



DB experiments

Informal Systems, CometBFT, Anton P.



Outline



Outline

- Goal
- Approach
- Compaction
- Virtual memory
- Sequential keys
- Bugs
- Conclusion



Goal



Issue

- Disk usage
- Different backends
- Time



Approach

A horizontal bar with a teal segment on the left and an orange segment on the right.

Approach

- 5 backends: *goleveldb*, *rocksdb*, *cleveldb*, *badgerdb*, *boltdb*

A horizontal bar with a teal segment on the left and an orange segment on the right.

Approach

- 5 backends: *goleveldb*, *rocksdb*, *cleveldb*, *badgerdb*, *boltdb*
- Artificial data*

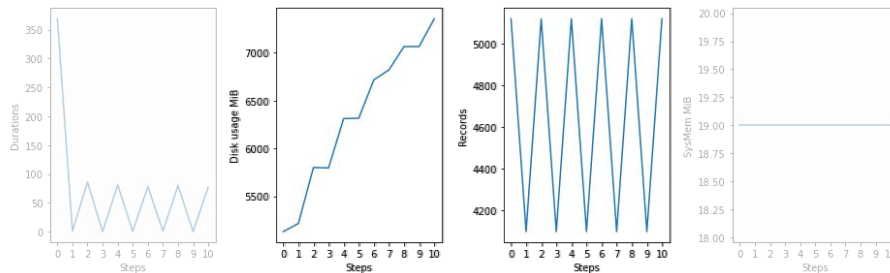
Compaction



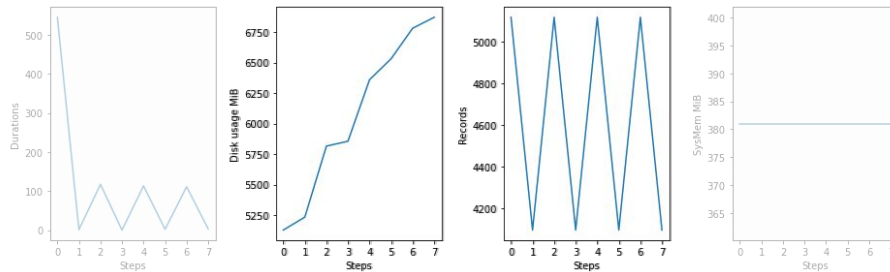
Compaction. Motivation



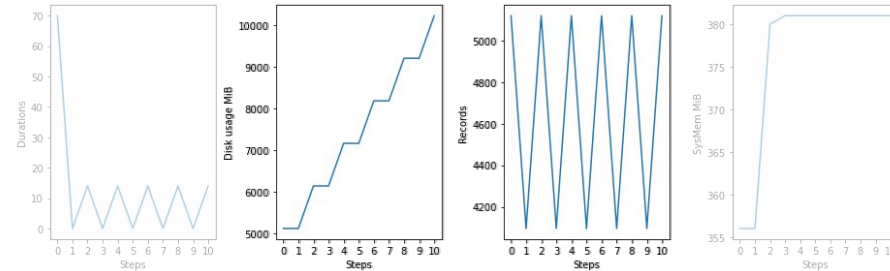
cleveldb



goleveldb

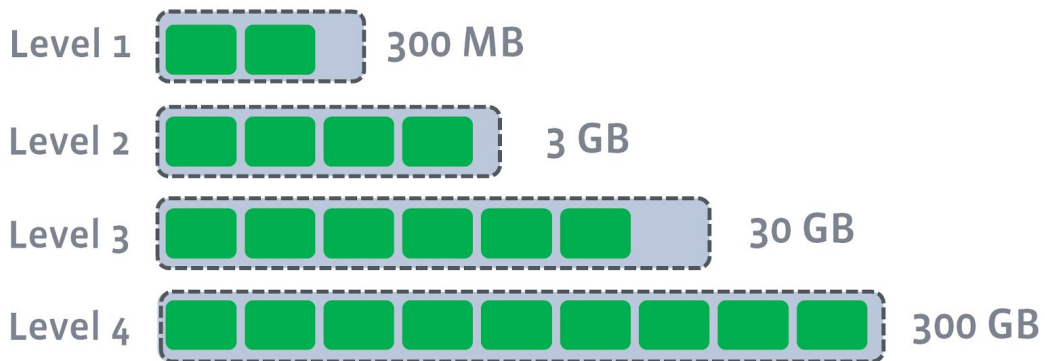


badgerdb





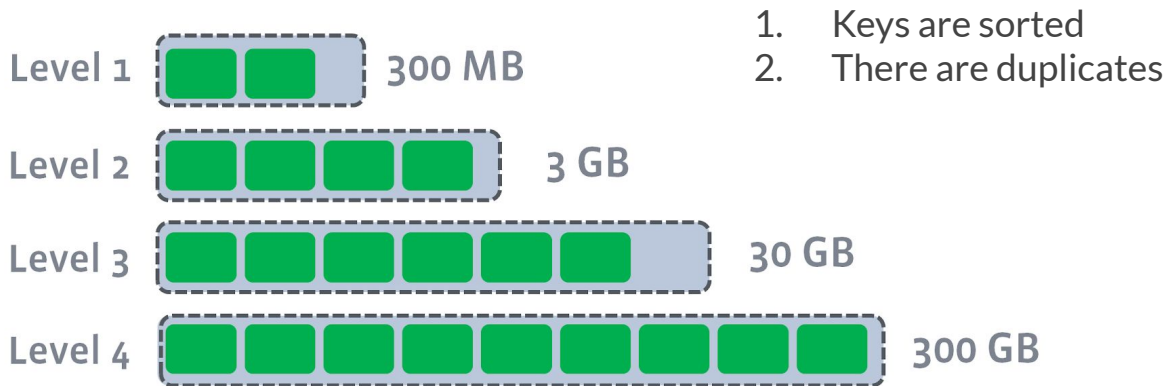
Compaction. Definition



Pictures from: <https://www.speedb.io/blog-posts/understanding-leveled-compaction>

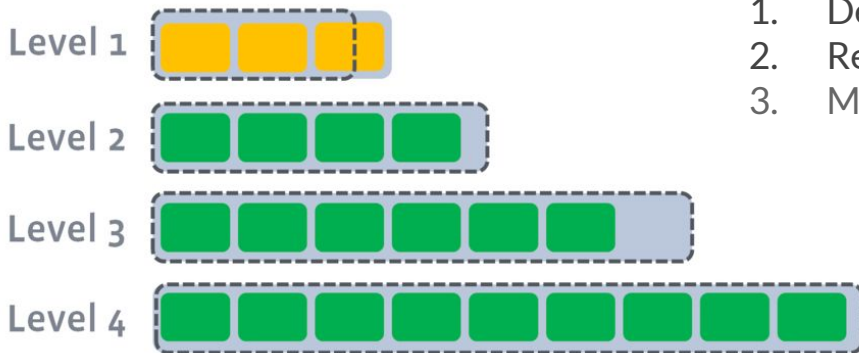


Compaction. Definition





Compaction. Definition



We compact to:

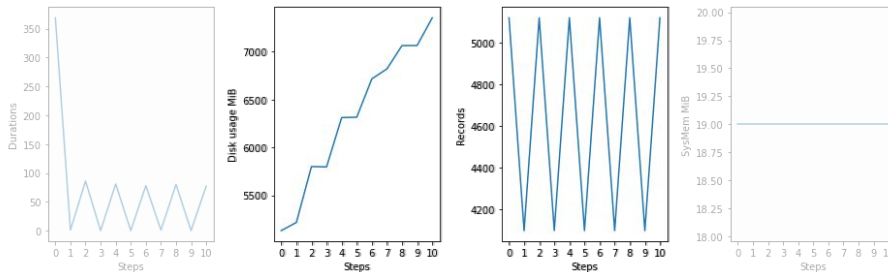
1. Delete obsolete values
2. Remove duplicates
3. Move data to lower levels



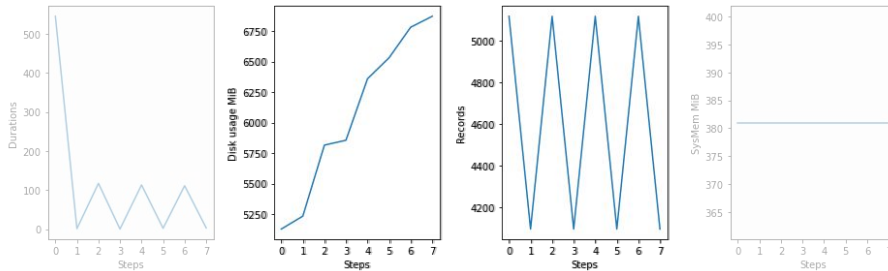
Compaction. Does it happen?



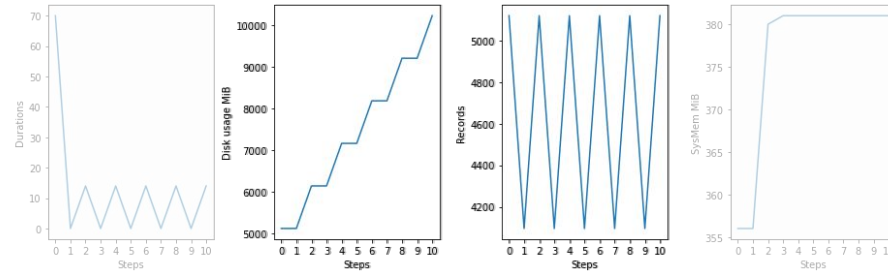
cleveldb



goleveldb



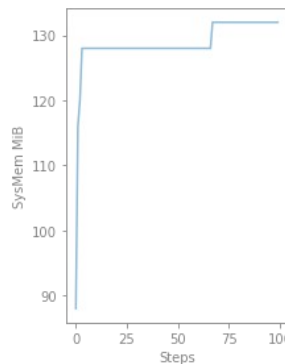
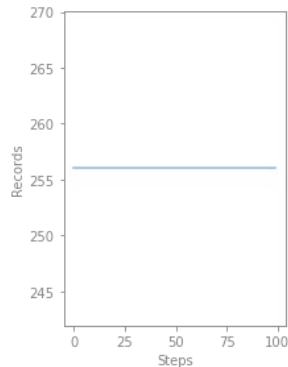
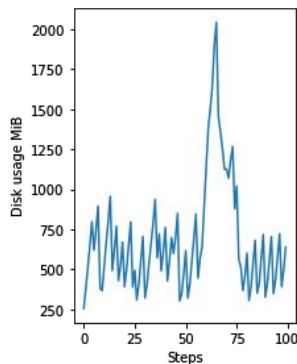
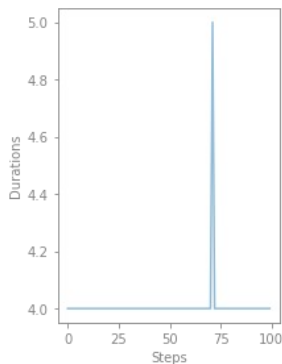
badgerdb



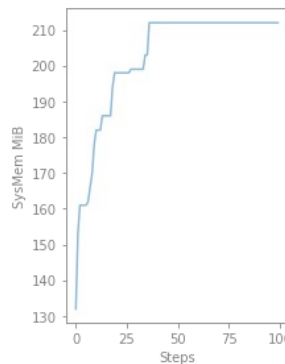
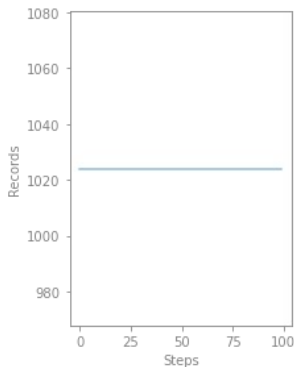
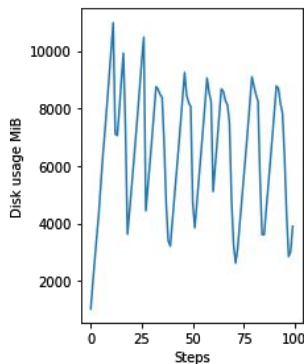
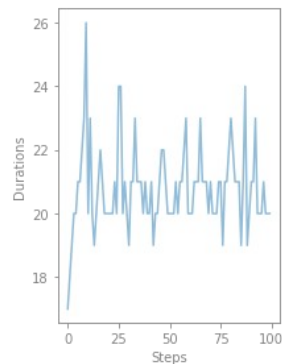


Compaction. Does it happen?

256MiB



1024MiB



GoLevelDB



Compaction. Can we trigger it?



Experiment step:

1.  Insert sequential keys

2.  Delete them

3.  *CompactRange*

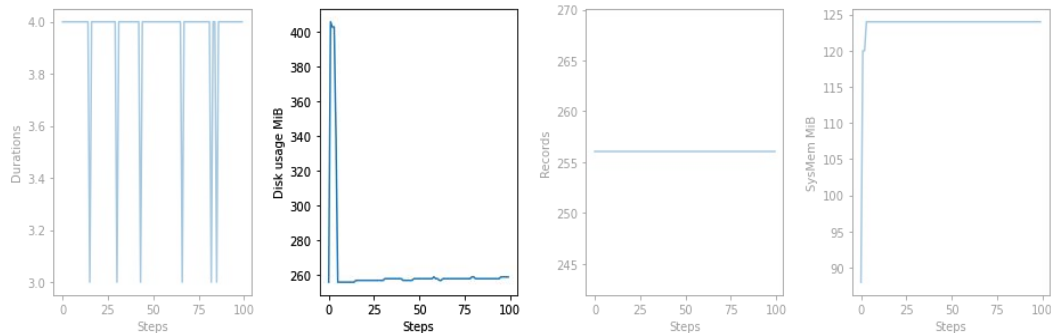
```
00000
00001
00010
00011
00100
...
```



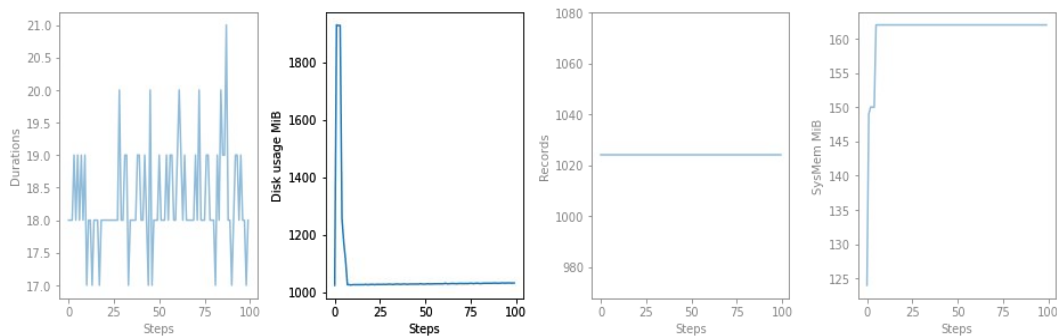
Compaction. Can we trigger it?



256MiB



1024MiB



GoLevelDB

A horizontal bar with a teal segment on the left and an orange segment on the right.

Compaction. Takeaways

- Is there to remove redundant key-value pairs → reduce disk usage
- If it's not triggered manually, the disk usage might grow quite large
- For *GoLevelDB* we can use *compactRange* if sequential keys are deleted

A horizontal bar with a teal segment on the left and an orange segment on the right.

Compaction. Further steps

- Perform analogues experiments for other backends
- Consider other methods such as *forceCompact*
- Investigate compaction parameters

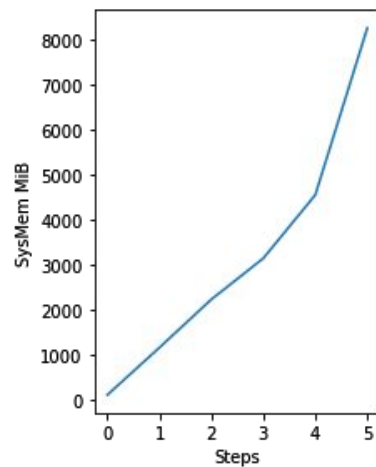
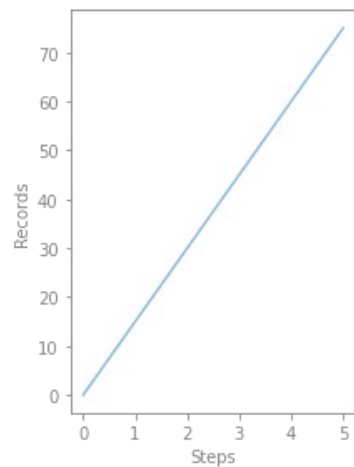
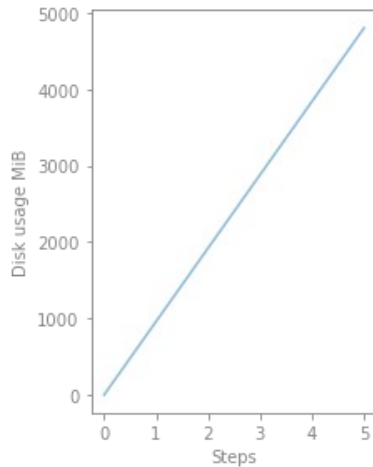
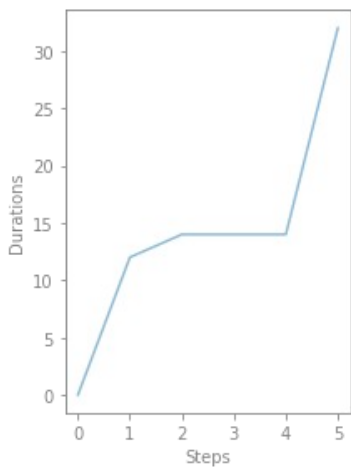


Virtual memory



BadgerDB VM overuse

badgerdb





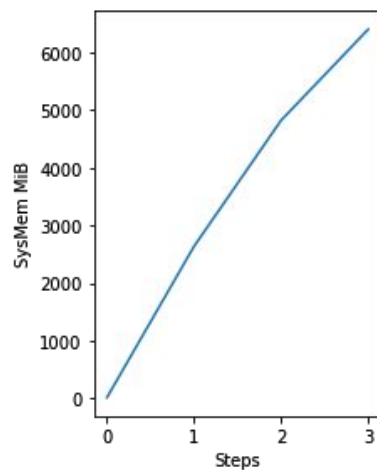
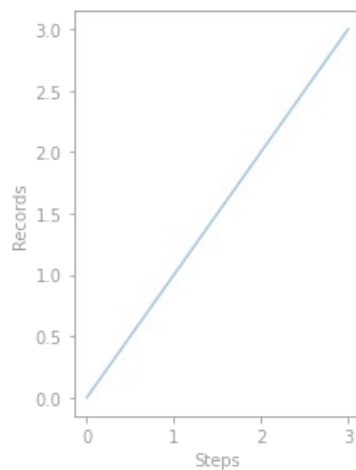
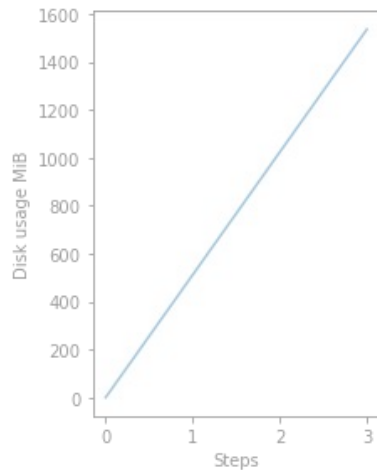
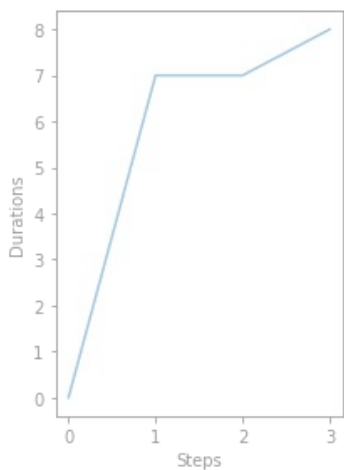
BadgerDB VM overuse. Reason

```
iterCount := 0
for ; iter.Valid(); iter.Next() {
    iterCount++
}
```



GoLevelDB VM Overuse

goleveldb



A horizontal bar with a teal segment on the left and an orange segment on the right.

GoLevelDB VM overuse. Reason

Memory profiler for **goleveldb** outputs the following space allocation percentage:

- 40%: [Put](#)
- 37%: [mCompaction](#)



Sequential keys



Sequential keys inserts

CLevelDB

Total time to insert 5120 sequential keys: 4.148798078s

Total time to insert 5120 random keys: 5m1.129714419s

GoLevelDB

Total time to insert 5120 sequential keys: 16.023511876s

Total time to insert 5120 random keys: 8m12.777516168s



Bugs



BoltDB is raw

- Int32 overflow
- Process hangs forever

Reproduced in:

https://github.com/cometbft/cometbft/blob/anton/db_measurements/tools/goleveldb_perf/misbehavior_test.go



Conclusion



Conclusion

- Disk usage by different backends

A horizontal bar with a teal segment on the left and an orange segment on the right.

Conclusion

- Disk usage by different backends
- Compaction
 - Might happen rarely
 - Can be triggered
 - Study backends other than *GoLevelDB*
 - Parameters

A horizontal bar with a teal segment on the left and an orange segment on the right.

Conclusion

- Disk usage by different backends
- Compaction
 - Might happen rarely
 - Can be triggered
 - Study backends other than *GoLevelDB*
 - Parameters
- Virtual memory

A horizontal bar with a teal segment on the left and an orange segment on the right.

Conclusion

- Disk usage by different backends
- Compaction
 - Might happen rarely
 - Can be triggered
 - Study backends other than *GoLevelDB*
 - Parameters
- Virtual memory
- Sequential keys

A horizontal bar with a teal segment on the left and an orange segment on the right.

Conclusion

- Disk usage by different backends
- Compaction
 - Might happen rarely
 - Can be triggered
 - Study backends other than *GoLevelDB*
 - Parameters
- Virtual memory
- Sequential keys
- BoltDB is raw



Conclusion

- Disk usage by different backends
- Compaction
 - Might happen rarely
 - Can be triggered
 - Study backends other than *GoLevelDB*
 - Parameters
- Virtual memory
- Sequential keys
- BoltDB is raw

Thank you for your
attention!

Plots and analysis:

https://github.com/cometbft/cometbft/blob/anton/db_measurements/tools/goleveldb_perf/Results.ipynb

https://github.com/cometbft/cometbft/blob/anton/db_measurements/tools/goleveldb_perf/GoLevelDBCompaction.ipynb