

논문 2011-48IE-4-10

## 클라우드 컴퓨팅 기반의 고가용성 복제시스템의 구현

## (Implement of High Available Replicate Systems Based on Cloud Computing)

박 성 원\*, 이 문 구\*\*, 이 남 용\*

(Sung-Won Park, Moon-Goo Lee, and Nam-yong Lee)

## 요 약

업무의 IT 의존도가 높아지면서 재해로부터 기업의 자산을 보호하는 것은 IT 운영 관리자들이 가장 고려해야 할 사항들 중 하나이고 데이터 및 정보는 기업 활동의 원천이기 때문에 데이터 보호는 업무 연속성 측면에서 IT 운영의 최대 우선순위가 되고 있다. 그러므로 본 논문은 이러한 재해로 인하여 중단된 정보기술 서비스를 재개하는 재해 복구를 위한 시스템을 클라우드 컴퓨팅 기반으로 구현하였다. 구현된 고가용성 복제 시스템은 복제성능 향상을 위해 다중 스레드 대상 데이터베이스 방법을 적용하였으며, 파일시스템의 실시간 동기화 기술로 네트워크 효율을 향상 시킬 수 있으며, 액티브 대 액티브 운영으로 백업 시스템 활용을 극대화함으로써, 근원지 데이터베이스 시스템의 부하를 분산시키는 효과를 갖게 한다. 이밖에도 복제 데이터의 정합성 검증 기능과 데이터 센터의 실제 운영에서 요구되는 모니터링 기법 등을 구현하였다. 성능평가에서도 다중 스레드 방식을 사용함으로써 단일 스레드 방식보다 약 3배 이상의 향상된 복제 기능 성능결과를 보여주었다.

## Abstract

As business management has a high level of dependence on Informational Technology (IT), protecting assets of a company from disaster is one of the most important thing that IT operating managers should consider. Because data or information is a major source of operation of the company, data security is the first priority as an aspect of continuity of business management. Therefore, this paper will realize disaster recovery system, which is suspended because of disaster, based on cloud computing system. Realized High Available Replicate System applied a method of multi thread target database to improve Replicate performance, and real time synchronize technology can improve efficiency of network. From Active to Active operation, it maximizes use of backup system, and it has a effect to disperse load of source database system. Also, High Available Replicate System realized consistency verification mechanism and monitoring technique. For Performance evaluation, High Available Replicate System used multi thread method, which shows more than threefold of replicate performance than single thread method.

**Keywords :** 복제 에이전트(Replicate Agent), 전송데이터 암호화 컬럼 복제(Transparent Data Encryption Colum Replicate), 정합성 검증(Consistency verification), 모니터링 기법(monitoring technique)

## I. 서 론

일반적으로 재해라고 하면 외부의 예방 및 통제가 불

가능한 사건으로 인해 정보 기술 서비스가 중단되거나, 정보시스템의 장애로부터의 예상 복구소요시간이 허용 가능한 범위를 초과하여, 정상적인 업무 수행에 지장을 초래하는 피해를 의미한다. 재해는 하드웨어 및 시스템 장애, 인간의 실수 및 오류, 컴퓨터 바이러스, 소프트웨어 결함, 자연 재해 등에 의해 발생하고 있다. 특히 업무의 IT 의존도가 높아지면서 재해로부터 기업의 자산을 보호하는 것은 IT 운영 관리자들이 가장 고려해야 할 사항들 중 하나이고 데이터 및 정보는 기업 활동의

\* 정회원, 숭실대학교 IT정책경영학과  
(Department of IT Policy Management, Soongsil University)

\*\* 정회원, 김포대학 IT학부 인터넷정보과  
(Div of IT, Department of Internet Information, Kimpo College)

접수일자: 2011년11월10일, 수정완료일: 2011년12월11일

원천이기 때문에 데이터 보호는 업무 연속성 측면에서 IT 운영의 최대 우선순위가 되고 있다. 그러므로 본 논문은 이러한 재해로 인하여 중단된 정보기술 서비스를 재개하는 재해 복구(Disaster Recovery)를 위한 시스템을 클라우드 컴퓨팅을 기반으로 구현하였으며, 이에 대한 이론적배경과 기반기술 등을 제안하였다.

본 논문의 구성은 다음과 같다. II장에서는 클라우드 컴퓨팅 기술에 대하여 기술하였으며, III장에서는 제안하는 복제 시스템의 구성과 기반기술에 대하여 기술하고, IV장에서는 제안하는 시스템의 주요기능에 대한 알고리즘과 소스코드 등을 기술하였으며, V장에서는 시스템의 구현과 성능평가결과를 제시하였다. 그리고 마지막으로 VI장에서는 결론과 차후 연구방향 등을 기술하였다.

## II. 클라우드 컴퓨팅 기술

클라우드 컴퓨팅 서비스란 서로 다른 물리적인 위치에 존재하는 IT 자원들을 가상화 기술로 통합하여 사용자의 요구에 적합한 환경을 제공하는 서비스를 총칭한다<sup>[1]</sup>. IT분야에서 분산된 서버나 스토리지, 네트워크와 같은 인프라 자원과 PC나 모바일 기기들의 디바이스 자원들을 가상화나 그리드컴퓨팅 기술 등을 이용해 각 자원간의 구분을 없애고 언제 어디서나 효율적인 서비스를 이용할 수 있도록 하는 클라우드 컴퓨팅 기술과 서비스 모델은 현재 핫이슈로 급부상하고 있다<sup>[2]</sup>. 이러한 클라우드 컴퓨팅은 규모의 경제를 실현할 수 있는 대용량 데이터 센터로부터 컴퓨팅을 필요에 따라 임대하여 이용할 수 있게 하므로써 보다 저렴하고 효과적인 컴퓨팅 방법을 제공하게되어 IT 기업과 학계 양쪽의 많은 관심과 주목을 받고 있다.

클라우드 컴퓨팅에서의 데이터 처리는 특정 데이터 센터들을 구성하는 수천, 수만 대의 노드 컴퓨터를 이용한 병렬 처리로 수행되며, 오늘날 다양한 클라우드 컴퓨팅 서비스들이 인터넷상에서 제공되고 있다<sup>[3]</sup>. 대용량 데이터의 처리와 관리에 관한 기술은 데이터 센터의 운영 기술과 더불어 클라우드 컴퓨팅의 주요 기술요소이며<sup>[4~5]</sup>, 고성능, 고가용성을 위한 데이터 복제의 적극적인 이용 등에 보다 적합한 기능이다. 이러한 클라우드 컴퓨팅을 통해 데이터가 연동되고 자원을 다양하게 활용하는 것에는 데이터 보호와 자원의 관리 정책, 기업 비밀 관리나 개인의 프라이버시 측면에서의 문제점

도 존재한다<sup>[6]</sup>.

그러나 본 논문에서는 클라우드 컴퓨팅 기반의 고가용성 복제 시스템에 강력한 보안기능으로 신뢰할 수 있는 컴퓨팅 기술을 기반으로 기밀성 보호와 데이터의 무결성 그리고 원격 확인을 위한 가상머신 보호와 관련한 클라우드 컴퓨팅의 보안정책 문제보다는 클라우드 컴퓨팅 기반의 고가용성 복제 시스템을 구현함<sup>[7]</sup>에 있어서 복제 데이터의 정합성 검증 기능과 데이터 센터의 실제 운영에서 요구되는 실시간 모니터링 기법 그리고 네트워크의 과부하를 최소화하기 위한 파일시스템의 실시간 동기화 기술 등을 지원할 수 있도록 구현하는데 그 초점을 맞추었다.

## III. 복제 시스템의 구성과 기반기술

### 1. 시스템의 구성

제안하는 복제시스템은 [그림 1]과 같이 근원지 시스템(Source System)의 복제 에이전트(Replicate Agent)와 대상(Target) 시스템의 복제 에이전트(Replicate Agent)로 각각 구성된다. 근원지 시스템의 복제 에이전트(Replicate Agent)는 파일 캡처 관리(File Capture Management)를 시행하는데 이는 파일시스템의 데이터에 대한 변경내역을 운영시스템에서 대상 시스템으로 이동하여 실시간 동기화할 수 있도록 설계하였다.

근원지 시스템 복제 에이전트(Replicate Agent)의 DBMS 캡처 관리자(DBMS Capture Manager)는 운영 데이터베이스의 변경내역을 추출하여 운영시스템에서 대상시스템으로 이동하여 실시간 동기화할 수 있도록 설계하였다.

파일시스템의 동기화를 위한 파일 캡처 관리(File Capture Manager)는 운영시스템에서 데몬 형태로 구동하며, 동기화의 대상이 되는 파일시스템의 변경내역

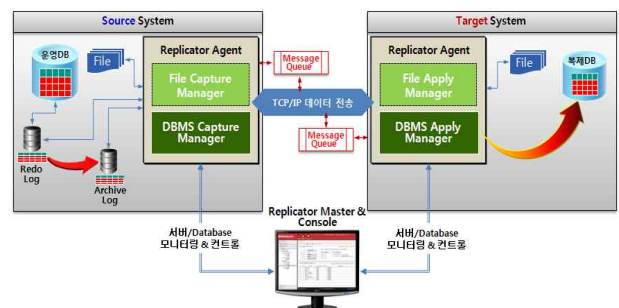


그림 1. 시스템 구성도  
Fig. 1. Configuration of System.

을 주기적으로 감시하여 변경된 파일을 대상시스템의 파일 공급 관리자(File Apply Manager)로 전송하여 반영하는 방식으로 동기화를 수행한다. 이때 변경된 파일 모두를 전송하지 않고, 블록단위의 체크섬(checksum)을 비교하여 변경된 블록만을 추출하여 압축전송하게 함으로써, 네트워크 부하를 최소화 할 수 있도록 설계되었다.

데이터베이스 동기화를 위한 DBMS 캡처 관리자(DBMS Capture Manager)는 운영시스템에서 대문 형태로 구동하며, 운영 데이터베이스의 변경내역을 오라클 다시쓰기 기록(Oracle redo log)로부터 실시간으로 추출하여 대상시스템으로 전송하고, 대상시스템의 DBMS 공급 관리자(DBMS Apply Manager)는 전송된 데이터 변경내역을 최적화된 SQL 구문으로 복제 데이터베이스에 반영하도록 하여 고성능의 실시간 데이터베이스 동기화가 수행될 수 있도록 설계되었다.

복제 마스터와 콘솔(Replicate Master & Console)은 동기화를 위한 모든 정책 설정 정보의 관리와 동기화 상태의 모니터링을 수행할 수 있는 관리 기능 및 모니터링 그래픽 사용자 인터페이스 기능을 수행한다.

2. 복제시스템의 기반 기술

가. 복제성능 향상을 위한 내부처리 구조

일반적인 적용방식은 대상 데이터베이스에 대하여 단일 스레드 데이터베이스 방식을 적용한다.

제안하는 시스템은 시스템의 복제 성능을 향상하기 위하여 대상 데이터베이스에 다중 스레드 방식을 적용하였다.

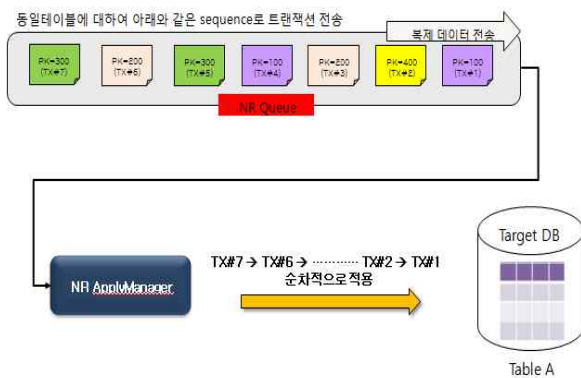


그림 2. 단일 스레드 방식  
Fig. 2. Single Thread Method.

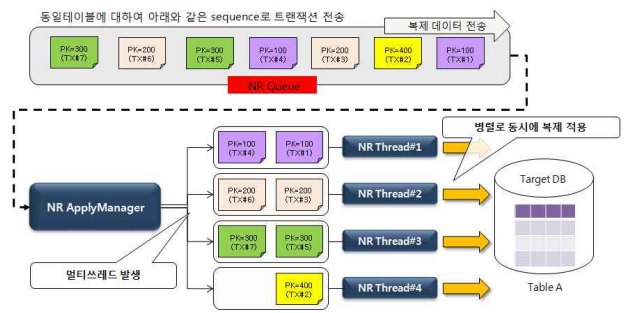


그림 3. 다중 스레드 적용방식  
Fig. 3. Multi Thread Method.

나. 파일시스템의 실시간 동기화 기술

파일시스템(파일 및 디렉토리)의 실시간 동기화를 위해서 근원지 시스템의 변경(생성/수정)된 파일만 동기화를 수행하며, 변경된 파일이 대상 시스템에 존재할 경우 파일 내에서 변경된 블록만 추출하여 전송함으로써 네트워크 전송량을 최소화하도록 하며, 파일의 user, group, 접근권한 등 파일 속성(attribute)과 부호화 링크(symbolic link) 파일 정보의 동기화를 수행하며 동기화 시행 시 압축 옵션을 제공함으로써 네트워크 효율을 향상시킬 수 있다<sup>[8]</sup>.

만약에 근원지 시스템에 장애가 발생할 경우는 대상 시스템으로 동기화된 파일 시스템 또는 특정 파일만 선택하여 역방향 동기화 기능을 제공하도록 한다.

또한 근원지 시스템과 대상 시스템의 파일 시스템 변경 사항에 대한 양방향 동기화 기능 제공으로 원격 클러스터 파일 시스템(Clustering File System)처리가 가능하도록 하였다.

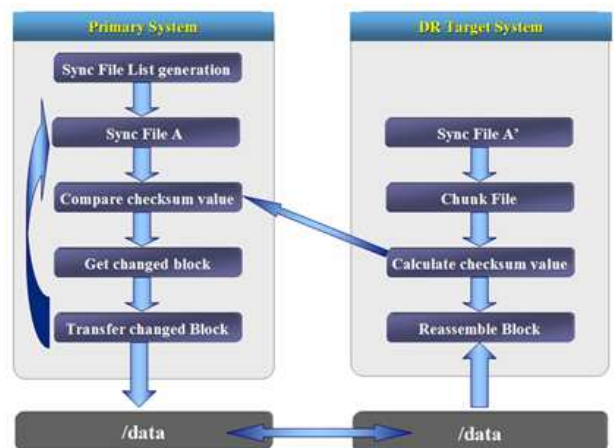


그림 4. 파일시스템의 실시간 동기화  
Fig. 4. Real-time Synchronization of File Systems.

다. 액티브-액티브 운영 기술

제안하는 복제 시스템은 액티브 대 액티브(Active-Active) 운영으로 백업시스템 활용을 극대화하였으며, 대상 데이터베이스 시스템은 실시간 복제 중에서도 활용 가능하여 대량의 질의 및 자료 추출 등의 부하분산 용으로 활용할 수 있다. 이는 근원지 데이터베이스 시스템의 부하를 분산시키는 효과를 갖게 한다.

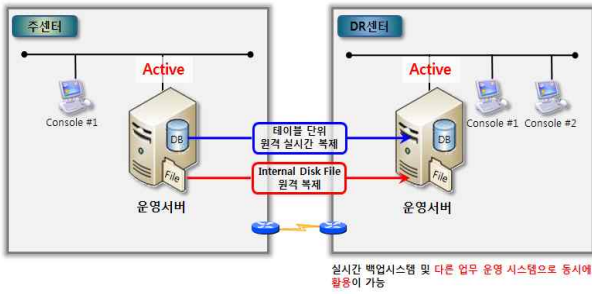


그림 5. 액티브-액티브 운영  
Fig. 5. Operation of Active to Active.

라. 메시지 큐 방식 적용 기술

제안하는 복제 시스템은 메시지 큐 방식 처리로 시스템, 데이터베이스, 네트워크 장애로부터 안전하도록 설계하였으며, 이는 영속적인 모드(Persistent mode)로 운영 되는데 이는 저장 후 전송(Store and Forward)방식으로 메시지를 디스크 캐시에 저장한 후 전송하게 된다. 또한 시스템과 에이전트의 종료(shutdown) 이후 재기동에도 데이터의 연속성이 보장된다.

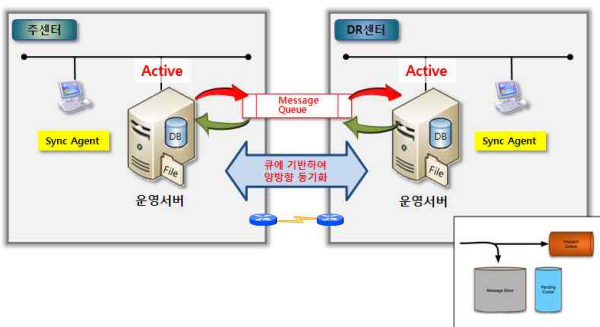


그림 6. 메시지 큐 방식  
Fig. 6. Message-Queue Method.

마. 전송데이터 암호화 컬럼 복제 기술

제안하는 복제시스템은 [그림 7]과 같이 전송 데이터 암호화(Transparent Data Encryption) 컬럼에 대한 복제 기능을 제공하며, 이러한 기능은 전송데이터 암호화

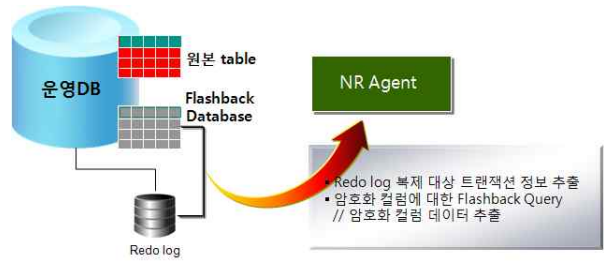


그림 7. 전송데이터 암호화 컬럼 복제 기능  
Fig. 7. A Function of Transparent Data Encryption Column Replicate.

가 적용된 컬럼의 데이터만 추출하여 최적화된 SQL 구문의 생성 및 복제를 수행한다<sup>[9]</sup>.

이처럼 전송데이터 암호화(Transparent Data Encryption) 컬럼에 대한 복제 기능은 사용자의 실수 또는 어플리케이션 오류로 인해 데이터베이스 테이블 데이터가 손상 되었을 때, 과거 특정시점의 데이터에 대하여 질의를 통해 확인하고 복구할 수 있는 기능이며, 데이터베이스 백업 본을 이용하여 복구 작업을 수행하지 않고도 데이터베이스의 과거 시점의 데이터를 확인할 수 있으므로 신속한 시점에 복구 작업을 수행할 수 있다<sup>[10]</sup>.

IV. 복제 시스템의 주요기능 알고리즘과 소스코드

1. 복제상태의 실시간 모니터링 기능

실시간 모니터링을 위해서는 별도의 데몬을 실행해야 하는데 주기적으로 각 서버의 상태나 구동 프로세스 상태를 감시하고, 이렇게 수집된 정보를 운영 데이터베이스에 갱신한다.

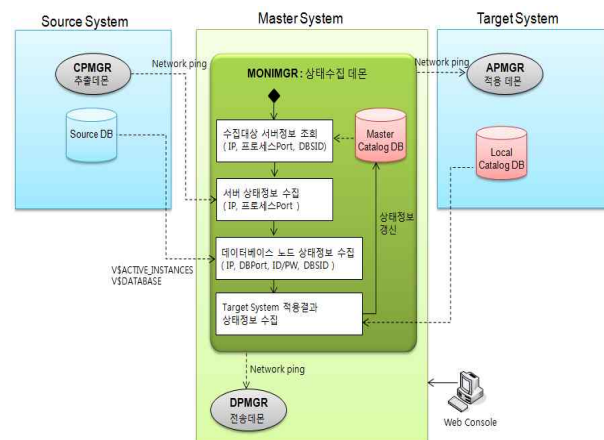


그림 8. 실시간 모니터링 기능  
Fig. 8. A Function of Real-time monitoring.

```

NRMoniInfoObj moniInfo = vt_grpInfo.get(i);
NRMoniLogger.out.info("-----");
NRMoniLogger.out.info(i + ". MONITORING GROUP CODE : "+
moniInfo.getGrp_code());
NRMoniLogger.out.info("-----");
// (0) 임계값 가지고 오기
Hashtable<String, NRMoniLimitObj> ht_limit = new Hashtable<String,
NRMoniLimitObj>();
Vector<Hashtable<String, String>> vt_limit =
NRMoniDBManager.getLimitValue(moniInfo.getGrp_code());
for (int k = 0; k < vt_limit.size(); k++) {
NRMoniLimitObj lmtObj = new NRMoniLimitObj();
lmtObj.makeInfo(vt_limit.get(k));
ht_limit.put(lmtObj.getLimit_type() + "_" + lmtObj.getPol_name(), lmtObj);
}
// (1) 노드 상태 CHECK
NRMoniLogger.out.info("1. NODE STATUS CHECK >");
if (moniInfo.isGrp_rac_yn()) {
NRMoniManager.monExe.checkNodeStatus(pstmt,
moniInfo.getGrp_code(), moniInfo.getVt_source_host());
}
// (2) CPMGR 상태 CHECK
NRMoniLogger.out.info("2. CPMGR STATUS CHECK >");
NRMoniManager.monExe.checkCPMGRStatus(pstmt, moniInfo);
// (3) CPMGR INFO CHECK
NRMoniLogger.out.info("3. CPMGR INFO CHECK >");
NRMoniManager.monExe.checkCPMGRInfo(pstmt, moniInfo);
// (4) DPMGR 상태 CHECK
NRMoniLogger.out.info("4. DPMGR STATUS CHECK >");
NRMoniManager.monExe.checkDPMGRStatus(pstmt, moniInfo);
    
```

그림 9. 실시간 모니터링 소스코드  
 Fig. 9. A Source-code of Real-time monitoring.

2. 복제 데이터의 정합성 검증 기능

[그림 10]은 복제 데이터의 정합성 검증 기능을 수행 하는 과정의 흐름도이며, 정합성 검증기능은 명령어 라인 인터페이스(Command Line Interface) 방식으로 데이터 검증 기능을 제공하도록 구현하였다.

데이터베이스 테이블에 대한 정합성 검증은 테이블

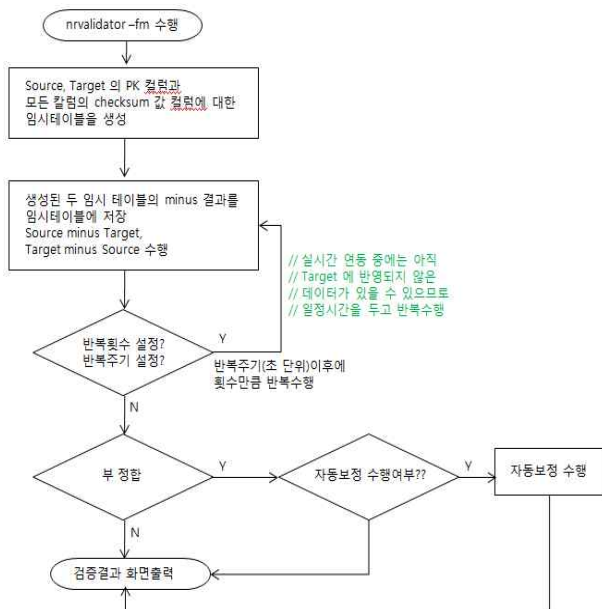


그림 10. 복제 데이터의 정합성 검증 알고리즘  
 Fig. 10. A Algorithm of consistency verification.

표 1. 데이터 정합성 검증 방법

Table 1. A method of data consistency verification.

```

#./nrvalidator

[USAGE]

## NR Data compare :

nrvalidator [ -fc | -pm | -fm ]

    -fc : Full count compare
    -pm : Primary Key column minus compare
    -fm : Full column minus compare
    
```

표 2. 복제 데이터의 정합성 검증 소m코드

Table 2. A Source-code of consistency verification.

```

// ## 2. 해당 테이블의 APMGR PORT 확인 ##
String sql = "SELECT A.POL_PORT, A.POL_CODE, B.TB_OBJID FROM
NR_POL A, NR_CLONETB B " +
" W H E R E
B.TB_OWNER = '"+NRCLValidator.consistInfo.getOwner()+"' +
"AND B.TB_NAME =
 '"+NRCLValidator.consistInfo.getTable()+"' +
"AND A.POL_CODE =
B.POL_CODE";
ResultSet rs = st.executeQuery(sql);
if ( rs.next() ) {
String port = rs.getString(1);
String pol = rs.getString(2);
NRCLValidator.consistInfo.setApmgrPort(port);
System.out.println("## Table Info. ##");
System.out.println("- Policy Code : " + pol);
System.out.println("- Policy Port : " + port);
} else {
System.out.println("[ERROR] Table is not exist : " +
NRCLValidator.consistInfo.getOwner() + " " +
NRCLValidator.consistInfo.getTable() + "\n");
System.exit(0);
}

// ## 사용할 테이블 DROP ##
NRCLValidatorUtil.dropTable4RM(targetCon1);

// ## 3. PK 정보 추출 ##
Vector vtPKInfo = NRCLValidatorUtil.getPKCol(catalogCon);

// ## 4. PKEY 임시테이블 생성을 위한 Query 생성 ##
NRCLValidatorUtil.makePKTbSQL4KM(vtPKInfo);

// ## 5. 임시테이블 생성 수행 ##
NRCLValidatorUtil.execPKTb4KM(targetCon1, targetCon2);

// ## 6. Minus 수행 Query 생성 ##
NRCLValidatorUtil.makeMinusSQL4KM(vtPKInfo);

// ## 7. Minus 수행
NRCLValidatorUtil.execMinus4KM(targetCon1, targetCon2);
    
```



표 3. 모든 컬럼의 정합성검증 소스코드

Table 3. A Source-code of consistency verification in all column.

```

## STEP 0.

PKEY Table Drop.
Table drop completed.

## STEP 1.
Get PK Column : PK value
- Primary Key : AGENCY_ID

Get PK Column Success.
## STEP 2.
PK, checksum table make
- Source Query exec
- Target Query exec

Source - PK, checksum table make start
Target - PK, checksum table make start
Source & Target - creating PK, checksum table...

Source PK, checksum table make completed.

## STEP 3.
Execute table minus

- Source minus Target exec
- Target minus Source exec
Source - PK, checksum table make start
Target - PK, checksum table make start
Source & Target - creating PK, checksum table...
S minus T - PK Minus start
S minus T - PK Minus count : 0
S minus T - PK Minus table : ncerti.ST_30

T minus S - PK Minus start
T minus S - PK Minus count : 0
T minus S - PK Minus table : ncerti.ST_40
validator complete!!
Elapsed : 648 sec
    
```

의 레코드 count 비교, Primary Key 에 대한 비교 그리고 전체 모든 데이터 값에 대한 검증을 수행할 수 있도록 구현하였으며 [표 1]은 사용방법에 대한 예시이다.

[표 2]는 복제 데이터의 정합성 검증을 위한 소스 코드의 각 단계에 주석을 첨부하여 제시하였다. [표 3]은 데이터베이스 테이블의 모든 컬럼을 비교대상으로 정합성 검증을 수행하는 예시이다.

## V. 구현결과 및 성능평가

### 1. 복제 시스템의 구현 환경

제안하는 시스템의 구현 환경은 다음과 같다.

표 4. 시스템 구현 환경

Table 4. Environment of Systems Implement.

분류	S/W 목록
OS	Solaris, HP-UX, AIX, Linux, MS Windows 등
실행 환경	Java 1.5.xx 이상 / Apache Tomcat 6.0.xx 이상
운영DB	Mysql 5.0.xx 이상 또는 Oracle
OS	Solaris, HP-UX, AIX, Linux, MS Windows 등
Java 실행 환경	Java 1.5.xx 이상
DBMS	ORACLE 9iR2 이상

## 2. 복제 시스템의 구현 결과

### 가. 정합성 검증 결과

제안하는 복제 시스템은 복제 데이터의 정합성 검증 기능을 지원하며 구현 결과는 [그림 11]과 같다.

시스템 각각의 프로세스 상태 및 클러스터의 활성화 여부, 이벤트 로그 발생, 지연 시간 관련정보를 실시간으로 모니터링 할 수 있다. 복제 적용 지연시간은 정책별로 지정한 임계값을 기준으로 정상, 경고, 심각 등의 세분화된 상태를 제공한다.

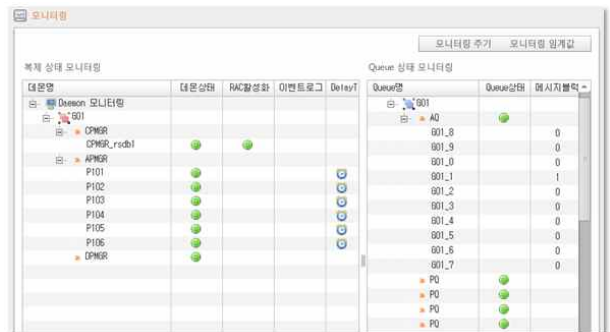


그림 11. 정합성 검증 화면

Fig. 11. A Display of consistency verification.

### 나. 실시간 상세모니터링 기능

제안하는 시스템은 서버의 상태 및 프로세스 구동 상태 등 [그림 12]와 같이 복제상태의 실시간 모니터링 기능을 제공한다.

상세모니터링 기능은 특정 복제그룹을 선택하면 현재 근원지 시스템의 캡처 진행 상태와 대상 시스템으로의 적용 상태를 확인할 수 있으며, 프로세스들의 활성화, 비활성 여부 등을 체크하여 장애가 발생한 프로세스들을 자동으로 구동한다.

제안하는 복제시스템은 복제기능의 성능을 향상하기

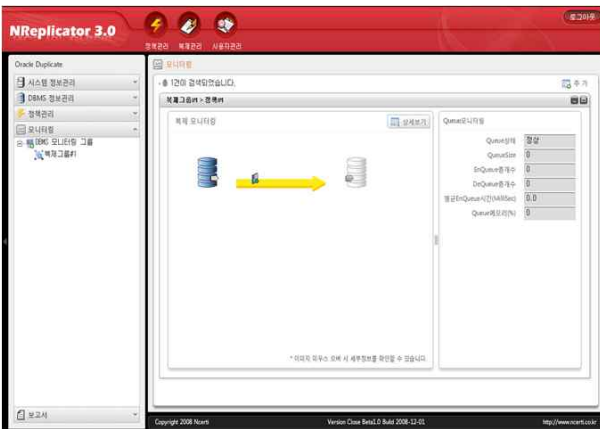


그림 12. 실시간 모니터링 화면  
Fig. 12. A Display of Real-time Monitoring.



그림 13. 상세 모니터링 화면  
Fig. 13. A Display of Detail Monitoring.

표 5. 기존 시스템과의 비교  
Table 5. Compare against others.

[ O:지원 X:미지원 ?:확인불가 ]

항목	Q4 S제품 (외산)	I4 S제품 (외산)	D4 S제품 (국산)	제안하는 시스템
다중 OS 플랫폼간 복제지원	O	O	X	O
복제대상 및 단위	전체데이터를 혹은 일부 테이블	전체데이터를 혹은 일부 테이블	전체데이터를	전체데이터를 혹은 일부 테이블
운영 형태	Active-Active	Active-Active	Active-Standby(Active)	Active-Active(Standby)
구축 형태	1:1, 1:N, N:1, N:N	1:1, 1:N, N:1, N:N	1:1, 1:N, N:1, N:N (서버구조기준)	1:1, 1:N, N:1, N:N (Instance구조기준)
관리 Console	Web기반 GUI	CS기반 GUI	CS기반 GUI	Web기반의 한글지원 관리화면
압축전송 지원	X	X	O	O
복제모니터링 가능	O	O	O	O
파일동기화 지원여부	X	X	O	O
역방향 복구 지원	O	O	O	O
운영자 실수 필터링 가능	O	?	?	O
TDE 클라이언트와 전송 가능	X	X	?	O
부하별 동적 스레드 적용 가능	X	X	X	O
정합성 보정, 검증기능	O	?	O	O
커스터마이징 가능여부	X	X	O	O
기술지원 (개발사레벨지원)	X	X	O	O

위하여 다중 스레드 대상 데이터베이스 적용하였으며, 이에 대한 성능평가를 실시하였다. 이때 성능평가의 조건으로 1개의 테이블에 대하여 약 1000만건의 건 단위

표 6. 성능평가  
Table 6. Performance evaluation.

항목	1 Thread	4 Thread	8 Thread
복제 수행시간 (분)	145	74	53
복제성능 (건수/sec)	1150	2248	3120

위탁(commit) 트랜잭션이 발생하도록 하였으며, 단일 스레드인 경우 1초당 1150건의 복제성능을 보인 반면에 8중 스레드 방식을 적용한 경우는 1초당 3120건의 복제 성능 결과를 얻었다. 이로서 본 연구에서 구현한 복제 시스템은 다중 스레드 방식을 적용함으로써 기존의 단일 스레드 적용 방식보다 약 3배 이상의 복제 성능이 향상된 결과를 얻었다.

그리고 단일 스레드와 다중 스레드 방식일 때의 복제 시스템의 성능평가를 하였는데 단일 스레드인 경우 145분이 소요되는 반면에 다중(8)스레드 방식의 경우는 복제 수행시간이 53분 소요됨으로써 본 연구에서 제안하는 다중 스레드 복제 처리방식은 기존 방식보다 약 3배 이상의 우수한 성능결과를 보였다.

## VI. 결 론

본 연구는 클라우드 컴퓨팅 기반의 고가용성 복제 시스템을 구현하였다. 구현한 복제 시스템은 복제성능 향상을 위해 다중 스레드 대상 데이터베이스 방법을 적용하였으며, 파일시스템의 실시간 동기화 기술로 네트워크 효율을 향상시킬 수 있으며, 액티브 대 액티브 운영으로 백업시스템 활용을 극대화함으로써, 근원지 데이터베이스 시스템의 부하를 분산시키는 효과를 갖게 한다. 이밖에도 복제 데이터의 정합성 검증 기능과 데이터 센터의 실제 운영에서 요구되는 모니터링 기법 등을 구현하였다. 성능평가에서도 다중 스레드 방식을 사용함으로써 단일 스레드 방식보다 약 3배 이상의 복제 성능을 보여주었다. 이처럼 고성능, 고가용성을 위한 데이터 복제의 적극적인 이용 등에 보다 적합한 기능이다. 이처럼 대용량 데이터의 처리와 관리에 관한 기술은 데이터 센터의 운영 기술과 더불어 클라우드 컴퓨팅의 주요 기술요소이지만 데이터가 연동되고 자원을 다양하게 활용하는 것에는 데이터 보호와 자원의 관리 정책, 기업 비밀 관리나 개인의 프라이버시 측면에서의 문제점

도 존재한다. 그러므로 향후 연구는 클라우드 컴퓨팅 기반의 고가용성 복제 시스템에 강력한 보안기능으로 신뢰할 수 있는 컴퓨팅 기술을 기반으로 기밀성 보호와 데이터의 무결성 그리고 원격 확인을 위한 가상머신 보호와 관련한 연구를 진행하여야 할 것이다.

참 고 문 헌

[1] 강승석, 손예진, 문은지, “클라우드 컴퓨팅 서비스 구현을 위한 네트워크 가상화 연구”, 한국지역정보화학회, 한국지역정보화학회지, 제13권 제3호 2010.9, page(s): 1-17

[2] 김성준, “클라우드 컴퓨팅환경에서의 기업정보보안 방안”, 공주대학교 KNU 경영컨설팅 연구소, 경영컨설팅리뷰, 제1권 제2호 2010.8, page(s): 194-208

[3] 용덕, 김상욱, “클라우드 컴퓨팅을 위한 클라우드 수퍼바이저 운영체제 설계”, 한국멀티미디어학회, 2010년도 추계학술발표논문집 2010.11, page(s): 245-247

[4] 김명호, 김재우, 장현춘, “클라우드 컴퓨팅의 오늘과 내일”, 한국정보보호학회, 정보보호학회지, 제20권 제2호 2010.4, page(s): 56-64

[5] 이경하, 최현식, 정연돈, “클라우드 컴퓨팅에서의 대용량 데이터 처리와 관리 기법에 관한 조사” 한국정보과학회, 정보과학회논문지 : 데이터베이스, 제38권 제2호 2011.4, page(s): 104-125

[6] 임철수, “클라우드 컴퓨팅 보안 기술”, 한국정보보호학회, 정보보호학회지, 제19권 제3호 2009.6, page(s): 14-17

[7] He Dian, Liang Ying, Hang Zhi, “Replicate Distribution Method of Minimum Cost in Cloud Storage for Internet of Things”, Network Computing and Information Security (NCIS), 2011 International Conference, 14-15 May 2011

[8] <http://www.oracle.com>

[9] <http://download.oracle.com/docs/>

[10] <http://www.oracle.com>

저 자 소 개



**박 성 원**(정회원)-교신저자  
 1984년 숭실대학교 전자계산학 학사 졸업.  
 2000년 한국과학기술원 공학석사 졸업  
 2010년~숭실대학교 IT정책 경영학과 박사과정  
 <주관심분야 : 클라우드컴퓨팅, 정보보호>



**이 문 구**(정회원)  
 1984년 숭실대학교 전자계산학 (학사)  
 1993년 이화여자대학교 대학원 전산교육학 (석사)  
 2000년 숭실대학교 대학원 컴퓨터 시스템 (공학 박사)  
 2000년 3월~현재 김포대학 IT 학부인터넷정보과 부교수  
 <주관심분야 : 인터넷 보안, 시스템 성능평가 및 검증, 암호화 알고리즘, 전자상거래 보안, 멀티미디어 콘텐츠 보안>



**이 남 용**(정회원)  
 1983년 고려대학원 경영대학원 석사 졸업.  
 1993년 미국미시시피주립대 박사 졸업, 경영학박사  
 <주관심분야 : SW테스트, 품질보증, MIS, 정보보호>