# Băng thông

kịch bản:

- node1, node2, node3 và node 1 là primary

- đang insert 500 bản ghi vào primary (writeConcern mặc định: 1)

- primary network bandwidth về gần 0 (các secondaries còn lại bình thường)

- sau đó network giữa primary về lại bình thường

hiện tượng:

- client insert: chậm hơn hẳn (gần như bị treo)

- client read from primary: cũng bị treo

- client read from secondary: nothing happened (trả về không nhất quán)

- sau đó một secondary lên làm primary

- tốc độ client insert về bình thường

- client báo đã insert 500 records tuy nhiên sau đó check chỉ còn 498 (bay mất 2 chú)

- ngay khi primary join lại thì 2 record đã được insert của primary bị rollback lại (batch cũng có thể bị rollback)

+ những record đã được rollback nằm ở <dbpath>/rollback/

VD: /data/db/rollback/d80ff428-954a-43dc-95df-23dd7bebf0bf/removed.2021-08-11T03-09-02.1.bson

details blog:

<https://www.percona.com/blog/2018/05/01/mongodb-rollback-in-replicaset/>

phòng chánh:

- để writeConcern > 1 ( 1 là mặc định) để đảm bảo ngoài primary thì đã có secondary khác nhận được

và không xảy ra tình huống auto rollback gây thất thoát dữ liệu (ĐÃ TEST)

<https://docs.mongodb.com/manual/reference/command/setDefaultRWConcern/>

- hoặc set mặc định cho db như sau

db.adminCommand({

"setDefaultRWConcern" : 1,

"defaultWriteConcern" : {

"w" : 2

}

})

# Đóng socket

kịch bản:

- node1, node2, node3 với node1 là primary

- client đang insert vào node1 primary

- primary đột ngột mất mạng

+ rút cáp, tắt interface, ...

- sau đó nối cáp mạng lại cho primary

hiện tượng:

- client write: exception and stop

- client read from primary: exception and stop

- client read from secondary: treo và sau đó tiếp tục bình thường

- primary local check có 750 record

- secondary check có 250 record

- primary rollback 500 bản ghi

phòng chánh:

- writeConcern > 1

# Member Stale State

kịch bản:

- node1, node2, node3 với node1 là primary

- node3 không thể kết nối tới node1 và node2 (sync source)

+ mất mạng, đứt cáp

+ mạng chậm

- "sync source" node1 oplog collections vượt kích thước (những oplog cũ bị xoá)

hiện tượng:

- node3 không thể tiếp tục sync, rơi vào trạng thái RECOVERING mãi mãi (stale)

giải pháp:

- resync member

hiện tượng member bị "đi sau" primary này rất khó xảy ra

# Member rejoin

kịch bản:

node1, node2, node3, node4 với node1 là primary, node4 là node bên ngoài replica set

node1, node2, node3 trong một replica set

node3 down, sau đó dns của node3 trỏ tới node4

rejoin node3 sau đó

hiện tượng:

khi down rs.status() báo node3 không thể truy cập

sau cập nhật dns, rs.status() báo node3 online (tức node4)

node4 sync data từ node1 như bình thường

node3 restart và tiếp tục sync

từ đây trở đi không có lỗi gì xảy ra cả 2 node3 gốc và node3 mới (node4) đều sync bình thường

# Transaction

tất cả operations trong transaction đều isolate với nhau và với các secondary (kể cả với write concern)

chỉ khi commit, đóng transaction thì các operations này mới tới các secondary khác và dữ liệu được cập nhật

CAUTIONS: transaction trên test database không có oplog ?

transaction sẽ sinh ra oplog command applyOps

oplog này thuộc loại command sẽ được áp trực tiếp lên các secondary

# Data Synchronization

primary die và secondary có độ ưu tiên thấp nhưng được data mới nhất

kịch bản:

trước khi primary down data được sync tới node có độ ưu tiên thấp hơn

node có độ ưu tiên cao hơn thì chưa có data

hiện tượng:

node có độ ưu tiên cao hơn vẫn làm secondary

node có độ ưu tiên thấp làm primary

node có độ ưu tên cao sync từ node có độ ưu tiên thấp

sau đó node có độ ưu tiên cao hơn lên làm primary