데이터베이스

분산 DB

2005년 하반기

INFOREVER Consulting co.,ltd.

분산 DB

1. 분산 Database의 이해
   가. 분산 Database의 정의
      - 하나의 논리적 데이터베이스가 통신 네트워크로 연결된 여러 컴퓨터에 물리적으로 저장되어 관리되는 데이터베이스
      - 각각의 컴퓨터에는 지역 데이터베이스 관리시스템 (Local DBMS) 과 분산 데이터베이스 관리시스템 (Distributed DBMS) 을 내장하고 있음
      - 다른 장소에 위치한 데이터 정보를 제공해 주는 분산 데이터 사전 (Distributed Data Dictionary) 및 주소록을 가지고 있음
   나. 분산 Database의 필요성
      - 기업의 성장에 따른 조직의 분권화, 유연한 확장성
      - 지역별, 부문별 분산 정보의 통합처리의 필요성
      - 컴퓨터 및 통신망, 분산처리 기술의 발달
      - 데이터베이스의 질의 처리 속도 향상
      - 데이터베이스의 데이터 공유, 신뢰성 및 가용성 제공
   다. 분산 DB의 목적
      - 데이터 처리의 지역화 : 통신비용의 감소 및 데이터 처리 집중화 방지
      - 데이터 운영 및 관리의 지역화 : 데이터에 대한 이해도가 높은 집단이 관리
      - 데이터 처리 부하의 분산 및 병렬 데이터 처리 : 데이터 처리 속도 향상
      - 데이터의 가용도와 신뢰성 향상 : 데이터를 복제

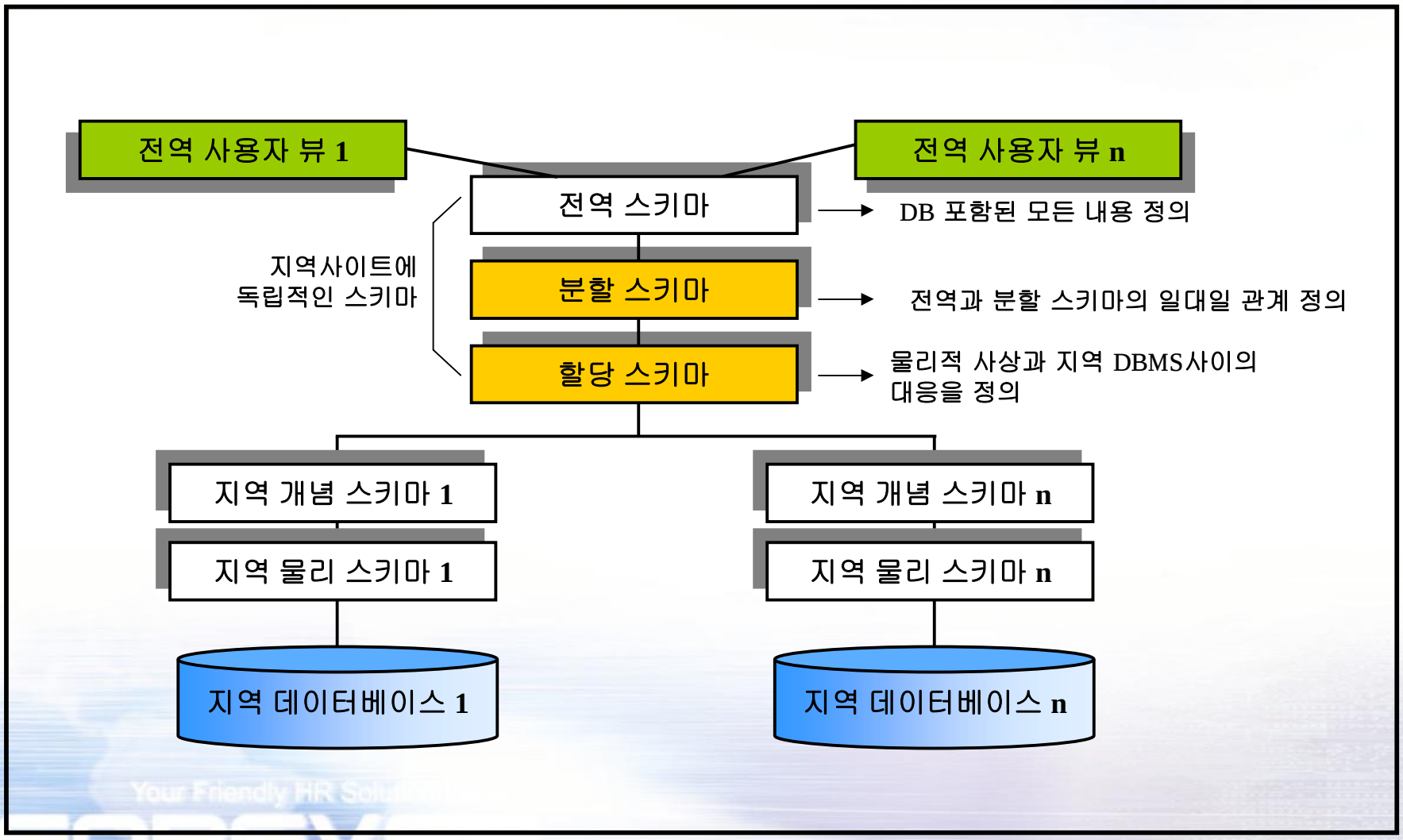
분산 DB

2. 분산 Database 시스템의 형태

가. 분산 Database 시스템의 유형

	동질 분산 DB 관리 시스템	이질 분산 DB 관리 시스템
정 의	- 모든 지역에 동일한 DBMS를 사용하는 방식 - 전역 Interface를 통해 완전 동질 투명성을 제공	- 지역간 이기종 DBMS가 지역 데이터베이스를 관리하고 분산 DBMS가 이들 데이터베이스를 관리함 - 기 존재하는 지역 데이터베이스를 사후 통합하기 위해 사용됨
특 징	- 전역스키마(global schema)를 통하여 하나의 데이터베이스를 사용하는 것처럼 이용 - 사용자들이 특정 지역 데이터베이스만은 직접 사용할 수 없음 - 전역 개념 스키마에 속한 테이블들을 지역별로 분할하여 할당하는 분할 스키마와 할당 스키마를 적용 - 전역에서 지역으로의 하향식 스키마 구조	- 지역데이터베이스는 전역 스키마에 포함될 내용을 자치적으로 결정함 (local autonomy, 지역자치성) - 지역 사용자들이 존재(local user) - 지역으로부터 전역으로의 상향식 스키마 구조

나. 분산 데이터베이스 관리시스템의 구성도

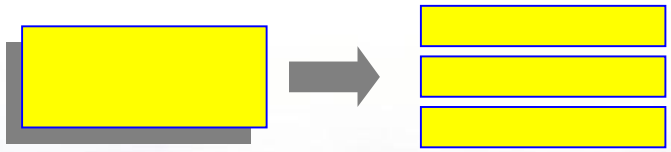
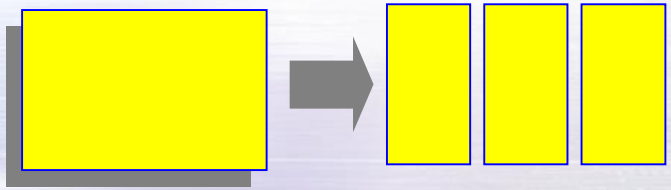


3. 분산 Database 설계

가. 분산 Database의 목표

구분	상세 내용
처리의 국지성	- 데이터 처리의 지역성을 고려한 데이터 분산
가용도 및 신뢰도	- 같은 정보를 여러 곳에 저장함으로써 가용도 및 신뢰도에 대한 보장
부하분산	- 각 사이트의 컴퓨터를 이용 데이터베이스 처리의 병렬화를 최대화 시킴
기억장소 및 처리량	- 각 사이트의 저장 및 처리량의 극대화

나. 데이터베이스 설계 전략

복제 (Replication)	- 동일한 DB의 사본을 둘 이상의 장소에 중복 저장 - 장점 : 빠른 응답속도, DB의 가용성 및 신뢰성 - 단점 : DB 갱신 시 모든 복제된 DB도 갱신(복잡, 처리비용증가) 저장장소 용량증가 (대용량DISK 필요)
수평분할 (Horizontal Partition)	- 동일한 FORMAT의 DB를 나누어 저장 -> RECORD별 분할 
수직분할 (Vertical Partition)	- 동일한 FORMAT의 DB를 기본 키를 기준으로 사용되는 내용별로 분할 -> FIELD별 분할 

4. 분산 Database의 투명성과 장단점

가. 분산 Database의 투명성

특 성	주 요 개 념
위치 투명성 (Location Transparency)	- 사용자나 응용 프로그램이 접근할 데이터의 물리적 위치를 알아야 할 필요가 없는 성질 - 이를 보장하기위해 DBMS 는 Distributed Data Dictionary /Directory 가 필요
복제 무관성 (Replication Transparency)	- 사용자나 응용 프로그램이 접근할 데이터가 물리적으로 여러 곳에 복제되어 있는지의 여부를 알 필요가 없는 성질
병행 무관성 (Concurrency Transparency)	- 여러 사용자나 응용 프로그램이 동시에 분산 데이터베이스에 대한 트랜잭션을 수행하는 경우에도 결과에 이상이 발생하지 않는 성질 - Locking, 타임스탬프순서 기법 이용
분할 투명성 (Partition Transparency)	- 사용자가 하나의 논리적 릴레이션이 여러 단편으로 분할되어 각 단편의 사본이 여러 site에 저장되어 있음을 알 필요가 없는 성질
장애 무관성 (Failure Transparency)	- 데이터베이스가 분산되어 있는 각 지역의 시스템이나 통신망에 이상이 생기더라도, 데이터의 무결성을 보존할 수 있는 성질 - 2 PC (Phase Commit) 활용

나. 분산 Database의 장단점 및 일반DB와 비교

1) 분산 Database의 장단점

장점	단점
- 빠른 응답 속도와 통신 비용 절감 - 데이터의 가용성과 신뢰성 증가 - 시스템 규모의 적절한 조절 - 각 지역 사용자의 요구 만족 - 특정 Level지역 장애 발생 시 타 지역의 영향 최소화	- 설계, 관리의 복잡성과 비용 - 통제의 어려움 - 데이터에 대한 무결성의 위협 - 처리 지역적 환경에 따른 특성에 치우침 - 장애 발생 시 복구 작업이 어려움 - 완벽한 무결성 자원 어려움

2) 분산 Database와 일반 Database의 비교

항목	일반 DB	분산 DB
통제방식	전역 DBA에 의한 중앙통제방식	지역 DBA에 의한 지역자치성
데이터 형태	데이터 독립성	데이터 독립성 및 분산투명성을 강조
데이터 중복성	데이터 공유를 통한 중복도 감소	중복성 및 지역성은 바람직
데이터 구조	데이터 액세스를 위한 복잡한 구조 운영	사이트간의 물리적 구조를 이용하여 효율적 액세스
트랜잭션 처리	무결성, 회복, 동시성제어	원자적 트랜잭션, 장애 및 동기화 문제해결이 매우 어려움
비밀 및 보안	DBA가 중앙 통제 역할을 하여 정당한 액세스만 허용	지역적 비밀자치 보안처리

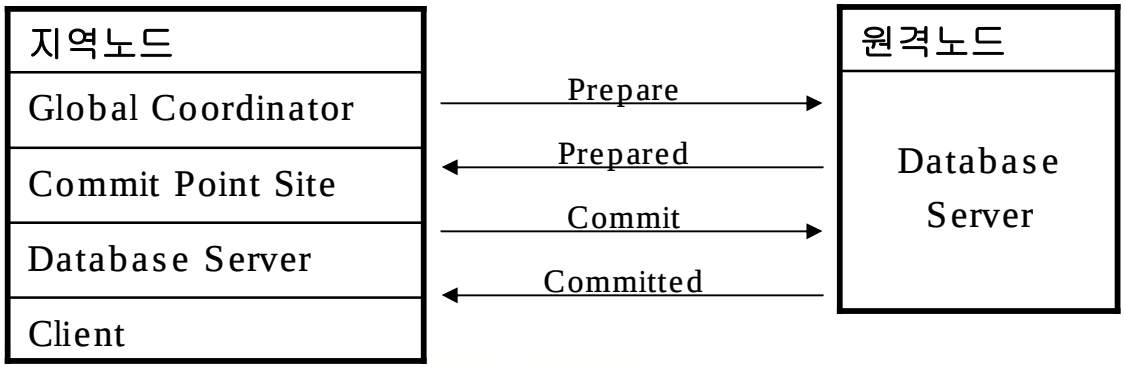
1. 2PC (Phase Commit)에 대한 이해
   가. 2PC의 정의
      - 분산 트랜잭션의 원자성을 보장하기 위해 분산 트랜잭션에 관여하는 모든 노드가 Commit 하거나 Rollback 하는 메커니즘
   나. 2PC의 필요성
      - 분산 데이터베이스 환경에서는 Commit과 Rollback만으로 여러 지역에 분산된 데이터베이스의 일관성이 보장되지 않음
      - 분산 데이터베이스에서는 모든 지역의 데이터베이스에서 트랜잭션이 성공 완료되었음을 확인한 후에 트랜잭션의 처리가 완료되어야 함
2. 2PC의 용어와 특징
   가. 2PC의 용어

용 어	주 요 개 념
서버 (Server)	- 원격 노드에서 요구하는 데이터를 가지고 있는 노드 - 조정자 또는 참여자임
조정자 (Global Coordinator)	- 분산 트랜잭션에 참여하는 참여자 목록을 가지며 - 분산 트랜잭션 및 Global Commit 을 시작하는 노드
참여자 (Participant)	- 분산 트랜잭션에서 지역(local) 트랜잭션을 수행하는 서버 - 조정자의 존재를 알고 그 결정을 따름
클라이언트 (Client)	- 분산 트랜잭션 응용을 실행하는 노드

나. 2PC의 특징

- 트랜잭션의 원자성을 보장하기 위한 수단
- 각 노드의 데이터 일관성을 보장하기 위해서 각 노드 마다 서로 협력
- 분산 데이터베이스에서 여러 단계를 거칠 수록 신뢰도는 증가하지만, 반대로 오버헤드도 또한 증가함.

3. 2PC의 수행 절차



가. Prepare Phase

- Global Coordinator가 분산 트랜잭션에 참여하는 노드들에 대하여 Prepare하도록 요청하는 단계
- 지역 노드의 Application에서 Commit Statement 발생
- Global Coordinator가 Commit Point Site를 결정
- Global Coordinator가 Prepare Message 전송하고 원격노드는 Prepared Message Reply

나. Commit Phase 단계

- 분산 트랜잭션에 참여하는 노드들이 Prepare되었다는 Message를 Global Coordinator에게 보내면 Coordinator는 트랜잭션 Commit요구
- Global Coordinator가 Commit Point Site에 Commit을 요청
- Commit Point Site는 Global Coordinator에게 Commit 여부 통보
- Global Coordinator는 원격 노드에 Commit Message 전송, 원격노드는 Commit Message Reply
- Global Coordinator는 Commit Point Site에 원격 노드의 Commit을 통보
- Commit Point Site는 분산 트랜잭션 정보를 삭제한 후 Global Coordinator에게 통보
- Global Coordinator는 트랜잭션에 대한 정보를 삭제한 후 트랜잭션을 종료

4. 2PC를 이용한 복제 분산 Database 구축 사례