JDCS
디지털콘텐츠학회논문지
Journal of Digital Contents Society
Vol. 24, No. 5, pp. 1083-1089, May. 2023

Check for updates

공공기록물 관리 블록체인에서 타임스탬프 기반의 효율적 탐색 방법

손 민 영[1] · 손 기 봉[1] · 김 영 학[2*]
[1]금오공과대학교 컴퓨터공학과 박사
[2*]금오공과대학교 컴퓨터공학과 교수

Efficient Search Method Based on Timestamp in Public Records Management Blockchain

Min-Young Son[1] · Ki-Bong Son[1] · Young-Hak Kim[2*]

[1]Ph.D, Department of Computer Engineering, Kumoh National Institute of Technology, Gumi 39177, Korea
[2*]Professor, Department of Computer Engineering, Kumoh National Institute of Technology, Gumi 39177, Korea

[요 약]

블록체인은 데이터의 위변조 방지 및 무결성을 위해 선형 연결리스트 방식으로 설계되어 빈번하게 데이터의 탐색이 발생하는 응용에서는 비효율적이다. 공공기록물과 같이 시간대별 기록을 중시하는 프라이빗 블록체인의 응용에서는 특정 기간에 대한 기록물 탐색이 빈번하게 일어날 수 있다. 일반적으로 블록체인에서 한 블록이 생성될 때 블록의 생성 시간이 타임스탬프 항목에 저장된다. 본 연구에서는 공공기록물 관리 블록체인에서 효율적인 탐색을 위해 타임스탬프 기반의 인덱스 관리 방법을 제안한다. 제안된 방법은 기존의 블록체인 특성을 그대로 유지하면서 특정 기간의 데이터는 타임스탬프를 사용하여 효율적으로 탐색할 수 있다. 제안된 방법의 평가를 위해 일반적인 선형 구조 블록체인과 제안한 블록체인과의 시뮬레이션 모델을 정의하고 이를 사용하여 성능을 비교한다. 시뮬레이션 결과 제안된 타임스탬프 기반의 방법이 일반 선형 블록체인에 비해 시간별 기록물 검색에서 훨씬 빠른 성능을 보였다.

[Abstract]

Because blockchain is designed in a linear linked manner to prevent forgery and falsification of data, it is inefficient in applications where data search occurs frequently. In private blockchain applications that have time-specific records, such as public records, searches for records of a specific period can occur frequently. In general, when a block is created in a blockchain, the creation time of the block is stored in a timestamp field. In this paper, we propose a timestamp-based index management for efficient searching in a public records management blockchain. The proposed method can efficiently search data of a specific period using timestamps while maintaining the characteristics of the existing blockchain. To evaluate the proposed method, we designed a simulation model between a general blockchain with linear structure and the proposed blockchain and compared the performance using it. As a result of the simulation, the proposed timestamp-based method showed much faster performance in retrieval of records by period compared to general linear blockchain.

색인어 : B+ 트리, 블록체인, 기록물 관리, 기록물 탐색, 타임스탬프
Keyword : B+ Tree, Block Chain, Records Management, Records Search, Timestamp

http://dx.doi.org/10.9728/dcs.2023.24.5.1083

Received 21 March 2023; **Revised** 17 April 2023
Accepted 24 April 2023

***Corresponding Author; Young-Hak Kim**

Tel: +82-54-478-7524
E-mail: kimyh@kumoh.ac.kr

Ⅰ. 서 론

데이터를 저장하는 방식과 합의는 블록체인 시스템에서 주요한 구성 요소이다. 데이터베이스 맥락에서 블록체인은 분산된 트랜잭션의 관리를 위한 솔루션으로 볼 수 있다. 블록체인 시스템과 일반적인 분산 데이터베이스 시스템의 주요한 차이는 데이터를 저장하는 방식이다. 일반적인 분산 데이터베이스 시스템은 데이터를 분할하여 서로 다른 장치에 고르게 분산시켜 저장한다. 이 경우 각 데이터가 저장된 분산 장치를 찾기 위해 매핑 함수를 생성한다.

그러나 블록체인의 동작 방식은 일반적인 분산 데이터베이스와 달리, 분산된 장치에 복제된 동일하거나 유사한 데이터를 저장한다. 즉, 모든 분산 장치는 데이터 일부가 아닌 동일한 전체 데이터를 복사하여 저장한다. 블록체인의 이러한 설계 방식은 보안에 특화되었다. 만약 특정 사용자가 블록체인의 데이터를 악의적으로 조작하려는 경우, 단 하나의 분산 장치가 아닌 전체 분산 장치의 데이터 복사본을 수정해야 하므로, 악의적 변조의 어려움과 데이터의 무결성을 증가한다[1].

블록체인에서 트랜잭션 데이터는 P2P 네트워크에 가입한 모든 노드에 전파되고 노드에 의해 검증된다. 비트코인 블록체인에서 모든 트랜잭션을 검증하는 풀 노드(Full Node)와 자신과 관련된 트랜잭션 일부를 검증하는 SPV(Simplified Payment Verification) 노드를 갖는다. 풀 노드는 다른 SPV 노드 등의 요청 시 트랜잭션을 전파한다. 블록체인 사용자의 측면에서 볼 때 SPV 노드는 풀 노드에 비해 훨씬 가볍다. SPV 노드는 트랜잭션 데이터를 저장하기 위한 스토리지 용량이나 확인해야 하는 트랜잭션 수를 크게 줄일 수 있기 때문이다[2].

블록체인에서 트랜잭션들은 블록으로 생성되고 이 블록들은 해시에 의해 선형 연결 리스트와 같은 방식으로 구성된다. 즉, 블록체인에서 최초에 생성된 블록을 제네시스 블록이라 하며 새로운 블록이 생성될 때마다 시간 순서대로 해시 주소로 블록이 연결된다. 블록체인은 데이터의 위변조 및 무결성을 강조하여 설계된 기술로 특정 트랜잭션을 탐색하는 등의 쿼리 작업에는 비효율적이다[3].

최근에 정부에서 “공공기록물 관리에 관한 법률”이 본격적으로 시행되어 공공기관에서는 기록물 관리를 의무화하고 있다[4]. 공공기록물은 연대 및 날짜별 순서대로 기록되며 블록체인에 적합한 응용의 한 예로 들 수 있다. 또한, 이러한 기록물이 블록체인으로 관리될 경우 특정 기간에 대한 기록물 탐색이 빈번하게 이루어질 수 있다. 이러한 상황에서 블록체인의 고유한 순차적 특성 때문에 빈번한 탐색에 대한 문제가 해결되어야 한다.

본 연구에서는 공공기록물과 같이 시간대별 기록을 중시하는 블록체인 응용에서 효율적인 탐색을 위해 타임스탬프 기반의 인덱스 관리 방법을 제안한다. 블록체인에서 한 블록이 생성될 때 타임스탬프 항목에는 해당 블록이 생성된 시간 정보가 저장된다. 제안된 방법은 타임스탬프를 킷값으로 인덱스를 구성하여 특정 시간대별로 원하는 블록을 탐색하도록 한다. 제안된 방법의 평가를 위해 시뮬레이션 모델을 정의하고 이를 기본으로 성능을 평가한다.

2장에서 관련 연구를 기술하고 3장에서 본 연구에서 제안한 새로운 형식의 블록체인 구조를 설명한다. 4장에서는 제안된 블록체인 구조를 기존의 방법과 성능 비교하며, 마지막으로 5장에서 결론을 맺는다.

Ⅱ. 관련 연구

홍기완 등은 대학에서 종이로 보관되고 있는 성적 기록물들을 대상으로 블록체인 기술을 사용하여 공공기록물을 관리하는 서비스 모델을 제안하였다[5]. 이 연구에서는 대학의 공공기록물 관리 현황을 분석하고 Hyperledger Fabric을 이용하여 서비스 모델에 대한 프로토타입을 설계하였다. 홍덕용은 공공기록물 관리를 위해 블록체인 기술의 적용 가능성을 검토하고 이에 대한 개념적인 측면에서 구축 절차 및 방법을 제안하였다[6]. 이들 선행연구에서 공공기관에서 공공기록물을 블록체인으로 관리하기 위한 개념적인 측면에서 서비스 방안을 제한하였다. 그러나 이들이 제안한 연구에서 공공기록물을 효율적으로 탐색하기 위한 구체적인 방법은 다루지 않았다.

블록체인 탐색 웹사이트에 따르면 비트코인 블록체인의 경우 2023년 2월 20일 기준 전체 블록의 수는 777,491개, 한 블록의 평균 크기는 2.146MB, 한 블록당 평균 트랜잭션의 수는 1837.29, 블록체인의 총 크기는 457.45GB이다[7]. 블록체인에서 시간이 지나면서 새로운 블록이 지속적으로 생성되기 때문에 전체 블록의 크기도 이에 비례하여 증가한다. 비트코인 블록체인에서는 전체 블록을 저장하는 풀 노드와 트랜잭션 검색을 위한 가벼운 정보를 저장하는 SPV 노드가 있다. 최근에 블록체인의 크기가 커지면서 전체 블록을 저장하기 위해 별도의 스토리지 관리 방법과 이러한 스토리지에서 효율적으로 블록을 탐색하기 위한 연구가 진행되고 있다.

Li 등은 블록체인 데이터를 외부 MongoDB 데이터베이스에 복사하여 이를 활용하여 블록체인 데이터 질의를 위한 방법을 제안하였다[8]. TU 등은 다중 트랜잭션 모드의 블록체인에서 참여하는 사용자 집합, 블록 이름 집합, 블록 이름 목록 등을 별도의 캐시 테이블로 관리하여 B+ 트리를 사용하여 빠른 탐색을 위한 방법을 제안하였다[9]. Feng 등은 기존에 제안된 블록체인 검색 관련 선행연구를 분석하여 블룸 필터와 변형된 B+ 트리를 사용하여 효율적인 탐색을 위한 기본구조 설계하고 그 결과를 검증하였다[10]. 이들 연구에서 제안한 방법은 일반적인 블록체인 환경에서 장점이 있지만 공공기록물 관리와 같은 응용에 적용하기 위해서 별도의 스토리지가 필요하고 무거운 측면이 있다.

전술한 것과 같이 블록체인은 고유한 특성으로 인해 빈번

하게 탐색이 이루지는 분야에서는 비효율적인 자료구조이다. 본 연구에서 공공기록물 등과 같은 프라이빗 블록체인 응용에서 특정 기간별로 효율적으로 블록 데이터를 탐색하는 타임스탬프 기반의 방법을 제안한다. 비트코인과 같은 블록체인의 경우 트랜잭션의 수가 많으며 차지하는 메모리 공간은 아주 작다. 비트코인 블록체인과 비교하면 공공기록물의 경우 트랜잭션의 개수는 작으며 한 트랜잭션이 차지하는 메모리 공간은 훨씬 크다.

Ⅲ. 제안 방법

이 장에서는 공공기록물 관리 블록체인 응용에서 효율적인 탐색을 위해 타임스탬프 기반의 인덱스 구성 방법을 설명한다. 제안된 방법에 대한 성능 평가 모델과 비교는 다음 장에서 기술한다.

3-1 타임스탬프 기반의 인덱스 구성

그림 1은 블록들이 해시 주소에 의해 연결된 블록체인의 분산 원장 데이터 구조를 보여준다. 블록체인에서 한 블록은 크게 헤더 부분과 바디 부분으로 구성된다. 헤더 부분은 이전 블록의 해시, 타임스탬프, 머클 루트 값 등이 포함되며 바디 부분은 트랜잭션들로 구성된다.

홍기완 등은 대학에서 성적 기록물에 관한 블록체인을 그림 1과 같이 서비스 레이어, 보존 레이어, 블록체인 레이어로 구분하여 설계하였다[5]. 블록체인 레이어는 일반적인 블록체인 구조를 사용하며, 블록에 포함되는 트랜잭션 기록물이 메타데이터 형식으로 정의되어 저장된다. 본 연구에서는 공공기록물에 대한 블록체인 구조는 선행연구 결과를 따르며, 타임스탬프 항목을 인덱스로 구성하여 특정 기간별로 원하는 자료를 효율적으로 탐색하는 방법을 제안한다.

블록체인의 블록 헤더에서 타임스탬프 필드는 블록이 생성된 날짜 및 시간 등의 기본 정보가 저장된다. 본 연구에서는 사용자가 원하는 기간별로 블록의 자료를 효율적으로 탐색하기 위해 타임스탬프를 키 인덱스로 사용하여 B+ 트리를 구성한다. B+ 트리는 데이터베이스 분야에 널리 사용되는 자료구조로 인덱스 노드와 리프 노드로 구성된다. 인덱스 노드에는 타임스탬프의 킷값과 다음 인덱스 노드에 대한 포인터가 저장되고, 리프 노드는 킷값의 영역에 해당하는 블록을 가리킨다. 리프 노드가 가리키는 블록들은 왼쪽부터 오른쪽으로 순서에 따라 연결되어 블록체인의 고유한 특징을 그대로 유지한다.

(알고리즘 1)은 이미 존재하는 블록체인에서 제네시스 블록부터 시작하여 모든 블록을 순서에 따라 가져와 블록의 타임스탬프의 킷값에 따라 B+ 트리를 구성하는 과정을 보여준다. 블록체인 참여자의 경우 블록체인에서 모든 블록을 소유한 풀 노드와 SPV 형태의 참여 노드가 있다. 만일 SPV 참여자의 경우 풀 노드에 요청하여 탐색에 필요한 타임스탬프를 기반으로 하는 가벼운 인덱스 구조를 갖는다.

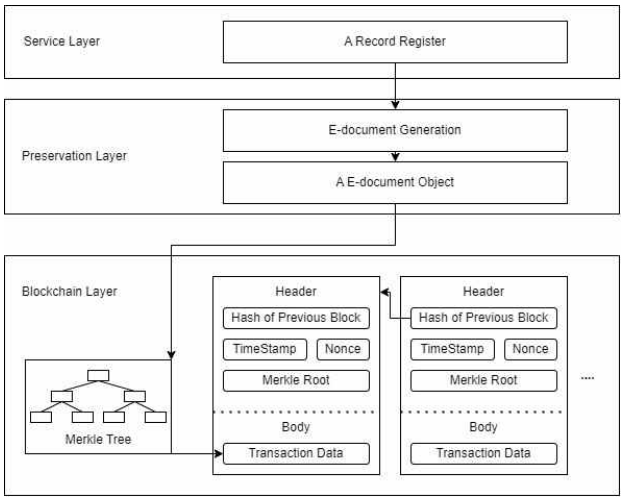

그림 1. 공공기록물의 블록체인 구조
Fig. 1. Blockchain structure of public records

(알고리즘 1)에서 기존의 블록체인 블록이 존재하지 않고 새로운 블록체인이 만들어질 경우 역시 같은 절차를 반복하면 된다. 일반적인 B+ 트리에서 리프 노드에 연결되는 데이터 블록의 경우 임의의 위치에 추가될 수 있지만, 제안된 방법에서 새로 생성된 블록은 가장 마지막 리프 노드에 추가된다. 따라서 마지막 리프 노드에 새로 생성된 블록이 추가될 경우 현재 리프 노드에서 부모 노드를 따라 거슬러 올라가면서 노드를 분할하여 트리를 조정하면 된다.

그림 2는 (알고리즘 1)을 적용하여 만들어진 타임스탬프를 킷값으로 하여 생성된 B+ 트리의 예를 보여준다. 그림 2에서 (a)는 한 모듈(박스)에 2개의 블록을 갖는 블록체인의 예를 보여주며 편의상 타임스탬프를 일련의 숫자로 표현하였다. 공공기록물 관리에서 타임스탬프는 블록이 생성된 날짜 및 시간 정보가 저장된다. 그림 2에서 (b)는 (a)의 블록체인 블록을 타임스탬프를 킷값으로 하여 인덱스 노드를 구성한 예를 보여준다. 그림 2에서 보인 것과 같이 인덱스 노드는 타임스탬프의 킷값과 다음 레벨 인덱스 노드에 대한 포인터 값을 갖는다. 리프 노드는 실제 블록체인 블록을 가리키며 첫 번째 리프 노드에서 데이터 블록을 따라가면 원래 블록체인 구조를 그대로 유지한다.

3-2 타임스탬프 기반의 탐색

(알고리즘 1)에서 블록체인 블록에서 타임스탬프 기반으로 효율적인 탐색을 위한 인덱스를 구성하는 방법을 설명하였다. 다음에 이 자료구조를 사용하여 타임스탬프 기반으로 블록체인의 블록 또는 블록에 포함된 트랜잭션을 탐색하는 방법을 살펴본다. (알고리즘 2)는 (알고리즘 1)에서 타임스탬프를 기반으로 하여 생성된 트리를 사용하여 원하는 타임스탬프 킷

알고리즘 1. 타임스탬프를 킷값으로 B+ 트리 생성

입력 : 공공기록물 관리 블록체인에서 한 블록
출력 : 블록의 타임스탬프를 인덱스 노드로 구성한 B+ 트리
1. 전체 블록을 갖지 않은 참여자의 경우 전체 블록을 갖는 참여자에게 타임스탬프 인덱스 구조 요청한다.
2. 전체 블록을 갖는 참여자가 다음을 수행하여 타임스탬프 기반의 B+ 트리를 생성한다.
   1) 블록체인에서 하나의 블록을 가져와 리프 노드에 할당한다. 최초 생성 시는 제네시스 블록을 가져온다.
   2) 리프 노드에 할당된 블록의 타임스탬프를 킷값으로 부모 노드를 조정한다. 기록물은 시간에 따른 순서대로 블록이 생성되기 때문에 생성된 블록은 마지막 리프 노드에 추가된다.
   3) 마지막 리프 노드에 블록을 추가하여 부모 노드가 꽉 차있을 경우 부모 노드를 분할하여 킷값을 조정한다.
   4) 부모 노드를 거슬러 올라가면서 분할이 이루어지지 않을 때까지 1)~4)의 과정을 반복한다.
   5) 루트 노드에서 분할이 일어날 경우 하나의 킷값과 두 개의 포인터를 갖는 새로운 루트 노드를 생성한다.
3. 타임스탬프를 키 인덱스로 하여 생성된 B+ 트리를 요청 참여자에 반환한다.

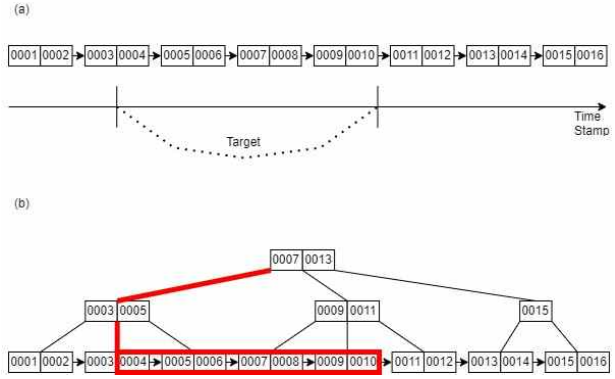

그림 2. 블록체인 블록의 타임스탬프를 킷값으로 B+ 트리를 구성한 예 (a) 한 모듈에 2개의 블록을 갖는 블록체인 (b) 블록의 타임스탬프를 킷값으로 인덱스 노드의 구성

Fig. 2. An example of configuring a B+ tree with the timestamp of a blockchain block as a key value (a) Blockchain with two blocks in one module (b) Composition of index node with a block timestamp as a key value

값에 일치하는 블록 또는 블록에 포함된 트랜잭션을 탐색하는 과정을 보여준다.

(알고리즘 2)는 타임스탬프의 킷값에 일치하는 탐색의 예를 보여주나, 이를 확장하면 특정 범위의 타임스탬프에 포함하는 모든 블록 또는 블록에 포함된 트랜잭션을 탐색할 수 있다. (알고리즘 2)의 수행 과정을 설명하기 위해 그림 2의 예를 사용한다. 그림 2의 (a)의 블록체인에는 16개의 데이터 블록이 있으며 각 사각형의 숫자는 블록이 생성된 타임스탬프 값을 보여준다. 그림 2에서 빨간색으로 표시된 것과 같이 블록체인에서 타임스탬프의 킷값 4와 10 범위에 포함된 모든 블록과 블록 내에 포함된 트랜잭션을 탐색한다고 가정해 본다.

타임스탬프의 시작 값 4를 킷값으로 하여 그림 2의 (b)에서 보인 것과 같이 트리의 루트 노드부터 탐색을 시작한다. 그림 2에서 루트 노드에서 시작하여 굵게 표시된 링크를 따라

알고리즘 2. 타임스탬프를 킷값으로 기록물의 탐색

입력 : 타임스탬프 킷값
출력 : 타임스탬프 킷값에 해당하는 블록 또는 블록 내의 트랜잭션
1. 트리의 루트 노드가 비어있을 경우 종료하고 그렇지 않으면 다음 단계를 수행한다.
2. 타임스탬프의 킷값을 현재 노드의 킷값과 비교하여 해당 포인터를 따라 자식 노드로 이동한다. 이러한 과정을 리프 노드가 나올 때까지 반복한다.
3. 리프 노드가 가리키는 블록에서 타임스탬프 킷값과 일치하는 트랜잭션을 순차 탐색한다.
4. 타임스탬프 킷값에 일치하는 블록 또는 블록 내의 트랜잭션을 반환한다.

리프 노드까지 이동하면 킷값 4에 해당하는 블록을 찾을 수 있다. 해당 노드에서는 타임스탬프에 따라 블록체인의 블록이 시간순으로 연결되어 있어 이 위치부터 블록을 순차 탐색하면 된다. 이러한 블록의 순차 탐색을 타임스탬프가 10인 블록까지 반복한다.

Ⅳ. 제안 방법의 평가

이 장에서는 본 연구에서 제안한 타임스탬프 기반의 효율적인 탐색에 대한 평가 모델을 정의한다. 또한, 이 평가 모델에 따라 시뮬레이션을 통하여 결과를 비교한다.

4-1 탐색 방법의 평가 모델

본 연구에서 제안한 블록체인의 성능이 우수함을 입증하기 위해서 기존 블록체인과 시뮬레이션을 수행하고 그 결과를 비교 분석한다. 두 블록체인을 시뮬레이션 하기 위해서는 기존 블록체인과 제안한 블록체인의 탐색 방법에 대한 평가 모델을 정의해야 한다. 이 절에서는 시뮬레이션에 사용된 평가 모델을 정의하고, 다음 절에서 시뮬레이션 결과를 비교 분석한다.

먼저 제안된 연구 결과와 비교를 위해 일반적인 블록체인에서 블록체인 탐색에 대한 평가 방법을 설명한다. 본 평가 모델에서는 블록에 포함된 바디 부분에서 트랜잭션을 읽는 시간만을 고려한다. 평가 모델을 위해 다음을 정의한다.

n_i: i번째 블록에서 트랜잭션의 수
t_{ij}: i번째 블록에서 j번째 트랜잭션을 읽는 시간

일반적인 블록체인의 경우 순차 탐색을 수행하기 때문에 k번째 블록을 탐색하기 위한 시간은 다음과 같다.

$$\sum_{i=1}^{k}\sum_{j=1}^{n_i} t_{ij} \quad (1)$$

식 (1)은 블록체인에서 한 블록에 포함되는 트랜잭션의 개수

와 개별 트랜잭션의 크기가 다른 상황을 고려한다. 시뮬레이션의 편의를 위해 한 블록에 포함된 트랜잭션의 수가 같고 트랜잭션의 크기가 정형화되어 있다고 가정한다. 그러면 식 (1)은 식 (2)와 같이 단순화될 수 있다. 여기서, n은 한 블록에서 트랜잭션의 수이고 t는 한 개의 트랜잭션을 읽는 시간을 의미한다.

$$k\times(n\times t) \tag{2}$$

만일 k번째 블록 내에서 r번째 트랜잭션만을 찾고자 하는 경우 탐색시간은 식 (3)과 같다.

$$(k-1)\times(n\times t)+(r\times t) \tag{3}$$

다음에 (알고리즘 2)에서 제안한 타임스탬프 기반의 탐색 방법에 대한 평가 모델을 정의한다. (알고리즘 2)는 (알고리즘 1)에서 생성된 타임스탬프 킷값 기반의 인덱스 노드에 저장된 B+ 트리를 사용한다. 일반적으로 B+ 트리에서 루트 노드를 제외한 내부 노드는 최대 m개의 자식 노드와 최소 m/2개의 자식 노드를 갖는다. 또한, 각 노드는 최대 m-1개의 킷값과 최소 ⌜m/2⌝-1개의 킷값을 갖는다. 여기서 m은 트리의 차수를 의미한다. N을 블록체인에서 전체 블록의 수라 할 때 B+ 트리에서 주어진 킷값을 찾기 위한 최선의 시간 복잡도는 $O(\log_m N)$, 최악의 시간 복잡도는 $O(\log N)$이다. 본 논문에서 시뮬레이션 모델을 위해 최악을 시간 복잡도인 $O(\log N)$을 고려한다. 이러한 시간을 고려하여 타임스탬프 기반의 탐색에 대한 시뮬레이션 모델을 아래와 같이 정의한다.

(1) 타임스탬프의 킷값을 갖는 블록 내에서 원하는 트랜잭션(r번째 트랜잭션)을 찾기 위한 수행시간은 식 (4)와 같다. 여기서 c는 트리에서 한 레벨에서 다음 레벨로 이동하는데 걸리는 시간을 의미한다. 실제로 c는 실행과정에서 무시할 수 있는 상수이다. r은 리프 노드가 가리키는 블록의 첫 트랜잭션에서 시작하여 원하는 트랜잭션을 찾기 위해 트랜잭션을 연속적으로 읽는 반복 횟수를 의미한다.

$$\log N\times c+r\times t \tag{4}$$

(2) 타임스탬프의 킷값을 갖는 블록부터 시작하여 k개의 블록을 연속적으로 탐색하기 위한 수행시간은 식 (5)와 같다. 즉, 리프 노드가 가리키는 블록에서 시작하여 k개의 블록을 연속적으로 탐색한다.

$$\log N\times c+k\times(n\times t) \tag{5}$$

4-2 시뮬레이션 결과 분석

본 절에서는 제안된 블록체인의 성능을 평가하기 위하여 시뮬레이션 결과를 분석한다. 시뮬레이션에 사용된 블록에 대한 변수들은 2022년 9월부터 2023년 2월까지 6개월간 비트코인 블록체인에 사용된 블록들의 데이터를 blockchair 사이트(https://blockchair.com)를 통해 수집 및 분석하여 반영하였다. 수집된 블록을 분석한 결과 하나의 블록에 포함된 트랜잭션의 개수는 평균 약 2,500개이고, 표준편차는 약 850개로 나타났다.

표 1. 시뮬레이션 변수와 값
Table 1. Simulation variables and values

Simulation Variables	Symbol	Values
Total number of blocks	N	[500,1000,5000]
The number of transactions included in each block	n	N(2500,850)
The location of the block containing the data to be searched	k	DU(N)
The location of the transaction containing the data to be searched	r	DU(2500)
Time to read one transaction	t	1μs (1×10^{-6} sec)
Time to move to the next level in the tree	c	1ms (1×10^{-3} sec)

이를 토대로 본 시뮬레이션의 변수를 설정하였으며, 그 내용은 표 1과 같다. 전체 블록의 개수는 500개, 1,000개, 그리고 5,000개로 나누어 실험을 진행하였으며, 블록에 포함된 트랜잭션 n이 평균 2,500개와 표준편차 850개의 정규분포를 따른다고 가정하여 시뮬레이션을 위해 난수를 생성하였다. 탐색하고자 하는 데이터의 위치는 전체 블록과 트랜잭션의 어디든지 같은 확률로 위치할 수 있으므로 첫 번째 블록의 첫 번째 트랜잭션부터 N 번째 마지막 블록의 마지막 트랜잭션까지 같은 확률을 가지고 선정될 수 있도록 이산 균등분포를 따르는 난수를 생성하였다. 트랜잭션을 읽고 블록을 찾는 변수 t와 c는 일반적인 디스크 접근 시간, 트랙 전송 시간 등을 고려하여 설정하였다. 검색하는 데이터는 타임스탬프이며, 블록과 트랜잭션은 타임스탬프 순서로 정렬되어 있다고 가정한다.

제안된 블록체인의 비교 분석이 되는 일반적인 블록체인은 순차탐색을 통해 검색을 수행하며, 4-1절 식(1)~(3)과 같은 시간 복잡도를 가진다. 제안된 블록체인은 B+ 트리의 최대 자식 노드의 값에 따라 탐색 시간의 차이가 있으며, 식 (4)~(5)의 시간 복잡도를 가진다. 제안된 블록체인의 성능을 평가하기 위하여 식 (3)과 식 (4)를 기반으로 변수를 사용하여 시뮬레이션을 수행한 결과는 그림 3과 같다.

그림 3의 세 그래프는 위에서 아래 순서대로 전체 블록의 개수를 500개, 1,000개와 5,000개로 지정하여 시뮬레이션을 수행한 결과이다. 그 결과 일반 블록체인에서는 탐색하고자 하는 트랜잭션의 위치인 k가 크면 클수록 탐색의 시간이 선형 증가함을 보였다. 본 연구에서 제안된 블록체인은 B+ 트리를 사용하여 k 값에 무관하게 일정한 속도를 보였다. 이때, 트리의 자식 수 m이 클수록 시간이 적게 소요되었는데, 이는 트리의 높이가 줄어들어 디스크 트랙을 탐색하는 시간이 줄어들기 때문이다. 전체 블록의 개수 N이 크면 클수록 기존 블록체

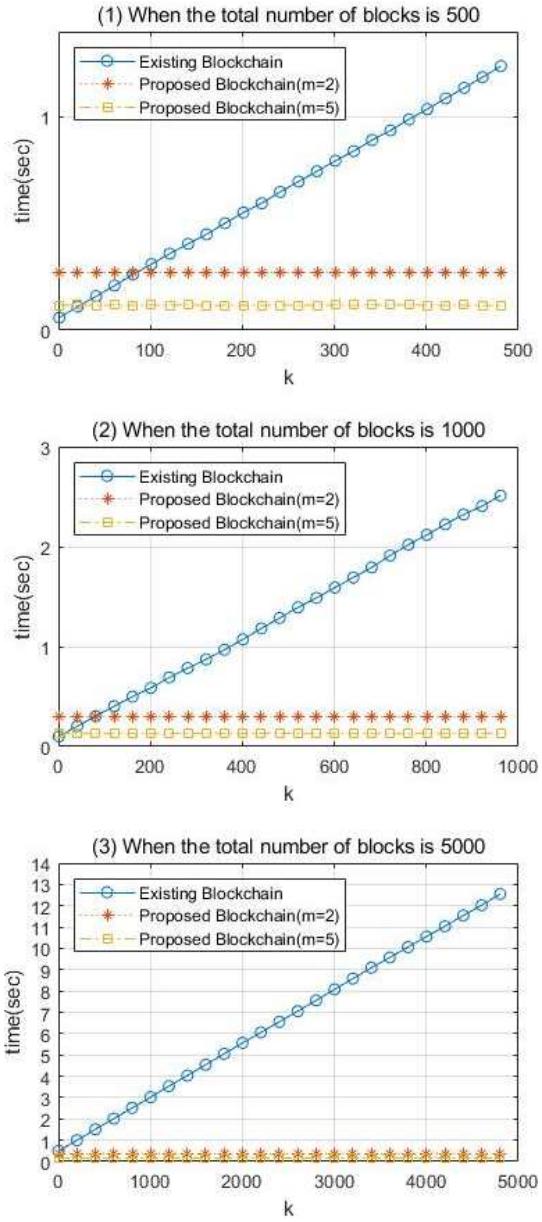

그림 3. 제안된 타임스탬프 기반 탐색의 성능 비교(최대 자식 노드의 수가 m=2, m=5일 때)

Fig. 3. Performance comparison of the proposed timestamp-based search(When the number of the maximum child node is m=2, m=5)

인과 제안한 B+ 트리를 사용한 블록체인 간의 탐색 속도는 더 크게 나타났다.

V. 결 론

본 논문에서는 공공기록물 등과 같은 블록체인 응용에서 효율적인 탐색 방법을 제안하였다. 블록체인은 데이터의 위변조 방지 및 무결성과 같은 고유의 특징을 갖기 때문에 선형 연결리스트 방식으로 구성된다. 이러한 블록체인의 특성으로 인해 빈번한 탐색이 일어나는 응용에서는 매우 비효율적이다. 공공기록물 등과 같은 블록체인 응용에서는 필요에 따라 특정 시간대별로 자료 조회가 빈번하게 일어날 수 있다.

본 논문에서는 이러한 문제를 효율적으로 해결하기 위해 타임스탬프 기반의 인덱스를 구성하여 시간대별로 효율적으로 원하는 자료를 탐색하는 방법을 제안하였다. 또한, 성능 평가를 위해 기존 방법과 제안된 방법의 평가 모델을 제시하여 시뮬레이션을 통하여 결과를 비교하였다. 시뮬레이션 결과 본 논문에서 제안한 타임스탬프 기반으로 방법을 사용하여 시간대별로 원하는 자료를 훨씬 빠르게 탐색할 수 있음을 보였다. 제안한 타임스탬프 기반 방법은 자식 노드의 개수가 많을수록 트리의 높이가 작아지므로 탐색시간이 더욱 빠르게 나타났다. 일반적인 블록체인은 전체 블록의 개수가 많고 탐색하고자 하는 데이터의 위치가 선형적인 블록체인의 뒤쪽에 위치할수록 선형적으로 증가하는 탐색시간을 가지는 특징과는 달리 제안한 타임스탬프 기반 방법은 전체 블록의 개수와 탐색 데이터의 위치에 영향을 적게 받는 것으로 나타났다.

본 연구는 시간대별로 생성되는 공공기록물의 관리에서 타임스탬프 기반의 블록체인 탐색 방법을 제안하여 일반적인 블록체인보다 기록물의 탐색 성능을 크게 향상하였다. 향후 본 연구의 실질적 사용을 위해 공공기록물과 연계하여 일반 사용자를 위한 구체적인 소프트웨어 구현이 필요하다. 본 연구 결과는 공공기록물 관리 이외에도 타임스탬프를 기반으로 하는 블록체인 응용 분야에 확대될 수 있다.

감사의 글

이 연구는 금오공과대학교 대학 학술연구비로 지원되었음(2021학년도).

참고문헌

[1] T. T. A. Dinh, R. Liu, M. Zhang, G. Chen, B. C. Ooi, and J. Wang, “Untangling Blockchain: A Data Processing View of Blockchain Systems,” in *Proceedings of IEEE Transactions on Knowledge and Data Engineering*, Vol. 30, No. 7, pp. 1366-1385, January 2018. https://doi.org/10.1109/TKDE.2017.2781227

[2] S. Morishima and H. Matsutani, “Accelerating Blockchain Search of Full Nodes Using GPUs,” in *Proceedings of 2018 26th Euromicro International Conference on Parallel, Distributed and Network-based Processing (PDP)*, Cambridge: UK, pp. 244-248, June 2018.

https://doi.org/10.1109/PDP2018.2018.00041
[3] T. L. Huang and J. Huang, “An Efficient Data Structure for Distributed Ledger in Blockchain Systems,” *2020 International Computer Symposium (ICS))*, Tainan, pp. 175-178, February 2021. https://doi.org/10.1109/ICS51289.2020.00043
[4] Enforcement of the Public Records Management Act, Partial Revision of Presidential Decree No. 32272, 2022.07.05.
[5] G. W. Hong and H. B. Chang, “Study on Blockchain Based University Public Records Management Service,” *The Journal of Society for e-Business Studies*, Vol. 26, No. 1, pp. 79-91, February 2021. https://doi.org/10.7838/jsebs.2021.26.1.079
[6] D. Hong, “A Study on the Application of Blockchain Technology to the Record Management Model,” *Journal of Korean Society of Archives and Records Management*, Vol. 19, No. 3, pp. 223-245, August 2019. http://dx.doi.org/10.14404/JKSARM.2019.19.3.223
[7] Blockchain.com. Bitcoin - BTC Price, Live Chart, and News [Internet]. Available: https://www.blockchain.com/.
[8] Y. Li, K. Zheng, Y. Yan, Q. Liu, and X. Zhou, “EtherQL: A Query Layer for Blockchain System,” in *Database Systems for Advanced Applications, Proceedings of the 22nd International Conference*, Suzhou: China, pp. 556-567, March 2017. https://doi.org//10.1007/978-3-319-55699-4_34
[9] J. Tu, J. Zhang, S. Chen, T. Weise, and L. Zou, “An Improved Retrieval Method for Multi-transaction Mode Consortium Blockchain,” *Electronics*, Vol. 9, No. 2, February 2020. https://doi.org/10.3390/electronics9020296
[10] H. Feng, J. Wang, and Y. Li, “An Efficient Blockchain Transaction Retrieval System,” *Future Internet*, Vol. 14, No. 9, September 2022. https://doi.org/10.3390/fi14090267

손민영(Min-Young Son)

2010년 : 고려대학교 정보경영공학과 (공학석사)
2017년 : 금오공과대학교 컴퓨터공학과 (공학박사)

※관심분야 : 네트워크, 분산처리, 그래프, 데이터마이닝, Blockchain

손기봉(Ki-Bong Son)

2015년 : 조선대학교 전기·전자·통신교육 (교육학석사)
2021년 : 금오공과대학교 컴퓨터공학과 (공학박사)

※관심분야 : Front-end Design & Verification Methodology, Blockchain

김영학(Young-Hak Kim)

1989년 : 서강대학교 전자계산학과(공학석사)
1997년 : 서강대학교 전자계산학과(공학박사)

1999년 3월～현 재: 금오공과대학교 컴퓨터공학과 교수
※관심분야 : 블록체인, 병렬알고리즘, 분산처리, 임베디드시스템 등