1일 1지문으로 1등급 달성 - 배인호 초격차(超格差) 국어 제공

160/200

新수능 국어 최적화 기출 분석

2021학년도 수능완성 실전 모의고사 3회
[16~19] 다음 글을 읽고 물음에 답하시오.

빅 데이터란 기존의 데이터베이스에서는 처리할 수 없을 정도로 대용량인 데이터를 말한다. 대용량의 데이터를 효율적으로 처리하기 위해서는 새로운 데이터 처리 기술이 필요하다. 그래서 컴퓨터 기술 개발자들은 데이터 처리 기술을 향상하기 위한 많은 기술을 개발해 왔는데, 그중 대표적인 기술이 맵리듀스이다.

맵리듀스는 여러 대의 서버*가 연결되어 하나의 시스템처럼 작동하는 컴퓨터 클러스터 환경에서 대용량의 데이터를 병렬 처리하기 위해 만들어진 프로그램 기술이다. 대용량의 데이터를 빠르게 처리하기 위해서는 데이터를 병렬적으로 처리하는 것이 효과적이다. 데이터를 일정한 블록 단위로 쪼개 여러 개의 서버에 분산한 후 병렬적으로 처리하면 데이터 처리 시간을 크게 줄일 수 있기 때문이다.

맵리듀스는 이렇게 분산된 데이터를 각 서버에서 처리하는 ㉠ 매핑과, 이 작업의 결과를 다시 몇 개의 리듀스 태스크*로 취합하여 처리하는 ㉡ 리듀싱의 과정을 기반으로 구성된다. 맵리듀스의 이 과정을 더 자세히 살펴보면 분할, 매핑, 셔플링, 리듀싱 등 네 개의 과정으로 나눌 수 있다.

데이터에 등장하는 단어별로 사용 빈도수를 파악하는 작업을 한다고 가정해 보자. 입력된 데이터는 일정한 크기로 나누어져 각 서버에 배포되는데 이 과정을 '분할'이라고 한다. 이후 분할된 데이터를 <키, 값>의 형태를 띤 레코드*로 생성하는 '매핑'이 진행된다. 이 경우 주어진 문장에서 사용된 단어의 사용 빈도수를 조사하는 것이 목적이므로, 공백을 기준으로 단어를 추출한 후 각각의 단어와 1로 구성된 레코드를 만들어 준다. 만약 분할된 데이터가 'The lion ate the cow.'라는 문장이었다면, <the, 1>, <lion, 1>, <ate, 1>, <the, 1>, <cow, 1>과 같이 단어 하나마다 하나의 레코드가 생성되는 것이다.

이렇게 만들어진 레코드는 '셔플링' 작업을 거치게 된다. 이 단계는 각각의 레코드에서 키가 같은 것들을 모아 <키, [값 1, 값 2, …]>와 같이 병합하여 새로운 레코드를 생성하는 것이다. 예를 들어 'the'가 2번 등장했으므로 <the, [1, 1]>이라는 레코드가 생성되는 것이다. 이러한 과정을 모두 거치면 <the, [1, 1]>, <lion, [1]>, <ate, [1]>, <cow, [1]>과 같이 레코드의 수를 줄일 수 있다. 그런 다음 레코드를 몇 개의 리듀스 태스크로 분산시키기 위해 레코드마다 고유한 해시 코드를 부여한다. 그리고 각각의 해시 코드를 리듀스 태스크의 개수로 나눈 후 그 나머지 값에 따라 레코드가 분산될 리듀스 태스크의 위치를 정한다. 예를 들어 리듀스 태스크 개수가 2개이면, 각각의 레코드에 부여된 해시 코드를 2로 나누고 그 나머지가 0이면 리듀스 태스크 0으로, 나머지가 1이면 리듀스 태스크 1로 보내는 것이다. 구체적으로 <lion, [1]>이라는 레코드에 128이라는 해시 코드가 주어졌다고 하면, 128을 리듀스 태스크의 개수인 2로 나눈다. 그러면 나머지가 0이 나오므로 리듀스 태스크 0으로 보내는 식이다.

여기까지의 과정이 모두 끝나면 취합한 정보를 처리하는 '리듀싱'이 진행된다. 리듀싱은 병합된 레코드에서 [값 1, 값 2, …]의 형식으로 나열되어 있는 값들이 리듀서 함수에 의해 새로운 결괏값으로 처리되는 단계를 말한다. 단어별 사용 빈도수를 파악하는 작업의 경우 리듀싱 과정에서는 주어진 레코드 값들의 합이 결괏값으로 만들어진다. 예를 들어 <the, [1, 1]>은 리듀서 함수에서는 <the, 2>가 되는 것이다. 리듀싱의 단계가 끝나면 중앙 서버는 이렇게 처리된 데이터를 받아 최종 결과를 출력한다.

이와 같이 맵리듀스 작업은 여러 개의 서버에서 동시에 이루어진다. 이로써 맵리듀스는 데이터 처리 속도를 크게 높일 수 있게 되었다. 예를 들어 100TB의 데이터를 초당 100MB의 데이터를 처리하는 한 대의 컴퓨터에서 순차적으로 처리하면 약 12일 정도의 시간이 걸린다. 이것을 1,000대의 컴퓨터에서 분산하여 처리하면 약 17분 만에 같은 데이터를 처리할 수 있는 것이다. 맵리듀스의 구조와 처리 과정은 단순한 편이지만 데이터를 하나의 서버에 모아 순차적으로 처리하는 전통적인 방식의 데이터 처리 기술에 비해 ⓐ 데이터의 처리 시간을 크게 줄일 수 있다.

*서버: 주된 정보의 제공이나 작업을 수행하는 컴퓨터 시스템. 클라이언트 시스템이 요청한 작업이나 정보의 수행 결과를 돌려줌.
*리듀스 태스크: 리듀스 작업이 실행되는 단위.
*레코드: 데이터 처리 시 하나의 단위로 취급되는 관련 정보들의 묶음.

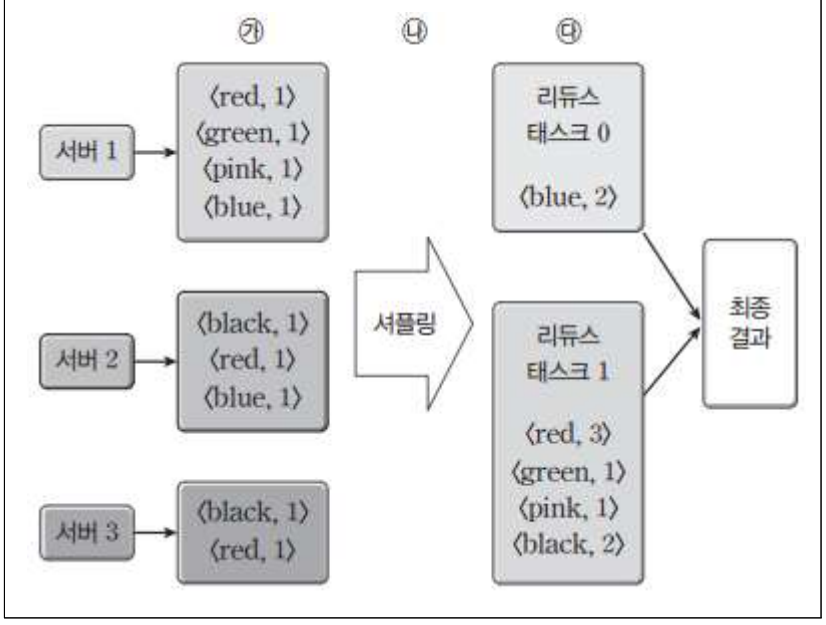
16. 윗글의 내용과 일치하는 것은?

① 셔플링을 거치고 나면 이전 단계보다 레코드의 수가 증가한다.
② 매핑을 거치고 나면 키의 종류에 따라 블록이 만들어져 분산된다.
③ 맵리듀스는 여러 대의 서버가 하나의 시스템처럼 작동하는 환경에서 주로 사용된다.
④ 맵리듀스 작업을 하려면 분산되어 있는 데이터를 병합하는 과정을 먼저 거쳐야 한다.
⑤ 레코드에 부여된 해시 코드를 시스템 내에 있는 레코드의 수로 나누어 리듀스 태스크의 위치를 결정한다.

17. 윗글을 바탕으로 <보기>에 대해 이해한 내용으로 적절하지 않은 것은?

<보 기>

다음은 한 검색 엔진에서 일정 기간 검색된 색채 관련어의 검색어별 빈도수를 구하는 맵리듀스의 과정을 간단하게 표현한 것이다.



① ㉯에서 셔플링 작업이 진행되면, <black, [1, 1]>이라는 새로운 레코드가 생성되겠군.

② ㉯에서 셔플링 작업이 진행되면, ㉮와 비교할 때 'red'의 키를 가진 레코드의 수가 가장 많이 줄어들겠군.

③ ㉰를 보면, ㉯에서 <blue, 2>에 부여된 해시 코드가 짝수임을 알 수 있겠군.

④ ㉰를 보면, 키가 'green'인 레코드와 키가 'pink'인 레코드에는 동일한 해시 코드가 부여되었음을 알 수 있겠군.

⑤ ㉰의 레코드에 있는 값을 보았을 때, 리듀싱 과정에서는 리듀서 함수에 의해 셔플링을 거친 레코드 값들의 합이 결괏값으로 처리되겠군.

18. ㉠과 ㉡에 대한 설명으로 적절하지 않은 것은?

① 맵리듀스 과정에서 ㉠은 ㉡보다 우선적으로 실행된다.

② ㉠은 분할된 하나의 단어당 하나의 레코드가 생성되도록 한다.

③ ㉡에서는 데이터의 처리 목적에 따라 다른 함수를 사용하게 된다.

④ ㉡을 위해 해시 코드를 부여하는 작업은 리듀스 태스크에서 병렬적으로 진행된다.

⑤ ㉠과 ㉡을 모두 거친 후의 레코드들은 모두 다른 키를 갖게 된다.

19. ⓐ의 주된 이유로 가장 적절한 것은?

① 여러 서버에서 데이터를 병렬적으로 처리할 수 있기 때문에

② 각각의 단어를 키로 사용하는 레코드를 만들어 사용하기 때문에

③ 데이터의 처리 목적에 따라 다양한 연산을 수행할 수 있기 때문에

④ 데이터에서 원하는 정보만을 추출해서 처리하는 과정을 거치기 때문에

⑤ 분산되어 있는 데이터를 중앙에 있는 서버로 모은 뒤에 레코드를 생성하기 때문에