

이기종 데이터베이스간 데이터 동기화의 성능 향상 기법의 설계 및 구현

허중건, 유헌창
고려대학교 컴퓨터정보통신대학원
e-mail : firstyanga@naver.com, yuhc@korea.ac.kr

Design and Implementation of Technique for Improving Performance of Data Synchronization between Heterogeneous DBs

Jonggun Heo, Heonchang Yu
Graduate School of Computer & Information Technology, Korea University

요 약

데이터들의 가치가 높아지고 대용량 데이터를 저장하기 위한 데이터베이스의 필요로 인해, 서로 다른 특징들을 가지는 다양한 유형의 대용량 데이터베이스가 많은 응용 분야에 적용되어 왔다. 그로 인해 다양한 종류의 데이터베이스에 대한 접근이 보다 쉬워짐으로 인해 이기종 데이터베이스간의 데이터 동기화가 필요하게 되었다. 본 연구에서는 이기종 데이터베이스간의 데이터 동기화의 성능 향상을 위한 기법을 제안하였고 데이터 동기화 속도, 다중처리, 그리고 정합성에서 기존의 기법보다 개선되었음을 보여 준다.

1. 서론

데이터들의 가치가 높아지고 대용량의 데이터를 저장할 위한 DB의 필요로 인해 각각의 특징들을 가지는 대용량 DB 종류가 많아졌다. 그로 인해 다양한 종류의 DB에 대한 접근이 보다 쉬워짐으로 인해 이기종 DB간의 데이터 동기화가 필요하게 되었다. 데이터 동기화란 각기 다른 환경의 DB에 동일한 데이터의 변경 사항을 업데이트 해주는 기술이다. 이기종간 데이터 동기화는 다양한 DB환경에서 데이터의 변경 사항을 보다 빠르고 정확하게 업데이트 해주고 데이터의 신뢰성을 유지시켜 주는 것을 일컫는다. 서로 다른 DB 환경에서의 DB자체만의 성능문제뿐만 아니라 네트워크 환경 서로 다른 자원 할당량을 가지고 있다. 또한 동기화된 데이터를 사용하는 방식과 형식에 있어서도 다른 구조를 가지고 있을 수 있기에 동기화 방식에 많은 제약을 가질 수 있다. 그로 인한 동기화 성능 및 데이터 신뢰에 대한 문제 및 동기화 성능에 대한 문제를 야기 시킬 수 있다. 다양한 환경에서의 데이터 동기화를 위한 소프트웨어 솔루션을 많이 개발되어 왔다. 하지만 각각의 DB환경에 따른 특성을 고려한 효율적인 동기화 구조는 이루어져 있지 않다. 따라서 본 논문에서는 다양한 DB환경에서 데이터 동기화를 위한 소프트웨어 솔루션의 구조변경을 통한 데이터 동기화의 성능 향상 기법을 제안한다. 이와 같이 기존 솔루션 구조의 변형을 통하여 데이터 동기화의 속도, 다중처리, 그리고 정합성에서 기존의 기법

보다 개선되었음을 보여 준다.

2. 관련연구

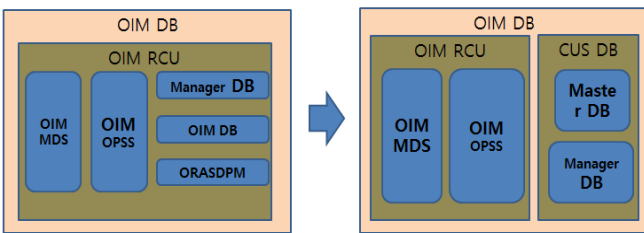
다양한 이기종 DB 가 늘어나고 혼재되어 사용되고 있으며 상용되고 있는 DB 동기화 기법들이 많이 연구되어 왔다. 로그나 타임스탬프 혹은 변경 이전 값을 저장해 놓고 비교하는 등의 DB 내 자체적인 기능을 이용하여 동기화 기법들이 있다. 기존의 방법들의 경우 DB 내부적인 기능의 종속성으로 인해 DB 환경 변화에 따른 대처가 어려울 수 있으며[1], 원본 테이블에 대한 복사 테이블 및 데이터 변경에 대한 스냅샷의 저장으로 인한 DB 저장공간의 부하 문제가 발생하며 데이터 변경에 대한 이력관리가 어려운 문제가 있다. 이러한 이 기종 DB 간 동기화 문제를 해결하기 위하여 동기화 소프트웨어 솔루션이 개발되어 왔다 [2]. 기존의 독립적인 에이전트 역할을 담당하던 웹 서버에 소프트웨어 솔루션을 탑재하여 기존의 장점들은 그대로 이용하며 데이터간 이력관리 및 각각의 독립적 프로세스를 이용하여 이 기종 DB 간 동기화 환경에 적용할 수 있으며 동기화 환경 구축이 쉽다는 장점이 있다. 또한 별도의 스냅샷 저장없이 데이터 변경에 대한 이력을 데이터 ROW 개념으로 저장하며 원본 테이블에 대한 복사 테이블의 생성이 필요하지 않아 DB 저장공간의 부하 문제도 해결할 수 있다. 본 논문에서는 이러한 소프트웨어 솔루션 중 오라클 DBMS 에서 데이터 동기화(계정관리) 솔루션의 구조를

분석하고 개선된 구조를 통한 성능 향상을 알아 본다

3. 제안하는 데이터 동기화의 성능 향상 기법

3.1. 내부테이블 구조 개선

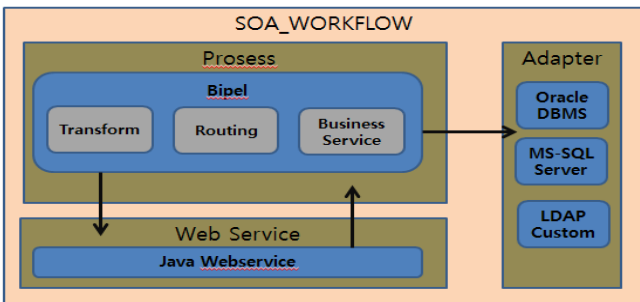
오라클 DBMS 에서 데이터 동기화(계정관리) 솔루션 특성은 내부 테이블만을 이용하여 데이터 동기화에 따른 모든 프로세스 및 사용자 정보, 결과 로그를 기록하며 참조한다. 하나의 데이터 동기화를 위해 동작 프로세스의 구분없이 스케줄러의 동작부터 동기화까지 너무 잦은 테이블 조회를 방지하고 데이터 비교 및 테이블 데이터 입출력을 없애기 위해 데이터 비교를 위한 데이터 원장 테이블과 이력을 위한 테이블, 커스텀 스케줄링 관리를 위한 별도의 테이블 등을 동기화 프로세스 별로 생성한다(그림 1).



(그림 1) 내부테이블 구조 변경

3.2. 동기화 프로세스 병렬화

데이터들을 동기화할 DB 에 전송하는 동작은 웹 로직에 SOA 워크플로우 프로세스를 사용한다. SOA 는 웹 로직을 이용하여 동기화할 목적에 맞게 웹 서비스를 호출하고 웹 서비스 내에 각각의 DB 드라이버를 포함하고 있다. 단일 SOA 워크플로우로 구성하면 동기화 대상이 복수 일시 하나의 동기화 대상의 네트워크 단절 혹은 DB 서버의 다운 등의 위험성에서 안전할 수 없다. 동기화할 대상과 목적에 따른 워크플로우를 각각 생성하여 각각의 동기화 목적에 맞게 프로세스를 운영한다(그림 2). 각각의 프로세스를 운영함으로써 데이터 동기화 속도를 한층 더 높일 수 있다. SOA 워크플로우를 사용함으로써 앞서 언급했던 이기종간의 DB 접속도 한층 더 수월하다.

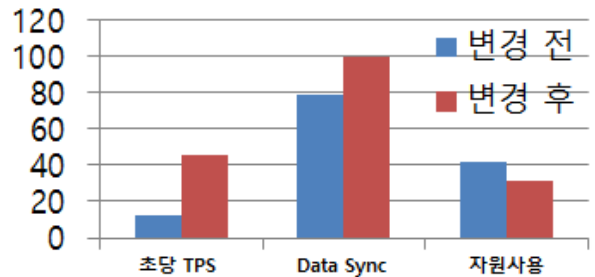


(그림 2) 병렬처리 SOA 워크플로우

4. 실행결과

제안한 구조변경으로 인해 데이터 동기화 솔루션의

성능을 비교 분석하여 솔루션의 성능향상 및 시스템 안정성을 보고자 한다. 테스트 환경은 각각의 16core CPU, 2GB 메모 환경에 Linux 6.5 Redhat 서버에 DB 환경을 구축한다. LDAP DB 와 오라클 DBMS 간 데이터 타입에 상관없이 10 만개의 동기화 요청에 따른 시간 측정(초당 TPS) 및 데이터 동기화 결과(Data Sync)를 확인하여 동기화의 정확성 및 성능 비교를 평가한다. 데이터 변경 후 스케줄러 동작부터 모든 데이터가 동기화를 완료한 시간까지의 시간을 측정하였다. 구조 변경 전은 데이터 동기화 중 특정 테이블의 DATA LOCK 으로 인해 데이터의 손실이 발생하였다. 또한 데이터유실로 인한 순차처리가 진행되지 않았다. 구조변경 후에는 테이블내의 데이터 LOCK 현상이 해소되어 데이터 안전성이 향상 되었으며 동기화 결과(Data Sync)측면에서도 100%에 가까운 결과를 보여주었다. 시간 별 측정 속도의 차이는 기존의 방식으로 는 초당 13 ~ 15 TPS 를 보였으나 구조 개선 후에는 42~45tps 의 초당 성능을 보인다. 내부 테이블의 구조변경으로 인해 테이블조회 및 무분별한 테이블 입출력의 최소화로 서버 자원사용량도 44%에서 33%로 줄어든 수치를 보였다.



(그림 3) 실행 결과

5. 결론

다양한 종류의 대용량 데이터베이스가 생겨나며 서로 다른 환경에서의 데이터 동기화 사례가 증가하고 있다. 이로 인해 데이터의 신뢰성 및 각각의 DB 를 사용하는 서비스 때문에 동기화 성능부분이 보다 더 중요하다. 각각의 환경이 다르기에 데이터 동기화에 대한 제약조건이 점점 더 까다로워지며 데이터 동기화 형식의 제약조건 및 DB 를 사용하는 서비스들의 환경에 맞는 데이터 가공으로 인해 동기화 성능을 유지하는 것이 점차 어려워진다. 그러므로 좀 더 유용한 구조 설계가 필요하다. 본 논문에서는 서로 다른 DB 환경에서의 동기화를 위한 솔루션 구조 변경을 제안하였다. 향후 연구에서는 언급하지 않은 데이터 정합성 및 실시간 동기화 처리에 대한 부분에 대한 연구를 진행하고자 한다.

참고문헌

- [1] Oracle, Oracle 9i Lite Administration Guide 5.0.1, Whitepaper
- [2] Oracle Identity Manager 11g Sizing Guide, Whitepaper