Hyperledger Fabric과 Erasure Coding의 자원 활용율 분석

유정현[1], 변홍수[1], 이명철[2], 최진춘[2], 김영재[1]
[1]서강대학교 컴퓨터공학과, [2]한국전자통신연구원 스마트데이터연구실
{jhryu, byhs, youkim}@sogang.ac.kr, {mclee, jcchoi}@etri.re.kr

Resource Utilization Analysis of Hyperledger Fabric with Erasure Coding

Junghyun Ryu[1], Hongsu Byun[1], Myungcheol Lee[2], Jinchun Choi[2], Youngjae Kim[1]
[1]Department of Computer Science and Engineering, Sogang University, Seoul, Republic of Korea
[2]Smart Data Research Section, ETRI, Daejeon, Republic of Korea

요 약

하이퍼레저 패브릭은 리눅스 파운데이션에서 개발한 대표적인 허가형 블록체인이다. 하이퍼레저 패브릭은 참여하는 모든 노드들이 모든 트랜잭션에 대한 데이터를 중복 저장해야 하므로 저장 공간을 크게 차지하는 문제가 있다. 하이퍼레저 패브릭에서 이러한 저장 공간 문제를 해결하기 위해 소거 코드 (Erasure Coding) 기반의 E-Fabric이 제안되었다. E-Fabric은 소거 코드 중 가장 대표적인 Reed-Solomon (RS) 코드를 적용하였으며, EC는 행렬곱 연산을 수행하기 때문에 peer의 CPU 사용율을 증가시킨다. 이에 본 연구에서는 E-Fabric에서 트랜잭션 처리와 EC를 동기적으로 수행할 때, EC로 인한 저장 효율성 증대와 리소스 소모 오버헤드 간의 trade-off에 대해 분석하였다. 그 결과 모든 실험 환경에서 (20, 14) EC 수행으로 인해 저장 데이터의 양은 93% 감소하였고, peer의 트랜잭션 처리 시간 중 평균 CPU 사용량은 최대 1.16배까지 증가하였으며, 트랜잭션 처리 시간은 1.07배 까지 증가하였다. 따라서 EC 도입을 통해 낮은 자원사용율 오버헤드와 낮은 트랜잭션 처리시간 오버헤드로 높은 저장공간 효율 증가를 기대할 수 있다.

1. 서론

하이퍼레저 패브릭 [1]은 리눅스 파운데이션에서 개발한 대표적인 허가형 블록체인이다. 허가형 블록체인은 허가받은 사용자만이 블록체인 네트워크에 참여할 수 있으므로 악의적인 공격의 위험을 줄일 수 있는 장점을 갖으며 금융, 의료와 같이 데이터 보안이 중요한 산업에서 활용된다. 하지만 하이퍼레저 패브릭은 참여하는 모든 노드들이 모든 트랜잭션 데이터를 중복 저장해야 하므로 저장 공간을 많이 차지하는 문제가 있다. 하이퍼레저 패브릭에서 이러한 저장 공간 문제를 해결하기 위해 소거 코드 (Erasure Coding) 기반의 E-Fabric [2]이 제안되었다.

Erasure Coding (EC)은 데이터에 패리티를 추가하고 여러 개의 분할된 청크로 인코딩하여, 데이터 저장 효율은 높이고 일부가 손실되어도 원본 데이터를 복구할 수 있는 기술이다. E-Fabric은 구성요소인 각 peer에서 일정 크기만큼 쌓인 블록들을 EC 중 대표적인 Reed-Solomon (RS) 코드 [3]로 인코딩하여 해당 peer에서 필요한 청크만 저장하므로써 전체 peer들의 저장 공간 효율을 높인다. 하지만, EC는 행렬곱 연산을 수행하기 때문에 각 peer의 CPU 사용율을 증가시킨다. 또한 저장된 블록들을 다시 읽은 뒤 인코딩을 수행하므로 디스크 I/O를 발생시키기 때문에 전체 트랜잭션 처리 시간에 영향을 끼친다. 이는 하이퍼레저 패브릭 네트워크 유지 및 체인코드 구동에 필요한 CPU 싸이클 점유 경쟁을 유발하며, 블록체인 시스템의 성능을 저하시킬 수 있다. 따라서 EC를 적용 시 저장 효율과 자원 사용율에 따른 오버헤드 간 trade-off의 면밀한 분석이 선행되어야 한다. 그럼에도 불구하고 E-Fabric은 EC에 따른 하이퍼레저 패브릭의 저장 공간 효율 향상만을 평가하였고 EC 수행으로 인한 트랜잭션 처리 시간 증가 및 자원 사용율 증가와 같은 오버헤드 분석이 부족하다.

이 논문은 2021년도 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원의 지원을 받아 수행된 연구임 (No.2021-0-00136, 다양한 산업 분야 활용성 증대를 위한 대규모/대용량 블록체인 데이터 고확장성 분산 저장 기술 개발)

본 연구에서는 E-Fabric에서 트랜잭션 처리와 EC를 동기적으로 수행할 때, EC로 인한 자원 사용율 오버헤드를 분석한다. 본 연구의 실험 환경에서 블록 생성율을 최대로 높였을 때, EC 수행 시 저장 공간은 93% 감소한 반면, CPU 사용량과 트랜잭션 처리 시간은 각각 최대 1.16배, 1.06배 증가함을 관찰하였고, 하이퍼레저 패브릭에서 EC 오버헤드가 예상보다 낮음을 밝혔다. 또한 이러한 동기적인 EC로 인한 오버헤드는 트랜잭션 처리와 EC를 비동기적으로 개선한다면 더욱 감소할 것으로 기대된다.

2. Erasure Coding 기반 하이퍼레저 패브릭

저장된 데이터의 양이 기하급수적으로 증가함에 따라, 저장 용량을 보다 효율적으로 사용하기 위한 방법으로 Erasure Coding가 채택 되고 있다. Erasure coding은 Reed-Solomon (RS) 코드 [3]가 대표적으로 사용되며 RS 코드는 N과 k 두개의 매개변수로 구성된다.

RS 코드는 원본 데이터를 k개로 chunking하고, $N-k$개의 패래티 블록을 추가하여 N개의 청크 블록으로 인코딩한다. 인코딩 된 N개의 청크 중 최대 $N-k$개 청크가 손실되더라도 원본 데이터를 디코딩하여 복구할 수 있다. 따라서 데이터 복제 방식 [4]에 비해 데이터 저장 용량을 크게 줄일 수 있는 장점이 있다.

그림 1은 Erasure Coding이 적용된 하이퍼레저 패브릭인 E-Fabric의 동작 흐름을 보여준다. ❶ 클라이언트가 endorsing peer들에게 트랜잭션 발생을 제안한다. ❷ *endorsing*: 제안 받은 트랜잭션을 endorsing peer에서 시뮬레이션을 통해 검증한다. ❸ 검증이 성공적으로 종료되면 트랜잭션 보증에 대한 서명을 클라이언트에게 전송한다. ❹ 클라이언트는 일정 개수 이상의 endorsing Peer에서에게 보증을 받은 트랜잭션을 orderer에게 보낸다. 그림 1 의 예시에서는 3개의 endorsing peer 중 2개 이상의 보증을 받자 마자 orderer에게 전송한다. ❺ orderer는 트랜잭션을 미리 정의된 크기 혹은 timeout동안 집계하여 하나의 블록을 생성

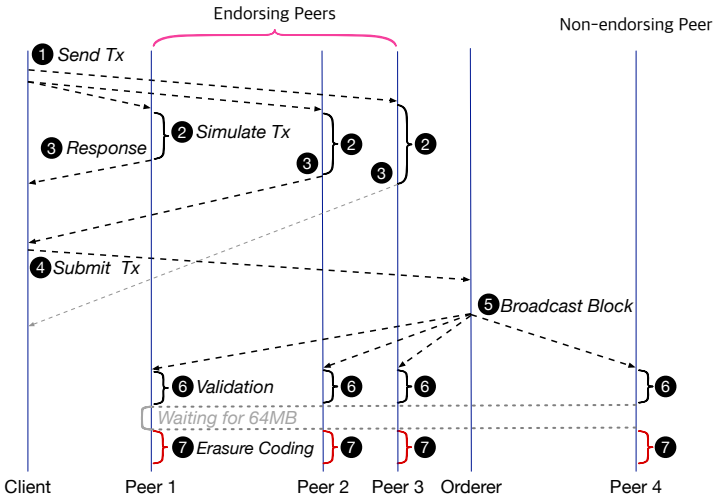


그림 1: E-Fabric [2]의 동작 흐름.

하고 최신 블록을 모든 노드에게 전파한다. ❻ *validation*: 모든 노드들은 새로 생성되어 전달받은 블록을 각각 검증한다. ❼ 각 노드는 블록이 일정 크기만큼 쌓이면 Erasure Coding을 실행하여 필요한 청크만 저장하고 나머지는 삭제한다. 그림 1의 예시에서 각 peer는 블록 들이 모인 블록파일이 64MB 까지 쌓일 때까지 대기한 후, 64MB 블록파일에 대해 (4,3) RS 코드를 수행하여 원본 데이터를 3(k)개의 청크로 분할한 뒤 4(N) - 3(k) = 1개의 패리티를 생성한다. 이때 각 청크의 크기는 원본 데이터의 1 / 3(k)인 33.3%의 크기를 가진다. 사전에 정의된 정책에 의해 생성된 4개의 청크 중 자신에게 필요한 1개의 청크만을 저장하고 나머지 3개의 청크를 삭제한다. 따라서 저장 공간을 66.7% 절감할 수 있으며 각 peer는 원본 데이터가 필요할 경우 주변 피어노드에게 k개의 청크를 요청하여 디코딩 할 수 있다.

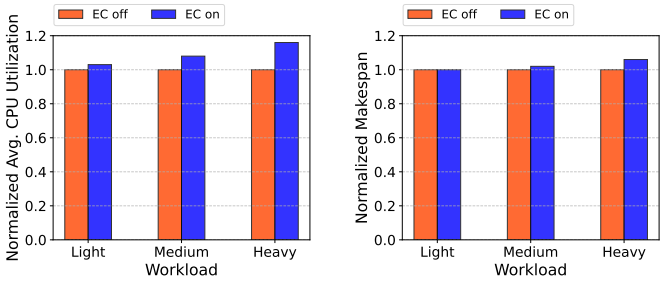
3. 실험 방법론

본 연구에서는 E-Fabric에서 *Erasure Coding* (EC)이 실행될 때 다음과 같은 질문에 대한 답을 찾고자 한다.

1) *단위시간당 블록 생성량이 증가함에 따라 EC 오버헤드는 얼마나 증가하는가?* 같은 단위시간 안에 발생하는 트랜잭션 양이 증가함에 따라 하이퍼레저 패브릭의 peer가 한번에 처리해야 할 블록의 수는 증가한다. 이때 EC 실행이 미치는 영향은 얼마나 되는지 분석이 필요하다.
2) *컴퓨팅 자원이 제한된 상황에서 EC 연산이 하이퍼레저 패브릭 네트워크의 각 노드에 미치는 영향은 어느 정도인가?* 하이퍼레저 패브릭의 각 peer는 그림1의 ❻ 에 해당하는 validation phase에서 *vscc*, *mvcc* 등 다양한 CPU-intensive한 [5] 연산을 수행한다. 이에 EC를 추가로 실행하였을 때 오버헤드가 존재하는지, 존재한다면 기존 연산량에 대한 비율은 얼마나 되는지 분석이 필요하다.

표 1: 실험 서버 상세 설명

CPU	AMD EPYC™ 7352, 48 Cores (96 Threads), 2.3GHz (Up to 3.2GHz) 128 MB L3 Cache
Socket	2 NUMA Node
Memory	256 GB (64 GB × 4) DRAM DDR4 3200MHz
OS	Centos 7.92.2009 (Core) / Linux Kernel 4.14



(a) CPU Utilization (b) Makespan

그림 2: 초당 블록 생성량 증가에 따른 Erasure Coding 실행시 총 실행시간과 평균 CPU 사용량의 변화

본 연구에서는 단위시간당 블록 생성량에 따라 워크로드를 정의하여 블록 생성량 증가에 따른 EC 오버헤드를 분석하고, 사용 가능한 CPU 코어 개수를 차등 제한함으로서 컴퓨팅 자원이 제한된 상황에서의 EC 오버헤드를 분석하며 위 질문에 답을 찾고자 하였다. 또한 EC는 CPU-intensive한 작업이므로, 본 연구에서는 컴퓨팅 자원에서 CPU와 memory 사용량 중, 특히 CPU 사용량에 끼치는 영향을 분석한다.

실험 환경 설정: 본 연구는 24개 코어를 갖는 AMD EPYC™ 7352 CPU 2개와 256 GB DRAM이 장착된, CentOS 7이 구동되는 서버에 하이퍼레져 패브릭 네트워크를 구성하였다. 상세한 스펙은 표 1에 나와있다.

본 연구는 실험에 사용할 하이퍼레저 패브릭 네트워크를 2개의 organization에 각 10개 peer를 할당하여 총 20개의 peer를 구성하였고, 1개의 orderer를 사용하였다. 각 구성요소는 한 호스트 서버 안에서 docker container로 분리되어 있다.

단위시간당 블록 생성량 증가로 인한 EC 오버헤드를 관찰하고자 총 1백만 건 트랜잭션으로 890MB 블록 생성에 대하여, 트랜잭션 발생율에 따라 *light*, *medium*, *heavy*로 워크로드를 나누어 실험하였다. 각 워크로드는 동시에 트랜잭션을 발생시키는 클라이언트 개수에 따라 정의하였으며, 클라이언트 개수가 증가할수록 초당 블록 생성량 증가에 따른 peer의 부하가 증가한다. 워크로드는 표 2와 같이 정의하였다.

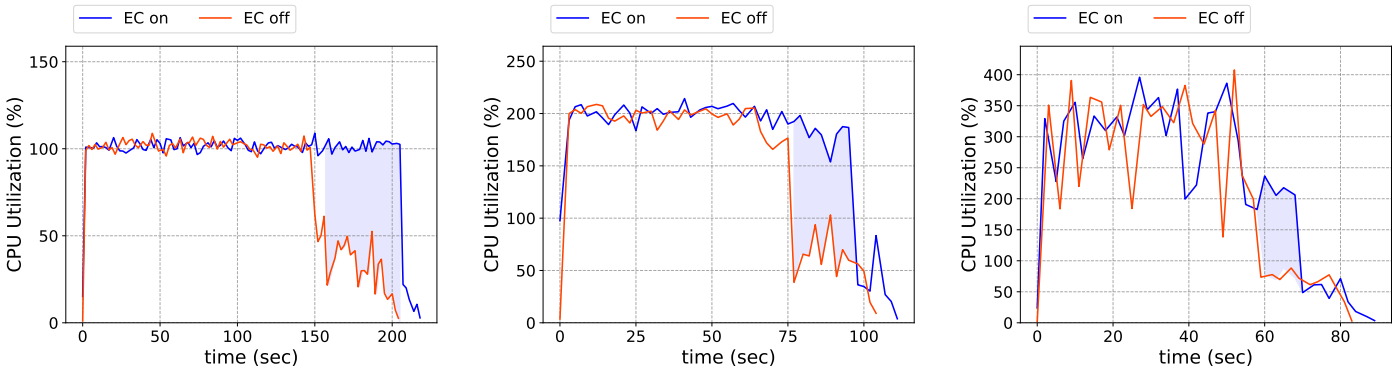
Erasure Coding 설정은 하나의 블록파일을 20개의 peer가 나누어 저장하도록 N=20, k=14 의 RS 코드 연산 수행을 고려했다. 이때 각 청크의 크기는 원본 데이터의 1/14이다.

4. 실험 및 평가

본 연구에서는 non-endorsing peer에서 총 890MB에 달하는 1백만 건 트랜잭션 처리 중 *Erasure Coding* (EC)을 실행하였을

표 2: 워크로드에 대한 상세 설명

Type	클라이언트 (#)	블록 생성율 (CPU 100%)	블록 생성율 (CPU 400%)
Heavy	16 개	7.8MB/s	20MB/s
Medium	8개	6.2MB/s	13MB/s
Light	1개	2.4MB/s	3MB/s



(a) CPU Core Max 100% (b) CPU Core Max 200% (c) CPU Core Max 400%

그림 3: 트랜잭션 처리시간 중 CPU 사용량 비교

때와 실행하지 않았을 때의 평균 CPU 사용량, 총 실행시간 변화를 다음과 같이 분석하였다.

워크로드 변화에 따른 분석: 본 실험에서는 워크로드 별 총 실행시간 및 트랜잭션 처리 중 CPU 사용량을 분석한다. 사용 가능한 CPU 코어 개수를 1개로 고정한 상태에서 표 2에서 정의한 워크로드에 따라 비교분석 하였다.

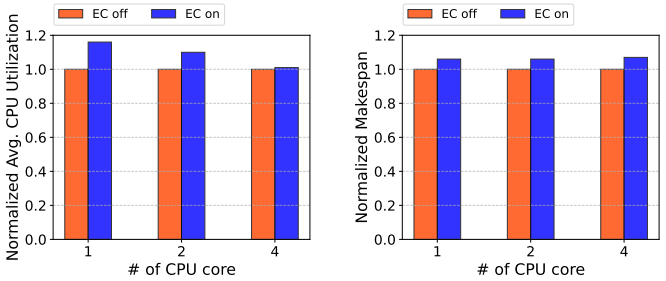
그림 2는 EC를 실행하지 않았을 때의 평균 CPU 사용량과 실행시간을 기준으로 정규화하여 EC를 실행하였을 때 증가한 비율을 나타낸다. 단위시간당 블록 생성량이 증가함에 따라 그림 2(a)에서 보이듯이 EC를 실행하였을 때 평균 CPU 사용량은 1.03배, 1.08배, 1.16배 증가하였다. 그림 2(b)에서 보이듯이 총 실행시간은 1.0배, 1.02배, 1.06배 증가하였다.

사용 가능한 CPU 제한에 따른 분석: 본 실험에서는 CPU 코어 개수 제한에 따라 EC 실행시 트랜잭션 처리 시간 및 CPU 사용량을 분석한다. 표 2에서 정의한 Heavy 워크로드를 사용하여 가용 CPU 코어를 1개, 2개, 4개로 제한하여 비교분석하였다.

그림 3은 사용 가능한 CPU 코어 개수 변화에 따라 트랜잭션 처리 시간 중 CPU 사용량을 분석하였다. EC 실행시 평균 CPU 사용량과 트랜잭션 처리 시간 모두 증가하였다. 또한 EC 실행시 가용 CPU의 절반 이상 사용한 구간이 EC를 수행하지 않았을 때에 비해 CPU 코어 1개일 때 61초, 2개일 때 23초, 4개일 때 15초 증가하였으며, 이는 그림 3의 그래프에 푸른색 음영 부분으로 표시되어 있다.

그림 4는 그림 2과 마찬가지로 EC를 실행하지 않았을 때의 평균 CPU 사용량과 실행시간 으로 정규화하여 EC를 실행하였을 때 증가한 비율을 나타낸다. 사용 가능한 CPU 코어 개수 변화에 따라 평균 CPU 사용량은 1.16배, 1.10배, 1.01배 증가하였으며, 실행시간은 1.06배, 1.06배, 1.07배로 변화하며 유의미한 차이를 보이지 못하였다.

공간 절약 분석: 모든 실험환경에 대해 (20, 14) EC를 수행하였다. 이에 원본 데이터의 1 / 14(k) 크기의 청크 20개를 각 peer가 하나씩 저장하게 된다. 따라서 각 peer가 저장하는 블록파일의 크기가 EC를 수행하지 않았을 때의 7% 로 줄어듦에 따라 93%의 공간을 절약한다.



(a) CPU Utilization (b) Makespan

그림 4: Erasure Coding 실행 시 트랜잭션 처리시간 및 CPU 사용량 증가 비율

5. 결론

하이퍼레저 패브릭과 같은 블록체인은 클라우드 환경, 분산 저장 시스템에 비해 단위시간당 데이터 arrival rate이 낮다. 따라서, EC를 도입하였을 때 CPU 오버헤드를 감수하고 얻는 저장공간의 이득을 고려해볼 만하다. 본 연구에서는 하이퍼레저 패브릭에서 EC를 적용하였을 때 저장 효율성과 CPU 오버헤드 간의 trade-off를 분석한 결과 93%의 저장 공간 효율을 증가시키면서 최대 1.16배의 CPU 사용량이 증가함을 밝혔다. 따라서 하이퍼레저 패브릭과 같은 블록체인 시스템은 클라우드 환경, 분산 저장 시스템에 비하여 데이터 I/O 처리량이 낮으므로 EC 도입을 통해 낮은 자원 사용율 오버헤드로 높은 저장공간 효율 증가를 기대할 수 있다.

참고 문헌

[1] E. Androulaki, A. Barger, V. Bortnikov, C. Cachin, K. Christidis, A. De Caro, D. Enyeart, C. Ferris, G. Laventman, Y. Manevich, *et al.*, "Hyperledger fabric: a distributed operating system for permissioned blockchains," in *Eurosys*, pp. 1–15, 2018.

[2] 박소현, 최병준, 김창수, 이명철, and 이일구, "저장 효율적인 하이퍼레저 패브릭 블록체인을 위한 소거 코드 기반 분산 저장 시스템," *한국통신학회 학술대회논문집*, pp. 555–556, 2022.

[3] I. S. Reed and G. Solomon, "Polynomial codes over certain finite fields," *Journal of the Society for Industrial and Applied Mathematics*, vol. 8, no. 2, pp. 300–304, 1960.

[4] D. Ford, F. Labelle, F. I. Popovici, M. Stokely, V.-A. Truong, L. Barroso, C. Grimes, and S. Quinlan, "Availability in globally distributed storage systems," in *OSDI*, 2010.

[5] H. Javaid, C. Hu, and G. Brebner, "Optimizing validation phase of hyperledger fabric," in *MASCOTS*, 2019.