

Fleta

체인 성능 측정 리포트

Innovations for a Better Life

Fleta.io



1 체인이 수용할 수 있는 부하 및 속도의 일관성 유지 부분

체인이 얼마나 안정적이고 일관성 있게 성능을 낼 수 있는 것에 대한 한계치를 확인하고 이를 테스트하는 것은 매우 중요합니다.

단순하게 순간적인 처리 속도를 높인 결과는 실제 체인의 서비스와 운영을 담보하지 못합니다. 이를 위해서는 일관성 있는 처리 능력이 어디까지 인지 확인하는 부분이 필요합니다.

Tx가 발생하여 생성되고 전파되고 완결되는 과정에는 여러가지 상황들과 다양한 요소들이 개입됩니다. 때문에 체인의 순수한 처리 속도만을 확인하는 목적만을 위한 측정 방식이 필요합니다. 그래서 이에 대한 측정 방식을 만들어 측정을 진행하였습니다.

2 실제적인 운영 환경에서의 체인 성능과 안전성 부분

상당수 많은 프로젝트가 체인의 테스트를 내부 실험실 테스트를 진행하는데 내부 실험실 테스트만으로는 실질적인 체인의 성능을 대표하기는 어렵습니다.

특히나 체인은 다른 프로젝트나 타 서비스에도 사용함을 목표로 개발되는 것이기 때문에 실제적인 운영 환경에서의 테스트결과가 중요합니다.

FLETA는 최대한 객관적인 측정 결과를 만들기 위해서 실제 운영환경을 최대한 구현하여 테스트를 진행하였습니다.

테스트 환경

FLETA는 블록생성 노드가 전 세계에 퍼져 있어도 잘 동작할 수 있도록 설계 되었으며 이를 위하여 6곳의 서로 다른 지역에 클라우드로 서버를 두고 테스트를 진행하였습니다.

1 서버 사양

Vultr High Frequency 8 Core (3.8 GHz) / 32 GB RAM / 512 GB NVMe

2 서버 네트워크 구성

테스트 서버의 모든 세팅은 메인넷과 동일한 환경으로 구성되어 있습니다. 모든 옵저버는 Private Network를 통해 TCP로 통신하고 포물레이터 (블록생성노드)와 옵저버는 Websocket을 통해 통신합니다. 그리고 포물레이터 (블록생성 노드) 노드는 TCP로 일반적인 통신합니다. 메인넷의 구성과 마찬가지로 클라우드 VM을 생성 후 바로 설치를 진행하였으며 VM에 대한 sysctl을 통한 네트워크 튜닝 등은 진행하지 않고 클라우드에서 제공되는 기본 값을 사용하였습니다.

3 서버 지역 분포 - 일관성 테스트

지역	서버 구성
US, New Jersey	옵저버 5 / 블록생성노드 3
US, Miami	블록생성노드 3
US, Dallas	블록생성노드 3
DE, Frankfurt	블록생성노드 3
FR, Paris	블록생성노드 3
UK, London	블록생성노드 3

4 서버 지역 분포 - 실제 환경 테스트

지역	서버 구성
US, New Jersey	옵저버 5 / 블록생성노드 2 / Tx생성기 1
US, Miami	블록생성노드 2 / Tx생성기 1
US, Dallas	블록생성노드 2 / Tx생성기 1
DE, Frankfurt	블록생성노드 2 / Tx생성기 1
FR, Paris	블록생성노드 2 / Tx생성기 1
UK, London	블록생성노드 2 / Tx생성기 1

측정 시나리오

1 공통 시나리오

성능 테스트에 사용되는 코드는 Genesis 설정 등 체인 기본 설정 값을 제외한 모든 부분은 메인넷과 동일합니다.

테스트에서는

- 1 트랜잭션 전파 단계에서의 서명 검사 및 트랜잭션 검증하는 것,
- 2 블록 생성 단계에서 트랜잭션 검증 및 Context에 반영하는 것,
- 3 블록 검증 단계에서 옵저버 서명 검사,
- 4 생성자 서명 검사,
- 5 생성 순서 검사,
- 6 트랜잭션 서명 검사,
- 7 트랜잭션 검증,
- 8 Context에 반영,
- 9 변경에 대해 Context Hash값 비교를 통한 바이트 수준의 완전 일치 검사,
- 10 fsync=on 상태에서 체인 데이터 로컬에 저장

등을 포함한 모든 사항을 진행합니다.

2 부하 및 속도 일관성 측정

부하 및 속도의 일관성 측정은 체인의 지속적인 최대 트랜잭션 처리 속도를 확인하기 위한 테스트입니다.

따라서 서명이 완료된 트랜잭션을 포물레이터가 블록 생성 단계에서 직접 블록에 지정된 개수를 포함시켜서 블록을 생성합니다. 이를 통해 트랜잭션이 블록 생성자인 포물레이터에 오는 과정을 제외하고 순수하게 트랜잭션을 블록에 포함시켜서 이를 컨센서스를 통해 합의하고 전파하여 연결하는 부분에 대한 최대 성능을 확인할 수 있습니다.

3 실제적인 운영 환경내에 체인 성능 측정

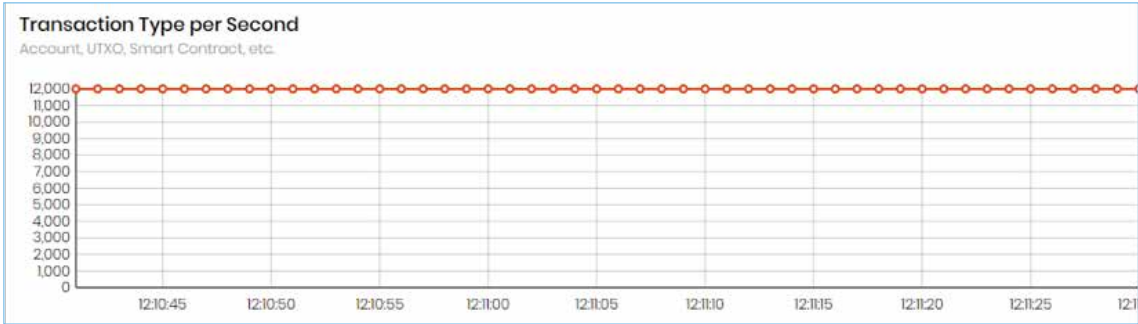
실제적인 운영 환경내에 체인 성능은 거래를 생성해서 브로드캐스트로 거래를 모든 노드에 전파하는 방식을 사용합니다. 이는 실제 체인이 구동되는 환경과 동일한 환경을 의미하며, 포물레이터가 아닌 일반 노드가 실제 트랜잭션 전파하고 받은 노드가 이를 검증하고 트랜잭션 풀에 저장한 후 다시 전파를 하는 과정을 거쳐 블록에 포함되는 실제 운영 환경과 동일한 방식을 이용하여 테스트를 진행합니다.

테스트 결과

최대 12,000 TPS

1 부하 및 속도 일관성 측정

체인의 성능을 테스트할 때 일반 노드로서 동작하는 블록 익스플로러를 실행시켜 거래 정보를 획득하고 체인 성능 그래프를 확인하였습니다. 생성된 블록을 600개 단위로 정보를 확인하여 해당 구간 생성에 걸린 시간을 통해 블록의 평균 생성 간격과 평균 트랜잭션 처리 수를 측정하였습니다.



거래 생성 그래프는 두 블록이 한 점으로 합쳐져서 1초에 한 점씩 찍히도록 되어 있습니다. 해당 그래프에서 12,000 TPS가 안정적으로 나오는 것을 확인할 수 있습니다.

구간(높이)	블록수	구간거래수	구간시작	구간종료	소요시간	TPS
1~600	600	3,600,000	12:07:21	12:12:36	315	11,428
601~1200	600	3,600,000	12:12:36	12:12:36	300	12,000
1201~1800	600	3,600,000	12:17:36	12:22:36	300	12,000
1801~2400	600	3,600,000	12:22:36	12:22:36	300	12,000
2401~3000	600	3,600,000	12:27:36	12:27:36	300	12,000

트랜잭션 처리를 구간별로 정리한 표를 확인해보면 첫 구간에서 포물레이터의 시작 속도가 다소 차이가 나기 때문에 약간의 지연이 발생하였지만, 이후 블록이 동기화되고 안정적으로 블록이 생성되었습니다. 이에 두번째 구간부터는 블록 지연이 발생하지 않았으며 안정적으로 트랜잭션을 처리하고 있습니다.

측정 내역	의미	결과(평균)
블록 패킷 압축 해제 시간	gzip으로 압축된 패킷 압축 해제	20ms
블록 Deserialization 시간	블록 패킷을 블록 메시지로 변환	20ms
트랜잭션 서명 검증 시간	블록 내 모든 트랜잭션 서명 검증	250ms
Context 생성 시간	트랜잭션을 검증하고 처리	70ms
블록 디스크 저장 시간	블록을 디스크에 저장(fsync=on)	60ms
블록 검사 합의 시간	옵저버가 블록을 검사하고 합의	30ms

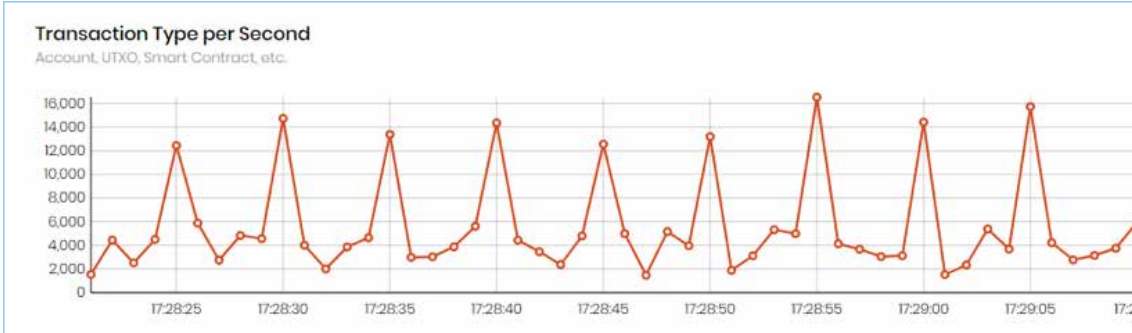
체인의 성능에 큰 영향을 주는 주요 작업의 시간을 측정한 결과를 보면 한 블록을 처리하는데 필요한 평균 시간은 블록 패킷 압축 해제부터 블록 디스크 저장 시간까지 420ms로 나타났으며, 옵저버가 합의하는 시간까지 포함하면 450ms로 나타났습니다. 이를 통해 한 블록 당 500ms 안으로 처리하게 되어 1초에 2 블록을 처리할 수 있음을 알 수 있습니다. 블록을 처리함에 있어서 네트워크 지연에 대한 영향을 최소로 할 수 있도록 병렬적인 처리 구조를 도입하여 옵저버와 블록 생성 노드와의 네트워크 지연이 100ms 정도라 할지라도 500ms에서 처리 시간을 제외한 50ms 정도의 여유로도 끊임 없이 처리할 수 있습니다.

테스트 결과

최대 12,000 TPS
평균 7,000 TPS

2 실제적인 운영 환경 내에 체인 성능 측정

체인의 성능을 테스트할 때 일반 노드로서 동작하는 블록 익스플로러를 실행시켜 거래 정보를 획득하고 체인 성능 그래프를 확인하였습니다. 생성된 블록을 600개 단위로 정보를 확인하여 구간 별 블록 지연을 확인하였으며, 평균 TPS는 [전체 거래수 / 블록 수 * 지연율]을 이용하였습니다.



거래 생성 그래프는 두 블록이 한 점으로 합쳐져서 1초에 한 점씩 찍히도록 되어 있습니다. 해당 그래프에서 트랜잭션 수가 블록마다 변하면서 생성되는 것을 확인할 수 있으며, 이는 트랜잭션이 노드를 통해 전파되어 포물레이터에 포함되기 때문에 매 블록 생성마다 서로 다른 트랜잭션이 포함되기 때문입니다.

구간(높이)	구간 거래 수	구간 시작	구간 종료	소요시간	최고 TPS	평균 TPS
1~600	2,131,248	16:29:42	16:34:51	309	12,428	6,897
601~1200	2,161,217	16:34:51	16:39:52	301	13,356	7,180
1201~1800	2,177,712	16:39:52	16:44:52	300	14,002	7,259
1801~2400	2,121,227	16:44:52	16:49:52	300	12,367	7,070
2401~3000	2,144,324	16:49:52	16:54:53	301	12,951	7,124

트랜잭션 처리를 구간별로 정리한 표를 확인해보면 첫 구간에서 포물레이터의 시작 속도가 다소 차이가 나기 때문에 약간의 지연이 발생하였지만, 이후 블록이 동기화되고 안정적으로 블록이 생성되었습니다. 일정한 트랜잭션을 주입하는 것과 다르게 포물레이터가 지속적으로 트랜잭션을 받고 검사하고 트랜잭션 풀에 넣게 되므로 CPU의 사용이 분산되어 약간의 지연이 발생하였습니다.

결론

블록체인의 처리 속도에 대한 개선은 블록체인이 발전하는 매우 중요한 부분입니다. FLETA의 블록체인은 실제 서비스에 활용되고 사용되는 블록체인 기술을 목표로 하고 있습니다.

FLETA는 체인 낼 수 있는 최대의 속도 14,000 TPS를 확인하였고, 지속적인 Tx거래 처리량은 12,000TPS 이상의 속도로 진행할 수 있도록 기술 개발을 진행하였습니다.

또한 해당 테스트를 진행한 환경이 FLETA 내부 개발 Lab에서 진행된 것이 아니라, 다수의 지역 외부 망에 블록생성노드와 환경을 설정하여 진행하여 얻어낸 결과라는 것은 객관성을 최대한 확보한 결과입니다.

FLETA는 앞으로도 지속적인 블록체인 기술 개발을 진행할 것이며, 기술적 목표이자 최대의 라이벌이 다른 누구도 아닌 항상 FLETA 자신이 되기 위해 최선을 다하겠습니다.





Thank You.

