

<http://dx.doi.org/10.7236/JIWIT.2012.12.6.147>

JIWIT 2012-6-19

## 백업시스템을 위한 복제기능의 정합성과 성능평가

### Consistency and Performance Evaluation of Duplication Function for Backup Systems

성해경\*, 이문구\*\*

HaeKyung Seong, Moongoo Lee

**요약** IT 환경이 점점 복잡해지고 인프라의 규모가 커지면서 정보시스템의 안전성과 데이터의 가용성에 대한 인식이 더욱 가중되고 있다. 이처럼 정보자산의 가치가 중요시되는 시점에서 재해복구 대응책을 위한 고성능 백업시스템에 대한 연구는 비즈니스에 직접적인 영향을 미치는 분야이다. 그러므로 본 연구는 고성능 백업시스템을 위한 복제 데이터의 정합성검증을 시행하였다. 그리고 DB 테이블 복제 처리 성능 비교와 네트워크 대역폭에 따른 복제 시스템의 트랜잭션 처리 속도와 성능 평가를 시행하였다.

**Abstract** As IT circumstance is complicated and the scale of infra increase, a recognition of security information system and data availability are raised. At this point, a value of information assets regards important, a study of high performance backup system for disaster recovery countermeasure is a field to impact on business directly. Therefore, this study executes duplication of high performance backup system and data consistency verification. And, a performance of comparison to DB table recovery process, a transaction process speed of duplication according to network bandwidth, and performance evaluation were executed.

**Key Words** : Consistency(정합성), Duplication(복제), Availability(가용성)

## I. 서론

IT 환경이 점점 복잡해지고 인프라의 규모가 커지면서 정보시스템의 안전성과 데이터의 가용성에 대한 인식이 더욱 가중되고 있으며, 정보자산의 가치가 중요시되는 시점에서 재해복구 대책을 위한 고성능 백업시스템에 대한 연구는 비즈니스에 직접적인 영향을 미치는 분야이다<sup>[1]</sup>.

본 논문의 구성은 다음과 같다. 2장에서는 기존의관련 연구에 대하여 기술하고, 3장에서는 복제시스템의 정합성 검증 기술, 4장에서는 다양한 조건과 환경에서의 성능

평가 결과에 대하여 기술하였으며, 5장에서는 결론을 맺는다.

## II. 관련 연구

### 1. 기존의 복제 및 백업 시스템

기존의 백업 방식은 운영 데이터베이스의 업데이트나 백업과정이 일어나면 그 이력이 개장 기록(redo log) 파일에 저장되어 있고 이 기록파일이 어느 정도의 지정된 크

\*정회원, 한양여자대학교 컴퓨터정보과

\*\*정회원, 김포대학교 인터넷정보과

접수일자 : 2012년 9월 5일, 수정완료 : 2012년 11월 3일

게재확정일자 : 2012년 12월 14일

Received: 5 September 2012 / Revised: 3 November 2012 /

Accepted: 14 December 2012

\*Corresponding Author: hkseong@hywoman.ac.kr

.Dept. of Computer Information, Hanyang Women's University, Korea

기를 넘어서면 순차적으로 기록보관소 기록(archive log) 파일들로 스위칭을 하며 저장한다. 예를 들어 기록보관소 기록 파일의 크기를 4GB 단위로 지정을 하면 데이터베이스의 변경되는 이력이 개장 기록에 4GB까지 저장이 되면 기록보관소 기록의 인덱스를 하나 증가 시켜 스위칭을 한다. 개장(redo) 기록 파일은 데이터베이스의 변경이 생기는 동안 계속 변경되기 때문에 기존 백업으로는 데이터를 받을 수 없다. 이처럼 기존의 데이터베이스 백업 방식은 백업되지 못한 기록보관소(archive) 기록(log) 파일의 손실이나 온라인 개장 기록(redo log) 데이터의 손실, 그리고 실시간 처리량이 많은 서버의 경우 데이터의 손실률이 증가 하여 데이터의 복구 시 시간이 오래 걸리고 복구의 어려움이 있다. 이러한 기존의 데이터베이스 백업 방식의 구성은 그림 1과 같다<sup>[2,3]</sup>.

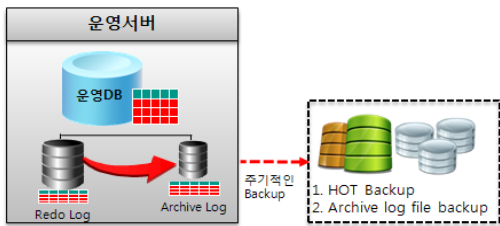


그림 1. 기존 백업 시스템의 구성  
Fig 1. Configuration of the existing Backup System

### 2. 제안하는 복제 및 백업 시스템의 구성

기존 방식의 문제를 해결하기 위해서 그림 2와 같이 하이브리드 백업 시스템을 구성하여 온라인 개장 기록(redo log)을 처리기록(transaction log) 데이터베이스로 실시간 백업하고 기존 방식의 백업에서 누락 될 수 있는 자료보관소 기록(archive log)으로 백업한다. 그러므로 데이터의 실시간 복구가 가능하고 복구 시 온라인으로 개장 기록(redo log)까지 복구가 가능함으로 데이터의 손실을 최소화 할 수 있다. 이러한 개장기록의 복구를 위해 변경된 데이터를 반영하는 SQL 스크립트의 변환이 가능하고, 또한 데이터베이스에서 같은 PK(primary key)에 대해서 변경되는 데이터들을 같은 스텀드로 모아 실행시키는 멀티스레드 대상 데이터베이스 적용 방식으로 실시간 변경되는 데이터에 대한 복제 성능도 향상 시킬 수 있다<sup>[4,5]</sup>.

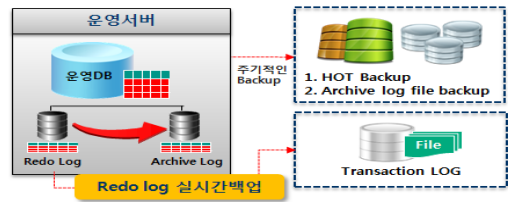


그림 2. 제안하는 백업 시스템의 구성  
Fig 2. Configuration of Proposed Backup system

### 3. 복제 및 백업 시스템의 복구 절차

실시간 데이터의 복제된 데이터의 복구 절차는 HOT 백업 및 자료보관소(archive) 기록을 이용한 데이터베이스 복구와 복제를 이용한 개장 기록영역을 복구하는 두 단계로 진행한다. 첫째 HOT 백업 및 자료보관소 기록을 이용한 복구는 HOT 백업 파일을 복사하고 데이터베이스의 자체 복구를 이용하여 데이터베이스를 복구한다. 둘째는 복제 에이전트(Agent)를 이용한 개장 기록(redo log) 영역 복구 절차는 다음과 같은 순서로 수행된다.

- ① 기록보관소 기록 복구가 완료된 시점의 SCN 확인
- ② 확인된 SCN으로 처리 기록(transaction log) 파일 선택  
: 카탈기록 정보를 이용하여 기록 파일 자동선택  
: 카탈기록 정보가 없을 경우 기록파일 이름을 이용하여 선택 가능  
: 파일 이름의 STARTSCN 정보 활용
- ③ 선택된 기록파일 Agent에서 순차적읽기수행
- ④ 읽은 기록파일의 데이터를 parsing한 후 적용

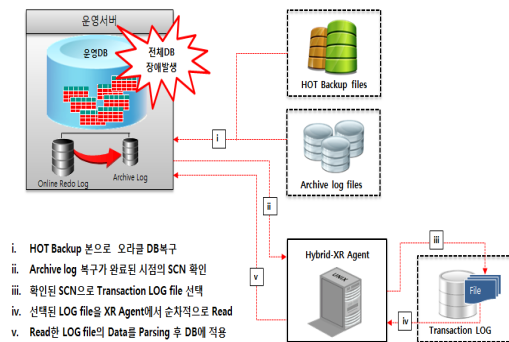


그림 3. 데이터베이스 전체 복구  
Fig 3. Full Recovery of DatabasePlease

하이브리드 실시간 데이터베이스 백업의 복구 시 데

이더베이스 전체에 문제가 발생한 경우 그림 3과 같은 순서로 의해서 데이터 복구가 진행된다. 만약, 특정 데이터베이스의 테이블 단위에서 문제가 발생한 경우 그림 4와 같은 순서로 데이터 복구가 진행된다.

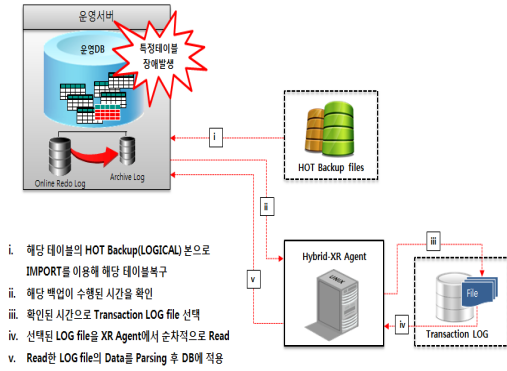


그림 4. 데이터베이스 테이블 단위 복구  
 Fig 4. Table Unit Recovery of Database

### III. 복제 데이터의 정합성 검증

#### 1. 복제기술

제안하는 백업시스템은 데이터의 무결성을 위하여 복제 데이터의 정합성 검증기술을 구현하였다. 복제 데이터의 정합성 기술은 그림 5와 같이 전송 데이터 암호화(Transparent Data Encryption) 컬럼에 대한 복제 기능을 제공하며, 이러한 기능은 전송데이터 암호화가 적용된 컬럼의 데이터만 추출하여 최적화된 SQL 구문의 생성 및 복제를 수행한다<sup>[6]</sup>.

이처럼 전송데이터 암호화 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 자료를 이용하여 복구 작업을 수행하지 않고도 데이터베이스의 과거 시점의 데이터를 확인할 수 있으므로 신속한 시점에 복구 작업을 수행할 수 있다. Flashback query는 Oracle 9i부터 제공되는 기능으로, 과거의 특정 시점을 기준으로 데이터베이스를 조회하고자 한다면, Flashback query 기능을 이용할 수 있다. Flashback query는 특정 시점 또는 SCN(system change number)을 기준으로, 해당 시점에

커밋된 데이터를 조회할 수 있게 한다. 컬럼 암호화가 적용된 DB 테이블의 복제시, 해당 트랜잭션이 발생한 시점의 컬럼값을 추출하기 위해 Flashback query가 사용되어 질 수 있다. 즉, 복제를 위해 기록(log)으로부터 추출된 트랜잭션의 컬럼이 암호화 되어 있을 경우, 해당 트랜잭션이 발생한 시점의 컬럼값을 Flashback query를 통해 추출하여 복제를 수행한다<sup>[7]</sup>.

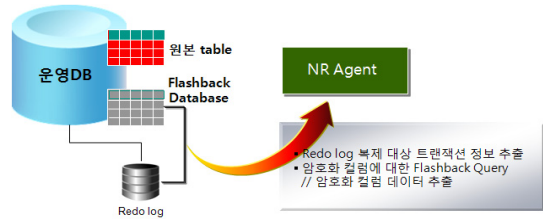


그림 5. 전송데이터 암호화 컬럼 복제 기능  
 Fig 5. Encryption Column Duplication Technique of Transport Data

#### 2. 복제 데이터의 정합성 검증

다음 6단계는 복제 데이터의 정합성 검증 기능을 수행하는 알고리즘으로 정합성 검증기능은 명령어 라인 인터페이스(Command Line Interface) 방식으로 데이터 검증 기능을 제공하도록 구현하였다.

- step1: Source, Target의 PK(primary key) 컬럼과 모든 컬럼의 Checksum 값 컬럼의 임시테이블 생성
- step2: 생성된 두 임시테이블의 minus 결과를 임시 테이블에 저장
- step3: Source minus Target과 Target minus Source 수행
- step4: 일정시간 step3 반복수행
- step5: 부정합이면 자동 보정과정을 수행하지 판단
- step6: 자동보정 수행 후 검증결과 화면 출력

DBMS 복제 데이터에 대한 검증은 근원시스템에 트랜잭션 발생이 중지된 상태와 트랜잭션이 발생하고 있는 운영중인 상태 모두 가능하도록 구현되어야 한다. 데이터 정합성 검증의 항목은 기본적으로 테이블에 대한 레코드 카운트, PK(Primary Key) 값에 대한 비교 그리고 모든 레코드들에 대한 정합성 검증 기능이 될 수 있다. 특히 모든 레코드들에 대한 효율적인 정합성 비교 검증을 위해서 본 논문에서는 레코드들의 모든 컬럼값에 대

한 체크섬(checksum)값을 기반하는 검증방식을 사용하였다. 운영 중인 상태에서도 정합성 검증을 수행하기 위해 정합성 검증 수행시점의 근원지시스템의 데이터를 기준으로 대상시스템에 트랜잭션 반영이 완료되는 시점까지 반복 수행하여 검증하는 방식을 사용하였다. 데이터 정합성 (data accuracy)에 대한 검증을 수행할 수 있도록 CLI(command line interface) 방식으로 구현하다.

그림 6 은 복제 데이터의 정합성 검증을 위한 소스 코드의 각 단계이다.

```

// ## 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 " + "WHERE
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 vPKInfo = NRCLValidatorUtil.getPKCol(catalogCon);
// ## 4. PKEY 임시테이블 생성을 위한 Query 생성 ##
NRCLValidatorUtil.makePKTbSQL4KM(vPKInfo);
// ## 5. 임시테이블 생성 수행 ##
NRCLValidatorUtil.execPKTb4KM(targetCon1, targetCon2);
// ## 6. Minus 수행 Query 생성 ##

NRCLValidatorUtil.makeMinusSQL4KM(vPKInfo);
// ## 7. Minus 수행
NRCLValidatorUtil.execMinus4KM(targetCon1, targetCon2);
    
```

그림 6. 데이터 정합성 검증을 위한 소스코드  
Fig 6. Source Code of Data Consistency Verification

```

## 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
    
```

그림 7. 모든 컬럼의 정합성검증 소스코드  
Fig 7. Consistency Verification Source Code of All Coloum

그림 7은 데이터베이스 테이블의 모든 컬럼을 비교 대상으로 정합성 검증을 수행하는 예시이다.

비교대상 테이블 PK 추출 등 사전 작업 후 동일한 시점에 QUERY 수행을 위해 해당 테이블이 존재하는 정책 별 APMGR 통신포트를 추출한다. 성능을 최적화하기 위하여 PK 및 각 모든 컬럼의 체크섬(checksum) 값을 임시 테이블에 생성해서 넣는다. 만약 임시 테이블이 이미 존재 할 경우 DROP TABLE을 수행한다. 추출한 임시 테이블은 S-T, T-S를 통해 정합성이 일치함을 확인 할 수 있다.

제안하는 복제 시스템은 복제 데이터의 정합성 검증 기능을 지원하며 구현 결과는 그림 8과 같다. 시스템 각자의 프로세스 상태 및 클러스터의 활성화 여부, 이벤트

기록(log) 발생, 지연 시간 관련정보를 실시간으로 모니터링 할 수 있다. 복제 적용 지연 시간은 정책별로 지정된 임계값을 기준으로 정상, 경고, 심각 등의 세분화된 상태를 제공한다.

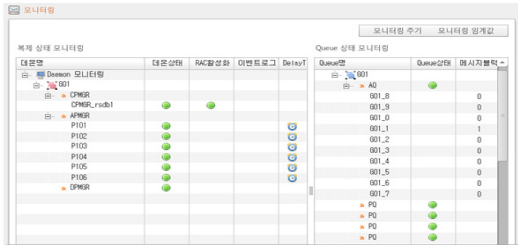


그림 8. 정합성 검증 화면  
Fig 8. Display of Consistency Verification

#### IV. 성능평가

제한한 백업시스템은 사용자의 실수 또는 어플리케이션 오류로 인해 데이터베이스 테이블 데이터가 손상되었을 때, 과거 특정시점의 데이터에 대하여 질의를 통해 확인하고 복구할 수 있다. 이는 데이터 복제기능을 기반으로 수행되므로 데이터베이스 백업 자료를 이용하여 복구 작업을 수행하지 않고도 데이터베이스의 과거 시점의 데이터를 확인할 수 있으므로 신속한 시점에 복구 작업을 수행 할 수 있다.

본 장에서는 데이터베이스 테이블 복제 처리능력에 대한 성능 비교와 네트워크 대역폭에 따른 복제 시스템의 트랜잭션 처리 속도에 대한 실무에서의 실험 및 성능평가 결과이다.

##### 성능평가 1 : DB 테이블 복제 처리 성능 비교

A통신사의 월별 고객 빌링 배치 작업으로 1,500만건 이상의 트랜잭션의 처리를 수행하는 작업이다. 그림 10에서 나타난 것처럼 멀티키 방식의 복제가 기존 복제 방식에 비해 약 2.1배 이상의 데이터 복제의 성능 우위를 나타내고 있다. 수행된 배치 작업은 통신사의 월정기 빌링 작업으로 초당 약 4000 ~ 5000건씩 1건단위 commit으로 이루어진 작업으로써 멀티스레드 적용시 1개 Thread 대비 약 50%이상 성능 향상 효과를 보이고 있다. 트랜잭션의 성격이 1건 단위 커밋(commit)일 경우, DB 특성상 대량으로 한번에 커밋(commit)을 수행 하는 것보다 성능

이 떨어지게 되어 있다. 따라서, 특히 1건 단위 커밋(commit)과 같이 소량의 커밋(commit)이 발생하는 특성을 가진 트랜잭션의 경우, 멀티키 적용 방식의 복제 시스템이 기존 복제 시스템에 비해 2배 이상의 성능 향상 효과가 있음을 확인 할 수 있다.

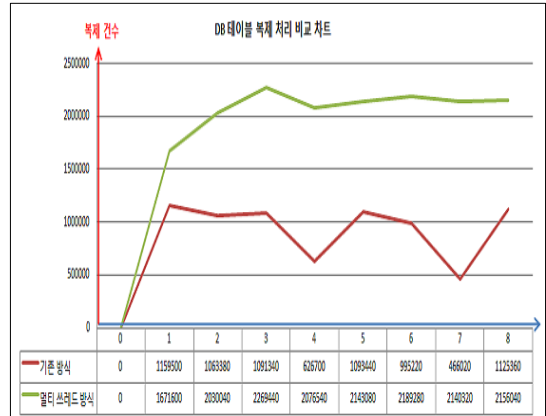


그림 9. DB 테이블 복제 처리 성능 비교  
Fig 9. Performance Comparison of DB Table Recovery Process

성능평가 2 : 네트워크 대역폭에 따른 복제 시스템의 트랜잭션 처리 속도

B통신사의 일일 배치 작업으로 초당 약 4000 건씩 발생하는 작업으로 네트워크 대역폭이 복제 성능에 미치는 영향을 테스트 하였다. 네트워크 대역폭이 좋지 않고 복제 데이터 량이 대량일 경우 네트워크 병목현상으로 인해 복제 성능이 떨어짐을 볼 수 있다. 이러한 결과는 네트워크가 100Mbps 일 경우, 네트워크 부하로 인해 데이터 전송 지연이 발생하여 대상DB에 반영하는 APMGR 프로세스의 멀티스레드 성능을 나타낼 수 없다. 즉, 대량 트랜잭션 발생 시 네트워크 부하가 심할 경우, 멀티스레드를 이용한 성능 효과를 극대화하기 위해서는 충분한 네트워크 대역폭의 확보가 필요하다.

기존 복제 시스템이 네트워크 대역폭의 크기에 비례해서 데이터 복제 성능이 증가하지 않는 반면에 멀티키 적용 방식의 복제 시스템은 네트워크 대역폭이 커질수록 비례해서 복제 성능도 높아짐을 알 수 있다. 네트워크 대역폭의 크기에 있어서도 멀티키 복제 시스템이 기존 복제 시스템의 복제 성능에 비해 약 2배 이상의 성능 우위를 나타내고 있다.

표 1. 네트워크 대역폭에 따른 복제 시스템의 트랜잭션 처리 속도  
Table 1. Transaction Process Speed of Duplication System according to Network Bandwidth

	네트워크	스레드 개수	1시간	2시간	3시간	4시간	5시간
기존 복제	100 Mbps	1	6,55 5,13 5	12,7 84,4 21	15,3 54,3 78	18,5 98,4 86	21,8 42,5 95
멀티키 복제		10	14,6 58,6 59	32,2 49,0 50	39,4 41,8 43	42,1 68,7 71	43,5 93,0 32
기존 복제	1 Gbps	1	6,82 9,82 1	15,0 25,6 05	21,1 72,4 44	27,3 19,2 83	32,7 83,1 39
멀티키 복제		10	15,2 21,4 30	33,4 87,1 46	48,7 08,5 76	60,8 85,7 20	73,0 62,8 64

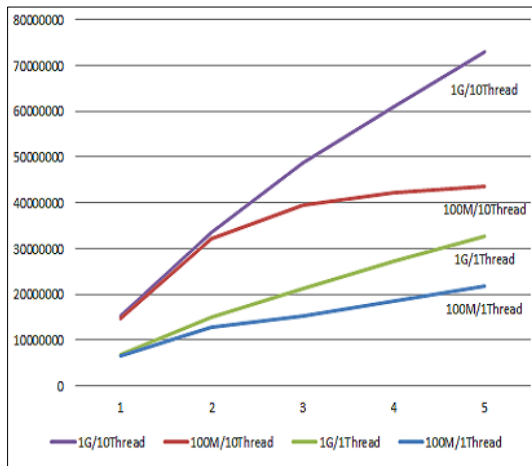


그림 10. 네트워크 대역폭에 따른 복제 시스템의 성능 평가  
Fig 10. Performance Evaluation of Duplication System according to Network Bandwidth

## V. 결론

본 연구에서 제안하는 백업시스템은 멀티키 적용 방식을 구현하였으며, 구현한 멀티키 적용 방식의 복제 시스템의 데이터 정합성(data accuracy)에 대해서 검증 을 하였다. 그리고 실제 구현 환경에서 데이터베이스 테이블 복제 처리 성능은 기존의 방법에 비해 2배로 향상되었다. 네트워크 대역폭에 따른 복제 시스템의 트랜잭션 처리속도와 성능은 트랜잭션 처리 속도기존에는 재해 발

생시 백업데이터를 기반으로 복구를 하였으나 복제기능을 기반으로 실시함으로써 시스템의 성능향상은 물론 1.5~2.0배 이상 성능의 우수성을 보여주었다.

## 참고 문헌

- [1] Aggarwal, Nidhi, "Achieving high availability with commodity hardware and software", The University of Wisconsin - Madison, 2008
- [2] Feiselian Tan, Sebastien Ardon, Max Ott, "Ubistore : Ubiquitous and Opportunistic Backup Architecture", Proceedings of the Fifth Annual, IEEE International Conference on Pervasive Computing and Communications Workshops, 2007,
- [3] G. Cao, L.Yin and C. Das, "A Cooperative Cache Based Data Access Framework for Ad Hoc Networks", IEEE Computer, Feb. 2004
- [4] Ji, Minwen, "Availability, scalability and a cost-effectiveness of cluster-based Internet infrastructures", Princeton University, 2001
- [5] Lars Frank, "Evaluation of the Basic Remote Backup and Replication Methods for High Availability Databases", Software-Practice and Experience, Vol. 29, No.15, 27 July, 1999.
- [6] Seshadri, "Enhancing availability in large scale storage systems and services: Architectures and techniques", Sangeetha Institute of Technology, 2009
- [7] X. Li, C. Xie, Q. Wei, and Cao, "A reliable Scheme for Cluster Storage System", IEEE, In proceedins of 3th International Conference on Semantics, Knowledge and Grid, pp. 394-397, Oct. 2007

※ 본 논문은 한양여자대학교 2011년2기 교내연구비의 지원으로 수행된 연구결과임.

### 저자 소개

#### 성 해 경(정회원)



- 1976년 : 홍익대학교 전자계산학과 (학사)
- 1978년 : 홍익대학교 대학원 전자계산학과(석사)
- 2000년 : 대전대학교 대학원 컴퓨터공학과 소프트웨어공학(공학박사)
- 1987년~현재 : 한양여자대학 컴퓨터

정보과 교수

<주관심분야 : 소프트웨어공학, 웹 프로그래밍, 멀티미디어 콘텐츠 보안, 인터넷 보안>

#### 이 문 구(정회원)



- 1984년 : 숭실대학교 전자계산학(학사)
- 1993년 : 이화여자대학교 대학원 전산교육학(석사)
- 2000년 : 숭실대학교 대학원 컴퓨터시스템 (공학 박사)
- 2000년~현재 : 김포대학 IT 학부 인

터넷정보과 부교수

<주관심분야 : 인터넷 보안, 시스템 성능평가 및 검증, 암호화 알고리즘, 전자상거래 보안, 멀티미디어 콘텐츠 보안>