

*Máster Universitario en Tecnologías Web, Computación
en la Nube y Aplicaciones Móviles*

Persistencia Relacional y No Relacional de Datos

Trabajo final

Curso 2018-2019

LUIS ARAUJO

Contenido

Introducción.....	3
Tecnología NoSQL: MongoDB	4
Modelo de Datos.....	5
Consultas.....	6

Introducción

Este trabajo presenta una propuesta de solución para una aerolínea que quiere llevar un control exhaustivo de la actividad de compras y tiempo de espera de los pasajeros en la zona de embarque. En concreto, se quiere diseñar un modelo de datos NoSQL que registre la fecha y hora de paso por el arco de seguridad de cada pasajero, la hora en la que realizó el embarque, la puerta de embarque y el aeropuerto al cual se dirigía. Además, se quiere registrar la información de los gastos que realizan los pasajeros en las diferentes tiendas situadas en la zona de embarque del aeropuerto.

El objetivo del modelo es realizar explotaciones del tiempo medio que pasan en la zona de embarque los pasajeros que se dirigen a un determinado aeropuerto, calculado para un año especificado, del tiempo medio que pasan los pasajeros que se dirigen a una determinada puerta de embarque, calculado para un año especificado, y también del importe total gastado y el número de pasajeros que han comprado en una tienda en concreto durante un año especificado, pudiéndose especificar o no el aeropuerto al que se dirigían.

Como medida del volumen de información, hay que considerar que la aerolínea opera en 80 aeropuertos, con una media de 40 vuelos diarios con 180 pasajeros cada uno (500K registros diarios).

Además del modelo, se explica la tecnología NoSQL elegida y por qué es la mejor opción en base a los requisitos y condiciones en que se implementará.

Tecnología NoSQL: MongoDB

Entre las tecnologías NoSQL consideradas para dar solución al problema estaban:

- Base de datos orientadas a columnas (Cassandra)
- Base de datos orientadas a documentos (MongoDB)

Luego de hacer un análisis del Teorema de CAP, en que solamente se pueden cumplir dos características de tres disponibles (Consistencia, Disponibilidad y Tolerancia a particiones).

Cassandra cumple Tolerancia a particiones y Disponibilidad, esto significa que el servicio a los usuarios siempre estará funcionando a pesar de que se puedan producir situaciones de avería que causen que el sistema quede particionado en diferentes componentes. La disponibilidad significa que las peticiones enviadas por los usuarios a un nodo que está disponible deben obtener su debida respuesta, siempre genera una respuesta, aunque esta no sea consistente con la información que esté en otras réplicas, esto supone alta disponibilidad.

Por otro lado, MongoDB cumple las características de Consistencia y Tolerancia a Particiones. Al igual que Cassandra, significa que el servicio a los usuarios siempre estará funcionando a pesar de que se puedan producir situaciones de avería que causen que el sistema quede particionado en diferentes componentes. Con la consistencia se recuperan siempre los mismos valores para unos mismos datos en un mismo instante de tiempo, es decir, perdiendo alta disponibilidad.

Considerando estas características la tecnología utilizada para dar solución al problema planteado fue MongoDB, es decir una base de datos orientadas a documentos. El problema planteado, hace mención del volumen de información empleado, que equivale a 500K registros diarios, lo cual hace que sea lógico utilizar un modelo de base de datos el cual sea consistente, en donde todas peticiones de información que se realicen se recuperan siempre los mismos valores y a su vez empleando el menor coste posible.

MongoDB permite utilizar índices compuestos, es esencial tener índices cuando se trabajan con colecciones con volúmenes grandes. La diferencia entre realizar una consulta sobre campo con índice, y realizarla sin él, puede ser crítica. En concreto, las consultas planteadas en el problema deberían tener índice por Aeropuerto, en el caso de las puertas de embarque, y otro índice por tiendas en el caso de las compras en las tiendas del aeropuerto.

Modelo de Datos

Una vez elegido MongoDB, se ha definido el modelo de datos, los cuales son documentos JSON. En concreto se ha establecido 3 colecciones:

- Passengers: Información personal de los pasajeros.
- AirportControl: Información de hora de paso por puerta de seguridad y hora de embarque.
- Shops: Información de las compras realizadas por un pasajero en una tienda en concreto.

Colección Passengers:

```
{
  "passport": "ce345f",
  "name": "Leyre",
  "lastName": "Spaliviero"
}
```

Colección airportControl:

```
{
  "identification": "21c3",
  "securityGateDate": "2019-06-03T23:00:00Z",
  "boardingDate": "2019-06-03T23:40:00Z",
  "gate": "A21",
  "airportDestiny": "LAX"
}
```

Colección shops:

```
{
  "customer": "21c3",
  "shopDate": "2019-06-05T13:00:00Z",
  "airportDestiny": "TGU",
  "shopID": "CandyShop",
  "items": [
    {
      "productName": "Water",
      "price": 50,
      "quantity": 4
    },
    {
      "productName": "Candies",
      "price": 2,
      "quantity": 5
    }
  ]
}
```

Consultas

Antes de realizar las consultas se establecieron los índices para los campos críticos de las dos colecciones principales.

```
db.airportControl.createIndex( { "airportDestiny": 1 } )
db.shops.createIndex( { "shopID": 1 } )
```

Para la primera consulta, se pide calcular el tiempo medio que pasan en la zona de embarque los pasajeros que se dirigen a un determinado aeropuerto, calculado para un año especificado.

```
db.airportControl.aggregate([
  { "$match": {
    "$and": [
      { "$expr": { "$eq": [{ "$year": "$boardingDate" }, 2018] } },
      { "airportDestiny": "MAD" }
    ]
  } },
  { "$group": {
    "_id": null,
    "average": {
      "$avg": { "$subtract": ["$boardingDate", "$securityGateDate"] }
    }
  } }
])
```

La segunda consulta solicita el tiempo medio que pasan los pasajeros que se dirigen a una determinada puerta de embarque, calculado para un año especificado.

```
db.airportControl.aggregate([
  { "$match": {
    "$and": [
      { "$expr": { "$eq": [{ "$year": "$boardingDate" }, 2018] } },
      { "gate": "A22" }
    ]
  } },
  { "$group": {
    "_id": null,
    "average": {
      "$avg": { "$subtract": ["$boardingDate", "$securityGateDate"] }
    }
  } }
])
```

Para la tercera consulta, se pide calcular el importe total gastado y el número de pasajeros que han comprado en una tienda en concreto durante un año especificado, pudiéndose especificar o no el aeropuerto al que se dirigían.

```
db.shops.aggregate([
  { $unwind: "$items" },
  { "$match": {
    "$expr": { "$eq": [{ "$year": "$shopDate" }, 2019] }
  } },
  {
    $group : {
      _id : "$shopName",
      totalSold: { $sum: { $multiply: [ "$items.price",
"$items.quantity" ] } },
      totalCustomers:{ $sum:1}
    }
  }
]);
```

```
db.shops.aggregate([
  { $unwind: "$items" },
  { "$match": {
    "$and": [
      { "$expr": { "$eq": [{ "$year": "$shopDate" }, 2018] } },
      { "airportDestiny": "TGU" }
    ]
  } },
  {
    $group : {
      _id : "$shopID ",
      totalSold: { $sum: { $multiply: [ "$items.price",
"$items.quantity" ] } },
      totalCustomers:{ $sum:1}
    }
  }
]);
```