(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2015-0061457
(43) 공개일자 2015년06월04일

(51) 국제특허분류(Int. Cl.)
G06F 17/00 (2006.01) *G06F 17/10* (2006.01)
(21) 출원번호 10-2013-0145615
(22) 출원일자 2013년11월27일
심사청구일자 2013년11월27일

(71) 출원인
경희대학교 산학협력단
경기도 용인시 기흥구 덕영대로 1732 (서천동, 경희대학교 국제캠퍼스내)
(72) 발명자
이승룡
경기 성남시 분당구 구미로144번길 25, 102동 102호 (구미동, 삼환빌라)
무하마드 비랄 아민
경기 용인시 기흥구 덕영대로 1732, 전자정보대학 351 (서천동, 경희대학교)
(74) 대리인
특허법인 신지

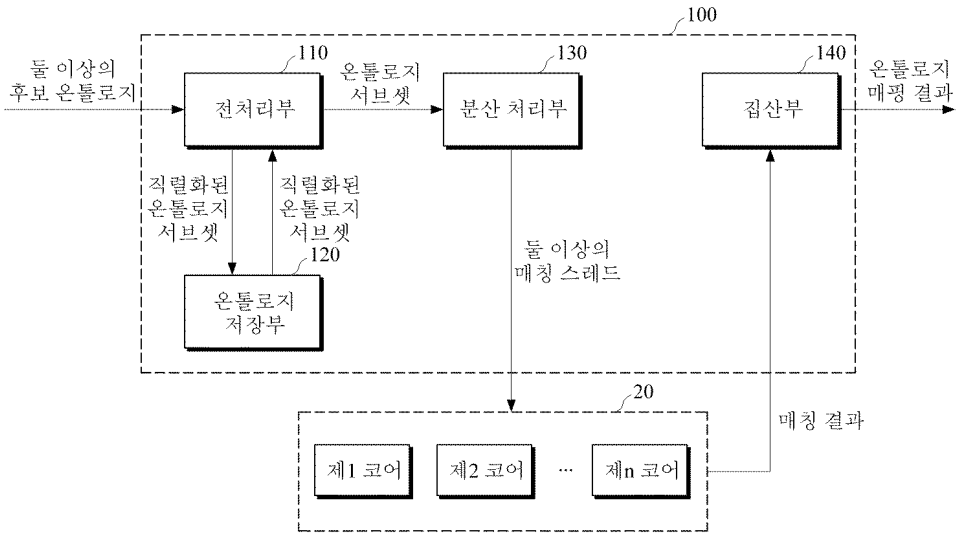
전체 청구항 수 : 총 19 항

(54) 발명의 명칭 **라지 스케일 생물의학 온톨로지 매칭 장치 및 매칭 방법**

(57) 요 약

본 발명에 따른 라지 스케일 온톨로지 매칭 장치는 수신된 후보 온톨로지를 하나 이상의 온톨로지 서브셋으로 분류하여 온톨로지 서브셋을 생성하는 전처리부, 생성된 온톨로지 서브셋을 분산 알고리즘을 적용하여 분할하고, 분할된 온톨로지 서브셋에 매칭 알고리즘을 적용하여 매칭 스레드를 생성하며, 생성된 매칭 스레드를 참여 노드의 개별 코어에 전달하는 분산 처리부 및 개별 코어에서 매칭 스레드에 기초하여 매칭 연산을 수행하여 생성된 매칭 결과를 수집 및 합산하여 온톨로지 매핑을 생성하는 집산부를 포함한다.

대 표 도



이 발명을 지원한 국가연구개발사업
과제고유번호 1415128832
부처명 지식경제부
연구관리전문기관 정보통신산업진흥원
연구사업명 대학 IT 연구센터 육성지원
연구과제명 동서신의학 u-라이프케어연구센터
기 여 율 1/1
주관기관 경희대학교 산학협력단
연구기간 2013.01.01 ~ 2013.12.31

명 세 서

청구범위

청구항 1

수신된 후보 온톨로지를 하나 이상의 온톨로지 서브셋으로 분류하여 온톨로지 서브셋을 생성하는 전처리부;

상기 생성된 온톨로지 서브셋을 분산 알고리즘을 적용하여 분할하고, 상기 분할된 온톨로지 서브셋에 매칭 알고리즘을 적용하여 매칭 스레드를 생성하며, 상기 생성된 매칭 스레드를 참여 노드의 개별 코어에 전달하는 분산처리부; 및

상기 개별 코어에서 상기 매칭 스레드에 기초하여 매칭 연산을 수행하여 생성된 매칭 결과를 수집 및 합산하여 온톨로지 매핑을 생성하는 집산부;

를 포함하는 것을 특징으로 하는 라지 스케일 온톨로지 매칭 장치.

청구항 2

제 1항에 있어서,

상기 전처리부는,

상기 생성된 온톨로지 서브셋을 바이너리(Binary) 형태로 직렬화(Serialization)하여 직렬화된 온톨로지 서브셋을 생성하는 것을 특징으로 하는 라지 스케일 온톨로지 매칭 장치.

청구항 3

제 2항에 있어서,

상기 직렬화된 온톨로지 서브셋을 저장하며, 상기 수신된 후보 온톨로지와 동일한 후보 온톨로지가 수신되면, 기 저장된 직렬화된 온톨로지 서브셋을 상기 전처리부로 제공하는 온톨로지 저장부를 더 포함하는 것을 특징으로 하는 라지 스케일 온톨로지 매칭 장치.

청구항 4

제 3항에 있어서,

상기 전처리부는,

상기 온톨로지 저장부로부터 수신된 직렬화된 온톨로지 서브셋을 역직렬화(De-serealization)를 통해 재구성하여 온톨로지 서브셋을 생성하는 것을 특징으로 하는 라지 스케일 온톨로지 매칭 장치.

청구항 5

제 1항에 있어서,

상기 매칭 스레드는 상기 참여 노드 각각에 일대일로 대응하는 하나 이상의 매칭 요청(Matching Request), 상기 매칭 요청에 일대일로 대응하는 참여 노드에 구비된 개별 코어에 일대일로 대응하는 하나 이상의 매칭 작업(Matching Job) 및 상기 개별 코어에서 매칭 연산을 수행하는 하나 이상의 매칭 태스크(Matching Task)를 포함하는 것을 특징으로 하는 라지 스케일 온톨로지 매칭 장치.

청구항 6

제 5항에 있어서,

상기 매칭 요청, 매칭 작업 및 매칭 태스크의 관계는

$$MR \leftarrow \sum_{i=1}^{Nodes} MR_i$$

$$MR_i \leftarrow \sum_{j=1}^{Cores} MJ_j$$

$$MJ_j \leftarrow \sum_{k=1}^{Copncepts} MT_k$$

$MT_k \leftarrow m \times n \mid m \in O_s \; n \in O_t$ 에 의해 산출되며,

상기 MR_i는 상기 참여 노드 각각에 할당된 매칭 요구를 나타내고, MR은 상기 참여 노드에 할당된 매칭 요구 MR_i의 집합 매칭 요구를 나타내며, MJ_j는 상기 참여 노드에 구비된 개별 코어에 할당되는 매칭작업을 나타내고, MT_k는 개별 코어에 할당되어 매칭 연산을 수행하는 매칭태스크를 나타내는 것을 특징으로 하는 라지 스케일 온톨로지 매칭 장치.

청구항 7

제 5항에 있어서,

상기 매칭 작업은 다른 매칭 작업 및 다른 참여 노드에서 동작하는 매칭 요구에 독립적인 것을 특징으로 하는 라지 스케일 온톨로지 매칭 장치.

청구항 8

제 1항에 있어서,

상기 분산 처리부는,

상기 후보 온톨로지가 수신되면, 매칭에 활용할 수 있는 참여 노드의 개수 및 상기 참여 노드에 구비된 개별 코어의 개수를 확인하고, 상기 참여 노드 및 상기 개별 코어의 개수를 분산 알고리즘에 적용하여 분산 개수를 설정하며, 상기 설정된 분산 개수를 고려하여 상기 온톨로지 서브셋을 분할하고 매칭 알고리즘을 적용하여 매칭 요청, 매칭 작업 및 매칭 태스크를 포함하는 매칭 스레드를 생성하는 것을 특징으로 하는 라지 스케일 온톨로지 매칭 장치.

청구항 9

제 1항에 있어서,

둘 이상의 참여 노드를 포함하는 분산 환경에서 상기 집산부의 매칭 결과를 수집 및 합산(Aggregating)하는 과정은

$$O_b^{Job} \leftarrow \bigcup_{i=1}^{MT} O_b^i,$$

$$O_b^{Node} \leftarrow \sum_{i=0}^{MJ} O_b^{job=i}$$

및

$O_b \leftarrow \sum_{i=1}^{TotalNode} O_b^{Node=i}$ 에 의해 산출되며,

O_b^i는 매칭 태스크의 매칭 결과(매칭 태스크의 브릿지 온톨로지)이고, O_b^{Job}은 모든 매칭 작업에서의 모든 단일 매칭 태스크의 결과를 모두 수집한 매칭 작업의 중앙 브릿지 온톨로지(Intermediate Bridge Ontology)이며, O_b^{Node}는 단일 참여 노드에서 실행되는 모든 매칭 작업에서 모든 중앙 브릿지 온톨로지 O_b^{Job}의 합이고, O_b^{Job}는 모든 노드에서 생성된 노드로부터의 중앙 브릿지 온톨로지이며, O_b는 최종 생성되는 브릿지 온톨로지로서 출력 파일인 온톨로지 매핑 결과로 전환되는 것을 특징으로 하는 라지 스케일 온톨로지 매칭 장치.

청구항 10

제 9항에 있어서,

상기 연산된 매칭 결과는 브릿지 온톨로지라고 불리는 하나의 온톨로지 오브젝트와 같이 합산되며, 상기 집산부는 분산 환경에 참여한 모든 참여 노드에 의해 연산된 매칭 결과인 브릿지 온톨로지를 합산하여 물리적 매핑 파일(Physical Mapping File)인 온톨로지 매핑 결과로 전환하여 출력하는 것을 특징으로 하는 라지 스케일 온톨로지 매칭 장치.

청구항 11

수신된 후보 온톨로지를 하나 이상의 온톨로지 서브셋으로 분류하는 단계;

상기 생성된 온톨로지 서브셋을 분산 알고리즘을 적용하여 분할하고, 상기 분할된 온톨로지 서브셋에 매칭 알고리즘을 적용하여 매칭 스레드를 생성하는 단계;

상기 생성된 매칭 스레드를 참여 노드의 개별 코어에 전달하는 단계;

상기 개별 코어에서 상기 매칭 스레드에 기초하여 매칭 연산을 수행하여 생성된 매칭 결과를 수집 및 합산하여 온톨로지 매핑을 생성하는 단계;

를 포함하는 것을 특징으로 하는 라지 스케일 온톨로지 매칭 방법.

청구항 12

제 11항에 있어서,

수신된 후보 온톨로지를 하나 이상의 온톨로지 서브셋으로 분류하는 단계는,

상기 생성된 온톨로지 서브셋을 바이너리(Binary) 형태로 직렬화(Serialization)하여 직렬화된 온톨로지 서브셋을 생성하는 것을 특징으로 하는 라지 스케일 온톨로지 매칭 방법.

청구항 13

제 12항에 있어서,

상기 직렬화된 온톨로지 서브셋을 저장하며, 상기 수신된 후보 온톨로지와 동일한 후보 온톨로지가 수신되면, 기 저장된 직렬화된 온톨로지 서브셋을 역직렬화(De-serealization)를 통해 재구성하여 온톨로지 서브셋을 생성하는 것을 특징으로 하는 라지 스케일 온톨로지 매칭 방법.

청구항 14

제 11항에 있어서,

상기 매칭 스레드는 상기 참여 노드 각각에 일대일로 대응하는 하나 이상의 매칭 요청(Matching Request), 상기

매칭 요청에 일대일로 대응하는 참여 노드에 구비된 개별 코어에 일대일로 대응하는 하나 이상의 매칭 작업(Matching Job) 및 상기 개별 코어에서 매칭 연산을 수행하는 하나 이상의 매칭 태스크(Matching Task)를 포함하는 것을 특징으로 하는 라지 스케일 온톨로지 매칭 방법.

청구항 15

제 14항에 있어서,

상기 매칭 요청, 매칭 작업 및 매칭 태스크의 관계는

$$MR \leftarrow \sum_{i=1}^{Nodes} MR_i$$

$$MR_i \leftarrow \sum_{j=1}^{Cores} MJ_j$$

$$MJ_j \leftarrow \sum_{k=1}^{Copncepts} MT_k$$

$MT_k \leftarrow m \times n \mid m \in O_s \; n \in O_t$ 에 의해 산출되며,

상기 MR_i는 상기 참여 노드 각각에 할당된 매칭 요구를 나타내고, MR은 상기 참여 노드에 할당된 매칭 요구 MR_i의 집합 매칭 요구를 나타내며, MJ_j는 상기 참여 노드에 구비된 개별 코어에 할당되는 매칭작업을 나타내고, MT_k는 개별 코어에 할당되어 매칭 연산을 수행하는 매칭태스크를 나타내는 것을 특징으로 하는 라지 스케일 온톨로지 매칭 방법.

청구항 16

제 14항에 있어서,

상기 매칭 작업은 다른 매칭 작업 및 다른 참여 노드에서 동작하는 매칭 요구에 독립적인 것을 특징으로 하는 라지 스케일 온톨로지 매칭 방법.

청구항 17

제 11항에 있어서,

상기 매칭 스레드를 생성하는 단계는,

분산 환경의 참여 노드의 개수 및 상기 참여 노드에 구비된 개별 코어의 개수를 확인하는 단계;

상기 참여 노드 및 상기 개별 코어의 개수를 분산 알고리즘에 적용하여 분산 개수를 설정하는 단계; 및

상기 설정된 분산 개수를 고려하여 상기 온톨로지 서브셋을 분할하고 매칭 알고리즘을 적용하여 매칭 요청, 매칭 작업 및 매칭 태스크를 포함하는 매칭 스레드를 생성하는 단계;

를 포함하는 것을 특징으로 하는 라지 스케일 온톨로지 매칭 방법.

청구항 18

제 11항에 있어서,

둘 이상의 참여 노드를 포함하는 분산 환경에서 상기 집산부의 매칭 결과를 수집 및 합산하는 과정은

$$O_b^{Job} \leftarrow \bigcup_{i=1}^{MT} O_b^i$$

,

$$O_b^{Node} \leftarrow \sum_{i=0}^{MJ} O_b^{job=i}$$ 및

$$O_b \leftarrow \sum_{i=1}^{TotalNode} O_b^{Node=i}$$ 에 의해 산출되며,

O_b^i는 매칭 태스크의 매칭 결과(매칭 태스크의 브릿지 온톨로지)이고, O_b^{Job}은 모든 매칭 작업에서의 모든 단일 매칭 태스크의 결과를 모두 수집한 매칭 작업의 중앙 브릿지 온톨로지(Intermediate Bridge Ontology)이며, O_b^{Node}는 단일 참여 노드에서 실행되는 모든 매칭 작업에서 모든 중앙 브릿지 온톨로지 O_b^{Job}의 합이고, O_b^{Job}는 모든 노드에서 생성된 노드로부터의 중앙 브릿지 온톨로지이며, O_b는 최종 생성되는 브릿지 온톨로지로서 출력 파일인 온톨로지 매핑 결과로 전환되는 것을 특징으로 하는 라지 스케일 온톨로지 매칭 방법.

청구항 19

제 18항에 있어서,

상기 상기 개별 코어에서 상기 매칭 스레드에 기초하여 매칭 연산을 수행하여 생성된 매칭 결과를 수집 및 합산하여 온톨로지 매핑을 생성하는 단계는 모든 참여 노드에 의해 연산된 매칭 결과인 브릿지 온톨로지를 합산하여 물리적 매핑 파일(Physical Mapping File)인 온톨로지 매핑 결과로 전환하여 출력하는 것을 특징으로 하는 라지 스케일 온톨로지 매칭 방법.

발명의 설명

기 술 분 야

[0001] 본 발명은 라지 스케일 생물의학 분야의 온톨로지 매칭에 관한 기술로서, 보다 상세하게는 온톨로지 매칭의 병렬 처리 및 분산 처리에 관한 기술이다.

배 경 기 술

[0002] 최근 들어, 생물의학(Biomedical) 분야의 정보 처리 및 정보 관리에 있어서 시맨틱 웹(Semantic Web) 기술이 적용되면서, 생물의학 시스템에서 많은 이익을 가져오고 있다. 특히, 생물의학 시스템에서 온톨로지(Ontology) 기술은 생물의학 정보의 표준화(Standardization), 지식 공유(Knowledge Sharing) 및 재사용능력(Reusability) 활용을 위해 많이 이용된다. 결과적으로, 유전자 온톨로지(Gene Ontology, GO), 국립 암 연구소 유의어사전(National Cancer Institute Thesaurus, NCI), 해부학 토대 모델(Foundation Model of Anatomy, FMA) 및 SNOMED CT(Systemized Nomenclature of Medicine Clinical Terms)와 같은 형태로 생물의학 분야에 온톨로지는 적용되고 있다. 이와 같이, 생물의학 분야에 보다 효과적으로 온톨로지를 적용하기 위해 여러 연구가 진행되고 있으며, 이미 개발된 위와 같은 온톨로지에 지속성을 제공하기 위한 연구 또한 진행되고 있다.

[0003] 생물의학 분야의 온톨로지는 거대한 규모(Large Scale)을 가지기 때문에 매우 복잡하며, 이는 통합(Integration) 및 정보처리 상호 운용(Interoperability)에 장애 요인으로 작용한다. OBO(Open Biomedical Ontologies) 컨소시엄은 온톨로지 진화를 위한 도입 전략(Introducing Strategy)을 통해 이러한 장애 요인을 해결하는데 노력하고 있다. 이러한 생물의학 온톨로지는 오버래핑(Overlapping) 정보를 포함한다. 오버래핑 정보는 생물의학 시스템의 통합 및 정보 처리 상호 운용을 위해 필요한 정보이다. 온톨로지에 있어서, 서로 다른 후보 온톨로지(Candidate Ontology) 사이의 관련성은 매핑(Mapping) 또는 얼라인먼트(Alignment)라고 불리운다. 후보 온톨로지는 서로 관의 관계를 정립하는 매핑 과정을 위한 온톨로지이며, 서로 다른 온톨로자 사이의 관계

를 정립하는 매핑 발견의 과정은 온톨로지 매칭(Ontology Matching)이라는 용어로 명명된다.

[0004] 거대한 규모의 생물의학 온톨로지 상에서 온톨로지 매핑을 발견하기 위한 온톨로지 매칭 과정은 2차의 연산 복잡도와 함께 연산 과도한 연산 작업을 필요로 한다. 2010년 Data Integration in the Life Sciences에 개재된 "On Matching Large Life Science Ontologies in Parallel"을 살펴보면, 온톨로지 매칭은 두 후보 온톨로지의 카테시안 곱(Cartesian Product)으로 계산된다. 이 작업은 리소스 기반의 매칭 알고리즘을 요구한다. 이러한 매핑 과정에서 2차의 연산 복잡도에 의한 과도한 연산 작업은 지연(Delay)를 발생시킬 수 있으며, 이러한 매핑 결과의 지연은 생물의학 시스템을 위한 온톨로지 매핑을 처리 요구 시간 이내에 처리하는 것을 효과적이지 못하게 한다.

선행기술문헌

비특허문헌

[0005] (비특허문헌 0001) Anika Groβ, Michael Hartung, Toralf Kirsten, Erhard Rahm. "On Matching Large Life Science Ontologies in Parallel". Data Integration in the Life Sciences, Lecture Notes in Computer Science Volume 6254, 2010, pp 35-49.

발명의 내용

해결하려는 과제

[0006] 본 발명이 해결하고자 하는 과제는 라지 스케일 생물의학 분야의 온톨로지에 있어서, 온톨로지 매핑을 위한 온톨로지 매칭 과정에서 다중 코어 상에서 성능 향상을 위한 병렬 매칭을 수행하는 온톨로지 매칭 장치 및 방법을 제공하는 것이다.

과제의 해결 수단

[0007] 본 발명에 따른 라지 스케일 온톨로지 매칭 장치는 수신된 후보 온톨로지를 하나 이상의 온톨로지 서브셋으로 분류하여 온톨로지 서브셋을 생성하는 전처리부, 생성된 온톨로지 서브셋을 분산 알고리즘을 적용하여 분할하고, 분할된 온톨로지 서브셋에 매칭 알고리즘을 적용하여 매칭 스레드를 생성하며, 생성된 매칭 스레드를 참여 노드의 개별 코어에 전달하는 분산 처리부 및 개별 코어에서 매칭 스레드에 기초하여 매칭 연산을 수행하여 생성된 매칭 결과를 수집 및 합산하여 온톨로지 매핑을 생성하는 집산부를 포함한다. 그리고, 직렬화된 온톨로지 서브셋을 저장하며, 수신된 후보 온톨로지와 동일한 후보 온톨로지가 수신되면, 기 저장된 직렬화된 온톨로지 서브셋을 전처리부로 제공하는 온톨로지 저장부를 더 포함할 수 있다.

[0008] 전처리부는 생성된 온톨로지 서브셋을 바이너리(Binary) 형태로 직렬화(Serialization)하여 직렬화된 온톨로지 서브셋을 생성할 수 있으며, 온톨로지 저장부로부터 수신된 직렬화된 온톨로지 서브셋을 역직렬화(De-serealization)를 통해 재구성하여 온톨로지 서브셋을 생성할 수 있다. 분산 처리부에서 생성되는 매칭 스레드는 하나 이상의 매칭 요청(Matching Request), 하나 이상의 매칭 작업(Matching Job) 및 하나 이상의 매칭 태스크(Matching Task)를 포함할 수 있다. 매칭 요청은 참여 노드 각각에 일대일로 대응하며, 매칭 작업은 매칭 요청에 일대일로 대응하는 참여 노드에 구비된 개별 코어에 일대일로 대응한다. 하나 이상의 매칭 요청의 최대 개수는 참여 노드의 수를 넘지 않으며, 하나 이상의 매칭 작업의 최대 개수는 참여 노드 전체에 구비된 개별 코어의 수를 넘지 않는다. 그리고, 매칭 작업은 다른 매칭 작업 및 다른 참여 노드에서 동작하는 매칭 요구에 독립적으로 동작한다.

[0009] 분산 처리부는 후보 온톨로지가 수신되면, 참여 노드의 개수 및 참여 노드에 구비된 개별 코어의 개수를 확인하고, 참여 노드 및 개별 코어의 개수를 분산 알고리즘에 적용하여 분산 개수를 설정하며, 설정된 분산 개수를 고려하여 온톨로지 서브셋을 분할하고 매칭 알고리즘을 적용하여 매칭 요청, 매칭 작업 및 매칭 태스크를 포함하는 매칭 스레드를 생성한다.

[0010] 본 발명에 따른 라지 스케일 온톨로지 매칭 방법은 먼저, 수신된 후보 온톨로지를 하나 이상의 온톨로지 서브셋으로 분류한다. 그리고, 생성된 온톨로지 서브셋을 분산 알고리즘을 적용하여 분할하고, 분할된 온톨로지 서브셋에 매칭 알고리즘을 적용하여 매칭 스레드를 생성한다. 매칭 스레드가 생성되며느 생성된 매칭 스레드를 참여 노드의 개별 코어에 전달한다. 다음으로, 개별 코어에서 매칭 스레드에 기초하여 매칭 연산을 수행하여 생성된

매칭 결과를 수집 및 합산하여 온톨로지 매핑을 생성한다.

[0011] 본 발명에 따른 라지 스케일 온톨로지 매칭 방법은 참여 노드의 개수 및 참여 노드에 구비된 개별 코어의 개수를 확인하고, 참여 노드 및 개별 코어의 개수를 분산 알고리즘에 적용하여 분산 개수를 설정한다. 그리고, 설정된 분산 개수를 고려하여 온톨로지 서브셋을 분할하고 매칭 알고리즘을 적용하여 매칭 요청, 매칭 작업 및 매칭 태스크를 포함하는 매칭 스레드를 생성한다.

발명의 효과

[0012] 본 발명에 따른 라지 스케일 온톨로지 매칭 장치 및 매칭 방법을 통해 온톨로지 매칭 연산을 참여 노드의 개별 코어에 분산시켜 병렬 처리하여 매칭 연산에 필요한 연산 자원 및 연산 시간을 효과적으로 줄일 수 있다.

도면의 간단한 설명

[0013] 도 1은 본 발명에 따른 생물의학 온톨로지의 일 실시예를 나타내는 구성도이다.

도 2는 본 발명에 따른 라지 스케일 온톨로지 매칭 장치의 일 실시예를 나타내는 구성도이다.

도 3은 본 발명의 일 실시예에 따른 라지 스케일 온톨로지 매칭 장치의 전처리부를 나타내는 상세도이다.

도 4는 본 발명의 일 실시예에 따른 라지 스케일 온톨로지 매칭 장치의 분산 처리부를 나타내는 상세도이다.

도 5는 본 발명의 일 실시예에 따른 라지 스케일 온톨로지 매칭 장치에서 개별 코어로 전달되는 매칭 스레드의 일 실시예를 나타내는 도면이다.

도 6은 본 발명의 일 실시예에 따른 라지 스케일 온톨로지 매칭 장치의 데이터 흐름을 나타내는 도면이다.

도 7은 본 발명의 일 실시예에 따른 라지 스케일 온톨로지 매칭 방법을 나타내는 흐름도이다.

도 8은 본 발명의 일 실시예에 따른 라지 스케일 온톨로지 매칭 방법의 전처리 생략 방법을 나타내는 흐름도이다.

발명을 실시하기 위한 구체적인 내용

[0014] 이하, 본 발명의 실시예를 첨부된 도면들을 참조하여 상세하게 설명한다. 본 명세서에서 사용되는 용어 및 단어들은 실시예에서의 기능을 고려하여 선택된 용어들로서, 그 용어의 의미는 발명의 의도 또는 관례 등에 따라 달라질 수 있다. 따라서 후술하는 실시예에서 사용된 용어는, 본 명세서에 구체적으로 정의된 경우에는 그 정의에 따르며, 구체적인 정의가 없는 경우는 당업자들이 일반적으로 인식하는 의미로 해석되어야 할 것이다.

[0015] 도 1은 본 발명에 따른 생물의학 온톨로지의 일 실시예를 나타내는 구성도이다.

[0016] 도 1을 참조하면, 본 발명에서 생물의학 온톨로지는 상호간의 관계가 정립된 하나 이상의 구성요소(10)로 형성된다. 하나 이상의 구성요소(10)는 온톨로지 매칭 과정을 통해 개별 구성요소(10)와 대응하는 구성요소(10)의 관계를 정립하게 되고, 이와 같은 개별 구성요소(10) 사이의 매칭 결과가 모여 온톨로지 매핑을 형성한다. 라지 스케일 온톨로지 매칭 장치는 대규모(large scale) 온톨로지(Ontology) 매칭(Matching) 과정에서 연산을 효율적으로 수행하기 위해 참여한 노드(Node)의 하드웨어를 활용한다. 분산된 환경에 참여한 노드는 가능한 분산 환경에 의존하는 하나 이상의 노드를 포함할 수 있다. 그리고 참여한 노드는 단일 코어(Single-core) 또는 멀티코어(Multi-core) 하드웨어를 모두 포함할 수 있다. 라지 스케일 온톨로지 매칭 장치는 하나 이상의 노드를 포함하는 분산 환경에서 각각의 노드가 보유한 싱글코어 프로세서 또는 멀티코어 프로세서에 온톨로지 서브셋(Ontology Subset)을 할당하고 데이터 병렬화 기술을 통해 처리한다.

[0017] 생물의학 온톨로지는 일반적인 온톨로지와 동일하게 시매틱-웹 온톨로지(Semantic-web Ontology) 그룹에 속한다. 하지만, 생물의학 온톨로지와 일반적인 온톨로지는 규모(size)와 진화(Evolution)에 있어서 차이가 존재한다. 일반 온톨로지는 생물의학 온톨로지에 비해 규모가 더 작으며, 진화 또한 더 느리다. 반면에, 생물의학 온톨로지는 생물의학 도메인 및 생물의학 데이터에서의 빠른 진화 때문에 더욱 빠르게 진화하고 이에 따라 규모 또한 더욱 크다.

[0018] 분산 환경에서 온톨로지 매칭은 소스 온톨로지(Source Ontology)와 타겟 온톨로지(Target Ontology) 사이의 개념에 기초하여 대응(Matching) 관계를 설정하거나 연결하는 과정으로, 이러한 과정을 통해 온톨로지의 분류 체

계를 확장해 줄 수 있다. 소스 온톨로지는 새로운 관계를 정립하기 위한 온톨로지이고, 타겟 온톨로지는 온톨로지 매칭 과정을 통해 소스 온톨로지와 매칭되는 대상이 되는 온톨로지이다.

[0019] 소스 온톨로지 및 타겟 온톨로지 사이의 매칭 작업의 총 개수는 수학식 1과 같이 가용한 연산 리소스(코어) 사이에 균등하게 분산된다.

수학식 1

[0020] $$MT_{total} \leftarrow (m \times n) \forall m \in O_S \; n \in O_t$$

[0021] $$MT_{core} \leftarrow \frac{m \times n}{numberofcores}$$

[0022] 수학식 1에서 MTtotal은 전체 매칭 태스크를 나타내고, Os는 소스 온톨로지를 나타내며, Ot는 타겟 온톨로지를 나타내고, MTcore는 개별 코어의 매칭 태스크를 나타내며, m은 소스 온톨로지의 컨셉(Concept)이고, n은 타겟 온톨로지의 컨셉이다. 대규모 생물의학 온톨로지 매칭을 위한 요구는 생물의학 분야의 전문가, 연구원, 생물의학/생물정보학 시스템 및 심지어 클라우드 플랫폼 상에서 실행되는 서드파티(Third Party) 헬스케어 정보 서비스(Healthcare Information Service)를 포함하는 몇몇의 소스로부터 생성될 수 있다.

[0023] 도 2는 본 발명에 따른 라지 스케일 온톨로지 매칭 장치의 일 실시예를 나타내는 구성도이다.

[0024] 도 2를 참조하면, 본 발명에 따른 라지 스케일 온톨로지 매칭 장치(100)는 전처리부(110), 온톨로지 저장부(120), 분산 처리부(130) 및 집산부(140)를 포함한다.

[0025] 전처리부(110)는 소스 온톨로지 및 타겟 온톨로지를 포함하는 후보 온톨로지를 입력받는다. 소스 온톨로지 및 타겟 온톨로지를 포함하는 후보 온톨로지는 서로 1:1로 매칭할 수 있으며, 하나의 소스 온톨로지가 둘 이상의 타겟 온톨로지에 매칭하거나, 둘 이상의 소스 온톨로지가 하나의 타겟 온톨로지에 매칭할 수 있다.

[0026] 전처리부(110)는 수신된 소스 온톨로지 및 타겟 온톨로지를 포함하는 후보 온톨로지를 하나 이상의 서브셋(Subset)으로 분류한다. 온톨로지는 일반적으로 5개의 서브셋으로 구성될 수 있다. 온톨로지를 구성하는 서브셋은 크게 이름(Name), 레이블(Lable), 관계(Relationship), 공리(Axiom) 및 특성(Property)으로 구분될 수 있다. 하지만, 온톨로지를 상술한 다섯 가지의 서브셋으로만 구분할 수 있는 것은 아니며, 전처리부(110)는 수신된 후보 온토로지의 종류나 특성에 따라 숫자에 관계없이 하나 이상의 서브셋으로 분류할 수 있다. 그리고, 전처리부(110)는 분류된 온톨로지 서브셋을 온톨로지 저장부(120) 및 분산 처리부(130)로 전달한다. 전처리부(110)는 분류된 온톨로지 서브셋을 분산 처리부(130)로 직접 전달할 수 있으며, 또한, 전처리부(110)는 온톨로지 저장부(120)에 저장되어 있던 온톨로지 서브셋을 수신하여 분산 처리부(130)로 전달할 수 있다. 특히, 전처리부(110)는 이전에 분류하여 온톨로지 저장부(120)에 저장한 온톨로지 서브셋과 동일한 후보 온톨로지가 수신되는 경우, 서브셋으로 분류하는 전처리 과정을 반복하지 않고, 온톨로지 저장부(120)에 직렬화되어 저장된 온톨로지 서브셋을 전처리 과정 없이 분산 처리부(130)로 전달함으로써, 전처리 과정에서 소모되는 연산 리소스를 줄일 수 있다. 또한, 전처리부(110)는 온톨로지 서브셋을 직렬화(Serialization)하여 온톨로지 저장부(120)로 전달할 수 있으며, 온톨로지 저장부(120)로부터 수신된 직렬화된 온톨로지 서브셋을 다시 역직렬화(De-derialization)하여 분산 처리부(130)로 전달할 수 있다.

[0027] 온톨로지 저장부(120)는 전처리부(110)로부터 수신된 온톨로지 서브셋을 직렬화(Serialization)하여 저장한다. 온톨로지 저장부(120)는 수신된 온톨로지 서브셋을 바이너리 형태로 직렬화하여 저장 공간을 효율적으로 사용할 수 있으며, 데이터 처리 효율을 높일 수 있다, 그리고, 전처리부(110)에서 이전에 온톨로지 저장부(120)는 바이너리 형식으로 직렬화하여 저장한 온톨로지 서브셋을 전처리부(110)의 요구에 따라 전달함으로써, 전처리부(110)에서 불필요한 전처리 과정의 반복(Re-processing)을 줄이거나 피할 수 있다.

[0028] 분산 처리부(130)는 전처리부(110)로부터 수신된 온톨로지 서브셋을 데이터 병렬화(Data Parallel)하기 위해 둘 이상의 매칭 스레드(Matching Thread)로 분할한다. 데이터 병렬화는 둘 이상의 분산 온톨로지로 분할된 후보 온톨로지의 조각 각각에 대해 매칭 작업을 수행하기 위해 각각의 프로세싱 코어(Processing Core)를 요구한다. 병렬 처리 프로세싱을 가능하게 하기 위해, 분산 처리부(130)는 온톨로지 서브셋을 분산 알고리즘을 고려하여 분

할한다. 먼저, 분산 처리부(130)는 먼저 개별 참여 노드의 멀티 코어 프로세서(Multi-core Processor)의 코어의 개수를 파악한다. 그리고, 분산 처리부(130) 파악된 참여 노드 및 개별 참여 노드의 코어 수를 분산 알고리즘에 적용하여 수신된 온톨로지 서브셋을 소정의 분산 온톨로지로 분할한다.

[0029] 그리고, 분산 처리부(130)는 분산 알고리즘에 기초하여 분할된 하나 이상의 온톨로지 서브셋에 매칭 라이브러리(Matching Library)를 적용하여 매칭 요구(Matching Request, 이하 MR이라 칭함), 매칭 작업(Matching Job, 이하 MJ라 칭함) 및 매칭 태스크(Matching Task, 이하 MT라 칭함)로 분할한다. 수학식 2는 분산 처리부(130)에 적용된 분산 알고리즘의 매칭 요구(Matching Request, 이하 MR이라 칭함), 매칭 작업(Matching Job, 이하 MJ라 칭함) 및 매칭 태스크(Matching Task, 이하 MT라 칭함)의 관계를 나타낸다.

수학식 2

[0030] $$MR \leftarrow \sum_{i=1}^{Nodes} MR_i$$

[0031] $$MR_i \leftarrow \sum_{j=1}^{Cores} MJ_j$$

[0032] $$MJ_j \leftarrow \sum_{k=1}^{Copncepts} MT_k$$

[0033] $$MT_k \leftarrow m \times n \mid m \in O_s \; n \in O_t$$

[0034] 수학식 2에서 MR_i는 각 노드에 수신된 매칭 요구를 나타내고, i는 분산 환경에서 병렬 매칭을 위한 참여 노드의 수를 나타내고, MR은 매칭을 위해 라지 스케일 온톨로지 매칭 장치(100)에 의해 수신된 매칭 요구를 나타내며, MJ_j는 각 참여노드를 구성하는 개별 코어(20)에 할당된 매칭작업을 나타내고, j는 참여 노드에 구비된 개별 코어(20)의 수를 나타내며, MT_k는 각 참여 노드의 하나의 개별 코어(20)에 할당되어 매칭 연산을 수행하는 매칭태스크를 나타내고, k는 하나의 개별 코어(20)에 할당된 매칭 작업에 포함된 매칭 태스크의 개수를 나타낸다. 그리고, m은 소스 온톨로지의 컨셉(Concept)이고, n은 타겟 온톨로지의 컨셉이다. 단일 매칭 태스크 MT_k는 소스 온톨로지 및 타겟 온톨로지 각각의 컨셉의 카티전 곱(Cartesian Product)이다.

[0035] 매칭 요구는 분산 환경의 참여 노드(또는 참여 노드의 CPU) 각각에 대응하여 전달되는 매칭 스레드(Mathing Thread)이다. 분산 처리부(130)는 먼저 참여 노드(또는 참여 노드의 CPU)의 수를 고려하여 온톨로지 서브셋을 각각의 참여 노드(또는 참여 노드의 CPU)에 대응하는 하나 이상의 매칭 요구로 분할한다. 매칭 요구의 최대 개수는 참여 노드(또는 참여 노드의 CPU)의 개수가 될 수 있다. 즉, 매칭 요구는 온톨로지 병렬 처리를 위해 개별 노드에 할당되는 병렬 처리 작업이 될 수 있다. 그리고, 분산 처리부(130)는 분할된 매칭 요구를 다시 해당 매칭 요구가 할당되는 참여 노드의 코어의 수에 따라 매칭 작업으로 분할하여 해당 매칭 요구에 포함시킨다. 즉, 매칭 작업은 개별 코어(20)에 할당되는 병렬 처리 작업으로서, 해당 매칭 요구가 할당된 참여 노드의 코어의 수가 4개의 코어를 가진다면, 해당 매칭 요구는 4개의 매칭 작업을 포함할 수 있다. 다음으로, 분산 처리부(130)는 개별 코어에 할당되는 매칭 작업을 실제 매칭 프로세싱을 수행하는 단위인 매칭 태스크로 분할하여 매칭 작업에 포함시킨다.

[0036] 매칭 요구, 매칭 작업 및 매칭 태스크는 전체 매칭 프로세스를 위한 세 개의 추상화 계층(Layer of Abstraction)이다. 병렬 처리 프로세싱을 위한 분산 알고리즘에서 서로 다른 레벨에서 분류를 제공해야 하기 때문에, 분산 처리부(130)의 분산 알고리즘은 실행중인 모든 작업의 트랙(Track)을 보존해야 한다.

[0037] 매칭 태스크는 매칭 프로세스의 유닛으로 가장 작은 전체 매칭 프로세스로 분류될 수 있다. 예를 들어, 해당 참여 노드의 코어에 할당된 매칭 작업에서 소스 온톨로지가 {A,B,C,D}이고 타겟 온톨로지가 {a,b,c,d}라면, 소스

온톨로지와 타겟 온톨로지를 비교하여 매칭 프로세스를 수행할 때, A=a는 첫 번째 MT, B=b는 두 번째 MT, C=c는 세 번째 MT, D=d는 네 번째 MT일 수 있으며, A=d를 비교하여 매칭할 수 있다면, 이 또한 다른 하나의 MT가 될 수 있다. 즉, 매칭 태스크는 참여 노드의 개별 코어 내부에서 개별 매칭 프로세싱을 처리하는 가장 작은 단위로서, 코어 내부에서 온톨로지를 구성하는 개별 용어(Term)의 매칭 프로세싱을 처리한다.

[0038] 매칭 작업은 참여 노드의 개별 코어(20)에 할당되어 매칭 프로세스를 수행하는 단위로서, 하나 이상의 매칭 태스크의 모음이 될 수 있다. 그리고, 매칭 요구는 하나의 참여 노드(또는 참여 노드의 CPU)에서 실행되는 모든 매칭 작업의 모음이다.

[0039] 예를 들어, 4개의 참여 노드가 매칭 프로세스에 존재하고, 4개의 참여 노드가 모두 4개의 코어를 가지며, 수신된 온톨로지 서브셋이 총 2000개의 매칭을 필요로 한다면, 분산 처리부(130)는 총 2000개의 매칭을 참여 노드의 수에 따라 4등분하여 총 4개의 매칭 요구로 분할한다. 분할된 4개의 매칭 요구 각각은 500개의 매칭을 포함한다. 그리고, 분산 처리부(130)는 4개의 매칭 요구 각각을 코어의 수를 고려하여 4개의 매칭 작업으로 분류한다. 그러면, 각각의 매칭 작업은 500개의 매칭을 4로 나눈 125개의 매칭을 포함한다. 결국, 하나의 매칭 작업은 125개의 매칭을 포함하며, 개별 매칭 작업에 포함된 매칭 태스크는 125개가 된다. 즉, 4개의 매칭 요청 각각은 4개의 매칭 작업을 포함하며, 4개의 매칭 작업 각각은 125개의 매칭 태스크를 포함하여 결국, 총 2000개의 매칭을 포함하는 하나 이상의 온톨로지 서브셋은 4개의 매칭 요청으로 분할되고, 각각의 매칭 요청은 4개의 매칭 작업을 포함하며, 각각의 매칭 작업은 125개의 매칭 태스크를 포함한다.

[0040] 매칭 요청, 매칭 작업 및 매칭 태스크의 개수는 참여 노드 및 개별 참여 노드의 수뿐만 아니라, 가장 최적화된 병렬 처리를 수행할 수 있는 시나리오를 고려하여 설정될 수 있다. 그리고, 각각의 매칭 작업은 스스로 다른 매칭 작업 및 원격에서 동작하는 다른 매칭 요구에 독립적이다. 또한, 매칭 요구를 매칭 작업으로 세분화하는 과정에서 하나 이상의 매칭 작업에 포함된 매칭 태스크의 수를 하나 이상의 매칭 작업 간에 동일하게 유지한다면, 대부분의 개별 코어가 유사한 시간에 매칭 연산을 종료하여 매칭 결과를 생성할 수 있어 가용한 매칭 코어가 유휴상태(Idel)을 유지하는 것을 막을 수 있다. 매칭 요청, 매칭 작업 및 매칭 태스크는 후술하는 도 5에서 다시 설명하도록 한다.

[0041] 분산 처리부(130)는 상술한 과정을 통해 하나 이상의 온톨로지 서브셋을 소정의 개수를 가지는 매칭 요청을 참여 노드로 전달한다. 하나의 매칭 요청은 대응하는 하나의 참여 노드로 전달된다. 하나의 참여 노드로 전달된 하나의 매칭 요청은 해당 참여 노드에 포함된 하나 이상의 코어 각각에 대응하는 매칭 작업을 포함한다. 하나의 매칭 요청에 포함된 하나 이상의 매칭 작업은 대응하는 코어에 일 대 일로 할당되다. 그리고, 하나의 코어에 할당된 하나의 매칭 작업은 포함된 매칭 태스크를 통해 해당 코어에서 매칭 프로세싱을 수행한다. 이를 통해, 본 발명에 따른 라지 스케일 온톨로지 매칭 장치(100)는 참여 노드의 코어 각각에서 매칭 프로세싱을 병렬처리할 수 있다.

[0042] 집산부(140)는 분산 처리부(130)로부터 매칭 스레드를 전달받은 참여 노드의 개별 코어(20)로부터 매칭 스레드를 통해 매칭 연산을 수행한 매칭 결과를 수신한다. 분산 처리부(130)는 매칭 요청, 매칭 작업 및 매칭 태스크를 포함하는 매칭 스레드를 참여 노드의 각 코어에 전달한다. 분산 처리부(130)로부터 매칭 요청, 매칭 작업 및 매칭 태스크를 포함하는 매칭 스레드를 전달받은 참여 노드의 개별 코어(20)는 수신된 매칭 스레드에 기초하여 매칭 연산을 수행한다. 매칭 스레드에 포함된 매칭 요청을 수신한 참여 노드는 매칭 요청에 포함된 매칭 작업에 따라 구비된 개별 코어(20)에 매칭 작업을 할당하고, 개별 코어(20)는 할당된 매칭 작업에 포함된 매칭 태스크를 수행함으로써, 수신된 후보 온톨로지의 매칭 프로세싱을 병렬로 처리하게 된다. 집산부(140)는 개별 코어에서 수행된 온톨로지 매칭 연산에 의해 생성된 매칭 결과(연산 결과)를 수집한다.

[0043] 집산부(140)는 개별 코어(20)에서 생성된 매칭 결과를 개별 코어(20)로부터 직접 수신할 수 있으며, 또는, 개별 코어(20)에서 생성된 매칭 결과를 개별 코어(20)가 포함된 참여 노드로부터 한번에 수신할 수 있다.

[0044] 집산부(140)는 개별 코어(20)로부터 수신된 매칭 결과를 합산하여 온톨로지 매핑을 생성한다. 온톨로지 매핑은 둘 이상의 후보 온톨로지의 구성요소 사이의 개별 매칭 결과의 집합이다. 그리고, 집산부(140)는 개별 코어(20)에서 매칭 작업의 연산에 의해 연산된 매칭 결과를 축적하고, 축적된 매칭 결과에 브릿지 패턴(Bridge Pattern)을 적용하여 온톨로지 매핑의 형식적 표현(Formal Representation)을 생성한다. 브릿지 패턴은 매핑 파일의 유형으로서, 개별 사용자에 따라 매핑 파일을 위해 정의된 개별 형식(custom defined format)이다. 예를 들어, 특정한 스키마(Scheme)와 함께 XML 포맷의 매핑 파일을 원하는 경우, 이는 다른 사용자가 정의한 매핑 포맷과 다를 수 있다. 이와 같은 모든 개별 형식은 브릿지 패턴으로 불린다. 즉, 집산부(140)는 브릿지 패턴을 통해 개별 사용자가 요구하는 개별 형식에 맞추어 온톨로지 매핑을 제공할 수 있다. 그리고, 집산부(140)는 사용

한 브릿지 패턴을 다른 사용자를 위해 저장한다.

[0045] 집산부(140)는 참여 노드의 개수에 따라 합산 과정이 달라질 수 있다.

수학식 3

[0046] $$O_b^{Job} \leftarrow \bigcup_{i=1}^{MT} O_b^i$$

[0047] $$O_b^{Node} \leftarrow \sum_{i=0}^{MJ} O_b^{job=i}$$

[0048] $$O_b \leftarrow \sum_{i=1}^{TotalNode} O_b^{Node=i}$$

[0049] 수학식 3은 둘 이상의 참여 노드가 연결된 분산 환경에서의 브릿지 온톨로지 생성을 위한 세 단계를 나타낸다. 첫 번째 단계는 모든 매칭 작업에서의 모든 단일 매칭 태스크의 결과를 모두 수집한다. 첫 번째 단계에서, O_b^i는 매칭 태스크의 매칭 결과(매칭 태스크의 브릿지 온톨로지)이고, O_b^{Job}은 매칭 작업의 중앙 브릿지 온톨로지(Intermediate Bridge Ontology)이다. 두 번째 단계는 하나의 참여 노드 상에서 실행되는 모든 매칭 작업의 결과를 합산한다. 두 번째 단계에서, O_b^{Node}는 단일 참여 노드에서 실행되는 모든 매칭 작업에서 모든 중앙 브릿지 온톨로지 O_b^{Job}의 합이다. 현재의 모든 노드에서 생성된 중앙 브릿지 온톨로지 O_b^{Job}는 노드로부터의 중앙 브릿지 온톨로지가 된다. 세 번째 단계는 분산 환경 상의 모든 참여 노드에서 실행되는 모든 매칭 요구의 모든 매칭 결과를 합산한다. 세 번째에서, O_b는 최종 생성되는 브릿지 온톨로지로서 O_b는 출력 파일인 온톨로지 매핑 결과로 전환된다.

수학식 4

[0050] $$O_b^{Job} \leftarrow \bigcup_{i=1}^{MT} O_b^i$$

[0051] $$O_b^{Node} \leftarrow \sum_{i=0}^{MJ} O_b^{job=i}$$

[0052] 수학식 4는 하나의 참여 노드가 연결된 분산 환경에서의 브릿지 온톨로지 생성을 위한 두 단계를 나타낸다. 참여 노드가 하나일 경우, 단일 참여 노드(CPU)의 개별 코어(20)에서 실행되는 매칭 작업의 결과를 합산해야 한다. 첫 번째 단계는 단일 매칭 작업에서 실행되는 모든 매칭 태스크의 결과 O_b^i를 통합하며, 매칭 태스크의 결과 O_b^i를 통합 결과는 매칭 작업의 중앙 브릿지 온톨로지 O_b^{Job}이다. 두 번째 단계는 하나의 참여 노드에서 실행되는 모든 매칭 작업의 결과를 합산한다. 단일 참여 노드에서 실행되는 매칭 작업에서 모든 중앙

브릿지 온톨로지 O_b^{Job} 의 합은 노드로부터의 중앙 브릿지 온톨로지 O_b^{Node} 가 된다.

[0053] 분산 환경에 참여한 모든 참여 노드에 의해 연산된 매칭 결과는 브릿지 온톨로지라고 불리는 하나의 온톨로지 오브젝트와 같이 합산된다. 그리고, 집산부(140)는 수집된 브릿지 온톨로지를 물리적 매핑 파일(Physical Mapping File)인 온톨로지 매핑 결과로 전환하여 출력한다.

[0054] 도 3은 본 발명의 일 실시예에 따른 라지 스케일 온톨로지 매칭 장치의 전처리부를 나타내는 상세도이다.

[0055] 도 3을 참조하면, 본 발명의 일 실시예에 따른 라지 스케일 온톨로지 매칭 장치(100)의 전처리부(110)는 온톨로지 모델부(111), 직렬화 처리부(112) 및 역직렬화 처리부(113)를 포함한다.

[0056] 온톨로지 모델부(111)는 소스 온톨로지 및 타겟 온톨로지를 포함하는 후보 온톨로지를 입력받는다. 소스 온톨로지 및 타겟 온톨로지를 포함하는 후보 온톨로지는 서로 1:1로 매칭할 수 있으며, 하나의 소스 온톨로지가 둘 이상의 타겟 온톨로지에 매칭하거나, 둘 이상의 소스 온톨로지가 하나의 타겟 온톨로지에 매칭할 수 있다.

[0057] 온톨로지 모델부(111)는 수신된 소스 온톨로지 및 타겟 온톨로지를 포함하는 후보 온톨로지를 하나 이상의 서브셋(Subset)으로 분류한다. 온톨로지는 일반적으로 5개의 서브셋으로 구성될 수 있다. 온톨로지를 구성하는 서브셋은 크게 이름(Name), 레이블(Lable), 관계(Relationship), 공리(Axiom) 및 특성(Property)으로 구분될 수 있다. 하지만, 온톨로지를 상술한 다섯 가지의 서브셋으로만 구분할 수 있는 것은 아니며, 온톨로지 모델부(111)는 수신된 후보 온토로지의 종류나 특성에 따라 숫자에 관계없이 하나 이상의 서브셋으로 분류할 수 있다. 또한, 온톨로지 모델부(111)는 매칭 알고리즘을 포함하는 매칭 라이브러리를 고려하여 후보 온톨로지를 하나 이상의 서브셋으로 분류할 수 있다. 온톨로지 모델부(111)는 분류된 온톨로지 서브셋을 분산 처리부(130)로 전달한다.

[0058] 직렬화 처리부(112)는 온톨로지 모델부(111)에서 생성된 온톨로지 서브셋을 직렬화한다. 직렬화 처리부(112)는 둘 이상의 서브셋으로 분류된 온톨로지 서브셋을 바이너리 형태로 직렬화하여 저장 공간을 효율적으로 사용할 수 있으며, 데이터 전송 및 처리의 효율성을 높일 수 있다. 직렬화 처리부(112)는 직렬화된 온톨로지 서브셋을 온톨로지 저장부(120)로 전달하여 저장할 수 있다. 온톨로지 저장부(120)는 직렬화된 온톨로지 서브셋을 저장함으로써 저장 공간을 효율적으로 사용할 수 있으며, 데이터 처리 효율을 높일 수 있다.

[0059] 그리고, 역직렬화 처리부(113)는 직렬화 처리부(112)에 의해 직렬화된 형태로 온톨로지 저장부(120)에 저장된 직렬화된 온톨로지 서브셋을 다시 원래의 온톨로지 서브셋 상태로 재구성한다. 이러한 과정을 통해, 역직렬화 처리부(113)는 온톨로지 저장부(120)에 바이너리 형식으로 직렬화하여 저장한 온톨로지 서브셋을 온톨로지 모델부(111)의 요구에 따라 전달함으로써, 온톨로지 모델부(111)에서 불필요한 전처리 과정의 반복(Re-processing)을 줄이거나 피할 수 있다. 즉, 이전에 분류하여 온톨로지 저장부(120)에 저장한 온톨로지 서브셋과 동일한 후보 온톨로지가 수신되는 경우, 서브셋으로 분류하는 전처리 과정을 반복하지 않고, 온톨로지 저장부(120)에 직렬화되어 저장된 온톨로지 서브셋을 바로 역직렬화 처리부(113)를 통해 다시 복원(재구성)하여 분산 처리부(130)로 전달할 수 있다. 이와 같은 과정을 통해 전처리 과정을 생략(Skip)함으로써, 전처리 과정에서 소모되는 연산 리소스를 줄일 수 있다.

[0060] 수신된 후보 온톨로지를 서브셋별로 세분화하여 서브셋 온톨로지를 생성하는 전처리부(110)의 전처리 과정은 실행 시간 동안 더 적은 메모리 풋프린트(Memory Footprint)를 가능하게 한다.

[0061] 도 4는 본 발명의 일 실시예에 따른 라지 스케일 온톨로지 매칭 장치의 분산 처리부를 나타내는 상세도이다.

[0062] 도 4를 참조하면, 본 발명의 일 실시예에 따른 라지 스케일 온톨로지 매칭 장치의 분산 처리부(130)는 병렬 전략부(131), 분산부(132) 및 병렬 인터페이스(133)를 포함한다.

[0063] 병렬 전략부(131)는 참여 노드의 개수와 각 참여 노드가 구비한 개별 코어(멀티 코어)의 수를 확인한다. 분산 환경은 많은 수의 참여 노드가 참여하게 된다. 참여 노드는 일반적으로 사용되는 데스크탑 및 노트북과 같은 개인용 컴퓨터(Personal Computer)뿐만 아니라, 워크스테이션(Workstation) 및 서버 컴퓨터와 같이 연산 처리가 가능한 단말이나 장치를 모드 포함할 수 있다. 본 발명에 따른 라지 스케일 온톨로지 매칭 장치는 후보 온톨로지의 매핑을 위한 온톨로지 매칭을 참여 노드의 개별 코어에 분산시켜 병렬 처리한다. 따라서, 병렬 전략부(131)는 먼저 참여 노드의 수와 각 첨여 노드가 구비한 개별 코어의 수를 확인하여, 가용 연산 자원을 파악한다. 그리고, 병렬 전략부(131)는 확인된 참여 노드 및 개별 코어의 수와 분산 알고리즘을 고려하여, 최적

의 분산 갯수를 설정한다.

[0064] 분산부(132)는 병렬 전략부(131)에서 참여 노드 및 개별 코어의 수와 분산 알고리즘을 고려하여 설정된 최적의 분산 갯수를 고려하여, 온톨로지 서브셋을 분할하고, 분할된 온톨로지 서브셋에 매칭 라이브러리의 매칭 알고리즘을 적용하여 매칭 스레드를 생성한다. 생성된 매칭 스레드는 매칭 요구, 매칭 작업 및 매칭 태스크를 포함한다. 매칭 라이브러리에 저장된 매칭 알고리즘은 후보 온톨로지 사이의 온톨로지 매핑을 생성하기 위해 후보 온톨로지를 구성하는 개별 구성요소 사이의 매칭을 연산하기 위한 알고리즘으로 동의어 기반 매칭(Synonym-based matching), 라벨 기반 매칭(Label-based matching), 광의어 기반 매칭(Broader Term-based matching) 및 자식 기반 매칭(Child-based matching) 등과 같이 온톨로지 매칭에 적용할 수 있는 매칭 알고리즘을 포함할 수 있다.

[0065] 분산부(132)에서 생성된 매칭 요구는 참여 노드(또는 참여 노드의 CPU) 각각에 대응하여 전달되는 매칭 스레드(Mathing Thread)이다. 병렬 전략부(131)는 먼저 참여 노드(또는 참여 노드의 CPU)의 수를 고려하여 온톨로지 서브셋을 각각의 참여 노드(또는 참여 노드의 CPU)에 대응하는 하나 이상의 매칭 요구로 분할한다. 매칭 요구의 최대 개수는 참여 노드(또는 참여 노드의 CPU)의 개수가 될 수 있다. 즉, 매칭 요구는 온톨로지 병렬 처리를 위해 개별 노드에 할당되는 병렬 처리 작업이 될 수 있다. 그리고, 분산부(132)는 분할된 매칭 요구를 다시 해당 매칭 요구가 할당되는 참여 노드의 코어의 수에 따라 매칭 작업으로 분할하여 해당 매칭 요구에 포함시킨다. 즉, 매칭 작업은 개별 코어에 할당되는 병렬 처리 작업으로서, 해당 매칭 요구가 할당된 참여 노드의 코어의 수가 4개의 코어를 가진다면, 해당 매칭 요구는 4개의 매칭 작업을 포함할 수 있다. 다음으로, 분산부(132)는 개별 코어에 할당되는 매칭 작업을 실제 매칭 프로세싱을 수행하는 단위인 매칭 태스크로 분할하여 매칭 작업에 포함시킨다.

[0066] 매칭 요구, 매칭 작업 및 매칭 태스크는 전체 매칭 프로세스를 위한 세 개의 추상화 계층(Layer of Abstraction)이다. 병렬 처리 프로세싱을 위한 분산 알고리즘에서 서로 다른 레벨에서 분류를 제공해야 하기 때문에, 분산부(132)의 분산 알고리즘은 실행중인 모든 작업의 트랙(Track)을 보존해야 한다.

[0067] 매칭 태스크는 매칭 프로세스의 유닛으로 가장 작은 전체 매칭 프로세스로 분류될 수 있다. 예를 들어, 해당 참여 노드의 코어에 할당된 매칭 작업에서 소스 온톨로지가 {A,B,C,D}이고 타겟 온톨로지가 {a,b,c,d}라면, 소스 온톨로지와 타겟 온톨로지를 비교하여 매칭 프로세스를 수행할 때, A=a는 첫 번째 MT, B=b는 두 번째 MT, C=c는 세 번째 MT, D=d는 네 번째 MT일 수 있으며, A=d를 비교하여 매칭할 수 있다면, 이 또한 다른 하나의 MT가 될 수 있다. 즉, 매칭 태스크는 참여 노드의 개별 코어 내부에서 개별 매칭 프로세싱을 처리하는 가장 작은 단위로서, 코어 내부에서 온톨로지를 구성하는 개별 용어(Term)의 매칭 프로세싱을 처리한다.

[0068] 매칭 작업은 참여 노드의 개별 코어에 할당되어 매칭 프로세스를 수행하는 단위로서, 하나 이상의 매칭 태스크의 모음이 될 수 있다. 그리고, 매칭 요구는 하나의 참여 노드(또는 참여 노드의 CPU)에서 실행되는 모든 매칭 작업의 모음이다.

[0069] 예를 들어, 4개의 참여 노드가 분산 환경에 존재하고, 4개의 참여 노드가 모두 4개의 코어를 가지며, 수신된 온톨로지 서브셋이 총 2000개의 매칭을 필요로 한다면, 분산부(132)는 총 2000개의 매칭을 참여 노드의 수에 따라 4등분하여 총 4개의 매칭 요구로 분할한다. 분할된 4개의 매칭 요구 각각은 500개의 매칭을 포함한다. 그리고, 분산부(132)는 4개의 매칭 요구 각각을 코어의 수를 고려하여 4개의 매칭 작업으로 분류한다. 그러면, 각각의 매칭 작업은 500개의 매칭을 4로 나눈 125개의 매칭을 포함한다. 결국, 하나의 매칭 작업은 125개의 매칭을 포함하며, 개별 매칭 작업에 포함된 매칭 태스크는 125개가 된다. 즉, 4개의 매칭 요청 각각은 4개의 매칭 작업을 포함하며, 4개의 매칭 작업 각각은 125개의 매칭 태스크를 포함하여 결국, 총 2000개의 매칭을 포함하는 하나 이상의 온톨로지 서브셋은 4개의 매칭 요청으로 분할되고, 각각의 매칭 요청은 4개의 매칭 작업을 포함하며, 각각의 매칭 작업은 125개의 매칭 태스크를 포함한다. 매칭 요청, 매칭 작업 및 매칭 태스크의 개수는 참여 노드 및 개별 참여 노드의 수뿐만 아니라, 가장 최적화된 병렬 처리를 수행할 수 있는 시나리오를 고려하여 설정될 수 있다. 그리고, 각각의 매칭 작업은 스스로 다른 매칭 작업 및 원격에서 동작하는 다른 매칭 요구에 독립적이다.

[0070] 병렬 인터페이스(133)는 참여 노드 및 각 참여 노드의 개별 코어 각각에 대응하는 매칭 스레드를 전달한다. 하나의 매칭 요청은 대응하는 하나의 참여 노드로 전달된다. 하나의 참여 노드로 전달된 하나의 매칭 요청은 해당 참여 노드에 포함된 하나 이상의 코어 각각에 대응하는 매칭 작업을 포함한다. 하나의 매칭 요청에 포함된 하나 이상의 매칭 작업은 대응하는 코어에 일 대 일로 할당된다. 그리고, 하나의 코어에 할당된 하나의 매칭 작업은 포함된 매칭 태스크를 통해 해당 코어에서 매칭 프로세싱을 수행한다. 이를 통해, 본 발명에 따른 라지 스케일 온톨로지 매칭 장치는 분산 환경을 구성하는 참여 노드의 코어 각각에서 매칭 프로세싱을 병렬처리할 수 있다.

[0071] 도 5는 본 발명의 일 실시예에 따른 라지 스케일 온톨로지 매칭 장치에서 개별 코어로 전달되는 매칭 스레드의 일 실시예를 나타내는 도면이다.

[0072] 도 5를 참조하면, 본 발명에 따른 라지 스케일 온톨로지 매칭 장치의 분산 처리부에서 생성된 매칭 스레드는 각각의 노드 및 각각의 개별 노드에 대응하는 매칭 요청, 매칭 작업 및 매칭 태스크를 포함할 수 있다. 예를 들어, 분산 환경에 제1 노드(510), 제2 노드(520) 및 제3 노드(530)를 포함하는 세 개의 참여 노드가 연결되어 있다고 가정한다. 제1 노드(510)는 두 개의 코어(511,512)를 포함하고, 제2 노드(520)는 네 개의 코어(521,522,523,524)를 포함하며, 제3 노드(530)는 두 개의 코어(531,532)를 포함한다. 라지 스케일 온톨로지 매칭 장치는 소스 온톨로지 및 타겟 온톨로지를 포함하는 후보 온톨로지로부터 전처리 과정 및 분산 처리 과정을 통해 둘 이상의 매칭 스레드를 생성하고, 생성된 둘 이상의 매칭 스레드를 각 참여 노드의 개별 코어에 전달한다.

[0073] 매칭 요구는 참여 노드(또는 참여 노드의 CPU) 각각에 대응하여 전달되는 매칭 스레드(Mathing Thread)이다. 분산 처리부(130)는 먼저 참여 노드(또는 참여 노드의 CPU)의 수를 고려하여 온톨로지 서브셋을 각각의 참여 노드(또는 참여 노드의 CPU)에 대응하는 하나 이상의 매칭 요구로 분할한다. 매칭 요구의 최대 개수는 참여 노드(또는 참여 노드의 CPU)의 개수가 될 수 있다. 즉, 매칭 요구는 온톨로지 병렬 처리를 위해 개별 노드에 할당되는 병렬 처리 작업이 될 수 있다. 그리고, 라지 스케일 온톨로지 매칭 장치는 분할된 매칭 요구를 다시 해당 매칭 요구가 할당되는 참여 노드의 코어의 수에 따라 매칭 작업으로 분할하여 해당 매칭 요구에 포함시킨다. 즉, 매칭 작업은 개별 코어에 할당되는 병렬 처리 작업으로서, 해당 매칭 요구가 할당된 참여 노드의 코어의 수가 4개의 코어를 가진다면, 해당 매칭 요구는 4개의 매칭 작업을 포함할 수 있다. 다음으로, 분산 처리부(130)는 개별 코어에 할당되는 매칭 작업을 실제 매칭 프로세싱을 수행하는 단위인 매칭 태스크로 분할하여 매칭 작업에 포함시킨다.

[0074] 도 5에서는 총 세 개의 노드(510,520,530)를 포함하고 있으며, 라지 스케일 온톨로지 매칭 장치는 후보 온톨로지를 서브셋으로 분류하고, 분산 알고리즘 및 매칭 알고리즘을 적용하여 세 개의 매칭 요구(MR1, MR2, MR3)를 포함하는 매칭 스레드를 생성한다. 제1 매칭 요구(MR1)는 제1 노드(510)에 대응하고, 제2 매칭 요구(MR2)는 제2 노드(520)에 대응하며, 제3 매칭 요구(MR3)는 제3 노드(530)에 각각 대응한다. 제1 매칭 요구(MR1)는 제1 노드(510)가 구비한 두 개의 코어(511,512)에 대응하여 두 개의 매칭 작업(511-1,512-1)을 포함하며, 제2 매칭 요구(MR2)는 제2 노드(520)가 구비한 네 개의 코어(521,522,523,524)에 대응하여 네 개의 매칭 작업(521-1,522-1,523-1,524-1)을 포함하고, 제3 매칭 요구(MR3)는 제3 노드(530)가 구비한 두 개의 코어(531,532)에 대응하여 두 개의 매칭 작업(531-1,532-1)을 포함한다.

[0075] 제1 내지 제3 매칭 요구(MR1,MR2,MR) 및 각 매칭 요구에 포함된 매칭 작업(511-1,512-1,521-1,522-1,523-1,524-1,531-1,532-1)은 분산 환경에 연결된 참여 노드(510,520,530) 및 각 참여 노드의 개별 코어(511,512,521,522,523,524,531,532)에 대응하여 생성된다. 생성된 제1 내지 제3 매칭 요구(MR1,MR2,MR3)는 각각 대응하는 제1 노드(510), 제2 노드(520) 및 제3 노드(530) 각각에 전달된다. 라지 스케일 온톨로지 매칭 장치에서 각 노드로 매칭 요구가 전달되는 과정은 라지 스케일 온톨로지 매칭 장치로부터 각 노드로 직접 전달될 수 있으며, 분산 환경의 구조에 따라 라지 스케일 온톨로지 매칭 장치로부터 하나의 노드로 전달된 후, 연결된 노드를 따라 순차적으로 전달될 수 있다.

[0076] 도 6은 본 발명의 일 실시예에 따른 라지 스케일 온톨로지 매칭 장치의 데이터 흐름을 나타내는 도면이다.

[0077] 도 6을 참조하면, 본 발명의 일 실시예에 따른 라지 스케일 온톨로지 매칭 장치의 데이터 흐름은 먼저 전처리부(110)는 소스 온톨로지 및 타겟 온톨로지를 포함하는 후보 온톨로지를 수신한다(S601). 소스 온톨로지 및 타겟 온톨로지를 포함하는 후보 온톨로지는 서로 1:1로 매칭할 수 있으며, 하나의 소스 온톨로지가 둘 이상의 타겟 온톨로지에 매칭하거나, 둘 이상의 소스 온톨로지가 하나의 타겟 온톨로지에 매칭할 수 있다.

[0078] 그리고, 전처리부(110)는 수신된 소스 온톨로지 및 타겟 온톨로지를 포함하는 후보 온톨로지를 하나 이상의 서브셋으로 분류한다(S602). 온톨로지는 일반적으로 5개의 서브셋으로 구성될 수 있다. 온톨로지를 구성하는 서브셋은 크게 이름, 레이블, 관계, 공리 및 특성으로 구분될 수 있다. 하지만, 온톨로지를 상술한 다섯 가지의 서브셋으로만 구분할 수 있는 것은 아니며, 전처리부(110)는 수신된 후보 온토로지의 종류나 특성에 따라 숫자에 관계없이 하나 이상의 서브셋으로 분류할 수 있다. 그리고, 전처리부(110)는 분류된 온톨로지 서브셋을 분산 처리부(130)로 전달한다(S603).

[0079] 다음으로 전처리부(110)는 분류된 온톨로지 서브셋을 직렬화한다(S604). 전처리부(110)는 온톨로지 서브셋을 바

이너리 형태로 직렬화하여 데이터 처리 효율 및 저장 공간의 효율을 높일 수 있다. 전처리부(110)는 직렬화된 온톨로지 서브셋을 온톨로지 저장부(120)로 전달하여 저장한다(S605). 이 후, 동일한 후보 온톨로지가 수신되는 경우, 온톨로지 저장부(120)에 저장된 대응하는 온톨로지 서브셋을 호출하여 사용한다. 이를 통해, 서브셋으로 분류하는 전처리 과정을 생략함으로써, 시간 및 연산 자원을 절약할 수 있다.

[0080] 전처리부(110)로부터 분산 처리부(130)로 온톨로지 서브셋이 전달(S603)되면, 분산 처리부(130)는 참여 노드(30)의 개수와 각 참여 노드(30)에 구비된 개별 코어의 개수를 확인한다(S606). 분산 처리부(130)는 먼저 참여 노드(30)의 수와 각 첨여 노드가 구비한 개별 코어의 수를 확인하여, 가용 연산 자원을 파악한다. 그리고, 분산 처리부(130)는 확인된 참여 노(30)드 및 개별 코어의 수와 분산 알고리즘을 고려하여, 최적의 분산 갯수를 설정한다.

[0081] 분산 처리부(130)는 참여 노드(30) 및 개별 코어의 수와 분산 알고리즘을 고려하여 설정된 최적의 분산 갯수를 고려하여, 온톨로지 서브셋을 분할하고(S607), 분할된 온톨로지 서브셋에 매칭 라이브러리의 매칭 알고리즘을 적용하여 매칭 스레드를 생성한다(S608). 생성된 매칭 스레드는 매칭 요구, 매칭 작업 및 매칭 태스크를 포함한다. 매칭 라이브러리에 저장된 매칭 알고리즘은 후보 온톨로지 사이의 온톨로지 매핑을 생성하기 위해 후보 온톨로지를 구성하는 개별 구성요소 사이의 매칭을 연산하기 위한 알고리즘으로 동의어 기반 매칭(Synonym-based matching), 라벨 기반 매칭(Label-based matching), 광의어 기반 매칭(Broader Term-based matching) 및 자식 기반 매칭(Child-based matching) 등과 같이 온톨로지 매칭에 적용할 수 있는 매칭 알고리즘을 포함할 수 있다.

[0082] 분산 처리부(130)는 먼저 참여 노드(30)의 수를 고려하여 온톨로지 서브셋을 각각의 참여 노드(30)에 대응하는 하나 이상의 매칭 요구로 분할한다. 매칭 요구의 최대 개수는 참여 노드(30)의 개수가 될 수 있다. 그리고, 분산 처리부(130)는 분할된 매칭 요구를 다시 해당 매칭 요구가 할당되는 참여 노드(30)의 코어의 수에 따라 매칭 작업으로 분할하여 해당 매칭 요구에 포함시킨다. 매칭 요구, 매칭 작업 및 매칭 태스크는 전체 매칭 프로세스를 위한 세 개의 추상화 계층(Layer of Abstraction)이다. 병렬 처리 프로세싱을 위한 분산 알고리즘에서 서로 다른 레벨에서 분류를 제공해야 하기 때문에, 분산부(132)의 분산 알고리즘은 실행중인 모든 작업의 트랙(Track)을 보존해야 한다. 매칭 요청, 매칭 작업 및 매칭 태스크의 개수는 참여 노드(30) 및 개별 참여 노드(30)의 수뿐만 아니라, 가장 최적화된 병렬 처리를 수행할 수 있는 시나리오를 고려하여 설정될 수 있다. 그리고, 각각의 매칭 작업은 스스로 다른 매칭 작업 및 원격에서 동작하는 다른 매칭 요구에 독립적이다.

[0083] 매칭 스레드를 생성한 분산 처리부(130)는 생성된 각각의 매칭 스레드를 대응하는 참여 노드(30)에 전달한다(S609). 그리고, 분산 처리부(130)로부터 대응하는 매칭 스레드를 수신받은 각 노드는 수신된 매칭 스레드의 매칭 요청에 포함된 매칭 작업을 구비된 개별 노드에 할당한다. 하나의 매칭 요청은 대응하는 하나의 참여 노드(30)로 전달된다. 하나의 참여 노드(30)로 전달된 하나의 매칭 요청은 해당 참여 노드(30)에 포함된 하나 이상의 코어 각각에 대응하는 매칭 작업을 포함한다. 하나의 매칭 요청에 포함된 하나 이상의 매칭 작업은 대응하는 코어에 일 대 일로 할당되다.

[0084] 참여 노드(30)에 의해 매칭 스레드에 포함된 매칭 작업을 할당받은 참여 노드(30)의 개별 코어는 매칭 작업에 포함된 매칭 태스크에 기초하여 매칭 연산을 수행한다(S610). 개별 코어에 할당된 매칭 작업은 하나 이상의 매칭 태스크를 포함한다. 매칭 태스크는 후보 온톨로지를 구성하는 구성 요소의 매칭 연산을 수행하기 위한 매칭 알고리즘 및 세분화되어 분할된 온톨로지 서브셋을 포함한다. 개별 코어는 매칭 태스크에 기초하여 온톨로지 매칭 연산을 수행한다. 그리고, 참여 노드(30)는 개별 코어에서 수행된 매칭 연산에 의해 생성된 매칭 결과를 집산부(140)로 전달한다(S611). 집산부(140)는 분산 처리부(130)로부터 매칭 스레드를 전달받은 참여 노드로부터 매칭 스레드를 통해 매칭 연산을 수행한 매칭 결과를 수신한다. 그리고, 집산부(140)는 참여 노드(30)로부터 수신된 매칭 결과를 합산하여 온톨로지 매핑을 생성한다(S612). 온톨로지 매핑은 둘 이상의 후보 온톨로지의 구성요소 사이의 개별 매칭 결과의 집합이다. 그리고, 집산부(140)는 참여 노드(30)의 개별 코어에서 매칭 작업의 연산에 의해 연산된 매칭 결과를 축적하고, 축적된 매칭 결과에 브릿지 패턴(Bridge Pattern)을 적용하여 온톨로지 매핑의 형식적 표현(Formal Representation)을 생성한다. 브릿지 패턴은 매핑 파일의 유형으로서, 개별 사용자에 따라 매핑 파일을 위해 정의된 개별 형식(custom defined format)이다. 즉, 집산부(140)는 브릿지 패턴을 통해 개별 사용자가 요구하는 개별 형식에 맞추어 온톨로지 매핑을 제공할 수 있다. 그리고, 집산부(140)는 사용한 브릿지 패턴을 다른 사용자를 위해 저장한다.

[0085] 도 7은 본 발명의 일 실시예에 따른 라지 스케일 온톨로지 매칭 방법을 나타내는 흐름도이다.

[0086] 도 7을 참조하면, 본 발명의 일 실시예에 따른 라지 스케일 온톨로지 매칭 방법은 먼저, 수신된 소스 온톨로지 및 타겟 온톨로지를 포함하는 후보 온톨로지를 하나 이상의 서브셋으로 분류한다(S701). 온톨로지는 일반적으로

5개의 서브셋으로 구성될 수 있다. 온톨로지를 구성하는 서브셋은 크게 이름, 레이블, 관계, 공리 및 특성으로 구분될 수 있다. 하지만, 온톨로지를 상술한 다섯 가지의 서브셋으로만 구분할 수 있는 것은 아니며, 수신된 후보 온토로지의 종류나 특성에 따라 숫자에 관계없이 하나 이상의 서브셋으로 분류할 수 있다.

[0087] 후보 온톨로지가 하나 이상의 서브셋으로 분류되면, 참여 노드의 개수와 각 참여 노드에 구비된 개별 코어의 개수를 확인한다(S702). 라지 스케일 온톨로지 매칭 장치는 먼저 참여 노드의 수와 각 첨여 노드가 구비한 개별 코어의 수를 확인하여, 가용 연산 자원을 파악한다. 그리고, 라지 스케일 온톨로지 매칭 장치는 확인된 참여 노드 및 개별 코어의 수와 분산 알고리즘을 고려하여, 최적의 분산 갯수를 설정한다(S703).

[0088] 참여 노드 및 개별 코어의 개수를 분산 알고리즘에 적용하여 최적의 분산 갯수가 설정되면, 설정된 최적의 분산 갯수를 고려하여, 온톨로지 서브셋을 분할하고, 분할된 온톨로지 서브셋에 매칭 라이브러리의 매칭 알고리즘을 적용하여 매칭 스레드를 생성한다(S704). 생성된 매칭 스레드는 매칭 요구, 매칭 작업 및 매칭 태스크를 포함한다. 매칭 라이브러리에 저장된 매칭 알고리즘은 후보 온톨로지 사이의 온톨로지 매핑을 생성하기 위해 후보 온톨로지를 구성하는 개별 구성요소 사이의 매칭을 연산하기 위한 알고리즘이다. 라지 스케일 온톨로지 매칭 장치는 먼저 참여 노드의 수를 고려하여 온톨로지 서브셋을 각각의 참여 노드에 대응하는 하나 이상의 매칭 요구로 분할한다. 그리고, 분할된 매칭 요구를 다시 해당 매칭 요구가 할당되는 참여 노드의 코어의 수에 따라 매칭 작업으로 분할하여 해당 매칭 요구에 포함시킨다.

[0089] 매칭 스레드가 생성되면, 생성된 각각의 매칭 스레드를 대응하는 참여 노드에 전달한다(S705). 매칭 스레드에 포함된 매칭 작업을 할당받은 참여 노드의 개별 코어는 매칭 작업에 포함된 매칭 태스크에 기초하여 매칭 연산을 수행한다.

[0090] 참여 노드의 개별 코어에서 매칭 스레드에 기초한 매칭 연산이 수행되면, 매칭 연산에 의한 매칭 결과를 수집하여 합산한다(S706). 라지 스케일 온톨로지 매칭 장치는 참여 노드의 개별 코어에서 수행된 매칭 연산에 의해 생성된 매칭 결과를 수집하고, 수집된 매칭 결과를 합산하여 온톨로지 매핑을 생성한다(S707). 온톨로지 매핑은 둘 이상의 후보 온톨로지의 구성요소 사이의 개별 매칭 결과의 집합이다. 그리고, 라지 스케일 온톨로지 매칭 장치는 참여 노드의 개별 코어에서 매칭 작업의 연산에 의해 연산된 매칭 결과를 축적하고, 축적된 매칭 결과에 브릿지 패턴을 적용하여 온톨로지 매핑의 형식적 표현을 생성한다.

[0091] 도 8은 본 발명의 일 실시예에 따른 라지 스케일 온톨로지 매칭 방법의 전처리 생략 방법을 나타내는 흐름도이다.

[0092] 도 8을 참조하면, 본 발명의 일 실시예에 따른 라지 스케일 온톨로지 매칭 방법에서 전처리 과정을 생략하기 위해서는 먼저, 수신된 후보 온톨로지를 서브셋으로 분류한다(S801). 온톨로지는 일반적으로 5개의 서브셋으로 구성될 수 있다. 온톨로지를 구성하는 서브셋은 크게 이름, 레이블, 관계, 공리 및 특성으로 구분될 수 있다.

[0093] 온톨로지 서브셋으로 분류되면, 분류된 온톨로지 서브셋을 직렬화한다(S802). 둘 이상의 서브셋으로 분류된 온톨로지 서브셋을 바이너리 형태로 직렬화하여 저장 공간을 효율적으로 사용할 수 있으며, 데이터 전송 및 처리의 효율성을 높일 수 있다. 그리고, 직렬화된 온톨로지 서브셋을 저장한다(S803). 온톨로지 서브셋을 직렬화하여 저장함으로써 저장 공간을 효율적으로 사용할 수 있으며, 데이터 처리 효율을 높일 수 있다.

[0094] 이 후, 이전에 수신되었던 후보 온톨로지와 동일한 후보 온톨로지가 수신되면, 직렬화된 형태로 저장된 직렬화된 온톨로지 서브셋을 다시 원래의 온톨로지 서브셋 상태로 재구성(De-serialize)한다(S804). 이러한 과정을 통해, 직렬화된 저장한 온톨로지 서브셋을 다시 역직렬화함으로써, 불필요한 전처리 과정의 반복(Re-processing)을 줄이거나 피할 수 있다. 즉, 이전에 저장한 온톨로지 서브셋과 동일한 후보 온톨로지가 수신되는 경우, 서브셋으로 분류하는 전처리 과정을 반복하지 않고, 직렬화된 온톨로지 서브셋을 바로 역직렬화하여 다시 복원(재구성)하여 전처리 과정을 생략함으로써, 전처리 과정에서 소모되는 연산 리소스를 줄일 수 있다.

[0095] 역직렬화를 통해 재구성된 온톨로지 서브셋을 분할하여 참여 노드의 개별 코어에서 분산처리한다(S805). 먼저, 참여 노드의 개수와 각 참여 노드에 구비된 개별 코어의 개수를 확인한다. 그리고, 라지 스케일 온톨로지 매칭 장치는 확인된 참여 노드 및 개별 코어의 수와 분산 알고리즘을 고려하여, 최적의 분산 갯수를 설정한다. 참여 노드 및 개별 코어의 개수를 분산 알고리즘에 적용하여 최적의 분산 갯수가 설정되면, 설정된 최적의 분산 갯수를 고려하여, 온톨로지 서브셋을 분할하고, 분할된 온톨로지 서브셋에 매칭 라이브러리의 매칭 알고리즘을 적용하여 매칭 스레드를 생성한다. 생성된 매칭 스레드는 매칭 요구, 매칭 작업 및 매칭 태스크를 포함한다. 매칭 스레드가 생성되면, 생성된 각각의 매칭 스레드를 대응하는 참여 노드에 전달한다. 매칭 스레드에 포함된 매칭 작업을 할당받은 참여 노드의 개별 코어는 매칭 작업에 포함된 매칭 태스크에 기초하여 매칭 연산을 수행한다.

참여 노드의 개별 코어에서 매칭 스레드에 기초한 매칭 연산이 수행되면, 매칭 연산에 의한 매칭 결과를 수집하여 합산한다. 라지 스케일 온톨로지 매칭 장치는 참여 노드의 개별 코어에서 수행된 매칭 연산에 의해 생성된 매칭 결과를 수집하고, 수집된 매칭 결과를 합산하여 온톨로지 매핑을 생성한다. S805 단계는 도 7의 S703 내지 S707 단계와 동일하게 수행될 수 있다.

[0096] 상술한 내용을 포함하는 본 발명은 컴퓨터 프로그램으로 작성이 가능하다. 그리고 상기 프로그램을 구성하는 코드 및 코드 세그먼트는 당분야의 컴퓨터 프로그래머에 의하여 용이하게 추론될 수 있다. 또한, 상기 작성된 프로그램은 컴퓨터가 읽을 수 있는 기록매체 또는 정보저장매체에 저장되고, 컴퓨터에 의하여 판독되고 실행함으로써 본 발명의 방법을 구현할 수 있다. 그리고 상기 기록매체는 컴퓨터가 판독할 수 있는 모든 형태의 기록매체를 포함한다.

[0097] 이상 바람직한 실시예를 들어 본 발명을 상세하게 설명하였으나, 본 발명은 전술한 실시예에 한정되지 않고, 본 발명의 기술적 사상의 범위 내에서 당분야에서 통상의 지식을 가진자에 의하여 여러 가지 변형이 가능하다.

부호의 설명

[0098] 100: 온톨로지 매칭 장치

110: 전처리부

111: 온톨로지 모델부

112: 직렬화 처리부

113: 역직렬화 처리부

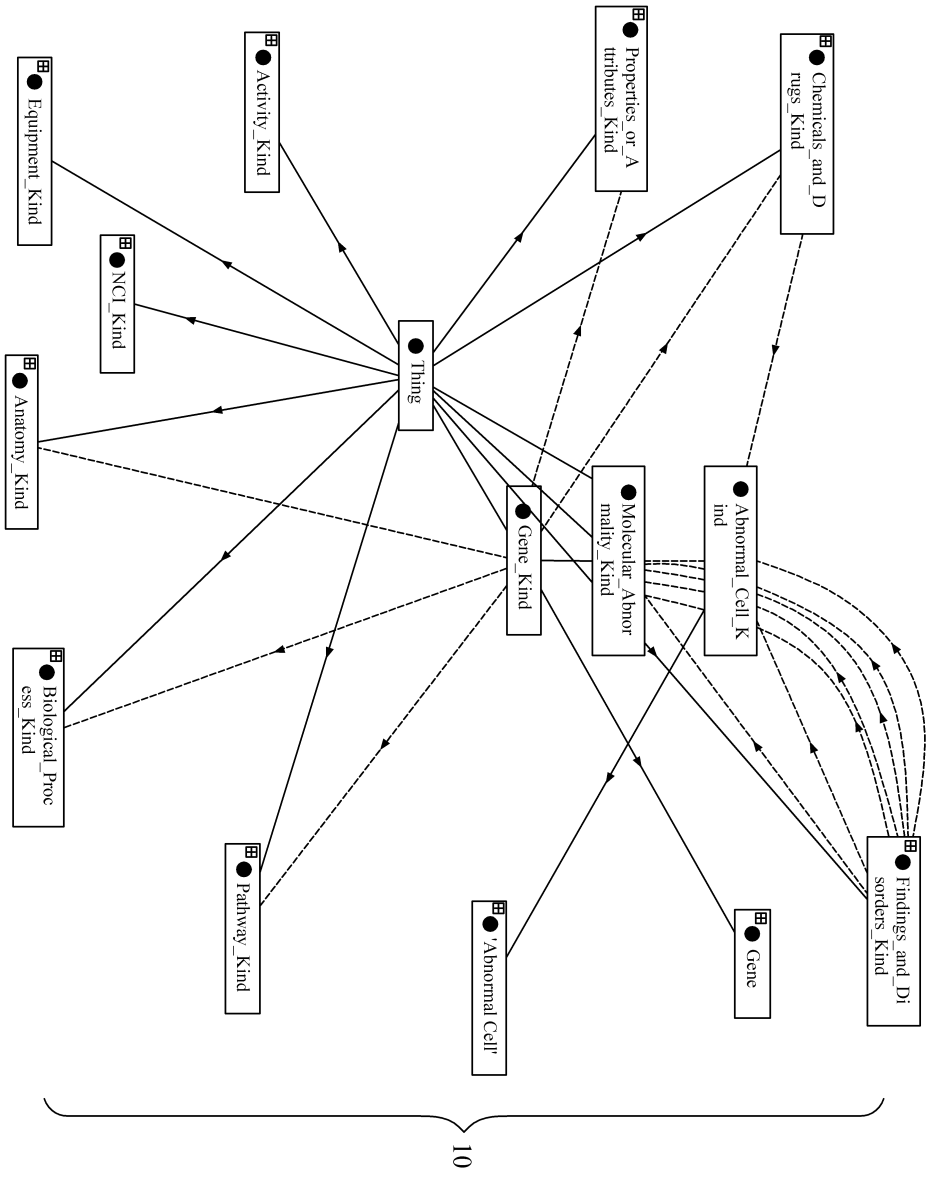
120: 온톨로지 저장부

130: 분산 처리부

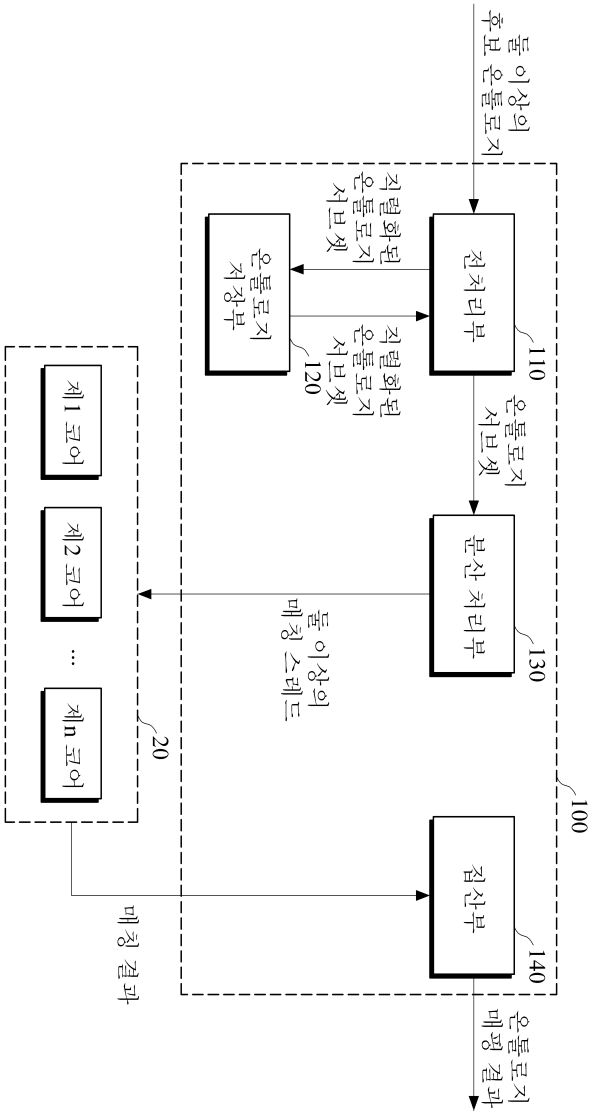
140: 집산부

도면

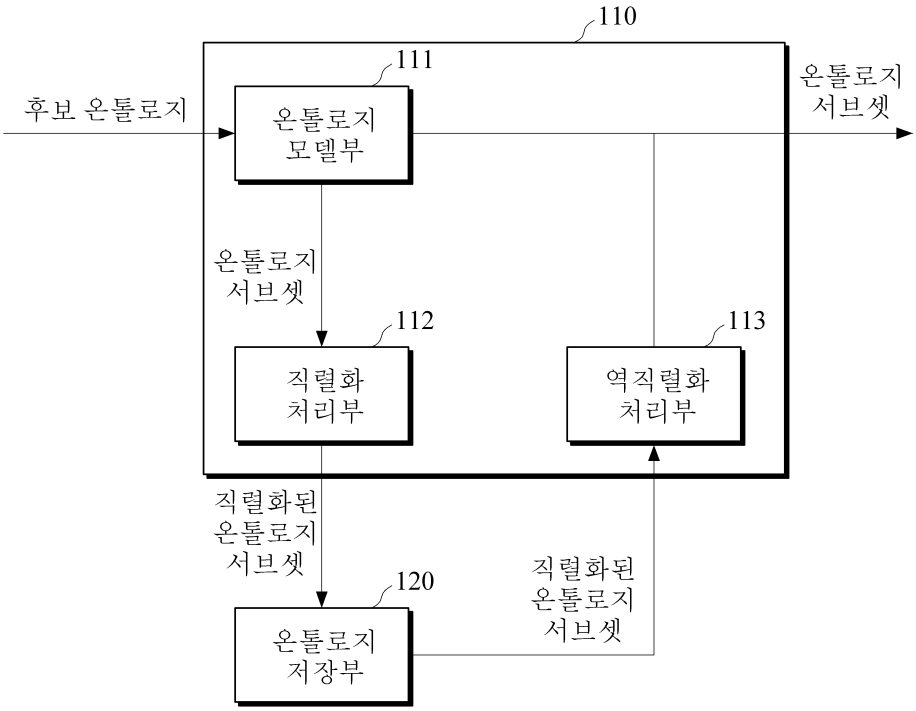
도면1


Thing
Activity_Kind
Properties_or_Attributes_Kind
Chemicals_and_Drugs_Kind
Equipment_Kind
NCI_Kind
Anatomy_Kind
Gene_Kind
Molecular_Abnormality_Kind
Abnormal_Cell_Kind
Biological_Process_Kind
Pathway_Kind
'Abnormal Cell'
Gene
Findings_and_Disorders_Kind
10

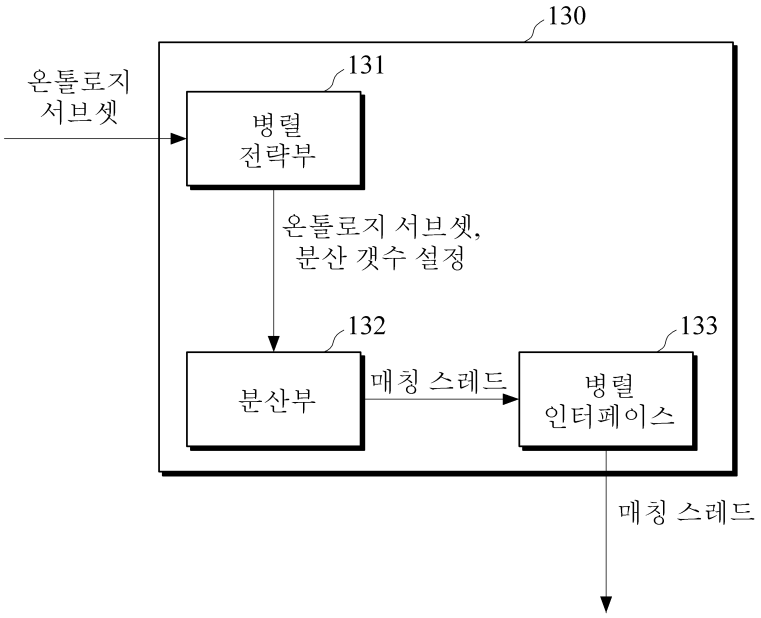
도면2


둘 이상의
후보 온톨로지
전처리부
110
직렬화된
온톨로지
서브셋
온톨로지
저장부
직렬화된
온톨로지
서브셋
120
온톨로지
서브셋
분산 처리부
130
둘 이상의
매칭 스레드
제1 코어
제2 코어
...
제n 코어
20
매칭 결과
집산부
140
100
온톨로지
매핑 결과

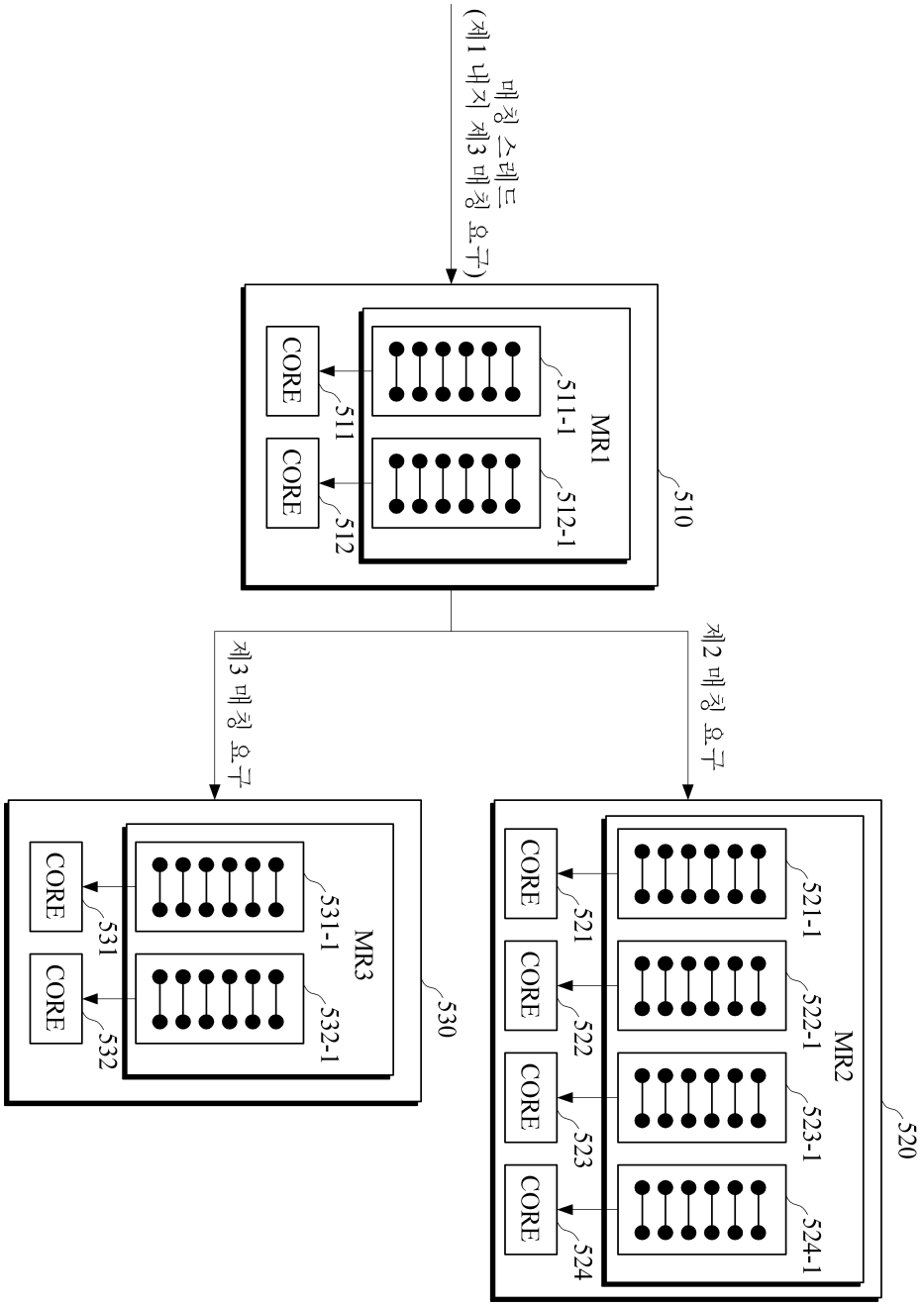
도면3


110
111
후보 온톨로지
온톨로지
모델부
온톨로지
서브셋
온톨로지
서브셋
112
113
직렬화
처리부
역직렬화
처리부
직렬화된
온톨로지
서브셋
120
직렬화된
온톨로지
서브셋
온톨로지
저장부

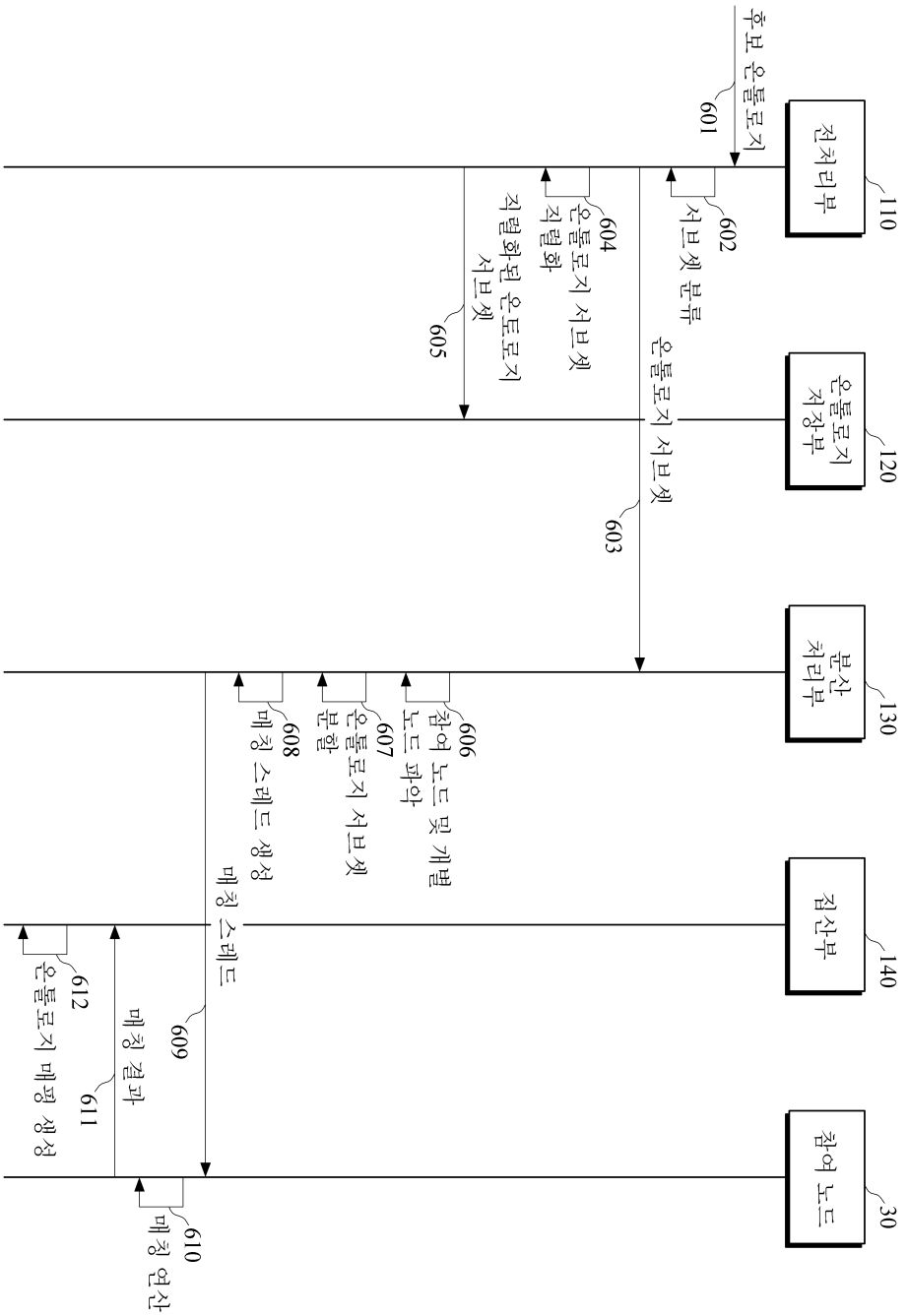
도면4


130
131
온톨로지
서브셋
병렬
전략부
온톨로지 서브셋,
분산 갯수 설정
132
133
분산부
매칭 스레드
병렬
인터페이스
매칭 스레드

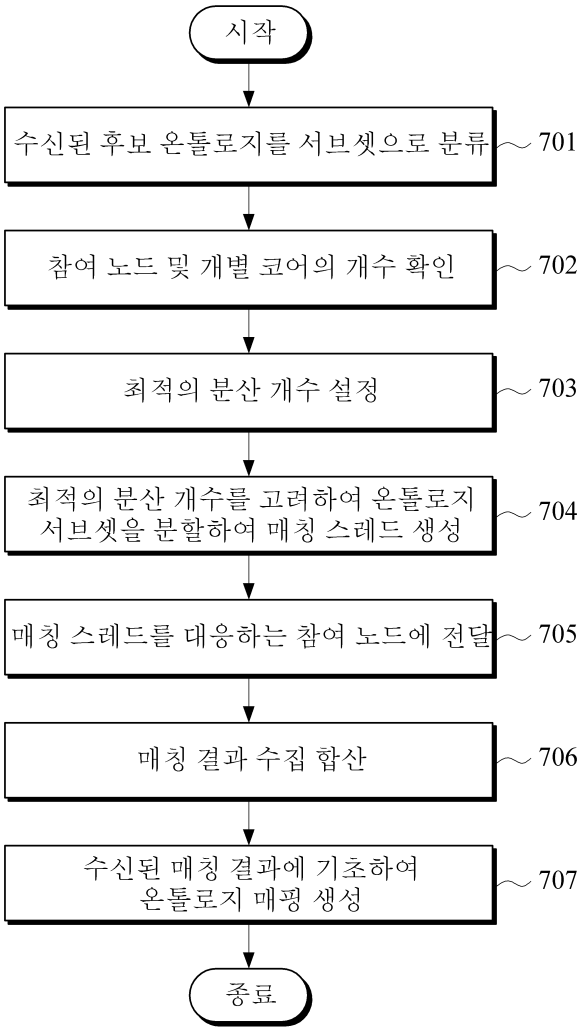
도면5


매칭 스레드
(제1 내지 제3 매칭 요구)
MR1
511-1
512-1
CORE
511
512
510
제2 매칭 요구
제3 매칭 요구
MR2
521-1
522-1
523-1
524-1
521
522
523
524
520
MR3
531-1
532-1
531
532
530

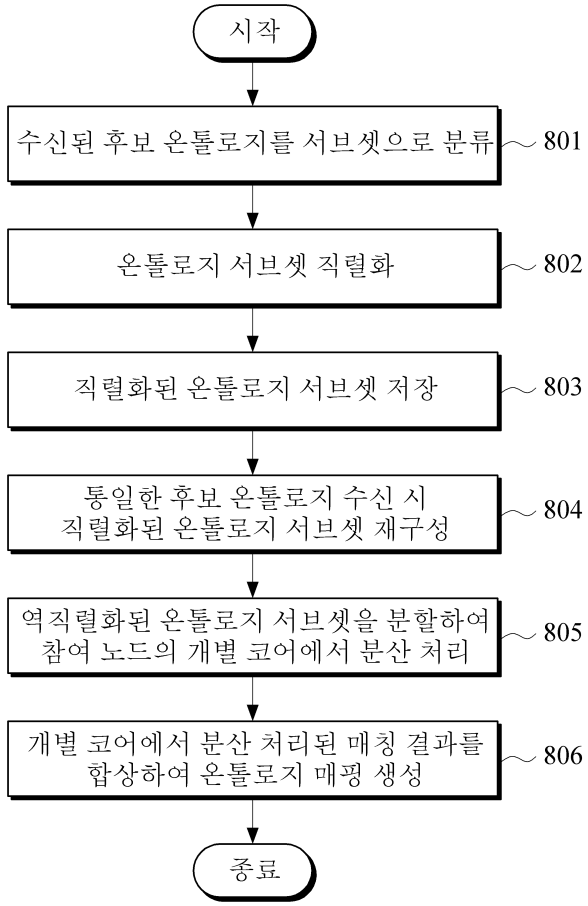
도면6


전처리부
110
온톨로지 저장부
120
분산 처리부
130
집산부
140
참여 노드
30
후보 온톨로지
601
서브셋 분류
602
온톨로지 서브셋
603
온톨로지 서브셋 직렬화
604
직렬화된 온토로지 서브셋
605
참여 노드 및 개별 노드 파악
606
온톨로지 서브셋 분할
607
매칭 스레드 생성
608
매칭 스레드
609
매칭 연산
610
매칭 결과
611
온톨로지 매핑 생성
612

도면7


시작
수신된 후보 온톨로지를 서브셋으로 분류
701
참여 노드 및 개별 코어의 개수 확인
702
최적의 분산 개수 설정
703
최적의 분산 개수를 고려하여 온톨로지 서브셋을 분할하여 매칭 스레드 생성
704
매칭 스레드를 대응하는 참여 노드에 전달
705
매칭 결과 수집 합산
706
수신된 매칭 결과에 기초하여 온톨로지 매핑 생성
707
종료

도면8


시작
수신된 후보 온톨로지를 서브셋으로 분류
801
온톨로지 서브셋 직렬화
802
직렬화된 온톨로지 서브셋 저장
803
통일한 후보 온톨로지 수신 시
직렬화된 온톨로지 서브셋 재구성
804
역직렬화된 온톨로지 서브셋을 분할하여
참여 노드의 개별 코어에서 분산 처리
805
개별 코어에서 분산 처리된 매칭 결과를
합상하여 온톨로지 매핑 생성
806
종료