(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2020년07월27일
(11) 등록번호 10-2134549
(24) 등록일자 2020년07월09일

(51) 국제특허분류(Int. Cl.)
H04L 29/06 (2006.01) *G06Q 20/02* (2012.01)
H04L 9/06 (2006.01)
(52) CPC특허분류
H04L 63/08 (2013.01)
G06Q 20/02 (2013.01)
(21) 출원번호 10-2019-7023532
(22) 출원일자(국제) 2018년12월13일
심사청구일자 2019년09월26일
(85) 번역문제출일자 2019년08월09일
(65) 공개번호 10-2020-0074912
(43) 공개일자 2020년06월25일
(86) 국제출원번호 PCT/CN2018/120873
(87) 국제공개번호 WO 2019/072296
국제공개일자 2019년04월18일
(56) 선행기술조사문헌
US20100162036 A1
KR1020190077000 A

(73) 특허권자
알리바바 그룹 홀딩 리미티드
케이만군도, 그랜드 케이만, 피오박스 847, 원 캐피탈 플레이스 4층
(72) 발명자
린 펭
중국 저지앙 311121 항저우 유 항 디스트릭트 웨스트 웬 이 로드 넘버 969 빌딩 3 알리바바 그룹 리갈 디파트먼트 5층
(74) 대리인
김태홍, 김진희

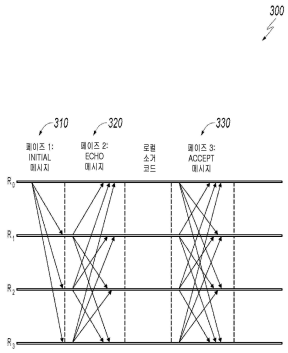
전체 청구항 수 : 총 27 항 심사관 : 문형섭

(54) 발명의 명칭 **분산 시스템에서 프라이머리 노드의 체인지 수행**

(57) 요 약

블록체인 네트워크에서 프라이머리 노드의 체인지를 수행하는 방법은, 블록체인 네트워크 내의 백업 노드가, 에포크 체인지가 수행될 필요가 있다고 결정하는 것, 현재의 에포크에서의 합의 프로세스의 세 개의 페이즈의 각각과 연관되는 백업 노드의 각각의 가중치를 결정하는 것, 각각의 가중치에 기초하여 백업 노드에 대한 가중치 합을 결정하는 것, 다른 네트워크 노드에 EPOCH_CHANGE 메시지를 전송하여 새로운 에포크에서 새로운 프라이머리 노드를 적용하는 것, 다른 네트워크 노드로부터 NEW_EPOCH 메시지를 수신하는 것, 다수의 유효한 NEW_EPOCH 메시지가 미리 결정된 제 2 임계치를 초과하는지의 여부를 결정하는 것, 및 유효한 NEW_EPOCH 메시지의 수가 제 2 미리 결정된 임계치를 초과한다고 결정하는 것에 응답하여 백업 노드를 새로운 에포크에서의 새로운 프라이머리 노드가 되는 것으로 결정하는 것을 포함한다.

대 표 도



(52) CPC특허분류
H04L 9/0643 (2013.01)
H04L 2209/38 (2013.01)

명 세 서

청구범위

청구항 1

복수의 네트워크 노드를 포함하는 블록체인 네트워크에서 프라이머리 노드(primary node)의 체인지를 수행하는 컴퓨터로 구현된(computer-implemented) 방법으로서,

상기 복수의 네트워크 노드는 상기 프라이머리 노드 및 하나 이상의 백업 노드를 포함하고, 상기 방법은:

백업 노드에 의해, 에포크 체인지(epoch change) - 상기 에포크 체인지는, 현재의 프라이머리 노드를 갖는 현재의 에포크로부터 새로운 프라이머리 노드를 갖는 새로운 에포크로의 체인지를 야기하고, 상기 현재의 에포크는 상기 프라이머리 노드를 사용하여 상기 복수의 네트워크 노드 사이에서 동의를 얻기 위한 합의 프로세스를 포함하고, 상기 합의 프로세스는 세 개의 페이즈(phase)를 포함함 - 가 수행될 필요가 있다고 결정하는 단계;

상기 백업 노드에 의해, 상기 현재의 에포크에서의 상기 합의 프로세스의 상기 세 개의 페이즈의 각각과 연관되는 상기 백업 노드의 각각의 가중치 - 상기 가중치는 상기 새로운 프라이머리 노드가 될 상기 백업 노드의 자격부여(qualification)의 메트릭임 - 를 결정하는 단계;

상기 백업 노드에 의해, 상기 현재의 에포크에서의 상기 세 개의 페이즈의 각각과 연관되는 상기 백업 노드의 상기 각각의 가중치에 기초하여 상기 백업 노드에 대한 가중치 합을 결정하는 단계;

상기 가중치 합이 제1 미리 결정된 임계치(threshold)에 도달한다고 결정하는 것에 응답하여, 상기 백업 노드에 의해, EPOCH_CHANGE 메시지 - 상기 EPOCH_CHANGE 메시지는, 상기 현재의 프라이머리 노드를 갖는 상기 현재의 에포크로부터 상기 백업 노드가 상기 새로운 프라이머리 노드인 상기 새로운 에포크로의 체인지에 대한 요청을 나타내고, 상기 EPOCH_CHANGE 메시지는 상기 백업 노드의 상기 가중치 합을 포함함 - 를 상기 네트워크 노드 이외의 상기 복수의 네트워크 노드로 전송하는 단계;

상기 백업 노드에 의해, 상기 백업 노드 이외의 상기 복수의 네트워크 노드 중 적어도 하나로부터 적어도 하나의 NEW_EPOCH 메시지 - 상기 NEW_EPOCH 메시지는 상기 새로운 프라이머리 노드가 될 상기 백업 노드의 확인 응답(acknowledgement)을 나타냄 - 를 수신하는 단계;

상기 백업 노드에 의해, 상기 적어도 하나의 NEW_EPOCH 메시지가 유효한지의 여부를 검증하는 단계;

상기 백업 노드에 의해, 상기 적어도 하나의 NEW_EPOCH 메시지 중의 유효한 NEW_EPOCH 메시지의 수가 제2 미리 결정된 임계치를 초과하는지의 여부를 결정하는 단계; 및

상기 유효한 NEW_EPOCH 메시지의 수가 상기 제2 미리 결정된 임계치를 초과한다고 결정하는 것에 응답하여, 상기 백업 노드에 의해, 상기 백업 노드를 상기 새로운 에포크에서 상기 새로운 프라이머리 노드가 되는 것으로 결정하는 단계를 포함하고,

상기 현재의 에포크에서의 상기 합의 프로세스의 상기 세 개의 페이즈의 각각과 연관되는 상기 백업 노드의 각각의 가중치를 결정하는 단계는:

상기 현재의 에포크에서의 상기 합의 프로세스의 제3 페이즈에서 쿼럼 검증(quorum verification)의 실패를 결정하는 것에 응답하여, 상기 합의 프로세스의 상기 제3 페이즈에 대한 상기 백업 노드의 가중치를 제3 값이 되는 것으로 결정하는 것, 및

상기 현재의 에포크에서의 상기 합의 프로세스의 상기 제3 페이즈에서 쿼럼 검증의 성공을 결정하는 것에 응답하여, 상기 합의 프로세스의 상기 제3 페이즈에 대한 상기 백업 노드의 상기 가중치를 제4 값 - 상기 제3 값은 상기 제4 값보다 더 작음 - 이 되는 것으로 결정하는 것을 포함하는 것인, 복수의 네트워크 노드를 포함하는 블록체인 네트워크에서 프라이머리 노드의 체인지를 수행하는 컴퓨터로 구현된 방법.

청구항 2

제1항에 있어서,

상기 현재의 에포크에서의 상기 합의 프로세스의 상기 세 개의 페이즈의 각각과 연관되는 상기 백업 노드의 각

각의 가중치를 결정하는 단계는, 상기 합의 프로세스의 제1 페이즈에 대한 상기 백업 노드의 가중치를 제1 값이 되는 것으로 결정하는 것을 포함하는 것인, 복수의 네트워크 노드를 포함하는 블록체인 네트워크에서 프라이머리 노드의 체인지를 수행하는 컴퓨터로 구현된 방법.

청구항 3

제1항에 있어서,

상기 현재의 에포크에서의 상기 합의 프로세스의 상기 세 개의 페이즈의 각각과 연관되는 상기 백업 노드의 각각의 가중치를 결정하는 단계는:

상기 현재의 에포크에서의 상기 합의 프로세스의 제2 페이즈에서 쿼럼 검증의 실패를 결정하는 것에 응답하여, 상기 합의 프로세스의 상기 제2 페이즈에 대한 상기 백업 노드의 가중치를 제1 값이 되는 것으로 결정하는 것; 및

상기 현재의 에포크에서의 상기 합의 프로세스의 상기 제2 페이즈에서 쿼럼 검증의 성공을 결정하는 것에 응답하여, 상기 합의 프로세스의 상기 제2 페이즈에 대한 상기 백업 노드의 상기 가중치를 제2 값 - 상기 제1 값은 상기 제2 값보다 더 작음 - 이 되는 것으로 결정하는 것을 포함하는 것인, 복수의 네트워크 노드를 포함하는 블록체인 네트워크에서 프라이머리 노드의 체인지를 수행하는 컴퓨터로 구현된 방법.

청구항 4

제3항에 있어서,

상기 네트워크 노드에 대한 상기 제2 페이즈에서의 상기 쿼럼 검증은, 다른 네트워크 노드로부터 미리 결정된 수의 ECHO 메시지를 수신하는 것을 포함하는 것인, 복수의 네트워크 노드를 포함하는 블록체인 네트워크에서 프라이머리 노드의 체인지를 수행하는 컴퓨터로 구현된 방법.

청구항 5

삭제

청구항 6

제1항에 있어서,

상기 네트워크 노드에 대한 상기 제3 페이즈에서의 상기 쿼럼 검증은, 다른 네트워크 노드로부터 미리 결정된 수의 수락 메시지를 수신하는 것을 포함하고, 다른 네트워크 노드로부터의 상기 수락 메시지의 각각은, 상기 다른 네트워크 노드의 각각이 미리 결정된 수의 ECHO 메시지를 수락하였다는 것을 나타내는 것인, 복수의 네트워크 노드를 포함하는 블록체인 네트워크에서 프라이머리 노드의 체인지를 수행하는 컴퓨터로 구현된 방법.

청구항 7

제1항에 있어서,

상기 EPOCH_CHANGE 메시지는 상기 복수의 네트워크 노드 중의 네트워크 노드의 세트와 연관되는 서명의 세트를 더 포함하고, 상기 NEW_EPOCH 메시지는 상기 EPOCH_CHANGE 메시지의 다이제스트(digest)를 포함하는 것인, 복수의 네트워크 노드를 포함하는 블록체인 네트워크에서 프라이머리 노드의 체인지를 수행하는 컴퓨터로 구현된 방법.

청구항 8

제7항에 있어서,

상기 적어도 하나의 NEW_EPOCH 메시지가 유효한지의 여부를 검증하는 단계는, 상기 적어도 하나의 NEW_EPOCH 메시지 내의 상기 EPOCH_CHANGE 메시지의 상기 다이제스트가 유효한지의 여부를 검증하는 것을 포함하고, 상기 적어도 하나의 NEW_EPOCH 메시지 내의 상기 EPOCH_CHANGE 메시지의 상기 다이제스트가 유효한지의 여부를 검증하는 것은, 상기 EPOCH_CHANGE 메시지 내의 상기 서명의 세트가 유효한지의 여부를 검증하는 것을 포함하는 것인, 복수의 네트워크 노드를 포함하는 블록체인 네트워크에서 프라이머리 노드의 체인지를 수행하는 컴퓨터로 구현된 방법.

청구항 9

제1항에 있어서,

에포크 체인지가 수행될 필요가 있다고 결정하는 단계는, 미리 결정된 시간 기간 내에 이전의 에포크(old epoch)에서 상기 합의가 달성되지 않았다고 결정하는 것에 응답하여 에포크 체인지가 수행될 필요가 있다고 결정하는 것을 포함하는 것인, 복수의 네트워크 노드를 포함하는 블록체인 네트워크에서 프라이머리 노드의 체인지를 수행하는 컴퓨터로 구현된 방법.

청구항 10

제1항에 있어서,

상기 새로운 프라이머리 노드를 갖는 상기 새로운 에포크에서 동작하는 단계를 더 포함하고, 상기 새로운 에포크는 상기 새로운 프라이머리 노드를 사용하여 상기 복수의 네트워크 노드 사이에서 합의를 달성하기 위한 합의 프로세스를 포함하는 것인, 복수의 네트워크 노드를 포함하는 블록체인 네트워크에서 프라이머리 노드의 체인지를 수행하는 컴퓨터로 구현된 방법.

청구항 11

동작들을 수행하도록 컴퓨터 시스템에 의해 실행가능한 하나 이상의 명령어가 저장된 비일시적 컴퓨터 판독 가능 매체로서,

상기 동작들은,

복수의 네트워크 노드를 포함하는 블록체인 네트워크의 백업 노드에 의해, 에포크 체인지가 수행될 필요가 있다고 결정하는 동작 - 상기 복수의 네트워크 노드는 프라이머리 노드 및 상기 백업 노드를 포함하는 하나 이상의 백업 노드를 포함하고, 상기 에포크 체인지는 현재의 프라이머리 노드를 갖는 현재의 에포크로부터 새로운 프라이머리 노드를 갖는 새로운 에포크로의 체인지를 야기하고, 상기 현재의 에포크는 상기 프라이머리 노드를 사용하여 상기 복수의 네트워크 노드 사이에서 동의를 얻기 위한 합의 프로세스를 포함하고, 상기 합의 프로세스는 세 개의 페이즈를 포함함 - ;

상기 백업 노드에 의해, 상기 현재의 에포크에서의 상기 합의 프로세스의 상기 세 개의 페이즈의 각각과 연관되는 상기 백업 노드의 각각의 가중치 - 상기 가중치는 상기 새로운 프라이머리 노드가 될 상기 백업 노드의 자격 부여의 메트릭임 - 를 결정하는 동작;

상기 백업 노드에 의해, 상기 현재의 에포크에서의 상기 세 개의 페이즈의 각각과 연관되는 상기 백업 노드의 상기 각각의 가중치에 기초하여 상기 백업 노드에 대한 가중치 합을 결정하는 동작;

상기 가중치 합이 제1 미리 결정된 임계치에 도달한다고 결정하는 것에 응답하여, 상기 백업 노드에 의해, EPOCH_CHANGE 메시지 - 상기 EPOCH_CHANGE 메시지는, 상기 현재의 프라이머리 노드를 갖는 상기 현재의 에포크로부터 상기 백업 노드가 상기 새로운 프라이머리 노드인 상기 새로운 에포크로의 체인지에 대한 요청을 나타내고, 상기 EPOCH_CHANGE 메시지는 상기 백업 노드의 상기 가중치 합을 포함함 - 를 상기 네트워크 노드 이외의 상기 복수의 네트워크 노드로 전송하는 동작;

상기 백업 노드에 의해, 상기 백업 노드 이외의 상기 복수의 네트워크 노드 중 적어도 하나로부터 적어도 하나의 NEW_EPOCH 메시지 - 상기 NEW_EPOCH 메시지는 상기 새로운 프라이머리 노드가 될 상기 백업 노드의 확인 응답을 나타냄 - 를 수신하는 동작;

상기 백업 노드에 의해, 상기 적어도 하나의 NEW_EPOCH 메시지가 유효한지의 여부를 검증하는 동작;

상기 백업 노드에 의해, 상기 적어도 하나의 NEW_EPOCH 메시지 중의 유효한 NEW_EPOCH 메시지의 수가 제2 미리 결정된 임계치를 초과하는지의 여부를 결정하는 동작; 및

상기 유효한 NEW_EPOCH 메시지의 수가 상기 제2 미리 결정된 임계치를 초과한다고 결정하는 것에 응답하여, 상기 백업 노드에 의해, 상기 백업 노드를 상기 새로운 에포크에서 상기 새로운 프라이머리 노드가 되는 것으로 결정하는 동작을 포함하고,

상기 현재의 에포크에서의 상기 합의 프로세스의 상기 세 개의 페이즈의 각각과 연관되는 상기 백업 노드의 각

각의 가중치를 결정하는 동작은:

상기 현재의 에포크에서의 상기 합의 프로세스의 제3 페이즈에서 쿼럼 검증의 실패를 결정하는 것에 응답하여, 상기 합의 프로세스의 상기 제3 페이즈에 대한 상기 백업 노드의 가중치를 제3 값이 되는 것으로 결정하는 것, 및

상기 현재의 에포크에서의 상기 합의 프로세스의 상기 제3 페이즈에서 쿼럼 검증의 성공을 결정하는 것에 응답하여, 상기 합의 프로세스의 상기 제3 페이즈에 대한 상기 백업 노드의 상기 가중치를 제4 값 - 상기 제3 값은 상기 제4 값보다 더 작음 - 이 되는 것으로 결정하는 것을 포함하는 것인, 비일시적 컴퓨터 판독 가능 매체.

청구항 12

제11항에 있어서,

상기 현재의 에포크에서의 상기 합의 프로세스의 상기 세 개의 페이즈의 각각과 연관되는 상기 백업 노드의 각각의 가중치를 결정하는 동작은, 상기 합의 프로세스의 제1 페이즈에 대한 상기 백업 노드의 가중치를 제1 값이 되는 것으로 결정하는 것을 포함하는 것인, 비일시적 컴퓨터 판독 가능 매체.

청구항 13

제11항에 있어서,

상기 현재의 에포크에서의 상기 합의 프로세스의 상기 세 개의 페이즈의 각각과 연관되는 상기 백업 노드의 각각의 가중치를 결정하는 동작은:

상기 현재의 에포크에서의 상기 합의 프로세스의 제2 페이즈에서 쿼럼 검증의 실패를 결정하는 것에 응답하여, 상기 합의 프로세스의 상기 제2 페이즈에 대한 상기 백업 노드의 가중치를 제1 값이 되는 것으로 결정하는 것; 및

상기 현재의 에포크에서의 상기 합의 프로세스의 상기 제2 페이즈에서 쿼럼 검증의 성공을 결정하는 것에 응답하여, 상기 합의 프로세스의 상기 제2 페이즈에 대한 상기 백업 노드의 상기 가중치를 제2 값 - 상기 제1 값은 상기 제2 값보다 더 작음 - 이 되는 것으로 결정하는 것을 포함하는 것인, 비일시적 컴퓨터 판독 가능 매체.

청구항 14

제13항에 있어서,

상기 네트워크 노드에 대한 상기 제2 페이즈에서의 상기 쿼럼 검증은, 다른 네트워크 노드로부터 미리 결정된 수의 ECHO 메시지를 수신하는 것을 포함하는 것인, 비일시적 컴퓨터 판독 가능 매체.

청구항 15

삭제

청구항 16

제11항에 있어서,

상기 네트워크 노드에 대한 상기 제3 페이즈에서의 상기 쿼럼 검증은, 다른 네트워크 노드로부터 미리 결정된 수의 수락 메시지를 수신하는 것을 포함하고, 다른 네트워크 노드로부터의 상기 수락 메시지의 각각은, 상기 다른 네트워크 노드의 각각이 미리 결정된 수의 ECHO 메시지를 수락하였다는 것을 나타내는 것인, 비일시적 컴퓨터 판독 가능 매체.

청구항 17

제11항에 있어서,

상기 EPOCH_CHANGE 메시지는 상기 복수의 네트워크 노드 중의 네트워크 노드의 세트와 연관되는 서명의 세트를 더 포함하고, 상기 NEW_EPOCH 메시지는 상기 EPOCH_CHANGE 메시지의 다이제스트를 포함하는, 비일시적 컴퓨터

판독 가능 매체.

청구항 18

제17항에 있어서,

상기 적어도 하나의 NEW_EPOCH 메시지가 유효한지의 여부를 검증하는 것은, 상기 적어도 하나의 NEW_EPOCH 메시지 내의 상기 EPOCH_CHANGE 메시지의 상기 다이제스트가 유효한지의 여부를 검증하는 것을 포함하고, 상기 적어도 하나의 NEW_EPOCH 메시지 내의 상기 EPOCH_CHANGE 메시지의 상기 다이제스트가 유효한지의 여부를 검증하는 것은, 상기 EPOCH_CHANGE 메시지 내의 상기 서명의 세트가 유효한지의 여부를 검증하는 것을 포함하는, 비일시적 컴퓨터 판독 가능 매체.

청구항 19

제11항에 있어서,

에포크 체인지가 수행될 필요가 있다고 결정하는 것은, 미리 결정된 시간 기간 내에 이전의 에포크에서 상기 합의가 달성되지 않았다고 결정하는 것에 응답하여 에포크 체인지가 수행될 필요가 있다고 결정하는 것을 포함하는, 비일시적 컴퓨터 판독 가능 매체.

청구항 20

제11항에 있어서,

상기 동작들은, 상기 새로운 프라이머리 노드를 갖는 상기 새로운 에포크에서 동작하는 동작을 더 포함하고, 상기 새로운 에포크는 상기 새로운 프라이머리 노드를 사용하여 상기 복수의 네트워크 노드 사이에서 합의를 달성하기 위한 합의 프로세스를 포함하는 것인, 비일시적 컴퓨터 판독 가능 매체.

청구항 21

시스템으로서,

하나 이상의 컴퓨터; 및

상기 하나 이상의 컴퓨터와 상호동작 가능하게 커플링되고, 상기 하나 이상의 컴퓨터에 의해 실행될 때 하나 이상의 동작을 수행하는 하나 이상의 명령어가 저장된, 유형의(tangible) 비일시적 머신 판독 가능 매체를 갖는 하나 이상의 컴퓨터 메모리 디바이스를 포함하고,

상기 하나 이상의 동작은,

복수의 네트워크 노드를 포함하는 블록체인 네트워크의 백업 노드에 의해, 에포크 체인지가 수행될 필요가 있다고 결정하는 동작 - 상기 복수의 네트워크 노드는 프라이머리 노드 및 상기 백업 노드를 포함하는 하나 이상의 백업 노드를 포함하고, 상기 에포크 체인지는 현재의 프라이머리 노드를 갖는 현재의 에포크로부터 새로운 프라이머리 노드를 갖는 새로운 에포크로의 체인지를 야기하고, 상기 현재의 에포크는 상기 프라이머리 노드를 사용하여 상기 복수의 네트워크 노드 사이에서 동의를 얻기 위한 합의 프로세스를 포함하고, 상기 합의 프로세스는 세 개의 페이즈를 포함함 - ;

상기 백업 노드에 의해, 상기 현재의 에포크에서의 상기 합의 프로세스의 상기 세 개의 페이즈의 각각과 연관되는 상기 백업 노드의 각각의 가중치 - 상기 가중치는 상기 새로운 프라이머리 노드가 될 상기 백업 노드의 자격 부여의 메트릭임 - 를 결정하는 동작;

상기 백업 노드에 의해, 상기 현재의 에포크에서의 상기 세 개의 페이즈의 각각과 연관되는 상기 백업 노드의 상기 각각의 가중치에 기초하여 상기 백업 노드에 대한 가중치 합을 결정하는 동작;

상기 가중치 합이 제1 미리 결정된 임계치에 도달한다고 결정하는 것에 응답하여, 상기 백업 노드에 의해, EPOCH_CHANGE 메시지 - 상기 EPOCH_CHANGE 메시지는, 상기 현재의 프라이머리 노드를 갖는 상기 현재의 에포크로부터 상기 백업 노드가 상기 새로운 프라이머리 노드인 상기 새로운 에포크로의 체인지에 대한 요청을 나타내고, 상기 EPOCH_CHANGE 메시지는 상기 백업 노드의 상기 가중치 합을 포함함 - 를 상기 네트워크 노드 이외의 상기 복수의 네트워크 노드로 전송하는 동작;

상기 백업 노드에 의해, 상기 백업 노드 이외의 상기 복수의 네트워크 노드 중 적어도 하나로부터 적어

도 하나의 NEW_EPOCH 메시지 - 상기 NEW_EPOCH 메시지는 상기 새로운 프라이머리 노드가 될 상기 백업 노드의 확인 응답을 나타냄 - 를 수신하는 동작;

상기 백업 노드에 의해, 상기 적어도 하나의 NEW_EPOCH 메시지가 유효한지의 여부를 검증하는 동작;

상기 백업 노드에 의해, 상기 적어도 하나의 NEW_EPOCH 메시지 중의 유효한 NEW_EPOCH 메시지의 수가 제2 미리 결정된 임계치를 초과하는지의 여부를 결정하는 동작; 및

상기 유효한 NEW_EPOCH 메시지의 수가 상기 제2 미리 결정된 임계치를 초과한다고 결정하는 것에 응답하여, 상기 백업 노드에 의해, 상기 백업 노드를 상기 새로운 에포크에서 상기 새로운 프라이머리 노드가 되는 것으로 결정하는 동작을 포함하고,

상기 현재의 에포크에서의 상기 합의 프로세스의 상기 세 개의 페이즈의 각각과 연관되는 상기 백업 노드의 각각의 가중치를 결정하는 동작은:

상기 현재의 에포크에서의 상기 합의 프로세스의 제3 페이즈에서 쿼럼 검증의 실패를 결정하는 것에 응답하여, 상기 합의 프로세스의 상기 제3 페이즈에 대한 상기 백업 노드의 가중치를 제3 값이 되는 것으로 결정하는 것, 및

상기 현재의 에포크에서의 상기 합의 프로세스의 상기 제3 페이즈에서 쿼럼 검증의 성공을 결정하는 것에 응답하여, 상기 합의 프로세스의 상기 제3 페이즈에 대한 상기 백업 노드의 상기 가중치를 제4 값 - 상기 제3 값은 상기 제4 값보다 더 작음 - 이 되는 것으로 결정하는 것을 포함하는 것인, 시스템.

청구항 22

제21항에 있어서,

상기 현재의 에포크에서의 상기 합의 프로세스의 상기 세 개의 페이즈의 각각과 연관되는 상기 백업 노드의 각각의 가중치를 결정하는 동작은, 상기 합의 프로세스의 제1 페이즈에 대한 상기 백업 노드의 가중치를 제1 값이 되는 것으로 결정하는 것을 포함하는 것인, 시스템.

청구항 23

제21항에 있어서,

상기 현재의 에포크에서의 상기 합의 프로세스의 상기 세 개의 페이즈의 각각과 연관되는 상기 백업 노드의 각각의 가중치를 결정하는 동작은:

상기 현재의 에포크에서의 상기 합의 프로세스의 제2 페이즈에서 쿼럼 검증의 실패를 결정하는 것에 응답하여, 상기 합의 프로세스의 상기 제2 페이즈에 대한 상기 백업 노드의 가중치를 제1 값이 되는 것으로 결정하는 것, 및

상기 현재의 에포크에서의 상기 합의 프로세스의 상기 제2 페이즈에서 쿼럼 검증의 성공을 결정하는 것에 응답하여, 상기 합의 프로세스의 상기 제2 페이즈에 대한 상기 백업 노드의 상기 가중치를 제2 값 - 상기 제1 값은 상기 제2 값보다 더 작음 - 이 되는 것으로 결정하는 것을 포함하는 것인, 시스템.

청구항 24

제23항에 있어서,

상기 네트워크 노드에 대한 상기 제2 페이즈에서의 상기 쿼럼 검증은, 다른 네트워크 노드로부터 미리 결정된 수의 ECHO 메시지를 수신하는 것을 포함하는 것인, 시스템.

청구항 25

삭제

청구항 26

제21항에 있어서,

상기 네트워크 노드에 대한 상기 제3 페이즈에서의 상기 쿼럼 검증은, 다른 네트워크 노드로부터 미리 결정된

수의 수락 메시지를 수신하는 것을 포함하고, 다른 네트워크 노드로부터의 상기 수락 메시지의 각각은, 상기 다른 네트워크 노드의 각각이 미리 결정된 수의 ECHO 메시지를 수락하였다는 것을 나타내는 것인, 시스템.

청구항 27

제21항에 있어서,

상기 EPOCH_CHANGE 메시지는 상기 복수의 네트워크 노드 중의 네트워크 노드의 세트와 연관되는 서명의 세트를 더 포함하고, 상기 NEW_EPOCH 메시지는 상기 EPOCH_CHANGE 메시지의 다이제스트를 포함하는 것인, 시스템.

청구항 28

제27항에 있어서,

상기 적어도 하나의 NEW_EPOCH 메시지가 유효한지의 여부를 검증하는 것은, 상기 적어도 하나의 NEW_EPOCH 메시지 내의 상기 EPOCH_CHANGE 메시지의 상기 다이제스트가 유효한지의 여부를 검증하는 것을 포함하고, 상기 적어도 하나의 NEW_EPOCH 메시지 내의 상기 EPOCH_CHANGE 메시지의 상기 다이제스트가 유효한지의 여부를 검증하는 것은, 상기 EPOCH_CHANGE 메시지 내의 상기 서명의 세트가 유효한지의 여부를 검증하는 것을 포함하는, 시스템.

청구항 29

제21항에 있어서,

에포크 체인지가 수행될 필요가 있다고 결정하는 것은, 미리 결정된 시간 기간 내에 이전의 에포크에서 상기 합의가 달성되지 않았다고 결정하는 것에 응답하여 에포크 체인지가 수행될 필요가 있다고 결정하는 것을 포함하는, 시스템.

청구항 30

제21항에 있어서,

상기 하나 이상의 동작은, 상기 새로운 프라이머리 노드를 갖는 상기 새로운 에포크에서 동작하는 동작을 더 포함하고, 상기 새로운 에포크는 상기 새로운 프라이머리 노드를 사용하여 상기 복수의 네트워크 노드 사이에서 합의를 달성하기 위한 합의 프로세스를 포함하는 것인, 시스템.

청구항 31

삭제

청구항 32

삭제

청구항 33

삭제

청구항 34

삭제

청구항 35

삭제

청구항 36

삭제

청구항 37

삭제

청구항 38

삭제

청구항 39

삭제

청구항 40

삭제

청구항 41

삭제

청구항 42

삭제

발명의 설명

기 술 분 야

배 경 기 술

[0001] 합의(consensus) 네트워크, 및/또는 블록체인 네트워크로도 또한 칭해질 수 있는 분산 원장 시스템(distributed ledger system; DLS)은 참가하는 엔티티가 데이터를 안전하고, 불변하게 저장하는 것을 가능하게 한다. DLS는 일반적으로 어떠한 특정한 유저 사례도 참조하지 않는 블록체인 네트워크로 일반적으로 칭해진다. 블록체인 네트워크의 예는 다음의 것을 포함할 수 있다: 퍼블릭 블록체인 네트워크(public blockchain network), 프라이빗 블록체인 네트워크(private blockchain network), 및 컨소시엄 블록체인 네트워크(consortium blockchain network). 퍼블릭 블록체인 네트워크는 모든 엔티티가 DLS를 사용하고, 합의 프로세스에 참가하도록 개방되어 있다. 프라이빗 블록체인 네트워크는, 판독 및 기록 퍼미션(permission)을 중앙 집중식으로 제어하는 특정한 엔티티에 대해 제공된다. 컨소시엄 블록체인 네트워크는, 합의 프로세스를 제어하는 엔티티의 선택된 그룹에 제공되며, 액세스 제어 레이어를 포함한다.

[0002] 합의 메커니즘은 분산 블록체인 시스템의 프라이머리 컴포넌트(primary component)이다. 합의 메커니즘은, 분산 프로세스 또는 시스템 사이에서 단일의 데이터 값에 대한 합의를 달성하기 위해 사용되는 컴퓨터 사이언스의 프로세스이다. 합의 메커니즘은 다수의 신뢰 불가능한 노드를 수반하는 네트워크에서 신뢰성을 달성하도록 설계된다. 그 이슈 - 합의 문제로 알려짐 - 를 해결하는 것은, 분산 컴퓨팅 및 다중 에이전트 시스템에서 중요하다.

[0003] 블록체인은 노드 간에 합의에 도달하기 위해 합의 메커니즘에 의존한다. 블록체인은 피어 투 피어(peer-to-peer; P2P) 네트워크 상의 분산 컴퓨터에 의해 관리되는 탈집중화된(decentralized) 데이터베이스이다. 각각의 피어는 단일 고장 지점(Single Point of Failure; SPOF)을 방지하기 위해 원장(ledger)의 사본을 유지한다. 업데이트 및 유효성 확인(validation)은 모든 사본에서 동시에 반영된다.

[0004] 비록 블록체인 시스템의 네트워크 노드 사이에서 합의를 수행하기 위해 다수의 현존하는 기술이 사용될 수 있지만, 합의를 수행하기 위한 더욱 효율적인 솔루션이 유리할 것이다.

발명의 내용

[0005] 본 명세서의 구현예는 분산 시스템(예를 들면, 블록체인 네트워크)에서 합의 문제를 해결하기 위한 컴퓨터 구현 방법을 포함한다. 특히, 본 명세서의 구현예는, 분산 시스템에서의 프라이머리 노드(primary node)의 체인지를 수행하는 것에 관한 것이다.

[0006] 몇몇 구현예에서, 동작은 다음의 것을 포함한다: 블록체인 네트워크의 백업 노드에 의해 에포크 체인지(epoch change) - 에포크 체인지는 현재의 프라이머리 노드를 갖는 현재의 에포크로부터 새로운 프라이머리 노드를 갖

는 새로운 에포크로의 체인지를 야기하고, 현재의 에포크는 프라이머리 노드를 사용하여 다수의 네트워크 노드 사이에서 합의를 달성하기 위한 합의 프로세스를 포함하고, 합의 프로세스는 세 개의 페이즈(phase)를 포함함 - 가 수행될 필요가 있다고 결정하는 것; 상기 백업 노드에 의해, 현재의 에포크에서의 합의 프로세스의 세 개의 페이즈의 각각과 연관되는 백업 노드의 각각의 가중치를 결정하는 것; 현재의 에포크에서의 세 개의 페이즈의 각각과 연관되는 백업 노드의 각각의 가중치에 기초하여 백업 노드에 의해 백업 노드에 대한 가중치 합을 결정하는 것; 가중치 합이 제1 미리 결정된 임계치에 도달한다고 결정하는 것에 응답하여 EPOCH_CHANGE 메시지 - EPOCH_CHANGE 메시지는, 현재의 프라이머리 노드를 갖는 현재의 에포크로부터, 백업 노드가 새로운 프라이머리 노드인 새로운 에포크로의 체인지에 대한 요청을 나타내고, EPOCH_CHANGE는 백업 노드의 가중치 합을 포함함 - 를 백업 노드에 의해 네트워크 노드 이외의 다수의 네트워크 노드로 전송하는 것; 백업 노드 이외의 다수의 네트워크 노드 중 적어도 하나로부터 백업 노드에 의해 적어도 하나의 NEW_EPOCH 메시지 - NEW_EPOCH 메시지는 새로운 프라이머리 노드가 될 백업 노드의 확인 응답을 나타냄 - 를 수신하는 것; 적어도 하나의 NEW_EPOCH 메시지가 유효한지의 여부를 백업 노드에 의해 검증하는 것; 적어도 하나의 NEW_EPOCH 메시지 중의 유효한 NEW_EPOCH 메시지의 수가 제2 미리 결정된 임계치를 초과하는지의 여부를 백업 노드에 의해 결정하는 것; 및 유효한 NEW_EPOCH 메시지의 수가 제2 미리 결정된 임계치를 초과한다고 결정하는 것에 응답하여, 백업 노드를 새로운 에포크에서 새로운 프라이머리 노드가 되는 것으로 백업 노드에 의해 결정하는 것.

[0007] 다른 구현예는, 컴퓨터 스토리지 디바이스 상에 인코딩되는, 방법의 동작(action)을 수행하도록 구성되는, 대응하는 시스템, 장치, 및 컴퓨터 프로그램을 포함한다.

[0008] 이들 및 다른 구현예 각각은, 옵션 사항으로(optionally) 다음의 피쳐 중 하나 이상을 포함할 수도 있다:

[0009] 백업 노드는 합의 프로세스의 제1 페이즈에 대한 백업 노드의 가중치를 제1 값이 되는 것으로 결정하는, 다음의 피쳐 중 임의의 것과 결합 가능한 제1 피쳐.

[0010] 백업 노드는, 현재의 에포크에서의 합의 프로세스의 제2 페이즈에서 쿼럼(quorum) 검증의 실패를 결정하는 것에 응답하여, 합의 프로세스의 제2 페이즈에 대한 백업 노드의 가중치를 제1 값이 되는 것으로 결정하고, 백업 노드는, 현재의 에포크에서의 합의 프로세스의 제2 페이즈에서 쿼럼 검증의 성공을 결정하는 것에 응답하여, 합의 프로세스의 제2 페이즈에 대한 백업 노드의 가중치를 제1 값보다 더 큰 제2 값이 되는 것으로 결정하는, 다음의 피쳐 중 임의의 것과 결합 가능한 제2 피쳐.

[0011] 네트워크 노드에 대한 제2 페이즈에서의 쿼럼 검증은, 다른 네트워크 노드로부터 미리 결정된 수의 ECHO 메시지를 수신하는 것을 포함하는, 다음의 피쳐 중 임의의 것과 결합 가능한 제3 피쳐.

[0012] 백업 노드는, 현재의 에포크에서의 합의 프로세스의 제3 페이즈에서 쿼럼 검증의 실패를 결정하는 것에 응답하여, 합의 프로세스의 제3 페이즈에 대한 백업 노드의 가중치를 제3 값이 되는 것으로 결정하고, 백업 노드는, 현재의 에포크에서의 합의 프로세스의 제3 페이즈에서의 쿼럼 검증의 성공을 결정하는 것에 응답하여, 합의 프로세스의 제3 페이즈에 대한 백업 노드의 가중치를 제3 값보다 더 큰 제4 값이 되는 것으로 결정하는, 다음의 피쳐 중 임의의 것과 결합 가능한 제4 피쳐.

[0013] 네트워크 노드에 대한 제3 페이즈에서의 쿼럼 검증은, 다른 네트워크 노드로부터 미리 결정된 수의 수락 메시지를 수신하는 것을 포함하는데, 여기서 다른 네트워크 노드로부터의 수락 메시지의 각각은, 다른 네트워크 노드의 각각이 미리 결정된 수의 ECHO 메시지를 수락하였다는 것을 나타내는, 다음의 피쳐 중 임의의 것과 결합 가능한 제5 피쳐.

[0014] EPOCH_CHANGE 메시지는 다수의 네트워크 노드 중의 네트워크 노드의 세트와 연관되는 서명의 세트를 더 포함하고, NEW_EPOCH 메시지는 EPOCH_CHANGE 메시지의 다이제스트(digest)를 포함하는, 다음의 피쳐 중 임의의 것과 결합 가능한 제6 피쳐.

[0015] 적어도 하나의 유효한 NEW_EPOCH 메시지가 유효한지의 여부를 검증하는 것은, 적어도 하나의 NEW_EPOCH 메시지 내의 EPOCH_CHANGE 메시지의 다이제스트가 유효한지의 여부를 검증하는 것을 포함하는, 다음의 피쳐 중 임의의 것과 결합 가능한 제7 피쳐.

[0016] 적어도 하나의 NEW_EPOCH 메시지 내의 EPOCH_CHANGE 메시지의 다이제스트가 유효한지 여부를 검증하는 것은, EPOCH_CHANGE 메시지에서의 서명의 세트가 유효한지 여부를 검증하는 것을 포함하는, 다음의 피쳐 중 임의의 것과 결합 가능한 제8 피쳐.

[0017] 백업 노드는, 미리 결정된 시간 기간 내에 이전의 에포크(old epoch)에서 합의가 달성되지 않았다고 결정하는

것에 응답하여 에포크 체인지가 수행될 필요가 있다고 결정하는, 다음의 피쳐 중 임의의 것과 결합 가능한 제9 피쳐.

[0018] 새로운 에포크는 새로운 프라이머리 노드를 사용하여 다수의 네트워크 노드 사이에서 합의를 달성하기위한 합의 프로세스를 포함하는, 다음의 피쳐 중 임의의 것과 결합 가능한 제10 피쳐.

[0019] 몇몇 구현예에서, 동작은 다음의 것을 포함한다: 다수의 네트워크 노드 중의 네트워크 노드에 의해, 네트워크 노드 이외의 백업 노드로부터 EPOCH_CHANGE 메시지 - EPOCH_CHANGE 메시지는 에포크 체인지가 수행될 필요가 있다는 표시(indication)를 포함하고, 에포크 체인지는 현재의 프라이머리 노드를 갖는 현재의 에포크로부터 새로운 프라이머리 노드를 갖는 새로운 에포크로의 체인지를 야기함 - 를 수신하는 것; 네트워크 노드에 의해, EPOCH_CHANGE 메시지 메시지가 유효한지의 여부를 검증하는 것; EPOCH_CHANGE 메시지가 유효하다고 검증하는 것에 응답하여, 네트워크 노드에 의해, NEW_EPOCH 메시지 - NEW_EPOCH 메시지는 EPOCH_CHANGE 메시지의 다이제스트를 포함함 - 를 다른 네트워크 노드로 전송하는 것; 네트워크 노드에 의해, 네트워크 노드 이외의 다수의 네트워크 노드 중 적어도 하나로부터 적어도 하나의 NEW_EPOCH 메시지를 수신하는 것; 네트워크 노드에 의해, 적어도 하나의 NEW_EPOCH 메시지가 유효한지의 여부를 검증하는 것; 백업 노드에 의해, 적어도 하나의 NEW_EPOCH 메시지 중의 유효한 NEW_EPOCH 메시지의 수가 미리 결정된 임계치를 초과하는지의 여부를 결정하는 것; 및 유효한 NEW_EPOCH 메시지의 수가 미리 결정된 임계치를 초과한다고 결정하는 것에 응답하여, 네트워크 노드에 의해, 백업 노드를 새로운 에포크에서 새로운 프라이머리 노드가 되는 것으로 결정하는 것.

[0020] 다른 구현예는, 컴퓨터 스토리지 디바이스 상에 인코딩되는, 방법의 동작을 수행하도록 구성되는, 대응하는 시스템, 장치, 및 컴퓨터 프로그램을 포함한다.

[0021] 이들 및 다른 구현예 각각은, 옵션 사항으로 다음의 피쳐 중 하나 이상을 포함할 수도 있다:

[0022] EPOCH_CHANGE 메시지는 백업 노드와 연관되는 가중치 합 및 다수의 네트워크 노드 중의 네트워크 노드의 세트와 연관되는 서명의 세트를 포함하는, 다음의 피쳐 중 임의의 것과 결합 가능한 제1 피쳐.

[0023] EPOCH_CHANGE 메시지가 유효한지의 여부를 검증하는 것은, EPOCH_CHANGE 메시지 내의 가중치 합이 유효한지의 여부를 검증하는 것을 포함하고, EPOCH_CHANGE 메시지 내의 가중치 합이 유효한지의 여부를 검증하는 것은 서명의 세트가 유효한지의 여부를 검증하는 것을 포함하는, 다음의 피쳐 중 임의의 것과 결합 가능한 제2 피쳐.

[0024] 적어도 하나의 NEW_EPOCH 메시지가 유효한지의 여부를 검증하는 것은, 적어도 하나의 NEW_EPOCH 메시지 내의 EPOCH_CHANGE 메시지의 다이제스트가 유효한지의 여부를 검증하는 것을 포함하고, 적어도 하나의 NEW_EPOCH 메시지 내의 EPOCH_CHANGE 메시지의 다이제스트가 유효한지의 여부를 검증하는 것은, EPOCH_CHANGE 메시지 내의 서명의 세트가 유효한지의 여부를 검증하는 것을 포함하는, 다음의 피쳐 중 임의의 것과 결합 가능한 제3 피쳐.

[0025] 본 명세서는 또한, 하나 이상의 프로세서에 커플링되며, 하나 이상의 프로세서에 의해 실행될 때, 하나 이상의 프로세서로 하여금, 본원에서 제공되는 방법의 구현예에 따른 동작을 수행하게 하는 명령어들이 저장된 하나 이상의 비일시적 컴퓨터 판독 가능 저장 매체를 제공한다.

[0026] 본 명세서는 또한, 본원에서 제공되는 방법을 구현하기 위한 시스템을 제공한다. 시스템은 하나 이상의 프로세서, 및 하나 이상의 프로세서에 의해 실행될 때, 하나 이상의 프로세서로 하여금, 본원에서 제공되는 방법의 구현예에 따른 동작을 수행하게 하는 명령어들이 저장된, 하나 이상의 프로세서에 커플링되는 컴퓨터 판독 가능 저장 매체를 포함한다.

[0027] 본 명세서는, 분산 시스템 내의 네트워크 노드 사이에서 합의를 달성하기 위한, 분산 시스템에서 프라이머리 노드의 체인지를 수행하기 위한, 그리고 분산 시스템 내의 네트워크 노드에 대한 복구 프로세스를 수행하기 위한 기술을 포함하는 향상된 합의 메커니즘을 개시한다. 설명되는 합의 메커니즘은 상이한 애플리케이션에서 다양한 이점을 달성할 수 있다.

[0028] 예를 들면, 하기에서 논의되는 바와 같은 합의 프로세스는, 블록체인 시스템의 동작을 향상시키고 네트워크 병목 현상을 완화하는 것을 돕는 많은 피쳐를 포함한다. 예를 들면, 설명되는 합의 프로세스는, 트랜잭션 요청을 소거 코드(erasure code; EC) 코드에 따라 다수의 EC 블록으로 변환하는 것 및 EC 블록 중 하나를 네트워크 노드의 각각으로 전송하는 것을 포함한다. EC 블록은 원래의 트랜잭션 요청보다 사이즈에서 더 작다. 따라서, 전체 트랜잭션 요청 대신 EC 블록을 네트워크 노드로 전송하는 것은, 블록체인 네트워크의 네트워크 노드 사이에서 송신되는 데이터 블록의 사이즈를 감소시키고, 그에 의해, 네트워크 대역폭을 보존하고 네트워크 부하를 감소시킨다. 이것은 또한, 네트워크 노드의 메모리 공간으로 기록되는 또는 네트워크 노드의 메모리 공간으로부터

판독되는 데이터의 사이즈를 더 감소시키고, 그에 의해, 네트워크 노드의 메모리 공간에 대한 부담을 감소시키고 전체 블록체인 시스템의 효율성을 향상시킨다.

[0029] 더구나, 본 명세서는, 합의 프로세스의 다수의 페이즈에 각각의 가중치(weight)를 할당하는 것, 다수의 페이즈의 각각의 가중치 합(weight sum)에 기초하여 가중치 합을 결정하는 것, 및 가중치 합에 기초하여 새로운 프라이머리 노드를 결정하는 것을 포함하는 에포크 체인지 프로세스를 설명한다. 라운드 로빈 방법(round robin method) 대신, 가중치 합에 기초하는 에포크 체인지 프로세스는, 장애가 없는(non-faulty) 새로운 프라이머리 노드를 적시의 방식으로 선택하는 것을 용이하게 할 수 있다. 라운드 로빈 방법과는 달리, 본 명세서에서의 에포크 체인지 프로세스는, 새로운 프라이머리 노드를 선택하기 위해 가중치 합에 의존하는데, 이것은 장애가 없는 새로운 프라이머리 노드를 찾는 데 있어서 레이턴시 또는 지연을 감소시킬 수 있다. 이것은, 블록체인 서비스를 제공함에 있어서 전체 블록체인 시스템의 효율성을 더 향상시킬 수 있다.

[0030] 또한, 본 명세서는, 새로운 프라이머리 노드가 되기 위해 신청하는 네트워크 노드에 의해 상태 요청 메시지를 전송하는 것 및 다른 네트워크 노드로부터 상태 응답 메시지를 수신하는 것과 같은 동작을 포함하는 복구 프로세스를 논의한다. 이들 동작은, 장애가 있는 네트워크 노드의 복구 프로세스가, 다른 장애가 없는 네트워크 노드 사이의 합의 프로세스의 정상 동작과 간섭하지 않도록 수행된다. 이것은 복구 프로세스의 복잡도를 감소시키는 것에 의해 장애가 있는 네트워크 노드를 복구하기 위한 컴퓨팅 및 네트워크 리소스의 절약을 용이하게 한다.

[0031] 본 명세서에 따른 방법은 본원에서 설명되는 양태 및 피쳐의 임의의 조합을 포함할 수도 있다는 것이 인정된다. 즉, 본 명세서에 따른 방법은 본원에서 구체적으로 설명되는 양태 및 피쳐의 조합으로 제한되는 것이 아니라, 제공되는 양태 및 피쳐의 임의의 조합을 또한 포함한다.

[0032] 본 명세서의 하나 이상의 구현예의 세부 사항은 첨부하는 도면 및 이하의 설명에서 기술된다. 본 명세서의 다른 피쳐 및 이점은, 상세한 설명 및 도면으로부터, 그리고 청구범위로부터 명백해질 것이다.

도면의 간단한 설명

[0033] 도 1은 본 명세서의 구현예를 실행하기 위해 사용될 수 있는 환경의 예를 묘사한다.

도 2는 본 명세서의 구현예에 따른 개념적 아키텍처의 예를 묘사한다.

도 3은 본 명세서의 구현예에 따라 실행될 수 있는 합의 프로세스의 예를 묘사한다.

도 4는 본 명세서의 구현예에 따라 실행될 수 있는 합의 프로세스의 예를 묘사한다.

도 5는 본 명세서의 구현예에 따른 해시 트리의 예를 묘사한다.

도 6은 본 명세서의 구현예에 따른, 분산 시스템의 네트워크 노드 사이에서 전달되는 메시지의 예를 묘사한다.

도 7은 본 명세서의 구현예에 따른, 분산 시스템에서 프라이머리 노드의 체인지를 수행하는 프로세스의 예를 묘사한다.

도 8은 본 명세서의 구현예에 따른, 분산 시스템에서 프라이머리 노드의 체인지를 수행하는 프로세스의 예를 묘사한다.

도 9는 본 명세서의 구현예에 따른, 분산 시스템의 네트워크 노드 사이에서 전달되는 메시지의 한 예를 묘사한다.

도 10은 본 명세서의 구현예에 따른, 분산 시스템에서 네트워크 노드의 복구 프로세스를 수행하는 프로세스의 예를 묘사한다.

도 11은 본 명세서의 구현예에 따른, 분산 시스템에서 네트워크 노드의 복구 프로세스를 수행하는 프로세스의 예를 묘사한다.

도 12는 본 명세서의 구현예에 따른, 분산 시스템의 네트워크 노드 사이에서 전달되는 메시지의 예를 묘사한다.

도 13은 본 명세서의 구현예에 따른, 합의 장치의 모듈을 예시하는 도면의 예를 묘사한다.

도 14는 본 명세서의 구현예에 따른, 합의 장치의 모듈을 예시하는 도면의 예를 묘사한다.

도 15는 본 명세서의 구현예에 따른, 프라이머리 노드 체인징 장치의 모듈을 예시하는 도면의 예를 묘사한다.

도 16은 본 명세서의 구현예에 따른, 프라이머리 노드 체인징 장치의 모듈을 예시하는 도면의 예를 묘사한다.

도 17은 본 명세서의 구현예에 따른, 복구 장치의 모듈을 예시하는 도면의 예를 묘사한다.

다양한 도면에서 동일한 참조 부호는 동일한 엘리먼트를 나타낸다.

발명을 실시하기 위한 구체적인 내용

[0034] 본 명세서의 구현예는 분산 시스템(예를 들면, 블록체인 네트워크)에서 합의 이슈를 해결하기 위한 컴퓨터 구현 방법을 포함한다. 특히, 본 명세서의 구현예는, 분산 시스템에서의 프라이머리 노드(primary node)의 체인지를 수행하는 것에 관한 것이다.

[0035] 몇몇 구현예에서, 동작은 다음의 것을 포함한다: 블록체인 네트워크의 백업 노드에 의해 에포크 체인지 - 에포크 체인지는 현재의 프라이머리 노드를 갖는 현재의 에포크로부터 새로운 프라이머리 노드를 갖는 새로운 에포크로의 체인지를 야기하고, 현재의 에포크는 프라이머리 노드를 사용하여 다수의 네트워크 노드 사이에서 합의를 달성하기 위한 합의 프로세스를 포함하고, 합의 프로세스는 세 개의 페이즈를 포함함 - 가 수행될 필요가 있다고 결정하는 것; 상기 백업 노드에 의해, 현재의 에포크에서의 합의 프로세스의 세 개의 페이즈의 각각과 연관되는 백업 노드의 각각의 가중치를 결정하는 것; 현재의 에포크에서의 세 개의 페이즈의 각각과 연관되는 백업 노드의 각각의 가중치에 기초하여 백업 노드에 의해 백업 노드에 대한 가중치 합을 결정하는 것; 가중치 합이 제1 미리 결정된 임계치에 도달한다고 결정하는 것에 응답하여 EPOCH_CHANGE 메시지 - EPOCH_CHANGE 메시지는, 현재의 프라이머리 노드를 갖는 현재의 에포크로부터, 백업 노드가 새로운 프라이머리 노드인 새로운 에포크로의 체인지에 대한 요청을 나타내고, EPOCH_CHANGE는 백업 노드의 가중치 합을 포함함 - 를 백업 노드에 의해 네트워크 노드 이외의 다수의 네트워크 노드로 전송하는 것; 백업 노드 이외의 다수의 네트워크 노드 중 적어도 하나로부터 백업 노드에 의해 적어도 하나의 NEW_EPOCH 메시지 - NEW_EPOCH 메시지는 새로운 프라이머리 노드가 될 백업 노드의 확인 응답을 나타냄 - 를 수신하는 것; 적어도 하나의 NEW_EPOCH 메시지가 유효한지의 여부를 백업 노드에 의해 검증하는 것; 적어도 하나의 NEW_EPOCH 메시지 중의 유효한 NEW_EPOCH 메시지의 수가 제2 미리 결정된 임계치를 초과하는지의 여부를 백업 노드에 의해 결정하는 것; 및 유효한 NEW_EPOCH 메시지의 수가 제2 미리 결정된 임계치를 초과한다고 결정하는 것에 응답하여, 백업 노드를 새로운 에포크에서 새로운 프라이머리 노드가 되는 것으로 백업 노드에 의해 결정하는 것.

[0036] 몇몇 구현예에서, 동작은 다음의 것을 포함한다: 다수의 네트워크 노드 중의 네트워크 노드에 의해, 네트워크 노드 이외의 백업 노드로부터 EPOCH_CHANGE 메시지 - EPOCH_CHANGE 메시지는 에포크 체인지가 수행될 필요가 있다는 표시를 포함하고, 에포크 체인지는 현재의 프라이머리 노드를 갖는 현재의 에포크로부터 새로운 프라이머리 노드를 갖는 새로운 에포크로의 체인지를 야기함 - 를 수신하는 것; 네트워크 노드에 의해, EPOCH_CHANGE 메시지 메시지가 유효한지의 여부를 검증하는 것; EPOCH_CHANGE 메시지가 유효하다고 검증하는 것에 응답하여, 네트워크 노드에 의해, NEW_EPOCH 메시지 - NEW_EPOCH 메시지는 EPOCH_CHANGE 메시지의 다이제스트를 포함함 - 를 다른 네트워크 노드로 전송하는 것; 네트워크 노드에 의해, 네트워크 노드 이외의 다수의 네트워크 노드 중 적어도 하나로부터 적어도 하나의 NEW_EPOCH 메시지를 수신하는 것; 네트워크 노드에 의해, 적어도 하나의 NEW_EPOCH 메시지가 유효한지의 여부를 검증하는 것; 백업 노드에 의해, 적어도 하나의 NEW_EPOCH 메시지 중의 유효한 NEW_EPOCH 메시지의 수가 미리 결정된 임계치를 초과하는지의 여부를 결정하는 것; 및 유효한 NEW_EPOCH 메시지의 수가 미리 결정된 임계치를 초과한다고 결정하는 것에 응답하여, 네트워크 노드에 의해, 백업 노드를 새로운 에포크에서 새로운 프라이머리 노드가 되는 것으로 결정하는 것.

[0037] 본 명세서의 구현예를 위한 추가적인 컨텍스트를 제공하기 위해, 그리고 상기에서 소개되는 바와 같이, 합의 네트워크(예를 들면, 피어 투 피어 노드로 구성됨) 또는 블록체인 네트워크로 또한 지칭될 수 있는 분산 원장 시스템(DLS)은, 참가 엔티티가 안전하게, 그리고 불변적으로 트랜잭션을 수행하는 것 및 데이터를 저장하는 것을 가능하게 한다. 용어 블록체인은, 본원에서, 어떠한 특정한 사용 사례도 참조하지 않는 DLS를 일반적으로 지칭하기 위해 사용된다. 상기에서 소개되는 바와 같이, 블록체인 네트워크는 퍼블릭 블록체인 네트워크, 프라이빗 블록체인 네트워크, 또는 컨소시엄 블록체인 네트워크로서 제공될 수 있다.

[0038] 블록체인은, 미래의 트랜잭션이 체인에 저장된 모든 이전 트랜잭션과의 일관성에 대해 검증되는 것을 허용하는 방식으로 트랜잭션을 저장하는 데이터 구조이다. 블록체인은 하나 이상의 블록을 포함한다. 체인 내의 각각의 블록은, 이전 블록의 암호화 해시(cryptographic hash)를 포함하는 것에 의해 체인 내에서 그 직전의 이전 블록에 연결된다. 각각의 블록은 또한 타임스탬프, 그 자신의 암호화 해시, 및 하나 이상의 트랜잭션을 포함한다. 블록체인 네트워크의 노드에 의해 이미 검증된 트랜잭션은 해싱되고(hashed) 머클(Merkle) 트리로 인코딩된다.

머클 트리는, 트리의 리프 노드에 있는 데이터가 해싱되고, 트리의 각각의 분기에 있는 모든 해시가 분기의 루트에서 사슬 연결되는 데이터 구조이다. 이 프로세스는, 트리 내의 모든 데이터를 나타내는 해시를 저장하는, 전체 트리의 루트까지 트리를 계속 올라간다. 트리에 저장된 트랜잭션의 것으로 주장되는 해시는, 그것이 트리의 구조와 일치하는지를 결정하는 것에 의해 신속하게 검증될 수 있다.

[0039] 반면 블록체인은 트랜잭션을 저장하기 위한 데이터 구조이지만, 블록체인 네트워크는 하나 이상의 블록체인 구조체를 관리, 업데이트, 및 유지 관리하는 컴퓨팅 노드의 네트워크이다. 상기에서 소개되는 바와 같이, 블록체인 네트워크는 퍼블릭 블록체인 네트워크, 프라이빗 블록체인 네트워크, 또는 컨소시엄 블록체인 네트워크로서 제공될 수 있다.

[0040] 퍼블릭 블록체인 네트워크에서, 합의 프로세스는 합의 네트워크의 노드에 의해 제어된다. 예를 들면, 수백, 수천, 심지어 수백만 개의 엔티티가 퍼블릭 블록체인 네트워크에서 협업할 수 있는데, 그 각각은 퍼블릭 블록체인 네트워크에서 적어도 하나의 노드를 동작시킨다. 따라서, 퍼블릭 블록체인 네트워크는 참가 엔티티와 관련하여 퍼블릭 네트워크로 간주될 수 있다. 몇몇 예에서, 블록이 유효하게 되고 블록체인 네트워크의 블록체인(분산 원장)에 추가되기 위해서는, 엔티티(노드)의 대부분이 모든 블록에 서명해야 한다. 예시적인 퍼블릭 블록체인 네트워크는, 블록체인으로 칭해지는 분산 원장을 활용하는 특정한 피어 투 피어 지불 네트워크를 포함한다. 그러나, 상기에서 언급되는 바와 같이, 용어 블록체인은, 어떠한 특정한 블록체인 네트워크에 대한 특별한 참조도 없는 분산 원장을 일반적으로 가리키기 위해 사용된다.

[0041] 일반적으로, 퍼블릭 블록체인 네트워크는 퍼블릭 트랜잭션을 지원한다. 퍼블릭 트랜잭션은 퍼블릭 블록체인 네트워크 내의 모든 노드와 공유되며, 전역적 블록체인(global blockchain)에 저장된다. 전역적 블록체인은 모든 노드에 걸쳐 복제되는 블록체인이다. 즉, 모든 노드는 전역적 블록체인에 대해 완벽한 상태로 합의한다. 합의(예를 들면, 블록체인에 대한 블록의 추가에 대한 동의)를 달성하기 위해, 합의 프로토콜이 퍼블릭 블록체인 네트워크 내에서 구현된다. 합의 프로토콜의 예는, 제한 없이, 작업 증명(proof of work; POW), 지분 증명(proof-of-stake; POS), 및 권한 증명(proof-of-authority; POA)을 포함한다. POW는 비제한적인 예로서 본원에서 추가로 참조된다.

[0042] 일반적으로, 판독 및 기록 퍼미션을 중앙 집중식으로 제어하는 특정한 엔티티에 프라이빗 블록체인 네트워크가 제공된다. 엔티티는, 어떤 노드가 블록체인 네트워크에 참가할 수 있는지를 제어하고, 따라서, 프라이빗 블록체인 네트워크는 일반적으로 네트워크에 참가하도록 누가 허용되는지에 대해, 그리고 (예를 들면, 소정의 트랜잭션에서만) 그들의 참가 레벨에 대해 제한을 두는 퍼미션이 부여된 네트워크(permissioned network)로서 일반적으로 참조된다. 다양한 타입의 액세스 제어 메커니즘이 사용될 수 있다(예를 들면, 현존하는 참가자가 새로운 엔티티를 추가하는 것에 대해 투표하고, 규제 기관이 입회를 제어할 수 있음).

[0043] 일반적으로, 컨소시엄 블록체인 네트워크는 참가하는 엔티티 사이에서 비공개이다. 컨소시엄 블록체인 네트워크에서, 합의 프로세스는 노드의 인가된 세트에 의해 제어되는데, 하나 이상의 노드는 각각의 엔티티(예를 들면, 금융 기관, 보험 회사)에 의해 운영된다. 예를 들면, 열(10) 개의 엔티티(예를 들면, 금융 기관, 보험 회사)의 컨소시엄이 컨소시엄 블록체인 네트워크를 운영할 수 있는데, 그 각각은 컨소시엄 블록체인 네트워크에서 적어도 하나의 노드를 운영한다. 따라서, 컨소시엄 블록체인 네트워크는 참가하는 엔티티와 관련하여 프라이빗 네트워크로 간주될 수 있다. 몇몇 예에서, 블록이 유효하게 되고 블록체인에 추가되기 위해서는, 각각의 엔티티(노드)는 모든 블록에 서명해야만 한다. 몇몇 예에서, 블록이 유효하게 되고 블록체인에 추가되기 위해서는, 엔티티(노드)의 적어도 서브세트(예를 들면, 적어도 7 개의 엔티티)가 모든 블록에 서명해야 한다.

[0044] 본 명세서의 구현예는, 컨소시엄 블록체인 네트워크와 관련하여 본원에서 더 상세하게 설명된다. 그러나, 본 명세서의 구현예는 임의의 적절한 타입의 블록체인 네트워크에서 실현될 수 있다는 것이 고려된다.

[0045] 본 명세서의 구현예는 상기의 컨텍스트를 고려하여 본원에서 더 상세하게 설명된다. 더 상세하게는, 그리고 상기에서 소개되는 바와 같이, 본 명세서의 구현예는 분산 시스템 내의 네트워크 노드에 대해 복구 프로세스를 수행하는 것에 관한 것이다.

[0046] 블록체인은 퍼블릭 또는 프라이빗 피어 투 피어 네트워크에서 트랜잭션을 기록하는 무단 변경 방지(tamper-proof)의, 공유된 디지털 원장이다. 원장은 네트워크 내의 모든 멤버 노드에 분산되며, 네트워크에서 발생하는 자산 트랜잭션(asset transaction)의 이력은 블록에 영구적으로 기록된다.

[0047] 합의 메커니즘은, 분산 블록체인 네트워크 내의 모든 네트워크 노드가 동일한 순서로 트랜잭션을 실행하고, 그 다음, 동일한 원장에 기록하는 것을 보장한다. 합의 모델이 다루고자 하는 하나의 이슈는 비잔틴 장애

(Byzantine failure)를 극복하는 것이다. 비잔틴 장애시, 분산 블록체인 네트워크의 서버 또는 네트워크 노드와 같은 컴포넌트가, 고장 검출 시스템에게 고장난 것 및 기능하는 것 둘 모두로 모순되게 보일 수 있는데, 이것은 상이한 관찰자에게 상이한 증상을 제시할 수 있다. 다른 네트워크 노드가 그것이 고장났다는 것을 선언하는 것 및 그것을 네트워크 밖으로 내쫓는 것은 어려운데, 그 이유는 그들이, 먼저, 어떤 네트워크 노드가 제1 장소에서 고장났는지에 관한 합의에 도달할 필요가 있기 때문이다.

[0048] 분산 시스템의 맥락에서, 비잔틴 장애 허용(Byzantine fault tolerance; BFT)은, 고장을 일으키는 또는 부정확한 정보를 다른 피어에게 전파하는 시스템의 악의적인 컴포넌트(즉, 블록체인 네트워크의 네트워크 노드)에도 불구하고 소망되는 대로 기능하는 그리고 충분한 합의에 정확하게 달성하는 분산 컴퓨터 네트워크의 능력이다. 목표는, 네트워크의 정확한 기능 및 시스템 내의 정직한 노드에 의해 달성되는 정확한 합의에 대해 이들 악의적인 노드가 갖는 영향을 완화하는 것에 의해 치명적인 시스템 장애에 대해 방어하는 것이다.

[0049] 그러나, 현존하는 BFT 메커니즘은 많은 양태에서 비효율적인 것으로 판명되었다. 예를 들면, 현존하는 BFT 메커니즘은 비잔틴 장애를 극복하려고 시도할 때 분산 블록체인 네트워크에 구현의 복잡도를 추가하였으며, 그 결과, 분산 블록체인 네트워크의 네트워크 노드 사이의 통신의 경우 레이턴시가 증가된다. 프랙티컬 비잔틴 장애 허용(Practical Byzantine Fault Tolerance; PBFT)은 현존하는 BFT 합의 메커니즘을 개선하는 것을 목표로 하는 최적화 중 하나이다. PBFT 모델은, 독립적인 노드 고장 및 특정한 독립 노드에 의해 전파되는 조작된 메시지가 있다는 가정을 통해, 비잔틴 장애(악성 노드)를 견디는 실용적인 비잔틴 상태 머신 복제(practical Byzantine state machine replication)를 제공하는 데 중점을 둔다.

[0050] PBFT 모델에서, 모든 노드는 하나의 노드가 프라이머리 노드(리더)이고 다른 것이 백업 노드로 칭해지는 상태에서 순서대로 정렬된다. 시스템 내의 모든 노드는 서로 통신하며, 목표는 대부분의 정직한 노드가 시스템의 상태의 동의에 도달하는 것이다. 노드는 서로 통신하며, 메시지가 특정한 피어 노드로부터 유래했다는 것을 증명해야 할 뿐만 아니라, 메시지가 송신 동안 수정되지 않았다는 것을 검증하는 것을 또한 필요로 한다.

[0051] PBFT 모델이 작동하기 위해서는, 네트워크 내의 악의적인 노드의 양이, 취약성의 주어진 윈도우 내에서, 동시적으로, 시스템 내의 전체 노드의 1/3 이상일 수 없다는 가정이 이루어진다. 시스템 내의 노드가 많을수록, 전체 노드의 1/3에 다가가는 수가 악의적으로 될 가능성은 수학적으로 더 낮다. 알고리즘은, 최대 (n-1)/3 개의 노드가 동시에 악의적이거나 또는 장애가 있는 한, 생존성(liveness) 및 안전성 둘 모두를 효과적으로 제공하는데, 여기서 n은 전체 노드를 나타낸다.

[0052] PBFT 합의(뷰(view)로 칭해짐)의 각각의 라운드는 4 개의 페이즈를 포함한다:

[0053] (1) 클라이언트가 서비스 동작을 호출하기 위해 리더 노드에게 요청을 전송함;

[0054] (2) 리더 노드는 요청을 백업 노드에 멀티캐스트함;

[0055] (3) 노드는 요청을 실행하고, 그 다음, 응답을 클라이언트로 전송함; 그리고

[0056] (4) 클라이언트는 동일한 결과를 갖는 상이한 노드로부터의 f + 1(f는 결함이 있을 수도 있는 노드의 최대 수를 나타냄) 개의 응답을 대기함.

[0057] 최종 결과는, 모든 정직한 노드가 기록의 순서에 대한 동의에 도달하고 그들이 그것을 수락하거나 또는 거절하는 것이다. 리더 노드는 모든 뷰(view) 동안 라운드 로빈 스킴(round robin scheme)으로 체인지되고, 심지어, 리더 노드가 요청을 멀티캐스트하지 않은 상태에서 특정한 양의 시간이 경과한 경우, 뷰 체인지(view change)로 칭해지는 프로토콜로 교체될 수 있다. 정직한 노드의 대다수는 또한, 리더가 장애가 있는지 여부를 결정할 수 있으며, 교체물(replacement)로서 라인 내의(in line) 다음 리더를 가진 상태에서 그들을 제거할 수 있다.

[0058] 그러나, PBFT 합의 메커니즘에는 약간의 제한이 있다. 예를 들면, PBFT 모델은, 노드 사이에 요구되는 번거로운 양의 통신에 기인하여 상대적으로 작은 합의 그룹 사이즈를 갖는 그것의 고전적인 형태에서 잘 작동할 수도 있다. 네트워크 노드 사이에서 송신되는 부피가 큰 블록 데이터는, 네트워크 부하 이슈를 야기하고 네트워크 병목 현상으로 이어진다. 더구나, PBFT 모델에서 인증 메시지에 대한 포맷으로서 메소드 인증 코드(method authentication codes; MAC)를 사용하는 것은, 암호 화폐 네트워크(cryptocurrency network)와 같은 큰 합의 그룹 내의 노드 사이에서 필요로 되는 통신의 양 및 MAC와는 비효율적일 수 있다. 써드파티에게 메시지의 진위성을 입증하는 내재해는 무능력성이 존재할 수 있다.

[0059] 또한, PBFT에 의해 사용되는 라운드 로빈 방법을 사용하여 리더 노드를 체인지하는 경우, 연속적인 악의적 노드와 조우하는 것은, 정직한 리더 노드를 발견함에 있어서 레이턴시 또는 지연을 도입하는 것에 의해 블록체인 서

비스에 영향을 미친다. 예를 들면, 새로운 리더 노드로서 제1 네트워크 노드를 선택할 때, 제1 네트워크 노드는 악의적인 노드일 수도 있고, 따라서, 새로운 리더 노드로서 선택될 수 없다. 라운드 로빈 방법에서, 라인 내의 제2 네트워크 노드가 새로운 리더 노드로서 선택될 수도 있다. 그러나, 제2 네트워크 노드도 또한 악성 노드인 경우, 라인 내의 다른 네트워크 노드가, 그것이 리더 노드가 되기에 적절한지의 여부로서, 검증될 것이다. 이 프로세스는, 정직한 새로운 리더 노드가 식별될 때까지 계속된다. 리더 노드의 그러한 빈번한 체인지는 블록체인 서비스에서 상당한 레이턴시를 도입한다.

[0060] 더구나, 블록체인 네트워크 내의 네트워크 노드는, 임의의 시간에 비잔틴 장애 또는 충돌 장애를 경험할 수도 있다. 예를 들면, 네트워크 노드는 악의적인 사이버 공격자에 의해 손상되어 부적절하게 거동할 수도 있다. 손상되는 네트워크 노드가 즉시 복구되지 않으면, 악의적인 사이버 공격자는, 검출되지 않으면서 네트워크 노드의 1/3 이상을 손상시키는 것에 의해, 블록체인 네트워크 및 서비스를 손상시킬 수도 있다.

[0061] 현존하는 BFT 합의 메커니즘 및 PBFT 합의 메커니즘과 연관되는 상기에서 설명된 이슈 및 관심사를 해결하기 위해, 본 명세서는, 분산 시스템 내의 네트워크 노드 사이에서 합의를 달성하기 위한, 분산 시스템 내의 프라이머리 노드의 체인지를 수행하기 위한, 그리고 분산 시스템 내의 네트워크 노드에 대해 복구 프로세스를 수행하기 위한 기술을 포함하는 향상된 합의 메커니즘을 개시한다. 설명되는 합의 메커니즘은 상이한 애플리케이션에서 다양한 이점을 달성할 수 있다.

[0062] 예를 들면, 하기에서 논의되는 바와 같은 합의 프로세스는, 블록체인 시스템의 동작을 향상시키고 네트워크 병목 현상을 완화하는 것을 돕는 많은 피쳐를 포함한다. 예를 들면, 설명되는 합의 프로세스는, 트랜잭션 요청을 소거 코드(erasure code; EC) 코드에 따라 다수의 EC 블록으로 변환하는 것 및 EC 블록 중 하나를 네트워크 노드의 각각으로 전송하는 것을 포함한다. EC 블록은 원래의 트랜잭션 요청보다 사이즈에서 더 작다. 따라서, 전체 트랜잭션 요청 대신 EC 블록을 네트워크 노드로 전송하는 것은, 블록체인 네트워크의 네트워크 노드 사이에서 송신되는 데이터 블록의 사이즈를 감소시키고, 그에 의해, 네트워크 대역폭을 보존하고 네트워크 부하를 감소시킨다. 이것은 또한, 네트워크 노드의 메모리 공간으로 기록되는 또는 네트워크 노드의 메모리 공간으로부터 판독되는 데이터의 사이즈를 더 감소시키고, 그에 의해, 네트워크 노드의 메모리 공간에 대한 부담을 감소시키고 전체 블록체인 시스템의 효율성을 향상시킨다.

[0063] 더구나, 본 명세서는, 합의 프로세스의 다수의 페이즈에 각각의 가중치를 할당하는 것, 다수의 페이즈의 각각의 가중치 합에 기초하여 가중치 합을 결정하는 것, 및 가중치 합에 기초하여 새로운 프라이머리 노드를 결정하는 것을 포함하는 에포크 체인지 프로세스를 설명한다. 라운드 로빈 방법 대신, 가중치 합에 기초하는 에포크 체인지 프로세스는, 장애가 없는 새로운 프라이머리 노드를 적시의 방식으로 선택하는 것을 용이하게 할 수 있다. 프라이머리 노드는, 프라이머리 노드를 포함하는 다수의 네트워크 노드 사이에서 합의 프로세스의 라운드를 개시하는 권한을 가진 리더 노드일 수도 있다. 블록체인 네트워크의 다른 네트워크 노드는 백업 노드로 칭해질 수 있다. 에포크 체인지 프로세스는, 새로운 프라이머리 노드에 대한 라인 내의 다수의 네트워크 노드가 장애가 있는 경우 프라이머리 노드의 빈번한 체인지를 야기하는 라운드 로빈 방법의 이슈를 해결하는 것을 도울 수 있다. 라운드 로빈 방법과는 달리, 본 명세서에서의 에포크 체인지 프로세스는, 새로운 프라이머리 노드를 선택하기 위해 가중치 합에 의존하는데, 이것은 장애가 없는 새로운 프라이머리 노드를 찾는 데 있어서 레이턴시 또는 지연을 감소시킬 수 있다. 이것은, 블록체인 서비스를 제공함에 있어서 전체 블록체인 시스템의 효율성을 더 향상시킬 수 있다.

[0064] 또한, 본 명세서는, 새로운 프라이머리 노드가 되기 위해 신청하는 네트워크 노드에 의해 상태 요청 메시지를 전송하는 것 및 다른 네트워크 노드로부터 상태 응답 메시지를 수신하는 것과 같은 동작을 포함하는 복구 프로세스를 논의한다. 이들 동작은, 장애가 있는 네트워크 노드의 복구 프로세스가, 다른 장애가 없는 네트워크 노드 사이의 합의 프로세스의 정상 동작과 간섭하지 않도록 수행된다. 이것은 복구 프로세스의 복잡도를 감소시키는 것에 의해 장애가 있는 네트워크 노드를 복구하기 위한 컴퓨팅 및 네트워크 리소스의 절약을 용이하게 한다.

[0065] 도 1은 본 명세서의 구현예를 실행하기 위해 사용될 수 있는 환경(100)의 예를 묘사한다. 몇몇 예에서, 환경(100)은 엔티티가 컨소시엄 블록체인 네트워크(102)에 참가하는 것을 가능하게 한다. 환경(100)은 컴퓨팅 디바이스 또는 시스템(106, 108), 및 네트워크(110)를 포함한다. 몇몇 예에서, 네트워크(110)는 근거리 통신망(local area network; LAN), 광역 통신망(wide area network; WAN), 인터넷, 또는 이들의 조합을 포함하고, 웹 사이트, 유저 디바이스(예를 들면, 컴퓨팅 디바이스), 및 백엔드 시스템을 연결한다. 몇몇 예에서, 네트워크(110)는 유선 및/또는 무선 통신 링크를 통해 액세스될 수 있다. 몇몇 예에서, 네트워크(110)는 컨소시엄 블록체인 네트워크(102)와의, 그리고 컨소시엄 블록체인 네트워크(102) 내에서의 통신을 가능하게 한다.

일반적으로, 네트워크(110)는 하나 이상의 통신 네트워크를 나타낸다. 몇몇 경우에 있어서, 컴퓨팅 디바이스(106, 108)는 클라우드 컴퓨팅 시스템(도시되지 않음)의 노드일 수 있거나, 또는 각각의 컴퓨팅 디바이스(106, 108)는, 네트워크에 의해 인터커넥트되며 분산 프로세싱 시스템으로서 기능하는 복수의 컴퓨터를 포함하는 개별 클라우드 컴퓨팅 시스템일 수 있다.

[0066] 묘사된 예에서, 컴퓨팅 시스템(106, 108)은, 컨소시엄 블록체인 네트워크(102)에서 노드로서의 참가를 가능하게 하는 임의의 적절한 컴퓨팅 시스템을 각각 포함할 수 있다. 예시적인 컴퓨팅 디바이스는, 제한 없이, 서버, 데스크탑 컴퓨터, 랩탑 컴퓨터, 태블릿 컴퓨팅 디바이스, 및 스마트 폰을 포함한다. 몇몇 예에서, 컴퓨팅 시스템(106, 108)은 컨소시엄 블록체인 네트워크(102)와 상호 작용하기 위한 하나 이상의 컴퓨터 구현 서비스를 호스팅한다. 예를 들면, 컴퓨팅 시스템(106)은, 제1 엔티티가 하나 이상의 다른 엔티티(예를 들면, 다른 유저)와의 자신의 트랜잭션을 관리하기 위해 사용하는 트랜잭션 관리 시스템과 같은 제1 엔티티(예를 들면, 유저 A)의 컴퓨터 구현 서비스를 호스팅할 수 있다. 컴퓨팅 시스템(108)은, 제2 엔티티가 하나 이상의 다른 엔티티(예를 들면, 다른 유저)와의 자신의 트랜잭션을 관리하기 위해 사용하는 트랜잭션 관리 시스템과 같은 제2 엔티티(예를 들면, 유저 B)의 컴퓨터 구현 서비스를 호스팅할 수 있다. 도 1의 예에서, 컨소시엄 블록체인 네트워크(102)는 노드의 피어 투 피어 네트워크로서 표현되고, 컴퓨팅 시스템(106, 108)은, 컨소시엄 블록체인 네트워크(102)에 참가하는, 제1 엔티티, 및 제2 엔티티의 노드를 각각 제공한다.

[0067] 도 2는 본 명세서의 구현예에 따른 개념적 아키텍처(200)의 예를 묘사한다. 개념적 아키텍처(200)의 예는, 참가자 A, 참가자 B, 및 참가자 C에 각각 대응하는 참가자 시스템(202, 204, 206)을 포함한다. 각각의 참가자(예를 들면, 유저, 기업)는, 복수의 노드(214)를 포함하는 피어 투 피어 네트워크로서 제공되는 블록체인 네트워크(212)에 참가하는데, 복수의 노드(214) 중 적어도 일부는 블록체인(216)에 정보를 불변하게 기록한다. 비록 단일의 블록체인(216)이 블록체인 네트워크(212) 내에서 개략적으로 묘사되지만, 본원에서 더 상세하게 설명되는 바와 같이, 블록체인(216)의 다수의 사본이 제공되며, 블록체인 네트워크(212)에 걸쳐 유지된다.

[0068] 묘사된 예에서, 각각의 참가자 시스템(202, 204, 206)은, 각각, 참가자 A, 참가자 B, 및 참가자 C에 의해, 또는 참가자 A, 참가자 B, 및 참가자 C를 대신하여 제공되고, 블록체인 네트워크 내에서 각각의 노드(214)로서 기능한다. 본원에서 사용될 때, 노드는 일반적으로, 블록체인 네트워크(212)에 연결되는 개개의 시스템(예를 들면, 컴퓨터, 서버)을 지칭하며, 각각의 참가자가 블록체인 네트워크에 참가하는 것을 가능하게 한다. 도 2의 예에서, 참가자는 각각의 노드(214)에 대응한다. 그러나, 참가자는 블록체인 네트워크(212) 내에서 다수의 노드(214)를 동작시킬 수 있고, 및/또는 다수의 참가자는 노드(214)를 공유할 수 있다는 것이 고려된다. 몇몇 예에서, 참가자 시스템(202, 204, 206)은, 프로토콜(예를 들면, 하이퍼텍스트 전송 프로토콜 보안(hypertext transfer protocol secure; HTTPS))을 사용하여, 및/또는 원격 프로시져 호출(remote procedure call; RPC)을 사용하여, 블록체인 네트워크(212)와, 또는 블록체인 네트워크(212)를 통해 통신한다.

[0069] 노드(214)는 블록체인 네트워크(212) 내에서 다양한 정도로 참가할 수 있다. 예를 들면, 몇몇 노드(214)는 (예를 들면, 블록체인(216)에 블록을 추가하는 마인더 노드(minder node)로서) 합의 프로세스에 참가할 수 있고, 한편, 다른 노드(214)는 합의 프로세스에 참가하지 않는다. 다른 예로서, 몇몇 노드(214)는 블록체인(216)의 완전한 사본을 저장하고, 한편 다른 노드(214)는 블록체인(216)의 부분의 사본만을 저장한다. 예를 들면, 데이터 액세스 특권은, 각각의 참가자가 그 각각의 시스템 내에 저장하는 블록체인 데이터를 제한할 수 있다. 도 2의 예에서, 참가자 시스템(202, 204, 206)은 블록체인(216)의 각각의 완전한 사본(216', 216", 216")을 저장한다.

[0070] 블록체인(예를 들면, 도 2의 블록체인(216))은 블록의 체인으로 구성되는데, 각각의 블록은 데이터를 저장한다. 데이터의 예는 두 개 이상의 참가자 사이의 트랜잭션을 나타내는 트랜잭션 데이터를 포함한다. 트랜잭션이 본원에서 비제한적인 예로서 사용되지만, 임의의 적절한 데이터(예를 들면, 문서(document), 이미지, 비디오, 오디오)가 블록체인에 저장될 수 있다는 것이 고려된다. 트랜잭션의 예는, 제한 없이, 가치가 있는 어떤 것(예를 들면, 자산, 제품, 서비스, 및 화폐)의 교환을 포함할 수 있다. 트랜잭션 데이터는 블록체인 내에 불변하게 저장된다. 즉, 트랜잭션 데이터는 체인지될 수 없다.

[0071] 블록에 저장하기 이전에, 트랜잭션 데이터는 해싱된다. 해싱은 트랜잭션 데이터(문자열 데이터로서 제공됨)를 고정 길이 해시 값(또한 문자열 데이터로서 제공됨)으로 변환하는 프로세스이다. 해시 값을 해시 해제하여(un-hash) 트랜잭션 데이터를 획득하는 것은 불가능하다. 해싱은, 트랜잭션 데이터에서의 약간의 변경조차도 완전히 상이한 해시 값으로 나타나는 것을 보장한다. 게다가, 그리고 상기에서 언급되는 바와 같이, 해시 값은 고정된 길이를 갖는다. 즉, 트랜잭션 데이터의 사이즈와 상관없이, 해시 값의 길이는 고정된다. 해싱은, 해시 함수를 통해 트랜잭션 데이터를 프로세싱하여 해시 값을 생성하는 것을 포함한다. 해시 함수의 예는, 제한 없이, 256

비트 해시 값을 출력하는 보안 해시 알고리즘(Secure Hash Algorithm; SHA)-256을 포함한다.

[0072] 다수의 트랜잭션의 트랜잭션 데이터는 해싱되어 블록에 저장된다. 예를 들면, 두 개의 트랜잭션의 해시 값이 제공되고, 다른 해시를 제공하기 위해 그들 자체로 해싱된다. 이 프로세스는, 블록에 저장될 모든 트랜잭션에 대해, 단일의 해시 값이 제공될 때까지, 반복된다. 이 해시 값은 머클 루트 해시(Merkle root hash)로 칭해지며, 블록의 헤더에 저장된다. 트랜잭션의 임의의 것에서의 변경은, 그 해시 값에서의 변경으로 나타날 것이고, 궁극적으로, 머클 루트 해시에서의 변경으로 나타날 것이다.

[0073] 블록은 합의 프로토콜을 통해 블록체인에 추가된다. 블록체인 네트워크 내의 다수의 노드가 합의 프로토콜에 참가하고, 블록체인에 블록이 추가되게 하기 위해 경쟁한다. 그러한 노드는 마이너(miner)(또는 마인더 노드)로 칭해진다. 상기에서 소개되는 POW가 비제한적인 예로서 사용된다.

[0074] 마이너 노드는 트랜잭션을 블록체인에 추가하기 위해 합의 프로세스를 실행한다. 비록 다수의 마이너 노드가 합의 프로세스에 참가하지만, 단지 하나의 마이너 노드만이 블록을 블록체인에 기록할 수 있다. 즉, 마이너 노드는 그들의 블록이 블록체인에 추가되게 하기 위해, 합의 프로세스에서 경쟁한다. 더 상세하게는, 마이너 노드가 (예를 들면, 만약 있다면, 블록에 포함될 수 있는 트랜잭션의 수에 대한 미리 정의된 한계까지) 트랜잭션 풀(transaction pool)로부터 보류 중인(pending) 트랜잭션을 주기적으로 수집한다. 트랜잭션 풀은 블록체인 네트워크의 참가자로부터의 트랜잭션 메시지를 포함한다. 마이너 노드는 블록을 구성하고, 트랜잭션을 블록에 추가한다. 트랜잭션을 블록에 추가하기 이전에, 마이너 노드는, 트랜잭션 중 임의의 것이 블록체인의 블록에 이미 포함되어 있는지의 여부를 체크한다. 트랜잭션이 이미 다른 블록에 포함되어 있다면, 트랜잭션은 폐기된다.

[0075] 마이너 노드는 블록 헤더를 생성하고, 블록 내의 모든 트랜잭션을 해싱하고, 블록 내의 모든 트랜잭션에 대해 단일의 해시 값이 제공될 때까지, 해시 값을 쌍으로 결합하여 또 다른 해시 값을 생성한다(머클 루트 해시). 이 해시는 블록 헤더에 추가된다. 마이너는 또한 블록체인에서 가장 최근 블록(즉, 블록체인에 추가된 마지막 블록)의 해시 값을 결정한다. 마이너 노드는 또한 논스(nonce) 값, 및 타임스탬프를 블록 헤더에 추가한다. 마이닝 프로세스에서, 마이너 노드는 필수 파라미터를 충족하는 해시 값을 찾으려고 시도한다. 마이너 노드는 필수 파라미터를 충족시키는 해시 값을 찾을 때까지 논스 값을 계속 변경한다.

[0076] 블록체인 네트워크 내의 모든 마이너는, 요구된 파라미터를 충족하는 해시 값을 찾으려고, 그리고, 이러한 방식에서, 서로 경쟁하려고 시도한다. 결국, 마이너 노드 중 하나는, 필수 파라미터를 충족하는 해시 값을 찾고, 이것을 블록체인 네트워크 내의 다른 모든 마이너 노드에 통고한다. 다른 마이너 노드는 해시 값을 검증하고, 올바른 것으로 결정되면, 블록 내의 각각의 트랜잭션을 검증하고, 블록을 수용하고, 블록을, 블록체인의 그들의 사본에 부가한다. 이러한 방식에서, 블록체인의 전역적 상태는 블록체인 네트워크 내의 모든 마이너 노드에 걸쳐 일관성이 있다. 상기에서 설명된 프로세스는 POW 합의 프로토콜이다.

[0077] 비제한적인 예가 도 2와 관련하여 제공된다. 이 예에서, 참가자 A는 일정 금액의 자금을 참가자 B에게 전송한다. 참가자 A는 트랜잭션 메시지(예를 들면, 발신인(From) 필드, 수신인(To) 필드 및 값 필드를 포함함)를 생성하고, 트랜잭션 메시지를, 트랜잭션 메시지를 트랜잭션 풀에 추가하는 블록체인 네트워크로 전송한다. 블록체인 네트워크 내의 각각의 마이너 노드는 블록을 생성하고, (예를 들면, 만약 있다면, 블록에 추가될 수 있는 트랜잭션의 수에 대한 미리 정의된 한계까지) 트랜잭션 풀로부터 모든 트랜잭션을 가져오고, 트랜잭션을 블록에 추가한다. 이 방식에서, 참가자 A에 의해 게시되는 트랜잭션은 마이너 노드의 블록에 추가된다.

[0078] 몇몇 블록체인 네트워크에서, 트랜잭션의 프라이버시를 유지하기 위해 암호화 방법(cryptography)이 구현된다. 예를 들면, 두 개의 노드가 트랜잭션을 비공개로 유지하기를 원하고, 그 결과, 블록체인 네트워크 내의 다른 노드가 트랜잭션의 세부 사항을 식별할 수 없는 경우, 노드는 트랜잭션 데이터를 암호화할 수 있다. 암호화 방법의 예는, 제한 없이, 대칭 암호화 및 비대칭 암호화를 포함한다. 대칭 암호화는, 암호화(일반 텍스트로부터 암호문을 생성함) 및 암호 해제(암호문으로부터 일반 텍스트를 생성함) 둘 모두에 대해 단일의 키를 사용하는 암호화 프로세스를 가리킨다. 대칭 암호화에서, 동일한 키가 다수의 노드에 대해 이용 가능하고, 따라서 각각의 노드는 트랜잭션 데이터를 암호화/암호 해제할 수 있다.

[0079] 비대칭 암호화는 프라이빗 키, 및 퍼블릭 키를 각각 포함하는 키 쌍을 사용하는데, 프라이빗 키는 각각의 노드에게만 알려져 있고, 퍼블릭 키는 블록체인 네트워크 내의 임의의 또는 모든 다른 노드에 알려져 있다. 노드는 다른 노드의 퍼블릭 키를 사용하여 데이터를 암호화할 수 있으며, 암호화된 데이터는 다른 노드의 프라이빗 키를 사용하여 암호 해제될 수 있다. 예를 들면, 그리고 도 2를 다시 참조하면, 참가자 A는 참가자 B의 퍼블릭 키를 사용하여 데이터를 암호화하고, 암호화된 데이터를 참가자 B에게 전송할 수 있다. 참가자 B는 자신의 프라이

빗 키를 사용하여 암호화된 데이터(암호문)를 암호 해제하고 원본 데이터(일반 텍스트)를 추출할 수 있다. 노드의 퍼블릭 키를 사용하여 암호화된 메시지는, 오로지 노드의 프라이빗 키를 사용하여 암호 해제될 수 있다.

[0080] 비대칭 암호화는 디지털 서명을 제공하기 위해 사용되는데, 이것은 트랜잭션에서의 참가자가 트랜잭션에서의 다른 참가자뿐만 아니라, 트랜잭션의 유효성을 확인하는 것을 가능하게 한다. 예를 들면, 노드는 메시지에 디지털 서명을 할 수 있고, 다른 노드는, 메시지가 참가자 A의 디지털 서명에 기초하여, 노드에 의해 전송되었다는 것을 확인할 수 있다. 디지털 서명은 또한, 메시지가 전송 중에 무단 변경되지 않는 것을 보장하기 위해 사용될 수 있다. 예를 들면, 그리고 도 2를 다시 참조하면, 참가자 A는 참가자 B에게 메시지를 전송한다. 참가자 A는 메시지의 해시를 생성하고, 그 다음, 자신의 프라이빗 키를 사용하여, 해시를 암호화하여 디지털 서명을 암호화된 해시로서 제공한다. 참가자 A는 디지털 서명을 메시지에 부가하고, 디지털 서명을 갖는 메시지를 참가자 B에게 전송한다. 참가자 B는 참가자 A의 퍼블릭 키를 사용하여 디지털 서명을 암호 해제하고, 해시를 추출한다. 참가자 B는 메시지를 해싱하고, 해시를 비교한다. 해시가 동일하면, 참가자 B는, 메시지가 실제로 참가자 A로부터 유래하였고, 무단 변경되지 않았다는 것을 확인할 수 있다.

[0081] 도 3은, 본 명세서의 구현예에 따라 실행될 수 있는 분산 시스템(예를 들면, 블록체인 네트워크(102 및 212))의 네트워크 노드(예를 들면, 노드(214)) 사이에서 합의를 달성하기 위한 프로세스(300)의 예를 묘사한다. 구체적으로, 도 3은, 본 명세서에 따른, 정상적인 경우에 합의를 달성하는 방법(300)의 예시적인 실시형태를 제시하는 도면을 예시한다. 도 3에서 예시되는 바와 같이, 합의 프로세스(300)는, 하기에서 논의되는 바와 같이, 세 개의 페이즈 또는 스테이지(310, 320, 및 330)를 포함한다.

[0082] 합의 프로세스(300)의 제1 페이즈(310)에서, 블록체인 네트워크의 프라이머리 노드(또는 리더 노드)는 제1 메시지를 다른 네트워크 노드(즉, 백업 노드)로 전송한다. 제1 메시지는 프라이머리 노드가 합의 프로세스를 개시하고 있다는 것을 나타낸다. 예를 들면, 도 3에서 예시되는 바와 같이, 프라이머리 노드(R_0)는 INITIAL 메시지를 블록체인 네트워크의 다른 네트워크 노드(R_1, R_2 및 R_3)로 전송한다. 프로세스(300)는 단지 예시적인 목적만을 위해 네 개의 네트워크 노드(R_0, R_1, R_2 및 R_3)를 포함하는 것으로 예시되며, 프로세스(300)는 임의의 적절한 수의 네트워크 노드를 포함할 수도 있다는 것을 유의한다. INITIAL 메시지의 제1 페이즈 및 포맷은, 도 4 내지 도 6과 관련하여 아래에서 더욱 상세하게 논의될 것이다.

[0083] 합의 프로세스(300)의 제2 페이즈(320)에서, 백업 노드의 각각은, 프라이머리 노드에 의해 전송되는 제1 메시지를 수신하고, 제1 메시지에 응답하여 제2 메시지를 준비하고, 제2 메시지를 다른 네트워크 노드에 멀티캐스트한다. 제2 메시지는, 백업 노드가 프라이머리 노드로부터 제1 메시지를 수신하였고 제1 메시지에 대한 응답에서 응답을 전송하고 있다는 것을 나타낸다. 예를 들면, 도 3에서 예시되는 바와 같이, 백업 노드(R_1)는, 프라이머리 노드(R_0)에 의해 전송되는 INITIAL 메시지를 수신하고, 제2 메시지의 한 예로서 ECHO 메시지로 프라이머리 노드(R_0)에 응답한다. 한편, 백업 노드(R_1)는 또한 ECHO 메시지를 다른 백업 노드, 예컨대, 백업 노드(R_2 및 R_3)로 멀티캐스트한다. 마찬가지로, 백업 노드(R_2 및 R_3) 각각은 프라이머리 노드(R_0)을 포함하는 다른 네트워크 노드로 ECHO 메시지를 멀티캐스트한다.

[0084] 예를 들면, 프라이머리 노드 또는 백업 노드와 같은 네트워크 노드가 다른 네트워크 노드로부터 ECHO 메시지를 수신하는 경우, 네트워크 노드는 ECHO 메시지의 정보를 검증할 수도 있다. ECHO 메시지의 제2 페이즈 및 포맷은, 도 4 내지 도 6과 관련하여 하기에서 더욱 상세하게 논의될 것이다.

[0085] 합의 프로세스(300)의 제3 페이즈(330)에서, 네트워크 노드의 각각은 제3 메시지를 다른 네트워크 노드로 멀티캐스트한다. 제3 메시지는, 네트워크 노드가 미리 결정된 수의 제2 메시지를 수락했다는 것을 나타낸다. 몇몇 구현예에서, 제3 메시지는 네트워크 노드가 트랜잭션을 실행할 준비가 되었다는 것을 나타낼 수 있다. 몇몇 구현예에서, 제3 메시지는, 트랜잭션이 네트워크 노드에서 성공적으로 재구성되었다는 것을 나타낼 수 있다. 예를 들면, 도 3에서 예시되는 바와 같이, 프라이머리 노드(R_0)는 ACCEPT 메시지를 백업 노드(R_1, R_2 및 R_3)로 멀티캐스트한다. 마찬가지로, 백업 노드(R_1, R_2 및 R_2) 각각은 ACCEPT 메시지를 다른 네트워크 노드로 멀티캐스트한다. 본 명세서의 몇몇 구현예에서, ACCEPT 메시지를 멀티캐스트하기 이전에, 네트워크 노드는, ACCEPT가 소거 코드(EC)에 따라 전송되고 ECHO 메시지 내의 정보가 제2 페이즈에서 수신되는 것인지의 여부를 결정한다. 제3 페이즈, EC 코드, 및 ACCEPT 메시지의 포맷은, 도 4 내지 도 6과 관련하여 하기에서 더욱 상세하게 논의될 것이다.

[0086] 네트워크 노드가 다른 네트워크 노드로부터 충분한 ACCEPT 메시지를 수신하는 경우, 네트워크 노드는, 합의가

달성되었다고 결정한다. 예를 들면, 프라이머리 노드(R_0) 또는 백업 노드(R_1, R_2 또는 R_3)가 쿼럼(예를 들면, 2f + 1, 여기서 f는 장애가 있는 네트워크 노드의 수를 나타냄) 개수의 ACCEPT 메시지를 수신하는 경우, 네트워크 노드 사이에서 합의가 자동적으로 달성된다.

[0087] 도 4는, 본 명세서의 구현예에 따라 실행될 수 있는 분산 시스템(예를 들면, 블록체인 네트워크(102 또는 212))의 네트워크 노드(예를 들면, 노드(214) 또는 노드(R_0, R_1, R_2, 및 R_3)) 사이에서 합의를 달성하기 위한 프로세스(400)의 예를 묘사한다. 몇몇 구현예에서, 프로세스(400)는 하나 이상의 컴퓨팅 디바이스를 사용하여 실행되는 하나 이상의 컴퓨터 실행 가능 프로그램을 사용하여 수행될 수도 있다. 제시의 명확성을 위해, 이하의 설명은, 이 설명의 다른 도면의 맥락에서 방법(400)을 일반적으로 설명한다. 방법(400)은, 예를 들면, 임의의 적절한 시스템, 환경, 소프트웨어, 및 하드웨어에 의해, 또는, 적절히, 시스템, 환경, 소프트웨어, 및 하드웨어의 조합에 의해 수행될 수도 있다는 것이 이해될 것이다. 몇몇 구현예에서, 방법(400)의 다양한 단계는 병렬로, 조합하여, 루프에서, 또는 임의의 순서로 실행될 수 있다.

[0088] 처음에, 프로세스(400)는 도 1 내지 도 3에서 예시되는 바와 같이 시스템(100-300)과 연계하여 구현될 수도 있다. 본 명세서의 몇몇 구현예에서, 블록체인 네트워크(102 및/또는 212)는 프라이머리 노드(404) 및 하나 이상의 백업 노드(406)를 포함한다. 블록체인 네트워크(102 및/또는 212)는 블록체인 서비스를 제공하기 위해 네트워크(110)를 통해 클라이언트 노드(402)와 같은 컴퓨팅 시스템(106 및/또는 108)과 통신한다. 클라이언트 노드(402), 프라이머리 노드(404), 및 백업 노드(406)의 각각은, 본원에서 논의되는 프로세스를 수행하도록 구성되는 특수 목적 컴퓨터 또는 다른 데이터 프로세싱 장치일 수 있다. 예를 들면, 클라이언트 노드(402)는 블록체인 네트워크와 상호 작용하는 클라이언트 단말 또는 클라이언트 디바이스로서 또한 칭해질 수 있다. 클라이언트 노드(402)는, 예를 들면, 블록체인 네트워크에 액세스하여 블록체인 네트워크와 통신하기 위한 클라이언트 애플리케이션 또는 클라이언트 소프트웨어 개발 키트(client software development kit; SDK)를 블록체인 네트워크와 관련하여 설치할 수 있다. 프라이머리 노드(404) 및 하나 이상의 백업 노드(406)는, 합의를 달성하고 블록체인 네트워크에 정보를 불변하게 기록하는 합의 노드 또는 네트워크 노드로서 또한 칭해질 수 있다.

[0089] 프로세스(400)는 클라이언트 노드(402)가 트랜잭션 요청을 생성하는 408에서 시작한다. 본 명세서의 몇몇 구현예에서, 트랜잭션 요청은 블록체인 네트워크(102 및/또는 212)에게 블록체인 서비스를 요청하는 요청을 포함할 수 있다.

[0090] 410에서, 클라이언트 노드(402)는 트랜잭션 요청을 블록체인 네트워크(102 및/또는 212)의 프라이머리 노드(404)로 멀티캐스트한다. 본 명세서의 몇몇 구현예에서, 프라이머리 노드(404)는, 클라이언트 노드(402)로부터 트랜잭션 요청을 수신한 이후, 트랜잭션 요청을 추적하기 위해 트랜잭션 요청에 시퀀스 번호를 할당한다.

[0091] 412에서, 프라이머리 노드(404)는 클라이언트 노드(402)로부터 트랜잭션 요청을 수신한 이후 다수의 EC 블록을 생성한다. 본 명세서의 몇몇 구현예에서, 프라이머리 노드(404)는 트랜잭션 요청을 사용하여 EC 코드에 따라 다수의 EC 블록을 생성한다. 예를 들면, 도 5를 참조하면, 프라이머리 노드(404)는 트랜잭션 요청(502)에 EC 코드(504)를 적용하고 EC 코드(504)를 사용하여 트랜잭션 요청(502)을 EC 메시지(506)로 변환한다. EC 코드(504)는 비트 소거의 가정 하에서의 순방향 에러 정정(forward error correction; FEC) 코드이다. EC 코드(504)는, 원래 트랜잭션 요청(502)이 EC 메시지(506)의 일부 또는 단편(fragment)으로부터 복구될 수 있도록, 원래 트랜잭션 요청(502)을 더 긴 EC 메시지(506)로 변환한다.

[0092] 본 명세서의 몇몇 구현예에서, EC 코드(504)는, 토네이도 코드(Tornado code) 또는 저밀도 패리티 체크 코드(low-density parity-check code)와 같은 근사 최적의 소거 코드(near-optimal erasure code)이다. 본 명세서의 대안적인 구현예에서, EC 코드(504)는, 분수 코드(fountain code), 온라인 코드, 루비 변환(Luby transform; LT) 코드, 또는 랩터 코드(raptor code)와 같은 근사 최적의 분수 코드(near optimal fountain code)이다. 본 명세서의 대안적인 구현예에서, EC 코드(504)는 패리티 코드, Parchive 코드, 리드 솔로몬(Reed-Solomon) 코드, 또는 재생성 코드(regenerating code)와 같은 최적의 소거 코드이다. 본 명세서의 몇몇 구현예에서, EC 코드(504)는 임의의 적절한 타입의 소거 코드일 수 있다.

[0093] 트랜잭션 요청(502)을 EC 메시지(506)로 변환한 이후, 프라이머리 노드(404)는 EC 메시지(506)를 사용하여 다수의 EC 블록(508)을 생성한다. 예를 들면, 도 5에서 예시되는 바와 같이, 프라이머리 노드(404)는 EC 메시지(506)를 분할하는 것에 의해 네 개의 EC 블록(508)인, EC 블록 A, EC 블록 B, EC 블록 C 및 EC 블록 D를 생성한다. EC 블록(508)은 예시적인 목적을 위해 네 개의 블록을 포함하는 것으로 도 5에서 예시되지만, EC 블록(508)은 임의의 적절한 수의 EC 블록(508)을 포함하는 것으로 생성될 수도 있다는 것을 유의한다. EC 블록(50

8)은 INITIAL 메시지 내의 각각의 백업 노드(406)로 전송될 것이다.

[0094] 본 명세서의 몇몇 구현예에서, EC 블록(508)은 동일한 사이즈를 갖는다. 그러나, 대안적인 구현예에서, EC 블록(508)은, 서로 상이한 사이즈를 가질 수도 있다.

[0095] 본 명세서의 몇몇 구현예에서, 프라이머리 노드(404)는 EC 블록(508)을 사용하여 해시 트리(500)(예를 들면, 머클 트리)를 생성한다. 해시 트리(500)는, 데이터 블록의 해시로 라벨링되는 다수의 리프 노드 및 그 자식 노드의 라벨의 암호화 해시로 라벨링되는 다수의 비 리프 노드(non-leaf node)를 포함한다. 예를 들면, 도 5에서 예시되는 바와 같이, 해시 트리(500)는, 그들 각각의 EC 블록(508)의 암호화 해시로서 생성되는 네 개의 리프 노드(510)인 해시 A, 해시 B, 해시 C, 및 해시 D, 그들 각각의 자식 노드(510)의 사슬 연결(concatenation)의 해시로서 생성되는 네 개의 비 리프 노드(512), 및 자신의 자식 노드(512)의 해시로서 생성되며 해시 트리(500)의 루트 해시인 비 리프 노드(514)를 포함하는 것으로 구성된다.

[0096] 해시 트리(500)는 큰 데이터 구조의 내용의 효율적이고 안전한 검증을 허용한다. 해시 트리(500)는 컴퓨터 내에서 그리고 사이에서 저장, 핸들링 및 전송되는 임의의 종류의 데이터를 검증하기 위해 사용될 수 있다. 그들은, P2P 네트워크 내의 다른 피어로부터 수신되는 데이터 블록이 손상되지 않고 그리고 변경되지 않고 수신되는 것을 보장하는 것을 도울 수 있고, 심지어, 다른 피어가 가짜 블록을 전송하지 않는 것을 체크하는 것을 도울 수 있다. 해시 트리(500)를 사용하는 데이터 블록의 검증은, 합의 프로세스(400)의 다음의 단계와 관련하여 하기에서 더욱 상세하게 논의될 것이다.

[0097] 도 4를 다시 참조하면, 프라이머리 노드(404)는 EC 블록(508) 및 해시 트리(500)를 생성한 이후 제1 메시지(예를 들면, INITIAL 메시지)를 생성한다. 제1 메시지는 프라이머리 노드가 합의 프로세스를 개시하고 있다는 것을 나타낸다. 몇몇 구현예에서, 제1 메시지의 예로서, INITIAL 메시지는 EC 블록(508) 및 해시 트리(500) 내의 정보를 사용하여 생성된다. 본 명세서의 몇몇 구현예에서, 도 6을 참조하면, INITIAL 메시지는 <에포크, tx_root_hash, ec_block_hash, ec_block, seq, j>의 포맷을 갖는데, 여기서 "에포크"는, 메시지가 전송되고 있는 합의의 라운드를 나타내고, "tx_root_hash"는 해시 트리(500)에서의 루트 해시(514)를 나타내고, "ec_block_hash"는 해시 트리(500)에서의 해시(510 및/또는 512)를 나타내고, ec_block"은 해시 트리(500)에서의 EC 블록(508)을 나타내고, "seq"는 트랜잭션 요청(502)과 연관되는 시퀀스 번호를 나타내고, "j"는 INITIAL 메시지를 생성 및 전송하는 네트워크 노드를 나타낸다. 몇몇 구현예에서, INITIAL 메시지는, 예를 들면, 추가적인 또는 상이한 필드를 포함하는 것에 의해, 상이한 포맷을 가질 수 있다.

[0098] 도 4를 다시 참조하면, 416에서, 합의 프로세스의 제1 페이즈에서, 프라이머리 노드(404)는 INITIAL 메시지를 다른 네트워크 노드(예를 들면, 백업 노드(406))로 멀티캐스트한다. 몇몇 구현예에서, 백업 노드(406)로 전송되는 INITIAL 메시지는 <에포크, tx_root_hash, ec_block_hash, ec_block, seq, j>의 포맷을 갖는다. 예를 들면, 프라이머리 노드(404)는 제1 INITIAL 메시지 <에포크 1, 해시 ABCD, {해시 B, 해시 C, 해시 D}, EC 블록 A, 1, 0>를 제1 백업 노드(406)로, 그리고 제2 INITIAL 메시지 <에포크 1, 해시 ABCD, {해시 A, 해시 C, 해시 D}, EC 블록 B, 1, 0>을 제2 백업 노드(406)로, 계속 그런 식으로 전송할 수도 있다. "ec_block"과 같은 INITIAL 메시지에서의 정보는 "ec_block_hash"와 함께 사용되어 해시 트리(500)를 재구성할 수 있다는 것을 유의한다. 예를 들면, 제1 INITIAL 메시지 <에포크 1, 해시 ABCD, {해시 B, 해시 C, 해시 D}, EC 블록 A, 1, 0>에서, EC 블록(508) "EC 블록 A"는 해싱되어 암호화 해시(510) "해시 A"를 생성할 수 있는데, 암호화 해시(510) "해시 A"는 또한 다른 해시(510) "{해시 B, 해시 C, 해시 D}"와 함께 사용되어 해시 트리(500)를 재구성한다. 재구성된 해시 트리(500)는, 합의 프로세스의 다음 단계와 관련하여 이하에서 더욱 상세하게 논의되는 바와 같이 ECHO 메시지를 검증하기 위해 사용될 것이다.

[0099] 418에서, 백업 노드(406)의 각각은, 프라이머리 노드(404)로부터 INITIAL 메시지를 수신한 이후 합의 프로세스의 제2 페이즈에서 제2 메시지(예를 들면, ECHO 메시지)를 생성한다. 제2 메시지는 백업 노드가 프라이머리 노드로부터 제1 메시지를 수신하였다는 것을 나타낸다. 제2 메시지는 제1 메시지에 응답하는 응답으로서 전송된다. 본 명세서의 몇몇 구현예에서, ECHO 메시지는, INITIAL 메시지 또는 INITIAL 메시지의 일부 및 INITIAL 메시지와 연관되는 백업 노드(406)의 서명을 포함하는 것으로서 백업 노드(406)에 의해 생성된다. 예를 들면, 백업 노드(406)는 프라이빗 키를 사용하여 INITIAL 메시지 또는 INITIAL 메시지의 다이제스트에 서명하는 것에 의해 서명을 생성할 수도 있다. 프라이빗 키 서명은, 프라이빗 키 서명을 포함하는 ECHO 메시지를 인증하기 위해, 프라이빗 키와 쌍을 이루는 퍼블릭 키를 사용하는 다른 네트워크 노드에 의해 사용될 수도 있다.

[0100] 본 명세서의 몇몇 구현예에서, 도 6을 참조하면, ECHO 메시지는 <에포크, tx_root_hash, ec_block_hash, ec_block, seq, sign_proof, j>의 포맷을 갖는데, 여기서 "에포크"는 메시지가 전송되고 있는 합의의 라운드를

나타내고, "tx_root_hash"는 해시 트리(500)에서의 루트 해시(514)를 나타내고, "ec_block_hash"는 해시 트리(500)에서의 해시(510 및/또는 512)를 나타내고, ec_block"은, 각각의 백업 노드(406)에 의해 수신되는, 해시 트리(500)에서의 EC 블록(508)을 나타내고, "seq"는 트랜잭션 요청(502)과 연관되는 시퀀스 번호를 나타내고, "sign-proof"는 INITIAL 메시지와 연관되는 백업 노드(406)의 서명을 나타내고, "j"는 ECHO 메시지를 생성 및 전송하는 네트워크 노드를 나타낸다. 몇몇 구현예에서, ECHO 메시지는, 예를 들면, 추가적인 또는 상이한 필드를 포함하는 것에 의해, 상이한 포맷을 가질 수 있다.

[0101] 도 4를 다시 참조하면, 420에서, 백업 노드(406)는 ECHO 메시지를 프라이머리 노드(404)로 전송한다. 421에서, 백업 노드(406)의 각각은 ECHO 메시지를 다른 백업 노드(406)로 전송한다. 423에서, 백업 노드(406)의 각각은 다른 백업 노드(406)로부터 ECHO 메시지를 수신할 수도 있다.

[0102] 422에서, 프라이머리 노드(404)는 백업 노드(406)에 의해 전송되는 ECHO 메시지를 검증한다. 본 명세서의 몇몇 구현예에서, 프라이머리 노드(404)는, ECHO 메시지가 해시 트리(500)에 따라 유효한지의 여부를 검증한다. 예를 들면, 프라이머리 노드(404)는 제1 백업 노드(406)로부터 제1 ECHO 메시지 <에포크 1, 해시 ABCD, {해시 B, 해시 C, 해시 D}, EC 블록 A, 1, 1>를 수신할 수도 있다. 프라이머리 노드(404)는 메시지로부터 EC 블록(508) "EC 블록 A"를 검색할 수 있고 그것을 해싱하여 암호화 해시(510) "해시 A"를 생성할 수 있다. 프라이머리 노드(404)는 또한 생성된 해시(510) "해시 A"를 메시지 내의 다른 해시(510) "{해시 B, 해시 C, 해시 D}"와 함께 사용하여 해시 트리(500)를 재구성한다. 그 다음, 프라이머리 노드(404)는 재구성된 해시 트리(500)의 루트 해시(514)를 결정하고, 그것을 ECHO 메시지 내의 루트 해시(514), 예컨대 "해시 ABCD"와 비교한다. 두 개의 루트 해시(514)가 매치하면, 프라이머리 노드(404)는 ECHO 메시지가 유효하다고 결정한다. 프라이머리 노드(404)는 유효한 ECHO 메시지를 저장할 수도 있고 무효한 것으로 결정되는 ECHO 메시지를 폐기할 수도 있다.

[0103] 424에서, 프라이머리 노드(404)는, 유효한 ECHO 메시지의 수가 미리 결정된 임계치를 초과하는지의 여부를 결정한다. 본 명세서의 몇몇 구현예에서, 프라이머리 노드(404)는 유효한 ECHO 메시지의 수가 쿼럼 개수(n-f 또는 2f + 1)에 도달하는지의 여부를 결정하는데, 여기서 n은 네트워크 노드의 총 수이고, f는, 네트워크가 견딜 수 있는 장애가 있는 노드의 최대 수이다.

[0104] 426에서, 프라이머리 노드(404)는 유효한 ECHO 메시지의 수가 쿼럼 개수에 도달한다고 결정하는 것에 응답하여 트랜잭션 요청(502)을 재구성한다. 본 명세서의 몇몇 구현예에서, 프라이머리 노드(404)는 EC 코드에 따라 유효한 ECHO 메시지의 적어도 서브세트에 기초하여 트랜잭션 요청(502)을 재구성한다. 예를 들면, 프라이머리 노드(404)는, 유효한 ECHO 메시지의 쿼럼 개수(예를 들면, n - f 또는 2f + 1) 내에 있는 EC 블록(508)의 n - 2f 또는 f + 1의 개수를 검색할 수도 있고, 검색된 EC 블록(508)을 사용하여 EC 코드(504)에 따라 트랜잭션 요청(502)을 재구성할 수도 있다.

[0105] 428에서, 합의 프로세스의 제3 페이즈에서, 프라이머리 노드(404)는, 트랜잭션 요청(502)이 성공적으로 재구성되었다고 결정하는 것에 응답하여 제3 메시지(예를 들면, ACCPET 메시지)를 생성한다. 제3 메시지는, 네트워크 노드가 미리 결정된 수의 제2 메시지를 수락했다는 것을 나타낸다. 몇몇 구현예에서, 제3 메시지는 네트워크 노드가 트랜잭션을 실행할 준비가 되었다는 것을 나타낼 수 있다. 몇몇 구현예에서, 제3 메시지는, 트랜잭션이 네트워크 노드에서 성공적으로 재구성되었다는 것을 나타낼 수 있다. 예를 들면, ACCPET 메시지는 트랜잭션 요청(502)이 성공적으로 재구성되었다는 것을 다른 네트워크 노드에게 나타내기 위해 사용될 수 있다. 프라이머리 노드(404)가 트랜잭션 요청(502)을 재구성하는 데 실패하면, 프라이머리 노드(404)는 ACCEPT 메시지를 생성하지 않을 수도 있다.

[0106] 본 명세서의 몇몇 구현예에서, 도 6을 참조하면, ACCEPT 메시지는 <에포크, tx_root_hash, seq, sign_proofs, j>의 포맷을 갖는데, 여기서 "에포크"는 메시지가 전송되고 있는 합의의 라운드를 나타내고, "tx_root_hash"는 해시 트리(500)에서의 루트 해시(514)를 나타내고, "seq"는 트랜잭션 요청(502)과 연관되는 시퀀스 번호를 나타내고, "sign-proofs"는 유효한 ECHO 메시지에서의 서명의 세트를 나타내고, "j"는 ACCEPT 메시지를 생성 및 전송하는 네트워크 노드를 나타낸다. 몇몇 구현예에서, ACCEPT 메시지는, 예를 들면, 추가적인 또는 상이한 필드를 포함하는 것에 의해, 상이한 포맷을 가질 수 있다.

[0107] 도 4를 다시 참조하면, 430에서, 프라이머리 노드(404)는 ACCPET 메시지를 백업 노드(406)로 전송한다.

[0108] 프라이머리 노드(404)와 마찬가지로, 백업 노드(406)의 각각은, 예를 들면, 프라이머리 노드(404)와 유사한 단계 422-428를 수행하는 것에 의해 트랜잭션 요청을 재구성할 수 있다. 432에서, 백업 노드(406)의 각각은, 트랜잭션 요청(502)이 백업 노드(406)에 의해 성공적으로 재구성되었다는 결정에 응답하여 ACCEPT 메시지를 생성한

다. 몇몇 구현예에서, 프라이머리 노드(404) 및 백업 노드(406)는, 예를 들면, 도 3에서 나타내어지는 바와 같이, 단계 422-428를 병렬 방식으로 수행할 수도 있다.

[0109] 434에서, 백업 노드(406)는 ACCEPT 메시지를 프라이머리 노드(404)로 전송한다. 한편, 백업 노드(406)의 각각은 ACCEPT 메시지를 다른 백업 노드(406)에 전송할 수도 있다.

[0110] 436에서, 프라이머리 노드(404)는, ACCEPT 메시지의 수가 미리 결정된 임계치를 초과한다고 결정하는 것에 응답하여 트랜잭션 요청(502)을 실행한다. 본 명세서의 몇몇 구현예에서, 프라이머리 노드(404)는, 수신된 ACCEPT 메시지가 동일한지의 여부 및 동일한 ACCEPT 메시지의 수가 쿼럼 개수(예를 들면, 2f + 1)에 도달하는지의 여부를 결정한다. 동일한 ACCEPT 메시지의 수가 쿼럼 개수에 도달하는 경우, 프라이머리 노드(404)는, 모든 네트워크 노드 사이에서 합의가 달성되었다고 결정하고, 그 다음, 트랜잭션 요청(502)을 로컬하게 실행한다. 본 명세서의 몇몇 구현예에서, 동일한 ACCEPT 메시지의 수가 쿼럼 개수에 도달하지 않는다는 것을 프라이머리 노드(404)가 결정하는 경우, 프라이머리 노드(404)는 모든 네트워크 노드 사이에서 합의가 달성되지 않았다고 결정하고, 그 다음, 트랜잭션 요청(502)의 실행을 억제한다.

[0111] 본 명세서의 몇몇 구현예에서, 백업 노드(406)의 각각은, 트랜잭션 요청(502)을 실행하기 이전에, 상기의 436에서 설명되는 바와 같이, 프라이머리 노드(404)에 의해 수행되는 동일한 동작을 수행할 수도 있다. 백업 노드(406)가 자신이 수신하는 ACCEPT 메시지가 미리 결정된 임계치를 초과한다고 결정하는 경우, 백업 노드(406)는, 네트워크 노드 사이에서 합의가 달성되었다고 결정하고 트랜잭션 요청(502)을 로컬하게 실행한다. 본 명세서의 몇몇 구현예에서, 동일한 ACCEPT 메시지의 수가 쿼럼 개수에 도달하지 않는다는 것을 백업 노드(406)가 결정하는 경우, 백업 노드(406)는 모든 네트워크 노드 사이에서 합의가 달성되지 않았다고 결정하고, 그 다음, 트랜잭션 요청(502)의 실행을 억제한다.

[0112] 438에서, 프라이머리 노드(404)는 트랜잭션 요청(502)을 실행한 이후 트랜잭션 결과를 클라이언트 노드(402)로 전송한다. 트랜잭션 요청(502)을 로컬하게 성공적으로 실행한 백업 노드(406)는 또한, 그들 각각의 트랜잭션 결과를 클라이언트 노드(402)로 전송할 수도 있다.

[0113] 상기에서 논의되는 바와 같은 합의 프로세스는, 전체 블록체인 시스템의 동작을 향상시키고 네트워크 병목 현상을 완화하는 것을 돕는 많은 피쳐를 포함한다. 예를 들면, 본 명세서에서의 합의 프로세스는, 트랜잭션 요청을 사용하여 EC 코드에 따라 다수의 EC 블록을 생성하는 것 및 EC 블록 중 하나를 네트워크 노드의 각각으로 전송하는 것을 포함한다. EC 블록은 원래의 트랜잭션 요청보다 사이즈에서 더 작다. 따라서, 전체 트랜잭션 요청 대신 EC 블록을 네트워크 노드로 전송하는 것은, 블록체인 네트워크의 네트워크 노드 사이에서 송신되는 데이터 블록의 사이즈를 감소시키고, 그에 의해, 네트워크 대역폭을 보존하고 네트워크 부하를 감소시킨다. 이것은 또한, 네트워크 노드의 메모리 공간으로 기록되는 또는 네트워크 노드의 메모리 공간으로부터 판독되는 데이터의 사이즈를 더 감소시키고, 그에 의해, 네트워크 노드의 메모리 공간에 대한 부담을 감소시키고 전체 블록체인 시스템의 효율성을 향상시킨다.

[0114] 합의 프로세스 동안, 백업 노드는 프라이머리 노드로부터의 요청을 대기하고 있다. 그러나, 프라이머리 노드는 비잔틴 장애 또는 충돌 장애와 조우할 수도 있고, 그 결과, 프라이머리 노드는 미리 결정된 시간 윈도우 내에서 요청을 브로드캐스트할 수 없게 된다. 프라이머리 노드가 요청을 멀티캐스트하지 않은 상태에서 특정한 양의 시간이 경과한 경우, 백업 노드가 실행할 요청을 무기한으로 대기하는 것을 방지하기 위해 새로운 프라이머리 노드가 선택될 필요가 있을 수도 있다.

[0115] 도 7은, 본 명세서의 구현예에 따라 실행될 수 있는, 분산 시스템(예를 들면, 블록체인 네트워크(102 및 212))의 프라이머리 노드(예를 들면, 노드(214 또는 404))의 체인지를 수행하기 위한 프로세스(700)의 예를 묘사한다. 구체적으로, 도 7은 본 명세서에 따른, 프라이머리 노드의 체인지를 수행하는 방법(700)의 예시적인 실시형태를 제시하는 도면을 예시한다. 몇몇 구현예에서, 프라이머리 노드는, 프라이머리 노드가 리더인 상태에서의 합의 프로세스를 포함하는 에포크와 연관된다. 프라이머리 노드의 체인지는 에포크의 체인지로 나타날 수 있다.

[0116] 몇몇 구현예에서, 현재의 에포크의 프라이머리 노드가 체인지될 필요가 있다고 결정하는 것에 응답하여, 블록체인 네트워크의 백업 노드는 제1 메시지를 다른 네트워크 노드로 전송한다. 제1 메시지는 백업 노드가 새로운 에포크에서 새로운 프라이머리 노드가 되기를 원한다는 것을 나타낸다. 예를 들면, 도 7에서 예시되는 바와 같이, 백업 노드(R_0)는, 현재 프라이머리 노드가 장애가 있다는 것 및 에포크 체인지가 수행될 필요가 있다고 백업 노드(R_0)가 결정하는 것에 응답하여, EPOCH_CHANGE 메시지를, 블록체인 네트워크 내의 다른 네트워크 노드(R_1, R_2

및 R_3)로 전송한다. EPOCH_CHANGE 메시지는 백업 노드(R_0)가 새로운 프라이머리 노드가 되기를 신청한다는 것을 나타내는 제1 메시지의 예이다. 에포크 체인지는, 현재의 프라이머리 노드를 갖는 현재의 에포크로부터, 새로운 프라이머리 노드를 갖는 새로운 에포크로의 체인지를 야기할 수 있다. 프로세스(700)는 단지 예시적인 목적을 위해 네 개의 네트워크 노드와 연계하여 구현되는 것으로 예시된다는 것을 유의한다. 프로세스(700)는 임의의 적절한 수의 네트워크 노드와 연계하여 구현될 수도 있다.

[0117] 그 다음, 네트워크 노드의 각각은, 백업 노드에 의해 전송되는 제1 메시지를 수신하고, 제1 메시지에 응답하여 제2 메시지를 준비하고, 제2 메시지를 다른 네트워크 노드로 멀티캐스트한다. 예를 들면, 도 7에서 예시되는 바와 같이, 네트워크 노드(R_1)는, 백업 노드(R_0)에 의해 전송되는 EPOCH_CHANGE 메시지를 수신하고, 백업 노드(R_0)가 새로운 프라이머리 노드가 될 수도 있다는 확인 응답(acknowledgement)을 나타내는 NEW_EPOCH 메시지로 백업 노드(R_0)에 응답한다. 한편, 네트워크 노드(R_1)는 또한, NEW_EPOCH 메시지를, 네트워크 노드(R_2 및 R_3)와 같은 다른 네트워크 노드로 멀티캐스트한다. 마찬가지로, 네트워크 노드(R_2 및 R_3) 각각은 NEW_EPOCH 메시지를 다른 네트워크 노드로 멀티캐스트한다.

[0118] 상기에서 논의되는 바와 같은 에포크 체인지 프로세스, EPOCH_CHANGE 메시지의 포맷, 및 NEW_EPOCH 메시지의 포맷이, 도 8 및 도 9와 관련하여 하기에서 더욱 상세하게 논의될 것이다.

[0119] 도 8은, 본 명세서의 구현예에 따라 실행될 수 있는, 분산 시스템(예를 들면, 블록체인 네트워크(102 또는 212))에서 프라이머리 노드의 체인지를 수행하기 위한 프로세스(800)의 예를 묘사한다. 몇몇 구현예에서, 프로세스(800)는 하나 이상의 컴퓨팅 디바이스를 사용하여 실행되는 하나 이상의 컴퓨터 실행 가능 프로그램을 사용하여 수행될 수도 있다. 제시의 명확성을 위해, 이하의 설명은, 이 설명의 다른 도면의 맥락에서 방법(800)을 일반적으로 설명한다. 방법(800)은, 예를 들면, 임의의 적절한 시스템, 환경, 소프트웨어, 및 하드웨어에 의해, 또는, 적절히, 시스템, 환경, 소프트웨어, 및 하드웨어의 조합에 의해 수행될 수도 있다는 것이 이해될 것이다. 몇몇 구현예에서, 방법(800)의 다양한 단계는 병렬로, 조합하여, 루프에서, 또는 임의의 순서로 실행될 수 있다.

[0120] 프로세스(800)는 806에서 시작하는데, 여기서, 백업 노드(802)는 에포크 체인지가 수행될 필요가 있다고 결정한다. 본원에서 논의되는 에포크 체인지는, 현재의 프라이머리 노드를 갖는 현재의 에포크로부터, 새로운 프라이머리 노드를 갖는 새로운 에포크로의 체인지를 야기한다. 예시적인 에포크는, 도 3 내지 도 6과 관련하여 상기에서 논의된 바와 같이 프라이머리 노드를 사용하여 다수의 네트워크 노드 사이에서 합의를 달성하기 위한 합의 프로세스(예를 들면, 합의 프로세스(300 또는 400))를 포함할 수 있다.

[0121] 본 명세서의 몇몇 구현예에서, 백업 노드(802)는, 현재의 프라이머리 노드로부터 요청을 수신하지 않은 상태에서 특정한 양의 시간이 경과한 이후 백업 노드(802)가 현재의 프라이머리 노드로부터의 요청을 여전히 대기하고 있다고 결정하는 것에 응답하여, 에포크 체인지가 수행될 필요가 있다고 결정한다. 예를 들면, 현재의 프라이머리 노드는 비잔틴 장애 또는 충돌 장애와 조우할 수도 있고, 그 결과, 현재의 프라이머리 노드는 미리 결정된 시간 윈도우 내에서 요청을 멀티캐스트할 수 없게 된다. 따라서, 에포크 체인지는, 백업 노드가 실행할 요청을 무기한으로 대기하는 것을 방지하는 타임아웃(timeout)에 의해 트리거된다. 본원에서 논의되는 에포크 체인지 프로세스는, 프라이머리 노드가 실패하는 경우에 시스템이 진행하는 것을 허용하는 것에 의해 생존성을 제공하고 네트워크 레이턴시를 감소시킨다.

[0122] 808에서, 백업 노드(802)는, 현재의 에포크에서의 합의 프로세스의 페이즈의 각각과 연관되는 백업 노드(802)의 각각의 가중치를 결정한다. 몇몇 구현예에서, 합의 프로세스는 도 3 내지 도 6과 관련하여 상기에서 설명되는 바와 같은 세 개의 페이즈를 포함한다. 가중치는, 새로운 에포크에서 새로운 프라이머리 노드가 될 백업 노드(802)의 자격 부여의 메트릭이다.

[0123] 본 명세서의 몇몇 구현예에서, 백업 노드(802)는, 현재의 에포크에서의 합의 프로세스의 제1 페이즈가 제1 값이 되도록 백업 노드(802)의 가중치를 결정한다. 예를 들면, 백업 노드(802)가 합의 프로세스의 제1 페이즈(예를 들면, 합의 프로세스(300)의 제1 페이즈(310))에 진입한 경우, 백업 노드(802)는 10 %의 초기 가중치를 할당받을 수도 있다. 본 명세서의 대안적인 구현예에서, 백업 노드(802)는 현재의 합의 프로세스의 제1 페이즈에 대해 임의의 적절한 가중 값을 백업 노드(802)에 할당할 수도 있다.

[0124] 본 명세서의 몇몇 구현예에서, 백업 노드(802)는 현재의 에포크에서의 합의 프로세스의 제2 페이즈(예를 들면, 합의 프로세스(300)의 제2 페이즈(320))에 대한 백업 노드(802)의 가중치를 쿼럼 검증 프로세스에 기초하여 결

정한다. 쿼럼 검증 프로세스는, 합의 프로세스의 제2 페이즈에서 백업 노드(802)가 다른 네트워크 노드로부터 미리 결정된 수(예를 들면, 2f + 1)의 ECHO 메시지를 수신하는지의 여부를 결정하는 것에 의해 수행된다.

[0125] 본 명세서의 몇몇 구현예에서, 백업 노드(802)가 쿼럼 검증에 실패하는 경우(예를 들면, 백업 노드(802)가 미리 결정된 임계치보다 더 낮은 수의 ECHO 메시지를 수신하는 경우), 백업 노드(802)는 합의 프로세스의 제2 페이즈에 대한 백업 노드(802)의 가중치를 제1 값이 되는 것으로 결정할 수도 있다. 백업 노드(802)가 쿼럼 검증을 통과하는 경우(예를 들면, 백업 노드(802)가 미리 결정된 임계치와 동일한 또는 미리 결정된 임계치를 초과하는 수의 ECHO 메시지를 수신하는 경우), 백업 노드(802)는 합의 프로세스의 제2 페이즈에 대한 백업 노드(802)의 가중치를 제2 값이 되는 것으로 결정할 수도 있다. 본 명세서의 몇몇 구현예에서, 제2 값은 제1 값보다 더 큰 것으로 결정된다. 예를 들면, 백업 노드(802)가 쿼럼 검증에 실패하는 경우, 백업 노드(802)는 합의 프로세스의 제2 페이즈에 대해 제로의 가중 값을 할당받을 수 있다. 백업 노드(802)가 쿼럼 검증을 통과하는 경우, 백업 노드(802)는 합의 프로세스의 제2 페이즈에 대해 백업 노드(802)에 대한 45 %의 가중 값을 할당받을 수 있다. 그러나, 본 명세서의 대안적인 구현예에서, 백업 노드(802)는 현재의 에포크에서의 합의 프로세스의 제2 페이즈에 대해 임의의 적절한 값을 백업 노드(802)에 할당할 수도 있다.

[0126] 본 명세서의 몇몇 구현예에서, 쿼럼 검증은, 합의 프로세스의 제2 페이즈 동안 백업 노드(802)가 다른 네트워크 노드로부터 수신하는 ECHO 메시지가 유효한지의 여부를 검증하는 것을 더 포함한다. 예를 들면, 백업 노드(802)는 퍼블릭 키를 사용하여 ECHO 메시지 내의 프라이빗 키 서명을 인증하여 ECHO 메시지가 유효한지의 여부를 결정할 수도 있다.

[0127] 제2 페이즈에 대한 가중치를 결정하는 것과 유사하게, 몇몇 구현예에서, 백업 노드(802)는, 현재의 에포크에서의 합의 프로세스의 제3 페이즈(예를 들면, 합의 프로세스(300)의 제3 페이즈(330))에 대한 백업 노드(802)의 가중치를 쿼럼 검증 프로세스에 기초하여 결정한다. 쿼럼 검증 프로세스는, 백업 노드(802)가 현재의 에포크에서의 합의 프로세스의 제3 페이즈에서 다른 네트워크 노드로부터 미리 결정된 수(예를 들면, 2f + 1)의 수락 메시지(accept message)를 수신하는지의 여부를 결정하는 것에 의해 수행된다. 다른 네트워크 노드로부터의 수락 메시지의 각각은, 다른 네트워크 노드의 각각이 미리 결정된 수의 ECHO 메시지를 수락하였다는 것을 나타낸다. 수락 메시지는, 예를 들면, 합의 프로세스(300)의 제3 페이즈(330)와 관련하여 상기에서 설명되는 ACCEPT 메시지일 수 있다.

[0128] 본 명세서의 몇몇 구현예에서, 백업 노드(802)가 쿼럼 검증에 실패하는 경우(예를 들면, 백업 노드(802)가 미리 결정된 임계치보다 더 낮은 수의 ACCEPT 메시지를 수신하는 경우), 백업 노드(802)는 합의 프로세스의 제3 페이즈에 대한 백업 노드(802)의 가중치를 제1 값이 되는 것으로 결정할 수도 있다. 백업 노드(802)가 쿼럼 검증을 통과하는 경우(예를 들면, 백업 노드(802)가 미리 결정된 임계치와 동일한 또는 미리 결정된 임계치를 초과하는 수의 ACCEPT 메시지를 수신하는 경우), 백업 노드(802)는 합의 프로세스의 제3 페이즈에 대한 백업 노드(802)의 가중치를 제2 값이 되는 것으로 결정할 수도 있다. 몇몇 구현예에서, 제2 값은 제1 값보다 더 큰 것으로 결정된다. 예를 들면, 백업 노드(802)가 쿼럼 검증에 실패하는 경우, 백업 노드(802)는 합의 프로세스의 제3 페이즈에 대해 백업 노드(802)에 대한 제로의 가중 값을 할당받을 수 있다. 백업 노드(802)가 쿼럼 검증을 통과하는 경우, 백업 노드(802)는 합의 프로세스의 제3 페이즈에 대해 백업 노드(802)에 대한45 %의 가중 값을 할당받을 수 있다. 그러나, 본 명세서의 대안적인 구현예에서, 백업 노드(802)는 현재의 에포크에서의 합의 프로세스의 제3 페이즈에 대해 임의의 적절한 값을 백업 노드(802)에 할당할 수도 있다.

[0129] 810에서, 현재의 에포크에서의 합의 프로세스의 페이즈에 대한 백업 노드(802)의 각각의 가중치를 결정한 이후, 백업 노드(802)는 각각의 가중치에 기초하여 합의 프로세스에 대한 백업 노드(802)의 가중치 합을 결정한다. 본 명세서의 몇몇 구현예에서, 가중치 합은, 현재의 에포크에서의 합의 프로세스의 페이즈의 각각과 연관되는 백업 노드의 각각의 합의 합이다. 예를 들면, 백업 노드(802)가 제1 페이즈에 대한 백업 노드(802)의 제1 가중 값을 10 %, 제2 페이즈에 대한 백업 노드(802)의 제2 가중 값을 45 %, 그리고 제3 페이즈에 대한 백업 노드(802)의 제3 가중 값을 45 %로 결정한 경우, 백업 노드(802)는 가중치 합을 100 %로 결정한다. 다른 예로서, 백업 노드(802)가 제1 페이즈에 대한 백업 노드(802)의 제1 가중 값을 10 %, 제2 페이즈에 대한 백업 노드(802)의 제2 가중 값을 45 %, 그리고 제3 페이즈에 대한 백업 노드(802)의 제3 가중 값을 0으로 결정한 경우, 백업 노드(802)는 가중치 합을 55 %로 결정한다.

[0130] 812에서, 810에서 결정된 가중치 합이 미리 결정된 임계치에 도달한다는 것 또는 미리 결정된 임계치를 초과한다는 것을 백업 노드(802)가 결정하는 경우, 백업 노드(802)는 EPOCH_CHANGE 메시지를 다른 네트워크 노드(804)로 전송한다. 예를 들면, 백업 노드(802)는, 810에서 결정되는 바와 같은 가중치 합이 100 %에 도달한다.

POCH_CHANGE 메시지는, 현재의 프라이머리 노드를 갖는 현재의 에포크로부터, 백업 노드가 새로운 프라이머리 노드인 새로운 에포크로의 체인지에 대한 요청을 나타낸다.

[0131] 본 명세서의 몇몇 구현예에서, 도 9를 참조하면, EPOCH_CHANGE 메시지는 <가중치, 에포크 + 1, ECHO{}, ACCEPT{}, j>의 포맷을 갖는데, 여기서, "가중치"는 합의 프로세스에 대한 810에서 앞서 결정되는 바와 같은 백업 노드(802)의 가중치 합을 나타내고, "에포크 + 1"은 새로운 프라이머리 노드와 연관되는 새로운 합의의 라운드(즉, 새로운 에포크)를 나타내고, "ECHO{}"는 합의 프로세스의 제2 페이즈 동안 백업 노드(802)가 수신하는 ECHO 메시지의 세트를 나타내고, "ACCEPT{}"는 합의 프로세스의 제3 페이즈 동안 백업 노드(802)가 수신하는 ACCEPT 메시지의 세트를 나타내고, "j"는 EPOCH_CHANGE 메시지를 생성 및 전송하는 네트워크 노드(예를 들면, 백업 노드(802))를 나타낸다. 몇몇 구현예에서, EPOCH_CHANGE 메시지는, 예를 들면, 추가적인 또는 상이한 필드를 포함하는 것에 의해, 상이한 포맷을 가질 수 있다.

[0132] 도 8을 다시 참조하면, 814에서, 백업 노드(802) 이외의 네트워크 노드(804)는, 백업 노드(802)에 의해 전송되는 EPOCH_CHANGE 메시지를 검증한다. 몇몇 구현예에서, 네트워크 노드(804)의 각각은, EPOCH_CHANGE 메시지 내의 가중치 합이 유효한지의 여부를 검증하는 것에 의해 EPOCH_CHANGE 메시지가 유효한지의 여부를 검증한다. 몇몇 구현예에서, EPOCH_CHANGE 메시지 내의 가중치 합이 유효한지의 여부를 검증하는 것은, EPOCH_CHANGE 메시지에 포함되는 ECHO 메시지 내의 서명 세트가 유효한지의 여부를 검증하는 것을 포함한다. 예를 들면, 네트워크 노드(804)의 각각은, 퍼블릭 키를 사용하여 EPOCH_CHANGE 메시지에 포함되는 ECHO 메시지 내의 프라이빗 키 서명의 세트를 인증할 수도 있다.

[0133] 816에서, 네트워크 노드(804)의 각각은, 백업 노드(802)에 의해 전송되는 EPOCH_CHANGE 메시지가 유효하다고 검증하는 것에 응답하여, NEW_EPOCH 메시지를 백업 노드(802)로 전송한다. NEW_EPOCH 메시지는, 새로운 프라이머리 노드가 될 백업 노드의 확인 응답을 나타낸다. 예를 들면, 네트워크 노드(804)에 의해 전송되는 NEW_EPOCH 메시지는, 백업 노드(802)가 새로운 에포크에서 새로운 프라이머리 노드가 될 것이다는 것을 네트워크 노드(804)가 승인하는 표시를 포함한다. 한편, 네트워크 노드(804)의 각각은 또한 NEW_EPOCH 메시지를 다른 네트워크 노드(804)로 전송한다.

[0134] 도 9를 참조하면, NEW_EPOCH 메시지는 <에포크 + 1, i, j, seq, ec_digest>의 포맷을 갖는 것으로 생성되고 있는데, 여기서 "epoch+1"은 새로운 프라이머리 노드와 연관되는 새로운 합의의 라운드(즉, 새로운 에포크)를 나타내고, "i"는 새로운 에포크에서의 새로운 프라이머리 노드를 나타내고, "j"는 NEW_EPOCH 메시지를 전송하는 네트워크 노드(804)를 나타내고, "ec_digest"는 EPOCH_CHANGE 메시지의 다이제스트를 나타낸다. 몇몇 구현예에서, EPOCH_CHANGE 메시지의 다이제스트는 EPOCH_CHANGE 메시지의 해시 값을 포함한다. 몇몇 구현예에서, NEW_EPOCH 메시지는, 예를 들면, 추가적인 또는 상이한 필드를 포함하는 것에 의해, 상이한 포맷을 가질 수 있다.

[0135] 도 8을 다시 참조하면, 818에서, 백업 노드(802)는, 네트워크 노드(804)에 의해 전송되는 NEW_EPOCH 메시지가 유효한지의 여부를 검증한다. 몇몇 구현예에서, 백업 노드(802)는, NEW_EPOCH 메시지 내의 EPOCH_CHANGE 메시지의 다이제스트가 유효한지의 여부를 검증하는 것에 의해 NEW_EPOCH 메시지를 검증한다. 다이제스트가 EPOCH_CHANGE 메시지의 정보를 포함하기 때문에, 다이제스트는 또한 EPOCH_CHANGE 메시지 내의 서명을 포함한다. 백업 노드(802)는, EPOCH_CHANGE 메시지 내의 서명의 세트가 유효한지의 여부를 검증하는 것에 의해 EPOCH_CHANGE 메시지의 다이제스트를 검증할 수 있다.

[0136] 820에서, 백업 노드(802)는, 818에서 결정되는 바와 같은 유효한 NEW_EPOCH 메시지의 수가 미리 결정된 임계치를 초과하는지의 여부를 결정한다. 몇몇 구현예에서, 미리 결정된 임계치는 쿼럼 개수(예를 들면, 2f + 1)이다.

[0137] 822에서, 백업 노드(802)는, 결정되는 바와 같은 유효한 NEW_EPOCH 메시지의 수가 미리 결정된 임계치를 초과한다고 결정하는 것에 응답하여, 백업 노드(802)를 새로운 에포크에서 새로운 프라이머리 노드가 되는 것으로 결정한다. 네트워크 노드(804)의 각각은 백업 노드(802)가 수행하는 것과 동일한 단계 818-822를 수행하고, 네트워크 노드(804) 및 백업 노드(802)는 단계 818-822를 병렬 방식으로 수행할 수도 있다는 것을 유의한다. 예를 들면, 네트워크 노드(804)의 각각은, 다른 네트워크 노드(804)로부터 전송되는 NEW_EPOCH 메시지의 세트를 검증할 수도 있고, 유효한 NEW_EPOCH 메시지의 수가 미리 결정된 임계치를 초과하는지의 여부를 결정할 수도 있고, 새로운 프라이머리 노드를 결정할 수도 있다.

[0138] 상기에서 논의되는 바와 같은 에포크 체인지 프로세스(예를 들면, 프로세스(700 또는 800))는, 전체 블록체인 시스템의 동작을 향상시키며 네트워크 병목 현상을 완화시키는 데 도움을 주는 많은 피처를 포함한다. 예를 들

면, 본 명세서에서의 에포크 체인지 프로세스는, 각각의 가중치를, 합의 프로세스의 세 개의 페이즈에 할당하는 것, 세 개의 페이즈의 각각의 가중치에 기초하여 가중치 합을 결정하는 것, 및 가중치 합에 기초하여 새로운 프라이머리 노드를 결정하는 것을 포함한다. 라운드 로빈 방법 대신, 가중치 합에 기초하는 에포크 체인지 프로세스는, 장애가 없는 새로운 프라이머리 노드를 적시의 방식으로 선택하는 것을 용이하게 할 수 있다. 라운드 로빈 방법은, 새로운 프라이머리 노드에 대한 라인 내의 다수의 네트워크 노드가 장애가 있는 경우, 프라이머리 노드의 빈번한 체인지를 야기할 수도 있다. 이것은, 장애가 없는 프라이머리 노드를 찾는 데 있어서 레이턴시 또는 지연을 도입하는 것에 의해 블록체인 서비스에 상당한 영향을 준다. 라운드 로빈 방법과는 달리, 본 명세서에서의 에포크 체인지 프로세스는, 새로운 프라이머리 노드를 선택하기 위해 가중치 합에 의존하는데, 이것은 장애가 없는 새로운 프라이머리 노드를 찾는 데 있어서 시간을 감소시킬 수 있다. 이것은, 블록체인 서비스를 제공함에 있어서 전체 블록체인 시스템의 효율성을 더 향상시킬 수 있다.

[0139] 블록체인 네트워크의 동작 동안, 몇몇 네트워크 노드의 실행 속도는, 네트워크 지터링(jittering), 갑작스러운 전력 장애, 디스크 장애, 및 등등에 기인하여, 대부분의 네트워크 노드의 실행 속도보다 뒤처질 수도 있다. 이 시나리오에서, 시스템 내의 네트워크 노드 중 1/3보다 많은 것이 실패할 수도 있다. BFT는, 시스템의 수명 시간 동안 네트워크 노드의 1/3 미만이 고장나는 경우, 안전성 및 생존성을 제공한다. 그러나, 상기에서 설명되는 바와 같은 시나리오에서 상한이 초과될 가능성이 있기 때문에, 이들 보증은 긴 수명의 시스템의 경우에는 불충분하다. 따라서, 장애가 있는 네트워크 노드가 다시 올바르게 거동하게 만들고 후속하는 합의 프로세스에 계속 참가하여 시스템이 자신의 수명에 걸쳐 f 개보다 더 많은 장애를 견디는 것을 허용하는 복구 프로세스가 바람직하다. 또한, 설명된 복구 프로세스는, 합의 프로세스(예를 들면, 합의 프로세스(300 또는 400))를 여전히 수행하고 있으며, 모든 네트워크 노드 사이에서 합의가 달성될 때까지 기다릴 필요가 없는 하나 이상의 네트워크 노드를 복구할 수 있다. 그러한 만큼, 설명된 복구 프로세스는, 시스템 레이턴시를 추가로 감소시킬 수 있고 블록체인 네트워크의 효율성을 향상시킬 수 있다.

[0140] 도 10은, 본 명세서의 구현예에 따라 실행될 수 있는 분산 시스템(예를 들면, 블록체인 네트워크(102 및 212))의 네트워크 노드(예를 들면, 노드(214 또는 404))의 복구 프로세스를 수행하기 위한 프로세스(1000)의 예를 묘사한다. 구체적으로, 도 10은 본 명세서에 따른, 네트워크 노드의 복구 프로세스를 수행하는 방법(1000)의 예시적인 실시형태를 제시하는 도면을 예시한다. 도 10에서 예시되는 바와 같이, 프로세스(1000)는 몇 개의 페이즈 및 스테이지를 포함한다.

[0141] 제1 페이즈(1010)에서, 타겟 시퀀스 번호(R_0)를 갖는 타겟 트랜잭션을 복구하기를 원하는 네트워크 노드(예를 들면, 네트워크 노드(R_0))는, 네트워크 노드가 복구될 것이다는 것을 나타내는 상태 요청 메시지(예를 들면, QUERY_STATE 메시지)를 다른 네트워크 노드에 멀티캐스트한다. 상태 요청 메시지는, 네트워크 노드(R_0)가 복구하고자 하는 타겟 시퀀스 번호를 포함할 수 있다. 제2 페이즈(1020)에서, 다른 네트워크 노드는 상태 요청 메시지를 수신하고 상태 응답 메시지(예를 들면, REPLY_STATE 메시지)를 네트워크 노드(R_0)로 전송한다. 제3 페이즈(1030)에서, 네트워크 노드(R_0)는 다른 네트워크 노드의 각각에게 ECHO 메시지를 요청하는 요청 메시지(예를 들면, FETCH_ECHO 메시지)를 다른 네트워크 노드로 전송한다. ECHO 메시지는 도 3 내지 도 6과 관련하여 상기에서 설명되는 바와 같이 합의 프로세스(300)의 제2 페이즈(320)에서 각각의 다른 네트워크 노드에 의해 전송되는 동일한 ECHO 메시지일 수 있다. 제4 페이즈(1040)에서, 다른 네트워크 노드의 각각은 FETCH_ECHO 메시지에 응답하여 ECHO 메시지를 네트워크 노드(R_0)로 전송한다. 제5 페이즈(1050)에서, 네트워크 노드(R_0)는, 예를 들면, 도 4와 관련하여 상기에서 설명되는 바와 같은 예시적인 재구성 기술에 따라, EC 코드에 따라 ECHO 메시지를 검증하고 타겟 트랜잭션을 복구한다. 제6 페이즈(1060)에서, 네트워크 노드(R_0)는 네트워크 노드가 복구되었다는 것을 나타내는 ACCEPT 메시지를 다른 네트워크 노드로 전송한다.

[0142] 프로세스(1000)는 단지 예시적인 목적을 위해 네 개의 네트워크 노드와 연계하여 구현되는 것으로 예시된다는 것을 유의한다. 프로세스(1000)는 임의의 적절한 수의 네트워크 노드와 연계하여 구현될 수도 있다. 프로세스(1000), QUERY_STATE 메시지의 포맷, 및 REPLY_STATE 메시지의 포맷이, 도 11 및 도 12와 관련하여 하기에서 더욱 상세하게 논의될 것이다.

[0143] 도 11은, 본 명세서의 구현예에 따라 실행될 수 있는, 분산 시스템(예를 들면, 블록체인 네트워크(102 또는 212))에서 네트워크 노드의 복구 프로세스를 수행하기 위한 프로세스(1100)의 예를 묘사한다. 몇몇 구현예에서, 프로세스(1100)는 하나 이상의 컴퓨팅 디바이스를 사용하여 실행되는 하나 이상의 컴퓨터 실행 가능 프로그램을 사용하여 수행될 수도 있다. 제시의 명확성을 위해, 이하의 설명은, 이 설명의 다른 도면의 맥락에서 방법

(1100)을 일반적으로 설명한다. 방법(1100)은, 예를 들면, 임의의 적절한 시스템, 환경, 소프트웨어, 및 하드웨어에 의해, 또는, 적절히, 시스템, 환경, 소프트웨어, 및 하드웨어의 조합에 의해 수행될 수도 있다는 것이 이해될 것이다. 몇몇 구현예에서, 방법(1100)의 다양한 단계는 병렬로, 조합하여, 루프에서, 또는 임의의 순서로 실행될 수 있다.

[0144] 프로세스(1100)는 1106에서 시작하는데, 여기서, 네트워크 노드(1102)는 상태 요청 메시지를 다른 네트워크 노드(1104)로 멀티캐스트한다. 상태 요청 메시지는 네트워크 노드(1102)가 어떤 타겟 시퀀스 번호를 갖는 어떤 타겟 트랜잭션을 복구할 것이다는 표시를 포함한다. 네트워크 노드(1102)는 프라이머리 노드 또는 백업 노드일 수도 있다.

[0145] 본 명세서의 몇몇 구현예에서, 도 12를 참조하면, 상태 요청 메시지의 한 예인 QUERY_STATE 메시지는 <j, seq>의 포맷을 갖는데, 여기서 "j"는 QUERY_STATE 메시지를 전송하는 네트워크 노드(1102)를 나타내고, "seq"는 현재의 합의 프로세스에서 네트워크 노드(1102)에 대한 최대 시퀀스 번호 또는 가장 최근의 시퀀스 번호를 나타낸다. 몇몇 구현예에서, QUERY_STATE 메시지는, 예를 들면, 추가적인 또는 상이한 필드를 포함하는 것에 의해, 상이한 포맷을 가질 수 있다.

[0146] QUERY_STATE 메시지를 다른 네트워크 노드(1104)로 브로드캐스트하는 것에 의해, 네트워크 노드(1102)는, 그들의 가장 최근의 시퀀스 번호를 네트워크 노드(1102)로 전송할 것을 다른 네트워크 노드(1104)에게 요청하고 있고, 그에 의해, 블록체인 시스템의 최신 블록 정보를 획득하고 있다. 그리고 전체 블록체인 시스템의 최신 블록 정보를 획득하는 것에 의해, 네트워크 노드(1102)는 전체 시스템의 최신 상태와 동기화할 수 있을 수도 있고, 그에 의해 그 자신을 복구하고 합의 프로세스에 계속 참가할 수도 있다.

[0147] 도 11을 다시 참조하면, 1108에서, 다른 네트워크 노드(1104)의 각각은, 상태 요청 메시지를 수신하는 것에 응답하여, 상태 응답 메시지(예를 들면, REPLY_STATE 메시지)를 네트워크 노드(1102)로 전송한다. 몇몇 구현예에서, 상태 응답 메시지는 네트워크 노드(1104)와 연관되는 이전 시퀀스 번호를 포함한다.

[0148] 몇몇 구현예에서, 도 12를 참조하면, REPLY_STATE 메시지는, 상태 응답 메시지의 한 예로서, <j, last_seq>의 포맷을 갖는데, 여기서 "j"는 REPLY_STATE 메시지를 전송하는 네트워크 노드(1104)를 나타내고, "last_seq"는 현재의 합의 프로세스에서의 네트워크 노드(1104)에 대한 이전 시퀀스 번호를 나타낸다. 몇몇 구현예에서, REPLY_STATE 메시지는, 예를 들면, 추가적인 또는 상이한 필드를 포함하는 것에 의해, 상이한 포맷을 가질 수 있다.

[0149] 도 11을 다시 참조하면, 1110에서, 네트워크 노드(1102)는 수신된 상태 응답 메시지의 수가 미리 결정된 임계치를 초과하는지의 여부를 결정한다. 예를 들면, 네트워크 노드(1102)는 수신된 REPLY_STATE 메시지의 수가 쿼럼 개수(예를 들면, 2f + 1 또는 n-f)를 초과하는지의 여부를 결정할 수도 있다. 몇몇 구현예에서, 네트워크 노드(1102)는 또한, 쿼럼 개수의 수신된 REPLY_STATE 메시지가 동일한 시퀀스 번호를 포함하는지의 여부를 결정한다. 쿼럼 개수의 수신된 REPLY_STATE 메시지가 동일한 시퀀스 번호를 포함한다는 것은, 네트워크 노드(1104)의 다수가 전체 시스템의 공통 상태에 대해 동의한다는 것을 의미한다.

[0150] 1112에서, 네트워크 노드(1104)로부터 수신되는 동일한 시퀀스 번호를 포함하는 상태 응답 메시지의 수가 미리 결정된 임계치를 초과한다는 것을 네트워크 노드(1102)가 결정하는 경우 네트워크 노드(1102)는 동일한 시퀀스 번호에 기초하여 타겟 시퀀스 번호를 결정한다. 예를 들면, 네트워크 노드(1102)는 타겟 시퀀스 번호를 동일한 시퀀스 번호(예를 들면, "last_seq")의 증분(예를 들면, "last_seq + 1")인 것으로 결정할 수도 있다.

[0151] 1114에서, 네트워크 노드(1102)는 요청 메시지(예를 들면, FETCH_ECHO 메시지)를 다른 네트워크 노드(1104)로 전송한다. FETCH_ECHO 메시지는 네트워크 노드(1102)에 의해 전송되어 다른 네트워크 노드(1104)의 각각에게 ECHO 메시지를 요청한다. 도 3 내지 도 6과 관련하여 상기에서 논의되는 바와 같이, ECHO 메시지는 타겟 트랜잭션에 대한 네트워크 노드(1104) 사이의 합의를 달성하기 위해 네트워크 노드(1104)에 의해 송신되는 메시지이다. ECHO 메시지는 타겟 트랜잭션의 일부(예를 들면, EC 블록) 및 ECHO 메시지를 전송하는 네트워크 노드(1104)의 서명을 포함한다.

[0152] 몇몇 구현예에서, 도 12를 참조하면, FETCH_ECHO 메시지는, 요청 메시지의 한 예로서, <j, last_seq+1>의 포맷을 갖는데, "j"는 FETCH_ECHO 메시지를 전송하는 네트워크 노드(1102)를 나타내고, "last_seq+1"은, 네트워크 노드(1102)가 다른 네트워크 노드(1104)에게 요청하고 있는 ECHO 메시지와 연관되는 타겟 시퀀스 번호를 나타낸다. 몇몇 구현예에서, FETCH_ECHO 메시지는, 예를 들면, 추가적인 또는 상이한 필드를 포함하는 것에 의해, 상이한 포맷을 가질 수 있다.

[0153] 본원에서 논의되는 바와 같은 FETCH_ECHO 메시지는, 가장 최근 시퀀스 번호 또는 가장 큰 시퀀스 번호를 포함하는 ECHO 메시지를 다른 네트워크 노드(1104)에게 요청하기 위해 네트워크 노드(1102)에 의해 전송된다. 가장 최근의 시퀀스 번호 또는 가장 큰 시퀀스 번호를 포함하는 ECHO 메시지를 다른 네트워크 노드(1104)로부터 수집하는 것에 의해, 네트워크 노드(1102)는 가장 최근의 시퀀스 번호와 연관되는 가장 최근의 상태로 복구할 수 있을 수도 있다.

[0154] 도 11을 다시 참조하면, 1116에서, 네트워크 노드(1104)의 각각은, FETCH_ECHO 메시지를 수신하는 것에 응답하여, ECHO 메시지를 네트워크 노드(1102)로 전송한다. 몇몇 구현예에서, 네트워크 노드(1104)의 각각은, ECHO 메시지를 네트워크 노드(1102)로 전송하기 이전에, FETCH_ECHO 메시지를 검증한다. 몇몇 구현예에서, 네트워크 노드(1104)의 각각은, FETCH_ECHO 메시지에 포함되는 시퀀스 번호가 네트워크 노드(1104)와 연관되는 가장 최근의 시퀀스 번호를 초과하는지의 여부를 결정하는 것에 의해 FETCH_ECHO 메시지를 검증한다. FETCH_ECHO 메시지에 포함되는 시퀀스 번호가 네트워크 노드(1104)와 연관되는 가장 최근의 시퀀스 번호와 동일하면, 네트워크 노드(1104)는, FETCH_ECHO 메시지가 유효하고 ECHO 메시지가 네트워크 노드(1102)로 전송될 것이다고 결정한다. FETCH_ECHO 메시지에 포함되는 시퀀스 번호가 네트워크 노드(1104)와 연관되는 가장 최근의 시퀀스 번호를 초과하면, 네트워크 노드(1104),는 FETCH_ECHO 메시지가 무효하다는 것 및 ECHO 메시지가 네트워크 노드(1102)로 전송되지 않을 것이다고 결정한다.

[0155] 1118에서, 네트워크 노드(1102)는 네트워크 노드(1104)에 의해 전송되는 ECHO 메시지가 유효한지의 여부를 검증한다. 몇몇 구현예에서, 네트워크 노드(1102)는 머클 트리를 사용하여 ECHO 메시지를 검증한다. 예를 들면, 네트워크 노드(1102)는 ECHO 메시지에 포함되는 데이터를 사용하여 머클 트리를 재구성하고 재구성된 메르켈 트리의 재구성된 루트 해시 값을 결정할 수도 있다. 그 다음, 네트워크 노드(1102)는 재구성된 루트 해시 값을 ECHO 메시지에 포함되는 루트 해시 값에 비교할 수도 있다. 재구성된 루트 해시 값이 ECHO 메시지에 포함되는 루트 해시 값과 매치하면, 네트워크 노드(1102)는 ECHO 메시지가 유효하다고 결정한다. 재구성된 루트 해시 값이 ECHO 메시지에 포함되는 루트 해시 값과 매치하지 않으면, 네트워크 노드(1102)는 ECHO 메시지가 무효하다고 결정하고 무효한 ECHO 메시지를 폐기할 수도 있다.

[0156] 몇몇 구현예에서, 네트워크 노드(1102)는, ECHO 메시지 내의 서명이 유효한지의 여부를 추가로 검증하는 것에 의해 ECHO 메시지가 유효한지의 여부를 검증한다. 예를 들면, 네트워크 노드(1102)는 서명을 검증하기 위해 프라이빗 키와 쌍을 이루는 퍼블릭 키를 사용하여 ECHO 메시지 내의 프라이빗 키 서명을 인증할 수도 있다.

[0157] 1120에서, 네트워크 노드(1102)는 다른 네트워크 노드(1104)로부터 수신되는 유효한 ECHO 메시지의 수가 미리 결정된 임계치를 초과하는지의 여부를 결정한다. 예를 들면, 네트워크 노드(1102)는, 다른 네트워크 노드(1104)로부터 수신되는 유효한 ECHO 메시지의 수가 쿼럼 개수(예를 들면, 2f + 1)를 초과하는지의 여부를 결정할 수도 있다.

[0158] 1122에서, 네트워크 노드(1102)는, 유효한 ECHO 메시지의 수가 미리 결정된 임계치를 초과한다고 결정하는 것에 응답하여 타겟 시퀀스 번호를 갖는 타겟 트랜잭션을 복구한다. 몇몇 구현예에서, 네트워크 노드(1102)는 다수의 유효한 ECHO 메시지에 포함되는 데이터를 사용하여 타겟 트랜잭션을 복구한다. 예를 들면, 네트워크 노드(1102)는 EC 코드에 따라 타겟 트랜잭션을 재구성하기 위해 ECHO 메시지에 포함되는 EC 블록의 서브세트를 검색할 수도 있다.

[0159] 1124에서, 네트워크 노드(1102)는 타겟 트랜잭션을 복구한 이후 ACCEPT 메시지를 다른 네트워크 노드(1104)로 멀티캐스트한다. 예를 들면, 네트워크 노드(1102)는, 타겟 트랜잭션을 성공적으로 재구성한 이후, ACCEPT 메시지를 다른 네트워크 노드(1104)로 멀티캐스트할 수도 있다. 몇몇 구현예에서, ACCEPT 메시지는 ECHO 메시지 내의 서명의 세트 및 타겟 시퀀스 번호를 포함한다. 서명 및 타겟 시퀀스 번호를 포함하는 ACCEPT 메시지를 다른 네트워크 노드(1104)로 전송하는 것에 의해, 네트워크 노드(1102)는, 네트워크 노드(1102)가 복구되어 시스템의 최신 상태에 동기화되었다는 것을 다른 네트워크 노드(1104)에게 나타낸다.

[0160] 본 명세서의 상기에서 논의되는 바와 같은 복구 프로세스는, 복구 프로세스를 구현하고 네트워크 병목 현상을 완화시키는데 도움을 주는 컴퓨터의 동작을 향상시키는 많은 피처를 포함한다. 예를 들면, 본 명세서에서의 복구 프로세스는, 새로운 프라이머리 노드가 되기 위해 신청하는 네트워크 노드에 의해 상태 요청 메시지를 전송하는 것, 다른 네트워크 노드로부터 상태 응답 메시지를 수신하는 것, 및 다른 네트워크 노드에게 FETCH_ECHO 메시지를 요청하기 위해 네트워크 노드에 의해 FETCH_ECHO 메시지를 전송하는 것을 포함하는 동작을 포함한다. 이들 동작은, 장애가 있는 네트워크 노드(faulty network node)의 복구 프로세스가, 다른 장애가 없는 네트워크 노드(other non-faulty network node) 사이의 합의 프로세스의 정상 동작과 간섭하지 않도록 수행된다. 이것은

복구 프로세스의 복잡도를 감소시키는 것에 의해 장애가 있는 네트워크 노드를 복구하기 위한 컴퓨팅 및 네트워크 리소스의 절약을 용이하게 한다.

[0161] 도 13을 참조하면, 도 13은, 본 명세서의 구현예에 따른, 합의 장치(1300)의 모듈을 예시하는 도면이다. 합의를 달성하기 위한 장치(1300)는 블록체인 기술에 기초하여 합의 시스템에 적용될 수 있다. 예를 들면, 장치(1300)는 도 1 내지 도 6에서 도시되는 구현예에 대응할 수 있다. 장치(1300)는 블록체인 네트워크에서의 프라이머리 노드에서 구현될 수 있다. 장치(1300)는 다음의 것을 포함한다: 트랜잭션 요청을 수신하도록 구성되는 수신기 또는 수신 유닛(1302); 트랜잭션 요청을 사용하여 다수의 소거 코드(EC) 블록을 EC 코드에 따라 생성하도록 구성되는 생성 유닛(1304); 다수의 제1 메시지- 다수의 제1 메시지의 각각은 다수의 EC 블록과 연관되는 합성 해시 값을 포함함 - 를 하나 이상의 백업 노드로 각각 전송하도록 구성되는 송신기 또는 송신 유닛(1306); 백업 노드 중 적어도 하나로부터 적어도 하나의 제2 메시지 - 적어도 하나의 제2 메시지는 다수의 제1 메시지 중 하나 및 다수의 제1 메시지 중 하나와 연관되는 백업 노드 중 적어도 하나의 서명을 포함함 - 를 수신하도록 또한 구성되는 수신기 또는 수신 유닛(1302); 백업 노드 중 적어도 하나로부터 적어도 하나의 제2 메시지를 수신하는 것에 응답하여 적어도 하나의 제2 메시지가 유효한지의 여부를 검증하도록 구성되는 검증 유닛(1308); 유효한 제2 메시지의 수가 미리 결정된 임계치를 초과하는지의 여부를 결정하도록 구성되는 결정 유닛(1310); 유효한 제2 메시지의 수가 미리 결정된 임계치를 초과한다고 결정하는 것에 응답하여 EC 코드에 따라 유효한 제2 메시지의 수의 서브세트에 기초하여 트랜잭션 요청을 재구성하도록 구성되는 재구성 유닛(1312); 트랜잭션 요청이 성공적으로 재구성되었다고 결정하는 것에 응답하여 제3 메시지 - 제3 메시지는 유효한 제2 메시지 내에 있는 서명의 세트를 포함함 - 를 다른 네트워크 노드로 전송하도록 또한 구성되는 송신기 또는 송신 유닛(1306); 백업 노드 중 적어도 하나로부터 적어도 하나의 제3 메시지를 수신하도록 또한 구성되는 수신기 또는 수신 유닛(1302); 및 미리 결정된 수의 동일한 제3 메시지를 수신하는 것에 응답하여 트랜잭션 요청을 실행하도록 구성되는 실행 유닛(1314).

[0162] 옵션 사항의 구현예(optional implementation)에서, 트랜잭션 요청은 시퀀스 번호와 연관된다.

[0163] 옵션 사항의 구현예에서, EC 코드에 따라 복수의 EC 블록을 생성하는 것은 다음의 것을 포함한다: EC 코드를 사용하여 트랜잭션 요청을 EC 메시지로 변환하는 것 및 EC 메시지를 다수의 EC 블록으로 분할하는 것.

[0164] 옵션 사항의 구현예에서, 다수의 EC 블록의 합성 해시 값은 해시 트리를 사용하여 생성된다.

[0165] 옵션 사항의 구현예에서, 해시 트리는 머클 트리를 포함하고, 합성 해시 값은 머클 트리의 루트 해시 값이다.

[0166] 옵션 사항의 구현예에서, 다수의 제1 메시지 중 하나와 연관되는 백업 노드 중 적어도 하나의 서명은, 다수의 제1 메시지 중 하나와 연관되는 백업 노드 중 적어도 하나의 프라이빗 키 서명을 포함한다.

[0167] 옵션 사항의 구현예에서, 적어도 하나의 제2 메시지는 다수의 EC 블록 중 적어도 하나를 더 포함한다.

[0168] 옵션 사항의 구현예에서, 적어도 하나의 제2 메시지가 유효한지의 여부를 검증하는 것은 다음의 것을 포함한다: 적어도 하나의 제2 메시지 내의 다수의 EC 블록 중 적어도 하나를 사용하여 재구성된 해시 트리를 생성하는 것; 재구성된 해시 트리의 재구성된 합성 해시 값을 결정하는 것; 및 재구성된 합성 해시 값이 적어도 하나의 제2 메시지 내의 합성 해시 값과 매치하는지의 여부를 결정하는 것.

[0169] 옵션 사항의 구현예에서, 결정 유닛(1310)은 또한, 재구성된 합성 해시 값이 제2 메시지 내의 합성 해시 값과 매치한다고 결정하는 것에 응답하여 적어도 하나의 제2 메시지가 유효하다고 결정하도록 구성된다.

[0170] 옵션 사항의 구현예에서, 미리 결정된 수의 동일한 제3 메시지는 서명의 동일한 세트를 갖는 미리 결정된 수의 제3 메시지를 포함한다.

[0171] 도 13은 합의 장치(1300)의 내부 기능 모듈 및 구조를 예시하는 개략도이다. 본질적으로 실행 본체는, 전자 디바이스일 수 있으며, 전자 디바이스는 다음의 것을 포함한다: 적어도 하나의 프로세서; 및 적어도 하나의 프로세서의 실행 가능 명령어들을 저장하도록 구성되는 메모리.

[0172] 적어도 하나의 프로세서는, 트랜잭션 요청을 수신하도록; 다수의 소거 코드(EC) 블록을, 트랜잭션 요청을 사용하여 EC 코드에 따라 생성하도록; 다수의 제1 메시지 - 다수의 제1 메시지의 각각은 다수의 EC 블록과 연관되는 합성 해시 값을 포함함 - 를 하나 이상의 백업 노드로 전송하도록; 백업 노드 중 적어도 하나로부터 적어도 하나의 제2 메시지 - 적어도 하나의 제2 메시지는 다수의 제1 메시지 중 하나 및 다수의 제1 메시지 중 하나와 연관되는 백업 노드 중 적어도 하나의 서명을 포함함 - 를 수신하도록; 백업 노드 중 적어도 하나로부터 적어도 하나의 제2 메시지를 수신하는 것에 응답하여 적어도 하나의 제2 메시지가 유효한지의 여부를 검증하도록; 유효

한 제2 메시지의 수가 미리 결정된 임계치를 초과하는지의 여부를 결정하도록; 유효한 제2 메시지의 수가 미리 결정된 임계치를 초과한다고 결정하는 것에 응답하여, EC 코드에 따라 유효한 제2 메시지의 수의 서브세트에 기초하여 트랜잭션 요청을 재구성하도록; 트랜잭션 요청이 성공적으로 재구성되었다고 결정하는 것에 응답하여 제3 메시지 - 제3 메시지는 유효한 제2 메시지 내에 있는 서명의 세트를 포함함 - 를 다른 네트워크 노드로 전송하도록; 백업 노드 중 적어도 하나로부터 적어도 하나의 제3 메시지를 수신하도록; 그리고 미리 결정된 수의 동일한 제3 메시지를 수신하는 것에 응답하여 트랜잭션 요청을 실행하도록 구성된다.

[0173] 옵션 사항으로, 트랜잭션 요청은 시퀀스 번호와 연관된다.

[0174] 옵션 사항으로, EC 코드에 따라 복수의 EC 블록을 생성하는 것은 다음의 것을 포함한다: EC 코드를 사용하여 트랜잭션 요청을 EC 메시지로 변환하는 것 및 EC 메시지를 다수의 EC 블록으로 분할하는 것.

[0175] 옵션 사항으로, 다수의 EC 블록의 합성 해시 값은 해시 트리를 사용하여 생성된다.

[0176] 옵션 사항으로, 해시 트리는 머클 트리를 포함하고, 합성 해시 값은 머클 트리의 루트 해시 값이다.

[0177] 옵션 사항으로, 다수의 제1 메시지 중 하나와 연관되는 백업 노드 중 적어도 하나의 서명은, 다수의 제1 메시지 중 하나와 연관되는 백업 노드 중 적어도 하나의 프라이빗 키 서명을 포함한다.

[0178] 옵션 사항으로, 적어도 하나의 제2 메시지는 다수의 EC 블록 중 적어도 하나를 더 포함한다.

[0179] 옵션 사항으로, 적어도 하나의 제2 메시지가 유효한지의 여부를 검증하는 것은 다음의 것을 포함한다: 적어도 하나의 제2 메시지 내의 다수의 EC 블록 중 적어도 하나를 사용하여 재구성된 해시 트리를 생성하는 것; 재구성된 해시 트리의 재구성된 합성 해시 값을 결정하는 것; 및 재구성된 합성 해시 값이 적어도 하나의 제2 메시지 내의 합성 해시 값과 매치하는지의 여부를 결정하는 것.

[0180] 옵션 사항으로, 적어도 하나의 프로세서는 또한, 재구성된 합성 해시 값이 제2 메시지 내의 합성 해시 값과 매치한다고 결정하는 것에 응답하여 적어도 하나의 제2 메시지가 유효하다고 결정하도록 구성된다.

[0181] 옵션 사항으로, 미리 결정된 수의 동일한 제3 메시지는 서명의 동일한 세트를 갖는 미리 결정된 수의 제3 메시지를 포함한다.

[0182] 도 14를 참조하면, 도 14는 본 명세서의 구현예에 따른, 합의 장치(1400)의 모듈을 예시하는 도면이다. 합의를 달성하기 위한 장치(1400)는 블록체인 기술에 기초하여 합의 시스템에 적용될 수 있다. 예를 들면, 장치(1400)는 도 1 내지 도 6에서 도시되는 구현예에 대응할 수 있다. 예를 들면, 장치(1400)는 블록체인 네트워크의 프라이머리 노드에서 구현될 수 있다. 장치(1400)는 다음의 것을 포함한다: 프라이머리 노드로부터 제1 메시지 - 제1 메시지는 다수의 EC 블록과 연관되는 합성 해시 값을 포함하고, 다수의 EC 블록은 트랜잭션 요청을 사용하여 EC 코드에 따라 프라이머리 노드에 의해 생성됨 - 를 수신하도록 구성되는 수신기 또는 수신 유닛(1402); 백업 노드에 의해, 제1 메시지를 수신하는 것에 응답하여 제2 메시지 - 제2 메시지는 상기 제1 메시지 및 제1 메시지와 연관되는 백업 노드의 서명을 포함함 - 를 다른 네트워크 노드로 전송하도록 구성되는 송신기 또는 송신 유닛(1404); 백업 노드 이외의 적어도 하나의 백업 노드로부터 적어도 하나의 제2 메시지를 수신하도록 또한 구성되는 수신기 또는 수신 유닛(1402); 적어도 하나의 백업 노드로부터 적어도 하나의 제2 메시지를 수신하는 것에 응답하여 적어도 하나의 제2 메시지가 유효한지의 여부를 검증하도록 구성되는 검증 유닛(1406); 유효한 제2 메시지의 수가 미리 결정된 임계치를 초과하는지의 여부를 결정하도록 구성되는 결정 유닛(1408); 유효한 제2 메시지의 수가 미리 결정된 임계치를 초과한다고 결정하는 것에 응답하여 EC 코드에 따라 유효한 제2 메시지의 수의 서브세트에 기초하여 트랜잭션 요청을 재구성하도록 구성되는 재구성 유닛(1410); 트랜잭션 요청이 성공적으로 재구성되었다고 결정하는 것에 응답하여 제3 메시지 - 제3 메시지는 유효한 제2 메시지 내에 있는 서명의 세트를 포함함 - 를 다른 네트워크 노드로 전송하도록 구성되는 송신기 또는 송신 유닛(1404); 백업 노드 중 적어도 하나로부터 적어도 하나의 제3 메시지를 수신하도록 또한 구성되는 수신기 또는 수신 유닛(1402); 및 미리 결정된 수의 동일한 제3 메시지를 수신하는 것에 응답하여 트랜잭션 요청을 실행하도록 구성되는 실행 유닛(1412).

[0183] 옵션 사항의 구현예에서, EC 코드에 따라 복수의 EC 블록을 생성하는 것은 다음의 것을 포함한다: EC 코드를 사용하여 트랜잭션 요청을 EC 메시지로 변환하는 것; 및 EC 메시지를 다수의 EC 블록으로 분할하는 것.

[0184] 옵션 사항의 구현예에서, 복수의 EC 블록의 합성 해시 값은 해시 트리를 사용하여 생성된다.

[0185] 옵션 사항의 구현예에서, 해시 트리는 머클 트리를 포함하고, 합성 해시 값은 머클 트리의 루트 해시 값이다.

[0186] 옵션 사항의 구현예에서, 제1 메시지와 연관되는 백업 노드의 서명은 제1 메시지와 연관되는 백업 노드의 프라이빗 키 서명을 포함한다.

[0187] 옵션 사항의 구현예에서, 적어도 하나의 제2 메시지는 다수의 EC 블록 중 적어도 하나를 더 포함한다.

[0188] 옵션 사항의 구현예에서, 적어도 하나의 제2 메시지가 유효한지의 여부를 검증하는 것은 다음의 것을 포함한다: 적어도 하나의 제2 메시지 내의 다수의 EC 블록 중 적어도 하나를 사용하여 재구성된 해시 트리를 생성하는 것; 재구성된 해시 트리의 재구성된 합성 해시 값을 결정하는 것; 재구성된 합성 해시 값을 적어도 하나의 제2 메시지 내의 합성 해시 값에 비교하는 것; 및 재구성된 합성 해시 값이 적어도 하나의 제2 메시지 내의 합성 해시 값과 매치하는지의 여부를 결정하는 것.

[0189] 옵션 사항의 구현예에서, 결정 유닛(1408)은 또한, 재구성된 합성 해시 값이 제2 메시지 내의 합성 해시 값과 매치한다고 결정하는 것에 응답하여 적어도 하나의 제2 메시지가 유효하다고 결정하도록 구성된다.

[0190] 옵션 사항의 구현예에서, 미리 결정된 수의 동일한 제3 메시지는 서명의 동일한 세트를 갖는 미리 결정된 수의 제3 메시지를 포함한다.

[0191] 도 14는 합의 장치(1400)의 내부 기능 모듈 및 구조를 예시하는 개략도이다. 본질적으로 실행 본체는, 전자 디바이스일 수 있으며, 전자 디바이스는 다음의 것을 포함한다: 적어도 하나의 프로세서; 및 적어도 하나의 프로세서의 실행 가능 명령어들을 저장하도록 구성되는 메모리.

[0192] 적어도 하나의 프로세서는, 프라이머리 노드로부터 제1 메시지 - 제1 메시지는 다수의 EC 블록과 연관되는 합성 해시 값을 포함하고, 다수의 EC 블록은 트랜잭션 요청을 사용하여 EC 코드에 따라 프라이머리 노드에 의해 생성됨 - 를 수신하도록; 백업 노드에 의해, 제1 메시지를 수신하는 것에 응답하여 제2 메시지 - 제2 메시지는 상기 제1 메시지 및 제1 메시지와 연관되는 백업 노드의 서명을 포함함 - 를 다른 네트워크 노드로 전송하도록; 백업 노드 이외의 적어도 하나의 백업 노드로부터 적어도 하나의 제2 메시지를 수신하도록; 백업 노드 중 적어도 하나로부터 적어도 하나의 제2 메시지를 수신하는 것에 응답하여 적어도 하나의 제2 메시지가 유효한지의 여부를 검증하도록; 유효한 제2 메시지의 수가 미리 결정된 임계치를 초과하는지의 여부를 결정하도록; 유효한 제2 메시지의 수가 미리 결정된 임계치를 초과한다고 결정하는 것에 응답하여, EC 코드에 따라 유효한 제2 메시지의 수의 서브세트에 기초하여 트랜잭션 요청을 재구성하도록; 트랜잭션 요청이 성공적으로 재구성되었다고 결정하는 것에 응답하여 제3 메시지 - 제3 메시지는 유효한 제2 메시지 내에 있는 서명의 세트를 포함함 - 를 다른 네트워크 노드로 전송하도록; 백업 노드 중 적어도 하나로부터 적어도 하나의 제3 메시지를 수신하도록; 그리고 미리 결정된 수의 동일한 제3 메시지를 수신하는 것에 응답하여 트랜잭션 요청을 실행하도록 구성된다.

[0193] 옵션 사항으로, EC 코드에 따라 복수의 EC 블록을 생성하는 것은 다음의 것을 포함한다: EC 코드를 사용하여 트랜잭션 요청을 EC 메시지로 변환하는 것; 및 EC 메시지를 다수의 EC 블록으로 분할하는 것.

[0194] 옵션 사항으로, 복수의 EC 블록의 합성 해시 값은 해시 트리를 사용하여 생성된다.

[0195] 옵션 사항으로, 해시 트리는 머클 트리를 포함하고, 합성 해시 값은 머클 트리의 루트 해시 값이다.

[0196] 옵션 사항으로, 제1 메시지와 연관되는 백업 노드의 서명은, 제1 메시지와 연관되는 백업 노드의 프라이빗 키 서명을 포함한다.

[0197] 옵션 사항으로, 적어도 하나의 제2 메시지는 다수의 EC 블록 중 적어도 하나를 더 포함한다.

[0198] 옵션 사항으로, 적어도 하나의 제2 메시지가 유효한지의 여부를 검증하는 것은 다음의 것을 포함한다: 적어도 하나의 제2 메시지 내의 다수의 EC 블록 중 적어도 하나를 사용하여 재구성된 해시 트리를 생성하는 것; 재구성된 해시 트리의 재구성된 합성 해시 값을 결정하는 것; 재구성된 합성 해시 값을 적어도 하나의 제2 메시지 내의 합성 해시 값에 비교하는 것; 및 재구성된 합성 해시 값이 적어도 하나의 제2 메시지 내의 합성 해시 값과 매치하는지의 여부를 결정하는 것.

[0199] 옵션 사항으로, 적어도 하나의 프로세서는 또한, 재구성된 합성 해시 값이 제2 메시지 내의 합성 해시 값과 매치한다고 결정하는 것에 응답하여 적어도 하나의 제2 메시지가 유효하다고 결정하도록 구성된다.

[0200] 옵션 사항으로, 미리 결정된 수의 동일한 제3 메시지는 서명의 동일한 세트를 갖는 미리 결정된 수의 제3 메시지를 포함한다.

[0201] 도 15를 참조하면, 도 15는, 본 명세서의 구현예에 따른, 프라이머리 노드 체인징 장치(1500)의 모듈을 예시하

는 도면이다. 프라이머리 노드를 체인지하기 위한 장치(1500)는 블록체인 기술에 기초한 합의 시스템에 적용될 수 있다. 장치(1500)는 도 7 내지 도 9에서 도시되는 구현예에 대응할 수 있다. 예를 들면, 장치(1500)는 블록체인 네트워크의 백업 노드에서 구현될 수 있다. 장치(1500)는 다음의 것을 포함한다: 에포크 체인지 - 에포크 체인지는, 현재의 프라이머리 노드를 갖는 현재의 에포크로부터 새로운 프라이머리 노드를 갖는 새로운 에포크로의 체인지를 야기하고, 현재의 에포크는 프라이머리 노드를 사용하여 다수의 네트워크 노드 사이에서 합의를 달성하기 위한 합의 프로세스를 포함하고, 합의 프로세스는 세 개의 페이즈를 포함함 - 가 수행될 필요가 있다고 결정하도록 구성되는 결정 유닛(1502); 현재의 에포크에서의 합의 프로세스의 세 개의 페이즈의 각각과 연관되는 백업 노드의 각각의 가중치 - 가중치는 새로운 프라이머리 노드가 될 백업 노드의 자격 부여의 메트릭임 - 를 결정하도록 또한 구성되는 결정 유닛(1502); 현재의 에포크에서의 세 개의 페이즈의 각각과 연관되는 백업 노드의 각각의 가중치에 기초하여 백업 노드에 대한 가중치 합을 결정하도록 또한 구성되는 결정 유닛(1502); 가중치 합이 제1 미리 결정된 임계치에 도달한다고 결정하는 것에 응답하여 EPOCH_CHANGE 메시지 - EPOCH_CHANGE 메시지는, 현재의 프라이머리 노드를 갖는 현재의 에포크로부터, 백업 노드가 새로운 프라이머리 노드인 새로운 에포크로의 체인지에 대한 요청을 나타내고, EPOCH_CHANGE 메시지는 백업 노드의 가중치 합을 포함함 - 를 네트워크 노드 이외의 다수의 네트워크 노드로 전송하도록 구성되는 송신기 또는 송신 유닛(1504); 백업 노드 이외의 다수의 네트워크 노드 중 적어도 하나로부터 적어도 하나의 NEW_EPOCH 메시지 - NEW_EPOCH 메시지는 새로운 프라이머리 노드가 될 백업 노드의 확인 응답을 나타냄 - 를 수신하도록 구성되는 수신기 또는 수신 유닛(1506); 적어도 하나의 NEW_EPOCH 메시지가 유효한지의 여부를 검증하도록 구성되는 검증 유닛(1508); 적어도 하나의 NEW_EPOCH 메시지 중의 유효한 NEW_EPOCH 메시지의 수가 제2 미리 결정된 임계치를 초과하는지의 여부를 결정하도록 또한 구성되는 결정 유닛(1502); 및 유효한 NEW_EPOCH 메시지의 수가 제2 미리 결정된 임계치를 초과한다고 결정하는 것에 응답하여 백업 노드를 새로운 에포크에서 새로운 프라이머리 노드가 되는 것으로 결정하도록 또한 구성되는, 결정 유닛(1502).

[0202] 옵션 사항의 구현예에서, 현재의 에포크에서의 합의 프로세스의 세 개의 페이즈의 각각과 연관되는 백업 노드의 각각의 가중치를 결정하는 것은, 합의 프로세스의 제1 페이즈에 대한 백업 노드의 가중치를 제1 값이 되는 것으로 결정하는 것을 포함한다.

[0203] 옵션 사항의 구현예에서, 현재의 에포크에서의 합의 프로세스의 세 개의 페이즈의 각각과 연관되는 백업 노드의 각각의 가중치를 결정하는 것은, 다음의 것을 포함한다: 현재의 에포크에서의 합의 프로세스의 제2 페이즈에서 쿼럼 검증의 실패를 결정하는 것에 응답하여, 합의 프로세스의 제2 페이즈에 대한 백업 노드의 가중치를 제1 값이 되는 것으로 결정하는 것; 및 현재의 에포크에서의 합의 프로세스의 제2 페이즈에서 쿼럼 검증의 성공을 결정하는 것에 응답하여, 합의 프로세스의 제2 페이즈에 대한 백업 노드의 가중치를 제2 값 - 제1 값은 제2 값보다 더 작음 - 이 되는 것으로 결정하는 것.

[0204] 옵션 사항의 구현예에서, 네트워크 노드에 대한 제2 페이즈에서의 쿼럼 검증은, 다른 네트워크 노드로부터 미리 결정된 수의 ECHO 메시지를 수신하는 것을 포함한다.

[0205] 옵션 사항의 구현예에서, 현재의 에포크에서의 합의 프로세스의 세 개의 페이즈의 각각과 연관되는 백업 노드의 각각의 가중치를 결정하는 것은, 다음의 것을 포함한다: 현재의 에포크에서의 합의 프로세스의 제3 페이즈에서 쿼럼 검증의 실패를 결정하는 것에 응답하여, 합의 프로세스의 제3 페이즈에 대한 백업 노드의 가중치를 제3 값이 되는 것으로 결정하는 것; 및 현재의 에포크에서의 합의 프로세스의 제3 페이즈에서 쿼럼 검증의 성공을 결정하는 것에 응답하여, 합의 프로세스의 제3 페이즈에 대한 백업 노드의 가중치를 제4 값 - 제3 값은 제4 값보다 더 작음 - 이 되는 것으로 결정하는 것.

[0206] 옵션 사항의 구현예에서, 네트워크 노드에 대한 제3 페이즈에서의 쿼럼 검증은, 다른 네트워크 노드로부터 미리 결정된 수의 수락 메시지를 수신하는 것을 포함하는데, 여기서 다른 네트워크 노드로부터의 수락 메시지의 각각은, 다른 네트워크 노드의 각각이 미리 결정된 수의 ECHO 메시지를 수락하였다는 것을 나타낸다.

[0207] 옵션 사항의 구현예에서, EPOCH_CHANGE 메시지는 다수의 네트워크 노드 중의 네트워크 노드의 세트와 연관되는 서명의 세트를 더 포함하고, NEW_EPOCH 메시지는 EPOCH_CHANGE 메시지의 다이제스트를 포함한다.

[0208] 옵션 사항의 구현예에서, 적어도 하나의 유효한 NEW_EPOCH 메시지가 유효한지의 여부를 검증하는 것은, 적어도 하나의 NEW_EPOCH 메시지 내의 EPOCH_CHANGE 메시지의 다이제스트가 유효한지의 여부를 검증하는 것을 포함하고, 적어도 하나의 NEW_EPOCH 메시지 내의 EPOCH_CHANGE 메시지의 다이제스트가 유효한지의 여부를 검증하는 것은, EPOCH_CHANGE 메시지 내의 서명의 세트가 유효한지의 여부를 검증하는 것을 포함한다.

[0209] 옵션 사항의 구현예에서, 에포크 체인지가 수행될 필요가 있다고 결정하는 것은, 미리 결정된 시간 기간 내에 이전의 에포크에서 합의가 달성되지 않았다고 결정하는 것에 응답하여 에포크 체인지가 수행될 필요가 있다고 결정하는 것을 포함한다.

[0210] 옵션 사항의 구현예에서, 프라이머리 노드 체인징 장치(1500)는 다음의 것을 더 포함한다: 새로운 프라이머리 노드를 갖는 새로운 에포크 - 새로운 에포크는 새로운 프라이머리 노드를 사용하여 복수의 네트워크 노드 사이에서 합의를 달성하기 위한 합의 프로세스를 포함함 - 에서 동작하도록 구성되는 동작 유닛(1510).

[0211] 도 15는 프라이머리 노드 체인징 장치(1500)의 내부 기능 모듈 및 구조를 예시하는 개략도이다. 본질적으로 실행 본체는, 전자 디바이스일 수 있으며, 전자 디바이스는 다음의 것을 포함한다: 적어도 하나의 프로세서; 및 적어도 하나의 프로세서의 실행 가능 명령어들을 저장하도록 구성되는 메모리.

[0212] 적어도 하나의 프로세서는, 에포크 체인지 - 에포크 체인지는, 현재의 프라이머리 노드를 갖는 현재의 에포크로부터 새로운 프라이머리 노드를 갖는 새로운 에포크로의 체인지를 야기하고, 현재의 에포크는 프라이머리 노드를 사용하여 다수의 네트워크 노드 사이에서 합의를 달성하기 위한 합의 프로세스를 포함하고, 합의 프로세스는 세 개의 페이즈를 포함함 - 가 수행될 필요가 있다고 결정하도록; 현재의 에포크에서의 합의 프로세스의 세 개의 페이즈의 각각과 연관되는 백업 노드의 각각의 가중치 - 가중치는 새로운 프라이머리 노드가 될 백업 노드의 자격 부여의 메트릭임 - 를 결정하도록; 현재의 에포크에서의 세 개의 페이즈의 각각과 연관되는 백업 노드의 각각의 가중치에 기초하여 백업 노드에 대한 가중치 합을 결정하도록; 가중치 합이 제1 미리 결정된 임계치에 도달한다고 결정하는 것에 응답하여 EPOCH_CHANGE 메시지 - EPOCH_CHANGE 메시지는, 현재의 프라이머리 노드를 갖는 현재의 에포크로부터, 백업 노드가 새로운 프라이머리 노드인 새로운 에포크로의 체인지에 대한 요청을 나타내고, EPOCH_CHANGE 메시지는 백업 노드의 가중치 합을 포함함 - 를 네트워크 노드 이외의 다수의 네트워크 노드로 전송하도록; 백업 노드 이외의 다수의 네트워크 노드 중 적어도 하나로부터 적어도 하나의 NEW_EPOCH 메시지 - NEW_EPOCH 메시지는 새로운 프라이머리 노드가 될 백업 노드의 확인 응답을 나타냄 - 를 수신하도록; 적어도 하나의 NEW_EPOCH 메시지가 유효한지의 여부를 검증하도록; 적어도 하나의 NEW_EPOCH 메시지 중의 유효한 NEW_EPOCH 메시지의 수가 제2 미리 결정된 임계치를 초과하는지의 여부를 결정하도록; 유효한 NEW_EPOCH 메시지의 수가 제2 미리 결정된 임계치를 초과한다고 결정하는 것에 응답하여 백업 노드를 새로운 에포크에서 새로운 프라이머리 노드가 되는 것으로 결정하도록 구성된다.

[0213] 옵션 사항으로, 현재의 에포크에서의 합의 프로세스의 세 개의 페이즈의 각각과 연관되는 백업 노드의 각각의 가중치를 결정하는 것은, 합의 프로세스의 제1 페이즈에 대한 백업 노드의 가중치를 제1 값이 되는 것으로 결정하는 것을 포함한다.

[0214] 옵션 사항으로, 현재의 에포크에서의 합의 프로세스의 세 개의 페이즈의 각각과 연관되는 백업 노드의 각각의 가중치를 결정하는 것은, 다음의 것을 포함한다: 현재의 에포크에서의 합의 프로세스의 제2 페이즈에서 쿼럼 검증의 실패를 결정하는 것에 응답하여, 합의 프로세스의 제2 페이즈에 대한 백업 노드의 가중치를 제1 값이 되는 것으로 결정하는 것; 및 현재의 에포크에서의 합의 프로세스의 제2 페이즈에서 쿼럼 검증의 성공을 결정하는 것에 응답하여, 합의 프로세스의 제2 페이즈에 대한 백업 노드의 가중치를 제2 값 - 제1 값은 제2 값보다 더 작음 - 이 되는 것으로 결정하는 것.

[0215] 옵션 사항으로, 네트워크 노드에 대한 제2 페이즈에서의 쿼럼 검증은, 다른 네트워크 노드로부터 미리 결정된 수의 ECHO 메시지를 수신하는 것을 포함한다.

[0216] 옵션 사항으로, 현재의 에포크에서의 합의 프로세스의 세 개의 페이즈의 각각과 연관되는 백업 노드의 각각의 가중치를 결정하는 것은, 다음의 것을 포함한다: 현재의 에포크에서의 합의 프로세스의 제3 페이즈에서 쿼럼 검증의 실패를 결정하는 것에 응답하여, 합의 프로세스의 제3 페이즈에 대한 백업 노드의 가중치를 제3 값이 되는 것으로 결정하는 것; 및 현재의 에포크에서의 합의 프로세스의 제3 페이즈에서 쿼럼 검증의 성공을 결정하는 것에 응답하여, 합의 프로세스의 제3 페이즈에 대한 백업 노드의 가중치를 제4 값 - 제3 값은 제4 값보다 더 작음 - 이 되는 것으로 결정하는 것.

[0217] 옵션 사항으로, 네트워크 노드에 대한 제3 페이즈에서의 쿼럼 검증은, 다른 네트워크 노드로부터 미리 결정된 수의 수락 메시지를 수신하는 것을 포함하는데, 여기서 다른 네트워크 노드로부터의 수락 메시지의 각각은, 다른 네트워크 노드의 각각이 미리 결정된 수의 ECHO 메시지를 수락하였다는 것을 나타낸다.

[0218] 옵션 사항으로, EPOCH_CHANGE 메시지는 다수의 네트워크 노드 중의 네트워크 노드의 세트와 연관되는 서명의 세트를 더 포함하고, NEW_EPOCH 메시지는 EPOCH_CHANGE 메시지의 다이제스트를 포함한다.

[0219] 옵션 사항으로, 적어도 하나의 유효한 NEW_EPOCH 메시지가 유효한지의 여부를 검증하는 것은, 적어도 하나의 NEW_EPOCH 메시지 내의 EPOCH_CHANGE 메시지의 다이제스트가 유효한지의 여부를 검증하는 것을 포함하고, 적어도 하나의 NEW_EPOCH 메시지 내의 EPOCH_CHANGE 메시지의 다이제스트가 유효한지의 여부를 검증하는 것은, EPOCH_CHANGE 메시지 내의 서명의 세트가 유효한지의 여부를 검증하는 것을 포함한다.

[0220] 옵션 사항으로, 에포크 체인지가 수행될 필요가 있다고 결정하는 것은, 미리 결정된 시간 기간 내에 이전의 에포크에서 합의가 달성되지 않았다고 결정하는 것에 응답하여 에포크 체인지가 수행될 필요가 있다고 결정하는 것을 포함한다.

[0221] 옵션 사항으로, 적어도 하나의 프로세서는 새로운 프라이머리 노드를 갖는 새로운 에포크에서 동작하도록 또한 구성되는데, 새로운 에포크는 새로운 프라이머리 노드를 사용하여 복수의 네트워크 노드 사이에서 합의를 달성하기 위한 합의 프로세스를 포함한다.

[0222] 도 16을 참조하면, 도 16는, 본 명세서의 구현예에 따른, 프라이머리 노드 체인징 장치(1600)의 모듈을 예시하는 도면이다. 프라이머리 노드를 체인지하기 위한 장치(1600)는 블록체인 기술에 기초한 합의 시스템에 적용될 수 있다. 장치(1600)는 도 7 내지 도 9에서 도시되는 구현예에 대응한다. 예를 들면, 장치(1400)는 블록체인 네트워크의 네트워크 노드에서 구현될 수 있다. 장치(1600)는 다음의 것을 포함한다: 네트워크 노드 이외의 백업 노드로부터 EPOCH_CHANGE 메시지 - EPOCH_CHANGE 메시지는 에포크 체인지가 수행될 필요가 있다는 표시를 포함하고, 에포크 체인지는 현재의 프라이머리 노드를 갖는 현재의 에포크로부터 새로운 프라이머리 노드를 갖는 새로운 에포크로의 체인지를 야기함 - 를 수신하도록 구성되는 수신기 또는 수신 유닛(1602); EPOCH_CHANGE 메시지가 유효한지의 여부를 검증하도록 구성되는 검증 유닛(1604); EPOCH_CHANGE 메시지가 유효하다고 검증하는 것에 응답하여 NEW_EPOCH 메시지 - NEW_EPOCH 메시지는 EPOCH_CHANGE 메시지의 다이제스트를 포함함 - 를 다른 네트워크 노드로 전송하도록 구성되는 송신기 또는 송신 유닛(1606); 네트워크 노드 이외의 다수의 네트워크 노드 중 적어도 하나로부터 적어도 하나의 NEW_EPOCH 메시지를 수신하도록 또한 구성되는 수신기 또는 수신 유닛(1602); 적어도 하나의 NEW_EPOCH 메시지가 유효한지의 여부를 검증하도록 또한 구성되는 검증 유닛(1604); 적어도 하나의 NEW_EPOCH 메시지 중의 유효한 NEW_EPOCH 메시지의 수가 미리 결정된 임계치를 초과하는지의 여부를 결정하도록 구성되는 결정 유닛(1608); 및 유효한 NEW_EPOCH 메시지의 수가 제2 미리 결정된 임계치를 초과한다고 결정하는 것에 응답하여 백업 노드를 새로운 에포크에서 새로운 프라이머리 노드가 되는 것으로 결정하도록 또한 구성되는, 결정 유닛(1608).

[0223] 옵션 사항의 구현예에서, EPOCH_CHANGE 메시지는 백업 노드와 연관되는 가중치 합 및 다수의 네트워크 노드 중의 네트워크 노드의 세트와 연관되는 서명의 세트를 포함한다.

[0224] 옵션 사항의 구현예에서, EPOCH_CHANGE 메시지가 유효한지의 여부를 검증하는 것은, EPOCH_CHANGE 메시지 내의 가중치 합이 유효한지의 여부를 검증하는 것을 포함하고, EPOCH_CHANGE 메시지 내의 가중치 합이 유효한지의 여부를 검증하는 것은 서명의 세트가 유효한지의 여부를 검증하는 것을 포함한다.

[0225] 옵션 사항의 구현예에서, 적어도 하나의 NEW_EPOCH 메시지가 유효한지의 여부를 검증하는 것은, 적어도 하나의 NEW_EPOCH 메시지 내의 EPOCH_CHANGE 메시지의 다이제스트가 유효한지의 여부를 검증하는 것을 포함하고, 적어도 하나의 NEW_EPOCH 메시지 내의 EPOCH_CHANGE 메시지의 다이제스트가 유효한지의 여부를 검증하는 것은, EPOCH_CHANGE 메시지 내의 서명의 세트가 유효한지의 여부를 검증하는 것을 포함한다.

[0226] 도 16은 프라이머리 노드 체인징 장치(1600)의 내부 기능 모듈 및 구조를 예시하는 개략도이다. 본질적으로 실행 본체는, 전자 디바이스일 수 있으며, 전자 디바이스는 다음의 것을 포함한다: 적어도 하나의 프로세서; 및 적어도 하나의 프로세서의 실행 가능 명령어들을 저장하도록 구성되는 메모리.

[0227] 적어도 하나의 프로세서는, 네트워크 노드 이외의 백업 노드로부터 EPOCH_CHANGE 메시지 - EPOCH_CHANGE 메시지는 에포크 체인지가 수행될 필요가 있다는 표시를 포함하고, 에포크 체인지는 현재의 프라이머리 노드를 갖는 현재의 에포크로부터 새로운 프라이머리 노드를 갖는 새로운 에포크로의 체인지를 야기함 - 를 수신하도록; EPOCH_CHANGE 메시지가 유효한지의 여부를 검증하도록; EPOCH_CHANGE 메시지가 유효하다고 검증하는 것에 응답하여 NEW_EPOCH 메시지 - NEW_EPOCH 메시지는 EPOCH_CHANGE 메시지의 다이제스트를 포함함 - 를 다른 네트워크 노드로 전송하도록; 네트워크 노드 이외의 다수의 네트워크 노드 중 적어도 하나로부터 적어도 하나의 NEW_EPOCH 메시지를 수신하도록; 적어도 하나의 NEW_EPOCH 메시지가 유효한지의 여부를 검증하도록; 적어도 하나의 NEW_EPOCH 메시지 중의 유효한 NEW_EPOCH 메시지의 수가 제2 미리 결정된 임계치를 초과하는지의 여부를 결정하도록; 유효한 NEW_EPOCH 메시지의 수가 미리 결정된 임계치를 초과한다고 결정하는 것에 응답하여 백업

노드를 새로운 에포크에서 새로운 프라이머리 노드가 되는 것으로 결정하도록 구성된다.

[0228] 옵션 사항으로, EPOCH_CHANGE 메시지는 백업 노드와 연관되는 가중치 합 및 다수의 네트워크 노드 중의 네트워크 노드의 세트와 연관되는 서명의 세트를 포함한다.

[0229] 옵션 사항으로, EPOCH_CHANGE 메시지가 유효한지의 여부를 검증하는 것은, EPOCH_CHANGE 메시지 내의 가중치 합이 유효한지의 여부를 검증하는 것을 포함하고, EPOCH_CHANGE 메시지 내의 가중치 합이 유효한지의 여부를 검증하는 것은 서명의 세트가 유효한지의 여부를 검증하는 것을 포함한다.

[0230] 옵션 사항으로, 적어도 하나의 NEW_EPOCH 메시지가 유효한지의 여부를 검증하는 것은, 적어도 하나의 NEW_EPOCH 메시지 내의 EPOCH_CHANGE 메시지의 다이제스트가 유효한지의 여부를 검증하는 것을 포함하고, 적어도 하나의 NEW_EPOCH 메시지 내의 EPOCH_CHANGE 메시지의 다이제스트가 유효한지의 여부를 검증하는 것은, EPOCH_CHANGE 메시지 내의 서명의 세트가 유효한지의 여부를 검증하는 것을 포함한다.

[0231] 도 17을 참조하면, 도 17은, 본 명세서의 구현예에 따른, 복구 장치(1700)의 모듈을 예시하는 도면이다. 복구를 위한 장치(1700)는 블록체인 기술에 기초한 합의 시스템에 적용될 수 있다. 장치(1700)는 도 10 내지 도 12에서 도시되는 구현예에 대응할 수 있다. 예를 들면, 장치(1700)는 블록체인 네트워크의 네트워크 노드에서 구현될 수 있다. 장치(1700)는 다음의 것을 포함한다: 블록체인 네트워크의 네트워크 노드 - 네트워크 노드는 타겟 시퀀스 번호의 타겟 트랜잭션을 복구할 것임 - 에 의해, 상태 요청 메시지를, 블록체인 네트워크의 다수의 다른 네트워크 노드로 브로드캐스트하도록 구성되는 브로드캐스팅 유닛(1702); 다수의 다른 네트워크 노드로부터 다수의 상태 응답 메시지 - 다수의 상태 응답 메시지의 각각은 시퀀스 번호를 포함함 - 를 수신하도록 구성되는 수신기(1704) 또는 수신 유닛(1704); 다수의 상태 응답 메시지 - 다수의 상태 응답 메시지의 각각은 타겟 시퀀스 번호를 포함함 - 가 미리 결정된 임계치를 초과한다고 결정하는 것에 응답하여 타겟 시퀀스 번호에 기초하여 타겟 시퀀스 번호를 식별하도록 구성되는 식별 유닛(1706); 요청 메시지 - 요청 메시지는 다수의 다른 네트워크 노드의 각각에게 ECHO 메시지를 요청하고, ECHO 메시지는, 타겟 시퀀스 번호를 갖는 타겟 트랜잭션에 대해 다수의 다른 네트워크 노드 사이에서 합의를 달성하기 위한, 다수의 다른 네트워크 노드의 각각에 의해 송신되는 메시지 포함하고, ECHO 메시지는 타겟 트랜잭션의 일부 및 다수의 다른 네트워크 노드의 각각의 서명을 포함함 - 를 다수의 다른 네트워크 노드로 전송하도록 구성되는 송신기(1708) 또는 송신 유닛(1708); 다수의 다른 네트워크 노드로부터 다수의 ECHO 메시지를 수신하도록 또한 구성되는 수신기(1704) 또는 수신 유닛(1704); 다수의 ECHO 메시지 중의 유효한 ECHO 메시지 - 다수의 유효한 ECHO 메시지의 각각은 타겟 시퀀스 번호를 포함함 - 의 수를 결정하도록 구성되는 결정 유닛(1710); 유효한 ECHO 메시지의 수가 미리 결정된 임계치를 초과한다고 결정하는 것에 응답하여 유효한 ECHO 메시지의 수에 기초하여 네트워크 노드에서 타겟 시퀀스 번호를 갖는 타겟 트랜잭션을 복구하도록 구성되는 복구 유닛(1712); 및 네트워크 노드가 복구되었다는 것을 나타내는 메시지를 다수의 다른 네트워크 노드로 전송하도록 또한 구성되는 송신기(1708).

[0232] 옵션 사항의 구현예에서, 다수의 네트워크 노드는 프라이머리 노드 및 하나 이상의 백업 노드를 포함한다.

[0233] 옵션 사항의 구현예에서, 네트워크 노드는 프라이머리 노드 또는 백업 노드이다.

[0234] 옵션 사항의 구현예에서, 요청 메시지는 타겟 시퀀스 번호를 포함한다.

[0235] 옵션 사항의 구현예에서, 복구 장치(1700)는 다음의 것을 더 포함한다: 네트워크 노드 이외의 다수의 다른 네트워크 노드의 각각에 의해, ECHO 메시지를 네트워크 노드로 전송하기 이전에 요청 메시지를 검증하도록 구성되는 검증 유닛(1714).

[0236] 옵션 사항의 구현예에서, 검증 유닛(1714)은 또한 ECHO 메시지의 각각이 유효한지의 여부를 검증하도록 구성되는데, ECHO 메시지의 각각이 유효한지의 여부를 검증하는 것은, 머클 트리를 사용하여 ECHO 메시지의 각각이 유효한지의 여부를 검증하는 것을 포함한다.

[0237] 옵션 사항의 구현예에서, ECHO 메시지의 각각이 유효한지의 여부를 검증하는 것은, ECHO 메시지의 서명이 유효한지의 여부를 검증하는 것을 더 포함한다.

[0238] 옵션 사항의 구현예에서, ECHO 메시지의 각각은, 타겟 트랜잭션과 연관되는 다수의 소거 코드(EC) 블록 중 적어도 하나를 더 포함하는데, 다수의 EC 블록은 타겟 트랜잭션을 사용하여 EC 코드에 따라 생성된다.

[0239] 옵션 사항의 구현예에서, 유효한 ECHO 메시지의 수에 기초하여 네트워크 노드에서 타겟 시퀀스 번호를 갖는 타겟 트랜잭션을 복구하는 것은, 다수의 유효한 ECHO 메시지 내에 있는 복수의 EC 블록의 서브세트를 사용하여 타

겟 트랜잭션을 재구성하는 것을 포함한다.

[0240] 옵션 사항의 구현예에서, 네트워크 노드가 복구되었다는 것을 나타내는 다수의 다른 네트워크 노드에 대한 메시지는 다수의 유효한 ECHO 메시지 내의 서명의 세트 및 타겟 시퀀스 번호를 포함한다.

[0241] 이전 구현예에서 예시되는 시스템, 장치, 모듈, 또는 유닛은, 컴퓨터 칩 또는 엔티티를 사용하는 것에 의해 구현될 수 있거나, 또는 소정의 기능을 갖는 제품을 사용하는 것에 의해 구현될 수 있다. 통상적인 구현 디바이스는 컴퓨터이고, 컴퓨터는, 퍼스널 컴퓨터, 랩탑 컴퓨터, 셀룰러 전화, 카메라 폰, 스마트폰, 개인 휴대형 정보 단말(personal digital assistant), 미디어 플레이어, 내비게이션 디바이스, 전자 메일 수신 및 전송 디바이스, 게임 콘솔, 태블릿 컴퓨터, 웨어러블 디바이스, 또는 이들 디바이스의 임의의 조합일 수 있다.

[0242] 장치 내의 각각의 유닛의 기능 및 역할의 구현 프로세스에 대해서는, 앞선 방법에서의 대응하는 단계의 구현 프로세스에 대한 참조가 이루어질 수 있다. 세부 사항은 간략화를 위해 여기서는 생략된다.

[0243] 장치 구현예가 기본적으로 방법 구현예에 대응하기 때문에, 관련된 부분에 대해서는, 방법 구현예에서의 관련된 설명에 대한 참조가 이루어질 수 있다. 앞서 설명된 장치 구현예는 예에 불과하다. 분리된 부분으로서 설명되는 유닛은 물리적으로 분리될 수도 있거나 또는 분리되지 않을 수도 있고, 유닛으로서 디스플레이 되는 부분은 물리적 유닛일 수도 있거나 또는 아닐 수도 있고, 하나의 위치에 위치될 수도 있거나, 또는 복수의 네트워크 유닛 상에 분산될 수도 있다. 모듈의 일부 또는 전부는 본 명세서의 솔루션의 목적을 달성하기 위한 실제 요구에 기초하여 선택될 수 있다. 기술 분야에서 통상의 기술을 가진 자는 창의적인 노력 없이 본 출원의 구현예를 이해하고 구현할 수 있다.

[0244] 도 17은 복구 장치(1700)의 내부 기능 모듈 및 구조를 예시하는 개략도이다. 본질적으로 실행 본체는, 전자 디바이스일 수 있으며, 전자 디바이스는 다음의 것을 포함한다: 적어도 하나의 프로세서; 및 적어도 하나의 프로세서의 실행 가능 명령어들을 저장하도록 구성되는 메모리.

[0245] 적어도 하나의 프로세서는, 블록체인 네트워크의 네트워크 노드 - 네트워크 노드는 타겟 시퀀스 번호의 타겟 트랜잭션을 복구할 것임 - 에 의해, 상태 요청 메시지를, 블록체인 네트워크의 다수의 다른 네트워크 노드로 브로드캐스트하도록; 다수의 다른 네트워크 노드로부터 다수의 상태 응답 메시지 - 다수의 상태 응답 메시지의 각각은 시퀀스 번호를 포함함 - 를 수신하도록; 다수의 상태 응답 메시지 - 다수의 상태 응답 메시지의 각각은 타겟 시퀀스 번호를 포함함 - 가 미리 결정된 임계치를 초과한다고 결정하는 것에 응답하여 타겟 시퀀스 번호에 기초하여 타겟 시퀀스 번호를 식별하도록; 요청 메시지 - 요청 메시지는 다수의 다른 네트워크 노드의 각각에게 ECHO 메시지를 요청하고, ECHO 메시지는, 타겟 시퀀스 번호를 갖는 타겟 트랜잭션에 대해 다수의 다른 네트워크 노드 사이에서 합의를 달성하기 위한, 다수의 다른 네트워크 노드의 각각에 의해 송신되는 메시지 포함하고, ECHO 메시지는 타겟 트랜잭션의 일부 및 다수의 다른 네트워크 노드의 각각의 서명을 포함함 - 를 다수의 다른 네트워크 노드로 전송하도록; 복수의 다른 네트워크 노드로부터 다수의 ECHO 메시지를 수신하도록; 다수의 ECHO 메시지 중의 유효한 ECHO 메시지 - 다수의 유효한 ECHO 메시지의 각각은 타겟 시퀀스 번호를 포함함 - 의 수를 결정하도록; 유효한 ECHO 메시지의 수가 미리 결정된 임계치를 초과한다고 결정하는 것에 응답하여 유효한 ECHO 메시지의 수에 기초하여 네트워크 노드에서 타겟 시퀀스 번호를 갖는 타겟 트랜잭션을 복구하도록; 그리고 네트워크 노드가 복구되었다는 것을 나타내는 메시지를 다수의 다른 네트워크 노드로 전송하도록 구성된다.

[0246] 옵션 사항으로, 다수의 네트워크 노드는 프라이머리 노드 및 하나 이상의 백업 노드를 포함한다.

[0247] 옵션 사항으로, 네트워크 노드는 프라이머리 노드 또는 백업 노드이다.

[0248] 옵션 사항으로, 요청 메시지는 타겟 시퀀스 번호를 포함한다.

[0249] 옵션 사항으로, 적어도 하나의 프로세서는 또한, 네트워크 노드 이외의 다수의 다른 네트워크 노드의 각각에 의해, ECHO 메시지를 네트워크 노드로 전송하기 이전에 요청 메시지를 검증하도록 구성된다.

[0250] 옵션 사항으로, 적어도 하나의 프로세서는 또한 ECHO 메시지의 각각이 유효한지의 여부를 검증하도록 구성되는데, ECHO 메시지의 각각이 유효한지의 여부를 검증하는 것은, 머클 트리를 사용하여 ECHO 메시지의 각각이 유효한지의 여부를 검증하는 것을 포함한다.

[0251] 옵션 사항으로, ECHO 메시지의 각각이 유효한지의 여부를 검증하는 것은, ECHO 메시지 내의 서명이 유효한지의 여부를 검증하는 것을 더 포함한다.

[0252] 옵션 사항으로, ECHO 메시지의 각각은, 타겟 트랜잭션과 연관되는 다수의 소거 코드(EC) 블록 중 적어도 하나를

더 포함하는데, 다수의 EC 블록은 타겟 트랜잭션을 사용하여 EC 코드에 따라 생성된다.

[0253] 옵션 사항으로, 유효한 ECHO 메시지의 수에 기초하여 네트워크 노드에서 타겟 시퀀스 번호를 갖는 타겟 트랜잭션을 복구하는 것은, 다수의 유효한 ECHO 메시지 내에 있는 다수의 EC 블록의 서브세트를 사용하여 타겟 트랜잭션을 재구성하는 것을 포함한다.

[0254] 옵션 사항으로, 네트워크 노드가 복구되었다는 것을 나타내는 다수의 다른 네트워크 노드에 대한 메시지는 다수의 유효한 ECHO 메시지 내의 서명의 세트 및 타겟 시퀀스 번호를 포함한다.

[0255] 본 명세서에서 설명되는 주제(subject matter)의 구현예 및 액션 및 동작은, 디지털 전자 회로부(circuitry)에서, 유형적으로 구체화된 컴퓨터 소프트웨어 또는 펌웨어에서, 본 명세서에서 개시되는 구조 및 그들의 구조적 등가물을 비롯한, 컴퓨터 하드웨어에서, 또는 그들 중 하나 이상의 조합에서 구현될 수 있다. 본 명세서에서 설명되는 주제의 구현예는 하나 이상의 컴퓨터 프로그램, 예를 들면, 데이터 프로세싱 장치의 실행에 의해, 또는 데이터 프로세싱의 동작을 제어하기 위해, 컴퓨터 프로그램 캐리어 상에 인코딩되는 하나 이상의 컴퓨터 프로그램, 예를 들면, 컴퓨터 프로그램 명령어들의 하나 이상의 모듈로서 구현될 수 있다. 캐리어는 유형의 비일시적 컴퓨터 저장 매체일 수도 있다. 대안적으로 또는 추가적으로, 캐리어는 인위적으로 생성되는 전파된 신호, 예를 들면, 데이터 프로세싱 장치에 의한 실행을 위해 적절한 수신기 장치로의 송신을 위해 정보를 인코딩하기 위해 생성되는 머신 생성 전기, 광학, 또는 전자기 신호일 수도 있다. 컴퓨터 저장 매체는, 머신 판독 가능 스토리지 디바이스, 머신 판독 가능 스토리지 기판, 랜덤 또는 직렬 액세스 메모리 디바이스, 또는 그들의 하나 이상의 조합일 수 있거나 또는 그 일부일 수 있다. 컴퓨터 저장 매체는 전파된 신호가 아니다.

[0256] 용어 "데이터 프로세싱 장치"는, 예로서, 프로그래머블 프로세서, 컴퓨터, 또는 다수의 프로세서 또는 컴퓨터를 비롯한, 데이터를 프로세싱하기 위한 모든 종류의 장치, 디바이스, 및 머신을 포괄한다. 데이터 프로세싱 장치는 특수 목적의 논리 회로부, 예를 들면, FPGA(field programmable gate array; 필드 프로그래머블 게이트 어레이), ASIC(application specific integrated circuit; 주문형 집적 회로), 또는 GPU(graphics processing unit; 그래픽 프로세싱 유닛)를 포함할 수 있다. 장치는 또한, 하드웨어 이외에, 컴퓨터 프로그램에 대한 실행 환경을 생성하는 코드, 예를 들면, 프로세서 펌웨어, 프로토콜 스택, 데이터베이스 관리 시스템, 오퍼레이팅 시스템, 또는 그들 중 하나 이상의 조합을 구성하는 코드를 포함할 수 있다.

[0257] 프로그램, 소프트웨어, 소프트웨어 애플리케이션, 앱, 모듈, 소프트웨어 모듈, 엔진, 스크립트, 또는 코드로 또한 칭해질 수도 있거나 또는 설명될 수도 있는 컴퓨터 프로그램은, 컴파일식 또는 인터프리트식 언어, 또는 선언적 또는 절차적 언어를 비롯한, 임의의 형태의 프로그래밍 언어로 작성될 수 있고; 그것은 독립형 프로그램으로서 또는 모듈, 컴포넌트, 엔진, 서브루틴, 또는 컴퓨팅 환경에서 실행하기에 적절한 다른 유닛으로서 배치될 수 있는 것을 비롯하여, 임의의 형태로 배치될 수 있는데, 환경은 하나 이상의 위치에서 데이터 통신 네트워크에 의해 인터커넥트되는 하나 이상의 컴퓨터를 포함할 수도 있다.

[0258] 컴퓨터 프로그램은 파일 시스템의 파일에 대응할 수도 있지만, 그러나 대응하지 않아도 된다. 컴퓨터 프로그램은, 다른 프로그램 또는 데이터를 유지하는 파일의 부분에, 예를 들면, 마크업 언어 문서(markup language document)에 저장되는 하나 이상의 스크립트에, 문제가 되는 프로그램에 전용되는 단일의 파일에, 또는 다수의 협력 파일(coordinated file)에, 예를 들면, 코드의 하나 이상의 모듈, 서브 프로그램, 또는 부분을 저장하는 파일에 저장될 수 있다.

[0259] 본 명세서에서 설명되는 프로세스 및 논리 흐름은 입력 데이터에 대해 동작하여 출력을 생성하는 것에 의해 동작을 수행하는 하나 이상의 컴퓨터 프로그램을 실행하는 하나 이상의 컴퓨터에 의해 수행될 수 있다. 프로세스 및 논리 흐름은 또한, 특수 목적 논리 회로부, 예를 들면, FPGA, ASIC, 또는 GPU에 의해, 또는 특수 목적 논리 회로부 및 하나 이상의 프로그래밍된 컴퓨터의 조합에 의해 수행될 수 있다.

[0260] 컴퓨터 프로그램의 실행에 적절한 컴퓨터는, 범용 또는 특수 목적 마이크로프로세서 또는 둘 모두, 또는 임의의 다른 종류의 중앙 프로세싱 유닛에 기초할 수 있다. 일반적으로, 중앙 프로세싱 유닛은 리드 온리 메모리 또는 랜덤 액세스 메모리 또는 둘 모두로부터 명령어 및 데이터를 수신할 것이다. 컴퓨터의 엘리먼트는 명령어들을 실행하기 위한 중앙 프로세싱 유닛 및 명령어 및 데이터를 저장하기 위한 하나 이상의 메모리 디바이스를 포함할 수 있다. 중앙 프로세싱 유닛 및 메모리는 특수 목적 논리 회로부에 의해 보충될 수 있거나, 또는 그 안에 통합될 수 있다.

[0261] 일반적으로, 컴퓨터는 적어도 하나의 비일시적 컴퓨터 판독 가능 저장 매체(컴퓨터 판독 가능 메모리로도 또한 칭해짐)에 커플링될 것이다. 컴퓨터에 커플링되는 저장 매체는 컴퓨터의 내부 컴포넌트(예를 들면, 내장된 하드

드라이브) 또는 외부 컴포넌트(예를 들면, 범용 직렬 버스(universal serial bus; USB) 하드 드라이브 또는 네트워크를 통해 액세스되는 저장 시스템)일 수 있다. 저장 매체의 예는, 예를 들면, 자기, 광자기, 또는 광학 디스크, 솔리드 스테이트 드라이브, 클라우드 스토리지 시스템과 같은 네트워크 저장 리소스, 또는 다른 타입의 저장 매체를 포함할 수 있다. 그러나, 컴퓨터는 그러한 디바이스를 구비할 필요가 없다. 또한, 컴퓨터는 다른 디바이스, 예를 들면, 단지 몇몇의 이름을 거론하자면, 이동 전화, 개인 휴대형 정보 단말(personal digital assistant; PDA), 모바일 오디오 또는 비디오 플레이어, 게임 콘솔, 전지구 위치 결정 시스템(global positioning system; GPS) 수신기, 또는 휴대용 스토리지 디바이스, 예를 들면, 범용 직렬 버스(USB) 플래시 드라이브에 임베딩될 수 있다.

[0262] 유저와의 상호 작용을 제공하기 위해, 본 명세서에서 설명되는 주제의 구현예는, 정보를 유저에게 디스플레이하기 위한 디스플레이 디바이스, 예를 들면, LCD(liquid crystal display; 액정 디스플레이) 모니터, 및 유저가 컴퓨터에 입력을 제공할 수 있게 하는 입력 디바이스, 예를 들면, 키보드 및 포인팅 디바이스, 예를 들면, 마우스, 트랙볼 또는 터치패드를 구비하는 컴퓨터 상에서 구현될 수 있거나, 또는 그 컴퓨터와 통신하도록 구성될 수 있다. 유저와의 상호 작용을 제공하기 위해 다른 종류의 디바이스가 또한 사용될 수 있으며; 유저에게 제공되는 피드백은 임의의 형태의 감각 피드백(sensory feedback), 예를 들면, 시각적 피드백, 청각적 피드백, 또는 촉각적 피드백을 사용할 수 있고; 유저로부터의 입력은, 음향, 음성, 또는 촉각 입력을 비롯한, 임의의 형태로 수신될 수 있다. 또한, 컴퓨터는 유저에 의해 사용되는 디바이스로 문서를 전송하는 것 및 유저에 의해 사용되는 디바이스로부터 문서를 수신하는 것에 의해; 예를 들면, 웹 브라우저로부터 수신되는 요청에 응답하여 웹 페이지를 유저의 디바이스 상의 웹 브라우저로 전송하는 것에 의해, 또는 유저 디바이스, 예를 들면, 스마트폰 또는 전자적 태블릿 상에서 실행되는 앱과 상호 작용하는 것에 의해, 유저와 상호 작용할 수 있다. 또한, 컴퓨터는, 텍스트 메시지 또는 다른 형태의 메시지를 개인용 디바이스, 예를 들면, 메시징 애플리케이션을 실행하고 있는 스마트폰으로 전송하는 것, 및 응답으로 유저로부터 응답 메시지를 수신하는 것에 의해 유저와 상호 작용할 수 있다.

[0263] 본 명세서는, 시스템, 장치, 및 컴퓨터 프로그램 컴포넌트와 관련하여, 용어 "하도록 구성되는(configured to)"을 사용한다. 하나 이상의 컴퓨터의 시스템이 특정한 동작 또는 액션을 수행하도록 구성되는 것은, 시스템이, 동작시 시스템으로 하여금 동작 또는 액션을 수행하게 하는 소프트웨어, 펌웨어, 하드웨어, 또는 그들의 조합을 시스템 상에 설치하였다는 것을 의미한다. 하나 이상의 컴퓨터 프로그램이 특정한 동작 또는 액션을 수행하도록 구성되는 것은, 데이터 프로세싱 장치에 의해 실행될 때, 장치로 하여금 동작 또는 액션을 수행하게 하는 명령어들을 하나 이상의 프로그램이 포함한다는 것을 의미한다. 특수 목적 논리 회로부가 특정한 동작 또는 액션을 수행하도록 구성되는 것은, 그 회로부가, 동작 또는 액션을 수행하는 전자적 논리를 구비한다는 것을 의미한다.

[0264] 본 명세서가 많은 특정한 구현 세부 사항을 포함하지만, 이들은, 청구범위 그 자체에 의해 정의되는 청구되고 있는 것의 범위에 대한 제한으로서 해석되는 것이 아니라, 오히려, 특정한 구현예에 고유할 수도 있는 피처의 설명으로 해석되어야 한다. 별개의 구현예의 맥락에서 본 명세서에서 설명되는 소정의 피처는 또한, 단일의 구현예에서 조합하여 실현될 수 있다. 반대로, 단일의 구현예의 맥락에서 설명되는 다양한 피처는 또한, 다수의 구현예에서 별개로, 또는 임의의 적절한 하위 조합으로 실현될 수 있다. 또한, 비록 피처가 소정의 조합에서 작용하는 것으로 상기에서 설명될 수도 있고 심지어 초기에 그와 같이 청구될 수도 있지만, 주장된 조합으로부터의 하나 이상의 피처는, 몇몇 경우에, 조합으로부터 제거될 수 있고, 청구항은 부조합 또는 부조합의 변형에 관한 것일 수도 있다.

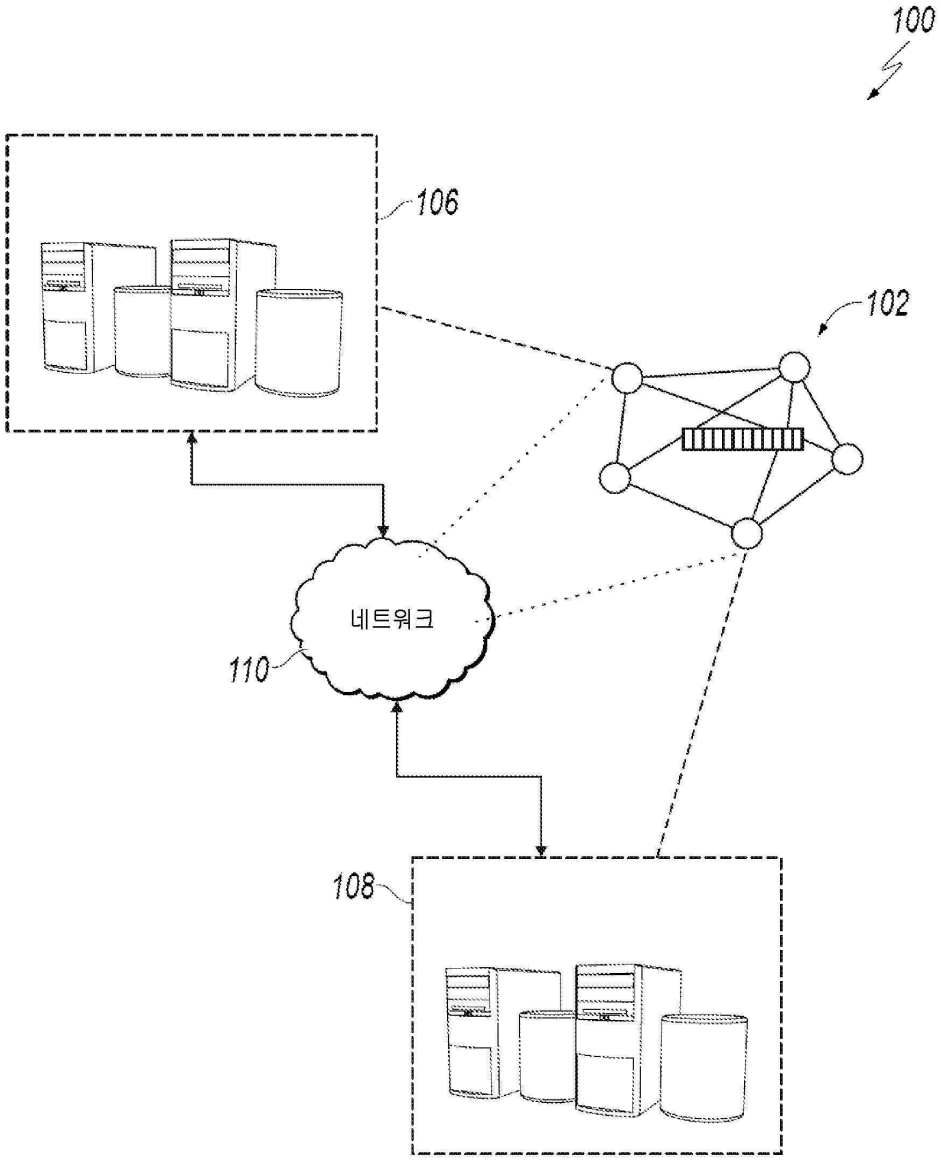
[0265] 마찬가지로, 동작이 도면에서 묘사되고 청구범위에서 특정한 순서로 열거되지만, 이것은, 바람직한 결과를 달성하기 위해, 그러한 동작이 도시되는 특정한 순서로 또는 순차적인 순서로 수행되어야 한다는 것, 또는 모든 예시된 동작이 수행되어야 한다는 것을 규정하는 것으로 이해되지 않아야 한다. 소정의 상황에서, 멀티태스킹 및 병렬 프로세싱이 유리할 수도 있다. 또한, 상기에서 설명되는 구현예에서의 다양한 시스템 모듈 및 컴포넌트의 분리는, 모든 구현예에서 그러한 분리를 필요로 하는 것으로 이해되어서는 안되며, 설명된 프로그램 컴포넌트 및 시스템은 일반적으로 단일의 소프트웨어 제품에 함께 통합될 수 있거나 또는 다수의 소프트웨어 제품으로 패키징될 수 있다는 것이 이해되어야 한다.

[0266] 본 주제의 특정한 구현예가 설명되었다. 다른 구현예는 다음의 청구범위의 범위 내에 있다. 예를 들면, 청구범위에서 열거되는 액션은 상이한 순서로 수행될 수 있으며 여전히 바람직한 결과를 달성할 수 있다. 하나의 예로서, 첨부의 도면에서 묘사되는 프로세스는, 바람직한 결과를 달성하기 위해, 도시되는 특정한 순서, 또는 순차

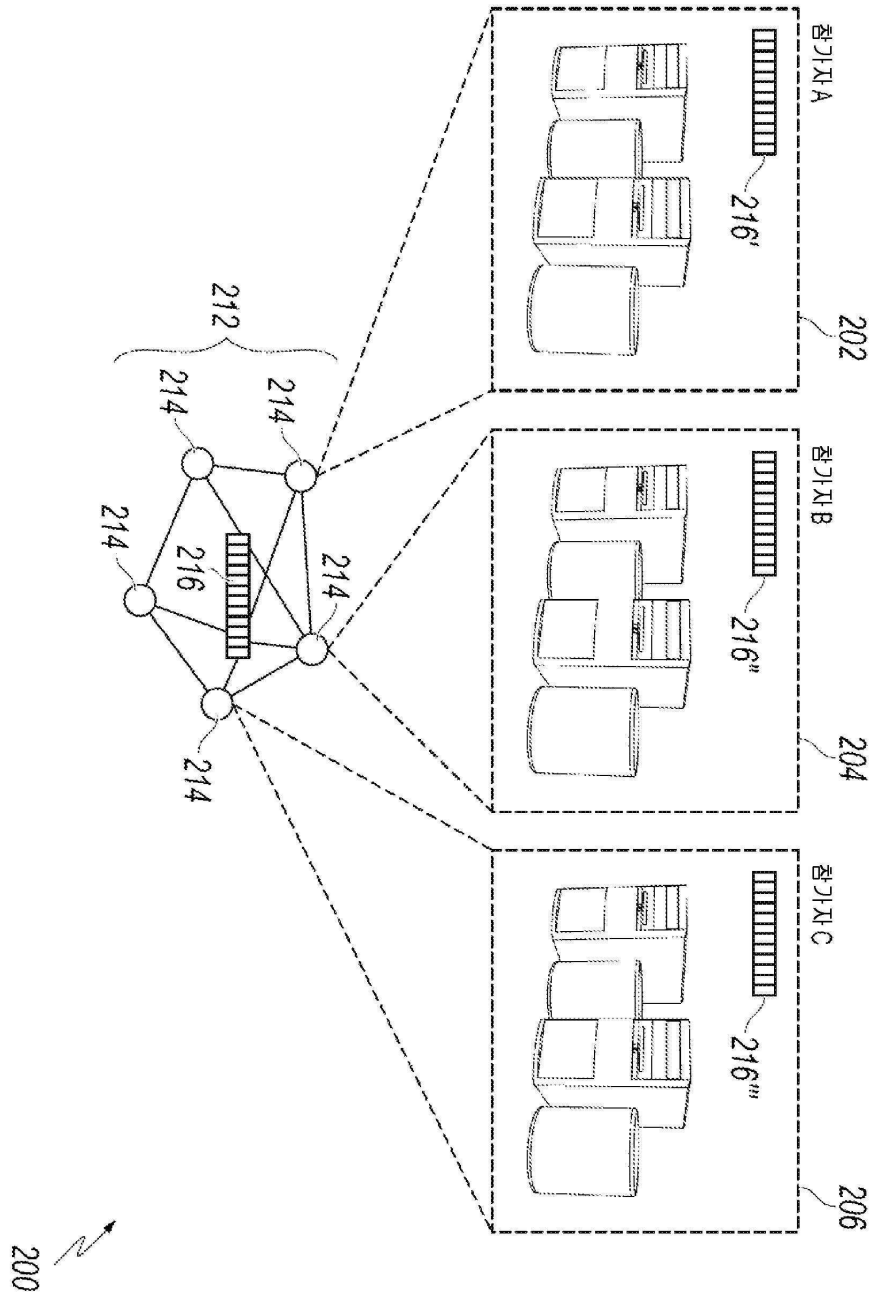
적인 순서를 반드시 필요로 하지는 않는다. 몇몇 경우에, 멀티태스킹 및 병렬 프로세싱이 유리할 수도 있다.

도면

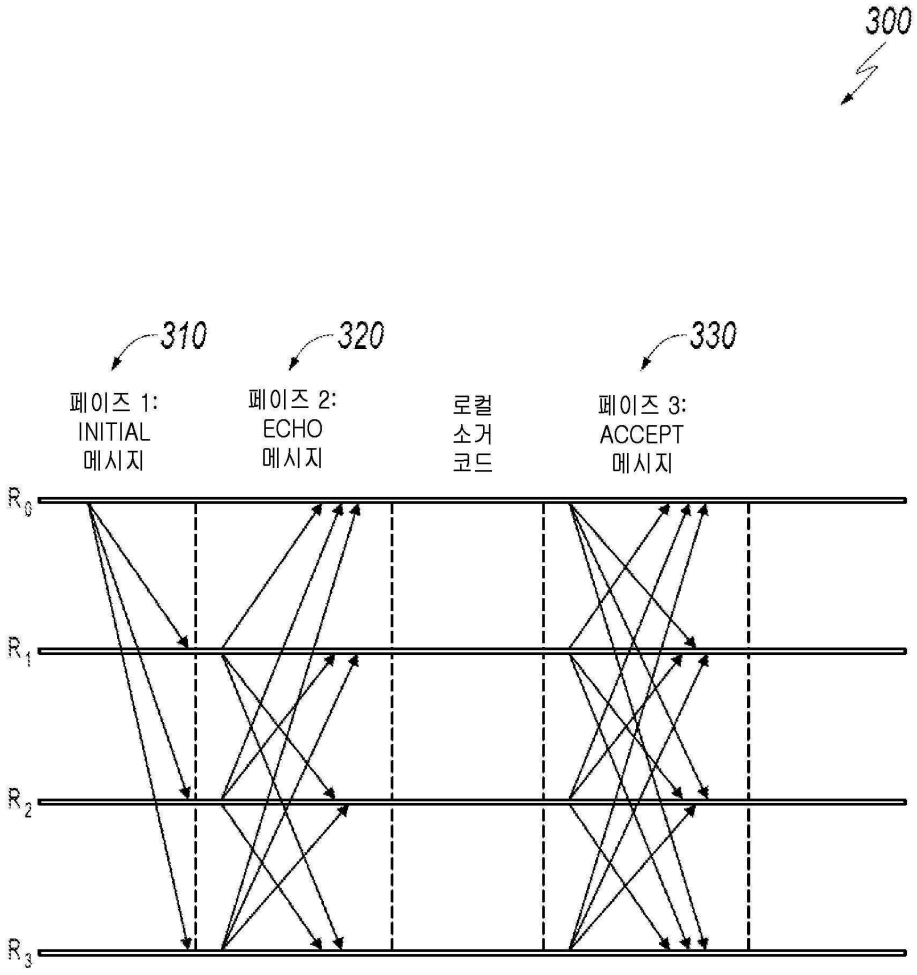
도면1



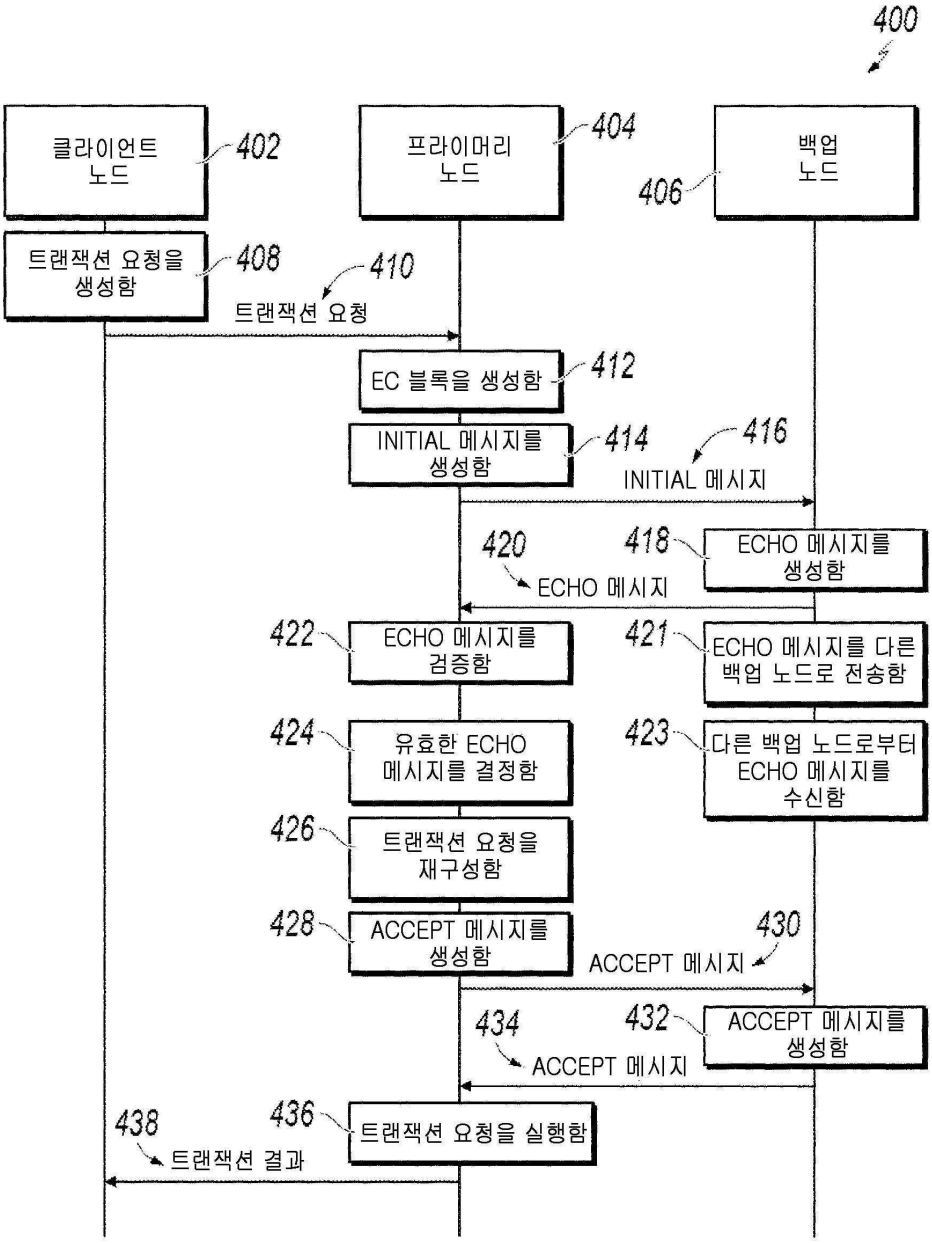
도면2


참가자 A
216'
202
참가자 B
216"
204
참가자 C
216'''
206
212
214
214
214
216
214
214
200

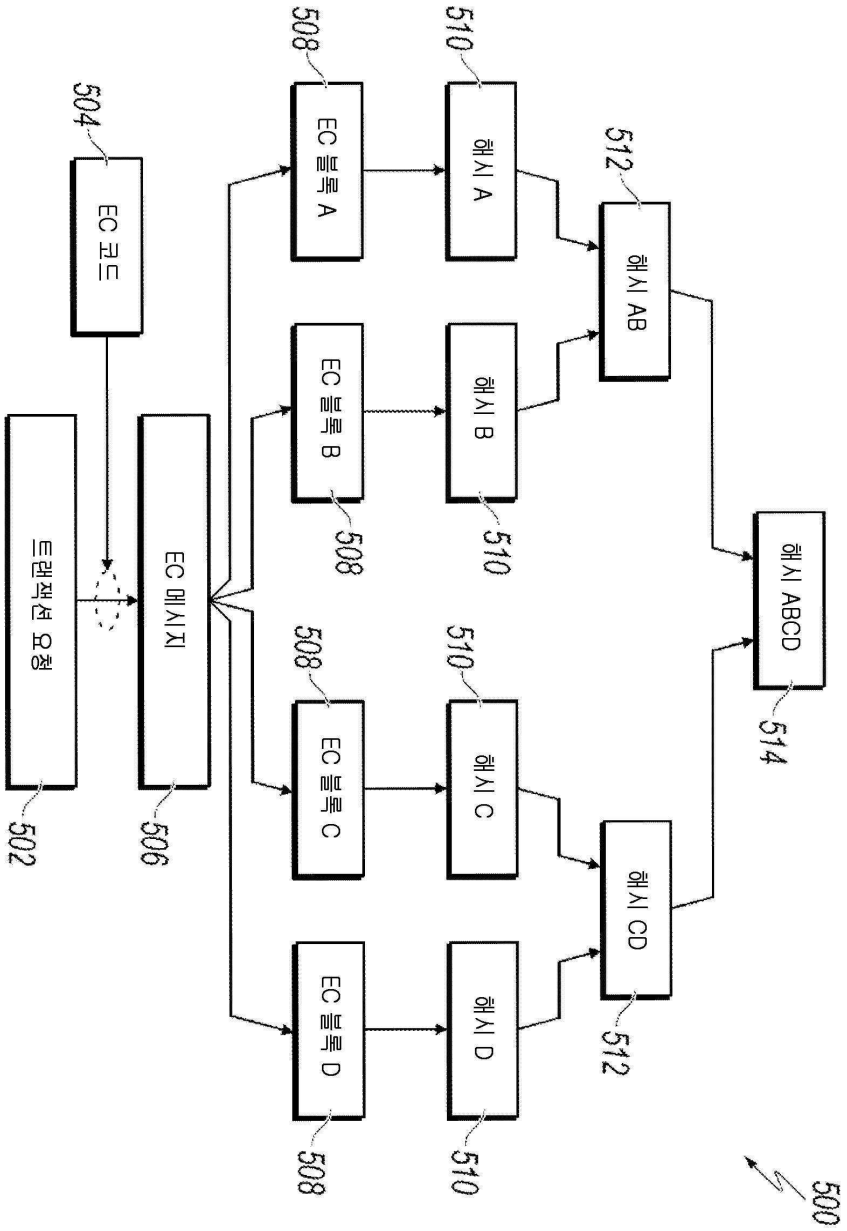
도면3


300
310
320
330
페이즈 1:
INITIAL
메시지
페이즈 2:
ECHO
메시지
로컬
소거
코드
페이즈 3:
ACCEPT
메시지
R0
R1
R2
R3

도면4


400
클라이언트 노드
402
프라이머리 노드
404
백업 노드
406
트랜잭션 요청을 생성함
408
410
트랜잭션 요청
EC 블록을 생성함
412
INITIAL 메시지를 생성함
414
416
INITIAL 메시지
420
ECHO 메시지
418
ECHO 메시지를 생성함
422
ECHO 메시지를 검증함
421
ECHO 메시지를 다른 백업 노드로 전송함
424
유효한 ECHO 메시지를 결정함
423
다른 백업 노드로부터 ECHO 메시지를 수신함
426
트랜잭션 요청을 재구성함
428
ACCEPT 메시지를 생성함
430
ACCEPT 메시지
434
ACCEPT 메시지
432
ACCEPT 메시지를 생성함
438
트랜잭션 결과
436
트랜잭션 요청을 실행함

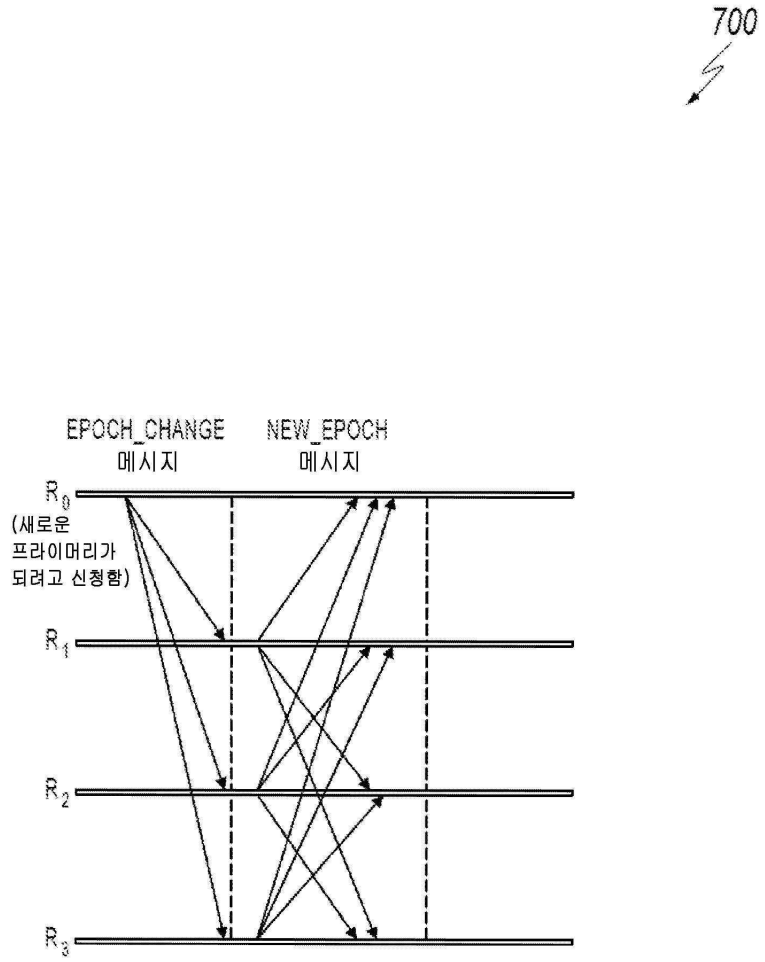
도면5



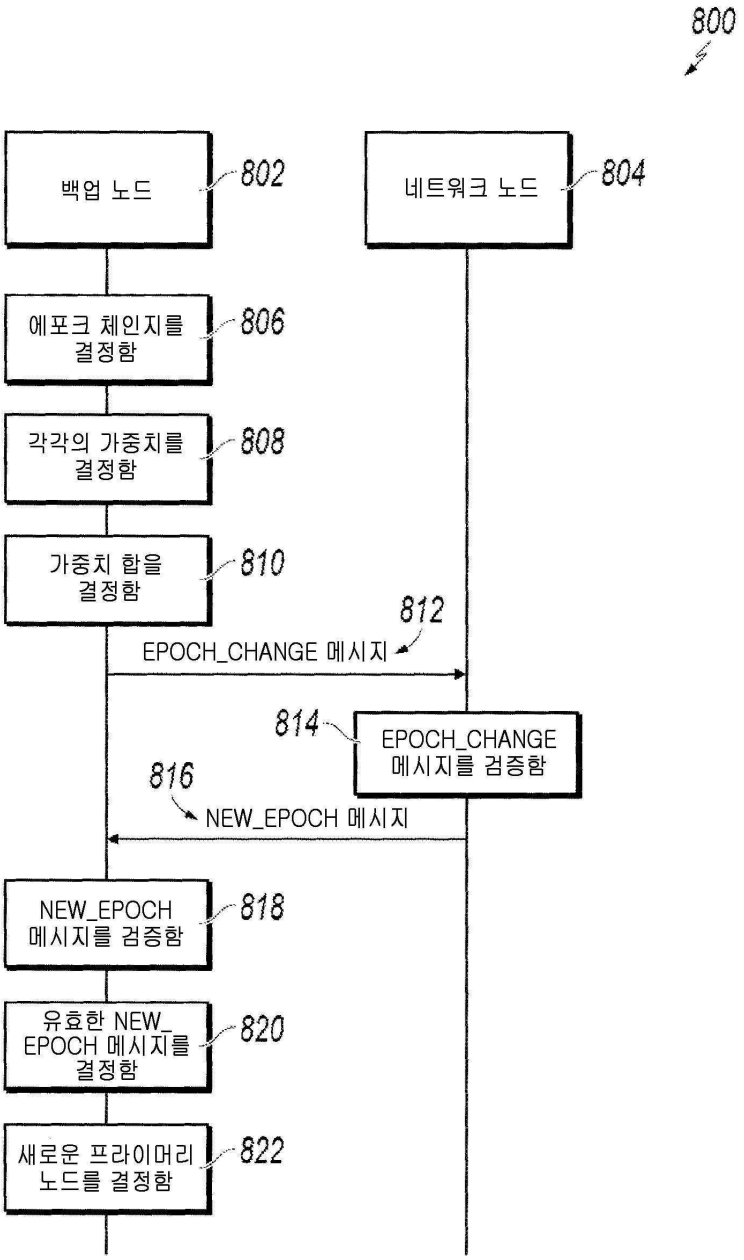
도면6

Initial<epoch, tx_root_hash, ec_block_hash, ec_block, seq, j>
Echo<epoch, tx_root_hash, ec_block_hash, ec_block, seq, sign_proof, j>
Accept<epoch, tx_root_hash, seq, sign_proofs, j>

도면7


700
EPOCH_CHANGE
메시지
NEW_EPOCH
메시지
R0
(새로운
프라이머리가
되려고 신청함)
R1
R2
R3

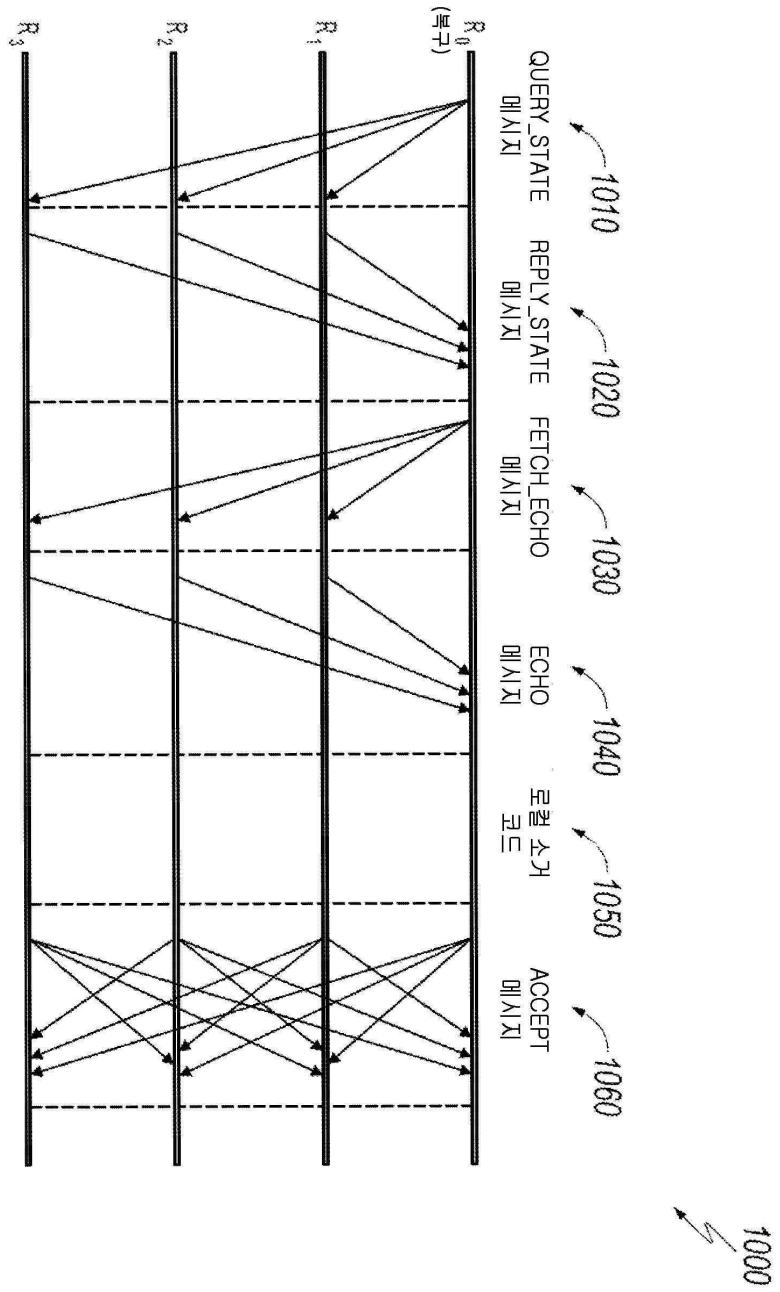
도면8



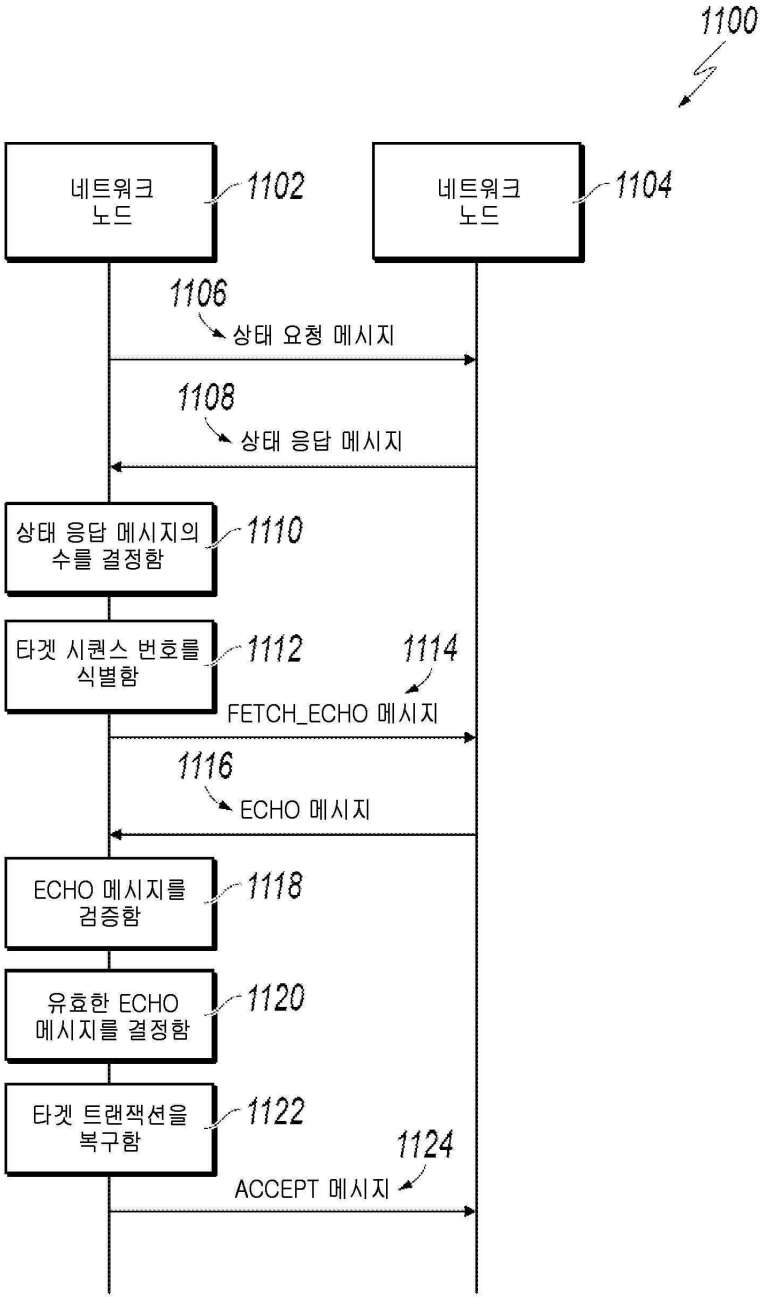
도면9

Epoch_change <weight, epoch+1, Echo{}, Accept{}, j>
Echo {<epoch, seq, sign_proof, j>}
Accept {<epoch, tx_root_hash, sign_proofs, seq, j>}
New_epoch <epoch+1, i, j, seq, ec_digest>

도면10


R_0 (복구)
R_1
R_2
R_3
QUERY_STATE 메시지
REPLY_STATE 메시지
FETCH_ECHO 메시지
ECHO 메시지
로컬 소거 코드
ACCEPT 메시지
1010
1020
1030
1040
1050
1060
1000

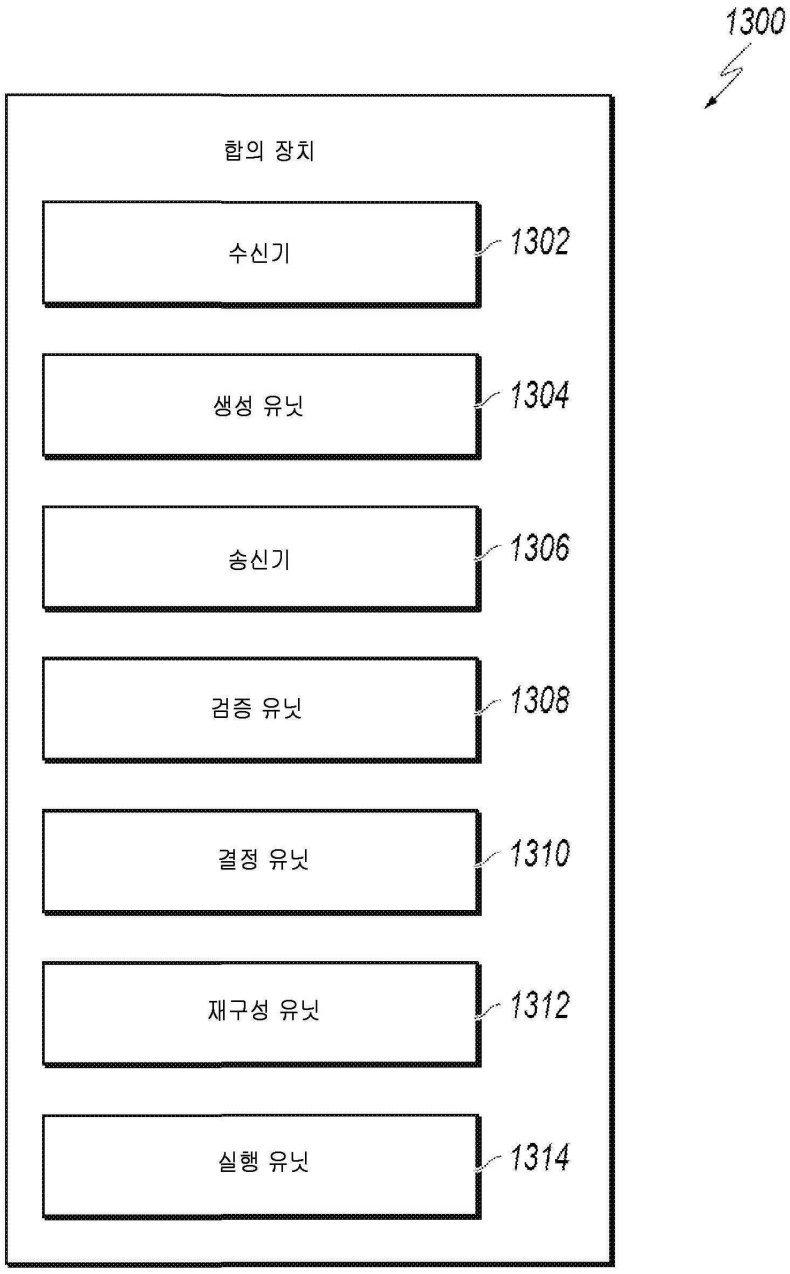
도면11



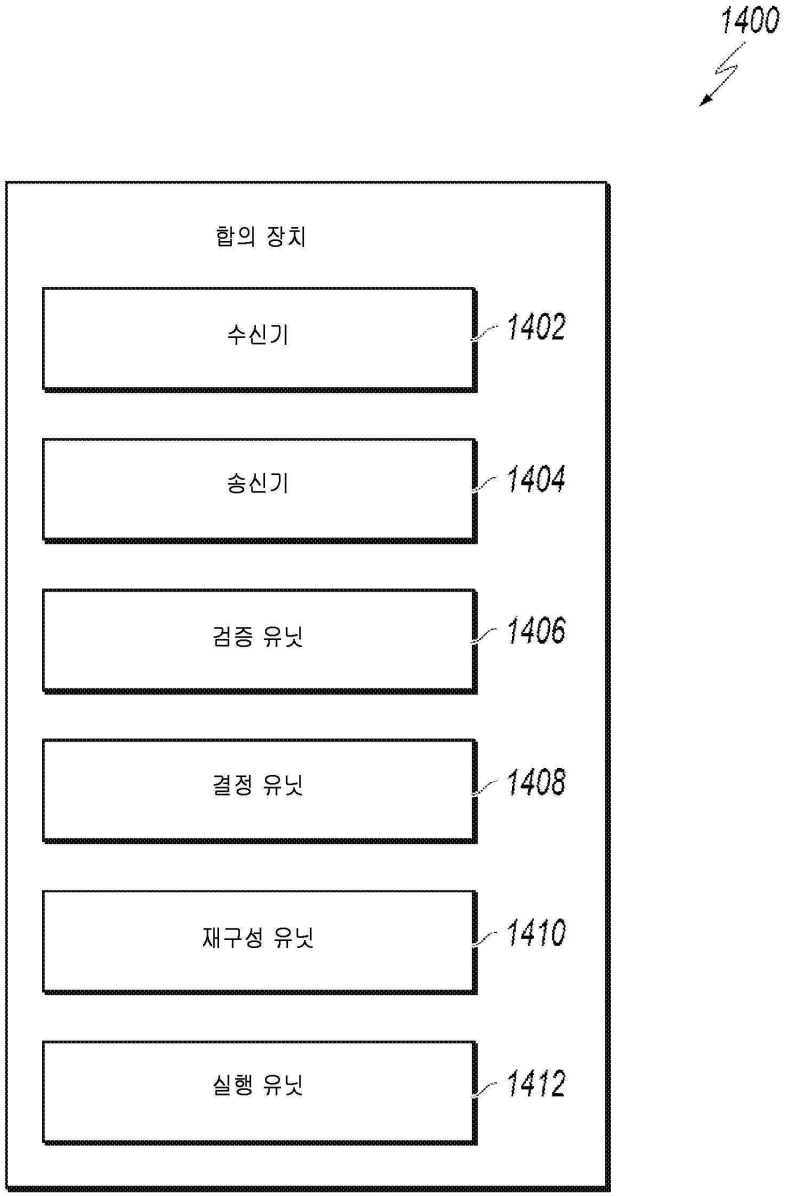
도면12

Query_state <j, seq>
Reply_state <j, last_seq>
Fetch_echo <j, last_seq+1>

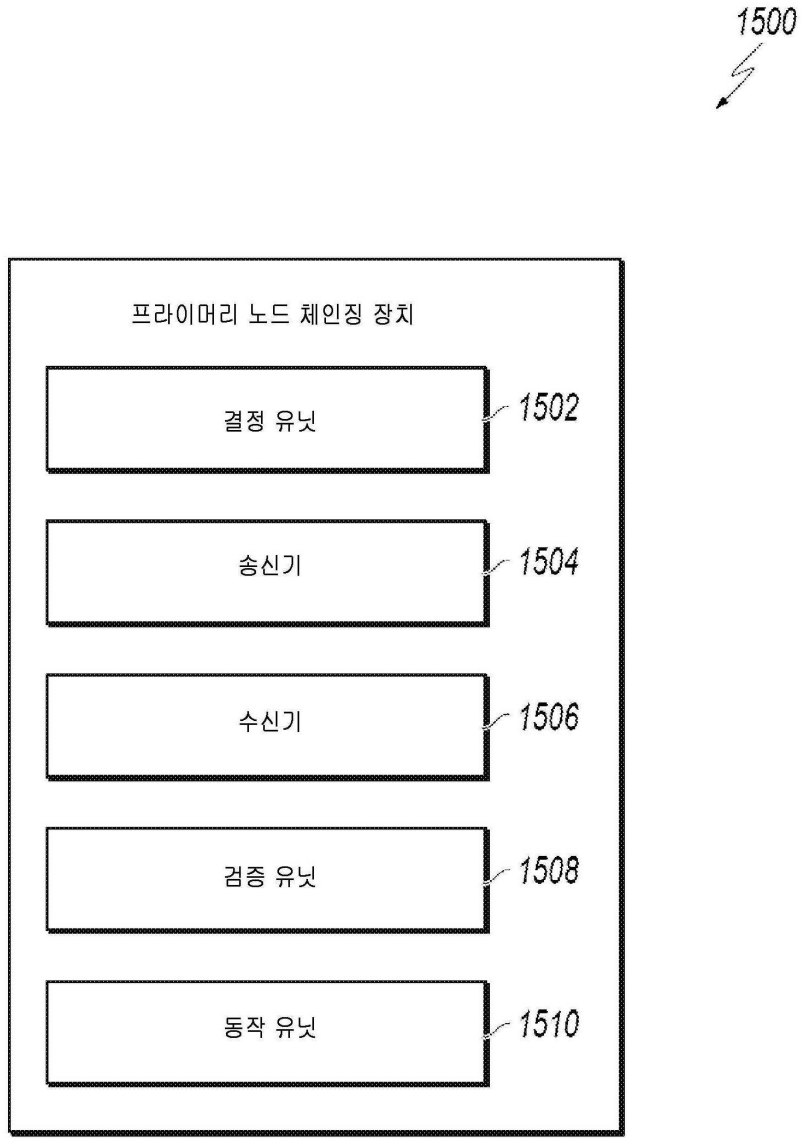
도면13


1300
합의 장치
수신기
1302
생성 유닛
1304
송신기
1306
검증 유닛
1308
결정 유닛
1310
재구성 유닛
1312
실행 유닛
1314

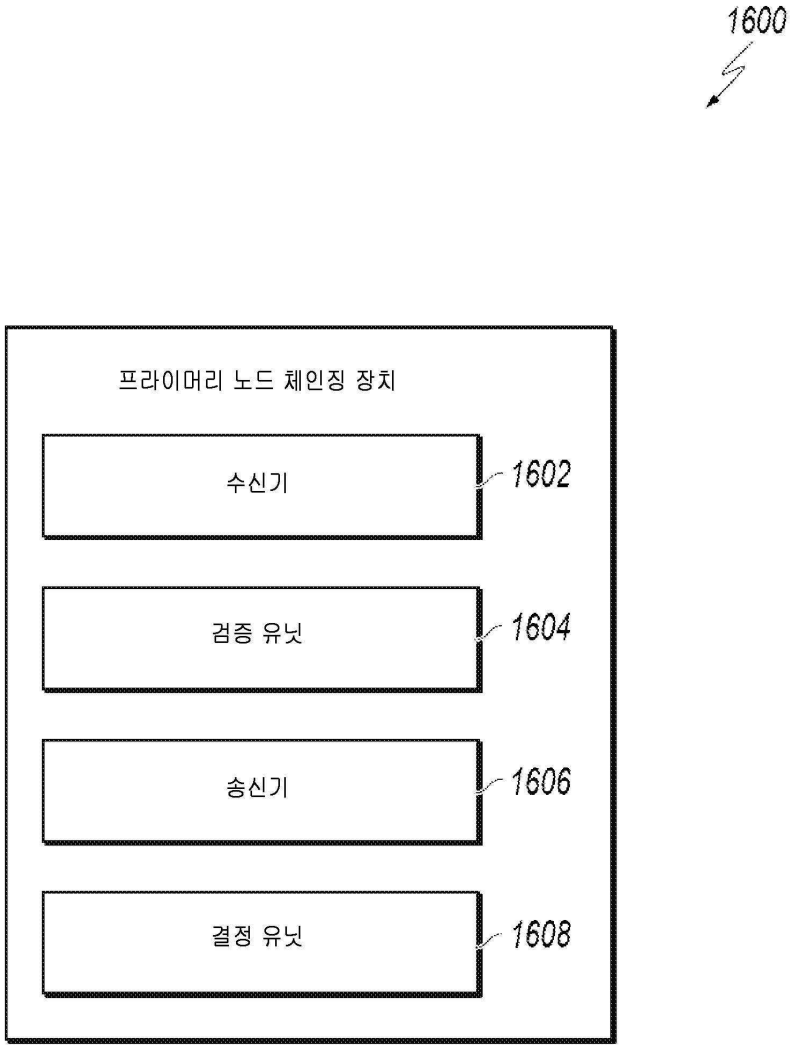
도면14


1400
합의 장치
수신기
1402
송신기
1404
검증 유닛
1406
결정 유닛
1408
재구성 유닛
1410
실행 유닛
1412

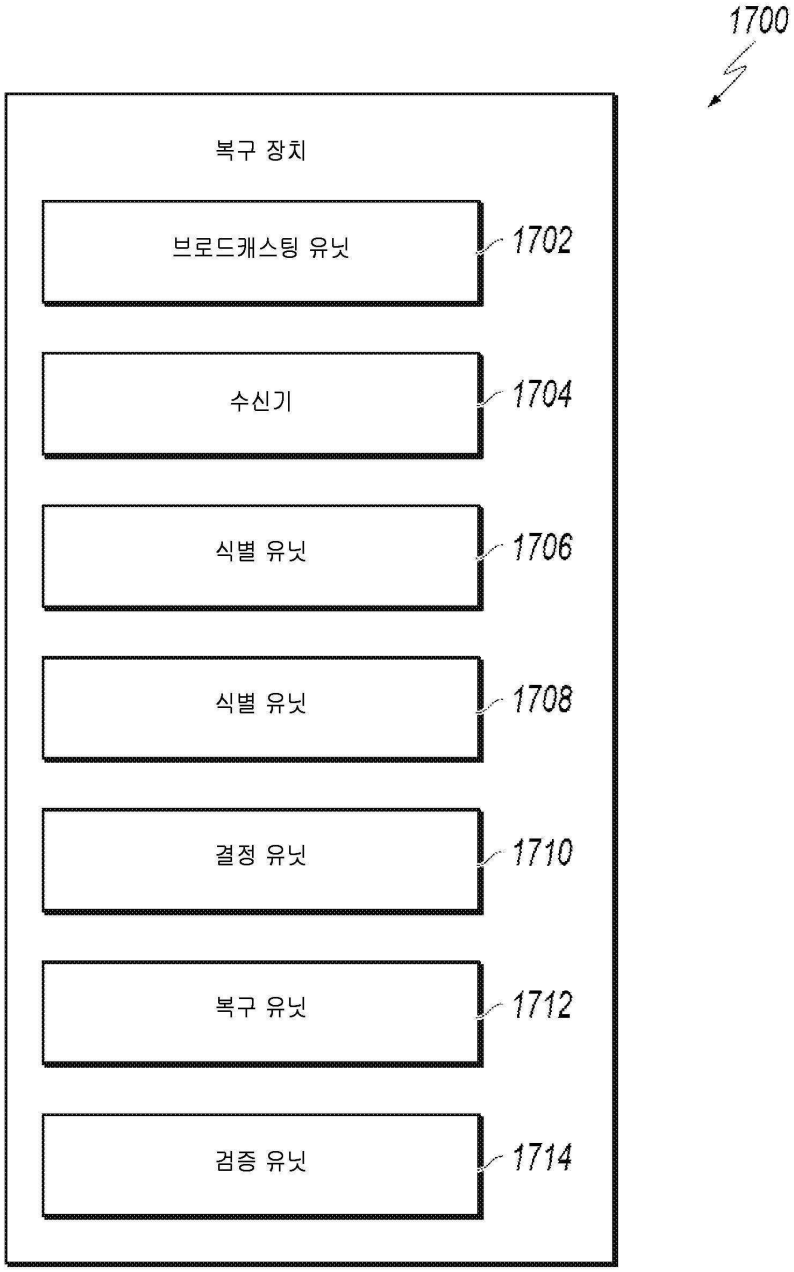
도면15


1500
프라이머리 노드 체인징 장치
결정 유닛
1502
송신기
1504
수신기
1506
검증 유닛
1508
동작 유닛
1510

도면16



도면17



【심사관 직권보정사항】

【직권보정 1】

【보정항목】 청구범위

【보정세부항목】 청구항 12

【변경전】

제11항에 있어서,

상기 현재의 에포크에서의 상기 합의 프로세스의 상기 세 개의 페이즈의 각각과 연관되는 상기 백업 노드의 각각의 가중치를 결정하는 동작은, 상기 합의 프로세스의 상기 제1 페이즈에 대한 상기 백업 노드의 가중치를 제1 값이 되는 것으로 결정하는 것을 포함하는 것인, 비일시적 컴퓨터 판독 가능 매체.

【변경후】

제11항에 있어서,

상기 현재의 에포크에서의 상기 합의 프로세스의 상기 세 개의 페이즈의 각각과 연관되는 상기 백업 노드의 각각의 가중치를 결정하는 동작은, 상기 합의 프로세스의 제1 페이즈에 대한 상기 백업 노드의 가중치

를 제1 값이 되는 것으로 결정하는 것을 포함하는 것인, 비일시적 컴퓨터 판독 가능 매체.