(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2023-0013429
(43) 공개일자 2023년01월26일

(51) 국제특허분류(Int. Cl.)
G06F 16/2453 (2019.01) *G06F 16/22* (2019.01)
G06F 16/2455 (2019.01)
(52) CPC특허분류
G06F 16/24547 (2019.01)
G06F 16/2264 (2019.01)
(21) 출원번호 10-2021-0094154
(22) 출원일자 2021년07월19일
심사청구일자 2021년07월19일

(71) 출원인
고려대학교 산학협력단
서울특별시 성북구 안암로 145, 고려대학교 (안암동5가)
(72) 발명자
정연돈
서울특별시 중구 청계천로 400, 105-812
최달수
경기도 수원시 영통구 봉영로1517번길 73, 933동 1102호
(74) 대리인
김홍석

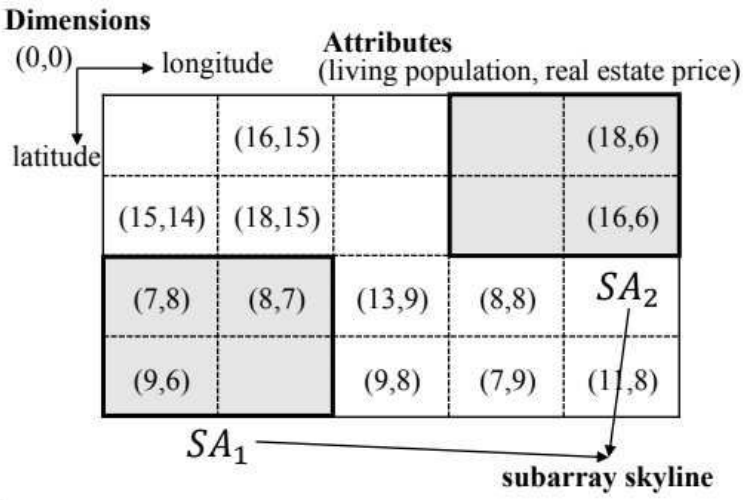
전체 청구항 수 : 총 10 항

(54) 발명의 명칭 **배열 데이터베이스에서의 하위 배열 스카이라인 질의 처리 방법 및 장치**

(57) 요 약

본 발명에 따른 배열 데이터베이스에서의 하위 배열 스카이라인 질의 처리 방법은 질의 처리와 연관된 제1 속성 값(attribute value) 및 제2 속성 값에 기초하여 구성된 복수의 하위 배열 (sub array)에 대하여 하위 배열 그룹들을 구성하는 하위 배열 그룹 구성 과정; 상기 하위 배열 그룹들의 세트들에서 특정 하위 배열 그룹들의 MBR (Minimum Bounding Rectangle)이 다른 하위 배열 그룹들의 MBR 에 의해 지배(dominate)되지 않는 상기 특정 하위 배열 그룹들을 탐색하는 하위 배열 그룹 탐색 과정; 및 상기 특정 하위 배열 그룹들의 각각에 대해 특정 하위 배열이 다른 하위 배열에 의해 지배되지 않는 상기 특정 하위 배열을 연산하는 특정 하위 배열 연산 과정을 포함할 수 있다.

대 표 도 - 도1a



(52) CPC특허분류
G06F 16/24542 (2019.01)
G06F 16/24554 (2019.01)
G06F 16/24564 (2019.01)
G06F 16/90335 (2019.01)

이 발명을 지원한 국가연구개발사업
과제고유번호 1711127329
과제번호 2020R1A2C2013286
부처명 과학기술정보통신부
과제관리(전문)기관명 한국연구재단
연구사업명 개인기초연구(과기정통부)(R&D)
연구과제명 대규모 분산 딥러닝을 위한 인메모리 텐서 데이터베이스
기 여 율 1/2
과제수행기관명 고려대학교
연구기간 2021.03.01 ~ 2022.02.28
이 발명을 지원한 국가연구개발사업
과제고유번호 1711126325
과제번호 2020-0-01819-002
부처명 과학기술정보통신부
과제관리(전문)기관명 정보통신기획평가원
연구사업명 정보통신방송혁신인재양성(R&D)
연구과제명 ICT명품인재양성(고려대학교)
기 여 율 1/2
과제수행기관명 고려대학교산학협력단
연구기간 2021.01.01 ~ 2021.12.31

명 세 서

청구범위

청구항 1

배열 데이터베이스에서의 하위 배열 스카이라인 질의 처리 방법에 있어서, 상기 방법은 질의 처리 장치에 의해 수행되고,

질의 처리와 연관된 제1 속성 값(attribute value) 및 제2 속성 값에 기초하여 구성된 복수의 하위 배열 (sub array)에 대하여 하위 배열 그룹들을 구성하는 하위 배열 그룹 구성 과정;

상기 하위 배열 그룹들의 세트들에서 특정 하위 배열 그룹들의 MBR (Minimum Bounding Rectangle)이 다른 하위 배열 그룹들의 MBR 에 의해 지배(dominate)되지 않는 상기 특정 하위 배열 그룹들을 탐색하는 하위 배열 그룹 탐색 과정;

상기 특정 하위 배열 그룹들의 각각에 대해 특정 하위 배열이 다른 하위 배열에 의해 지배되지 않는 상기 특정 하위 배열을 연산하는 특정 하위 배열 연산 과정;

상기 특정 하위 배열 그룹의 탐색 결과 및 상기 특정 하위 배열의 연산 결과에 기초하여, 복수의 하위 프로세서들 각각이 상기 특정 하위 배열 그룹들의 각각의 작업(task)을 분산 처리하도록 상기 복수의 하위 프로세서들을 제어하는 분산 처리 제어 과정을 포함하는,

질의 처리 방법.

청구항 2

제1 항에 있어서,

상기 하위 배열 그룹 탐색 과정에서,

제1 하위 배열 그룹에 대응하는 제1 MBR의 제1 상한 및 제1 하한과 상기 제2 하위 배열 그룹에 대응하는 제2 MBR의 제2 상한 및 제2 하한을 비교하여, 상기 제1 하위 배열 그룹과 상기 제2 하위 배열 그룹 간의 지배 관계를 판단하는,

질의 처리 방법.

청구항 3

제2 항에 있어서,

상기 하위 배열 연산 과정에서,

상기 특정 하위 배열 그룹들의 각각에 대해 제1 하위 배열에 대응하는 제1 MBR의 제1 상한 및 제1 하한과 상기 제2 하위 배열에 대응하는 제2 MBR의 제2 상한 및 제2 하한을 비교하여, 상기 제1 하위 배열과 상기 제2 하위 배열 간의 지배 관계를 판단하는,

질의 처리 방법.

청구항 4

제2 항에 있어서,

상기 제1 상한을 상기 제2 하한과 비교하여, 상기 제1 하위 배열 그룹이 상기 제2 하위 배열 그룹을 상기 제1 속성 값 및 상기 제2 속성 값에 대해 지배하는지 판단하고,

상기 제1 하한을 상기 제2 상한과 비교하여, 상기 제1 하위 배열 그룹이 상기 제2 하위 배열 그룹을 상기 제1 속성 값 및 상기 제2 속성 값 중 어느 하나에 대해 부분 지배하는지 판단하는,

질의 처리 방법.

청구항 5

제1 항에 있어서,

상기 하위 배열 그룹 구성 과정에서, 상기 복수의 하위 프로세서들 각각이 하위 배열의 집계 속성 값(aggregated attribute values)을 연산한 값에 기초하여, 코디네이터 프로세서가 상기 하위 배열 그룹을 형성하고,

상기 하위 배열 그룹 탐색 과정은 코디네이터 프로세서가 상기 복수의 하위 프로세서들로부터 수집되어 샘플링된 제1 로컬 정보에 기초하여, 글로벌 샘플링 기반의 상기 특정 하위 배열 그룹의 탐색을 수행하는,

질의 처리 방법.

청구항 6

제5 항에 있어서,

상기 특정 하위 배열 연산 과정은 상기 코디네이터 프로세서가 상기 복수의 하위 프로세서들로부터 수집되어 샘플링된 제2 로컬 정보에 기초하여, 상기 특정 하위 배열 그룹들의 각각에 대해 글로벌 샘플링 기반의 상기 특정 하위 배열을 연산하는,

질의 처리 방법.

청구항 7

제6 항에 있어서,

상기 하위 배열 그룹 구성 과정에서,

상기 코디네이터 프로세서가 네트워크 사용량과 상기 복수의 하위 프로세서들의 로드 상태에 기초한 로드 밸런싱을 수행하여, 상기 하위 배열 그룹들을 구성하는,

질의 처리 방법.

청구항 8

제7 항에 있어서,

상기 분산 처리 제어 과정에서,

상기 로드 밸런싱에 따라 구성된 상기 하위 배열 그룹들에서 다른 하위 배열 그룹에 의해 지배되지 않는 상기 특정 하위 배열 그룹들의 각각을 상기 복수의 하위 프로세서들 각각이 분산 처리하고,

상기 샘플링된 제1 로컬 정보 및 상기 샘플링된 제2 로컬 정보에 기초하여, 글로벌 샘플링 기반으로 연산된 상기 특정 하위 배열을 상기 복수의 하위 프로세서들 각각이 다른 하위 배열보다 우선적으로 처리하도록 제어하는,

질의 처리 방법.

청구항 9

배열 데이터베이스에서 하위 배열 스카이라인 질의 처리를 수행하는 질의 처리 장치에 있어서,

코디네이터 프로세서; 및

상기 코디네이터 프로세서와 동작 가능하게 결합되고, 분산 처리를 수행하도록 구성된 복수의 하위 프로세서들을 포함하고,

상기 코디네이터 프로세서는,

질의 처리와 연관된 제1 속성 값(attribute value) 및 제2 속성 값에 기초하여 구성된 복수의 하위 배열 (sub array)에 대하여 하위 배열 그룹들을 구성하고,

상기 하위 배열 그룹들의 세트들에서 특정 하위 배열 그룹들의 MBR (Minimum Bounding Rectangle)이 다른 하위

배열 그룹들의 MBR 에 의해 지배(dominate)되지 않는 상기 특정 하위 배열 그룹들을 탐색하고,

상기 특정 하위 배열 그룹들의 각각에 대해 특정 하위 배열이 다른 하위 배열에 의해 지배되지 않는 상기 특정 하위 배열을 연산하고,

상기 특정 하위 배열 그룹의 탐색 결과 및 상기 특정 하위 배열의 연산 결과에 기초하여, 상기 복수의 하위 프로세서들 각각이 상기 특정 하위 배열 그룹들의 각각의 작업(task)을 분산 처리하도록 상기 복수의 하위 프로세서들을 제어하는,

질의 처리 장치.

청구항 10

제9 항에 있어서,

상기 코디네이터 프로세서는,

제1 하위 배열 그룹에 대응하는 제1 MBR의 제1 상한 및 제1 하한과 상기 제2 하위 배열 그룹에 대응하는 제2 MBR의 제2 상한 및 제2 하한을 비교하여, 상기 제1 하위 배열 그룹과 상기 제2 하위 배열 그룹 간의 지배 관계를 판단하고,

상기 제1 상한을 상기 제2 하한과 비교하여, 상기 제1 하위 배열 그룹이 상기 제2 하위 배열 그룹을 상기 제1 속성 값 및 상기 제2 속성 값에 대해 지배하는지 판단하고,

상기 제1 하한을 상기 제2 상한과 비교하여, 상기 제1 하위 배열 그룹이 상기 제2 하위 배열 그룹을 상기 제1 속성 값 및 상기 제2 속성 값 중 어느 하나에 대해 부분 지배하는지 판단하는,

질의 처리 장치.

발명의 설명

기 술 분 야

[0001] 본 발명은 데이터베이스 질의 처리에 관한 것이며, 보다 상세하게는 배열 데이터베이스에서 효율적으로 하위 배열 스카이라인을 찾아내는 질의 처리 방법 및 장치에 관한 것이다.

배 경 기 술

[0002] 질의 처리(query processing)는 공간 데이터와 같은 다차원 데이터에 대하여 이루어질 수 있다. 스카이라인(skyline)은 전체 객체 집합에서 다른 객체보다 나쁘지 않는 관심 있을 만한 객체 집합이다.

[0003] 스카이라인 질의는 다양하고 복잡하며 서로 상충하는 사용자의 요구에 가장 부합하는 데이터를 효율적으로 검색하기 위한 질의로서 사용자의 의사 결정 과정에 영향을 미칠 수 있다. 이와 같은 다차원 데이터에 대한 질의는 각 데이터의 차원 별 비교를 통하여 해 집합으로의 포함 여부가 결정되기 때문에 많은 계산 량이 요구된다.

[0004] 특히, 다양하고 복잡하며 서로 상충하는 사용자의 요구에 가장 부합하는 데이터를 효율적으로 검색하기 위한 질의로서 스카이라인 질의를 처리하는 경우, 많은 계산 량이 요구된다는 문제점이 있다.

발명의 내용

해결하려는 과제

[0005] 본 발명은 상기와 같은 문제를 해결하기 위해 제안된 것으로, 배열 데이터베이스에서의 하위 배열 스카이라인 질의 처리 방법 및 장치를 제안하고자 한다.

[0006] 또한, 본 발명의 목적은 위치를 나타내는 차원(dimension)과 별개로 속성 값(attribute)을 갖는 대용량 배열 데이터를 빠르게 분석하기 위한 질의 처리 방법 및 장치를 제안하고자 한다.

[0007] 또한, 본 발명의 목적은 차원(dimension)과 별개로 속성 값(attribute)을 갖는 대용량 배열 데이터 특성을 동시에 고려할 수 있는 스카이라인 질의를 하위 배열 단위로 제공하고자 한다.

[0008] 또한, 본 발명은 한국연구재단의 '프라이버시 침해 없이 감염병을 예방하는 데이터 기반 시스템' 과제(과제번호

: 2020K2A9A1A0109589411), 한국연구재단의 '대규모 분산 딥러닝을 위한 인메모리 텐서 데이터베이스' 과제(과제번호 : 2020R1A2C201328612), 및 정보통신기획평가원원 'ICT명품인재양성' 과제(과제번호 : 20200018190021001)의 결과물이다.

과제의 해결 수단

[0009] 본 발명에 따른 배열 데이터베이스에서의 하위 배열 스카이라인 질의 처리 방법은 질의 처리 장치에 의해 수행되고, 질의 처리와 연관된 제1 속성 값(attribute value) 및 제2 속성 값에 기초하여 구성된 복수의 하위 배열(sub array)에 대하여 하위 배열 그룹들을 구성하는 하위 배열 그룹 구성 과정; 상기 하위 배열 그룹들의 세트들에서 특정 하위 배열 그룹들의 MBR (Minimum Bounding Rectangle)이 다른 하위 배열 그룹들의 MBR에 의해 지배(dominate)되지 않는 상기 특정 하위 배열 그룹들을 탐색하는 하위 배열 그룹 탐색 과정; 및 상기 특정 하위 배열 그룹들의 각각에 대해 특정 하위 배열이 다른 하위 배열에 의해 지배되지 않는 상기 특정 하위 배열을 연산하는 특정 하위 배열 연산 과정; 및 상기 특정 하위 배열 그룹의 탐색 결과 및 상기 특정 하위 배열의 연산 결과에 기초하여, 복수의 하위 프로세서들 각각이 상기 특정 하위 배열 그룹들의 각각의 작업(task)을 분산 처리하도록 상기 복수의 하위 프로세서들을 제어하는 분산 처리 제어 과정을 포함할 수 있다.

[0010] 실시 예에 따르면, 상기 하위 배열 그룹 탐색 과정에서 제1 하위 배열 그룹에 대응하는 제1 MBR의 제1 상한 및 제1 하한과 상기 제2 하위 배열 그룹에 대응하는 제2 MBR의 제2 상한 및 제2 하한을 비교하여, 상기 제1 하위 배열 그룹과 상기 제2 하위 배열 그룹 간의 지배 관계를 판단할 수 있다.

[0011] 실시 예에 따르면, 상기 하위 배열 연산 과정에서 상기 특정 하위 배열 그룹들의 각각에 대해 제1 하위 배열에 대응하는 제1 MBR의 제1 상한 및 제1 하한과 상기 제2 하위 배열에 대응하는 제2 MBR의 제2 상한 및 제2 하한을 비교하여, 상기 제1 하위 배열과 상기 제2 하위 배열 간의 지배 관계를 판단할 수 있다.

[0012] 실시 예에 따르면, 상기 제1 상한을 상기 제2 하한과 비교하여, 상기 제1 하위 배열 그룹이 상기 제2 하위 배열 그룹을 상기 제1 속성 값 및 상기 제2 속성 값에 대해 지배하는지 판단할 수 있다. 한편, 상기 제1 하한을 상기 제2 상한과 비교하여, 상기 제1 하위 배열 그룹이 상기 제2 하위 배열 그룹을 상기 제1 속성 값 및 상기 제2 속성 값 중 어느 하나에 대해 부분 지배하는지 판단할 수 있다.

[0013] 실시 예에 따르면, 상기 하위 배열 그룹 구성 과정에서, 상기 복수의 하위 프로세서들 각각이 하위 배열의 집계 속성 값(aggregated attribute values)을 연산한 값에 기초하여, 코디네이터 프로세서가 상기 하위 배열 그룹을 형성할 수 있다. 상기 하위 배열 그룹 탐색 과정은 코디네이터 프로세서가 상기 복수의 하위 프로세서들로부터 수집되어 샘플링된 제1 로컬 정보에 기초하여, 글로벌 샘플링 기반의 상기 특정 하위 배열 그룹의 탐색을 수행할 수 있다.

[0014] 실시 예에 따르면, 상기 특정 하위 배열 연산 과정은 상기 코디네이터 프로세서가 상기 복수의 하위 프로세서들로부터 수집되어 샘플링된 제2 로컬 정보에 기초하여, 상기 특정 하위 배열 그룹들의 각각에 대해 글로벌 샘플링 기반의 상기 특정 하위 배열을 연산할 수 있다.

[0015] 실시 예에 따르면, 상기 하위 배열 그룹 구성 과정에서 상기 코디네이터 프로세서가 네트워크 사용량과 상기 복수의 하위 프로세서들의 로드 상태에 기초한 로드 밸런싱을 수행하여, 상기 하위 배열 그룹들을 구성할 수 있다.

[0016] 실시 예에 따르면, 상기 분산 처리 제어 과정에서, 상기 로드 밸런싱에 따라 구성된 상기 하위 배열 그룹들에서 다른 하위 배열 그룹에 의해 지배되지 않는 상기 특정 하위 배열 그룹들의 각각을 상기 복수의 하위 프로세서들 각각이 분산 처리하고, 상기 샘플링된 제1 로컬 정보 및 상기 샘플링된 제2 로컬 정보에 기초하여, 글로벌 샘플링 기반으로 연산된 상기 특정 하위 배열을 상기 복수의 하위 프로세서들 각각이 다른 하위 배열보다 우선적으로 처리하도록 제어할 수 있다.

[0017] 본 발명의 다른 양상에 따른 배열 데이터베이스에서 하위 배열 스카이라인 질의 처리를 수행하는 질의 처리 장치가 제공된다. 상기 질의 처리 장치는 코디네이터 프로세서; 및 상기 코디네이터 프로세서와 동작 가능하게 결합되고, 분산 처리를 수행하도록 구성된 복수의 하위 프로세서들을 포함한다. 상기 코디네이터 프로세서는 질의 처리와 연관된 제1 속성 값(attribute value) 및 제2 속성 값에 기초하여 구성된 복수의 하위 배열 (sub array)에 대하여 하위 배열 그룹들을 구성하고, 상기 하위 배열 그룹들의 세트들에서 특정 하위 배열 그룹들의 MBR (Minimum Bounding Rectangle)이 다른 하위 배열 그룹들의 MBR에 의해 지배(dominate)되지 않는 상기 특정 하위 배열 그룹들을 탐색하고, 상기 특정 하위 배열 그룹들의 각각에 대해 특정 하위 배열이 다른 하위 배열에 의

해 지배되지 않는 상기 특정 하위 배열을 연산할 수 있다.

[0018] 상기 코디네이터 프로세서는 상기 특정 하위 배열 그룹의 탐색 결과 및 상기 특정 하위 배열의 연산 결과에 기초하여, 상기 복수의 하위 프로세서들 각각이 상기 특정 하위 배열 그룹들의 각각의 작업(task)을 분산 처리하도록 상기 복수의 하위 프로세서들을 제어할 수 있다.

[0019] 상기 코디네이터 프로세서는 제1 하위 배열 그룹에 대응하는 제1 MBR의 제1 상한 및 제1 하한과 상기 제2 하위 배열 그룹에 대응하는 제2 MBR의 제2 상한 및 제2 하한을 비교하여, 상기 제1 하위 배열 그룹과 상기 제2 하위 배열 그룹 간의 지배 관계를 판단한다. 이와 관련하여, 상기 코디네이터 프로세서는 상기 제1 상한을 상기 제2 하한과 비교하여, 상기 제1 하위 배열 그룹이 상기 제2 하위 배열 그룹을 상기 제1 속성 값 및 상기 제2 속성 값에 대해 지배하는지 판단할 수 있다. 상기 코디네이터 프로세서는 상기 제1 하한을 상기 제2 상한과 비교하여, 상기 제1 하위 배열 그룹이 상기 제2 하위 배열 그룹을 상기 제1 속성 값 및 상기 제2 속성 값 중 어느 하나에 대해 부분 지배하는지 판단할 수 있다.

발명의 효과

[0020] 본 발명의 실시예에 따르면, 기존에 존재하는 질의 처리 방법 대비 질의 처리 소요시간을 대폭 감소시킬 수 있다.

[0021] 또한, 본 발명의 실시예에 따르면, 기존 방법들이 제공하지 못하던 분산 처리를 지원함으로써 초대용량 배열 데이터에 대한 질의 처리도 가능하다.

[0022] 또한, 본 발명의 실시예에 따르면, 다양한 분야에서 생성되는 배열 데이터에 대한 빠른 분석을 통해 의사 결정 과정에 크게 기여할 수 있다.

[0023] 상술한 본 발명의 특징 및 효과는 첨부된 도면과 관련한 다음의 상세한 설명을 통하여 보다 분명해 질 것이며, 그에 따라 본 발명이 속하는 기술 분야에서 통상의 지식을 가진 자가 본 발명의 기술적 사상을 용이하게 실시할 수 있을 것이다.

도면의 간단한 설명

[0024] 도 1a 및 도 1b는 본 발명에 따른 배열 데이터에 대한 하위 배열 구성 방법을 이용한 질의 처리의 개념도를 나타낸다.

도 2는 본 발명에 따른 MBR들의 지배 관계를 나타낸 것이다.

도 3은 본 발명에 따른 하위 배열 그룹화의 일 예를 나타낸다.

도 4는 본 발명에 따른 하위 배열 그룹들에서 의미 있는 특정 하위 배열 그룹을 탐색하는 개념도를 나타낸다.

도 5는 본 발명의 일 예시에 따른 하위 배열 그룹을 탐색하는 알고리즘을 나타낸다.

도 6은 본 발명에 따른 특정 하위 배열을 탐색하는 개념도를 나타낸다.

도 7은 본 발명의 다양한 예시에 따른 3가지 경우의 하위 배열 그룹 간의 부분 지배 관계를 나타낸다.

도 8a 및 도 8b는 본 발명의 일 예시에 따른 하위 배열을 탐색하는 알고리즘을 나타낸다.

도 9는 본 발명의 일 예시에 따른 배열이 청크 단위로 배분되어 각각의 하위 프로세서 노드로 배분되는 개념도를 나타낸다.

도 10a는 특정 하위 배열 그룹들 및 특정 하위 배열의 탐색과 관련된 절차를 나타낸다.

도 10b는 본 발명에 따른 하위 배열 스카이라인 질의 처리 방법을 수행하는 질의 처리 장치의 구성을 나타낸다.

도 11a는 하위 배열 지배(subarray dominance)와 MBR 지배 및 MBR 부분 지배에 관한 정의를 나타낸다. 한편, 도 11b는 제1 및 제2 MBR 간의 지배 관계의 판단과 관련된 기준을 나타낸다.

도 12는 본 발명에 따른 배열 데이터베이스에서의 하위 배열 스카이라인 질의 처리 방법의 흐름도를 나타낸다.

발명을 실시하기 위한 구체적인 내용

[0025] 상술한 본 발명의 특징 및 효과는 첨부된 도면과 관련한 다음의 상세한 설명을 통하여 보다 분명해 질 것이며,

그에 따라 본 발명이 속하는 기술 분야에서 통상의 지식을 가진 자가 본 발명의 기술적 사상을 용이하게 실시할 수 있을 것이다.

[0026] 본 발명은 다양한 변경을 가할 수 있고 여러 가지 실시 예를 가질 수 있는바, 특정 실시 예들을 도면에 예시하고 상세한 설명에 구체적으로 설명하고자 한다. 그러나 이는 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.

[0027] 각 도면을 설명하면서 유사한 참조부호를 유사한 구성요소에 대해 사용한다.

[0028] 제1, 제2등의 용어는 다양한 구성요소들을 설명하는데 사용될 수 있지만, 상기 구성요소들은 상기 용어들에 의해 한정되어서는 안 된다. 상기 용어들은 하나의 구성요소를 다른 구성요소로부터 구별하는 목적으로만 사용된다.

[0029] 예를 들어, 본 발명의 권리 범위를 벗어나지 않으면서 제1 구성요소는 제2 구성요소로 명명될 수 있고, 유사하게 제2 구성요소도 제1 구성요소로 명명될 수 있다. 및/또는 이라는 용어는 복수의 관련된 기재된 항목들의 조합 또는 복수의 관련된 기재된 항목들 중의 어느 항목을 포함한다.

[0030] 다르게 정의되지 않는 한, 기술적이거나 과학적인 용어를 포함해서 여기서 사용되는 모든 용어들은 본 발명이 속하는 기술분야에서 통상의 지식을 가진 자에 의해 일반적으로 이해되는 것과 동일한 의미가 있다.

[0031] 일반적으로 사용되는 사전에 정의되어 있는 것과 같은 용어들은 관련 기술의 문맥상 가지는 의미와 일치하는 의미를 가지는 것으로 해석되어야 하며, 본 출원에서 명백하게 정의하지 않는 한, 이상적이거나 과도하게 형식적인 의미로 해석되지 않아야 한다.

[0032] 이하의 설명에서 사용되는 구성요소에 대한 접미사 모듈, 블록 및 부는 명세서 작성의 용이함만이 고려되어 부여되거나 혼용되는 것으로서, 그 자체로 서로 구별되는 의미 또는 역할을 갖는 것은 아니다.

[0033] 이하, 본 발명의 바람직한 실시 예를 첨부한 도면을 참조하여 당해 분야에 통상의 지식을 가진 자가 용이하게 실시할 수 있도록 설명한다. 하기에서 본 발명의 실시 예를 설명함에 있어, 관련된 공지의 기능 또는 공지의 구성에 대한 구체적인 설명이 본 발명의 요지를 불필요하게 흐릴 수 있다고 판단되는 경우에는 그 상세한 설명을 생략한다.

[0034] 이하에서는, 본 발명에 따른 배열 데이터베이스에서의 하위 배열 스카이라인 질의 처리 방법 및 장치에 대해 도면을 참조하여 설명한다. 도 1a 및 도 1b는 본 발명에 따른 배열 데이터에 대한 하위 배열 구성 방법을 이용한 질의 처리의 개념도를 나타낸다. 도 1a는 제1 및 제2 속성에 대한 분포도를 비교한 것이다.

[0035] 도 1a (a)는 본 발명에 따른 하위 배열 구성 방식에 따른 제1 및 제2 속성 값의 평균 기반 분포도이다. 반면에, 도 1a (a)는 본 발명과 관련하여 제1 및 제2 속성 값의 분포도이다. 제1 및 제2 속성 값으로 거주 인구(living population) 및 부동산 가격(real estate price)이 선택되었으나, 이에 한정되는 것은 아니고 응용에 따라 변경 가능하다.

[0036] 도 1b는 도 1a (a)의 분포도를 고려하여 2x2 하위 배열 크기로 하위 배열을 구성한 것이다. 이와 관련하여, “스카이라인(SKyline)”의 의미는 속성을 고려하여 더 의미 있는 하위 배열들을 탐색하거나 덜 의미 있는 하위 배열들을 필터링한다는 의미로 해석될 수 있다. 따라서, 본 발명은 하위 배열 단위로 스카이라인을 계산함으로써 새로운 배열 탐색 방법을 제공하고자 한다.

[0037] 이를 위해 먼저 하위 배열 단위의 지배 관계를 정의한다. 다시 말해, 하위 배열에 속하는 속성의 집계 값(aggregated value)을 바탕으로 하위 배열 간의 지배 관계를 정의할 수 있다. 그 다음으로 MBR(Minimum Bounding Rectangle) 단위의 두 가지 지배 관계를 정의할 수 있다. 이러한 지배 관계의 파악에 따라 하위 배열을 그룹화하여 적용할 수 있으며, 결과적으로 전체 CPU 사용량을 낮출 수 있다.

[0038] 전술한 기술적 특징과 관련하여, 도 2는 본 발명에 따른 MBR들의 지배 관계를 나타낸 것이다. 도 2(a)를 참조하면, MBR1의 상한(upper bound, UB)과 MBR2의 하한(lower bound, LB)을 비교하면, MBR1이 MBR2를 지배(dominate)한다고 판단할 수 있다. 한편, 도 2(b)를 참조하면, MBR1의 하한(LB)과 MBR2의 상한(MB)을 비교하면, MBR1이 MBR2를 부분적으로 지배(partially dominate)한다고 판단할 수 있다. 또한, 도 2(c)를 참조하면, MBR1의 상한(UB) 및 하한(LB)과 MBR2의 상한(UB) 및 하한(LB)을 비교하면, MBR1과 MBR2 상호 간에 부분적 지배 관계가 없다고 판단할 수 있다.

[0039] 한편, 본 발명에 따른 배열 데이터베이스에서의 하위 배열 스카이라인 질의 처리 방법과 관련하여 하위 배열 그룹화 및 하위 배열 그룹 스카이라인 탐색과 하위 배열 스카이라인 탐색에 대해 설명한다.

[0040] 먼저, 도 3은 본 발명에 따른 하위 배열 그룹화의 일 예를 나타낸다. 도 3에 도시된 바와 같이, 하위 배열 SA(0, 0) 및 SA(0, 1)이 하위 배열 그룹 SG(0, 0)로 그룹화 될 수 있다. 이와 같은 방식으로 하위 배열 SA(2, 2) 및 SA(2, 3)이 하위 배열 그룹 SG(2, 2)으로 그룹화 될 수 있다.

[0041] 한편, 도 4는 본 발명에 따른 하위 배열 그룹들에서 의미 있는 특정 하위 배열 그룹을 탐색하는 개념도를 나타낸다. 도 4를 참조하면, 하위 배열 그룹들의 MBR들 중에서 SG-Skyline, 즉 의미 있는 특정 하위 배열 그룹을 탐색할 수 있다. 도 4(a)와 같이 각각의 하위 배열 그룹들의 MBR의 상한 및 하한에 기반하여 지배 관계를 판단할 수 있다. 이에 따라, 도 4(b)와 같이 다른 하위 배열 그룹들에 대해 지배되지 않는 특정 하위 배열 그룹들이 선택될 수 있다.

[0042] 이와 관련하여, 도 5는 본 발명의 일 예시에 따른 하위 배열 그룹을 탐색하는 알고리즘을 나타낸다. 구체적으로, 도 5는 하위 배열 그룹을 탐색하는 알고리즘 1 (SG_SKY)을 나타낸다. 또한, 도 5는 하위 배열 그룹이 다른 하위 배열 그룹에 의해 지배되는지 여부에 따라 서로 다른 논리 값, 즉 참/거진(true/false)을 반환하는 알고리즘 2 (SET_{SG})를 나타낸다.

[0043] 도 5를 참조하면, 알고리즘 1과 2는 *SG-Skyline*의 멤버가 아닌 하위 배열 그룹 *SG*가 다른 하위 배열 그룹에 의해 지배된다는 것을 나타낸다. 왜냐하면, *SG*가 지배될 때까지 다른 모든 하위 배열 그룹이 {*SG'*|*(addr (SG'.UB)* | *addr (SG.LB)) = addr (SG.LB)*}이 성립함을 확인하기 때문이다. 여기서, '|'의 의미는 비트 단위의 연산을 의미한다. 이와 관련하여, 본 발명은 상대적으로 연산 속도가 굉장히 빠른 비트 단위의 연산을 통해 불필요한 MBR 단위의 지배 관계 비교 및 하위 배열 단위의 지배 관계 비교를 생략할 수 있다는 장점이 있다.

[0044] 도 6은 본 발명에 따른 특정 하위 배열을 탐색하는 개념도를 나타낸다. 도 6(a)를 참조하면, 다른 하위 배열 그룹들에 의해 지배되지 않는 의미 있는 특정 하위 배열 그룹들(SG-Skyline)을 나타낸 것이다. 한편, 도 6(b)를 참조하면, 특정 하위 배열이 다른 하위 배열에 의해 지배되지 않는 특정 하위 배열(Subarray-Skyline)의 탐색 결과이다.

[0045] 한편, 도 7은 본 발명의 다양한 예시에 따른 3가지 경우의 하위 배열 그룹 간의 부분 지배 관계를 나타낸다. 도 7(a), (b)를 참조하면, 다른 하위 배열에 의해 지배되는 하위 배열들을 폐기(discard)하면서 SG1의 하위 배열들의 모든 쌍들을 비교할 수 있다. SG1. LB가 SG2. LB를 지배하면, 모든 SG2의 하위 배열들을 SG1의 하위 배열들과 비교한다. 반면에, 도 7(c)를 참조하면, SG1. LB가 SG2. LB를 지배하지 않는 경우가 성립할 수도 있다.

[0046] 이와 관련하여, 도 8a 및 도 8b는 본 발명의 일 예시에 따른 하위 배열을 탐색하는 알고리즘을 나타낸다. 구체적으로, 도 8a는 하위 배열을 탐색하는 알고리즘 1 (SUB_SKY)을 나타낸다. 또한, 도 8b는 하위 배열이 다른 하위 배열에 의해 지배되는지 여부에 따 서로 다른 논리 값, 즉 true/false를 반환하는 알고리즘 2 (SET_{SUB})를 나타낸다.

[0047] 본 발명은 도 1a 내지 도 8b에서 설명한 바와 같이 배열 데이터에 대한 하위 배열 스카이라인을 탐색하는 효율적인 방법, 즉 ReSKY에 대한 것이다. 따라서, 본 발명은 기존의 스카이라인 질의와는 다르게 하위 배열 단위로 스카이라인을 계산함으로써 새로운 배열 탐색 방법을 제공하는 것이다. 이를 위해, 본 발명에서는 먼저 하위 배열 단위의 지배 관계를 정의하였다. 요약하면, 하위 배열 (그룹)에 속하는 속성의 집계 값을 바탕으로 하위 배열 (그룹) 간의 지배 관계를 정의하였다. 하위 배열 (그룹) 간의 지배 관계에 대해서 후술할 도 11a에서 상세하게 다시 설명한다.

[0048] 한편, 본 발명에서는 MBR(Minimum Bounding Rectangle) 단위의 두 가지 지배 관계를 정의하였다. 이러한 지배 관계의 판단을 통해 하위 배열을 그룹화하여 적용할 수 있으며, 결과적으로 전체 CPU 사용량을 낮출 수 있다.

[0049] 전술한 바와 같이 정의된 지배 관계를 바탕으로 ReSKY의 핵심 개념을 도 11b의 Observation 1 & 2에서 상세하게 다시 설명한다. ReSKY의 핵심적인 개념은 상대적으로 연산 속도가 굉장히 빠른 비트 단위의 연산(bit-wise operation)을 수행하는 것이다. 이러한 연산 속도가 굉장히 빠른 비트 단위의 연산을 통해 불필요한 MBR 단위의 지배 관계 비교, 및 하위 배열 단위의 지배 관계 비교를 생략하는 것이다.

[0050] 한편, 본 발명에 따른 배열 데이터베이스에서의 하위 배열 스카이라인 질의 처리 방법은 분산 처리(distributed processing)에 적용될 수 있다. 이와 관련하여, 배열 데이터베이스의 청크 전략(chunk strategy)이 적용될 수

있다. 도 9는 본 발명의 일 예시에 따른 배열이 청크 단위로 배분되어 각각의 하위 프로세서 노드로 배분되는 개념도를 나타낸다. 도 9를 참조하면, 배열은 {1, 1} 크기의 청크 중첩(chunk overlap)을 갖는 4x4 크기의 청크들로 분배될 수 있다. 4개의 청크 세트들과 12개의 청크들이 2개의 하위 프로세서들, 제1 노드(Node 1) 및 제2 노드(Node 2)로 분배될 수 있다.

[0051] 또한, 본 발명에 따른 배열 데이터베이스에서의 하위 배열 스카이라인 질의 처리 방법에서 ReSKY, 즉 특정 하위 배열 그룹들의 탐색도 분산 처리 방식으로 이루어질 수 있다. 이와 관련하여, 도 10a는 특정 하위 배열 그룹들 및 특정 하위 배열의 탐색과 관련된 절차를 나타낸다. 한편, 도 10b는 본 발명에 따른 하위 배열 스카이라인 질의 처리 방법을 수행하는 질의 처리 장치의 구성을 나타낸다. 도 10b를 참조하면, 질의 처리 장치는 상위 프로세서에 해당하는 코디네이터 프로세서(100) 및 복수의 하위 프로세서들(200)을 포함하도록 구성될 수 있다. 복수의 하위 프로세서들(200)은 제1 프로세서(210) 및 제2 프로세서(220)를 포함할 수 있다. 한편, 복수의 하위 프로세서들(200)의 개수는 특정 개수에 한정되는 것은 아니고 분산 처리되는 독립적인 작업(task)의 수에 따라 n개의 프로세서로 확장될 수 있다.

[0052] 도 10a 및 도 10b를 참조하면, worker, 즉 worker node는 복수의 하위 프로세서들(200) 중 하나에 해당할 수 있다. coordinator는 상위 프로세서에 해당하는 코디네이터 프로세서(100)로 구현될 수 있다.

[0053] 도 10a에 도시된 단계들 (S1 내지 S7)은 다음과 같이 구체화될 수 있다. 각각의 하위 프로세서가 각각의 하위 배열의 집계 속성 값(aggregated attribute values)을 연산하여 하위 배열 그룹을 형성하는 하위 배열 그룹 형성 과정(S1)이 수행된다. 각각의 하위 프로세서가 로컬 SG-Skyline을 탐색하는 제1 탐색 과정(S2)이 수행된다. 각각의 하위 프로세서가 자신의 SG들을 가지치기(pruning)할 수 있도록, 코디네이터 프로세서가 글로벌 샘플 SG-Skyline을 연산하는 글로벌 샘플 SG-Skyline 연산 과정(S3)이 수행된다. 또한, 각각의 하위 프로세서가 로컬 하위배열-Skyline을 탐색하는 제2 탐색 과정(S4)이 수행된다.

[0054] 한편, 각각의 하위 프로세서가 상기 각각의 하위 배열을 가지치기(pruning)할 수 있도록, 코디네이터 프로세서가 글로벌 샘플 SG-하위 배열을 연산하는 글로벌 샘플 SG-하위 배열 연산 과정(S5)이 수행된다. 또한, 코디네이터 프로세서가 로컬 하위 배열-Skyline을 수집하는 로컬 하위 배열-Skyline 수집 과정(S6)이 수행된다. 이에 따라, 코디네이터 프로세서가 글로벌 하위 배열-Skyline을 결정하는 글로벌 하위 배열-Skyline을 결정 과정(S7)이 수행된다.

[0055] 전술한 과정들(S1 내지 S7)이 수행됨에 따라 분산 처리 기반의 하위 배열 스카이라인 질의 처리 방법이 수행될 수 있다. 전술한 기술적 특징들을 고려하여 본 발명에서 청구하고자 하는 배열 데이터베이스에서의 하위 배열 스카이라인 질의 처리 방법에 대해 설명한다. 이와 관련하여, 도 11a는 하위 배열 지배(subarray dominance)와 MBR 지배 및 MBR 부분 지배에 관한 정의를 나타낸다. 한편, 도 11b는 제1 및 제2 MBR 간의 지배 관계의 판단과 관련된 기준을 나타낸다.

[0056] 도 11a를 참조하면, 제1 및 제2 속성 값에 한정되는 것은 아니고 m-차원의 n 개의 속성들로 확장될 수 있다. 이와 관련하여, n 개의 속성들 중 선택된 k개의 속성들 {A'_1, A'_2, …, A'_k}에 대해 하위 배열(SA)의 표현(representation)은 튜플(tuple) SA.R = (SA.R_1, SA.R_2, …, SA.R_k)로 나타낼 수 있다. 튜플 단위로 SA_1.R이 SA_2.R을 지배(dominate)하면, 하위 배열 SA_1이 다른 하위 배열 SA_2를 지배한다고 할 수 있다.

[0057] MBR 지배와 관련하여, MBR1. UB가 MBR2. LB를 지배하면, MBR1이 MBR2를 지배한다고 할 수 있다. 즉, MBR1의 임의의 튜플이 MBR2의 모든 튜플들을 지배한다는 것을 의미하기 때문에, MBR1이 MBR2를 지배한다고 할 수 있다.

[0058] MBR 부분 지배(partial dominance)와 관련하여, MBR1. LB가 MBR2. UB를 지배하면, MBR1이 MBR2를 부분적으로 지배한다고 할 수 있다. 즉, MBR1의 일부 튜플들이 MBR2의 튜플들을 지배할 수 있다는 것을 의미하기 때문에, MBR1이 MBR2를 부분적으로 지배한다고 할 수 있다.

[0059] 도 11b를 참조하면, *(addr (MBR1.UB, PV) | addr (MBR2.LB, PV)) = addr (MBR2.LB PV)*가 성립하면, MBR1이 MBR2를 지배한다고 판단될 수 있다. 또한, *(addr (MBR1.LB, PV) | addr (MBR2.UB, PV)) = addr (MBR2.UB, PV)* 가 성립하면, MBR1 이 MBR2를 부분적으로 지배한다고 판단될 수 있다. 여기서, '|' 의 의미는 비트 단위의 연산을 의미한다. 이와 관련하여, 본 발명은 상대적으로 연산 속도가 굉장히 빠른 비트 단위의 연산을 통해 불필요한 MBR 단위의 지배 관계 비교 및 하위 배열 단위의 지배 관계 비교를 생략할 수 있다는 장점이 있다.

[0060] 한편, 도 12는 본 발명에 따른 배열 데이터베이스에서의 하위 배열 스카이라인 질의 처리 방법의 흐름도를 나타

낸다. 도 12를 참조하면, 질의 처리 방법은 하위 배열 그룹 구성 과정(S100), 하위 배열 그룹 탐색 과정(S200), 특정 하위 배열 연산 과정(S300) 및 분산 처리 제어 과정(S400)을 포함할 수 있다. 분산 처리 제어 과정(S400)과 관련하여, 본 발명에 따른 하위 배열 그룹 및 하위 배열 단위의 질의 처리 방법의 원리가 분산 노드에서의 처리로 확장될 수 있다. 이를 "distributed processing steps of ReSKY"으로 지칭할 수 있다. 본 발명을 통해 분산 처리에서 가장 문제가 되는 네트워크 사용량과 로드 밸런싱(load balancing, 각 노드가 균등한 양의 일을 처리할 수 있도록 작업을 효과적으로 분배하는 문제)을 해결할 수 있다. 따라서, 본 발명에서는 네트워크 사용량과 로드 밸런싱 문제를 고려하여 샘플링(sampling) 기반의 새로운 질의 처리 방법을 제안할 수 있다.

[0061] 한편, 질의 처리 방법에서 하위 배열 그룹 구성 과정(S100)은 일정 주기마다 또는 이벤트 기반으로 반복적으로 수행될 수 있다. 이와 관련하여, 코디네이터 프로세서는 복수의 하위 프로세서들로부터 수신된 정보 및 전체 시스템의 성능 평가 결과에 따라 하위 배열 그룹이 재구성될 필요가 있는지 판단하는 하위 배열 그룹 재구성 판단 과정(S500)을 더 수행할 수도 있다.

[0062] 따라서, 질의 처리 방법은 하위 배열 그룹 재구성 판단 과정(S500)을 더 포함할 수 있다. 하위 배열 그룹 재구성 판단 과정(S500)에서, 하위 배열 그룹의 재구성이 필요하다고 판단되면 하위 배열 그룹 구성 과정(S100) 이하의 과정을 다시 반복적으로 수행할 수 있다. 이와 관련하여, 일부 과정들은 이전 수행 결과를 이용하여 수행되거나 다른 일부 과정들은 생략될 수도 있다. 하위 배열 그룹 재구성 판단 과정(S500)에서, 하위 배열 그룹의 재구성이 필요하지 않다고 판단되면 특정 하위 배열 연산 과정(S300) 및 분산 처리 제어 과정(S400)을 더 수행할 수도 있다.

[0063] 하위 배열 그룹 구성 과정(S100)에서, 질의 처리와 연관된 제1 속성 값(attribute value) 및 제2 속성 값에 기초하여 구성된 복수의 하위 배열 (sub array)에 대하여 하위 배열 그룹들을 구성할 수 있다. 하위 배열 그룹 탐색 과정(S200)에서, 하위 배열 그룹들의 세트들에서 특정 하위 배열 그룹들의 MBR (Minimum Bounding Rectangle)이 다른 하위 배열 그룹들의 MBR 에 의해 지배(dominate)되지 않는 특정 하위 배열 그룹들을 탐색할 수 있다.

[0064] 특정 하위 배열 연산 과정(S300)에서, 특정 하위 배열 그룹들의 각각에 대해 특정 하위 배열이 다른 하위 배열에 의해 지배되지 않는 상기 특정 하위 배열을 연산할 수 있다. 또한, 분산 처리 제어 과정(S400)에서 복수의 하위 프로세서들 각각이 상기 특정 하위 배열 그룹들의 각각의 작업(task)을 분산 처리하도록 상기 복수의 하위 프로세서들을 제어할 수 있다. 이와 관련하여, 특정 하위 배열 그룹의 탐색 결과 및 상기 특정 하위 배열의 연산 결과에 기초하여, 복수의 하위 프로세서들 각각이 특정 하위 배열 그룹들의 각각의 작업(task)을 분산 처리하도록 할 수 있다.

[0065] 하위 배열 그룹 탐색 과정(S200)에서, 제1 하위 배열 그룹에 대응하는 제1 MBR의 제1 상한 및 제1 하한과 제2 하위 배열 그룹에 대응하는 제2 MBR의 제2 상한 및 제2 하한을 비교할 수 있다. 따라서 하위 배열 그룹 탐색 과정(S200)에서, 제1 하위 배열 그룹과 제2 하위 배열 그룹 간의 지배 관계를 판단할 수 있다.

[0066] 지배 관계 판단과 관련하여, 제1 상한을 제2 하한과 비교하여, 제1 하위 배열 그룹이 제2 하위 배열 그룹을 제1 속성 값 및 제2 속성 값에 대해 지배하는지 판단할 수 있다. 부분 지배 관계 판단과 관련하여, 제1 하한을 제2 상한과 비교하여, 제1 하위 배열 그룹이 제2 하위 배열 그룹을 제1 속성 값 및 제2 속성 값 중 어느 하나에 대해 부분 지배하는지 판단할 수 있다.

[0067] 한편, 하위 배열 연산 과정(S300)에서, 특정 하위 배열 그룹들의 각각에 대해 제1 하위 배열에 대응하는 제1 MBR의 제1 상한 및 제1 하한과 제2 하위 배열에 대응하는 제2 MBR의 제2 상한 및 제2 하한을 비교할 수 있다. 따라서 하위 배열 연산 과정(S300)에서, 제1 하위 배열과 상기 제2 하위 배열 간의 지배 관계를 판단할 수 있다.

[0068] 하위 배열 그룹 구성 과정(S100)은 코디네이터 프로세서가 하위 배열 그룹을 형성하여 수행될 수 있다. 하위 배열 그룹 구성 과정(S100)에서, 복수의 하위 프로세서들 각각이 하위 배열의 집계 속성 값(aggregated attribute values)을 연산한 값에 기초하여, 코디네이터 프로세서가 하위 배열 그룹을 형성할 수 있다. 하위 배열 그룹 탐색 과정(S200)에서, 코디네이터 프로세서가 복수의 하위 프로세서들로부터 수집되어 샘플링된 제1 로컬 정보에 기초하여, 글로벌 샘플링 기반의 상기 특정 하위 배열 그룹의 탐색을 수행할 수 있다.

[0069] 특정 하위 배열 연산 과정(S300)에서, 코디네이터 프로세서가 복수의 하위 프로세서들로부터 수집되어 샘플링된 제2 로컬 정보에 기초하여, 특정 하위 배열 그룹들의 각각에 대해 글로벌 샘플링 기반의 특정 하위 배열을 연산할 수 있다.

[0070] 하위 배열 그룹 구성 과정(S100)에서, 코디네이터 프로세서가 네트워크 사용량과 복수의 하위 프로세서들의 로드 상태에 기초한 로드 밸런싱을 수행하여, 하위 배열 그룹들을 구성할 수도 있다. 이와 관련하여, 분산 처리 제어 과정(S400)에서, 로드 밸런싱에 따라 구성된 하위 배열 그룹들에서 다른 하위 배열 그룹에 의해 지배되지 않는 특정 하위 배열 그룹들의 각각을 복수의 하위 프로세서들 각각이 분산 처리할 수 있다. 또한, 분산 처리 제어 과정(S400)에서, 특정 하위 배열을 복수의 하위 프로세서들 각각이 다른 하위 배열보다 우선적으로 처리하도록 제어할 수 있다. 이와 관련하여, 샘플링된 제1 로컬 정보 및 샘플링된 제2 로컬 정보에 기초하여 글로벌 샘플링 기반으로 연산된 특정 하위 배열을 복수의 하위 프로세서들 각각이 다른 하위 배열보다 우선적으로 처리하도록 제어할 수 있다. 제1 로컬 정보 및 제2 로컬 정보는 각각 하위 배열 그룹 탐색 과정(S200) 및 특정 하위 배열 연산 과정(S300)에서 복수의 하위 프로세서들로부터 수집되어 샘플링된 정보이다.

[0071] 본 명세서의 또 다른 양상에 따른 질의 처리 장치에 대해 설명한다. 이와 관련하여, 도 10b를 참조하면 질의 처리 장치(1000)는 상위 프로세서에 해당하는 코디네이터 프로세서(100) 및 복수의 하위 프로세서들(200)을 포함하도록 구성될 수 있다. 복수의 하위 프로세서들(200)은 제1 프로세서(210) 및 제2 프로세서(220)를 포함할 수 있다. 한편, 복수의 하위 프로세서들(200)의 개수는 특정 개수에 한정되는 것은 아니고 분산 처리되는 독립적인 작업(task)의 수에 따라 n개의 프로세서로 확장될 수 있다.

[0072] 한편, 질의 처리 장치(1000)는 질의(query)와 연관된 메시지를 수신하도록 구성된 인터페이스(10)를 더 포함할 수 있다. 한편, 코디네이터 프로세서(100)와 복수의 하위 프로세서들(200) 간에 인터페이스(10b)가 더 구비될 수 있다. 코디네이터 프로세서(100) 및 복수의 하위 프로세서들(200) 각각은 프로세서와 메모리를 구비한 프로세서 장치일 수 있다.

[0073] 복수의 하위 프로세서들(200)은 코디네이터 프로세서(100)와 동작 가능하게 결합되고, 분산 처리를 수행하도록 구성될 수 있다. 코디네이터 프로세서(100)는 질의 처리와 연관된 제1 속성 값(attribute value) 및 제2 속성 값에 기초하여 구성된 복수의 하위 배열 (sub array)에 대하여 하위 배열 그룹들을 구성할 수 있다.

[0074] 코디네이터 프로세서(100)는 하위 배열 그룹들의 세트들에서 특정 하위 배열 그룹들의 MBR (Minimum Bounding Rectangle)이 다른 하위 배열 그룹들의 MBR 에 의해 지배(dominate)되지 않는 특정 하위 배열 그룹들을 탐색할 수 있다. 또한, 코디네이터 프로세서(100)는 특정 하위 배열 그룹들의 각각에 대해 특정 하위 배열이 다른 하위 배열에 의해 지배되지 않는 특정 하위 배열을 연산할 수 있다.

[0075] 이에 따라, 코디네이터 프로세서(100)는 복수의 하위 프로세서들(200) 각각이 특정 하위 배열 그룹들의 각각의 작업(task)을 분산 처리하도록 복수의 하위 프로세서들(200)을 제어할 수 있다. 이때, 코디네이터 프로세서(100)는 특정 하위 배열 그룹의 탐색 결과 및 특정 하위 배열의 연산 결과에 기초하여, 작업(task)을 분산 처리하도록 복수의 하위 프로세서들(200)을 제어할 수 있다.

[0076] 코디네이터 프로세서(100)는 제1 하위 배열 그룹에 대응하는 제1 MBR의 제1 상한 및 제1 하한과 제2 하위 배열 그룹에 대응하는 제2 MBR의 제2 상한 및 제2 하한을 비교할 수 있다. 따라서 코디네이터 프로세서(100)는 제1 하위 배열 그룹과 제2 하위 배열 그룹 간의 지배 관계를 판단할 수 있다.

[0077] 코디네이터 프로세서(100)는 제1 상한을 제2 하한과 비교하여, 제1 하위 배열 그룹이 제2 하위 배열 그룹을 제1 속성 값 및 제2 속성 값에 대해 지배하는지 판단할 수 있다. 코디네이터 프로세서(100)는 제1 하한을 제2 상한과 비교하여, 제1 하위 배열 그룹이 제2 하위 배열 그룹을 제1 속성 값 및 제2 속성 값 중 어느 하나에 대해 부분 지배하는지 판단할 수 있다.

[0078] 이상에서는 본 발명에 따른 E배열 데이터베이스에서의 하위 배열 스카이라인 질의 처리 방법 및 이를 수행하는 질의 처리 장치에 대해 설명하였다.

[0079] 본 발명의 실시예에 따르면, 기존에 존재하는 질의 처리 방법 대비 질의 처리 소요시간을 대폭 감소시킬 수 있다.

[0080] 또한, 본 발명의 실시예에 따르면, 기존 방법들이 제공하지 못하던 분산 처리를 지원함으로써 초대용량 배열 데이터에 대한 질의 처리도 가능하다.

[0081] 또한, 본 발명의 실시예에 따르면, 다양한 분야에서 생성되는 배열 데이터에 대한 빠른 분석을 통해 의사 결정 과정에 크게 기여할 수 있다.

[0082] 상술한 본 발명의 특징 및 효과는 첨부된 도면과 관련한 다음의 상세한 설명을 통하여 보다 분명해 질 것이며, 그에 따라 본 발명이 속하는 기술 분야에서 통상의 지식을 가진 자가 본 발명의 기술적 사상을 용이하게 실시할

수 있을 것이다.

[0083] 본 발명은 다양한 변경을 가할 수 있고 여러 가지 실시 예를 가질 수 있는바, 특정 실시 예들을 도면에 예시하고 상세한 설명에 구체적으로 설명하고자 한다. 그러나 이는 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.

[0084] 소프트웨어적인 구현에 의하면, 본 명세서에서 설명되는 절차 및 기능뿐만 아니라 각각의 구성 요소들에 대한 설계 및 파라미터 최적화는 별도의 소프트웨어 모듈로도 구현될 수 있다. 적절한 프로그램 언어로 쓰여진 소프트웨어 어플리케이션으로 소프트웨어 코드가 구현될 수 있다. 상기 소프트웨어 코드는 메모리에 저장되고, 제어부(controller) 또는 프로세서(processor)에 의해 실행될 수 있다.

부호의 설명

[0085] 100: 코디네이터 프로세서

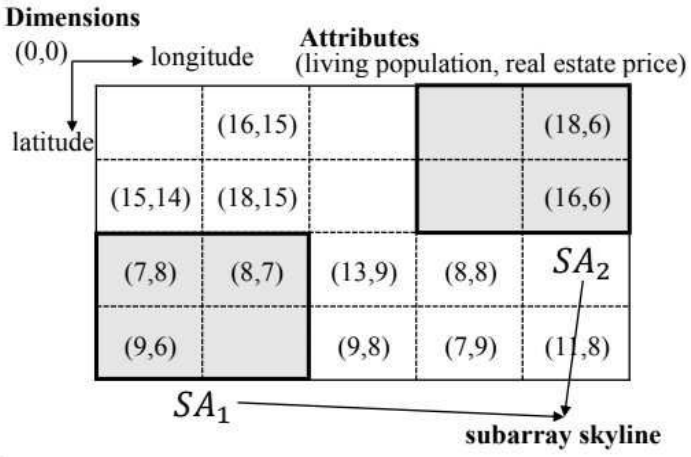
200: 하위 프로세서들

210: 제1 프로세서

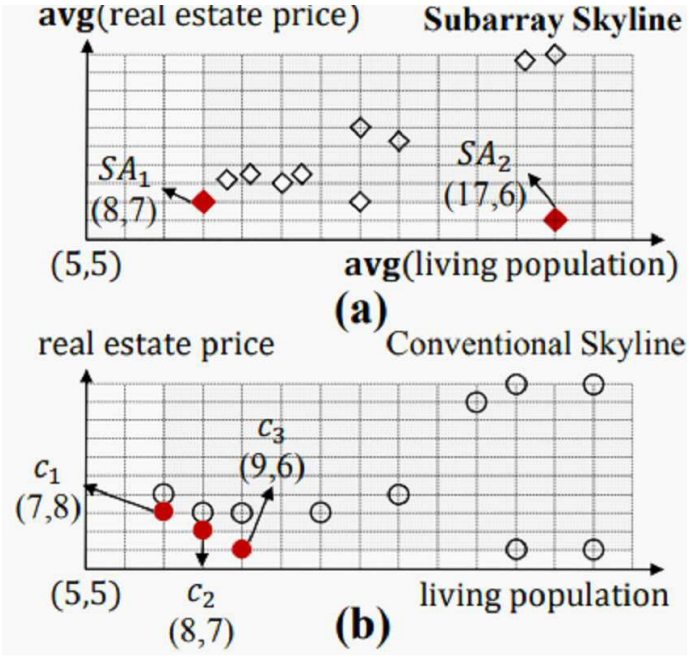
220: 제2 프로세서

도면

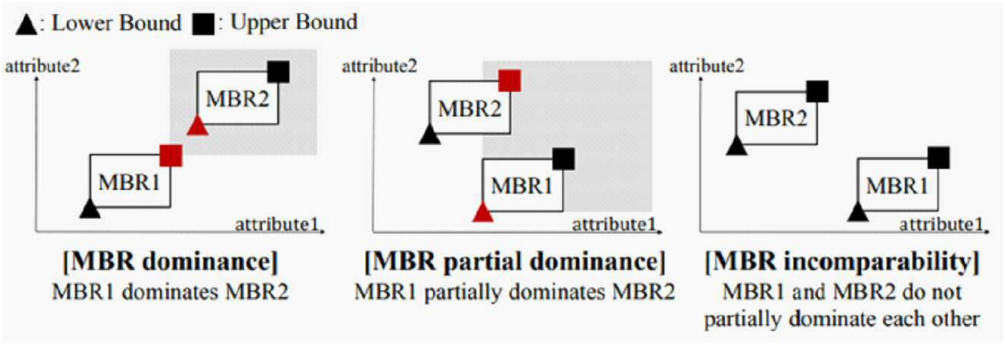
도면1a



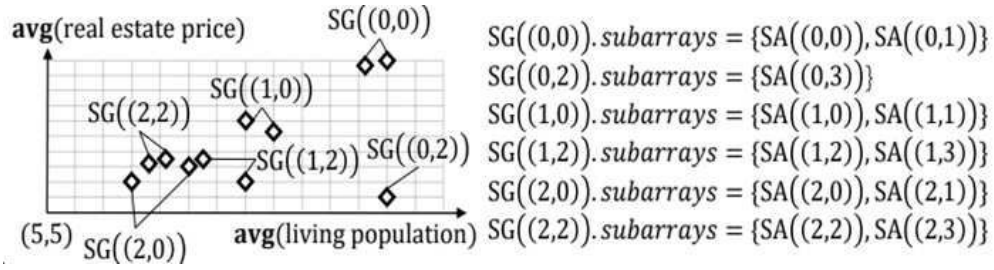
도면1b


avg(real estate price)
Subarray Skyline
SA_1
(8,7)
SA_2
(17,6)
(5,5)
avg(living population)
(a)
real estate price
Conventional Skyline
c_3
(9,6)
c_1
(7,8)
(5,5)
c_2
(8,7)
living population
(b)

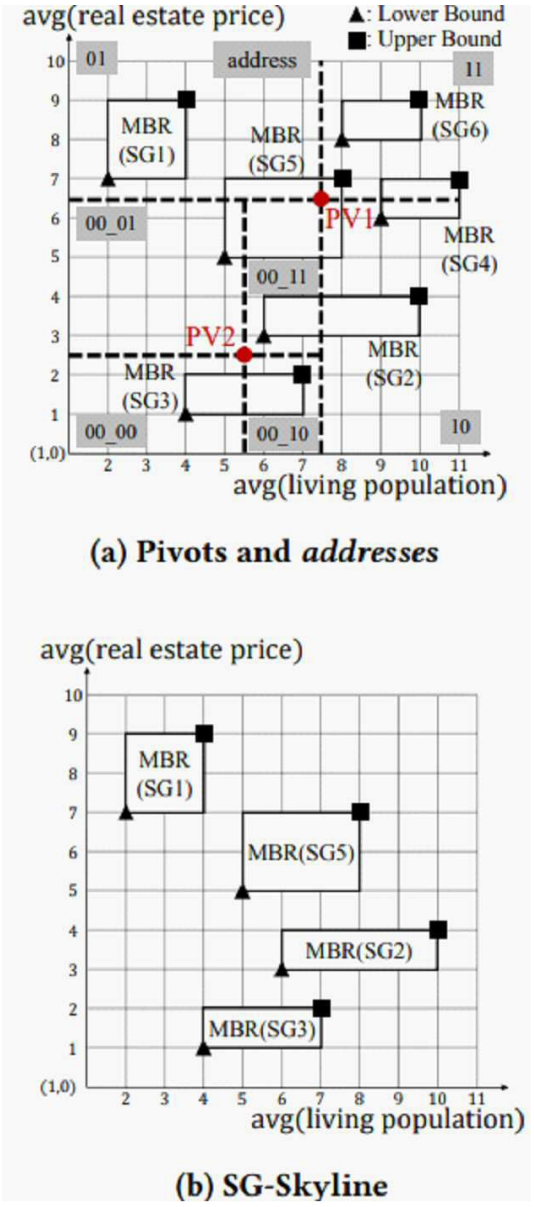
도면2


▲: Lower Bound ■: Upper Bound
attribute2
MBR2
MBR1
attribute1
[MBR dominance]
MBR1 dominates MBR2
[MBR partial dominance]
MBR1 partially dominates MBR2
[MBR incomparability]
MBR1 and MBR2 do not partially dominate each other

도면3


avg(real estate price)
SG((0,0))
SG((1,0))
SG((2,2))
SG((1,2))
SG((0,2))
(5,5)
SG((2,0))
avg(living population)
$SG((0,0)).subarrays = \{SA((0,0)), SA((0,1))\}$
$SG((0,2)).subarrays = \{SA((0,3))\}$
$SG((1,0)).subarrays = \{SA((1,0)), SA((1,1))\}$
$SG((1,2)).subarrays = \{SA((1,2)), SA((1,3))\}$
$SG((2,0)).subarrays = \{SA((2,0)), SA((2,1))\}$
$SG((2,2)).subarrays = \{SA((2,2)), SA((2,3))\}$

도면4


avg(real estate price)
▲: Lower Bound
■: Upper Bound
01
address
11
MBR (SG1)
MBR (SG5)
MBR (SG6)
PV1
00_01
MBR (SG4)
00_11
PV2
MBR (SG2)
MBR (SG3)
00_00
00_10
10
(1,0)
avg(living population)
(a) Pivots and addresses
avg(real estate price)
MBR (SG1)
MBR(SG5)
MBR(SG2)
MBR(SG3)
(1,0)
avg(living population)
(b) SG-Skyline

도면5

Algorithm 1: *SG_SKY*

Input : a set of *LB*s and *UB*s
Output : *SG-Skyline*

```
1  SG-Skyline ← ∅;
2  Select median of all LBs and UBs in the set as a pivot;
3  Calculate addresses of LBs and UBs regarding the pivot, and
     divide them into subsets according to their addresses;
4  foreach S ∈ subsets in ascending order of addresses do
5  |  foreach LB ∈ S (and its corresponding SG) and
   |    foreach S' ∈ dominating subsets(S,pivot)-S,
   |    if CHECK_SET_SG(LB, S') = true,
   |    delete LB from S and SG.
6  |  if the number of LBs in S ≤ 1 then
7  |  |  Check whether LB is dominated by a UB in S;
8  |  |  SG-Skyline ← SG-Skyline ∪ {LB's SG};
9  |  else
10 |  |  SG-Skyline ← SG-Skyline ∪ SG_SKY(S);
11 |  end
12 end
13 return SG-Skyline;
```

(a)

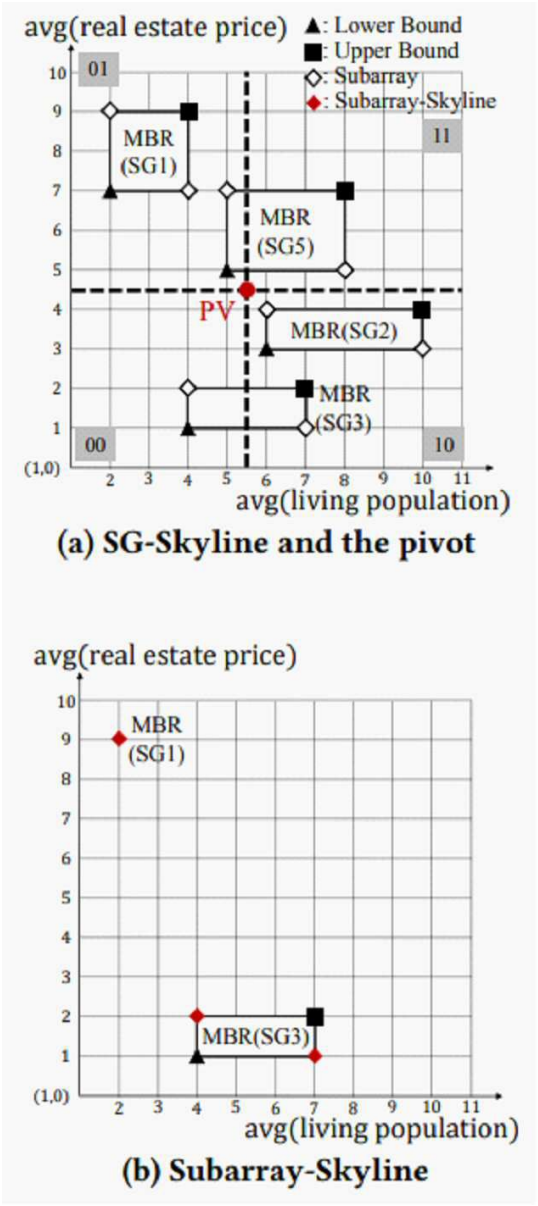
Algorithm 2: *CHECK_SET*$_{SG}$

Input : *LB*, a set *S* of *LB*s and *UB*s
Output : *true* if *LB* is dominated, else *false*

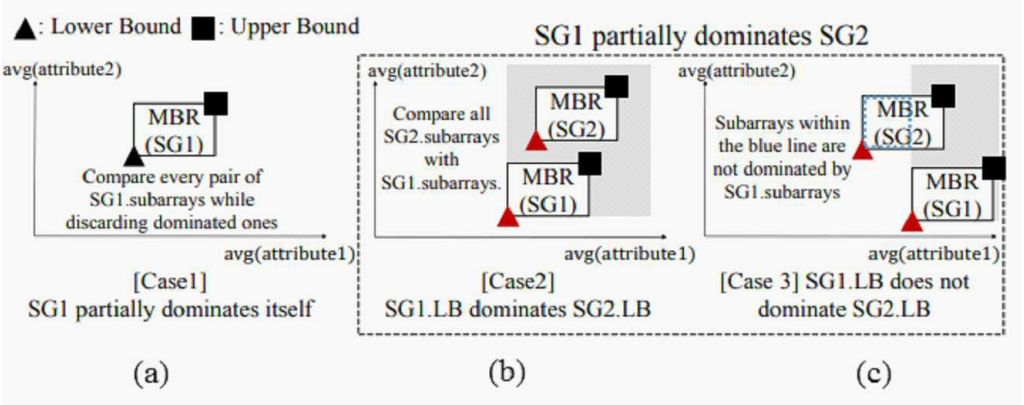
```
1  if S is not divided anymore then
2  |  if LB is dominated by a UB in S, return true;
3  else
4  |  Calculate the subset S_LB of LB regarding the pivot of S;
5  |  foreach S' ∈ dominating subsets(S_LB,pivot of S) do
6  |  |  if CHECK_SET_SG(LB, S'), return true;
7  |  end
8  end
9  return false;
```

(b)

도면6


avg(real estate price)
▲: Lower Bound
■: Upper Bound
◇: Subarray
◆: Subarray-Skyline
01
11
00
10
MBR (SG1)
MBR (SG5)
PV
MBR(SG2)
MBR (SG3)
(1,0)
avg(living population)
(a) SG-Skyline and the pivot
avg(real estate price)
MBR (SG1)
MBR(SG3)
(1,0)
avg(living population)
(b) Subarray-Skyline

도면7


▲: Lower Bound ■: Upper Bound
SG1 partially dominates SG2
avg(attribute2)
MBR (SG1)
Compare every pair of SG1.subarrays while discarding dominated ones
avg(attribute1)
[Case1]
SG1 partially dominates itself
(a)
avg(attribute2)
Compare all SG2.subarrays with SG1.subarrays.
MBR (SG2)
MBR (SG1)
avg(attribute1)
[Case2]
SG1.LB dominates SG2.LB
(b)
avg(attribute2)
Subarrays within the blue line are not dominated by SG1.subarrays
MBR (SG2)
MBR (SG1)
avg(attribute1)
[Case 3] SG1.LB does not dominate SG2.LB
(c)

도면8a

Algorithm 3: *SUB_SKY*

Input : a set of *LBs* and *UBs*
Output : *Subarray-Skyline*

Subarray-Skyline ← ∅;
Select median of all *LBs* and *UBs* in the set as a pivot;
Calculate *addresses* of *LBs* and *UBs* regarding the pivot, and divide them into subsets according to their *addresses*;
foreach *S* ∈ *subsets in ascending order of addresses* **do**
  foreach *UB* ∈ *S* (and its corresponding *SG*) and **foreach** *S′* ∈ **dominating subsets**(*S*,pivot)-*S*, if $CHECK_SET_{SUB}(SG, S') = true$, delete *UB* from *S* and *SG*.
  if *the number of UBs in S* ≤ 1 **then**
    For the subarray group *SG* of *UB* and each subarray group *SG′* that partially dominates *SG* where *SG′.LB* ∈ *S*, check whether *SG.subarrays* are dominated by *SG′.subarrays*;
    Subarray-Skyline ← *Subarray-Skyline*∪*SG.subarrays*;
    recalculate *MBR*(*SG*) if it may shrink;
  else
    Subarray-Skyline ← *Subarray-Skyline*∪*SUB_SKY*(*S*);
  end
end
return *Subarray-Skyline*;

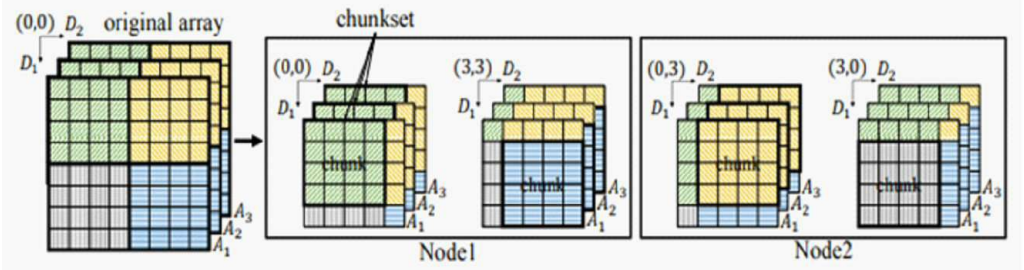
도면8b

Algorithm 4: $CHECK_SET_{SUB}$

Input : *SG*, a set *S* of *LBs* and *UBs*
Output : *true* if |*SG.subarrays*| = ∅; else, *false*

if *S is not divided anymore* **then**
  foreach subarray group *SG′* that *SG′.LB* ∈ *S* dominates *SG.UB* (= *SG′* partially dominates *SG*), check whether *SG.subarrays* are dominated by *SG′.subarrays*. If |*SG.subarrays*| = ∅, return true;
else
  Calculate the subset $S_{SG.UB}$ of *SG.UB* regarding the pivot of *S*;
  foreach *S′* ∈ ***dominating subsets***($S_{SG.UB}$,*pivot of S*) **do**
    if $CHECK_SET_{SUB}(SG, S')$, return *true*;
  end
end
return *false*;

도면9

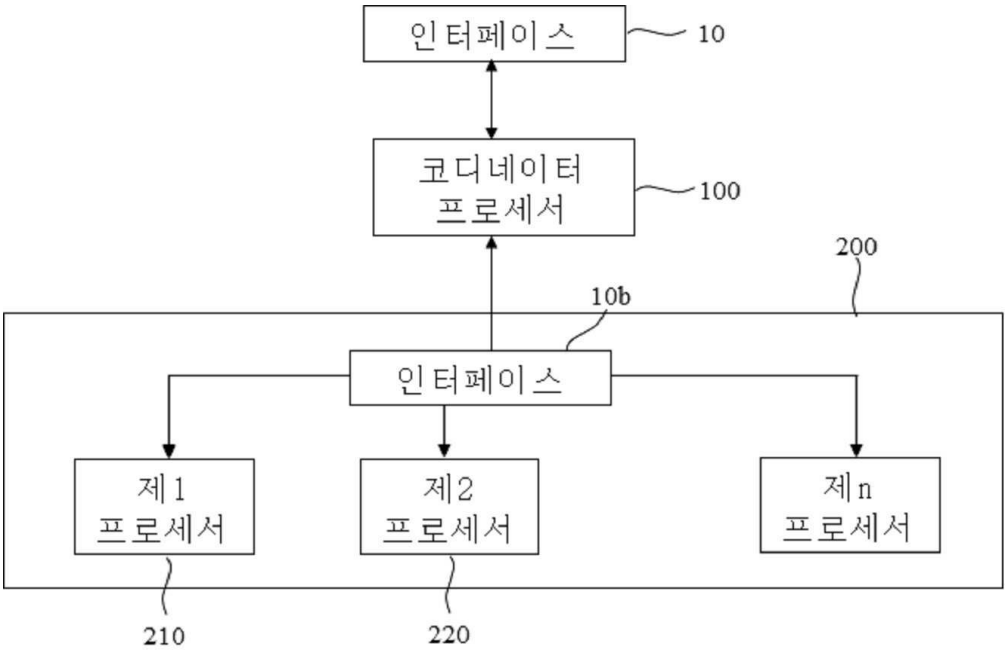


도면10a

Distributed processing steps of ReSKY

Step1 (S1): Each worker computes representatives of its own subarrays and forms the subarray groups.
Step2 (S2): Each worker finds local SG-Skyline.
Step3 (S3): The coordinator computes global sample SG-Skyline, which enables each worker to prune its own SGs.
Step4 (S4): Each worker finds local Subarray-Skyline.
Step5 (S5): The coordinator computes global sample Subarray-Skyline, which enables each worker to prune its own subarrays.
Step6 (S6): The coordinator gathers local Subarray-Skylines.
Step7 (S7): The coordinator determines global Subarray-Skyline.

도면10b



도면11a

Definition 2.3 (***Subarray Dominance***). Suppose (1) an array with m dimensions and n attributes, (2) a subarray size, and (3) $k(\leq n)$ selected attributes $\{A'_1, A'_2, ..., A'_k\}$ with corresponding aggregate functions $f = \{f_1, f_2, ..., f_k\}$. The **representative of a subarray** SA is a tuple $SA.R = (SA.R_1, SA.R_2, ..., SA.R_k)$ where $SA.R_i$ is f_i output value for all A'_i values in SA. A subarray SA_1 dominates another subarray SA_2 iff $SA_1.R$ dominates $SA_2.R$. Iff $SA_1.R$ and $SA_2.R$ are incomparable, then SA_1 and SA_2 are incomparable. □

Definition 4.1 (***MBR Dominance***). Given two MBRs, *MBR*1 and *MBR*2, iff *MBR*1.*UB* dominates *MBR*2.*LB*, *MBR*1 dominates *MBR*2, which means that any tuple in *MBR*1 dominates all tuples in *MBR*2.

Definition 4.2 (***MBR Partial Dominance***). Given two MBRs, *MBR*1 and *MBR*2, iff *MBR*1.*LB* dominates *MBR*2.*UB*, *MBR*1 partially dominates *MBR*2, which means that some tuples in *MBR*1 may dominate tuples in *MBR*2. □

도면11b

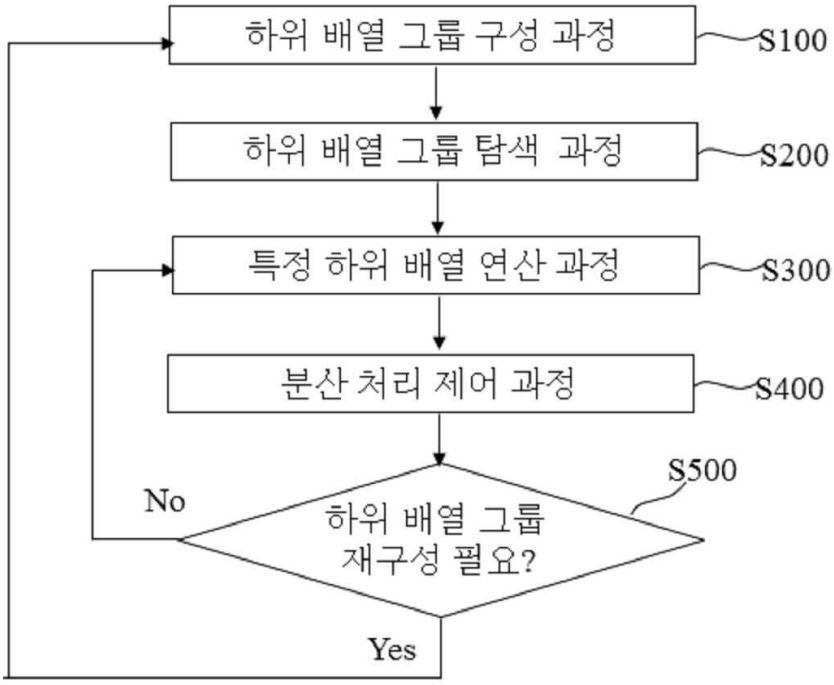
OBSERVATION 1. *Given (1) two MBRs, MBR*1 *and MBR*2*, and (2) a pivot PV, if MBR*1 *dominates MBR*2*,* $(addr(MBR1.UB, PV) \mid addr(MBR2.LB, PV)) = addr(MBR2.LB, PV)$.

Proof. *MBR*1.*UB* dominates *MBR*2.*LB* by Definition 4.1; therefore, $(addr(MBR1.UB, PV) \mid addr(MBR2.LB, PV)) = addr(MBR2.LB, PV)$ by Lemma 3.2. □

OBSERVATION 2. *Given (1) two MBRs, MBR*1 *and MBR*2*, and (2) a pivot PV, if MBR*1 *partially dominates MBR*2*,* $(addr(MBR1.LB, PV) \mid addr(MBR2.UB, PV)) = addr(MBR2.UB, PV)$.

Proof. *MBR*1.*LB* dominates *MBR*2.*UB* by Definition 4.2; therefore, $(addr(MBR1.LB, PV) \mid addr(MBR2.UB, PV)) = addr(MBR2.UB, PV)$ by Lemma 3.2. □

도면12


하위 배열 그룹 구성 과정
S100
하위 배열 그룹 탐색 과정
S200
특정 하위 배열 연산 과정
S300
분산 처리 제어 과정
S400
S500
No
하위 배열 그룹
재구성 필요?
Yes