03. 3주차 샤딩

샤딩을 통한 시스템 확장

MongoDB의 샤딩은 데이터베이스를 작은 조각으로 분할하여 단일 머신이 모든 데이터를 저장하거나 전체 로드를 처리하지 않도록 하기위해 고안된 것.
MongoDB의 샤딩은 애플리케이션에 투명성을 가지는데, 이것은 샤딩 클러스터를 쿼리하기 위한 인터페이스가 복제 세트 또는 단일 mongod 서버 인스턴스를 쿼리하기 위한 인텁페이스와 완전히 동일하다는 것을 뜻함.

샤딩이란?

대규모 데이터 세트를 작고 관리하기 쉬운 조각으로 분할하는 과정. 데이터 크기가 커지고 애플리케이션이 더 많은 읽기-쓰기 처리량을 필요로 함에 따라 작업 부하를 효율적으로 관리하기 위해 사용.
샤딩은 저장소 분산과 부한 분산이라는 두 가지 주요한 목적을 가지고 있음. 샤딩은 모든 성능 문제를 해결하는 것이 아니며, 추가적인 복잡성과 오버 헤드를 발생시키므로 왜 샤딩을 해야 하는지 이해하는 것이 중요.
샤딩이 최적의 솔루션이 아닐수도 있다.

저장소 분산

MongoDB는 모든 데이터를 --dbpath 옵션으로 지정된 경로에 저장함으로 호스트 운영체제를 모니터링할 수 있어야 한다. Mongo 셸에서 db.stats() 또는 db.collection.stats()를 실행하여 데이터베이스의 저장소 사용에 대한 현재 상태의 통계와 컬렉션에 대한 통계가 각각 출력된다.

부하 분산

시스템에서 지원해야 하는 부하(Load), 즉 클라이언트의 요청에 의해 사용된 CPU, RAM 및 Disk I/O 대역폭에 대한 자원 사용량에 대한 분산을 의미한다. 특히 MongoDB의 샤딩은 인덱스와 작업 데이터 셋을 메모리에 유지하는 것이 중요한데, 인덱스와 데이터 셋이

무한하게 증가하게 되면 더 이상 메모리에 적재할 수 없는 순간이 온다. 여러 서버에 메모리를 증설하여 통합해서 사용할 수 있는 크기를 늘리는 것이 샤딩의 가장 중요한 목적이다.

샤드 클러스터의 구성요소

Shard: 애플리케이션 데이터를 저장. 샤드 클러스터는 오직 mongos 라우터 또는 시스템 관리자만 직접 샤드에 연결해야 한다.

Mongos Router: 클러스터 메타데이터를 캐시하고 이를 사용하여 올바른 샤드에 작업을 라우팅하거나 샤딩한다.

Config Server: 클러스터에 대한 메타데이터를 영구적으로 저장하며, 어떤 샤드가 어떤 데이터 하위 집합을 가지는가 포함된다.

샤드된 클러스터에서의 데이터 분산

샤드된 MongoDB에는 도큐먼트, 청크, 컬렉션, 데이터베이스 이렇게 네 가지 단위가 있다.

- 도큐먼트: MongoDB에서 가장 작은 데이터 단위.
- 청크: 필드의 값 별로 클러스터된 도큐먼트 그룹. 청크는 샤드에만 존재하는 개념. 이는 샤드 키라는 필드 또는 필드 집합에 대한 값을 기반으로 하는 도큐먼트의 논리적 그룹.
- 컬렉션: 데이터베이스 내에 명명된 그룹화된 도큐먼트.
- 데이터베이스: 도큐먼트 모음을 포함하는 시스템의 최상위 레벨 그룹.

샤드된 클러스터에 데이터를 분산시킬 수 있는 방법

컬렉션의 파티션 또는 청크 수준에서 컬렉션의 도큐먼트 자체가 분할되어 도큐먼트의 샤드 키(shard key)라는 필드 또는 집합의 값에 따라 여러 샤드에 분산된다. 그럼 청크를 사용하는 이유는 무엇일까?
MongoDB에서 데이터를 저장할 때 분할해서 저장하는 것 뿐만아니라 분할한 데이터를 다시 찾아야 하는 것을 고려해야 한다. 이 경우 도큐먼트 수준에서 분할을 하게되면 모든 단일 도큐먼트를 독립적으로 추적하여 config 서버에 메타데이터를 저장해야 할 필요가

있게 된다. 작은 도큐먼트가 많은 시스템에서 실제 저장되는 데이터의 위치 정보를 가진 메타데이터 만으로 config서버의 데이터가 가득 차게 될 것이다.

청크는 미리 정의된 샤드 키(Shard key)라는 필드 세트를 기반으로 도큐먼트의 논리적 그룹의 개념을 정의한 것이다.

코어에서 MongoDB의 샤딩은 범위기반(range-base)이다. 즉, 각 '청크'는 샤드 키(Shard key)의 범위를 나타낸다.

샤드 클러스터 구성

https://rastalion.me/mongodb-shard-cluster-구성하기/

샤드 확인

```
> db.shard.find()

> db.getSiblingDB("config").shards.find()

> use admin
> db.runCommand({listShards: 1})

> sh.status()
```

샤딩 컬렉션

```
> sh.enableSharding("<database>")
```

샤드에 있는 데이터 보기

```
> db.chunk.count({"shard-a"})
```

샤딩 클러스터 질의 및 인덱싱

애플리케이션 시점에서는 샤드 클러스터에 질의하는 것과 단일 Mongod에 질의 하는것 사이에 차이가 없다.

쿼리 라우팅

Config 서버는 키의 범위를 샤드에 대해 매핑한 것을 가지고 있다. 이 것은 즉, 청크에 대한 맵핑이다.

샤드 키가 쿼리에 포함되어 있으면 mongos는 신속하게 청크데이터를 참고해서 질의의 결과값이 어느 샤드에 포함되어 있는지 결정한다. 이 것을 **타깃 쿼리**라고 한다.

만약 샤드 키가 쿼리에 들어있지 않다면 쿼리 플래너는 모든 샤드를 다 방문해야한다. 이 걸은 **글로벌 쿼리** 혹은 **스캐터/개더 쿼리** 라고 한다.

샤드 클러스터 인덱싱하기

- 각 샤드는 자신만의 인덱스를 가진다. 샤드 컬렉션에 대해 인덱스를 선언하면 각 샤드는 자신이 담당하고 있는 부분의 컬렉션에 대해 별도의 인덱스를 생성한다. mongos 라우터에 연결된 상태에서 `db.users.createIndex()` 명령을 실행하면 각 샤드가 개별적으로 인덱스 생성 명령을 수행한다.
- 각 샤드에 대한 샤드 컬렉션은 같은 인덱스가 있어야 한다. 그렇지 않으면 성능이 일관적이지 못하게된다.
- 샤드 컬렉션은 _id필드와 샤드 키에 대해서만 Unique 인덱스를 허용한다. 다른 필드에 대한 Unique 인덱스는 금지되는데, 이것을 허용하는 것은 MongoDB의 샤딩 기본 설계에 위배되는것이기 때문이다.