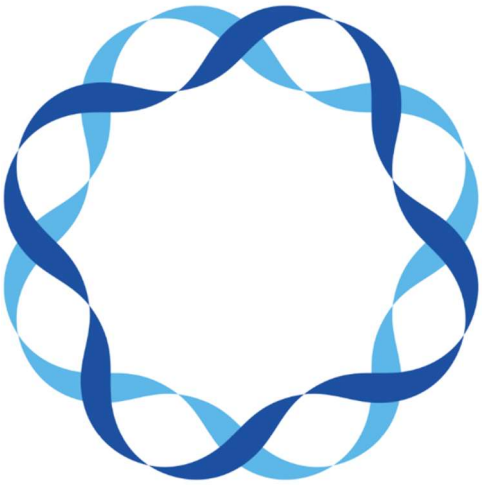


Locus Chain Tech Whitepaper

Posted: May 6, 2024

Revised: May 20, 2024

Revised: May 31, 2024 (일부 오타 수정)

Contents

1. 중요공지

이 섹션을 자세히 읽어주시기 바랍니다. 귀하가 로커스체인 관련 의사결정을 하여야 하는 경우 귀하의 법적, 재정, 세금, 또는 기타 적합한 전문 어드바이저(들)와 상의하십시오.

이 백서의 정보는 변경 또는 갱신될 수 있습니다. 토큰 및 코인의 사용과 관련된 서비스의 미래 사용성과 관련하여 로커스체인 또는 이 백서에 언급된 내용을 개인으로서 또는 단체로서 귀하께 약속 또는 보증해드리는 것으로 해석해서는 안됩니다.

이 백서는 주식이나 증권을 팔기 위한 제안이나 권유를 위한 목적이 아닙니다. 따라서 이 백서가 어떠한 계약이나 약속과 관련하여 어떠한 증빙자료로 사용되거나 의존하시면 안됩니다. 어떠한 유가증권의 매도, 청약 또는 매수 또는 청약의 권유로도 해석되어서는 안됩니다. 로커스체인은 이 백서에 포함된 정보에 대한 의존, 그러한 정보의 오류, 누락 또는 부정확함 또는 이로 인한 조치에 의해 직간접적으로 발생하는 모든 종류의 직접적 또는 결과적 손실/손해에 대한 모든 책임을 명시적으로 부인합니다.

이 백서는 구매를 권유하거나 재정적 조언을 위한 것이 아니며 정보제공을 목적으로 합니다. 이 정보에 기반하여 토큰, 코인, 회사 주식 등의 자산을 거래하거나 투자하지 마십시오. 모든 투자에는 가격 변동성 원금 손실 가능성의 위험이 수반됩니다. 투자자께서는 투자 결정을 내리시기 전에 이 백서에서 논의된 주제 또는 이슈들에 대해 전문 금융, 법률 및 세무 전문가의 도움을 받아 자체 조사나 기업실사를 하셔야 합니다.

이 백서에 포함된 모든 정보는 정확하고 신뢰할 수 있는 출처에서 작성되도록 최선을 다하였습니다. 그럼에도 이 백서에서 인용되는 시장 가격, 데이터 및 기타 정보는 완전성 또는 정확성이 완전히 보증되지 않습니다. 이 백서에서 인용된 공개 시장 정보 및 경쟁 시장 환경에 기반한 저희 견해나 평가는 시장 사정에 따라 사전 통지 없이 변경될 수 있습니다. 다만 변동되는 사항에 대해서는 이 백서의 업데이트 버전에 반영될 수 있습니다.

이 백서는 시장에 대한 전망이 포함되어 있습니다. 그러한 전망의 정확성에 대해 어떠한 진술이나 보증도 하지 않습니다. 이 문서에 포함된 모든 예측이나 전망은 합리적 추정에 근거하며 특정한 가정에 기반합니다. 이러한 미래 예측은 부정확한 가정 또는 알려지거나 알려지지 않은 위험 등 불확실성 및 기타 요인의 영향을 받을 수 있으므로 로커스체인이 통제할 수 없습니다. 따라서 미래 예측은 추후 실제 결과와 다를 수 있으며 이는 로커스체인이 통제할 수 있는 범위 밖이라는 점을 분명하게 밝혀 드립니다. 그러한 미래 지향적인 가정의 일부 또는 전부가 실현되지 않거나 실제 결과와 크게 다르다는 점을 투자 등 주요 의사결정에 반영하셔야 합니다.

2. 들어가기

A. 로커스체인 사업개요 Locuschain Business Overview

Locus Chain 은 다자간 위변조 불가능한 통신을 고효율(대규모, 고속, 저비용)로 가능하게 하는 퍼블릭 분산원장 프로토콜이다. 로커스체인은 거래와 참여자가 대규모로 늘어도 속도가 느려지지 않고 실시간 처리가 가능한 퍼블릭 블록체인 프로토콜이다. 이를 구현하기 위해 DAG-AWTC + BFT 와 Dynamic Sharding, Verifiable Pruning 기술을 채택하여 원장의 구조와 합의 방식은 물론 원장의 저장 공간과 네트워크 부하를 낮은 수준에서 유지하는 구조까지 구현하였다.

로커스체인의 고도의 확장성과 초경량 노드는 인터넷상 모든 데이터에 대한 신뢰, 소유, 인증 등에 대해서 제 3 자의 개입없이 알고리즘에 의한 완전 탈중앙화 처리를 무한히 확장할 수 있다. 로커스체인은 탈중앙화 기술로 왜곡없이 안전하게 인증된 데이터를 병목 없이 고속으로 처리할 수 있어 기존 인터넷 서비스에 비해 업그레이드된 데이터 가치 생태계를 형성하는데 기여할 수 있다.

(1) 사업적 비전

로커스체인이라는 거대 규모의 스케일러블한 퍼블릭 블록체인 네트워크를 통해 데이터들에 새로운 가치가 매겨지고 다양한 목적에 의해 제공자와 이용자간에 직접 전송과 거래가 이루어지는 완전히 새로운 차원의 탈중앙화(Web3) 데이터 생태계가 생성되어 새로운 비즈니스 영역이 생겨나고 새로운 신용 사회를 구현하는 것이다. 로커스체인에 기반하여 기존 연계성이 낮은 비즈니스들을 융복합하여 신수종 비즈니스 영역을 창출하고 그들이 또다시 융복합 과정을 거치면서 데이터 생태계의 진화를 선도하고자 한다.

(2) 개발철학

우리는 개발초기부터 로커스체인을 통해 광범위한 확장성을 가진 고도의 탈중앙화 네트워크로 중개인 없이도 불특정 다자간 신뢰를 광범위하게 일반화하여 사회적 신용의 기저를 근본적으로 제고시키는 것을 목표로 한다.

(3) 로커스체인의 성능

보통의 가정용 PC 와 네트워크 환경에서 추가비용 없이도 원장 작성에 참여가능하도록 하여 진정한 퍼블릭 환경을 구현하고자 하였다. 구체적으로 보면, CPU 점유율을 5%이내로, Network Bandwidth 점유율을 100Mbps 기준 5% 이내로, On-Device Ledger Size 는 10GB 이내로, TPS 는

최소 4k 로 설계되었다. Scalability 는 블록체인 자체 처리속도로는 제한이 없으므로 네트워크 속도까지가 처리 용량의 한계이다. 건당 처리 시간은 0 sec ~ Few sec (가장 느릴 때)를 목표로 한다. 로커스체인은 다음의 자체적인 기술들을 개발하고 이를 결합하여 구현할 수 있게 되었다.

(4) 활용 분야

로커스체인은 탈중앙화 기술로 서버 없이도 왜곡없이 안전하게 인증된 데이터를 병목 없이 고속으로 처리가능하다. 대부분의 기존 인터넷 서비스에도 탈중앙화 환경의 적용이 가능하다. 이를 통해 탈중앙화 기반 새로운 데이터 가치 생태계를 만들고 이에 기반한 새로운 어플리케이션과 서비스가 가능하게 할 수 있다.

데이터의 가치가 비용대비 낮아 거래 가치가 없는 기존 데이터도 거래 비용을 획기적으로 낮춤으로서 데이터 교환 경제의 혁신하는 탈중앙화 환경을 제공한다. 노드 자체가 모든 종류의 어플리케이션과 동시에 동작이 가능한 가벼운 규모의 Add-On 프로토콜이므로 탈중앙화 어플리케이션과 서비스의 개발이 매우 용이하다. 특히, 각자의 디바이스에 산재하는 데이터를 로커스체인의 검증 기능만을 이용해 진위를 인증할 수 있으므로 데이터 보안과 주권이 중요한 On-Device 서비스에 적합한 성능과 구조를 가지고 있다.

B. 로커스체인 기술개요 Locuschain Technology Overview

로커스체인은 고속 고성능 트랜잭션 기록과 합의 기능을 주로 하는 레이어 1 블록체인이다.

블록체인 시스템의 대표격인 비트코인과 이더리움의 정보 처리 성능은 초당 약 20 트랜잭션(20tps)정도이다. 즉 1 초에 20 명 정도가 블록체인에 정보를 기록하는 것이 가능하다. 이러한 성능은 실세계의 일반적인 기준으로 보아 그다지 만족스럽지 못한 처리이다. 예를 들어 글로벌 신용카드 브랜드인 VISA 카드의 거래량은 초당 4 천 트랜잭션 정도이다.

그러나 블록체인 시스템의 처리량을 높이려 하면 여러 기술적 문제에 부딪히게 된다. 대표적인 문제는 처리 정보량에 따른 원장사이즈와 통신량의 증가이다. 로커스체인이 VISA 카드와 같은 초당 4,000 트랜잭션을 처리할 수 있다고 하였을 때 하루 발생하는 정보량은 약 350 기가바이트 정도가 된다. 이 정보량은 일반적인 PC 의 1TB 저장공간을 3 일만에 소비하는 용량으로, 개인이 일상적으로 사용하는 디바이스로 처리하기에는 지나치게 큰 부담이다.

로커스체인은 이러한 처리 속도와 용량 문제를 해결하는 방안으로 고도의 탈중앙화 환경에서 광범위한 확장성을 가진 퍼블릭 분산원장 프로토콜을 디자인하였다. 이를 구현하기 위해 DAG

AWTC + BFT, Verifiable Pruning, Dynamic Sharding 을 개발하였으며 이들 중에서 핵심기술들에 대해서는 특허등록을 진행중이다. 2024 년 4 월 25 일 현재 10 건의 국내외 특허를 출원하였으며 이중 4 건의 특허는 등록을 완료하였고 6 건의 특허는 심사 중이다. 추가적인 복수의 기술개발들에 대해서도 특허출원을 준비중이다. 다만 이 백서 작성일 기준 특허출원을 준비중인 기술들에 관한 설명은 특허 출원 요건을 충족시켜야 하는 사정으로 이번 백서업데이트에서 제외되었다. 2024 년 5 월 6 일 기준 국내외 로커스체인 특허 출원 및 등록의 상세 현황은 각각 붙임자료 1 과 붙임자료 2 와 같다.

(1) 처리 성능의 향상을 위한 원장 시스템과 합의 방식의 혁신

로커스체인은 합의 방식에 따른 속도 제한을 해결하고자 DAG-AWTC 를 채택하였다. 합의 방식에 따른 제한이 사라지면 네트웍 대역폭의 제한에 접하는데, 이를 극복하기 위해서는 샤딩이 필요하다. DAG 를 이용하는 기존의 멀티 엔드포인트 원장 시스템은 BFT 확정합의를 적용하기 어렵고, 이는 샤딩을 통한 성능 확장에 걸림돌이 되어 왔다. 샤딩과 프루닝을 가능하게 한 기반 기술로서 로커스체인은 방향성 비순환 그래프(Directed Acyclic Graph) 정보 구조에 의한 AWTC 원장 시스템과 가중 지분 증명(Proof of Stake)방식으로 구성되는 합의체에 의한 BFT 합의를 사용한다. 로커스체인은 개발 초기부터 Dynamic Sharding 을 염두에 두고 DAG-AWTC 를 개발해 확정합의 기반 알고리즘 기술을 개발하여 장착했다. DAG + BFT 기술은 로커스체인이 세계 최초로 개발하였고 2019 년 8 월 1 일자로 특허 출원하여 등록 완료하였다.

(2) 샤딩을 통한 처리 성능의 향상과 공정성의 유지

컴퓨터 기술 분야에서 이러한 처리 용량 문제를 해결하기 위한 대표적인 접근방법 중 하나로 **샤딩**이 있다. 샤딩은 간단히 말하면 해결해야 할 큰 대상을 작게 여럿으로 분할하여 처리하는 접근 수법이다. 데이터베이스 및 네트워크 등의 완성된 분야에서 잘 연구되고 적용된 방법이지만, 블록체인 분야의 적용은 아직 드물다.

로커스체인은 원장과 네트웍을 처리량에 따라 자동으로 공평하게 나눠 분할하는 **동적 샤딩 (dynamic sharding)** 기술을 통해 처리량의 증가에 유연하게 대응한다.

로커스체인의 모든 샤드는 대체적으로 비슷한 양의 트랜잭션을 처리하도록 설계되어 있다. 트랜잭션이 한 샤드에 집중되면 샤드 재구성을 통해 다른 샤드로 처리량을 공정하게 다시 나누게 되어 있고, 시스템 전체의 트랜잭션이 늘어나면 샤드 숫자를 늘려서 한 샤드의 처리량을 작게

유지하게 되어 있다. 이에 따라 로커스체인의 참여자가 어떤 샤드에서 어떤 처리를 하여도 공평한 처리량 부담과 이에 따른 마이닝 보수를 얻을 수 있도록 되어 있다.

로커스체인은 Dynamic Sharding 을 통해 네트워크를 필요한 만큼의 여러 샤드로 분할하여 네트워크 부하를 줄이고 동시에 처리할 수 있는 Throughput 을 기존 대비 수십~수천배 확장 가능하다. 다이나믹 샤딩(Dynamic Sharding)을 통해 사용량 증가에 대응하여 블록체인 네트워크를 여러 개의 샤드로 분할 처리하여 네트워크의 확장성과 보안 이슈를 해결함으로써 Locus Chain 의 무한 확장성을 장착하였다. Dynamic Sharding 기술도 로커스체인이 세계 최초로 개발 완료하였으며 2022 년 6 월 3 일~11 월 3 일까지 총 8 건의 출원된 특허 중 2024 년 4 월 25 일 현재 2 건이 등록 완료되었고 6 건이 심사 중이다.

(3) 프루닝을 통한 정보 저장량 경감

블록체인이 처리하는 정보는 서로 연관지어져 정리된 원장(ledger)으로서 블록체인 시스템 상에 존재하게 된다. 기존의 블록체인에 참여하는 컴퓨터는 원장 전체를 전부 로컬 컴퓨터 상에 기록하고 저장할 필요가 있었다. 그러나 로커스체인이 바라보는 고속 대용량 처리 블록체인에서는 일반적인 PC 상에 도저히 저장할 수 없는 대용량의 정보가 발생하게 된다.

로커스체인은 이 처리량 문제를 해결하는 또 하나의 접근 방법으로서, 이용빈도가 적은 과거의 원장 정보를 로컬 컴퓨터에서 삭제하는 **원장 프루닝(pruning)**을 채택하고 있다.

로커스체인은 AWTC 라 이름지어진 방향성 비순환 그래프(Directed Acyclic Graph; DAG)원장구조를 채택하고 있다. 로커스체인의 독자적인 **검증 가능 프루닝 (Verifiable Pruning)**은 이 그래프 원장 구조를 통해 최신의 원장 상태가 과거의 트랜잭션과 어느정도 관련성이 있는지를 수학적으로 파악하여, 과거의 무관한 트랜잭션 정보를 대부분 삭제한 상태에서도 현재의 정보와 새로 수신한 정보의 정당성(validity) 검증을 가능하게 한 기술이다.

이를 통해 원장 작성에 참여하는 노드 디바이스들의 하드웨어 비용을 획기적으로 절감해주는 방법이다. 원장과 네트워크 부하를 줄이기 위한 Sharding 에 더해 원장 사이즈를 획기적으로 줄이는 기술이다. 대부분의 블록체인은 고성능 컴퓨터에서만 노드 운영할 수 있으나 로커스체인은 모바일 디바이스같은 저사양 컴퓨터로도 노드 운영 가능하다. 로커스체인은 Verifiable Pruning 기술을 세계 최초 개발 하여 2019 년 8 월 1 일자로 특허출원하여 등록 완료하였다.

(4) 컨트랙트 확장 플러그인 (VME)

컨트랙트 확장 플러그인은 로커스체인의 Ecosystem 을 외부 Smart Contract 은 물론 서버기반 프로젝트들까지 확장시킬 수 있는 Interface Architecture Solution 이다. 로커스체인이 1.0 개발 과정에서 기존 로드맵을 일부 수정하여 확장팩 구축에 나서게 된 것은 개발 과정에서 로커스체인의 범용성을 획기적으로 확장할 게이트웨이를 발견했기 때문이다. 당초 기획규모를 뛰어넘는 혁신이자 강점으로 로커스체인의 진화를 견인할 구조전환이라 할 수 있다.

로커스체인의 스마트 컨트랙트 엔진은 초당 수천트랜잭션의 처리량 중 서로 연관된 스마트컨트랙트 트랜잭션만을 분별하여 독립적으로 실행하기 위한 **"플러그인" VM 구조**를 갖추고 있다.

그리고 샤드 내 정보 교환과 샤드간 정보 교환을 위해 P2P 통신을 주로 하는 네트웍 레이어를 갖추고 있다. 네트웍 레이어는 로커스체인의 원장 및 합의 기구 외에도, 원장에 기록된 트랜잭션을 바탕으로 스마트 컨트랙트를 실행하기 위한 스마트 컨트랙트 엔진의 하부구조로서의 기능도 갖는다.

컨트랙트 확장 플러그인으로 이더리움 등 다른 블록체인 메인넷 기반 Smart Contract 들을 로커스체인 코어엔진에 장착하면 완벽하게 작동되는 로커스체인 서브블록체인 프로젝트로 동작하게 된다. 금융, 스마트 시티, 게임 등의 프로젝트들도 VME 를 통해 서브시스템으로 구동하게 하거나 코어엔진 위에서 직접 구동시키는 등 프로젝트의 특성에 따라 다양한 구조로 로커스체인 경제생태계에 확장 편입할 수 있다.

위와 같은 기술 구현을 통해 개발된 로커스체인은 현재 블록체인 시스템을 통신네트워크 및 상태저장 데이터베이스로 이용하는 실제 어플리케이션에 적용하는 단계에 와 있다.

(5) 로커스체인 기술 개발 기대효과

대부분의 블록체인 프로젝트들은 분산장부에 기반한 프로젝트에 한정된 사용성을 갖는다. 로커스체인은 VME 를 통해 서버기반 시스템도 코어엔진에 연동이 가능하다. 완전분산성을 유지하면서도 블록체인 생태계의 한계를 뛰어넘어 서버에 기반한 시스템도 포용해내는 범용성을 갖게 되었다.

로커스체인은 Micro Device 등을 Edge 에서 구동시켜 정보처리 및 교환을 unlimited 하고 효율적으로 수행하는 핵심 Protocol 이다. 현재 외부 개발자들도 Locuschain Protocol 에 기반한 프로젝트 개발이 가능하게 하기 위한 SDK 를 배포하기 위한 준비작업이 진행되고 있다.

이러한 일정이 마무리되면 로커스체인의 범용성은 모든 블록체인 프로젝트들까지 확장될 것으로 기대된다. 또한 서버기반 시스템들도 완전분산 퍼블릭 블록체인인 로커스체인에 확장팩으로 연동 가능하게 되어 분산원장기반은 물론 서버기반 등까지 적용범위를 넓힐 수 있을 것으로 기대된다.

3. 로커스체인 원장 및 트랜잭션 구조

(1) 원장

일반적인 블록체인과 마찬가지로 로커스체인의 원장(ledger)은 참여자가 자유롭게 정보를 추가할 수 있는 분산 정보 저장 시스템이다. 좁은 의미에서 원장은 블록체인의 참여자가 발행하여 추가한 트랜잭션의 총합이다. 넓은 의미에서는 나아가서 트랜잭션을 확정하기 위해 생성되는 합의 관련 정보와 블록체인의 운용 관련 정보를 포함한다.

(2) 어카운트/계정

어카운트는 로커스체인의 원장에 트랜잭션을 발행하여 정보를 능동적으로 추가할 수 있는 행위자이다. 어카운트는 발행한 트랜잭션을 서명하기 위해 암호학적 비밀키와 공개키 쌍을 갖는다. 어카운트는 공개키로부터 계산된 주소를 통해 유일하게 식별 가능하다.

어카운트 주소와 공개키는 누구나 참조할 수 있는 공개 정보이고, 비밀키는 어카운트 소유자만이 알고 있다. 어카운트는 비밀키를 사용하여 트랜잭션을 발행하는 등의 본인을 증명하는 행위를 하게 된다.

(3) 어카운트 상태 및 트랜잭션

어카운트 상태(account state)는 어카운트의 내부 정보 값의 상태이다. 대표적인 어카운트 상태를 나타내는 값으로 어카운트의 소지 코인량이 있다. 그 외에도 임의의 정보값을 트랜잭션을 통해 생성 및 갱신할 수 있다.

어카운트 상태는 어카운트 소유자 만이 변경할 수 있다. 어카운트 상태는 변경 내용을 담은 트랜잭션을 발행함으로써 변경된다. 예를 들어 어카운트 A가 10코인을 어카운트 B에게 보낼 경우, 어카운트 A의 소유자는 다음과 같은 내용을 담은 트랜잭션을 발행하여 어카운트 상태의 소지 코인량을 변경한다.

```
{
  issuer: A,   // the issuer
  tx_no: 4,    // transaction number
  partner: B,  // related Account
  pseudo_command: {
    MOVE coin(10) FROM A TO B;
  }
}
```

Figure 1: A 가 발행하는 트랜잭션의 예

어카운트 상태는 공개되는 내용이지만 어카운트 상태의 모든 값 자체가 원장에 기록되는 것은 아니다. 하지만 트랜잭션에는 언제 어떤 값을 변경했는지의 내용과 변경 전후의 상태를 나타내는 대표값을 포함될 수 있으므로, 참여자는 자기가 알고 있는 어카운트 상태가 올바르다는 것을 트랜잭션 기록을 확인하여 검증할 수 있다.

(4) 메시지 트랜잭션

로커스체인의 트랜잭션은 어카운트간 메시지와 같은 역할을 갖는다.

위 예제를 보았을 때 A가 트랜잭션을 발행한 시점에서 A의 코인량은 줄어든 것으로 간주한다. 그러나 어카운트 B의 상태는 어카운트 B의 소유자만이 변경할 수 있으므로, A가 트랜잭션을 발행한 직후에는 아직 어카운트 B의 코인량은 늘어나지 않는다. 어카운트 B의 코인량을 늘리기 위해서 B의 소유자는 자기 코인량을 늘리는 트랜잭션을 명시적으로 발행해야만 하고, 이 때 코인량을 늘리는 명목으로서 A가 발행한 기존의 트랜잭션을 참조한다.

```
{
  issuer: B,     // the issuer
  tx_no: 7,      // transaction number
  input: [A:4],  // reference to other TX
  pseudo_command: {
    ADD coin(10) USING [A:4]
  }
}
```

Figure 2: B 가 발행하는 트랜잭션의 예

B가 위와 같이 트랜잭션을 발행하였을 때 비로소 B의 어카운트 상태 내의 코인량이 갱신되고, A가 발행한 트랜잭션은 역할을 마치게 된다. 그리고 B입장에서 보았을 때 어카운트 A가 발행한 트랜잭션은 A가 B에 코인을 보내는 메시지 역할을 하게 된다.

로커스체인의 트랜잭션을 어카운트 간의 메시지라고 보았을 때, 로커스체인의 원장은 비트코인의 UTXO원장과 이더리움의 어카운트 정보 원장의 중간 형태라 볼 수 있다. 로커스체인에서 둘 이상의 어카운트가 관계하는 트랜잭션은 상대방 어카운트에 도달할 때 까지는 미사용(unspent)상태로 존재하고, 상대방 어카운트가 트랜잭션을 수신하여 이를 확정하면 비로소 사용후(spent)상태가 된다고 볼 수 있다. 하지만 BitCoin처럼 UTXO를 받아 UTXO를 출력하는 경우와 달리, 입력이 트랜잭션이 아니라 공유된 어카운트 상태이므로 트랜잭션 자체를 더블스펜딩하는 경우는 일어나지 않는다. 그러나 적대적인 참여자가 같은 어카운트 상태 시점에 둘 이상의 상반된 트랜잭션을 발행하는 것은 상정할 수 있다. 이러한 경우 다른 노드가 수신한 트랜잭션에 따라 파악하는 어카운트 상태가 달라지는 경우가 존재할 수 있고, 의사적으로 더블스펜딩으로 볼 수 있는 경우가 발생한다.

(5) 트랜잭션 번호

로커스체인의 어카운트 상태는 트랜잭션이 발생을 통해 변경된다. 이 때 트랜잭션은 한번에 단 하나만 발행할 수 있다. 따라서 어떤 시점의 어카운트 상태와 그 상태에서 발행된 트랜잭션에는 일련번호를 부여할 수 있다. 이를 트랜잭션 번호라 한다. 어카운트가 새로 생성되는 순간 최초로 자동 발행되는 트랜잭션이 번호 0 번을 갖고, 이후 트랜잭션을 통해 어카운트 상태 변동이 일어날 때마다 1 씩 증가한다.

이 트랜잭션 번호를 이용하여 일종의 더블스펜딩 상태를 검출할 수 있다. 같은 어카운트에 동일한 트랜잭션 번호를 갖는 서로 다른 트랜잭션을 발견하면 그 트랜잭션은 일종의 더블스펜딩 상태로 간주된다.

(6) 로커스체인 어카운트별 트랜잭션 체인 (AWTC)

트랜잭션은 한번에 하나만 발행될 수 있고, 발행된 트랜잭션은 자동적으로 같은 어카운트의 직전 트랜잭션에 대한 참조 링크를 갖게 된다. 한 어카운트가 발행한 모든 트랜잭션은 한 줄로 연결될 수 있고, 이를 로커스체인에서는 어카운트 트랜잭션 체인 (Account-wise Transaction Chain; AWTC)라 칭한다. 다시 말해 모든 어카운트는 자기만의 트랜잭션 체인을 갖고, 언제든지 자유롭게 자기 체인에 트랜잭션을 추가할 수 있다.

또한 앞 코인 전송 예와 같이 발행되는 트랜잭션은 다른 어카운트가 발행한 트랜잭션을 참조할 수 있다. 트랜잭션이 참조하는 다른 트랜잭션들 항상 시간 관계상 과거의 트랜잭션이므로, 참조는 시간 방향과 같은 단방향 링크로 기능한다. 따라서 트랜잭션간의 관계는 수학적으로 방향성 비순환 그래프 (Directed Acyclic Graph, DAG)를 구성하게 된다.

한 트랜잭션은 복수의 트랜잭션을 참조할 수 있고, 또한 복수의 관련 어카운트를 지정할 수 있다. 따라서 트랜잭션 DAG의 각 노드는 복수의 입력 엣지와 복수의 출력 엣지를 가질 수 있다.

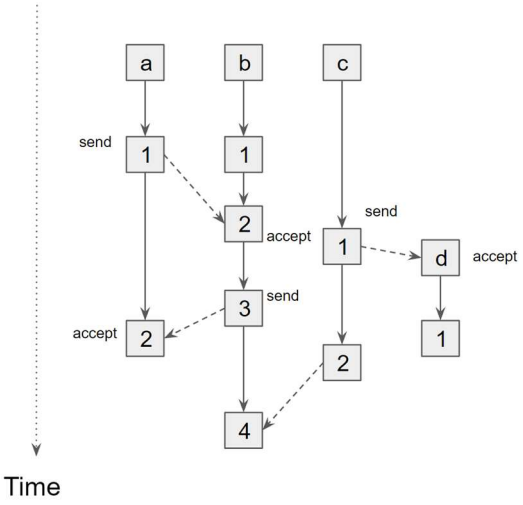


Figure 3: AWTC 체인의 DAG 구성

로커스체인의 원장은 트랜잭션의 집합이고, 따라서 서로 참조로 연결되어 DAG를 구성하는 어카운트 AWTC로 구성되는 DAG원장이 된다. DAG원장의 각 어카운트는 단 하나의 룰, 트랜잭션 번호가 연속된 유일한 번호여야 한다는 규정을 지키는 한 언제든지 자유롭게 트랜잭션을 발행할 수 있다.

(7) 원장 샤드 분할

필요에 따라 로커스체인의 원장은 샤드로 분할된다. 분할된 어떤 한 샤드가 담당하는 원장을 샤드 원장이라 한다.

트랜잭션은 어카운트 단위로 샤드에 소속된다. 다시 말해 어카운트는 어떤 시점에서 항상 어떤 한 특정 샤드에 유일하게 소속되고, 그 어카운트가 발행한 과거의 트랜잭션은 전부 어카운트가 소속된 샤드의 원장에 포함되는 것으로 간주한다.

로커스체인의 동적 샤딩을 통해 샤드 재구성이 일어날 수 있다. 트랜잭션이 어카운트 단위로 묶여지므로 샤드가 재구성될 때 어카운트를 통째로 다른 샤드로 이동시키게 된다. 다시 말해, 샤드 재구성시 어카운트의 샤드간 이동을 통해 실행되고, 이 때 이동되는 어카운트의 모든 트랜잭션은 기존의 샤드 원장에서 새로운 샤드 원장으로 소속이 변경된다.

샤드간 트랜잭션 전달

트랜잭션은 항상 어카운트 소속 샤드의 원장에 포함되지만, 샤드간 정보 교환을 위해 트랜잭션이 다른 샤드에 전달될 필요가 있다. 예를 들어 소속하는 샤드가 서로 다른 두 어카운트

간에 정보 교환이 이루어지는 경우, 각 어카운트가 발행한 트랜잭션은 상대방 샤드에 검증 가능한 형태로 전달되어야 한다.

(8) 노드 및 네트워크 샤드

로커스체인의 샤딩은 논리적으로 원장을 샤딩하고, 물리적으로 네트웍을 샤딩한다. 다시 말해 네트워크에 참여하는 노드를 샤드별로 그룹지어 해당 그룹의 노드 간에 주로 샤드 원장의 정보가 교환되고 공유된다. 로커스체인에서 네트워크 샤드의 갯수와 원장 샤드의 갯수는 동일하고, 한 네트워크 샤드가 한 원장 샤드를 담당한다.

(9) 노드

로커스체인의 노드는 인터넷에 접속된 컴퓨터 프로그램이다. 다른 노드와 통신하며 P2P 방식으로 정보를 교환한다.

노드에는 노드를 대표하는 로커스체인 어카운트가 존재한다. 이 어카운트를 노드 소유자 (owner) 어카운트, 혹은 대표 어카운트라 한다. 어카운트는 항상 어떤 특정한 한 샤드에 소속되고, 노드는 그 소유자 어카운트가 소속하는 샤드에 속하게 된다.

로커스체인의 네트워크에 참여하는 모든 노드는 기본적인 트랜잭션 전송 및 검증 역할을 수행할 필요와 의무가 있다. 다른 블록체인에서 볼 수 있는, 제한적인 역할만을 수행하는 경용량 노드 (light node)는 존재하지 않는다.

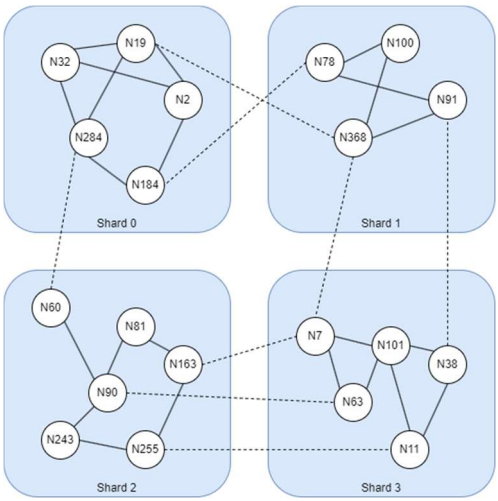


Figure 4: 로커스체인 노드 샤딩과 P2P 통신 개념도

(10) P2P통신 네트워크

로커스체인의 통신망은 노드간의 직접 통신 (Peer-to-Peer Communication, P2P 통신)으로 유지된다. P2P 통신은 일반적으로 사용되는 상식적인 수준의 P2P 통신이다. 다시 말해, 노드에서 발생한 정보는 연결된 모든 주변 노드에 대해 발신되고, 어떤 노드가 다른 노드로부터 수신한 정보는 연결된 다른 노드에 대해 재발송된다. 만약 다른 노드로부터 정보의 요청이 들어왔을 경우 해당 정보를 갖고 있다면 특별한 이유가 없는 한 응답 송신한다.

각 노드는 보통 4~8개 정도의 다른 노드와 P2P 접속을 유지한다. 접속은 수 분 ~ 수십분 정도의 비교적 긴 시간 동안 지속된다. 접속한 노드 중 절반 정도는 같은 샤드 내의 노드이고, 나머지는 다른 샤드와의 노드이다.

(11) 샤드 네트워크 분할

어떤 노드가 다른 노드와 접속을 새로 개설(오픈)할 때 상대방 노드의 대표 어카운트 정보로부터 상대방 노드의 소속 샤드를 알 수 있다. 따라서 상대방 노드의 소속 샤드에 따라 송수신하는 정보를 달리함으로써 실질적으로 네트웍을 샤드 분할할 수 있다.

예를 들어, 샤드에서 새로 발행된 트랜잭션은 같은 샤드 내의 노드에 대해서는 무조건적으로 전송하지만 다른 샤드의 노드에 대해서는 기본적으로는 송신하지 않는다. 그러나 신규 발행 트랜잭션이 다른 샤드의 어카운트에 대해 발행되고, 접속된 노드 중에 해당 샤드로 통하는 노드가 있는 경우는 트랜잭션을 전달한다.

4. 합의알고리즘

A. 샤드 합의 Shard Consensus

샤드 원장은 샤드에 속하는 어카운트의 트랜잭션의 집합이다. 어떤 샤드에 속하는 어카운트가 트랜잭션을 발행하면 그 샤드의 원장에 추가되어야 한다. 이 동작을 위해서는 발행된 트랜잭션의 정보가 샤드에 속하는 노드 대부분에 대해 안정적으로 전달되었다는 사실을 확정짓는 방법이 필요하다.

이 때, 로커스 체인의 원장이 샤드로 분할된다는 점은 중요한 고려 사항이 된다. 분할된 서로 다른 샤드간에는 기본적으로 정보가 공유되지 않는다. 따라서 어떤 샤드가 어떤 다른 샤드로부터

온 트랜잭션을 사용하기 위해서는 그 트랜잭션의 정당성(validity)을 보증할 수 있는 충분한 검증 정보(proof)를 같이 받아서 정당성을 검증할 필요가 있다.

이러한 검증 정보는 제공하는 시점 이후로 변하지 않는 최종적인 상태로 고정될 필요가 있다. 이를 위해 로커스 체인에서는 복수 노드에 의한 합의를 통해 완전하고 불변하게 블록 상태를 확정하는 방법을 채용한다.

로커스 체인은 샤드별로 발행된 트랜잭션을 모아서 합의 블록을 생성, BFT 합의 알고리즘을 통해 확정적으로 결정짓는 방식을 채용하고 있다. 샤드 내에서 블록을 생성하여 서명, 확정하는 과정을 샤드 합의라고 한다.

로커스 체인의 샤드 합의 블록 생성은 복수의 리더 노드가 존재하는 BFT 합의 알고리즘을 통해 실행된다. 합의에 참여하는 노드는 가중 지분증명(Weighted Proof-of-Stake)방식 랜덤 함수를 통해 선출된다.

BFT합의 이전 단계로, 중복된 더블스펜딩 상태의 트랜잭션이 검출된 경우 이를 해소하기 위해 샤드 내의 노드 전체가 참여하는 트랜잭션 전송 단계의 수렴 투표 합의가 실행된다.

각 샤드의 합의는 샤드별로 독립적으로 실행된다.

(1) 합의 이전 단계: 트랜잭션의 정당성 (validity) 평가

어카운트는 정보를 트랜잭션 형식에 담아 이에 서명함으로써 트랜잭션을 생성한다. 어카운트는 임의의 내용을 담은 트랜잭션을 생성할 수 있고, 따라서 더블스펜딩 트랜잭션과 같은 부정직한 트랜잭션을 생성하는것이 기술적으로 가능하다. 블록체인 시스템에서는 과거의 합의된 블록체인의 맥락에 부합하는 트랜잭션만을 정당하다(valid)고 판단한다. 정직한 참여 노드는 정당한 트랜잭션만을 처리에 포함시키고 부정(invaild)한 트랜잭션은 각하 혹은 무시하여야 한다. 로커스 체인 역시 블록체인 기술에서 보아 상식적인 방법으로 트랜잭션의 정당성을 판별한다.

로커스 체인의 각 어카운트 및 노드는 형식적이고 절차적인 방법을 통해 트랜잭션의 정당성을 판단한다. 먼저 모든 트랜잭션은 유효한 어카운트의 비밀키로 서명이 되어야 한다. 같은 어카운트의 모든 트랜잭션은 순서대로 1씩 증가하는 0을 포함한 자연수의 일련번호를 가져야 한다.

아직 합의가 되지 않은 트랜잭션은 어카운트의 마지막 합의된 트랜잭션으로부터 연속된 번호가 부여되어야 한다. 같은 번호를 가진 트랜잭션이 존재하는 경우 합의 과정을 통해 그중 하나의 트랜잭션이 선택된다.

합의가 완료된 트랜잭션은 합의된 샤드 블록 체인 및 월드 상태 체인에 그 트랜잭션의 정보가

포함되어 있어야 한다.

로커스 체인에서 어카운트가 임의로 트랜잭션에 담은 내용은 기본적으로 정당성 판별에 사용되지 않는다. 트랜잭션은 임의의 정보를 담을 수 있고 이러한 임의의 정보의 내용을 로커스 체인 참여자가 제3자 입장에서 평가하는것은 적절하지 않고 또한 가능하지 않기 때문이다. 하지만 코인량 등 로커스 체인에서 정의되어 공개적으로 알 수 있는 정보는 정당성 평가에 이용된다.

(2) 트랜잭션 전송 과정의 중복 검출 및 수렴 투표 경쟁 합의

새로 발행된 신규 트랜잭션은 샤드 내에서 P2P통신을 통해 전파된다. 어떤 노드가 수신한 서로 다른 신규 트랜잭션이 같은 어카운트와 같은 트랜잭션 번호를 갖는 경우 그 트랜잭션들은 중복된 트랜잭션이 된다. 같은 트랜잭션 번호를 갖는 중복된 복수의 트랜잭션 중 샤드 전체에 가장 많이 전파된 트랜잭션 하나만이 정당하고, 나머지는 부적절한 트랜잭션으로 간주되어 제거된다.

중복된 트랜잭션 중 한 트랜잭션을 선택하는 기능으로서 트랜잭션 전송 과정에서의 중복 검출 투표를 통한 경쟁 합의가 실행된다. 이 과정은 샤드 내의 모든 노드가 참여하는 과정이고, 중복된 트랜잭션이 발생하는 순간부터 비동기적으로 자동 진행된다.

중복 트랜잭션 검출 투표 메시지

노드가 한 라운드 내에서 같은 어카운트의 같은 트랜잭션 번호를 갖는 둘 이상의 트랜잭션을 발견하면 그 노드는 중복된 트랜잭션을 발견하였다는 중복 검출 투표 메시지를 P2P네트웍을 통해 샤드에 전파시킨다. 중복 검출 투표 메시지에는 그 노드가 발견한 중복된 트랜잭션의 정보와, 이들 중 어떤 트랜잭션을 가장 먼저 수신하였는지가 포함되어 있다.

각 노드가 샤드 전체에서 발생한 중복 검출 메시지를 집계하면 어떤 트랜잭션이 더 많이 전파되었는지 대략 파악이 가능하다. 노드는 자기 자신이 가장 먼저 수신한 트랜잭션이 아니라 집계에 의해 다수가 된 트랜잭션을 우선한다. 블록 후보 생성 노드는 이 중복 검출 메시지의 집계를 블록 생성에 활용하여 중복된 트랜잭션 중 가장 많은 득표를 한 트랜잭션을 선택하여 블록에 포함시킨다.

중복검출 메시지가 발생하지 않은 트랜잭션의 실질적 확정화

중복된 두 트랜잭션의 발생 시간이 일정 이상 떨어져 있다면 실질적으로 첫번째 트랜잭션이 지배적인 우선순위를 갖게 된다. P2P통신의 전제를 통해 트랜잭션은 발생시점으로부터 랜덤 그래프의 직경 D정도의 전송 스텝 후에 샤드 전체에 전달된다. 따라서 중복된 트랜잭션 중 첫번째 트랜잭션의 발생으로부터 D스텝만큼의 전송이 이루어져 대부분의 노드에 전달된 이후에는, 두번째 이후의 트랜잭션의 발생으로 인해 발생되는 중복 검출 메시지에 아주 높은 확률로 첫번째 트랜잭션이 우선이라 기록되고, 따라서 두번째 트랜잭션이 중복 검출 메세지의 집계 결과를 뒤집는 것

은 실질적으로 불가능하다.

따라서 정직한 노드는, 수신한 어떤 트랜잭션이 샤드 내에 완전히 전달되기에 충분한 시간 동안 중복이 발생하지 않는다면 그 트랜잭션이 확정적이라 생각하고 처리를 계속 진행하는 것이 가능하다. 로커스 체인에서 샤드 내에 트랜잭션이 충분히 전달되는데 필요한 시간은 약 10초 정도로 설정된다. 다시 말해 어떤 트랜잭션은 수신 후 약10초 동안 중복검출 메시지가 발생하지 않는다면 블록 생성 합의 과정이 시작되기 이전에라도 실질적으로 유효하고 확정적이라 간주하고 처리가 진행된다.

대부분의 사용자 어플리케이션에서는 실질적 확정 시간이 지난 트랜잭션을 그대로 유효로 보고 처리를 진행하여도 충분하지만, 샤딩과 프루닝의 기준점으로삼기 위해서는 샤드 내의 원장 전체 상태에 대해 되돌릴 수 없는 최종적인 상태를 기록할 필요가 있으며, 이를 위해서 BFT 합의를 수행한다.

(3) 로커스체인 BFT합의의 전제 사항

로커스체인의 합의 알고리즘을 실제 인터넷 상에 존재하는 노드들 간에 실행하는데 있어서 다음과 같은 통신상의 전제조건을 기대한다.

P2P통신의 안정성

로커스체인은 P2P 통신을 통해 트랜잭션 등의 정보를 교환한다. 따라서 로커스체인에서는 발행된 트랜잭션이 P2P 통신을 통해 충분히 짧은 시간 내에 샤드 내의 노드 대다수에 전달될 수 있다고 가정한다. 구체적으로, 샤드 내의 노드간 통신망의 연결 상태는 랜덤 그래프(random graph)에 가깝고, 네트워크 내의 임의의 두 노드간의 거리는 랜덤 그래프의 직경 (diameter) D 이하이다. 따라서 어떤 노드에서 발생한 정보는 $D+\alpha$ 스텝의 통신을 통해 샤드 내의 모든 노드에 전달될것으로 기대된다.

노드 시계의 정확성

로커스체인의 합의 알고리즘은 타임아웃 등 실시간에 의존하는 처리를 포함한다. 따라서 로커스체인에 참여하는 대다수의 노드는 비교적 정확한 실시간 시계를 갖고 있다고 가정한다. 구체적으로, 노드의 시계 오차는 네트워크 통신시에 처리하는 상식적인 타임아웃 이내의 오차를 갖고 있다고 가정하고, 실질 10 초 이내의 오차를 갖고 있다고 기대한다. 시간이 충분히 정확하지 않은 노드는 적대적(byzantine)인 노드라 간주한다.

정직한 노드의 분포

로커스체인에 참여하는 노드는 적대적인 행위를 할 가능성이 있다. 부정한 트랜잭션을 발행하거나, 합의에 부정직한 투표를 실시하는 등의 행위를 상정한다. 그러나 충분히 많은 수의 노드는 로커스체인의 운용을 위해 정직하게 행동한다는 전제를 갖는다.

특히 합의 과정에 있어, 랜덤으로 선택된 노드 중 최소한 $\frac{2}{3}$를 초과하는 노드가 항상 정직하다고 가정한다. 로커스체인의 합의 과정은 합의 실패 및 재시도를 용인하며, 완전한 내용으로 합의에 도달하는 것 보다 잘못된 내용으로 합의에 도달하는 것을 막는 것을 우선한다. 잘못된 합의를 막을 수 있다면 재시도를 통해 블록 체인 생성을 계속 진행하게 된다. 전체 투표 노드 중 $\frac{2}{3}$를 초과하는 노드가 정직하다면 잘못된 내용으로 합의가 성공하는 것을 막을 수 있다.

(4) 합의 블록

합의 블록은 트랜잭션을 확정시키기 위한 정보 구조이다. 일반적인 블록체인에서 말하는 블록과 역할이 크게 다르지 않으나, 트랜잭션들 자체를 포함하고 있지 않고 샤드의 루트 해시값만 갖고 있다는 점은 차이가 있다. 컨택스트에 따라 라운드 블록 또는 라운드 상태라고도 지칭된다.

블록에 포함되는 가장 중요한 정보는 생성된 트랜잭션의 해쉬값의 리스트이다. 추가로 블록체인과 샤드의 상태를 나타내는 정보로서 앞 블록의 해쉬값, 샤드 전체의 지분량 변화, 어카운트의 변화 내역 등이 추가로 들어간다.

블록은 적절하고 검증 가능한 방법으로 암호학적으로 서명됨으로써 확정된다.

(5) 합의 시간 구분 단위

합의 라운드

로커스 체인에서 블록은 특정 시간 단위마다 생성된다. 현재 2 분마다 하나의 블록을 생성한다. 다시 말해 약 2 분에 한번씩 합의 알고리즘이 실행되고, 한 블록에는 2 분 분량의 트랜잭션이 포함된다. 이 2 분 단위를 한 라운드(round)라 한다.

합의 알고리즘은 라운드 종료로부터 40 초가 지나면 시작된다. 이 40 초는 라운드 종료 시각 직전에 발생한 트랜잭션이 샤드 내에 충분히 전달되게 하기 위한 여유 시간이다.

에포크

또다른 시간 단위로서 에포크(epoch)가 있다. 에포크는 편의상 샤드의 내부 상태를 고정하기 위한 단위이다. 에를 들어 합의 노드를 선출하기 위한 지분증명 가중치는 매 에포크 시작 시점에 결정되고 에포트 도중에는 바뀌지 않는다. 현재 한 에포크는 24 시간, 720 라운드 단위이다.

(6) 합의 커미티

한번의 합의에는 평균적으로 50 정도의 노드가 참여한다. 합의 커미티 노드는 기본적으로 가중치가 고려된 랜덤에 의해 선출된다.

합의 참여 노드의 종류

합의에 참여하는 노드에는 블록 제안 노드 (proposer node)와 투표 노드 (voter node) 또는 커미티 노드가 있다. 블록 제안 노드는 후보 블록을 생성하는 역할이다. 커미티 노드는 생성된 후보 블록에 대해 합의 알고리즘을 실행, 적절한 블록에 대해 서명을 생성하여 블록을 확정한다.

합의 참여 노드의 선출

특정 노드의 블록 제안 노드 선출 여부는, 에포크 시작 시점에 그때까지의 stake 변화에 의거한 고정값과 현재 라운드 번호, 노드의 대표 어카운트 주소로부터 파악한다. 이는 각 노드가 계산한 선출값을 비교함으로서 실행된다. 커미티 노드 선출의 경우 여기에 실제로 커미티에 참가해 활동한 기록 정보를 추가로 사용한다.

각 노드는 라운드마다 적절한 검증 가능한 방법으로 자기 자신의 "선출값"을 계산한다. 만약 선출값이 충분히 큰 경우, 이 선출값을 P2P 통신을 통해 샤드내에 공유한다. 선출값은 임의의 노드에 의해 검증이 가능하다. 각 노드는 다른 노드들의 선출값을 검증, 적절한 기준으로 소팅, 상위로부터 적절한 갯수를 선택하여 새로 선택된 노드를 파악하게 된다. 커미티 노드는 일정 숫자를 선택하나, 선택된 블록 제안 노드의 갯수는 라운드에 따라 달라질 수 있다.

한번 커미티에 선출된 노드는 특정 라운드 동안 커미티 자격을 유지한다. 현재 선출된 노드는 7 라운드동안 커미티 참여가 가능하다. 다시 말해 매 라운드마다 전체 커미티 노드의 약 14.3%가 변경된다.

합의 과정의 통신

합의 과정에서 합의 그룹의 노드들은 노드간의 직접 통신에 준하는 방법으로 긴밀히 통신한다.

합의에 참여하지 않는 노드는 합의 과정 도중에 발생하는 통신을 볼 수 없다. 합의 최종 결과만을 볼 수 있다.

B. BFT 합의 알고리즘

로커스 체인의 합의 알고리즘은 4 단계로 진행된다. 합의에 참여하는 노드는 실시간 시계에 맞춰 각 단계를 자율적으로 실행한다. 각 단계는 10 초 간격으로 실행되고, 단계 진행의 동기를 위한 별도 통신은 발생하지 않는다.

합의 단계는 다음과 같다.

- 블록 제안 노드가 블록 후보를 생성
- 투표 노드가 1 차 투표로 블록을 선택
- 투표 노드가 2 차 투표로 블록 선택을 확정
- 블록 서명

(1) 라운드 블록 후보 생성

라운드 종료 후 일정 시간이 지나면, 각 블록 후보 생성 노드는 노드가 해당 라운드에서 수신한 트랜잭션을 모아서 블록 후보를 생성한다.

블록 후보는 머클 트리(Merkle Tree)상에 배치된 각 트랜잭션의 해쉬값이다. 블록 후보 생성 노드는 수신한 트랜잭션을 적절한 방법으로 배열, 각 트랜잭션의 해쉬값을 배열된 순서로 나열하여 블록을 생성한다.

수신한 트랜잭션이 동일하다면 생성되는 블록 후보는 같다. 매 라운드마다 블록 후보 생성 노드의 갯수 만큼의 블록 후보가 생성된다. 이 경우 모든 블록 후보 생성 노드가 같은 트랜잭션을 수신하였다면 생성되는 모든 블록 후보는 같다.

블록 후보 생성 노드는 생성한 블록 후보를 모든 투표 노드에 송신한다.

부적절한 트랜잭션의 검출

각 후보 노드가 블록을 생성할 때 부적절한 트랜잭션은 포함하지 않는다.

먼저, 각 트랜잭션의 트랜잭션 번호를 확인한다. 트랜잭션 번호는 어카운트가 발행한 트랜잭션에 부여되는 일련번호이고 항상 연속되어야 한다. 따라서 어떤 어카운트의 트랜잭션 번호가 중복되거나 불연속인 경우 이는 부적절한 트랜잭션이다.

번호가 불연속인 경우 불연속되는 번호 이후의 트랜잭션은 무시한다.

번호가 중복되는 경우, 만약 같은 번호의 트랜잭션이 이전의 블록에 존재한다면 새로운 트랜잭션은 무시한다.

같은 번호의 서로 다른 트랜잭션이 동일한 라운드에 신규로 중복 발생한 경우 중복 검출 메시지의 집계를 통해 이중 가장 많이 전파된 트랜잭션이 선택된다.

(2) 라운드 블록 선택 투표 (1차 투표)

블록 후보 생성 노드가 생성한 블록 후보는 각 투표 노드에 전달된다. 각 투표 노드는 수신한 블록 후보와 해당 노드가 직접 수신한 트랜잭션을 비교하여, 자신이 수신한 트랜잭션 결과와 가장 근사한 블록을 선택, 그 블록의 해쉬값에 서명하여 1 차 투표 메시지를 작성한다.

만약 수신한 블록 후보에 자신이 수신하지 않은 트랜잭션이 포함되어 있다면 해당 트랜잭션을 다른 노드에 대해 요청하여 자신의 수신 트랜잭션을 갱신, 블록 선택을 도울 수 있다.

작성된 라운드 블록 선택 투표 메시지는 다른 투표 노드에 전달된다.

(3) 라운드 블록 확정 투표 (2차 투표)

각 투표 노드는 다른 노드와 자기 자신의 1 차 투표 메시지를 집계, 전체 투표 노드 수의 $\frac{2}{3}$을 초과하는 득표수를 갖는 블록 후보를 찾는다. 그러한 블록 후보가 존재하면 각 투표 노드는 해당 블록에 대한 "라운드 블록 확정 메시지"를 발행및 서명한다.

만약 투표 노드수의 $\frac{2}{3}$을 초과하는 투표를 얻은 블록 후보가 존재하지 않는다면 투표 노드는 해당 라운드에 대한 "라운드 블록 확정 실패 메시지"를 발행 및 서명한다.

서명된 라운드 블록 확정 및 확정 실패 메시지는 P2P 로 샤드 전체에 공표된다. 샤드 내의 모든 노드는 각 투표 노드의 라운드 블록 확정 메시지를 받아볼 수 있다.

(4) 블록 확정

샤드 내의 임의의 노드는 어떤 특정 블록 후보에 대해 서로 다른 투표 노드가 발행한 라운드 블록 확정 메시지를 전체 투표 노드수의 $\frac{2}{3}$개 이상 수신하면, 그 블록 후보가 해당 라운드 상태를 대표하는 블록으로 확정되었다고 간주한다.

블록 확정 상태를 영속적으로 남기기 위해, 라운드 블록 확정 메시지는 다음 라운드의 블록에 포함된다.

(5) 블록 생성 합의 실패 판정

만약 노드가 한 노드에 대한 블록 확정 메시지를 $\frac{2}{3}$이하로 수신하고, 라운드 블록 확정 실패 메시지를 $\frac{1}{3}$이상 수신하였을 경우 해당 라운드 합의는 실패한 것으로 간주한다.

구체적이 스텝별로 실패하는 경우는 다음과 같은 경우를 생각할 수 있다.

블록 후보가 생성되지 않는 경우

모든 블록 제안 노드가 블록을 생성하지 않거나, 생성하지 못하거나, 생성한 블록이 제대로 전파되지 않는 경우가 있을 수 있다.

라운드 블록 상태 투표 시작 타이밍까지 블록 후보가 발생하지 않으면 투표 노드는 즉시 라운드 블록 확정 실패 메시지를 발행한다. 샤드 내에 투표 노드 수 $\frac{1}{3}$이상의 라운드 블록 확정 실패 메시지가 발생하면 해당 라운드 블록 생성을 실패로 간주한다.

라운드 블록 상태 투표가 실패하는 경우

만약 투표 노드가 생성된 블록 후보 검증에 전부 실패하는 경우 투표 노드는 라운드 블록 확정 실패 메시지를 발행한다. 실질적으로 블록 후보가 생성되지 못한 경우와 마찬가지로 블록 확정 실패 메시지가 전체 투표의 $\frac{1}{3}$이상이면 블록 확정 실패가 발생한다.

라운드 블록 확정 투표가 실패하는 경우

투표 노드가 충분한 수의 라운드 블록 확정 메시지 수신에 실패하는 경우는 네트웍의 신뢰성 또는 노드의 정직성 전제가 깨지는 상황이다. 블록 확정 실패 메시지를 발행하게 된다.

(6) 합의 재시도

블록 생성 합의에 실패한 경우, 다음 라운드 합의시에 재합의를 실시한다. 다음 라운드에서는 동시에 병렬적으로 앞 라운드와 현재 라운드 두 합의를 각각 실행한다.

재합의가 발생하는 경우 새로 선출하는 합의 커미티의 갯수를 일시적으로 증가시켜 적대적인 노드의 선출 확률을 감소시킬 수 있다.

C. 월드 라운드 상태 합의 World Round State consensus

샤드 원장은 샤드별로 독립적으로 관리되고 실행된다. 따라서 노드는 다른 샤드에서 발행된 트랜잭션이 정상적으로 블록으로 확정되었는지를 직접 파악할 수 없다.

월드 라운드 상태 합의(WRS 합의)는 각 샤드의 라운드별 합의 결과를 통합하여 월드 라운드 상태 블록을 생성하는 과정이다. 월드 라운드 상태 합의 결과는 로커스 체인의 모든 노드에 공유되고, 이를 루트로 삼아서 다른 샤드의 라운드 블록의 정당성과 블록에 포함된 트랜잭션의 정당성을 검증하는데 사용된다.

(1) 월드 라운드 합의 블록

샤드가 생성한 샤드 라운드 블록의 정보를 합하여 생성한 블록이 월드 라운드 상태 블록(WRS 블록)이다.

WRS 블록은 라운드마다 생성된다. WRS 블록에는 앞 라운드에서 합의된 모든 샤드의 블록의 해쉬와 확정 서명 메시지가 포함된다.

합의된 WRS 블록은 모든 노드에 대해 전달된다.

(2) WRS블록 합의 알고리즘

월드 라운드 블록은 월드 라운드 상태 합의 커미티(WRS 합의 커미티)에 의해 생성된다.

사용되는 알고리즘은 기본적으로 샤드 라운드 합의와 같다. 각 라운드마다 앞 라운드에 합의된 모든 샤드의 샤드 라운드 블록의 정보가 월드 라운드 상태 블록 합의 커미티에 전달된다. 이 정보에는 라운드 블록의 해쉬와 라운드 블록 확정 메시지가 포함된다. WRS 합의 커미티에서

생성된 블록 후보에 대해 WRS 합의 커미티가 투표를 통해 WRS 블록 확정 메시지를 서명, 블록을 확정짓는다.

(3) WRS블록 합의 커미티

WRS 블록 합의 커미티는 샤드 합의 커미티의 확장판이다.

WRS 블록 합의 커미티의 멤버는 각 샤드의 기존의 합의 커미티 멤버이다. 각 라운드마다 한 샤드가 랜덤으로 선택되어, 선택된 샤드의 합의 커미티 노드는 WRS 블록 합의 커미티에 추가된다. 샤드 합의와 마찬가지로 추가된 노드는 특정 라운드 후 이탈한다. 현재 7 라운드이다.

5. 동적 샤딩 Dynamic Sharding

로커스체인은 블록체인의 성능 확장성(scability)문제를 동적 샤딩을 통해 해결한다. 발생하는 트랜잭션의 양을 바탕으로 원장 및 네트워크를 샤드로 분할한다. 시스템 전체의 트랜잭션 양이 늘어나면 한 샤드에서 처리하는 양이 어느 이상 늘어나지 않도록 추가로 샤드를 생성한다. 그리고 어떤 샤드의 트랜잭션양이 많아지거나 적어지면, 어카운트 단위로 원장을 샤드간에 이동시켜서 비슷한 크기로 유지시키는 샤드 재구성을 실시한다.

(1) 로커스체인의 동적 샤딩

로커스체인의 샤딩은 트랜잭션 처리량 (throughput)을 늘리는 것이 제 1 목적이다. 한 샤드에서 처리할 수 있는 트랜잭션의 양은 네트웍의 크기 및 참여 노드의 숫자에 따라 한계가 있다. 따라서 시스템 전체의 트랜잭션 량이 늘어날 경우 샤드 수 자체를 늘릴 필요가 있다. 또한, 이상적인 경우 모든 샤드는 같은 수의 트랜잭션을 처리하는 것이 공평하다.

로커스체인의 동적 샤딩은 월드 전체가 이상적인 상태에 가까워지도록 샤드의 갯수와 크기를 알고리즘에 의해 조절하는 동작을 의미한다. 이 동작에는 월드의 상태를 파악하기 위한 월드 상태 계측과, 각 샤드의 크기 밸런스를 조절하기 위한 노드 이동을 통한 샤드 재구성, 그리고 월드 전체의 성능 조절을 위한 샤드 갯수 조절 동작이 포함된다.

(2) 월드 상태 계측 System State Evaluation

샤드 분할 및 재구성을 수행하기 위해 먼저 각 샤드 및 전체 월드의 상태를 파악할 필요가 있다. 월드의 상태는 노드 수, 트랜잭션 발생량, 노드의 지분량 등으로 나타낸다

각 샤드의 처리 부하는 해당 샤드의 트랜잭션 발생량으로 파악된다. 모든 샤드의 트랜잭션 발생량으로부터 샤드간의 처리량 밸런스를 파악한다. 샤드간의 부하량이 차이가 어느 이상으로 커지면 샤드간의 노드 및 어카운트 이동을 통해 부하량의 재분배를 실행한다. 이 때 부차적인 기준으로 각 샤드의 노드수 및 지분량이 가능하면 비슷해지도록 노력한다. 만약 월드 전체의 부하량이 어느 이상으로 늘어나면 샤드가 추가된다.

월드 상태는 에포크 단위로 계측되는 정보와 라운드 단위로 계측되는 정보가 있다. 샤드는 각 에포크 종료에 가까운 시점에 아래와 같은 정보를 수집하여 합의에 기록, 이를 다음 에포크의 PoS 계산 등의 근거로 사용한다.

- 합의 참여 및 샤딩에 고려할 어카운트수

- 어카운트의 stake 량

그리고 각 라운드마다 아래와 같은 정보를 합의에 기록, 샤드 재구성 판단에 사용한다.

- 트랜잭션 발생량
- 어카운트 참여수

(3) 샤드간 어카운트 및 노드 이동을 통한 샤드 재구성

샤드에 어떤 어카운트를 포함할지는 샤드간의 트랜잭션 발생량을 고려하여 결정한다. 어떤 샤드의 트랜잭션 발생량이 많다면 트랜잭션이 발생할 확률이 높은 어카운트를 발생량이 적은 샤드로 이동시킨다.

어카운트의 샤드간 이동시에 어카운트에 관련된 AWTC 원장 정보가 이동후 샤드로 카피된 후 원래 샤드에서 삭제된다. 어카운트가 이동할 때 그 어카운트와 관련된 노드도 같이 이동한다. 이동하는 노드는 자기 자신이 관리하는 어카운트 이외의 정보를 새 샤드로부터 수신하여 새 샤드 네트웍에 참여한다.

노드 및 어카운트의 이동 여부는 각 노드가 개별로 판단한다. 계측된 월드 및 샤드 정보는 모든 노드에 공유되고, 이로부터 동일한 판단 결과가 나오면 각 노드는 판단 결과에 따라 어카운트 이동을 실시한다. 만약 적대적인 노드가 판단 결과에 따르지 않는다면 그 노드는 정직한 노드와의 정보 교환이 실패하게 되고 결과적으로 네트웍에서 도태된다.

어카운트 및 노드의 이동 정보는 합의에 의해 각 샤드의 라운드 블록에 기록된다.

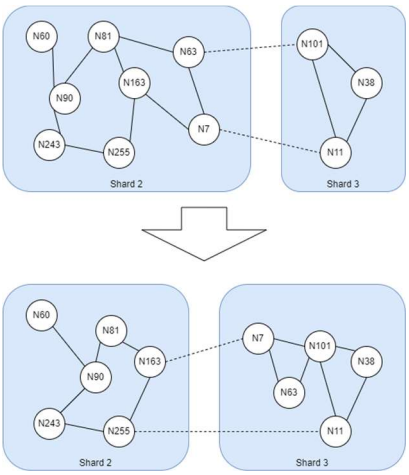


Figure 5: 노드 이동을 통한 샤드 재구성

(4) 샤드 수 변경 shard increment and decrement

월드의 트랜잭션량이 많아지면 샤드수를 늘려서 전체의 트랜잭션 처리용량을 늘린다. 이 때 기존의 샤드를 둘로 분할하여 샤드수를 늘리는 수법이 사용된다.

분할 대상 샤드 번호의 결정

샤드에는 일련번호(ID)가 붙어있다. 새로 샤드를 추가할 때 ID 가 증가되고, 새 샤드의 ID 의 최상위비트가 반전된 샤드를 둘로 분할하여 새 샤드를 만든다. 예를들어 현재 ID 가 0 번부터 4 번까지 5 개의 샤드가 있다면, 추가되는 6 번째 샤드의 ID 는 5 번이 된다. 샤드를 늘리기 위해 십진수 ID 5 의 이진표현 101의 최상위 비트(MSB)가 반전된 1 번 샤드(두번째 샤드)가 분할 대상이 된다. 이 방법을 통해 공평하게 순서대로 샤드 분할이 이루어진다.

샤드 분할전의 사이즈 조절

샤드를 분할할 때 분할 후의 노드 수가 적절한 크기가 되도록 만들 필요가 있다. 이를 위해 샤드 분할이 결정되면 분할 대상 샤드를 다른 샤드보다 1.8 배정도 크게 부풀린다.

분할 대상 샤드 번호가 결정되면, 수 라운드에 걸쳐 분할 대상 샤드에 대해 다른 샤드로부터 노드를 이동시켜, 노드 숫자를 다른 샤드의 1.8 배정도로 늘린다. 이 때 가능하면 트랜잭션 발생이 적은 노드를 이동시켜 트랜잭션 처리량이 비슷하게 유지되도록 노력한다.

분할시에 노드수가 커진 분할 대상 샤드를 둘로 나누면 분할 후 각 샤드는 비슷한 숫자의 노드를 갖게 된다.

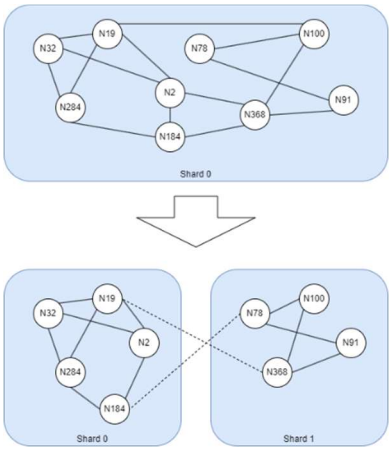


Figure 6: 샤드 분할을 통한 샤드수 증가

(5) 샤드 디렉토리와 홈샤드 shard directory and home shard

샤드 재구성을 통해 어카운트와 노드는 임의의 라운드에 소속 샤드가 변경될 수 있다. 다시 말해 여러 어카운트가 관련된 트랜잭션을 상대방 어카운트에 전달할 때 대상 어카운트가 현재 어느 샤드에 있는지 파악할 방법이 필요하다.

로커스체인에서는 어카운트의 샤드 파악을 위해 어카운트 주소를 바탕으로 한 어카운트 디렉토리 관리 기구를 갖고 있다.

각 샤드는 샤드 번호(ID)를 갖고 있다. 현재 어카운트가 속한 샤드를 워킹샤드라고 부른다. 어카운트의 워킹샤드와 별도로, 어카운트 주소로부터 고정된 샤드 ID를 계산할 수 있다. 예를들어 어카운트 주소를 현재 샤드 갯수로 나눈 나머지를 취하면 샤드 갯수와 같은 갯수의 숫자값을 도출할 수 있다. 이렇게 계산한 어카운트별로 고정된 샤드 ID 를 홈샤드라고 한다. 어카운트가 샤드간에 이동하여 워킹샤드 ID 가 변경되어도 홈샤드 ID 는 변하지 않는다. 따라서 어떤 샤드가 그 샤드를 홈샤드로 갖는 어카운트가 현재 어떤 워킹샤드에 속해 있는지를 디렉토리 관리함으로써 어카운트의 현재 샤드를 파악하는 것이 가능하다.

어떤 노드가 어떤 어카운트의 현재 워킹샤드 ID 를 알고 싶은 경우 그 노드는 자기가 연결된 노드중 해당 어카운트의 홈샤드에 속하는 노드에 질의(query)통신을 요청하여 어카운트의 워킹샤드를 파악할 수 있다.

어카운트의 샤드 이동이 발생하면 이동 전 샤드와 이동 후 샤드 간에 원장 정보의 이동이 발생한다. 동시에 어카운트의 홈샤드에 대해 워킹샤드 디렉토리 정보 갱신을 요청한다.

6. 검증가능 원장 프루닝 Verifiable Ledger Pruning

블록체인 시스템이 정보량 1,000 바이트의 트랜잭션을 초당 1 천개 처리 가능하다 할 때 하루에 발생하는 원장 정보량은 86.4 기가바이트이다. 샤딩에 의해 각 노드가 처리하는 정보량은 $1/(\text{샤드수})$ 로 절감되지만, 그래도 수년의 기간에 걸쳐 발생하는 정보량은 모든 노드가 전부 저장하기에는 현실적이지 못한 양이다.

로커스체인에서는 각 노드 및 어카운트가 직접적으로 관심을 갖지 않는 정보를 노드의 로컬 저장공간에서 제거하는 프루닝을 통해 저장공간을 절약할 수 있다.

프루닝을 통한 저장공간의 절약은 모든 노드가 필수적으로 실행할 필요는 없으나, 로커스체인은 디폴트 상태에서는 노드 대부분이 프루닝을 하는 것을 전제로 움직인다.

(1) 트랜잭션 정보의 관련성 파악

각 어카운트 입장에서 봤을 때 블록체인 시스템 상에서 발생하여 원장에 포함되는 정보의 대부분은 어카운트 자신과는 직접적인 관련성이 없는 정보이다. 원장에는 어카운트의 AWTC DAG 관련 트랜잭션과 블록 생성 합의 관련 정보가 포함되고, 각 정보는 수학적 그래프의 연결 관계로부터 관련성을 파악할 수 있다.

프루닝을 실행하는 각 어카운트 및 노드는 관련성이 적은 트랜잭션을 적극적으로 삭제할 수 있다.

어카운트 AWTC 관련 트랜잭션

로커스체인의 원장은 트랜잭션들이 시간 방향 링크로 연결된 방향 그래프(DAG)로 구성된다. 어카운트가 발행하는 트랜잭션은 항상 같은 어카운트의 직전 트랜잭션을 가리키는 해쉬값으로 구성된 링크를 갖고, 경우에 따라 그 외의 트랜잭션을 가리키는 해쉬값을 통한 링크를 추가로 갖는다. 이러한 DAG 원장 구조 상에서 어떤 어카운트는 자신이 발행한 트랜잭션이 참조하는 과거의 트랜잭션의 링크를 추적함으로써 어카운트 자신과 관련된 모든 트랜잭션을 파악할 수 있다. 이 때 자신의 트랜잭션으로부터 다른 트랜잭션까지의 링크 거리를 계산함으로써 연관 정도를 수치화할 수 있다.

이를 이용하여 어카운트는 자신의 AWTC 원장으로부터 관련성이 어느 이상 먼 트랜잭션을 삭제할 수 있다. 어카운트 입장에서 대부분의 트랜잭션은 링크가 존재하지 않는 거리 ∞ 의

트랜잭션이고, 노드가 스스로 보존할 이유가 없다. 반대로 관련성이 가까운 트랜잭션은 필히 보존하여야 한다.

원장 합의 관련 정보

블록 생성 및 합의에 참여하는 노드는 소속 샤드 내에서 새로 발생한 모든 트랜잭션을 검증해야 한다. 신규 트랜잭션의 검증을 위해서는 신규 트랜잭션이 참조하는 과거의 트랜잭션 정보가 필요하다. 신규 트랜잭션이 필수적으로 참조하는 트랜잭션은 그 트랜잭션을 발행한 어카운트의 직전 트랜잭션이다. 따라서 합의에 참여하는 노드는 소속 노드의 모든 어카운트의 최종 발행 트랜잭션을 검증 가능한 상태로 갖고 있는 것이 적절하다.

신규 트랜잭션이 참조하는 트랜잭션 중 직전 트랜잭션 이외의 기타 트랜잭션은 필요에 따라 취득한다. 노드가 검증에 필요한 트랜잭션을 취득할 수 있다는 최소 보장으로써, 트랜잭션을 새로 발행하는 어카운트는 새 트랜잭션과 함께 검증에 필요한 참조 트랜잭션 및 검증 정보를 제공할 필요가 있다. 이를 통해 검증에 필요한 과거의 트랜잭션이 그 샤드 내에서 완전히 프루닝 된 경우라도 새 트랜잭션의 발행과 함께 (일시적으로) 다시 참조 가능하게 된다.

(2) 프루닝된 과거 트랜잭션의 검증

로커스체인에서 어떤 트랜잭션의 검증에 필요한 다른 과거의 트랜잭션이 프루닝되어 로컬에 존재하지 않는 경우, 이를 다른 노드로부터 새로 수신 받아 다시 정당성을 검증할 필요가 있다.

일반적인 블록체인에서 트랜잭션은 1 차원 링크드 리스트 상에 배치된다. 따라서 어떤 임의의 트랜잭션의 정당성을 검증하기 위해서는 그 트랜잭션이 참조하는 과거의 트랜잭션과 그 트랜잭션을 참조하는 미래의 트랜잭션을 차례로 따라가서 최종적으로 알려진 보증된 트랜잭션에 도달하는 것을 확인할 필요가 있다. 이 검증 작업에는 $O(n)$의 계산량이 필요하다.

로커스체인에서는 트랜잭션을 개선된 데이터 구조 및 검증 알고리즘인 계층적 편향 머클 트리(Hierarchical Skewed Merkle Tree)상에 배치함으로써 $O(\log\ n)$의 계산량으로 트랜잭션을 검증하고 있다.

(3) 계층적 편향 머클 트리

머클 트리는 각 잎 노드(leaf node)가 어떠한 암호학적 해쉬값을 내용으로 갖고, 내부 노드(non-leaf, branch node)는 각 자식 노드(child node)의 내용으로부터 도출된 암호학적 해쉬값을 갖는 트리 데이터 구조이다. 머클 트리는 어떤 리스트에 특정 정보가 존재하는지를 증명하기 위해 널리 사용되는 구조이다.

단순한 편향 머클 트리(Skewed Merkle Tree, **SMT**)는 각 내부 노드가 두 개의 자식 노드를 갖는 머클 트리이다. 자식 노드 하나는 잎 노드(leaf node)이고, 또 하나는 직전의 내부 노드이다. 편향 머클 트리는 머클 트리를 기반으로 한 링크드 리스트처럼 동작한다. 새로운 정보가 SMT 에 추가되는 경우 기존의 SMT 의 루트 노드(root node) 위에 새 루트 노드를 링크함으로써 추가된다. 새 루트 노드의 첫번째 자식 노드는 새로 추가되는 정보를 가리키는 잎 노드이고, 두번째 자식 노드는 기존의 루트 노드가 된다. SMT 의 노드의 검증은 머클 트리의 해쉬값 검증이다. 해당 노드의 자식 노드의 정보로부터 도출된 해쉬값과 노드에 포함된 내용 해쉬값을 비교함으로써 이루어진다. 두번째 링크인 머클 트리의 해쉬값 검증이 재귀적으로 이루어지므로, 실질적으로 과거의 모든 정보의 해쉬값 계산을 필요로 한다.

계층적 편향 머클 트리(Hierarchical SMT, **H-SMT**)의 내부 노드는 자식 노드를 3 개 갖는다. 두개는 SMT 와 마찬가지로 잎 노드 하나와 직전의 내부 노드이다. 세번째 노드는 어떠한 과거의 잎이 아닌 자식 노드의 해쉬값을 갖는다. 이 세번째 자식 노드를 점프 링크(jump link)라 하고, H-SMT 에서는 이 점프 링크를 통해 지수함수적으로 링크드 리스트를 쫓아가게 된다.

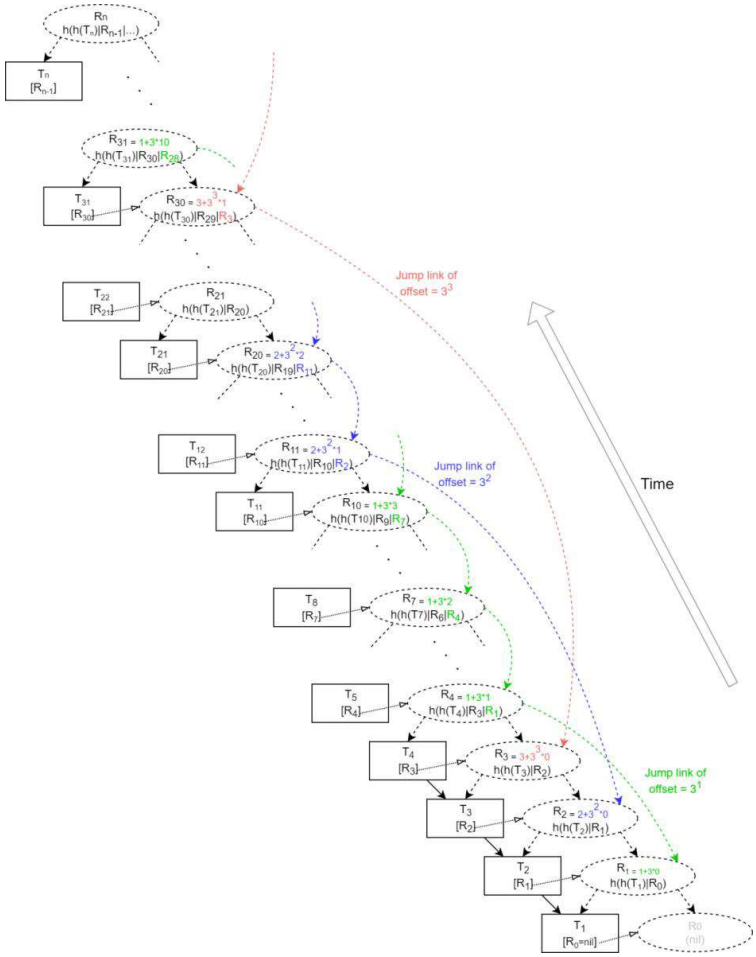


Figure 7: A Hierarchical SMT of base=3

점프 링크 거리 계산

점프 링크는 과거의 다른 머클 트리 노드를 가리킨다. 얼마나 먼 과거의 링크를 참조할지를 결정하기 위해 H-SMT 는 기준거리 b의 거듭제곱값을 사용한다.

기준거리 b인 H-SMT 에서 높이 n 인 머클 트리 노드가 참조하는 점프 링크의 거리는 다음과 같다.

$$\text{거리 d} = -\left(b^{(1+(n-1) \bmod b)}\right)$$

높이 n 인 노드는 계산된 d 만큼 과거의 노드를 참조한다. 예를 들어 기준거리 $b = 3$인 경우, 높이 4 인 노드는 $d = -(3^1)$만큼 떨어진 높이 1 의 노드를 참조한다. 높이 11 인 노드는 $d = -(3^2)$ 만큼 떨어진 높이 2 의 노드를 참조, 계속해서 높이 30 인 노드는 $d = -(3^3)$ 만큼 떨어진 높이 3 의 노드를 참조한다.

H-SMT 노드 검증

점프 링크를 통해 H-SMT 는, 링크드 리스트 대신 머클 트리를 통해 검증되는, 스킵 리스트와 비슷한 데이터 구조로 기능한다.

따라서 H-SMT 에 포함된 정보를 검증하는 것은 해당 정보의 잎 노드로부터 H-SMT 의 루트 노드까지 가는 최단거리 경로를 찾아서 경로에 해당되는 머클 해쉬값을 찾는 문제와 같다.

점프 링크를 통해 한번에 기준거리 b 의 거듭제곱 거리 분량을 건너뛰어 단축할 수 있고,, 따라서 검증에 사용되는 정보의 양은 그래프 전체 높이 h의 로그함수 정도로 줄어든다.

(4) 검증 가능 프루닝 Verifiable Pruning

로커스체인은 트랜잭션을 H-SMT 상에 배치함으로써 임의의 트랜잭션을 $O(\log\ n)$ 계산량으로 검증하도록 구현되어 있다.

트랜잭션의 검증이 필요한 정보는트랜잭션 발행자가 발행 시점에 갖고 있는 정보, 혹은 검증에 성공한 중간 노드가 그 시점에 갖고 있는 정보의 일부를 추출함으로써 즉시 생성 가능하고, 어떤 트랜잭션을 다른 어카운트 및 노드에 송신하는 경우 이 검증 정보를 같이

송신함으로써 트랜잭션의 정당성을 보증할 수 있다. 이를 통해 상대방 쪽의 정보 프루닝 여부와 상관 없이 정당성이 보증되는 정보 교환이 가능하다.

검증 가능한 프루닝의 부차적인 효과로서, 새로 참여하는 신규 노드의 기동 시간이 단축된다. 원장의 정당성 확인에 필요한 프루닝을 고려한 최소한의 트랜잭션과, 트랜잭션의 검증을 필요한 최소한의 검증 정보만으로 노드의 재구성이 가능하다. 실제 환경에서 새 노드의 초기 기동에 필요한 시간은 수십초 정도이다.

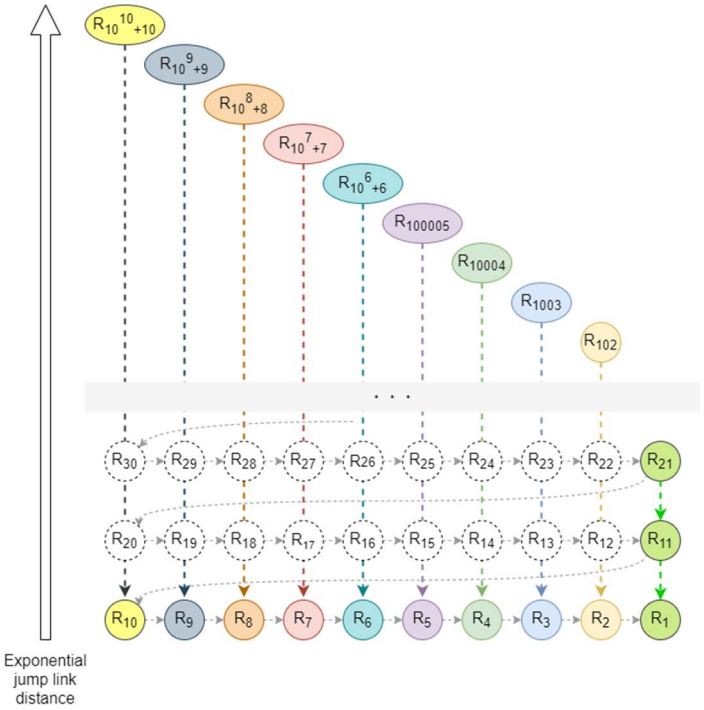


Figure 8: 계층적 편향 머클 트리와 지수함수적 거리를 뛰어넘는 점프 링크

7. 샤드간 통신 Inter-Shard Communication

(1) 샤드간 통신의 필요성

각 샤드는 기본적으로는 독립이고 서로간의 정보를 갖고 있지 않다. 그러나 트랜잭션의 전달 및 어카운트 이동 등에 샤드간의 정보 교환이 필요하므로 샤드간의 통신 방법이 필요하다.

(2) 노드의 샤드간 통신

각 노드의 통신 커넥션의 절반 정도는 샤드 내 통신(intra-shard communication), 나머지 절반 정도는 다른 샤드의 노드와의 통신(inter-shard communication)이다. 샤드 내 통신인 경우 트랜잭션 및 메시지는 별다른 고려 없이 직접 송수신된다. 수신한 메시지의 정당성은 각 노드가 개별적으로 확인한다.

샤드간 통신은 어떤 트랜잭션이 서로 다른 두 샤드에 속한 어카운트 간에 이루어지는 경우에 주로 발생한다. 예를 들어 어카운트 A 가 어카운트 B 에 코인을 보내기 위한 트랜잭션을 발행한다 하자. 이 때 어카운트 A 가 i 샤드 소속, B 가 j 샤드 소속이라 하면 A 의 트랜잭션은 i 샤드에서 발행되지만 B 가 속한 j 샤드에도 전달되어야만 한다. 이 때 i 샤드로부터 j 샤드를 향해 샤드간 통신을 통해 트랜잭션이 전달된다.

다른 샤드와의 통신시 간단한 쿼리 메시지는 별다른 고려 없이 직접 통신 및 실행하지만, 트랜잭션 등 명확한 검증이 필요한 경우 검증에 필요한 정보를 같이 송수신하는 것이 바람직하다.

(3) 샤드간 통신 참여 노드

샤드내의 모든 노드는 샤드간 통신에 참여한다. 어떤 노드가 샤드내 통신을 통해 수신한 트랜잭션이 다른 샤드 j 에 전달되어야 하는 경우, 만약 그 노드가 통신중인 인접 노드중 샤드 j 에 속한 노드가 있다면 그 트랜잭션을 전달하게 된다. 그러나 샤드간 통신만을 위해 새로 통신을 개설하지는 않는다.

노드가 다른 샤드로부터 트랜잭션을 수신하였을 경우 이 트랜잭션을 다시 통상적인 샤드내 통신을 통해 샤드내에 전파한다.

(4) 검증 정보의 강도

트랜잭션을 검증하는데 있어서 강도에 차이가 있는 여러 단계의 검증 정보가 존재한다. 크게 확정 검증 정보(hard proof)와 신속 검증 정보(soft proof)가 있다.

확정 검증 정보 hard proof

확정 검증 정보는 월드 상태로부터 트랜잭션의 합의 여부를 완전히 증명 가능한 정보이다. 예를 들어 트랜잭션이 샤드내 합의에 의해 생성된 블록에 포함된 경우 WRS 블록으로부터 도출 가능한 머클 트리 경로를 함께 송신한다.

완전한 확정 검증 정보는 월드 합의 확정 이후에 검증이 가능하므로, 트랜잭션 생성으로부터 최소 1 라운드 이상의 시간이 지나야만 생성된다. 강력하지만 신속성이 결여되는 경우가 있다.

신속 검증 정보 soft proof

블록에 포함되지 않은 미확정 상태의 트랜잭션을 다른 샤드로 보낼 때, WRS 로부터 확인 가능한 기존의 트랜잭션으로부터 연동되는 정보를 보낼 수 있다. 예를 들어 새로 발생한 트랜잭션의 바로 앞 트랜잭션을 검증하는 정보를 송신함으로써, 트랜잭션이 완전히 날조된 것이 아니라는 힌트를 줄 수 있다.

신속 검증 정보는 어떤 트랜잭션에 대해 완전한 합의 검증을 제공하지는 못 하지만, 합의 실행 노드 등 충분한 정보를 갖고 있는 노드 간에서는 신뢰 가능한 정보로 사용하기에 충분한 검증 능력을 갖는다.

(5) 검증 정보 요청 request for proof

만약 어떤 트랜잭션을 수신한 샤드에 그 트랜잭션을 검증하는데 충분한 정보가 존재하지 않는다면, 송신측에 대해 역으로 검증정보를 요청할 수 있다. 예를 들어 제 3 의 샤드로부터 입수한 트랜잭션이 다른 미지의 트랜잭션을 참조하는 경우, 미지의 트랜잭션이 속하는 샤드에 대해 검증 정보를 요청하여 트랜잭션을 검증할 수 있다.

검증 정보를 요청하는 상대는 임의의 다른 노드가 될 수 있으므로, 노드는 확률적으로 제 3 의 노드를 통해 검증 정보를 요청하여 교차 검증을 실시하는 것이 가능하다.

8. 스마트컨트랙트

로커스체인은 초당 수천 트랜잭션을 처리하기 위한 고성능 블록체인 시스템이다. 이러한 고속 트랜잭션 처리가 어떠한 기술적 난관을 가져오고, 로커스체인이 어떻게 이를 극복하였는지 앞장에서 이미 보인 바가 있다. 마찬가지로, 이러한 트랜잭션 처리 속도 하에서 스마트컨트랙트를 실행하는 경우 기존에 보이지 않았던 문제점이 발생한다.

(1) 스마트컨트랙트 실행 계산 모델

현재의 스마트컨트랙트는 실질적으로 일반 컴퓨터 프로그램과 동등한 기능을 갖는다. 따라서 스마트컨트랙트를 실행하기 위해서는 일반 컴퓨터 프로그램을 여러 개 실행하는 것과 같이 컴퓨터 CPU 및 메모리 등의 하드웨어 성능이 추가로 필요한다. 스마트컨트랙트 실행에 필요한 리소스 양은 실행하는 스마트컨트랙트의 갯수에 비례해서 늘어나므로, 기존 블록체인의 수백배 성능을 목표로 하는 로커스체인에서는 단순 계산만으로도 기존 스마트컨트랙트의 수백배의 CPU 와 메모리를 필요로 하게 된다. 이는 일반 가정용 PC 가 부담할 수 있는 성능을 벗어나고, 기존의 각 노드가 모든 스마트컨트랙트를 실행하는 스마트컨트랙트 모델로는 탑재된 모든 스마트컨트랙트를 전부 실행하는 것이 불가능하다.

로커스체인은 이 문제를 해결하기 위해 스마트컨트랙트 실행에도 분할적 접근을 시도한다. 간단히 말해, 스마트컨트랙트 실행을 여러 그룹이 나눠서 병렬적으로 처리한다. 하나의 그룹은 문제없이 실행 가능한 정도의 제한된 양의 스마트컨트랙트만을 처리하고, 여러 그룹이 서로 다른 스마트컨트랙트를 나누어 병렬적으로 실행함으로써 총합적으로 대량의 스마트컨트랙트 실행을 가능하게 하는 모델이다.

(2) 스마트컨트랙트 어카운트와 실행 그룹

로커스체인에서 실행되는 각 스마트컨트랙트는 고유의 주소를 갖는다. 이 주소는 일반적인 로커스체인 어카운트 주소와 마찬가지의 주소이다.

스마트컨트랙트 실행 그룹은 스마트컨트랙트 어카운트 주소가 발행하는 트랜잭션을 생성하는 역할을 한다. 먼저 일반 어카운트가 스마트컨트랙트 주소에 대해 입력 트랜잭션을 발행함으로써 스마트컨트랙트 실행을 요구한다. 스마트컨트랙트 실행 그룹은 발행된 입력 트랜잭션을 적절한 순서대로 실행하여, 해당 스마트컨트랙트 주소의 트랜잭션을 생성한다. 스마트컨트랙트 실행 결과의 출력은 해당 주소의 AWTC체인으로써 원장에 기록되고, 일반 어카운트의 원장과 동일하게 샤딩 및 프루닝된다.

다시 말해 스마트컨트랙트 실행 그룹은 해당 그룹이 관리하는 스마트컨트랙트를 실행, 결과를 확정지어서 로커스체인 원장에 기록하는 역할을 하는 로커스 체인의 노드 컴퓨터의 그룹이다.

(3) 노드의 스마트컨트랙트 실행 참여

노드는 스마트컨트랙트 실행에 참여할지 여부를 스스로 결정할 수 있다.

로커스체인은 프루닝 및 샤딩을 통해 저성능 저용량 IoT 기기 상에서도 잘 동작한다는 부차적 장점을 갖고 있다. 하지만 스마트컨트랙트 실행에는 원장 검증 기능에 더해 추가적인 CPU 성능과 메모리의 여유가 필요하고, 특정 목적을 위해 컴팩트하게 설계된 IoT 기기 하드웨어에서는 스마트컨트랙트 실행을 위한 추가 성능을 제공할 여유가 없을 수 있다.

따라서 로커스체인에서는 모든 노드에 스마트컨트랙트 실행을 강제하지 않고, 충분한 하드웨어 성능 여유가 있는 노드가 선택적으로 스마트컨트랙트 실행에 참여할 수 있도록 되어 있다. 스마트컨트랙트에 참여하는 노드는 로커스체인 트랜잭션 관리와의 별도로 추가 인센티브가 발생한다. 이를 통해 여유가 있는 노드는 스마트컨트랙트 실행에 참여할 것을 장려한다.

노드는 어떤 스마트컨트랙트 실행 그룹에 참여할지를 스스로 결정할 수 있다. 스마트컨트랙트 실행 그룹은 그룹별로 필요한 하드웨어 요구 사항이 다를 수 있다. 노드는 자기의 실행 성능에 맞는 그룹을 골라 참여할 수 있다.

스마트 컨트랙트 실행 그룹은 참여자에 의해 언제든지 새로 생성될 수 있다. 실행 그룹을 생성하고자 하는 어카운트는 지원 가능한 스마트컨트랙트 언어 VM 환경을 선택, 새로 생성되는 그룹에 초기 자본으로 사용되는 자산을 공탁함으로써 스마트컨트랙트 그룹을 생성한다. 일단 실행 그룹이 생성되면 다른 노드가 비교적 자유롭게 그 그룹에 참여하거나 이탈할 수 있다.

(4) 로커스체인 VME: 스마트컨트랙트 VM실행환경 서비스

로커스체인은 고성능 트랜잭션 처리 기반 위에서 독립적으로 운용되는 복수의 스마트컨트랙트 그룹을 실행함으로써 대량의 스마트컨트랙트를 실행하는 구조를 갖고 있다. 이러한 구조를 위해 로커스체인 원장과 스마트컨트랙트 그룹을 연결짓는 시스템 인터페이스가 존재하고, 이를 "로커스체인 가상 머신 환경 (Virtual Machine Environment, **VME**)"라 한다.

로커스체인 VME 는 스마트컨트랙트 실행결과 트랜잭션 생성용 합의 노드 선정 등을 지원하는 블록체인 레벨의 기능과 노드간의 패킷 통신 지원 등의 네트워크 레벨 기능 등을 포함하는,

일종의 "레이어 1.5" 시스템 서비스이다. 예를 들어, VME 인터페이스를 통해 표준적인 트랜잭션을 발행하는 스마트컨트랙트 실행 그룹 간에는 트랜잭션을 발행을 통한 상호 동작이 손쉽게 가능하다.

각 로커스체인의 스마트컨트랙트 실행 그룹은 완전 독립된 그룹이고 기본적으로 로커스체인 원장 상의 트랜잭션을 통해서만 상호작용한다. 따라서 각 그룹이 서로 다른 기능과 성질을 가질 수 있다. 대표적인 예로 각 그룹은 서로 다른 스마트컨트랙트 언어를 실행하는 것이 가능하다. 그룹은 일종의 스마트컨트랙트 가상 머신(Virtual Machine, **VM**)으로 간주되고, 각 VM 은 로커스체인 VME 인터페이스를 만족하는 임의의 언어 환경을 실행할 수 있다. VM 의 실행 정보는 해당 그룹에 참여하는 노드 간에만 공유되고, 노드는 자신의 실행 능력과 목적에 적합한 VM 을 플러그인하여 원하는 스마트 컨트랙트 실행 그룹에 유연하게 참여하는 것이 가능하다.

로커스 체인은 현재 대표적으로 이더리움과 언어 레벨에서 호환성을 갖는 EVM 엔진을 갖고 있고 그 외에 Move 언어 VM 및 WASM 바탕의 VM 엔진 등 서로 다른 특징을 갖는 여러 엔진의 추가 구현을 검토하고 있다. VME 엔진은 원장과 네트웍 레벨의 서비스를 공통적으로 지원하므로 제 3 자가 새로운 언어 VM 을 개발하는 실험 기반으로 사용될 수도 있다.

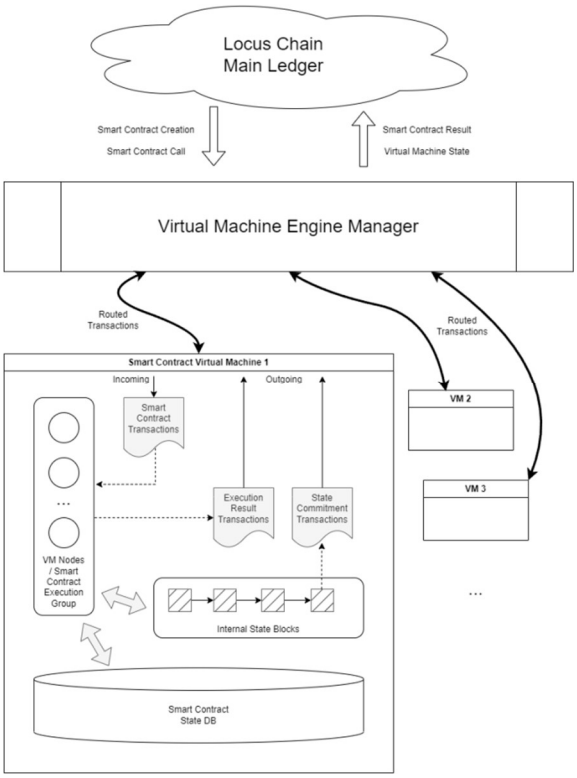


Figure 9: 로커스체인 VME 구조

(5) VME의 “Plug-in”확장 구조를 통한 외부 시스템 연계

일반 사용자 입장에서는 스마트컨트랙트 실행 구조는 일종의 블랙박스로 간주될 수 있다. 사용자가 스마트컨트랙트 어카운트 주소에 대해 요청을 보내면 스마트 컨트랙트가 실행되어 그 결과를 받아볼 수 있다는 사실만이 중요하다. 따라서 스마트컨트랙트 엔진 자체는 필요에 따라 유연하게 설계될 수 있고, 나아가서 스마트컨트랙트 VM 이 외부 정보와 연계되어 결과를 생성하는, 실 세계에 대해 열려 있는 운용이 가능해진다.

외부 정보와 연계하기 위해서는 스마트컨트랙트 실행 그룹에 참여하는 모든 노드가 공통으로 접근하여 참조할 수 있는 외부 정보원(oracle)이 존재하고, 각 실행 노드는 이를 통해 동일한 외부 정보를 참조할 수 있어야 한다. 이러한 기준을 충족하는 것은 현대의 웹 서비스 API 에서 그렇게 어렵지 않은 일이고, 따라서 로커스체인 상에서 어렵지 않게 외부 API 시스템과 직접 연계되는 스마트컨트랙트 프로그램을 구현 운용하는 것이 가능하다.

범용적인 예로, 신용할 만한 외환 환율 기록과 연계하여 자율적으로 거래의 방향성을 판단하는 스마트컨트랙트 계약을 생각해 볼 수 있다. 혹은, 공개적으로 검증이 가능한 다른 블록체인 시스템에서 발행한 트랜잭션을 참조하여 블록체인 상호간에 정보를 교환하는 시스템도 있을 수 있다.

혹은, 게임 등 특정 서비스가 공개하는 API 를 이용하여 로커스체인 어카운트가 게임서비스 내의 리소스를 파악하여 상호작용하는 시스템도 구현 가능하다. 신규 서비스 쪽에서도 로커스체인 상에 트랜잭션을 발행함으로써 로커스체인 상에 존재하는 스마트컨트랙트를 이용하는 것이 가능해진다.

(6) 로커스체인 코어스크립트

로커스체인의 VME 구조는 튜링 완전성을 지닌 컴퓨터 언어를 구동하기 위해 충분한 하드웨어 성능 리소스를 확보한 노드만이 선택적으로 참여하는 것을 전제로 한 구조이다. 하지만, 간단한 트랜잭션 레벨의 처리를 위한 간략화된 스크립트 언어를 운용하는 것은 저성능 장치에서도 충분히 가능하다. 대표적인 예로 비트코인의 스택기반 스크립트 시스템이 있다.

로커스체인에도 코어스크립트(Corescript)라 불리는 간단한 스크립트 시스템이 존재하고, 이를 통해 간단한 트랜잭션 관련 처리가 가능하다. 코어스크립트 명령문은 각 트랜잭션 내에 포함되는

제한된 길이와 제한된 명령어만을 갖는 바이트코드 스크립트이다. 모든 노드는 트랜잭션 처리 시에 의무적으로 이 코어스크립트를 실행한다.

코어스크립트 자체만으로 트랜잭션 발행 등의 고급 처리를 실행하는 것은 불가능하지만, 발행된 트랜잭션의 적용 조건을 판단하는 등의 간단한 조건 판단 처리가 가능하다. 예를 들어 흔히 말하는 아토믹 스왑 등의 처리는 코어스크립트 만으로 구현 가능하다.

9. 암호키

(1) 로커스체인의 암호 키 계층구조

다른 많은 블록체인 플랫폼과 마찬가지로 로커스체인의 어카운트는 유저가 생성한 공개키-비밀키 쌍과 연관되어 있다. 어카운트는 자신의 비밀키를 엄밀하게 관리하여 트랜잭션을 비밀키로 서명하게 된다.

로커스체인에서 어카운트와 연관된 암호키는 마스터키(master key)와 노멀키(normal key), 그리고 검증키(validation key)가 있다.

검증키는 각 노드가 라운드 합의 등에 사용하는 키로, 노멀키로부터 자동으로 생성된다. 합의 알고리즘의 기술적 요구사항을 우선하는 키이고, 빈번하게 재생성되는 것을 상정하고 있다.

노멀키는 트랜잭션의 서명 등 일반적인 시스템 운용에 사용하는 키다. 적절한 암호 강도와 운용 편리성 사이에서 균형이 잡힌 알고리즘을 사용한다. 현재 로커스체인은 타원함수 암호체계를 사용하여 노멀키를 생성하고 있다.

마스터키는 노멀키를 생성하는데 사용된다. 암호 강도를 우선적으로 고려하는 강력한 마스터키는 일상적인 운용에 적합하지 않은 다량의 계산량을 필요로 하는 것을 상정하고 있다. 현재 마스터 키는 양자내성 암호와 타원함수 암호를 복합적으로 적용하는 것을 검토하고 있다.

만약 현재의 노멀키를 이용하는 것이 더이상 적합하지 않은 경우, 어카운트가 마스터키로부터 새로운 노멀키를 생성하여 키 교환 트랜잭션을 발행함으로써 기존의 키를 폐기하고 새로운 키를 등록할 수 있다. 등록된 새로운 노멀키는 합의가 완료된 다음 라운드부터 적용된다.

나아가서 키 교환 트랜잭션을 통한 노멀키 알고리즘 자체의 교환도 고려하고 있다. 장래 양자계산기 등 현재의 암호 알고리즘이 적용 불가능한 상황이 왔을 때를 대비한 구조이다.

(2) 양자내성 암호서명

블록체인 프로젝트에 장래 위협이 될 가능성 중 하나로 양자컴퓨터의 등장을 들 수 있다. 충분한 성능의 양자컴퓨터를 이용하면 현재 주류로 사용되고 있는 많은 서명 알고리즘을 무효화할 수 있을 것으로 예상된다.

다행히도 이에 대비한 양자내성암호(PQC) 연구가 존재한다. 다만 지금까지 발표된 양자내성암호는 현재의 비 양자내성 암호(non-PQC)에 비해 계산량 또는 데이터량이 막대하여,

개인용 PC 나 모바일기기에서 처리하기에는 성능 부담이 있다. 그리고 양자내성암호는 아직 표준이 정착되지 않아 실제 사용했을 때 안전한가에 대한 수학적, 기술적 검증이 부족한 점이 있다.

로커스체인은 이러한 상황을 고려하여 암호키 서명 체계를 마스터 서명과 노멀 서명으로 이원화하여, 일반적인 트랜잭션에서는 현재의 암호체계를 적용한 노멀 서명과 이를 위한 키(페어)를 사용하고, 노멀키를 분실하거나 타인에게 노출되었을 때는 양자내성암호를 적용한 마스터 서명을 사용해 노멀키를 교체하는 방식을 고안했다.

마스터 서명은 꼭 필요한 경우 이외에는 사용하지 않기 때문에 양자내성암호의 데이터량 및 계산량 부담이 적다. 그리고 노멀 서명은 키 이외에 알고리즘 자체를 플러그인 방식으로 교환하는 것이 가능하다. 향후 양자컴퓨터가 상용화되거나 개인용 PC 로 양자내성암호 알고리즘을 처리 가능한 시대가 오면 로커스체인은 노멀 서명 자체를 양자내성 알고리즘으로 교체하는 것이 가능하다. 그리고 양자내성암호에 대한 안전성 자체도 아직 증명되지 않았으므로 마스터 서명은 당분간 양자내성암호와 기존의 암호시스템을 병렬로 사용한 하이브리드 체계로 운영한다는 계획이다. 향후 양자내성암호서명 알고리즘에 취약점이 발견되어도 현용 암호서명 알고리즘으로 커버가 가능하다.

10. 경제 구조(보상, 코인, 그랜트)

(1) 로커스체인 참여자에 대한 보상

로커스체인의 계정과 노드의 자발적인 참여는 로커스체인 시스템의 원활한 운영에 중요한 역할을 한다. 다른 블록체인 시스템과 마찬가지로 로커스체인은 참여하는 노드와 계정에 인센티브를 제공하기 위해 코인 및 그랜트 개념을 도입했다. 로커스체인 시스템에는 다양한 모든 유형의 데이터를 추가할 수 있지만, 코인과 그랜트는 시스템 상 가치를 표시하는 단위로 로커스체인이 정상적인 작동을 하는데 반드시 필요한 특별한 데이터다.

(2) 코인 및 그랜트

코인은 로커스체인 상에 존재하는 가장 보편적 가치 측정 단위다. 코인은 양의 정수의 수치로 표현된다. 코인은 어카운트 간에 증여가 가능하다. 어카운트는 항상 어느 정도의 코인을 갖고 있어야 한다. 어카운트가 생성될 때는 반드시 다른 어카운트로부터 일정량 이상의 코인을 증여받아야 한다. 다시 말해, 제네시스 라운드 이후에 생성되는 모든 어카운트의 첫 트랜잭션은 코인을 증여받는 트랜잭션이 첫 블록이 된다.

그랜트는 로커스체인의 운용에 사용되는 플랫폼 내부 비용 측정 단위다. 예를 들어, 어카운트가 트랜잭션을 발행할 경우 그랜트를 소모하게 된다. 트랜잭션이 스마트컨트랙트 등의 서비스를 이용하는 경우에는 추가적인 그랜트가 필요하다. 이때 어카운트가 충분한 그랜트를 갖고 있지 못하다면 코인을 그랜트로 환전하여 사용할 수 있다.

코인과 그랜트는 공개된 정보이다. 모든 참여자는 과거의 트랜잭션 이력으로부터 모든 어카운트의 코인과 그랜트 소유량을 계산하는 것이 가능하다.

(3) 코인 지분량

코인은 로커스체인 시스템 상에서 지분량(Stake)을 계산하는데 중요한 역할을 한다. 각 샤드에서 라운드 합의를 위해 노드들 중에서 확률적으로 투표인단을 선출하는 과정에서 지분량이 높은 노드는 선출될 확률이 높아진다. 이를 통해 지분량이 상대적으로 더 큰 노드가 시스템 유지에 더 높은 기여를 하게 되는 DPoS[1] 구조를 갖는다.

---

[1] 이오스 에서 사용하는 dPoS 와 로커스체인의 위임 가능 PoS 는 같은 개념이 아니다. 이오스의 경우 합의에 참여하고 보상을 받을 수 있는 노드가 21 개로 제한되어 있기 때문에, 일반 어카운트가 코인을 보유하고 있어도 합의에 참여할 수 없고 사전에 선임된 21 개의 노드 중 한 곳에 강제 위임을 하는 방식이다. 하지만, 로커스체인은 노드의 참여를 제한하지

로커스체인의 어카운트는 자기 지분량을 다른 어카운트에 위탁(Delegate)하는 것이 가능하다. 이를 통해, 네트워크에 항상 접속하지 않는 어카운트가 자신의 지분량을 유효하게 활용할 수 있다. 노드의 지분량은 해당 노드의 호스트 어카운트가 가진 자기 지분량과 다른 어카운트로부터 위탁 받은 지분량의 합계가 된다.

(4) 에포크: 인센티브 계산 단위

코인은 로커스체인을 유지하는데 기여하는 노드 및 어카운트가 받게 되는 인센티브(보상)다. 코인 인센티브는 노드들이 지금까지 라운드 합의에 기여해온 기록을 토대로 미리 정해진 계산식에 의해 산출된다. 따라서 어떤 노드가 인센티브 내용을 계산하여도 같은 계산 결과를 도출하게 된다.

트랜잭션은 라운드 단위로 합의가 이루어진다. 인센티브 계산은 계산의 효율화를 위해 과거의 일정 기간의 라운드들을 모아서 한번에 한다. 현재 로커스체인은 하루 24 시간 분량의 라운드를 모아서 '일 단위의 보상'을 계산하고 있다. 인센티브 계산이 이루어지는 시점을 에포크(Epoch)라 한다. 즉 인센티브의 배분은 에포크 단위로 발생한다.

에포크 계산의 기준이 되는 라운드를 에포크 기준 라운드(Epoch Pivot Round)라 한다. 에포크 계산이 필요한 라운드에, 각 샤드는 에포크 기준 라운드 시점의 원장 정보를 이용하여, 인센티브 계산과 합의 알고리즘에 사용되는 중요한 파라미터인 기준 지분량과 기준 노드 수의 계산을 실행한다. 계산이 완료된 다음 라운드부터, 샤드 내의 모든 노드는 계산된 기준 지분량 수치를 다음 번 에포크로 바뀔 때까지 고정된 값을 적용한다.

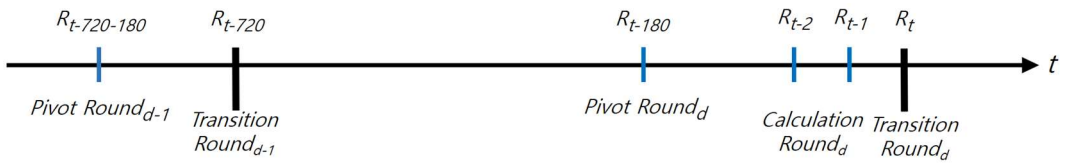


Figure 10: Epoch Pivot Round, Epoch Calculation Round, and Epoch Transition Round

에포크 기준 라운드에 계산된 인센티브 등의 수치는 해당 라운드의 다른 트랜잭션과 함께 합의를 실행하여 원장에 결과를 남긴다. 원장에 기록된 바로 다음 라운드(에포크 전환 라운드)부터 샤드는 새로 계산된 정보를 매 라운드 합의 알고리즘에 사용한다.

않기에 이오스와는 근본적으로 다르다.

인센티브를 수령할 각 어카운트는 에포크 계산 결과에 따라 인센티브 수령 트랜잭션을 발행함으로써 실제로 코인을 받게 된다.

참고로 코인과는 달리 그랜트는 합의에 참여하는 어카운트 및 참여하지 않지만 권한을 참여하는 어카운트에 위임한 어카운트에게 일정한 라운드 단위로 자동으로 계산되어 부여된다. 다만 그랜트는 이 두 경우에 대한 각각의 상한이 있어서 상한 이상으로 축적되지는 않는다.

(5) 인센티브로서의 코인

인센티브는 에포크 단위로 배정된 채굴용 코인을 총액으로 각 어카운트의 로컨스체인에 대한 기여도를 반영하여 분배된다.

로컨스체인 플랫폼의 활성화를 위해 초기에 참여하는 노드에 대해서는 더 높은 기여도를 인정하도록 상대적으로 더 많은 코인을 부여한다. 에포크가 진행될수록, 기발행되어 리저브에 보관중인 로컨스코인으로 지불되는 인센티브는 조금씩 줄어들도록 설계되어 있다.

에포크 단위의 인센티브의 크기는 수수료로 지불된 코인이 환수되는 로컨스코인 리저브의 코인 보유량 변동에 영향을 받는다.

한편 오랫동안 로컨스체인 합의 과정에 주축이 되어 참여해 온 어카운트들에 비해 신규 참여자들이 Stake 배정에서 불평등이 발생하지 않도록, 지분(stake)량과 코인 보유량의 관계가 완전히 비례하지는 않도록 조절한다.

(6) 그랜트

그랜트는 트랜잭션 발행 시 필요한 내부 자원 단위이다. 그랜트는 증여가 불가능하며 상한이 있어 상한 이상으로 가질 수 없다. 그랜트를 획득하는 방법에는 두 가지가 있다. 합의에 참여하는 노드의 호스트 어카운트는 일정 기간마다 합의에 참여한 기여도에 비례해서 그랜트를 지급받는다. 호스트 어카운트는 그랜트의 기간별 상한을 초과할 수 없다. 합의에 참여하지 않고 호스트 어카운트에게 지분을 위임한 게스트 어카운트는 일정 기간마다 정해진 수량의 무료 그랜트를 지급받는다.

그랜트는 어카운트가 소량의 트랜잭션을 코인 소비 없이 사용할 수 있도록 하여 마이크로 트랜잭션을 가능하게 한다. 또한 그랜트 량의 제한을 통해 수많은 트랜잭션의 동시 발송을 막는다. 그랜트가 부족한 경우 코인을 소비해서 트랜잭션을 발송해야 한다.

11. 토큰노믹스 설계 (Tokenomics Design)

(1) 토큰

현재 발행되어 유통중인 로커스토큰은 싱가포르에 본사를 둔 로커스체인 재단이 발행한 이더리움 기반 ERC20 유형이다. 그러나 현행 이더리움 네트워크 기반 토큰을 로커스체인에서 바로 사용할 수는 없다. 로커스체인 코어엔진이 본격 활용되는 시점에 현행 토큰을 로커스체인 기반 코인으로 1 대 1 의 비율로 Atomic Swap 할 예정이다. 로커스체인 코어엔진이 본격 비즈니스 지원을 시작하면 로커스체인 플랫폼 생태계의 확장에 필요한 유동성을 확보하기 위해 새로운 코인체계로 이행할 것이다. 로커스토큰은 일정한 유예기간을 두고 새로운 로커스코인으로 Swap 이 완료되도록 적극적으로 지원할 것이다.

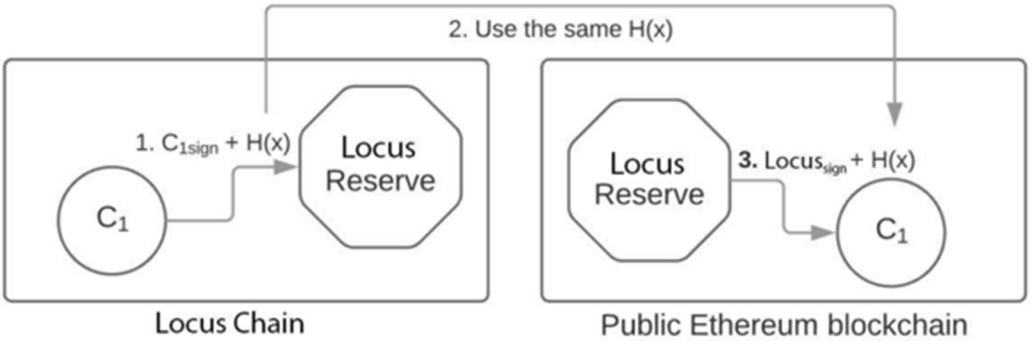


Figure 11: 아토믹 (Atomic Swap) (로커스체인 <> 이더리움 메인넷)

(2) 토큰 경제

로커스체인의 토큰 분배 및 경제 인센티브 구조를 디자인에서는 로커스체인 생태계 가치의 지속성과 플랫폼 구성요소들의 자발적 참여를 유인하는 메커니즘의 효율성을 제고시키는데 가장 중점을 두었다. 이를 설계하는데 다음의 세가지 핵심 개념이 기준이 되도록 노력하였다.

- 가치의 안정성 및 보안
- 생태계 기반조성 및 활성화에 대한 기여도에 기반 인센티브의 타당성
- 소유구조 및 의사결정구조의 분산화를 통한 탈중앙화

로커스체인은 이러한 세가지 기준을 충족시키는데 효율성이 뛰어난 위임지분증명(DPOS) 방식 기반 BFT 합의 매커니즘을 채택하였다.

로커스체인이 사용하는 DPoS 의 주요 특징은 투표인이 네트워크가 형성될 때 사전에 결정되지 않고 네트워크 내의 참여 노드들에 의해 자율적으로 결정된다는 것이다. 이 시스템 내에서는 누구나 원장을 다운로드하고 투표인 참여 기준을 충족하면 노드가 될 수 있다. 현재 디자인된 기준으로는 네트워크에 연결하여 자신의 일정 수준의 지분을 유지하면 노드가 투표인으로 선정될 자격을 갖는다.

이 모델은 일반적으로 인터넷에 연결된 데스크탑 PC 에서 백그라운드를 실행하는 저사양 환경에서도 노드에 참여가 가능한 수준이다. 이 선택은 하드웨어 사양 및 해시 파워에 관계없이 지분 요건과 인터넷 접속 시간 요건을 충족하면 누구나 로커스코인을 채굴하는 노드가 될 수 있다.

어카운트가 지분을 위임하면 "게스트 어카운트(Guest Account)"로 분류된다. 게스트 어카운트는 코인을 채굴하지 않지만 원장을 보관하지 않고도 자신의 지분을 위임한 노드를 통해 거래할 수 있으므로, 이 모델은 노드로 참여할 의사가 없는 코인 보유자들이 채용 가능한 시스템 참여방식이다.

로커스체인 DPoS 는 로커스 코인 채굴에 대한 장벽이 낮고, 누구나 쉽게 채굴할 수 있다. 즉 장비의 성능 등에 따른 참여의 불평등은 존재하지 않는다. 시스템은 궁극적으로 오류 또는 비잔틴 장애(악의적인 해킹 시도)를 방지할 수 있는 프로그래밍 규칙만 허용한다. 이 규칙들을 제외하고 다른 인위적인 개입은 시스템에서 허용되지 않으며, 이는 높은 수준의 탈중앙화를 유지한다.

로커스체인 인센티브 메커니즘의 일반적인 장점은 알려진 바와 같이 우선 처리 속도가 상당히 빠르고 원장 작성에 불필요한 에너지를 낭비하지 않아도 된다. 또한 신규 블록체인 플랫폼 사업자들이 사업초기 당면하는 과제인 안정적이고 지속적인 플랫폼의 성장을 가능케 하는 리소스의 조달에 매우 효과적이다. 이는 플랫폼 디자인 단계에서는 물론 초기 활성화 단계부터 참여에 따른 리스크와 기회비용에 대한 보상을 고려하는데 합리적인 매커니즘의 도입이 가능하다는 점이다. 또한 참여자들의 다양한 기여도에 따른 보상을 반영하는데 용이하다. Stake 된 금전적 가치에만 초점을 맞추는 대신 새로운 플랫폼 구축에 수반되는 리스크를 감내한 초기 참여자들에게 공정한 보상을 함으로써 우리의 플랫폼 제작자들과 플랫폼의 성장을 가능케 해준 이들을 소중히 여긴다는 메시지를 전할 수 있다.

(3) 토큰 배분

총 70 억개의 로커스 토큰이 사전 채굴되었으며 전체 토큰 공급은 공개 참여가 아닌 Private Sales 로 배분될 것이다.

모든 기초 물자는 로커스체인 재단이 소유하며, 비즈니스 혜택을 위해서만 할당된다. 어드바이저와 파트너 업체들을 위한 토큰은 대부분 6 개월에서 24 개월 동안 잠겨서 보관된다.

70 억개의 로커스토큰을 제외하고는 메인 넷의 출시 이후 진행되는 채굴 이외에 더 이상의 추가 발행은 없을 것이다. 로커스토큰은 메인 넷의 출시와 동시에 로커스코인으로 완전히 Atomic Swap 될 예정이다.

업체	토큰 양	비율	설명
Foundation reserved	1,200,000,000	17%	보다 나은 전략적 혜택을 위해 로커스 체인에서 보유하고 있는 토큰
창립자 및 팀	600,000,000	9%	2 년간 lock-up
어드바이저 및 파트너	1,200,000,000	17%	6~24 개월간 lock-up
공헌자 (contributor)	4,000,000,000	57%	
합계	7,000,000,000	100%	

Figure 12: 토큰 배분 수량

로커스코인의 채굴은 코어엔진 기반 플랫폼의 정식 가동과 함께 시작되며, 채굴될 수 있는 코인의 최대량은 50 억개다. 채굴수수료로 지급되는 코인은 로커스코인 리저브에 보관된다. 현재 채굴 메커니즘에서 설정에 의하면 50 년 동안 50 억 개의 추가 로커스코인이 채굴될 예정이다. 따라서 로커스코인의 총 공급량은 120 억 개로 한정된다.

다만, 로커스체인에서 거래수수료는 채굴수수료 지급을 위한 로커스코인 리저브에 회수된다. 회수된 코인은 이후의 채굴 보상량이 결정되는 코인리저브에 영향을 주게 된다. 원래 코인 채굴로 획득 가능한 기간은 50 년이지만 거래수수료가 로커스코인 리저브로 회수가 되므로 리저브의 코인보유가 증가하여 채굴가능 기간이 비례적으로 지속적으로 늘어나게 된다. 따라서 총 공급량 120 억개를 두고 채굴기간은 이론적으로는 무한히 늘어날 수 있다.

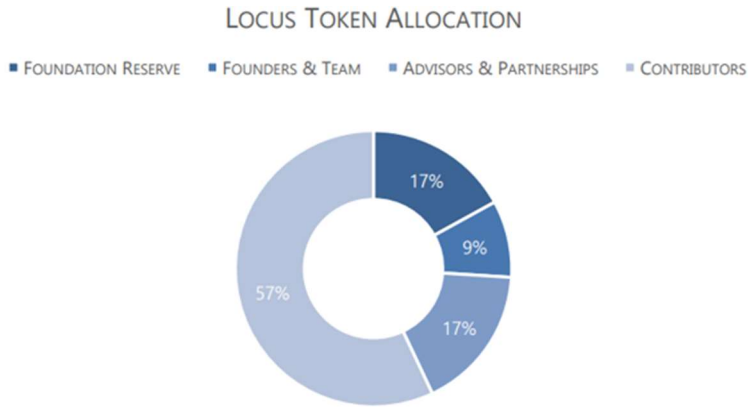


Figure 13: Locus 토큰 배분

(4) 유치된 자금의 활용

Contribution 물량을 판매하여 조성된 재원은 로커스체인을 기획하고 개발, 마케팅, 홍보 등의 사업개발 및 생태계를 조성하는 용도로 사용될 것이다. 블록체인 기술 개발은 프로젝트 성공의 핵심이므로 블록체인 기술개발과 관련된 부분에 기금의 50%를 사용할 예정이다. 나머지 기금의 20%는 로커스체인 생태계를 지원하거나 생태계에 참여하는 다양한 생태계 참여자들에 대한 지원과 생태계 조성비용으로 사용할 예정이다. 15%는 로커스체인의 마케팅과 홍보를 위해 사용할 예정이며, 10%는 프로젝트의 운영을 위한 프로젝트 오퍼레이션 비용, 마지막 5%는 예비로 유보시킬 예정이다.

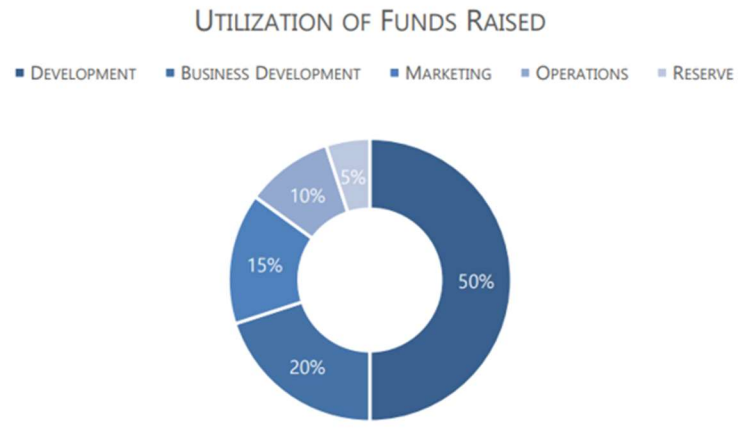


Figure 14: 유치된 자금의 활용

(5) 거버넌스(GOVERNANCE)

로커스체인은 다음과 같은 거버넌스 운영 계획을 가지고 있다. DPoS 프로세스로 설명되는 정책에 의하면, 의결권은 각 노드가 보유한 코인 보유량의 크기에 따라 부여된다. 게스트

어카운트의 보유량이 호스트어카운트에 위임될 때, 의결권도 위임된다. 다양한 정책 제안, 정책 변경, 투표에 관한 세부 사항은 재단에서 제공하는 공식 지갑을 통해 진행된다. 이니셔티브, 제한, 투표, 실행 및 처벌에 대한 자세한 내용은 추후 발표될 예정이다.

12. 로커스 체인 기술의 응용례

로커스 체인은 동적 샤딩, 검증 가능 프루닝 등의 기술 혁신을 통해 PC 및 IoT기기 상에서 고속 대량의 트랜잭션 처리를 가능하게 하였다. 이러한 성능 향상을 통해 로커스 체인은 기존 블록체인의 한계로 여겨져 왔던 실시간 정보교환 및 정보저장 시스템에의 응용이 가능하게 되었다.

로커스 체인은 이론적인 개발을 넘어 이미 응용 프로그램에 구현 탑재가 되는 수준이 되어 있다. 백서 발간일 현재 서버리스 온라인 게임기반인 "로커스 게임 체인", 서버리스 원격 회의 시스템 "로커스 웹미팅" 등의 프로덕트에 이용되고 있다.

나아가 추가로 다른 온라인 게임 프로덕트 및 메타버스 월드, 실시간 거래 플랫폼과 같은 용도에 대한 개발이 진행중이다.

아래에서는 현재 공개된 응용례에 대해 살펴본다.

A. 로커스 게임 체인

(1) 기존 게임 서버 시스템의 한계

기존의 중앙 집중 서버 방식 게임의 한계

일반적인 네트워크 게임 서비스의 구성은 PC 혹은 휴대폰 등의 단말에서 동작하는 **게임 클라이언트 프로그램**과 게임 서비스 회사에서 운영하는 **중앙 서버 시스템**으로 이루어진다. 플레이어가 실행하는 게임 클라이언트 프로그램은 인터넷을 통해 중앙집중 서버 시스템에 접속하여, 통신을 통해 중앙 서버에 플레이어의 게임 내용을 저장하고 실행한다.

이러한 중앙집중 서버 시스템에서는 모든 게임 정보의 저장과 처리는 중앙 서버에 집중된다. 따라서 유저수가 늘어나면 필수적으로 이에 비례하는 고성능의 중앙 서버 시스템을 갖추고 운영해야 하는 부담이 생긴다. 반대로 유저수가 줄어드는 경우 중앙집중 서버 시스템의 높은 운영 비용이 줄어든 수입을 웃돌아 서비스의 조기 종료를 가져오는 원인으로 작용한다.

피어간 통신 (P2P) 방식의 한계

중앙집중 서버 시스템이 아닌 모양으로 게임 서비스를 운영하기 위한 기술적 방법으로 피어간 통신(Peer-to-Peer, P2P)기술의 응용이 있다. 피어(peer)는 인터넷에 접속한 하나의 컴퓨터라는 의미이다. P2P통신기술을 응용한 게임에서는 게임 프로그램을 실행하는 컴퓨터가 중앙 서버에 접속하는 대신 게임 프로그램을 실행하는 컴퓨터끼리 직접 접속하여 통신 및 게임 진행을 하게 된다.

P2P방식에는 명확한 한계가 있다. P2P 방식을 이용하면 중앙 서버의 통신 부담과 게임 진행 부담은 어느 정도 줄어들게 되지만, 반면에 부정행위를 감시하는 중앙 시스템이 존재하지 않으므로 악의를 가진 플레이어들끼리 서로 접속하여 게임 정보를 조작하는 등의 부정행위를 하기 쉬워지는 단점이 있다. 마찬가지로 다른 플레이어의 게임 저장 결과를 신뢰하기 어려워, 게임 진행 내역을 검증 저장하는 역할을 하는 중앙 서버가 여전히 필요하다는 한계를 갖고 있다.

(2) 로커스 게임 체인

로커스 게임 체인은 이러한 기존의 중앙집중 서버 기술과 P2P기술의 한계를 로커스 체인의 분산 원장 시스템을 활용하여 극복한 분산형 게임 서비스 시스템이다.

로커스 게임체인은 로커스 체인 기반기술을 응용하여 개발된 윈도우즈용 분산 서버 어플리케이션이다. 게임 클라이언트 프로그램은 로커스 게임체인에 접속함으로써 기존의 중앙집중형 게임 서버가 제공하는 것과 유사한 데이터 관리, 다른 플레이어와의 통신, 노드간 정보교환 기능 등의 서비스를 제공받을 수 있다.

“서버리스” 네트워크 게임 시스템

로커스 체인의 분산 원장 시스템은 분산 컴퓨팅 환경에서 검증 가능한 통일된 정보의 저장을 가능하게 한다. 다시 말해 중앙 서버가 존재하지 않아도 플레이어의 게임 진행 정보를 안정되고 검증 가능한 방식으로 저장하는 역할을 한다.

분산 환경에서의 정보 저장 기능 (분산 DB기능)은 블록체인 기술의 일반적인 특성이다. 블록 생성작업을 통해 확정된 정보가 올바른 정보 상태로 인정되고, 이를 통해 분산 환경에서 누구나 검증 확인할 수 있는 정확한 정보를 공유하는 것이 가능하다. 이러한 특성을 통해 기존의 거의 모든 블록체인 시스템은 이론적으로 분산 DB로서 이용이 가능하다. 그러나 기존의 블록체인 시스템은 초당 20트랜잭션 정도의 처리량이 한계이고, 따라서 게임 프로그램과 같은 대규모의 플레이어가 동시에 이용하는 프로그램에 기존의 블록체인 시스템을 사용하는 것은 현실적이지 못하였다.

로커스 게임 체인은 로커스 체인의 **동적 샤딩 기술**을 통해 대규모 정보 처리 및 저장을 가능하게 하였다. 게임에 참여하는 플레이어는 곧 로커스 게임 체인에 참여하는 노드가 된다. 플레이어 숫자가 늘어나면 로커스 게임체인 네트워크에 참여하는 노드수가 늘어나고, 늘어난 노드수에 동적 샤딩 기술이 적용되어 자동으로 로커스 체인 샤드수가 증가된다. 로커스 체인은 샤드수에 비례하여 트랜잭션 처리능력이 향상되므로 저장되는 정보의 양도 같이 늘어난다. 그러므로 동적 샤딩 기술이 적용된 로커스 체인에서는 참여하는 플레이어 숫자가 늘어나면 따라서 자연스럽게 정보 처리량이 스케일 확장된다.

또한 로커스 체인의 **검증 가능 프루닝 기술**을 통해 플레이어는 필요 최소한만의 게임 정보를

관리하게 된다. 플레이어가 실행하는 게임 클라이언트는 실시간에 같이 게임을 진행하는 주변 클라이언트의 정보 만을 일시적으로 보유하고, 필요없어진 정보는 프루닝에 의해 삭제하게 된다. 플레이어가 게임을 같이 플레이하는 상대방이 바뀌어도 바뀐 상대방끼리 필요한 게임 정보를 자동으로 교환하여 계속 게임이 진행된다. 이 때 검증 가능 프루닝 기술을 통해 상대방으로부터 수신한 정보가 정당한지가 확인되므로 부정행위를 방지하는 것이 가능하다.

위와 같이, 로커스 게임 체인은 로커스 체인 기술을 응용하여 중앙서버가 존재하지 않는 서버리스(server-less) 네트워크 게임을 구현하는데 사용되는 프로그램이다.

현재 로커스 게임 체인은 리얼타임 전략 시뮬레이션 "킹덤언더파이어"에 적용되어 유명 게임 플랫폼인 "Steam"에 공개되어 있다.

B. 로커스 웹미팅

(1) 로커스 웹미팅: 중앙서버가 존재하지 않는 웹 회의 시스템

로커스 게임체인이 제공하는 정보 저장 및 교환 능력은 게임 이외의 네트워크 어플리케이션에도 적용 가능하다. 로커스 웹미팅은 로커스 체인의 정보 교환 능력을 웹 어플리케이션에 적용한 대표적인 예이다.

로커스 웹미팅은 웹 브라우저 상에서 동작하는 화상 회의 어플리케이션이다.

오늘날 일반적으로 쓰이는 화상 회의 시스템은 먼저 가상적인 회의실을 개설, 이 회의실에 각 사용자가 접속 URL 등을 통해 참여함으로써 회의에 참가하는 형식을 지닌다. 로커스 웹 미팅에서는 이러한 가상 회의실 정보와 사용자 정보의 관리에 로커스 체인 기술을 응용한다. 개설된 가상 회의실 정보는 블록체인 상에 기록되고 이를 통해 다른 참여자가 회의실 상태를 직접 확인하여 회의에 참여하게 된다.

(2) 로커스 웹미팅의 보안상 이점

로커스 웹미팅을 통해 기록된 회의 정보는 보안상 유리한 여러 잇점을 갖는다.

로커스 웹미팅은 로커스 체인 기술의 응용을 통해, 중앙 서버가 존재하지 않는 직접 통신 방식으로 화상회의를 진행한다. 통신 내역은 회의를 진행하는 당사자 컴퓨터 간에만 교환되고, 그 외의 장소에는 전혀 남지 않는다. 제 3의 장소에 정보가 기록되지 않으므로 회의 프로그램 레벨에서는 도청 및 유출의 가능성이 없다.

로커스 웹미팅의 블록체인 상에는 가상 회의실에 관련된 정보가 기록된다. 하지만 로커스 체인의 **검증 가능 프루닝**의 성질상 로커스 체인에 기록되는 정보는 인터넷 상에 아주 작은 흔적만을 남긴다. 로커스 체인의 검증 가능 프루닝에 따라 로커스 체인이 동작하는 컴퓨터 노드는 자기 자신이 직접 필요로 하지 않는 정보는 기록하지 않고 프루닝한다. 다시 말해, 회의에 직접 참여하지 않은 컴퓨터 노드에 회의 정보가 남아있을 가능성은 거의 없다.

나아가, 회의실 정보를 참여자만이 알 수 있는 방법으로 암호화함으로써 제3자가 완전히 알 수 없는 형태로 참여자간에만 회의실 정보를 공유하는 것이 가능하다. 회의실 참여자는 로커스 체인 어카운트 주소로만 파악되고, 회의 도중에 사용한 닉네임 등의 정보는 전혀 알 수 없게 된다. 또한 블록체인에는 IP어드레스 등의 메타정보는 저장되지 않는다. 동시에 검증 가능 프루닝을 통해 암호를 해독할 수 있는 회의 참가자 입장에서는 회의 개설 시간 및 참여자 등의 기록이 객관적으로 증명 가능한 형태로 남겨진다.

부차적으로, 서버 통신 오버헤드 없이 회의 참여자간 직접 통신으로 화상통신을 실행함으로써 고화질 고음질의 회의 진행이 가능하다.

로커스 웹미팅은 분산 원장 시스템의 안전성과 보안성을 직접 웹 회의 솔루션에 적용함으로써 로커스 체인의 확장성을 보여주고 있다. 백서 발간일 현재 로커스 웹미팅은 개발팀에서 시범서비스를 준비 중이다.

13. 맺음말

로커스체인은 탈중앙화의 가치를 훼손하지 않으면서 성능과 확장성 문제를 해결하는 혁신적인 블록체인 플랫폼이다. 로커스체인은 다양한 하드웨어 장치를 사용하는, 다양한 사용자를 위한 고속, 고성능 블록체인 시스템이다.

이 백서에서는 로커스체인 시스템의 탁월한 성능을 지원하는 기술적 측면을 설명했다.

로커스체인은 원장에 병렬성과 유연성을 도입하기 위해 비선형 구조인 DAG 를 채용하였다. 로커스체인의 AWTC(Accountwise Transaction Chain)구조는 어카운트/유저를 중심으로 트랜잭션 그래프를 구성하여 각 트랜잭션을 관리하는 DAG 데이터 구조이다. 각 트랜잭션의 전후 관계와 다른 트랜잭션들과의 관계가 그래프 상에 직접 배치됨으로써 고속 참조가 가능하면서도, 어카운트 단위로 정보를 총합 관리함으로써 샤드간 이동과 통합을 가능하게 하는 데이터 구조이다. AWTC 의 병렬 특성은 원장 샤딩을 현실로 만든다.

로커스체인의 합의 알고리즘은 꼭 필요할 때만 선택적으로 수행되는 트랜잭션 단위의 경쟁 합의에 더해 PoS (Proof-of-Stake)를 기반으로 하는 BFT 합의를 채택하고 있다. PoW 에 의거한 Nakamoto 합의가 가지는 비효율적인 CPU 계산량 소모와 불확정성을 피하기 위한 목적이다. BFT 합의의 고속화를 위해, 로커스체인은 전체 네트워크 노드 중 합의에 참여하는 노드를 공정한 방법으로 확률적으로 샘플링하는 방식을 채용하고 있다. 이 확률적 선출에는 각 노드의 로커스체인 네트워크에 대한 여러 가지 기여도가 반영된다. PoS 를 통해 코인 지분 소유량이 가장 중요한 지표이고, 노드의 온라인 시간 등 코인량 이외의 내용들도 반영된다.

로커스체인은 여러 리소스 문제를 해결하기 위해 다이나믹 샤딩(Dynamic Sharding)을 도입했다. 샤딩은 원장을 분할하여 각 노드의 네트워크 통신 부하를 포함한 데이터 처리 부하를 크게 줄여준다. 각 샤드는 독립적으로 BFT 합의 알고리즘을 수행하며, 총 트랜잭션 처리량은 샤드수가 늘어날수록 비례하여 증가한다. 로커스체인은 샤드의 노드 수를 동적으로 조정하여 샤드 간의 공정성을 보장한다.

베리파이어블 프루닝 기술은 저장 공간 문제를 해결하는 핵심 기술이다. 베리파이어블 프루닝은 원장의 완전성을 유지하면서 각 노드의 저장 공간 요건을 획기적으로 줄여준다. 로커스체인은 IoT 장치 등 저용량 기기에서도 완벽하게 작동할 수 있다.

붙임1. 로커스체인 국내 특허 출원 및 등록 현황

백서 업데이트일(2024년 5월 6일) 기준 10개 출원된 특허 중 4건은 등록완료, 6건은 심사중

A. 등록 완료(4종) 및 국제특허출원중 (2종)

1. BFT 확정 합의 방식의 DAG-AWTC 원장 시스템 (2019년 8월 1일)
2. 원장의 증명 가능 프루닝 시스템 (2019년 8월 1일)
3. 블록체인 네트워크에서 인터샤드 트랜잭션 시스템 및 방법 (2022년 6월 3일)
4. 블록체인 네트워크에서 어카운트의 워킹샤드 변경 시스템 및 방법 (2022년 6월14일)

B. 출원 중 (6종)

1. 블록체인 네트워크에서 검증 가능 인터샤드 트랜잭션 시스템 및 방법 (2022년 7월12일)
2. 블록체인 네트워크에서 다이나믹 샤딩 시스템 및 방법 (2022년 8월17일)
3. 블록체인 합의 시스템 및 방법 (2022년 9월13일)
4. 블록체인 네트워크에서 어카운트 디렉토리 변경 시스템 및 방법(2022년 9월22일)
5. 블록체인 네트워크에서 어카운트 생성 시스템 및 방법(2022년 10월 25일)
6. 블록체인 네트워크에서 노드 분배 시스템 및 방법 (2022년 11월 3일)

특허증
CERTIFICATE OF PATENT

특 허 Patent Number 제 10-2218297 호

출원번호 Application Number 제 10-2019-0093684 호
출원일 Filing Date 2019년 08월 01일
등록일 Registration Date 2021년 02월 16일

발명의 명칭 Title of the Invention
원장의 증명 가능 프루닝 시스템

특허권자 Patentee
주식회사 블룸테크놀로지(110111-*******)
경기도 성남시 분당구 판교로228번길 15, 2동 10층 1003호(삼평동, 판교세븐벤처밸리)

발명자 Inventor
주영현([redacted])
[redacted]

위의 발명은 「특허법」에 따라 특허등록원부에 등록되었음을 증명합니다.
This is to certify that, in accordance with the Patent Act, a patent for the invention has been registered at the Korean Intellectual Property Office.

2021년 02월 16일

특허청장
COMMISSIONER,
KOREAN INTELLECTUAL PROPERTY OFFICE
김 용 래

특허청
Korean Intellectual Property Office

QR코드로 현재기준 등록사항을 확인하세요

특허증
CERTIFICATE OF PATENT

특 허 Patent Number 제 10-2273160 호

출원번호 Application Number 제 10-2019-0093679 호
출원일 Filing Date 2019년 08월 01일
등록일 Registration Date 2021년 06월 29일

발명의 명칭 Title of the Invention
BFT 확정 합의 방식의 DAG-AWTC 원장 시스템

특허권자 Patentee
주식회사 블룸테크놀로지(110111-*******)
경기도 성남시 분당구 판교로228번길 15, 2동 8층 802호 (삼평동, 판교세븐벤처밸리)

발명자 Inventor
주영현([redacted])
[redacted]

위의 발명은 「특허법」에 따라 특허원부에 등록되었음을 증명합니다.
This is to certify that, in accordance with the Patent Act, a patent for the invention has been registered at the Korean Intellectual Property Office.

2021년 06월 29일

특허청장
COMMISSIONER,
KOREAN INTELLECTUAL PROPERTY OFFICE
김 용 래

특허청
Korean Intellectual Property Office

QR코드로 현재기준 등록사항을 확인하세요

특허증
CERTIFICATE OF PATENT

특 허 Patent Number 제 10-2596700 호

출원번호 Application Number 제 10-2022-0068036 호
출원일 Filing Date 2022년 06월 03일
등록일 Registration Date 2023년 10월 27일

발명의명칭 Title of the Invention
블록체인 네트워크에서 인터샤드 트랜잭션 시스템 및 방법

특허권자 Patentee
주식회사 블룸테크놀로지(110111-*******)
경기도 성남시 분당구 판교로228번길 15, 2동 8층 802호 (삼평동, 판교세븐벤처밸리)

발명자 Inventor
주영현([redacted])
[redacted]

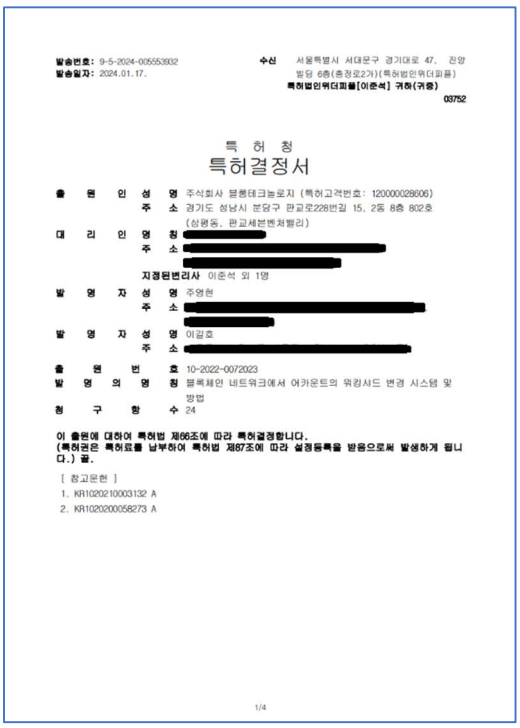
위의 발명은 「특허법」에 따라 특허원부에 등록되었음을 증명합니다.
This is to certify that, in accordance with the Patent Act, a patent for the invention has been registered at the Korean Intellectual Property Office.

2023년 10월 27일

특허청장
COMMISSIONER,
KOREAN INTELLECTUAL PROPERTY OFFICE
이 인 실

특허청
Korean Intellectual Property Office

QR코드로 현재기준 등록사항을 확인하세요



발송번호: 9-5-2024-0055539332
발송일자: 2024.01.17.

수신 서울특별시 서대문구 경기대로 47, 진양빌딩 6층(충정로2가)(특허법인엔더피플)
특허법인엔더피플[이준석] 귀하(귀중)
03752

특 허 청

특허결정서

출 원 인 성 명 주식회사 블룸테크놀로지 (특허고객번호: 120000028606)
주 소 경기도 성남시 분당구 판교로228번길 15, 2동 8층 802호 (삼평동, 판교세븐벤처밸리)
대 리 인 성 명 [redacted]
주 소 [redacted]
지정된변리사 이준석 외 1명
발 명 자 성 명 주영현
주 소 [redacted]
발 명 자 성 명 이광호
주 소 [redacted]
출 원 번 호 10-2022-0072023
발 명 의 명 칭 블록체인 네트워크에서 어카운트의 위킹사드 변경 시스템 및 방법
청 구 항 수 24

이 출원에 대하여 특허법 제66조에 따라 특허결정합니다.
(특허권은 특허료를 납부하여 특허법 제87조에 따라 설정등록을 받음으로써 발생하게 됩니다.) 끝.

[참고문헌]
1. KR1020210003132 A
2. KR1020200058273 A

1/4

붙임2: 로커스체인 국외 7개국 특허 출원 및 등록 현황

백서 업데이트일(2024년 5월 6일) 기준 7개국에 출원된 특허 2건중 4개국 6건에 대해 등록완료, 8건은 심사중

A. 미국

1. BFT 확정 합의 방식의 DAG-AWTC 원장 시스템 (특허진입일: 2022/01/19)
2. 원장의 증명 가능 프루닝 시스템 (특허진입일: 2022년 1월 14일, 등록일: 2024년 4월 2일)

B. 일본

1. BFT 확정 합의 방식의 DAG-AWTC 원장 시스템 (특허진입: 2022/01/18, 등록: 2023/06/27)
2. 원장의 증명 가능 프루닝 시스템 (특허진입: 2022/01/18, 등록: 2023/06/02)

C. 러시아

1. BFT 확정 합의 방식의 DAG-AWTC 원장 시스템 (특허진입: 2022/02/02, 등록: 2023/12/14)
2. 원장의 증명 가능 프루닝 시스템 (특허진입: 2022/02/02, 등록: 2023/02/15)

D. 호주

1. BFT 확정 합의 방식의 DAG-AWTC 원장 시스템 (특허진입: 2022/01/21)
2. 원장의 증명 가능 프루닝 시스템 (특허진입: 2022/01/21, 등록: 2023/12/14)

E. 유럽

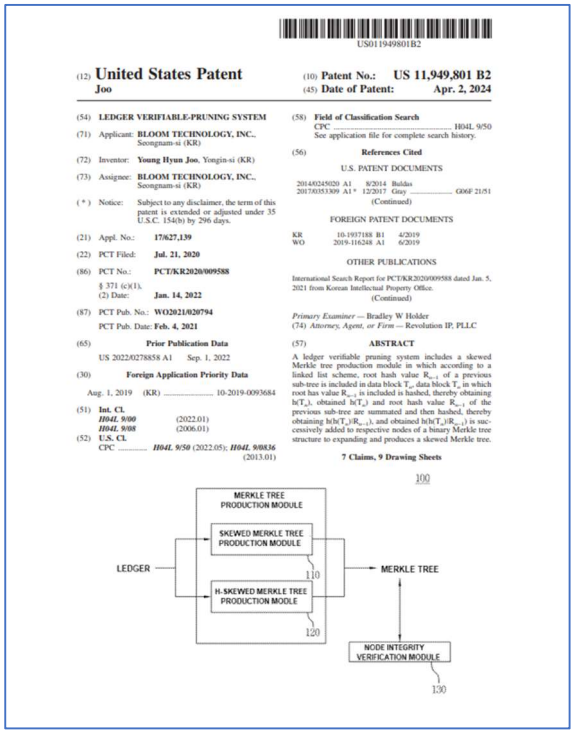
1. BFT 확정 합의 방식의 DAG-AWTC 원장 시스템 (특허진입: 2021/12/29)
2. 원장의 증명 가능 프루닝 시스템 (특허진입: 2021/12/29)

F. 캐나다

1. BFT 확정 합의 방식의 DAG-AWTC 원장 시스템 (특허진입: 2022/01/06)
2. 원장의 증명 가능 프루닝 시스템 (특허진입: 2022/01/05)

G. 중국

1. BFT 확정 합의 방식의 DAG-AWTC 원장 시스템 (특허진입: 2022/01/21)
2. 원장의 증명 가능 프루닝 시스템 (특허진입: 2022/01/21)



US011949801B2

(12) **United States Patent**
Joo

(10) **Patent No.: US 11,949,801 B2**
(45) **Date of Patent: Apr. 2, 2024**

(54) **LEDGER VERIFIABLE-PRUNING SYSTEM**

(71) Applicant: **BLOOM TECHNOLOGY, INC.**, Seongnam-si (KR)

(72) Inventor: **Young Hyun Joo**, Yongin-si (KR)

(73) Assignee: **BLOOM TECHNOLOGY, INC.**, Seongnam-si (KR)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 296 days.

(21) Appl. No.: **17/627,139**

(22) PCT Filed: **Jul. 21, 2020**

(86) PCT No.: **PCT/KR2020/009588**
§ 371 (c)(1),
(2) Date: **Jan. 14, 2022**

(87) PCT Pub. No.: **WO2021/020794**
PCT Pub. Date: **Feb. 4, 2021**

(65) **Prior Publication Data**

US 2022/0278858 A1 Sep. 1, 2022

(30) **Foreign Application Priority Data**

Aug. 1, 2019 (KR) 10-2019-0093684

(51) **Int. Cl.**
H04L 9/00 (2022.01)
H04L 9/08 (2006.01)

(52) **U.S. Cl.**
CPC ***H04L 9/50*** (2022.05); ***H04L 9/0836*** (2013.01)

(58) **Field of Classification Search**
CPC H04L 9/50
See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

2014/0245020 A1 8/2014 Baldus
2017/0353309 A1* 12/2017 Gray G06F 21/51
(Continued)

FOREIGN PATENT DOCUMENTS

KR 10-1937188 B1 4/2019
WO 2019-116248 A1 6/2019

OTHER PUBLICATIONS

International Search Report for PCT/KR2020/009588 dated Jan. 5, 2021 from Korean Intellectual Property Office.
(Continued)

Primary Examiner — Bradley W Holder
(74) *Attorney, Agent, or Firm* — Revolution IP, PLLC

(57) **ABSTRACT**

A ledger verifiable pruning system includes a skewed Merkle tree production module in which according to a linked list scheme, root hash value R_{n-1} of a previous sub-tree is included in data block T_n, data block T_n in which root has value R_{n-1} is included is hashed, thereby obtaining $h(T_n)$, obtained $h(T_n)$ and root hash value R_{n-1} of the previous sub-tree are summated and then hashed, thereby obtaining $h(h(T_n)|R_{n-1})$, and obtained $h(h(T_n)|R_{n-1})$ is successively added to respective nodes of a binary Merkle tree structure to expanding and produces a skewed Merkle tree.

7 Claims, 9 Drawing Sheets

特許証
(CERTIFICATE OF PATENT)

特許第７３１８１０４号
(PATENT NUMBER)

発明の名称
(TITLE OF THE INVENTION)
ＢＦＴ確定合意方式のＤＡＧ－ＡＷＴＣ元帳システム

特許権者
(PATENTEE)
大韓民国キョンギド、ソンナムシ、ブンダング、パンギョロ２２８ボンギル１５、２ドン１０フロア・１００３ホ（サムピョンドン、パンギョ・セブン・ベンチャー・バレー）
国籍・地域　大韓民国
ブルーム・テクノロジー・インコーポレイテッド

発明者
(INVENTOR)
ジュ，ヨンヒョン

出願番号
(APPLICATION NUMBER)
特願２０２２－５０３８２８

出願日
(FILING DATE)
令和　２年　７月２１日(July 21, 2020)

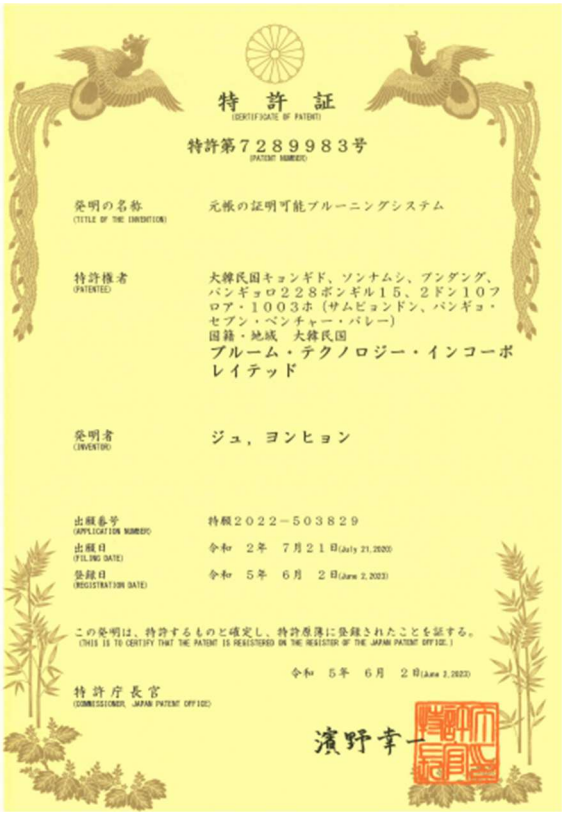
登録日
(REGISTRATION DATE)
令和　５年　７月２１日(July 21, 2023)

この発明は、特許するものと確定し、特許原簿に登録されたことを証する。
(THIS IS TO CERTIFY THAT THE PATENT IS REGISTERED ON THE REGISTER OF THE JAPAN PATENT OFFICE.)

令和　５年　７月２１日(July 21, 2023)

特許庁長官
(COMMISSIONER, JAPAN PATENT OFFICE)

濱野幸一



特許証
(CERTIFICATE OF PATENT)

特許第７２８９９８３号
(PATENT NUMBER)

発明の名称
(TITLE OF THE INVENTION)
元帳の証明可能プルーニングシステム

特許権者
(PATENTEE)
大韓民国キョンギド、ソンナムシ、ブンダング、パンギョロ２２８ボンギル１５、２ドン１０フロア・１００３ホ（サムピョンドン、パンギョ・セブン・ベンチャー・バレー）
国籍・地域　大韓民国
ブルーム・テクノロジー・インコーポレイテッド

発明者
(INVENTOR)
ジュ，ヨンヒョン

出願番号
(APPLICATION NUMBER)
特願２０２２－５０３８２９

出願日
(FILING DATE)
令和　２年　７月２１日(July 21, 2020)

登録日
(REGISTRATION DATE)
令和　５年　６月　２日(June 2, 2023)

この発明は、特許するものと確定し、特許原簿に登録されたことを証する。
(THIS IS TO CERTIFY THAT THE PATENT IS REGISTERED ON THE REGISTER OF THE JAPAN PATENT OFFICE.)

令和　５年　６月　２日(June 2, 2023)

特許庁長官
(COMMISSIONER, JAPAN PATENT OFFICE)

濱野幸一

РОССИЙСКАЯ ФЕДЕРАЦИЯ

ПАТЕНТ

НА ИЗОБРЕТЕНИЕ

№ 2809645

СИСТЕМА DAG-AWTC-РЕЕСТРОВ С ИСПОЛЬЗОВАНИЕМ КОНСЕНСУСНОГО МЕХАНИЗМА BFT-ВЕРИФИКАЦИИ

Патентообладатель: *БЛУМ ТЕКНОЛОДЖИ, ИНК. (KR)*

Автор(ы): *ЧО, Ёун Хюн (KR)*

Заявка № **2022102374**
Приоритет изобретения **01 августа 2019 г.**
Дата государственной регистрации
в Государственном реестре изобретений
Российской Федерации **14 декабря 2023 г.**
Срок действия исключительного права
на изобретение истекает **21 июля 2040 г.**

Руководитель Федеральной службы по интеллектуальной собственности

Ю.С. Зубов

РОССИЙСКАЯ ФЕДЕРАЦИЯ

ПАТЕНТ

НА ИЗОБРЕТЕНИЕ

№ 2790181

СИСТЕМА ВЕРИФИЦИРУЕМОГО ОТСЕЧЕНИЯ РЕЕСТРОВ

Патентообладатель: *БЛУМ ТЕКНОЛОДЖИ, ИНК. (KR)*

Автор(ы): *ЧО, Ёун Хюн (KR)*

Заявка № **2022102390**
Приоритет изобретения **01 августа 2019 г.**
Дата государственной регистрации
в Государственном реестре изобретений
Российской Федерации **15 февраля 2023 г.**
Срок действия исключительного права
на изобретение истекает **21 июля 2040 г.**

Руководитель Федеральной службы по интеллектуальной собственности

Ю.С. Зубов

Australian Government
IP Australia

CERTIFICATE OF GRANT

STANDARD PATENT

Patent number: 2020323807

The Commissioner of Patents has granted the above patent on 14 December 2023, and certifies that the below particulars have been registered in the Register of Patents.

Name and address of patentee(s):

Bloom Technology, Inc. of (Sampyeong-dong, Pangyo seven venture valley), 2-dong 10F 1003-ho, 15, Pangyo-ro 228beon-gil Bundang-gu, Seongnam-si Gyeonggi-do 13487 Republic of Korea

Title of invention:

Ledger verifiable-pruning system

Name of inventor(s):

JOO, Young Hyun

Term of Patent:

Twenty years from 21 July 2020

Priority details:

Number	*Date*	*Filed with*
10-2019-0093684	1 August 2019	KR

Dated this 14th day of December 2023

Commissioner of Patents

PATENTS ACT 1990

The Australian Patents Register is the official record and should be referred to for the full details pertaining to this IP Right.

Bibliography

Association, T. L. (n.d.). *An Introduction to Libra*. Retrieved from https://libra.org/en-us/whitepaper

Bentov, I., Gabizon, A., & Mizrahi, A. (2016). *Cryptocurrencies without proof of work.* Retrieved from https://arxiv.org/abs/1406.5694

Bernstein, D. J. (2015). Multi-user Schnorr Security, Revisited. *Cryptology ePrint Archive; Vol. 2015/996.* IACR.

Bitcoin: what a waste of resources. (2017, 11). Retrieved from New Scientist: https://www.newscientist.com/article/mg23631503-300-bitcoin-what-a-waste-of-resources/

Buterin, V. (n.d.). *Ethereum Sharding FAQ*. Retrieved from Ethereum Wiki: https://github.com/ethereum/wiki/wiki/Sharding-FAQ

Buterin, V., & Gavin, W. (2013). *Ethereum*. Retrieved from https://www.ethereum.org/

Castro, M., & Liskov, B. (1999). Practical byzantine fault tolerance. *Proceedings of the Third Symposium on Operating Systems Design and Implementation (OSDI'99)*, (pp. 173-186).

Chen, L., Jordan, S., Liu, Y.-K., Moody, D., Peralta, R., Perlner, R., et al. (2016). *Report on Post-Quantum Cryptography.* National Institute of Standars and Technology, Internal Report 8105.

Dang, H., Dinh, T. T., Chang, D. L.-C., Lin, Q., & Ooi, B. C. (2019). Towards Scaling Blockchain Systems via Sharding. *2019 International Conference on Management of Data (SIGMOD '19).* ACM.

Demers, A., Greene, D., Hauser, C., Irish, W., Larson, J., Shenker, S., et al. (1987). Epidemic Algorithms for Replicated Database Maintenance. *Proceedings of the Sixth Annual ACM Symposium on Principles of Distributed Computing (PODC '87)* (pp. 1-12). New York: ACM.

Directed Acyclic Graph. (n.d.). Retrieved from Wikipedia: https://en.wikipedia.org/wiki/Directed_acyclic_graph

Gilad, Y., Hemo, R., Micali, S., Vlachos, G., & Zeldovich, N. (2018). *Algorand: Scaling Byzantine Agreements for Cryptocurrencies.* Retrieved from https://algorandcom.cdn.prismic.io/algorandcom%2Fa26acb80-b80c-46ff-a1ab-a8121f74f3a3_p51-gilad.pdf

Lamport, L., Shostak, R., & Pease, M. (1982). The Byzantine generals problem. *ACM Transactions on Programming Languages and Systems, 4.3*, 382-401.

LeMahieu, C. (2017, 11). *Nano: A Feeless Distributed Cryptocurrency Network.* Retrieved from https://nano.org/en/whitepaper

Maymounkov, P., & Mazières, D. (2002). Kademlia: A Peer-to-Peer Information System Based on the XOR Metric. *International Workshop on Peer-to-Peer Systems (IPTPS 2002)*, (pp. 53-65).

Merkle, R. C. (1987). A digital signature based on a conventional encryption function. *A Conference on the Theory and Applications of Cryptographic Techniques on Advances in Cryptology (CRYPTO' 87)*, (pp. 369-378).

Merkle-Patrica Trie Specification. (n.d.). Retrieved from Ethereum Wiki: https://github.com/ethereum/wiki/wiki/Patricia-Tree#main-specification-merkle-patricia-trie

Micali, S., Vadhan, S., & Rabin, M. (1999). Verifiable Random Functions. *IEEE 40th Annual Symposium on Foundations of Computer Science (FOCS'99)*, 120-130.

Mills, D., Wang, K., Malone, B., Ravi, A., Marquardt, J., Chen, C., et al. (2016). *Distributed ledger technology in payments, clearing, and settlement.* Divisions of Research & Statistics and Monetary Affairs, Federal Reserve Board.

Nair, G. R., & Shoney, S. (2017). BlockChain Technology: Centralised Ledger to Distributed Ledger. *International Research Journal of Engineering and Technology, Vol.4, Issue 3*.

Nakamoto, S. (2008). *Bitcoin: A Peer-to-Peer Electronic Cash System*. Retrieved from https://bitcoin.org/bitcoin.pdf

Popov, S. (2018, 4). *The Tangle.* Retrieved from https://www.iota.org/research/academic-papers: https://assets.ctfassets.net/r1dr6vzfxhev/2t4uxvsIqk0EUau6g2sw0g/45eae33637ca92f85dd9f4a3a218e1ec/iota1_4_3.pdf

Proof of Stake. (n.d.). Retrieved from Wikipedia: https://en.wikipedia.org/wiki/Proof_of_stake

Proof of Work. (n.d.). Retrieved from Wikipedia: https://en.wikipedia.org/wiki/Proof_of_work

Schnorr, C. P. (1991). Efficient signature generation by smart cards. *Journal of Cryptology, Vol. 4, no.3*, 161-174.

Shard (database architecture). (n.d.). Retrieved from Wikipedia: https://en.wikipedia.org/wiki/Shard_(database_architecture)

Visa acceptance for retailers - Security and reliability. (n.d.). Retrieved from visa.com: https://usa.visa.com/run-your-business/small-business-tools/retail.html

Wood, G. (2016). *Ethereum: A Secure Decentralised Generalised Transaction Ledger.* Retrieved from http://gavwood.com/paper.pdf

Zamani, M., Movahedi, M., & Raykova, M. (n.d.). RapidChain: Scaling Blockchain via Full Sharding. *Proceedings of ACM SIGSAC Conference 2018*, (pp. 931-948).