

Sharding

데이터베이스를 분산하여 부하를 해결하는 방법



1. 발표자

샤딩의 개념 및 아키텍처



지현이

샤딩의 종류 및 적용



김선재

2. 목차

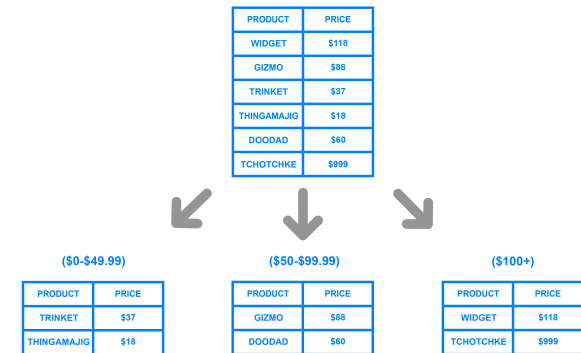
1. Sharding이란?
2. Sharding vs Partitioning
3. 아키텍처
4. Sharding의 종류
5. 스프링에서 적용하기

3. Sharding 이란?

- 같은 테이블 스키마를 가진 데이터를 다수의 데이터베이스에 분산하여 저장하고 처리하는 기술
- 분산 처리를 위한 솔루션 중 하나
- 고가용성을 위한 전략과는 다르다

Sharding을 위한 구성요소

1. 라우터 서버
2. 컨피그 서버
3. 샤드 서버



4. Sharding vs Partitioning

수평 파티셔닝 == 행 기준

수직 파티셔닝 == 열 기준

장점

1. 가용성

- 전체 데이터의 훼손 가능성이 줄어들고 데이터 가용성이 향상된다.

2. 관리용이성

- 큰 테이블이 관리하기 쉬운 용량으로 분리된다.

3. 성능

- 특정 DML과 Query의 성능을 향상시키며, 대용량 Data Write 에서 효율적이다.
- 많은 Insert 가 있는 시스템에서 Insert 작업을 분리된 파티션들로 분산시켜 준다.

4. Sharding vs Partitioning

수평 파티셔닝 == 행 기준

수직 파티셔닝 == 열 기준

단점

1. 수직 파티셔닝 시 테이블간 **조인에 대한 비용**이 증가한다.
2. 테이블과 인덱스를 별도로 파티션할 수는 없으며,
테이블과 인덱스를 같이 파티셔닝 해야 한다.

4. Sharding vs Partitioning

- 하나의 DB에 데이터가 늘어나면 용량 이슈도 생기고 느려져서 자연스레 서비스 성능에 영향을 끼침
=> DB 트래픽을 분산하는 목적으로 고려해야함

장점

1. 나누어 처리하여 한대의 DB에 네트워크 과부하가 걸릴 가능성이 낮아진다.
2. 대용량 정보 처리 시 분산되어 있어 용량 부담이 저해된다.
3. 특정 DB의 장애가 전면적인 장애로 이어지지 않는다.

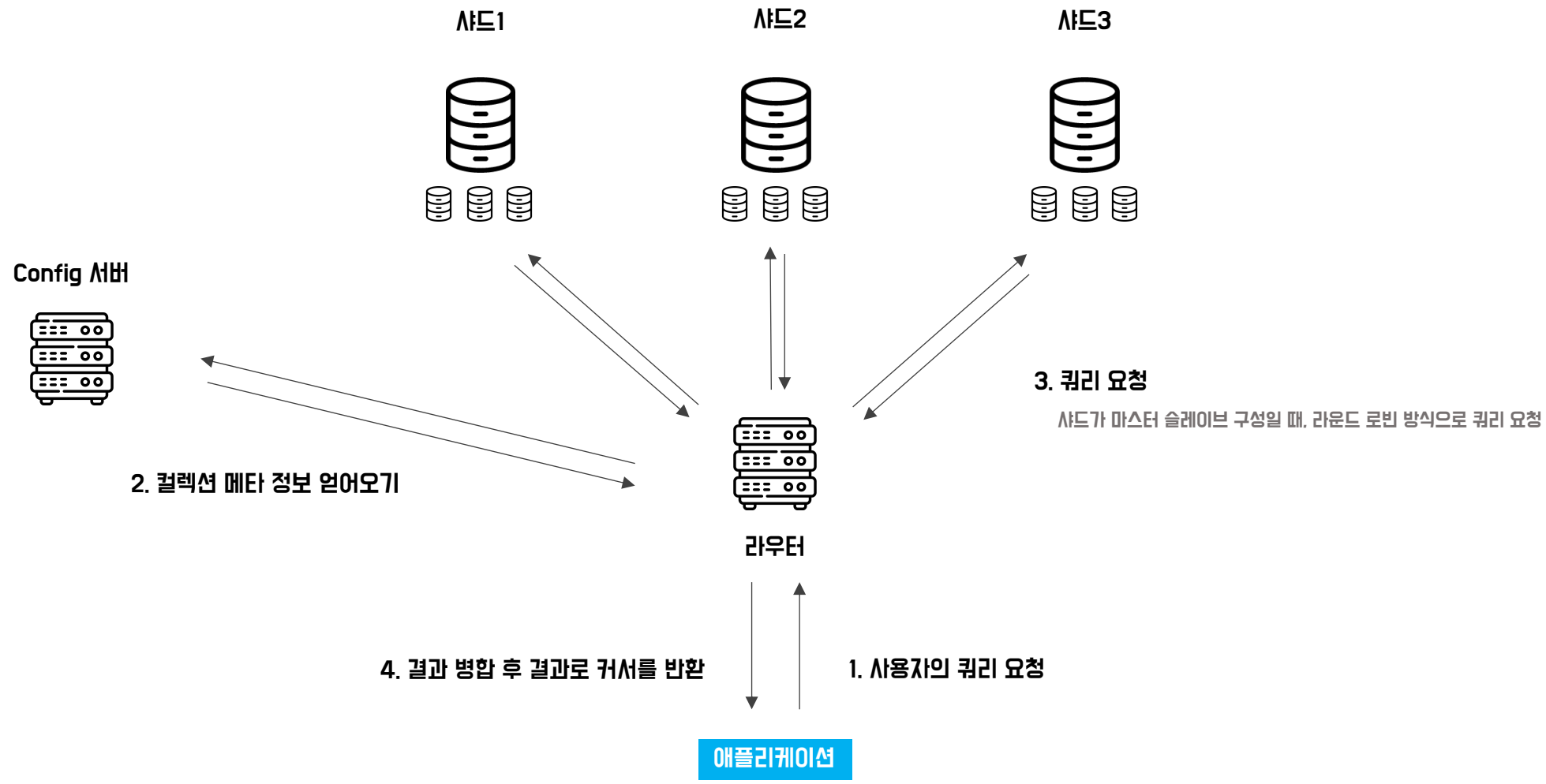
단점

1. 프로그래밍 운영 복잡도가 높아진다.

4. Sharding vs Partitioning

샤딩은 파티셔닝을 수행하는 방식 중의 일부!
수평 파티셔닝을 진행할 때의 디자인 패턴 중 하나이다.

4. Sharding 아키텍처



5. 샷팅의 종류

모듈러 샷팅

레인지 샷팅

5. 샷팅의 종류

모듈러 샷팅

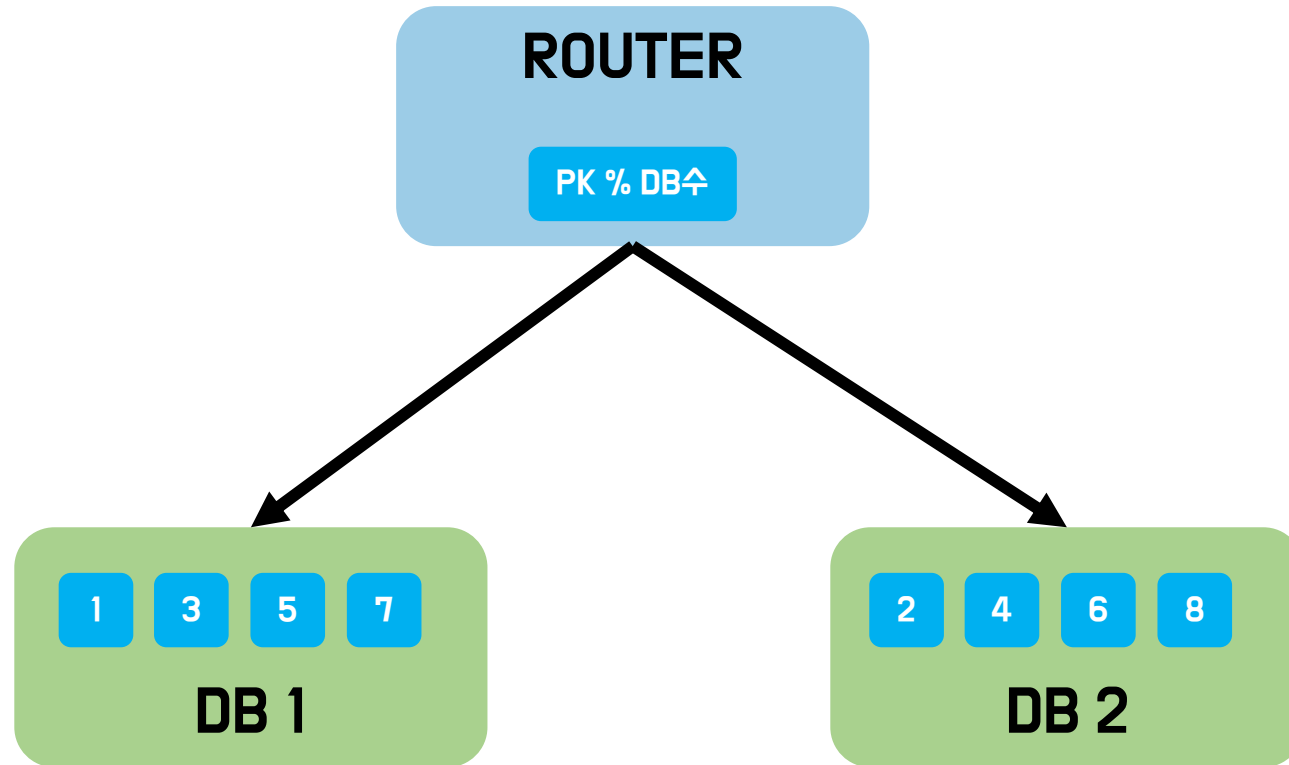
레인지 샷팅

5. 샷팅의 종류

모듈러 샷팅

레인지 샷팅

5. 샤딩의 종류 - 모듈러 샤딩



5. 샤딩의 종류 - 모듈러 샤딩

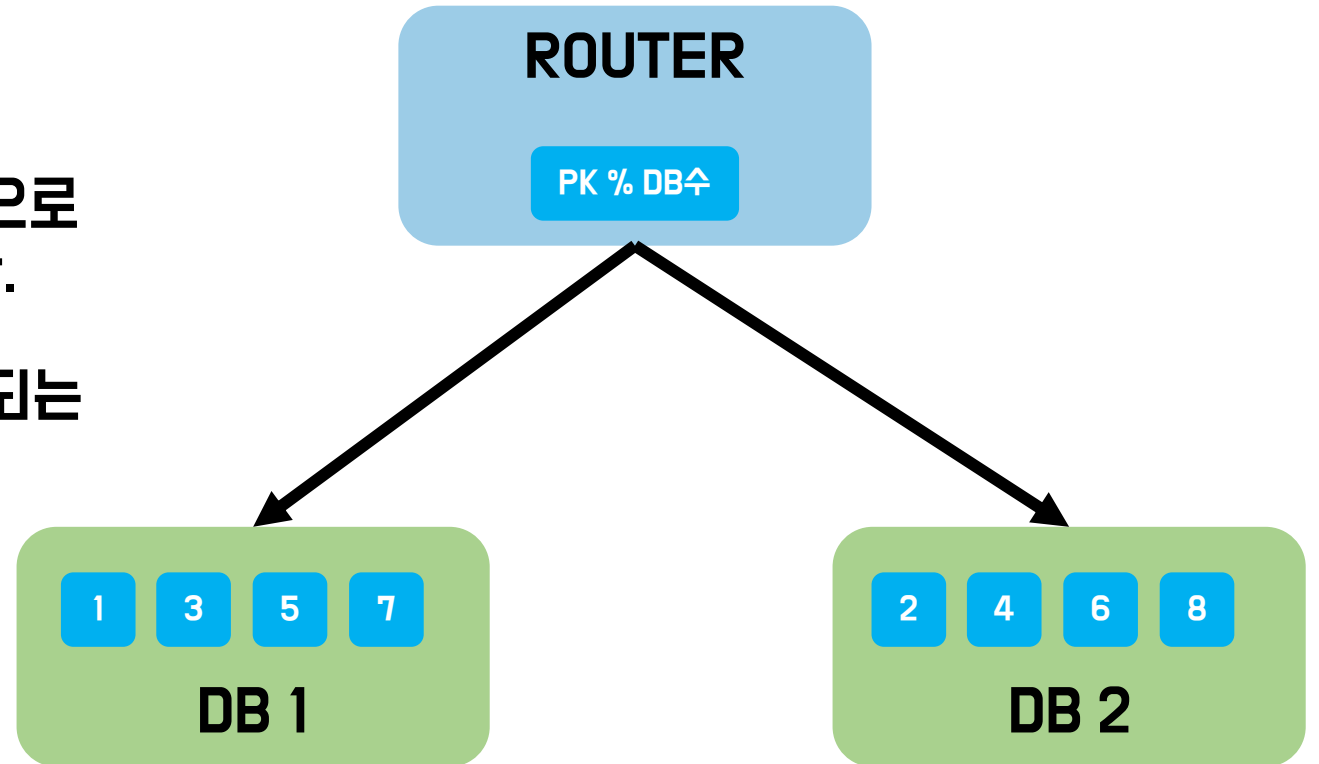
장점

데이터가 균일하게 분산된다. 트래픽을 안정적으로 소화하면서 DB리소스를 최대한 활용할 수 있다.

데이터량이 일정 수준에서 유지될 것으로 예상되는 곳에 어울린다.

단점

DB 증설 과정에서 이미 적재된 데이터의 재정렬이 필요하다.



5. 샤딩의 종류 - 모듈러 샤딩

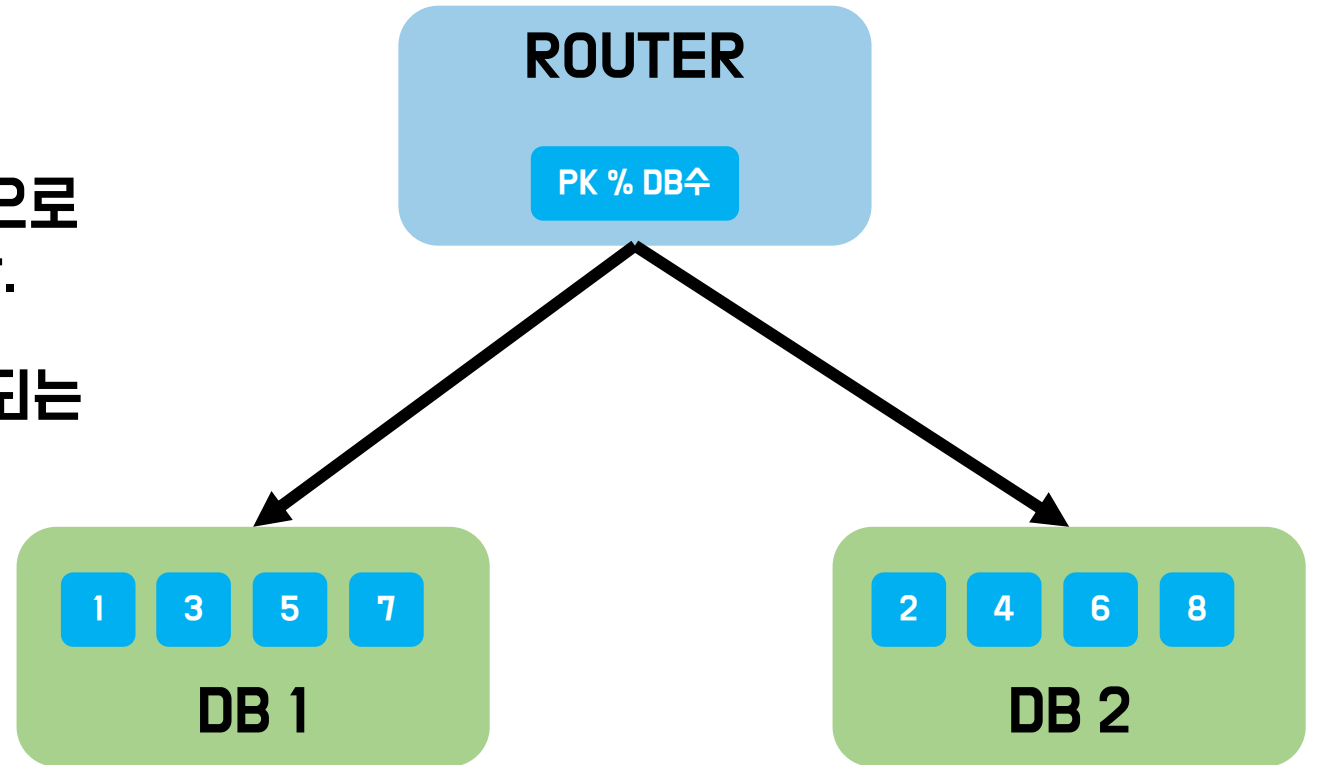
장점

데이터가 균일하게 분산된다. 트래픽을 안정적으로 소화하면서 DB리소스를 최대한 활용할 수 있다.

데이터량이 일정 수준에서 유지될 것으로 예상되는 곳에 어울린다.

단점

DB 증설 과정에서 이미 적재된 데이터의 재정렬이 필요하다.



5. 샤딩의 종류 - 모듈러 샤딩

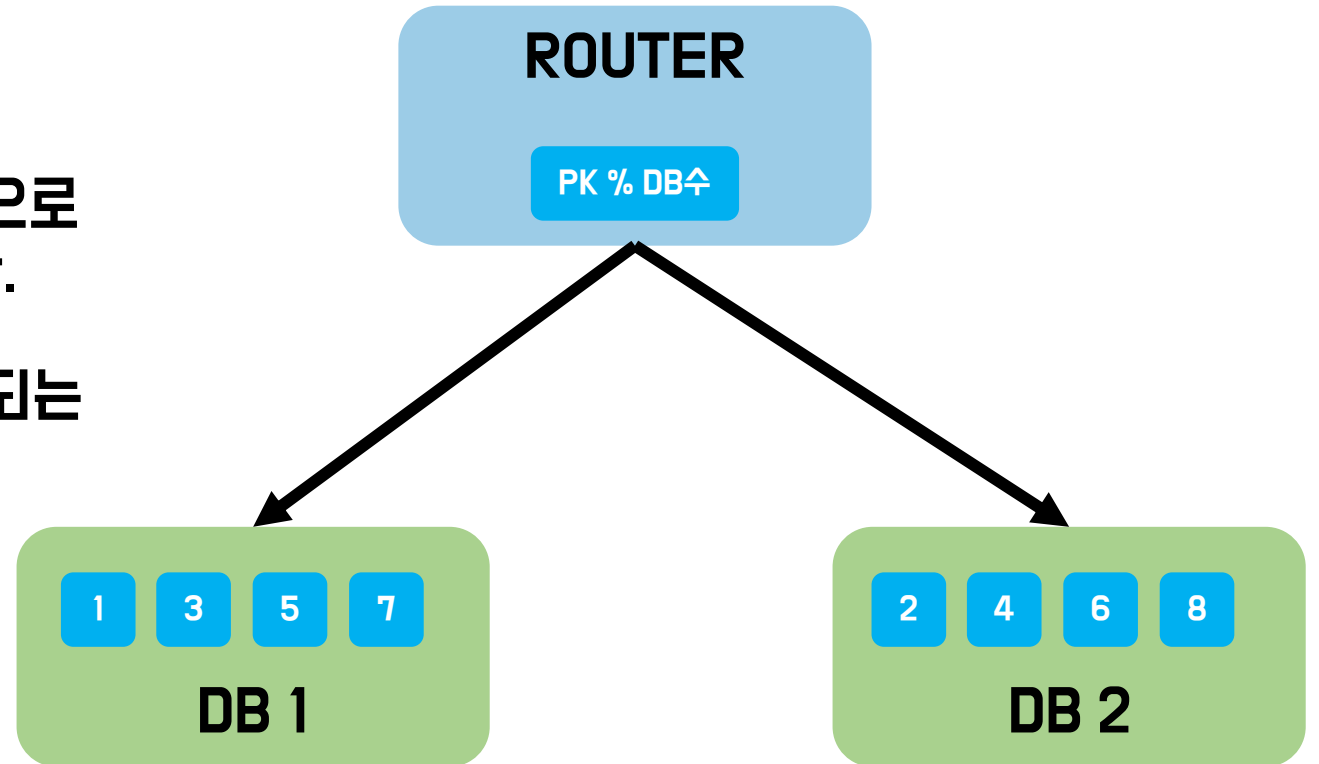
장점

데이터가 균일하게 분산된다. 트래픽을 안정적으로 소화하면서 DB리소스를 최대한 활용할 수 있다.

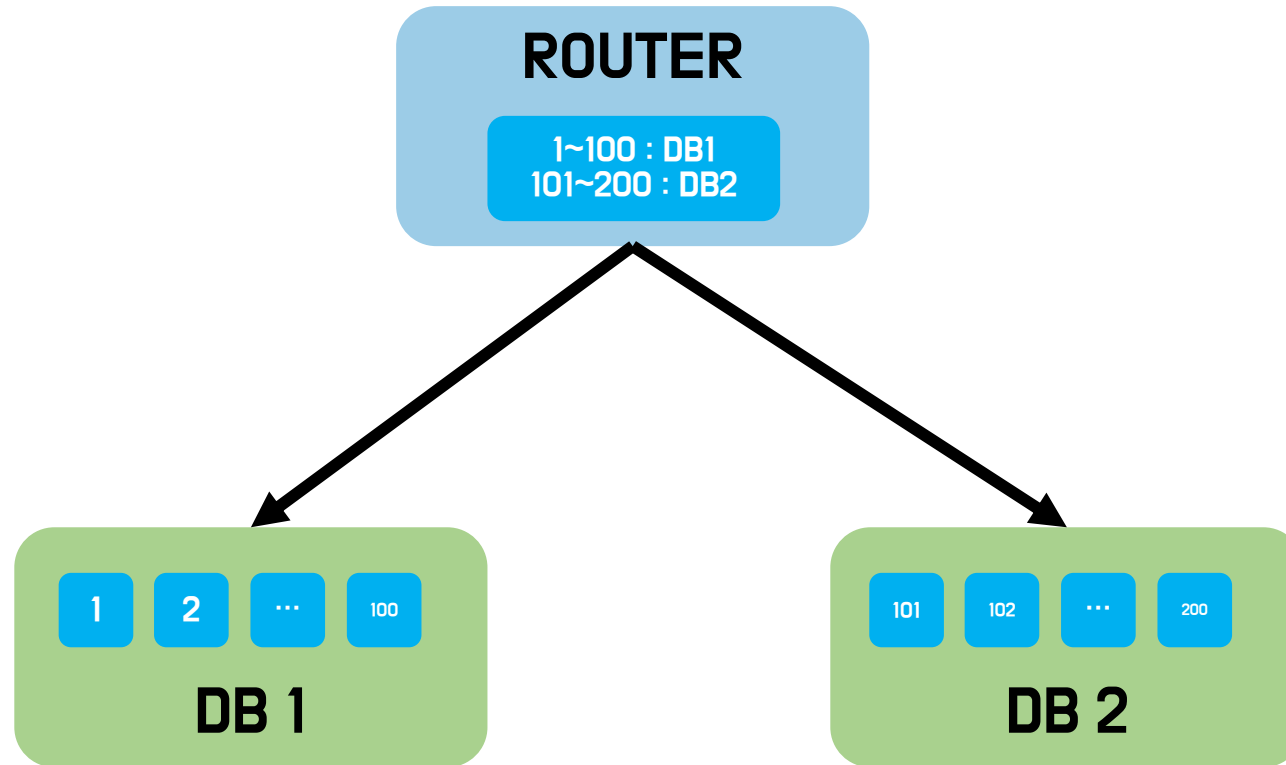
데이터량이 일정 수준에서 유지될 것으로 예상되는 곳에 어울린다.

단점

DB 증설 과정에서 이미 적재된 데이터의 재정렬이 필요하다.



5. 샤딩의 종류 - 레인지 샤딩



5. 샤딩의 종류 - 레인지 샤딩

장점

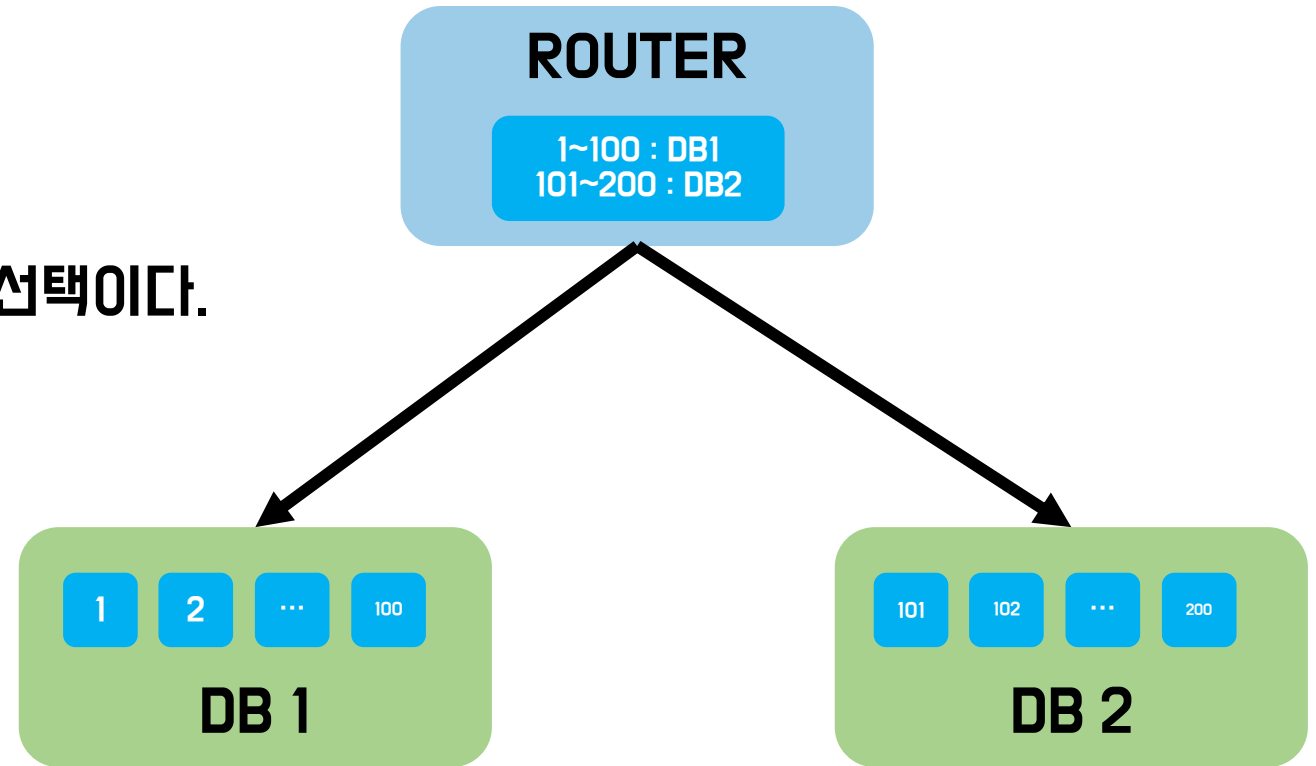
DB 증설에 재정렬 비용이 들지 않는다.

데이터가 급격히 증가할 여지가 있다면, 좋은 선택이다.

단점

일부 DB에 데이터가 몰릴 수 있다.

활성유저가 몰린 DB로 트래픽이나 데이터량이 몰릴 수 있다.
트래픽이 저조한 DB는 통합작업으로 유지비용을 아끼도록 관리해야 한다.



5. 샤딩의 종류 - 레인지 샤딩

장점

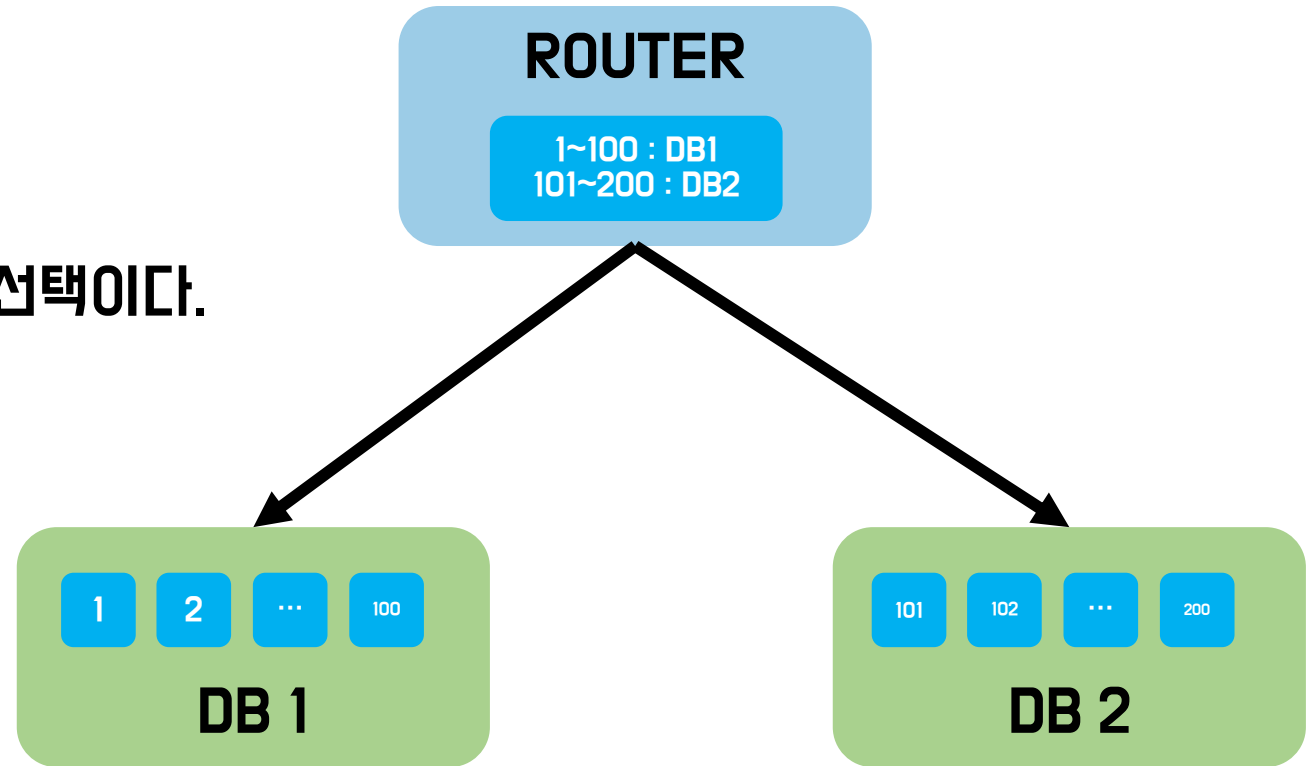
DB 증설에 재정렬 비용이 들지 않는다.

데이터가 급격히 증가할 여지가 있다면, 좋은 선택이다.

단점

일부 DB에 데이터가 몰릴 수 있다.

활성유저가 몰린 DB로 트래픽이나 데이터량이 몰릴 수 있다.
트래픽이 저조한 DB는 통합작업으로 유지비용을 아끼도록 관리해야 한다.



5. 샤딩의 종류 - 레인지 샤딩

장점

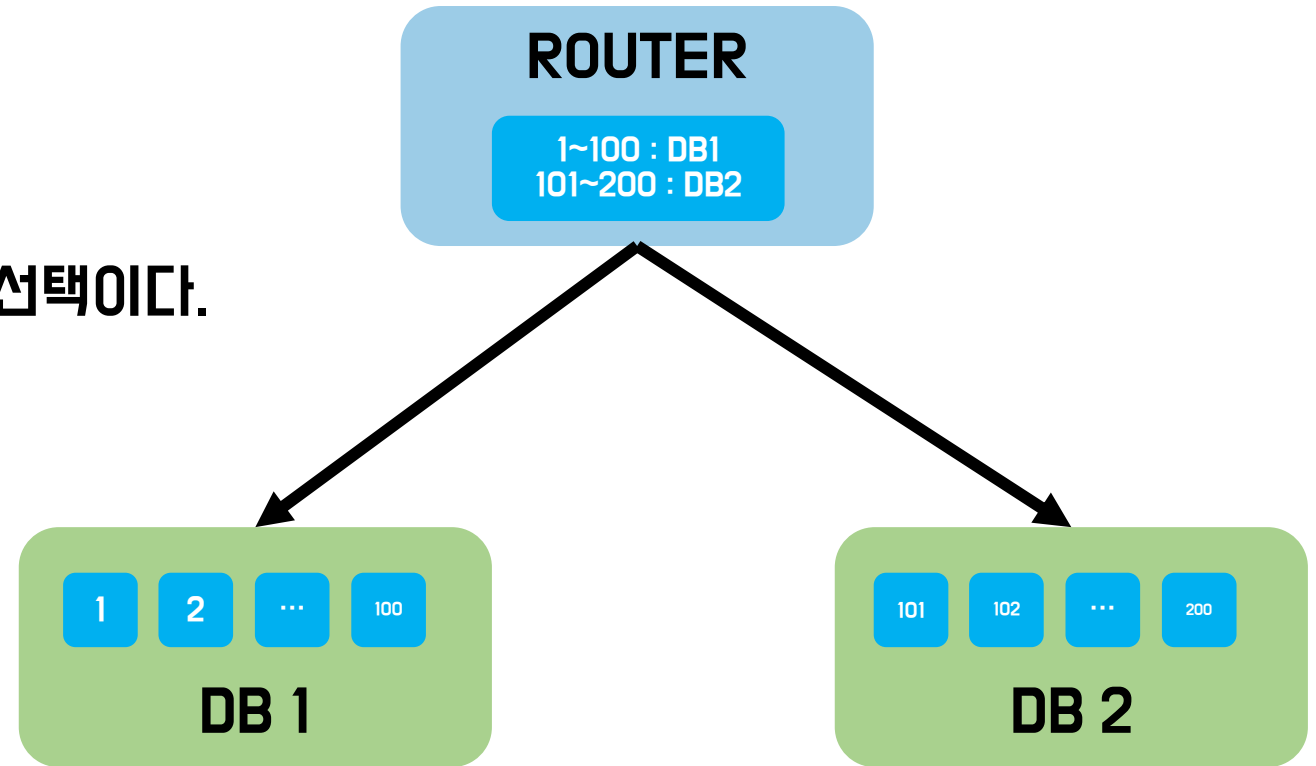
DB 증설에 재정렬 비용이 들지 않는다.

데이터가 급격히 증가할 여지가 있다면, 좋은 선택이다.

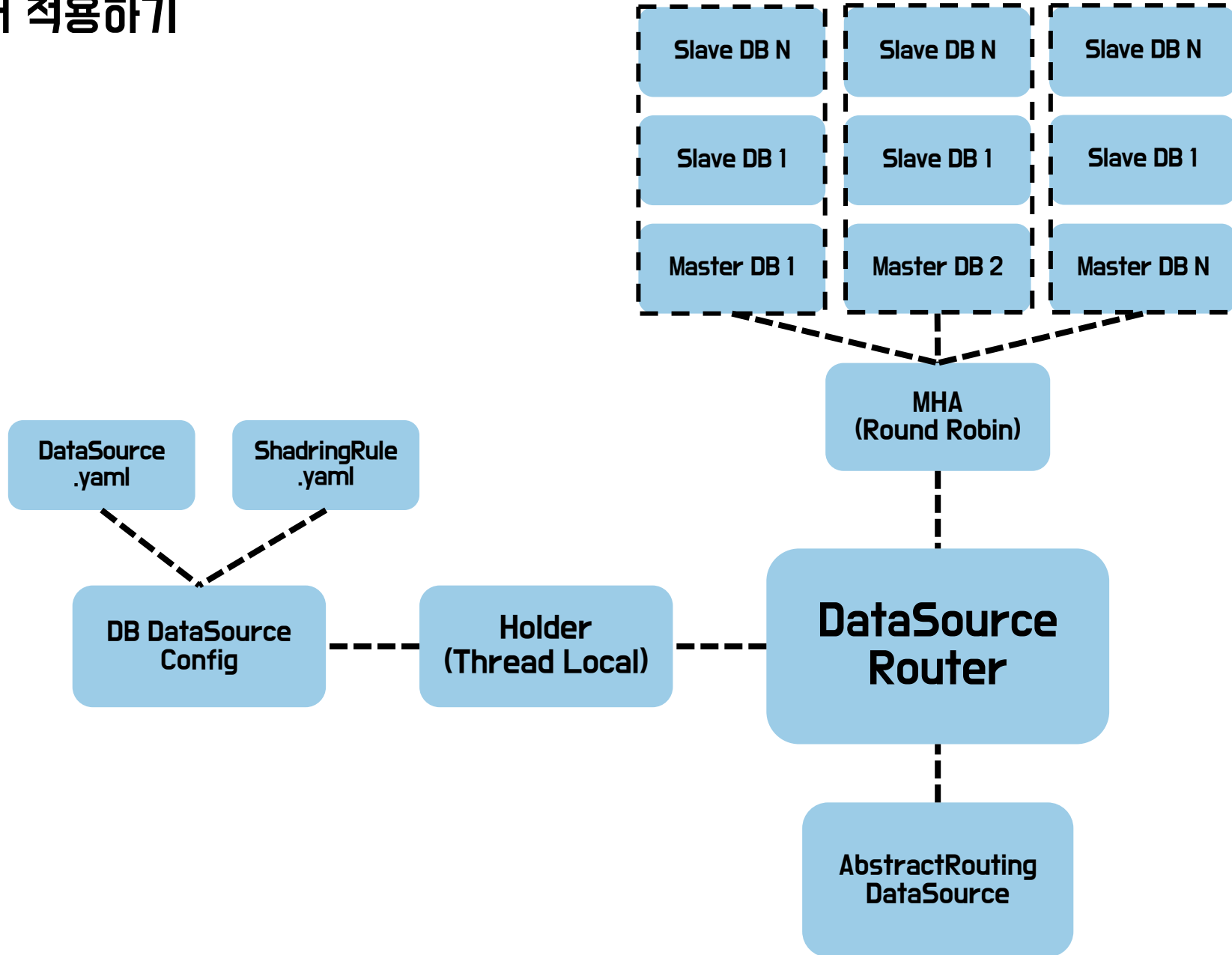
단점

일부 DB에 데이터가 몰릴 수 있다.

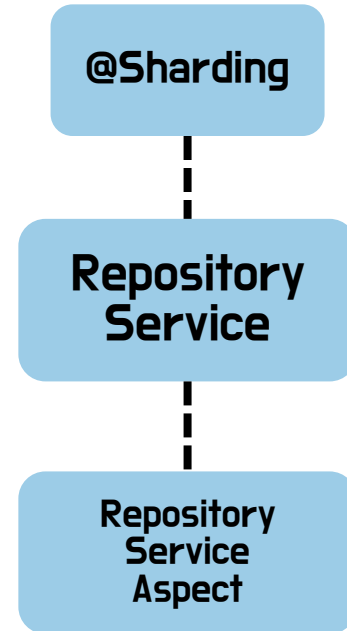
활성유저가 몰린 DB로 트래픽이나 데이터량이 몰릴 수 있다.
트래픽이 저조한 DB는 통합작업으로 유지비용을 아끼도록 관리해야 한다.



6. 스프링에서 적용하기



6. 스프링에서 적용하기



7. 참고 자료

1. <https://nesoy.github.io/articles/2018-05/Database-Shard>
2. <https://woowabros.github.io/experience/2020/07/06/db-sharding.html>
3. <https://engineering.linecorp.com/ko/blog/line-manga-database/>
4. <https://nesoy.github.io/articles/2018-02/Database-Partitioning>

감사합니다.