

# 데이터 전반의 재구축 능력 탁월

90년대부터 본격적으로 국내 기업에 적용되기 시작한 관계형 데이터베이스는, 현재 그 규모나 크기가 기하급수적으로 증가하고 있어, 도입 초기에 생각하지 못했던 부분들이 업무 진행의 큰 장애물로 등장하고 있다.

**데**이터베이스 관리와 관련되어 문제가 되고 있는 점을 나열하면 아래와 같다.

- 데이터의 증가 : 데이터를 정보 자산화 하기 위해 처리되는 데이터량의 급격한 증가로 대규모 데이터에 대한 백업 및 데이터베이스 재구축에 대한 어려움
- 백업의 어려움 : 오라클 관계형 데이터베이스 내부의 기능을 사용하여 데이터베이스를 Unload시킬 경우 GB의 데이터 백업에 막대한 시간을 초래
- 테이블스페이스 활용의 미흡 : 대규모 애플리케이션에서 다수의 사용자가 데이터베이스를 접근하다 보니 프래그멘테이션(fragmentation) 증가로 테이블스페이스를 효율적으로 활용하지 못함
- 속도 저하 : 인덱스, 테이블이 꼬여있어 데이터 조회에 대한 속도 저하가 일어난다.
- 기존의 데이터 재편성(data reorganization) 방법에 대한 불만 및 다량의 시간 소요

이상과 같은 관리, 사용상의 어려움을 해결하고자 플라티늄에서는 오랜기간 축적해온 기술을 바탕으로 백업 및 데이터베이스 재구축에 대한 실질적인 제품을 생산함으로써 IT관계자들에게 솔루션을

제공하였다.

즉, 일상의 백업에 대해서는 오라클의 기능을 사용하는 것보다 5-10배 정도 빠른 Fast Unload를 사용하여 백업을 담당하고, 데이터베이스에 대한 속도 및 데이터 구조를 정렬하기 위해 TSreorg를 사용하여 데이터베이스 최적화를 구현할 수 있게 하였다.

사용 중인 Active 데이터베이스에서 성능 저하(Performance Degrading) 현상이 발생할 수 있는데, 그 이유는 데이터베이스 내에 Space, Row fragmentation이 과도한 Disk I/O, 병목현상(Bottle-neck), 에러를 유발시키기 때문이다. 이에 대한 해결책은 손상된 구조를 재편성시키는 것이다.

기존의 데이터 재편성 방법은 데이터베이스 전체를 Rebuild 시키는 것이다. 그러나 의외로 이러한 절차를 진행하는데 있어 한눈에 전체적으로 파악하기가 어렵고 관리자가 친숙하지 못한 것이 일반적이다.

또한 처리 중인 Active 데이터베이스에는 액세스할 수 없으며, Rebuild 작업이 순조롭더라도 꽤 많



〈그림 1〉 데이터 재편성



은 시간을 필요로 한다. 매일 또는 일정 시점의 백업 작업시 데이터베이스를 Unload 받을 경우 데이터량이 많을수록 그 시간은 길어져서 작업을 진행하는데 있어 이에 대한 부담없이 가중되고 있고, 대량의 데이터 추출시 야간작업을 하더라도 일과시간 시작 전에 작업이 완료되지 못하는 한계성을 지니고 있다.

한편 데이터베이스 추출 및 Rebuild 과정에서 운영 담당자에 의해 발생하는 실수 및 하드웨어상의 장애 등도 치명적인 문제를 야기시킬 수 있고 이러한 문제는 데이터베이스가 커질 수록 발생할 가능성이 높다고 할 수 있다.

Fast Unload와 TSreorg 를 사용하면 위의 문제점을 해결하면서 다음과 같은 효과를 기대할 수 있다.

- 신속한 데이터베이스 추출로 생산성 및 경제성 보장
  - 다양한 포맷으로 데이터 산출
  - 선택적 옵션 추출로 사용자의 편리성 고려
  - 신속한 데이터 추출로 시간 절감
- 데이터베이스 구조에 대한 모니터링과 재편성을 통한 성능 향상
  - 현 상황을 정확히 모니터링함으로써 상태를 정확히 진단
  - 물리적 구조에 대한 최적화
  - 신뢰성 보장
- 대량의 데이터에 대한 백업 및 복구 대책의 확립으로 발생될 수 있는 각종 위험 최소화
- 백업장치와의 연계로 네트워크 백업 또는 HSM 등 다양한 저장 방법 구현
- 데이터베이스의 프래그멘테이션, 헝클어진 인덱스를 재정립함으로써 성능 극대화

### Fast Unload의 기능 및 특징

- 오라클 데이터베이스를 추출하기 위한 최상의 성능과 유연성을 제공

Fast Unload는 오라클 데이터베이스를 직접 접근할 수 있는 오라클 제품들 이외의 유일한 제품이

다. 즉, 오라클의 SQL 엔진과 커널을 완전히 우회하기 때문에, Fast Unload는 내부의 오라클 라이브러리와 SQL 콜(Call)의 오버헤드를 피할 수 있게 된다. 따라서 Fast Unload는 다른 툴들보다 5 내지 10배 정도 빠르게 데이터를 추출할 수 있으며, 데이터 추출 시간을 수시간 또는 수일씩 단축시킬 수 있다. 사용자는 향상된 성능과 효율적인 언로딩(Unloading)을 위해 Fast Unload를 사용할 수 있다.

- 다양한 포맷으로 데이터 산출을 통해 사용의 편의성 도모

Fast Unload는 산출된 데이터에 대한 다양한 포맷 옵션을 제공한다. 또한 산출된 자료들이 다양한 데이터베이스 로드(Load) 툴과 호환이 될 수 있도록 포맷할 수 있다.

- Oracle SQL Loader
- Sybase BCP
- Informix DBLoad
- Standard delimited

또한, 산출된 파일(Files)들을 최소한의 가공을 통해 다른 데이터베이스로 변환시킬 수 있다.

- Row와 Column을 선택적으로 산출함으로써 활용도를 높임

대부분의 어플리케이션들은 오라클 테이블의 모든 Row와 Column을 필요로 하지 않지만, 다른 언로드(Unload) 유틸리티들은 이러한 선택적인 기능을 제공하지 않는다. Fast Unload를 사용하면, 사용할 Row를 제한적으로 선택할 수 있을 뿐만 아니라, 원하는 Column들만을 선택할 수도 있다.

- 복수 개의 파일로 산출할 수 있는 기능 제공

Fast Unload를 사용하면, 복수 개의 파일로 언로드(Unload)시킬 수 있다. 또한, 2기가바이트(오늘날 대부분의 유닉스 시스템들이 제공하는 최대의 파일 시스템 크기)까지의 테이블을 언로드(Unload)시킬 수 있다. Fast Unload는 최적의 속도와 신뢰성을 제공하기 위해 라운드 로빈(Round-



robin) 방식을 통해 복수 개의 파일로 언로드(Unload)시킬 수 있다.

병렬 로딩(Parallel Loading)이 지원되는 곳에 서는 복수 개의 Load Path도 제공한다.

- 다른 플라티늄 제품들과 결합하여 효용 극대화
- 오라클로부터 데이터를 추출하기 위해 필요한 플라티늄 제품(TSreorg, NetArchive, InfoPump 및 AutoSys)들과 결합하여 성능과 유연성을 향상시킬 수 있다.

### TSreorg의 기능 및 특징

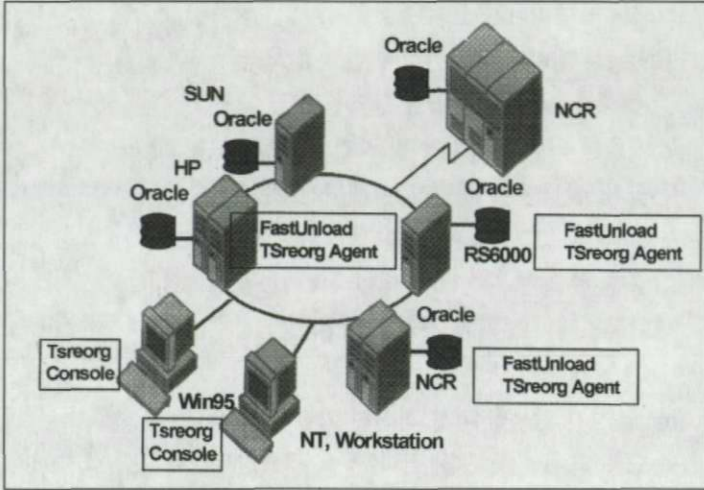
- Oracle을 위한 완벽한 Reorg 관리자
  - Oracle 테이블 공간을 위한 Reorg와 Defragmentation을 제공(개별적 테이블과 인덱스 등)
  - Fragment된 테이블 스페이스 원상회복
  - 공간 활용 및 테이블 스페이스 관리 최적화
- 데이터베이스 재구축시 시간 단축 및 위험 감소
  - 개별적 테이블스페이스, 테이블, 인덱스, Reorg 가능
  - 기존 방법으로는 데이터베이스가 약 2기가바이트의 5개 테이블스페이스 내에 10기가바이트의 데이터로 재구축시 1기가바이트가 프로세스 1시간이 걸린다면 시스템 10시간 동안 정지, TSreorg는 각 테이블스페이스, 테이블, 인덱스 단위로 수행하므로 시간 단축 및 효율적 관리
- 분산구조 환경에 적합한 Reorg 솔루션
  - 클라이언트 (운영자 콘솔)과 서버(Reorg 에이전트) 사이에 최적 구분 (연계운영과 독립운영이 가능)
  - 네트워크 통신량을 최소화 위한 TSreorg 영향력 유지(콘솔과 에이전트 독립)
  - 서버의 추가 및 Oracle 데이터베이스의 플랫폼 변경시 쉽게 구현
- SQL\*Net, Oracle 고유의 통신기술을 사용하

로 도의 유틸리티가 필요없음

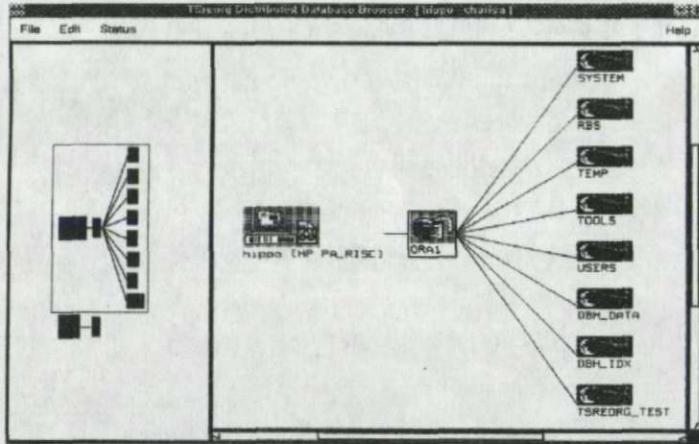
- Reorg시 데이터의 안전성과 신뢰성 보장
- 실패 예측 및 회복을 위한 강력한 기능 제공
- 스케줄에 의한 특정일자 및 시간에 수행 가능
- 콘솔과 에이전트 독립성은 에이전트는 콘솔이 고장난 후에도 기능이 계속되며, 에이전트가 고장나면 콘솔은 계속해서 데이터베이스를 읽고, 테이블스페이스 맵과 스케줄 작업 보호

- Reorg 시간 단축을 위한 최신의 기술 제공
  - 속도를 최우선시, SQL이나 롤백 세그먼트없이 데이터베이스에 직접 데이터 블럭을 작성하는 직접적인 로드 경로 선택
  - 대형의 테이블스페이스 Reorg시 Oracle은 8시간, TSreorg는 2시간
  - 언로드시 압축하는 옵션 제공과 로드시 압축을 해제함으로 공간 활용이 용이
  - 직접적인 OCI(Oracle Call Interface)호출, 어레이 삽입, 인출(fetch), 프로세싱을 효과적 이용
- 유저 인터페이스 및 테이블스페이스의 정보를 그래픽 화면으로 제공 작업능률 향상
  - 플랫폼에 상관없이 Motif, Open Look, OS/2 프리젠테이션 관리자 또는 윈도우즈 선택
  - 콘솔의 분산된 데이터베이스 브라우저는 연결된 호스트 기계, Oracle Instance, 사용자, 테이블스페이스를 아이콘화함으로써 사용이 용이
  - 테이블, 인덱스, 클러스터, 롤백 세그먼트, 테이블스페이스 내의 여유공간을 나타내는 Proportional Tablespace Map 제공으로 문제 해결이 용이
- Reorg 프로세스의 자동화
  - 특정 시간과 날짜로 한정할 수 있고, 상이한 테이블스페이스를 위한 분리된 타임 윈도우 정의
- Reorg 작업시 데이터를 압축
  - Date가 Unload되는 동안 압축되고, Load되는 동안 풀림

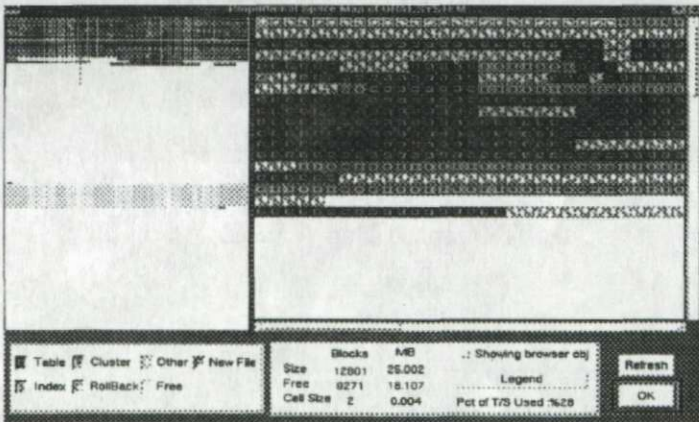




〈그림 2〉 TSreorg와 Fast Unload 구성



〈그림 3〉 데이터베이스 재구축



〈그림 4〉 데이터베이스 현황을 맵 형태로 분석

· 특징

- I/O 병목현상을 최소화 위한 다중 디스크 전역에서 무작위로 데이터를 분할
- 테이블 스페이스, Oracle Instance 또는 CPU

에 걸쳐 마우스 클릭과 그래픽 메뉴로서 테이블을 옮기고 복사.

- 테이블스페이스 데이터가 백업파일에 언로드된 후에 프로세싱을 일시 정지한다.
- 자동으로 또는 수동으로 저장 파라미터를 교정.
- 좀더 신속한 데이터 검색을 위하여 Row를 Recluster
- 공간 절약과 성능향상을 위한 칼럼 다시 정리
- Oracle이 어떤 관계된 명령을 제공하지 않을 때에도 테이블 구조 변경 가능

· 옵션

- 테이블 프로세싱을 커스터마이징하는 옵션
- 다중 디스크에 걸쳐있는 테이블 데이터를 쉽게 분할 및 재배치에 의한 성능향상
- 데이터 Export시 Row의 첨가 및 삭제, 수정이 GUI 환경에서 가능

TSreorg · Fast Unload의 구성 및 활용

엔터프라이즈 시스템 통합 관리와 최상의 성능을 위한 환경

TSreorg는 부하가 적고, 고도의 확장성을 가진 구조를 가지고 있다. TSreorg는 클라이언트 컴포넌트(콘솔)와 서버 컴포넌트(reorg agent)으로 구성되어 있다. 콘솔과 에이전트는 밀접한 관계를 맺고 있지만, 서로 다른 머신 또는 서로 다른 플랫폼에서 독립적으로 작동한다. 반면 Fast Unload는 제품의 특성상 한 서버에 모두 설치되어야 한다.

〈그림 2〉와 같이 구성된 환경하에서 사용자는 먼저 데이터베이스의 상황을 맵 형태로 볼 수 있으며, 재구축 결정이 나면 Fast Unload를 통해 먼저 데이터베이스를 Unload받아 이를 Storage Box에 NetArchive를 사용하여 백업을 받은 후 데이터베이스를 재구축하는 것이 보다 바람직하다.

이 사항을 요약하면 〈그림 3〉과 같다.

TSreorg를 사용하여 다수의 데이터베이스를 모니터링한다. 한 개의 콘솔에서 다수 기계에 있는 오라클 데이터베이스도 조회 가능함으로 구성의 확실성을 고려, 데이터베이스에 대한 현황을 맵 형태로



분석하여 인덱스 및 테이블 등이 최적화되지 못했을 때 이를 재구축하는 것으로 결정한다(그림 4, 5 참조).

먼저 Fast Unload를 사용하여 데이터베이스를 Unload하고 TSreorg를 사용하여 데이터베이스를 재구축한다. 콘솔에서 structure, scheduling reorg, 그리고 progress 상황을 모니터링하는 것을 위하여 그래픽적 화면 디스플레이가 제공된다.

유닉스, 윈도우즈95, 윈도우즈NT, 그리고 VMS 환경에서 작동된다. 에이전트는 모든 reorg 과정을 통제하는 프로그램이며 유닉스, VMS, 윈도우즈 NT 환경에서 수행된다.

스케일업(Scale up) 하기를 원할 때 콘솔 에이전트의 독립이 왜 중요한가? TSreorg의 impact가 네트워크의 트래픽을 최소화하기 때문이다. 콘솔과 에이전트 사이의 지속적인 연결이 되어있지 않을 시에는 TSreorg가 필요한 수 이상의 서버를 자동 지원한다.

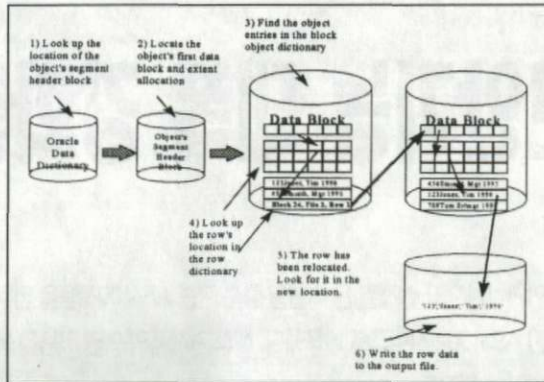
Reorg를 ad-hoc job 대신에 에이전트에 의해 자동으로 스케줄되어 진다면 네트워크 트래픽은 전혀 없다. TSreorg는 시스템을 확장할 때 추가적인 비싼 네트워크 대역폭의 비용을 줄이거나 또는 제거할 수 있게 한다.

또한 TSreorg는 또 다른 중요한 특성들을 제공한다. 서버가 추가될 때 에이전트는 쉽고 간단하게 설치되며 DBA가 원하는 대로 reorg에 대해 쉽게 정의할 수 있다.

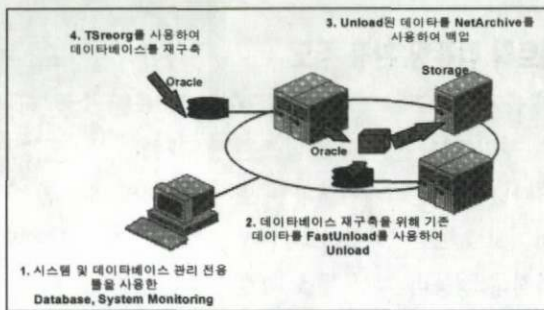
Oracle DB를 보다 강력한 플랫폼이나 다른 플랫폼으로 이식시키려 할 때 에이전트는 새로운 시스템에 적용시키기가 쉽다. 그 이유는 오라클 테이블을 기본으로 한 '잡 파라미터'를 사용하기 때문이다.

**시스템 관리, DB관리, 백업관리와의 상호 연동성**

Fast Unload와 TSreorg는 시스템 관리 측면에서는 시스템 저장 스페이스를 최대한 활용할 수 있다는 장점을 가지고 있고, 데이터베이스 관리 측면에서는 인덱스, 테이블 등 데이터베이스 전반에 걸



〈그림 5〉 데이터베이스 최적화를 위한 재구축



〈그림 6〉 통합적인 시스템 구축

쳐 상황을 모니터링하고 재구축함으로써 데이터베이스 접근에 대한 성능을 향상시키고, 대량의 데이터베이스의 경우 Fast Unload를 사용하여 Unload 받은 데이터를 플라티늄 제품인 NetArchive를 사용하여 백업을 받을 수 있도록 하여 통합적인 시스템 구축이 되도록 할 수 있다. 이를 도식화하면 〈그림 6〉과 같다. **DC**

〈참조 사이트〉

**TSreorg**

국민생명, 삼성SDS, 한국이동통신, 현대정보기술, 현대전자, 한진정보통신, 한국중공업, LG 전자

**Fast Unload**

상업은행, 주택은행, LG-EDS