

7장 - 분산 시스템을 위한 유일 ID 생성

≡ Tags

다중 마스터 복제(multi-master replication)

- 데이터 베이스의 auto_increment
- 여러 데이터 센터에 걸쳐 규모를 늘리기가 어렵다.
- ID의 유일성은 보장되지만 그 값이 시간 흐름에 맞추어 커지도록 보장할 수 는 없다.
- 서버를 추가하거나 삭제할 때도 잘 동작하도록 만들기 어렵다.

UUID(Universally Unique Identifier)

- 128비트 짜리 수이다.
- UUID 값은 충돌 가능성이 지극히 낮다.
- 중복이 생길 확률을 50%로 끌어올리려면 초당 10억 개의 UUID를 100년동안 계속해서 만들어야 한다.

장점

- UUID를 만드는 것은 단순하다. 서버 사이의 조율이 필요없으므로 동기화 이슈도 없다.
- 각 서버가 자기가 쓸 ID를 알아서 만드는 구조이다. (규모 확장도 쉽다)

단점

- ID가 128비트로 길다.
 - 요구 사항이 작은 경우 부적합하다.
- ID를 시간순으로 정렬할 수 없다.

- ID에 숫자가 아닌 값이 포함될 수 있다.

티켓서버 (Ticket Server)

- 티켓 서버는 유일성 보장되는 ID를 만들어 내는 데 쓰일 수 있다.

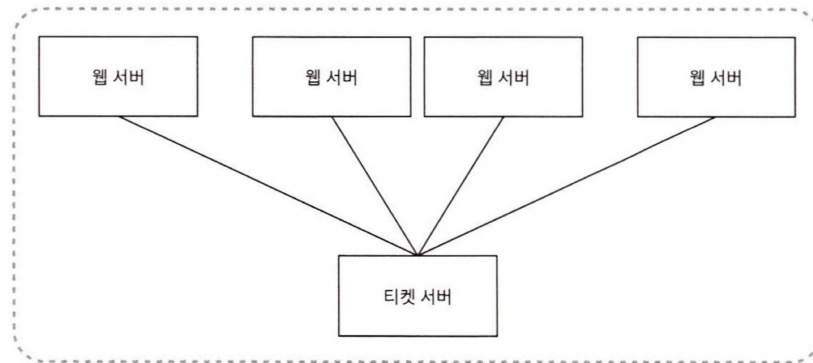


그림 7-4

- auto_increment 기능을 갖춘 데이터베이스 서버를 티켓서버로 사용하는 것이다.

장점

- 유일성이 보장되는 오직 숫자로만 구성된 ID를 쉽게 만들 수 있다.
- 구현하기 쉽고, 중소 규모 애플리케이션에 적합하다.

단점

- 티켓 서버가 SPOF(Single-Point-of-Failure)가 된다.
 - 이 서버가 장애가 발생된다면 서비스가 중지된다.

트위터 스노우플레이크(twitter snowflake)

- 스노우 플레이크 ID 생성 기법

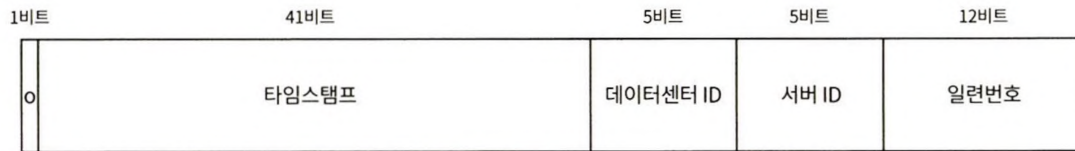


그림 7-5

- 사인(sign) 비트 : 1비트를 할당한다. 음수와 양수를 구별하는데 사용할 수 있다.
- 타임스탬프(timestamp): 41비트를 할당한다. 기원 시각(epoch) 이후 몇 밀리초 (millisecond)가 경과했는지 나타내는 값이다.
 - 타임스탬프 최댓값은 $2^{41}-1 = \text{약 } 69\text{년}$

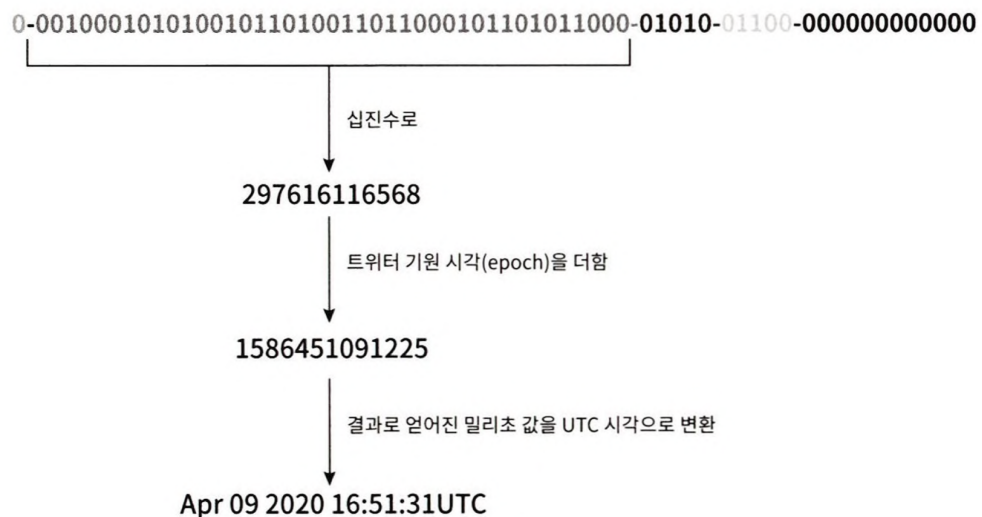


그림 7-7

- 데이터 센터 ID: 5비트를 할당한다. 따라서 $2^5 = 32$ 개 데이터센터를 지원할 수 있다.
- 서버 ID: 5비트를 할당한다. 따라서 데이터 센터를 32개 서버를 사용할 수 있다.
- 일련번호 : 12비트를 할당한다. 각 서버에서는 ID를 생성할 때마다 일련번호를 1만큼 증가시킨다.
 - 1밀리초가 경과할 때마다 0으로 초기화 시킨다

마무리

- 시계 동기화 (clock synchronization): 이번 설계를 진행하면서 우리는 ID 생성 서버들이 전부 같은 시계를 사용한다고 가정했다. 시계의 동기화가 문제가 생긴다면 ID생성에 문제가 될 수 있다.
 - NTP(Network Time Protocol)은 이문제를 해결하는 보편적인 수단이다.
- 각 절(section)의 길이 최적화: 예를 들어 동시성(concurrency)이 낮고 수명이 긴 애플리케이션이라면 일련번호 적을 줄이고 타임스탬프 절의 길이를 늘리는 것이 효과적이다
- 고가용성(high Availability): ID 생성기는 필수 불가결(mission critical) 컴포넌트이므로 아주 높은 가용성을 제공해야한다.