

차세대 클라우드 저장 시스템을 위한 소실 복구 코딩 기법 동향

김정현, 박진수, 박기현, 남미영, 송홍엽
연세대학교

요약

클라우드 컴퓨팅과 빅데이터 시대의 개막으로 클라우드에 저장되는 데이터가 급속도로 증가함에 따라 최근 클라우드 컴퓨팅의 주요한 요소로 클라우드 저장 시스템이 주목받고 있다.

클라우드 저장 시스템은 크게 두 가지 목적에 의해 동작한다. 첫 번째는 사용자에게 데이터를 소실 없이 정확하게 전달해주는 것이고, 두 번째는 네트워크 상에서 소실된 데이터를 복구해 내는 것이다. 데이터 소실은 분산 노드 내 장비의 결함, 소프트웨어 업데이트 등과 같은 요인에 의해 발생하는데, 이와 같은 데이터 소실에 대응하기 위해 소실 복구 코딩 기법을 사용한다.

본 고에서는 클라우드 저장 시스템의 요구사항들을 토대로 현재 클라우드 저장 시스템에 사용되는 다양한 코딩 기법을 살펴보고 차세대 클라우드 저장 시스템을 위한 코딩 기법에 대해 논의해본다.

씩 증가하고 있다[4]. 또한 Facebook의 조사에 의하면 3000개의 저장 노드 중 매일 약 20개 이상의 노드들이 소실됨이 확인되었다[5]. 따라서 이러한 데이터 소실에 대응하고 저장 공간을 효율적으로 사용할 수 있는 코딩 기법에 대한 연구가 학계와 산업계에서 활발히 진행되고 있다. 특히, Microsoft의 경우 자체적으로 제안한 코딩 기법을 사용하여 데이터 저장 센터 운영 비용을 약 50% 절감했다고 발표하였다[6].

본 고에서는 이러한 시대적 흐름에 따라 클라우드 저장 시스템에서 요구되는 지표들을 소개하고 현재 사용되는 다양한 코딩 기법을 살펴본다. 이를 토대로 차세대 클라우드 저장 시스템을 위한 코딩 기법에 대해 논의해본다. 본 고의 구성은 다음과 같다. 서론에 이어 II장에서 클라우드 저장 시스템 구조와 목적을 설명하고 III장에서는 클라우드 저장 시스템의 요구사항을 살펴본다. IV장에서는 클라우드 저장 시스템을 위한 코딩 기법들을 소개하고 V장에서는 차세대 클라우드 저장 시스템을 위한 코딩 기법들에 대해 논의한다. 그리고 VI장에서 결론을 내린다.

I. 서론

전 세계적으로 클라우드 도입을 고려중인 기업이 90% 이상일 정도로 클라우드가 본격 확산기로 접어들고 있다. 2011년 전 세계 클라우드 트래픽은 전체 트래픽의 39%를 차지했고, 연평균 44%의 성장이 예상되어 2016년에는 비중이 64%에 육박할 것이라는 전망이 나오고 있다. 이렇게 다가오는 클라우드 시대에는 언제 어디서나 인터넷 연결 기기로 데이터에 접근해 쉽게 자료를 공유하고 저렴하게 빅데이터를 저장/처리할 수 있어, 이를 통해 구현되는 새로운 가치들을 통해 서비스, 디바이스, 생활, 기업경영, 시장 등이 변화할 것으로 예상된다[1].

Google, Microsoft, Facebook 그리고 Amazon 등을 필두로 한 클라우드 시장의 급성장으로 데이터 저장 센터에 저장되는 데이터의 양이 Moore's Law 보다 더욱 빠르게 증가하고 있다 [2], [3]. 그 예로, Facebook의 경우 사진 파일을 위한 저장 공간이 2011년 기준 약 20PB(1PBbytes)에서 매주 60TB(1TBbytes)

II. 클라우드 저장 시스템 구조와 목적

클라우드 저장 시스템의 구조는 <그림 1>과 같다. 원본 데이터 소스는 여러 개의 데이터 블록들로 나누어져 코딩을 통해 분산 클라우드 노드에 저장된다. 이때 장비의 결함, 소프트웨어 업데이트 등과 같은 요인에 의해 발생할 수 있는 데이터 소실에

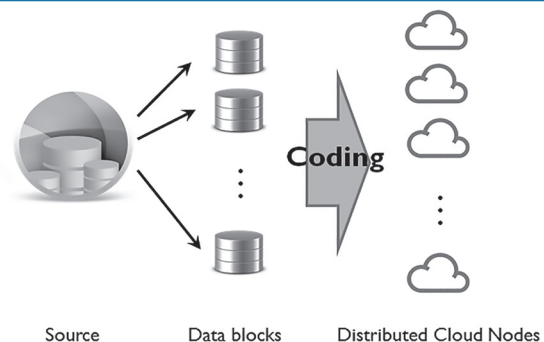


그림 1. 클라우드 저장 시스템의 구조

대비하기 위하여 코딩 기법을 사용한다.

클라우드 저장 시스템은 크게 두 가지 목적에 의해 동작한다. 첫 번째는 사용자에게 데이터를 소실 없이 정확하게 전달해주는 것이고, 두 번째는 네트워크 상에서 소실된 데이터를 복구해 내는 것이다.

1. 데이터 수집

사용자는 분산 노드들로부터 저장된 데이터를 수집하여 복호 과정을 통해 원하는 데이터를 획득한다. <그림 2>는 데이터 수집 과정을 도식화한 것이다. 사용자는 개의 분산 노드들 중 k 개의 노드를 선택하여 a 만큼씩의 데이터를 수집한다. 이 과정을 통해 사용자는 최종적으로 크기가 M 인 데이터를 획득하게 된다. 데이터 수집 과정에서 임의의 분산 노드가 소실되더라도 사용자는 k 개의 노드에서 소실 없이 데이터를 수집하기만 하면 원하는 데이터를 획득할 수 있다.

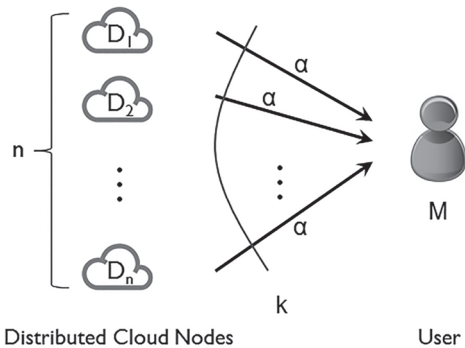


그림 2. 클라우드 저장 시스템의 데이터 수집 과정

2. 데이터 복구

클라우드 저장 시스템에서는 데이터 수집 성공 확률을 보장하기 위하여 주기적으로 소실된 데이터를 복구한다. 이 과정을 데이터 복구 과정이라고 한다. 예를 들어, <그림 3>과 같이 n 개의

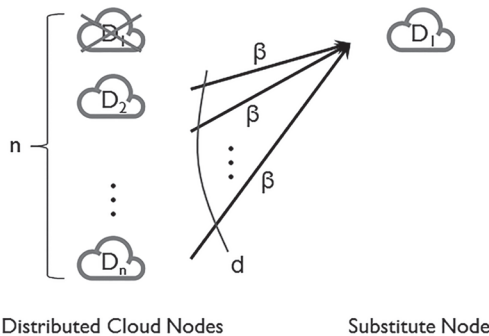


그림 3. 클라우드 저장 시스템의 데이터 복구 과정

분산 노드들 중 첫 번째 노드가 소실되었을 때, $(n-1)$ 개의 노드들 중 d 개를 선택하여 β 만큼씩의 데이터를 다운로드하여 소실된 데이터 D_1 을 복구하여 대체 노드에 저장한다. 데이터 복구 과정에서 임의의 분산 노드가 소실되더라도 대체 노드는 d 개의 기존 노드들에서 소실 없이 데이터를 다운로드하기만 하면 적절한 디코딩 연산을 거쳐서 소실된 데이터를 복구할 수 있다.

Ⅲ. 클라우드 저장 시스템의 요구사항

1. 분산 노드의 저장 공간 용량

분산 노드들은 사용자의 데이터 수집과 소실된 데이터 복구를 위해 일정한 양의 데이터를 저장하게 된다. 이때 각 분산 노드가 최대한 저장할 수 있는 데이터양을 분산 노드의 저장 공간 용량이라고 정의한다. 이 값은 그림 2에서 a 에 해당하며 일반적으로 동일 시스템 내에서 각 분산 노드는 같은 저장 공간 용량을 갖는다고 가정한다. 클라우드 저장 시스템에서는 데이터 수집 성공률과 소실 데이터 복구 성공률을 유지하면서 분산 노드의 저장 공간 용량을 최소화하는 것이 요구된다.

2. 복구 대역폭

노드 소실이 발생하면 대체 노드는 소실된 노드가 저장하고 있던 데이터를 복구하여 저장해야 한다. 이때 남아있는 노드들로부터 데이터를 다운로드하여 소실된 데이터를 복구하게 되며 이 과정에서 다운로드하는 총 데이터 양을 복구 대역폭이라고 정의한다. 이 값은 그림 3에서 복구를 위해 접속하는 노드의 수와 각 접속 노드로부터 다운받는 데이터양의 곱, a 에 해당한다. 복구 대역폭은 일반적으로 작을수록 좋다. 그러나 사용자가 수집하고자 하는 데이터양과 데이터 수집 시 접속하는 노드 수, 그리고 분산 노드의 수가 고정되어 있을 때 앞서 소개한 분산 노드의 저장 공간 용량과 트레이드오프(trade-off) 관계이기 때문에 시스템에 적합한 저장 공간 용량과 복구 대역폭을 갖도록 코딩 기법을 설계하는 것이 중요하다.

3. 복구를 위한 접속 노드 수

일부 시스템에서는 소실 노드를 복구하기 위해 다운로드하는 총 데이터 양보다 접속해야 하는 노드의 수가 더 중요한 지표가 된다. 이러한 소실된 데이터 복구를 위해 접속해야 하는 노드의 수를 로컬리티(locality)[7], 복구 디그리(repair degree)[8],[9] 등으로 정의한다. 최근 앞서 정의한 분산 노드의 저장 공간과

복구 대역폭뿐만 아니라 복구를 위한 접속 노드 수도 함께 줄이는 다양한 코딩 기법들이 활발히 연구되고 있다[10][11].

IV. 클라우드 저장 시스템을 위한 코딩 기법

1. 반복(Repetition) 부호

클라우드 저장 시스템에서 사용되는 코딩 기법 중 가장 간단한 형태인 반복 부호는 데이터를 여러 개의 동일한 복사본을 만들어 여러 분산 노드에 저장시킨다. 예를 들어 원본 데이터에 대해 2개의 복사본을 추가로 생성하여 3개의 분산 노드에 저장하면 임의로 발생하는 2개의 데이터 손실에 대해 복구가 가능하다. 반복 부호는 부호화를 위한 별도의 연산이 없기 때문에 데이터 수집이나 복구 시 부호화를 위한 연산 또한 필요 없는 장점이 있다.

2. 최대 거리 분리가능(Maximum Distance Separable) 부호

최대 거리 분리가능 부호는 싱글톤 한계식(Singleton bound)에서 등호를 만족하는 부호로 정의된다[12]. 이러한 최대 거리 분리가능 부호를 사용하면 최소한의 저장 공간으로 데이터 수집 성공률 혹은 소실 데이터 복구 성공률을 보장할 수 있다. 그러나 클라우드 저장 시스템에서 소실된 데이터를 복구하기 위해 필요한 최소한의 데이터양이 반복 부호보다 일반적으로 많다. 따라서 복구 효율성 측면에서는 반복 부호보다 비효율적이다[13][14].

예를 들어, <그림 4>에서 첫 번째 분산 노드가 소실될 경우 반복 부호의 경우 세 번째 노드로부터 데이터 한 블록을 다운로드하여 소실된 데이터 A를 복구할 수 있다. 최대 거리 분리가능 부호는 두 번째, 세 번째 노드들로부터 데이터 두 블록을 다운로드하여 복호과정을 통해 소실된 데이터 A를 복구할 수 있다. 이때 사용되는 데이터양을 비교해보면 반복 부호가 더 효율적임을 알 수 있다. 그러나 만일 첫 번째, 세 번째 노드가 동시에 소실되면 반복 부호는 소실된 데이터를 복구할 방법이 없는 반면 최대 거리 분리가능 부호는 두 번째, 네 번째 노드들로부터 데이터를 다운로드하여 복호 과정을 통해 소실된 데이터를 모두 복구할 수 있다. 다르게 표현하면, 동일한 복구 성공률을 보장하기 위해서는 반복 부호는 더 많은 저장 공간이 필요하다. 따라서 소실 데이터 복구 성공률 즉, 신뢰도 면에서는 최대 거리 분리가능 부호가 더 뛰어난 것을 알 수 있다.

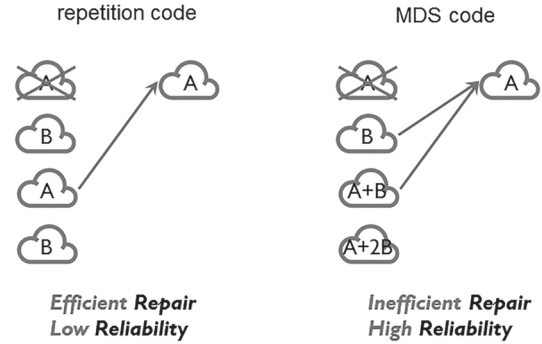


그림 4. 반복 부호와 최대 거리 분리가능 부호

3. 재생(Regenerating) 부호

클라우드 저장 시스템에서는 앞서 설명한 복구 효율성과 신뢰도를 보장할 뿐만 아니라 데이터를 저장하기 위한 저장 공간과 복구 대역폭을 최소화하는 것이 요구된다. 이러한 요구에 따라 Dimakis 교수에 의해 소개된 것이 재생 부호이다[15]. 재생 부호는 데이터 저장 공간 용량과 복구 대역폭 중 어느 쪽에 더 비중을 두느냐에 따라 최소 저장 용량 재생(Minimum Storage Regenerating) 부호와 최소 복구 대역폭 재생(Minimum Bandwidth Regenerating) 부호로 나뉜다[15]. 최소 저장 용량 재생 부호는 식 (1)을 만족하는 부호이고, 최소 복구 대역폭 재생 부호는 식 (2)를 만족하는 부호이다.

$$(\alpha_{MSR}, \gamma_{MSR}) = \left(\frac{M}{k}, \frac{dM}{k(d-k+1)} \right) \quad (1)$$

$$(\alpha_{MBR}, \gamma_{MBR}) = \left(\frac{2dM}{k(2d-k+1)}, \frac{2dM}{k(2d-k+1)} \right) \quad (2)$$

여기서 α 는 저장 공간 용량, γ 는 복구 대역폭, M 은 시스템 용량, k 는 데이터 수집을 위한 접속 노드 수, d 는 데이터 복구를 위한 접속 노드 수이다.

이 둘 부호에 대해 다양한 부호 생성 알고리즘들이 최근까지도 제안되고 있다. 대표적인 부호 생성 방식으로, 최소 저장 용량 재생 부호는 간섭정렬 알고리즘[16]과 Product-Matrix 알고리즘[17]이 있고, 최소 복구 대역폭 재생 부호는 Repair-by-Transfer 알고리즘[18]과 Product-Matrix 알고리즘[17]이 있다.

4. 부분접속 복구(Locally Repairable) 부호

일반적인 최대 거리 분리가능 부호보다 저장 공간을 조금 더 많이 사용하지만 복구를 위해 접속하는 노드 수를 크게 줄일 수 있는 방식이 부분접속 복구 부호이다. 부분접속 복구 부호는 그

그룹별 지역 패리티를 추가함으로써 소실된 데이터가 속한 그룹의 데이터들만 다운로드하여 소실된 데이터를 복구할 수 있다는 장점이 있다. 예를 들어, <그림 5>에서 첫 번째 노드가 소실되면 같은 그룹의 노드들로부터 d_2, d_3, d_4, p_3 를 다운로드하여 소실된 데이터 d_1 를 복구할 수 있다. 이러한 장점으로 인하여 Facebook과 Microsoft 등에서 부분접속 복구 부호에 대해 활발히 연구하고 있다[5][6].

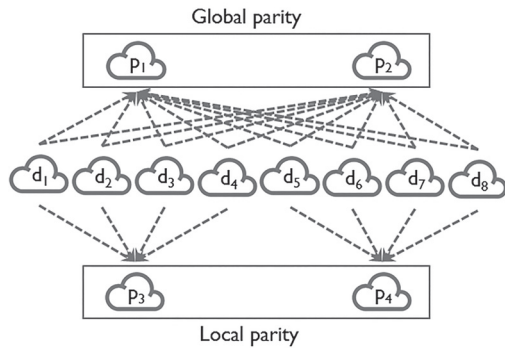


그림 5. 부분접속 복구 부호

V. 차세대 클라우드 저장 시스템을 위한 코딩 기법

본 절에서는 차세대 클라우드 저장 시스템에서 고려되어야 할 네 가지 이슈(무선 전송 채널, 협력적 데이터 복구, 보안 능력, 낮은 복잡도)에 대해 소개하고 차세대 클라우드 저장 시스템이 나아가야 할 방향에 대해 예상해본다.

1. 무선 전송 채널

기존의 클라우드 저장 시스템을 위한 코딩 기법들은 주로 유

선 환경에서 연구되었다. 그러나 최근 효과적인 모바일 클라우드 컴퓨팅 지원을 위하여 무선 환경에서 적용 가능한 기법들이 발표되고 있다. [19]에서는 무선 환경에서는 각 분산 노드로부터 저장되어 있는 데이터 전체를 다운받는 것보다 채널 상황에 따라 일부를 좀 더 신뢰도 있는 상태로 받는 것이 효율적이라는 결과가 발표되었다. 또한 [20]에서는 무선 이중 네트워크 환경에서 사용자가 분산 저장노드로부터 수집할 수 있는 최대 정보량인, 클라우드 저장 시스템의 용량이 분석되었다. 향후, 무선 채널을 통한 데이터 복구에 적합한 코딩 기법의 연구가 더욱 활발해질 것이다.

2. 협력적 데이터 복구

기존의 클라우드 저장 시스템을 위한 코딩 기법은 노드 소실 시 각 소실된 데이터에 대해 개별적으로 복구를 수행하였다. 그러나 만약 노드 간 협력을 통해 데이터 복구할 경우 복구 대역폭을 크게 줄일 수 있다[21]. <그림 6>에서 개별적 데이터 복구는 복구 대역폭이 4이고 협력적 데이터 복구는 복구 대역폭이 3으로 25%의 절감 효과가 있음을 확인할 수 있다. 또한 이때 사용되는 부호의 저장 공간이나 접속 노드수가 유지된다는 장점이 있다. 이와 같이 노드 간 협력을 통하여 복잡도의 증가 없이 복구대역폭을 줄이는 코딩 기법에 대한 연구가 중요해질 전망이다.

3. 보안 능력

클라우드 저장 시스템에서 사용자가 데이터를 수집하는 과정 또는 소실 데이터를 복구하는 과정에서 악의적인 접속자가 전송되는 데이터를 임의로 변경시키는 경우에도 데이터는 신뢰성을 유지해야한다[22][23][24]. 따라서 단순한 데이터 소실뿐만 아니라 데이터 오류 및 악의적 변조에 대해서도 복구 능력을 갖춘 코딩 기법이 설계되어야 한다.

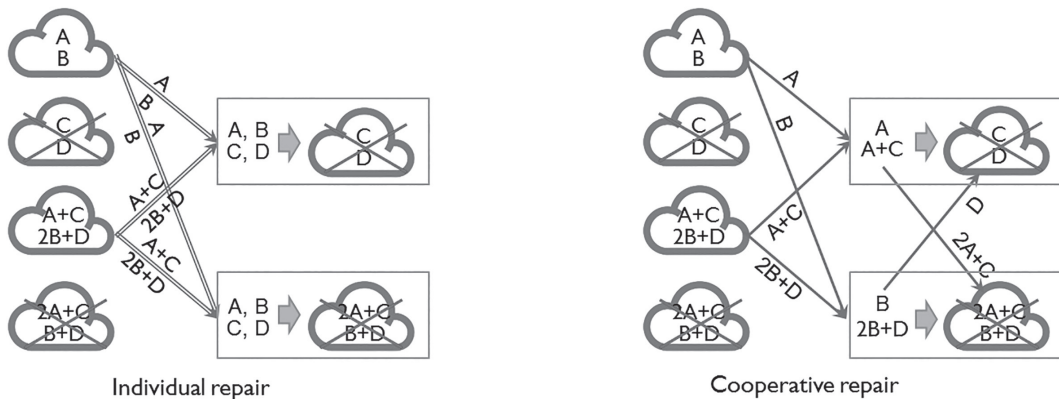


그림 6. 개별적 데이터 복구와 협력적 데이터 복구

4. 낮은 복잡도

클라우드 저장 시스템은 사용하는 코딩 기법에 따라 다른 복잡도를 갖는다. 이러한 복잡도는 데이터를 얻기까지 걸리는 대기 시간(latency)에 영향을 준다. 예를 들어 최대 거리 분리가 가능 부호의 경우 반복 부호보다 데이터 복호 시 큰 대기시간을 필요하다. 따라서 부호화 및 복호화 복잡도를 최소화하기 위한 다양한 코딩 기법들이 연구되고 있다. 기존의 많은 연구 결과들은 데이터 수집과 데이터 복구 과정에서 복잡한 비이진 연산을 필요로 한다. 그러나 최근 제안된 [25]와 [26]에서는 데이터 복구 시 복호과정 없이 데이터 다운로드만으로 복구가 가능한 기법을 제안하였고, [27]에서는 간단한 Xor 연산만으로 데이터 복구를 수행하는 기법을 제안하였다. 또한 [28]에서는 이진 연산만을 통해 데이터 수집과 복구 과정을 수행하는 기법을 제안하였다. 이처럼 향후 새로 제안되는 코딩 기법은 복잡도 면에서도 우수한 성능을 가져야 할 것이다.

VI. 결론

본 고에서는 현재 클라우드 저장 시스템에 사용되는 다양한 코딩 기법을 살펴보고 차세대 클라우드 저장 시스템을 위한 코딩 기법에 대해 논의해보았다. 향후 더욱 효과적인 모바일 클라우드 컴퓨팅을 제공하기 위해서는 유선뿐만 아니라 무선 전송 채널에 적합한 클라우드 저장 기법에 대한 연구가 필요하다. 또한 안정적인 서비스 제공을 위해 악의적인 접속자의 데이터 변조에도 강인한 기법에 대한 연구도 필수적이다. 이와 더불어 노드 간 협력을 통해 복구 대역폭을 절감하고 낮은 복잡도를 갖는 기법의 적용으로 대기시간을 절감하여 더욱 효율성을 높이는 노력도 차세대 클라우드 저장 시스템에서 필히 다뤄져야 할 이슈이다.

Acknowledgement

이 논문은 2013년도 정부(교육부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임(No.2013R1A1A2062061).

참고 문헌

[1] 삼성경제연구소, “클라우드 혁명이 바꾸는 미래,” SERI경영노트, 제181호, 2013. 3. 28.

- [2] World data more than doubling every two years — Driving big data opportunity, new IT roles, <http://www.emc.com/about/news/press/2011/20110628-01.htm>, 2013.
- [3] IDC says world's storage is breaking Moore's law, more than doubling every two years, <http://enterprise.media.seagate.com/2011/06/inside-it-storage/idc-says-worlds-storage-is-breaking-moores-law-more-than-doubling-every-two-years/>, 2012.
- [4] D. Beaver, S. Kumar, H. C. Li, J. Sobel, and P. Vajgel, “Finding a needle in Haystack: Facebook's photo storage,” in Proc. of 9th USENIX Conference on Operating Systems Design and Implementation (OSDI), Oct. 2010.
- [5] M. Sathiamoorthy, M. Asteris, D. Papailiopoulos, A. G. Dimakis, R. Vadali, S. Chen, D. Borthakur, “XORing Elephants: Novel Erasure Codes for Big Data,” in Proc. of the 39th International Conf. on Very Large Data Bases, Aug. 2013.
- [6] C. Huang, H. Simitci, Y. Xu, A. Ogus, B. Calder, P. Gopalan, J. Li, and S. Yekhanin, “Erasure Coding in Windows Azure Storage,” in Proc. of 2012 USENIX Annual Technical Conference, June 2012.
- [7] P. Gopalan, C. Huang, H. Simitci, S. Yekhanin, “On the locality of codewords symbols”, Electronic Colloquium on Computational Complexity (ECCC) vol. 18, 2011.
- [8] G. M. Kamath, N. Prakash, V. Lalitha, and P. V. Kumar, “Codes with local regeneration,” in Proc. of Information Theory and Applications Workshop (ITA), San Diego, USA, Feb. 2013.
- [9] L. P. Juarez, D. L. Henk, and F. Oggier, “locally repairable codes with multiple repair alternatives,” in Proc. of International Symposium on Information Theory (ISIT), July 2013.
- [10] N. Silberstein, A. S. Rawat, O. O. Koyluoglu, and S. Vishwanath, “Locally-Repairable Codes via Rank-Metric Codes,” in Proc. of International Symposium on Information Theory (ISIT), July 2013.
- [11] G. M. Kamath, N. Silberstein, N. Prakash, A. S. Rawat, V. Lalitha, O. O. Koyluoglu, P. V. Kumar,

- and S. Vishwanath, "Explicit MBR All-Symbol Locality Codes," in Proc. of International Symposium on Information Theory (ISIT), July 2013.
- [12] T. Richardson and R. Urbanke, *Modern Coding Theory*, Cambridge Univ. Press, 2008.
- [13] Jung-Hyun Kim, Jin Soo Park, Ki-Hyeon Park, Inseon Kim, Mi-Young Nam, and Hong-Yeop Song, "Reliability Comparison of Various Regenerating Codes for Cloud Services," in Proc. of ICTC2013, Jeju Island, Korea, Oct. 2013.
- [14] Jung-Hyun Kim, and Hong-Yeop Song, "Coding Techniques for Distributed Storage Systems," in Proc. of 2013 CITS 3rd CITW, Samsung Electronics, Seocho, Seoul, Korea, Oct. 2013.
- [15] A. G. Dimakis, P. B. Godfrey, Y. Wu, M. J. Wainwright, and K. Ramchandran, "Network Coding for Distributed Storage Systems," *IEEE Trans. on Inf. Theory*, vol. 56, no. 9, pp. 4539–4551, Sept. 2010.
- [16] N. B. Shah, K. V. Rashmi, P. V. Kumar, and K. Ramchandran, "Interference Alignment in Regenerating Codes for Distributed Storage: Necessity and Code Constructions," *IEEE Trans. Inf. Theory*, vol. 58, no. 4, pp. 2134–2158, April 2012.
- [17] K. V. Rashmi, N. B. Shah, and P. V. Kumar, "Optimal exact-regenerating codes for the MSR and MBR points via a product-matrix construction," *IEEE Trans. Inf. Theory*, vol. 57, no. 8, pp. 5227–5239, Aug. 2011.
- [18] N. B. Shah, K. V. Rashmi, P. V. Kumar, and K. Ramchandran, "Distributed Storage Codes With Repair-by-Transfer and Nonachievability of Interior Points on the Storage-Bandwidth Tradeoff," *IEEE Trans. Inf. Theory*, vol. 58, no. 3, pp. 1837–1852, March 2012.
- [19] C. Gong and X. Wang, "On partial downloading for wireless distributed storage networks," *IEEE Transactions on Signal Processing*, vol. 60, no. 6, pp. 3278–3288, 2012.
- [20] T. Ernvall, S. E. Rouayheb, C. Hollanti, and H. V. Poor, "Capacity and security of heterogeneous distributed storage systems," in Proc. of International Symposium on Information Theory (ISIT), July 2013.
- [21] Kenneth W. Shum, and Yuchong Hu, "Cooperative Regenerating Codes," *IEEE Trans. Inf. Theory*, vol. 59, no. 11, pp. 7229–7258, Nov. 2013.
- [22] S. Pawar, S. E. Rouayheb, and K. Ramchandran, "Securing dynamic distributed storage systems against eavesdropping and adversarial attacks," *IEEE Transactions on Information Theory*, vol. 57, no. 10, pp. 6734–6753, 2011.
- [23] A. S. Rawat, O. O. Koyluoglu, N. Silberstein, and S. Vishwanath, "Secure locally repairable codes for distributed storage systems," in Proc. of International Symposium on Information Theory (ISIT), July 2013.
- [24] S. Goparaju, S. E. Rouayheb, R. Calderbank, and H. V. Poor, "Data secrecy in distributed storage systems under exact repair," in Proc. of International Symposium on Network Coding (NetCod), June 2013.
- [25] S. Pawar, N. Noorshams, S. El Rouayheb, and K. Ramchandran, "Dress codes for the storage cloud: Simple randomized constructions," in Proc. of International Symposium on Information Theory (ISIT), July 2011.
- [26] K. W. Shum, and Y. Hu, "Functional-Repair-by-Transfer Regenerating Codes," in Proc. of International Symposium on Information Theory (ISIT), Cambridge, MA, July 2012.
- [27] D. Papailiopoulos, J. Luo, A. Dimakis, C. Huang, and J. Li, "Simple regenerating codes: Network coding for cloud storage," in Proc. of International Conference Computing and Communications (INFOCOM), March 2012.
- [28] H. Houyix, K. W. Shum, M. Chen and H. Li, "BASIC regenerating code: Binary addition and shift for exact repair," in Proc. of International Symposium on Information Theory (ISIT), July 2013.

약 력



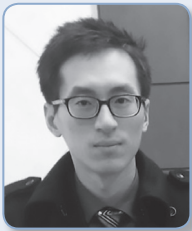
김 정 현

2006년 연세대학교 전기전자공학과 학사
 2008년 연세대학교 전기전자공학과 석사
 2010년~2013년 한국전자통신연구원 연구원
 2013년~현재 연세대학교 전기전자공학과
 박사과정
 관심분야: 분산저장시스템, 정보이론, 부호이론,
 OFDMA 시스템



박 진 수

2009년 연세대학교 전기전자공학과 학사
 2009년~현재 연세대학교 전기전자공학과 석박사
 통합과정
 관심분야: 분산저장시스템, Massive MIMO 시스템,
 부호이론



박 기 현

2007년 연세대학교 전기전자공학과 학사
 2009년 연세대학교 전기전자공학과 석사
 2009년~현재 연세대학교 전기전자공학과 박사과
 정
 관심분야: 통신공학, 정보이론, 암호이론, 이산수학



남 미 영

2005년 연세대학교 전기전자공학과 학사
 2005년~2007년 삼성전자 연구원
 2009년 연세대학교 전기전자공학과 석사
 2009년~현재 연세대학교 전기전자공학과
 박사과정
 관심분야: 오류 정정 부호



송 흥 엽

1984년 연세대학교 전자공학과 학사
 1986년 미국 USC 대학교 전자공학과 석사
 1991년 미국 USC 대학교 전자공학과 박사
 1992년~1994년 미국 USC Communication
 Science Institute, 박사 후 연구원
 1994년~1996년 미국 Qualcomm Inc., San
 Diego, USA, 선임연구원
 1995년~현재 연세대학교 전기전자공학과 교수
 관심분야: 통신공학, 정보이론, 부호이론, 암호이론,
 이산수학