(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2018-0069253
(43) 공개일자 2018년06월25일

(51) 국제특허분류(Int. Cl.)
G06F 17/30 (2006.01)
(52) CPC특허분류
G06F 17/30194 (2013.01)
(21) 출원번호 10-2016-0171272
(22) 출원일자 2016년12월15일
심사청구일자 2016년12월15일

(71) 출원인
조선대학교산학협력단
광주광역시 동구 필문대로 309 (서석동)
전남대학교산학협력단
광주광역시 북구 용봉로 77 (용봉동)
(72) 발명자
김영식
광주광역시 서구 풍암2로 62 금호2차아파트 201동 1005호
박호성
광주광역시 북구 서하로 416, 대주3차아파트 102동 1107호
(74) 대리인
김효성

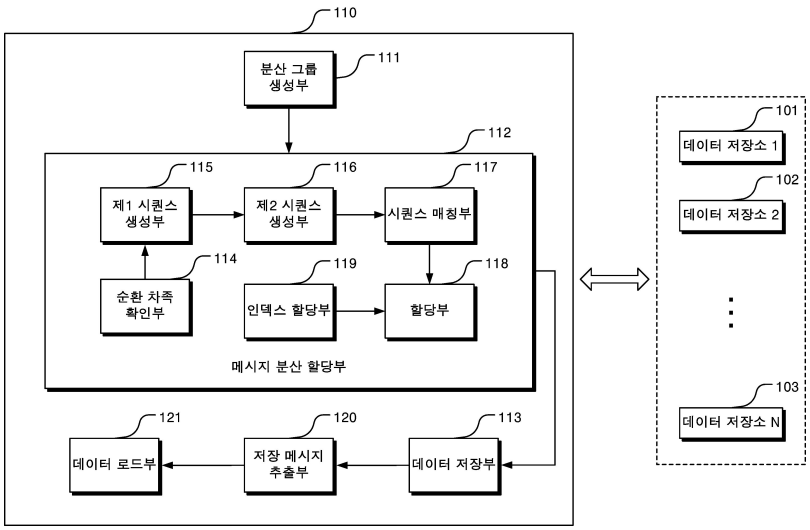
전체 청구항 수 : 총 16 항

(54) 발명의 명칭 **순환 차족을 활용한 데이터 분산 저장 장치 및 방법**

(57) 요 약

순환 차족(Cyclic Difference Family)을 활용한 데이터 분산 저장 장치 및 방법이 개시된다. 본 발명은 다수의 메시지들을 여러 데이터 저장소들에 분산 저장하는 방법에 있어서, 최소한의 데이터 저장소들만을 활용해서 상기 다수의 메시지들을 분산 저장한 후 상기 데이터 저장소들 중 최소한의 데이터 저장소들만을 선택하여 액세스하더라도 상기 다수의 메시지들이 로드될 수 있도록 하여 상기 다수의 메시지들에 대한 데이터 손실에 대비함과 동시에 데이터 저장소들로부터 빠른 데이터 로드가 가능하도록 하는 기법을 제공하고자 한다.

대 표 도 - 도1



이 발명을 지원한 국가연구개발사업
과제고유번호 1711043033
부처명 미래창조과학부
연구관리전문기관 미래창조과학부
연구사업명 정보보호핵심원천기술개발
연구과제명 (창조씨앗 1단계) 자율 주행차 침입 감내 시스템 기반 기술 개발
기 여 율 1/2
주관기관 조선대학교 산학협력단
연구기간 2016.09.01 ~ 2016.12.31
이 발명을 지원한 국가연구개발사업
과제고유번호 1345241259
부처명 교육부
연구관리전문기관 한국연구재단
연구사업명 이공학개인기초연구지원
연구과제명 빅데이터의 신뢰성 향상 및 효율적 관리를 위한 분산 저장 부호 기법
기 여 율 1/2
주관기관 전남대학교
연구기간 2015.11.01 ~ 2018.10.31

명 세 서

청구범위

청구항 1

N - N은 N=tv의 조건을 만족하는 자연수이고, v는 v=k(k-1)t+1의 조건을 만족하는 자연수이며, k는 2이상의 자연수이고, t는 1이상의 자연수임 - 개의 메시지들에 대해 v개의 데이터 저장소들로의 분산 저장 명령이 인가되면, 상기 v개의 데이터 저장소들에 대응하는 v개의 메시지 분산 그룹들을 생성하는 분산 그룹 생성부;

상기 v개의 메시지 분산 그룹들 각각에 대해 상기 N개의 메시지들을 tk - tk는 t와 k의 곱을 의미함 - 개씩 분산 할당 - 상기 v개의 메시지 분산 그룹들에 분산 할당되는 메시지들은 서로 중복될 수 있음 - 하는 메시지 분산 할당부; 및

상기 v개의 메시지 분산 그룹들 각각에 할당되어 있는 tk개의 메시지들을 상기 v개의 메시지 분산 그룹들 별로, 상기 v개의 데이터 저장소들 중 각 메시지 분산 그룹에 대응되는 데이터 저장소에 분리해서 저장하는 데이터 저장부

를 포함하는 순환 차족을 활용한 데이터 분산 저장 장치.

청구항 2

제1항에 있어서,

상기 메시지 분산 할당부는

"0"부터 "v-1"까지의 정수들을 원소로 포함하고 있는 집합인 Z_v - 상기 Z_v에 포함되어 있는 원소들의 개수는 v개임 - 에 대한 k개의 원소를 갖는 복수의 부분 집합들 중, 상기 복수의 부분 집합들 각각에 포함되어 있는 k개의 원소들 중에서 서로 다른 두 원소 간의 모든 차들 각각이 상기 Z_v에 포함되어 있는 원소들에서 "0"을 제외한 서로 다른 1개의 원소와만 일치하는 것으로 확인되는 t개의 부분 집합들 - 상기 t개의 부분 집합들의 모임은 (v, k, 1)의 파라미터를 갖는 순환 차족(Cyclic Difference Family)으로 정의됨 - 을 확인하는 순환 차족 확인부;

상기 t개의 부분 집합들 각각에 대해서, 상기 t개의 부분 집합들 각각에 포함되어 있는 k개의 원소들에 기초하여 상기 Z_v에 포함되어 있는 "0"부터 "v-1"까지의 정수들에 대해 "0"부터 "v-1" 순으로 "1"과 "0"의 코드 값을 할당함으로써, v개의 "1"과 "0"의 코드 값으로 구성된 t개의 부분 시퀀스들을 생성하고, 상기 t개의 부분 시퀀스들을 조합하여 총 N개의 "1"과 "0"의 코드 값으로 구성된 제1 시퀀스 - N=tv이며, 상기 t개의 부분 집합들 각각에 대해서 v개씩 "1"과 "0"의 코드 값을 할당함으로써 생성한 상기 t개의 부분 시퀀스들을 조합하여 상기 제1 시퀀스를 생성하였으므로, 상기 제1 시퀀스를 구성하는 "1"과 "0"의 코드 값은 총 N개가 됨 - 를 생성하는 제1 시퀀스 생성부;

상기 v개의 "1"과 "0"의 코드 값으로 구성된 상기 t개의 부분 시퀀스들 각각에 대해 선정된(predetermined) 방향으로 순환시프트 연산을 v-1회 수행하면서, 각 순환시프트 연산이 수행될 때마다 순환시프트 연산이 완료된 상기 t개의 부분 시퀀스들을 조합하여 제2 시퀀스를 생성함으로써, 총 v-1개의 제2 시퀀스들을 생성하는 제2 시퀀스 생성부;

상기 v개의 메시지 분산 그룹들 각각에 대해 상기 제1 시퀀스와 상기 v-1개의 제2 시퀀스들을 하나씩 매칭시키는 시퀀스 매칭부; 및

상기 v개의 메시지 분산 그룹들 각각에 대해, 상기 v개의 메시지 분산 그룹들 각각에 매칭되어 있는 각 시퀀스를 구성하는 N개의 코드 값에 기초하여 상기 N개의 메시지들을 tk개씩 분산 할당하는 할당부

를 포함하는 순환 차족을 활용한 데이터 분산 저장 장치.

청구항 3

제2항에 있어서,

상기 제1 시퀀스 생성부는

상기 t개의 부분 집합들 각각에 대해서, 상기 Z_v에 포함되어 있는 "0"부터 "v-1"까지의 정수들 중 상기 t개의 부분 집합들 각각에 포함되어 있는 k개의 원소들과 일치하는 정수들에 대해 "1"의 코드 값을 할당하고, 상기 t개의 부분 집합들 각각에 포함되어 있는 k개의 원소들과 일치하지 않는 나머지 정수들에 대해 "0"의 코드 값을 할당함으로써, 상기 v개의 "1"과 "0"의 코드 값으로 구성된 상기 t개의 부분 시퀀스들을 생성하는 순환 차족을 활용한 데이터 분산 저장 장치.

청구항 4

제3항에 있어서,

상기 메시지 분산 할당부는

상기 N개의 메시지들 각각에 대해 순번 정보와 연관된 인덱스 값을 할당하는 인덱스 할당부

를 더 포함하고,

상기 할당부는

상기 v개의 메시지 분산 그룹들 각각에 대해, 상기 v개의 메시지 분산 그룹들 각각에 매칭되어 있는 각 시퀀스를 구성하는 N개의 코드 값의 배열 순서와 상기 N개의 메시지들 각각에 할당되어 있는 인덱스 값을 비교한 후 상기 N개의 메시지들에 할당되어 있는 인덱스 값과 일치하는 배열 순서의 코드 값에 기초하여 상기 N개의 메시지들을 tk개씩 분산 할당하는 순환 차족을 활용한 데이터 분산 저장 장치.

청구항 5

제4항에 있어서,

상기 할당부는

상기 v개의 메시지 분산 그룹들 각각에 대해, 상기 v개의 메시지 분산 그룹들 각각에 매칭되어 있는 각 시퀀스를 구성하는 N개의 코드 값 중 tk개씩 존재하는 "1"의 코드 값을 확인한 후, 상기 v개의 메시지 분산 그룹들 각각에 대해, 상기 N개의 메시지들 중 상기 "1"의 코드 값에 대한 배열 순서와 일치하는 인덱스 값이 할당되어 있는 tk개의 메시지들을 분산 할당하는 순환 차족을 활용한 데이터 분산 저장 장치.

청구항 6

제5항에 있어서,

상기 v개의 데이터 저장소들에 상기 N개의 메시지들이 tk개씩 분산 저장된 이후, 상기 N개의 메시지들에 대한 로드(load) 명령이 인가되면, 상기 로드 명령에 기초하여 상기 v개의 데이터 저장소들 중 tk개의 데이터 저장소들을 랜덤하게 선택하고, 상기 tk개의 데이터 저장소들 각각으로부터 상기 tk개의 데이터 저장소들 각각에 저장되어 있는 tk개의 저장 메시지들을 추출하는 저장 메시지 추출부; 및

상기 tk개의 데이터 저장소들 각각으로부터 tk개씩 추출된 저장 메시지들을 상기 N개의 메시지들에 대한 로드 명령에 따른 데이터로 로드하는 데이터 로드부

를 더 포함하는 순환 차족을 활용한 데이터 분산 저장 장치.

청구항 7

제6항에 있어서,

상기 N개의 메시지들은 일부 메시지의 손실이 발생할 경우, 나머지 메시지들을 기초로 손실 메시지들의 복구가 가능한 최대 거리 분리 가능(Maximum Distance Separable: MDS) 코드이고,

상기 데이터 로드부는

상기 tk개의 데이터 저장소들 각각으로부터 tk개씩 저장 메시지들이 추출되면, 상기 추출된 저장 메시지들에 할당되어 있는 인덱스 값과 상기 N개의 메시지들에 할당된 인덱스 값을 비교하여 상기 N개의 메시지들 중 상기 추출된 저장 메시지들에 포함되어 있지 않은 잔여 메시지가 존재하는 것으로 확인되면, 상기 추출된 저장 메시지

들을 기초로 상기 잔여 메시지를 복구하여 상기 추출된 저장 메시지들과 상기 복구된 잔여 메시지를 상기 N개의 메시지들에 대한 로드 명령에 따른 데이터로 로드하는 순환 차족을 활용한 데이터 분산 저장 장치.

청구항 8

N - N은 N=tv의 조건을 만족하는 자연수이고, v는 v=k(k-1)t+1의 조건을 만족하는 자연수이며, k는 2이상의 자연수이고, t는 1이상의 자연수임 - 개의 메시지들에 대해 v개의 데이터 저장소들로의 분산 저장 명령이 인가되면, 상기 v개의 데이터 저장소들에 대응하는 v개의 메시지 분산 그룹들을 생성하는 단계;

상기 v개의 메시지 분산 그룹들 각각에 대해 상기 N개의 메시지들을 tk - tk는 t와 k의 곱을 의미함 - 개씩 분산 할당 - 상기 v개의 메시지 분산 그룹들에 분산 할당되는 메시지들은 서로 중복될 수 있음 - 하는 단계; 및

상기 v개의 메시지 분산 그룹들 각각에 할당되어 있는 tk개의 메시지들을 상기 v개의 메시지 분산 그룹들 별로, 상기 v개의 데이터 저장소들 중 각 메시지 분산 그룹에 대응되는 데이터 저장소에 분리해서 저장하는 단계

를 포함하는 순환 차족을 활용한 데이터 분산 저장 방법.

청구항 9

제8항에 있어서,

상기 v개의 메시지 분산 그룹들 각각에 대해 상기 N개의 메시지들을 tk개씩 분산 할당하는 단계는

"0"부터 "v-1"까지의 정수들을 원소로 포함하고 있는 집합인 Z_v - 상기 Z_v에 포함되어 있는 원소들의 개수는 v개임 - 에 대한 k개의 원소를 갖는 복수의 부분 집합들 중, 상기 복수의 부분 집합들 각각에 포함되어 있는 k개의 원소들 중에서 서로 다른 두 원소 간의 모든 차들 각각이 상기 Z_v에 포함되어 있는 원소들에서 "0"을 제외한 서로 다른 1개의 원소와만 일치하는 것으로 확인되는 t개의 부분 집합들 - 상기 t개의 부분 집합들의 모임은 (v, k, 1)의 파라미터를 갖는 순환 차족(Cyclic Difference Family)으로 정의됨 - 을 확인하는 단계;

상기 t개의 부분 집합들 각각에 대해서, 상기 t개의 부분 집합들 각각에 포함되어 있는 k개의 원소들에 기초하여 상기 Z_v에 포함되어 있는 "0"부터 "v-1"까지의 정수들에 대해 "0"부터 "v-1" 순으로 "1"과 "0"의 코드 값을 할당함으로써, v개의 "1"과 "0"의 코드 값으로 구성된 t개의 부분 시퀀스들을 생성하고, 상기 t개의 부분 시퀀스들을 조합하여 총 N개의 "1"과 "0"의 코드 값으로 구성된 제1 시퀀스 - N=tv이며, 상기 t개의 부분 집합들 각각에 대해서 v개씩 "1"과 "0"의 코드 값을 할당함으로써 생성한 상기 t개의 부분 시퀀스들을 조합하여 상기 제1 시퀀스를 생성하였으므로, 상기 제1 시퀀스를 구성하는 "1"과 "0"의 코드 값은 총 N개가 됨 - 를 생성하는 단계;

상기 v개의 "1"과 "0"의 코드 값으로 구성된 상기 t개의 부분 시퀀스들 각각에 대해 선정된(predetermined) 방향으로 순환시프트 연산을 v-1회 수행하면서, 각 순환시프트 연산이 수행될 때마다 순환시프트 연산이 완료된 상기 t개의 부분 시퀀스들을 조합하여 제2 시퀀스를 생성함으로써, 총 v-1개의 제2 시퀀스들을 생성하는 단계;

상기 v개의 메시지 분산 그룹들 각각에 대해 상기 제1 시퀀스와 상기 v-1개의 제2 시퀀스들을 하나씩 매칭시키는 단계; 및

상기 v개의 메시지 분산 그룹들 각각에 대해, 상기 v개의 메시지 분산 그룹들 각각에 매칭되어 있는 각 시퀀스를 구성하는 N개의 코드 값에 기초하여 상기 N개의 메시지들을 tk개씩 분산 할당하는 단계

를 포함하는 순환 차족을 활용한 데이터 분산 저장 방법.

청구항 10

제9항에 있어서,

상기 제1 시퀀스를 생성하는 단계는

상기 t개의 부분 집합들 각각에 대해서, 상기 Z_v에 포함되어 있는 "0"부터 "v-1"까지의 정수들 중 상기 t개의 부분 집합들 각각에 포함되어 있는 k개의 원소들과 일치하는 정수들에 대해 "1"의 코드 값을 할당하고, 상기 t개의 부분 집합들 각각에 포함되어 있는 k개의 원소들과 일치하지 않는 나머지 정수들에 대해 "0"의 코드 값을 할당함으로써, 상기 v개의 "1"과 "0"의 코드 값으로 구성된 상기 t개의 부분 시퀀스들을 생성하는 순환 차족을

활용한 데이터 분산 저장 방법.

청구항 11

제10항에 있어서,

상기 v개의 메시지 분산 그룹들 각각에 대해 상기 N개의 메시지들을 tk개씩 분산 할당하는 단계는

상기 N개의 메시지들 각각에 대해 순번 정보와 연관된 인덱스 값을 할당하는 단계

를 더 포함하고,

상기 N개의 코드 값에 기초하여 상기 N개의 메시지들을 tk개씩 분산 할당하는 단계는

상기 v개의 메시지 분산 그룹들 각각에 대해, 상기 v개의 메시지 분산 그룹들 각각에 매칭되어 있는 각 시퀀스를 구성하는 N개의 코드 값의 배열 순서와 상기 N개의 메시지들 각각에 할당되어 있는 인덱스 값을 비교한 후 상기 N개의 메시지들에 할당되어 있는 인덱스 값과 일치하는 배열 순서의 코드 값에 기초하여 상기 N개의 메시지들을 tk개씩 분산 할당하는 순환 차족을 활용한 데이터 분산 저장 방법.

청구항 12

제11항에 있어서,

상기 N개의 코드 값에 기초하여 상기 N개의 메시지들을 tk개씩 분산 할당하는 단계는

상기 v개의 메시지 분산 그룹들 각각에 대해, 상기 v개의 메시지 분산 그룹들 각각에 매칭되어 있는 각 시퀀스를 구성하는 N개의 코드 값 중 tk개씩 존재하는 "1"의 코드 값을 확인한 후, 상기 v개의 메시지 분산 그룹들 각각에 대해, 상기 N개의 메시지들 중 상기 "1"의 코드 값에 대한 배열 순서와 일치하는 인덱스 값이 할당되어 있는 tk개의 메시지들을 분산 할당하는 순환 차족을 활용한 데이터 분산 저장 방법.

청구항 13

제12항에 있어서,

상기 v개의 데이터 저장소들에 상기 N개의 메시지들이 tk개씩 분산 저장된 이후, 상기 N개의 메시지들에 대한 로드(load) 명령이 인가되면, 상기 로드 명령에 기초하여 상기 v개의 데이터 저장소들 중 tk개의 데이터 저장소들을 랜덤하게 선택하고, 상기 tk개의 데이터 저장소들 각각으로부터 상기 tk개의 데이터 저장소들 각각에 저장되어 있는 tk개의 저장 메시지들을 추출하는 단계; 및

상기 tk개의 데이터 저장소들 각각으로부터 tk개씩 추출된 저장 메시지들을 상기 N개의 메시지들에 대한 로드 명령에 따른 데이터로 로드하는 단계

를 더 포함하는 순환 차족을 활용한 데이터 분산 저장 방법.

청구항 14

제13항에 있어서,

상기 N개의 메시지들은 일부 메시지의 손실이 발생할 경우, 나머지 메시지들을 기초로 손실 메시지들의 복구가 가능한 최대 거리 분리 가능(Maximum Distance Separable: MDS) 코드이고,

상기 로드하는 단계는

상기 tk개의 데이터 저장소들 각각으로부터 tk개씩 저장 메시지들이 추출되면, 상기 추출된 저장 메시지들에 할당되어 있는 인덱스 값과 상기 N개의 메시지들에 할당된 인덱스 값을 비교하여 상기 N개의 메시지들 중 상기 추출된 저장 메시지들에 포함되어 있지 않은 잔여 메시지가 존재하는 것으로 확인되면, 상기 추출된 저장 메시지들을 기초로 상기 잔여 메시지를 복구하여 상기 추출된 저장 메시지들과 상기 복구된 잔여 메시지를 상기 N개의 메시지들에 대한 로드 명령에 따른 데이터로 로드하는 순환 차족을 활용한 데이터 분산 저장 방법.

청구항 15

제8항 내지 제14항 중 어느 한 항의 방법을 수행하는 프로그램을 기록한 컴퓨터 판독 가능 기록 매체.

청구항 16

제8항 내지 제14항 중 어느 한 항의 방법을 컴퓨터와의 결합을 통해 실행시키기 위한 저장매체에 저장된 컴퓨터 프로그램.

발명의 설명

기 술 분 야

[0001] 본 발명은 데이터의 손실이 발생하였을 때, 데이터의 복구가 용이하도록 다수의 데이터 저장소들에 다수의 메시지들을 분산해서 저장하는 기법에 대한 기술과 관련된다.

배 경 기 술

[0002] 최근, 인터넷의 보급이 활발하게 이루어짐에 따라, 다양한 인터넷 서비스 제공 업체들이 등장하고 있다. 이러한 인터넷 서비스 제공 업체들은 이메일 서비스를 제공하거나, 블로그, 카페, 클라우드 데이터 저장 서비스 등과 같은 다양한 서비스들을 제공하고 있다.

[0003] 특히, 최근에는 소셜 네트워크 서비스의 인기가 급증함에 따라, 소셜 네트워크 서비스를 제공하는 인터넷 서비스 제공 업체들도 매우 많이 증가하고 있다.

[0004] 이러한 인터넷 서비스 제공 업체들은 이메일 서비스, 블로그, 카페, 클라우드 서비스, 소셜 네트워크 서비스 등을 제공하기 위해 다수의 클라이언트 단말들로부터 업로드되는 대용량의 데이터들을 자체의 데이터 저장소에 저장을 해야하는데, 이러한 데이터들이 데이터 저장소 내에서 손실되게 되면, 고객들에게 정상적인 서비스를 제공하지 못하기 때문에 데이터의 손실을 방지하기 위한 다양한 노력을 하고 있다.

[0005] 그러한 노력의 일환 중 하나로 데이터 저장소에 저장되어야 하는 사용자 메시지를 여러 데이터 저장소들에 백업을 해두는 방식이 도입되고 있다. 이러한 데이터 백업 방식은 원본 메시지가 손실되더라도 백업을 해둔 메시지를 이용하여 상기 원본 메시지를 대체할 수 있다는 점에서 데이터 손실에 대처하기 위한 가장 이상적인 방법 중 하나이다.

[0006] 이렇게, 하나의 원본 메시지를 복수의 데이터 저장소들에 여러 번 저장을 해두게 되면, 상기 원본 메시지에 대한 데이터 로드 명령이 인가될 때, 상기 복수의 데이터 저장소들 중 가장 동작 속도가 빠른 임의의 데이터 저장소로부터 상기 원본 메시지에 대응하는 메시지를 바로 로드하면 되기 때문에, 하나의 데이터 저장소에 원본 메시지를 저장하는 방식보다 데이터 로드에 있어서도 빠른 속도를 제공할 수 있다.

[0007] 하지만, 기존의 데이터 백업 방식은 단순히 원본 메시지에 대한 복사 메시지를 복수의 데이터 저장소들에 분산해서 저장하는 방식만을 도입하고 있어서, 다수의 메시지들을 데이터 저장소에 저장해야 하는 인터넷 서비스 제공 업체들의 입장에서는 상기 다수의 메시지들 각각에 대한 복사 메시지들을 생성해서 각 데이터 저장소들에 분산 저장해야 하기 때문에 많은 데이터 저장 용량이 필요하다는 문제가 발생할 수 있다.

[0008] 따라서, 다수의 메시지들을 여러 데이터 저장소들에 분산 저장하는 방법에 있어서, 최소한의 데이터 저장소들만을 활용해서 상기 다수의 메시지들을 분산 저장한 후 상기 데이터 저장소들 중 최소한의 데이터 저장소들만을 선택하여 액세스하더라도 상기 다수의 메시지들이 모두 로드될 수 있도록 하여 상기 다수의 메시지들에 대한 데이터 손실에 대비함과 동시에 데이터 저장소들로부터 빠른 데이터 로드가 가능하도록 하는 기법에 대한 연구가 필요하다.

발명의 내용

해결하려는 과제

[0009] 본 발명은 다수의 메시지들을 여러 데이터 저장소들에 분산 저장하는 방법에 있어서, 최소한의 데이터 저장소들만을 활용해서 상기 다수의 메시지들을 분산 저장한 후 상기 데이터 저장소들 중 최소한의 데이터 저장소들만을 선택하여 액세스하더라도 상기 다수의 메시지들이 로드될 수 있도록 하여 상기 다수의 메시지들에 대한 데이터 손실에 대비함과 동시에 데이터 저장소들로부터 빠른 데이터 로드가 가능하도록 하는 기법을 제공하고자 한다.

과제의 해결 수단

[0010] 본 발명의 일실시예에 따른 순환 차족을 활용한 데이터 분산 저장 장치는 N - N은 N=tv의 조건을 만족하는 자연수이고, v는 v=k(k-1)t+1의 조건을 만족하는 자연수이며, k는 2이상의 자연수이고, t는 1이상의 자연수임 - 개의 메시지들에 대해 v개의 데이터 저장소들로의 분산 저장 명령이 인가되면, 상기 v개의 데이터 저장소들에 대응하는 v개의 메시지 분산 그룹들을 생성하는 분산 그룹 생성부, 상기 v개의 메시지 분산 그룹들 각각에 대해 상기 N개의 메시지들을 tk - tk는 t와 k의 곱을 의미함 - 개씩 분산 할당 - 상기 v개의 메시지 분산 그룹들에 분산 할당되는 메시지들은 서로 중복될 수 있음 - 하는 메시지 분산 할당부 및 상기 v개의 메시지 분산 그룹들 각각에 할당되어 있는 tk개의 메시지들을 상기 v개의 메시지 분산 그룹들 별로, 상기 v개의 데이터 저장소들 중 각 메시지 분산 그룹에 대응되는 데이터 저장소에 분리해서 저장하는 데이터 저장부를 포함한다.

[0011] 또한, 본 발명의 일실시예에 따른 순환 차족을 활용한 데이터 분산 저장 방법은 N - N은 N=tv의 조건을 만족하는 자연수이고, v는 v=k(k-1)t+1의 조건을 만족하는 자연수이며, k는 2이상의 자연수이고, t는 1이상의 자연수임 - 개의 메시지들에 대해 v개의 데이터 저장소들로의 분산 저장 명령이 인가되면, 상기 v개의 데이터 저장소들에 대응하는 v개의 메시지 분산 그룹들을 생성하는 단계, 상기 v개의 메시지 분산 그룹들 각각에 대해 상기 N개의 메시지들을 tk - tk는 t와 k의 곱을 의미함 - 개씩 분산 할당 - 상기 v개의 메시지 분산 그룹들에 분산 할당되는 메시지들은 서로 중복될 수 있음 - 하는 단계, 상기 v개의 메시지 분산 그룹들 각각에 할당되어 있는 tk개의 메시지들을 상기 v개의 메시지 분산 그룹들 별로, 상기 v개의 데이터 저장소들 중 각 메시지 분산 그룹에 대응되는 데이터 저장소에 분리해서 저장하는 단계를 포함한다.

발명의 효과

[0012] 본 발명은 다수의 메시지들을 여러 데이터 저장소들에 분산 저장하는 방법에 있어서, 최소한의 데이터 저장소들만을 활용해서 상기 다수의 메시지들을 분산 저장한 후 상기 데이터 저장소들 중 최소한의 데이터 저장소들만을 선택하여 액세스하더라도 상기 다수의 메시지들이 로드될 수 있도록 하여 상기 다수의 메시지들에 대한 데이터 손실에 대비함과 동시에 데이터 저장소들로부터 빠른 데이터 로드가 가능하도록 하는 기법을 제공할 수 있다.

도면의 간단한 설명

[0013] 도 1은 본 발명의 일실시예에 따른 순환 차족(Cyclic Difference Family)을 활용한 데이터 분산 저장 장치의 구조를 도시한 도면이다.

도 2는 본 발명의 일실시예에 따른 순환 차족을 활용한 데이터 분산 저장 장치의 동작을 설명하기 위한 도면이다.

도 3은 본 발명의 일실시예에 따른 순환 차족을 활용한 데이터 분산 저장 방법을 도시한 순서도이다.

발명을 실시하기 위한 구체적인 내용

[0014] 이하에서는 본 발명에 따른 실시예들을 첨부된 도면을 참조하여 상세하게 설명하기로 한다. 이러한 설명은 본 발명을 특정한 실시 형태에 대해 한정하려는 것이 아니며, 본 발명의 사상 및 기술 범위에 포함되는 모든 변경, 균등물 내지 대체물을 포함하는 것으로 이해되어야 한다. 각 도면을 설명하면서 유사한 참조부호를 유사한 구성요소에 대해 사용하였으며, 다르게 정의되지 않는 한, 기술적이거나 과학적인 용어를 포함해서 본 명세서 상에서 사용되는 모든 용어들은 본 발명이 속하는 기술분야에서 통상의 지식을 가진 사람에 의해 일반적으로 이해되는 것과 동일한 의미를 가지고 있다.

[0015] 도 1은 본 발명의 일실시예에 따른 순환 차족(Cyclic Difference Family)을 활용한 데이터 분산 저장 장치의 구조를 도시한 도면이다.

[0016] 도 1을 참조하면, 본 발명의 일실시예에 따른 순환 차족을 활용한 데이터 분산 저장 장치(110)는 분산 그룹 생성부(111), 메시지 분산 할당부(112) 및 데이터 저장부(113)를 포함한다.

[0017] 분산 그룹 생성부(111)는 N개의 메시지들에 대해 v개의 데이터 저장소들(101, 102, 103)로의 분산 저장 명령이 인가되면, v개의 데이터 저장소들(101, 102, 103)에 대응하는 v개의 메시지 분산 그룹들을 생성한다.

[0018] 여기서, N은 N=tv의 조건을 만족하는 자연수이고, v는 v=k(k-1)t+1의 조건을 만족하는 자연수이며, k는 2이상의 자연수이고, t는 1이상의 자연수이다.

[0019] 그리고, 메시지란 v개의 데이터 저장소들(101, 102, 103)에 저장될 일종의 데이터 심볼(symbol)을 의미하는 것

으로 상기 N개의 메시지들은 각각 서로 다른 데이터 심볼이다.

[0020] 메시지 분산 할당부(112)는 상기 v개의 메시지 분산 그룹들 각각에 대해 상기 N개의 메시지들을 tk(tk는 t와 k의 곱을 의미함)개씩 분산 할당한다.

[0021] 이때, 상기 v개의 메시지 분산 그룹들에 분산 할당되는 메시지들은 서로 중복될 수 있다.

[0022] 데이터 저장부(113)는 상기 v개의 메시지 분산 그룹들 각각에 할당되어 있는 tk개의 메시지들을 상기 v개의 메시지 분산 그룹들 별로, v개의 데이터 저장소들(101, 102, 103) 중 각 메시지 분산 그룹에 대응되는 데이터 저장소에 분리해서 저장한다.

[0023] 이때, 본 발명의 일실시예에 따르면, 메시지 분산 할당부(112)는 순환 차족 확인부(114), 제1 시퀀스 생성부(115), 제2 시퀀스 생성부(116), 시퀀스 매칭부(117) 및 할당부(118)를 포함할 수 있다.

[0024] 순환 차족 확인부(114)는 "0"부터 "v-1"까지의 정수들을 원소로 포함하고 있는 집합인 Z_v(상기 Z_v에 포함되어 있는 원소들의 개수는 v개임)에 대한 k개의 원소를 갖는 복수의 부분 집합들 중, 상기 복수의 부분 집합들 각각에 포함되어 있는 k개의 원소들 중에서 서로 다른 두 원소 간의 모든 차들 각각이 상기 Z_v에 포함되어 있는 원소들에서 "0"을 제외한 서로 다른 1개의 원소와만 일치하는 것으로 확인되는 t개의 부분 집합들(상기 t개의 부분 집합들의 모임은 (v, k, 1)의 파라미터를 갖는 순환 차족으로 정의됨)을 확인한다.

[0025] 제1 시퀀스 생성부(115)는 상기 t개의 부분 집합들 각각에 대해서, 상기 t개의 부분 집합들 각각에 포함되어 있는 k개의 원소들에 기초하여 상기 Z_v에 포함되어 있는 "0"부터 "v-1"까지의 정수들에 대해 "0"부터 "v-1" 순으로 "1"과 "0"의 코드 값을 할당함으로써, v개의 "1"과 "0"의 코드 값으로 구성된 t개의 부분 시퀀스들을 생성하고, 상기 t개의 부분 시퀀스들을 조합하여 총 N개의 "1"과 "0"의 코드 값으로 구성된 제1 시퀀스(N=tv이며, 상기 t개의 부분 집합들 각각에 대해서 v개씩 "1"과 "0"의 코드 값을 할당함으로써 생성한 상기 t개의 부분 시퀀스들을 조합하여 상기 제1 시퀀스를 생성하였으므로, 상기 제1 시퀀스를 구성하는 "1"과 "0"의 코드 값은 총 N개가 됨)를 생성한다.

[0026] 이때, 제1 시퀀스 생성부(115)는 상기 t개의 부분 집합들 각각에 대해서, 상기 Z_v에 포함되어 있는 "0"부터 "v-1"까지의 정수들 중 상기 t개의 부분 집합들 각각에 포함되어 있는 k개의 원소들과 일치하는 정수들에 대해 "1"의 코드 값을 할당하고, 상기 t개의 부분 집합들 각각에 포함되어 있는 k개의 원소들과 일치하지 않는 나머지 정수들에 대해 "0"의 코드 값을 할당함으로써, 상기 v개의 "1"과 "0"의 코드 값으로 구성된 상기 t개의 부분 시퀀스들을 생성할 수 있다.

[0027] 그리고, 제2 시퀀스 생성부(116)는 상기 v개의 "1"과 "0"의 코드 값으로 구성된 상기 t개의 부분 시퀀스들 각각에 대해 선정된(predetermined) 방향으로 순환시프트(cyclic shift) 연산을 v-1회 수행하면서, 각 순환시프트 연산이 수행될 때마다 순환시프트 연산이 완료된 상기 t개의 부분 시퀀스들을 조합하여 제2 시퀀스를 생성함으로써, 총 v-1개의 제2 시퀀스들을 생성한다.

[0028] 여기서, 순환 차족(Cyclic Difference Family)에 대해 설명하면 다음과 같다.

[0029] "0"부터 "v-1"까지의 정수들을 원소로 포함하고 있는 집합인 Z_v가 있을 때, 상기 Z_v에 대한 k개의 원소를 갖는 복수의 부분 집합들을 $B_i = \{b_{i0}, b_{i1}, ..., b_{i(k-1)}\}$라고 하면, 상기 복수의 부분 집합들 B_i 중, 각 부분 집합에 포함되어 있는 k개의 원소들 중에서 서로 다른 두 원소 간의 모든 차들 각각이 상기 Z_v에 포함되어 있는 원소들에서 "0"을 제외한 서로 다른 λ개의 원소와 일치하는 것으로 연산되는 부분 집합들의 모임을 (v, k, λ)의 파라미터를 갖는 순환 차족이라고 한다.

[0030] 여기서, 상기 복수의 부분 집합들 B_i 각각에 포함되어 있는 k개의 원소들 중에서 서로 다른 두 원소 간의 모든 차들을 D라고 하는 경우, 상기 D는 하기의 수학식 1에 따라 연산된다. 만약, 상기 서로 다른 두 원소 간의 차가 음수 값이 되는 경우, 상기 서로 다른 두 원소 간의 차는 하기의 수학식 1에 따라 상기 음수 값에 대해 v로 나눈 나머지 값이 될 수 있다.

수학식 1

[0032]
$$D = (b_{il} - b_{im}) \mod v$$

[0034] 이때, (v, k, 1)의 파라미터를 갖는 순환 차족이란 "0"부터 "v-1"까지의 정수들을 원소로 포함하고 있는 집합인 Z_v에 대한 k개의 원소를 갖는 복수의 부분 집합들 중, 상기 복수의 부분 집합들 각각에 포함되어 있는 k개의 원소들 중에서 상기 수학식 1에 따라 연산되는 서로 다른 두 원소 간의 모든 차들 각각이 상기 Z_v에 포함되어 있는 원소들에서 "0"을 제외한 서로 다른 1개의 원소와만 일치하는 것으로 확인되는 부분 집합들의 모임을 (v, k, 1)의 파라미터를 갖는 순환 차족이라 정의할 수 있다.

[0035] 이때, v가 "v=k(k-1)t+1"의 조건을 만족한다고 하는 경우, (v, k, 1)의 파라미터를 갖는 순환 차족에 포함되는 부분 집합들은 총 t개 존재할 수 있다.

[0036] 예컨대, t=2, k=3, v=13이라고 가정하면, "0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12"를 원소로 포함하고 있는 집합인 Z_v가 있을 때, 3개의 원소를 포함하고 있는 Z_v의 부분 집합은 복수개가 존재할 수 있다. 이러한 복수의 부분 집합들은 각각 3개의 원소를 포함하고 있기 때문에 각 부분 집합에 포함되어 있는 서로 다른 두 원소 간의 차는 총 6개가 존재할 수 있고, 이러한 6개의 두 원소 간의 차가 상기 Z_v에 포함되어 있는 "0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12"이라는 정수 중 "0"을 제외한 오직 1개의 정수와만 일치하는 것으로 확인되는 부분 집합들의 모임을 (13, 3, 1)의 파라미터를 갖는 순환 차족이라 정의할 수 있다.

[0037] 관련해서, (13, 3, 1)의 파라미터를 갖는 순환 차족에 포함되는 부분 집합은 "{0, 1, 4}", "{0, 2, 7}"가 존재할 수 있다. 여기서, "{0, 1, 4}"의 경우 서로 다른 두 원소 간의 차는 상기 수학식 1에 따라 연산하게 되면 "1, 3, 4, 9, 10, 12"가 나올 수 있고, 이때, "1, 3, 4, 9, 10, 12" 각각은 Z_v에 포함되어 있는 정수들 중 1개의 정수와만 일치하기 때문에 "{0, 1, 4}"는 (13, 3, 1)의 파라미터를 갖는 순환 차족 중 하나에 포함되는 부분 집합이 될 수 있다. 그리고, "{0, 2, 7}"의 경우 서로 다른 두 원소 간의 차는 상기 수학식 1에 따라 연산하게 되면 "2, 5, 6, 7, 8, 11"이 나올 수 있고, 이때, "2, 5, 6, 7, 8, 11" 각각은 Z_v에 포함되어 있는 정수들 중 1개의 정수와만 일치하기 때문에 "{0, 2, 7}"도 (13, 3, 1)의 파라미터를 갖는 순환 차족 중 하나에 포함되는 부분 집합이 될 수 있다.

[0038] 이렇게, v가 "v=k(k-1)t+1"의 조건을 만족한다고 하는 경우, (v, k, 1)의 파라미터를 갖는 순환 차족에 포함되는 t개의 부분 집합들이 있을 때, 상기 t개의 부분 집합들 각각에 대해서, 상기 Z_v에 포함되어 있는 "0"부터 "v-1"까지의 정수들 중 상기 t개의 부분 집합들 각각에 포함되어 있는 k개의 원소들과 일치하는 정수들에 대해 "1"의 코드 값을 할당하고, 상기 t개의 부분 집합들 각각에 포함되어 있는 k개의 원소들과 일치하지 않는 나머지 정수들에 대해 "0"의 코드 값을 할당하게 되면, v개의 "1"과 "0"의 코드 값으로 구성된 시퀀스들을 t개 획득할 수 있다.

[0039] 이때, 상기 t개의 시퀀스들을 t개의 부분 시퀀스들이라고 하고, 상기 t개의 부분 시퀀스들을 일렬로 나열하여 서로 조합하게 되면, 총 N개(N=tv임)의 "1"과 "0"의 코드 값으로 구성된 제1 시퀀스를 획득할 수 있다.

[0040] 그리고, 상기 v개의 "1"과 "0"의 코드 값으로 구성된 상기 t개의 부분 시퀀스들 각각에 대해 선정된 방향으로 순환시프트 연산을 v-1회 수행하면서, 각 순환시프트 연산이 수행될 때마다 순환시프트 연산이 완료된 상기 t개의 부분 시퀀스들을 일렬로 나열하여 조합함에 따라 제2 시퀀스를 생성하게 되면, 총 v-1개의 제2 시퀀스들을 추가로 획득할 수 있다.

[0041] 이때, 상기 제1 시퀀스와 상기 제2 시퀀스들 내에 존재하는 "1"의 코드 값의 개수는 항상 tk개로 일정하며, v가 "v=k(k-1)t+1"의 조건을 만족한다고 하는 경우, 상기 제1 시퀀스와 상기 제2 시퀀스들 중에서 서로 다른 tk개의 시퀀스들을 랜덤하게 선택한 후 랜덤하게 선택된 tk개의 시퀀스들을 모두 "OR"연산하게 되면, "OR" 연산이 수행된 결과 값에서 존재하는 "1"의 코드 값의 개수는 항상 $t^2k^2 - {}_{tk}C_2$ 개 이상이 되고, "0"의 코드 값의 개수

는 항상 $_{tk}C_2-(tk-1)t$ 개 이하가 되는 특징을 가지게 된다.

[0042] 예컨대, k=3, t=2라고 가정하고, "v=k(k-1)t+1"라 할 때, v를 13이라고 하면, "0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12"를 원소로 포함하고 있는 집합인 Z_v에서 (13, 3, 1)의 파라미터를 갖는 순환 차족에 포함되는 부분 집합들은 "{0, 1, 4}", "{0, 2, 7}"이 존재하게 된다.

[0043] 이때, 상기 Z_v에 포함되어 있는 "0"부터 "12"까지의 정수들 중 상기 2개의 부분 집합들 "{0, 1, 4}", "{0, 2, 7}" 각각에 포함되어 있는 3개의 원소들과 일치하는 정수들에 대해 "1"의 코드 값을 할당하고, 상기 2개의 부분 집합들 "{0, 1, 4}", "{0, 2, 7}" 각각에 포함되어 있는 3개의 원소들과 일치하지 않는 나머지 정수들에 대해 "0"의 코드 값을 할당하게 되면, "1100100000000"과 "1010000100000"이라는 2개의 부분 시퀀스들을 획득할 수 있다.

[0044] 이때, "1100100000000"과 "1010000100000"이라는 2개의 부분 시퀀스들을 일렬로 나열하여 조합하게 되면, 총 26개의 "1"과 "0"의 코드 값으로 구성된 "11001000000001010000100000"이라는 제1 시퀀스를 획득할 수 있다.

[0045] 그리고, "1100100000000"과 "1010000100000"이라는 상기 2개의 부분 시퀀스들에 대해 오른쪽 방향으로 순환시프트 연산을 12회 수행하면서, 각 순환시프트 연산이 수행될 때마다 순환시프트 연산이 완료된 2개의 부분 시퀀스들을 일렬로 나열하여 조합함에 따라 제2 시퀀스를 생성하게 되면, 총 12개의 제2 시퀀스들을 추가로 획득할 수 있다.

[0046] 관련해서, "1100100000000"과 "1010000100000"이라는 상기 2개의 부분 시퀀스들에 대해 오른쪽 방향으로 1회 순환시프트 연산을 수행하게 되면, "0110010000000"과 "0101000010000"이라는 2개의 추가 부분 시퀀스들을 획득할 수 있고, "0110010000000"과 "0101000010000"이라는 2개의 추가 부분 시퀀스들을 서로 일렬로 나열하여 조합하게 되면, "01100100000000101000010000"이라는 추가 시퀀스를 획득할 수 있다. 이러한 방식으로 2개의 부분 시퀀스들에 대해 오른쪽 방향으로 순환시프트 연산을 총 12번 수행하면서 추가 시퀀스를 획득하게 되면, 순환시프트 연산이 12회 수행되었으므로, 총 12개의 시퀀스들을 추가로 획득할 수 있다.

[0047] 이러한 방식으로, 시퀀스를 생성하게 되면, 최종적으로 총 13개의 시퀀스들이 생성될 수 있고, 이때, 각 시퀀스들 내에 존재하는 "1"의 코드 값의 개수는 모두 6개로 일정하며, 상기 13개의 시퀀스들 중에서 서로 다른 6개의 시퀀스들을 랜덤하게 선택한 후 랜덤하게 선택된 6개의 시퀀스들을 모두 "OR"연산하게 되면, "OR" 연산이 수행된 결과 값에서 존재하는 "1"의 코드 값의 개수는 항상 21개 이상이 되며, "0"의 코드 값의 개수는 항상 5개 이하가 되는 특징을 가지게 된다.

[0048] 지금까지 설명한 순환 차족을 이용한 시퀀스 생성 과정에 따라, 순환 차족 확인부(114)는 N이 N=tv의 조건을 만족하는 자연수이고, v가 v=k(k-1)t+1의 조건을 만족하는 자연수이며, k가 2이상의 자연수이고, t가 1이상의 자연수일 때, "0"부터 "v-1"까지의 정수들을 원소로 포함하고 있는 집합인 Z_v에 대한 k개의 원소를 갖는 복수의 부분 집합들 중, 상기 복수의 부분 집합들 각각에 포함되어 있는 k개의 원소들 중에서 서로 다른 두 원소 간의 모든 차들 각각이 상기 Z_v에 포함되어 있는 원소들에서 "0"을 제외한 서로 다른 1개의 원소와만 일치하는 것으로 확인되는 t개의 부분 집합들(상기 t개의 부분 집합들의 모임은 (v, k, 1)의 파라미터를 갖는 순환 차족으로 정의됨)을 확인할 수 있다.

[0049] 이렇게, (v, k, 1)의 파라미터를 갖는 순환 차족에 포함되는 상기 t개의 부분 집합들에 대한 확인이 완료되면, 제1 시퀀스 생성부(115)는 상기 t개의 부분 집합들 각각에 대해서, 상기 Z_v에 포함되어 있는 "0"부터 "v-1"까지의 정수들 중 상기 t개의 부분 집합들 각각에 포함되어 있는 k개의 원소들과 일치하는 정수들에 대해 "1"의 코드 값을 할당하고, 상기 t개의 부분 집합들 각각에 포함되어 있는 k개의 원소들과 일치하지 않는 나머지 정수들에 대해 "0"의 코드 값을 할당함으로써, v개의 "1"과 "0"의 코드 값으로 구성된 t개의 부분 시퀀스들을 생성할 수 있다.

[0050] 그러고 나서, 제1 시퀀스 생성부(115)는 상기 t개의 부분 시퀀스들을 조합하여 총 N개(N=tv임)의 "1"과 "0"의 코드 값으로 구성된 제1 시퀀스를 생성할 수 있다.

[0051] 그리고, 제2 시퀀스 생성부(116)는 상기 v개의 "1"과 "0"의 코드 값으로 구성된 상기 t개의 부분 시퀀스들 각각에 대해 선정된 방향으로 순환시프트 연산을 v-1회 수행하면서, 각 순환시프트 연산이 수행될 때마다 순환시프트 연산이 완료된 상기 t개의 부분 시퀀스들을 조합하여 제2 시퀀스를 생성함으로써, 총 v-1개의 제2 시퀀스들

을 생성할 수 있다.

[0052] 이렇게, 상기 제1 시퀀스와 상기 v-1개의 제2 시퀀스들이 생성되면, 시퀀스 매칭부(117)는 상기 v개의 메시지 분산 그룹들 각각에 대해 상기 제1 시퀀스와 상기 v-1개의 제2 시퀀스들을 하나씩 매칭시킨다.

[0053] 그리고, 할당부(118)는 상기 v개의 메시지 분산 그룹들 각각에 대해, 상기 v개의 메시지 분산 그룹들 각각에 매칭되어 있는 각 시퀀스를 구성하는 N개의 코드 값에 기초하여 상기 N개의 메시지들을 tk개씩 분산 할당한다.

[0054] 이때, 본 발명의 일실시예에 따르면, 메시지 분산 할당부(112)는 상기 N개의 메시지들 각각에 대해 순번 정보와 연관된 인덱스 값을 할당하는 인덱스 할당부(119)를 더 포함할 수 있다.

[0055] 이때, 할당부(118)는 상기 v개의 메시지 분산 그룹들 각각에 대해, 상기 v개의 메시지 분산 그룹들 각각에 매칭되어 있는 각 시퀀스를 구성하는 N개의 코드 값의 배열 순서와 상기 N개의 메시지들 각각에 할당되어 있는 인덱스 값을 비교한 후 상기 N개의 메시지들에 할당되어 있는 인덱스 값과 일치하는 배열 순서의 코드 값에 기초하여 상기 N개의 메시지들을 tk개씩 분산 할당할 수 있다.

[0056] 이때, 본 발명의 일실시예에 따르면, 할당부(118)는 상기 v개의 메시지 분산 그룹들 각각에 대해, 상기 v개의 메시지 분산 그룹들 각각에 매칭되어 있는 각 시퀀스를 구성하는 N개의 코드 값 중 tk개씩 존재하는 "1"의 코드 값을 확인한 후, 상기 v개의 메시지 분산 그룹들 각각에 대해, 상기 N개의 메시지들 중 상기 "1"의 코드 값에 대한 배열 순서와 일치하는 인덱스 값이 할당되어 있는 tk개의 메시지들을 분산 할당할 수 있다.

[0057] 본 발명의 일실시예에 따르면, 순환 차족을 활용한 데이터 분산 저장 장치(110)는 저장 메시지 추출부(120) 및 데이터 로드부(121)를 더 포함할 수 있다.

[0058] 저장 메시지 추출부(120)는 v개의 데이터 저장소들(101, 102, 103)에 상기 N개의 메시지들이 tk개씩 분산 저장된 이후, 상기 N개의 메시지들에 대한 로드(load) 명령이 인가되면, 상기 로드 명령에 기초하여 v개의 데이터 저장소들(101, 102, 103) 중 tk개의 데이터 저장소들을 랜덤하게 선택하고, 상기 tk개의 데이터 저장소들 각각으로부터 상기 tk개의 데이터 저장소들 각각에 저장되어 있는 tk개의 저장 메시지들을 추출한다.

[0059] 데이터 로드부(121)는 상기 tk개의 데이터 저장소들 각각으로부터 tk개씩 추출된 저장 메시지들을 상기 N개의 메시지들에 대한 로드 명령에 따른 데이터로 로드한다.

[0060] 이때, 본 발명의 일실시예에 따르면, 상기 N개의 메시지들은 일부 메시지의 손실이 발생할 경우, 나머지 메시지들을 기초로 손실 메시지들의 복구가 가능한 최대 거리 분리 가능(Maximum Distance Separable: MDS) 코드일 수 있다.

[0061] 여기서, MDS 코드란 복수의 메시지들 중 일부 메시지의 손실이 발생하였을 때, 남아 있는 메시지들을 기초로 손실된 일부 메시지들을 복원할 수 있는 코드로, 소정의 원본 메시지에 대해 MDS 코딩을 수행함으로써, MDS 코드로 변경해 둔 후 MDS 코드로 변경된 메시지들 중 일부 메시지에 대한 손실이 발생하였을 때, 손실된 메시지를 복원하는데 사용될 수 있다. 이때, 상기 MDS 코드로부터 다시 원본 메시지를 생성하고자 한다면, MDS 디코딩을 수행함으로써, 상기 원본 메시지를 복원할 수 있다. 따라서, 중요 메시지들을 저장할 때, MDS 코드로 변경해서 저장해 둠으로써, 메시지의 손실이 발생할 경우를 미리 대비할 수 있다.

[0062] 이때, 데이터 로드부(121)는 상기 tk개의 데이터 저장소들 각각으로부터 tk개씩 저장 메시지들이 추출되면, 상기 추출된 저장 메시지들에 할당되어 있는 인덱스 값과 상기 N개의 메시지들에 할당된 인덱스 값을 비교하여 상기 N개의 메시지들 중 상기 추출된 저장 메시지들에 포함되어 있지 않은 잔여 메시지가 존재하는 것으로 확인되면, 상기 추출된 저장 메시지들을 기초로 상기 잔여 메시지를 복구하여 상기 추출된 저장 메시지들과 상기 복구된 잔여 메시지를 상기 N개의 메시지들에 대한 로드 명령에 따른 데이터로 로드할 수 있다.

[0063] 이하에서는 도 2를 참조하여, 본 발명에 따른 순환 차족을 활용한 데이터 분산 저장 장치(110)의 동작을 예를 들어 상세히 설명하기로 한다.

[0064] 도 2는 본 발명의 일실시예에 따른 순환 차족을 활용한 데이터 분산 저장 장치(110)의 동작을 설명하기 위한 도면이다.

[0065] 우선, k=3, t=2라고 가정하고, v=k(k-1)t+1임에 따라, v=13, N=tv임에 따라, N=26이라고 할 때, v개의 데이터 저장소들(101, 102, 103)이 도 2에 도시된 바와 같이, 데이터 저장소 1(201), 데이터 저장소 2(202), 데이터 저장소 3(203), 데이터 저장소 4(204), 데이터 저장소 5(205), 데이터 저장소 6(206), 데이터 저장소 7(207), 데이터 저장소 8(208), 데이터 저장소 9(209), 데이터 저장소 10(210), 데이터 저장소 11(211), 데이터 저장소

12(212), 데이터 저장소 13(213)으로 존재하며, N=26인 관계로 데이터 저장소 1~13(201, 202, 203, 204, 205, 206, 207, 208, 209, 210, 211, 212, 213)에 저장할 메시지들의 개수가 26개인 것으로 가정하기로 한다.

[0066] 이때, 분산 그룹 생성부(111)는 26개의 메시지들에 대해 데이터 저장소 1~13(201, 202, 203, 204, 205, 206, 207, 208, 209, 210, 211, 212, 213)로의 분산 저장 명령이 인가되면, 데이터 저장소 1~13(201, 202, 203, 204, 205, 206, 207, 208, 209, 210, 211, 212, 213)에 대응하는 13개의 메시지 분산 그룹들(401, 402, 403, 404, 405, 406, 407, 408, 409, 410, 411, 412, 413)을 생성할 수 있다.

[0067] 그리고, 메시지 분산 할당부(112)에 포함되어 있는 인덱스 할당부(119)는 상기 26개의 메시지들에 대해서 순번 정보와 연관된 인덱스 값을 할당할 수 있는데, 인덱스 할당부(119)가 상기 26개의 메시지들에 대해 순서대로 1부터 26까지의 인덱스 값을 할당한다고 하면, 상기 26개의 메시지들은 "m_1, m_2, m_3, m_4, m_5, m_6, m_7, m_8, m_9, m_{10}, m_{11}, m_{12}, m_{13}, m_{14}, m_{15}, m_{16}, m_{17}, m_{18}, m_{19}, m_{20}, m_{21}, m_{22}, m_{23}, m_{24}, m_{25}, m_{26}"으로 표시할 수 있다.

[0068] k=3, t=2, v=13이라고 가정하였기 때문에, 순환 차족 확인부(114)는 "0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12"를 원소로 포함하고 있는 집합인 Z_v에 대한 3개의 원소를 갖는 복수의 부분 집합들 중, 상기 복수의 부분 집합들 각각에 포함되어 있는 3개의 원소들 중에서 서로 다른 두 원소 간의 모든 차들 각각이 상기 Z_v에 포함되어 있는 원소들에서 "0"을 제외한 서로 다른 1개의 원소와만 일치하는 것으로 확인되는 2개의 부분 집합들을 확인할 수 있다.

[0069] 이때, 상기 2개의 부분 집합들의 모임은 (13, 3, 1)의 파라미터를 갖는 순환 차족으로 정의된다.

[0070] 이때, 제1 시퀀스 생성부(115)는 (13, 3, 1)의 파라미터를 갖는 순환 차족에 포함되는 상기 2개의 부분 집합들 각각에 대해, 상기 2개의 부분 집합들 각각에 포함되어 있는 3개의 원소들에 기초하여 상기 Z_v에 포함되어 있는 "0"부터 "12"까지의 정수들에 대해 "0"부터 "12" 순으로 "1"과 "0"의 코드 값을 할당함으로써, 13개의 "1"과 "0"의 코드 값으로 구성된 2개의 부분 시퀀스들을 생성하고, 상기 2개의 부분 시퀀스들을 조합하여 총 26개의 "1"과 "0"의 코드 값으로 구성된 제1 시퀀스를 생성할 수 있다.

[0071] 관련해서, 제1 시퀀스 생성부(115)가 (13, 3, 1)의 파라미터를 갖는 순환 차족인 상기 2개의 부분 집합을 "{0, 1, 4}", "{0, 2, 7}"로 확인하였다고 가정하자.

[0072] 이때, 제1 시퀀스 생성부(115)는 상기 Z_v에 포함되어 있는 "0"부터 "12"까지의 정수들 중 상기 2개의 부분 집합들 "{0, 1, 4}", "{0, 2, 7}" 각각에 대해, 상기 2개의 부분 집합들 "{0, 1, 4}", "{0, 2, 7}" 각각에 포함되어 있는 3개의 원소들과 일치하는 정수들에 대해 "1"의 코드 값을 할당하고, 상기 2개의 부분 집합들 "{0, 1, 4}", "{0, 2, 7}" 각각에 포함되어 있는 3개의 원소들과 일치하지 않는 나머지 정수들에 대해 "0"의 코드 값을 할당함으로써, "1100100000000"과 "1010000100000"이라는 2개의 부분 시퀀스들을 생성할 수 있다.

[0073] 그러고 나서, 제1 시퀀스 생성부(115)는 "1100100000000"과 "1010000100000"이라는 2개의 부분 시퀀스들을 일렬로 나열하여 조합함으로써, 총 26개의 "1"과 "0"의 코드 값으로 구성된 11001000000001010000100000(301)이라는 제1 시퀀스를 획득할 수 있다.

[0074] 그리고, 제2 시퀀스 생성부(116)는 "1100100000000"과 "1010000100000"이라는 상기 2개의 부분 시퀀스들에 대해 오른쪽 또는 왼쪽 중 어느 하나의 선정된 방향으로 순환시프트 연산을 12회 수행(본 실시예에서는 오른쪽 방향으로 순환시프트 연산을 수행함)하면서, 각 순환시프트 연산이 수행될 때마다 순환시프트 연산이 완료된 2개의 부분 시퀀스들을 일렬로 나열하여 조합하여 제2 시퀀스를 생성함으로써, 총 12개의 01100100000000101000010000(302), 00110010000000010100001000(303), 00011001000000001010000100(304), 00001100100000000101000010(305), 00000110010000000010100001(306), 00000011001001000001010000(307), 00000001100100100000101000(308), 00000000110010010000010100(309), 10000000011000001000001010(310), 01000000001100000100000101(311), 00100000000111000010000010(312), 10010000000010100001000001(313)이라는 제2 시퀀스들을 생성할 수 있다.

[0075] 이러한 제1 시퀀스인 11001000000001010000100000(301)와 12개의 제2 시퀀스들인 01100100000000101000010000(302), 00110010000000010100001000(303), 00011001000000001010000100(304), 00001100100000000101000010(305), 00000110010000000010100001(306), 00000011001001000001010000(307), 00000001100100100000101000(308), 00000000110010010000010100(309), 10000000011000001000001010(310), 01000000001100000100000101(311), 00100000000111000010000010(312), 10010000000010100001000001(313)는 앞

서 설명한 바와 같이, 항상 tk개의 "1"의 코드 값을 갖게 되는데, 본 실시예에서 t=2, k=3으로 가정하였기 때문에 13개의 시퀀스들(301, 302, 303, 304, 305, 306, 307, 308, 309, 310, 311, 312, 313) 각각에 포함된 "1"의 코드 값은 항상 6개이다.

[0076] 그리고, 제1 시퀀스와 제2 시퀀스들은 서로 다른 tk개의 시퀀스들 간에 "OR" 논리 연산을 수행하면, 해당 논리 연산의 결과 값에 포함된 "1"의 코드 값은 항상 $t^2k^2 - {}_{tk}C_2$ 개 이상이 되고, "0"의 코드 값은 항상 ${}_{tk}C_2 - (tk-1)t$ 개 이하가 되기 때문에 본 실시예에서는 t=2, k=3이라고 가정하였으므로, 13개의 시퀀스들(301, 302, 303, 304, 305, 306, 307, 308, 309, 310, 311, 312, 313) 중 어느 6개의 시퀀스들 간에 "OR" 논리 연산이 수행되면, 그 연산 결과 값에 포함되어 있는 "1"의 코드 값은 항상 21개 이상이 되고, "0"의 코드 값은 항상 5개 이하가 된다.

[0077] 이렇게, 13개의 시퀀스들(301, 302, 303, 304, 305, 306, 307, 308, 309, 310, 311, 312, 313)이 생성되면, 시퀀스 매칭부(117)는 데이터 저장소 1~13(201, 202, 203, 204, 205, 206, 207, 208, 209, 210, 211, 212, 213)에 각각 대응하는 13개의 메시지 분산 그룹들(401, 402, 403, 404, 405, 406, 407, 408, 409, 410, 411, 412, 413) 각각에 대해 13개의 시퀀스들(301, 302, 303, 304, 305, 306, 307, 308, 309, 310, 311, 312, 313)을 하나씩 매칭시킬 수 있다.

[0078] 관련해서, 시퀀스 매칭부(117)는 도 2에 도시된 바와 같이, 메시지 분산 그룹 1(401)에 대해 1100100000000101000010000(301)을 매칭시킬 수 있고, 메시지 분산 그룹 2(402)에 대해 0110010000000010100001000(302)을 매칭시킬 수 있으며, 메시지 분산 그룹 3(403)에 대해 0011001000000001010000100(303)을 매칭시킬 수 있고, 메시지 분산 그룹 4(404)에 대해 0001100100000000101000010(304)을 매칭시킬 수 있고, 메시지 분산 그룹 5(405)에 대해 0000110010000000010100001(305)을 매칭시킬 수 있고, 메시지 분산 그룹 6(406)에 대해 0000011001000000001010000(306)을 매칭시킬 수 있고, 메시지 분산 그룹 7(407)에 대해 0000001100100100000101000(307)을 매칭시킬 수 있고, 메시지 분산 그룹 8(408)에 대해 0000000110010010000010100(308)을 매칭시킬 수 있고, 메시지 분산 그룹 9(409)에 대해 0000000011001001000001010(309)을 매칭시킬 수 있고, 메시지 분산 그룹 10(410)에 대해 1000000001100000100000101(310)을 매칭시킬 수 있고, 메시지 분산 그룹 11(411)에 대해 0100000000110000010000010(311)을 매칭시킬 수 있고, 메시지 분산 그룹 12(412)에 대해 0010000000011100001000001(312)을 매칭시킬 수 있고, 메시지 분산 그룹 13(413)에 대해 1001000000001010000100001(313)을 매칭시킬 수 있고,

[0079] 그러고 나서, 할당부(118)는 13개의 메시지 분산 그룹들(401, 402, 403, 404, 405, 406, 407, 408, 409, 410, 411, 412, 413) 각각에 대해, 13개의 메시지 분산 그룹들(401, 402, 403, 404, 405, 406, 407, 408, 409, 410, 411, 412, 413) 각각에 매칭되어 있는 각 시퀀스를 구성하는 26개의 코드 값에 기초하여 상기 26개의 메시지들인 "m_1, m_2, m_3, m_4, m_5, m_6, m_7, m_8, m_9, m_{10}, m_{11}, m_{12}, m_{13}, m_{14}, m_{15}, m_{16}, m_{17}, m_{18}, m_{19}, m_{20}, m_{21}, m_{22}, m_{23}, m_{24}, m_{25}, m_{26}"을 할당할 수 있는데, 할당부(118)는 13개의 메시지 분산 그룹들(401, 402, 403, 404, 405, 406, 407, 408, 409, 410, 411, 412, 413) 각각에 매칭되어 있는 각 시퀀스를 구성하는 26개의 코드 값의 배열 순서와 상기 26개의 메시지들인 "m_1, m_2, m_3, m_4, m_5, m_6, m_7, m_8, m_9, m_{10}, m_{11}, m_{12}, m_{13}, m_{14}, m_{15}, m_{16}, m_{17}, m_{18}, m_{19}, m_{20}, m_{21}, m_{22}, m_{23}, m_{24}, m_{25}, m_{26}" 각각에 할당되어 있는 인덱스 값을 비교한 후 상기 26개의 메시지들인 "m_1, m_2, m_3, m_4, m_5, m_6, m_7, m_8, m_9, m_{10}, m_{11}, m_{12}, m_{13}, m_{14}, m_{15}, m_{16}, m_{17}, m_{18}, m_{19}, m_{20}, m_{21}, m_{22}, m_{23}, m_{24}, m_{25}, m_{26}"에 할당되어 있는 인덱스 값과 일치하는 배열 순서의 코드 값에 기초하여 상기 26개의 메시지들인 "m_1, m_2, m_3, m_4, m_5, m_6, m_7, m_8, m_9, m_{10}, m_{11}, m_{12}, m_{13}, m_{14}, m_{15}, m_{16}, m_{17}, m_{18}, m_{19}, m_{20}, m_{21}, m_{22}, m_{23}, m_{24}, m_{25}, m_{26}"을 13개의 메시지 분산 그룹들(401, 402, 403, 404, 405, 406, 407, 408, 409, 410, 411, 412, 413)에 할당할 수 있다.

[0080] 이때, 할당부(118)는 13개의 메시지 분산 그룹들(401, 402, 403, 404, 405, 406, 407, 408, 409, 410, 411, 412, 413) 각각에 매칭되어 있는 각 시퀀스를 구성하는 26개의 코드 값 중 6개씩 존재하는 "1"의 코드 값을 확인한 후 상기 26개의 메시지들인 "m_1, m_2, m_3, m_4, m_5, m_6, m_7, m_8, m_9, m_{10}, m_{11}, m_{12}, m_{13}, m_{14}, m_{15}, m_{16}, m_{17}, m_{18}, m_{19}, m_{20}, m_{21}, m_{22}, m_{23}, m_{24}, m_{25}, m_{26}" 중 "1"의 코드 값에 대한 배열 순서와 일치하는 인덱스 값이 할당되어 있

는 6개의 메시지들을 13개의 메시지 분산 그룹들(401, 402, 403, 404, 405, 406, 407, 408, 409, 410, 411, 412, 413)에 할당할 수 있다.

[0081] 예컨대, 메시지 분산 그룹 1(401)에 대해서는 1100100000000101000010000(301)라는 시퀀스가 매칭되어 있기 때문에 할당부(118)는 1100100000000101000010000(301)로부터 "1"의 코드 값을 확인한 후 상기 26개의 메시지들인 "m_1, m_2, m_3, m_4, m_5, m_6, m_7, m_8, m_9, m_{10}, m_{11}, m_{12}, m_{13}, m_{14}, m_{15}, m_{16}, m_{17}, m_{18}, m_{19}, m_{20}, m_{21}, m_{22}, m_{23}, m_{24}, m_{25}, m_{26}" 중에서 "1"의 코드 값에 대한 배열 순서와 일치하는 인덱스 값이 할당되어 있는 6개의 메시지들을 메시지 분산 그룹 1(401)에 할당할 수 있다.

[0082] 관련해서, 1100100000000101000010000(301)에서는 "1"의 코드 값이 1번, 2번, 5번, 14번, 16번, 21번 위치에 배열되어 있기 때문에 할당부(118)는 도 2에 도시된 바와 같이, 메시지 분산 그룹 1(401)에 대해 1번, 2번, 5번, 14번, 16번, 21번 위치에 대응하는 인덱스 값이 할당되어 있는 "m_1, m_2, m_5, m_{14}, m_{16}, m_{21}"을 할당할 수 있다.

[0083] 이러한 방식으로, 할당부(118)는 13개의 메시지 분산 그룹들(401, 402, 403, 404, 405, 406, 407, 408, 409, 410, 411, 412, 413) 각각에 대해 도 2에 도시된 바와 같이 각 메시지 분산 그룹들(401, 402, 403, 404, 405, 406, 407, 408, 409, 410, 411, 412, 413)에 매칭되어 있는 각 시퀀스에 기초하여 상기 26개의 메시지들인 "m_1, m_2, m_3, m_4, m_5, m_6, m_7, m_8, m_9, m_{10}, m_{11}, m_{12}, m_{13}, m_{14}, m_{15}, m_{16}, m_{17}, m_{18}, m_{19}, m_{20}, m_{21}, m_{22}, m_{23}, m_{24}, m_{25}, m_{26}" 을 6개씩 분산 할당할 수 있다.

[0084] 이렇게, 상기 26개의 메시지들인 "m_1, m_2, m_3, m_4, m_5, m_6, m_7, m_8, m_9, m_{10}, m_{11}, m_{12}, m_{13}, m_{14}, m_{15}, m_{16}, m_{17}, m_{18}, m_{19}, m_{20}, m_{21}, m_{22}, m_{23}, m_{24}, m_{25}, m_{26}"이 모두 13개의 메시지 분산 그룹들(401, 402, 403, 404, 405, 406, 407, 408, 409, 410, 411, 412, 413)에 6개씩 분산 할당이 완료되면, 데이터 저장부(113)는 13개의 메시지 분산 그룹들(401, 402, 403, 404, 405, 406, 407, 408, 409, 410, 411, 412, 413) 각각에 할당되어 있는 6개의 메시지들을 13개의 메시지 분산 그룹들(401, 402, 403, 404, 405, 406, 407, 408, 409, 410, 411, 412, 413) 별로 데이터 저장소 1~13(201, 202, 203, 204, 205, 206, 207, 208, 209, 210, 211, 212, 213) 중 각 메시지 분산 그룹에 대응되는 데이터 저장소에 분리해서 저장할 수 있다.

[0085] 예컨대, 데이터 저장부(113)는 메시지 분산 그룹 1(401)에 할당되어 있는 6개의 메시지들인 "m_1, m_2, m_5, m_{14}, m_{16}, m_{21}"을 대응되는 데이터 저장소인 데이터 저장소 1(201)에 저장할 수 있다.

[0086] 이러한 방식으로, 데이터 저장부(113)는 13개의 메시지 분산 그룹들(401, 402, 403, 404, 405, 406, 407, 408, 409, 410, 411, 412, 413) 각각에 할당되어 있는 6개의 메시지들을 13개의 메시지 분산 그룹들(401, 402, 403, 404, 405, 406, 407, 408, 409, 410, 411, 412, 413) 별로 데이터 저장소 1~13(201, 202, 203, 204, 205, 206, 207, 208, 209, 210, 211, 212, 213) 중 각 메시지 분산 그룹에 대응되는 데이터 저장소에 분리해서 저장할 수 있다.

[0087] 이렇게, 상기 26개의 메시지들인 "m_1, m_2, m_3, m_4, m_5, m_6, m_7, m_8, m_9, m_{10}, m_{11}, m_{12}, m_{13}, m_{14}, m_{15}, m_{16}, m_{17}, m_{18}, m_{19}, m_{20}, m_{21}, m_{22}, m_{23}, m_{24}, m_{25}, m_{26}"이 데이터 저장소 1~13(201, 202, 203, 204, 205, 206, 207, 208, 209, 210, 211, 212, 213)에 분산 저장이 완료되면, 추후 상기 26개의 메시지들인 "m_1, m_2, m_3, m_4, m_5, m_6, m_7, m_8, m_9, m_{10}, m_{11}, m_{12}, m_{13}, m_{14}, m_{15}, m_{16}, m_{17}, m_{18}, m_{19}, m_{20}, m_{21}, m_{22}, m_{23}, m_{24}, m_{25}, m_{26}"에 대해서 데이터 로드 명령이 인가됨에 따라 상기 26개의 메시지들인 "m_1, m_2, m_3, m_4, m_5, m_6, m_7, m_8, m_9, m_{10}, m_{11}, m_{12}, m_{13}, m_{14}, m_{15}, m_{16}, m_{17}, m_{18}, m_{19}, m_{20}, m_{21}, m_{22}, m_{23}, m_{24}, m_{25}, m_{26}"에 대한 데이터 로드를 수행할 때, 데이터 저장소 1~13(201, 202, 203, 204, 205, 206, 207, 208, 209, 210, 211, 212, 213) 중에서 어느 6개의 데이터 저장소에만 액세스를 한 후 상기 6개의 데이터 저장소만에 저장되어 있는 메시지들을 추출하게 되면, 상기 26개의 메시지들인 "m_1, m_2, m_3, m_4, m_5, m_6, m_7, m_8, m_9, m_{10}, m_{11}, m_{12}, m_{13}, m_{14}, m_{15}, m_{16}, m_{17}, m_{18}, m_{19}, m_{20}, m_{21}, m_{22}, m_{23}, m_{24}, m_{25}, m_{26}" 중에서 적어도 21개 이상의 메시지는 모두 추출되게 된다.

[0088] 왜냐하면, 데이터 저장소 1~13(201, 202, 203, 204, 205, 206, 207, 208, 209, 210, 211, 212, 213)에는 각 시퀀스들(301, 302, 303, 304, 305, 306, 307, 308, 309, 310, 311, 312, 313)의 "1"의 코드 값에 따라서 상기 26개의 메시지들인 "m_1, m_2, m_3, m_4, m_5, m_6, m_7, m_8, m_9, m_{10}, m_{11}, m_{12}, m_{13}, m_{14}, m_{15}, m_{16}, m_{17}, m_{18}, m_{19}, m_{20}, m_{21},

m_{22}, m_{23}, m_{24}, m_{25}, m_{26}"이 분산 할당되어 저장되었고, 각 시퀀스들(301, 302, 303, 304, 305, 306, 307, 308, 309, 310, 311, 312, 313)의 특성상 상기 13개의 시퀀스들(301, 302, 303, 304, 305, 306, 307, 308, 309, 310, 311, 312, 313) 중 어느 6개의 시퀀스들 간에 "OR"논리 연산을 수행하게 되면, "1"의 코드 값이 항상 21개 이상이 나오기 때문에 데이터 저장소 1~13(201, 202, 203, 204, 205, 206, 207, 208, 209, 210, 211, 212, 213) 중 어느 6개의 랜덤한 데이터 저장소로부터 각각 6개씩 메시지들을 모두 추출하게 되면, 상기 26개의 메시지들인 "m_1, m_2, m_3, m_4, m_5, m_6, m_7, m_8, m_9, m_{10}, m_{11}, m_{12}, m_{13}, m_{14}, m_{15}, m_{16}, m_{17}, m_{18}, m_{19}, m_{20}, m_{21}, m_{22}, m_{23}, m_{24}, m_{25}, m_{26}" 중에서 적어도 21개 이상의 메시지들은 모두 확보하게 되는 것이다.

[0089] 이로 인해서, 상기 26개의 메시지들인 "m_1, m_2, m_3, m_4, m_5, m_6, m_7, m_8, m_9, m_{10}, m_{11}, m_{12}, m_{13}, m_{14}, m_{15}, m_{16}, m_{17}, m_{18}, m_{19}, m_{20}, m_{21}, m_{22}, m_{23}, m_{24}, m_{25}, m_{26}"이 데이터 저장소 1~13(201, 202, 203, 204, 205, 206, 207, 208, 209, 210, 211, 212, 213)에 분산 저장이 완료된 이후에 상기 26개의 메시지들인 "m_1, m_2, m_3, m_4, m_5, m_6, m_7, m_8, m_9, m_{10}, m_{11}, m_{12}, m_{13}, m_{14}, m_{15}, m_{16}, m_{17}, m_{18}, m_{19}, m_{20}, m_{21}, m_{22}, m_{23}, m_{24}, m_{25}, m_{26}"에 대한 로드 명령이 인가되면, 저장 메시지 추출부(120)는 데이터 저장소 1~13(201, 202, 203, 204, 205, 206, 207, 208, 209, 210, 211, 212, 213) 중 데이터 트래픽이 가장 낮은 데이터 저장소 6개만을 랜덤하게 선택하여 6개의 데이터 저장소에 각각 저장되어 있는 6개의 저장 데이터를 추출할 수 있고, 데이터 로드부(121)는 상기 6개의 데이터 저장소를 통해 추출된 각 저장 데이터를 기초로 상기 26개의 메시지들인 "m_1, m_2, m_3, m_4, m_5, m_6, m_7, m_8, m_9, m_{10}, m_{11}, m_{12}, m_{13}, m_{14}, m_{15}, m_{16}, m_{17}, m_{18}, m_{19}, m_{20}, m_{21}, m_{22}, m_{23}, m_{24}, m_{25}, m_{26}" 중 총 21개 이상의 메시지들을 빠르게 로드할 수 있다.

[0090] 이때, 만약 상기 26개의 메시지들인 "m_1, m_2, m_3, m_4, m_5, m_6, m_7, m_8, m_9, m_{10}, m_{11}, m_{12}, m_{13}, m_{14}, m_{15}, m_{16}, m_{17}, m_{18}, m_{19}, m_{20}, m_{21}, m_{22}, m_{23}, m_{24}, m_{25}, m_{26}"이 임의의 21개의 메시지들만으로 상기 26개의 메시지들인 "m_1, m_2, m_3, m_4, m_5, m_6, m_7, m_8, m_9, m_{10}, m_{11}, m_{12}, m_{13}, m_{14}, m_{15}, m_{16}, m_{17}, m_{18}, m_{19}, m_{20}, m_{21}, m_{22}, m_{23}, m_{24}, m_{25}, m_{26}"을 모두 복원할 수 있도록 구성된 (26, 21) MDS 코드인 경우, 데이터 로드부(121)는 데이터 저장소 1~13(201, 202, 203, 204, 205, 206, 207, 208, 209, 210, 211, 212, 213) 중에서 랜덤하게 선택된 6개의 데이터 저장소들로부터 6개씩 저장 메시지가 추출되면, 상기 추출된 저장 메시지들에 할당되어 있는 인덱스 값과 상기 26개의 메시지들인 "m_1, m_2, m_3, m_4, m_5, m_6, m_7, m_8, m_9, m_{10}, m_{11}, m_{12}, m_{13}, m_{14}, m_{15}, m_{16}, m_{17}, m_{18}, m_{19}, m_{20}, m_{21}, m_{22}, m_{23}, m_{24}, m_{25}, m_{26}"에 할당된 인덱스 값을 비교하여 상기 26개의 메시지들인 "m_1, m_2, m_3, m_4, m_5, m_6, m_7, m_8, m_9, m_{10}, m_{11}, m_{12}, m_{13}, m_{14}, m_{15}, m_{16}, m_{17}, m_{18}, m_{19}, m_{20}, m_{21}, m_{22}, m_{23}, m_{24}, m_{25}, m_{26}" 중 상기 추출된 저장 메시지들에 포함되어 있지 않은 잔여 메시지가 존재하는 것으로 확인되는 경우, 상기 추출된 저장 메시지들을 기초로 상기 잔여 메시지를 복구하여 상기 추출된 저장 메시지들과 상기 복구된 잔여 메시지를 상기 26개의 메시지들인 "m_1, m_2, m_3, m_4, m_5, m_6, m_7, m_8, m_9, m_{10}, m_{11}, m_{12}, m_{13}, m_{14}, m_{15}, m_{16}, m_{17}, m_{18}, m_{19}, m_{20}, m_{21}, m_{22}, m_{23}, m_{24}, m_{25}, m_{26}"에 대한 로드 명령에 따른 데이터로 로드할 수 있다.

[0091] 관련해서, 상기 랜덤하게 선택된 6개의 데이터 저장소를 데이터 저장소 2(202), 데이터 저장소 5(205), 데이터 저장소 7(207), 데이터 저장소 8(208), 데이터 저장소 10(210), 데이터 저장소 13(213)이라고 하면, 저장 메시지 추출부(120)는 데이터 저장소 2(202)로부터 "m_2, m_3, m_6, m_{15}, m_{17}, m_{22}"를 추출할 수 있고, 데이터 저장소 5(205)로부터 "m_5, m_6, m_9, m_{18}, m_{20}, m_{25}"를 추출할 수 있으며, 데이터 저장소 7(207)로부터 "m_7, m_8, m_{11}, m_{14}, m_{20}, m_{22}"를 추출할 수 있고, 데이터 저장소 8(208)로부터 "m_8, m_9, m_{12}, m_{15}, m_{21}, m_{23}"을 추출할 수 있고, 데이터 저장소 10(210)으로부터 "m_1, m_{10}, m_{11}, m_{17}, m_{23}, m_{25}"를 추출할 수 있고, 데이터 저장소 13(213)으로부터 "m_1, m_4, m_{13}, m_{15}, m_{20}, m_{26}"을 추출할 수 있다.

[0092] 이로 인해, 저장 메시지 추출부(120)는 결국, 데이터 저장소 2(202), 데이터 저장소 5(205), 데이터 저장소 7(207), 데이터 저장소 8(208), 데이터 저장소 10(210), 데이터 저장소 13(213)로부터 "m_2, m_3, m_5, m_6, m_7, m_8, m_9, m_{10}, m_{11}, m_{12}, m_{13}, m_{14}, m_{15}, m_{17}, m_{18}, m_{20}, m_{21}, m_{22}, m_{23}, m_{25}, m_{26}"을 추출해 낼 수 있다.

[0093] 이때, 데이터 로드부(121)는 상기 26개의 메시지들인 "m_1, m_2, m_3, m_4, m_5, m_6, m_7, m_8, m_9, m_{10}, m_{11}, m_{12}, m_{13}, m_{14}, m_{15}, m_{16}, m_{17}, m_{18}, m_{19}, m_{20}, m_{21}, m_{22}, m_{23}, m_{24}, m_{25}, m_{26}"에 할당되어 있는 인덱스 값과 상기 추출된 "m_2, m_3,

m_5, m_6, m_7, m_8, m_9, m_{10}, m_{11}, m_{12}, m_{13}, m_{14}, m_{15}, m_{17}, m_{18}, m_{20}, m_{21}, m_{22}, m_{23}, m_{25}, m_{26}"에 할당되어 있는 인덱스 값을 서로 비교하여 상기 26개의 메시지들인 "m_1, m_2, m_3, m_4, m_5, m_6, m_7, m_8, m_9, m_{10}, m_{11}, m_{12}, m_{13}, m_{14}, m_{15}, m_{16}, m_{17}, m_{18}, m_{19}, m_{20}, m_{21}, m_{22}, m_{23}, m_{24}, m_{25}, m_{26}" 중에서 상기 추출된 "m_2, m_3, m_5, m_6, m_7, m_8, m_9, m_{10}, m_{11}, m_{12}, m_{13}, m_{14}, m_{15}, m_{17}, m_{18}, m_{20}, m_{21}, m_{22}, m_{23}, m_{25}, m_{26}"에 포함되어 있지 않은 잔여 메시지를 확인할 수 있다.

[0094] 관련해서, 본 실시예에서는 "m_1, m_4, m_{16}, m_{19}, m_{24}"가 잔여 메시지이기 때문에 데이터 로드부(121)는 "m_1, m_4, m_{16}, m_{19}, m_{24}"를 잔여 메시지로 확인한 후 (26, 21) MDS 코딩이 되어 있는 "m_2, m_3, m_5, m_6, m_7, m_8, m_9, m_{10}, m_{11}, m_{12}, m_{13}, m_{14}, m_{15}, m_{17}, m_{18}, m_{20}, m_{21}, m_{22}, m_{23}, m_{25}, m_{26}"으로부터 "m_1, m_4, m_{16}, m_{19}, m_{24}"를 복원해 낼 수 있다.

[0095] 이로 인해, 결국, 데이터 로드부(121)는 상기 복원된 "m_1, m_4, m_{16}, m_{19}, m_{24}"와 데이터 저장소 2(202), 데이터 저장소 5(205), 데이터 저장소 7(207), 데이터 저장소 8(208), 데이터 저장소 10(210), 데이터 저장소 13(213)에서 추출된 "m_2, m_3, m_5, m_6, m_7, m_8, m_9, m_{10}, m_{11}, m_{12}, m_{13}, m_{14}, m_{15}, m_{17}, m_{18}, m_{20}, m_{21}, m_{22}, m_{23}, m_{25}, m_{26}"을 로드함으로써, 상기 26개의 메시지들인 "m_1, m_2, m_3, m_4, m_5, m_6, m_7, m_8, m_9, m_{10}, m_{11}, m_{12}, m_{13}, m_{14}, m_{15}, m_{16}, m_{17}, m_{18}, m_{19}, m_{20}, m_{21}, m_{22}, m_{23}, m_{24}, m_{25}, m_{26}"을 모두 로드해 낼 수 있다.

[0096] 결국, 본 발명에 따른 순환 차족을 활용한 데이터 분산 저장 장치(110)는 상기 26개의 메시지들인 "m_1, m_2, m_3, m_4, m_5, m_6, m_7, m_8, m_9, m_{10}, m_{11}, m_{12}, m_{13}, m_{14}, m_{15}, m_{16}, m_{17}, m_{18}, m_{19}, m_{20}, m_{21}, m_{22}, m_{23}, m_{24}, m_{25}, m_{26}"을 순환 차족에 기초하여 생성한 상기 제1 시퀀스와 상기 제2 시퀀스들에 따라 데이터 저장소 1~13(201, 202, 203, 204, 205, 206, 207, 208, 209, 210, 211, 212, 213)에 6개씩 분산 저장해 놓음으로써, 추후 데이터를 로드할 때, 오직 6개의 데이터 저장소에만 액세스하여 데이터를 복원해 낼 수 있기 때문에 메시지 분산 저장 시스템에 있어서 시스템 자원 활용의 효율성을 도모할 수 있다.

[0097] 본 발명의 다른 실시예에 따르면, 순환 차족을 활용한 데이터 분산 저장 장치(110)는 저장 대상이 되는 메시지의 개수가 N개를 초과하고, 데이터 저장소들(101, 102, 103)의 개수도 상기 메시지의 개수만큼 증가시킬 수 있다면, (v, k, 1)의 파라미터를 갖는 순환 차족 중에서 완전 차족(Perfect Difference Family)에 속하는 t개의 부분 집합들을 확인하여 상기 제1 시퀀스를 생성하고, 상기 제1 시퀀스를 구성하는 코드 값에서 "0"의 코드 값을 상기 저장 대상이 되는 메시지의 초과 개수만큼 추가로 삽입하는 방식으로 상기 제1 시퀀스와 상기 제2 시퀀스들을 생성함으로써, 데이터 저장소(101, 102, 103)에 저장할 메시지의 개수를 증가시킬 수 있다.

[0098] 예컨대, 저장 대상이 되는 메시지의 개수가 N+ta라고 하고, 데이터 저장소들(101, 102, 103)의 개수가 v+a라고 하는 경우, 순환 차족을 활용한 데이터 분산 저장 장치(110)는 (v, k, 1)의 파라미터를 갖는 완전 차족에 속하는 t개의 부분 집합들을 확인하여 N개의 "0"과 "1"의 코드 값으로 구성된 상기 제1 시퀀스를 생성한 후 상기 제1 시퀀스를 구성하는 t개의 부분 시퀀스들 각각에 대해 a개 만큼의 "0"의 코드 값을 삽입하여 상기 제1 시퀀스를 N+ta개의 "0"과 "1"의 코드 값으로 구성된 시퀀스로 변경한 후 상기 변경된 제1 시퀀스를 구성하는 부분 시퀀스들에 대해 선정된 방향으로 v+a-1회 만큼 순환시프트 연산을 수행함으로써, v+a-1개의 제2 시퀀스들을 생성할 수 있다.

[0099] 그러고 나서, 순환 차족을 활용한 데이터 분산 저장 장치(110)는 상기 제1 시퀀스와 상기 v+a-1개의 제2 시퀀스들을 이용해서 N+ta개의 메시지들을 v+a개의 데이터 저장소들(101, 102, 103)에 분산 저장할 수 있다.

[0100] 이러한 방식으로 본 발명에 따른 순환 차족을 활용한 데이터 분산 저장 장치(110)는 상기 제1 시퀀스의 길이를 증가시킴으로써, 저장 메시지들의 개수를 자유롭게 증가시킬 수 있다.

[0101] 여기서, 완전 차족에 대해 설명하면, v=k(k-1)t+1로 주어진 (v, k, 1) 순환 차족 $B_i = \{b_{i0}, b_{i1}, \ldots, b_{i(k-1)}\}$가 있고, B_i가 $b_{i0} < b_{i1} < \cdots < b_{i(k-1)}$를 만족할 때, $0 \le m < l \le k-1$에 대하여 B_i에 포함되어 있는 원소들 중 tk(k-1)/2개의 서로 다른 원소들 간 차 ($D = (b_{il} - b_{im}) \bmod v$)들의 집합이 $\{1, 2, \ldots, (v-1)/2\}$가 된다면 그 순환 차족은 (v, k, 1) 파라미터를 갖는 완전 차족이라 정의할 수 있다.

[0102] 도 3은 본 발명의 일실시예에 따른 순환 차족을 활용한 데이터 분산 저장 방법을 도시한 순서도이다.

[0103] 단계(S31)에서는 N(N은 N=tv의 조건을 만족하는 자연수이고, v는 v=k(k-1)t+1의 조건을 만족하는 자연수이며, k는 2이상의 자연수이고, t는 1이상의 자연수임)개의 메시지들에 대해 v개의 데이터 저장소들로의 분산 저장 명령이 인가되면, 상기 v개의 데이터 저장소들에 대응하는 v개의 메시지 분산 그룹들을 생성한다.

[0104] 단계(S32)에서는 상기 v개의 메시지 분산 그룹들 각각에 대해 상기 N개의 메시지들을 tk(tk는 t와 k의 곱을 의미함)개씩 분산 할당(상기 v개의 메시지 분산 그룹들에 분산 할당되는 메시지들은 서로 중복될 수 있음)한다.

[0105] 단계(S33)에서는 상기 v개의 메시지 분산 그룹들 각각에 할당되어 있는 tk개의 메시지들을 상기 v개의 메시지 분산 그룹들 별로, 상기 v개의 데이터 저장소들 중 각 메시지 분산 그룹에 대응되는 데이터 저장소에 분리해서 저장한다.

[0106] 이때, 본 발명의 일실시예에 따르면, 단계(S32)에서는 "0"부터 "v-1"까지의 정수들을 원소로 포함하고 있는 집합인 Z_v(상기 Z_v에 포함되어 있는 원소들의 개수는 v개임)에 대한 k개의 원소를 갖는 복수의 부분 집합들 중, 상기 복수의 부분 집합들 각각에 포함되어 있는 k개의 원소들 중에서 서로 다른 두 원소 간의 모든 차들 각각이 상기 Z_v에 포함되어 있는 원소들에서 "0"을 제외한 서로 다른 1개의 원소와만 일치하는 것으로 확인되는 t개의 부분 집합들(상기 t개의 부분 집합들의 모임은 (v, k, 1)의 파라미터를 갖는 순환 차족으로 정의됨)을 확인하는 단계, 상기 t개의 부분 집합들 각각에 대해서, 상기 t개의 부분 집합들 각각에 포함되어 있는 k개의 원소들에 기초하여 상기 Z_v에 포함되어 있는 "0"부터 "v-1"까지의 정수들에 대해 "0"부터 "v-1" 순으로 "1"과 "0"의 코드 값을 할당함으로써, v개의 "1"과 "0"의 코드 값으로 구성된 t개의 부분 시퀀스들을 생성하고, 상기 t개의 부분 시퀀스들을 조합하여 총 N개의 "1"과 "0"의 코드 값으로 구성된 제1 시퀀스(N=tv이며, 상기 t개의 부분 집합들 각각에 대해서 v개씩 "1"과 "0"의 코드 값을 할당함으로써 생성한 상기 t개의 부분 시퀀스들을 조합하여 상기 제1 시퀀스를 생성하였으므로, 상기 제1 시퀀스를 구성하는 "1"과 "0"의 코드 값은 총 N개가 됨)를 생성하는 단계, 상기 v개의 "1"과 "0"의 코드 값으로 구성된 상기 t개의 부분 시퀀스들 각각에 대해 선정된 방향으로 순환 시프트 연산을 v-1회 수행하면서, 각 순환시프트 연산이 수행될 때마다 순환시프트 연산이 완료된 상기 t개의 부분 시퀀스들을 조합하여 제2 시퀀스를 생성함으로써, 총 v-1개의 제2 시퀀스들을 생성하는 단계, 상기 v개의 메시지 분산 그룹들 각각에 대해 상기 제1 시퀀스와 상기 v-1개의 제2 시퀀스들을 하나씩 매칭시키는 단계 및 상기 v개의 메시지 분산 그룹들 각각에 대해, 상기 v개의 메시지 분산 그룹들 각각에 매칭되어 있는 각 시퀀스를 구성하는 N개의 코드 값에 기초하여 상기 N개의 메시지들을 tk개씩 분산 할당하는 단계를 포함할 수 있다.

[0107] 이때, 본 발명의 일실시예에 따르면, 상기 제1 시퀀스를 생성하는 단계는 상기 t개의 부분 집합들 각각에 대해서, 상기 Z_v에 포함되어 있는 "0"부터 "v-1"까지의 정수들 중 상기 t개의 부분 집합들 각각에 포함되어 있는 k개의 원소들과 일치하는 정수들에 대해 "1"의 코드 값을 할당하고, 상기 t개의 부분 집합들 각각에 포함되어 있는 k개의 원소들과 일치하지 않는 나머지 정수들에 대해 "0"의 코드 값을 할당함으로써, 상기 v개의 "1"과 "0"의 코드 값으로 구성된 상기 t개의 부분 시퀀스들을 생성할 수 있다.

[0108] 또한, 본 발명의 일실시예에 따르면, 단계(S32)에서는 상기 N개의 메시지들 각각에 대해 순번 정보와 연관된 인덱스 값을 할당하는 단계를 더 포함할 수 있다.

[0109] 이때, 상기 N개의 코드 값에 기초하여 상기 N개의 메시지들을 tk개씩 분산 할당하는 단계는 상기 v개의 메시지 분산 그룹들 각각에 대해, 상기 v개의 메시지 분산 그룹들 각각에 매칭되어 있는 각 시퀀스를 구성하는 N개의 코드 값의 배열 순서와 상기 N개의 메시지들 각각에 할당되어 있는 인덱스 값을 비교한 후 상기 N개의 메시지들에 할당되어 있는 인덱스 값과 일치하는 배열 순서의 코드 값에 기초하여 상기 N개의 메시지들을 tk개씩 분산 할당할 수 있다.

[0110] 이때, 본 발명의 일실시예에 따르면, 상기 N개의 코드 값에 기초하여 상기 N개의 메시지들을 tk개씩 분산 할당하는 단계는 상기 v개의 메시지 분산 그룹들 각각에 대해, 상기 v개의 메시지 분산 그룹들 각각에 매칭되어 있는 각 시퀀스를 구성하는 N개의 코드 값 중 tk개씩 존재하는 "1"의 코드 값을 확인한 후, 상기 v개의 메시지 분산 그룹들 각각에 대해, 상기 N개의 메시지들 중 상기 "1"의 코드 값에 대한 배열 순서와 일치하는 인덱스 값이 할당되어 있는 tk개의 메시지들을 분산 할당할 수 있다.

[0111] 또한, 본 발명의 일실시예에 따르면, 순환 차족을 활용한 데이터 분산 저장 방법은 상기 v개의 데이터 저장소들에 상기 N개의 메시지들이 tk개씩 분산 저장된 이후, 상기 N개의 메시지들에 대한 로드 명령이 인가되면, 상기 로드 명령에 기초하여 상기 v개의 데이터 저장소들 중 tk개의 데이터 저장소들을 랜덤하게 선택하고, 상기 tk개

의 데이터 저장소들 각각으로부터 상기 tk개의 데이터 저장소들 각각에 저장되어 있는 tk개의 저장 메시지들을 추출하는 단계 및 상기 tk개의 데이터 저장소들 각각으로부터 tk개씩 추출된 저장 메시지들을 상기 N개의 메시지들에 대한 로드 명령에 따른 데이터로 로드하는 단계를 더 포함할 수 있다.

[0112] 이때, 본 발명의 일실시예에 따르면, 상기 N개의 메시지들은 일부 메시지의 손실이 발생할 경우, 나머지 메시지들을 기초로 손실 메시지들의 복구가 가능한 MDS 코드이고, 상기 로드하는 단계는 상기 tk개의 데이터 저장소들 각각으로부터 tk개씩 저장 메시지들이 추출되면, 상기 추출된 저장 메시지들에 할당되어 있는 인덱스 값과 상기 N개의 메시지들에 할당된 인덱스 값을 비교하여 상기 N개의 메시지들 중 상기 추출된 저장 메시지들에 포함되어 있지 않은 잔여 메시지가 존재하는 것으로 확인되면, 상기 추출된 저장 메시지들을 기초로 상기 잔여 메시지를 복구하여 상기 추출된 저장 메시지들과 상기 복구된 잔여 메시지를 상기 N개의 메시지들에 대한 로드 명령에 따른 데이터로 로드할 수 있다.

[0113] 이상, 도 3을 참조하여 본 발명의 일실시예에 따른 순환 차족을 활용한 데이터 분산 저장 방법에 대해 설명하였다. 여기서, 본 발명의 일실시예에 따른 순환 차족을 활용한 데이터 분산 저장 방법은 도 1과 도 2를 이용하여 설명한 순환 차족을 활용한 데이터 분산 저장 장치(110)의 동작에 대한 구성과 대응될 수 있으므로, 이에 대한 보다 상세한 설명은 생략하기로 한다.

[0114] 본 발명의 일실시예에 따른 순환 차족을 활용한 데이터 분산 저장 방법은 컴퓨터와의 결합을 통해 실행시키기 위한 저장매체에 저장된 컴퓨터 프로그램으로 구현될 수 있다.

[0115] 또한, 본 발명의 일실시예에 따른 순환 차족을 활용한 데이터 분산 저장 방법은 다양한 컴퓨터 수단을 통하여 수행될 수 있는 프로그램 명령 형태로 구현되어 컴퓨터 판독 가능 매체에 기록될 수 있다. 상기 컴퓨터 판독 가능 매체는 프로그램 명령, 데이터 파일, 데이터 구조 등을 단독으로 또는 조합하여 포함할 수 있다. 상기 매체에 기록되는 프로그램 명령은 본 발명을 위하여 특별히 설계되고 구성된 것들이거나 컴퓨터 소프트웨어 당업자에게 공지되어 사용 가능한 것일 수도 있다. 컴퓨터 판독 가능 기록 매체의 예에는 하드 디스크, 플로피 디스크 및 자기 테이프와 같은 자기 매체(magnetic media), CD-ROM, DVD와 같은 광기록 매체(optical media), 플롭티컬 디스크(floptical disk)와 같은 자기-광 매체(magneto-optical media), 및 롬(ROM), 램(RAM), 플래시 메모리 등과 같은 프로그램 명령을 저장하고 수행하도록 특별히 구성된 하드웨어 장치가 포함된다. 프로그램 명령의 예에는 컴파일러에 의해 만들어지는 것과 같은 기계어 코드뿐만 아니라 인터프리터 등을 사용해서 컴퓨터에 의해서 실행될 수 있는 고급 언어 코드를 포함한다. 상기된 하드웨어 장치는 본 발명의 동작을 수행하기 위해 하나 이상의 소프트웨어 모듈로서 작동하도록 구성될 수 있으며, 그 역도 마찬가지이다.

[0116] 이상과 같이 본 발명에서는 구체적인 구성 요소 등과 같은 특정 사항들과 한정된 실시예 및 도면에 의해 설명되었으나 이는 본 발명의 보다 전반적인 이해를 돕기 위해서 제공된 것일 뿐, 본 발명은 상기의 실시예에 한정되는 것은 아니며, 본 발명이 속하는 분야에서 통상적인 지식을 가진 자라면 이러한 기재로부터 다양한 수정 및 변형이 가능하다.

[0117] 따라서, 본 발명의 사상은 설명된 실시예에 국한되어 정해져서는 아니되며, 후술하는 특허청구범위뿐 아니라 이 특허청구범위와 균등하거나 등가적 변형이 있는 모든 것들은 본 발명 사상의 범주에 속한다고 할 것이다.

부호의 설명

[0118] 110: 순환 차족을 활용한 데이터 분산 저장 장치

111: 분산 그룹 생성부 112: 메시지 분산 할당부

113: 데이터 저장부 114: 순환 차족 확인부

115: 제1 시퀀스 생성부 116: 제2 시퀀스 생성부

117: 시퀀스 매칭부 118: 할당부

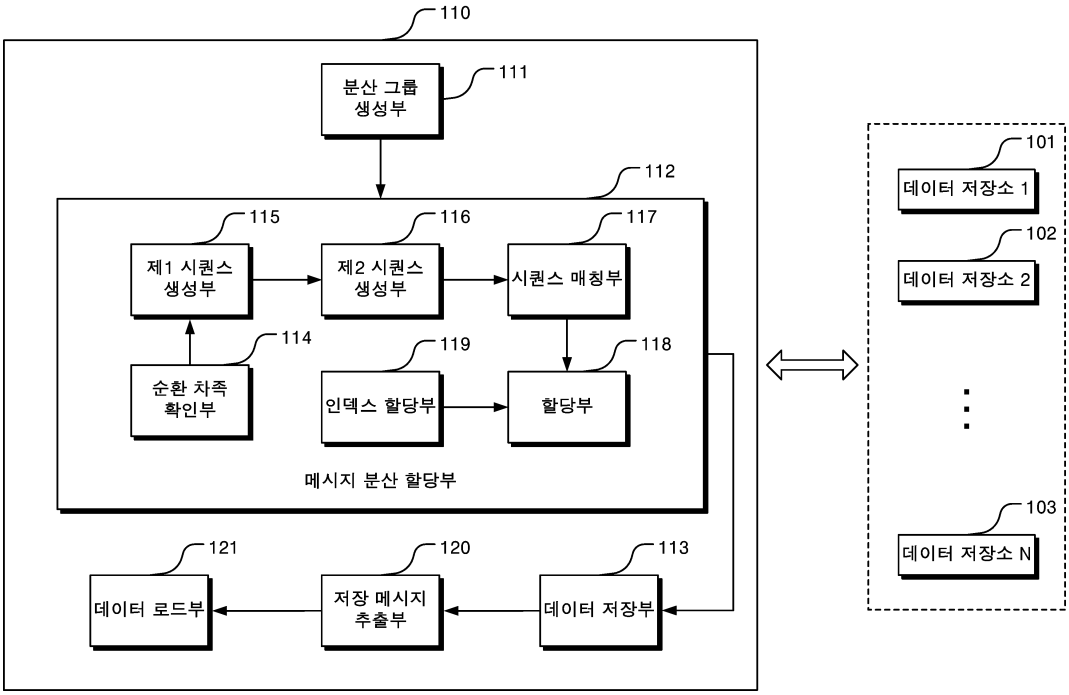
119: 인덱스 할당부 120: 저장 메시지 추출부

121: 데이터 로드부

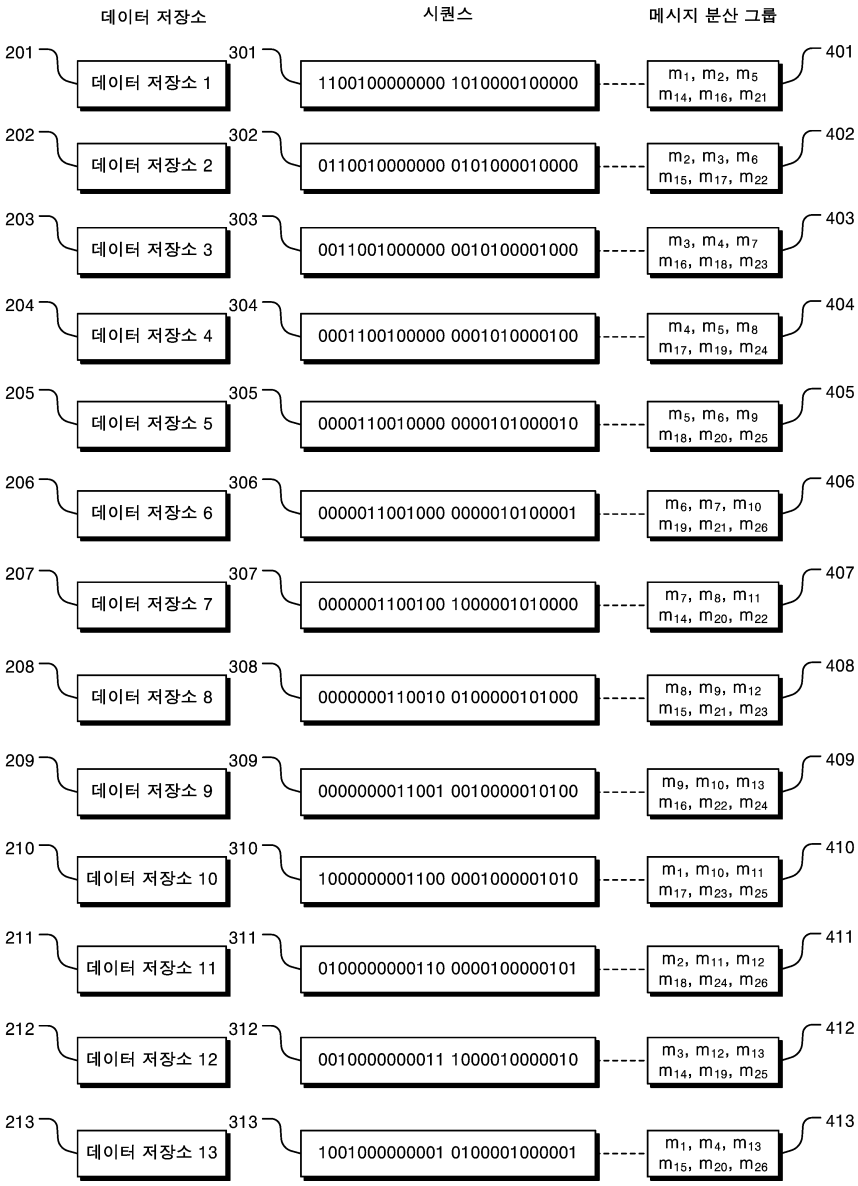
101, 102, 103: v개의 데이터 저장소들

도면

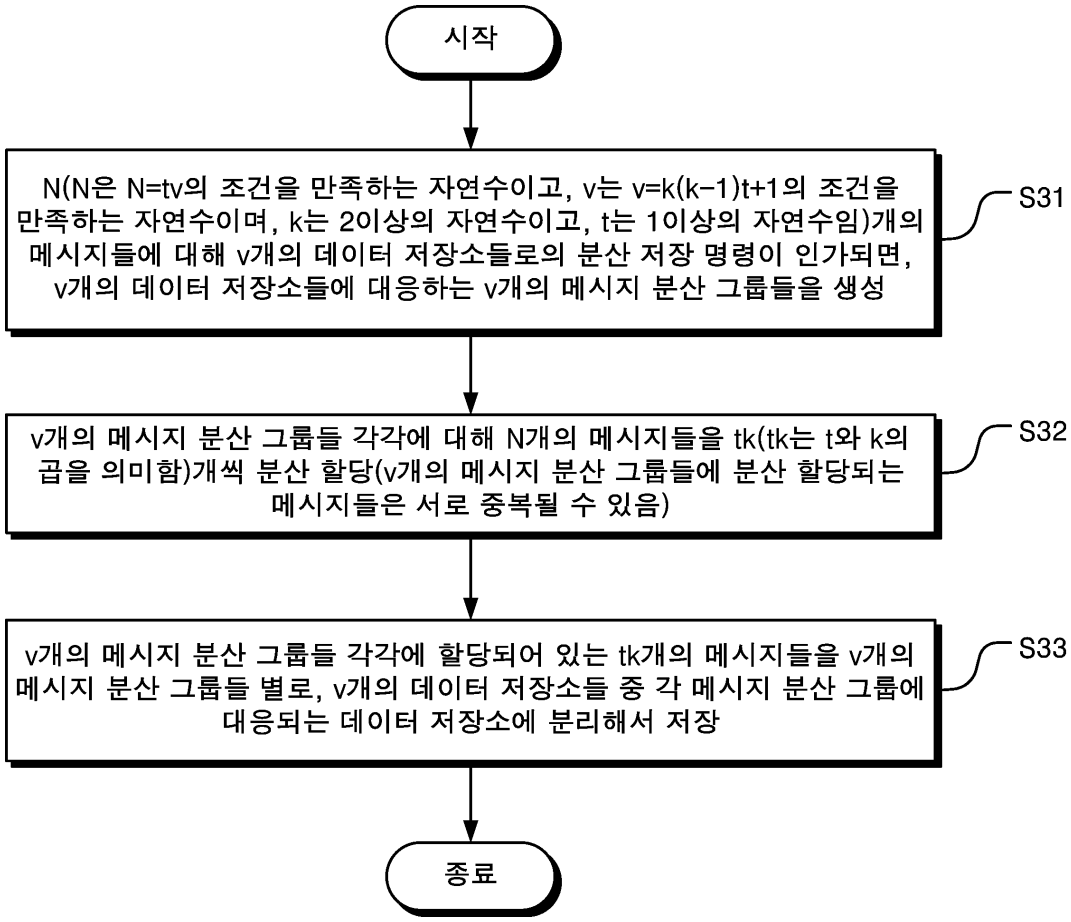
도면1


110
분산 그룹 생성부
111
112
115
제1 시퀀스 생성부
116
제2 시퀀스 생성부
117
시퀀스 매칭부
114
순환 차족 확인부
119
인덱스 할당부
118
할당부
메시지 분산 할당부
121
데이터 로드부
120
저장 메시지 추출부
113
데이터 저장부
101
데이터 저장소 1
102
데이터 저장소 2
103
데이터 저장소 N

도면2


데이터 저장소
시퀀스
메시지 분산 그룹
201
데이터 저장소 1
301
1100100000000 1010000100000
401
m1, m2, m5
m14, m16, m21
202
데이터 저장소 2
302
0110010000000 0101000010000
402
m2, m3, m6
m15, m17, m22
203
데이터 저장소 3
303
0011001000000 0010100001000
403
m3, m4, m7
m16, m18, m23
204
데이터 저장소 4
304
0001100100000 0001010000100
404
m4, m5, m8
m17, m19, m24
205
데이터 저장소 5
305
0000110010000 0000101000010
405
m5, m6, m9
m18, m20, m25
206
데이터 저장소 6
306
0000011001000 0000010100001
406
m6, m7, m10
m19, m21, m26
207
데이터 저장소 7
307
0000001100100 1000001010000
407
m7, m8, m11
m14, m20, m22
208
데이터 저장소 8
308
0000000110010 0100000101000
408
m8, m9, m12
m15, m21, m23
209
데이터 저장소 9
309
0000000011001 0010000010100
409
m9, m10, m13
m16, m22, m24
210
데이터 저장소 10
310
1000000001100 0001000001010
410
m1, m10, m11
m17, m23, m25
211
데이터 저장소 11
311
0100000000110 0000100000101
411
m2, m11, m12
m18, m24, m26
212
데이터 저장소 12
312
0010000000011 1000010000010
412
m3, m12, m13
m14, m19, m25
213
데이터 저장소 13
313
1001000000001 0100001000001
413
m1, m4, m13
m15, m20, m26

도면3


시작
N(N은 N=tv의 조건을 만족하는 자연수이고, v는 v=k(k-1)t+1의 조건을 만족하는 자연수이며, k는 2이상의 자연수이고, t는 1이상의 자연수임)개의 메시지들에 대해 v개의 데이터 저장소들로의 분산 저장 명령이 인가되면, v개의 데이터 저장소들에 대응하는 v개의 메시지 분산 그룹들을 생성
S31
v개의 메시지 분산 그룹들 각각에 대해 N개의 메시지들을 tk(tk는 t와 k의 곱을 의미함)개씩 분산 할당(v개의 메시지 분산 그룹들에 분산 할당되는 메시지들은 서로 중복될 수 있음)
S32
v개의 메시지 분산 그룹들 각각에 할당되어 있는 tk개의 메시지들을 v개의 메시지 분산 그룹들 별로, v개의 데이터 저장소들 중 각 메시지 분산 그룹에 대응되는 데이터 저장소에 분리해서 저장
S33
종료