

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

**Проектування високонавантажених систем**

Лабораторна робота №7  
Налаштування шардінгу в MongoDB

**Виконав:**

Студент 4-го курсу  
групи ФІ-21

Климент'єв Максим

**Перевірив:**

---

# **Зміст**

<b>1</b>	<b>Код реалізації</b>	<b>3</b>
<b>2</b>	<b>Результати</b>	<b>4</b>

# **1 Код реалізації**

## 2 Результати

1. Сконфігуруйте 3 інстанси (сервери) MongoDB у якості шард. Ці шарди потім будуть зв'язані з Zones (<https://www.mongodb.com/docs/manual/core/zone-sharding/>)
  - Ranged Sharding (with Zone Ranges)
  - Zones
2. Створіть колекцію Users

Окрім інших атрибутів, записи User мають містити атрибут region, який може приймати одне з трьох значень region = EU, USA, Asia

1. Сконфігуруйте шардинг таким чином, щоб кожна з шард була проасоційована з одним з регіоном: Shard 1 - EU, Shard 2 - USA, Shard 3 - Asia

І записи які будуть вставлятись в колекцію Users мають зберігатись на відповідній шарді в залежності від значення region

1. Створіть колекцію Tweets

Серед інших атрибутів, Tweet має містити кількість лайків - атрибут likes

1. Сконфігуруйте шарди таким чином, щоб кожна з них була проасоційована з діапазоном кількості лайків (Ranged Sharding): Shard 1 - 0-100, Shard 2 - 101-200, Shard 3 - 200 - ...

І записи які будуть вставлятись в колекцію Tweets мають зберігатись на відповідній шарді в залежності від значення likes

1. Перевірте появу шард і зон командою sh.status()

```
shardingVersion
{
  _id: ObjectId('69431a48b58cf6b823272255'),
  clusterId: ObjectId('69431a48b58cf6b823272254')
}
---
shards
[
  {
    _id: 'rs_shard1',
    host: 'rs_shard1/shard1:27017',
    state: 1,
    topologyTime: Timestamp({ t: 1766005323, i: 32 }),
    replSetConfigVersion: Long('-1'),
    tags: [ 'ZONE_EU', 'LIKES_LOW' ]
  },
  {
    _id: 'rs_shard2',
    host: 'rs_shard2/shard2:27017',
    state: 1,
    topologyTime: Timestamp({ t: 1766005323, i: 89 }),
    replSetConfigVersion: Long('-1'),
    tags: [ 'ZONE_USA', 'LIKES_MID' ]
  }
]
```

```

},
{
  _id: 'rs_shard3',
  host: 'rs_shard3/shard3:27017',
  state: 1,
  topologyTime: Timestamp({ t: 1766005323, i: 144 }),
  replSetConfigVersion: Long('-1'),
  tags: [ 'ZONE_ASIA', 'LIKES_HIGH' ]
}
]
---
active mongoses
[ { '8.2.2': 1 } ]
---
autosplit
{ 'Currently enabled': 'yes' }
---
balancer
{
  'Currently enabled': 'yes',
  'Failed balancer rounds in last 5 attempts': 0,
  'Currently running': 'no',
  'Migration Results for the last 24 hours': 'No recent migrations'
}
---
shardedDataDistribution
[
  {
    ns: 'RLS.Users',
    shards: [
      {
        shardName: 'rs_shard1',
        numOrphanedDocs: 0,
        numOwnedDocuments: 0,
        ownedSizeBytes: 0,
        orphanedSizeBytes: 0
      }
    ]
  },
  {
    ns: 'RLS.Tweets',
    shards: [
      {
        shardName: 'rs_shard1',
        numOrphanedDocs: 0,
        numOwnedDocuments: 0,
        ownedSizeBytes: 0,
        orphanedSizeBytes: 0
      }
    ]
  }
]
---
databases
[
  {
    database: { _id: 'config', primary: 'config', partitioned: true },
    collections: {}
  },
  {
    database: {
      _id: 'RLS',
      primary: 'rs_shard1',
      version: {
        uuid: UUID('38c823ff-2f01-4dd9-8cd3-9aa806cef7b3'),
        timestamp: Timestamp({ t: 1766005323, i: 172 }),
        lastMod: 1
      }
    },
    collections: {
      'RLS.Tweets': {
        shardKey: { likes: 1 },
        unique: false,
        balancing: true,
        allowMigrations: true,
        chunkMetadata: [ { shard: 'rs_shard1', nChunks: 1 } ],
        chunks: [
          { min: { likes: MinKey() }, max: { likes: MaxKey() }, 'on shard': 'rs_shard1', 'last modified': Timestamp({ t: 1, i: 0 }) }
        ],
        tags: [
      
```

```

        {
          tag: 'LIKES_LOW',
          min: { likes: 0 },
          max: { likes: 101 }
        },
        {
          tag: 'LIKES_MID',
          min: { likes: 101 },
          max: { likes: 201 }
        },
        {
          tag: 'LIKES_HIGH',
          min: { likes: 201 },
          max: { likes: MaxKey() }
        }
      ],
    },
    'RLS.Users': {
      shardKey: { region: 1, _id: 1 },
      unique: false,
      balancing: true,
      allowMigrations: true,
      chunkMetadata: [ { shard: 'rs_shard1', nChunks: 1 } ],
      chunks: [
        { min: { region: MinKey(), _id: MinKey() }, max: { region: MaxKey(), _id: MaxKey() }, 'on shard': 'rs_shard1', 'last modified': Timestamp({ t: 1, i: 0 }) }
      ],
      tags: [
        {
          tag: 'ZONE_ASIA',
          min: { region: 'Asia', _id: MinKey() },
          max: { region: 'Asia', _id: MaxKey() }
        },
        {
          tag: 'ZONE_EU',
          min: { region: 'EU', _id: MinKey() },
          max: { region: 'EU', _id: MaxKey() }
        },
        {
          tag: 'ZONE_USA',
          min: { region: 'USA', _id: MinKey() },
          max: { region: 'USA', _id: MaxKey() }
        }
      ]
    }
  ]
}

```

2. Продемонструйте роботу шардінгу (тобто що записи зберігаються на різних нодах):

- відключити одну з ноди

- спробувати додати записи зі значеннями shard key (Ranged та Zones), що потрапляють на відключенну ноду

- спробувати додати записи зі значеннями shard key (Ranged та Zones), що потрапляють на працюочу ноду

- спробувати знайти всі записи з shard key для Zone, яка відповідає ноді яка працює/яка не працює

- спробувати знайти записи для shard key з певного проміжку, який входить до проміжку працюючої ноди для Ranged Sharding

- спробувати знайти записи для shard key з певного проміжку, який входить до проміжку ноди яка не працює для Ranged Sharding

3. Включити відключену ноду та перевірити працездатність запитів з попереднього пункту

Всі завдання можуть бути виконані з командного рядка

Протокол має містити команди та кінцеві налаштування з sh.status()