
CUBRID 2008 R2.0

릴리스 노트

목차

1. CUBRID 2008 R2.0 소개	4
2. CUBRID 2008 R2.0 시작하기	5
라이선스 안내	5
오픈소스 라이선스	5
권장 설치 사양	5
지원 운영체제	5
최소 하드웨어 요구 사항	5
소프트웨어 요구 사항	5
호환성	5
이전 버전과의 호환성	6
데이터베이스 호환성	6
상호 운용성	6
설치	6
업그레이드 주의사항	6
운영 환경에 맞는 패키지 선택	7
RedHat 기반 Linux에서 설치	7
Windows에서 설치	8
JAVA 환경 설정	9
이전 버전에서 CUBRID 2008 R2.0으로 마이그레이션	9
마이그레이션 순서 요약	9
마이그레이션 시나리오	10
마이그레이션 도구를 이용한 마이그레이션	10
unload/load을 이용한 마이그레이션	11
3. 새로운 기능	12
CUBRID 2008 R2.0의 특징	12
64Bit 운영체제 지원	12
2GB의 제약 없는 블록 사용 가능	12
새로운 데이터 타입 지원	12
High Availability 기능 지원	12
온라인 파라미터 정보 출력	13
장기 실행(long-duration) 질의/트랜잭션 검출 기능	13
CUBRID MANAGER R2.0의 특징	13
Eclipse 스타일의 GUI 지원	13
다중 호스트 관리 기능	14
CUBRID Manager 사용자 권한 관리 기능 향상	14
실시간 모니터링 기능 지원	14
4. 변화된 기능	15
성능 개선	15
CCI Connection/Statement Pooling 기능 추가	15
인덱스 키 추가/삭제 속도 개선	15
복제 기능의 동시성 향상	15
PAGE_FLUSH_THREAD_WAKEUP_INTERVAL_IN_SECS 파라미터 추가	15

LOG_FILE_SWEEP_CLEAN 파라미터의 기본값을 TRUE로 설정	15
질의문 실행 변경	15
EVALUATE 문 실행 시 데이터 변환 규칙 변경	15
ALTER SERIAL 구문을 이용하여 시작 번호 수정	15
관리 기능 변경	16
CAS 프로세스 재시작 조건 변경	16
에러 로그 파일이 생성되는 기본 디렉터리 변경	16
cubrid restoredb 유틸리티의 backuptime 매크로의 의미 변경	16
SQL_LOG 파라미터 기능 확장	16
복제 기능 변경	16
5페이지를 초과하는 트랜잭션 로그 레코드 지원	16
복제 에이전트 프로세스가 네트워크 자원을 독점하는 현상 수정	17
복제 에이전트 프로세스가 재 시작하지 못하는 오류 수정	17
응용 프로그램 인터페이스 개선	17
JDBC 드라이버의 Rollback 함수가 blocking 모드로 작동하도록 변경	17
주의 사항	18
HA 구성에서 JAVA stored procedure를 사용하면 failback 시간이 지연될 수 있음	18
R2.0 데이터베이스의 내보내기(unloaddb)한 파일을 이용하여 R1.x에서 가져오기(loaddb)하지 말 것	18
복제 기능을 사용하는 경우, 하나의 UPDATE 질의로 여러 레코드를 수정하는 질의 사용 시 주의사항	18
버그 리포트 및 피드백	19

1. CUBRID 2008 R2.0 소개

CUBRID 2008 R2.0은 기존 버전에 비해 적은 비용으로 기업의 정보 인프라를 운영/관리할 수 있으며, 고객의 요구에 따라 아래의 주요 기능을 지원한다.

- 64Bit 메모리 공간 지원
- 대용량 데이터 처리를 위한 볼륨 크기 지원
- High Availability 기능 지원
- 가져오기(loaddb)/내보내기(unloadb) 성능 최대 248%, 레코드 수정/삭제 성능 최대 525% 향상

CUBRID 2008 R1.x은 2GB 이하의 프로세스 메모리와 디스크 볼륨만 사용할 수 있으므로 대용량 데이터를 효율적으로 처리하기 어려웠다. CUBRID 2008 R2.0은 TB급 프로세스 메모리와 32TB의 단일 볼륨을 사용하여, 대용량 데이터베이스를 효율적으로 구성하고 관리할 수 있다.

또한, CUBRID 2008 R2.0은 네트워크 컴퓨팅 환경에서 지속적인 서비스를 제공하기 위해서 High Availability 기능을 지원한다. 이로 인해 시스템 정기 점검이나 장애 발생과 같은 상황에서도 데이터베이스 서비스를 중단 없이 제공할 수 있다.

데이터베이스의 가져오기(loaddb)/내보내기(unloadb) 성능이 최대 248% 향상되었다. 또한 레코드를 수정하는 경우에는 성능이 최대 525%, 레코드를 삭제하는 경우에는 성능이 최대 189% 향상되었다. 이로 인해 다중 사용자가 사용하는 대용량 데이터베이스에서 TPS/QPS를 향상시킬 수 있고, 사용자들의 대기 시간을 줄일 수 있다.

그림 1은 지난 7개월 동안 주 고객과 사용자로부터 신고된 주요 버그 수와 해결된 버그 수의 추이를 나타낸다. 고객들의 요구를 기반으로 버그의 우선 순위를 결정하고, 이에 따라 수많은 버그를 수정하였다. 이로 인해 R1.4 출시 이후 신고된 주요 버그 수의 증가율이 감소하고 있음을 알 수 있다. CUBRID 2008 R2.0을 사용하면 고객의 데이터를 보다 안정적으로 관리할 수 있을 것이다.

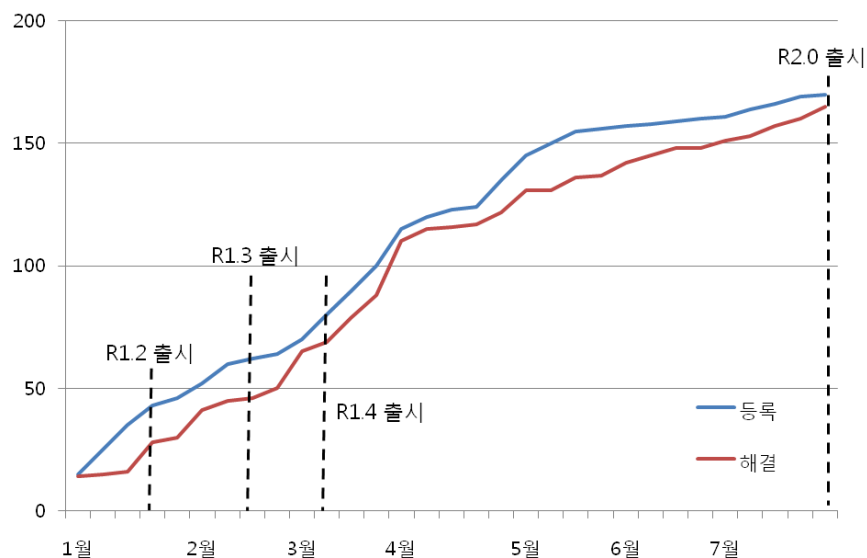


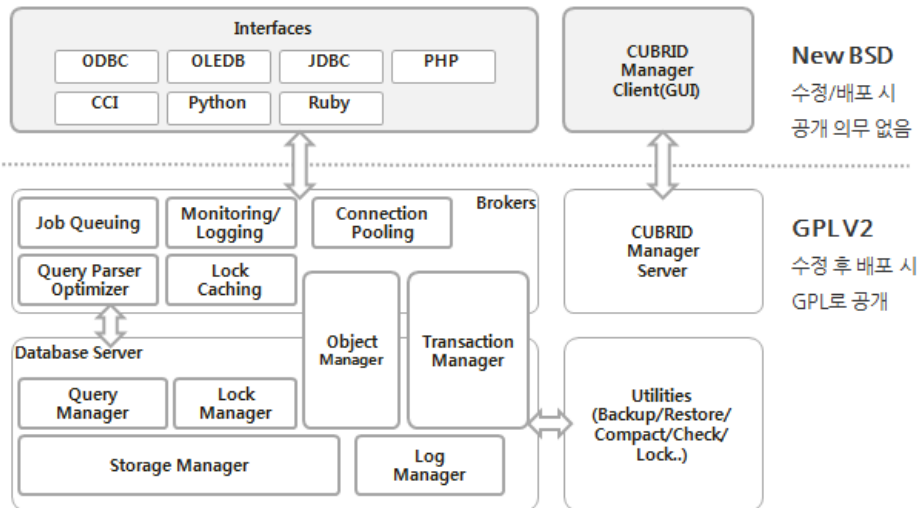
그림 1 이슈 트래커의 주요 버그 신고/해결 추이

2. CUBRID 2008 R2.0 시작하기

라이선스 안내

오픈소스 라이선스

CUBRID 2008은 서버 엔진과 Interface에 다른 오픈소스 라이선스가 적용된다. 아래 그림에 표시된 바와 같이, CUBRID 서버 엔진에는 GPL 라이선스를, CUBRID Manager 및 인터페이스(API)에는 BSD 라이선스를 적용하여, 다양한 응용 소프트웨어의 소스 공개 의무 없이 자유롭게 개발/배포/판매될 수 있도록 하였다.



권장 설치 사양

지원 운영체제

- Microsoft Windows: XP, 2003, Vista 32/64Bit 지원
- Linux Kernel 2.4 이상 (glibc 2.3.4 이상)

최소 하드웨어 요구 사항

- CPU: 400Mhz Pentium II 이상 권장
- 메모리: RAM 32MB 이상, 원활한 시스템 운영을 위해 128MB 이상 권장.
- HDD: 최소 설치를 위해 기본적으로 60MB~180MB 필요하지만, 사용자 데이터를 저장하기 위한 추가 공간 필요. 원활한 시스템 운영을 위해 200~300MB 이상을 권장

소프트웨어 요구 사항

CUBRID Manager를 실행하기 위해서 운영체제와 상관없이 JAVA 런타임 환경(JDK/JRE) 버전 1.5 이상을 설치해야 한다. Windows 플랫폼은 "Microsoft Visual C++ 2008 재배포 가능 패키지(x86)"를 설치해야 한다(자세한 사항은 Windows에서 설치 절의 2. 설치 항목을 참고한다).

호환성

이전 버전과의 호환성

CUBRID 2008 R2.0은 응용 소프트웨어와 하위 호환성을 유지하고 있다. 즉, 기존의 응용 소프트웨어를 수정할 필요 없이 그대로 CUBRID 2008 R2.0 에서 사용할 수 있다. 단, CUBRID C API나 CCI를 사용한 응용은 반드시 CUBRID 2008 R2.0 버전의 라이브러리를 링크해야 한다.

CUBRID Manager R1.x는 CUBRID 2008 R2.0에 연결할 수 없다. CUBRID Manager R2.0을 반드시 설치한다.

데이터베이스 호환성

1. CUBRID 2008 R1.x 버전과 CUBRID 2008 R2.0 버전

CUBRID 2008 R2.0 버전은 볼륨(디스크) 이미지가 바뀌었기 때문에 CUBRID 2008 R1.x 버전과 데이터베이스 볼륨이 호환되지 않는다. CUBRID 2008 R1.x 버전의 데이터베이스를 CUBRID 2008 R2.0 버전에서 사용하려면, 반드시 마이그레이션해야 한다. (마이그레이션 방법은 이전 버전에서 CUBRID 2008 R2.0으로 마이그레이션 장 참고한다.)

2. CUBRID 2008 R2.0 32Bit 버전과 64Bit 버전

CUBRID 2008 R2.0은 64Bit 버전이든, 32Bit 버전이든 같은 구조의 볼륨 이미지를 사용한다. 따라서 CUBRID 2008 R2.0으로 마이그레이션을 완료한 데이터베이스는 추후 32Bit 버전과 64Bit 버전의 CUBRID로 이동이 자유롭다.

상호 운용성

32Bit 버전 브로커는 32Bit 버전 서버와, 64Bit 버전 브로커는 64Bit 버전 서버와 운용 가능하다.

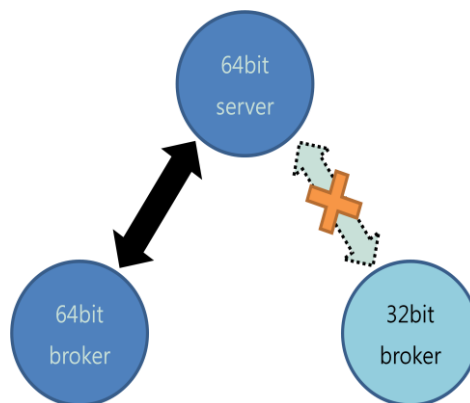


그림 2 32/64Bit 버전 상호운용성

기존과 마찬가지로 운영 체제가 다르더라도 상호 운용이 가능하다. 예를 들어, Linux용 64Bit 버전 서버와 Windows용 64Bit 버전 브로커는 운용 가능하다.

예외적으로, CUBRID 2008 R2.0 64Bit 버전과 CUBRID 2008 R2.0 32Bit 버전으로 마스터와 슬레이브간 복제를 구성하는 것은 가능하다.

설치

이전 버전 마이그레이션 사용자는 이전 버전에서 CUBRID 2008 R2.0으로 마이그레이션 장을 참고한다.

업그레이드 주의사항

● 환경 설정 파일 백업

실수로 기존의 환경 설정 파일을 덮어쓰지 않도록 업그레이드 전에 환경 설정 파일을 따로 보관한다 (자세한 내용은 이전 버전에서 CUBRID 2008 R2.0으로 마이그레이션 장의 3. 기존 CUBRID 디렉터리 보관 항목 참고한다).

● JAVA 런타임 환경 버전 확인

설치할 버전이 CUBRID 32Bit 버전인지 64Bit 버전인지 확인하고, 이에 따라 JAVA 런타임 환경 버전이 일치하는지 확인한다(자세한 내용은 JAVA 환경 설정 절 참고한다).

● 파라미터 확인

아래의 파라미터는 의미/기능 변경이 있으므로, 아래 파라미터를 사용하는 경우에는 내용을 확인한다.

- APPL_SERVER_MAX_SIZE: 기본값 변경 (자세한 내용은 CAS 프로세스 재시작 조건 변경 항목 참고한다)
- SQL_LOG: 기능 확장 (자세한 내용은 SQL_LOG 파라미터 기능 확장 항목 참고한다)

● 복제 재구성

CUBRID 2008 R1.x의 복제 기능을 사용하던 시스템은 복제 환경을 재구성해야 한다(자세한 내용은 7. 복제의 재구성

항목 참고한다).

운영 환경에 맞는 패키지 선택

CUBRID 2008 R1.x와 CUBRID 2008 R2.0에서 사용 가능한 설치 패키지는 아래와 같다. CUBRID 2008 R2.0에서는 제공되는 4가지 설치 패키지 중 1가지를 선택해서 설치해야 한다.

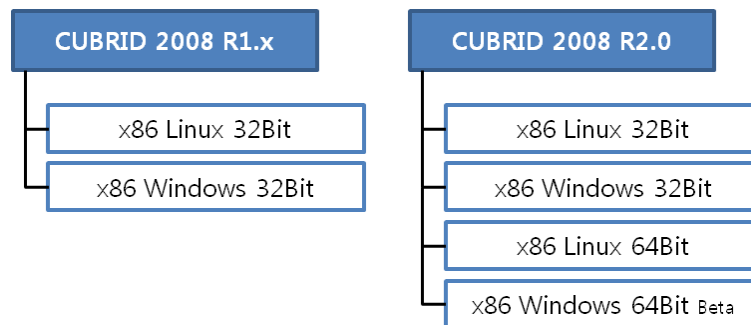


그림 3 R1.x와 R2.0의 사용 가능한 패키지

RedHat 기반 Linux에서 설치

1. 라이브러리 버전 확인

CUBRID는 glibc 2.3.4 버전 이상만 지원하므로, 아래 명령으로 glibc의 버전을 확인하고 필요시 업그레이드한다.

```
% rpm -q glibc // RPM 기반 배포판 (RedHat, Fedora, SuSe 등)
```

2. 운영체제 버전 확인

64Bit Linux를 사용한다면 약간의 차이는 있더라도 아래와 같이 64Bit임을 알 수 있는 x86_64 같은 문구를 발견할 수 있다. 32Bit Linux에는 CUBRID 32Bit 버전만, 64Bit Linux에는 CUBRID 32Bit/64Bit 버전 모두를 설치해서 사용할 수 있다.

```
% uname -a
Linux host_name 2.6.18-53.1.14.el5xen #1 SMP Wed Mar 5 12:08:17 EST 2008 x86_64 x86_64 x86_64
GNU/Linux
```

3. 설치

설치 패키지는 Linux RPM, tar.gz, 바이너리 셀로 제공되며, 기존 운영하던 CUBRID를 업그레이드할 고객은 기존 설치 디렉터리와 다른 디렉터리에 설치할 것을 권장한다. 기존 운영하던 디렉터리에 설치하면 기존 환경 설정 파일의 보관 여부를 결정해야 한다.

Windows에서 설치

1. Windows 버전 확인

“내 컴퓨터> 시스템 등록 정보” 창을 활성화하여 버전을 확인한다. XP의 경우, 32Bit Windows면 그림 4와 같이 Windows 이름이 그대로 출력되고, 64Bit Windows면 그림 5와 같이 x64 Edition이라는 문구가 추가로 출력된다. 32Bit Windows에는 CUBRID 32Bit 버전만, 64Bit Windows에는 CUBRID 32Bit/64Bit 버전 모두를 설치해서 사용할 수 있다.

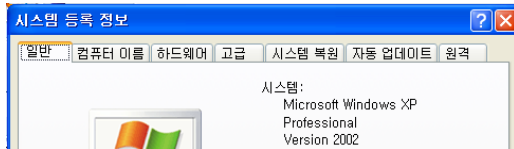


그림 4. 32Bit Windows



그림 5. 64Bit Windows

2. 설치

설치 Wizard를 사용하여 CUBRID 2008 R2.0을 설치하면 “Microsoft Visual C++ 2008 재배포 가능 패키지(x86)”가 설치되어 있는지 확인한다(그림 6). 재배포 가능 패키지가 설치되어 있지 않으면 설치를 중단하고, 관련 홈페이지(<http://www.microsoft.com/downloads/details.aspx?FamilyID=9b2da534-3e03-4391-8a4d-074b9f2bc1bf&DisplayLang=ko>)에서 내려 받은 후 다시 CUBRID를 설치한다. CUBRID 2008 R2.0을 전부 설치하려면, ALL을 선택하고, CUBRID Manager와 Interface만 설치하려면 CLIENT_ONLY를 선택한다(그림 7).

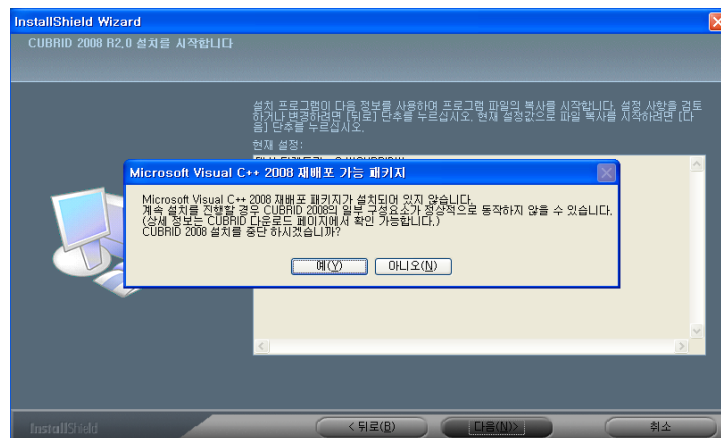
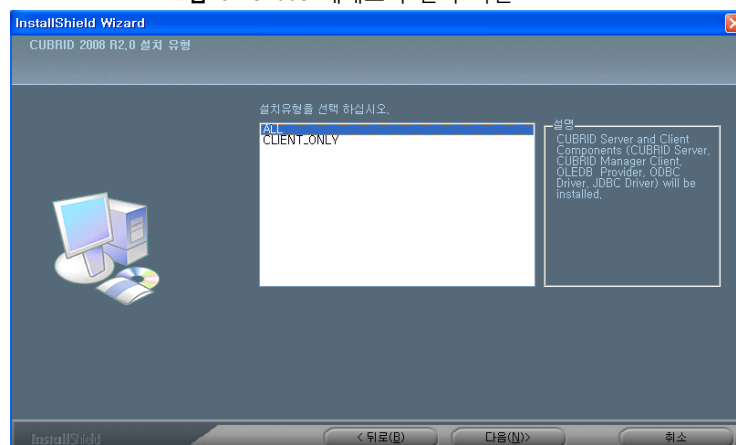


그림 6 VS2008 재배포팩 설치 확인



JAVA 환경 설정

CUBRID Manager와 JAVA stored procedure를 사용하기 위해서는 JAVA 런타임 환경 버전 1.5 이상이 설치되어 있어야 한다. 설치되어 있지 않으면 JAVA 런타임 환경을 JAVA 홈페이지에서 내려 받는다.

cubrid.conf에 java_stored_procedure 파라미터가 yes로 설정되어 있으면, CUBRID 2008 R2.0 64Bit 버전은 64Bit JAVA 런타임 환경이, CUBRID 2008 R2.0 32Bit 버전은 32Bit JAVA 런타임 환경이 필요하다. 예를 들어, 32Bit JAVA 런타임 환경이 설치된 컴퓨터에서 CUBRID 2008 R2.0 64Bit 버전을 실행하면 아래와 같은 에러가 발생한다.

```
% cubrid server start demodb
This may take a long time depending on the amount of recovery works to do.
WARNING: Java VM library is not found : /usr/java/jdk1.5.0_15/jre/lib/amd64/server/libjvm.so: cannot
open shared object file: No such file or directory.
Consequently, calling java stored procedure is not allowed
```

● JAVA 런타임 환경 버전 확인하는 방법

JAVA 런타임 환경의 버전은 아래 명령으로 확인할 수 있고, 64Bit JAVA 런타임 환경이 설치되어 있으면 아래 밑줄 친 64Bit와 같은 문구를 발견할 수 있다.

```
% java -version
Java(TM) SE Runtime Environment (build 1.6.0_05-b13)
Java HotSpot(TM) 64-Bit Server VM (build 10.0-b19, mixed mode)
```

● Linux 플랫폼에서 JAVA 환경 설정하기

JAVA_HOME 환경 변수와 LD_LIBRARY_PATH 환경 변수를 설정해야 한다.

- 64Bit JDK 1.6을 설치하고, Bash 셸에서 환경 변수를 설정한 예

```
JAVA_HOME=/usr/java/jdk1.6.0_05
LD_LIBRARY_PATH=$JAVA_HOME/jre/lib/amd64:$JAVA_HOME/jre/lib/amd64/server:$LD_LIBRARY_PATH

export JAVA_HOME
export LD_LIBRARY_PATH
```

- 32Bit JDK 1.5를 설치하고, Bash 셸에서 환경 변수를 설정한 예

```
JAVA_HOME=/usr/java/jdk1.5.0_10
LD_LIBRARY_PATH=$JAVA_HOME/jre/lib/i386:$JAVA_HOME/jre/lib/i386/client:$LD_LIBRARY_PATH

export JAVA_HOME
export LD_LIBRARY_PATH
```

● Windows 플랫폼에서 환경 설정하기

“내 컴퓨터>시스템 등록 정보>고급>환경 변수>시스템 변수 편집” 창을 실행하여, 아래 그림과 같이 JAVA_HOME 환경 변수를 설정하고 JAVA 실행 파일이 있는 디렉터리를 Path에 추가한다.

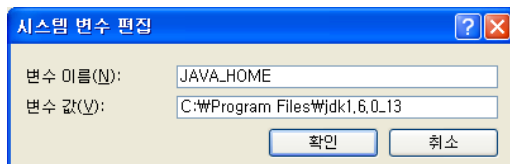


그림 8 JAVA_HOME 환경 변수 수정

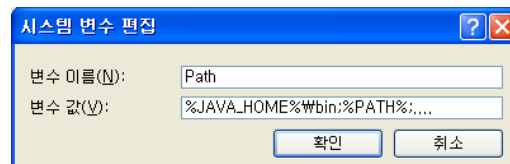


그림 9 JAVA 디렉터리를 Path에 추가

이전 버전에서 CUBRID 2008 R2.0으로 마이그레이션

마이그레이션 순서 요약

CUBRID 2008 R1.x(이하 R1.x)에서 CUBRID 2008 R2.0(이하 R2.0)으로 마이그레이션하는 과정을 요약하면 아래와 같다.

R1.x의 복제 기능을 사용하던 시스템은 복제 환경을 재구성해야 하며, 마이그레이션한 마스터 데이터베이스의 백업을 활용하여 슬레이브 데이터베이스를 구성한다.



그림 10. 마이그레이션 도구를 이용한 마이그레이션 절차

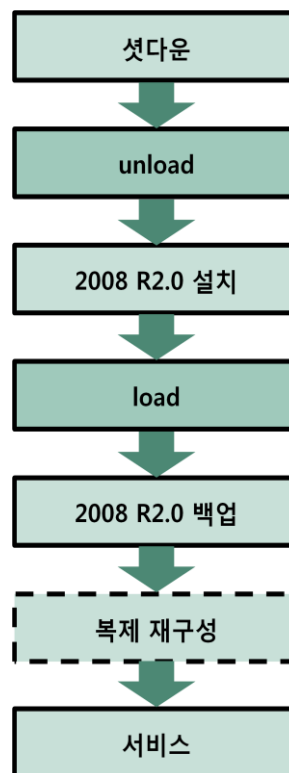


그림 11. unload/load를 이용한 마이그레이션 절차

마이그레이션 시나리오

- 64Bit 시스템에서 운영하는 R1.x를 R2.0 64Bit 버전으로 마이그레이션

운영체제를 포함한 64Bit 시스템을 그대로 사용하고 CUBRID만 업그레이드하는 경우이다. 그림 10의 순서로 마이그레이션한다. 64Bit Linux에서 CUBRID 2008 R1.x를 운영하는 대부분의 시스템이 이에 해당한다.

- 32Bit 시스템에서 운영하는 R1.x를 R2.0 64Bit 버전으로 마이그레이션

운영체제를 새로 설치하고 데이터베이스를 새로 구성하는 경우이다. 기존 버전의 CUBRID 데이터베이스 볼륨을 활용할 수 없기 때문에, 그림 11의 순서로 마이그레이션한다. 새로 구성되는 시스템이 이에 해당한다.

- 32Bit 시스템에서 운영하는 R1.x를 R2.0 32Bit 버전으로 마이그레이션

운영체제를 포함한 32Bit 시스템을 그대로 유지한 상태에서 CUBRID만 업그레이드하는 경우로, 32Bit Windows에서 CUBRID 2008 R1.x를 운영하는 대부분의 시스템이 이에 해당한다. 그림 10의 순서로 마이그레이션한다.

마이그레이션 도구를 이용한 마이그레이션

1. 데이터베이스 셧다운

R2.0으로 마이그레이션하기 전에, 아래 명령을 실행하여 데이터베이스 프로세스를 셧다운해야 한다.

```
% cubrid service stop
```

2. R1.x 백업

R2.0으로 운영 중 R1.x로 되돌리는 경우를 고려하여 R1.x의 데이터베이스를 백업한다. 또한, 마이그레이션 과정 중 시스템의 문제로 인해 데이터베이스가 손상되는 경우에 대비할 필요도 있다. 아래 명령을 실행하여 마이그레이션할 데이터베이스를 모두 백업한다.

```
% cubrid backupdb -S <database_name>
```

3. 기존 CUBRID 디렉터리 보관

기존 설치된 CUBRID 디렉터리를 보관한다. 마이그레이션 도구는 데이터베이스 볼륨만 변환해 주기 때문에 기존 설치를 삭제할 경우 최소한 \$CUBRID/conf 디렉터리의 cubrid.conf, cubrid_broker.conf, cm.conf 등의 환경 설정 파일과 databases.txt 파일을 보관해야 한다.

4. R2.0 설치

설치 방법은 RedHat 기반 Linux에서 설치 절차 Windows에서 설치 절을 참고한다.

5. 마이그레이션 도구 실행

CUBRID_DATABASES 환경 변수를 수정하거나, R1.x에서 사용한 databases.txt를 CUBRID 2008 R2.0의 설치 디렉터리에 복사하여 마이그레이션할 데이터베이스 정보를 가져온다. R2.0을 설치한 디렉터리 내의 %CUBRID/bin/migrate_r20 유틸리티를 아래와 같이 실행하여, 구 버전의 데이터베이스를 마이그레이션한다.

```
% export CUBRID_DATABASES=/old/databases
% cp /old/databases/databases.txt /new/databases/
% migrate_r20 <database name>
```

6. R2.0 백업

마이그레이션한 데이터베이스를 운영 중 복구할 수 있도록 백업해야 한다. R1.x 백업 파일에 R2.0 백업 파일을 덮어 쓰는 일이 없도록, R2.0의 데이터베이스를 백업하기 전에 기존의 R1.x 백업 파일을 따로 보관한다. 백업 파일은 <database_name>_bkxxxxx 파일이며, 여기서는 demodb 데이터베이스를 마이그레이션한다고 가정한다.

```
% mkdir R1x_backup
% mv demodb_bk* R1x_backup
% cubrid backupdb -S demodb
```

7. 복제의 재구성

기존 시스템이 복제 기능을 사용하고 있었다면, 슬레이브 데이터베이스는 마이그레이션하지 않고 마이그레이션한 마스터 데이터베이스의 백업 파일을 사용해서 복제 환경을 새로 구축한다. 복제를 구축하는 방법은 매뉴얼을 참고한다.

8. CUBRID 환경 설정 파일 수정 & 데이터베이스 시동

기존 버전의 CUBRID 환경 설정 파일을 가져와서 R2.0에 맞게 수정한다. 바뀐 내용은 파라미터 확인 항목과, SQL_LOG 파라미터 기능 확장 항목을 참고한다. 마지막으로 데이터베이스와 서비스를 시작한다.

```
% cubrid service start
% cubrid server start <database_name>
```

unload/load를 이용한 마이그레이션

마이그레이션 절차는 CUBRID 2008 R2.0 매뉴얼(관리자 안내서의 데이터베이스 마이그레이션 장)을 참고한다.

설치 방법은 RedHat 기반 Linux에서 설치 절차 Windows에서 설치 절을 참고한다.

기존 시스템이 복제 기능을 사용하고 있었다면, 슬레이브 데이터베이스는 마이그레이션하지 않고 마이그레이션한 마스터 데이터베이스의 백업 파일을 사용해서 복제 환경을 새로 구축한다. 복제를 구축하는 방법은 매뉴얼을 참고한다.

3. 새로운 기능

CUBRID 2008 R2.0의 특징

64Bit 운영체제 지원

64Bit 버전은 64Bit 메모리 주소 공간을 지원하므로 2GB 이상의 프로세스 메모리를 사용할 수 있다. 최대 프로세스 메모리 크기는 사용하는 운영체제의 제약에 따라 결정된다. 예를 들어, Linux 64Bit 환경에서는 64TB, Windows 64Bit 환경에서는 16TB까지 프로세스 메모리로 사용할 수 있다.

64Bit 버전은 32Bit 버전에 비해서 페이지 버퍼의 수를 지정하는 `data_buffer_pages` 파라미터의 값을 더 크게 지정할 수 있으며, 이를 통해 대용량 데이터베이스를 운영하는 경우, 성능 향상을 기대할 수 있다.

2GB의 제약 없는 볼륨 사용 가능

CUBRID 2008 R1.x에서 2GB미만의 볼륨 파일만 사용할 수 있었지만, CUBRID 2008 R2.0에서는 데이터 볼륨 및 백업 볼륨의 크기를 32TB까지 설정할 수 있다.

새로운 데이터 타입 지원

- **BIGINT**

64Bit 정수를 지원하기 위한 데이터 타입이다. BIGINT는 8바이트를 사용하는 정수형 데이터 타입으로, -9,223,372,036,854,775,808 부터 (+)9,223,372,036,854,775,807 까지 표현할 수 있다.

- **DATETIME**

연도, 월, 날짜, 시, 분, 초, 밀리 초로 이루어진 날짜/시간을 나타내는 데이터 타입이다.

기존의 TIMESTAMP 타입은 GMT 기준으로 1970-01-01 00:00:00 부터 2038-01-19 03:14:07 까지만 표현할 수 있는데 반해, DATETIME 타입은 밀리 초를 포함하여 0001-01-01 00:00:00.000 부터 9999-12-31 23:59:59.999 까지 표현할 수 있다.

관련 함수는 SYS_DATETIME, SYSDATETIME, CURRENT_DATETIME, TO_DATETIME이며, 자세한 사항은 CUBRID 2008 R2.0 매뉴얼(SQL설명서의 데이터 타입 변환 함수 장)을 참고한다.

High Availability 기능 지원

High Availability¹기능을 CUBRID에 적용한 것이 CUBRID HA 기능이며, 이를 통해 여러 서버 시스템의 데이터베이스를 항상 동기화된 상태로 유지하여 서비스를 제공할 수 있다.

CUBRID HA 기능은 서비스 제공 중인 마스터 데이터베이스 서버에 장애가 발생할 경우 대기 중인 슬레이브 데이터베이스 서버로 서비스를 자동 전환하고, 장애 복구 후에도 자동으로 동기화 과정을 수행하여 항상 동일한 데이터를 유지한다.

또한, CUBRID HA기능은 Shared-nothing 구조이므로, 서버의 마스터 데이터베이스에 장애가 발생하더라도 다른 데이터베이스를 사용하여 서비스를 제공할 수 있다. 또한 슬레이브 데이터베이스를 사용하여 서비스를 제공할 수 있으므로, 시스템 사용률도 높다는 장점이 있다.

¹ High Availability(HA)는 하드웨어, 소프트웨어, 네트워크 등에 장애가 발생하여도 지속적인 서비스를 제공해 주는 기능이다. 이 기능은 하루 24시간 1년 내내 서비스를 제공해야 하는 네트워킹 컴퓨팅 부분에서 필수적인 요소이다.

HA 기능은 CUBRID 2009 R1.x의 복제 기능과 비슷하지만, 차이는 아래 표와 같다.

지원 항목	복제	HA
복제 기반	기본키 기반 복제	기본키 기반 복제
동기화 방식	비동기식	동기식, 비동기식 지원
복제 구성용 데이터베이스 구축	별도 구축 필요	필요 없음
마스터 데이터베이스 절체	수동 절체	자동 절체
브로커 서버간 연결 절체	수동 절체	자동 절체
복제 그룹 설정 (일부 테이블 복제)	지원 O	지원 X
트리거 복제	지원 X	지원 O
데이터베이스 사용자 복제	지원 X	지원 O
데이터 동기화 정보 시스템 테이블	없음	DB_HA_APPLY_INFO 테이블

온라인 파라미터 정보 출력

CUBRID 데이터베이스 서버의 파라미터 정보는 cubrid.conf 파일에 저장된다. 이전 버전에서는 데이터베이스 서버가 구동 중일 때 cubrid.conf 파일을 수정하면 변경된 파라미터 정보를 확인할 수 없었다. 하지만, CUBRID 2008 R2.0에서는 아래의 명령을 사용하여 구동 중인 CUBRID 서버/클라이언트의 파라미터를 확인할 수 있다.

```
% cubrid paramdump -b <database_name>
```

장기 실행(long-duration) 질의/트랜잭션 검출 기능

서비스 품질에 영향을 주는 장기 실행 질의/트랜잭션을 검출할 수 있다. 아래의 두 파라미터를 cubrid_broker.conf 파일에 추가하여, 시스템의 특성에 맞게 장기 실행 질의/트랜잭션을 설정할 수 있다.

```
// long-duration query/transaction 시간 설정
LONG_QUERY_TIME = 60          // 단위: 초, 기본값: 60, (0 일 경우 long-duration query 판단하지 않음)
LONG_TRANSACTION_TIME = 60    // 단위: 초, 기본값: 60, (0 일 경우 long-duration transaction 판단하지 않음)
```

로그 파일을 확인할 필요 없이, 브로커 상태에서 장기 실행 질의/트랜잭션의 수와 에러 질의의 수를 확인할 수 있다.

```
% cubrid broker status -b
@ cubrid broker status
NAME          PID PORT AS  JQ      REQ TPS QPS LONG-T LONG-Q ERR-Q
=====
* query editor 32716 30000 5  0      0  0  0  0/60  0/60  0
* broker1     32726 33000 5  0      0  0  0  0/60  0/60  0
```

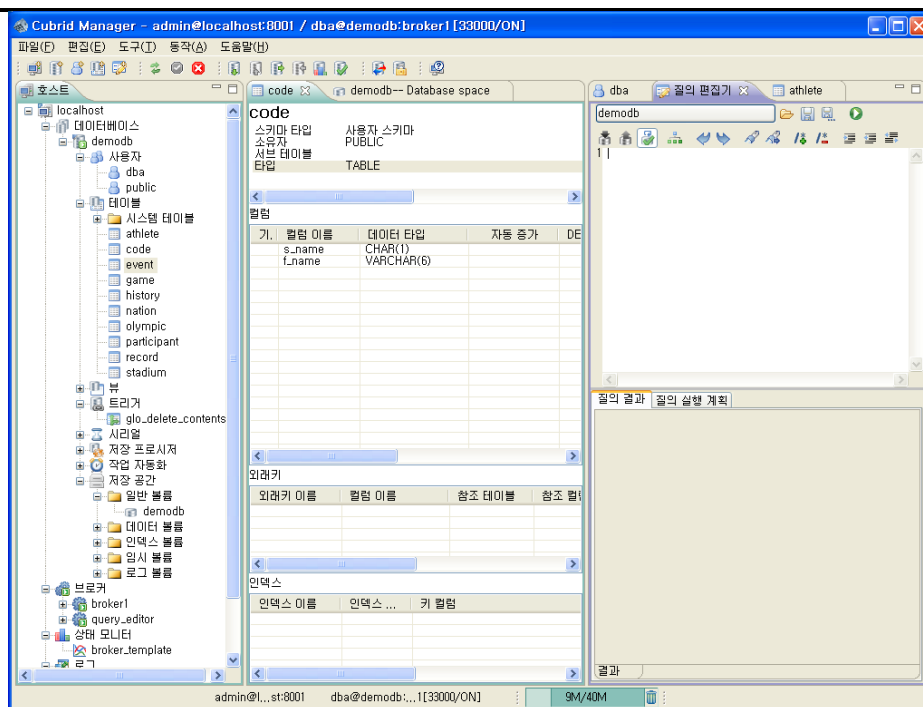
- * LONG-Q : LONG_QUERY_TIME 시간을 초과한 query의 수
- * LONG-T : LONG_TRANSACTION_TIME 시간을 초과한 트랜잭션 수
- * ERROR-Q : 에러가 발생한 query의 수

또한 SQL_LOG 파라미터를 설정하여 장기 실행 질의/트랜잭션에 대한 내역만 파일에 기록할 수 있다. (변화된 기능의 SQL_LOG 파라미터 기능 확장 항목 참고한다)

CUBRID Manager R2.0의 특징

Eclipse 스타일의 GUI 지원

CUBRID Manager R2.0은 CUBRID Manager R1.x의 기능을 계승하는 한편, 개발과 운영 편의성을 향상시키고자 Eclipse 스타일의 GUI를 지원한다. 또한, 데이터베이스 시스템 정보를 그래프로 출력하여 사용자가 쉽게 운영 정보 및 시스템 정보를 파악할 수 있도록 하였다.



다중 호스트 관리 기능

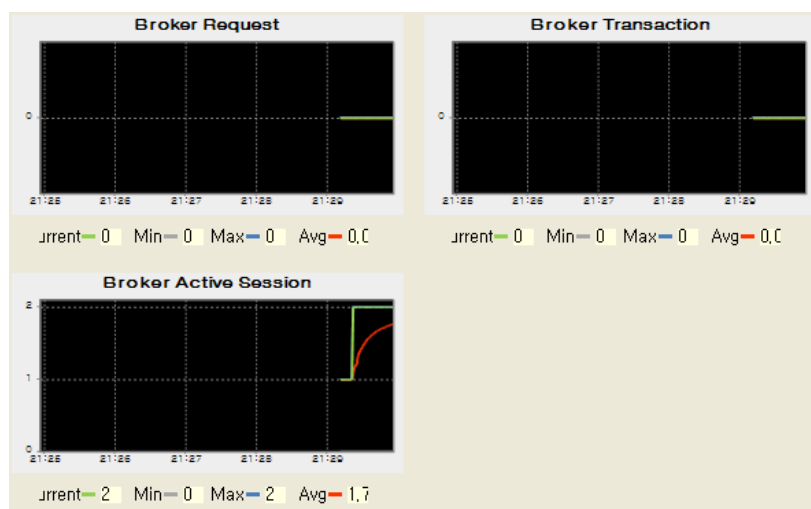
이전 버전의 CUBRID Manager에서는 1개의 실행 창에서 동시에 여러 호스트에 접속할 수 있는 기능을 지원하지 않았기 때문에 다중 호스트 관리에 불편함이 있었다. 한편, CUBRID Manager R2.0은 다중 호스트 관리 기능을 제공하므로, 1개의 실행 창에서 여러 호스트에 동시 접속하여 데이터베이스를 관리할 수 있다.

CUBRID Manager 사용자 권한 관리 기능 향상

동일한 CUBRID Manager에 여러 사용자가 접속하는 경우, CUBRID Manager 사용 권한을 사용자 별로 다르게 부여할 수 있다. 예를 들어, 데이터베이스를 생성할 수 있는 권한, 브로커를 관리/모니터링할 수 있는 권한, 상태 모니터를 추가/편집/삭제/실행할 수 있는 권한을 부여할 수 있다.

실시간 모니터링 기능 지원

데이터베이스와 브로커의 성능 지표 혹은 상태를 모니터링 항목으로 지정할 수 있으며, 여러 항목의 현재값/최소값/최대값/평균값을 출력할 수 있다.




```
ALTER SERIAL auto_increment_name START WITH 4; // 예
```

주의. R2.0에서 cubrid unloaddb 유틸리티를 사용하여 생성된 데이터에 ALTER SERIAL 구문이 포함되어 있는 경우에는 R1.x에서 cubrid loaddb 유틸리티를 사용하여 로딩 할 수 없다.

관리 기능 변경

CAS 프로세스 재시작 조건 변경

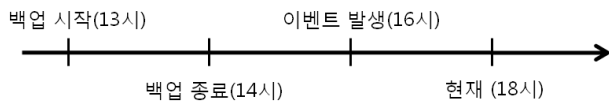
CAS 프로세스가 최초 크기의 두 배를 넘거나 아니면 cubrid_broker.conf에 APPL_SERVER_MAX_SIZE로 지정된 크기를 넘었을 경우 재시작하도록 변경하였다. APPL_SERVER_MAX_SIZE의 기본값은 32bit 버전의 경우는 40, 64bit 버전의 경우는 80이다.

에러 로그 파일이 생성되는 기본 디렉터리 변경

csql 인터프리터와 같은 대화형 유틸리티를 제외한 프로세스들의 기본 로그 디렉터리를 \$CUBRID/log로 변경하였다. 이로 인해, 데이터베이스 관리 내역을 알아내기 위해 여러 디렉터리에 흩어진 로그 파일을 찾아 헤맬 필요가 없다.

cubrid restoredb 유틸리티의 backuptime 매크로의 의미 변경

이전 버전과 달리 -d 옵션과 backuptime 매크로를 사용하여 cubrid restoredb 유틸리티를 실행하면 복구 시점을 백업이 종료한 시간으로 설정한다. 아래와 같이 이벤트가 발생하였고 "cubrid restoredb -d backuptime <database_name>"를 실행하면, backuptime 매크로가 기존 버전에서는 13시로 지정되었지만 R2.0에서는 14시로 지정된다.



SQL_LOG 파라미터 기능 확장

이전 버전에서는 SQL_LOG 파라미터를 ON, OFF로만 설정할 수 있었던 것을 아래의 표와 같이 설정된 값에 따라 필요한 로그만 출력하거나 모두 출력할 수 있도록 기능을 확장 하였다.

파라미터 값	동작 설명
OFF	모든 질의 로그를 출력하지 않음
ERROR	에러가 발생한 질의에 대한 로그만 기록
NOTICE	장기 실행 질의/트랜잭션에 대한 로그와 에러가 발생한 질의에 대한 로그 출력
TIMEOUT	장기 실행 질의/트랜잭션에 대한 로그만 출력
ON/ALL	발생하는 모든 로그를 출력

(장기 실행 질의/트랜잭션 검출은 새로운 기능의 장기 실행(long-duration) 질의/트랜잭션 검출 기능 항목 참고)

복제 기능 변경

5페이지를 초과하는 트랜잭션 로그 레코드 지원

이전 버전에서는 트랜잭션 로그 레코드의 크기가 5페이지를 넘는 경우에는 복제 기능이 제한 되었으나 페이지 수의 제한 없이 복제 기능이 정상적으로 수행되도록 복제 기능을 개선하였다.

복제 에이전트 프로세스가 네트워크 자원을 독점하는 현상 수정

특정 상황에서 복제 에이전트 프로세스(repl_agent)가 계속 같은 로그 페이지를 요청하여, repl_agent 프로세스가 네트워크 자원을 독점하는 현상을 수정하였다.

복제 에이전트 프로세스가 재 시작하지 못하는 오류 수정

복제 에이전트 프로세스(repl_agent) 종료 시 사용 중인 파일 기술자(file descriptor)를 완전히 정리하지 않아 발생할 수 있는 문제를 수정하였다.

응용 프로그램 인터페이스 개선

JDBC 드라이버의 Rollback 함수가 blocking 모드로 작동하도록 변경

서버의 트랜잭션 철회 작업이 모두 완료된 후 함수가 반환되도록 JDBC 드라이버의 Rollback 함수를 수정하였다. 기존 버전의 JDBC는 Rollback을 서버에 요청만하고 바로 반환 되었다.

주의 사항

HA 구성에서 JAVA stored procedure를 사용하면 failback 시간이 지연될 수 있음

HA 구성 환경에서 JAVA stored procedure를 사용하는 경우 Failover 시간이 지연되는 현상이 발생할 수 있다.

R2.0 데이터베이스의 내보내기(unloaddb)한 파일을 이용하여 R1.x에서 가져오기(loaddb)하는 경우 주의할 것

R2.0에서 SERIAL이 포함된 데이터베이스를 내보내기(unloaddb)하면 생성된 파일 내에 ALTER SERIAL 구문이 포함되어 있다. R1.x에서 이 파일을 사용해서 가져오기(loaddb)하면 에러가 발생하기 때문에 cubrid loaddb 유틸리티를 사용하여 R1.x의 데이터베이스로 마이그레이션 할 수 없다

복제 기능을 사용하는 경우, 하나의 UPDATE 질의로 여러 레코드를 수정하는 질의 사용 시 주의사항

복제 기능을 사용하는 경우, 하나의 UPDATE 질의로 여러 레코드를 수정하여 unique 제약을 위반하는 경우를 허용하지 않는다. 이 경우에는 데이터베이스 서버의 로그에 아래의 에러 메시지가 출력한다.

```
// 예
CREATE TABLE example (id INT);
INSERT INTO example VALUES (1);
INSERT INTO example VALUES (2);
UPDATE example SET id = id + 1 WHERE id > 0; // 오류 질의

// 영문 에러 메시지
Current version of replication does not allow changing multiple rows with a single UPDATE statement
which can violate the UNIQUE constraint.

// 한글 에러 메시지
현재 버전의 복제는 하나의 UPDATE 질의로 여러 레코드를 변경시키는 것이 UNIQUE 제약을 위반하는 경우 허용되지 않습니다.
```

버그 리포트 및 피드백

CUBRID는 사용자의 거침없는 버그 리포트와 솔직한 피드백을 기다리고 있으며, 아래 URL에 등록할 수 있다.

- 버그 리포트

CUBRID 오픈소스 프로젝트의 이슈 메뉴(<http://dev.naver.com/projects/cubrid/issue>)

- 피드백

CUBRID 오픈소스 프로젝트의 포럼 메뉴(<http://dev.naver.com/projects/cubrid/forum>)

CUBRID 공식 홈페이지(<http://www.cubrid.com>)