

# EAI에서 대용량 배치 데이터의 통합 성능 향상을 위한 Process-level 방식

김용희\*, 권주흠\*\*

\*고려대학교 컴퓨터과학기술대학원

\*\*고려대학교 컴퓨터학과

e-mail : yong0517@empal.com

## Process-level integration method for performance improvement of large scaled batch data processing in EAI environment

Yonghee Kim\*, Juhum Kwon\*\*

\*Dept. of Computer Science and Technology, Korea University

\*\*Dept. of Computer Science, Korea University

### 요 약

분산 시스템 환경에서 여러 시스템에 정보를 전송하기 위한 방법으로 최근 EAI의 DB Trigger 및 Redo Log 등을 이용한 실시간 데이터 통합 방식을 적용해 왔다. 그러나 기업에서 순차적인 배치 프로세스들을 통해 처리하는 대량의 데이터에 대해 기존의 EAI의 데이터 통합 방식을 사용할 경우 모든 변경 건수에 대해 이벤트가 발생하여 Source 시스템의 부하 및 통합 성능상의 문제점이 있다. 본 논문에서는 순차적인 배치 프로세스들을 EAI의 프로세스 레벨 통합을 적용하여 최종 변경된 데이터에 대해서만 통합하도록 하여 통합 처리 시간을 단축할 수 있는 방법을 제시하고자 한다.

### 1. 서론

온라인 업무 종료 후 배치 처리하는 대량의 데이터를 기업 내 분산된 여러 시스템에 전송하기 위해 전통적인 IT 환경에서는 주기적으로 FTP를 이용해 파일 형태로 전송하여 다시 저장하는 작업을 실행하였다.

이후 분산 환경에서 기존 애플리케이션에 변경을 최소화하면서 변경된 데이터에 대해 실시간으로 전송하기 위해 등장한 전사적 애플리케이션 통합(EAI : Enterprise Application Integration) 솔루션에서의 전통적인 데이터 레벨 통합 방식은 테이블의 변경 이벤트를 자동으로 감지하기 위한 방법으로 DB trigger를 사용하여 변경 내용을 임시 테이블로 저장하거나 redo log를 주기적으로 polling여 변경 분을 추출하여 각 시스템에 맞는 형식으로 변환하여 적용한다 [1, 2, 3, 4].

EAI Message Broker의 자동화된 기능을 이용한 개발 생산성 향상 및 유지보수 편의성 등으로 비용적인 효

과를 얻을 수 있으나, 순차적인 배치 프로세스들에 의해 동일한 DB에 반복하여 변경된 대용량 데이터의 경우 기존의 DB에 대한 event-driven 방식은 source 시스템 부하 및 통합 성능 문제를 발생시킨다.

EAI의 프로세스 레벨 통합 기능을 사용하여 순차적인 배치 프로세스들을 관리하고, Message Broker의 자동화된 데이터 변환 및 통합 기능을 활용해 최종 배치 프로세스 종료 후에 source DB의 최종적인 변경 부분만 여러 target 시스템에 적용하는 방법을 제안한다.

### 2. 관련 연구

#### 2.1 주기적인 FTP-based 파일 전송

EAI를 도입하기 전에는 source 시스템에서 온라인 업무 종료 후 배치 프로세스들을 순차적으로 실행하여 DB에 변경을 하고, 익일 업무 개시 전 또는 주

단위, 월 단위로 변경 분을 파일 형태로 저장하여 관련 target 시스템에 전송하였다.

이 경우 기업 내 여러 시스템간 DB 에 저장된 데이터 내용이 시간적인 차이에 의해 일치하지 않을 수 있고, 데이터 추출 및 파일 전송 등을 위해 직접 개발함으로써 개발생산성 저하 및 오류 복구 및 재적용 어려움으로 인한 비용문제가 발생할 수 있다.

## 2.2 EAI 의 event-driven DB 통합

비즈니스 이벤트와 적절한 조치 사이의 지연을 줄일 수 있는 주요 방법으로 EAI 의 실시간 정보 통합을 들 수 있다 [6]. EAI 의 경우 기존의 애플리케이션에 변경을 주지 않고 애플리케이션간 인터페이스 부분을 담당하여, 데이터의 변경이 발생한 시점에 자동으로 변경을 감지하여 실시간으로 관련 시스템들에 데이터를 적용할 수 있게 하는 방법으로 활용되고 있다.

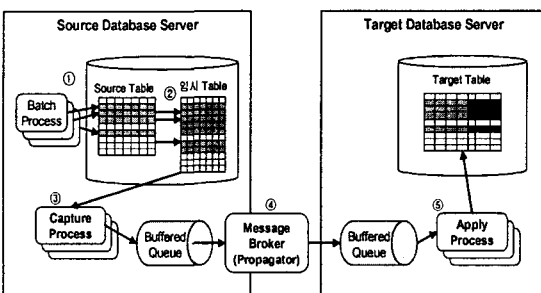
통신사의 고객 정보에 대한 배치 처리 프로세스를 예로 들어보자.

- Batch Process 1 : 고객 정보 변경 요청사항 적용
- Batch Process 2 : 서비스 요금 납부 사항 적용
- Batch Process 3 : 고객별 서비스 등급 조정

### 2.2.1 EAI : Trigger-based DB 통합

[그림 1]은 Trigger 기술을 사용하여 source 테이블의 변경을 감지하여 target 시스템에 적용하는 과정이다.

- ① 배치 프로세스에서 source 테이블 변경. 프로세스들이 동일한 row 를 중복하여 변경하는 경우가 발생할 수 있다.



[그림 1] Trigger-based DB 통합

- ② ①에서 SQL 문 실행 전후에 변경 내역이 임시 테이블에 저장된다. 여러 프로세스에서 동일한 row 를 변경한 경우에도 변경한 횟수만큼 임시 테이블에 저장된다. 이때 Capture Process 가 식별할 수 있는 Sequence ID 등을 추가한다.
- ③ Capture Process 는 주기적으로 임시테이블의 Sequence ID 를 검사하여 최종 반영한 부분 다음의 데이터를 읽어 Message Broker 에게 전달한다.

- ④ Message Broker 는 큐에 있는 데이터를 Target 에 맞는 형식으로 변환하여 각 Target 시스템의 큐로 전송한다.

- ⑤ Target 시스템에 대한 Apply Process 는 전송 받은 메시지를 Target 테이블에 적용한다.

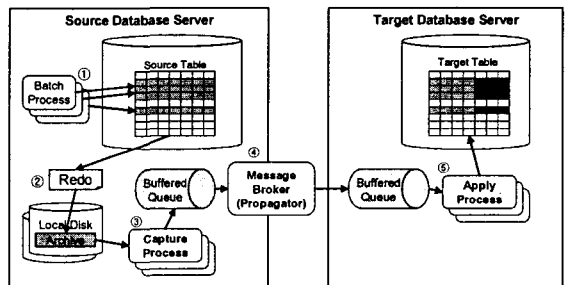
Trigger 방식은 DBMS 에 내장되어 source 시스템의 DB SQL 문 실행 전후에 자동으로 지정된 임시 테이블에 변경 데이터를 적용해 준다는 장점 [1, 2, 3]이 있지만, 실질적으로 애플리케이션 실행 중에 한 개의 SQL 문에 대해 두 번의 query 가 발생하여 source 시스템의 성능 저하 발생 [4], EAI message Broker 입장에서는 source DB 에 발생한 삽입/변경/삭제 모든 건별로 메시지 변환 및 전송을 하여야 하기 때문에 순차적인 여러 개의 배치 프로세스에 의해 중복 적용된 대량의 데이터 변환에 대해서는 관련 시스템들간 데이터 통합의 성능이 크게 떨어진다는 단점이 있다.

또한 DBMS 에 따라 Trigger 가 지원되지 않는 경우가 있기 때문에 모든 DB 환경에 동일하게 적용할 수 없다는 단점이 있다.

### 2.2.2 EAI : Log-based 데이터 레벨 통합

Trigger 의 특성 상 발생하는 source 시스템의 성능 저하를 해결하기 위해 최근 Oracle 9iR2 와 10g 버전에서 대용량 정보 공유를 위한 솔루션으로 Oracle Streams 를 소개하고 있다 [3, 5]. 이 솔루션은 Source 데이터의 변경 분을 얻기 위해 Trigger 대신 과거에 소개된 Log 기능을 이용하여 Redo log / Archive 파일을 읽어 원하는 테이블, 원하는 스키마 혹은 전체 DB 의 변경을 추출하여 원격지 DB 로 전달 적용한다 [3].

[그림 2]는 Log-based 데이터 통합 과정이다. 이 방식은 [그림 1]의 Trigger 방식 중에 ②와 ③ 부분에 차이가 있다.



[그림 2] Log-based DB 통합

- ② Source 테이블의 모든 변경 내역이 주기적으로 Redo 프로세스에 의해 Local Disk 에 저장된다.
- ③ Capture Process 는 주기적으로 Local Disk 의 log 파일 또는 Archive 파일을 읽어 Message Broker 에게 전달한다.

Trigger 대신 Redo Log 를 사용하기 때문에 Source

에플리케이션에 주는 성능 상의 문제를 일부 해결한다는 장점이 있지만, 이 방법 역시 source DB에 발생한 삽입/변경/삭제 모든 건 별로 메시지 변환 및 전송을 하여야 하기 때문에 순차적인 여러 개의 배치 프로세스에 의해 중복 적용된 대량의 데이터 변환에 대해서는 관련 시스템들간 데이터 통합의 성능이 크게 떨어진다는 단점을 그대로 가지고 있다.

또한 Oracle 9iR2, 10g 이상의 버전과 같이 솔루션으로 제공되는 경우가 아니면 직접 Log 파일을 읽어 처리하는 프로그램을 개발해야 하는 추가 요건이 발생하게 된다.

### 3. EAI에서 대용량 배치 처리 데이터의 프로세스 레벨 통합

본 장에서는 대용량 배치 처리 데이터 통합을 위해 순차적인 배치 프로세스들에 대한 프로세스 레벨 통합을 제안한다.

#### 3.1 특징

제안하는 프로세스 레벨 통합 방식은 기존의 방법과 비교할 때 다음과 같은 특징을 갖고 있다.

- DBMS에 따라 지원되지 않거나 지원 방식이 달라 사용 가능성 및 방법이 일관성이 없던 Trigger나 Log를 사용하지 않고 SQL만으로 모든 DBMS에 적용할 수 있으며, 기존의 Trigger 방식이나 Log 방식에서 사용하던 임시 테이블 또는 Log 파일이 필요하지 않게 된다.
- 대신, 기존의 Source 테이블에 mark 필드를 추가하고, EAI의 Capture Process에서 access할 수 있는 권한 부여해주어야 하지만 이는 제안 방안을 적용하기 전에 한번의 명령으로 해결할 수 있으므로 기존 방식과 비교해 매우 간단한 부분이라 할 수 있다.
- 기존 EAI의 Event-Driven DB 통합방식과 달리, 배치 프로세스의 SQL 실행하는 부분을 변경해 주어야 한다. 즉 Source 테이블을 변경하는 SQL 실행시 mark 필드를 set하도록 application의 일부를 수정하는 작업 필요하지만, 이는 한 줄의 코딩으로 해결할 수 있는 적은 부분이다.
- 순차적으로 처리되는 배치 프로세스들을 EAI의 BPM tool을 통해 관리할 수 있고, 맨 마지막 프로세스 다음에 Capture Process를 자동으로 기동하므로 배치 처리가 종료되는 최종 단계에서는 실시간으로 최종 데이터를 target 시스템에도 적용할 수 있게 된다.

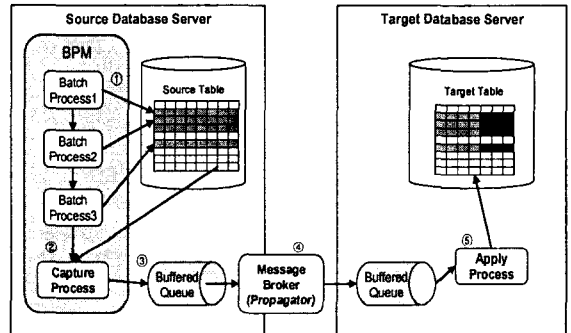
#### 3.2 통합 과정

프로세스 레벨 통합을 위한 과정은 [그림 3]과 같은 흐름을 통해 구현된다.

- ① EAI의 BPM 툴을 이용해 순차적으로 배치 프로세

스를 실행시킨다. 배치 프로세스들은 source 테이블에 삽입/변경하는 경우 mark 필드에 'I'/'U'를 set한다.

기존 방식에서는 각 배치 프로세스에서 테이블 변경 시에 변경된 row 별로 임시테이블 또는 Log 파일에 저장하게 되어 배치 프로세스들의 수행 시간이 길어지는 단점이 있었으나, 본 제안 방식에서는 배치 프로세스는 자신의 테이블만 변경하면 되기 때문에 배치 프로세스의 수행 시간이 짧아진다. 또한 BPM 툴을 이용해 중앙에서 전체시스템들의 통합 과정을 관리 함으로써 프로세스 진행 중에 오류가 발생할 경우 오류 처리과정을 추가할 수 있고, 담당자에게 Alert 하는 기능을 추가할 수 있게 된다.



[그림 3] 프로세스 레벨 DB 통합

- ② BPM 툴은 맨 마지막 배치 프로세스의 정상 완료 시 EAI의 Capture Process를 activate 시킨다.

- ③ Capture Process는 source 테이블의 mark 필드를 검사하여 변경된 부분을 큐에 넣고 Source 테이블의 mark 필드는 'F'로 set함

이 단계가 기존 방식에 비해 성능을 향상시킬 수 있는 부분이다. 기존 방식은 배치 프로세스에서 발생한 모든 삽입/변경/삭제 건 별로 변경 데이터를 capture 하기 때문에 대용량 데이터에 대해 프로세스에서 중복 변경할 경우 성능 저하가 발생하게 된다.

제안 방식에서는 연관된 배치 프로세스의 실행 후 최종적으로 source 테이블에 변경이 발생한 row 수 만큼만 데이터를 capture 하여 적용하기 때문에 한 테이블에 중복 변경이 많은 대용량 배치 처리에서 성능향상 효과를 볼 수 있게 된다.

- ④ Message Broker는 큐에 있는 데이터를 Target에 맞는 형식으로 변환하여 각 Target 시스템의 큐로 전송한다.

- ⑤ Target 시스템에 대한 Apply Process는 전송 받은 메시지를 Target 테이블에 적용한다.

- ④, ⑤의 단계는 기존 EAI의 Message Broker의 전송 보장 기능을 그대로 사용한다.

4. 평가

4.1 실험 환경

실험환경은 Windows 2000 Server 에서 DBMS 는 Oracle 9i 를 사용하였고, 테스트 프로그램은 C 와 Pro-C 로 구현하였다. 배치 프로세스 실행은 본 실험 BPM 들의 여러 기능을 테스트하는 목적이 아니므로 test.bat 파일을 실행하여 배치 프로세스를 순차적으로 호출하는 방식을 사용하였다.

Source 테이블에 100,000 건의 데이터에 대한 변경의 경우만 가정하고 각각의 배치 프로세스는 50,000 건을 UPDATE 하며, 제안 방식의 비교를 위해 최종 변경되는 row 의 수를 50,000 건 / 70,000 건 / 100,000 건 변경하는 경우로 테스트하였다.

또한 EAI Message Broker 는 초당 100 건을 처리하는 것으로 가정하였다.

4.2 비교 평가

각 프로세스의 DB 처리 시간은 SQL 문의 복잡도 및 비즈니스 로직에 따라 달라질 수 있으나, 본 실험에서 비교한 결과는 [표 1]과 같다.

[표 1] Trigger 방식과 프로세스 레벨 통합 방식 비교

처리시간 (초)	Trigger 방식	제안 방식 (프로세스 레벨 통합)		
		최종변경 50,000 건	최종변경 70,000 건	최종변경 100,000 건
Batch Proc.1	56	26	26	26
Batch Proc. 2	81	39	39	39
Batch Proc. 3	48	12	12	12
Msg. Broker	1,500	500	700	1,000
<b>Elapsed Time (최종소요)</b>	<b>1,505</b>	<b>577</b>	<b>777</b>	<b>1,077</b>

[표 1]에서 Trigger 방식의 경우 각각의 배치 프로세스에서 SQL 실행 시 Trigger 기동으로 인해 각 배치 프로세스의 처리 시간은 길어지지만 Message Broker 는 첫 번째 배치 프로세스의 첫 SQL 실행 직후 Event 방식으로 처리를 시작하게 된다는 장점이 있는 반면, 배치 프로세스들이 각각 50,000 건씩 변경을 하면 임시 테이블에 150,000 건이 저장되게 되어 결국 최종 소요되는 시간은 Message Broker 의 150,000 건 처리 성능에 좌우되게 된다.

순차적인 배치 프로세스에 의한 대용량 데이터의 통합 성능은 프로세스 레벨의 통합을 통해 최종 변경된 데이터만 추출하여 전달하게 되므로 여러 프로세스에 의해 중복 처리되는 건수가 많을수록 성능 향상을 볼 수 있다.

특히 모든 배치 프로세스들이 똑 같은 데이터를 변경하여 50,000 건에 대해서만 Message Broker 가 처리하

게 되는 경우 각 배치 프로세스의 처리 이후에 기동되더라도 가시적인 통합 성능 향상을 볼 수 있다.

5. 결론 및 향후 연구

본 논문에서는 순차적인 배치 처리 프로세스들에 의한 대용량 데이터 변경에 대해 EAI 의 프로세스 통합 기술을 적용하여 동일 데이터를 변경하는 배치 프로세스들을 그룹으로 관리하고, 최종 배치 프로세스의 완료 후 EAI 의 데이터 Capture 프로세스는 변경된 최종 데이터만 추출하여 적용할 수 있게 함으로써 통합 성능상의 효과를 높일 수 있는 방안을 제시하였다.

향후 연구에서는 삭제의 경우에 대한 처리 방안이 필요하다. 또한 기업에서 현실적으로 통합 프로젝트를 추진할 경우, 배치 프로세스에서의 데이터 변경과 관련된 로직을 모두 EAI 의 Capture Process 로 대체하여 개발하는 것과의 개발생산성, 통합 성능 또는 향후 유지 보수 등 조건에 따른 효율성을 비교하는 것도 필요하다.

참고문헌

- [1] Maya Lincoln, "Seamless Expansion of DB Servers for Advanced Telecom Services," <http://dsl.cs.technion.ac.il>, 2003
- [2] San Diego Supercomputer Center, "Distributed Object Computation Testbed (DOCT) Technical Report Task Group C - Develop Archival Storage and Retrieval Systems," <http://www.sdsc.edu/DOCT/Publications/c3-1c3-2/c3-1c3-2.html>
- [3] 김형일, "Oracle Streams 9i R2, 10g (Concept and Best Practices)," 한국 Oracle, 2004년 1월 6일
- [4] <http://www.experts-exchange.com/> (Search for 'Trigger Optimize')
- [5] Steve McClure, "Oracle's Solution for Heterogeneous Data Integration," IDC White Paper, Aug. 2003
- [6] Florian Melchert and Robert Winter, "The Enabling Role of Information Technology for Business Performance Management," Proc. of the 2004 IFIP International Conference on Decision Support Systems (DSS2004), Prato, Italy, 06.07.2004, 2004, pp. 535-546