

Oracle Advanced Compression

오라클 백서
2007년 6월

주:

본 자료에서는 오라클의 일반적인 제품 방향을 소개하고 정보를 제공할 목적으로 작성되었으며 어떠한 계약에도 포함되지 않을 수 있습니다. 따라서 이 정보는 어떤 자료, 코드 또는 기능을 제공한다는 약속을 의미하는 것이 아니므로 구매 결정에 참고해서는 안됩니다. 여기 설명되어 있는 오라클 제품에 대한 모든 특징 또는 기능의 개발, 발표 및 시기 조정은 오라클의 단독 재량에 따라 이루어집니다.

Oracle Advanced Compression

들리는 글	3
ORACLE ADVANCED COMPRESSION	3
관계형 데이터 압축	4
혁신적인 알고리즘	4
테이블 압축의 이점	5
성능 오버헤드 최소화	5
비정형 데이터 압축	6
SecureFile 중복 제거	6
SecureFile 압축	7
백업 데이터 압축	7
Data Pump 압축	8
Recovery Manager 압축	8
네트워크 트래픽 압축	9
결론	9

Oracle Advanced Compression

들리는 글

기업이 업무를 효율적으로 수행하는 데 필요한 데이터의 양이 폭발적으로 증가하고 있습니다. 이와 같은 데이터 증가 동향의 원인으로 몇 가지 주요 요소를 들 수 있습니다. 사베인즈 옥슬리 법(Sarbanes-Oxley) 및 HIPPA와 같은 규제 환경의 최근 변화는 데이터 증가의 한 원인으로 꼽히고 있습니다. 즉, 이와 같은 규제들로 인해 기업들은 방대한 양의 정보를 장기간 보관해야 합니다. 또한 브로드밴드 기술의 발전으로 인터넷에서 풍부한 멀티 미디어 콘텐츠를 대량으로 배포할 수 있게 됨에 따라 전체 데이터 볼륨을 증가시키는 결과를 낳았습니다. 데이터의 폭발적 증가 추세에 기름을 부은 것은 바로 웹 2.0의 도래였습니다. 이로 인해 협업 애플리케이션은 방대한 양의 사용자 생성 콘텐츠를 만들어 냈습니다. 여러 측정 지표에 따르면 2~3년 마다 데이터 양이 거의 두 배씩 증가하고 있는 것으로 나타났습니다.

데이터의 폭발적 증가가 IT 관리자에게는 엄청난 관리 부담으로 다가왔습니다. 가장 먼저 떠오른 과제는 엄청나게 증가하는 스토리지 비용입니다. 최근 몇 년 사이 스토리지 MB당 비용이 현저하게 감소하고는 있지만 온라인으로 유지해야 하는 데이터의 양이 엄청나게 증가한 탓에 스토리지는 IT 예산의 가장 큰 부분을 차지하게 되었습니다. 또한 데이터의 양이 증가하더라도 애플리케이션의 확장성 및 성능은 지속적으로 비즈니스 요구 사항을 충족시켜야 합니다.

Oracle Database 11g는 Advanced Compression 옵션을 출시하여 고객들이 이런 문제를 해결할 수 있도록 했습니다. 오라클의 압축 기술 혁신을 통해 대량의 데이터 관리에 소요되는 비용과 인력을 줄일 수 있게 되었습니다. 한때는 신기술로 여겨졌던 테라바이트급 데이터베이스가 이제 엔터프라이즈 데이터 센터에서 일반적으로 사용되는 이때에 흥미로운 신기술이 출시되었습니다.

ORACLE ADVANCED COMPRESSION

Oracle Database 11g Advanced Compression 옵션은 포괄적 압축 기능 세트를 제공하여 자원 활용도를 최대화하고 비용을 절감하도록 돕습니다. IT 관리자는 이 솔루션을 사용하여 모든 유형의 데이터를 압축하여 전체 데이터베이스 스토리지 공간을 크게 줄일 수 있습니다. 이와 같은 데이터 유형에는 규칙적인 관계형 데이터(정형), 비정형 데이터(문서, 스프레드시트 등), 백업 데이터가 포함됩니다. Advanced Compression 옵션이 제공하는 혁신적인 압축 기술의 가장 대표적인 유형(有形)의 이점은

스토리지 비용 절감이지만, 그 이외에도 메모리 및 네트워크 대역폭을 포함한 모든 IT 인프라 구성 요소에 대해 자원 요구 사항 및 기술 비용을 절감할 수 있다는 이점도 있습니다.

관계형 데이터 압축

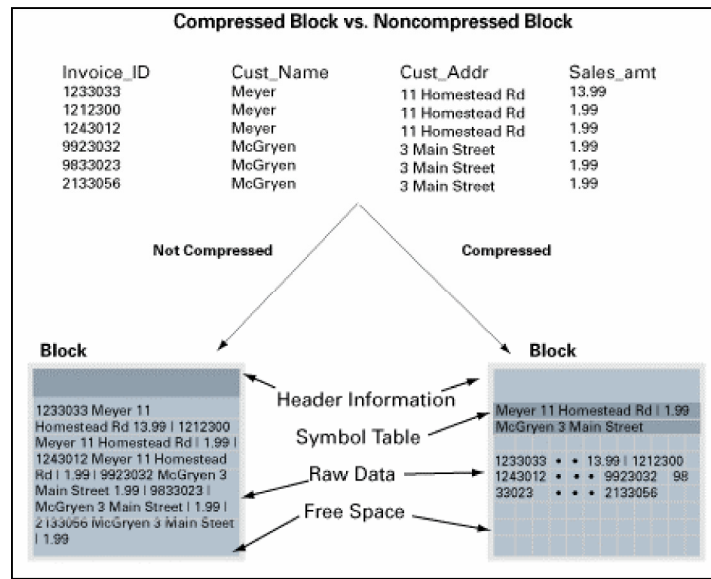
오라클은 데이터베이스 압축 기술을 도입한 선도적인 기업 중 하나입니다. Oracle Database 9i는 수년 전 테이블 압축(Table Compression)을 도입하여 직접 경로 로드, CTAS(CREATE TABLE AS SELECT) 등과 같은 대량 로드 작업 중에 데이터를 압축할 수 있도록 했습니다. 이와 같은 압축 형식은 일괄(batch) 처리를 사용하여 데이터베이스에 대부분의 데이터를 로드하는 데이터 웨어하우징 환경에 이상적이었습니다. Oracle Database 11g에서는 OLTP 테이블 압축이라는 새로운 기능을 도입하여 모든 유형의 데이터 처리 연산(INSERT, UPDATE, DELETE와 같은 전통적인 DML 포함) 중에 데이터를 압축할 수 있습니다. 또한 새로운 기능은 쓰기 연산의 오버헤드를 줄여 성능을 획기적으로 향상시킨다는 점에서 트랜잭션 환경 또는 OLTP 환경에도 적합합니다. 따라서 이와 같은 획기적인 혁신은 압축의 이점을 모든 애플리케이션 워크로드까지 확장합니다.

Oracle Database 9i에서 새롭게 선보인 테이블 압축 기능은 Enterprise Edition(EE)의 기본 기능이며 Oracle Database 11g에서도 계속 제공됩니다. 그러나 새로운 OLTP 테이블 압축 기능은 Oracle Advanced Compression 옵션의 일부로 포함되며 Enterprise Edition 이외에도 라이선스를 구입해야 합니다.

혁신적인 알고리즘

오라클은 관계형 데이터와 함께 작동하도록 설계된 고유의 압축 알고리즘을 사용합니다. 이 알고리즘은 데이터베이스 블록 내에서 여러 열에서 중복된 값을 제거함으로써 작용합니다. 압축된 블록에는 압축 메타데이터를 유지하는 심볼 테이블(symbol table)이라는 구조가 포함되어 있습니다. 블록을 압축하는 경우 우선 중복 값의 단일 복제를 심볼 테이블에 추가함으로써 중복 값을 제거합니다. 심볼 테이블의 해당 항목에서 각 중복 값은 요약 참조로 대체됩니다. 이러한 혁신적인 설계를 통해 압축 데이터가 데이터베이스 블록 내에 내장되어 있으며, 압축 데이터를 원래 상태로 변환하는데 사용하는 메타데이터도 블록 내에 포함되어 있습니다. 글로벌 데이터베이스 심볼 테이블을 유지하는 경쟁사의 압축 알고리즘과 비교하면, 오라클의 고유 접근법은 압축 데이터에 액세스할 때 추가로 I/O를 수행할 필요가 없기 때문에 상당한 성능 이점을 제공합니다.

그림1. 압축 블록 대 비압축 블록



테이블 압축의 이점

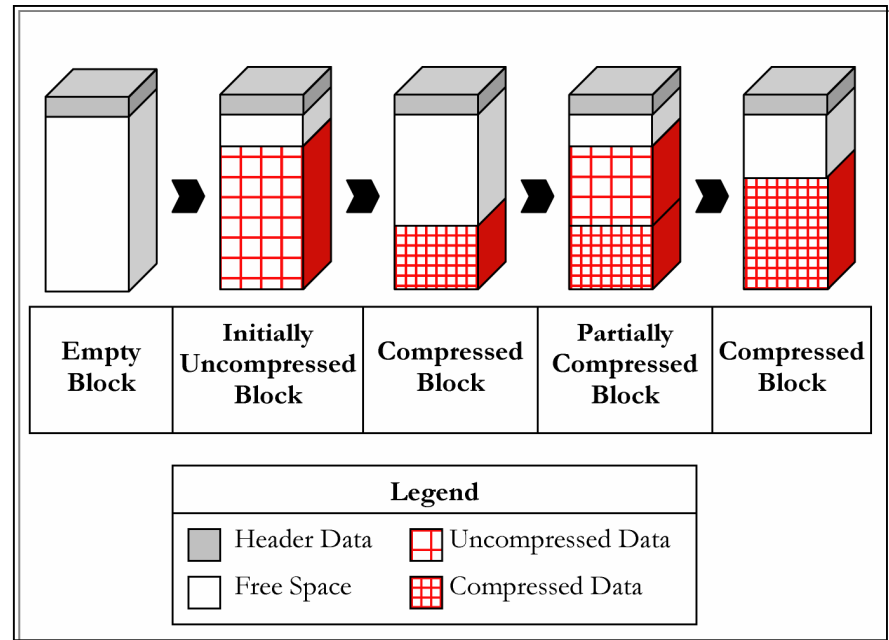
지정된 환경에서 달성되는 압축율은 압축되는 데이터의 속성에 따라 다르며 특히 데이터의 카디널리티(cardinality)에 좌우됩니다. 일반적으로 고객들은 테이블 압축 기능을 사용함으로써 저장 공간의 사용이 2-3배 정도 줄어들기를 원합니다. 즉 비압축 데이터 볼륨이 사용하는 공간이 압축 데이터가 사용하는 공간보다 2-3배 정도 많다는 것을 의미합니다. 압축은 단순한 디스크 스토리지 절약 이상의 이점을 제공합니다. 한 가지 중요한 이점 중 하나는 압축 블록을 해제하지 않고도 직접 압축된 블록을 읽을 수 있는 Oracle의 기능입니다. 또한 압축 데이터에 액세스하더라도 성능이 저하되지도 않습니다. 오라클은 보다 적은 수의 블록에 액세스하여 I/O 수가 줄어들기 때문에 실제 많은 경우, 성능이 향상됩니다. 또한 메모리 공간을 늘리지 않고도 캐시에 더 많은 데이터를 저장함으로써 버퍼 캐시를 더욱 효율적으로 사용할 수 있습니다.

성능 오버헤드 최소화

위에서 설명한 것처럼 테이블 압축 기능은 읽기 작업에 전혀 부정적인 영향을 미치지 않습니다. 그러나 데이터를 쓰는 동안 압축을 할 경우에는 추가 작업을 수행해야 합니다. 이는 쓰기 작업의 성능 오버헤드를 제거하기 위해 꼭 필요한 사항입니다. 그러나 오라클은 OLTP 테이블 압축의 오버헤드를 최소화하기 위해 상당한 노력을 기울였습니다. 오라클은 쓰기 작업이 발생할 때마다 데이터를 압축하기 보다는 일괄 모드에서 하나의 블록을 압축합니다. 블록 내의 데이터가 내부 제어 임계값에 도달하기 전까지는 새로 초기화된 블록은 압축되지 않고 남아있습니다. 트랜잭션 처리로 인해 블록 내의 데이터가 해당 임계값에 도달할 경우 블록의 모든 콘텐츠가 압축됩니다. 이어서 더 많은 데이터가 블록에 추가되어 임계값에 다시 도달하면 전체 블록이 최고 수준으로 재압축됩니다. Oracle이 블록에서 더 이상 압축으로 인해 이익을 얻을 수 없다고 판단할 때까지

이러한 프로세스가 반복됩니다. 블록 압축을 실행하는 트랜잭션만으로 압축 오버헤드가 최소화됩니다. 따라서 압축된 블록에 있는 대부분의 OLTP 트랜잭션은 비압축 블록 내에서도 완벽하게 동일한 성능을 갖게 됩니다.

그림2. 블록 압축 프로세스



비정형 데이터 압축

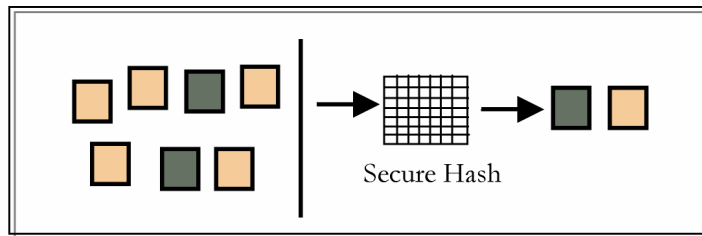
Oracle Database 11g의 새 기능인 SecureFile은 문서, 스프레드시트, XML 파일과 같은 비정형 콘텐츠 저장을 위해 양측 환경의 최고 이점을 모두 갖춘 아키텍처를 제공합니다. SecureFile은 Oracle Database의 이점을 그대로 유지하는 동시에 기존 파일 시스템의 성능에 필적하는 고성능을 파일 데이터에 제공합니다. SecureFile은 ANSI 표준 LOB의 슈퍼 세트로서 설계되었으며 구형 LOB 또는 BasicFile에서 손쉽게 마이그레이션할 수 있습니다. SecureFile을 사용하면 Oracle에서 모든 관계형 데이터 및 관련 파일 데이터를 관리할 수 있으며 단일 보안/감사 모델과 통합된 백업/복구 프로세스를 사용할 수 있고 모든 정보에서 원활하게 검색을 수행할 수 있습니다. Oracle Database 11g의 Advanced Compression 옵션은 SecureFile 데이터의 저장 공간을 대폭 줄여주는 기술이 포함되어 있습니다.

SecureFile 중복 제거

애플리케이션에서 파일의 완벽한 복제본을 저장하는 것은 매우 일반적입니다. 대표적인 예로 여러 사용자가 동일한 첨부 파일을 수신하는 e-메일 애플리케이션을 들 수 있습니다. SecureFile 중복 제거는 SecureFile 데이터의 중복된 복제본을 제거하는 지능형 기술입니다.

Oracle은 SecureFile 데이터에 대한 하나의 이미지를 저장한 후 중복 복제본을 해당 이미지에 대한 참조로 대체합니다. 10명의 사용자가 동일한 1MB 파일이 첨부된 e-메일을 수신하는 e-메일 애플리케이션을 생각해보십시오. SecureFile 중복 제거 기능이 없다면 이 시스템은 10명의 사용자 각각을 위해 하나의 파일 복제본을 저장하게 되며 총 10MB의 스토리지가 필요합니다. e-메일 애플리케이션이 SecureFile 중복 제거를 사용한다면 1MB 첨부 파일을 한번만 저장합니다. 이에 따라 스토리지 요구의 90%를 절약할 수 있습니다. 스토리지 절약 외에도 SecureFile 중복 제거는 또한 애플리케이션 성능을 향상시켜 줍니다. 특히 SecureFile 이미지에 대한 단일 참조만 기록하기 때문에 쓰기 및 복사 작업이 훨씬 효율적으로 수행됩니다. 중복된 SecureFile 데이터가 버퍼 캐시에 이미 존재하는 경우 읽기 작업도 향상됩니다.

그림3. SecureFile 중복 제거



SecureFile 압축

Oracle Database 11g의 Advanced Compression 옵션은 SecureFile 데이터의 크기를 제어하는 또 다른 메커니즘을 제공합니다. 앞서 설명한 중복 제거 외에도 SecureFile 압축은 업계 표준 압축 알고리즘을 활용하여 SecureFile 데이터의 스토리지 요구 사항을 최소화합니다. 문서 또는 XML 파일과 같은 일반 파일의 압축을 통해 크기를 2 - 3배 정도 줄일 수 있습니다. SecureFile 압축은 지능형 기술이 내장되어 있어 타사 툴로 압축된 문서와 같이 압축을 통해 효과를 거둘 수 없는 데이터는 하나의 SecureFile로서 데이터베이스에 삽입되기 전에 자동으로 압축을 우회합니다.

현재 두 가지 수준의 압축을 지원하고 있으며 보다 높은 수준으로 압축을 수행하는 경우, 더 많은 CPU 활용이 요구됩니다. SecureFile 압축의 일반적인 CPU 오버헤드는 3% - 5%입니다. 압축 데이터를 작은 조각의 데이터로 분할하기 때문에 애플리케이션은 압축된 SecureFile 데이터에서 계속해서 임의의 읽기 및 쓰기를 수행할 수 있습니다. 이는 데이터베이스에 파일을 삽입하기 전에 전체 파일을 압축하는 것에 비해 성능을 크게 향상시킵니다.

백업 데이터 압축

Oracle Advanced Compression은 데이터베이스 내부에 저장된 데이터를 압축하는 것 외에도 백업 데이터를 압축하는 기능도 제공합니다. Recovery Manager(RMAN) 및 Data Pump는 Oracle Database 내부에 저장된 데이터를 백업하는데 사용하는 가장 일반적인 툴입니다. RMAN은

데이터베이스 데이터의 블록별 백업을 수행하는데 이를 "물리적" 백업이라고도 합니다. 이 백업은 Secure Hash를 사용하여 데이터베이스, 테이블 스페이스, 블록 레벨 복구를 수행할 수 있습니다. 이에 반해 Data Pump는 하나 이상의 테이블에서 플랫폼 파일로 데이터를 보냄으로써 "논리적" 백업을 수행하는데 사용됩니다. Oracle Advanced Compression에는 이 두 가지 툴 모두에서 생성되는 백업 데이터를 압축하는 기능이 포함되어 있습니다.

Data Pump 압축

Data Pump 작업과 연관된 메타데이터를 압축하는 기능이 Oracle Database 10g Release 2에서 제공되었습니다. Oracle Database 11g에서는 이 기능이 확장되어 테이블 데이터를 익스포트하여 압축할 수 있습니다. Data Pump 압축은 인라인(inline) 작업이기 때문에 덤프 파일 크기를 줄일 수 있으며 이에 따라 디스크 공간이 상당히 절약됩니다. 운영 시스템 또는 파일 시스템 압축 유틸리티와는 달리 Data Pump 압축은 임포트 측에서도 완벽한 인라인 방식으로 수행되기 때문에 임포트하기 전에 덤프 파일의 압축을 해제할 필요가 없습니다. 압축된 덤프 파일 세트는 데이터베이스 관리자가 추가 단계를 수행할 필요 없이 임포트 작업 동안 자동으로 압축이 해제됩니다.

다음에 제시된 오라클 샘플 데이터베이스의 압축 예에서는 OE 및 SH 스키마를 익스포트하면서 동시에 모든 데이터 및 메타데이터를 압축합니다. 이로 인해 덤프 파일 크기가 74.67% 감소했습니다.

gzip(GNU zip) 유틸리티의 3가지 버전 및 1개의 UNIX 압축 유틸리티를 사용하여 6.0 MB 덤프 파일 세트를 압축합니다. 덤프 파일 크기는 Data Pump 압축과 비슷하게 감소합니다. 덤프 파일 크기의 감소는 데이터 유형 및 기타 요소에 따라 다릅니다.

전체 Data Pump 기능을 압축 파일에 사용할 수 있습니다. 정규 파일에서 사용되는 모든 명령은 압축 파일에서도 작동합니다. 사용자는 다음 옵션을 사용하여 덤프 파일 세트의 어느 부분을 압축할지 결정할 수 있습니다.

- ALL은 전체 익스포트 작업을 압축할 수 있습니다.
- DATA-ONLY는 덤프 파일에 기록한 모든 데이터를 압축합니다.
- METADATA-ONLY는 덤프 파일에 기록된 모든 메타데이터만 압축합니다. 이것이 기본값입니다.
- NONE은 전체 익스포트 작업에 대해 압축을 수행하지 않습니다.

Oracle Data Pump에 대한 자세한 정보는 사이트

(<http://www.oracle.com/technology/products/database/utilities/index.html>)

를 참조하십시오.

Recovery Manager 압축

기업 데이터베이스의 지속적인 증가는 데이터베이스 관리자에게 상당한 어려움을 안겨주고 있습니다. 데이터베이스 백업 및 백업 프로시저의 성능을 유지하기 위한 스토리지 요구 사항은 데이터베이스 크기에

직접적으로 영향을 받습니다. 오라클의 백업 및 복구 유틸리티인 Recovery Manager(RMAN)는 Oracle Database 10g에서 압축 기능을 도입했습니다. RMAN 압축은 백업에 필요한 스토리지를 대폭 줄여 줍니다. RMAN은 Oracle Database와 긴밀하게 통합되기 때문에, 디스크나 테이프에 기록하기 전에 백업 데이터를 압축하며 복구 전에 압축을 해제할 필요가 없기 때문에 스토리지 비용이 대폭 감소합니다. 하지만, 백업 성능은 광범위한 압축율에 영향을 받기 때문에 백업 실행 시간이 길어집니다.

Oracle Advanced Compression은 새로운 RMAN 압축 기능을 추가해 RMAN 성능을 향상시키면서도 백업의 스토리지 요구 사항을 대폭 줄일 수 있도록 했습니다. 산업 표준 ZLIB 압축 알고리즘을 기반으로 한 RMAN 압축 백업은 Oracle Database 10g의 압축 백업보다 최대 40% 빠릅니다. 오라클은 대폭적인 성능 향상을 달성하면서 압축율도 20% 정도 약간 감소시켰습니다. 빠른 RMAN 압축은 정규 비즈니스 시간에 증분식 백업을 수행하는데 적합한 솔루션입니다.

네트워크 트래픽 압축

Data Guard는 관리, 모니터링, 자동화 소프트웨어 인프라를 제공하여 하나 이상의 대기 데이터베이스를 작성, 유지보수, 모니터링함으로써 장애, 고장, 오류, 데이터 손상으로부터 기업의 데이터를 보호합니다. Data Guard는 redo 데이터(트랜잭션 복구에 필요한 정보)를 사용하여 1차 및 대기 데이터베이스의 동기화를 유지합니다. 1차 데이터베이스에서 트랜잭션이 발생하면 redo 데이터가 생성되어 로컬 redo 로그 파일에 기록됩니다. Data Guard Redo Transport Service를 사용하여 이 redo 데이터를 대기 사이트에 전송합니다.

네트워크 또는 대기 서버가 중단되면 redo 데이터가 대기 서버에 전송되지 못합니다. 중단 문제가 해결되면 오라클은 대기 데이터베이스 동기화에 필요한 모든 redo 데이터를 전송함으로써 redo gap를 자동으로 제거합니다. Oracle Advanced Compression은 redo gap resolution 도중 네트워크에서 전송할 때 redo 데이터를 압축하는 기능을 도입했습니다. 이를 통해 압축 네트워크 대역폭이 극대화되어 gap resolution 처리량을 증가시켰습니다. 압축을 사용하여 gap resolution를 최대 두 배 빠르게 수행할 수 있고, 대기 데이터베이스를 빠르게 동기화하며 고가용성을 달성할 수 있습니다.

Oracle Data Guard에 관한 자세한 정보는 사이트

(<http://www.oracle.com/technology/deploy/availability/index.html>)를 참조하십시오.

결론

데이터 양의 폭발적 증가로 인해 기업들은 심각한 문제를 겪고 있습니다. 기업은 수익에 영향을 미치지 않으면서 변화하는 비즈니스 환경에 빠르게 적응해야 합니다. IT 관리자들은 기존 인프라를 효율적으로 관리하여 비용을 통제하면서도 뛰어난 애플리케이션 성능을 지속적으로 제공해야 합니다.

Oracle Database 11g의 Advanced Compression 옵션은 강력한 압축 기능 세트를 제공하기 때문에 IT 관리자들은 이와 같은 복잡한 환경을 성공적으로 관리할 수 있습니다. Advanced Compression 옵션을 활용하여 기업들은 데이터 센터의 모든 구성 요소 전반에서 증가하는 데이터 요구 사항을 효율적으로 관리하고 비용을 최소화하면서도 최고 수준의 애플리케이션 성능을 지속적으로 제공할 수 있습니다.



Oracle Database 11g

Advanced Compression 옵션

2007년 6월

작성자: William Hodak

공동 저자: Carol Palmer, Timothy Chien, Sushil Kumar

Oracle Corporation

본사

500 Oracle Parkway

Redwood Shores, CA 94065

U.S.A.

문의처:

전화: +1.650.506.7000

팩스: +1.650.506.7200

oracle.com

Copyright © 2007, Oracle. All rights reserved.

본 문서는 정보의 목적으로만 사용되며 일체의 내용은 고지 없이 변경될 수 있습니다.
본 문서는 오류에 대해 책임지지 않으며 특정 목적에 대한 적격성 및 적합성과 관련된
묵시적 보증 및 계약 조건을 포함해서 명시적, 묵시적 기타 모든 보증 또는 계약
조건에 의해 구속 받지 않습니다. Oracle은 본 문서와 관련해 어떠한 법적 책임도
지지 않으며, 본 문서로 인해 직간접적인 어떠한 계약 구속력도 발생하지 않습니다.
본 문서는 Oracle의 사전 서면 승인 없이는 어떠한 형식이나 수단(전자적 또는
기계적) 또는 목적으로도 복제되거나 배포할 수 없습니다.

Oracle은 Oracle Corporation 및 그 계열사의 등록 상표입니다 기타 명칭은
해당 소유업체의 상표입니다.