

Міністерство освіти і науки України
НТУУ «Київський політехнічний інститут»
Фізико-технічний інститут

Проектування високонавантажених систем
Лабораторна робота №4
Налаштування реплікації та перевірка відмовостійкості MongoDB

Виконав:
Студент 4-го курсу
групи ФІ-21
Климент'єв Максим
Перевірив:

Зміст

1	Код реалізації	3
2	Результати	4

1 Код реалізації

2 Результати

I Налаштування реплікації

1. Налаштувати реплікацію в конфігурації: Primary with Two Secondary Members (P-S-S) (всі ноди можуть бути запущені як окремі процеси або у Docker контейнерах)

```
replicaset [direct: primary] test> rs.status()
{
  set: 'replicaset',
  date: ISODate('2025-12-14T14:45:33.199Z'),
  myState: 1,
  term: Long('1'),
  syncSourceHost: '',
  syncSourceId: -1,
  heartbeatIntervalMillis: Long('2000'),
  majorityVoteCount: 2,
  writeMajorityCount: 2,
  votingMembersCount: 3,
  writableVotingMembersCount: 3,
  optimes: {
    lastCommittedOpTime: { ts: Timestamp({ t: 1765723531, i: 1 }), t: Long('1') },
    lastCommittedWallTime: ISODate('2025-12-14T14:45:31.010Z'),
    readConcernMajorityOpTime: { ts: Timestamp({ t: 1765723531, i: 1 }), t: Long('1') },
    appliedOpTime: { ts: Timestamp({ t: 1765723531, i: 1 }), t: Long('1') },
    durableOpTime: { ts: Timestamp({ t: 1765723531, i: 1 }), t: Long('1') },
    writtenOpTime: { ts: Timestamp({ t: 1765723531, i: 1 }), t: Long('1') },
    lastAppliedWallTime: ISODate('2025-12-14T14:45:31.010Z'),
    lastDurableViewOpTime: ISODate('2025-12-14T14:45:31.010Z'),
    lastWrittenWallTime: ISODate('2025-12-14T14:45:31.010Z')
  },
  lastStableRecoveryTimestamp: Timestamp({ t: 1765723511, i: 1 }),
  electionCandidateMetrics: {
    lastElectionReason: 'electionTimeout',
    lastElectionDate: ISODate('2025-12-14T14:35:30.954Z'),
    electionTerm: Long('1'),
    lastCommittedOpTimeAtElection: { ts: Timestamp({ t: 1765722919, i: 1 }), t: Long('-1') },
    lastSeenWrittenOpTimeAtElection: { ts: Timestamp({ t: 1765722919, i: 1 }), t: Long('-1') },
    lastSeenOpTimeAtElection: { ts: Timestamp({ t: 1765722919, i: 1 }), t: Long('-1') },
    numVotesNeeded: 2,
    priorityAtElection: 2,
    electionTimeoutMillis: Long('10000'),
    numCatchUpOps: Long('0'),
    newTermStartDate: ISODate('2025-12-14T14:35:31.003Z'),
    wMajorityWriteAvailabilityDate: ISODate('2025-12-14T14:35:31.473Z')
  },
  members: [
    {
      _id: 0,
      name: 'mongo1:27017',
      health: 1,
      state: 1,
      stateStr: 'PRIMARY',
      uptime: 620,
      optime: { ts: Timestamp({ t: 1765723531, i: 1 }), t: Long('1') },
      optimeDate: ISODate('2025-12-14T14:45:31.000Z'),
      optimeWritten: { ts: Timestamp({ t: 1765723531, i: 1 }), t: Long('1') },
      optimeWrittenDate: ISODate('2025-12-14T14:45:31.000Z'),
      lastAppliedWallTime: ISODate('2025-12-14T14:45:31.010Z'),
      lastDurableViewOpTime: ISODate('2025-12-14T14:45:31.010Z'),
      lastWrittenWallTime: ISODate('2025-12-14T14:45:31.010Z'),
      syncSourceHost: '',
      syncSourceId: -1,
      infoMessage: '',
      electionTime: Timestamp({ t: 1765722930, i: 1 }),
      electionDate: ISODate('2025-12-14T14:35:30.000Z'),
      configVersion: 1,
      configTerm: 1,
      self: true,
      lastHeartbeatMessage: ''
    },
    {
      _id: 1,
      name: 'mongo2:27017',
      health: 1,
    }
  ]
}
```

```

state: 2,
stateStr: 'SECONDARY',
uptime: 613,
optime: { ts: Timestamp({ t: 1765723531, i: 1 }), t: Long('1') },
optimeDurability: { ts: Timestamp({ t: 1765723531, i: 1 }), t: Long('1') },
optimeWritten: { ts: Timestamp({ t: 1765723531, i: 1 }), t: Long('1') },
optimeDate: ISODate('2025-12-14T14:45:31.000Z'),
optimeDurabilityDate: ISODate('2025-12-14T14:45:31.000Z'),
optimeWrittenDate: ISODate('2025-12-14T14:45:31.000Z'),
lastAppliedWallTime: ISODate('2025-12-14T14:45:31.010Z'),
lastDurableWallTime: ISODate('2025-12-14T14:45:31.010Z'),
lastWrittenWallTime: ISODate('2025-12-14T14:45:31.010Z'),
lastHeartbeat: ISODate('2025-12-14T14:45:32.968Z'),
lastHeartbeatRecv: ISODate('2025-12-14T14:45:31.970Z'),
pingMs: Long('0'),
lastHeartbeatMessage: '',
syncSourceHost: 'mongo1:27017',
syncSourceId: 0,
infoMessage: '',
configVersion: 1,
configTerm: 1
},
{
_id: 2,
name: 'mongo3:27017',
health: 1,
state: 2,
stateStr: 'SECONDARY',
uptime: 613,
optime: { ts: Timestamp({ t: 1765723531, i: 1 }), t: Long('1') },
optimeDurability: { ts: Timestamp({ t: 1765723531, i: 1 }), t: Long('1') },
optimeWritten: { ts: Timestamp({ t: 1765723531, i: 1 }), t: Long('1') },
optimeDate: ISODate('2025-12-14T14:45:31.000Z'),
optimeDurabilityDate: ISODate('2025-12-14T14:45:31.000Z'),
optimeWrittenDate: ISODate('2025-12-14T14:45:31.000Z'),
lastAppliedWallTime: ISODate('2025-12-14T14:45:31.010Z'),
lastDurableWallTime: ISODate('2025-12-14T14:45:31.010Z'),
lastWrittenWallTime: ISODate('2025-12-14T14:45:31.010Z'),
lastHeartbeat: ISODate('2025-12-14T14:45:32.969Z'),
lastHeartbeatRecv: ISODate('2025-12-14T14:45:31.970Z'),
pingMs: Long('0'),
lastHeartbeatMessage: '',
syncSourceHost: 'mongo1:27017',
syncSourceId: 0,
infoMessage: '',
configVersion: 1,
configTerm: 1
}
],
ok: 1,
'$clusterTime': {
clusterTime: Timestamp({ t: 1765723531, i: 1 }),
signature: {
hash: Binary.createFromBase64('AAAAAAAAAAAAAAAAAAAAAAA='),
keyId: Long('0')
}
},
operationTime: Timestamp({ t: 1765723531, i: 1 })
}

```

2. Спробувати зробити запис з однією відключеною нодою та write concern рівнем 3 та нескінченім таймаутом. Спробувати під час таймаута включити відключену ноду

```

[INFO] [PRIMARY] Starting 1 client with w=3 write concern and no timeout...
[INFO] [START RESULT] Right before stopping PRIMARY...
{'_id': '1', 'likes': 0}
docker stop mongo1
mongo1
[INFO] [MID RESULT] Right after stopping PRIMARY...
[TIMEOUT] ServerSelectionTimeoutError
[TIMEOUT] ServerSelectionTimeoutError
[INFO] [MID RESULT] After 10 seconds stopping PRIMARY...
[TIMEOUT] ServerSelectionTimeoutError
docker start mongo1
mongo1

```

```

[INFO] [FINAL RESULT]
{'_id': '1', 'likes': 0}

[INFO] [SECONDARY] Starting 1 client with w=3 write concern and no timeout...
[INFO] [START RESULT] Right before stoping SECONDARY...
{'_id': '1', 'likes': 0}
docker stop mongo3
mongo3
[INFO] [MID RESULT] Right after stoping SECONDARY...
{'_id': '1', 'likes': 0}
[INFO] [MID RESULT] After 10 seconds stoping SECONDARY...
{'_id': '1', 'likes': 1}
docker start mongo3
mongo3
[INFO] [FINAL RESULT]
{'_id': '1', 'likes': 1}

```

3. Аналогічно попередньому пункту, але задати скінченний таймаут та дочекатись його закінчення. Перевірити чи данні записались і чи доступні на читання з рівнем readConcern: “majority”

```

[INFO] [PRIMARY] Starting 1 client with w=3 write concern and 'majority' read concern, but timeout = 3000...
[INFO] [START RESULT] Right before stoping PRIMARY...
{'_id': '1', 'likes': 1}
docker stop mongol
mongol
[INFO] [MID RESULT] Right after stoping PRIMARY...
[TIMEOUT] ServerSelectionTimeoutError
[TIMEOUT] ServerSelectionTimeoutError
[INFO] [MID RESULT] After 10 seconds stoping PRIMARY...
[TIMEOUT] ServerSelectionTimeoutError
docker start mongol
mongol
[INFO] [FINAL RESULT]
{'_id': '1', 'likes': 1}

[INFO] [SECONDARY] Starting 1 client with w=3 write concern and 'majority' read concern, but timeout = 3000...
[INFO] [START RESULT] Right before stoping SECONDARY...
{'_id': '1', 'likes': 1}
docker stop mongo3
mongo3
[INFO] [MID RESULT] Right after stoping SECONDARY...
{'_id': '1', 'likes': 1}
[TIMEOUT] WTimeoutError
[INFO] [MID RESULT] After 10 seconds stoping SECONDARY...
{'_id': '1', 'likes': 2}
docker start mongo3
mongo3
[INFO] [FINAL RESULT]
{'_id': '1', 'likes': 2}

```

4. Продемонстрував перевибори primary node відключивши поточний primary

5. і що після відновлення роботи старої primary на неї реплікуються нові дані, які з'явилися під час її простою

II Аналіз продуктивності та перевірка цілісності

Створити колекцію з каунтером лайків. Далі з 10 окремих клієнтів одночасно запустити інкрементацію каунтеру лайків по 10000 на кожного клієнта з різними опціями взаємодії з MongoDB.

1. Вказавши у параметрах findOneAndUpdate writeConcern = 1

```
db.collection.findOneAndUpdate({ _id: "5d3145e0f28f3d0017000001" }, { $inc: { value: 1 } }, { writeConcern: 1 })
```

2. Вказавши у параметрах findOneAndUpdate writeConcern = majority

```
db.collection.findOneAndUpdate({ _id: "5d3145e0f28f3d0017000001" }, { $inc: { value: 1 } }, { writeConcern: "majority" })
```

3. Повторно запустить код при writeConcern = 1, але тепер під час роботи відключіть Primary ноду і подивитись що буде обрана інша Primary нода, яка продовжить обробку запитів, і чи кінцевий результат буде коректним

```
db.collection.findOneAndUpdate({ _id: "5d3145e0f28f3d0017000001" }, { $inc: { value: 1 } }, { writeConcern: 1 })
```

4. Повторно запустить код при writeConcern = majority, але тепер під час роботи відключіть Primary ноду і подивитись що буде обрана інша Primary нода, яка продовжить обробку запитів, і чи кінцевий результат буде коректним

```
db.collection.findOneAndUpdate({ _id: "5d3145e0f28f3d0017000001" }, { $inc: { value: 1 } }, { writeConcern: "majority" })
```