

트랜잭션 성능 향상 모듈 탑재가 가능한 블록체인 플랫폼 연동 Agent 설계 및 개발

이부형, 김민규, 김태현
(주)디지캡

bhlee@digicaps.com, mgkim@digicaps.com, thkim@digicaps.com

Design and Implementation of Blockchain Agent for Pluggable Transaction Performance Enhancement Module

Lee Boo Hyung, Kim Min Gyu, Kim Tae Hyun
DigiCAP Co., Ltd.

요 약

블록체인 기술이 대중화되면서 IoT, 물류, 금융, 제조 등 많은 분야에서 블록체인을 도입하기 위한 노력을 하고 있으나 다수의 기업이 플랫폼 고유의 거래처리속도 문제로 적용하는 데 어려움을 겪고 있다. 본 논문에서는 거래처리속도 문제 해결을 위한 모듈 개발 후 블록체인에 추가할 수 있도록 환경을 구성해주는 Chain-Agent 를 설계했다.

I. 서 론

Private 블록체인의 대표적인 플랫폼인 Hyperledger Fabric[1]은 허가된 Peer 들끼리의 서비스를 위해 설계된 플랫폼으로 Public 블록체인 플랫폼보다는 빠른 처리속도를 지원하지만, 순간 거래처리 요구량이 많고 낮은 지연시간 보장을 요구하는 결제나 이커머스 등의 분야에서는 거래가 지연될 수 있어 사용자의 불편함을 야기하기 때문에 적용하는 데 어려움을 겪고 있다.

Hyperledger Fabric 은 소스 코드가 공개되어 있어 서비스 제공자가 직접 수정하여 트랜잭션 처리 방식을 간소화하거나 알고리즘을 개선하는 방법으로 성능을 향상할 수도 있지만, 모듈을 직접 수정하는 것은 기술적 난이도가 매우 높고 빠르게 업데이트되는 오픈소스와의 호환성을 유지하기 어렵다. 그렇기 때문에 Pluggable 한 성능 향상 모듈을 개발하여 기존의 플랫폼에 연동하는 방법이 있다. 2019 년 2 월, 삼성 SDS 가 발표한 Nexledger Accelerator[2]는 멀티 레벨 큐 스케줄링 방식의 트랜잭션 처리를 통해 Hyperledger Fabric 에서 많은 양의 트랜잭션을 한꺼번에 처리할 수 있도록 설계되었다. 기존 방식은 각 거래에 대해 매번 합의 과정이 필요하지만, Nexledger Accelerator 는 접근하려는 키나 주소가 중복되지 않는 다수의 거래를 분류하여 하나의 신규 트랜잭션으로 결합하고 합의하는 방식을 사용한다. 단일 합의 단계에서 더 많은 트랜잭션을 처리할 수 있기 때문에 시간당 처리 성능이 향상된다.

이와 같은 성능 향상 모듈은 다양한 방식으로 개발될 수 있으며 클라이언트 앱과 블록체인 플랫폼과의 연결 인터페이스 기능도 함께 지원해야 한다. 성능 향상 모듈을 탑재하여 블록체인 플랫폼에 연결해주는 Agent 가 제공된다면 성능 향상 모듈 개발자는 트랜잭션

처리 알고리즘 설계나 개발에 더 집중할 수 있고 이는 확장성 개선을 위한 연구에 도움이 될 수 있다.

본 논문에서는 Hyperledger Fabric 기반 성능 향상 모듈을 쉽게 연동할 수 있는 Agent 의 설계 내용을 제안하고자 한다. 제안하는 Agent 는 블록체인의 거래 처리속도 향상을 위한 다양한 모듈과의 연동을 지원한다.

II. 본론

블록체인의 트릴레마는 확장성, 탈중앙화, 보안성으로 알려져 있다[3]. 이 중에 확장성은 사용자 수의 증가에 대응하는 정도를 뜻하며, 사용자 증가에 따라 전송 처리량도 증가할 수 있음을 의미한다. 최근 1 초당 거래를 처리하는 속도인 TPS(Transaction Per Second) 향상이 블록체인을 활성화하기 위한 방안으로 대두되면서 확장성을 해결하기 위해 합의 알고리즘 변경하거나 기존 블록체인에 성능 향상 모듈을 추가하는 등의 방법들이 제시되었다[4].

제시된 방안 중 블록체인에 성능 향상 모듈을 추가하는 방법은 블록체인뿐만 아니라 클라이언트 앱과의 연동을 고려해야 하며, 이를 위해 성능 향상 모듈을 추가하기 위해서는 클라이언트 앱과 블록체인 네트워크 연결을 위한 개발을 따로 해야 했다. 따라서 본 논문에서는 블록체인 기반의 확장성 증가를 위해 개발된 모듈이 블록체인에 바로 추가될 수 있는 환경을 구성해주는 Agent 인 “Chain-Agent”의 시스템 설계를 제안한다.

Chain-Agent 는 블록체인 네트워크에서 클라이언트 앱과 블록체인 사이에 기본 환경을 구성하고 트랜잭션 처리 속도 향상을 위해 개발된 모듈을 탑재할 수 있는 기능을 제공한다. 이를 통해 성능 향상 모듈 개발자들은 클라이언트 앱 및 블록체인의 연동을 고려하지 않고도 Chain-Agent 를 사용하여 블록체인에 추가할 수 있기 때문에 모듈 개발 후 적용이 용이하며 다른 모듈과 성능 비교를 위한 동일 환경을 쉽게 구축할 수 있는 장점이 있다.

1. Chain-Agent 설계 제안

Chain-Agent 시스템은 사용자로부터 Transaction 을 수신하는 Application Interface, 사용자와 통신을 위한 Stub 서버, 모듈과 블록체인 연동 인터페이스를 관리하는 Agent-Adapter, 모듈과 연동을 위한 Batch Interface 그리고 블록체인 연동을 위한 Chain Interface 로 구성된다. Chain-Agent 의 시스템 아키텍처는 그림 1 과 같다.

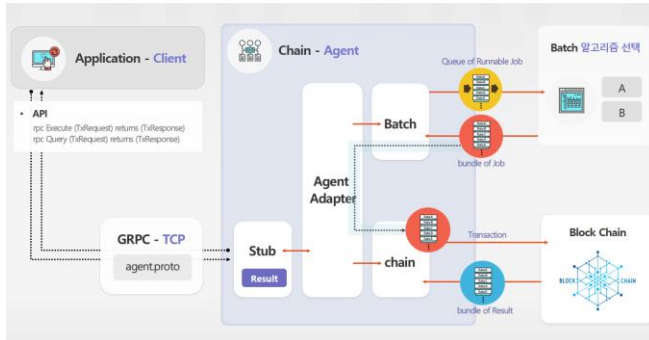


그림 1 Chain-Agent Architecture

Application Interface 에 정의된 API 를 사용하여 클라이언트가 거래를 요청하면 stub 서버는 TCP protocol 의 gRPC(gRPC Remote Procedure Calls)를 사용하여 클라이언트가 Application 으로 요청한 내용을 수신한다. Application Interface 의 Execution 과 Query API 는 클라이언트의 write, read 작업을 요청받는다. Agent Adapter 는 사용자에게 요청받은 입출력을 추상화하여 Batch 모듈 인터페이스와 Chain 인터페이스를 통합하는 역할을 수행한다. Batch 모듈 인터페이스는 성능 향상을 위한 배치 모듈과 쉽게 연동하기 위한 인터페이스로써 추상화된 클라이언트 요청 자료를 모듈로 전달하는 역할을 한다. Chain 인터페이스는 Batch 알고리즘 모듈에서 응답받은 Transaction 번들을 블록체인의 SDK 를 통하여 Peer 로 전송하여 일괄 처리하고, Peer 로부터 처리 결과를 수신하는 역할을 한다.

2. Chain-Agent 시스템 구현 및 분석

Chain-Agent 시스템 설계에서 제안한 설계를 기반으로 Chain-Agent 의 프로토타입을 구현하여 실험했다. 프로토타입 구현을 위해 동시성 제어 및 서버 시스템의 구현이 가능한 Golang 언어를 사용하고 Hyperledger Fabric 의 블록체인을 연결했으며, 클라이언트 앱과 통신은 gRPC 를 사용하여 서버를 구현했다.

Chain-Agent 에 배치 모듈 테스트를 위해 내부에서 배치 모듈을 구성하여 탑재했다. 그림 2 는 Chain-Agent 가 구현된 시스템에 내부 배치 모듈을 탑재하고 블록체인과 연동한 로그다. Tx 부분에서 클라이언트가 요청한 여러 개의 Transaction 을 내부 배치 모듈로 전달하여 하나의 번들로 구성하고, 구성된 번들을 블록체인의 Peer 로 전달하여 처리하는 것을 Send 부분에서 확인할 수 있다.

프로토타입 구현 및 테스트를 통해 논문에서 제안된 설계대로 Chain-Agent 을 구현하여 성능 향상 모듈을 탑재할 수 있었다. 정의된 프로토콜을 사용하여 내부 모듈 교체도 가능했으며, 성능 향상 모듈만 개발하면 Chain-Agent 에 탑재해서 블록체인에 바로 적용할 수 있는 것을 확인했다. 성능 향상을 위한 Batch 알고리즘만 집중 해서 개발한 후 블록체인에 바로 적용할 수 있기 때문에 앞으로 블록체인에서 확장성 문제를 해결하는데 도움이 될 수 있을 것이다.

```
At block #6 of@6
Forest(6): <68>
TreeID(0):
Keys: 7 6
Tx(81): [6 7]
Tx(69): [7 6 6 6]
Tx(64): [6 7 6]
Tx(39): [6 6 7]
Tx(21): [6 6 7]
Tx(1): [6 7 6]

Send
[6 map[0:([81 [] [6 7] 10 value of 81) (69 [] [7 6 6 6] 10 value of 69) (64 [] [6
of 1]) map[6:true 7:true] 68)] 68]

At block #7 of@7
Forest(7): <98>
TreeID(0):
Keys: 8 9
Tx(76): [8 8]
Tx(81): [9 9 9]
Tx(90): [9 9 9]
Tx(72): [9 8 9 8]
Tx(52): [8 8 9 9]
Tx(50): [9 9]
Tx(20): [9 9 8 8]
Tx(16): [9 8 9 8]
Tx(11): [8 9]

Send
[7 map[0:([76 [] [8 8] 10 value of 76) (91 [] [9 9 9] 10 value of 91) (90 [] [9 9
of 50) (20 [] [9 9 8 8] 10 value of 20) (16 [] [9 8 9 8] 10 value of 16) (11 [] [8
```

그림 2 Chain-Agent 로그

III. 결론

본 논문에서는 거래 처리 성능 향상 모듈을 탑재하여 블록체인 플랫폼과 어플리케이션 사이의 간편한 연동을 제공하는 Chain-Agent 를 제안하였다. 여러 개의 트랜잭션을 한꺼번에 처리할 수 있는 내부 배치 모듈을 탑재하고 Hyperledger Fabric 네트워크와 연동하여 Chain-Agent 의 동작을 확인했다. 성능 향상 모듈의 알고리즘과 관계없이 블록체인 플랫폼과 연동하여 거래 처리 결과를 확인할 수 있는 Agent 를 제공함으로써 추가 인터페이스 개발에 들어가는 노력을 최소화하고 블록체인 거래처리 속도 개선 연구에 도움이 될 수 있다. 향후에는 다른 블록체인 플랫폼과의 연동을 지원할 수 있도록 구현하여 다양한 성능 향상 모듈의 탑재를 제공할 예정이다.

ACKNOWLEDGMENT

이 논문은 2020 년도 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원의 지원을 받아 수행된 연구임 (No.2020-0-00063, 소규모 노드환경에서 성능 극대화를 위한 멀티채널 기반 블록체인 기술 개발)

This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) (No.2020-0-00063, Development of multichannel-based blockchain technology for maximizing performance in small node environment)

참 고 문 헌

- [1] Androulaki, Elli, et al. "Hyperledger fabric: a distributed operating system for permissioned blockchains." Proceedings of the thirteenth EuroSys conference. 2018.
- [2] Lee, Hyojung, et al. "Multi-batch Scheduling for Improving Performance of Hyperledger Fabric based IoT Applications." 2019 IEEE Global Communications Conference (GLOBECOM). IEEE, 2019.
- [3] Abadi, Joseph and Markus Brunnermeier. "Blockchain economics." 2018 Tech. rep., mimeo Princeton University
- [4] Ethereum EIP-1011, transition Ethereum main net from Proof of Work (PoW) to Proof of Stake (PoS) (<https://github.com/ethereum/EIPs/blob/master/EIPS/eip-1011.md>)