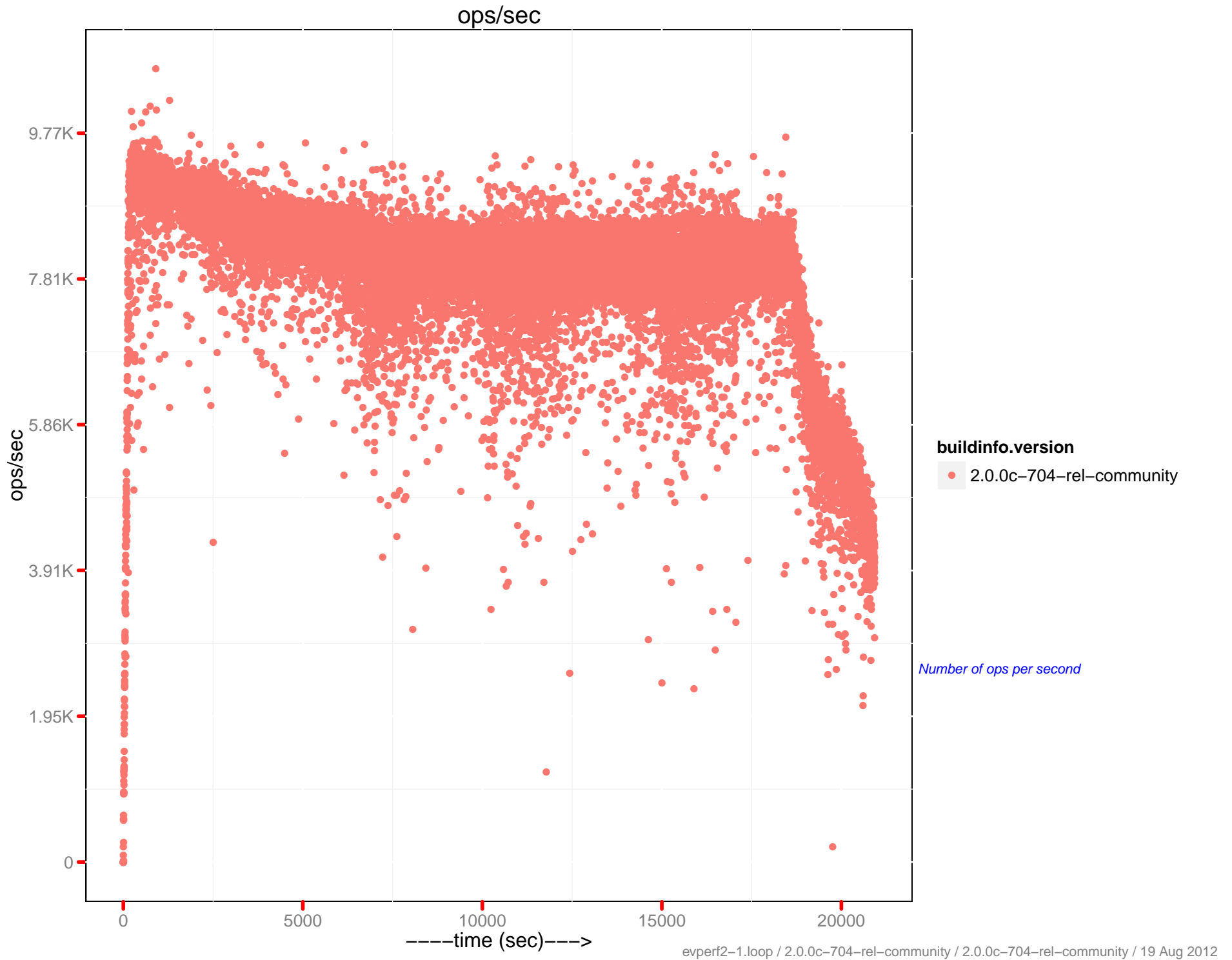
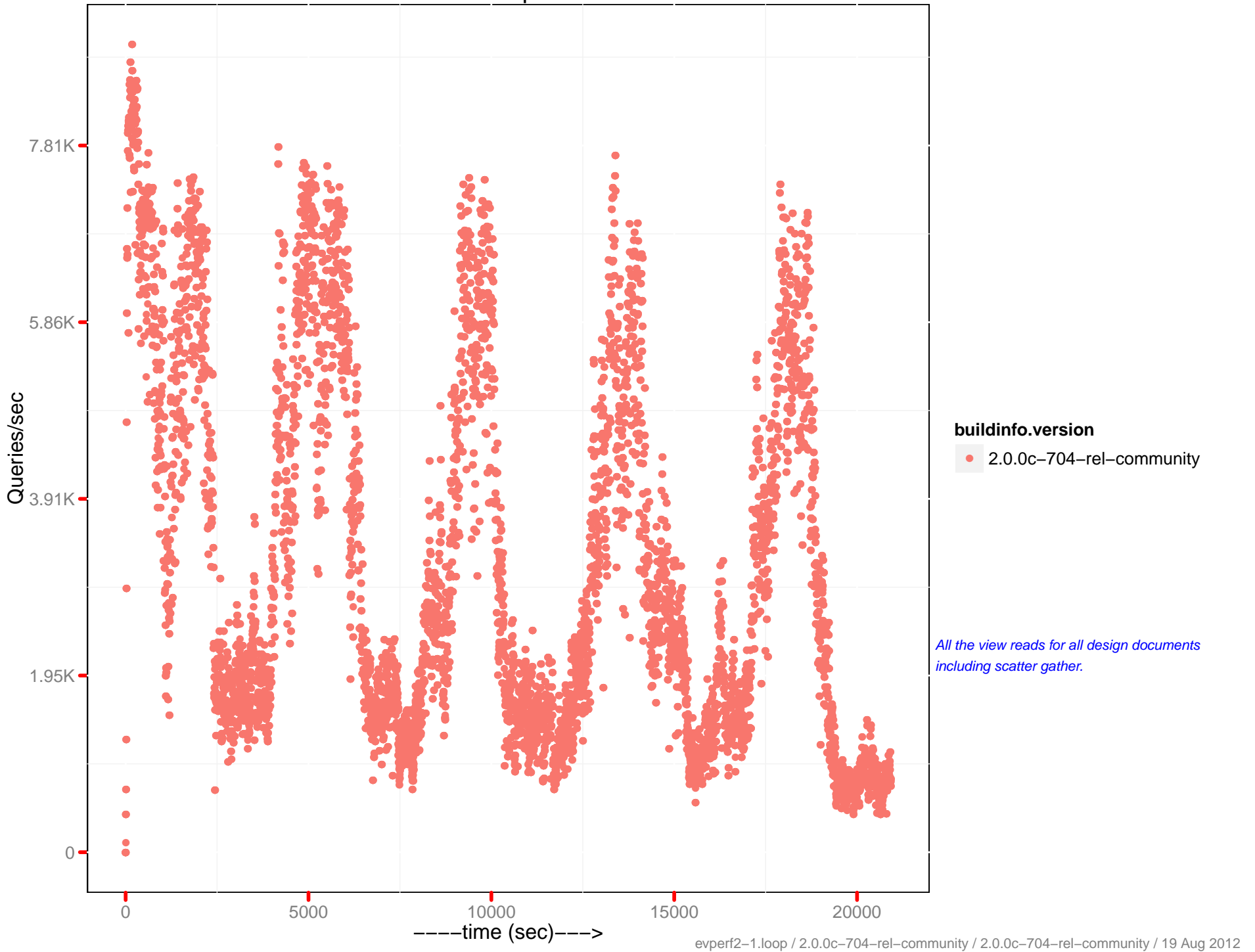


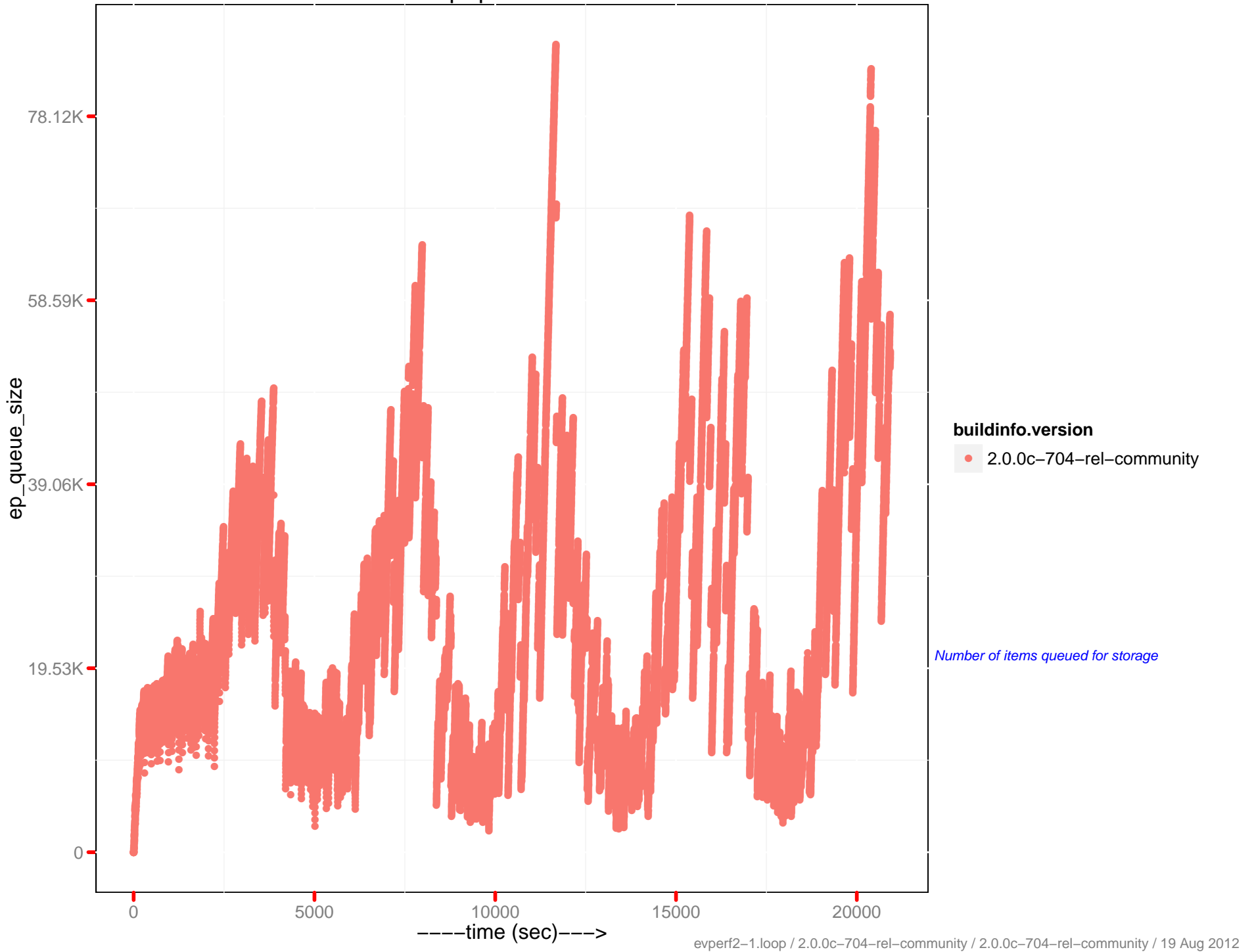
	2.0.0c – 704	2.0.0c – 704
<i>Runtime (in hr)</i>	5.82	NA
<i>Avg. Drain Rate</i>	194.92	NANA
<i>Peak Disk (GB)</i>	116.55	NA
<i>Peak Memory (GB)</i>	19.94	NA
<i>Avg. OPS</i>	7.85K	NANA
<i>Avg. mem memcached (GB)</i>	18.99	NA
<i>Avg. mem beam.smp (MB)</i>	458.79	NA
<i>Latency-get (90th) (ms)</i>	1.24	NA
<i>Latency-get (95th) (ms)</i>	3.26	NA
<i>Latency-get (99th) (ms)</i>	33.46	NA
<i>Latency-set (90th) (ms)</i>	1.29	NA
<i>Latency-set (95th) (ms)</i>	3.4	NA
<i>Latency-set (99th) (ms)</i>	23.69	NA
<i>Latency-query (80th) (ms)</i>	20.72	NA
<i>Latency-query (90th) (ms)</i>	51.59	NA
<i>Latency-query (95th) (ms)</i>	110.39	NA
<i>Latency-query (99th) (ms)</i>	338.54	NA
<i>Latency-query (99.9th) (ms)</i>	733.63	NA
<i>Avg. QPS</i>	885.62	NA
<i>Rebalance Time (sec)</i>	0	NA
<i>Testrunner Version</i>	2c2d570	NA



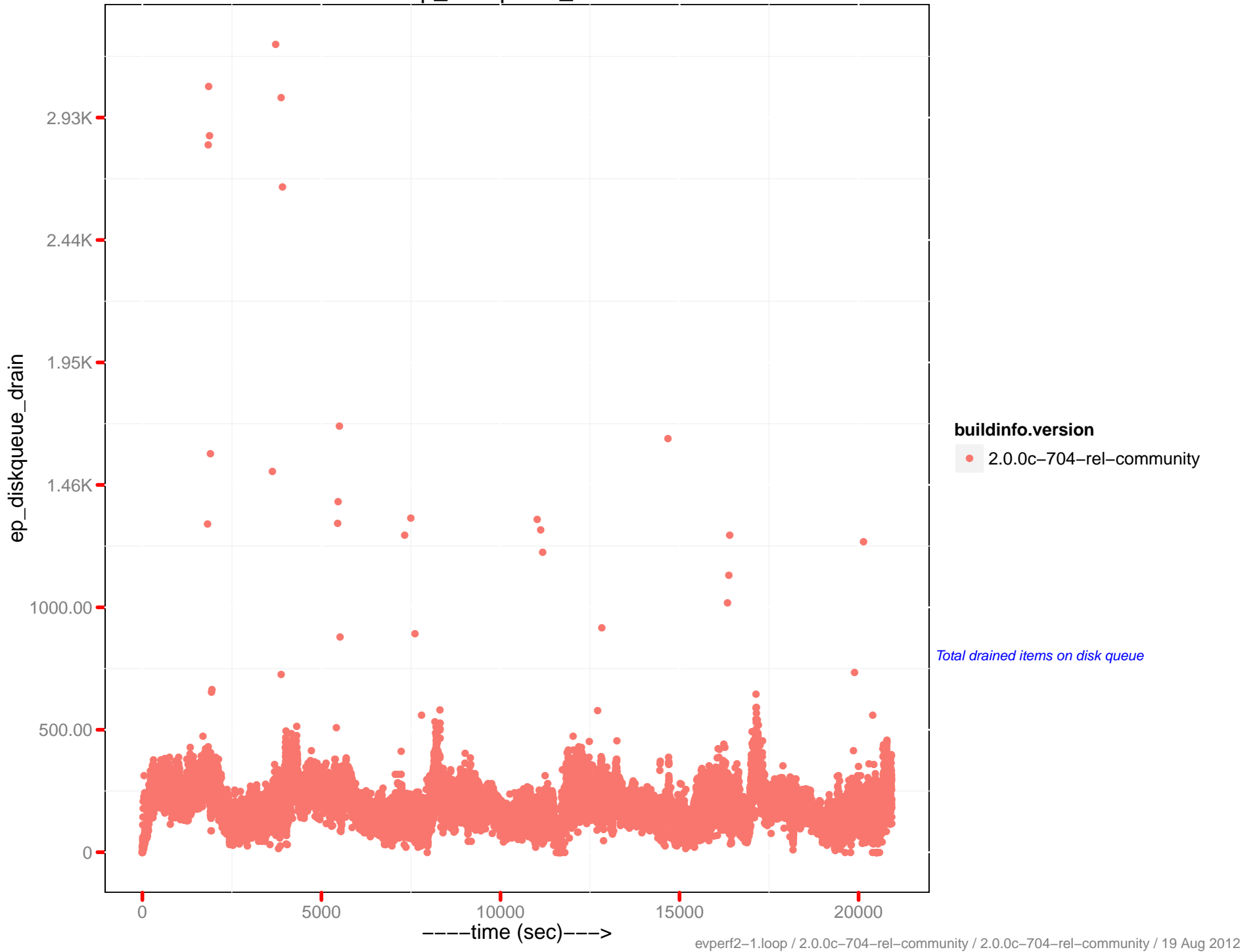
View read per sec.



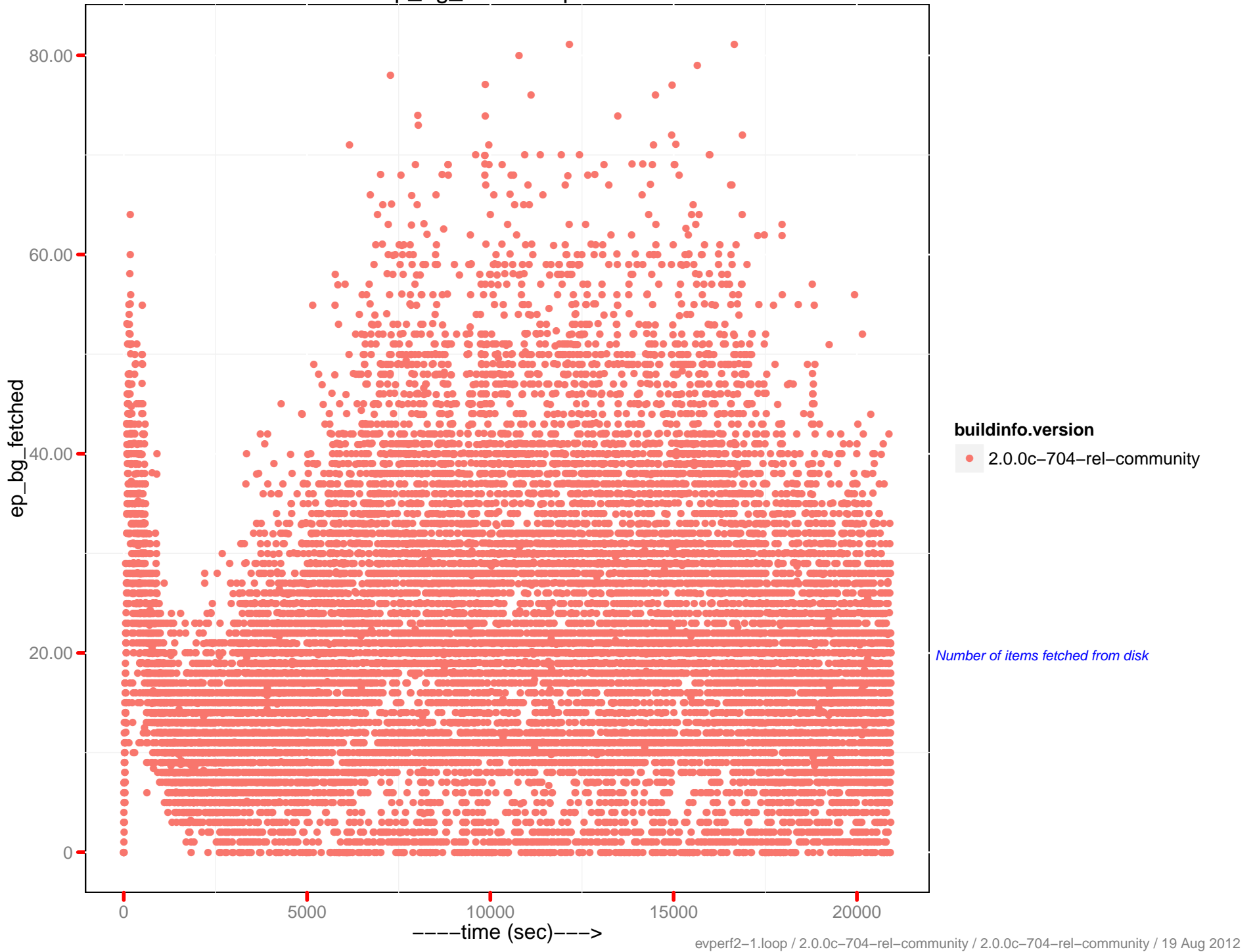
ep queue size



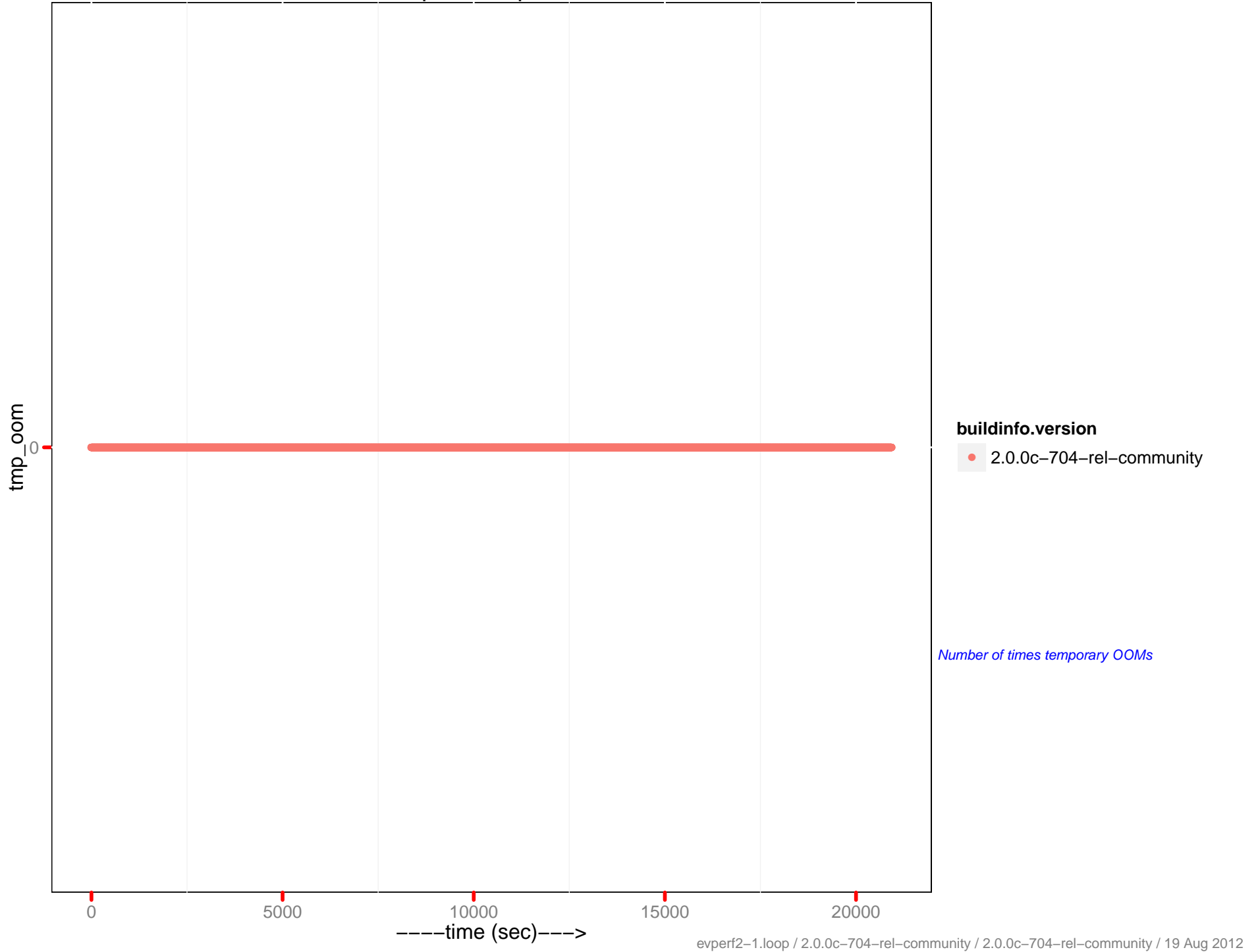
ep_diskqueue_drain



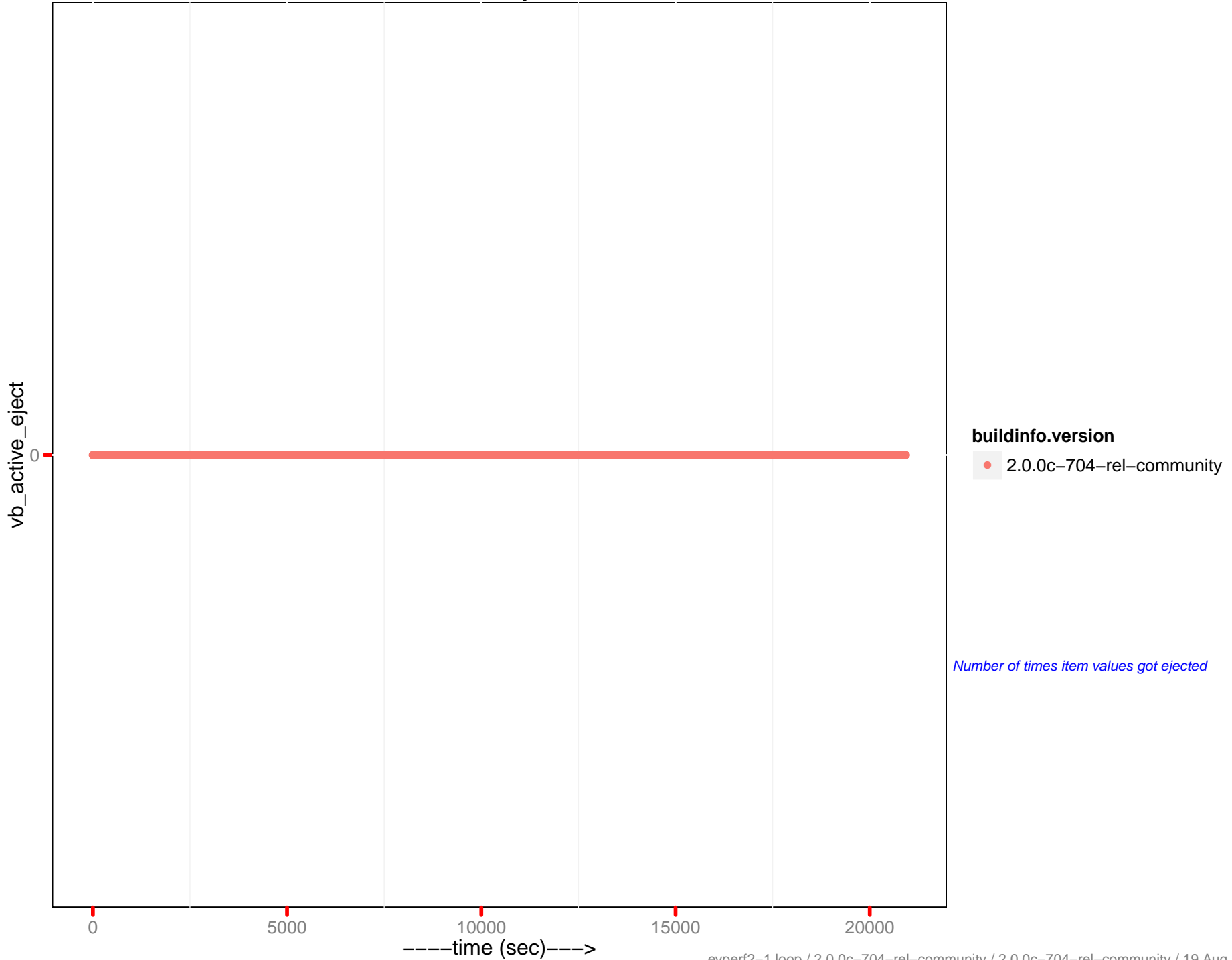
ep_bg_fetched ops/sec



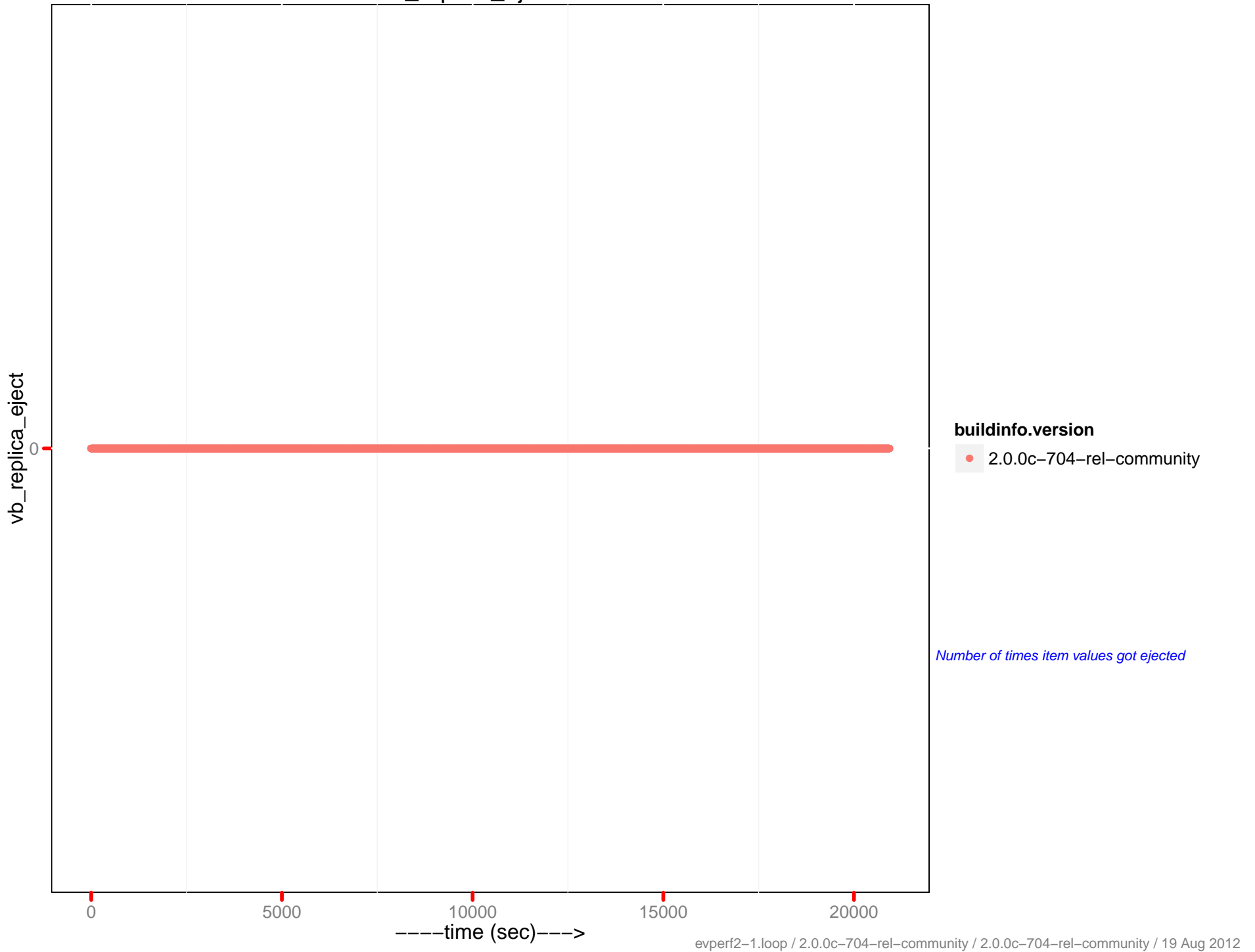
tmp_oom ops/sec



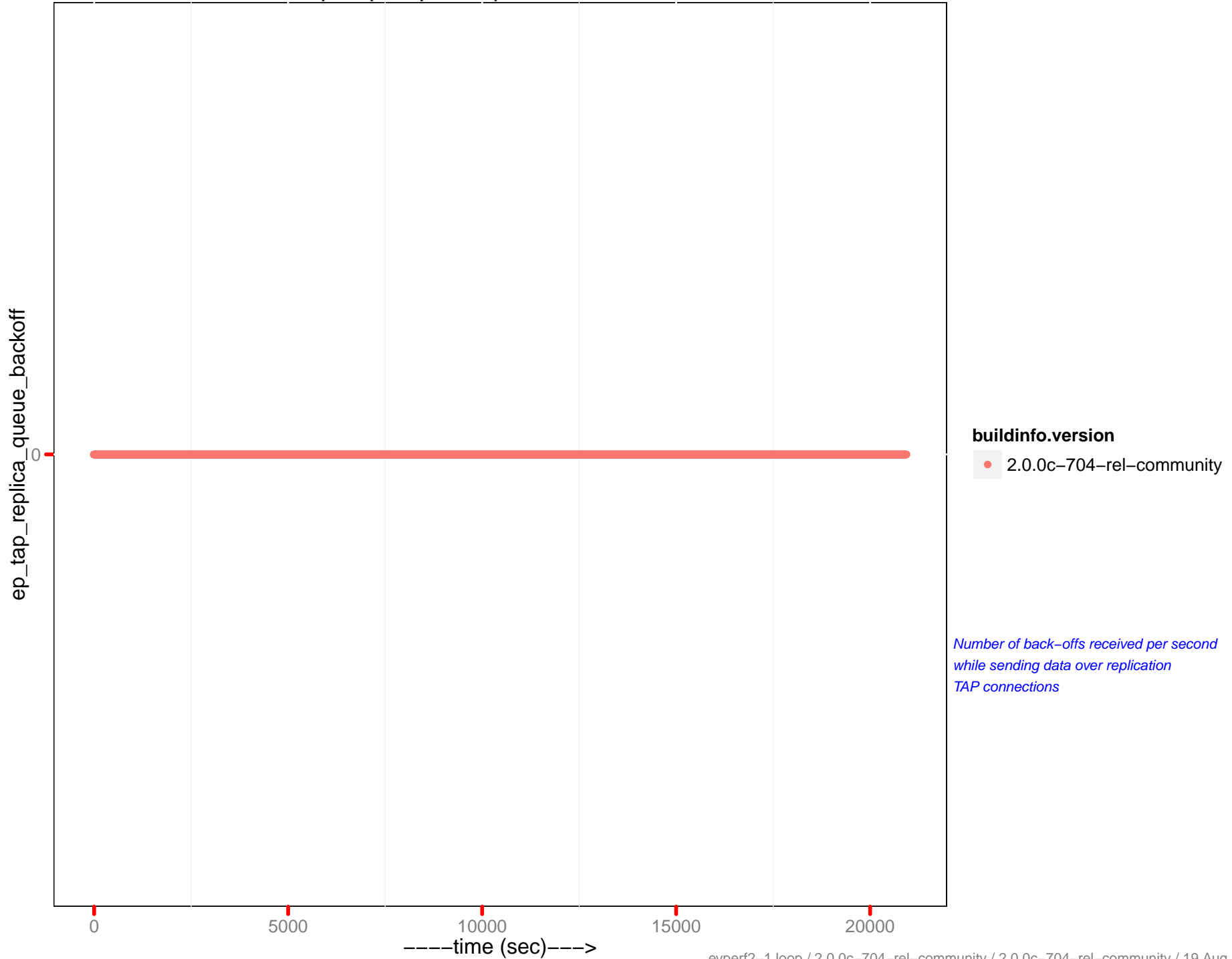
vb_active_eject/sec



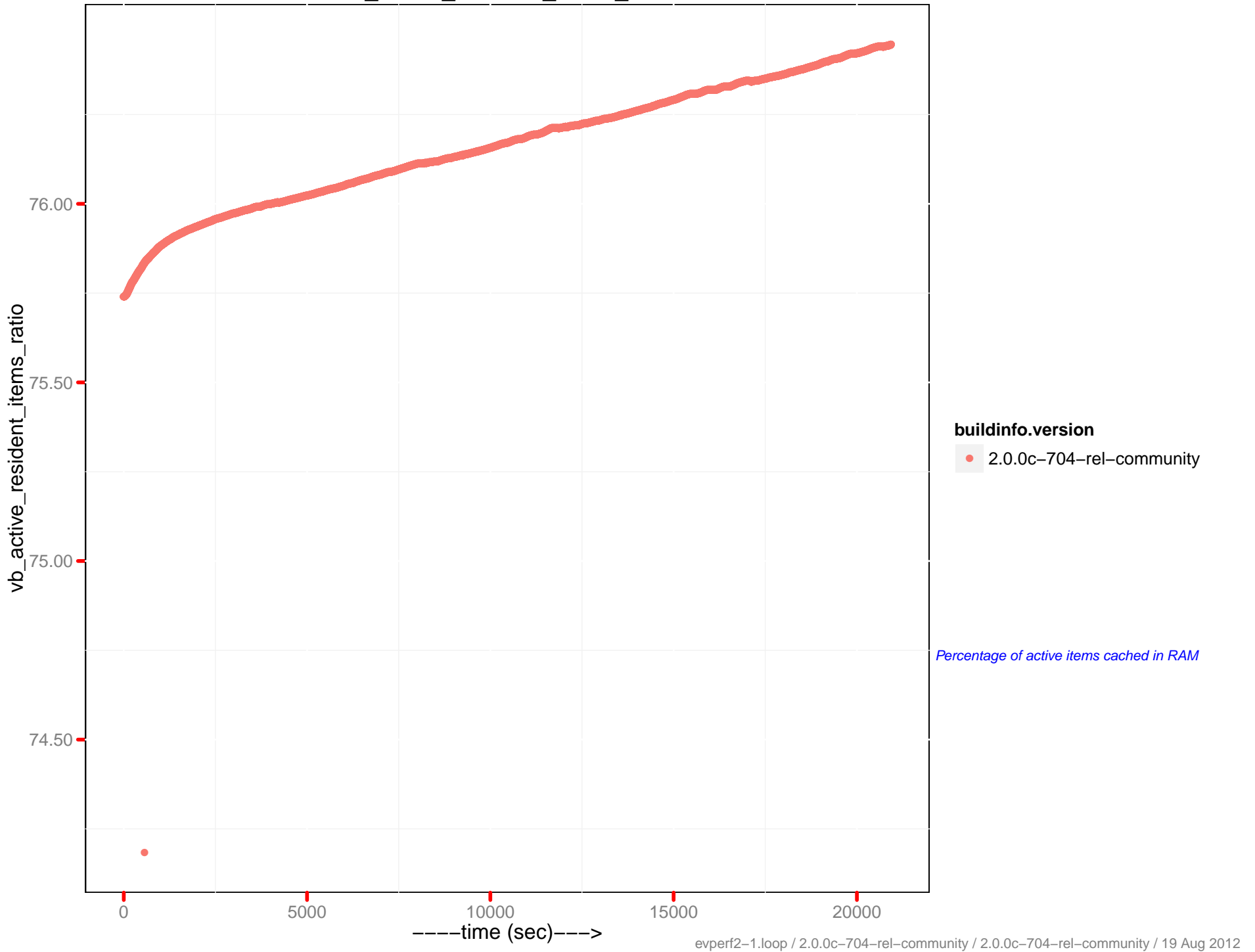
vb_replica_eject/sec



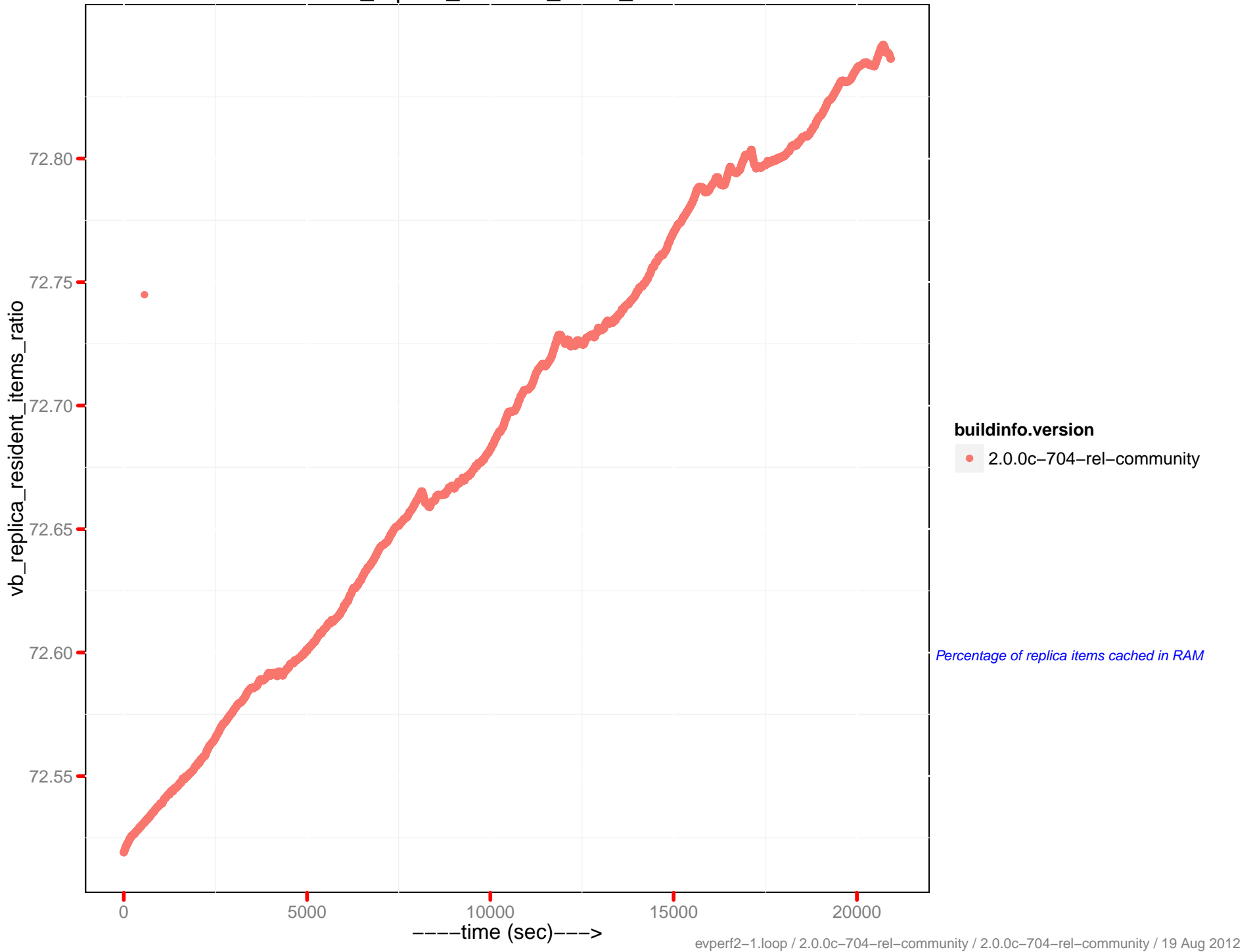
ep_tap_replica_queue_backoff/sec



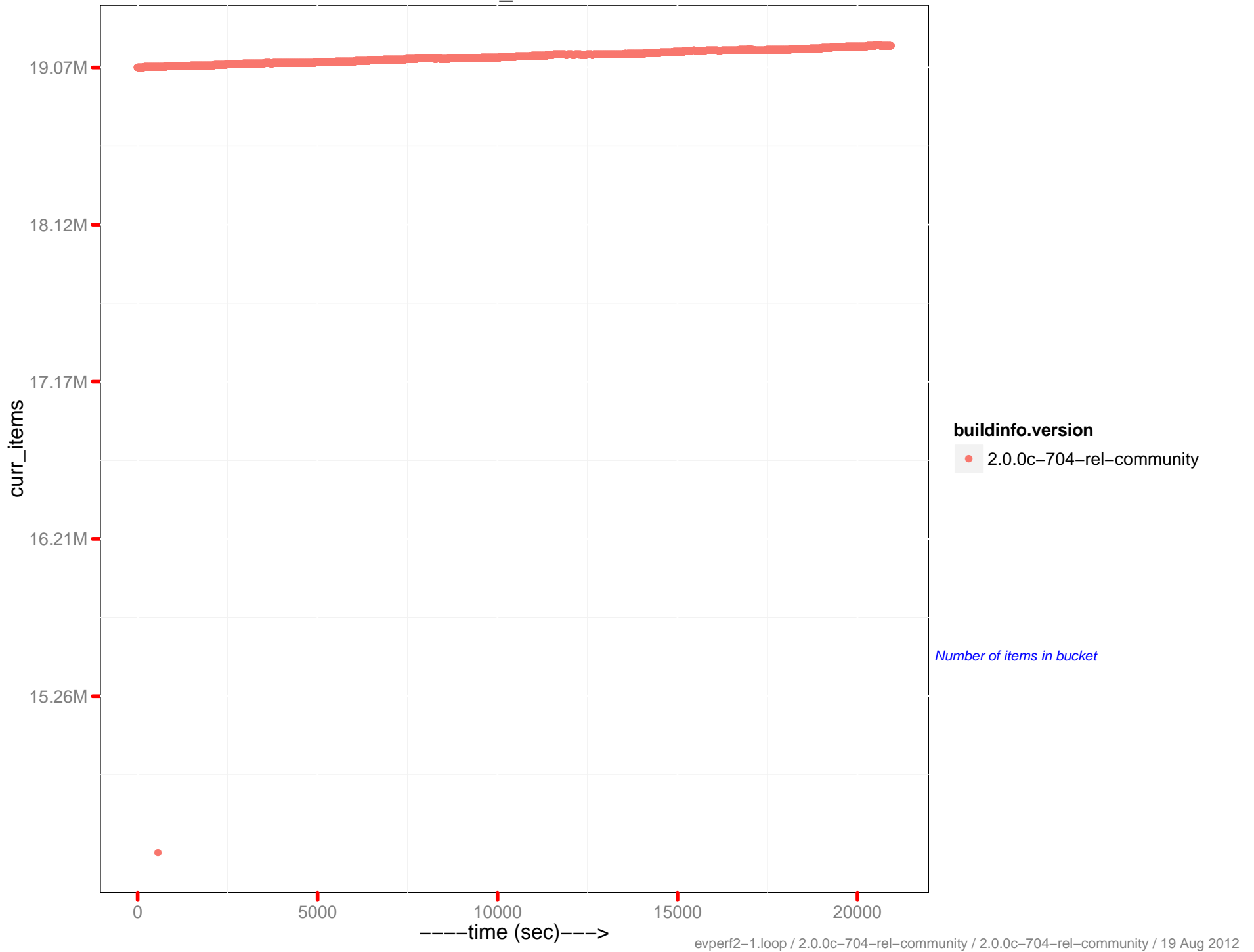
vb_active_resident_items_ratio



vb_replica_resident_items_ratio

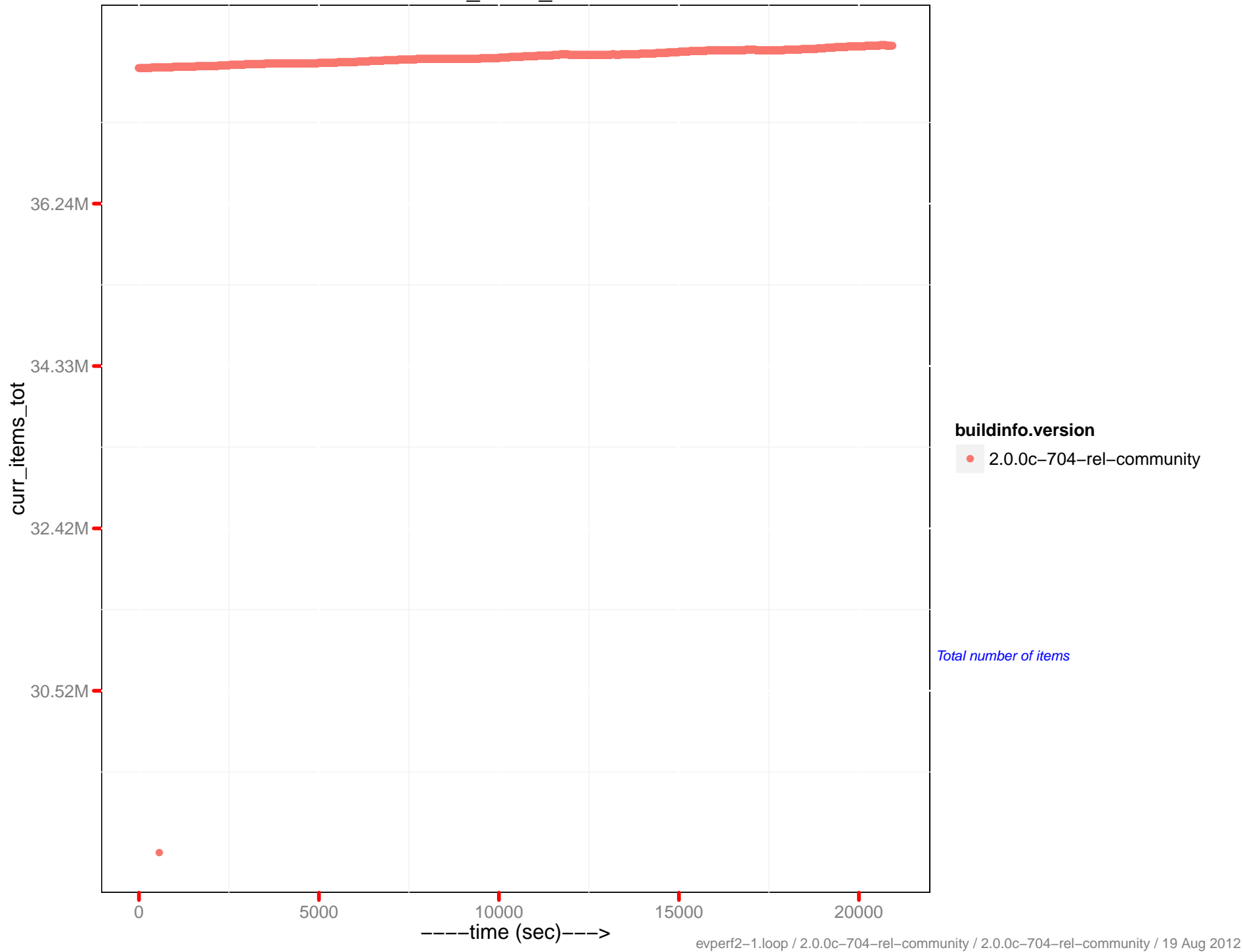


curr_items

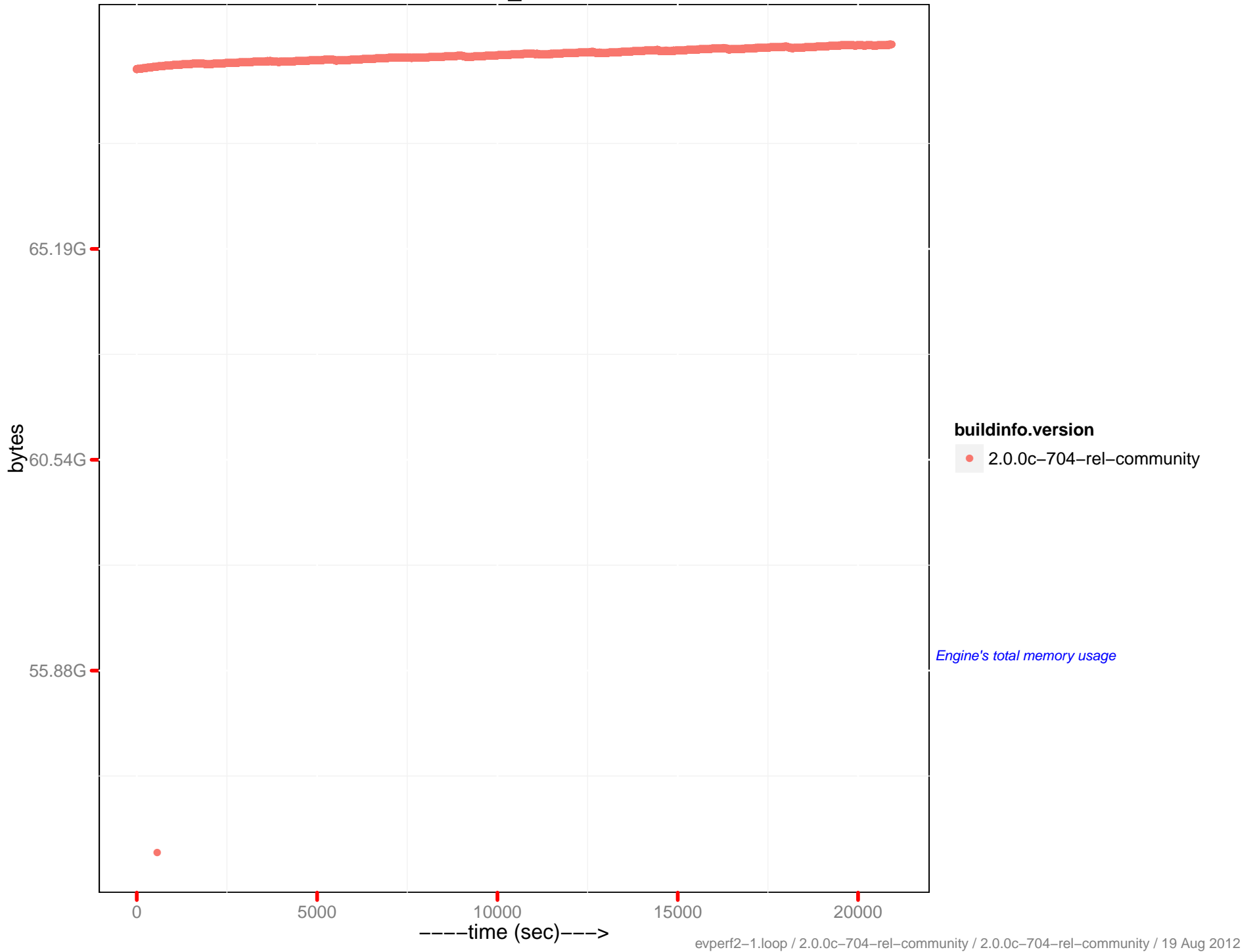


Number of items in bucket

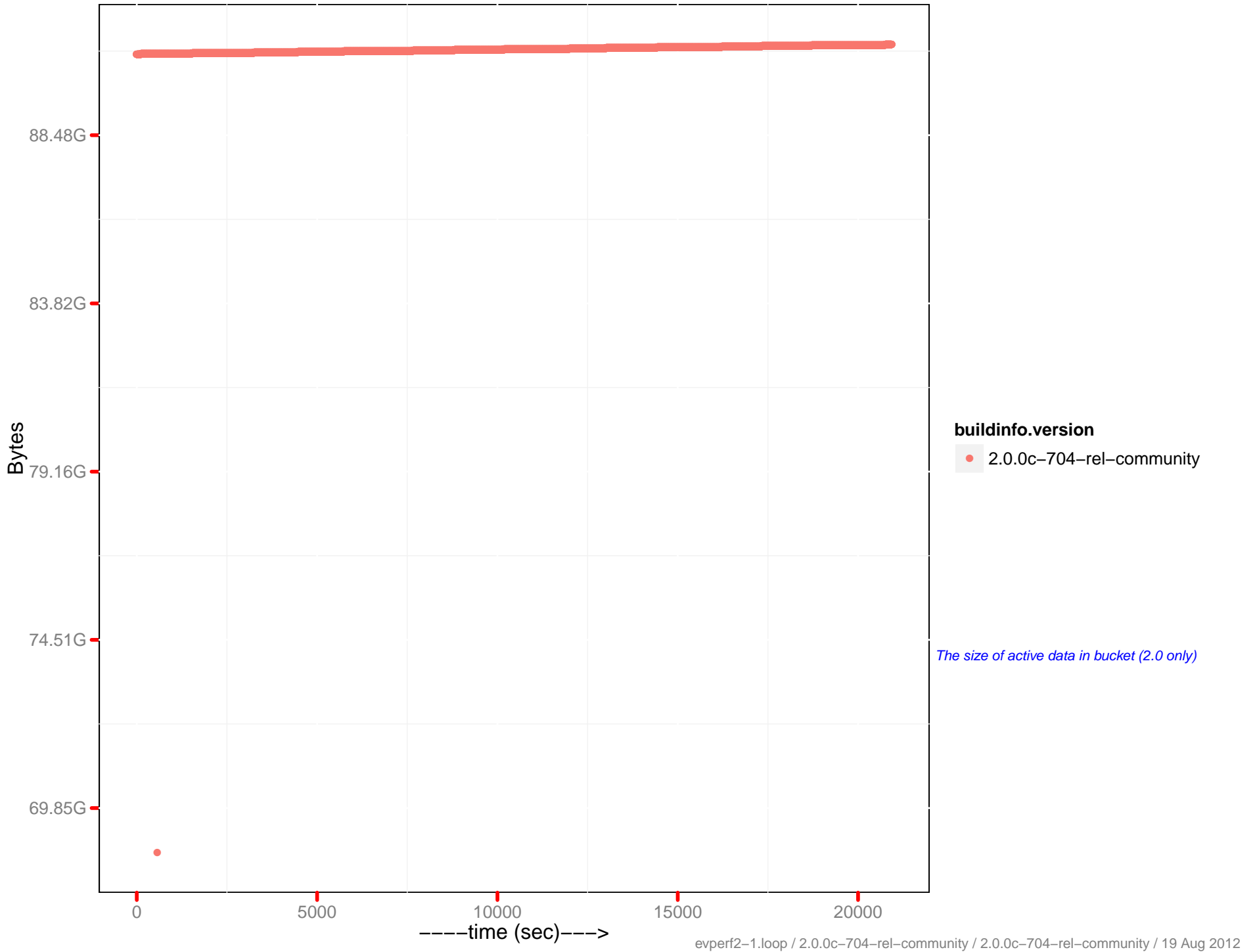
cur_items_total



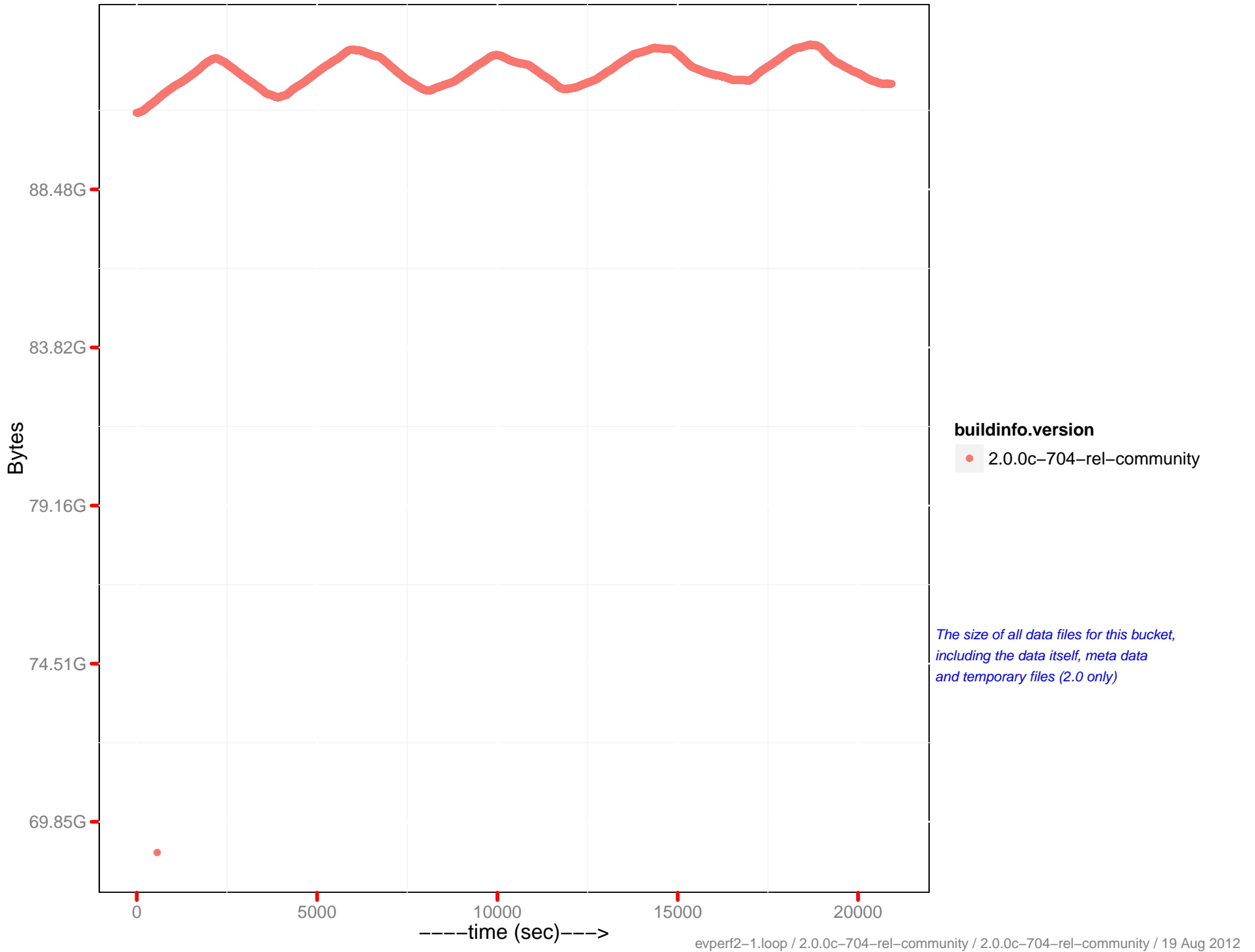
mem_used



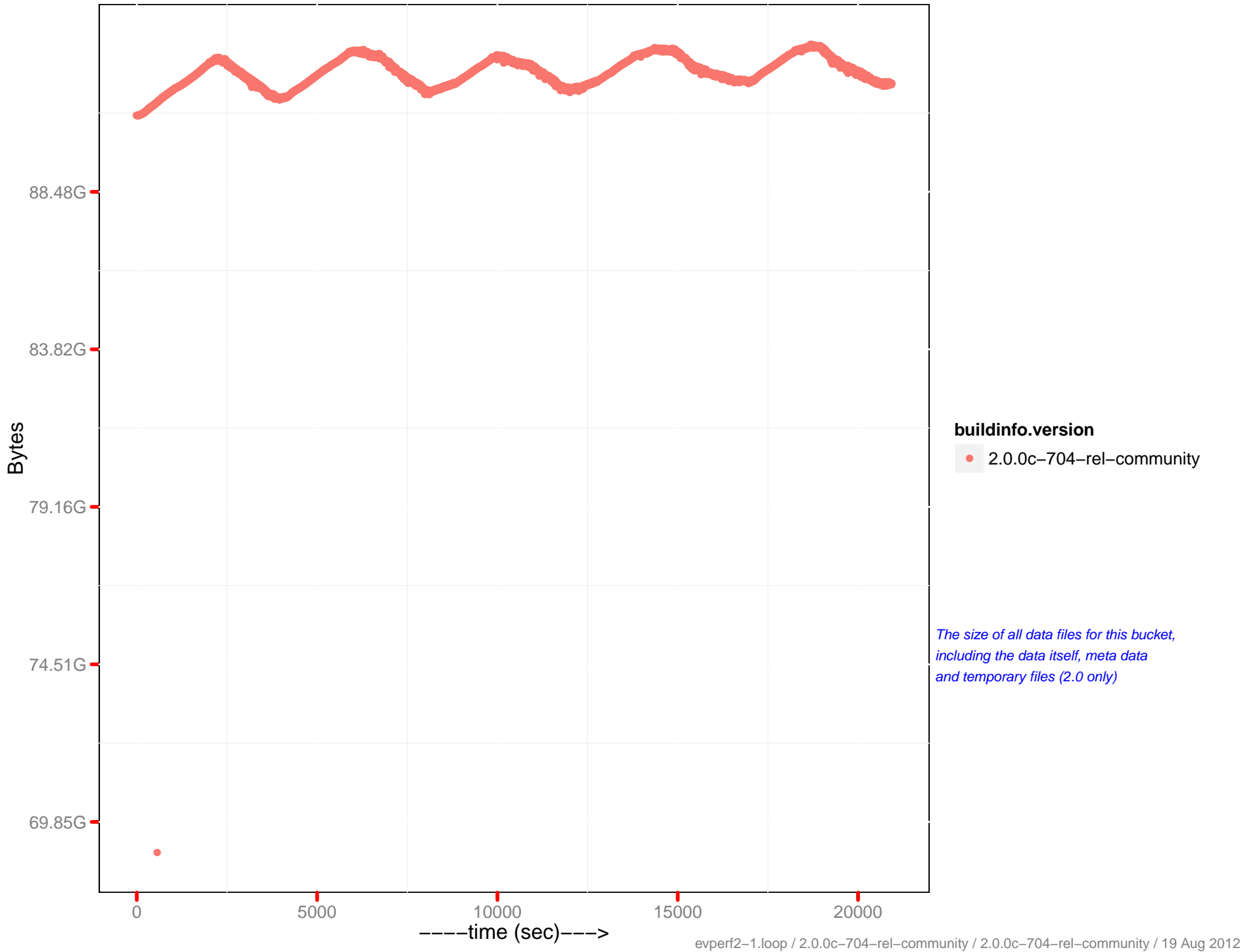
Docs data size



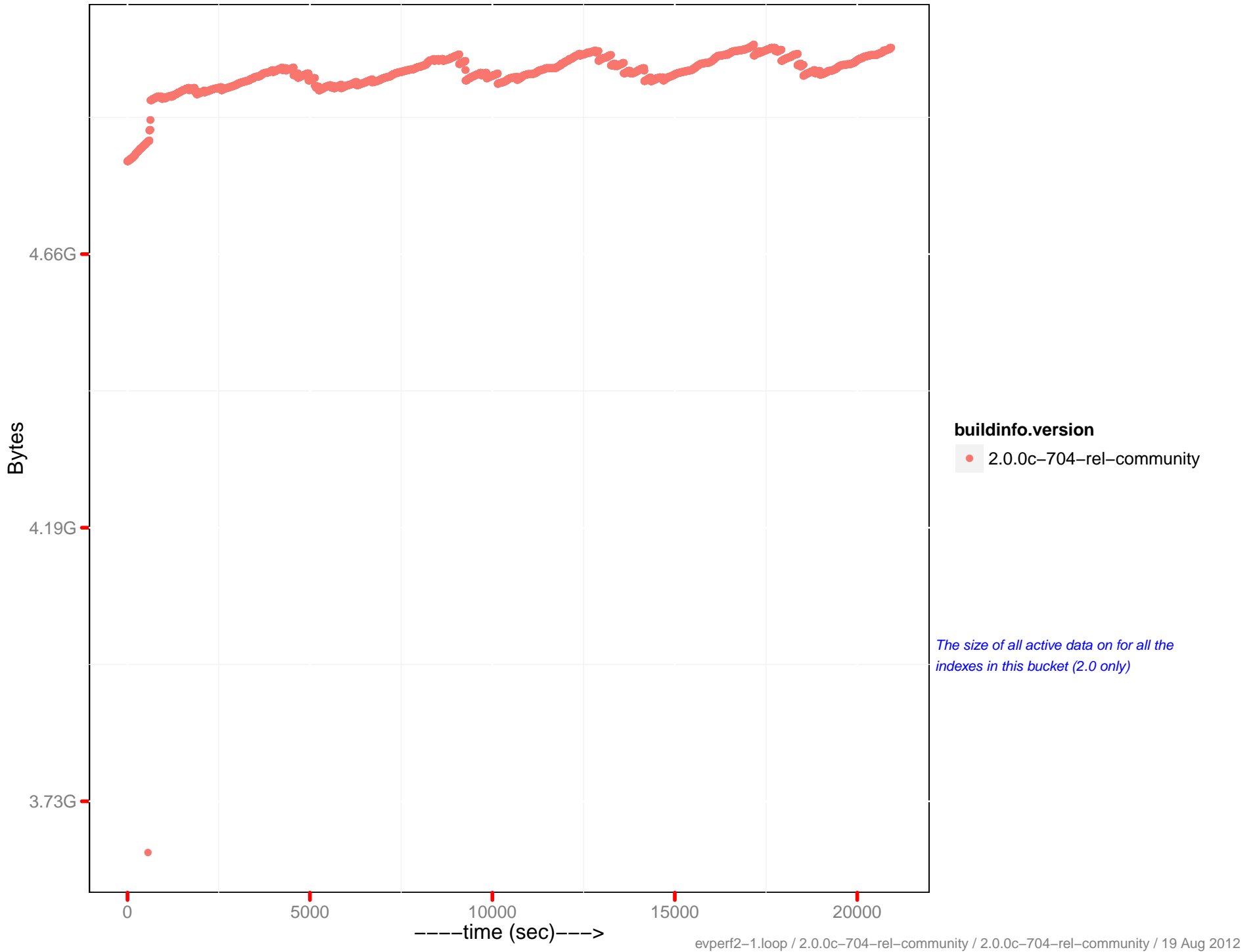
Docs disk size



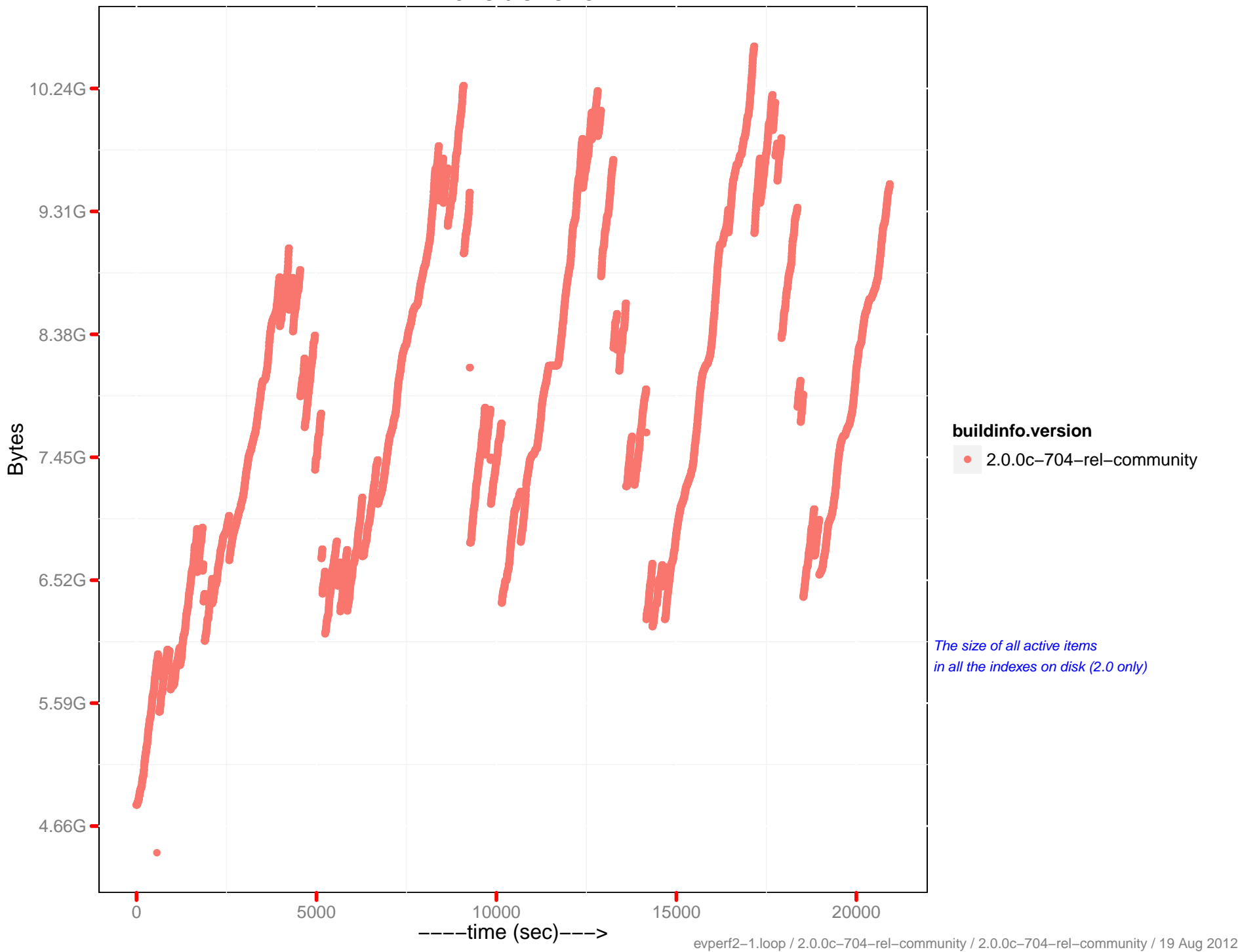
Docs actual disk size



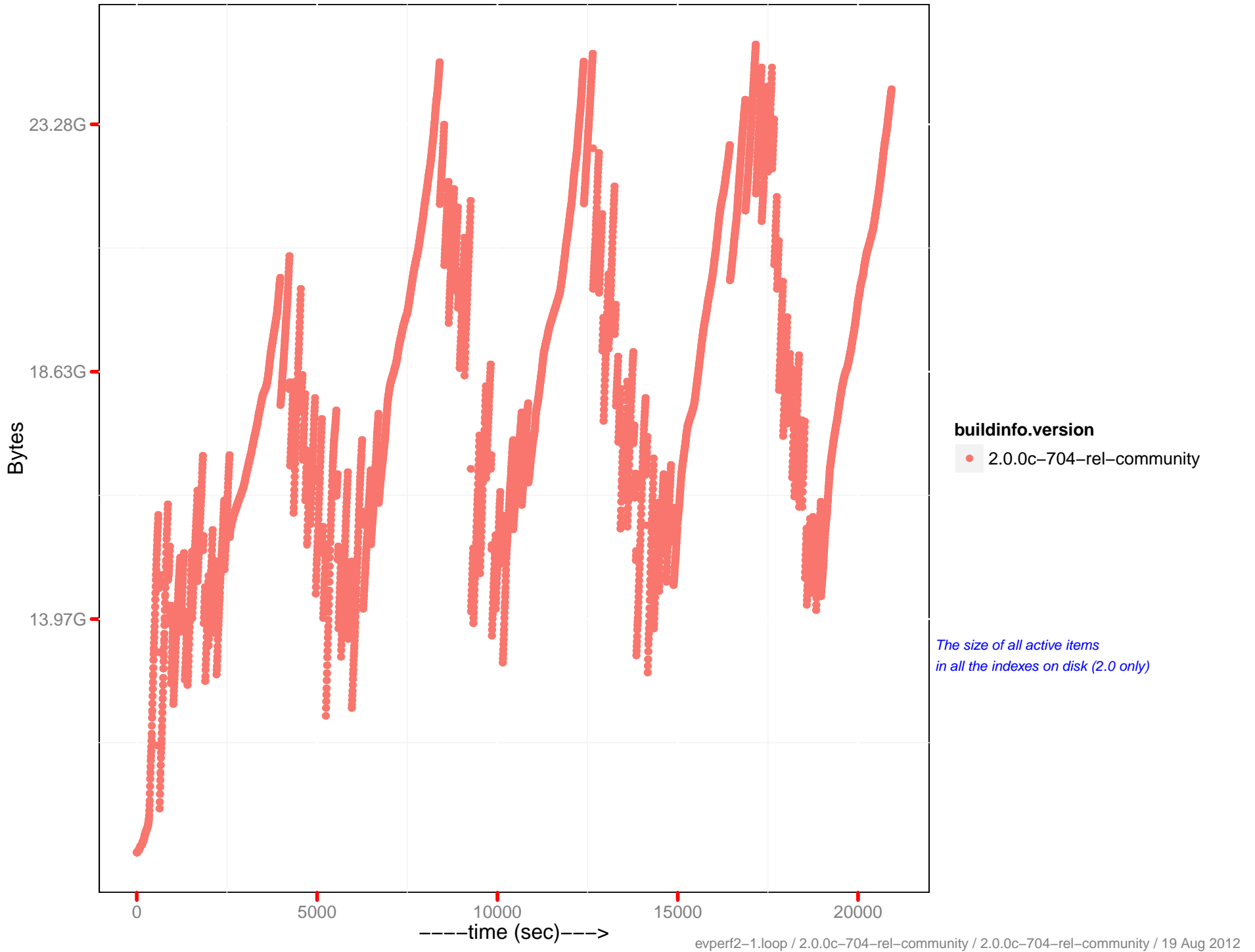
Views data size



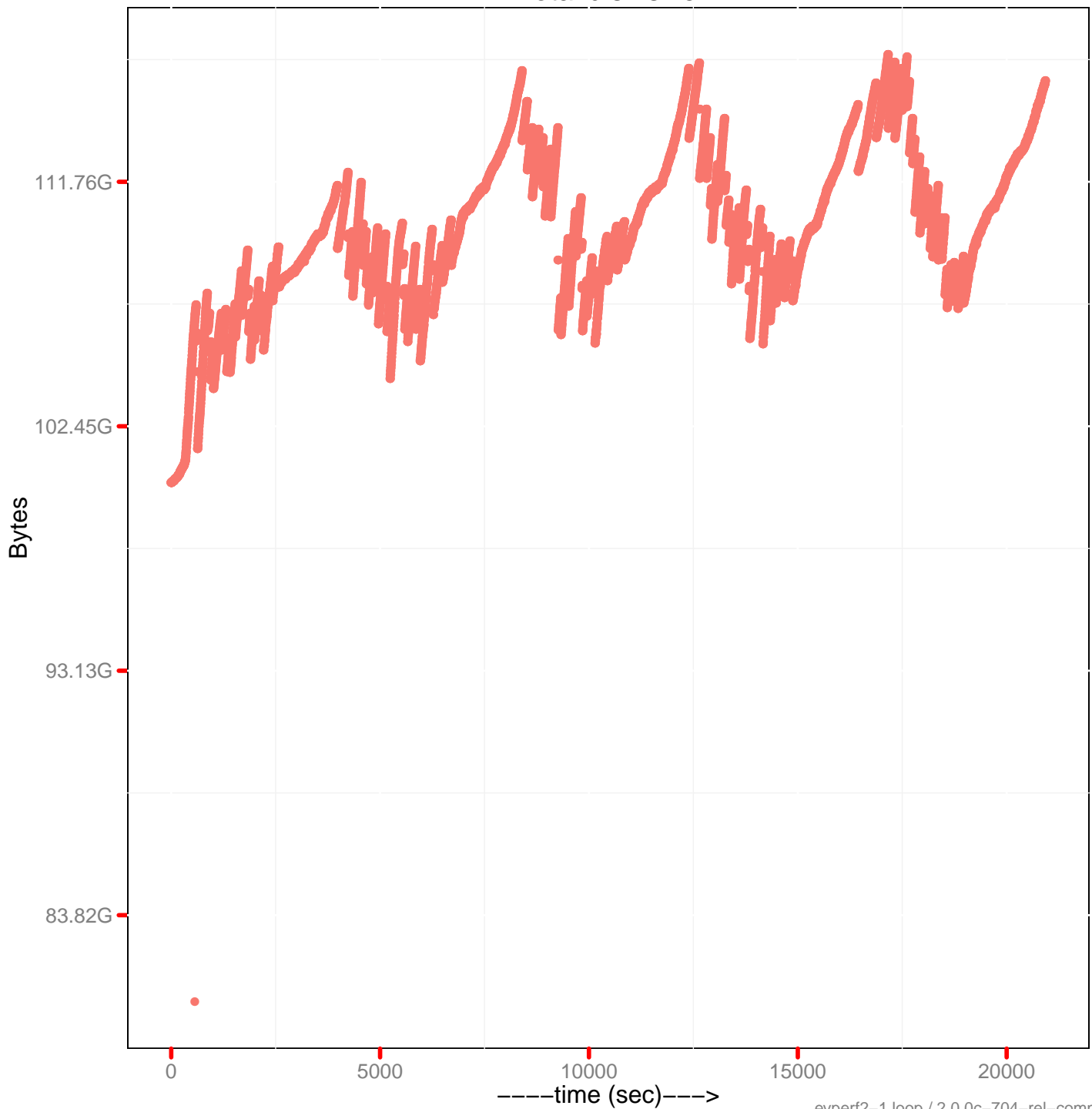
Views disk size



Views actual disk size



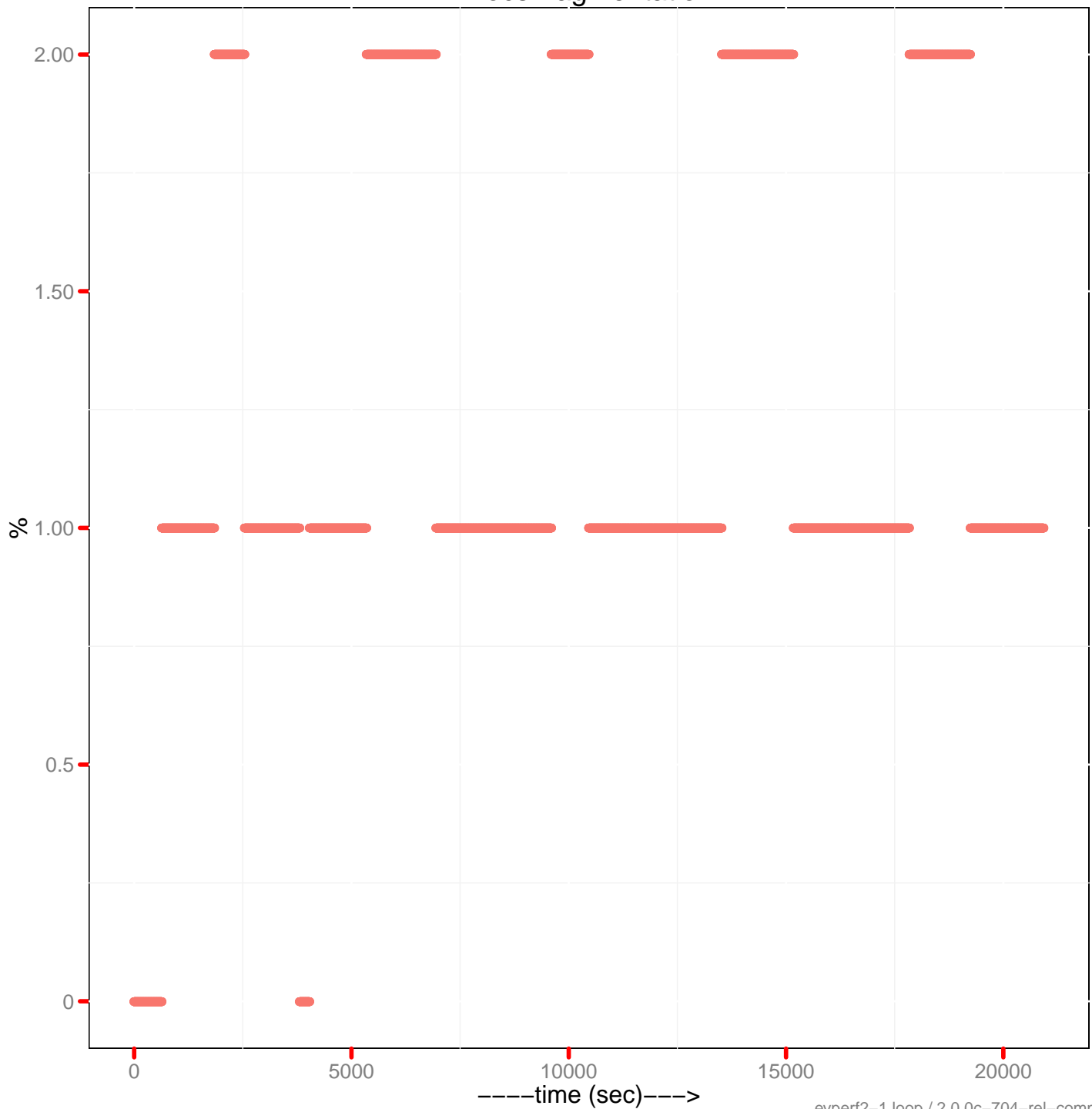
Total disk size



buildinfo.version
● 2.0.0c-704-rel-community

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

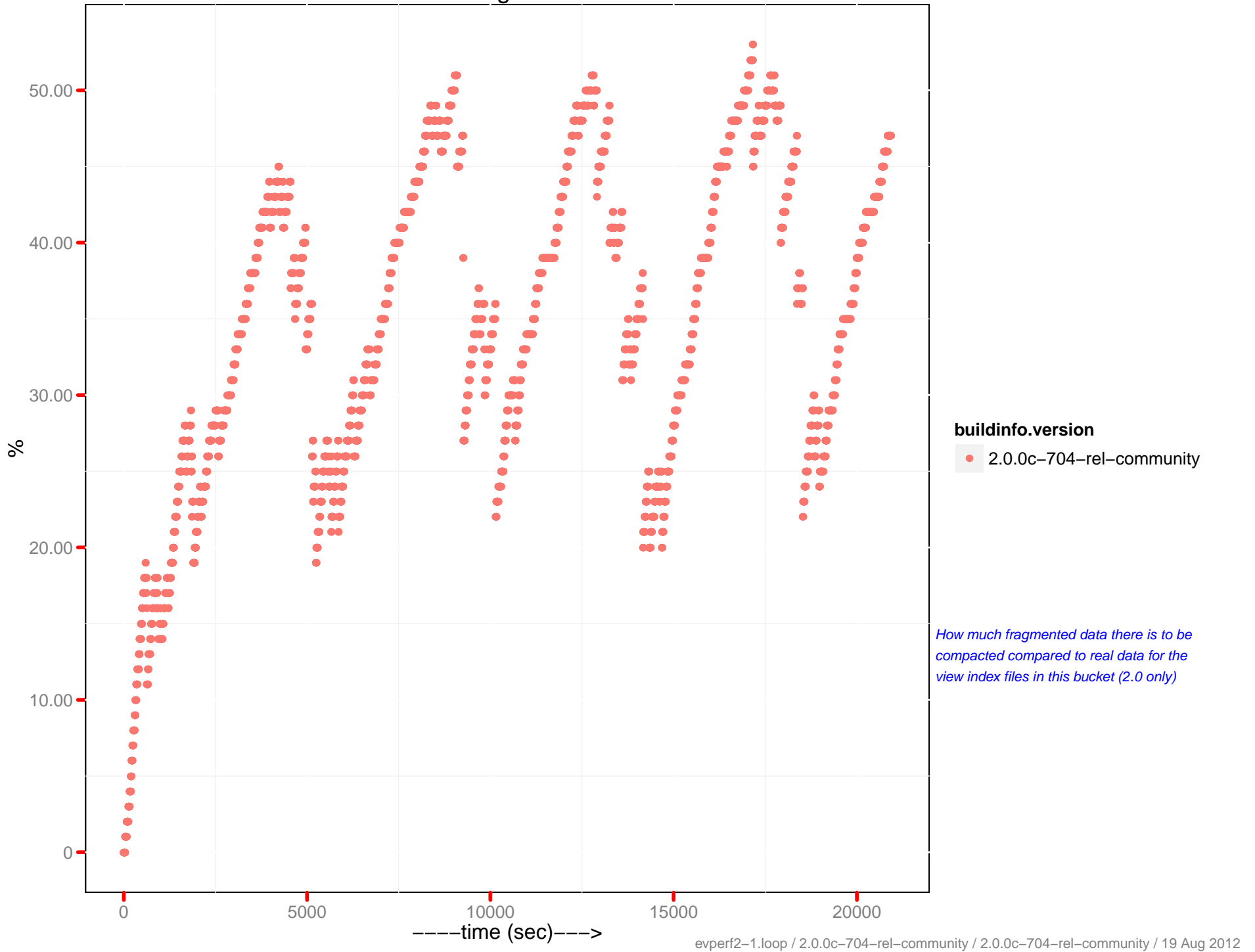
Docs fragmentation



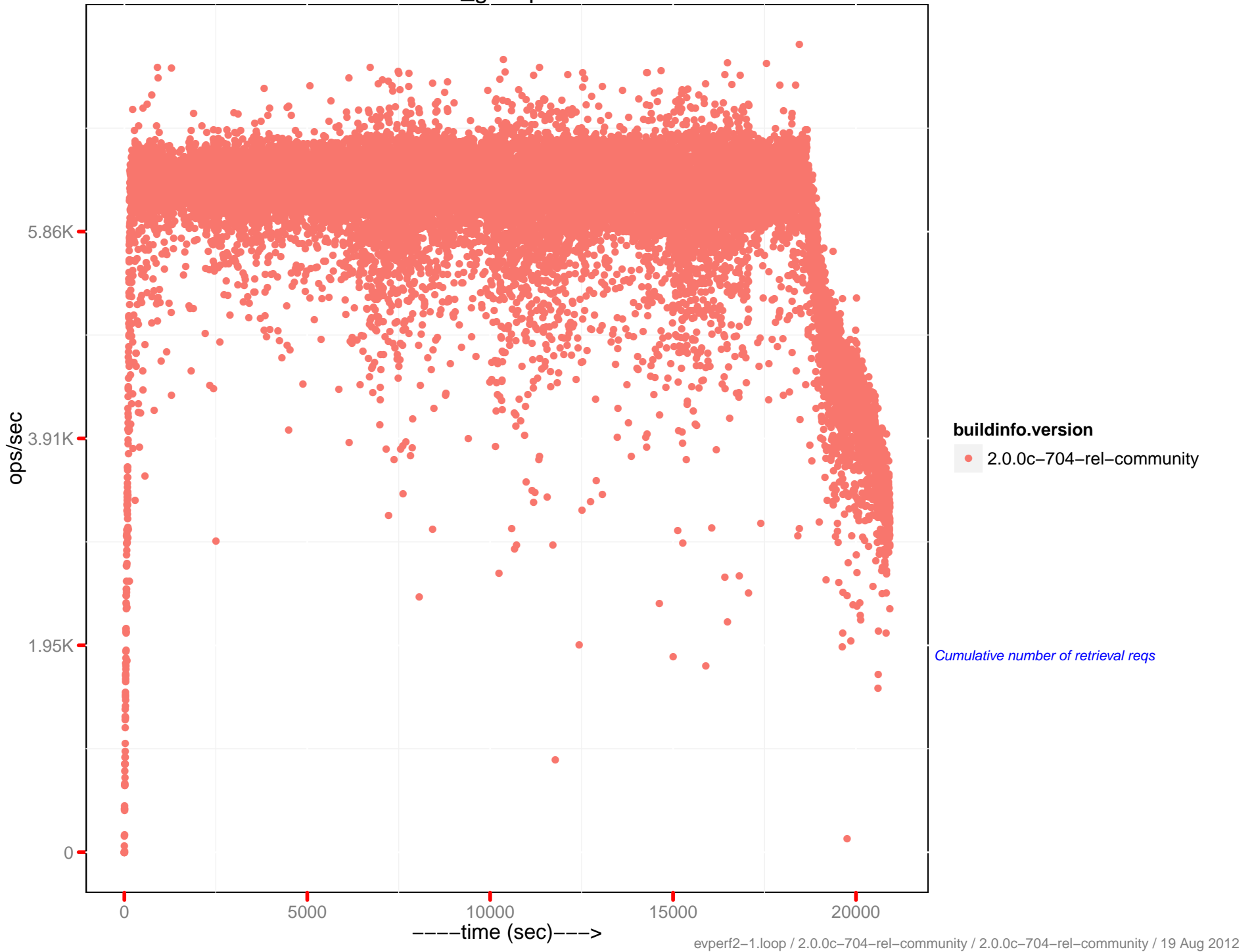
buildinfo.version
• 2.0.0c-704-rel-community

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

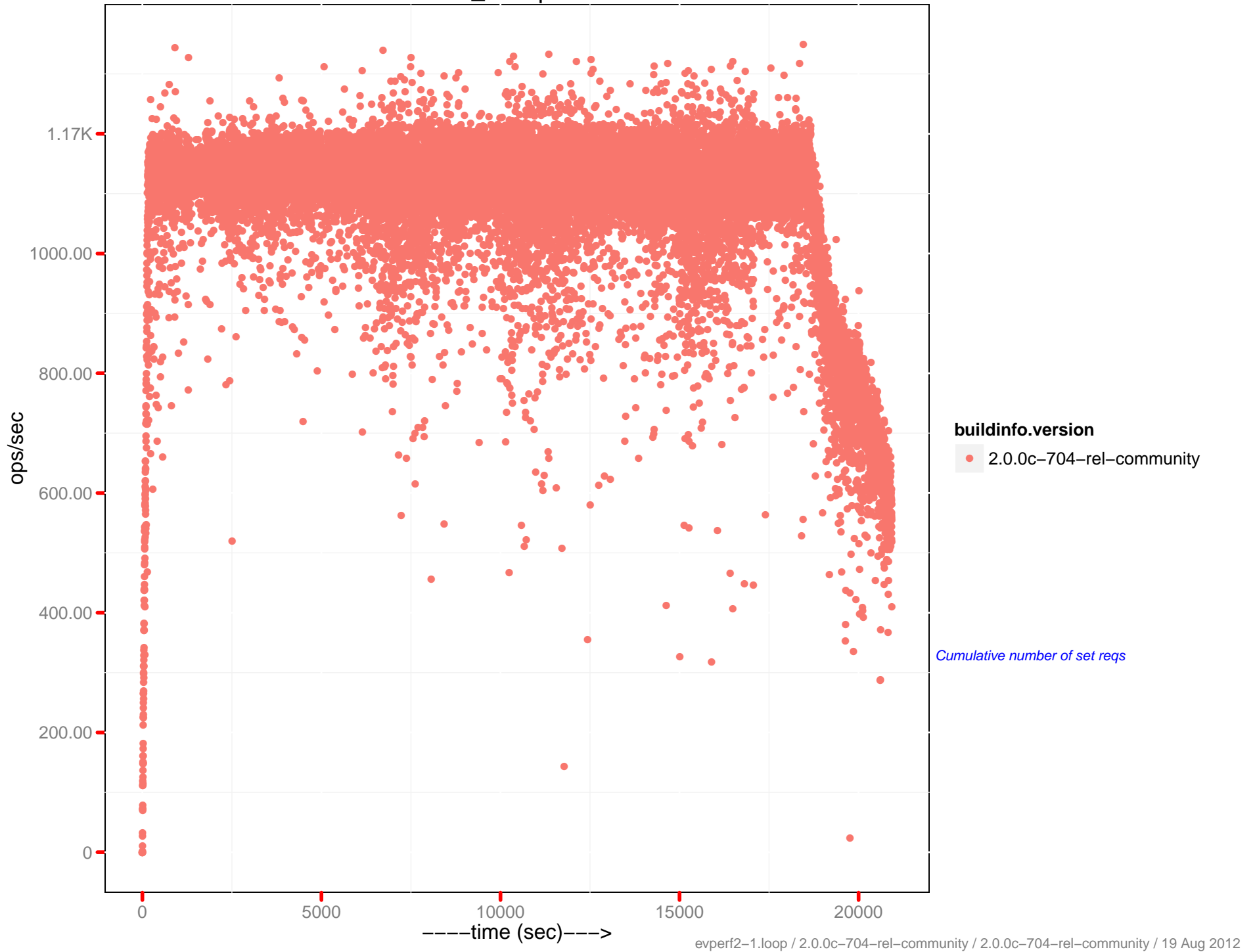
Views fragmentation



cmd_get ops/sec



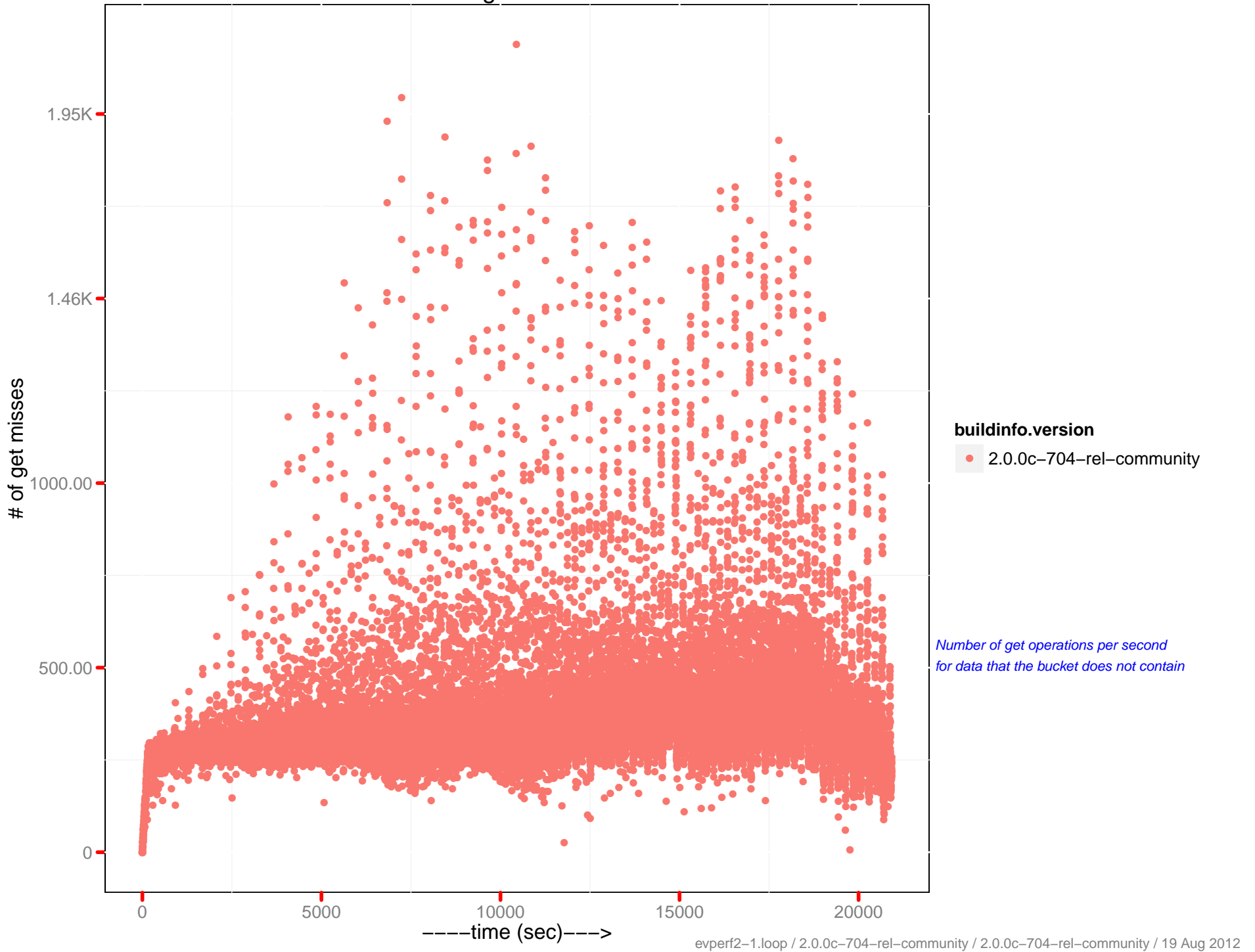
cmd_set ops/sec



buildinfo.version
● 2.0.0c-704-rel-community

Cumulative number of set reqs

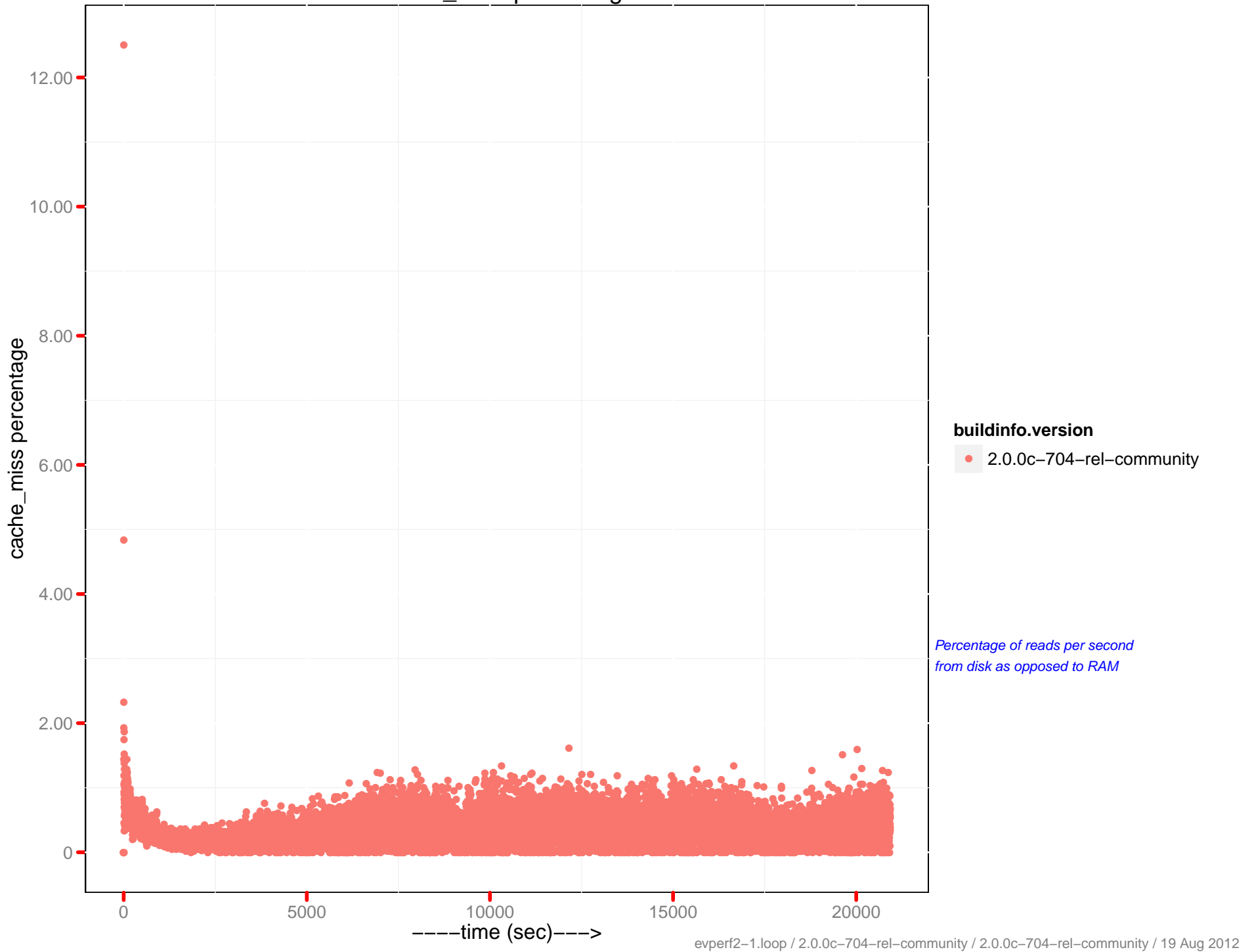
of get misses



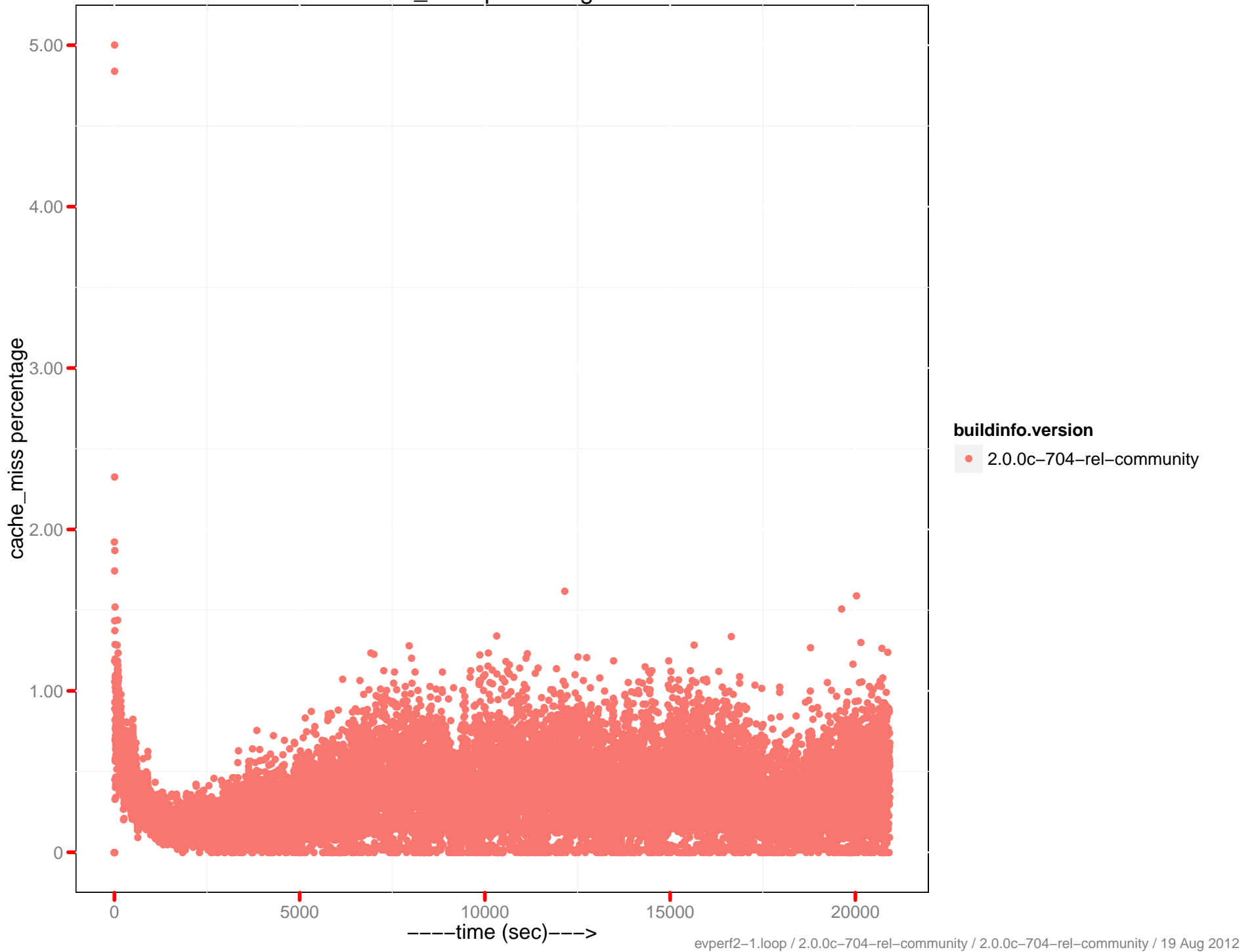
of get hits



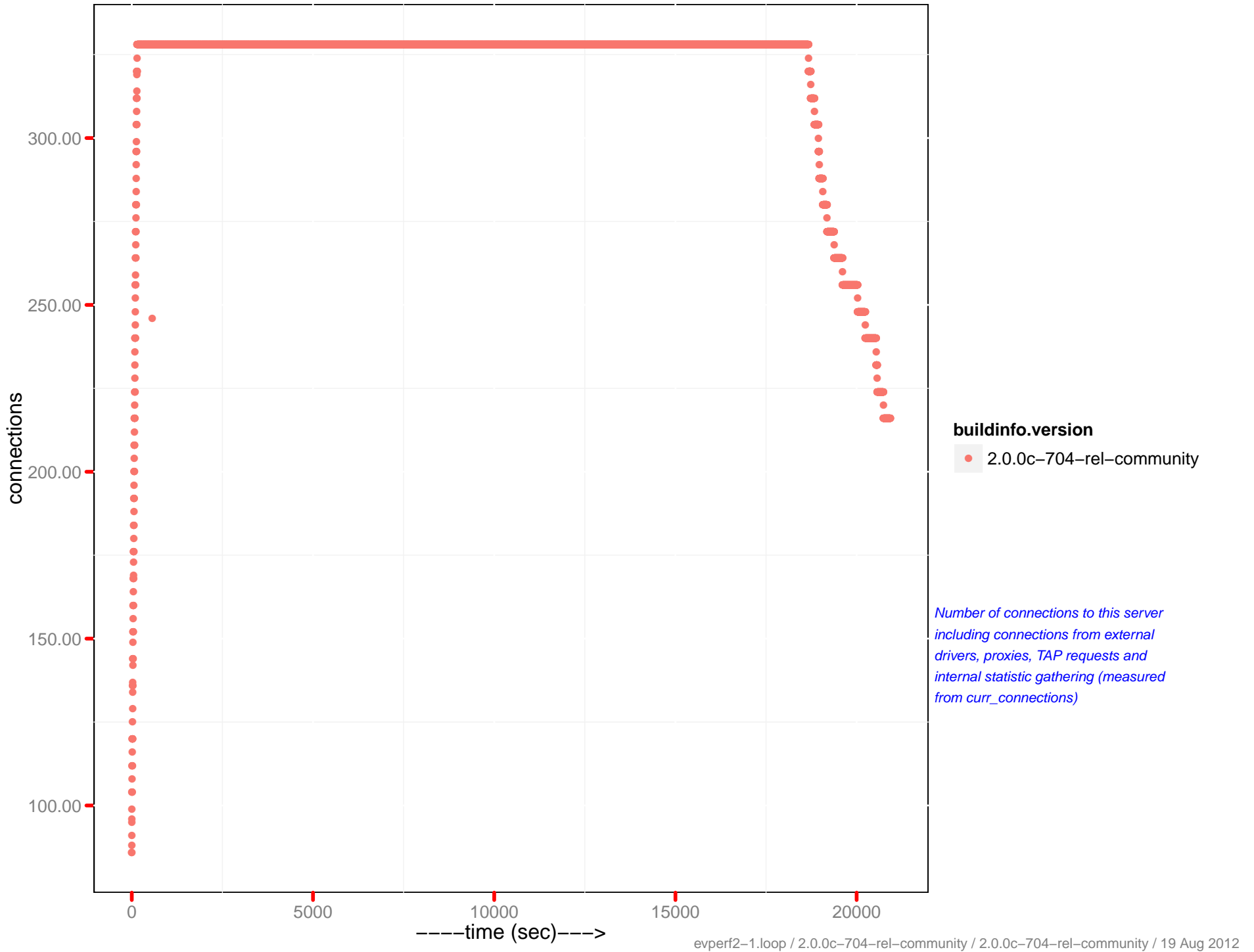
cache_miss percentage



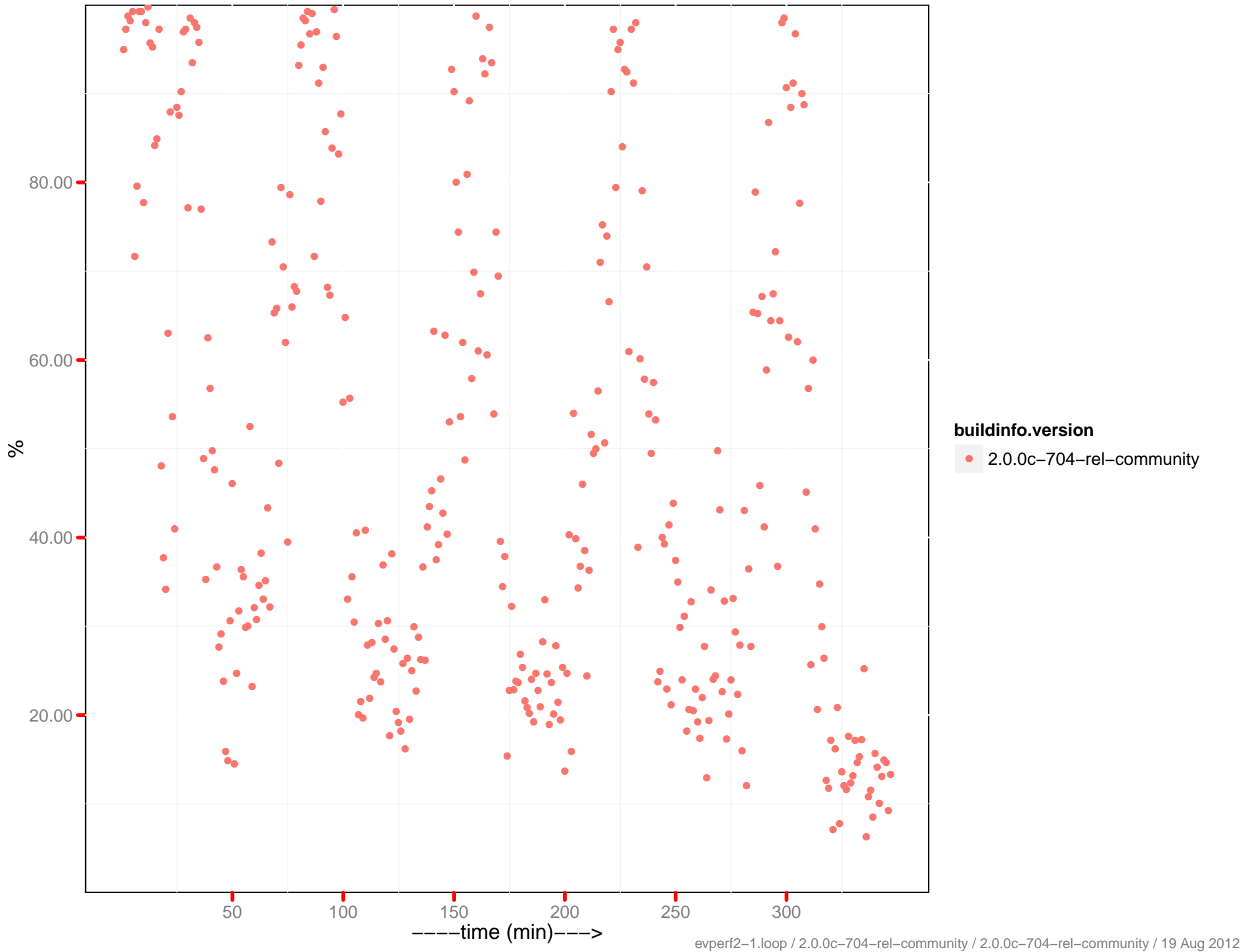
cache_miss percentage 0-5



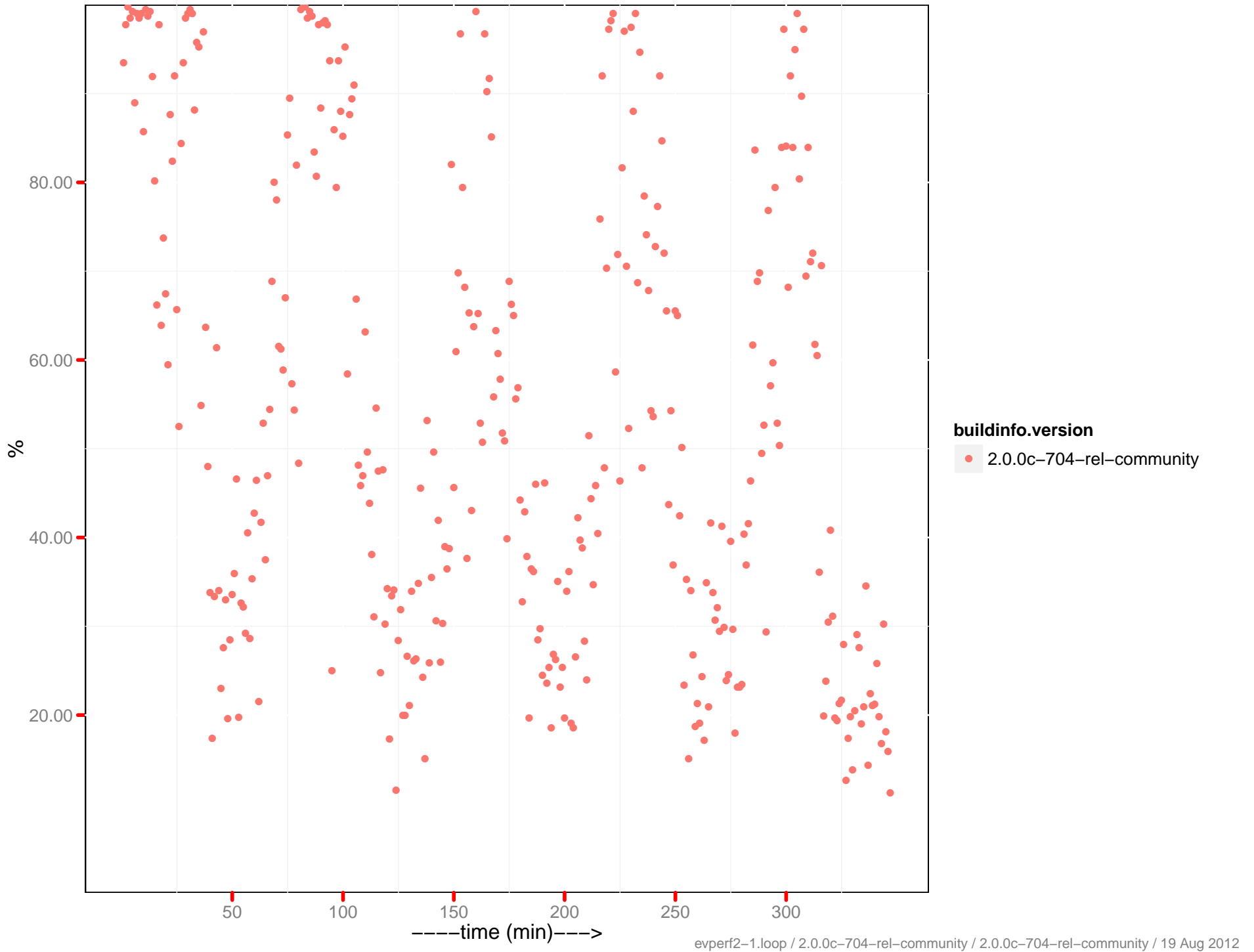
Number of connections



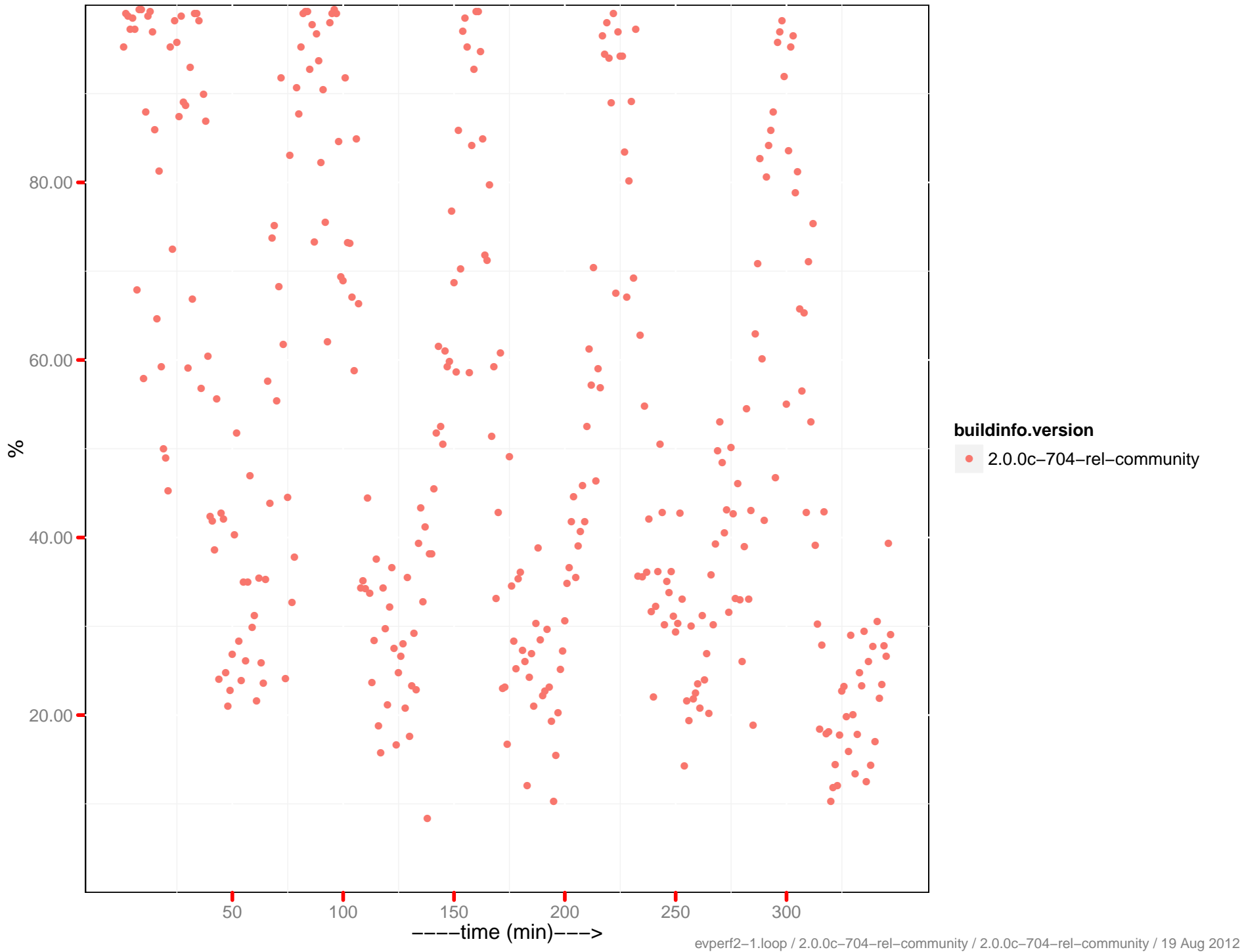
CPU utilization – 10.2.1.58:8091



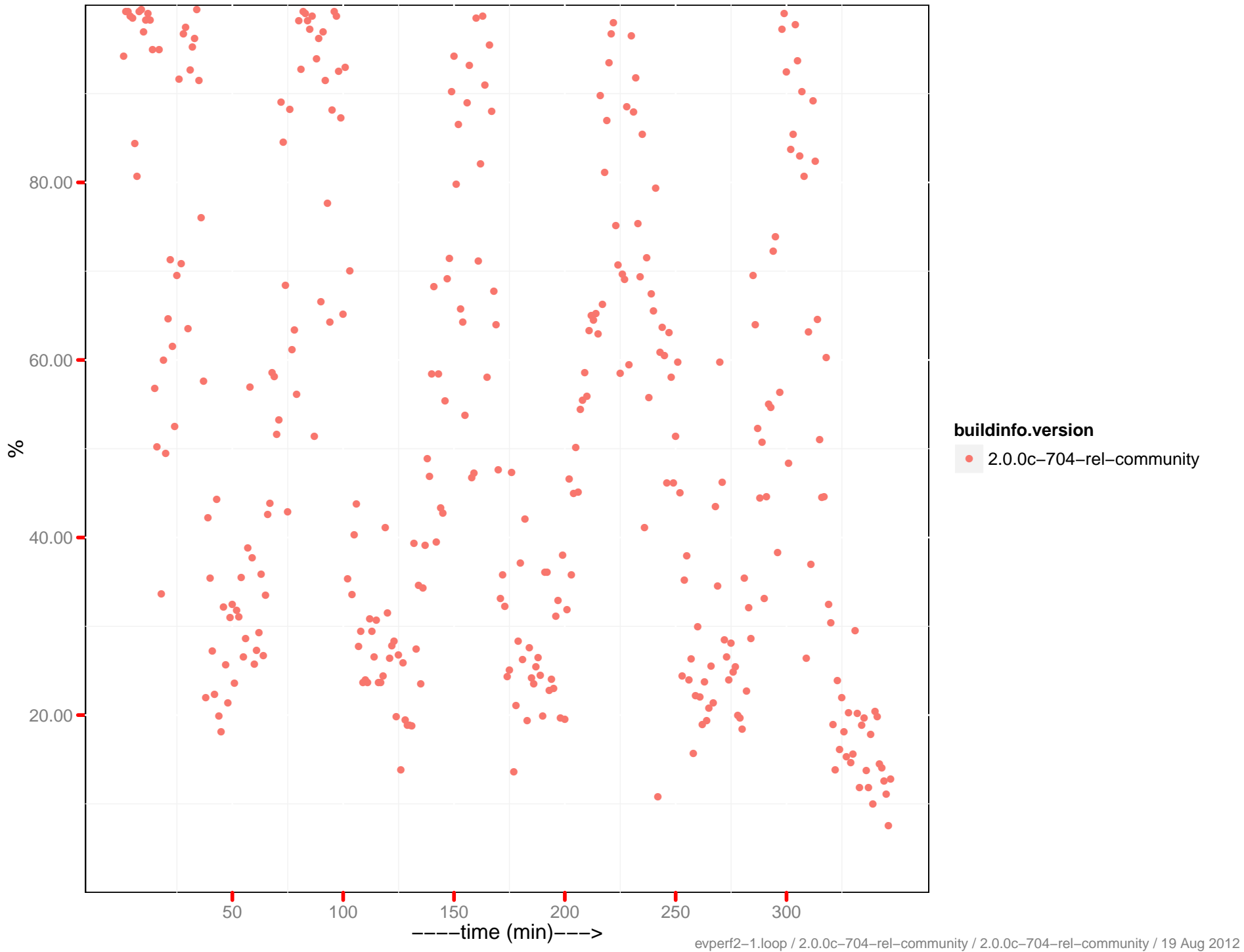
CPU utilization – 10.2.1.61:8091



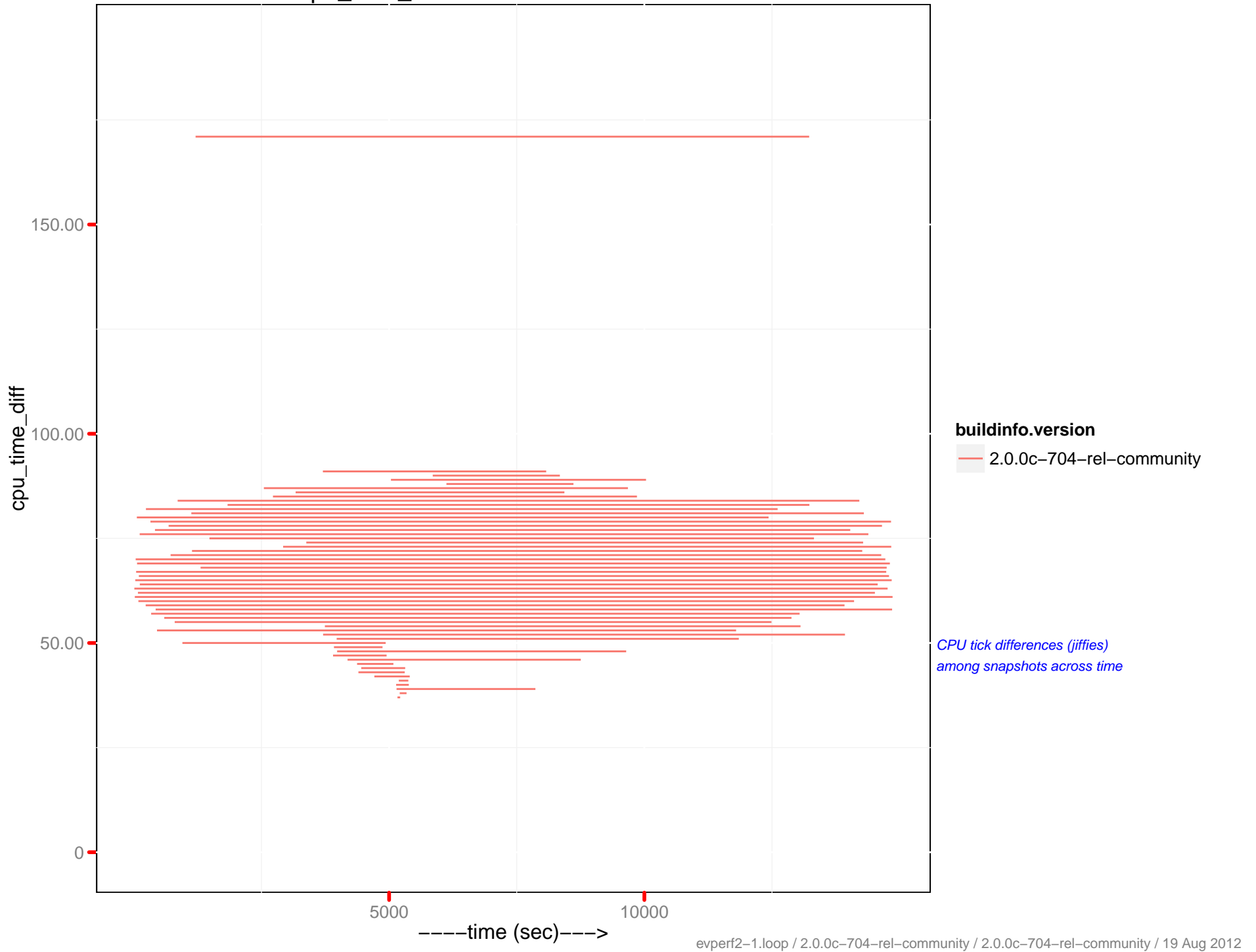
CPU utilization – 10.2.1.63:8091



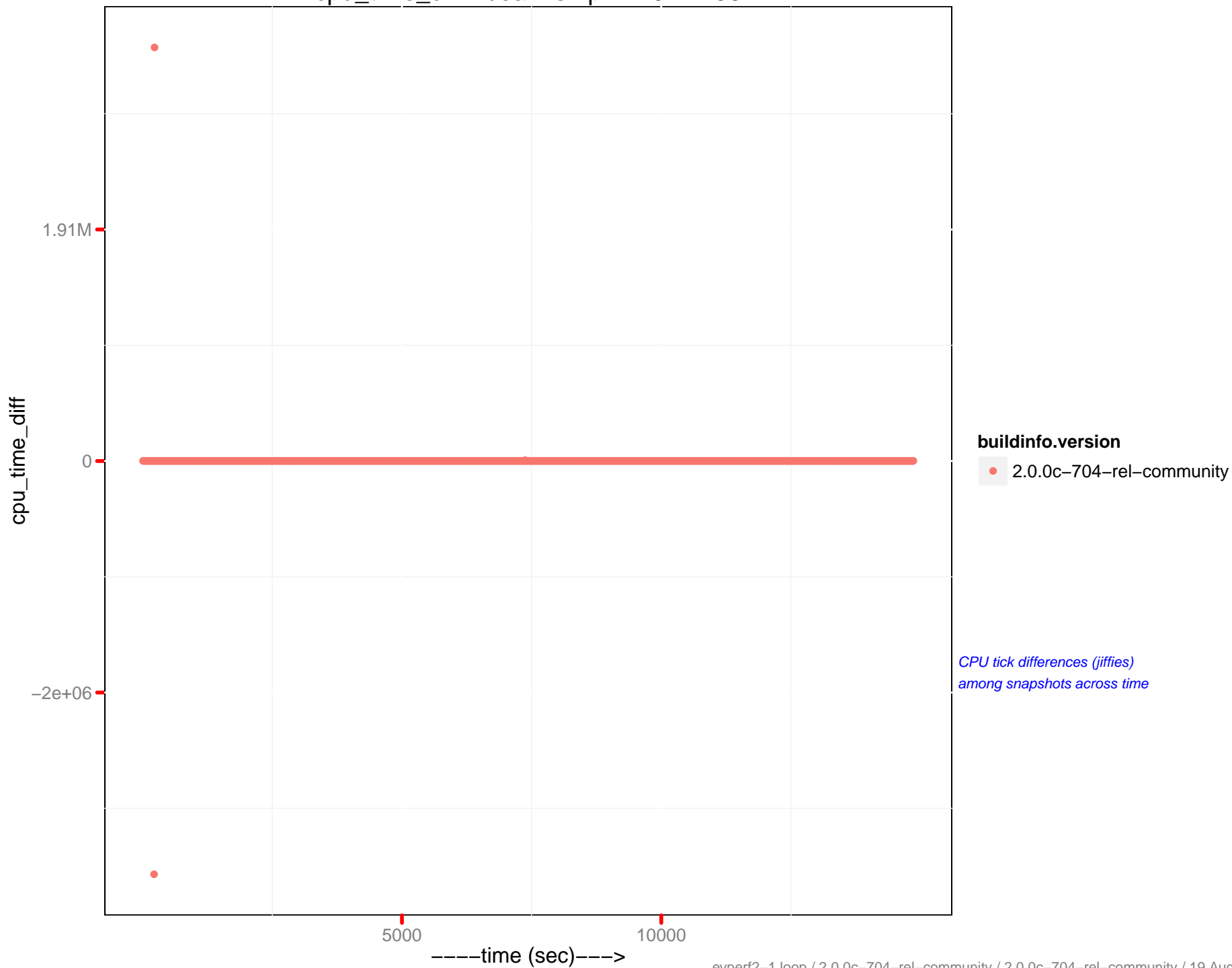
CPU utilization – 10.2.1.64:8091



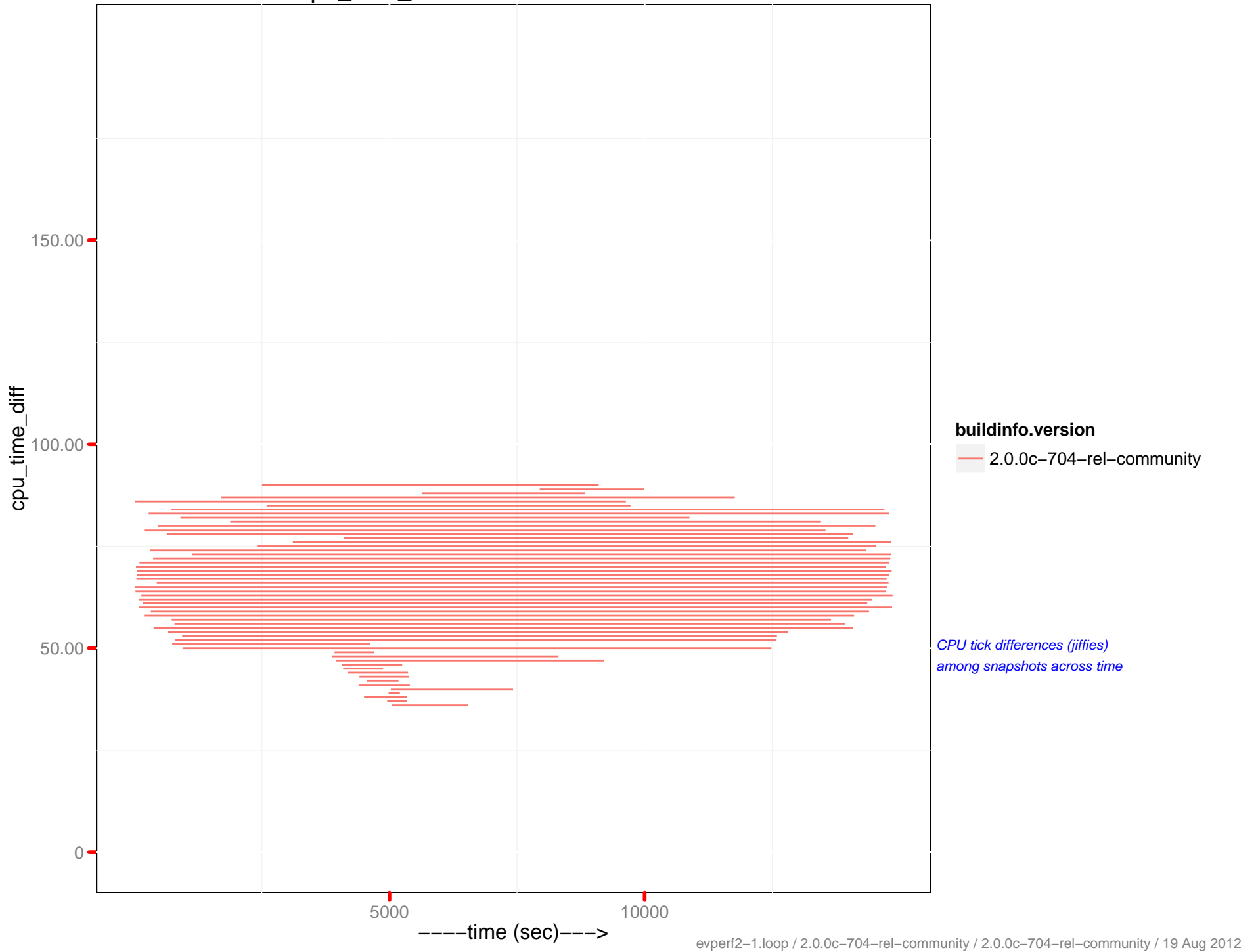
cpu_time_diff: memcached - 10.2.1.58



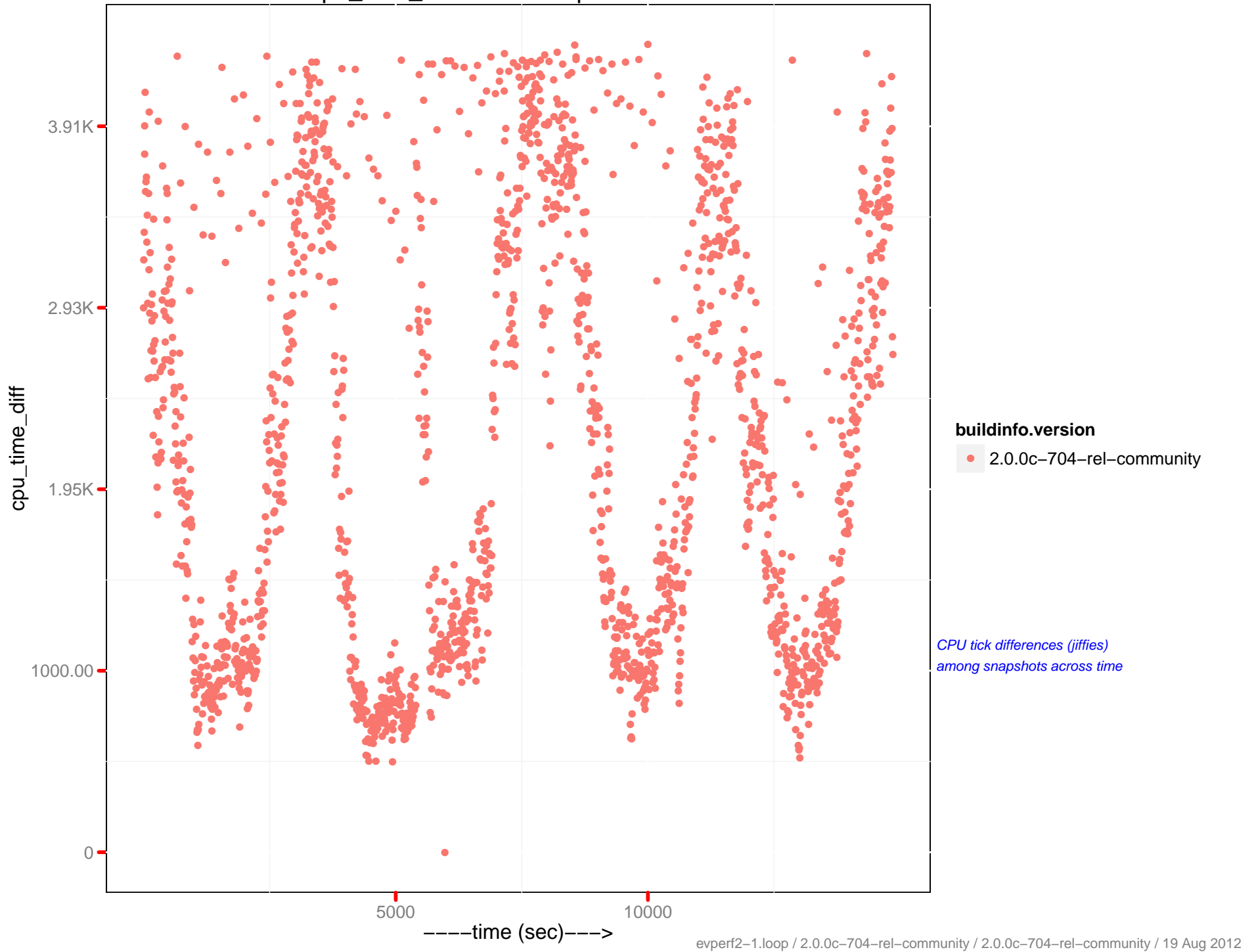
cpu_time_diff : beam.smp - 10.2.1.58



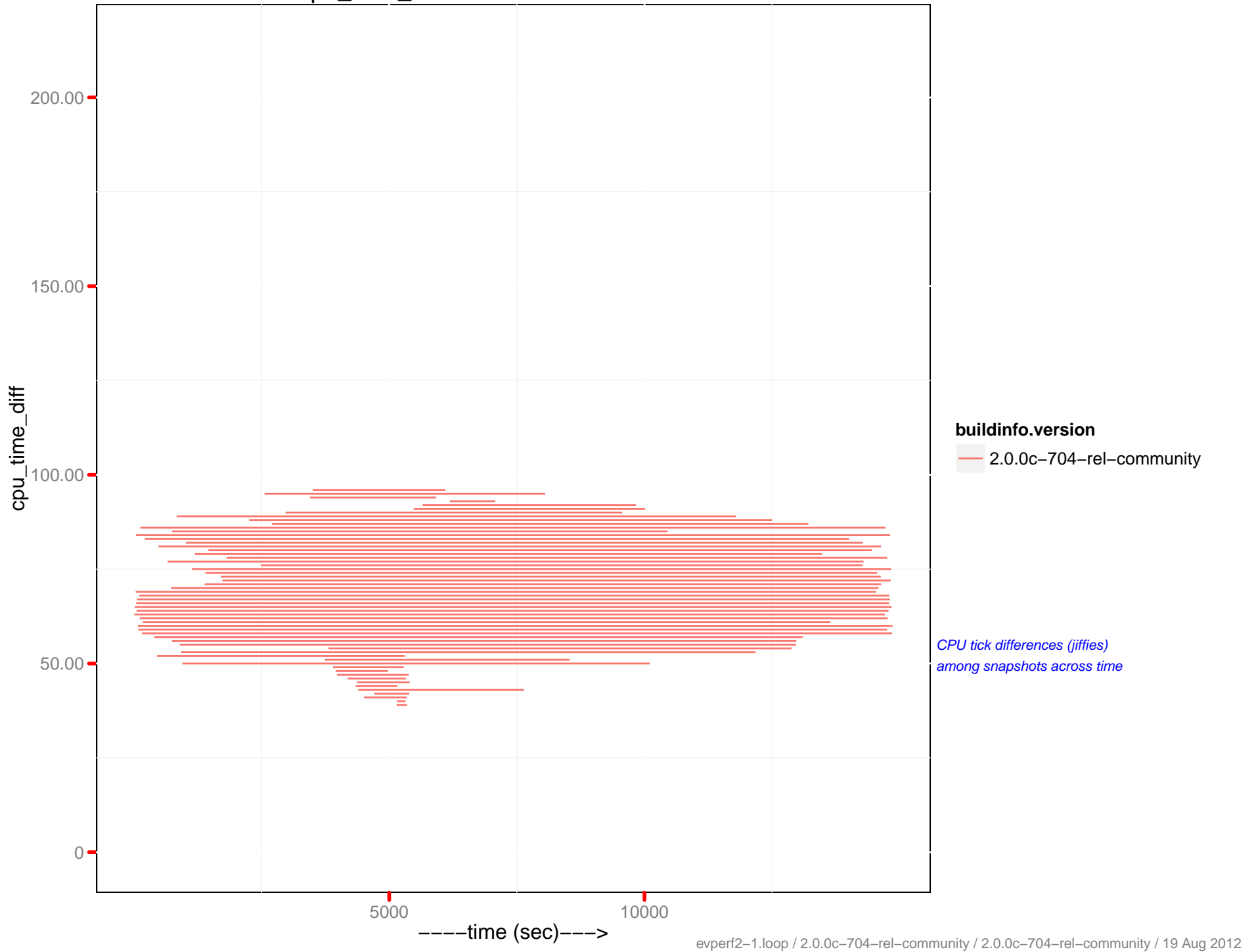
cpu_time_diff: memcached - 10.2.1.61



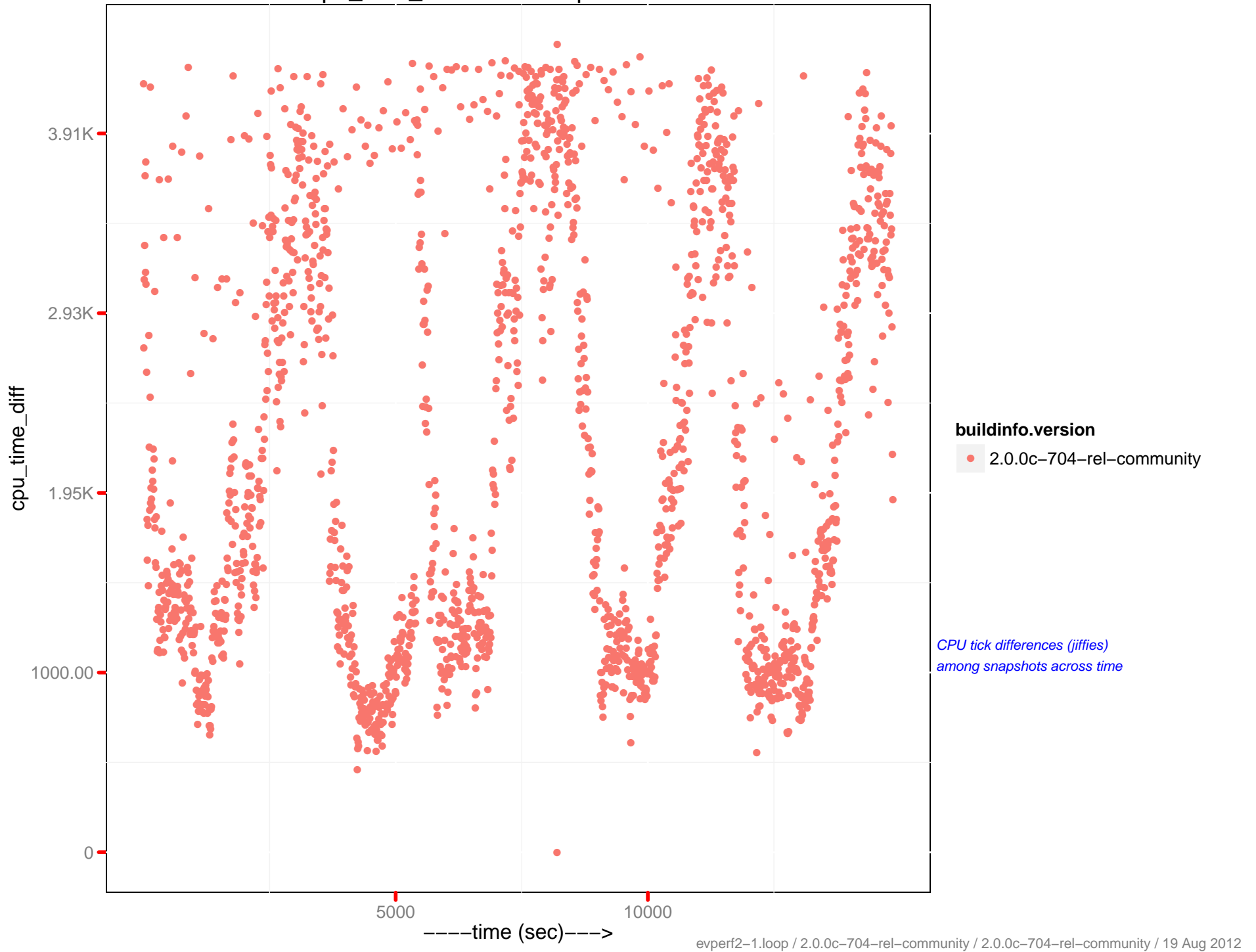
cpu_time_diff : beam.smp - 10.2.1.61



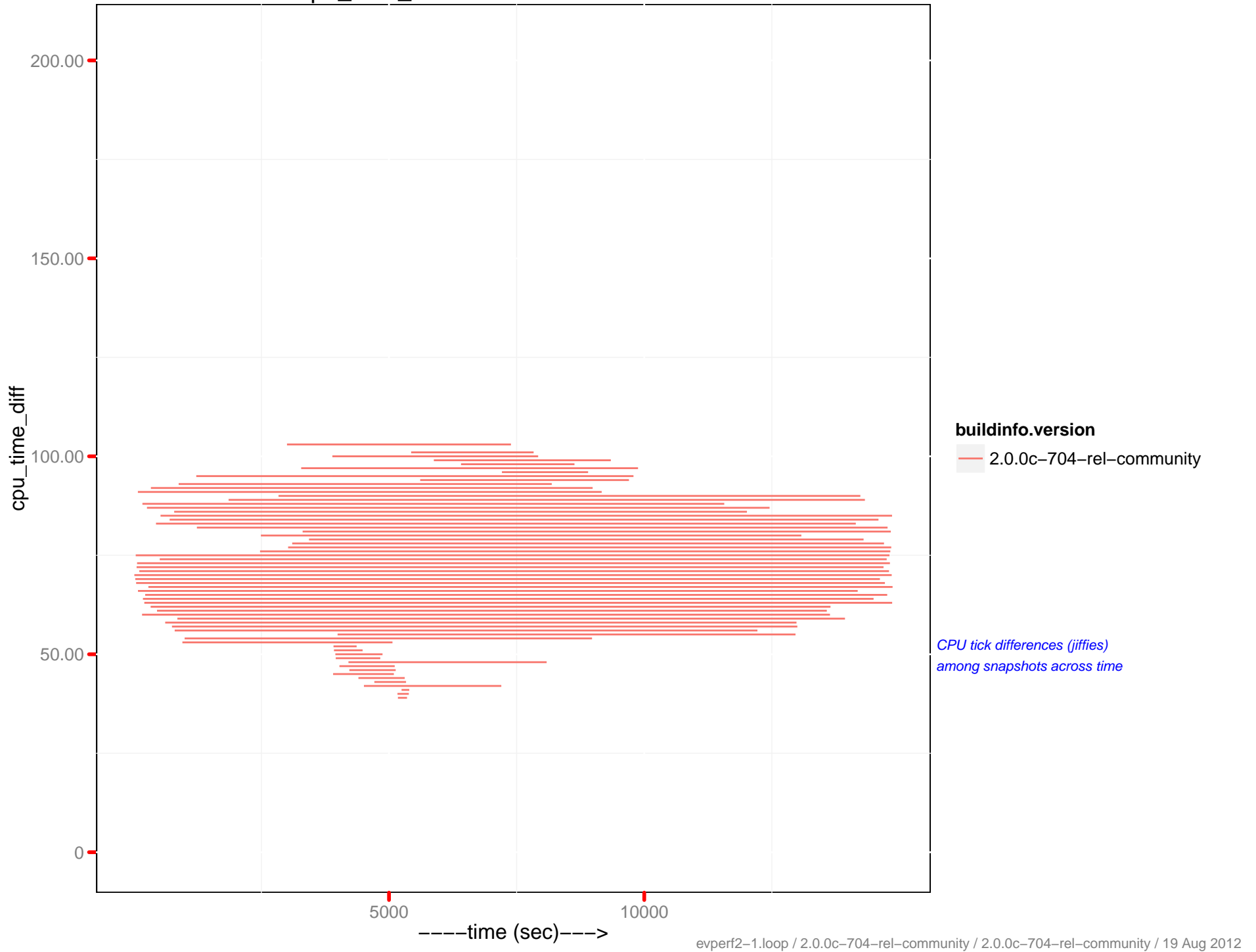
cpu_time_diff: memcached - 10.2.1.63



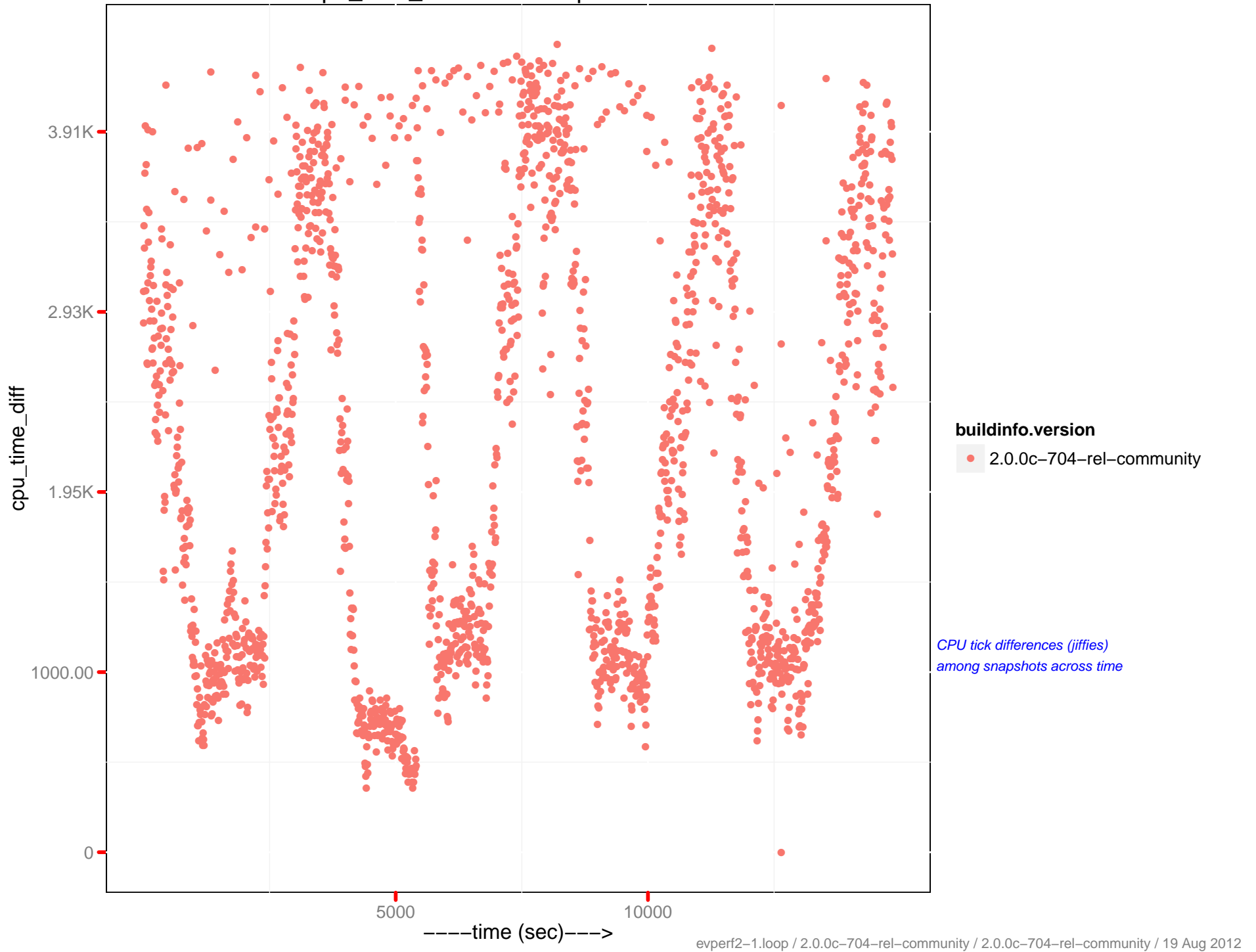
cpu_time_diff : beam.smp - 10.2.1.63



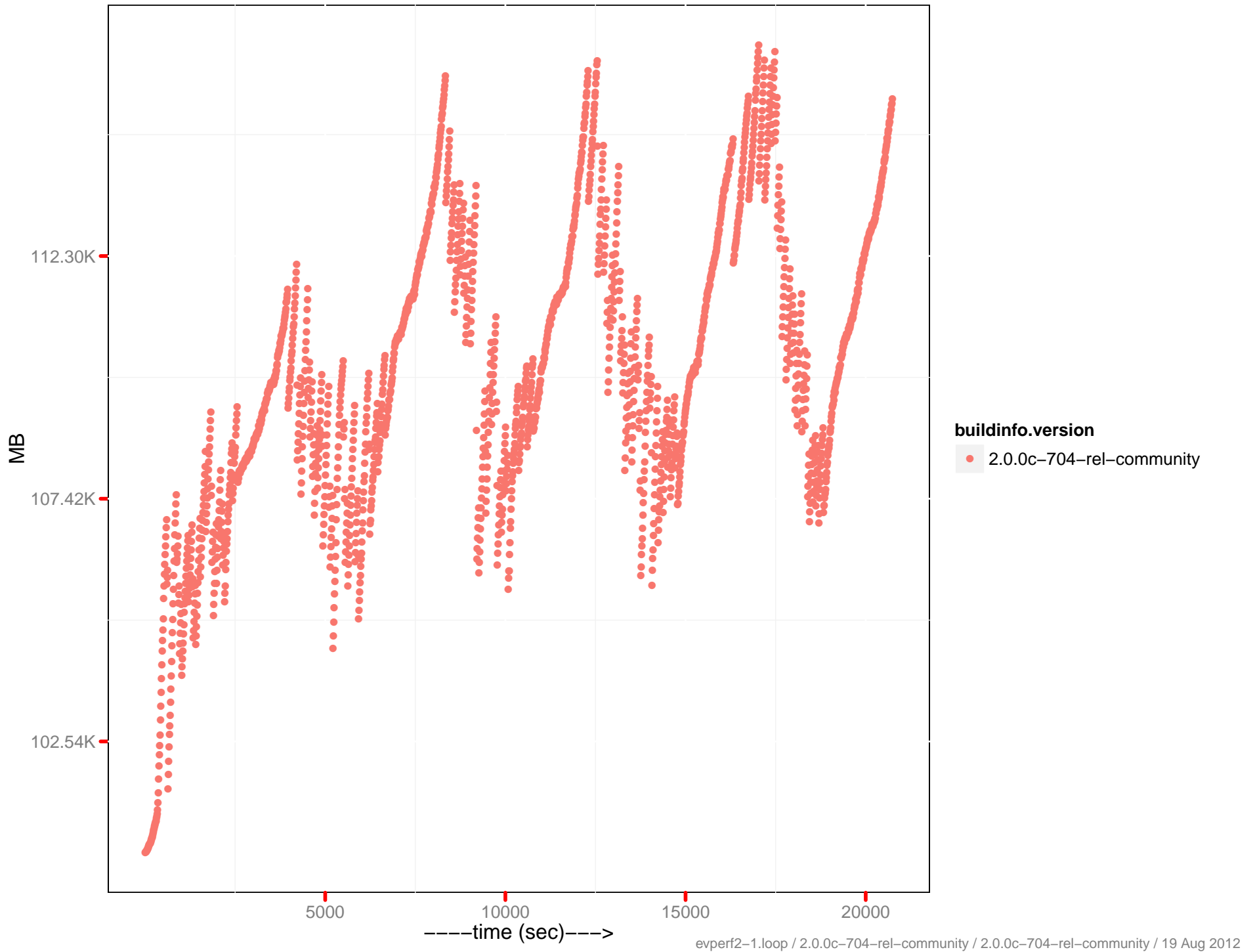
cpu_time_diff: memcached - 10.2.1.64



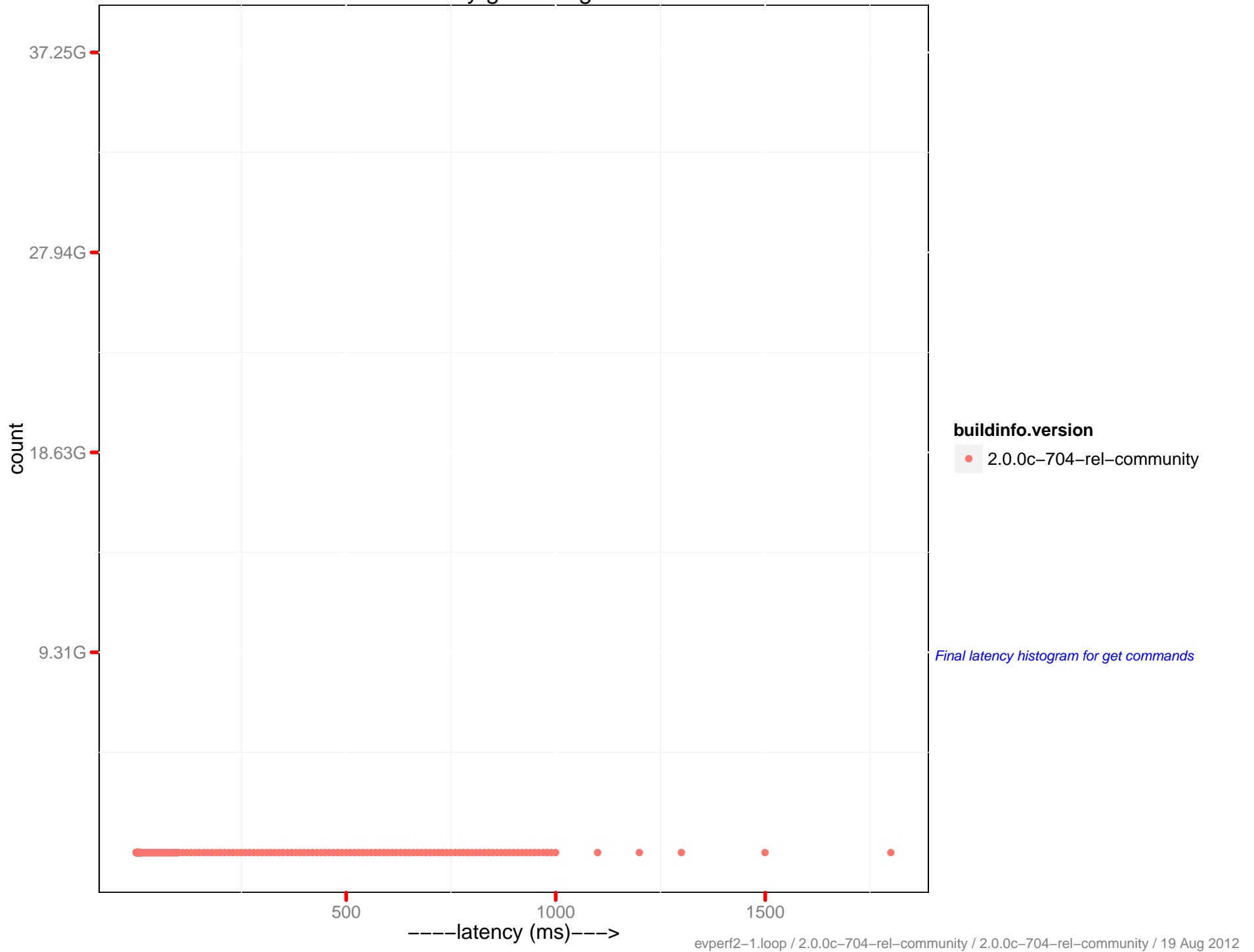
cpu_time_diff : beam.smp - 10.2.1.64



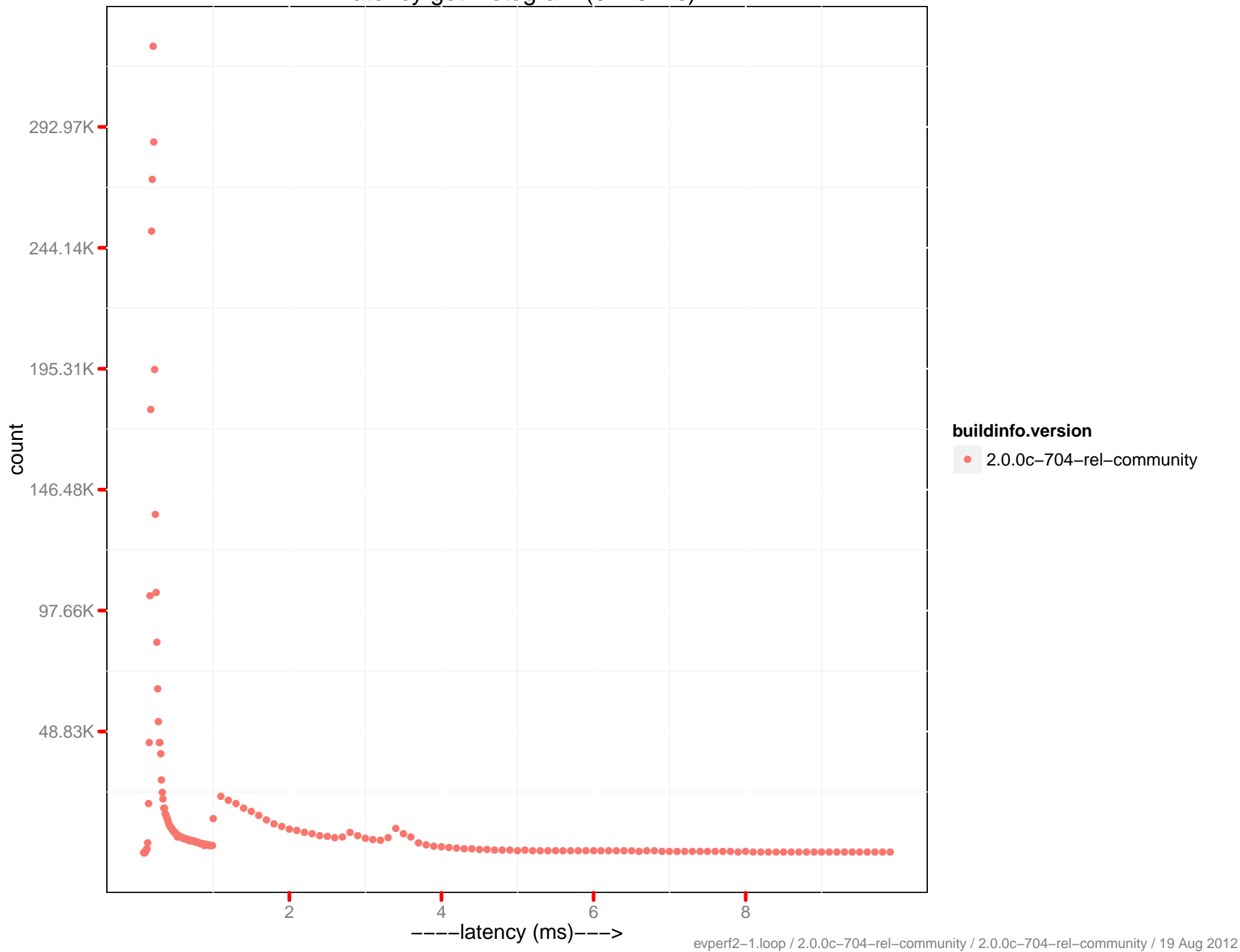
Data disk size



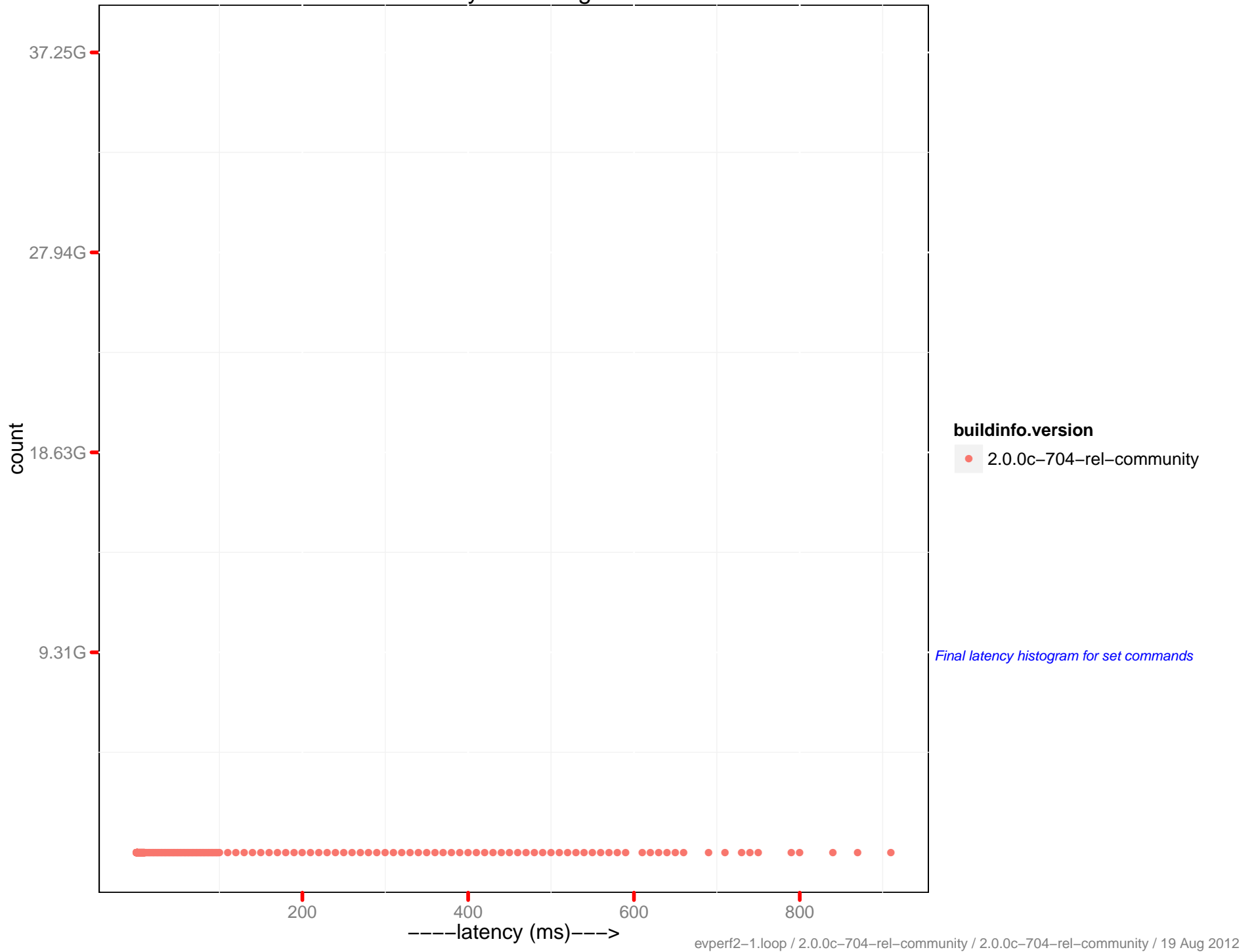
Latency get histogram



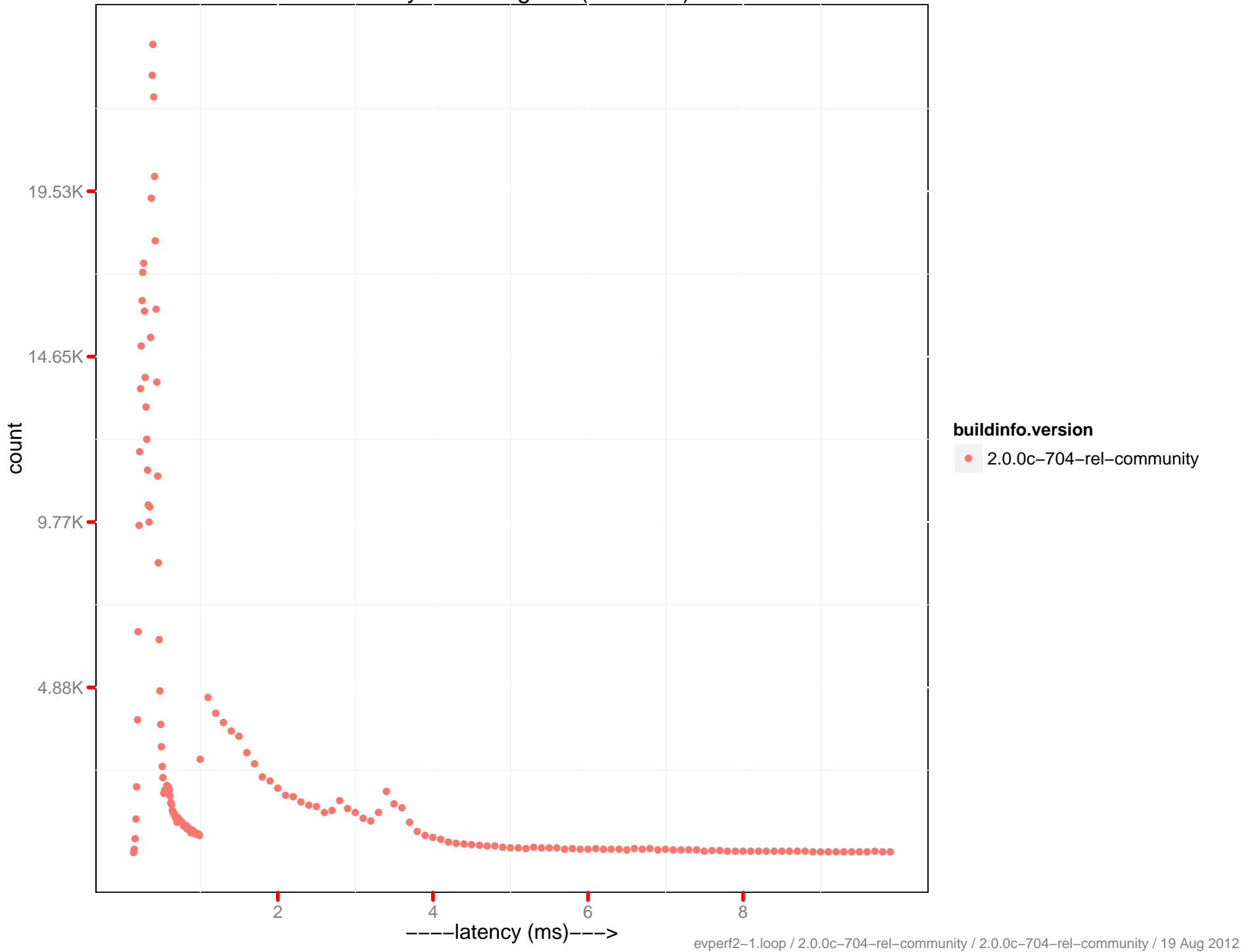
Latency get histogram (0–10 ms)



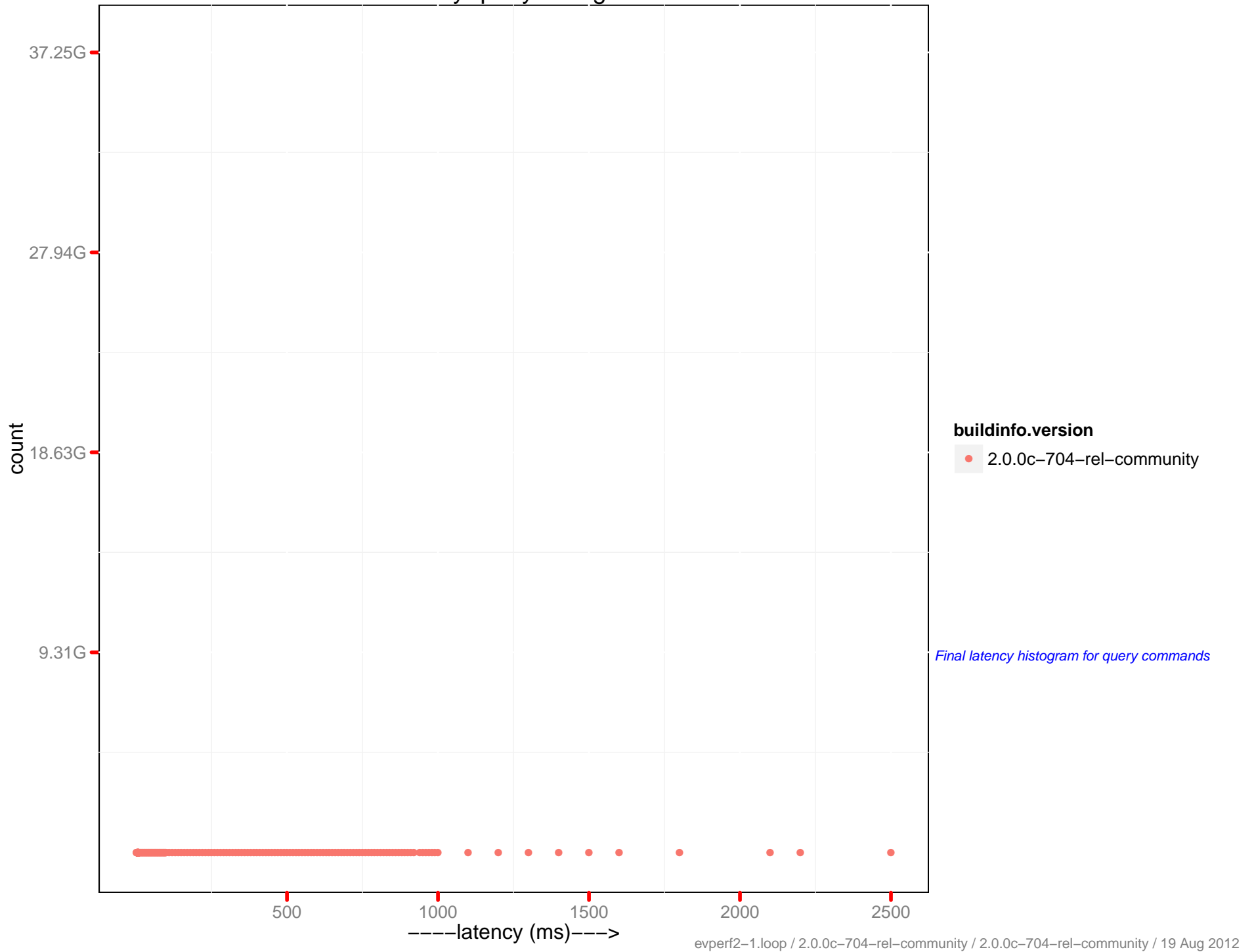
Latency set histogram



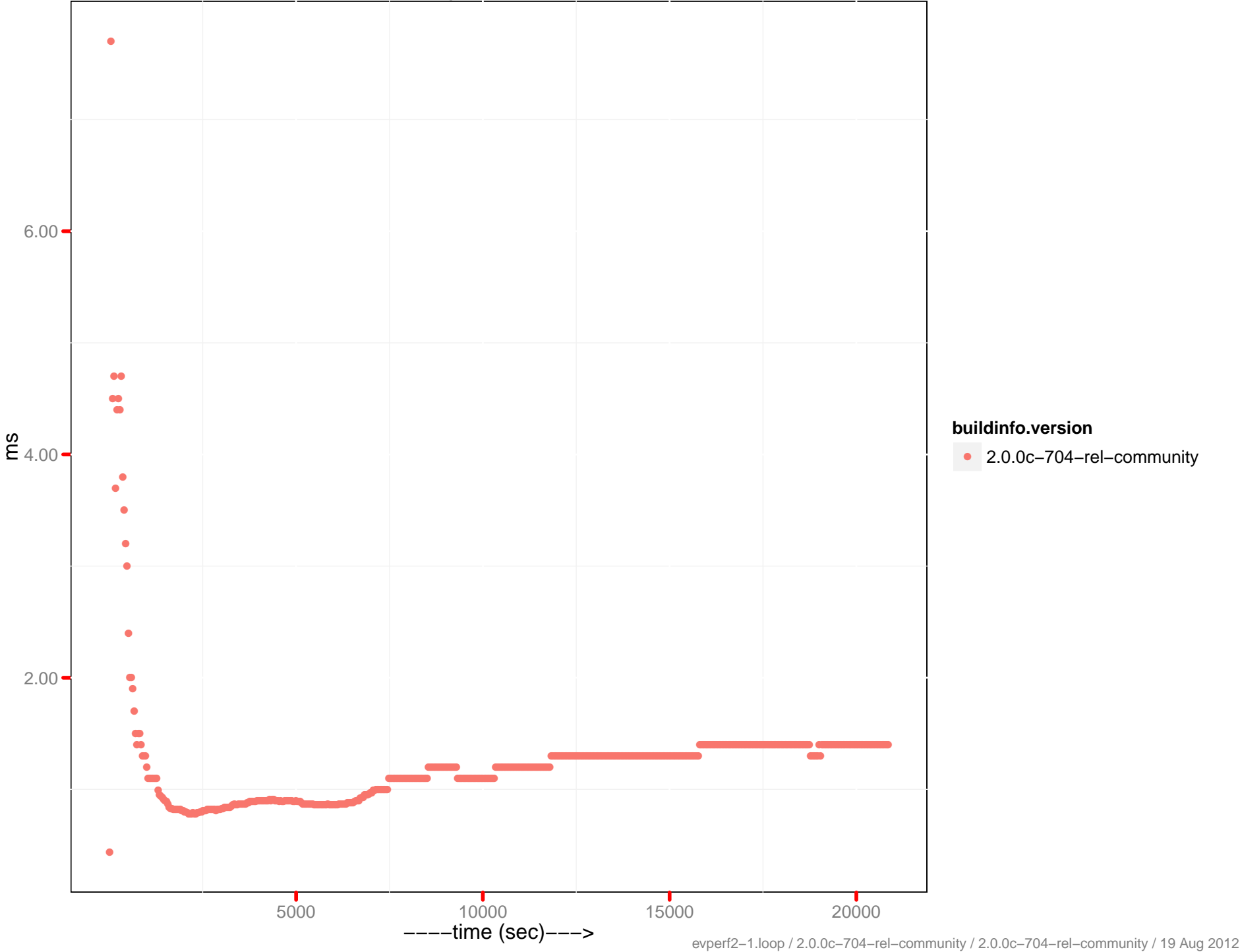
Latency set histogram (0–10 ms)



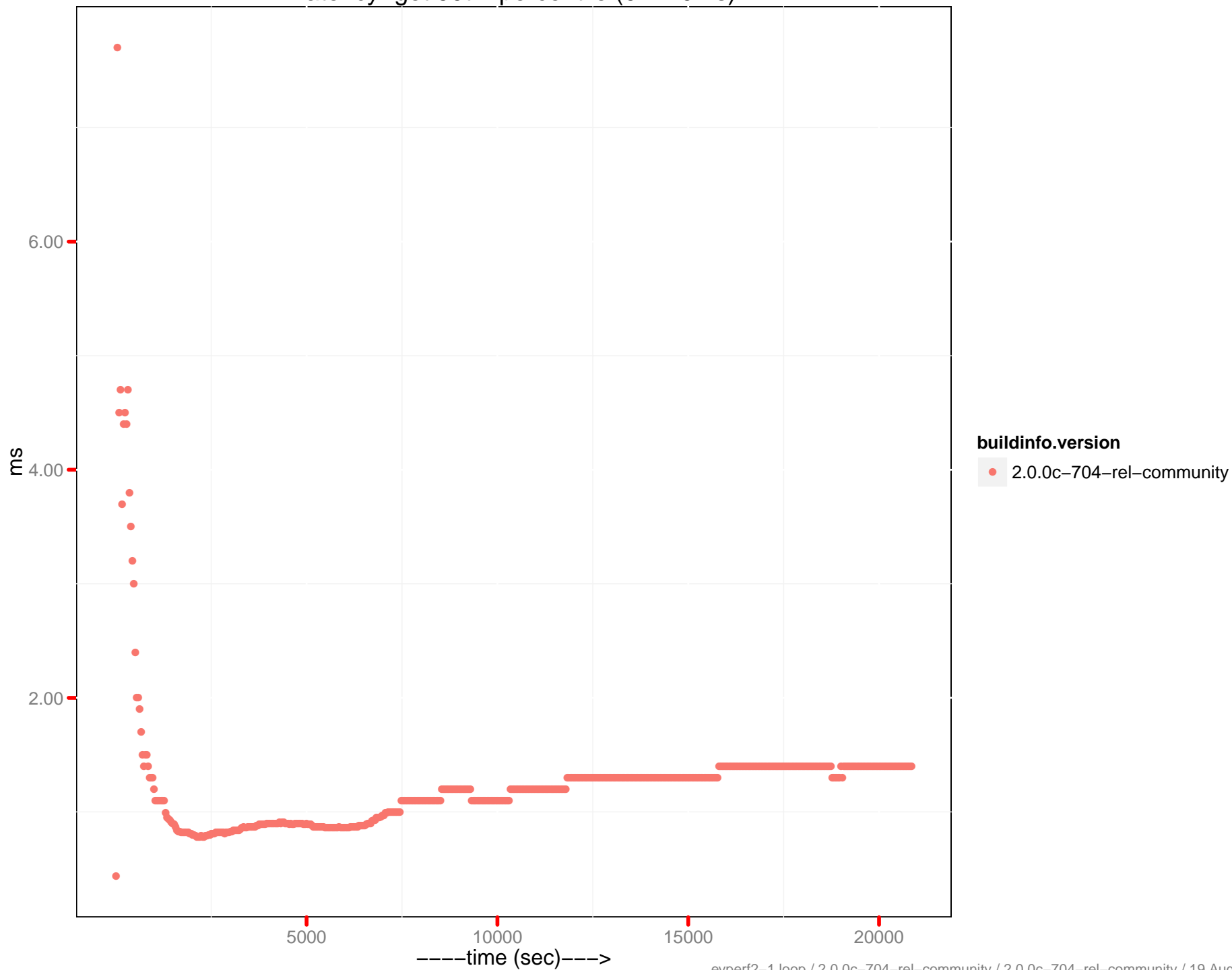
Latency query histogram



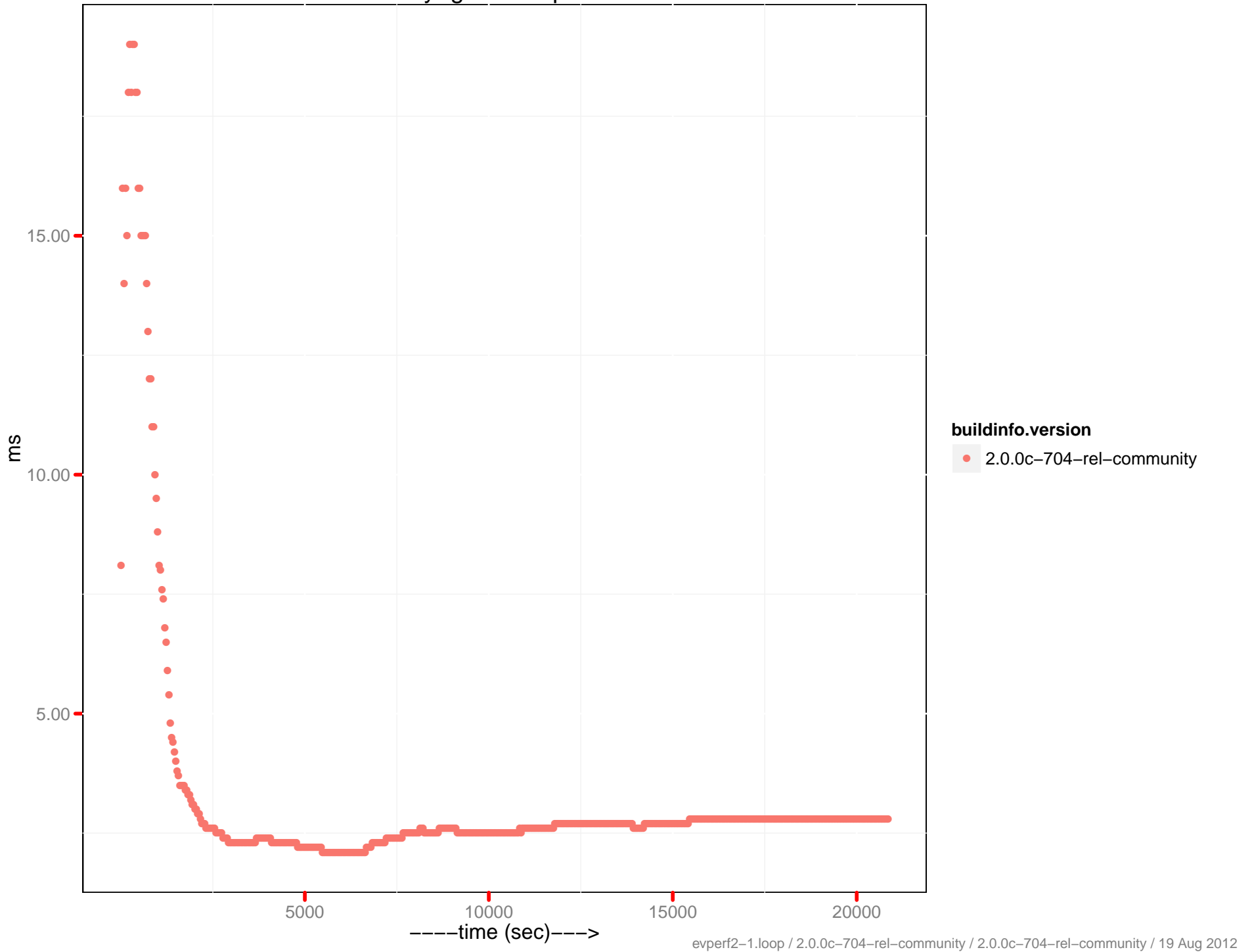
Latency-get 90th percentile



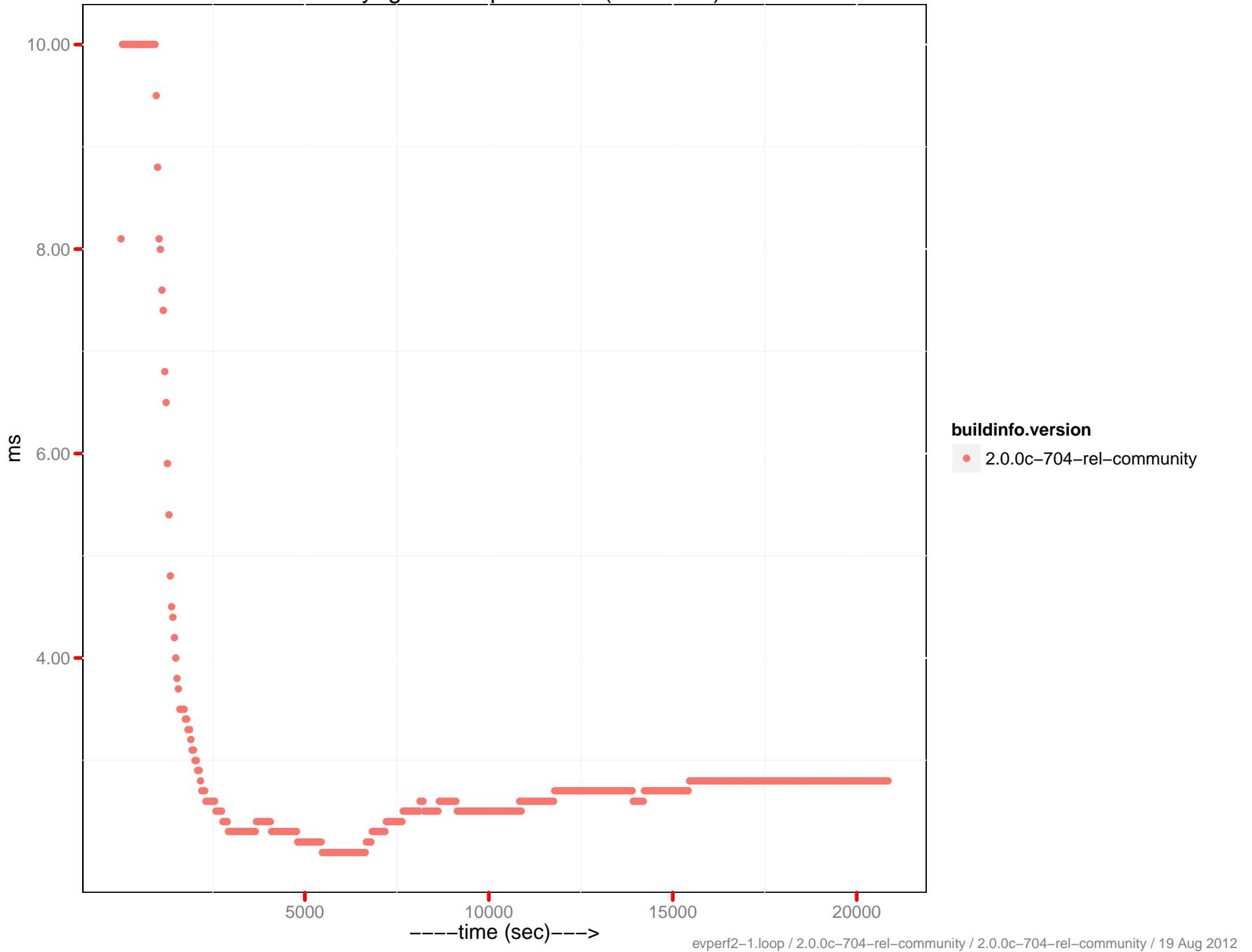
Latency-get 90th percentile (0 - 10ms)



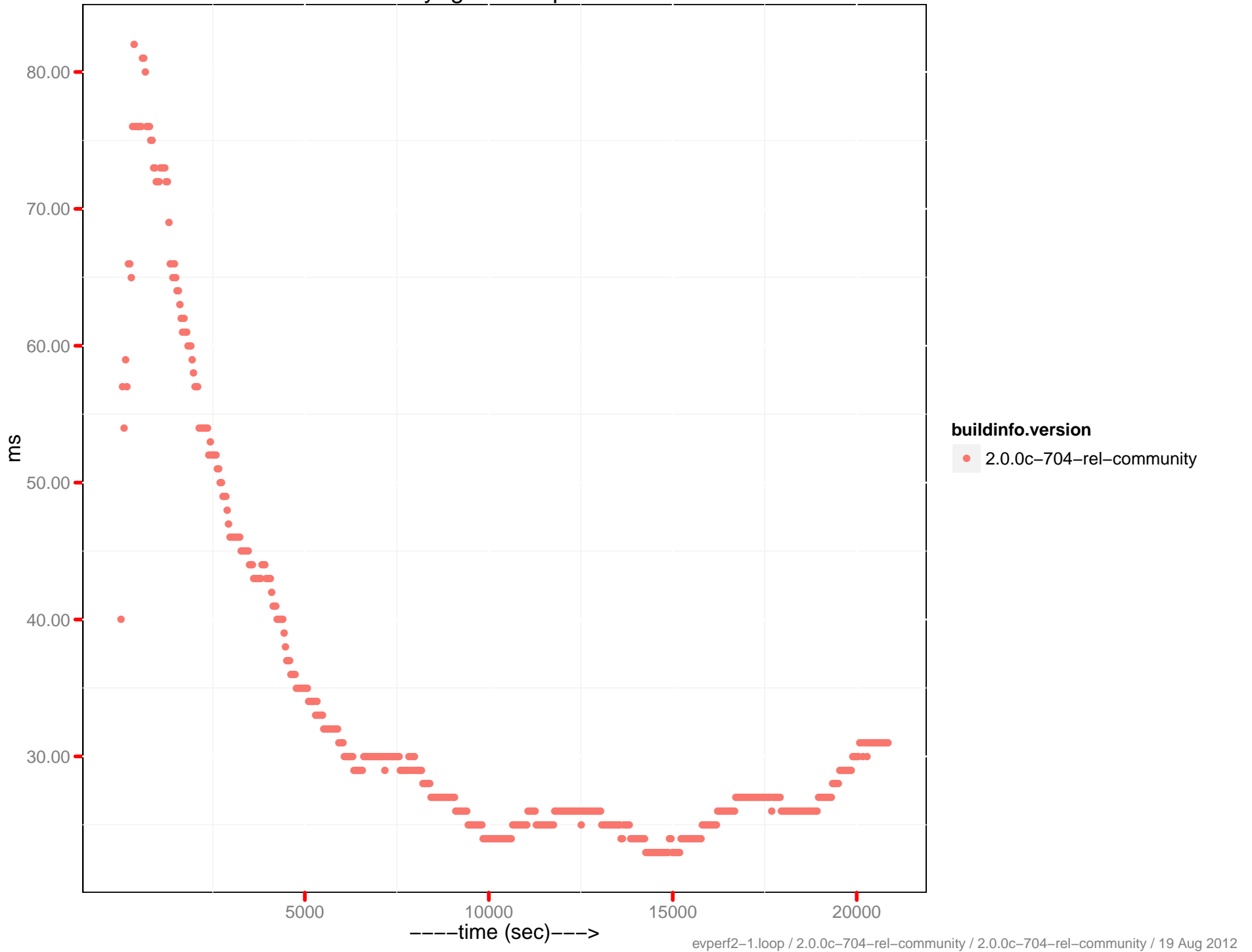
Latency-get 95th percentile



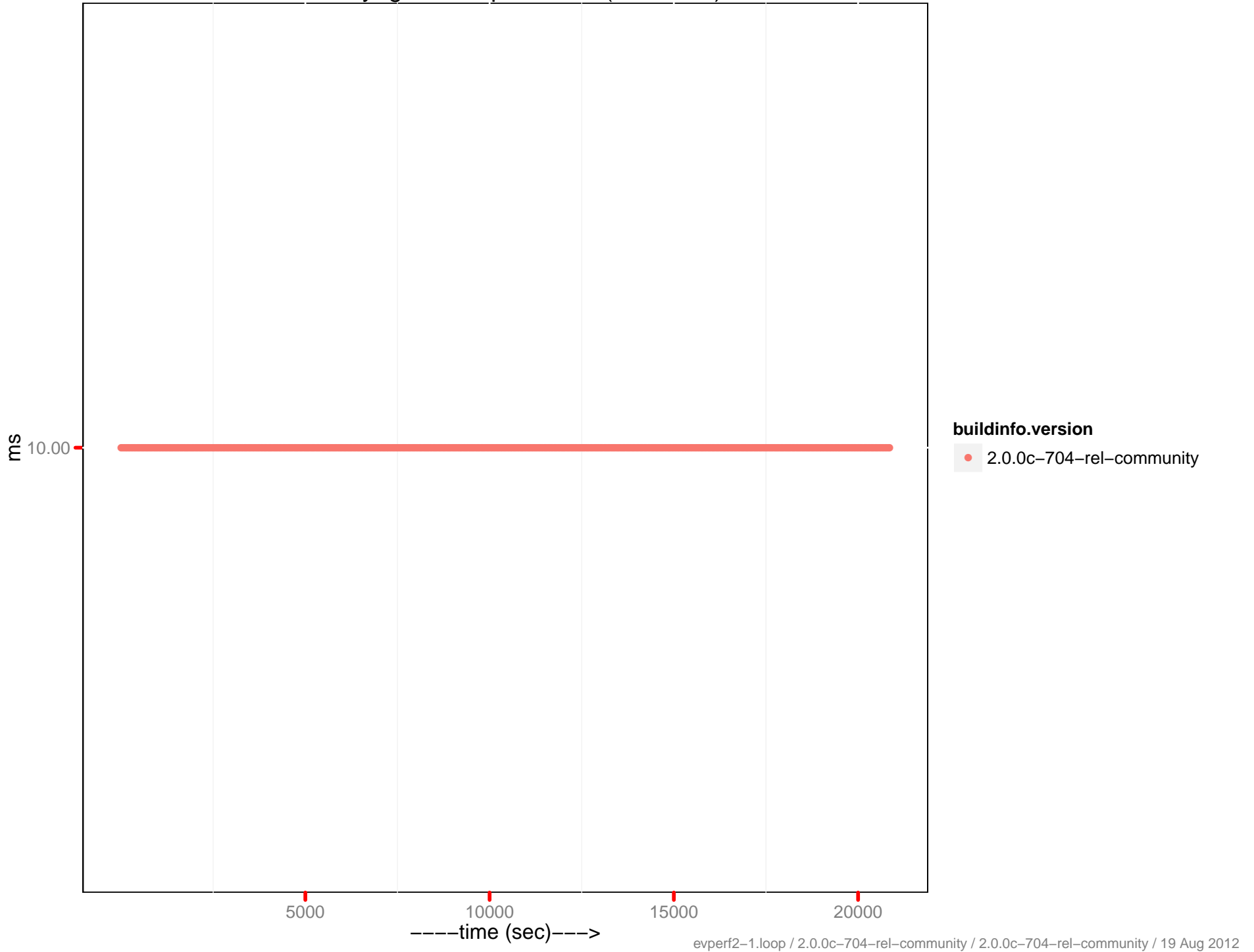
Latency-get 95th percentile (0 - 10ms)



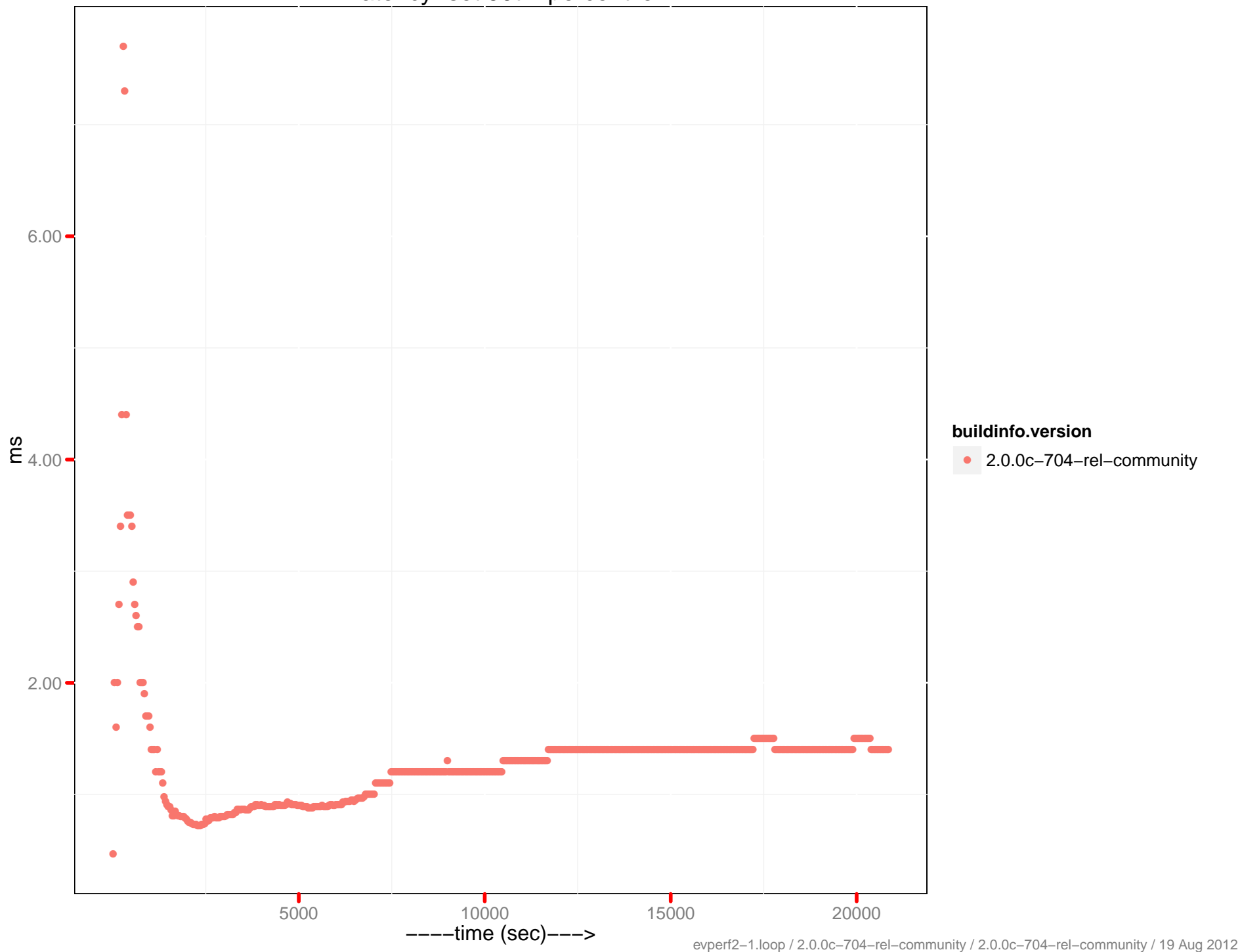
Latency-get 99th percentile



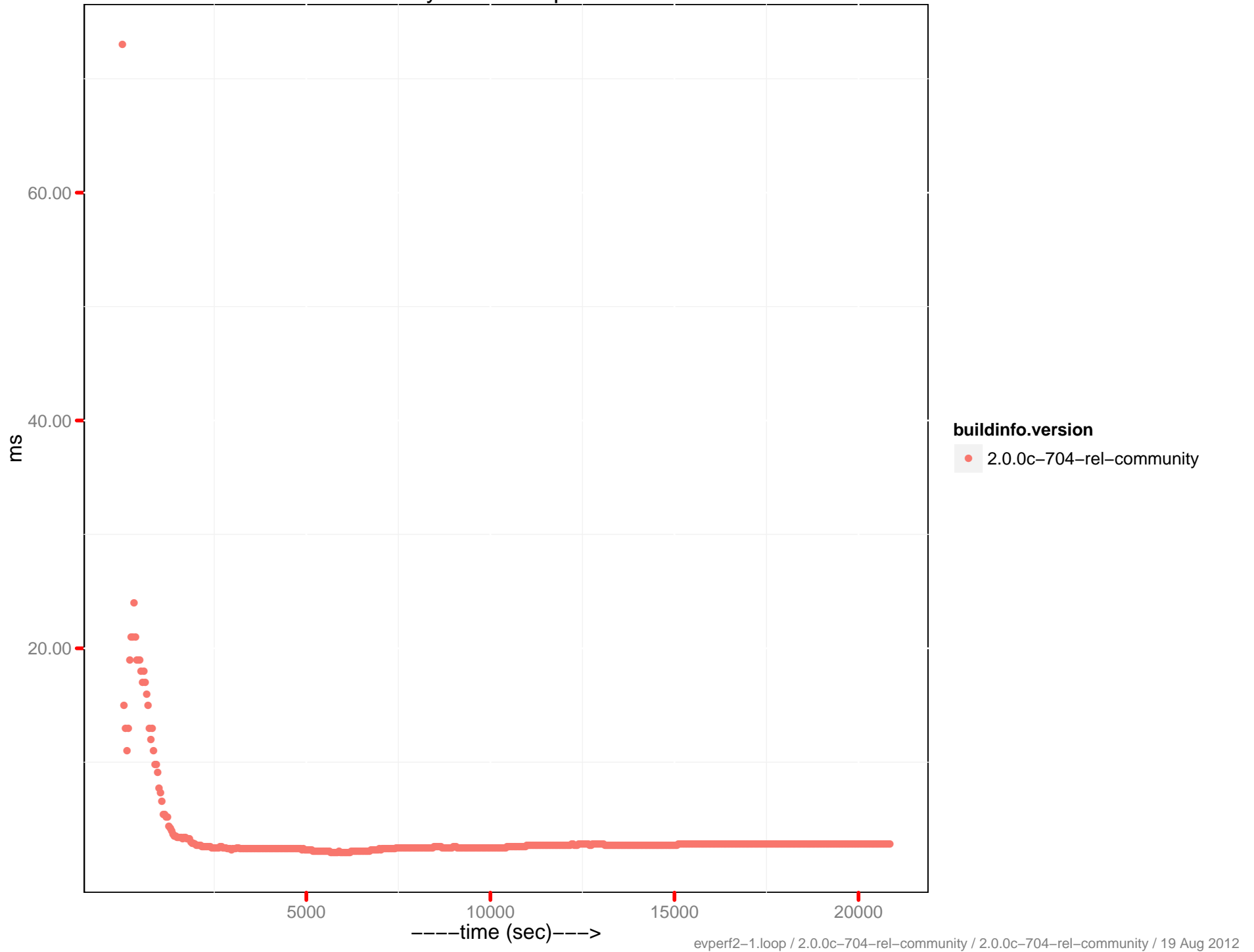
Latency-get 99th percentile (0 - 10ms)



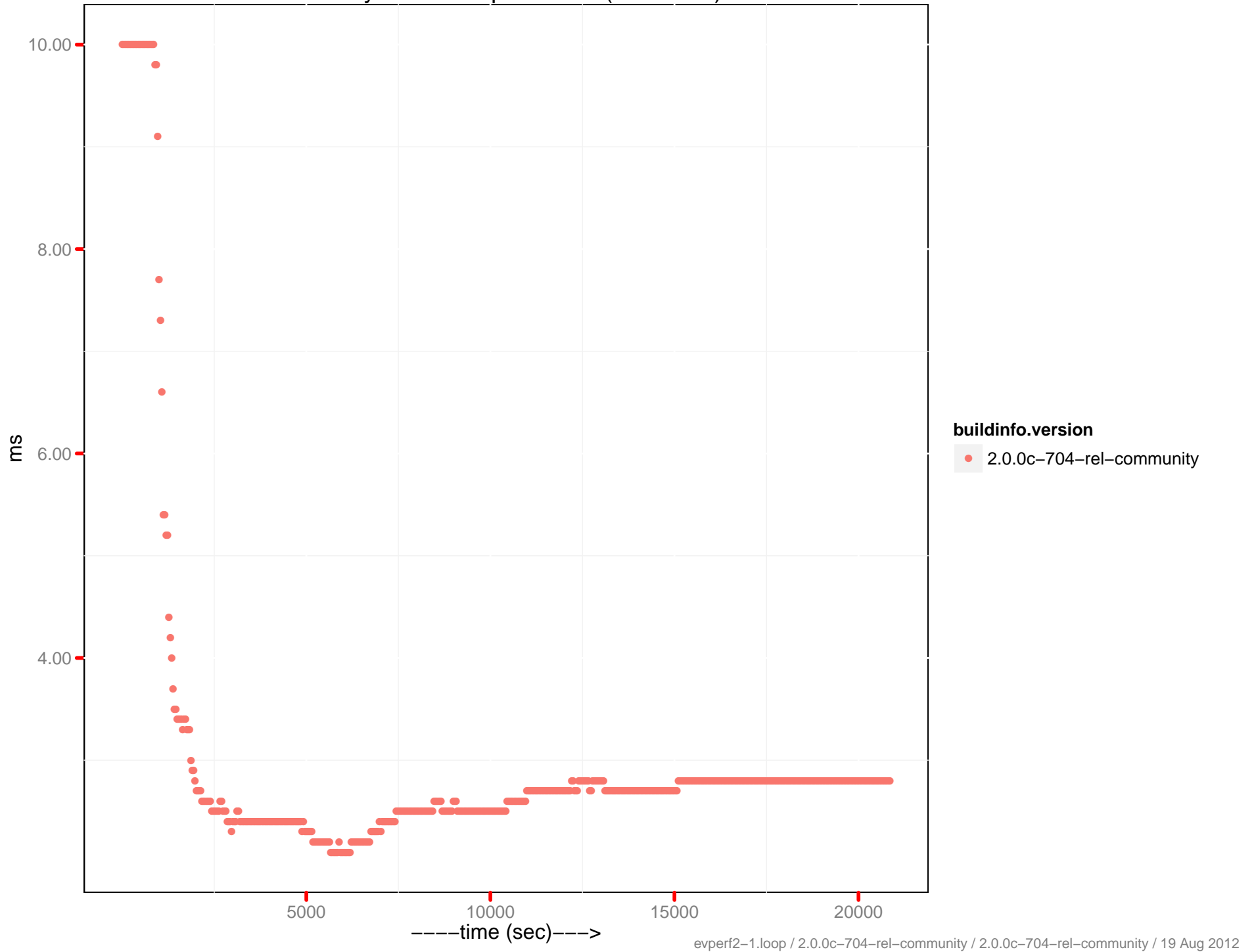
Latency-set 90th percentile



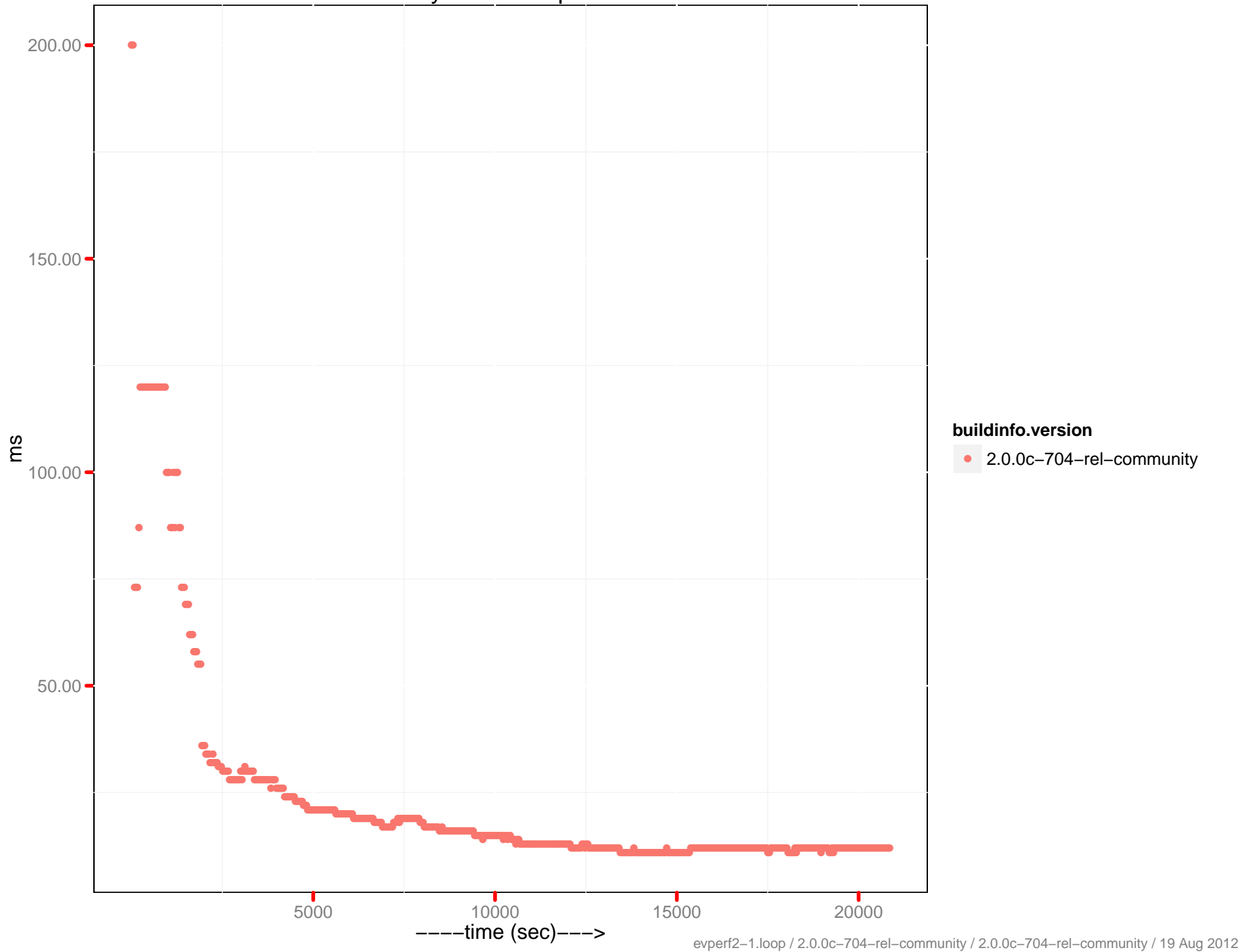
Latency-set 95th percentile



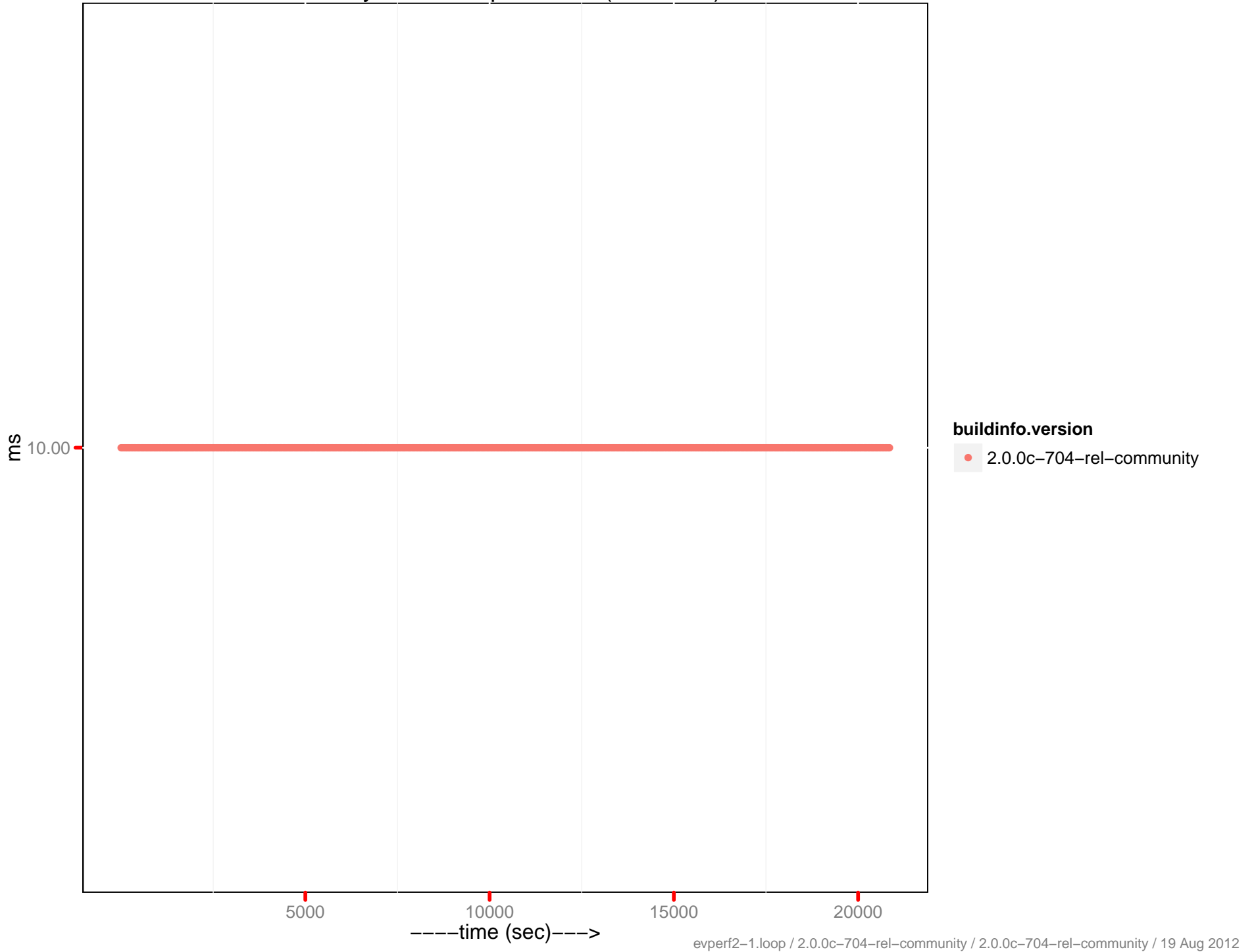
Latency-set 95th percentile (0 - 10ms)



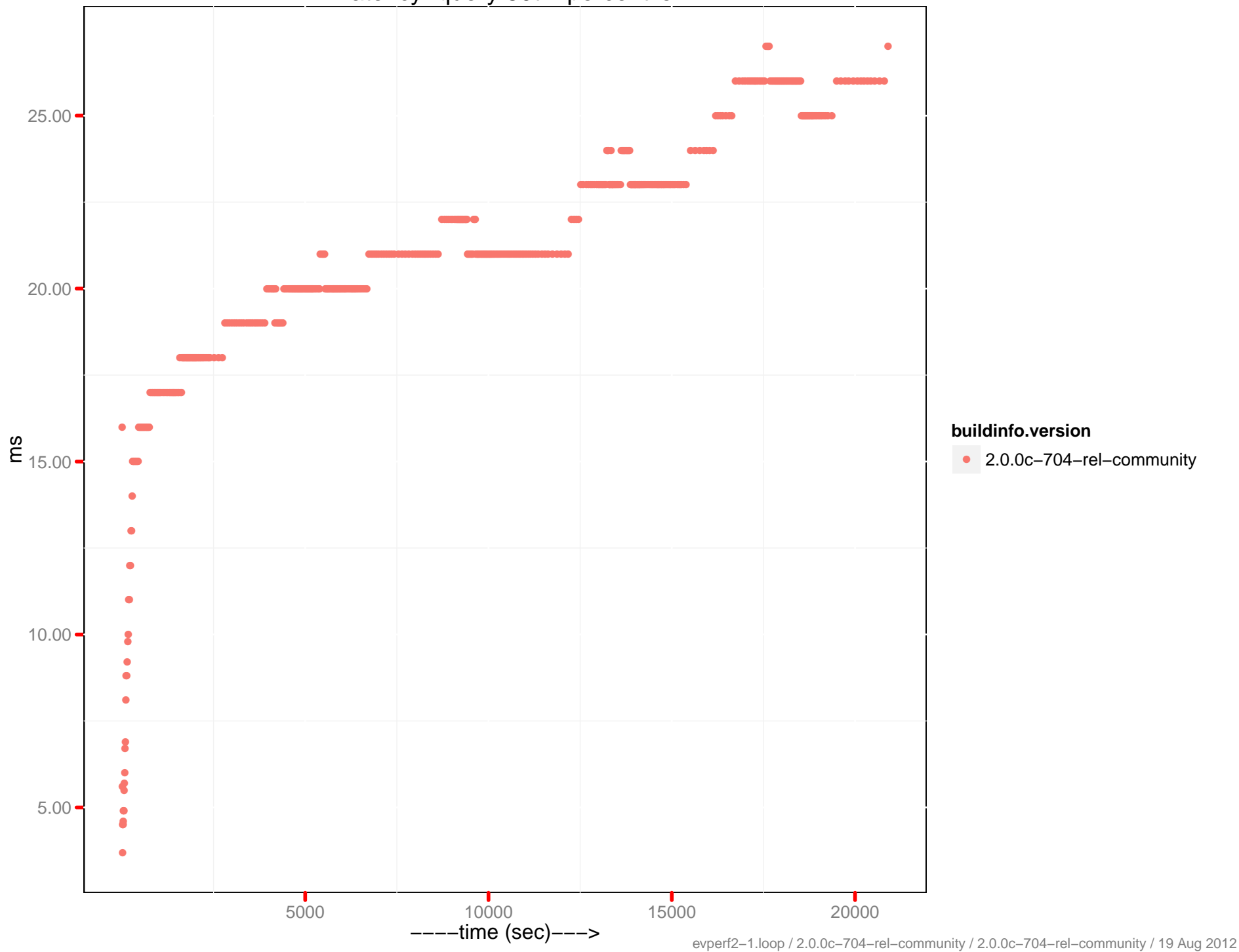
Latency-set 99th percentile



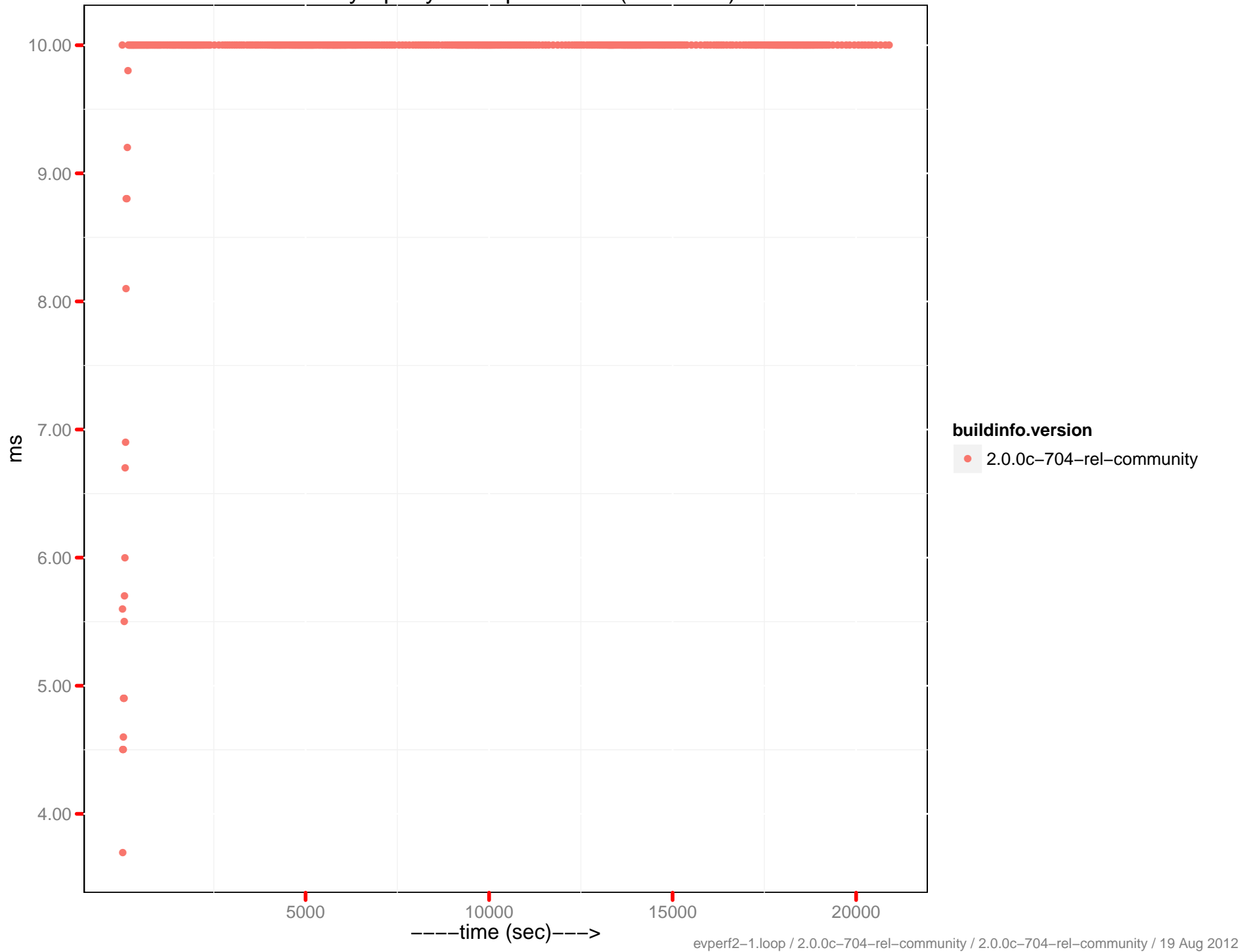
Latency-set 99th percentile (0 - 10ms)



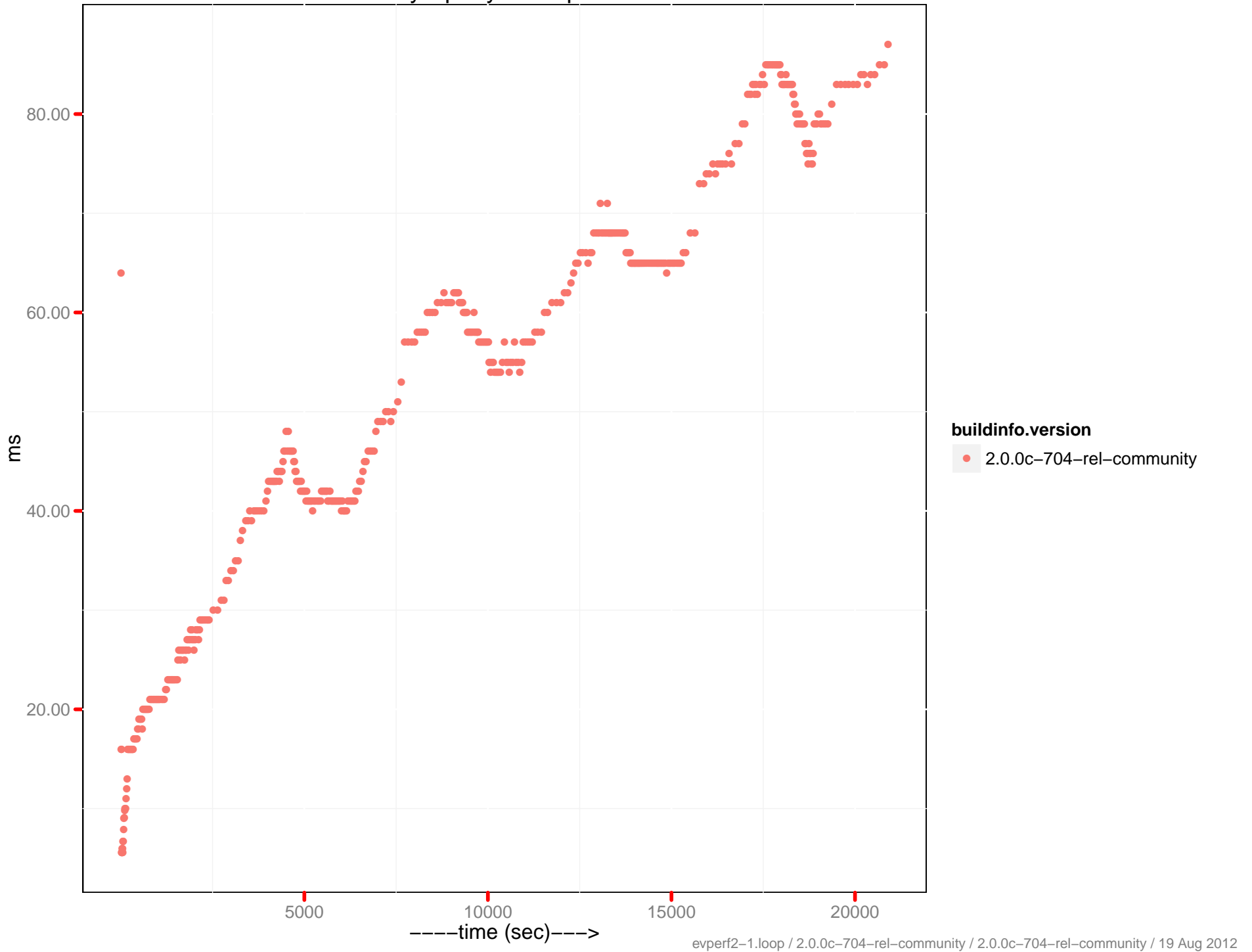
Latency-query 80th percentile



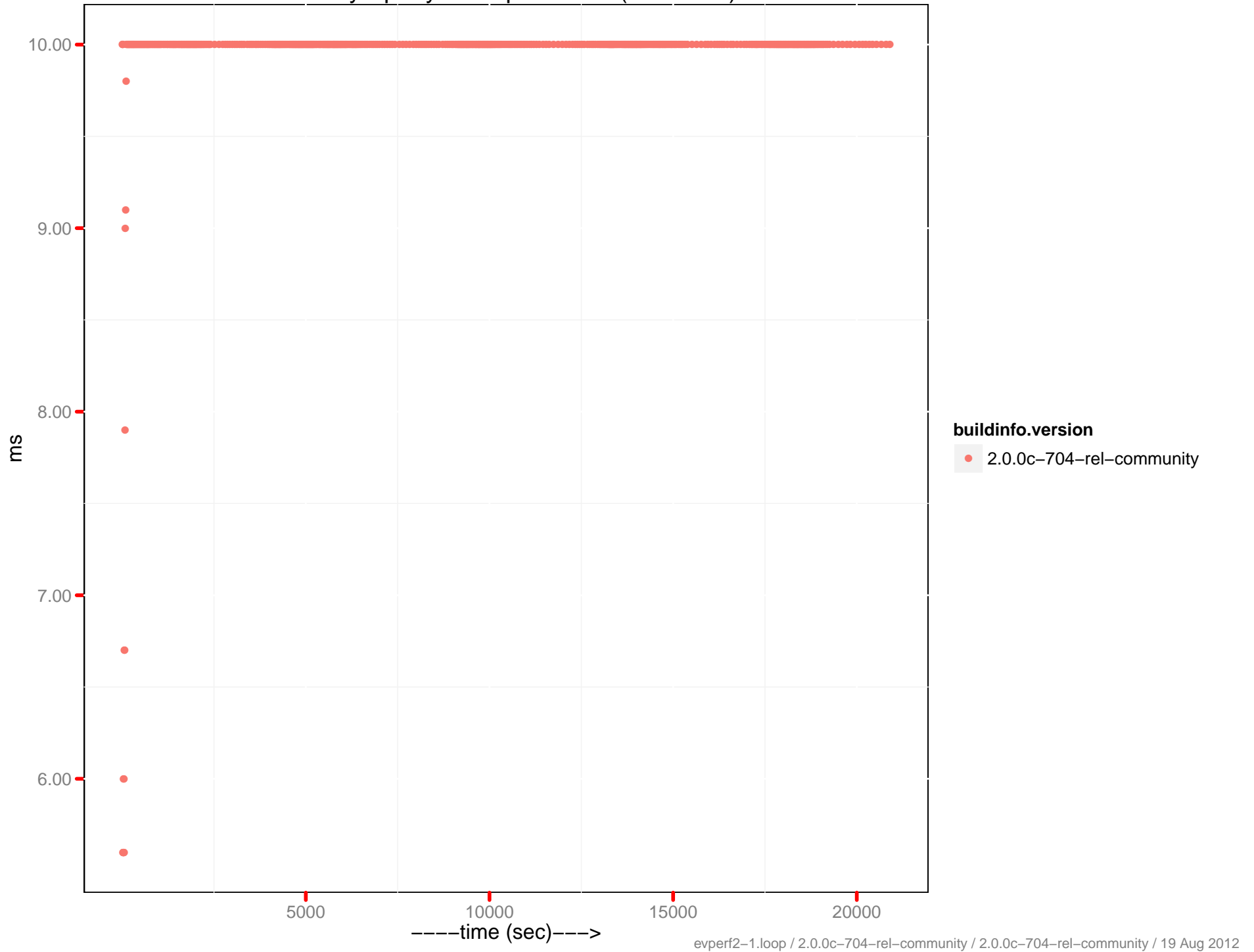
Latency-query 80th percentile (0 – 10ms)



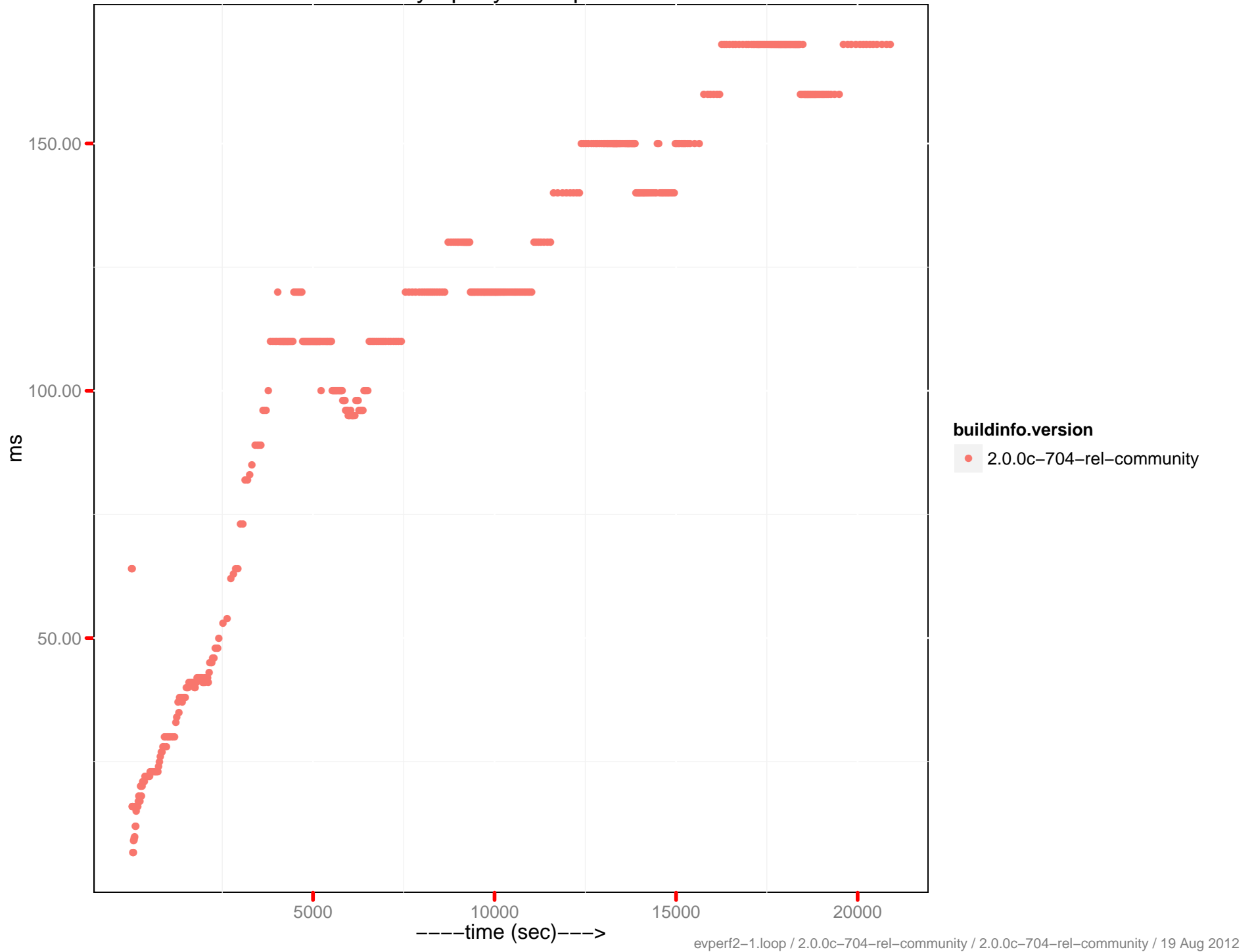
Latency-query 90th percentile



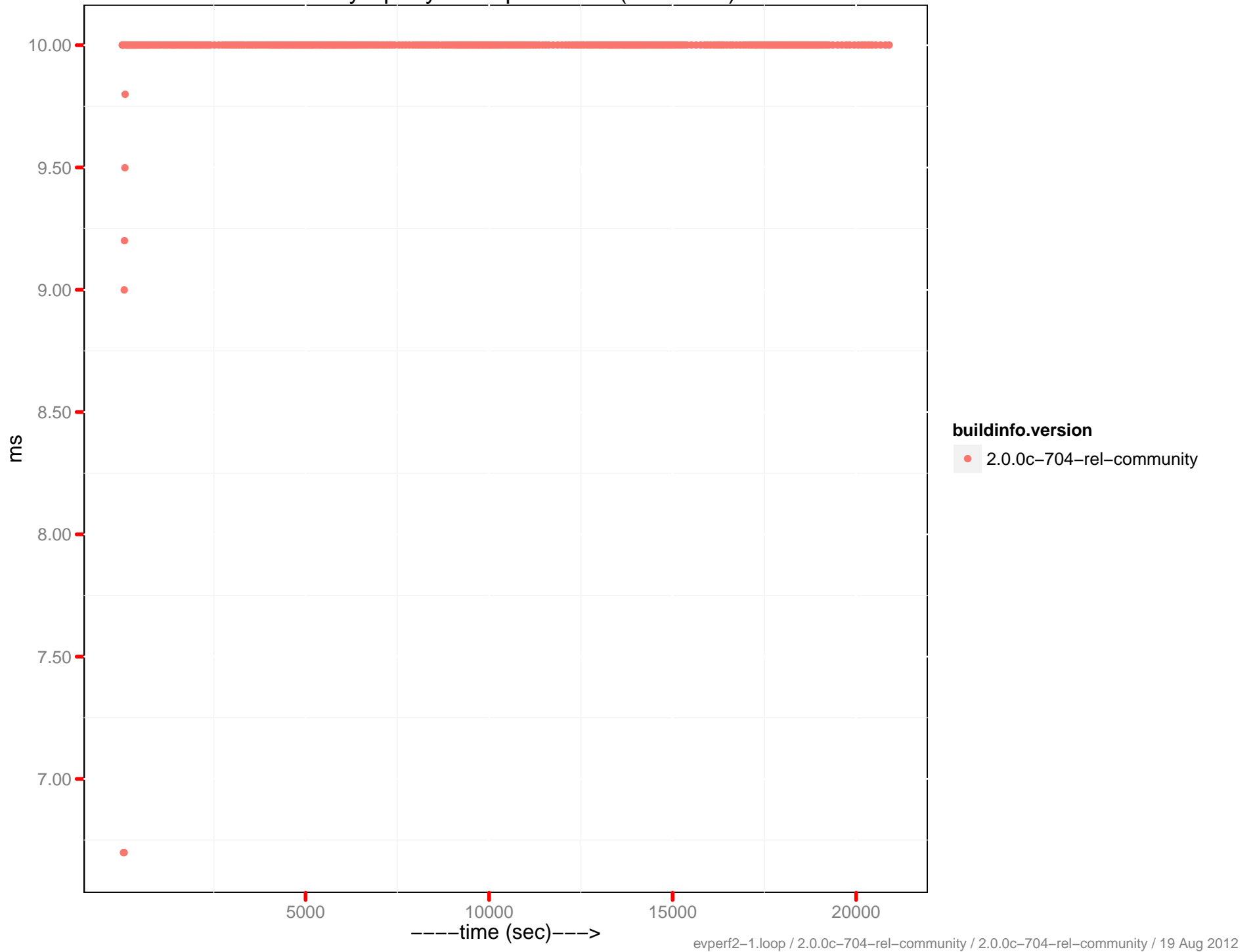
Latency-query 90th percentile (0 – 10ms)



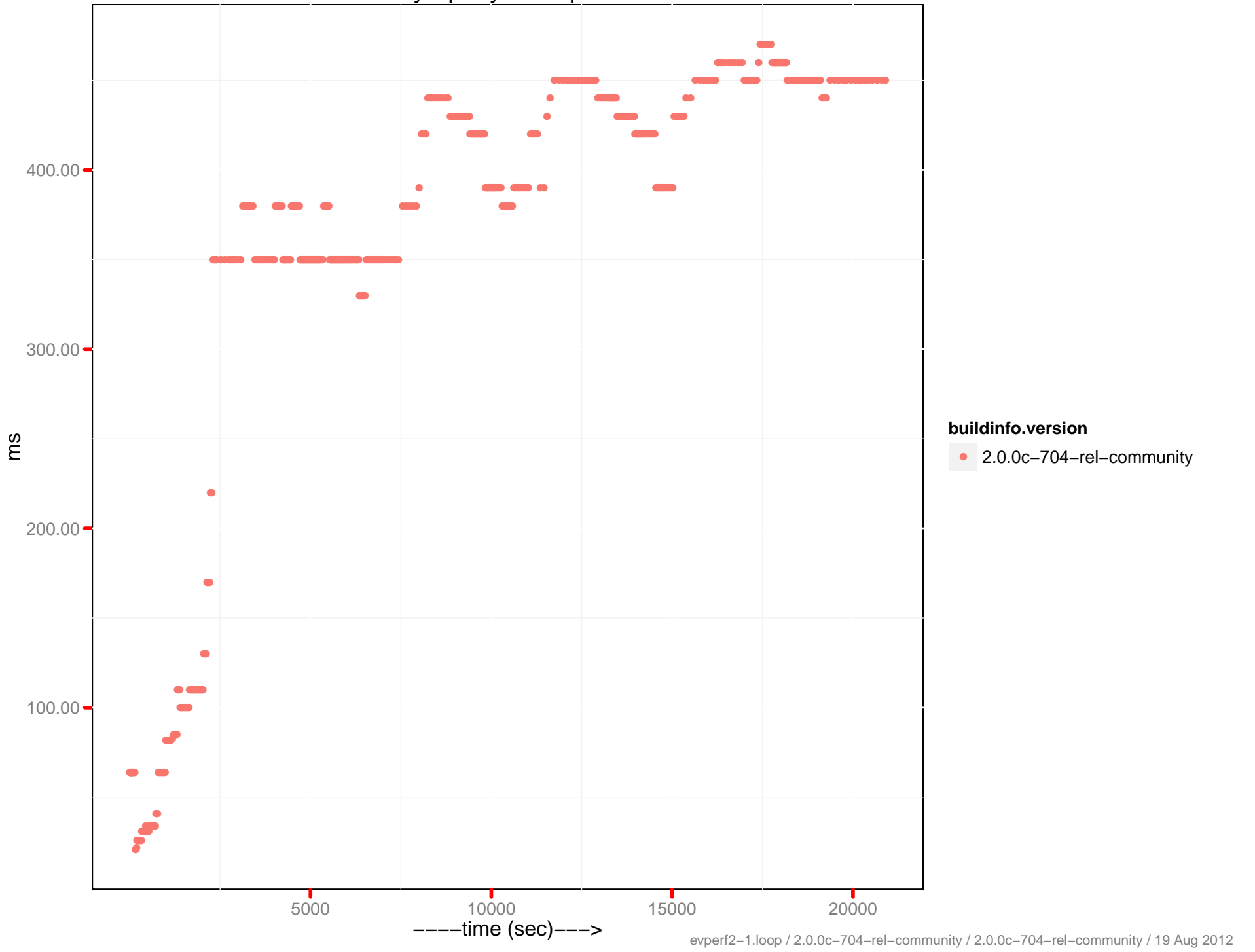
Latency-query 95th percentile



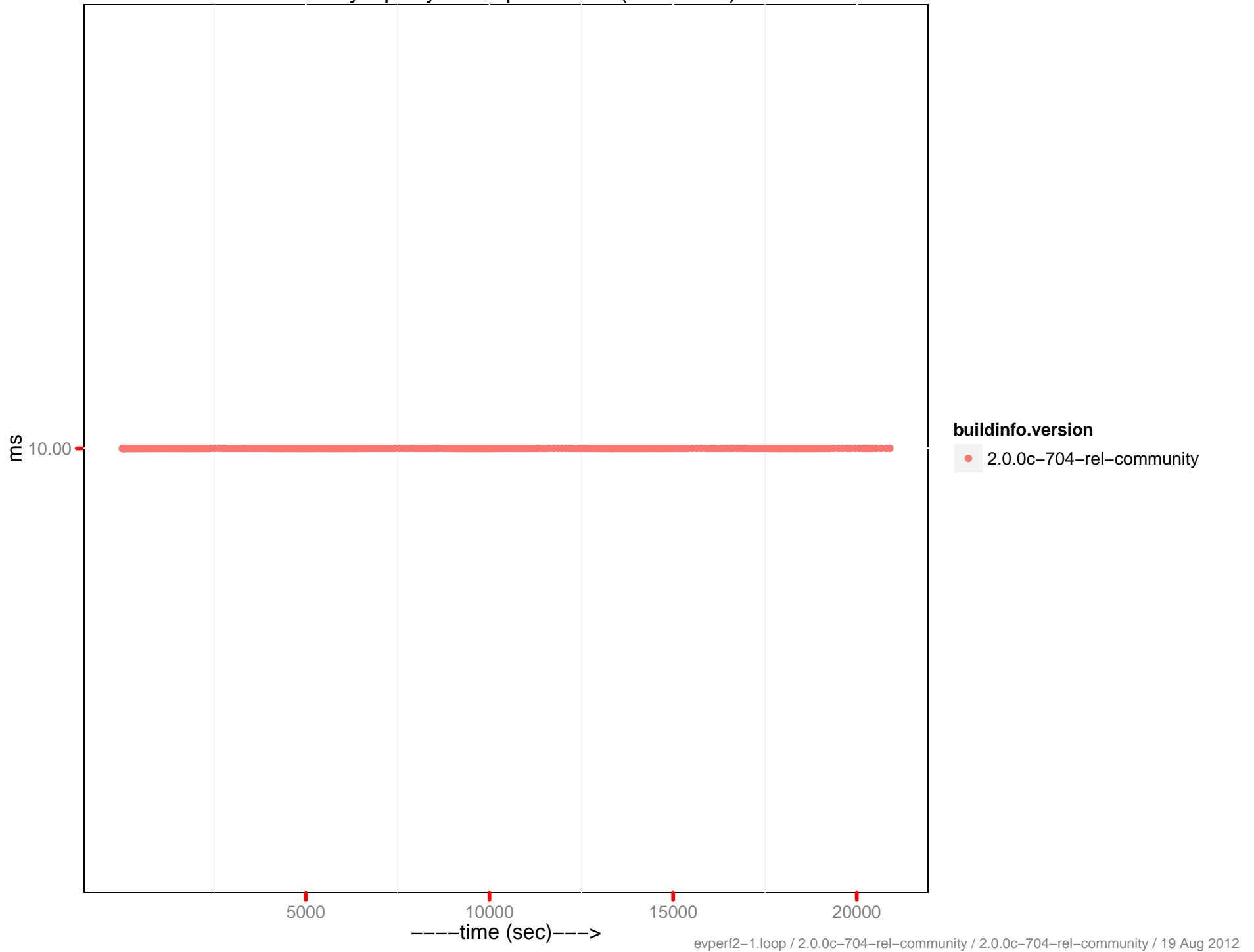
Latency-query 95th percentile (0 – 10ms)



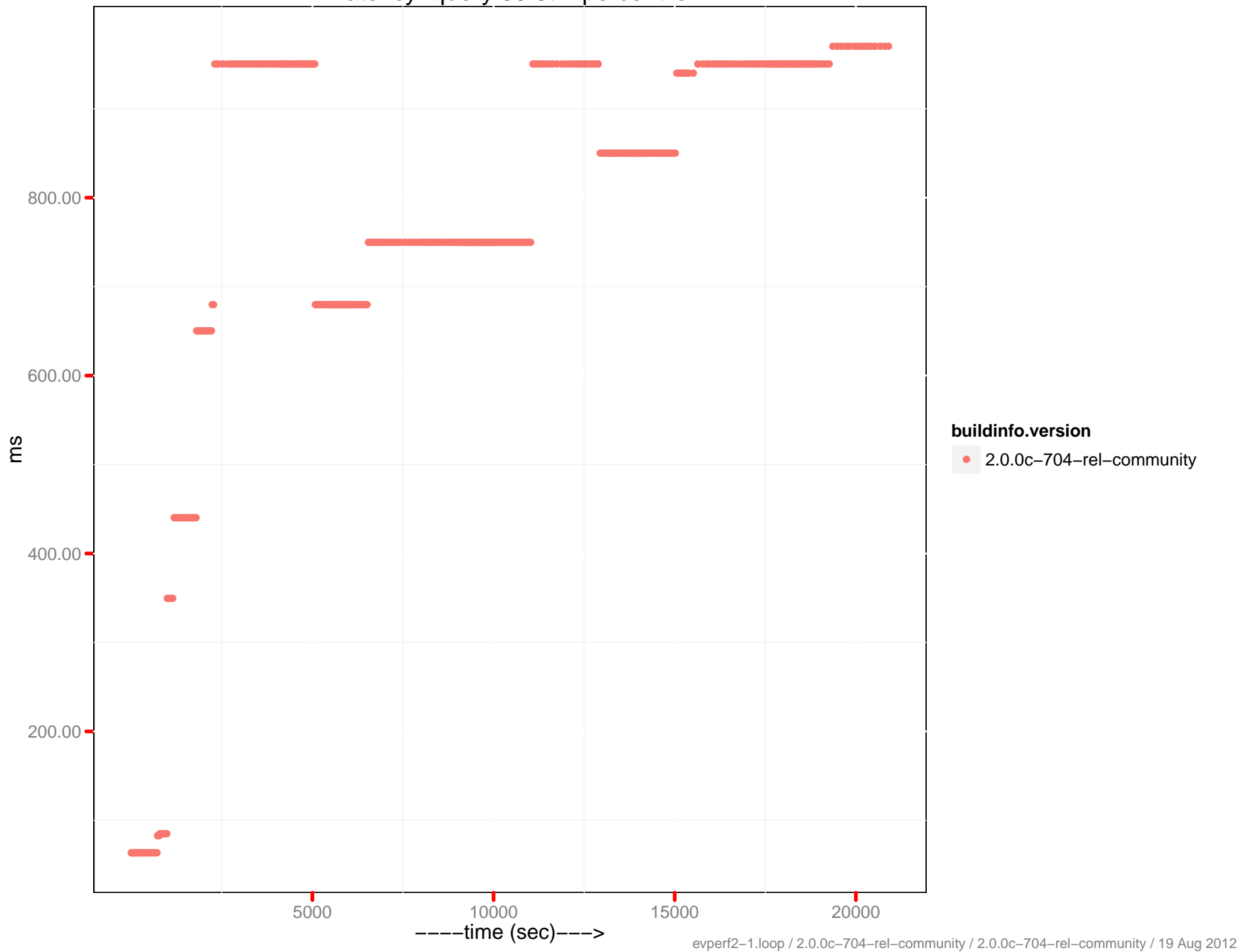
Latency-query 99th percentile



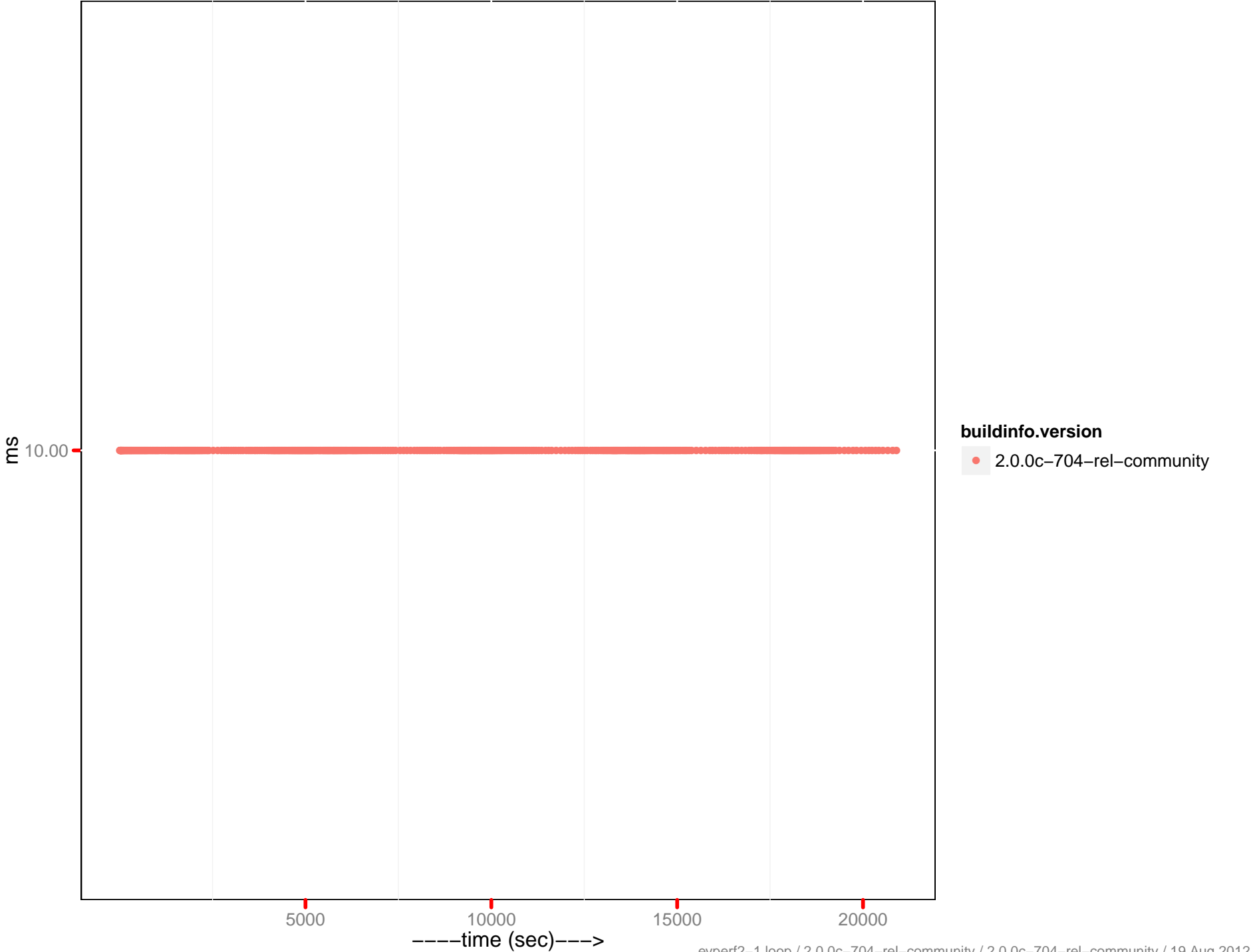
Latency-query 99th percentile (0 - 10ms)



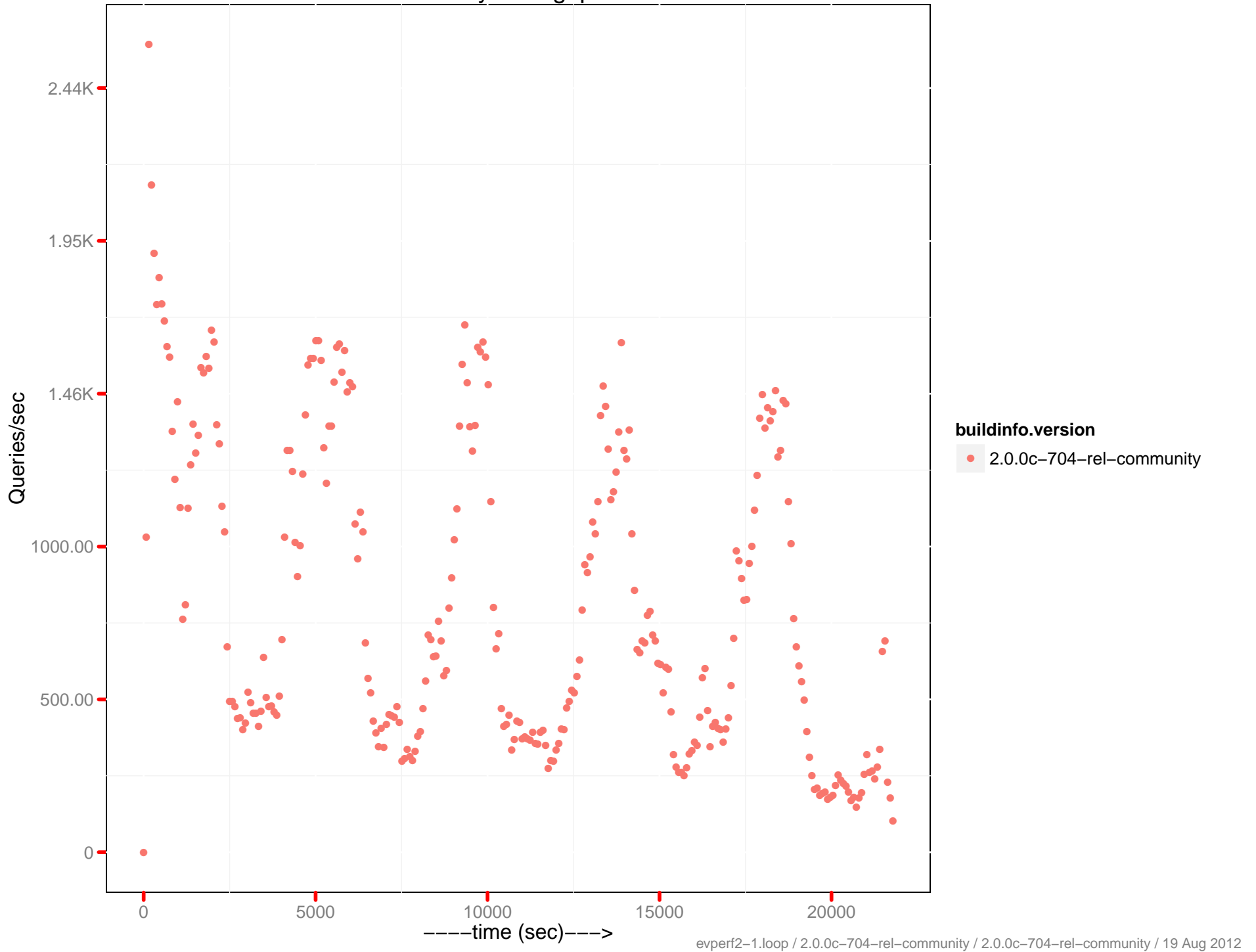
Latency-query 99.9th percentile



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



Query throughput




```
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
```