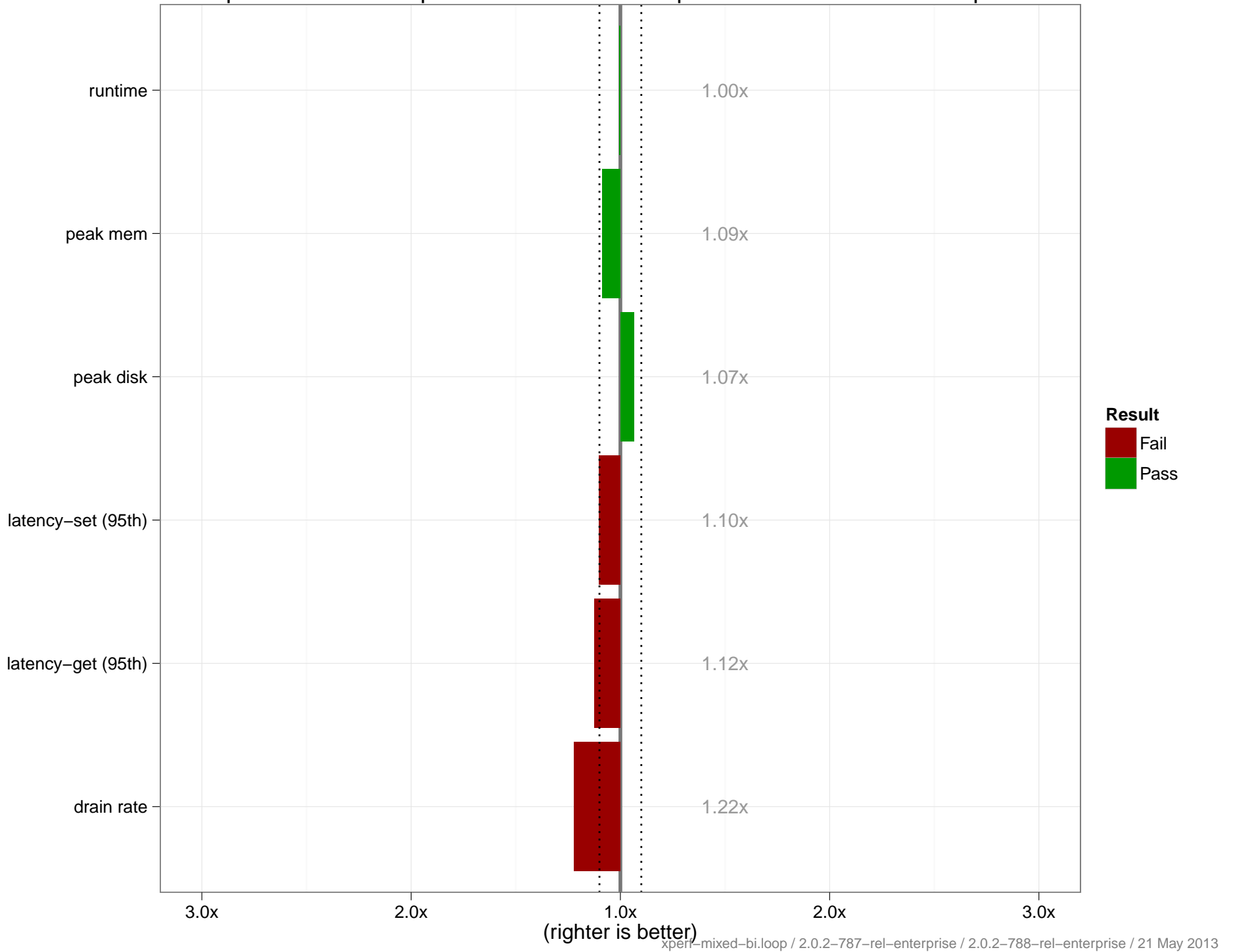


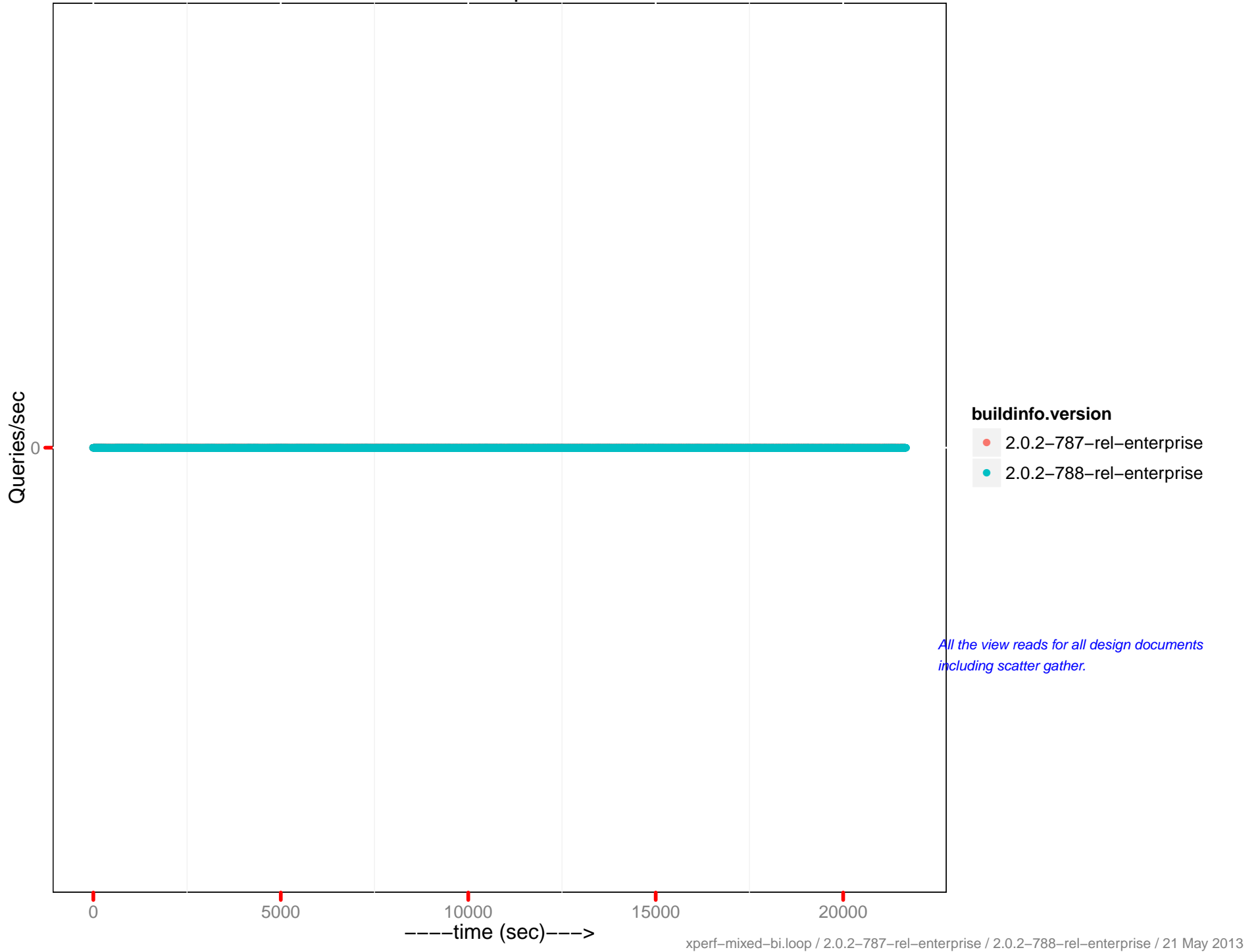
xperf-mixed-bi.loop : 2.0.2-787-rel-enterprise : 2.0.2-788-rel-enterprise



	<b>2.0.2 – 787</b>	<b>2.0.2 – 788</b>
<i>Runtime (in hr)</i>	6.03	6.05
<i>Avg. Drain Rate</i>	12.55K	10.28K
<i>Peak Disk (GB)</i>	80.58	75.63
<i>Peak Memory (GB)</i>	70124.33	76220.14
<i>Avg. OPS</i>	17.48K	13.33K
<i>Avg. mem memcached (GB)</i>	57154.24	57443.51
<i>Avg. mem beam.smp (MB)</i>	918218.52	931545.69
<i>Avg. CPU rate (%)</i>	82.5	83.11
<i>Latency-get (90th) (ms)</i>	7.13	7.85
<i>Latency-get (95th) (ms)</i>	9.67	10.88
<i>Latency-get (99th) (ms)</i>	22.34	23.72
<i>Latency-set (90th) (ms)</i>	7.35	7.95
<i>Latency-set (95th) (ms)</i>	9.89	10.89
<i>Latency-set (99th) (ms)</i>	22.53	23.48
<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>	5342.18	2680.05
<i>Avg. XDC docs to replicate</i>	80163.17	1477713.84
<i>Rebalance Time (sec)</i>	0	0
<i>Testrunner Version</i>	26873dc	f98ef42



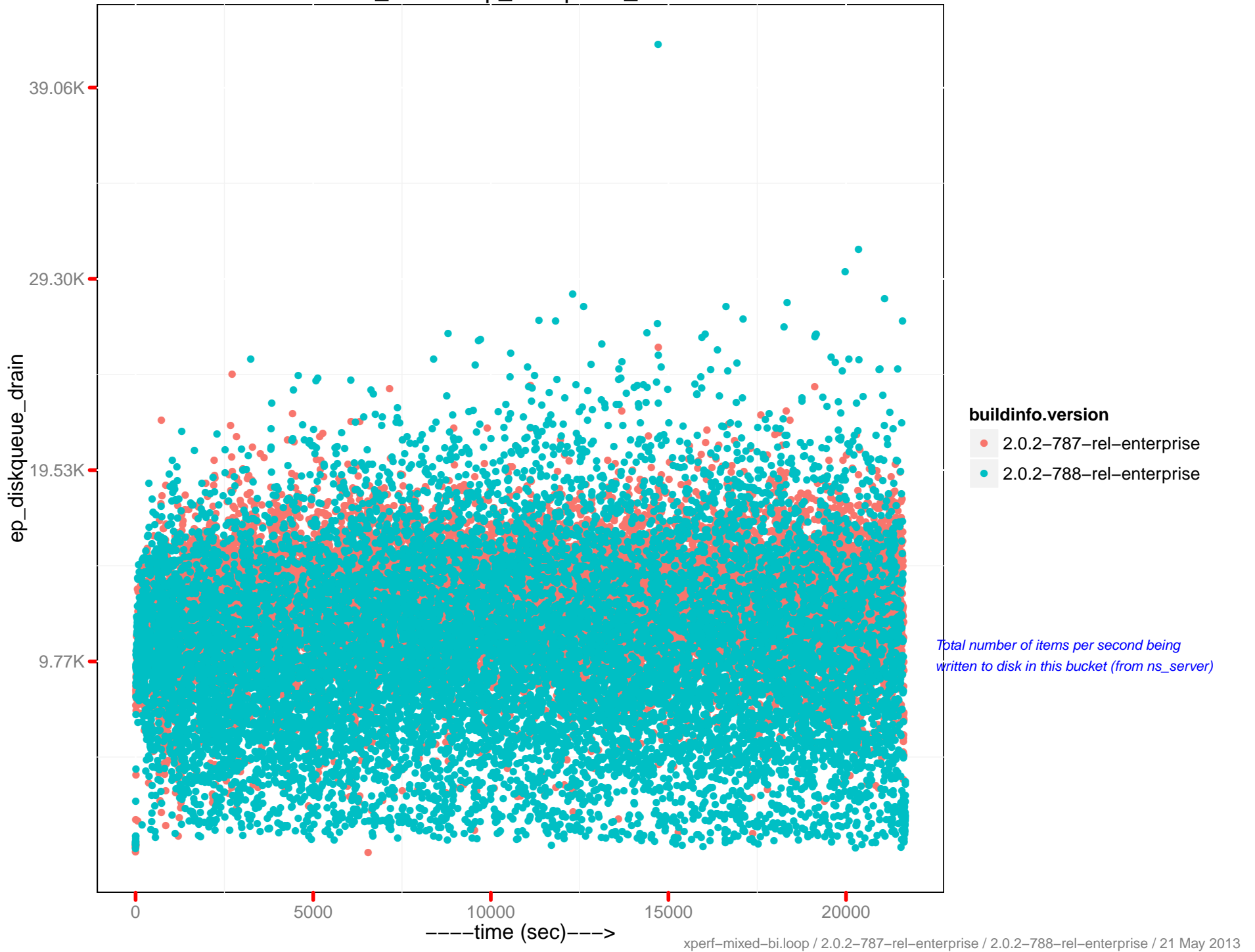
# View read per sec.



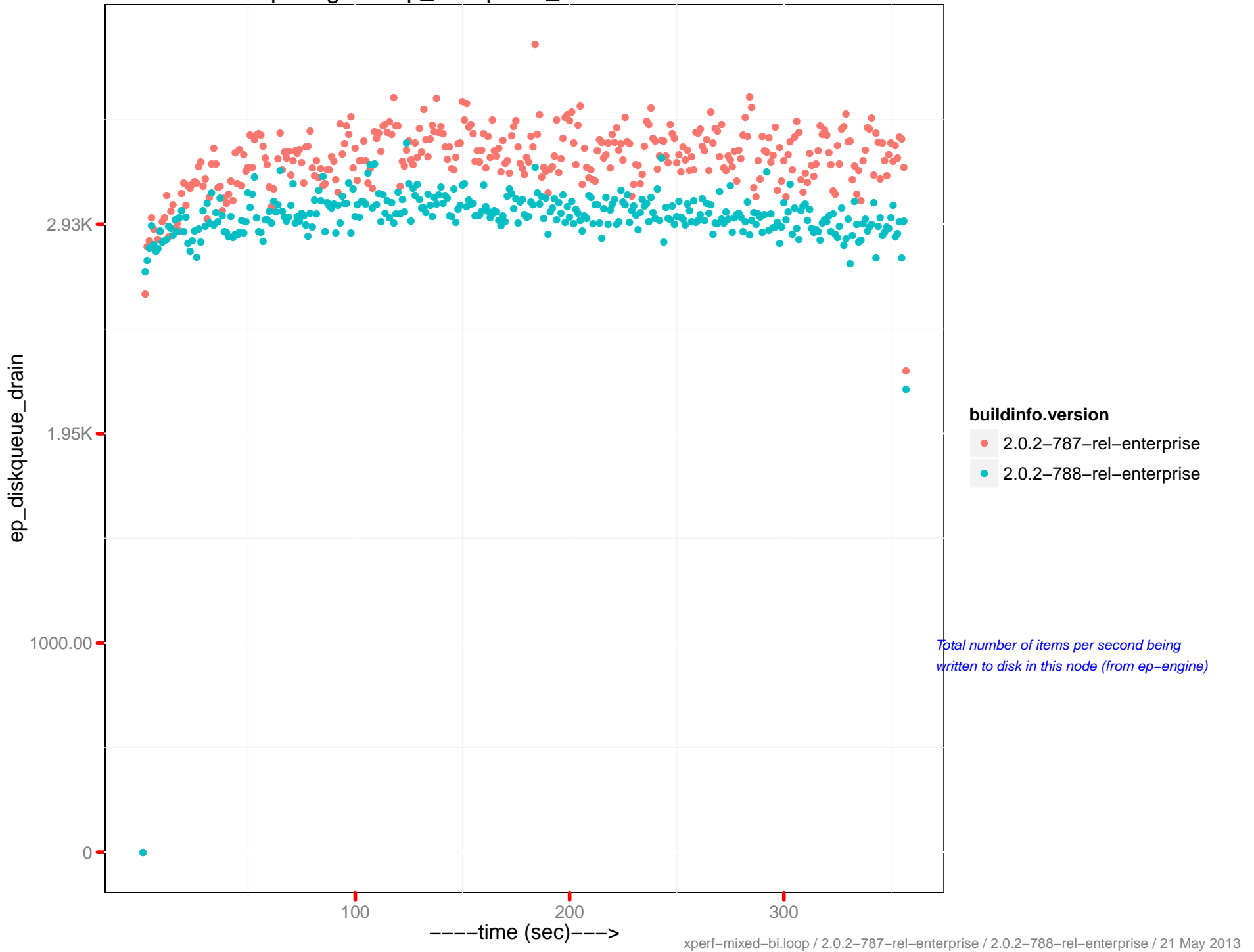
ep queue size



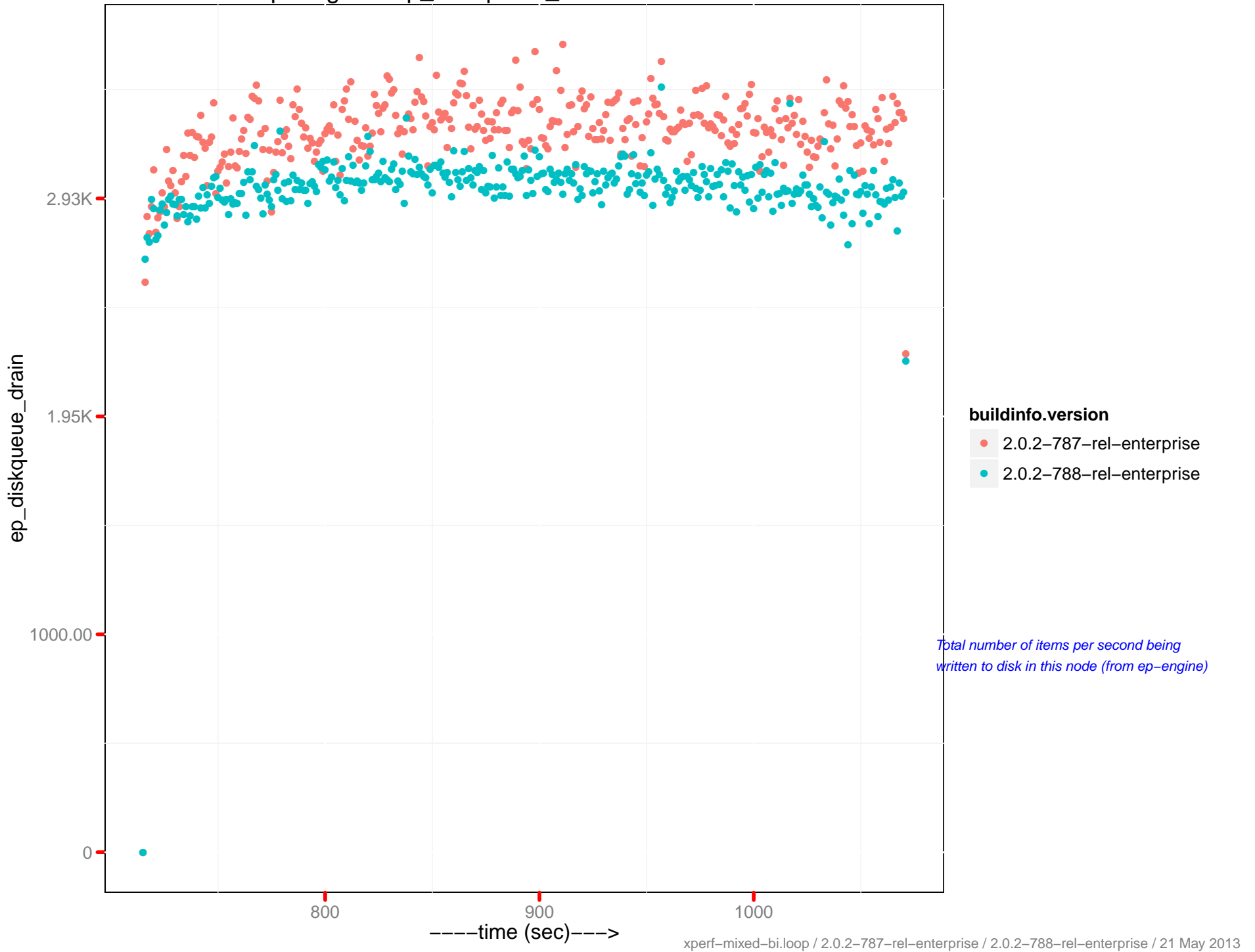
# ns\_server: ep\_diskqueue\_drain



# ep-engine : ep\_diskqueue\_drain - 172.23.97.53

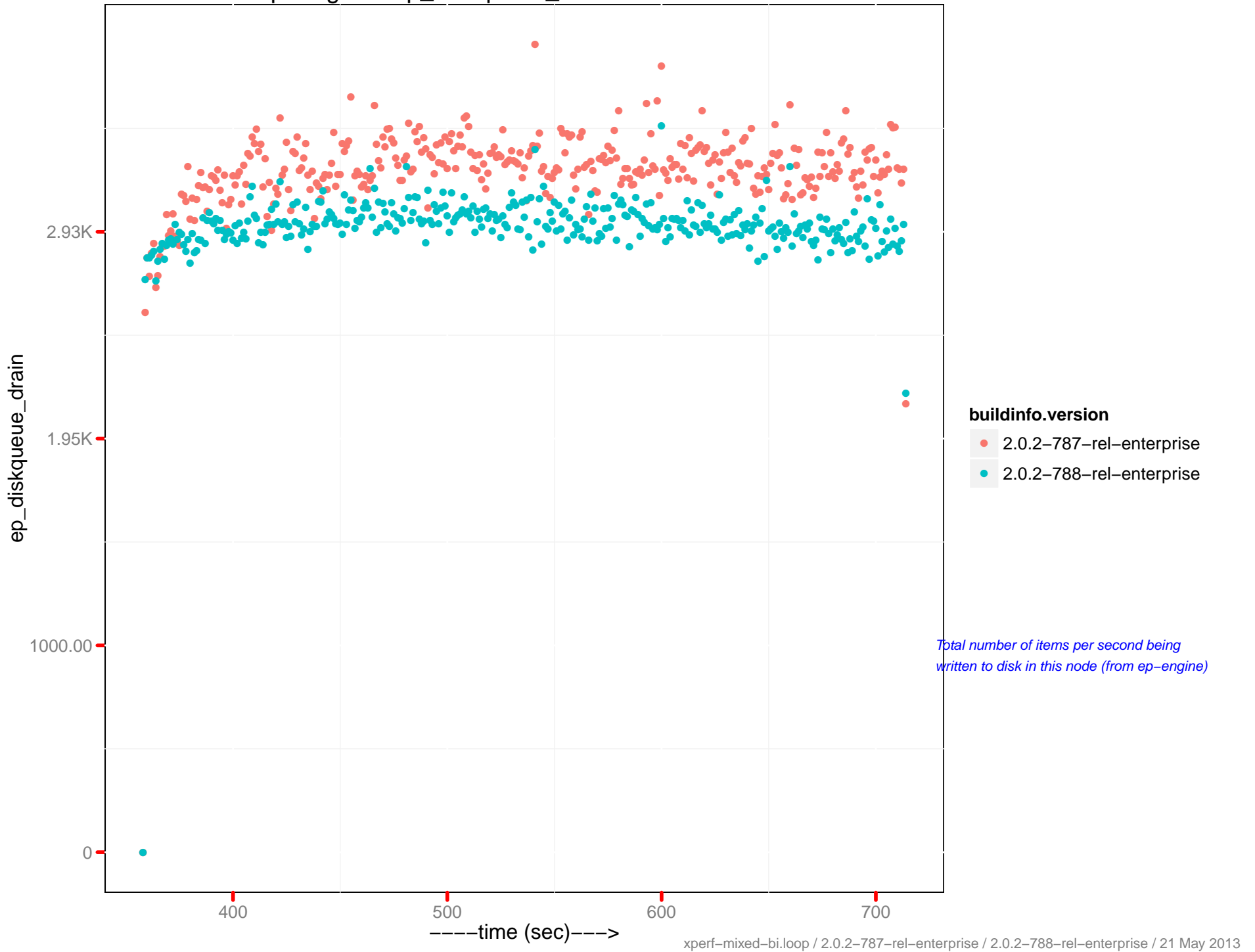


ep-engine : ep\_diskqueue\_drain - 172.23.97.54

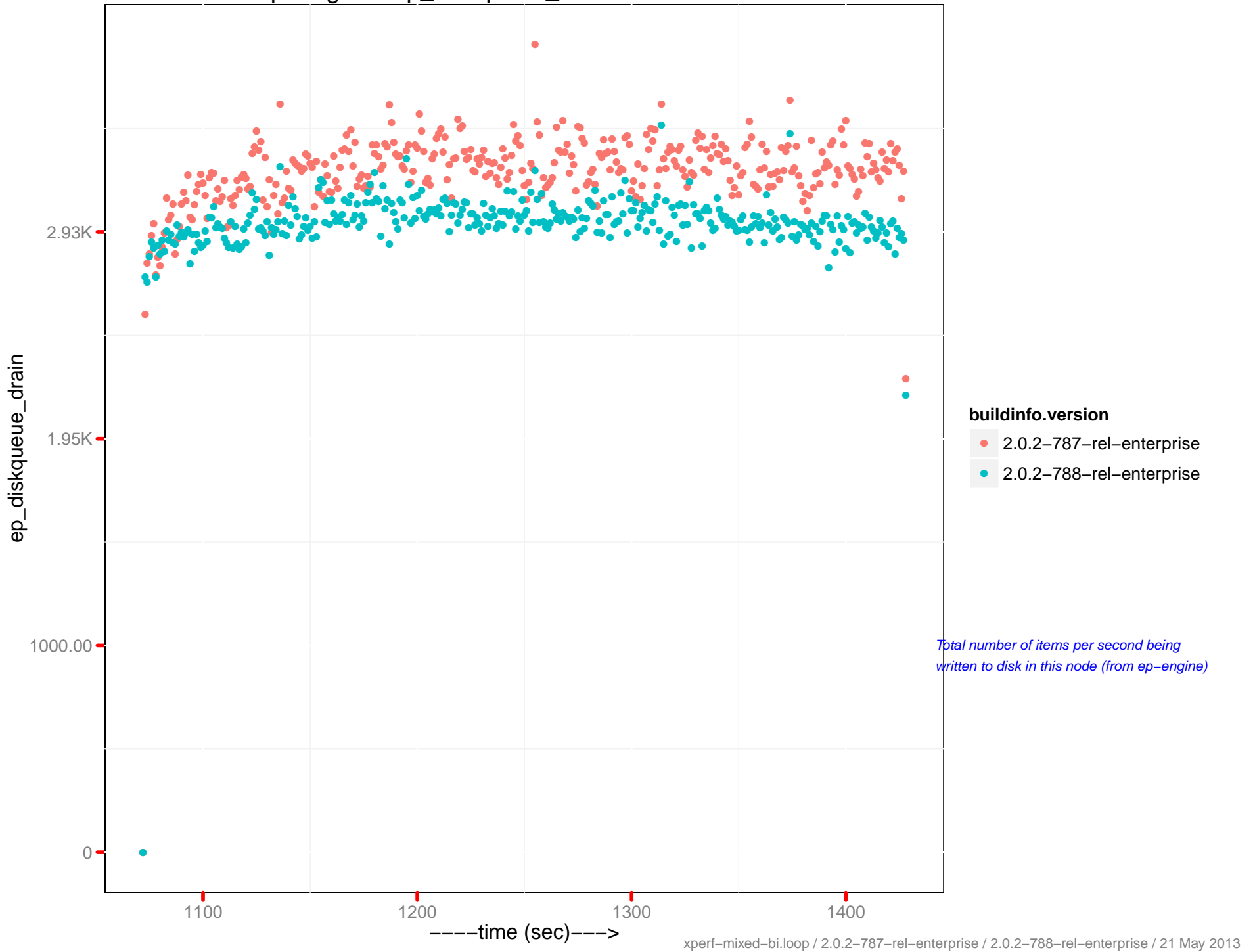




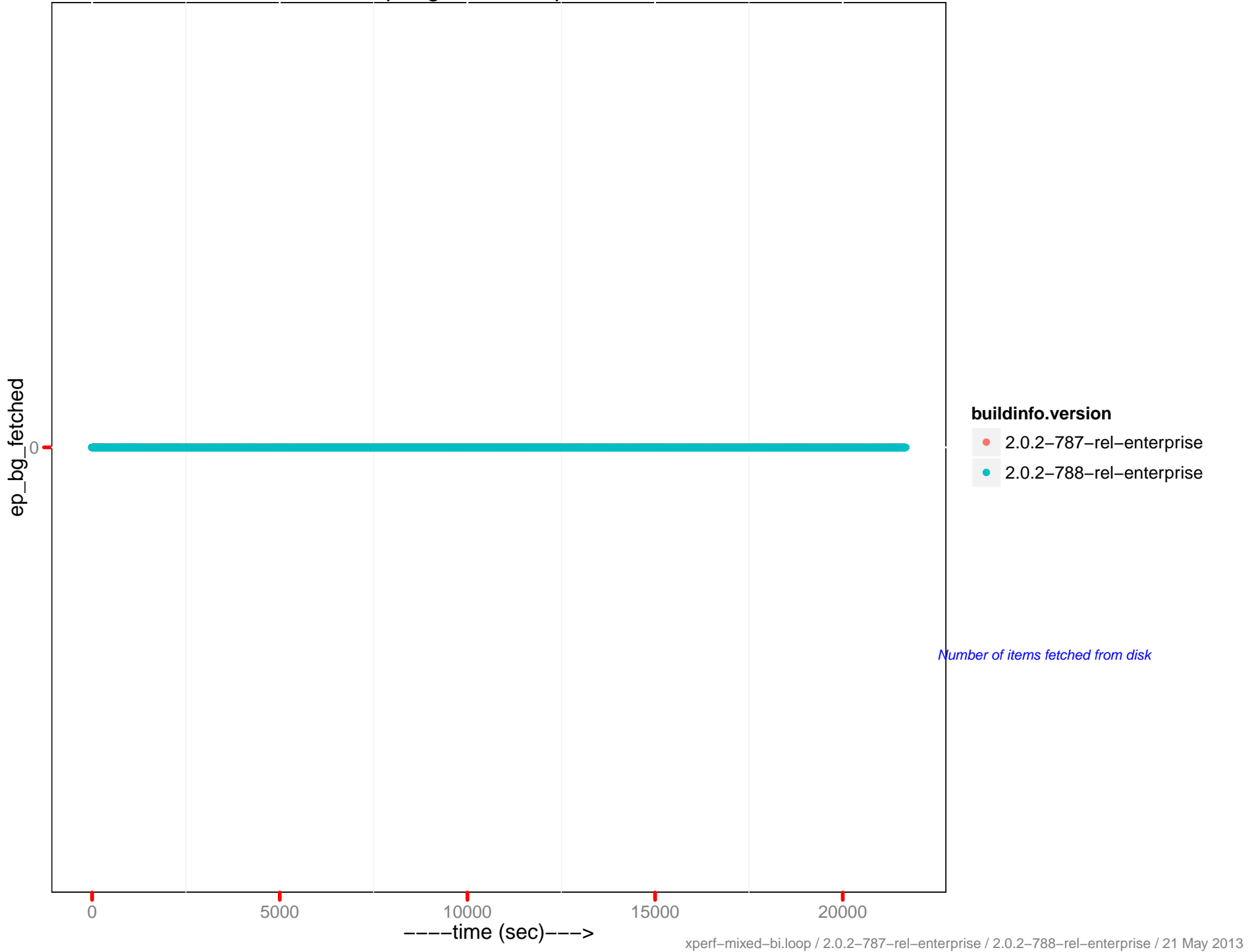
ep-engine : ep\_diskqueue\_drain - 172.23.97.55



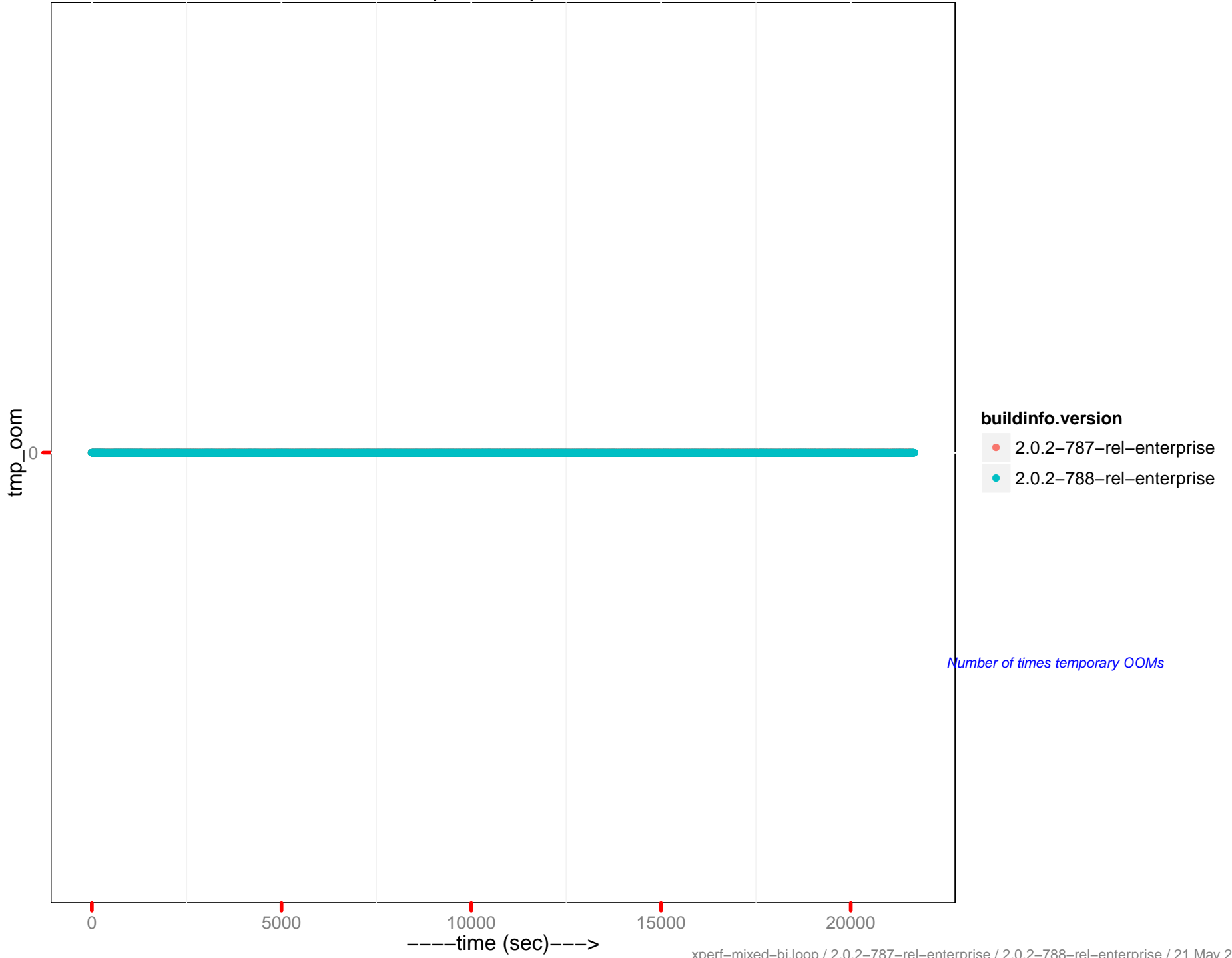
# ep-engine : ep\_diskqueue\_drain - 172.23.97.56



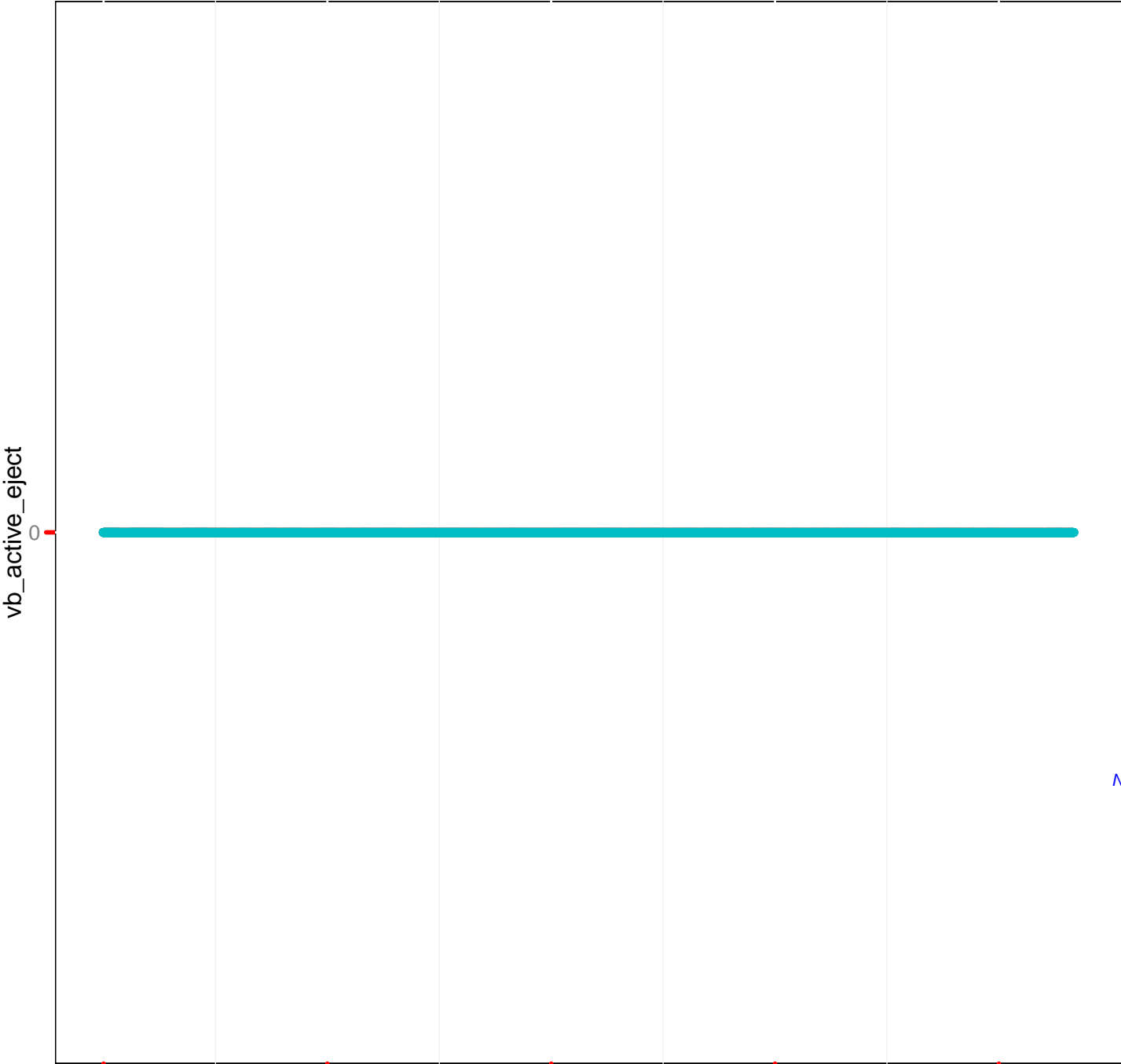
# ep\_bg\_fetched ops/sec



# tmp\_oom ops/sec



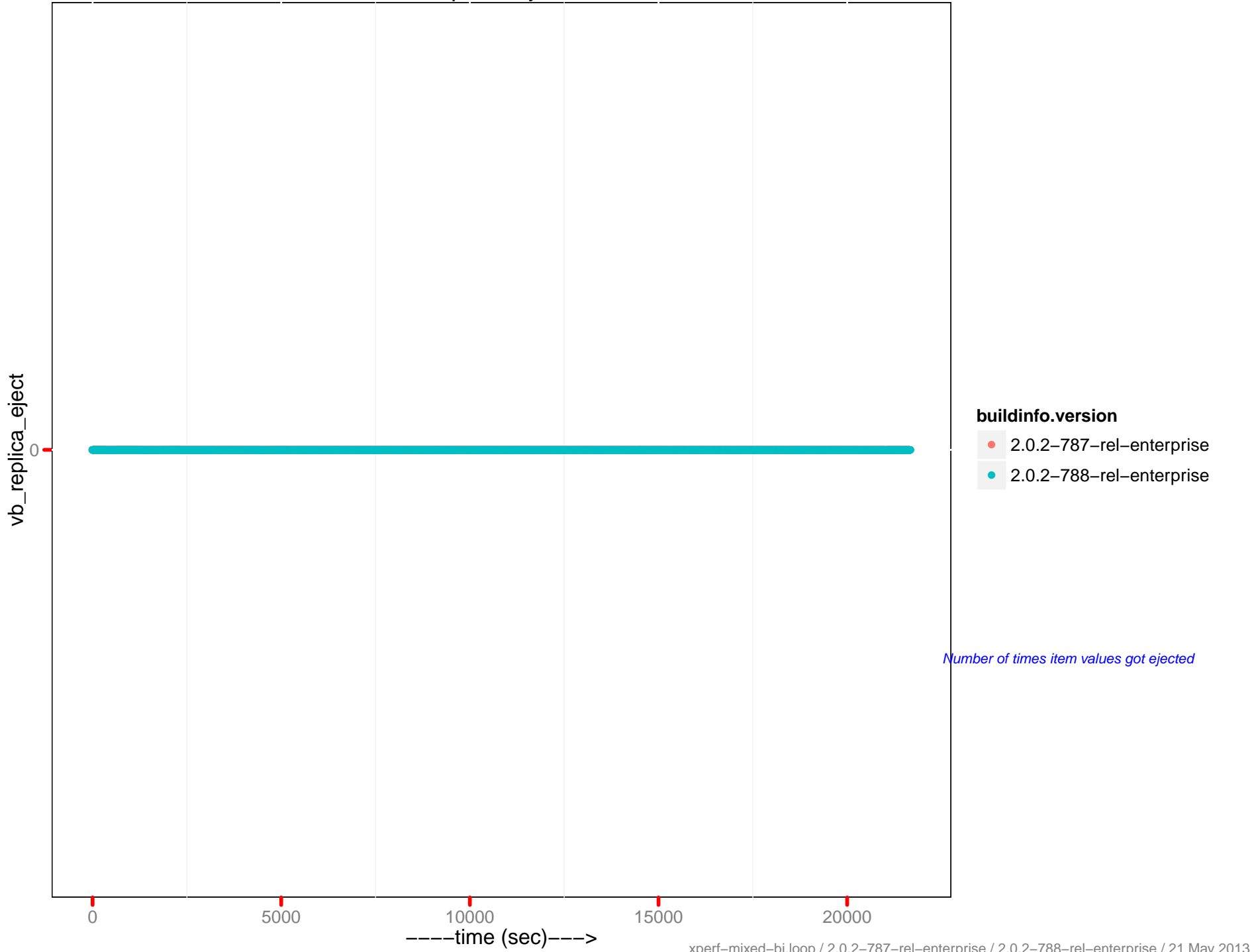
vb\_active\_eject/sec



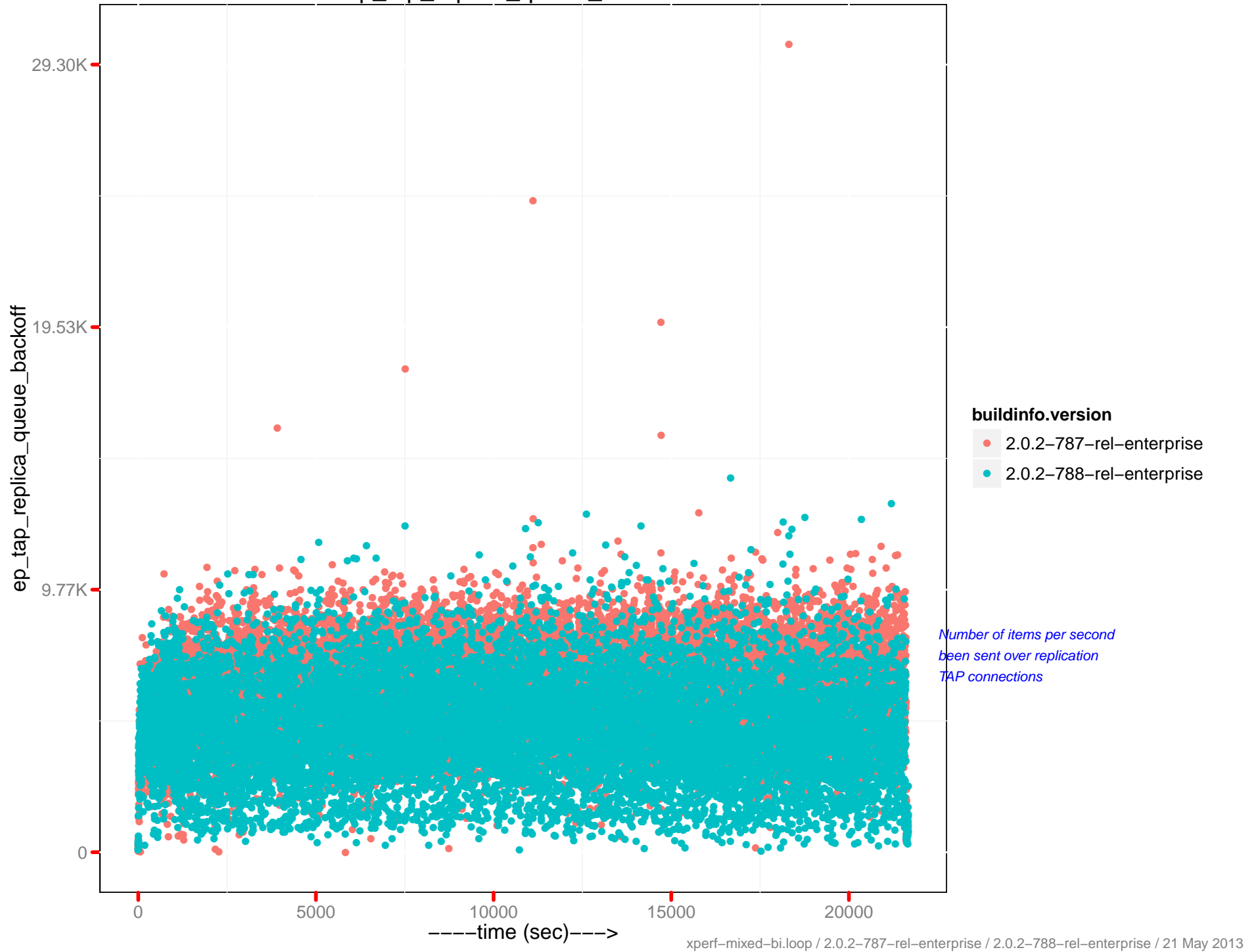
- buildinfo.version**
- 2.0.2-787-rel-enterprise
  - 2.0.2-788-rel-enterprise

*Number of times item values got ejected*

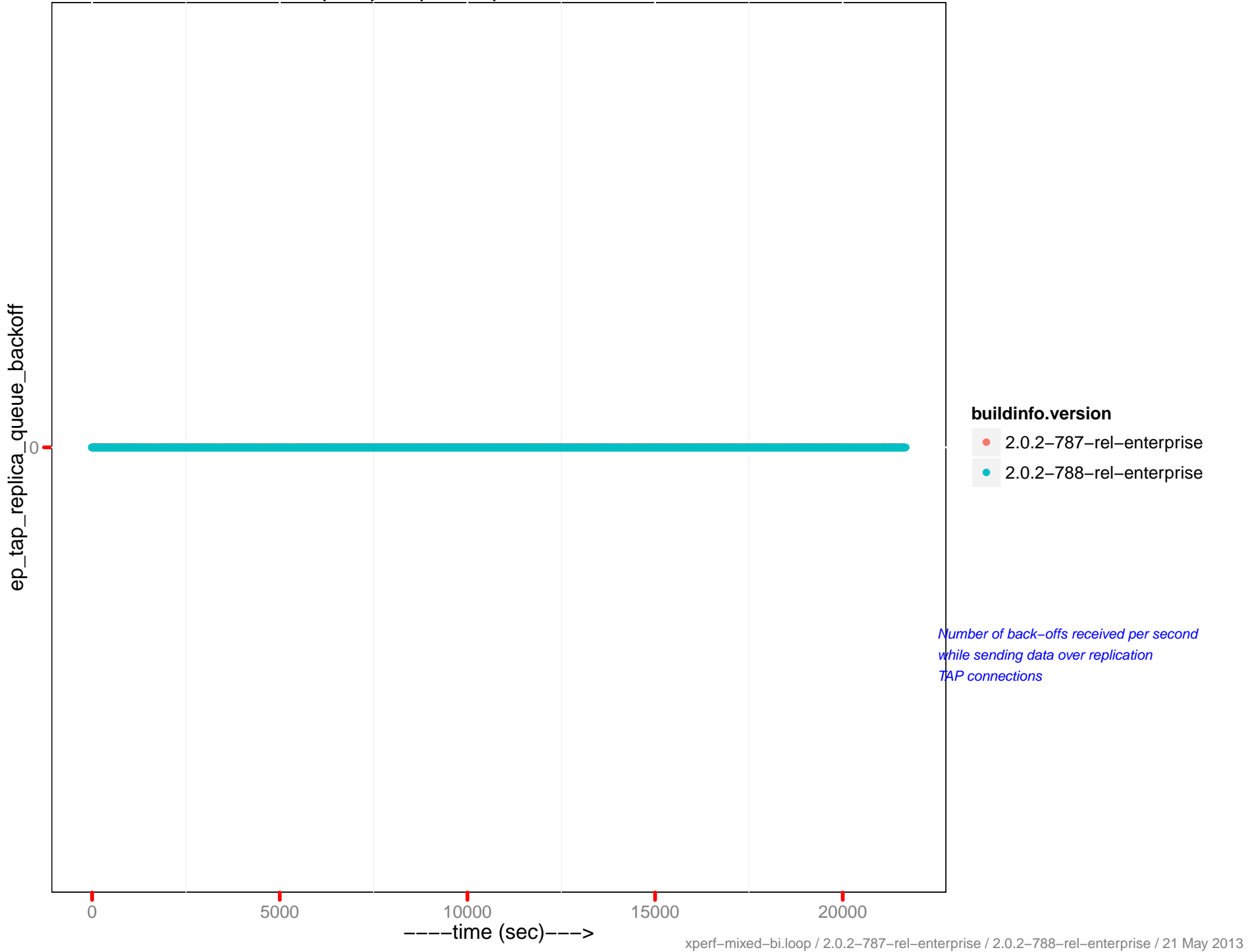
# vb\_replica\_eject/sec



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



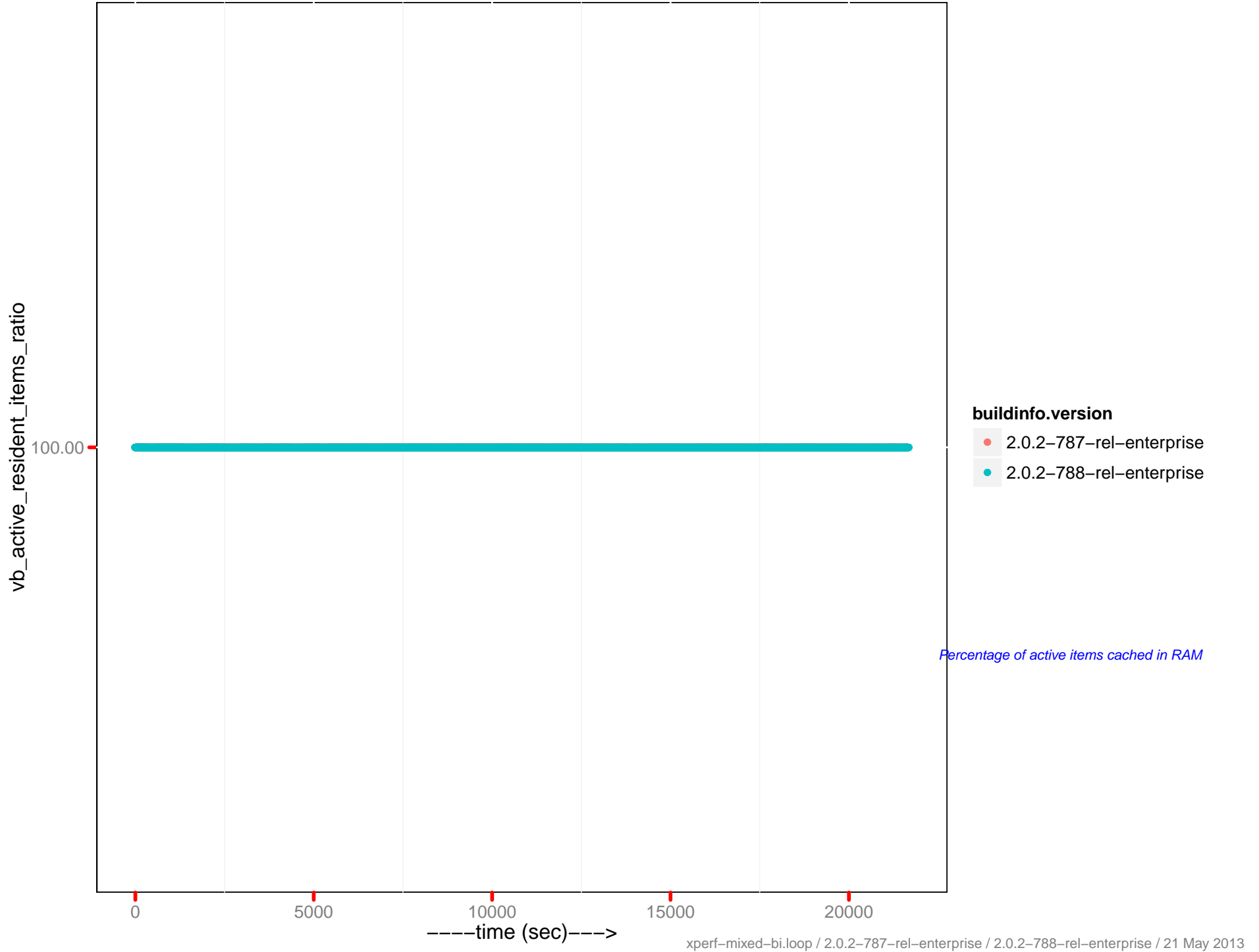
# ep\_tap\_replica\_queue\_backoff/sec



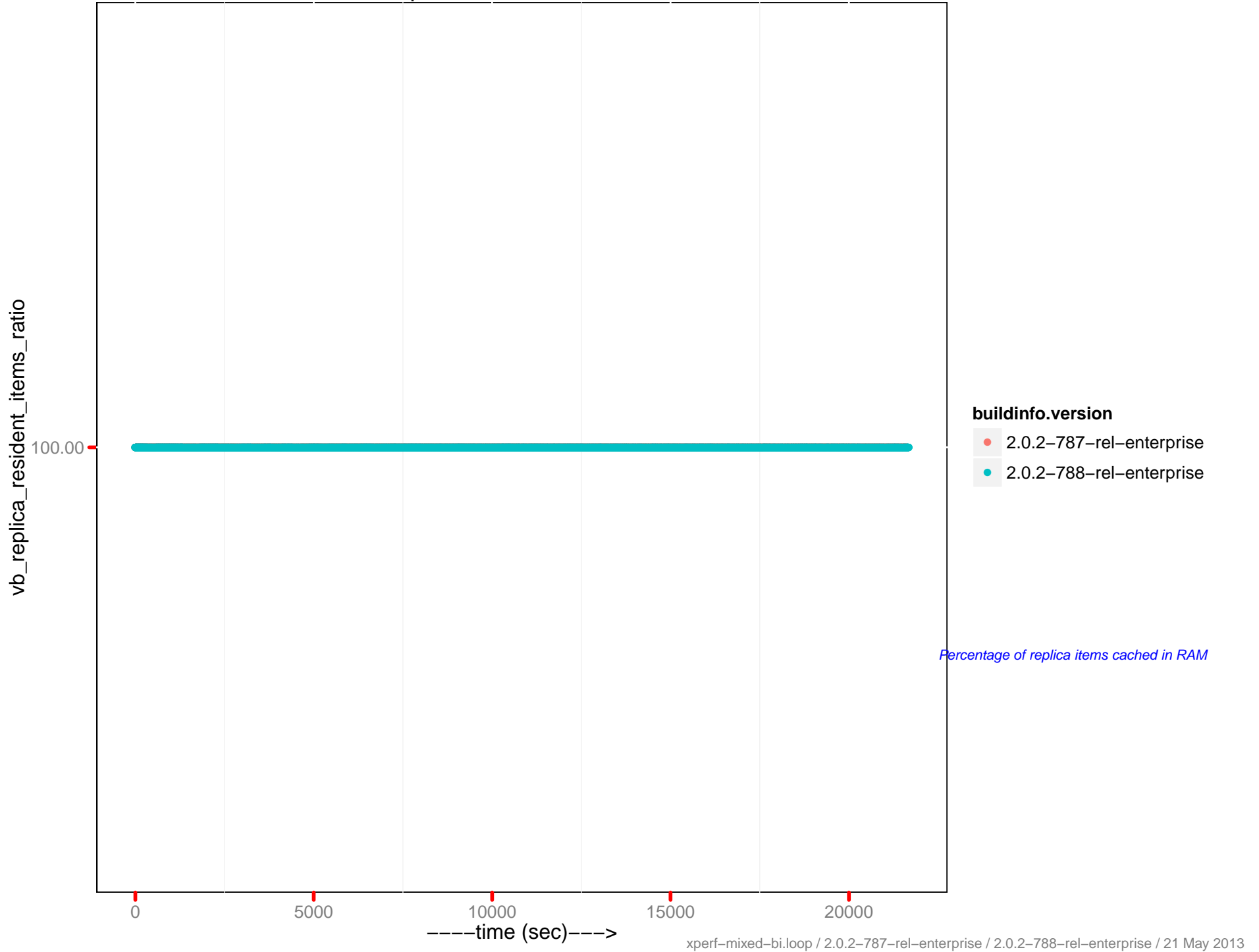
*Number of back-offs received per second while sending data over replication TAP connections*



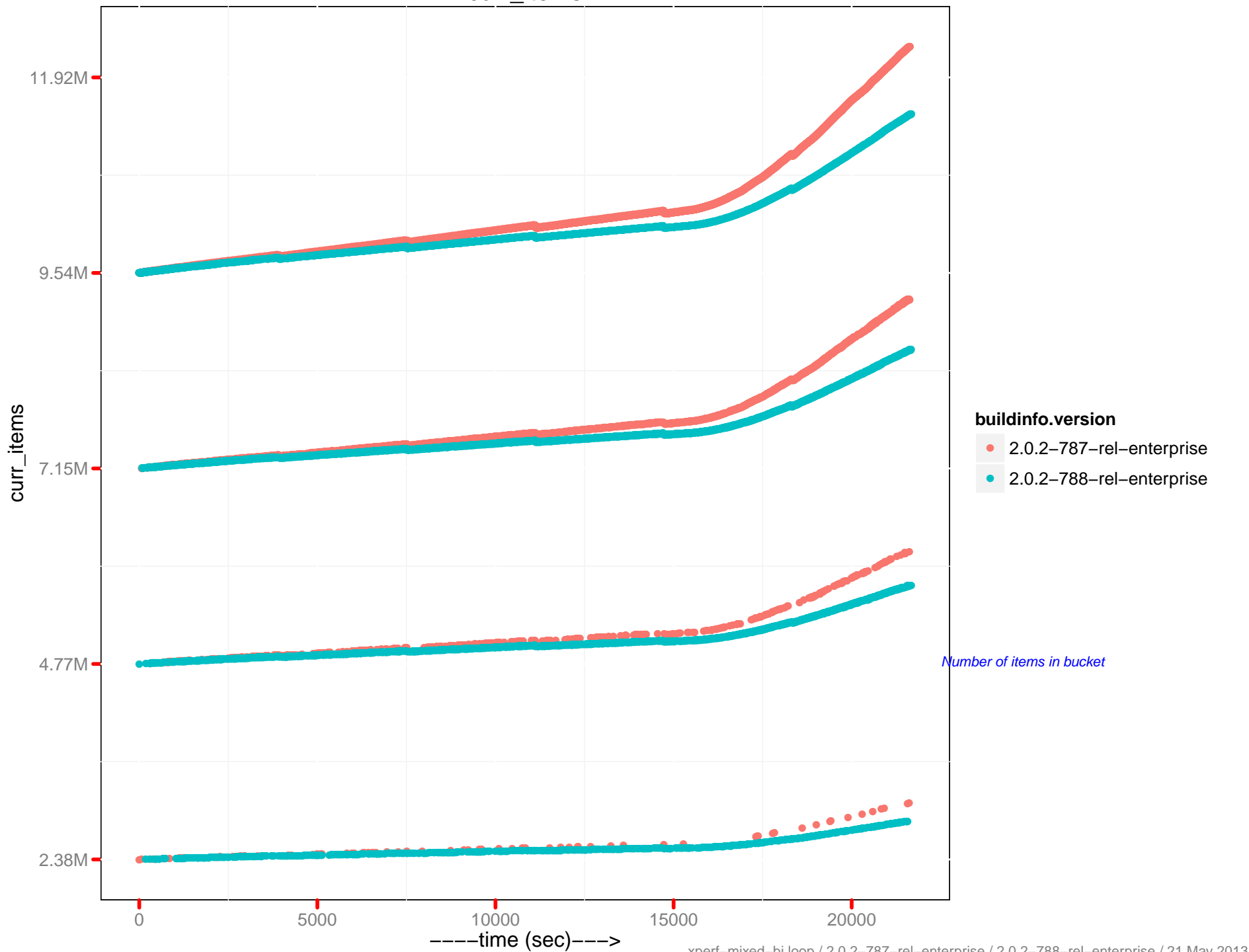
# vb\_active\_resident\_items\_ratio



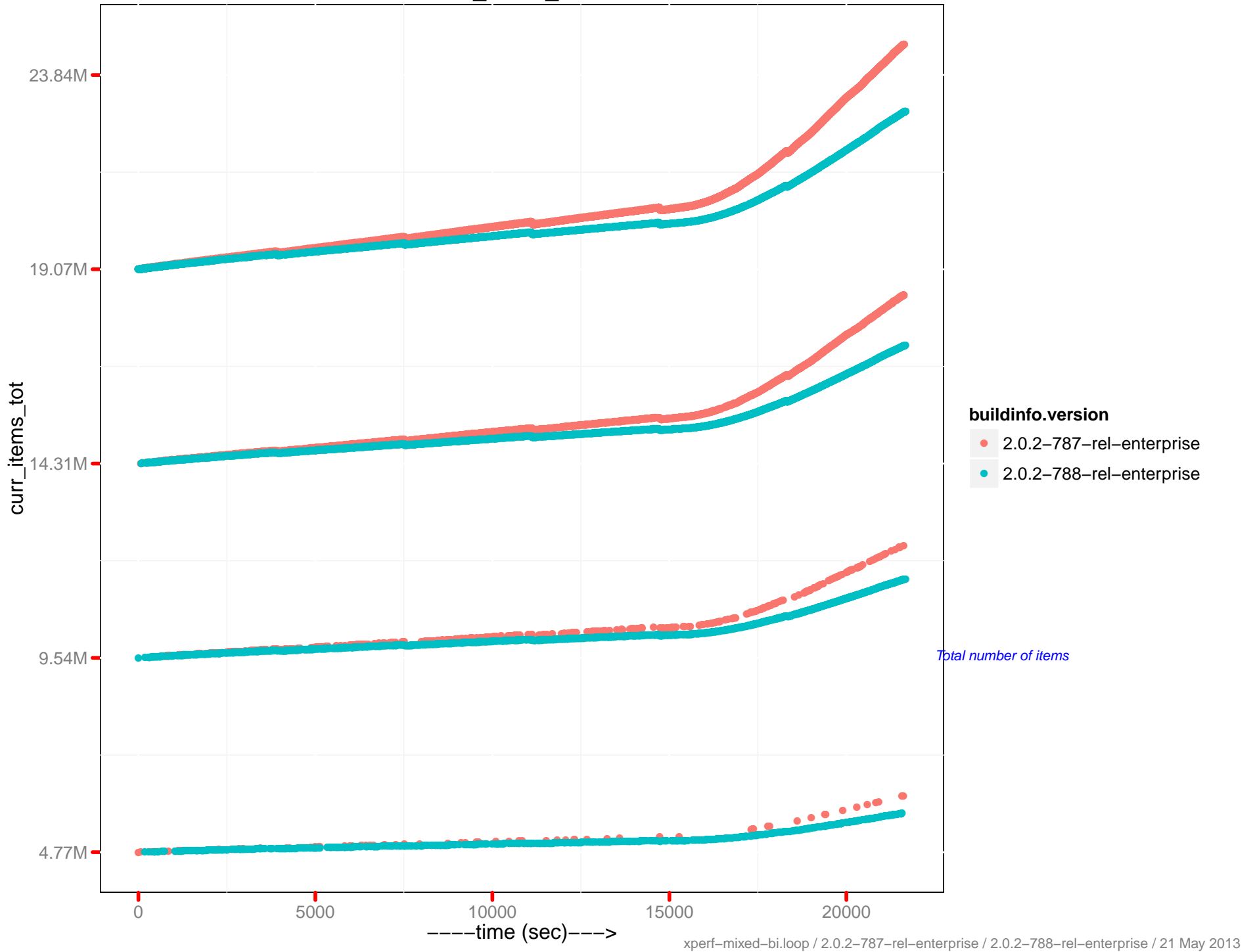
# vb\_replica\_resident\_items\_ratio



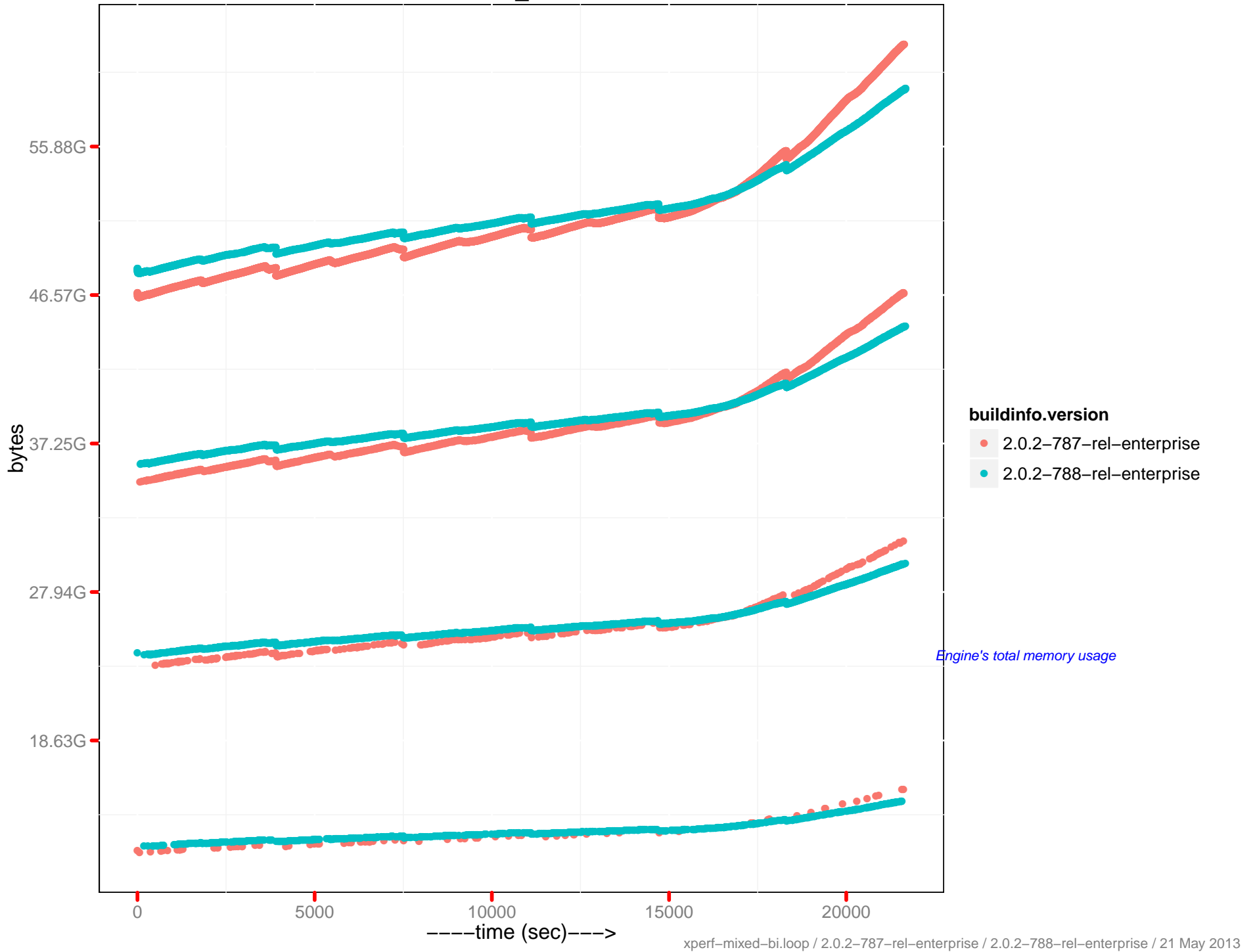
curr\_items



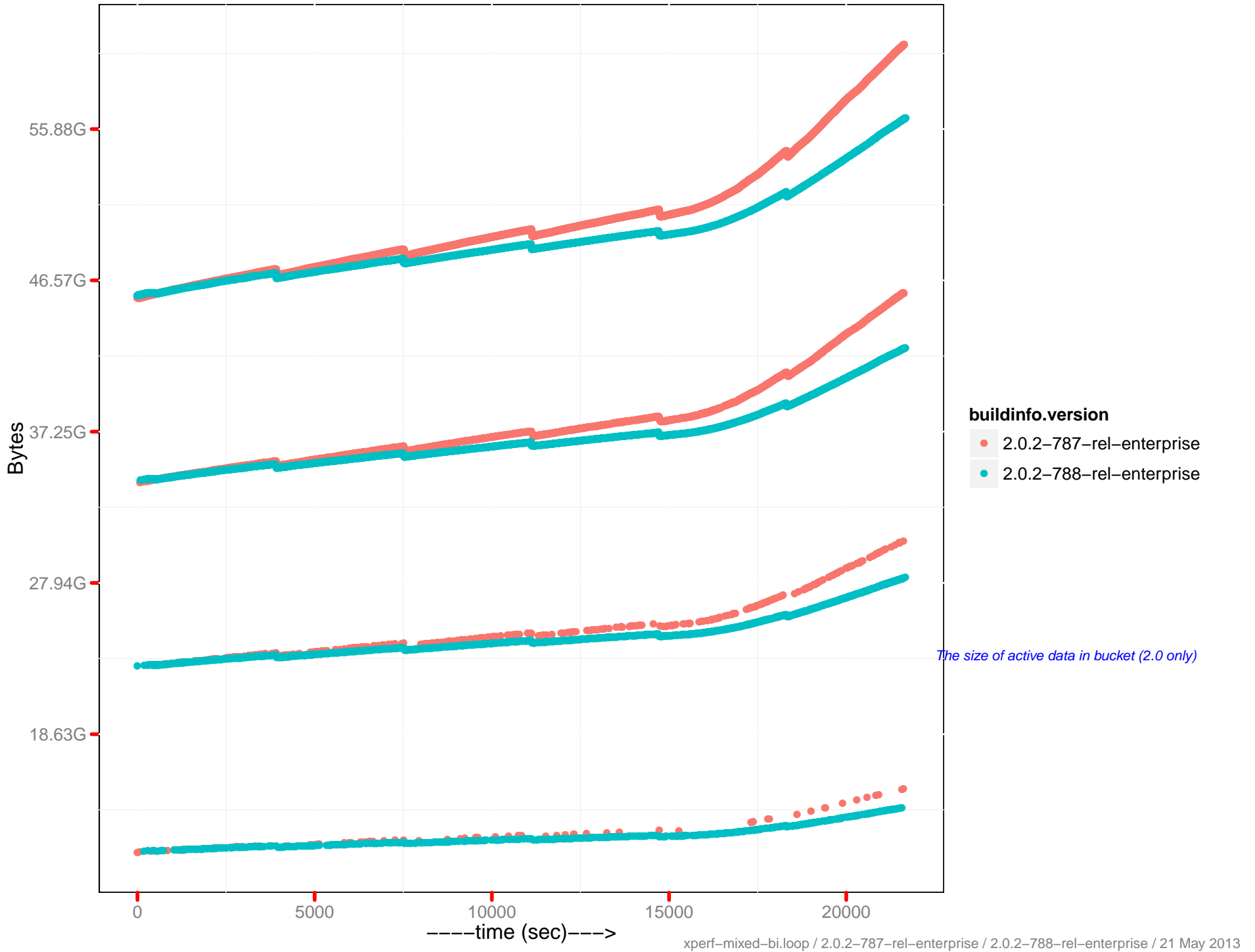
cur\_items\_total



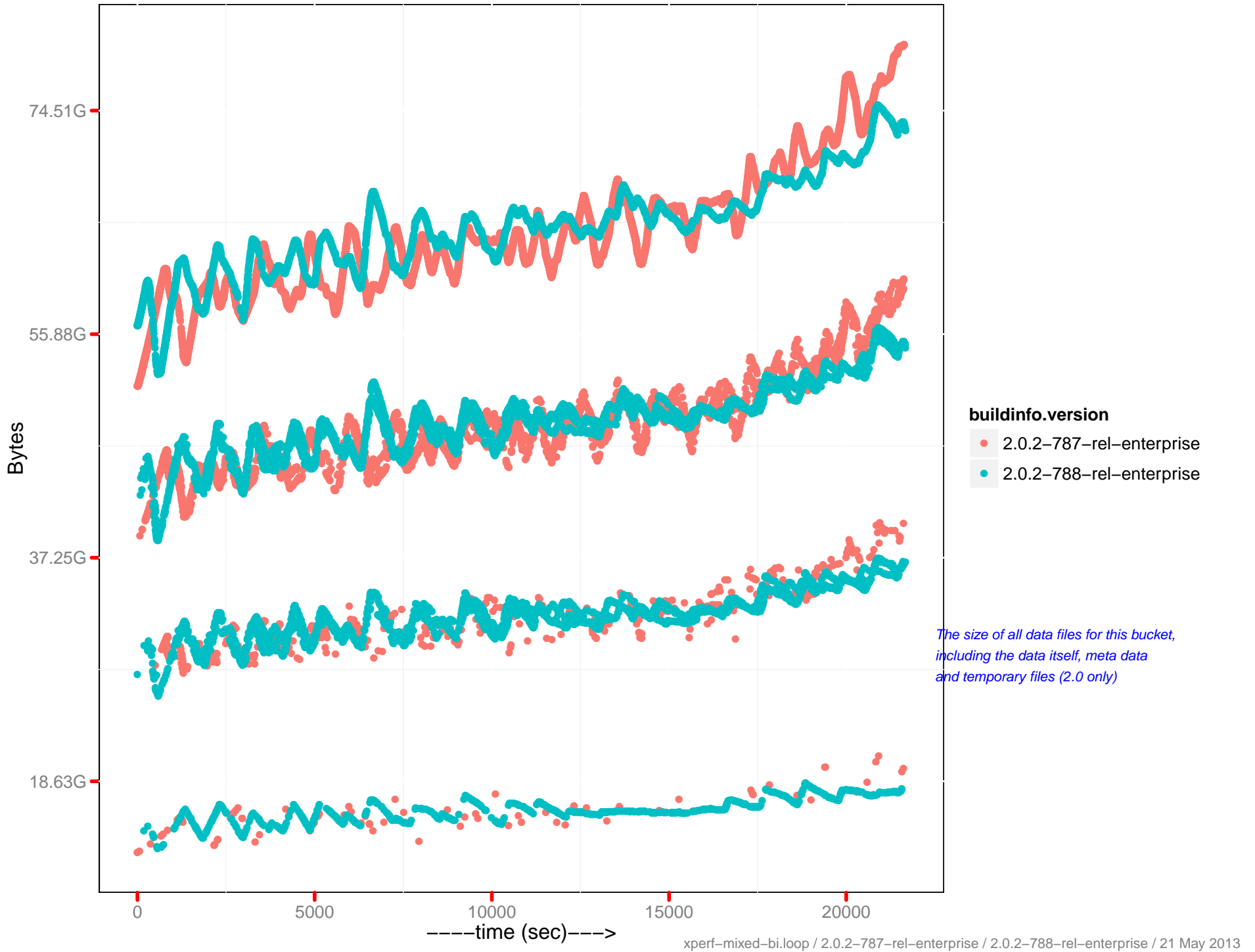
mem\_used



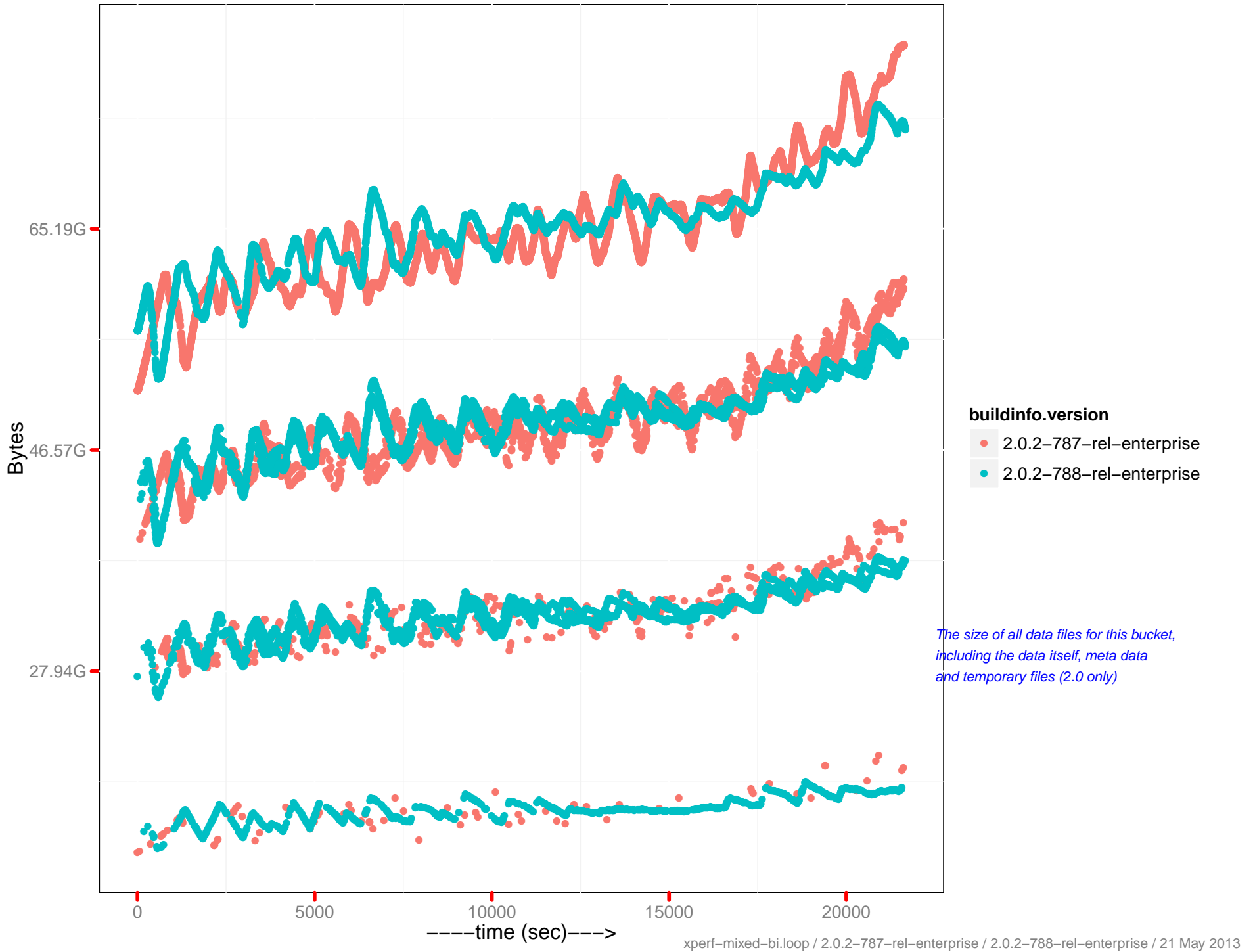
# Docs data size



# Docs disk size

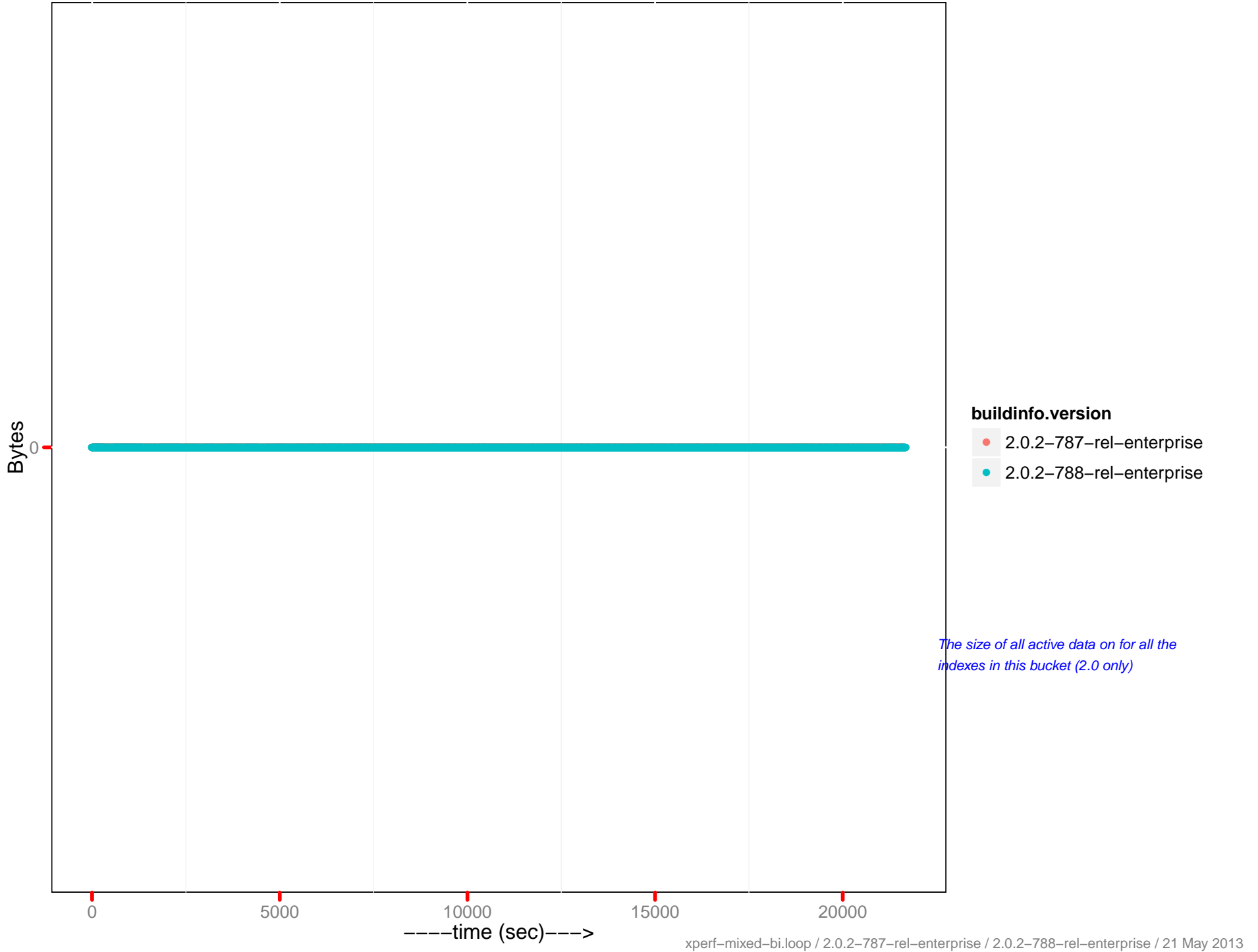


# Docs actual disk size

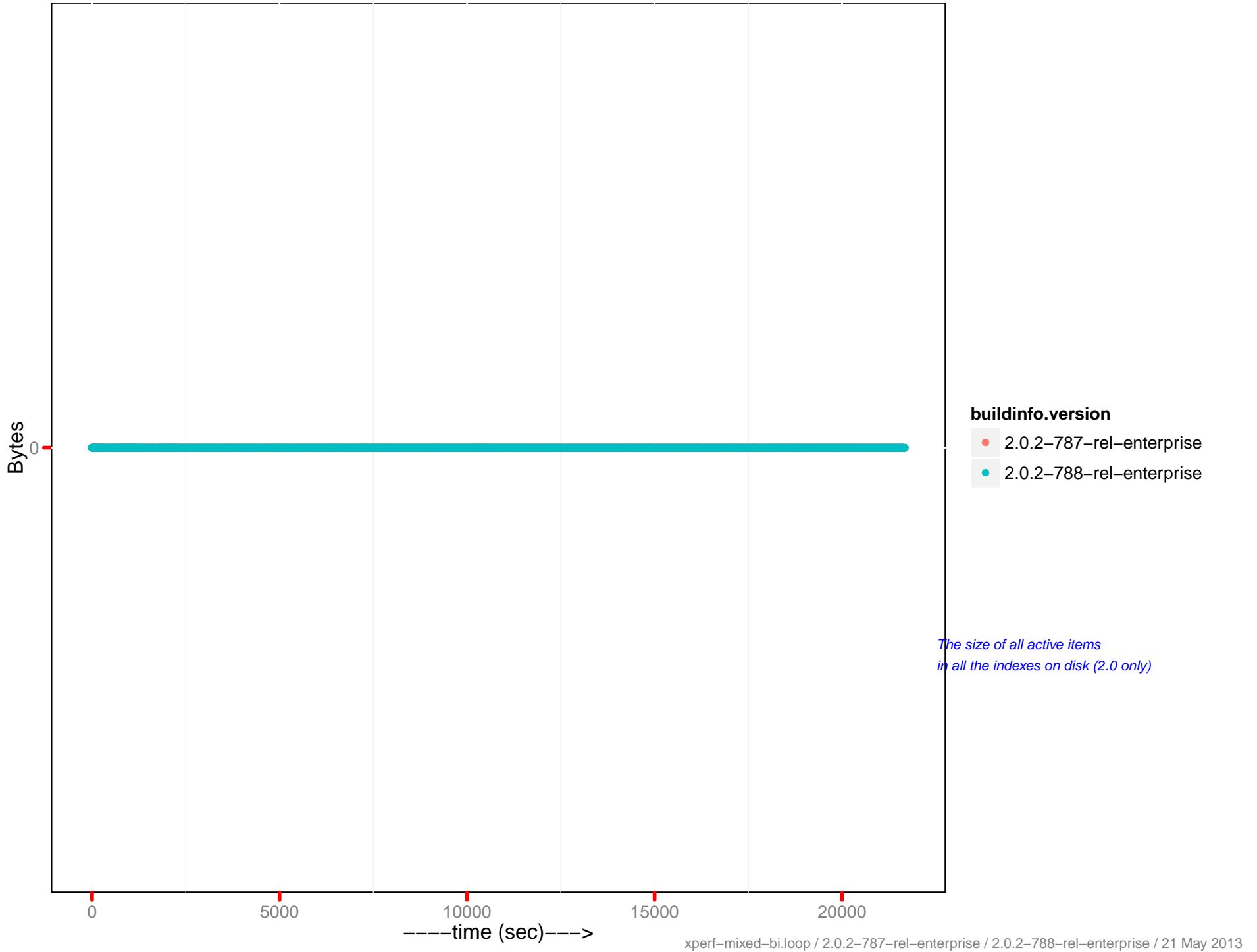




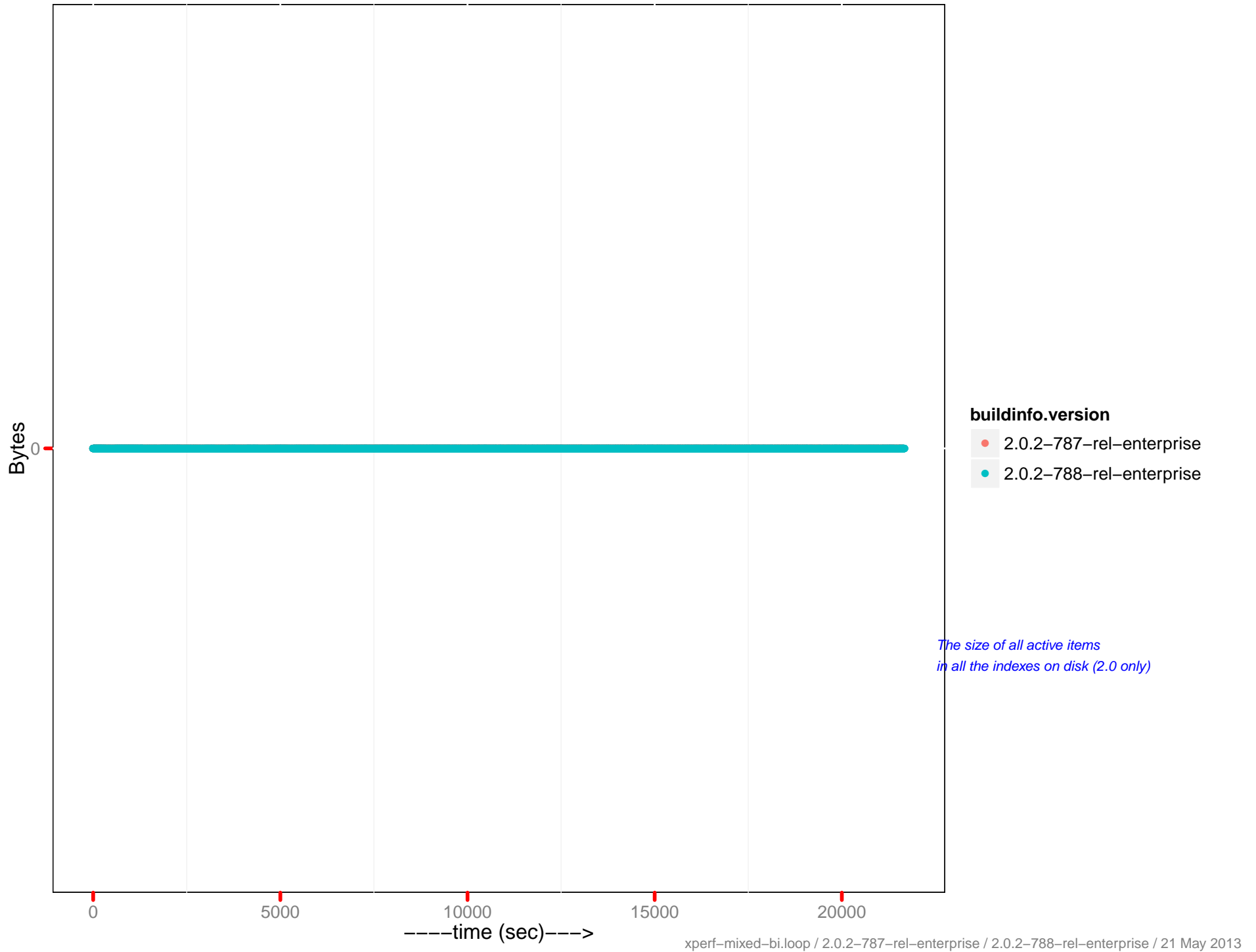
# Views data size



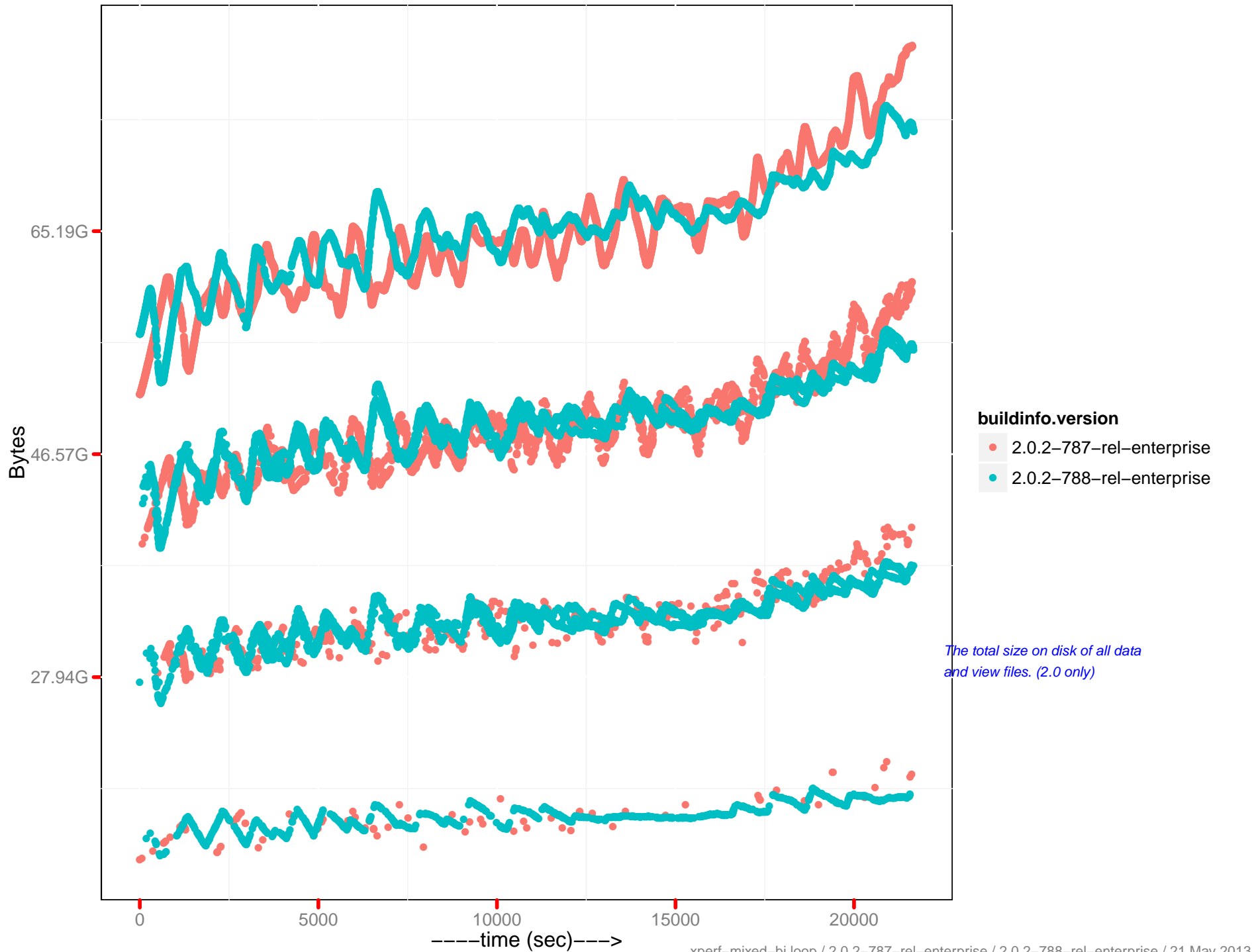
# Views disk size



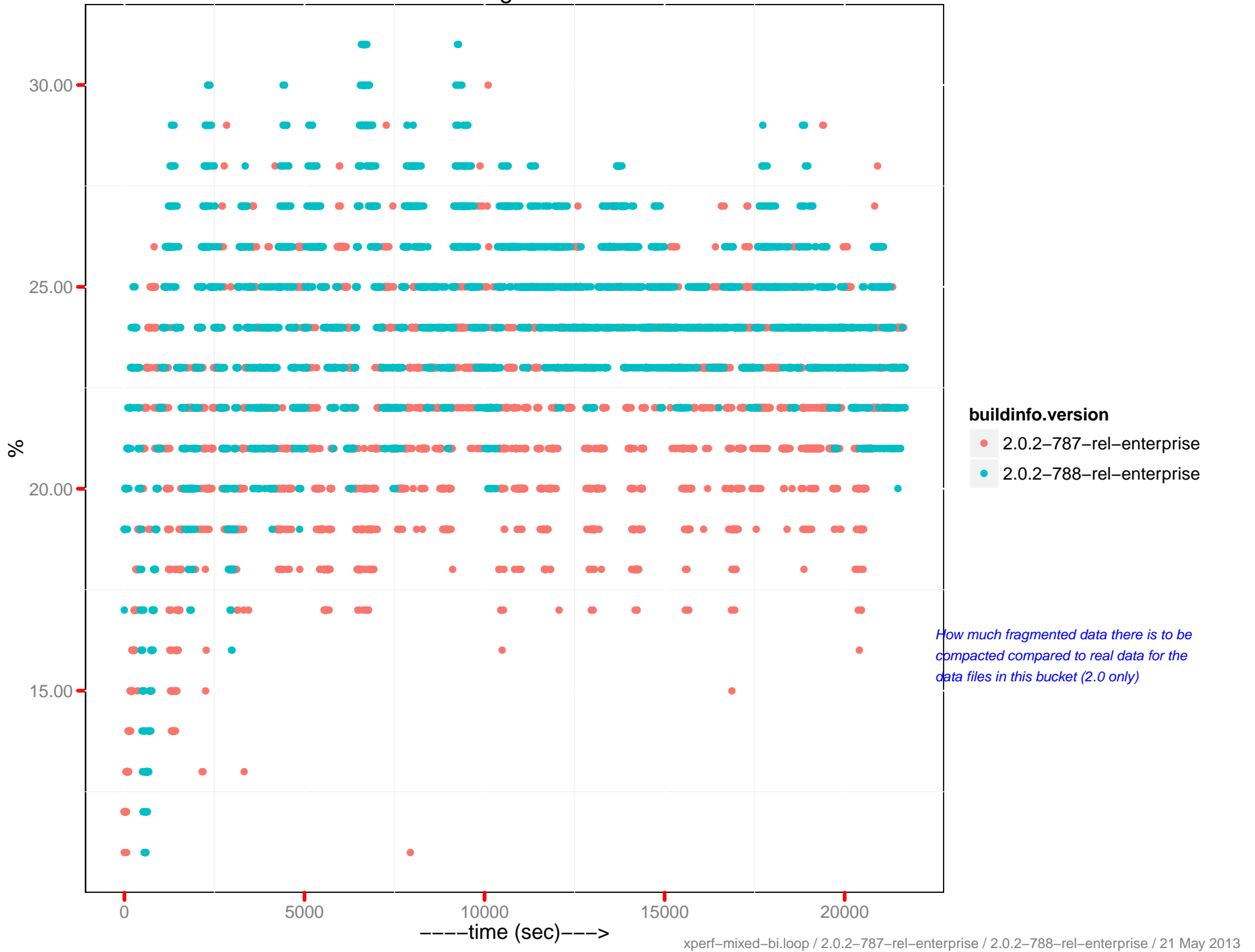
# Views actual disk size



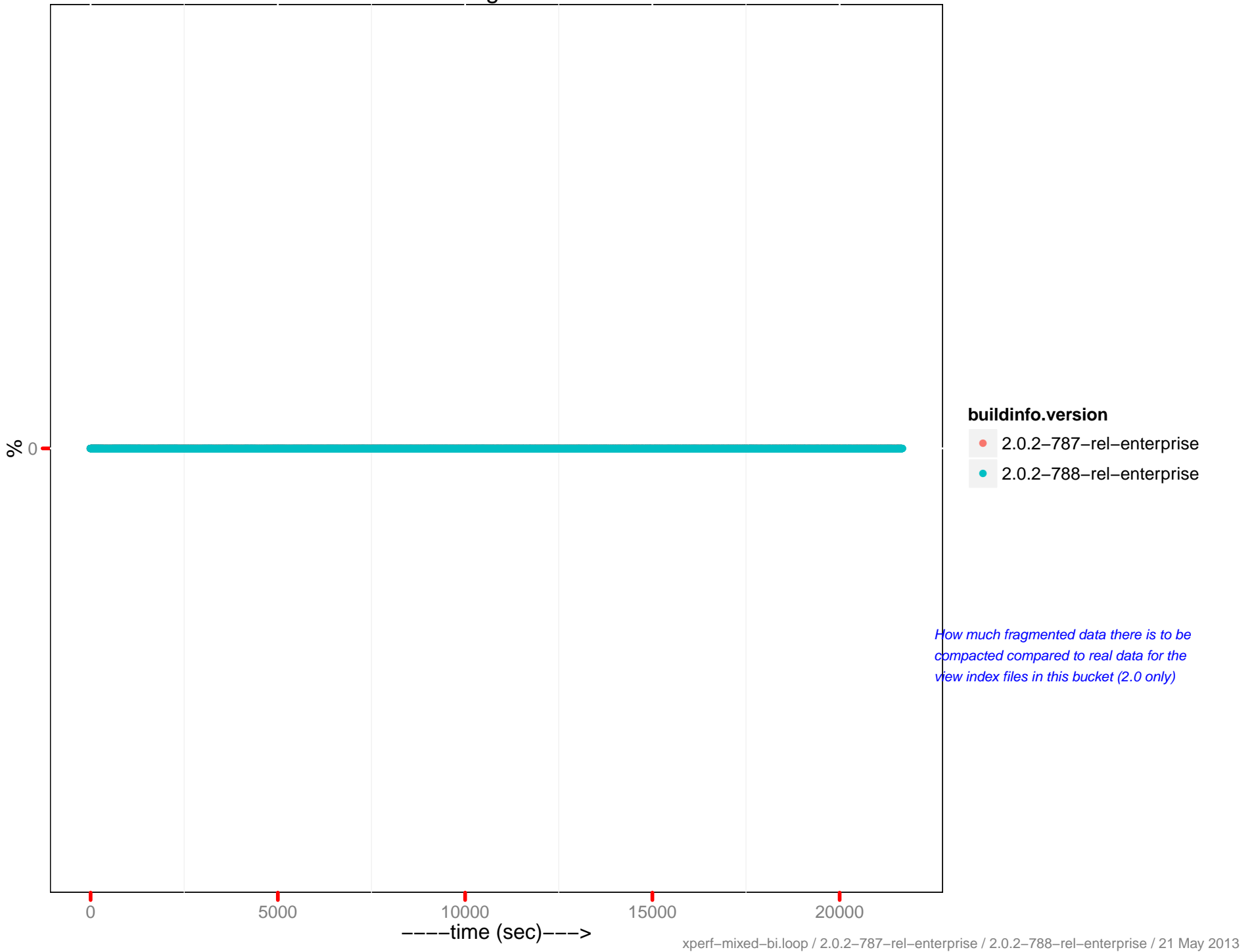
# Total disk size



# Docs fragmentation

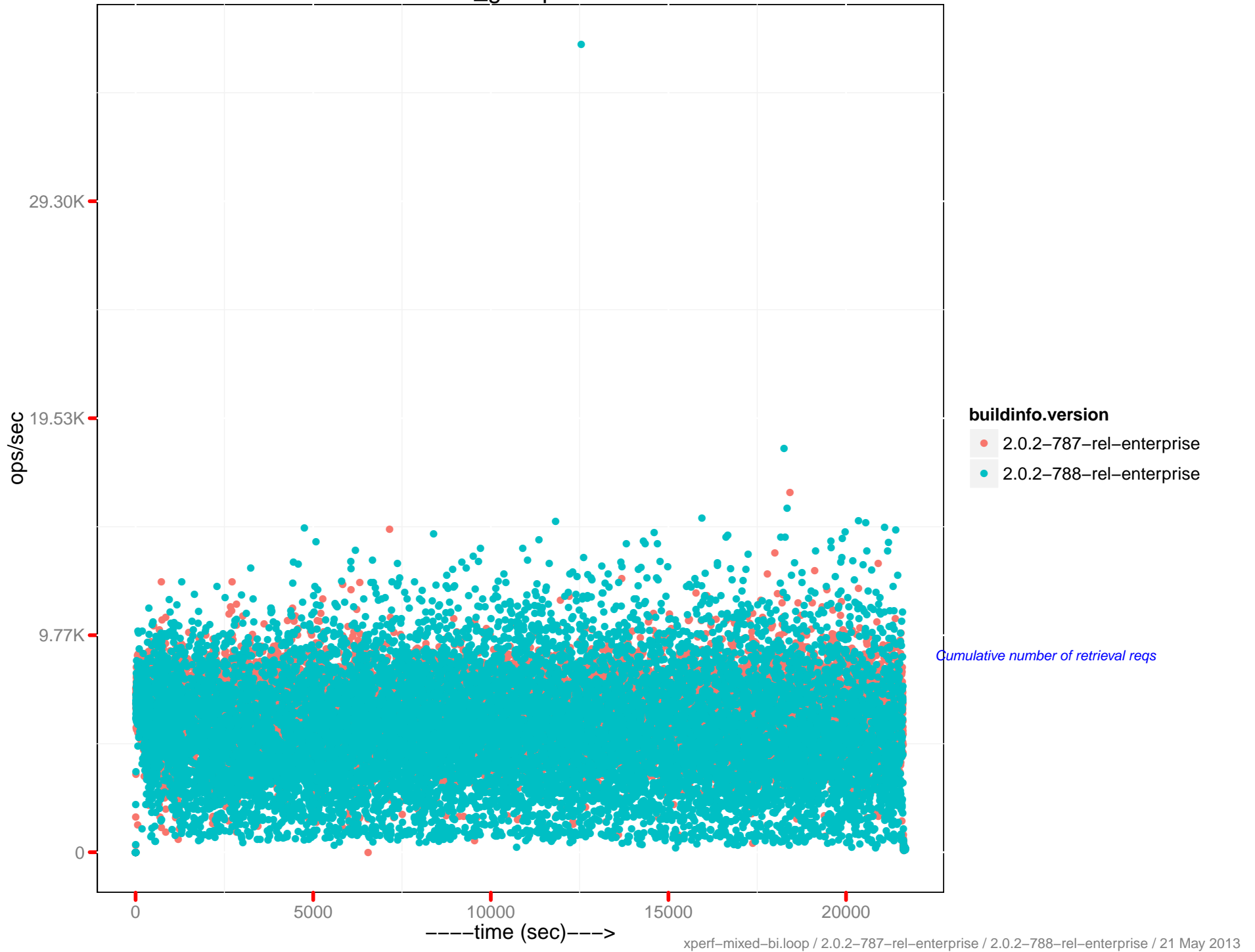


# Views fragmentation

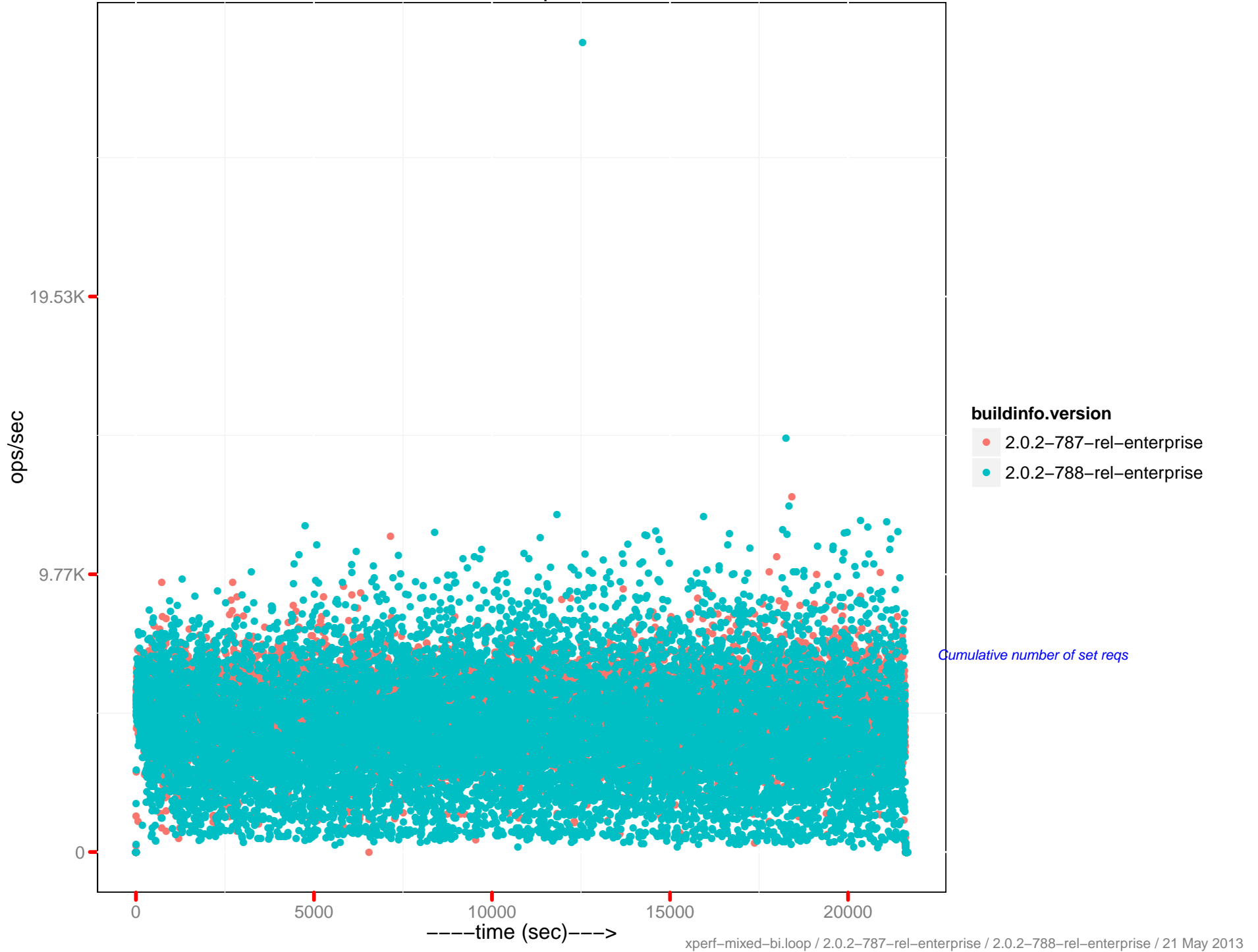


*How much fragmented data there is to be compacted compared to real data for the view index files in this bucket (2.0 only)*

# cmd\_get ops/sec

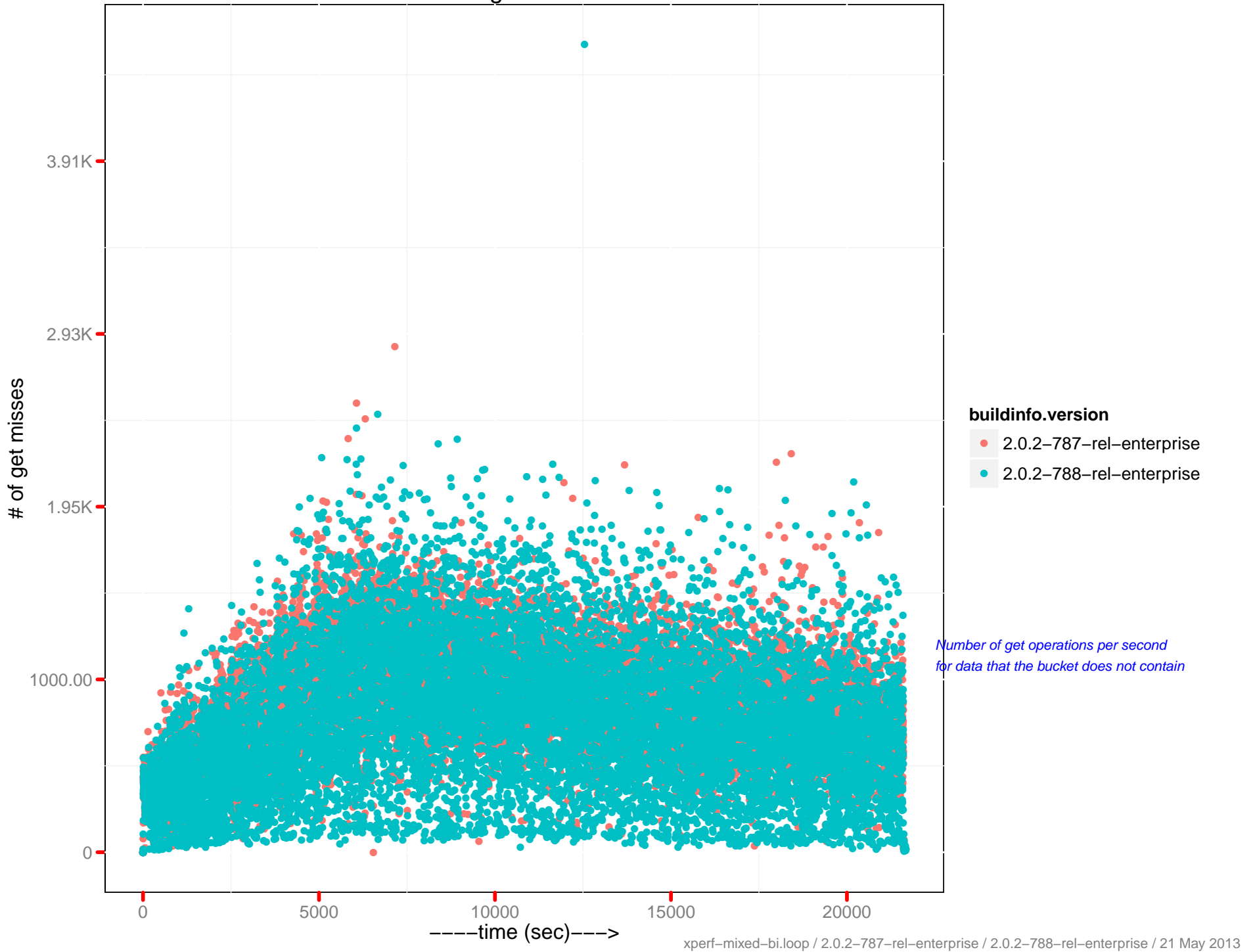


# cmd\_set ops/sec

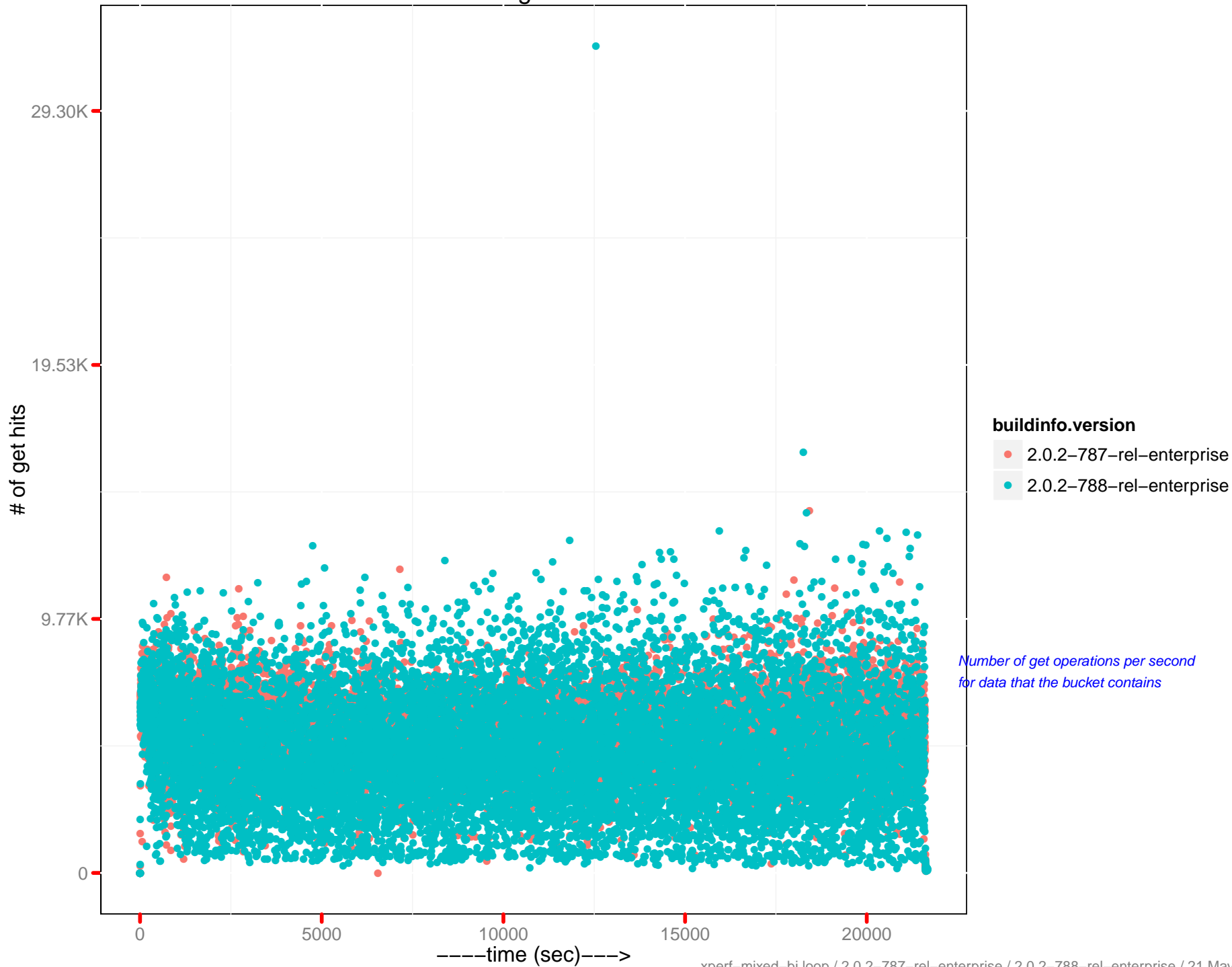




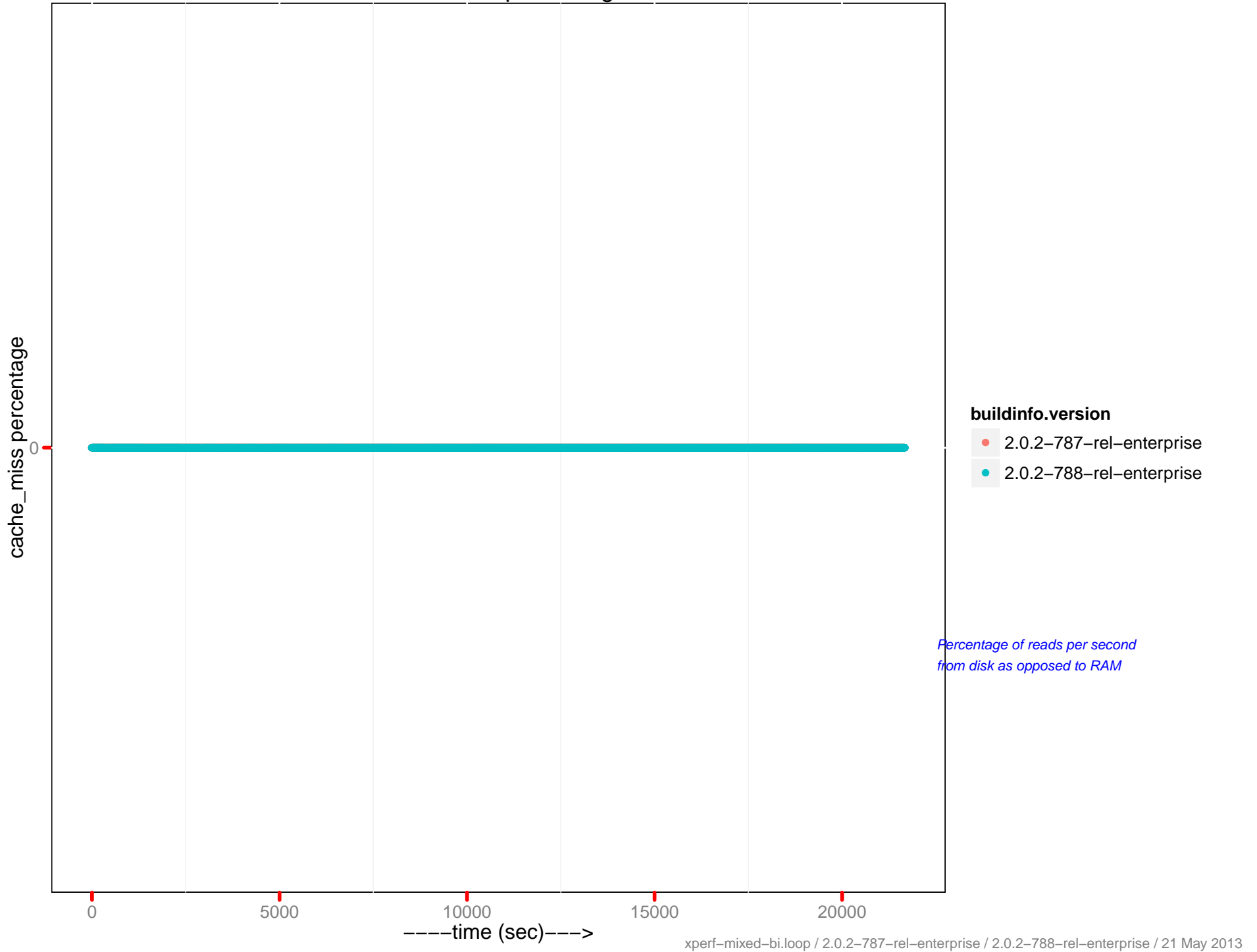
# # of get misses



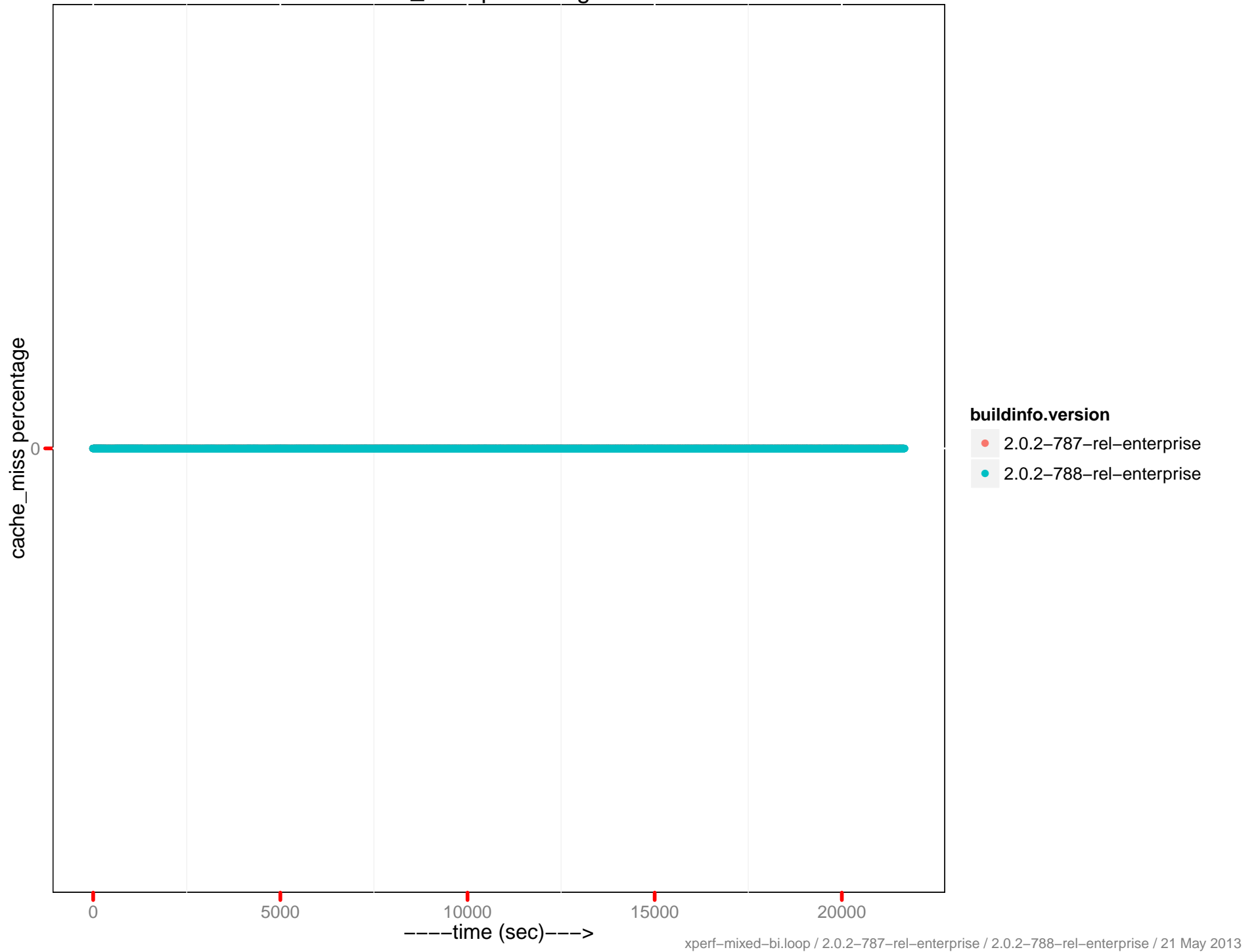
# # of get hits



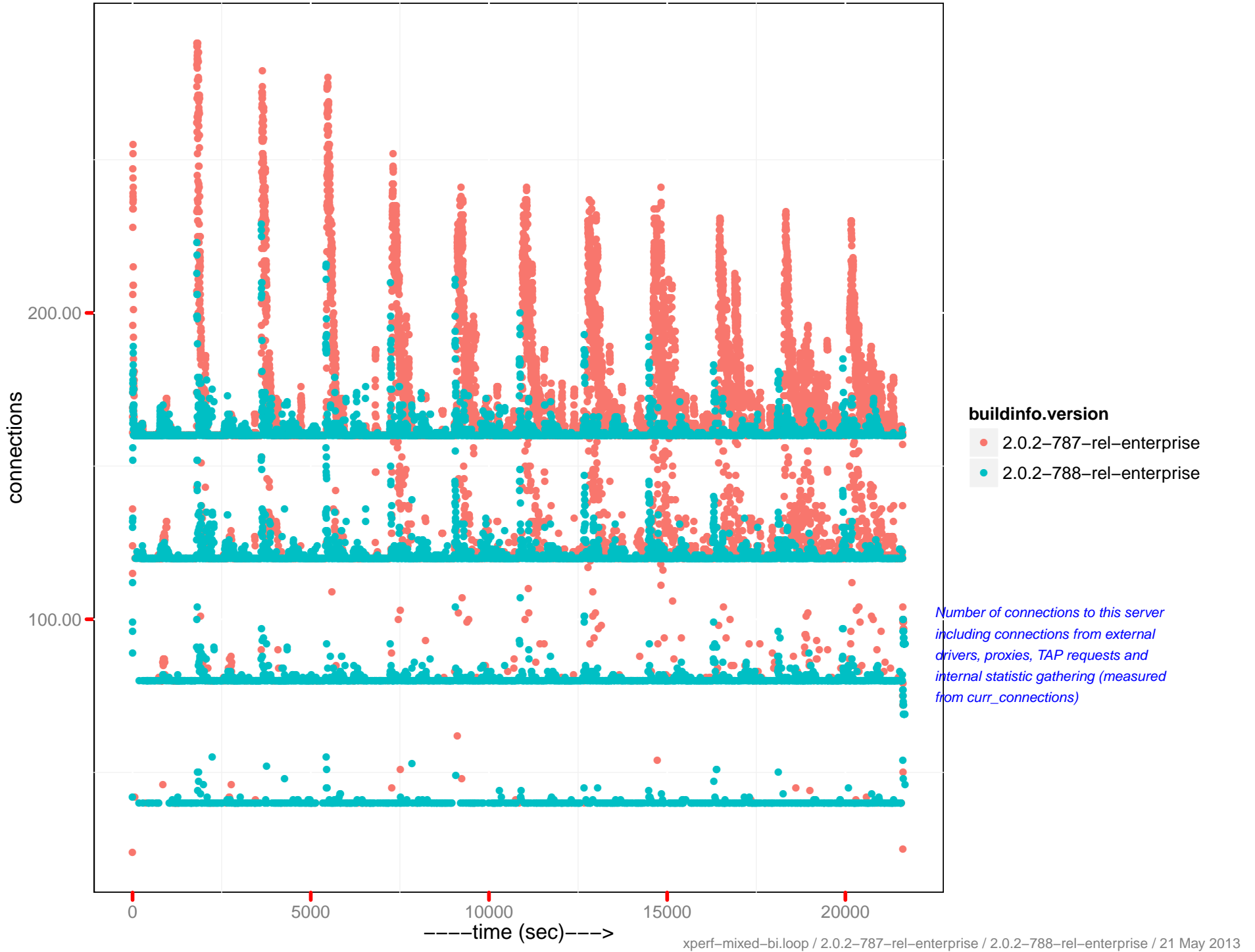
# cache\_miss percentage



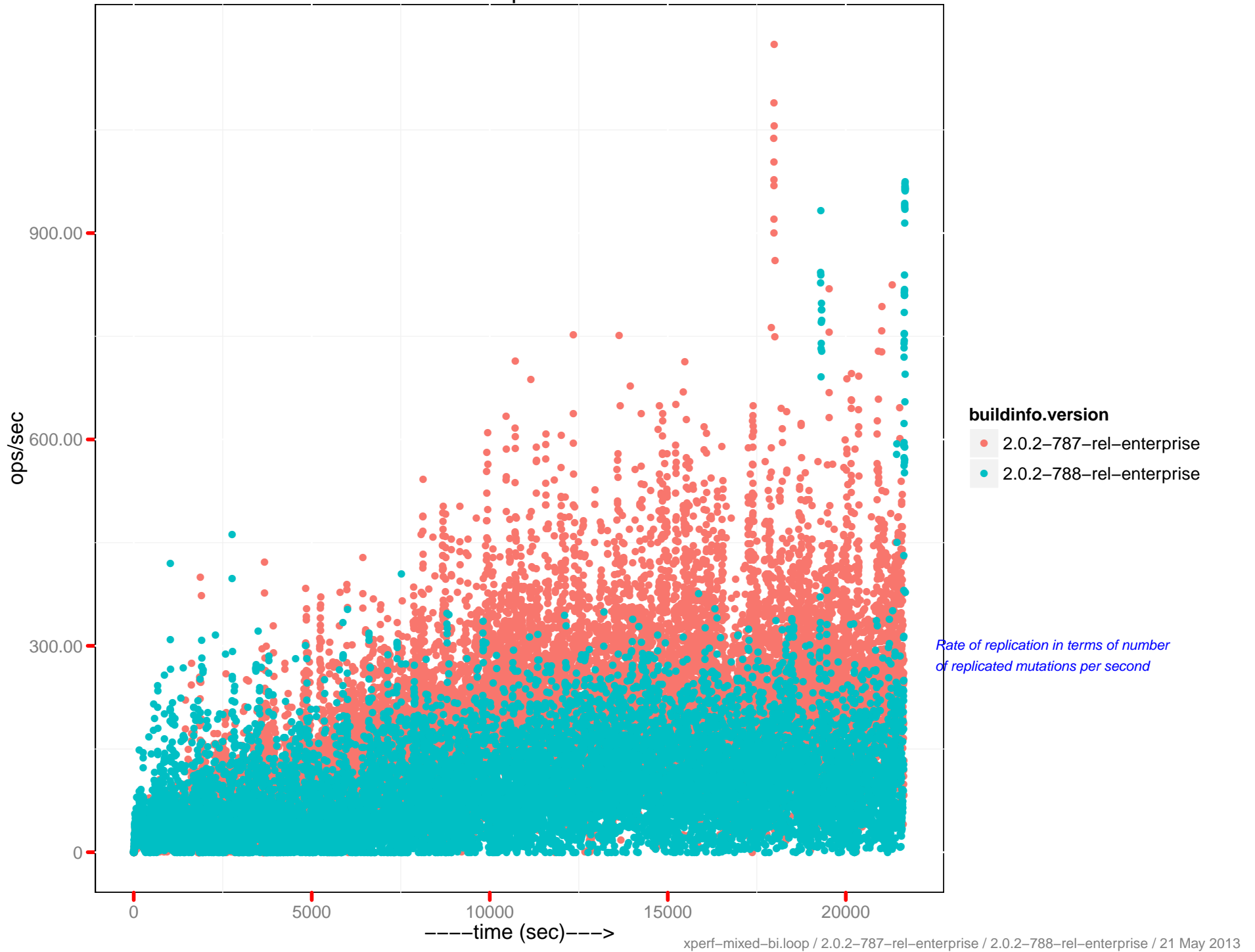
# cache\_miss percentage 0-5



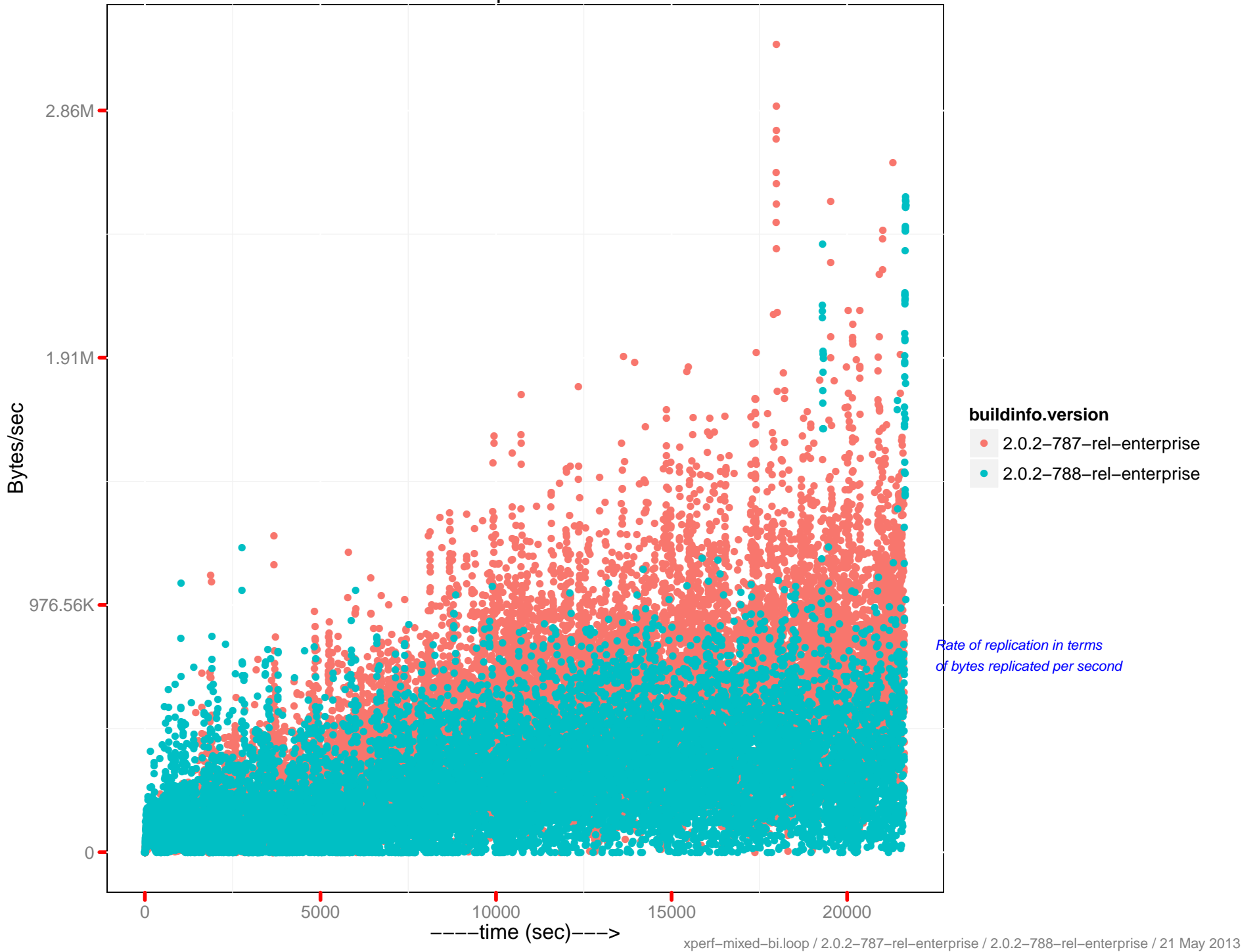
# Number of connections



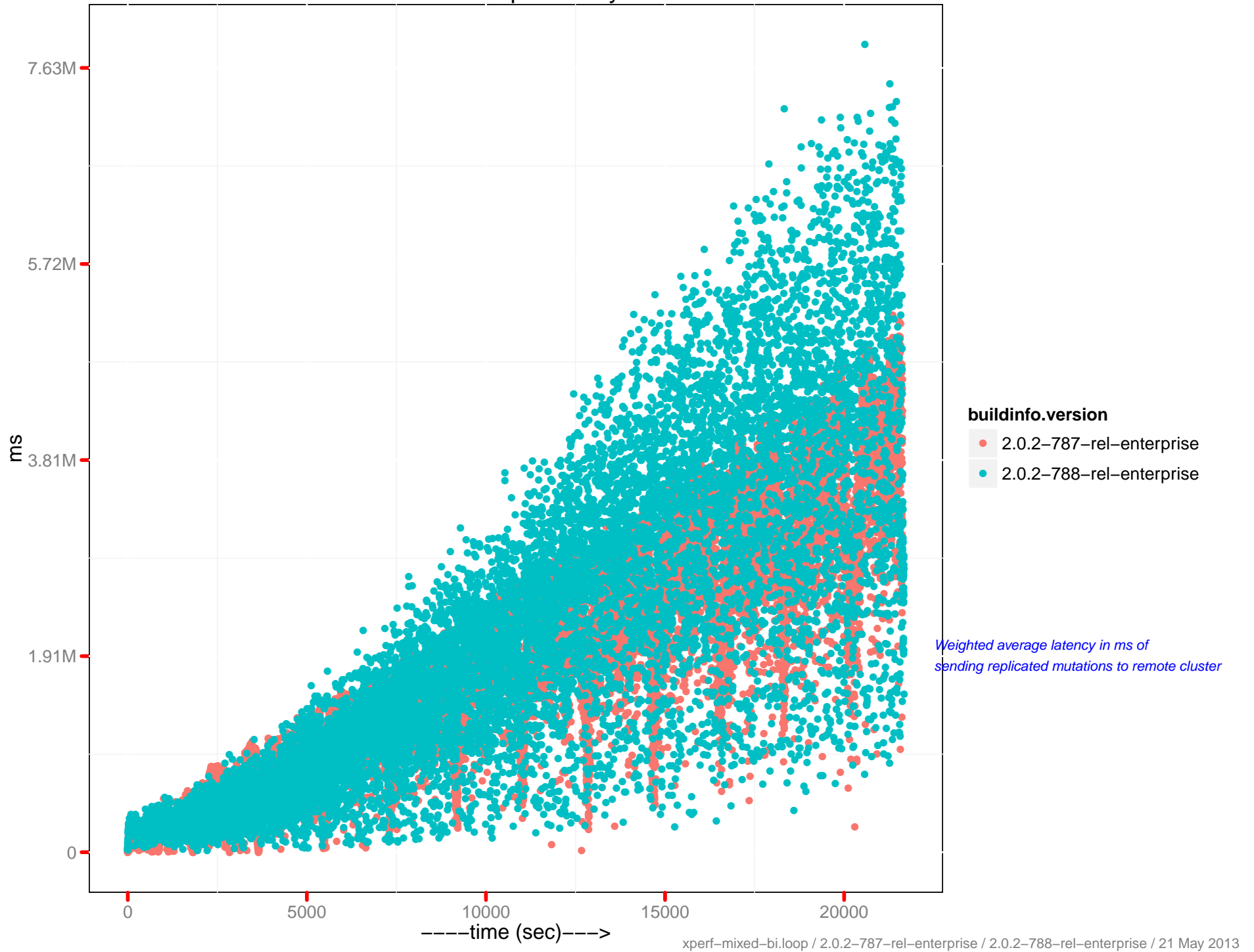
# Mutation replication rate



# Data replication rate

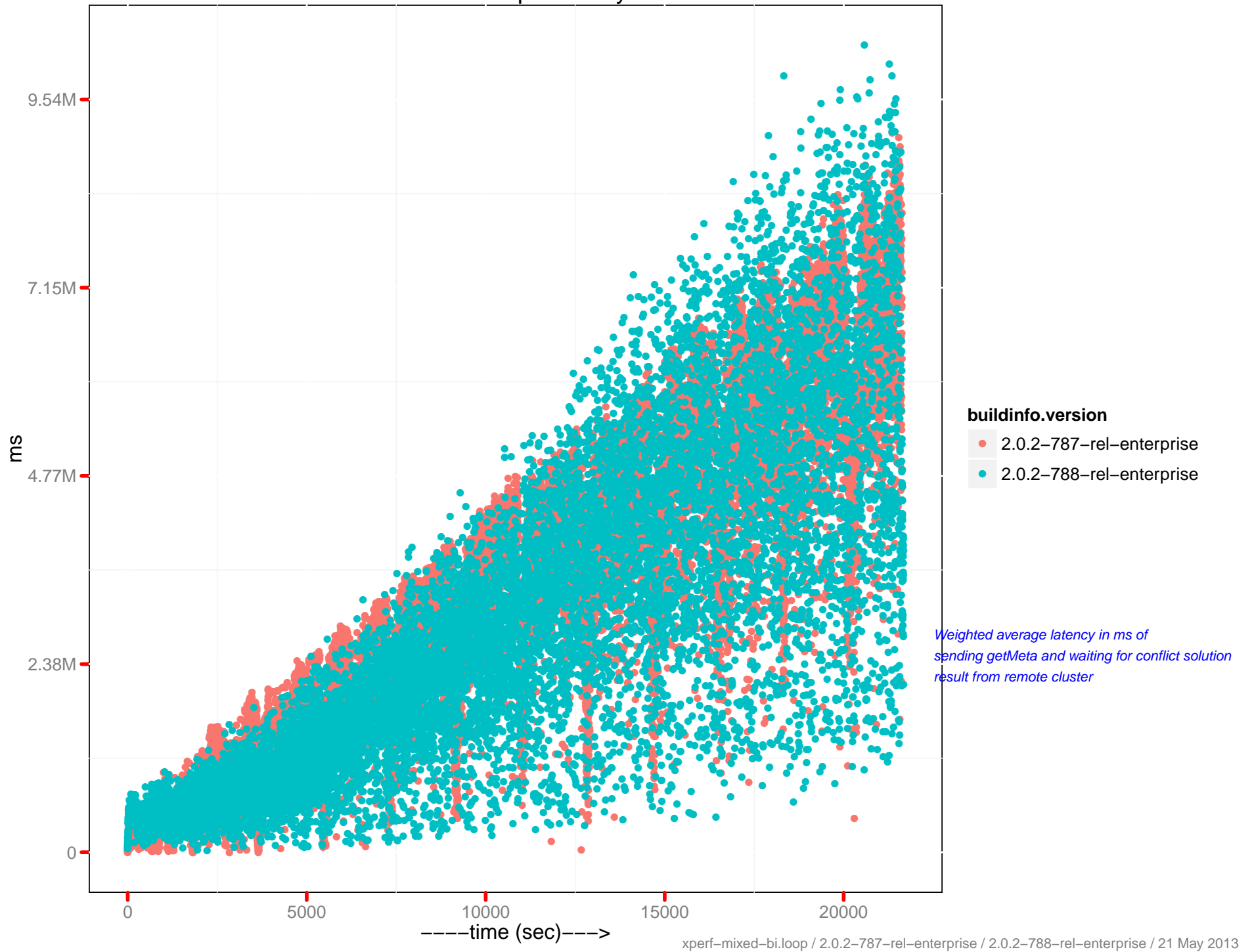


# ms doc ops latency

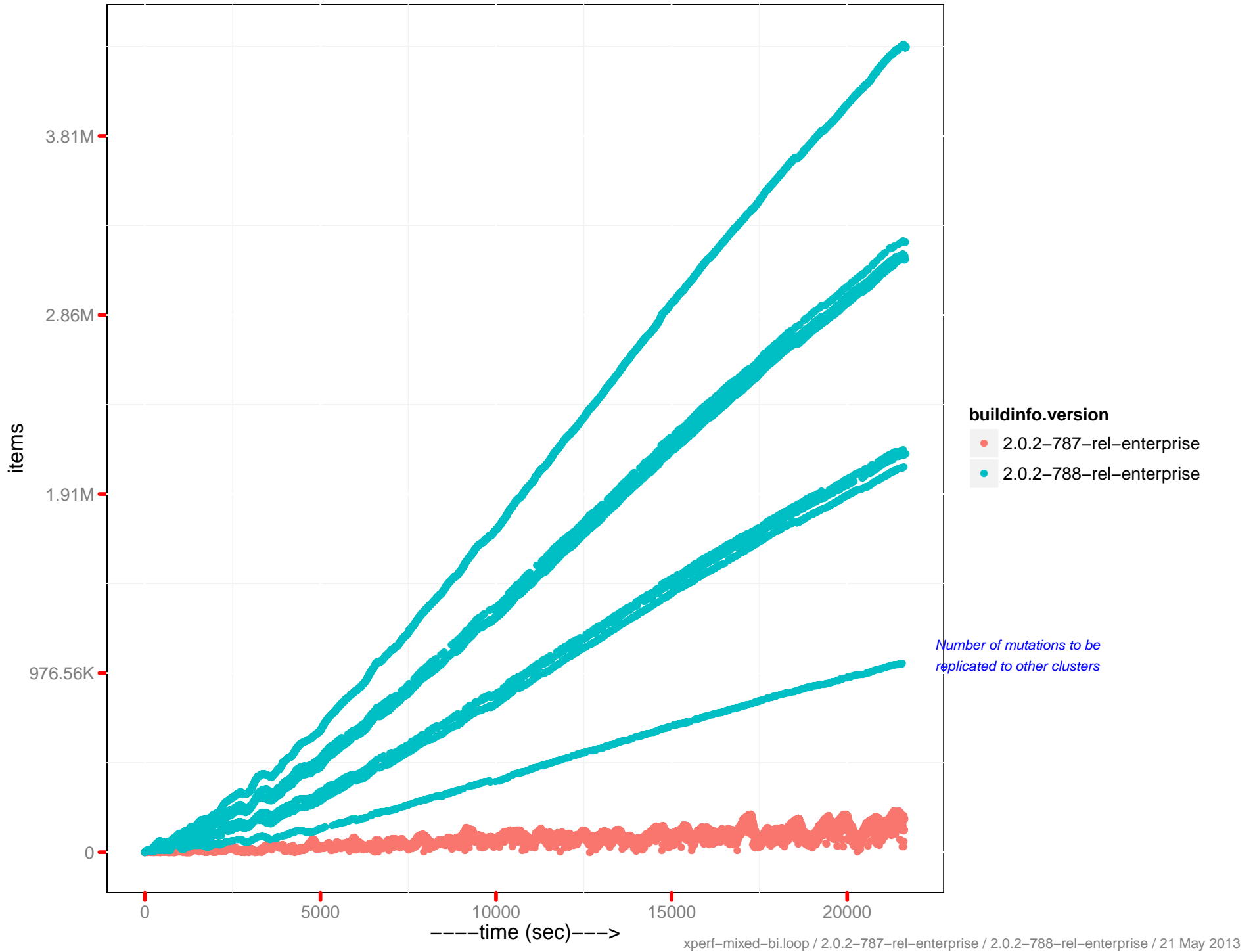




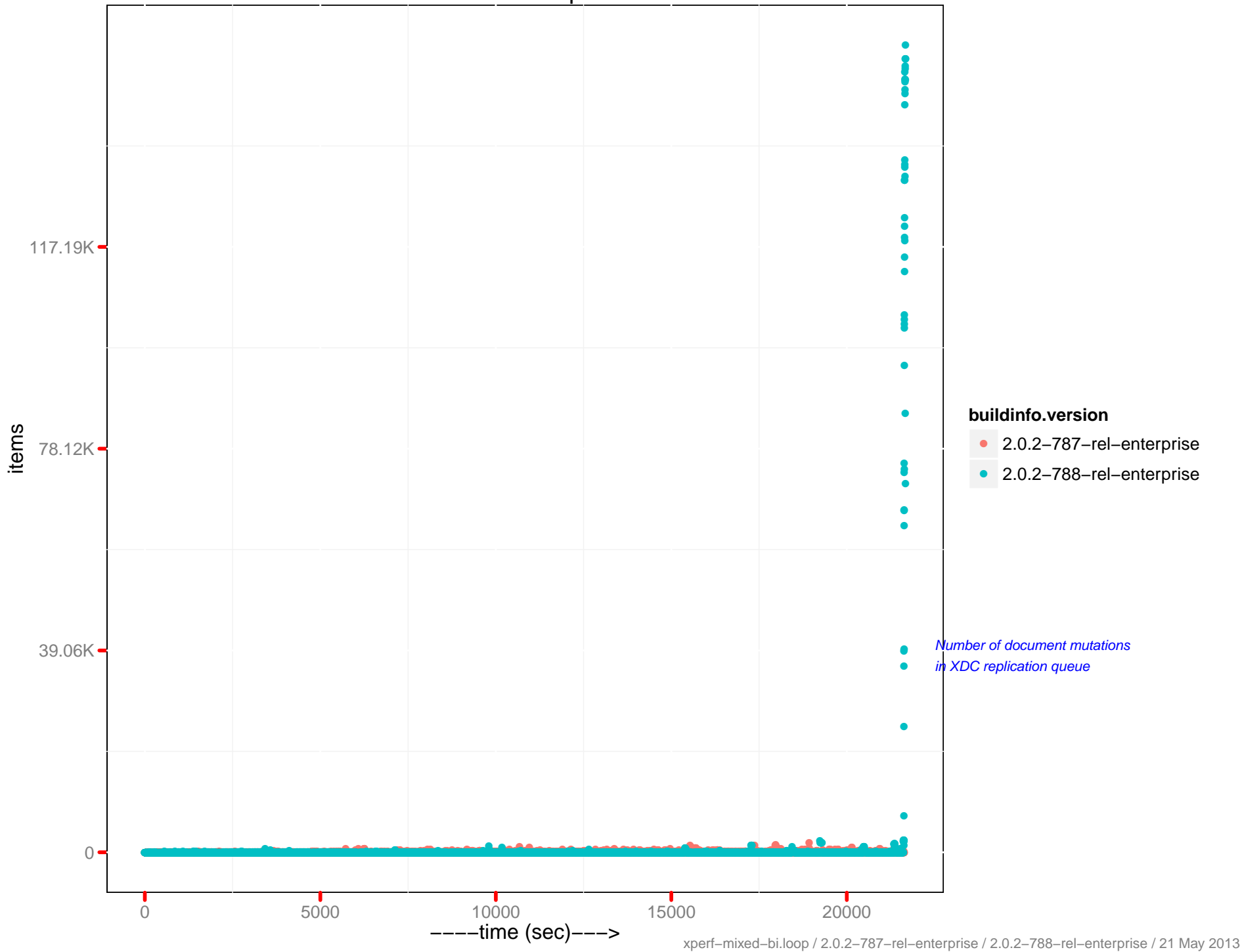
# ms meta ops latency



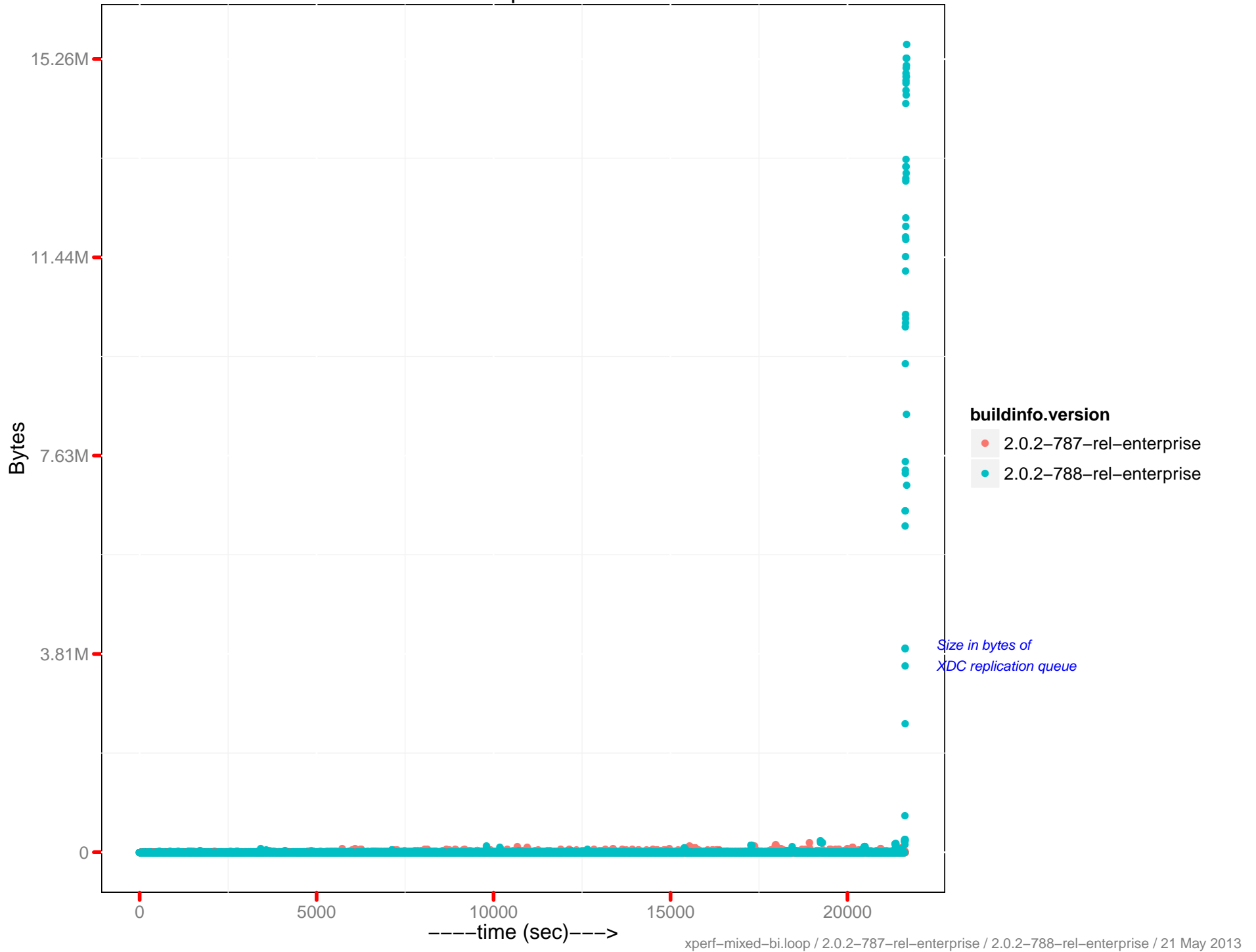
# Outbound XDCR mutations



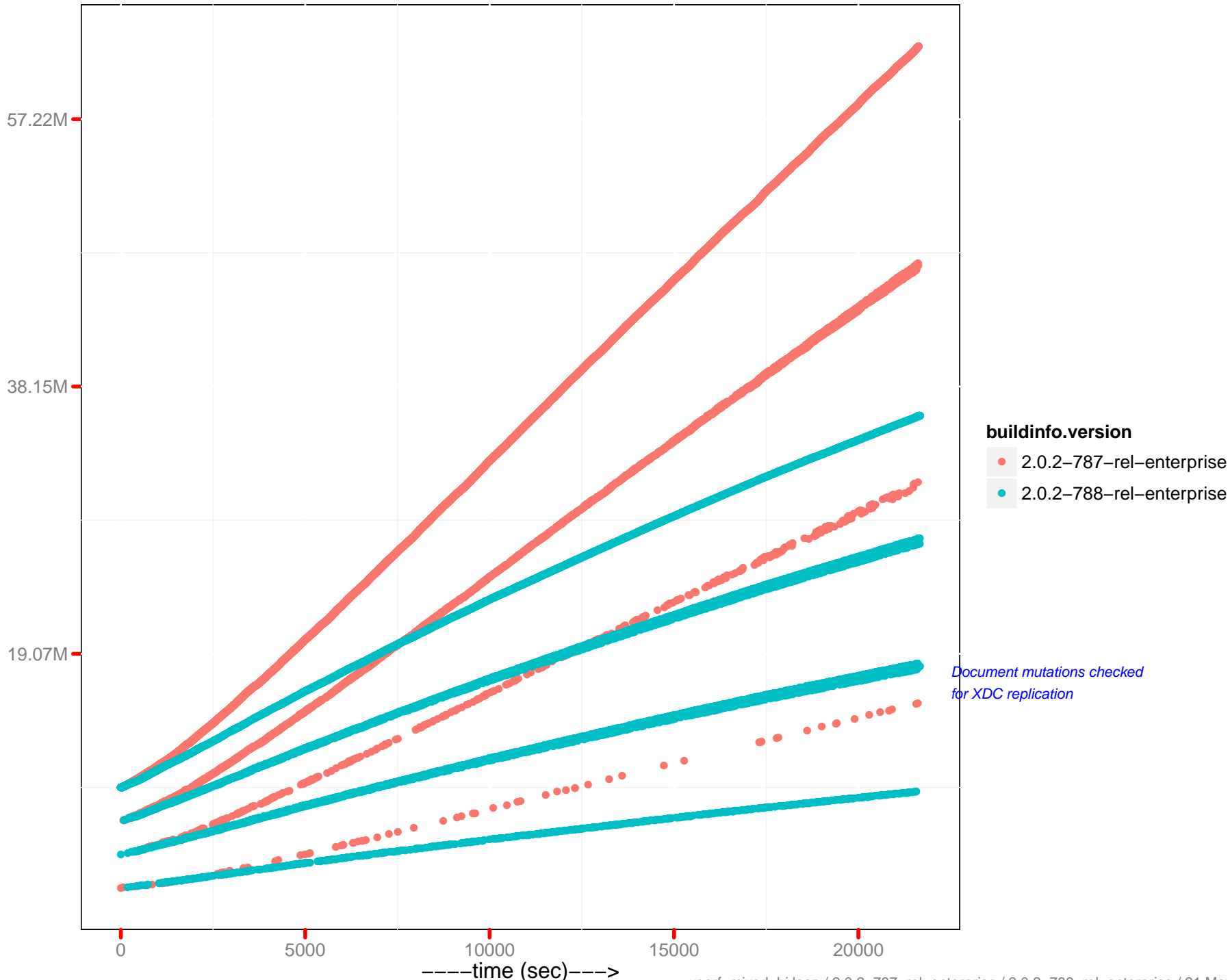
# Mutations in queue



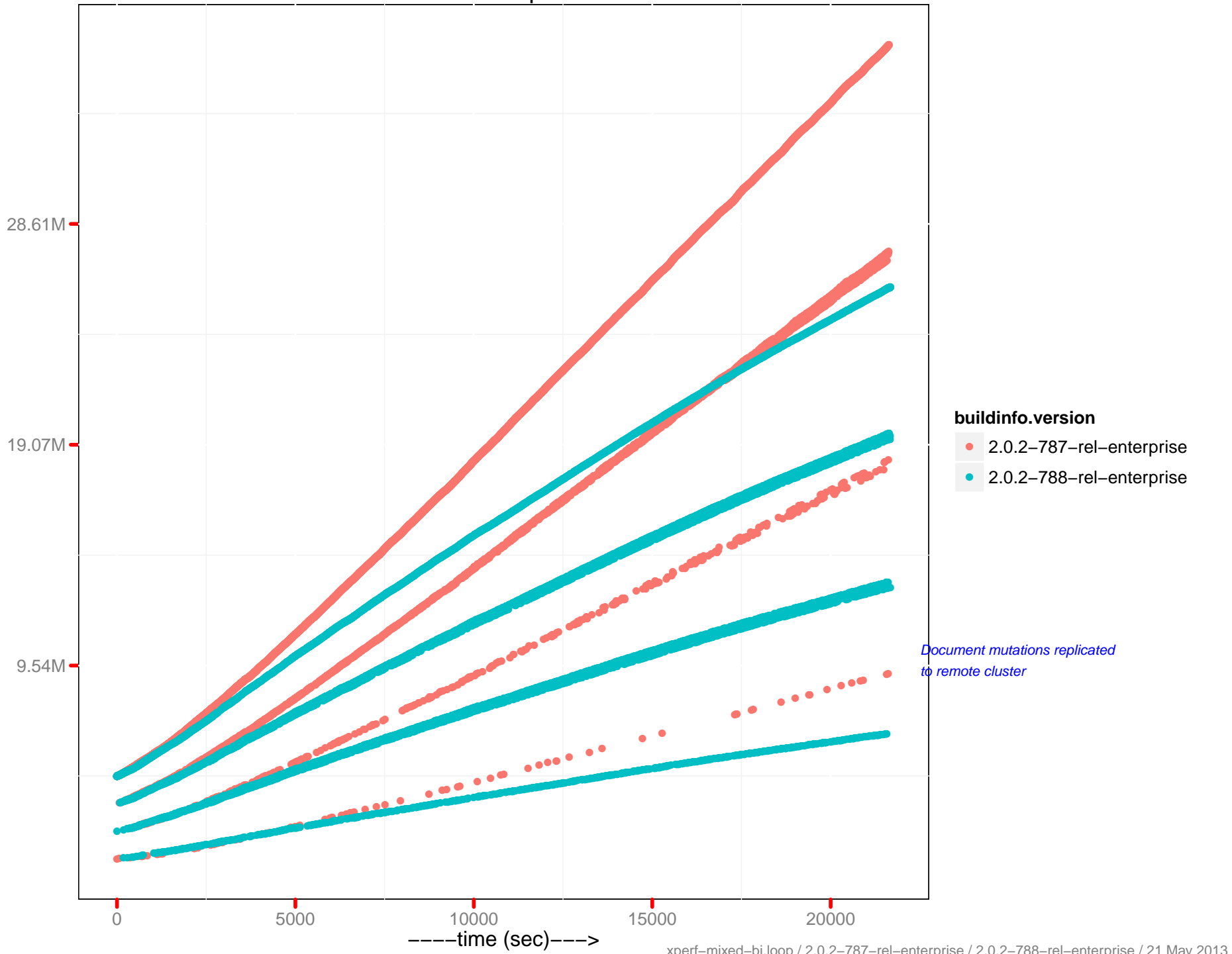
# XDCR queue size



# Mutations checked



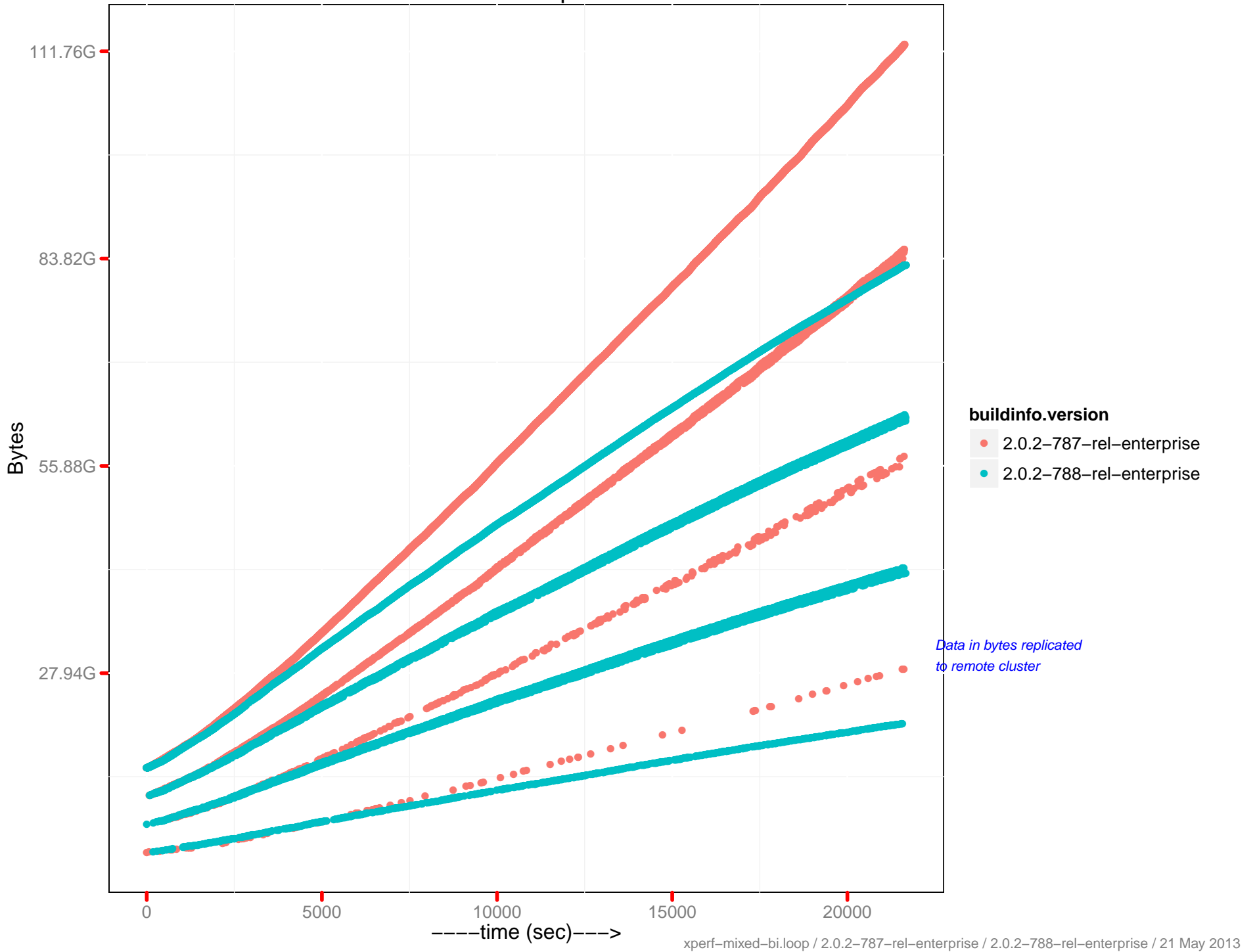
# Mutations replicated



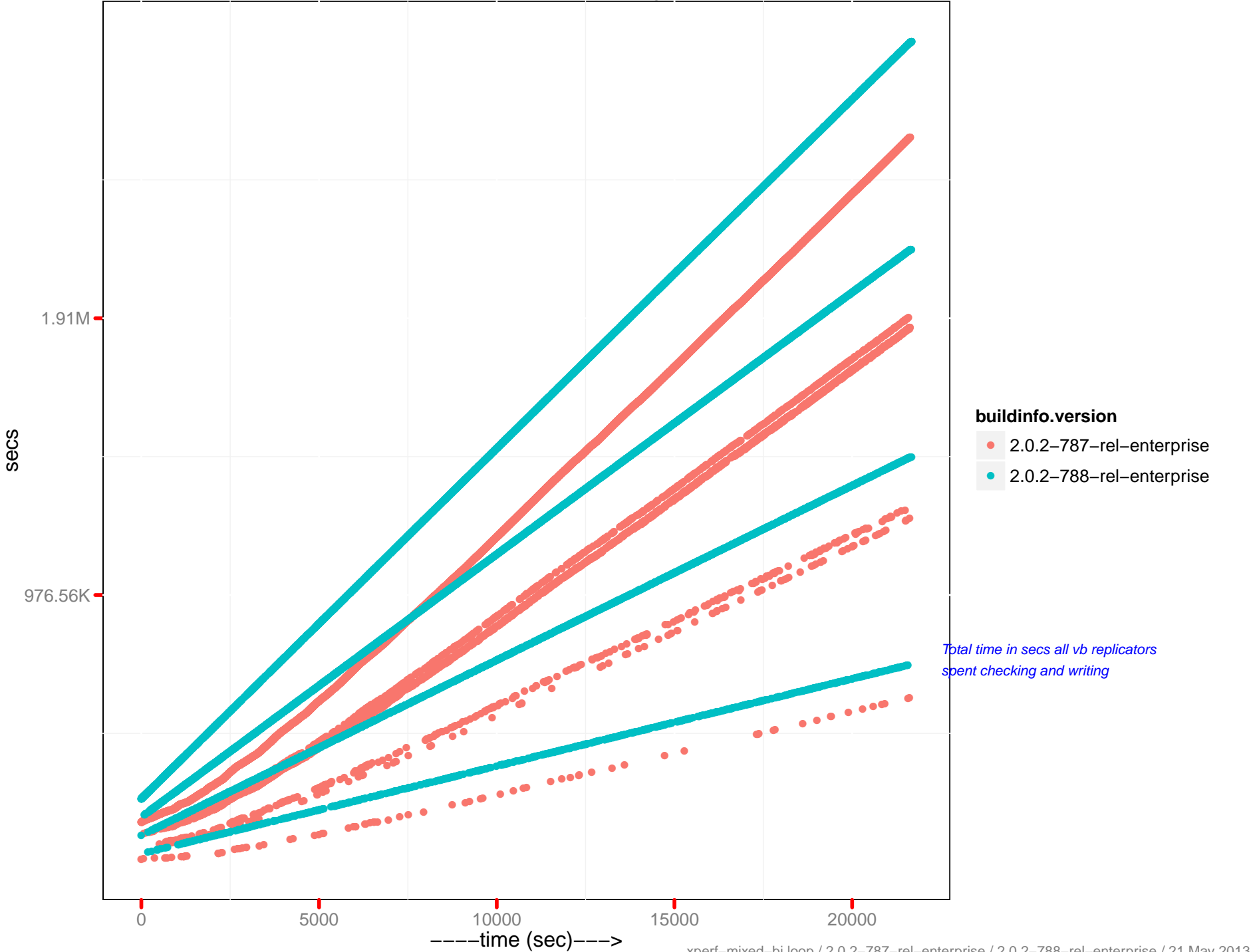
**buildinfo.version**  
● 2.0.2-787-rel-enterprise  
● 2.0.2-788-rel-enterprise

*Document mutations replicated to remote cluster*

# XDCR data replicated

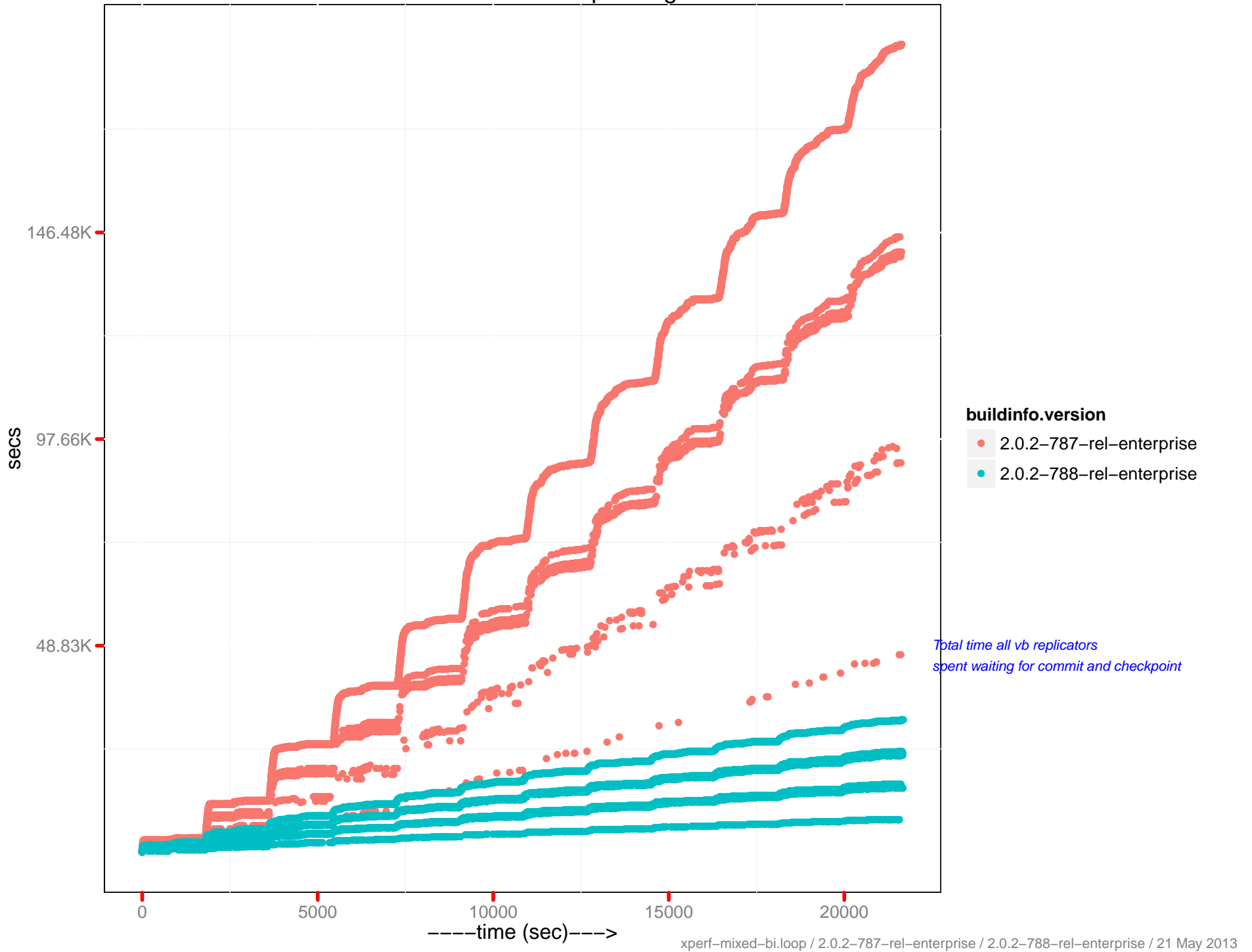


# XDCR secs in replicating

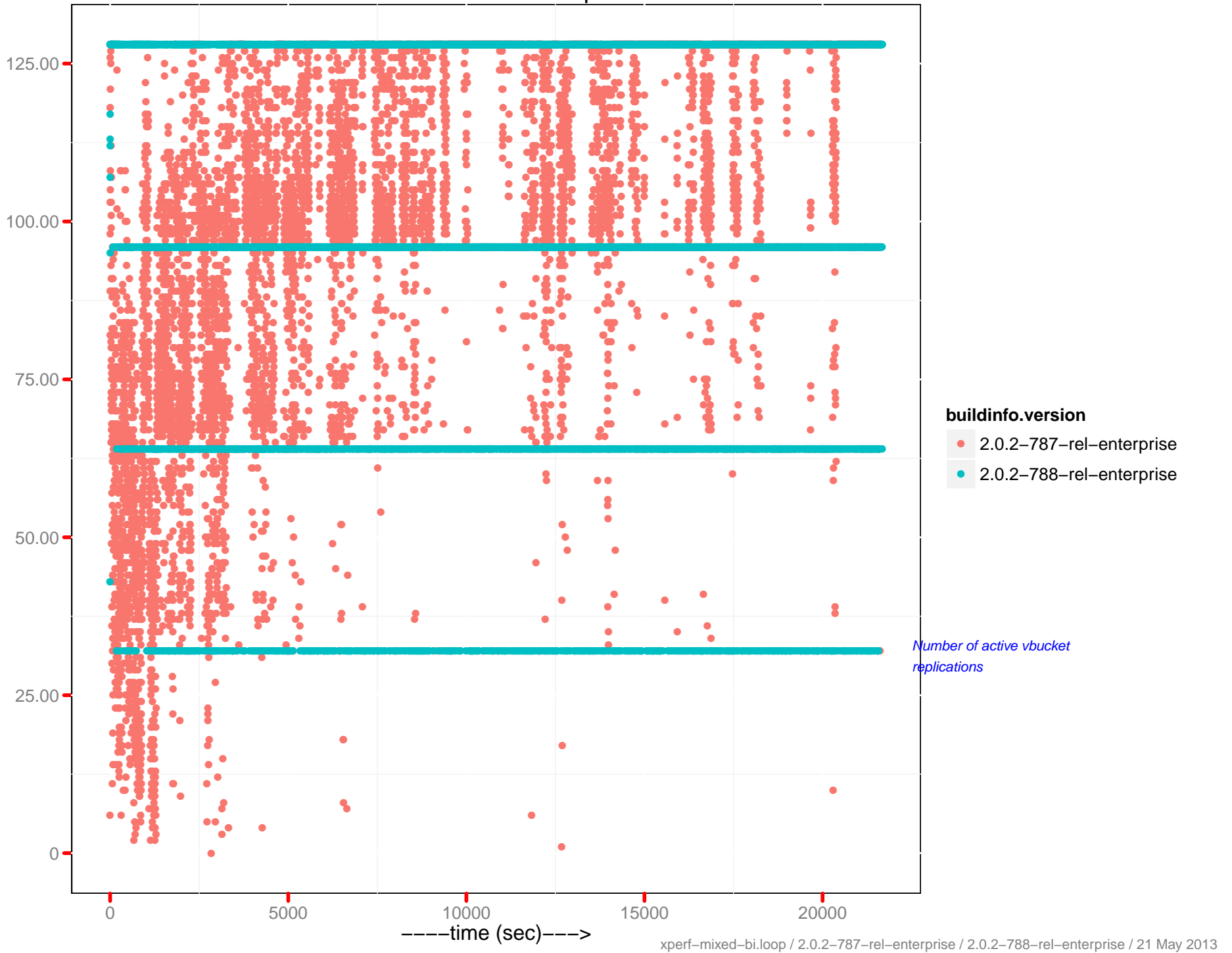




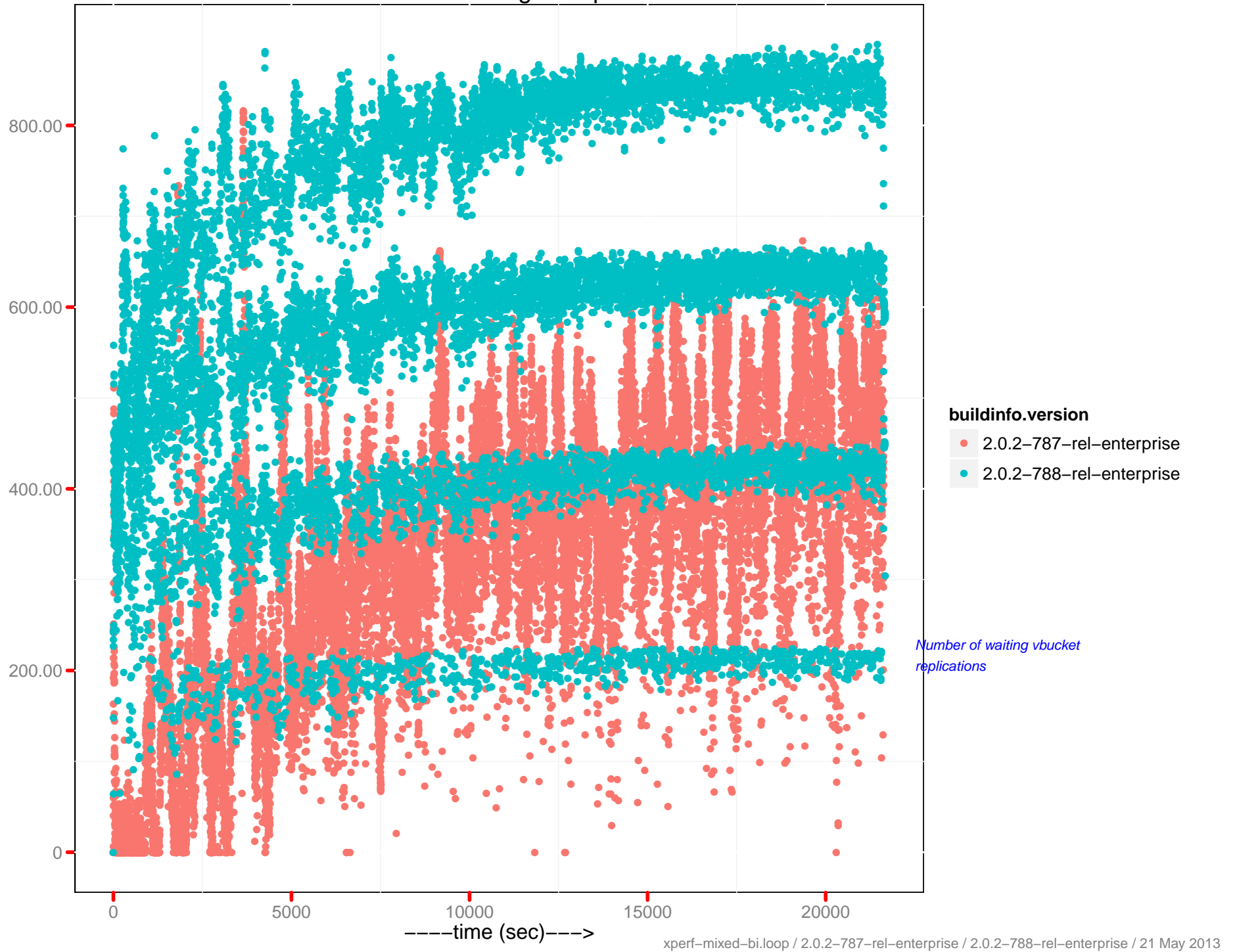
# XDCR secs in checkpointing



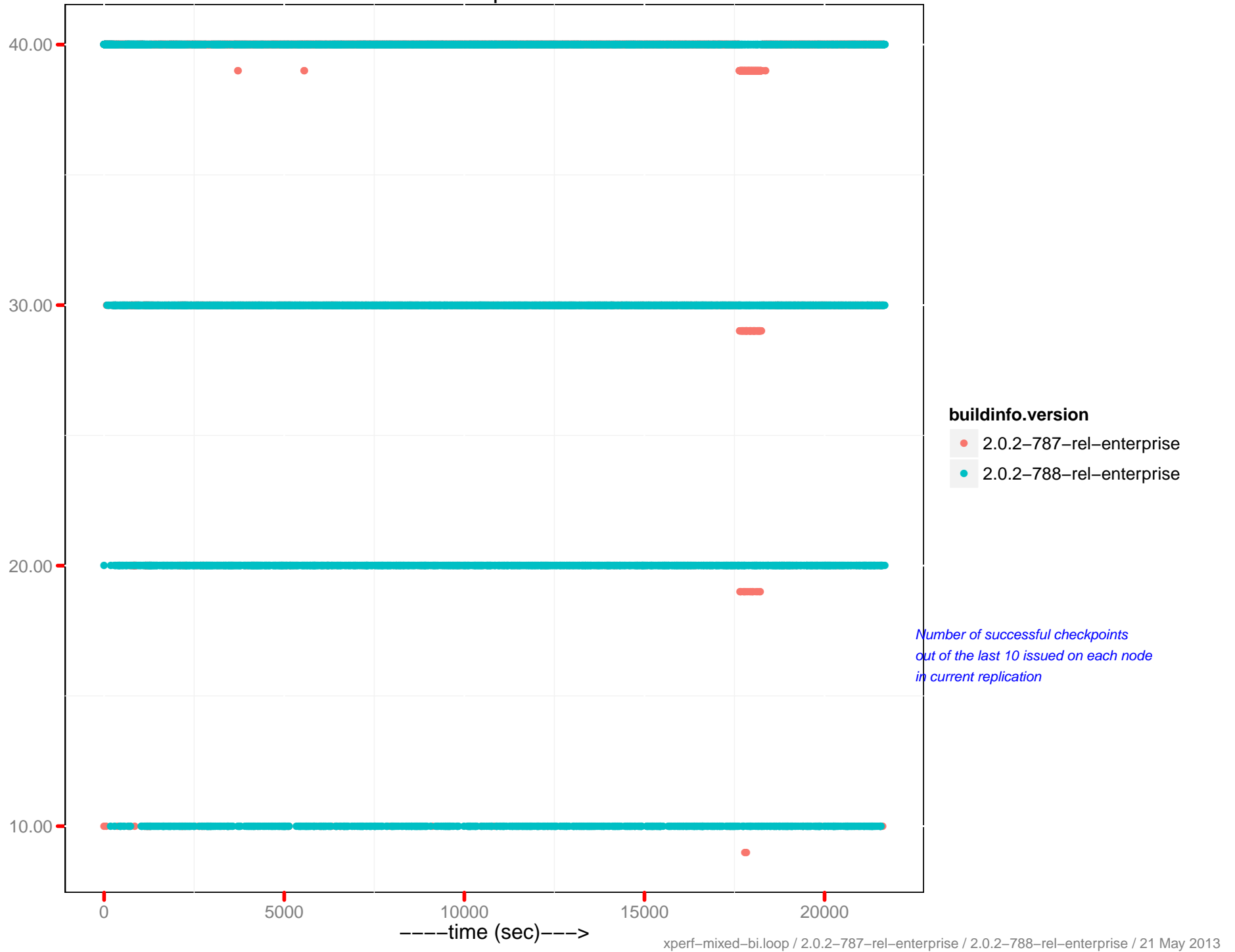
# XDCR active vb reps



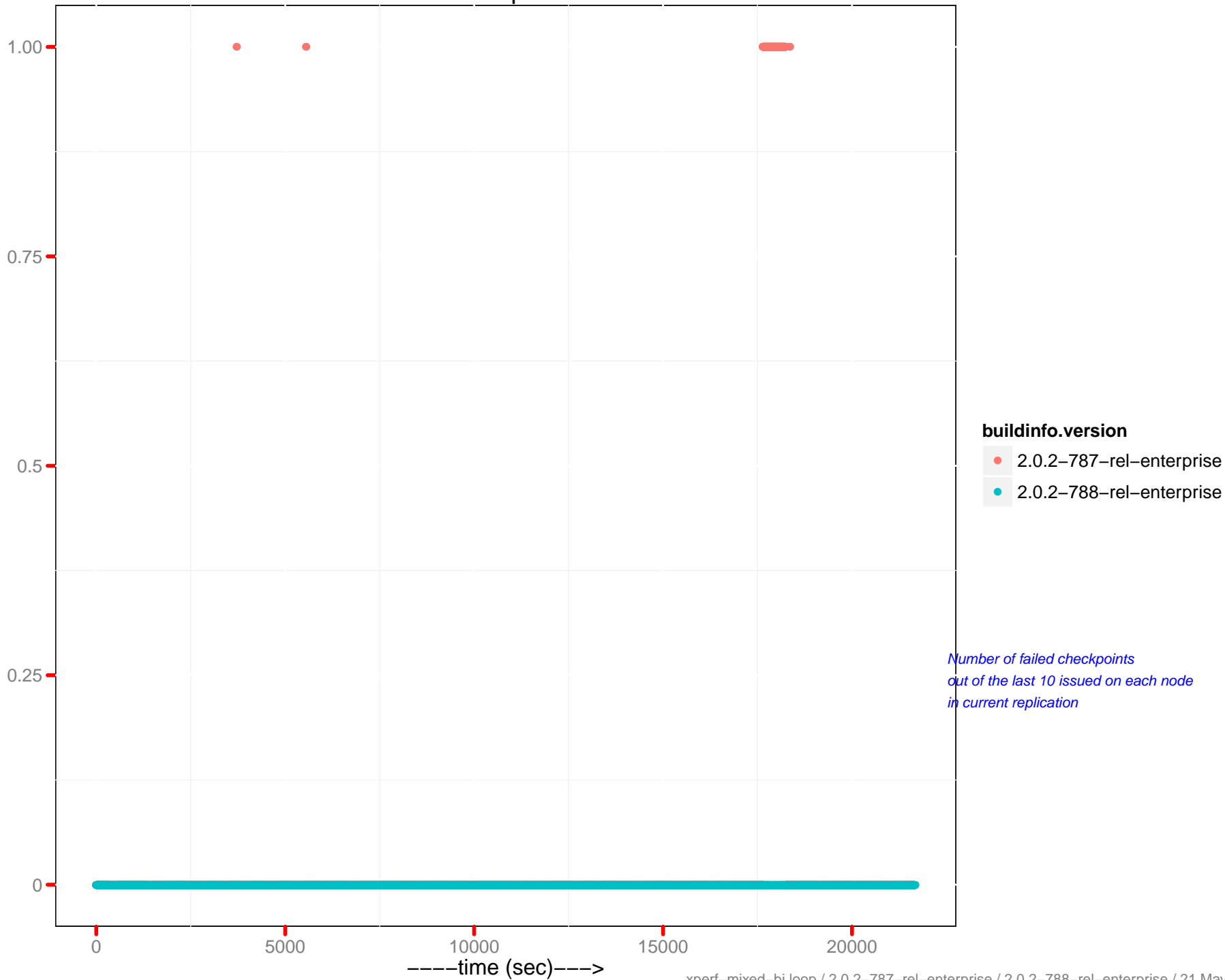
# XDCR waiting vb reps



# XDCR checkpoints issued

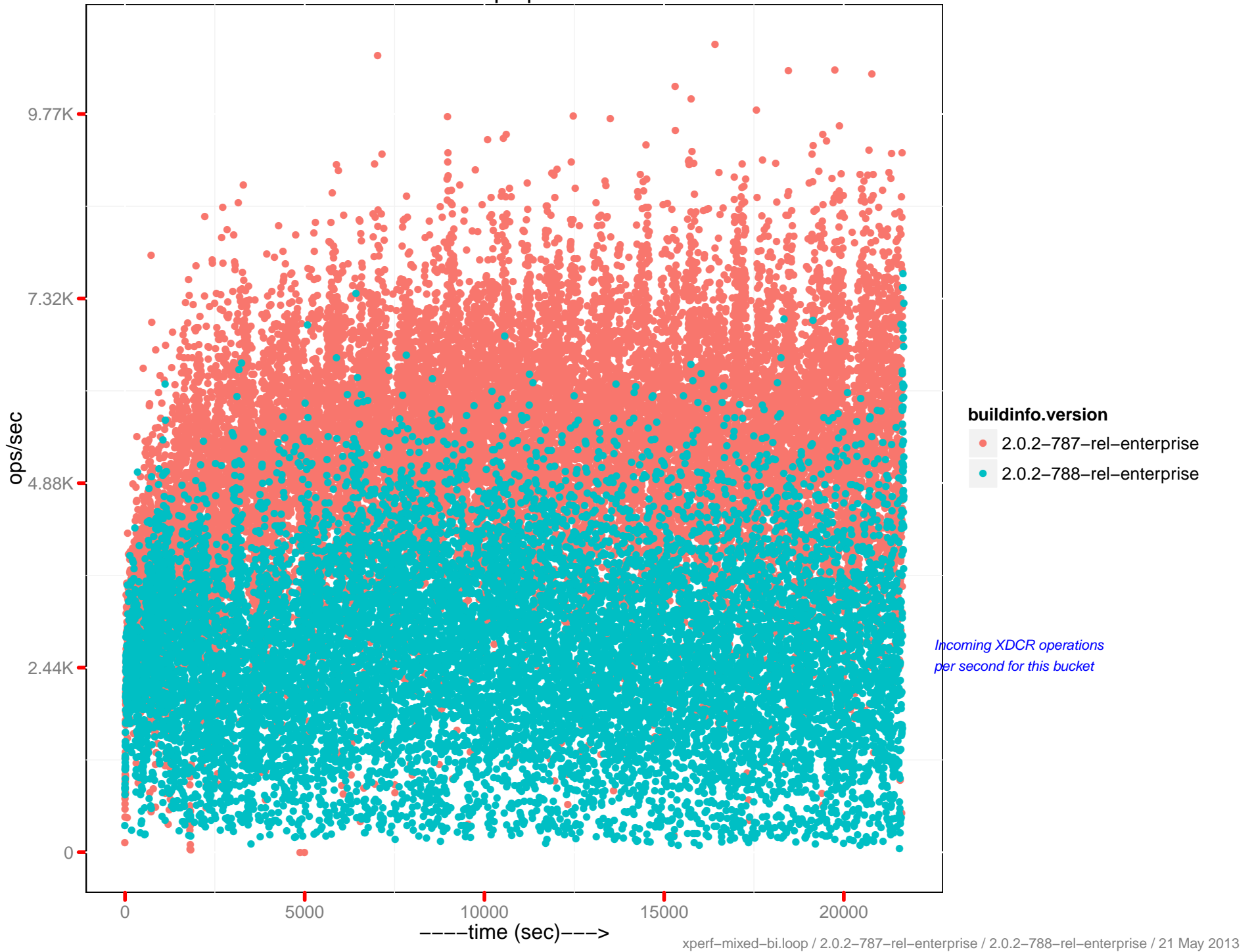


# XDCR checkpoints failed



*Number of failed checkpoints  
out of the last 10 issued on each node  
in current replication*

# XDC ops per sec



# Metadata gets per sec

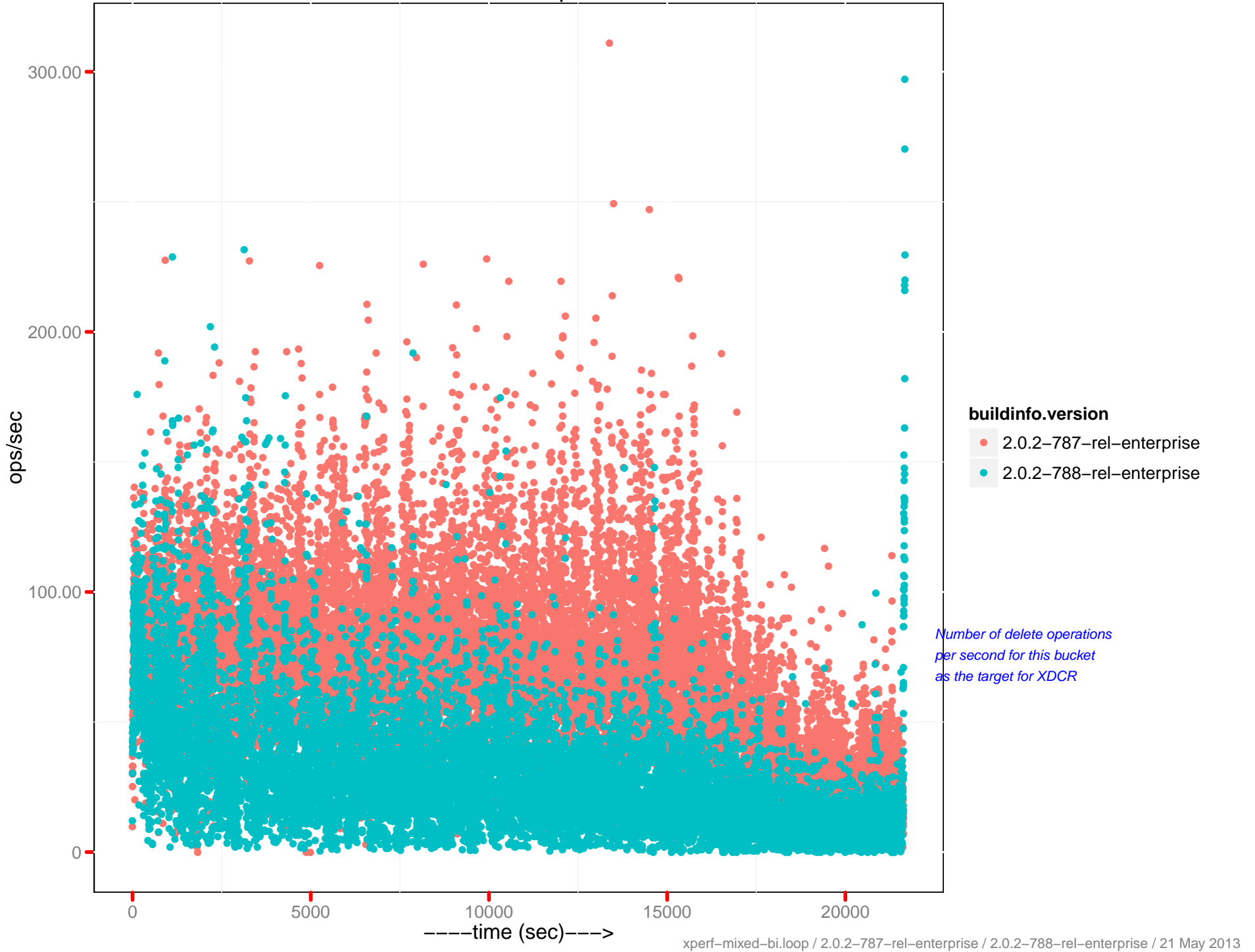


# Metadata sets per sec

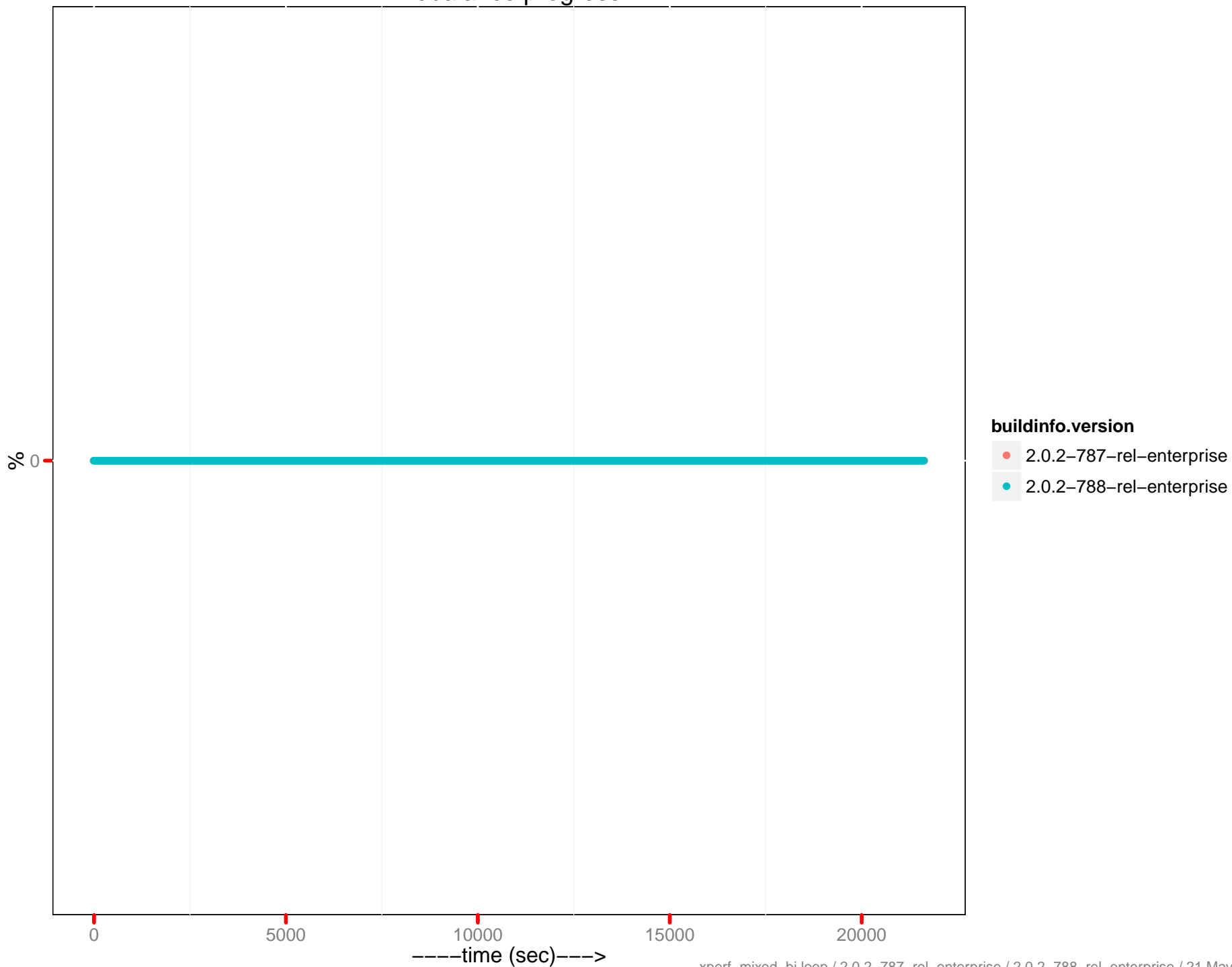




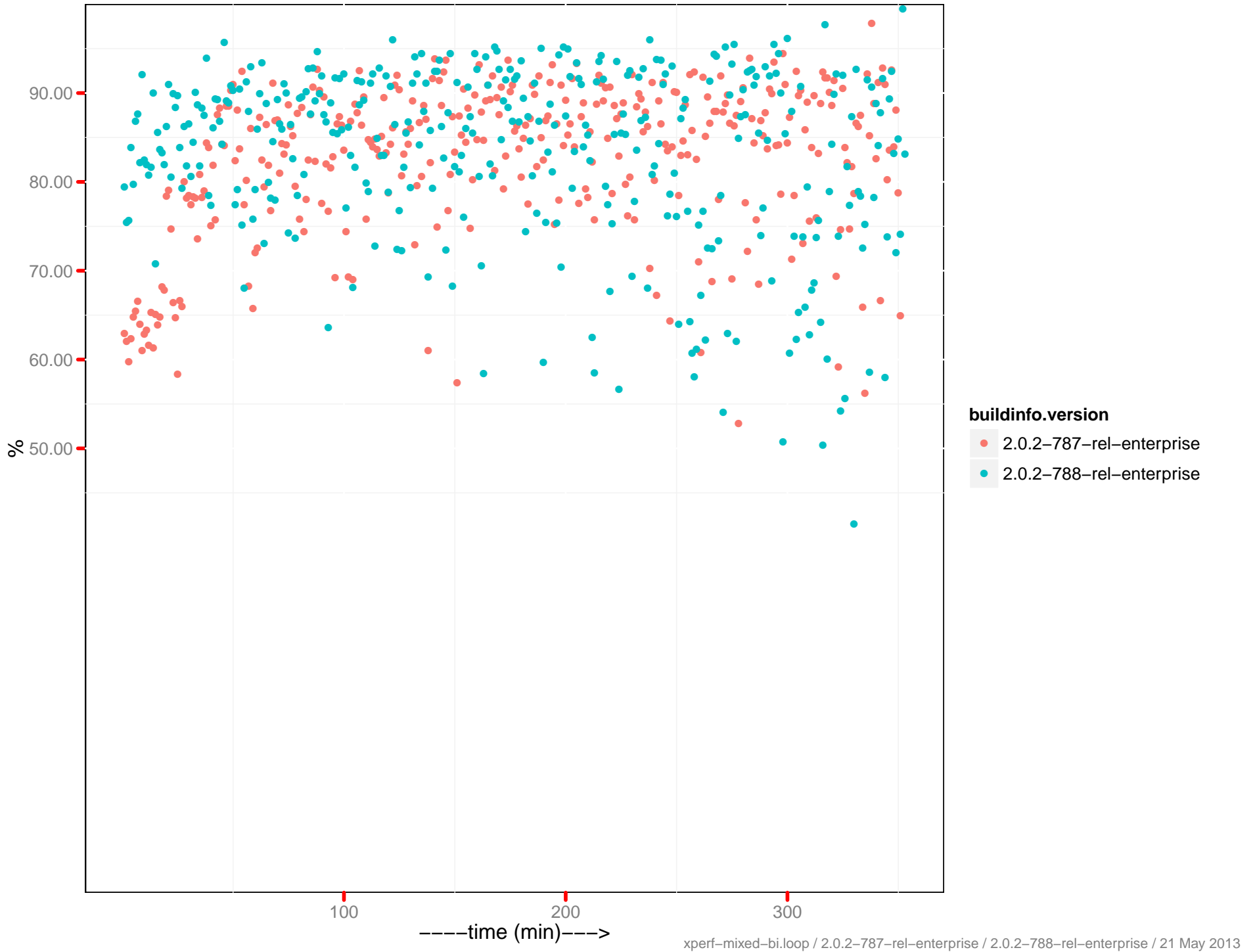
# Metadata dels per sec



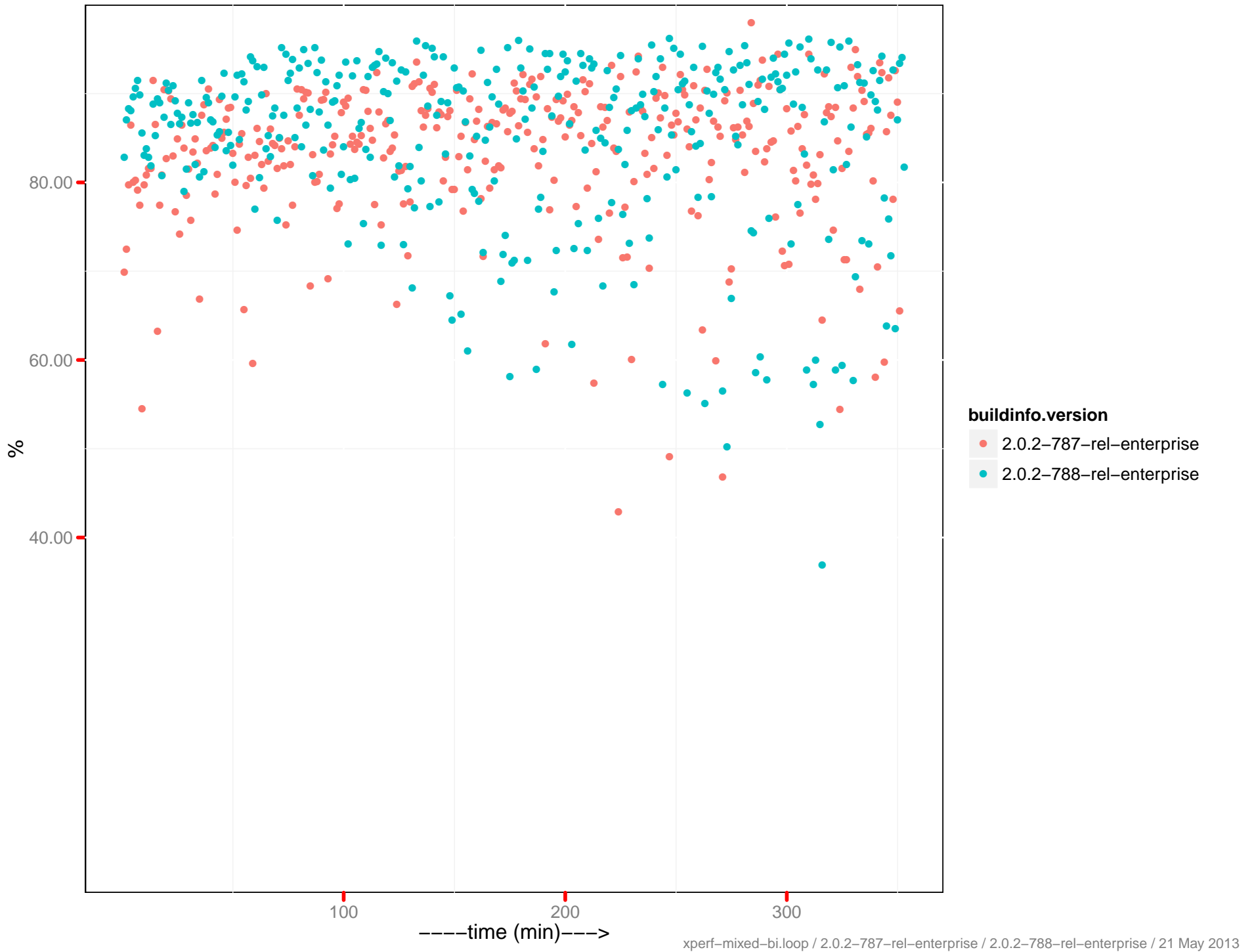
# Rebalance progress



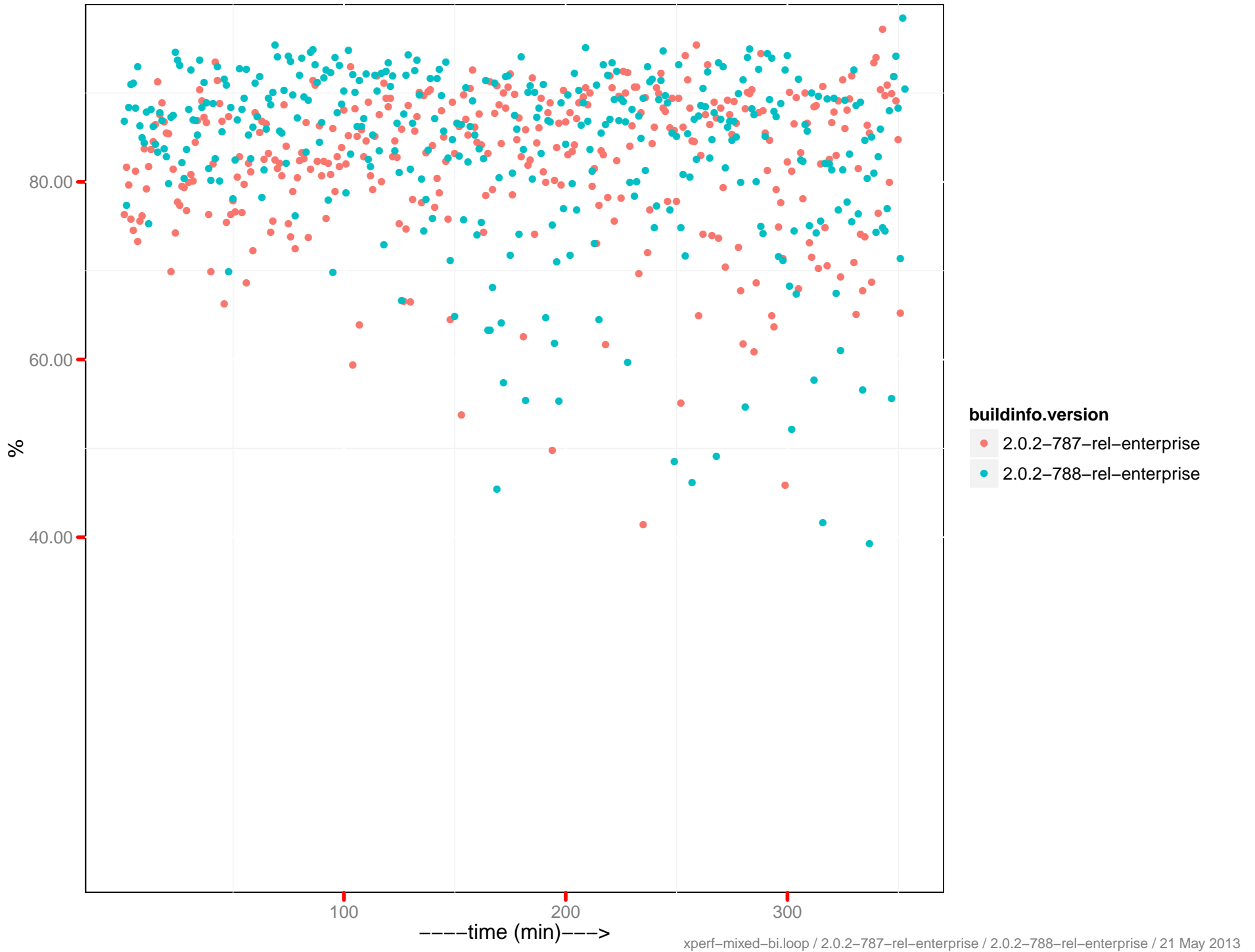
# CPU utilization – 172.23.97.53:8091



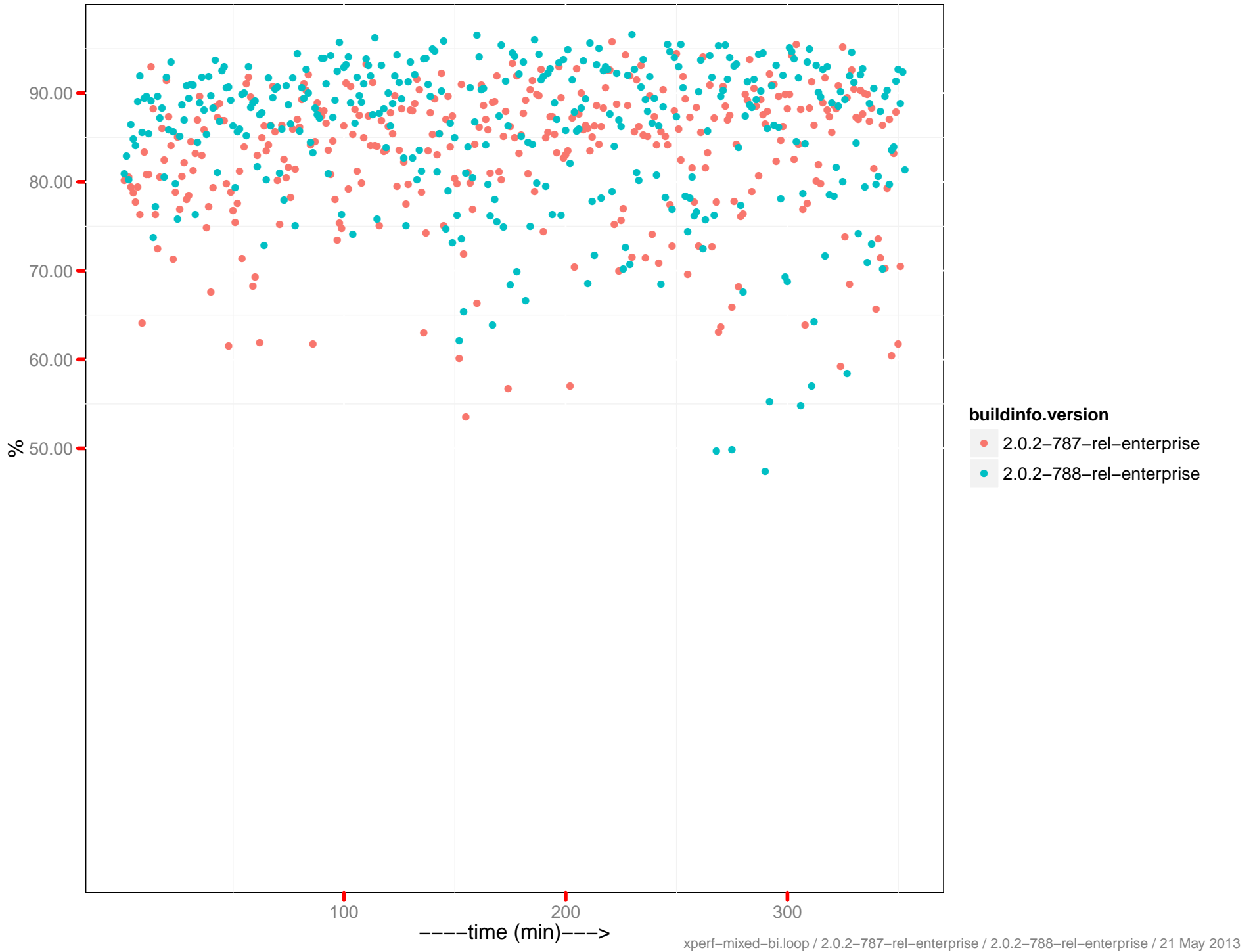
# CPU utilization – 172.23.97.54:8091



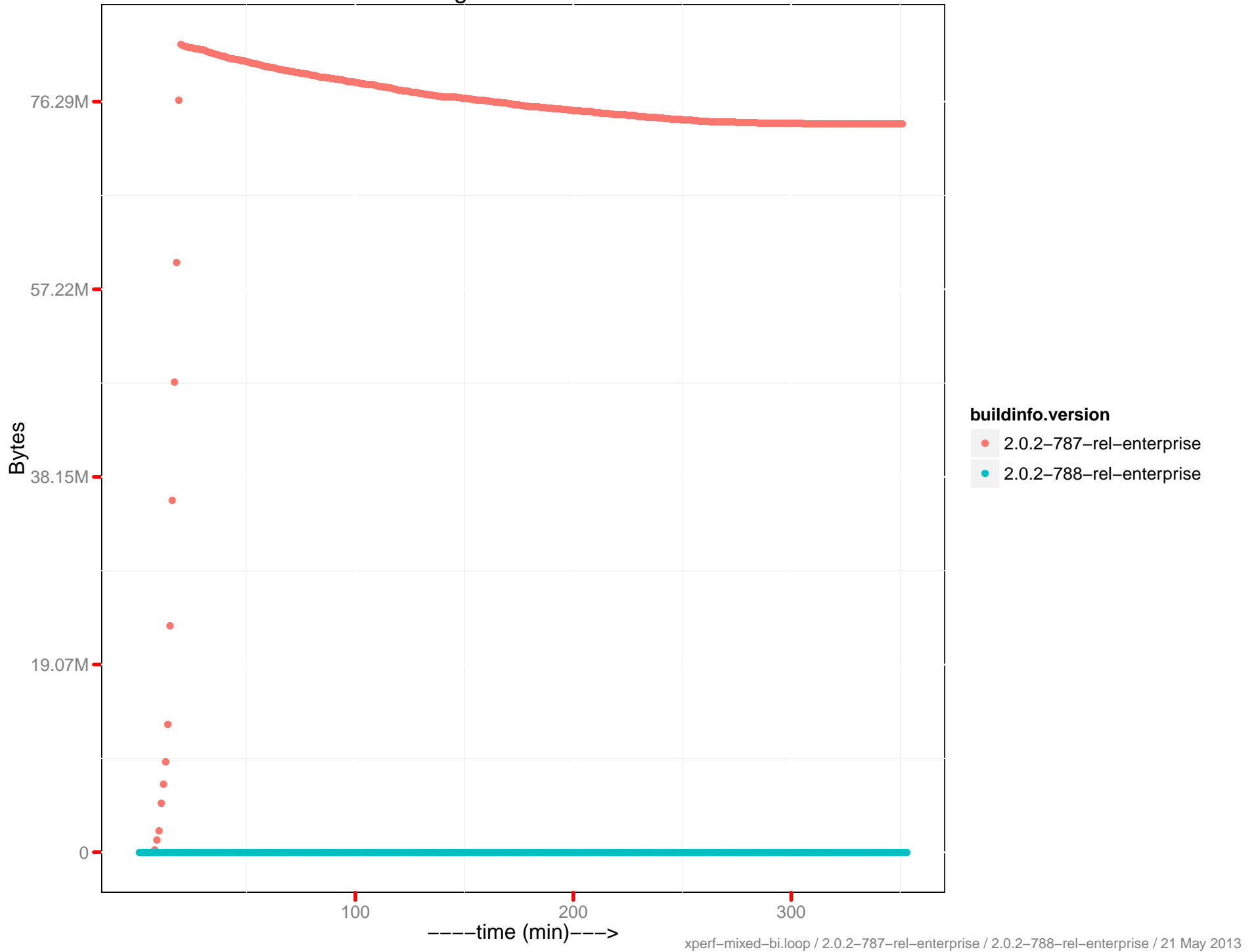
# CPU utilization – 172.23.97.55:8091



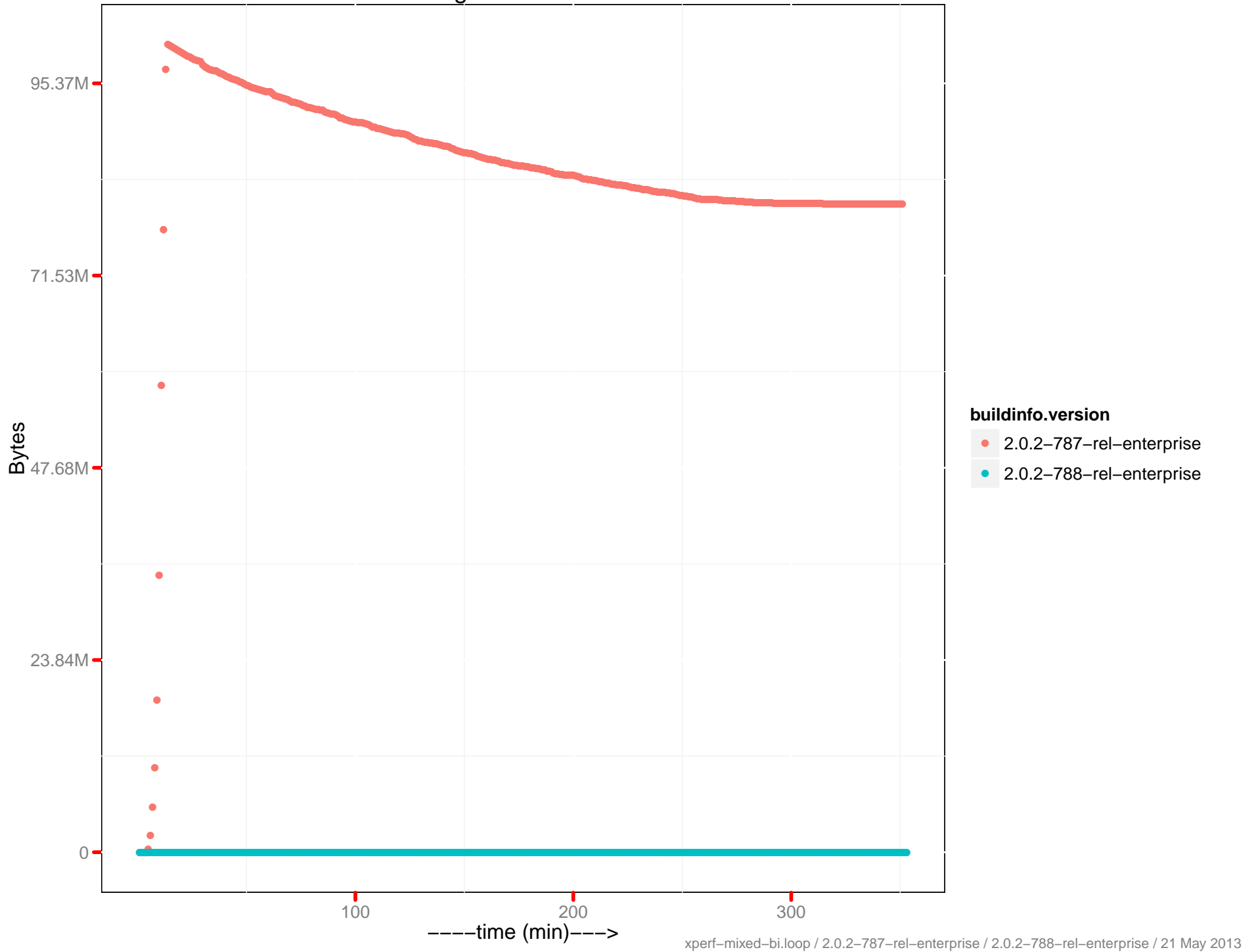
# CPU utilization - 172.23.97.56:8091



# SWAP Usage – 172.23.97.53:8091

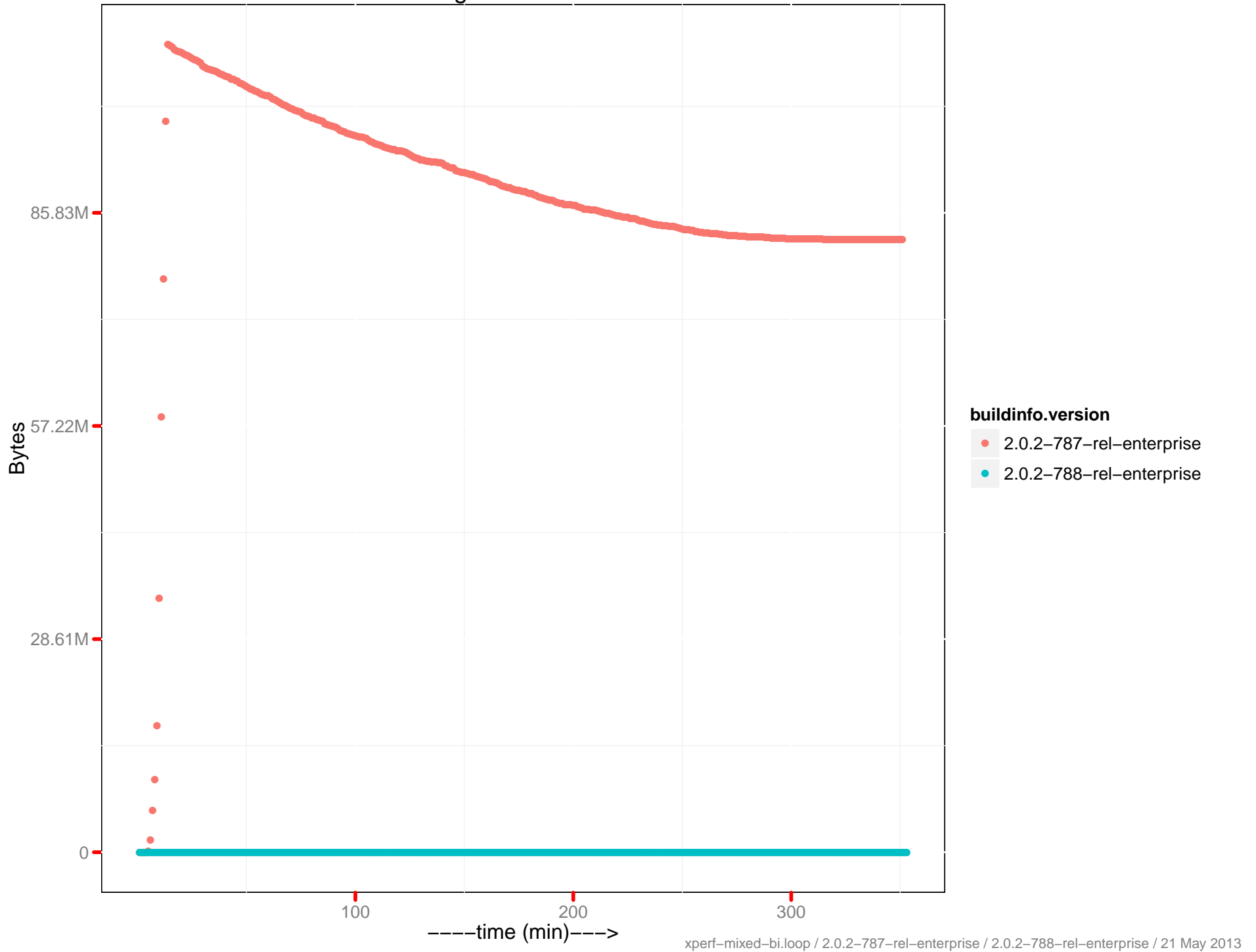


# SWAP Usage – 172.23.97.54:8091

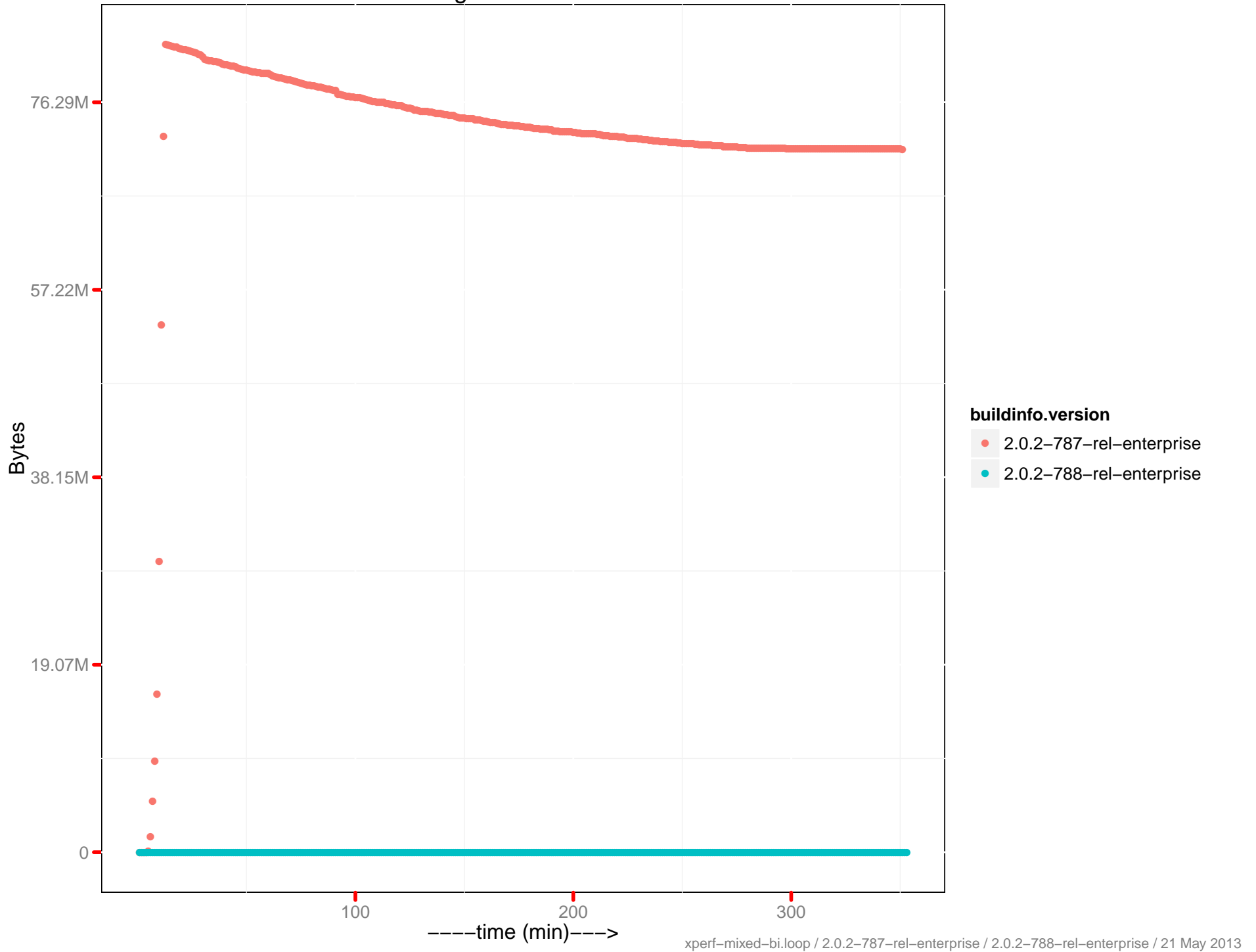




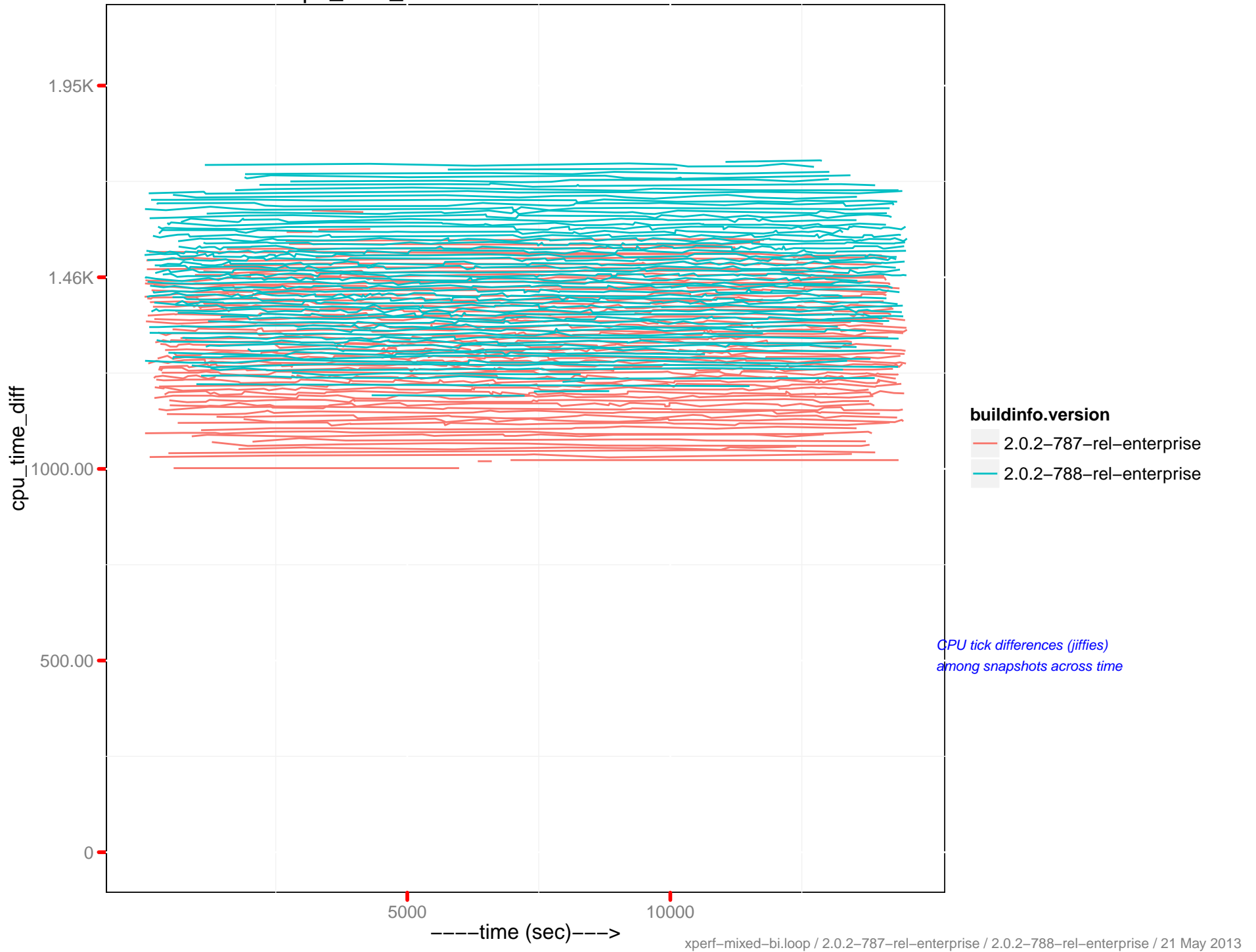
# SWAP Usage – 172.23.97.55:8091



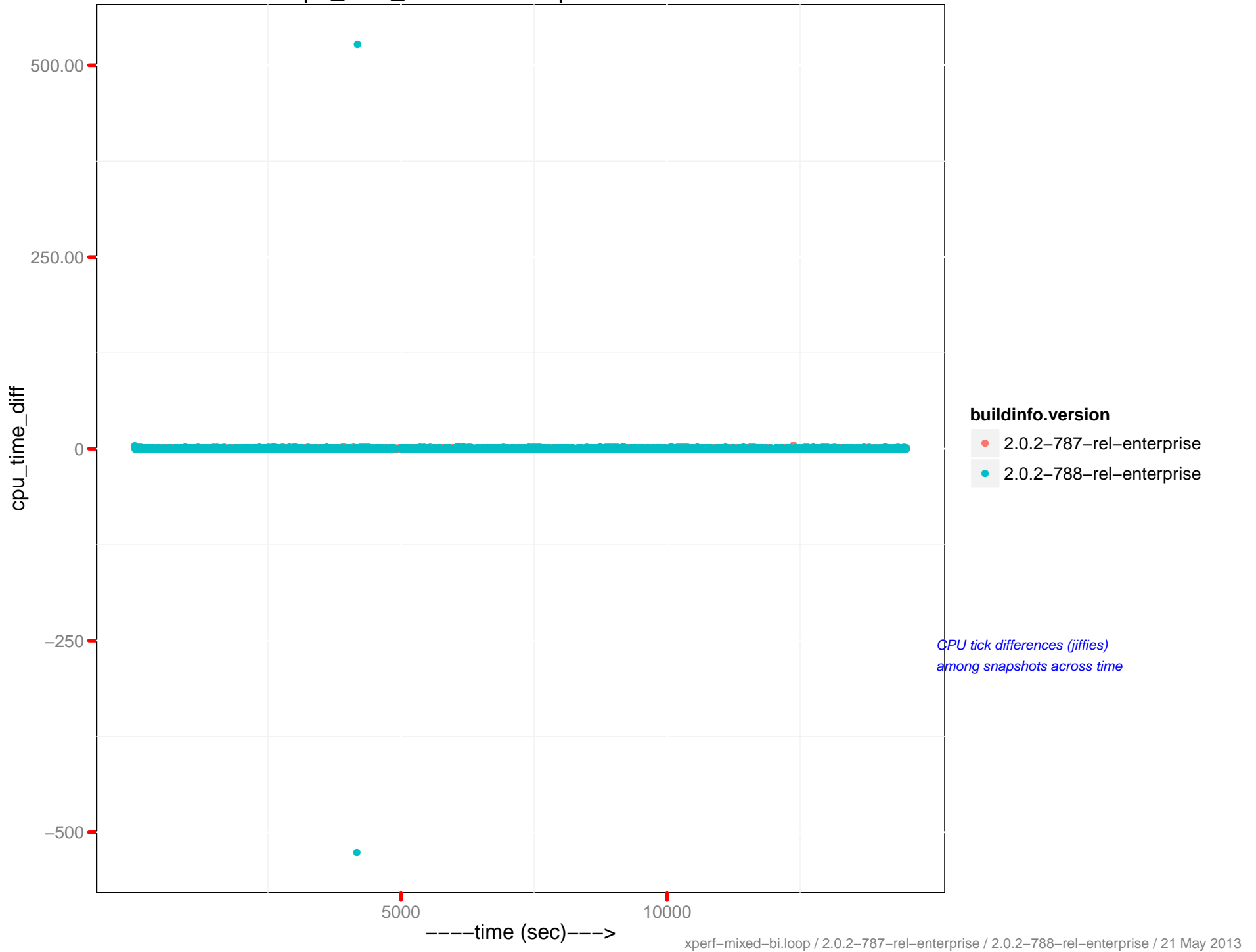
# SWAP Usage – 172.23.97.56:8091



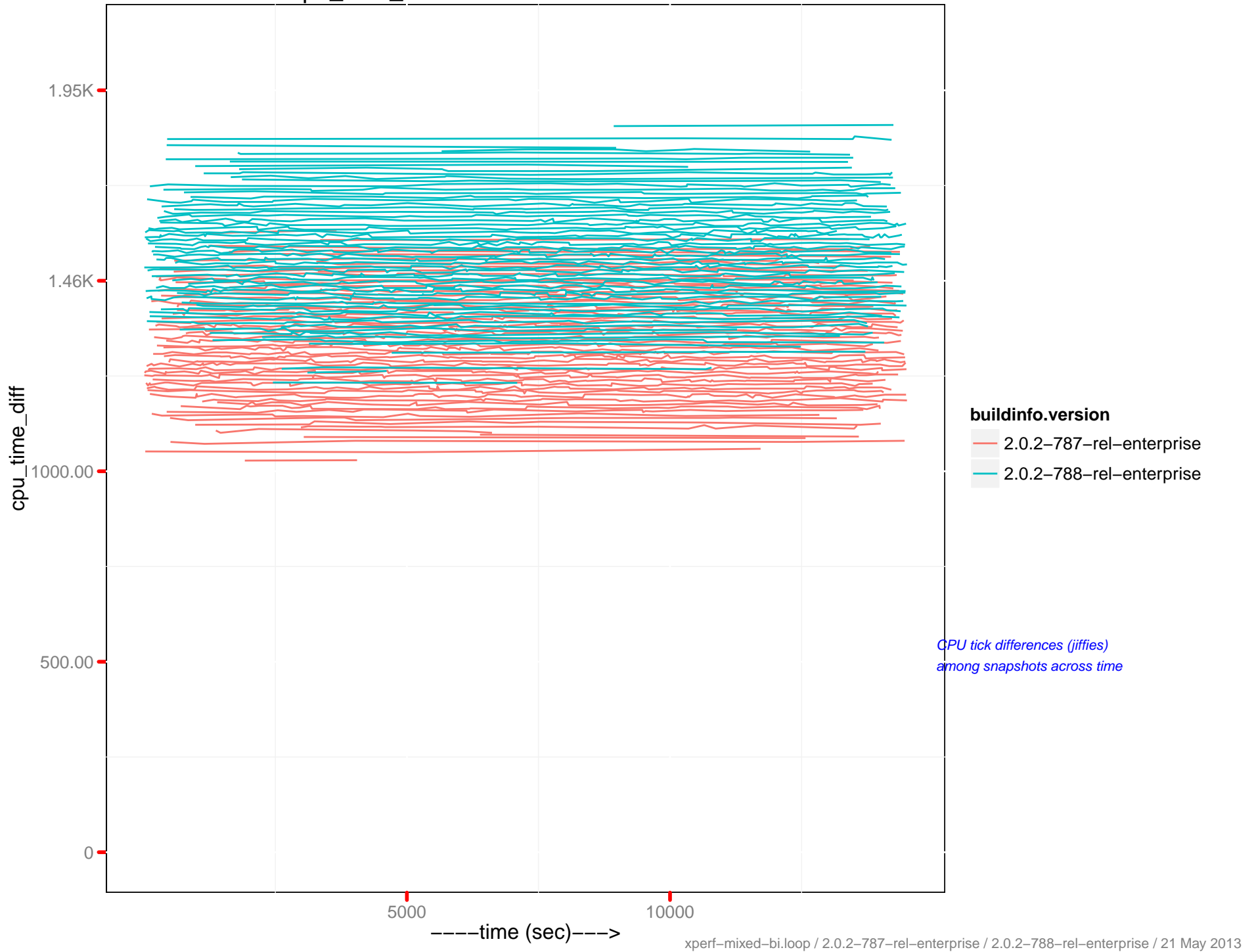
# cpu\_time\_diff: memcached - 172.23.97.53



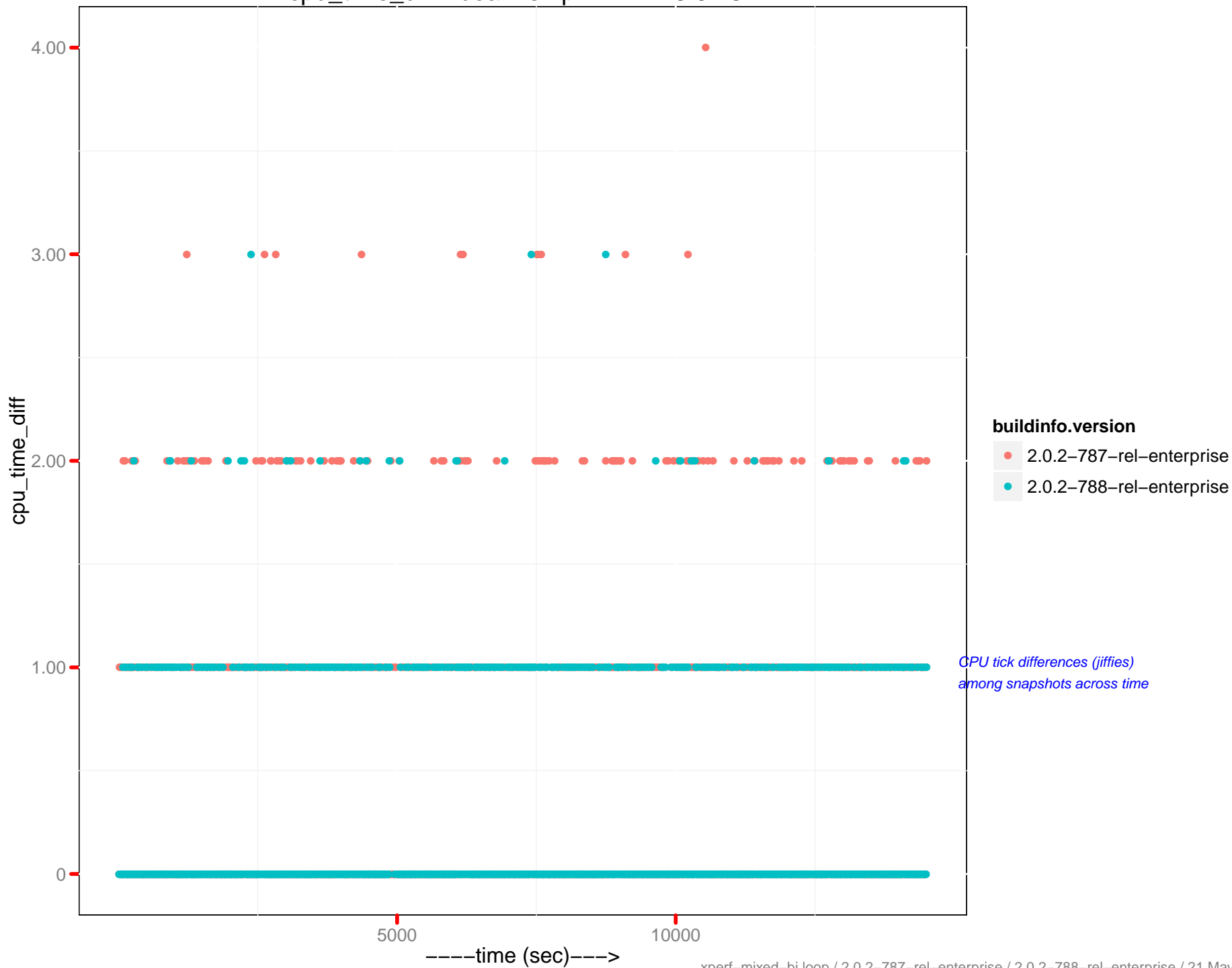
cpu\_time\_diff : beam.smp - 172.23.97.53



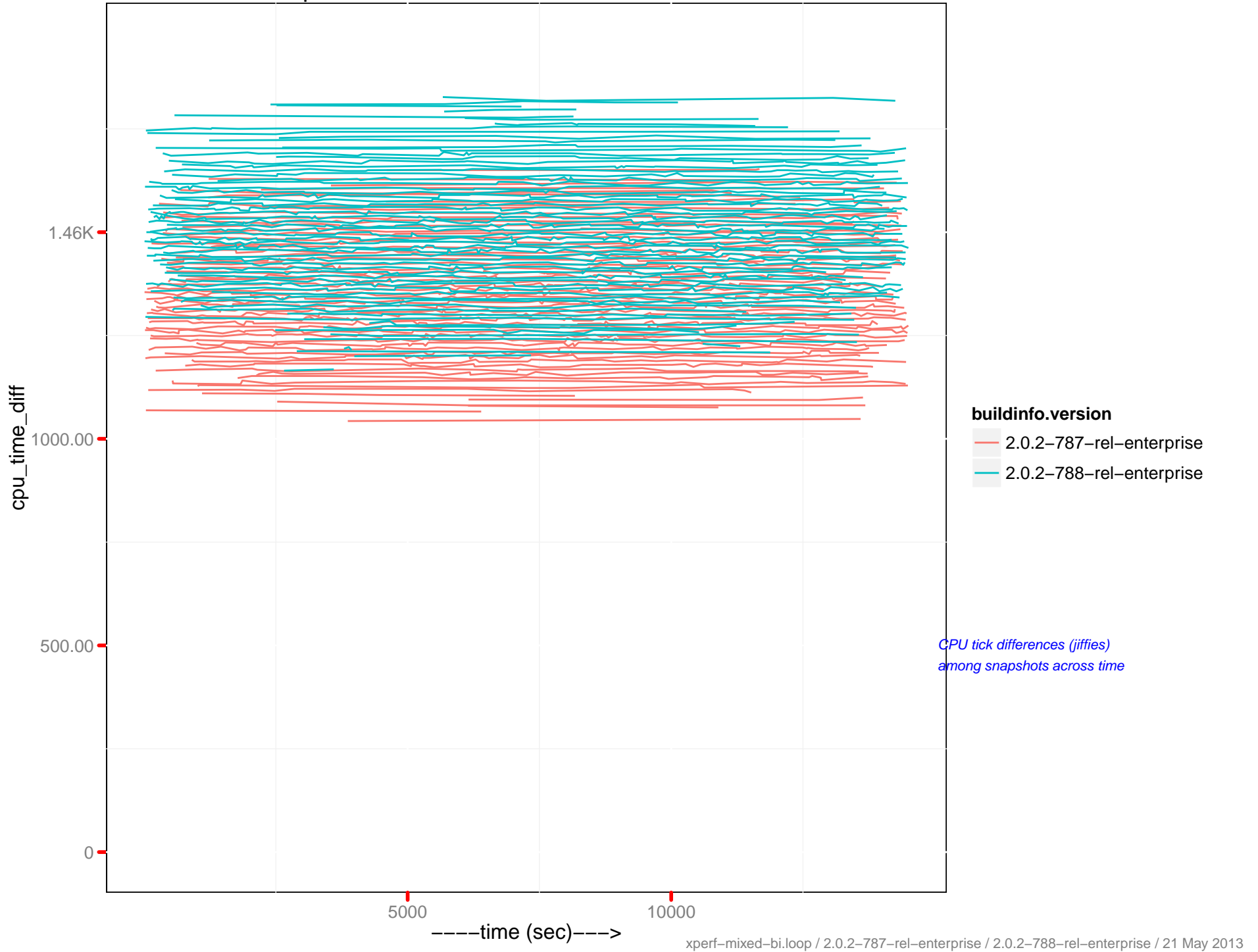
cpu\_time\_diff: memcached - 172.23.97.54



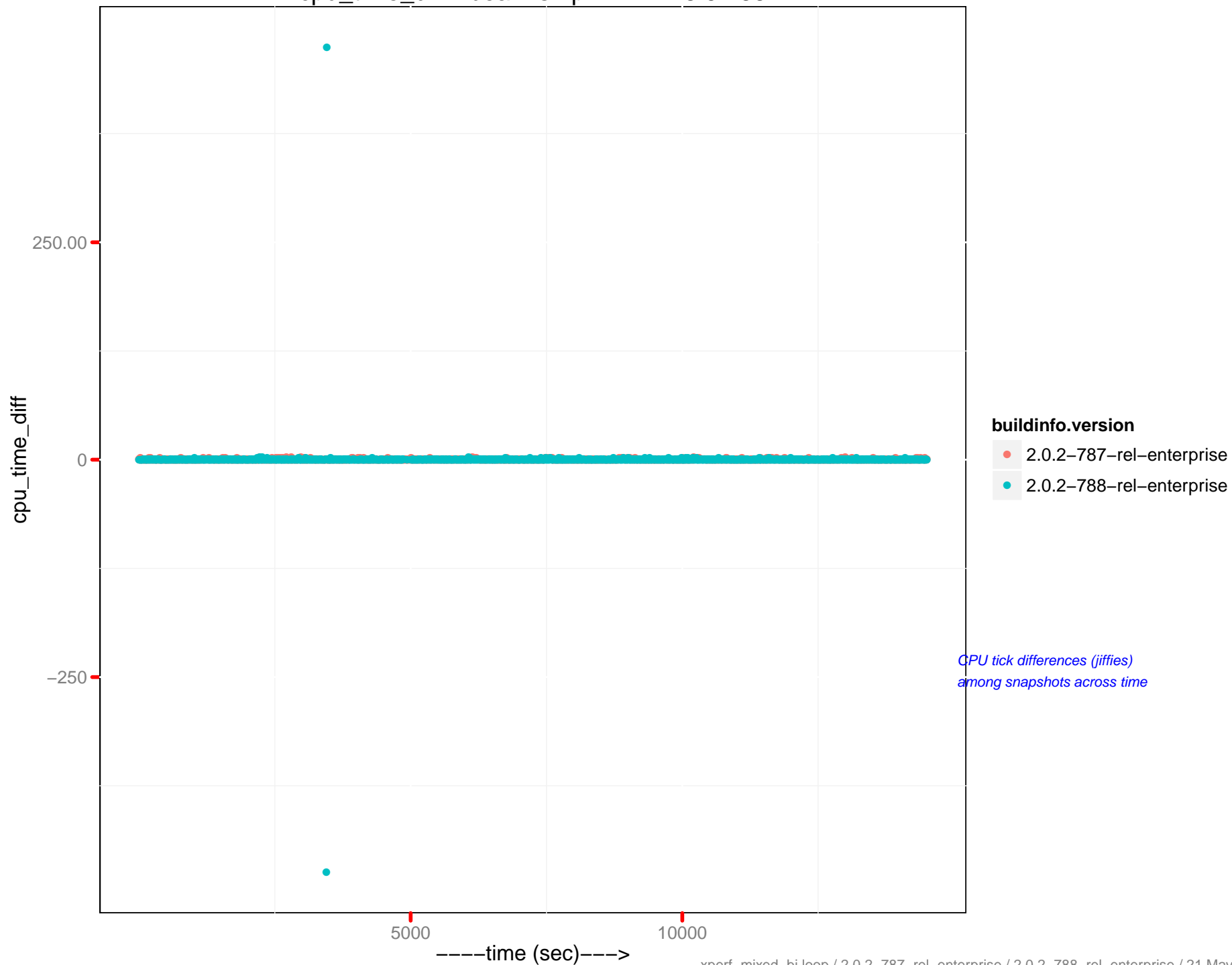
cpu\_time\_diff : beam.smp - 172.23.97.54



cpu\_time\_diff: memcached - 172.23.97.55



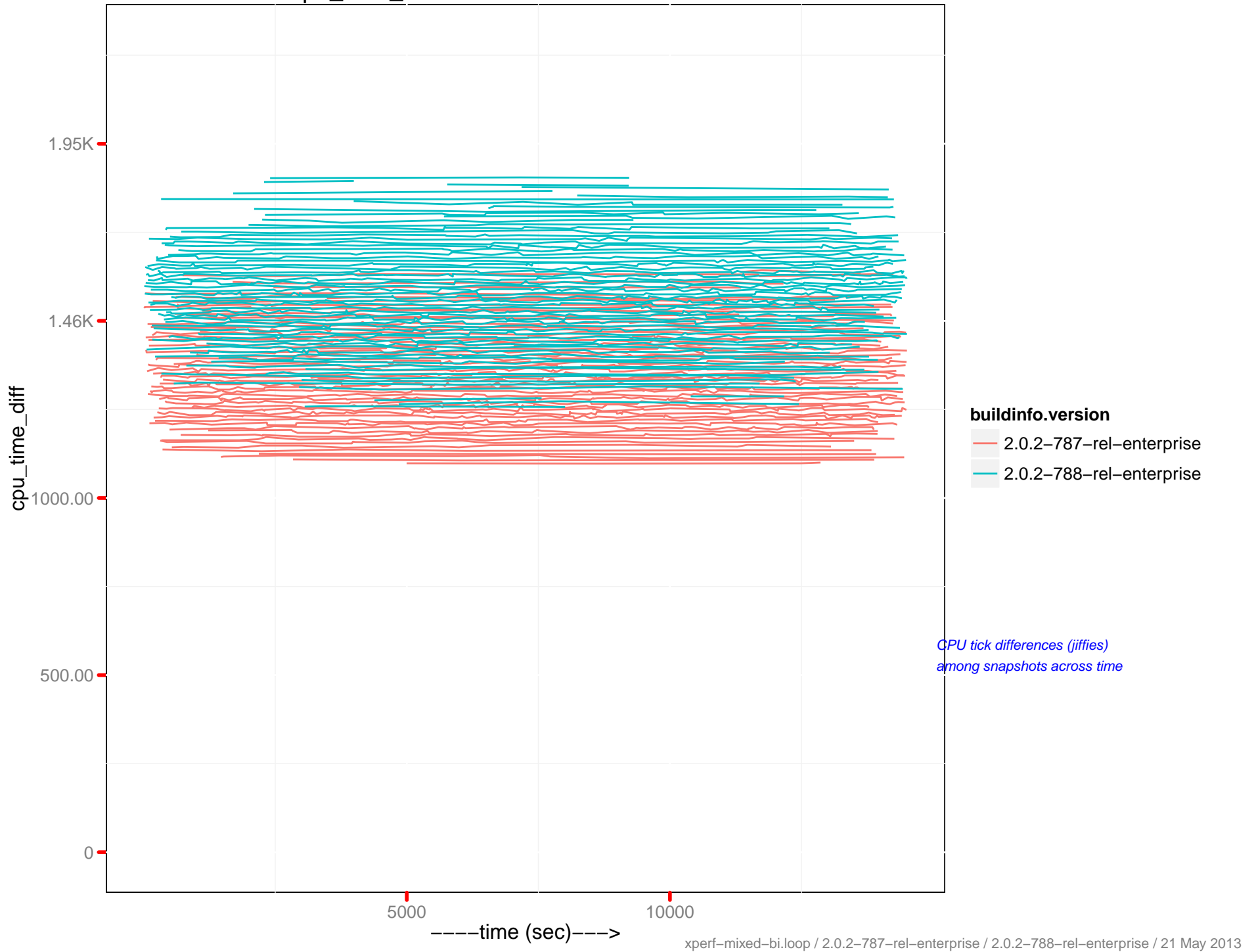
cpu\_time\_diff : beam.smp - 172.23.97.55



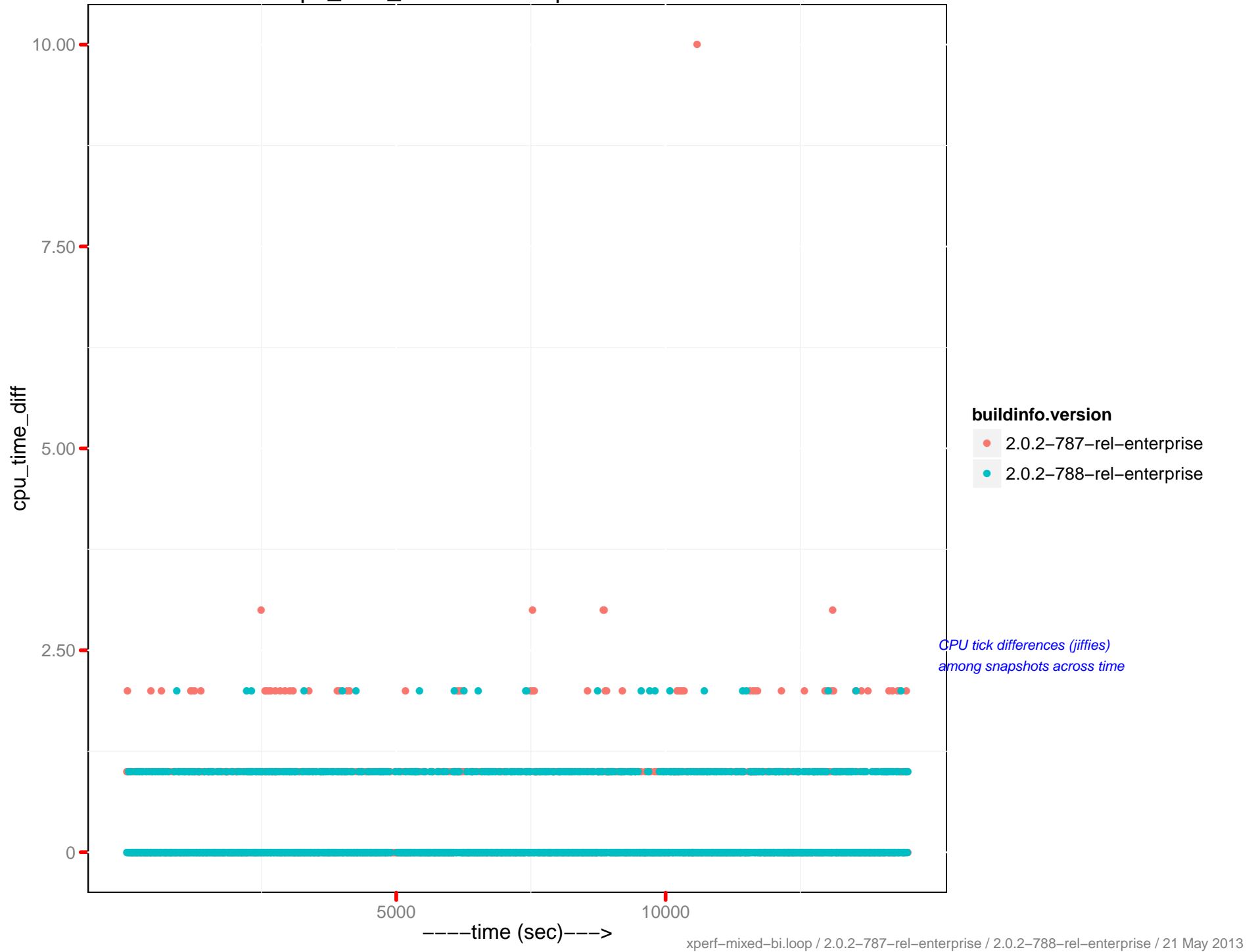
*CPU tick differences (jiffies)  
among snapshots across time*



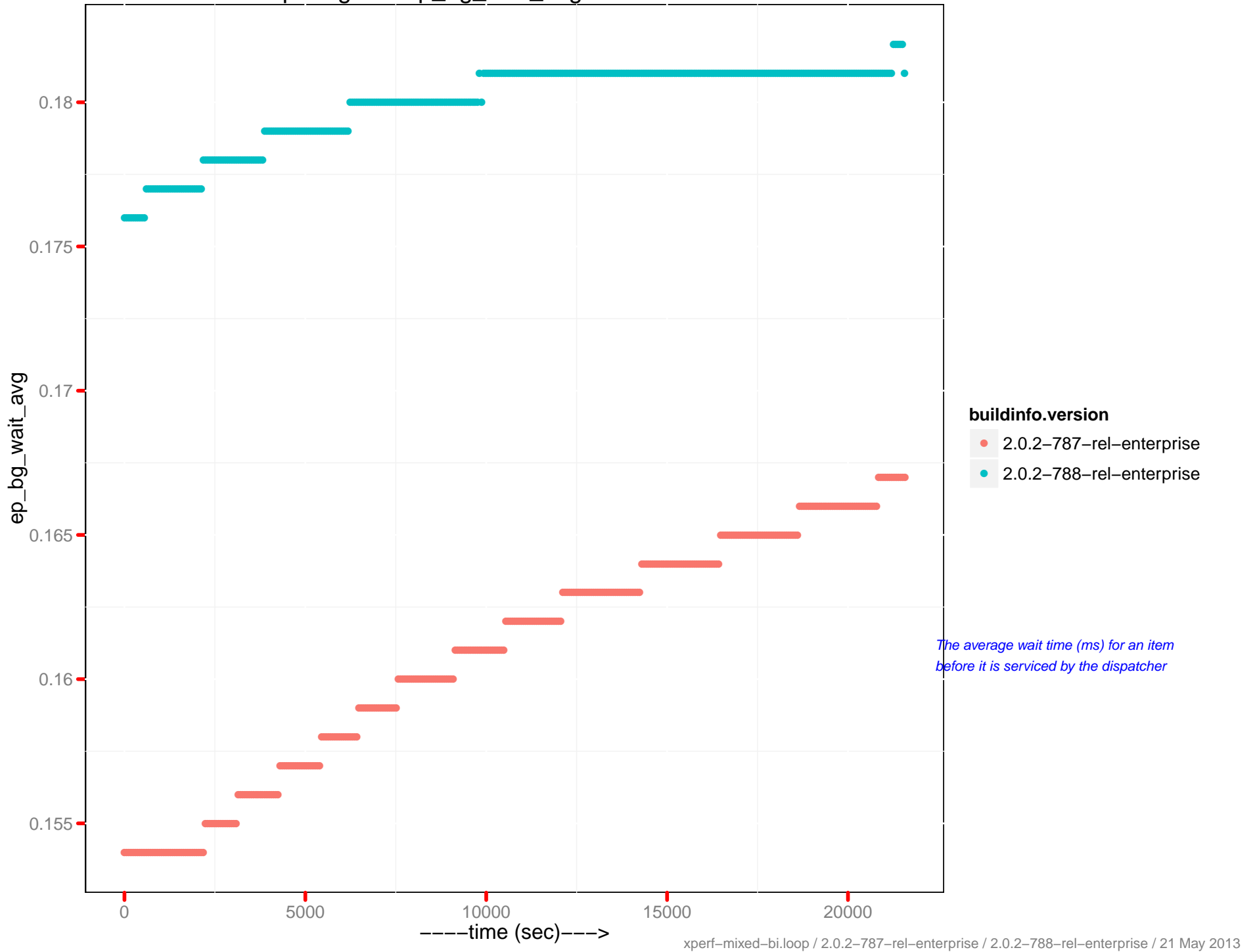
cpu\_time\_diff: memcached - 172.23.97.56



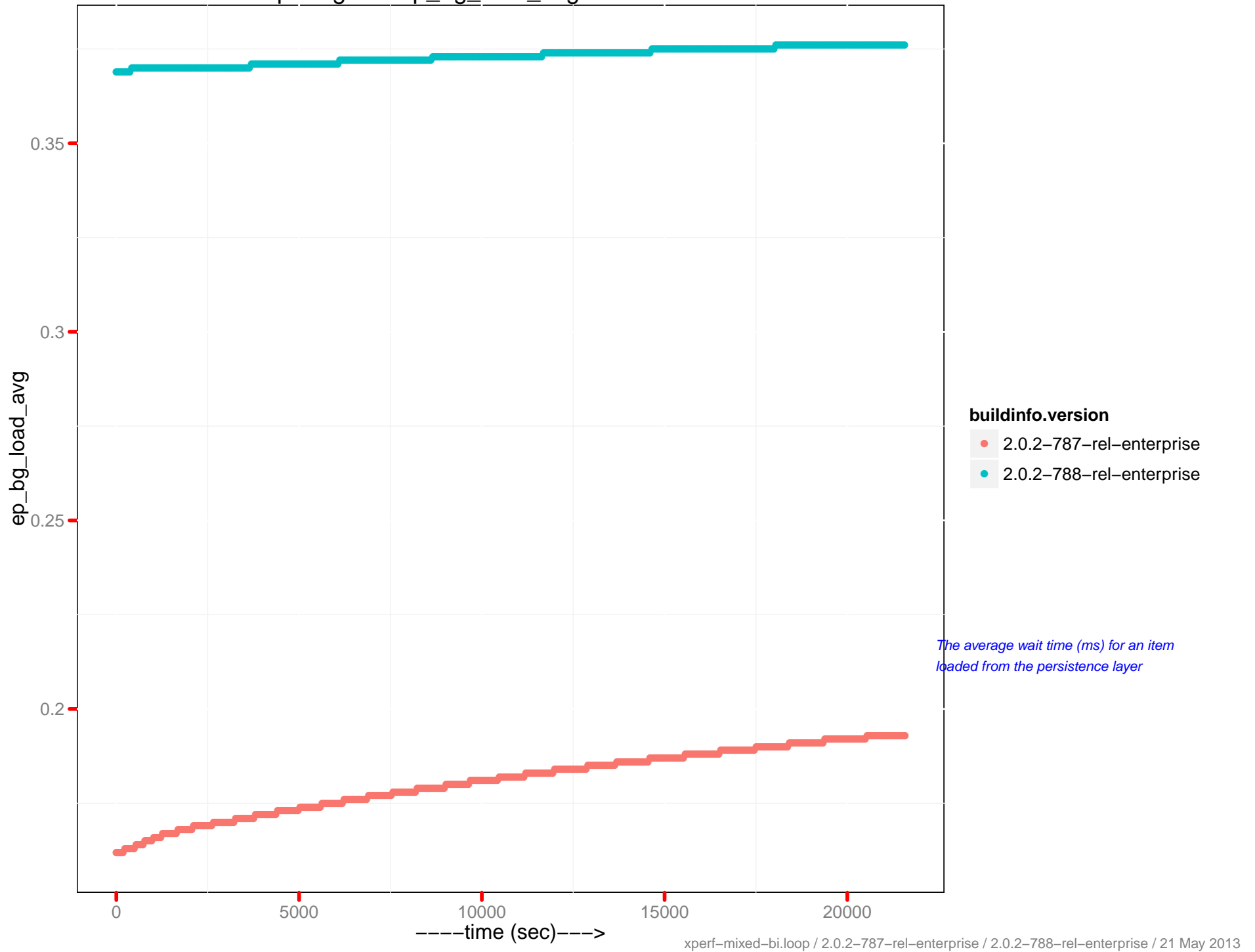
# cpu\_time\_diff : beam.smp - 172.23.97.56



ep-engine : ep\_bg\_wait\_avg - 172.23.97.53

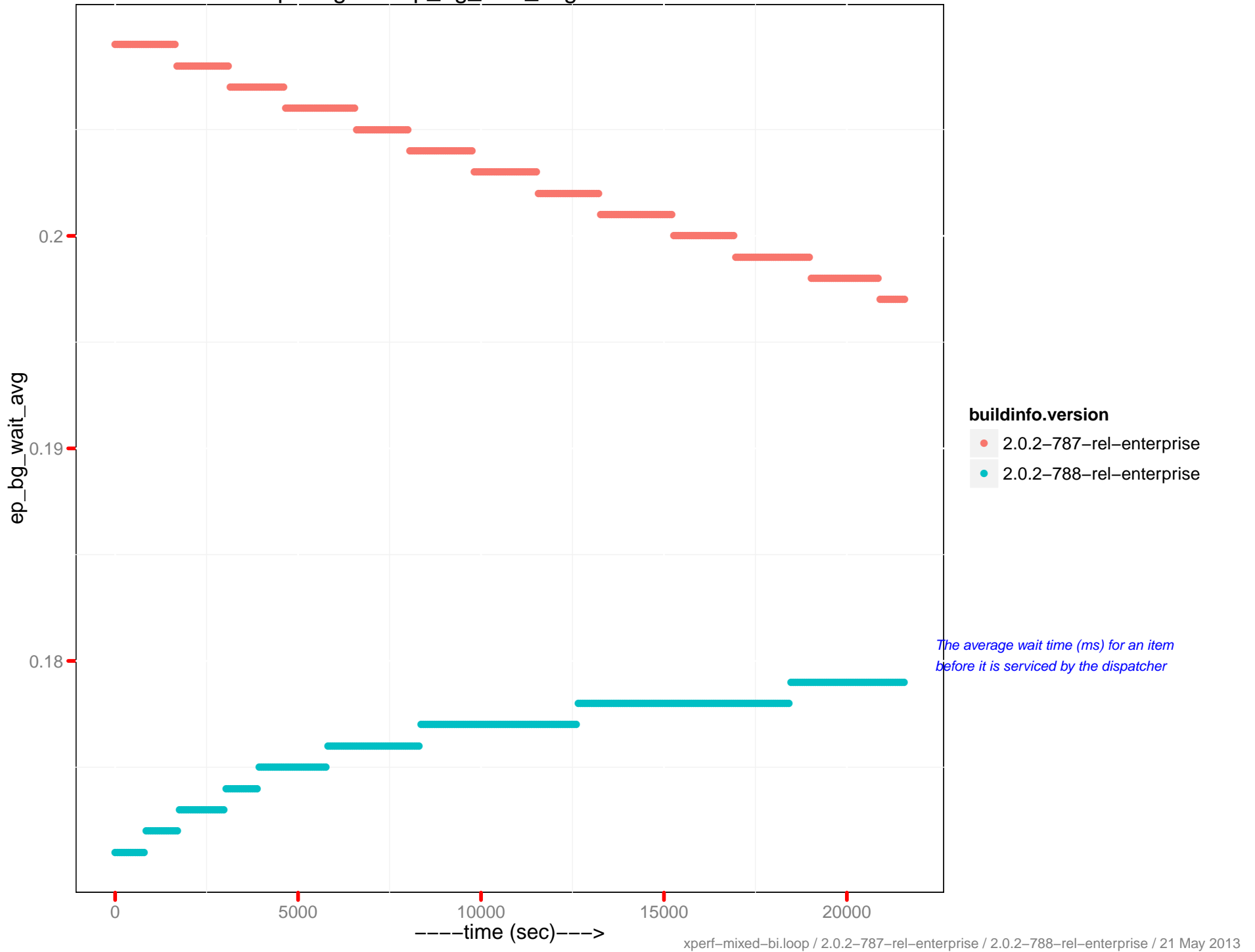


# ep-engine : ep\_bg\_load\_avg - 172.23.97.53

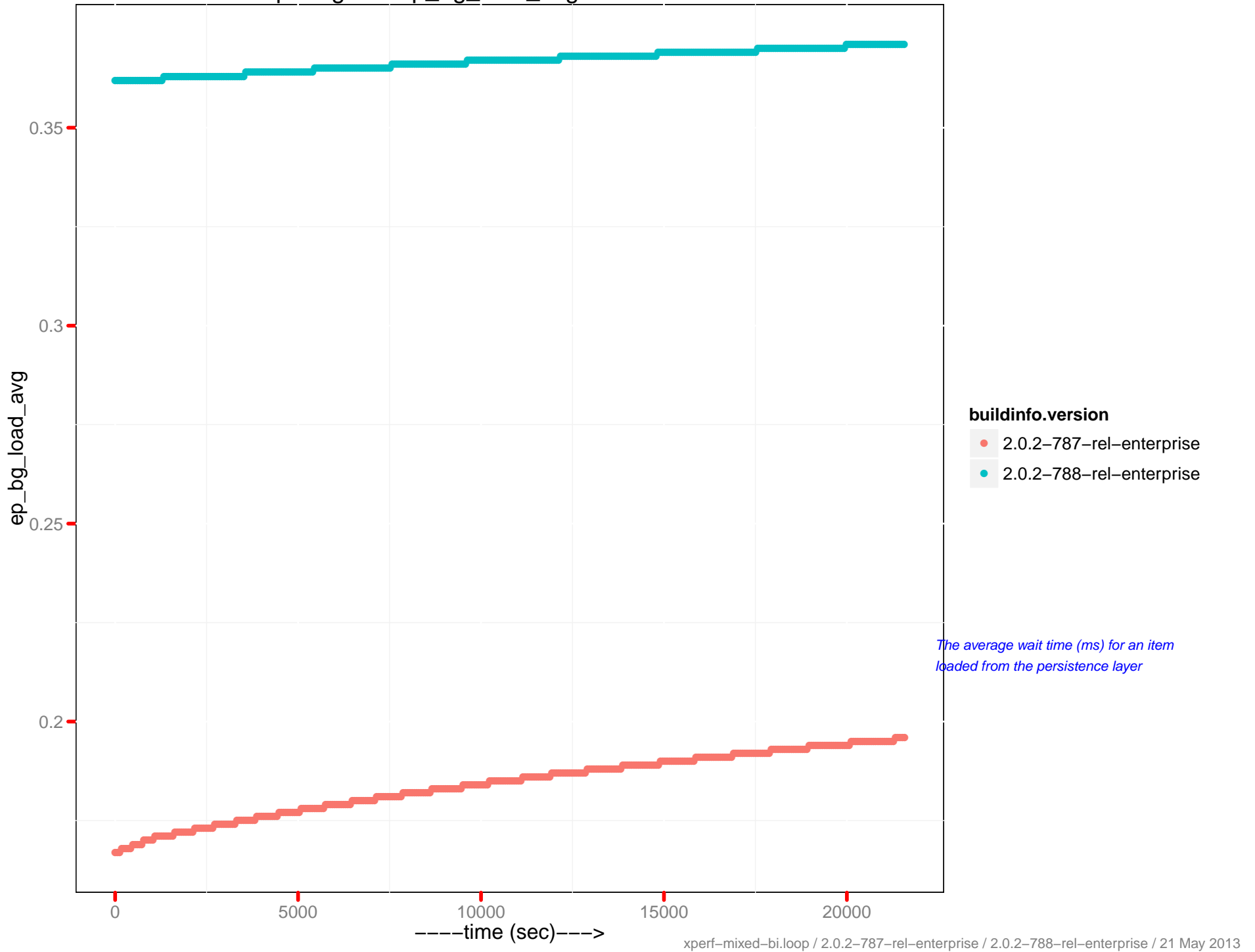


*The average wait time (ms) for an item loaded from the persistence layer*

ep-engine : ep\_bg\_wait\_avg - 172.23.97.54

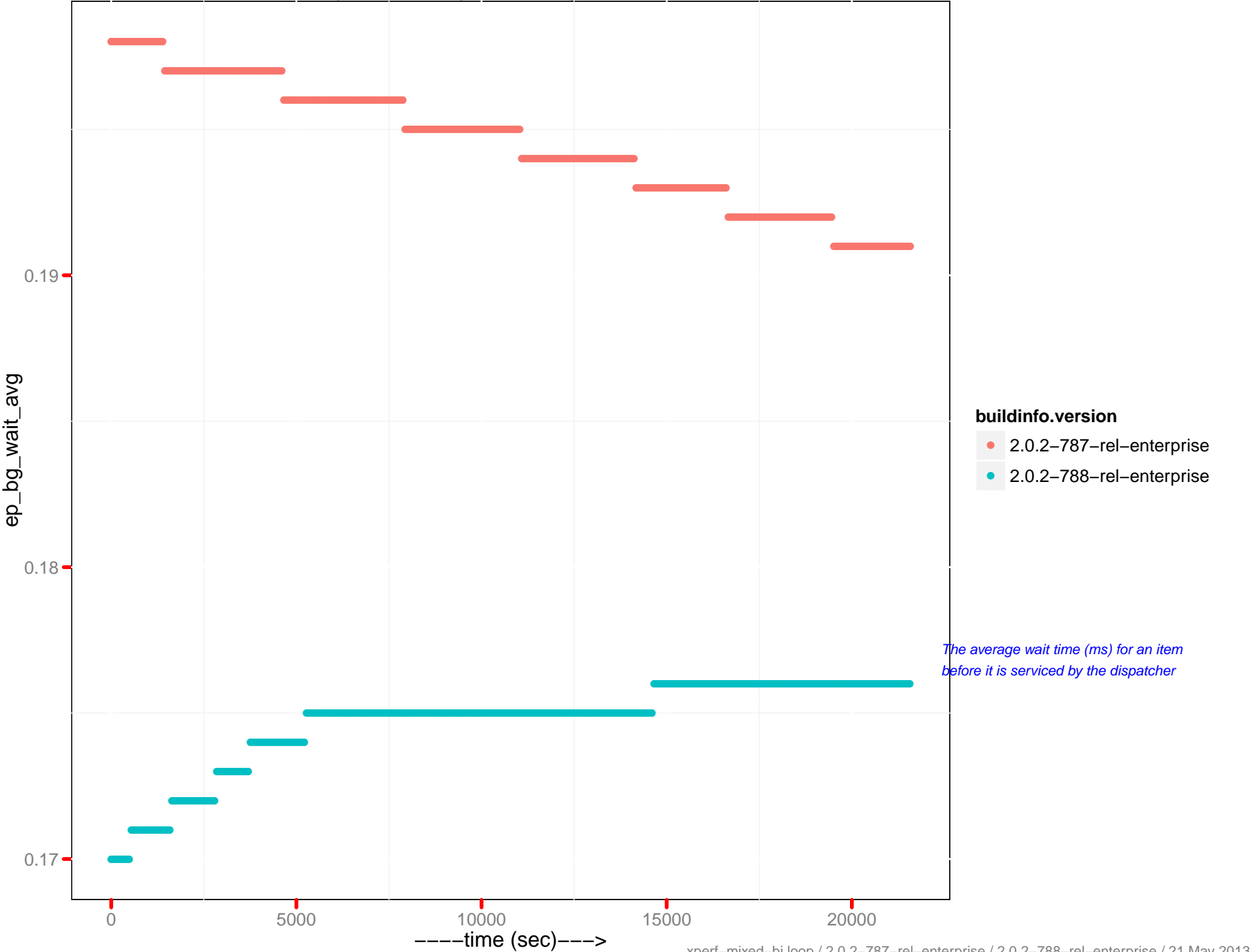


ep-engine : ep\_bg\_load\_avg - 172.23.97.54



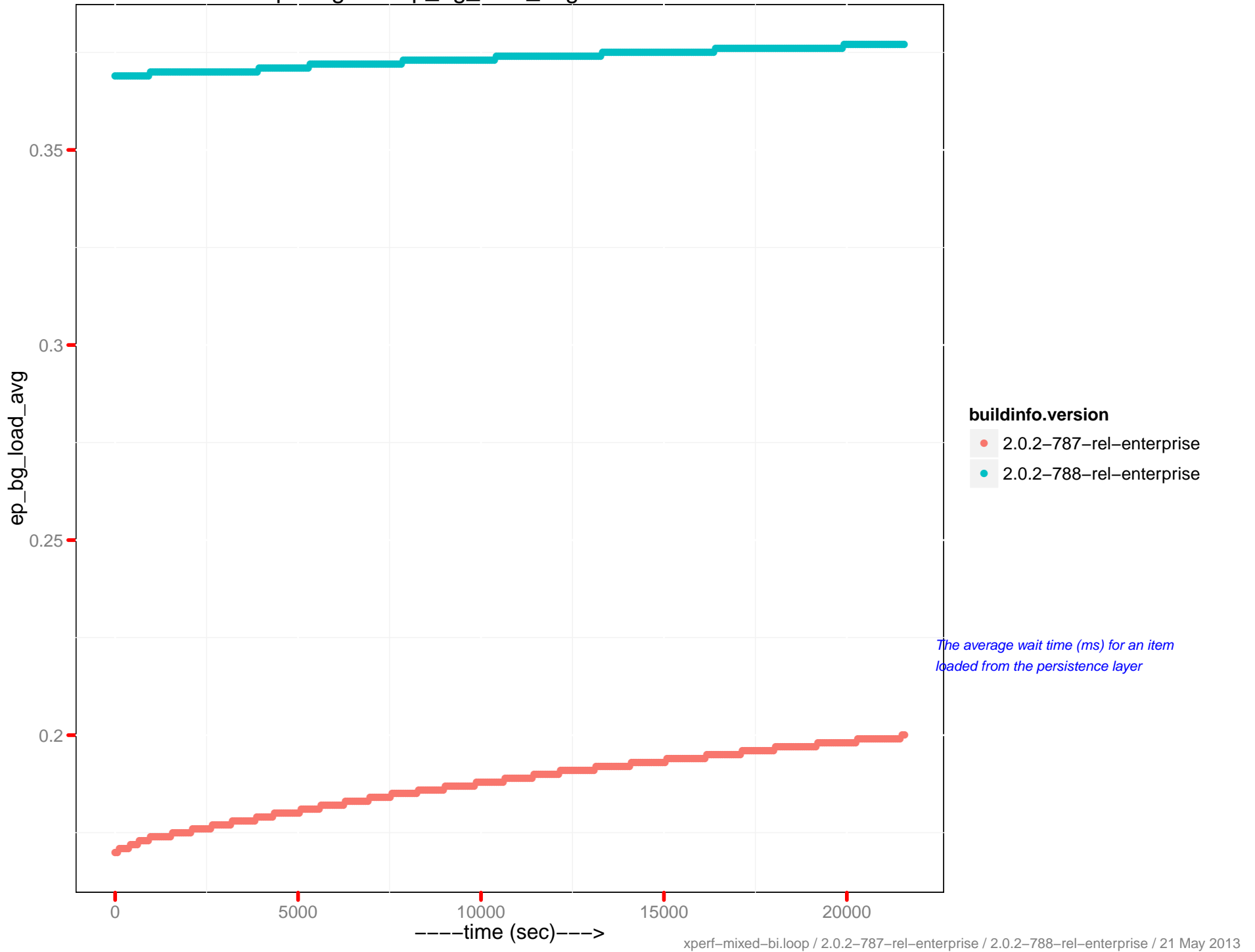
*The average wait time (ms) for an item loaded from the persistence layer*

ep-engine : ep\_bg\_wait\_avg - 172.23.97.55



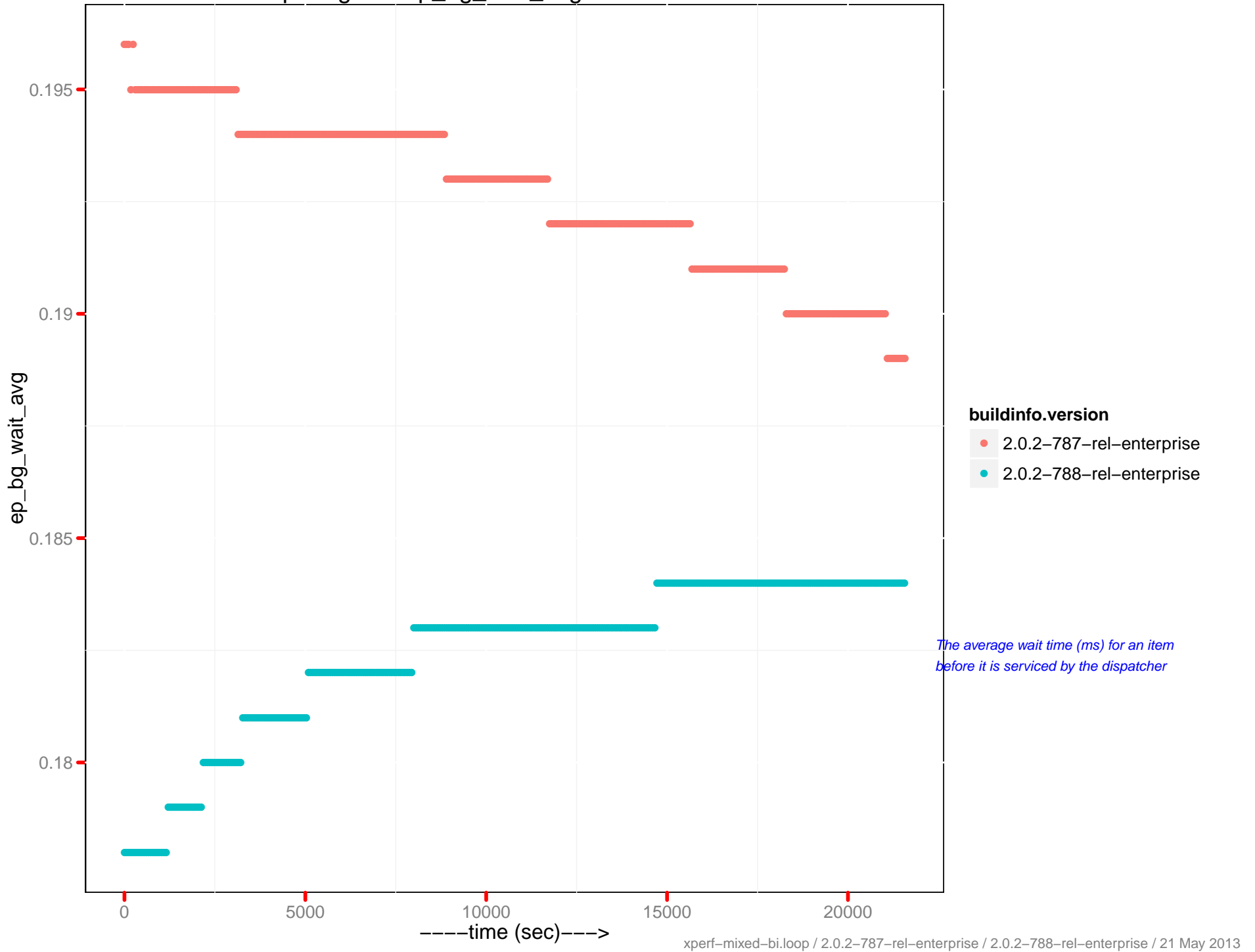
The average wait time (ms) for an item before it is serviced by the dispatcher

ep-engine : ep\_bg\_load\_avg - 172.23.97.55

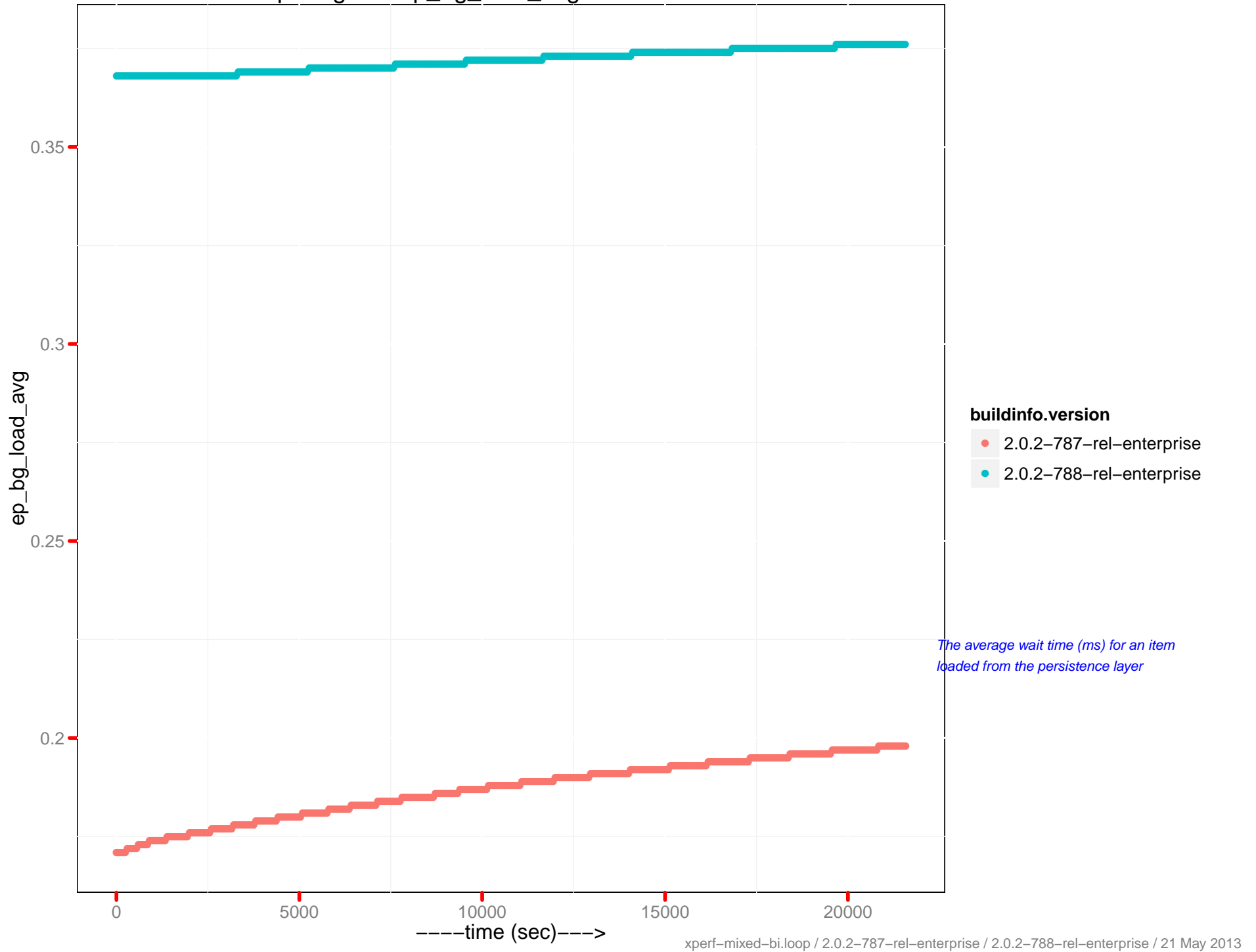




ep-engine : ep\_bg\_wait\_avg - 172.23.97.56

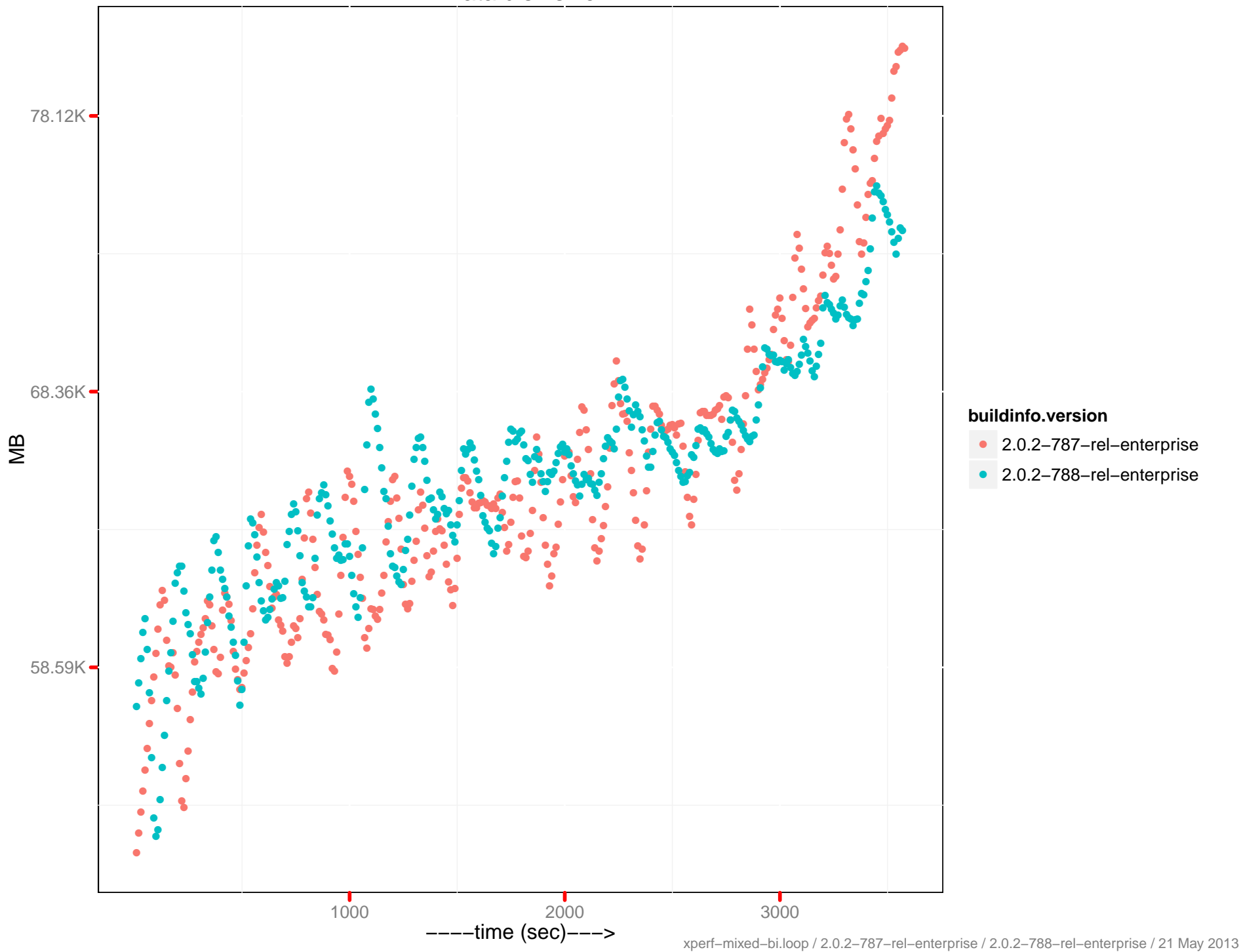


ep-engine : ep\_bg\_load\_avg - 172.23.97.56

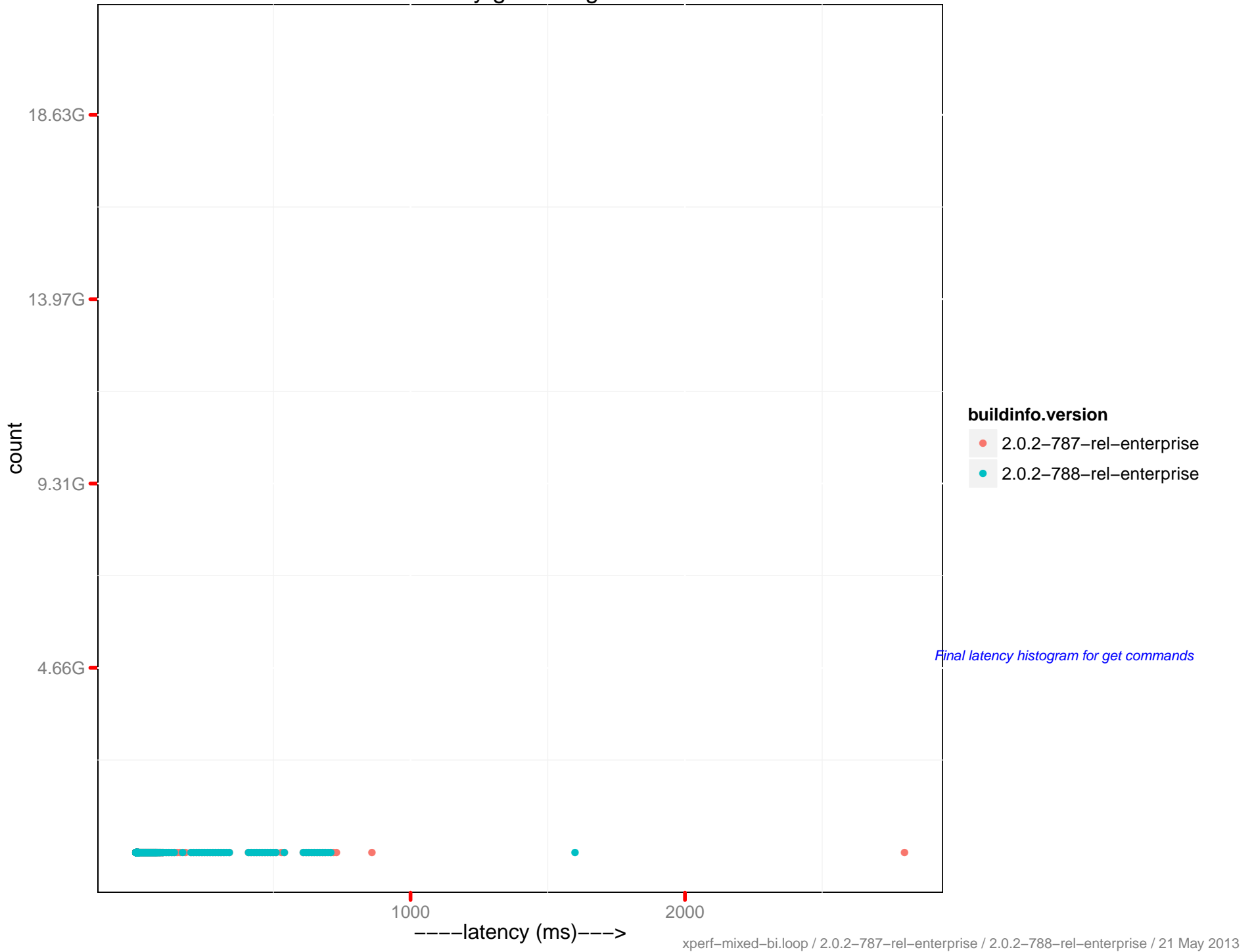


The average wait time (ms) for an item loaded from the persistence layer

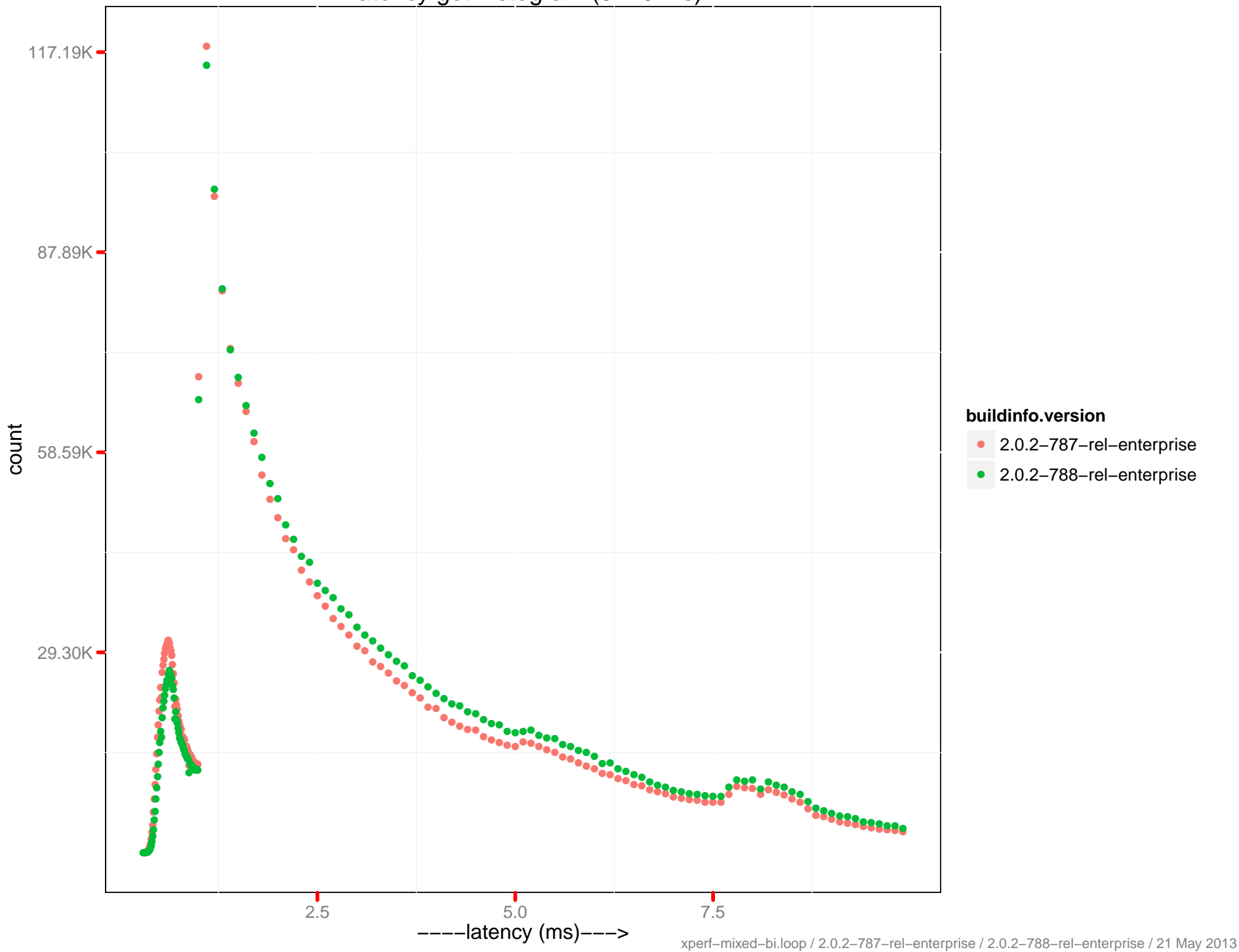
# Data disk size



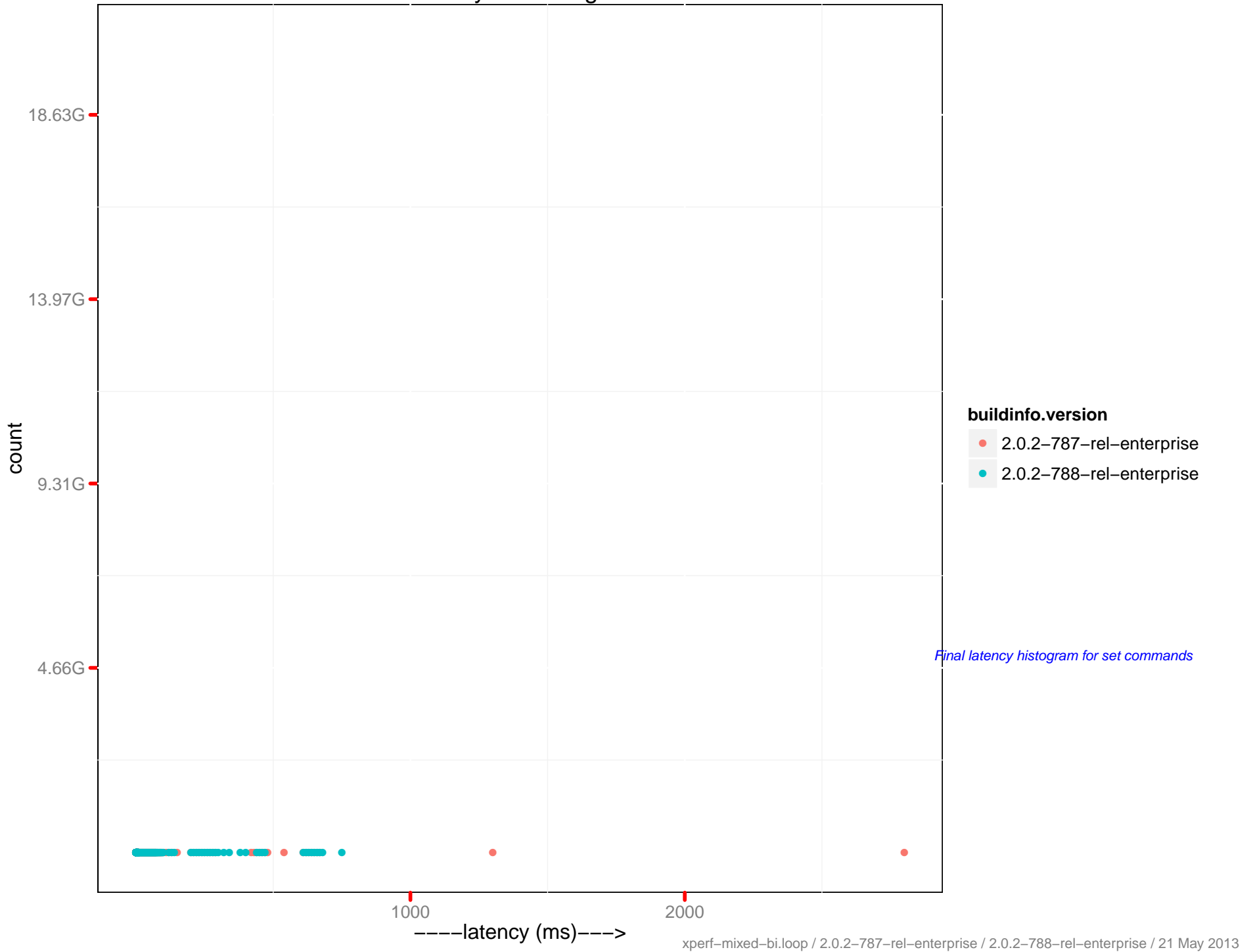
# Latency get histogram



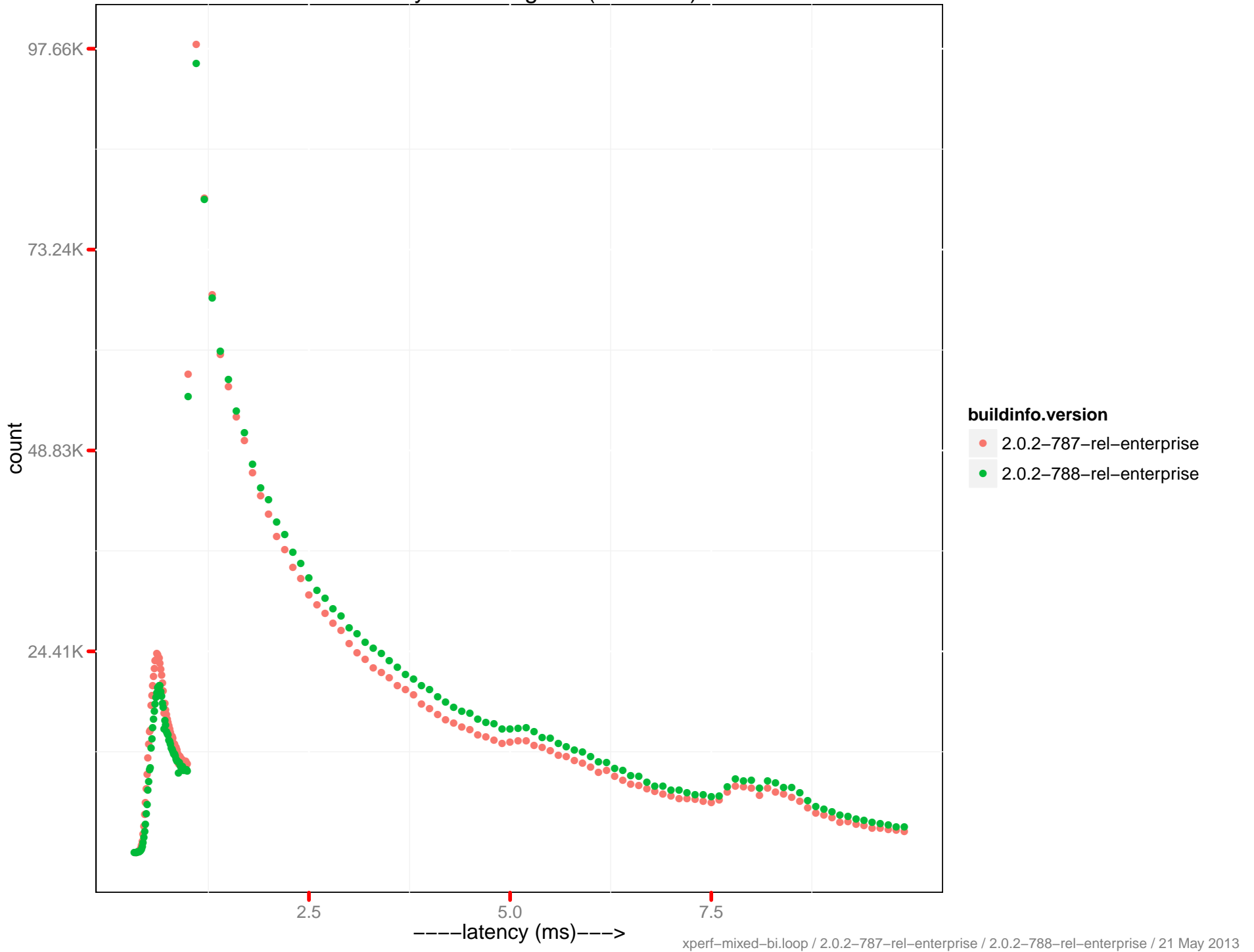
Latency get histogram (0–10 ms)



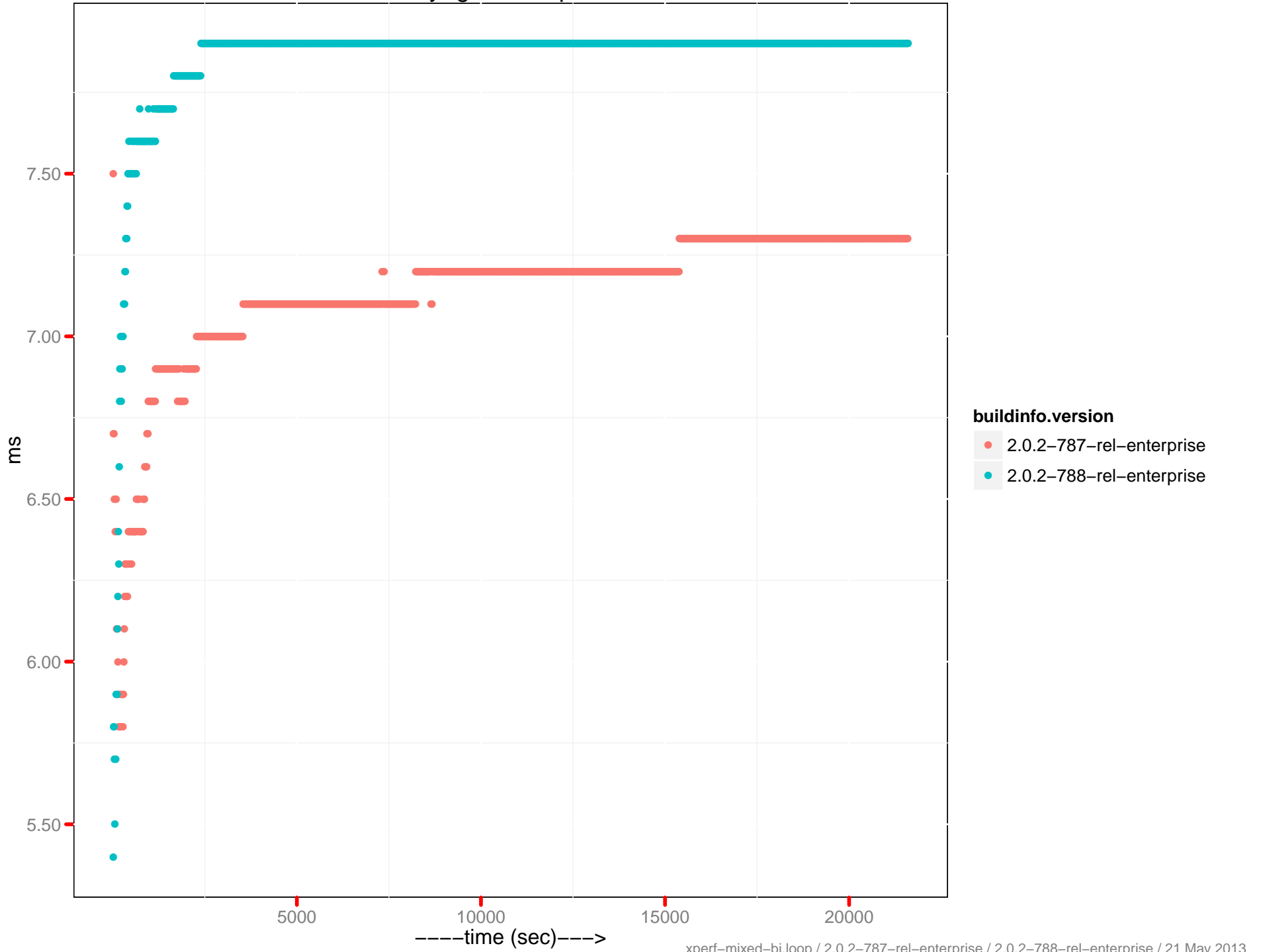
# Latency set histogram



Latency set histogram (0–10 ms)

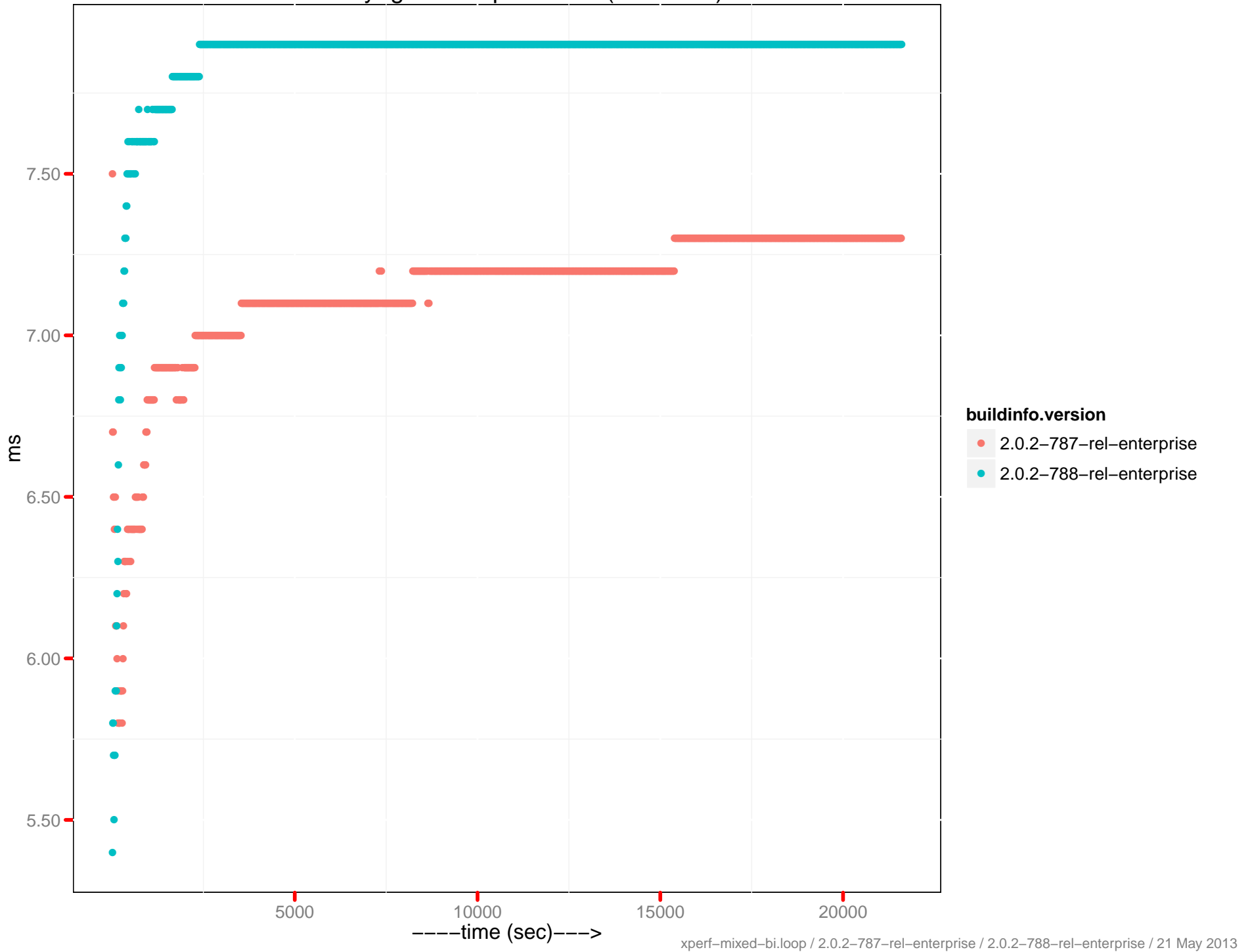


# Latency-get 90th percentile

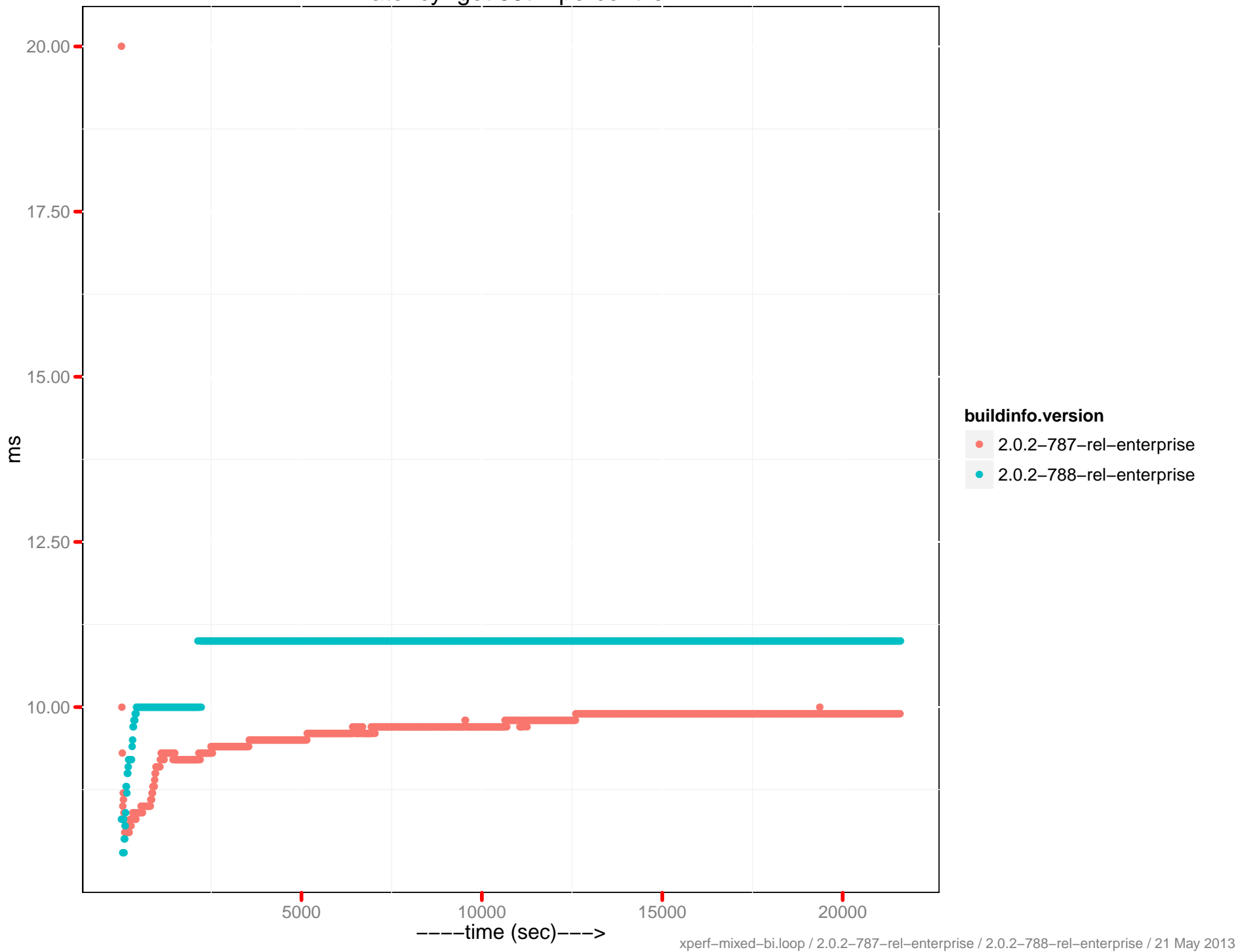




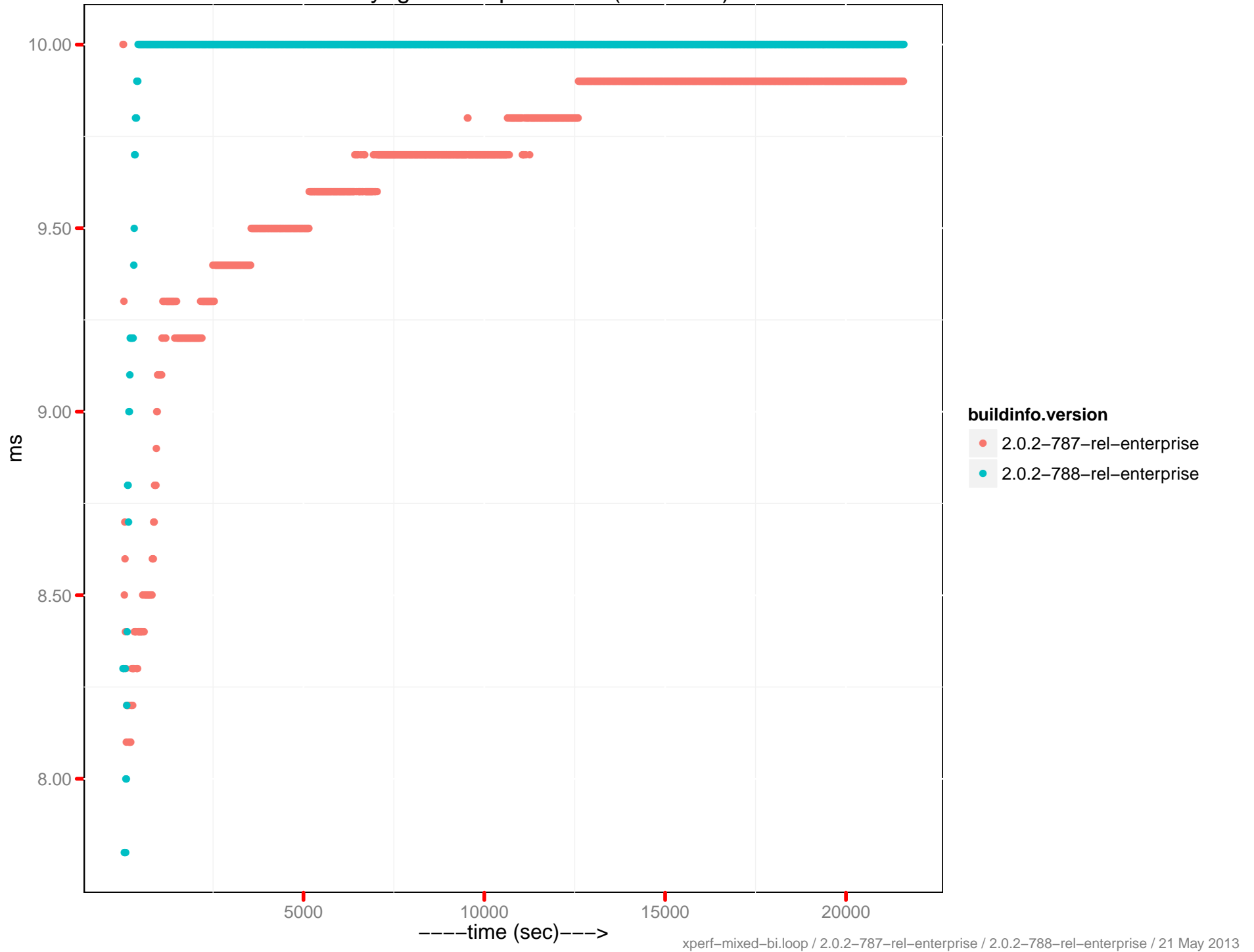
Latency-get 90th percentile (0 - 10ms)



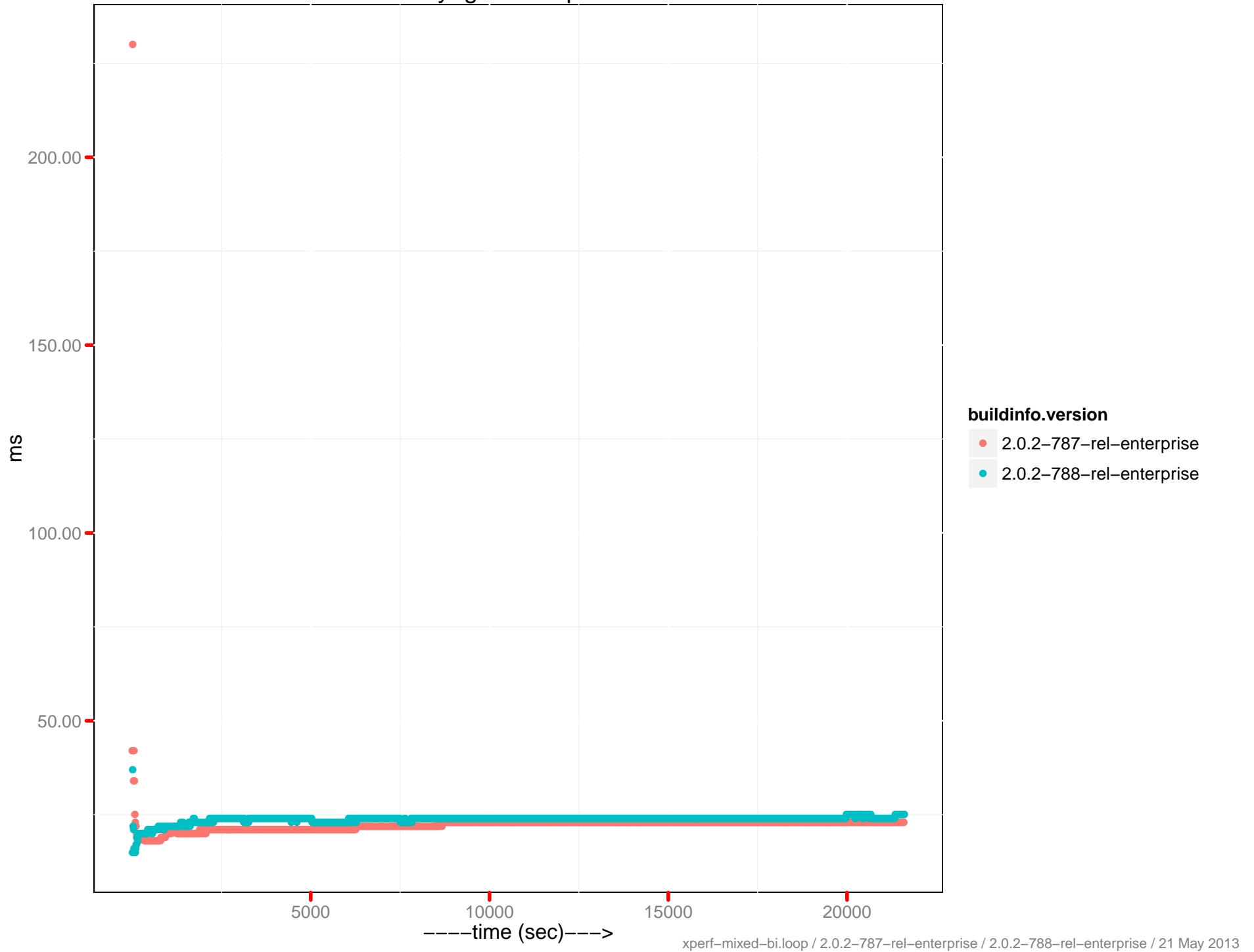
Latency-get 95th percentile



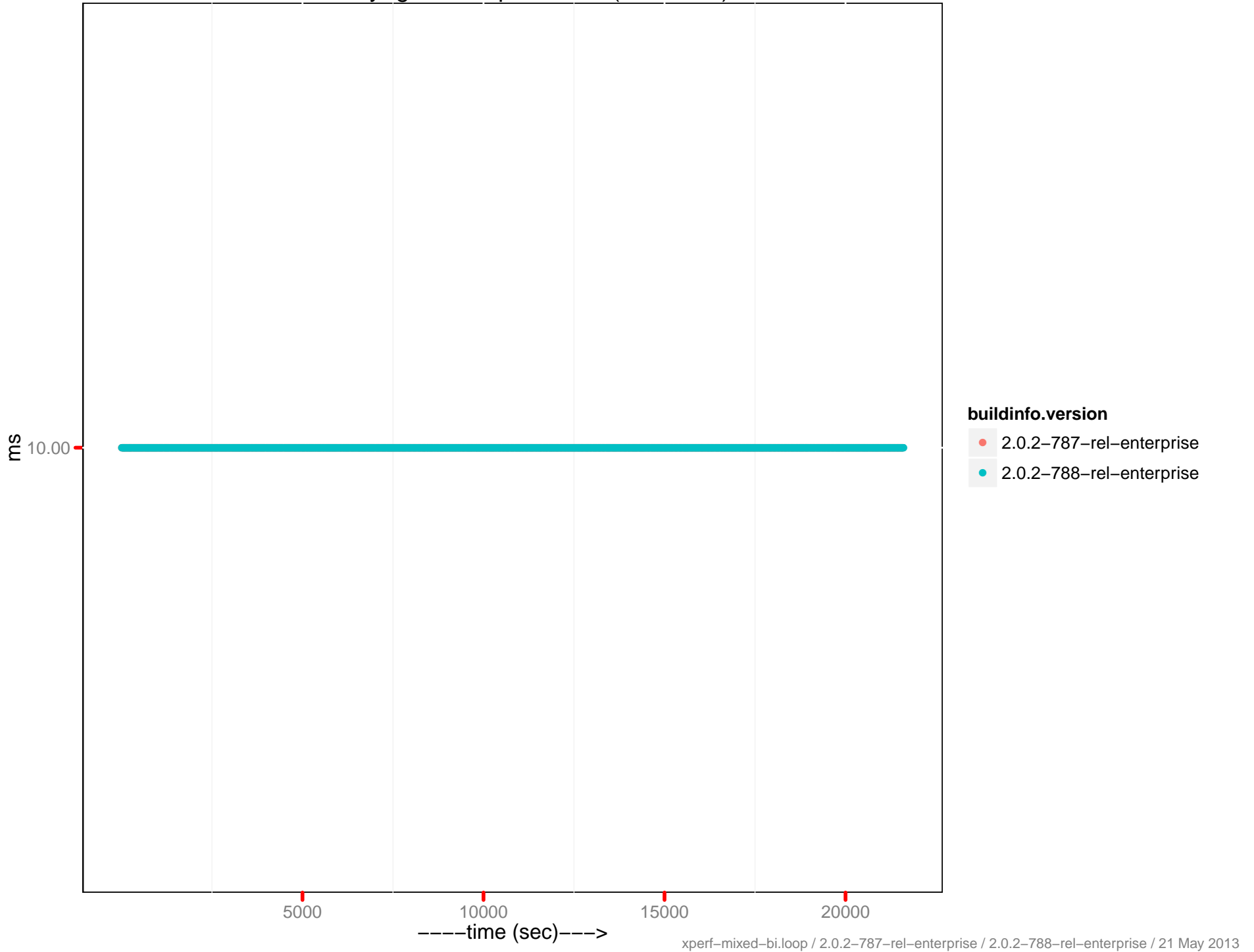
Latency-get 95th percentile (0 - 10ms)



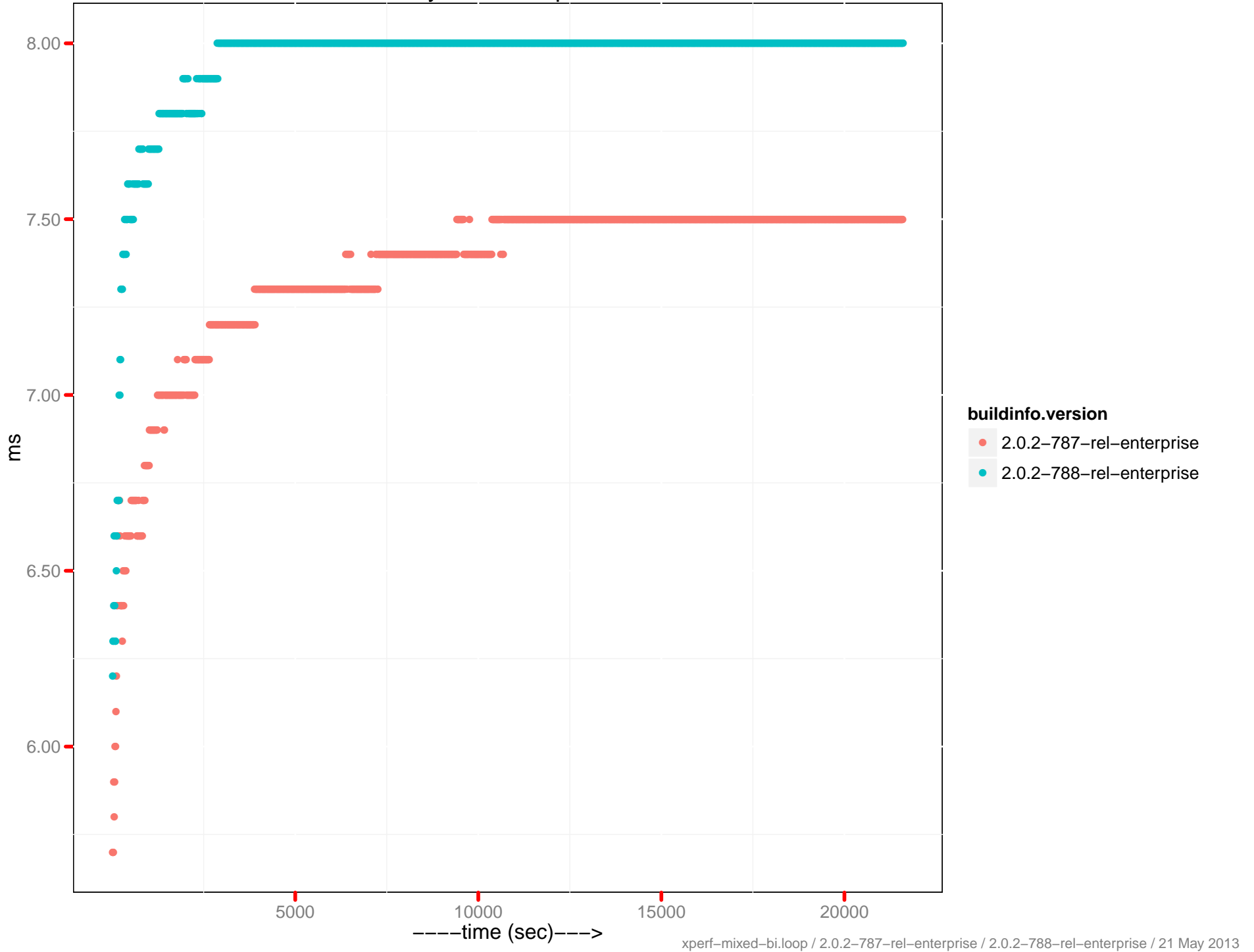
# Latency-get 99th percentile



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

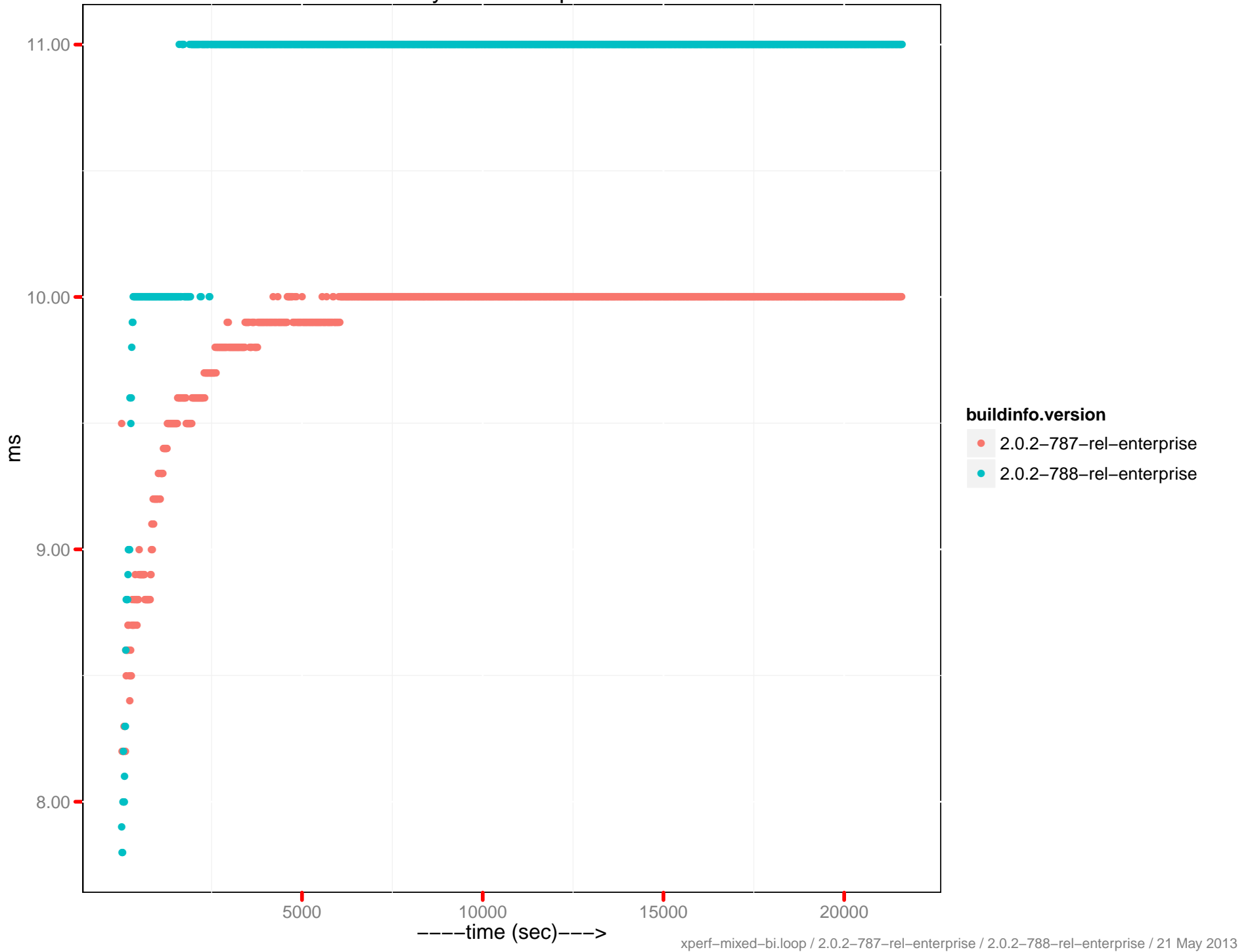


# Latency-set 90th percentile



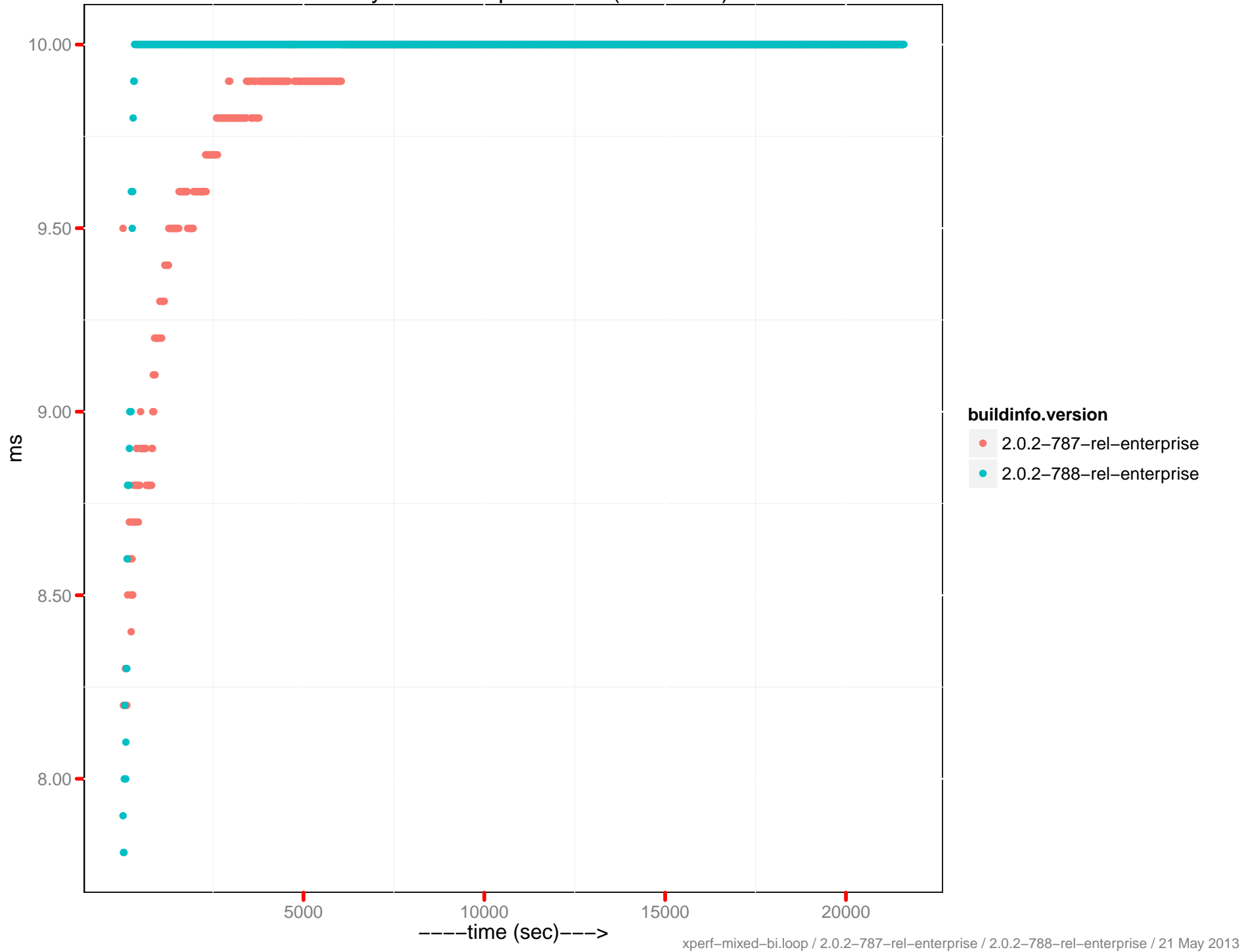


# Latency-set 95th percentile

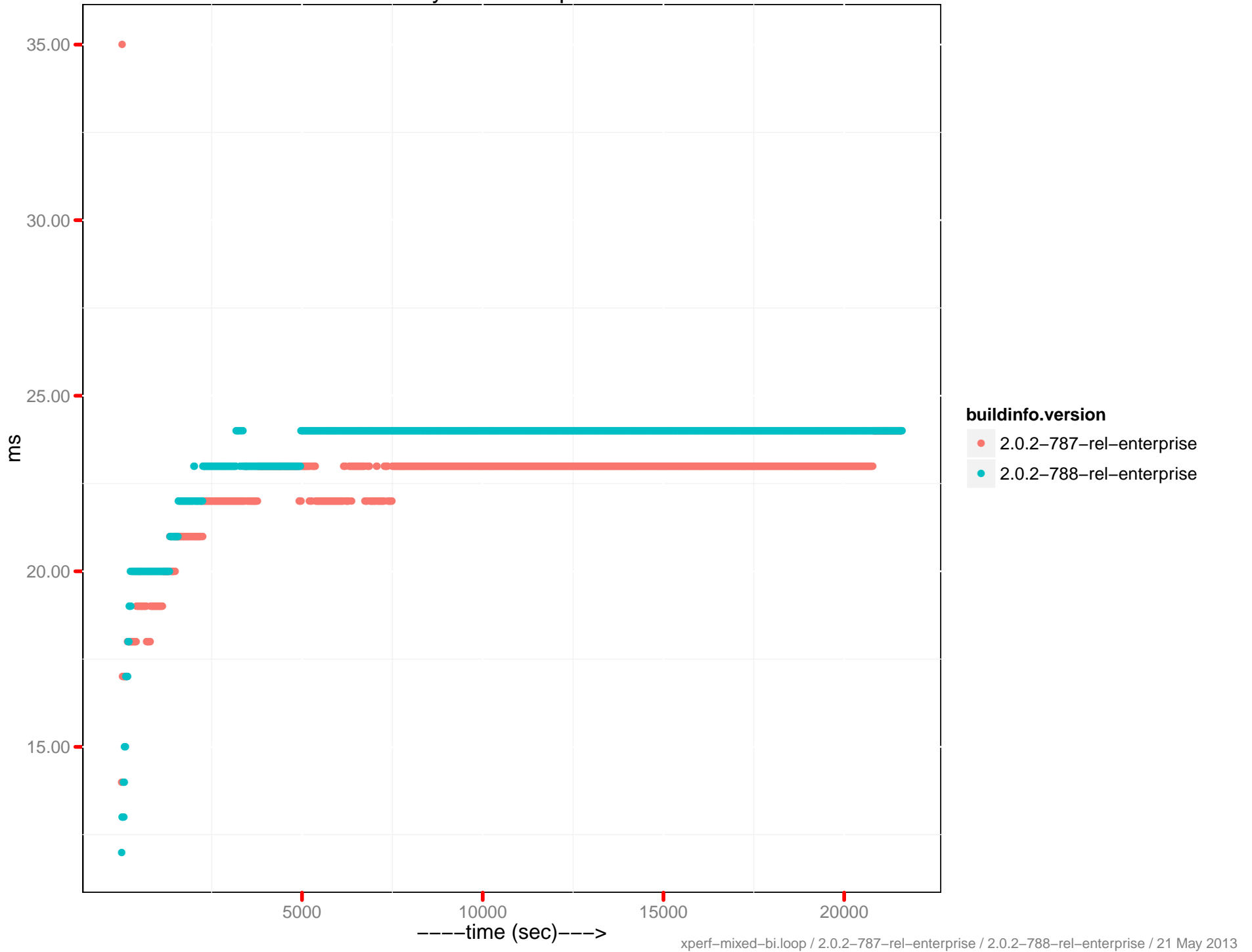




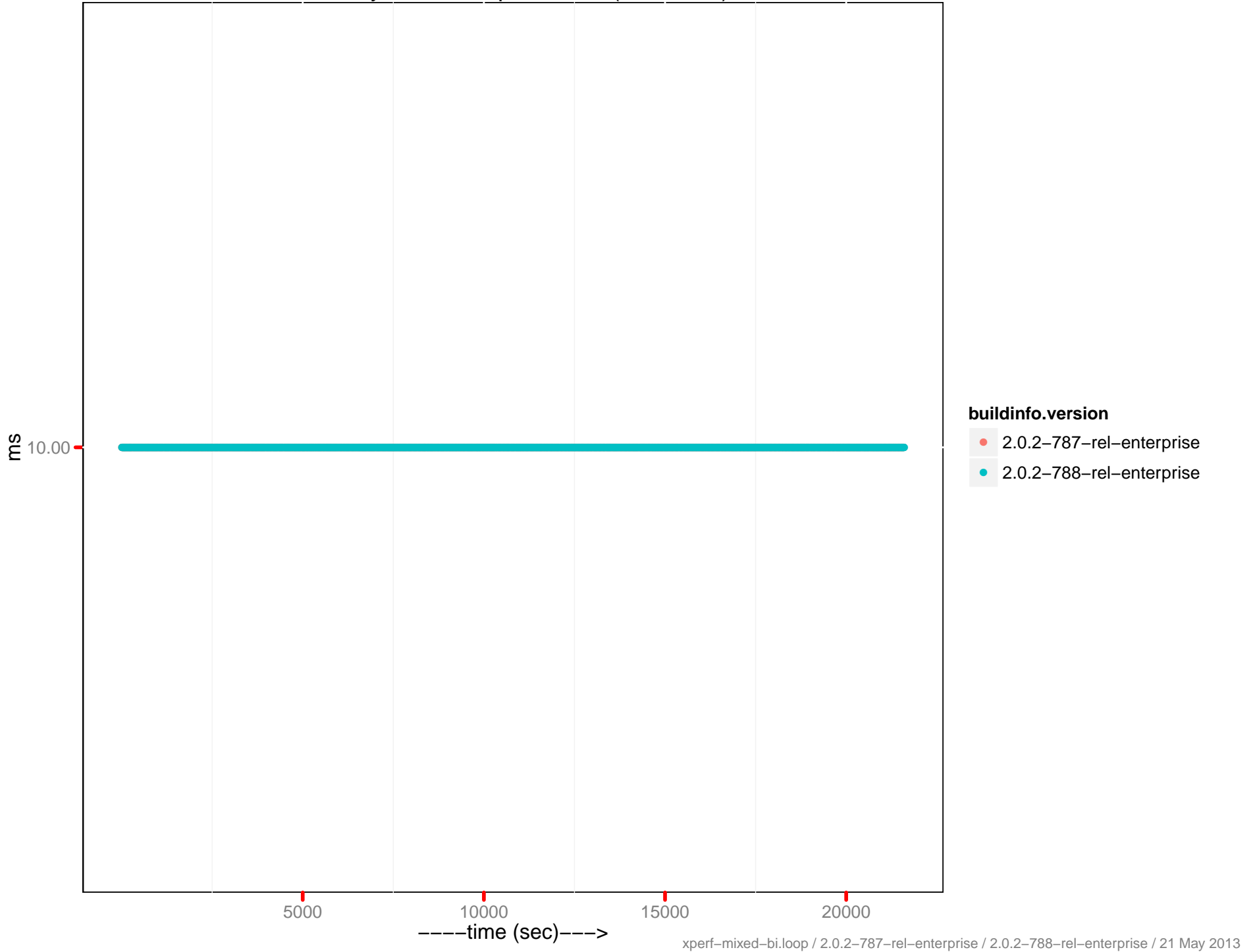
Latency-set 95th percentile (0 - 10ms)



Latency-set 99th percentile



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



# Query throughput

