



(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2019-0079517
(43) 공개일자 2019년07월05일

(51) 국제특허분류(Int. Cl.)
G06F 16/00 (2019.01) G06Q 20/36 (2012.01)
G06Q 20/38 (2012.01)
(52) CPC특허분류
G06F 16/2246 (2019.01)
G06Q 20/3678 (2013.01)
(21) 출원번호 10-2018-0160728
(22) 출원일자 2018년12월13일
심사청구일자 2018년12월13일
(30) 우선권주장
1020170170895 2017년12월13일 대한민국(KR)

(71) 출원인
서강대학교산학협력단
서울특별시 마포구 백범로 35 (신수동, 서강대학교)
(72) 발명자
박수용
서울특별시 마포구 독막로18길 5, 101동 1001호(상수동, 상수두산위브아파트)
이동영
서울특별시 마포구 독막로21길 15-6, 302호(창전동)
(74) 대리인
특허법인충현

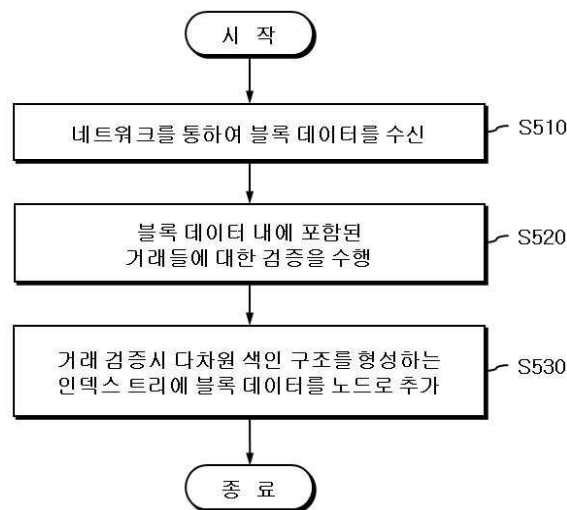
전체 청구항 수 : 총 20 항

(54) 발명의 명칭 **블록체인 기반 IoT 환경에서의 다중 검색을 지원하는 데이터 구조체를 이용한 검색 방법 및 그 방법에 따른 장치**

(57) 요약

본 발명은 블록체인 기반 IoT 환경에서의 다중 검색을 지원하는 데이터 구조체를 이용한 검색 기술에 관한 것으로, 블록체인을 위한 검색 기능의 제공 방법은, 네트워크를 통하여 블록 데이터를 수신하고, 수신된 블록 데이터 내에 포함된 거래들에 대한 검증을 수행하며, 거래 검증시 각 거래에 대한 정보에 기초하여 다차원 색인 구조(multidimensional index structures)를 형성하는 인덱스 트리(index tree)에 블록 데이터를 노드로 추가한다.

대표도 - 도5



(52) CPC특허분류

G06Q 20/3827 (2013.01)

이 발명을 지원한 국가연구개발사업

과제고유번호 2017-0-00325

부처명 과학기술정보통신부

연구관리전문기관 정보통신기술진흥센터

연구사업명 사물인터넷융합기술개발(R&D)

연구과제명 블록체인 기반 IoT 신뢰성 제어 및 관리 기술 개발

기 여 율 1/1

주관기관 달리웍스 주식회사

연구기간 2018.01.01 ~ 2018.12.31

명세서

청구범위

청구항 1

네트워크를 통하여 블록 데이터를 수신하는 단계;

수신된 상기 블록 데이터 내에 포함된 거래들에 대한 검증을 수행하는 단계; 및

거래 검증시 각 거래에 대한 정보에 기초하여 다차원 색인 구조(multidimensional index structures)를 형성하는 인덱스 트리(index tree)에 상기 블록 데이터를 노드로 추가하는 단계;를 포함하는, 블록체인을 위한 검색 기능의 제공 방법.

청구항 2

제 1 항에 있어서,

상기 다차원 색인 구조의 인덱스 트리는,

PAM(Point Access Method)에 따라 검색 키(key)가 복수 개의 필드로 구성되며, 거래 데이터를 다차원의 점(point)으로 구성하여 디스크에 저장하는, 블록체인을 위한 검색 기능의 제공 방법.

청구항 3

제 2 항에 있어서,

상기 다차원 색인 구조의 인덱스 트리는,

디스크 페이지 크기와 같은 일정한 크기의 노드들로 구성되는 균형 트리(balanced tree)로서, 각 노드는 페이지에 저장되고, 내부 노드(internal node)는 다차원 공간에서 하나의 영역을 담당하고, 리프 노드(leaf node)는 데이터 페이지 정보를 저장하며, 각 거래 데이터는 자신이 속한 공간을 나타내는 색인 노드를 따라 탐색 공간(search space) 내에 위치하는, 블록체인을 위한 검색 기능의 제공 방법.

청구항 4

제 1 항에 있어서,

상기 거래 검증시, 아직 거래가 성립되지 않은 트랜잭션(transaction)을 저장하는 'Mempool'을 업데이트하는 단계;를 더 포함하는, 블록체인을 위한 검색 기능의 제공 방법.

청구항 5

제 1 항에 있어서,

상기 거래 검증이 완료된 후, 거래 정보를 토대로 'Merkle Root'을 계산하고, 블록 헤더(block header)에 대한 검증을 수행하는 단계;를 더 포함하는, 블록체인을 위한 검색 기능의 제공 방법.

청구항 6

제 1 항에 있어서,

상기 거래 검증이 완료된 블록 데이터를 로컬 디스크에 저장하는 단계;를 더 포함하는, 블록체인을 위한 검색 기능의 제공 방법.

청구항 7

수신자가 네트워크를 통하여 송신자로부터 다차원 색인 구조(multidimensional index structures)에 따른 블록 데이터를 수신하는 단계;

상기 수신자가 수신된 상기 블록 데이터 내에 포함된 거래들에 대한 검증을 수행하는 단계; 및

상기 수신자가 거래 검증시 각 거래에 대한 정보에 기초하여 다차원 색인 구조를 형성하는 인덱스 트리(index tree)에 상기 블록 데이터를 노드로 추가하는 단계;를 포함하고,

수신된 상기 블록 데이터는 상기 다차원 색인 구조에 따른 상기 송신자의 인덱스 트리에 저장된 노드의 개수에 관한 정보로부터 생성된 제 1 해시(hash)값을 포함하는, 블록체인을 위한 검색 기능의 제공 방법.

청구항 8

제 7 항에 있어서,

상기 다차원 색인 구조의 인덱스 트리는,

PAM(Point Access Method)에 따라 검색 키(key)가 복수 개의 필드로 구성되며, 거래 데이터를 다차원의 점(point)으로 구성하여 디스크에 저장하는, 블록체인을 위한 검색 기능의 제공 방법.

청구항 9

제 8 항에 있어서,

상기 다차원 색인 구조의 인덱스 트리는,

디스크 페이지 크기와 같은 일정한 크기의 노드들로 구성되는 균형 트리(balanced tree)로서, 'K-D-B Tree'인, 블록체인을 위한 검색 기능의 제공 방법.

청구항 10

제 7 항에 있어서,

상기 수신자가 상기 거래 검증이 완료된 후 추가 생성된 자신의 인덱스 트리에 저장된 내부 노드의 개수를 산출하여 이로부터 제 2 해시값을 생성하는 단계; 및

상기 수신자가 상기 송신자로부터 수신된 상기 블록 데이터 내에 포함된 제 1 해시값과 자신이 생성한 상기 제 2 해시값을 비교함으로써 무결성을 검증하는 단계;를 더 포함하는, 블록체인을 위한 검색 기능의 제공 방법.

청구항 11

제 10 항에 있어서,

상기 제 1 해시값 및 상기 제 2 해시값 각각은,

이전 블록까지 생성된 인덱스 트리에 대한 정보, 현재 입력된 블록 데이터 내의 거래 정보에 대한 무결성을 보장하는 'Merkle Root', 및 현재 입력된 블록 데이터를 통해 인덱스 트리를 구성한 후의 노드의 개수를 이용하여 산출되는, 블록체인을 위한 검색 기능의 제공 방법.

청구항 12

제 10 항에 있어서,

상기 무결성 검증에 성공한 경우, 상기 수신자가 블록 헤더(block header)에 대한 검증을 수행하는 단계; 및

상기 수신자가 상기 블록 데이터를 다른 사용자에게 전파하는 단계;를 더 포함하는, 블록체인을 위한 검색 기능의 제공 방법.

청구항 13

제 7 항에 있어서,

상기 거래 검증이 완료된 후, 거래 정보를 토대로 'Merkle Root'을 계산하는 단계;를 더 포함하는, 블록체인을 위한 검색 기능의 제공 방법.

청구항 14

제 1 항 내지 제 13 항 중에 어느 한 항의 방법을 컴퓨터에서 실행시키기 위한 프로그램을 기록한 컴퓨터로 읽

을 수 있는 기록매체.

청구항 15

블록체인을 위한 사용자 기기에 있어서,

네트워크를 통하여 블록 데이터를 송수신하는 통신부;

상기 통신부를 통해 수신된 상기 블록 데이터 내에 포함된 거래들에 대한 검증을 수행하고, 거래 검증시 각 거래에 대한 정보에 기초하여 다차원 색인 구조(multidimensional index structures)를 형성하는 인덱스 트리(index tree)에 상기 블록 데이터를 노드로 추가하는 처리부; 및

상기 인덱스 트리 및 상기 블록 데이터를 저장하는 저장부;를 포함하는, 사용자 기기.

청구항 16

제 15 항에 있어서,

상기 다차원 색인 구조의 인덱스 트리는,

PAM(Point Access Method)에 따라 검색 키(key)가 복수 개의 필드로 구성되며, 거래 데이터를 다차원의 점(point)으로 구성하여 디스크에 저장하는, 사용자 기기.

청구항 17

제 16 항에 있어서,

상기 다차원 색인 구조의 인덱스 트리는,

디스크 페이지 크기와 같은 일정한 크기의 노드들로 구성되는 균형 트리(balanced tree)로서, 'K-D-B Tree'인, 사용자 기기.

청구항 18

제 15 항에 있어서,

상기 통신부를 통해 수신된 상기 블록 데이터는 다차원 색인 구조에 따른 송신자의 인덱스 트리에 저장된 노드의 개수에 관한 정보로부터 생성된 제 1 해시(hash)값을 포함하며,

상기 처리부는,

상기 거래 검증이 완료된 후 추가 생성된 자신의 인덱스 트리에 저장된 내부 노드의 개수를 산출하여 이로부터 제 2 해시값을 생성하고,

상기 송신자로부터 수신된 상기 블록 데이터 내에 포함된 제 1 해시값과 자신이 생성한 상기 제 2 해시값을 비교함으로써 무결성을 검증하는, 사용자 기기.

청구항 19

제 18 항에 있어서,

상기 제 1 해시값 및 상기 제 2 해시값 각각은,

이전 블록까지 생성된 인덱스 트리에 대한 정보, 현재 입력된 블록 데이터 내의 거래 정보에 대한 무결성을 보장하는 'Merkle Root', 및 현재 입력된 블록 데이터를 통해 인덱스 트리를 구성한 후의 노드의 개수를 이용하여 산출되는, 사용자 기기.

청구항 20

제 15 항에 있어서,

상기 무결성 검증에 성공한 경우, 상기 처리부는,

블록 헤더(block header)에 대한 검증을 수행하고,

상기 블록 데이터를 다른 사용자에게 전파하는, 사용자 기기.

발명의 설명

기술 분야

[0001] 본 발명은 블록체인(blockchain)에 관한 기술로서, 특히 IoT 환경에서의 블록체인 기술에 있어서 검색 기능을 수행할 수 있는 데이터 구조체, 이를 활용한 검색 방법 및 장치에 관한 것이다.

배경 기술

[0002] 블록체인 기술은 2009년 출시된 비트코인을 시작으로 현재까지 많은 발전을 보여주고 있다. 블록체인에 대한 정의는 다양하지만, 대표적으로 이코노미지는 신뢰할 수 없는 환경에서 제3의 인증 기관 없이 신뢰를 만들어 내는 기술이라 정의하였다. 이러한 신뢰를 만들기 위하여 블록체인 기술은 여러 사용자들이 동일한 분산장부(Distributed Ledger)를 독립적으로 관리하여 네트워크와 합의 알고리즘을 통하여 분산 장부의 동일성을 유지시키는 방안을 제공한다. 위와 같은 방식을 통하여 사용자들은 거래와 블록에 대한 이중 보안성과 PKI(Public Key Infrastructure), 해시(hash) 알고리즘 등 기존 보안 기술들을 통하여 네트워크 상에서의 거래에 대한 신뢰성을 높일 수 있다. 실제로 가장 대표적인 블록체인인 비트코인의 네트워크 파워는 슈퍼 컴퓨터 500대에 해당하며, 51% Attack 등의 공격이 성공하기 위해서 공격자는 슈퍼 컴퓨터 251대는 보유하여야 공격에 성공할 가능성을 가지게 되는 것이다. 이렇듯 비트코인 기술은 암호화 화폐의 등장과 함께 전 세계에서 통용할 수 있는 글로벌 화폐로서의 가치를 보여주었지만, 기반 기술인 블록체인 기술은 기존 보안과 신뢰에 사용되던 기술들을 한 단계 더 발전시킬 수 있는 기술로써 가치를 인정받고 있다.

[0003] 현재 많은 기업들은 블록체인의 신뢰성 있는 데이터 관리에 대해 관심을 가지고 기존 사업에 대한 블록체인 적용을 고려 중이다. 기존 디지털 화폐의 기반 기술로 등장했던 블록체인 기술에 의해 금융 산업에 한정되었던 블록체인 기술은 스마트 컨트랙트 기술의 등장과 함께 IoT(Internet of Things), 스마트 시티, 에너지그리드 사업 등 다양한 분야에 적용되고 있다.

[0004] 산업 적용 범위가 넓어짐에 따라 블록체인에 대한 산업 특성에 따른 변화가 필요하다. 이에 기반하여 현재 출시되고 있는 블록체인 플랫폼들은 각 산업 특성에 맞는 장점들을 가지고 있다. 대표적으로 IoT를 위한 블록체인 플랫폼인 Tangle(IOTA)의 경우 블록이 없는 형태로 DAG(Directed Acyclic Graph) 형태로, IoT 마이크로페이먼트를 지원하기 위해 기존 블록체인과는 다른 구조를 가지고 있다. 또한 'Slock.it'은 공유 경제에서의 블록체인 활용을 위하여 삼성전자와 IBM, Linux 재단 등과 함께 이더리움 컴퓨터가 내재된 도어락 개발을 성공하였다. IoT 환경에 대한 블록체인 기술 적용이 활발해짐에 따라 글로벌 테크 기업들인 Foxconn, Bosch 등은 함께 IoT를 위한 블록체인 플랫폼 개발을 위하여 컨소시엄을 구성하는 등 많은 IoT 관련 기업들은 블록체인 기술을 이해하고 도입하기 위한 노력이 진행 중이다.

[0005] 블록체인에서 핵심 기술들 이외에도 중요하게 여겨지는 분야가 바로 검색 기능이다. 블록체인은 기본적으로 자체적으로 검색 기능을 제공하고 있다. 대표적인 블록체인인 비트코인, 이더리움 등은 구글에서 개발한 키 기반의 LevelDB를 이용하여 거래, 블록, 계정 정보에 대한 검색 기능을 제공하고 있다. 이와 더불어 검색 기능을 제공하는 웹 브라우저들이 운영되고 있다. 비트코인은 약 20여개의 검색을 위한 웹 사이트가 운영되고 있으며, 이더리움의 경우 약 5개가 운영되고 있다.

[0006] 산업계에서 활용되기 위한 블록체인 컨소시엄인 'Hyperleger'에서 또한 검색에 대한 요구사항이 반영되고 있다. 'Hyperleger'는 전 세계 155개의 유수의 기업들이 참여하는 컨소시엄으로 기업들 간의 요구사항이 프로젝트로 진행된다. 대표적인 프로젝트로, 오픈 블록체인 플랫폼인 'Fabric'과 이외에도 약 7개의 프로젝트가 진행 중이다. 여기서 진행 중인 프로젝트 중 세 번째로 시작된 것이 바로 'Hyperledger Explorer'로, 블록체인 내부 데이터에 대한 시각화와 검색 기능을 제공하기 위해 시작되었다. 이와 더불어 자체 블록체인인 'Fabric'에 대한 검색 기능 향상에 대해서도 요구되고 있다. 'Fabric' 프로젝트의 발전을 위한 제안(proposal) 중 하나인 'Next-Ledger-Architecture-Proposal'에 따르면 'Fabric'에 대한 요구사항으로 'Point in time queries', 'SQL like query language'를 뽑으며 기존에 제공하지 못했던 다양한 검색 기능 제공이 필요함을 시사하고 있다.

선행기술문헌

비특허문헌

[0007] (비특허문헌 0001) Satoshi Nakamoto, "Bitcoin: A Peer-to-Peer Electronic Cash System", 2008

발명의 내용

해결하려는 과제

[0008] 본 발명이 해결하고자 하는 기술적 과제는, 종래의 비트코인과 같은 블록체인 환경이 통상적인 데이터베이스가 제공하는 다중 조건 검색 내지 범위 검색 기능을 제공하지 못하는 한계를 극복하고, 낮은 수준의 처리량의 문제를 해결하며, 완벽한 검색 결과값의 보장을 위한 신뢰 방안이 부재한 약점을 해소하고자 한다.

과제의 해결 수단

[0009] 상기 기술적 과제를 해결하기 위하여, 본 발명의 일 실시예에 따른 블록체인을 위한 검색 기능의 제공 방법은, 네트워크를 통하여 블록 데이터를 수신하는 단계; 수신된 상기 블록 데이터 내에 포함된 거래들에 대한 검증을 수행하는 단계; 및 거래 검증시 각 거래에 대한 정보에 기초하여 다차원 색인 구조(multidimensional index structures)를 형성하는 인덱스 트리(index tree)에 상기 블록 데이터를 노드로 추가하는 단계;를 포함한다.

[0010] 일 실시예에 따른 블록체인을 위한 검색 기능의 제공 방법에서, 상기 다차원 색인 구조의 인덱스 트리는, PAM(Point Access Method)에 따라 검색 키(key)가 복수 개의 필드로 구성되며, 거래 데이터를 다차원의 점(point)으로 구성하여 디스크에 저장할 수 있다. 또한, 상기 다차원 색인 구조의 인덱스 트리는, 디스크 페이지 크기와 같은 일정한 크기의 노드들로 구성되는 균형 트리(balanced tree)로서, 각 노드는 페이지에 저장되고, 내부 노드(internal node)는 다차원 공간에서 하나의 영역을 담당하고, 리프 노드(leaf node)는 데이터 페이지 정보를 저장하며, 각 거래 데이터는 자신이 속한 공간을 나타내는 색인 노드를 따라 탐색 공간(search space) 내에 위치할 수 있다.

[0011] 일 실시예에 따른 블록체인을 위한 검색 기능의 제공 방법은, 상기 거래 검증시, 아직 거래가 성립되지 않은 트랜잭션(transaction)을 저장하는 'Mempool'을 업데이트하는 단계;를 더 포함할 수 있다.

[0012] 일 실시예에 따른 블록체인을 위한 검색 기능의 제공 방법은, 상기 거래 검증이 완료된 후, 거래 정보를 토대로 'Merkle Root'를 계산하고, 블록 헤더(block header)에 대한 검증을 수행하는 단계;를 더 포함할 수 있다.

[0013] 일 실시예에 따른 블록체인을 위한 검색 기능의 제공 방법은, 상기 거래 검증이 완료된 블록 데이터를 로컬 디스크에 저장하는 단계;를 더 포함할 수 있다.

[0014] 상기 기술적 과제를 해결하기 위하여, 본 발명의 다른 실시예에 따른 블록체인을 위한 검색 기능의 제공 방법은, 수신자가 네트워크를 통하여 송신자로부터 다차원 색인 구조(multidimensional index structures)에 따른 블록 데이터를 수신하는 단계; 상기 수신자가 수신된 상기 블록 데이터 내에 포함된 거래들에 대한 검증을 수행하는 단계; 및 상기 수신자가 거래 검증시 각 거래에 대한 정보에 기초하여 다차원 색인 구조를 형성하는 인덱스 트리(index tree)에 상기 블록 데이터를 노드로 추가하는 단계;를 포함하고, 수신된 상기 블록 데이터는 상기 다차원 색인 구조에 따른 상기 송신자의 인덱스 트리에 저장된 노드의 개수에 관한 정보로부터 생성된 제 1 해시(hash)값을 포함한다.

[0015] 다른 실시예에 따른 블록체인을 위한 검색 기능의 제공 방법에서, 상기 다차원 색인 구조의 인덱스 트리는, PAM(Point Access Method)에 따라 검색 키(key)가 복수 개의 필드로 구성되며, 거래 데이터를 다차원의 점(point)으로 구성하여 디스크에 저장할 수 있다. 또한, 상기 다차원 색인 구조의 인덱스 트리는, 디스크 페이지 크기와 같은 일정한 크기의 노드들로 구성되는 균형 트리(balanced tree)로서, 'K-D-B Tree'인 것이 바람직하다.

[0016] 다른 실시예에 따른 블록체인을 위한 검색 기능의 제공 방법은, 상기 수신자가 상기 거래 검증이 완료된 후 추가 생성된 자신의 인덱스 트리에 저장된 내부 노드의 개수를 산출하여 이로부터 제 2 해시값을 생성하는 단계; 및 상기 수신자가 상기 송신자로부터 수신된 상기 블록 데이터 내에 포함된 제 1 해시값과 자신이 생성한 상기 제 2 해시값을 비교함으로써 무결성을 검증하는 단계;를 더 포함할 수 있다. 이때, 상기 제 1 해시값 및 상기 제 2 해시값 각각은, 이전 블록까지 생성된 인덱스 트리에 대한 정보, 현재 입력된 블록 데이터 내의 거래 정보에 대한 무결성을 보장하는 'Merkle Root', 및 현재 입력된 블록 데이터를 통해 인덱스 트리를 구성한 후의 노드의 개수를 이용하여 산출될 수 있다.

- [0017] 다른 실시예에 따른 블록체인을 위한 검색 기능의 제공 방법은, 상기 무결성 검증에 성공한 경우, 상기 수신자가 블록 헤더(block header)에 대한 검증을 수행하는 단계; 및 상기 수신자가 상기 블록 데이터를 다른 사용자에게 전파하는 단계;를 더 포함할 수 있다.
- [0018] 다른 실시예에 따른 블록체인을 위한 검색 기능의 제공 방법은, 상기 거래 검증이 완료된 후, 거래 정보를 토대로 'Merkle Root'을 계산하는 단계;를 더 포함할 수 있다.
- [0019] 한편, 이하에서는 상기 기재된 블록체인을 위한 검색 기능의 제공 방법들을 컴퓨터에서 실행시키기 위한 프로그램을 기록한 컴퓨터로 읽을 수 있는 기록매체를 제공한다.
- [0020] 상기 기술적 과제를 해결하기 위하여, 본 발명의 또 다른 실시예에 따른 블록체인을 위한 사용자 기기는, 네트워크를 통하여 블록 데이터를 송수신하는 통신부; 상기 통신부를 통해 수신된 상기 블록 데이터 내에 포함된 거래들에 대한 검증을 수행하고, 거래 검증시 각 거래에 대한 정보에 기초하여 다차원 색인 구조(multidimensional index structures)를 형성하는 인덱스 트리(index tree)에 상기 블록 데이터를 노드로 추가하는 처리부; 및 상기 인덱스 트리 및 상기 블록 데이터를 저장하는 저장부;를 포함한다.
- [0021] 또 다른 실시예에 따른 블록체인을 위한 사용자 기기에서, 상기 다차원 색인 구조의 인덱스 트리는, PAM(Point Access Method)에 따라 검색 키(key)가 복수 개의 필드로 구성되며, 거래 데이터를 다차원의 점(point)으로 구성하여 디스크에 저장할 수 있다. 또한, 상기 다차원 색인 구조의 인덱스 트리는, 디스크 페이지 크기와 같은 일정한 크기의 노드들로 구성되는 균형 트리(balanced tree)로서, 'K-D-B Tree'인 것이 바람직하다.
- [0022] 또 다른 실시예에 따른 블록체인을 위한 사용자 기기에서, 상기 통신부를 통해 수신된 상기 블록 데이터는 다차원 색인 구조에 따른 송신자의 인덱스 트리에 저장된 노드의 개수에 관한 정보로부터 생성된 제 1 해시(hash)값을 포함하며, 상기 처리부는, 상기 거래 검증이 완료된 후 추가 생성된 자신의 인덱스 트리에 저장된 내부 노드의 개수를 산출하여 이로부터 제 2 해시값을 생성하고, 상기 송신자로부터 수신된 상기 블록 데이터 내에 포함된 제 1 해시값과 자신이 생성한 상기 제 2 해시값을 비교함으로써 무결성을 검증할 수 있다. 이때, 상기 제 1 해시값 및 상기 제 2 해시값 각각은, 이전 블록까지 생성된 인덱스 트리에 대한 정보, 현재 입력된 블록 데이터 내의 거래 정보에 대한 무결성을 보장하는 'Merkle Root', 및 현재 입력된 블록 데이터를 통해 인덱스 트리를 구성한 후의 노드의 개수를 이용하여 산출될 수 있다.
- [0023] 또 다른 실시예에 따른 블록체인을 위한 사용자 기기에서, 상기 무결성 검증에 성공한 경우, 상기 처리부는, 블록 헤더(block header)에 대한 검증을 수행하고, 상기 블록 데이터를 다른 사용자에게 전파할 수 있다.

발명의 효과

- [0024] 본 발명의 실시예들은, 다중 검색 기능을 제공하는 트리 인덱스 구조를 채택함으로써 기존 블록체인의 성능 저하 없이 다양한 검색 기능을 제공할 수 있고, 이를 통해 전자화폐 사용자들의 편의성과 기업들의 데이터 활용방안을 향상시킬 수 있으며, 블록체인의 거래 데이터에 대한 신뢰뿐만 아니라 검색 결과에 대한 신뢰 또한 보장할 수 있다.

도면의 간단한 설명

- [0025] 도 1은 IoT 플랫폼(Daliworks)에서의 실제 거래 데이터를 예시한 도면이다.
- 도 2는 비트코인 블록체인 코어와 웹 브라우저의 검색을 위한 API를 예시한 도면이다.
- 도 3은 MongoDB의 다차원 인덱스 관리를 위한 배열 구조체를 도시한 도면이다.
- 도 4는 본 발명의 일 실시예에 따른 검색 구조 및 검색 방식을 도시한 전체 구성도이다.
- 도 5는 본 발명의 일 실시예에 따른 블록체인을 위한 검색 기능의 제공 방법을 도시한 흐름도이다.
- 도 6는 본 발명의 실시예들이 제안하는 블록체인 데이터 처리에 활용될 수 있는 색인 트리인 K-D-B Tree를 예시한 도면이다.
- 도 7은 본 발명의 실시예들에 따른 인덱스 트리의 무결성 보장을 위한 검증 과정을 설명하기 위한 도면이다.
- 도 8은 본 발명의 또 다른 실시예에 따른 블록체인을 위한 검색 기능을 제공하는 장치를 도시한 블록도이다.
- 도 9는 본 발명의 실시예들에 따른 프로토타입을 통해 인덱스 트리 생성에 소요되는 시간을 비교한 실험예이다.

도 10은 본 발명의 실시예들에 따른 프로토타입을 통해 인덱스 트리에 10개 데이터를 추가 삽입하는데 소요된 시간을 비교한 실험예이다.

도 11은 본 발명의 실시예들에 따른 프로토타입을 통해 단일 검색과 범위 검색 시나리오의 성능을 비교한 실험예이다.

도 12은 본 발명의 실시예들에 따른 프로토타입을 통해 다차원 검색 성능을 비교한 실험예이다.

발명을 실시하기 위한 구체적인 내용

- [0026] 이하에서는 도면을 참조하여 본 발명의 실시예들을 구체적으로 설명하도록 한다. 다만, 하기의 설명 및 첨부된 도면에서 본 발명의 요지를 흐릴 수 있는 공지 기능 또는 구성에 대한 상세한 설명은 생략한다. 덧붙여, 명세서 전체에서, 어떤 구성 요소를 '포함'한다는 것은, 특별히 반대되는 기재가 없는 한 다른 구성요소를 제외하는 것이 아니라, 다른 구성요소를 더 포함할 수 있는 것을 의미한다.
- [0027] 또한, 제 1, 제 2 등의 용어는 다양한 구성요소들을 설명하는데 사용될 수 있지만, 상기 구성 요소들은 상기 용어들에 의해 한정되어서는 안 된다. 상기 용어들은 하나의 구성 요소를 다른 구성 요소로부터 구별하는 목적으로 사용될 수 있다. 예를 들어, 본 발명의 권리 범위로부터 이탈되지 않은 채 제 1 구성 요소는 제 2 구성 요소로 명명될 수 있고, 유사하게 제 2 구성 요소도 제 1 구성 요소로 명명될 수 있다.
- [0028] 본 발명에서 사용한 용어는 단지 특정한 실시예를 설명하기 위해 사용된 것으로, 본 발명을 한정하려는 의도가 아니다. 단수의 표현은 문맥상 명백하게 다르게 뜻하지 않는 한, 복수의 표현을 포함한다. 본 출원에서, "포함하다" 또는 "구비하다" 등의 용어는 실시된 특징, 숫자, 단계, 동작, 구성요소, 부분품 또는 이들을 조합한 것이 존재함을 지정하려는 것이지, 하나 또는 그 이상의 다른 특징들이나 숫자, 단계, 동작, 구성요소, 부분품 또는 이들을 조합한 것들의 존재 또는 부가 가능성을 미리 배제하지 않는 것으로 이해되어야 한다.
- [0029] 특별히 다르게 정의되지 않는 한, 기술적이거나 과학적인 용어를 포함해서 여기서 사용되는 모든 용어들은 본 발명이 속하는 기술분야에서 통상의 지식을 가진 자에 의해 일반적으로 이해되는 것과 동일한 의미이다. 일반적으로 사용되는 사전에 정의되어 있는 것과 같은 용어들은 관련 기술의 문맥상 가지는 의미와 일치하는 의미인 것으로 해석되어야 하며, 본 출원에서 명백하게 정의하지 않는 한, 이상적이거나 과도하게 형식적인 의미로 해석되지 않는다.
- [0030] 다양한 검색 기능 제공에 대한 요구사항이 필요로 하는 이유는 바로 다양한 산업 환경에 대한 기술 적용의 문제이다. 기존 비트코인 거래 데이터를 보았을 때, 검색에 활용될 데이터 필드는 송신자 Address, 수신자 Address, value, Timestamp 등 4가지 정도가 있다. 그러나, IoT 플랫폼(Daliworks)에서의 실제 거래 데이터를 예시한 도 1을 참조하면, IoT 환경의 경우 하나의 거래에 포함된 데이터들의 종류가 보다 다양화됨을 알 수 있다. 다양한 데이터들 중 중복되는 정보를 제외하고 크게 6~8개의 데이터 필드로 구성될 수 있으며, 블록체인 상에 적용될 경우 데이터 필드는 추가될 가능성이 높다. 또한 IoT 환경의 경우 기존 하나의 노드를 한 사용자가 관리하는 것과 달리 여러 기기에 대한 관리가 한명의 사용자에게 의해 이루어지며 사용자는 기기 관리에 대한 시각화된 정보가 필요하다.
- [0031] 따라서, 이하에서 제시되는 본 발명의 실시예들에서는 IoT 환경에서 다양한 검색 기능을 수행할 수 있으며, 기존 블록체인 검색의 한계점을 파악하여 이에 대한 해결안을 제시한다.
- [0032] 도 2는 비트코인 블록체인 코어와 웹 브라우저의 검색을 위한 API를 예시한 도면이다. 기존 블록체인의 경우 앞서 언급한 것과 같이 검색 기능을 제공하고 있지만, 기능적 한계점을 가지고 있으며 이러한 한계점에는 블록체인의 구조적인 측면이 원인이 된다. 먼저 도 2를 참조하면, 현재의 블록체인 구조체는 검색에 용이한 형태가 아니다. 블록체인은 이름과 마찬가지로 사용자들에게 전달받은 거래들을 토대로 생성된 블록들이 체인 형태로 연결된 구조이다. 이와 같은 형태는 임의의 거래에 대한 검색을 위해서 블록에 대한 선형 검색과 블록 내부에 대한 선형 검색이 이루어져야 하는 구조임에 따라 검색 성능이 비효율적이다. 기존 검색 연구의 경우, 데이터에 대한 인덱스(Index) 정보를 통하여 검색을 하지만 현재 출시되어 있는 비트코인이나 이더리움의 경우 LevelDB를 통하여 블록 넘버와 거래와 블록의 해시값에 대한 검색을 제공하고 있으며, 인덱스 정보의 경우 블록 넘버를 통해서 블록에 대해서만 관리하며 거래정보에 대한 인덱스를 가지지 않으므로 이전 거래 정보에 대한 검색에 제한적이다. 따라서 현재 비트코인의 경우 대부분의 데이터베이스가 제공하고 있는 다중 조건 검색이나 범위 검색 기능을 제공하고 있지 않다.
- [0033] 또한, 현재의 구조는 블록체인에 대한 정보를 관리하는 데이터베이스와 실제 데이터인 블록체인 데이터와의 연

계성이 떨어진다. 이러한 현상의 원인은 블록체인에서 생성된 블록에 대한 추후 활용도가 떨어지기 때문이다. 뿐만 아니라, 검색 노드에 대한 질의(query) 결과에 대한 무결성 보장 방안이 존재하지 않기 때문에, 질의 결과의 신뢰도가 낮다는 문제가 지적되었다.

[0034] 도 3은 MongoDB의 다차원 인덱스 관리를 위한 배열 구조체를 도시한 도면으로서, 블록체인에서 블록에 대한 파일 생성은 다음과 같은 프로세스를 가지고 있다. 먼저, 사용자는 다른 사용자에게 의해 채굴된 블록 데이터를 전달받는다. 사용자는 블록 데이터에 대한 검증을 LevelDB를 통하여 진행하며 거래와 채굴에 필요한 정보를 데이터베이스에 업데이트한다. 그리고 검증이 완료된 데이터를 파일 형태(비트코인의 경우 .dat)형태로 로컬 디스크에 저장한다. 이렇게 저장된 파일 형태의 블록체인 데이터들은 전체 네트워크의 무결성 보장을 위해서 보관될 뿐 실제로 거래 검증이 이루어지며, 이전 블록 헤더 정보를 가지고 오는 것은 모두 데이터베이스를 통해서 이루어진다. 따라서 파일 형태의 데이터에 대해 사용되는 경우는 오직 검색이 발생할 경우이다. 따라서 검색 기능이 추가 아닌 현재의 블록체인 플랫폼에서는 거래 정보에 대한 인덱스를 가지는 것은 필수적인 요소가 아니다.

[0035] 이러한 노드 자체의 검색에 대한 한계점은 자연스럽게 제3의 검색 기능 제공 기관을 발생시켰지만, 이에 대한 신뢰도 문제가 발생할 수 있다. 현재 비트코인의 경우 약 20여개, 이더리움의 경우 5개의 전문 검색 웹 사이트가 존재한다. 실제로 비트코인 사용자들은 대부분에 대한 거래 정보를 확인을 위하여 웹 사이트를 통해서 검증한다. 검색 웹 사이트는 검색 기능 제공을 위하여 하나의 노드에 저장되어 있는 블록체인 데이터와 네트워크를 통해 전달받은 데이터들에 대한 재가공을 통하여 거래 정보를 제공하고 있다. 하지만, 재가공 중 거래 데이터에 대한 위변조 발생 후 사용자에게 잘못된 거래 정보를 전달하여도 사용자는 이에 대한 검증을 즉시 실행할 수 없다. 또한 위변조를 발견하더라도 거래 정보에 대한 검증을 위해서는 직접 블록체인 데이터에 대한 접근을 통하여 수동으로 비교해야 하는 한계점이 있다. 그리고 현재 검색 웹사이트는 앞서 이야기한 코어 프로그램과 마찬가지로 범위 검색, 다중 조건 검색 기능을 제공하고 있지 않으며, 제공하는 API 또한 초당 3개로 낮은 수준의 처리량을 보여주고 있다.

[0036] "Etherql : A Query Layer for Blockchain System" 연구는 기존 이더리움에서의 검색 기능의 한계점을 해결하고자 LevelDB가 아닌 MongoDB를 활용하여 보다 다양한 검색 기능을 제공하고자 하는 방안이다. 새로운 검색 레이어 추가를 위하여 Sync manager, Handler Chain, Persistence Framework, Developer Interface로 크게 4가지로 구분하여 구현하였다. Sync manager의 경우 이더리움 블록체인에서 다른 노드들과의 동일한 블록체인 데이터 유지하기 위한 기능을 하며, 분기(fork) 발생시 이에 대한 처리를 한다. Handler Chain은 블록 내에 포함된 정보들을 SQL scheme에 맞게 재정의 한다. Persistence Framework은 MongoDB와의 연결을 통하여 사용자의 검색 요청에 따른 질의(Query)를 전송하여 처리한다. 마지막으로 Developer Interface의 경우 Rest API를 통하여 개발자들이 보다 넓은 블록체인 데이터에 대한 처리가 가능하도록 하였다. 연구 결과 일차원 검색 분야에서 LevelDB보다 높은 처리량을 보였으며, 충분한 성능의 범위 검색이 가능함을 보여주었다.

[0037] MongoDB를 활용함에 따라 다양한 검색 기능을 제공하였지만 다중 조건 검색의 시나리오에서 활용되기에는 성능적 한계점이 존재한다. 먼저 MongoDB의 경우 버킷 형태를 가진 B-Tree를 인덱스로 하여 검색 성능을 향상시키고자 하였다. 하지만 B-Tree는 기존 이진 트리와 마찬가지로 일차원의 데이터에 대한 처리에 중점을 두었으며, 2차원 이상의 데이터에 대한 처리를 위해 도 3과 같은 배열 형태를 인덱스 구조체로 가지고 있다. 이와 같은 구성은 다중 조건 처리를 위한 시간 복잡도는 $O(N)$ 으로 성능이 기존 블록체인 데이터에 대한 선형 검색과 다를 바 없다. 따라서 MongoDB는 복잡해지는 검색 시나리오에는 큰 성능 향상을 기대할 수 없는 것이다.

[0038] 또한, 메모리 기반의 인덱스 트리 관리를 통해 성능이 빠른 장점이 있지만, 메모리의 크기가 작은 IoT 기기의 특성에 적합하지 않으며, 블록체인과 같이 삭제없이 늘어나는 특성에 따라 메모리의 크기가 부족한 상황이 발생할 수 있다.

[0039] 그리고, 인덱스 트리과 블록체인 파일 데이터에 대한 연결시 외부 데이터에 대한 주소값이 아닌 인덱스 정보가 저장되어 인덱스 결과값과 주소값을 관리해주는 테이블이 추가로 존재해야 되며, 검색 결과를 가져오기 위해서는 디스크 기반 서비스와 동일하게 디스크 접근이 필요하다.

[0040] 마지막으로 검색 결과에 대한 신뢰성의 문제가 발생할 수 있다. 현재 자체 노드의 데이터베이스를 통하여 얻은 검색 결과값에 대해서는 본인이 이미 한번 검증된 데이터들에 대한 검색이므로 무조건적인 신뢰를 보내고 있지만, 전체 네트워크의 보안성을 장점으로 가지는 블록체인의 특성상 하나의 노드에 대한 보안성까지 보장하지는 않는다. 따라서 완벽한 검색 결과값의 보장을 위한 신뢰 방안이 현재는 부재한 상태이다.

[0041] 이하에서는 상기된 문제점 인식에 기초하여 안출된 다중 조건 검색을 위한 블록체인 검색 모듈을 제안하도록 한다. 따라서, 본 발명의 실시예들에서는 현재 존재하고 있는 블록체인의 큰 구조 변화없이 다양한 검색 기능을 제공하며 이에 대한 무결성 보장방안을 제시한다.

[0042] 먼저 다양한 검색 기능을 제공하기 위해서 데이터베이스가 아닌 거래 정보에 대한 인덱스 트리를 추가하여 검색 기능을 제공하고자 한다. 이러한 방법은 기존 블록체인의 한계점인 거래 정보에 대한 인덱스 관리의 부재 문제를 해결하여 다양한 검색 시나리오 지원이 가능하게 한다. 또한 블록체인 거래 데이터에 대한 무결성 보장 방안을 추가하여 사용자가 검색 요청 결과에 대한 신뢰도를 보장할 수 있도록 한다. 마지막으로 본 발명의 결과물인 추가적인 검색 모듈 장착 후에도 기존 시스템에 비하여 성능의 하락이 발생하지 않아야 한다.

[0044] (1) 검색 모듈 아키텍처

[0045] 기존 블록체인의 구조적 변화없이 검색 레이어를 추가할 수 있는 이유는 현재 구조가 기능별로 분리되었기 때문이다. 일반적으로 블록체인 구조체는 데이터 레이어, 네트워크 레이어, 합의(Consensus) 레이어, 암호화 레이어 등으로 구분되어 있다. 이 중 검색과 블록체인 파일 데이터와 직접적인 연관이 있는 것은 데이터와 네트워크 레이어이다. 데이터 레이어란 데이터베이스에 의해 검증된 블록데이터를 파일 형태로 저장하거나, 검증된 거래를 Mempool(거래임시저장소)에 기록해주는 역할을 한다. 네트워크 레이어는 다른 노드와의 연결을 유지하며 블록 또는 거래 데이터들을 전달하거나 전송받으며 데이터를 데이터베이스에 전달해주는 역할을 한다. 기존 네트워크 레이어와 데이터 레이어 사이에서의 처리를 현재는 데이터베이스(LevelDB)를 통해서만 진행하였다면 인덱스 트리를 추가하여 도 4와 같은 구조로 다양한 검색 기능을 제공할 수 있다.

[0046] 도 4는 본 발명의 일 실시예에 따른 검색 구조 및 검색 방식을 도시한 전체 구성도로서, 기존 블록체인의 데이터 처리 과정의 큰 변화없이 인덱스 처리를 보여주게 된다.

[0047] STEP 1: 네트워크 레이어를 통하여 다른 사용자에게 의해 생성(채굴)된 블록 데이터를 전송 받는다.

[0048] STEP 2: 블록 내에 포함된 거래들에 대한 검증을 수행한다.

[0049] STEP 3: 거래 검증 시 이용되는 Mempool에 대한 Update를 수행한다.

[0050] STEP 4: 거래 검증 시 각 거래에 대한 정보를 토대로 인덱스 트리에 노드로 추가한다.

[0051] STEP 5: 거래 검증이 끝난 후 거래 정보를 토대로 Merkle Root를 계산하고 이와 더불어 블록 헤더에 대한 검증을 수행한다.

[0052] STEP 6: 데이터 레이어를 통하여 검증이 끝난 블록 데이터를 자신의 로컬 디스크에 저장한다.

[0053] 위와 같은 검증 과정에서 실제로 인덱스 처리를 위해 추가된 과정은 STEP 4번 항목뿐이다. 따라서 기존 처리 과정에서 큰 오버헤드 없이 처리가 가능함을 보여준다. 또한, 검증시 사용된 데이터들을 토대로 인덱스 트리 검증에 활용하여 신뢰도 상승과 시스템 성능 유지를 가능하도록 하였다.

[0054] 이는 기존 블록체인의 한계점이던 블록체인 파일 데이터와 데이터베이스와의 연결성을 확보하여 주며 존재하던 데이터 레이어 또는 네트워크 레이어의 구조적이거나 기능적인 변화 없이 구현이 가능하다는 장점이 있다.

[0055] 또한, 사용자의 선택에 따른 검색 성능 보장을 위하여 검색 시나리오에 따른 인덱스 트리의 재구성이 가능하도록 한다. 이는 대용량의 블록체인 데이터들을 인덱스 트리로 재구성하는데 소요되는 시간이 상당하기 때문에 초기 사용자가 원하는 시나리오에 적합한 인덱스 트리로 설정하는 것이다. 특히, 다중 조건 검색의 특성상 인덱스 트리에 저장되는 키의 값, 저장되는 키의 순서에 따라 다른 성능을 보여줄 수 있다. 예를 들어, 사용자가 시간대와 온도 값에 대한 검색 기능이 필요할 때, 인덱스 트리를 수신자 주소, 송신자 주소, 습도 값, 기기 관리자, 시간, 온도 값으로 구성할 경우 사용자가 원하는 데이터를 찾기 위하여서는 인덱스 트리의 가장 깊은 곳까지 내려가야 원하는 데이터 필드에 대한 검색이 가능하게 되는 것이다.

[0056]

[0057] (2) 인덱스 트리 구성

[0058] 앞서 언급한 것과 같이 다중 조건 검색 시나리오를 처리하기 위한 인덱스 트리 구성이 필요하다. 본 발명의 일 실시예에 따르면 인덱스 트리의 구조는 'K-D-B 트리'를 통해 이루어지는 것이 가장 적합하다. K-D-B 트리는 기

존 데이터베이스의 다차원 색인 연구 중 블록체인 데이터 관리에 가장 적합한 형태이다. 다차원 색인(Multidimensional Access Method) 연구는 크게 PAM(Point Access Method)와 SAM(Spatial Access Method)로 구분된다. PAM은 블록체인 거래 정보와 같이 점으로 이루어진 거래 데이터들이 데이터베이스에 저장되었을 때 이용되는 방법이며 SAM의 경우 다양한 도형, 지형 정보 등에 대한 데이터 처리에 이용된다. PAM은 크게 4가지 기술인 K-D Tree, QuadTree, Grid File, K-D-B Tree로 구분할 수 있다.

- [0059] 본 발명의 실시예들을 구현하는 과정에서, 대표적인 4가지 기술 중 블록체인 데이터 처리에 가장 적합한 형태는 K-D-B Tree임을 발견하였다. K-D-B Tree는 K-D Tree와 B-Tree의 특성을 조합한 구조체이다. 대용량 블록체인 데이터를 처리하기에 적합하며, 특히 IoT 센서기기는 점차 쌓여가는 데이터를 처리하기에 메모리 용량의 한계점이 존재한다는 점에 주목하였다. 따라서 보다 저렴하게 많은 용량을 사용할 수 있는 디스크 기반의 인덱스 트리 사용이 필수적이다. 또한 K-D Tree의 경우 메모리 기반의 특성과 비롯하여 데이터에 따라 한쪽 트리에 데이터가 몰리는 상황이 발생할 수 있으며 이와 같은 경우 관리해야 하는 트리의 깊이(Depth)와 성능 하락이 발생할 수 있다. 따라서 디스크 기반의 다차원 색인 기술인 QuadTree, Grid File, K-D-B Tree 중에 인덱스 트리가 선정되어야 한다. QuadTree의 경우 4차원의 트리 형태로 구성하여 SE, SW, NE, NW으로 네 방향의 트리 구성이 가능하다. Grid File의 경우 K-D 트리와 동일하게 영역 구분을 통한 인덱스를 구성하지만, 해시 연산을 통하여 인덱스에 대한 접근이 이루어진다. 본 발명의 일 실시예에 따르면 위 3가지 기술 중 3차원 이상의 데이터에 대한 처리 성능이 가장 뛰어난 것은 K-D-B Tree임을 확인하였다. 따라서 블록체인 데이터 처리에 가장 적절한 색인 트리로서, K-D-B Tree를 선택하였다.
- [0060] 도 5는 본 발명의 일 실시예에 따른 블록체인을 위한 검색 기능의 제공 방법을 도시한 흐름도로서, 도 4의 검색 기능 제공 과정을 개괄적으로 재구성하였다.
- [0061] S510 단계에서, 블록체인을 위한 검색 기능의 제공 장치는, 네트워크를 통하여 블록 데이터를 수신한다.
- [0062] S520 단계에서, 상기 장치는, S510 단계를 통해 수신된 상기 블록 데이터 내에 포함된 거래들에 대한 검증을 수행한다. 이후, 상기 거래 검증시, 아직 거래가 성립되지 않은 트랜잭션(transaction)을 저장하는 'Mempool'을 업데이트하는 과정을 더 수행할 수 있다.
- [0063] S530 단계에서, 상기 장치는, 거래 검증시 각 거래에 대한 정보에 기초하여 다차원 색인 구조(multidimensional index structures)를 형성하는 인덱스 트리(index tree)에 상기 블록 데이터를 노드로 추가한다.
- [0064] 이때, 상기 다차원 색인 구조의 인덱스 트리는, PAM(Point Access Method)에 따라 검색 키(key)가 복수 개의 필드로 구성되며, 거래 데이터를 다차원의 점(point)으로 구성하여 디스크에 저장할 수 있다. 특히, 상기 다차원 색인 구조의 인덱스 트리는, 디스크 페이지 크기와 같은 일정한 크기의 노드들로 구성되는 균형 트리(balanced tree)로서, 각 노드는 페이지에 저장되고, 내부 노드(internal node)는 다차원 공간에서 하나의 영역을 담당하고, 리프 노드(leaf node)는 데이터 페이지 정보를 저장하며, 각 거래 데이터는 자신이 속한 공간을 나타내는 색인 노드를 따라 탐색 공간(search space) 내에 위치하는, 데이터 구조체인 것이 바람직하다. 이와 같이 각 노드를 페이지에 저장하는 것은 입출력의 효율을 위한 것으로, 탐색 공간은, 하나의 필드의 도메인 값을 비교한 결과에 따라 두 개의 소공간(subspace)으로 분할될 수 있다. 이러한 다차원 색인 구조의 인덱스 트리로서, 예를 들어, 'K-D-B Tree'를 선택하여 활용할 수 있다.
- [0065] 한편, 상기 거래 검증이 완료된 후, 거래 정보를 토대로 'Merkle Root'를 계산하고, 블록 헤더(block header)에 대한 검증을 수행할 수 있다. 그런 다음, 상기 거래 검증이 완료된 블록 데이터를 로컬 디스크에 저장할 수 있다.
- [0066] 도 6는 본 발명의 실시예들이 제안하는 블록체인 데이터 처리에 활용될 수 있는 색인 트리인 K-D-B Tree를 예시한 도면이다. 인덱스 트리의 주 역할은 앞서 말한 것과 같이 블록체인 파일 데이터 내에 포함된 거래 정보에 대한 인덱스를 쉽게 전달해주는 것이다. 그렇기 위해서 K-D-B Tree의 노드 구조인 (Key, Value)에 저장되는 값을 정의하여야 한다. Key의 경우 앞서 언급한 것과 같이 시나리오에 따른 적합한 키값과 순서가 정의된다. Value의 경우 현재 존재하고 있는 블록체인 파일 데이터 내의 거래 정보 위치 값에 대해 저장한다.
- [0067] 이와 같은 인덱스 트리 생성에 걸리는 시간은 초기 생성시 N 개의 데이터 대해서 $O(N \log N)$ 의 복잡도를 가지며, 이는 Median of medians 알고리즘을 활용할 경우이다. K차원인 하나의 데이터에 대한 삽입의 경우 $O(\log N)$, 검색의 경우 $O(N^{1-1/k} + M)$ 의 복잡도를 가진다.

[0069] (3) 검색 데이터 무결성 보장 방안

[0070] 블록체인 기술은 자신이 가진 데이터가 모든 네트워크 사용자의 데이터와 동일하게 유지시키는 것이다. 따라서 다른 사용자에게 전송받은 데이터를 저장되어 있는 분산장부(블록체인 파일 데이터)를 통해서 스스로 검증이 가능하며, 검증의 결과에 대해 신뢰할 수 있는 것이다.

[0071] 검색의 경우 자신이 가진 블록체인 데이터에 대해 요청을 보내고 이에 대한 결과 값을 돌려받는 형태이다. 따라서 위에서 언급한 것과 같이 블록체인 기술을 통하여 저장된 블록체인 파일 데이터에 대해서는 기본적으로 신뢰를 할 수 있다고 본다. 결론적으로 검색 요청과 반환 중인 값에 대한 위변조가 발생하지 않는다면 보안성을 유지한 채 검색 기능을 제공할 수 있다. 기존 검색 기능을 제공하는 웹 사이트의 경우 이 두 가지에 대한 신뢰성을 가질 수 없음으로 한계점이 존재한다고 언급하였다.

[0072] 그렇다면 본 발명의 실시예들에서 제안한 구조체에서 위변조 방지를 위해서 중점적으로 관리해야 하는 것은 검색 요청을 처리해주는 인덱스 트리이다. 기존 블록체인의 보안성 유지 방안과 같이 다른 사용자와 동일한 인덱스 트리를 유지한다면 검색의 신뢰성을 보장할 수 있는 것이다. 하지만 다양한 시나리오에 따른 사용자의 인덱스 트리의 구성이 달라지게 되며, 최종적으로 동일한 형태가 아닌 인덱스 트리 간에 일지성을 확보하여야 한다.

[0073] 서로 다른 인덱스 트리지만 동일한 입력값(거래 정보)을 가지고 있으므로, 동일한 결과값이 나온다면 내부에 저장된 데이터들이 동일하다고 볼 수 있다. 서로 다른 인덱스 트리지만 입력값이 동일할 경우 동일한 결과값이 나오는 유일한 경우는 인덱스 트리에 저장된 노드의 개수이다. 사용자가 블록체인에 저장된 5000개의 거래를 인덱스 트리 구성할 경우 내부가 어떤 구조의 트리 형태로 되었는지 5000개의 내부 leaf 노드가 생성될 것이다. 따라서 이러한 정보를 토대로 인덱스 트리에 대한 무결성을 보장하는 다음의 수학적 식 1을 구할 수 있다.

수학적 식 1

$Index_Tree_Hash =$

[0074] $Hash(Previous_Index_Tree_Hash + Merkle\ Root + Tree's\ node\ count)$

[0075] Previous_Index_Tree_Hash란 이전 블록까지 생성된 인덱스 트리에 대한 정보이다. Merkle Root의 경우 거래 입력값에 대한 무결성을 보장하는 값으로 현재 입력받은 블록 데이터 내에 대한 거래 정보가 무결성 하다는 것을 보장한다. 이는 기존 블록체인 검증 프로세스에 의해 연산된 값이다. Tree's Node Count는 입력받은 블록 데이터를 통하여 인덱스 트리를 구성한 뒤 나오는 값으로 올바른 인덱스 트리 생성을 보장한다. 최종적으로 Index_Tree_Hash는 기존 블록 헤더 정보에 저장되며 모든 사용자가 Index_Tree_Hash 값을 통하여 동일한 인덱스 트리를 가지고 있음을 확인할 수 있다.

[0076] 위와 같은 연산은 기존 블록체인의 연산 메커니즘과 마찬가지로 모든 사용자들이 동일하게 연산하며 이를 통해 전체 네트워크 사용자들이 하나의 인덱스 트리 정보를 유지할 수 있다.

[0077] 인덱스 트리 구성에 대한 네트워크 관점에서의 처리 프로세스는 다음과 같다.

[0078] STEP 1: 블록 생성자(채굴자)는 미검증 거래들을 임시 블록에 포함 시키고 이를 토대로 자신의 인덱스 트리를 업데이트 하고 수학적 식 1을 포함한 기존 블록체인 채굴 연산을 통해 성공한 블록을 네트워크에 전파한다.

[0079] STEP 2: 채굴된 블록을 전달받은 다른 네트워크 참여자들은 블록에 대한 검증을 수행한다.

[0080] STEP 3: 블록 내에 포함된 각 거래에 대한 검증 성공 시 인덱스 트리에 삽입하고, 최종 거래 검증 후 Merkle_RootUser를 기록한다.

[0081] STEP 4: 거래 검증이 끝난 뒤 추가 생성된 인덱스 트리에 저장된 내부 노드 개수(Tree_Node's_CountUser)를 반환받는다.

[0082] STEP 5: 이를 토대로 다음과 같은 수식 연산을 통하여 채굴된 블록과 자신의 기존 데이터와의 무결성을 확인한다.

수학식 2

$Index_Tree_HashMiner =$

$$Hash(Previous_Index_Tree_HashUser + Merkle\ RootUser + Tree's\ node\ countUser)$$

[0083]

[0084]

즉, 블록 생성자 등의 송신자는 앞서 제시한 수학식 1에 따라 인덱스 트리의 정보를 블록 생성에 포함시켜 해당 블록 데이터를 전송하고, 수신자는 송신자로부터 전송받은 블록 내에 포함된 거래 정보를 통해 인덱스 트리를 업데이트하며, 업데이트 후 생성된 자신의 인덱스의 트리의 노드의 수를 반환한다. 결과적으로 수학식 1의 해시값과 수학식 2의 해시값이 일치하는지 여부를 검사함으로써, 수신자 자신이 생성한 인덱스 트리의 무결성을 확인할 수 있다. 만약, 양자가 일치하지 않는다면, 생성된 인덱스를 트리를 기존의 인덱스 트리으로 재변환할 수 있을 것이다.

[0085]

STEP 6: 연산 결과 동일한 Index_Tree_Hash가 반환될 경우, 전체 블록 헤더에 대한 검증을 수행한 뒤 블록을 타 사용자에게 전파한다.

[0086]

위의 과정에 대한 예시는 도 7과 같으며, 도 7에는 본 발명의 실시예들에 따른 인덱스 트리의 무결성 보장을 위한 일련의 검증 과정을 도시하였다.

[0087]

무결성 보장을 위한 블록체인 검색 기능의 제공 방법에 관한 일련의 처리 과정을 요약하면 다음과 같다.

[0088]

우선, 수신자는 네트워크를 통하여 송신자로부터 다차원 색인 구조(multidimensional index structures)에 따른 블록 데이터를 수신한다. 이때, 수신된 상기 블록 데이터는 상기 다차원 색인 구조에 따른 상기 송신자의 인덱스 트리에 저장된 노드의 개수에 관한 정보로부터 생성된 제 1 해시(hash)값을 포함한다.

[0089]

다음으로, 상기 수신자는 수신된 상기 블록 데이터 내에 포함된 거래들에 대한 검증을 수행하고, 거래 검증시 각 거래에 대한 정보에 기초하여 다차원 색인 구조를 형성하는 인덱스 트리(index tree)에 상기 블록 데이터를 노드로 추가한다. 물론, 상기 다차원 색인 구조의 인덱스 트리는, PAM(Point Access Method)에 따라 검색 키(key)가 복수 개의 필드로 구성되며, 거래 데이터를 다차원의 점(point)으로 구성하여 디스크에 저장할 수 있다. 특히, 상기 다차원 색인 구조의 인덱스 트리는, 디스크 페이지 크기와 같은 일정한 크기의 노드들로 구성되는 균형 트리(balanced tree)로서, 'K-D-B Tree'인 것이 바람직하다

[0090]

또한, 상기 수신자는 상기 거래 검증이 완료된 후 추가 생성된 자신의 인덱스 트리에 저장된 내부 노드의 개수를 산출하여 이로부터 제 2 해시값을 생성하고, 상기 송신자로부터 수신된 상기 블록 데이터 내에 포함된 제 1 해시값과 자신이 생성한 상기 제 2 해시값을 비교함으로써 무결성을 검증할 수 있다.

[0091]

이때, 상기 제 1 해시값 및 상기 제 2 해시값 각각은, 이전 블록까지 생성된 인덱스 트리에 대한 정보, 현재 입력된 블록 데이터 내의 거래 정보에 대한 무결성을 보장하는 'Merkle Root', 및 현재 입력된 블록 데이터를 통해 인덱스 트리를 구성한 후의 노드의 개수를 이용하여 산출될 수 있다.

[0092]

한편, 상기 무결성 검증에 성공한 경우, 상기 수신자는 블록 헤더(block header)에 대한 검증을 수행하고, 상기 블록 데이터를 다른 사용자에게 전파하게 된다. 또한, 상기 거래 검증이 완료된 후, 거래 정보를 토대로 'Merkle Root'를 계산하는 과정을 더 포함할 수 있다.

[0093]

도 8은 본 발명의 또 다른 실시예에 따른 블록체인을 위한 검색 기능을 제공하는 장치를 도시한 블록도로서, 앞서 기술한 도 5의 과정을 하드웨어의 구성 관점에서 재구성한 것이다. 따라서, 여기서는 설명의 중복을 피하고자 각 구성의 기능만을 요약하도록 한다.

[0094]

블록체인을 위한 사용자 기기(20)는, 통신부(21), 처리부(23) 및 저장부(25)를 포함하며, 네트워크(40)를 통해 송신자(10) 및 다른 사용자들(30)과 연결될 수 있다.

[0095]

통신부(21)는, 네트워크(40)를 통하여 블록 데이터를 송수신하는 구성이다.

[0096]

처리부(23)는, 상기 통신부(21)를 통해 수신된 상기 블록 데이터 내에 포함된 거래들에 대한 검증을 수행하고, 거래 검증시 각 거래에 대한 정보에 기초하여 다차원 색인 구조(multidimensional index structures)를 형성하는 인덱스 트리(index tree)에 상기 블록 데이터를 노드로 추가한다. 여기서, 상기 다차원 색인 구조의 인덱스 트리는, PAM(Point Access Method)에 따라 검색 키(key)가 복수 개의 필드로 구성되며, 거래 데이터를 다차원의

점(point)으로 구성하여 디스크에 저장할 수 있다. 특히, 상기 다차원 색인 구조의 인덱스 트리는, 디스크 페이지 크기와 같은 일정한 크기의 노드들로 구성되는 균형 트리(balanced tree)로서, 'K-D-B Tree'인 것이 바람직하다.

[0097] 저장부(25)는, 상기 인덱스 트리 및 상기 블록 데이터를 저장하는 수단이다.

[0098] 상기 통신부(21)를 통해 수신된 상기 블록 데이터는 다차원 색인 구조에 따른 송신자의 인덱스 트리에 저장된 노드의 개수에 관한 정보로부터 생성된 제 1 해시(hash)값을 포함하며, 상기 처리부(23)는, 상기 거래 검증이 완료된 후 추가 생성된 자신의 인덱스 트리에 저장된 내부 노드의 개수를 산출하여 이로부터 제 2 해시값을 생성하고, 상기 송신자(10)로부터 수신된 상기 블록 데이터 내에 포함된 제 1 해시값과 자신이 생성한 상기 제 2 해시값을 비교함으로써 무결성을 검증할 수 있다. 이때, 상기 제 1 해시값 및 상기 제 2 해시값 각각은, 이전 블록까지 생성된 인덱스 트리에 대한 정보, 현재 입력된 블록 데이터 내의 거래 정보에 대한 무결성을 보장하는 'Merkle Root', 및 현재 입력된 블록 데이터를 통해 인덱스 트리를 구성한 후의 노드의 개수를 이용하여 산출될 수 있다.

[0099] 한편, 상기 무결성 검증에 성공한 경우, 상기 처리부(23)는, 블록 헤더(block header)에 대한 검증을 수행하고, 상기 블록 데이터를 다른 사용자(30)에게 전파할 수 있다.

[0101] (4) 실험 및 검증

[0102] 본 발명의 실시예들에서 제안한 것과 같이 기존 블록체인의 큰 구조적 변화 없이 추가된 검색 모듈을 통하여 다양한 검색 기능을 제공할 수 있으며, 이에 대한 성능은 기존 데이터베이스와 큰 차이가 없이 활용이 가능하다는 것을 실험적으로 검증한다.

[0103] 이를 위하여 관련 연구에서 제안한 MongoDB와의 다중 검색 조건, 범위 검색등 다양한 검색 시나리오에 따른 검색 결과 반환에 대한 성능을 비교 평가하며, 또한 인덱스 트리 생성 시간과 MongoDB 생성 시간을 비교하여 기존 블록체인과 비교하여 성능 하락이 발생하지 않음을 실험한다. 특히 다중조건 검색의 경우 앞서 언급한 것과 마찬가지로 K-D-B Tree가 3차원 이상의 검색에 대한 효율성이 가장 높음에 따라 3차원 이상에 대한 시나리오로 진행한다.

[0104] 실험은 i5-6500K, 16g ram, Sandisk 256G SSD, Window 8.1 64bit 의 PC환경에서 진행되었다. MongoDB와 K-D-B Tree에 대한 성능 비교를 위하여 동일한 환경(Node.js)과 프로그래밍 언어(Javascript)로 구현되었다. 수행 성능 평가는 시나리오 따른 검색 쿼리 후 결과값 반환에 걸리는 시간을 평가하였으며, 실험 데이터는 실제 IoT 플랫폼인 Daliworks에서 제공하는 데이터 구성에 임의로 생성된 Data 값을 추가하여 150,000개의 거래 데이터를 활용하였다. 또한 실험을 위해 다음과 같은 제한점을 두었다. 블록체인 분기로 인한 데이터의 변화를 막고자 분기가 발생하지 않는 DPoS(Delegated Proof of Stake) 합의 알고리즘을 사용한다는 가정하에 진행한다. 이를 통하여 K-D-B Tree에서 인덱스 트리의 삭제(Deletion) 수행은 고려하지 않는다.

[0105] 시나리오에 따른 다중 조건 검색 요청의 예시는 다음과 같다.

[0106] 사업장에 IoT 기기들을 설치하고 이를 통해 데이터를 관리하는 기업 A는 고객들의 요구사항에 따라 일정 조건을 벗어나는 값을 발견하면 고객에게 경고 알람을 해주는 기업이다. IoT 기기들의 관리의 편의성을 위하여 게이트웨이 기기를 통하여 IoT 센서 기기 100대를 관리하고 이러한 게이트웨이 간의 블록체인 네트워크 구성을 통하여 데이터의 신뢰성을 확보하는 기술을 가지고 있다.

[0107] 고객의 일정 조건을 벗어나는 거래에 대한 발견을 위하여 게이트웨이 노드들은 생성된 블록체인 내 데이터들을 주기적으로 검증한다. 대표적인 예시로 "IoT 기기 X가 이를 담당하는 게이트웨이 B에게 보낸 거래 중 온도 값이 1.9도 이상이 되는 거래를 모두 검색하여 반환한다." 라는 검색 시나리오를 작성하고 이를 검색 모듈을 통하여 검색할 수 있어야 한다. 이를 본 발명의 실시예들에서 실험한 방안으로 표현하자면 아래와 같다.

[0108] - K-D-B Tree

[0109] Ex) kdb.query([X의 주소], [Y의 주소], [1.9,])

[0110] - MongoDB

[0111] Ex) db.collection("tree").find({'sender_address': X의 주소, 'receiver_address': Y의 주소, 'temp': {

'\$gt': 1.9 }}

- [0112] 이하에서는 기존 블록체인 관련 검색 연구의 기반 기술인 MongoDB와 K-D-B Tree의 생성과 검색 성능을 비교 분석하였다. 이를 위하여 MongoDB와 K-D-B Tree에 150,000 개의 거래 데이터를 생성하여 각 테이블을 구성하였으며, 이와 더불어 추가 블록 삽입을 통한 재구성을 실험하였다. 또한 다양한 검색 시나리오에 대한 검색 결과 반환 시간을 평가하여 본 발명의 실시예들의 검증에 진행하였다.
- [0113] 앞서 말한 것과 같이 본 발명의 실시예들의 최종적인 목적은 기존 블록체인 시스템의 성능 변화 없이 다양한 블록체인 검색 시나리오를 제공하는 것이다. 따라서 기존 블록체인 시스템에서 이용되는 LevelDB보다 성능이 뛰어난 것으로 검증된 MongoDB와의 다양한 블록체인 검색 시나리오를 동일하게 비교 평가 후 목적에 맞는 결과를 예상한다.
- [0114] 먼저 블록체인 초기 참여 시 다른 노드와의 동기화를 통해 기존 블록체인 데이터에 대한 전송을 받는 경우를 대비한 실험이다. 이러한 경우 많은 데이터를 처리해야 하며, 일반적으로 비트코인의 경우 약 2억 6천개의 거래가 존재하며 이를 동기화 하는데 걸리는 시간은 4~5일(432,000 초)정도 걸리는 것으로 알려져 있다. 이는 LevelDB에서의 거래 데이터에 대한 처리를 포함한 시간이다.
- [0115] 도 9은 본 발명의 실시예들에 따른 프로토타입을 통해 인덱스 트리 생성에 소요되는 시간을 비교한 실험예이다. 도 9에서 보이는 것과 같이 기존 블록체인 생성에 걸리는 시간(6588 msec)보다 작은 시간(641 msec)안에 K-D-B Tree 생성이 가능하다는 것을 보여준다. 도 10은 본 발명의 실시예들에 따른 프로토타입을 통해 인덱스 트리 10개 데이터를 추가 삽입하는데 소요된 시간을 비교한 실험예로서, 초기 블록체인 데이터 동기화 후 추가적으로 생성되는 블록에 대한 업데이트에 대한 경우도 도 10과 같이 동일한 결과를 보여준다. 위와 같이 블록체인 데이터 동기화 및 업데이트에 대해서 월등한 성능향상을 보여주고 있다. 이를 통해 기존 블록체인의 동기화에 걸리는 시간의 문제점을 일정 부분 해결할 수 있는 방안으로 적용될 수 있을 것이다.
- [0116] 다음으로, 본 발명의 실시예들의 핵심 주제인 다양한 검색 시나리오에 대한 실험 결과이다. 먼저 기존 블록체인이 제공하는 단일 검색과 관련 연구에서 실험한 범위 검색 시나리오에 대한 실험 결과이다.
- [0117] 도 11는 본 발명의 실시예들에 따른 프로토타입을 통해 단일 검색과 범위 검색 시나리오의 성능을 비교한 실험예이다. 도 11에서 보이는 것과 같이 단일 검색 시나리오의 경우 거래의 해시값을 통한 실험으로 K-D-B Tree가 MongoDB보다 성능이 떨어짐을 알 수 있다. 이는 MongoDB에서 단일 검색시 이용 되는 B-Tree인덱스 트리가 K-D-B Tree보다 단일 검색에 용이한 형태로 되어 있기 때문이다. B-Tree의 경우 검색에 소요되는 시간 복잡도는 $O(\log N)$ 이며 1차원 데이터에 대한 K-D-B Tree의 검색 시간 복잡도는 $O(N)$ 이므로 당연한 결과를 보여준다. 하지만 범위 검색의 경우 위에서 보이는 것과 같이 동일한 요청에 대하여 비슷한 수준의 결과를 보여주고 있음을 알 수 있다.
- [0118] 마지막으로 K-D-B Tree의 특성이 가장 잘 드러날 수 있는 다차원 검색 시나리오에 대한 결과이다. 도 12은 본 발명의 실시예들에 따른 프로토타입을 통해 다차원 검색 성능을 비교한 실험예이다. 도 12에서 보이는 것과 같이 다차원 검색에서의 결과는 크게 차이나는 것을 확인할 수 있다. 이는 MongoDB에서의 장점인 메모리 기반의 인덱스 트리 구성이 되지 않기 때문에 기존보다 성능이 확인할 수 있다. 또한 MongoDB의 경우 다차원 검색에 대하여 앞서 언급한 것과 같이 배열 형태의 인덱스 트리를 구성하게되어 성능이 하락하게 된다. 하지만 K-D-B Tree의 경우 다차원 검색에 대한 범위를 검색 단위마다 줄여나가면서 검색 속도의 향상을 보여주게 된다. 특징점으로는 4차원과 5차원 데이터에 대한 실험의 경우, 4차원에서 이용되는 검색 요청에 추가하여 5차원 데이터를 구성하였는데 4차원에서 검색이 완료되어 5차원과의 성능 변화가 없음을 두 가지 실험 방안 모두에서 확인할 수 있었다. 이를 통하여 실험에 이용되는 키의 순서 변화에 따라 검색의 성능이 변화할 수 있으며 검색에 사용되는 키의 값의 정의가 중요하다는 점을 알 수 있다.
- [0119] 실험 결과에 따라 다양한 검색 시나리오에 따라 검색의 성능이 차이가 존재함을 알 수 있다. 따라서 블록체인 개발자들은 자신의 사업에 적합한 데이터베이스 또는 인덱스 트리를 선택할 수 있다. 결과적으로 보았을 때 기존 연구와 본 발명의 실시예들과의 차이점에 대해서 표 1과 같은 결과를 얻을 수 있다.

표 1

	LevelDB	MongoDB	본 제안 방안
Index 방법	Key-Value Map	B-Tree	K-D-B Tree
단일 검색 성능	O	O	▲
범위 검색 성능	X (지원 하지 않음)	O	O
다중 조건 검색 성능	X (지원 하지 않음)	▲	O
지원 언어	Python, Javascript, C++	Javascript, C++, Python	Javascript
디스크 기반 여부	O	X (메모리 기반)	O

[0120]

[0121]

실험을 통하여 검증된 것처럼 본 발명의 실시예들에서 제안한 K-D-B Tree는 기존 블록체인의 성능 저항 없이 다양한 검색 기능을 제공할 수 있다. 이러한 점을 통하여 비트코인, 이더리움 등 사용자들의 편의성을 높일 수 있으며 기업들의 데이터 활용방안을 높일 수 있다. 또한 개발자들은 이를 통하여 블록체인 점차 쌓이는 데이터를 통하여 다양한 사업을 할 수 있는 기회 제공이 가능하다. 그리고 블록체인 거래 데이터가 신뢰되는 것과 마찬가지로 기존에 없던 검색 결과에 대한 신뢰도 또한 보장되는 장점이 존재한다.

[0122]

한편, 본 발명의 실시예들은 컴퓨터로 읽을 수 있는 기록 매체에 컴퓨터가 읽을 수 있는 코드로 구현하는 것이 가능하다. 컴퓨터가 읽을 수 있는 기록 매체는 컴퓨터 시스템에 의하여 읽혀질 수 있는 데이터가 저장되는 모든 종류의 기록 장치를 포함한다.

[0123]

컴퓨터가 읽을 수 있는 기록 매체의 예로는 ROM, RAM, CD-ROM, 자기 테이프, 플로피디스크, 광 데이터 저장장치 등을 포함한다. 또한, 컴퓨터가 읽을 수 있는 기록 매체는 네트워크로 연결된 컴퓨터 시스템에 분산되어, 분산 방식으로 컴퓨터가 읽을 수 있는 코드가 저장되고 실행될 수 있다. 그리고 본 발명을 구현하기 위한 기능적인 (functional) 프로그램, 코드 및 코드 세그먼트들은 본 발명이 속하는 기술 분야의 프로그래머들에 의하여 용이하게 추론될 수 있다.

[0124]

이상에서 본 발명에 대하여 그 다양한 실시예들을 중심으로 살펴보았다. 본 발명에 속하는 기술 분야에서 통상의 지식을 가진 자는 본 발명이 본 발명의 본질적인 특성에서 벗어나지 않는 범위에서 변형된 형태로 구현될 수 있음을 이해할 수 있을 것이다. 그러므로 개시된 실시예들은 한정적인 관점이 아니라 설명적인 관점에서 고려되어야 한다. 본 발명의 범위는 전술한 설명이 아니라 특허청구범위에 나타나 있으며, 그와 동등한 범위 내에 있는 모든 차이점은 본 발명에 포함된 것으로 해석되어야 할 것이다.

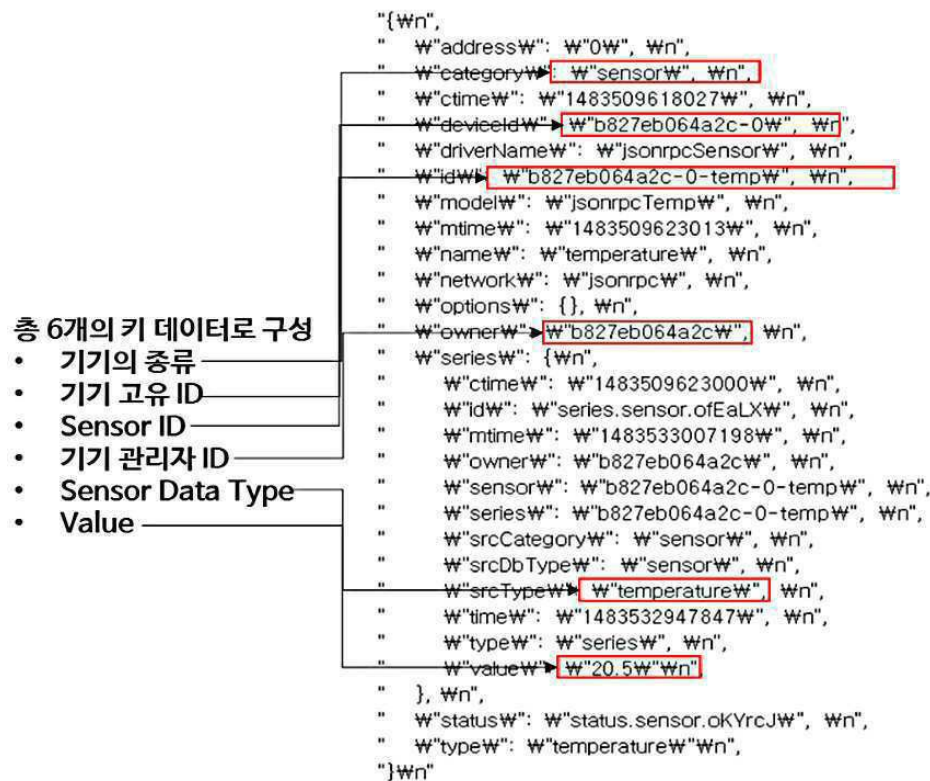
부호의 설명

[0125]

10: 송신자
 20: 수신자/사용자 기기 21: 통신부
 23: 처리부 25: 저장부
 30: 다른 사용자
 40: 네트워크

도면

도면1



도면2

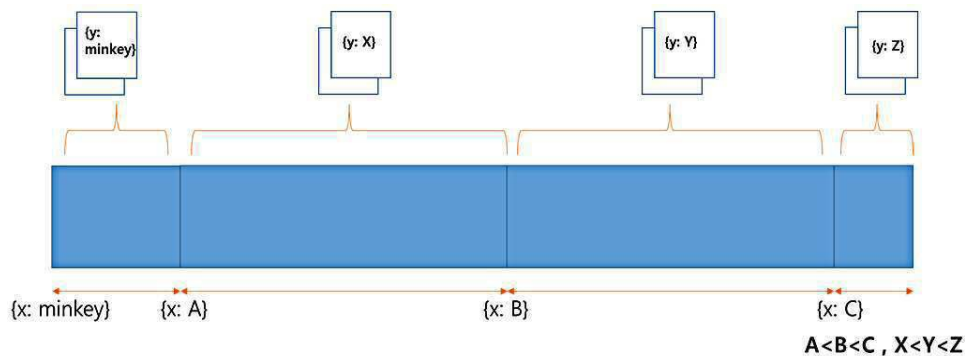
Bitcoin Core API call lists (v0.8.0)

- GetBalance
 - GetBlock
 - GetBlockHash
 - GetTransaction
 - GetReceivedByAddress
 - ListReceivedbyAddress
 - ListSinceBlock
 - ListTransactions
- 단일 조건 검색
- 범위 검색

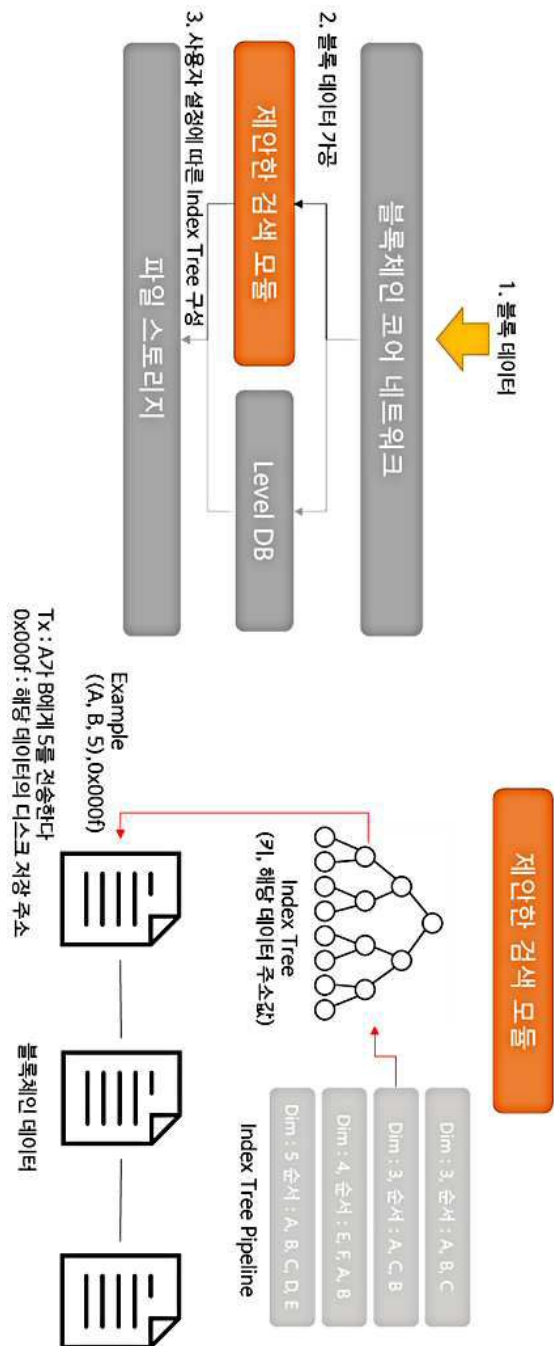
Bitcoin Search Web browser API call lists (v0.8.0)

- /rawblock/\$block_hash
 - /rawtx/\$tx_hash
 - /block-height/\$block_height
 - /rawaddr/\$bitcoin_address
 - /balance?active=\$address
 - /latestblock
- 단일 조건 검색

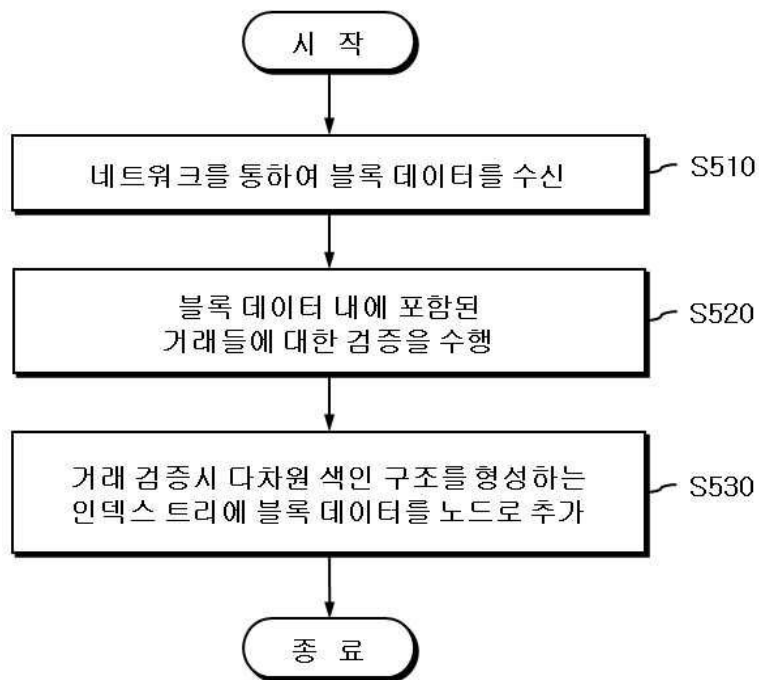
도면3



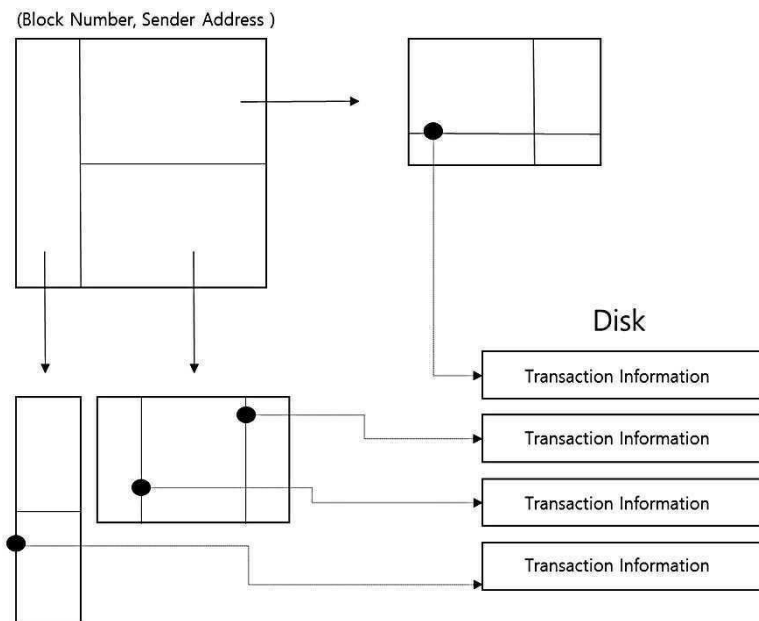
도면4



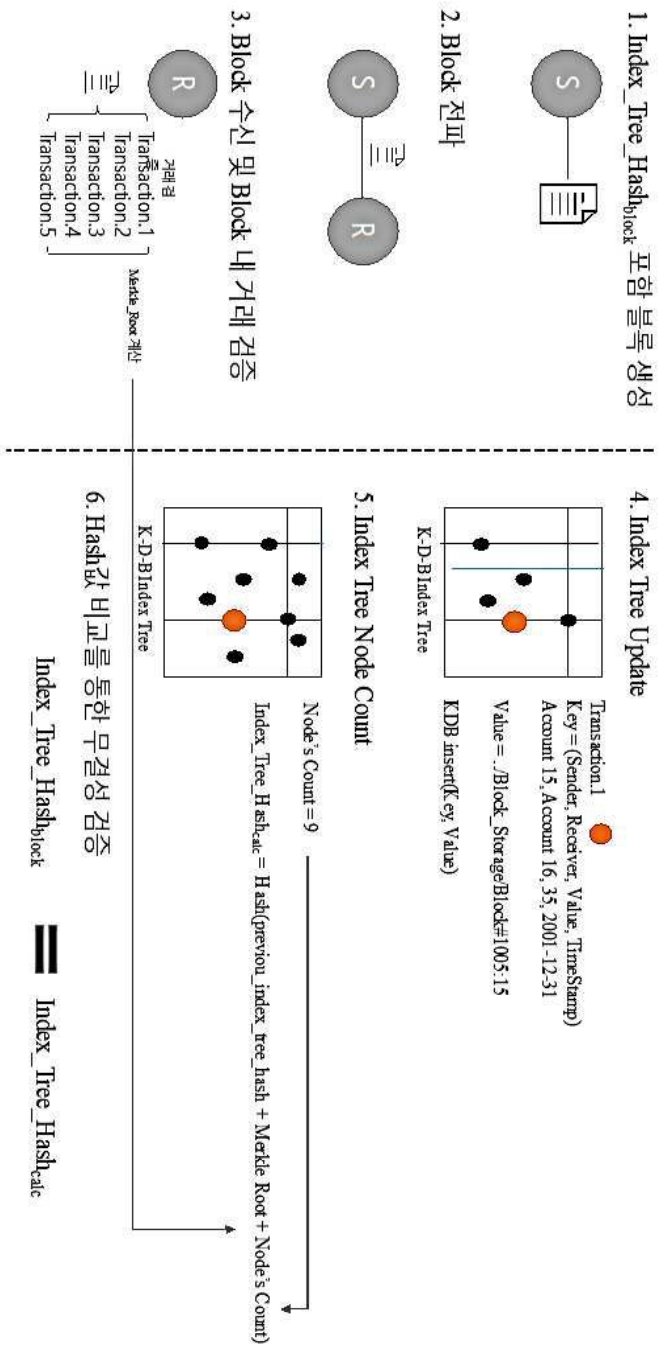
도면5



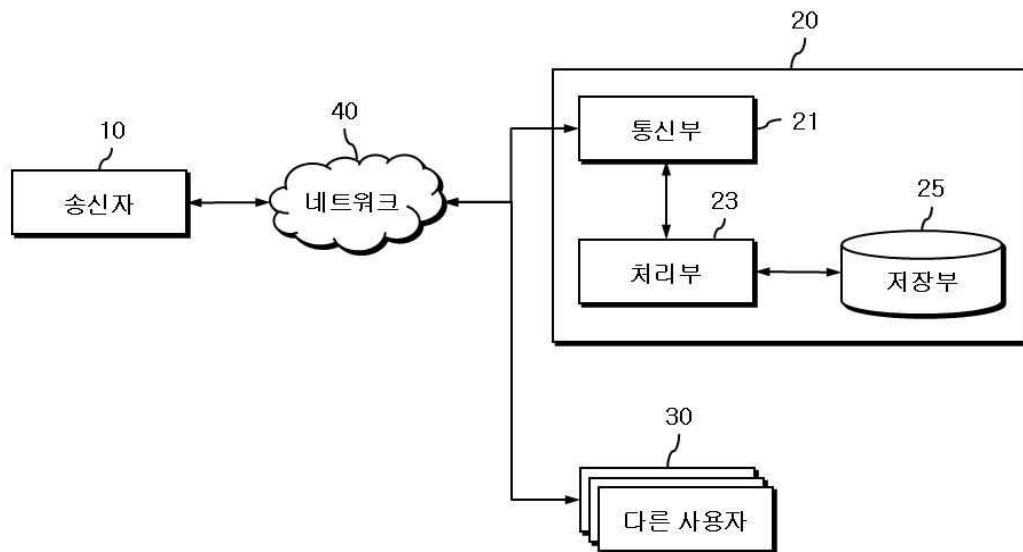
도면6



도면7



도면8



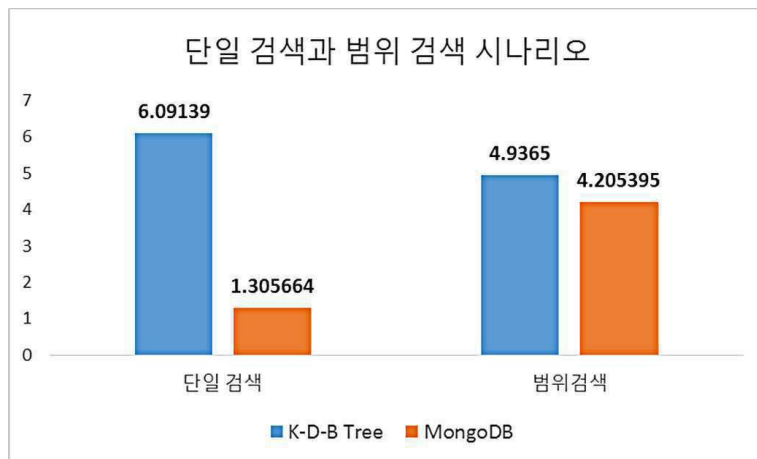
도면9



도면10



도면11



도면12

