

로그 파일을 이용한 CDC 분석에 관한 연구

김영환*, 임영운*, 김상형**, 김 룡**, 최현호***

*알에프인터내셔널(주)

**충남대학교 컴퓨터공학과

***대전보건대학교 컴퓨터정보통신과

e-mail:james@TheRFI.com

A Study on CDC Analysis Using Redo-Log File

Young-Whan Kim* , Yeung-Woon Im* ,

Sang-Hyong Kim** , Kim Ryong** ,

Hyun-Ho Choi***

*RF International Co.,Ltd.

**Chungnam National Univ., Dept. of Computer Science & Engineering

***Daejeon Health Science College, Dept. of Computer Information

요 약

현재와 같이 처리해야 할 데이터가 폭주하는 상황에서 대부분의 시스템은 자료 저장을 위해 데이터베이스를 사용하지만, 누적되는 데이터 관리를 위해 빈번한 문제가 발생한다. 이때 대부분의 시스템들에서는 상용버전의 데이터 백업 시스템이나 이중화 시스템 등을 두어 여러 곳에 분산 배치함으로써 데이터 보관의 안전성을 도모한다. 실제 모든 데이터베이스 시스템들은 데이터를 레코드에 기록할 때 마다 고유의 로그기록을 남겨놓게 되어있다. 로그기록들은 결국 아카이브 형태로 저장되는데, 그전에 실시간으로 로그를 남기는 과정을 거치게 된다.

본 논문에서는 현재 많은 기관 및 단체에서 사용하는 오라클 데이터베이스를 기본으로 하여, 실시간으로 로그기록을 저장하게 되는 리두 로그(Redo-Log) 파일에 대하여 알아보고, 로그기록의 절차 및 응용 가능성에 대하여 보여준다.

1. 서론

시스템의 기능은 여러 가지가 있지만, 그중에서도 정보를 저장하고 유지하기 위한 데이터베이스에 대한 부분은 사물인터넷과 빅데이터 시대를 맞이하여 그 중요성이 다시금 부각되고 있다. 현재, 대부분의 상용 데이터베이스 시스템들은 백업, 이중화, 실시간 전달, 클라우드 저장 등을 이용하여 시스템을 구성하고 있다. 특히, 국내의 경우 대부분의 기관 및 단체의 데이터베이스는 오라클을 위주로 편성되어있고, 그 외에 IBM의 DB2, 오픈소스 계열 등으로 이루어져있다.

대표적인 상용 솔루션으로는 오라클의 골든게이트(Golden Gate)가 있으며, 이는 다양한 데이터 안전을 위한 기능을 제공한다. 이외에서도 여러 솔루션들이 존재하지만, 시장에서 사용되는 기본 시스템이 미미한 관계로 그 점유율은 높지 않은 상태다.

그러나 이러한 상용 솔루션들의 초기 구입비용이 상대적으로 높아 많은 기관들이 사용에 많은 제약이 있으며, 특히 교육기관, 소규모 단체, 기업은 사용 접근조차 하지 못하고 있다. 따라서, 컨설팅 업체에서는 이를 해소하기 위한 시스템의 이중화를 통한 데이터 복사 기능을 지원하는

자문을 해주고 있는데, 이것 역시 실시간으로 많은 데이터를 다루는 데에는 한계가 따른다.

본 논문에서는 오라클의 실시간 로그 정보를 이용하여 데이터를 어느 정도까지 안전하게 이중화할 수 있는지에 대한 기반 연구로서 그 활용 가능성을 제시하고자 한다.

본 논문의 구성은 서론에서는 데이터를 안전하게 보관하기 위한 다양한 CDC(Changed Data Capture)[1-6] 대하여 알아보고, 관련연구로는 안전한 데이터 이중화를 위해서 사용되는 솔루션들을 살펴 본다. 실제 리두 로그(Redo Log) 파일에 대한 분석을 통해 실험결과로서 로그기록의 활용성을 보이며 마지막으로 결론 및 향후 연구 내용을 제시하고자 한다.

2. 관련연구

(1) CDC(Changed Data Capture)

데이터베이스의 이중화를 위해서 여러 가지 데이터 복사 방법이 사용되는데, <표 1>에서와 같이 다양한 기법이 사용된다. 전통적인 데이터베이스의 쿼리 명령문을 이용하여 데이터를 복사하기도 하고, 데이터베이스에 접근하는 커넥터를 이용하기도 한다.

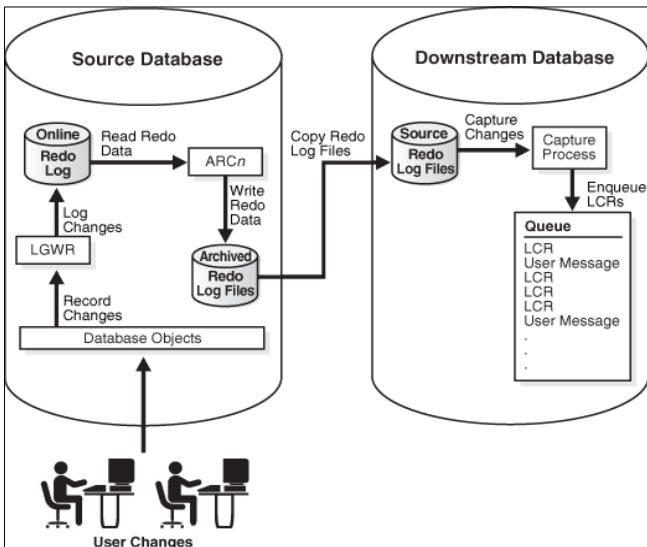
<표 1> CDC 구분

종 류
ODBC, JDBC(DB Connector)
SQL Query
Database Trigger
Table for Logging Change Data
Database Comparison
Database Log File

데이터베이스 시스템에는 고유의 데이터 복사 기능을 할 수 있는 방법들이 존재한다. 그러나 이러한 기능들은 시스템 관리자로 하여금 추가적인 작업을 요구하게 되고, 실시간성을 제공하지 않는다는 점에서 상용 솔루션을 채택하여 점차 자동화시키는 방향으로 나아가고 있다.

(2) 오라클의 골든게이트

자타가 공인하듯이 세계 최대의 데이터베이스 전문 기업으로 전용 솔루션을 이용한 데이터 이중화 방법을 제공한다. (그림 1)과 같이 기본 시스템의 로그인 리두 로드(Redo Log)와 아카이브 로그(Archive Log)를 이용하여, 실시간 또는 비실시간으로 원본 데이터베이스를 지정하는 시스템에 이중화 및 백업을 할 수 있도록 한다[7].



(그림 1) Oracle CDC 개념도

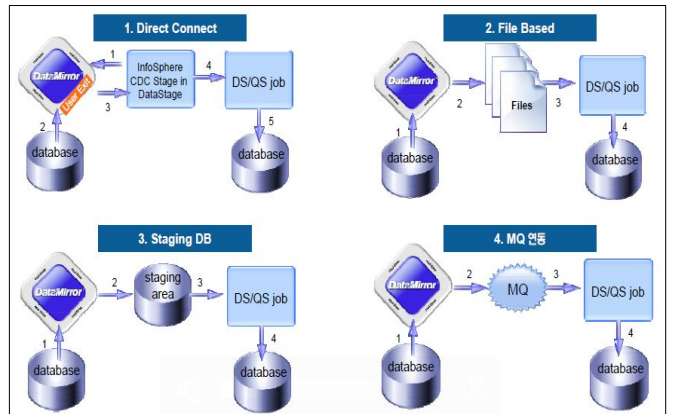
기본적으로, 오라클 자체의 시스템 파일은 로그파일에 대한 분석과 함께 제공하기 때문에, 현재 많은 시스템들에 설치되어있는 오라클 데이터베이스 시스템에 대한 솔루션으로 경쟁력을 가지고 있다.

(3) 아이비엠(IBM)의 CDC

자사의 DB2를 기본으로 다양한 제품의 데이터 복제를 지원한다. 실시간으로 데이터의 변경 내용을 추적하고, 변경사항을 해당 데이터가 필요한 지정된 시스템으로 전달 및 변환, 복제해주는 솔루션을 제공한다[8].

데이터의 변환, 복제 등의 기본 방식은 오라클의 골든

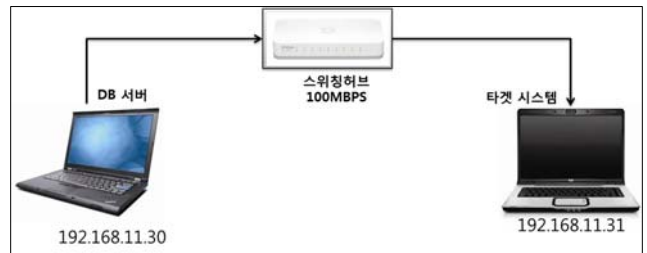
게이트 솔루션과 동일하며, (그림 2)과 같이 자체의 시스템 최적화 기술을 통한 통합솔루션 제공을 위주로 하고 있다.



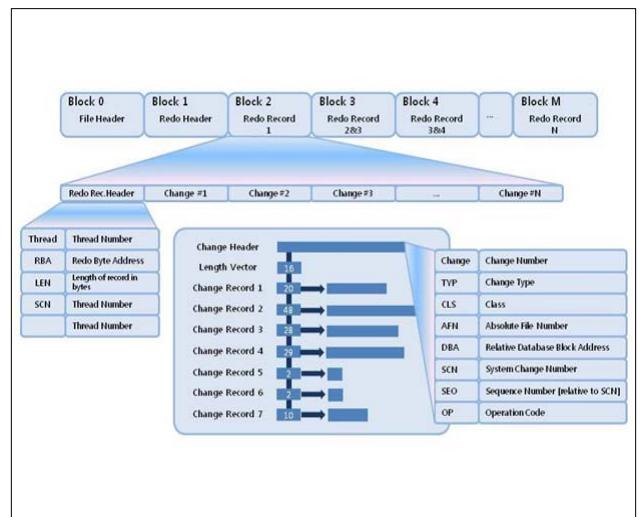
(그림 2) IBM CDC Solution

3. 실험 및 구조 분석

본 논문에서는 오라클 10g, 11g을 기본으로 한 오라클 데이터베이스 시스템의 리두 로그 파일을 기본으로 진행한다. 로그를 이용하여 (그림 3)과 같이 데이터를 원격지에도 동일하게 유지시키기 위한 방법으로 실험 환경을 구성하였다.



(그림 3) 실험 환경



(그림 4) 리두 로그 파일 구조

실시간으로 로그가 기록되기 위하여, 오라클에서는 중간 과정에 리두 로그(Redo Log)라 하여 실시간 로그를 임의의 파일에 기록하고(기본으로 세 개의 리두 로그파일 운

영) 첫 번째에서 두 번째, 세 번째 순서로 순차적으로 로그 저장이 이동되면서, 이전 로그를 아카이브 로그 (Archive Log) 파일로서 정하게 된다. <그림 4>는 리두 로그의 구조를 보여주고 있다[9].

전체 리두 로그 파일은 파일의 헤더와 함께 각 레코드 로써 구성되고, 각 레코드는 각 레코드에 대한 헤더 부분과 변경된 데이터를 위한 정보 부분으로 구성되어 진다.

리두 로그 파일은 기본적으로 이진파일의 형태로 저장되며 <표 2>와 같다. 여기에는 블록 크기 및 헤더 그리고 블록의 개수 등과 같은 아래의 정보가 포함된다.

<표 2> Redo Log File

Redo log sample file	
00 22 00 00 00 00 C0 FF 00 00 00 00 00 00 00	
67 C8 00 00 00 02 00 00 00 90 01 00 7D 7C 7B 7A	

```

u_char zero0 = 00 = 0x00
u_char type = 22 = 0x22
u_short zero1 = 00 00 = 0x0000
u_int_unknown_1[4] = {
    00 00 C0 FF, = 0xFFC00000
    00 00 00 00, = 0x00000000
    00 00 00 00, = 0x00000000
    67 C8 00 00 = 0x0000C867
}
u_int block_size = 00 02 00 00 = 0x00000200
u_int maximum_block = 00 90 01 00 = 0x00019000
u_char endian_magic[4]; = {7D, 7C, 7B, 7A}
    
```

u_char zero0	값이 0
u_char type	자료 불충분
u_short zero1	값이 0
u_int_unknown	-
u_int block_size	OS 블록 사이즈
u_int maximum_block	최대 블록 개수
u_char endian_magic[4]	헤더의 끝

<표 3> Redo Log File

Redo log sample file	
01 22 00 00 01 00 00 00 F5 00 00 00 00 80 40 43	

```

u_int block_type = 01 22 00 00 = 0x00002201
u_int minor = 01 00 00 00 = 0x00000001
u_int major = F5 00 00 00 = 0x000000F5
u_short micro = 00 80 = 0x8000
u_short crc = 40 43 = 0x4340
    
```

u_int block_type	블록 타입
u_int minor	몇 번째 블록 위치 값
u_int major	몇 번째 읽은 파일인지 알려주는 값
u_short micro	첫번째 블록의 새 레코드 헤더의 위치값
u_short crc	crc 오류체크 변수

<표 3>과 같이 리두 로그 파일의 내용은 다양한 정보로 이루어져 있으며, 이 리두 로그의 내용을 분석함으로써 실시간으로 데이터베이스 시스템의 변경사항을 확인할 수 있게 된다.

<표 4> Redo Log Dump

Redo log sample file(dump)	
REDO RECORD - Thread:1 RBA: 0x00002f.00000002.0010	
LEN: 0x06f4 VLD: 0x0d	
SCN: 0x0000.0002f36c SUBSCN: 1 11/29/2010 15:47:56	

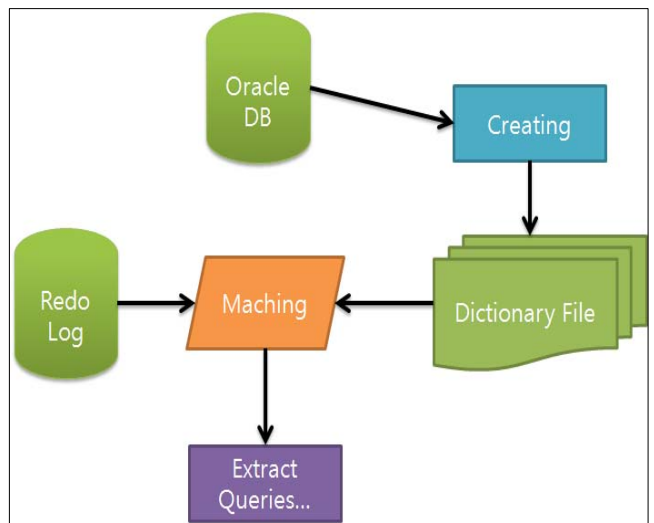
u_int len	redo record
u_char vld	see below
u_int scn_minor	record의 SCN minor
u_short scn_major	record의 SCN major
u_int epoch	record의 timestamp

<표 4>는 리두 로그 파일에 대하여 덤프를 한 것이다. 위 표에서 보면, 해당 파일에 대하여, 레코드의 크기 및 SCN에 대한 정보값, 그리고 타임 스탬프를 저장하고 있다. 이는 추후 오라클의 자체 유틸리티인 로그 마이너(Log Miner)를 이용할 때 복구를 위해 사용되는 정보이기도 하다.

4. 실험 결과

본 논문의 실험은 <그림 5>와 같이 리두 로그 파일을 분석 후 그것을 이용하여 데이터를 복제하는 과정을 끌어낼 수 있다. 리두 로그를 분석하고, 원본 데이터베이스의 디셔너리 정보를 추출하고, 이 두 가지의 정보를 비교하여, 최종적으로 데이터의 변경을 발생시킨 쿼리문을 추출한다.

추출된 쿼리문을 이용하여, 로컬이던 원격지의 서버이던 상관없이 변경된 데이터에 대한 동일한 데이터 변경을 유지할 수 있게 된다.



(그림 5) 로그 파일을 이용한 데이터 복제 방법

본 논문의 실험을 통해서 <표 5>와 같은 형태의 리두 로그 구조체를 정의하였다.

<표 5> 리두 로그 구조체

u_int compatibility_version; // DB Version
char db_name[8]; // DB Name
u_int file_size; // file header에 있는 maximum_block과 의미가 동일한 변수
u_int block_size; // file header에 있는 block_size와 의미가 동일한 변수
u_short file_number; // 실제 Redo log file 이름에 명시된 번호
u_int low_scn_minor; // Log file의 첫 번째 Redo Record의 SCN minor번호
u_short low_scn_major; // Log file의 첫 번째 Redo Record의 SCN major번호
u_int next_scn_minor; // 다음 Log file의 첫 번째 Redo Record의 SCN minor번호
u_short next_scn_major; // 다음 Log file의 첫 번째 Redo Record의 SCN major번호
//current인 Log file인 경우 이 값은 0xffff.ffffffff으로 세팅 됨

5. 결론 및 향후 연구 과제

본 논문에서는 데이터베이스 시스템의 데이터 복제를 위한 실시간 리두 로그 정보를 분석 실험하였다. 이를 통해서, 현재 대부분의 상용 데이터베이스 시스템을 차지하고 있는 오라클 데이터베이스 시스템의 로그 기록에 대한 구조를 정의하였으며, 이것을 이용한 다양한 로그 기반 솔루션에 대한 추가적인 연구를 위한 기반 기술을 확보하였다.

향후 연구 과제로는 본 논문에서 나온 결과를 바탕으로, 대규모 환경에서 로그파일의 직접 액세스를 통한 데이터 복제 및 이중화 그리고 백업과 같은 다양한 기능을 구현하는데 기반기술에 대한 추가 연구가 필요하며, 이를 통하여 빅데이터 및 클라우드 환경에서의 데이터에 대한 안전한 저장을 보장할 수 있는 방안에 대한 추가 연구가 필요하다.

참고문헌

[1] An Attunity White Paper, "Efficient and Real Time Data Integration With Change Data Capture"

[2] I Made Sukarsa, Ni Wayan Wisswani, I K. Gd. Darma Putra, Linawati, "Change Data Capture on OLTP Staging Area for Nearly Real Time Data Warehouse base on Database Trigger", International Journal of Computer Applications (0975 - 8887)

[3] "Incremental Updates vs Full Reload Change Data Capture" CONNX Solution, 2007

[4] Mitchell John Eccles, "Pragmatic Development of Service Based Real-Time Change Data Capture",

Aston University, April 2012

[5] Jonas T. Hansen, Stig Jrgensen, "A Real-Time Data Warehouse Solution for Analysis on Indoor Tracking Data", Department of Computer Science @ Aalborg University

[6] Mitchell J. Eccles, David J. Evans, Anthony J. Beaumont, "True Real-Time Change Data Capture with Web Service Database Encapsulation", IEEE Computer Society

[7] An Oracle® White Paper, "Data Integration Architectures for Operational Data Warehousing", Oracle Corporation, September 2012

[8] Awajeet Kumar Arya, "Change Data Capture for Sybase with a Remote Database Server", Technical White Paper On Implementing IBM® InfoSphere™

[9] <http://wiki.gurubee.net/pages/viewpage.action?pageId=9207913>