하이퍼레저 패브릭에서 보증 시간 감소를 위한 체인코드 실행 비용 기반 보증 피어 라우팅 방안 연구

장성일[O], 권재환[*], 김지용[*], 임채현[*], 김명호[**]
[O]숭실대학교 융합소프트웨어학과,
[*]숭실대학교 융합소프트웨어학과,
[**]숭실대학교 소프트웨어학부
e-mail: sungil@soongsil.ac.kr[O], {jaehwan, jyk, immanual1995}@soongsil.ac.kr[*], kmh@ssu.ac.kr[**]

Study on Chaincode execution cost base endorsing peer routing method to reduce endorsement time in Hyperledger Fabric

Sung-Il Jang[O], Jae-Hwan Kwon[*], Ji-Yong Kim[*], Chae-Hyun Im[*], Myung-Ho Kim[**]
[O]Dept. of Convergence Software, SoongSil University,
[*]Dept. of Convergence Software, SoongSil University,
[**]Dept. of Software, SoongSil University

● 요 약 ●

최근 블록체인이 활성화되면서 블록체인 시장 및 블록체인 모델의 활용도가 늘어나고 있다. 그중 하이퍼레저 패브릭은 프라이빗 블록체인의 대표적인 플랫폼이다. 하이퍼레저 패브릭에서 클라이언트는 트랜잭션을 보증 피어에게 전송할 때 사전에 정해진 보증 피어에게만 전송한다. 이는 트랜잭션의 실행 비용 및 보증 피어의 성능을 고려하지 않아 보증 시간을 증가시키는 문제가 발생한다. 본 논문은 이 문제를 해결하기 위해 트랜잭션의 실행 비용에 따라 효율적으로 보증 피어를 라우팅하는 기법을 제안한다.

키워드: 블록체인(blockchain), 하이퍼레저 패브릭(hyperledger fabric), 체인코드(chaincode)

I. Introduction

새로운 비즈니스 모델 중 하나인 블록체인이 활성화되면서 블록체인 시장 및 블록체인 모델의 활용도가 높아지고 있다. 최근 블록체인 네트워크에서도 다양한 연구가 진행되고 있다. 한국정보통신기술협회의 보고서에 따르면 블록체인 플랫폼 활성화를 위해 R3 CEV 컨소시엄을 구성하고 기술 개발 및 특허 등 활발한 연구가 진행되고 있다[1]. 또한 Ripple Lab에서는 Interledger(ILP)을 공개소프트웨어로 개발하여 은행과 직접적인 테스트 등을 진행하고 있으며, 2020년 2월, IBM에서는 블록체인 기반 하이퍼레저 패브릭 2.0을 공개하였고 글로벌 공개 표준 모델 서비스를 개발하는데 주력하고 있다[2].

본 논문에서는 하이퍼레저 패브릭의 클라이언트가 트랜잭션을 보증 피어에게 보낼 때 사전에 정의된 보증 피어에게만 보내는 문제점을 개선하여 트랜잭션의 비용과 보증 피어의 성능을 고려한다.

본 논문의 구성은 다음과 같다. 2장에서는 관련 연구에 대해 기술하고 3장에서는 실행 비용을 정의하고 실험 환경 구성 및 실험을 진행한다. 4장에서는 결론 및 향후 연구 방향에 대한 내용을 기술한다.

II. Preliminaries

1. Related works

1.1 블록체인

블록체인은 여러 트랜잭션으로 구성된 블록들을 연결시킨 데이터 집합이다. 트랜잭션은 일종의 거래내역이며, 블록체인 네트워크의 규칙에 따라 분산되어 저장된다. 원장들은 변경되는 데이터의 기록을 순차적으로 저장하고 있기 때문에, 임의로 조작이 불가능하다. 블록체인은 크게 두 종류의 블록체인으로 구분할 수 있는데 누구나 참여할 수 있는 퍼블릭 블록체인과 허가받은 참여자만 블록체인 네트워크에 접근 가능한 프라이빗 블록체인이 있다. 퍼블릭 블록체인의 대표적인 플랫폼은 비트코인과 이더리움이 있고, 프라이빗 블록체인의 대표적인 플랫폼은 하이퍼레저 패브릭, 리플 등이 있다[3, 4, 5, 6].

1.2 하이퍼레저 패브릭

하이퍼레저 패브릭은 리눅스 재단 블록체인 오픈소스 프로젝트 중 하나이다. 이 프로젝트는 블록체인 애플리케이션을 쉽게 개발할 수 있도록 블록체인 기반 기술을 개발자가 쉽게 사용할 수 있는 형태로 제공한다.

하이퍼레저 패브릭 네트워크의 참여자는 크게 클라이언트, 피어, 오더러로 구성된다. 클라이언트는 체인코드라 불리는 스마트 컨트랙트를 실행하는 주체이고, 피어는 원장을 유지하고 클라이언트로부터 트랜잭션을 전달받아 처리하는 노드이다. 피어는 원장을 유지만 할 수도 있고 유저의 트랜잭션을 처리할 수도 있는데 전자의 역할만 수행하면 커밋 피어라 부르고 후자의 역할도 겸하면 보증 피어라고 한다. 그리고 오더링 서비스를 제공하는 오더러가 있다.

체인코드는 하이퍼레저 패브릭 네트워크에 배포된 블록체인 애플리케이션이며 클라이언트는 체인코드의 함수를 호출하는 것으로 트랜잭션을 생성한다. 체인코드는 체인코드명과 버전으로 관리되고 각 피어에 설치된다. 만약 체인코드가 변경되면 업데이트를 통해 체인코드를 사용하는 모든 피어에 설치된 체인코드를 업데이트해야 한다. 피어에는 체인코드가 설치되어 있고 트랜잭션을 통해 피어의 원장 데이터를 다룰 수 있다.

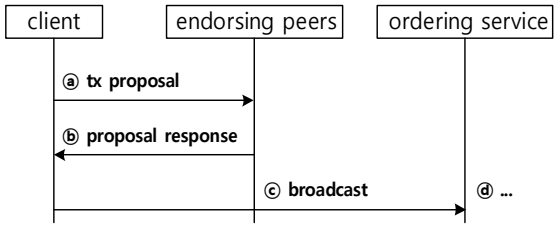

Fig. 1. Hyperledger Fabric Transaction Flow

트랜잭션은 Fig 1과 같은 과정을 거친다[5]. ⓓ 단계부터는 본 논문의 범위에서 벗어나기 때문에 설명하지 않고 앞선 3단계의 트랜잭션 처리 과정을 설명한다. ⓐ 단계는 클라이언트가 보증 피어에게 트랜잭션을 전송하는 단계이다. 이 단계에서 클라이언트는 체인코드의 보증정책에 맞게 요청 보낼 보증 피어들을 지정하고 체인코드의 규칙에 맞는 함수 및 인자를 정해서 보증 피어에게 트랜잭션 실행을 요청한다. ⓑ 단계는 보증 피어가 클라이언트로부터 요청받은 트랜잭션을 실제로 실행시켜보고 RWset을 반환하는 단계이다. ⓒ 단계에서는 클라이언트가 보증 피어들로부터 전달받은 응답들을 수집하여 오더링 서비스를 제공하는 노드(오더러)에게 브로드캐스트 한다.

본 논문에서는 ⓐ 단계와 ⓑ 단계를 수행한 시간을 보증 시간이라고 정의하고 실험에서는 이 시간을 실행 시간으로 하여 측정한다.

III. The Proposed Scheme

1. Chaincode Function Execution Cost

하이퍼레저 패브릭의 트랜잭션은 체인코드의 함수를 호출하기 위한 메시지다. 본 논문은 체인코드 연산들의 실행 비용을 미리 정의하여 트랜잭션의 실행 비용을 계산하고, 이를 기반으로 고비용 트랜잭션과 저비용 트랜잭션으로 구분한다.

각 트랜잭션의 실행 비용은 체인코드 함수가 내부적으로 호출하는 연산들의 실행 비용의 합이다. 체인코드는 작성하는 방식에 따라 다양한 연산을 사용할 수 있는데 본 논문에서는 accelerator에서 실험할 때 사용한 smallbank라는 체인코드를 기준으로 연산들의 실행 비용을 아래 Table 1과 같이 정의하였다[7].

Table 1. Execution Cost

Operation	EC (Execution Cost)
stub.GetState	10
stub.PutState	100
json.Marshal	100
json.Unmarshal	100
Return Value	100

그리고 체인코드에서 호출하는 연산들 중에서 실행 시간에 영향을 주는 연산을 선별하였다. GetState는 블록체인 데이터베이스에서 값을 읽어오는 함수인데 실행 시간은 짧지만 사용빈도가 높아 적은 실행 비용을 책정하였다. 이외에 함수들은 모두 동일하게 100의 실행 비용을 책정하였다. 그리고 smallbank의 Invoke 함수 중에서 4개를 선택하여 함수가 사용한 연산을 기준으로 실행 비용을 아래 Table 2와 같이 계산하였다.

Table 2. Smallbank Invoke Function

Name	Operation	EC
create_account	GetState: 1	210
	Marshal: 1	
	PutState: 1	
deposit_checking	GetState: 1	310
	Marshal: 1	
	PutState: 1	
	Unmarshal: 1	
send_payment	GetState: 2	620
	Marshal: 2	
	PutState: 2	
	Unmarshal: 2	
query	GetState: 1	110
	Return Value: 1	

본 논문에서 책정한 실행 비용이 실제 체인코드 실행 시간과 유사한지 확인해보기 위해 컨테이너 환경에서 실험을 진행하였다. 컨테이너의 CPU 사용량을 100%, 20%로 설정했을 때의 실행 시간은 다음 Fig 2와 같다. 그래프에서는 각 트랜잭션을 500개씩, 고성능 보증 피어와 저성능 보증 피어에 전송하였다. 100%를 사용할 때는 본 논문이 계산한 실행 비용과 실험 결과가 일치하였다, 하지만 20%를 사용할 때, deposit_checking이 create_account보다 빠르게 계산된 것으로 보아 실행환경에 따라 약간의 오차가 발생하는 것을 확인하였다.

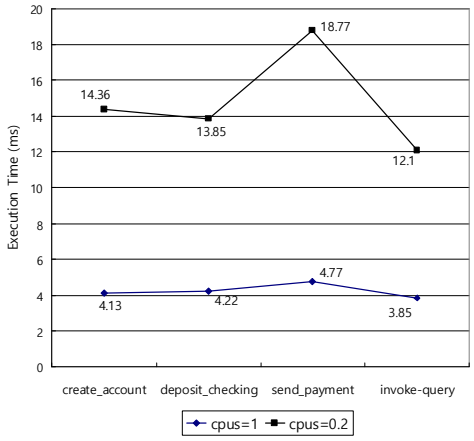

Fig. 2. Compare of Execution Time (cpus)

2. Experiment Environment

실험 환경의 CPU는 i5-8259U, 메모리는 16GB, 운영체제는 MacOS Catalina다. 실험을 수행한 하이퍼레저 패브릭의 네트워크는 아래 Fig 3과 같다.

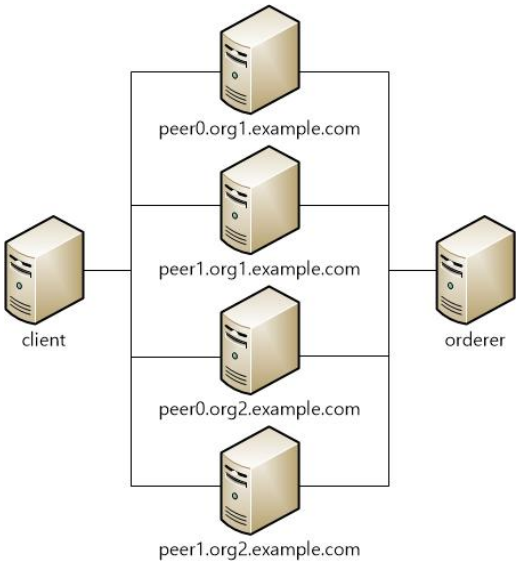

Fig. 3. Experiment Environment

보증 피어들의 성능은 peer0.org1.example.com이 1, 나머지 보증 피어들이 0.2의 CPU 성능을 가지고 있다. 클라이언트는 표 2의 각 함수마다 1000개의 트랜잭션을 생성한다. 기존 방식의 클라이언트는 트랜잭션의 비용 및 보증 피어의 성능을 고려하지 않고 트랜잭션을 고성능 보증 피어와 저성능 보증 피어에 각각 500개씩 라운드로빈 방식으로 보내고, 제안 방식의 클라이언트는 고비용 트랜잭션은 고성능 보증 피어에, 저비용 트랜잭션은 저성능 보증 피어에 각각 1000개씩 보낸다.

3. Experiment

실험 결과는 아래 Fig 4와 같다. 기존 방식의 클라이언트는 고성능 보증 피어와 저성능 보증 피어에 500개씩 라운드로빈 방식으로 트랜잭션을 보냈기 때문에 실행 시간이 평균적으로 완만한 추세를 보인다. 제안 방식의 클라이언트는 트랜잭션의 비용에 따라 보증 피어를 선택하기 때문에 아래 결과와 같이 고비용 트랜잭션에서 상대적으로 짧은 실행 시간을 가진다.

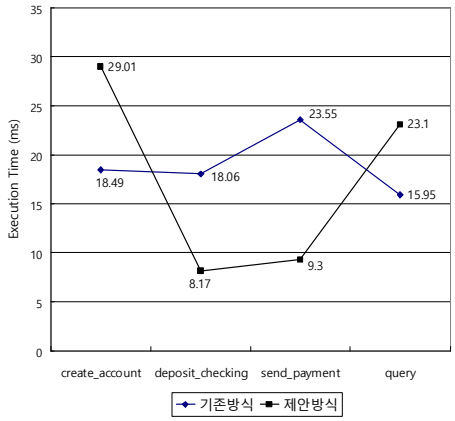

Fig. 4. Execution Time for each function

기존 방식과 제안 방식의 각각 4,000개의 트랜잭션 평균 실행 시간은 아래 Table 3과 같다. 기존 방식에 비해 제안 방식은 약 8.5%의 성능 개선을 보였다.

Table 3. Comparison of Execution Time

Method	Average Time (ms)
기존 방식	19.012
제안 방식	17.395

IV. Conclusions

본 논문에서는 하이퍼레저 패브릭의 보증 시간을 감소시킬 수 있는 방안에 대해 연구하였다. 체인코드 연산의 실행 비용을 기반으로 고비용의 트랜잭션은 고성능의 보증 피어로부터 보증을 받도록 라우팅 해주는 방안을 제안하였다.

실험을 통해 체인코드 함수의 연산들이 실행 시간에 영향을 미치는 것을 확인하였고, 각 연산의 실행 비용을 정의하여 고비용의 트랜잭션은 고성능의 보증 피어가 보증을 하는 방법에 대해 실험하였다. 그리고 실험 결과를 통해 고비용의 트랜잭션을 고성능의 보증 피어가 담당하면 전체 트랜잭션의 보증 시간을 감소시킬 수 있다는 것을 확인하였다.

향후에 실험 환경을 컨테이너 방식에서 베어메탈 방식으로 전환하여 좀 더 정밀한 실험을 한다면 성능 개선의 여지가 있을 것으로 보인다.

ACKNOWLEDGEMENT

이 논문은 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임(과제번호: 1711116903)

REFERENCES

[1] Telecommunications Technology Association, ICT Standa rdization Strategy Map, 2019.

[2] A.Hope-Bailie, S. Thomas, "Interledger: Creating a stan dard for payments." , Proceedings of the 25th Internatio nal Conference Companion on World Wide Web, pp.28 1-282 ,2016.

[3] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system", 2008.

[4] G.Wood, "Ethereum: A secure decentralised generalised transaction ledger." Ethereum project yellow 151, pp.1-32, 2014.

[5] E. Androulaki, A. Barger, V. Bortnikov, C. Cachin, K. Christidis, A. D. Caro, et al., "Hyperledger fabric: A distributed operating system for permissioned blockchains", Proceedings of the 13th ACM SIGOPS European Conference on Computer Systems, pp. 1-15, 2018.

[6] F. Armknecht, G. O. Karame, A. Mandal, F. Youssef, "Ripple: Overview and outlook." International Conference on Trust and Trustworthy Computing. Springer, Cham, 2015.

[7] KS. Lee, CS. Yoon, KW. Sung, N. Lincoln, KW. Heo, R. Vaculin, R. B. Hartley K. K. Um, “Accelerating Troughput in Permissioned Blockchain Networks.”, 2019