

EDGE STACK: 고빈도 온체인 파생상품을 위한 앱 특화 실행 레이어

EDGEX DAO INC

초록

모놀리식 블록체인의 순차적 실행 모델은 분산형 파생상품에 근본적인 지연 하한선을 부과하여, 자산별 트레이딩 로직이 글로벌 상태 접근을 위해 경쟁하도록 강제한다. 본 논문은 고빈도 온체인 파생상품 거래를 위해 설계된 앱 특화 실행 레이어인 EDGE Stack을 소개한다. 범용 롤업과 달리, EDGE Stack은 시장 샵딩된 상태 접근을 활용하여 충돌하지 않는 오더북을 동시에 실행하는 결정론적 병렬 트랜잭션 실행(PTE) 엔진을 구현한다. 또한 EDGE Stack은 리소스 격리를 위한 모듈식 멀티-VM 아키텍처와, 매칭 지연을 비동기 정산에서 분리하는 이중 계층 트랜잭션 우선순위 메커니즘인 FlashLane을 통합한다. 이 병렬화된 상태의 검증을 보장하기 위해, 시스템은 병렬 머클화(Parallel Merklization)를 채택하여 상태 트리 해시를 동시에 계산하고 결정론적 글로벌 상태 루트로 수렴시킨다. 결과적인 아키텍처는 중앙화된 엔진의 선형 확장성 및 1초 미만의 소프트 완결성과, 분산 네트워크의 검증 가능한 암호학적 보증을 결합하며, 현재 이더리움의 보안 보증을 활용하고 있다.

키워드: 결정론적 병렬 실행 · 병렬 머클화 · 모듈식 멀티-VM · 확장 접근 목록 · 상태 격리

1. 서론

1.1 분산형 파생상품의 인프라 격차

탈중앙화 금융(DeFi)은 시장 접근성을 근본적으로 재편했지만, 그 중 가장 복잡한 부문인 무기한 선물 및 파생상품은 여전히 기반 인프라에 의해 심각하게 제약받고 있다. 자동화 시장 조성자(AMM)가 현물 거래를 혁신했지만, 성숙한 파생상품 시장이 요구하는 레버리지 기반 고빈도 거래 역량에는 본질적으로 자본 비효율적이다. 기관급 파생상품은 정밀한 가격 발견과 깊은 유동성을 위해 중앙 지정가 주문서(CLOB)를 필요로 한다.

그러나 현재의 블록체인 인프라에 고성능 CLOB을 배포하는 것은 구조적 모순을 야기한다. 모놀리식 블록체인은 단일 글로벌 상태 머신 위에서 순차적 실행 모델을 사용한다. 이 환경에서 고빈도 트레이딩 로직은 다른 모든 네트워크 활동과 함께 희소한 블록 공간 및 실행 자원을 두고 경쟁해야 한다. 이른바 "노이지 이웃 (noisy neighbor)" 문제는 예측 불가능한 지연 하한선과 처리량 상한선을 부과하여, 범용 체인이 전문 트레이딩 기업이 요구하는 밀리초 이하의 속도와 결정론적 신뢰성을 지원하는 것을 불가능하게 만든다.

업계는 심각한 병목 현상에 직면해 있다: 분산형 파생상품에 대한 수요가 범용 실행 환경의 역량을 크게 앞지르고 있다.

1.2 EDGE Stack: 앱 특화 실행 레이어

EDGE Stack은 이 인프라 격차에 대한 결정적 해결책으로 설계되었다. 단 하나의 사명을 위해 처음부터 설계된 특수 목적의 앱 특화 실행 레이어로서, 분산 네트워크의 암호학적 검증 가능성을 유지하면서 중앙화된 거래소의 성능을 제공하는 것이 그 목표다.

고성능 실행과 기반 레이어 정산을 근본적으로 분리함으로써, **EDGE Stack**은 범용 롤업의 제약을 넘어선다. 이는 트레이딩 활동이 무관한 네트워크 혼잡에 의해 방해받지 않도록, 금융 처리량에 독점적으로 최적화된 독자적 환경을 제공한다. 궁극적으로 **EDGE Stack**은 차세대 온체인 무기한 및 파생상품 시장의 토대가 되는 목적 지향적 고성능 인프라를 제공한다.

1.3 핵심 아키텍처 기둥

고빈도 성능과 분산 신뢰 간의 충돌을 해결하기 위해, **EDGE Stack**은 트레이딩 생애주기에 최적화된 세 가지 독자적 기술 기둥을 기반으로 한 새로운 아키텍처를 도입한다.

- **모듈식 멀티-VM 아키텍처: 리소스 격리 달성.** 모든 트랜잭션이 단일 런타임을 공유하는 모놀리식 시스템과 달리, **EDGE Stack**은 고성능 트레이딩 로직을 범용 스마트 컨트랙트 실행으로부터 물리적으로 격리하는 이분화된 환경을 구현한다. 이 아키텍처적 방화벽은 핵심 매칭 엔진에 전용 컴퓨팅 대역폭을 보장하여, "노이지 이웃"으로 인한 자원 경쟁을 제거하고 광범위한 네트워크 부하에 관계없이 일관된 성능을 보증한다. (3절에서 상세 설명)
- **결정론적 PTE 엔진: 선형 확장성 확보.** 순차 처리의 처리량 한계를 넘어, **EDGE Stack**은 결정론적 병렬화 엔진을 채택한다. 서로 다른 파생상품 시장 간에 본질적으로 존재하는 상태 격리를 활용하여, 충돌하지 않는 오더북을 동시에 실행한다. 결과적으로 시스템 처리량은 사용 가능한 하드웨어 자원에 비례하여 선형적으로 확장되며, 낙관적 병렬 모델에서 흔히 나타나는 비결정성 없이 초고속을 달성한다. (4절에서 상세 설명)
- **FlashLane: 기관급 지연 시간 제공.** 표준 블록체인 맴폴의 지연 한계를 극복하기 위해, **EDGE Stack**은 프로토콜 네이티브 우선순위 메커니즘인 **FlashLane**을 도입한다. 트레이딩 지시를 비동기 작업으로부터 능동적으로 구별함으로써, **FlashLane**은 트레이더가 즉각적이고 암호학적으로 보증된 소프트 확인을 받을 수 있도록 한다. 이는 거래 실행 속도를 기반 레이어의 느린 완결성으로부터 효과적으로 분리한다. (5절에서 상세 설명)

1.4 상태 수렴 및 보안 기반

초고속 실행 처리량을 달성하더라도 기반 상태 커밋이 이를 따라가지 못하면 무의미하다. 전통적인 시스템에서 암호학적 상태 해시(머클화)의 직렬 재계산 과정은 고부하 시 주요 병목이 되어, 기관 트레이딩에서 허용될 수 없는 예측 불가능한 지연 지터를 유발한다.

이를 해결하기 위해 **EDGE Stack**은 실행 엔진과 동시에 작동하도록 설계된 고급 상태 관리 레이어를 통합한다. 시스템은 병렬 머클화를 활용하여 격리된 시장 샷드 간의 상태 변경을 동시에 계산한다. 이 메커니즘은 암호학적 일관성이 매칭 엔진 속도와 정확히 일치하도록 보장하여, 최고 수준의 성능을 저해하지 않으면서 완전한 온체인 검증 가능성을 유지한 채 상태 커밋에 예측 가능하고 결정론적인 지연 시간을 보증한다. (6절에서 상세 설명)

1.5 결론

결론적으로 **EDGE Stack**은 고빈도 성능과 분산 신뢰 간의 근본적 충돌을 해소한다. 앱 특화 실행이라는 새로운 패러다임을 확립함으로써, 이 아키텍처는 파생상품의 즉각적인 확장성 병목을 극복할 뿐 아니라, 글로벌 온체인 시장 인프라를 위한 탄력적이고 미래 지향적인 토대를 구축한다.

2. 아키텍처 개요

2.1 설계 철학: 애플리케이션 특화 실행 패러다임

EDGE Stack의 아키텍처 기반은 모듈식 블록체인 테제, 즉 고빈도 금융은 범용 합의 레이어와 근본적으로 구별되는 특화된 실행 레이어를 필요로 한다는 원칙에 기반한다. 모놀리식 체인은 고위험 파생상품 시장이 무관한 트래픽과 공유 블록 공간 자원을 두고 경쟁하도록 강제하여, 불가피한 지연 급등과 MEV 누출을 초래한다. 이와 대조적으로 EDGE Stack은 금융 처리량에 독점적으로 최적화된 독자적 실행 환경으로 설계되었다.

우리의 설계 철학은 범용적 최적화보다 풀스택 특화를 우선시한다. 시퀀서의 사전 확인 로직부터 특정 가상 머신 오피코드에 이르기까지 전체 트랜잭션 생애주기를 커스터마이징함으로써, 범용 상태 머신에 내재된 컴퓨팅 오버헤드를 제거한다. 이 패러다임에서 트레이딩 기능은 단순한 스마트 컨트랙트를 넘어 1급 프로토콜 기본 요소(first-class protocol primitives)로 진화한다. 실행과 정산의 이분리는 기반 분산 네트워크의 신뢰 최소화 보안 속성을 계승하면서 결정론적, 초저지연 성능을 가능하게 한다.

2.2 스택 계층 구조

EDGE Stack은 사용자 인입부터 최종 정산에 이르는 명확한 3계층 아키텍처를 통해 엄격한 관심사 분리를 위한 계층형 토폴로지로 운영된다.

그림 1: EDGE Stack 계층 아키텍처

2.2.1 I. 인입 및 시퀀싱 레이어

스택의 최상단에는 FlashLane 시퀀서가 위치하여 트랜잭션 인입의 주요 게이트웨이 역할을 한다. 그 기능은 단순한 맴폴 집계를 넘어서, 운영 긴급도에 따라 트래픽을 능동적으로 형성하고 정렬한다. 혼란스러운 네트워크 입력을 구조화된 배치 데이터로 조직화함으로써, 시퀀서는 실행 전에 클라이언트에게 즉각적인 결정론적 소프트 확인을 제공하는 고성능 버퍼 역할을 한다.

2.2.2 II. 실행 레이어 (핵심부)

스택의 핵심에는 결정론적 상태 전이를 담당하는 독자적 엔진인 실행 레이어가 있다. 이 핵심부는 전통적인 직렬 실행 모델의 처리량 한계를 초월하기 위해 PTE 엔진으로 구동되는 모듈식 멀티-VM 환경을 통합한다. 시퀀서로부터 정렬된 트랜잭션 배치를 받으면, 실행 레이어는 워크로드를 격리된 목적별 런타임 환경으로 동적 라우팅한다. 지연에 민감한 트레이딩 로직은 특화된 edgeVM으로, 표준 스마트 컨트랙트 상호작용은 edgeEVM으로 처리된다. 궁극적으로 이 병렬 처리의 출력은 단순히 분리된 상태 변경의 집합이 아니라, 정산을 위해 준비된 단일 검증 가능한 머클 루트로 암호학적으로 수렴된 일련의 병렬 전이다.

2.2.3 III. 모듈식 정산 및 데이터 가용성 레이어

기반 레이어는 암호경제적 보안, 불변성, 데이터 가용성(DA)을 처리한다. 결정적으로 EDGE Stack은 이 레이어를 고정된 의존성이 아닌 플러그 가능한 모듈식 인터페이스로 취급한다. 그 주요 역할은 완결성의 앵커로 기능하는 것으로, 압축된 트랜잭션 데이터와 증명된 상태 루트가 신뢰 불필요한 검증을 위해 게시된다. 초기 구현은 강력한 보안 보증을 위해 이더리움 생태계 같은 검증된 네트워크를 활용하지만, 아키텍처는 특정 기반 합의 메커니즘에 구애받지 않는다. 이 모듈식 설계는 전략적 유연성을 유지하여, EDGE Stack이 다양한 DA 솔루션과 통합하거나 미래에 독자적 정산 네트워크로 발전할 수 있도록 한다.

2.3 핵심 차별점: 검증 가능하고 결정론적인 병렬성

EDGE Stack을 구별하는 정의적 특성은 고처리량 멀티 모듈 병렬성과 엄격한 암호학적 검증 가능성을 조화시키는 능력이다.

기존의 병렬 아키텍처에서 속도를 달성하는 것은 종종 결정론을 희생시켜, 신뢰 없는 감사를 불가능하게 만드는 경쟁 조건이나 하드웨어 특정 타이밍 변동을 초래한다. EDGE Stack은 이 타협을 제거한다. PTE 엔진과 병렬 머클화 알고리즘을 활용함으로써, 아키텍처는 정렬된 입력 배치가 주어졌을 때 기반 하드웨어 인프라와 관계없이 병렬 실행이 항상 정확히 동일한 상태 루트로 수렴할 것을 보증한다.

결정론에 대한 이 엄격한 준수는 FlashLane의 즉각적 사전 실행 메커니즘의 극단적 최적화 압박 하에서도 유지된다. 이 역량은 Web3의 "신뢰하지 말고 검증하라" 정신의 기반으로서, 중앙화된 엔진의 저지연 성능과 퍼블릭 블록체인의 신뢰 불필요한 감사 가능성 사이의 간극을 메운다.

3. 기둥 I: 모듈식 멀티-VM 아키텍처

3.1 모놀리식 실행의 한계

최대의 다목적성을 위해 설계된 범용 블록체인은 특화된 금융 애플리케이션에게 부담이 된다. 모놀리식 EVM 아키텍처에서는 복잡한 파생상품 가격 산정부터 간단한 거버넌스 투표에 이르기까지 이질적인 작업들이 균일한 미터링 로직 하에 동일한 자원을 두고 경쟁한다. 이 본질적인 자원 경쟁은 계산 집약적인 트레이딩 시스템의 성능에 구조적 상한선을 부과한다.

이 병목을 극복하기 위해, EDGE Stack은 관심사 분리 원칙에 기반한 모듈식 멀티-VM 아키텍처를 채택한다. 고빈도 트레이딩 로직을 범용 스마트 컨트랙트 실행으로부터 아키텍처적으로 분리함으로써, 시스템은 매끄러운 상호운용성을 유지하면서 각각의 특정 워크로드에 맞게 고유한 런타임 환경을 최적화한다.

3.2 핵심 구성 요소: 병렬 런타임 환경

EDGE Stack 실행 레이어는 특화된 확장 메커니즘과 함께 두 가지 주요 환경을 병렬로 운영하는 이분화된 런타임 아키텍처를 구현한다. 이 설계는 고도로 특화된 트레이딩 워크로드가 범용 스마트 컨트랙트 로직으로부터 논리적 및 물리적으로 분리되어, 각각의 고유한 운영 프로필에 맞게 최적화되도록 보장한다.

3.2.1 I. edgeVM: 고성능 트레이딩 런타임

edgeVM은 매칭 엔진, 실시간 리스크 관리, 청산 로직 등 고빈도 금융 작업에 독점적으로 설계된 미니멀리스트 실행 환경이다.

- **네이티브에 가까운 실행:** 범용 계산을 배제함으로써 edgeVM은 명령어 디코딩 및 복잡한 미터링 등 표준 스택 기반 EVM 아키텍처에 내재된 상당한 오버헤드를 제거한다. 대신 트레이딩 로직은 고효율 WebAssembly(WASM)로 컴파일되어(Arbitrum Stylus 같은 고급 스택 활용), Rust 또는 C++ 같은 언어로 작성된 성능 중요 코드가 베어메탈 실행에 가까운 속도를 달성할 수 있다.
- **전용 컴퓨팅 대역폭:** edgeVM은 보장된 CPU 및 메모리 할당을 갖춘 보호된 실행 컨텍스트 내에서 운영된다. 이 아키텍처적 방화벽은 범용 DeFi 트래픽의 "노이지 이웃" 혼잡으로부터의 면역을 보장하여, 광범위한 컴퓨팅 부하와 무관하게 핵심 거래 실행에 결정론적 지연 시간을 보증한다.

3.2.2 II. edgeEVM: 표준 호환성 런타임

트레이딩 핵심부와 병렬로 운영되는 edgeEVM은 광범위한 생태계와의 강력한 상호운용성을 유지하도록 설계된 완전 준수 이더리움 가상 머신이다.

- **표준 DeFi 로직:** edgeEVM은 ERC-20/721 자산 발행, DAO 거버넌스 메커니즘, 자금 관리 등 EVM 동등성이 필요한 작업을 처리한다.
- **개발자 경험:** 이 레이어는 매끄러운 컴포저빌리티를 보장하여, 기존 Solidity 스마트 컨트랙트와 표준 업계 툴링(예: Hardhat, Foundry)이 수정 없이 배포 및 활용될 수 있도록 한다.

3.2.3 III. VM 액터: 모듈식 확장성 프레임워크

핵심 트레이딩 및 호환성을 위한 주요 이분화 런타임 외에, EDGE Stack 아키텍처는 안전하고 미래 지향적인 확장성을 위한 전용 프레임워크를 포함한다. 이 프레임워크는 빠른 혁신과 시스템적 안정성 간의 긴장을 해소하도록 설계된 "VM 액터" 모델에 기반한다.

- **미래 지향적 확장성:** 모놀리식 업그레이드나 네트워크 포크 없이 옵션 거래소, 예측 시장, 실험적 수익 전략 등 완전히 새로운 금융 기본 요소를 추가하여 프로토콜이 수평적으로 진화할 수 있도록 한다.
- **핫스왑 모듈:** 신규 제품 라인은 동적으로 배포 및 통합될 수 있는 독립적이고 모듈적인 액터로 취급된다. 이는 프로토콜을 정적 코드베이스에서 신흥 DeFi 트렌드와 자산 클래스에 적응할 수 있는 유연한 플랫폼으로 전환시킨다.
- **아키텍처적 안전성:** 결정적으로 이 프레임워크는 새롭고 잠재적으로 실험적인 로직의 추가가 핵심 무기한 트레이딩 엔진의 보안이나 성능 무결성을 침해할 수 없도록 처음부터 설계되었다. 강력한 금융 인프라 위에서 허가 없는 혁신을 위한 가드레일을 제공한다.

3.3 결정론적 VM 간 상호운용성 프레임워크

그림 2: 모듈식 멀티-VM 실행 환경

모듈식 실행 아키텍처의 근본적 과제는 이질적인 환경 간에 동기적 컴포저빌리티를 유지하는 것이다. 표준적 접근 방식은 종종 상태를 별도의 사일로로 분산시켜, 허용할 수 없는 지연을 초래하고 원자적 금융 트랜잭션을 깨뜨리는 비동기 브리징이나 메시지 패싱에 의존한다.

EDGE Stack은 브리지가 아닌 통합된 실행 컨텍스트 내에서 운영되는 내재적이고 엄격하게 정의된 상호작용 프레임워크로서 호스트 함수 인터페이스(HFI)를 확립함으로써 이를 해결한다.

Arbitrum Stylus 같은 고급 통합 패러다임에서 영감을 받아, edgeVM과 edgeEVM은 별도의 프로세스로 운영되지 않는다. 대신 단일 블록 전이 사이클을 공유한다. HFI는 엄격하고 결정론적인 ABI(응용 프로그램 이진 인터페이스) 경계를 정의하여, 고성능 트레이딩 코어가 표준 EVM 상태 전이를 트리거하는 직접적이고 동기적인 시스템 호출을 수행할 수 있도록 하고, 그 역방향도 가능하게 한다.

3.3.1 I. 원자적 크로스 도메인 상태 전이

사용자가 edgeEVM의 표준 컨트랙트를 통해 USDC 담보를 예치하는 경우를 생각해보자. EDGE Stack에서 이것은 체인 간 비동기 "전송" 이벤트로 모델링되지 않는다. 대신 예치는 정확히 정의된 호스트 I/O 호출

을 트리거하여 동일한 실행 작업에서 edgeVM 트레이딩 엔진 내의 사용자 마진 잔액 상태를 원자적으로 변경한다.

3.3.2 II. 제로 복사 공유 메모리 아키텍처

VM 간 컨텍스트 전환과 관련된 컴퓨팅 오버헤드를 완화하기 위해, 이 프레임워크는 가능한 모든 곳에 제로 복사 기술을 활용한다. 경계를 넘어 직렬화 및 역직렬화 비용을 발생시키는 대신 제한된 공유 메모리 힙 내의 검증된 포인터를 통해 데이터를 전달함으로써, EDGE Stack은 복잡한 금융 작업에 대한 엄격한 결정론을 유지하면서 매끄럽고 원자적인 컴포저빌리티를 달성한다.

3.4 미리 컴파일된 컨트랙트를 통한 네이티브 최적화

edgeVM의 아키텍처가 일반적인 매칭 로직에 탁월한 성능을 제공하지만, 온체인 파생상품 거래는 가상화된 환경의 실질적 한계를 초과하는 고유한 종류의 컴퓨팅 부담을 초래한다. 실시간 마진 계산에 요구되는 산술 밀도와 고빈도 주문 제출의 암호학적 부하는 결정론적 지연 시간을 위협하는 병목을 만들어낸다.

이러한 도메인 특정 과제를 극복하기 위해 EDGE Stack은 특화된 미리 컴파일된 컨트랙트를 통합한다. 이는 범용 가속기가 아니라, 무기한 선물 거래소가 필요로 하는 가장 무거운 수학적 기본 요소를 처리하도록 목적에 맞게 설계되어 제로 오버헤드 호출을 위해 노드의 호스트 인프라에 직접 내장된 네이티브 실행 모듈이다.

3.4.1 I. 파생상품 컴퓨팅 부담 해결: 벡터화된 리스크 엔진

현물 AMM과 달리, 무기한 거래소는 지속적이고 복잡한 수학적 재평가를 요구한다. 시스템적 지급 능력 유지를 위해서는 수만 개의 활성 포지션에 걸쳐 교차 마진 포트폴리오, 마크 가격 업데이트, 펀딩 비율 계산을 실시간으로 모니터링해야 한다.

EDGE Stack의 PTE 같은 병렬화된 시스템에서 느린 리스크 점검은 전체 실행 웨이브를 지연시키는 "스트래글러 작업"이 된다. 리스크 계산이 주문 매칭을 절대 차단하지 않도록 하기 위해, EDGE Stack은 전용 리스크 미리 컴파일을 구현한다. 이를 통해 시스템은 극단적 변동성 중에도 마이크로초 내에 전체 시장 세그먼트의 상태를 재계산할 수 있어, 거래소 반응성을 저해하지 않으면서 강력한 청산 메커니즘을 보장한다.

3.4.2 II. HFT 인입 활성화: 네이티브 암호학적 기본 요소

EDGE Stack의 주요 사용자는 특히 FlashLane을 통해 대규모 주문, 취소, 수정 버스트를 생성하는 기관 시장 조성자들이다. 모든 메시지는 진위 증명을 위한 암호학적 서명 검증이 필요하다.

HFT 규모에서 초당 수천 개의 secp256k1 또는 ed25519 서명을 검증하는 데 필요한 누적 CPU 사이클은 지배적인 인입 병목이 된다. EDGE Stack은 서명 검증을 VM 바이트코드가 아닌 네이티브 프로토콜 기본 요소로 취급한다. 이 집약적인 워크로드를 고도로 최적화된 네이티브 미리 컴파일로 오프로딩함으로써, 시스템은 게이트웨이에서 지연 큐를 유발하지 않고 성숙한 금융 시장의 특성인 버스트적, 고용량 트래픽을 처리할 수 있도록 한다.

3.5 플러그 앤 플레이 확장 및 격리

3.2절에서 설명된 확장성 프레임워크를 기반으로, EDGE Stack은 구체적인 "VM 액터" 구현 모델을 통해 미래 지향적 확장성을 실현한다. 이 패러다임에서 새로운 예측 시장 코어나 실험적 옵션 엔진 등 별개의 기능 모듈은 긴밀하게 결합된 코드가 아닌 독립적이고 핫스왑 가능한 실행 단위로 취급된다.

3.5.1 I. 플러그 앤 플레이 배포 생애주기

"VM 액터"는 기술적으로 특정 PTE 인터페이스를 준수하도록 설계된 특화되고 미리 컴파일된 WASM 아티팩트(또는 표준 EVM 바이트코드 세트)로 구현된다. 새로운 기능 추가는 네트워크 포크를 피하는 허가 불필요 배포 생애주기를 따른다:

1. **배포:** 새 액터의 바이트코드가 edgeEVM 호환성 레이어에 대한 표준 트랜잭션을 통해 온체인에 배포된다.
2. **등록:** 배포된 컨트랙트가 중앙 온체인 프로토콜 레지스트리에 자신을 등록하여 운영 파라미터를 정의하고 특정 상태 접근 범위를 요청한다.
3. **활성화:** 일단 승인되면(거버넌스 프로세스 또는 사전 정의된 기준을 통해), PTE 스케줄러는 새 액터 ID를 동적으로 인식하고 이후 에폭부터 관련 트랜잭션 유형을 전용 실행 컨텍스트로 라우팅하기 시작한다.

3.5.2 II. 런타임 격리 강제 (샌드박스)

PTE 엔진이 트랜잭션 의존성을 기반으로 스케줄링을 처리하는 반면, 크로스 모듈 상태 손상에 대한 궁극적 보안 보장은 런타임 강제에 있다. 격리는 호스트 실행 환경에 의해 암호학적으로 강제된다.

등록된 각 VM 액터에는 엄격한 EIP-2930 원칙을 모델로 한 사전 정의된 읽기/쓰기 권한 범위인 상태 슬롯 매니페스트가 할당된다. 액터가 상태를 읽거나 쓰는 오퍼코드를 실행하려 할 때:

- **차단:** 작업이 글로벌 상태에 접근하기 전에 기반 호스트 환경에 의해 차단된다.
- **검증:** 대상 상태 슬롯이 액터의 허용된 매니페스트에 대해 엄격하게 점검된다.
- **강제:** 접근 시도가 액터의 지정된 샌드박스를 벗어나면(예: 실험적 모듈이 핵심 BTC-PERP 마진 잔액을 변경하려는 경우), 호스트는 즉시 작업을 되돌린다.

이 아키텍처적 방화벽은 새롭고 실험적인 모듈의 버그가 고유의 경계 내에 완전히 밀봉되도록 보장하여, 핵심 금융 인프라의 무결성을 보존한다.

4. 기동 II: 결정론적 병렬 트랜잭션 실행 (PTE)

4.1 앱 특화 우위: 낙관적 모델 대비 결정론적 실행

범용 블록체인은 고도로 이질적인 워크로드 전반에 걸쳐 최대의 다목적성을 위해 최적화해야 하며, 이는 종종 필요한 아키텍처적 타협으로 이어진다. 병렬 실행 영역에서 선도적인 범용 체인들은 일반적으로 "낙관적" 모델(예: Block-STM)을 채택한다. 이러한 엔진들은 상태 충돌이 드물다고 가정하여 처음에는 병렬로 트랜잭션을 실행하고 이후 충돌을 비용이 많이 드는 롤백 및 재실행을 통해 해결한다. 경쟁이 낮은 범용 트래픽에는 효과적이지만, 이 접근 방식은 높은 시장 변동성 시기에 뚜렷한 성능 상한에 도달하여, 금융 거래소가 최대 안정성을 요구하는 바로 그 시점에 예측 불가능한 지연 급등을 초래한다.

특화된 앱 특화 실행 레이어로서 EDGE Stack은 범용적 최적화를 넘어설 전략적 유연성을 보유한다. 임의의 스마트 컨트랙트를 지원할 필요에서 자유로운 아키텍처는 더 높은 상한을 가진 "결정론적" 병렬화 모델을

구현하기 위해 고빈도 파생상품 거래의 고유한 도메인 특성을 활용한다.

런타임에 충돌을 동적으로 반응하는 대신, PTE 엔진은 접근 목록을 통해 정적으로 정의된 파생상품 트랜잭션의 알려진 구조를 활용하여 실행이 시작되기 전에 의존성을 분석한다. 이 도메인 인식 접근 방식은 병렬로 실행되도록 예약된 트랜잭션이 충돌 없이 성공할 것을 보장하여, 범용 환경에서는 달성 불가능한 재실행 오버헤드 제로와 예측 가능한 초저지연 프로파일을 달성한다.

4.2 VM 액터를 통한 시장 샵딩 실행

EDGE Stack의 아키텍처는 중요한 도메인 통찰에 기반한다: 고빈도 무기한 선물 거래는 높은 수준의 자연적 상태 격리를 나타낸다.

범용 DeFi 프로토콜에서 단일 트랜잭션은 종종 여러 상호 연결된 컨트랙트와 상호작용한다(예: AMM 라우터가 세 개의 다른 유동성 풀을 통해 스왑을 라우팅). 이는 순차 실행을 강제하는 복잡하게 얽힌 상태 의존성을 만들어낸다.

반면 파생상품 거래에서 BTC-USDT 중앙 지정가 주문서(CLOB)에서 발생하는 매칭 이벤트는 ETH-USDT 오더북과 기능적으로 독립적이다. 이들은 매칭 단계에서 공통 상태를 공유하지 않는다—별도의 주문 목록, 별도의 인덱스 가격 피드, 별도의 매칭 엔진 로직 스레드를 보유한다.

PTE는 엄격한 액터 모델 설계 패턴을 사용하여 구현된 시장 샵딩 실행을 통해 이 내재적 도메인 구조를 활용한다.

그림 3: 결정론적 병렬 트랜잭션 실행(PTE) 흐름

4.2.1 I. 자율 VM 액터 ("공유 없음" 접근 방식)

이 프레임워크에서 모든 상장된 파생상품 시장(예: BTC-PERP, SOL-PERP)은 자율 VM 액터로 인스턴스화된다.

- **상태 캡슐화:** 각 액터는 자체의 변경 가능한 상태(CLOB 데이터 구조, 로컬 시장 파라미터, 최근 거래 이력)를 캡슐화하고 전용 단일 스레드 실행 컨텍스트를 보유한다.
- **엄격한 경계:** 결정적으로 액터들은 "공유 없음" 아키텍처로 운영된다. 메모리를 공유하지 않으며 서로의 상태를 직접 읽거나 수정할 수 없다. 시장 간 통신(예: 이후 생애주기에서의 교차 마진 점검)은 엄격한 격리 경계를 보장하는 PTE 스케줄러가 처리하는 결정론적 비동기 메시지를 통해서만 이루어진다.

4.2.2 II. 선형 확장성 달성

이 공유 없는 액터 모델의 근본적 이점은 매칭 엔진에 대한 거의 선형적인 확장성 실현이다.

모놀리식 블록체인 시스템에서 글로벌 상태 경쟁은 더 많은 CPU 코어를 추가하더라도 빠르게 수익 체감이 발생함을 의미한다. 통합된 상태 데이터베이스 전반에 걸쳐 잠금 및 스레드 동기화를 관리하는 오버헤드가 결국 추가 컴퓨팅 파워의 이점을 능가한다(암달의 법칙에 의해 지배되는 현상).

반면 EDGE Stack의 매칭 단계에서 시장 간 충돌이 사실상 존재하지 않기 때문에, 시스템은 주문 처리에 대한 잠금 경쟁을 제거한다. 따라서 물리적 하드웨어 스레드 추가는 비례적으로 더 높은 총 시스템 처리량(TPS)으로 직접 변환된다. 이상적으로 64코어 서버에서 64개의 별개 시장 액터를 실행하는 검증자 노드는 단일 코어 설정의 64배에 달하는 집계 매칭 처리량에 근접한다.

실제 운영에서는 CPU 코어가 개별 VM 액터에 독점적으로 잠기지 않고 필수 보조 시스템 작업, 특히 6.1절에서 상세히 설명된 병렬 머클화 프로세스와 동적으로 공유된다. 이 필요한 자원 공유에도 불구하고, 이 아키텍처는 EDGE Stack이 모놀리식 설계에 비해 최소한의 성능 저하로 다양한 자산 클래스에 걸쳐 대규모 동시 변동성 급등을 흡수할 수 있도록 한다.

4.3 충돌 해결 및 결정론 엔진

결국 일관성이나 잠재적 롤백을 수용한다면 고처리량 병렬성 달성은 상대적으로 간단하다. 그러나 금융 블록체인 인프라의 최고 과제는 직렬 의미론(serial semantics)을 엄격히 유지하면서 최대 병렬성을 달성하는 것, 즉 최종 상태가 트랜잭션을 정규 순서대로 하나씩 실행한 것과 수학적으로 동일하도록 보장하는 것이다.

EDGE Stack PTE 엔진의 핵심은 속도를 희생하지 않고 상태 경쟁을 엄격하게 해결하는 능력이다. 시퀀스 트랜잭션의 직렬 스트림을 병렬 작업의 고도로 최적화된 방향 비순환 그래프(DAG)로 변환한다. 이 변환은 충돌 감지를 런타임에서 사전 실행 분석으로 이동시키는 것에 의존한다.

4.3.1I. 기반: 확장 접근 목록(EAL) 및 공식화된 충돌 로직

표준 EVM 환경에서 트랜잭션의 상태 영향은 실행 중에 동적으로 발견된다. 스마트 컨트랙트는 변수 온체인 조건에 따라 다른 컨트랙트를 호출할 수 있으며, 코드가 실제로 실행될 때까지 정확히 어떤 스토리지 슬롯이 읽히거나 쓰일지 예측하는 것이 불가능하다. 이 동적 특성이 결정론적 병렬성의 주요 장애물이다.

EDGE Stack은 트랜잭션이 확장 접근 목록(EAL)을 통해 "상태 발자국"을 사전 선언하도록 요구하는 패러다임 전환을 도입한다. EAL은 트랜잭션이 접근하려는 상태의 범위를 정확하게 지정하는 필수 메타데이터 레이어다. 이 정적 선언을 강제함으로써 PTE 엔진은 실행이 시작되기 전에 잠재적 충돌에 대한 "신의 시점"을 확보한다.

EAL 기반 병렬 호환성 형식화: 임의의 트랜잭션 T_i 를 읽기 집합(R_i)과 쓰기 집합(W_i)의 튜플로 정적 정의할 수 있다:

$$| \quad T_i = (R_i, W_i) \dots (1)$$

정렬된 배치에서 임의의 두 트랜잭션 T_i 와 T_j ($i < j$)에 대해, 관련 상태 집합의 교집합이 공집합인 경우에만 이들은 병렬 호환($T_i \parallel T_j$)이다:

$$| \quad (W_i \cap R_j = \emptyset) \wedge (R_i \cap W_j = \emptyset) \wedge (W_i \cap W_j = \emptyset) \dots (2)$$

이는 동일한 실행 웨이브 내에서 쓰기 후 읽기(RAW), 읽기 후 쓰기(WAR), 쓰기 후 쓰기(WAW) 충돌의 부재를 보장한다.

4.3.2II. 결정론적 스케줄러 실행 웨이브

결정론적 스케줄러는 PTE 엔진의 두뇌다. 위에서 정의된 공식적 충돌 로직에 기반하여 원시 정렬 배치를 일련의 점진적 "실행 웨이브"로 재구성하기 위해 그리디 알고리즘을 활용한다.

1. **웨이브 구성:** 스케줄러가 배치를 순차적으로 분석하여 현재 웨이브에 큐 앞단의 최대 비충돌 트랜잭션 집합을 그룹화한다.
2. **병렬 실행:** 완성된 웨이브 내의 모든 트랜잭션이 사용 가능한 VM 액터 전반에 걸쳐 동시에 디스패치되어 완전한 격리 속에서 실행된다.

3. **의존성 해결:** 이전 웨이브의 상태 출력에 의존하는 트랜잭션은 이후 웨이브에서 예약되기 전에 해당 웨이브가 원자적으로 완료될 때까지 기다려야 한다.

이 웨이브 기반 접근 방식은 항상 모든 사용 가능한 CPU 코어가 유효하고 비충돌 트랜잭션을 처리하도록 보장하여 하드웨어 활용을 최대화한다.

4.3.3 III. 결정론적 프로세스 흐름

PTE 엔진을 통한 트랜잭션 배치의 전체 생애주기는 룰백에 의존하는 낙관적 실행 모델과 구별되는 엄격하고 재현 가능한 경로를 따른다:

1. **수집 및 검증:** 노드가 정렬된 배치를 수신하고 모든 트랜잭션의 형식과 EAL을 검증한다.
2. **DAG 생성(정적 분석):** 순전히 입력 순서와 EAL에 기반하여 어떤 트랜잭션이 다른 트랜잭션을 선행해야 하는지 매핑하는 인메모리 의존성 그래프가 구성된다.
3. **웨이브 스케줄링:** DAG가 위에서 설명된 알고리즘을 사용하여 결정론적 병렬 실행 웨이브 시퀀스로 평탄화된다.
4. **동시 실행 및 수렴:** 웨이브가 순차적으로 실행되며, 각 웨이브 내의 트랜잭션은 VM 액터 전반에 걸쳐 동시에 실행된다. 결과가 병합되어 새로운 최종 상태 루트를 형성한다.

전체 스케줄링 로직이 고정된 입력 배치와 정적 EAL의 순수 함수이기 때문에, 시스템은 스레드 타이밍이나 하드웨어 변동과 관계없이 직렬 의미를 엄격히 준수할 것을 보장한다. 이 정적 접근 방식은 고빈도 트레이딩 워크로드가 요구하는 예측 가능한 성능 프로파일을 제공한다.

4.4 핵심 서브시스템의 수평 확장

시장 샵딩 실행이 자산 격리에 기반하여 주문 매칭을 효과적으로 병렬화하는 반면, 강력한 무기한 선물 거래소는 종종 개별 시장 경계를 초월하는 계산 집약적인 보조 서브시스템에 의존한다.

모놀리식 아키텍처에서 글로벌 포트폴리오 리스크 계산 같은 함수는 방대한 상태 배열을 반복 처리하는 것을 포함한다. 높은 시장 변동성 기간에는 이러한 계산의 컴퓨팅 오버헤드가 트레이딩 볼륨이 가장 높은 시점에 정확히 급등한다. 주요 실행 루프 내에서 동기적으로 처리된다면, 이 서브시스템들은 매칭 엔진의 자원을 "굶기는" 주요 지연 원인이 된다.

EDGE Stack은 이러한 핵심 기능들을 자율적이고 비동기적인 VM 액터로 분리함으로써 이를 해결한다. 이 설계는 즉각적인 주문 순서에 엄격히 의존하지 않지만 계산적으로 무거운 서브시스템들이 핵심 매칭 흐름과 병렬로 실행될 수 있도록 한다.

4.4.1 I. 비동기 리스크 청산 엔진

파생상품 거래소의 안전성과 지급 능력은 두 가지 핵심 엔진에 의존하며, 둘 다 EDGE Stack에서 별도의 병렬 액터로 구현된다:

- **글로벌 리스크 엔진 액터:** 시스템 지급 능력 유지는 전체 포트폴리오에 걸쳐 모든 사용자의 마진 비율을 실시간 모니터링하는 것을 필요로 한다. 가격이 변동함에 따라, 시스템은 수만 개의 활성 계정에 대한 순자산 가치와 유지 마진 요건을 동시에 재계산해야 한다. 리스크 엔진을 자체 VM 액터로 격리함으로써, 이 대규모 계산 작업들이 비동기적으로 실행된다. 리스크 엔진은 완료된 거래와 오라클 가격 업

데이트 스트림을 소비하여 백그라운드에서 계정 상태를 업데이트하고, 무거운 계산이 새 주문의 매칭 엔진 진입을 절대 차단하지 않도록 보장한다.

- **청산 실행 액터:** 청산은 단일 이벤트가 아니라 복잡한 다단계 상태 전이 시퀀스로, 미결 주문 취소, 담보 압류, 포지션 청산을 위한 시장가 주문 실행, 보험 기금 업데이트를 포함한다. 동기적으로 실행된다면 대규모 청산 연쇄는 수십 밀리초 동안 시장을 정지시킬 수 있다. **EDGE Stack**에서 계정이 리스크 엔진에 의해 플래그되면, 청산 워크플로우가 전담 청산 액터로 넘겨진다. 이는 이러한 복잡한 작업들이 건 강한 시장 참가자들에 대한 지연 시간을 저하시키지 않고 원자적이고 효율적으로 실행되도록 보장한다.

4.4.2 II. 기관 수요를 위한 동적 탄력성

논리적 액터와 물리적 하드웨어 자원의 분리는 **EDGE Stack**에 전례 없는 탄력적 확장성을 제공한다. 리스크 엔진 같은 서브시스템들이 결정론적 메시지를 통해 통신하는 격리된 액터이기 때문에, 매칭 엔진과 동일한 물리적 CPU 코어에 바인딩되지 않는다.

- **정상 부하 시:** 단일 검증자 노드가 동일한 고성능 서버의 다른 코어에서 모든 시장 액터와 서브시스템 액터를 실행할 수 있다.
- **극단적 부하 시:** 네트워크 활동이 급증함에 따라, 시스템 아키텍처는 동적 부하 분산을 지원한다. 계산 집약적인 액터들(예: 폭락 중 실행되는 글로벌 리스크 엔진)은 검증자 클러스터 내의 전용 하드웨어 스레드 또는 심지어 별도의 물리적 서버 인스턴스에서 실행되도록 논리적으로 재배치될 수 있다.

이 수평 확장 역량은 **EDGE Stack**이 모놀리식 설계를 괴롭히는 자원 경쟁 없이 기관 알고리즘 트레이더의 대규모 처리량 수요를 흡수할 수 있도록 보장한다.

5. 기동 III: FlashLane: 프로토콜 네이티브 트랜잭션 우선순위

표준 모놀리식 블록체인에서 모든 트랜잭션은 동등하게 취급된다. 스테일 호가를 긴급하게 취소하려는 고빈도 트레이더가 에어드롭을 청구하거나 거버넌스 투표에 참여하는 사용자와 동일한 블록 공간 및 실행 자원을 위해 경쟁한다. 이러한 차별화 부재는 긴급하고 지연에 민감한 작업이 대용량의 긴급하지 않은 트랜잭션에 의해 지연되는 선두 라인 차단(**Head-of-Line Blocking**)으로 이어져, 예측 가능한 트레이딩 전략을 불가능하게 만든다.

EDGE Stack은 **FlashLane**이라는 프로토콜 네이티브 트랜잭션 우선순위 메커니즘을 도입함으로써 이 근본적 결함을 해결한다. **FlashLane**은 금융 시장에서 속도는 사치가 아니라 효과적인 리스크 관리의 근본 요건이라는 전제 위에 구축된다.

그림 4: FlashLane 프로토콜 네이티브 우선순위

5.1 트래픽 계층화: 이중 레인 아키텍처

기존 블록체인 아키텍처에서 시퀀서는 수동적인 모놀리식 **FIFO**(선입선출) 큐로 운영된다. 이 차별화되지 않은 접근 방식은 고성능 거래소에 근본적으로 결함이 있다. 핵심 시장 조성 호가가 무관하고 계산 집약적인 트랜잭션(예: 복잡한 거버넌스 투표 초기화) 뒤에 대기해야 하여, 선두 라인 차단으로 알려진 허용할 수 없는 지연 급등이 발생한다.

EDGE Stack은 시퀀서 인입 표면을 애플리케이션 인식 인입 컨트롤러로 근본적으로 재설계한다. 들어오는 트랜잭션 유형에 대한 실시간 의미론적 검사를 수행함으로써, 시스템은 운영 긴급도와 자원 프로파일에 기반하여 트래픽을 별개의 처리 레인으로 능동적으로 이분화한다.

5.1.1 I. 패스트 레인 (지연 민감 경로)

이 레인은 타이트한 스프레드 유지와 리스크 관리에 결정론적 지연 시간이 필수적인 원자적 고빈도 트레이딩 작업을 독점적으로 위해 예약된 전용 최적화 익스프레스 루트 역할을 한다.

- **작업 범위:** 지연에 민감한 트레이딩 지시로 엄격히 제한:
 - 신규 주문 제출(지정가, 시장가, IOC, FOK)
 - 주문 취소
 - 주문 수정/교체
- **선점 스케줄링 우선순위:** 패스트 레인으로 식별된 트랜잭션들은 단순히 우선순위가 부여되는 것이 아니라 선점 스케줄링 상태를 부여받는다. 시퀀서는 이 트랜잭션들이 정렬 큐에서 대기 중인 슬로 레인 작업을 우선 처리하도록 한다. PTE 엔진 내에서 패스트 레인 배치는 각각의 edgeVM 시장 액터 내에서 즉시 이용 가능한 다음 컴퓨팅 슬롯을 할당받아 최소한의 큐 대기 시간을 보장한다.

5.1.2 II. 슬로 레인 (비동기 정산 경로)

이 레인은 마이크로초 수준의 실행 보증이 필요하지 않은 거래소의 필수 운영 백본을 처리한다. 이러한 트랜잭션은 일반적으로 상태 집약적(복잡한 EVM 로직) 또는 데이터 집약적(DA를 위한 대용량 콜데이터 페이로드)으로 특징지어진다.

- **작업 범위:** 광대역 작업 포함:
 - 자산 입출금(브리징 작업)
 - 담보 관리 및 마진 조정
 - 거버넌스 행동 및 스테이킹 작업
 - edgeEVM에 배포된 스마트 컨트랙트와의 일반적 상호작용
- **아키텍처적 자원 격리:** 여기서 핵심 혁신은 단순한 트래픽의 논리적 분리가 아닌 물리적 자원 격리다. 대용량 슬로 레인 트랜잭션 처리는 주 고빈도 매칭 루프에 대해 엄격히 비동기적으로 처리된다.

edgeEVM 내의 무거운 상태 계산과, 최종 완결성 및 DA 커밋에 필요한 대용량 데이터 블록의 집약적 직렬화는 전용 별도 스레드 풀로 오프로딩된다. 이 아키텍처적 방화벽은 슬로 레인의 복잡하고 자원 집약적인 작업이 절대 "세계를 멈추거나" 패스트 레인 매칭 엔진의 마이크로초 수준 반응성을 저하시킬 수 없도록 보장한다.

5.2 FlashLane 커밋: 결정론적 소프트 확인

불변 온체인 정산(이더리움 L1 완결성 또는 표준 L2 합의)과 고빈도 트레이딩의 요건 사이의 지연 간극은 심각한 시장 비효율성을 구성한다. 기관 시장 조성자에게 헤지 실행에 대한 수초 간의 불확실성은 경제적으로 허용될 수 없어, 정확한 실시간 재고 관리와 가격 모델 업데이트를 방해한다.

저지연과 분산 정산 보증 사이의 간극을 메우기 위해 EDGE Stack은 즉각적이고 암호학적으로 보안된 소프트웨어 확인을 제공하도록 설계된 프로토콜 메커니즘인 FlashLane 커밋을 도입한다.

5.2.1 I. 즉각적 결정론적 사전 실행

패스트 레인으로의 인입 시, EDGE Stack 시퀀서는 단순히 미래 블록 포함을 위해 트랜잭션을 큐에 넣지 않는다. 대신 결정론적 PTE 엔진의 로컬 인스턴스를 사용하여 현재 헤드 상태에 대해 즉시 실행한다.

PTE의 상태 전이 함수가 순전히 고정된 입력 순서와 정적 EAL의 산물이기 때문에—네트워크 지터나 확률적 충돌 같은 런타임 비결정론이 없기 때문에—이 로컬 실행은 트랜잭션이 다음 최종 배치에 나타날 정확한 결과를 예측한다. 이는 근사치가 아니라 수학적으로 보장된 사전 계산이다.

5.2.2 II. 암호학적 커밋 수령

인입 후 밀리초 이하의 시간 내에, 시퀀서는 WebSocket을 통해 발신 트레이더에게 서명된 FlashLane 커밋을 발행한다. 이는 막연한 확인이 아니라 시퀀서의 권위 있는 개인 키로 서명된 구조화된 암호학적 수령이다. 수령에는 정확한 실행 메타데이터가 포함된다:

- 할당된 배치 ID
- 해당 배치 내 트랜잭션의 정확한 정렬 인덱스
- 결과적 사후 실행 상태 델타(예: 확인된 주문 체결 수량 및 가격, 업데이트된 계정 마진 잔액)

5.2.3 III. 경제적 완결성 및 전략적 체이닝

궁극적 프로토콜 완결성은 이더리움 L1 정산에 있지만, FlashLane 커밋은 트레이더에게 포함 순서와 실행 결과에 대한 수학적으로 엄격한 보증을 제공한다.

이 메커니즘은 "경제적 완결성"으로 알려진 것을 제공한다. 엔진의 결정론적 특성과 시퀀서의 암호학적 서명으로 인해, 재정렬 위험은 이론적 최악의 시퀀서 실패 시나리오로 최소화된다(이는 탈중앙화 로드맵 계획에 의해 추가로 완화된다). 이 높은 확신의 소프트웨어 확인은 시장 조성자들이 블록 확인을 기다리지 않고 즉시 내부 리스크 모델을 업데이트하고 후속 의존적 전략을 "체이닝"(예: 수 밀리초 후 다른 거래소에서 헤지 거래 배치)할 수 있도록 한다.

5.3 최적화된 I/O 및 마이크로 배칭

트래픽 계층화와 결정론적 소프트웨어 확인이 논리적 처리 흐름을 최적화하는 반면, 시스템의 초저지연의 물리적 무결성은 시퀀서의 데이터 수집 및 배치 형성 파이프라인을 급진적으로 아키텍팅하는 것에 달려 있다. 모놀리식 시스템에서 이러한 프로세스들은 종종 처리량(배치당 바이트 최대화)을 위해 최적화되어, 고빈도 트레이딩과 양립할 수 없는 지연 하한선을 생성한다.

EDGE Stack은 두 가지 주요 내부 메커니즘을 통해 "실행까지의 시간"을 무엇보다 우선시함으로써 시퀀서 내에서 이러한 최적화 파라미터를 역전시킨다.

5.3.1 I. 시간 트리거 마이크로 배칭

트랜잭션 제출과 실행 완결성 사이의 지연을 최소화하기 위해, EDGE Stack은 패스트 레인 트래픽에 대한 적극적 시간 트리거 마이크로 배칭 전략을 위해 크기 기반 배칭 임계값을 포기한다.

배치가 채워지기를 기다리는 동안 트랜잭션이 메모에서 유휴 상태로 있는 대신—범용 시스템에서 가변 지연의 일반적 원인—시퀀서는 초고속 시간 간격(밀리초 단위)으로 배치를 완결하고 디스패치한다. 이는 들어오는 시장 주문이 최소한의 큐 지연으로 패키징되어 PTE 엔진에 전송되도록 하여, 드문 대용량 블록 대신 실행 가능한 마이크로 배치의 연속적이고 고빈도 스트림을 생성한다.

5.3.2 II. 비동기 "핫 패스" 격리

인메모리 매칭 엔진의 예측 가능한 성능 유지는 무엇보다 중요하다. 많은 아키텍처에서 배치 완결과 관련된 무거운 I/O 오버헤드—데이터 구조 직렬화, 대용량 머클 상태 트리 차등 재계산, 네트워크 디스크 지속성 처리—는 매칭 연속성을 방해하는 간략한 "세계 정지" 일시 정지를 초래할 수 있다.

EDGE Stack은 주문 수집 및 매칭의 핵심 "핫 패스"가 이러한 "콜드 패스" 관리 작업에 의해 절대 차단되지 않도록 엄격한 아키텍처적 격리를 강제한다.

- **무거운 작업 분리:** 배치 직렬화, 암호학적 해싱, 지속적 저장 로깅의 물리적 프로세스들이 전용 비동기 스레드 풀로 오프로딩된다.
- **비차단 실행:** 이 설계는 이전 배치를 완결하는 부담스러운 작업이 현재 들어오는 흐름을 처리하는 시퀀서 핵심 스레드의 마이크로초 수준 반응성을 저하시키지 않고 완전히 백그라운드에서 이루어지도록 보장한다.

6. 상태 관리 및 보안

고성능 블록체인 아키텍처에서 실행 레이어에서 초고속 트랜잭션 처리량을 실현하는 것은 단지 초기 과제일 뿐이다. 실제 확장성에 영향을 미치는 주요 제약은 상태 관리 병목, 특히 글로벌 머클 패트리샤 트리(MPT)를 순차적으로 업데이트하는 데 필요한 방대한 암호학적 I/O 오버헤드다. 초당 수천 개의 거래를 처리하는 PTE 엔진은 기반 상태 커밋이 이를 따라가지 못하면 무용지물이 된다.

EDGE Stack은 상태 생애주기 관리의 전체적 재설계를 통해 이 근본적 과제를 해결한다. 결정론적 상태 수렴과 쌍을 이루는 동시 머클화를 구현함으로써, 시스템은 암호학적 무결성이 매칭 엔진 속도와 정확히 일치하도록 보장한다. 또한 이 고성능 상태 레이어를 모듈식 보안 아키텍처 내에 구성함으로써, EDGE Stack은 분산 네트워크에 필수적인 신뢰 최소화 검증 가능성을 완전히 계승하면서 중앙화 수준의 반응성을 달성한다.

그림 5: 상태 생애주기 및 완결성 아키텍처

6.1 병렬 머클화: 암호학적 I/O 병목 타파

6.1.1 I. 모듈식 병목: 직렬화된 상태 업데이트

모듈식 블록체인 시스템에서 상태 무결성은 일반적으로 MPT를 사용하는 단일 전역 변경 가능 데이터 구조를 통해 유지된다. MPT는 암호학적 검증에 매우 효율적이지만, 업데이트는 본질적으로 직렬적이다. 모든 개별 상태 전이(예: 잔액 수정 또는 주문 상태 업데이트)에 대해, 시스템은 쓰기 잠금을 획득하고 전체 "루트까지의 경로"를 따라 암호학적 해시를 재계산해야 한다.

트랜잭션 속도가 초당 수천 건으로 가속화됨에 따라, 이 반복적인 재해싱은 심각한 쓰기 증폭을 유발한다. 데이터 커밋 프로세스는 암달의 법칙에 의해 지배되는 비병렬화 직렬화 병목이 되어, 원시 실행 엔진 속도와 무관하게 총 시스템 성능을 제한한다.

6.1.2 II. EDGE Stack 솔루션: 표준 월드 상태 병렬화

EDGE Stack은 확립된 표준을 포기하지 않고 이 병목을 해결한다. 글로벌 계정 상태와 개별 컨트랙트 스토리지 상태 모두에 표준 MPT를 활용하는 이더리움 월드 상태 모델과 완전한 아키텍처적 정렬을 유지한다. 이는 미래 생태계 마이그레이션 및 툴링에 대한 최대 호환성을 보장한다.

데이터 구조를 재설계하는 대신, EDGE Stack은 결정론적 PTE 모델과 정확히 일치하도록 설계된 병렬 머클화를 통해 업데이트 프로세스를 혁신한다.

이는 세 가지 조율된 메커니즘을 통해 달성된다:

- **접근 목록 기반 병렬 해싱:** PTE 엔진이 수행하는 정적 분석(EAL을 통해)을 활용하여, 시스템은 주어진 실행 웨이브에서 수정될 상태 슬롯을 사전에 식별한다. PTE가 동시 트랜잭션들이 글로벌 트리 내의 수학적으로 분리된 경로를 건드리는 것을 보장하기 때문에, 다른 CPU 코어들이 경쟁 조건의 우려 없이 이러한 별도 경로를 따라 중간 해시를 동시에 재계산하는 무거운 작업을 할당받을 수 있다.
- **독립적 컨트랙트 스토리지 트리:** 표준 이더리움 아키텍처를 준수하여, BTC-PERP 엔진 같은 별개 VM 액터를 포함한 모든 컨트랙트 계정은 내부 상태를 보유하는 자체 독립적 스토리지 트리를 유지한다. 다른 시장들이 동시에 활성화되면, 시스템은 이 독립적 스토리지 트리에 대한 업데이트를 병렬 스레드로 계산한다. 이 스토리지 트리의 새 루트들은 이후 주 글로벌 계정 트리에 업데이트된다.
- **비동기 I/O 분리:** 결정적으로 이 집약적인 암호학적 작업은 주문 매칭 및 거래 실행의 핵심 "핫 패스"로부터 분리된다. 새 주문의 인입과 핵심 매칭 루프는 트리 해시가 완결되거나 디스크에 시각화될 때까지 절대 대기하지 않는다. 상태 커밋과 해시 계산은 백그라운드 프로세스로 취급되어, 지속성에 내재된 암호학적 지연이 거래소 인터페이스의 실시간 반응성을 저하시키지 않도록 보장한다.

그림 6: 병렬 머클화 전략

6.2 상태 루트 커밋: 결정론적 수렴

분리된 상태 경로에 대해 병렬화된 스레드를 통해 실행 및 초기 해싱이 이루어지지만, 표준 이더리움 월드 상태 패러다임 준수는 각 블록 높이에서 단일 정규 진실 정의를 요구한다. 병렬 처리의 분산된 출력들은 하나의 원자적 커밋으로 합성되어야 한다.

이 절은 이 수렴을 결정론적이고 효율적으로 달성하는 메커니즘을 설명한다.

6.2.1 I. 기반: 결정론적 인과 순서

안전한 상태 수렴의 전제 조건은 PTE 엔진이 제공하는 엄격한 순서 보증(4절에서 상세 설명)이다. 트랜잭션이 물리적으로 병렬로 실행되지만, 논리적 인과 시퀀스는 실행이 시작되기 전에 입력 순서와 의존성 분석에 의해 엄격히 고정된다.

이는 결과적 상태 변경이 글로벌 트리에 적용되는 시퀀스가 수학적으로 미리 결정됨을 의미한다. 업데이트 순서가 고정되어 있기 때문에 수렴 단계에서 경쟁 조건이 없어, 상태 재조립이 어떤 관찰자도 읽을 수 있는 완전히 결정론적 프로세스임을 보장한다.

6.2.2 II. 집계 메커니즘: 신속한 최상위 해싱

실행 웨이브 완료 후, 시스템은 최종 집계 단계에 진입한다. 이 시점에서 깊은 해시 재계산의 무거운 작업은 이미 병렬로 완료되었다(6.1의 메커니즘을 통해).

- **컨트랙트 루트 업데이트:** 독립적 VM 액터의 새롭고 최종화된 스토리지 루트(예: 업데이트된 BTC-PERP 시장 상태 트리)가 주 상태 관리자에게 전달된다.
- **버블 업:** 이 새로운 중간 루트들이 주 글로벌 계정 트리에 리프 업데이트로 삽입된다.
- **최종 커밋:** 이 특정 컨트랙트 계정으로 이어지는 경로들만 변경되었기 때문에, 시스템은 글로벌 계정 트리의 최상위 레이어만 신속하게 직렬로 재해시하면 된다.

이 최종 작업은 단일 글로벌 상태 루트를 생성한다. 이 해시는 특정 블록 높이에서 전체 거래소 생태계 전반에 걸쳐 모든 사용자 잔액, 미결 주문, 리스크 파라미터의 정확한 상태를 나타내는 불변의 암호학적 커밋을 형성하며, 정산 준비가 완료된다.

6.3 모듈식 보안 아키텍처: 신뢰의 앵커

EDGE Stack이 고성능 독자적 실행 환경으로 운영되는 반면, 그 보안 모델은 모듈식 블록체인 테제에 기반한다. 초기 반복에서 최종 보안을 위해 격리된 검증자 집합을 부트스트랩하는 대신, EDGE Stack은 실행 속도를 정산 보안으로부터 분리하여 가장 강력한 인프라 레이어에 합의 및 데이터 가용성을 아웃소싱한다.

이 아키텍처는 거래소의 보안 속성이 운영자의 정직성이 아닌 강력한 기반 레이어(예: 이더리움)의 암호경제적 보증에 고정되도록 보장한다.

6.3.1 I. 플러그 가능한 정산 앵커

EDGE Stack 시퀀서가 생성하는 결정론적 글로벌 상태 루트는 암호학적 체크포인트 역할을 한다. 이 루트들은 주기적으로 모듈식 정산 레이어에 게시된다. 이 고정된 거래소의 최종화된 상태가 선택된 기반 레이어의 불변 완결성 속성을 계승하여, 모든 상태 전이에 대한 객관적이고 변조 방지된 이력을 제공하도록 보장한다.

6.3.2 II. 데이터 가용성 기반

보안은 근본적으로 데이터 가시성에 의존한다. EDGE Stack은 상태를 재구성하는 데 필요한 압축된 원시 트랜잭션 입력을 고무결성 DA 레이어에 브로드캐스트한다. 이 데이터가 기반 레이어에서 공개적으로 접근 가능하고 검열에 강하도록 보장함으로써, 시스템은 시퀀서 운영 상태와 무관하게 항상 검증의 원자재를 사용할 수 있음을 보증한다.

6.4 투명성 및 검증 가능성: "신뢰하지 말고 검증하라" 메커니즘

분산 시스템의 궁극적 안전장치인 허가 없는 감사다. 6.3절이 데이터와 커밋이 저장되는 곳을 설명하는 반면, 이 절은 참가자들이 무결성을 보장하기 위해 어떻게 그 인프라를 활용하는지 상세히 설명한다.

EDGE Stack은 앞서 설명된 모듈식 데이터 가용성과 롤백 없는 결정론적 PTE 엔진을 결합하여 급진적 투명성 체제를 가능하게 한다. 이 조합은 높은 성능이 검증 가능한 정확성을 희생시키지 않도록 보장한다.

- **I. 왓치 노드를 통한 허가 없는 감사:** 기관 시장 조성자, 감사자, 커뮤니티 연구자 등 모든 외부 참가자가 왓치 노드를 운영할 수 있다. DA 레이어에 저장된 원시 트랜잭션 배치를 독립적으로 다운로드하고 로컬에서 검증된 EDGE Stack 소프트웨어 인스턴스에 공급함으로써, 노드는 메인 네트워크에서 활성화된 상태 전이의 정확한 시퀀스를 재현한다.
- **II. 암호학적 책임:** 실행 경로가 사전 실행 분석을 통해 엄격히 고정되어 있기 때문에(비결정론적 런타임 충돌 해결을 피하여), 왓치 노드는 시퀀서가 정산 레이어에 게시한 것과 수학적으로 동일한 글로벌 상태 루트에 도달해야 한다.

로컬로 계산된 루트가 게시된 루트와 수학적으로 다르다면, 이는 시퀀서에 의한 부정행위의 불변 암호학적 증거를 구성한다. 이 수학적 확실성은 사기 증명 같은 신뢰 없는 책임 프로토콜의 기반을 형성하여, 신뢰된 운영자 없이도 시스템이 정직하게 유지되도록 보장한다.

7. 결론

EDGE Stack은 고빈도 분산 금융을 방해하는 구조적 모순을 해결하도록 설계된 블록체인 인프라의 패러다임 전환을 도입한다. 스택을 범용 블록체인이 아닌 트레이딩을 위해 설계된 앱 특화 실행 레이어로 재상상함으로써, 핵심 Web3 원칙을 타협하지 않고 중앙화된 거래소와 온체인 환경 사이의 광대한 성능 격차를 메운다.

단일 글로벌 상태 머신에서 자원 경쟁을 위해 금융 로직이 경쟁하도록 강제하는 모놀리식 설계와 달리, EDGE Stack은 새로운 모듈식 멀티-VM 아키텍처를 통해 실행을 분리한다. 낙관적 모델을 결정론적 PTE 엔진으로 대체하여, 시장 샵딩된 상태를 활용하여 선형 확장성과 재실행 오버헤드 제로를 달성한다. 또한 프로토콜 네이티브 트랜잭션 우선순위 메커니즘인 FlashLane을 구현함으로써, 기관 시장 조성자가 요구하는 밀리초 이하의 지연 시간을 제공한다.

궁극적으로 EDGE Stack은 비수탁형이고 완전히 검증 가능하며 초저지연의 독자적 실행 환경을 확립한다. 이 아키텍처는 온체인 금융 인프라의 경계를 재정의를 하며, EDGE Stack을 차세대 글로벌 시장 구조의 토대 레이어로 포지셔닝한다.

8. 향후 연구

그 핵심에서 EDGE Stack은 다양한 금융 작업을 위한 탄력적 기반으로 설계되었다. 초기 배포가 무기한 선물 거래 발전에 초점을 맞추는 반면, 기반 아키텍처—특히 PTE 엔진과 모듈식 런타임 환경—는 여러 자산 클래스와 시장 유형에 걸쳐 일반화되도록 목적에 맞게 설계되었다.

EDGE Stack의 비전은 현재 구성을 넘어 확장된다. 우리는 금융 역량을 확대하고 신흥 기술을 통합하는 여러 전략적 방향을 적극적으로 탐색하고 있다:

- **모듈식 및 독자적 정산 진화:** EDGE Stack의 보안 아키텍처를 발전시키는 경로를 연구하고 있다. 이는 정산 레이어를 완전히 분리하기 위한 모듈식 설계를 활용하여, 시스템이 대안적 고처리량 데이터 가용성(DA) 레이어에 상태를 고정하거나 미래에 자체 합의를 갖춘 독자적 네트워크로 확립하는 것을 허용한다. 이 유연성은 진화하는 블록체인 환경에 대한 장기적 적응성을 보장한다.

- **기관 프라이버시 존:** 대규모 참가자들의 요구를 인식하여, 선택적 프라이버시 보존 트레이딩 존을 도입할 계획이다. 전용 VM 액터 내에서 영지식 증명(ZKP) 또는 신뢰 실행 환경(TEE) 같은 기술을 활용하여, 이 존들은 정보 유출 및 프론트러닝을 완화하기 위한 암호화된 오더북 같은 기능을 제공하여 기관 자산을 위한 시장 무결성을 더욱 강화할 것이다.
- **크로스 체인 상호운용성 표준:** 멀티 체인 생태계 전반에 걸쳐 매끄러운 유동성 이동과 통합 자산 관리를 가능하게 하기 위해 Circle의 크로스 체인 전송 프로토콜(CCTP) 같은 네이티브 신뢰 없는 상호운용성 표준을 통합하는 데 전념하고 있다.
- **네이티브 AI 기반 금융 인텔리전스:** EDGE Stack의 고성능 결정론적 실행 환경은 머신러닝을 트랜잭션 생애주기에 직접 통합하는 데 독보적으로 적합하다. 금융에 특화된 ML 모델을 실행하는 등 온체인 추론에 최적화된 전용 "AI VM 액터"를 탐색하고 있다. 이는 시장 유동성과 함께 실행 엣지에 직접 코로케이션된 정교하고 데이터 기반의 트레이딩 전략 및 실시간 동적 리스크 평가를 실행할 수 있는 새로운 종류의 자율 금융 에이전트를 가능하게 할 것이다.

이러한 발전을 통해 EDGE Stack은 고효율의 목적 지향적 트레이딩 기본 요소를 프라이버시, 상호운용성, 네이티브 지능 역량과 결합한 포괄적이고 다중 패러다임 금융 인프라로 발전하는 것을 목표로 한다.

감사의 말

EDGE Stack의 개발은 고성능 컴퓨팅 및 분산 금융 커뮤니티의 선도적 전문가들과의 지속적인 대화로부터 크게 도움을 받았다. 본 논문과 기반 연구를 형성한 병렬 실행 모델, 상태 머신 아키텍처, 시장 미시구조 설계에 관한 비판적 피드백과 기술적 통찰에 감사드린다.

특히 Offchain Labs(Arbitrum) 팀과 차세대 실행 환경을 조사하는 광범위한 연구 커뮤니티의 선구적인 작업에 감사를 표하고자 한다. 이 분야에 대한 그들의 기여는 문제 공간을 정의하고 앱 특화 실행 레이어에 대한 우리의 접근 방식에 영감을 주는 데 중요한 역할을 했다.

참고문헌

[1] Offchain Labs. (2024). "Stylus: Rust, C, and C++ on Arbitrum." Arbitrum Documentation. <https://docs.arbitrum.io/stylus/gentle-introduction>

[2] Wood, G. (2016). "Polkadot: Vision for a heterogeneous multi-chain framework." White Paper.

[3] Amdahl, G. M. (1967). "Validity of the single processor approach to achieving large scale computing capabilities." In Proceedings of the April 18-20, 1967, spring joint computer conference (pp. 483-485).

[4] Gelashvili, R., Spiegelman, A., Xiang, Z., et al. (2023). "Block-stm: Scaling blockchain execution by turning ordering curse to a performance blessing." Proceedings of the 28th ACM SIGPLAN Annual Symposium on Principles and Practice of Parallel Programming (pp. 232-244).

[5] Buterin, V., & Swende, M. (2020). "EIP-2930: Optional access lists." Ethereum Improvement Proposals. <https://eips.ethereum.org/EIPS/eip-2930>

- [6] Ethereum Foundation. (n.d.). "Merkle Patricia Trie." Ethereum Developers Documentation. <https://ethereum.org/en/developers/docs/data-structures-and-encoding/patricia-merkle-trie/>
- [7] Shrier, I., & Platt, R. W. (2008). "Reducing bias through directed acyclic graphs." *BMC Medical Research Methodology*, 8(1), 70.
- [8] Alchemy. (2023). "Modular vs. Monolithic Blockchains." Alchemy Overviews. <https://www.alchemy.com/overviews/modular-vs-monolithic-blockchains>
- [9] Buterin, V. (2021). "An Incomplete Guide to Rollups." Vitalik.ca. <https://vitalik.eth.limo/general/2021/01/05/rollup.html>
- [10] Celestia. (n.d.). "Data Availability Layer." Celestia Documentation. <https://docs.celestia.org/learn/how-celestia-works/data-availability-layer>
- [11] Dodmane R, KR R, NS K R, et al. Blockchain-based automated market makers for a decentralized stock exchange[J]. *Information*, 2023, 14(5): 280.
- [12] Zhang, F., Cecchetti, E., Croman, K., Juels, A., & Shi, E. (2016). "Town Crier: An Authenticated Data Feed for Smart Contracts." In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security* (pp. 270-282).
- [13] Herlihy, M., & Moss, J. E. B. (1993). "Transactional memory: Architectural support for lock-free data structures." *ACM SIGARCH Computer Architecture News*, 21(2), 289-300.
- [14] Al-Bassam, M., Sonnino, A., Bano, S., Hrycyszyn, D., & Danezis, G. (2019). "LazyLedger: A Distributed Data Availability Ledger with Client-Side Validation." *arXiv preprint arXiv:1905.09274*.
- [15] Papadimitriou, C. H. (1979). "The serializability of concurrent database updates." *Journal of the ACM (JACM)*, 26(4), 631-653.
- [16] Circle Internet Financial. (2023). "Cross-Chain Transfer Protocol (CCTP) Technical Documentation." Circle Developer Docs.
- [17] Daian, P., Goldfeder, S., Kell, T., Li, Y., Zhao, X., Bentov, I., ... & Juels, A. (2020). "Flash boys 2.0: Frontrunning in decentralized exchanges, miner extractable value, and consensus instability." In *2020 IEEE Symposium on Security and Privacy (SP)* (pp. 910-927). IEEE.
- [18] Nakamoto, S. (2008). "Bitcoin: A Peer-to-Peer Electronic Cash System."
- [19] Dijkstra, E. W. (1982). "On the role of scientific thought." *Selected writings on computing: a personal perspective* (pp. 60-66). Springer, New York, NY.
- [20] Buterin, V. (2022). "The different types of ZK-EVMs." Vitalik.ca. <https://vitalik.eth.limo/general/2022/08/04/zkevm.html>
- [21] Croman, K., Decker, C., Eyal, I., Gencer, A. E., Juels, A., Kosba, A., ... & Wattenhofer, R. (2016). "On scaling decentralized blockchains." In *International Conference on Financial Cryptography and Data Security* (pp. 106-125). Springer, Berlin, Heidelberg.

[22] Scharf, M., & Kiesel, S. (2006). "Head-of-line Blocking in TCP and SCTP: Analysis and Measurements." In IEEE Globecom 2006 (pp. 1-5). IEEE.

[23] Mohan V. Automated market makers and decentralized exchanges: a DeFi primer[J]. Financial Innovation, 2022, 8(1): 20.