

# SQL Server 를 실행하는 컴퓨터 간에 데이터베이스를 이동하는 방법에 관한 연구

용환전\*, 조정수, 김응모  
\*성균관대학교 정보통신공학부  
e-mail : [hjyong7@hotmail.com](mailto:hjyong7@hotmail.com)

## How to move database between computers that are running SQL Server

Hwanjeon Yong\*, Kyung-Soo Cho, Ung-mo Kim  
\*School of Information and Communication Engineering,  
Sungkyunkwan University

### 요 약

시스템을 구축하여 운용하다 보면 하드웨어 또는 DBMS 업그레이드를 위해 데이터베이스를 이동할 필요가 있게 된다. 그 상황은 동일 OS 동일 DBMS 로 이동, 스토리지 환경의 변화에 의한 이동, 상이한 OS 동일한 DBMS 로 이동, 상이한 OS 상이한 DBMS 로 이동 등이 있다. 본 연구에서는 동일 OS 동일 DBMS 로 이동 상황 즉 Windows Server, SQL Server 환경에서 하드웨어 업그레이드 및 SQL Server 및 OS 버전 업그레이드를 하는 방법에 국한하여 실제 시뮬레이션을 통해 장단점 및 서비스 중지 시간을 확인해보려 한다.

### 1. 서론

본 연구에서 기술하는 내용은 금융정보를 제공하는 B 업체가 SQL Server 2000 에서 SQL Server 2008 로 업그레이드 하기 위해 시뮬레이션 한 내용을 바탕으로 기술하였다.

금융정보를 제공하는 업체의 특성상 다양한 경로를 통해 세계 각국의 금융정보를 수신하여 데이터베이스를 저장하게 된다. 그래서 데이터베이스를 이동하면서 발생하는 서비스 중단시간에 민감하다.

중단시간에는 이동 시간 뿐 아니라 이동 후에도 서비스가 정상적으로 되고 있는지 테스트를 해야 하기 때문에 테스트 시간도 감안해야 한다.

그런데 데이터베이스 이동은 어떤 방법을 사용하는가에 따라 그 수행 속도는 천차만별이다. 이러한 사실을 대부분 외면하는 경우가 많다. 하지만 동일한 데이터에 대해 어떤 데이터베이스 이동방법을 사용하면 중단시간이 5 시간이 되고, 다른 방법을 사용하면 30 분이 소요될 수 있다.

시간이 많이 소요되는 작업에 대해서 서비스 중지 시간 등을 고려하기 위해 작업 시간을 산정하게 된다.

예정된 시간 안에 모든 작업을 수행해야 한다. 그래서 작업시간 산정은 중요하다. 실제로 작업을 수행하다 보면 예기치 않은 상황이 발생할 수도 있으므로 그에 대비한 시간을 감안하는 것도 중요하지만 사전에 충분한 시뮬레이션으로 방법을 숙지하여 발생할 수 있는 많은 상황을 파악하여 예방할 필요가 있다.

그래서 본 연구에서는 몇 가지 방법을 통해 중지시간을 측정해 보고 장단점을 파악해 각 업체별 상황에 맞는 최적의 방법을 선택할 수 있는 지침을 제공하고자 한다.

논문의 구성은 다음과 같다. 2 장에서 관련연구에 대해서 기술하고, 3 장에서는 데이터베이스 이동 시뮬레이션 결과를 설명하고, 4 장에서 향후 연구 방향에 대해 제시하며 결론을 맺는다.

### 2. 관련연구

#### 2.1 마이그레이션(migration)의 정의

윈도 95에서 윈도 98로 업그레이드하거나, 윈도 NT 서버에서 윈도 2000 서버로 옮겨가는 것과 같이 하나의 운영체제로부터 더 나은 다른 운영체제로 옮겨가는 과정을 말한다. 크게는 윈도 NT 에서 유닉스를 기반으

로 하는 운영체제로 옮겨가는 과정이나 그 반대 과정, 새로운 하드웨어나 소프트웨어 또는 둘 다 바뀌는 환경으로 이주하는 것, 데이터를 하나의 저장장치에서 다른 저장장치로 옮기는 과정 등을 포함한다.

이전의 운영체제에서 설정된 프로그램을 바꿀 필요 없이 현재의 응용프로그램을 새로운 환경에서도 계속 운영할 수 있고, 새로운 운영체제만이 가지고 있는 특성들을 이용할 수 있을 때 흔히 행해진다. 단일 시스템에서 다른 단일 시스템으로 옮겨가는 소규모 마이그레이션[1]에서부터 많은 시스템들이 새로운 응용 프로그램이나 네트워크로 옮겨가는 대규모 마이그레이션에 이르기까지 규모나 형식도 다양하다.

그러나 데이터베이스의 경우 새로운 데이터베이스가 이전의 데이터베이스와 구성 요소가 다를 수도 있기 때문에 실행 파일들을 처리할 수 있는 프로그램이 필요한 경우도 있다. 따라서 이전의 데이터베이스를 마이그레이션할 때는 새로운 데이터베이스와 공통된 형식으로 데이터를 변환하는 작업이 필요하다.

### 3. 데이터베이스 이동 시뮬레이션

#### 3.1 원본 및 서버 환경

원본 서버 환경

- Windows Server 2003 R2 Enterprise Edition SP2
- Intel(R) Xeon 3.00GHz CPU \* 2 개
- PC2-3200R DDR2 8GB
- SQL Server 2000 Enterprise Edition SP2
- Storage : RAID 1+0 구성

데이터베이스 용량

Filename	Total(MB)	Used(MB)	UnUsed(MB)
Main.mdf	198,737	174,192	24,545
Second.ndf	105,299	45,542	59,756

대상 서버 환경

- Windows Server 2008 R2 Enterprise Edition SP2
- Intel(R) Xeon 2.30GHz CPU \* 4 개(QuadCore)
- PC3-10600R DDR3-1333 64GB
- SQL Server 2008 Enterprise Edition
- Storage : RAID 1+0 구성

#### 3.2 백업 및 복원에 의한 방법

- 1) 데이터베이스 전체백업[2]을 실행한다.  
(2 시간 26 분)

- 2) 백업파일 대상서버에 복사한다.(3 시간 43 분)

- 3) 대상서버에서 백업파일을 이용하여 복원한다.  
(2 시간 30 분)

- 4) 트랜잭션 로그백업을 실행한다.

- 5) 트랜잭션 로그 백업파일을 대상서버에 복사한다.

- 6) 트랜잭션 로그 백업파일을 이용하여 복원한다.

※ Downtime = 트랜잭션 로그 백업시간 + 트랜잭션 로그파일 대상서버 복사 시간 + 트랜잭션 로그파일 복원 시간

※ 본 연구에서 수행 시간(Downtime) : 25 분

[장점]

- 전체 백업하는 동안 데이터베이스를 사용할 수 있다.

- 최소 시스템 중단 시간(Downtime)으로 이동 가능하다.

[단점]

- 원본 서버의 데이터베이스의 크기와 대상서버의 데이터 베이스의 크기가 같다. 데이터베이스 크기를 줄이려면 수행하기 전에 원본 데이터베이스를 줄이거나 복원을 완료한 후 대상 데이터베이스를 줄여야 한다.

- SQL Server 2000 의 경우 백업을 받는 동안 운영에 영향을 받을 수 있다. 트랜잭션이 가장 작은 시간 대에 작업을 할 필요가 있다.

- 전체 백업 및 트랜잭션로그 백업 두 번에 걸쳐서 수행해야 한다.

- 문자집합, 정렬순서, 유니코드 데이터 정렬이 동일해야 한다.

#### 3.3 sp\_detach\_db 및 sp\_attach\_db 에 의한 방법

- 1) sp\_detach\_db 저장 프로시저[2]를 사용하여 원본서버에서 데이터베이스를 분리한다.

- 2) 해당 데이터베이스 연관된 .mdf, .ndf, .ldf 파일을 대상 서버로 복사한다.

- 3) sp\_attach\_db 저장 프로시저[2]를 사용하여 복사한 파일을 대상서버에 데이터베이스를 연결한다.

※ 이 방법에서 Downtime 은 원본서버에서 대상서버로 복사시간이 대부분이다.

※ 본 연구에서 수행 시간(Downtime) : 5 시간 35 분

[장점]

- 작업이 간단하다. 어느 정도의 서비스 중단을 허용

할 수 있다면 이 방법으로 하는 것이 좋다.

- 오류의 가능성이 가장 작다.

[단점]

- 원본 서버의 데이터베이스의 크기와 대상서버의 데이터베이스의 크기가 같다.
- 시간이 오래 걸린다. 단, 디스크 및 네트워크 속도에 따라 단축될 수 있다.
- 문자집합, 정렬순서, 유니코드 데이터 정렬이 동일해야 한다.

### 3.4 데이터 가져오기 및 내보내기 방법

- 1) 대상서버에서 SQL Server Management Studio 를 연 다음 해당 데이터베이스 선택하고 태스크>데이터 가져오기[3]를 실행한다.
- 2) 데이터 원본을 선택한다.(서버 및 데이터베이스)
- 3) 대상을 선택한다.(서버 및 데이터베이스)
- 4) 개체를 복사할지 쿼리결과를 복사할지 결정한다.  
("테이블 또는 뷰에서 데이터 복사"를 선택)
- 5) 복사할 테이블 또는 뷰를 선택한다.  
(전체 테이블 및 뷰 선택)
- 6) 데이터 형식을 변경하여 복사할 필요가 있는 경우 변경하여 매핑한다.
- 7) 즉시 실행을 선택하여 실행한다.
- 8) 테이블 및 뷰를 제외한 저장프로시저, 함수, 유형 등은 원본서버에서 스크립트로 저장하여 대상서버에서 실행하여 생성한다.

※ Downtime = 가져오기 전체 시간

※ 본 연구에서 수행시간(Downtime) : 2 시간 23 분

[장점]

- 전송하는 동안 원본 데이터베이스를 사용할 수 있다.
- 문자집합, 정렬순서, 유니코드 데이터 정렬이 원본 서버와 동일해야 할 필요가 없다. 원본 데이터베이스의 사용되지 않는 공간은 이동되지 않으므로 대상 데이터베이스가 원본 데이터베이스만큼 클 필요가 없다.
- 필요 없는 개체를 제외하고 필요한 개체만 이동할 수 있다.
- 데이터 형식을 변경하여 매핑 할 수 있다.

[단점]

- 테이블 및 뷰만 이동된다. 기타 프로시저, 함수, 형

식 등은 별도로 스크립트를 작성해서 대상서버에 생성해야 한다.

### 3.5 작업, 경고 및 운영자를 이동하는 방법

- 1) 원본서버에서 SQL Server 엔터프라이즈 관리자를 연 다음 [관리] 폴더를 확장한다.
- 2) SQL Server 에이전트를 확장한 다음 [경고], [작업] 또는 [운영자]를 마우스 오른쪽 단추로 클릭하여 [모든작업]을 선택하여 [SQL 스크립트 생성]을 실행하여 스크립트를 생성한다.
- 3) 생성된 스크립트를 복사하여 대상서버에서 실행하여 [경고], [작업] 또는 [운영자]를 생성한다.

## 4. 결론 및 향후 연구 과제

각 업체별로 상황은 다양각색일 것이다. 그러므로 서비스 중지시간을 최소화 하여 이동하고자 하는 업체는 데이터베이스 백업, 복원에 의한 방법으로 진행하여야 할 것이며, 심야시간이나 휴일에 서비스를 중지하고 이동 작업을 진행할 수 있는 회사는 sp\_detach\_db 및 sp\_attcah\_db 에 의한 방법을 선택하여 진행하는 것이 가장 안전하고 좋은 방법이라 할 수 있겠다. 그리고 데이터베이스 일부 개체에 대해서만 이동을 해야 하는 경우나 이동하면서 데이터 형식을 변경할 필요가 있는 경우는 데이터 가져오기 및 내보내기 방법을 이용하여 데이터베이스 이동을 진행하면 된다. 데이터베이스 이동은 정확하고 신속해야 하고, 버전 업그레이드로 인해 실제 서비스에 문제가 없어야 한다. 그러므로 사전에 이동 시뮬레이션을 통해 방법을 철저히 숙지하고 이동 후 환경에서 충분한 테스트를 거친 후 실제 이동작업을 진행해야 한다.

본 연구에서는 SQL Server 의 버전 업그레이드로 인한 데이터베이스 이동에 대해 시뮬레이션을 진행하고 장단점을 살펴 보았다. 그러나 향후에는 현업에서 빈번하게 발생할 수 MySQL 에서 SQL Server, MySQL 에서 Oracle, SQL Server 에서 Oracle, Oracle 에서 SQL Server 이동에 대한 이슈를 파악해 보고 실제 시뮬레이션을 통해 각 방법 별로 장단점을 파악해 보는 연구가 필요할 것이다. 더 나아가 단순히 데이터베이스 아키텍처의 변경 없이 이동하는 경우만이 아니라 데이터아키텍처를 새로 설계하여 데이터를 매핑 하여 이동하는 것에 대한 연구도 필요할 것이다. 그래서

데이터 이동에 대한 다양한 지침을 제공하는 것이 필요할 것이다.

#### 감사의 글

이 논문은 2009 년도 정부(교육과학기술부)의 재원으로 한국과학재단의 지원을 받아 수행된 연구임 (No.2009-0075771)

#### 참고문헌

- [1] 두산백과사전 EnCyber & EnCybler.com  
[http://www.encyber.com/search\\_w/ctdetail.php?gs=ws&gd=&cd=&d=&k=&inqr=&indme=&p=1&q=%B8%B6%C0%CC%B1%D7%B7%B9%C0%CC%BC%C7&masterno=771521&contentno=771521](http://www.encyber.com/search_w/ctdetail.php?gs=ws&gd=&cd=&d=&k=&inqr=&indme=&p=1&q=%B8%B6%C0%CC%B1%D7%B7%B9%C0%CC%BC%C7&masterno=771521&contentno=771521)
- [2] Robert Vieira 저/류광 편역, Professional SQL Server 2000 Programming, 정보문화사, 2001, pp.1286-1294, 1304
- [3] 우재남 저, 뇌를 자극하는 SQL Server 2008, 한빛미디어, 2009, PP.1015-1034