

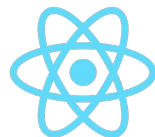
DDBS Project

Louis Rollet & Théodore Billotte

Project Objectives

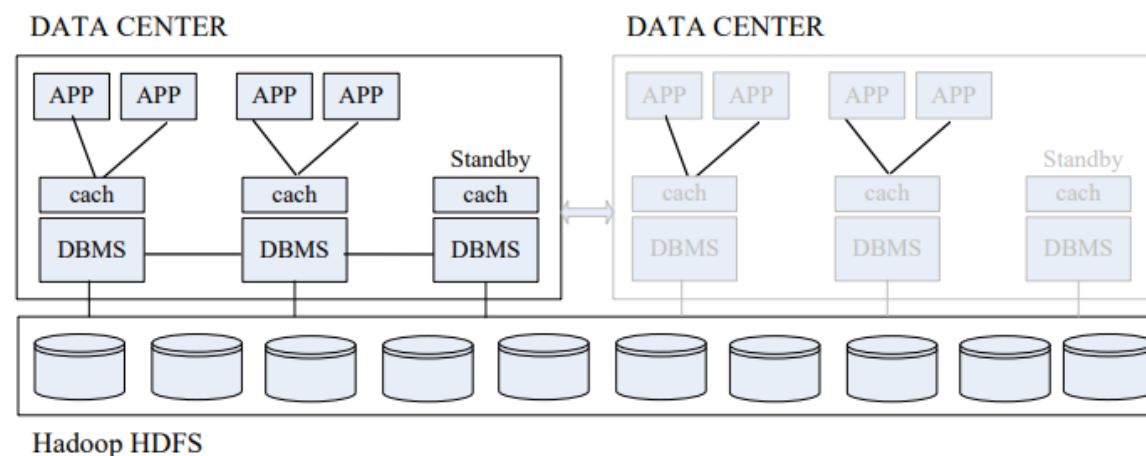
- **Goal:** Design and implement a distributed data center for a news application.
- **Core Challenge:** Manage massive amounts of structured (relational) and unstructured (media) data across geographically separated nodes.
- **Key Deliverable:** A system capable of bulk loading, efficient querying, and live monitoring.

The Technology Stack

- Containerization:  Docker  Compose
- Database Engine:  mongoDB®
- Caching:  redis
- File Storage:  HDFS
- Backend:   Flask
web development,
one drop at a time
- Frontend:  React

System Architecture

- Simulates a real-world Data Center context.
- **Components:**
 - **Frontend Client**
 - **API Gateway:** Routes queries to the correct shard based on "tags".
 - **Shards (DBMS):** Physical storage nodes
 - **Config Server:** Stores cluster metadata.

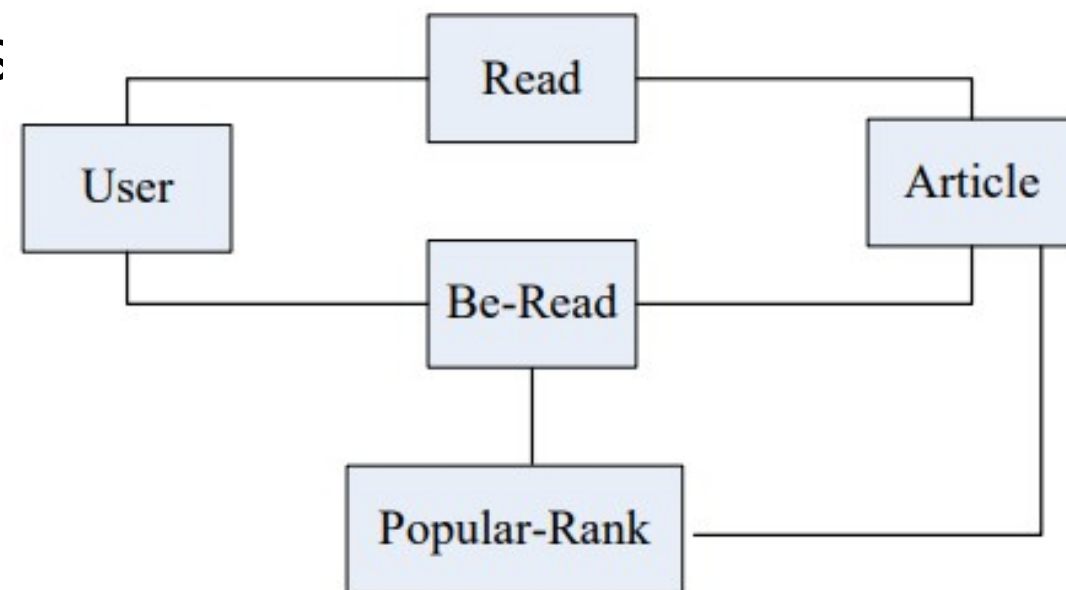


Data Model & Relationships

- **Structured Data:** 5 Relational Tables

- **Relations:**

- Users *Read* Articles.
- Articles generate *Be-Read* stats.
- Stats aggregate into *Popular-Rank*.



Sharding Strategy (Structured Data)



- **User & Read Tables:** Geo-partitioned by Region.
 - *Beijing* → DBMS 1.
 - *Hong Kong* → DBMS 2.
- **Article & Be-Read Tables:** Partitioned by Category.
 - *Science* → Replicated on DBMS 1 & 2.
 - *Technology* → DBMS 2 (initially).
- **Popular-Rank:** Partitioned by Time.
 - *Daily* → DBMS 1.
 - *Weekly/Monthly* → DBMS 2.

Unstructured Data (HDFS)



- Manage text, images, and video associated with articles.
- **Implementation:**
 - Files stored in Hadoop HDFS.
 - MongoDB stores the *file paths*.
 - API fetches content on-demand during article retrieval.

Advanced Features

- Expansion at the DBMS-level allowing a new server to join.
- **We use Configuration-Driven Architecture:**
 - We only need to update a JSON config.
 - **Capabilities:**
 - Hot/Cold Standby nodes.
 - Live Data Migration
 - Zero-downtime expansion.

```
"shards": [  
  {  
    "id": "rsShard1",  
    "host": "rsShard1/shard1:27018,shard1_standby:27018",  
    "tags": ["Beijing", "Daily", "Science_Original"]  
  },  
  {  
    "id": "rsShard2",  
    "host": "rsShard2/shard2:27018",  
    "tags": ["HongKong", "Technology", "Aggregated", "Science_Replica"]  
  }  
],
```

{ j s o n }

Demo

```
#-----  
# SHARD 1 (Beijing/Science/Daily) -- Replica Set  
#-----  
shard1:  
  image: mongo:8.0  
  container_name: shard1  
  command: mongod --shardsvr --replSet rsShard1 --port 27018 --bind_ip_all  
  ports: ["27018:27018"]  
  networks: [app_net]  
  volumes: [shard1_data:/data/db]  
  
shard1_standby:  
  image: mongo:8.0  
  container_name: shard1_standby  
  command: mongod --shardsvr --replSet rsShard1 --port 27018 --bind_ip_all  
  networks: [app_net]  
  depends_on: [shard1]  
  
#-----  
# SHARD 2 (HongKong/Tech/Aggregated) -- Replica Set  
#-----  
shard2:  
  image: mongo:8.0  
  container_name: shard2  
  command: mongod --shardsvr --replSet rsShard2 --port 27018 --bind_ip_all  
  ports: ["27028:27018"]  
  networks: [app_net]  
  volumes: [shard2_data:/data/db]
```

```
"shards": [  
  {  
    "id": "rsShard1",  
    "host": "rsShard1/shard1:27018,shard1_standby:27018",  
    "tags": ["Beijing", "Daily", "Science_Original"]  
  },  
  {  
    "id": "rsShard2",  
    "host": "rsShard2/shard2:27018",  
    "tags": ["HongKong", "Technology", "Aggregated", "Science_Replica"]  
  }  
],
```

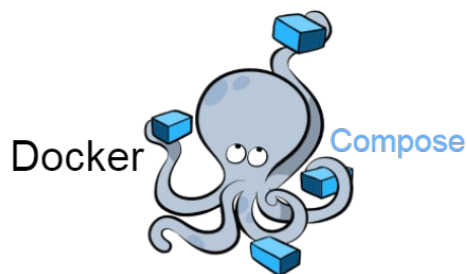
{j s o n}

Docker

Compose

Using 3 DBMS

```
shard3:
  image: mongo:8.0
  container_name: shard3
  command: mongod --shardsvr --replSet rsShard3 --port 27018 --bind_ip_all
  ports: ["27038:27018"]
  networks: [app_net]
  volumes: [shard3_data:/data/db]
```



```
"shards": [
  {
    "id": "rsShard1",
    "host": "rsShard1/shard1:27018,shard1_standby:27018",
    "tags": ["Beijing", "Daily", "Science_Original"]
  },
  {
    "id": "rsShard2",
    "host": "rsShard2/shard2:27018",
    "tags": ["HongKong", "Aggregated", "Science_Replica"]
  },
  {
    "id": "rsShard3",
    "host": "rsShard3/shard3:27018",
    "tags": ["Technology"]
  }
],
```

{j s o n}

Using 3 DBMS Results

Article Collection

Strategy:

rsShard1

4466

rsShard2

4466

rsShard3

5534

System Architecture

 **rsShard1**
rsShard1

Tags:

Beijing Daily Science_Original

Collections:

article
popular_rank
read
be_read
user

 **rsShard2**
rsShard2

Tags:

HongKong Aggregated Science_Replica

Collections:

article
popular_rank
read
be_read
user

 **rsShard3**
rsShard3

Tags:

Technology

Collections:

article
be_read

Thanks for listening !

Louis Rollet & Théodore Billotte