J. Internet Comput. Serv.
ISSN 1598-0170 (Print) / ISSN 2287-1136 (Online)
http://www.jics.or.kr

금융산업 분야에서의 EVM 기반 블록체인 선정을 위한 성능 측정 연구

Performance Measurement for Selecting EVM-based Blockchain in the Financial Industry

모 상 일[1] 이 재 준[1] 서 병 완[2*]
Sang Il Mo Jae Jun Lee Byung Wan Suh

요 약

이더리움 가상 머신의 등장으로 블록체인 플랫폼을 활용한 탈중앙화 서비스들이 다양한 산업 분야에 적용되었고 특히 금융 산업 분야에서 가장 큰 비중을 차지하고 있다. 그러나 블록체인 플랫폼의 낮은 거래 처리 속도는 높은 성능을 요구하는 금융 서비스에 적합하지 못하였고 이를 해결하고자 다수의 이더리움 가상 머신 기반 블록체인 플랫폼들이 출시되었다. 하지만 블록체인 플랫폼 간의 성능 측정 및 비교 연구가 희소하여 금융 서비스에 적합한 플랫폼 선정에 어려움이 있다. 따라서 본 논문에서는 금융 산업 분야를 겨냥한 대표적인 이더리움 가상 머신 기반의 블록체인 플랫폼을 선정하여 성능을 측정하였다. 결과를 기반으로 금융 산업 분야에서 서비스 운영에 적합한 블록체인 플랫폼 선정에 기여할 수 있을 것으로 기대한다.

☞ 주제어 : 블록체인, 이더리움 가상 머신, 성능 측정, 스마트 컨트랙트, 금융 플랫폼

ABSTRACT

The emergence of the Ethereum Virtual Machine(EVM) has facilitated the application of decentralized services across various industries, with the financial sector representing the largest portion of these applications. However, the low transaction processing speed of blockchain platforms has proven inadequate for high-performance financial services. To address this issue, numerous blockchain platforms based on the Ethereum Virtual Machine have been developed. Nevertheless, there is a scarcity of studies comparing the performance of these blockchain platforms, making it difficult to select the most suitable platform for financial services. Therefore, in this paper, we evaluate the performance of several prominent EVM-based blockchain platforms targeting the financial sector. Based on the results, we aim to contribute to the selection of blockchain platforms that are best suited for service operations within the financial industry.

☞ keyword : Blockchain, Ethereum Virtual Machine, Performance Evaluation, Smart Contract, Financial Platform

1. 서 론

블록체인은 원장에 기록된 내용을 모든 참여자가 검증하고 관리하여 위변조가 불가능하고 신뢰성과 투명성을 제공하는 분산원장기술로, 2008년 사토시 나카모토의 비트코인에 의해 처음 소개되었다[1]. 이는 중앙화된 기관 없이 암호학적 알고리즘을 통해 누구나 신뢰할 수 있는 거래를 가능하게 하며 전통적인 금융시장에서 법적 제제와 중앙금융기관의 관리를 통해 안전을 보장하는 것과는 반대되는 개념이다[2].

비트코인은 거래 시스템을 최적화하기 위해 거래에 특화된 연산에 필요한 연산자(e.g. OP_PUSHDATA, OP_IF)만으로 한정하여 시스템을 구성하였다. 이는 비트코인이 튜링 불완전성(turing-incompleteness)을 가지게 하여 범용적 시스템에는 적용이 불가능했고, 이에 따라 실제 산업에 응용하기에는 어려운 점이 있었다. 이를 해결하기 위해 2013년 등장한 이더리움은 튜링 완전성을 갖춘 Ethereum Virtual Machine(이더리움 가상 머신, 이하 EVM)을 개발하였고 블록체인상에서 응용 애플리케이션을 제작할 수 있는 기반이 되었다[3].

이에 금융, 유통, 헬스케어, 그리고 에너지 등 여러 서비스 산업에서 블록체인을 적용하고 있으며 특히 금융 분야에서의 움직임이 활발하다. Fortune Business Insights

1 R&D Center, FairsquareLab, Seoul, 06524, Korea
2 The Institute for Industrial Policy Studies, Seoul, 03767, Korea
* Corresponding author (byungwan.suh@gmail.com)
[Received 1 August 2024, Reviewed 14 August 2024, Accepted 4 October 2024]

의 2023년 블록체인 시장 연구 보고서에 따르면 EVM 기반 금융 분야 서비스가 전체 서비스의 32% 이상 비중을 차지하는 것으로 밝혀졌다. 이는 블록체인의 사용자 금융 데이터를 안전하게 관리해 줄 수 있는 신뢰성을 바탕으로 금융시장에서의 활용성이 풍부해졌다고 볼 수 있다[4-8].

이러한 활용성을 바탕으로 EVM에 기반한 다양한 블록체인 플랫폼이 금융시장의 주요 플랫폼으로 자리 잡기 위해 추가로 등장했다[9]. 먼저, Avalanche(이하 아발란체)는 금융 서비스의 높은 트래픽을 처리하기 위해 개발되었으며 자체적인 합의 알고리즘을 통해 블록 확정 시간을 크게 줄였다[10,11]. 다음으로 Sei(이하 세이)는 EVM 기반의 트랜잭션 병렬 실행 방안을 처음으로 제안하며 기존 블록체인의 순차적 트랜잭션 실행으로 인한 성능 병목 현상을 해결하였다[12]. 마지막으로 Hyperledger Besu(하이퍼렛저 베수, 이하 베수)는 Hyperledger Foundation의 프로젝트 중 하나로 Java 기반의 EVM 호환가능한 블록체인이다[13]. 베수는 금융 분야의 높은 보안 수준을 준수하기 위해 트랜잭션 암호화와 노드 및 사용자 계정에 대한 권한 설정을 제공한다[14].

이처럼 블록체인 플랫폼이 다양해지는 반면, 서비스 제공자 입장에서는 각 플랫폼 간에 객관적인 성능을 비교한 실험이 매우 희소하여, 기존에 제안된 멀티 블록체인 플랫폼에 대한 성능 측정 프레임워크 COCONUT[15] 및 Hyperledger Caliper[16] 등은 자체적인 테스트 환경을 제공하지만 이로 인해 성능 측정 가능한 기능들은 제한적이게 되었다[15]. 따라서 금융 서비스에 대해 구체적으로 요구되는 기능에 대해 객관적인 성능 측정이 어려운 문제가 있다.

본 논문에서는 EVM 기반 블록체인 플랫폼들에서 실제 금융 서비스 환경에 있어 주요한 성능 지표들을 정리하고 측정 가능한 객관적인 프레임워크를 제안한다. 또한, 위에서 살펴본 세 가지 플랫폼(아발란체, 세이, 베수)에 대해 성능 측정 및 비교 연구를 통해 금융 산업을 겨냥한 EVM 블록체인 플랫폼 선정에 객관적인 지표를 제시하고자 한다.

2. 이론적 배경

2.1 블록체인

2.1.1 블록체인 개념

블록체인은 P2P 네트워크 기반 분산원장 기술로, 거래 기록을 중앙서버가 아닌 각 참여 노드들이 공동으로 합의하고 저장하여 데이터의 투명성을 제공한다. 블록체인은 다수의 거래(트랜잭션)를 블록 단위로 묶어 저장하고, 새로운 블록은 이전 블록의 해시와 연결되어 체인 구조를 형성한다. 이는 임의의 블록 데이터를 위변조할 경우 해당 블록 이후의 블록 해시들이 유효성을 상실함(invalid)으로 데이터의 불변성과 무결성을 보장한다. 또한, 블록체인은 암호학을 활용하여 분산된 환경에서 서로의 안전한 메시지 전송 및 검증이 가능한 무신뢰성(trustlessness)을 보장한다[17].

2.1.2 튜링 완전성과 비트코인

튜링 완전성(turing-completeness)이란 수학자 앨런 튜링이 1936년에 제시한 개념으로 어떤 프로그래밍 언어나 가상 머신이 튜링 머신과 동일한 계산 능력으로 문제를 풀 수 있다는 것을 의미한다[18]. '튜링 머신'이란 컴퓨터의 일반적 개념을 설명하기 위한 가상의 머신이다[19]. 블록체인이 등장하면서 이러한 튜링 완전성에 대한 평가도 동시에 이루어졌다. 비트코인 스크립트 언어는 화폐 송금 기능을 수행하는 단순한 연산자만을 지원하며, 복잡한 명령어 또는 함수의 구현을 지원하지 못한다[20]. 이는 비트코인의 단점으로 간주될 수 있으나, 실행 복잡도가 높은 명령을 의도적으로 방지하여 플랫폼의 안전성을 강화하려는 목적도 있다. 반면, 이러한 문제를 극복하기 위해 개발된 이더리움은 조건문, 반복문 등의 기능을 포함한 언어를 도입하여 블록체인상에서 복잡한 명령어와 함수의 실행이 가능하다.

2.1.3 스마트 컨트랙트(Smart Contract)

스마트 컨트랙트는 계약 실행 조건과 계약 내용이 코드로 작성되어 조건 충족 여부에 따라 자동으로 계약 실행이 가능한 디지털 형태의 프로그램이다[21]. 블록체인 플랫폼에서 스마트 컨트랙트는 먼저 계약 코드와 상태(state)가 블록체인에 저장되고, 노드들은 저장된 코드에 대한 실행문과 파라미터를 트랜잭션으로 제안할 수 있다. 이 트랜잭션들은 노드들의 합의를 거쳐 실행됨으로써 계약이 성사되고 상태가 업데이트된다. 이러한 블록체인 플랫폼 기반의 스마트 컨트랙트는 다음과 같은 4가지의 주요 특징을 가진다[22].

첫째, 자동화가 가능하다. 이는 블록체인 플랫폼에서 인간의 개입 없이 특정 조건을 만족한 경우 요구되는 계약 내용을 자동으로 실행시킬 수 있다.

둘째, 데이터에 대한 투명성을 보장한다. 스마트 컨트랙트에 작성된 계약 내용의 코드와 상태가 모두 블록체인에 기록되므로 누구나 언제든지 확인이 가능하다.

셋째, 높은 보안성을 가진다. 스마트 컨트랙트의 데이터는 블록체인 플랫폼의 성질인 불변성을 가지며, 노드들의 합의에 의해서만 실행이 가능하므로 악의적인 행위를 방지할 수 있다.

마지막으로, 분산 네트워크상에서 동작함에 따라 탈중앙성을 보장한다. 따라서 제 3자에 대한 신뢰가 필요하지 않으며 참여 노드들의 검증을 통해 보다 안전한 서비스 구현이 가능하다.

2.2 이더리움 가상 머신(EVM)

2.2.1 EVM 정의 및 목적

EVM은 이더리움에서 비트코인의 튜링 불완전성을 해결하고자 개발한 튜링 완전 머신이다. 스마트 컨트랙트를 통해 블록체인상에서 동작하는 복잡한 애플리케이션을 구현할 수 있도록 하였다. EVM은 스마트 컨트랙트를 실행할 때 컨트랙트 코드를 컴파일하여 바이트 코드로 변환하고 각 바이트에 해당하는 연산자 코드(Opcode)를 실행한다. 따라서 EVM은 Opcode 별로 정해진 프로세스를 통해 트랜잭션을 처리하며 해당 Opcode들로 범용적 애플리케이션을 모두 구현 가능함에 따라 튜링 완전한(turing-completed) 머신으로 구분된다[23]. 또한 블록체인의 전체 상태에 대한 관리 규칙이 EVM에 정의되어 사용자의 주소, 잔액 그리고 스마트 컨트랙트의 코드와 상태에 대한 저장과 갱신을 담당한다[24].

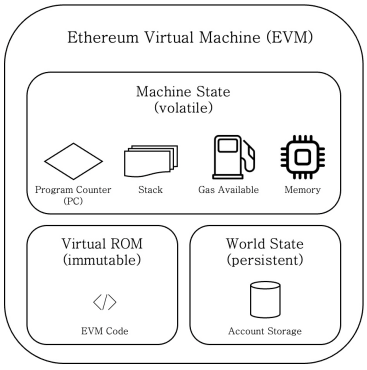
2.2.2 EVM 구조

EVM의 구조는 그림 1과 같이 Machine State와 Virtual ROM 그리고 World State로 구성된다[25].

Machine State는 트랜잭션을 실행하는 모듈로 Opcode를 처리하기 위해 필요한 컴포넌트인 Program Counter, Memory 그리고 Stack으로 구성된다. Machine State는 무한 반복과 같은 과도한 연산으로 인해 블록체인 플랫폼에 오류가 발생하는 것을 방지하고자 Gas Available을 통해 실행 연산량에 비례하여 소모되는 Gas를 측정함으로 실행 가능 여부를 사전 검사한다. 따라서 실행이 불가능한 트랜잭션은 Machine State에서 1차적으로 거절된다. 트랜잭션 실행 이후 관련된 데이터는 더 이상 사용되지 않아 삭제되므로 영구적 저장소가 아닌 휘발성(volatile) 메모리를 사용한다.

Virtual ROM(Read-Only Memory)은 Opcode에 대해 실행 가능한 형태의 바이트 코드인 EVM code를 저장하는 공간이다. EVM code는 변경되지 않으므로 읽기 전용 메모리를 사용하며 이에 따라 불변적(immutable) 성질을 가진다.

World State는 블록체인 전체 상태를 저장하는 스토리지이다. 해당 스토리지는 사용자 및 스마트 컨트랙트 주소를 키(key)로 하여 상태를 값(value)으로 저장하는 키 벨류(key-value) 형태의 스토리지로 이루어져 있다.



(그림 1) EVM 컴포넌트 구조
(Figure 1) EVM Component Structure

2.2.3 EVM 장단점

EVM은 블록체인상에서 애플리케이션을 실행할 수 있는 환경을 제공하며, 튜링 완전성, 가스 시스템, 그리고 샌드박스 환경 제공이라는 장점이 있다.

첫째, EVM은 튜링 머신을 만족하기 위한 모든 Opcode를 지원함으로써 튜링 완전성을 확보했으며, 이에 따라 개발자들은 다양하고 복잡한 로직과 기능을 가진 스마트 컨트랙트를 작성할 수 있게 되었다. 이는 블록체인이 단순한 화폐 송금을 넘어 금융 서비스, 게임, 공급망 관리 등 여러 분야에서 요구되는 복잡한 기능의 탈중앙화 애플리케이션을 구현할 수 있도록 하였다. 따라서 EVM의 튜링 완전성으로 인해 이더리움은 많은 분야에서의 활용도를 높여 사용성을 극대화했다[26].

둘째, EVM의 가스 시스템은 스마트 컨트랙트의 연산 자원 사용을 측정하고 제어하는 메커니즘으로, 각 연산은 고유의 가스 비용을 가지며, 사용자는 스마트 컨트랙트 실행 시 이에 대한 비용을 지불해야 한다. 가스 시스템은 네트워크 자원의 효율적인 사용을 촉진하고, 무한

루프와 같은 과도한 자원 소모를 방지하여 네트워크의 안정성을 유지시킨다. 또한, 가스 시스템은 스마트 컨트랙트 개발자가 컨트랙트 코드의 연산 비용을 절감하는 효율적인 방향으로 개발하도록 유도하고 이는 전체 네트워크의 성능 향상에 기여한다[27].

셋째, EVM은 스마트 컨트랙트를 샌드박스 환경에서 사전 실행하여 실행 결과가 네트워크나 다른 컨트랙트에 직접적인 해를 끼치지 않는지 검증함으로써 네트워크의 안전에 기여한다. 다시 말해, 샌드박스 환경은 스마트 컨트랙트가 제한된 영역 내에서만 작동하게 하여, 잠재적인 보안 위협을 최소화하고 네트워크 전체의 보안성을 높이며 스마트 컨트랙트 실행에 대한 일관성을 보장한다. 개발자들은 샌드박스 환경을 통해 외부의 영향을 받지 않고 스마트 컨트랙트를 테스트하고 배포할 수 있으며, 이를 통해 안전하고 신뢰성 있는 블록체인 애플리케이션 개발이 가능하다[28].

반면, EVM의 단점으로는 네트워크 혼잡도에 따른 가스 비용의 변동성이 있다. 이는 사용자에게 일시적으로 높은 트랜잭션 실행 비용을 부담하게 함으로써 서비스 제공자나 소비자에게 예기치 못한 경제적 손해를 줄 수 있다.

2.2.4 EVM 기반 블록체인 플랫폼과 금융 서비스

EVM의 등장과 함께 탈중앙화 거래소, 대출 서비스, 스테이킹 서비스, 중앙은행 디지털 화폐(CBDC) 등 다양한 금융 산업 서비스들이 출시되면서, EVM 호환 가능한 다양한 블록체인 플랫폼들이 후속 개발되었다. 이 중 이더리움을 포함한 대표적인 네 가지 EVM 호환 가능한 블록체인 플랫폼의 주요 장점과 활용 사례를 분석하였다[4,10,12,13,23].

첫째, 이더리움은 스마트 컨트랙트 개념을 처음 도입한 블록체인으로 다양한 탈중앙화 금융 서비스의 등장을 야기했다[4]. 현재 존재하는 탈중앙화 서비스는 약 60개 이상이며, 금융 분야 서비스는 33개에 달한다[24,29]. 금융 분야의 서비스에는 대출(Loan), 거래소(DEX), 그리고 스테이킹(Staking) 등으로 다양한 분류가 있다. 대출 서비스는 유동성 풀을 제공하여 사용자가 예치한 돈을 다른 사용자가 대출할 수 있는 서비스이며 대표적인 예로는 Aave가 있다[30]. 거래소 서비스는 스마트 컨트랙트를 통해 사용자 간에 토큰을 주고받는 행위인 스왑(swap) 서비스를 제공한다. Uniswap이 대표적인 이더리움 기반 탈중앙화 거래소이다[31]. 마지막으로 스테이킹 서비스는 이더리움의 지분 증명(PoS) 합의 방식과 관련이 있다. 지분 증명은 노드가 검증자가 되기 위해 일정 금액을 네트워크에 보관(스테이킹)하여 지분을 얻고, 블록 검증의 보상을 지분에 따라 얻게 되는 방식이다. 이에 여러 사용자가 소액으로 스테이킹에 참여하고 비율에 따라 보상을 돌려받을 수 있도록 매개해 주는 서비스가 스테이킹 서비스이며, Lido가 대표적인 예시이다[32].

둘째, 아발란체는 자체적으로 합의 알고리즘을 개발하여 블록 합의 시간을 1초 이내로 줄임으로써 확장성을 개선하였다. 아발란체는 퍼블릭한 메인넷과 이를 기업용으로 확장한 프라이빗 서브넷인 Evergreen을 제공한다. Evergreen은 메인넷의 보안을 상속받음과 동시에 서비스에 특화된 개인화가 가능하다. Citi은행은 개인 간 송금 시스템 구축을 위해 진행된 Project Guardian을 Evergreen 서브넷을 활용하여 PoC(Proof of Concept)를 진행하였다[33]. 뿐만 아니라 개인 투자 포트폴리오 관리 플랫폼인 Republic은 아발란체를 기용하여 새로운 수익 공유 디지털 자산인 Republic Note를 출시하였다[34].

(표 1) 블록체인 플랫폼에서 동작하는 금융 서비스
(Table 1) Financial Services by Blockchain Platform

블록체인	카테고리	금융 서비스	출시년도
이더리움	DEX	UniSwap	2018
	Loan	Aave	2020
	Staking	Lido	2020
아발란체	Profit Sharing	Republic	2021
	Tokenization	Project Guardian	2021
세이	Tokenization	CodedEstate	2022
	NFT Market	COPYCAT	2021
베수	Payment	Visa	2021
		MasterCard	2024
	CBDC	MAS	2021

셋째, 세이는 EVM에서 트랜잭션의 병렬 실행 처리를 통해 낮은 트랜잭션 확정 지연시간을 제공하는 최초의 블록체인 플랫폼이다. 이러한 세이를 활용하는 금융 서비스로는 Coded Estate과 COPYCAT 등이 있으며 각각 부동산 조각 투자 및 NFT 거래 서비스를 제공한다[35,36].

넷째, 베수는 Ethash, Clique, IBFT, QBFT의 네 가지의 합의 알고리즘을 모두 지원하고 트랜잭션 암호화 및 권한 설정 등의 보안 강화 기능을 제공하여 기업용 블록체인으로써의 강점을 보유하고 있다[13]. Visa, Mastercard와 같은 대표적 글로벌 금융 기업들이 베수를 활용하고자

하며, Visa는 금융 서비스 분야로 스테이블코인, 탈중앙화 거래소, 대출 및 저축 그리고 지불 시스템 등에 베수가 활용될 수 있다고 분석했다[37]. 각 플랫폼을 활용한 금융 서비스 사례들은 표 1과 같다.

2.3 블록체인의 성능 측정

(표 2) 성능 지표[35]
(Table 2) Performance Metrics[35]

구분	성능지표	설명
전체적 성능	TPS	초당 트랜잭션 처리량
		Transaction Per Second
	ARD	평균 트랜잭션 처리응답 대기시간
		Average Response Delay
	TPC	CPU 별 트랜잭션 처리량
		Transaction Per CPU
	TPMS	메모리초 당 트랜잭션 처리량
		Transaction Per Memory Second
	TPDI/O	디스크 입출력 별 트랜잭션 처리량
		Transaction Per Disk I/O
	TPND	네트워크 데이터별 트랜잭션 처리량
		Transaction Per Network Data
세부적 성능	PDR	초당 연결된 노드 수
		Peer Discovery Rate
	RRR	초당 RPC 응답 비율
		RPC Response Rate
	TPR	초당 전파된 트랜잭션 수
		Transaction Propagating Rate
	CET	컨트랙트 당 실행 소요 시간
		Contract Execution Time
	SUT	트랜잭션당 상태 업데이트 소요시간
		State Updating Time
	CCT	트랜잭션 당 합의 소요 시간
		Consensus-Cost Time

2.3.1 성능 지표

블록체인 플랫폼의 성능은 전체적(overall) 성능과 세부적(detailed) 성능으로 나눌 수 있다[38]. 전체적 성능이란 사용자 혹은 관리자 입장에서 지각하는 블록체인 시스템의 전반적인 성능을 의미한다. 전체적 성능을 나타내는 지표로는 대표적으로 Transaction Per Second(초당 트랜잭션 처리량, 이하 TPS)과 Average Response Delay (평균 응답 대기시간, 이하 Latency)가 있다. TPS는 블록에 삽입되고 합의까지 완료된 트랜잭션들의 초당 발생량을 측정한다. Latency는 각 트랜잭션이 전파된 시점부터 블록에 삽입되고 합의가 완료될 때까지의 시간차를 계산한다. 두 지표는 사용자 입장에서 블록체인 서비스를 사용하면서 발생시킨 트랜잭션이 신속히 처리되는지와 관련성이 있으므로, 서비스에 가장 적합한 블록체인을 선정하는 데 대표적으로 측정되는 지표이다. 본 논문에서도 위 두 지표를 동일하게 성능 측정 실험에 적용하였다.

세부적 성능은 블록체인 시스템을 구성하는 각 프로세스의 성능을 의미한다. 대표적인 지표로는 P2P 네트워크에서 초당 노드 간 연결 시도 횟수 대비 연결된 노드의 비율을 나타내는 Peer Discovery Rate(노드 탐색 비율, 이하 PDR)과 블록체인 시스템에서 제공하는 초당 RPC (Remote Procedure Call) 요청 응답률을 나타내는 RPC Response Rate(RPC 응답 비율, 이하 RRR)이 있다. 각 지표는 블록체인 시스템을 구성하는 각각의 프로세스 성능이며, 수치에 따라서 병목 현상이 발생하는 구간을 유추할 수 있다[38]. 이는 개발자에게 플랫폼 유지보수 및 성능 향상 작업에 있어 주요한 지표로 활용된다. 전체적 성능과 세부적 성능에 대한 종합지표는 표 2와 같다.

2.3.2 성능 측정 실험 환경에 대한 선행연구

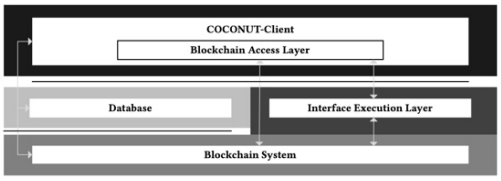
기존 선행연구에서는 다양한 블록체인 플랫폼 간의 객관적인 성능 측정을 위해서는 각 플랫폼에 대한 일관적인 실험 환경 조성의 중요성을 강조하였다[2,39].

먼저, 비트코인과 이더리움의 낮은 초당 트랜잭션 처리량(비트코인 5 TPS, 이더리움 10 TPS)[40]을 해결하기 위해 개발된 솔루션을 적용한 블록체인 플랫폼들의 성능을 비교 분석하는 연구가 진행된 바 있다[39]. 해당 연구에서는 일관적인 실험 환경을 위해 플랫폼을 솔루션 종류에 따라 3가지 Layer로 구분하고 각 Layer별 성능을 비교하였다. Layer0는 트랜잭션이 블록과 같은 데이터가 다른 노드에게 전파되는 과정을 최적화한 솔루션, Layer1은 자료구조(data-structure) 이론 기반으로 블록 처리 알고리즘을 최적화한 솔루션, 마지막으로 Layer2는 오프체인에서 트랜잭션을 실행하고 온체인에서 검증하는 오프체인 솔루션이 적용된 플랫폼들로 구분하였다.

기업용 블록체인 플랫폼 간의 성능을 측정 및 비교한 연구에서는 기업용 블록체인은 대체로 노드의 네트워크 참여가 제한되어 있는 프라이빗 블록체인이므로, 비트코인, 이더리움과 같은 퍼블릭 블록체인에 비해 스마트 컨트랙트 로직 처리 방식과 같은 내부적인 요소가 성능을 좌우한다[2]. 이러한 점에 주목하여 여러 스마트 컨트랙트 로직에 대하여 기업용 블록체인 플랫폼별 성능을 비교하고 성능 측정 지표로는 초당 트랜잭션 처리량과 트랜잭션 응답 지연 시간을 활용했다.

2.3.3 성능 측정 프레임워크

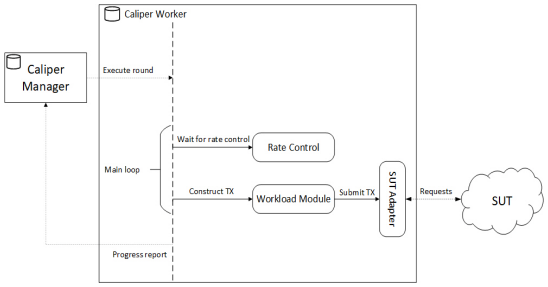
블록체인 플랫폼 성능 측정 프레임워크에는 다음 두 가지 요소가 필요하다. 첫째, 블록체인 플랫폼에 사용자가 정의한 임의의 트랜잭션을 사용해 일정한 트래픽을 발생시킬 수 있는 트래픽 생성 모듈이다. 이는 블록체인 플랫폼에 동일한 부하를 발생시켜 객관적인 성능 측정을 가능하게 한다. 뿐만 아니라 사용자 정의에 따른 트래픽 발생 정도를 조절하여 다양한 네트워크 상황을 구현할 수 있다. 둘째로, 블록체인 플랫폼과 호환가능한 API 지원 인터페이스 모듈이다. 블록체인 플랫폼에서는 개발자와 사용자를 위해 필요한 API를 제공한다. 이를 활용하여 트랜잭션 생성, 서명이 가능할 뿐 아니라 성능 측정에 필요한 데이터도 수집할 수 있다. 따라서 각 블록체인 플랫폼별 API를 지원하는 미들 인터페이스를 제공해야 하며 다양한 플랫폼의 API가 호환될수록 더욱 많은 플랫폼들에 대한 일관적인 성능 측정이 가능하다. 위 요구 사항들을 모두 만족한 대표적인 두 가지 선행 연구의 성능 측정에 활용한 프레임워크를 분석한다[15,17].



(그림 2) COCONUT 컴포넌트 아키텍처
(Figure 2) COCONUT Component Architecture

COCONUT[15]은 그림 2와 같은 아키텍쳐를 가지는 성능 측정 프레임워크이다. Blockchain Access Layer (BAL)는 여러 블록체인 플랫폼 각각에 대한 개별적인 API-Driver를 제공하는 컴포넌트이며, Interface Execution Layer(IEL)는 특히 트래픽 발생 관련 API를 제공하는 컴포넌트이다. 각 플랫폼별 API를 변경 없이 그대로 사용하여 플랫폼 자체의 고유성을 유지하였고, 이를 통해 측정된 성능의 오차를 최소화하였다. 또한 각 플랫폼에서 사용가능한 트랜잭션 설정값 등을 개별적으로 지원하여 하나의 프레임워크에서 다양한 플랫폼들에 대해 유연한 성능 측정이 가능하다는 것이 장점이다.

Database에는 블록체인 플랫폼에서 발생한 이벤트를 수신하는 구독형(subscription) 기법을 활용하여 수집한 데이터를 저장하는 컴포넌트이다. 구독형 기법은 데이터를 얻기 위한 별도의 네트워크 요청이 발생하지 않으므로 효율적이지만, 데이터가 발생한 시점과 데이터를 수신한 시점 간의 차이가 존재하게 되고 네트워크 혼잡도에 따라 그 시간 간격은 더욱 커질 수 있다. 이에 따라 올바른 측정값을 얻지 못할 수 있다는 단점이 있다.



(그림 3) Hyperledger Caliper 동작 흐름
(Figure 3) Hyperledger Caliper Workflow

다음으로, Hyperledger Caliper[16]는 Hyperledger 재단에서 제작한 블록체인 플랫폼 벤치마킹 툴이다. Hyperledger Caliper 역시 특정 블록체인 플랫폼에 의존하지 않고 다양한 플랫폼에 대해 통합된 성능 측정 및 비교가 가능하다. 그러나 COCONUT과는 달리 짧은 주기로 데이터를 요청하는 폴링(polling) 방식의 데이터 수집 기법을 사용하여 데이터가 생성된 시간과 비슷한 시점에 수집이 가능하다. 따라서 데이터의 발생 시간 오차 범위를 줄이고 보다 정확한 성능 수치를 얻을 수 있다.

그림 3은 Hyperledger Caliper 동작 흐름을 나타낸다. Caliper Manager는 중심 컴포넌트로 초기화 및 결과 수집과 생성 등의 전체 프로세스를 관리한다. Caliper Worker는 Manager와 직접 소통하며 실제 트래픽 발생과 데이터 수집 역할을 한다. 트래픽 발생 시 Rate Control 모듈에 정의된 다양한 부하 발생 패턴에 따라 발생 속도를 조절한다. 이때 Workload Module에서 생성하는 사용자 정의에 따라 임의의 로직이 포함된 트랜잭션을 사용한다. 생성된 트랜잭션은 System Under Test(SUT)로 정의되는 테스트 대상 시스템, 곧 블록체인 플랫폼에 전달되는데 SUT Adapter에서 각 블록체인 플랫폼이 제공하는 API 형태에 따라 SUT에 전달하게 된다. 이러한 동작 흐름은 실제 시나리오와 유사한 트랜잭션들을 다양한 블록체인 플랫폼에서 테스트할 수 있도록 돕는다.

그러나 공통된 환경을 보장하기 위해서 일관된 스마트 컨트랙트 구현 방식을 사용하여 이더리움에서 사용하는 Solidity[41] 언어로 제작된 스마트 컨트랙트를 활용할

수 없으며, 복잡도가 높은 시나리오에 대해서는 프레임워크에 사용가능한 구성 개발에 대한 추가적 비용이 발생하게 되어, 요구 기능에 대한 유연한 테스트가 어렵고 단순한 테스트 이외에 활용도가 떨어진다. COCONUT과 Hyperledger Caliper의 공통점 및 차이점은 표 3과 같다.

(표 3) 성능 측정 프레임워크 별 특징
(Table 3) Benchmarks Features

항목		COCONUT	Hyper-Ledger Caliper
공통점		멀티 플랫폼 지원 가능	
차이점	데이터 수집 방식	데이터 구독형	주기적 요청
	트랜잭션 생성 방식	플랫폼 별 제공 API 활용	자체적 메커니즘 활용
	플랫폼 환경 설정 방식	플랫폼 별 환경변수 제공	자체적 통합 환경변수 제공

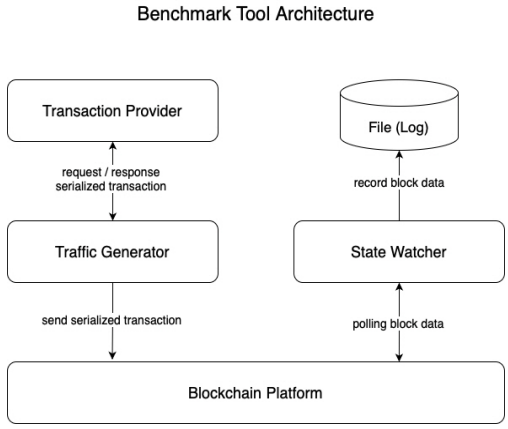
3. 제안하는 성능 측정 프레임워크

3.1 아키텍처

본 장에서는 기존 프레임워크들의 장점을 채택하고 단점을 보완한 성능 측정 프레임워크를 제안한다. 트래픽 생성 모듈은 COCONUT[15]과 유사하게 트랜잭션 트래픽 발생기와 트랜잭션 생성기로 구성되며, 각 블록체인 플랫폼에서 제공하는 API를 활용하여 다양한 트랜잭션 발행 및 다수의 블록체인 플랫폼에 대응할 수 있는 구조이다. 데이터 수집 모듈은 Hyperledger Caliper[16]의 폴링(polling) 기반의 주기적 데이터 요청을 통한 수집 방법을 사용하여 데이터 생성 시간과 수집 시간에 대한 오차율을 줄였다. 또한 Solidity를 이용한 스마트 컨트랙트 구현을 지원하여 복잡도가 높은 시나리오에 대한 개발 비용을 절감하고 유연한 테스트가 가능하게 하였다.

성능 측정 프레임워크는 그림 4와 같은 아키텍처로 구성된다. 트래픽 생성 모듈은 Traffic Generator와 Transaction Provider로 구성된다. Traffic Generator는 최초 설정된 값에 따라 트랜잭션 트래픽을 발생시킨다. Transaction Provider는 Solidity로 작성된 스마트 컨트랙트를 기반으로 트랜잭션을 생성한다. State Watcher는 데이터 수집 모듈에 해당한다. State Watcher는 블록체인 플랫폼에서 제공하는 API를 0.5초 주기로 호출하여, 처리한 트랜잭션과 블록 데이터를 가져와 파일에 저장한다. 주기는 각 블록체인 플랫폼의 블록 생성 주기가 평균적으로 1초인 것을 감안하여 0.5초로 설정하였다[42].

요약하면, 본 프레임워크는 첫째로 플랫폼별 API를 지원하여 다양한 플랫폼에 대한 확장성과 유연성을 가진다. 둘째로, 폴링기반의 주기적 데이터 요청 방식을 활용하여 측정 오차율을 낮게 유지한다. 마지막으로, Solidity 기반의 스마트 컨트랙트를 지원하여 복잡한 서비스 시나리오를 트랜잭션으로 구현, 테스트할 수 있도록 한다.



(그림 4) 성능 측정 프레임워크 아키텍처
(Figure 4) Benchmark Tool Architecture

3.2 성능 측정 방안 및 절차

3.2.1 실험을 위한 성능 측정 프레임워크

본 장에서는 블록체인 플랫폼의 전체적 성능을 나타내는 대표적인 지표인 TPS(초당 트랜잭션 처리량)와 Latency(평균 응답 대기시간)를 정의하고, 본 프레임워크에서 지표를 측정하는 방식을 설명한다.

먼저, TPS는 식 (1)을 사용하여 구할 수 있다. 트랜잭션 개수(N)만큼 트랜잭션을 전송했을 때, N을 최초 트랜잭션 전송 시간(T_{st})부터 마지막 트랜잭션 처리 완료되기까지 걸린 시간(T_{end})의 차로 나눈 값이다. 다음, Latency는 식 (2)를 사용하여 구할 수 있다. 트랜잭션 N개를 전송하였을 때, 각 트랜잭션이 전송된 시간($T_{k_{sent}}$)과 처리 완료된 시간($T_{k_{confirm}}$)의 차이를 모두 더한 값의 평균 값이다.

$$TPS = N \div (T_{end} - T_{st}) \quad (1)$$

$$Latency = \sum_{k=1}^{N} (T_{k_{confirm}} - T_{k_{sent}}) \div N \quad (2)$$

본 프레임워크는 다음 과정에 따라 각 블록체인 플랫폼별로 TPS와 Latency를 측정한다.

1) 트랜잭션 유형에 따라 정해진 트래픽 발생량으로 트랜잭션을 발생시켜 총 N개의 트랜잭션을 전송한다. 전송 시작 시간(T_{st})을 기록한다. 또한 각 트랜잭션의 전송된 시점의 시간($T_{k_{sent}}$)을 모두 기록한다.
2) 1)과 동시에 0.5초 주기로 데이터를 요청하기 시작하여 처리된 트랜잭션을 확인한다. 각 트랜잭션이 처리 완료된 시간($T_{k_{confirm}}$)을 기록한다.
3) 전송된 모든 트랜잭션이 처리되기까지 2)를 지속하며, 블록에 처리된 트랜잭션 개수가 발생시킨 트랜잭션 개수와 같은 경우 종료 시간(T_{end})을 기록하고 중단한다.
4) 기록된 데이터를 기반으로 TPS와 Latency를 산출한다.

4. 결과 분석실험

4.1 실험 환경 및 트랜잭션 유형

본 실험에서는 총 세 가지 블록체인 플랫폼(아발란체, 세이, 베수)을 대상으로 성능 측정 실험을 진행하였다. 먼저, 블록체인 노드 개수는 모든 플랫폼에 대해 비잔틴 장애 허용(Byzantine Fault Tolerance) 네트워크의 최소 노드 구성 개수인 4개[43]로 고정하였다. 이는 노드 수가 많아질수록 성능이 저하되기 때문이다[15]. 또한 트랜잭션을 요청하는 클라이언트 수와 초당 전송하는 트랜잭션 수를 고정하여 플랫폼에 발생하는 트래픽을 동일하게 하였다. 이와 같이 블록체인 플랫폼 간에 트랜잭션 처리 성능에 영향을 줄 수 있는 변수들은 모두 동일하게 설정하였으며, 상세 실험 환경은 표 4와 같다.

각 블록체인 노드의 실행 클라이언트는 각 Github 레포지토리에서 2024년 6월 30일 기준으로 마지막 업데이트된 버전을 활용하였다. 구체적으로, 아발란체는 v1.11.9, 세이는 v5.6.0, 베수는 v24.6.0을 사용하였다.

실험에서 사용한 트랜잭션은 2가지로 토큰 전송 트랜잭션과 금융 거래 트랜잭션이 있다. 토큰 전송 트랜잭션은 금융 서비스에서 가장 기본적이면서도 빈번하게 사용되는 트랜잭션으로, 전반적인 금융 서비스 트랜잭션 처리 성능을 측정할 수 있다. 구체적인 구현에는 Open Zeppelin에서 제공하는 ERC20 구현체의 *transfer* 함수를 사용했다. 해당 함수는 자신의 계정이 소유한 일정 수량의 토큰을 다른 계정으로 전송한다. 금융 거래 트랜잭션은 금융 서비스에서 실제로 사용하는 기능 중에서도 실행 복잡도가 높은 트랜잭션을 일컫는다. 본 실험에서는 실제 배포된 탈중앙화 거래소 Dexalot의 기능을 채택하였는데, 오더북(OrderBook) 컨트랙트의 함수 중 주문을 요청하는 *addOrder* 함수를 사용하였다. *addOrder*는 사용자의 주문을 기록해 두었다가, 미리 기록되어 있는 주문과 매칭하여 금융 거래를 성사시킨다.

(표 4) 하드웨어 성능 및 실험 설정
(Table 4) H/W Specification & Environment Settings

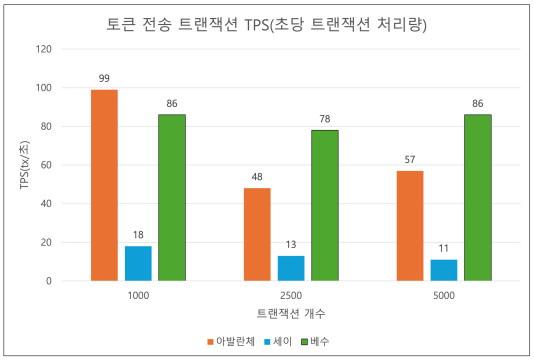
항목		데이터	
HW	CPU	AMD Ryzen 9 3900X 12-Core Processor	
	RAM	64 GB	
	OS	Linux/Ubuntu 22.04	
환경 설정	노드 수	4 (개)	
	계정 수	5000 (개)	
	초당 트랜잭션 발생량	토큰 전송	100 (tx/초)
		금융 거래	20 (tx/초)
기록하는 데이터		트랜잭션 전송 시작 시간	
		모든 트랜잭션 처리완료 시간	
		각 트랜잭션의 전송 시작 시간	
		각 트랜잭션의 처리 완료 시간	

발생하는 트랜잭션 개수 N은 각 트랜잭션 유형 별로 점점 증가하는 값 3개로 설정하였다. 토큰 전송 트랜잭션의 경우 N은 1000, 2500, 그리고 5000이며, 금융 거래 트랜잭션의 경우 N은 100, 500, 그리고 1000이다. 금융 거래 트랜잭션이 토큰 전송 트랜잭션보다 낮게 설정된 이유는 금융 거래 트랜잭션의 가스 비용이 상대적으로 높아 블록에 담기는 트랜잭션의 개수가 더 적으므로, 트랜잭션 처리 속도가 더 낮기 때문이다.

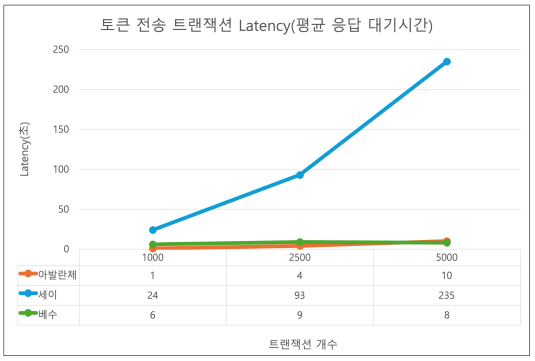
4.2 실험 결과

4.2.1 토큰 전송 트랜잭션 처리 성능 측정 결과

그림 5와 그림 6은 각각 토큰 전송 트랜잭션 처리에 대한 TPS와 Latency를 측정한 결과를 나타낸다.



(그림 5) 토큰 전송 트랜잭션 TPS 비교
(Figure 5) Comparison of TPS by Token Transfer



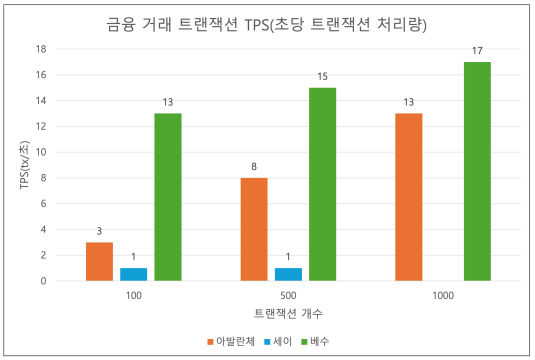
(그림 6) 토큰 전송 트랜잭션 Latency 비교
(Figure 6) Comparison of Latency by Token Transfer

토큰 전송 트랜잭션의 경우 전체적으로 트랜잭션 개수가 많아질수록 TPS가 낮아지는 경향을 보였다. 이것은 처리되는 트랜잭션 개수 대비 전송되는 개수가 더 많으므로 네트워크 혼잡도가 증가하기 때문이다. 아발란체는 다른 플랫폼에 비해 가장 낮은 트랜잭션 개수에서 가장 높은 TPS 수치를 보이므로, 네트워크가 여유로울 때는 트랜잭션 처리 성능이 가장 좋다고 볼 수 있다. 그러나 네트워크가 혼잡할 때는 성능이 낮아, 높은 트래픽의 트랜잭션을 효율적으로 처리하지 못한다. 세이의 경우에는 전체적으로 TPS 수치가 낮으며, 네트워크가 혼잡해짐에 따라 성능이 더욱 낮아졌다. 베수의 경우 각 트랜잭션 개수가 늘어남에도 TPS 수치가 거의 일정하므로 다른 플랫폼에 비해 트랜잭션 처리가 안정적이라 할 수 있다.

그림 6에서는 전체적으로 네트워크 혼잡도가 올라갈수록 Latency가 높아지는 경향이 있음을 알 수 있다. 이때, 아발란체는 트랜잭션 개수가 1000, 2500개일 때 다른 플랫폼 대비 가장 낮은 Latency를 기록했으며, 5000개 인 경우에도 가장 낮은 베수와 2초 차이로 비슷하였다. 반면, 베수는 트래픽이 낮은 환경에서는 아발란체보다 2에서 6배 높은 Latency를 보였으나, 복잡한 네트워크 환경에서는 오히려 낮은 수치를 보이므로 성능이 안정적이라고 할 수 있다. 세이는 모든 경우에서 가장 높은 Latency를 기록할 뿐 아니라, 네트워크가 혼잡해질수록 매우 큰 폭으로 증가하기 때문에 트랜잭션 처리 능력이 다른 플랫폼에 비해 다소 떨어지는 것으로 보인다.

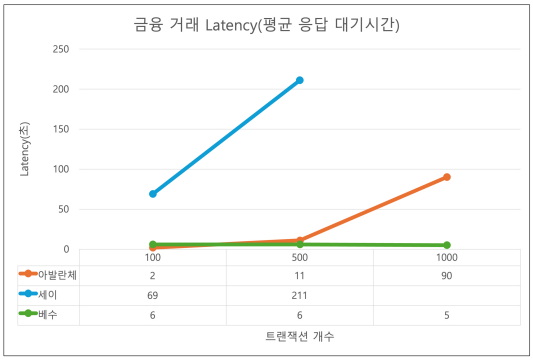
4.2.2 금융 거래 트랜잭션 처리 성능 측정 결과

그림 7과 그림 8에서는 각각 금융 거래 트랜잭션 처리에 대한 TPS와 Latency를 측정한 결과를 보여준다.



(그림 7) 금융 거래 트랜잭션 TPS 비교
(Figure 7) Comparison of TPS by Financial Transaction

그림 7에서 금융 거래 트랜잭션은 네트워크 혼잡도가 높아질수록 TPS가 증가하는 양상을 보이고 있다. 이는, 네트워크 혼잡도가 비교적 낮은 환경에서 측정되어, 트랜잭션 개수에 비례하여 안정적으로 TPS도 증가한 것으로 보인다. 아발란체와 베수가 트랜잭션 개수가 증가함에 따라 TPS가 증가했으며, 특히 베수는 각 환경에서 가장 높은 TPS 수치를 기록하였다. 반면, 세이는 1 TPS로 매우 낮은 수치를 기록하였고, 1000개 금융 거래 트랜잭션 발생 시 부분적으로 실패하여 측정이 불가능하였다. 세이는 다른 플랫폼 대비 복잡도가 높은 금융 거래 트랜잭션을 처리하는 데에는 적합성이 떨어진다고 볼 수 있다.



(그림 8) 금융 거래 트랜잭션 Latency 비교
(Figure 8) Comparison of Latency by Financial Transaction

그림 8에서 아발란체의 경우 1000개 트랜잭션 실험 시 8배 이상의 높은 폭으로 Latency가 상승하였다. 이는 TPS와 달리 Latency는 금융 거래 트랜잭션에 대해 네트워크가 혼잡해짐에 따라 응답속도가 크게 느려질 수 있다는 것을 암시한다. 세이는 트랜잭션 100개, 500개에서 다른 플랫폼 대비 10배 이상의 Latency를 보였고, 개수가 많아지면서 Latency도 3배가량 상승하였다. 반면, 베수의 경우 각 환경에서 가장 높은 TPS를 기록하면서, Latency는 매우 낮고 안정적인 수치를 기록하고 있다. 이는 베수의 금융 거래 트랜잭션 처리 능력이 다른 플랫폼 대비 동일한 환경에서 뛰어난 것으로 볼 수 있다.

(표 5) 성능 측정 실험 결과
(Table 5) Results of Performance Evaluation

트랜잭션 유형	발행 개수	성능 지표	아발란체	세이	베수
토큰 전송	1000	TPS	99	18	86
		Latency	1	24	6
	2500	TPS	48	13	78
		Latency	4	93	9
	5000	TPS	57	11	86
		Latency	10	235	8
금융 거래	100	TPS	3	1	13
		Latency	2	69	6
	500	TPS	8	1	15
		Latency	11	211	6
	1000	TPS	13	-	17
		Latency	90	-	5

표 5는 트랜잭션 유형과 발행 개수별 TPS와 Latency 측정값을 보여준다. 실험의 결과를 미루어보아 아발란체는 낮은 트랜잭션 개수에서 다른 플랫폼 대비 대부분 뛰어난 성능을 보였다. 하지만 트랜잭션 개수가 높아지면서 네트워크가 혼잡해질수록 처리 능력이 다소 감소했다. 세이는 각 실험에서 성능이 가장 낮았으며 금융 거래 트랜잭션의 높은 트래픽에서는 트랜잭션을 일부만 처리했다. 마지막으로 베수는 트랜잭션 처리 능력이 다른 플랫폼 대비 가장 안정적인 것으로 볼 수 있다.

5. 결 론

블록체인 플랫폼은 이더리움의 EVM을 통해서 탈중앙화 환경에서의 서비스들이 가능하며 금융 산업 분야에서도 활발하게 사용되고 있다. 하지만 블록체인 플랫폼의 낮은 트랜잭션 처리 속도로 인한 확장성 문제가 존재하여, 이를 해결하고자 다양한 블록체인 플랫폼들이 등장하였다. 탈중앙화 서비스 시장에서 EVM 기반 서비스 점유율이 매우 높아, 많은 블록체인 플랫폼들이 EVM 호환 가능한 환경을 제공하고 있는 반면, 이러한 EVM 기반 블록체인 플랫폼들의 성능에 대한 측정 및 비교한 사례 연구가 매우 드물어 플랫폼 선정에 어려움이 있었다.

따라서, 본 논문에서는 금융 산업에서 EVM 블록체인 플랫폼들 중 대표적인 3가지를 선정하여 성능을 측정하고 비교하였다. 또한 기존 연구들의 한계를 극복한 새로운 성능 측정 프레임워크를 제안하였다. 측정 결과 아발란체는 네트워크의 여유가 있을 때 가장 높은 성능 수치를 기록했으나, 네트워크가 크게 혼잡할수록 성능이 감소했다. 베수는 각 네트워크 상황에서 일정한 수치를 기록하여 안정적인 성능을 보였다. 세이는 다른 플랫폼 대비 가장 낮은 수치를 기록하였다. 이에 따라 금융 산업 서비스에 적합한 블록체인 플랫폼 선정에 필요한 연구 결과로 활용될 수 있을 것으로 기대된다.

반면 본 논문은 주요한 토큰 전송과 금융 거래 두 가지 트랜잭션 유형에만 국한되어, 대출 또는 스왑 등 다양한 금융 산업 서비스 트랜잭션에 대한 성능 비교가 추가될 필요가 있다. 따라서 본 논문을 확장하여 금융 산업에 각 서비스들의 대표적인 트랜잭션들을 선정하고, 다양한 성능 측정 기법을 통해 블록체인 플랫폼 선정을 보다 용이하게 할 수 있을 것이다.

참고문헌(Reference)

[1] Nakamoto, S., “Bitcoin: A Peer-to-Peer Electronic Cash System,” Bitcoin.org, 2008.

https://bitcoin.org /bitcoin.pdf

[2] Kuzlu, M., Pipattanasomporn, M., and Rahman, S., "Performance evaluation of permissioned blockchain platforms," IEEE Transactions on Engineering Management, Vol. 68, No. 1, pp. 99-112, 2021. https://doi.org/10.1109/CSDE50874.2020.9411380

[3] Buterin, V., "Ethereum Whitepaper," Ethereum.org, 2014. https://static.peng37.com/ethereum_white-paper_laptop_3.pdf

[4] Tapscott, D., Tapscott, A., and Kirkland, R., "Blockchain and its coming impact on financial services," Journal of Corporate Accounting & Finance, Vol. 31, No. 1, pp. 53-62, 2020. https:// doi.org/10.1002/jcaf.22179

[5] Kumar, M., Mohr, J., and Kumar, S., "Blockchain-enabled supply chain: An experimental study," Computers & Industrial Engineering, Vol. 136, pp. 206-214, 2019. https://doi.org/10.1016 /j.cie.2019.07.026

[6] Mettler, M., "Lightweight blockchain for healthcare," IEEE Access, Vol. 7, pp. 66519-66529, 2019. https://doi.org/10.1109/ACCESS.2019.2947613

[7] Pang, C., Wang, Z., and Wang, Y., "When energy trading meets blockchain in electrical power system: The state of the art," Applied Sciences, Vol. 9, No. 8, pp. 1561, 2019. https://doi.org/10.3390/app9081561

[8] Fortune Business Insights, "Blockchain Market Size, Share & COVID-19 Impact Analysis," Fortune Business Insights, 2021. https://www.fortunebusinessinsights.com/ko/industry-reports/blockchain-market -100072

[9] Fan, C., Ghaemi, S., Khazaei, H., Rashidi, B., and Xia, H., "A survey of blockchain from the perspectives of applications, challenges, and opportunities," IEEE Communications Surveys & Tutorials, Vol. 22, No. 2, pp. 1021-1045, 2020. https://doi.org/10.1109/ACCESS.2019.2936094

[10] Avalanche, "Avalanche Platform Whitepaper," Avalanche.org, 2020. https://cdn.prod.website-files.com/5d80307810123f5ffbb34d6e/6008d7bbf8b10d1eb01e7e16_Avalanche%20Platform%20Whitepaper.pdf

[11] Avalanche, "Avalanche Consensus Whitepaper," 2020. https://cdn.prod.website-files.com/5d80307810123f5ffbb34d6e/6009805681b416f34dcae012_Avalanche%20Consensus%20Whitepaper.pdf

[12] Sei Protocol, "Sei Whitepaper," GitHub, 2021. https://github.com/sei-protocol/sei-chain/blob/3c9576fee3494ce039df684624f918dd8066ba3f/whitepaper/Sei_Whitepaper.pdf

[13] Hyperledger, "Hyperledger Overview," Hyperledger.org, 2021. https://8112310.fs1.hubspotusercontent-na1.net/hubfs/8112310/Hyperledger/Offers/HL_Paper_HyperledgerOverview_102721.pdf

[14] Hyperledger, "Why Hyperledger Besu is a Top Choice for Financial Use Cases," Hyperledger.org, 2021. https://www.hyperledger.org/blog/why-hyperledger-besu-is-a-top-choice-for-financial-use-cases

[15] Androulaki, E., Barger, A., Bortnikov, V., Cachin, C., Christidis, K., De Caro, A., Enyeart, D., Ferris, C., Laventman, G., Manevich, Y., Muralidharan, S., Murthy, C., Nguyen, B., Sethi, M., Singh, G., Smith, K., Sorniotti, A., Stathakopoulou, C., Vukoli₩, M., and Yellick, J., "An End-to-End Performance Comparison of Seven Permissioned Blockchain Systems," ACM Transactions on Privacy and Security, Vol. 26, No. 2, Article 7, 2023. https://doi.org/10.1145/3590140.3629106

[17] Mettler, M., "A vademecum on blockchain technologies: When, which, and how," IEEE Access, Vol. 7, pp. 9198-9214, 2019. https://doi.org/10.1109/COMST.2019.2928178

[18] alg0r1thm, "Turing completeness and Bitcoin," Velog, 2023. https://velog.io/@alg0r1thm/Blockchain-turing-completeness-and-pheano-axiomatics

[19] Alan M. Turing, "On Computable Numbers, with an Application to the Entscheidungsproblem," Proceedings of the London Mathematical Society, Vol. 42, No. 1, pp. 230-265, 1936. https://doi.org/10.1112/plms/s2-42.1.230

[20] Decenter News Team, "What Turing completeness

means for blockchains," Decenter, 2023. https://www.decenter.kr/NewsView/1VE1PVFSZQ/GZ05

[21] Nick Szabo, "The Idea of Smart Contracts," Nick Szabo's Papers and Concise Tutorials, 1997. https://web.archive.org/web/20140406003401/szabo.best.vwh.net/idea.html

[22] Anderson, J., "Smart Contract Development with Blockchain Technology," Medium, 2021. https://kasata.medium.com/smart-contract-development-with-blockchain-technology-c9d12076f490

[23] Chen, T., Li, Z., Luo, X., and Zhang, X., "DefectChecker: Automated Smart Contract Defect Detection by Analyzing EVM Bytecode," IEEE Transactions on Software Engineering, Vol. 47, No. 11, pp. 2386-2403, 2021. https://doi.org/10.1109/TSE.2021.3054928

[24] Ethereum, "Ethereum Virtual Machine (EVM) Overview," Ethereum.org, 2021. https://ethereum.org/en/developers/docs/evm/

[25] Takenobu Hosomi, "Illustrated Ethereum Virtual Machine," 2020. https://takenobu-hs.github.io/downloads/ethereum_evm_illustrated.pdf

[26] Gavin Wood, "Ethereum: A Secure Decentralised Generalised Transaction Ledger," Ethereum Yellow Paper, 2014. https://ethereum.github.io/yellowpaper/paper.pdf

[27] Andreas M. Antonopoulos and Gavin Wood, "Mastering Ethereum: Building Smart Contracts and DApps," 2018.

[28] Yoichi Hirai, "Defining the Ethereum Virtual Machine for Interactive Theorem Provers," IACR Cryptology ePrint Archive, 2017. https://doi.org/10.1007/978-3-319-70278-0_33

[29] Ethereum, "Decentralized Applications (DApps) in Finance," Ethereum.org, 2021. https://ethereum.org/en/dapps/?category=finance

[30] Aave, "Aave V3 Technical Paper," GitHub, 2021. https://github.com/aave/aave-v3-core/blob/master/techpaper/Aave_V3_Technical_Paper.pdf

[31] Uniswap, "Uniswap v3 Whitepaper," Uniswap.org, 2021. https://uniswap.org/whitepaper-v3.pdf

[32] Lido, "Lido Ethereum Liquid Staking," Lido.fi, 2021. https://lido.fi/static/Lido:Ethereum-Liquid-Staking.pdf

[33] Avalanche, "Citi FX Solution on Avalanche," Avalanche.org, 2021. https://www.avax.network/blog/citi-fx-solution-avalanche

[34] Avalanche, "Republic Selects Avalanche for its Profit-Sharing Digital Asset," Avalanche.org, 2021. https://www.avax.network/blog/republic-selects-avalanche-for-its-profit-sharing-digital-asset

[35] Sei Protocol, "Coded Estate Leverages Sei to Revolutionize Real Estate and Rental Industry on Blockchain," Sei.io, 2021. https://blog.sei.io/coded-estate-leverages-sei-to-revolutionize-real-estate-and-rental-industry-on-blockchain/

[36] Copycat Finance, "Copycat Finance," Copycat.finance, 2021. https://copycat.finance/

[37] Visa, "Enterprise Blockchain Solutions," Visa.com, 2021. https://usa.visa.com/solutions/crypto/enterprise-blockchain.html

[38] Dinh, T. T. A., Liu, R., Zhang, M., Chen, G., Ooi, B. C., and Wang, J., "A detailed and real-time performance monitoring framework for blockchain systems," Proceedings of the 2018 International Conference on Management of Data, pp. 1375-1390, 2018. https://doi.org/10.1186/s13677-024-00682-0

[39] Raj, S., Chatterjee, S., and Mukhopadhyay, D., "Performance-based analysis of blockchain scalability metric," Economic Review, Vol. 72, No. 2, pp. 59-73, 2020. https://doi.org/10.31803/tg-20210205103310

[40] Binance Academy, "Transactions Per Second (TPS)," Binance Academy, 2023. https://academy.binance.com/en/glossary/transactions-per-second-tps

[41] Solidity Team, "Solidity Documentation," Solidity, 2023. https://docs.soliditylang.org/en/v0.8.11/index.html

[42] ChainSpect, "ChainSpect Dashboard," ChainSpect, 2023. https://chainspect.app/dashboard/

[43] Miguel Castro and Barbara Liskov, “Practical Byzantine Fault Tolerance,” OSDI, Vol. 99, No. 1999, 1999.
https://dl.acm.org/doi/10.5555/296806.296824

◐ 저 자 소 개 ◑

모 상 일(Sang Il Mo)
2022년 광운대학교 소프트웨어학과(공학사)
2024년 한양대학교 대학원 블록체인융합학과(공학석사)
2024년~현재 ㈜페어스퀘어랩 선임연구원
관심분야 : 분산시스템, 합의알고리즘, 블록체인 etc.
E-mail : sangil.mo@fairsquarelab.com

이 재 준(Jae Jun Lee)
2021년 한국과학기술원(KAIST) 생명학과(이학사)
2023년 한국과학기술원(KAIST) 대학원 전산학과(공학석사)
2023년~현재 ㈜페어스퀘어랩 선임연구원
관심분야 : 데이터베이스, 블록체인 etc.
E-mail : jaejun.lee@fairsquarelab.com

서 병 완(Byung Wan Suh)
1993년 미국 일리노이대학교 Information & Decision Science (공학사)
1995년 미국 조지워싱턴대학교 대학원 Information System (공학석사)
2013년 서울종합과학대학원 경영학과 (경영학박사)
2017년~현재 산업정책연구원 연구교수
2023년~현재 ㈜페어스퀘어랩 기업부설연구소장
관심분야 : 정보시스템, IT통합, 디지털전환, 블록체인 etc.
E-mail : byungwan.suh@gmail.com