(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2024-0055447
(43) 공개일자 2024년04월29일

(51) 국제특허분류(Int. Cl.)
G06F 16/27 (2019.01) *G06F 16/176* (2019.01)
G06F 16/23 (2019.01) *G06F 9/46* (2006.01)
G06F 9/54 (2018.01)
(52) CPC특허분류
G06F 16/27 (2019.01)
G06F 16/176 (2019.01)
(21) 출원번호 10-2022-0135739
(22) 출원일자 2022년10월20일
심사청구일자 2022년10월20일

(71) 출원인
포항공과대학교 산학협력단
경상북도 포항시 남구 청암로 77 (지곡동)
(72) 발명자
박찬익
경상북도 포항시 남구 청암로 77
황제영
경상북도 포항시 남구 청암로 77
(74) 대리인
특허법인이상

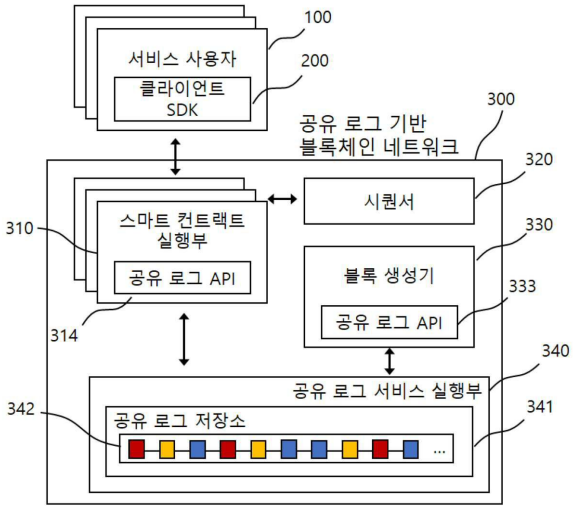
전체 청구항 수 : 총 19 항

(54) 발명의 명칭 **공유 로그 기반 블록체인 네트워크 시스템 및 블록체인 네트워크의 사용자 트랜잭션 처리 방법**

(57) 요 약

본 발명의 목적을 달성하기 위한 일 실시예에 따른 블록체인 네트워크 시스템은, 사용자 트랜잭션 생성에 필요한 정보를 제공하는 서비스 사용자로부터 사용자 정보에 기반하여 사용자 트랜잭션을 생성하는 클라이언트 측 소프트웨어 인터페이스를 경유하여 전송되는 사용자 트랜잭션을 수신하고, 사용자 트랜잭션을 실행하는 복수 개의 스마트 컨트랙트 실행 노드들; 및 복수 개의 스마트 컨트랙트 실행 노드들에 의하여 실행되는 사용자 트랜잭션의 실행 결과로 생성된 커밋 레코드를 수신하고, 커밋 레코드를 공유 로그로서 저장하는 적어도 하나 이상의 공유 로그 스토리지를 포함한다.

대 표 도 - 도1



(52) CPC특허분류
G06F 16/2365 (2019.01)
G06F 16/2379 (2019.01)
G06F 9/466 (2013.01)
G06F 9/544 (2013.01)

이 발명을 지원한 국가연구개발사업

과제고유번호	1711134661
과제번호	2021-0-00484-001
부처명	과학기술정보통신부
과제관리(전문)기관명	정보통신기획평가원
연구사업명	데이터경제를위한블록체인기술개발
연구과제명	노드 간 메시지 전달과 합의를 위한 최적 경로 네트워크 프로토콜 기술개발
기 여 율	1/1
과제수행기관명	포항공과대학교 산학협력단
연구기간	2021.04.01 ~ 2021.12.31

명 세 서

청구범위

청구항 1

사용자 트랜잭션 생성에 필요한 정보를 제공하는 서비스 사용자로부터 사용자 정보에 기반하여 사용자 트랜잭션을 생성하는 클라이언트 측 소프트웨어 인터페이스를 경유하여 전송되는 상기 사용자 트랜잭션을 수신하고, 상기 사용자 트랜잭션을 실행하는 복수 개의 스마트 컨트랙트 실행 노드들; 및

상기 복수 개의 스마트 컨트랙트 실행 노드들에 의하여 실행되는 상기 사용자 트랜잭션의 실행 결과로 생성된 커밋 레코드를 수신하고, 상기 커밋 레코드를 공유 로그로서 저장하는 적어도 하나 이상의 공유 로그 스토리지;

를 포함하는 공유 로그 기반 블록체인 네트워크 시스템.

청구항 2

제1항에 있어서,

상기 복수개의 스마트 컨트랙트 실행 노드들 각각은 상기 복수개의 스마트 컨트랙트 실행 노드들 각각에 할당되는 사용자 트랜잭션을 병렬적으로 처리하는,

공유 로그 기반 블록체인 네트워크 시스템.

청구항 3

제1항에 있어서,

상기 복수개의 스마트 컨트랙트 실행 노드들로부터 상기 커밋 레코드를 수신하고, 상기 커밋 레코드의 유효성을 검증하고, 상기 커밋 레코드가 유효하다고 판단되는 경우 공유 로그 스토리지 주소 토큰을 생성하여 상기 복수개의 스마트 컨트랙트 실행 노드들로 전송하는 시퀀서 노드;

를 더 포함하는,

공유 로그 기반 블록체인 네트워크 시스템.

청구항 4

제3항에 있어서,

상기 시퀀서 노드는 상기 복수개의 스마트 컨트랙트 실행 노드들에 의하여 병렬적으로 실행되는 사용자 트랜잭션 간 유효성 검사를 실행하고, 상기 적어도 하나 이상의 공유 로그 스토리지에 쓰기 가능한 주소를 포함하는 상기 공유 로그 스토리지 주소 토큰을 생성함으로써 상기 복수개의 스마트 컨트랙트 실행 노드들 간의 사용자 트랜잭션 실행 순서에 대한 합의를 지원하는,

공유 로그 기반 블록체인 네트워크 시스템.

청구항 5

제3항에 있어서,

상기 시퀀서 노드는 상기 커밋 레코드의 유효성을 검증하기 위해 상기 커밋 레코드 내 읽기 집합에 대하여 각 개체에 대한 키-버전 쌍과 상기 읽기 집합에 포함된 개체들의 버전 정보를 비교하는 다중 버전 충돌 체크(MVCC, Multi-version conflict check)를 수행하는,

공유 로그 기반 블록체인 네트워크 시스템.

청구항 6

제1항에 있어서,

상기 적어도 하나 이상의 공유 로그 스토리지에 기록된 사용자 트랜잭션 커밋 레코드를 포함하는 로그 데이터를 공유 로그 인터페이스를 통하여 읽어들이고, 상기 로그 데이터를 이용하여 블록을 생성하는 블록 생성 노드;

를 더 포함하는 공유 로그 기반 블록체인 네트워크 시스템.

청구항 7

제6항에 있어서,

상기 블록 생성 노드는 상기 블록을 생성함에 있어 기존 블록체인들과 호환되는 새로운 블록을 생성하고, 상기 블록을 전파하는,

공유 로그 기반 블록체인 네트워크 시스템.

청구항 8

제6항에 있어서,

상기 블록 생성 노드는 상기 로그 데이터에 관련되는 사용자 트랜잭션들의 순차적인 기록 및 스마트 컨트랙트 실행 과정에서 이미 합의된 순서를 이용하여 상기 블록을 생성하는,

공유 로그 기반 블록체인 네트워크 시스템.

청구항 9

제1항에 있어서,

상기 복수개의 스마트 컨트랙트 실행 노드들은 공유 로그 인터페이스를 포함하고,

상기 복수개의 스마트 컨트랙트 실행 노드들에 의하여 생성되는 상기 커밋 레코드를 포함하는 로그 데이터가 상기 공유 로그 인터페이스를 경유하여 상기 적어도 하나 이상의 공유 로그 스토리지로 전송되고,

상기 커밋 레코드를 포함하는 로그 데이터가 상기 적어도 하나 이상의 공유 로그 스토리지에 저장되는,

공유 로그 기반 블록체인 네트워크 시스템.

청구항 10

제1항에 있어서,

상기 복수개의 스마트 컨트랙트 실행 노드들은 공유 로그 인터페이스를 포함하고,

상기 복수개의 스마트 컨트랙트 실행 노드들은 상기 사용자 트랜잭션을 실행할 때 상기 커밋 레코드를 생성하기 위하여 필요한 최신 상태정보를 획득하기 위하여 상기 공유 로그 인터페이스를 경유하여 상기 적어도 하나 이상의 공유 로그 스토리지로부터 최신 로그 데이터를 읽어들이는,

공유 로그 기반 블록체인 네트워크 시스템.

청구항 11

제1항에 있어서,

상기 적어도 하나 이상의 공유 로그 스토리지는 상기 커밋 레코드를 포함하는 로그 데이터를 복수개의 속성으로 저장하고, 상기 복수개의 속성은 트랜잭션 타입, 클라이언트 ID, 트랜잭션 ID, 자바 프로그램 정보, 전달 인자, 및 실행 결과 중 적어도 하나 이상을 포함하는,

공유 로그 기반 블록체인 네트워크 시스템.

청구항 12

사용자 트랜잭션 생성에 필요한 정보를 제공하는 서비스 사용자로부터 사용자 정보에 기반하여 생성된 사용자 트랜잭션을 클라이언트 측 소프트웨어 인터페이스를 경유하여 복수개의 스마트 컨트랙트 실행 노드들 중 적어도 일부로 전송하는 단계;

상기 복수개의 스마트 컨트랙트 실행 노드들 중 적어도 일부가 상기 사용자 트랜잭션을 실행하는 단계;

상기 복수개의 스마트 컨트랙트 실행 노드들 중 적어도 일부가 상기 사용자 트랜잭션이 실행된 결과로 커밋 레코드를 생성하는 단계; 및

적어도 하나 이상의 공유 로그 스토리지가 상기 커밋 레코드를 공유 로그로서 저장하는 단계;

를 포함하는,

공유 로그 기반 블록체인 네트워크의 사용자 트랜잭션 처리 방법.

청구항 13

제12항에 있어서,

상기 사용자 트랜잭션을 상기 복수개의 스마트 컨트랙트 실행 노드들 중 적어도 일부로 전송하는 단계는

상기 클라이언트 측 소프트웨어 인터페이스가 복수개의 상기 사용자 트랜잭션을 상기 복수개의 스마트 컨트랙트 실행 노드들에 할당하는 단계;

를 포함하고,

상기 사용자 트랜잭션을 실행하는 단계는

상기 복수개의 스마트 컨트랙트 실행 노드들 각각이 상기 복수개의 스마트 컨트랙트 실행 노드들 각각에 할당되는 상기 복수개의 상기 사용자 트랜잭션을 병렬적으로 처리하는,

공유 로그 기반 블록체인 네트워크의 사용자 트랜잭션 처리 방법.

청구항 14

제12항에 있어서,

상기 커밋 레코드를 생성하는 단계의 이후 및 상기 커밋 레코드를 상기 공유 로그로서 저장하는 단계의 이전에,

시퀀서 노드가 상기 복수개의 스마트 컨트랙트 실행 노드 중 적어도 일부로부터 상기 커밋 레코드를 수신하는 단계;

상기 시퀀서 노드가 상기 커밋 레코드의 유효성을 검증하는 단계;

상기 커밋 레코드가 유효하다고 판단되는 경우 상기 시퀀서 노드가 공유 로그 스토리지 주소 토큰을 생성하는 단계; 및

상기 시퀀서 노드가 상기 공유 로그 스토리지 주소 토큰을 상기 복수개의 스마트 컨트랙트 실행 노드들 중 적어도 일부로 전송하는 단계;

를 더 포함하는,

공유 로그 기반 블록체인 네트워크의 사용자 트랜잭션 처리 방법.

청구항 15

제14항에 있어서,

상기 시퀀서 노드가 상기 커밋 레코드의 유효성을 검증하는 단계에서

상기 복수개의 스마트 컨트랙트 실행 노드들에 의하여 병렬적으로 실행되는 사용자 트랜잭션 간 유효성 검사가 실행되고,

상기 공유 로그 스토리지 주소 토큰을 생성하는 단계에서

상기 적어도 하나 이상의 공유 로그 스토리지에 쓰기 가능한 주소를 포함하는 상기 공유 로그 주소 토큰이 생성됨으로써 상기 복수개의 스마트 컨트랙트 실행 노드들 간의 사용자 트랜잭션 실행 순서에 대한 합의가 지원되는,

공유 로그 기반 블록체인 네트워크의 사용자 트랜잭션 처리 방법.

청구항 16

제14항에 있어서,

상기 시퀀서 노드가 상기 커밋 레코드의 유효성을 검증하는 단계에서

상기 시퀀서 노드는 상기 커밋 레코드의 유효성을 검증하기 위해 상기 커밋 레코드 내 읽기 집합에 대하여 각 개체에 대한 키-버전 쌍과 상기 읽기 집합에 포함된 개체들의 버전 정보를 비교하는 다중 버전 충돌 체크(MVCC, Multi-version conflict check)를 수행하는,

공유 로그 기반 블록체인 네트워크의 사용자 트랜잭션 처리 방법.

청구항 17

제12항에 있어서,

블록 생성 노드가 상기 적어도 하나 이상의 공유 로그 스토리지에 기록된 사용자 트랜잭션 커밋 레코드를 포함하는 로그 데이터를 공유 로그 인터페이스를 통하여 읽어들이는 단계;

상기 블록 생성 노드가 상기 로그 데이터를 이용하여 기존 블록체인들과 호환되는 새로운 블록을 생성하는 단계; 및

상기 블록을 전파하는 단계;

를 더 포함하는,

공유 로그 기반 블록체인 네트워크의 사용자 트랜잭션 처리 방법.

청구항 18

제17항에 있어서,

상기 블록 생성 노드가 상기 새로운 블록을 생성하는 단계에서,

상기 블록 생성 노드는 상기 로그 데이터에 관련되는 사용자 트랜잭션들의 순차적인 기록 및 스마트 컨트랙트 실행 과정에서 이미 합의된 순서를 이용하여 상기 블록을 생성하는,

공유 로그 기반 블록체인 네트워크의 사용자 트랜잭션 처리 방법.

청구항 19

제12항에 있어서,

상기 적어도 하나 이상의 공유 로그 스토리지가 상기 커밋 레코드를 공유 로그로서 저장하는 단계에서,

상기 적어도 하나 이상의 공유 로그 스토리지는 상기 커밋 레코드를 포함하는 로그 데이터를 복수개의 속성으로 저장하고, 상기 복수개의 속성은 트랜잭션 타입, 클라이언트 ID, 트랜잭션 ID, 자바 프로그램 정보, 전달 인자, 및 실행 결과 중 적어도 하나 이상을 포함하는,

공유 로그 기반 블록체인 네트워크의 사용자 트랜잭션 처리 방법.

발명의 설명

기 술 분 야

[0001] 본 발명은 블록체인 서비스를 제공하는 블록체인 네트워크에서 사용자 트랜잭션을 처리하는 방법에 관한 것으로, 사용자 트랜잭션을 병렬적으로 처리할 수 있는 공유 로그 환경 기반의 블록체인 네트워크 시스템 및 그 블록체인 네트워크의 사용자 트랜잭션 처리 기술에 관한 것이다.

배 경 기 술

[0002] 블록체인(Blockchain)은 P2P 네트워크를 기반으로 탈중앙화, 무결성, 투명성을 보장하는 분산 원장 시스템이다. 네트워크에 참여하는 사용자는 블록이라고 하는 동일한 데이터 구조에 모든 거래 내역을 포함하여 저장하고 합

의 프로토콜에 따라 새로운 블록을 생성한다. 즉, 블록체인은 트랜잭션(Transaction)들의 집합으로 구성된 블록이 이전 블록의 해시(hash)값을 담아 모든 블록을 체인 형식으로 연결하는 데이터 구조로서, 블록체인 네트워크에 참여하는 모든 노드(node)가 상기 데이터 구조를 동일하게 유지한다. 새로운 블록이 블록체인에 반영되기 위해서는 노드 간 합의가 필요하며, 각 블록체인 네트워크에 사용되는 합의 알고리즘(Consensus algorithm)은 상이하다. 블록체인은 특정 노드의 블록 데이터가 임의로 조작되더라도 이전 블록의 해시값을 가지고 있으므로 유효성 검증이 가능하며, 조작된 데이터는 블록체인에 반영되지 않는다. 이처럼 블록체인은 데이터를 임의로 위변조하는 것이 불가능하여 데이터의 무결성 및 투명성을 보장할 수 있다.

[0003] 블록체인은 비허가형 블록체인(Permissionless blockchain)과 허가형 블록체인(Permissioned blockchain)으로 구분된다. 비허가형 블록체인은 사용자 및 노드가 아무런 제약 없이 블록체인 네트워크에 참여할 수 있는 블록체인이다. 대표적인 비허가형 블록체인으로는 비트코인(Bitcoin) 및 이더리움(Ethereum)이 있다. 허가형 블록체인 혹은 컨소시엄 블록체인은 허가된 사용자 및 노드들만 블록체인 네트워크에 참여할 수 있는, 비즈니스 환경에서 활용하기에 적합한 블록체인이다. 대표적인 허가형 블록체인으로는 하이퍼레저 패브릭(Hyperledger Fabric)이 있다. 하이퍼레저 패브릭은 Execute-Order-Validate (EOV) 실행 모델을 따르는 블록체인으로, 기존 Order-Execute(O-X) 실행 모델을 따르던 블록체인 시스템의 단점을 보완하고자 도입되었다.

[0004] 스마트 컨트랙트(smart contract)란 블록체인을 기반으로 공증, 부동산 계약 등 다양한 형태의 계약을 체결하고 이행하는 분산 응용 프로그램을 의미한다. 스마트 컨트랙트는 노드들의 합의로 실행되는 프로그램으로, 이는 노드들이 동일한 상태 하에 정해진 동작을 수행하도록 하여 제 3의 신뢰기관 없이도 거래가 정상적으로 이행되도록 한다. 스마트 컨트랙트를 통해 비즈니스 로직을 구성하여 분산 애플리케이션(distributed application: DApp)을 개발 및 운영할 수 있다.

[0005] 상기 선행기술에 의하더라도 블록체인 기반 분산형 애플리케이션이 사용자 요구 트랜잭션을 처리하기 위해서는 복잡한 분산 합의 과정을 거쳐 블록이 원장에 추가되어야 하므로 트랜잭션 처리 지연 시간이 길어지는 문제점이 여전하다.

[0006] 또한 블록체인 시스템은 각 노드가 블록을 수신하고 독립적으로 관리되는 구조로, 공간 오버헤드가 노드 수에 비례적으로 증가하며, 또한 월드 스테이트(World state) 업데이트에 대한 오버헤드가 상당한 문제점이 있다.

발명의 내용

해결하려는 과제

[0007] 블록체인 기반 분산형 애플리케이션(DApp)의 성능 확장성 요구를 만족하기 위해서 다양한 노력이 시도되고 있다. 특히, 컨소시엄 블록체인 또는 허가형 블록체인 시스템은 기존 비허가형 블록체인 플랫폼의 성능 한계점을 보완하기 위해 도입되었지만, 위에서 설명한 이유로 여전히 실제 응용 프로그램에 적용하기 어려운 성능 제약점이 있다.

[0008] 상기와 같은 문제점을 해결하기 위한 본 발명의 목적은, 분산 환경에서 응용 프로그램 일관성을 유지하는 서비스인 공유 로그를 활용함으로써, 복수 개의 노드에서 스마트 컨트랙트를 병렬적으로 실행하는 새로운 블록체인 네트워크 시스템 및 그 블록체인 네트워크의 사용자 트랜잭션 처리 방법을 제안하는 것이다.

[0009] 또한 본 발명의 목적은 새로운 블록체인 네트워크 시스템을 통해 성능 확장성을 노드 개수 확장을 통해 지원하며, 공유 로그 서비스에 블록체인 정보를 기록함으로써 저장 공간 오버헤드를 감소시키고, 동시에 상태정보 유지를 효과적으로 지원하는 것이다.

[0010] 이때 공유 로그는 기본적으로 append-only 저장 구조를 채택함으로써 복수 개 노드들 간 상태정보 합의를 용이하게 지원할 수 있다.

과제의 해결 수단

[0011] 본 발명의 목적을 달성하기 위한 일 실시예에 따른 공유 로그 기반 블록체인 네트워크 시스템은, 사용자 트랜잭션 생성에 필요한 정보를 제공하는 서비스 사용자로부터 사용자 정보에 기반하여 사용자 트랜잭션을 생성하는 클라이언트 측 소프트웨어 인터페이스를 경유하여 전송되는 사용자 트랜잭션을 수신하고, 사용자 트랜잭션을 실행하는 복수 개의 스마트 컨트랙트 실행 노드들; 및 복수 개의 스마트 컨트랙트 실행 노드들에 의하여 실행되는 사용자 트랜잭션의 실행(endorsement) 결과로 생성된 커밋 레코드를 수신하고, 커밋 레코드를 공유 로그로서 저

장하는 적어도 하나 이상의 공유 로그 스토리지를 포함한다.

[0012] 복수개의 스마트 컨트랙트 실행 노드들 각각은 복수개의 스마트 컨트랙트 실행 노드들 각각에 할당되는 사용자 트랜잭션을 병렬적으로 처리할 수 있다.

[0013] 본 발명의 일 실시예에 따른 공유 로그 기반 블록체인 네트워크 시스템은, 복수개의 스마트 컨트랙트 실행 노드들로부터 커밋 레코드를 수신하고, 커밋 레코드의 유효성을 검증하고, 커밋 레코드가 유효하다고 판단되는 경우 공유 로그 스토리지 주소 토큰을 생성하여 복수개의 스마트 컨트랙트 실행 노드들로 전송하는 시퀀서 노드를 더 포함할 수 있다.

[0014] 시퀀서 노드는 복수개의 스마트 컨트랙트 실행 노드들에 의하여 병렬적으로 실행되는 사용자 트랜잭션 간 유효성 검사를 실행할 수 있고, 적어도 하나 이상의 공유 로그 스토리지에 쓰기 가능한 주소를 포함하는 공유 로그 스토리지 주소 토큰을 생성함으로써 복수개의 스마트 컨트랙트 실행 노드들 간의 사용자 트랜잭션 실행 순서에 대한 합의를 지원할 수 있다.

[0015] 시퀀서 노드는 커밋 레코드의 유효성을 검증하기 위해 커밋 레코드 내 읽기 집합에 대하여 각 개체에 대한 키-버전 쌍과 읽기 집합에 포함된 개체들의 버전 정보를 비교하는 다중 버전 충돌 체크(MVCC, Multi-version conflict check)를 수행할 수 있다.

[0016] 본 발명의 일 실시예에 따른 공유 로그 기반 블록체인 네트워크 시스템은, 적어도 하나 이상의 공유 로그 스토리지에 기록된 사용자 트랜잭션 커밋 레코드를 포함하는 로그 데이터를 공유 로그 인터페이스를 통하여 읽어들이고, 로그 데이터를 이용하여 블록을 생성하는 블록 생성 노드를 더 포함할 수 있다.

[0017] 블록 생성 노드는 블록을 생성함에 있어 기존 블록체인들과 호환되는 새로운 블록을 생성할 수 있고, 블록을 전파할 수 있다.

[0018] 블록 생성 노드는 로그 데이터에 관련되는 사용자 트랜잭션들의 순차적인 기록 및 스마트 컨트랙트 실행 과정에서 이미 합의된 순서를 이용하여 블록을 생성할 수 있다.

[0019] 복수개의 스마트 컨트랙트 실행 노드들은 공유 로그 인터페이스를 포함할 수 있다.

[0020] 복수개의 스마트 컨트랙트 실행 노드들에 의하여 생성되는 커밋 레코드를 포함하는 로그 데이터가 공유 로그 인터페이스를 경유하여 적어도 하나 이상의 공유 로그 스토리지로 전송될 수 있다.

[0021] 커밋 레코드를 포함하는 로그 데이터가 적어도 하나 이상의 공유 로그 스토리지에 저장될 수 있다.

[0022] 복수개의 스마트 컨트랙트 실행 노드들은 사용자 트랜잭션을 실행할 때 커밋 레코드를 생성하기 위하여 필요한 최신 상태정보를 획득하기 위하여 공유 로그 인터페이스를 경유하여 적어도 하나 이상의 공유 로그 스토리지로부터 최신 로그 데이터를 읽어들일 수 있다.

[0023] 적어도 하나 이상의 공유 로그 스토리지는 커밋 레코드를 포함하는 로그 데이터를 복수개의 속성으로 저장할 수 있다. 이때 복수개의 속성은 트랜잭션 타입, 클라이언트 ID, 트랜잭션 ID, 자바 프로그램 정보, 전달 인자, 및 실행 결과 중 적어도 하나 이상을 포함할 수 있다.

[0024] 본 발명의 일 실시예에 따른 공유 로그 기반 블록체인 네트워크의 사용자 트랜잭션 처리 방법은, 사용자 트랜잭션 생성에 필요한 정보를 제공하는 서비스 사용자로부터 사용자 정보에 기반하여 생성된 사용자 트랜잭션을 클라이언트 측 소프트웨어 인터페이스를 경유하여 복수개의 스마트 컨트랙트 실행 노드들 중 적어도 일부로 전송하는 단계; 복수개의 스마트 컨트랙트 실행 노드들 중 적어도 일부가 사용자 트랜잭션을 실행하는 단계; 복수개의 스마트 컨트랙트 실행 노드들 중 적어도 일부가 사용자 트랜잭션이 실행된 결과로 커밋 레코드를 생성하는 단계; 및 적어도 하나 이상의 공유 로그 스토리지가 커밋 레코드를 공유 로그로서 저장하는 단계를 포함한다.

[0025] 사용자 트랜잭션을 복수개의 스마트 컨트랙트 실행 노드들 중 적어도 일부로 전송하는 단계는, 클라이언트 측 소프트웨어 인터페이스가 복수개의 사용자 트랜잭션을 복수개의 스마트 컨트랙트 실행 노드들에 할당하는 단계를 포함할 수 있다.

[0026] 사용자 트랜잭션을 실행하는 단계는, 복수개의 스마트 컨트랙트 실행 노드들 각각이 복수개의 스마트 컨트랙트 실행 노드들 각각에 할당되는 복수개의 사용자 트랜잭션을 병렬적으로 처리할 수 있다.

[0027] 본 발명의 일 실시예에 따른 공유 로그 기반 블록체인 네트워크의 사용자 트랜잭션 처리 방법은, 커밋 레코드를 생성하는 단계의 이후 및 커밋 레코드를 공유 로그로서 저장하는 단계의 이전에, 시퀀서 노드가 복수개의 스마

트 컨트랙트 실행 노드 중 적어도 일부로부터 커밋 레코드를 수신하는 단계; 시퀀서 노드가 커밋 레코드의 유효성을 검증하는 단계; 커밋 레코드가 유효하다고 판단되는 경우 시퀀서 노드가 공유 로그 스토리지 주소 토큰을 생성하는 단계; 및 시퀀서 노드가 공유 로그 스토리지 주소 토큰을 복수개의 스마트 컨트랙트 실행 노드들 중 적어도 일부로 전송하는 단계를 더 포함할 수 있다.

[0028] 시퀀서 노드가 커밋 레코드의 유효성을 검증하는 단계에서, 복수개의 스마트 컨트랙트 실행 노드들에 의하여 병렬적으로 실행되는 사용자 트랜잭션 간 유효성 검사가 실행될 수 있다.

[0029] 공유 로그 스토리지 주소 토큰을 생성하는 단계에서, 적어도 하나 이상의 공유 로그 스토리지에 쓰기 가능한 주소를 포함하는 공유 로그 주소 토큰이 생성됨으로써 복수개의 스마트 컨트랙트 실행 노드들 간의 사용자 트랜잭션 실행 순서에 대한 합의가 지원될 수 있다.

[0030] 시퀀서 노드가 커밋 레코드의 유효성을 검증하는 단계에서, 시퀀서 노드는 커밋 레코드의 유효성을 검증하기 위해 커밋 레코드 내 읽기 집합에 대하여 각 개체에 대한 키-버전 쌍과 읽기 집합에 포함된 개체들의 버전 정보를 비교하는 다중 버전 충돌 체크(MVCC, Multi-version conflict check)를 수행할 수 있다.

[0031] 본 발명의 일 실시예에 따른 공유 로그 기반 블록체인 네트워크의 사용자 트랜잭션 처리 방법은, 블록 생성 노드가 적어도 하나 이상의 공유 로그 스토리지에 기록된 사용자 트랜잭션 커밋 레코드를 포함하는 로그 데이터를 공유 로그 인터페이스를 통하여 읽어들이는 단계; 블록 생성 노드가 로그 데이터를 이용하여 기존 블록체인들과 호환되는 새로운 블록을 생성하는 단계; 및 블록을 전파하는 단계를 더 포함할 수 있다.

[0032] 블록 생성 노드가 새로운 블록을 생성하는 단계에서, 블록 생성 노드는 로그 데이터에 관련되는 사용자 트랜잭션들의 순차적인 기록 및 스마트 컨트랙트 실행 과정에서 이미 합의된 순서를 이용하여 블록을 생성할 수 있다.

[0033] 적어도 하나 이상의 공유 로그 스토리지가 커밋 레코드를 공유 로그로서 저장하는 단계에서, 적어도 하나 이상의 공유 로그 스토리지는 커밋 레코드를 포함하는 로그 데이터를 복수개의 속성으로 저장할 수 있다. 이때 복수개의 속성은 트랜잭션 타입, 클라이언트 ID, 트랜잭션 ID, 자바 프로그램 정보, 전달 인자, 및 실행 결과 중 적어도 하나 이상을 포함할 수 있다.

발명의 효과

[0034] 본 발명의 실시예에 따르면, 공유 로그를 블록체인 시스템에 도입한 환경에서 대용량 사용자 트랜잭션을 처리할 때 사용자 트랜잭션을 복수 개 노드로 분산하여 병렬적으로 스마트 컨트랙트를 실행할 수 있도록 블록체인 네트워크 시스템을 구성함으로써, 블록체인 성능 확장성이 지원될 수 있다.

[0035] 또한, 본 발명의 실시예에 따르면, 기존 블록체인 시스템과 달리 사용자 트랜잭션이 공유 로그로써 기록되면 처리가 완료되었음을 의미하므로 기존 블록체인 시스템과 비교하여 더 빠른 처리 결정이 지원될 수 있다.

도면의 간단한 설명

[0036] 도 1은 본 발명의 일 실시예에 따른 공유 로그 기반 블록체인 네트워크 시스템을 도시하는 개념도이다.

도 2는 도 1의 스마트 컨트랙트 실행부 노드의 구조를 상세하게 도시하는 개념도이다.

도 3은 도 1의 블록 생성기 노드의 구조를 상세하게 도시하는 개념도이다.

도 4는 본 발명의 일 실시예에 따른 공유 로그 기반 블록체인 네트워크 시스템에서 이용되는 커밋 레코드 속성 구조의 일 예를 도시하는 개념도이다.

도 5는 본 발명의 일 실시예에 따른 공유 로그 기반 블록체인 네트워크의 사용자 트랜잭션 처리 방법의 개요를 도시하는 동작 흐름도이다.

도 6은 본 발명의 일 실시예에 따른 공유 로그 기반 블록체인 네트워크의 사용자 트랜잭션 처리 방법과 관련되는 공유 로그 작성의 순서 결정 및 로그 기록에 관련된 과정을 도시하는 동작 흐름도이다.

도 7은 본 발명의 일 실시예에 따른 공유 로그 기반 블록체인 네트워크 시스템의 스마트 컨트랙트 실행부 노드를 중심으로 수행되는 사용자 트랜잭션 처리 방법의 과정의 일부를 도시하는 개념도이다.

도 8은 블록 생성기 노드를 중심으로 수행되는 사용자 트랜잭션 처리 방법의 과정의 일부를 도시하는 개념도이

다.

도 9는 본 발명의 일 실시예에 따른 일반화된 블록체인 네트워크 내의 노드 또는 컴퓨팅 시스템을 도시하는 블록도이다.

발명을 실시하기 위한 구체적인 내용

[0037] 본 발명은 다양한 변경을 가할 수 있고 여러 가지 실시예를 가질 수 있는 바, 특정 실시예들을 도면에 예시하여 상세하게 설명하고자 한다. 그러나, 이는 본 발명을 특정한 실시 형태에 대해 한정하려는 것이 아니며, 본 발명의 사상 및 기술 범위에 포함되는 모든 변경, 균등물 내지 대체물을 포함하는 것으로 이해되어야 한다. 각 도면을 설명하면서 유사한 참조부호를 유사한 구성요소에 대해 사용하였다.

[0038] 제1, 제2, A, B 등의 용어는 다양한 구성요소들을 설명하는 데 사용될 수 있지만, 상기 구성요소들은 상기 용어들에 의해 한정되어서는 안 된다. 상기 용어들은 하나의 구성요소를 다른 구성요소로부터 구별하는 목적으로만 사용된다. 예를 들어, 본 발명의 권리 범위를 벗어나지 않으면서 제1 구성요소는 제2 구성요소로 명명될 수 있고, 유사하게 제2 구성요소도 제1 구성요소로 명명될 수 있다. "및/또는"이라는 용어는 복수의 관련된 기재된 항목들의 조합 또는 복수의 관련된 기재된 항목들 중의 어느 항목을 포함한다.

[0039] 어떤 구성요소가 다른 구성요소에 "연결되어" 있다거나 "접속되어" 있다고 언급된 때에는, 그 다른 구성요소에 직접적으로 연결되어 있거나 또는 접속되어 있을 수도 있지만, 중간에 다른 구성요소가 존재할 수도 있다고 이해되어야 할 것이다. 반면에, 어떤 구성요소가 다른 구성요소에 "직접 연결되어" 있다거나 "직접 접속되어" 있다고 언급된 때에는, 중간에 다른 구성요소가 존재하지 않는 것으로 이해되어야 할 것이다.

[0040] 본 출원에서 사용한 용어는 단지 특정한 실시예를 설명하기 위해 사용된 것으로, 본 발명을 한정하려는 의도가 아니다. 단수의 표현은 문맥상 명백하게 다르게 뜻하지 않는 한, 복수의 표현을 포함한다. 본 출원에서, "포함하다" 또는 "가지다" 등의 용어는 명세서상에 기재된 특징, 숫자, 단계, 동작, 구성요소, 부품 또는 이들을 조합한 것이 존재함을 지정하려는 것이지, 하나 또는 그 이상의 다른 특징들이나 숫자, 단계, 동작, 구성요소, 부품 또는 이들을 조합한 것들의 존재 또는 부가 가능성을 미리 배제하지 않는 것으로 이해되어야 한다.

[0041] 다르게 정의되지 않는 한, 기술적이거나 과학적인 용어를 포함해서 여기서 사용되는 모든 용어들은 본 발명이 속하는 기술 분야에서 통상의 지식을 가진 자에 의해 일반적으로 이해되는 것과 동일한 의미를 가지고 있다. 일반적으로 사용되는 사전에 정의되어 있는 것과 같은 용어들은 관련 기술의 문맥 상 가지는 의미와 일치하는 의미를 가지는 것으로 해석되어야 하며, 본 출원에서 명백하게 정의하지 않는 한, 이상적이거나 과도하게 형식적인 의미로 해석되지 않는다.

[0042] 종래 기술로서 본 발명의 출원일 전에 개시된 사항은 본 발명의 목적에 부합하는 범위 내에서 본 발명의 구성의 일부 또는 전부로서 포함될 수 있다. 당업자라면 종래 기술 문헌들의 내용으로부터 본 발명의 목적 및 구성과의 연관성을 자명하게 유추할 수 있을 것이므로 본 발명의 취지를 흐릴 수 있는 지나치게 자세한 설명은 생략한다.

[0043] 이하, 첨부한 도면들을 참조하여, 본 발명의 바람직한 실시예를 보다 상세하게 설명하고자 한다. 본 발명을 설명함에 있어 전체적인 이해를 용이하게 하기 위하여 도면상의 동일한 구성요소에 대해서는 동일한 참조부호를 사용하고 동일한 구성요소에 대해서 중복된 설명은 생략한다.

[0044] 도 1은 본 발명의 일 실시예에 따른 공유 로그 기반 블록체인 네트워크 시스템을 도시하는 개념도이다.

[0045] 도 1에 도시된 바와 같이, 전체 시스템은 서비스 사용자(100), 및 공유 로그 기반 블록체인 네트워크 시스템(300)을 포함한다.

[0046] 서비스 사용자(100)는 공유 로그 기반 블록체인 네트워크 시스템(300)으로부터 서비스를 제공받는 단말을 나타내며 스마트폰, 퍼스널 컴퓨터(PC) 등을 포함할 수 있다. 서비스 사용자(100)는 스마트 컨트랙트 실행(endorsement)을 위해 클라이언트 SDK(200)를 통해 공유 로그 기반 블록체인 네트워크 시스템(300) 내 스마트 컨트랙트 실행부 노드들(310)로 트랜잭션을 전송할 수 있다.

[0047] 본 발명의 일 실시예에 따른 공유 로그 기반 블록체인 네트워크 시스템(300)은, 사용자 트랜잭션 생성에 필요한 정보를 제공하는 서비스 사용자(100)로부터 사용자 정보에 기반하여 사용자 트랜잭션을 생성하는 클라이언트 측 소프트웨어 인터페이스/클라이언트 SDK(200)를 경유하여 전송되는 사용자 트랜잭션을 수신하고, 사용자 트랜잭션을 실행하는 복수 개의 스마트 컨트랙트 실행 노드들(310); 및 복수 개의 스마트 컨트랙트 실행 노드들(310)에 의하여 실행되는 사용자 트랜잭션의 실행 결과로 생성된 커밋 레코드를 수신하고, 커밋 레코드를 공유 로그

로서 저장하는 적어도 하나 이상의 공유 로그 스토리지(341)를 포함한다.

[0048] 복수개의 스마트 컨트랙트 실행 노드들(310) 각각은 복수개의 스마트 컨트랙트 실행 노드들(310) 각각에 할당되는 사용자 트랜잭션을 병렬적으로 처리할 수 있다.

[0049] 본 발명의 일 실시예에 따른 공유 로그 기반 블록체인 네트워크 시스템(300)은, 복수개의 스마트 컨트랙트 실행 노드들(310)로부터 커밋 레코드를 수신하고, 커밋 레코드의 유효성을 검증하고, 커밋 레코드가 유효하다고 판단되는 경우 공유 로그 스토리지 주소 토큰을 생성하여 복수개의 스마트 컨트랙트 실행 노드들(310)로 전송하는 시퀀서 노드(320)를 더 포함할 수 있다.

[0050] 시퀀서 노드(320)는 복수개의 스마트 컨트랙트 실행 노드들(310)에 의하여 병렬적으로 실행되는 사용자 트랜잭션 간 유효성 검사를 실행할 수 있고, 적어도 하나 이상의 공유 로그 스토리지에 쓰기 가능한 주소를 포함하는 공유 로그 스토리지 주소 토큰을 생성함으로써 복수개의 스마트 컨트랙트 실행 노드들(310) 간의 사용자 트랜잭션 실행 순서에 대한 합의를 지원할 수 있다.

[0051] 시퀀서 노드(320)는 커밋 레코드의 유효성을 검증하기 위해 커밋 레코드 내 읽기 집합에 대하여 각 개체에 대한 키-버전 쌍과 읽기 집합에 포함된 개체들의 버전 정보를 비교하는 다중 버전 충돌 체크(MVCC, Multi-version conflict check)를 수행할 수 있다.

[0052] 본 발명의 일 실시예에 따른 공유 로그 기반 블록체인 네트워크 시스템(300)은, 적어도 하나 이상의 공유 로그 스토리지(341)에 기록된 사용자 트랜잭션 커밋 레코드를 포함하는 로그 데이터를 공유 로그 인터페이스/공유 로그 API(333)를 통하여 읽어들이고, 로그 데이터를 이용하여 블록을 생성하는 블록 생성 노드(330)를 더 포함할 수 있다.

[0053] 블록 생성 노드(330)는 블록을 생성함에 있어 기존 블록체인들과 호환되는 새로운 블록을 생성할 수 있고, 블록을 전파할 수 있다. 이때 블록 생성 노드(330)는 새로운 블록을 공유 로그 기반 블록체인 네트워크(300)의 외부로 전달할 수 있다.

[0054] 블록 생성 노드(330)는 로그 데이터에 관련되는 사용자 트랜잭션들의 순차적인 기록 및 스마트 컨트랙트 실행 과정에서 이미 합의된 순서를 이용하여 블록을 생성할 수 있다.

[0055] 복수개의 스마트 컨트랙트 실행 노드들(310)은 공유 로그 인터페이스/공유 로그 API(314)를 포함할 수 있다.

[0056] 복수개의 스마트 컨트랙트 실행 노드들(310)에 의하여 생성되는 커밋 레코드를 포함하는 로그 데이터가 공유 로그 인터페이스/공유 로그 API(314)를 경유하여 적어도 하나 이상의 공유 로그 스토리지(341)로 전송될 수 있다.

[0057] 커밋 레코드를 포함하는 로그 데이터가 적어도 하나 이상의 공유 로그 스토리지(341)에 저장될 수 있다.

[0058] 복수개의 스마트 컨트랙트 실행 노드들(310)은 사용자 트랜잭션을 실행할 때 커밋 레코드를 생성하기 위하여 필요한 최신 상태정보를 획득하기 위하여 공유 로그 인터페이스/공유 로그 API(314)를 경유하여 적어도 하나 이상의 공유 로그 스토리지로(341)부터 최신 로그 데이터를 읽어들일 수 있다.

[0059] 적어도 하나 이상의 공유 로그 스토리지(341)는 커밋 레코드를 포함하는 로그 데이터를 복수개의 속성으로 저장할 수 있다. 이때 복수개의 속성은 트랜잭션 타입, 클라이언트 ID, 트랜잭션 ID, 자바 프로그램 정보, 전달 인자, 및 실행 결과 중 적어도 하나 이상을 포함할 수 있다.

[0060] 공유 로그 기반 블록체인 네트워크 시스템(300)은 스마트 컨트랙트 실행부 노드들(310), 시퀀서 노드(320), 블록 생성기 노드(330), 공유 로그 서비스 실행부 노드(340)를 포함할 수 있다. 스마트 컨트랙트 실행부 노드들(310)은 클라이언트 SDK(200)로부터 수신한 트랜잭션을 사전에 설치된 스마트 컨트랙트(315)를 공유 로그 환경에서 실행(endorsement)함으로써 처리하고, 그 결과를 포함한 커밋 레코드를 생성하고, 커밋 레코드를 이를 공유 로그 형태로 기록할 수 있다.

[0061] 클라이언트 SDK(200)는 서비스 사용자(100), 즉, 클라이언트 측에 설치되는 소프트웨어일 수 있다. 클라이언트 SDK(200)가 사용자 트랜잭션을 복수개의 스마트 컨트랙트 실행부 노드들(310)에 할당하여 전송할 수 있다. 즉, 클라이언트 SDK(200)는 복수개의 사용자 트랜잭션들을 자동으로 로드 밸런싱하는 역할을 수행할 수 있다.

[0062] 스마트 컨트랙트 실행 노드들(310)이 보유하고 있는 스마트 컨트랙트(315)는 동일한 블록체인 응용 서비스를 제공하는 경우 서로 동일할 수 있다. 다만 서로 다른 노드들의 스마트 컨트랙트(315)가 처리하는 사용자 트랜잭션들은 서로 다를 수 있다. 다수의 서비스 사용자(100)가 각각 서로 다른 트랜잭션들을 요청할 때, 이들 트랜잭션

들이 복수개의 스마트 컨트랙트 실행 노드들(310)에 분할 할당되어 병렬적으로 사용자 트랜잭션이 처리될 수 있다. 즉, 서로 다른 스마트 컨트랙트 실행 노드들(310) 간에는 서로 다른 사용자의 트랜잭션 및/또는 서로 다른 내용의 트랜잭션이 병렬적으로 실행될 수 있다.

[0063] 시퀀서 노드(320)는 복수 개의 스마트 컨트랙트 실행부 노드들(310)에서 동일한 로그 위치에 대해 트랜잭션을 기록할 때 발생하는 충돌을 제어해주기 위해 공유 로그 저장소 주소 토큰을 발행해주는 노드이다. 시퀀서 노드(320)는 공유 로그의 마지막 로그 주소를 저장하는 카운터와 각 개체에 대한 키-버전 쌍을 보유할 수 있다. 이때 키는 개체의 키를 의미하고 버전은 개체를 마지막으로 업데이트한 공유 로그 주소 정보를 의미할 수 있다. 시퀀서 노드(320)는 이 두 가지 데이터 구조를 기반으로 다음의 기능들을 수행한다. 첫째, 시퀀서 노드(320)는 스마트 컨트랙트 실행부 노드들(310)이 제출한 커밋 레코드의 읽기 집합에 대해 MVCC 읽기 충돌 확인(read conflict check)을 수행할 수 있다. 둘째, 읽기 집합에 대해 충돌이 없는 커밋 레코드에 대해 공유 로그 저장소 주소 토큰을 발급할 수 있다.

[0064] 시퀀서 노드(320)는 트랜잭션이 실행된 후 공유 로그에 저장되는 순서를 결정하는 노드이다. 즉, 공유 로그에 트랜잭션 정보 및 실행 결과 데이터가 저장되는 순서가 시퀀서 노드(320)에 의하여 결정될 수 있다. 상황에 따라서는 시퀀서 노드(320)가 구성되지 않더라도, 사용자 트랜잭션 실행이 정상적으로 진행될 수 있으며, 시퀀서 노드(320) 은 트랜잭션 실행 성공율을 높임으로써, 결과적으로 성능을 높이는데 영향을 줄 수 있다.

[0065] 본 발명의 실시예에 따라서는, 저장되는 데이터(공유 로그)의 integrity check를 시퀀서 노드(320)가 실행할 수도 있다. 이때 시퀀서 노드(320)가 MVCC를 수행함으로써 공유 로그의 integrity check 역할을 수행할 수 있다.

[0066] 블록 생성기 노드(330)는 공유 로그 서비스 실행부(340)의 공유 로그 저장소 /스토리지(341)에 기록되어 있는 트랜잭션 로그 데이터를 참조하여 이를 포함한 블록을 생성하고 이를 블록체인 형태로 기록할 수 있으며, 블록체인 정보를 외부 노드와 연계할 수 있다.

[0067] 본 발명의 일 실시예에 따른 블록체인 네트워크 시스템(300)은 스마트 컨트랙트(315)의 배포 및 실행, 사용자 트랜잭션의 처리 결과 등이 공유 로그 저장소/스토리지(341)에 실행 순서에 따라 기록되어 있으며, 외부 연계를 위해 블록의 형태로 구성하여 블록체인을 생성함으로써, 일종의 경량 합의 과정을 가지는 고성능 블록체인 네트워크로 간주할 수 있다.

[0068] 블록 생성기 노드(330)는 공유 로그(342)를 이용하여 블록을 생성하고 블록체인을 형성할 수 있다. 블록체인으로 형성된 정보는 변경이 어려우며 비합의된 변경 시 발견될 수 있다.

[0069] 본 발명의 실시예에서는 블록 생성기 노드(330)는 공유 로그 저장소(341)과만 통신할 수 있다. 블록 생성기 노드(330)는 다른 엔티티/개체/노드들과는 독립적으로 동작할 수 있다.

[0070] 블록 생성기 노드(330)에서 반드시 블록의 유효성 등에 대한 검사 또는 검증이 이루어질 필요는 없다.

[0071] 공유 로그 서비스 실행부(340)의 로그 데이터는 모든 스마트 컨트랙트 실행부 노드들(310)에서 공유될 수 있다. 공유 로그 서비스 실행부(340)에서는 로그의 순서가 정해지므로 트랜잭션 순서의 일관성이 유지될 수 있다. 따라서 스마트 컨트랙트 실행부 노드들(310)에서 실행한 스마트 컨트랙트(315)의 트랜잭션 정보가 로그로서 기록될 수 있다.

[0072] 공유 로그 스토리지(341)는 독립적인 서버로 구현될 수도 있고, 클라우드 서버의 형태로 구현될 수도 있다.

[0073] 공유 로그 스토리지(341)는 하나 또는 그 이상의 서버와 클라우드 서버가 혼재되어 구현될 수도 있다.

[0074] 공유 로그 스토리지(341)에 공유 로그가 저장될 때 데이터 integrity 체크는 이루어진 것으로 볼 수 있다. 데이터 integrity 체크의 역할의 일부를 시퀀서 노드(320)가 분담하여 간접적으로 지원할 수 있다.

[0075] 공유 로그(342)가 존재하므로, 스마트 컨트랙트(315) 내의 object들의 상태가 변경되더라도 이러한 변경 내용이 공유 로그(342)에 의하여 각각의 스마트 컨트랙트 실행 노드들(310) 간에 공유되고 업데이트될 수 있으며, 동기화될 수 있다.

[0076] 본 발명의 일 실시예에서는 스마트 컨트랙트(315) 자체는 트랜잭션의 처리/실행에 대하여 변동이 없고 object들의 사용 상태가 변경될 수 있다. 이러한 object들의 사용 상태의 변경이 공유 로그(342)를 통하여 여러 노드들에 의하여 공유될 수 있다.

[0077] 본 발명의 다른 일 실시예에서는 스마트 컨트랙트(315) 자체가 트랜잭션의 처리/실행에 의하여 변동될 수도 있

다. 이 경우에도 스마트 컨트랙트(315)의 변경 내역이 공유 로그(342)를 통하여 여러 노드들 간에 공유될 수 있다.

[0078] 이때 트랜잭션의 실행에 따른 스마트 컨트랙트(315)의 변경이 공유 로그(342)를 통하여 블록체인 네트워크(300) 내의 개체들에게 공유되고 스마트 컨트랙트(315) 자체의 변동 사항이 각 실행 노드들(310)에 의하여 업데이트될 수도 있다.

[0079] 일반적인 블록체인 시스템의 두가지 주요 과제는 개별적으로 상태를 저장하고 있는 노드들에서 실행되는 분산 응용 프로그램인 Smart contract 실행(endorsement/execution) 결과의 일치와 성능 확장성이 있다. 블록체인 시스템은 종종 애플리케이션 상태(State)를 공유하는 여러 노드로 구성될 수 있다. 그러나 각 노드들이 블록을 통해 업데이트되는 독립적인 스토리지를 유지하는 구조에서는 노드들의 상태를 일관되게 유지하며 성능 확장성을 달성하기에 어려움이 있다. 특히 블록체인 시스템은 사용자 트랜잭션 처리 성능이 중앙 집중식 서비스에 비해 현저히 낮다. 따라서 실제 비즈니스 애플리케이션에 적용하기 위해서 블록체인 시스템도 중앙 집중식 서비스에 준하는 성능 요구사항을 만족할 필요가 있다.

[0080] 탈중앙화 거래소 서비스 뿐 아니라, 최근 메타버스와 대체불가 토큰 NFT 연계, 탈중앙화 금융 (Decentralized Finance DeFi), P2E (Play-to-Earn) 등의 블록체인 게임 등에 대한 사용자들의 관심이 증대됨에 따라, 블록체인 성능 확장성의 중요성은 지속적으로 증가하고 있다. 즉, 블록체인 시스템으로서의 특성 및 장점을 유지하면서도, 블록체인 노드들의 추가로 인한 성능의 확장이 가능하도록 설계된 블록체인 시스템에 대한 수요가 증대되는 상황이다.

[0081] 본 발명의 실시예에 따르면 공유 로그 기반 고확장성 블록체인 네트워크 시스템 및 그 네트워크 내의 사용자 트랜잭션 처리 방법이 개시된다. 본 발명의 실시예에 따르면 공유 로그 환경에서의 복수 개 노드에 의해 트랜잭션을 병렬적으로 처리함으로써 블록체인 성능 확장성이 지원될 수 있다.

[0082] 공유 로그(shared log) 방식은 분산 환경에서 응용 프로그램 일관성을 유지하는 한편 높은 장애 허용(fault-tolerant)을 제공하기 위해 분산 데이터 저장소에 사용되었으며 이후 다양한 분산 시스템에서 활용되고 있는 프로토콜이다. 공유 로그 방식은 한 시스템 내에 있는 복수의 클라이언트들이 동시에 로그에 액세스하고 기록할 수 있도록 기능을 제공한다. 어떤 경우에는 공유 로그(342)는 분석 또는 장애 복구를 위해 영구적으로 저장되어야 하며, 시스템이 원활하게 작동되기 위해 공유 로그(342)가 구현되는 방식은 시스템 요구사항에 따라 달라질 수 있다.

[0083] 공유 로그(342)는 장애 허용 합의(fault-tolerant consensus)를 공유 로그 서비스 실행부(340)에서 내부적으로 처리할 수 있는 API를 제공하여 분산 애플리케이션에 적용될 때 장애 허용 합의 처리에 대한 부담을 줄일 수 있다.

[0084] 공유 로그 서비스 실행부(340)는 분산 애플리케이션의 요구 사항에 따라 스토리지 서버를 추가하거나 제거할 수 있으며, 각각의 서버에 저장된 전체 로그에 대하여 순서를 유지할 수 있다. 따라서, 공유 로그 방식은 사용함에 있어 단순하지만 장애 허용 시스템을 효율적으로 구축할 수 있도록 지원해주는 프로토콜이다.

[0085] 본 발명의 일 실시예에 따른 공유 로그 기반 고확장성 블록체인 네트워크 시스템(300)의 구조는, 사용자 트랜잭션을 생성하는 서비스 사용자(100), 서비스 사용자(100) 측에서 스마트 컨트랙트 실행부(Smart Contract Executor) 노드들(310)과 통신할 수 있도록 인터페이스를 제공하는 클라이언트 SDK(200), 사용자 트랜잭션을 수신하고 처리하는 스마트 컨트랙트 실행부 노드들(310), 사용자 트랜잭션 처리 결과를 로그 형태로 기록하는 공유 로그 서비스 실행부(Shared-log Service)(340), 그리고 병렬적으로 실행되는 복수 개의 스마트 컨트랙트(315)에 대응하는 복수 개의 스마트 컨트랙트 실행부 노드들(310)에서 접근하는 스마트 컨트랙트(315)의 개체(Object)들에 대한 상태정보 일관성과 트랜잭션 실행 순서에 대한 합의를 수행하는 시퀀서(Sequencer) 노드(320), 그리고 기존 블록체인(예: 이더리움, 하이퍼레저 패브릭 등)과의 호환성을 위해 공유 로그 저장소(341)에 기록된 트랜잭션 정보를 기반으로 블록을 생성하고 전파하는 블록 생성기(Block generator) 노드(330) 등 총 6개 구성요소를 포함할 수 있다. 본 발명의 실시예에 따른 공유 로그 기반 고확장성 블록체인 시스템은 스마트 컨트랙트를 분산 환경에서 병렬로 실행하고 블록을 생성하기 위한 다양한 방법을 포함할 수 있다.

[0086] 또한, 시퀀서(Sequencer) 노드(320)는 복수 개의 스마트 컨트랙트 실행부 노드들(310)에서 동일한 공유 로그 위치에 대해 트랜잭션을 기록할 때 발생하는 충돌을 제어해주기 위해 공유 로그 저장소 주소 토큰을 발행해주는 노드이다. 공유 로그 서비스 실행부(340)는 사용자 트랜잭션 처리 결과를 로그로 기록하고 개체 저장소 역할을 할 수 있다. 더욱 구체적으로는, 독립적인 로컬 상태 저장소를 각각의 스마트 컨트랙트 실행부 노드들(310)이

유지하는 대신 공유 로그(342)를 분산 개체 저장소로 활용하여 모든 스마트 컨트랙트 실행부 노드들(310)에 동일한 개체 상태 뷰(in-memory view)를 제공할 수 있다. 서비스 사용자(100)는 블록체인 서비스를 이용하기 위해 클라이언트 SDK(200)를 통해 스마트 컨트랙트 실행부 노드들(310)과 통신할 수 있다.

[0087] 개체 저장소는 개체의 상태 정보를 저장하는 저장소(스토리지)로서, 개체의 상태 정보는 스마트 컨트랙트와 관련된 개체의 버전, 업데이트 상태 정보를 포함할 수 있다.

[0088] 본 발명의 일 실시예에 따르면, 서비스 사용자(100)가 전송하는 트랜잭션을 공유 로그로써 기록하여 관리함으로써 성능 확장성을 보장하는 기법이 개시된다. 공유 로그 기반 고확장성 블록체인을 지원하는 방법은, 서비스 사용자(100)가 트랜잭션을 스마트 컨트랙트 실행부 노드들(310)로 전송하는 단계와, 상기 트랜잭션을 스마트 컨트랙트 실행부 노드들(310)이 공유 로그 환경에서 실행하는 단계와, 상기 트랜잭션 실행에 대한 결과를 시퀀서 노드(320)에서 검증하고 토큰을 발행하는 단계와, 상기 트랜잭션 실행에 대한 정보를 공유 로그 서비스 실행부(340)에 기록하는 단계와, 공유 로그 서비스 실행부(340)에 기록된 로그 데이터를 기반으로 블록 생성기 노드(330)에서 블록을 생성하는 단계를 포함할 수 있다.

[0089] 또한, 상기 공유 로그로 기록되는 커밋 레코드의 속성으로 트랜잭션 타입, 클라이언트 ID, 트랜잭션 ID, 자바 프로그램 정보, 전달 인자, 및 실행 결과 등을 포함할 수 있다. 다른 실시예에 따르면, 상기 커밋 레코드(commit record)의 속성은 필요에 따라 추가적인 속성이 정의되어 포함될 수 있다.

[0090] 도 2는 도 1의 스마트 컨트랙트 실행부 노드(310)의 구조를 상세하게 도시하는 개념도이다.

[0091] 도 2에 도시된 바와 같이, 스마트 컨트랙트 실행부 노드(310)는 API 게이트웨이(311)와 스마트 컨트랙트 실행 모듈(312), 스마트 컨트랙트 스터브(313), 공유 로그 API(314), 그리고 스마트 컨트랙트(315)를 포함할 수 있으며, 스마트 컨트랙트(315)가 공유 로그 환경에서 실행되도록 지원할 수 있다.

[0092] 스마트 컨트랙트 실행부 노드(310)는 사용자 트랜잭션을 처리할 수 있다. 스마트 컨트랙트 실행부 노드(310)는 5개의 하위 모듈을 포함할 수 있으며, 5개의 하위 모듈은 서비스 사용자(100)에 단일 엔드 포인트를 제공하고 요청에 따라 적절한 서비스 라우팅하는 API 게이트웨이(API gateway)(311), 스마트 컨트랙트 트랜잭션을 실행하는 스마트 컨트랙트 실행 모듈(312), 응용 프로그램 논리를 포함하고 있는 스마트 컨트랙트(315), 스마트 컨트랙트(315) 측에서 공유 로그 서비스 실행부(340)와 통신할 수 있도록 지원하는 스마트 컨트랙트 스터브(smart contract stub)(313), 스마트 컨트랙트 실행부 노드(310) 내부의 다른 하위 모듈이 공유 로그 서비스 실행부(340)와 통신할 수 있도록 인터페이스를 제공하는 공유 로그 API(shared-log API)(314)를 포함할 수 있다.

[0093] API 게이트웨이(311)는 서비스 사용자(100)의 트랜잭션 요청을 수신하고 요청에 따라 서비스를 처리할 수 있다. API 게이트웨이(311)는 서비스 사용자(100)에게 단일 엔드포인트를 제공하고 서비스 사용자(100)로부터 요청을 받으면 요청의 유효성을 검사하고 적절한 서비스로 라우팅하고 서비스 사용자(100)에게 요청에 대한 응답을 반환할 수 있다.

[0094] 스마트 컨트랙트 실행 모듈(312)은 API 게이트웨이(311)로부터 요청을 수신하면 서비스 사용자(100)가 요청한 스마트 컨트랙트(315)에 대한 로직을 실행하며, 실행 결과(읽기-쓰기 집합)와 트랜잭션 정보(트랜잭션 ID, 서비스 사용자(100) ID, etc.)를 포함하는 커밋 레코드(로그 엔트리)를 생성하고, 시퀀서 노드(320)로 이를 전송하여 공유 로그 저장소 주소 토큰 발급을 요청할 수 있다. 시퀀서 노드(320)로부터 토큰을 성공적으로 발급받으면 공유 로그 API(314)를 통해 커밋 레코드를 공유 로그 서비스 실행부(340)에 기록할 수 있다.

[0095] 스마트 컨트랙트 스터브(313)는 스마트 컨트랙트(315) 측에서 개체 상태의 액세스 및 업데이트 요청을 지원하는 모듈로, 공유 로그 서비스 실행부(340)와 통신하기 위한 인터페이스를 포함할 수 있다. 스마트 컨트랙트(315)에서 개체 상태 읽기 및 업데이트 작업은 공유 로그 서비스 실행부(340)에서 유지되는 저장소(341)를 통해 지원될 수 있다. 스마트 컨트랙트(315)는 프로그램 코드 내에 로컬 상태를 유지하지 않으며, 공유 로그 서비스 실행부(340)에 저장된 개체 상태만을 참조할 수 있다. 스마트 컨트랙트 스터브(313)는 공유 로그 API(314)를 통해 공유 로그 서비스 실행부(340)의 개체 상태에 액세스할 수 있고 읽기 집합을 생성할 수 있다. 이때 업데이트 요청의 경우 단순히 개체 업데이트를 위한 쓰기 집합이 생성될 수 있다.

[0096] 공유 로그 API(314)는 공유 로그 클라이언트(스마트 컨트랙트 실행부 노드(310))가 공유 로그 서비스 실행부(340)와 통신할 수 있도록 인터페이스를 제공하는 API 집합을 의미할 수 있다. 따라서, 개체 저장소로 공유 로그 서비스 실행부(340)와 상호 작용하기 위해 스마트 컨트랙트 실행부 노드(310)는 공유 로그 API(314)를 로드할 수 있다. 공유 로그 API(314)는 각 스마트 컨트랙트 실행부 노드(310)에게 개체 상태에 대한 메모리 내 로컬 뷰(in-memory local view)를 제공할 수 있으며, 제공된 개체 상태에 대한 메모리 내 로컬 뷰는 공유 로그 서비

스 실행부(340)를 통해 복수개의 스마트 컨트랙트 실행부 노드들(310) 간에 동기화될 수 있다.

[0097] 스마트 컨트랙트(315)는 응용 프로그램 논리를 구현하는 프로그램 코드로서, 모든 비즈니스 로직이 스마트 컨트랙트(315) 내부에 있기 때문에 본 발명의 실시예에 있어서 가장 주요한 구성요소 중 하나이다. 서비스 사용자(100)는 스마트 컨트랙트 실행부 노드(310)를 통해 스마트 컨트랙트(315)와 상호 작용을 할 수 있다. 따라서 트랜잭션을 수행하는 모든 스마트 컨트랙트 실행부 노드(310)에는 동일한 스마트 컨트랙트(315)가 설치되어 있을 것을 전제로 한다.

[0098] 도 3은 도 1의 블록 생성기 노드(330)의 구조를 상세하게 도시하는 개념도이다.

[0099] 또한, 블록 생성기 노드(330)는 3개의 하위 모듈을 포함할 수 있으며, 3개의 하위 모듈은 공유 로그 서비스 실행부(340)에서 트랜잭션 로그를 읽어와 블록에 포함되는 트랜잭션 엔트리 형태로 변환하여 블록 생성 모듈(332)로 전달하는 리더(reader) 모듈(331), 그리고 리더 모듈(331)로부터 수신한 트랜잭션 엔트리들을 블록으로 생성하는 블록 생성(block generation) 모듈(332), 마지막으로 공유 로그 서비스 실행부(340)로부터 공유 로그 데이터를 참조하기 위한 인터페이스인 공유 로그 API(333)을 포함할 수 있다.

[0100] 블록 생성기 노드(330)는 리더(reader) 모듈(331), 블록 생성 모듈(332), 및 공유 로그 API(333) 외에 블록 저장소(334)를 더 포함할 수 있다. 리더 모듈(331)은 공유 로그 서비스 실행부(340)에 기록된 트랜잭션 로그를 읽어와 블록에 포함될 트랜잭션 엔트리 형태로 변환하여 블록 생성 모듈(332)로 전달할 수 있다. 블록 생성 모듈(332)은 리더 모듈(331)로부터 수신한 트랜잭션 엔트리들을 블록으로 생성할 수 있다. 블록 생성 과정에서 공유 로그 서비스 실행부(340)에 기록된 로그 데이터는 스마트 컨트랙트 실행부 노드(310)에서 발생한 트랜잭션을 이미 순차적으로 기록하였기 때문에 순서에 대한 합의는 이루어진 상태로 해석될 수 있다. 블록 생성기 노드(330)는 스마트 컨트랙트(315) 실행에 관여하지 않고 트랜잭션의 실행 결과를 포함하는 로그 데이터만을 필요로 한다. 보다 구체적으로는, 블록 생성기 노드(330)에서 공유 로그 API(333)는 공유 로그 서비스 실행부(340)에 기록된 로그 데이터를 참조하기 위해 필요하다.

[0101] 도 4는 본 발명의 일 실시예에 따른 공유 로그 기반 블록체인 네트워크 시스템에서 이용되는 커밋 레코드 속성 구조의 일 예를 도시하는 개념도이다.

[0102] 커밋 레코드는 한 트랜잭션에서 발생하는 변경 사항을 모두 기록하고 최종적으로 공유 로그 서비스 실행부(340)에 기록되는 로그 엔트리이다. 커밋 레코드는 시스템에서 발생한 트랜잭션들의 순서를 나타내므로 원자성(Atomicity)을 보장해야 하며 이러한 원자성을 제공하기 위해서 커밋 레코드는 트랜잭션의 메타데이터와 실행 결과(읽기-쓰기 집합)를 포함해야 한다. 원자성 외에 일관성(Consistency), 고립성(Isolation), 지속성(Durability)를 포함하여 ACID를 데이터베이스의 트랜잭션이 안전하게 수행되기 위한 필수적인 성질로 간주할 수도 있다. 공유 로그 서비스 실행부(340)의 실시예에 따라서 기본적으로 커밋 레코드에 들어가는 속성들은 상이할 수 있다. 공유 로그 기반 블록체인 네트워크(300)에서 공유 로그 서비스 실행부(340)의 로그 데이터를 참조하여 블록을 생성하기 위해서는 블록 생성에 요구되는 속성들을 커밋 레코드에 기록해야 한다. 따라서, 공유 로그 기반 블록체인 네트워크(300)에서는 커밋 레코드의 속성을 정의할 수 있다. 도 4에 도시된 바와 같이, 커밋 레코드의 속성은 공유 로그 주소, 트랜잭션 타입, 클라이언트 ID, 트랜잭션 ID, 스마트 컨트랙트 이름 및 버전, 스마트 컨트랙트 함수 및 그 인자 정보, 타임스탬프, 및 읽기-쓰기 집합 중 적어도 하나 이상을 포함할 수 있다.

[0103] 도 5는 본 발명의 일 실시예에 따른 공유 로그 기반 블록체인 네트워크(300)의 사용자 트랜잭션 처리 방법의 개요를 도시하는 동작 흐름도이다.

[0104] 본 발명의 일 실시예에 따른 사용자 트랜잭션 처리 방법은 공유 로그 기반 블록체인 네트워크(300) 내의 각 기능을 분담하는 노드들에 의하여 실행될 수 있다. 각 노드들은 내부의 구성요소와 전자적으로 연결되는 프로세서(processor), 및 명령어를 저장하는 메모리(memory)를 포함하는 컴퓨팅 시스템으로서 구현될 수 있다. 사용자 트랜잭션 처리 방법은 이러한 메모리 및 프로세서를 포함하는 컴퓨팅 시스템에 의하여 실행될 수 있다.

[0105] 본 발명의 일 실시예에 따른 공유 로그 기반 블록체인 네트워크(300)의 사용자 트랜잭션 처리 방법은, 사용자 트랜잭션 생성에 필요한 정보를 제공하는 서비스 사용자(100)로부터 사용자 정보에 기반하여 생성된 사용자 트랜잭션을 클라이언트 측 소프트웨어 인터페이스를 경유하여 복수개의 스마트 컨트랙트 실행 노드들(310) 중 적어도 일부로 전송하는 단계(S510); 복수개의 스마트 컨트랙트 실행 노드들(310) 중 적어도 일부가 사용자 트랜잭션을 실행하는 단계(S530); 복수개의 스마트 컨트랙트 실행 노드들(310) 중 적어도 일부가 사용자 트랜잭션이 실행된 결과로 커밋 레코드를 생성하는 단계(S550); 및 적어도 하나 이상의 공유 로그 스토리지(341)가 커밋 레

코드를 공유 로그(342)로서 저장하는 단계(S570)를 포함한다.

[0106] 사용자 트랜잭션을 복수개의 스마트 컨트랙트 실행 노드들(310) 중 적어도 일부로 전송하는 단계(S510)는, 클라이언트 측 소프트웨어 인터페이스가 복수개의 사용자 트랜잭션을 복수개의 스마트 컨트랙트 실행 노드들(310)에 할당하는 단계를 포함할 수 있다.

[0107] 사용자 트랜잭션을 실행하는 단계(S530)에서는, 복수개의 스마트 컨트랙트 실행 노드들(310) 각각이 복수개의 스마트 컨트랙트 실행 노드들(310) 각각에 할당되는 복수개의 사용자 트랜잭션을 병렬적으로 처리할 수 있다.

[0108] 본 발명의 일 실시예에 따른 공유 로그 기반 블록체인 네트워크(300)의 사용자 트랜잭션 처리 방법은, 커밋 레코드를 생성하는 단계(S550)의 이후 및 커밋 레코드를 공유 로그(342)로서 저장하는 단계(S570)의 이전에, 시퀀서 노드(320)가 복수개의 스마트 컨트랙트 실행 노드(310) 중 적어도 일부로부터 커밋 레코드를 수신하는 단계; 시퀀서 노드(320)가 커밋 레코드의 유효성을 검증하는 단계; 커밋 레코드가 유효하다고 판단되는 경우 시퀀서 노드(320)가 공유 로그 스토리지 주소 토큰을 생성하는 단계; 및 시퀀서 노드(320)가 공유 로그 스토리지 주소 토큰을 복수개의 스마트 컨트랙트 실행 노드들(310) 중 적어도 일부로 전송하는 단계를 더 포함할 수 있다.

[0109] 시퀀서 노드(320)가 커밋 레코드의 유효성을 검증하는 단계에서, 복수개의 스마트 컨트랙트 실행 노드들(310)에 의하여 병렬적으로 실행되는 사용자 트랜잭션 간 유효성 검사가 실행될 수 있다.

[0110] 공유 로그 스토리지 주소 토큰을 생성하는 단계에서, 적어도 하나 이상의 공유 로그 스토리지에 쓰기 가능한 주소를 포함하는 공유 로그 주소 토큰이 생성됨으로써 복수개의 스마트 컨트랙트 실행 노드들(310) 간의 사용자 트랜잭션 실행 순서에 대한 합의가 지원될 수 있다.

[0111] 시퀀서 노드(320)가 커밋 레코드의 유효성을 검증하는 단계에서, 시퀀서 노드(320)는 커밋 레코드의 유효성을 검증하기 위해 커밋 레코드 내 읽기 집합에 대하여 각 개체에 대한 키-버전 쌍과 읽기 집합에 포함된 개체들의 버전 정보를 비교하는 다중 버전 충돌 체크(MVCC, Multi-version conflict check)를 수행할 수 있다.

[0112] 본 발명의 일 실시예에 따른 공유 로그 기반 블록체인 네트워크(300)의 사용자 트랜잭션 처리 방법은, 블록 생성 노드(330)가 적어도 하나 이상의 공유 로그 스토리지(341)에 기록된 사용자 트랜잭션 커밋 레코드를 포함하는 로그 데이터를 공유 로그 인터페이스(333)를 통하여 읽어들이는 단계; 블록 생성 노드(330)가 로그 데이터를 이용하여 기존 블록체인들과 호환되는 새로운 블록을 생성하는 단계; 및 블록을 전파하는 단계를 더 포함할 수 있다. 이때 블록 생성 노드(330)는 새로운 블록을 공유 로그 기반 블록체인 네트워크(300)의 외부로 전달할 수 있다.

[0113] 블록 생성 노드(330)가 새로운 블록을 생성하는 단계에서, 블록 생성 노드(330)는 로그 데이터에 관련되는 사용자 트랜잭션들의 순차적인 기록 및 스마트 컨트랙트 실행 과정에서 이미 합의된 순서를 이용하여 블록을 생성할 수 있다.

[0114] 적어도 하나 이상의 공유 로그 스토리지(341)가 커밋 레코드를 공유 로그로서 저장하는 단계에서, 적어도 하나 이상의 공유 로그 스토리지(341)는 커밋 레코드를 포함하는 로그 데이터를 복수개의 속성으로 저장할 수 있다. 이때 복수개의 속성은 트랜잭션 타입, 클라이언트 ID, 트랜잭션 ID, 자바 프로그램 정보, 전달 인자, 및 실행 결과 중 적어도 하나 이상을 포함할 수 있다.

[0115] 도 6은 본 발명의 일 실시예에 따른 공유 로그 기반 블록체인 네트워크의 사용자 트랜잭션 처리 방법과 관련되는 공유 로그 작성의 순서 결정 및 로그 기록에 관련된 과정을 도시하는 동작 흐름도이다.

[0116] 스마트 컨트랙트 실행부 노드(310)의 스마트 컨트랙트 실행 모듈(312)에서 스마트 컨트랙트 실행을 마치면 커밋 레코드를 생성할 수 있다(S550). 스마트 컨트랙트 실행 모듈(312)은 상기 커밋 레코드를 시퀀서 노드(320)로 전달하며 공유 로그 토큰 발급을 요청할 수 있다(S610). 시퀀서 노드(320)는 스마트 컨트랙트 실행 모듈(312)로부터 공유 로그 토큰 발급 요청을 수신하면(S610) 커밋 레코드 내 읽기 집합에 대해 MVCC(Multi-version conflict check) 읽기 충돌을 확인/검증할 수 있다(S620). 단계(S620)에서 시퀀서 노드(320)는 각 개체에 대한 키-버전 쌍을 가지고 있어 키-버전 쌍과 읽기 집합에 포함된 개체들의 버전 정보를 비교하여 충돌 여부를 확인/검증할 수 있다. 읽기 집합에 대한 개체 상태 충돌 확인이 완료되면(S620) 시퀀서 노드(320)는 커밋 레코드에 대해 공유 로그 저장소 주소 토큰을 발급할 수 있다(S630). 만일, 충돌이 발생하면 토큰을 발급하지 않고 트랜잭션은 중단될 수 있다. 이와 같은 과정을 거쳐 시퀀스 노드(320)에 의하여 공유 로그(342)가 작성되는 순서가 결정될 수 있다.

[0117] 마지막으로, 토큰을 발급받은(S630) 스마트 컨트랙트 실행 모듈(312)은 해당 토큰의 공유 로그 위치에 커밋 레

코드를 기록하는 요청(S640) 또는 트랜잭션 로그를 추가하는 요청(S642)을 공유 로그 서비스 실행부(340)로 전송할 수 있다. 요청을 받은(S640, S642) 공유 로그 서비스 실행부(340)는 별도의 검증 과정 없이 토큰에 적힌 주소 위치에 커밋 레코드를 기록하거나(S570) 또는 추가되는 트랜잭션 로그를 기록할 수 있다(S572). 커밋 레코드 또는 추가 트랜잭션 로그가 기록 완료되면(S570, S572), 공유 로그 서비스 실행부(340)로부터 스마트 컨트랙트 실행부 노드(310)로 커밋 레코드 기록 완료(S650) 또는 트랜잭션 로그 추가 완료(S652)를 나타내는 응답이 전달될 수 있다. 본 발명의 실시예에 따라서는, 이와 같은 과정을 거쳐 시퀀스 노드(320)에 의하여 결정되는 공유 로그(342)가 작성되는 순서가, 커밋 레코드, 공유 로그(342), 및 트랜잭션 로그 추가의 기록 순서에 영향을 미칠 뿐 아니라, 스마트 컨트랙트(315)의 실행 순서, 및 사용자 트랜잭션의 처리 순서에도 영향을 미칠 수 있다.

[0118] 도 7은 본 발명의 일 실시예에 따른 공유 로그 기반 블록체인 네트워크 시스템(300)의 스마트 컨트랙트 실행부 노드(310)를 중심으로 수행되는 사용자 트랜잭션 처리 방법의 과정의 일부를 도시하는 개념도이다.

[0119] 서비스 사용자(100)는 클라이언트 SDK(200)를 경유하여 트랜잭션을 스마트 컨트랙트 실행부 노드(310)로 요청/제출할 수 있다(S510). 스마트 컨트랙트 실행부 노드(310)에서는 API 게이트웨이(311)가 트랜잭션 요청을 스마트 컨트랙트 실행 모듈(312)로 전달할 수 있다(S710). 스마트 컨트랙트 실행 모듈(312)은 지정된 스마트 컨트랙트(315)를 호출할 수 있고, 공유 로그 서비스 실행부(340)와 연동하여 트랜잭션이 실행되도록 스마트 컨트랙트(315)의 실행을 요청할 수 있다(S720).

[0120] 스마트 컨트랙트(315)의 실행이 공유 로그 서비스 실행부(340)와 연동되어 이루어지는 과정은, 스마트 컨트랙트 스터브(313)에 의하여 공유 로그 API(314)로 트랜잭션 또는 스마트 컨트랙트(315)와 관련되는 개체의 상태가 요청되고(S730), 공유 로그 스토리지(341)로부터 개체의 상태가 수신되어 공유 로그 API(314)를 경유하여 개체 상태가 스마트 컨트랙트 스터브(313)로 응답됨으로써(S732) 수행될 수 있다.

[0121] 스마트 컨트랙트(315) 실행 결과로 읽기-쓰기 집합이 생성될 수 있으며, 읽기 집합에는 스마트 컨트랙트(315)에서 참조한 모든 개체 상태정보가 포함될 수 있고 쓰기 집합에는 실행 중에 생성된 모든 개체 상태 업데이트 정보가 포함될 수 있다. 스마트 컨트랙트(315) 실행 결과가 스마트 컨트랙트 실행 모듈(312)로 전달될 수 있다(S740).

[0122] 스마트 컨트랙트 실행 모듈(312)은 트랜잭션 정보와 읽기-쓰기 집합이 포함된 커밋 레코드를 생성하고 공유 로그 토큰을 시퀀서 노드(320)로 요청할 수 있다(S610). 이때 커밋 레코드가 시퀀서 노드(320)로 함께 전송될 수도 있다. 시퀀서 노드(320)는 커밋 레코드를 수신하면 읽기 집합에 포함된 개체에 대한 업데이트 여부를 확인하고 이를 제출한 스마트 컨트랙트 실행 모듈(312)로 공유 로그 저장소 주소 토큰을 발행/응답할 수 있다(S630). 토큰을 받은 스마트 컨트랙트 실행 모듈(312)은 토큰에 기록된 주소에 커밋 레코드를 추가할 수 있다.

[0123] 토큰에 기록된 주소에 커밋 레코드가 추가되는 과정은, 스마트 컨트랙트 실행 모듈(312)로부터 공유 로그 API(314)로 로그 기록이 요청되고(S750), 공유 로그 API(314)를 경유하여 공유 로그 스토리지(341)로 로그 기록이 요청되고(S640), 공유 로그(342)가 성공적으로 기록/업데이트되면(S570, S572) 공유 로그 스토리지(341)가 로그 기록 완료되었음을 공유 로그 API(314)로 응답할 수 있다(S650). 이때 단계(S640)는 도 6의 단계(S642)를 포함할 수 있고, 단계(S650)는 도 6의 단계(S652)를 포함할 수 있다.

[0124] 공유 로그 API(314)에 의하여 로그 기록/추가 완료되었음이 스마트 컨트랙트 실행 모듈(312)로 응답되면(S752)는 스마트 컨트랙트 실행 모듈(312)는 로그가 성공적으로 기록 또는 추가(업데이트)되었음을 인지할 수 있다. 로그가 성공적으로 기록/추가되었음이 인지되면 스마트 컨트랙트 실행 모듈(312)는 트랜잭션 실행 결과를 API 게이트웨이(311)로 응답하고(S760), 스마트 컨트랙트 실행부 노드(310)는 API 게이트웨이(311)를 경유하여 트랜잭션 실행 결과를 트랜잭션을 요청한 서비스 사용자(100)에게 반환할 수 있다(S762).

[0125] 도 8은 블록 생성기 노드(330)를 중심으로 수행되는 사용자 트랜잭션 처리 방법의 과정의 일부를 도시하는 개념도이다.

[0126] 트랜잭션 처리 결과가 서비스 사용자(100)에게 반환된(S762) 이후, 블록 생성기(330)의 리더 모듈(331)은 공유 로그 서비스 실행부(340)에 추가된 트랜잭션 로그를 공유 로그 API(333)를 경유하여 조회 요청하고(S810, S820), 공유 로그 API(333)를 경유하여 응답된(S830, S840) 로그를 읽어들이며 블록 생성에 필요한 데이터를 추출하여 블록에 담기는 트랜잭션 엔트리로 변환할 수 있다. 변환된 트랜잭션 엔트리는 블록 생성 모듈(332)로 전송될 수 있다(S850). 블록 생성 모듈(332)은 트랜잭션들을 블록에 담아 블록을 생성할 수 있다(S860).

[0127] 스마트 컨트랙트 실행부 노드(310)가 공유 로그 서비스 실행부(340)와 동기화하여 트랜잭션을 실행하므로 복수

개의 스마트 컨트랙트 실행부 노드들(310)은 공유 로그 서비스 실행부(340)에 저장된 개체에 대하여 동일한 상태 정보를 공유할 수 있다. 따라서, 스마트 컨트랙트 실행부 노드(310)는 서로 다른 트랜잭션을 병렬로 실행할 수 있다. 이로 인하여 스마트 컨트랙트 실행부 노드(310)의 확장성이 보장될 수 있으며, 트랜잭션이 로그로 기록되면 완료되었음을 의미하므로 기존 블록체인 시스템과 비교하여 더 빠른 커밋 결정이 허용될 수 있다.

[0128] 본 발명의 실시예와 대비되는 종래의 허가형/비허가형 블록체인 네트워크 시스템에서는 일반적으로 트랜잭션의 병렬 분산 실행이 어려운 것으로 알려져 있다.

[0129] 이에 비하여, 본 발명의 실시예에서는 공유 로그(342)가 완전한 블록체인 대신 경량 합의 과정을 지원하며, 또한 트랜잭션의 처리 내역, 버전 기록 등을 제공할 수 있으므로 기존의 블록체인 시스템보다 신속한 커밋 결정이 가능하다. 또한 시퀀스 노드(320)가 공유 로그(342)의 저장 순서를 결정함으로써 병렬적인 트랜잭션 처리 시 커밋 결과가 순차적으로 공유 로그(342)로서 저장될 수 있도록 보장할 수 있다.

[0130] 도 9는 본 발명의 일 실시예에 따른 일반화된 블록체인 네트워크 내의 노드 또는 컴퓨팅 시스템을 도시하는 블록도이다. 본 발명의 일 실시예에 따른 도 1 내지 도 8의 과정의 적어도 일부를 수행할 수 있는 노드가 컴퓨팅 시스템으로서 구현될 수 있다.

[0131] 도 1 내지 도 8의 실시예에서도 도면 상으로는 생략되었으나 프로세서, 및 메모리가 전자적으로 각 구성 요소와 연결되고, 프로세서에 의하여 각 구성 요소의 동작이 제어되거나 관리될 수 있다.

[0132] 본 발명의 일 실시예에 따른 블록체인의 사용자 트랜잭션 처리 방법의 적어도 일부의 과정은 도 9의 컴퓨팅 시스템(1000)에 의하여 실행될 수 있다.

[0133] 도 9를 참조하면, 본 발명의 일 실시예에 따른 컴퓨팅 시스템(1000)은, 프로세서(1100), 메모리(1200), 통신 인터페이스(1300), 저장 장치(1400), 입력 인터페이스(1500), 출력 인터페이스(1600) 및 버스(bus)(1700)를 포함하여 구성될 수 있다.

[0134] 본 발명의 일 실시예에 따른 컴퓨팅 시스템(1000)은, 적어도 하나의 프로세서(processor)(1100) 및 상기 적어도 하나의 프로세서(1100)가 적어도 하나의 단계를 수행하도록 지시하는 명령어들(instructions)을 저장하는 메모리(memory)(1200)를 포함할 수 있다. 본 발명의 일 실시예에 따른 방법의 적어도 일부의 단계는 상기 적어도 하나의 프로세서(1100)가 상기 메모리(1200)로부터 명령어들을 로드하여 실행함으로써 수행될 수 있다.

[0135] 프로세서(1100)는 중앙 처리 장치(central processing unit, CPU), 그래픽 처리 장치(graphics processing unit, GPU), 또는 본 발명의 실시예들에 따른 방법들이 수행되는 전용의 프로세서를 의미할 수 있다.

[0136] 메모리(1200) 및 저장 장치(1400) 각각은 휘발성 저장 매체 및 비휘발성 저장 매체 중에서 적어도 하나로 구성될 수 있다. 예를 들어, 메모리(1200)는 읽기 전용 메모리(read only memory, ROM) 및 랜덤 액세스 메모리(random access memory, RAM) 중에서 적어도 하나로 구성될 수 있다.

[0137] 또한, 컴퓨팅 시스템(1000)은, 무선 네트워크를 통해 통신을 수행하는 통신 인터페이스(1300)를 포함할 수 있다.

[0138] 또한, 컴퓨팅 시스템(1000)은, 저장 장치(1400), 입력 인터페이스(1500), 출력 인터페이스(1600) 등을 더 포함할 수 있다.

[0139] 또한, 컴퓨팅 시스템(1000)에 포함된 각각의 구성 요소들은 버스(bus)(1700)에 의해 연결되어 서로 통신을 수행할 수 있다.

[0140] 본 발명의 컴퓨팅 시스템(1000)의 예를 들면, 통신 가능한 데스크탑 컴퓨터(desktop computer), 랩탑 컴퓨터(laptop computer), 노트북(notebook), 스마트폰(smart phone), 태블릿 PC(tablet PC), 모바일폰(mobile phone), 스마트 워치(smart watch), 스마트 글래스(smart glass), e-book 리더기, PMP(portable multimedia player), 휴대용 게임기, 네비게이션(navigation) 장치, 디지털 카메라(digital camera), DMB(digital multimedia broadcasting) 재생기, 디지털 음성 녹음기(digital audio recorder), 디지털 음성 재생기(digital audio player), 디지털 동영상 녹화기(digital video recorder), 디지털 동영상 재생기(digital video player), PDA(Personal Digital Assistant) 등일 수 있다.

[0141] 본 발명의 실시예에 따른 방법의 동작은 컴퓨터로 읽을 수 있는 기록매체에 컴퓨터가 읽을 수 있는 프로그램 또는 코드로서 구현하는 것이 가능하다. 컴퓨터가 읽을 수 있는 기록매체는 컴퓨터 시스템에 의해 읽힐 수 있는 정보가 저장되는 모든 종류의 기록장치를 포함한다. 또한 컴퓨터가 읽을 수 있는 기록매체는 네트워크로 연결된

컴퓨터 시스템에 분산되어 분산 방식으로 컴퓨터로 읽을 수 있는 프로그램 또는 코드가 저장되고 실행될 수 있다.

[0142] 또한, 컴퓨터가 읽을 수 있는 기록매체는 롬(rom), 램(ram), 플래시 메모리(flash memory) 등과 같이 프로그램 명령을 저장하고 수행하도록 특별히 구성된 하드웨어 장치를 포함할 수 있다. 프로그램 명령은 컴파일러(compiler)에 의해 만들어지는 것과 같은 기계어 코드뿐만 아니라 인터프리터(interpreter) 등을 사용해서 컴퓨터에 의해 실행될 수 있는 고급 언어 코드를 포함할 수 있다.

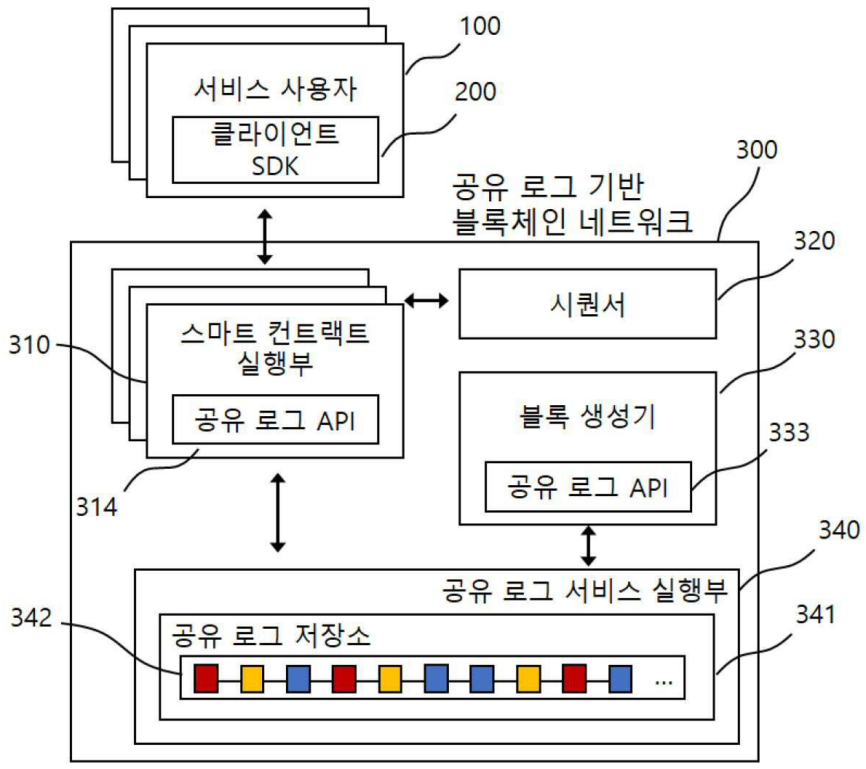
[0143] 본 발명의 일부 측면들은 장치의 문맥에서 설명되었으나, 그것은 상응하는 방법에 따른 설명 또한 나타낼 수 있고, 여기서 블록 또는 장치는 방법 단계 또는 방법 단계의 특징에 상응한다. 유사하게, 방법의 문맥에서 설명된 측면들은 또한 상응하는 블록 또는 아이템 또는 상응하는 장치의 특징으로 나타낼 수 있다. 방법 단계들의 몇몇 또는 전부는 예를 들어, 마이크로프로세서, 프로그램 가능한 컴퓨터 또는 전자 회로와 같은 하드웨어 장치에 의해(또는 이용하여) 수행될 수 있다. 몇몇의 실시 예에서, 가장 중요한 방법 단계들의 적어도 하나 이상은 이와 같은 장치에 의해 수행될 수 있다.

[0144] 실시예들에서, 프로그램 가능한 로직 장치(예를 들어, 필드 프로그래머블 게이트 어레이)가 여기서 설명된 방법들의 기능의 일부 또는 전부를 수행하기 위해 사용될 수 있다. 실시예들에서, 필드 프로그래머블 게이트 어레이(field-programmable gate array)는 여기서 설명된 방법들 중 하나를 수행하기 위한 마이크로프로세서(microprocessor)와 함께 작동할 수 있다. 일반적으로, 방법들은 어떤 하드웨어 장치에 의해 수행되는 것이 바람직하다.

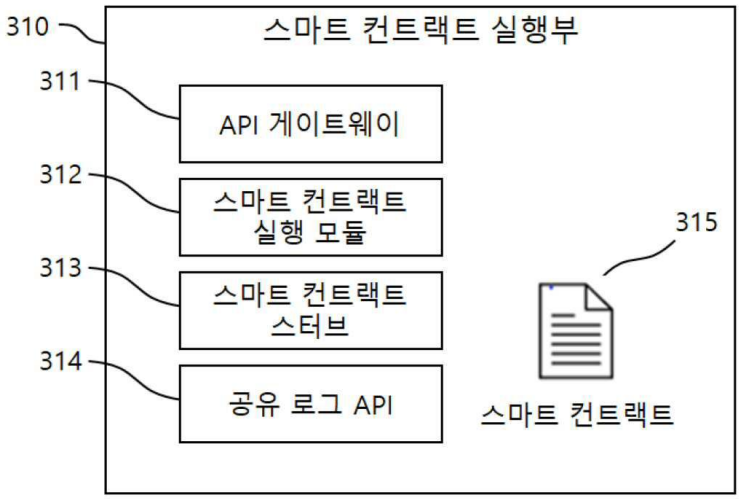
[0145] 이상 본 발명의 바람직한 실시 예를 참조하여 설명하였지만, 해당 기술 분야의 숙련된 당업자는 하기의 특허 청구의 범위에 기재된 본 발명의 사상 및 영역으로부터 벗어나지 않는 범위 내에서 본 발명을 다양하게 수정 및 변경시킬 수 있음을 이해할 수 있을 것이다.

도면

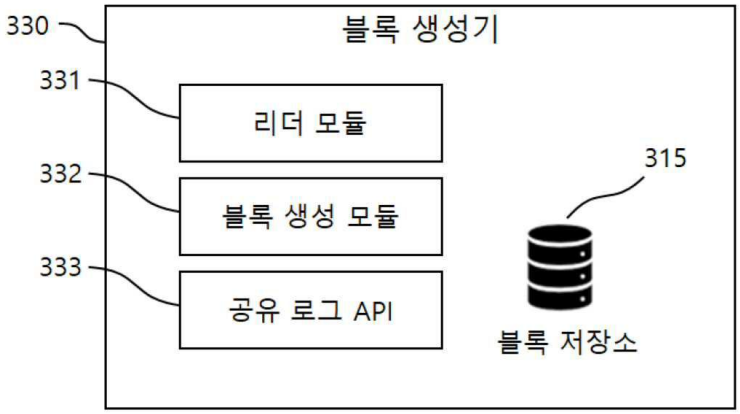
도면1



도면2


310
스마트 컨트랙트 실행부
311
API 게이트웨이
312
스마트 컨트랙트 실행 모듈
313
스마트 컨트랙트 스터브
314
공유 로그 API
315
스마트 컨트랙트

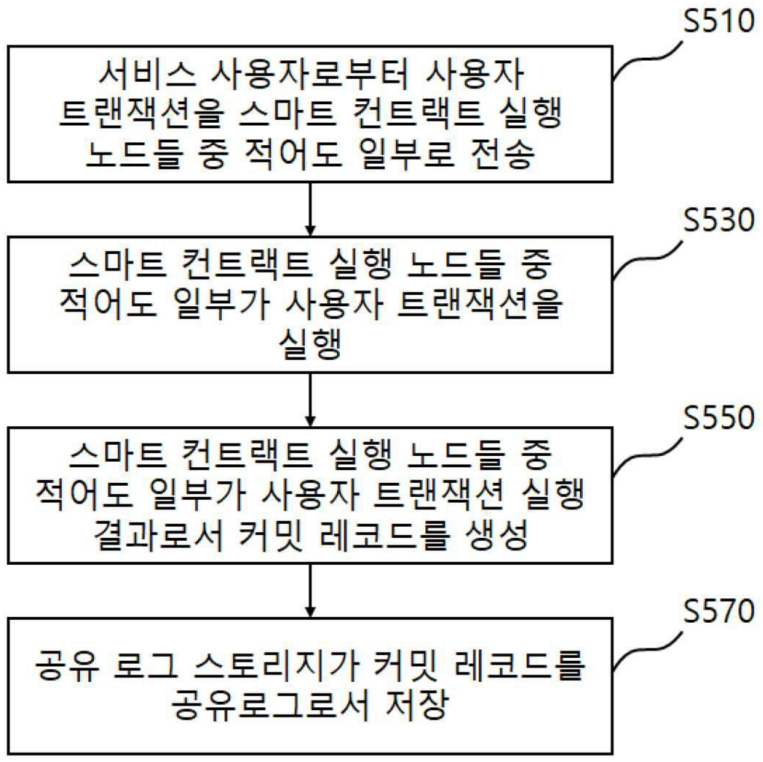
도면3


330
블록 생성기
331
리더 모듈
332
블록 생성 모듈
333
공유 로그 API
315
블록 저장소

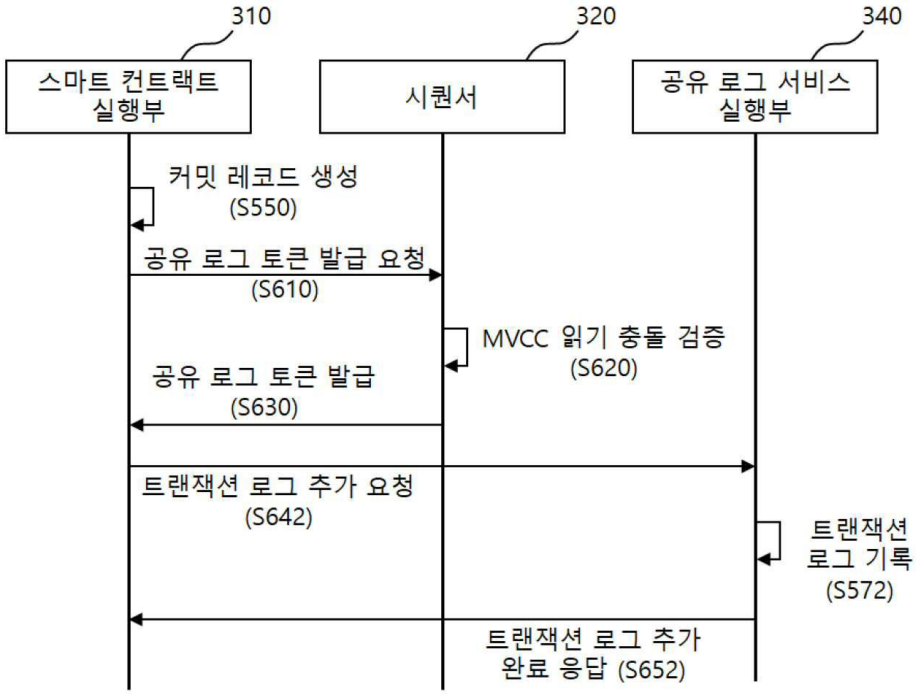
도면4

속 성	내 용
공유 로그 주소	공유 로그 프로토콜에 기록된 주소 정보
트랜잭션 타입	트랜잭션의 타입
클라이언트 ID	서비스 사용자 ID 정보
채널 ID	서비스 사용자가 트랜잭션을 전송하고자 하는 채널 정보
트랜잭션 ID	트랜잭션 ID 정보
스마트 컨트랙트 이름	서비스 사용자가 호출한 스마트 컨트랙트 이름
스마트 컨트랙트 버전	서비스 사용자가 호출한 스마트 컨트랙트 버전
전달인자	스마트 컨트랙트 함수명 및 입력값 & arguments
넌스(Nonce)	서비스 사용자에 의해 생성된 랜덤 변수
타임스탬프	트랜잭션이 생성된 시점의 시간 정보
읽기 집합	트랜잭션 실행 중 참조한 개체 집합
쓰기 집합	트랜잭션 실행 중 업데이트한 개체 집합

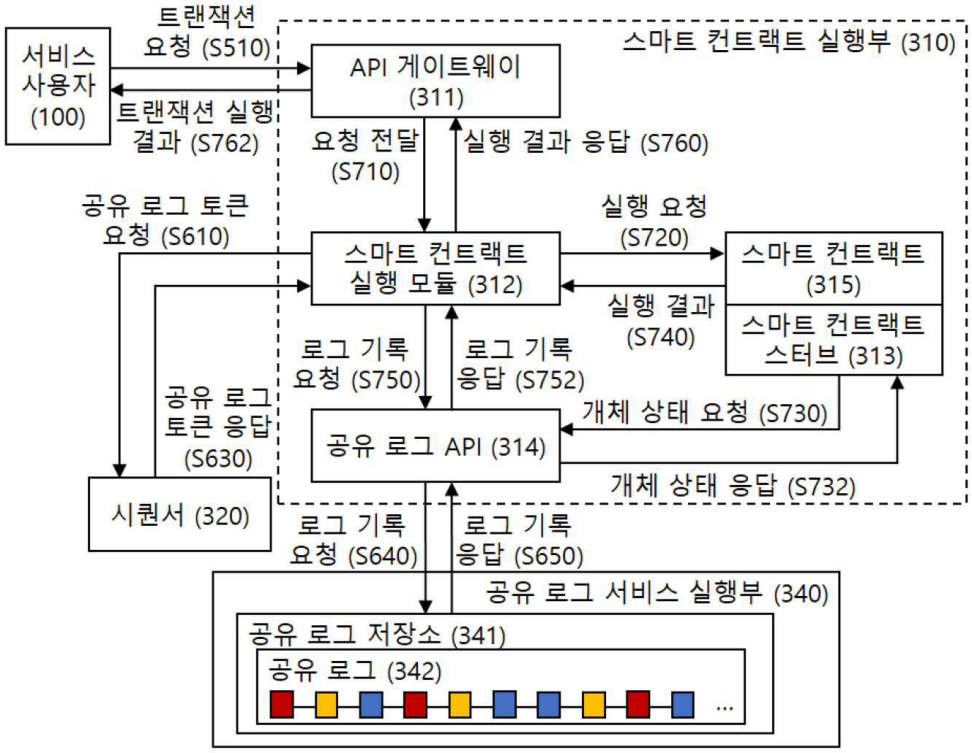
도면5



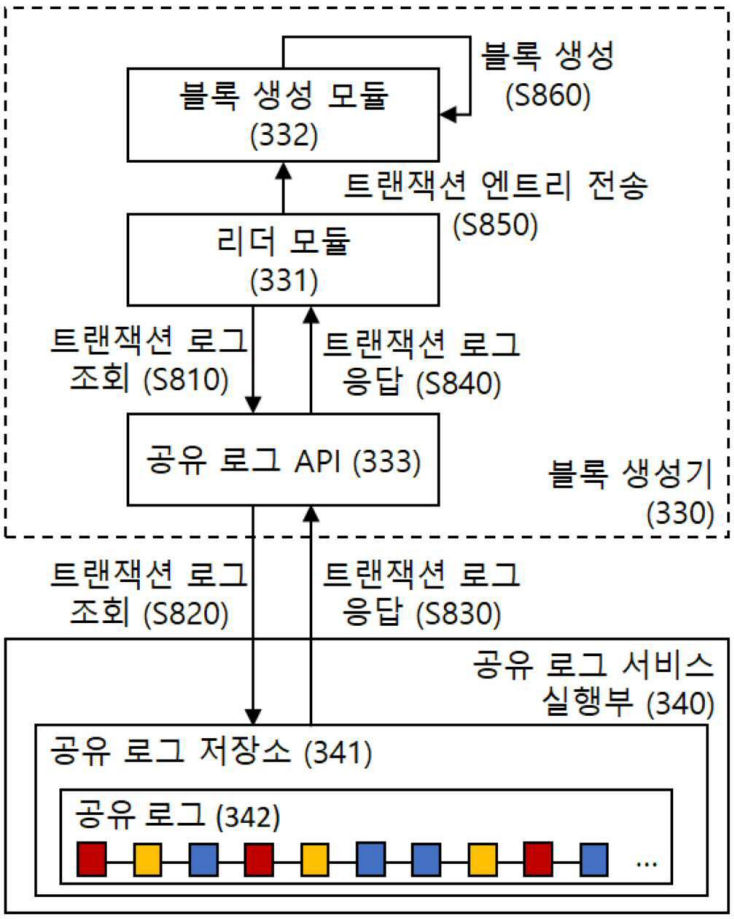
도면6


310
320
340
스마트 컨트랙트 실행부
시퀀서
공유 로그 서비스 실행부
커밋 레코드 생성 (S550)
공유 로그 토큰 발급 요청 (S610)
MVCC 읽기 충돌 검증 (S620)
공유 로그 토큰 발급 (S630)
트랜잭션 로그 추가 요청 (S642)
트랜잭션 로그 기록 (S572)
트랜잭션 로그 추가 완료 응답 (S652)

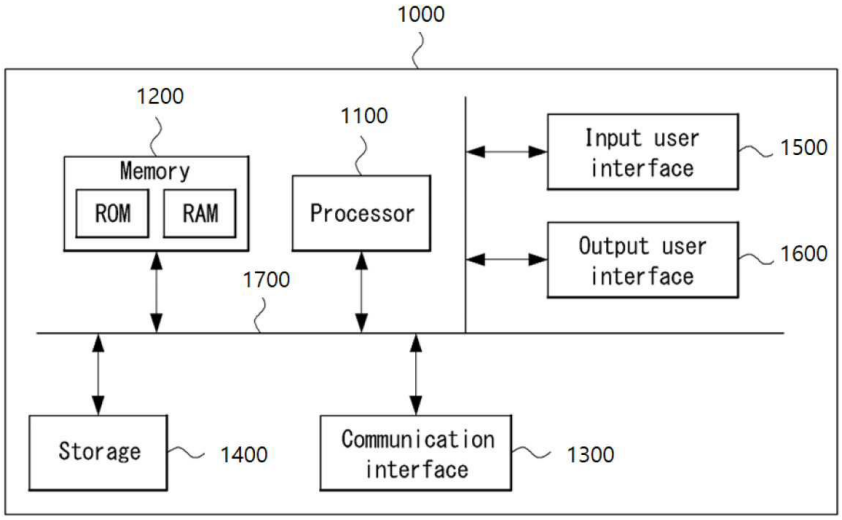
도면7


서비스 사용자 (100)
트랜잭션 요청 (S510)
트랜잭션 실행 결과 (S762)
API 게이트웨이 (311)
스마트 컨트랙트 실행부 (310)
요청 전달 (S710)
실행 결과 응답 (S760)
공유 로그 토큰 요청 (S610)
실행 요청 (S720)
스마트 컨트랙트 실행 모듈 (312)
스마트 컨트랙트 (315)
실행 결과 (S740)
스마트 컨트랙트 스터브 (313)
로그 기록 요청 (S750)
로그 기록 응답 (S752)
공유 로그 토큰 응답 (S630)
개체 상태 요청 (S730)
공유 로그 API (314)
개체 상태 응답 (S732)
시퀀서 (320)
로그 기록 요청 (S640)
로그 기록 응답 (S650)
공유 로그 서비스 실행부 (340)
공유 로그 저장소 (341)
공유 로그 (342)
...

도면8


블록 생성
(S860)
블록 생성 모듈
(332)
트랜잭션 엔트리 전송
(S850)
리더 모듈
(331)
트랜잭션 로그
조회 (S810)
트랜잭션 로그
응답 (S840)
공유 로그 API (333)
블록 생성기
(330)
트랜잭션 로그
조회 (S820)
트랜잭션 로그
응답 (S830)
공유 로그 서비스
실행부 (340)
공유 로그 저장소 (341)
공유 로그 (342)
...

도면9


1000
1200
Memory
ROM
RAM
1100
Processor
Input user
interface
1500
Output user
interface
1600
1700
Storage
1400
Communication
interface
1300