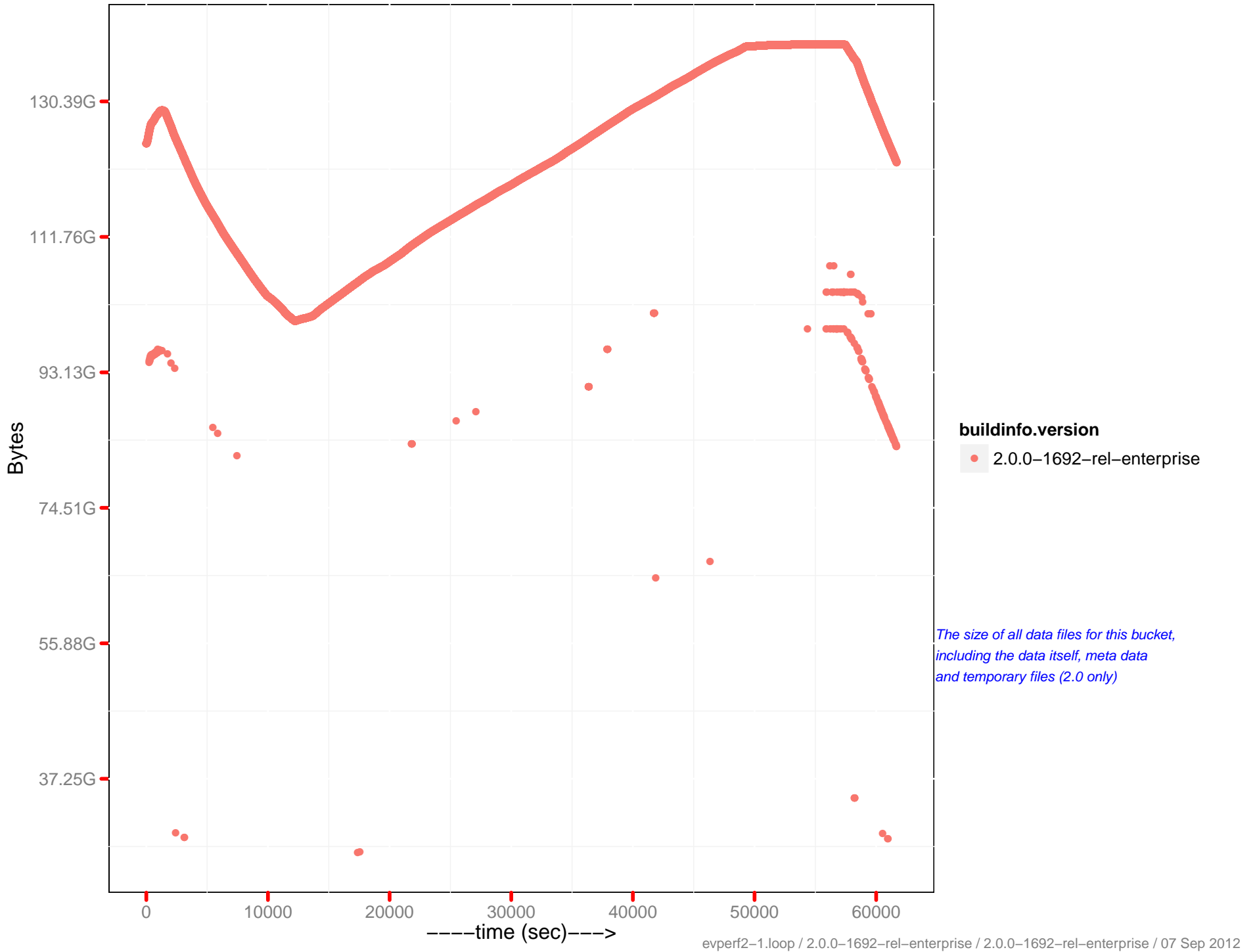
*Engine's total memory usage*

# Docs data size

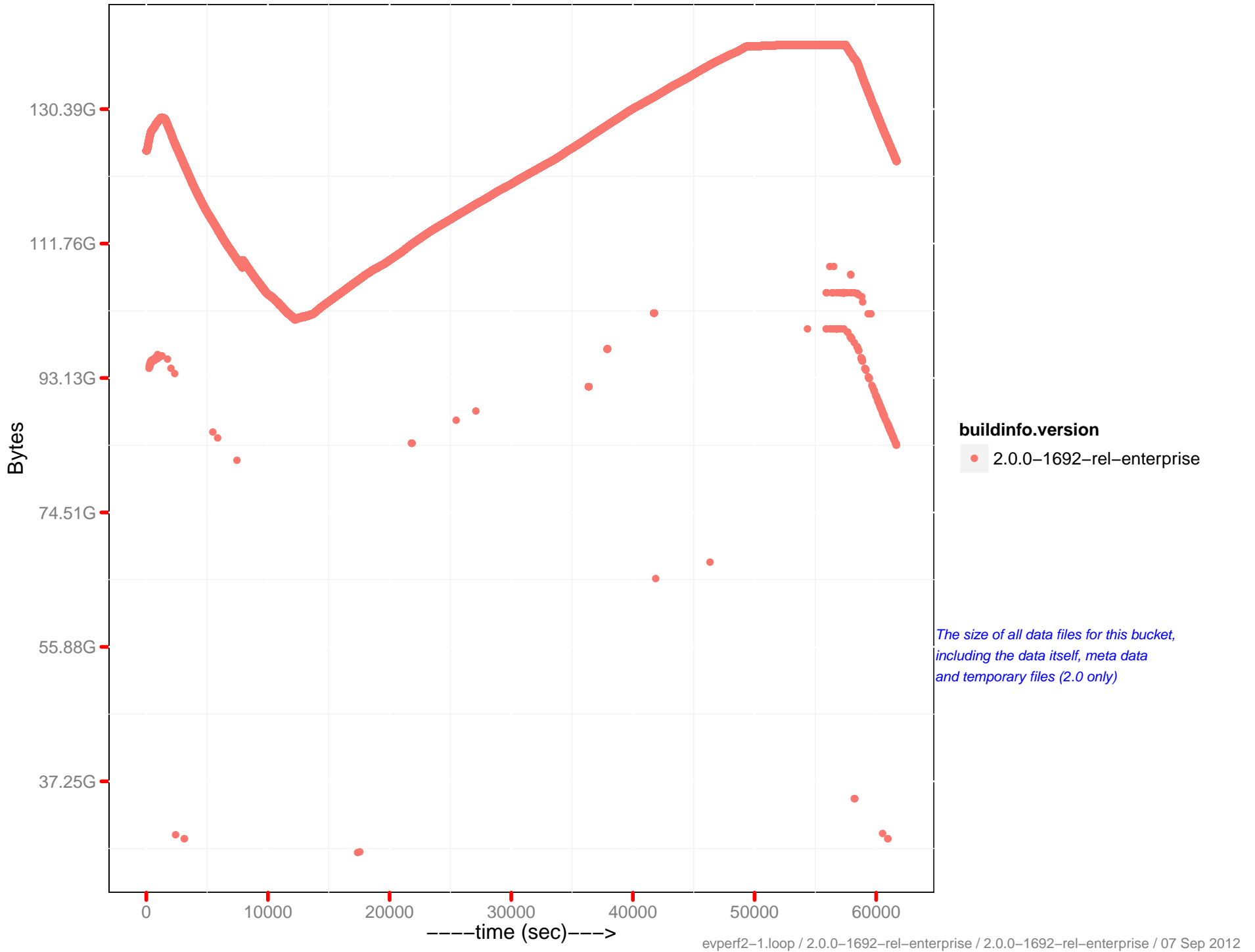




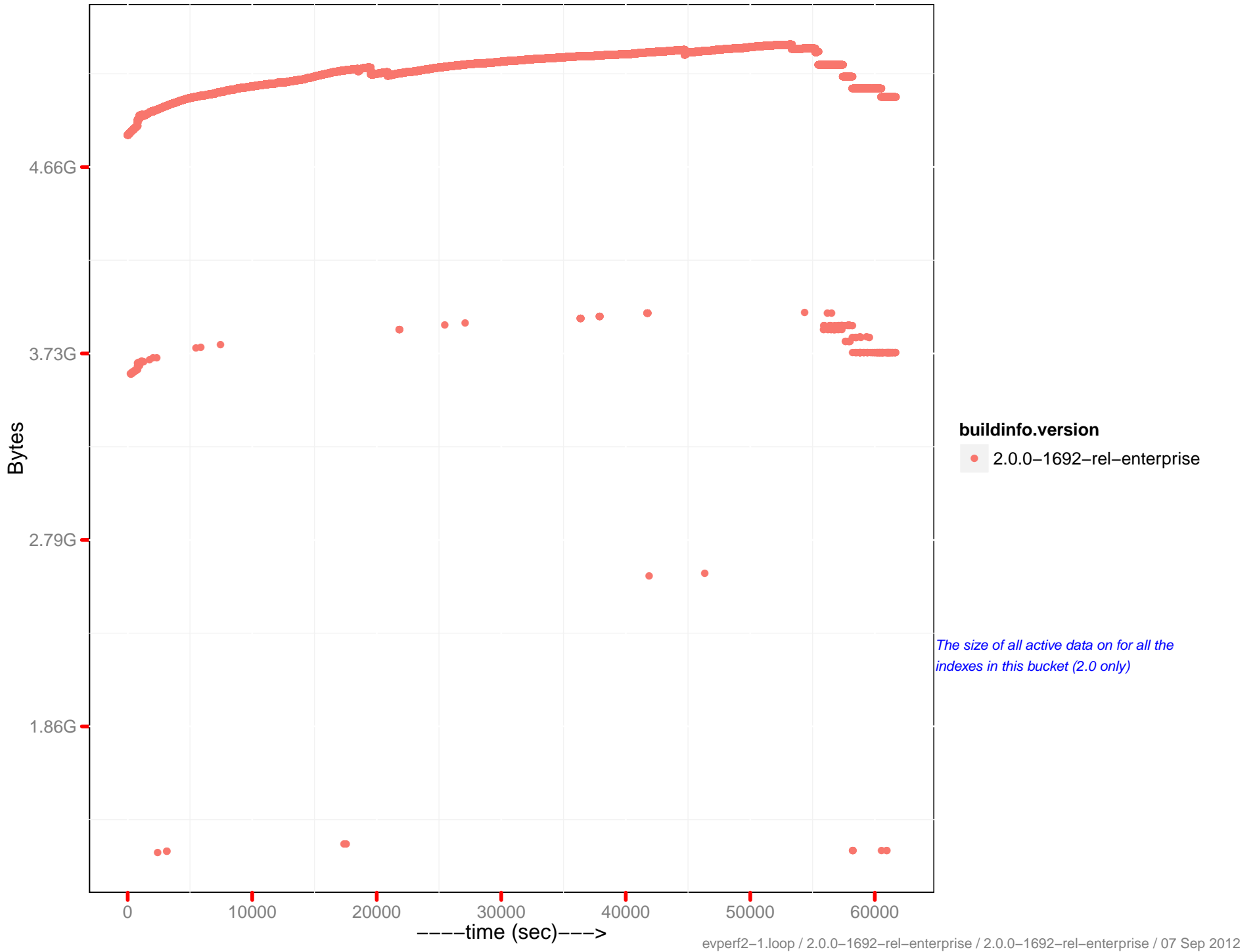
Docs disk size



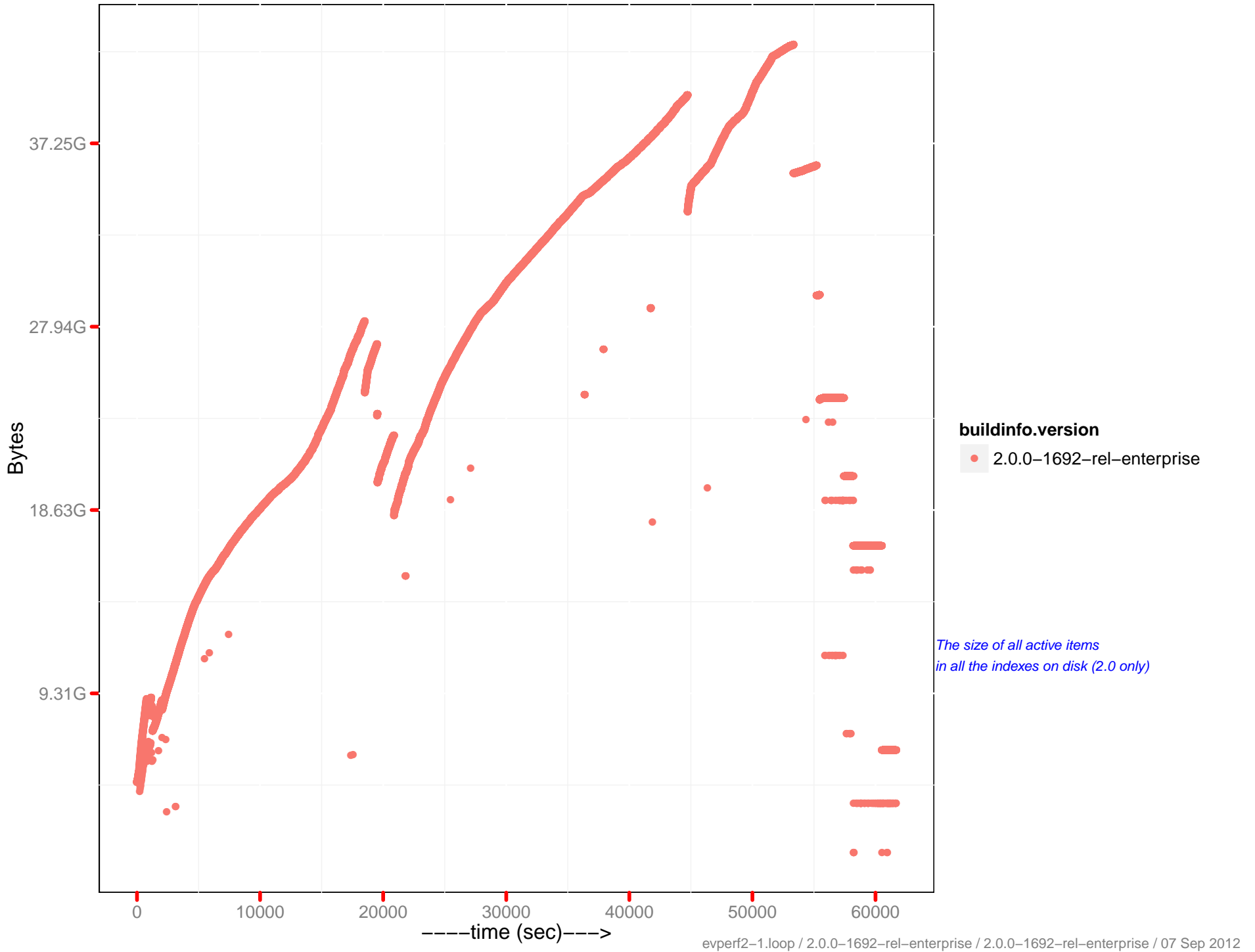
# Docs actual disk size



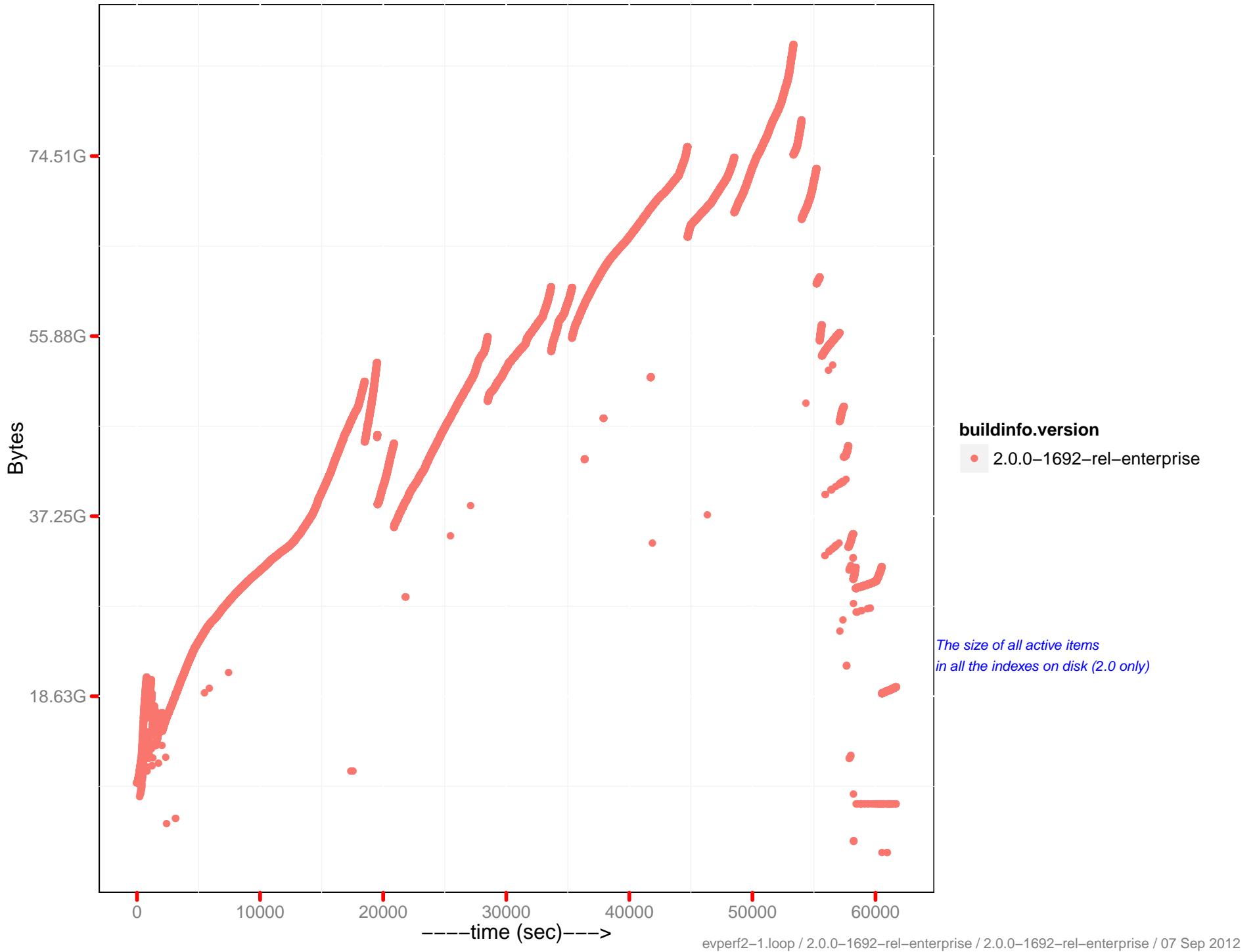
# Views data size



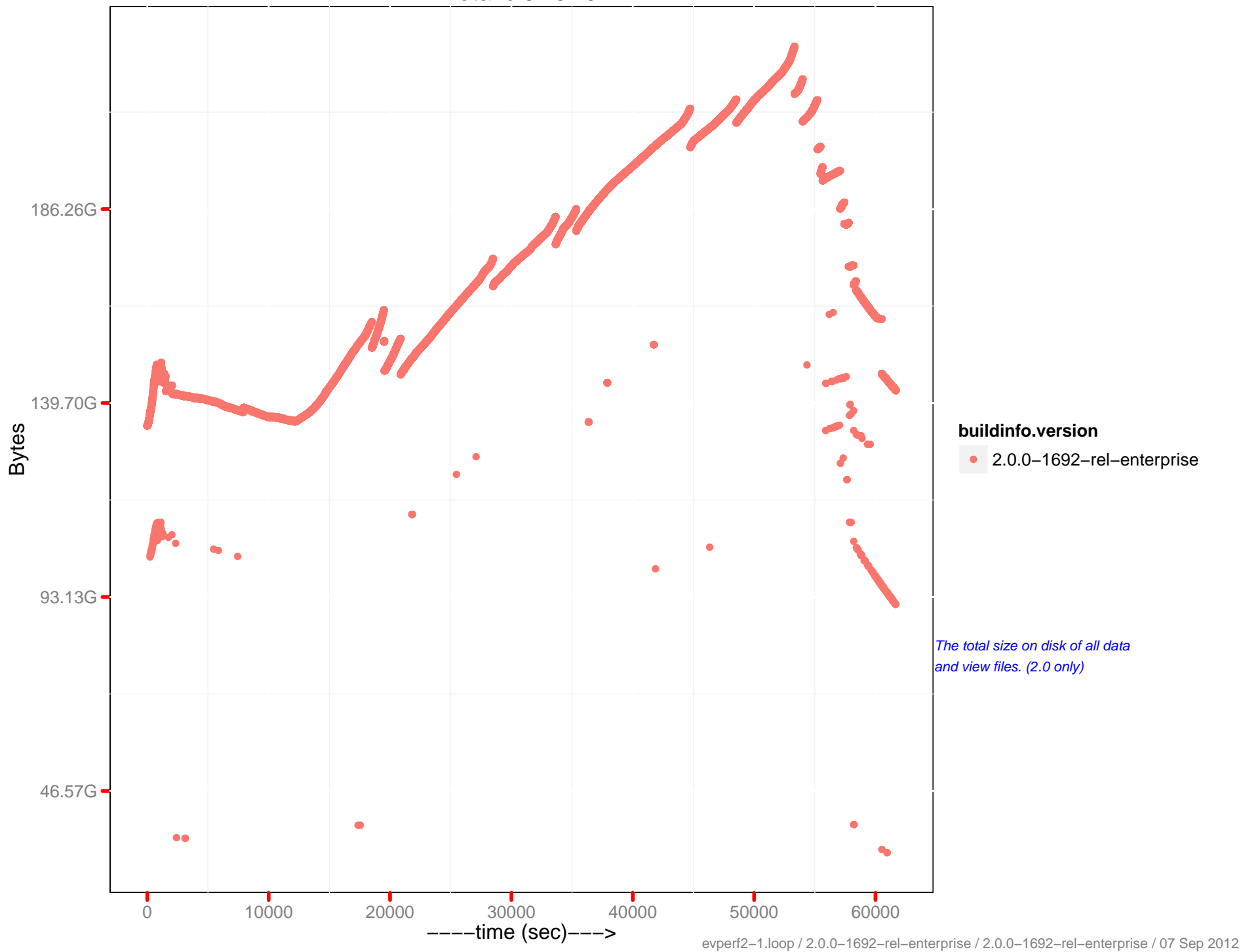
# Views disk size



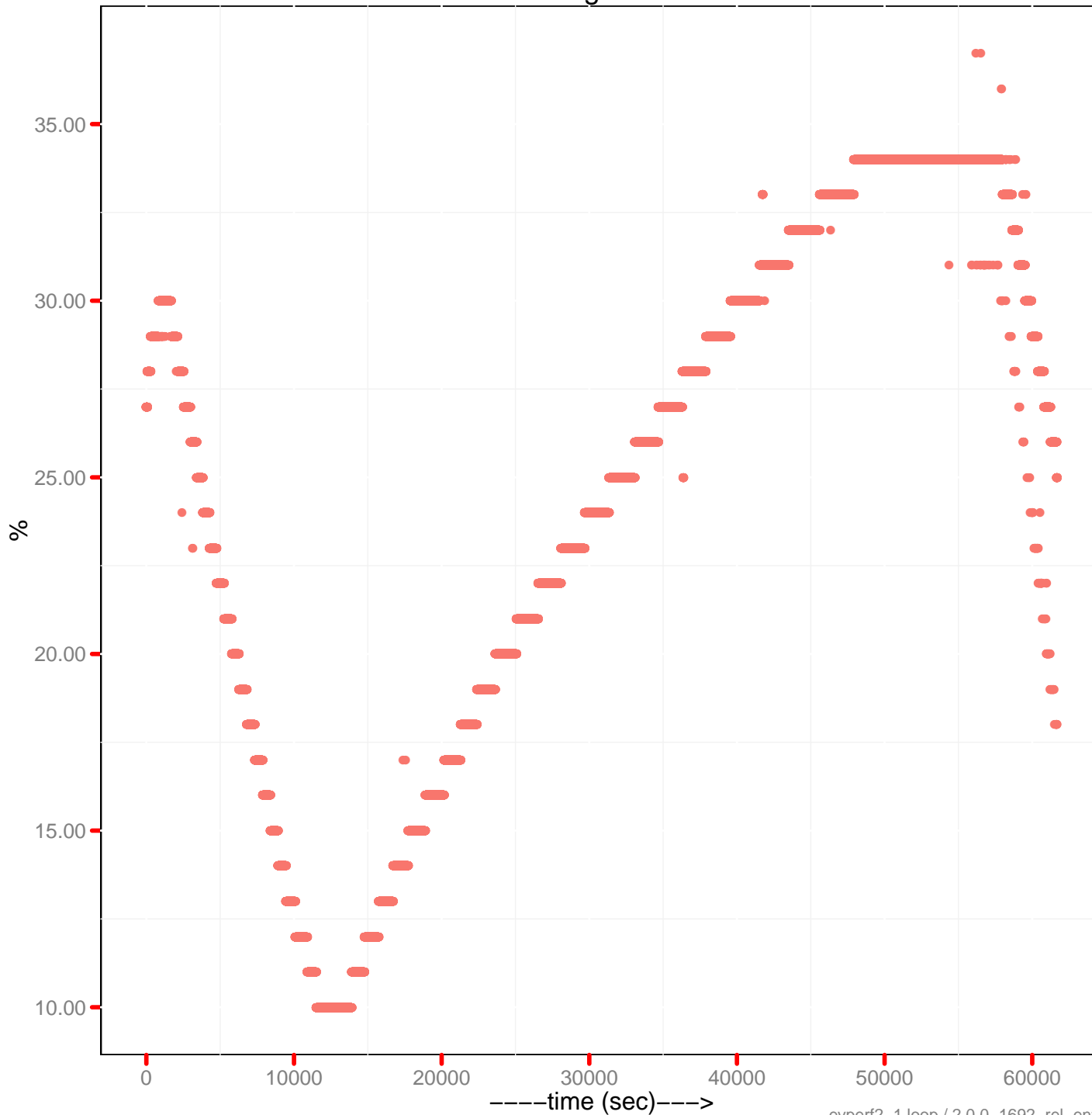
Views actual disk size



# Total disk size



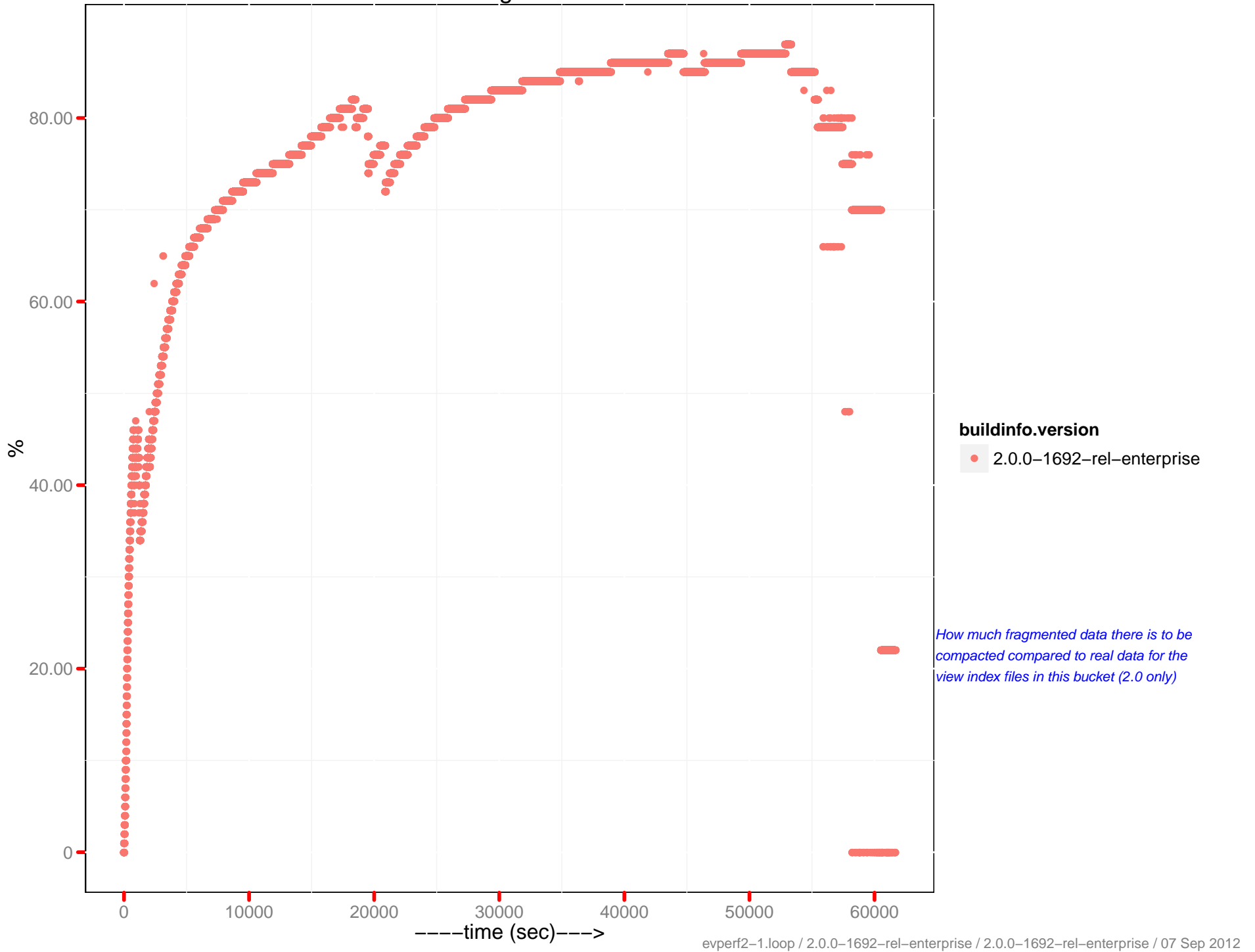
# Docs fragmentation



**buildinfo.version**  
2.0.0-1692-rel-enterprise

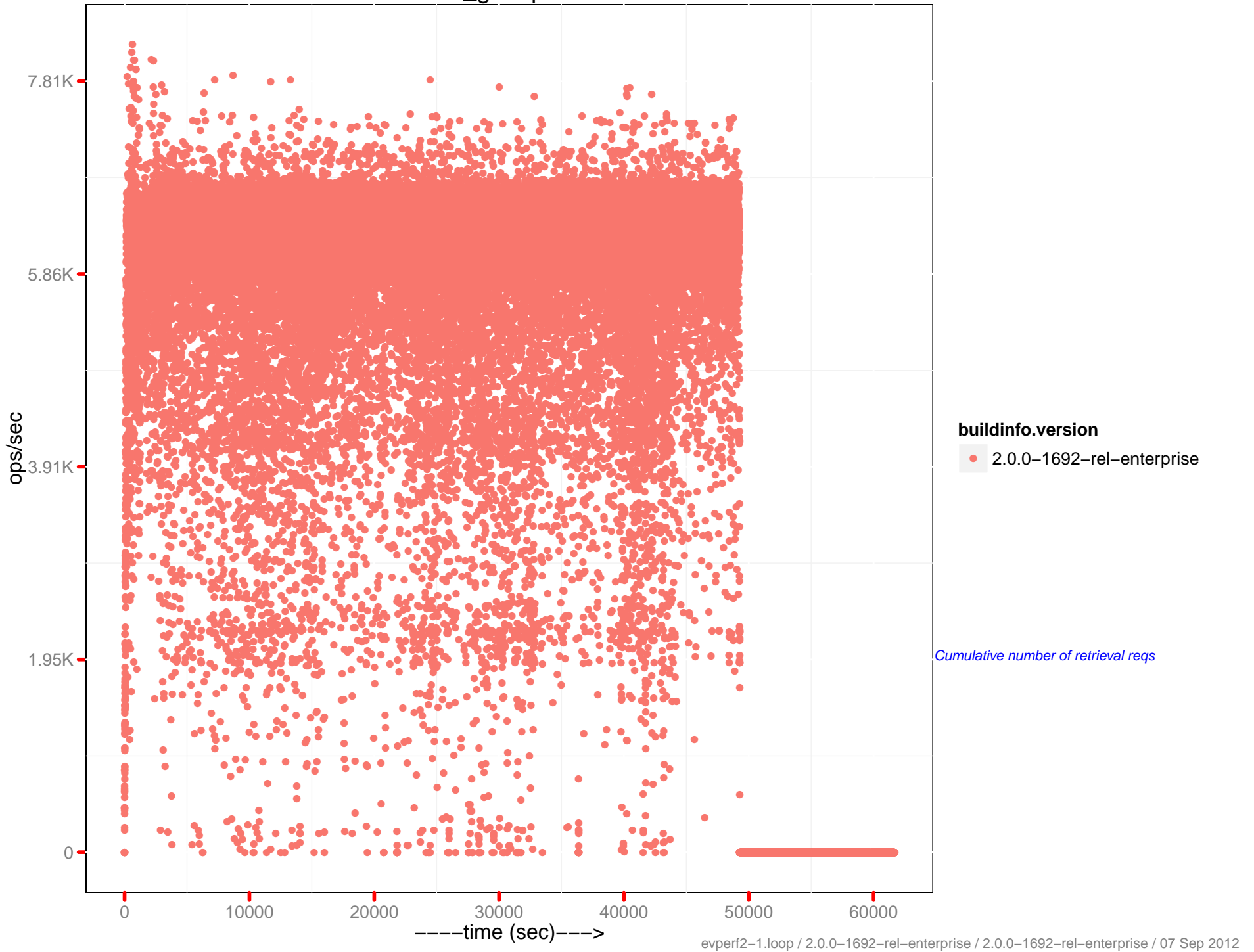
*How much fragmented data there is to be compacted compared to real data for the data files in this bucket (2.0 only)*

# Views fragmentation

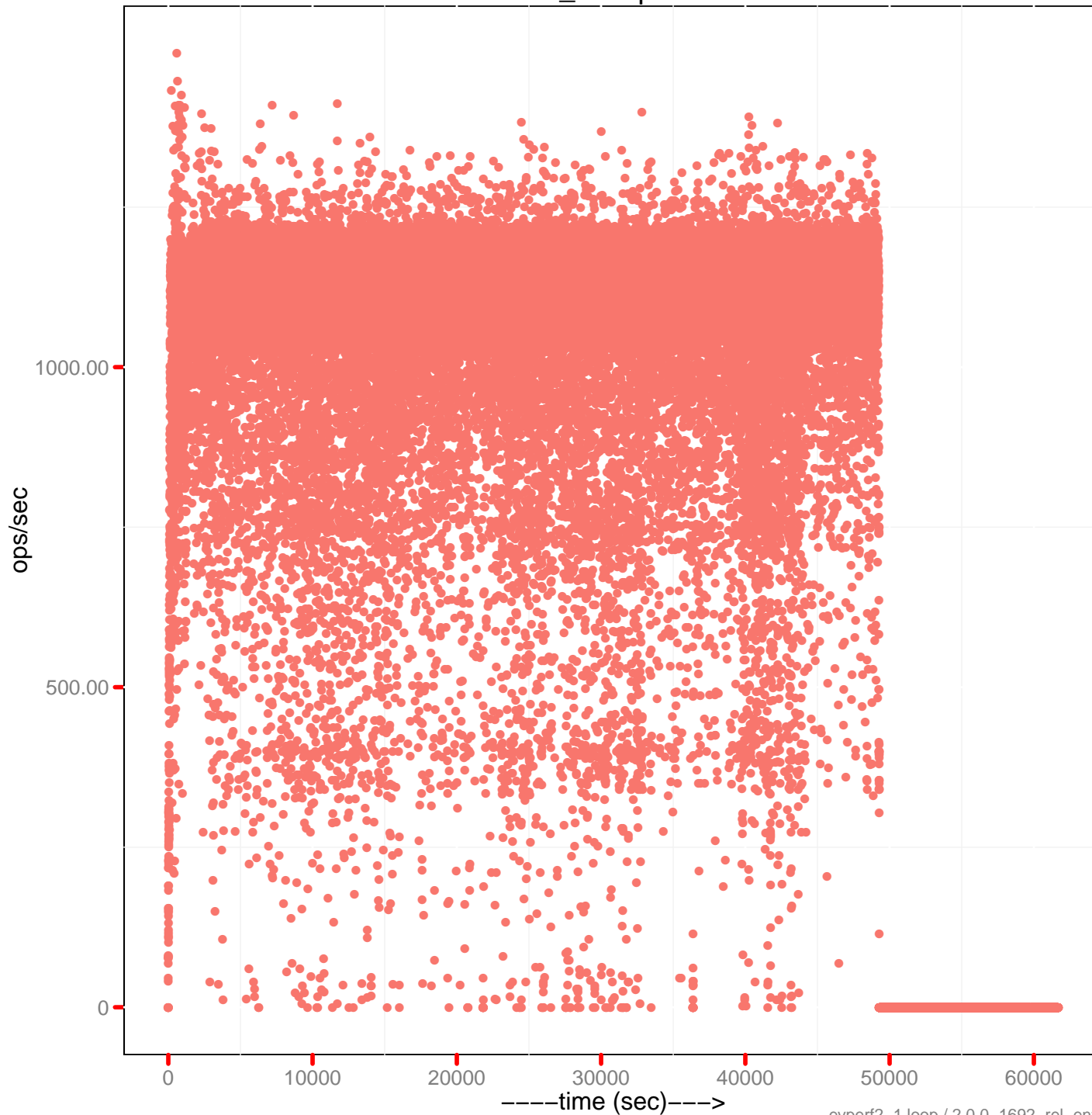




# cmd\_get ops/sec



# cmd\_set ops/sec



## buildinfo.version

• 2.0.0-1692-rel-enterprise

*Cumulative number of set reqs*

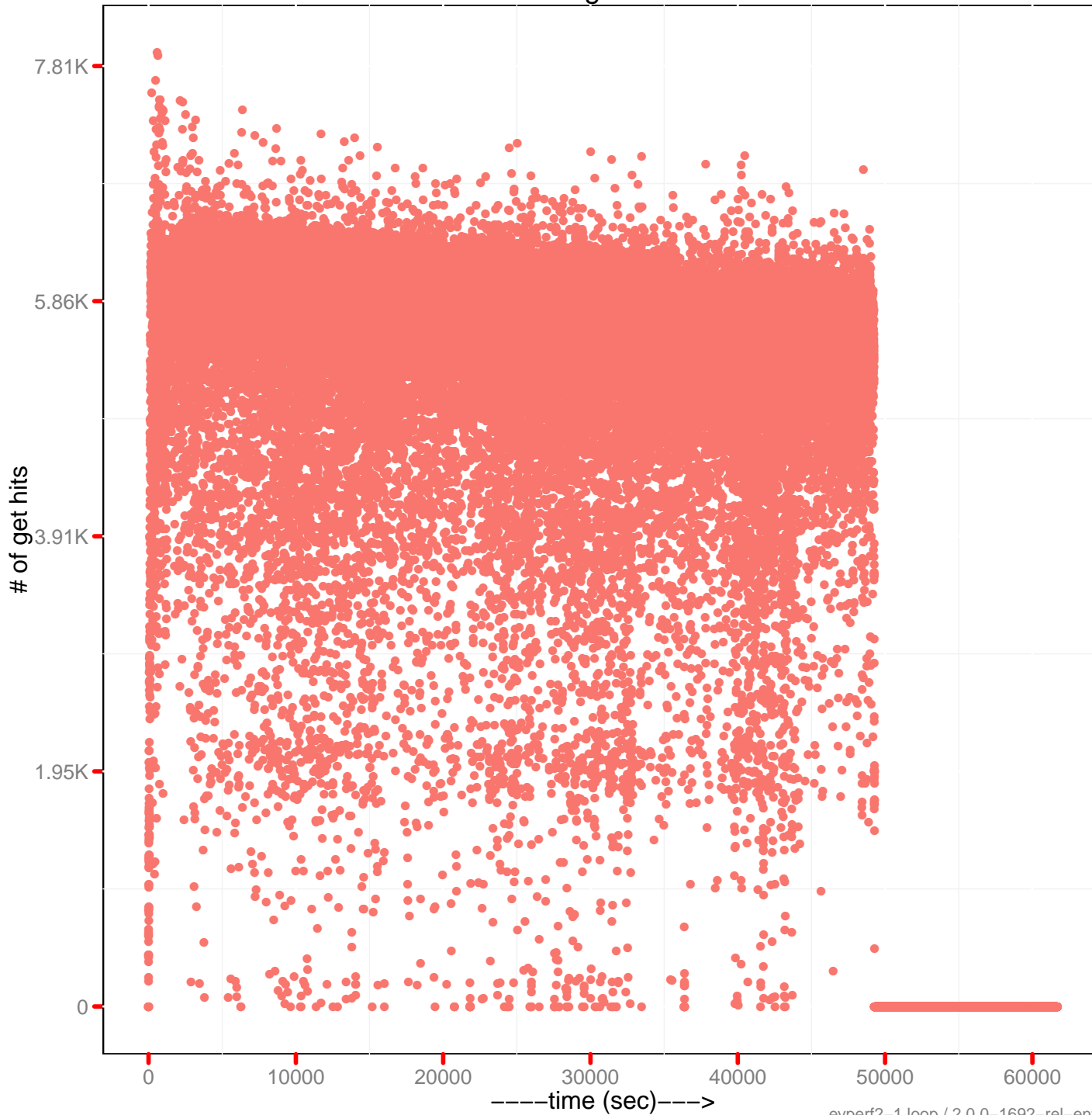
# # of get misses



**buildinfo.version**  
● 2.0.0-1692-rel-enterprise

*Number of get operations per second  
for data that the bucket does not contain*

# # of get hits

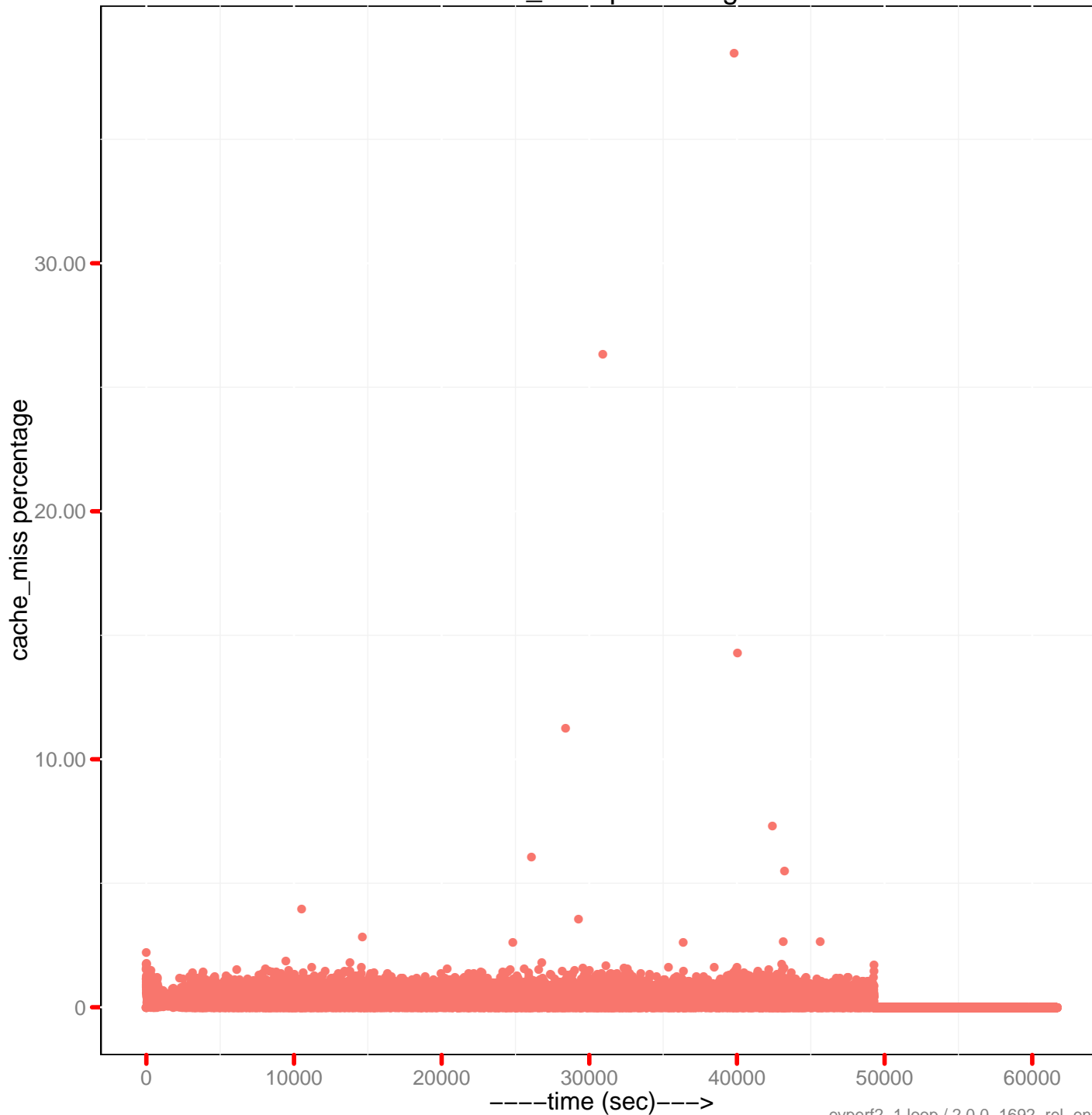


## buildinfo.version

- 2.0.0-1692-rel-enterprise

*Number of get operations per second  
for data that the bucket contains*

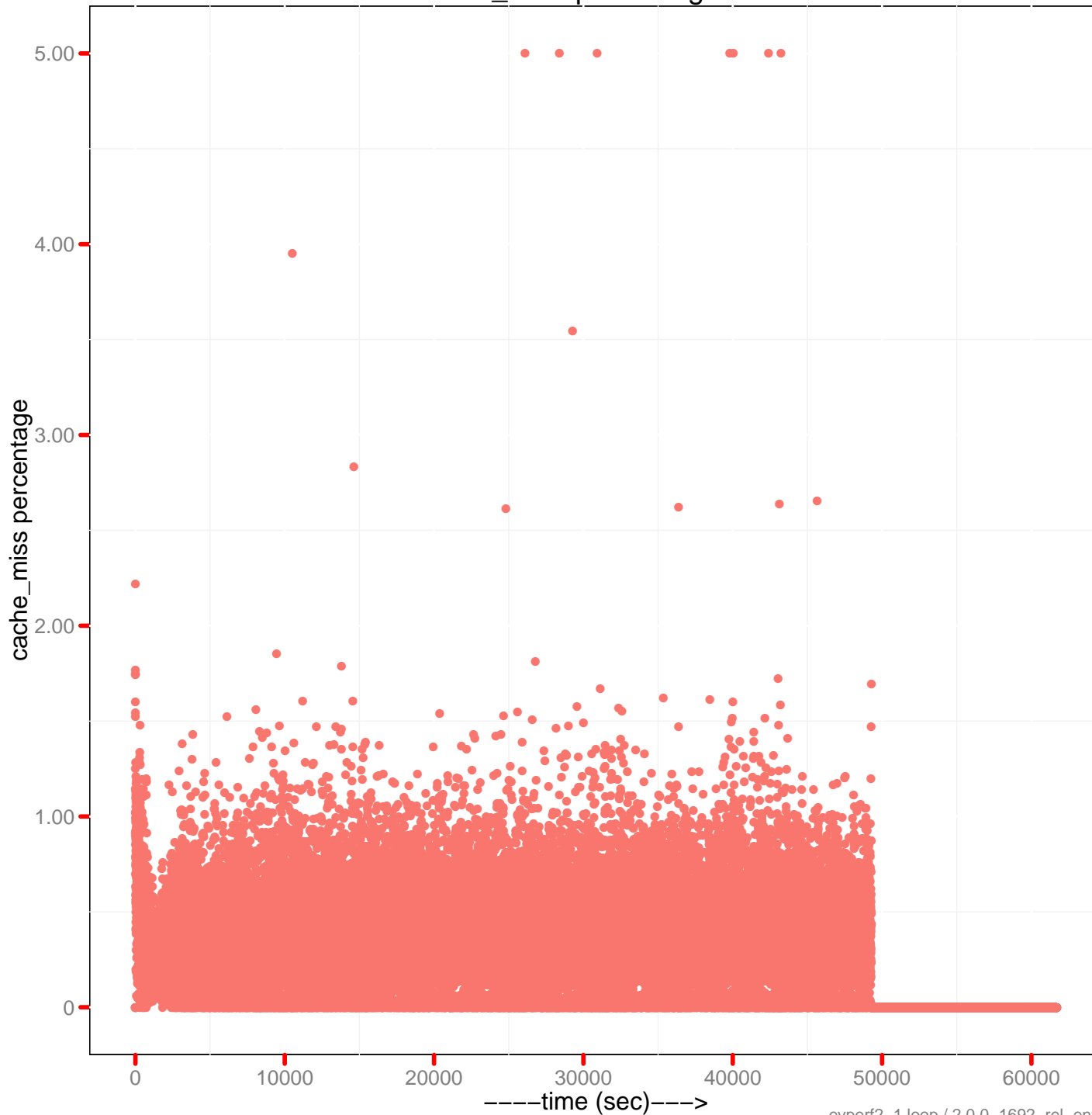
# cache\_miss percentage



**buildinfo.version**  
● 2.0.0-1692-rel-enterprise

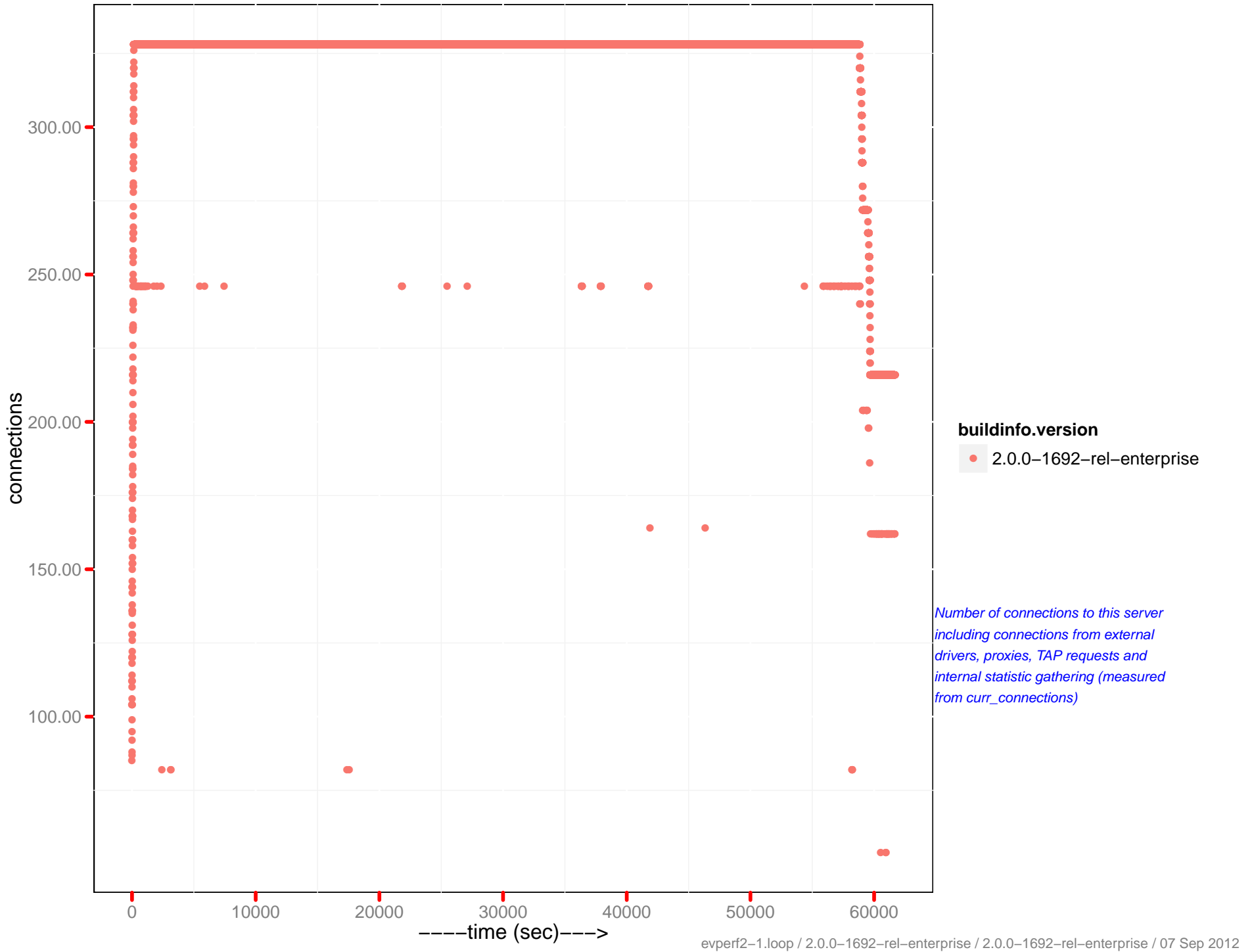
*Percentage of reads per second  
from disk as opposed to RAM*

cache\_miss percentage 0-5

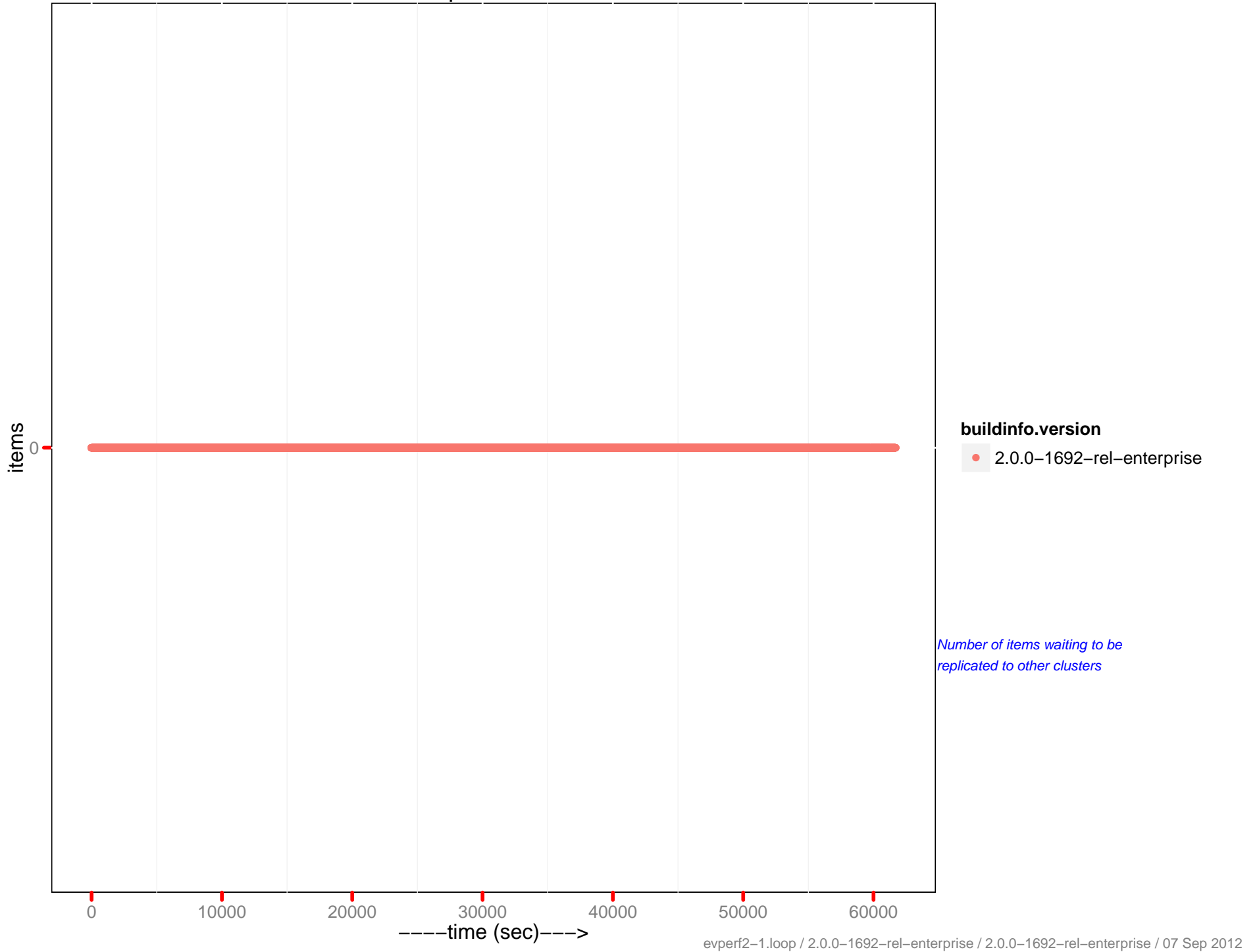


**buildinfo.version**  
● 2.0.0-1692-rel-enterprise

# Number of connections

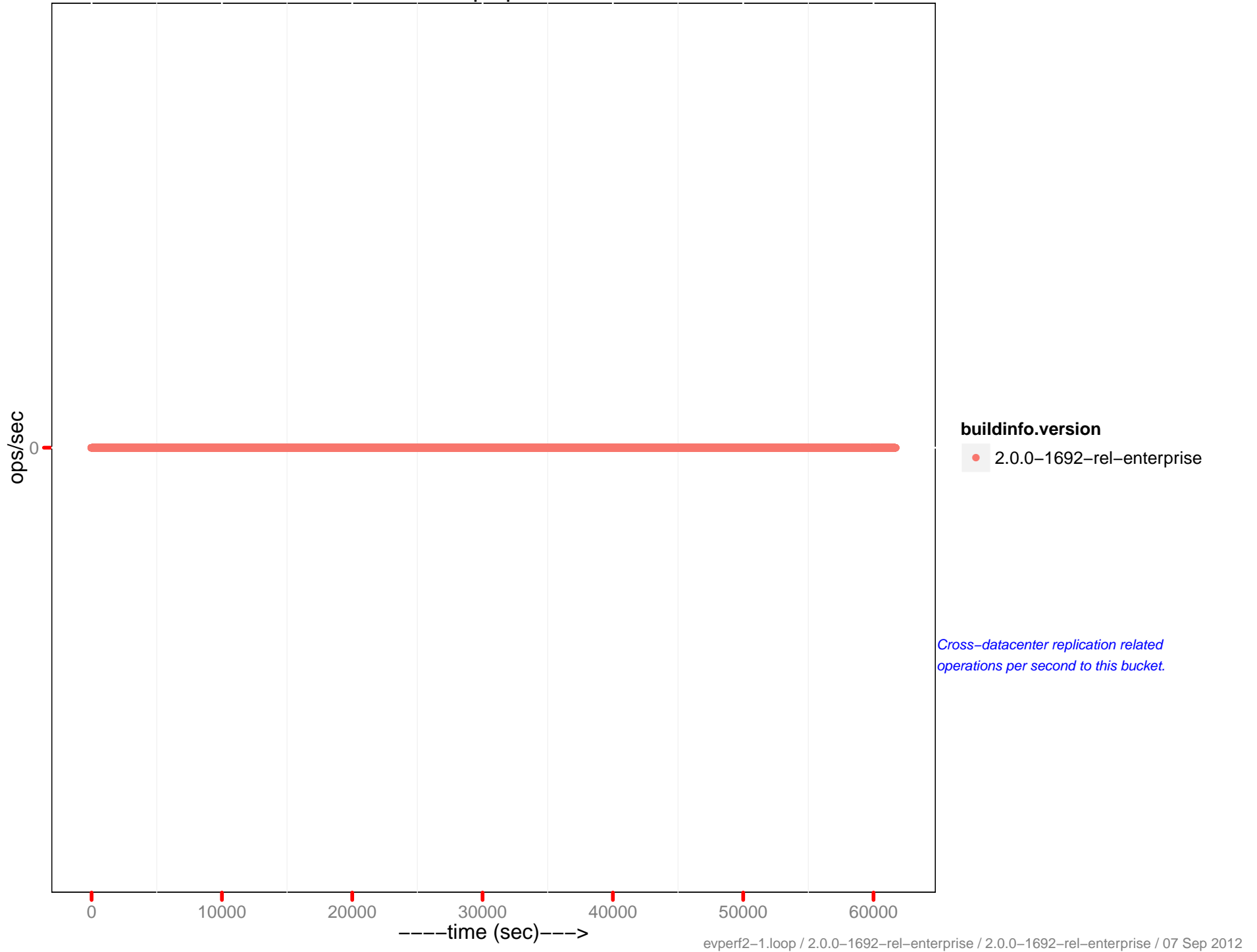


# XDC replication Queue

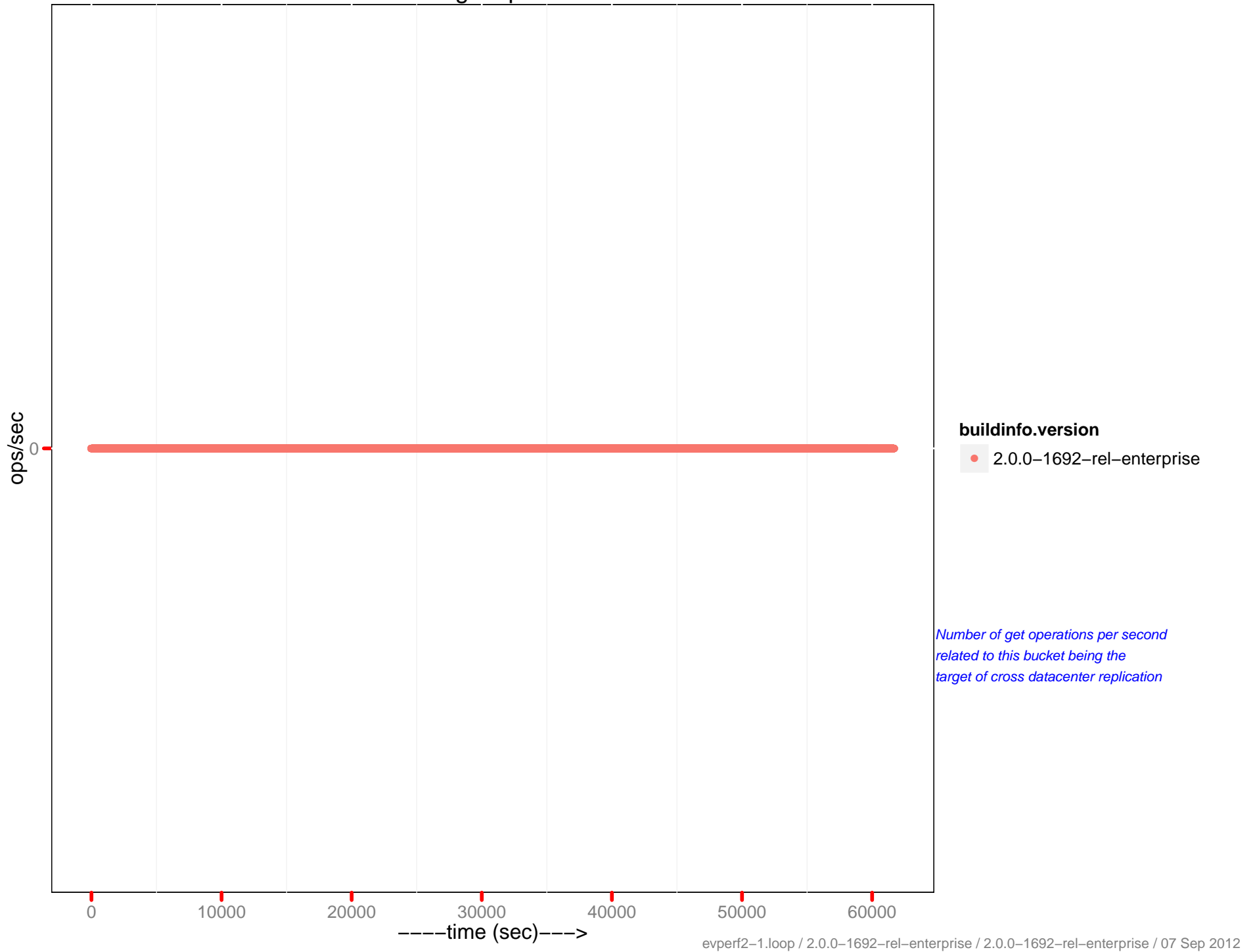




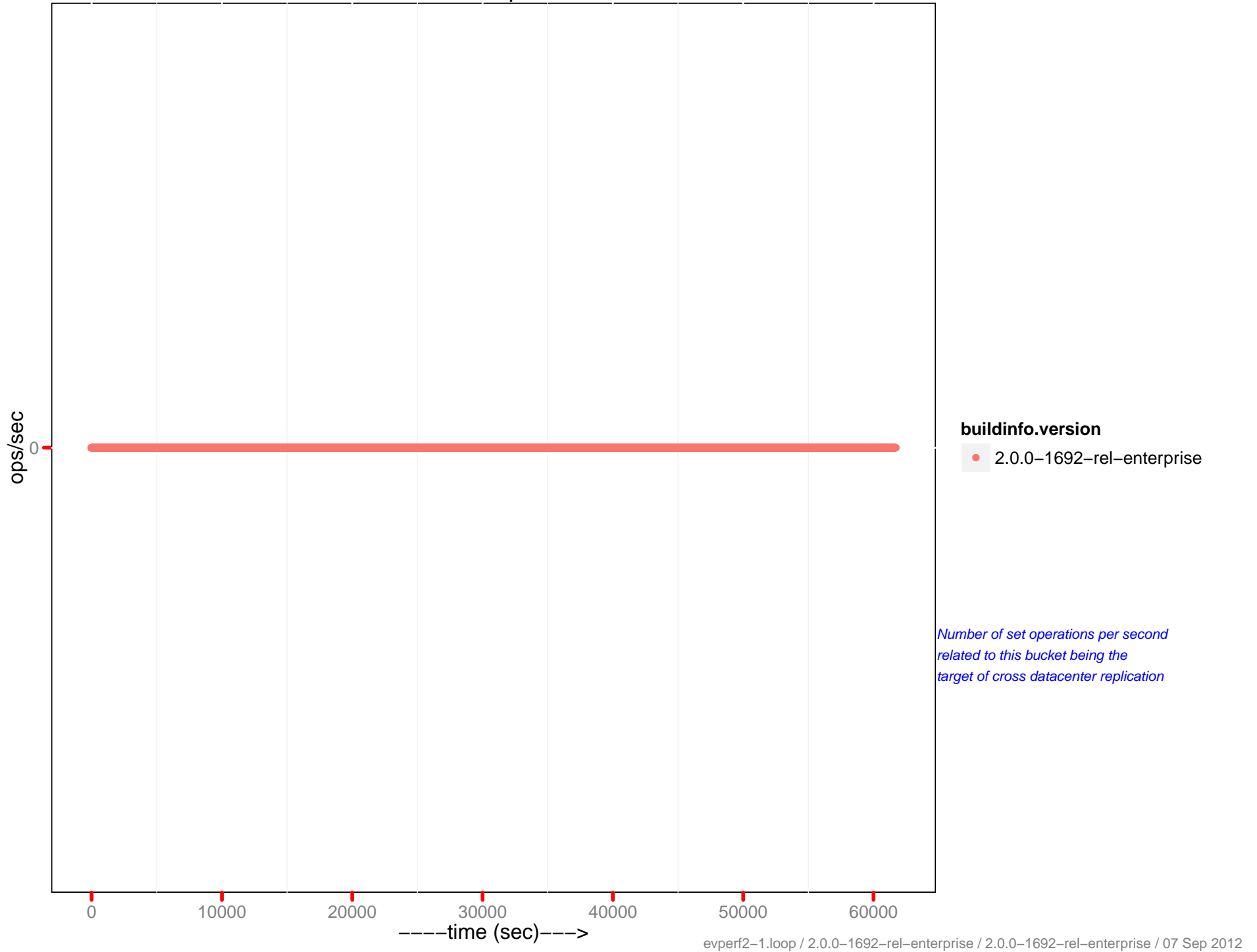
# XDC ops per sec



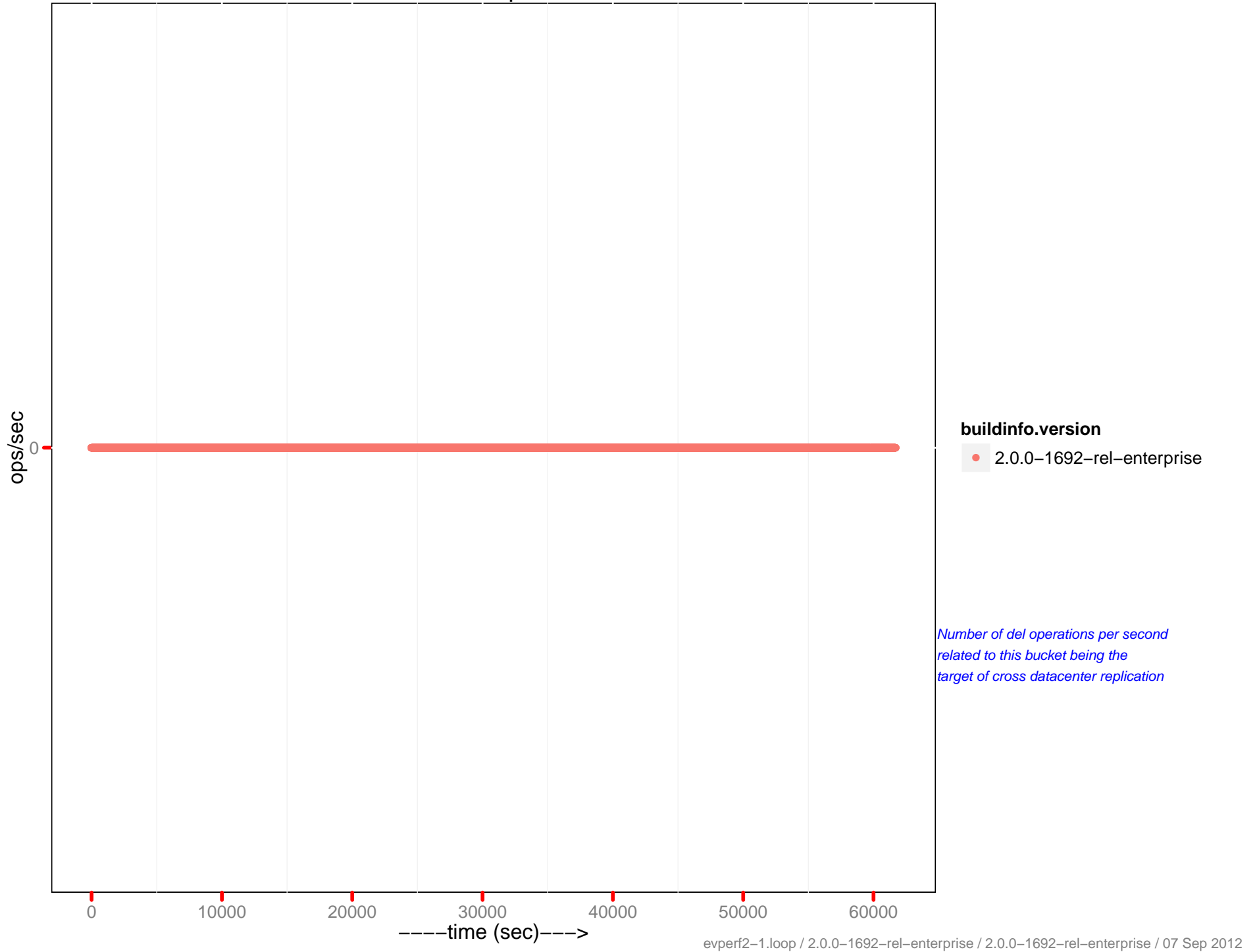
# XDC gets per sec



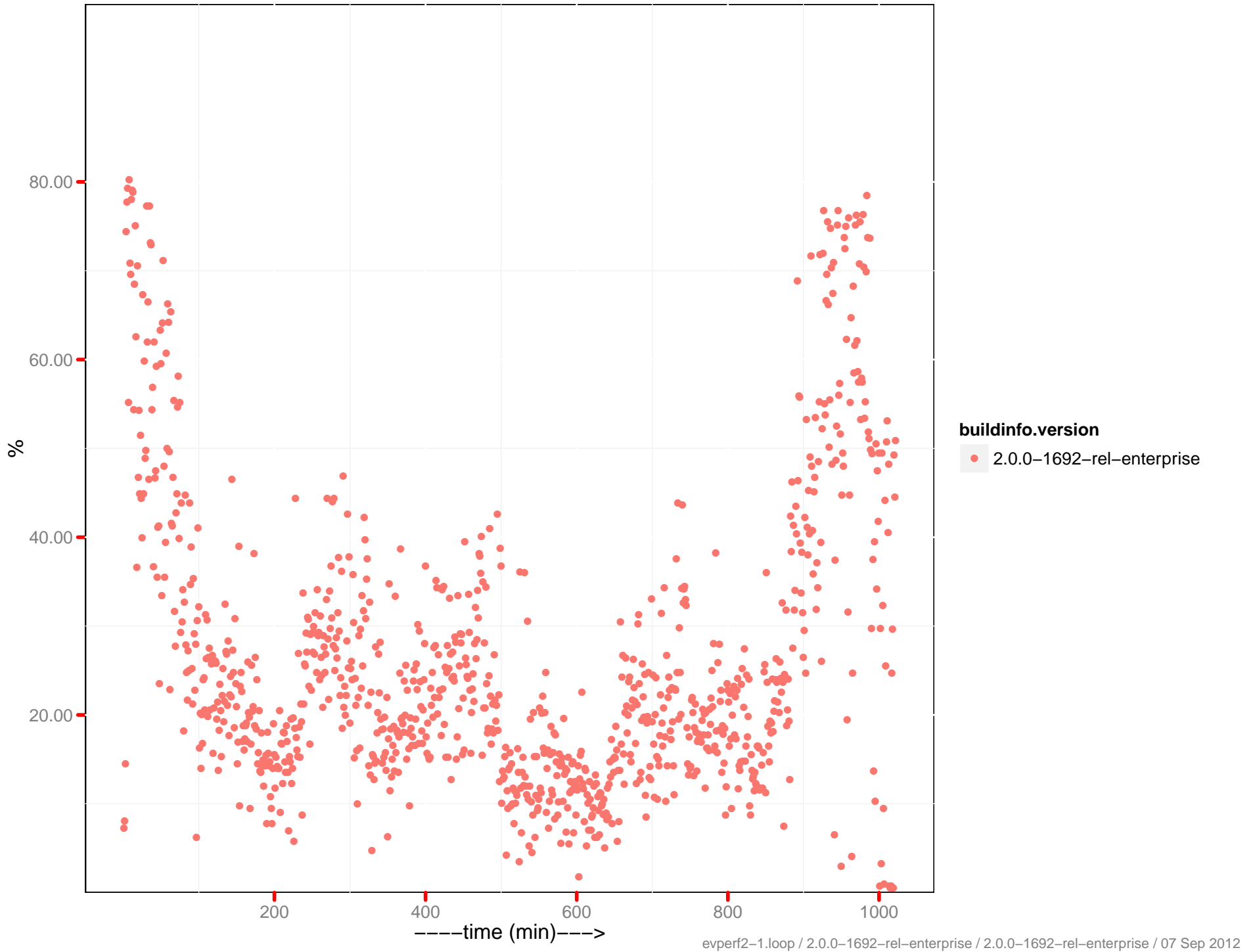
# XDC sets per sec



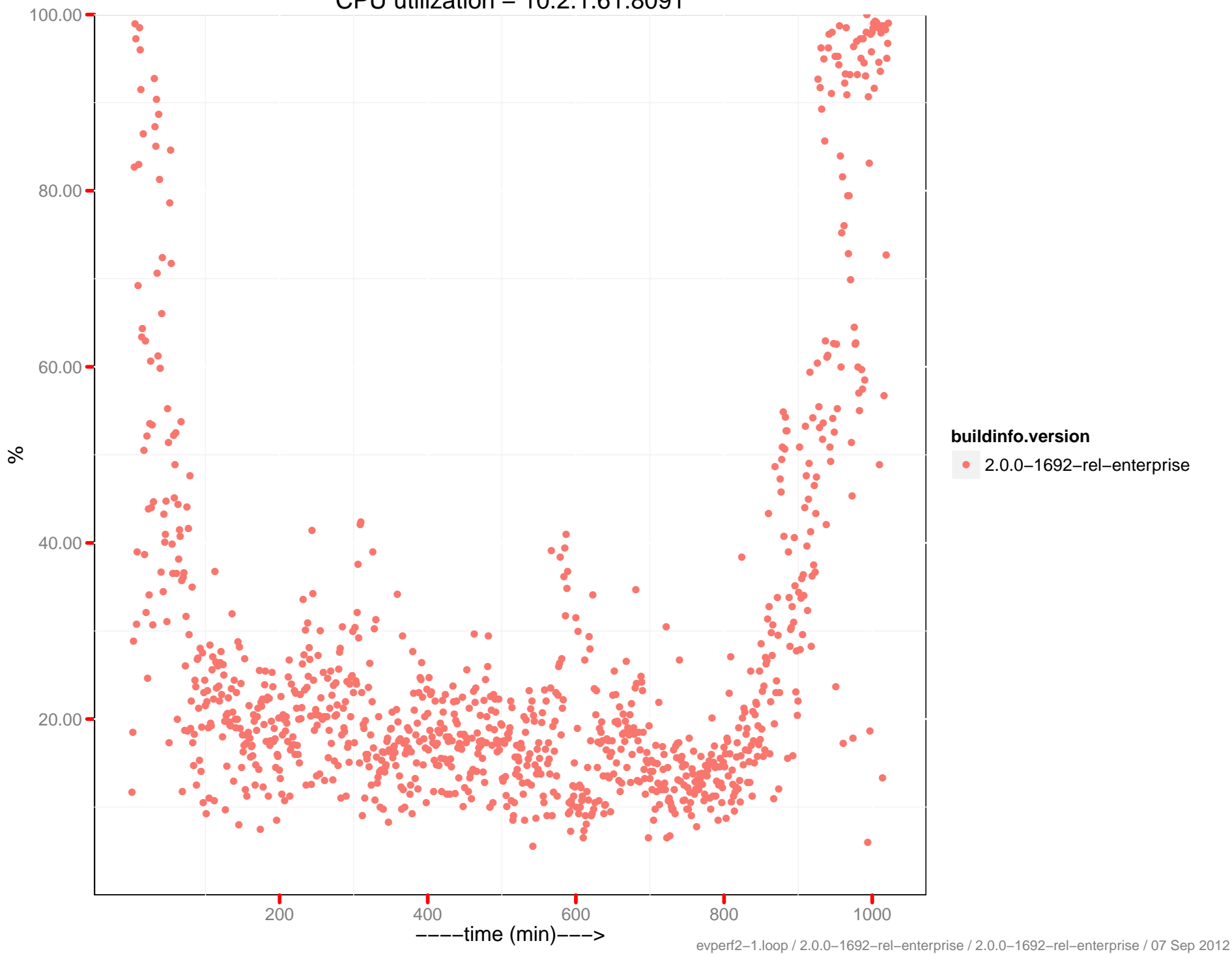
# XDC dels per sec



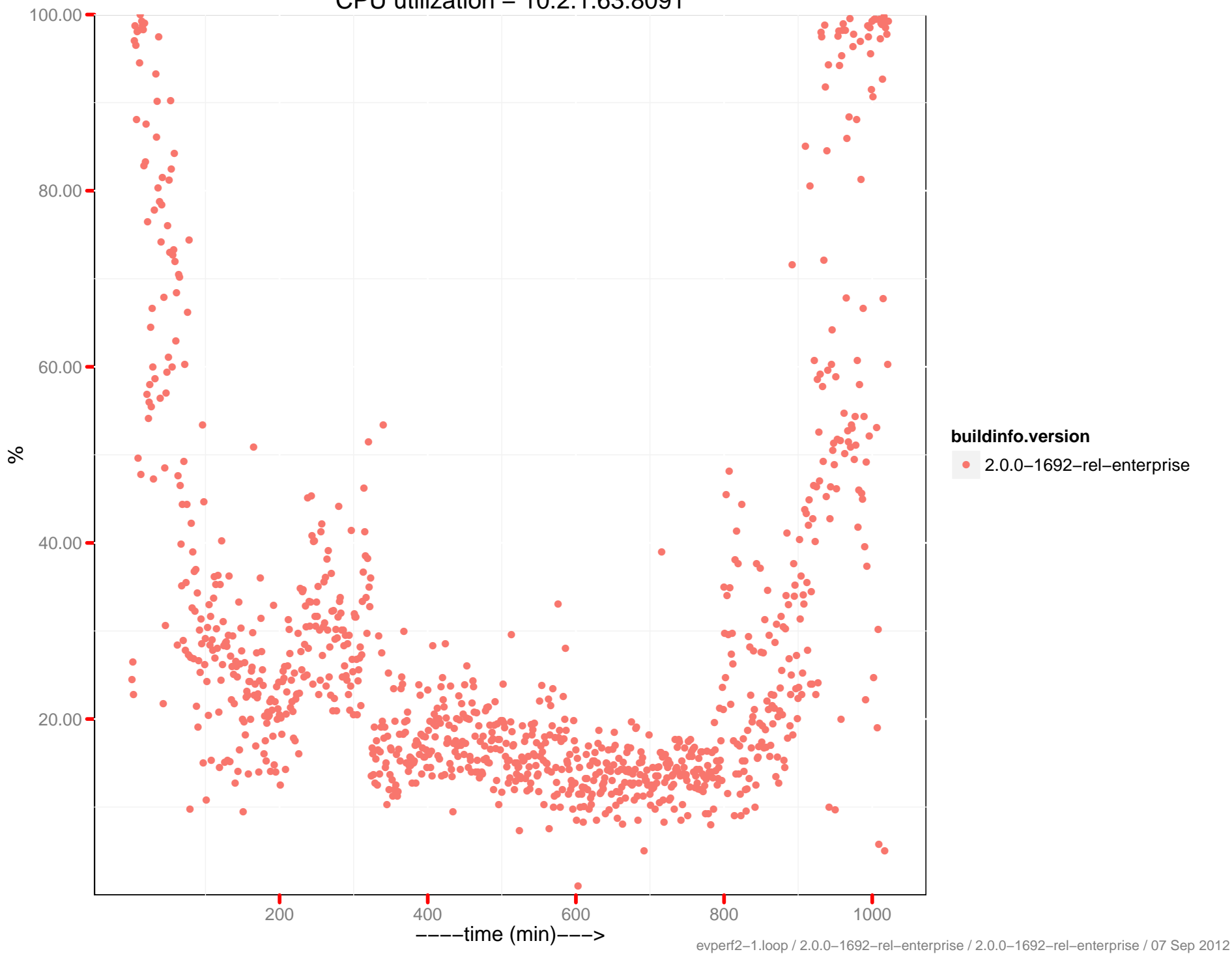
# CPU utilization – 10.2.1.58:8091



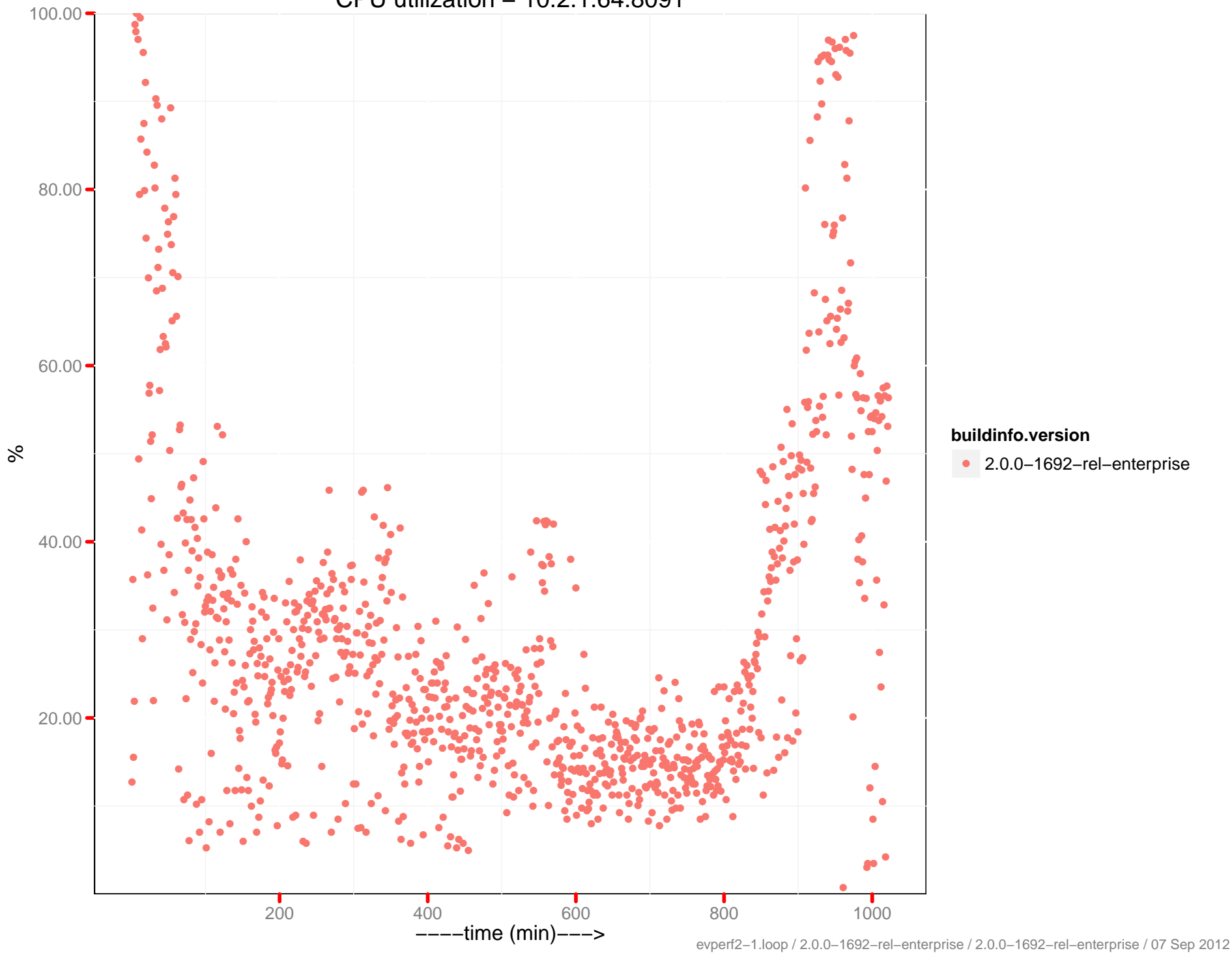
# CPU utilization – 10.2.1.61:8091



# CPU utilization – 10.2.1.63:8091

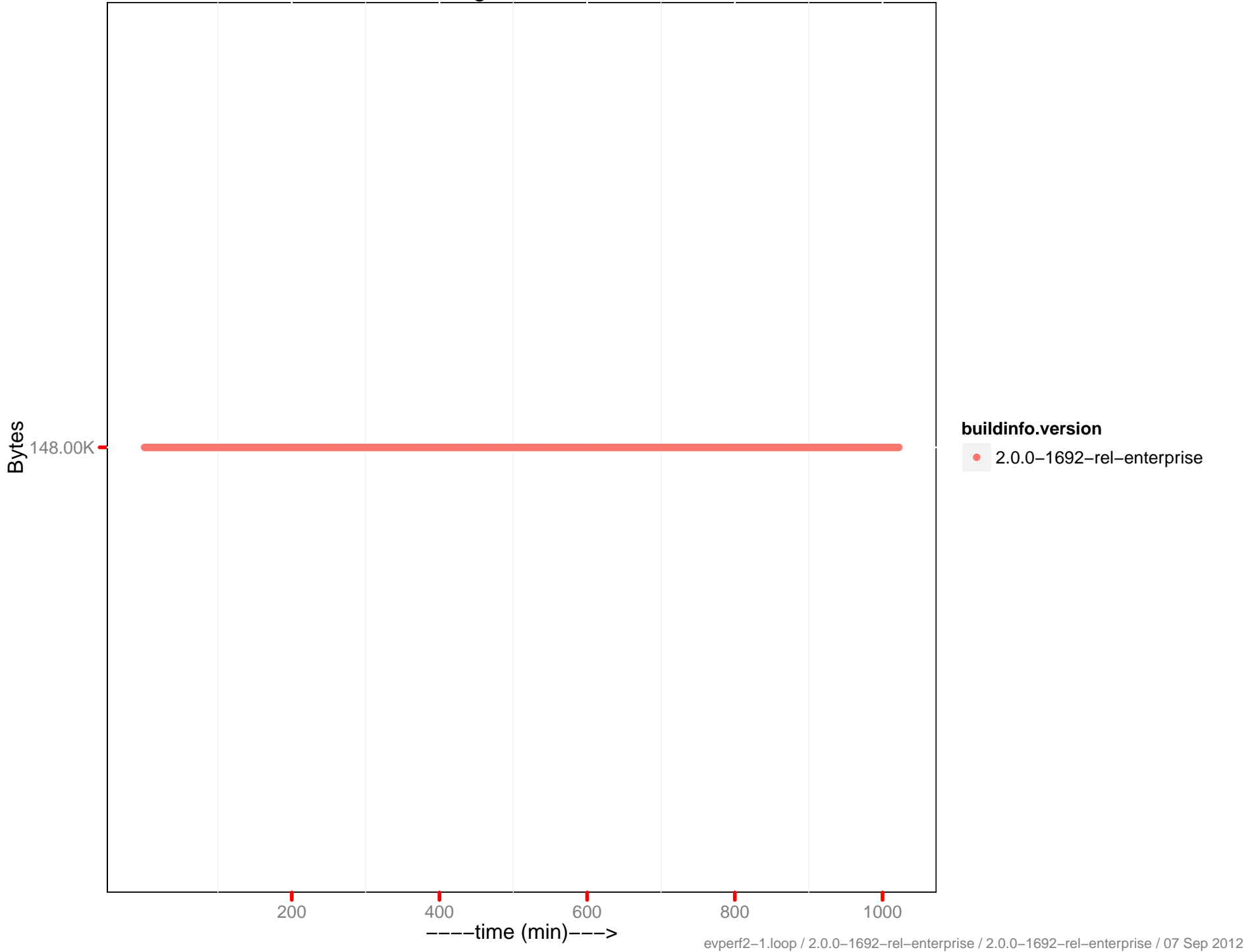


# CPU utilization – 10.2.1.64:8091

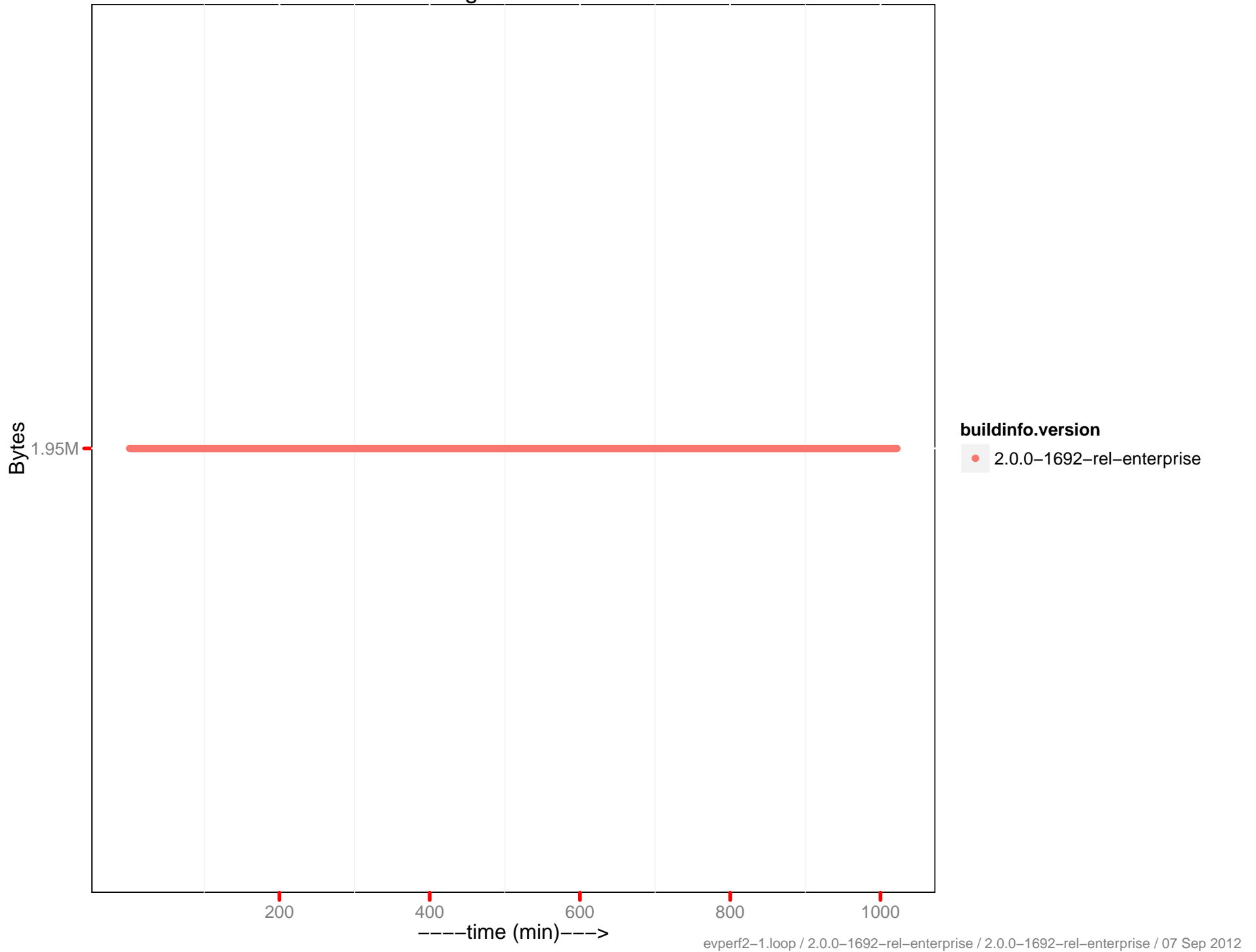




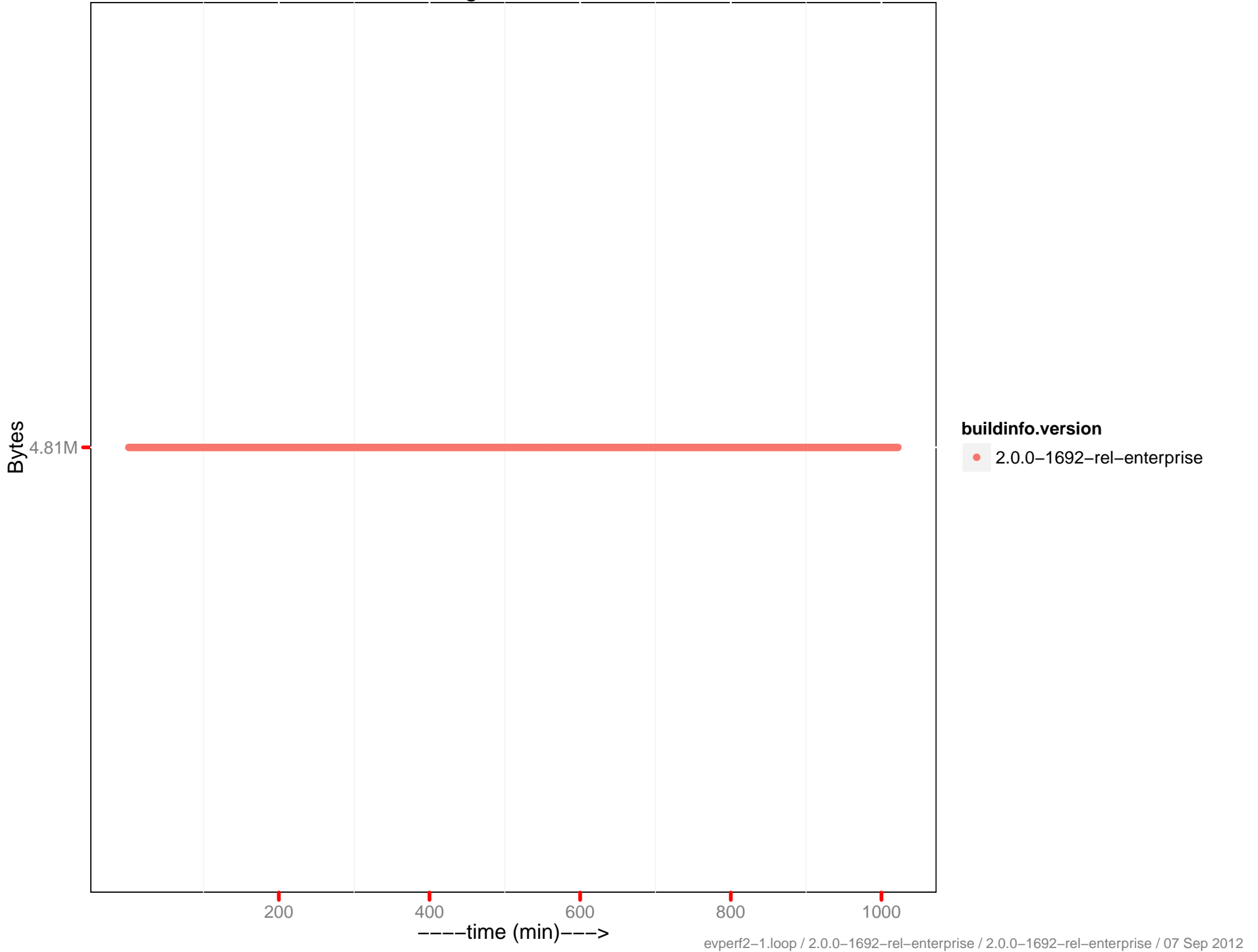
# SWAP Usage – 10.2.1.58:8091



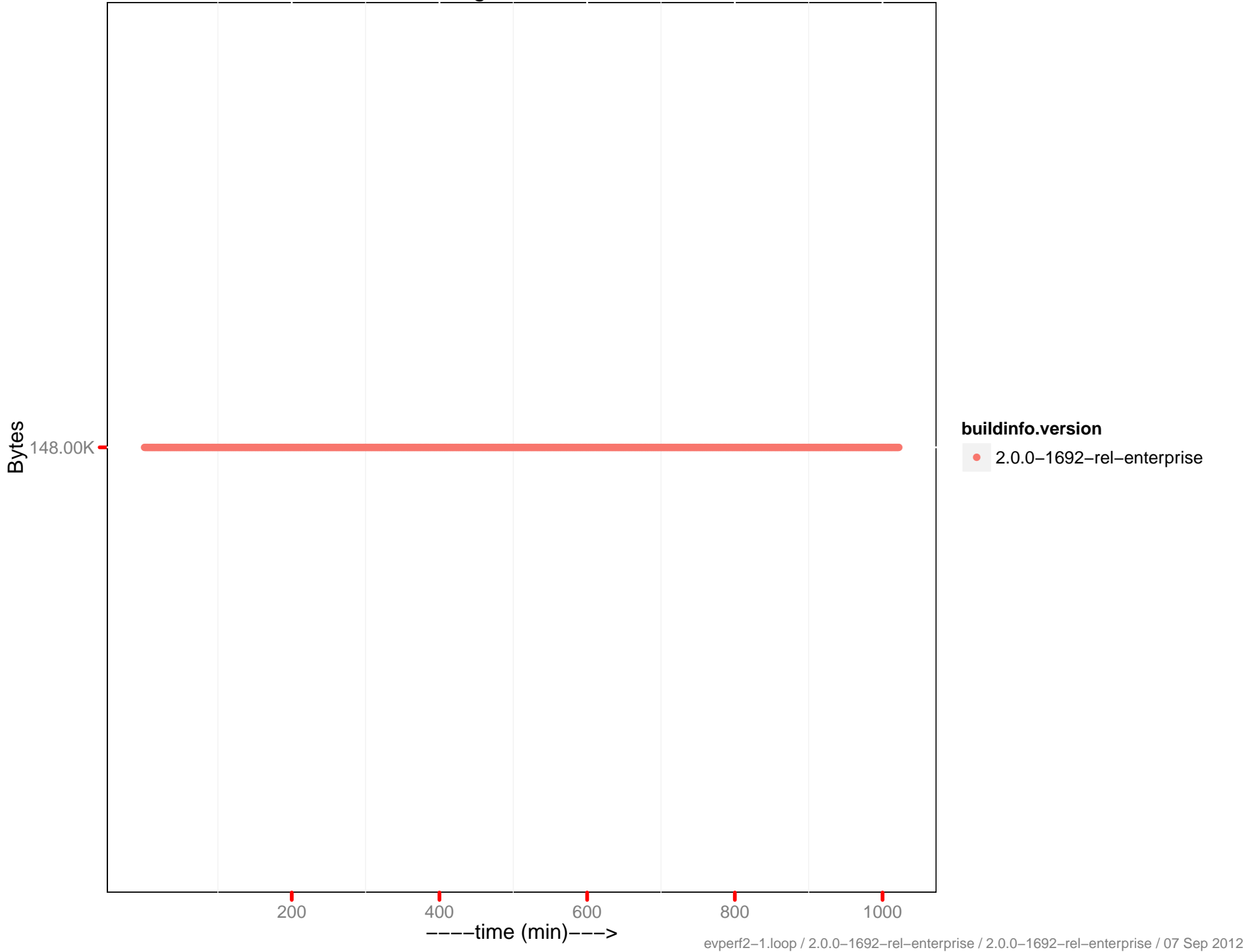
# SWAP Usage – 10.2.1.61:8091



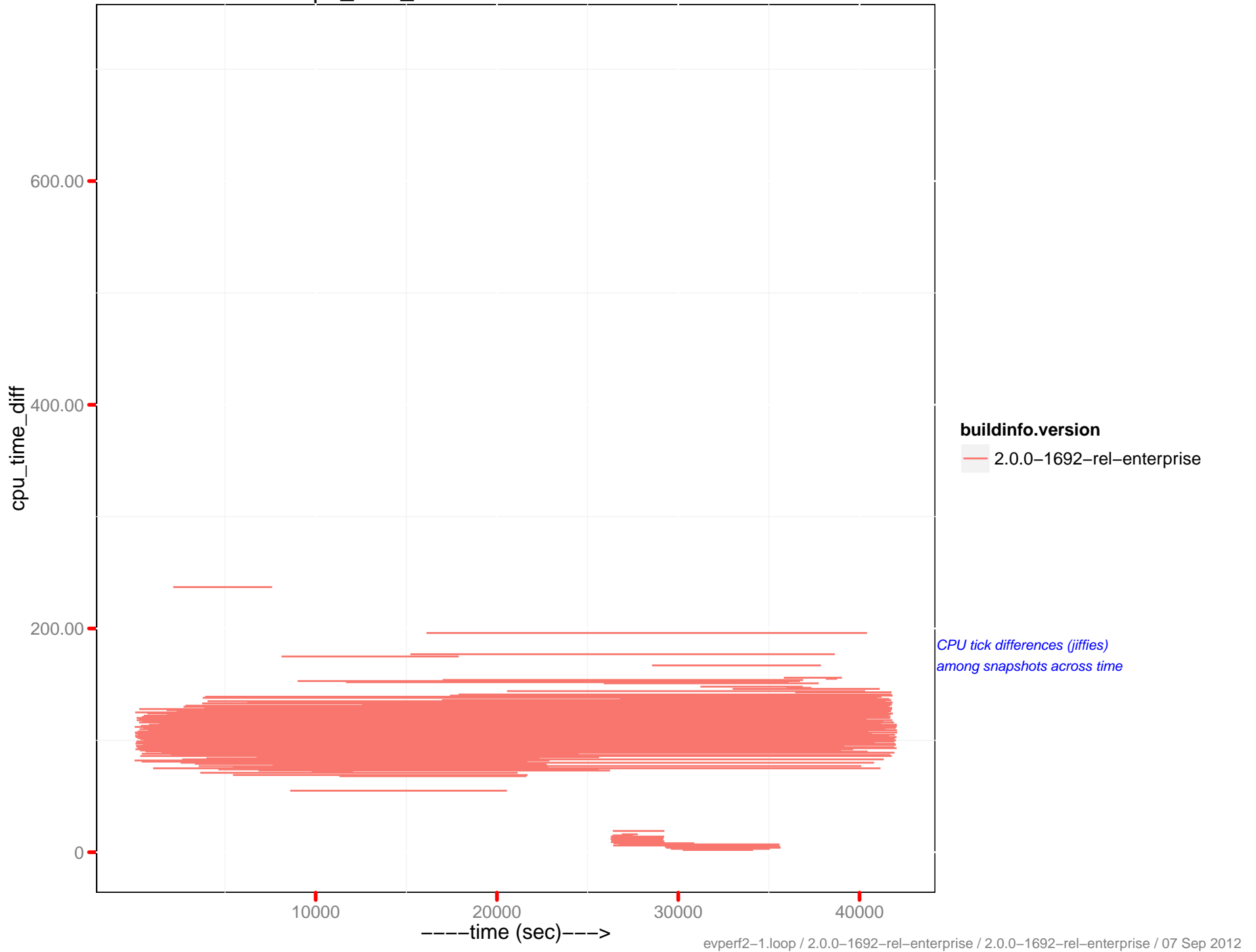
# SWAP Usage – 10.2.1.63:8091



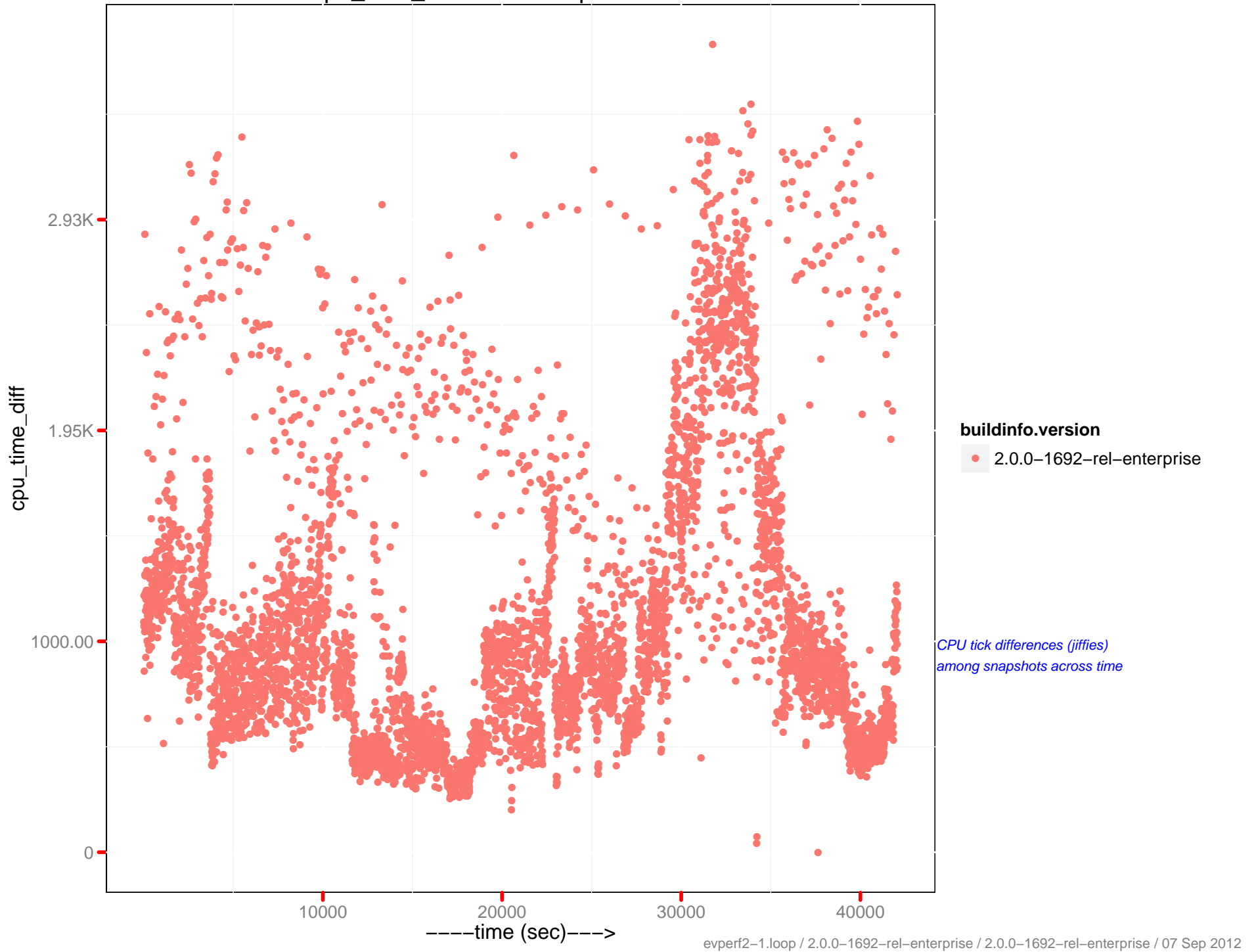
# SWAP Usage - 10.2.1.64:8091



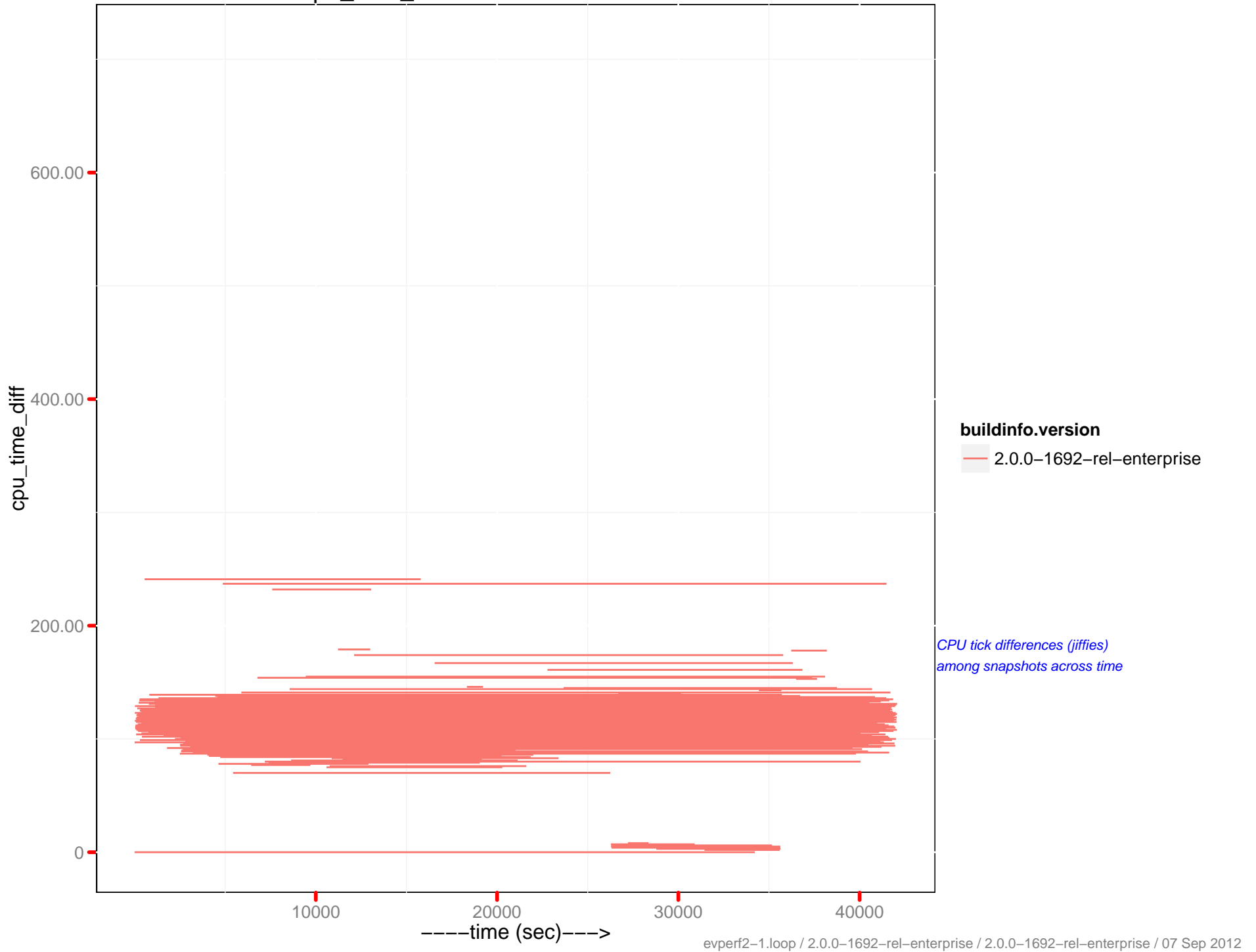
# cpu\_time\_diff: memcached - 10.2.1.58



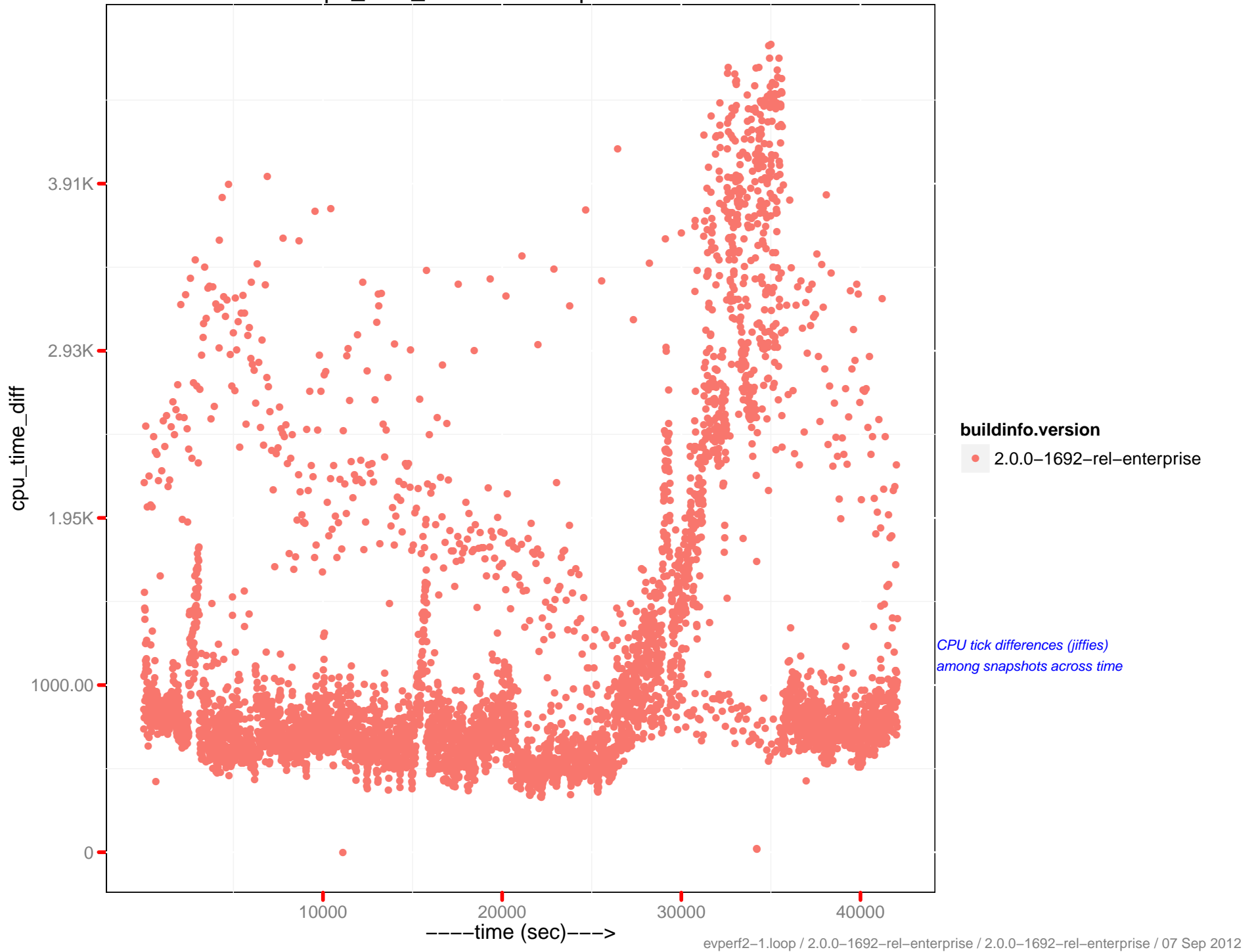
cpu\_time\_diff : beam.smp - 10.2.1.58



# cpu\_time\_diff: memcached – 10.2.1.61

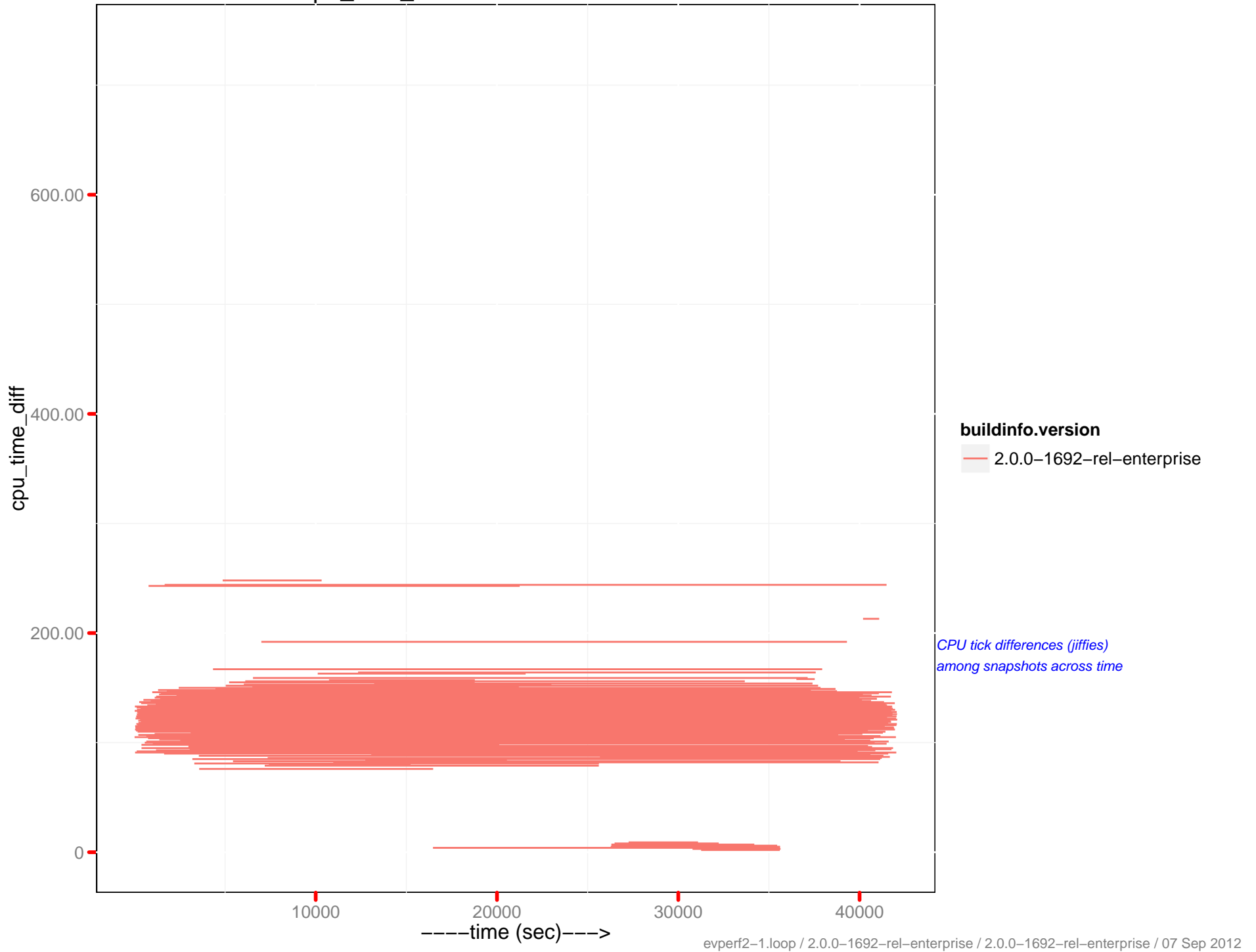


cpu\_time\_diff : beam.smp - 10.2.1.61

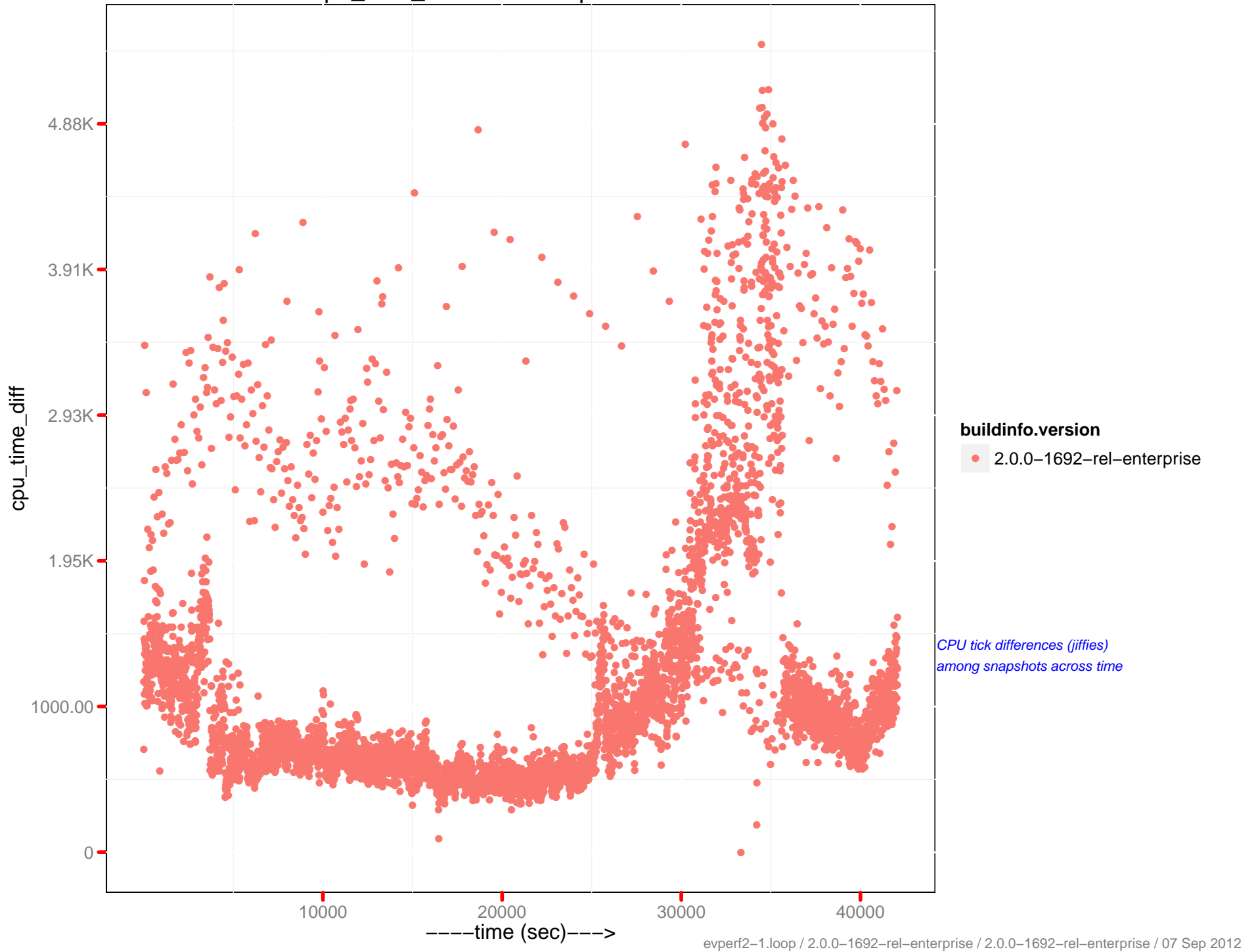




# cpu\_time\_diff: memcached - 10.2.1.63



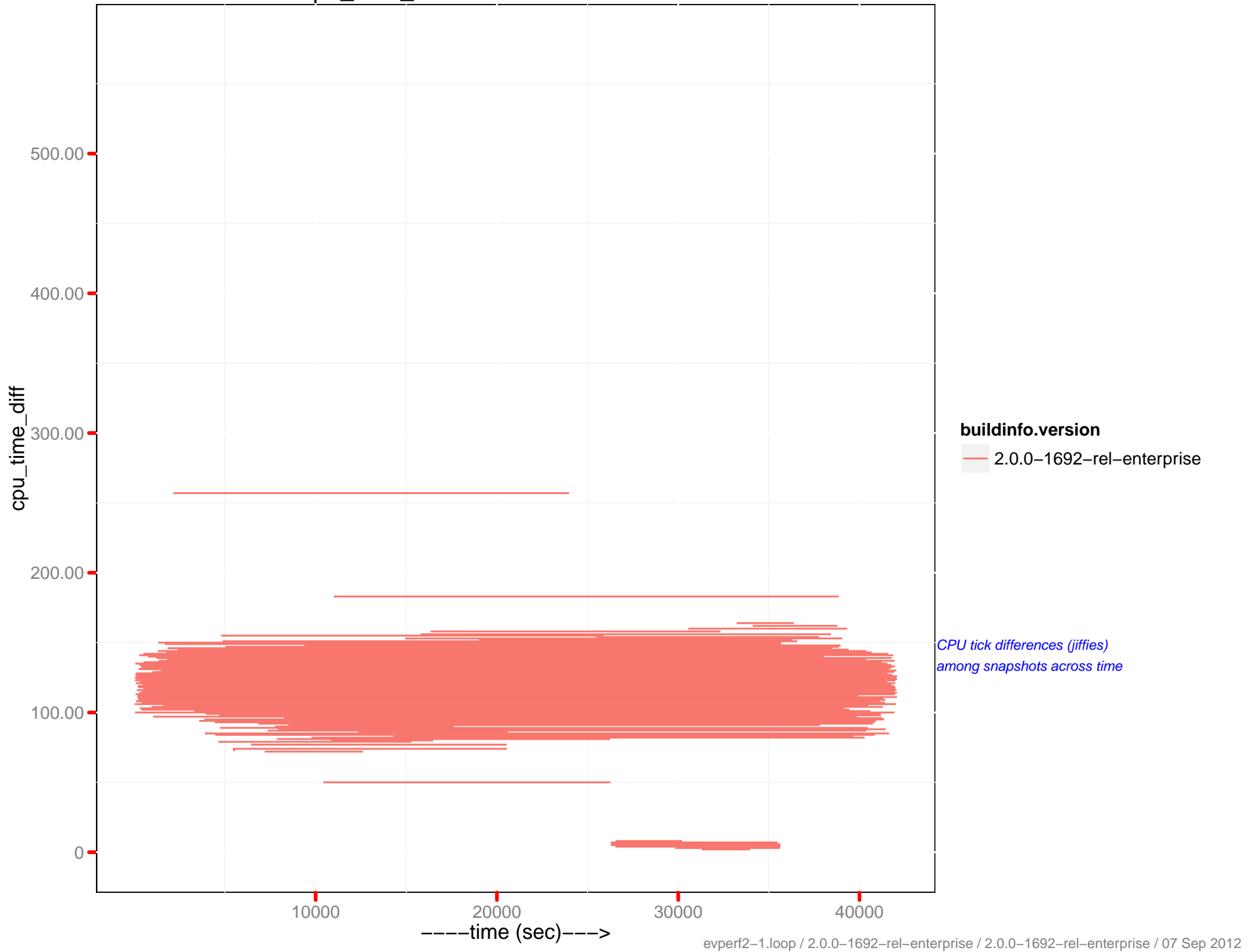
cpu\_time\_diff : beam.smp - 10.2.1.63



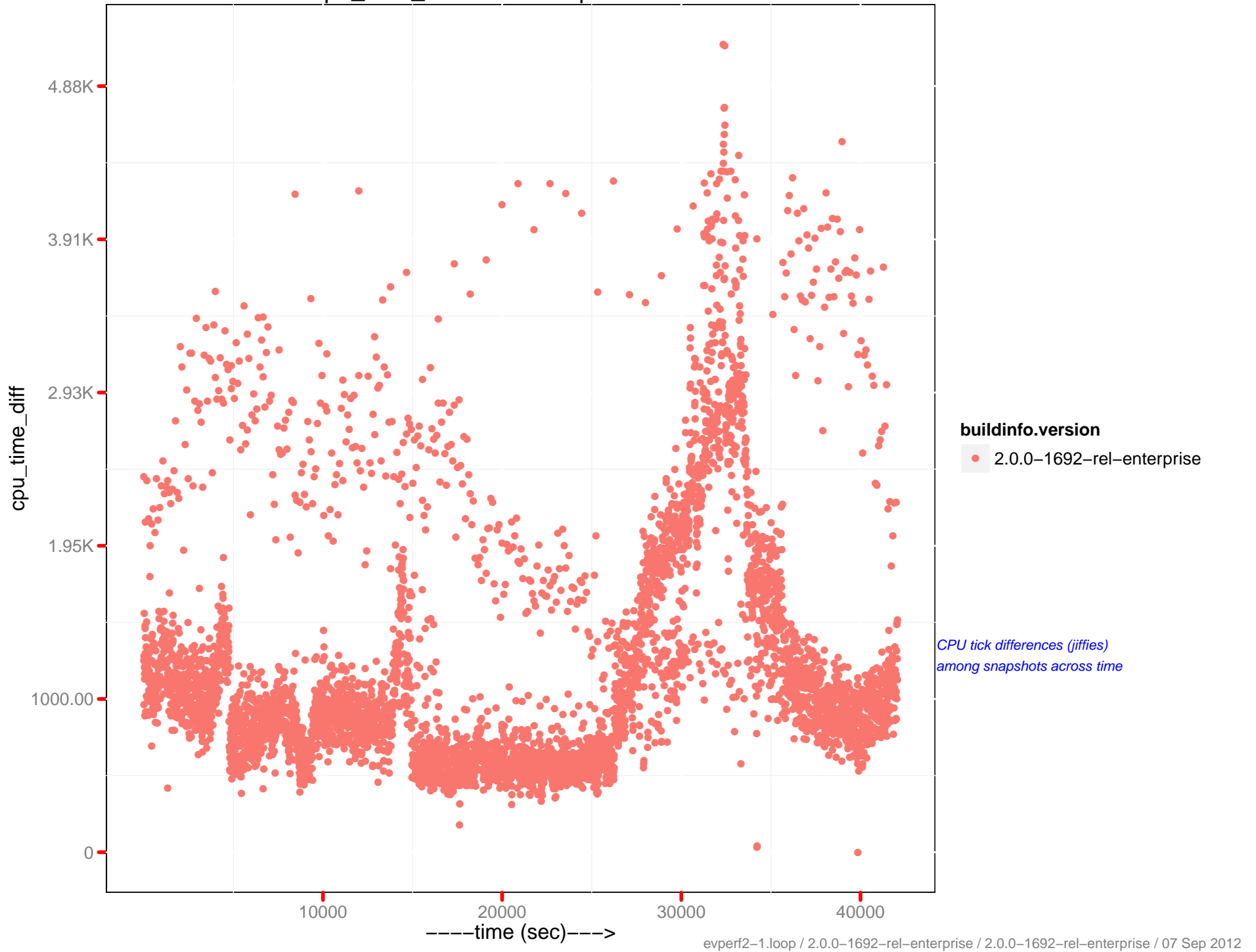
**buildinfo.version**  
● 2.0.0-1692-rel-enterprise

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

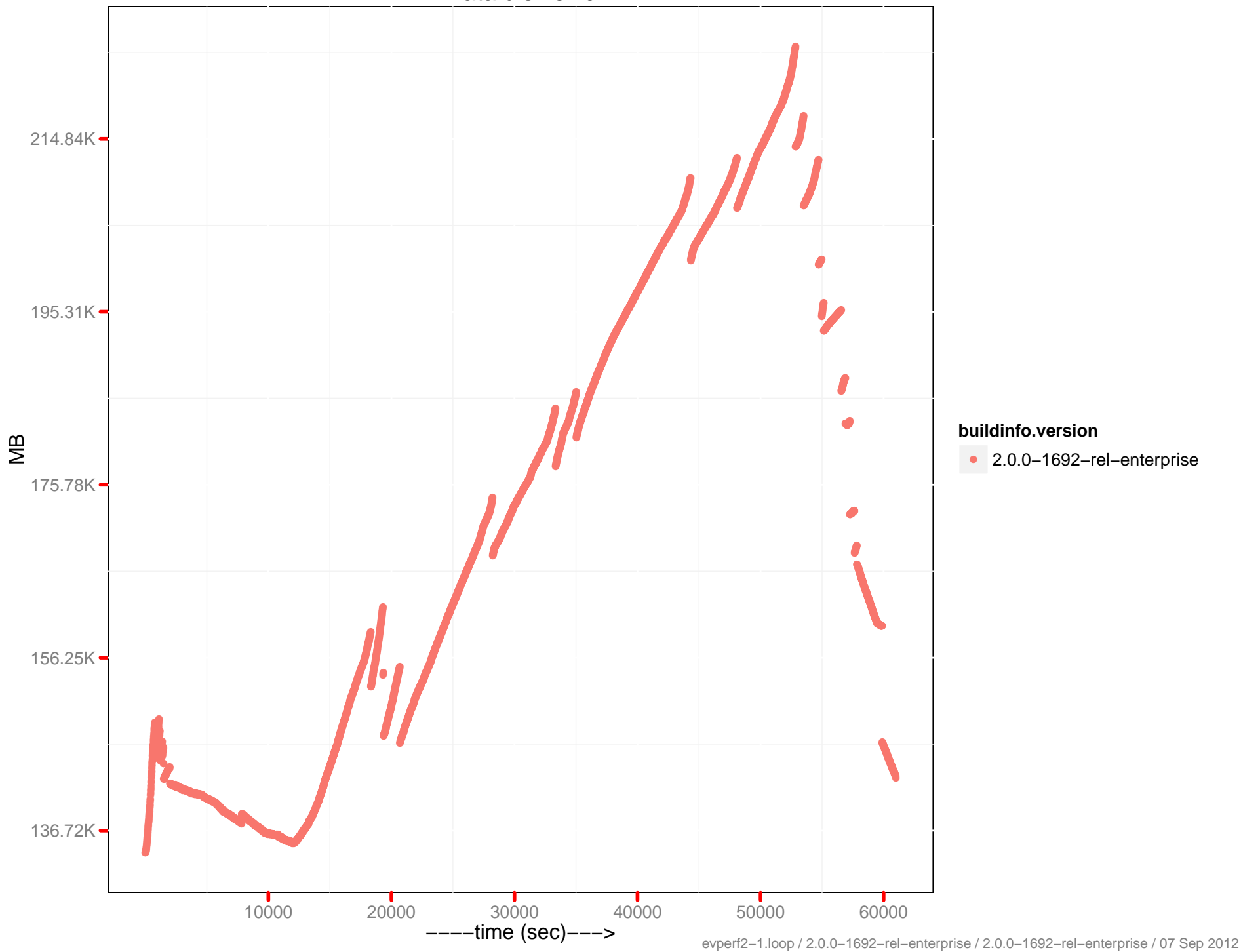
cpu\_time\_diff: memcached – 10.2.1.64



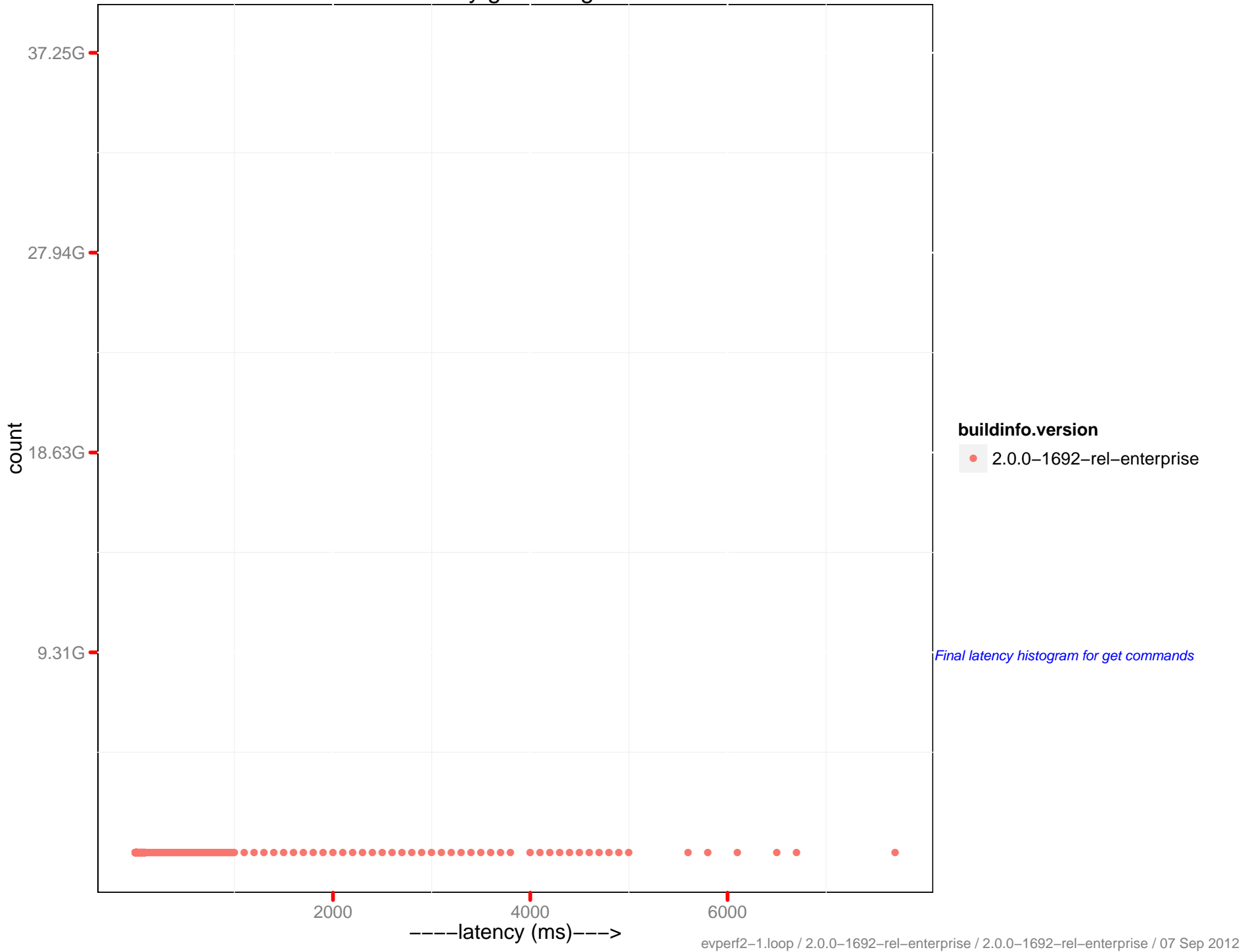
cpu\_time\_diff : beam.smp - 10.2.1.64



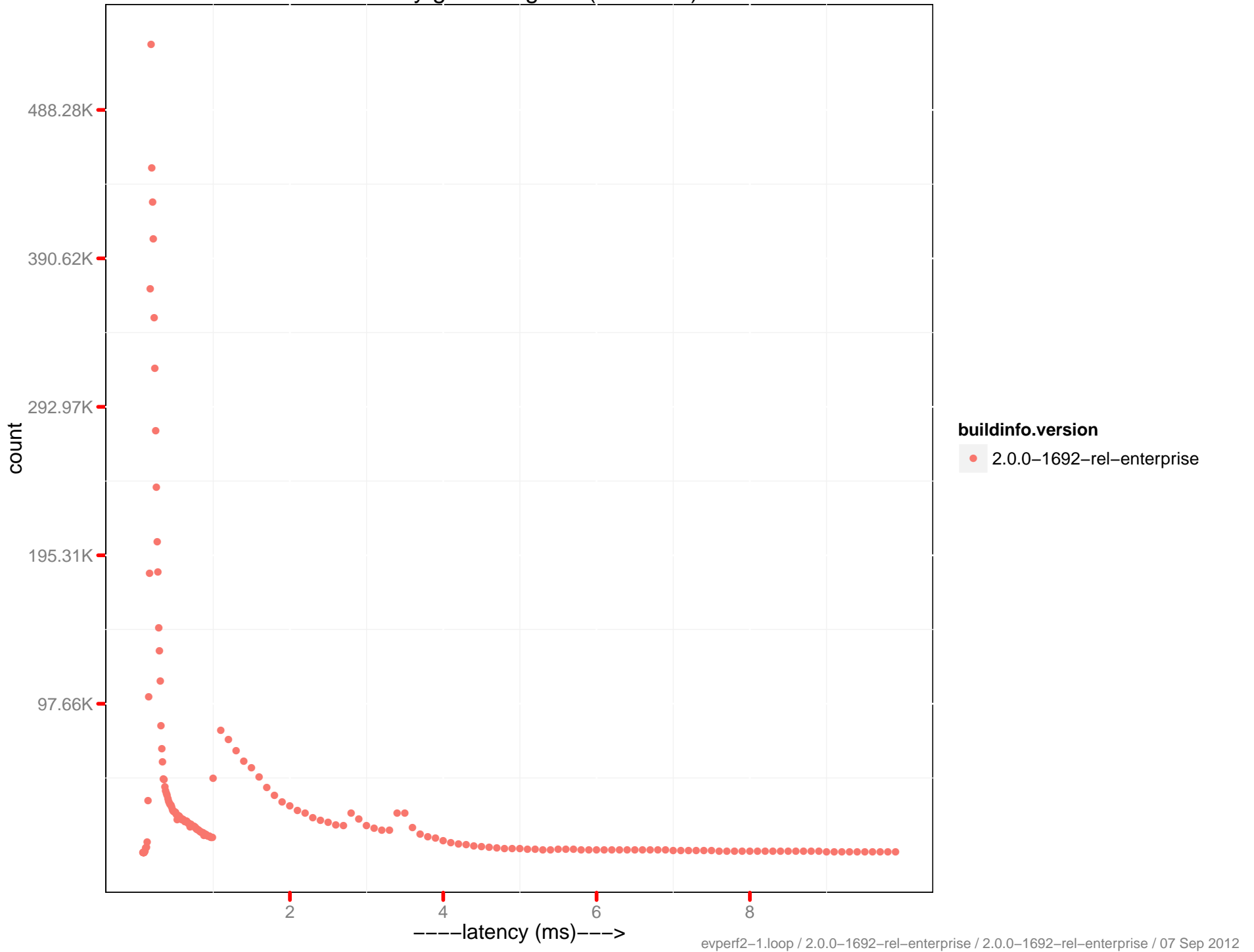
Data disk size



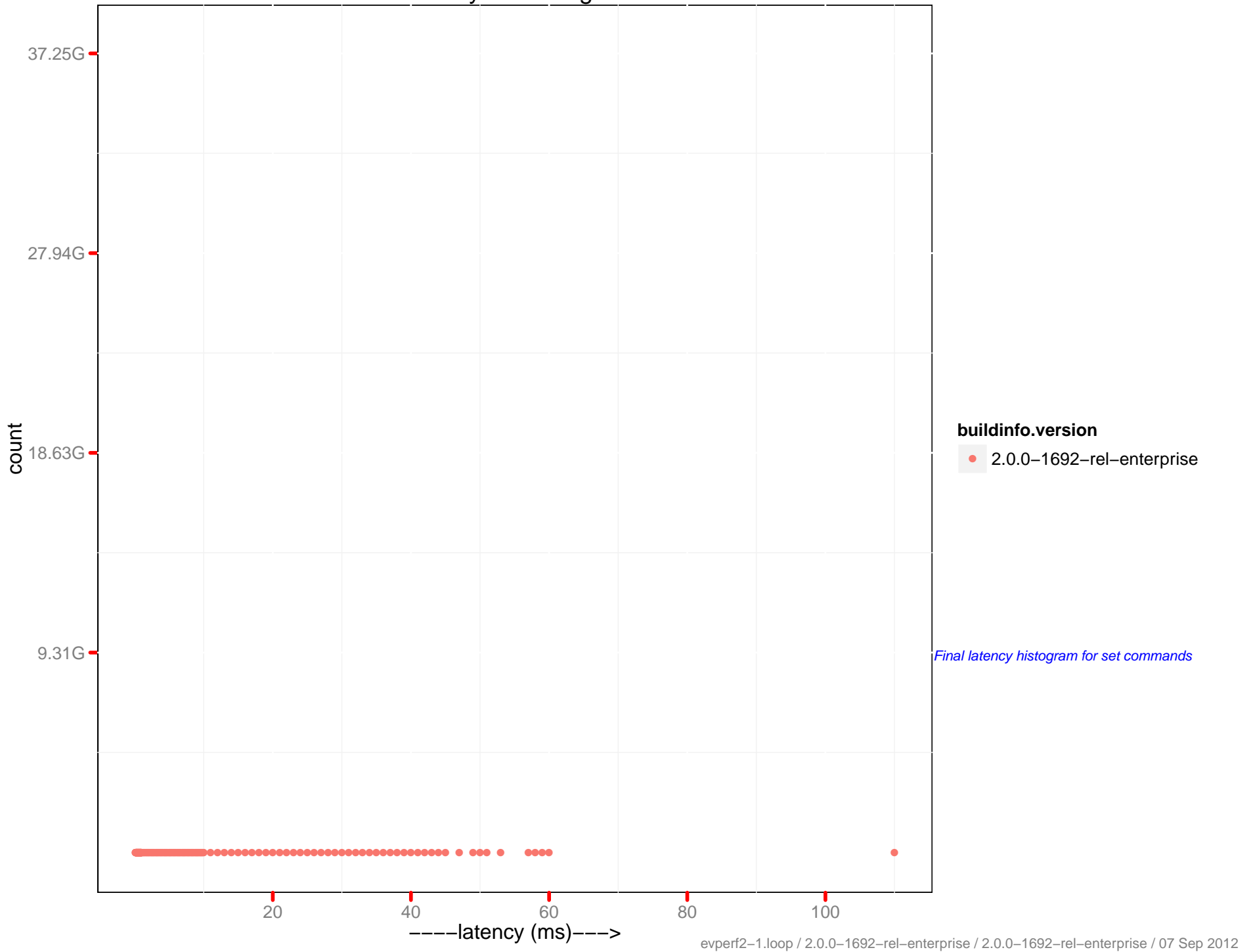
# Latency get histogram



Latency get histogram (0–10 ms)

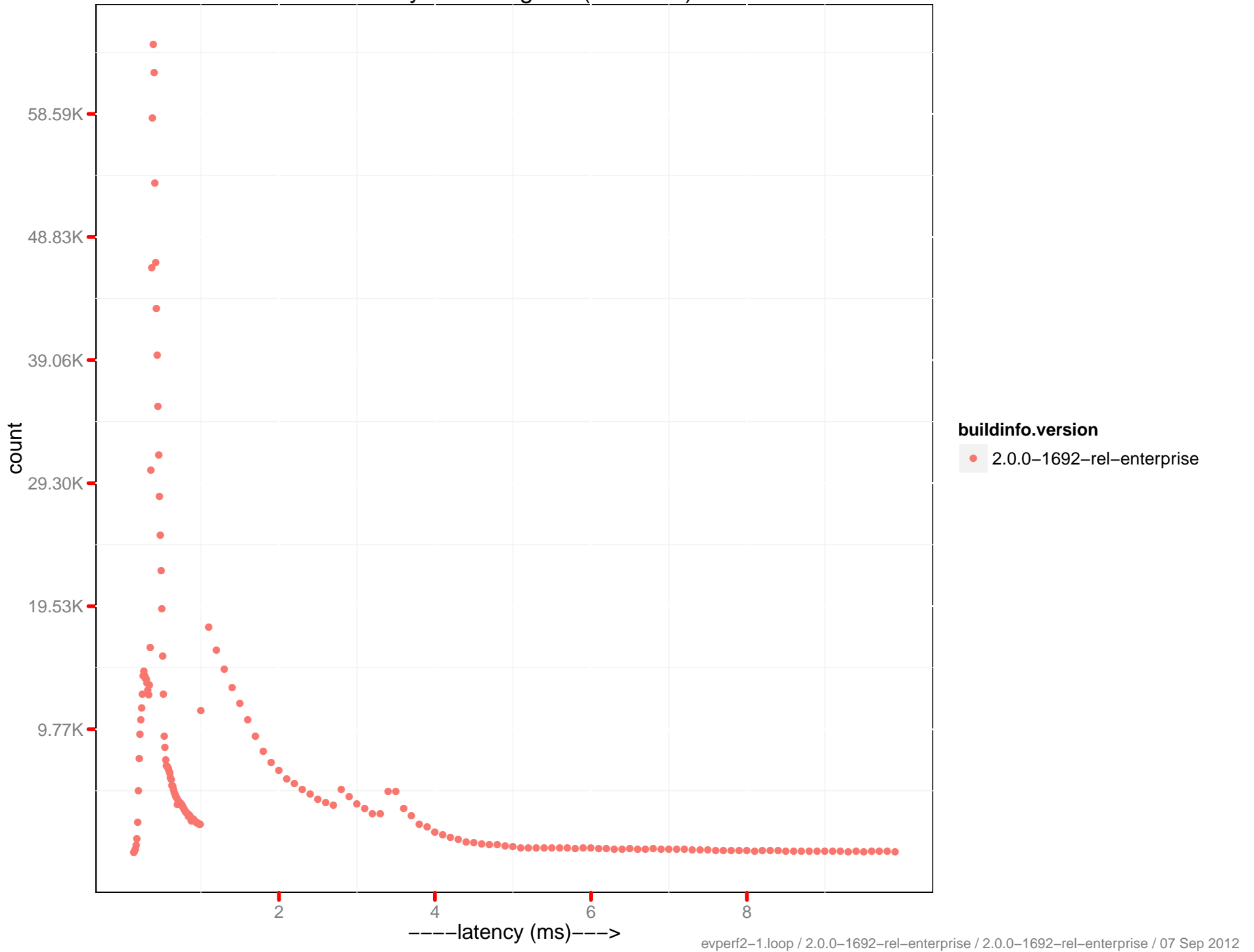


# Latency set histogram

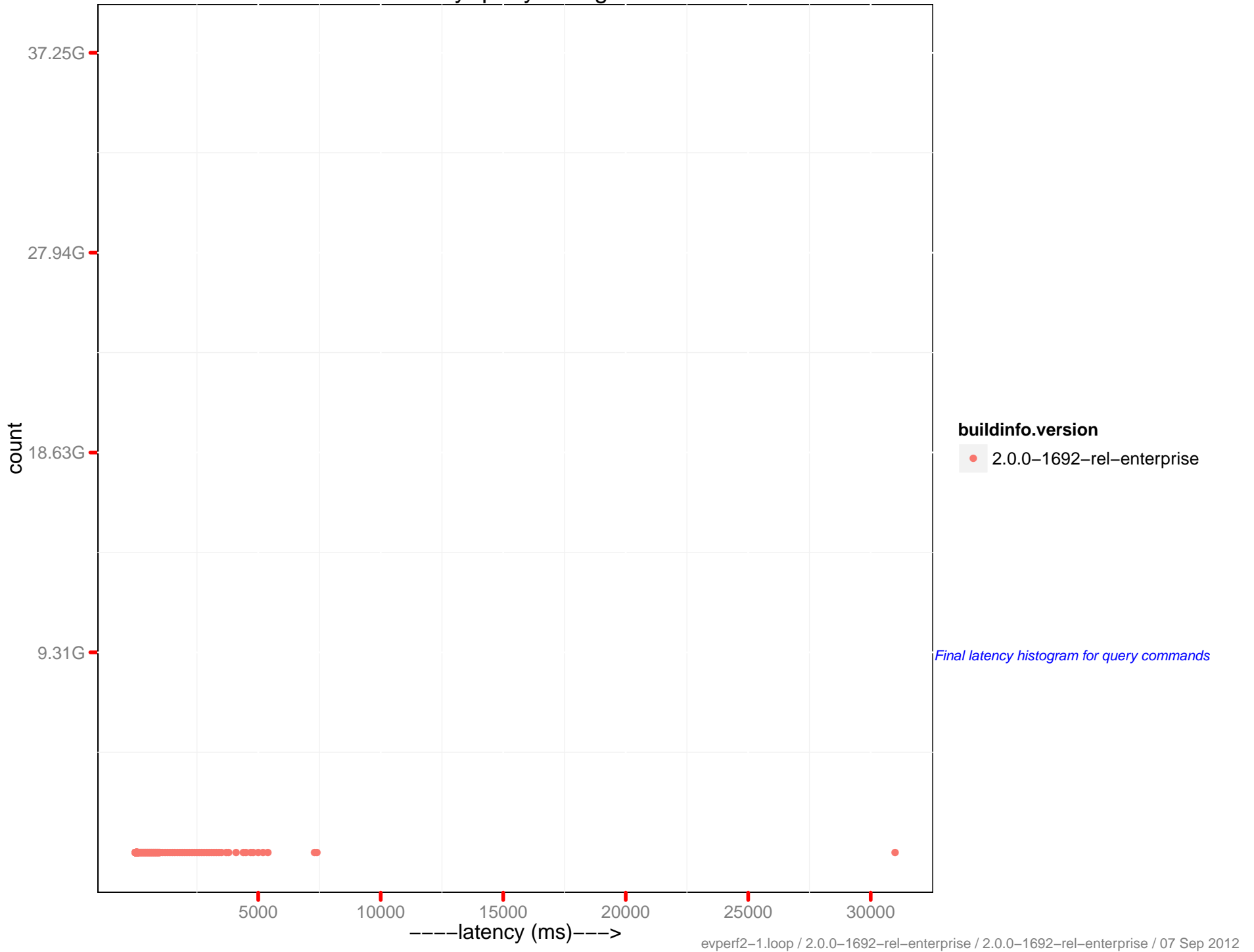




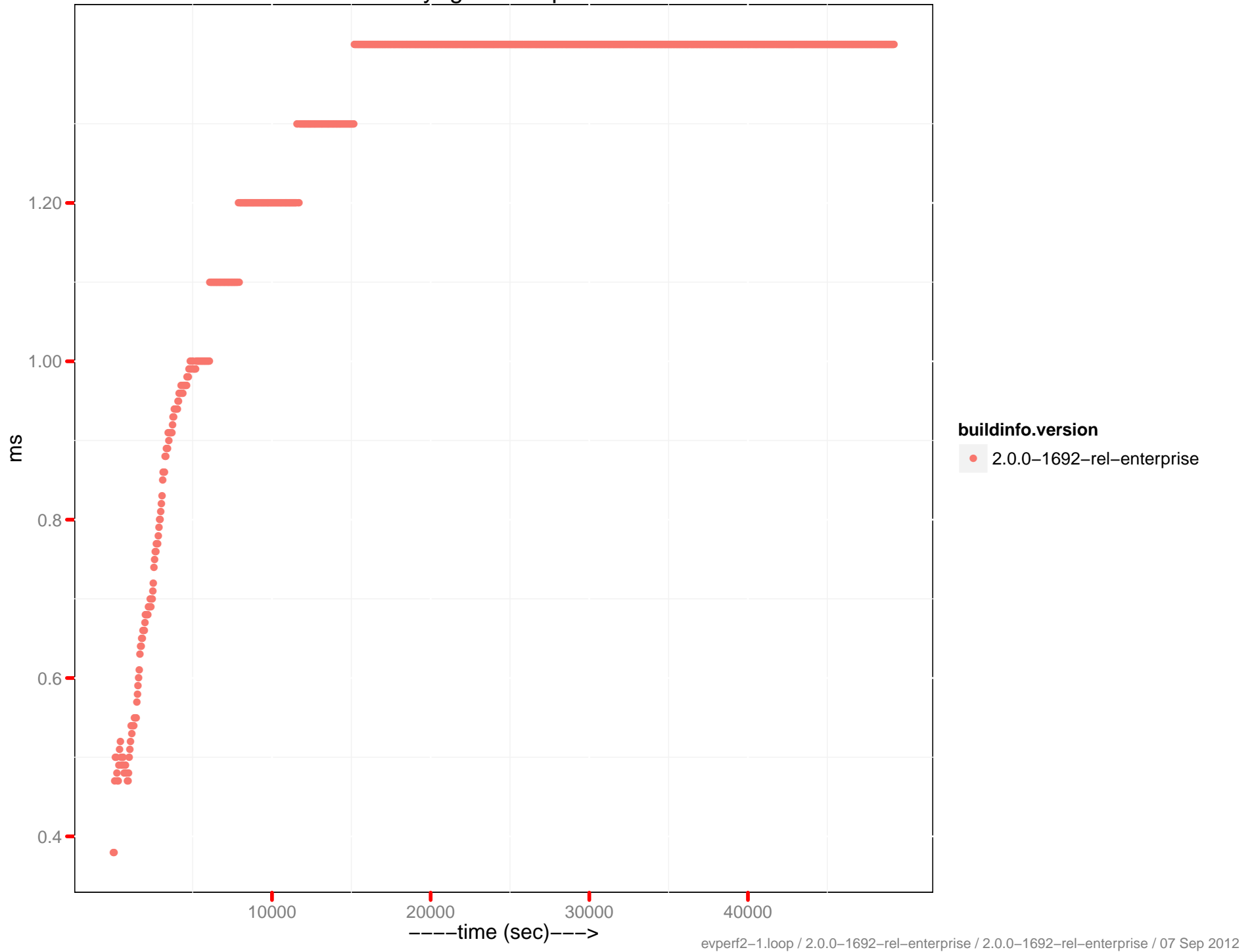
Latency set histogram (0–10 ms)



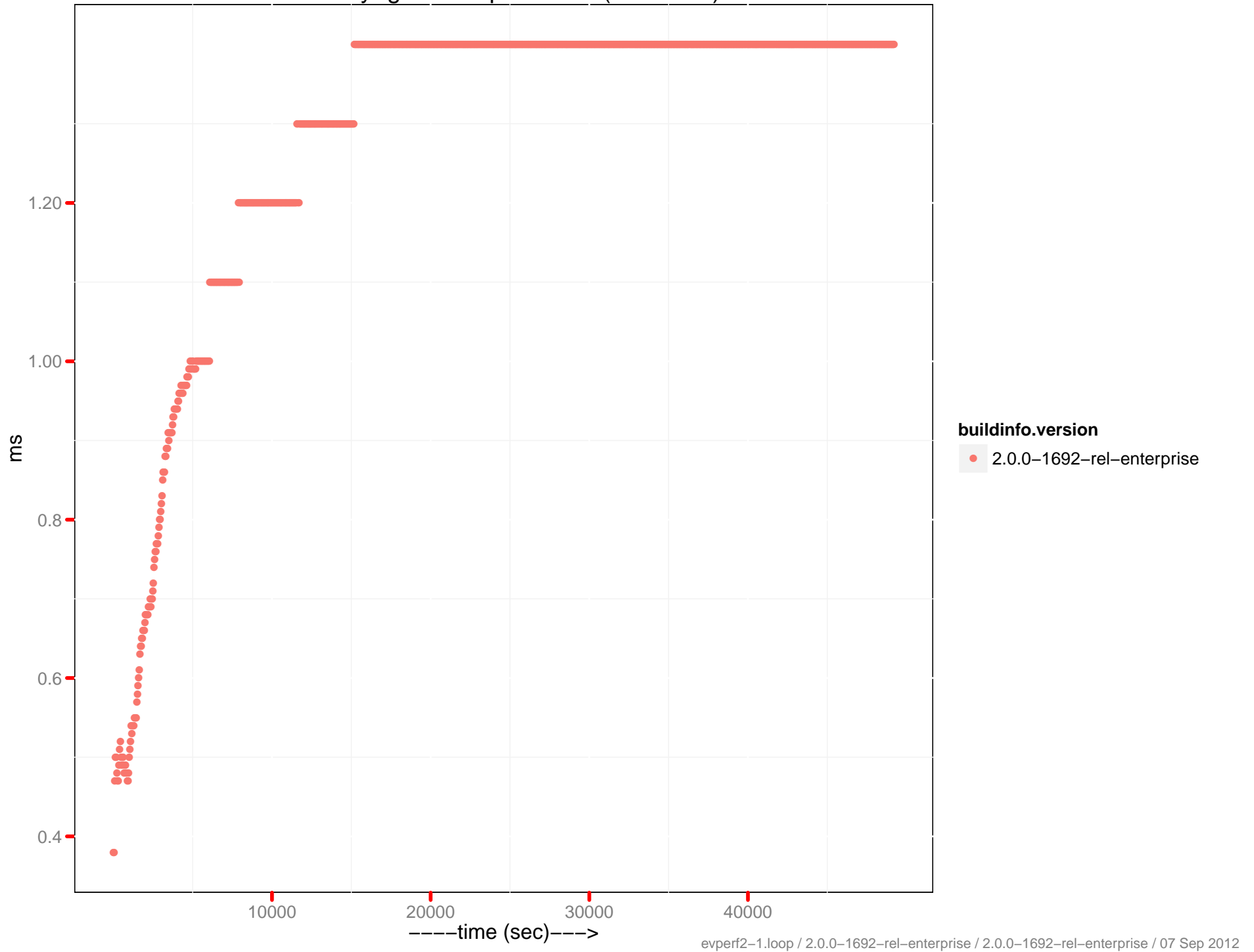
# Latency query histogram



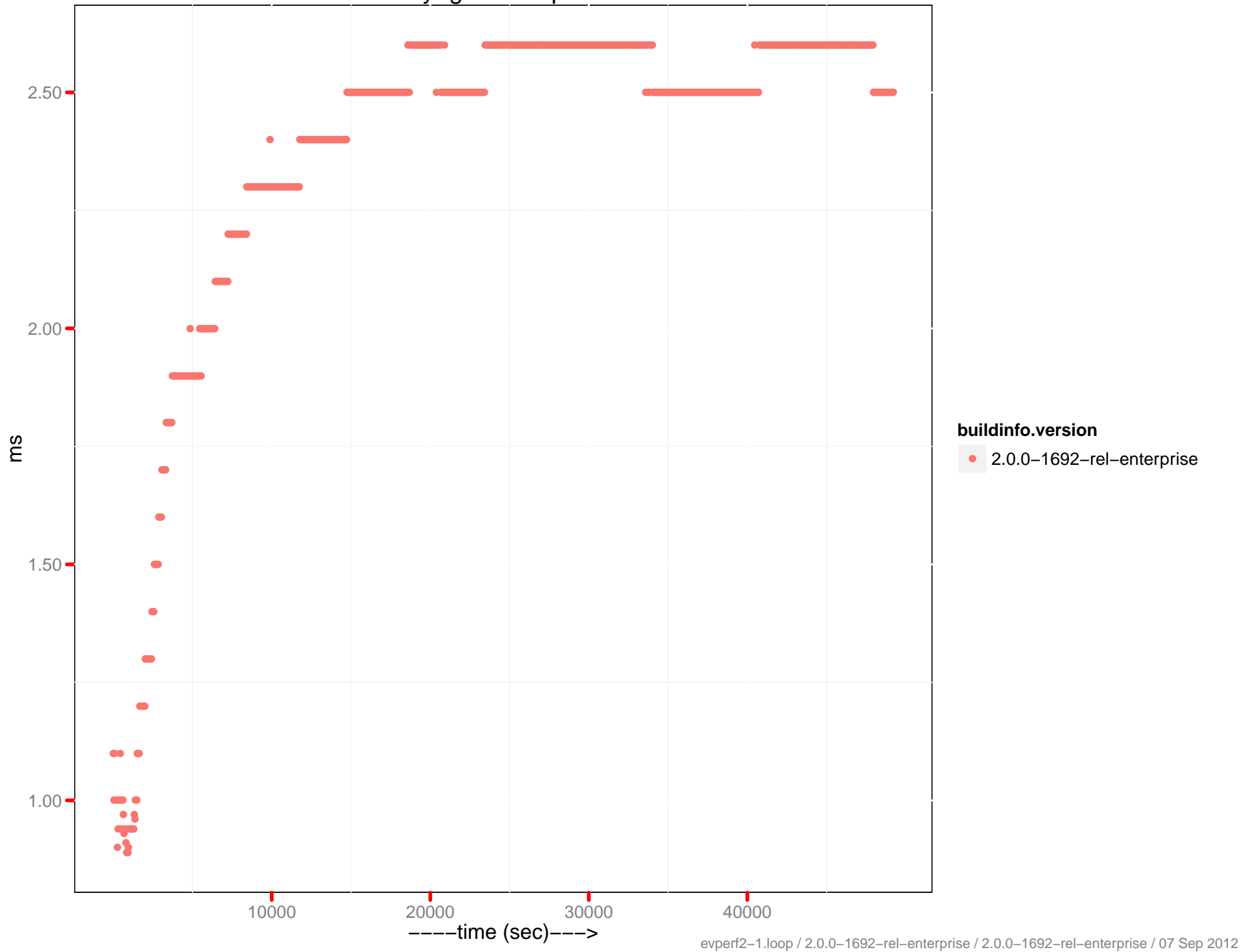
# Latency-get 90th percentile



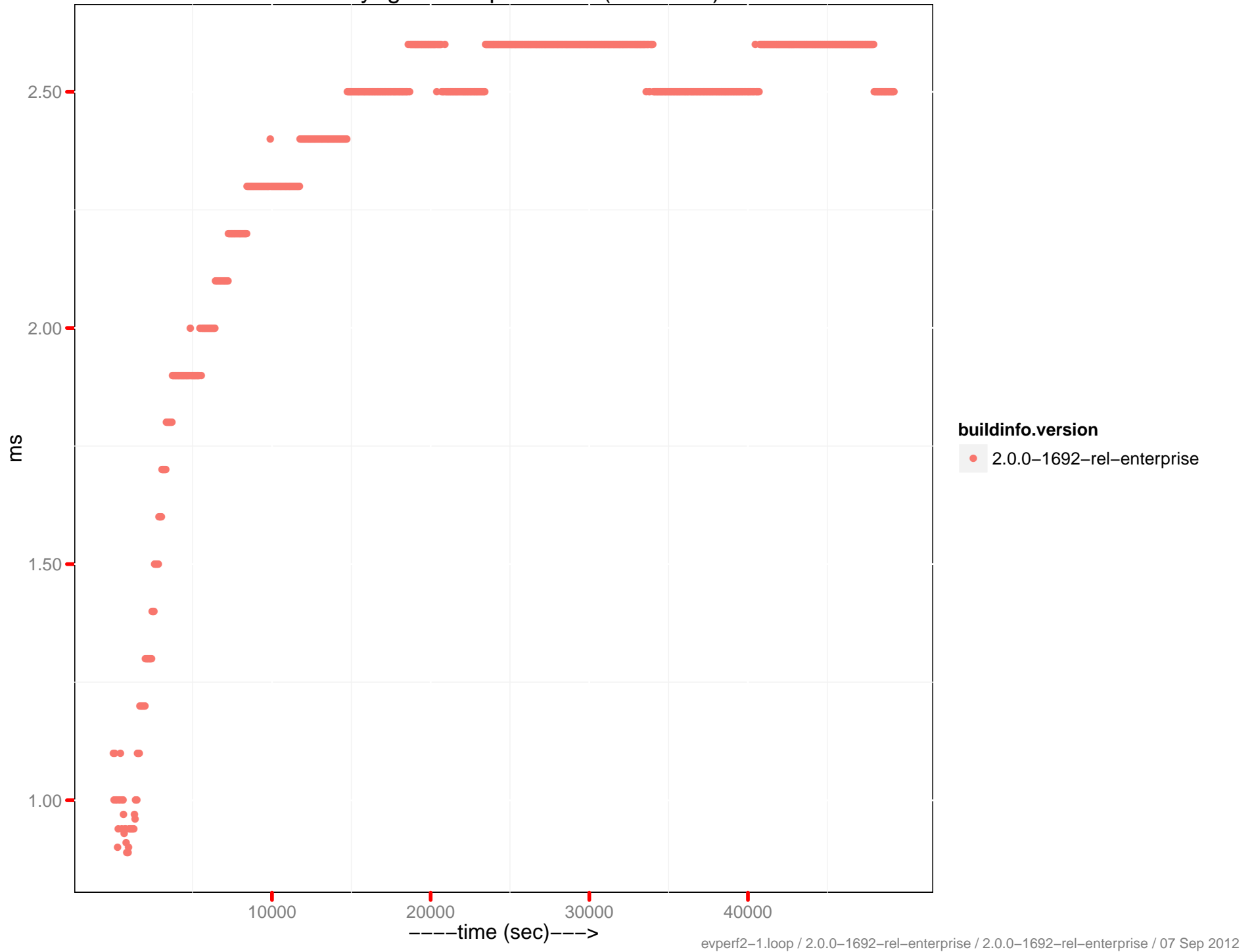
Latency-get 90th percentile (0 - 10ms)



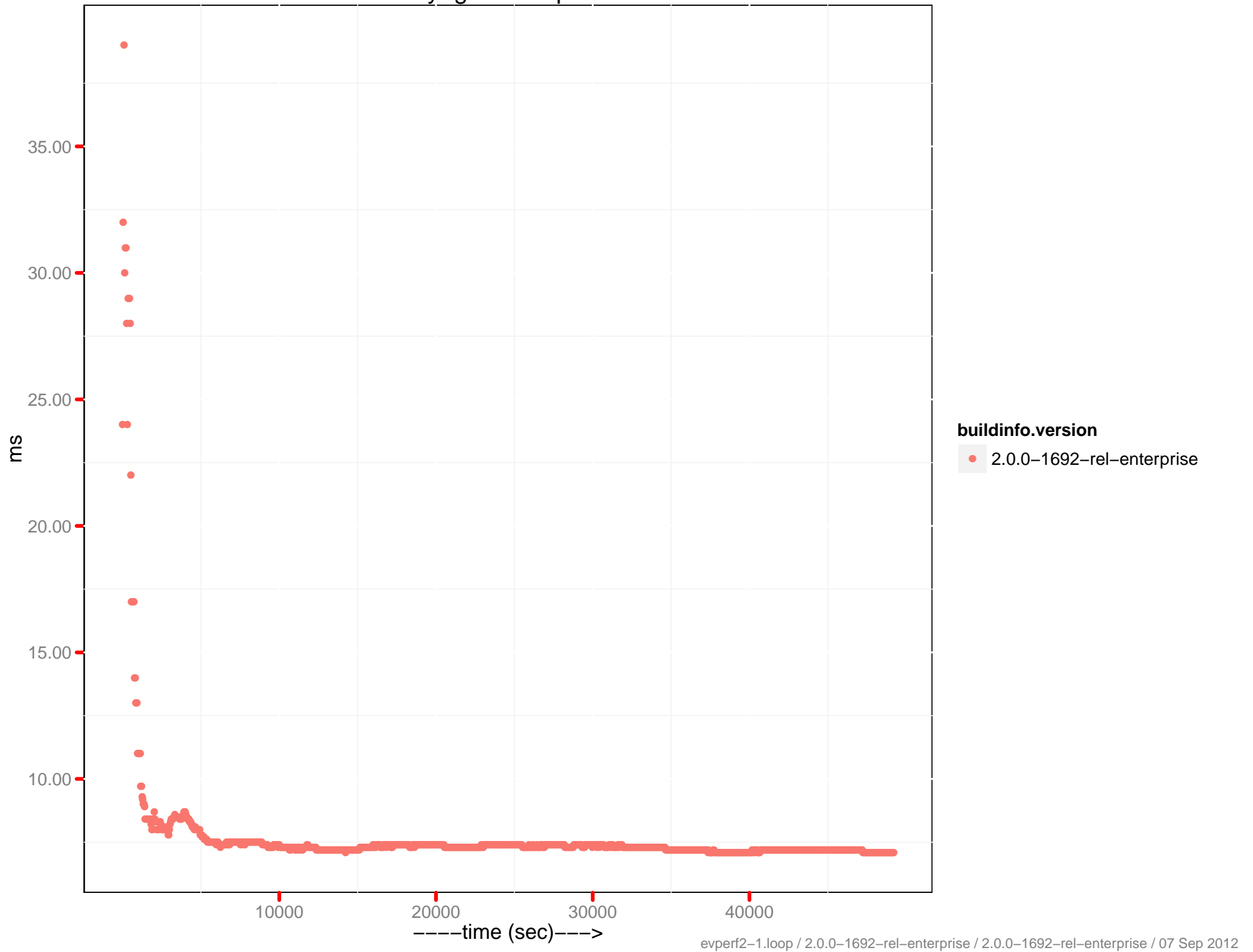
# Latency-get 95th percentile



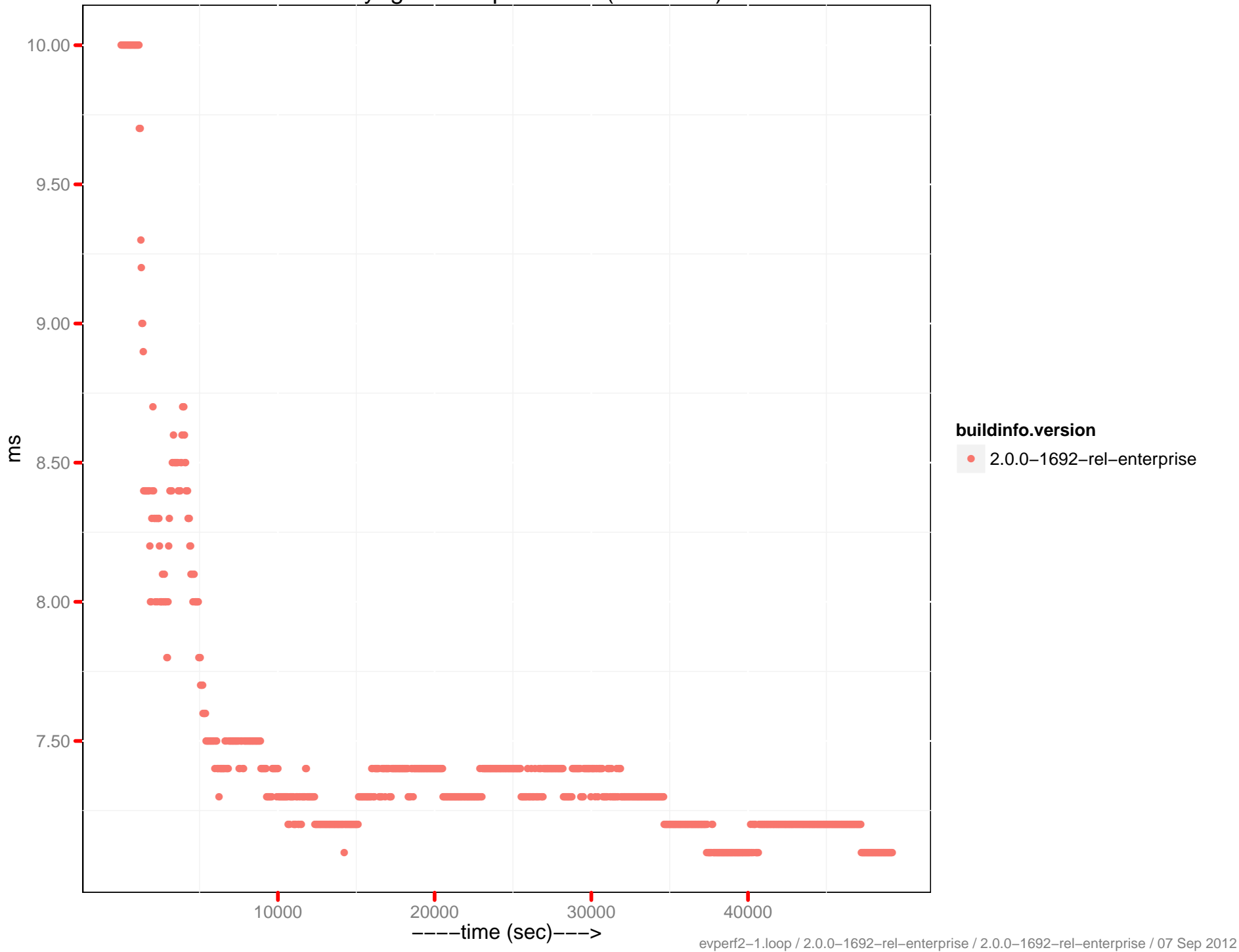
Latency-get 95th percentile (0 - 10ms)



Latency-get 99th percentile

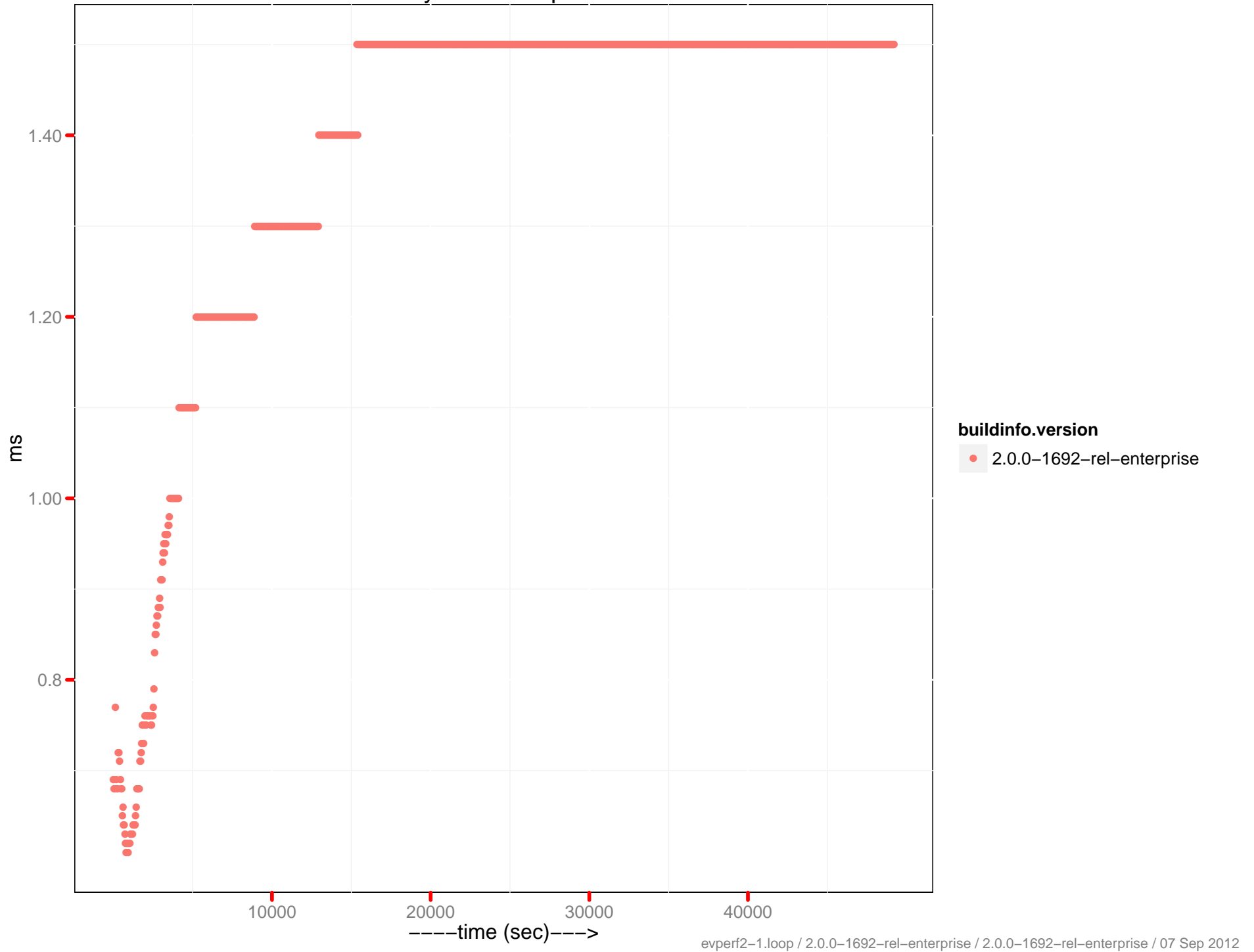


Latency-get 99th percentile (0 - 10ms)

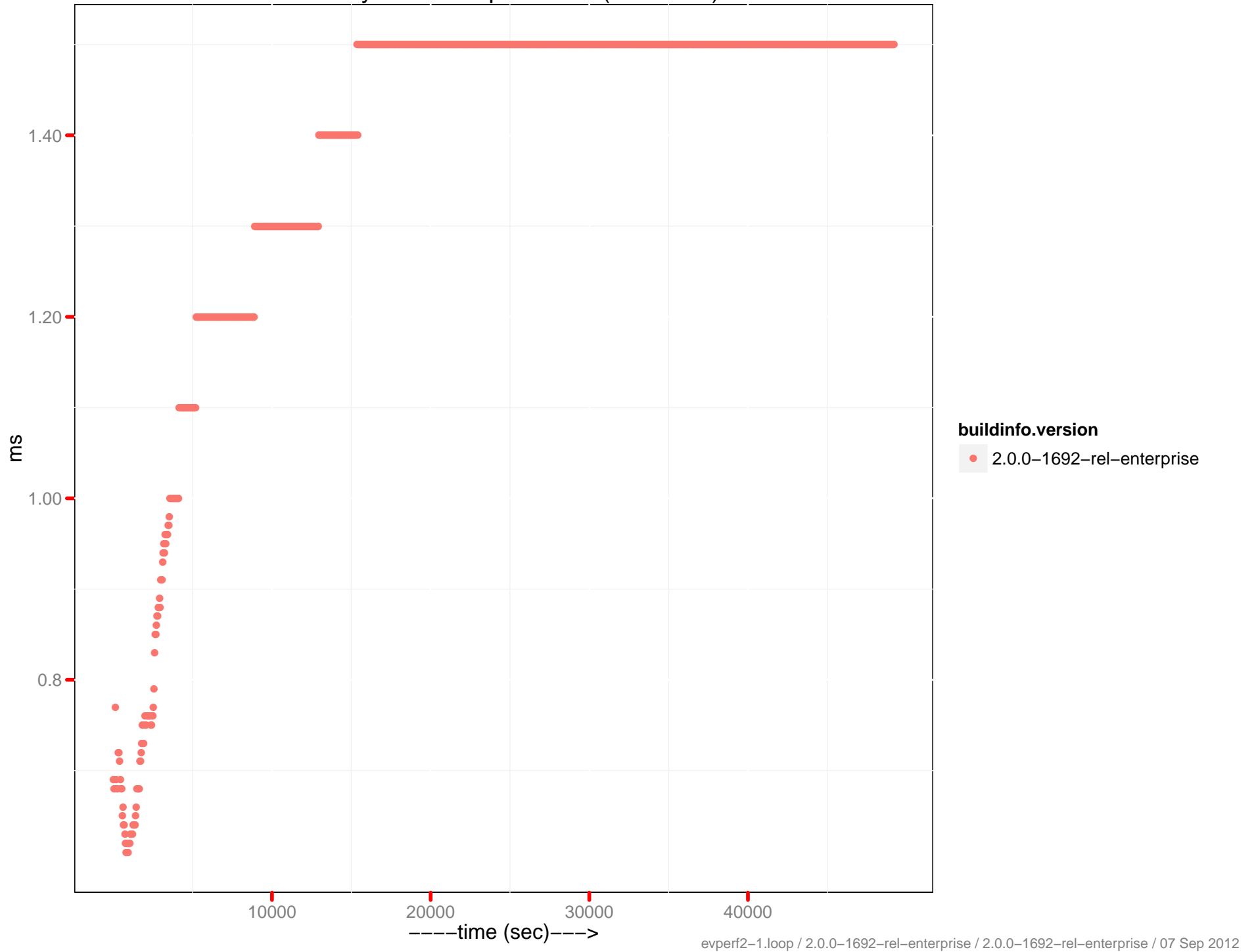




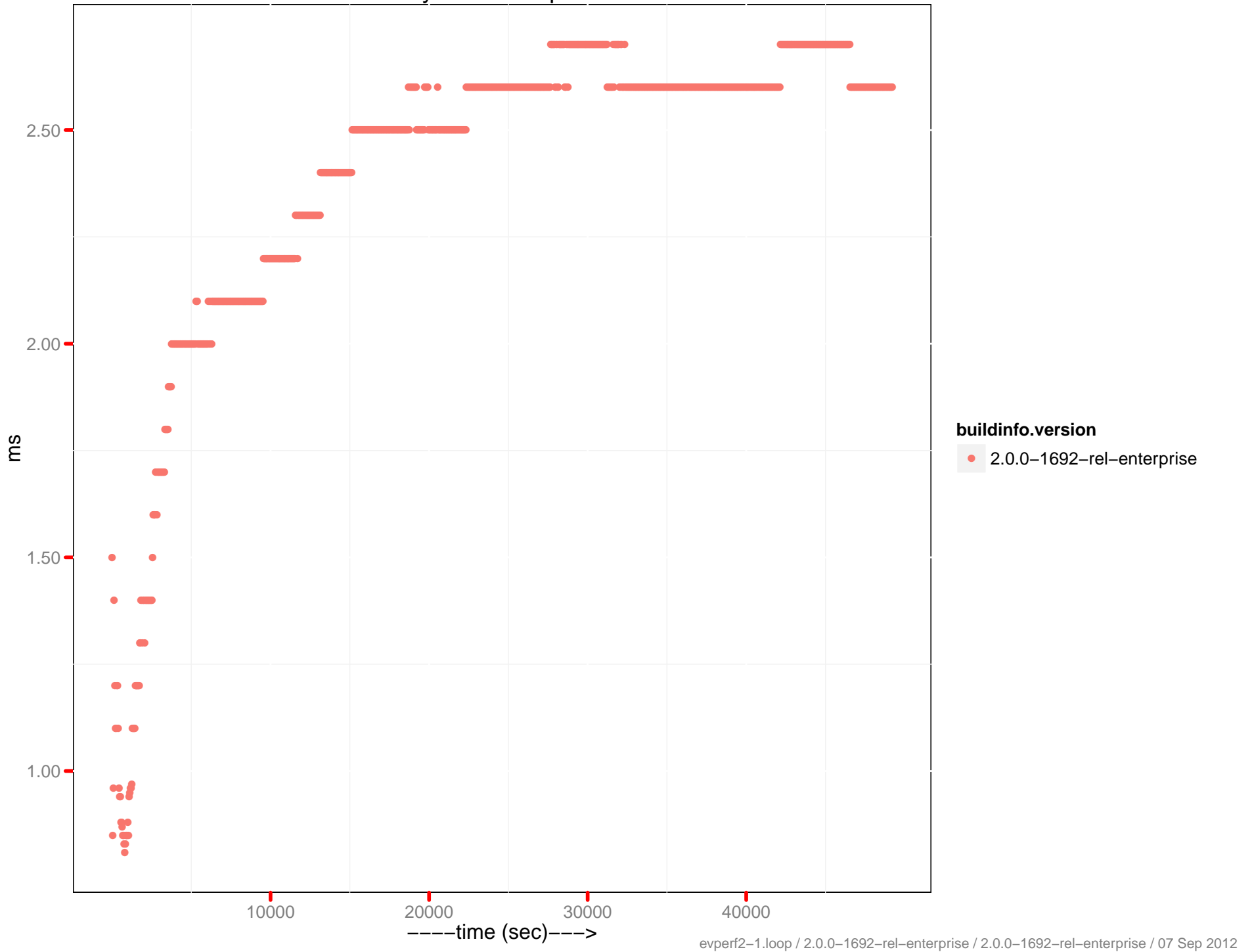
# Latency-set 90th percentile



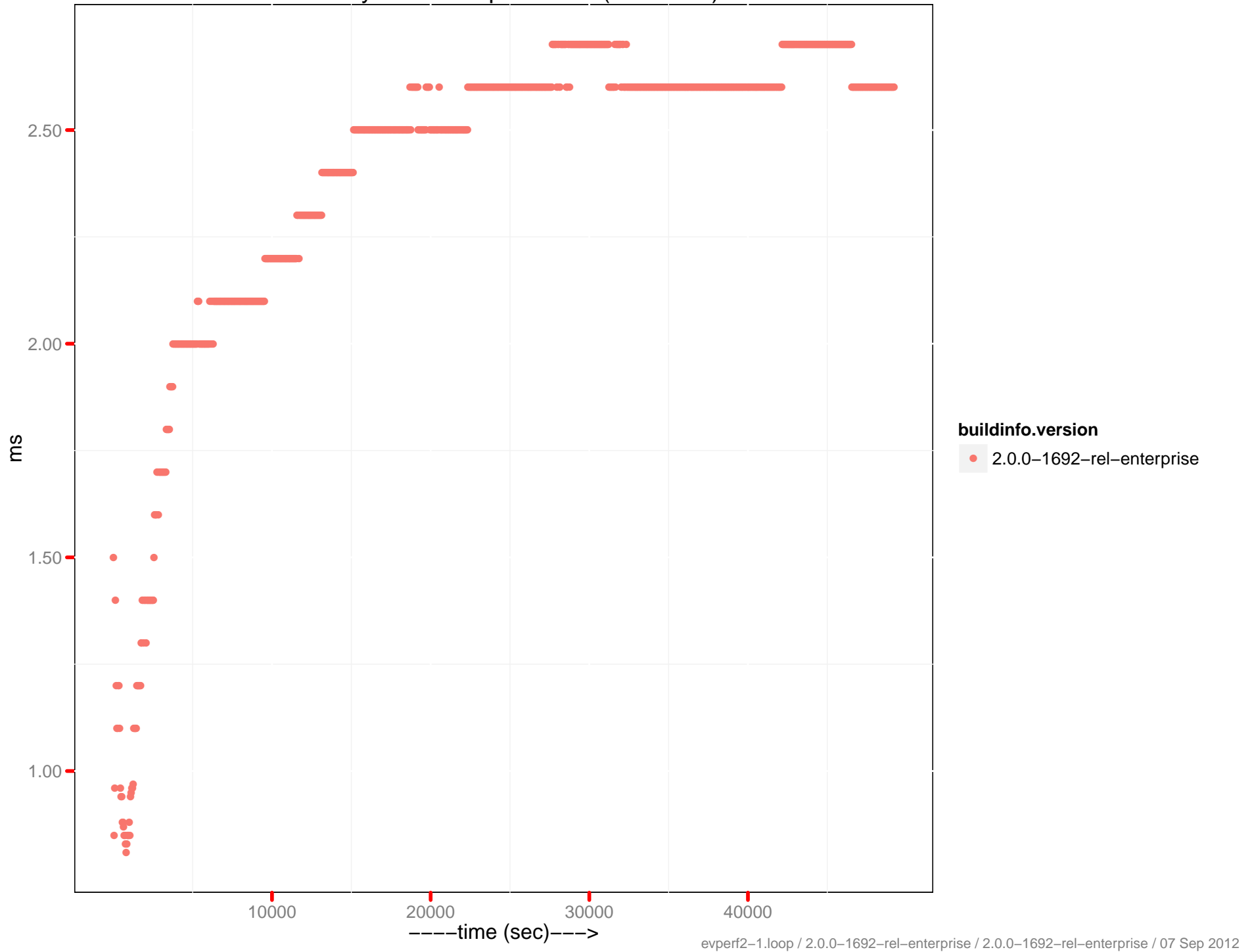
Latency-set 90th percentile (0 - 10ms)



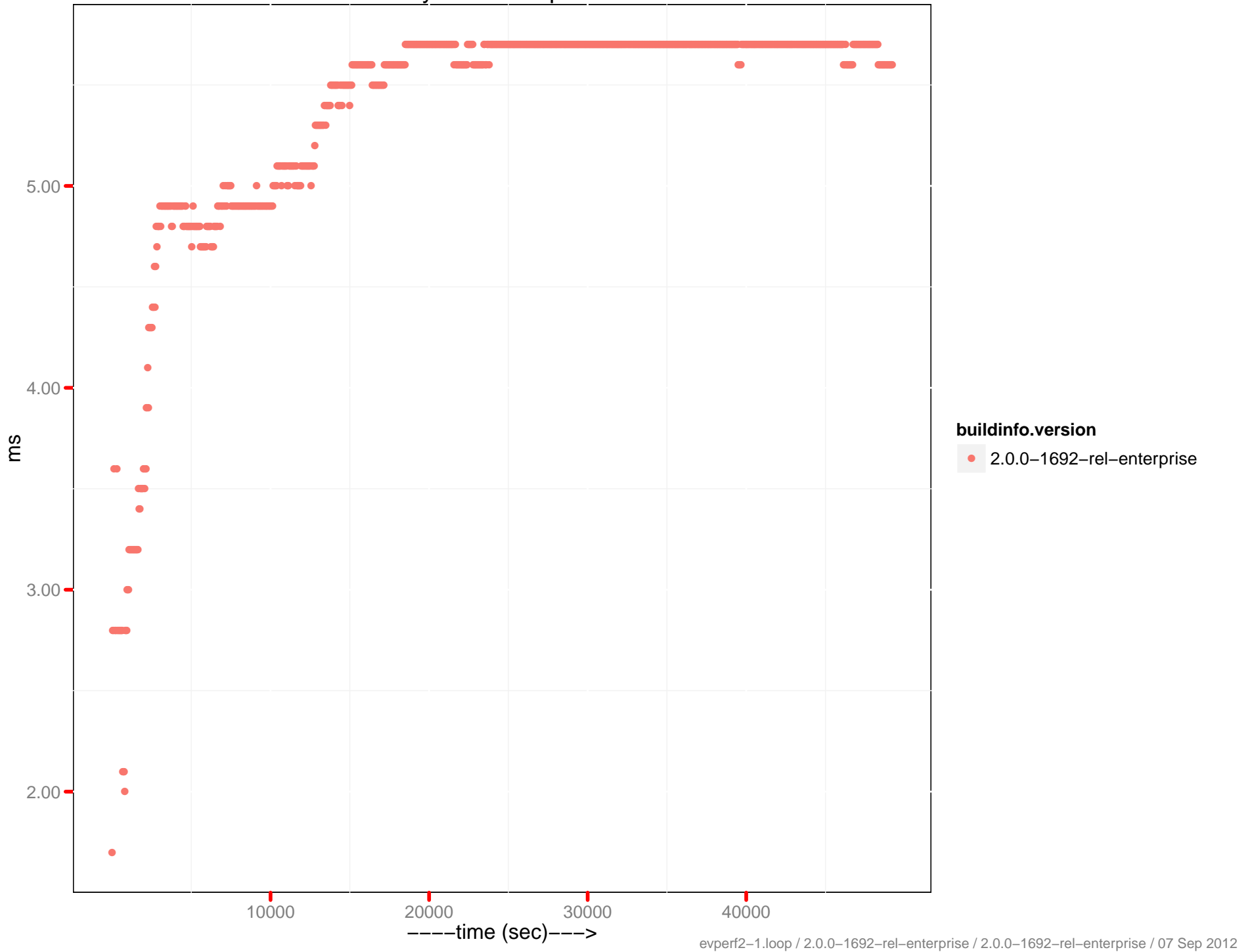
Latency-set 95th percentile



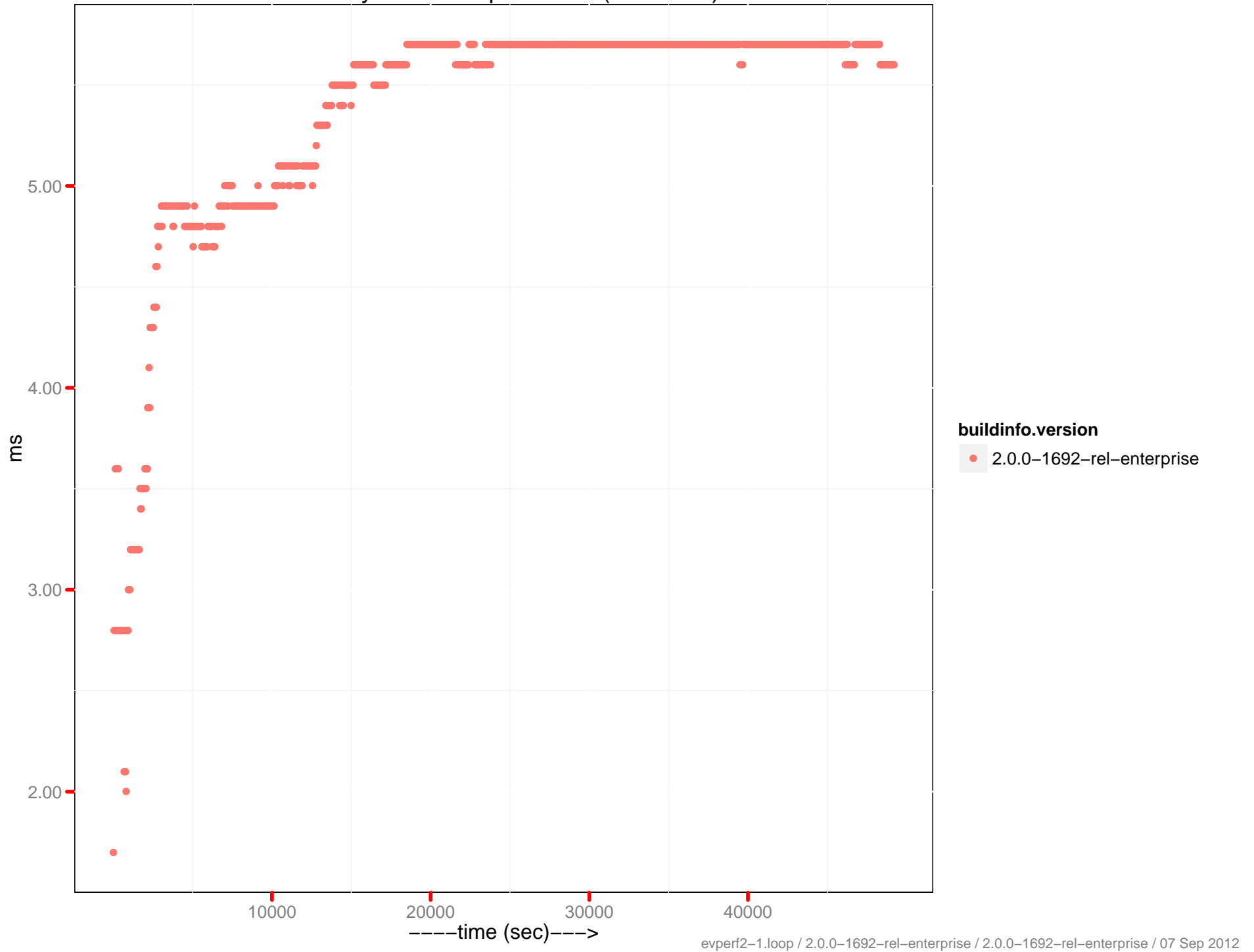
Latency-set 95th percentile (0 - 10ms)



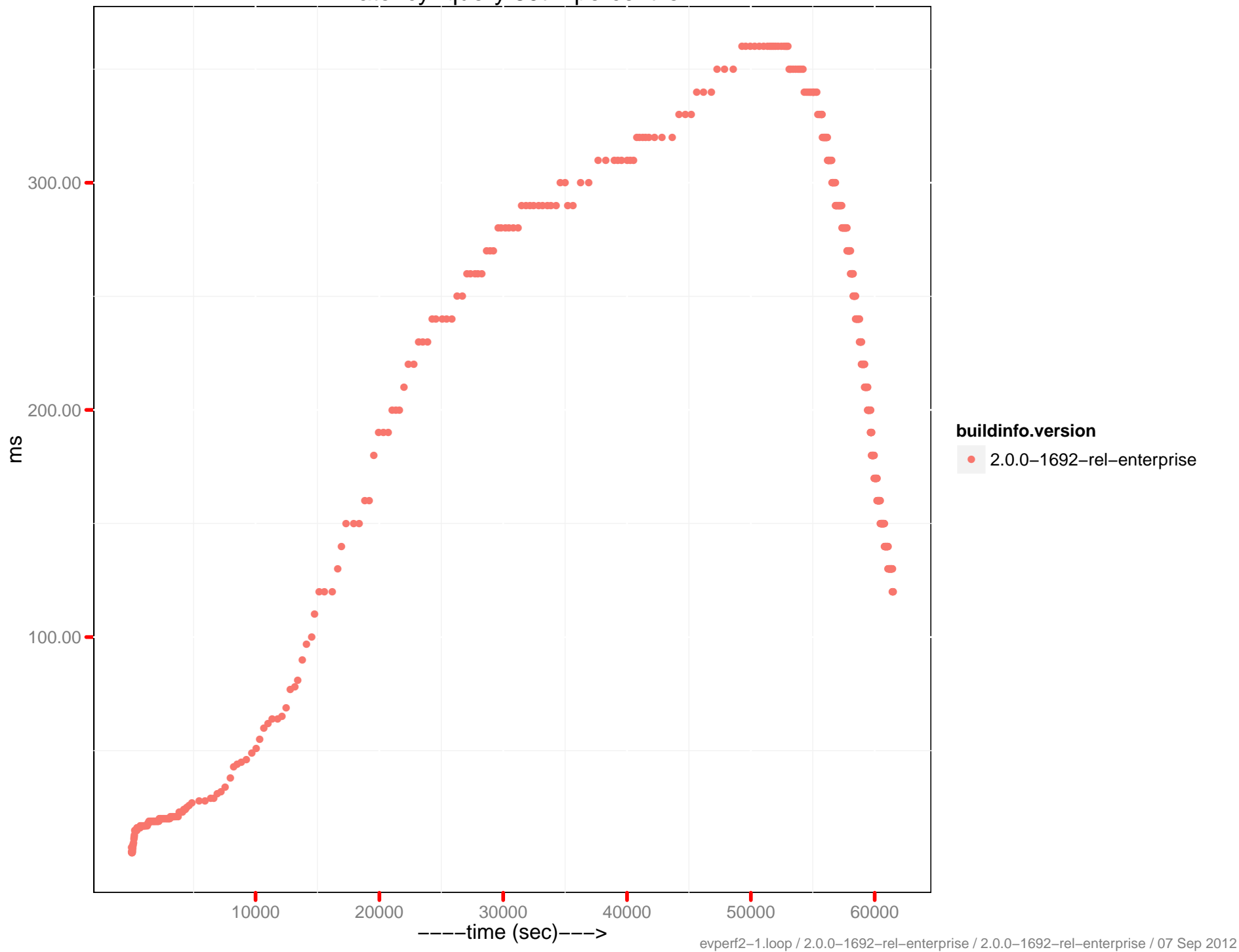
Latency-set 99th percentile



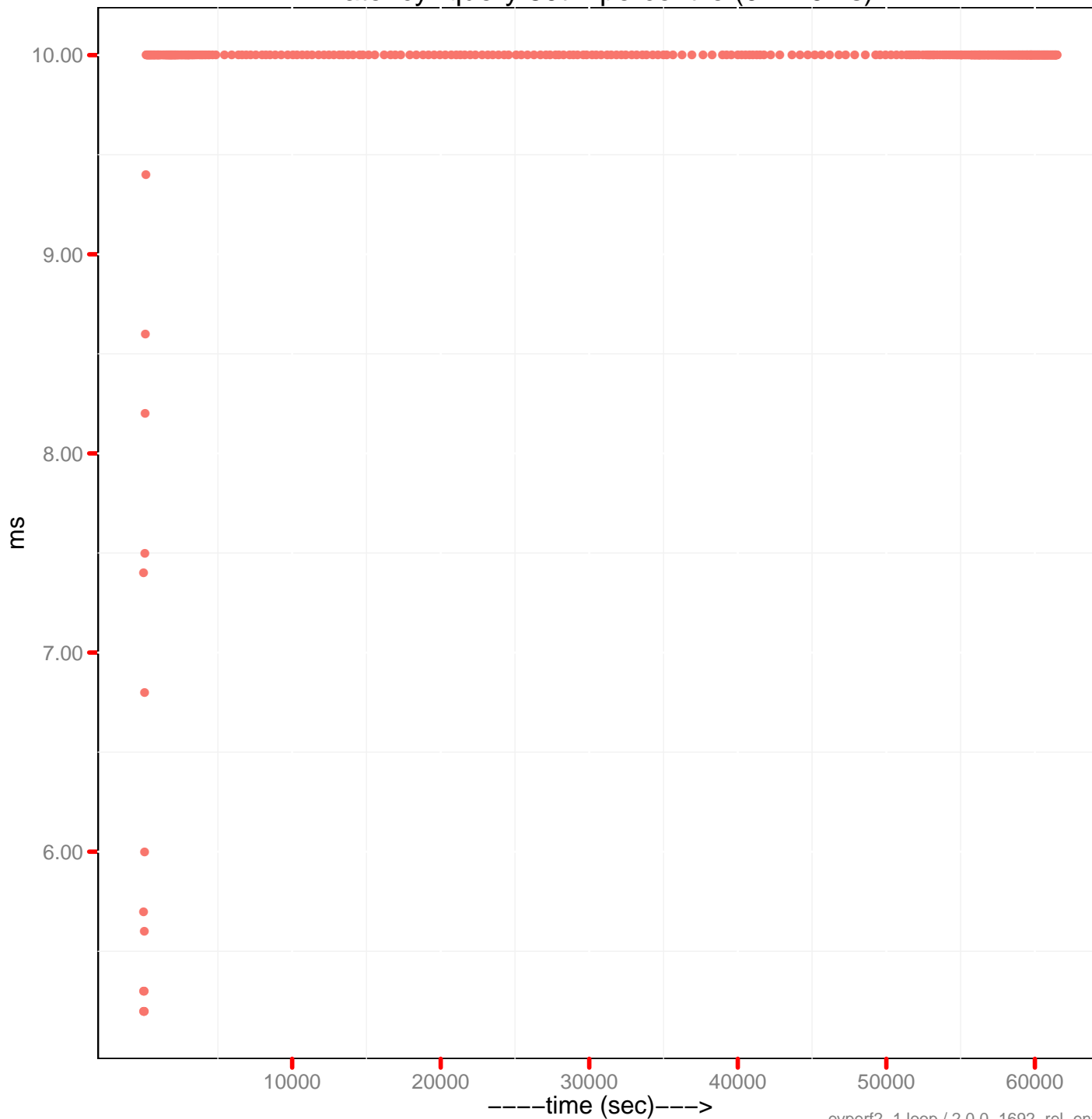
Latency-set 99th percentile (0 - 10ms)



Latency-query 80th percentile



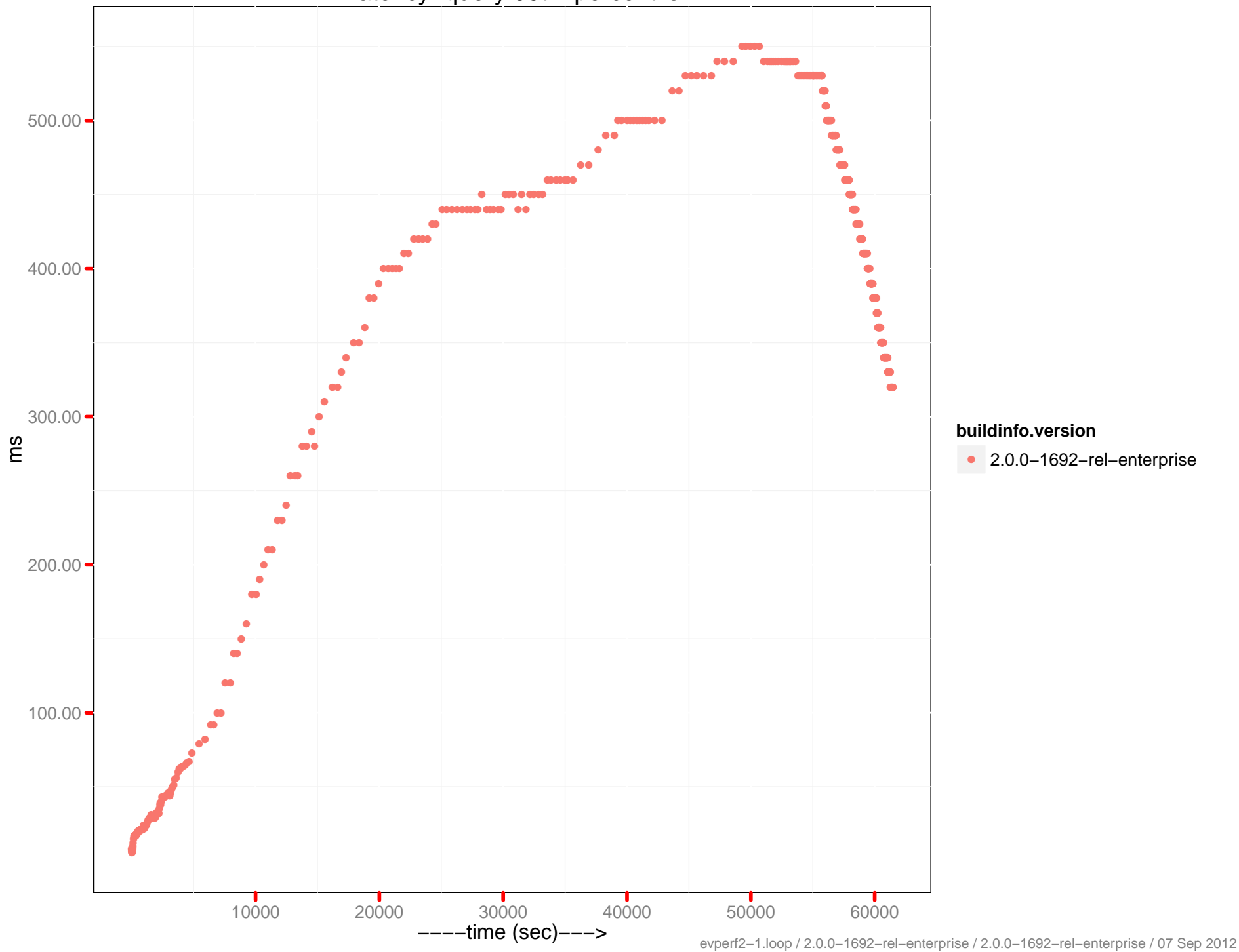
Latency-query 80th percentile (0 - 10ms)



**buildinfo.version**  
● 2.0.0-1692-rel-enterprise

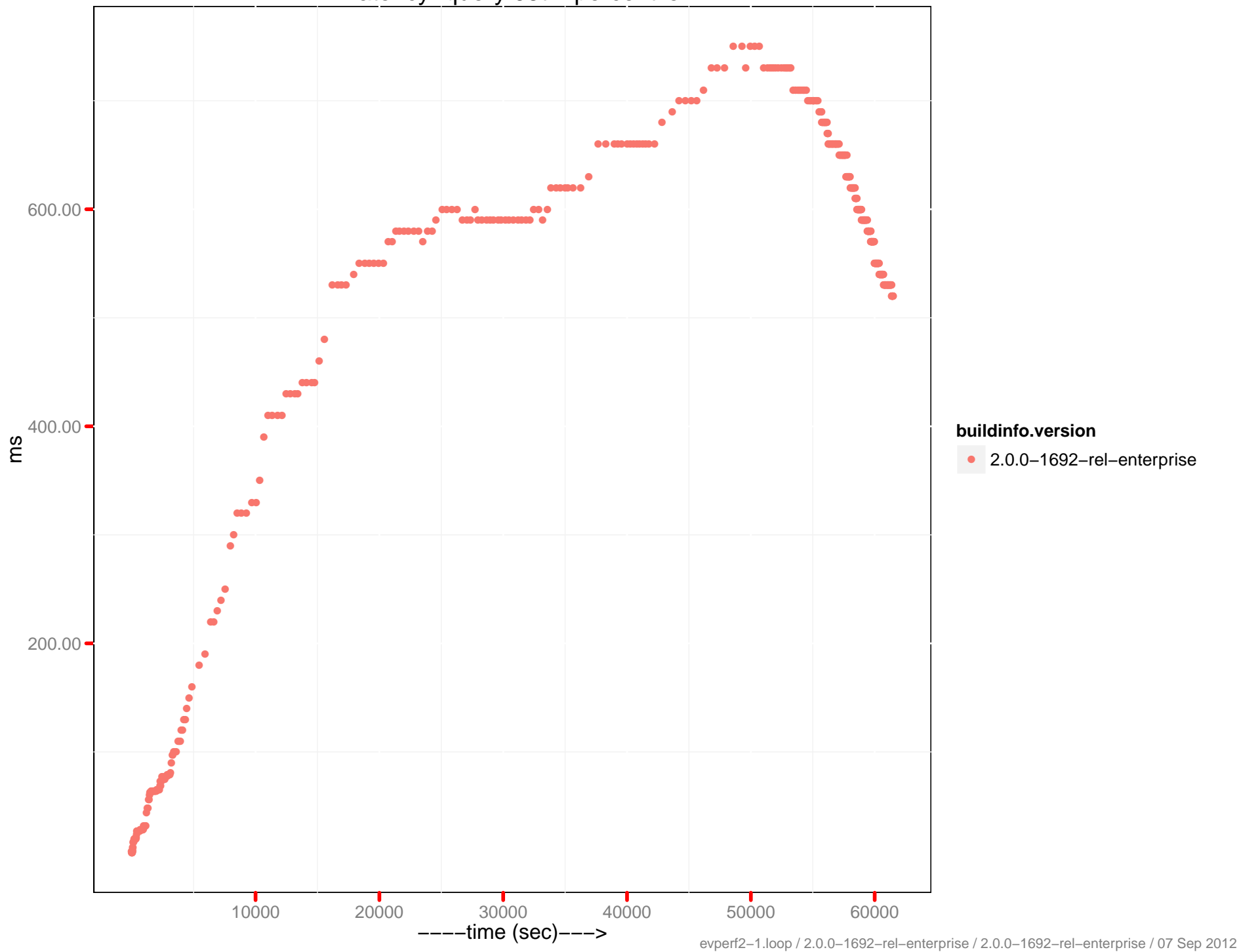


Latency-query 90th percentile

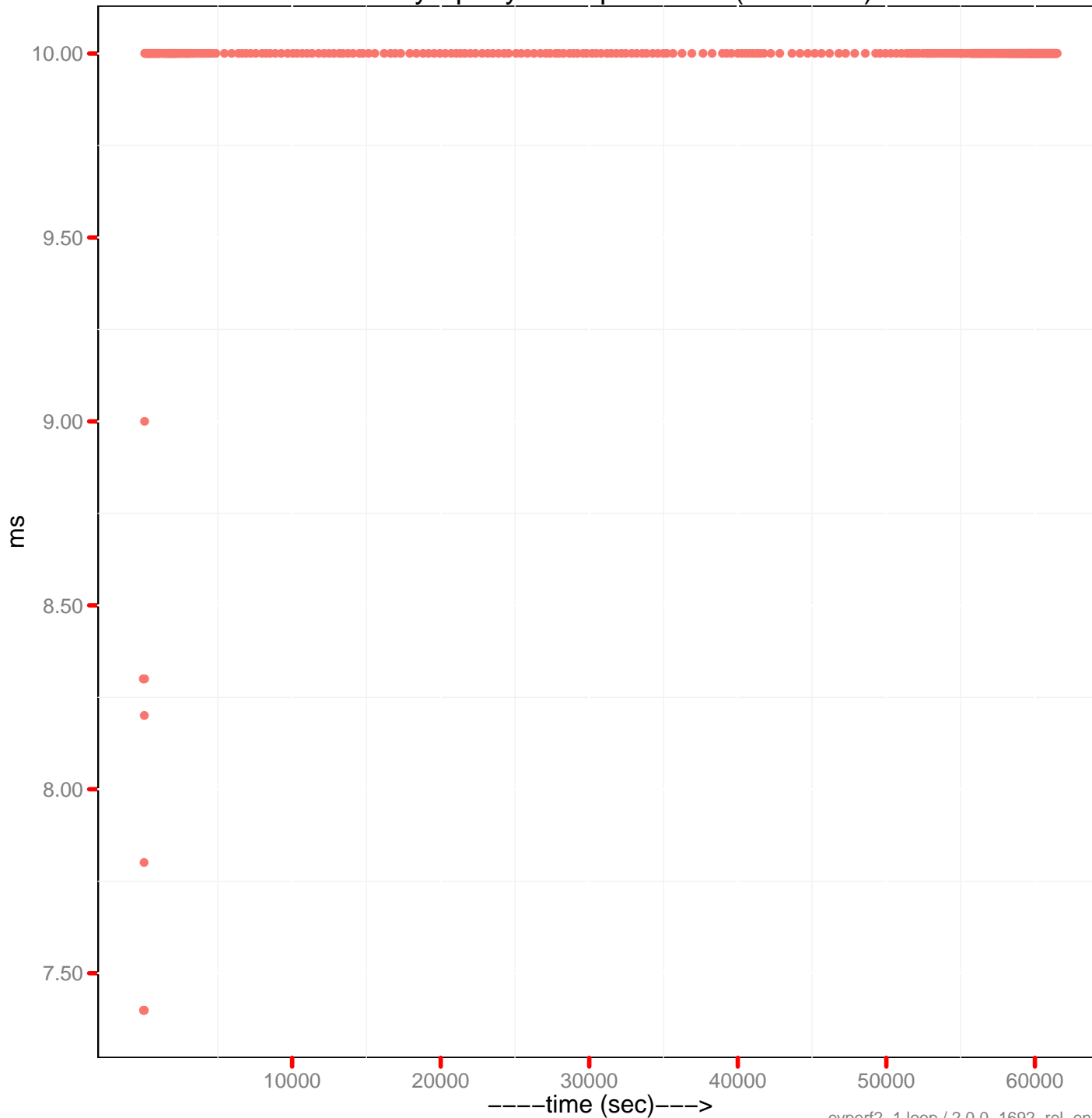




Latency-query 95th percentile



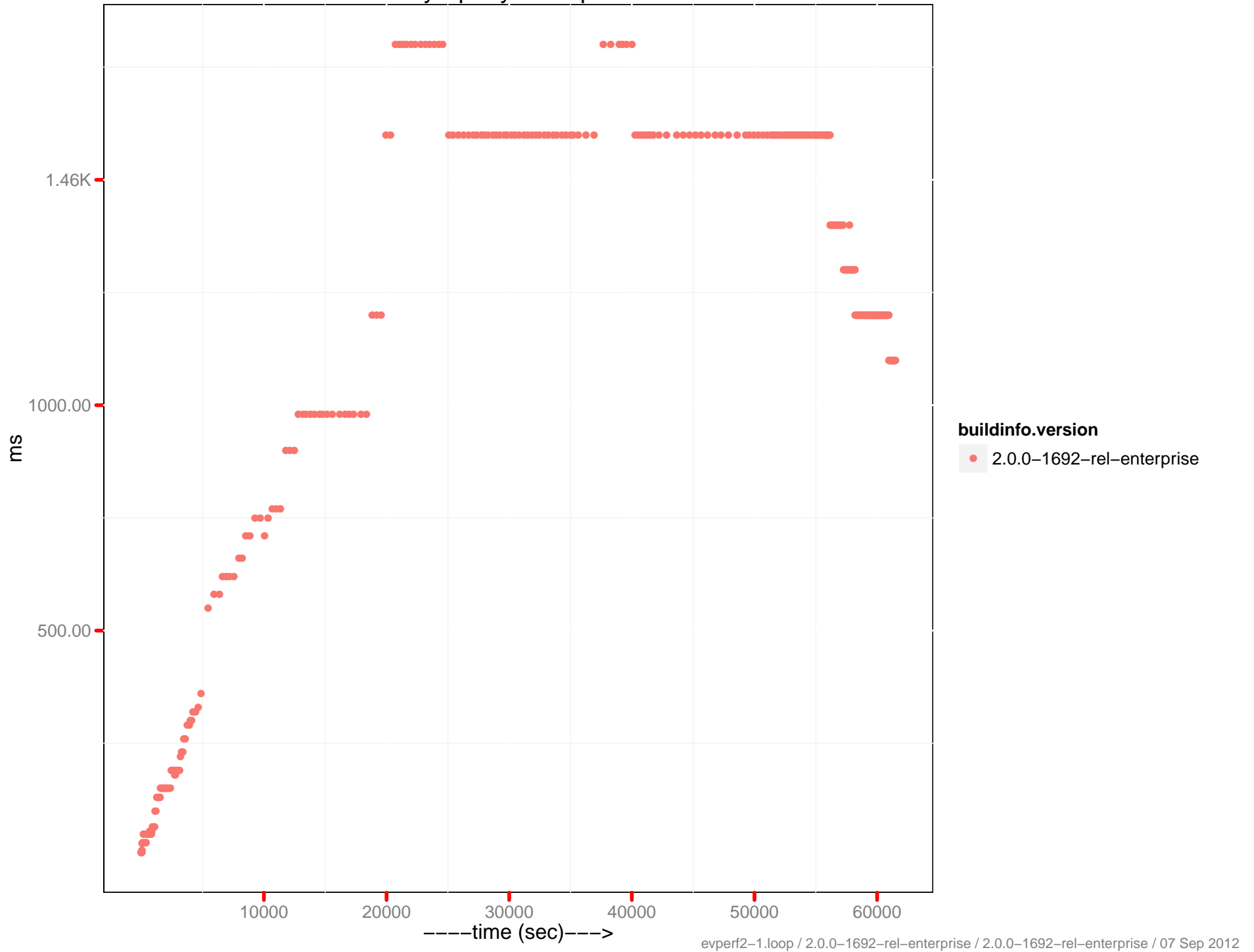
# Latency-query 95th percentile (0 - 10ms)



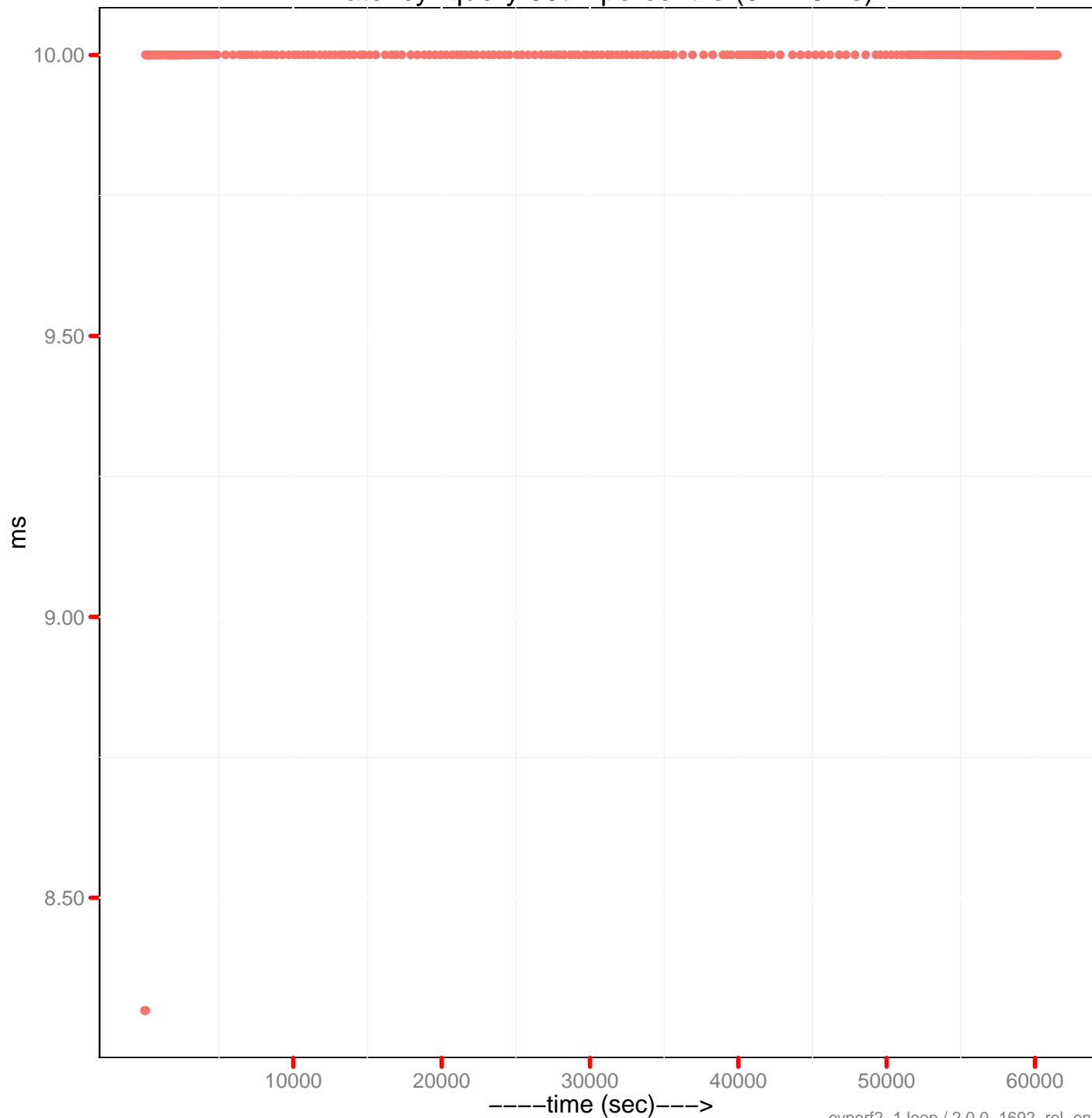
## buildinfo.version

• 2.0.0-1692-rel-enterprise

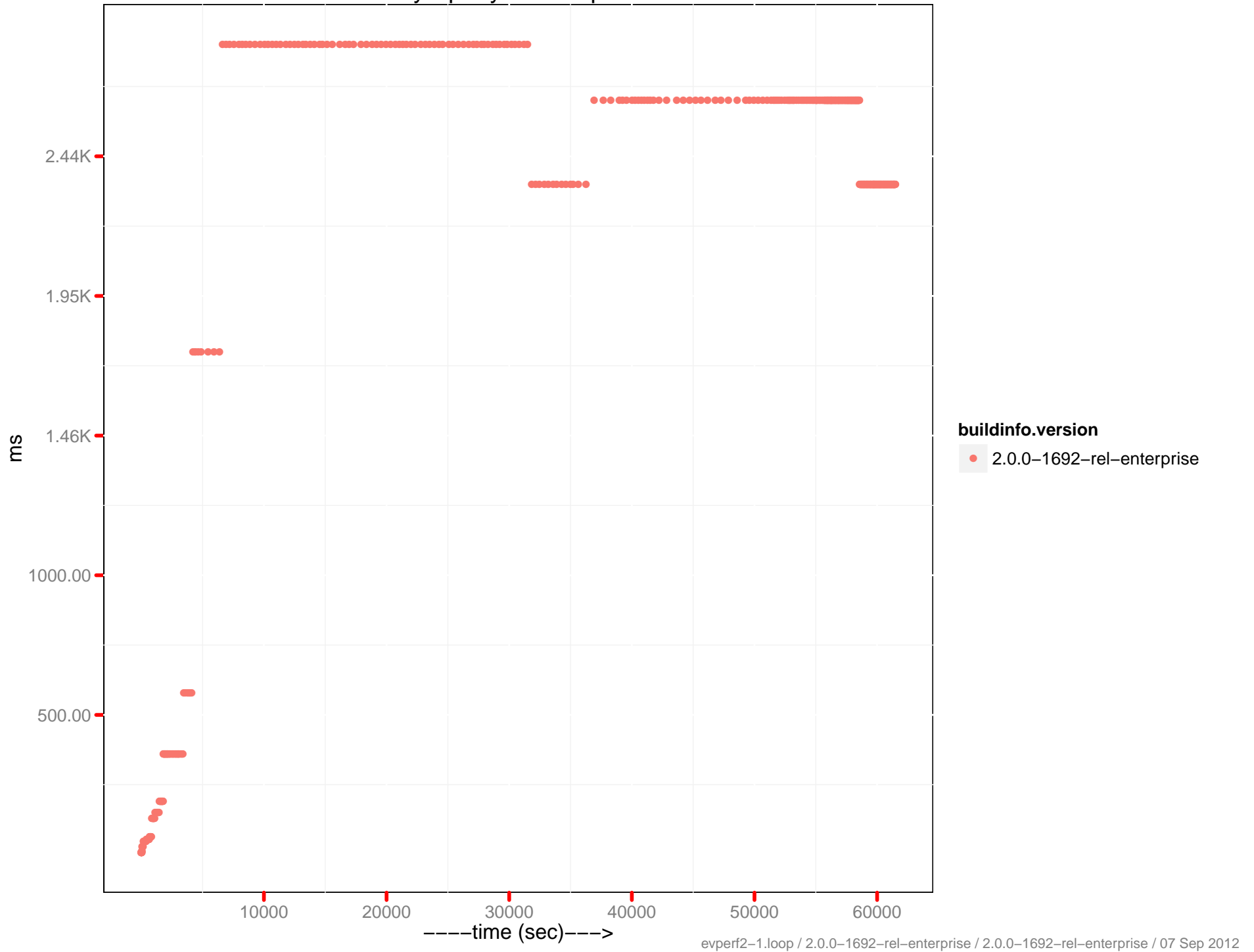
Latency-query 99th percentile



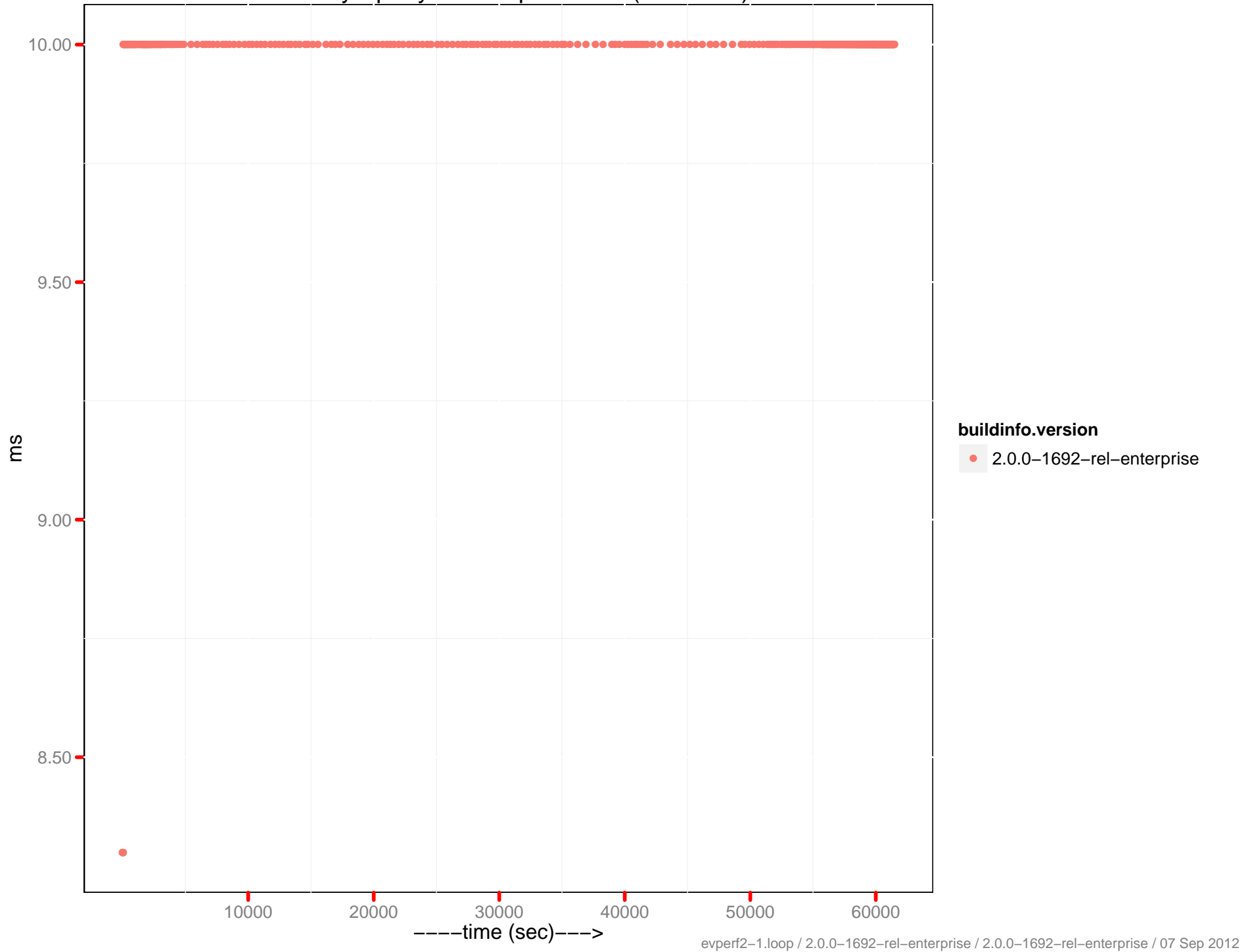
Latency-query 99th percentile (0 - 10ms)



# Latency-query 99.9th percentile

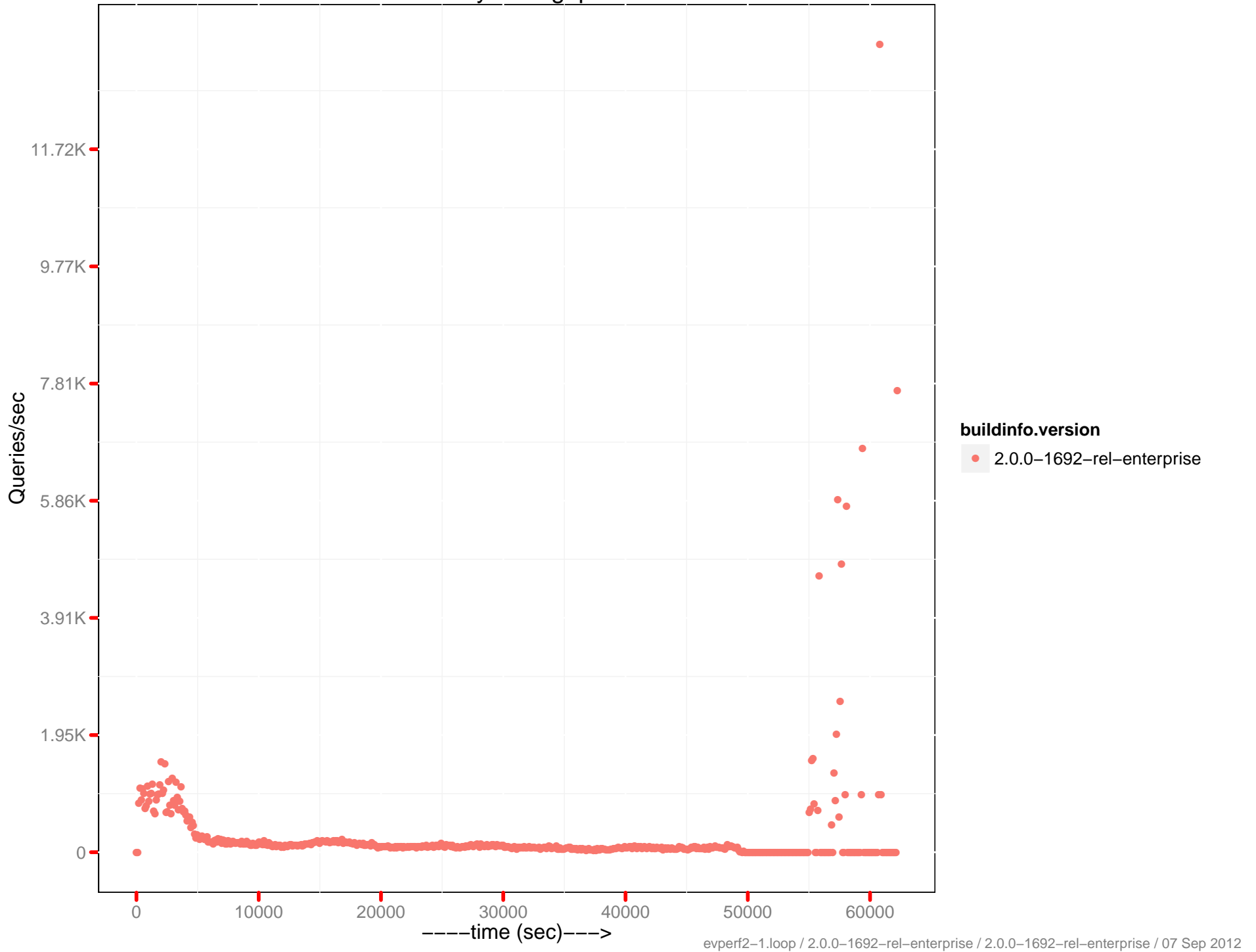


Latency-query 99.9th percentile (0 - 10ms)

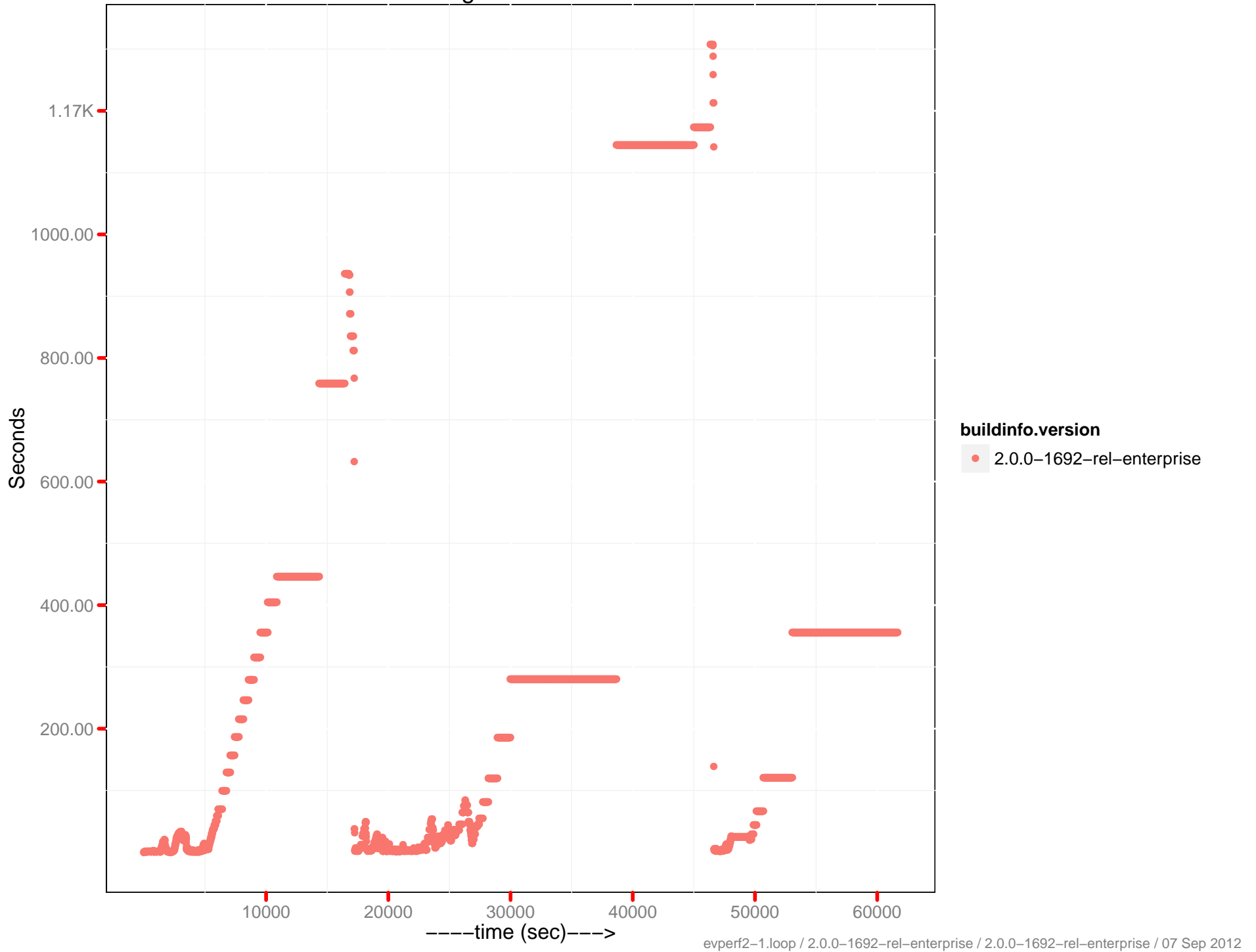




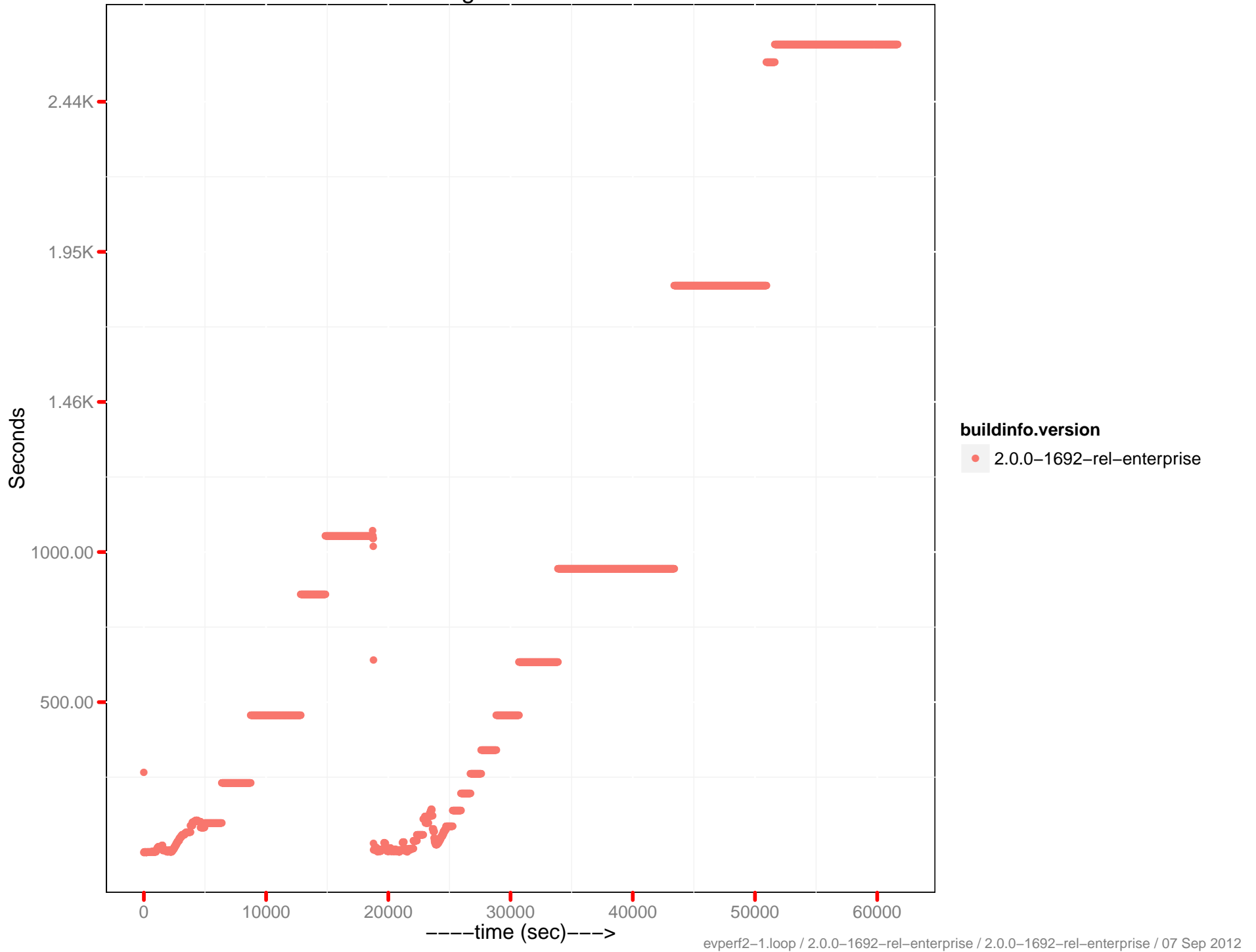
# Query throughput



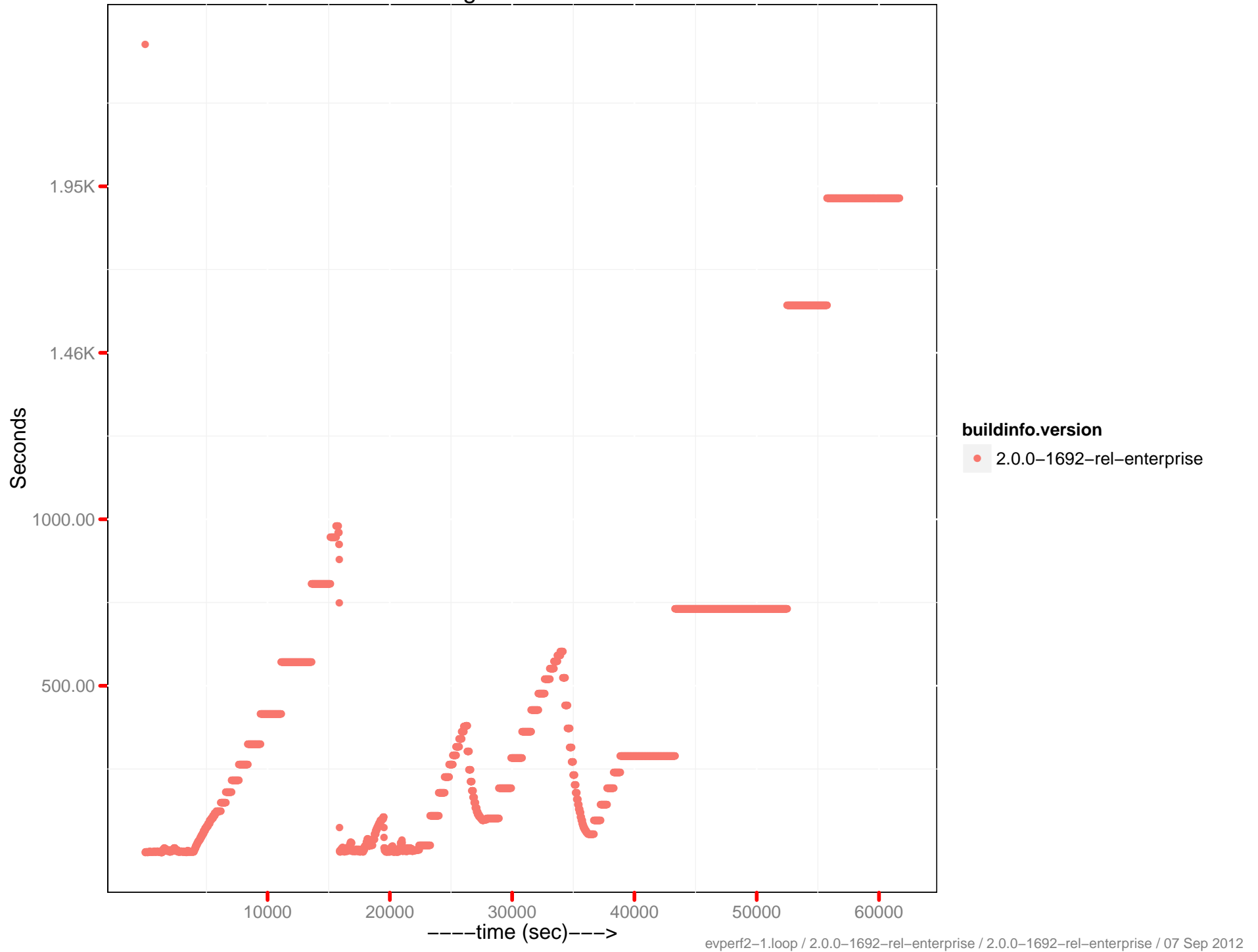
# Indexing time – 10.2.1.58



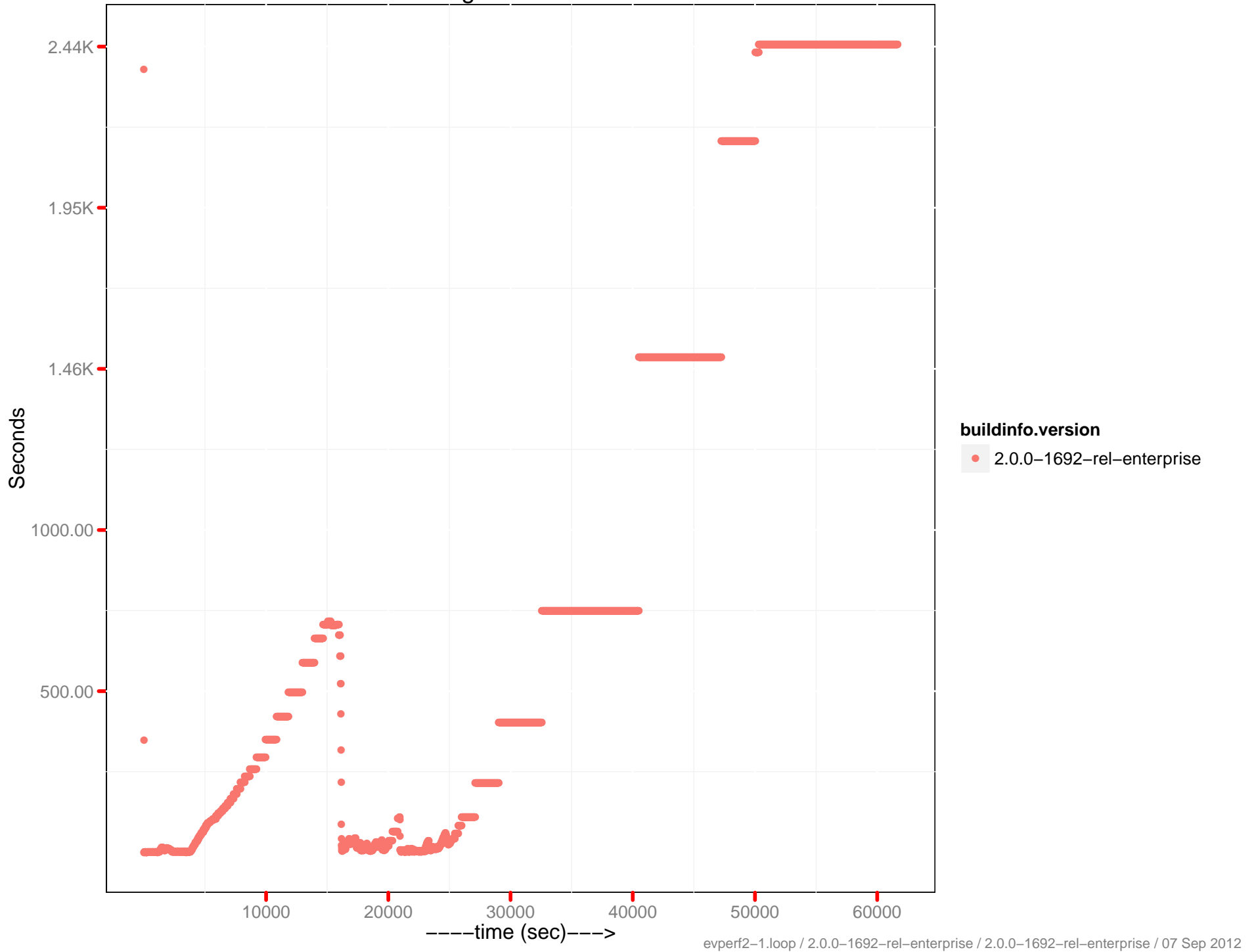
# Indexing time – 10.2.1.61



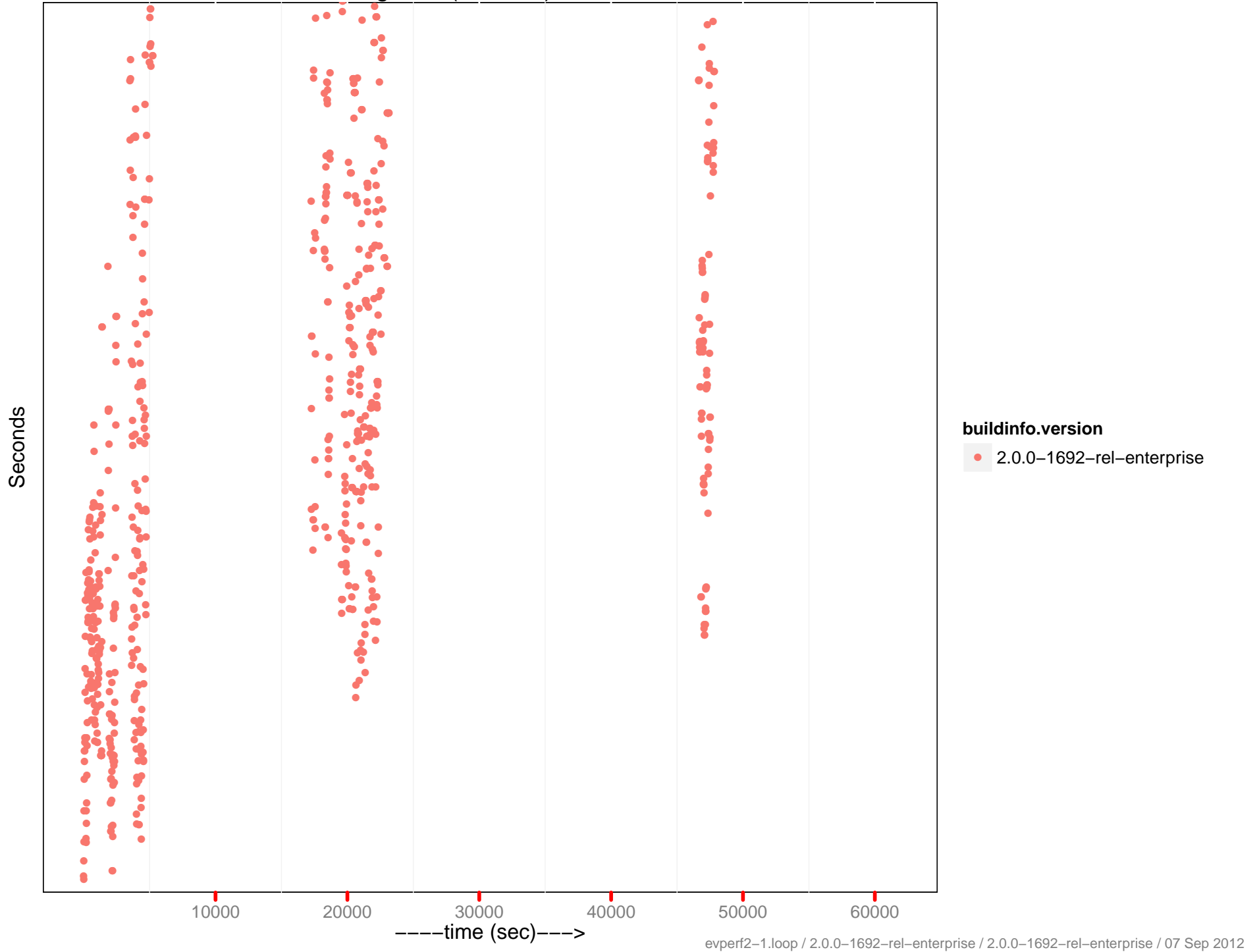
# Indexing time – 10.2.1.63



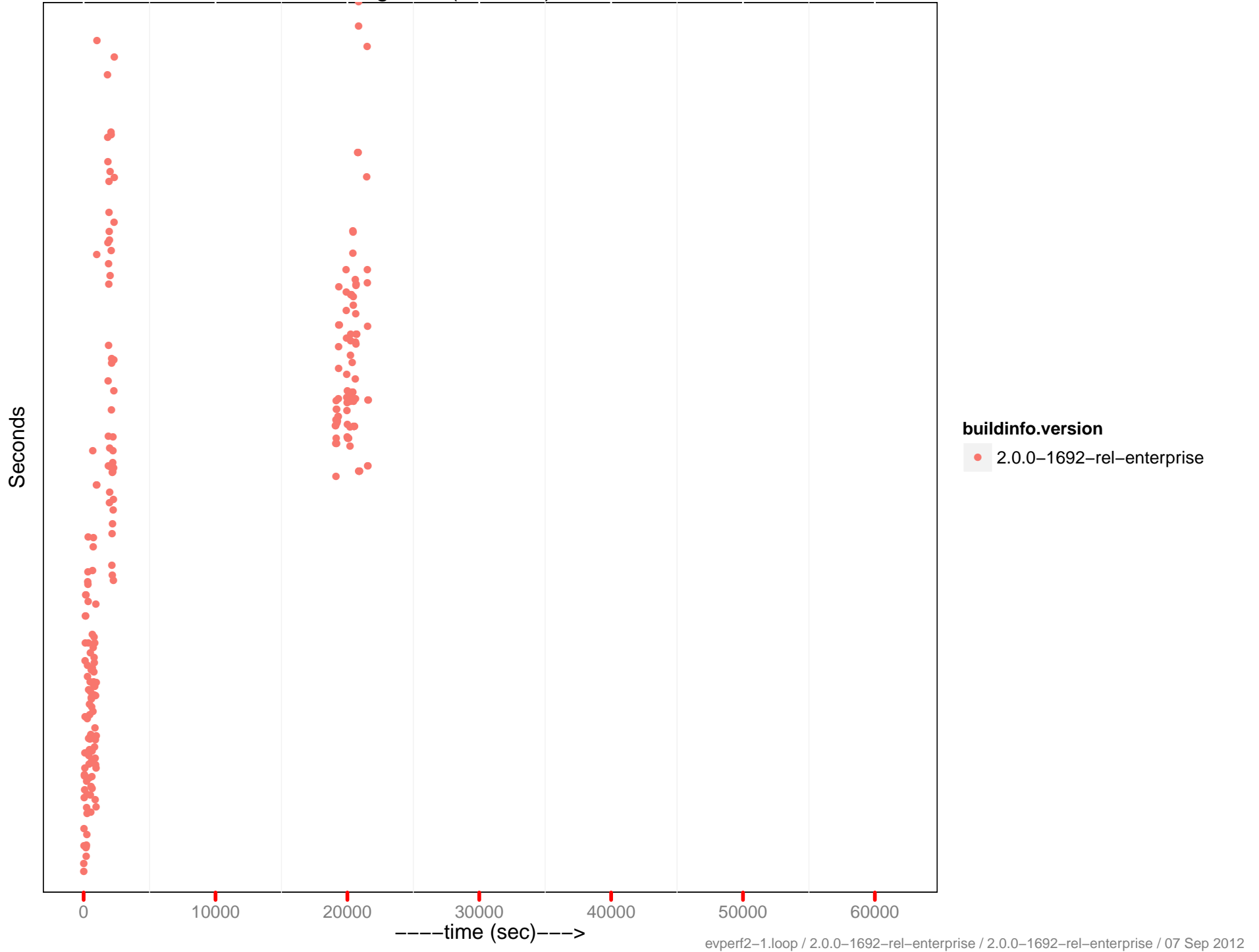
# Indexing time – 10.2.1.64



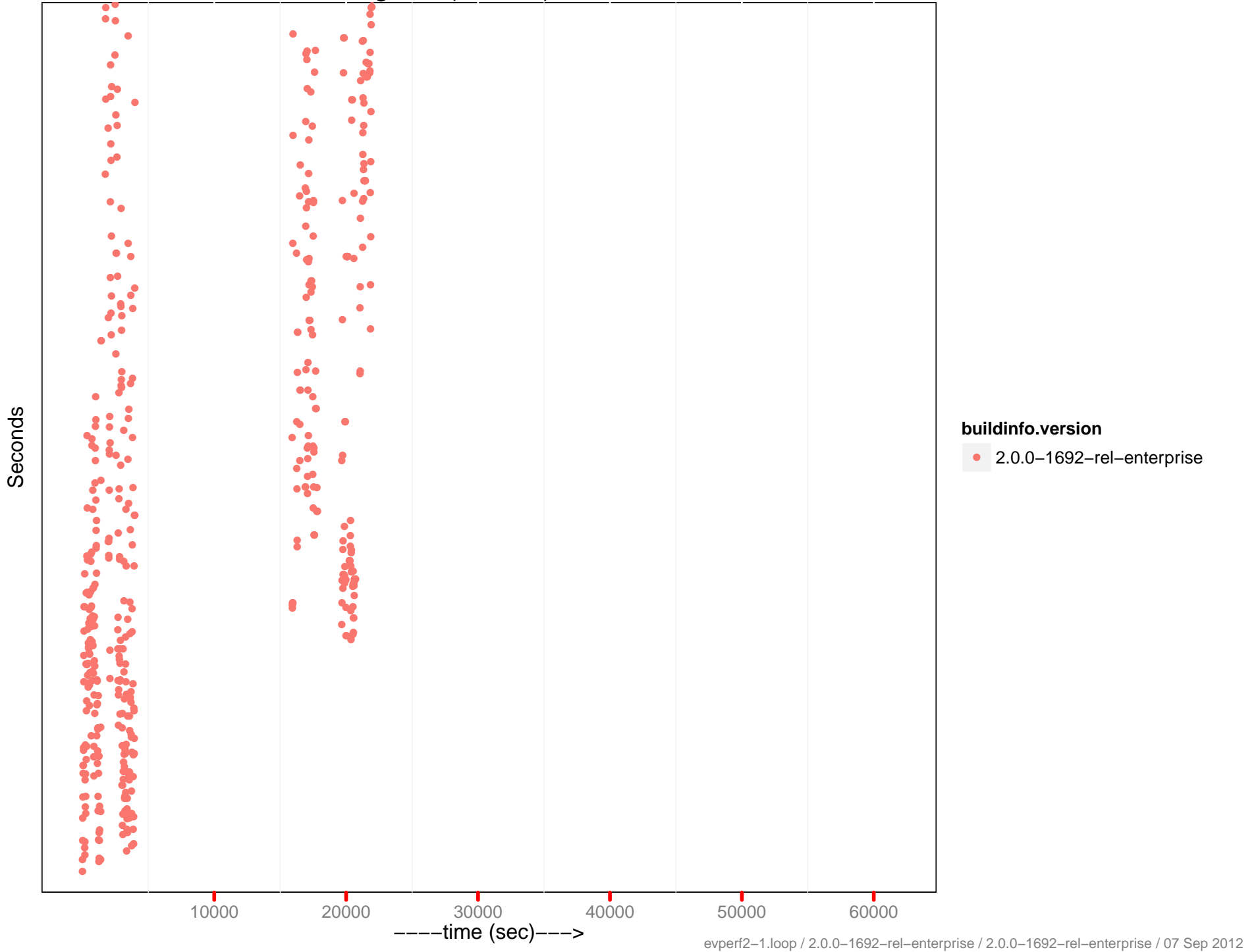
# Indexing time (0-5 sec) - 10.2.1.58



# Indexing time (0-5 sec) - 10.2.1.61

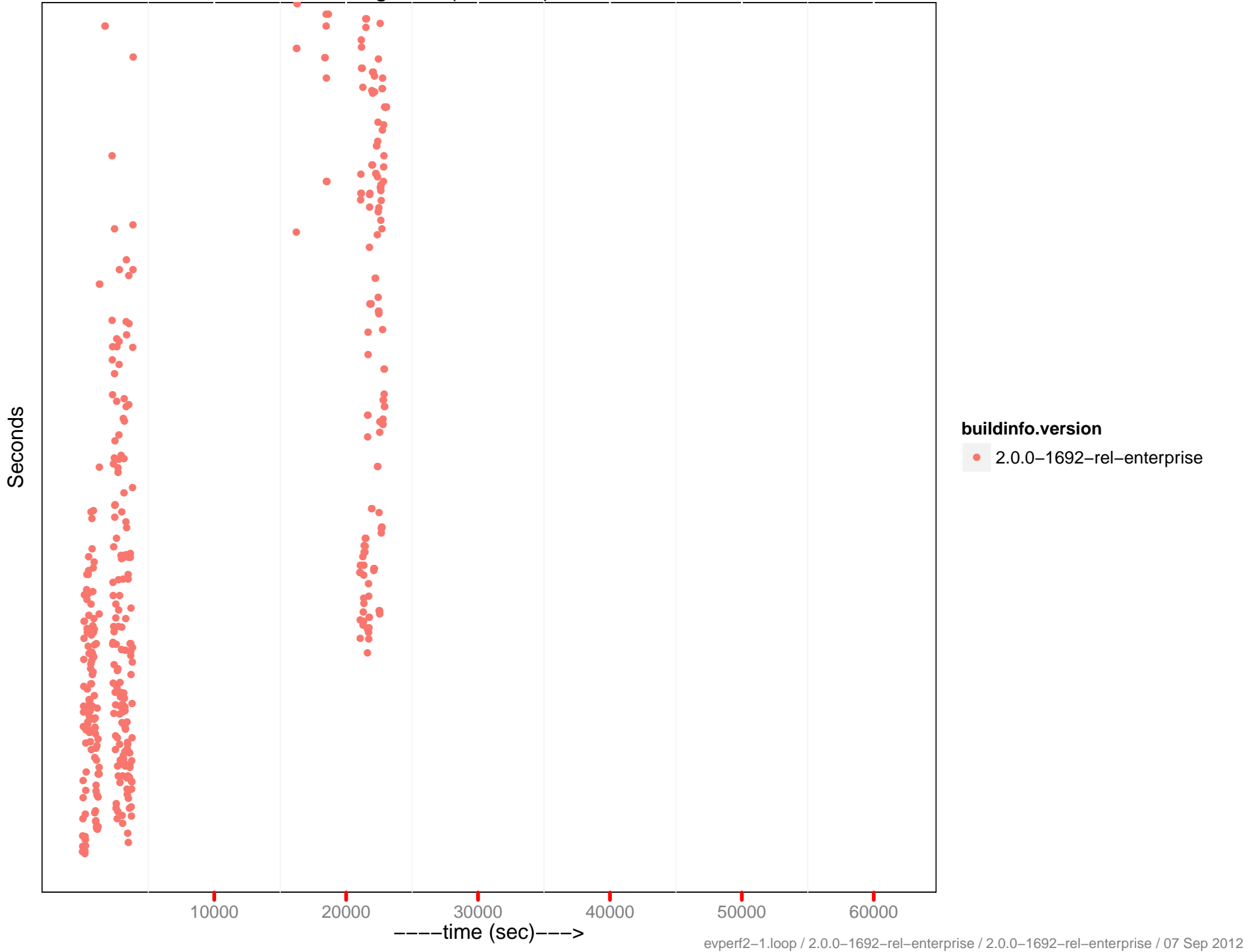


# Indexing time (0-5 sec) - 10.2.1.63





# Indexing time (0-5 sec) - 10.2.1.64



```
evperf2-1.conf
# "EVPERF'2" view performance test:
# 3 ddocs with 8 views per ddoc
# 20M initial items
# 25GB RAM quota (32GB total RAM)
# DGM
# 30 clients
# 8K ops/sec total background load (memcached commands)
# 80% reads, 20% write (12% updates/deletes, 8% inserts)
# Cache miss ratio < 1%
# Stop on 18M total queries (tuned to be 6 hours)

performance.ipperf.MultiClientTests.test_evperf2

params:

# general
batch=50
kind=json
mem_quota=25000
loglevel=error

# load phase
items=20000000
hot_init_items=16000000

# access phase
ratio_sets=0.2
ratio_misses=0.04
ratio_creates=0.40
ratio_deletes=0.50
ratio_hot=0.2
ratio_hot_gets=0.975
ratio_hot_sets=0.975
ratio_expirations=0.0
bg_max_ops_per_sec=265
fg_max_ops=18000000
total_clients=30
start_delay=5

# 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
```

```
terra.ini
[global]
username:root
password:couchbase
port:8091
data_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
```