

Tibero 소개

Tibero 7



Copyright © 2022 TmaxTibero Co., Ltd. All Rights Reserved.

Copyright Notice

Copyright © 2022 TmaxTibero Co., Ltd. All Rights Reserved.

대한민국 경기도 성남시 분당구 황새울로258번길 29, BS 타워 9층 우)13595

Website

<http://www.tmaxtibero.com>

기술서비스센터

Tel : +82-1544-8629

E-Mail : info@tmax.co.kr

Restricted Rights Legend

All TmaxTibero Software (Tibero®) and documents are protected by copyright laws and international convention. TmaxTibero software and documents are made available under the terms of the TmaxTibero License Agreement and this document may only be distributed or copied in accordance with the terms of this agreement. No part of this document may be transmitted, copied, deployed, or reproduced in any form or by any means, electronic, mechanical, or optical, without the prior written consent of TmaxTibero Co., Ltd. Nothing in this software document and agreement constitutes a transfer of intellectual property rights regardless of whether or not such rights are registered) or any rights to TmaxTibero trademarks, logos, or any other brand features.

This document is for information purposes only. The company assumes no direct or indirect responsibilities for the contents of this document, and does not guarantee that the information contained in this document satisfies certain legal or commercial conditions. The information contained in this document is subject to change without prior notice due to product upgrades or updates. The company assumes no liability for any errors in this document.

이 소프트웨어(Tibero®) 사용설명서의 내용과 프로그램은 저작권법과 국제 조약에 의해서 보호받고 있습니다. 사용설명서의 내용과 여기에 설명된 프로그램은 TmaxTibero Co., Ltd.와의 사용권 계약 하에서만 사용이 가능하며, 사용설명서는 사용권 계약의 범위 내에서만 배포 또는 복제할 수 있습니다. 이 사용설명서의 전부 또는 일부분을 TmaxTibero의 사전 서면 동의 없이 전자, 기계, 녹음 등의 수단을 사용하여 전송, 복제, 배포, 2차적 저작물작성 등의 행위를 하여서는 안 됩니다.

이 소프트웨어 사용설명서와 프로그램의 사용권 계약은 어떠한 경우에도 사용설명서 및 프로그램과 관련된 지적 재산권(등록 여부를 불문)을 양도하는 것으로 해석되지 아니하며, 브랜드나 로고, 상표 등을 사용할 권한을 부여하지 않습니다. 사용설명서는 오로지 정보의 제공만을 목적으로 하고, 이로 인한 계약상의 직접적 또는 간접적 책임을 지지 아니하며, 사용설명서 상의 내용은 법적 또는 상업적인 특정한 조건을 만족시키는 것을 보장하지는 않습니다. 사용설명서의 내용은 제품의 업그레이드나 수정에 따라 그 내용이 예고 없이 변경될 수 있으며, 내용상의 오류가 없음을 보장하지 아니합니다.

Trademarks

Tibero® is a registered trademark of TmaxTibero Co., Ltd. Other products, titles or services may be registered trademarks of their respective companies.

Tibero®는 TmaxTibero Co., Ltd.의 등록 상표입니다. 기타 모든 제품들과 회사 이름은 각각 해당 소유주의 상표로서 참조용으로만 사용됩니다.

Open Source Software Notice

Some modules or files of this product are subject to the terms of the following licenses. : OpenSSL, RSA Data Security, Inc., Apache Foundation, Jean-loup Gailly and Mark Adler, Paul Hsieh's hash

Detailed Information related to the license can be found in the following directory : \${INSTALL_PATH}/license/oss_licenses

본 제품의 일부 파일 또는 모듈은 다음의 라이선스를 준수합니다. : OpenSSL, RSA Data Security, Inc., Apache Foundation, Jean-loup Gailly and Mark Adler, Paul Hsieh's hash

관련 상세한 정보는 제품의 다음의 디렉터리에 기재된 사항을 참고해 주십시오. : \${INSTALL_PATH}/license/oss_licenses

안내서 정보

안내서 제목: Tibero 소개

발행일: 2024-04-30

소프트웨어 버전: Tibero 7.2.1

안내서 버전: v7.2.1

내용 목차

안내서에 대하여	ix
제1장 Tibero 소개	1
1.1. 개요	1
1.2. 주요 기능	1
1.3. 데이터베이스로서의 기본 기능	3
1.4. 프로세스 구조	4
1.4.1. 리스너	5
1.4.2. 워커 프로세스	5
1.4.3. 백그라운드 프로세스	7
제2장 Tibero 환경	9
2.1. 관리 툴	9
2.1.1. tbSQL	9
2.1.2. T-Up	10
2.1.3. tbExport	10
2.1.4. tbImport	12
2.1.5. tbLoader	12
2.2. 디렉터리 구조	13
2.3. 설치 환경	18
2.4. 사용자 환경설정 파일의 적용 및 확인	19
제3장 Tibero 안내서 구성	21
3.1. 개요	21
3.2. 안내서 목록	21

그림 목차

[그림 1.1] Tibero 프로세스 구조	4
-------------------------------	---

안내서에 대하여

안내서의 대상

본 안내서는 Tibero[®](이하 Tibero)를 사용하여 데이터베이스를 생성하고 원활한 Tibero의 동작을 보장하려는 데이터베이스 관리자(Database Administrator, 이하 DBA)를 대상으로 기술한다.

안내서의 전제 조건

본 안내서는 Tibero를 관리하는 방법을 설명한 안내서이다.

따라서 본 안내서를 원활히 이해하기 위해서는 다음과 같은 사항을 미리 알고 있어야 한다.

- 데이터베이스의 이해
- RDBMS의 이해
- 운영체제 및 시스템 환경의 이해
- UNIX 계열(Linux 포함)의 기본 지식

안내서의 제한 조건

본 안내서는 Tibero를 실무에 적용하거나 운용하는 데 필요한 모든 사항을 포함하고 있지 않다. 따라서 설치, 환경설정 등 운용 및 관리에 대해서는 각 제품 안내서를 참고하기 바란다.

참고

Tibero의 설치 및 환경설정에 관한 내용은 "Tibero 설치 안내서"를 참고한다.

안내서 구성

Tibero 소개 안내서는 총 3개의 장으로 구성되어 있다.

각 장의 주요 내용은 다음과 같다.

- 제1장: Tibero 소개

DBA 측면에서 Tibero의 기본 개념과 이를 구성하는 프로세스와 디렉터리 구조를 기술한다.

- 제2장: Tibero 환경

Tibero 환경을 위한 관리 툴, 디렉터리, 환경변수, 환경설정 파일에 대해서 설명한다.

- 제3장: Tibero 안내서 구성

Tibero 안내서에 대한 내용을 간략히 소개한다.

안내서 규약

표기	의미
<<AaBbCc123>>	프로그램 소스 코드의 파일명
<Ctrl>+C	Ctrl과 C를 동시에 누름
[Button]	GUI의 버튼 또는 메뉴 이름
진하게	강조
" "(따옴표)	다른 관련 안내서 또는 안내서 내의 다른 장 및 절 언급
'입력항목'	화면 UI에서 입력 항목에 대한 설명
하이퍼링크	메일 계정, 웹 사이트
>	메뉴의 진행 순서
+----	하위 디렉터리 또는 파일 있음
----	하위 디렉터리 또는 파일 없음
<u>참고</u>	참고 또는 주의사항
<u>주의</u>	주의할 사항
[그림 1.1]	그림 이름
[예 1.1]	예제 이름
AaBbCc123	Java 코드, XML 문서
[command argument]	옵션 파라미터
< xyz >	'<'와 '>' 사이의 내용이 실제 값으로 변경됨
	선택 사항. 예) A B: A나 B 중 하나
...	파라미터 등이 반복되어서 나옴
\${ }	환경변수

시스템 사용 환경

	요구 사항
Platform	HP-UX 11i v3(11.31)
	Solaris (Solaris 11)
	AIX (AIX 7.1/AIX 7.2/AIX 7.3)
	GNU (X86, 64, IA64)
	Red Hat Enterprise Linux 7 kernel 3.10.0 이상
	Windows(x86) 64bit
Hardware	최소 2.5GB 하드디스크 공간
	1GB 이상 메모리 공간
Compiler	PSM (C99 지원 필요)
	tbESQL/C (C99 지원 필요)

제1장 Tiberio 소개

본 장에서는 Tiberio의 기본 개념과 기능 프로세스 구조에 대해서 설명한다.

1.1. 개요

현재 기업의 비즈니스는 폭발적인 데이터의 증가와 다양한 환경 및 플랫폼의 등장으로 빠르게 확장되고 있다. 새로운 비즈니스 환경이 도래함에 따라 보다 더 효율적이고 유연한 데이터 서비스와 정보의 처리, 데이터 관리 기능이 필요하게 되었다.

Tiberio는 이러한 변화에 맞춰 기업 비즈니스 구현의 기반이 되는 데이터베이스 인프라 구성을 지원하며 고성능, 고가용성 및 확장성의 문제를 해결하는 엔터프라이즈 데이터베이스 관리 시스템이다.

기존 DB의 단점을 보완하기 위해 Tiberio는 독자적인 Tiberio Thread Architecture를 채택하고 구현하였다. 한정된 서버 프로세스의 CPU 및 메모리 등의 시스템 리소스를 효율적으로 사용하면서 뛰어난 성능과 안정성 및 확장성을 보장하고 편리한 개발 환경과 관리 기능을 제공한다. Tiberio는 초기 설계부터 대규모 사용자, 대용량 데이터, 강화된 안정성, 향상된 호환성 측면 등에서 다른 DBMS와 차별화를 고려하여 개발되었다.

Tiberio는 이처럼 기업이 원하는 최적의 데이터베이스 환경을 제공하는 대표적인 DB이다.

1.2. 주요 기능

대용량의 데이터를 관리하고 안정적인 비즈니스의 연속성을 보장하는 데이터 관리 솔루션인 Tiberio는 DB 환경에서 요구되는 주요 기능을 다음과 같이 갖추고 있다.

- **분산 데이터베이스 링크(Distributed Database Link)**

데이터베이스 인스턴스별로 각각 서로 다른 데이터를 저장하는 기능이다. 이 기능을 통해 원격 데이터베이스에 저장된 데이터를 네트워크를 통해 읽기 및 쓰기를 수행할 수 있다. 또한 이 기능은 다양한 벤더의 DB 제품을 연결하여 읽기 및 쓰기를 수행할 수 있다.

- **데이터 이중화(Data Replication)**

현재 운영 중인 데이터베이스에서 변경된 모든 내용을 Standby DB로 복제하는 기능이다. 즉, 네트워크를 통해서 변경 로그(Change log)만 전송하면 Standby DB에서 데이터에 적용하는 방식이다.

- **데이터베이스 클러스터(Database Cluster)**

기업용 DB의 최대 이슈인 고가용성과 고성능을 모두 해결하는 기능이다. Tiberio는 고가용성과 고성능을 보장하기 위해 **Tiberio Active Cluster** 기술을 보유하고 있다.

이 기술로 인해 여러 개의 데이터베이스 인스턴스가 공유 디스크를 이용하여 동일한 데이터베이스를 공유할 수 있다. 이때 각 데이터베이스 인스턴스는 내부의 데이터베이스 캐시(Database cache) 사이의 일관성을 유지하는 기술이 매우 중요하다. 따라서 이러한 기술도 Tibero Active Cluster에 포함되어 제공하고 있다. 보다 자세한 내용은 "Tibero 관리자 안내서"의 "제15장 Tibero Active Cluster"를 참고한다.

● 병렬 쿼리 처리(Parallel Query Processing)

기업의 데이터 크기는 계속적으로 증가하고 있다. 대용량 데이터를 처리하기 위해 서버의 리소스를 최대한 활용할 수 있는 병렬 처리 기술이 필수적으로 요구되고 있다.

Tibero는 이러한 요구사항에 맞추어 온라인 트랜잭션 처리(On-Line Transaction Processing, 이하 OLTP) 환경에 최적화된 기능을 제공할 뿐만 아니라 OLAP(On-Line Analytical Processing) 환경에 최적화된 SQL 병렬 처리 기능을 제공하고 있다. 이로 인해 쿼리는 빠른 응답 속도로 수행되며 기업의 빠른 의사 결정을 돕는다.

● 쿼리 최적화기(Optimizer)

쿼리 최적화기는 스키마 객체의 통계 정도를 바탕으로 다양한 데이터 처리 경로들을 고려하여 어떤 실행 계획이 가장 효율적인지를 결정한다.

쿼리 최적화기는 논리적으로 다음과 같은 단계를 통해 수행된다.

1. 주어진 SQL 문을 처리하는 다양한 실행 계획들을 만들어 낸다.
2. 데이터의 분산도에 대한 통계 정보와 테이블, 인덱스, 파티션 등의 특징을 고려하여 각각의 실행 계획의 비용을 계산한다. 여기서 비용이란 특정 실행 계획을 수행하는 데 필요한 상대 시간을 나타내고, 최적화기는 I/O, CPU, 메모리 등의 컴퓨터 자원을 고려하여 그 비용을 계산해 낸다.
3. 실행 계획들의 비용을 비교하여 가장 비용이 작은 계획을 선택한다.

쿼리 최적화기의 주요 기능은 다음과 같다.

– 최적화 목표

사용자는 최적화기의 최종 목표를 바꿀 수도 있는데 다음의 두 가지 목표를 선택할 수 있다.

구분	설명
전체 처리 시간	ALL_ROWS 힌트를 사용하면 최적화기는 마지막 row까지 얻어내는 시간을 최대한 단축하도록 최적화한다.
최초 반응 시간	FIRST_ROWS 힌트를 사용하면 최적화기는 첫 번째 row를 얻어내는 시간을 최대한 단축하도록 최적화한다.

– 질의 변형

질의의 형태를 바꿔서 더 좋은 실행 계획을 만들 수 있도록 한다. 질의 변형의 예에는 뷰 병합, 부질의 언네스트, 실체화 뷰의 사용 등이 있다.

– 데이터 접근 경로 결정

데이터를 데이터베이스로부터 꺼내오는 작업은 전체 테이블 스캔, 인덱스 스캔, rowid 스캔 등의 다양한 방법을 통해서 수행될 수 있다. 각 방법마다 필요한 데이터의 양이나 필터링의 형태 등에 따라 장단점이 있어서 질의에 따라 최적의 접근 방식이 다르다.

– 조인 처리 방식 결정

여러 테이블에서 데이터를 꺼내오게 되는 조인의 경우 최적화기는 조인의 순서와 방법을 결정해야 한다. 여러 테이블 간의 조인일 때 어떤 테이블을 먼저 조인할지에 대한 순서와 각각의 조인에 있어서 중첩 루프 조인, 합병 조인, 해시 조인 등의 다양한 방법 중 어떤 것을 사용할지가 실행 속도에 큰 영향을 미치게 된다.

– 비용 추정

각각의 실행 계획에 대해 비용을 추정한다. 비용 추정을 위해서 필요한 predicate의 선택도나 각 실행 단계에서 데이터의 row 수 등을 통계 정보를 사용해서 추정하고 이를 바탕으로 각 단계에서의 비용을 추정한다.

1.3. 데이터베이스로서의 기본 기능

Tibero는 데이터베이스의 영속성과 일관성을 유지하기 위하여 SQL 문장의 묶음인 트랜잭션을 다음의 4가지 성질을 통해 보장한다.

- Atomicity

All-or-nothing. 즉, 트랜잭션이 행한 모든 일이 적용되던가 아니면 모두 적용되지 않아야 함을 의미한다. Tibero에서는 이를 위하여 undo 데이터를 사용한다.

- Consistency

트랜잭션이 데이터베이스의 Consistency를 깨뜨리는 일은 여러 방면에서 생겨날 수 있다. 간단한 예는 테이블과 인덱스 간에 서로 다른 내용을 담고 있어서 Consistency가 깨지는 것이다. 이를 막기 위해 Tibero에서는 트랜잭션이 적용한 일들 중 일부만 자신이나 남에게 적용되는 것을 막고 있다. 즉, 테이블만 수정했고 아직 인덱스를 수정하지 않은 상태라고 해도 다른 트랜잭션에서는 이를 예전 모습으로 돌려 보아서 테이블과 인덱스가 항상 Consistency가 맞는 형태로 보이게 된다.

- Isolation

트랜잭션은 혼자만 돌고 있는 것처럼 보이게 된다. 물론 다른 트랜잭션이 수정한 데이터에 접근할 때는 이를 기다릴 수는 있지만 다른 트랜잭션이 수정 중이므로 접근할 수 없다고 에러가 나지는 않는다. 이를 위해 Tibero에서는 Multi version concurrency control 기법과 row-level locking 기법을 사용한다.

데이터를 참조하는 경우에는 MVCC 기법을 이용하여 다른 트랜잭션과 무관하게 참조 가능하며, 데이터를 수정할 때도 row level의 fine-grained lock control을 통하여 최소한의 충돌만을 일으키고 같은 데이터에 접근한다고 해도 단지 기다림으로써 이를 해결한다.

- Durability

Tibero에서는 이를 위하여 Redo 로그와 write-ahead logging 기법을 사용한다.

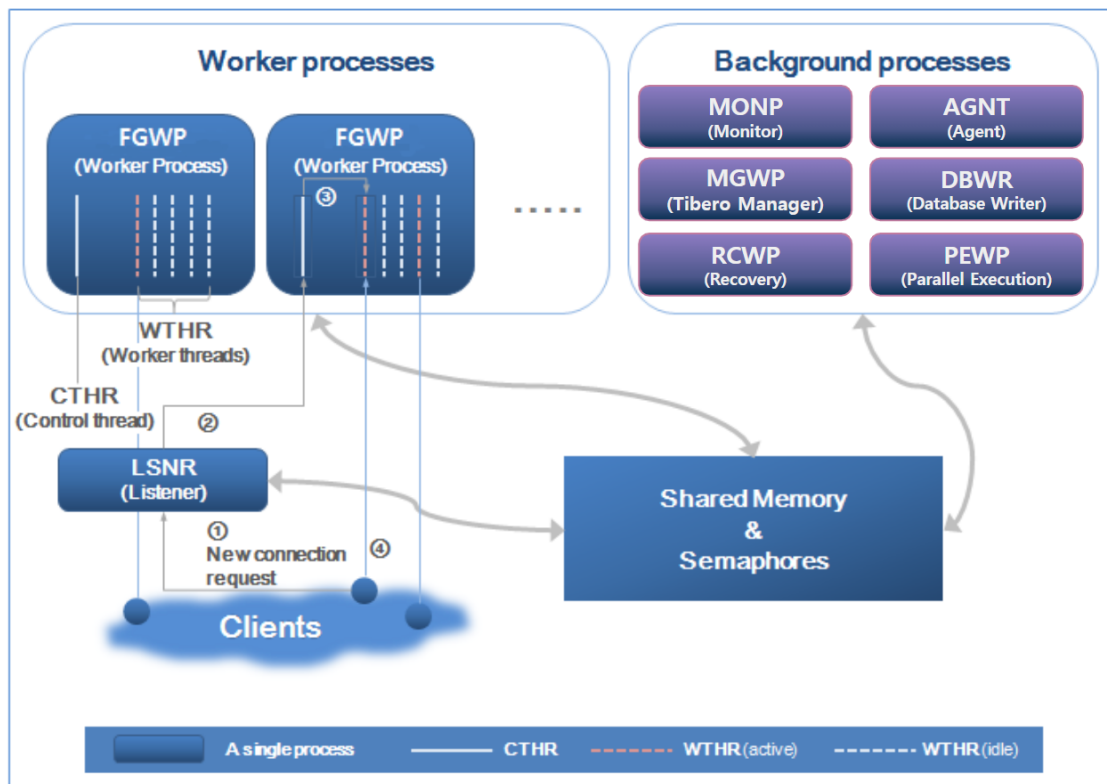
트랜잭션이 커밋할 때에 해당 Redo 로그가 디스크에 기록되어 트랜잭션의 영속성을 보장해 준다. 또한 블록이 디스크에 내려가기 전에 항상 Redo 로그가 먼저 내려가서 데이터베이스 전체가 일관성을 지니게 한다.

1.4. 프로세스 구조

Tibero는 대규모 사용자 접속을 수용하는 다중 프로세스 및 다중 스레드 기반의 아키텍처를 갖추고 있다.

다음은 Tibero의 프로세스 구조를 나타내는 그림이다.

[그림 1.1] Tibero 프로세스 구조



Tibero의 프로세스는 크게 3가지로 구성된다.

- 리스너(Listener)
- 워커 프로세스(Worker Process 또는 Foreground Process)
- 백그라운드 프로세스(Background Process)

1.4.1. 리스너

리스너(Listener)는 클라이언트의 새로운 접속 요청을 받아 이를 유향한 워커 프로세스에 할당한다. 즉, 클라이언트와 워커 프로세스간의 중계 역할을 담당하며 이는 별도의 실행 파일인 **tblistener**를 사용하여 작업을 수행한다. Tibero 6부터 MONP에 의해서 생성되며 외부에서 강제 종료하더라도 다시 생성된다.

다음은 [\[그림 1.1\]](#)를 기준으로 클라이언트의 새로운 접속 요청이 이루어지는 순서이다.

1. 현재 유향한 워커 스레드가 있는 워커 프로세스를 찾아서 클라이언트의 접속 요청(①)을 한다.
2. 이때 File descriptor와 함께 할당되므로 클라이언트는 서버의 내부 동작과 상관없이 마치 처음부터 워커 스레드에 접속한 것처럼 동작하게 된다.
3. 리스너의 요청을 받은 컨트롤 스레드(CTHR: control thread)는 자기 자신에 속한 워커 스레드의 상태를 검사(②)하여 현재 유향한 워커 스레드에 클라이언트의 접속을 할당(③)한다.
4. 할당된 워커 스레드는 클라이언트와 인증 절차를 거친 후 세션을 시작(④)한다.

1.4.2. 워커 프로세스

워커 프로세스(Worker Process)는 클라이언트와 실제로 통신을 하며 사용자의 요구 사항을 처리하는 프로세스이다. 이 프로세스의 개수는 WTHR_PROC_CNT 초기화 파라미터로 조절할 수 있으며, 일단 Tibero가 기동된 뒤에는 변경할 수 없다. 따라서 시스템 환경을 고려하여 적절한 값을 설정해야 한다.

Tibero 6부터 워커 프로세스는 용도에 따라 두 그룹으로 나눌 수 있다. 포어그라운드 워커 프로세스(Foreground Worker Process)는 리스너를 통해 들어온 온라인 요청을 처리하는 반면, 백그라운드 워커 프로세스(Background Worker Process)는 인터널 태스크(Internal Task)나 잡 스케줄러에 등록된 배치 작업을 수행한다. 그룹은 MAX_BG_SESSION_COUNT 초기화 파라미터로 조절할 수 있다.

참고

초기화 파라미터에 대한 자세한 내용은 "Tibero 참조 안내서"를 참고한다.

Tibero는 효율적인 리소스의 활용을 위해 **스레드(Thread)** 기반으로 작업을 수행한다. Tibero를 설치하면 기본적으로 하나의 워커 프로세스 안에는 1개의 컨트롤 스레드와 10개의 워커 스레드가 존재한다.

프로세스당 워커 스레드 개수는 WTHR_PER_PROC 초기화 파라미터로 조절할 수 있으며, WTHR_PROC_CNT 처럼 일단 Tibero가 기동된 뒤에는 변경할 수 없다. 따라서 시스템 환경을 고려하여 적절한 값을 설정해야 한다.

WTHR_PROC_CNT와 WTHR_PER_PROC 초기화 파라미터 값을 직접 바꾸는 것보다는 MAX_SESSION_COUNT 초기화 파라미터를 통해 서버에서 제공하는 최대 세션 개수를 지정할 것을 권장한다.

MAX_SESSION_COUNT 값에 따라 WTHR_PROC_CNT와 WTHR_PER_PROC 값이 자동으로 설정된다. 만약 WTHR_PROC_CNT와 WTHR_PER_PROC를 직접 설정할 경우 $WTHR_PROC_CNT * WTHR_PER_PROC$ 값이 MAX_SESSION_COUNT 값과 같게 두 값을 설정해야 한다.

MAX_BG_SESSION_COUNT 값은 MAX_SESSION_COUNT 보다 작은 값을 설정해야하며 WTHR_PER_PROC 값의 배수가 되어야 한다.

워커 프로세스는 컨트롤 스레드와 워커 스레드를 통해 작업을 수행한다.

● 컨트롤 스레드

워커 프로세스마다 하나씩 존재하며 다음과 같은 역할을 담당한다.

- Tibero가 기동될 때 초기화 파라미터에 설정된 수만큼 워커 스레드를 생성한다.
- 클라이언트의 새로운 접속 요청이 오면 현재 유향한 워커 스레드에 클라이언트의 접속을 할당한다.
- 시그널 처리를 담당한다.
- Tibero 6부터 I/O multiplexing을 지원하며 필요한 경우 워커 스레드 대신 메시지를 보내거나 받는 역할을 수행한다.

● 워커 스레드

워커 스레드는 클라이언트와 1:1로 통신하며, 클라이언트가 보내는 메시지를 받아 처리하고, 그 결과를 돌려준다. 주로 SQL 파싱, 최적화 수행 등 DBMS가 하는 작업 대부분이 워커 프로세스에서 일어난다.

그리고 워커 스레드는 하나의 클라이언트와 접속하므로 Tibero에 동시 접속이 가능한 클라이언트 수는 $WTHR_PROC_CNT * WTHR_PER_PROC$ 이다. Tibero는 세션 멀티플렉싱(Session multiplexing)을 지원하지 않으므로 하나의 클라이언트 접속은 곧 하나의 세션과 같다. 그러므로 최대 세션이 생성될 수 있는 개수는 $WTHR_PROC_CNT * WTHR_PER_PROC$ 를 연산한 값과 같다.

워커 스레드는 클라이언트와 접속이 끊긴다고 해도 없어지지 않으며, Tibero가 기동될 때 생성된 이후부터 종료할 때까지 계속 존재하게 된다. 이러한 구조에서는 클라이언트와 접속을 빈번하게 발생시키더라도 매번 스레드를 생성하거나 제거하지 않으므로 시스템의 성능을 높일 수 있다.

반면 실제 클라이언트의 수가 적더라도 초기화 파라미터에 설정된 개수만큼 스레드를 생성해야 하므로 운영체제의 리소스를 계속 소모하는 단점은 있으나, 운영체제 대부분이 유향한 스레드 하나를 유지하는 데 드는 리소스는 매우 적으므로 시스템을 운영하는 데는 별 무리가 없다.

주의

많은 수의 워커 스레드가 동시에 작업을 수행하려고 할 때 심한 경우 운영체제에 과도한 부하를 일으켜 시스템 성능이 크게 저하될 수 있다. 그러므로 대규모 시스템을 구축할 경우에는 Tibero와 클라이언트의 애플리케이션 프로그램 사이에 미들웨어를 설치하여 3-Tier 구조로 시스템을 구축할 것을 권장한다.

1.4.3. 백그라운드 프로세스

백그라운드 프로세스(Background Process)는 클라이언트의 접속 요청을 직접 받지 않고 워커 스레드나 다른 백그라운드 프로세스가 요청할 때 또는 정해진 주기에 따라 동작하는 주로 시간이 오래 걸리는 디스크 작업을 담당하는 독립된 프로세스이다.

백그라운드 프로세스에 속해 있는 프로세스는 다음과 같다.

- Monitor Process(MONP: 감시 프로세스)

Tibero 6부터 영문 약자가 스레드에서 프로세스로 변경되었으며 실제로 하나의 독립된 프로세스이다. Tibero가 기동할 때 최초로 생성되며 Tibero가 종료하면 맨 마지막에 프로세스를 끝마친다.

Tibero가 기동할 때 리스너를 포함한 다른 프로세스를 생성하거나 주기적으로 각 프로세스의 상태를 점검하는 역할을 담당한다. 또한 교착 상태(deadlock)도 검사한다.

- Tibero 매니저 프로세스(MGWP)

시스템을 관리하기 위한 용도의 프로세스이다. 관리자의 접속 요청을 받아 이를 시스템 관리 용도로 예약된 워커 스레드에 접속을 할당한다. 기본적으로 워커 프로세스와 동일한 역할을 수행하지만 리스너를 거치지 않고 스페셜 포트를 통해 직접 접속을 처리한다. SYS 계정만 접속이 허용된다.

- Agent Process(AGNT: 에이전트 프로세스)

시스템 유지를 위해 주기적으로 처리해야 하는 Tibero 내부의 작업을 담당한다.

Tibero 4 SP1까지는 시퀀스 캐시(sequence cache)의 값을 디스크에 저장하는 작업도 담당했으나, Tibero 5 이후로 각 워커 스레드가 담당하는 것으로 변경되었다. 이전에는 SEQW라는 명칭을 사용했으나 Tibero 6부터 AGNT로 명칭이 변경되었다. 시퀀스를 사용하는 방법은 "Tibero 관리자 안내서"의 "4.7.1. 시퀀스 생성, 변경, 제거"를 참고한다.

Tibero 6부터 다중 스레드(Multi-threaded) 기반 구조로 동작하며, 서로 다른 용도의 업무를 스레드별로 나누어 수행한다.

- DataBase Wirte Process(DBWR: 데이터베이스 쓰기 프로세스)

데이터베이스에서 변경된 내용을 디스크에 기록하는 일과 연관된 스레드들이 모여 있는 프로세스이다. 사용자가 변경한 블록을 디스크에 주기적으로 기록하는 스레드, Redo 로그를 디스크에 기록하는 스레드, 이 두 스레드를 통해 데이터베이스의 체크포인트 과정을 관할하는 체크포인트 스레드 등이 이 프로세스에 포함되어 있다.

- Recovery Worker Process(RCWP: 리커버리 워커 프로세스)

리커버리와 백업을 담당하는 프로세스이다. 부팅과정에서 NOMOUNT 이후의 부팅 단계를 올리는 역할을 수행하고 리커버리가 필요한 상황인지 여부를 판단하여 필요할 경우 리두로그를 읽어서 리커버리를 진행한다. tbrmgr을 이용하여 백업 작업을 하거나 백업본을 이용하여 미디어 리커버리를 진행하는 경우에도 이 프로세스에서 진행한다.

- Parallel Execution Worker Process(PEWP: Parallel Execution 워커 프로세스)

Parallel Execution Process(이하 PEP)는 PE 수행을 위해 도입된 PE 전용 프로세스이다. PE SQL을 처리할 때에 locality를 극대화하기 위해서 WTHR들을 하나의 PEP에서 할당한다. 또한 일반적인 클라이언트 세션을 위한 WTHR과 분리되어 모니터링 및 관리가 용이하다.

제2장 Tibero 환경

본 장에서는 Tibero 환경을 위한 관리 툴, 디렉터리, 환경변수, 환경설정 파일에 대해서 설명한다.

2.1. 관리 툴

본 절에서는 Tibero 환경을 위한 관리 툴에 대해서 설명한다.

2.1.1. tbSQL

tbSQL은 Tibero[®](이하 Tibero)에서 제공하는 SQL 문장을 처리하는 대화형 유틸리티이다. 이 유틸리티로 SQL 질의, 데이터 정의어(DDL: Data Definition Language), 트랜잭션과 관련된 SQL 문장을 실행할 수 있다. 또한, PSM 프로그램을 생성하고 실행할 수 있으며 DBA는 Tibero의 시스템 관리를 위한 명령을 실행할 수 있다.

tbSQL은 이러한 기본 기능 외에도 자동 커밋을 설정하거나 운영체제 관련 명령어의 실행, 출력 저장, 스크립트 기능 등을 제공한다. 특히 스크립트 기능은 여러 SQL 문장 및 PSM 프로그램, tbSQL 유틸리티의 명령어를 하나의 스크립트 파일로 생성할 수 있어 편리하다.

tbSQL 유틸리티는 Tibero의 유틸리티 중에서 가장 빈번히 사용되는 것 중의 하나이며, SQL 문장의 실행 이외에 다음과 같은 기능을 제공한다.

- 일반적인 SQL 문장 및 PSM 프로그램의 입력, 편집, 저장, 실행
- 트랜잭션의 설정 및 종료
- 스크립트를 통한 일괄 작업의 실행
- DBA에 의한 데이터베이스 관리
- 데이터베이스의 기동 및 종료
- 외부 유틸리티 및 프로그램의 실행
- tbSQL 유틸리티의 환경설정

2.1.2. T-Up

T-Up은 Tibero DB에서 제공하는 유틸리티로 이 유틸리티는 실제 마이그레이션을 수행 전 호환성 평가 기능과 데이터베이스 스키마 오브젝트 중 전체 또는 일부를 Tibero로 옮기는 마이그레이션 기능으로 구성되어 있다.

T-Up 유틸리티의 기능은 다음과 같다.

- 호환성 평가 기능

호환성 평가 기능은 소스 데이터베이스의 스키마 오브젝트와 SQL 구문 그리고 응용 프로그램에서 사용된 데이터베이스 관련 API에 대해 Tibero에서의 지원가능 여부 및 미지원 원인 분석을 수행한다 .

- 실제 마이그레이션을 수행하기 전에 소스 데이터베이스의 스키마 오브젝트, SQL 구문, 응용 프로그램의 API에 대한 호환성을 평가한다.
- 호환성 평가 결과를 HTML 형식의 보고서로 출력하여 제공한다.

- 마이그레이션 기능

마이그레이션은 소스 데이터베이스에 저장된 테이블, 인덱스, 뷰 등의 스키마 오브젝트와 PSM 프로그램 등을 Tibero 데이터베이스로 옮겨 이전의 데이터베이스와 같은 기능을 수행한다.

- 사용자가 원하는 데이터를 선택하여 Tibero로 마이그레이션한다.
- 테이블, 인덱스, 뷰, 동의어 등의 스키마 객체와 테이블에 정의된 각종 제약조건을 마이그레이션한다.
- 사용자 특권(privilege) 및 역할(role)을 마이그레이션한다.
- 마이그레이션 타겟 데이터베이스에 대한 정보를 제공한다.
- **[Option]** 버튼을 사용하여 다양한 방법으로 마이그레이션한다.
- **Progress** 화면을 통해서 마이그레이션의 진행사항을 파악할 수 있다.

참고

1. T-Up은 Java 언어로 구현되어 있으며, Java 6 이상에서 사용할 수 있다. 또한 실행하기 전에 접속하려는 DB의 JDBC 드라이버 파일의 경로를 실행 스크립트 내의 classpath 설정에 추가해야 한다.
 2. 본 장에 설명하는 T-Up의 기능은 Tibero 6 FS06까지의 내용을 포함한다.
-

2.1.3. tbExport

tbExport는 Tibero에서 제공하는 Export 유틸리티이다. 이 유틸리티를 통해 Tibero 데이터베이스에 저장된 스키마 객체의 전체 또는 일부를 추출해 고유 형식의 파일로 저장하므로 데이터베이스의 백업과 다른 머신 간의 데이터베이스를 전송할 때 유용하다.

tbExport 유틸리티에서 하나의 스키마 객체를 추출하면 그와 연관된 스키마 객체가 자동으로 함께 추출된다. 예를 들어 하나의 테이블을 추출하면 그 테이블에 대해 생성된 인덱스와 제약조건 등이 함께 추출된다. 필요에 따라서 연관된 일부 스키마 객체가 함께 추출되지 않도록 지정할 수 있다.

Export 모드에는 전체 데이터베이스 모드, 사용자 모드, 테이블 모드가 있다. DBA만이 사용할 수 있으며, DBA 권한을 줄 수 없을 경우 SELECT ANY DICTIONARY 권한을 부여하여 사용하는 것을 권장한다.

tbExport 유틸리티를 실행한 결과로 생성된 파일은 운영체제 파일이다. 따라서 Tibero 데이터베이스 파일과는 달리 일반 파일과 같은 작업을 실행할 수 있다. 예를 들어 파일을 FTP를 이용하여 전송하거나 CD-ROM 등에 저장하여 원격지의 Tibero 데이터베이스로 옮길 수도 있다.

Export가 실행되는 과정에서 발생하는 로그는 LOG 파라미터를 사용하여 지정한다.

다음은 tbExport 유틸리티를 실행한 결과로 생성되는 완료, 경고 및 에러 메시지에 대한 설명이다.

항목	설명
완료 메시지	Export가 성공적으로 완료된 후에 출력된다.
경고 메시지	Export가 완료되었으나 경고가 발생한 경우에 출력된다. 존재하지 않는 테이블에 대해 Export를 시도할 때와 같은 경우에 발생하며 이런 경우 tbExport 유틸리티는 경고 메시지를 출력한 후 해당 테이블을 건너뛰고 다음 객체에 대해 Export를 계속한다.
에러 메시지	Export가 실행되는 과정에서 에러가 발생하여 Export를 계속할 수 없는 경우에 출력된다. 시스템 메모리가 부족한 상황이나 Export에 필요한 뷰가 생성되지 않는 경우처럼 Export를 계속 실행할 수 없는 상황에 출력되며, 에러 메시지를 출력한 후 Export 세션을 종료한다.

tbExport 유틸리티의 특징은 다음과 같다.

- 논리적인 백업
Tibero의 내부 스키마 및 데이터를 SQL 문장으로 추출한다.
- 서로 다른 시점의 데이터
여러 개의 테이블을 Export할 때 추출한 각 테이블의 데이터는 동일 시점의 데이터가 아니라 Export 작업이 실행되는 시점의 순차적인 데이터이다.
- 테이블 정의를 저장
데이터의 존재 여부에 상관 없이 테이블 정의(테이블의 DDL 스크립트)를 저장한다.
- 테이블의 재구성

테이블 생성 후 수많은 DML 작업으로 인해 발생한 마이그레이션이 된 로우(migrated row)나 단편화(fragmentation)를 제거한다.

2.1.4. tbImport

tbImport는 Tibero에서 제공하는 Import 유틸리티이다. 이 유틸리티를 통해 외부 파일에 저장된 스키마 객체를 Tibero 데이터베이스에 다시 저장하므로, **tbExport** 유틸리티와 함께 데이터베이스의 백업과 다른 머신 간의 데이터베이스 전송 등을 할 때 유용하다. **tbImport** 유틸리티는 기능에서 **tbExport** 유틸리티와 대칭적이거나 유사한 것이 많다.

하나의 스키마 객체를 저장하면 그와 연관된 스키마 객체가 자동으로 함께 저장된다. 필요에 따라서 연관된 일부 스키마 객체가 저장되지 않도록 지정할 수 있다.

Import 모드에는 **tbExport**에서와 같이 전체 데이터베이스 모드, 사용자 모드, 테이블 모드가 있다. DBA만이 사용할 수 있으며, DBA 권한을 줄 수 없을 경우 **SELECT ANY DICTIONARY** 권한을 부여하여 사용하는 것을 권장한다.

Import가 실행되는 과정에서 발생하는 로그는 LOG 파라미터를 사용하여 지정한다.

다음은 **tbImport** 유틸리티를 실행한 결과로 생성되는 완료, 경고 및 에러 메시지에 대한 설명이다.

항목	설명
완료 메시지	Import가 성공적으로 완료된 후에 출력된다.
경고 메시지	Import가 완료되었으나 경고가 발생한 경우에 출력된다.
에러 메시지	Import 과정에서 에러가 발생하여 Import를 계속할 수 없는 경우에 출력된다.

2.1.5. tbLoader

tbLoader는 대량의 데이터를 한번에 Tibero 데이터베이스에 저장하기 위한 유틸리티이다. 이 유틸리티를 통해 SQL 문장을 일일이 작성하여 데이터베이스에 입력할 데이터를 입력하는 대신, 컬럼 데이터만 일반 텍스트 파일로 만들어서 한꺼번에 적재할 수 있다. 따라서 많은 데이터를 Tibero의 데이터베이스에 한 번에 저장할 때 유용하다.

tbLoader 유틸리티를 사용하면 사용자가 데이터 파일을 쉽게 작성할 수 있으며, 데이터를 적재하기 위한 시간을 단축할 수 있다.

2.2. 디렉터리 구조

Tibero가 설치되면 다음과 같은 디렉터리가 생성된다.

```
$TB_HOME
+- bin
|   |
|   +- update
|
+- client
|   |
|   +- bin
|   +- config
|   +- include
|   +- lib
|   |   |
|   |   +- jar
|   |   +- php
|   +- ssl
|   |   |
|   |   +- misc
|   +- epa
|       |
|       +- java
|           |
|           +- config
|           +- lib
|
+- config
|
+- database
|   +- $TB_SID
|       |
|       +- java
|
+- instance
|   |
|   +- $TB_SID
|       |
|       +- audit
|       +- dump
|       |   |
|       |   +- act
|       |   +- diag
|       |   +- tracedump
|       +- log
|       +- dlog
```

```

|      | +- ilog
|      | +- lsnr
|      | +- slog
|      | +- sqltrace
|      +- path
|
+- lib
|
+- license
| |
| +- oss_licenses
|
+- nls
| |
| +- zoneinfo
|
+- scripts
  |
  +- pkg

```

위의 디렉터리 구조에서 \$TB_SID라고 보이는 부분은 각각의 시스템 환경에 맞는 서버의 SID로 바뀌서 읽어야 한다.

Tibero에서 사용하는 기본 디렉터리는 다음과 같다.

bin

Tibero의 실행 파일과 서버 관리를 위한 유틸리티가 위치한 디렉터리이다. 이 디렉터리에 속한 파일 중에서 tbsvr과 tblastener는 Tibero를 구성하는 실행 파일이며, tbboot와 tbdnwn은 각각 Tibero를 기동하고 종료하는 역할을 담당한다. tbsvr과 tblastener 실행 파일은 반드시 tbboot 명령어를 이용하여 실행되어야 하며, 절대로 직접 실행해서는 안 된다.

client

다음은 하위 디렉터리에 대한 설명이다.

하위 디렉터리	설명
bin	Tibero의 클라이언트 실행 파일이 있는 디렉터리이다. 이 디렉터리에는 다음과 같은 유틸리티가 있다. - tbSQL : 기본적인 클라이언트 프로그램으로 사용자가 직접 SQL 질의를 하고 그 결과를 확인할 수 있다. - T-Up : 다른 데이터베이스에서 Tibero로의 호환성 평가와 마이그레이션을 지원한다. - tbExport : 논리적 백업이나 데이터베이스 간에 데이터 이동을 위해 데이터 베이스의 내용을 외부 파일로 저장한다.

하위 디렉터리	설명
	– tbImport : 외부 파일에 저장된 내용을 데이터베이스로 가져온다. – tbLoader : 대량의 데이터를 데이터베이스로 한꺼번에 읽어 들인다. – tbpc : C 언어로 작성된 프로그램 안에서 내장 SQL(Embedded SQL)을 사용하는 프로그램을 개발할 때 이를 C 프로그램으로 변환한다. 이렇게 변환된 프로그램을 C 컴파일러를 통해 컴파일할 수 있도록 도와주는 역할도 담당한다. 유틸리티에 대한 내용은 "Tibero 유틸리티 안내서"를 참고한다. 단, tbpc 유틸리티는 "Tibero tbESQL/C 안내서"를 참고한다.
config	Tibero의 클라이언트 프로그램을 실행하기 위한 설정 파일이 위치하는 디렉터리이다.
include	Tibero의 클라이언트 프로그램을 작성할 때 필요한 헤더 파일이 위치하는 디렉터리이다.
lib	Tibero의 클라이언트 프로그램을 작성할 때 필요한 라이브러리 파일이 위치하는 디렉터리이다. 자세한 내용은 "Tibero 애플리케이션 개발자 안내서"와 "TiberotbESQL/C 안내서"를 참고한다.
ssl	서버 보안을 위한 인증서와 개인 키를 저장하는 디렉터리이다.
epa	External Procedure와 관련된 설정 파일과 로그 파일이 있는 디렉터리이다. 이에 대한 자세한 내용은 "Tibero External Procedure 안내서"를 참고한다.

config

Tibero의 환경설정 파일이 위치하는 디렉터리이다. 이 위치에 존재하는 \$TB_SID.tip 파일이 Tibero의 환경설정을 결정한다.

database

다음은 하위 디렉터리에 대한 설명이다.

– \$TB_SID

Tibero의 데이터베이스 정보를 별도로 설정하지 않는 한 모든 데이터베이스 정보가 이 디렉터리와 그 하위 디렉터리에 저장된다. 이 디렉터리에는 데이터 자체에 대한 메타데이터(metadata)뿐만 아니라 다음과 같은 종류의 파일이 있다.

파일	설명
컨트롤 파일	다른 모든 파일의 위치를 담고 있는 파일이다.
데이터 파일	실제 데이터를 저장하고 있는 파일이다.
로그 파일	데이터 복구를 위해 데이터에 대한 모든 변경 사항을 저장하는 파일이다.

– \$TB_SID/java

JAVA_CLASS_PATH가 정의되지 않은 경우 Java EPA Class File이 저장되는 디렉터리이다.

instance

다음은 하위 디렉터리에 대한 설명이다.

– \$TB_SID/audit

데이터베이스 사용자가 시스템 특권 또는 스키마 객체 특권을 사용하는 것을 감시(AUDIT)한 내용을 기록한 파일이 저장되는 디렉터리이다.

– \$TB_SID/log

Tibero의 시스템 로그 파일(slog)과 DBMS 로그(dlog), Internal 로그(ilog), Listener 로그(lsnr), memlog 파일이 저장되는 디렉터리이다.

파일	설명
시스템 로그 파일(slog)	디버깅을 위한 파일이다. 서버가 하는 중요한 일이 기록되는 파일이며, 서버 성능이 저하되는 원인을 찾거나 Tibero 자체의 버그를 해결하는 데 사용될 수 있다.
DBMS 로그 파일(dlog)	시스템 로그 파일에 기록되는 정보보다 좀 더 중요한 정보가 기록되는 파일이며, 서버 기동 및 종료, DDL 문장의 수행 등이 기록되는 파일이다.
Internal 로그 파일(ilog)	스레드별로 설정된 이벤트에 대한 시스템 로그가 기록되는 파일이며, Internal 로그를 보려면 tbiv를 이용해야 한다.
Listener 로그 파일(lsnr)	Listener의 디버깅을 위한 파일이다. 리스너에서 일어난 중요한 일이 기록되는 파일이며, 리스너의 버그를 해결하는 데 사용될 수 있다.

시스템 로그 파일, DBMS 로그 파일, Internal 로그 파일, Listener 로그 파일은 데이터베이스를 사용할수록 계속 누적되어 저장된다. 또한 전체 디렉터리의 최대 크기를 지정할 수 있으며, Tibero는 그 지정된 크기를 넘어가지 않도록 오래된 파일을 삭제한다.

로그 파일을 설정하는 초기화 파라미터는 다음과 같다.

초기화 파라미터	설명
DBMS_LOG_FILE_SIZE	DBMS 로그 파일 하나의 최대 크기를 설정한다.
DBMS_LOG_TOTAL_SIZE_LIMIT	DBMS 로그 파일이 저장된 디렉터리의 최대 크기를 설정한다.
SLOG_FILE_SIZE	시스템 로그 파일 하나의 최대 크기를 설정한다.
SLOG_TOTAL_SIZE_LIMIT	시스템 로그 파일이 저장된 디렉터리의 최대 크기를 설정한다.
ILOG_FILE_SIZE	Internal 로그 파일 하나의 최대 크기를 설정한다.
ILOG_TOTAL_SIZE_LIMIT	Internal 로그 파일이 저장된 디렉터리의 최대 크기를 설정한다.
LSNR_LOG_FILE_SIZE	Listener 로그 파일 하나의 최대 크기를 설정한다.
LSNR_LOG_TOTAL_SIZE_LIMIT	Listener 로그 파일이 저장된 디렉터리의 최대 크기를 설정한다.

– \$TB_SID/dump

Tibero의 DDL 또는 실행 중 에러에 의해 발생하는 덤프 파일이 저장되는 디렉터리이다.

하위 디렉터리	설명
act	스레드 액티비티 모니터에 의한 정보가 남는 디렉터리이다.
diag	TAC의 diag 기능을 사용하는 경우 로그 및 덤프가 남는 디렉터리이다.
tracedump	SQL 수행 중 에러가 발생하는 경우 디버깅을 위해 sql, psm 정보를 수집해서 남긴다. 이외에도 DDL dump 명령을 통해 남긴 덤프가 남는 디렉터리이다.

– \$TB_SID/path

Tibero의 프로세스 간에 통신을 위한 소켓 파일이 있는 디렉터리이다. Tibero가 운영 중일 때 이 위치에 존재하는 파일을 읽거나 수정해서는 안 된다.

lib

Tibero 서버에서 Spatial과 관련된 함수를 사용하기 위한 라이브러리 파일이 있는 디렉터리이다.

license

Tibero의 라이선스 파일(license.xml)이 있는 디렉터리이다. XML 형식이므로 일반 텍스트 편집기로도 라이선스의 내용을 확인할 수 있다.

다음은 하위 디렉터리에 대한 설명이다.

하위 디렉터리	설명
oss_licenses	반드시 준수해야 하는 오픈소스 라이선스의 대한 정보를 확인할 수 있는 디렉터리이다.

nls

다음은 하위 디렉터리에 대한 설명이다.

하위 디렉터리	설명
zoneinfo	Tibero에서 사용하는 시간대 파일이 있는 디렉터리이다.

scripts

Tibero의 데이터베이스를 생성할 때 사용하는 각종 SQL 문장이 있는 디렉터리이다. 또한 Tibero의 현재 상태를 보여주는 각종 뷰의 정의도 이 디렉터리에 있다.

다음은 하위 디렉터리에 대한 설명이다.

하위 디렉터리	설명
pkg	Tibero에서 사용하는 패키지의 생성문이 저장되는 디렉터리이다.

2.3. 설치 환경

Tibero를 정상적으로 설치했다면 시스템에 다음과 같은 환경변수가 설정된다.

환경변수	설명
\$TB_HOME	Tibero가 설치된 홈 디렉터리이다. Tibero 서버, 클라이언트 라이브러리, 다양한 부가 기능을 수행하는 파일이 설치된다.
\$TB_SID	한 머신에서 Tibero의 인스턴스를 여러 개로 운영할 때 필요한 서비스 ID이다.
\$PATH	파일 시스템을 통해 특정 파일에 접근하기 위한 디렉터리 경로를 설정한다.

환경변수를 제대로 설정하지 않으면 Tibero를 사용할 수 없다. 따라서 환경변수를 확인하는 절차가 필요하다.

참고

본 안내서에서는 UNIX 셸 명령어를 실행할 때 **GNU Bash**(<http://www.gnu.org/software/bash/>) 문법을 따른다. 사용하는 셸의 종류에 따라 문법이 다를 수 있으며, 자세한 내용은 현재 사용 중인 운영체제의 안내서를 참고한다.

UNIX 셸 프롬프트에서 환경변수를 확인하는 방법은 다음과 같다.

• \$TB_HOME

```
$ echo $TB_HOME
/home/tibero/tibero7
```

이 디렉터리 안에는 Tibero 서버, 클라이언트 라이브러리, 부가 기능을 지원하는 파일이 있다.

• \$TB_SID

```
$ echo $TB_SID
Tb7
```

서비스 ID는 데이터베이스의 이름과 동일하게 설정할 것을 권장한다.

• \$PATH

```
$ echo $PATH
...:/home/tibero/tibero7/bin:/home/tibero/tibero7/client/bin:...
```

\$PATH는 다음의 디렉터리를 포함하고 있어야 한다.

디렉터리	설명
\$TB_HOME/bin	Tibero의 실행 파일과 서버 관리를 위한 유틸리티가 위치한 디렉터리이다.
\$TB_HOME/client/bin	Tibero의 클라이언트 실행 파일이 있는 디렉터리이다.

2.4. 사용자 환경설정 파일의 적용 및 확인

Tibero가 설치된 후에 사용자의 환경설정 파일에는 다음의 내용이 추가된다.

환경변수	설명
TB_HOME	Tibero가 설치된 디렉터리이다. 만약, Tibero를 설치하는 과정에서 설치 디렉터리를 입력하지 않았다면 기본적으로 {사용자의 홈 디렉터리}/Tibero/tibero7 가 TB_HOME으로 설정된다.
TB_SID	설치 과정 중에 입력한 SID가 여기에 해당한다. 설치 과정 중에 입력하지 않았다면 기본적으로 tibero 가 SID로 설정된다.
LD_LIBRARY_PATH	Tibero를 사용할 때 필요한 공유 라이브러리가 위치한 경로이다. 필요한 라이브러리는 모두 \$TB_HOME/lib와 \$TB_HOME/client/lib 안에 있고, OS별로 환경변수가 다르게 지정된다. – SunOS, Linux : LD_LIBRARY_PATH – AIX : LIBPATH
PATH	Tibero를 사용하기 위한 디렉터리 경로이다. 기본적으로 \$TB_HOME/bin과 \$TB_HOME/client/bin이 설정된다.

UNIX에서는 비정상적으로 프로세스를 종료할 경우 가상 메모리 정보를 디스크로 남긴다. (Core Dump)

이 과정에서 많은 컴퓨팅 자원을 요구하여 시스템 성능을 순간적으로 저하시키며, 이때 생성되는 Core 파일은 많은 디스크 용량을 필요로 한다. 따라서 Tibero에서는 이 기능을 사용하지를 권장하지 않는다.

Core Dump 기능을 사용하지 않으려면 사용자 계정의 환경설정 파일(.profile)에 다음의 명령어를 추가한다.

```
ulimit -c 0
```

위와 같이 사용자의 환경설정 파일이 생성되어도 실제로 적용된 것은 아니다. 환경설정 파일을 적용하기 위해서는 다음과 같은 과정을 거쳐야 한다.

1. 다음의 명령어를 입력하여 사용자 환경설정 파일을 적용한다. UNIX 계열(LINUX 포함)에서만 아래 명령어를 입력한다.

```
sh .profile
```

2. 환경설정 파일이 제대로 적용되었는지 확인하려면 다음의 명령어를 입력한다.

– Linux 계열의 경우

```
echo $TB_HOME
```

– Windows 계열의 경우

```
echo %TB_HOME%
```

제3장 Tibero 안내서 구성

본 장에서는 Tibero 안내서에 대한 내용을 간략히 소개한다.

3.1. 개요

제품에 대한 안내서가 있어도 제품에 대해 여러 가지 안내서가 존재하다 보니 제품을 처음 접하거나 제품에 익숙하지 않은 사용자는 원하는 안내서를 찾는데 문제를 겪을 수 있다. 본 장에서는 안내서 사용에 있어 사용자의 편의를 돕기 위해 Tibero 안내서를 어떻게 활용할 수 있는지 기술한다. 즉, Tibero 안내서에는 어떤 것들이 있고, 각 안내서에서 어떤 내용을 기술하고 있으며, 각 안내서가 다른 안내서와 어떻게 연관을 맺고 있는지에 대해 기술한다.

Tibero 안내서를 처음 접하는 사용자는 본 장을 주의 깊게 읽어볼 것을 권장한다. 비록 본 장이 실제 제품 사용법과 직접적인 관련이 있는 것은 아니지만, 안내서 구성을 전체적으로 이해하는 데 도움이 되도록 작성되었다.

3.2. 안내서 목록

Tibero 안내서 목록은 다음과 같다.

No	안내서	설명
1	Tibero Release Note	릴리즈된 버전의 주요 변경내역에 대해서 기술한다.
2	Tibero 소개	Tibero의 기본 개념과 주요 기능에 대해서 기술한다.
3	Tibero 관리자 안내서	Tibero 운영에 필요한 관리의 기본 사항에 대해서 기술한다.
4	Tibero tbCLI 안내서	Call Level Interface인 tbCLI의 개념과 구성요소, 프로그램 구조를 소개하고 tbCLI 프로그램을 작성하는 데 필요한 데이터 타입, 함수, 에러 메시지를 기술한다.
5	Tibero 애플리케이션 개발자 안내서	각종 애플리케이션 라이브러리를 이용하여 애플리케이션 프로그램을 개발하는 방법을 기술한다.
6	Tibero tbESQL/C 안내서	C 프로그래밍 언어를 사용해 데이터베이스 작업을 수행하는 각종 애플리케이션 프로그램을 작성하는 방법을 기술한다.
7	Tibero tbESQL/COBOL 안내서	COBOL 프로그래밍 언어를 사용해 데이터베이스 작업을 수행하는 각종 애플리케이션 프로그램을 작성하는 방법을 기술한다.

No	안내서	설명
8	Tibero External Procedure 안내서	External Procedure를 소개하고 이를 생성하고 사용하는 방법을 기술한다.
9	Tibero Hadoop Connector 안내서	Hadoop Connector의 사용 방법을 설명한다.
10	Tibero 설치 안내서	Tibero 설치 시 필요한 시스템 요구사항과 설치 및 제거 방법을 기술한다.
11	Tibero JDBC 개발자 안내서	Tibero에서 제공하는 JDBC 기능을 이용하여 애플리케이션 프로그램을 개발하는 방법을 기술한다.
12	Tibero tbPSM 참조 안내서	저장 프러시저 모듈인 tbPSM의 패키지를 소개하고, 이러한 패키지에 포함된 각 프러시저와 함수의 프로토타입, 파라미터, 예제 등을 기술한다.
13	Tibero tbPSM 안내서	저장 프러시저 모듈인 tbPSM의 개념과 문법, 구성요소를 소개하고, 프로그램을 작성하는 데 필요한 제어 구조, 복합 타입, 서브 프로그램, 패키지와 SQL 문장을 실행하고 에러를 처리하는 방법을 기술한다.
14	Tibero Spatial 참조 안내서	Tibero에서 Geometry 타입에 대한 설명과 Spatial 기능 관련 프러시저 함수 목록 및 사용 방법 등을 기술한다.
15	Tibero TAS 안내서	Tibero Active Cluster (TAS)를 사용해서 Tibero의 파일을 관리하고자 하는 관리자를 대상으로 기술한다.
16	Tibero TEXT 참조 안내서	Tibero의 제공하는 Text Index를 소개하고, Text Index를 생성하고 사용하는 방법을 기술한다.
17	Tibero TDP.NET 안내서	Tibero Data Provider for .NET 기능을 기술한다.
18	Tibero IMCS 안내서	Tibero에서 제공하는 In-Memory Column Store(이하 IMCS) 기능을 기술한다.
19	Tibero 유틸리티 안내서	데이터베이스와 관련된 작업을 수행하기 위해 필요한 유틸리티의 설치 및 환경설정, 사용 방법을 기술한다.
20	Tibero 에러 참조 안내서	Tibero를 사용하는 도중에 발생할 수 있는 각종 에러의 원인과 해결 방법을 기술한다.
21	Tibero 참조 안내서	Tibero의 동작과 사용에 필요한 초기화 파라미터와 데이터 사전, 정적 뷰, 동적 뷰를 기술한다.
22	Tibero SQL 참조 안내서	데이터베이스 작업을 수행하거나 애플리케이션 프로그램을 작성할 때 필요한 SQL 문장을 기술한다.
23	용어집	Tibero 안내서에서 사용하는 용어에 대해서 설명한다.