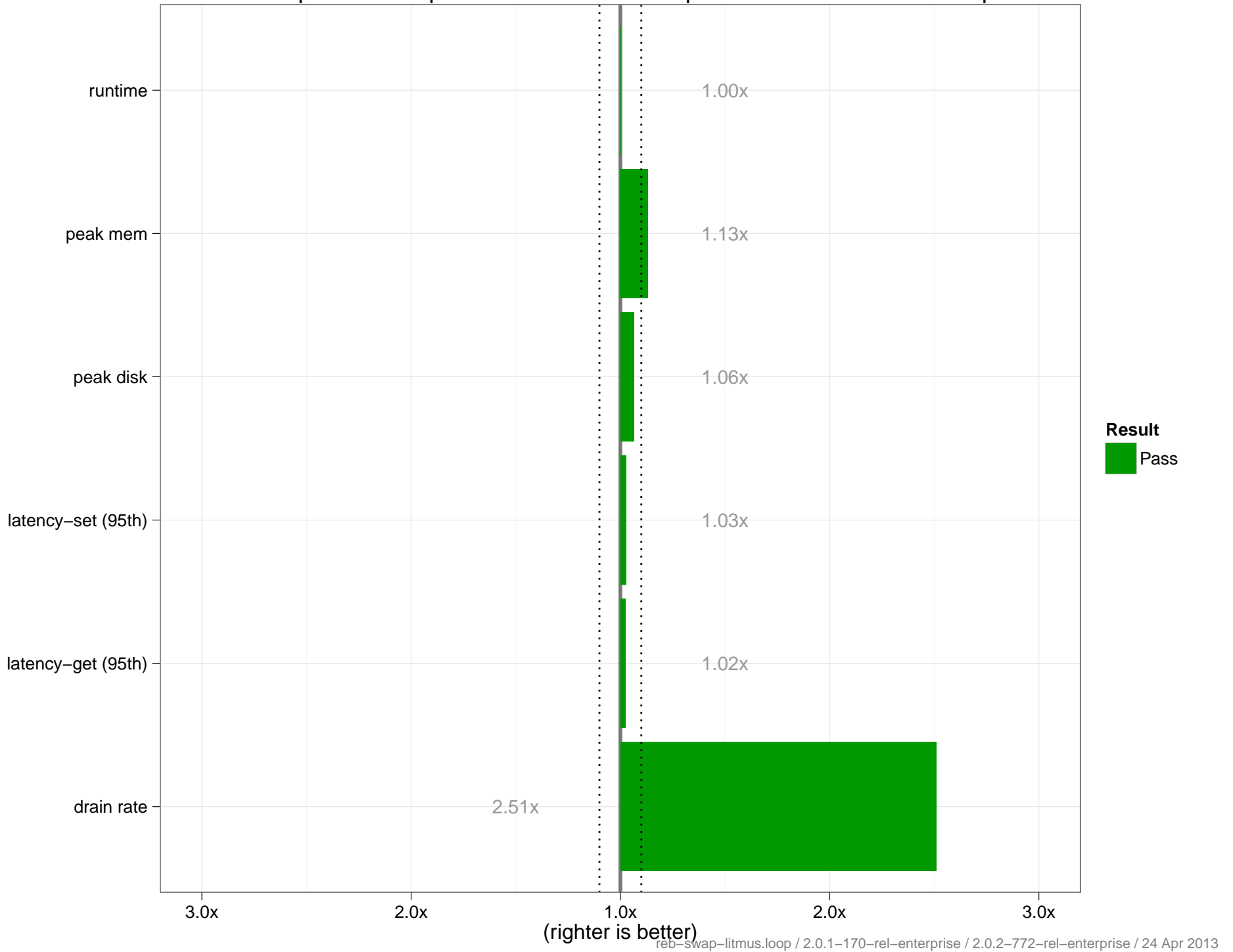
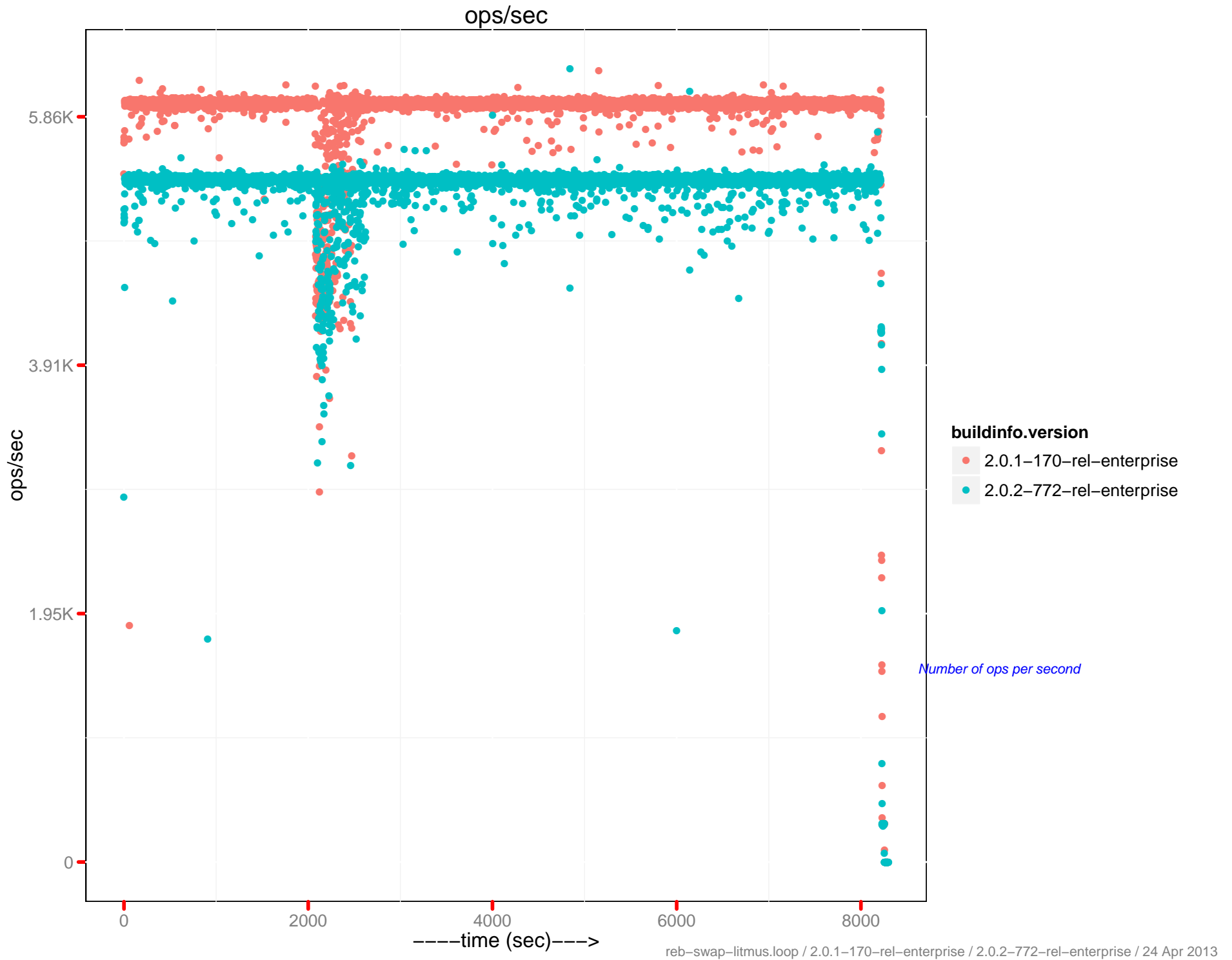


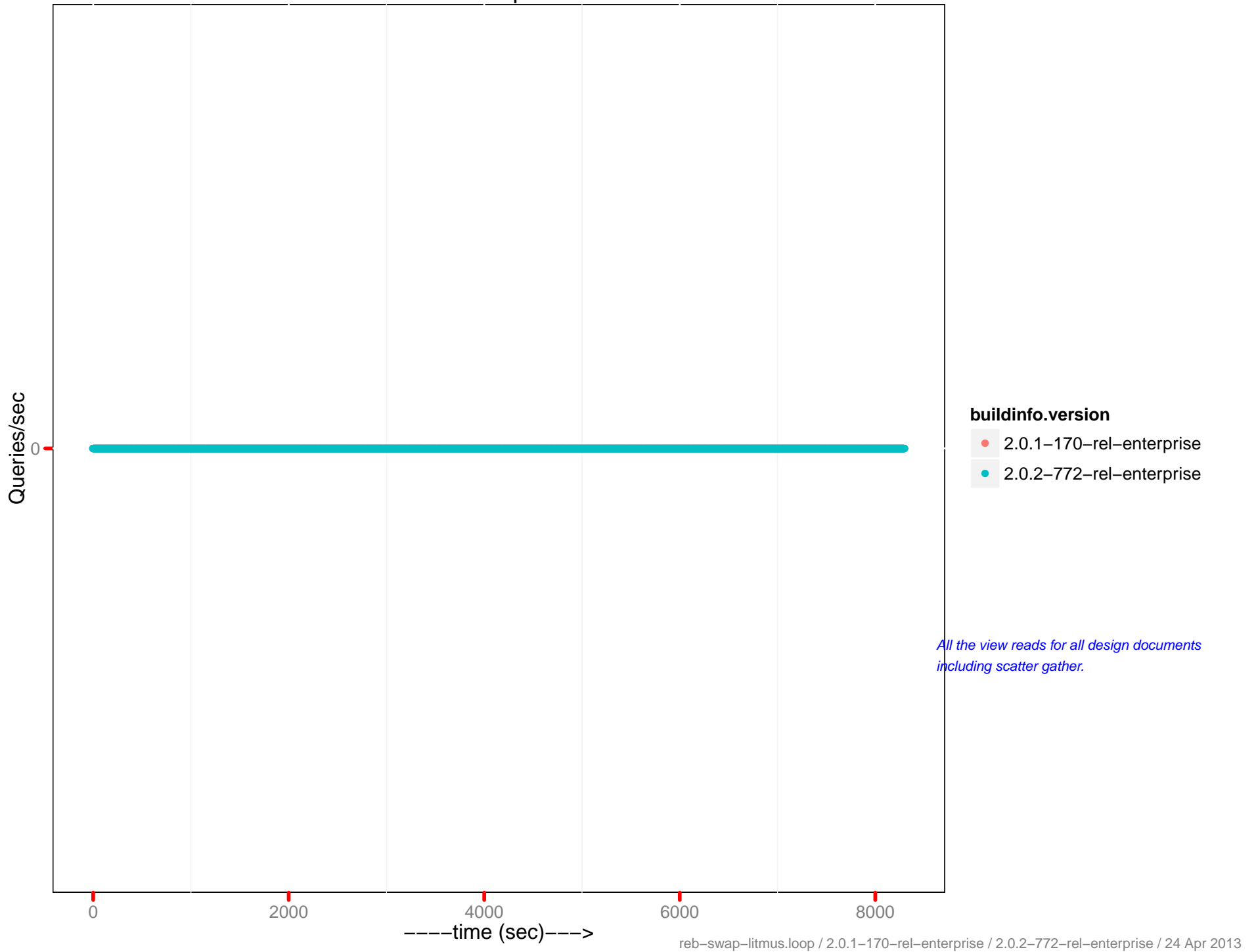
reb-swap-litmus.loop : 2.0.1-170-rel-enterprise : 2.0.2-772-rel-enterprise



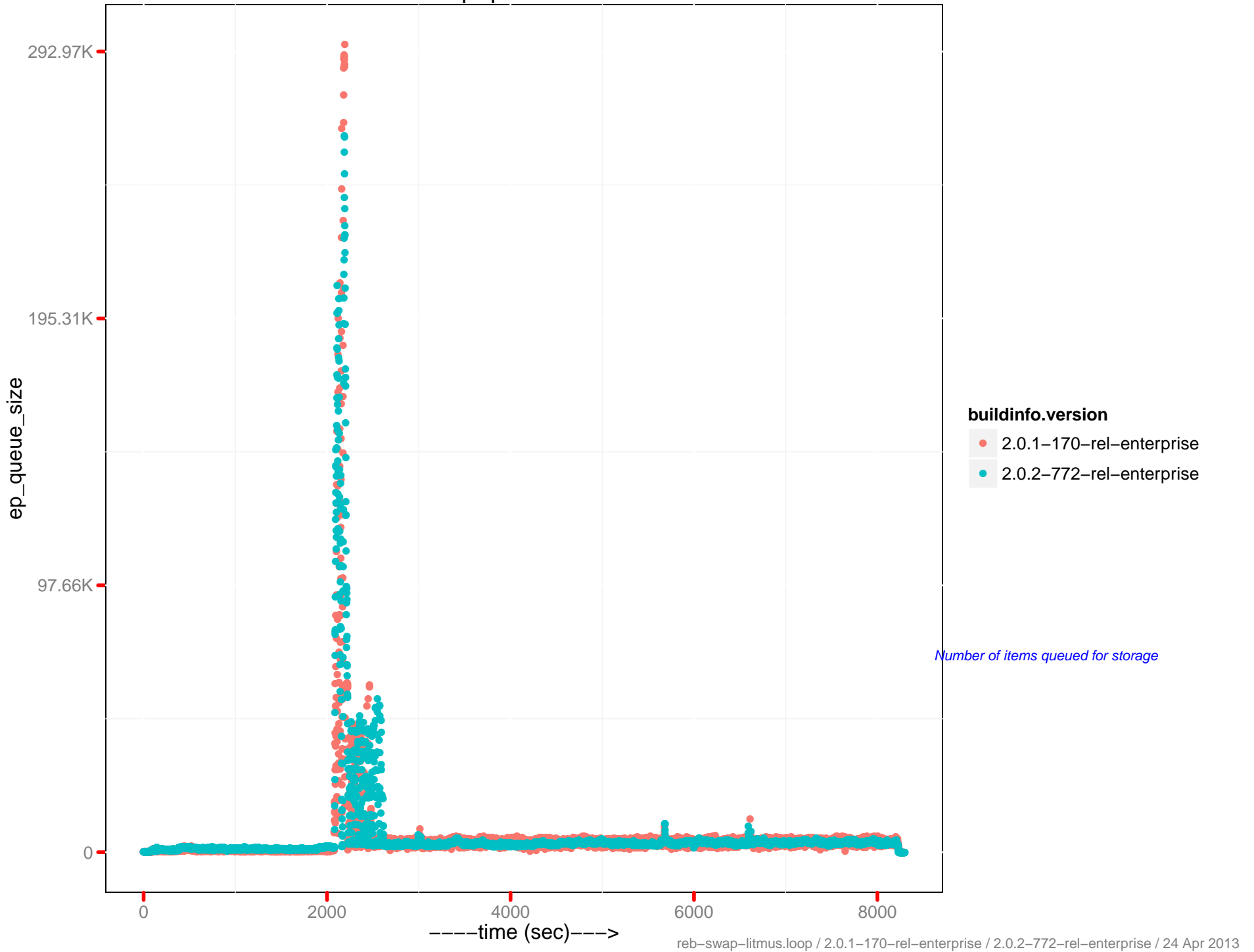
	<b>2.0.1 – 170</b>	<b>2.0.2 – 772</b>
<i>Runtime (in hr)</i>	2.31	2.31
<i>Avg. Drain Rate</i>	1.46K	3.66K
<i>Peak Disk (GB)</i>	22.87	21.49
<i>Peak Memory (GB)</i>	33781.14	29831.95
<i>Avg. OPS</i>	5.91K	5.32K
<i>Avg. mem memcached (GB)</i>	26408.07	24740.2
<i>Avg. mem beam.smp (MB)</i>	6993860.31	5127892.34
<i>Avg. CPU rate (%)</i>	41.01	37.69
<i>Latency-get (90th) (ms)</i>	2.91	2.88
<i>Latency-get (95th) (ms)</i>	4.02	3.93
<i>Latency-get (99th) (ms)</i>	8.13	8.26
<i>Latency-set (90th) (ms)</i>	3.16	3.06
<i>Latency-set (95th) (ms)</i>	4.22	4.11
<i>Latency-set (99th) (ms)</i>	8.1	8.45
<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	0
<i>Avg. XDC ops/sec</i>	NaN	NaN
<i>Avg. XDC docs to replicate</i>	NaN	NaN
<i>Rebalance Time (sec)</i>	428.02	563.04
<i>Testrunner Version</i>	4a2b75f	95e8c41



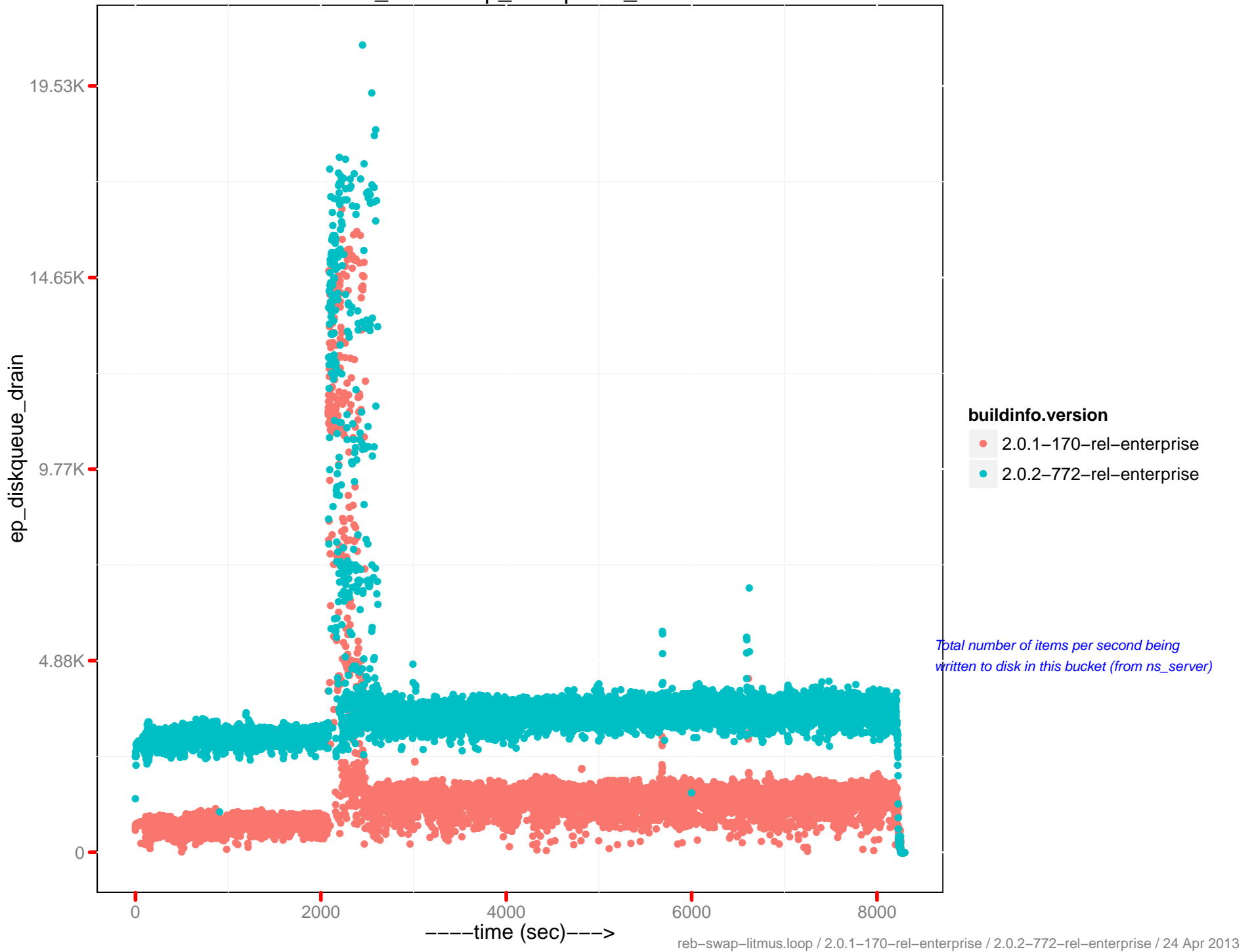
# View read per sec.



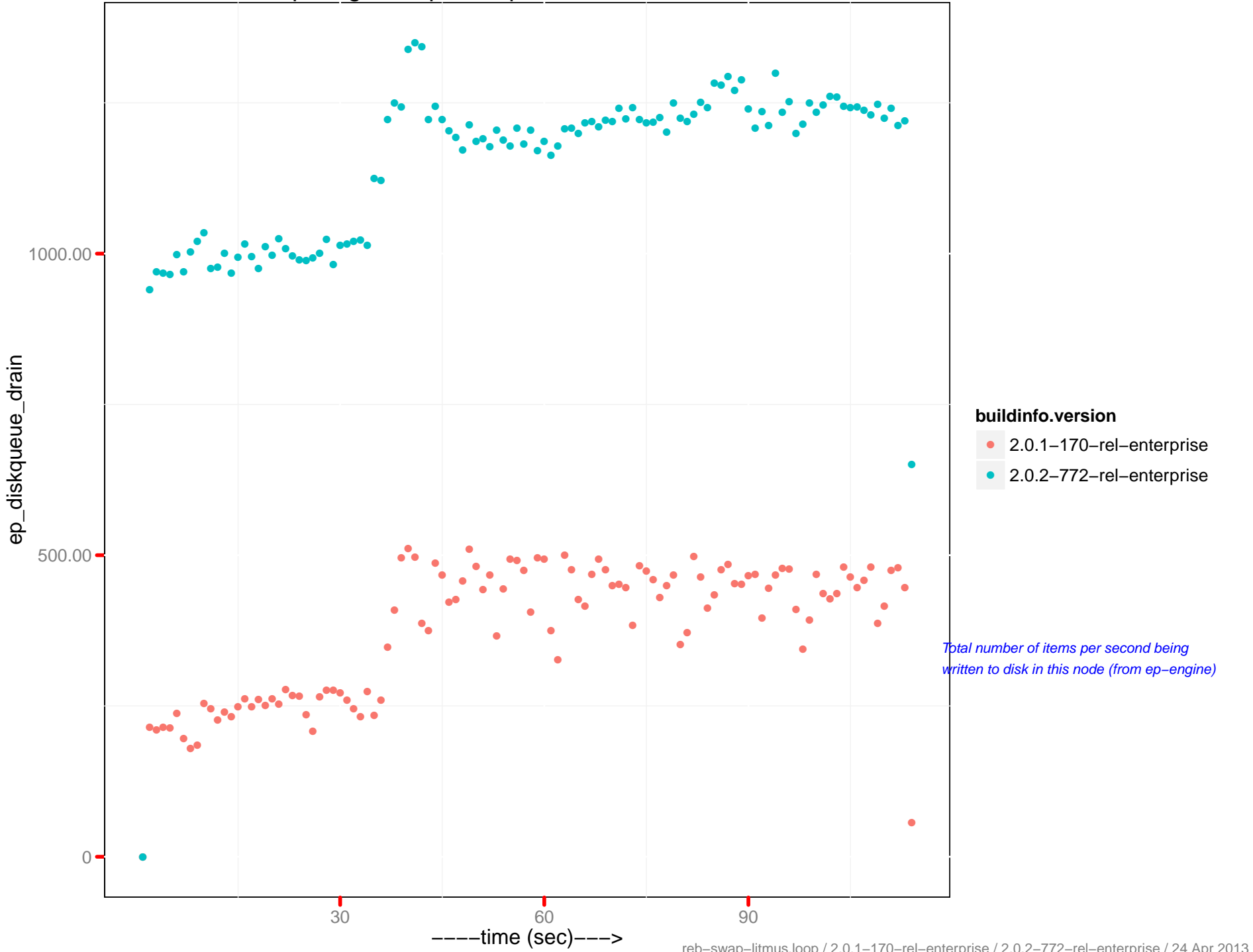
# ep queue size



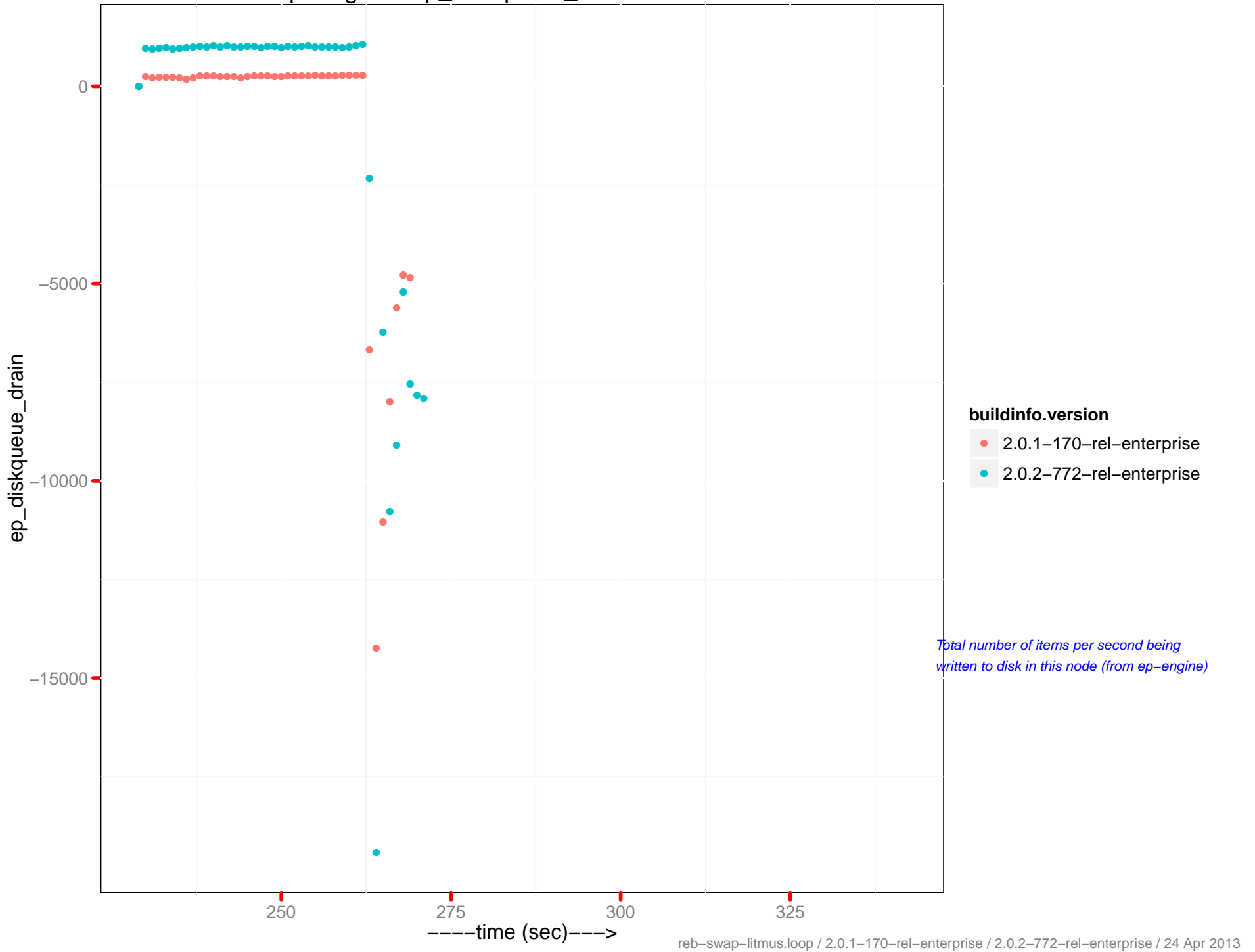
# ns\_server: ep\_diskqueue\_drain



ep-engine : ep\_diskqueue\_drain - 10.6.2.37

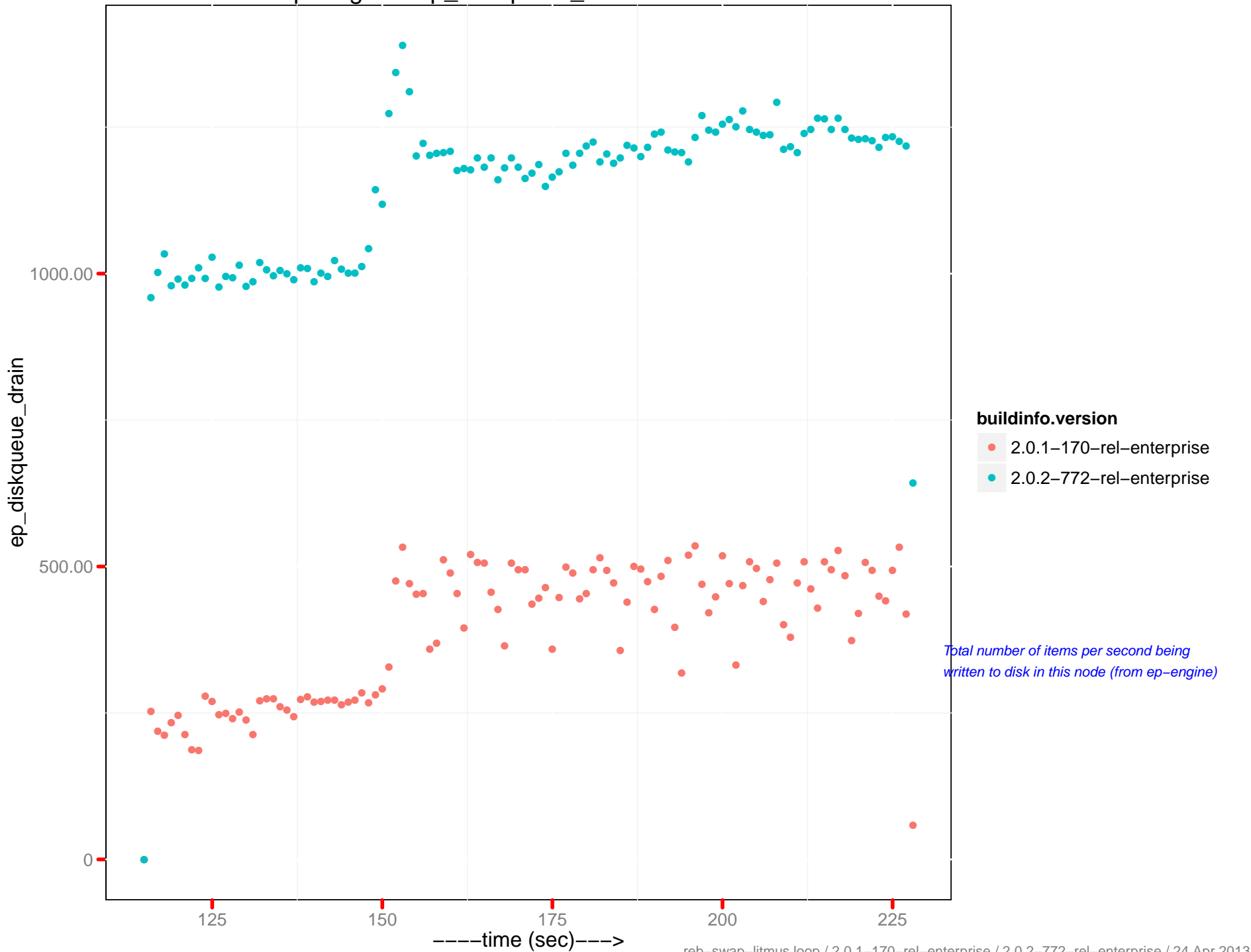


# ep-engine : ep\_diskqueue\_drain - 10.6.2.38

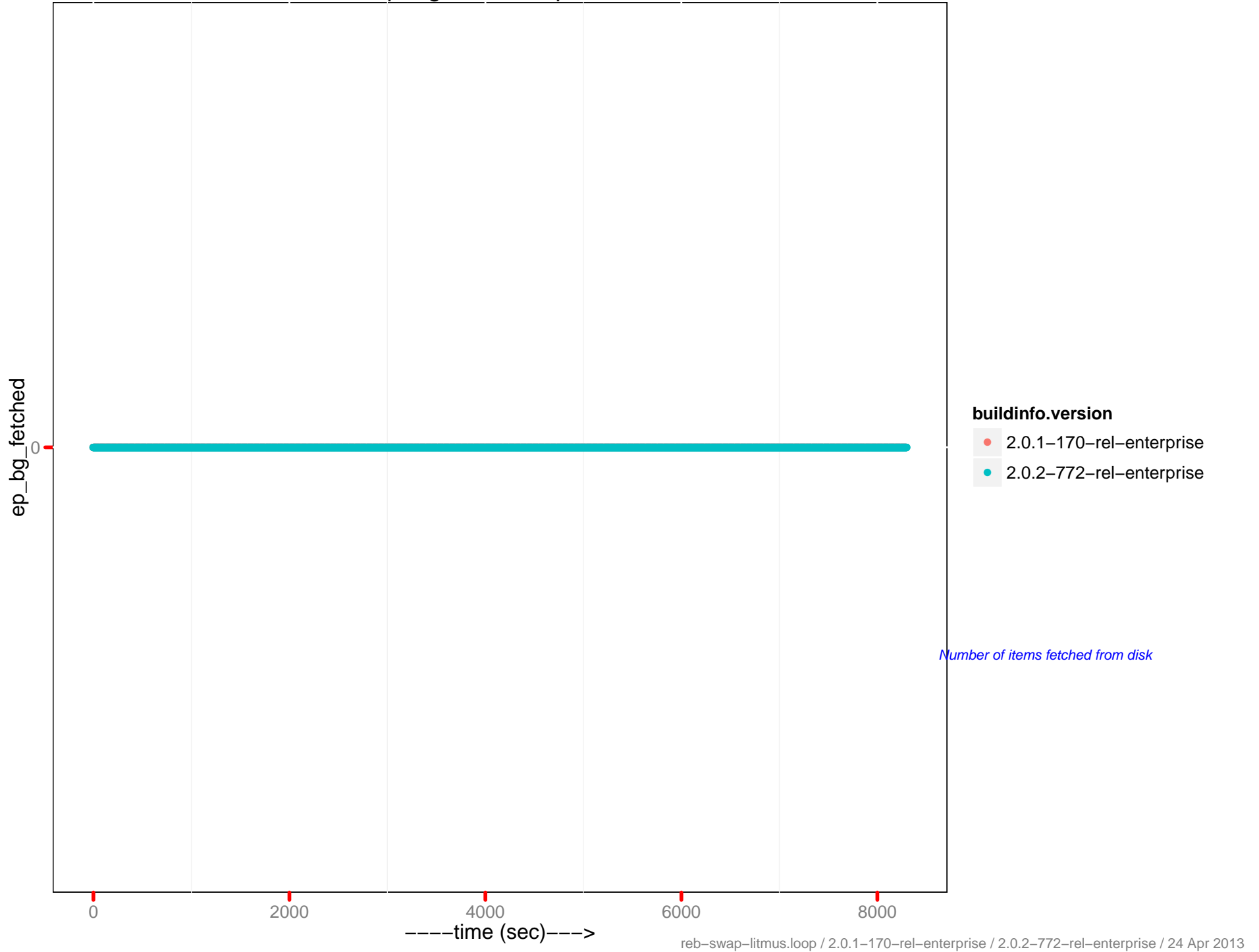




ep-engine : ep\_diskqueue\_drain - 10.6.2.39

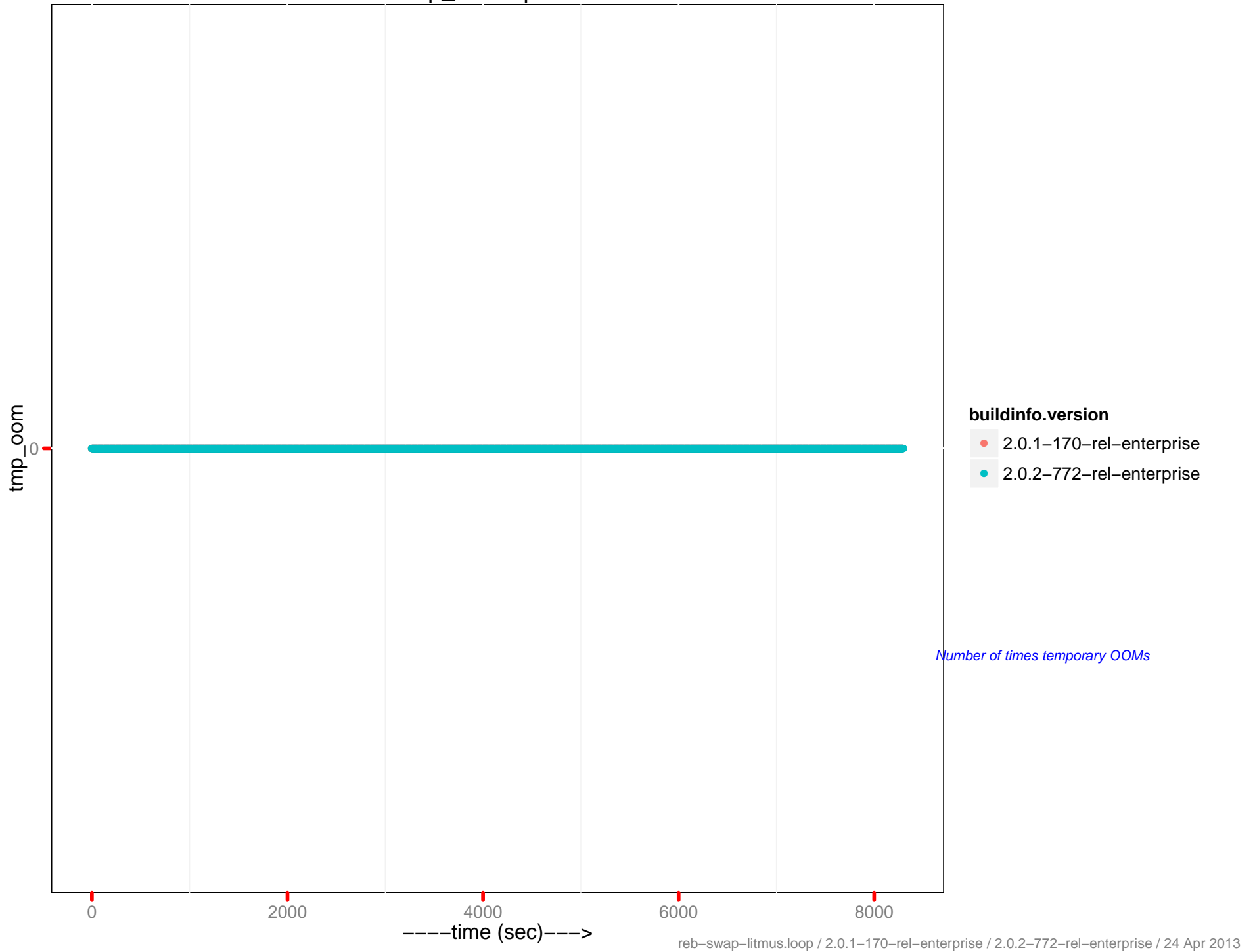


# ep\_bg\_fetched ops/sec

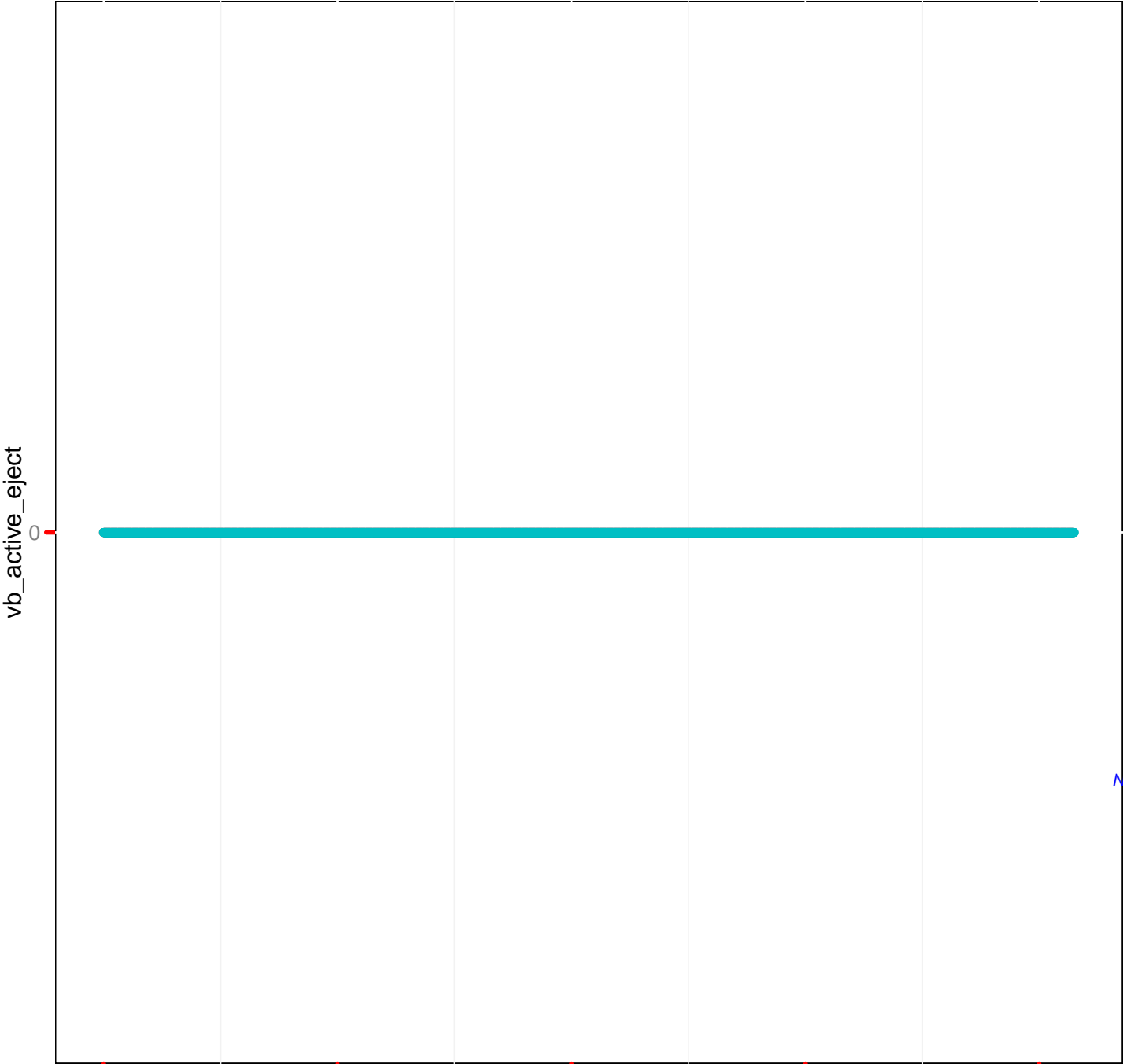


Number of items fetched from disk

# tmp\_oom ops/sec



vb\_active\_eject/sec



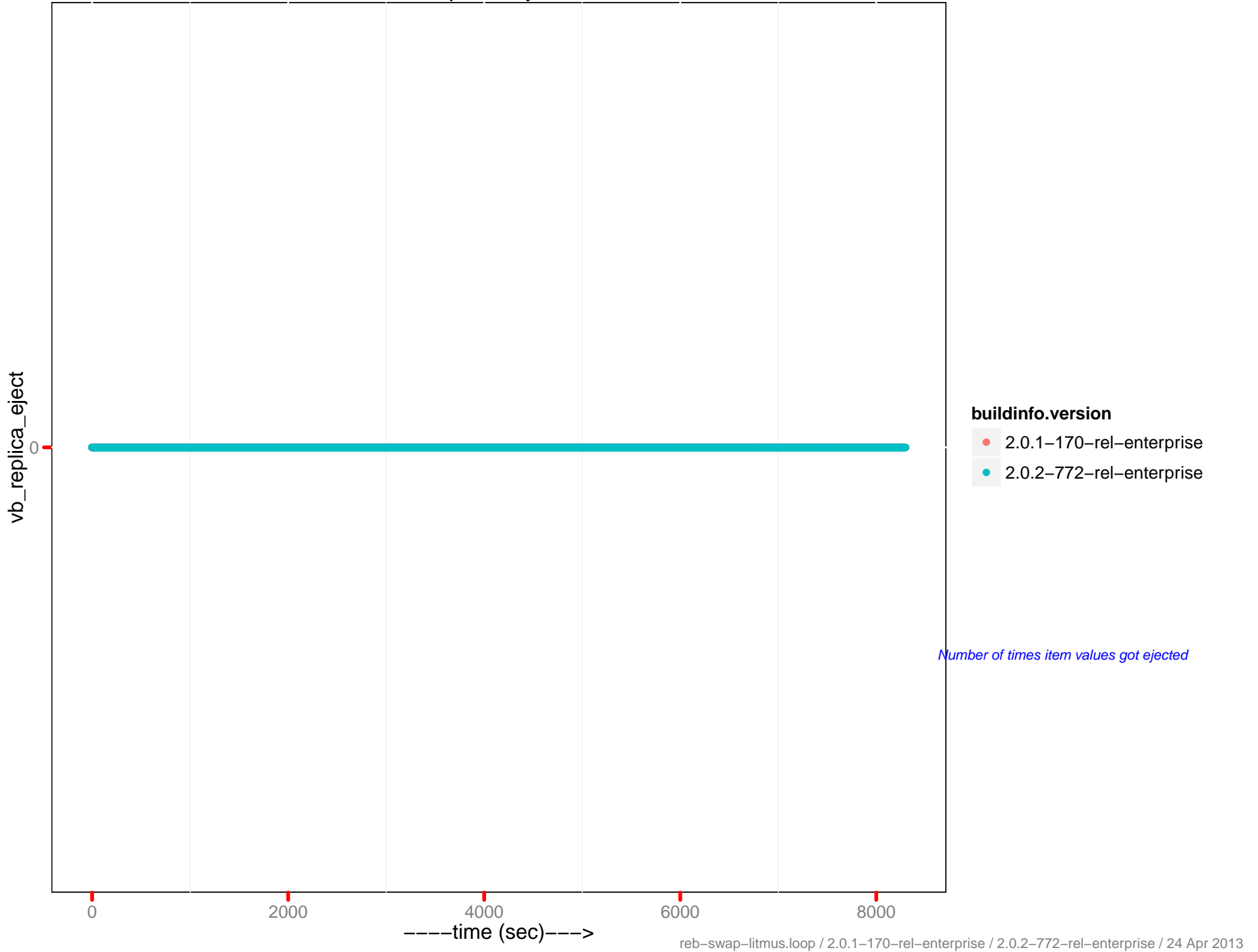
- buildinfo.version**
- 2.0.1-170-rel-enterprise
  - 2.0.2-772-rel-enterprise

*Number of times item values got ejected*

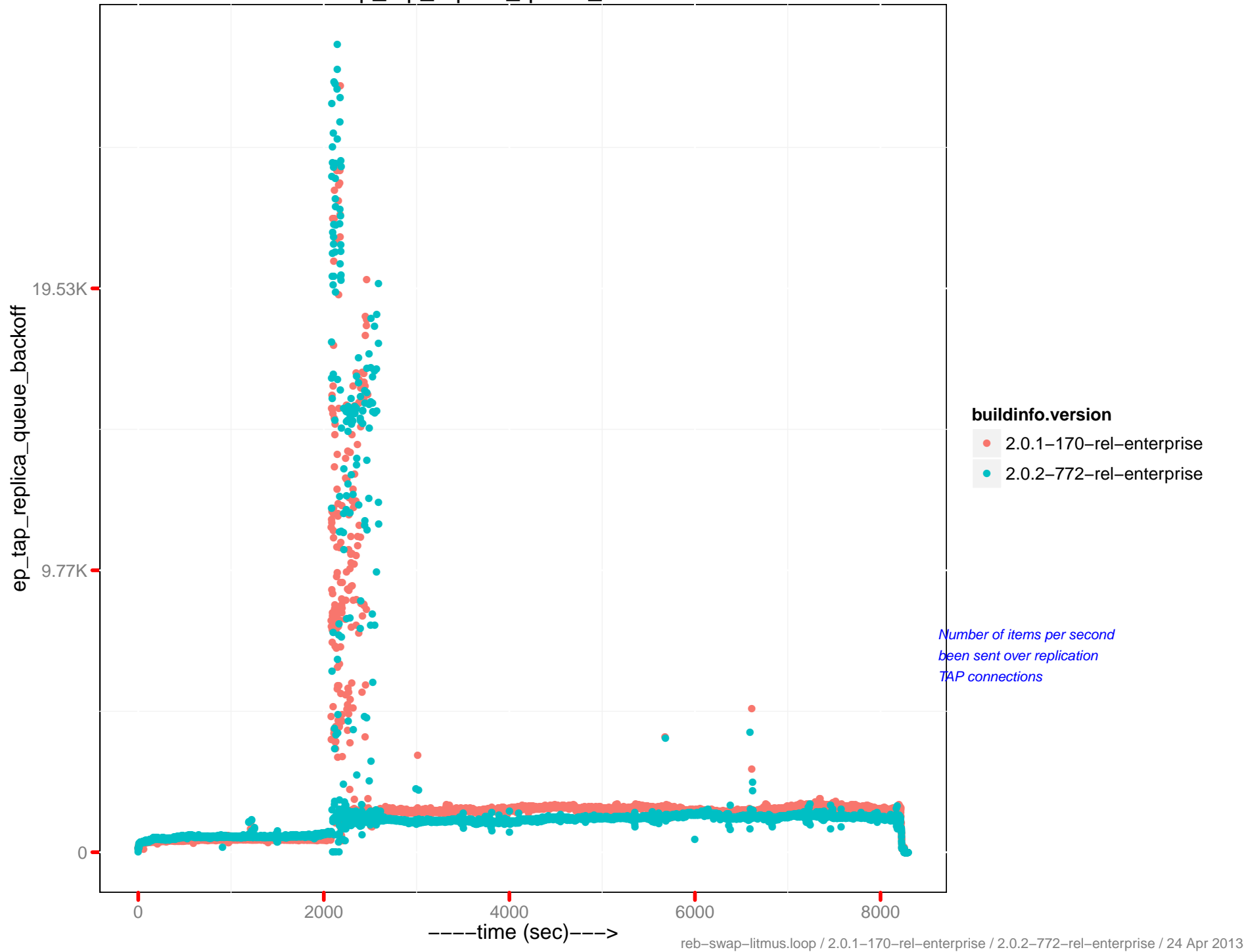
0 2000 4000 6000 8000

----time (sec)---->

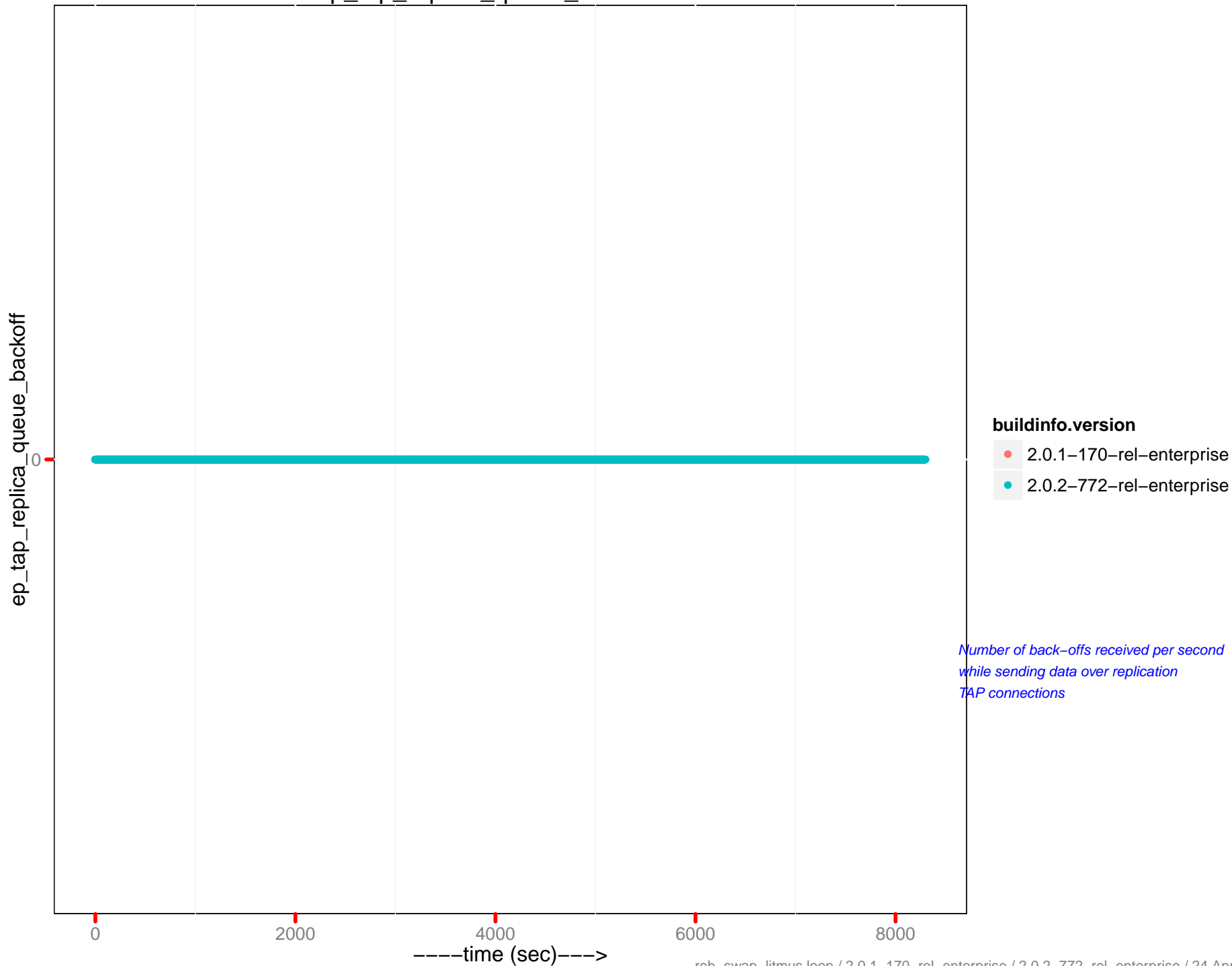
# vb\_replica\_eject/sec



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

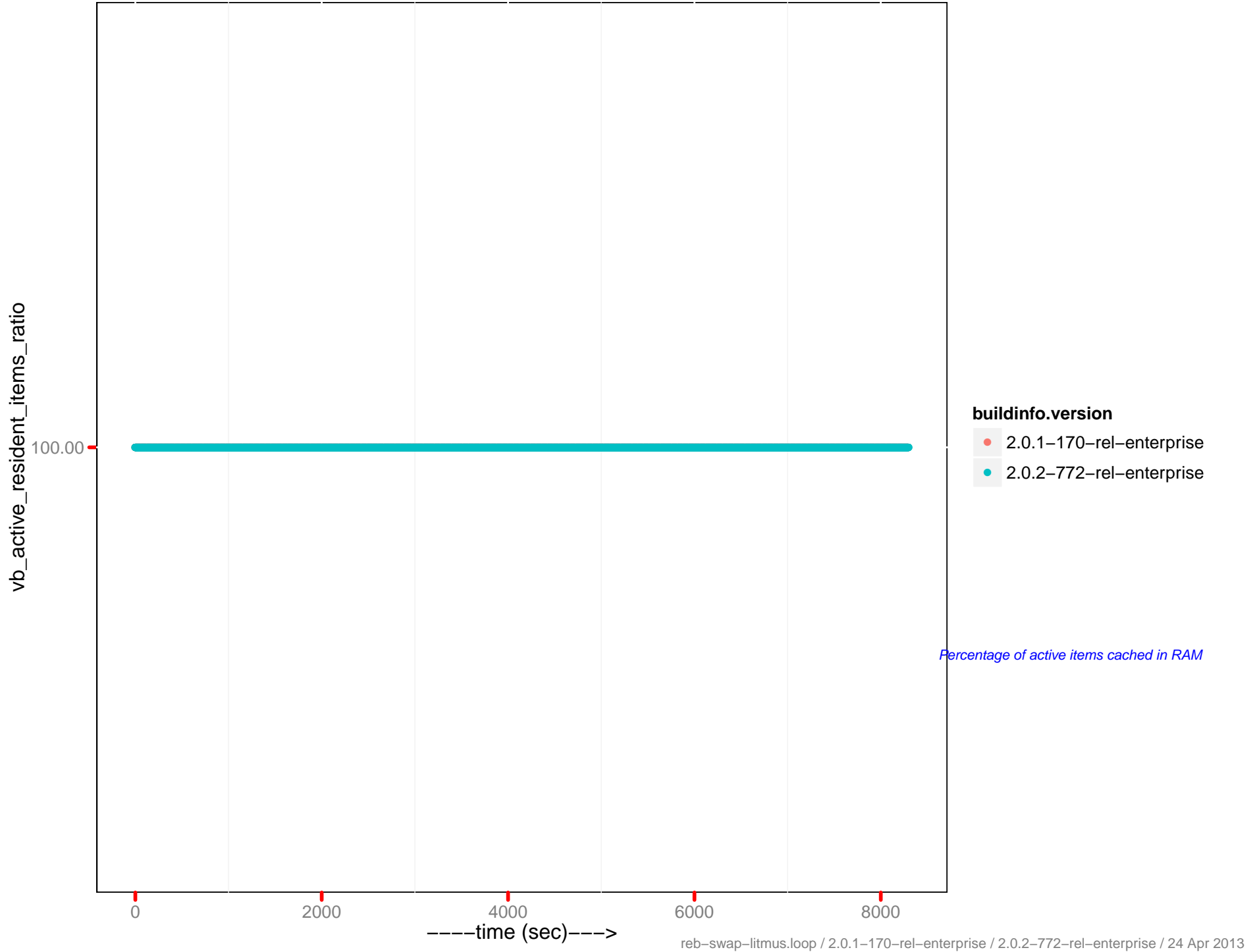


# ep\_tap\_replica\_queue\_backoff/sec



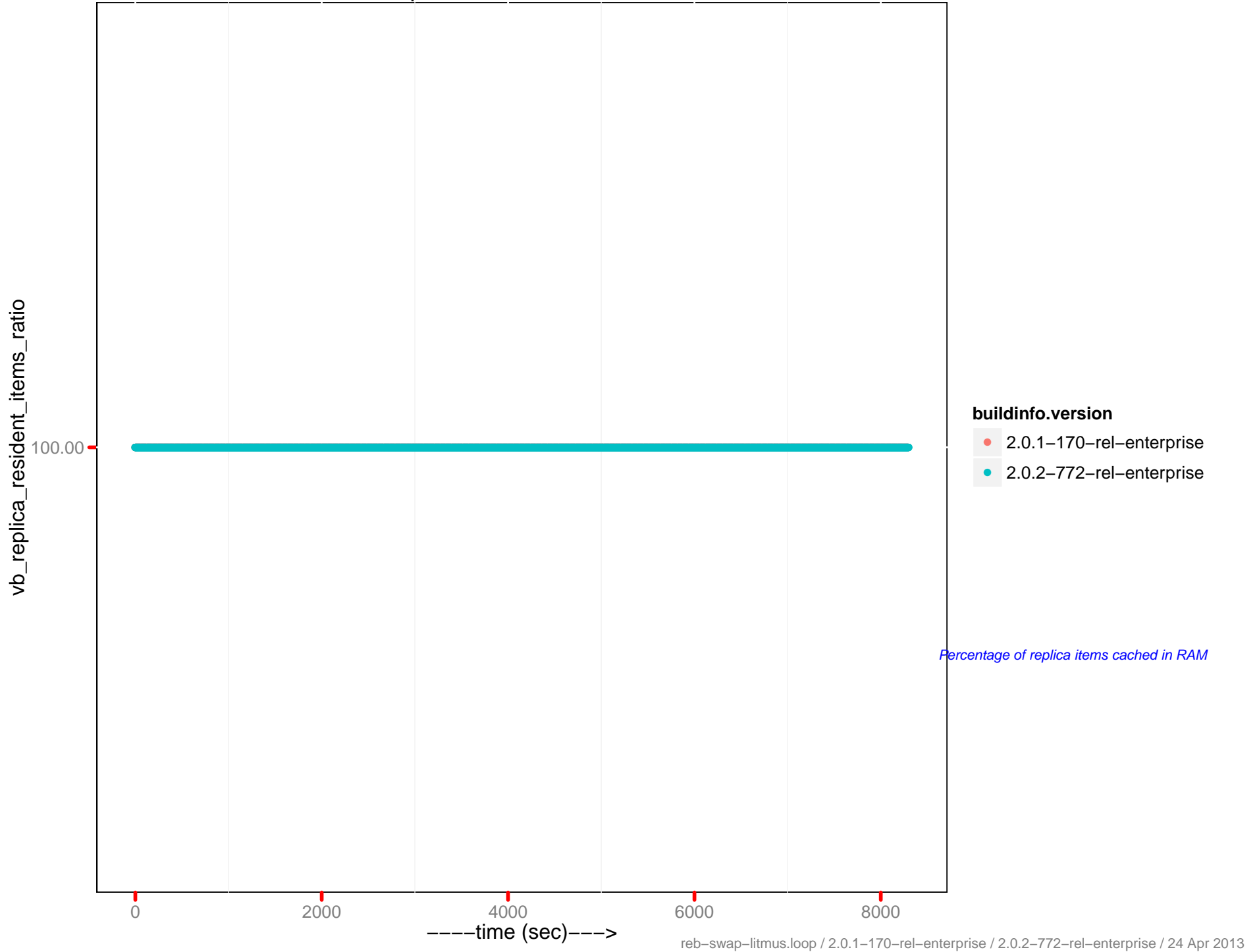
*Number of back-offs received per second  
while sending data over replication  
TAP connections*

# vb\_active\_resident\_items\_ratio



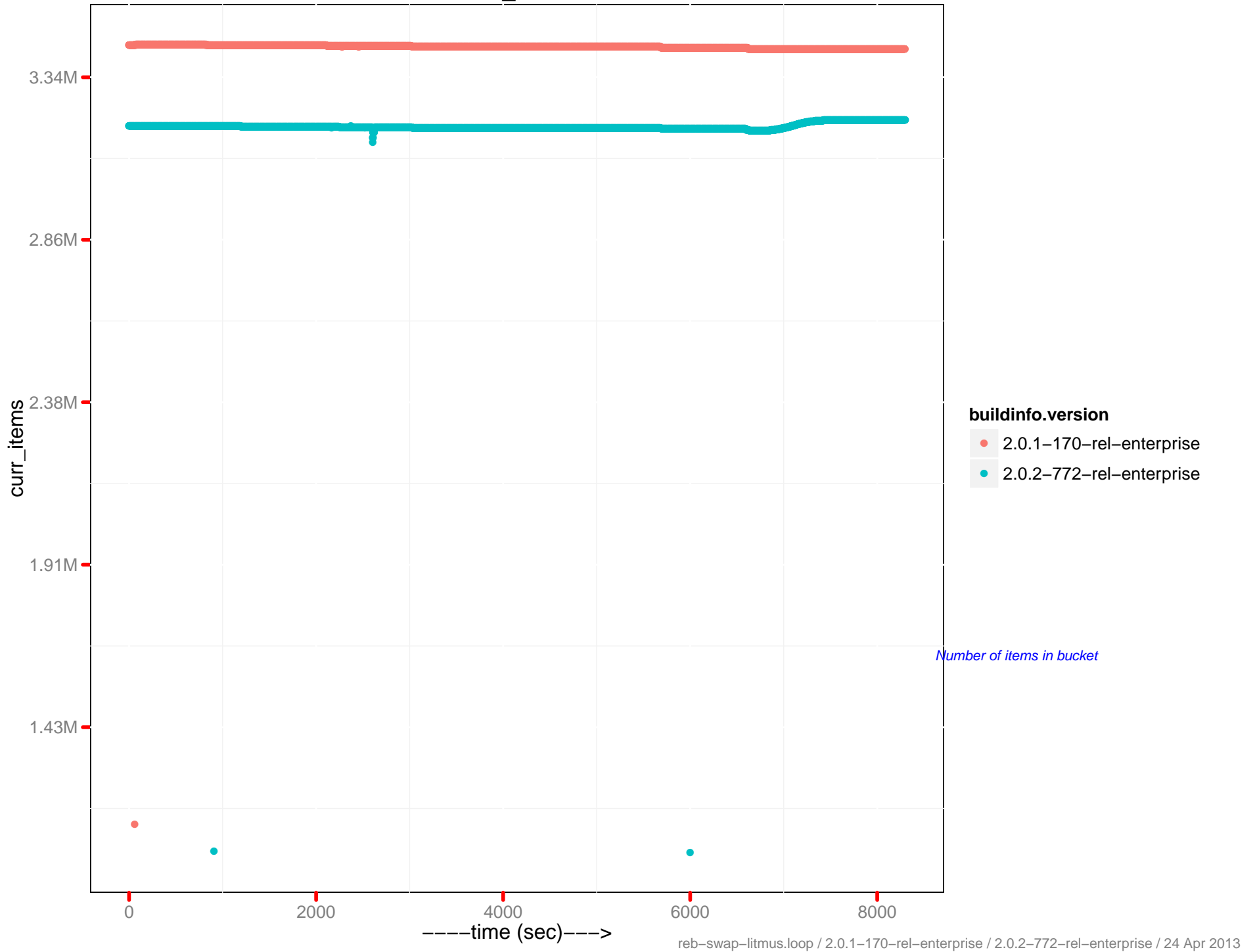


# vb\_replica\_resident\_items\_ratio



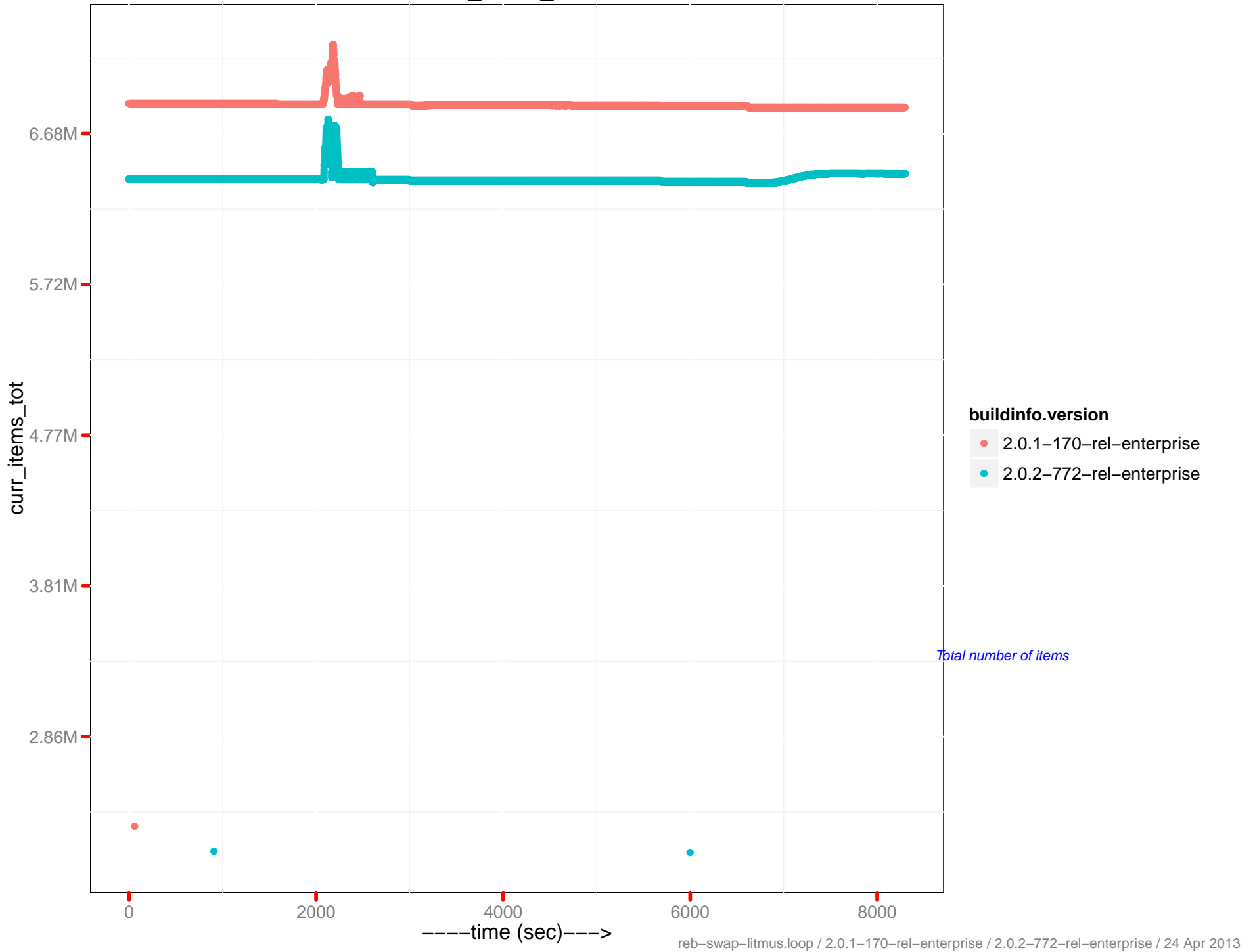
*Percentage of replica items cached in RAM*

# curr\_items

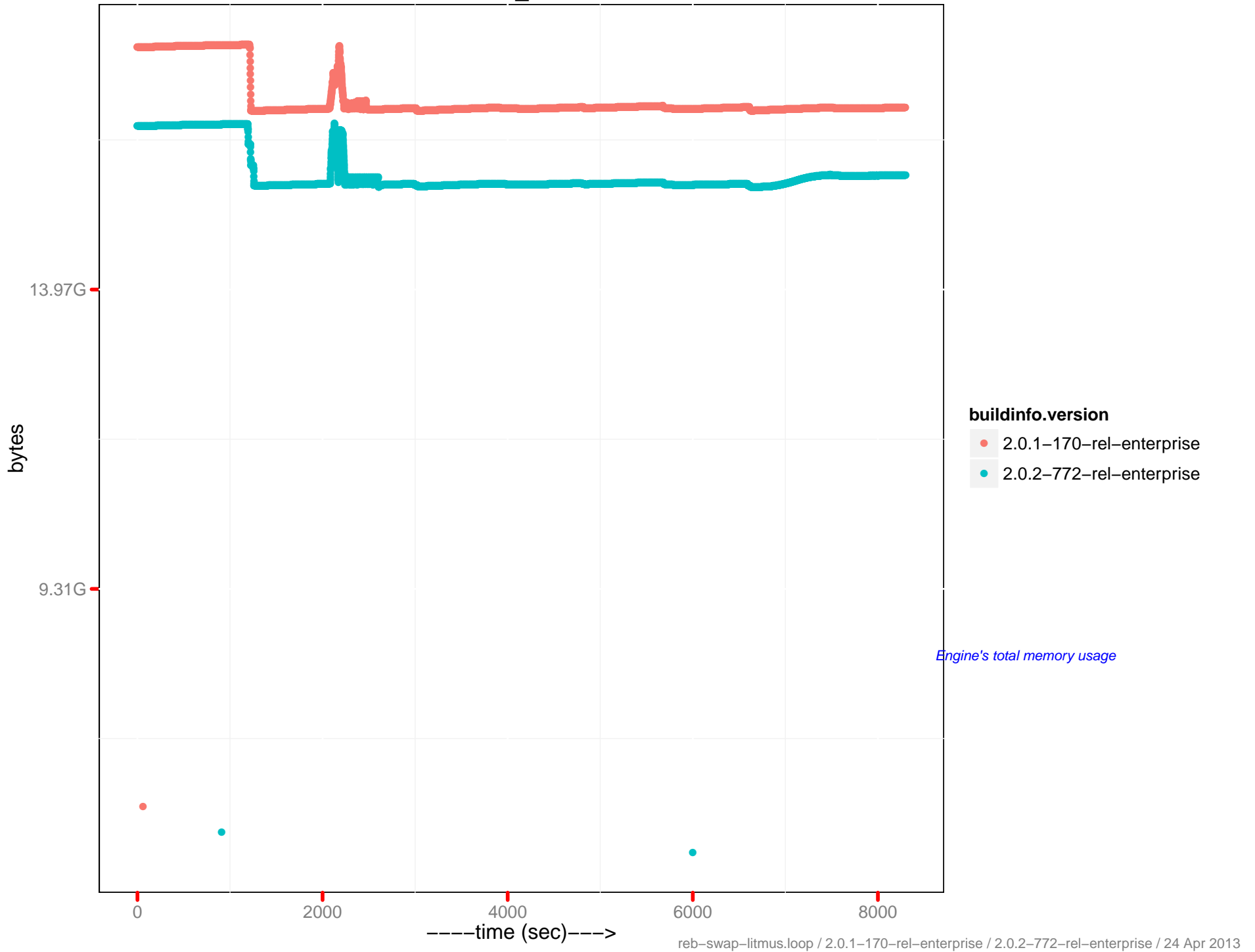


Number of items in bucket

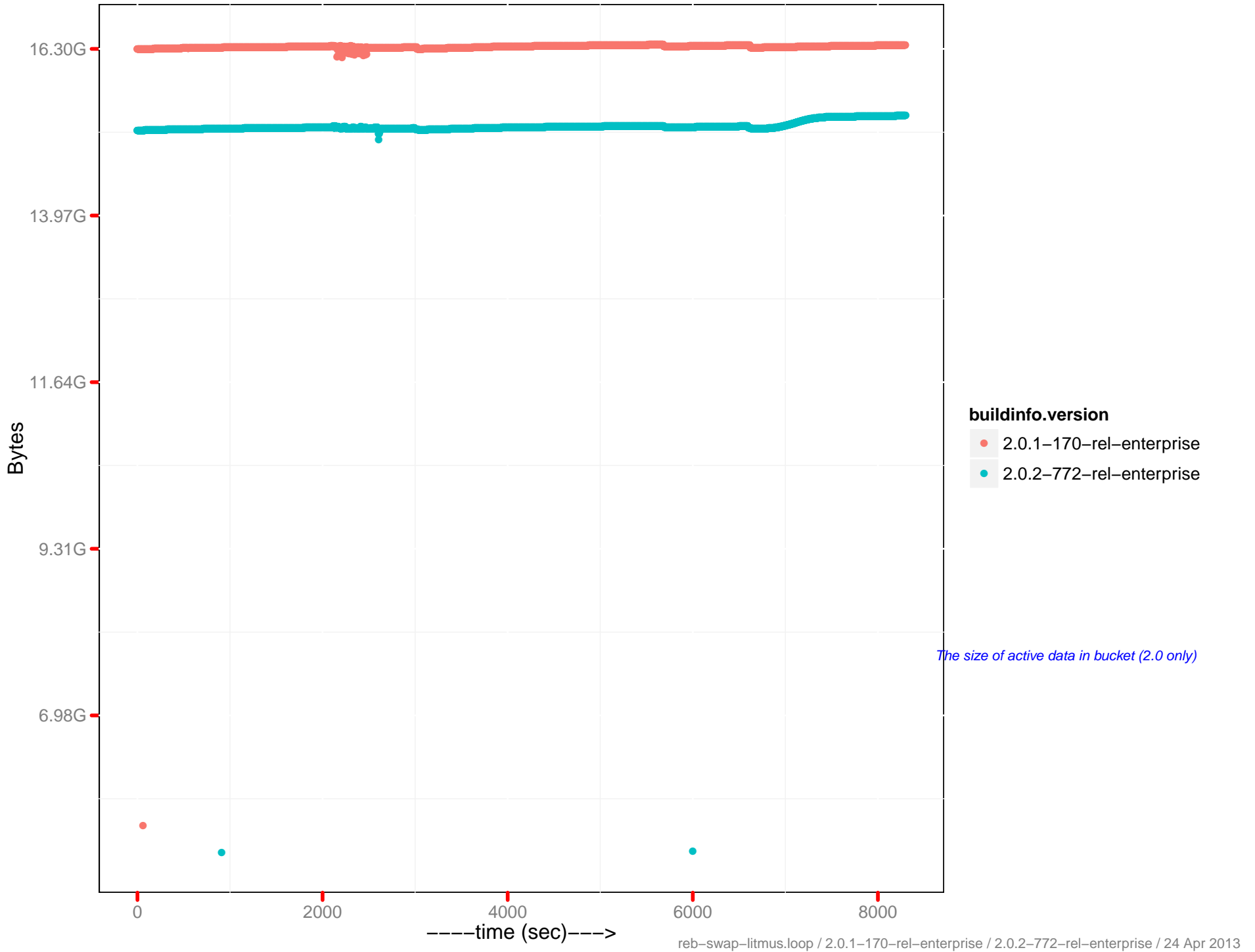
# cur\_items\_total



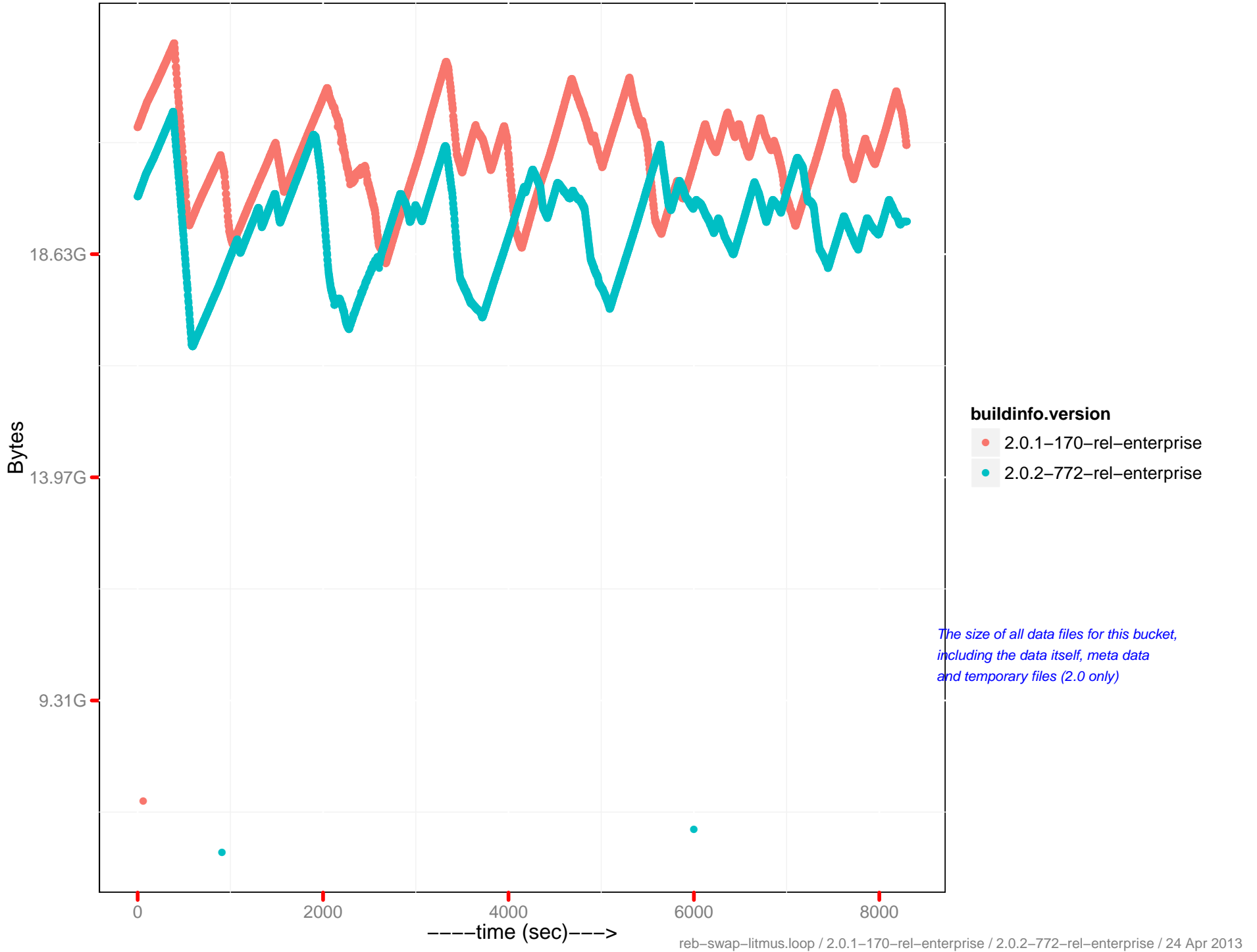
# mem\_used



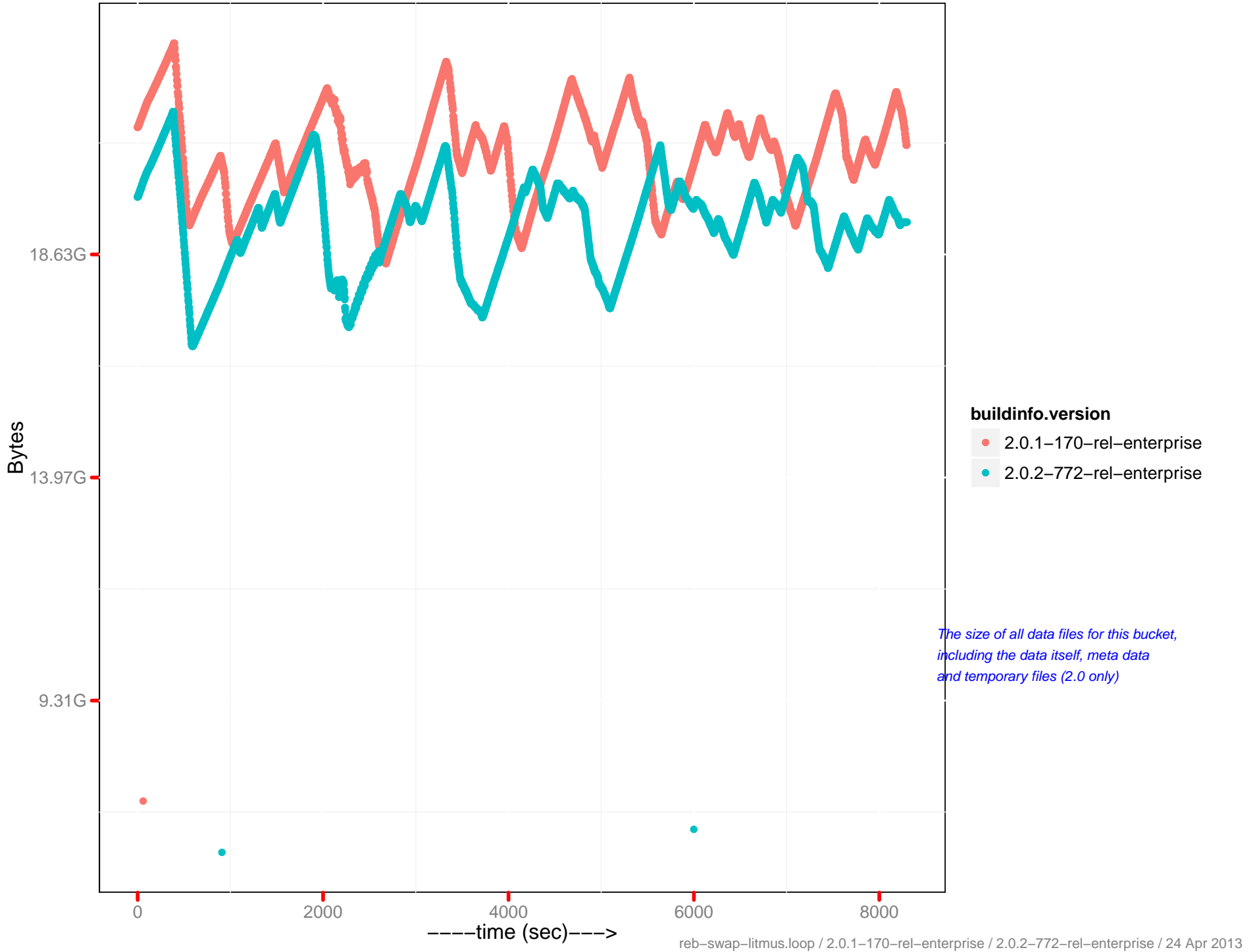
# Docs data size



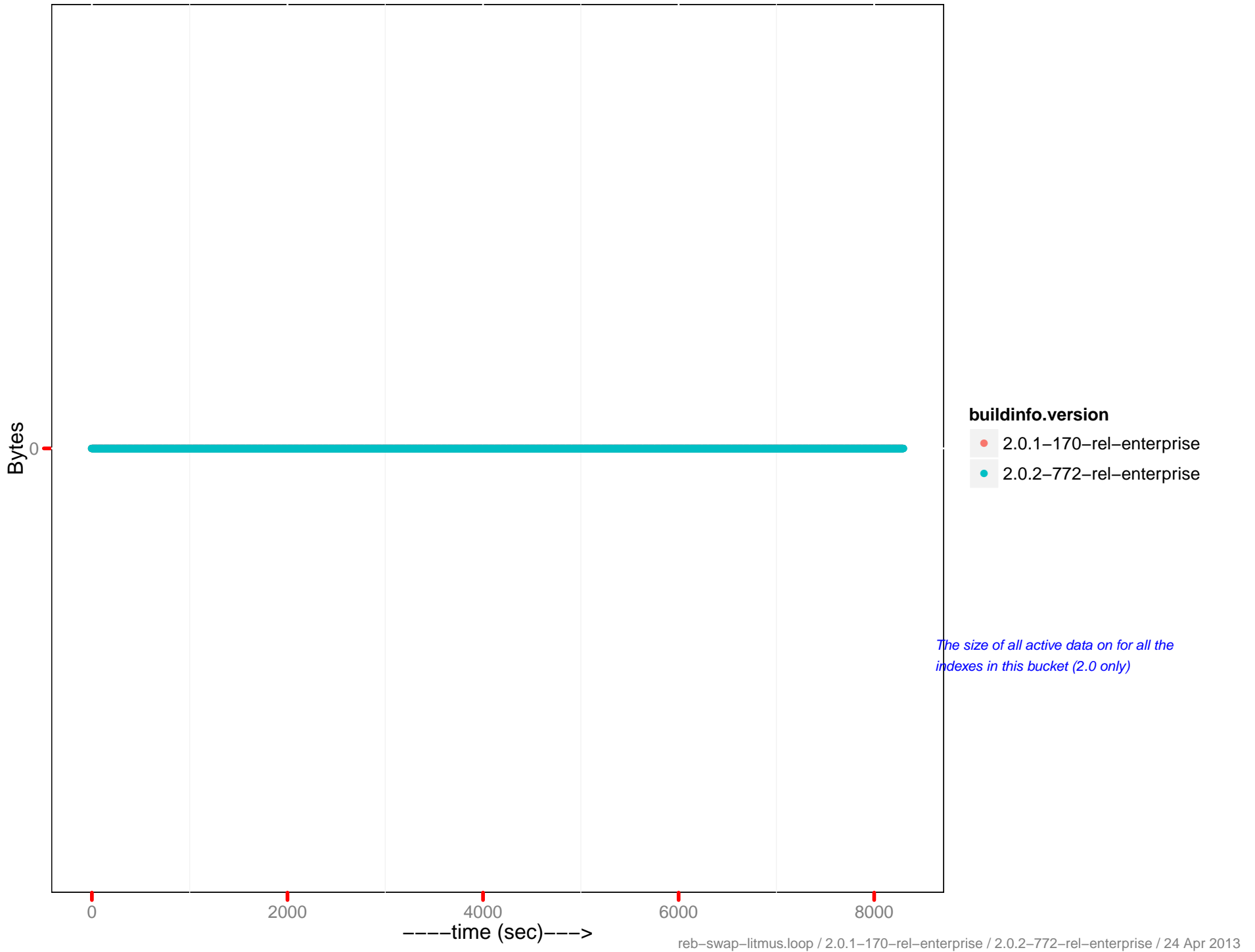
# Docs disk size



# Docs actual disk size

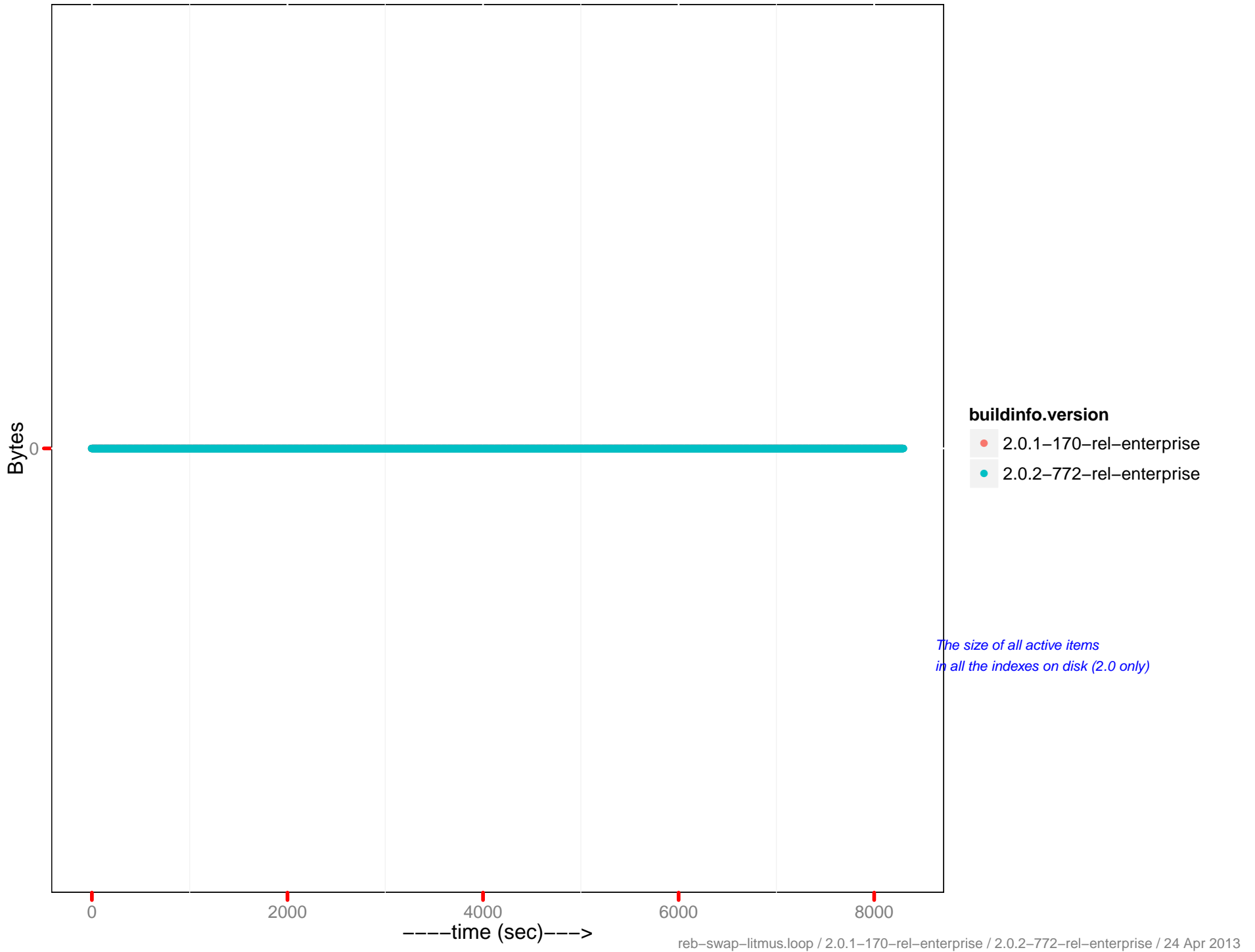


# Views data size

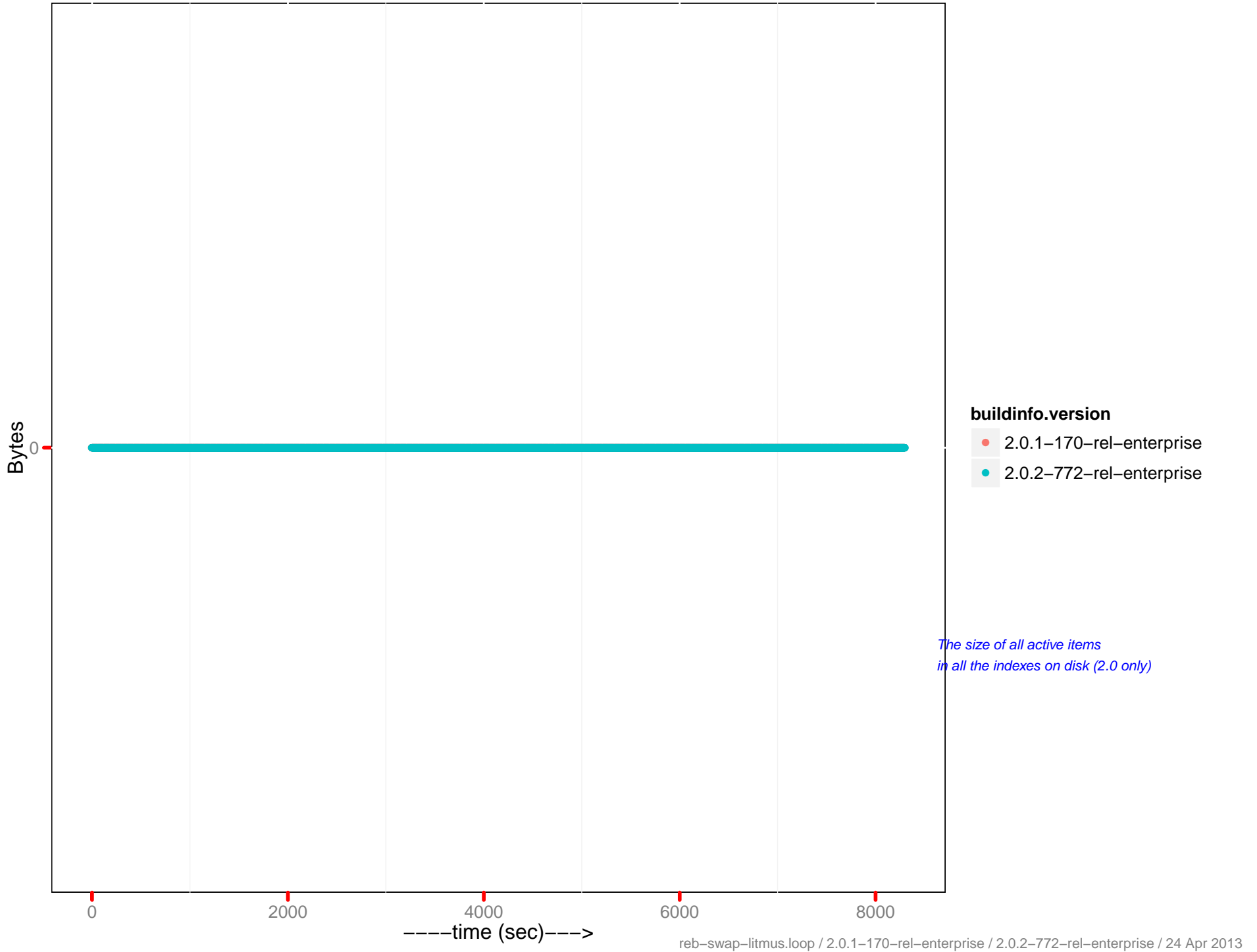




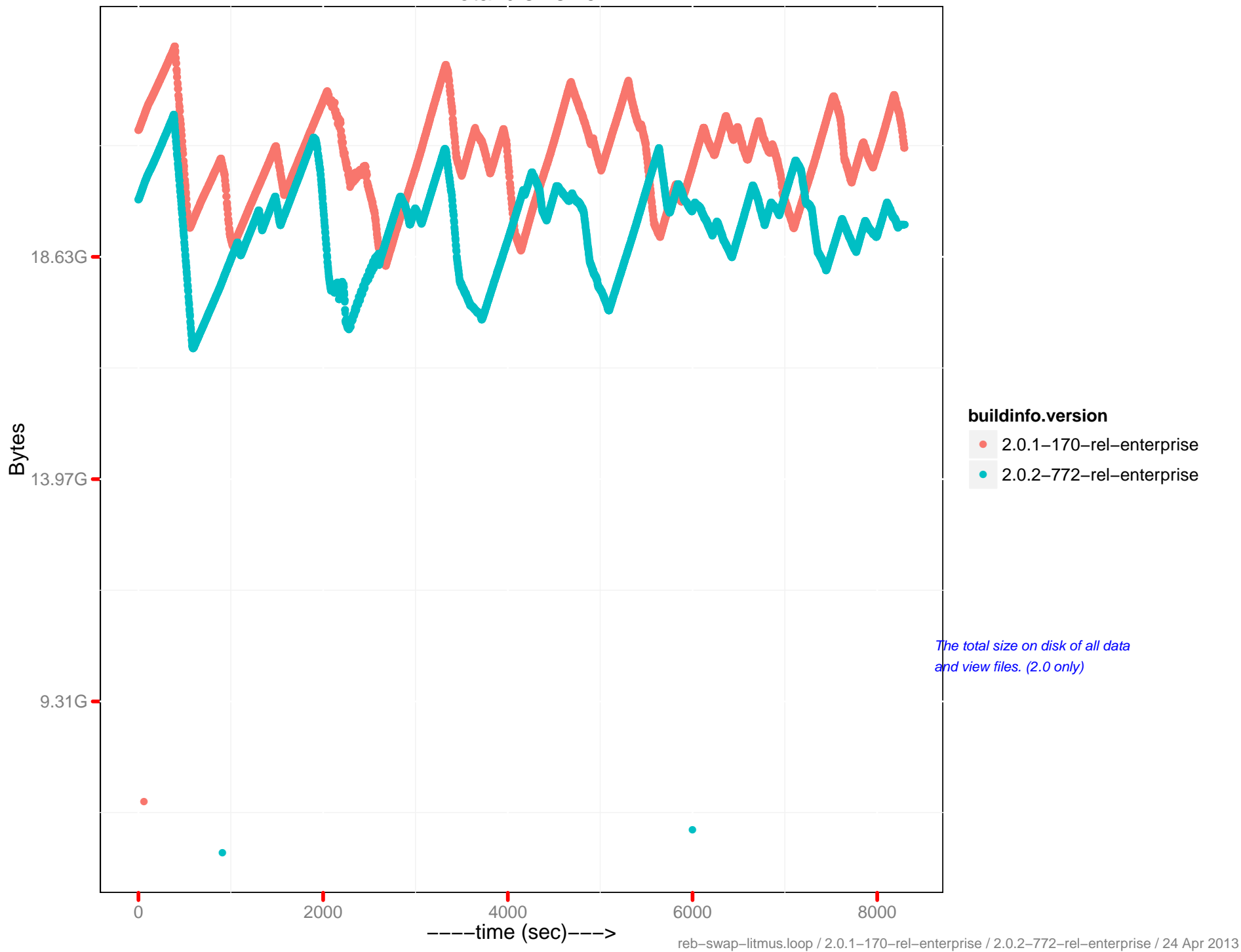
# Views disk size



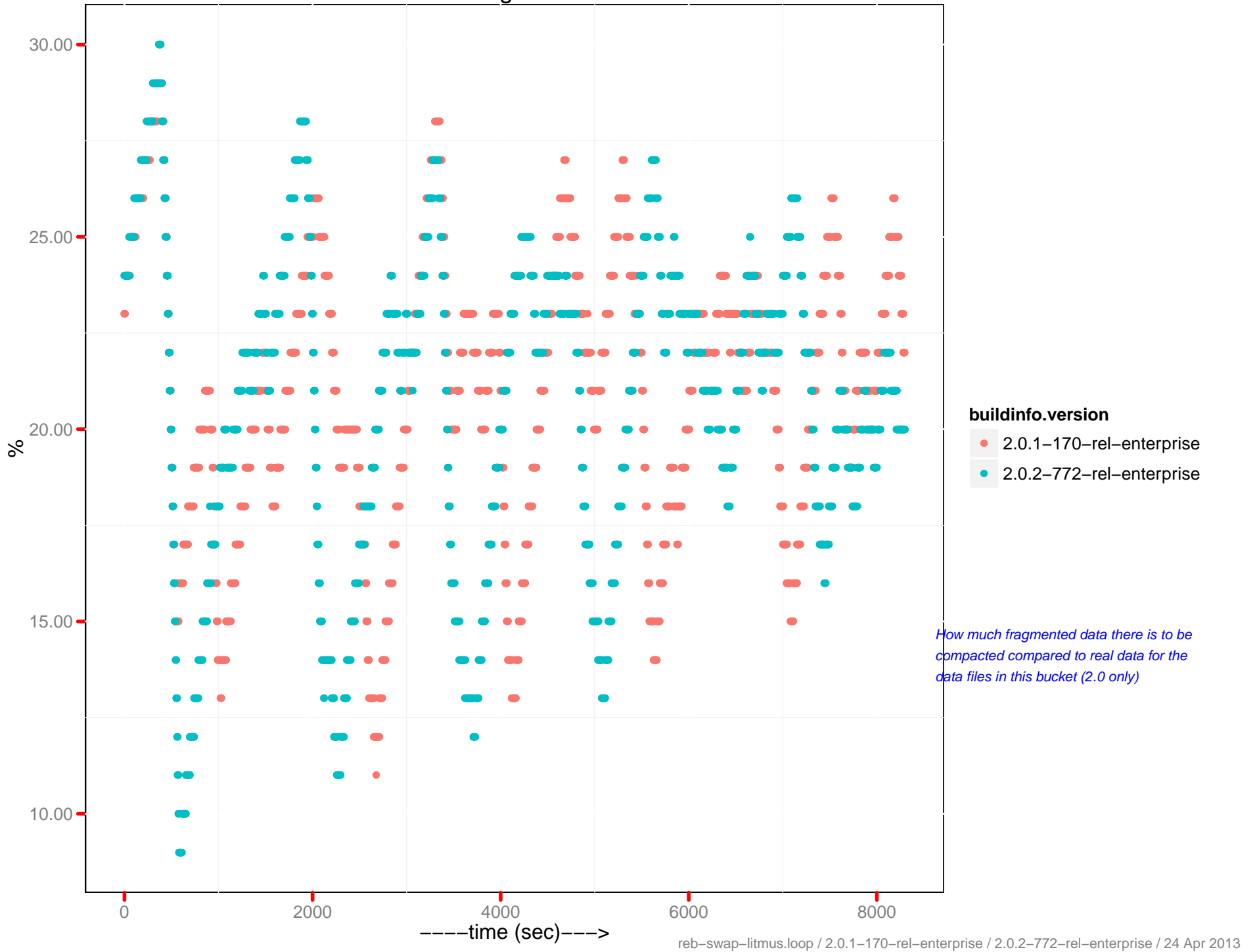
# Views actual disk size



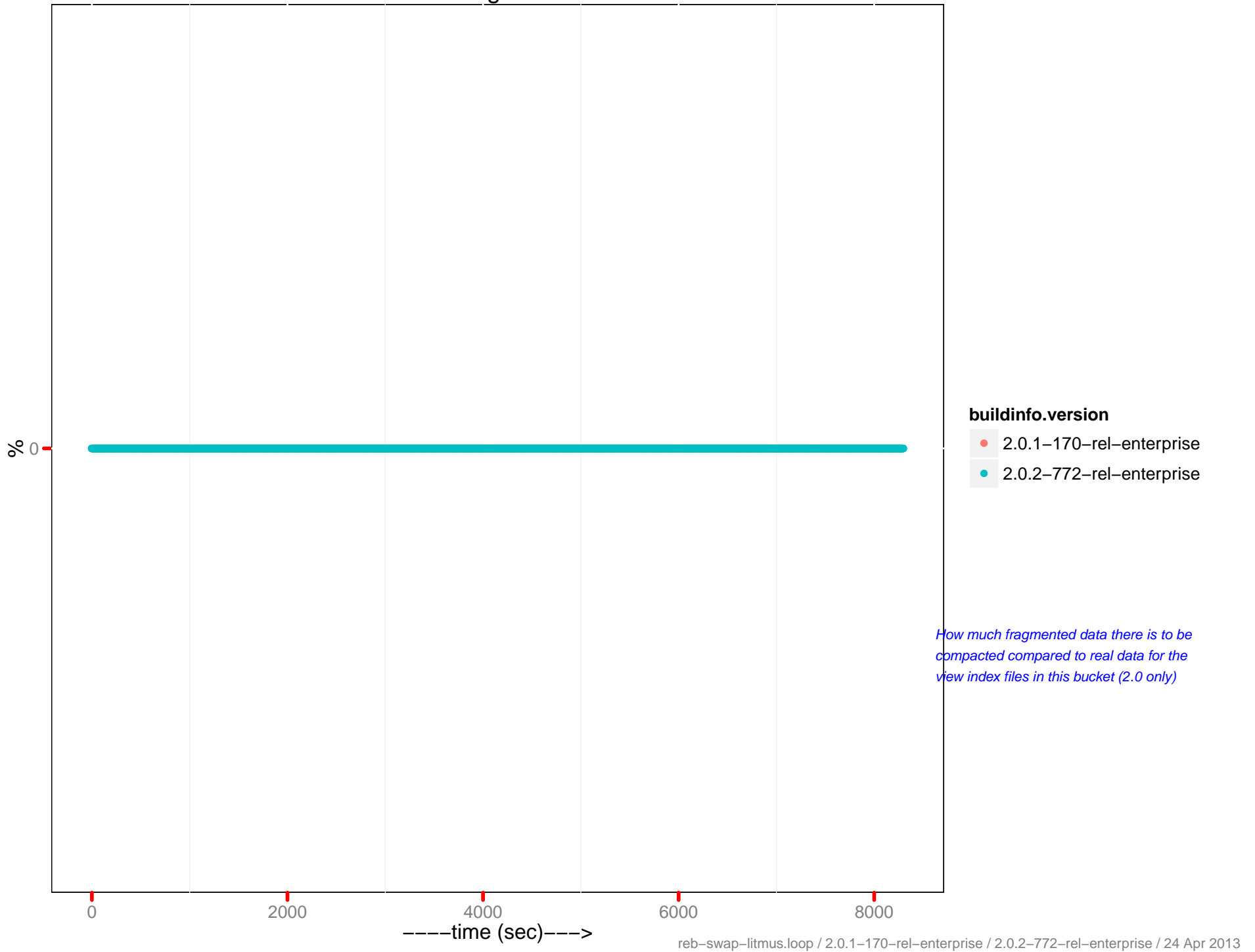
# Total disk size



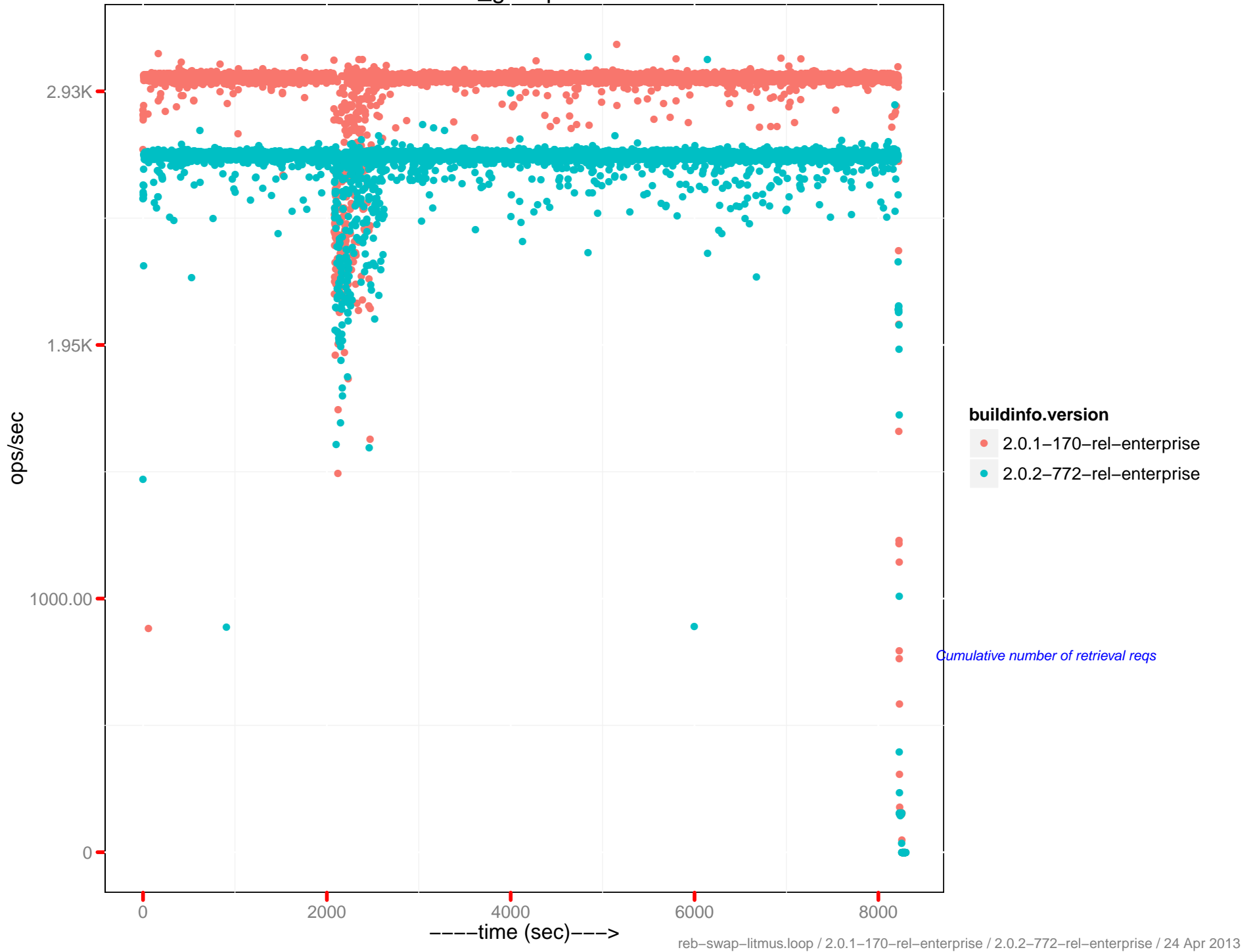
# Docs fragmentation



# Views fragmentation



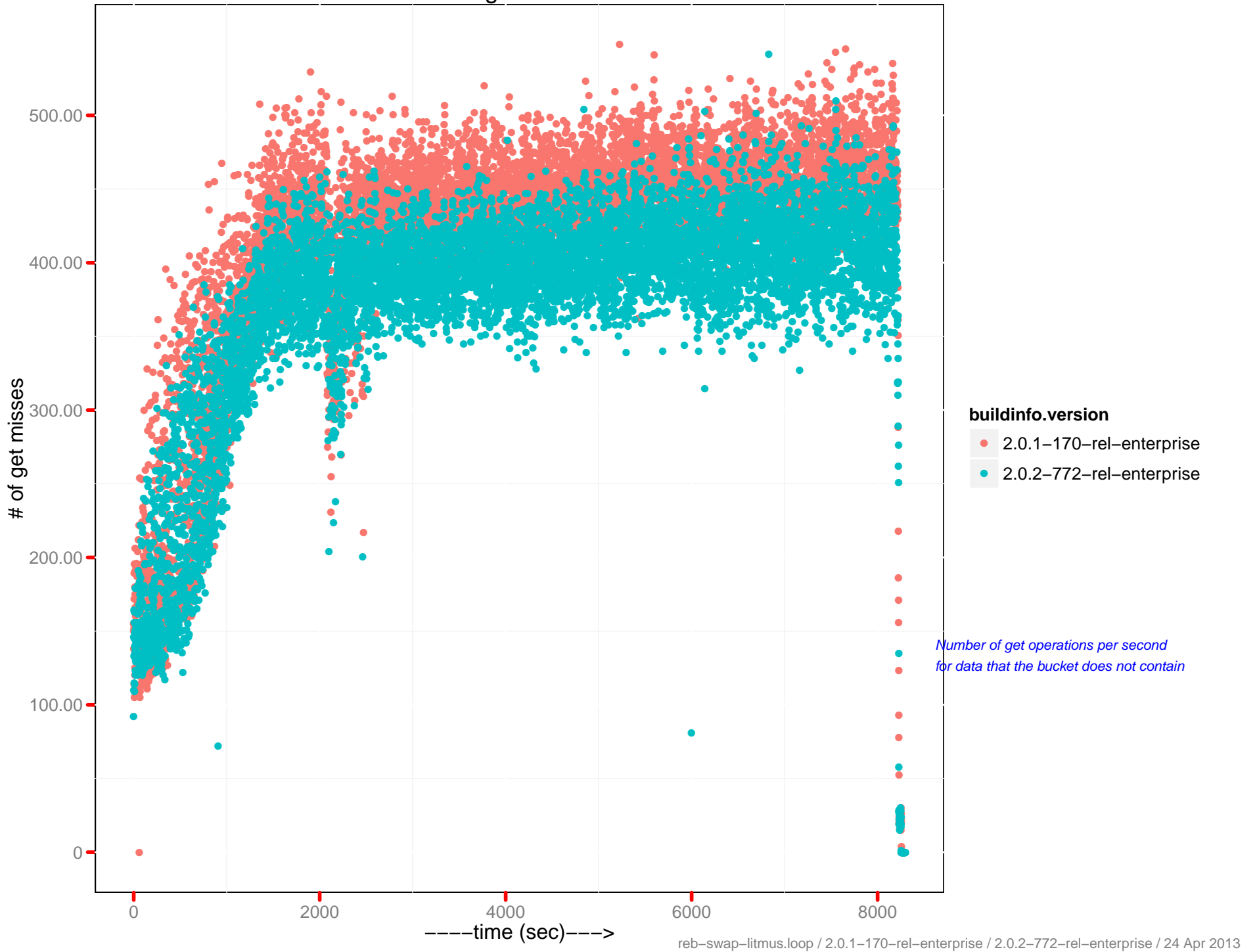
# cmd\_get ops/sec



# cmd\_set ops/sec



# # of get misses

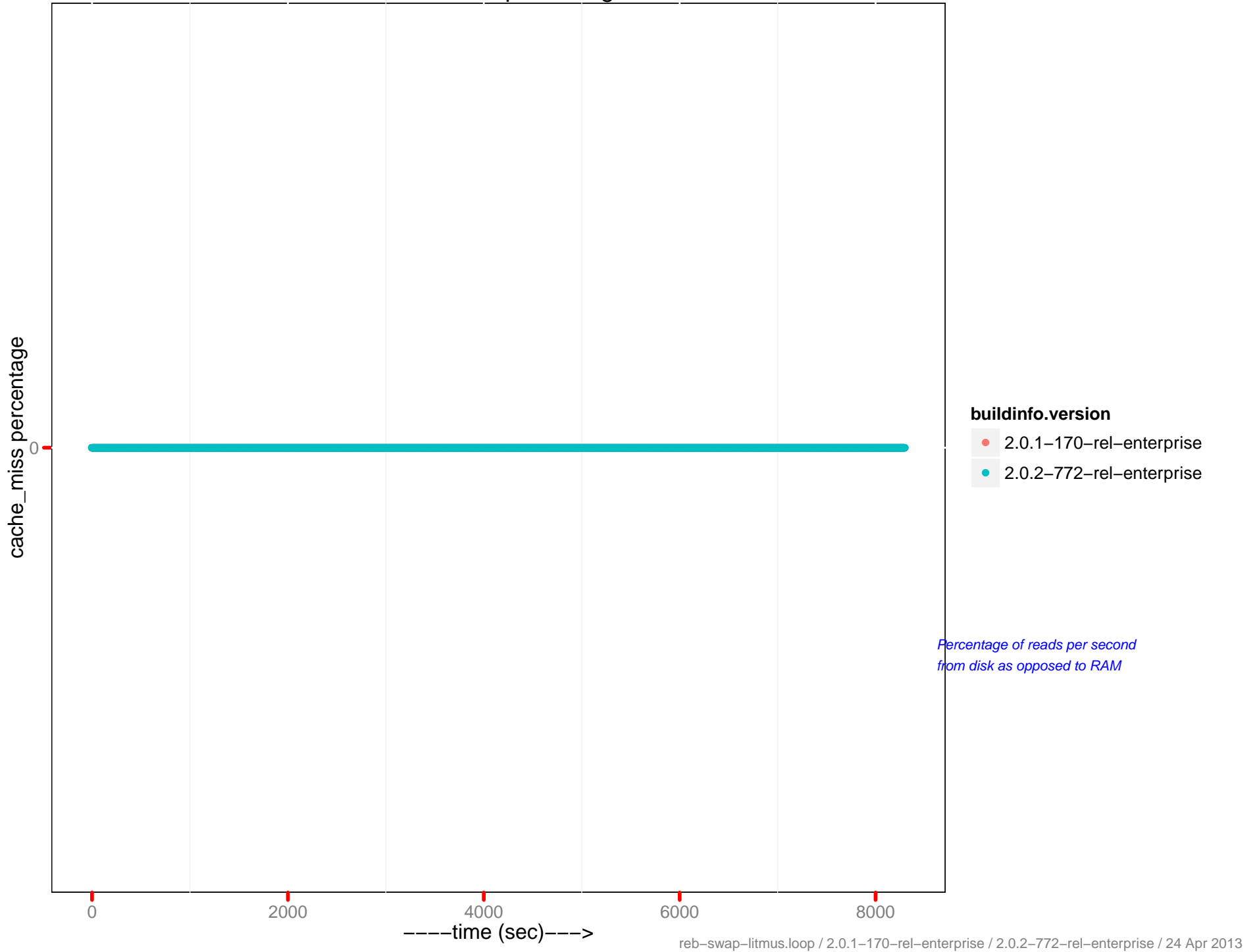




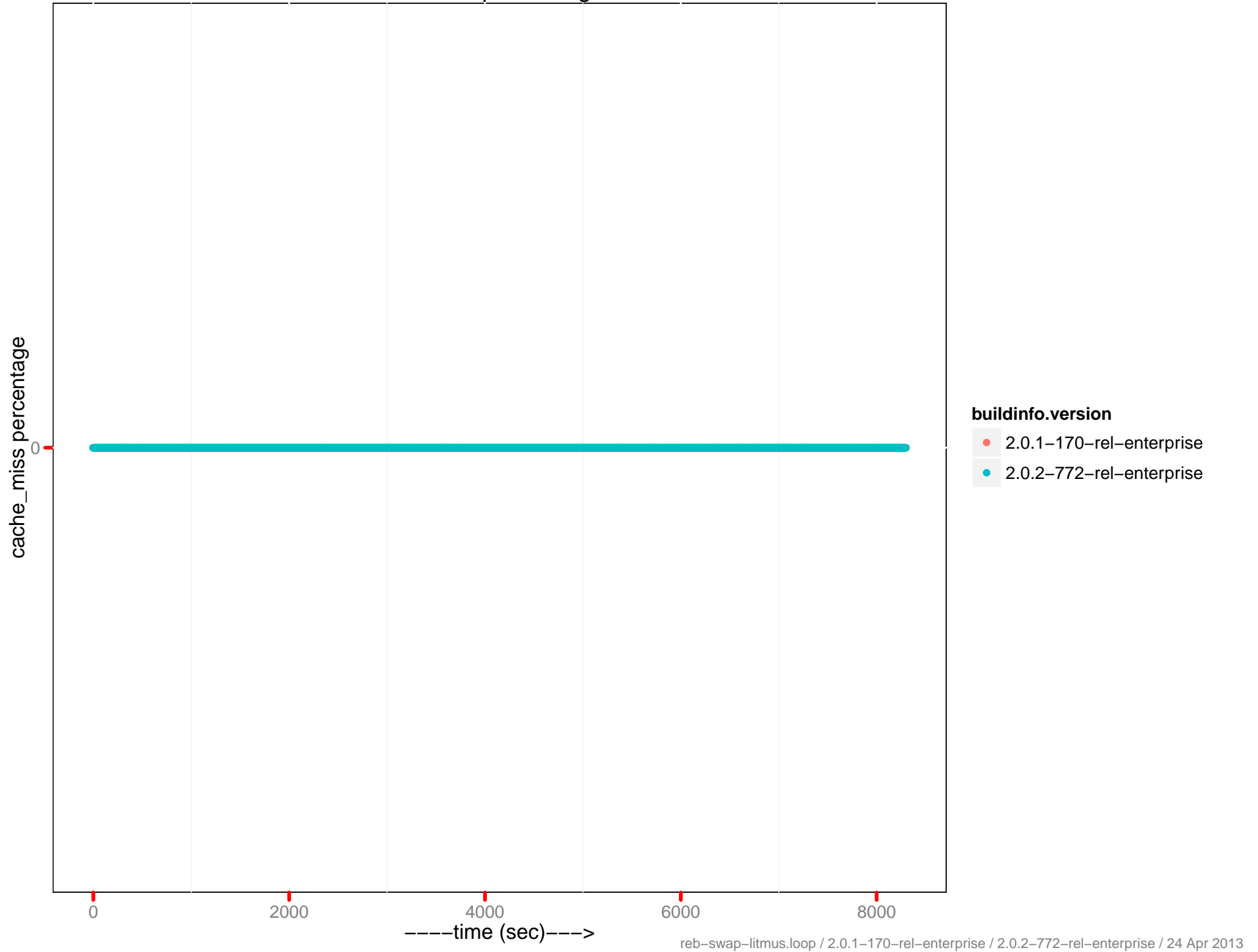
# # of get hits



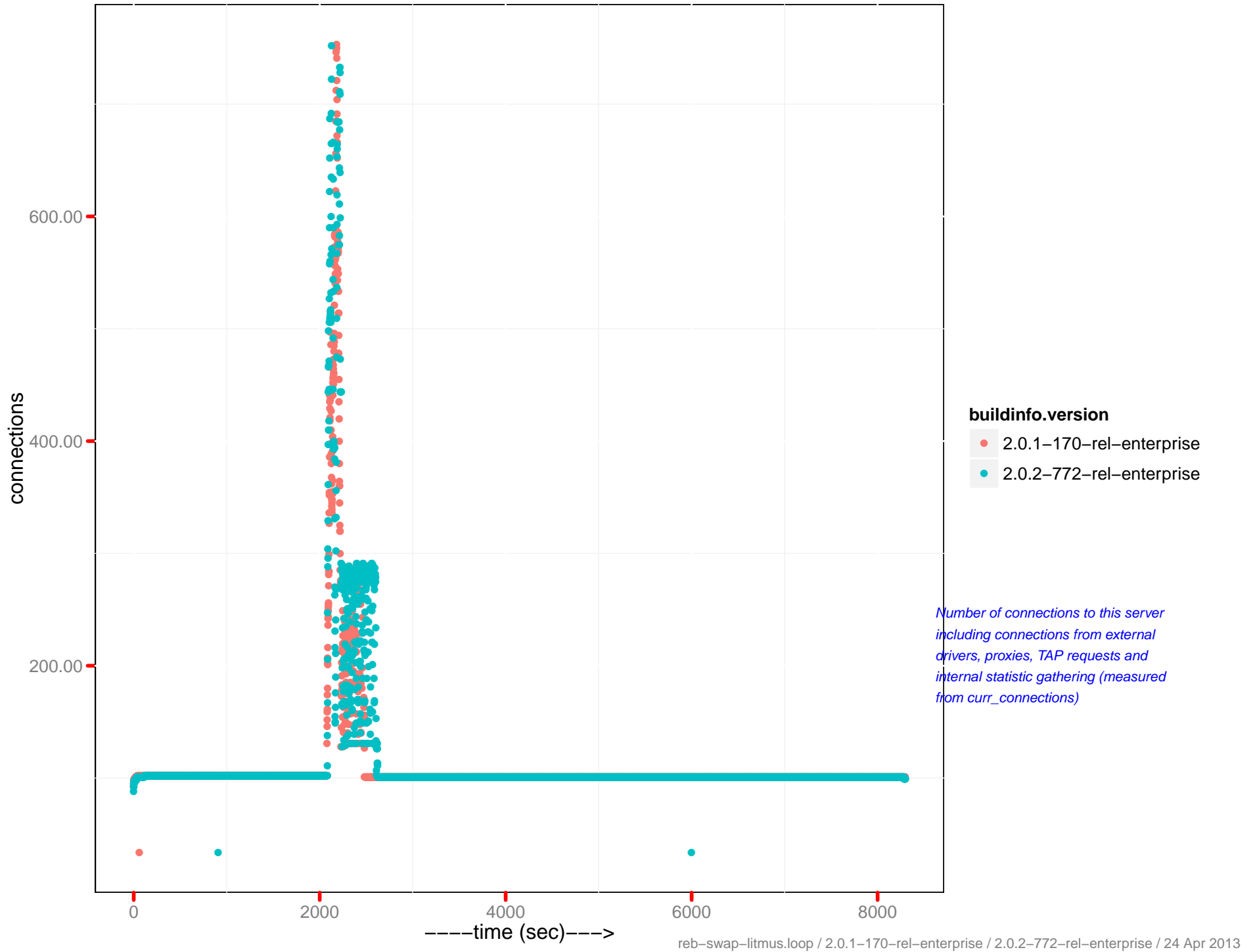
# cache\_miss percentage



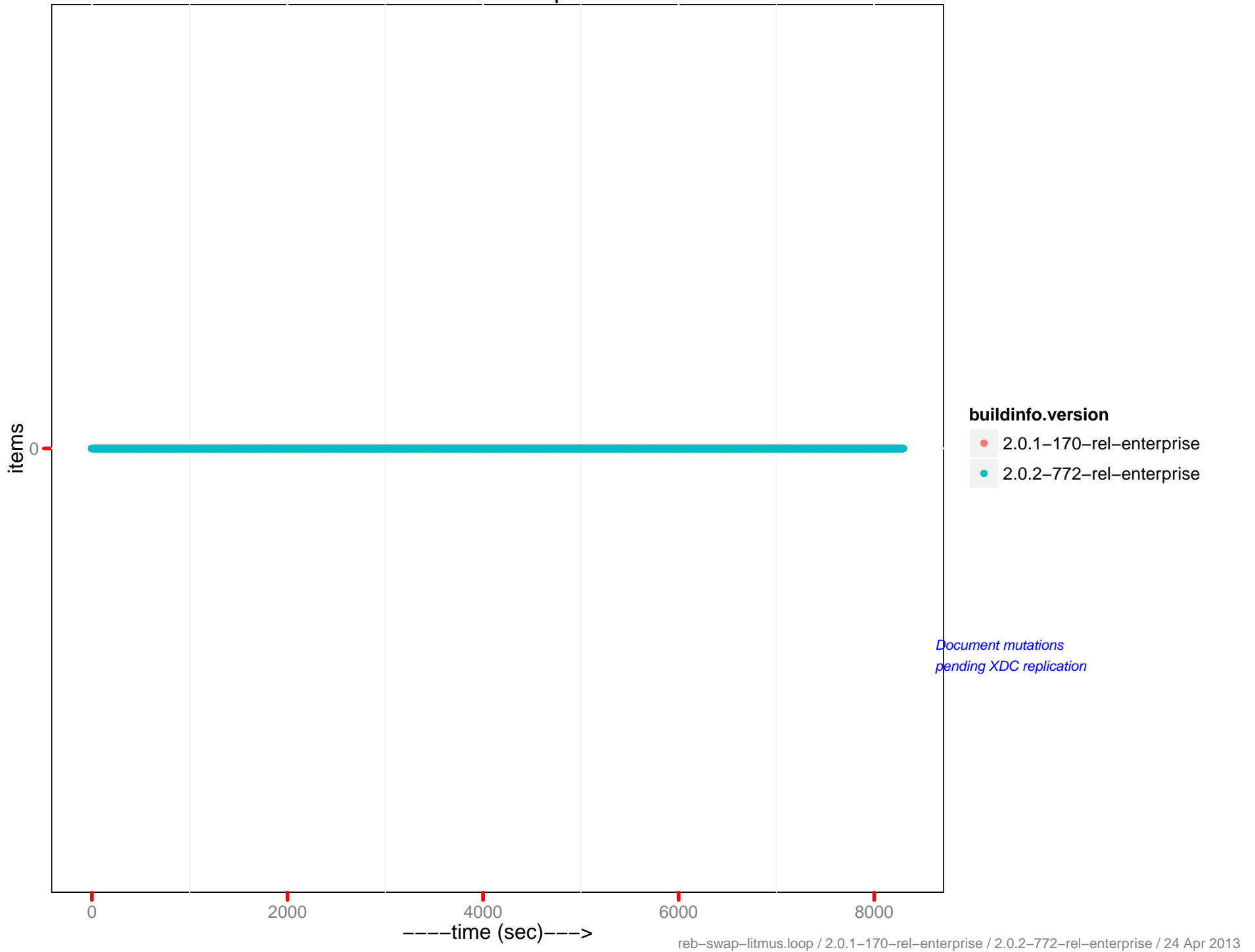
# cache\_miss percentage 0-5



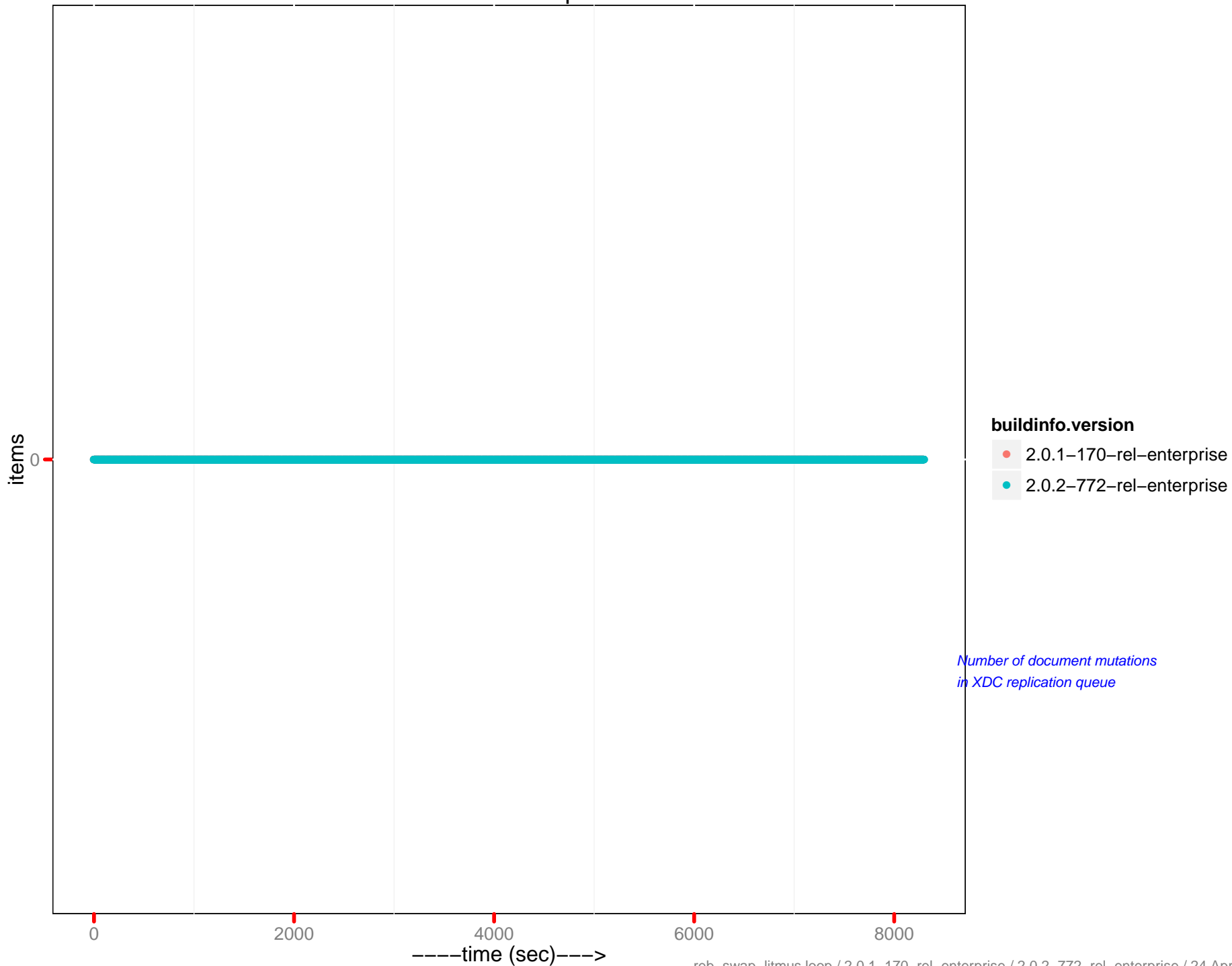
# Number of connections



# XDCR docs to replicate

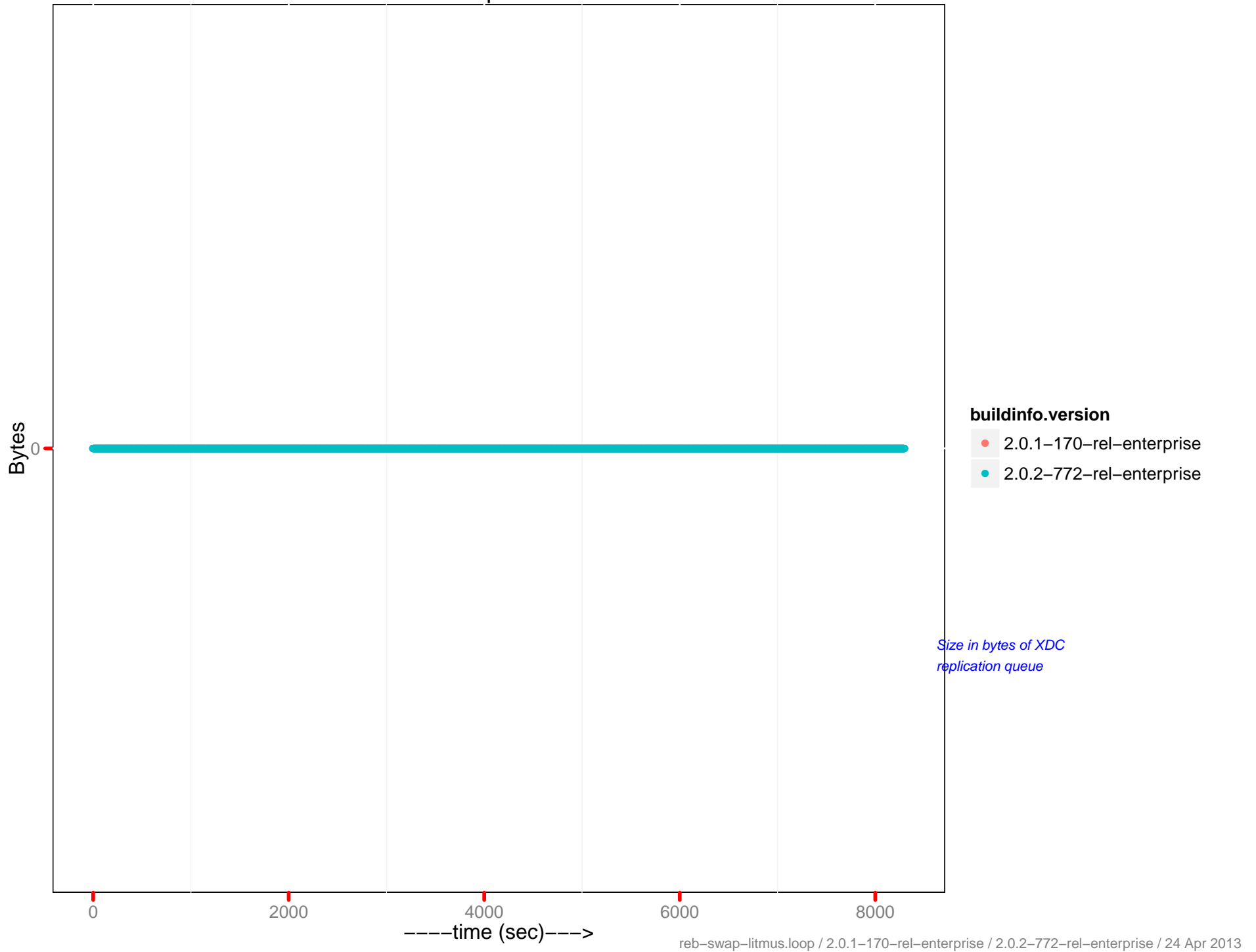


# XDCR docs in queue

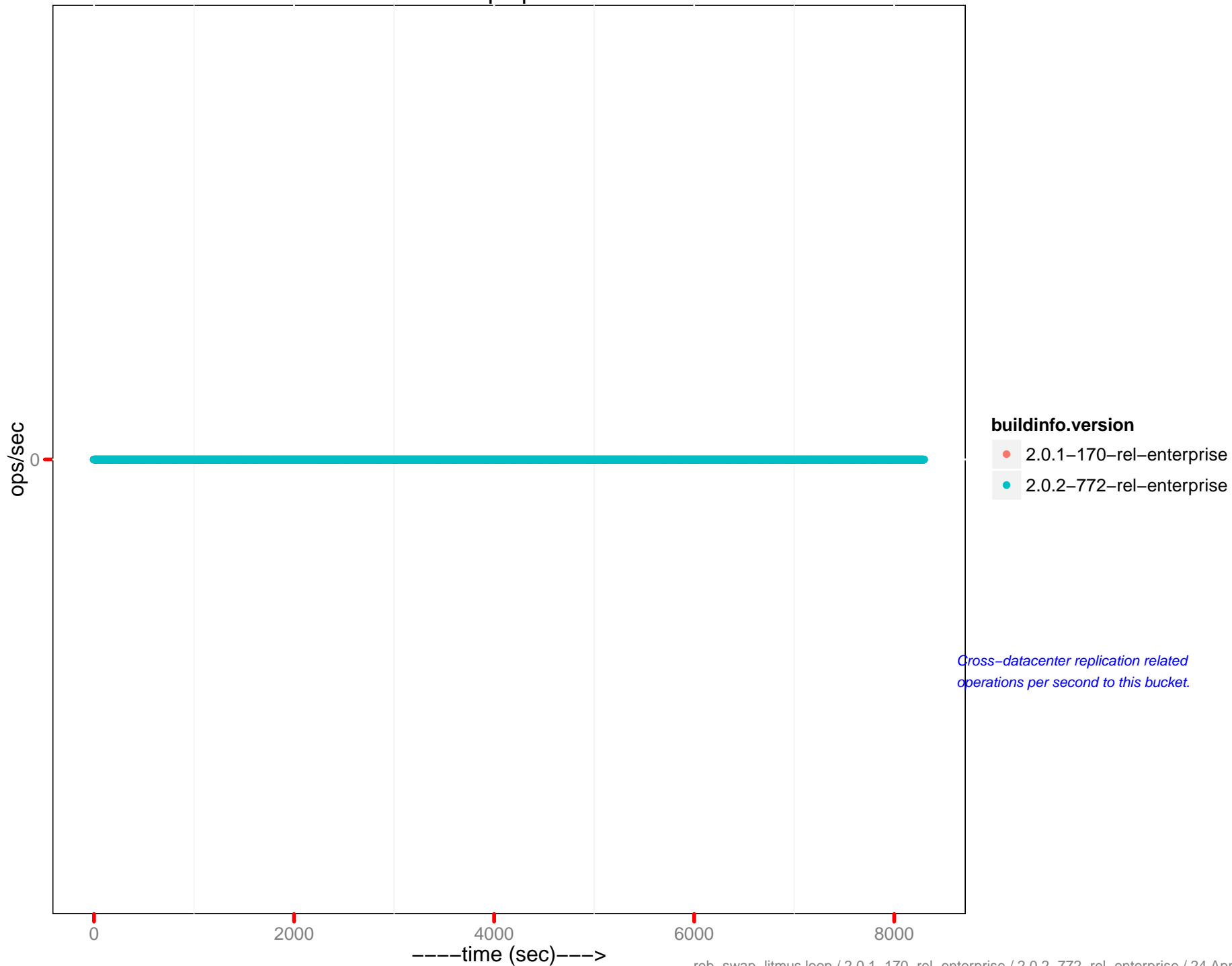


*Number of document mutations  
in XDC replication queue*

# XDCR queue size

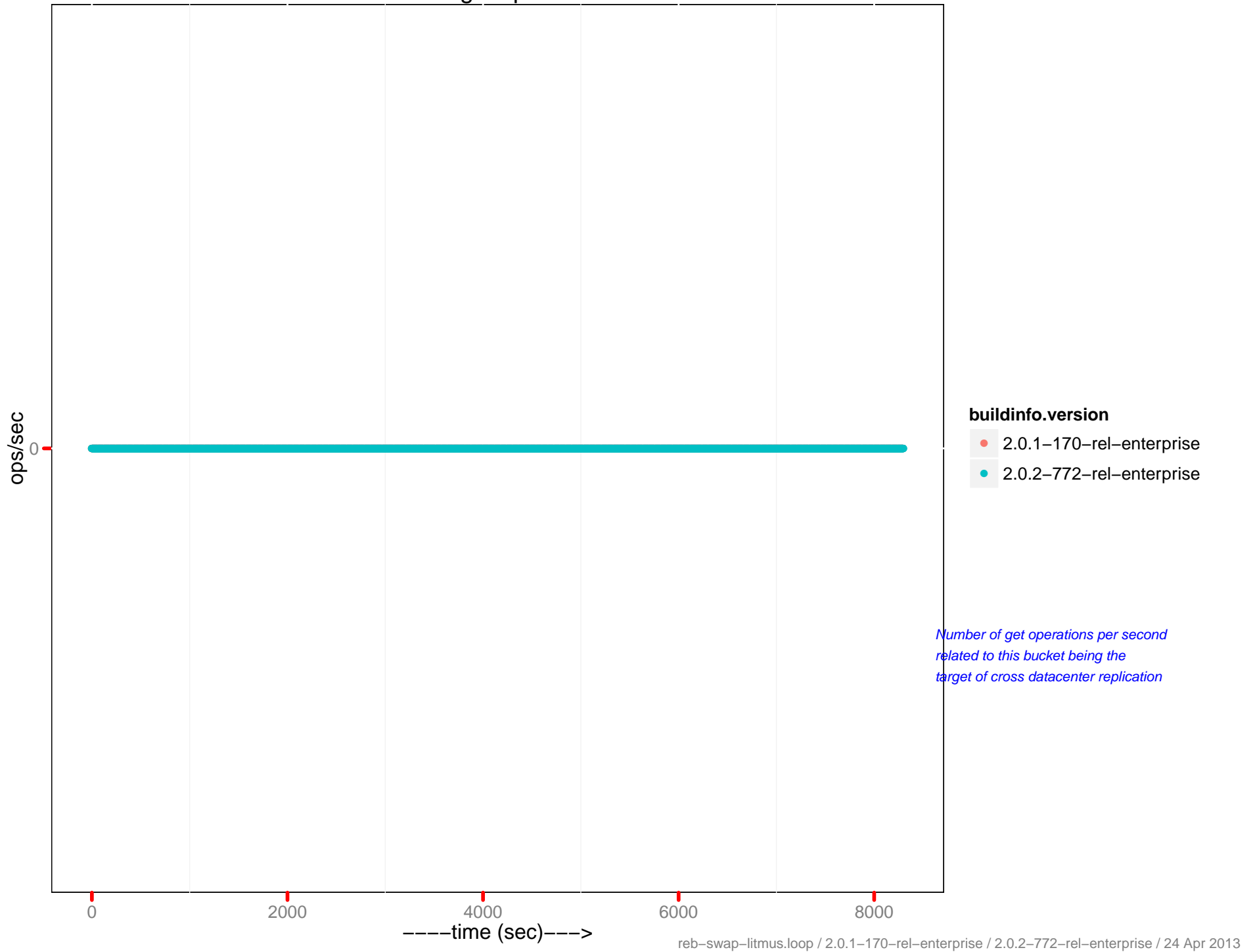


# XDC ops per sec

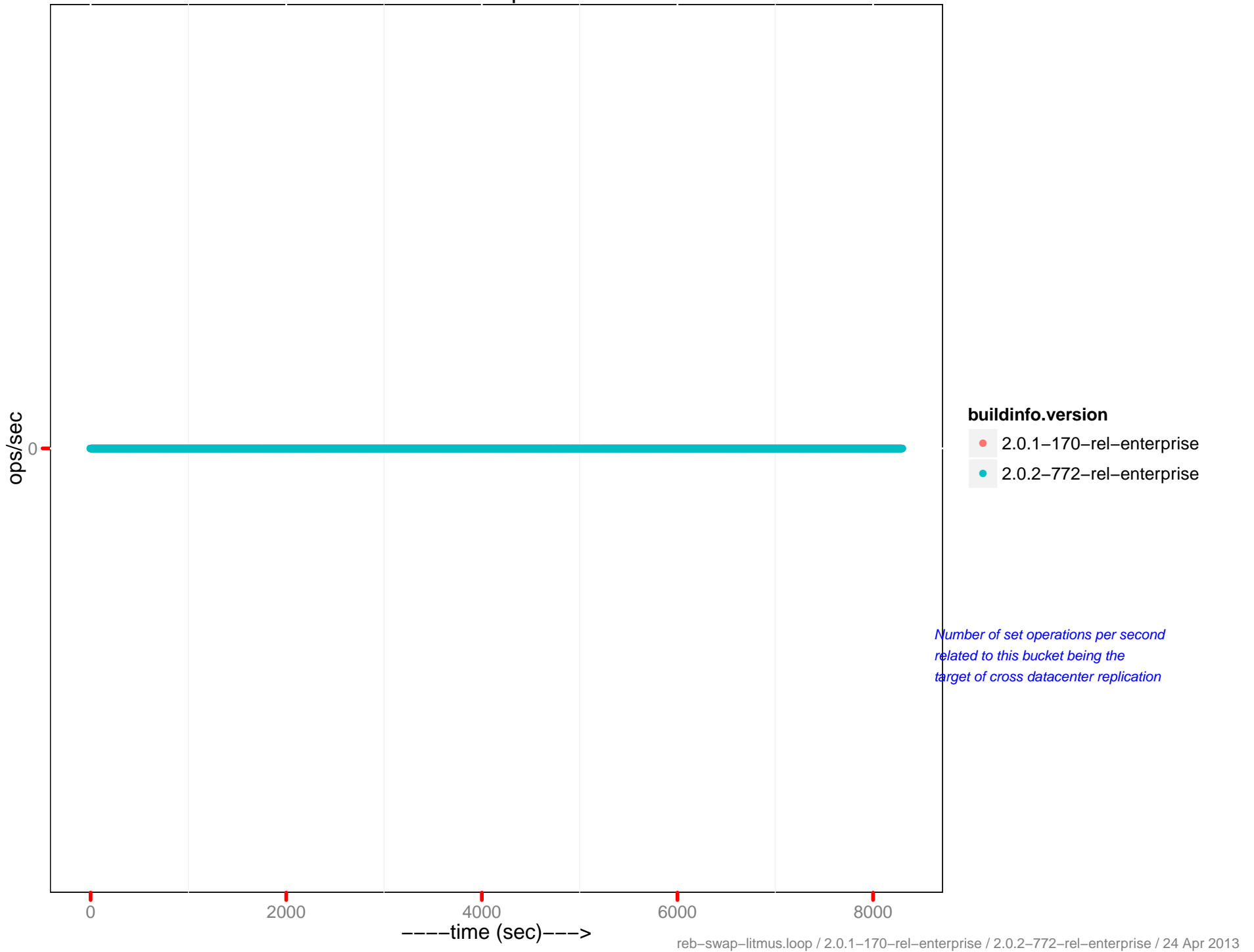




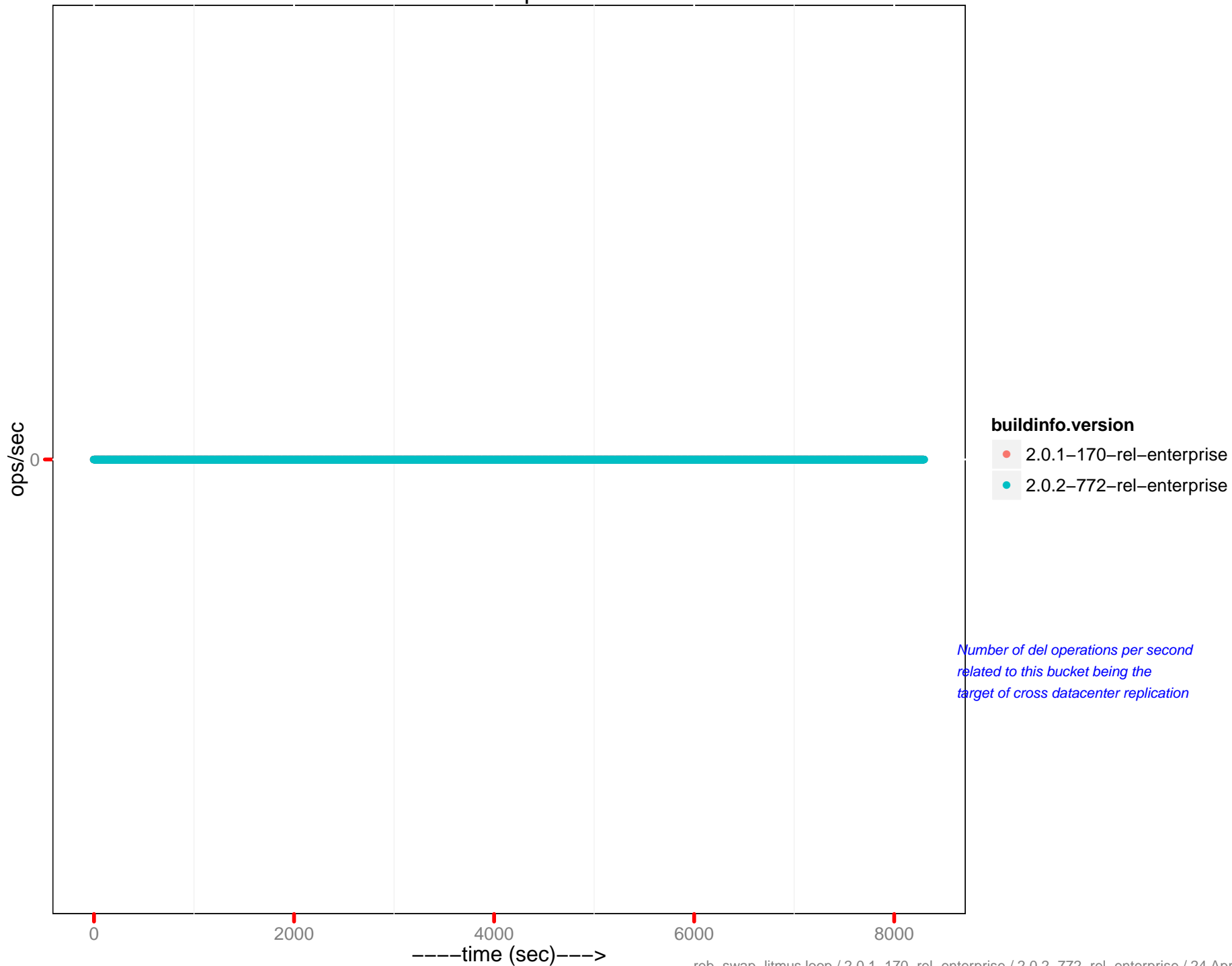
# XDC gets per sec



# XDC sets per sec

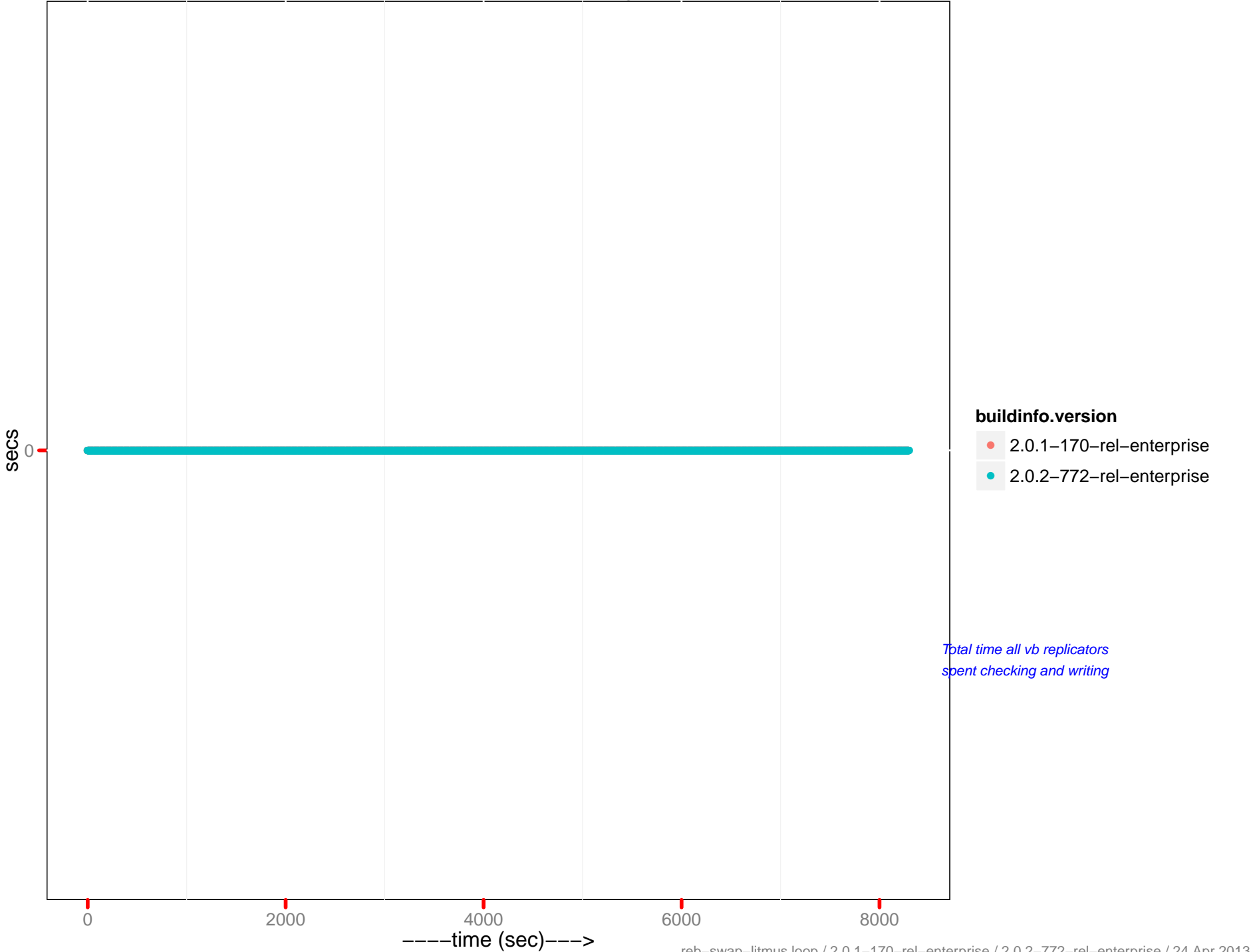


# XDC dels per sec

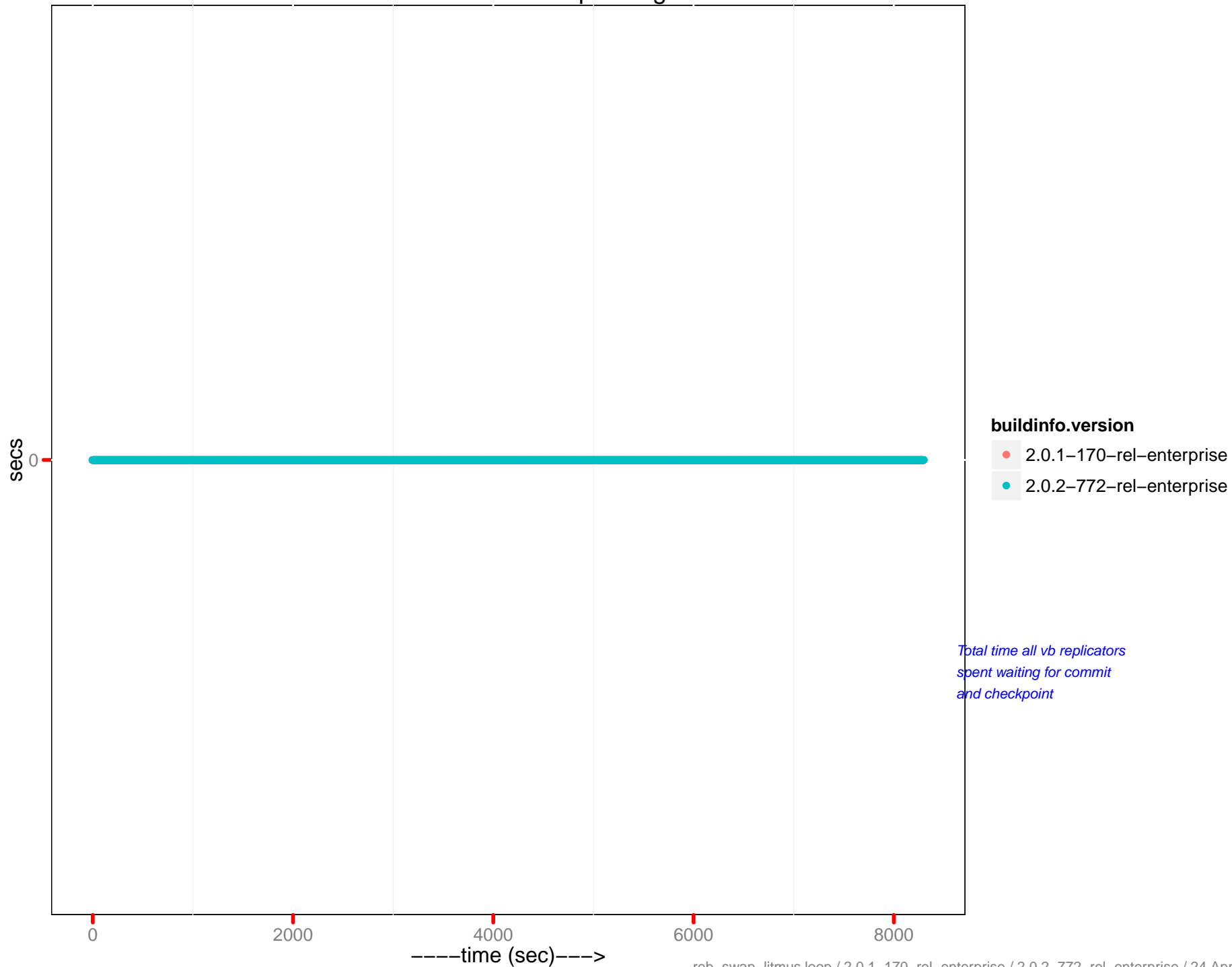


*Number of del operations per second related to this bucket being the target of cross datacenter replication*

# XDCR secs in replicating

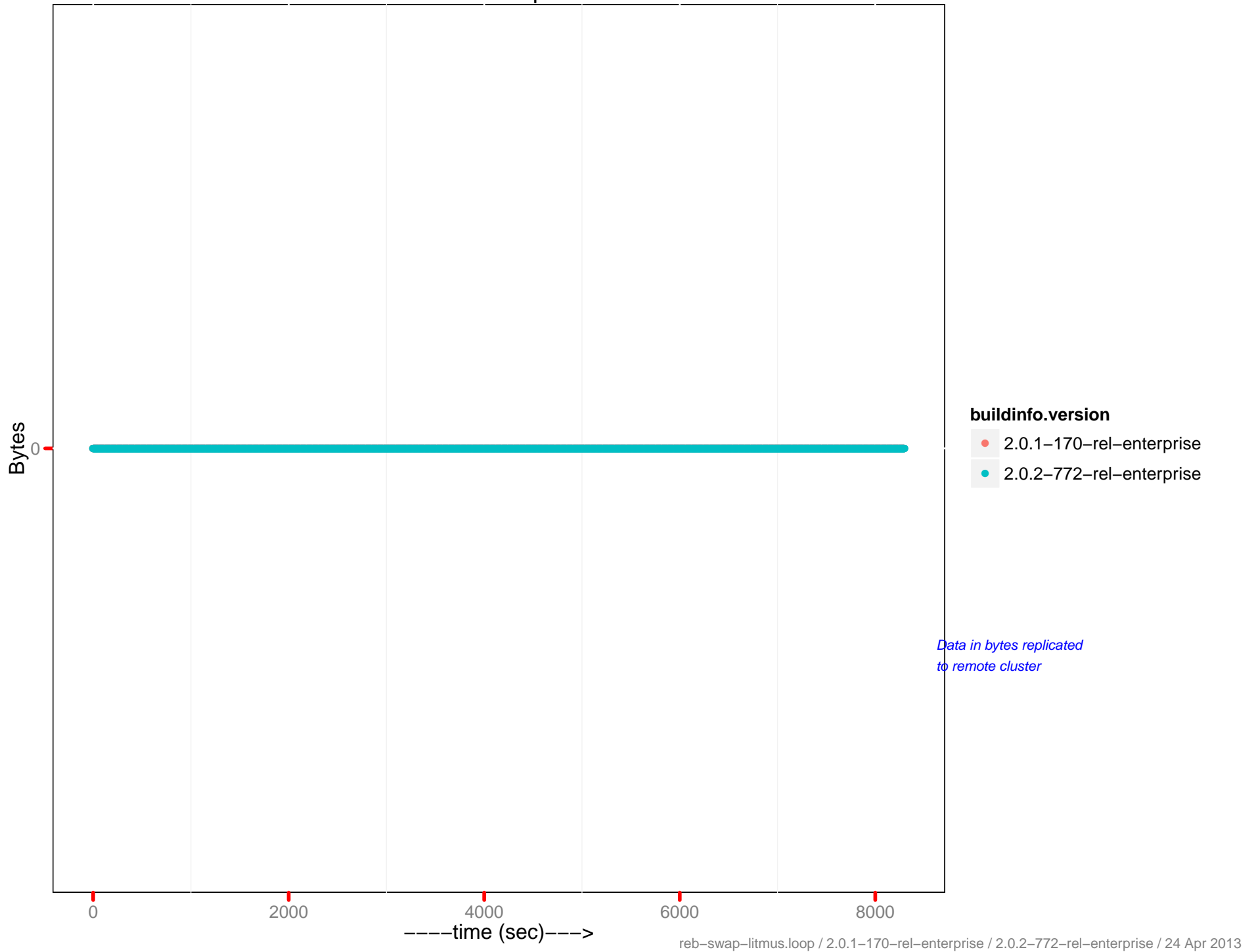


# XDCR secs in checkpointing

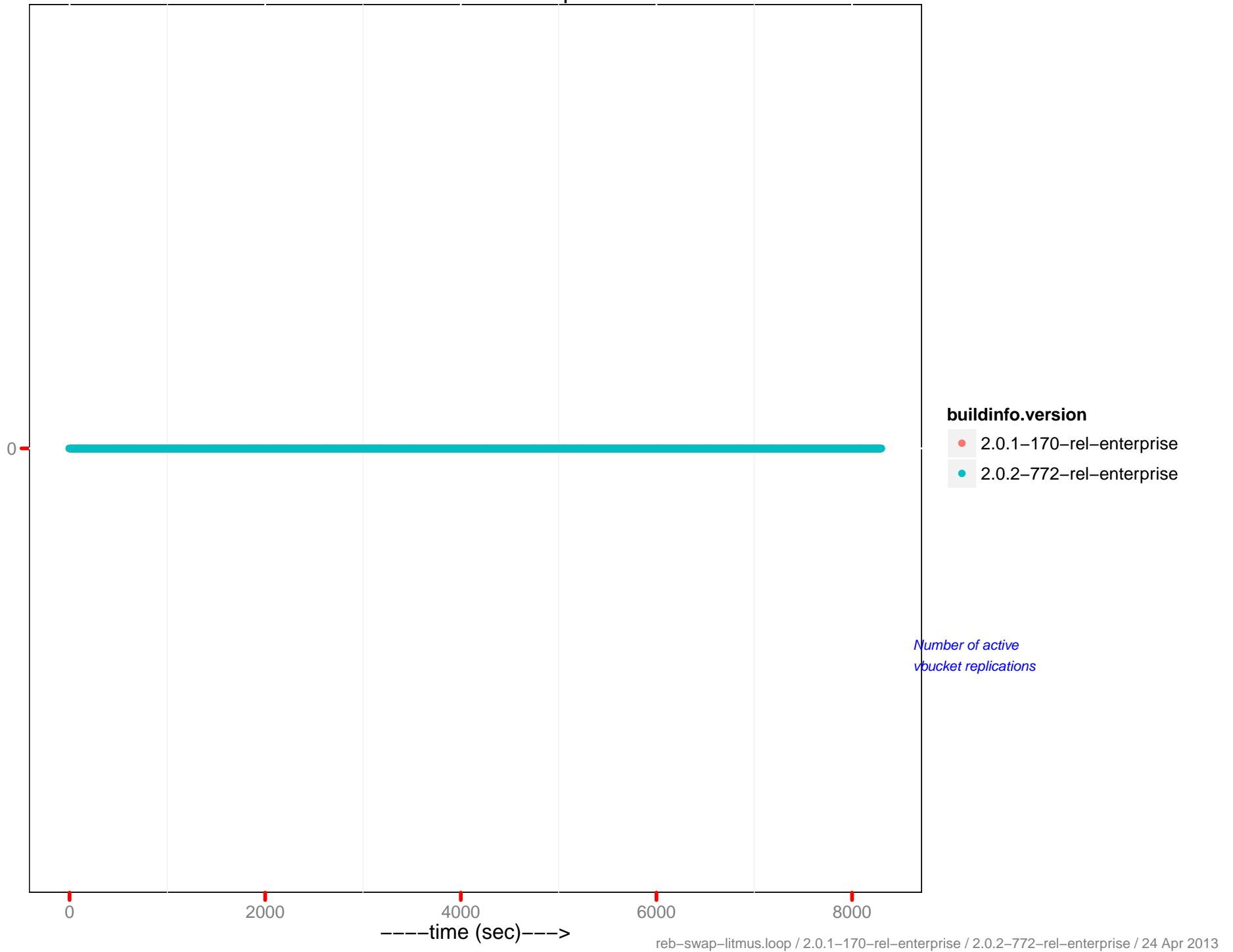


*Total time all vb replicators  
spent waiting for commit  
and checkpoint*

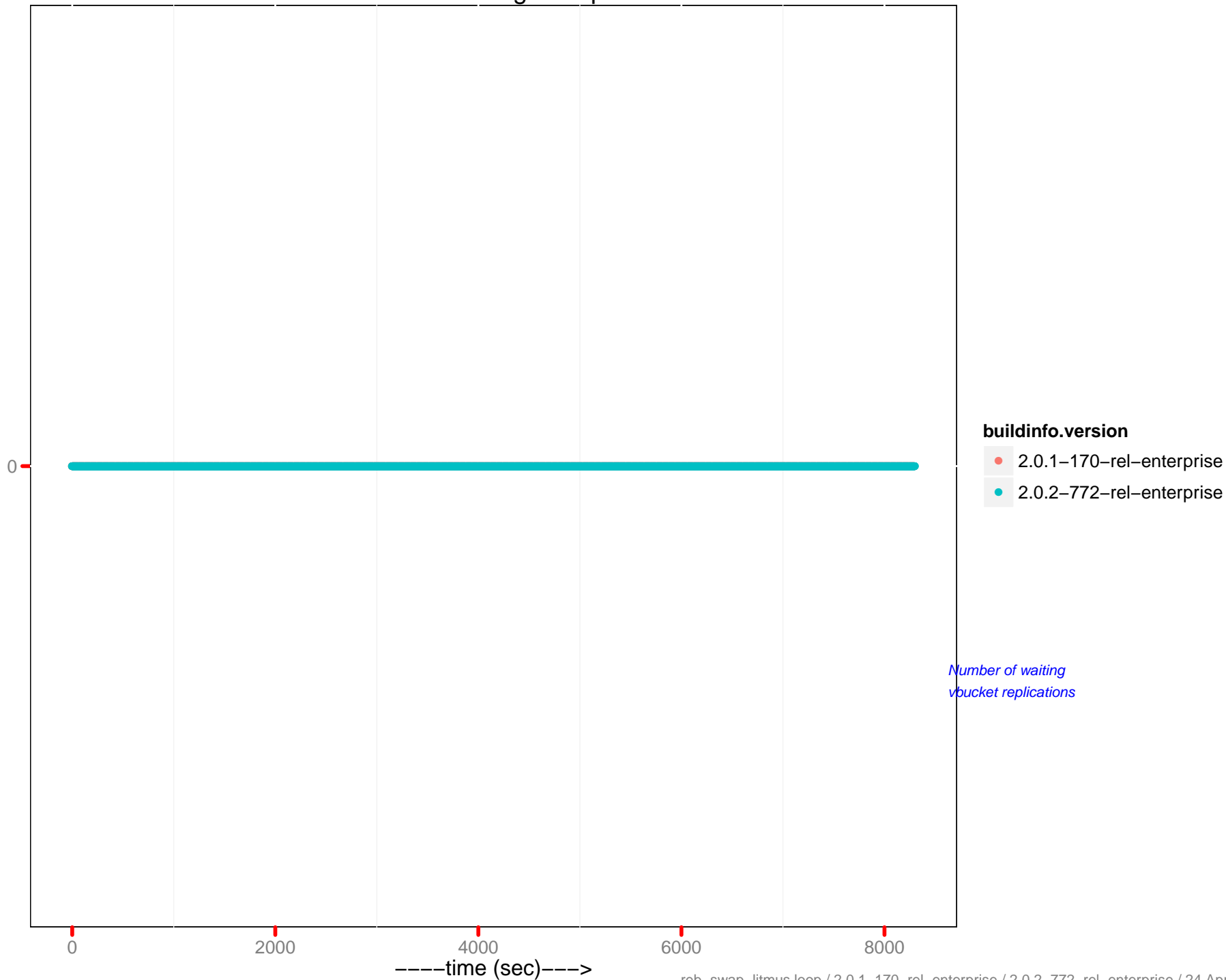
# XDCR data replicated



# XDCR active vb reps



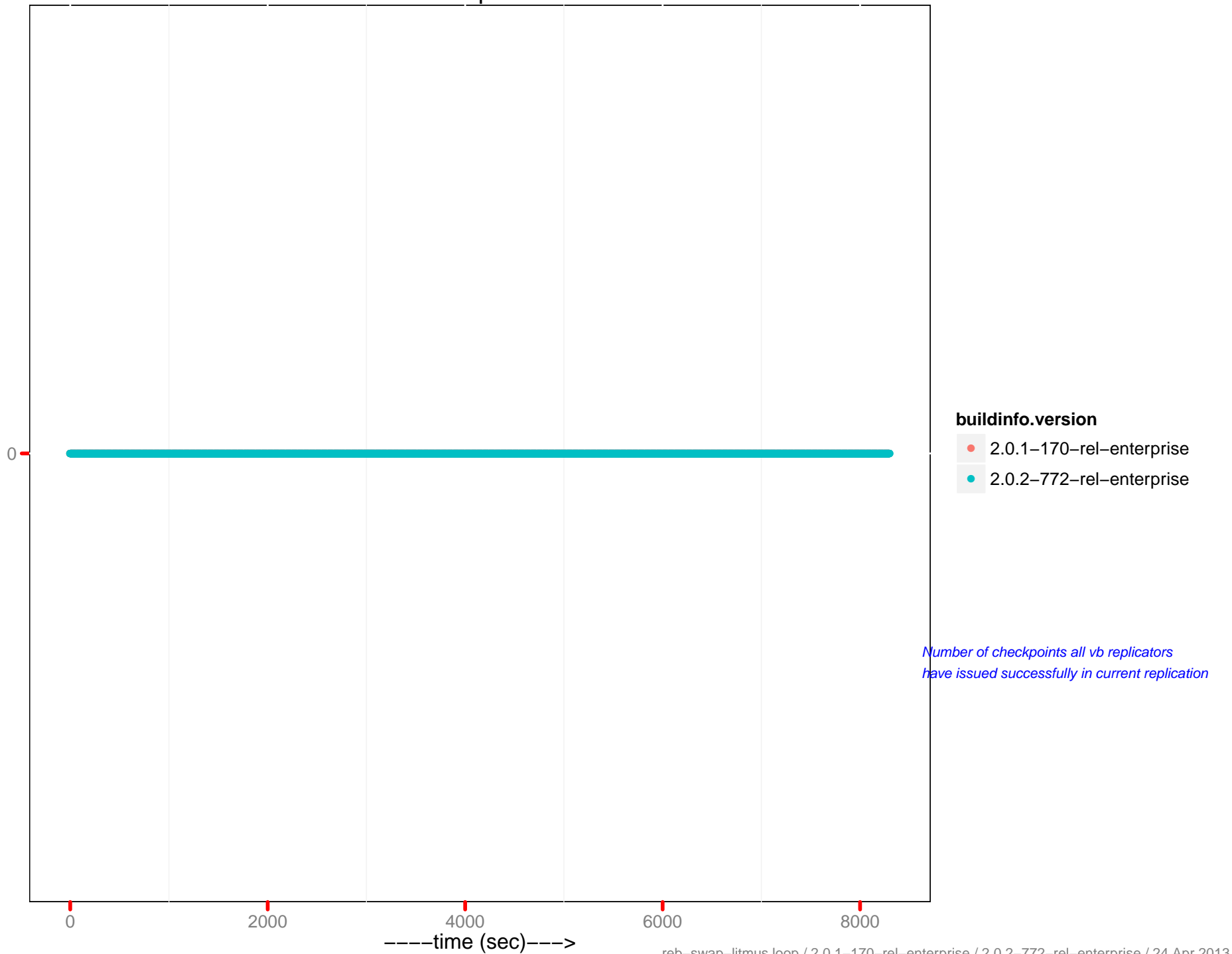
# XDCR waiting vb reps



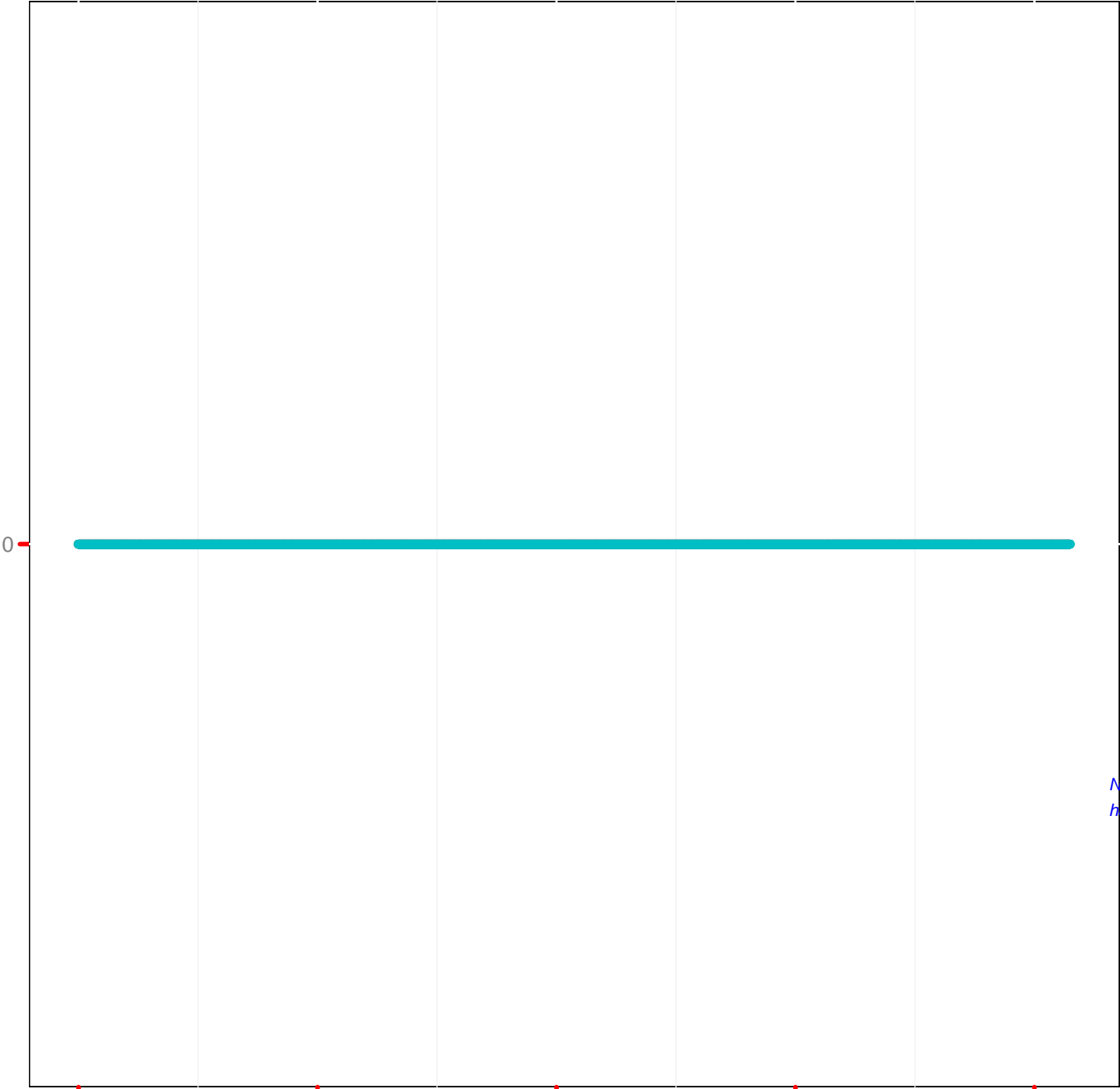
Number of waiting  
vbucket replications



# XDCR checkpoints issued



# XDCR checkpoints failed

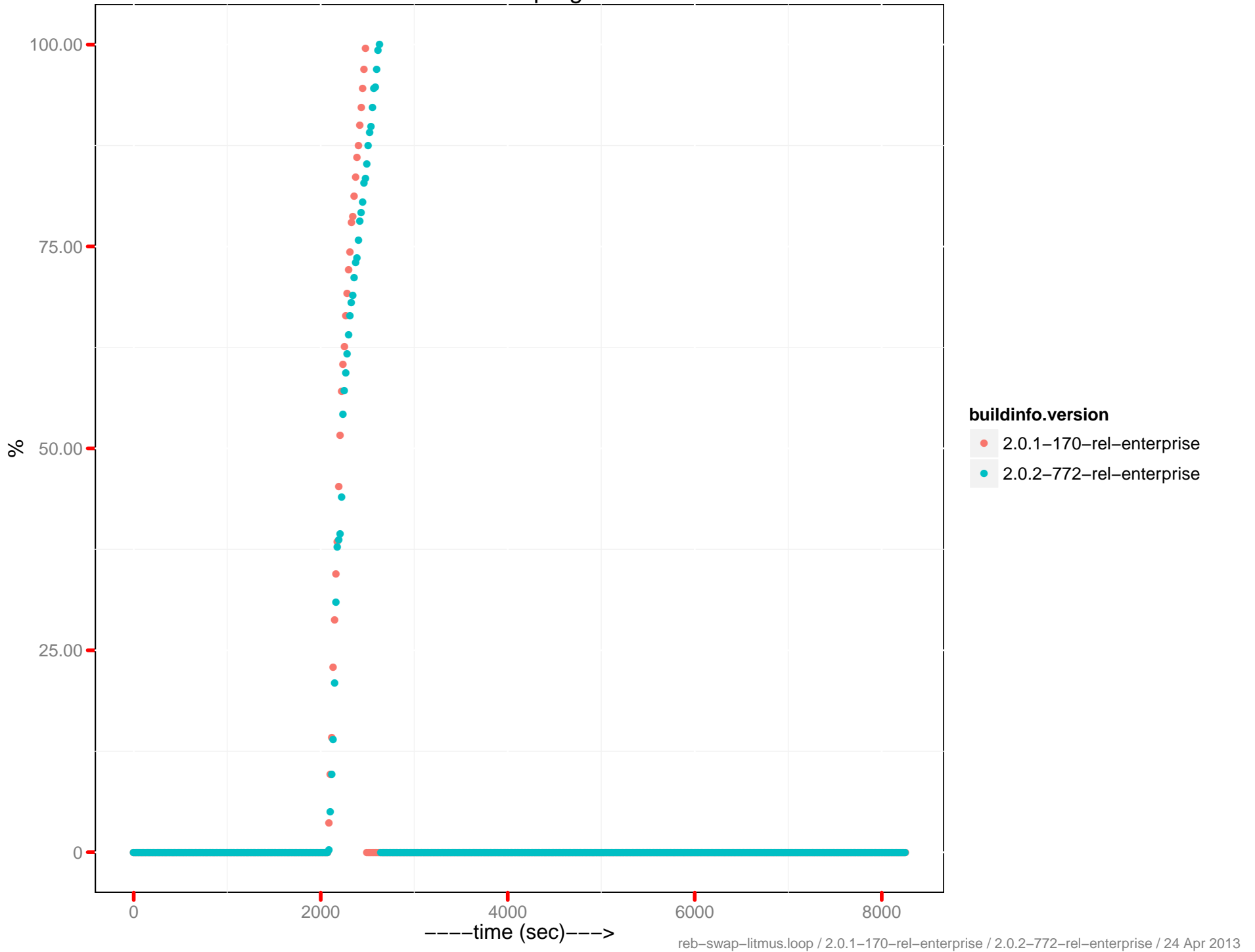


- buildinfo.version**
- 2.0.1-170-rel-enterprise
  - 2.0.2-772-rel-enterprise

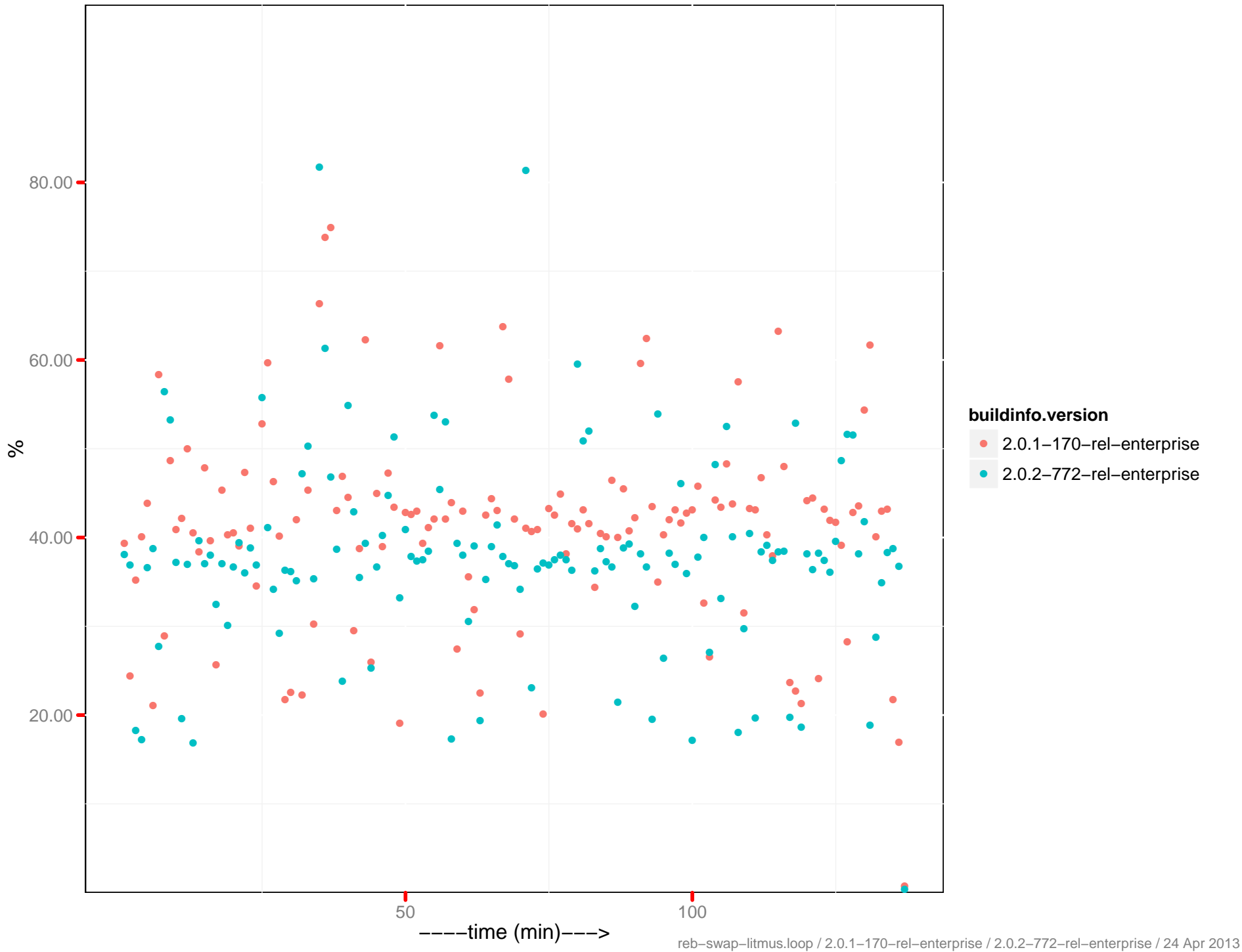
*Number of checkpoints all vb replicators have failed to issue in current replication*

----time (sec)---->

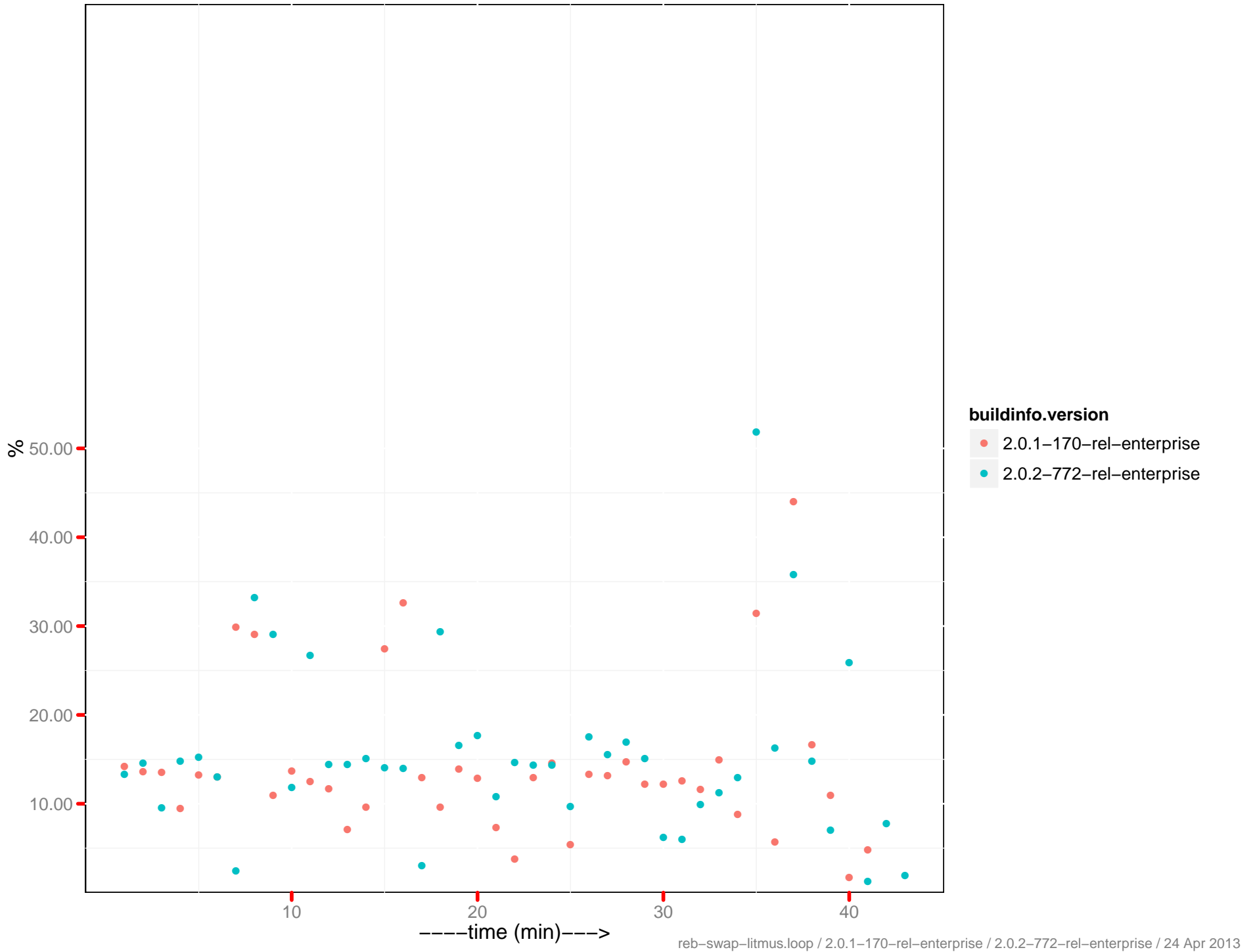
# Rebalance progress



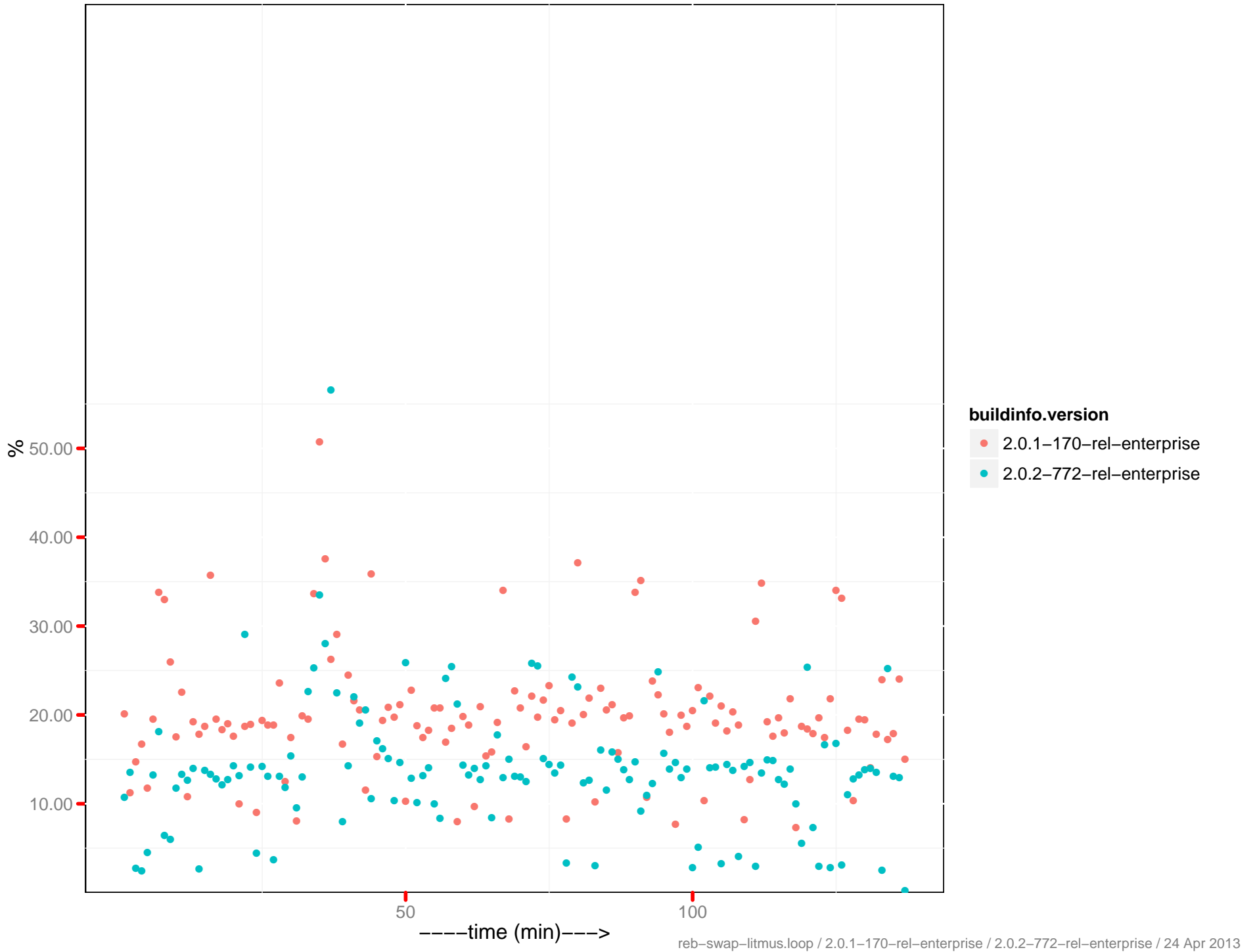
# CPU utilization – 10.6.2.37:8091



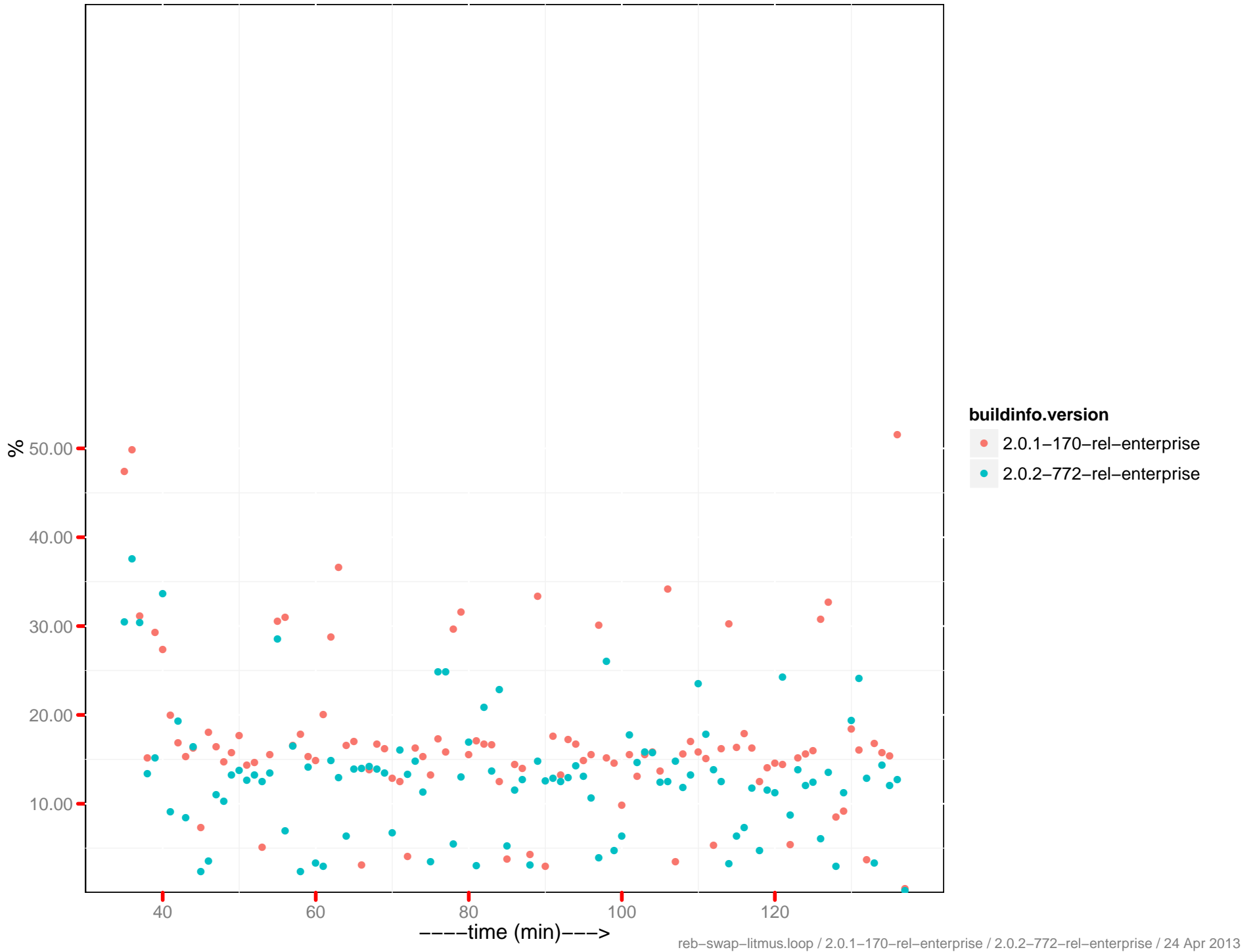
# CPU utilization - 10.6.2.38:8091



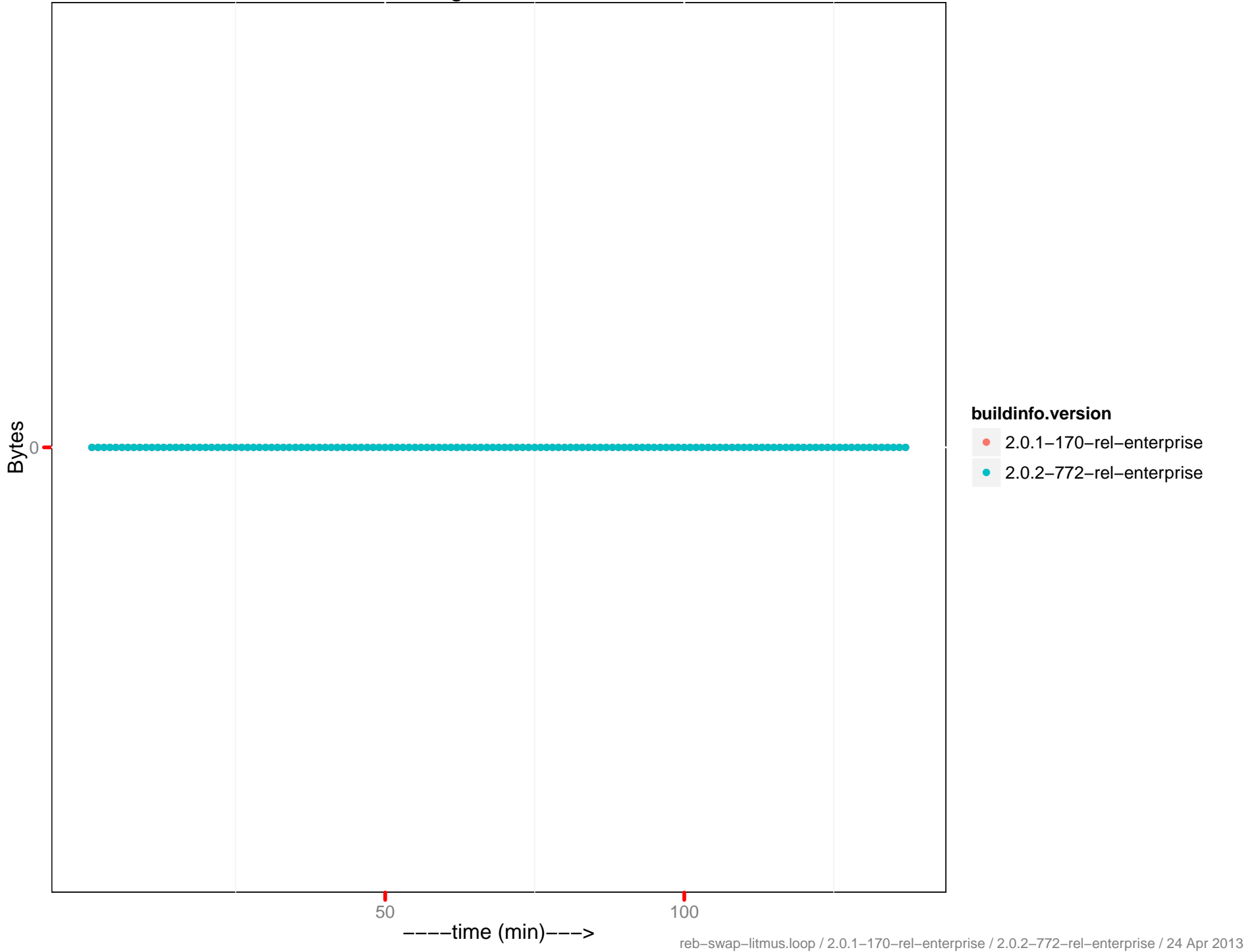
# CPU utilization – 10.6.2.39:8091



# CPU utilization – 10.6.2.40:8091

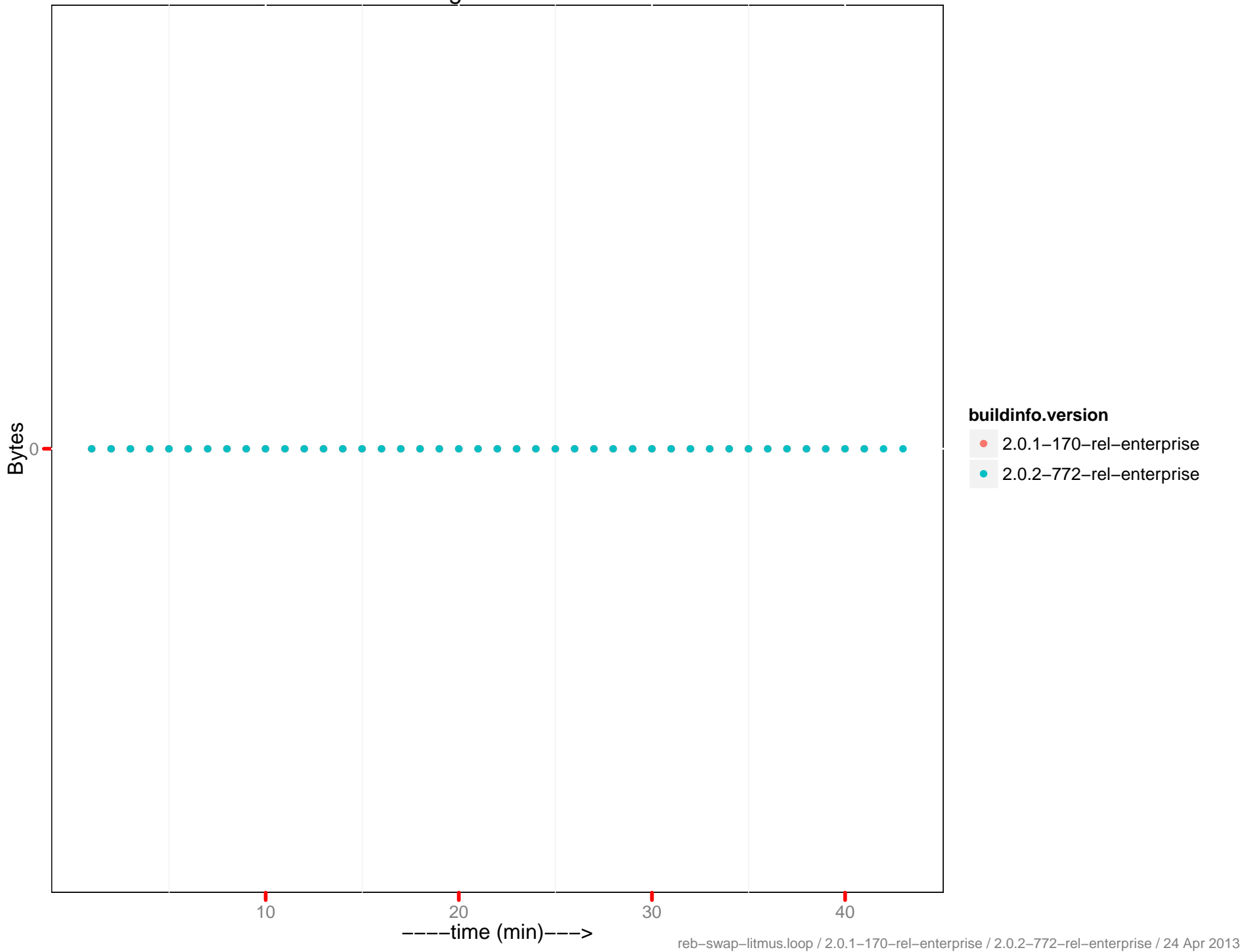


# SWAP Usage - 10.6.2.37:8091

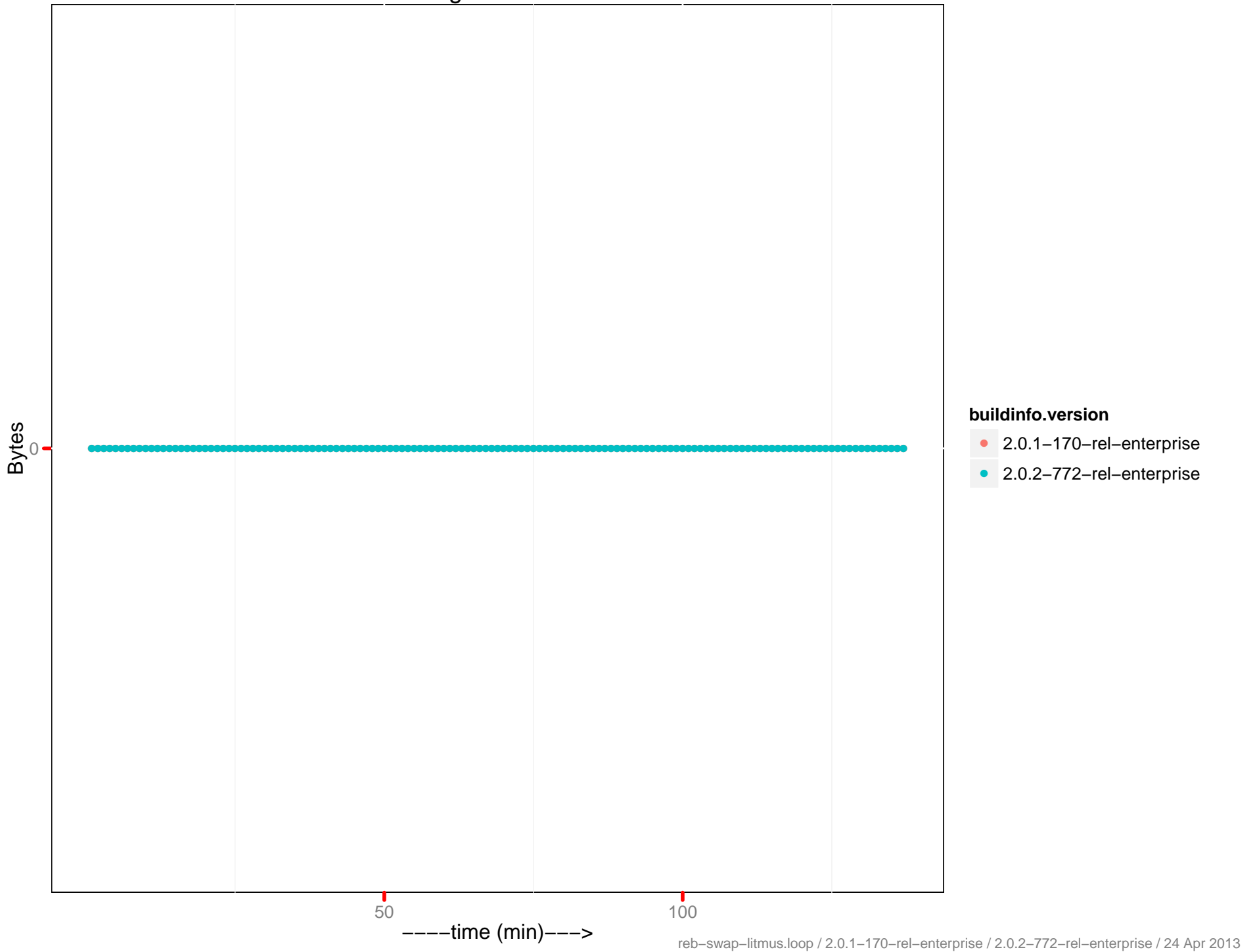




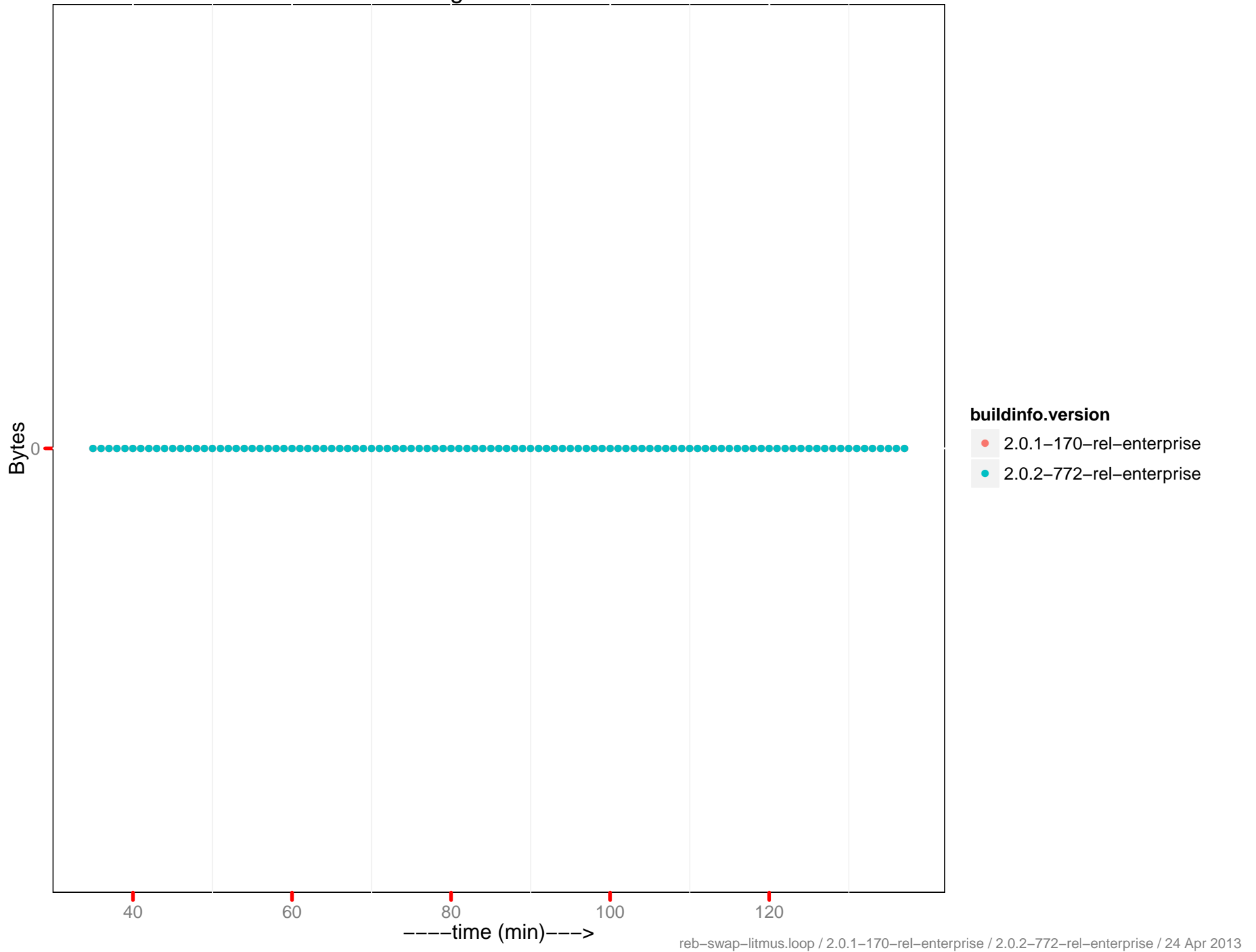
# SWAP Usage - 10.6.2.38:8091



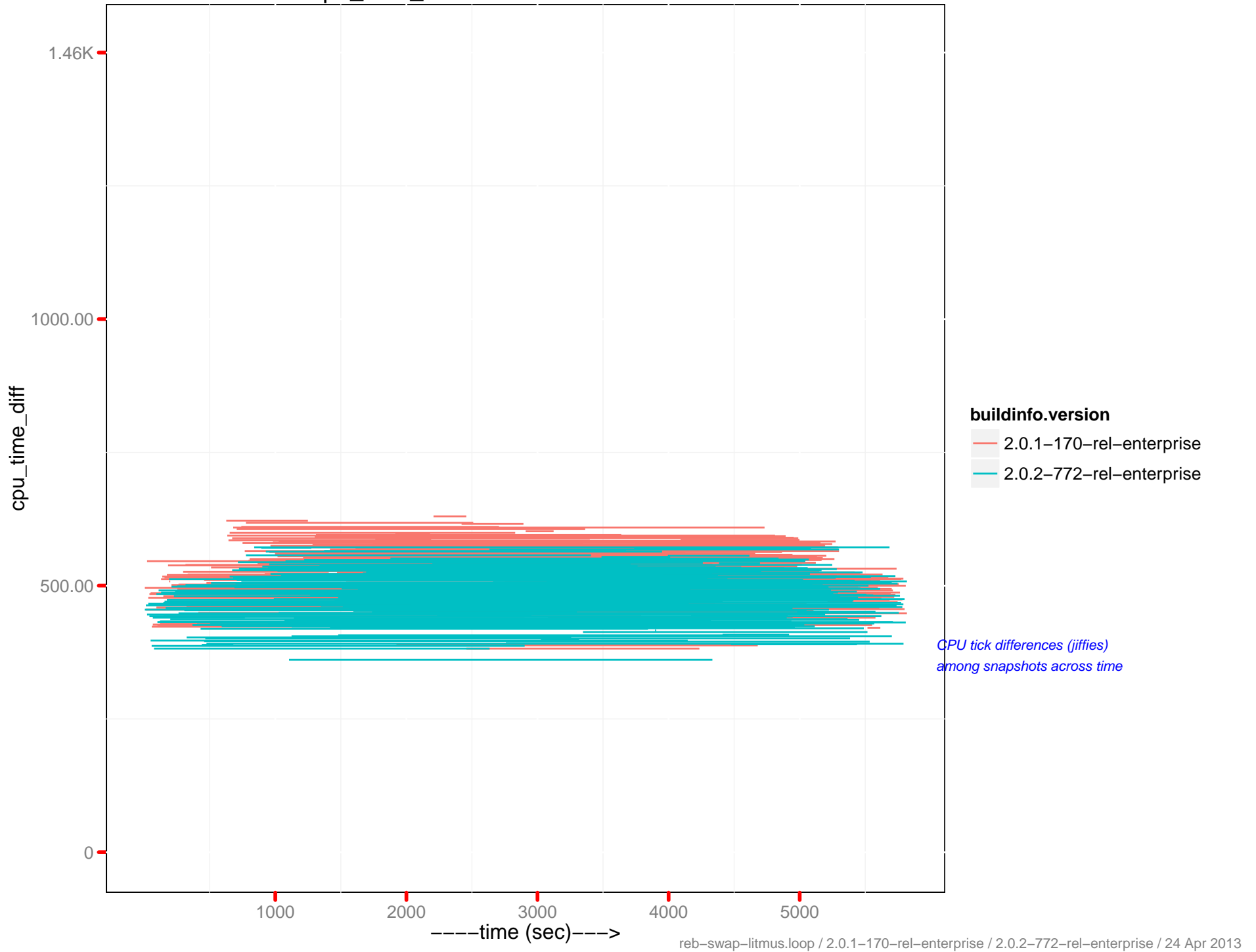
# SWAP Usage - 10.6.2.39:8091



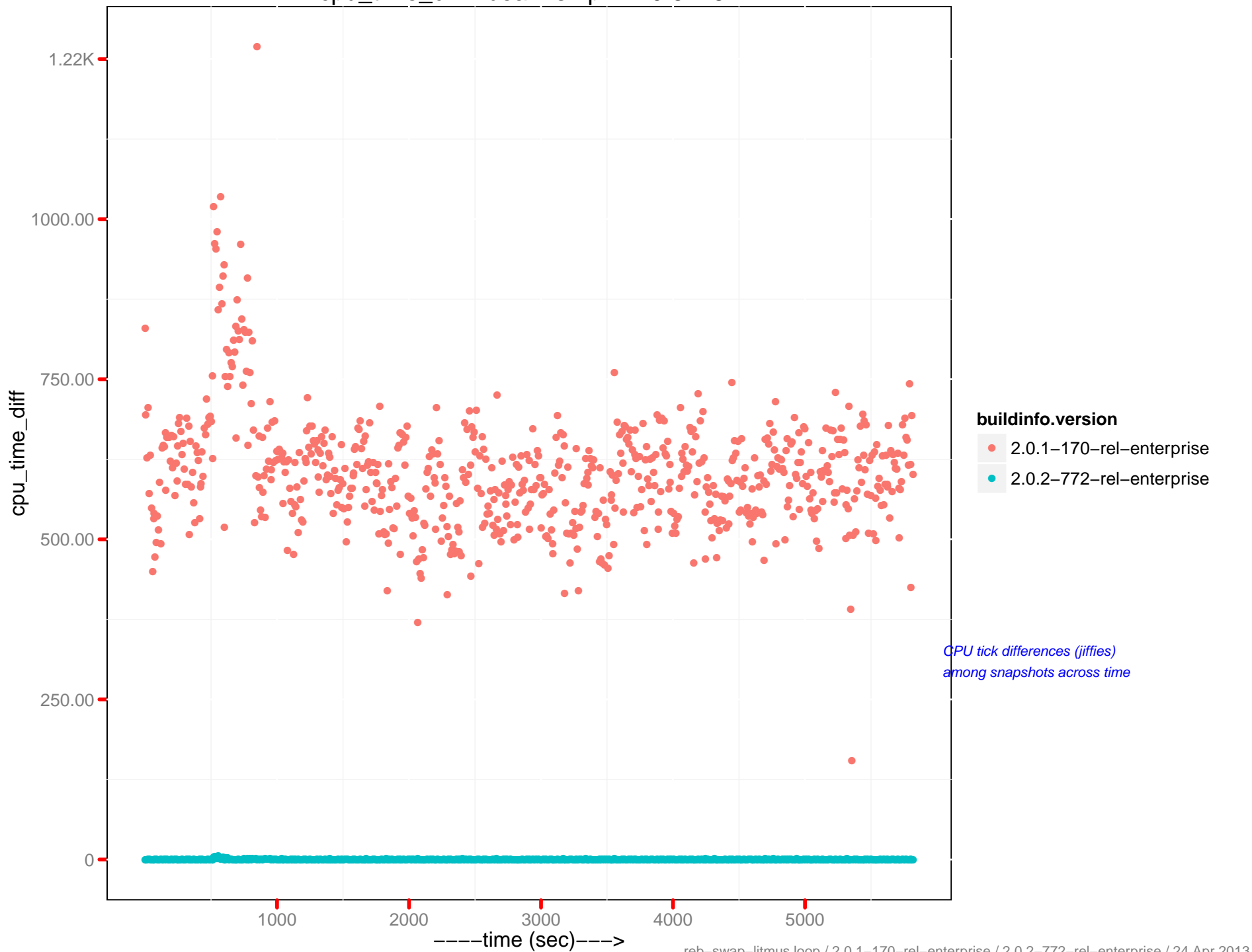
# SWAP Usage – 10.6.2.40:8091



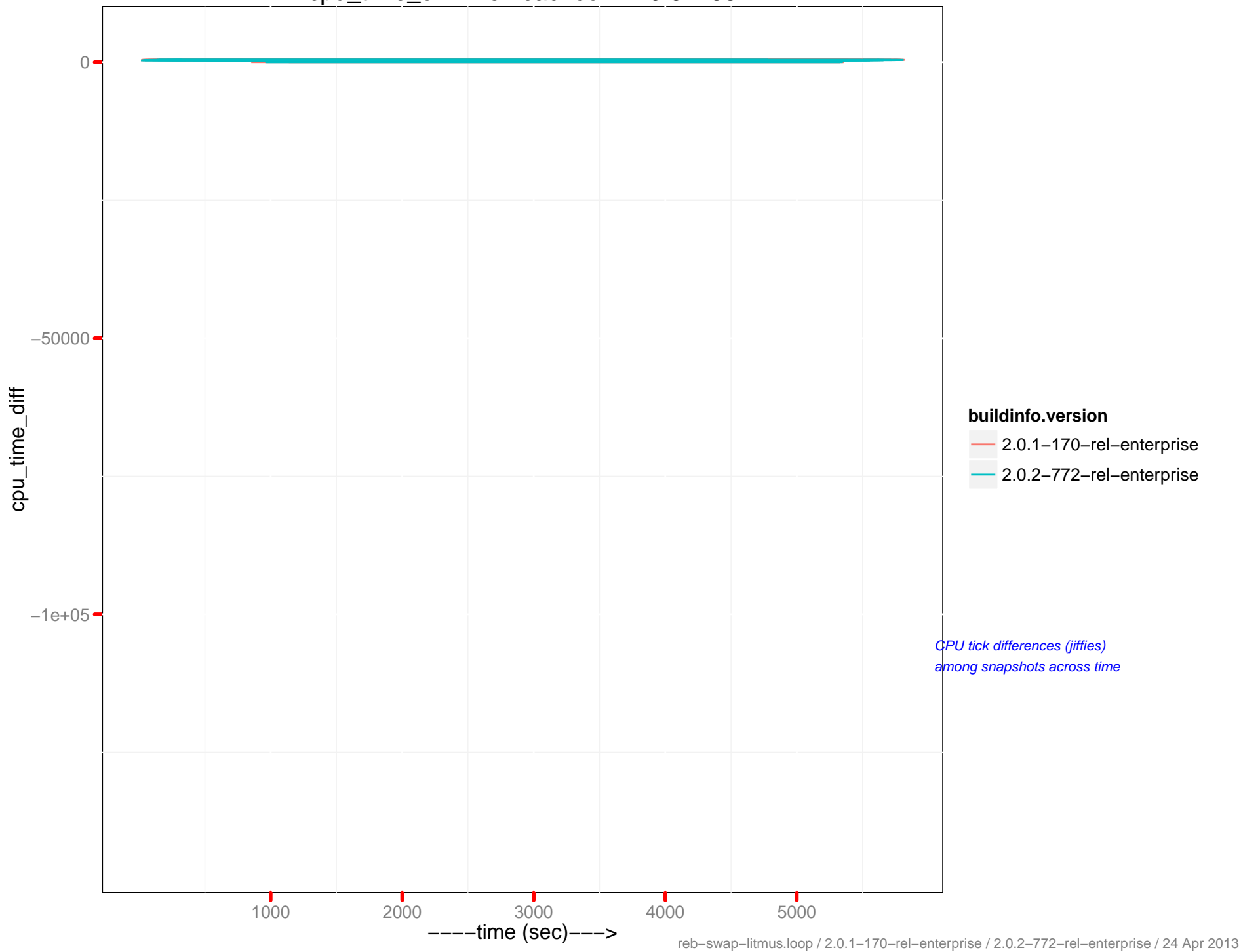
# cpu\_time\_diff: memcached - 10.6.2.37



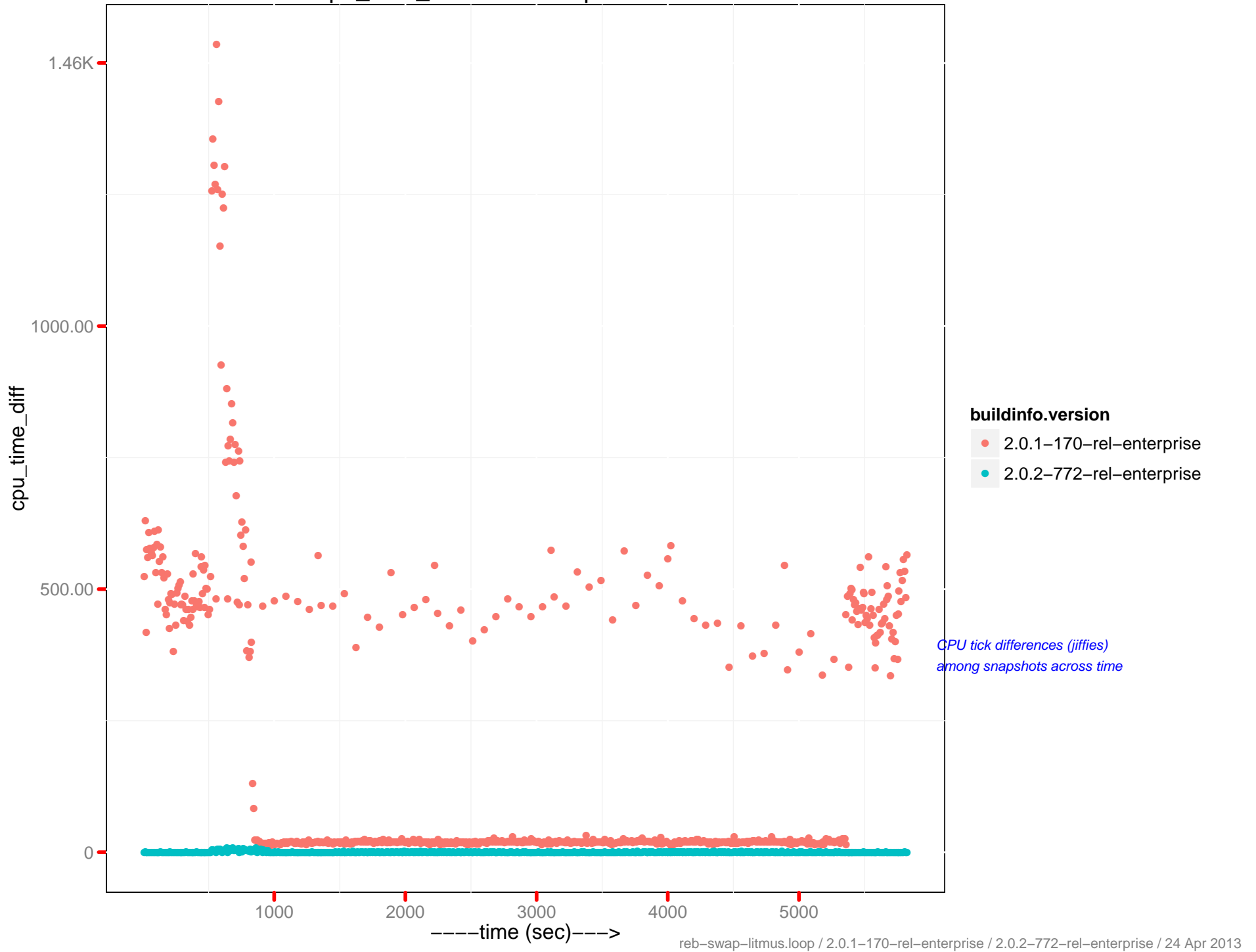
# cpu\_time\_diff : beam.smp - 10.6.2.37



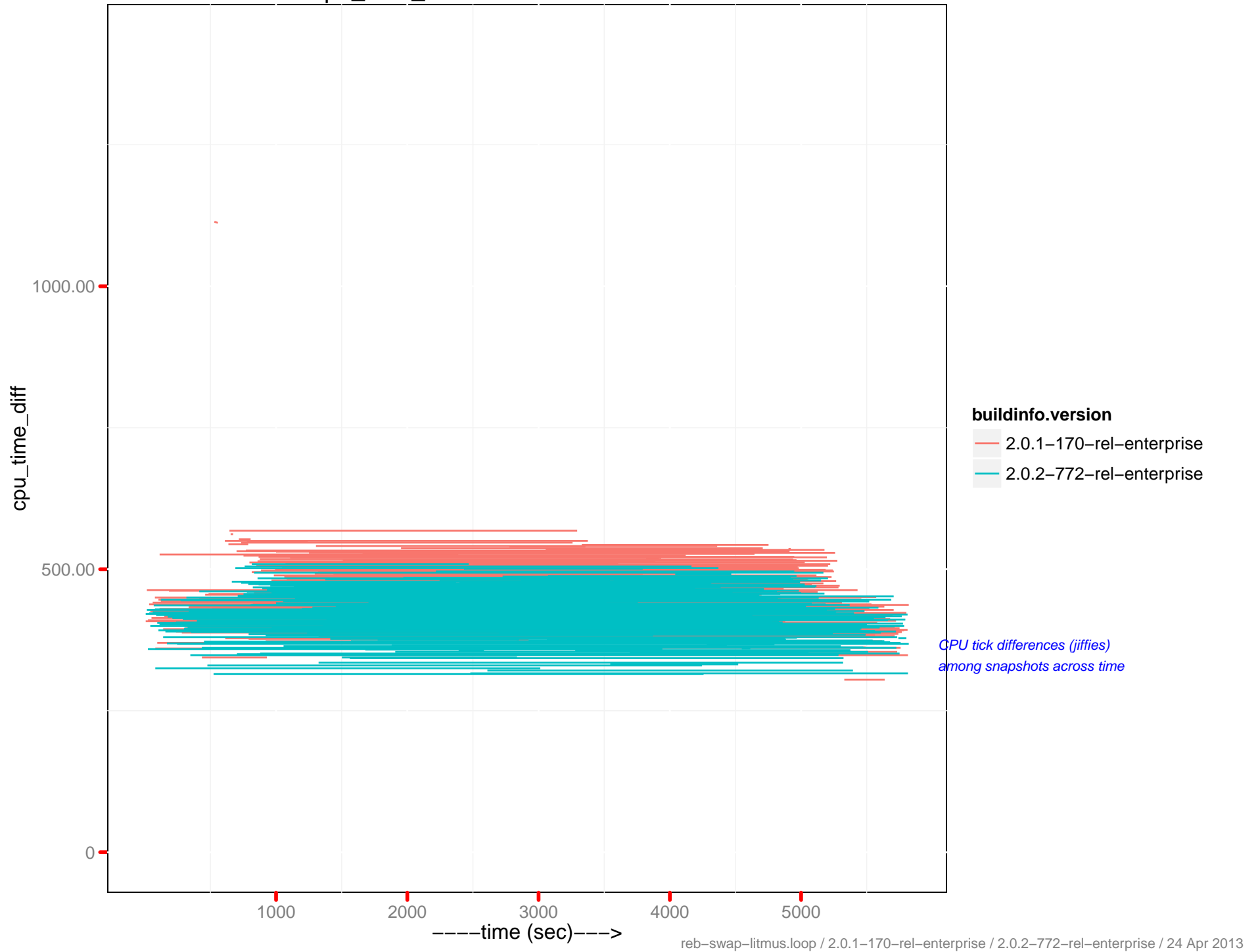
# cpu\_time\_diff: memcached - 10.6.2.38



cpu\_time\_diff : beam.smp - 10.6.2.38

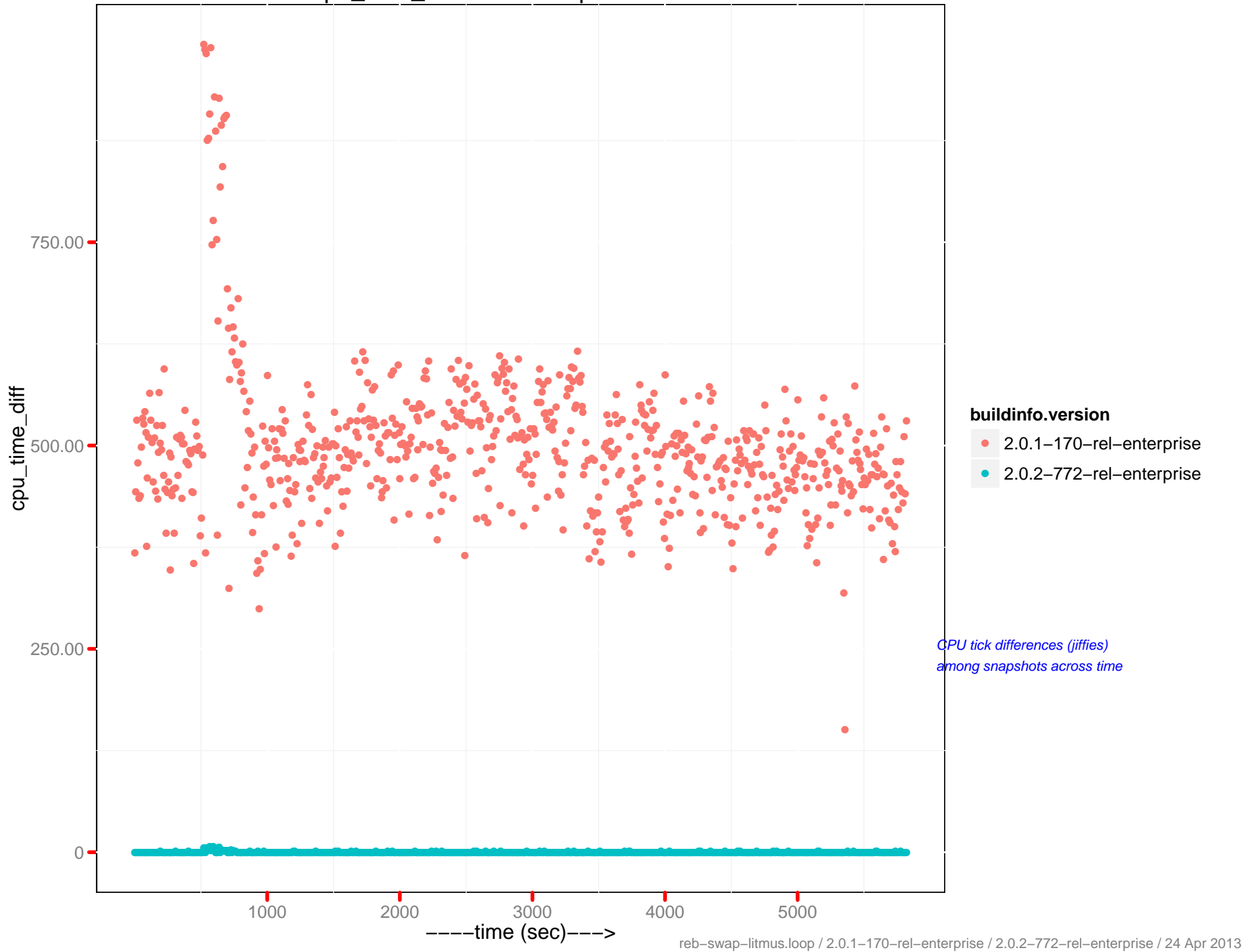


# cpu\_time\_diff: memcached - 10.6.2.39

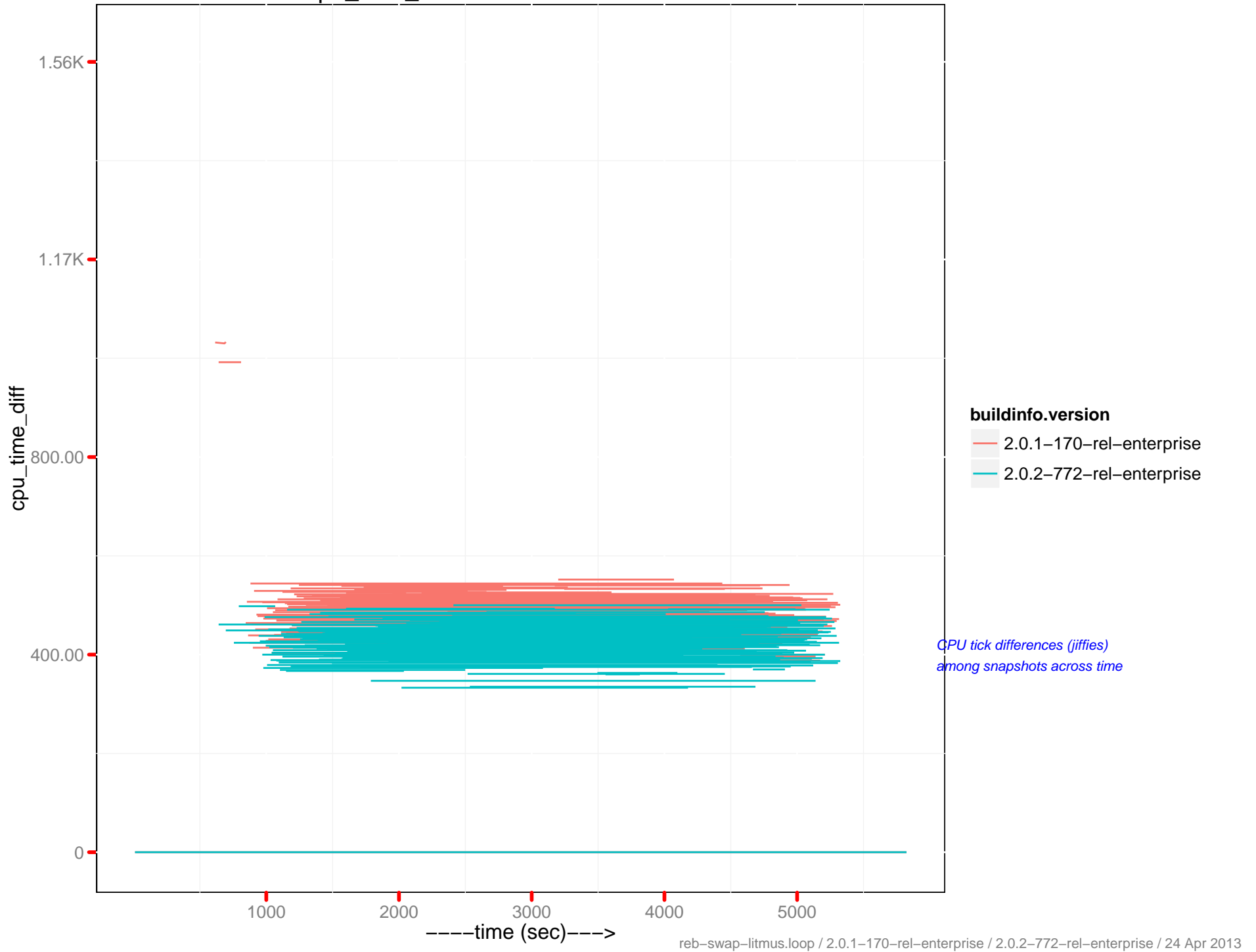




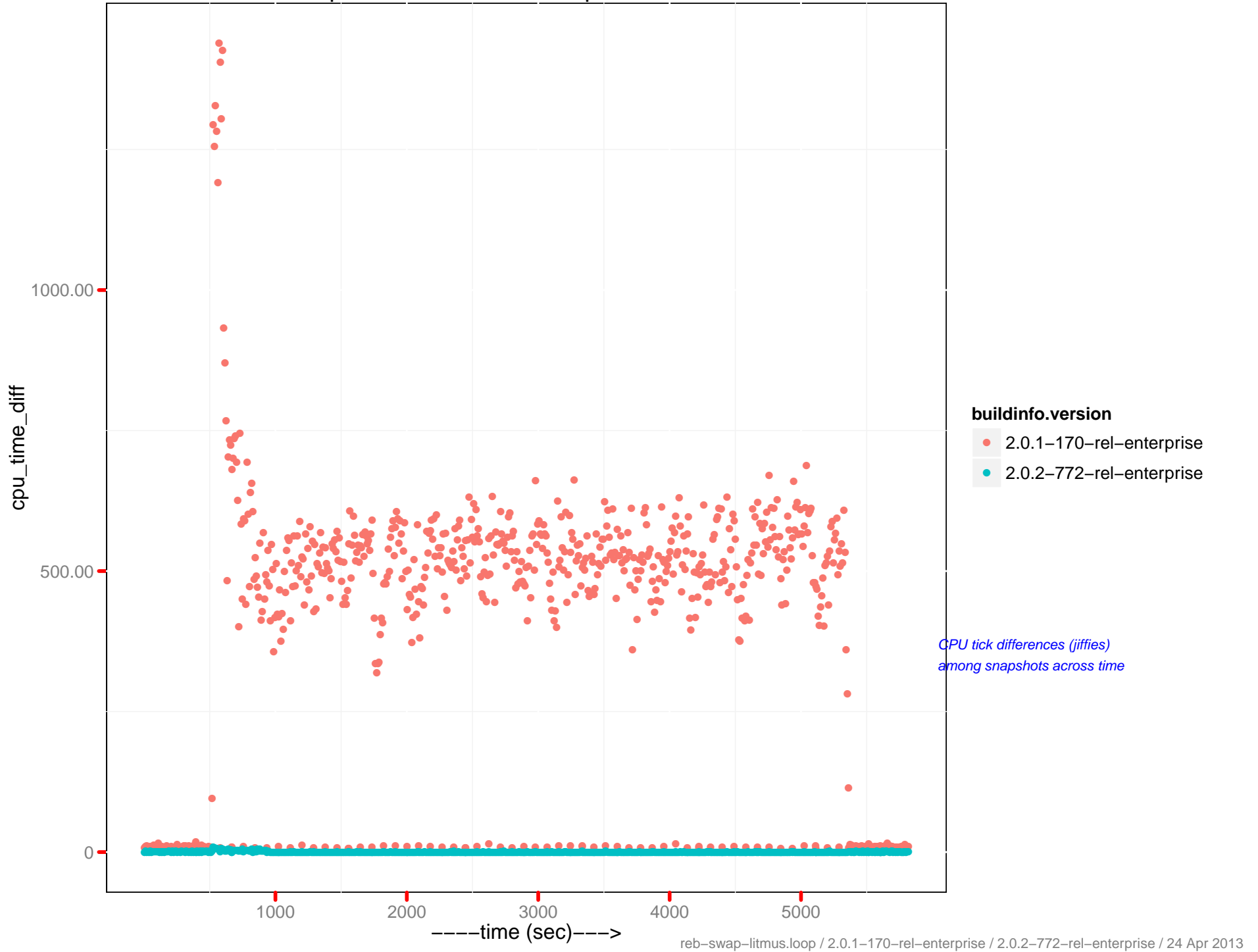
# cpu\_time\_diff : beam.smp - 10.6.2.39



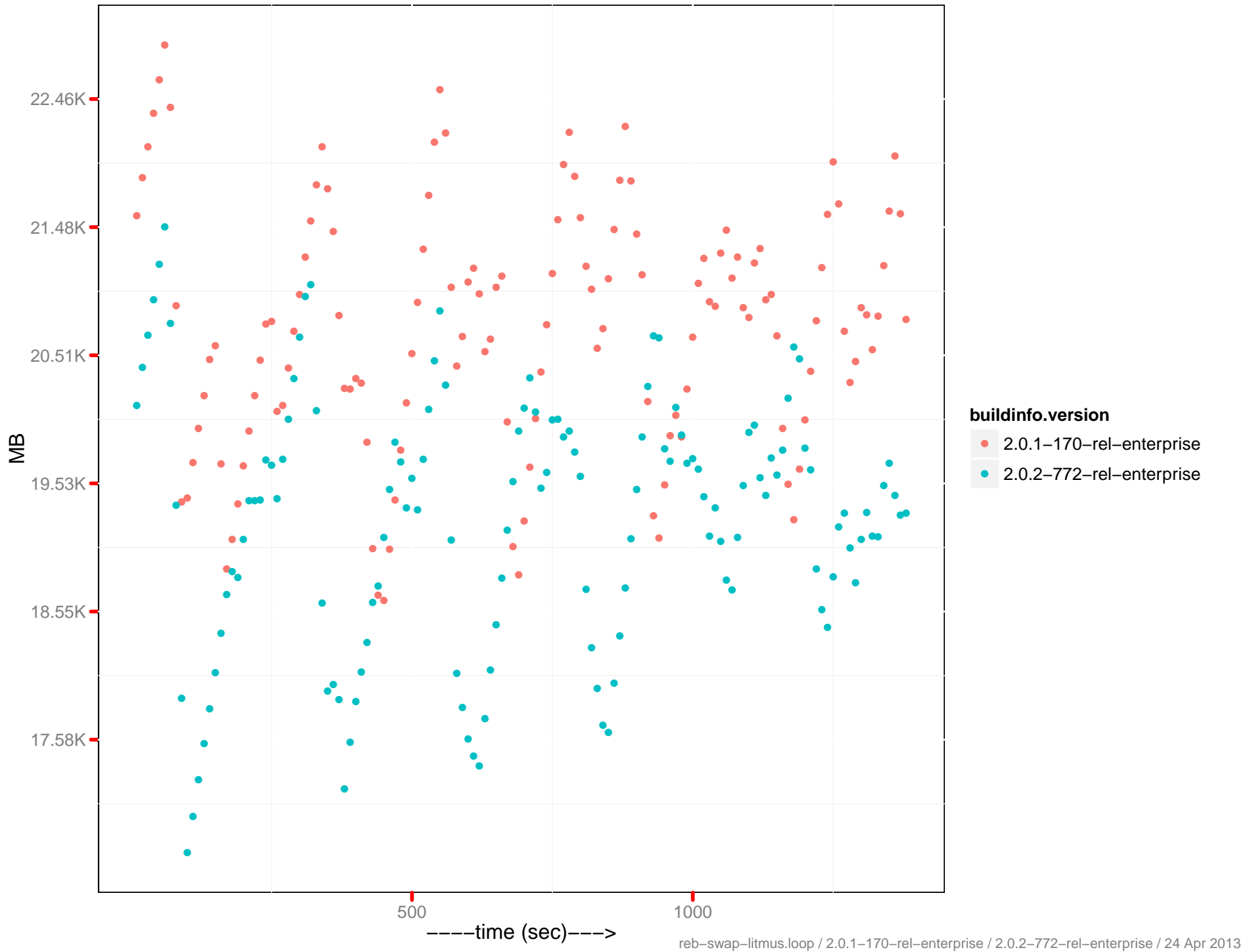
# cpu\_time\_diff: memcached – 10.6.2.40



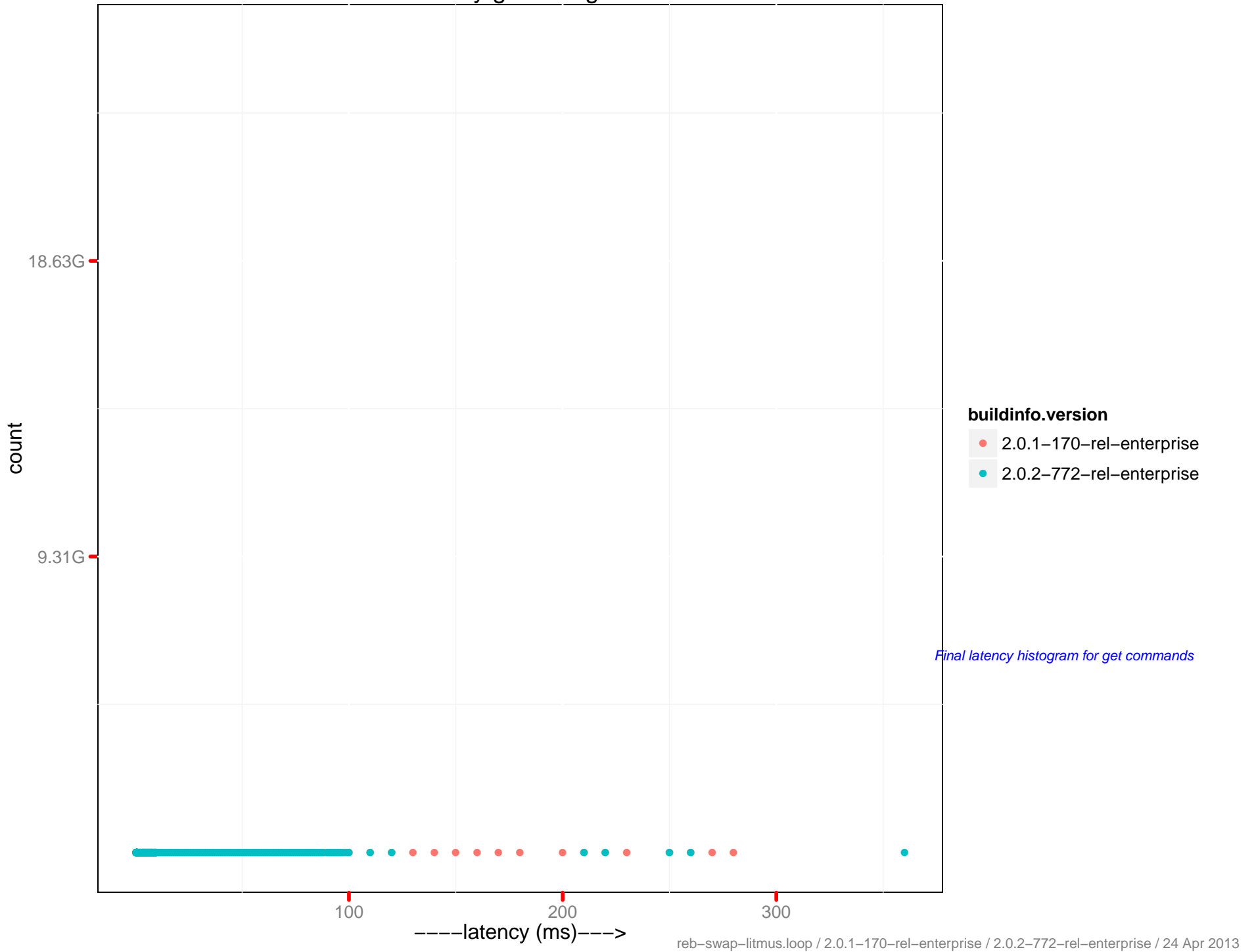
cpu\_time\_diff : beam.smp - 10.6.2.40



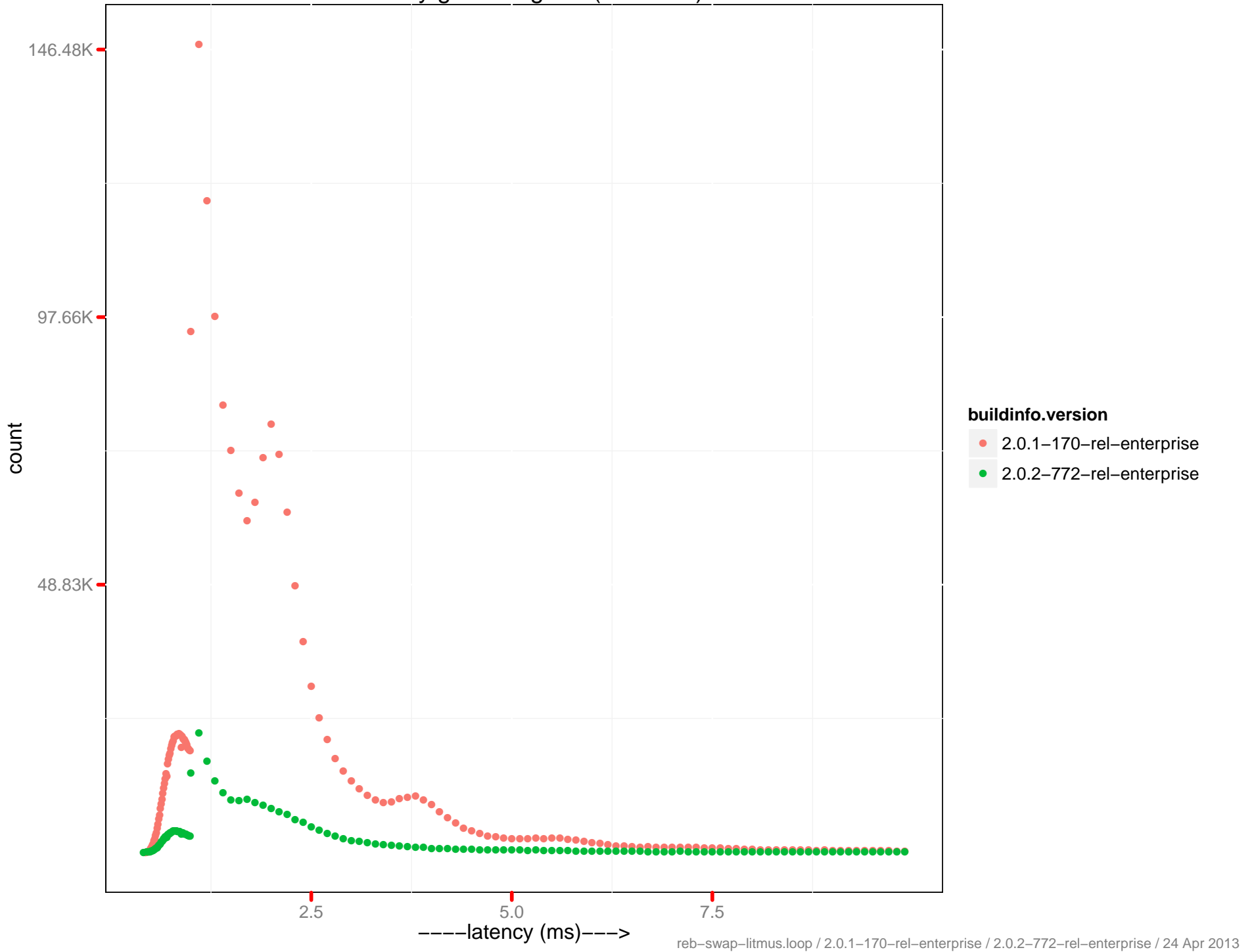
# Data disk size



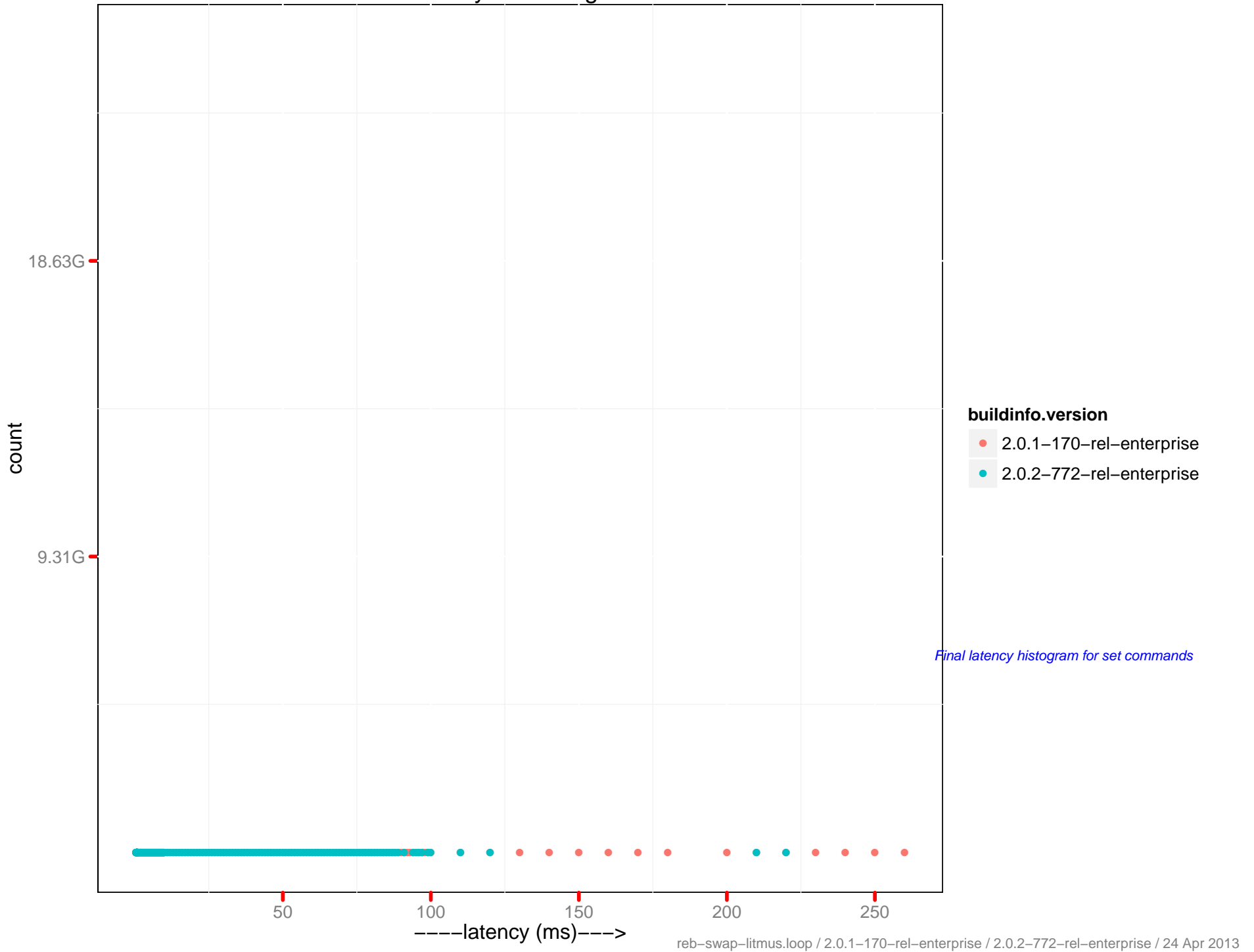
# Latency get histogram



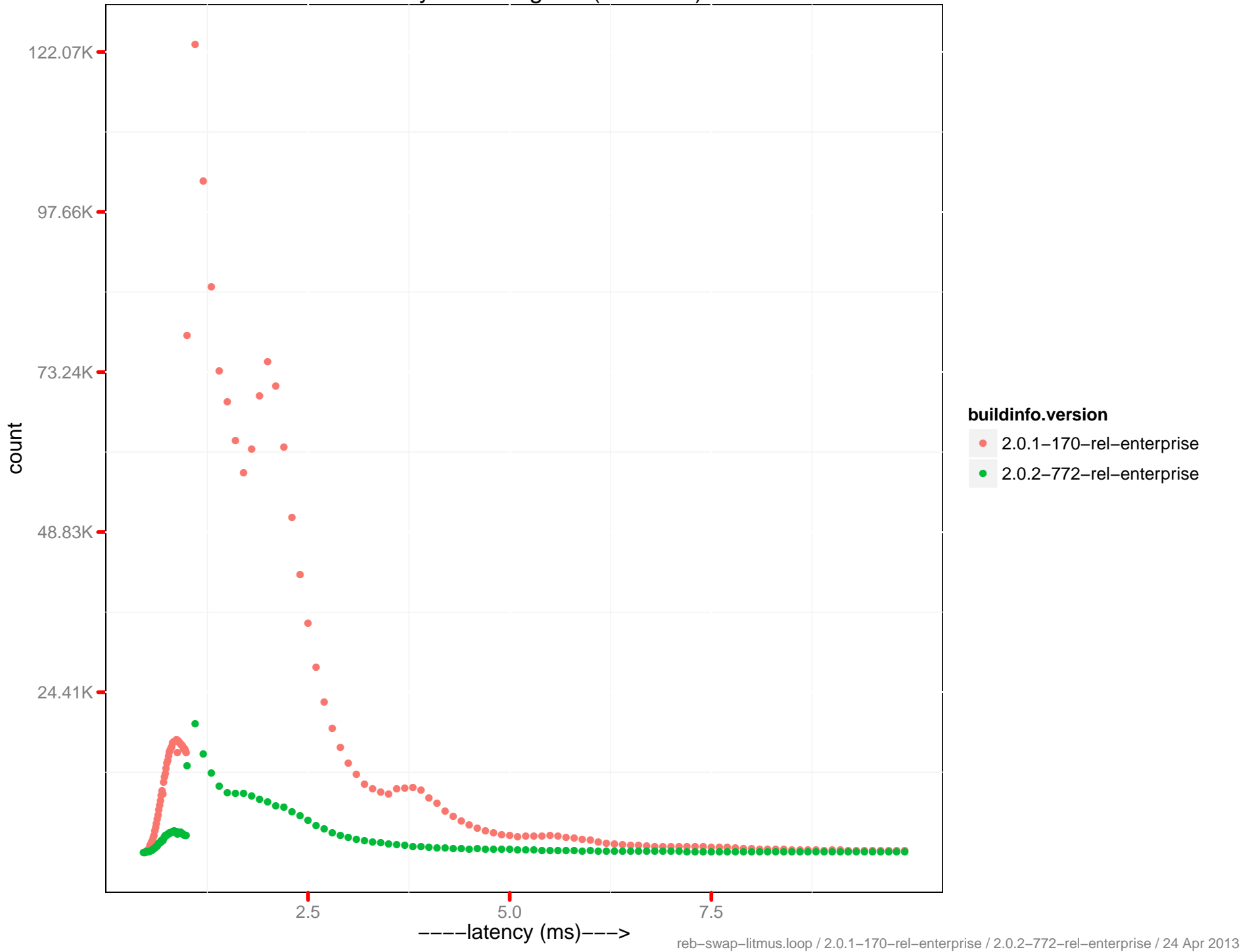
Latency get histogram (0–10 ms)



# Latency set histogram

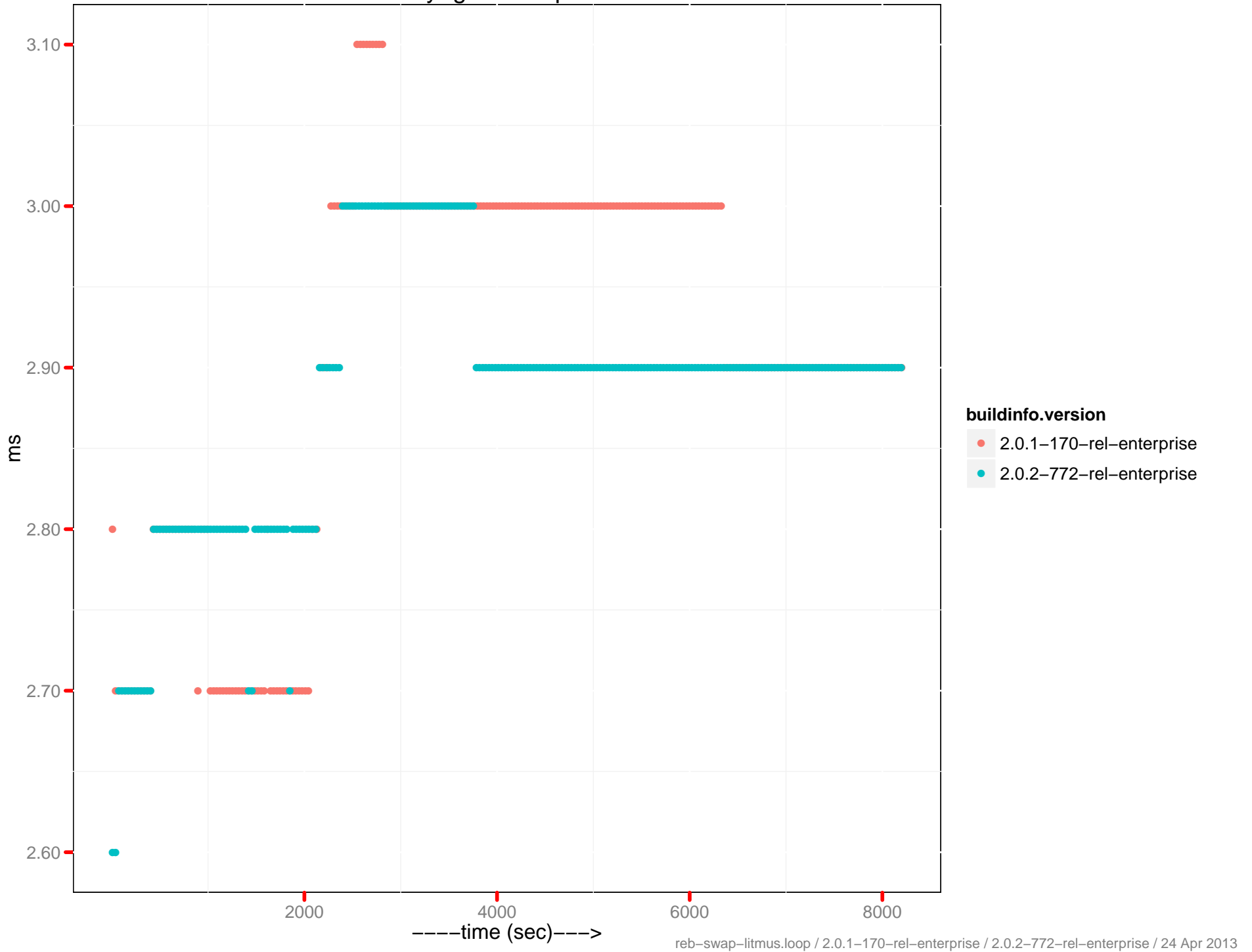


Latency set histogram (0–10 ms)

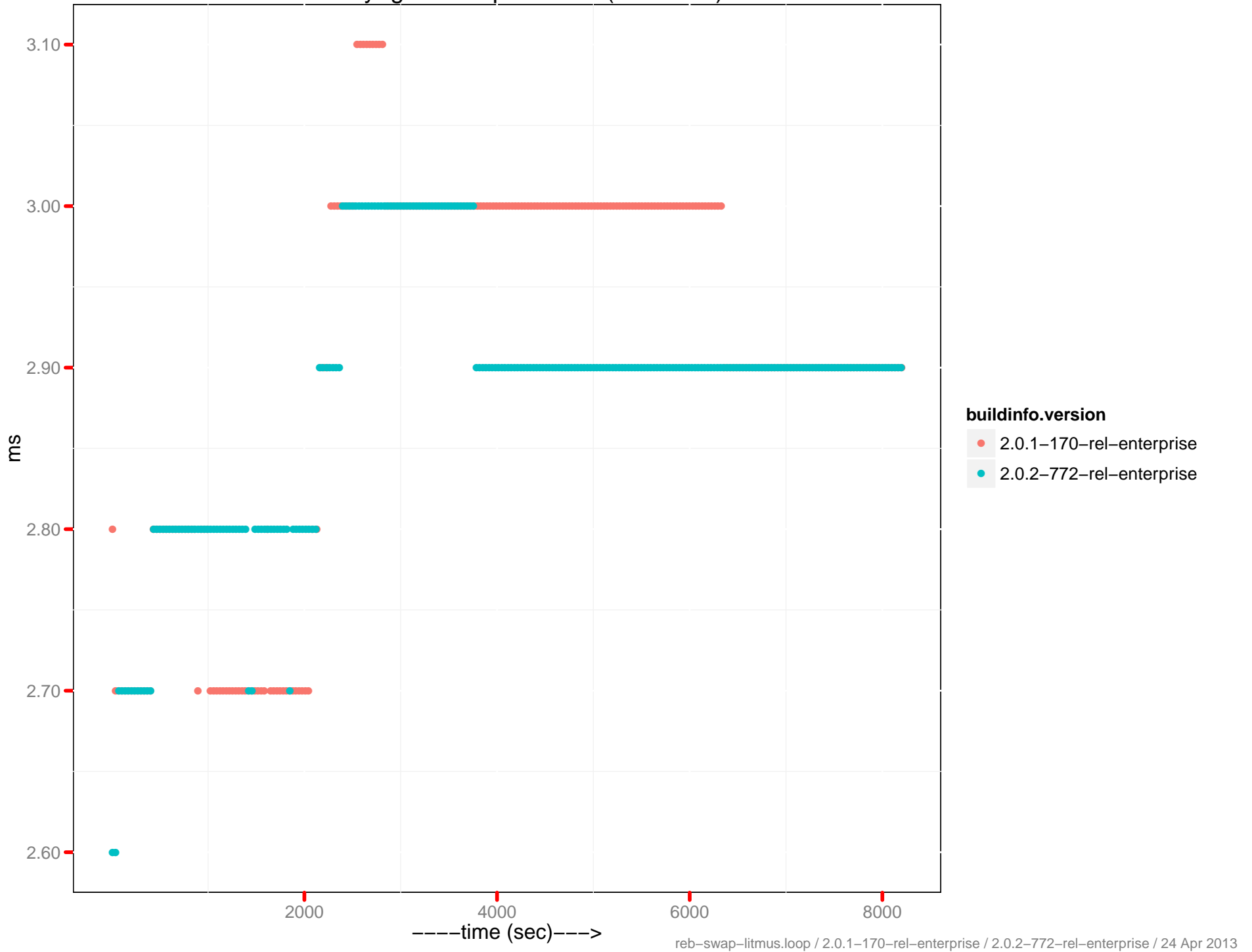




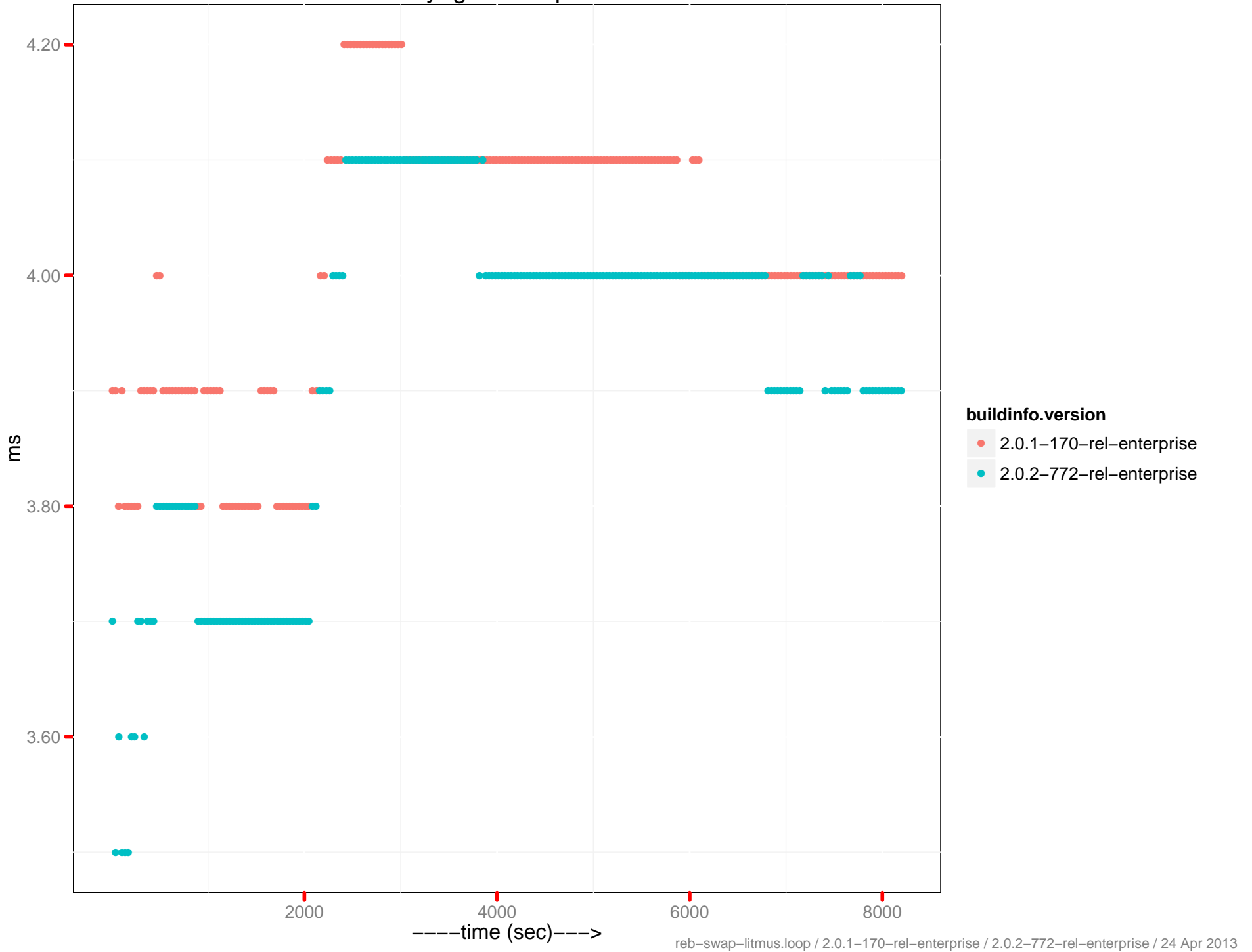
# Latency-get 90th percentile



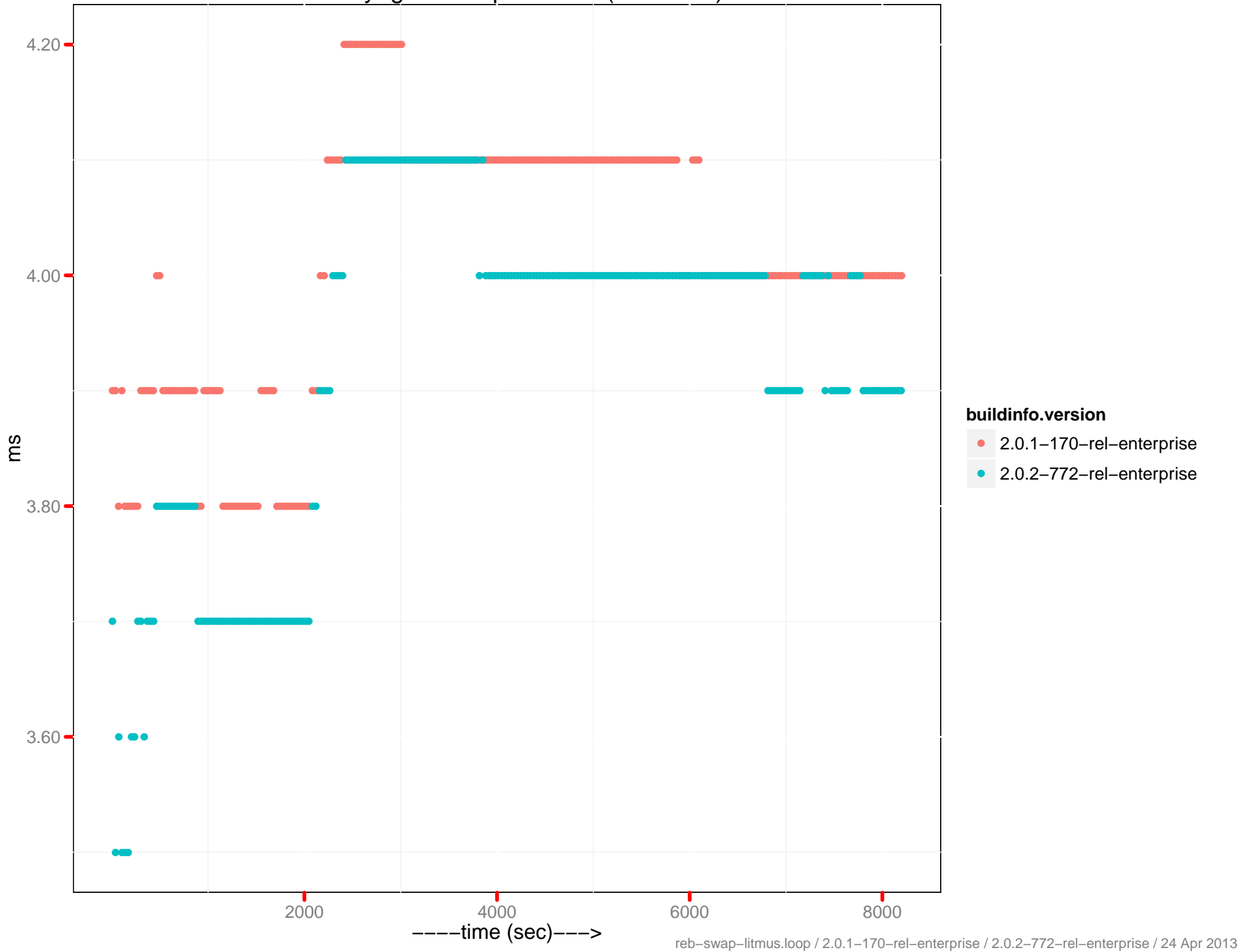
# Latency-get 90th percentile (0 - 10ms)



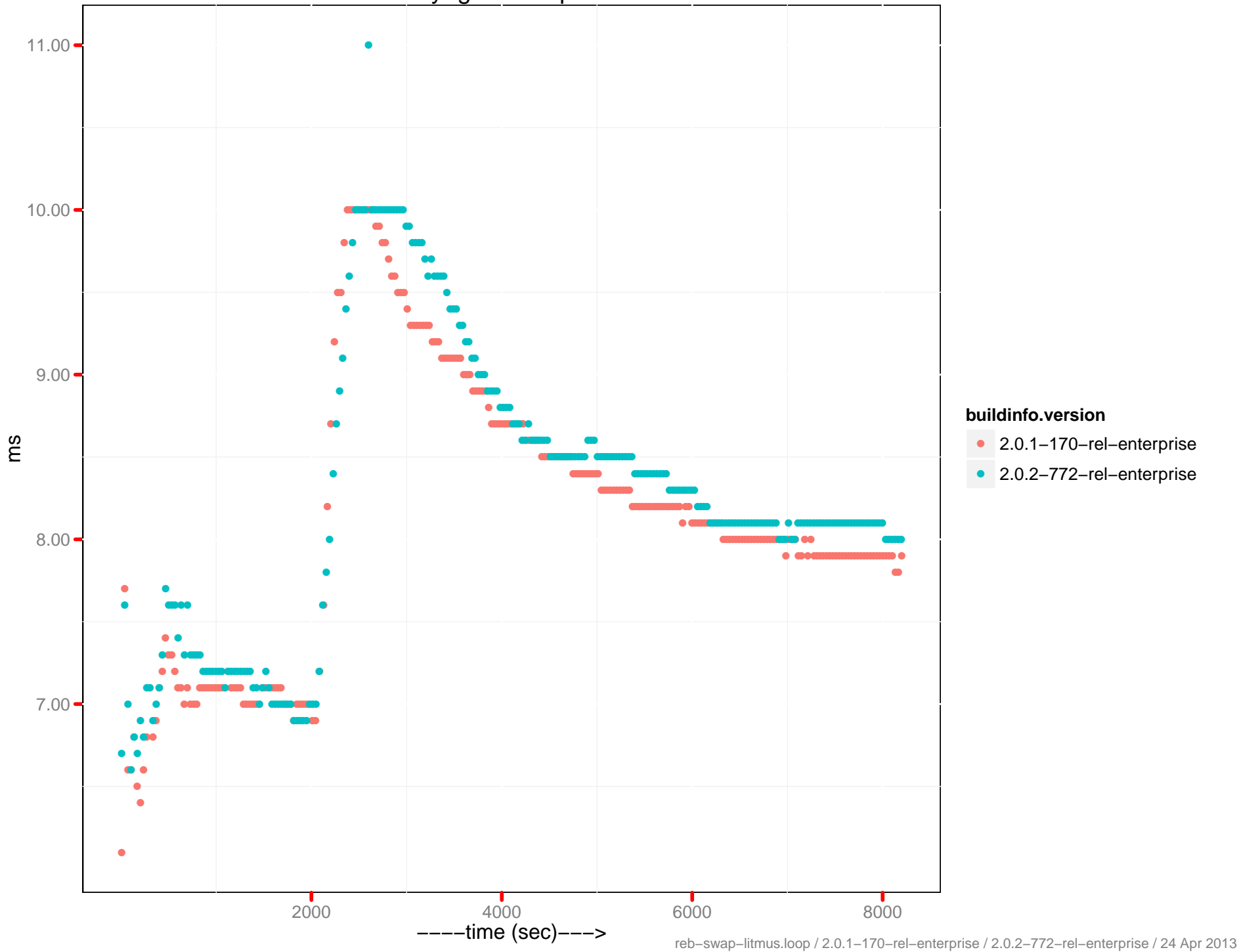
# Latency-get 95th percentile



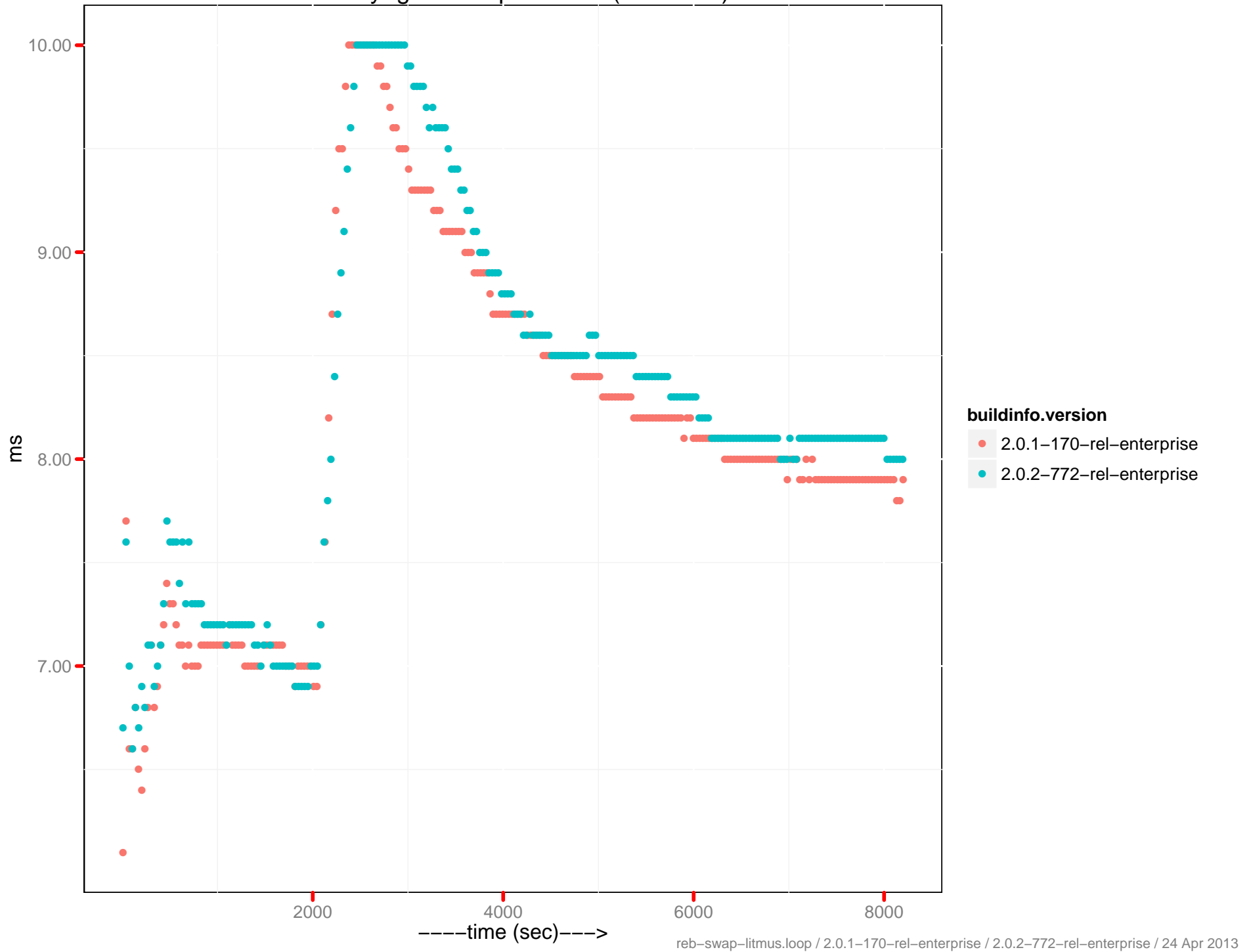
Latency-get 95th percentile (0 - 10ms)



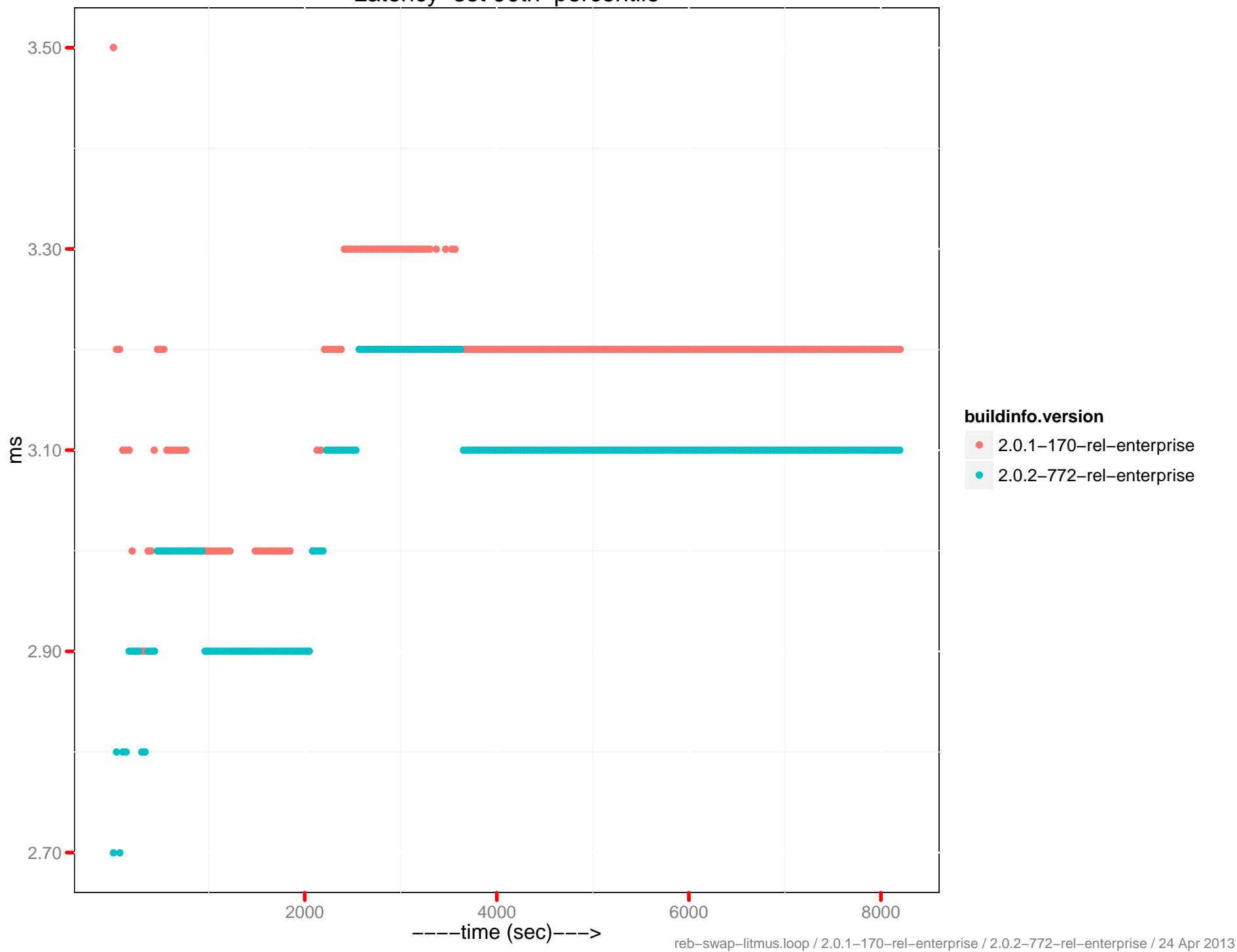
# Latency-get 99th percentile



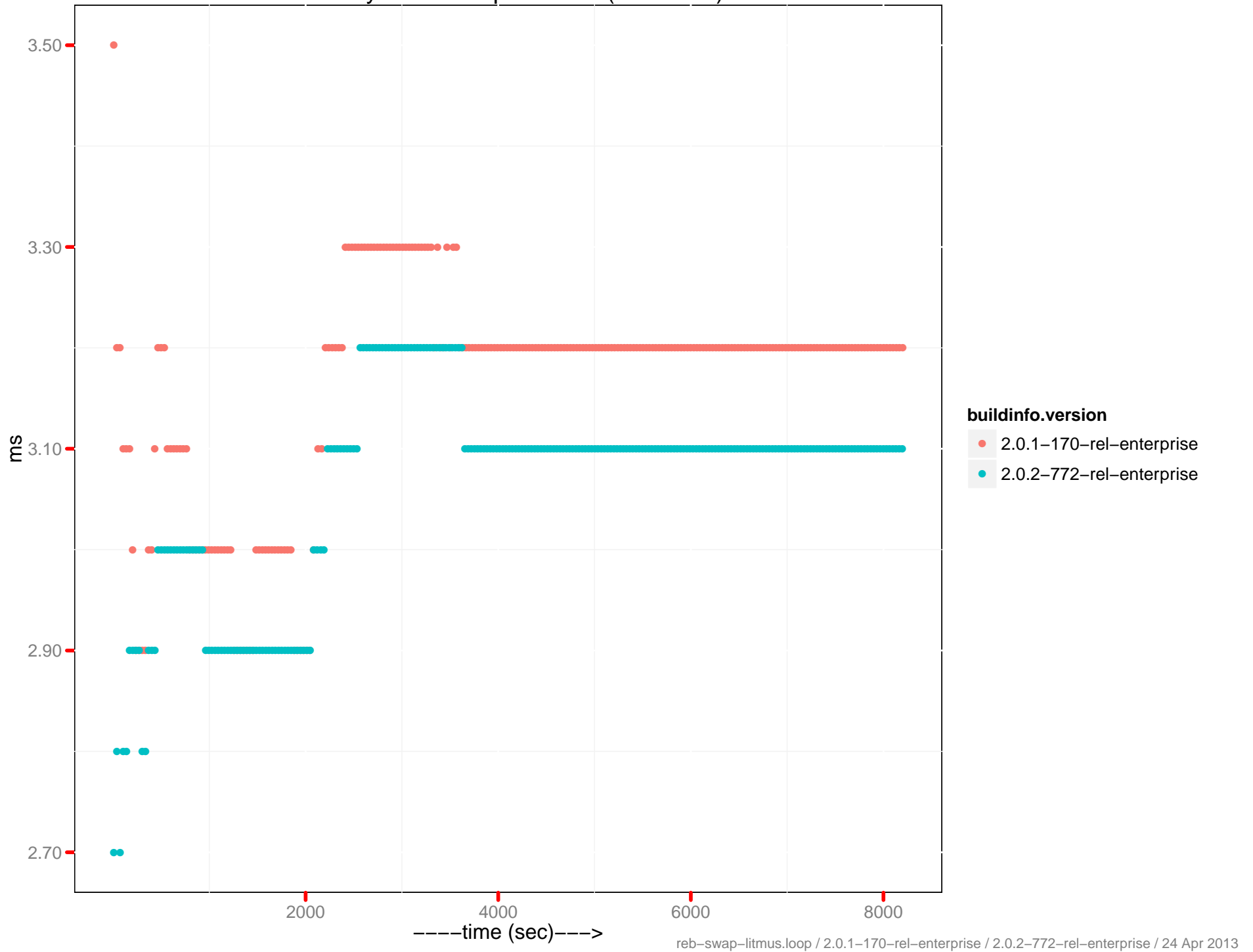
Latency-get 99th percentile (0 - 10ms)



# Latency-set 90th percentile

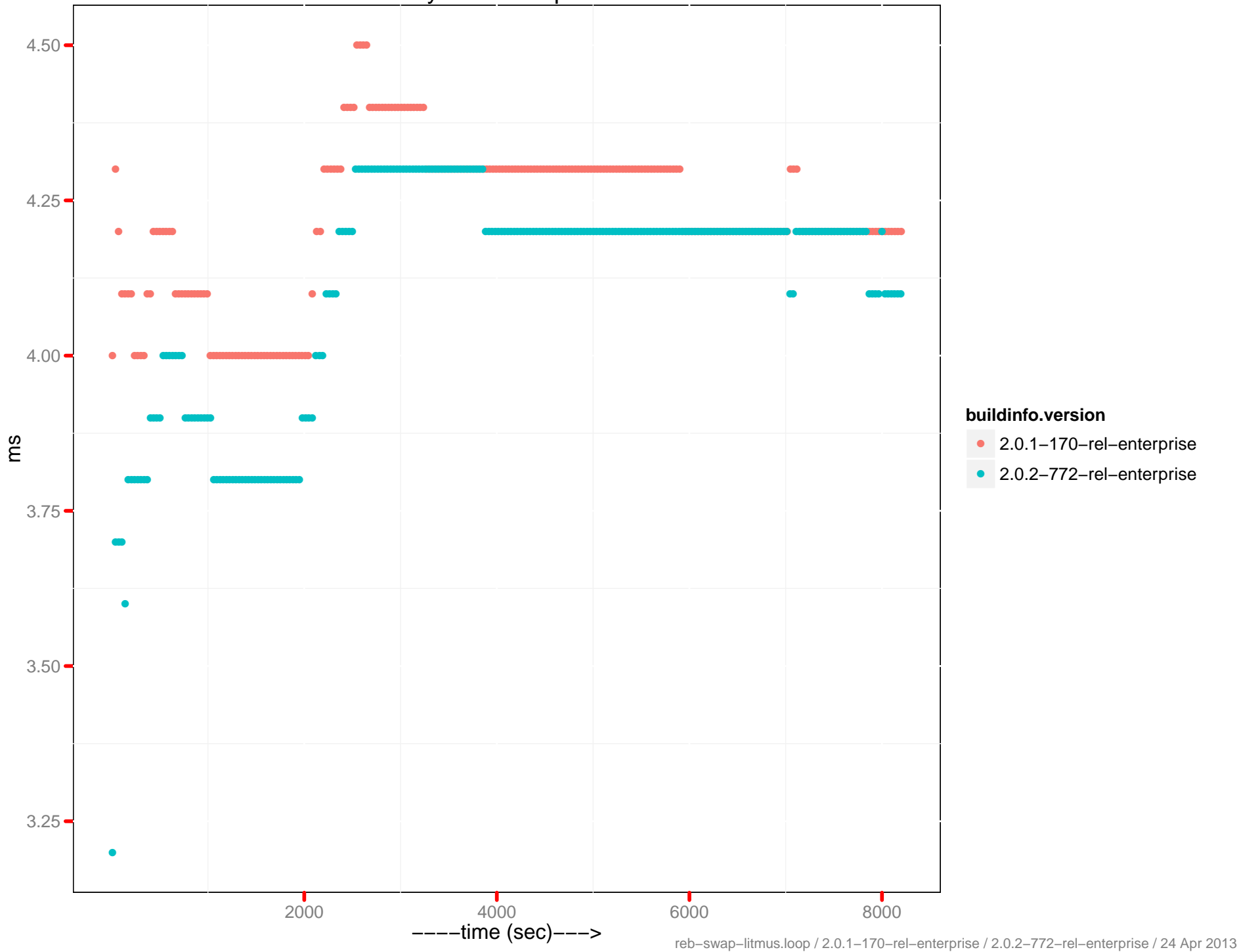


# Latency-set 90th percentile (0 - 10ms)

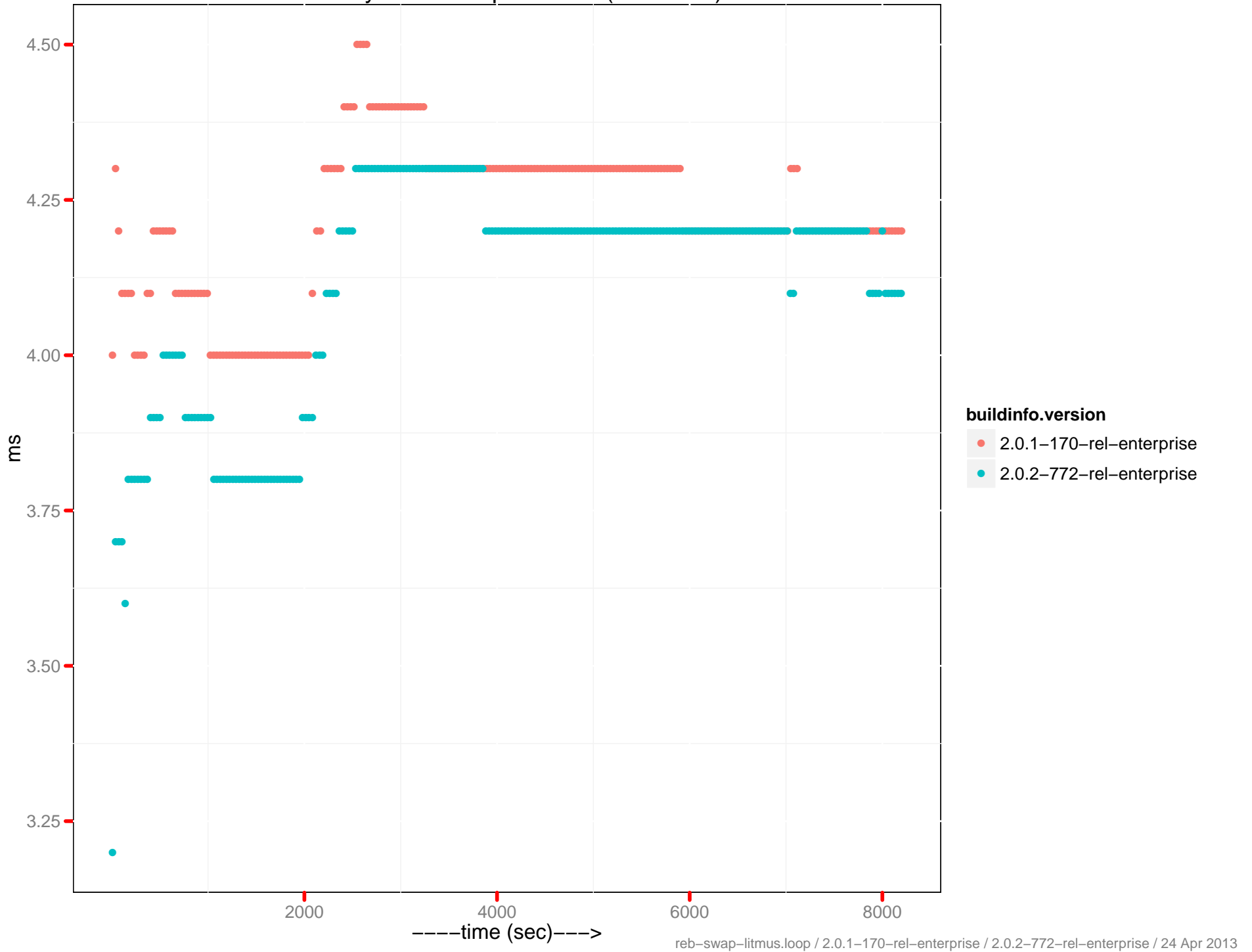




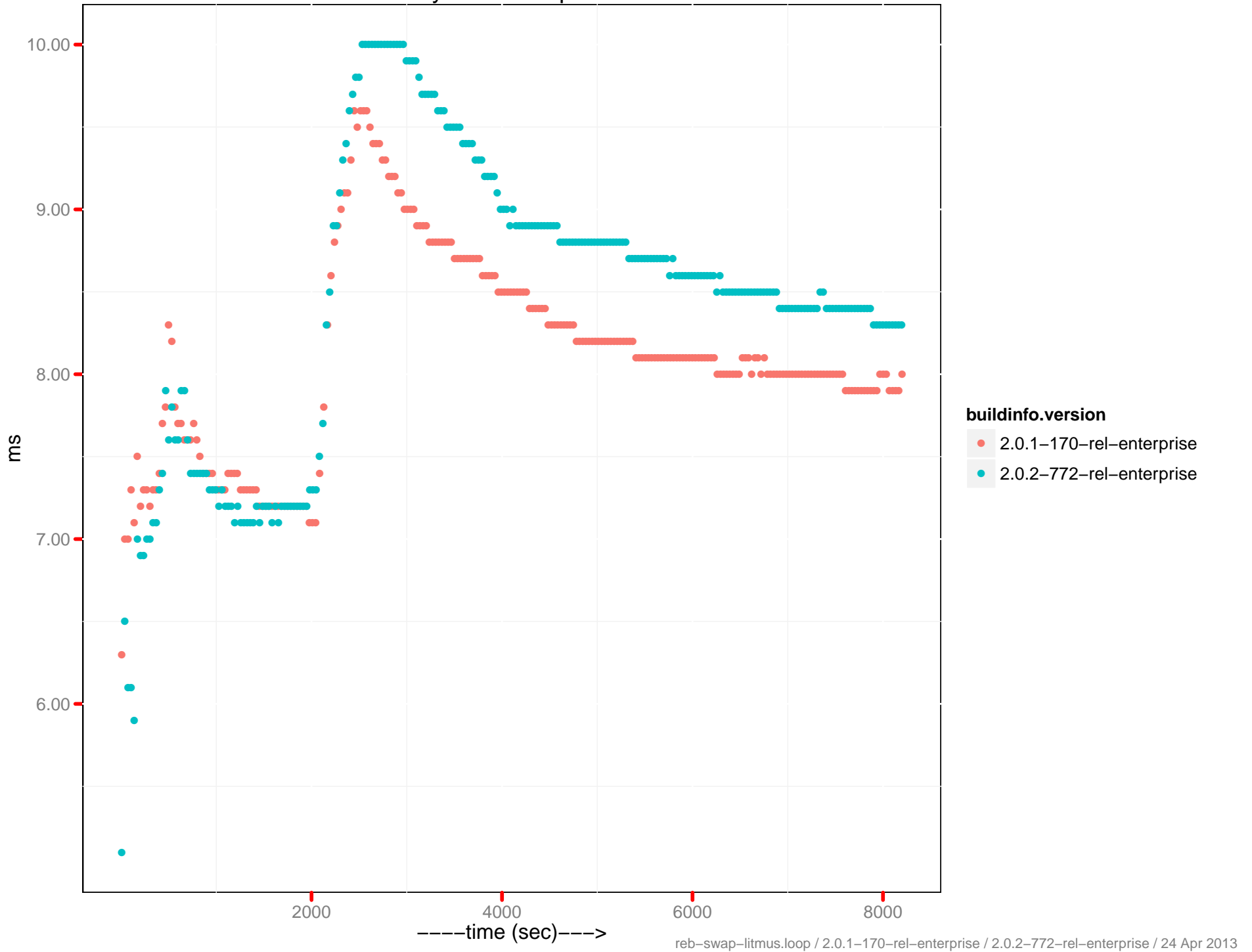
# Latency-set 95th percentile



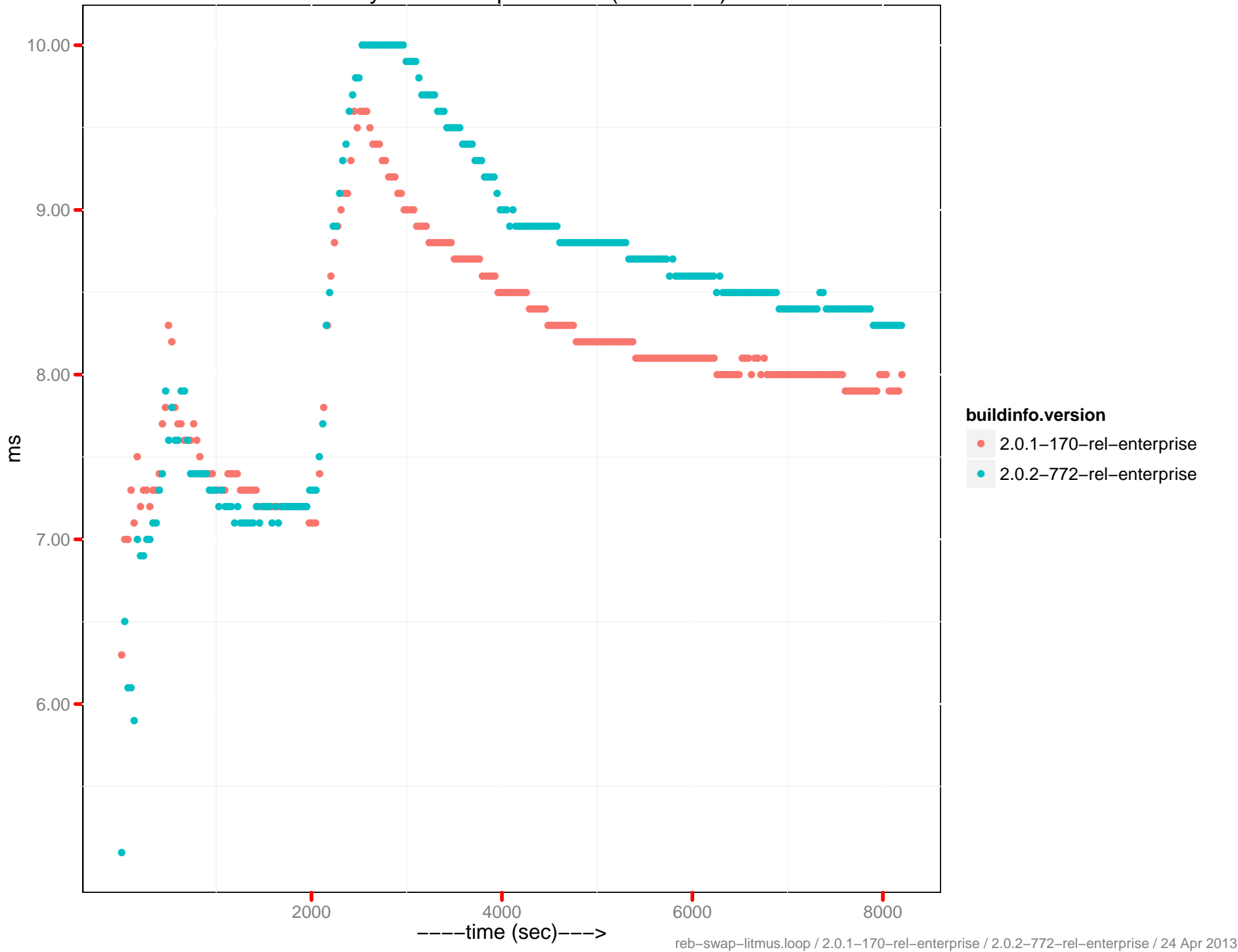
Latency-set 95th percentile (0 - 10ms)



Latency-set 99th percentile



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



# Query throughput

