
Goodus 기술노트 [37 회]

TimesTen In-Memory DB

Author	박덕우, 김동우
Creation Date	2009-01-04
Last Updated	2009-01-04
Version	1.0
Copyright(C) 2004 Goodus Inc. All Rights Reserved	

Version	변경일자	변경자(작성자)	주요내용
1	2009-01-04	박덕우, 김동우	문서 최초 작성
2			
3			

Contents

1. Oracle TimesTen In-Memory Database.....	3
1.1. Oracle TimesTen In-Memory Database Overview?	3
1.2. Oracle TimesTen의 구성	4
1.2.1. Oracle TimesTen In-Memory Database.....	4
1.2.2. Cache Connect to Oracle.....	6
1.2.3. Replication – TimesTen to TimesTen	7
2. Oracle TimesTen의 설치.....	8
2.1. 환경설정	8
2.2. TimesTen 설치.....	8
2.3. TimesTen 설치 제거.....	11
2.4. TimesTen 설치 확인.....	12
2.5. TimesTen DB 생성	13
2.5.1. TimesTen DB 생성	13
2.5.2. Cache Connect를 설정한 TimesTen DB 생성	14
2.6. Cache Connect to Oracle Concepts	16
2.6.1. TimesTen 과 Oracle 설정	16
2.6.2. Readonly Cache Group 생성.....	17
2.6.3. SQL Passthrough 방식으로 전환.....	21
2.7. TimesTen Replication (복제)	24
3. Oracle TimesTen의 관리.....	29
3.1. TimesTen 관리를 위한 주요 명령어.....	29
3.2. TimesTen 관리를 위한 ttlsql 주요 명령어	29
3.2.1. ttlsql 기본 명령어	29
3.2.2. Commit, Rollback, Sqlquerytimeout.....	32
3.2.3. Data Store 관련 정보 확인	32
3.2.4. Passthrough 설정 확인 및 변경.....	35
3.2.5. Query Optimizer 실행계획의 변경 및 확인.....	35
3.2.6. Timing ODBC function calls	38
4. Oracle TimesTen 백업 및 복구	39
4.1.1. TimesTen 백업 Test를 위한 Access control 활성화.....	39
4.1.2. TimesTen 백업 및 복구 예제.....	40
5. Tools for Troubleshooting TimesTen.....	41
5.1. ttStatus 유틸리티.....	41
5.2. ttCapture 유틸리티	43
5.3. ttTraceMon 유틸리티	44
5.3.1. SQL Trace 사용하기.....	44
5.3.2. LOCK Trace 사용하기	44
5.3.3. ERR Trace 사용하기	45
6. Oracle TimesTen 활용 사례.....	46
6.1. In-Memory Database의 일반적인 적용 분야.....	46
6.2. 제조 분야의 TimesTen 활용 사례.....	47
6.3. 금융 분야의 TimesTen 활용 사례.....	48
6.4. 통신 분야의 TimesTen 활용 사례.....	49
7. 참고 문헌 및 사이트.....	50

1. Oracle TimesTen In-Memory Database

1.1. Oracle TimesTen In-Memory Database Overview?

최근 몇 년동안 오라클은 활발한 전략적 기업 인수 작업을 수행해왔습니다. 그 중에서도 특히 주목할만한 것이 바로 기존 오라클 DB의 읽기, 쓰기 처리속도를 10배이상 향상시킬 수 있는 신개념 DB솔루션인 TimesTen In-Memory Database 입니다. 그 동안 In-Memory Database 분야는 디스크 기반의 DBMS의 부록과 같은 존재감이었으나, 오라클이 TimesTen을 전격적으로 인수하고, 인수 후 두달여 만에 오라클 제품에 맞춰 새 버전을 출시하고, 공격적인 영업을 전개하여 전세계적으로 가장 많이 사용되는 In-Memory Database의 자리를 지켜오고 있으며, 다량의 트랜잭션을 처리하면서 빠른 서비스를 요구하는 업무영역이 늘어나면서 그 활용범위도 나날이 늘어나고 있다. 오라클은 2005년 6월에 인메모리 데이터베이스 전문 업체인 타임스텐을 인수한 후 2006년도에 TimesTen 6.0 버전을 출시하였고, 2007년 2월에 TimesTen 7.0 버전을 출시하였습니다.

그럼 여기서 In-Memory Database의 등장 배경에 대해서 알아보도록 하겠습니다.
우선 첫번째로 수십에서 수백 기가 바이트의 대용량 메모리를 탑재한 머신들이 나온 것입니다. 기존에는 메모리를 프로그램의 일부 실행을 위해서 주로 사용하였으나, 이제는 데이터를 저장하는 저장소로서의 역할을 할 수 있을 정도로 양적인 확장이 되었습니다.

두번째로 64비트 체제가 되면서 메모리를 무한대로 사용할 수 있게 되었다는 것입니다. 기존의 32비트 체제에서는 사용가능한 공간이 최대 4GB까지였으나 64비트 체제에서는 테라단위 이상의 메모리 어드레싱이 가능해졌기 때문에 대용량 메모리를 관리하고 활용할 수 있게 되었습니다.

세번째로는 메모리 가격의 인하로 인한 시장의 인식 변화입니다.
최근 몇 년동안 시스템에 사용되는 메모리의 가격이 계속 떨어지고 있어 대용량 메모리 시스템을 구축하기가 훨씬 수월해졌습니다. 그로인해 데이터가 많이 많고, 고성능을 원하는 사용자들이 In-Memory Database의 도입을 고려하게 되었습니다.

마지막으로 네번째는 안정적이고 빠른 실시간 처리를 요구하는 업무나 서비스를 받는 고객의 요구가 한 단계 높아졌다는 것입니다.
빠르게 증가하는 정보의 양을 보다 신속하고 안정적이게 처리하는 것이 요구되고 있으며, 많은 트랜잭션을 한꺼번에 처리하는 통신이나 금융 등의 분야에서는 고객의 만족도를 높이고 업무를 원활하게 수행하기 위해서 In-Memory Database를 도입하고 있는 추세입니다.

갈수록 안정적이고 빠른 성능을 요구하는 기관이나 기업이 늘어남에 따라 TimesTen이나 알티베이스와 같은 In-Memory Database의 사용이 늘어날 것으로 보이며, 이는 오라클이 임베디드 단의 Oracle Lite와 백엔드 단의 Oracle Database 에서부터 중간 단의 Oracle TimesTen에 이르기까지 Total System 구축에 공을 들이고 있는 이유이기도 합니다.

그렇다면 이번엔 메모리 DB(In-Memory DB 또는 MMDBMS)와 디스크 DB(Disk Resident DataBase Management System, DRDBMS)의 성능적인 차이가 발생하는 이유를 살펴해보도록 하겠습니다.
메모리 DB와 디스크 DB의 성능상, 기능상의 가장 큰 차이점은 데이터 접근 방식에 있습니다. 디스크 DB는 데이터 접근 속도를 높이고, 효율적으로 하기 위해서 DBMS가 직접 관리하는 DB 버퍼를 이용합니다. 한 번 접근한 데이터의 내용을 블록단위로 버퍼에 올려놓아 동시에 접근하는 세션이나 향후에 같은 데이터를 접근하는 트랜잭션을 위하여 디스크 I/O가 발생되지 않도록 하여 일정 성능을 보장합니다. 반면에 메모리 DB는 DB 버퍼라는 개념이 존재하지 않습니다. 버퍼는 원래 디스크 I/O의 횟수를 줄여서 성능을 높이는데 주로 사용되는데 메모리 DB의 경우에는 모든 데이터가 메모리 위에 이미 존재하기 때문에 버퍼를 위한 메모리 공간을 둘 필요가 없기 때문입니다.

DB 버퍼를 통하여 데이터를 접근할 경우 버퍼에서의 데이터 존재 유무에 따라 처리 성능의 차이가 발생합니다. 만일 필요한 데이터가 버퍼에 존재하지 않아 데이터를 디스크를 참조하여 버퍼에 Load 하는 작업을 하게 된다면 디스크 I/O가 발생되기 때문에 버퍼에 존재하는 데이터를 바로 참조하는 것과 비교할 때 처리속도가 떨어지게 되며, 이것을 줄이는 것이 튜닝의 중요한 포인트 입니다.

그러나 이러한 TimesTen의 여러 장점에도 불구하고 TimesTen 도입을 막고 있는 요소들이 있습니다. 우선 TimesTen의 Reference 부족입니다. Oracle Database를 관리하고 복구할 수 있는 DBA는 많이 있는 반면, TimesTen이라는 메모리 DB를 접해보고 관리하고 백업, 복구할 수 있는 DBA는 많지 않기 때문입니다. 오라클에서 제공하는 Reference Site인 MetaLink 에서도 TimesTen에 대한 내용은 아직 많지 않고, 아직 한국오라클에는 TimesTen을 배울 수 있는 과정이 없다는 것도 한 원인일 것입니다.

그리고 메모리 DB라는 것이 중요한 데이터를 휘발성 저장소인 메모리에 올려놓았다가 서버가 내려가기라도 하면 자칫 전체 또는 일부 데이터를 유실하는 것이 아닌가 하는 걱정이 들게 합니다. 하지만 메모리 DB는 체크포인트 파일과 로그파일을 사용하여 디스크에 메모리에 있는 데이터의 내용이 저장되기 때문에 소중한 데이터를 안전하게 고객에게 서비스할 수 있습니다.

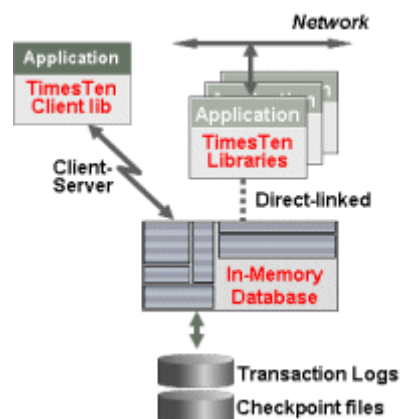
이처럼 메모리 DB에 대한 인식부족과 TimesTen의 Reference가 많지 않는 등의 여러 이유로, 이번에 Oracle TimesTen In-Memory Database에 대한 기술노트를 작성하여 배포함으로써 메모리 DB에 대한 인식을 변화 시키고, TimesTen의 주요 기능과 관리방법 등을 소개하여 TimesTen을 보다 효율적이고 편안하게 사용할 수 있도록 구성하였습니다.

1.2. Oracle TimesTen의 구성

1.2.1. Oracle TimesTen In-Memory Database

Oracle TimesTen In-Memory Database 는 통신, 금융, 국방 등 즉각적인 응답시간과 매우 높은 처리성을 요구하는 실시간 엔터프라이즈 환경을 위한 메모리-최적화 관계형 데이터베이스입니다.

Oracle TimesTen In-Memory Database 는 런타임의 데이터 위치에 대한 기본 가정을 뒤집음으로써 실시간 성능을 구현한 제품입니다. 데이터를 메모리에서 관리하고, 최적화된 데이터 구조와 액세스 알고리즘을 제공함으로써 데이터베이스의 실행 효율성을 극대화하고, 응답 시간 및 처리 속도를 극적으로 개선하여 완전하게 캐시 된 디스크 기반 RDBMS 환경에 비해서도 훨씬 뛰어난 성능을 제공합니다. Oracle TimesTen In-Memory Database 는 애플리케이션 내에 임베디드 데이터베이스 형태로 구현되며, 프로세스 간 커뮤니케이션 및 불필요한 네트워크 오버헤드를 완전히 제거하여 데이터베이스의 성능을 추가적으로 개선합니다.



때때로 인-메모리 데이터베이스는 단일 사용자용 애플리케이션에서만 사용 가능하며 서버가 다운되면 데이터가 소실되는 것으로 오해하는 경우가 많습니다. 하지만 Oracle TimesTen In-Memory Database 에서는 그렇지 않습니다. 멀티-유저, 멀티-쓰레드 애플리케이션이 일반적으로 사용되며 커밋 메커니즘과 로우-레벨 락킹 또한 지원됩니다. 또 내부 레코드 버전 관리를 통해 읽기 작업과 쓰기 작업 간의 락 경쟁 요소가 제거되므로, 일관성 있는 응답시간과 높은 수준의 동시성을 보장할 수 있습니다. Oracle TimesTen 데이터베이스는 영구적으로 저장되며 복구가 가능합니다. 각 애플리케이션은 성능 요구 사항에 따라 트랜잭션 ACID 속성 설정을 조정할 수 있습니다. 표준 관계형 데이터 모델을 기반으로 Oracle TimesTen 데이터베이스의 접근을 위해 SQL, JDBC, ODBC 등을 사용할 수 있습니다.

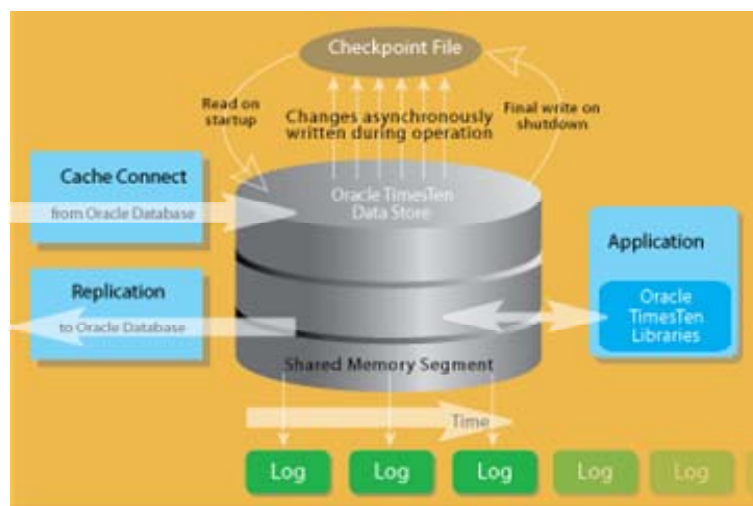
RDBMS 및 SQL 인터페이스에 친숙한 개발자라면 누구든 Oracle TimesTen In-Memory Database 를 이용하여 실시간 애플리케이션을 신속하게 개발하는 것이 가능합니다.

Oracle TimesTen In-Memory Database 는 일회성 록업 캐시, 운영 데이터 저장소, 미션 크리티컬 트랜잭션 프로세싱 시스템 등 다양한 구축 환경과 광범위한 시나리오를 지원합니다.

대부분의 고객들은 미션 크리티컬 애플리케이션 지원을 위해,고가용성 및 로드 밸런싱을 지원하는 Replication- TimesTen to TimesTen 옵션을 추가로 사용하고 있습니다. SOA(Service-Oriented Architecture) 지원을 위한 애플리케이션-티어 캐시를 구현하는 경우라면 Cache Connect to Oracle 옵션을 이용하여 인-메모리 데이터베이스를 업데이트 가능한 캐시(updatable cache)로 확장하고, 데이터 로딩 작업 및 오라클 데이터베이스와 Oracle TimesTen In-Memory Database 간의 동기화를 관리할 수 있습니다.

TimesTen의 구성을 간단히 살펴보면 TimesTen은 밀리초가 아닌 마이크로초(microsecond) 단위의 미션-크리티컬 실시간 애플리케이션을 개발할 수 있는 환경을 제공합니다. Oracle TimesTen은 스탠드얼론 형태로 사용될 수도 있고, 또는 오라클 데이터베이스에서 자주 접근되는 데이터를 위한 양방향 캐시(bidirectional cache)로 사용될 수도 있습니다.

아래 그림은 Oracle TimesTen 의 기반을 이루는 아키텍처를 보여 주고 있습니다. Oracle TimesTen 의 관점에서 볼 때, 데이터베이스는 데이터 스토어(data store)라 불립니다. 디스크 상에 구현된 데이터 스토어는 체크포인트 파일(checkpoint file)이라 부릅니다. 데이터 스토어가 처음으로 열리면, Oracle TimesTen 은 체크포인트 파일로부터 데이터의 전체 콘텐츠를 메모리로 읽어 들입니다. 이후 INSERT, DELETE, SELECT, UPDATE 등의 작업은 메모리 상에서 실행됩니다. 이러한 작업에 의해 발생한 데이터 스토어의 변경 사항은 정기적, 비동기적으로 디스크 상의 체크포인트 파일에 기록됩니다. 데이터 스토어가 셧다운 처리되면, 데이터 스토어를 종료하기 전에 아직 기록되지 않은 변경 사항을 모두 체크포인트 파일에 기록합니다.



Oracle TimesTen 아키텍처

Oracle TimesTen 는 또 디스크 스토리지를 트랜잭션 보호를 위한 복구 메커니즘으로 활용합니다. 트랜잭션은 로그 파일에 기록됩니다. 시스템에 장애가 발생하여 재시작되면, 체크포인트 파일을 메모리로 읽어 들이고, 트랜잭션 로그를 적용하고, 데이터 스토어를 여는 작업이 순차적으로 실행됩니다. 또 데이터베이스의 변경 사항을 주기적으로 체크포인트 파일에 저장함으로써, 이러한 복구 작업에 소요되는 시간을 최소화할 수 있게 합니다.

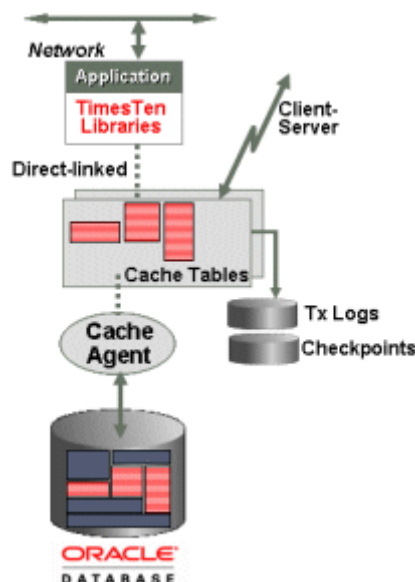
Oracle TimesTen 는 스탠드얼론 데이터베이스로 실행될 수도 있고, 또는 오라클 데이터베이스를 위한 일종의 성능 가속기로 활용될 수도 있습니다. 그림 1 은 Oracle TimesTen 을 오라클 데이터베이스에 연결하기 위한 캐시 에이전트와 복제 서비스를 보여주고 있습니다. 본 문서의 시나리오에서는,

(스탠드얼론 방식을 예시하기 위해) 일부 데이터는 Oracle TimesTen 에 저장되며, (성능 가속기로의 활용 방법을 예시하기 위해) 다른 데이터는 오라클 데이터베이스에 저장됩니다. 이처럼 두 가지 방법을 필요에 따라 조합하여 사용하는 것이 가능합니다.

1.2.2. Cache Connect to Oracle

Cache Connect to Oracle은 Oracle TimesTen In-Memory Database에서 오라클 데이터베이스에 대한 "updatable" 실시간 캐싱 기능을 제공하기 위한 옵션입니다. 캐시 테이블은 애플리케이션 계층에 구현되므로 백엔드 시스템의 부하가 최소화되며, 혁신적으로 향상된 응답 시간과 확장성을 제공하는 실시간 애플리케이션 환경이 지원됩니다.

기업 데이터베이스에 저장된 대부분의 데이터는 히스토리 정보로서 자주 접근되지 않는 특성을 가집니다. 하지만 이러한 데이터 중에는 요청되는 즉시 즉각적으로 접근이 가능해야만 하는 극히 일부의 데이터가 존재합니다. VIP 고객 정보, 현재 처리 중인 주문, 최근 거래 내역, 제품 카탈로그 등의 그 예가 될 수 있습니다. Cache Connect to Oracle 옵션은 이러한 정보가 실시간 접근을 위해 Oracle TimesTen In-Memory Database 에 자동으로 복사, 관리될 수 있도록 합니다. Cache Connect to Oracle 은 오라클 데이터베이스와 Oracle TimesTen In-Memory Database 간의 연결 및 양방향 데이터 전송 기능을 지원합니다.



대부분의 캐싱 제품은 읽기 전용 환경만을 지원하며, 따라서 매우 제한된 기능만을 포함하고 있습니다. 하지만 Cache Connect to Oracle 는 SQL 읽기/쓰기 작업을 완전하게 지원하므로 훨씬 다양한 용도로 활용이 가능합니다. 예를 들어 데이터를 네트워크 속도로 캡처, 처리하고, 온라인 상거래, 증권 거래 시스템, 빌링, 실시간 비즈니스 프로세스 모니터링, 실시간, 데이터 분석 및 기타 읽기 전용 캐시로는 지원 불가능한 트랜잭션 기반 시스템을 구현할 수 있습니다. 그 밖에도 구독자 프로파일 등의 참조 데이터, 인증 및 네트워크 설정을 위한 록업 테이블 등 변경 사항이 발생할 때마다 캐시가 업데이트되어야 하는 경우에도 활용이 가능합니다.

Cache Connect to Oracle 은 오라클 데이터베이스 기능 및 인터페이스를 최대한 활용할 수 있도록 설계되었습니다. Cache Connect to Oracle 은 오라클 데이터베이스와 동일한 데이터베이스 문자 셋과 데이터 타입을 지원하므로, 데이터가 동일한 방식으로 저장, 처리됨을 보장할 수 있습니다.

Cache Connect to Oracle 은 캐시 그룹(cache group)의 개념을 지원합니다. 캐시 그룹이란 오라클 데이터베이스의 테이블 전체 또는 일부에 매핑되는 인-메모리 데이터베이스 테이블의 집합을 의미합니다. 캐시 그룹은 테이블의 로우 전체 또는 로우 중 일부만으로도 구성될 수 있습니다. 여러

개의 캐시 그룹을 사용하여 오라클 데이터베이스의 여러 테이블을 각각 다른 테이블 셋으로 캐시 처리하는 것이 가능합니다.

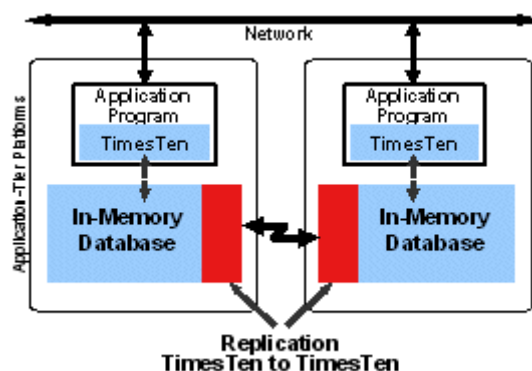
Oracle TimesTen의 캐시 테이블은 오라클 데이터베이스에 동기식, 또는 비동기식으로 "write-through" 처리되며, 동기식/비동기식 중 어떤 방법을 이용할 것인지는 성능/일관성에 대한 고려 결과에 따라 달라집니다. 읽기 전용 캐시 그룹의 경우에는 오라클 데이터베이스에 대한 증분 업데이트가 Oracle TimesTen의 캐시 테이블에 비동기적으로 리프레시 됩니다.

Cache Connect to Oracle은 오라클 데이터베이스 서버가 다운되거나 네트워크 연결이 끊어진 경우에도 계속적으로 동작할 수 있도록 설계되었습니다. Oracle TimesTen의 캐시 테이블에 커밋 되는 트랜잭션은 계속적으로 모니터링되며, 연결이 재개되는 즉시 오라클 데이터베이스에 전달됩니다. 마찬가지로 오라클 데이터베이스의 캐시 테이블에 커밋 되는 트랜잭션은 계속적으로 모니터링되며, 연결이 재개되는 즉시 Oracle TimesTen에서 업데이트됩니다.

1.2.3. Replication – TimesTen to TimesTen

Replication – TimesTen to TimesTen은 Oracle TimesTen In-Memory Database의 옵션 중 하나로, TimesTen 데이터베이스와 TimesTen 캐시 그룹 간의 실시간 트랜잭션 기반 복제를 통해 고가용성 및 로드 밸런싱을 지원합니다.

대부분의 실시간 애플리케이션에서 가용성은 가장 핵심적인 요구 사항의 하나입니다. 24x7 기준으로 비즈니스를 운영하는 통신 업체 및 웹 기반 서비스 업체들은 한 순간의 서비스 다운타임도 용납할 수 없습니다. 또 증권 거래 시스템은 증권 시장이 개장되어 있는 동안 반드시 계속적으로 운영되어야 합니다. 시스템의 운영 방식이 실시간에 가까울 수록, 고가용성에 대한 요구는 더욱 높아집니다.



Oracle TimesTen In-Memory Database는 데이터 복제 기술을 기반으로 고가용성 환경을 구현합니다. TimesTen Replication 옵션은 마스터-구독자 모델을 기반으로 하며, 지정된 테이블 또는 전체 데이터베이스의 커밋된 변경 사항이 소스에서 하나 또는 그 이상의 구독자 데이터베이스로 복제됩니다. 데이터베이스를 마스터 및 구독자로 동시에 지정한 경우에는 멀티-노드 N-웨이 복제를 포함하는 양방향 복제의 구성이 가능합니다. 양방향 복제 과정에서 충돌이 발생한 경우, 타임스탬프 기반의 충돌 탐지 및 해결 메커니즘이 동작하여 복제본의 일관성을 보장해 줍니다.

트랜잭션 로그 기반의 복제 메커니즘을 통해 높은 효율성과 낮은 오버헤드를 보장할 수 있습니다. 또 비동기식, 동기식 데이터 복제가 모두 지원됩니다. 비동기식 복제는 극대화된 성능을 가능하게 하지만, 애플리케이션이 복제 메커니즘과 완전하게 분리된 상태로 운영된다는 단점이 있습니다. 마스터/구독자 데이터베이스 간의 완벽한 일관성을 요구하는 애플리케이션 환경에서는 동기식 복제가 됩니다.

Replication – TimesTen to TimesTen 옵션은 Cache Connect to Oracle 옵션과 완벽하게 호환합니다. 오라클 데이터베이스로부터 캐시된 테이블에 대해 고가용성을 구현하기 위해서는, 서로 다른 Oracle TimesTen 데이터베이스의 캐시 그룹 간에도 복제가 지원되어야 합니다.

2. Oracle TimesTen의 설치

2.1. 환경설정

Oracle TimesTen은 root 유저와 non-root 유저에서의 설치를 모두 지원하며, 이번 설치 는 non-root 유저를 기준으로 작성되었다.

```
[ultra60:/]#vi /etc/system      ← 수정 후 시스템 리부팅 필요함

set semsys:seminfo_semmni = 20
set semsys:seminfo_semmns = 100
set semsys:seminfo_semmns = 2000
set semsys:seminfo_semmnu = 2000

[ultra60:/]#groupadd -g 200 timesten      ← timesten 그룹 생성
[ultra60:/]#useradd -g 200 -u 250 -d /oracle/TimesTen -s /bin/ksh timesten      ← timesten 유저 생성

[ultra60:/]#mkdir /etc/TimesTen      ← timesten 설치전에 반드시 만들어져 있어야함
[ultra60:/]#chmod 755 /etc/TimesTen
[ultra60:/]#chown timesten:timesten /etc/TimesTen      ← timesten 유저가 사용가능하도록 owner 수정

[ultra60:/oracle/TimesTen]#cd /etc/TimesTen
[ultra60:/etc/TimesTen]#ls -al
drwxrwxr-x   2 timesten timesten   512 12월  2일 16:14 ./
drwxr-xr-x  62 root      sys       4096 12월  2일 14:49 ../

[ultra60:/oracle/TimesTen]#vi .profile
export LD_LIBRARY_PATH=$ORACLE_HOME/LIB:$ORACLE_HOME/NETWORK/LIB      ← 추가해주어야 함

※ 생성된 TimesTen 유저가 $ORACLE_HOME 으로 change directory 할 수 있는지 확인
```

2.2. TimesTen 설치

```
[ultra60:/install/TimesTen]$su - timesten      ← timesten 유저로 로그인
[ultra60:/install/TimesTen]$id
uid=250(timesten) gid=200(timesten)
[ultra60:/install/timesten]$ls
./                  ../                  timesten70500.sparc64.tar.gz
[ultra60:/install/timesten]$gunzip *.gz      ← 압축해제
[ultra60:/install/timesten]$ls
./                  ../                  timesten70500.sparc64.tar
[ultra60:/install/timesten]$tar xvf timesten70500.sparc64.tar
[ultra60:/install/timesten/sparc64]$ls
./                  README.html*         doc/                  uninst.sh*
../                  SPARC64/             install.pl*
3rdparty/           behaviorchanges.txt* setup.sh*
[ultra60:/install/timesten/sparc64]$./setup.sh      ← 옵션은 -install과 -uninstall이 있으며,
                                                    Default 옵션은 -install임
                                                    -uninstall 옵션은 timesten 지울때 사용함

NOTE: Each TimesTen installation is identified by a unique instance name.
      The instance name must be a non-null alphanumeric string, not longer than 255 characters.
```


Please choose an instance name for this installation? [tt70]

Instance name will be 'tt70'.

Is this correct? [yes]

Please select a product :

[1] Oracle TimesTen In-Memory Database

[2] Oracle In-Memory Database Cache ← 이전 버전에서는 옵션명이 “Oracle TimesTen In-Memory Database with Cache Connect to Oracle” 이었음

Which product would you like to install? [1] 2 ← 2번 : Cache Connect 기능 활성화
(TimesTen 내에 Oracle database table들을 Cache 할 수 있음)

Of the three components:

[1] Client/Server and Data Manager ← TimesTen 전체 구성요소 설치

[2] Data Manager Only ← TimesTen Data Manager 구성요소만 설치

[3] Client Only ← TimesTen Client 구성요소만 설치

Which would you like to install? [1]

Where would you like to install the tt70 instance of TimesTen? [/oracle/TimesTen]

← TimesTen software가 설치되는 위치 지정

Where would you like to create the daemon home directory? [/oracle/TimesTen/tt70/info]

← daemon의 정보나 환경 파일이 저장되는 위치 지정

The directory /oracle/TimesTen/tt70/info does not exist.

Do you want to create it? [yes]

Installing into /oracle/TimesTen/tt70 ...

Uncompressing ...

The TimesTen Demo applications can take up to 64 Mbytes of disk space.

Depending on how your system is configured, you may not want to create the

DemoDataStore directory in the default location,

/oracle/TimesTen/tt70/info/DemoDataStore

Where would you like to create the DemoDataStore directory? [/oracle/TimesTen/tt70/info]

← demo database files의 위치 선택

Creating /oracle/TimesTen/tt70/info/DemoDataStore ...

NOTE: All installations that replicate to each other must use the same daemon port number that is set at installation time. The daemon port number can be verified by running 'ttVersion'.

The default port number is 17001. ← 32bit = 17000 / 64bit = 17001 사용

Do you want to use the default port number for the TimesTen daemon? [yes]

The daemon will run on the default port number (17001).

Processing /oracle/TimesTen/tt70/PERL/perl.tar ...

Would you like to enable datastore access control? [no]

← DNS 내에서 사용자를 생성하기 위해서는 access control을 생성

The daemon logs will be located in /oracle/TimesTen/tt70/info

Would you like to specify a different location for the daemon logs? [no]

← 해당 디렉토리에 tterros.log, ttmesg.log 등의 파일이 생성됨

NOTE: Cache Connect to Oracle requires the ORACLE_HOME environment variable.
If you cannot provide a valid path to your Oracle installation,
you must modify the script :
 /oracle/TimesTen/tt70/bin/ttThunk
as well as the daemon options file :
 /oracle/TimesTen/tt70/info/ttendaemon.options
when you have acquired the appropriate value.
Cache Connect to Oracle will not function properly without
ORACLE_HOME defined in these files.
Note that you can add or change this instance's ORACLE_HOME setting
by running <install_dir>/bin/ttmodinstall -changeOracleSetting.

Would you like to specify a value for ORACLE_HOME ? [yes]

Please enter a value for ORACLE_HOME (q=quit)? [] /oracle/10.2/app ← Oracle의 HOME 설정
← .profile에 ORACLE_HOME이 설정되어 있다면 생략됨

The following variables have been set in the file :

/oracle/TimesTen/tt70/bin/ttThunk

ORACLE_HOME=/oracle/10.2/app

Would you like to enable the Cache Connect to Oracle Administrator? [yes] ← Admin 설정

What TCP/IP port number would you like Cache Connect to Oracle Administrator to listen on?
[17005]

NOTE: To access the TimesTen Cache Connect to Oracle Administrator
go to the url: http://localhost:17005/cache

(여기서 몇 분 정도의 시간이 소요됨)

Installing server components ...

Would you like to log all server Connects/Disconnects? [yes] ← 서버접속로그 남길지 여부 확인

What is the TCP/IP port number that you want the TimesTen Server to listen on? [17003]

← TimesTen 서버가 사용할 listen 포트

Starting the daemon ...

Warning: directory /oracle/10.2/app/lib does not exist. Check ORACLE_HOME for proper operation
of Cache Connect.

TimesTen Daemon startup OK. ← TimesTen Daemon이 실행됨

Installing client components ...

What is the name of the host running the TimesTen server? [ultra60] ← hostname 명

What is the TCP/IP port number that the TimesTen server is listening on? [17003] ← port 번호

What is the name of the instance running the TimesTen server? [tt70] ← instance 명

Creating new /oracle/TimesTen/tt70/info/sys.ttconnect.ini

Extracting 3rd party tools ...

Would you like to install the documentation? [yes] ← documentation 설치여부

Where would you like to create the doc directory (s=skip)? [/oracle/TimesTen/tt70/doc]

The directory /oracle/TimesTen/tt70/doc does not exist.

Do you want to create it? [yes]

NOTE: The TimesTen daemon startup/shutdown scripts have not been installed.

Run the script '/oracle/TimesTen/tt70/bin/setuproot' as root.

This will move the TimesTen startup script into its appropriate location.

← DB 머신이 재시작될 때 TimesTen이 자동으로 시작되도록 설정하려면 setuproot 스크립트를 실행

The startup script is currently located here :

'/oracle/TimesTen/tt70/startup/tt_tt70'.

End of TimesTen installation.

[ultra60:/oracle/TimesTen]\$vi .profile

export PATH=\$HOME/TimesTen/tt70/bin:\$PATH

export LD_LIBRARY_PATH=\$HOME/TimesTen/tt70/lib:\$LD_LIBRARY_PATH

← PATH 와 LD_LIBRARY_PATH 에 TimesTen bin과 lib 디렉토리 추가

2.3. TimesTen 설치 제거

[ultra60:/oracle/TimesTen]\$/oracle/TimesTen/tt70/bin/setup.sh -uninstall

** WARNING **

The uninstallation has been executed by a non-root user.

If the TimesTen daemon startup scripts were installed,

you must run <install_dir>/bin/setuproot -uninstall

to remove them. If you proceed with this uninstallation, you will have to remove the startup scripts manually.

Would you like to proceed with the uninstallation? [yes]

Are you sure you want to completely remove the instance tt70 (TimesTen7.0.5) and all of its components? [yes]

NOTE: /oracle/TimesTen/tt70/info contains information related to the data

stores that have been created with this release. If you remove

/oracle/TimesTen/tt70/info

you will no longer be able to access your data stores,

nor would you be able to restore nor migrate your data.

Would you also like to remove all files in /oracle/TimesTen/tt70/info? [no] yes

/oracle/TimesTen/tt70 Removed

TimesTen uninstall completed.

2.4. TimesTen 설치 확인

- TimesTen Damon 과 Process 상태 확인

[ultra60:/oracle/TimesTen]\$ttStatus ← 설치가 정확하게 되었는지 데몬과 프로세스 상태 확인

[ultra60:/oracle/TimesTen]\$ttstatus

TimesTen status report as of Sun Dec 28 02:33:06 2008

Daemon pid 4845 port 17001 instance tt70

TimesTen server pid 4851 started on port 17003

TimesTen webserver pid 4850 started on port 17005

Data store /tmp/ttdemo

There are 9 connections to the data store

Data store is in shared mode

Shared Memory KEY 0x07000332 ID 1002

Type	PID	Context	Connection Name	ConnID
Cache Agent	4858	0x00000001002e1cc0	Handler	2
Cache Agent	4858	0x000000010041fcd0	Timer	3
Cache Agent	4858	0x0000000100433330	Aging	4
Subdaemon	4848	0x000000010015ab00	Worker	2042
Subdaemon	4848	0x000000010026c5d0	Flusher	2043
Subdaemon	4848	0x000000010027d290	Checkpoint	2044
Subdaemon	4848	0x000000010028df50	Aging	2047
Subdaemon	4848	0x00000001002deed0	Monitor	2045
Subdaemon	4848	0x000000010030ffa0	HistGC	2046

Replication policy : Manual

Cache agent policy : Manual

TimesTen's Cache agent is running for this data store

End of report

2.5. TimesTen DB 생성

2.5.1. TimesTen DB 생성

```
- DB 생성전에 DSN(Data source name)을 반드시 정의해주어야 함
[ultra60:/oracle/TimesTen]$cp $HOME/Timesten/tt70/info/sys.odbc.ini $HOME/.odbc.ini
    ← sample DSN 파일로부터 home directory로 DSN 정의를 위한 .odbc.ini 파일을 복사
[ultra60:/oracle/TimesTen]$vi .odbc.ini

[ODBC Data Sources]
TT_tt70=TimesTen 7.0 Driver
TpcbData_tt70=TimesTen 7.0 Driver
TptbmDataRepSrc_tt70=TimesTen 7.0 Driver
TptbmDataRepDst_tt70=TimesTen 7.0 Driver
TptbmData_tt70=TimesTen 7.0 Driver
BulkInsData_tt70=TimesTen 7.0 Driver
WiscData_tt70=TimesTen 7.0 Driver
RunData_tt70=TimesTen 7.0 Driver
CacheData_tt70=TimesTen 7.0 Driver
ttdemo=TimesTen 7.0 Driver    ← 해당 항목 추가
TpcbDataCS_tt70=TimesTen 7.0 Client Driver
TptbmDataCS_tt70=TimesTen 7.0 Client Driver
BulkInsDataCS_tt70=TimesTen 7.0 Client Driver
WiscDataCS_tt70=TimesTen 7.0 Client Driver
RunDataCS_tt70=TimesTen 7.0 Client Driver

- 설정정보는 여기에 정의함
#####
#
# New data source definitions can be added below.
#
#####

- 아래 6줄 삽입
[ttdemo]    ← DSN(Data Source Name) 정의
Driver=/oracle/TimesTen/tt70/lib/libtten.so
DataStore=/tmp/ttdemo    ← Data Store 위치 정의
PermSize=20
TempSize=20
DatabaseCharacterSet=K016MSWIN949    ← Database의 CharacterSet 정의
(만일 Oracle로 Cache Connect를 사용한다면, 반드시 Oracle DB Character set과 같게 설정해야 함)

- ttdemo Database로 접속하고 ttisql을 이용하여 테이블을 생성
[ultra60:/oracle/TimesTen]$ttisql    ← ttisql 실행

Copyright (c) 1996-2008, Oracle. All rights reserved.
Type ? or "help" for help, type "exit" to quit ttisql.
All commands must end with a semicolon character.

Command> connect ttdemo;    ← ttdemo DB에 접속
Connection successful: DSN=ttdemo;UID=timesten;DataStore=/tmp/ttdemo
;DatabaseCharacterSet=K016MSWIN949;ConnectionCharacterSet=US7ASCII
;DRIVER=/oracle/TimesTen/tt70/lib/libtten.so;PermSize=20;TempSize=20;TypeMode=0;
```

(Default setting AutoCommit=1)

```
Command> create table mytable          ← mytable 테이블 생성
> (id number,
> value varchar2(50),
> create_date date);
Command> desc mytable;                 ← mytable 테이블 생성정보 확인

Table TIMESTEN.MYTABLE:
  Columns:
    ID                                NUMBER
    VALUE                            VARCHAR2 (50) INLINE
    CREATE_DATE                      DATE

1 table found.
(primary key columns are indicated with *)
Command> exit
Disconnecting...
Done.
```

2.5.2. Cache Connect를 설정한 TimesTen DB 생성

```
- DB 생성전에 DSN(Data source name)을 반드시 정의해주어야 함
[ultra60:/oracle/TimesTen]$cp $HOME/Timesten/tt70/info/sys.odbci.ini $HOME/.odbc.ini
                                ← sample DSN 파일로부터 home directory로 DSN 정의를 위한 .odbc.ini 파일을 복사
[ultra60:/oracle/TimesTen]$vi .odbc.ini

[ODBC Data Sources]
TT_tt70=TimesTen 7.0 Driver
TpcbData_tt70=TimesTen 7.0 Driver
TptbmDataRepSrc_tt70=TimesTen 7.0 Driver
TptbmDataRepDst_tt70=TimesTen 7.0 Driver
TptbmData_tt70=TimesTen 7.0 Driver
BulkInsData_tt70=TimesTen 7.0 Driver
WiscData_tt70=TimesTen 7.0 Driver
RunData_tt70=TimesTen 7.0 Driver
CacheData_tt70=TimesTen 7.0 Driver
ttdemo=TimesTen 7.0 Driver          ← 해당 항목 추가
TpcbDataCS_tt70=TimesTen 7.0 Client Driver
TptbmDataCS_tt70=TimesTen 7.0 Client Driver
BulkInsDataCS_tt70=TimesTen 7.0 Client Driver
WiscDataCS_tt70=TimesTen 7.0 Client Driver
RunDataCS_tt70=TimesTen 7.0 Client Driver

- 설정정보는 여기에 정의함
#####
#
# New data source definitions can be added below.
#
#####
```

- 아래 8줄 삽입

```
[ttdemo]                                ← DSN(Data Source Name) 정의
(주의 : DSN은 32자를 넘을 수 없고, 공백을 넣을 수 없으며,
       옆의 특수기호는 넣을 수 없다. ( [ ] { } , ; ? * = ! @ % ' )
Driver=/oracle/TimesTen/tt70/lib/libtten.so
DataStore=/tmp/ttdemo                  ← Data Store 위치 정의
PermSize=20
TempSize=20
DatabaseCharacterSet=K016MSWIN949     ← Database의 CharacterSet 정의
(만일 Oracle로 Cache Connect를 사용한다면, 반드시 Oracle DB Character set과 같게 설정해야 함)
UID=hr                                ← HR 샘플 스키마의 테이블을 캐싱하도록 설정
OracleID=ora10                        ← Oracle Net service명 설정
```

- ttdemo Database로 접속하고 ttisql을 이용하여 테이블을 생성

```
[ultra60:/oracle/TimesTen]$ttisql      ← ttisql 실행
```

Copyright (c) 1996-2008, Oracle. All rights reserved.
Type ? or "help" for help, type "exit" to quit ttisql.
All commands must end with a semicolon character.

```
Command> connect "dsn=ttdemo;oraclepwd=hr";    ← ttdemo DB에 접속
Connection successful: DSN=ttdemo;UID=hr;DataStore=/tmp/ttdemo
;DatabaseCharacterSet=K016MSWIN949;ConnectionCharacterSet=US7ASCII
;DRIVER=/oracle/TimesTen/tt70/lib/libtten.so;PermSize=20;TempSize=20;TypeMode=0;
(Default setting AutoCommit=1)              ← 처음 접속과 동시에 /tmp/ttdemo에 data store 생성됨
Command> create table mytable                ← mytable 테이블 생성
> (id number,
> value varchar2(2000),
> create_date date);
Command> desc mytable;                      ← mytable 테이블 생성정보 확인
```

```
Table HR.MYTABLE:
Columns:
  ID                NUMBER
  VALUE             VARCHAR2 (2000) NOT INLINE
  CREATE_DATE       DATE
```

1 table found.
(primary key columns are indicated with *)

```
Command> autocommit 0;
Command> set passthrough 2;
Command> select * from v$version;    ← 이 SQL은 Oracle DB로 보내지고, 그 결과가 ttisql로 보내짐
< Oracle Database 10g Enterprise Edition Release 10.2.0.4.0 - 64bi >
< PL/SQL Release 10.2.0.4.0 - Production >
< CORE 10.2.0.4.0 Production >
< TNS for Solaris: Version 10.2.0.4.0 - Production >
< NLSRTL Version 10.2.0.4.0 - Production >
5 rows found.
Command> exit
Disconnecting...
Done.
```


2.6. Cache Connect to Oracle Concepts

Cache Group은 TimesTen data store로 cache된 오라클 데이터를 말하며, 단일 오라클 테이블의 일부나 전체 또는 여러 오라클 테이블들을 cache 하기 위해 만들어진다.

2.6.1. TimesTen 과 Oracle 설정

- 오라클 계정 생성

```
[ultra60:ora10:/oracle/10.2]$ sqlplus system/manager@ora10
```

```
SQL> create user testuser identified by mypsswrđ
```

```
2 default tablespace users;
```

```
User created.
```

```
SQL> grant connect, resource, create any trigger to testuser;
```

```
Grant succeeded.
```

```
SQL> commit;
```

```
Commit complete.
```

```
SQL> exit
```

- TimesTen에 계정 생성

```
[ultra60:/oracle/TimesTen]$ ttisql
```

```
Command> connect dsn=ttđemo;
```

```
Connection successful:
```

```
DSN=ttđemo;UID=timesten;DataStore=/tmp/ttdemo;DatabaseCharacterSet=KO16MSWIN949;ConnectionCharacterSet=US7ASCII;DRIVER=/oracle/TimesTen/tt70/lib/libtten.so;PermSize=20;TempSize=20;TypeMode=0;  
(Default setting AutoCommit=1)
```

```
Command> create user testuser identified by 'mypsswrđ';
```

```
Command> grant admin, ddl to testuser;
```

```
Command> exit
```

```
Disconnecting...
```

```
Done.
```

- TimesTen DSN 생성

- Oracle Database의 CharacterSet 확인

```
SQL> select value from nls_database_parameters  
where parameter='NLS_CHARACTERSET';
```

```
VALUE
```

```
-----  
KO16MSWIN949
```

```
[ultra60:/oracle/TimesTen]$ ls
```

```
./ .odbc.ini .sh_history profile
```

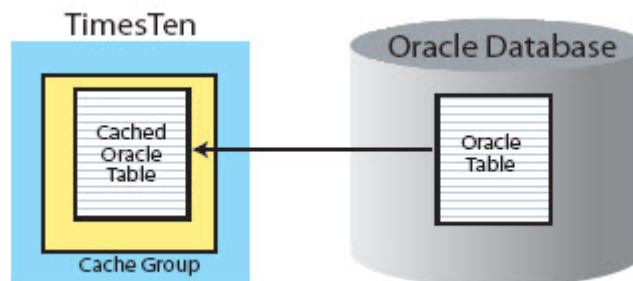
```
../ .profile history.txt tt70/
```

```
[ultra60:/oracle/TimesTen]$ vi .odbc.ini
```

```
[ttdemo]
Driver=/oracle/TimesTen/tt70/lib/libtten.so
DataStore=/tmp/ttdemo          ← Data Store의 위치
PermSize=20
TempSize=20
DatabaseCharacterSet=K016MSWIN949 ← CharacterSet은 Data를 가져오는 Oracle DB와 동일해야 함
UID=testuser                   ← Oracle DB 서버의 접속 유저명
OracleId=ora10                  ← tnsnames.ora에 등록되어 있는 서버명
OraclePwd=myspsswr d           ← Oracle DB 서버의 접속 유저 암호
```

2.6.2. Readonly Cache Group 생성

이번에는 Oracle Database의 single table의 내용을 simple READONLY cache group으로 어떻게 만드는지 알아보도록 하자.

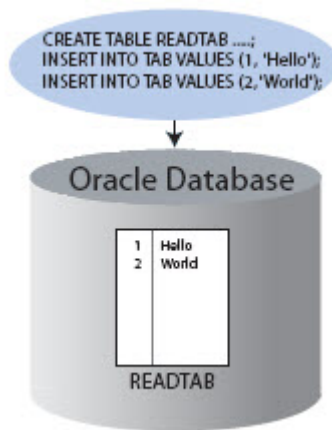


- TimesTen odbc.ini 파일 설정

```
[ultra60:/oracle/TimesTen]$vi .odbc.ini

[ttdemo]
Driver=/oracle/TimesTen/tt70/lib/libtten.so
DataStore=/tmp/ttdemo
PermSize=20
TempSize=20
UID=testuser
OracleId=ora10
OraclePwd=myspsswr d
```

- Oracle table 생성



```
[ultra60:ora10:/oracle/10.2]$ sqlplus testuser/myppswrd@ora10
```

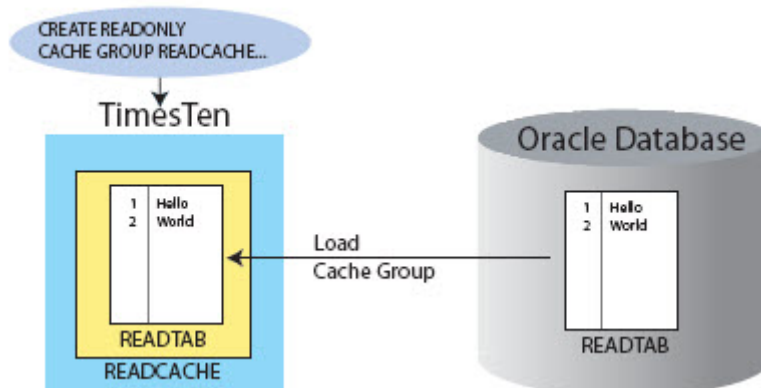
```
SQL> create table readtab  
  2  (a number not null primary key,  
  3  b varchar2 (31));  
Table created.
```

```
SQL> insert into readtab values (1, 'hello');  
1 row created.
```

```
SQL> insert into readtab values (2, 'world');  
1 row created.
```

```
SQL> commit;  
Commit complete.
```

- Cache group 생성



```
[ultra60:/oracle/TimesTen]$ ttisql ttdemo
```

Copyright (c) 1996-2008, Oracle. All rights reserved.
Type ? or "help" for help, type "exit" to quit ttlsq.
All commands must end with a semicolon character.

```
connect "DSN=ttdemo";
Connection successful:
DSN=ttdemo;UID=testuser;DataStore=/tmp/ttdemo;DatabaseCharacterSet=K016MSWIN949;ConnectionCharac
terSet=US7ASCII;DRIVER=/oracle/TimesTen/tt70/lib/libtten.so;OracleId=ora10;PermSize=20;TempSize=
20;TypeMode=0;
(Default setting AutoCommit=1)
Command> call ttcacheuidpwdset('testuser','mypsswrld');
Command> call ttcachestart();
Command> create readonly cache group readcache
> autorefresh interval 5 seconds
> from readtab
> (a number not null primary key,
> b varchar2(31));
```

- Cache group 로드하기

```
Command> select * from readtab;
0 rows found.

Command> load cache group readcache commit every 256 rows;
2 cache instances affected.

Command> select * from readtab;
< 1, hello >
< 2, world >
2 rows found.

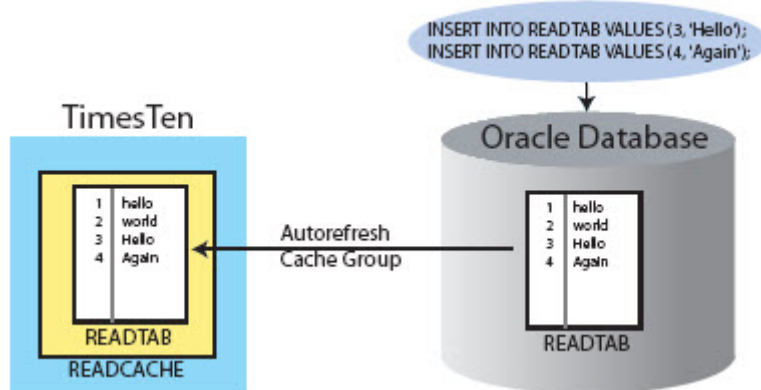
Command> cachegroups;
Cache Group TESTUSER.READCACHE:

Cache Group Type: Read Only
Autorefresh: Yes
Autorefresh Mode: Incremental
Autorefresh State: On
Autorefresh Interval: 5 Seconds

Root Table: TESTUSER.READTAB
Table Type: Read Only

1 cache group found.
```

- Oracle table 데이터 업데이트



Command> **select * from readtab;**

< 1, hello >
< 2, world >
2 rows found.

- Oracle Database에서 cache된 table의 데이터를 업데이트

SQL> **insert into readtab values (3, 'Hello');**
1 row created.

SQL> **insert into readtab values (4, 'Again');**
1 row created.

SQL> **commit;**
Commit complete.

(5초 후에 확인) ← TimesTen은 Oracle로부터 cache된 데이터를 자동적으로 refresh 함

Command> **select * from readtab;**

< 1, hello >
< 2, world >
< 3, Hello >
< 4, Again >
4 rows found.

- Cache group 삭제 및 Cache agent 정지

Command> **drop cache group readcache;**

← cache group 삭제

Command> **call ttcachestop();**

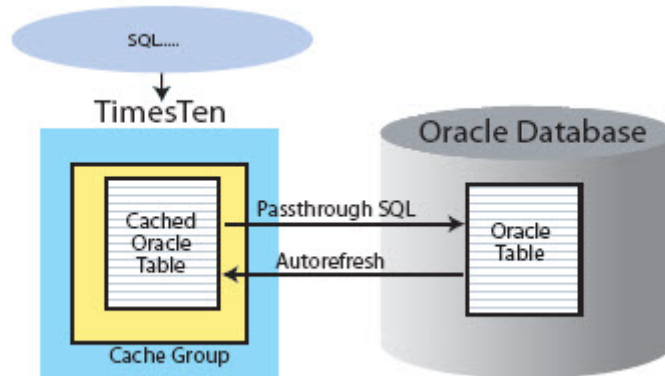
← cache agent 정지

Command> **select * from readtab;**

2206: Table TESTUSER.READTAB not found
The command failed.
Command> **cache groups;**

0 cache groups found.

2.6.3. SQL Passthrough 방식으로 전환



- TimesTen odbc.ini 파일 설정

```
[ultra60:/oracle/TimesTen]$vi .odbc.ini
```

```
[ttdemo]
```

```
Driver=/oracle/TimesTen/tt70/lib/libtten.so
```

```
DataStore=/tmp/ttdemo
```

```
PermSize=20
```

```
TempSize=20
```

```
DatabaseCharacterSet=K016MSWIN949
```

← 연결하고자 하는 Oracle DB의 characterSet과 동일하게 설정

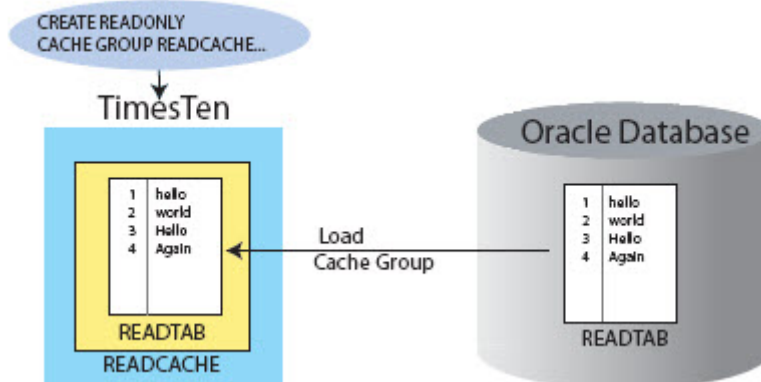
```
UID=testuser
```

```
OracleId=ora10
```

```
OraclePwd=mypsswr d
```

```
Passthrough=2
```

- READONLY cache group 생성



```
[ultra60:/oracle/TimesTen]$ttisql
```

```
Command> connect dsn=ttdemo;
```

```
Connection successful:
```

```
DSN=ttdemo;UID=testuser;DataStore=/tmp/ttdemo;DatabaseCharacterSet=K016MSWIN949;ConnectionCharacterSet=US7ASCII;DRIVER=/oracle/TimesTen/tt70/lib/libtten.so;OracleId=ora10;PermSize=20;TempSize=20;PassThrough=2;TypeMode=0;  
(Default setting AutoCommit=1)
```

```

Command> cachegroups;                                ← 생성되어 있는 cache group을 확인
Cache Group TESTUSER.READCACHE:

Cache Group Type: Read Only
Autorefresh: Yes
Autorefresh Mode: Incremental
Autorefresh State: On
Autorefresh Interval: 5 Seconds

Root Table: TESTUSER.READTAB
Table Type: Read Only

1 cache group found.
Command> drop cache group readcache;                  ← 생성되어 있는 cache group을 삭제
Command> cachegroups;
0 cache groups found.

Command> call ttcacheuidpwdset('testuser','mypsswr');
Command> call ttcachestart();
Command> create readonly cache group readcache        ← cache group 생성
> autorefresh interval 5 seconds
> from readtab
> (a number not null primary key,
> b varchar2 (31));

Command> load cache group readcache commit every 256 rows; ← cache group을 Oracle 데이터를 로드
4 cache instances affected.

```

- Cache group table에 데이터를 업데이트

```

Command> select * from readtab;
< 1, hello >
< 2, world >
< 3, Hello >
< 4, Again >
4 rows found.

Command> insert into readtab values (5, 'Just');      ← readtab으로 데이터를 입력
1 row inserted.
Command> insert into readtab values (6, 'Passing');   ← readtab으로 데이터를 입력
1 row inserted.
Command> insert into readtab values (7, 'Through');    ← readtab으로 데이터를 입력
1 row inserted.

Command> select * from readtab;                       ← TimesTen cache group의 readtab table에서 데이터 확인
< 1, hello >
< 2, world >
< 3, Hello >
< 4, Again >
< 5, Just >
< 6, Passing >
< 7, Through >
7 rows found.

```



```
SQL> select * from readtab;
```

← Oracle Database의 readtab table에서 데이터 확인

```
  A B
```

```
-----  
 1 hello  
 2 world  
 3 Hello  
 4 Again  
 5 Just  
 6 Passing  
 7 Through
```

```
7 rows selected.
```

- Cache group 삭제 및 Cache agent 정지

```
Command> drop cache group readcache; ← cache group 삭제
```

```
Command> call ttcachestop(); ← cache agent 정지
```

```
Command> select * from readtab;
```

```
2206: Table TESTUSER.READTAB not found
```

```
The command failed.
```

```
Command> cachegroups;
```

```
0 cache groups found.
```

- 특정 cache만 재시작 하는 방법

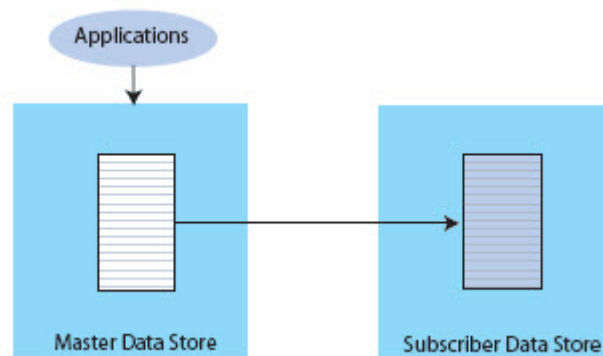
```
[ultra60:/oracle/TimesTen]$ttadmin -cachestop ttdemo
```

```
RAM Residence Policy      : inUse  
Replication Agent Policy  : manual  
Replication Manually Started : False  
Cache Agent Policy        : manual  
Cache Agent Manually Started : False
```

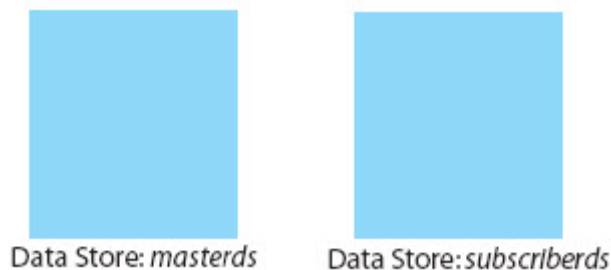
```
[ultra60:/oracle/TimesTen]$ttadmin -cachestart ttdemo
```

```
RAM Residence Policy      : inUse  
Replication Agent Policy  : manual  
Replication Manually Started : False  
Cache Agent Policy        : manual  
Cache Agent Manually Started : True
```

2.7. TimesTen Replication (복제)



- Master data store 와 Subscriber data store 생성



```
[ultra60:/oracle/TimesTen]$vi .odbc.ini
```

[ODBC Data Sources]

```
masterds=TimesTen 7.0 Driver
subscriberds=TimesTen 7.0 Driver
```

← ODBC Data Sources 항목에 masterds 추가

← ODBC Data Sources 항목에 subscriberds 추가

[masterds]

← masterds Data Source 정의 추가

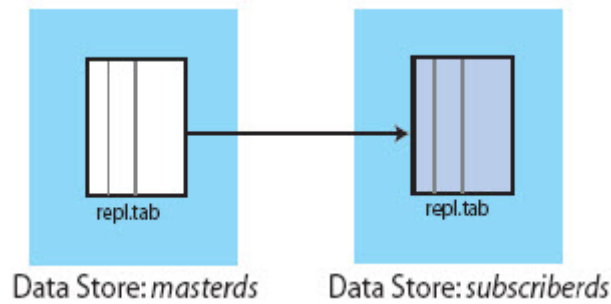
```
Driver=/oracle/TimesTen/tt70/lib/libtten.so
DataStore=/oracle/TimesTen/replication/masterds
PermSize=20
TempSize=20
DatabaseCharacterSet=K016MSWIN949
```

[subscriberds]

← masterds Data Source 정의 추가

```
Driver=/oracle/TimesTen/tt70/lib/libtten.so
DataStore=/oracle/TimesTen/replication/subscriberds
PermSize=20
TempSize=20
DatabaseCharacterSet=K016MSWIN949
```

- Table 과 Replication 스키마 생성



- repl.tab 테이블과 repl.repscheme Replication을 생성할 SQL 스크립트 생성

[ultra60:/oracle/TimesTen]\$vi repscheme.sql

```
create table repl.tab
(a number not null,
b number,
c char(8),
primary key (a));
```

```
create replication repl.repscheme
element e table repl.tab
master masterds
subscriber subscriberds;
```

- Master data source에서 repscheme.sql 실행

[ultra60:/oracle/TimesTen]\$ttisql -f repscheme.sql masterds

```
connect "DSN=masterds";
Connection successful:
DSN=masterds;UID=timesten;DataStore=/oracle/TimesTen/replication/masterds;DatabaseCharacterSet=K
016MSWIN949;ConnectionCharacterSet=US7ASCII;DRIVER=/oracle/TimesTen/tt70/lib/libtten.so;PermSize
=20;TempSize=20;TypeMode=0;
(Default setting AutoCommit=1)
```

```
run "repscheme.sql";
```

```
create table repl.tab
(a number not null,
b number,
c char(8),
primary key (a));
```

```
create replication repl.repscheme
element e table repl.tab
master masterds
subscriber subscriberds;
```

```
exit;
Disconnecting...
Done.
```

- Subscribers data source에서 repscheme.sql 실행

```
[ultra60:/oracle/TimesTen]$ttisql -f repscheme.sql subscribers
```

```
connect "DSN=subscribers";
```

```
Connection successful:
```

```
run "repscheme.sql";
```

```
create table repl.tab
```

```
(a number not null,
```

```
b number,
```

```
c char(8),
```

```
primary key (a));
```

```
create replication repl.repscheme
```

```
element e table repl.tab
```

```
master masters
```

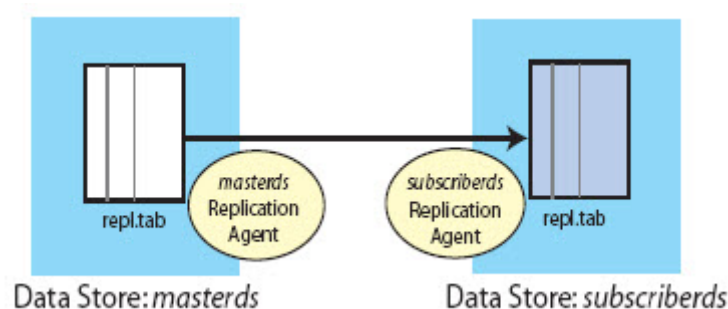
```
subscriber subscribers;
```

```
exit;
```

```
Disconnecting...
```

```
Done.
```

- Replication agent 시작



- Master data store에서 replication agent 시작

```
[ultra60:/oracle/TimesTen]$ttadmin -repstart masters
```

```
RAM Residence Policy      : inUse
```

```
Replication Agent Policy  : manual
```

```
Replication Manually Started : True
```

```
Cache Agent Policy        : manual
```

```
Cache Agent Manually Started : False
```

- Subscribers data store에서 replication agent 시작

```
[ultra60:/oracle/TimesTen]$ttadmin -repstart subscribers
```

```
RAM Residence Policy      : inUse
```

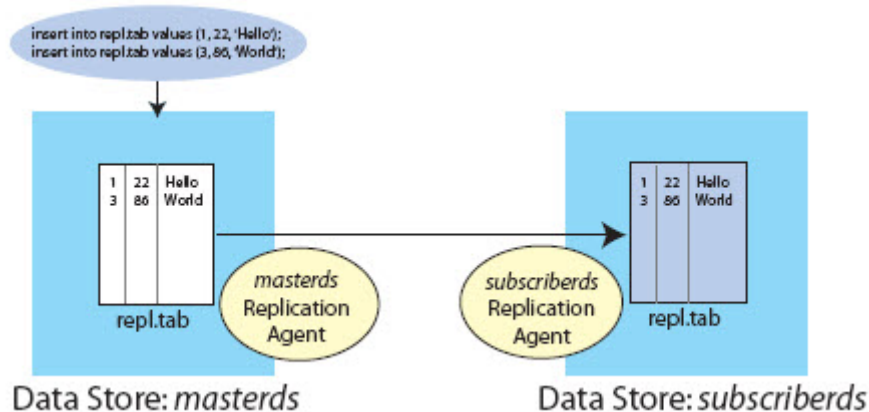
```
Replication Agent Policy  : manual
```

```
Replication Manually Started : True
```

```
Cache Agent Policy        : manual
```

```
Cache Agent Manually Started : False
```

- 복제된 테이블로 데이터 입력



- Master data store 의 repl.tab 테이블에 데이터 입력

```
[ultra60:/oracle/TimesTen]$ttisql masters
```

```
connect "DSN=mastersd";
Connection successful:
```

```
Command> insert into repl.tab values (1, 22, 'Hello');
1 row inserted.
Command> insert into repl.tab values (3, 86, 'World');
1 row inserted.
```

(Master에 입력된 데이터가 Subscriber로 적용되는데는 약간의 delay가 있을 수 있음)

- Subscriberds data store 의 repl.tab 테이블에서 데이터가 복제된 것을 확인

```
[ultra60:/oracle/TimesTen]$ttisql subscriberds
```

```
connect "DSN=subscriberds";
Connection successful:
```

```
Command> select * from repl.tab;
< 1, 22, Hello >
< 3, 86, World >
2 rows found.
```

- Replication 스키마와 테이블 삭제

- Master data store의 replication 정지

```
[ultra60:/oracle/TimesTen]$ttadmin -repstop masters
```

← master data store의 replication 정지

```
RAM Residence Policy      : inUse
Replication Agent Policy  : manual
Replication Manually Started : False
Cache Agent Policy        : manual
Cache Agent Manually Started : False
```

- Master data store의 replication 정지

```
[ultra60:/oracle/TimesTen]$ttadmin -repstop subscriberds
```

← Subscriber data store의 replication 정지

```
RAM Residence Policy      : inUse
Replication Agent Policy   : manual
Replication Manually Started : False
Cache Agent Policy        : manual
Cache Agent Manually Started : False
```

-

```
[ultra60:/oracle/TimesTen]$vi droprepscheme.sql ← replication 스키마 삭제 스크립트 작성
```

```
drop replication repl.repscheme;
drop table repl.tab;
```

- Master data store 에서 replication 스키마, 테이블 삭제

```
[ultra60:/oracle/TimesTen]$ttisql -f droprepscheme.sql masterds
```

```
connect "DSN=masterds";
Connection successful:
```

```
run "droprepscheme.sql";
```

```
drop replication repl.repscheme;
drop table repl.tab;
```

```
exit;
Disconnecting...
Done.
```

- Subscriber data store 에서 replication 스키마, 테이블 삭제

```
[ultra60:/oracle/TimesTen]$ttisql -f droprepscheme.sql subscriberds
```

```
connect "DSN=subscriberds";
Connection successful:
```

```
run "droprepscheme.sql";
```

```
drop replication repl.repscheme;
drop table repl.tab;
```

```
exit;
Disconnecting...
Done.
```

※ replication 스키마, 테이블 삭제시 반드시 replication 스키마를 먼저 삭제하고,
테이블을 나중에 삭제해야함.
그렇지 않으면 아래와 같은 에러 메시지가 나타나면서 삭제가 되지 않음.

“ 8002: Cannot drop replicated table, index, or sequence “

3. Oracle TimesTen의 관리

3.1. TimesTen 관리를 위한 주요 명령어

Name	Description
ttAdmin	TimesTen을 관리하기 위한 일반적인 유틸리티이며, Cache agent나 Replication agent의 작동뿐만 아니라 RAM으로부터 DB를 자동 또는 메뉴얼적으로 loading, unloading하는 정책을 지정하는데 주로 사용
ttBackup & ttRestore	DB의 백업본을 만들고 나중에 그것을 복원하는데 사용
ttBulkCp	TimesTen 테이블과 ASCII 파일간의 데이터 이동시 사용
ttIsql	TimesTen에서 SQL 구문을 실행하는데 사용
ttMigrate	TimesTen 테이블과 Cache group 정의를 binary data 파일로 저장하는데 사용
ttRepAdmin	Replication 상태를 모니터하는데 사용
ttSize	DB의 테이블별 할당공간을 측정하는데 사용
ttStatus	TimesTen의 현재 상태정보를 확인하는데 사용
ttTraceMon	TimesTen의 internal tracing 상태를 활성화, 비활성화 시키는데 사용
ttXactAdmin	각 transaction에 대한 ownership, status, log, lock 정보 확인하고, XA transaction을 commit, abort, forget 하는데 사용

3.2. TimesTen 관리를 위한 ttIsql 주요 명령어

3.2.1. ttIsql 기본 명령어

- ttIsql 접속하기

```
[ultra60:/oracle/TimesTen]$ttIsql
```

Copyright (c) 1996-2008, Oracle. All rights reserved.

Type ? or "help" for help, type "exit" to quit ttIsql.

All commands must end with a semicolon character.

Command> **connect dsn=ttdemo;** ← data source name이 'ttdemo'인 TimesTen DB로 접속

Connection successful:

DSN=ttdemo;UID=timesten;DataStore=/tmp/ttdemo;DatabaseCharacterSet=K016MSWIN949;ConnectionCharac
terSet=US7ASCII;DRIVER=/oracle/TimesTen/tt70/lib/libtten.so;PermSize=20;TempSize=20;TypeMode=0;
(Default setting AutoCommit=1)

Command>

- 온라인 도움말 사용하기

Command> **help connect** ← help 뒤에 사용법과 설명을 보고자 하는 명령어를 입력

All commands must end with a semicolon character.

Arguments in <> are required.

Arguments in [] are optional.

Command Usage: connect [DSN|connection_string] [as <connection_id>]

Command Aliases: (none)

Description: Connects to the data source specified by the optional DSN or connection string argument. If an argument is not given, then the DSN or connection string from the last successful connection is used. A connection ID

may optionally be specified, for use in referring to the connection when multiple connections are enabled. The DSN is used as the default connection ID. If that ID is already in use, the connection will be assigned the ID "conN", where N is some number larger than 0.

Requires an active connection: NO

Requires autocommit turned off: NO

Reports elapsed execution time: YES

Works only with a TimesTen data source: NO

Example: connect; -or- connect RunData; -or- connect "DSN=RunData;Overwrite=1";

-or- connect RunData as rundata1;

- 가장 최근에 사용한 10개 명령어 확인 (ttlsq1 모드에서 사용한 최근 100개까지 저장됨)

Command> h ;

```
5      insert into t2 values (2);
6      insert into t3 values (3);
7      insert into t4 values (4);
8      insert into t5 values (5);
9      insert into t6 values (6);
10     insert into t7 values (7);
11     insert into t8 values (8);
12     insert into t9 values (9);
13     insert into t10 values (10);
14     insert into t11 values (11);
```

Command> ! 5 ; ← 최근 사용한 명령어 목록 중 5번째 라인에 있는 명령어를 실행

```
insert into t2 values (2);
2206: Table TIMESTEN.T2 not found
The command failed.
```

Command> !! ; ← 가장 최근에 사용한 명령어 실행

```
insert into t2 values (2);
2206: Table TIMESTEN.T2 not found
The command failed.
```

Command> savehistory history.txt; ← 명령어 목록에 있는 명령어들을 파일로 저장함
만일 만들고자 하는 이름의 파일이 있을 경우
추가는 (savehistory -a 파일명), 덮어쓰기기는 (savehistory -f 파일명)

[ultra60:/oracle/TimesTen]\$vi history.txt

```
/* 1      */ conn dsn=ttdemo;
/* 2      */ connect dsn=ttdemo;
/* 3      */ insert into t1 values (1);
/* 4      */ /;
/* 5      */ insert into t2 values (2);
/* 6      */ insert into t3 values (3);
/* 7      */ insert into t4 values (4);
/* 8      */ insert into t5 values (5);
/* 9      */ insert into t6 values (6);
/* 10     */ insert into t7 values (7);
/* 11     */ insert into t8 values (8);
/* 12     */ insert into t9 values (9);
/* 13     */ insert into t10 values (10);
/* 14     */ insert into t11 values (11);
```

```
/* 15      */ insert into t2 values (2);
```

Command> **clearhistory;** ← ttlsq에 저장되어 있는 명령어 목록 삭제

Command> **h;**

Command>

- 테이블 목록 확인

Command> **tables;**

SYS.CACHE_GROUP

SYS.COLUMNS

SYS.COLUMN_HISTORY

SYS.COL_STATS

SYS.DUAL

SYS.INDEXES

SYS.MONITOR

SYS.OBJ_ACC_RIGHT

SYS.PLAN

SYS.SEQUENCES

SYS.SYNONYMS

SYS.SYS_ACC_RIGHT

SYS.TABLES

SYS.TABLE_HISTORY

SYS.TBL_STATS

SYS.TCOL_STATS

SYS.TINDEXES

SYS.TRANSACTION_LOG_API

SYS.TTABLES

SYS.TTBL_STATS

SYS.USERS

SYS.VIEWS

SYS.XLASUBSCRIPTIONS

TTREP.REPELEMENTS

TTREP.REPLICATIONS

TTREP.REPNETWORK

TTREP.REPPEERS

TTREP.REPSTORES

TTREP.REPSUBSCRIPTIONS

TTREP.REPTABLES

TTREP.TTSTORES

TIMESTEN.LOOKUP

TIMESTEN.T1

TIMESTEN.T2

TIMESTEN.T3

35 tables found.

3.2.2. Commit, Rollback, Sqlquerytimeout

- TimesTen의 commit, rollback, sqlquerytimeout 등에 대한 설명

```
Command> connect dsn=ttdemo;          ← ttdemo 라는 메모리DB에 접속
Connection successful:
DSN=ttdemo;UID=timesten;DataStore=/tmp/ttdemo;DatabaseCharacterSet=K016MSWIN949;ConnectionCharac
terSet=US7ASCII;DRIVER=/oracle/TimesTen/tt70/lib/libtten.so;PermSize=20;TempSize=20;TypeMode=0;
(Default setting AutoCommit=1)
Command> autocommit 0;                ← autocommit 옵션을 사용안함으로 수정
Command> create table lookup
> (key number not null primary key,
> value char (64));
Command> commit;
Command> insert into lookup values (1,'ABC');
1 row inserted.
Command> select * from lookup;
< 1, ABC                                >
1 row found.
Command> rollback;                    ← 이전 commit 상태로 되돌림
Command> select * from lookup;
0 rows found.                          ← 입력된 데이터가 rollback 되었음
Command> isolation;                  ← 현재의 isolation level 확인
isolation = READ_COMMITTED
Command> commitdurable;
Command> sqlquerytimeout 10;         ← SQL구문의 timeout 기간을 설정 (초단위)
                                       SQL구문의 실행시간이 설정된 timeout 기간을 초과하면,
                                       해당구문은 실행되지 않으며 일반적으로 6111 에러 메시지 발생
Command> sqlquerytimeout;            ← SQL구문의 timeout 기간을 확인
Query timeout = 10 seconds
Command> disconnect;
Disconnecting...
Command> exit
Done.
[ultra60:/oracle/TimesTen]$
```

3.2.3. Data Store 관련 정보 확인

- data store 정보 확인

```
Command> create table t1
> (key number not null primary key,
> value char (64));
Command> describe t1;                ← t1 테이블에 대한 정보확인

Table TIMESTEN.T1:
Columns:
  *KEY          NUMBER NOT NULL      ← '*'은 primary key column을 의미함
  VALUE         CHAR (64)

1 table found.
(primary key columns are indicated with *)
```

Command> **describe select * from t1 where key=?;**

Prepared Statement:

Parameters:

Parameter 1 NUMBER

Columns:

KEY NUMBER NOT NULL

VALUE CHAR (64)

Command> **describe ttoptuseindex;**

Procedure TTOPTUSEINDEX:

Parameters:

INDOPTION TT_VARCHAR (1024)

Columns:

(none)

1 procedure found.

Command> **dssize;** ← data source 크기정보 확인 (default는 KB)

PERM_ALLOCATED_SIZE: 20480

PERM_IN_USE_SIZE: 1639

PERM_IN_USE_HIGH_WATER: 1639

TEMP_ALLOCATED_SIZE: 20480

TEMP_IN_USE_SIZE: 4122

TEMP_IN_USE_HIGH_WATER: 4185

Command> **dssize m;** ← data source 크기정보를 MB 단위로 확인

PERM_ALLOCATED_SIZE: 20

PERM_IN_USE_SIZE: 1.6005859375

PERM_IN_USE_HIGH_WATER: 1.6005859375

TEMP_ALLOCATED_SIZE: 20

TEMP_IN_USE_SIZE: 4.025390625

TEMP_IN_USE_HIGH_WATER: 4.0869140625

Command> **monitor;** ← dssize의 정보를 포함하여 connection수, checkpoint, lock timeout, commit, rollback, memory load time 정보 등을 제공

TIME_OF_1ST_CONNECT: Thu Dec 25 11:10:15 2008

DS_CONNECTS: 7

DS_DISCONNECTS: 0

DS_CHECKPOINTS: 4

DS_CHECKPOINTS_FUZZY: 4

DS_COMPACTS: 0

PERM_ALLOCATED_SIZE: 20480

PERM_IN_USE_SIZE: 1639

PERM_IN_USE_HIGH_WATER: 1639

TEMP_ALLOCATED_SIZE: 20480

TEMP_IN_USE_SIZE: 4185

TEMP_IN_USE_HIGH_WATER: 4249

SYS18: 0

XACT_BEGINS: 68

XACT_COMMITS: 67

XACT_D_COMMITS:	0
XACT_ROLLBACKS:	0
LOG_FORCES:	4
DEADLOCKS:	0
LOCK_TIMEOUTS:	0
LOCK_GRANTS_IMMED:	481
LOCK_GRANTS_WAIT:	0
SYS19:	0
CMD_PREPARES:	21
CMD_REPREPARES:	0
CMD_TEMP_INDEXES:	0
LAST_LOG_FILE:	0
REPHOLD_LOG_FILE:	-1
REPHOLD_LOG_OFF:	-1
REP_XACT_COUNT:	0
REP_CONFLICT_COUNT:	0
REP_PEER_CONNECTIONS:	0
REP_PEER_RETRIES:	0
FIRST_LOG_FILE:	0
LOG_BYTES_TO_LOG_BUFFER:	22872
LOG_FS_READS:	0
LOG_FS_WRITES:	4
LOG_BUFFER_WAITS:	0
CHECKPOINT_BYTES_WRITTEN:	273672
SYS1:	24
SYS2:	24
SYS3:	0
SYS4:	0
SYS5:	0
SYS6:	0
SYS7:	4
SYS8:	0
REQUIRED_RECOVERY:	0
SYS11:	0
SYS12:	1
TYPE_MODE:	0
SYS13:	0
SYS14:	0
SYS15:	0
SYS16:	0
SYS17:	0
SYS9:	

3.2.4. Passthrough 설정 확인 및 변경

- passthrough 설정 확인 및 변경

```

Command> show passthrough;    ← passthrough 단계를 확인
PassThrough = 0
Command> autocommit;          ← autocommit 설정 확인
autocommit = 1 (ON)
Command> autocommit 0;        ← autocommit을 '0'으로 설정
                               (passthrough 단계를 수정하려면 autocommit은 반드시 '0'으로 설정되어야 함)
Command> autocommit;
autocommit = 0 (OFF)
Command> set passthrough 1;    ← passthrough 단계를 '1'로 수정
Command> show passthrough;
PassThrough = 1                ← passthrough 단계가 수정된 것을 확인
  
```

※ passthrough [0|1|2|3]

Cache Connect passthrough 단계를 설정

- 0 - TimesTen에서 모든 SQL 구문을 수행함 (default)
- 1 - target table을 TimesTen에서 발견할 수 없다면 Oracle DB에서 DML 구문을 수행함
- 2 - similar to level1 plus executes any DML statements targeted at READONLY cache groups on Oracle.
- 3 - 모든 SQL 구문을 Oracle DB에서 수행함

3.2.5. Query Optimizer 실행계획의 변경 및 확인

- Query Optimizer의 실행계획 변경에 사용되는 힌트

힌트	설명
optprofile	현재의 optimizer hint 설정과 join order를 보여줌
setjoinorder	join order를 설정
setuseindex	index hint를 설정
tryhash	hash index의 사용여부 설정
trymergejoin	merge join의 사용여부 설정
trynestedloopjoin	nested loop join의 사용여부 설정
tryserial	serial scan의 사용여부 설정
trytmphash	temporary hash index의 사용여부 설정
trytmptable	intermediate result table의 사용여부 설정
trytmpttree	temporary ttree index의 사용여부 설정
tryttree	ttree index의 사용여부 설정
tryrowid	rowid scan의 사용여부 설정
trytbllocks	table lock의 사용여부 설정
unsetjoinorder	join order의 설정내용 삭제
unseruseindex	index hint의 설정내용 삭제

- Hint 적용 및 실행계획 확인 예제

```
Command> create table t1 (a number);
Command> create table t2 (b number);
Command> create table t3 (c number);
Command>
Command> insert into t1 values (3);
1 row inserted.
Command> insert into t2 values (3);
1 row inserted.
Command> insert into t3 values (3);
1 row inserted.
Command> insert into t1 values (4);
1 row inserted.
Command> insert into t2 values (5);
1 row inserted.
Command> insert into t3 values (6);
1 row inserted.
Command> autocommit 0;
Command> showplan;           ← TimesTen에서 실행된 query optimizer 실행계획을 보여주는 명령어
Command> select * from t1, t2, t3 where a=b and b=c and a=b;
```

Query Optimizer Plan:

STEP:	1
LEVEL:	3
OPERATION:	TbLkSerialScan
TBLNAME:	T1
IXNAME:	<NULL>
INDEXED CONDITION:	<NULL>
NOT INDEXED:	<NULL>
STEP:	2
LEVEL:	3
OPERATION:	TbLkSerialScan
TBLNAME:	T2
IXNAME:	<NULL>
INDEXED CONDITION:	<NULL>
NOT INDEXED:	T1.A = T2.B AND T1.A = T2.B
STEP:	3
LEVEL:	2
OPERATION:	NestedLoop
TBLNAME:	<NULL>
IXNAME:	<NULL>
INDEXED CONDITION:	<NULL>
NOT INDEXED:	<NULL>
STEP:	4
LEVEL:	2
OPERATION:	TbLkSerialScan
TBLNAME:	T3
IXNAME:	<NULL>
INDEXED CONDITION:	<NULL>
NOT INDEXED:	T1.A = T3.C AND T2.B = T3.C


```

STEP:          5
LEVEL:         1
OPERATION:     NestedLoop
TBLNAME:       <NULL>
IXNAME:        <NULL>
INDEXED CONDITION: <NULL>
NOT INDEXED:   <NULL>

```

< 3, 3, 3 >

1 row found.

Command> **trytbllocks 0;**

Command> **tryserial 0;**

Command> **select * from t1, t2, t3 where a=b and b=c and a=b;**

Query Optimizer Plan:

```

STEP:          1
LEVEL:         3
OPERATION:     TmpTtreeScan    ← 힌트 적용 후 실행계획이 변경된 것을 확인
TBLNAME:       T1
IXNAME:        <NULL>
INDEXED CONDITION: <NULL>
NOT INDEXED:   <NULL>

```

```

STEP:          2
LEVEL:         3
OPERATION:     RowLkSerialScan ← 힌트 적용 후 실행계획이 변경된 것을 확인
TBLNAME:       T2
IXNAME:        <NULL>
INDEXED CONDITION: <NULL>
NOT INDEXED:   T1.A = T2.B AND T1.A = T2.B

```

```

STEP:          3
LEVEL:         2
OPERATION:     NestedLoop
TBLNAME:       <NULL>
IXNAME:        <NULL>
INDEXED CONDITION: <NULL>
NOT INDEXED:   <NULL>

```

```

STEP:          4
LEVEL:         2
OPERATION:     RowLkSerialScan ← 힌트 적용 후 실행계획이 변경된 것을 확인
TBLNAME:       T3
IXNAME:        <NULL>
INDEXED CONDITION: <NULL>
NOT INDEXED:   T1.A = T3.C AND T2.B = T3.C

```

```

STEP:          5
LEVEL:         1
OPERATION:     NestedLoop
TBLNAME:       <NULL>
IXNAME:        <NULL>

```

```
INDEXED CONDITION: <NULL>
NOT INDEXED: <NULL>
```

```
< 3, 3, 3 >
1 row found.
Command>
```

3.2.6. Timing ODBC function calls

- ODBC function call이 실행되는데 소요된 시간 측정

```
Command> timing 1;
```

```
Command> connect dsn=ttdemo;
```

Connection successful:

DSN=ttdemo;UID=timesten;DataStore=/tmp/ttdemo;DatabaseCharacterSet=K016MSWIN949;ConnectionCharacterSet=US7ASCII;DRIVER=/oracle/TimesTen/tt70/lib/libtten.so;PermSize=20;TempSize=20;TypeMode=0;
(Default setting AutoCommit=1)

Execution time (SQLDriverConnect) = 0.705767 seconds.

← SQLDriverConnect를 수행하는데 약 0.7초가 소요됨

- query 수행시간 측정

```
Command> verbosity 0; ← display할 결과를 프린트하고 포맷을 맞추는데 소요되는 시간 배제시킴
```

```
Command> select * from t1;
```

Execution time (SQLExecute + Fetch Loop) = 0.001045 seconds.

```
Command>
```

4. Oracle TimesTen 백업 및 복구

4.1.1. TimesTen 백업 Test를 위한 Access control 활성화

```
Command> create user kdw identified by 'kdw';
15007: Access control not enabled    ← Access control 기능이 enable 되지않아 유저생성 실패
The command failed.
Command> exit
Disconnecting...
Done.
```

- Access Control enable 시키기

```
[ultra60:/oracle/TimesTen/kdw]$ ttmodinstall
The daemon for instance 'tt70' is currently configured to use port 17001.
Would you like to change this port? [ no ]
The server for instance 'tt70' is currently configured to use port 17003.
Would you like to change this port? [ no ]
The webserver for instance 'tt70' is currently configured to use port 17005.
Would you like to change this port? [ no ]
ORACLE_HOME for the instance 'tt70' is currently set to '/oracle/10.2/app'.
Would you like to change ORACLE_HOME for this instance? [ no ]
Would you like to enable access control for this instance? [ no ] yes    ← Access control enable
NOTE: The daemon must be stopped before enabling access control.
Would you like to stop the daemon? [ yes ]
Enabling access control ...
Patching successful ...
Restarting the daemon ...
TimesTen Daemon startup OK.
Access control is now enabled for this TimesTen instance.
Done ...
```

```
[ultra60:/oracle/TimesTen/kdw]$ ttisql tt_test
Command> create user kdw identified by 'kdw';
Command> grant connect, ddl to kdw;    ← 유저에게 접속, DDL 권한 부여
Command> exit
```

```
[ultra60:/oracle/TimesTen/kdw]$ ttisql 'uid=kdw;pwd=kdw;dsn=tt_test';
Command> select user from dual;    ← 현재 접속한 User 를 확인
< KDW >
1 row found.
```

TimesTen 에서 백업은 ttBackup 유틸리티를 사용합니다.

ttBackup 은 백업이 시작된 시점부터의 data store 의 스냅샷 이미지를 생성합니다.

백업시 생성되는 파일은 *.sta, *.bac, *.bac0 입니다.

ttBackup 으로 생성된 백업 파일로 restore 를 하기 위해서는 ttRestore 유틸리티를 사용합니다

4.1.2. TimesTen 백업 및 복구 예제

아래는 ttBackup, ttRestore 를 사용한 백업, 복구 예제입니다.

```
Command> select * from kdw.kdw_test;                                ← kdw_test Table 확인
0 rows found.

- Backup 수행 -
[ultra60:/oracle/TimesTen/kdw]$ ttbackup -dir ./ -type filefull -fname kdw dsn=tt_test
Backup started ...
Backup complete

- DSN 삭제 -
[ultra60:/oracle/TimesTen/kdw]$ ttdestroy tt_test                  ← tt_test DSN 삭제
[ultra60:/oracle/TimesTen/kdw]$ ttisql tt_test
Command> select * from kdw.kdw_test;                                ← 테이블 확인
2206: Table KDW.KDW_TEST not found                                  ← Table 이 존재하지 않음
The command failed.

- Restore 수행 -
[ultra60:/oracle/TimesTen/kdw]$ ttrestore -fname kdw -dir ./ dsn=tt_test
Restore started ...
Restore complete
[ultra60:/oracle/TimesTen/kdw]$ ttisql tt_test
Command> select * from kdw.kdw_test;                                ← Table 확인
0 rows found.                                                       ← Table 생성된것 확인
```

ttBackup 에서 사용할 수 있는 백업 타입

fileFull : 단순한 Full 백업이며, Incremental 백업은 불가능합니다.

fileFullEnable : 이후 Incremental 백업이 가능하도록 Full 백업을 수행합니다.

fileIncremental : Incremental 백업을 수행합니다.

fileIncrOrFull : Incremental 백업이 가능할 시, Incremental 백업을 수행하며,
불가능하면 Full 백업을 수행합니다.

5. Tools for Troubleshooting TimesTen

5.1. ttStatus 유틸리티

```
[ultra60:/oracle/TimesTen]$ttdaemonadmin -stop      ← TimesTen daemon 을 멈춤
TimesTen Daemon stopped.
[ultra60:/oracle/TimesTen]$ttstatus                  ← TimesTen 상태 확인
ttStatus: Could not connect to the TimesTen daemon.
If the TimesTen daemon is not running, please start it
by running "ttDaemonAdmin -start".
[ultra60:/oracle/TimesTen]$ttdaemonadmin -start      ← TimesTen daemon 을 시작
TimesTen Daemon startup OK.

[ultra60:/oracle/TimesTen]$ttstatus                  ← TimesTen 상태 확인
TimesTen status report as of Sun Jan  4 16:46:04 2009

Daemon pid 8528 port 17001 instance tt70
TimesTen server pid 8534 started on port 17003
TimesTen webserver pid 8533 started on port 17005

-----
Data store /oracle/TimesTen/replication/masterds    ← masterds 데이터 스토어 확인
There are no connections to the data store
Replication policy   : Manual
Cache agent policy   : Manual
-----

Access control enabled.
End of report

- 다른 터미널에서 masterds 데이터 스토어에 접속
[ultra60:/oracle/TimesTen]$ttisql masterds          ← masterds 데이터 스토어에 접속

connect "DSN=masterds";

Connection                                                    successful:
DSN=masterds:UID=timesten:DataStore=/oracle/TimesTen/replication/masterds:DatabaseCharacterSet=K
016MSWIN949:ConnectionCharacterSet=US7ASCII:DRIVER=/oracle/TimesTen/tt70/lib/libtten.so:PermSize
=20;TempSize=20;TypeMode=0;
(Default setting AutoCommit=1)
Command>
```

- 원래 터미널창에서 TimesTen 상태 확인

[ultra60:/oracle/TimesTen]\$**ttstatus**

← TimesTen 상태 확인

TimesTen status report as of Sun Jan 4 16:48:59 2009

Daemon pid 8528 port 17001 instance tt70

TimesTen server pid 8534 started on port 17003

TimesTen webserver pid 8533 started on port 17005

Data store /oracle/TimesTen/replication/masterds

There are 7 connections to the data store

← masterds 에 7 개의 접속이 생긴 것을 확인

Data store is in shared mode

Shared Memory KEY 0x1800033b ID 3202

Type	PID	Context	Connection Name	ConnID
Process	8572	0x000000010022d390	masterds	1
Subdaemon	8531	0x000000010015ab00	Worker	2042
Subdaemon	8531	0x000000010026c5d0	Flusher	2045
Subdaemon	8531	0x000000010027d290	Checkpoint	2043
Subdaemon	8531	0x00000001002ee620	Aging	2047
Subdaemon	8531	0x000000010033f5a0	Monitor	2046
Subdaemon	8531	0x00000001003d07e0	HistGC	2044

Replication policy : Manual

Cache agent policy : Manual

Access control enabled.

End of report

5.2. ttCapture 유틸리티

-ttcapture 유틸리티는 시스템에 문제가 발생하였을 때 시스템의 상태를 Technical Support가 가능하도록 TimesTen 시스템 상태와 설정정보 등을 파일로써 캡처하는 유틸리티입니다. TimesTen data store 등에 문제가 발생한 시점에 ttcapture 프로그램을 실행해야 좀 더 정확한 문제해결이 가능합니다.

- ttcapture 유틸리티 실행

```
[ultra60:/oracle/TimesTen]$ttcapture masterds
```

```
Capturing information to file /oracle/TimesTen/ttcapture.20090104.1708.log
```

- ttcapture 유틸리티 실행 결과 파일 내용

```
[ultra60:/oracle/TimesTen]$vi /oracle/TimesTen/ttcapture.20090104.1708.log
```

```
==> uname -a
==> /oracle/TimesTen/tt70/bin/ttVersion
==> /oracle/TimesTen/tt70/bin/ttVersion -m
==> /oracle/TimesTen/tt70/bin/ttStatus
==> /bin/cat /etc/hosts
==> df -k /oracle/TimesTen/replication
==> /bin/ls -lR /oracle/TimesTen/replication
==> /oracle/TimesTen/tt70/unsupported/dsmap -connectOptions -hdr -dsname /oracle/
==> /bin/ls -lR /oracle/TimesTen/tt70/bin
==> /bin/ls -lR /oracle/TimesTen/tt70/lib
==> /bin/ls -lR /oracle/TimesTen/tt70/info
==> /bin/cat /oracle/TimesTen/.odbc.ini
==> /bin/cat /oracle/TimesTen/tt70/info/sys.odbc.ini
==> /bin/cat /oracle/TimesTen/tt70/info/.install_log
==> /bin/cat /etc/release
==> who -r
==> showrev
==> psrinfo -v
==> /usr/sbin/zoneadm list -cv
==> /usr/platform/SUNW,Ultra-60/sbin/prtdiag -v
==> ifconfig -a
==> /bin/cat /etc/inet/ipnodes
==> sysdef
==> mount -p
==> df -k
```

< 후 략 >

5.3. ttTraceMon 유틸리티

5.3.1. SQL Trace 사용하기

- SQL Trace 설정

```
[ultra60:/oracle/TimesTen]$tttracemon masterds
Trace monitor; empty line to exit
Trace> outfile sqltrace.txt          ← 결과 파일명 설정
Trace> level sql 4                   ← SQL Trace 레벨 설정
Trace> flush                         ← Trace buffer를 초기화
[ultra60:/oracle/TimesTen]$         ← 'Enter' 버튼을 사용하여 Trace mode를 빠져나옴
```

- masterds data store에서 SQL 구문 실행

```
[ultra60:/oracle/TimesTen]$ttisql masterds
```

```
Command> create table test (id number);
Command> insert into timesten.test values (1);
1 row inserted.
Command> select * from timesten.test;
< 1 >
1 row found.
```

- SQL Trace 파일 내용 확인

```
[ultra60:/oracle/TimesTen]$vi sqltrace.txt
18:31:51.877      103 SQL      2L      2C      9302P Preparing: select * from timesten.test
18:31:51.878      104 SQL      4L      2C      9302P sbSqlCmdCompile()(E): (Found already compiled
version: refCount:01, bucket:25) cmdType:100, cmdNum:1000122.
18:31:51.878      105 SQL      4L      2C      9302P Opening: select * from timesten.test;
18:31:51.878      106 SQL      4L      2C      9302P Fetching: select * from timesten.test;
18:31:51.878      107 SQL      4L      2C      9302P Fetching: select * from timesten.test;
18:31:51.878      108 SQL      4L      2C      9302P Closing: select * from timesten.test;
                                     ← SQL 구문이 실행단계별로 기록되고 있는 것을 확인 할 수 있음
```

- SQL Trace 모니터가 끝나면 Trace Level을 Default 값인 '0' 으로 재설정해주어야 함.

```
Trace> level sql 0
```

※ SQL Trace 레벨은 1~4 단계가 있으며, SQL Trace는 3 또는 4단계로 실행해야 자세한 SQL 구문 실행단계를 확인할 수 있다.

5.3.2. LOCK Trace 사용하기

- LOCK Trace 설정

```
[ultra60:/oracle/TimesTen]$tttracemon masterds
Trace> level lock 4                  ← LOCK Trace 레벨 설정
Trace> flush                         ← Trace buffer를 초기화
Trace> dump                          ← Trace buffer에 있는 내용 확인
0 records dumped
```

- masterds data store에서 SQL 구문 실행

```
[ultra60:/oracle/TimesTen]$ttisql masterds
Command> insert into timesten.test values (2);
1 row inserted.
```


- SQL 구문 실행 후 dump 확인

Trace> dump

```
18:47:56.912      119 SQL      2L      2C      9302P Preparing: insert into timesten.test values (2)
18:47:56.912      120 SQL      4L      2C      9302P sbSqlCmdCompile()(E): cmdType:100,
cmdNum: 1000123.
```

< 요약 >

```
18:47:56.916      141 LOCK      4L      2C      9302P ENQ: xcb:00002 <DB 0x1312d0001312d00,0x0>
0+IX=>GR activity 0 IX cnt=1
18:47:56.916      142 LOCK      4L      2C      9302P DEQ: xcb:00002 <DB 0x1312d0001312d00,0x0> All:Y,
mode=IX, cnt=1
34 records dumped
```

- LOCK Trace 모니터가 끝나면 Trace Level을 Default 값인 '0' 으로 재설정해주어야 함.

Trace> level lock 0

5.3.3. ERR Trace 사용하기

- ERR Trace 설정

[ultra60:/oracle/TimesTen]\$tttracemon masterds

Trace monitor; empty line to exit

Trace> level err 4

← ERROR Trace 레벨 설정

Trace> flush

← Trace buffer를 초기화

- ERR Trace 설정한 상태에서 에러 발생

[ultra60:/oracle/TimesTen]\$tttracemon masterds

Command> tables;

TIMESTEN.TEST

32 tables found.

Command> create table test (id number);

← 존재하고 있는 테이블을 다시 생성하려하여 에러 발생

2207: Table TEST already exists

The command failed.

Command> insert into test values 'abcd');

← insert 구문의 문법이 틀려 에러 발생

1001: Syntax error in SQL statement before or at: "abcd", character position: 26

insert into test values 'abcd');

^^^^

The command failed.

- 에러를 발생시킨 SQL 구문 실행 후 dump 확인

Trace> dump

```
18:58:53.054      215 ERR      4L      2C      9302P TT2207: Table TEST already
```

```
18:59:13.628      216 ERR      4L      2C      9302P TT1001: Syntax error in SQL statement before or
at: "abcd", character position: 26
```

```
insert into test values 'abcd');
```

^^^^

2 records dumped

※ ERR Trace Level에는 1과 4가 있으며, Default인 '1' 은 심각한 에러일때만 표시하며, '4' 는 발생하는 모든 에러에 대한 내용들을 표시한다.

6. Oracle TimesTen 활용 사례

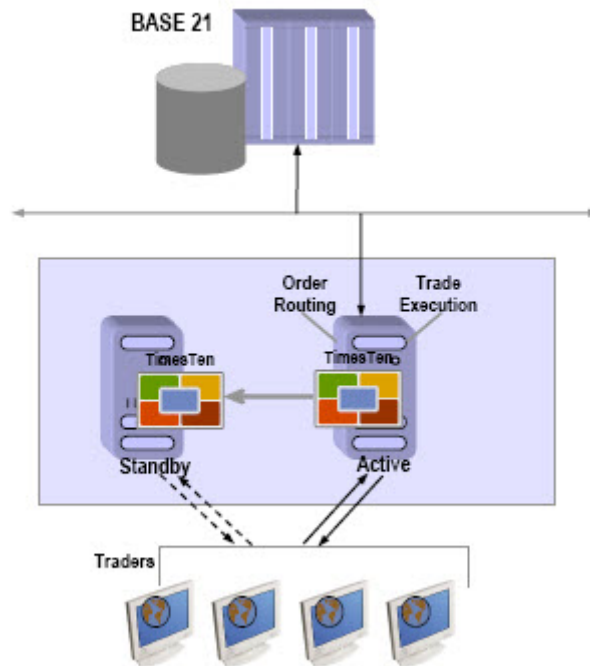
6.1. In-Memory Database의 일반적인 적용 분야

메모리 상에서 전체 데이터 베이스를 관리하므로 고성능의 실시간 transaction 처리를 필요로 하는 분야에 적합함.

- 높은 읽기/쓰기 처리량이 필요한 분야
- 고 가용성을 갖추고, 높은 읽기 처리량이 필요한 분야
예 : wireless billing, Stock trading
- 빠른 응답 시간이 필요한 분야
- 새로운 어플리케이션 또는 심각한 성능 문제를 안고 있는 기존 어플리케이션
- 기존 데이터베이스를 유지하면서, “hot” 데이터에 대한 캐시 기능이 필요한 부분



6.3. 금융 분야의 TimesTen 활용 사례



● 차세대 Trading 시스템

1. 현행 및 요구사항

- 시스템을 통한 단위시간당 집중도 증가
- 외부 시스템 성능개선에 따른 trading 시스템 주문 처리 속도 개선 요구
- H/W 증설시의 과도한 비용 증가 발생

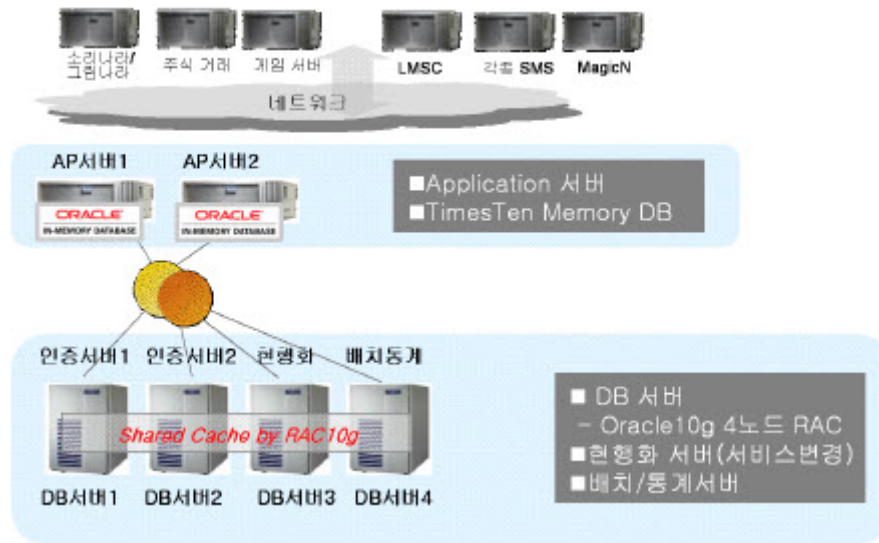
2. 구성환경

- 4-CPU * 2 System (for Replication)
- Sun/Solaris O/S
- C - applications (ODBC)
- Application에 embedded 되어 Direct-Data Access

3. 기대 효과

- 빠른 응답 속도와 안정성을 제공한 주문 처리 속도 개선 (10건/초 -> 100건/초)
- 기존 DBMS 대비 비용 절감
- 완벽한 실시간 이중화 및 장애 요인 (H/W Disk) 감소
- near-zero administration

6.4. 통신 분야의 TimesTen 활용 사례



● 신인증 시스템

1. 현행 및 요구사항

- 무선 contents 서비스를 위한 가입자/서비스 인증 기능 제공, 서비스 사용을 위한 정보 제공
- 무선 contents를 통한 부가 서비스 등록/해지 등의 기본 프로세스 처리
- 2-node TimesTen 서버 : 실시간 인증 MEMORY DB 서버
- 4-node Oracle 10g RAC DB 서버 : 인증 서버 #1, #2, 현행화 서버, 배치/통계서버

2. 시스템 구성

- Sun Solaris 10 SF25K 8cpu(16core) * 2ea
- TimesTen 7.0.2.4 (64bit)
- 각 100GB Data Store
- 2-node Replication (Active-Active)
- 16's Read-Only Cache Groups (with Oracle 10g 4Node RAC)
- Direct Access Mode

3. 개선된 성능

- 약 1000 TPS의 인증 transaction 요구 -> TimesTen 최대 10,000 TPS 성능 확보
- 일일 최대 8천만 건 처리 요구 -> 일일 최대 8억 6천만건 처리 가능
- 평균 30ms 이하의 처리 속도 요구 -> 평균 0.1ms 처리속도 확보

7. 참고 문헌 및 사이트

1. Oracle TimesTen In-Memory Database Installation Guide Release 7.0
2. Oracle TimesTen In-Memory Database Architectural Overview Release 6.0
3. Oracle TimesTen In-Memory Database Operations Guide Release 7.0
4. TimesTen Cache Connect to Oracle Guide Release 7.0
5. TimesTen to TimesTen Replication Guide Release 7.0
6. Using Oracle TimesTen to Accelerate the Oracle Database (An Oracle Technical White Paper)
7. Oracle TimesTen을 활용한 수강 신청 성능 고도화 방안 (Powerpoint)
8. www.oracle.com/technology/global/kr/index.html (Oracle Technology Network)
9. www.Dbguide.net (DB 구축, 운영 종합정보 사이트)