

CUBRID 데이터베이스 복제

고객지원팀



CUBRID
More than open source!

목차



- CUBRID 복제 시스템
- 복제 기능
- 복제 시나리오
- 복제 전체 구성도
- 복제 제약 사항
- 복제 주의 사항

CUBRID 복제 시스템



● CUBRID2008 복제 시스템

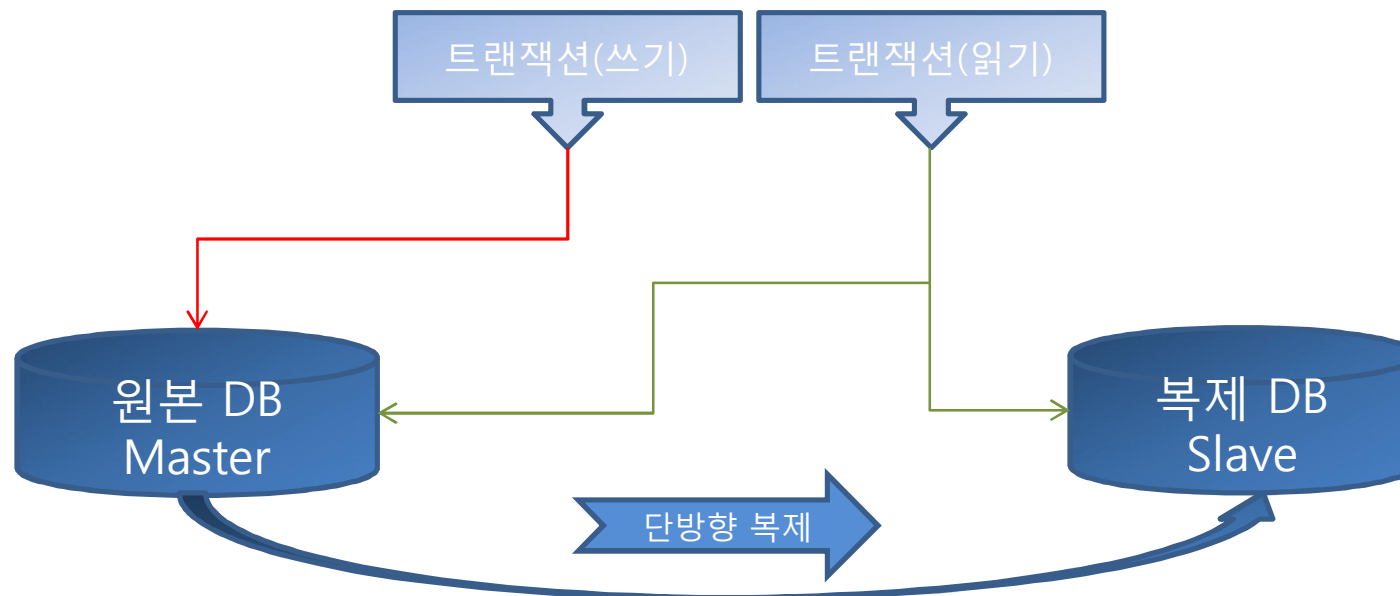
- CUBRID 복제는 하나의 데이터베이스에 저장된 객체를 물리적으로 분리된 다른 데이터베이스에 복사하여 같은 객체를 이용하는 응용의 접근을 마스터와 슬레이브 데이터베이스 서버로 분산시켜 성능을 높일 수 있다.
- CUBRID 복제는 원본 데이터베이스를 마스터 데이터베이스라고 하고, 복제되는 데이터베이스를 슬레이브 데이터베이스라 한다.
- CUBRID 복제는 단방향 복제로 마스터 데이터베이스에서 슬레이브 데이터베이스로만 복제가 가능하며, 테이블의 기본 키(Primary Key) 기반으로 이루어 진다. 즉, 복제 반영할 테이블에 기본 키가 설정되어야 복제가 가능하다.
- 복제는 마스터 데이터베이스의 트랜잭션 로그 기반으로 한다.

복제 기능-단방향 복제



● 단방향 복제

- CUBRID 복제는 단방향 복제만을 지원한다. 즉, 아래 그림과 같이 마스터에서 읽기/쓰기 연산이 모두 가능하지만 슬레이브에서는 읽기만을 허용한다.



복제 기능-복제 기법

● 데이터 복제 기법

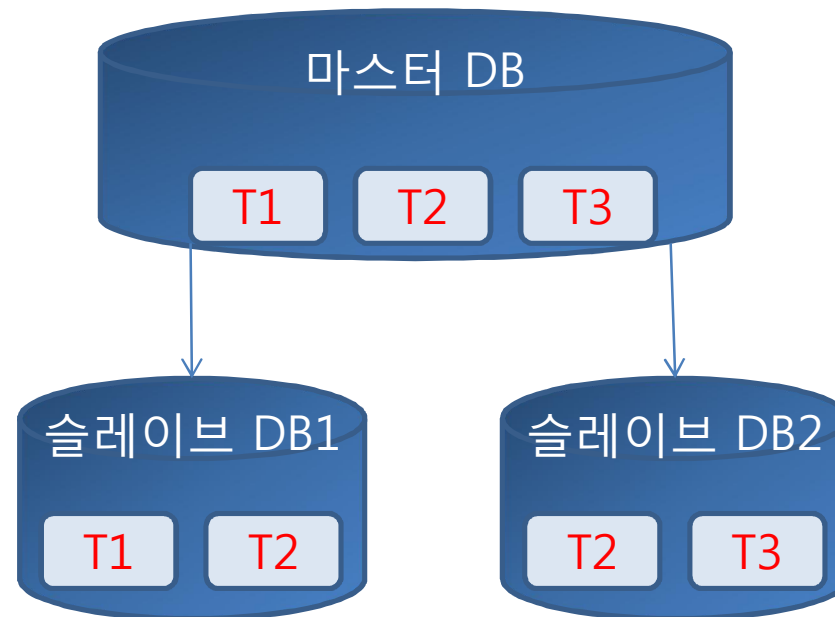
- CUBRID 복제는 마스터의 데이터를 갱신하는 작업과 슬레이브 데이터를 동일하게 갱신하는 작업이 하나의 트랜잭션으로 처리되어 데이터의 일관성을 보장한다.
- CUBRID는 트랜잭션 로그를 이용하여 비동기식 방식으로 복제 기능을 제공하여 데이터 일관성 보장은 물론 최대한의 가용성을 확보할 수 있도록 한다.



복제 기능-복제 그룹

- 복제 그룹(Replication Group)

- CUBRID 복제는 마스터 데이터베이스의 일부 일부만 복제 그룹으로 지정하여 슬레이브 데이터베이스로 복제할 수 있다. 즉, 복제대상 테이블과 복제 제외 테이블을 구분하여 복제대상 테이블만 슬레이브로 복제가 가능하다.

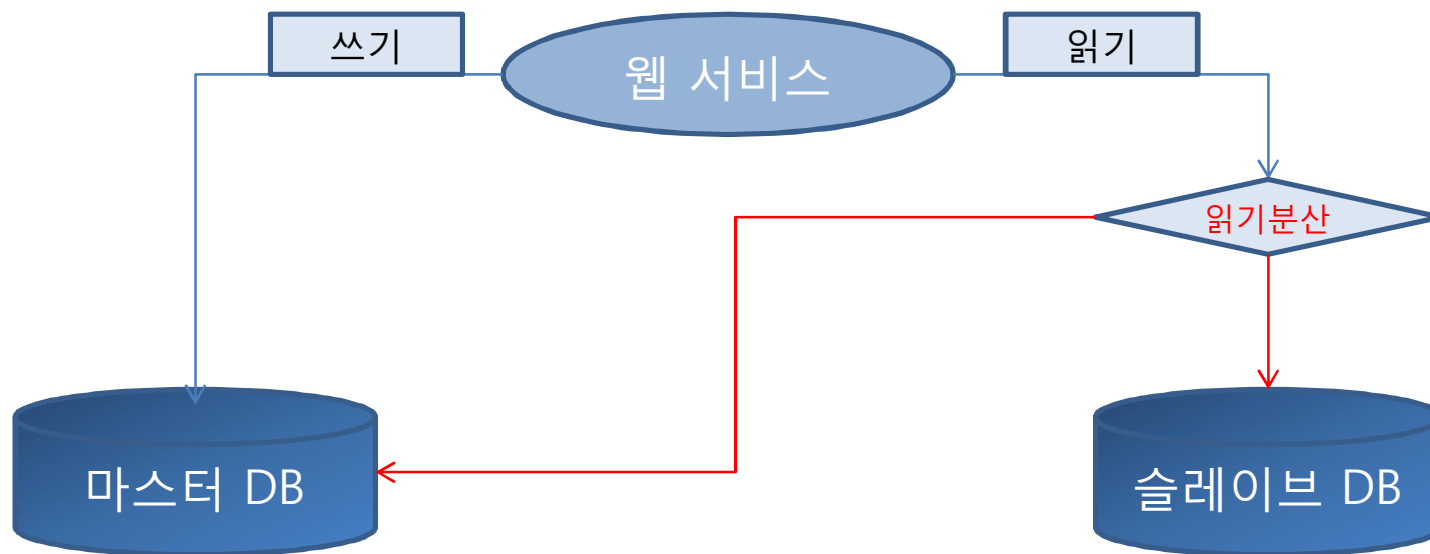


복제 시나리오 - 업무 부하 분산



● 부하 분산을 통한 성능 향상

- CUBRID 복제는 하나의 데이터베이스가 응용의 성능 요구 사항을 만족하기 어려운 경우 읽기 연산을 분산할 때 성능을 향상시키는데 활용할 수 있다.
- 인터넷 서비스에 사용되고 있는 블로그, 게시판, 뉴스등 읽기 연산이 많은 응용에 사용하면 그림과 같이 부하 분산을 통해 성능 향상 효과를 얻을 수 있다.

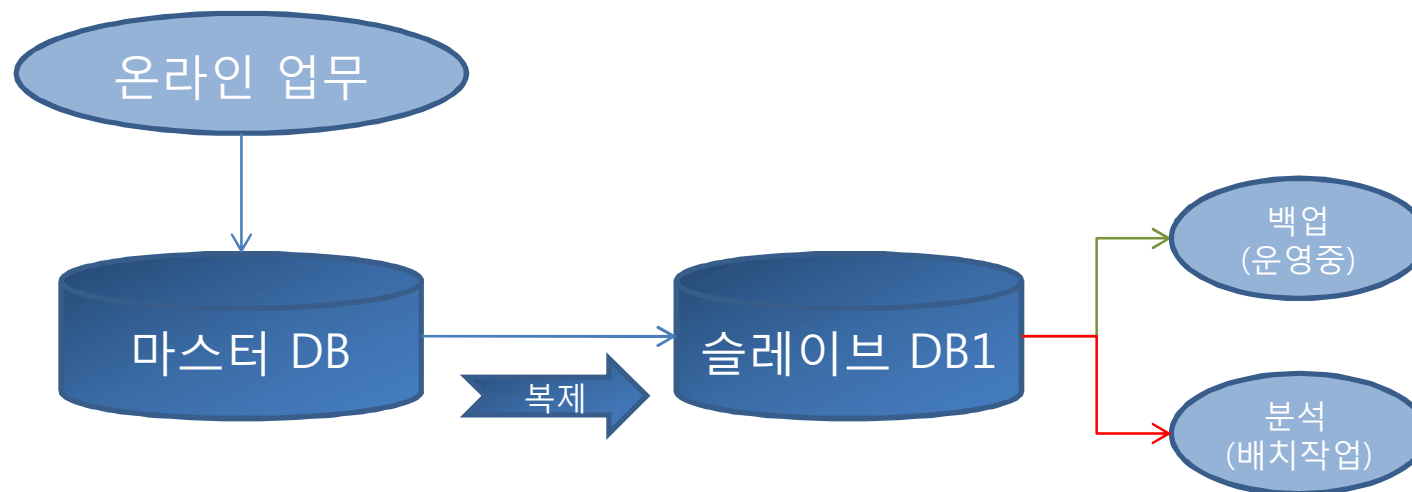


복제 시나리오-분산과 유연성 향상



- 분리 작업을 통한 유연성 향상

- CUBRID 복제를 이용하면 별도의 호스트에 마스터와 같은 내용을 갖는 데이터베이스를 확보할 수 있기 때문에 마스터 데이터베이스에 영향을 주지 않고, 백업/분석(배치작업) 등 데이터베이스에 부하를 주는 작업을 그림과 같이 분산하여 수행할 수 있다.



복제 시나리오-장애대비(1)



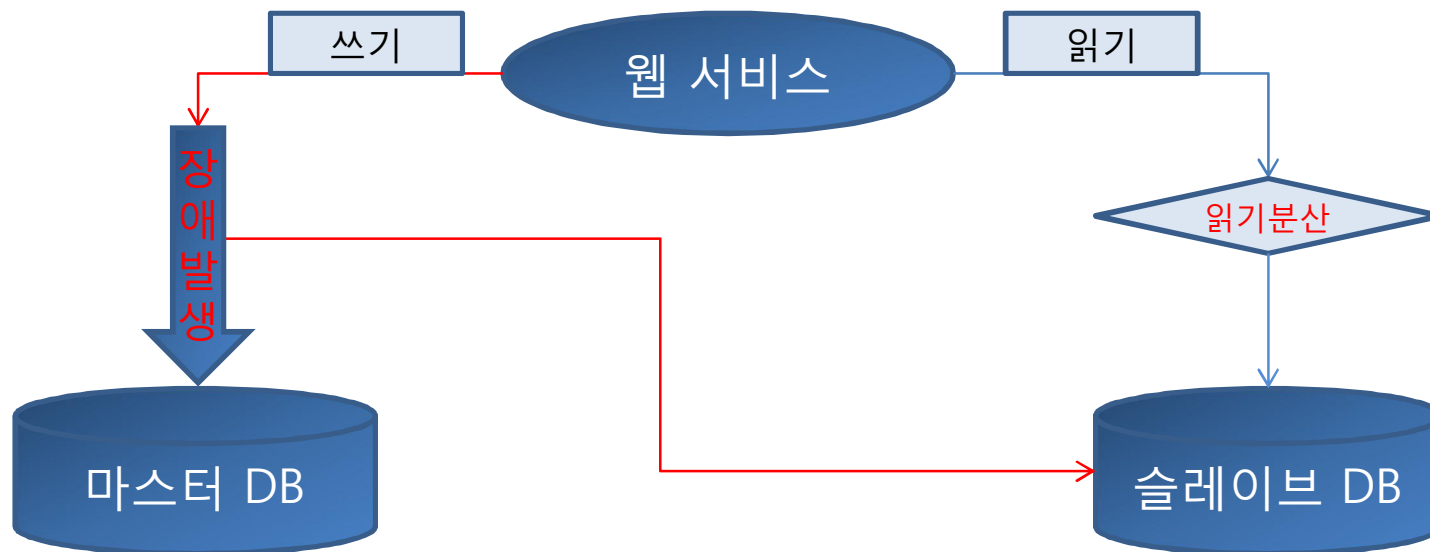
● 장애 대비 가용성 향상

- CUBRID 복제 구성 시 마스터에 장애가 발생하면 슬레이브를 마스터로 교체하여 장애에 효과적으로 대응할 수 있다. 장애종류에 따라, 마스터 데이터베이스의 백업을 이용한 복구 기능으로 충분할 수도 있으나, 만약, 마스터 호스트(서버) 자체의 장애가 발생할 경우에는 마스터 호스트(서버)의 이중화(HA)나 복제를 통하여 대처할 수 있다.
- 장애가 발생하면 복제 오류와 동기화가 정상적으로 이루어 졌는지 먼저 확인하고, 슬레이브 데이터베이스를 마스터로 교체할 수 있다.(일부 테이블만 복제하는 그룹복제 사용 시 슬레이브 데이터베이스로 전환할 수 없다.)
- 슬레이브를 마스터로 교체하는 작업은 그리 단순하지 않다. 특히, 응용 프로그램에 모든 작업을 마스터에서 슬레이브로 변경 설정해야 하므로, 마스터 데이터베이스의 복구 상황을 잘 고려하여 전환해야 된다.

복제 시나리오-장애대비(2)

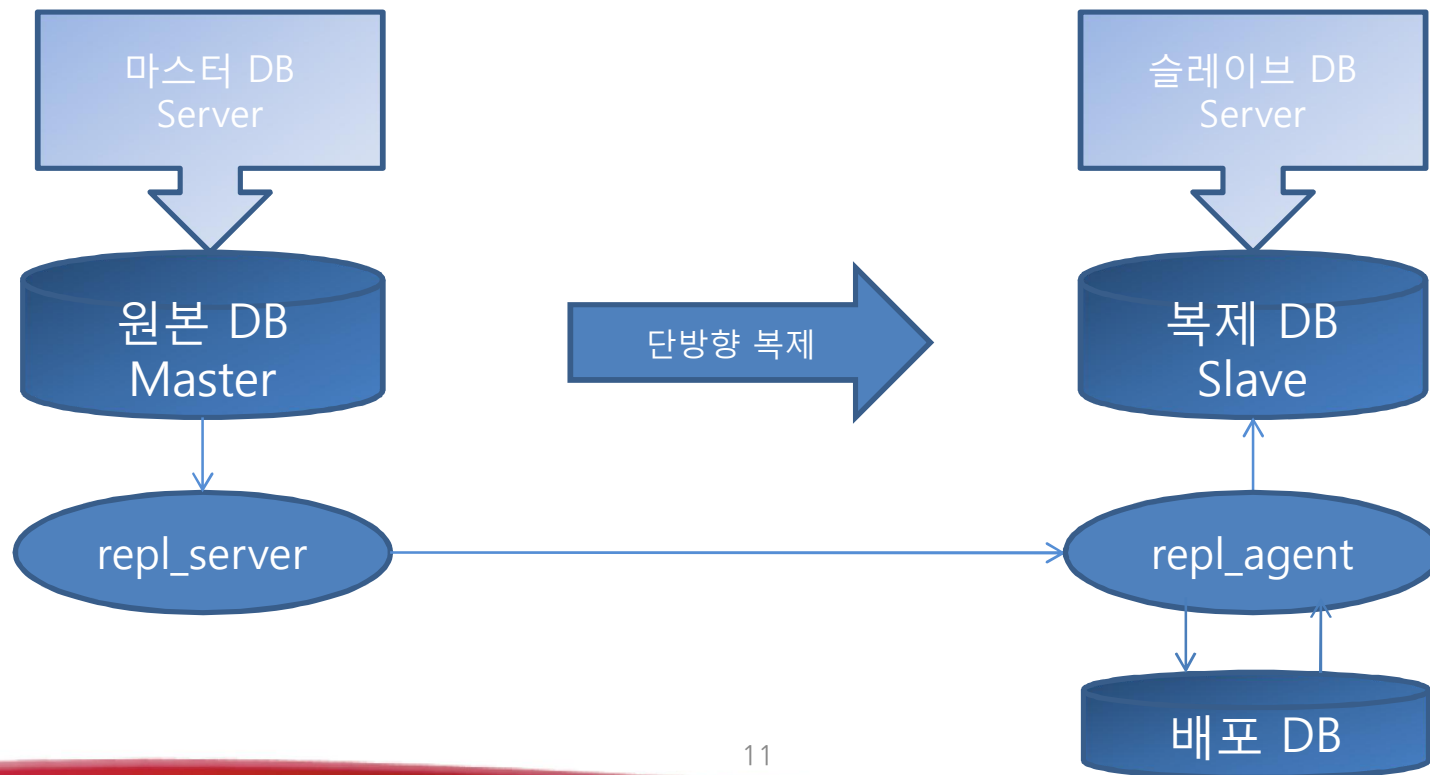
- 장애 대비 시 슬레이브 데이터베이스로 전환

- 마스터 데이터베이스 장애가 발생하고 복구 시간과 비용이 많이 소요되어 슬레이브 데이터베이스로 전환이 필요할 경우, 사전 복제 오류와 동기화가 이상이 없는지 확인하여 아래 그림과 같이 슬레이브 데이터베이스로 전환할 수 있다.



DB 복제 전체 구성도

- 복제 구성 시 데이터 복제 위한 구성도.
- 복제 시스템은 트랜잭션 로그 기반으로 동작하는데 복제 서버(repl_server)가 트랜잭션 로그를 복제 에이전트(repl_agent)에게 전달하고 복제 에이전트가 트랜잭션 로그를 분석하여 그 결과를 슬레이브 데이터베이스에 전달하는 구조로 복제를 진행 한다.



CUBRID 복제 제약 사항



● 복제의 제약 사항

- 복제는 Linux 및 Solaris 플랫폼을 비롯한 **Unix 계열**에서만 지원하고, 기본 키가 설정된 테이블에 대해서만 복제할 수 있다
- 마스터 데이터베이스와 슬레이브 데이터베이스는 **반드시 동일 플랫폼**으로 구성해야 한다. 예를 들어 마스터 데이터베이스는 Linux, 슬레이브 데이터베이스는 Solaris로 구축할 수 없다.
- 객체 타입을 속성으로 갖는 테이블의 경우에는 복제 시 오류가 발생할 수 있다. 객체 타입을 복제에서 사용하려면 외래키 **ON CACHE OBJECT**를 이용할 수 있다
- 복제는 마스터 데이터베이스의 모든 테이블에 **기본키가 설정되어 있어야만** 마스터 데이터베이스를 슬레이브 데이터베이스로 **교체**할 수 있다.

CUBRID 복제 주의 사항(1)



● 복제의 주의 사항

- 마스터 데이터베이스의 변경 사항은 대부분 실시간으로 슬레이브 데이터베이스에 반영되나, 한번에 다량의 데이터를 갱신하는 긴 트랜잭션의 경우 복제 지연 시간이 증가할 수 있다.
- 복제가 지연될 경우 복제 서버가 마스터 데이터베이스의 트랜잭션 보관 로그에서 데이터 변경 항목을 읽어야 하므로 가급적 5개에서 10개 정도의 트랜잭션 보관(아카이브) 로그는 삭제하지 않는 것이 좋다.
- repl_make_slavedb를 수행하여 슬레이브 데이터베이스를 새로 구축하는 경우에는 복제 에이전트 수행 전에 반드시 이전 복제 구성에서 사용했던 복제 로그, 추적 로그와 같은 로그 파일을 삭제해야 한다.
- TIMESTAMP는 슬레이브 데이터베이스에 반영된 시각이 아니라 마스터와 동일한 값으로 복제된다. .

CUBRID 복제 주의 사항(2)



● 복제의 주의 사항

- 마스터 데이터베이스 서버가 장애로 중지된 경우에는 복제 서버나 복제 에이전트는 중단되지 않고 장애 발생 이전에 변경된 데이터를 슬레이브 시스템에 복제한다. 그러나 슬레이브 데이터베이스 서버가 장애로 중지된 경우에는 복제 에이전트도 같이 중지된다. 따라서 슬레이브 데이터베이스 서버가 재시작한 후 수동으로 복제 에이전트를 재시작해야 한다.
- 복제가 지연될 경우 복제 서버가 마스터 데이터베이스의 트랜잭션 보관 로그에서 데이터 변경 항목을 읽어야 하므로 가급적 5개에서 10개 정도의 트랜잭션 보관(아카이브) 로그는 삭제하지 않는 것이 좋다.
- 복제 로그 반드시 손상되지 않도록 한다. 또한 배포 데이터베이스에 접속하여 수작업으로 데이터 항목을 수정하지 않을 것을 권장한다.

Thank You...

CUBRID
More than open source!