



(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2015-0061959
(43) 공개일자 2015년06월05일

(51) 국제특허분류(Int. Cl.)

G06F 17/30 (2006.01) G06F 17/40 (2006.01)

(21) 출원번호 10-2013-0146386

(22) 출원일자 2013년11월28일

심사청구일자 2013년11월28일

(71) 출원인

고려대학교 산학협력단

서울특별시 성북구 안암로 145, 고려대학교 (안암동5가)

(72) 발명자

정연돈

서울 중구 청계천로 400, 105동 812호 (황학동, 롯데캐슬베네치아)

손지훈

서울 도봉구 해동로 242-11, 102동 603호 (쌍문동, 성원아파트)

(74) 대리인

송인호, 민영준, 최관락

전체 청구항 수 : 총 18 항

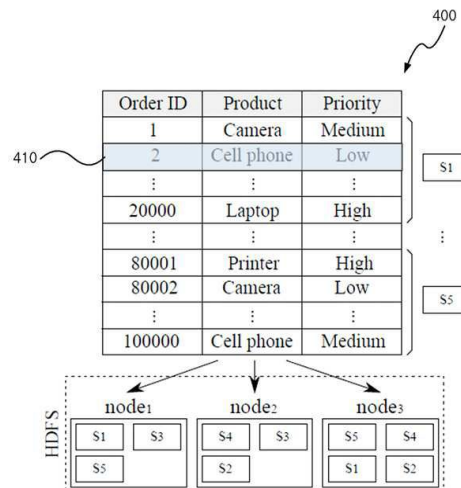
(54) 발명의 명칭 컬럼 기반 데이터 처리 방법 및 장치

(57) 요약

본 발명은 컬럼 기반 데이터 처리 방법 및 장치에 관한 것이다.

본 발명의 일실시예에 의한 컬럼 기반 데이터 처리 방법은 m 개의 컬럼을 포함하는 입력 데이터를 수신하는 단계-상기 m은 2 이상의 자연수임-; 상기 입력 데이터를 복수 개의 샤드(shard)로 수평적으로 분할하는 단계-상기 샤드는 상기 m 개의 컬럼 파일을 포함하고, 상기 컬럼 파일의 크기는 태스크의 기본 단위를 고려하여 결정됨-; 및 상기 복수 개의 샤드를 n 개의 노드에 분산시켜 저장하는 단계-상기 n은 2 이상의 자연수임-를 포함할 수 있다.

대표도 - 도4



이 발명을 지원한 국가연구개발사업

과제고유번호 C00128220200394121

부처명 중소기업청

연구관리전문기관 중소기업산학협력센터

연구사업명 산학연공동기술개발사업

연구과제명 (2차년도)대화형 질의가 가능한 빅데이터 기반의 웹로그 분석기 개발

기 여 율 1/1

주관기관 고려대학교 산학협력단

연구기간 2013.06.01 ~ 2014.05.31

명세서

청구범위

청구항 1

m 개의 컬럼을 포함하는 입력 데이터를 수신하는 단계-상기 m은 2 이상의 자연수임-;

상기 입력 데이터를 복수 개의 샤드(shard)로 수평적으로 분할하는 단계- 상기 샤드는 상기 m 개의 컬럼 파일을 포함하고, 상기 컬럼 파일의 크기는 태스크의 기본 단위를 고려하여 결정됨-; 및

상기 복수 개의 샤드를 n 개의 노드에 분산시켜 저장하는 단계- 상기 n은 2 이상의 자연수임-를 포함하는 것을 특징으로 하는 컬럼 기반 데이터 처리 방법.

청구항 2

제 1 항에 있어서, 상기 복수 개의 샤드를 노드에 저장하는 단계는

상기 복수 개의 샤드를 각각 복제하는 단계;

상기 복제된 샤드를 상기 n 개의 노드에 분산시켜 저장하는 단계를 포함하는 것을 특징으로 하는 컬럼 기반 데이터 처리 방법.

청구항 3

제 1 항에 있어서, 상기 태스크의 기본 단위는

하둡 분산 파일 시스템(Hadoop Distributed File System) 블록인 것을 특징으로 하는 컬럼 기반 데이터 처리 방법.

청구항 4

제 3 항에 있어서, 상기 입력 데이터를 복수 개의 샤드(shard)로 수평적으로 분할하는 단계는

상기 입력 데이터에 포함된 m 개의 컬럼 중 데이터의 크기가 가장 큰 컬럼을 베이스 컬럼으로 선택하는 단계; 및

상기 베이스 컬럼을 기준으로 상기 입력 데이터를 복수 개의 샤드(shard)로 수평적으로 분할하는 단계를 포함하는 것을 특징으로 하는 컬럼 기반 데이터 처리 방법.

청구항 5

제 3 항에 있어서, 상기 베이스 컬럼의 컬럼 파일의 크기는

상기 하둡 분산 파일 시스템 블록의 크기를 초과하지 않으면서 최대값을 갖도록 설정되는 것을 특징으로 하는 컬럼 기반 데이터 처리 방법.

청구항 6

제 3 항에 있어서, 상기 컬럼 기반 데이터 처리 방법은

사용자 쿼리를 수신하는 단계; 및

상기 사용자 쿼리에 근거하여 스플릿(split)의 크기를 조정하는 단계- 상기 스플릿은 태스크에서 처리하는 데이터를 의미함-를 더 포함하는 것을 특징으로 하는 컬럼 기반 데이터 처리 방법.

청구항 7

제 6 항에 있어서, 상기 스플릿의 크기는

하기 수학적 식 1에 의해 결정되는 것을 특징으로 하는 컬럼 기반 데이터 처리 방법.

[수학식 1]

$$T = \theta \times \frac{\sum |c_r|}{\sum |c_q|} (c_r \in col(R), c_q \in col(Q))$$

(T는 스플릿의 크기이고, θ 는 하둡 분산 파일 시스템 블록의 크기이고, R은 입력 데이터이고, Q는 쿼리이고, c_r 은 입력 데이터에 포함된 컬럼의 크기이고, c_q 는 쿼리에 포함된 컬럼의 크기이다.)

청구항 8

제 3 항에 있어서, 상기 컬럼 파일은

데이터를 포함하고, 청크(chunk) 단위로 나누어진 복수 개의 벡터- 상기 청크는 하둡 분산 파일 시스템에서 한번에 읽어 들일 수 있는 데이터 크기를 의미함-;

시작 레코드의 식별자 및 벡터의 저장 위치에 대한 정보인 오프셋을 포함하는 인덱스; 및

메타 데이터를 포함하는 헤드를 포함하는 것을 특징으로 하는 컬럼 기반 데이터 처리 방법.

청구항 9

제 3 항에 있어서, 상기 입력 데이터에 포함된 레코드는

동일 노드에 저장되는 것을 특징으로 하는 컬럼 기반 데이터 처리 방법.

청구항 10

마스터 처리 장치 및 n 개의 노드를 포함하는 컬럼 기반 데이터 처리 장치에 관한 것으로서-상기 n은 2 이상의 자연수임-,

상기 마스터 처리 장치는

m 개의 컬럼을 포함하는 입력 데이터를 수신하는 데이터 수신부- 상기 m은 2 이상의 자연수임-;

상기 입력 데이터를 복수 개의 샤드(shard)로 수평적으로 분할하는 샤드 생성부- 상기 샤드는 상기 m 개의 컬럼 파일을 포함하고, 상기 컬럼 파일의 크기는 태스크의 기본 단위를 고려하여 결정됨-;

상기 복수 개의 샤드를 상기 n 개의 노드에 분산시켜 저장하는 샤드 할당부; 및

상기 데이터 수신부, 상기 샤드 생성부, 및 상기 샤드 할당부를 제어하는 제어부를 포함하는 것을 특징으로 하는 컬럼 기반 데이터 처리 장치.

청구항 11

제 10 항에 있어서, 상기 샤드 할당부는

상기 복수 개의 샤드를 각각 복제하고, 상기 복제된 샤드를 상기 n 개의 노드에 분산시켜 저장하는 것을 특징으로 하는 컬럼 기반 데이터 처리 장치.

청구항 12

제 10 항에 있어서, 상기 태스크의 기본 단위는

하둡 분산 파일 시스템(Hadoop Distributed File System) 블록인 것을 특징으로 하는 컬럼 기반 데이터 처리 장치.

청구항 13

제 12 항에 있어서, 상기 샤드 생성부는

상기 입력 데이터에 포함된 m 개의 컬럼 중 데이터의 크기가 가장 큰 컬럼을 베이스 컬럼으로 선택하고, 상기

베이스 컬럼을 기준으로 상기 입력 데이터를 복수 개의 샤드(shard)로 수평적으로 분할하는 것을 특징으로 하는 컬럼 기반 데이터 처리 장치.

청구항 14

제 12 항에 있어서, 상기 베이스 컬럼의 컬럼 파일의 크기는

상기 하둡 분산 파일 시스템 블록의 크기를 초과하지 않으면서 최대값을 갖도록 설정되는 것을 특징으로 하는 컬럼 기반 데이터 처리 장치.

청구항 15

제 12 항에 있어서, 상기 마스터 처리 장치는

사용자 쿼리를 수신하는 쿼리 수신부; 및

상기 사용자 쿼리에 근거하여 스플릿(split)의 크기를 조정하는 스플릿 생성부- 상기 스플릿은 태스크에서 처리하는 데이터를 의미함-를 더 포함하는 것을 특징으로 하는 컬럼 기반 데이터 처리 장치.

청구항 16

제 15 항에 있어서, 상기 스플릿의 크기는

하기 수학적 식 1에 의해 결정되는 것을 특징으로 하는 컬럼 기반 데이터 처리 장치.

[수학적 식 1]

$$T = \theta \times \frac{\sum |c_r|}{\sum |c_q|} (c_r \in col(R), c_q \in col(Q))$$

(T는 스플릿의 크기이고, θ 는 하둡 분산 파일 시스템 블록의 크기이고, R은 입력 데이터이고, Q는 쿼리이고, C_r 은 입력 데이터에 포함된 컬럼의 크기이고, C_q 는 쿼리에 포함된 컬럼의 크기이다.)

청구항 17

제 12 항에 있어서, 상기 컬럼 파일은

데이터를 포함하고, 청크(chunk) 단위로 나누어진 복수 개의 벡터- 상기 청크는 하둡 분산 파일 시스템에서 한 번에 읽어 들일 수 있는 데이터 크기를 의미함-;

시작 레코드의 식별자 및 벡터의 저장 위치에 대한 정보인 오프셋을 포함하는 인덱스; 및

메타 데이터를 포함하는 헤드를 포함하는 것을 특징으로 하는 컬럼 기반 데이터 처리 장치.

청구항 18

제 12 항에 있어서, 상기 입력 데이터에 포함된 레코드는

동일 노드에 저장되는 것을 특징으로 하는 컬럼 기반 데이터 처리 장치.

발명의 설명

기술 분야

[0001]

본 발명은 데이터 처리 방법 및 데이터 처리 장치에 관한 것으로, 보다 상세하게는 컬럼 기반 데이터 처리 방법 및 장치에 관한 것이다.

배경 기술

[0002]

인터넷이 발전함에 따라 하루에도 수없이 많은 데이터가 인터넷 상에서 생성 및 유통이 되고 있으며, 최근 많은

기업, 특히 검색 엔진 회사 및 웹 포탈들 간에는 이와 같이 엄청난 양의 데이터를 가능한 많이 수집 및 축적하고 수집된 데이터를 처리하여 가능한 빨리 의미 있는 정보를 추출하는 능력이 기업의 경쟁력이 되고 있다.

[0003] 이를 위하여, 현재 많은 기업에서 저비용으로 대규모 클러스터를 구축하여 대용량 데이터 분산 관리 및 작업 분산 처리하는 기술에 대하여 많은 연구를 하고 있으며, 작업 분산 병렬 처리 기술 중에서 하둡 분산 파일 시스템(HDFS, Hadoop Distributed File System)이 대표적인 작업 분산 병렬 처리 방법 중에 하나로 주목을 받고 있다.

[0004] 한편, 최근에는 컬럼 기반 데이터베이스 시스템(column-based database system)에 대해 상당한 연구가 이루어지고 있다. 기존에는 데이터를 릴레이션(relation)의 형태로 저장할 때 통상적으로 열(row) 형태로 묶어 저장한 반면, 컬럼 기반 데이터베이스 시스템은 데이터를 저장할 때 같은 도메인(domain)을 취하는 데이터를 묶어 릴레이션(relation)의 컬럼(column) 단위로 저장한다.

[0005] 대한민국 등록특허 제10-1200773호에는 가변 길이의 데이터 레코드 블록을 갖는 바이너리 패킷 데이터를 하둡 분산처리 시스템에서 효과적으로 처리할 수 있도록 하는 입력포맷 추출방법이 개시된다.

[0006] 하지만, 등록특허 제10-1200773호에는 분산 처리의 오버헤드(overhead)를 감소시키는 기술에 대해서는 개시되어 있지 않다.

[0007] 따라서 분산 처리의 오버헤드를 효과적으로 감소시키는 기술에 대한 연구가 필요한 실정이다.

발명의 내용

해결하려는 과제

[0008] 본 발명의 목적은 태스크(task) 수를 줄여 분산 처리 시의 오버헤드(overhead)를 감소시킬 수 있는 컬럼 기반 데이터 처리 방법 및 장치를 제공하는 데 있다.

[0009] 본 발명의 목적은 컬럼 기반 데이터 처리의 최적화 기법을 지원할 수 있는 컬럼 기반 데이터 처리 방법 및 장치를 제공하는 데 있다.

과제의 해결 수단

[0010] 상기 목적을 달성하기 위해 본 발명의 일실시예에 의하면, m 개의 컬럼을 포함하는 입력 데이터를 수신하는 단계-상기 m은 2 이상의 자연수임-; 상기 입력 데이터를 복수 개의 샤드(shard)로 수평적으로 분할하는 단계-상기 샤드는 상기 m 개의 컬럼 파일을 포함하고, 상기 컬럼 파일의 크기는 태스크의 기본 단위를 고려하여 결정됨-; 및 상기 복수 개의 샤드를 n 개의 노드에 분산시켜 저장하는 단계-상기 n은 2 이상의 자연수임-를 포함하는 컬럼 기반 데이터 처리 방법이 제공된다.

[0011] 상기 목적을 달성하기 위해 본 발명의 일실시예에 의하면, 마스터 처리 장치 및 n 개의 노드를 포함하는 컬럼 기반 데이터 처리 장치에 관한 것으로서-상기 n은 2 이상의 자연수임-, 상기 마스터 처리 장치는 m 개의 컬럼을 포함하는 입력 데이터를 수신하는 데이터 수신부-상기 m은 2 이상의 자연수임-; 상기 입력 데이터를 복수 개의 샤드(shard)로 수평적으로 분할하는 샤드 생성부-상기 샤드는 상기 m 개의 컬럼 파일을 포함하고, 상기 컬럼 파일의 크기는 태스크의 기본 단위를 고려하여 결정됨-; 상기 복수 개의 샤드를 상기 n 개의 노드에 분산시켜 저장하는 샤드 할당부; 및 상기 데이터 수신부, 상기 샤드 생성부, 및 상기 샤드 할당부를 제어하는 제어부를 포함하는 컬럼 기반 데이터 처리 장치가 제공된다.

발명의 효과

[0012] 본 발명의 일실시예에 의한 컬럼 기반 데이터 처리 방법 및 장치는 태스크 수를 줄임으로써, 오버헤드를 감소시킬 수 있다.

[0013] 본 발명의 일실시예에 의하면, 컬럼 기반 데이터 처리의 최적화 기법을 지원할 수 있다.

도면의 간단한 설명

[0014] 도 1은 본 발명의 일실시예와 관련된 컬럼 기반 데이터 처리 장치의 블록도이다.

도 2는 도 1의 마스터 처리 장치의 블록도이다.

도 3은 본 발명의 일실시예와 관련된 컬럼 기반 데이터 처리 방법을 나타내는 흐름도이다.

도 4는 본 발명의 일실시예와 관련된 컬럼 기반 데이터 처리 방법에서 입력 데이터를 샤드(shard)로 분할하는 예를 나타내는 도면이다.

도 5는 본 발명의 일실시예와 관련된 컬럼 기반 데이터 처리 방법에서 컬럼 파일의 크기를 결정하는 예를 설명하기 위한 도면이다.

도 6은 본 발명의 일실시예와 관련된 컬럼 파일의 구조를 나타내는 도면이다.

도 7 내지 도 8은 본 발명의 일실시예와 관련된 컬럼 기반 데이터 처리 방법에서 쿼리에 따라 스플릿의 크기가 조정되는 예를 나타내는 도면이다.

도 9은 본 발명의 일실시예와 관련된 컬럼 기반 데이터 처리 방법과 다른 데이터 처리 방법의 성능을 비교한 그래프이다.

발명을 실시하기 위한 구체적인 내용

[0015] 이하, 본 발명의 일실시예와 관련된 컬럼 기반 데이터 처리 방법 및 장치에 대해 도면을 참조하여 설명하도록 하겠다.

[0016] 본 명세서에서 사용되는 단수의 표현은 문맥상 명백하게 다르게 뜻하지 않는 한, 복수의 표현을 포함한다. 본 명세서에서, "구성된다" 또는 "포함한다" 등의 용어는 명세서상에 기재된 여러 구성 요소들, 또는 여러 단계들을 반드시 모두 포함하는 것으로 해석되지 않아야 하며, 그 중 일부 구성 요소들 또는 일부 단계들은 포함되지 않을 수도 있고, 또는 추가적인 구성 요소 또는 단계들을 더 포함할 수 있는 것으로 해석되어야 한다.

[0017] 도 1은 본 발명의 일실시예와 관련된 컬럼 기반 데이터 처리 장치의 블록도이다.

[0018] 도시된 바와 같이, 상기 컬럼 기반 데이터 처리 장치(100)는 마스터 처리 장치(200) 및 n 개의 노드(300)를 포함할 수 있다. 상기 마스터 처리 장치(200)는 m 개의 컬럼(column)을 포함한 입력 데이터(예: 테이블 형태로 구성된 데이터)를 복수 개의 샤드(shard)로 분할하고, 분할된 복수 개의 샤드를 n 개의 노드(300)에 분산시킬 수 있다. 이 경우, 상기 마스터 처리 장치(200)는 데이터 손실을 막기 위해 상기 복수 개의 샤드를 복제하여 복제된 복수 개의 샤드도 n 개의 노드(300)에 분산시킬 수 있다. 샤드에 대한 자세한 설명은 후술하도록 하겠다.

[0019] 상기 n 개의 노드(300)는 분산되어 저장된 데이터를 처리하는 서버 컴퓨터라 할 수 있다. 상기 n 개의 노드(300)는 마스터 처리 장치(200)의 제어에 의해 데이터 압축(compression), 데이터 쓰기(Data Writing), 데이터 읽기(Data Reading) 등을 수행할 수 있다.

[0020] 도 2는 도 1의 마스터 처리 장치의 블록도이다.

[0021] 도시된 바와 같이, 마스터 처리 장치(200)는 데이터 수신부(210), 샤드 생성부(220), 샤드 할당부(230), 쿼리 수신부(240), 스플릿 생성부(250), 및 제어부(260)를 포함할 수 있다.

[0022] 데이터 수신부(210)는 m 개의 컬럼(column)을 포함한 입력 데이터를 수신할 수 있다. 예를 들어, 상기 입력 데이터는 테이블 형태로 구성될 수 있다. 상기 m 은 2 이상의 자연수이다.

[0023] 샤드 생성부(220)는 상기 입력 데이터를 복수 개의 샤드(shard)로 수평적으로 분할할 수 있다. 본 명세서에서 수평적으로 분할한다는 의미는 테이블 형태로 구성된 입력 데이터를 가로 방향(즉, 행(row) 방향)으로 분할한다는 것을 의미한다. 또한, 본 명세서에서 수직적으로 분할한다는 의미는 테이블 형태로 구성된 입력 데이터를 세로 방향(즉, 컬럼 방향)으로 분할한다는 것을 의미한다. 상기 샤드는 m 개의 컬럼 파일을 포함할 수 있다. 컬럼 파일의 크기는 태스크의 기본 단위를 고려하여 결정될 수 있다. 예를 들어, 상기 태스크의 기본 단위는 하둡 분산 파일 시스템(HDFS, Hadoop Distributed File System) 블록을 포함할 수 있다.

[0024] 샤드 할당부(230)는 상기 복수 개의 샤드를 n 개의 노드(300)에 분산시켜 할당할 수 있다. 또한, 샤드 할당부(230)는 상기 복수 개의 샤드를 복제하고, 복제된 샤드도 n 개의 노드(300)에 분산시켜 할당될 수 있다. 할당된 샤드는 각각 해당 노드에 저장된다. 이 경우, 원본 샤드와 복제본 샤드는 다른 노드에 할당되어 저장될 수 있다. 샤드를 복제하여 n 개의 노드(300)에 분산시키는 이유는 서버 또는 하드디스크 고장으로 인해 데이터가 손실되는 것을 막기 위함이다.

[0025] 쿼리 수신부(240)는 사용자로부터 쿼리를 수신할 수 있다. 입력 데이터에 포함된 m 개의 컬럼 모두 상기 쿼리에

관련될 수도 있지만, m 개의 컬럼 중 일부만이 쿼리에 관련될 수도 있다.

- [0026] 스플릿 생성부(250)는 상기 쿼리에 근거하여 스플릿(split)을 생성할 수 있다. 상기 스플릿은 태스크에서 처리하는 데이터로서, 상기 쿼리에 따라 가변될 수 있다. 예를 들어, 상기 스플릿 생성부(250)는 상기 쿼리에 관련된 컬럼만을 고려하여 스플릿을 생성할 수 있다. 상기 쿼리에 관련된 컬럼의 크기가 증가할수록 스플릿의 크기는 작아진다. 스플릿 생성에 대한 상세한 설명은 후술하도록 하겠다.
- [0027] 이하에서는 본 발명의 일실시예와 관련된 컬럼 기반 데이터 처리 방법을 상세히 설명하도록 하겠다. 이하, 실시예에서는 하둡 분산 파일 시스템(HDFS, Hadoop Distributed File System)에 기반한 데이터 처리 방법을 예로 설명하도록 하겠다. 하둡 분산 파일 시스템(HDFS, Hadoop Distributed File System) 하에서는 태스크의 기본 단위가 HDFS 블록이 된다.
- [0028] 도 3은 본 발명의 일실시예와 관련된 컬럼 기반 데이터 처리 방법을 나타내는 흐름도이다.
- [0029] 먼저, 데이터 수신부(210)는 m 개의 컬럼을 포함한 입력 데이터를 수신할 수 있다(S310). 상기 입력 데이터는 테이블 형태로 구성될 수 있다. m 은 2 이상의 자연수이다.
- [0030] 샤드 생성부(220)는 상기 입력 데이터의 크기에 근거하여 샤드를 생성할 수 있다(S320). 즉, 샤드 생성부(220)는 상기 입력 데이터의 크기에 근거하여 상기 입력 데이터를 복수 개의 샤드로 수평적으로 분할할 수 있다.
- [0031] 도 4는 본 발명의 일실시예와 관련된 컬럼 기반 데이터 처리 방법에서 입력 데이터를 샤드(shard)로 분할하는 예를 나타내는 도면이다.
- [0032] 도시된 바와 같이, 입력 데이터(400)는 복수 개의 샤드로서 수평적으로 분할된다. 본 실시예에서는 입력 데이터(400)가 5개의 샤드($S1, \dots, S5$)로 수평적으로 분할된다. 입력 데이터가 5개의 샤드로 분할된다는 말은 5개의 샤드가 생성된다는 의미와 동일하다. 생성되는 샤드의 개수는 입력 데이터의 크기에 근거하여 결정될 수 있다. 그리고 하나의 샤드는 m 개의 컬럼(본 실시예에서는 3개의 컬럼)으로 수직적으로 분할될 수도 있다. 그리고 샤드에 포함된 컬럼은 각각 하나의 컬럼 파일로 생성될 수 있다. 따라서 각 샤드는 3개의 컬럼 파일을 포함할 수 있다. 또한, 레코드(420)는 동일 노드에 저장된다. 상기 레코드는 관련된 데이터 필드의 집합을 의미한다. 즉, 튜플(tuple) 값들(예: 2, Cell phone, Low)이 동일 노드에 저장된다.
- [0033] 샤드 할당부(230)는 생성된 5개의 샤드를 각각 복제하여 3개에 노드에 분산시켜 할당할 수 있다(S330). 샤드를 복제하여 노드에 분산시키는 이유는 서버 또는 하드디스크 고장으로 인해 데이터가 손실되는 것을 막기 위함이다. 따라서 동일한 샤드는 동일 노드에 같이 할당되지 않는다. 각 노드는 할당받은 샤드를 저장한다.
- [0034] 도 5는 본 발명의 일실시예와 관련된 컬럼 기반 데이터 처리 방법에서 컬럼 파일의 크기를 결정하는 예를 설명하기 위한 도면이다.
- [0035] 샤드에 포함된 컬럼 파일의 크기는 HDFS 블록 크기(510)에 맞추어질 수 있다. 이 경우, HDFS 블록 크기(510)가 태스크의 기본 단위가 된다. 컬럼 파일의 크기는 HDFS 블록 크기(510)를 초과하지 않으면서 증가될 수 있다.
- [0036] 컬럼 파일의 크기가 HDFS 블록 크기(510)를 초과하지 않으면, 하둡 분산 파일 시스템은 하나의 컬럼 파일 내에서 데이터 로컬리티(data locality)를 보장한다. 이 경우, 동일 레코드에 속하는 값들이 동일 노드에 저장되기에 때문에, 어떠한 네트워크 커뮤니케이션 없이도 레코드에 대한 데이터 처리가 가능하다. 따라서 데이터 로컬리티(data locality)가 보장될 수 있다. 그러나 컬럼 파일의 크기가 HDFS 블록 크기(510)를 초과하면, 데이터 로컬리티(data locality)가 보장되지 않는다.
- [0037] 한편, 컬럼 파일들은 일반적으로 다른 데이터 타입으로 인해 다른 크기를 가지기 때문에, 모든 컬럼 파일을 HDFS 블록 크기에 맞추는 것은 불가능하다. 따라서 HDFS 블록 크기에 맞춰지는 기준이 되는 베이스 컬럼을 선택할 필요가 있다.
- [0038] 베이스 컬럼은 입력 데이터에 포함된 m 개의 컬럼 중 데이터 크기가 가장 큰 컬럼을 베이스 컬럼으로 선택할 수 있다. 예를 들어, "Product"에 해당하는 컬럼이 가장 큰 데이터 크기를 가진다면, 상기 "Product"에 해당하는 컬럼이 베이스 컬럼으로 선택될 수 있다. 그리고 선택된 베이스 컬럼을 기준으로 입력 데이터가 복수 개의 샤드로 수평적으로 분할된다.
- [0039] 그리고 베이스 컬럼 파일은 HDFS 블록 크기에 맞춰진다. 즉, 베이스 컬럼 파일의 크기는 HDFS 블록 크기(510)를 초과하지 않으면서 최대값을 갖도록 설정될 수 있다. HDFS 블록 크기가 64Mb 인 경우, 베이스 컬럼 파일의 크기는 64Mb를 초과하지 않을 때까지 계속 증가될 수 있다. 따라서 최종 베이스 컬럼의 크기는 64Mb와 동일하거나

64Mb 보다 약간 작은 값이 될 수 있다.

- [0040] 베이스 컬럼 외에 다른 컬럼들은 베이스 컬럼 파일의 밸류(value) 수와 동일한 밸류 수를 가진다. 본 명세서에서 밸류(value)는 컬럼에서 행(row)에 해당하는 값을 의미한다. 예를 들어, "Order ID" 컬럼에서는 1, 2, ..., 20000 각각이 하나의 밸류가 된다. 본 실시예에서는 베이스 컬럼 뿐만 아니라 다른 컬럼들도 20000 개의 밸류를 가진다. 상기와 같이, 데이터 크기가 가장 큰 컬럼을 베이스 컬럼으로 선택해야만, 다른 컬럼 파일도 HDFS 블록 크기에 맞출 수 있다.
- [0041] 도 6은 본 발명의 일실시예와 관련된 컬럼 파일의 구조를 나타내는 도면이다.
- [0042] 도시된 바와 같이, 컬럼 파일은 복수 개의 벡터(Vector), 인덱스(Index), 헤더(Header)로 구성될 수 있다.
- [0043] 상기 벡터는 실질적인 데이터를 포함하는 섹션(section)이다. 상기 벡터는 청크(chunk) 단위로 나뉘질 수 있다. 상기 청크는 하둡 분산 파일 시스템에서 한번에 읽어 들일 수 있는 데이터의 크기이다. 예를 들어, 하둡 분산 파일 시스템의 청크가 64KB인 경우, 벡터의 크기는 64KB에 맞춰진다. 상기와 같이, 벡터를 청크 단위로 구분함으로써, 데이터 처리 시의 효율성을 높일 수 있다. 도시된 실시예에서는 벡터는 20000 개의 밸류가 3개의 벡터로 분할된 것을 나타낸다. 처음 두 개의 벡터(Vector 1, Vector 2)에는 8,192개의 밸류가 각각 저장되고, Vector 3에는 3,616개의 밸류가 저장된다.
- [0044] 인덱스에는 각 벡터에 대한 시작 레코드 식별자(start record id) 및 오프셋(offset)이 포함되어 있다. 상기 시작 레코드는 식별자는 각 벡터의 첫 번째 레코드 식별자로서, Vector 1의 시작 레코드는 "1"이 된다. 또한, 오프셋은 각 벡터가 하드디스크에서 어디에 저장되어 있는지에 대한 저장 위치 정보이다.
- [0045] 상기 헤더는 메타 데이터(metadata)를 포함하고 있다. 상기 메타 데이터는 압축 타입(compression type), 데이터 타입(data type), 인덱스에 대한 오프셋 등을 포함할 수 있다.
- [0046] 상기와 같은 컬럼 파일의 구조로 인해, 레코드 식별자가 주어진 경우, 상기 레코드 식별자에 대한 값으로 바로 접근이 가능하다.
- [0047] 한편, 상기와 같은 구조의 m 개의 컬럼 파일을 포함하는 샤드가 n 개의 노드에 분산되어 저장된 상태에서, 쿼리 수신부(240)는 사용자로부터 쿼리를 수신할 수 있다(S350).
- [0048] 스플릿 생성부(250)는 스플릿을 생성할 수 있다(S350). 상기 스플릿 생성부(250)는 상기 쿼리에 따라 스플릿의 크기를 조정하여 스플릿을 생성할 수 있다. 상기 스플릿은 태스크에서 처리하는 데이터로서, 상기 스플릿의 크기는 태스크에서 처리하는 데이터 양이 된다. 스플릿 생성부(250)는 상기 쿼리에 관련된 컬럼만을 고려하여 스플릿의 크기를 결정할 수 있다.
- [0049] 도 7 내지 도 8은 본 발명의 일실시예와 관련된 컬럼 기반 데이터 처리 방법에서 쿼리에 따라 스플릿의 크기가 조정되는 예를 나타내는 도면이다.
- [0050] 도 7은 쿼리가 "Order ID"라는 1개의 컬럼에만 관련된 경우이고, 도 8은 쿼리가 "Order ID, Product, Priority"라는 3개의 컬럼에 모두 관련된 경우를 나타낸다.
- [0051] 쿼리가 "Order ID"라는 1개의 컬럼에만 관련된 경우는 샤드의 수와 동일하게 5개의 태스크가 생성되지만, 쿼리가 "Order ID, Product, Priority"라는 3개의 컬럼에 모두 관련된 경우는 스플릿 크기가 줄어들어 태스크가 15개로 늘어난다. 즉, 태스크의 수는 쿼리에 관련된 컬럼의 크기가 작을수록 줄어들고, 스플릿의 크기는 증가된다.
- [0052] 상기 스플릿의 크기는 하기 수학적 식 1에 의해 결정될 수 있다.

수학적 식 1

$$T = \theta \times \frac{\sum |c_r|}{\sum |c_q|} (c_r \in col(R), c_q \in col(Q))$$

[0053]

- [0054] 여기서 T는 스플릿의 크기이고, Θ 는 하둡 분산 파일 시스템 블록의 크기이고, R은 입력 데이터이고, Q는 쿼리이고, C_r 은 입력 데이터에 포함된 컬럼의 크기이고, C_q 는 쿼리에 포함된 컬럼의 크기이다. 상기 수학적 식 1에 의하면, 쿼리에 관련된 컬럼의 크기가 작을수록 스플릿의 크기는 증가되고, 쿼리에 관련된 컬럼의 크기가 클수록 스플릿의 크기는 감소된다. 따라서 최소의 스플릿 크기는 쿼리에 포함된 컬럼의 크기가 입력 데이터에 포함된 컬럼의 크기와 동일 경우이고, 그 크기는 Θ 이다.
- [0055] 한편, 스플릿의 크기가 클수록 태스크의 수는 줄어들고, 스플릿의 크기가 작을수록 태스크의 수는 증가된다.
- [0056] 상기와 같은 방식으로 스플릿이 생성되면, 각 노드(300)는 해당 노드에 저장된 샤드에 대해 분산 작업을 수행할 수 있다(S360).
- [0057] 도 9은 본 발명의 일실시예와 관련된 컬럼 기반 데이터 처리 방법과 다른 데이터 처리 방법의 성능을 비교한 그래프이다.
- [0058] RowFile은 행기반 바이너리 파일 포맷(row-oriented binary file format)을 이용한 데이터 처리 방법을 나타내고, RCFile Hadoop을 위해 최초로 제안된 컬럼 기반 데이터 처리 방법으로, 행 그룹(row group)을 이용한 데이터 처리 방법을 나타낸다. 또한, Trevni는 행 그룹이 HDFS 블록에 맞춰지도록 설계된 파일 포맷을 이용한 데이터 처리 방법을 나타내고, CIF는 입력 데이터(relation)의 각 컬럼이 개별 파일로서 저장되는 파일 포맷을 이용한 데이터 처리 방법을 나타낸다. 그리고 SSFile는 본 발명의 일실시예에 의한 컬럼 기반 데이터 처리 방법을 나타낸다. 도 9를 통해 본 발명의 일실시예에 의한 컬럼 기반 데이터 처리 방법이 다른 방법보다 데이터 처리 시간이 더 적게 소요되므로 데이터 성능이 우수하다는 것을 확인할 수 있다.
- [0059] 전술한 본 발명의 일실시예에 의한 컬럼 기반 데이터 처리 방법 및 장치는 late materialization, column-specific compression, and direct, operation on compressed data 등의 컬럼 기반 데이터 처리의 최적화 기법을 지원할 수 있다.
- [0060] 전술한 컬럼 기반 데이터 처리 방법 및 장치는 스플릿의 크기를 쿼리에 근거하여 가변적으로 조정하여 태스크 수를 줄일 수 있다. 따라서 오버헤드가 감소될 수 있다.
- [0061] 상술한 컬럼 기반 데이터 처리 방법은 다양한 컴퓨터 수단을 통하여 수행될 수 있는 프로그램 명령 형태로 구현되어 컴퓨터로 판독 가능한 기록 매체에 기록될 수 있다. 이때, 컴퓨터로 판독 가능한 기록매체는 프로그램 명령, 데이터 파일, 데이터 구조 등을 단독으로 또는 조합하여 포함할 수 있다. 한편, 기록매체에 기록되는 프로그램 명령은 본 발명을 위하여 특별히 설계되고 구성된 것들이거나 컴퓨터 소프트웨어 당업자에게 공지되어 사용 가능한 것일 수도 있다.
- [0062] 컴퓨터로 판독 가능한 기록매체에는 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-광매체(Magneto-Optical Media), 및 롬(ROM), 램(RAM), 플래시 메모리 등과 같은 프로그램 명령을 저장하고 수행하도록 특별히 구성된 하드웨어 장치가 포함된다.
- [0063] 한편, 이러한 기록매체는 프로그램 명령, 데이터 구조 등을 지정하는 신호를 전송하는 반송파를 포함하는 광 또는 금속선, 도파관 등의 전송 매체일 수도 있다.
- [0064] 또한, 프로그램 명령에는 컴파일러에 의해 만들어지는 것과 같은 기계어 코드뿐만 아니라 인터프리터 등을 사용해서 컴퓨터에 의해서 실행될 수 있는 고급 언어 코드를 포함한다. 상술한 하드웨어 장치는 본 발명의 동작을 수행하기 위해 하나 이상의 소프트웨어 모듈로서 작동하도록 구성될 수 있으며, 그 역도 마찬가지이다.
- [0065] 상기와 같이 설명된 컬럼 기반 데이터 처리 방법 및 장치는 상기 설명된 실시예들의 구성과 방법이 한정되게 적용될 수 있는 것이 아니라, 상기 실시예들은 다양한 변형이 이루어질 수 있도록 각 실시예들의 전부 또는 일부가 선택적으로 조합되어 구성될 수도 있다.

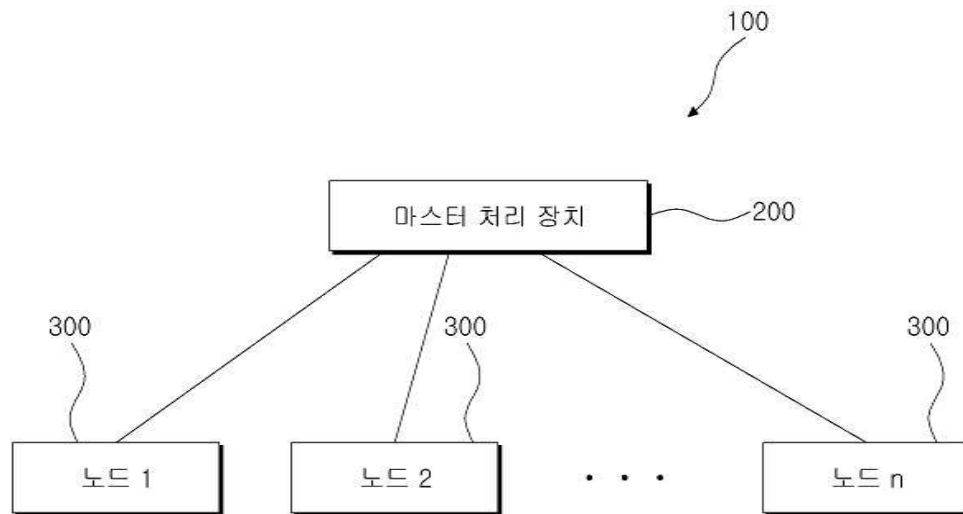
부호의 설명

- [0066] 100: 컬럼 기반 데이터 처리 장치
200: 마스터 처리 장치
210: 데이터 수신부

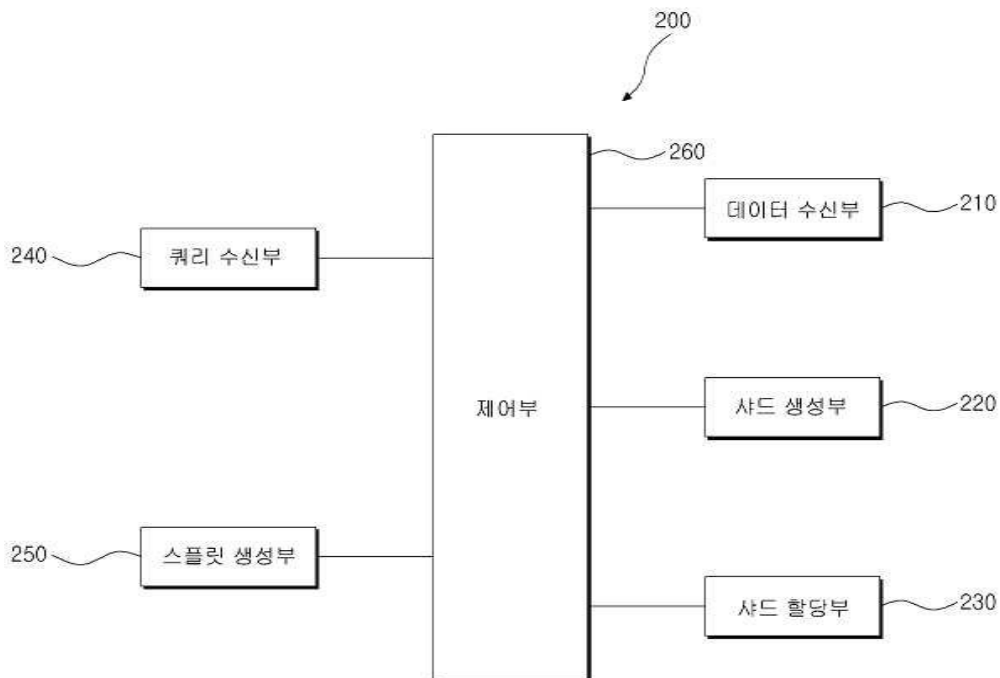
220: 샤드 생성부
230: 샤드 할당부
240: 쿼리 수신부
250: 스플릿 생성부
260: 제어부
300: 노드

도면

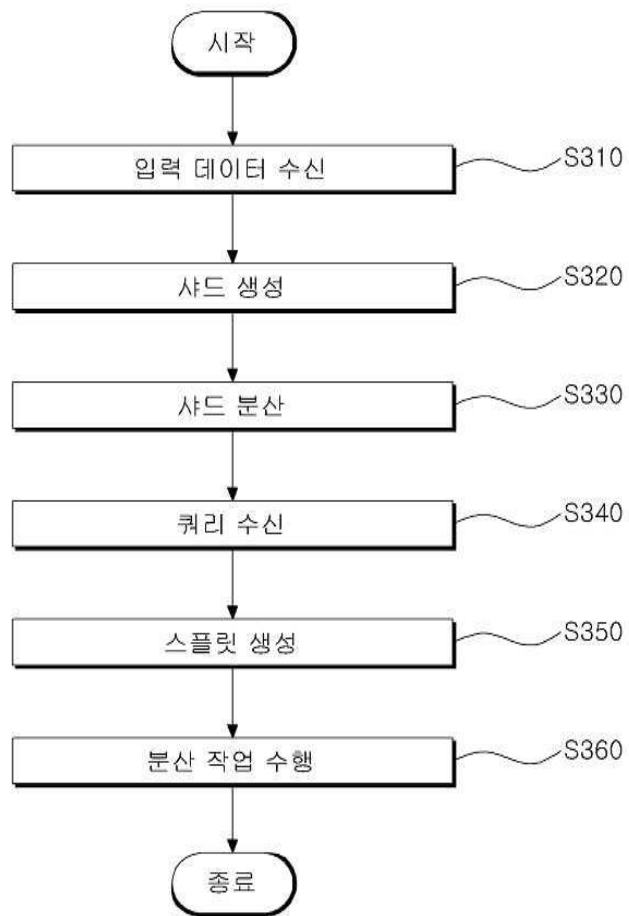
도면1



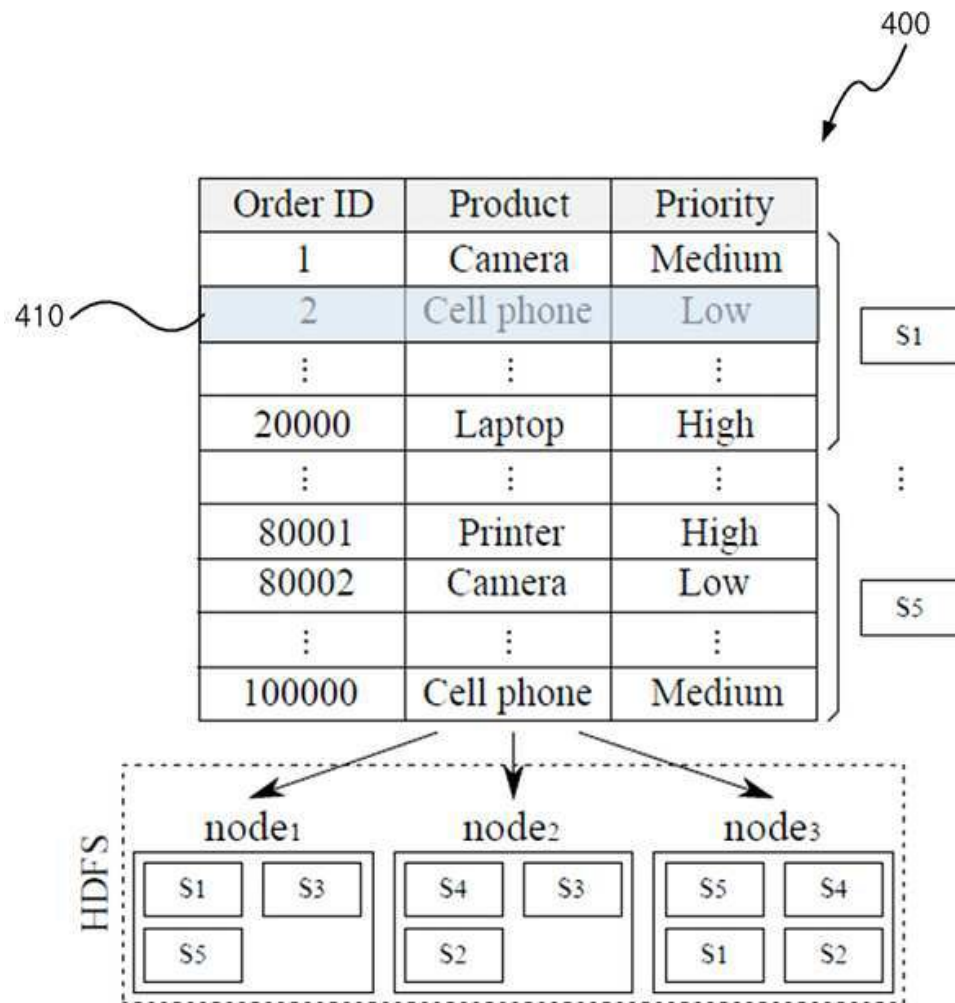
도면2



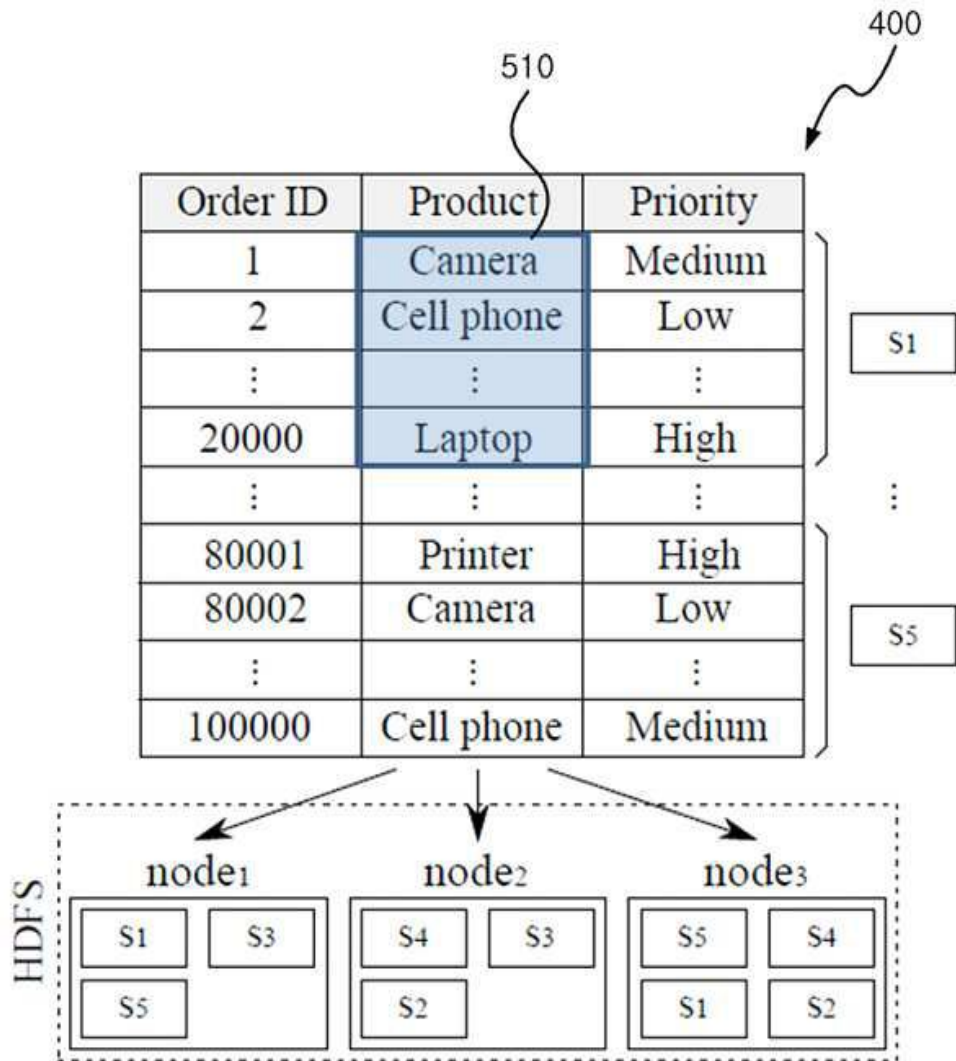
도면3



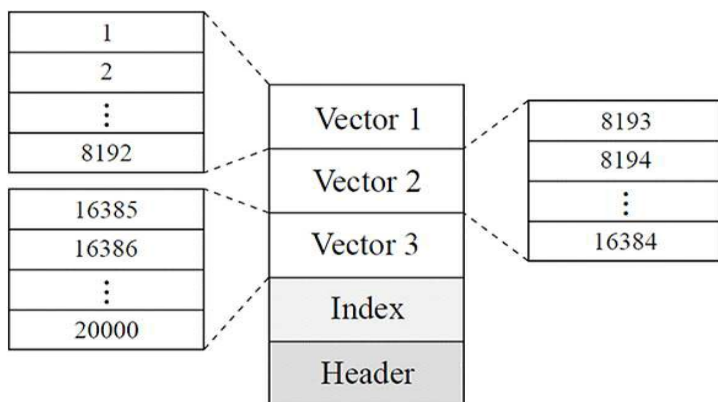
도면4



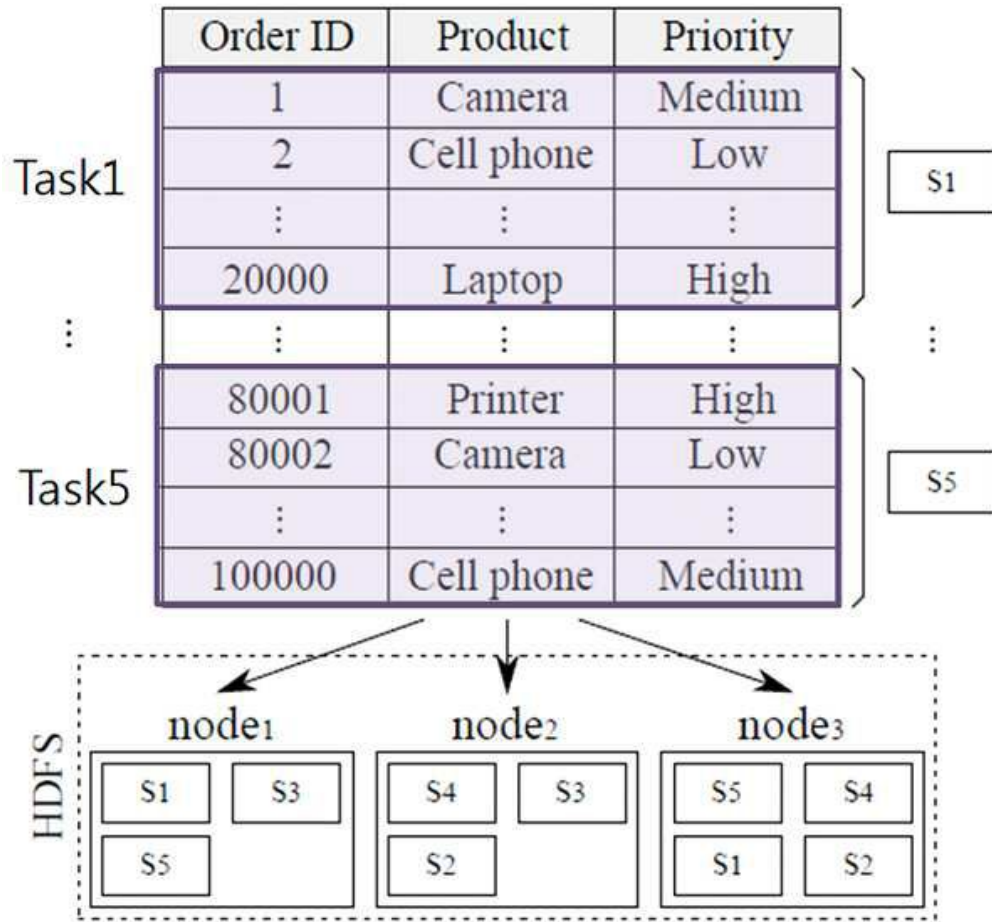
도면5



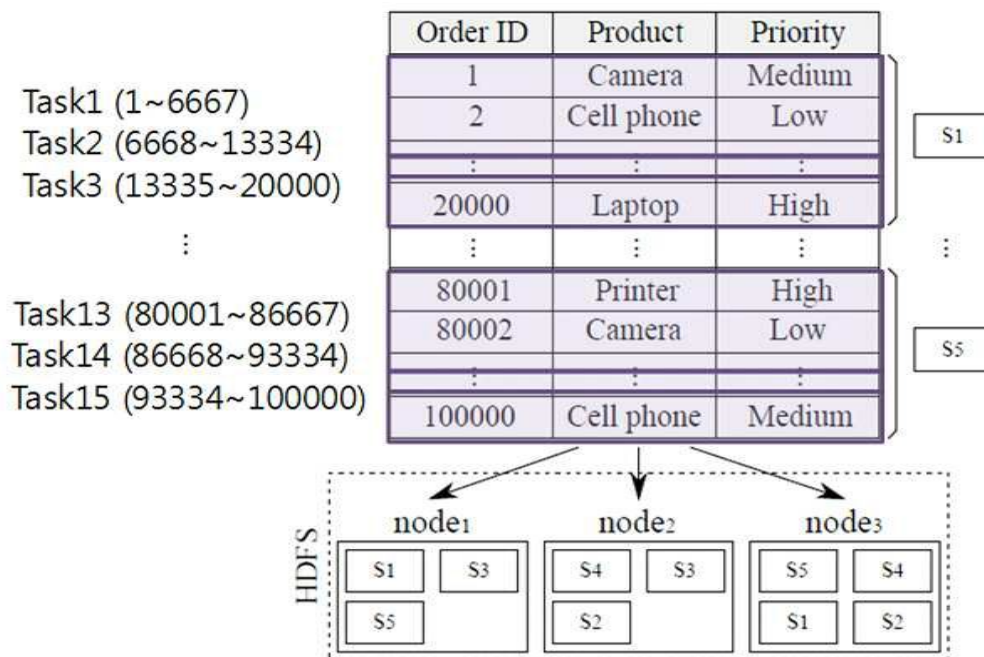
도면6



도면7



도면8



도면9

