

논문 2012-49CI-2-17

클라우드 컴퓨팅에 적합한 LDPC 부호 복제 기법

(An LDPC Code Replication Scheme Suitable for Cloud Computing)

김 세 희*, 이 원 주**, 전 창 호***

(Se Hoe Kim, Won Joo Lee, and Chang Ho Jeon)

요 약

본 논문에서는 클라우드 컴퓨팅에 적합한 LDPC 부호 복제 기법을 제안한다. 이를 위해 먼저 파일 가용성과 스토리지 오버헤드 분석을 통해 클라우드 컴퓨팅에 적합한 블록 수와 LDPC 부호 종류를 결정한다. 그리고 LDPC 부호를 정의하는 Tanner 그래프의 무작위 생성 방법과 생성한 LDPC 부호들의 반복적인 복호화 과정을 통한 성능 비교 방법을 제시한다. 또한 실험을 통하여 클라우드 컴퓨팅에 적합한 LDPC 부호 복제 기법의 그래프 정규성과 총 간선 수에 따른 성능 변화를 분석하였다. 이를 통해 LDPC 부호의 그래프 정규성이 좌-정규와 약 좌-정규 일 때와 총 간선 수가 최소이거나 최소에 근접한 값을 가질 때 가장 성능이 좋음을 확인할 수 있었다.

Abstract

This paper analyze an LDPC code replication method suitable for cloud computing. First, we determine the number of blocks suitable for cloud computing through analysis of the performance for the file availability and storage overhead. Also we determine the type of LDPC code appropriate for cloud computing through the performance for three types of LDPC codes. Finally we present the graph random generation method and the comparing method of each generated LDPC code's performance by the iterative decoding process. By the simulation, we confirmed the best graph's regularity is left-regular or least left-regular. Also, we confirmed the best graph's total number of edges are minimum value or near the minimum value.

Keywords : Cloud computing, Cluster file system, LDPC code, Replication method

I. 서 론

사용자들이 개별적으로 IT인프라를 구축하고 유지, 보수하는 부담이 없이 컴퓨팅 자원이나 서비스를 필요

한 만큼 이용할 수 있다는 점에서 클라우드 컴퓨팅이 각광받고 있다. 이런 추세에 따라 최근에는 Google, MS, Amazon, IBM 등과 같은 글로벌 기업들이 클라우드 서비스 사업에 나서고 있다^[1~2]. 그러나 서비스 품질과 운영비용의 균형이 이루어지는 클라우드 서비스를 제공할 수 있으려면 여러 가지 기술적 기반을 갖추어야 한다. 먼저 경쟁력 있는 가격으로 일정 수준의 성능을 보장하는 서비스를 제공하기 위해서는 대단위의 클러스터 구축이 필수적이다. 또한 클러스터에 저장된 수많은 사용자들의 데이터를 안전하게 관리하기 위해서는 대용량의 데이터 관리 기술이 필요하다. 특히 대용량 데이터를 효율적으로 저장 및 관리하면서 네트워크나 노드

* 정회원, SK커뮤니케이션즈 기술연구소 MI팀
(SK Communications, Technology Innovation Lab.,
,Media Intelligence Team)

** 정회원, 인하공업전문대학 컴퓨터정보과
(Department of Computer Science, Inha Technical
College)

*** 정회원, 한양대학교 ERICA 캠퍼스 컴퓨터공학과
(Department of Computer Science & Engineering,
Hanyang University ERICA Campus)

접수일자: 2011년10월22일, 수정완료일: 2012년3월5일

의 장애에도 서비스의 중단 없이 빠른 복구가 가능한 클러스터 파일 시스템이 필수적이다^[3].

대표적인 클러스터 파일 시스템으로는 GFS(Google File System)^[4], HDFS(Hadoop Distributed File System)^[5], 그리고 OwFS(Owner-based File System)^[6] 등이 있다. 이러한 클러스터 파일 시스템에서는 대용량 데이터를 효율적으로 저장 및 관리하면서, 네트워크 고장이나 노드 실패와 같은 장애 발생 시에도 서비스의 중단이나 데이터 손실 없이 빠르게 복구를 하기 위해 공통적으로 데이터 복제 기법을 적용하고 있다.

클러스터 파일 시스템들에서 주로 사용하는 데이터 복제 기법은 전체-파일(Whole-file) 복제 기법이다^[7]. 이 기법은 원본 파일에 대해서 미리 정해진 수만큼 복제본을 생성하고 분산 저장하기 때문에 LDPC 복제 기법에 비해 구현이 간단하다. 또한 복제본 배치 정책에 따라서 시스템에 분산 저장하므로 다중 데이터 읽기 성능을 높일 수 있다. 기본적으로 3개의 복제본을 저장하는데 복제본 수가 커질수록 고가용성을 보장한다는 장점을 가진다. 하지만 복제본 수만큼 데이터를 중복 저장하므로 스토리지 오버헤드가 커서 시스템 구축비용을 증가시키는 단점을 가진다.

본 논문에서는 클라우드 컴퓨팅 환경에서 전체-파일 복제 기법보다 우수한 데이터 복제 기법을 제안하고 그 유효성을 검증한다. 우리가 제안하는 복제기법은 LDPC (Low-Density Parity-Check)부호 복제 기법에 기반한 것으로써 전체-파일 복제 기법보다는 스토리지 오버헤드가 작으면서 파일 가용성은 더 높은 장점을 보인다.

본 논문의 구성은 다음과 같다. II장에서 관련 연구를 소개하고 III장에서는 우리가 제안하는 클라우드 컴퓨팅에 적합한 LDPC 부호 복제 기법에 대해 기술한다. IV장에서는 성능평가 과정과 분석 결과를 제시하고, V장에서 결론을 내린다.

II. 관련 연구

1. LDPC 부호

LDPC 부호는 검사행렬(parity-check matrix)의 원소가 많은 수의 0과 극히 적은 수의 1을 포함하는 선형 블록 부호(linear block code)로써 Shannon의 채널 용량의 한계에 근접하는 우수한 부호이다. 그림 1은 Tanner 그래프를 사용해서 LDPC 부호를 정의한 예이다^[8].

그림 1에서 그래프는 n개 좌측 노드와 m개 우측 노

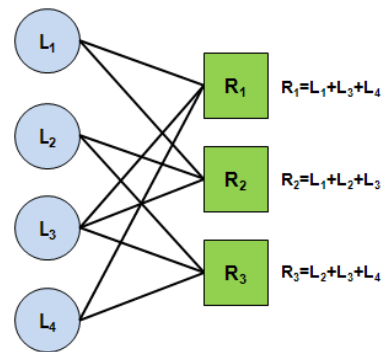


그림 1. (4, 3) LDPC 부호의 Tanner 그래프 예제
Fig. 1. An example Tanner graph of (4, 3) LDPC code.

드, 그리고 좌측 노드와 우측 노드를 연결하는 간선들을 가진다. 이때 각 m개의 우측 노드는 각각의 우측 노드에 연결된 좌측 노드들의 exclusive-OR연산으로 생성한다.

LDPC 부호를 정의하는 Tanner 그래프에서 좌측 노드와 우측 노드간의 연결 형태를 그래프 정규성(Regularity)이라 말한다. 이러한 그래프 정규성은 좌-정규(Left-regular), 우-정규(Right-regular), 정규(Regular), 비정규(Irregular), 약 좌-정규(LLR: Loosely left-regular) 그리고 약 우-정규(LRR: Loosely right-regular)가 있다^[9]. 좌-정규란 그래프 상의 모든 좌측 노드들의 간선 수가 동일한 것이며, 반대로 우-정규는 모든 우측 노드들의 간선 수가 동일한 것을 말한다. 완화된 우-정규(좌-정규)는 모든 좌측(우측) 노드들의 간선 수가 동일하거나 1개 차이 나는 경우를 말한다. 만약 그래프가 좌-정규이면서 우-정규이면 정규 그래프라고 한다. 이 외의 그래프를 모두 비정규 그래프라고 한다.

2. LDPC 부호의 유형

대표적인 LDPC 부호의 유형으로는 Systematic 유형, Gallager 유형 그리고 IRA(Irregular Repeat Accumulate) 유형 등이 있다^[10]. Systematic 유형은 그림 1과 같이 좌측 노드들은 각각 대응되는 하나의 데이터 블록을 가지고 우측 노드들은 각각 하나의 코딩 블록을 가진다. 이때 각각의 우측 노드들은 해당 노드에 연결된 모든 좌측 노드들이 가지고 있는 데이터 블록들의 exclusive-OR연산을 통해서 코딩 블록을 생성한다. Gallager 유형은 n+m개 좌측 노드와 m개 우측 노드를 가지며, 좌측 노드와 우측 노드가 간선으로 연결된다. 또한 좌측 노드들은 각각 대응되는 하나의 데이터 블록

또는 코딩 블록을 가진다. 이때 각각의 우측 노드들에 연결된 모든 좌측 노드들의 exclusive-OR연산 결과가 0인 특징을 가진다. IRA 유형은 n 개 좌측 노드와 m 개 우측 노드를 가지며 추가로 m 개 누적(accumulator) 노드를 더 가진다. 그리고 좌측 노드와 우측 노드 그리고 누적 노드가 간선으로 연결되며 인접한 누적 노드들도 간선으로 연결된다. 이때 블록은 좌측 노드와 누적 노드만이 가지며, 우측 노드는 단순히 부호화와 복호화의 계산에서만 사용되는 특징을 가진다.

3. LDPC 부호 복제 기법

LDPC 부호 복제 기법에 대한 연구는 주로 WAN기반의 분산 파일 시스템에서 연구되었으며, LAN 기반의 클러스터 파일 시스템에서는 아직 활발한 연구가 진행되고 있지는 않다^[11~13]. LDPC 부호 복제 기법에서 사용되는 용어를 정리하면 표 1과 같다.

표 1에서 n 과 m 은 각각 데이터 블록 수와 코딩 블록 수를 의미한다. R 은 부호화율로 $R = n/(n+m)$ 로 정의된다. T 는 LDPC 부호를 정의하는 Tanner 그래프 상의 총 간선 수를 말한다. Λ 와 P 는 LDPC 부호를 정의하는 Tanner 그래프에서 각 좌측 노드와 우측 노드의 간선 수에 대한 확률분포이다. Λ 와 P 는 벡터로 각각 $\Lambda_1, \Lambda_2, \dots, \Lambda_n$ 과 P_1, P_2, \dots, P_m 으로 구성된다. 이때 $\sum \Lambda_i = 1, \sum P_j = 1$ 이다. 그리고 Λ_i 는 그래프에서 좌측 노드의 간선 수가 i 개일 확률이며, $\Lambda_i \times n$ 는 간선 수가 i 개인 좌측 노드의 수를 나타낸다. 이와 마찬가지로 P_j 는 그래프에서 우측 노드의 간선 수가 j 개일 확률이며, $P_j \times m$ 는 간선 수가 j 개인 우측 노드의 수를 나타낸다. n, m, Λ, P 가 결정되면 LDPC 부호를 정의하는 Tanner 그래프를 생성할 수 있다^[10]. 이 생성 방법에 대해서는 III장에서 설명하도록 하겠다. o 는 원본파일 복구를 위해 필

표 1. LDPC 부호 복제 기법에 사용되는 기호 정의
Table 1. Definition of symbols used in the LDPC code replication method.

기 호	정 의
n	데이터 블록 수(the number of data blocks)
m	코딩 블록 수(the number of coding blocks)
R	부호화율(code rate), $R = n/(n+m)$
T	총 간선 수
Λ	좌측 노드 간선 수에 대한 확률분포
P	우측 노드 간선 수에 대한 확률분포
o	평균 다운로드 블록 수(overhead)
f	오버헤드 요인(overhead factor), $f = o/n$

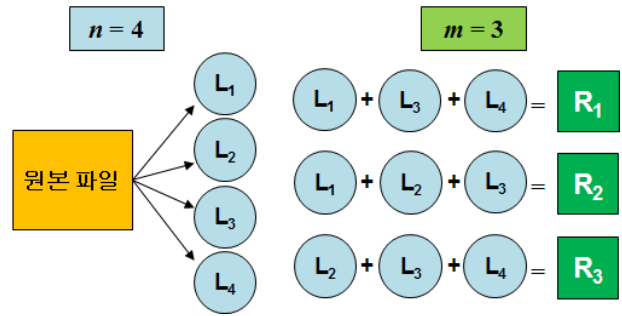


그림 2. (4,3) LDPC 부호의 부호화 과정
Fig. 2. Encoding processes of (4, 3) LDPC code.

요한 평균 다운로드 블록 수를 말한다. 그리고 f 는 오버헤드 요인으로 $f = o/n$ 으로 정의된다. LDPC 부호는 준최적(near optimal)으로 항상 n 개 이상의 블록을 다운로드 해야 하기 때문에 $o \geq n$ 이다. 따라서 f 는 항상 1과 같거나 크며, f 가 1에 가까울수록 원본파일을 위해 다운로드 해야 하는 블록수가 작다. 따라서 f 는 LDPC 부호의 성능 평가 척도로 쓰이며 $n \rightarrow \infty$ 이면 $f \approx 1$ 인 특징을 가진다^[9].

그림 2는 (4,3) LDPC 부호의 부호화 과정을 나타낸 것이다. 그림 2에서 먼저 원본 파일을 동일 크기의 4개 데이터 블록으로 나눈다. 다음으로 3개 코딩 블록을 생성하기 위해 Tanner 그래프 상의 연결 정보를 바탕으로 각각의 코딩 블록에 연결된 데이터 블록들을 모두 exclusive-OR 연산한다. 이렇게 생성된 총 7개 블록이 클러스터 파일 시스템에 분산 저장된다. LDPC 부호 복제 기법의 부호화는 간단한 exclusive-OR 연산을 사용하므로 RS 부호에 비해 상대적으로 계산 비용이 크지 않다. 또한 부호화 시간은 Tanner 그래프의 간선 수에 비례하여 선형적으로 증가한다^[11].

III. 클라우드 컴퓨팅에 적합한 LDPC 부호 복제 기법

LDPC 부호 복제 기법의 성능은 데이터 블록 수와 코딩 블록 수, 사용할 LDPC 부호의 유형에 영향을 받는다^[10]. 따라서 본 논문에서는 LDPC 부호 복제 기법의 성능에 미치는 사항들을 고려하여 클라우드 컴퓨팅에 적합한 LDPC 부호 복제 기법을 제안한다.

1. 데이터 블록 수와 코딩 블록 수의 결정

LDPC 부호의 성능은 데이터 블록 수와 코딩 블록 수에 따라 달라진다. 클러스터 파일 시스템은 가용성을

높이기 위해 원본파일을 여러 블록으로 나누어 저장한다. 이때 블록의 수가 증가할수록 가용성은 높아진다. 하지만 블록 수를 무한정 증가시킨다면 메타 데이터의 양이 증가하여 성능이 저하되는 문제점이 발생한다. 따라서 GFS와 HDFS와 같은 대부분의 클러스터 파일 시스템들은 대부분 64MB정도의 블록으로 나누며 LDPC 부호 복제 기법을 적용하기 위해서는 $n, m < 1,000$ 정도의 작은 블록 수를 사용해야 한다^[11].

본 논문에서는 LDPC 부호의 작은 블록 수에 대해서 어떤 값이 클라우드 컴퓨팅에 적합한지를 파일 가용성과 스토리지 오버헤드를 통해 분석한다.

1) 파일 가용성과 스토리지 오버헤드 개요

파일 가용성과 스토리지 오버헤드를 분석에 사용되는 기호를 정의하면 표 2와 같다.

표 2. 파일 가용성 분석에 사용되는 기호 정의
Table 2. Definition of symbols used in the availability analysis.

기 호	정 의
μ	노드 가용성
S	스토리지 오버헤드, $S = (n + m) / n$
R	부호화율, $R = n / (n + m)$
n'	$f \times n$
m'	$(1 - f) \times (n + m)$
A_w	전체-파일 복제 기법의 파일 가용성
A_L	LDPC 부호 복제 기법의 파일 가용성
F	실패율(Failure rate), $F = 1 - A$

표 2에서 S는 스토리지 오버헤드를 의미하며 데이터 블록 수에 대한 총 블록 수의 비율인 $(n + m)/n$ 으로 구한다^[7, 12]. R은 부호화율로 $n / (n + m)$ 으로 구한다. 즉, S와 R은 반비례 관계이므로 부호화율이 크면 스토리지 오버헤드는 작아진다.

파일 가용성은 A로 나타내며, 현재 해당 파일의 접근 가능 정도를 나타낸다. 전체-파일 복제 기법의 파일 가용성은 A_w 로 나타내며, S개의 복제본이 생성된다고 가정하면 A_w 는 식 (1)과 같다.

$$A_w(\mu, S) = \sum_{i=1}^S \binom{S}{i} \mu^i (1 - \mu)^{S-i} \quad (1)$$

이때 S개 복제본 중에서 최소한 하나의 복제본에 접근이 가능하면 원본 파일을 복구 할 수 있으므로 식 (1)이 성립한다.

다음으로 LDPC 부호 복제 기법의 파일 가용성은 A_L 로 나타내며, 데이터 블록 수가 n이고 코딩 블록 수가 m일 때 식 (2)와 같다.

$$A_L(\mu, n', m') = \sum_{i=1}^{n'+m'} \binom{n'+m'}{i} \mu^i (1 - \mu)^{n'+m'-i} \quad (2)$$

LDPC 부호는 원본 파일 복구를 하기 위해서는 전체 (i+m)개 블록 중에서 최소한 n개 블록이상이 필요하므로 식(2)가 성립한다.

실패율(Failure rate) F는 파일 가용성과 반대되는 개념으로 F가 작으면 파일 가용성이 크다는 것을 의미하므로 LDPC 부호 복제 기법의 성능이 좋다.

2) 파일 가용성과 스토리지 오버헤드 분석

데이터 블록 수와 코딩 블록 수가 변화에 따른 LDPC 부호 복제 기법의 파일 가용성과 스토리지 오버헤드를 분석한 결과는 그림 3과 같다. 이때 $\mu = 0.95, f = 1.1, (n, m) \leq 20$ 로 설정하였다.

그림 3에서 S = 3일 때 전체-파일 복제 기법의 실패율은 약 0.000125이다. n = 4 일 때 LDPC(R=1/3, S=3)의 실패율이 가장 낮지만, 전체-파일(S=3)인 전체-파일 복제 기법과 S=3으로 동일하다. 또 n=10 일 때 LDPC(R=1/2, S=2)의 실패율이 가장 낮다. 하지만 전체-파일(S=3)에 비해 S 값은 작고, 다른 LDPC 기법에 비해 R값은 크다. 따라서 그림 3과 같이 LDPC 부호 복제 기법을 사용할 경우 전체-파일 복제 기법보다 낮은 스토리지 오버헤드를 가지면서 동시에 높은 파일 가용성을 가질 수 있는 n과 m이 존재하는 것을 알 수 있다.

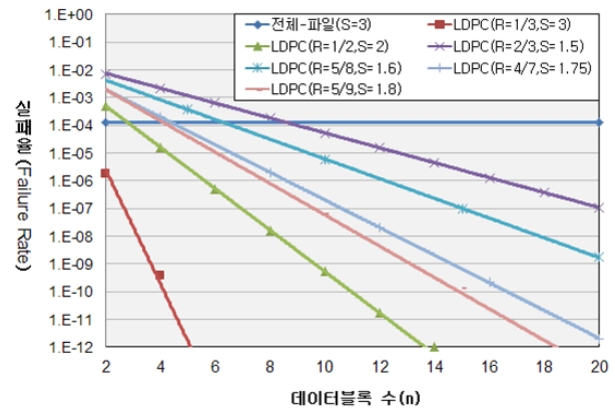


그림 3. 블록 수에 따른 실패율 변화
Fig. 3. Failure rate vs. Number of blocks.

2. LDPC 부호 유형의 성능 비교

데이터 블록 수 변화에 따른 LDPC 부호의 세 가지 유형의 성능을 비교한다. 그림 4은 R = 1/2일 때 데이터 블록 수의 변화에 따른 LDPC 부호의 세 가지 유형의 성능을 비교한 결과이다.

그림 4에서 n < 100 경우, Systematic 유형이 다른 두 유형보다 성능이 좋은 것을 알 수 있다. 하지만 n이 증가하면 Overhead Factor(f)도 함께 증가함을 볼 수 있다. 그림 4에서는 LDPC 부호 복제 기법을 적용할 때는 데이터 블록 수와 코딩 블록 수에 따라 최적의 성능을 낼 수 있는 LDPC 부호 유형을 선택해야 함을 알 수 있다.

따라서 본 논문에서는 클러스터 파일 시스템에 적합한 작은 블록 수(n < 100)에서 성능이 가장 우수한 Systematic 유형을 선택하여 실험을 진행한다.

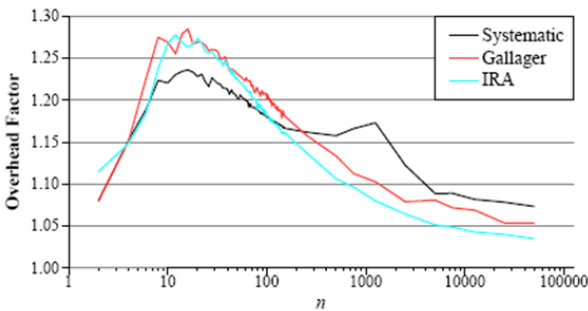


그림 4. 데이터 블록 수의 변화에 따른 LDPC 코드 유형의 성능 비교
Fig. 4. Comparing the three LDPC code types.

3. 그래프 생성

본 논문에서는 작은 블록 수를 가지는 LDPC 부호의 Systematic 유형의 그래프 생성 방법에 대하여 설명한다.

단계 1: n, m, Λ, P 결정

Tanner 그래프는 사용할 부호의 종류와 n, m, Λ, P가 결정되면 생성할 수 있다. 블록 수를 나타내는 (n, m)은 파일 가용성과 스토리지 오버헤드 분석을 통해 결정할 수 있다. 여기서는 예를 들기 위해 각각 n = 4, m = 3, Λ = {0, 0.075, 0.025}, P = {0, 0, 1}로 한다.

단계 2: 노드 간선 수 결정

각 노드의 간선 수 확률 분포인 Λ와 P는 의사-난수

(Pseudo-random number)를 이용하여 무작위로 정의할 수 있으며, 먼저 최소 간선 수와 최대 간선 수를 설정한다. 본 논문에서는 블록의 가용성을 위해 최소 간선 수를 2로 설정하였고, 최대 간선 수는 블록 수와 동일하게 설정한다. 그리고 Λ, P의 노드의 간선 수에 대한 확률값 Λ_i, P_j는 의사-난수를 이용하여 무작위로 정의한다. 단, 아래 조건 1, 2를 만족해야 하며 만족할 때까지 생성 과정을 반복한다.

조건 1: ΣΛ_i = 1, ΣP_j = 1

조건 2: ΣΛ_i × n = ΣP_j × m < n × m

그림 5는 총 간선 수 T = 9일 때 무작위로 각 노드들의 간선 수를 결정한 예이다.

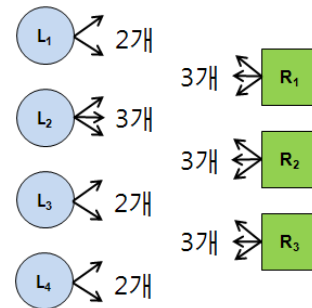


그림 5. 노드의 간선 수 결정 예제
Fig. 5. An example of determining the number of node's edges.

단계3 : 노드 간선 연결 설정

단계 2에서 각 노드들의 간선 수가 결정되면 각 노드들의 간선 연결을 설정한다.

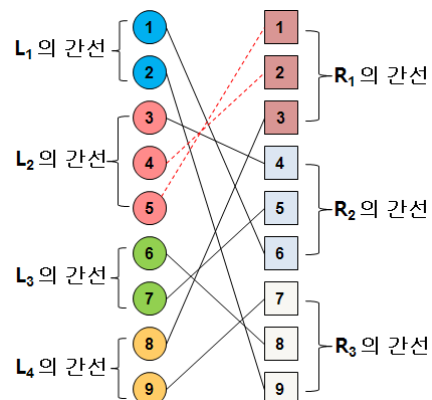


그림 6. 노드의 간선 수 결정 예제
Fig. 6. An example of determining the node's edge connection.

그림 6에서 좌측 노드와 우측 노드 수가 모두 9인 그래프를 생성하고 좌측 노드와 우측 노드를 1:1 무작위로 연결한다. 그리고 각 노드들이 가지는 간선 수를 순서대로 그래프의 간선 연결을 이용하여 설정한다. 예를 들면 첫 번째 좌측 노드의 간선 수가 2이면 첫 번째와 두 번째 좌측 노드의 간선이 모두 첫 번째 좌측 노드의 간선이 된다. 이러한 과정으로 모든 노드들의 간선을 차례대로 반복 설정한다.

단계4: 중복 간선 제거

마지막으로 각 노드 간선이 서로 중복되는지를 검사한다. 그림 6에서 점선으로 표시된 간선 (4-2)와 간선 (5-1)은 모두 두 번째 좌측 노드와 첫 번째 우측 노드를 연결하는 동일한 간선이다. 따라서 이 두 간선 중 하나를 제거하여 중복을 피한다. 중복 간선 제거 과정을 통하여 최종적으로 생성된 그래프는 그림 7과 같다.

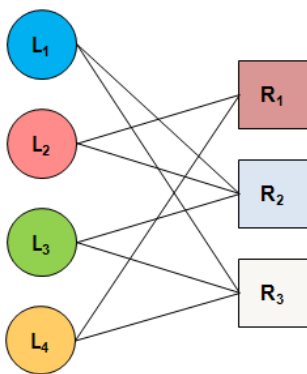


그림 7. 최종 그래프의 예제
Fig. 7. An example of the final graph.

4. 복호화 방법

LDPC 부호 복제 기법의 복호화 과정은 초기 단계, 반복 단계 그리고 종료 단계로 나눌 수 있다.

먼저 초기 단계에서는 n개의 좌측 노드들 중에서 접근 가능한 모든 노드들을 다운로드하여 그래프 상에서 다운로드한 좌측 노드들과 연결되어 있는 간선들을 모두 제거한다. 이때 n개 좌측 노드들을 모두 다운로드한 경우에는 순서대로 연결시켜 원본 파일을 복구할 수 있다. 이때 $f = 1$ 이면 복호화 과정을 종료한다.

반복 단계에서는 간선 수가 1인 우측 노드를 검색하여 해당 노드를 다운로드한다. 그 노드에 연결된 좌측 노드들을 exclusive-OR 연산을 통해 복구한다. 그래프 상에서 다운로드한 우측 노드와 복구한 좌측 노드 그리고

연결되어 있는 간선을 모두 제거한다. 이 단계를 반복 실행한다.

종료 단계에서는 반복 단계에서 더 이상 간선 수가 1인 우측 노드가 없거나 그래프의 모든 간선이 제거된 경우 반복을 종료한다. 이때 그래프의 간선이 모두 제거되지 않은 경우는 복호화에 실패한다. 반대로 모든 간선이 제거된 경우 $f = \text{다운로드한 블록 수}/n$ 이 된다.

IV. 성능 평가

클라우드 컴퓨팅에 적합한 LDPC 부호 복제 기법은 블록 수가 작아야 한다는 것을 파일 가용성과 스토리지 오버헤드 분석 결과로 알 수 있었다. LDPC 부호 유형 중에서 Systematic 유형이 작은 블록 수에서 성능이 우수함을 알 수 있었다.

따라서 본 논문에서는 다양한 정규성을 가지는 그래프들을 무작위로 생성한다. 그리고 생성된 각 그래프들의 성능을 복호화 과정을 통해 평균 오버헤드 요인(f)을 측정하여 성능을 평가한다. 이러한 시뮬레이션 결과를 바탕으로 Systematic 유형의 LDPC 부호 복제 기법에 적합한 그래프 정규성을 분석하고, 생성된 그래프의 총 간선 수가 LDPC 부호의 성능에 미치는 영향을 알아본다. 시뮬레이션에 적용할 파라미터는 표 3과 같다.

시뮬레이션에서는 n과 m에 대해서 각 정규성마다 500개씩의 그래프를 생성하고, 10,000회 복호화 과정을 반복 실행하여 평균 f 를 계산한다. 이때 모든 노드는 동일한 노드 가용성을 가지며, 다양한 실제 시스템 환경을 고려하기 위해 고장 노드 수는 무작위로 결정한다. 또한 복호화 과정에서 블록의 다운로드 순서를 무작위로 결정하고 복호화가 실패한 경우는 제외하고 평

표 3 시뮬레이션 파라미터
Table 3. Simulation parameters.

파라미터	설정 값
데이터 블록 수 범위	$8 \leq n \leq 10$
코딩 블록 수 범위	$5 \leq m \leq 8$
부호 유형	Systematic 유형
각 정규성에 대한 그래프 생성 수	500
복호화 과정 반복 횟수	10,000
노드 가용성	0.95
실패 좌측 노드 수 범위	0 - 3
실패 우측 노드 수 범위	0 - m

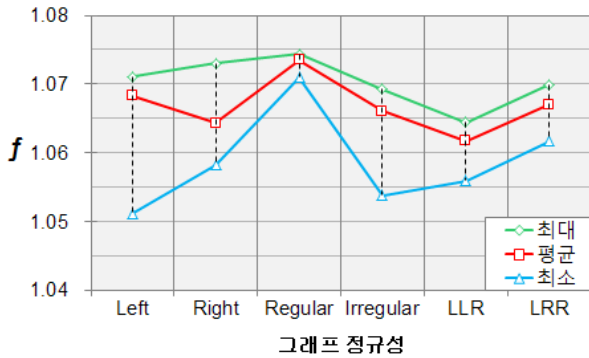


그림 8. 그래프 정규성 vs. 오버헤드 요인(f)
Fig. 8 Graph Regularity Distribution vs. Overhead Factor.

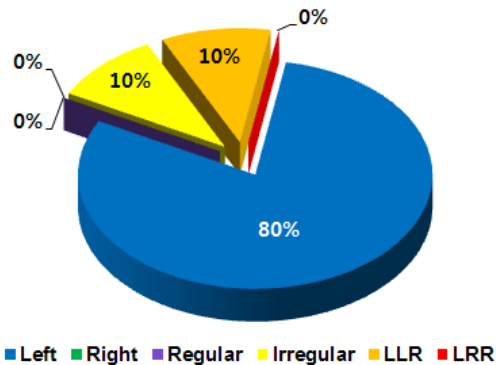


그림 9. (8, 6) LDPC 부호의 베스트 10 그래프의 정규성 분포
Fig. 9 The regularity distribution of a (8, 6) LDPC Code's best 10 graphs.

균 f 를 계산한다.

(n, m)이 (8, 6)일 때 그래프 정규성에 따른 LDPC 부호 복제 기법의 성능은 그림 8과 같다.

그림 8의 최소 f 에서 좌-정규성(Left)의 $f=1.0512$ 로 가장 낮지만 정규(Regular)의 $f=1.0710$ 로 가장 높다. 또한 f 의 편차는 좌-정규성(Left)와 우-정규성(Right)일 때 크지만 정규(Regular) 일때는 작음을 알 수 있다. 따라서 그래프 정규성이 LDPC 부호 복제 기법의 성능에 영향을 미치는 요인임을 알 수 있다.

(n, m)이 (8, 6)일 때 생성한 2,500개의 그래프 중에서 가장 성능이 우수한 그래프 10개의 정규성 분포를 나타낸 것이 그림 9이다.

그림 9에서 총 10개의 베스트 그래프 중에서 좌-정규가 8개로 가장 많으며 약 좌-정규와 비정규 그래프가 각각 1개 그리고 우-정규나 약 우-정규 그리고 정규 그래프는 없는 것을 알 수 있다. 이 결과를 통해서 좌-정규와 약 좌-정규는 다른 정규성에 비해 성능이 우수함

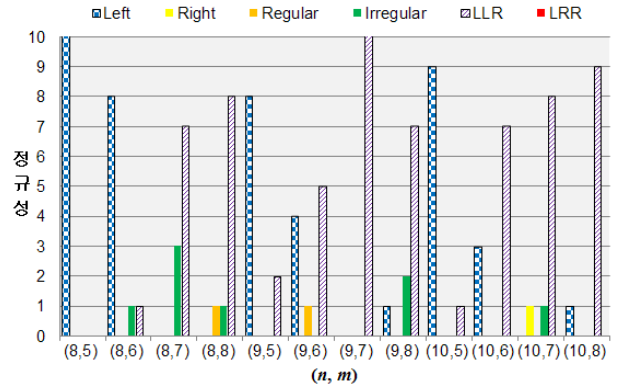


그림 10. 다양한 블록 수에 대한 베스트 10 그래프의 정규성 분포
Fig. 10. The regularity distribution of the best 10 graphs.

을 알 수 있다.

이러한 결과가 다양한 블록 수에 대해서도 동일한 결과를 보이는지 알아보기 위해 다양한 블록 수(n, m)에 대해서 베스트 10 그래프의 정규성 분포를 나타낸 것이 그림 10 이다.

그림 10에서 대부분의 블록 수에서 체크 무늬 막대인 좌-정규(Left)와 사선 무늬 막대의 약 좌-정규(LLR)가 베스트 10 그래프의 대부분을 차지하는 것을 알 수 있다. 따라서 대부분의 블록 수에서도 좌-정규(Left)와 약 좌-정규(LLR)가 다른 정규성에 비해 성능이 우수함을 알 수 있다.

LDPC 부호 복제 기법의 복호화 과정은 그래프의 간선이 모두 제거될 때까지 반복 진행된다. 이때 그래프의 총 간선 수의 변화에 따라 LDPC 부호 복제 기법의 성능이 달라질 수 있다. 따라서 그래프의 총 간선 수의 변화가 LDPC 부호 복제 기법의 성능에 미치는 영향을

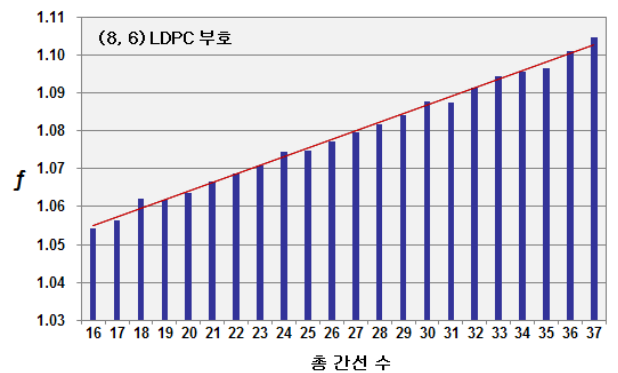


그림 11. (8, 6) LDPC 부호의 총 간선 수에 따른 f 의 변화
Fig. 11. Total number of edges vs. Overhead factor of a (8, 6) LDPC code.

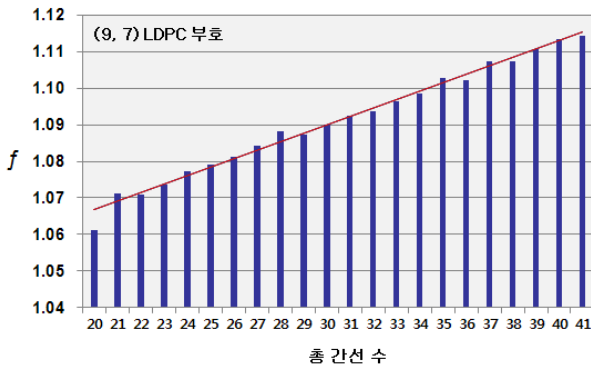


그림 12. (9, 7) LDPC 부호의 총 간선 수에 따른 f의 변화

Fig. 12. Total number of edges vs. Overhead factor of a (9, 7) LDPC code.

표 4 시뮬레이션 파라미터
Table 4. Simulation parameters.

(n, m)	정규성	총 간선 수	생성 가능한 최소 총 간선 수
(8, 5)	좌-정규	16	16
(8, 6)	좌-정규	16	16
(8, 7)	약 좌-정규	18	16
(8, 8)	약 좌-정규	18	16
(9, 5)	좌-정규	18	18
(9, 6)	약 좌-정규	19	18
(9, 7)	약 좌-정규	20	20
(9, 8)	좌-정규	18	18
(10, 5)	좌-정규	20	20
(10, 6)	약 좌-정규	22	20
(10, 7)	약 좌-정규	22	21
(10, 8)	좌-정규	20	20

알아본다.

(n, m)이 (8, 6)일 때 총 간선 수에 따른 LDPC 부호의 성능 변화는 그림 11과 같다.

그림 11을 살펴보면 총 간선 수가 증가함에 따라 f도 선형적으로 증가하는 것을 볼 수 있다. 따라서 총 간선 수가 LDPC 부호의 성능에 영향을 미친다는 것을 알 수 있다.

(n, m)이 (9, 7)일 때 총 간선 수에 따른 LDPC 부호의 성능 변화는 그림 12와 같다.

그림 12에서도 총 간선 수 증가에 따라 f가 선형적으로 증가하는 것을 알 수 있다. 따라서 총 간선 수가 작을수록 LDPC 부호 복제 기법의 성능이 우수함을 알 수 있다.

각 블록 수에 대해 베스트 그래프의 정규성과 총 간선 수 그리고 그래프 생성 단계에서 생성 가능한 최소 총 간선 수를 나타내면 표 4와 같다.

표 4를 보면 모든 블록 수에서 베스트 그래프의 총 간선 수가 생성 가능한 최소 총 간선 수와 동일하거나 근접한 값을 알 수 있다. 따라서 LDPC 부호 복제 기법의 성능은 그래프의 총 간선 수가 최소이거나 근접한 값일 때 가장 좋은 것을 알 수 있다.

V. 결 론

본 논문에서는 클라우드 컴퓨팅에 적합한 LDPC 부호 복제 기법에 기반한 새로운 복제기법을 제안하고 성능을 검증한다. 먼저 파일 가용성과 스토리지 오버헤드 분석을 통해 작은 블록 수가 클라우드 컴퓨팅에 적합하다는 것을 알 수 있었다. 또한 세 가지 LDPC 부호 유형 중에서 Systematic 유형이 작은 블록 수에서 가장 성능이 좋기 때문에 클라우드 컴퓨팅에 적합하다는 것을 알 수 있었다. 그리고 LDPC 부호를 정의하는 Tanner 그래프의 무작위 생성 방법과 LDPC 부호의 복호화 방법을 통하여 클라우드 컴퓨팅에 적합한 LDPC 부호 복제 기법의 그래프 정규성과 총 간선 수에 따른 성능 변화를 분석하였다. 그 결과 LDPC 부호의 그래프 정규성이 좌-정규와 약 좌-정규일 때 LDPC 부호 복제 기법의 성능이 우수함을 검증하였다. 또한 총 간선 수가 최소이거나 근접한 값을 가질 때 LDPC 부호 복제 기법의 성능이 가장 우수함을 검증하였다.

참 고 문 헌

- [1] 이종숙, 박형우, “국내의 클라우드 컴퓨팅 동향 및 전망,” 정보처리학회지, 제16권, 제2호, 17-30쪽, 2009년 3월.
- [2] L. M. Vaquero, L. R. Merino, J. Caceres and M. Lindner, “A Break in the clouds: towards a cloud definition,” ACM SIGCOMM Computer Communication Review, Vol. 39, no. 1, Jan. 2009.
- [3] 민영수, 김홍연, 김영균, “클라우드 컴퓨팅을 위한 분산 파일 시스템 기술,” 한국정보과학회, 정보과학회지, 제27권, 제5호, 86-94쪽, 2009년 5월.
- [4] S. Ghemawat, H. Gobioff and S. T. Leung, “The Google file system,” ACM SIGOPS Operating Systems Review, Vol. 37, no. 5, pp. 29-43, Dec. 2003.
- [5] Hadoop, <http://hadoop.apache.org/>
- [6] 김진수, 김태웅, “OwFS: 대규모 인터넷 서비스를 위한 분산 파일 시스템,” 한국정보과학회지, 제27권, 제5호, 77-85쪽, 2009년 5월.

- [7] W. K.Lin, D. M.Chui, Y. B.Lee, "Erasure code replication revisited," IEEE, in proceedings of the 4th International Conference on Peer-to-Peer Computing, pp. 90-97, Aug. 2004.
- [8] 김성훈, 이문호, "LDPC 부호화를 위한 알고리즘," 대한전자공학회 논문지, 제45권, TC편, 제2호, 1-5 쪽, 2008년 2월.
- [9] M. Luby, M. Mitzenmacher, M. A. Shokrollahi, D. Spielman and V. Stemann, "Small parity-check erasure codes - exploration and observations," IEEE, in proceedings of International Conference on Dependable System and Networks, pp. 326-335, July 2005.
- [10] J. S. Plank and M. G. Thomason, "A Practical analysis of low-density parity-check erasure codes for wide-area storage applications," IEEE, in proceedings of International Conference on Dependable Systems and Networks, pp. 115-124, June 2004.
- [11] B. Gaidioz, B. Koblitz and N. Santos, "Exploring high performance distributed file storage using LDPC codes," Parallel Computing, Vol. 33, no. 4-5, pp. 264-274, Feb. 2007.
- [12] X. Li, C. Xie, Q. Wei and Q. Cao, "A Reliable scheme for cluster storage system," IEEE, in proceedings of 3th International Conference on Semantics, Knowledge and Grid, pp. 394-397, Oct. 2007.
- [13] S. G. Harihara, B. Janakiam, M. G. Chandra, K. G. Aravind, S. Kadhe, P. Balamuralidhar and B. S. Adiga, "SpreadStore: A LDPC erasure code scheme for distributed storage system," IEEE, in proceedings of International Conference on Data Storage and Data Engineering, pp. 154-158, Feb. 2010.

 저 자 소 개



김 세 희(정회원)
2009년 한양대학교
컴퓨터공학과 학사.

2011년 한양대학교
컴퓨터공학과 석사.

현재 SK커뮤니케이션즈
기술연구소 MI팀 연구원

<주관심분야 : 클라우드 컴퓨팅, 클러스터 파일시스템>



이 원 주(정회원)-교신저자

1989년 한양대학교
전자계산학과 학사.

1991년 한양대학교
컴퓨터공학과 석사.

2004년 한양대학교
컴퓨터공학과 박사.

현재 인하공업전문대학 컴퓨터정보공학부
컴퓨터정보과 부교수.

<주관심분야 : 병렬처리시스템, 모바일 컴퓨팅,
Grid 컴퓨팅, 클라우드 컴퓨팅>



전 창 호(평생회원)

1977년 한양대학교
전자공학과 학사.

1982년 Cornell University,
컴퓨터공학과 석사.

1986년 Cornell University,
컴퓨터공학과 박사.

1977년~1979년 전자통신연구소 연구원.

현재 한양대학교 ERICA 캠퍼스 컴퓨터공학과 교수.

<주관심분야 : 병렬처리시스템, 모바일 컴퓨팅,
Grid 컴퓨팅, 클라우드 컴퓨팅>