

블록 재인코딩을 위한 QUIC 기반 데이터 전송 성능 평가

노윤지¹, 이명철²

¹울산대학교 IT 융합학과 학부생

²한국전자통신연구원 스마트데이터연구실 책임연구원

younji0711@mail.ulsan.ac.kr, mclee@etri.re.kr

Performance Evaluation of QUIC-based Data Transmission for Block Re-encoding

Youn-Ji Noh¹, Myungcheol Lee²

¹Dept. of Computer Science, Ulsan University

²Smart Data Research Section, ETRI

요 약

블록체인의 데이터 중복 저장 문제를 해결하기 위해 소거 코드 기반의 분산 저장 방법이 제안되었고, 이에 따라 청크 데이터의 효율적인 전파가 중요해졌다. 본 연구는 이를 위해 QUIC 프로토콜을 도입하여 청크 데이터의 신속한 전파 기능을 구현하였다. 더불어 Reed-Solomon 코드 기반 시스템에서 QUIC 프로토콜의 효용성을 검증한 결과, QUIC 프로토콜은 분산 저장 시스템에서 청크 데이터 전파와 재인코딩 성능에 결정적 역할을 수행함을 확인하였다. 이는 QUIC 프로토콜이 블록체인과 같은 네트워크 환경에서 효과적으로 사용될 수 있음을 나타낸다.

1. 서론

블록체인은 데이터 무결성과 투명성을 보장하기 위해 네트워크에 참여하는 모든 노드가 동일한 데이터를 중복 저장하는 분산 원장 기술이다. 이러한 원장 중복 저장으로 인한 저장 공간 문제의 해결책으로 소거 코드(Erasure Code) 기반의 원장 분산 저장 방법이 제안되었다.[1]

대표적인 소거 코드인 Reed-Solomon 코드를 원장에 적용하면, 각 노드는 전체 원장이 아닌 일부 부호화된 청크만을 갖고도 원장의 가용성을 보장할 수 있게 된다. 그러나 일부 노드에 장애가 발생하면, 다른 노드로부터 청크들을 수집하여 원본 파일을 복구하는 재인코딩 작업이 필요하며, 이때 노드 간 청크 데이터의 효율적인 전파가 복구 성능에 큰 영향을 끼친다.

본 연구에서는 QUIC 프로토콜을 활용하여 Reed-

Solomon 코드가 적용된 블록체인 환경의 블록 복구 시나리오에서 노드 간 청크 데이터 전파 기능을 구현하고 성능 평가를 수행하였다.

2. 본론

QUIC(Quick UDP Internet Connections) 프로토콜은 구글에서 개발된 UDP 기반 전송 계층 프로토콜이다. QUIC은 TCP의 신뢰성과 UDP의 속도 장점을 결합하여, 빠른 연결 설정 및 안정적인 데이터 전달을 제공한다.[2] 이러한 특징은 블록체인 환경에서 Reed-Solomon 코드로 부호화된 청크 데이터의 신속하고 안정적인 전파에 중요한 역할을 할 수 있다.

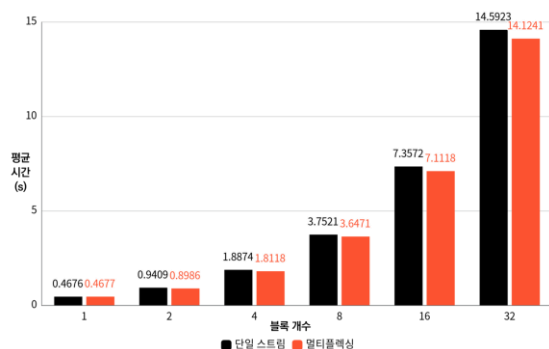
QUIC은 네트워크 자원 활용성을 증가시키고 병목 현상을 줄일 수 있도록 두 가지 주요 기능인 0-RTT(Zero-Round Trip Time)와 멀티플렉싱을 제공한다. 0-RTT는 초기 연결 설정 단계를 건너뛰어 최소한의

자연 시간으로 효율적인 통신을 시작하는 기능으로, 청크 데이터가 신속하게 네트워크 상으로 전파되어야 하는 블록체인 환경에서 매우 유용하다. 멀티플렉싱은 하나의 커넥션 위에서 여러 개의 스트림을 동시에 운용할 수 있게 하며, 여러 청크 데이터가 다른 스트림의 장애에 영향 없이 전파되도록 한다.

본 연구의 블록체인 네트워크는 CentOS 운영체제가 설치된 10 대의 노드로 구성되며, 10Gbps 네트워크로 서로 연결된다. 각 노드는 1/3 까지의 장애에 견딜 수 있는 BFT (Byzantine Fault Tolerance) 보장을 위해 (7,3) RS 코드를 적용하여 64MB 크기의 원본 블록 파일을 약 9.3MB 크기의 청크 데이터 10 개로 인코딩을 수행하였다. 장애가 발생하면, BFT 보장을 위해 재인코딩 과정이 시작되며 각 노드는 남은 노드들과 필요한 청크 데이터를 교환하여 원본 파일을 복구한다.

실험의 목적은 QUIC 프로토콜의 성능을 측정하고, 그 결과를 바탕으로 Reed-Solomon 이 적용된 블록체인 환경에서 블록 재인코딩 성능을 평가하는 것이다.

1 차 실험에서는 QUIC 기반 청크 전송 기능의 성능을 평가하였다. 총 10 대의 노드가 나머지 노드로부터 n 개의 블록 파일에 대한 청크 $n*9$ 개를 요청하였



으며, 이때 단일 스트림과 멀티플렉싱 방식의 블록 개수에 따른 평균 전송 시간을 비교 분석하였다.

(그림 1) 블록 개수에 따른 평균 전송 시간 비교

실험 결과, 멀티플렉싱은 단일 스트림 방식에 비해 평균 약 3.7% 더 짧은 전송 시간을 기록하여, 단일 스트림 대비 전송 용량(throughput)이 많고, 전체 전송

시간(latency)은 빨랐다. 그러나 개별 청크 전송 시간 기준으로, 단일 스트림이 청크를 먼저 받고 디코딩과 같은 다른 작업을 시작할 수 있는 이점을 가진다.

1 차 실험의 결과를 기반으로, 2 차 실험에서는 멀티플렉싱을 활용하여 Reed-Solomon 이 적용된 블록체인 환경에서 블록 재인코딩 시뮬레이션의 성능을 분석하였다. 네트워크의 노드 수가 10 에서 9 로 감소하는 경우, 남아있는 노드들은 하트비트 메커니즘을 통해 특정 노드의 부재를 인지한다. 총 10 개의 블록파일에 대한 청크 데이터들을 교환하여 원본 파일을 복구하고, (6,3) RS 코드를 적용하여 인코딩을 수행하고, 이때 전체 실행 시간과 각각의 작업(QUIC 전송, 디코딩/인코딩 과정 및 I/O 작업)에 걸린 시간을 측정했다.

<표 1> 멀티플렉싱을 활용한 블록 재인코딩 과정에서 각 작업의 소요 시간

작업	소요 시간
QUIC 전송	4.9978s (76.21%)
디코딩	1.0865s (16.57%)
인코딩	0.4342s (6.62%)
I/O	0.0071s (0.11%)
기타	0.0319s (0.49%)
총 소요 시간	6.5575s (100%)

실험 결과, QUIC 전송 시간은 전체 실행 시간의 약 76.21%를 차지하였다. 이는 QUIC 과 같은 통신 프로토콜을 잘 활용하여 데이터 전송 성능을 향상하는 것이 블록 재인코딩 성능에 결정적인 영향을 미칠 수 있음을 보여준다. 또한, 디코딩 및 인코딩 과정은 각각 실행 시간의 약 16.57%와 6.62%으로 상대적으로 적은 비중을 차지하지만, 원본 데이터 복구와 부호화된 청크를 재생성하는 데 중요한 역할을 한다. 청크 데이터를 파일 시스템에서 읽고 쓰는 I/O 작업과 기타 작업은 각각 약 0.11%, 0.49%를 차지하여 재인코딩 성능에 영향이 적었다.

3. 결론

본 연구는 Reed-Solomon 부호화가 적용된 블록체인 환경에서 부호화된 청크 데이터의 효율적인 전파를 위해 QUIC 프로토콜을 적용하고, 성능 평가를 수행하였다.

첫 번째로, 멀티플렉싱이 단일 스트림 방식에 비해 청크 데이터의 전송 시간을 평균적으로 약 3.7% 단축시켰고, 이는 10 개의 노드가 상호 연결된 환경에서 멀티플렉싱이 전체 데이터를 빨리 전송하는데 더 효율적임을 나타낸다.

두 번째로, QUIC 멀티플렉싱을 Reed-Solomon 재인코딩 과정에 활용하여 성능 평가를 수행하였고, 전체 재인코딩 과정에서 데이터 전송 시간이 상당한 시간(76.21%)을 차지함을 알 수 있었다. 결론적으로, 통신 프로토콜은 분산 저장 시스템과 같은 네트워크 상황에서 청크 데이터의 신속한 전파와 재인코딩 과정에 큰 기여를 할 수 있다는 것을 알게 되었다. 향후, gRPC[3] 와 같은 다른 통신 프로토콜과 블록체인 환경에서의 상대적인 성능 비교를 위한 추가적인 연구가 필요하다. 이러한 비교 연구는 분산 저장 시스템에 최적화된 네트워크 솔루션 선택에 도움을 줄 것으로 예상된다.

* 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).

참고문헌

- [1] 최병준 외 2 명, “블록체인 트랜잭션 데이터 분산 저장 기술 동향,” 전자통신동향분석, 제 37 권 제 3 호, 085-096, 2022
- [2] QUIC, “QUIC Version 2,” IETF, 2023, Available: <https://datatracker.ietf.org/doc/html/rfc9369>
- [3] gRPC, “A high performance, open source universal RPC framework,” 2023, Available: <https://grpc.io>