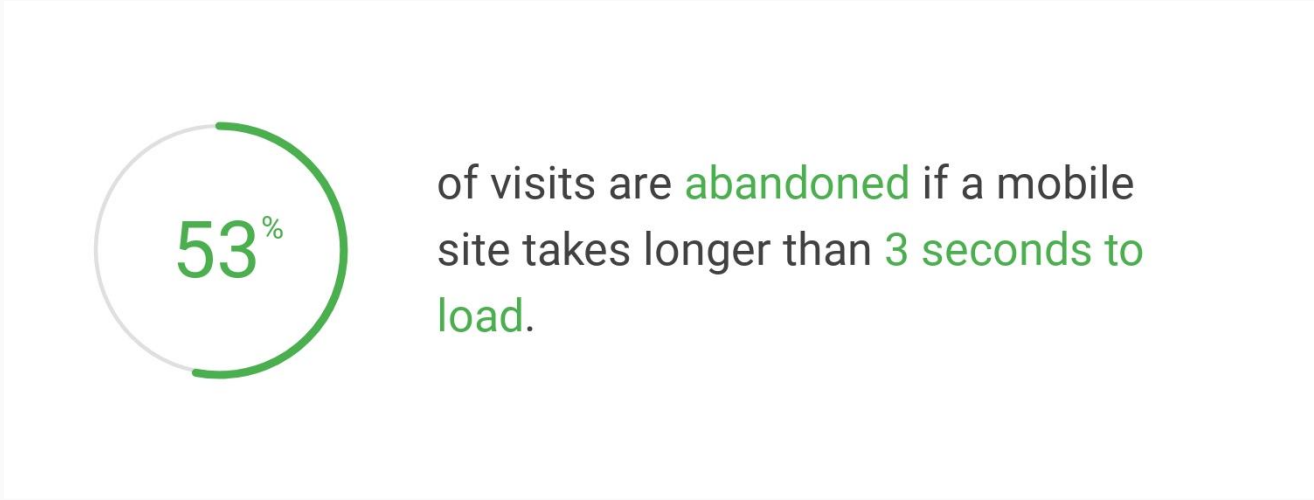
Using MongoDB in Cloud Based Commercial Systems

Ethan Uphoff
Division of Science and Mathematics
University of Minnesota, Morris
Morris, Minnesota, USA

MongoDB use in Commercial Systems



Think with Google

Google Data, Global, n=3,700 aggregated, anonymized Google Analytics data from a sample of mWeb sites opted into sharing benchmark data, March 2016.

MongoDB use in Commercial Systems

Overview

1. Background
    a. MongoDB Overview
    b. Replicas Sets
    c. Consistency
    d. Write Concern
    e. MongoDB Drivers
2. MongoDB Driver Selection
3. Maintaining Logical Clocks with Causal Consistency
4. Conclusion

MongoDB

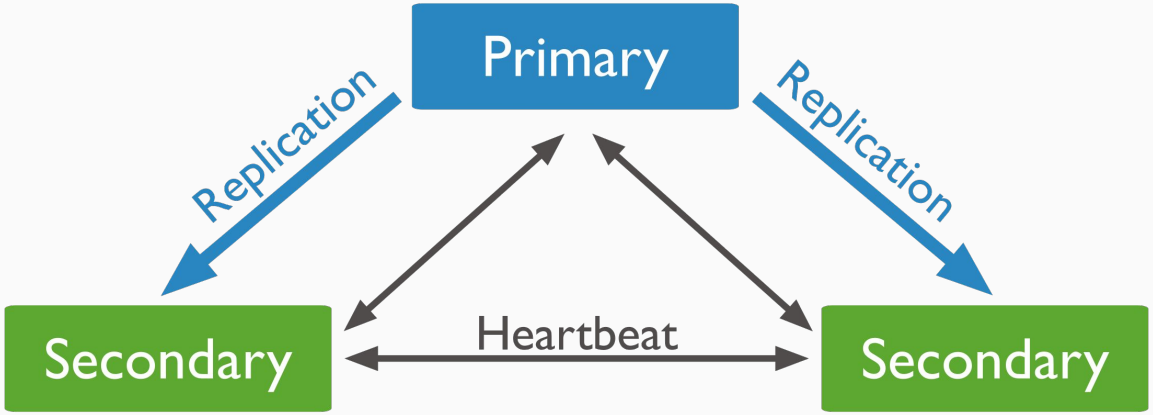
	GPA	Graduation_ Year
Student 1	4.0	2020
Student 2	3.1	2022

```
[
  {
    Name: “Student 1”,
    GPA: 4
  },
  {
    Name: “Student 2”,
    GPA: 3.1,
    Graduation_Year: 2022
  }
]
```

MongoDB

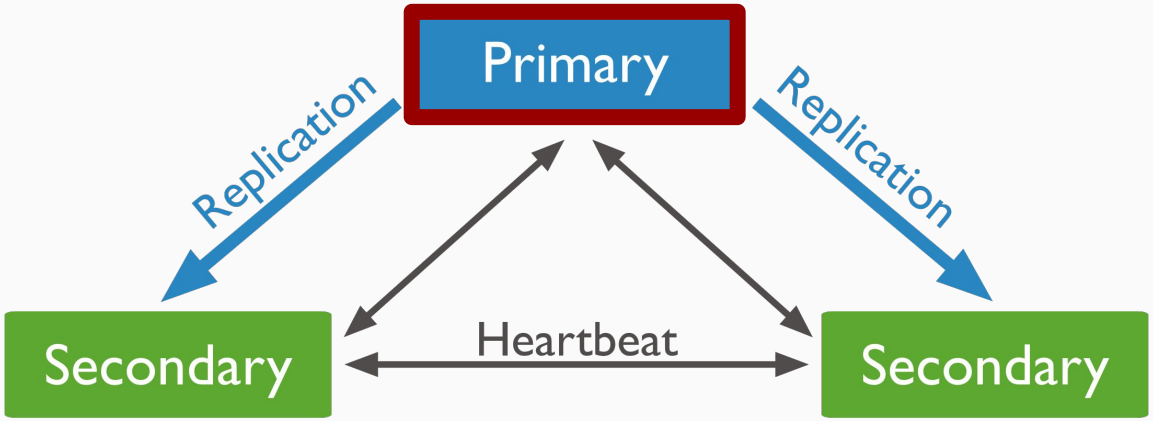
- Document store database, stores documents in JSON format [9]
- MongoDB eliminates many rules around data consistency
- This leads MongoDB to be a rather simple database in terms of development

Replica Sets



MongoDB manages replicas using primary nodes and secondary nodes [4]

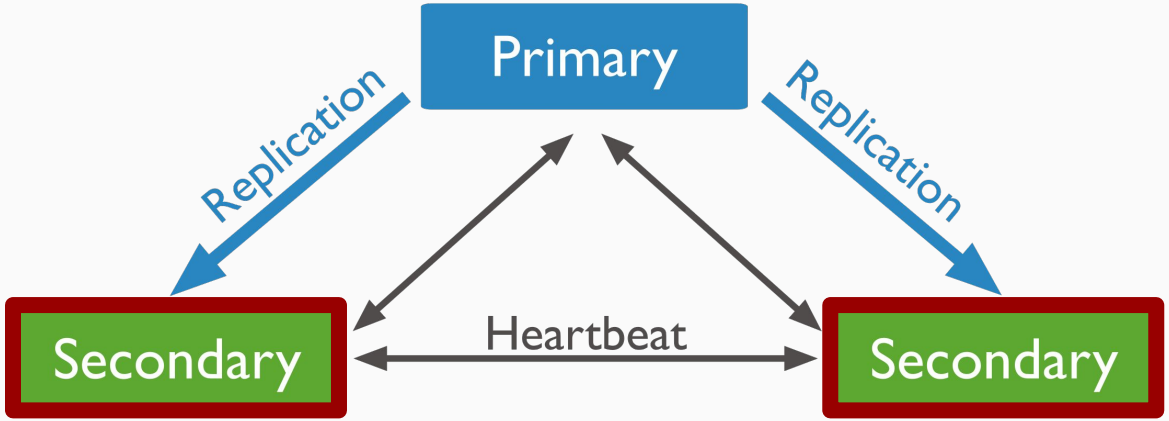
Replica Sets



The primary node acts as the source of truth

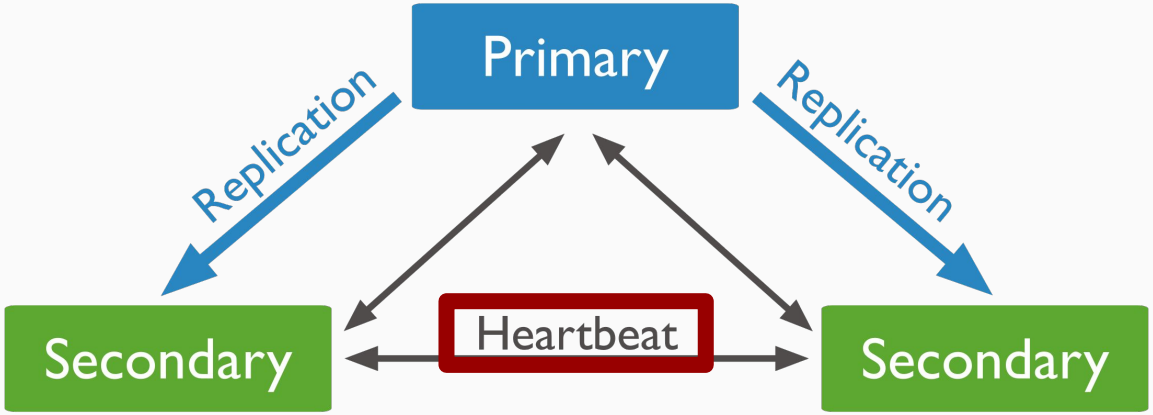
All writes go to the primary node [4]

Replica Sets



The secondary nodes contain a replica of the primary [4]

Replica Sets

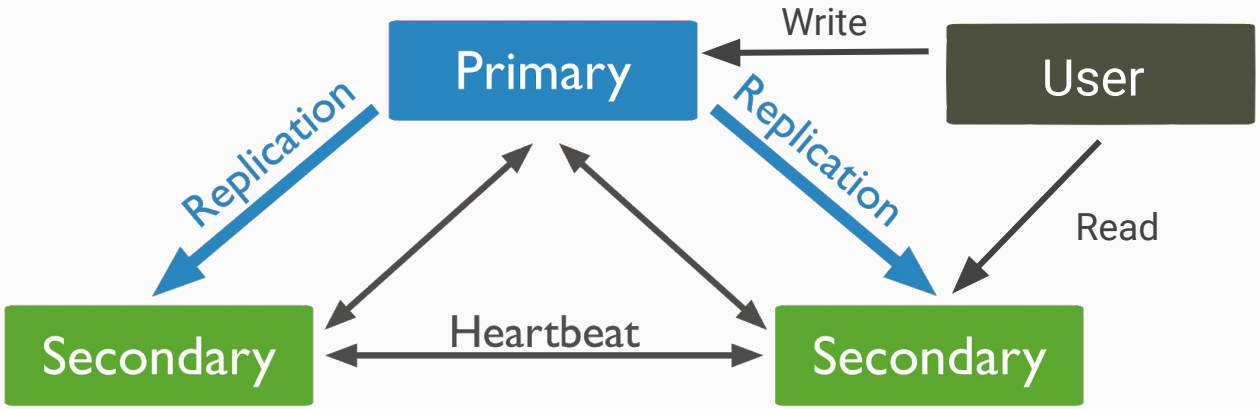


Information is passed between nodes using a heartbeat [4]

Consistency

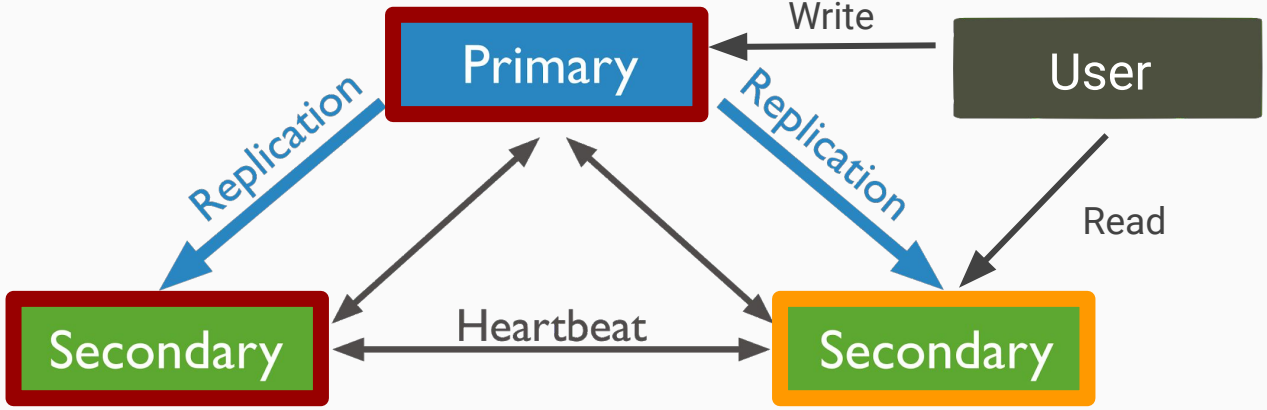
- Consistency is required in most multi-node systems
- Without consistency, users reading replicas will not have up to date data
- Consistency models allow MongoDB to replicate data to secondary nodes
- MongoDB uses eventual consistency as a default

Eventual Consistency



Data will be replicated as soon as possible to secondary nodes [8]

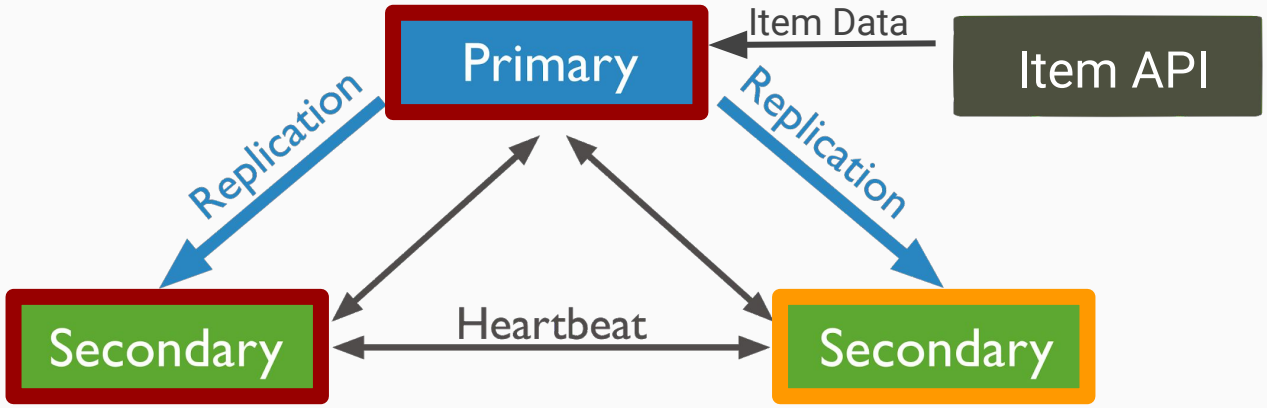
Eventual Consistency



The users data **is not** on the node they are reading from.

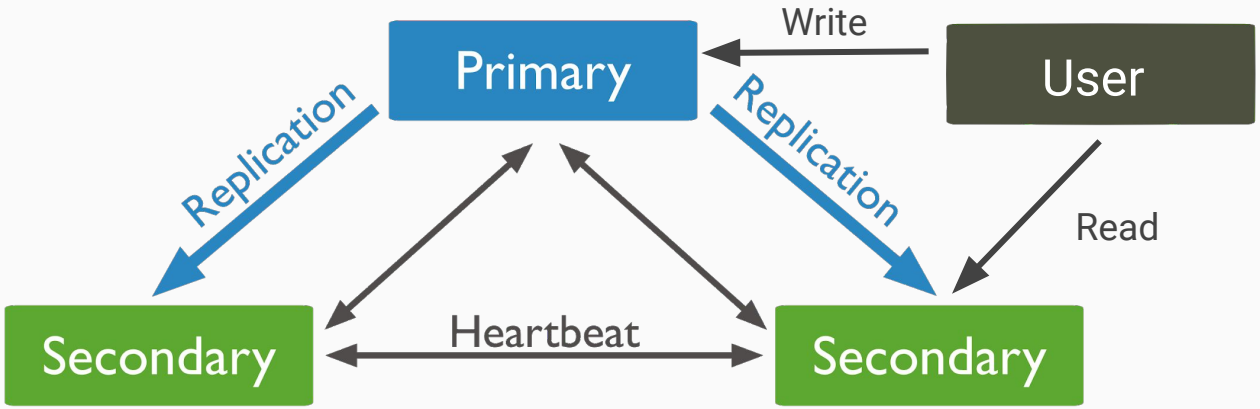
This means they cannot read the data they just wrote.

Eventual Consistency Use Case



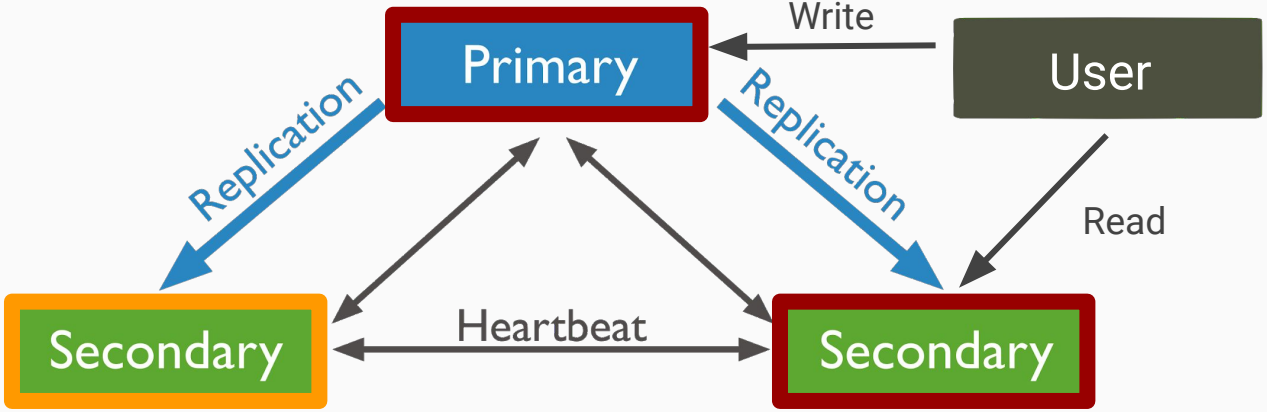
Systems which manage item information may not have to worry about consistency as it does not directly affect users as long as it eventually goes to the secondar nodes

Causal Consistency



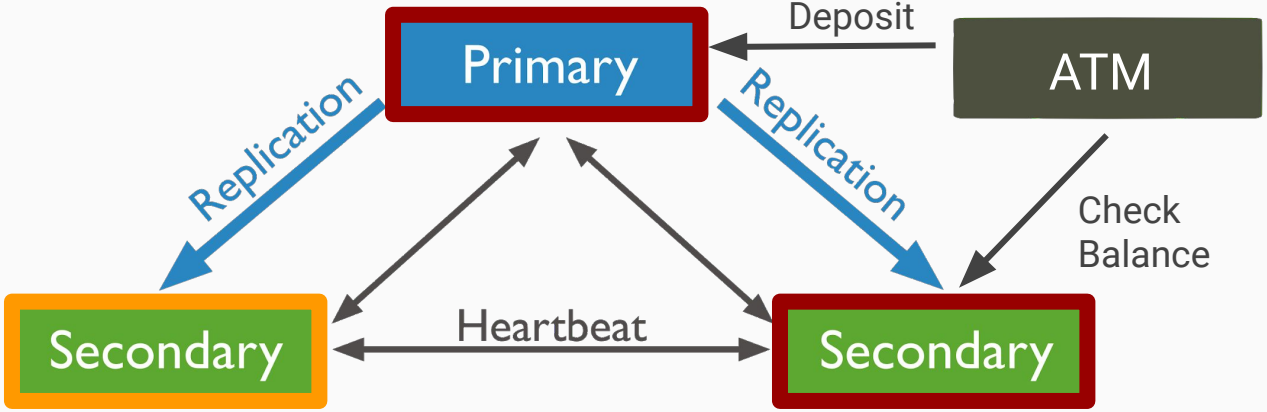
Data will be immediately replicated to the node the user is reading from [7]

Causal Consistency



The users data **is** on the node they are reading from

Causal Consistency Use Case



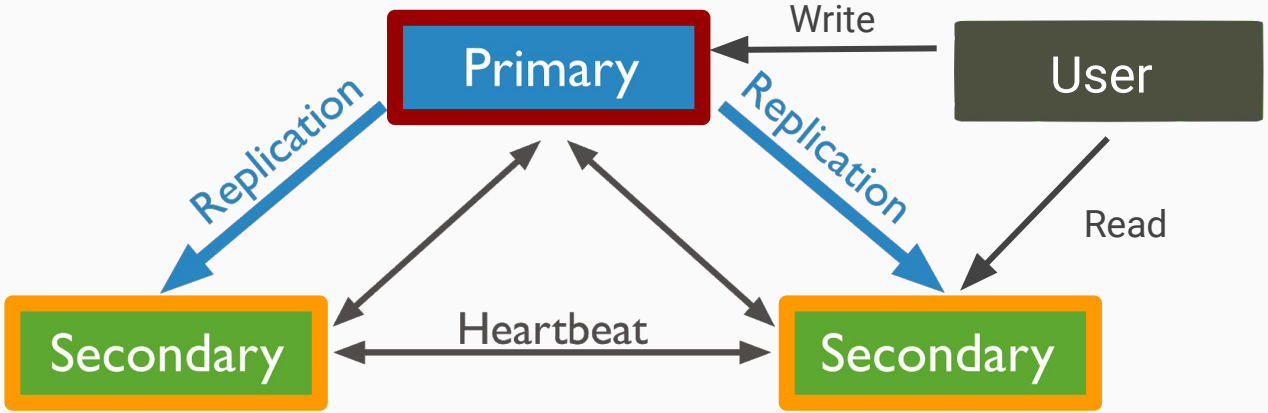
ATMs require data consistency at all times

Causal Consistency allows users to deposit money and check their balance immediately [6]

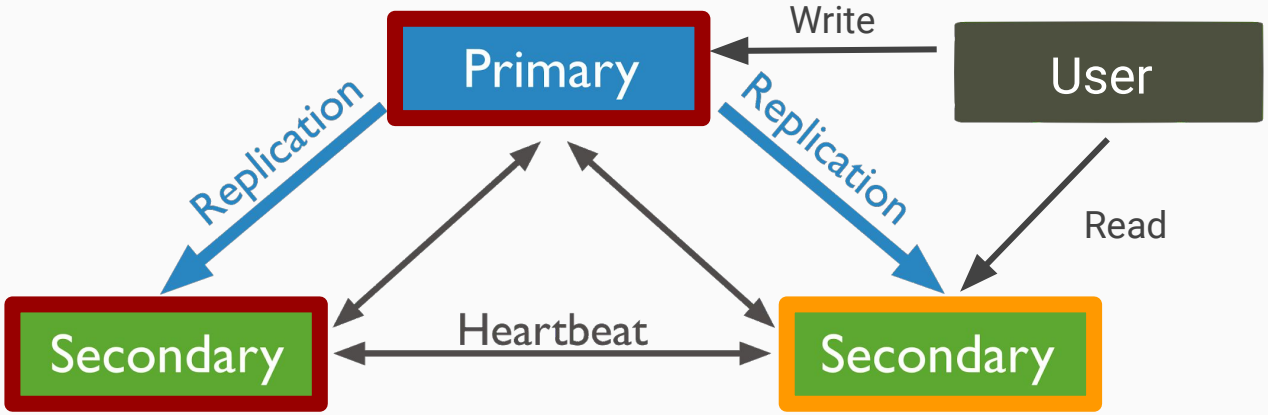
Write Concern

- Write concern is how MongoDB manages write durability
- Durability is determined by how difficult it is to completely delete the data
- Specifies number of nodes being written to with w:<number>
- A write concern of w:1 would be a non-durable write
- A write concern of w:majority is considered durable [5]
- Cannot replace causal consistency as majority isn't all nodes, only most

Write Concern w:1



Write Concern w:majority



w:majority needs to be spread to 50% or more nodes to be considered complete. [5]

MongoDB Drivers

- MongoDB has a multitude of drivers which developers can use
- These drivers choose the replica to use when reading data
- Drivers have similar functionalities but are not the same overall
- The drivers can directly affect latencies

Overview

1. Background
2. MongoDB Driver Selection
    a. Latency Evaluation
    b. GeoPerf
    c. Evaluation of MongoDB Drivers
3. Maintaining Logical Clocks with Causal Consistency
4. Conclusion

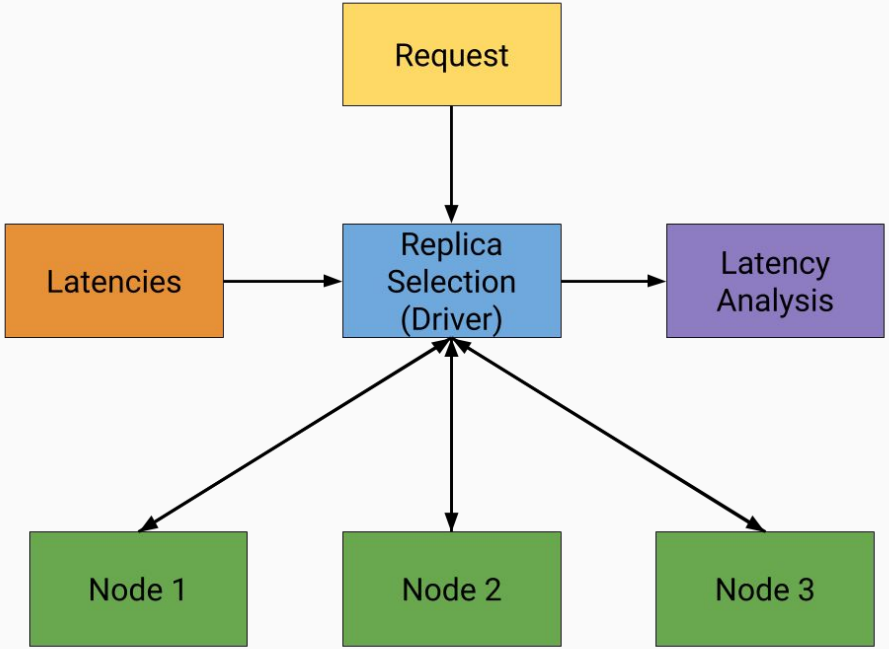
Latency Evaluation

- Bogdanov et al. looked at how the C++ and Java drivers in MongoDB influenced latency
- Gathered latency data from Amazon data centers around the world
- Using this data, they were able to emulate latencies in testing

GeoPerf

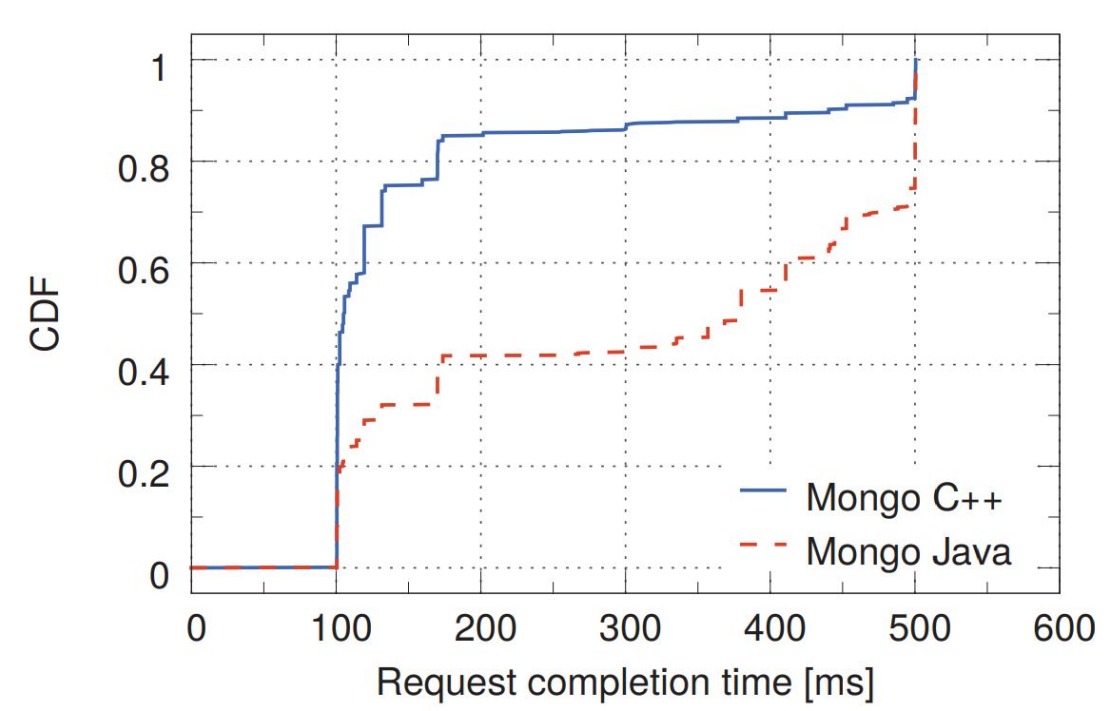
- GeoPerf is a tool which was created by Bogdanov et al. to analyse latencies
- It runs MongoDB in an environment to simulate the latencies Bogdanov et al. gathered
- By doing this they were able to analyse how the drivers responded to latency changes [1]

GeoPerf



Adapted from [1]

Evaluation of MongoDB Drivers



[1]

Evaluation of MongoDB Drivers

- The C++ driver was significantly better than Java in a cloud environment.
- Java could not quickly adapt to changes in the datacenters
- It’s difficulty adapting meant it picked worse replicas
- This led C++ to be the better driver overall for use in the cloud if deploying globally [1]
- Since this paper was published the drivers have been updated so these results could be different now

Overview

1. Background
2. MongoDB Driver Selection
3. Maintaining Logical Clocks with Causal Consistency
    a. Logical Clocks
    b. Dependencies
    c. Synchronization
    d. Causal Consistencies Effect on Throughput
4. Conclusion

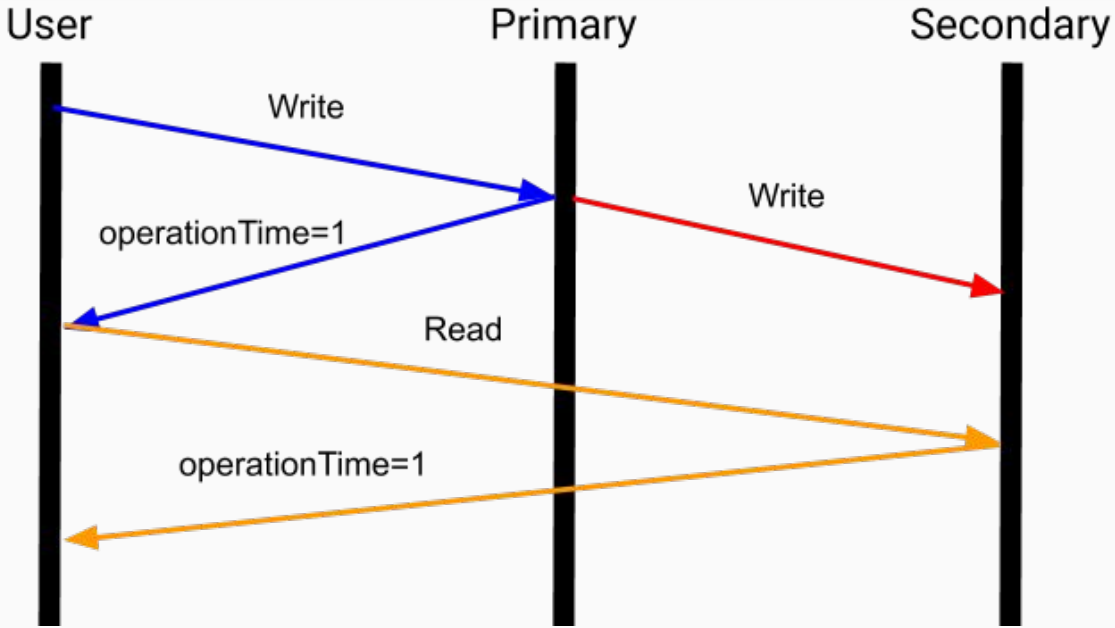
Logical Clocks

- Tyulenev et al. wanted to test the effect on throughput of causal consistency
- To test this they decided to use a logical clock
- Logical clocks are a clock which instead of incrementing or decrementing every second, increment after certain operations [7]

Logical Clocks

- In this case, Tyulenev et al. used an logical clock which increments specifically when a write occurs.
- They call the clock they used a “Hybrid Logical Clock”
- This clock increments like a normal clock but also tracks the real time
- Real time allows developers to rollback databases as needed [7]

Why Logical Clocks?



Logical clocks track requests so it can be used to make sure data being read is up to date

It is also lightweight as it is only tracking a single value

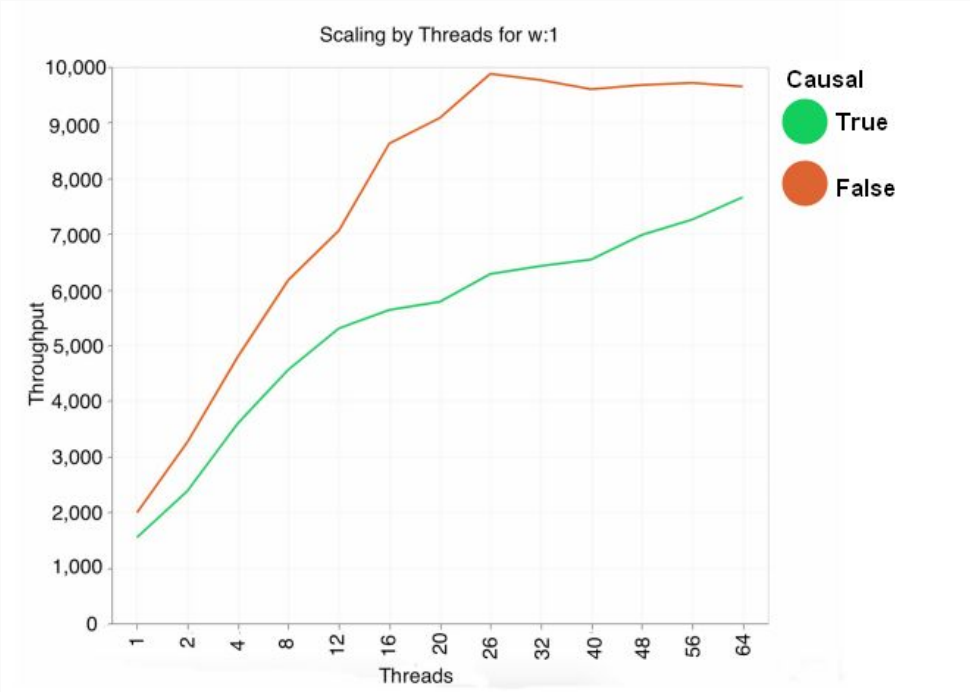
Dependencies

- Dependency on causal consistency was tracked using operationTime
- This operationTime is an incremented value in order to show the ordering in which operations took place
- This allowed Tyulenev et al. to make sure causal consistency was in effect [7]

Synchronizing Clocks

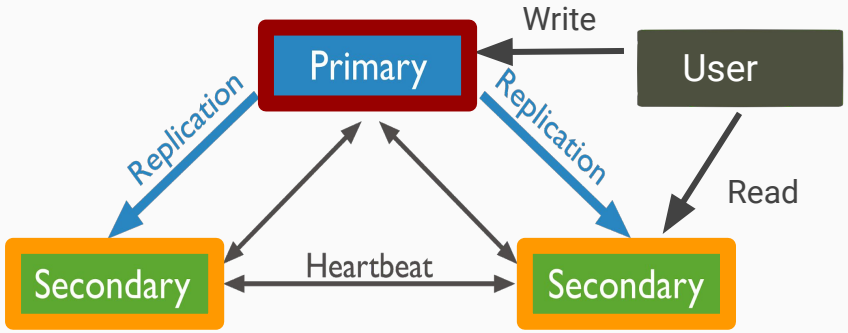
- Tyulenev et al. decided to synchronize clocks using stable cluster time (SCT)
- Uses the mongoDB oplog, operation log, to track writes and increment accordingly
- Few extra resources are used as it’s using features already built into MongoDB
- Uses a noop write to increase the time [7]

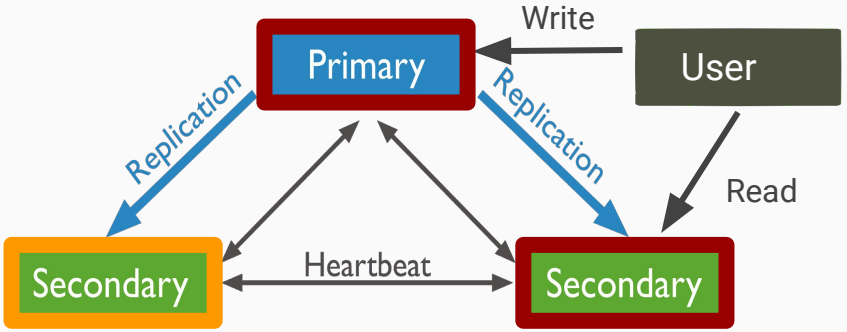
Effect on Throughput: Non-Durable Writes



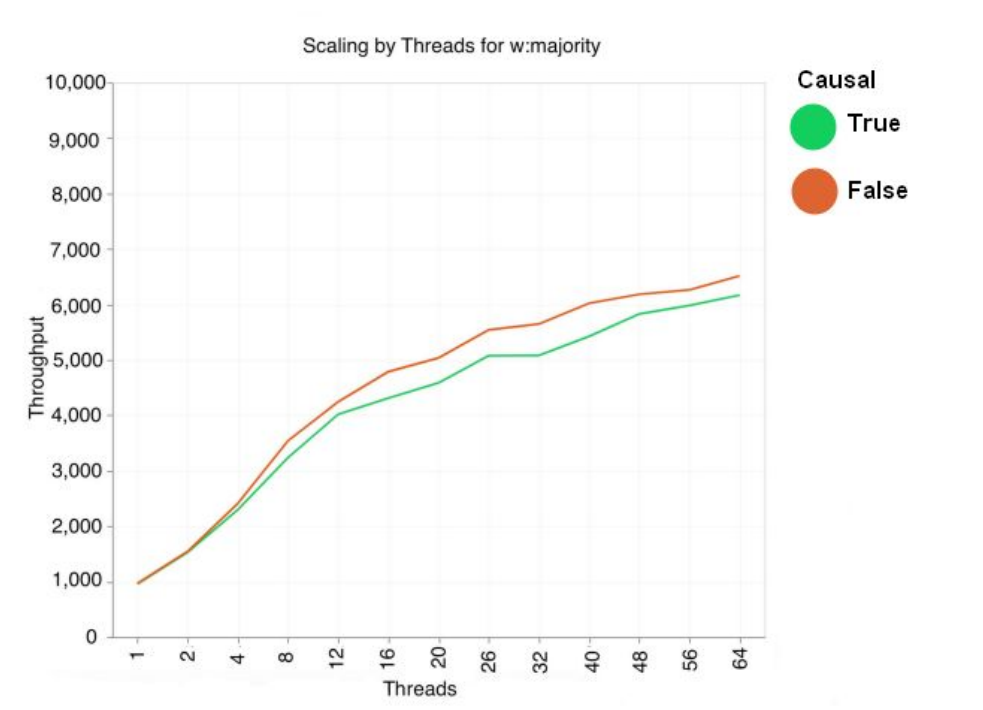
[7]

Effect on Throughput: Non-Durable Writes



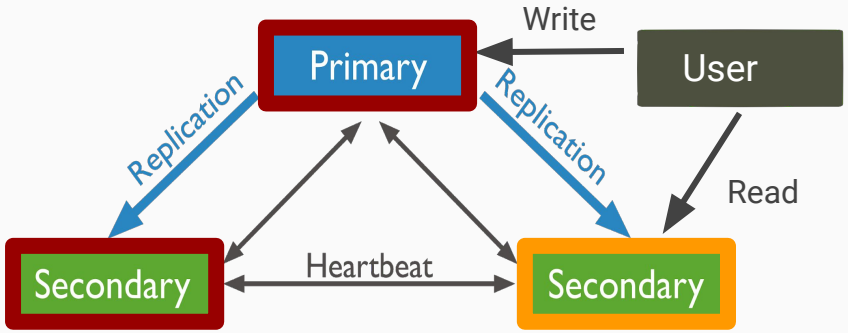


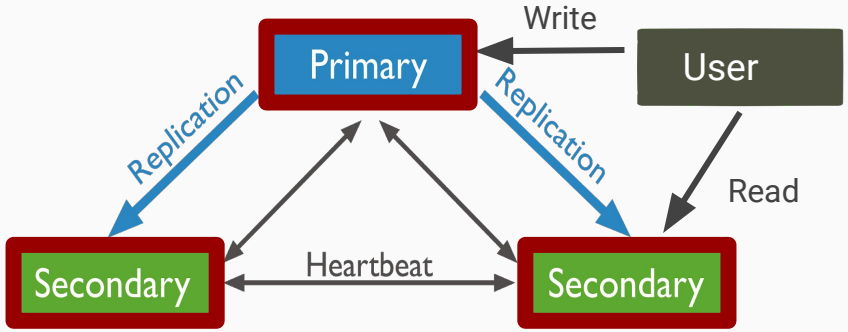
Effect on Throughput: Durable Writes



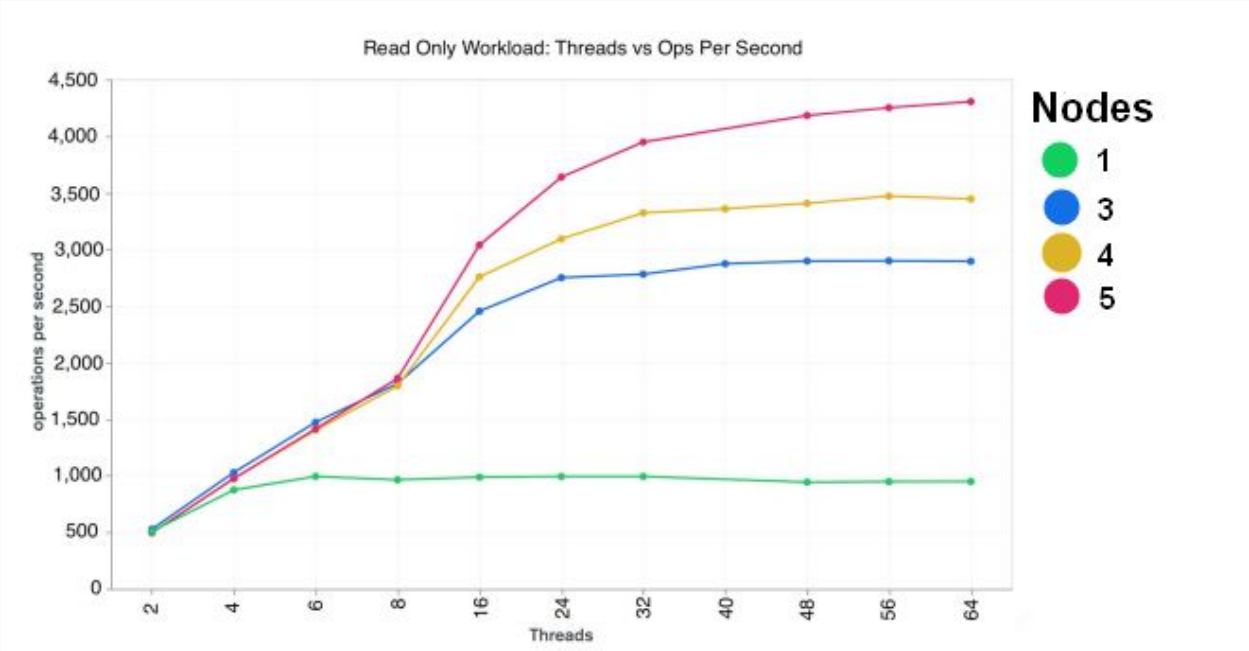
[7]

Effect on Throughput: Durable Writes





Effect on Throughput: Reads



[7]

Conclusion

- Bogdanov et al. found that the C++ driver was significantly better than the Java driver
- Tyulenev et al. found that causal consistency does bring down throughput, but doesn’t affect durable writes as much
- This means companies should use the C++ driver in the cloud and can use causal consistency without very little throughput loss if their data is durable

Acknowledgements

Thank you to Nic Mcphee, KK Lamberty, and Kevin Arhelger for feedback

Questions?

References

[1] K. Bogdanov, M. Pe´on-Quir´os, G. Q. Maguire, Jr., and D. Kosti´c. The nearest replica can be farther than you think. In Proceedings of the Sixth ACM Symposium on Cloud Computing, SoCC '15, pages 16–29, New York, NY, USA, 2015. ACM.
[2] GoogleData. Mobile site load time statistics. [Online; accessed 03-November-2019].

References

[3] MongoDB. Mongodb and mysql compared. [Online; accessed 03-November-2019].
[4] MongoDB. Replication. [Online; accessed 03-November-2019].
[5] MongoDB. Write concern. [Online; accessed 03-November-2019].
[6] M. Tyulenev, A. Schwerin, A. Kamsky, R. Tan, A. Cabral, and J. Mulrow. Implementation of cluster-wide logical clock and causal consistency in mongodb. In Proceedings of the 2019 International Conference on Management of Data, SIGMOD '19, pages 636–650, New York, NY, USA, 2019. ACM.

References

[7] Wikipedia. Causal consistency — Wikipedia, The Free Encyclopedia, 2019. [Online; accessed 03-November-2019].
[8] Wikipedia. Eventual consistency — Wikipedia, The Free Encyclopedia, 2019. [Online; accessed 03-November-2019].
[9] Wikipedia. Mongodb — Wikipedia, The Free Encyclopedia, 2019. [Online; accessed 03-November-2019].