



(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2019-0017127
(43) 공개일자 2019년02월20일

- | | |
|---|---|
| (51) 국제특허분류(Int. Cl.)
G06F 16/00 (2019.01)
(52) CPC특허분류
G06F 16/214 (2019.01)
(21) 출원번호 10-2017-0101458
(22) 출원일자 2017년08월10일
심사청구일자 2017년08월10일 | (71) 출원인
네이버 주식회사
경기도 성남시 분당구 불정로 6, 그린팩토리 (정자동)
(72) 발명자
강철규
경기도 성남시 분당구 불정로 6(정자동, 그린팩토리)
강동완
경기도 성남시 분당구 불정로 6(정자동, 그린팩토리)
(뒷면에 계속)
(74) 대리인
양성보 |
|---|---|

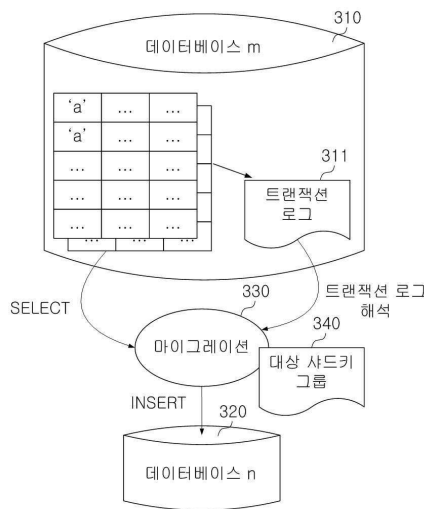
전체 청구항 수 : 총 12 항

(54) 발명의 명칭 **데이터베이스 샤딩 환경에서의 복제 로그 기반의 마이그레이션**

(57) 요약

데이터베이스 샤딩 환경에서의 복제 로그 기반의 마이그레이션 기술을 제공한다. 일실시예에 따른 마이그레이션 방법은 샤딩(sharding)된 데이터의 샤드키(shardkey)들을 분류하여 생성된 기설정된 수의 복수의 그룹들 각각의 그룹 식별자들과 트랜잭션 로그를 이용하여 데이터베이스간 마이그레이션을 처리할 수 있다.

대표도 - 도3



(72) 발명자

김갑영

경기도 성남시 분당구 불정로 6(정자동, 그린팩토리)

박부식

경기도 성남시 분당구 불정로 6(정자동, 그린팩토리)

서경식

경기도 성남시 분당구 불정로 6(정자동, 그린팩토리)

명세서

청구범위

청구항 1

데이터 마이그레이션(migration) 방법에 있어서,

샤딩(sharding)된 데이터의 샤드키(shardkey)들을 분류하여 생성된 기설정된 수의 복수의 그룹들 각각의 그룹 식별자들과 관련하여, 제1 데이터베이스의 그룹 식별자와 제2 데이터베이스의 그룹 식별자를 관리하는 단계;

상기 제1 데이터베이스에서 상기 제2 데이터베이스로 복제하고자 하는 데이터를 위한 샤드키들을 식별하는 단계;

상기 식별된 샤드키들에 대응하는 데이터를 상기 제1 데이터베이스에서 상기 제2 데이터베이스로 마이그레이션 하는 단계;

상기 마이그레이션 도중에, 상기 제1 데이터베이스에 반영된 트랜잭션을, 트랜잭션 로그를 이용하여 상기 제2 데이터베이스에 동시 반영하는 단계; 및

상기 데이터의 마이그레이션에 따라 상기 제1 데이터베이스 및 상기 제2 데이터베이스 각각에서 관리되는 그룹 식별자들을 변경하는 단계

를 포함하는 것을 특징으로 하는 데이터 마이그레이션 방법.

청구항 2

제1항에 있어서,

상기 그룹 식별자들을 변경하는 단계는,

상기 마이그레이션되는 데이터로의 접근 차단 시간을 최소화하기 위해, 상기 데이터의 마이그레이션이 종료되는 시점에 상기 그룹 식별자들을 변경하는 것을 특징으로 하는 마이그레이션 방법.

청구항 3

제1항에 있어서,

상기 반영하는 단계는,

상기 제1 데이터베이스에서 상기 제2 데이터베이스로 마이그레이션이 완료된 샤드키의 데이터에 대한 트랜잭션을 상기 트랜잭션 로그에 대한 분석을 통해 확인하여 상기 제2 데이터베이스에 반영하는 것을 특징으로 하는 마이그레이션 방법.

청구항 4

제1항에 있어서,

상기 복제하고자 하는 데이터를 위한 샤드키들을 식별하는 단계는,

그룹에 포함된 샤드키들과 상기 그룹의 그룹 식별자를 연계하여 저장하는 샤드키-그룹 식별자 테이블에서 상기 제1 데이터베이스가 관리하는 그룹 식별자를 통해 복제하고자 하는 데이터를 위한 샤드키들을 식별하는 것을 특징으로 하는 마이그레이션 방법.

청구항 5

제1항에 있어서,

상기 마이그레이션하는 단계는,

상기 식별된 샤드키를 이용하여 상기 제1 데이터베이스에서 복제하고자 하는 데이터를 선택(select)하고, 상

기 선택된 데이터를 상기 제2 데이터베이스로 insert하는 것을 특징으로 하는 마이그레이션 방법.

청구항 6

컴퓨터와 결합되어 제1항 내지 제5항 중 어느 한 항의 방법을 컴퓨터에 실행시키기 위해 컴퓨터 판독 가능한 기록매체에 저장된 컴퓨터 프로그램.

청구항 7

제1항 내지 제5항 중 어느 한 항의 방법을 컴퓨터에 실행시키기 위한 프로그램이 기록되어 있는 것을 특징으로 하는 컴퓨터에서 판독 가능한 기록매체.

청구항 8

컴퓨터 장치에 있어서,

컴퓨터에서 판독 가능한 명령을 실행하도록 구현되는 적어도 하나의 프로세서

를 포함하고,

상기 적어도 하나의 프로세서는,

샤딩(sharding)된 데이터의 샤드키(shardkey)들을 분류하여 생성된 기설정된 수의 복수의 그룹들 각각의 그룹 식별자들과 관련하여, 제1 데이터베이스의 그룹 식별자와 제2 데이터베이스의 그룹 식별자를 관리하고,

상기 제1 데이터베이스에서 상기 제2 데이터베이스로 복제하고자 하는 데이터를 위한 샤드키들을 식별하고,

상기 식별된 샤드키들에 대응하는 데이터를 상기 제1 데이터베이스에서 상기 제2 데이터베이스로 마이그레이션 하고,

상기 마이그레이션 도중에, 상기 제1 데이터베이스에 반영된 트랜잭션을, 트랜잭션 로그를 이용하여 상기 제2 데이터베이스에 동시 반영하고,

상기 데이터의 마이그레이션에 따라 상기 제1 데이터베이스 및 상기 제2 데이터베이스 각각에서 관리되는 그룹 식별자들을 변경하는 것

을 특징으로 하는 데이터 컴퓨터 장치.

청구항 9

제8항에 있어서,

상기 적어도 하나의 프로세서는,

상기 마이그레이션되는 데이터로의 접근 차단 시간을 최소화하기 위해, 상기 데이터의 마이그레이션이 종료되는 시점에 상기 그룹 식별자들을 변경하는 것을 특징으로 하는 컴퓨터 장치.

청구항 10

제8항에 있어서,

상기 적어도 하나의 프로세서는,

상기 제1 데이터베이스에서 상기 제2 데이터베이스로 마이그레이션이 완료된 샤드키의 데이터에 대한 트랜잭션을 상기 트랜잭션 로그에 대한 분석을 통해 확인하여 상기 제2 데이터베이스에 반영하는 것을 특징으로 하는 컴퓨터 장치.

청구항 11

제8항에 있어서,

상기 적어도 하나의 프로세서는,

그룹에 포함된 샤드키들과 상기 그룹의 그룹 식별자를 연계하여 저장하는 샤드키-그룹 식별자 테이블에서 상기 제1 데이터베이스가 관리하는 그룹 식별자를 통해 복제하고자 하는 데이터를 위한 샤드키들을 식별하는 것을 특

징으로 하는 컴퓨터 장치.

청구항 12

제8항에 있어서,

상기 적어도 하나의 프로세서는,

상기 식별된 샤드키를 이용하여 상기 제1 데이터베이스에서 복제하고자 하는 데이터를 선택(select)하고, 상기 선택된 데이터를 상기 제2 데이터베이스로 insert하는 것을 특징으로 하는 컴퓨터 장치.

발명의 설명

기술 분야

[0001] 아래의 설명은 데이터베이스 샤딩 환경에서의 복제 로그 기반의 마이그레이션 기술에 관한 것이다.

배경 기술

[0002] 데이터베이스들 간의 데이터 마이그레이션(migration)을 위한 기술들이 존재한다. 예를 들어, 한국등록특허 제 10-1748912호는 소스(source) 저장소의 데이터를 타겟(target) 저장소로 이전(migration)하는 기술을 개시하고 있다.

[0003] 이때, 제1 데이터베이스에서 제2 데이터베이스로 데이터를 복제하는 도중에도 트랜잭션이 발생할 수 있다. 종래기술에서는 이러한 트랜잭션을 소스 데이터베이스인 제1 데이터베이스에 먼저 반영한 후, 트랜잭션 로그를 제2 데이터베이스로 전달하며, 제2 데이터베이스는 트랜잭션 로그에 따라 트랜잭션들을 순차적으로 반영함으로써, 제1 데이터베이스와 제2 데이터베이스가 동기화될 수 있다.

[0004] 그러나, 제2 데이터베이스에서는 데이터의 마이그레이션이 완료된 이후에 트랜잭션 로그에 따른 트랜잭션들이 트랜잭션들의 순서에 따라 순차적으로 진행되기 때문에 복제 지연이 발생할 수 있으며, 마이그레이션을 위한 데이터로의 접근 차단 시간이 길어지게 되는 문제점이 있다.

발명의 내용

해결하려는 과제

[0005] 제1 데이터베이스와 제2 데이터베이스간의 데이터 마이그레이션(migration) 도중에 제1 데이터베이스에 반영되는 트랜잭션을 트랜잭션 로그를 이용하여 제2 데이터베이스에 동시 반영함으로써, 복제 지연 시간을 줄일 수 있으며, 마이그레이션되는 데이터로의 접근 차단 시간을 줄일 수 있는 데이터 마이그레이션 방법 및 데이터 마이그레이션 방법을 수행하는 컴퓨터 장치, 그리고 컴퓨터와 결합되어 본 발명의 실시예들에 따른 데이터 마이그레이션 방법을 컴퓨터에 실행시키기 위해 컴퓨터 판독 가능한 기록매체에 저장된 컴퓨터 프로그램과 그 기록매체를 제공한다.

[0006] 트랜잭션을 동시에 반영하면서 데이터의 마이그레이션이 종료되는 시점에 제1 데이터베이스와 제2 데이터베이스에서 각각 관리되는 그룹 식별자들을 변경함에 따라 마이그레이션되는 데이터로의 접근 차단 시간을 최소화할 수 있는 데이터 마이그레이션 방법 및 데이터 마이그레이션 방법을 수행하는 컴퓨터 장치, 그리고 컴퓨터와 결합되어 본 발명의 실시예들에 따른 데이터 마이그레이션 방법을 컴퓨터에 실행시키기 위해 컴퓨터 판독 가능한 기록매체에 저장된 컴퓨터 프로그램과 그 기록매체를 제공한다.

과제의 해결 수단

[0007] 데이터 마이그레이션(migration) 방법에 있어서, 샤딩(sharding)된 데이터의 샤드키(shardkey)들을 분류하여 생성된 기설정된 수의 복수의 그룹들 각각의 그룹 식별자들과 관련하여, 제1 데이터베이스의 그룹 식별자와 제2 데이터베이스의 그룹 식별자를 관리하는 단계; 상기 제1 데이터베이스에서 상기 제2 데이터베이스로 복제하고자 하는 데이터를 위한 샤드키들을 식별하는 단계; 상기 식별된 샤드키들에 대응하는 데이터를 상기 제1 데이터베이스에서 상기 제2 데이터베이스로 마이그레이션하는 단계; 상기 마이그레이션 도중에, 상기 제1 데이터베이스에 반영된 트랜잭션을, 트랜잭션 로그를 이용하여 상기 제2 데이터베이스에 동시 반영하는 단계; 및 상기 데이터의 마이그레이션에 따라 상기 제1 데이터베이스 및 상기 제2 데이터베이스 각각에서 관리되는 그룹 식별자들

을 변경하는 단계를 포함하는 것을 특징으로 하는 데이터 마이그레이션 방법을 제공한다.

- [0008] 상기 데이터 마이그레이션 방법을 컴퓨터에 실행시키기 위한 컴퓨터 프로그램이 기록되어 있는 것을 특징으로 하는 컴퓨터 판독 가능한 기록매체를 제공한다.
- [0009] 컴퓨터와 결합하여 상기 데이터 마이그레이션 방법을 컴퓨터에 실행시키기 위해 컴퓨터 판독 가능한 기록매체에 저장된 컴퓨터 프로그램을 제공한다.
- [0010] 컴퓨터 장치에 있어서, 컴퓨터에서 판독 가능한 명령을 실행하도록 구현되는 적어도 하나의 프로세서를 포함하고, 상기 적어도 하나의 프로세서는, 샤딩(sharding)된 데이터의 샤드키(shardkey)들을 분류하여 생성된 기설정된 수의 복수의 그룹들 각각의 그룹 식별자들과 관련하여, 제1 데이터베이스의 그룹 식별자와 제2 데이터베이스의 그룹 식별자를 관리하고, 상기 제1 데이터베이스에서 상기 제2 데이터베이스로 복제하고자 하는 데이터를 위한 샤드키들을 식별하고, 상기 식별된 샤드키들에 대응하는 데이터를 상기 제1 데이터베이스에서 상기 제2 데이터베이스로 마이그레이션하고, 상기 마이그레이션 도중에, 상기 제1 데이터베이스에 반영된 트랜잭션을, 트랜잭션 로그를 이용하여 상기 제2 데이터베이스에 동시 반영하고, 상기 데이터의 마이그레이션에 따라 상기 제1 데이터베이스 및 상기 제2 데이터베이스 각각에서 관리되는 그룹 식별자들을 변경하는 것을 특징으로 하는 데이터 컴퓨터 장치를 제공한다.

발명의 효과

- [0011] 제1 데이터베이스와 제2 데이터베이스간의 데이터 마이그레이션(migration) 도중에 제1 데이터베이스에 반영되는 트랜잭션을 트랜잭션 로그를 이용하여 제2 데이터베이스에 동시 반영함으로써, 복제 지연 시간을 줄일 수 있으며, 마이그레이션되는 데이터로의 접근 차단 시간을 줄일 수 있다.
- [0012] 트랜잭션을 동시에 반영하면서 데이터의 마이그레이션이 종료되는 시점에 제1 데이터베이스와 제2 데이터베이스에서 각각 관리되는 그룹 식별자들을 변경함에 따라 마이그레이션되는 데이터로의 접근 차단 시간을 최소화할 수 있다.

도면의 간단한 설명

- [0013] 도 1은 본 발명의 일실시예에 있어서, 데이터베이스에 저장된 데이터의 예를 도시한 도면이다.
- 도 2는 본 발명의 일실시예에 있어서, 데이터의 샤딩 예를 도시한 도면이다.
- 도 3은 본 발명의 일실시예에 있어서, 데이터 마이그레이션의 과정의 예를 도시한 도면이다.
- 도 4는 본 발명의 일실시예에 있어서, 그룹 식별자의 활용 예를 도시한 도면이다.
- 도 5는 본 발명의 일실시예에 있어서, 컴퓨터 장치의 내부 구성을 설명하기 위한 블록도이다.
- 도 6은 본 발명의 일실시예에 있어서, 데이터 마이그레이션 방법의 예를 도시한 흐름도이다.
- 도 7 및 도 8은 본 발명의 일실시예에 따른 데이터 마이그레이션 방법이 적용되는 과정의 예를 도시한 도면들이다.

발명을 실시하기 위한 구체적인 내용

- [0014] 이하, 실시예를 첨부한 도면을 참조하여 상세히 설명한다.
- [0015] 본 발명의 실시예들에 따른 데이터 마이그레이션(migration) 방법은 컴퓨터 장치에 의해 수행될 수 있다. 일례로, 컴퓨터 장치에는 컴퓨터 장치가 데이터 마이그레이션 방법을 수행하도록 제어하기 위한 컴퓨터 프로그램이 설치 및 구동될 수 있으며, 컴퓨터 프로그램은 컴퓨터 장치와 결합되어 데이터 마이그레이션 방법을 컴퓨터 장치에 실행시키기 위해 컴퓨터 판독 가능한 기록매체에 저장될 수 있다.
- [0016] 도 1은 본 발명의 일실시예에 있어서, 데이터베이스에 저장된 데이터의 예를 도시한 도면이고, 도 2는 본 발명의 일실시예에 있어서, 데이터의 샤딩 예를 도시한 도면이다. 도 1 및 도 2는 데이터베이스(110)에 저장된 데이터에 대한 데이터베이스 샤딩(database sharding)에 의해, 데이터들이 n 개의 샤드 노드들(210, 220 및 230)로 분산된 예를 나타내고 있다. 예를 들어, 도 1 및 도 2에서는 레코드들을 식별하기 위한 식별자인 'id1', 'id2', 'id3', 'id4'를 샤드키(shardkey)로 이용하여 동일한 샤드키의 레코드들을 모아서 n 개의 샤드들(210, 220 및 230)로 분산한 예를 나타내고 있다. 이때, 샤드키 단위의 데이터는 동일한 샤드키를 통해 식별되는 데

이터를 의미할 수 있다. 예를 들어, 도 2에서는 샤드키 'id2'를 통해 식별되는 데이터를 굵은 실선(240)을 통해 표시하고 있다.

- [0017] 샤드키 단위의 데이터는 용량과 QPS(Query Per Second)는 상대적으로 작은 편이나, 샤드키의 개수가 매우 많고, 따라서 결과적으로 데이터베이스 전체 용량과 QPS는 매우 크다는 특징을 갖는다. 또한, 샤드키를 통해 식별되는 데이터간의 관계(relation)은 의미가 있지만, 서로 다른 샤드키를 통해 식별되는 데이터간의 관계는 의미가 없다는 특징을 갖는다. 다시 말해, 하나의 샤드키를 통해 식별되는 데이터는 다른 샤드키를 통해 식별되는 데이터에 대해 의존성을 갖지 않으며, 샤드키 단위로 독립적인 개별 연산(트랜잭션)의 처리가 가능하다.
- [0018] n 개의 샤드 노드들(210, 220 및 230) 각각은 제1 샤드 노드(210)을 통해 나타나는 바와 같이, 하나의 샤드키에 대응하는 데이터를 포함할 수도 있으나, 제2 샤드 노드(220)을 통해 나타나는 바와 같이, 다수의 샤드키에 대응하는 데이터를 포함할 수도 있다.
- [0019] 이때, 본 발명의 실시예들에서는 앞서 설명한 샤드키 단위의 데이터와 관련하여 샤드키들을 기설정된 수의 복수의 그룹으로 분류하고, 복수의 그룹들의 그룹 식별자들을 통해 데이터에 대한 정합성 검사를 처리할 수 있다. 이때, 각각의 데이터베이스 서버들은 자신이 관리하는 샤드키가 분류된 그룹의 그룹 식별자들을 관리할 수 있다.
- [0020] 도 3은 본 발명의 일실시예에 있어서, 데이터 마이그레이션의 과정의 예를 도시한 도면이다. 도 3은 데이터베이스 m(310)의 데이터를 데이터베이스 n(320)으로 마이그레이션(330)하는 예를 나타내고 있다.
- [0021] 이때, 마이그레이션(330) 도중에 데이터베이스 m(310)에 반영되는 트랜잭션은 트랜잭션 로그(311)를 이용하여 데이터베이스 n(320)에 동시 반영될 수 있다. 예를 들어, 마이그레이션(330) 도중에 제1 연산이 데이터베이스 m(310)의 특정 데이터에 대해 반영되는 경우, 제1 연산에 대한 트랜잭션 로그를 분석하여 제1 연산이 제2 데이터베이스에도 동시에 반영될 수 있다. 따라서, 복제 지연 시간을 줄일 수 있으며, 해당 데이터에 대한 접근 차단 시간을 줄일 수 있게 된다.
- [0022] 이러한 트랜잭션의 동시 반영을 가능하게 하기 위한 기술들에 대해 보다 자세히 설명한다.
- [0023] 이미 설명한 바와 같이, 샤드키 단위의 데이터는 독립적으로 연산 처리가 가능하다. 따라서, 샤드키 단위로 데이터를 마이그레이션하는 경우, 이미 마이그레이션된 샤드키 단위의 데이터에 대해서는 트랜잭션을 동시 반영할 수 있게 된다. 예를 들어, 데이터베이스 m(310)에서 샤드키 'a'를 통해 식별되는 데이터가 데이터베이스 n(320)으로 마이그레이션된 경우를 고려할 수 있다. 이때, 샤드키 'a'와 연관된 제1 연산이 요청되는 경우, 데이터베이스 m(310)에 샤드키 'a'에 대응하는 데이터에 제1 연산을 수행할 수 있으며, 트랜잭션 로그의 분석을 통해 데이터베이스 n(320)에도 샤드키 'a'에 대응하는 마이그레이션된 데이터에 제1 연산을 수행할 수 있다.
- [0024] 이를 위해, 본 발명의 실시예들에 따른 컴퓨터 장치는 샤드키 단위로 데이터베이스 m(310)에 저장된 데이터를 선택(select)하고, 선택된 데이터를 데이터베이스 n(320)에 insert할 수 있다.
- [0025] 한편, 본 발명의 실시예들에서는 정합성 검사를 위해 그룹 식별자가 활용될 수 있다.
- [0026] 도 4는 본 발명의 일실시예에 있어서, 그룹 식별자의 활용 예를 도시한 도면이다. 본 실시예들에 따른 데이터베이스 서버들 각각은 삽입(INSERT), 갱신(UPDATE), 삭제(DELETE)와 같은 모든 DML(Data Manipulation Language) 질의 수행 시 샤딩 데이터에 대한 정합성 검사를 수행할 수 있다.
- [0027] 도 4의 실시예에서는 입력된 쿼리문(410)에서 JDBC(Java DataBase Connectivity, 420)가 샤드키 'id1'을 추출하고, 샤드키 'id1'을 해싱하여 해시값 '3'을 획득하는 예를 나타내고 있다. 이때, 해시값 '3'은 샤드키 'id1'이 포함된 그룹의 그룹 식별자가 될 수 있다. 예를 들어, 모든 샤드키들 각각이 기설정된 해시 함수의 파라미터로 이용되어 모든 샤드키들 각각을 위한 그룹 식별자들이 계산될 수 있다. 보다 구체적인 예로, 100만 개의 샤드키들이 해시 함수를 거쳐 1만 개의 그룹 식별자들이 계산될 수 있고, 그룹 식별자 각각에 대한 1만개의 그룹으로 분류될 수 있다.
- [0028] 이 경우, 새로운 샤드키가 추가된다 하더라도 해시 함수를 통해 해시되는 과정에서 1만개의 그룹 중 하나로 분류되기 때문에 그룹이나 그룹 식별자는 추가되지 않는다. 따라서, 각각의 데이터베이스 서버들은 100만개의 샤드키들 중 자신이 관리하는 샤드키들을 관리하는 것이 아니라, 1만개의 그룹 식별자들 중 자신이 관리하는 그룹 식별자만을 관리함으로써, 특정 샤드키가 자신이 관리하는 그룹에 포함되어 있는지 여부를 확인할 수 있다.
- [0029] 이때, 데이터베이스 서버들간의 데이터 이동은 그룹 식별자 또는 샤드키에 대응하는 데이터 단위로 이루어질 수

있다. 예를 들어, 하나의 그룹 식별자에 대응하는 데이터를 이동시킴에 있어서, 해당 그룹 식별자의 그룹에 포함된 샤드키의 단위로 데이터들이 선택 및 인서트될 수 있다. 예를 들어, 데이터베이스 서버 a이 그룹 식별자 1에 대한 데이터를 저장할 때, 데이터베이스 서버 a에서 데이터베이스 서버 b로의 데이터의 이동은 그룹 식별자 1에 대응하는 데이터의 단위로 이루어질 수 있으며, 이때, 그룹 식별자 1에 대응하는 그룹의 샤드키들의 단위로 데이터들이 선택 및 인서트될 수 있다.

[0030] 한편, 도 3에 도시된 바와 같이 데이터베이스 서버(430)는 그룹 식별자들 각각에 대해 자신이 관리하는 데이터에 대응하는 그룹 식별자와 자신이 관리하지 않는 데이터에 대응하는 그룹 식별자를 나타내기 위한 그룹 식별자 비트맵(440)을 관리할 수 있다. 예를 들어, 도 4에서 데이터베이스 서버(430)는 그룹 식별자 1, 3, n에 대응하는 데이터를 데이터베이스 j(450)에 저장하여 관리하며(그룹 식별자 1, 3, n의 값이 1로 설정됨), 그룹 식별자 2에 대응하는 데이터는 관리하지 않음(그룹 식별자 2의 값이 0으로 설정됨)을 알 수 있다.

[0031] 이때, 앞서 설명한 바와 같이, JDBC(420)는 쿼리문(410)에서 샤드키 'id1'를 추출하고, 추출된 샤드키 'id1'를 해시 함수를 통해 해싱하여 그룹 식별자 '3'을 추출할 수 있다. 이때, 추출된 그룹 식별자 '3'은 데이터베이스 서버(430)로 전달될 수 있고, 데이터베이스 서버(430)은 추출된 그룹 식별자 '3'을 그룹 식별자 비트맵(440)에서 검색하여 그룹 식별자 '3'에 대응하는 데이터가 데이터베이스 j(450)에 저장되어 데이터베이스 서버(430)에 의해 관리됨을 확인할 수 있다. 다시 말해, 데이터베이스 서버(430)는 요구된 연산과 연계된 데이터의 정합성을 검사할 수 있게 된다. 만약, 추출되어 수신된 그룹 식별자가 '2'라면, 데이터베이스 서버(430)는 잘못된 데이터의 변경 요구가 입력되었음을 파악할 수 있게 된다.

[0032] 이러한 그룹 식별자는 데이터의 정합성을 검사하기 위해 활용되며, 샤드키와 대응하는 그룹 식별자가 서로 연계되어 관리되는 샤드키-그룹 식별자 테이블이 활용될 수 있다. 예를 들어, 도 3을 통해 설명한 컴퓨터 장치는 샤드키-그룹 식별자 테이블에서 이동할 그룹 식별자에 대응하는 샤드키들을 식별할 수 있으며, 샤드키별로 각 테이블에서 선택(SELECT)과 인서트(INSERT)를 수행함으로써 샤드키 단위의 데이터 이동을 처리할 수 있다. 이러한 샤드키 단위의 데이터의 이동은 적어도 하나의 그룹 식별자에 대응하는 데이터에 대해 수행될 수 있다. 다시 말해, 데이터의 이동은 적어도 하나의 그룹 식별자에 대응하는 데이터들의 이동이 될 수 있으며, 이때, 하나의 그룹 식별자에 대응하는 데이터들은 샤드키 단위로 선택(SELECT)과 인서트(INSERT)를 반복하면서 이루어질 수 있다.

[0033] 이때, 그룹 식별자는 마이그레이션이 종료되는 시점에 변경될 수 있다. 앞서 도 4를 통해 설명한 바와 같이 그룹 식별자는 데이터의 정합성을 위해 이용된다. 반면, 도 3에서는 이미 설명한 바와 같이 이미 이전된 샤드키 단위의 데이터에 대해 데이터베이스 m(310)에 먼저 트랜잭션을 적용한다. 이 경우, 데이터베이스 m(310)에서 데이터의 마이그레이션에 따라 바로 특정 그룹 식별자를 변경해버리면, 해당 그룹 식별자에 대한 연산은 데이터베이스 m(310)과 관련하여 정합성 측면의 오류가 발생하게 된다. 예를 들어, 데이터베이스 m(310)이 그룹 식별자 1의 데이터를 관리(일례로, 그룹 식별자 비트맵에서 그룹 식별자 1에 대응하는 값이 '1'로 설정된 상태)하고 있었다고 가정한다. 또한, 도 3에서 데이터베이스 m(310)이 그룹 식별자 1의 데이터를 데이터베이스 n(320)으로 마이그레이션한 직후, 그룹 식별자 비트맵에서 그룹 식별자 1에 대응하는 값을 '0'으로 설정하였다고 가정한다.

[0034] 이 경우, 그룹 식별자 1과 연관된 연산(실제로는 그룹 식별자 1의 그룹에 포함된 샤드키와 연관된 연산)은 데이터베이스 m(310)에 적용될 수 없기 때문에(정합성 오류), 데이터베이스 m(310)과 데이터베이스 n(320)의 데이터는 동기화될 수 없게 된다. 이 경우 데이터의 동기화를 위해서는 마이그레이션되는 데이터로의 접근을 차단해야만 하고, 따라서 이용자의 SQL 진입을 막는 구간이 길어지게 된다.

[0035] 이에, 도 3의 실시예에서는 데이터의 마이그레이션이 종료되는 시점에 데이터베이스 m(310)과 데이터베이스 n(320) 각각에서 관리되는 그룹 식별자들을 변경함으로써, 마이그레이션되는 데이터로의 접근 차단 시간을 최소화할 수 있다.

[0036] 도 5는 본 발명의 일실시예에 있어서, 컴퓨터 장치의 내부 구성을 설명하기 위한 블록도이다. 컴퓨터 장치(500)는 메모리(510), 프로세서(520), 통신 인터페이스(530) 그리고 입출력 인터페이스(540)를 포함할 수 있다. 메모리(510)는 컴퓨터에서 판독 가능한 기록매체로서, RAM(random access memory), ROM(read only memory) 및 디스크 드라이브와 같은 비소멸성 대용량 기록장치(permanent mass storage device)를 포함할 수 있다. 여기서 ROM과 디스크 드라이브와 같은 비소멸성 대용량 기록장치는 메모리(510)와는 구분되는 별도의 영구 저장 장치로서 컴퓨터 장치(500)에 포함될 수도 있다. 또한, 메모리(510)에는 운영체제와 적어도 하나의 프로그램 코드가 저장될 수 있다. 이러한 소프트웨어 구성요소들은 메모리(510)와는 별도의 컴퓨터에서 판독 가능한 기록매체로

부터 메모리(510)로 로딩될 수 있다. 이러한 별도의 컴퓨터에서 판독 가능한 기록매체는 플로피 드라이브, 디스크, 테이프, DVD/CD-ROM 드라이브, 메모리 카드 등의 컴퓨터에서 판독 가능한 기록매체를 포함할 수 있다. 다른 실시예에서 소프트웨어 구성요소들은 컴퓨터에서 판독 가능한 기록매체가 아닌 통신 인터페이스(530)를 통해 메모리(510)에 로딩될 수도 있다. 예를 들어, 소프트웨어 구성요소들은 네트워크(560)를 통해 수신되는 파일들에 의해 설치되는 컴퓨터 프로그램에 기반하여 컴퓨터 장치(500)의 메모리(510)에 로딩될 수 있다.

[0037] 프로세서(520)는 기본적인 산술, 로직 및 입출력 연산을 수행함으로써, 컴퓨터 프로그램의 명령을 처리하도록 구성될 수 있다. 명령은 메모리(510) 또는 통신 인터페이스(530)에 의해 프로세서(520)로 제공될 수 있다. 예를 들어 프로세서(520)는 메모리(510)와 같은 기록 장치에 저장된 프로그램 코드에 따라 수신되는 명령을 실행하도록 구성될 수 있다.

[0038] 통신 인터페이스(530)은 네트워크(560)를 통해 컴퓨터 장치(500)가 다른 장치(일례로, 앞서 설명한 저장 장치들)와 서로 통신하기 위한 기능을 제공할 수 있다. 일례로, 컴퓨터 장치(500)의 프로세서(520)가 메모리(510)와 같은 기록 장치에 저장된 프로그램 코드에 따라 생성한 요청이나 명령, 데이터, 파일 등이 통신 인터페이스(530)의 제어에 따라 네트워크(560)를 통해 다른 장치들로 전달될 수 있다. 역으로, 다른 장치로부터의 신호나 명령, 데이터, 파일 등이 네트워크(560)를 거쳐 컴퓨터 장치(500)의 통신 인터페이스(530)를 통해 컴퓨터 장치(500)로 수신될 수 있다. 통신 인터페이스(530)를 통해 수신된 신호나 명령, 데이터 등은 프로세서(520)나 메모리(510)로 전달될 수 있고, 파일 등은 컴퓨터 장치(500)가 더 포함할 수 있는 저장 매체(상술한 영구 저장 장치)로 저장될 수 있다.

[0039] 입출력 인터페이스(540)는 입출력 장치(550)와의 인터페이스를 위한 수단일 수 있다. 예를 들어, 입력 장치는 마이크, 키보드 또는 마우스 등의 장치를, 그리고 출력 장치는 디스플레이, 스피커와 같은 장치를 포함할 수 있다. 다른 예로 입출력 인터페이스(540)는 터치스크린과 같이 입력과 출력을 위한 기능이 하나로 통합된 장치와의 인터페이스를 위한 수단일 수도 있다. 입출력 장치(550)는 컴퓨터 장치(500)와 하나의 장치로 구성될 수도 있다.

[0040] 또한, 다른 실시예들에서 컴퓨터 장치(500)는 도 5의 구성요소들보다 더 적은 혹은 더 많은 구성요소들을 포함할 수도 있다. 그러나, 대부분의 종래기술적 구성요소들을 명확하게 도시할 필요성은 없다. 예를 들어, 컴퓨터 장치(500)는 상술한 입출력 장치(550) 중 적어도 일부를 포함하도록 구현되거나 또는 트랜시버(transceiver), GPS(Global Positioning System) 모듈, 카메라, 각종 센서, 데이터베이스 등과 같은 다른 구성요소들을 더 포함할 수도 있다.

[0041] 이러한 컴퓨터 장치(500)는 일례로, 도 3 및 도 4를 통해 설명한 데이터베이스 서버들(330, 410, 420)에 대응할 수 있다.

[0042] 도 6은 본 발명의 일 실시예에 있어서, 데이터 마이그레이션 방법의 예를 도시한 흐름도이다. 본 실시예에 따른 데이터 마이그레이션 방법은 앞서 설명한 컴퓨터 장치(500)에 의해 수행될 수 있다. 예를 들어, 컴퓨터 장치(500)의 프로세서(520)는 메모리(510)에 로딩된 운영체제의 코드나 적어도 하나의 프로그램의 코드에 따른 제어 명령(instruction)을 실행하도록 구현될 수 있다. 여기서, 프로세서(520)는 이러한 제어 명령에 따라 컴퓨터 장치(500)가 도 6의 데이터 마이그레이션 방법이 포함하는 단계들(610 및 650)을 수행하도록 컴퓨터 장치(500)를 제어할 수 있다.

[0043] 단계(610)에서 컴퓨터 장치(500)는 샤딩(sharding)된 데이터의 샤드키(shardkey)들을 분류하여 생성된 기설정된 수의 복수의 그룹들 각각의 그룹 식별자들과 관련하여, 제1 데이터베이스의 그룹 식별자와 제2 데이터베이스의 그룹 식별자를 관리할 수 있다.

[0044] 단계(620)에서 컴퓨터 장치(500)는 제1 데이터베이스에서 제2 데이터베이스로 복제하고자 하는 데이터를 위한 샤드키들을 식별할 수 있다. 예를 들어, 컴퓨터 장치(500)는 그룹에 포함된 샤드키들과 그룹의 그룹 식별자를 연계하여 저장하는 샤드키-그룹 식별자 테이블에서 제1 데이터베이스가 관리하는 그룹 식별자를 통해 복제하고자 하는 데이터를 위한 샤드키들을 식별할 수 있다. 도 4를 통해 설명한 바와 같이 각각의 데이터베이스를 위한 그룹 식별자들이 관리될 수 있으며, 이러한 그룹 식별자들과 샤드키-그룹 식별자 테이블을 통해 마이그레이션되어야 할 데이터의 샤드키들이 식별될 수 있다.

[0045] 단계(630)에서 컴퓨터 장치(500)는 식별된 샤드키들에 대응하는 데이터를 제1 데이터베이스에서 제2 데이터베이스로 마이그레이션할 수 있다. 이미 설명한 바와 같이 컴퓨터 장치(500)는 식별된 샤드키를 이용하여 제1 데이터베이스에서 복제하고자 하는 데이터를 선택(select)하고, 선택된 데이터를 제2 데이터베이스로 insert

(insert)하는 방식을 통해 샤드키 단위로 데이터를 마이그레이션할 수 있다.

- [0046] 단계(640)에서 컴퓨터 장치(500)는 마이그레이션 도중에, 제1 데이터베이스에 반영된 트랜잭션을, 트랜잭션 로그를 이용하여 제2 데이터베이스에 동시 반영할 수 있다. 이때, 컴퓨터 장치(500)는 상기 제1 데이터베이스에서 상기 제2 데이터베이스로 마이그레이션이 완료된 샤드키의 데이터에 대한 트랜잭션을 상기 트랜잭션 로그에 대한 분석을 통해 확인하여 상기 제2 데이터베이스에 반영할 수 있다. 이미 설명한 바와 같이, 샤드키 단위의 데이터는 독립적인 연산이 가능하기 때문에 마이그레이션이 완료된 샤드키의 데이터에 대해서는 제1 데이터베이스와 제2 데이터베이스 모두에 트랜잭션의 적용이 가능하다. 이를 통해 마이그레이션이 완료된 샤드키의 데이터에 대한 접근을 차단하지 않고, 마이그레이션 도중에 연산을 바로 제1 데이터베이스와 제2 데이터베이스의 데이터 모두에 수행하여 데이터를 동기화시킬 수 있게 된다.
- [0047] 단계(650)에서 컴퓨터 장치(500)는 데이터의 마이그레이션에 따라 제1 데이터베이스 및 제2 데이터베이스 각각에서 관리되는 그룹 식별자들을 변경할 수 있다. 앞서 설명한 바와 같이, 컴퓨터 장치(500)는 마이그레이션되는 데이터로의 접근 차단 시간을 최소화하기 위해, 데이터의 마이그레이션이 종료되는 시점에 제1 데이터베이스와 제2 데이터베이스의 그룹 식별자들을 변경할 수 있다.
- [0048] 예를 들어, 컴퓨터 장치(500)는 소스 데이터베이스인 제1 데이터베이스와 타겟 데이터베이스인 제2 데이터베이스에서 그룹 식별자 비트맵(일레로, 도 4에서 설명한 그룹 식별자 비트맵(440))의 하나의 비트(bit)를 서로 바꾸는 연산을 마이그레이션하고자 하는 그룹의 개수만큼 반복할 수 있다. 이때, 마이그레이션은 그룹에 속하는 복수의 샤드키에 대응하는 데이터를 순차적으로 이동시키는 과정과 그룹에 속하는 트랜잭션 로그를 반영하는 과정의 두 가지 과정을 마이그레이션하고자 하는 그룹의 개수만큼 반복하여 진행될 수 있다.
- [0049] 이때, 샤드키에 대응하는 데이터를 이동시키는 과정은 앞서 설명한 바와 같이, 제1 데이터베이스에서 해당 샤드키에 대응하는 데이터가 있는 다수의 테이블들에 대해서 각 테이블마다 데이터를 선택(SELECT)해서 선택된 데이터를 제2 데이터베이스의 해당 테이블로 insert(INsert)하는 것일 수 있다. 이 경우, 컴퓨터 장치(500)는 제1 데이터베이스에서 특정 샤드키를 조건으로 데이터를 선택할 때에는 배타적 선택(예를 들어, SQL에서 업데이트를 위한 select for update) 구문을 수행하고, 그 결과를 제2 데이터베이스에 insert한 다음 상술한 배타적 선택을 마칠 수 있다. 배타적 선택이 유지되는 동안에 제1 데이터베이스에서 해당 샤드키로 데이터를 변경하는 것은 락(lock)을 통해 블록(block)될 수 있다. 배타적 선택이 종료되면, 락이 해제되면서 블록되어 있던 데이터 변경(트랜잭션의 반영)이 진행될 수 있으며, 이러한 제1 데이터베이스에서의 데이터의 변경은 트랜잭션 로그를 통해서 제2 데이터베이스에 반영(해당 샤드키에 대응하는 데이터에 반영)될 수 있다. 다시 말해, 샤드키에 대응하는 옮기는 구간과 해당 샤드키에 대응하는 데이터를 변경하는 연산이 서로 겹치지 않도록 함으로써, 데이터의 이동에 따른 무결성이 보장될 수 있다.
- [0050] 하나의 그룹에 속하는 모든 샤드키의 데이터를 이동하고, 데이터의 이동 도중에 발생하는 트랜잭션까지 트랜잭션 로그를 이용하여 모두 반영한 경우, 제1 데이터베이스(보다 자세하게는 제1 데이터베이스를 관리하는 데이터베이스 서버)는 현재 진행 중인 트랜잭션 중에서 해당 그룹과 관련된 트랜잭션이 존재하지 않을 때까지 대기할 수 있으며, 현재 진행 중인 트랜잭션 중에서 해당 그룹과 관련된 트랜잭션이 존재하지 않을 때, 제1 데이터베이스의 그룹 식별자 비트맵에서 해당 그룹에 해당하는 비트를 오프(off)시킬 수 있다. 예를 들어, 제1 데이터베이스는 해당 그룹의 그룹 식별자와 연관하여 설정된 비트의 값을 '1'에서 '0'으로 변경할 수 있다. 한편, 컴퓨터 장치(500)는 트랜잭션 로그를 통해 제1 데이터베이스의 그룹 식별자 비트맵에서 해당 그룹에 해당하는 비트를 오프(off)시키는 로그에 따라 제2 데이터베이스의 그룹 식별자 비트맵에서 해당 그룹에 해당하는 비트를 온(On)시킬 수 있다. 예를 들어, 컴퓨터 장치(500)는 제2 데이터베이스(보다 자세하게는 제2 데이터베이스를 관리하는 데이터베이스 서버)가 해당 그룹의 그룹 식별자와 연관하여 설정된 비트의 값을 '0'에서 '1'으로 변경하도록 제어할 수 있다.
- [0051] 도 7 및 도 8은 본 발명의 일실시예에 따른 데이터 마이그레이션 방법이 적용되는 과정의 예를 도시한 도면들이다.
- [0052] 우선, 도 7은 새로운 노드가 추가됨에 따라 마이그레이션이 발생하는 예를 설명하고 있다. 데이터베이스 샤딩 환경에서의 모든 노드들은 동일한 스키마를 유지할 수 있으며, 글로벌 테이블을 공유할 수 있다. 도 7에 도시된 기존 노드인 노드 1(710) 역시 스키마와 글로벌 테이블을 공유할 수 있다. 도 7에서는 노드 1(710)이 마스터 데이터베이스 1(711)과 슬레이브 데이터베이스 1(712)을 포함하는 예를 나타내고 있다.
- [0053] 이때, 새로운 노드인 노드 2(720)가 추가되는 상황을 고려할 수 있다. 이미 설명한 바와 같이, 데이터베이스

샤딩 환경에서의 모든 노드들은 동일한 스키마와 글로벌 테이블을 공유하기 때문에, 노드 2(720)에도 이러한 스키마와 글로벌 테이블이 공유될 수 있다. 이때, 공유될 스키마와 글로벌 테이블이 도 6을 통해 설명한 데이터 마이그레이션 방법에 따라 노드 1(710)의 마스터 데이터베이스 1(711)에서 노드 2(720)의 마스터 데이터베이스 2(721)로 복제될 수 있다. 여기서 마이그레이터(migrator, 730)는 앞서 설명한 컴퓨터 장치(500)가 데이터의 마이그레이션을 위해 포함하는 모듈일 수 있다.

- [0054] 다만, 스키마나 글로벌 테이블은 이용자의 요구에 따라 임의로 처리되는 데이터나 샤딩된 데이터가 아니기 때문에 마이그레이션 도중에 트랜잭션의 반영이나 그룹 식별자의 변경을 고려할 필요가 없다.
- [0055] 한편, 도 8은 기존 노드(710)의 샤딩된 데이터를 새로운 노드인 노드 2(720)로 마이그레이션하여 데이터를 리밸런싱하는 예를 나타내고 있다. 예를 들어, 노드 1(710)이 관리하는 그룹 식별자 '3'에 대응하는 데이터를 노드 2(720)로 마이그레이션하여 데이터를 리밸런싱하는 과정을 고려할 수 있다.
- [0056] 마이그레이터(730)는 그룹 식별자 '3'에 대응하는 데이터를 위한 샤드키들을 식별할 수 있다. 이미 설명한 바와 같이, 컴퓨터 장치(500)는 샤드키-그룹 식별자 테이블을 통해 그룹 식별자 '3'에 대응하는 샤드키들을 확인할 수 있다. 또한, 마이그레이터(730)는 노드 1(710)의 슬레이브 데이터베이스 1(712)에서 상기 식별된 샤드키들에 대응하는 데이터를 노드 2(720)의 마스터 데이터베이스 2(721)로 마이그레이션할 수 있다. 이 경우, 마이그레이션하고자 하는 데이터의 소스 데이터베이스는 노드 1(710)의 슬레이브 데이터베이스 1(712)가 될 수 있고, 데이터가 전달되는 타겟 데이터베이스는 노드 2(720)의 마스터 데이터베이스 2(721)가 될 수 있다. 실시 예에 따라 노드 1(710)의 마스터 데이터베이스 1(711)이 소스 데이터베이스가 될 수도 있다.
- [0057] 예를 들어 그룹 식별자 '3'에 대응하는 샤드키들이 'a', 'b', 'c', 'd'라 가정한다.
- [0058] 마이그레이션 도중에 샤드키 'a'에 대응하는 데이터에 대한 트랜잭션이 발생하였고, 샤드키 'a'에 대응하는 데이터는 슬레이브 데이터베이스 1(712)에서 마스터 데이터베이스 2(721)로 마이그레이션이 완료되었다고 가정한다. 이 경우, 마스터 데이터베이스 1(711)에 저장된 샤드키 'a'에 대응하는 데이터와 슬레이브 데이터베이스 1(712) 및 마스터 데이터베이스 2(721)에 저장된 샤드키 'a'에 대응하는 데이터는 서로 달라질 수 있다.
- [0059] 따라서, 마스터 데이터베이스 1(711)에 반영된 트랜잭션이 각각 슬레이브 데이터베이스 1(712) 및 마스터 데이터베이스 2(721)에도 반영되어야 한다. 이때, 마이그레이터(730)는 트랜잭션 로그를 분석하여 해당 트랜잭션을 슬레이브 데이터베이스 1(712) 및 마스터 데이터베이스 2(721)에 각각 동시 반영할 수 있다. 이미 설명한 바와 같이 샤드키 단위의 데이터들은 서로에 대해 독립적인 처리가 가능하기 때문에, 마이그레이션이 특정 샤드키에 대응하는 데이터에 대해 완료되었다면, 전체 마이그레이션이 완료되기 이전에도 트랜잭션을 동시 반영하여 데이터를 동기화할 수 있다. 따라서, 복제 지연 시간을 줄일 수 있다.
- [0060] 한편, 그룹 식별자 '3'을 관리하는 노드에 대한 정보는 전체 샤드키들에 대한 마이그레이션이 모두 완료된 후에 변경될 수 있다. 예를 들어, 마이그레이션이 진행 중인 경우에는 그룹 식별자 '3'의 데이터를 노드 1(710)이 관리하는 것으로 되어 있기 때문에 앞서 설명한 바와 같이 그룹 식별자 '3'의 데이터에 대해 발생하는 트랜잭션은 마스터 데이터베이스 1(711)에 반영된 후 트랜잭션 로그를 이용하여 슬레이브 데이터베이스 1(712) 및 마스터 데이터베이스 2(721)에 각각 동시 반영될 수 있다. 반면, 마이그레이션이 모두 완료된 후에는 그룹 식별자 '3'을 관리하는 노드가 노드 1(710)에서 노드 2(720)로 변경될 수 있다. 이때, 그룹 식별자 '3'을 관리하는 노드를 변경하는 아주 짧은 시간 동안만 그룹 식별자 '3'에 대응하는 데이터로의 접근이 차단될 수 있다. 따라서, 마이그레이션되는 데이터로의 접근 차단 시간을 획기적으로 줄일 수 있게 된다. 그룹 식별자 '3'을 관리하는 노드가 변경되고 나면, 그룹 식별자 '3'에 포함된 샤드키들과 관련된 트랜잭션들이 노드 2(720)에 적용될 수 있다.
- [0061] 이처럼, 마스터 데이터베이스와 슬레이브 데이터베이스간의 데이터 복제뿐만 아니라, 노드간 데이터 복제에도 본 발명의 실시예들에 따른 데이터 마이그레이션 방법이 활용될 수 있다.
- [0062] 이처럼 본 발명의 실시예들에 따르면, 제1 데이터베이스와 제2 데이터베이스간의 데이터 마이그레이션(migration) 도중에 제1 데이터베이스에 반영되는 트랜잭션을 트랜잭션 로그를 이용하여 제2 데이터베이스에 동시 반영함으로써, 복제 지연 시간을 줄일 수 있으며, 마이그레이션되는 데이터로의 접근 차단 시간을 줄일 수 있다. 또한, 트랜잭션을 동시에 반영하면서 데이터의 마이그레이션이 종료되는 시점에 제1 데이터베이스와 제2 데이터베이스에서 각각 관리되는 그룹 식별자들을 변경함에 따라 마이그레이션되는 데이터로의 접근 차단 시간을 최소화할 수 있다.
- [0063] 이상에서 설명된 시스템 또는 장치는 하드웨어 구성요소, 소프트웨어 구성요소 또는 하드웨어 구성요소 및 소프

트웨어 구성요소의 조합으로 구현될 수 있다. 예를 들어, 실시예들에서 설명된 장치 및 구성요소는, 예를 들어, 프로세서, 콘트롤러, ALU(arithmetic logic unit), 디지털 신호 프로세서(digital signal processor), 마이크로컴퓨터, FPGA(field programmable gate array), PLU(programmable logic unit), 마이크로프로세서, 또는 명령(instruction)을 실행하고 응답할 수 있는 다른 어떠한 장치와 같이, 하나 이상의 범용 컴퓨터 또는 특수 목적 컴퓨터를 이용하여 구현될 수 있다. 처리 장치는 운영 체제(OS) 및 상기 운영 체제 상에서 수행되는 하나 이상의 소프트웨어 어플리케이션을 수행할 수 있다. 또한, 처리 장치는 소프트웨어의 실행에 응답하여, 데이터를 접근, 저장, 조작, 처리 및 생성할 수도 있다. 이해의 편의를 위하여, 처리 장치는 하나가 사용되는 것으로 설명된 경우도 있지만, 해당 기술분야에서 통상의 지식을 가진 자는, 처리 장치가 복수 개의 처리 요소(processing element) 및/또는 복수 유형의 처리 요소를 포함할 수 있음을 알 수 있다. 예를 들어, 처리 장치는 복수 개의 프로세서 또는 하나의 프로세서 및 하나의 콘트롤러를 포함할 수 있다. 또한, 병렬 프로세서(parallel processor)와 같은, 다른 처리 구성(processing configuration)도 가능하다.

[0064] 소프트웨어는 컴퓨터 프로그램(computer program), 코드(code), 명령(instruction), 또는 이들 중 하나 이상의 조합을 포함할 수 있으며, 원하는 대로 동작하도록 처리 장치를 구성하거나 독립적으로 또는 결합적으로(collectively) 처리 장치를 명령할 수 있다. 소프트웨어 및/또는 데이터는, 처리 장치에 의하여 해석되거나 처리 장치에 명령 또는 데이터를 제공하기 위하여, 어떤 유형의 기계, 구성요소(component), 물리적 장치, 가상 장치(virtual equipment), 컴퓨터 저장 매체 또는 장치에 구체화(embody)될 수 있다. 소프트웨어는 네트워크로 연결된 컴퓨터 시스템 상에 분산되어서, 분산된 방법으로 저장되거나 실행될 수도 있다. 소프트웨어 및 데이터는 하나 이상의 컴퓨터 판독 가능 기록매체에 저장될 수 있다.

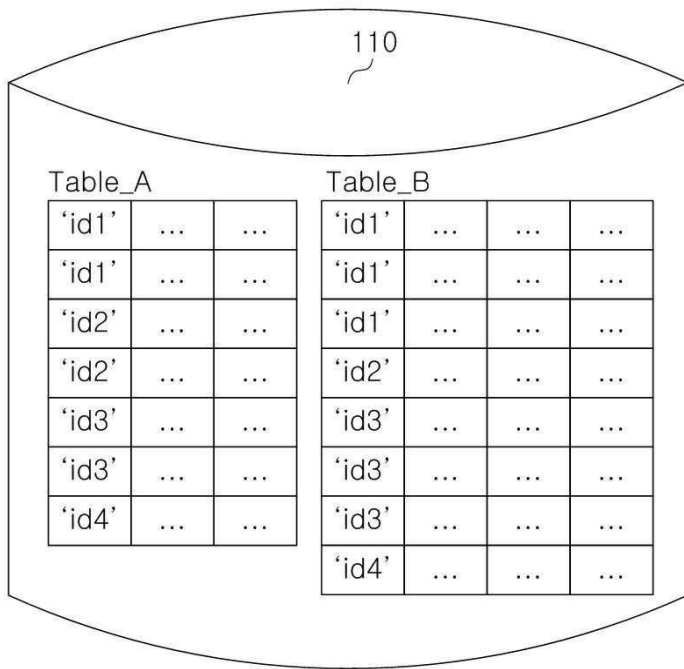
[0065] 실시예에 따른 방법은 다양한 컴퓨터 수단을 통하여 수행될 수 있는 프로그램 명령 형태로 구현되어 컴퓨터 판독 가능 매체에 기록될 수 있다. 상기 컴퓨터 판독 가능 매체는 프로그램 명령, 데이터 파일, 데이터 구조 등을 단독으로 또는 조합하여 포함할 수 있다. 매체는 컴퓨터로 실행 가능한 프로그램을 계속 저장하거나, 실행 또는 다운로드를 위해 임시 저장하는 것일 수도 있다. 또한, 매체는 단일 또는 수개 하드웨어가 결합된 형태의 다양한 기록수단 또는 저장수단일 수 있는데, 어떤 컴퓨터 시스템에 직접 접속되는 매체에 한정되지 않고, 네트워크 상에 분산 존재하는 것일 수도 있다. 매체의 예시로는, 하드 디스크, 플로피 디스크 및 자기 테이프와 같은 자기 매체, CD-ROM 및 DVD와 같은 광기록 매체, 플롭티컬 디스크(floptical disk)와 같은 자기-광 매체(magneto-optical medium), 및 ROM, RAM, 플래시 메모리 등을 포함하여 프로그램 명령어가 저장되도록 구성된 것이 있을 수 있다. 또한, 다른 매체의 예시로, 애플리케이션을 유통하는 앱 스토어나 기타 다양한 소프트웨어를 공급 내지 유통하는 사이트, 서버 등에서 관리하는 기록매체 내지 저장매체도 들 수 있다. 프로그램 명령의 예에는 컴파일러에 의해 만들어지는 것과 같은 기계어 코드뿐만 아니라 인터프리터 등을 사용해서 컴퓨터에 의해서 실행될 수 있는 고급 언어 코드를 포함한다.

[0066] 이상과 같이 실시예들이 비록 한정된 실시예와 도면에 의해 설명되었으나, 해당 기술분야에서 통상의 지식을 가진 자라면 상기의 기재로부터 다양한 수정 및 변형이 가능하다. 예를 들어, 설명된 기술들이 설명된 방법과 다른 순서로 수행되거나, 및/또는 설명된 시스템, 구조, 장치, 회로 등의 구성요소들이 설명된 방법과 다른 형태로 결합 또는 조합되거나, 다른 구성요소 또는 균등물에 의하여 대치되거나 치환되더라도 적절한 결과가 달성될 수 있다.

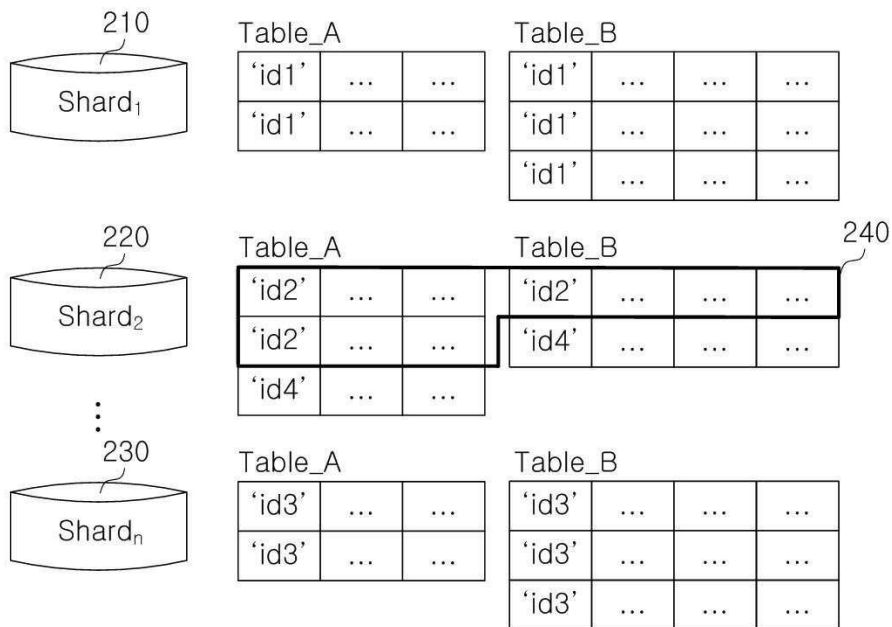
[0067] 그러므로, 다른 구현들, 다른 실시예들 및 특허청구범위와 균등한 것들도 후술하는 특허청구범위의 범위에 속한다.

도면

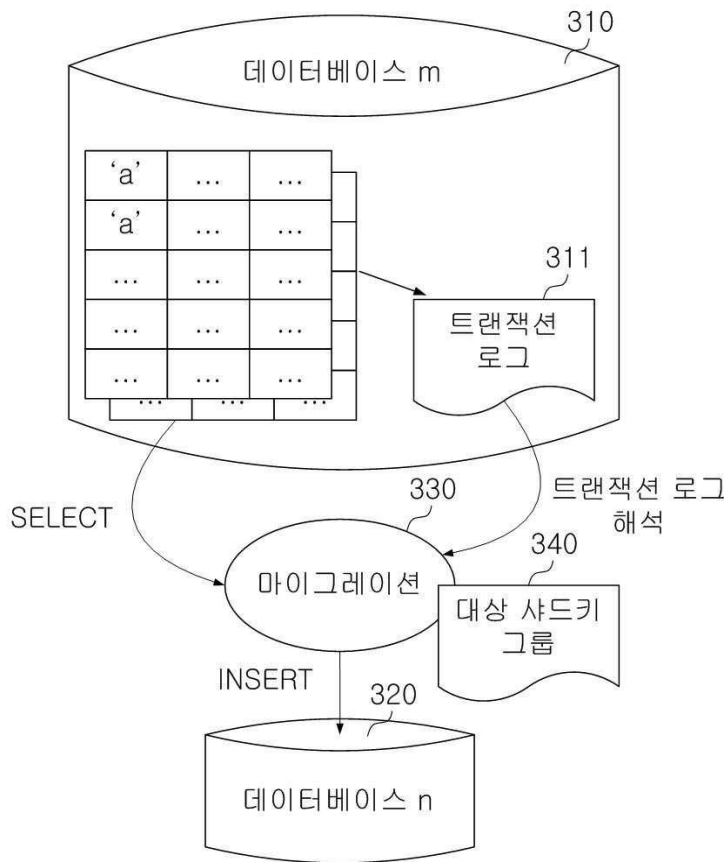
도면1



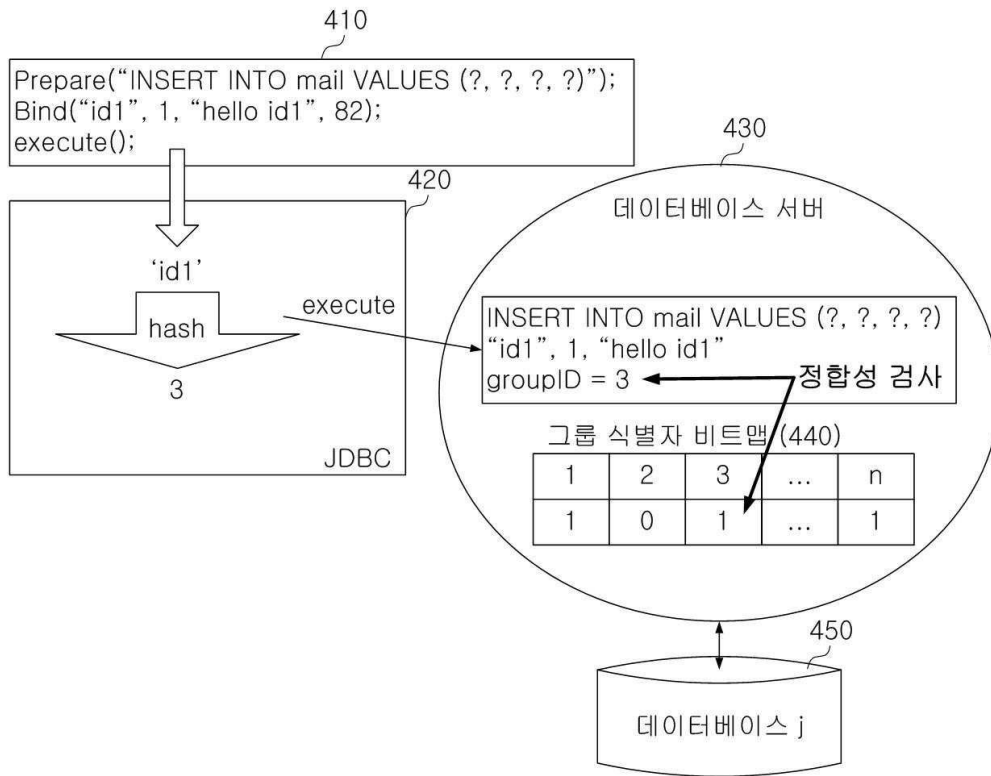
도면2



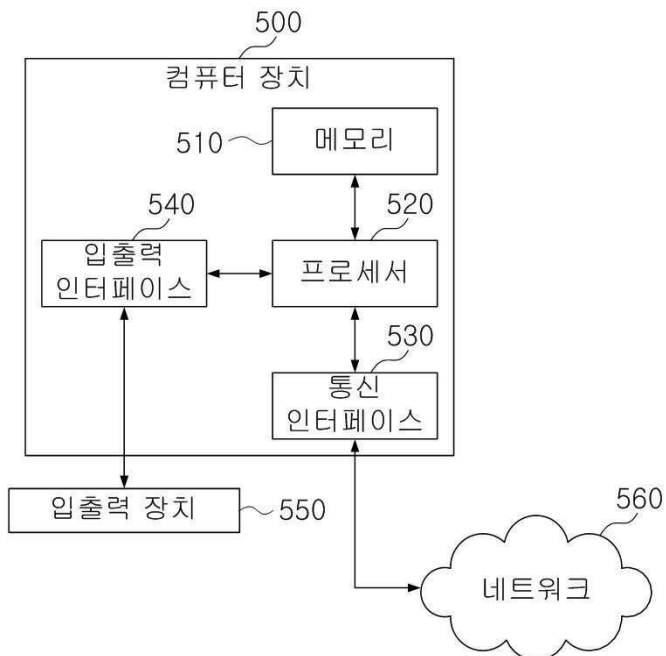
도면3



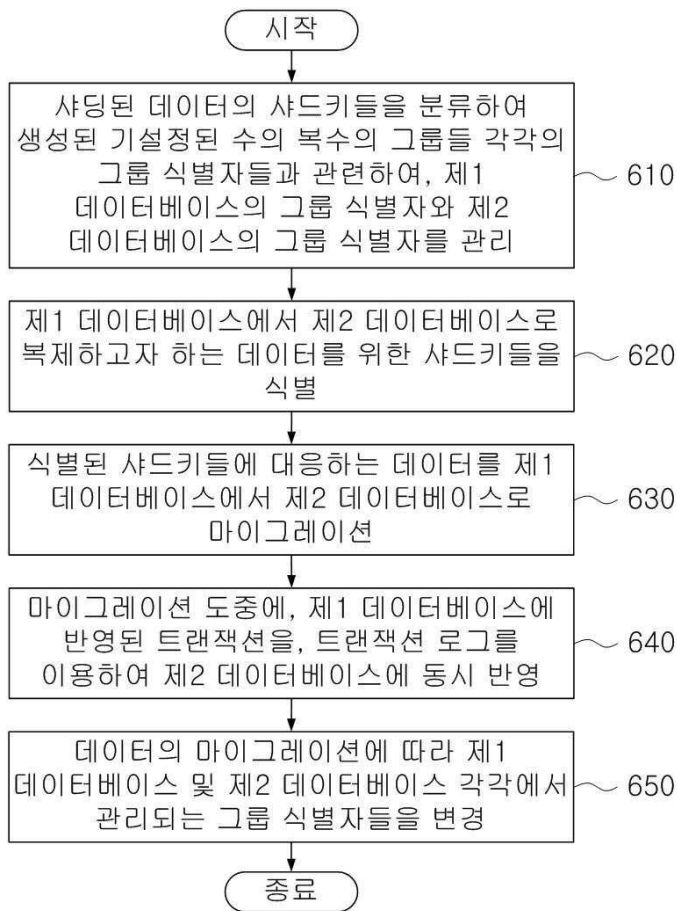
도면4



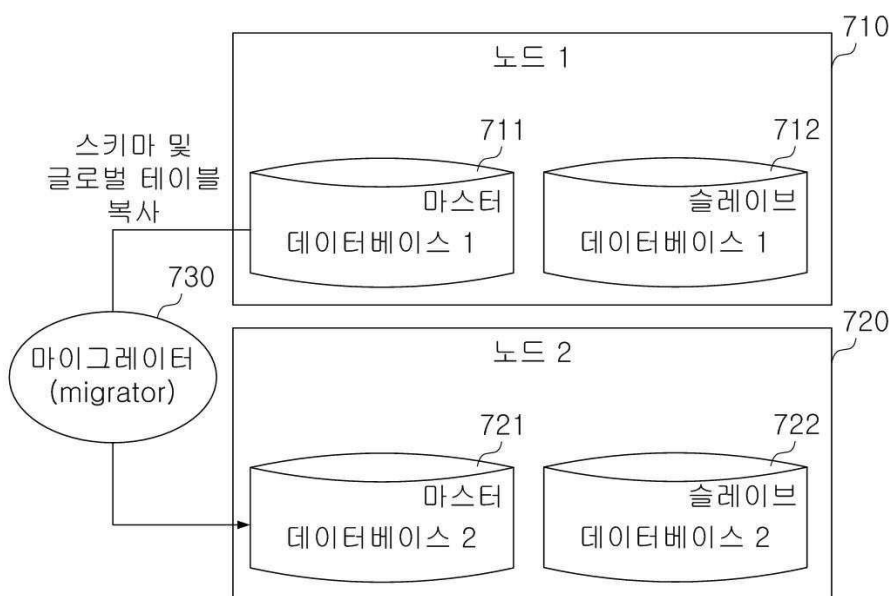
도면5



도면6



도면7



도면8

