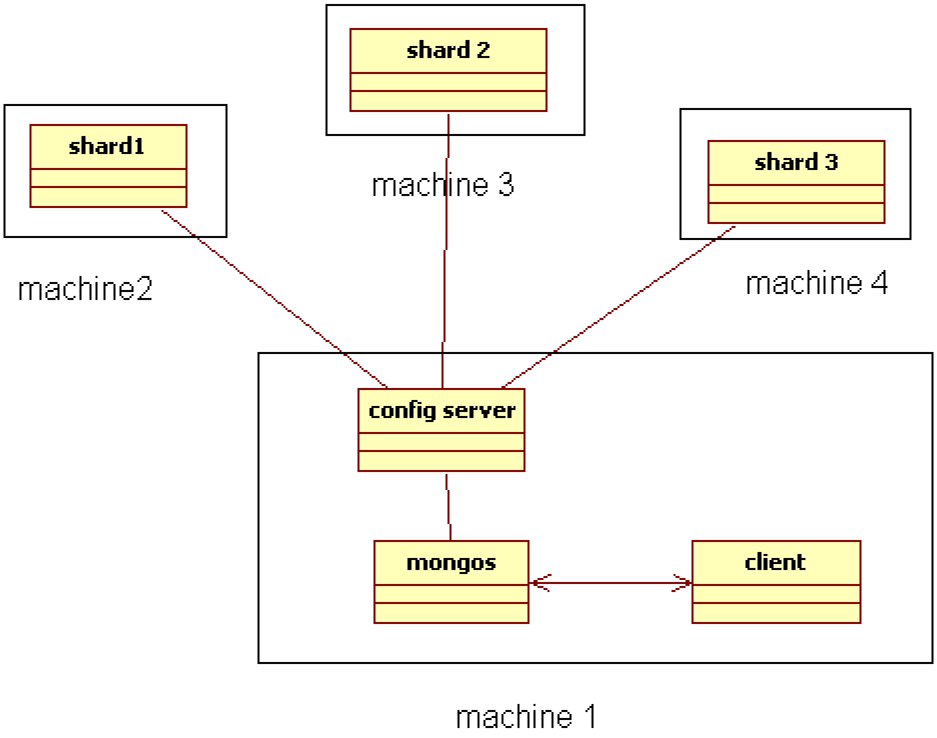
Deployment and Goals

A MongoDB cluster is deployed with 3 shards located on 3 machines (Windows Server 2003, 64 bit), for each 8G virtual memory is assigned. A diagram illustrates the cluster below. We focus on:
1) Compare performance between 3-shard cluster and single node mode, and figure out the influence of network transmission. As for single node mode, each shard is replaced by a process running on one single machine (for which 40G virtual memory is assigned);
2) Investigate performance of cluster and figure out the reasons of throughput degradation, for which a 4-shard cluster is deployed.



Comparisons between cluster and single node mode

Insert & Select under variant bulk size

We continuously insert documents into a shard-enabled collection for 10 minutes. Number of insert and select is recorded once a minute, and each insert/select affects one document containing 11 records.

For cluster mode: insert & select with bulk size = 1, 100 and 1000, and for single node mode: insert & select with bulk size = 100. Results will reflect the influence of network transmission.

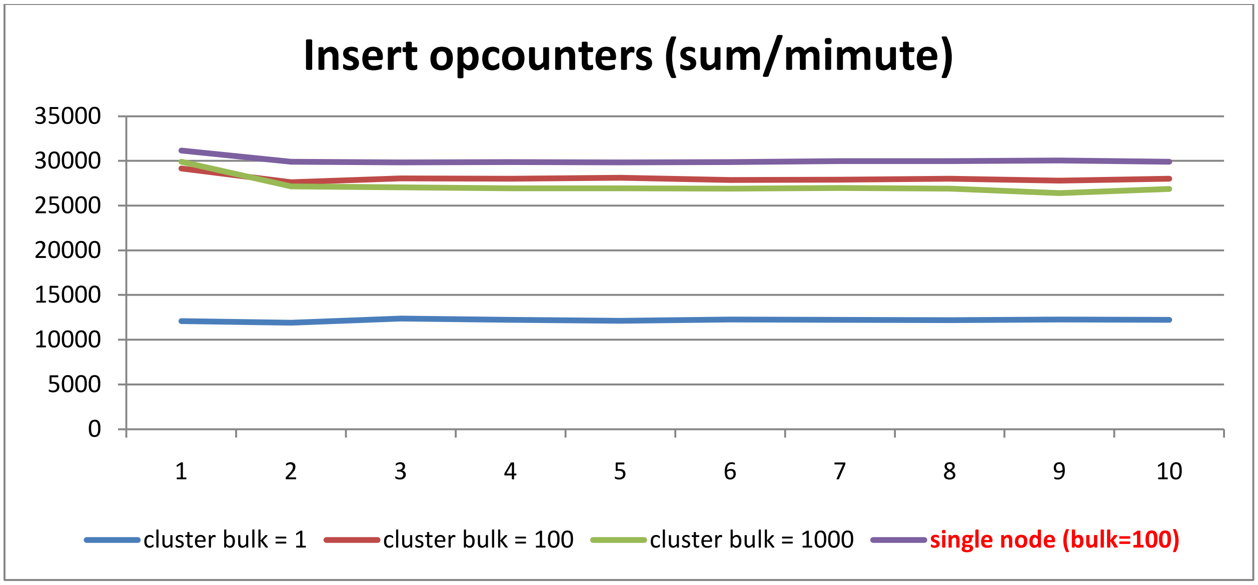


Fig 1(a)

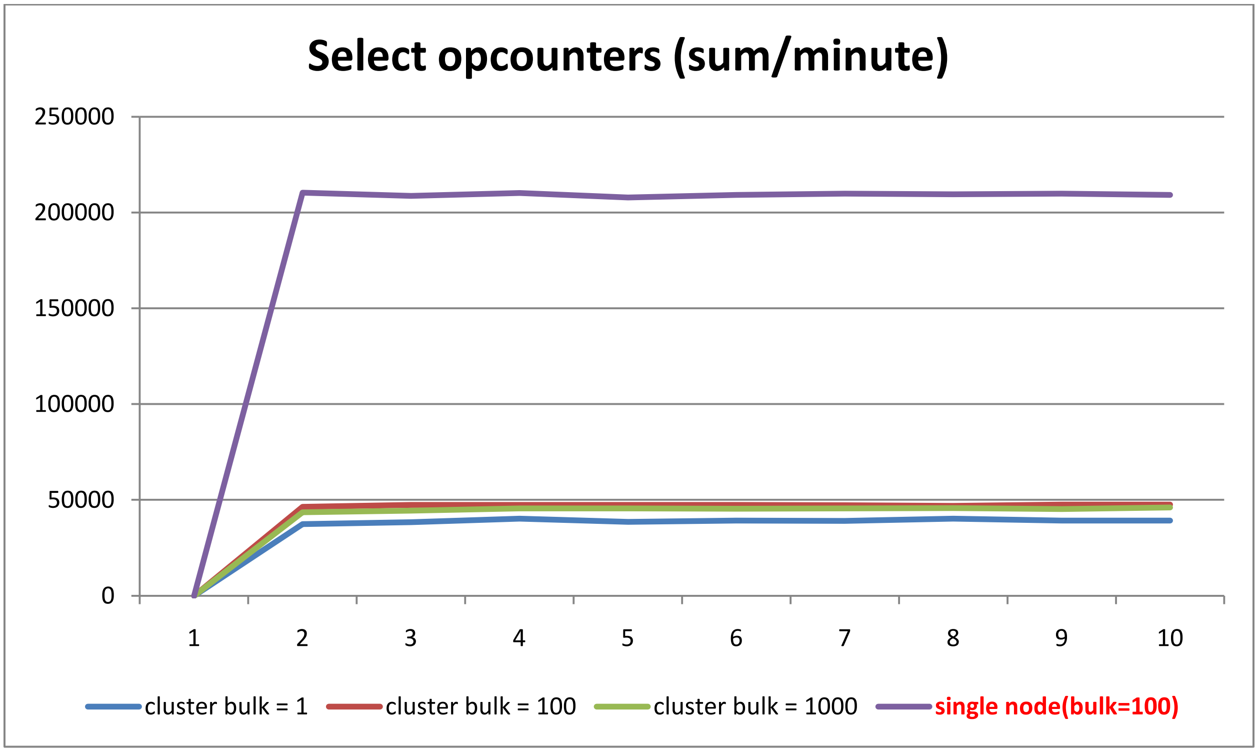


Fig 1(b)

In last report we did not find significant influence of bulk size for insert under single node mode, but we do under cluster mode this time.
From fig 1(a) we can see: 1) for cluster mode, throughput enhancement from bulk size=1 to 100 is significant; 2) for bulk size = 100, throughput decrease is acceptable for cluster mode.
From fig 1(b) we can see a large gap between cluster mode and single node mode for select operation.

Because select under single mode is so fast that even a small influence of network transmission will affect a lot.

Long time insert

We continuously insert data for 6 hours. From Fig2(a), throughput generally decreases during the first 3 hours, but became unstable for the second 3 hours. There might be more disk IO for memory swap and chunk movements. I will try to figure out the reasons in the next section. Anyway, cluster mode is more stable than single mode for the first 3 hours during which period about 6,000,000 documents are inserted.

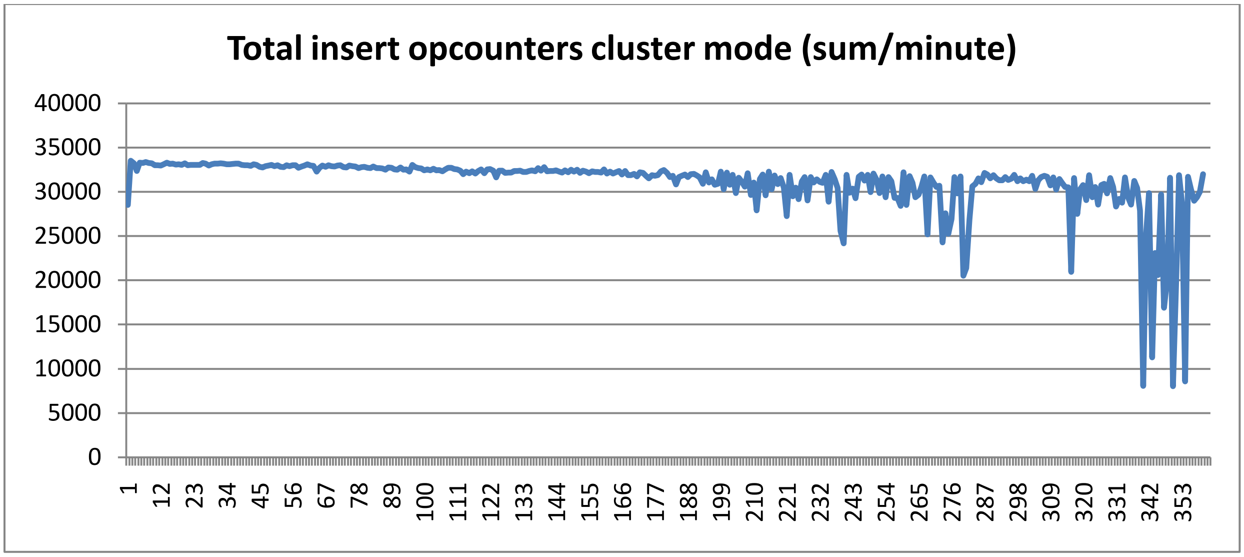


Fig 2(a)

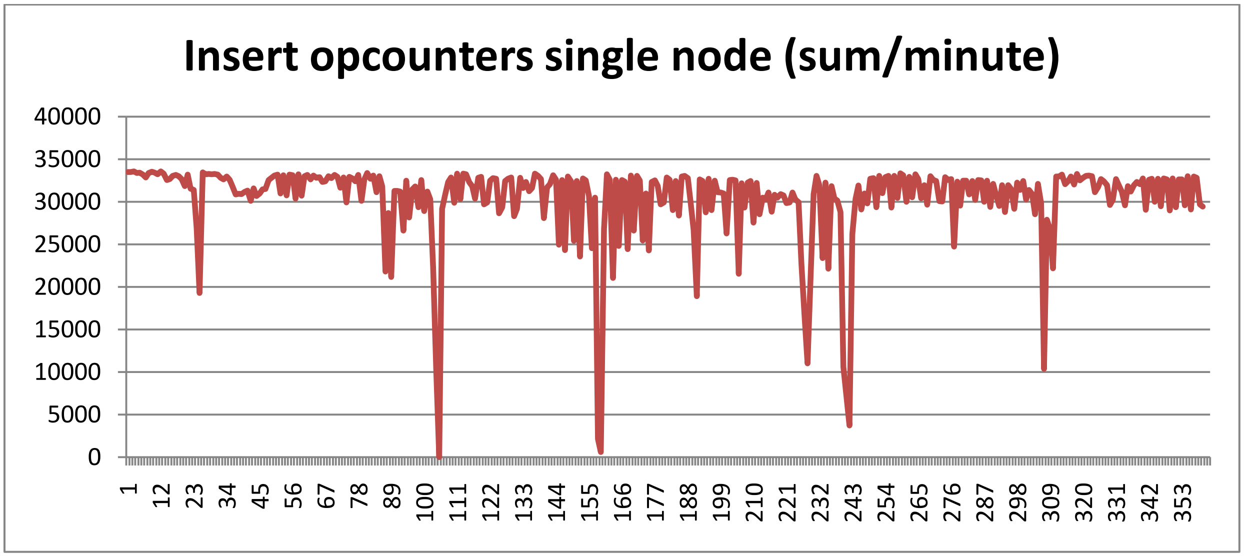


Fig 2(b)

Observe the cluster

Several tests are done to investigate cluster's performance. Here is a summary for a 4-shard cluster. We start three threads to insert/select/update documents on one single collection for six hours. Insert bulk is 100, and interval is 0.

We are going to figure out why the throughput decreases by observing RAM usage, global write lock etc.

Insert throughput

Total cluster insert throughput and that of each shard are illustrated as below. We can find throughput is stable until about the 287th minute, after which all documents are inserted to the machine CHN-LUJI (Fib3-b). We should try to figure out the reason by analyze the status of this shard.

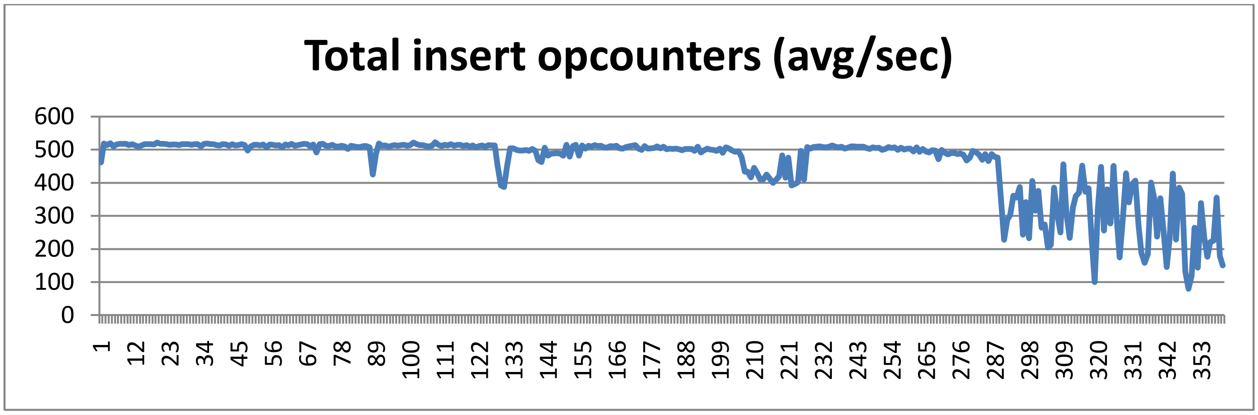


Fig3(a)

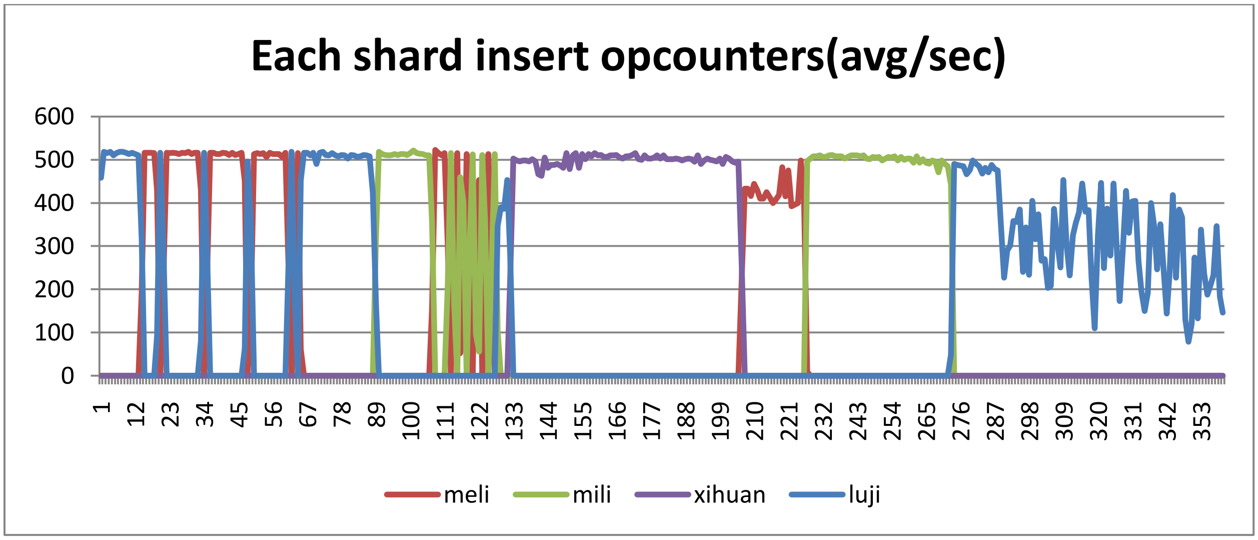


Fig3(b)

Records of RAM usage can be found from Fig 4. The shard CHN-LUJI uses memory up to about 8G which is the total RAM size. By reading some related documents, I get the knowledge: it is the working set size plus MongoDB's indexes which should ideally reside in RAM at all times i.e. the amount of available RAM should ideally be at least the working set size plus the size of indexes plus what the rest of the OS and other software running on the same machine needs. If the available RAM is less than that, LRUing is what happens and we might therefore get significant slowdown.

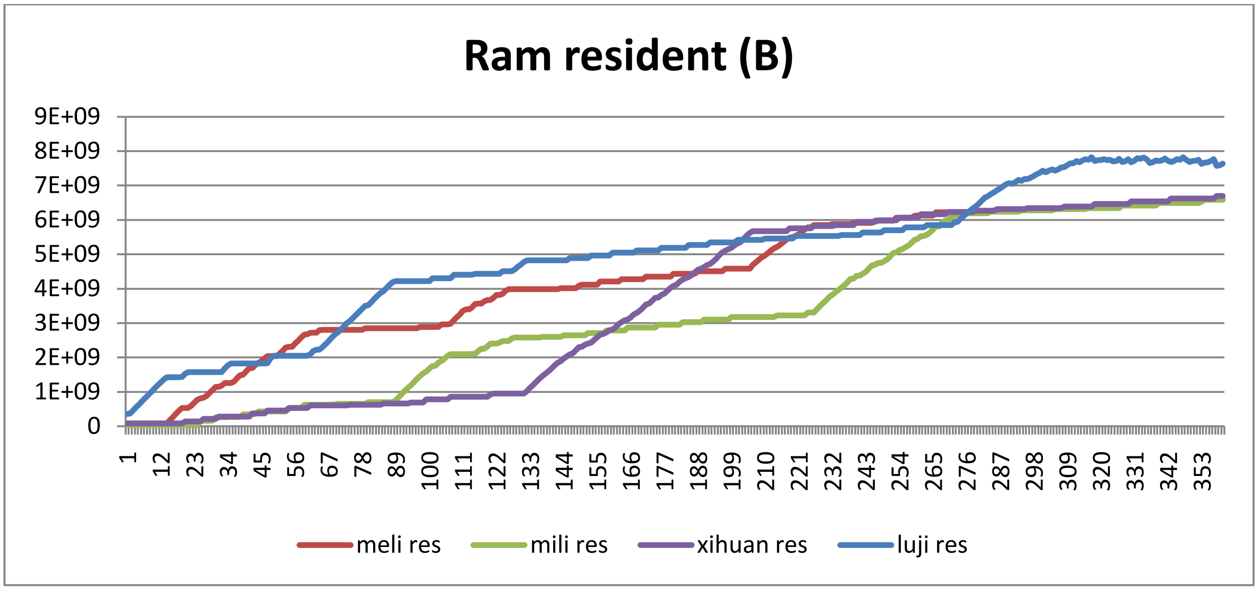


Fig 4

Get significant slowdown will cause heavier write lock percentage. Bad news is that, right now we have a global lock which applies across all databases on a Mongod instance, and a write lock acquisition is greedy i.e. a pending write lock acquisition will prevent further read lock acquisitions until fulfilled. So, the heavy write lock percentage will further cause all operations down which can be observed from Fig 5.

So, the performance decrease is caused by:
RAM limit -> slow down -> heavier global write lock -> whole slow down

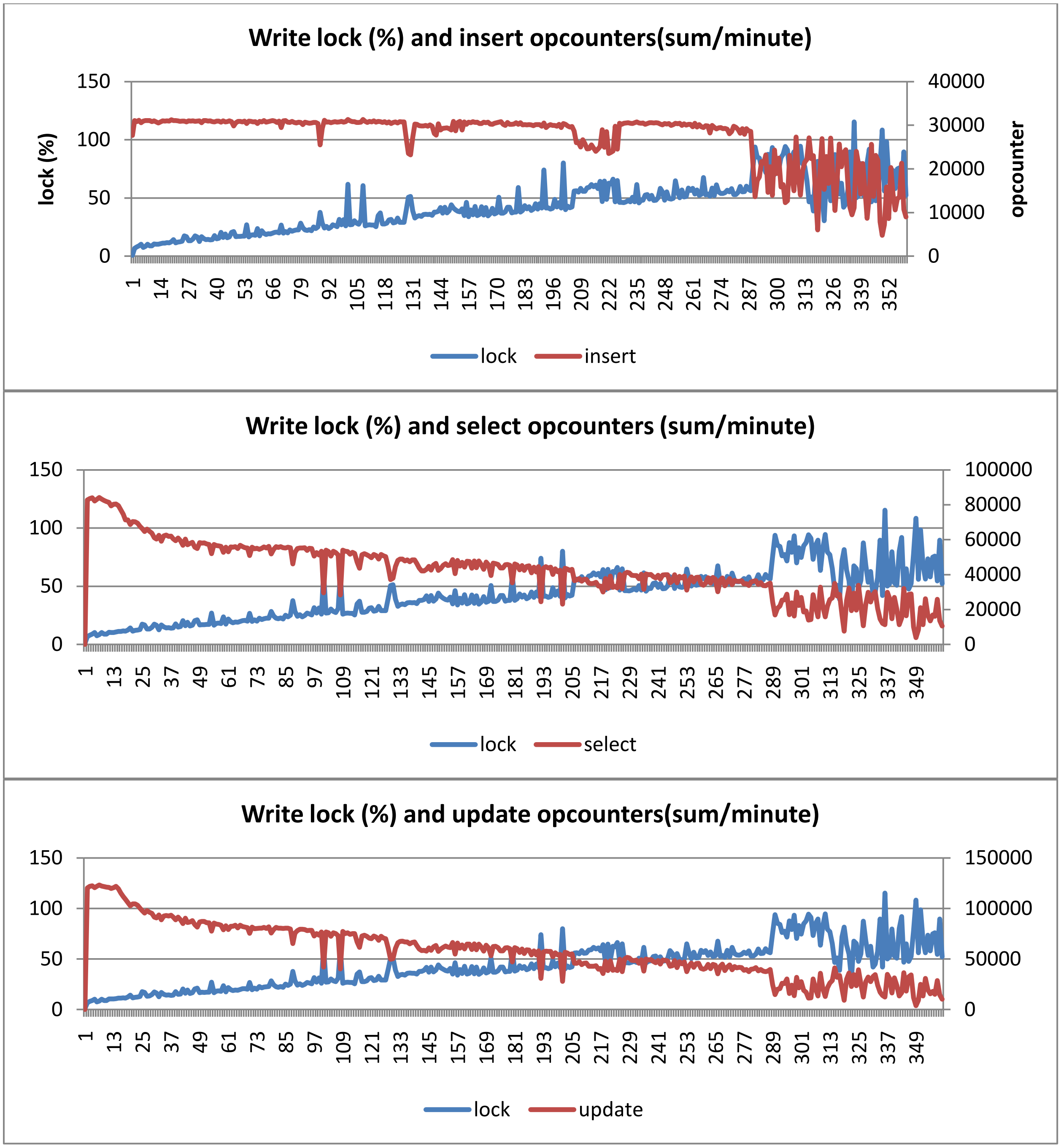


Fig 5

PS: Write clock percentage value is the summary value of the four shards, so values may exceed 100.

Conclusions

1. MongoDB gets many advantages from cluster mode for data insertion (Fig 2 and Fig 3), while network transmission causes select throughput decrease;

2. RAM size significantly impacts MongoDB’s performance, and we’d better get enough RAM to avoid degradation of performance;
3. More shards may enhance the whole performance (meanwhile bring more RAM);