

파로스 (Pharos)

해당 번역본은 이용자가 디지털자산 관련 정보를 편리하게 확인할 수 있도록 단순 참고용으로 번역된 것이며, 투자 권유를 목적으로 하지 않습니다.

그 내용이 정확하지 않을 수 있으므로 정확한 정보 습득을 위해서는 원문을 참고하시거나 원문 작성 주체 측에 문의하시기 바랍니다.

두나무는 해당 번역본 내용의 진실성 및 정확성을 보장하지 않고, 제공된 정보에 의한 투자 결과에 대하여 어떠한 책임도 지지 않으며, 투자 손실은 투자자에게 귀속됩니다.

Pharos Network: 확장 가능한 혁신을 구현하는 심층 병렬 레이어 1 네트워크

Pharos Team

1 서론

“우리는 러프 컨센서스와 실행 가능한 코드를 신념으로 삼는다.” – David Clark, 1992

Web3는 차세대 인터넷 프로토콜 및 서비스로서, 소유권 중심(ownership-centric) 및 신뢰 불필요(trustless) 기술의 혁신을 강조합니다. 최초의 Web3 네트워크인 비트코인(Bitcoin)은 전 세계적 합의(global consensus)와 간편 결제 검증(SPV, Simplified Payment Verification) 보안 메커니즘을 통해 15년간의 공개 네트워크 테스트를 거치며, 상태가 없는 인터넷(stateless Internet)을 상태를 가지는 구조(stateful design)로 전환시켰습니다. 최초의 Web3 탈중앙화 애플리케이션 플랫폼인 이더리움(Ethereum)은 이더리움 가상머신(EVM, Ethereum Virtual Machine)을 통해 자산의 프로그래머빌리티를 확장하였으며, ERC 토큰 표준은 사실상 디지털 자산의 표준으로 자리잡았습니다. 최근에는 Solana, Aptos, Sui와 같은 새로운 Web3 네트워크가 등장하여 성능 및 자산 보안 측면에서 Web3 기술을 더욱 발전시키고 있습니다.

결제 네트워크 설계에서 출발하여 가치 인터넷(value Internet)의 새로운 시대를 지향하는 Web3 네트워크는 여전히 전통 금융 시스템에 상응하는 인프라가 부족한 상황입니다. 예를 들어, 2021년 기준 디지털 지갑은 전 세계 거래의 28.6%를 차지하였으며, 이는 수십억 명의 사용자를 기반으로 13.3조 달러 이상의 거래 규모에 해당합니다. 이러한 디지털 지갑 환경에서 슈

퍼 앱(Super-App) 사용자 경험을 향상시키기 위해서는 안전하고 신뢰할 수 있으며 안정적인 실시간 결제 시스템이 필수적입니다.

Pharos는 신뢰 불필요 혁신을 위해 설계된 새로운 Web3 네트워크입니다. Pharos는 암호자산(Crypto Assets)과 실물자산(Real-World Assets, RWA)을 모두 포괄하는 인프라를 기반으로 자산의 활용 가능성을 전례 없는 수준으로 확장하는 것을 목표로 합니다. 이를 위해 Pharos는 성능, 확장성, 보안성, 확장 가능성(extensibility)을 기반으로 고밀도 자산 가치를 지원하는 탈중앙화 인프라를 제공합니다. 또한 실시간 결제, 탈중앙화 애플리케이션(dApps), 거래, 유동성을 지원함으로써, 수십억 계정을 대상으로 우수한 사용자 경험과 확장성을 제공하는 차세대 금융 기술의 기반을 마련하고 있습니다. Pharos는 자산의 연산(compute)과 저장(storage)을 레이어 1(L1) 프레임워크에 직접 통합함으로써 보다 우수한 L1 블록체인을 구현하고자 합니다. 이러한 통합 구조는 디지털 자산 관리의 보안성과 효율성을 동시에 향상시키는 역할을 합니다. 또한 Pharos는 커뮤니티 내 기술 공유와 협력적 문제 해결을 기반으로 성장하는 생태계를 지향합니다.

Web3 혁신의 최전선에서, Pharos는 전통 금융 인프라와 미래의 탈중앙화 경제 간의 격차를 해소하는 것을 목표로 합니다. 또한 블록체인 기술의 혜택을 보다 폭넓게 제공하고, 커뮤니티 주도의 지속적인 발전을 통해 진화하는 생태계를 구축하고자 합니다.

2 설계 근거

블록체인은 단순한 원장이 아니라 컴퓨팅 시스템입니다. Web3 애플리케이션의 복잡성이 증가함에 따라 확장성과 효율성에 대한 요구는 더욱 중요해지고 있습니다. 산업 전반에서는 대규모 채택(mass adoption)의 요구를 충족하기 위해 보다 강력한 병렬 처리 역량으로의 전환이 이루어지고 있습니다. 순차 처리(sequential processing)에서 고도로 병렬화된 아키텍처로의 진화는 블록체인 발전 과정에서 중요한 이정표로 평가됩니다. 또한 레이어 1 네트워크 프로토콜, 실행 엔진, 스케줄링 알고리즘, 저장소 관리 측면에서의 최적화를 통해 전체 처리량을 한 단계 끌어올리는 성과가 지속적으로 이루어지고 있습니다.

기존 및 미래 L1 병렬화 기술에 대한 종합적인 관점을 제공하기 위해, 본 문서에서는 네트워크 병렬화 수준을 병렬성 수준(Degree of Parallelism, DP)으로 정의하고, 각 단계에서 기대되는 성능 향상(p)을 다음과 같이 구분합니다:

DP0: 순차 모델

병렬 실행 없이 선형적으로 처리되는 구조입니다.

DP1: 확장 가능한 합의

확장성, 적응성, 보안성 및 고성능 블록 전파를 갖춘 개선된 합의 메커니즘으로, 즉각적인 파이널리티를 보장합니다.

DP2: DP1 + 병렬 트랜잭션 처리

병렬 추측 실행(parallel speculation), 스케줄링, 멀티스레드 및 멀티 명령 처리 능력을 활용하여 트랜잭션 단위에서 높은 처리량의 병렬 처리를 제공합니다.

DP3: DP2 + 파이프라이닝

다단계 파이프라인 처리 구조를 도입하여 전체 처리량과 자원 활용도를 크게 향상시키며, 다중 수준 파이널리티와 보안을 동시에 보장합니다.

DP4: DP3 + 병렬 머클화 및 가속 상태 접근

입출력 효율성과 상태 비대화(state bloat) 문제를 해결하기 위해 통합된 머클화 및 인덱싱 구조를 적용한 병렬성의 고도 단계입니다. 이를 통해 데이터 접근의 완전한 병렬 처리가 가능하며, 저장 비용을 최소화할 수 있습니다.

DP5: DP4 + 병렬 이기종 연산

노드의 비대칭 연산 능력을 최대한 활용하고, GPU, 신뢰 실행 환경(TEE), 완전 동형 암호(FHE) 가속기, ZK 가속기, 스마트 네트워크 카드 등 이기종 하드웨어의 병렬 처리 능력을 지원함으로써 네트워크 및 애플리케이션 전반의 보안을 강화하고 병렬 처리 역량을 극대화합니다. 또한 체인은 승인된 블록의 해시를 이전 해시로 사용하는 구조를 유지합니다.

시나리오의 규모가 커질수록 더 높은 수준의 병렬성(DP)이 요구됩니다. Web3 애플리케이션의 발전과 함께 일부 플랫폼은 최근 DP2에서 DP3 수준으로 진입하였습니다. 그러나 확장성 문제를 근본적으로 해결하고 대규모 채택을 실현하기 위해서는 DP4로의 전환이 필수적입니다. Pharos는 블록체인 기술을 차세대 수준으로 발전시키고 Web3의 광범위한 확산을 촉진하기 위해, DP4 및 나아가 DP5 수준의 병렬성을 수용하는 핵심 설계 원칙을 기반으로 구축되었습니다.

확장성 (Scalability, DP4): Pharos는 심층 병렬 처리와 초고성능 기술을 통해 인터넷 규모의 사용자 및 트랜잭션을 지원합니다. Pharos는 블록체인의 모든 계층에서 병렬성을 적용하여 성능을 향상시키고 지연을 감소시키며, 복잡한 dApp 환경에서 자원 활용을 최적화합니다:

1. 확장 가능한 합의 프로토콜: 네트워크 전체 자원을 활용하는 고처리량·저지연 BFT 합의 프로토콜입니다.
2. 이중 VM 병렬 실행: 고급 컴파일 기법을 적용한 EVM 및 WASM 기반의 병렬 이중 VM 실행 계층입니다.
3. 전 생애주기 비동기 파이프라이닝: 트랜잭션의 전체 생애주기(실행, 합의, I/O) 및 블록 간 처리 전반에 걸쳐 병렬 및 비동기 처리를 가능하게 합니다.
4. 인증 데이터 구조 기반 고성능 저장소: 높은 처리량, 저지연 I/O, 비용 효율적인 상태 저장을 제공하며, 수십억 계정 규모까지 안전하게 확장 가능합니다.

확장성: Pharos는 모듈형 특수 처리 네트워크(SPN, Special Processing Networks)를 통해 새로운 소프트웨어, 하드웨어 및 지리적 탈중앙화를 유연하게 통합할 수 있으며, 다양한 사용 사례와 신기술을 지원합니다.

또한 Pharos는 네트워크, 자산 및 애플리케이션 보안을 최우선으로 고려하며, 페타바이트급 데이터에 대한 클라이언트 SPV, 비잔틴 장애 허용(Byzantine Fault Tolerance), 감사 가능성(auditability), 분석 도구, 인증 저장소(authenticated storage) 등 핵심 보안 메커니즘을 통해 데이터 무결성과 보안을 보장합니다.

Pharos는 무한한 확장성, 견고한 보안성, 유연한 확장성을 기반으로 탈중앙화 미래를 구축하는 것을 목표로 합니다. 구체적인 기술적 세부사항은 다음 섹션에서 다룹니다.

3 Pharos의 기술 구조

3.1 확장 가능한 네트워크 및 합의

Pharos는 높은 처리량과 낮은 지연을 동시에 제공하는 합의 프로토콜을 채택하고 있습니다. 해당 프로토콜은 광역 네트워크(wide-area network) 환경에서 대규모 노드를 대상으로 설계되어 시스템의 견고성과 합의 보안을 강화합니다. Pharos 합의의 설계 목표는 크게 두 가지로, 응답성(responsiveness)과 최적 자원 활용(optimal resource utilization)입니다. 응답성이란 블록체인 시스템의 처리 속도가 인위적인 타임아웃이나 대기 시간 없이 실제 네트워크 지연에 의해서만 제한되는 특성을 의미합니다. 자원 활용 측면에서는 합의 노드들이 대칭적인 통신 패턴을 유지하며 동시에 동작할 수 있어야 합니다.

그러나 대부분의 기존 블록체인 합의 프로토콜은 이러한 두 가지 목표를 달성하지 못하며, 이는 성능 및 확장성 문제로 이어집니다. 예를 들어 일부 블록체인은 연속된 블록 간에 일정한 시간 간격을 강제하여 블록 생성 속도를 고정시키고 있으며, 이로 인해 네트워크나 연산 최적화와 관계 없이 시스템 처리량에 상한이 존재하게 됩니다. 또한 가장 일반적인 통신 방식은 제안-투표(proposing-voting) 구조로, 단일 제안자가 블록을 생성하고 전파하며, 다른 노드는 비교적 작은 투표 메시지만을 전송합니다. 이로 인해 제안자가 병목 지점으로 작용하게 됩니다. 검증자 수가 증가할수록 제안자의 네트워크 자원 부담이 증가하여 성능 저하가 발생합니다. 반면 검증자 수 증가에 따라 전체 네트워크 대역폭은 증가하므로, 모든 검증자의 자원을 균등하게 활용할 경우 확장 가능한 합의 네트워크를 구현할 수 있습니다.

이러한 문제를 해결하기 위해 Pharos 합의는 두 가지 핵심 기능을 도입합니다. 첫째, 고정된 시간 가정에 의존하지 않으며, 블록 생성이 사전 정의된 타임아웃이 아니라 실제 네트워크 지연에 따라 동적으로 이루어집니다. 둘째, 모든 합의 노드가 동시에 블록 제안을 수행할 수 있도록 하여 단일 제안자 의존 구조를 제거합니다. 이러한 설계는 확장성을 크게 향상시키고 시스템 처리량을 실질적으로 증가시킵니다. 또한 Pharos 합의는 유연한 진행 패러다임을 도입하여, 다양한 워크로드 및 네트워크 환경에 대응할 수 있도록 노드가 보다 유연하게 블록을 제안하고 확정할 수 있도록 합니다. 예를 들어 지리적으로 멀거나 성능이 낮은 노드는 제안 빈도를 낮출 수 있으며, 제안이 억제되는 문제 없이 참여할 수 있습니다. 100개 노드로 구성된 글로벌 테스트 환경에서, 프로토타입 구현은 초당 130,000건 이상의 트랜잭션 처리량을 달성하였습니다.

3.2 병렬 트랜잭션 처리

Pharos 네트워크의 트랜잭션은 높은 성능, 확장성 및 효율성을 기반으로 실행됩니다. 해당 아키텍처는 스케줄러(scheduler)와 실행기(executor)라는 두 가지 핵심 구성 요소로 이루어져 있습니다. 스케줄러는 병렬 스케줄링 및 트랜잭션 실행을 담당하는 핵심 모듈로, 최적화된 알고리즘을 통해 병렬성을 극대화하고 충돌을 최소화합니다. 실행기는 이중 가상머신(EVM 및 WASM) 엔진을 통해 빠르고 유연한 스마트 컨트랙트 실행을 지원합니다. EVM 엔진은 Solidity 기반 컨트랙트와의 호환성을 제공하며, WASM 엔진은 다양한 스마트 컨트랙트 언어에 대해 고성능 실행 환경을 제공합니다.

3.2.1 스케줄러

Pharos 스케줄러는 높은 동시성을 기반으로 거의 최적에 가까운 병렬 트랜잭션 그룹을 구성하여, 무결성과 정확성을 유지하면서 빠른 트랜잭션 실행, 효율적인 검증, 결정론적 파이널리티(deterministic finality), 그리고 충돌 처리를 보장합니다. 병렬 실행 효율성과 사전 정의된 결정론적 파이널리티를 확보하기 위해 Pharos는 다음과 같은 전략을 적용합니다:

- 병렬 힌트 생성: 스마트 컨트랙트의 정적 분석과 추측 실행(speculative execution)을 활용하여 트랜잭션의 읽기-쓰기 집합(read-write set)에 대한 높은 정확도의 힌트를 생성하며, 이를 통해 충돌을 최소화하고 병렬성을 향상시킵니다. 정교한 컴파일 프레임워크와 세밀한 분석 알고리즘을 통해 제어 흐름(control flow) 및 데이터 흐름(data flow)에 대한 상세 정보를 도출하며, 실행 시 부담을 줄이기 위해 가능한 한 정확한 힌트를 제공합니다.
- 트랜잭션 의존성 분석: 생성된 읽기-쓰기 집합을 기반으로 트랜잭션 간 의존성을 분석하고, 유니온-파인드(union-find) 알고리즘을 활용하여 병렬 실행 가능한 최적의 트랜잭션 그룹을 구성합니다. 각 그룹은 실행 워커 수에 맞게 병합되며, 실행 전에 해당 그룹이 접근할 상태 객체를 단일 배치 I/O 작업으로 사전 로딩함으로써 실행 시간을 크게 단축합니다.
- 낙관적 실행 및 파이프라인 파이널리티: 낙관적 실행과 "파이프라인 파이널리티" 알고리즘을 활용하여 실행 결과를 빠르게 수렴시키고 최종 상태를 결정합니다.

또한 Pharos는 병렬성 향상을 지속적으로 추진합니다. 검증자는 추측 실행과 같은 작업을 수행하면서 멀티코어 CPU 및 I/O 자원을 최대한 활용할 수 있습니다. 실행기 또한 스케줄링 작업을 일부 담당하여 전체 자원 활용을 최적화하고 특정 자원에 대한 집중을 방지합니다. 전역 카운터와 같은 읽기-쓰기 충돌이 발생할 수 있는 데이터는 병렬 처리 및 저장 구조 최적화를 통해 성능을 개선합니다. 더불어 세밀한 알고리즘 제어를 통해 충돌을 신속히 탐지하고 최소 비용으로 재실행을 수행함으로써, 최적의 병렬 성능을 달성합니다.

3.2.2 실행기

Pharos 실행기는 간결한 인터페이스와 내장형 호스트 구조를 기반으로 구현되었으며, 최신 컴파일 최적화 기법과 블록체인 특화 기능을 결합하여 높은 성능과 낮은 수수료를 제공합니다. 일반적인 컨트랙트 기준으로 기존 EVM 구현 대비 최대 10배 수준의 성능 향상이 가능합니다. Pharos 실행기는 EVM과 WASM을 모두 지원하며, 두 환경 간 상호운용성을 제공합니다. 이를 통해 개발자는 EVM 생태계의 이점을 유지하면서 Rust, C++, Go, Java와 같은 고급 언어

기반 라이브러리를 활용한 고성능 개발이 가능합니다. 실행 계층의 주요 처리 과정은 다음과 같습니다.

3.2.3 디알렉트 변환

디알렉트 변환은 EVM과 WASM 간의 opcode 및 시스템 함수 호출을 매핑하여, 서로 다른 플랫폼 간 컨트랙트 변환을 원활하게 수행할 수 있도록 합니다. Pharos IR을 활용한 중간 표현 계층은 확장성과 다른 가상머신과의 호환성을 제공합니다.

3.2.4 Pharos IR

Pharos IR은 LLVM 및 MLIR 프레임워크를 기반으로 설계된 레지스터 기반 중간 표현(intermediate representation)입니다. 해당 IR은 다양한 최적화를 수행하고, EVM opcode와 WASM 바이트코드를 통합된 디알렉트 표현으로 변환합니다. 이러한 구조는 확장성을 높이고, 일관성 및 정확성을 개선하며, 개발 효율성을 향상시킵니다.

3.2.5 심볼릭 검증기

Pharos IR은 검증 친화적으로 설계되어 있으며, 강력한 타입 시스템과 높은 수준의 정적 결정성을 제공합니다. Pharos Symbolic Verifier는 Pharos IR 기반의 심볼릭 실행 엔진으로, 추상 해석(abstract interpretation)과 정제 실행(refinement execution) 기법을 활용하여 실행 경로와 프로그램 상태를 효율적으로 탐색합니다. 해당 검증기는 다음과 같은 목표를 가집니다:

- 컴파일 무결성: EVM 디알렉트와 WASM 코드 간, 그리고 서로 다른 최적화 단계 간의 로직 일관성을 보장합니다.
- 추측 실행 무결성: 추측 실행의 복잡성으로 인해 발생할 수 있는 보안 문제를 완화하기 위해, 각 추측 롤백 과정에 대해 형식적 검증을 수행하여 무결성을 보장합니다.
- 사용자 정의 명세 기반 검증: Pharos Spec은 도메인 특화 명세 언어로, 타입 안전성, 메모리 안전성, 제어 흐름, 데이터 흐름 무결성 등 사용자 정의 안전 속성을 검증할 수 있도록 합니다. Pharos Spec과 Symbolic Verifier의 결합은 스마트 컨트랙트 보안을 위한 추가적인 보호 계층을 제공합니다.

3.2.6 최적화 패스

디알렉트에 적용되는 최적화 패스에는 죽은 코드 제거(dead code elimination), 상수 폴딩(constant folding), 함수 인라이닝(function inlining), opcode 융합(opcode fusion), 레지스터 승격(register promotion) 등이 포함됩니다. 이러한 최적화는 극대화된 성능을 위한 최적의 머신 코드를 생성합니다. 또한 가상머신은 고빈도 I/O 연산을 최대한 메모리 연산으로 치환하고, 메모리 연산을 레지스터 연산으로 최적화합니다. 예를 들어 일부 전역 저장소 또는 컨트랙트 비용 카운터와 같은 요소에 대해 이러한 최적화가 적용됩니다. 이를 통해 명령어 소비를 줄이는 동시에 읽기-쓰기 충돌을 감소시키고 병렬성을 향상시켜 전체 비용을 절감합니다.

3.3 파이프라이닝

Pharos는 처리량, 성능 및 자원 활용도를 향상시키기 위해 현대 컴퓨터의 슈퍼스칼라 프로세서 파이프라인과 유사한 블록 단위 다단계 파이프라인 메커니즘을 적용합니다. 이러한 파이프라인 설계는 전체 블록 처리 생애주기를 최적화함으로써 트랜잭션 수준 병렬 실행 대비 유의미한 속도 향상을 달성할 수 있도록 합니다. 그림 3.3은 Pharos에서 해당 파이프라인이 작동하는 방식을 시각적으로 설명합니다.

Pharos 파이프라인은 블록 생애주기를 여섯 개의 세분화된 단계로 구분하여 효율성을 극대화합니다: 합의 순서 결정(consensus ordering), 병렬 실행(parallel execution), 데이터베이스 읽기(database reading), 머클화(Merklization), 블록 생성(block building), 데이터베이스 플러시(database flushing). 이러한 구조는 서로 다른 프로세서가 여러 블록을 동시에 처리할 수 있도록 하여 최대 수준의 병렬성을 확보합니다. 또한 파이프라인 스케줄링은 I/O 집약 단계와 CPU 집약 단계를 분리하고, 가장 최근 블록의 실행 결과를 우선적으로 처리하도록 설계되어 있습니다.

Pharos는 파이프라인 내에서 다음과 같은 세 가지 유형의 파이널리티를 도입합니다:

1. 순서 파이널리티: 모든 트랜잭션이 블록 내에서 영구적으로 순서화되어 일관된 실행 순서와 트랜잭션 해시를 보장합니다.
2. 트랜잭션 파이널리티: 각 트랜잭션은 결정론적으로 실행되며, 사용자에게 실행 결과 및 계정 상태 변경을 제공합니다.
3. 블록 파이널리티: 블록은 생성 및 저장 과정을 통해 최종 확정되며, 사용자 또는 API가 머클 루트 생성 및 조회 지연 없이 즉시 전체 블록 정보를 확인할 수 있도록 합니다.

Pharos 파이프라인은 설계 구조를 통해 무결성과 보안을 보장합니다. 모든 검증자가 동시에 장애를 겪더라도, 이미 파이널리티가 확정된 단계에서는 데이터 손실이 발생하지 않도록 보장합니다.

검증자는 장애 이후 최신 상태로 재동기화 및 복구할 수 있으며, 새로운 블록의 저장은 이전 블록이 완전히 저장된 이후에만 수행되도록 보장됩니다. 또한 모든 단계에서 더티 리드(dirty read) 및 팬텀 리드(phantom read)를 엄격히 관리하여 데이터 일관성을 유지합니다.

이와 같은 견고한 파이프라이닝 구조를 통해 Pharos는 네트워크의 성능과 확장성을 향상시킬 뿐만 아니라 높은 수준의 보안성과 데이터 무결성을 확보하여, 복잡한 Web3 애플리케이션의 원활하고 효율적인 실행을 지원합니다.

3.4 병렬 머클화 및 가속 상태 접근

대규모 채택을 위한 병렬 처리 역량이 확대됨에 따라, 상태 비대화(state bloat)로 인해 입출력 성능(I/O performance)과 저장 확장성이 주요 과제로 부각됩니다.

기존 블록체인 저장 구조는 일반적으로 상태 데이터를 인덱싱하기 위해 2계층 아키텍처를 사용합니다: LSM 트리 기반의 범용 키-값 저장 시스템이 백엔드 저장소로 활용되며, 그 위에 인증 데이터 구조(ADS, Authenticated Data Structure)가 구성됩니다. ADS는 일반적으로 머클 트리이며, 각 노드는 키-값 저장 시스템에 개별적으로 저장됩니다. 예를 들어, 이더리움은 상태 데이터 저장 엔진으로 MPT+LevelDB를 사용합니다. 그러나 이러한 전통적 구조는 다음과 같은 한계를 가집니다:

1. 긴 I/O 경로: 2계층 구조에서는 I/O 경로가 길어지며, 단일 키-값 접근 시 여러 트리 노드를 조회해야 하고, 각 조회는 다층 LSM 트리 기반 데이터베이스에서 다수의 디스크 I/O를 발생시킵니다.
2. 쓰기 및 공간 증폭: 머클 트리의 인코딩 방식은 쓰기 증폭과 공간 증폭을 유발합니다. 단일 객체를 업데이트할 경우 트리 상의 경로에 있는 모든 노드의 해시 값을 갱신해야 하며, 다중 버전 지원을 위해 일부 필드만 변경되더라도 전체 노드를 재기록하여 저장해야 합니다.
3. 해시 기반 주소 지정: 이는 상당한 I/O 대역폭 소모를 초래합니다. LSM 기반 키-값 저장 시스템은 백그라운드에서 컴팩션(compaction)을 수행하여 낮은 레벨의 SST 파일을 상위 레벨로 병합하는데, 머클 해싱의 랜덤성으로 인해 키 분포가 무작위로 나타나며 다수의 SST 파일 간 중복을 유발하여 I/O 대역폭을 과도하게 소모합니다.

이러한 상태 비대화 문제로 인한 한계를 해결하기 위해 Pharos Store가 개발되었습니다. Pharos Store는 블록체인 네이티브 검증 가능 저장 솔루션으로, 고성능 읽기/쓰기와 효율적인 머클화를 동시에 제공합니다. 이를 통해 최대 15.8배의 처리량 향상과 80.3%의 저장 비용 절감을 달성할 수 있습니다.

Pharos Store의 주요 혁신은 다음과 같습니다:

- ADS 푸시다운: ADS를 저장 엔진 내부에 통합하여, ADS와 백엔드 저장소 간 경계를 제거하고 세밀한 I/O 최적화를 가능하게 합니다.
- 델타 인코딩: 트리 노드를 델타 인코딩 방식으로 저장하여 변경된 데이터만 기록함으로써 I/O 및 공간 증폭을 감소시킵니다.
- 버전 기반 주소 지정: 해시 기반 주소 체계를 버전 기반 주소 체계로 대체하여 데이터 버전 순서를 활용함으로써 컴팩션 오버헤드를 제거합니다.

3.4.1 ADS 내장 단일 계층 구조

기존 2계층 구조의 한계를 극복하기 위해 Pharos Store는 ADS를 저장 엔진 내부로 통합하는 새로운 아키텍처를 채택하며, 저장 인덱스와 사용자 메타데이터를 분리하여 I/O 증폭을 추가적으로 감소시킵니다.

Pharos Store는 다음 세 가지 주요 구성 요소로 이루어집니다:

1. DMM-Tree (Delta-encoded Multi-version Merkle Tree): 사전(dictionary) 트리, B-트리 등 다양한 트리 구조를 지원하는 내장형 다중 버전 머클 트

리이며, 리프 노드에는 사용자 데이터의 파일 번호 및 오프셋이 기록됩니다.

2. LSVPS (Log-Structured Versioned Page Store): DMM-Tree를 위한 메모리와 보조 저장소 간 페이지 인덱스 추상화를 제공합니다.
3. VDLs (Versioned Data Logging Stream): 사용자 메타데이터를 append-only 방식으로 저장합니다.

또한 Pharos Store는 API 계층과 append-only 영속성 계층을 제공하며, SSD/NVMe 및 클라우드 스토리지 등 최신 저장 장치를 유연하게 지원하도록 설계되어 있습니다. 해당 구조는 EXT4, NAS, HDFS, CEPH 등 백엔드 저장 시스템에 대한 추상화 계층으로 작동하며, append 쓰기 및 효율적인 랜덤 접근 읽기를 지원합니다.

3.4.2 델타 인코딩

Pharos Store의 LSVPS 계층에서는 각 페이지가 점진적으로 업데이트된 데이터를 DeltaPage로 집계하여 저장합니다. DeltaPage의 크기를 제어하기 위해 임계값이 설정되며, 일정 수준의 업데이트가 누적되면 해당 DeltaPage는 고정(freeze)되고 새로운 DeltaPage가 생성되어 업데이트를 이어받습니다. 페이지 로딩 오버헤드를 최소화하기 위해, 일정 횟수 이상의 업데이트 이후에는 전체 상태를 포함하는 BasePage가 생성됩니다.

3.4.3 버전 기반 주소 지정

블록체인 데이터의 단조 증가(monotonic)하는 버전 특성을 활용하여, Pharos Store는 해시 기반 인덱싱을 버전 기반 주소 체계로 대체함으로써 디스크 대역폭 효율성을 향상시킵니다.

- API 지원: (V, k) 형태의 인덱스 키를 사용하여 특정 버전의 모든 이력 데이터를 조회할 수 있는 API를 제공합니다.
- 버전 관리: 각 트리 노드와 페이지는 버전 번호를 유지하며, 재정렬이나 컴팩션 없이 효율적인 데이터 접근이 가능합니다.
- 정렬된 저장: 페이지를 버전 순으로 저장하여 범위 기반 연산 및 이력 데이터의 배치 삭제를 효율적으로 수행할 수 있습니다.

Pharos Store는 블록체인 네이티브 저장 구조의 중요한 발전을 의미하며, 암호학적 위변조 방지 기능과 함께 뛰어난 성능 및 자원 효율성을 제공합니다. 이러한 혁신은 수십억 사용자 규모의 확장성과 증가하는 성능 요구를 충족할 수 있도록 하며, 다른 블록체인에도 적용 가능한 범용 솔루션으로 기능합니다.

3.5 병렬 이기종 연산

견고하고 확장 가능하며 탈중앙화된 블록체인 네트워크를 구축하기 위해 Pharos는 검증자 노드(validator nodes), 풀 노드(full nodes), 릴레이어 노드(relayer nodes)의 세 가지 핵심 노드 유형을 통합한 정교한 네트워크 토폴로지와 합의 메커니즘을 채택하고 있습니다. 이러한 구조는 네트워크의 무결성, 성능 및 보안을 동시에 보장합니다.

검증자 노드는 Pharos 합의 메커니즘의 핵심 구성 요소로, 비잔틴 장애 허용(BFT, Byzantine Fault Tolerance) 기반 지분증명(Proof of Stake) 프로토콜을 사용합니다. 수백 개의 검증자가 협력하여 안전하고 신뢰할 수 있는 네트워크를 유지하며, 사용자 트랜잭션을 신속하게 처리합니다. 또한 기존의 트랜잭션 수수료 및 스테이킹 보상 외에도, 검증자는 리스 테이킹(Restaking)을 통해 추가 수익을 창출할 수 있습니다. 검증자는 자신의 토큰 증서를 SPN 또는 dApp에 할당함으로써 확장성, 보안성 및 유동성을 강화하고, 이에 따른 추가 보상을 획득할 수 있습니다.

풀 노드는 전체 블록 및 상태 트랜잭션을 저장하며, 네트워크의 핵심 지원 기능을 수행합니다. 합의 과정에는 직접 참여하지 않지만, Fast State Sync를 통해 다른 노드가 최신 상태로 빠르게 동기화할 수 있도록 지원하며, Parallel Hint를 제공하여 검증자의 병렬 실행 효율성을 향상시킵니다.

릴레이어 노드는 경량 클라이언트로서 최신 상태를 저장하고 최근 트랜잭션을 처리합니다. 낮은 수준의 하드웨어 요구사항을 가지므로 DePIN 네트워크 참여에 적합합니다. 릴레이어 노드는 트랜잭션 시뮬레이션 및 전달을 포함한 핵심 노드 서비스를 제공하며, SPN 네트워크 내에서 메시지를 효율적으로 전달함으로써 추가 보상을 획득할 수 있습니다.

이와 같이 고도화된 노드 아키텍처와 합의 프로토콜을 결합함으로써 Pharos는 다양한 애플리케이션을 지원하면서도 다양한 환경에서 높은 성능을 유지할 수 있는 확장 가능하고 효율적인 블록체인 네트워크를 구현합니다. 더 나아가 Pharos는 경량 특수 처리 네트워크(SPN, Special Processing Networks)의 네이티브 배포를 지원합니다. 사용자는 Pharos 네트워크 내 검증자를 선택하여 전용 SPN을 생성할 수 있으며, 해당 SPN은 AIoT 프라이빗 네트워크 또는 다자간 프라이버시 강화 연산(MPT) 전용 네트워크와 같이 완전히 다른 프로토콜을 채택할 수 있습니다. 또한 경량 SPN은 트랜잭션 기밀성 및 MEV 대응을 위한 TEE 기반 네트워크 또는 특수 하드웨어를 활용하는 네트워크를 지원합니다.

3.5.1 이기종 컴퓨팅 네트워크

SPN은 특화된 데이터 및 연산 자원에 대한 요구를 충족시키고, 이기종 연산 자원의 스케줄링을 최적화하는 역할을 수행합니다. 이를 통해 다양한 블록체인 생태계 및 컴퓨팅 환경 간의 원활한 상호작용을 지원하며, 자원 활용을 최적화하고 성능을 향상시키며 비용을 절감할 수 있습니다. 또한 SPN은 동적 확장 기능을 지원하여 증가하는 수요와 복잡한 애플리케이션을 안정적으로 처리할 수 있도록 합니다.

리스테이킹(Restaking)은 검증자가 스테이킹된 자산을 SPN 또는 dApp에 동적으로 할당할 수 있도록 하여 보안성과 유동성을 강화하고 유연한 자원 분배를 가능하게 합니다. 검증자는 SPN을 지원하기 위해 자산을 리스테이킹함으로써 네트워크의 보안성과 신뢰성을 향상시킵니다. 또한 리스테이킹은 스테이킹 자산의 재분배를 통해 네트워크 유동성과 경제적 인센티브를 개선하며, 네트워크 수요에 따라 자원을 유연하게 할당하여 최적의 성능을 달성할 수 있도록 합니다.

보안 측면에서 SPN은 안전한 다자간 연산(Secure Multi-Party Computation), TEE, 동형 암호(homomorphic encryption) 등 프라이버시 강화 기술을 적용하여 AI 처리 과정에서 데이터를 보호합니다. SPN과 검증자 간의 효율적이고 안전한 자원 스케줄링은 리스테이킹을 통해 구현되며, 이는 전체 네트워크 보안을 강화하는 역할을 합니다. 또한 고급 검증 기법을 통해 연산 무결성을 보장하고 악의적 행위를 방지합니다.

Pharos의 SPN 구조는 자원 할당 최적화, 상호운용성, 보안성을 결합하여 디지털 혁신을 위한 견고한 인프라를 제공합니다. SPN과 리스테이킹을 기반으로 Pharos는 빠르게 변화하는 디지털 환경에 적응 가능한 유연하고 안전하며 효율적인 블록체인 생태계를 구축하는 것을 목표로 합니다.

4 결론

Pharos는 블록체인 기술의 중요한 진전을 의미하며, Web3의 미래를 위한 확장 가능하고 효율적이며 안전한 솔루션을 제공합니다. 모든 계층에서 병렬성을 적극적으로 활용하고 혁신적인 저장 및 실행 기술을 통합함으로써, Pharos는 탁월한 성능과 높은 신뢰성을 보장합니다. 또한 포용성, 보안성, 확장성을 핵심 가치로 삼아 차세대 탈중앙화 애플리케이션과 디지털 혁신을 선도할 준비가 되어 있습니다. 향후 로드맵에는 블록체인 기술의 가능성을 지속적으로 확장할 다양한 주요 이정표와 발전 계획이 포함되어 있습니다. 이를 통해 전 세계 사용자에게 권한을 부여하고 혁신을 촉진하는 탈중앙화 미래를 함께 구축해 나갈 수 있을 것입니다.

참조문헌

- [1] The global e-wallets landscape: 2023 and beyond, <https://www.helloza.com/blog/the-global-e-wallets-landscape-2023-and-beyond>.
- [2] S. Liu, W. Xu, C. Shan, et al., “Flexible advancement in asynchronous bft consensus,” in Proceedings of the 29th Symposium on Operating Systems Principles, 2023, pp. 264–280.
- [3] R. Gelashvili, A. Spiegelman, Z. Xiang, et al., “Block-stm: Scaling blockchain execution by turning ordering curse to a performance blessing,” in Proceedings of the 28th ACM SIGPLAN Annual Symposium on Principles and Practice of Parallel Programming, 2023, pp. 232–244.
- [4] Y. Lu, C. Liu, M. Zhao, et al., “Fsc: A fast smart contract transaction execution approach via read-write static analysis,” Authorea Preprints, 2023.
- [5] C. Gorenflo, L. Golab, and S. Keshav, “Xox fabric: A hybrid approach to blockchain transaction execution,” in 2020 IEEE International Conference on Blockchain and Cryptocurrency (ICBC), IEEE, 2020, pp. 1–9.
- [6] Road to high performance: Parallel evm for bnb chain, <https://www.bnbchain.org/en/blog/road-to-high-performance-parallel-evm-for-bnb-chain>.
- [7] Y. Shi, K. Casey, M. A. Ertl, and D. Gregg, “Virtual machine showdown:

- Stack versus registers,” ACM Transactions on Architecture and Code Optimization (TACO), vol. 4, no. 4, pp. 1–36, 2008.
- [8] Dialect conversion, <https://mlir.llvm.org/docs/DialectConversion/>.
- [9] C. Lattner, M. Amini, U. Bondhugula, et al., “Mlir: Scaling compiler infrastructure for domain specific computation,” in 2021 IEEE/ACM International Symposium on Code Generation and Optimization (CGO), IEEE, 2021, pp. 2–14.
- [10] Y. Chen, Z. Guo, R. Li, et al., “Forerunner: Constraint-based speculative transaction execution for ethereum,” in Proceedings of the ACM SIGOPS 28th Symposium on Operating Systems Principles, 2021, pp. 570–587.
- [11] J. Cocke, “Global common subexpression elimination,” in Proceedings of a symposium on Compiler optimization, 1970, pp. 20–24.
- [12] S. Muchnick, Advanced compiler design implementation. Morgan kaufmann, 1997.
- [13] A. Sastry and R. D. Ju, “A new algorithm for scalar register promotion based on ssa form,” in Proceedings of the ACM SIGPLAN 1998 Conference on Programming Language Design and Implementation, 1998, pp. 15–25.
- [14] X. Qi, Z. Chen, H. Zhuo, et al., “Schain: Scalable concurrency over flexible permissioned blockchain,” in 2023 IEEE 39th International Conference on Data Engineering (ICDE), IEEE, 2023, pp. 1901–1913.
- [15] Merkle patricia trie, <https://ethereum.org/en/developers/docs/data-structures-and-encoding/patricia-merkle-trie/>.
- [16] S. Tian, Z. Lu, H. Zhuo, et al., “Letus: A log-structured efficient trusted universal blockchain storage,” in Companion of the 2024 International Conference on Management of Data, 2024, pp. 161–174.
- [17] E. Team, “Eigenlayer: The restaking collective,” URL: <https://docs.eigenlayer.xyz/overview/whitepaper>, 2024.