부호화된 블록체인 블록 파일의 안전한 복구를 위한 동기화 제어 방안 설계

김환웅 [1,2], 최진춘 [2], 이명철 [2]
[1] 충북대학교 정보통신공학부 학부생
[2] 한국전자통신연구원 스마트데이터연구실 연구원

hwkim2966@chungbuk.ac.kr, {jcchoi, mclee}@etri.re.kr

Design of a Synchronous Control Scheme for Safe Recovery of Encoded Blockchain Block Files

Hwanwoong Kim[1,2], Jinchun Choi[2], Myungcheol Lee[2]
[1]School of Information and Communication Engineering, Chungbuk National University
[2]Smart Data Research Section, ETRI

요 약

블록체인은 네트워크 참여 노드 간 합의를 통해 동일한 원장을 분산 저장하여, 원장에 저장된 데이터의 신뢰성을 보장할 수 있다. 하지만 원장 데이터가 모든 노드에 중복 저장되는 문제가 있고, 이를 해결하기 위해 Reed-Solomon 코드를 적용하여 저장 효율성을 높일 수 있다. 이 논문에서는 허가형 블록체인 하이퍼레저 패브릭에서 Reed-Solomon 코드를 적용하여 원장을 저장 저장하는 경우, 분산 저장된 부호화 블록 파일의 비동기적 복구과정으로 인한 문제를 다루고, 동기화 제어 방안을 설계하여 해결 방안을 제시한다.

1. 서론

블록체인의 무결성, 가용성 같은 특징은 금융, 의료 등에서 사용되는 민감한 데이터의 안전한 저장과 관리를 위해 사용될 수 있다[1]. 다만 블록체인 참여 노드가 블록체인의 특징을 위해 같은 내용의 원장을 복제해서 보관하기 때문에 저장 용량이 지속적으로 커지는 문제가 논의되고 있다. 공개형 블록체인인 이더리움의 경우 이미 풀 노드 기준 원장의 크기가 1TB 를 넘어섰다[2]. 이와 같이 블록체인에 저장되는 트랜잭션의 크기가 커지면서, 원본 데이터에 패리티를 추가한 부호화를 통해 높은 저장 효율성을 보이면서 데이터 가용성을 보장하는 소거 코드 (Erasure Code)를 저장에 활용할 수 있다. 박소현 등은 대표적인 소거 코드 방법인 Reed-Solomon 코드 (RS 코드)를 활용하여 노드가 데이터를 분산 저장하는 방식으로 노드들이 비잔틴 장애 내성을 가지면서 저장 공간을 복제 방식 대비 효율적으로 사용할 수 있도록 설계했다[1]. Liang 등은 블록체인에 소거코드를 적용하여 1/2 피어에 대해 장애 내성 보장을 하면서 전체 원장 크기를 53% 감소시킬 수 있었다[2]. 이 논문에서는 허가형 블록체인인 하이퍼레저 패브릭에서 저장 효율적인 관리를 하기 위해 RS 코드를 사용할 때, 여러 노드에 분산 저장된 트랜잭션 파일의 비동기 복구 과정으로 인한 문제를 다루고, 동기화 제어 관리를 통한 해결 방안을 제시한다.

2. 본론

하이퍼레저 패브릭의 피어들은 합의를 통해 원장을 기록한다. 이때 원장 공간의 효율을 위해 블록 또는 블록파일에 RS 코드를 적용하여 인코딩 청크로 분산 저장할 수 있다. 피어 장애를 탐지하기 위해, 피어들은 특정 시점마다 하트비트 신호를 모든 피어에게 전송해 본인이 온라인 상태임을 알린다. 만약 한 피어가 장애 발생으로 하트비트 신호를 보내지 못하면, 나머지 피어들은 해당 피어가 오프라인임을 인지하고, 데이터 가용성 보장을 위해 분산 저장된 원장 조각(인코딩 청크)을 모아 디코딩과 재인코딩을 수행하게 된다.

[1] Frost&Sullivan, TechVision 2024: Blockchain
[2] https://etherscan.io/chartsync/chaindefault

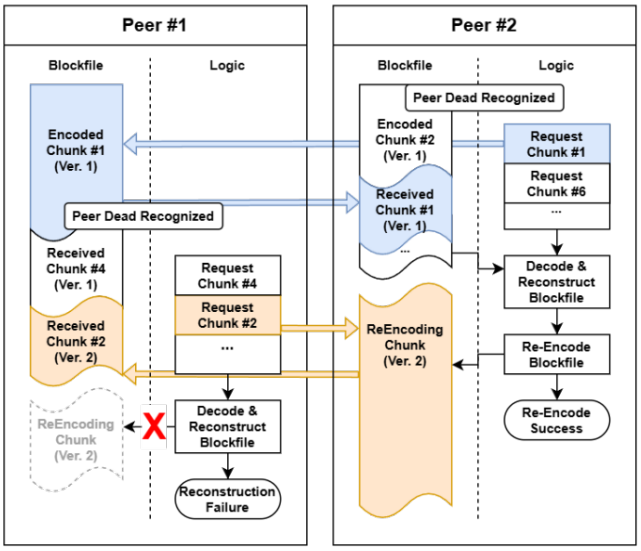

그림 1. 피어 장애 인지 시점의 차이에 따른 블록파일 복구 성공(우) 및 실패(좌) 예시

모든 피어가 재인코딩을 비동기적으로 수행하는 경우, 기존 인코딩 청크가 지워지는 시점은 새로운 인코딩 청크가 만들어지는 시점에 따라 달라진다. 즉, 모든 피어가 동시에 같은 버전의 인코딩 청크를 가질 확률은 매우 드물고, 이로 인해 디코딩이 실패하게 된다.

그림 1 은 피어 장애 인지 시점 차이에 따라 복구가 실패하는 예시이며, 피어 i 는 i 번째 인코딩 청크를 갖고, 피어 2 가 피어 1 보다 피어 장애를 먼저 인식했다고 가정한다. 피어 2 는 피어 1 이 보유한 버전 1 청크를 수신하여 재인코딩을 마친 뒤 버전 2 의 새 청크를 생성하는 반면, 피어 1 은 피어 2 에게 청크를 요청한 시점에서 버전 2 청크를 받아 디코딩에 실패한다.

이 문제를 해결하기 위해, 그림 2 와 같이 복구 과정의 동기화 제어 방법을 제안한다. 먼저 RS 코드를 활용해 블록파일을 k 개의 청크로 부호화 할 경우, 블록파일을 복구하기 위해서는 총 k 개의 청크들을 가져야 한다. 따라서 하나의 피어는 자신을 제외한 $k-1$ 개의 피어를 선택하여 인코딩 청크를 요청해야 한다. 이때 모든 피어가 받는 요청 수를 동일하고 최소한이 되는 $k-1$ 개가 되도록 하는 피어 집합을 정의할 수 있다. 예를 들어, 피어 N_i 가 선택한 피어 집합 $T(N_i)$ 는 다음과 같이 정의할 수 있다. 이때 n 은 네트워크에 참여한 피어의 수다.

$$T(N_i) = \{\overbrace{N_{(i+1) \bmod n}, N_{(i+2) \bmod n}, \dots N_{(i+k-1) \bmod n}}^{k-1}\}$$

집합 $T(N_i)$ 에 포함된 피어들은 청크 요청을 받으면 요청받은 횟수를 기록하고, 그 횟수가 $k-1$ 번에 도달하기 전까지는 캐시에 저장된 기존 버전의 인코딩 청크를 전송하여 모든 피어가 같은 버전의 청크를 디코딩할 수 있도록 동기화를 제어한다. 만약 네트워크 문제 등으로 청크 수신에 문제가 발생할 경우 해당 피어

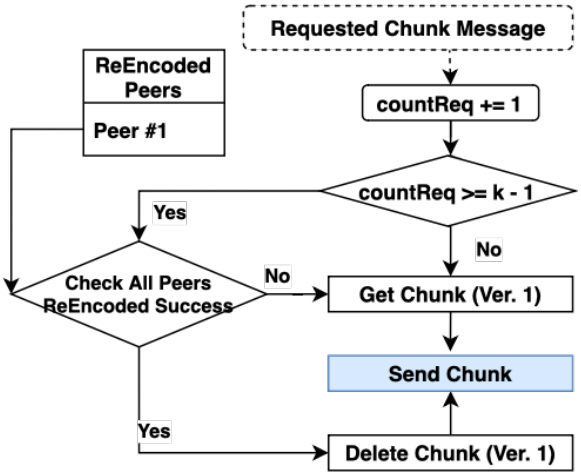

그림 2. 받은 요청 수를 기반으로 피어의 인코딩 청크 관리와 복구 과정 동기화를 제어하는 과정

는 청크를 재요청하고 모든 피어의 복구 완료를 확인한다. 이때 복구가 미완료된 피어 목록에 재요청한 피어가 있다면 기존 버전의 청크를 전송한다. 그림 2 의 인코딩 청크 관리 과정을 따라 모든 피어가 같은 버전의 블록파일을 복구하도록 과정을 동기화하고, 캐시를 비우는 것으로 저장공간의 오버헤드를 줄일 수 있다.

3. 결론

본 논문에서는 RS 코드를 적용한 하이퍼레저 패브릭에서 다수의 청크로 분할된 원장을 비동기적으로 복구할 때, 다른 버전 청크를 수신하는 문제를 발견했다. 그리고 해결방안으로 인코딩 청크 관리와 복구 과정 동기화 방법을 제안했다. 그러나 이전 청크를 일정 기간 보관하는 점, 모든 피어에 복구 완료 여부를 물어보는 점 등에서 오버헤드가 발생하게 된다. 향후에는 복구 과정을 동기적으로 수행하는 연구 등이 필요할 것이다.

* 이 논문은 2021 년도 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). 이 논문은 과학기술정보통신부 및 정보통신기획평가원의 SW 중심대학사업(2019-0-01183)의 지원을 받아 작성되었음.

참고문헌

[1] 박소현 외 4 명, “저장 효율적인 하이퍼레저 패브릭 블록체인을 위한 소거 코드 기반 분산 저장 시스템,” 한국통신학회 학술대회논문집, pp. 555-556, 2022.

[2] X. Liang, et al., “EduChain: A highly available education consortium blockchain platform based on Hyperledger Fabric,” Concurrency and Computation: Practice and Experience, vol. 35, no. 18, p. e6330, 2023