Journal of the Korea Academia-Industrial cooperation Society
Vol. 19, No. 8 pp. 535-544, 2018

https://doi.org/10.5762/KAIS.2018.19.8.535
ISSN 1975-4701 / eISSN 2288-4688

맵리듀스 온라인 프레임워크에서 공간 데이터 스트림 처리를 위한 동적 부하 관리 기법

정원일
호서대학교 컴퓨터정보공학부

Dynamic Load Management Method for Spatial Data Stream Processing on MapReduce Online Frameworks

Weonil Jeong
Division of Computer and Information Engineering, Hose University

요 약 다양한 센서를 내장하고 고품질의 무선 네트워크 통신 기능을 탑재한 이동 장치의 보급이 확대됨에 따라 다양한 서비스 환경에서 이동 장치로부터 생성되는 시공간 데이터 량도 빠르게 증가하고 있다. 이와 같이 실시간 특성을 갖는 대량의 공간 데이터 스트림을 처리하기 위한 기존의 연구에서 하둡 기반의 공간 빅 데이터 시스템은 일괄 처리 방식의 플랫폼으로 공간 데이터 스트림에 대한 실시간 서비스에 적용하기에는 매우 어렵다. 이에 본 논문에서는 맵리듀스 온라인 프레임워크를 확장하여 연속적으로 입력되는 공간 데이터 스트림에 대한 실시간 질의 처리를 지원하고, 질의 처리 과정에서 야기될 수 있는 부하 문제를 효과적으로 분산하는 부하 관리 기법을 제안한다. 제안 기법에서는 공간 분할 영역을 기반으로 입력 데이터의 유입율과 부하율을 이용하여 노드들에 대해 동적으로 부하를 분산하는 기법을 제시하였다. 실험에서는 특정 공간 영역에서의 부하 관리가 요구될 때 해당 영역에서의 공간 데이터 스트림을 공유하는 자원들에게 분배함으로써 효과적인 질의 처리를 지원할 수 있음을 보인다.

Abstract As the spread of mobile devices equipped with various sensors and high-quality wireless network communications functionsexpands, the amount of spatio-temporal data generated from mobile devices in various service fields is rapidly increasing. In conventional research into processing a large amount of real-time spatio-temporal streams, it is very difficult to apply a Hadoop-based spatial big data system, designed to be a batch processing platform, to a real-time service for spatio-temporal data streams. This paper extends the MapReduce online framework to support real-time query processing for continuous-input, spatio-temporal data streams, and proposes a load management method to distribute overloads for efficient query processing. The proposed scheme shows a dynamic load balancing method for the nodes based on the inflow rate and the load factor of the input data based on the space partition. Experiments show that it is possible to support efficient query processing by distributing the spatial data stream in the corresponding area to the shared resources when load management in a specific area is required

Keywords : Spatial Big Data, Spatial Data Stream Processing, Load Management, Load Balance, MapReduce Online

1. 서론

무선 네트워크 통신 기술의 발달과 위치 정보 관리 기술의 확산, 그리고 스마트폰과 같은 이동 기기의 보급이 급속하게 증가하고 있다. 이에 시간 흐름에 따라 변화하는 이동 기기의 위치를 기반으로 하는 길 안내, 차량 추

*Corresponding Author : Weonil Jeong(Hoseo Univ.)
Tel: +82-41-540-5984 email: wnchung@hoseo.edu
Received April 13, 2018 Revised May 25, 2018
Accepted August 3, 2018 Published August 31, 2018

적, 주변 정보 조회 등의 생활 편의 서비스에서부터 환경, 교통, 생태, 재난 재해 등의 사회 및 공공 서비스, 온라인에서 사용자들의 다양한 취미나 관심 분야를 공유하는 네트워크 서비스인 소셜 네트워크 서비스에 이르는 서비스 영역도 빠르게 확산되고 있다[1-2]. 이러한 서비스 환경에서는 이동 기기가 지속적으로 생성하는 위치 정보와 연계된 소셜 데이터, 위치 로깅 등과 관련된 대량의 실시간 정보들이 기하급수적으로 증가함에 따라 대용량 시공간 데이터에 대한 효과적인 저장 및 분석을 지원할 수 있는 공간 빅 데이터 처리 연구들이 수행되어 있다[3-6].

기존의 공간 빅 데이터 시스템들은 대규모 클러스터 환경에서 작동하는 분산 응용을 지원하는 병렬처리 시스템인 하둡(Hadoop) 프레임워크[7]와 하둡 클러스터에서 분산 데이터 처리를 지원하는 맵리듀스(MapReduce)[8]를 이용하여 구현되고 있다. 대규모 공간 질의 처리를 지원하는 확장 가능한 고성능 공간 데이터웨어하우스 시스템인 Hadoop-GIS[3]는 맵리듀스를 기반으로 다양한 공간 질의를 병렬 처리하기 위해 공간 데이터 분할, 글로벌 분할 색인과 지역 공간 색인, 정확한 질의 처리 결과 도출을 위한 경계 객체 처리 방법을 제시하고 있다. SpatialHadoop[4]은 하둡 프레임워크를 확장하여 공간 데이터 및 연산을 위한 언어와 공간 데이터 처리를 위한 공간 색인을 지원하며 맵리듀스에서 공간 조인과 검색 기능을 제공하고 있다. 그리고 하둡 프레임워크 환경에서 효과적인 맵리듀스 프로그래밍 개발을 지원하기 위해 기존의 SQL에 OGC 심플 피처 모델에 따른 공간 데이터 표준 데이터 형식과 함수를 확장하여 공간 빅 데이터를 처리하는 시스템에 대한 연구[5]와 위치 정보를 기반으로 시계열 변화에 따라 축적되는 데이터들에 대한 활용 체계로 공간 빅 데이터에 대한 개념을 설명하고, 다양한 공간 분석을 통한 서비스 요구사항을 명세하는 연구들이 수행되었다[6]. 이와 같이 맵리듀스를 기반으로 하는 기존의 연구들에서는 HDFS(Hadoop File System)이나 데이터베이스로부터 입력된 데이터를 변환하여 처리하는 맵(Map) 단계와 맵 단계의 결과를 통합 처리하여 최종 결과를 생성하는 리듀스(Reduce) 단계가 반복적으로 수행될 수 있다. 이때 맵 단계와 리듀스 단계가 복합적으로 수행될 경우 각 단계별로 데이터의 저장 및 검색을 위한 반복적인 디스크 접근으로 인해 프로세스 성능 및 시스템 이용률이 저하되므로 일괄처리 방식의 맵리듀스 프레임워크에서는 연산 수행 결과의 실시간 응답성을 보장할 수 없다. 이에 실시간 응답성을 보장하도록 하둡의 맵리듀스를 메인 메모리 기반으로 구현하여 디스크 스트리밍을 배제함으로써 대량의 입력 데이터 스트림에 대해 이벤트 모니터링 및 데이터 스트림 처리, 온라인 집계 처리, 연속 질의 처리 등을 지원하는 연구[9]와 방대한 양의 센서 스트림 데이터에 대해 다차원 중복 질의 영역을 맵 단계에서 다중 킷값으로 변환하여 분할한 후 리듀스 단계에서 동일 킷값을 그룹화하여 연속적인 질의 처리를 지원하는 맵리듀스 기반의 다차원 연속 질의 처리 연구[10], 맵리듀스와 구글의 액터 모델의 결합된 기능을 응용한 고확장성 스트림 처리 시스템에서 서로 다른 스트림 사이의 데이터 편향 분포를 해결하고 다중 질의에 대한 중복 실행을 최소화할 수 있는 온라인 집계 알고리즘에 관한 연구[11]등이 수행되었다. 실시간 응답성을 보장하기 위한 기존의 맵리듀스 기반의 빅 데이터 플랫폼의 연구들은 고정된 크기의 파일이나 데이터 스트림을 지정된 크기로 분할하여 분산된 노드의 맵태스크들에게 균등한 양의 데이터가 할당토록 하고 있다. 그러나 실시간으로 유입되는 데이터 스트림의 크기와 빈도는 매우 유동적이어서 한정된 주기억 장치의 저장 공간을 초과하는 경우가 발생할 수 있고, 시간의 흐름에 따른 입력 데이터 스트림의 양을 예측할 수 없으므로 입력 데이터 스트림을 일정한 크기로 분할할 수 없어 입력 데이터 스트림을 임의의 맵태스크에 할당하는 경우 해당 맵태스크에서의 부하 증가를 야기하여 실시간 응답성을 저하시킬 수 있는 단점이 존재한다. 이에 맵리듀스 기반의 빅 데이터 플랫폼에서 높은 연산 효율과 낮은 응답 시간을 위해 Hoeffding tree 알고리즘을 기반으로 센서 데이터 스트림을 분류하고 처리하는 기술[12], 맵리듀스 온라인의 파이프라인 기능을 이용하여 부하 문제를 해결하기 위한 스트림 할당 및 분할 기법[13], 실행되는 인스턴스 간의 작업 부하를 계산하고 동적 작업 부하 변화에 따라 자원 사용을 조정하는 동적 부하 관리 기술[14]들에 대한 연구가 수행되었으나 현재까지 진행된 기존의 부하 관리 연구는 위치 데이터를 포함한 공간 정보의 특성을 고려하지 않기 때문에 공간 연산에 대한 연산 정확도와 성능을 감소시킬 수 있다.

따라서 본 논문에서는 위치 정보와 연계된 다양한 데이터들이 실시간 데이터 스트림으로 입력되는 맵리듀스 온라인[15]을 확장한 공간 빅 데이터 플랫폼에서 공간

데이터의 특성을 반영하여 공간 질의의 처리 성능을 향상시킬 수 있는 부하 관리 기법을 제안한다. 제안 기법에서는 쿼드 트리[19]를 이용하여 질의 대상 영역을 분할하고 쿼드 트리의 각 분할 영역은 대응하는 고유의 맵 태스크에 매핑한다. 공간 데이터 스트림이 입력되면 쿼드 트리에서 해당 스트림 데이터의 공간 정보를 분석하여 해당 데이터가 전송되어야 하는 맵태스크의 위치를 결정한다. 입력되는 공간 데이터 스트림이 쿼트 트리의 특정 분할 영역에 집중되어 임계점을 초과하여 부하가 발생하거나 쿼트 트리의 각 노드에 대응하는 특정 맵태스크에서 연산 처리 지연으로 부하가 감지될 경우 입력 데이터의 수량과 입력 빈도 변화, 질의 공간 영역에 대한 연산 선택도와 연산 처리 속도 등으로부터 부하 정도를 분석하여 부하를 야기한 공간 영역에 대한 쿼트 트리 노드를 재분할한다. 이때 쿼트 트리의 노드가 재분할되면 분할된 개별 쿼드 트리에 대응하는 새로운 노드를 추가하고 처리 데이터를 재분배하여 부하를 감소시킨다.

본 논문의 구성은 다음과 같다. 2장에서는 빅데이터 분산 처리 플랫폼 맵리듀스와 실시간 빅데이터 처리를 지원하는 맵리듀스 온라인, 그리고 빅데이터 플랫폼에서의 부하관리 기법에 대해 기술하고, 3장에서는 공간 빅데이터 플랫폼에서의 제안 부하 관리 기법에 대해 설명한다. 4장에서는 제안 기법에 대한 다양한 실험 결과를 보이며, 5장에서 결론을 맺고 향후 연구 방향에 대해 기술한다.

2. 관련연구

2.1 맵리듀스

맵리듀스는 빅 데이터 처리를 위한 분산 컴퓨팅 플랫폼으로 맵 단계과 리듀스 단계로 처리된다. 맵 단계에서는 HDFS의 파일이나 데이터베이스 레코드 등으로부터 키와 값을 입력으로 처리한 결과 키와 값을 저장 출력한다. 모든 맵 처리 단계가 종료되면 하둡의 맵 출력들로부터 합병 정렬된 결과가 리듀스 단계의 입력으로 전달되며, 리듀스 단계에서는 입력 키와 값의 리스트로부터 그 처리 결과를 HDFS, 데이터베이스, 파일시스템 등에 저장한다[7,8].

2.2 맵리듀스 온라인

맵리듀스 온라인은 맵리듀스 프레임워크의 프로그래밍 인터페이스와 결함 허용 모델을 유지하면서 연산자들 사이에 파이프라인 형태로 데이터를 처리하는 구조이다. 매퍼가 생성한 데이터는 곧바로 리듀서로 전달되어 처리되므로 실시간 온라인 집계 기능을 제공할 수 있으며, 파이프라이닝 구조는 연속 질의 처리가 요구되는 이벤트 모니터링이나 스트림 처리와 같은 응용을 지원할 뿐 아니라 스트림 연산자들에 대한 신속한 병렬 처리를 통해 시스템 이용률과 질의응답 시간을 단축시킬 수 있다[15].

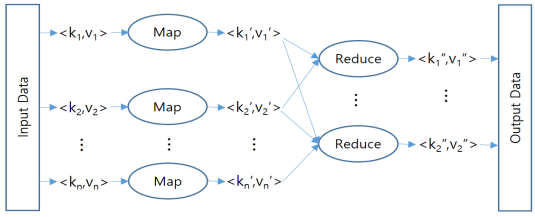

Fig. 1. MapReduce Online Processing Model

맵리듀스 온라인 프레임워크에서의 데이터 처리 흐름은 Fig. 1에서 나타낸 바와 같이 파이프라이닝 기법을 이용하여 맵 단계의 처리 결과를 실시간으로 리듀스 단계로 전송하여, 기존의 맵리듀스 구조에 비해 리듀스 단계의 수행 속도를 향상시킬 수 있고 전체 노드의 이용률을 높일 수 있다[9,10,15]. 본 연구에서는 맵리듀스 온라인을 활용하여 공간 데이터 스트림을 처리하는 확장 구조를 기반으로 한다.

2.3 부하관리 기법

공간 빅 데이터 플랫폼에서 효과적으로 부하를 관리하기 위해서는 실시간으로 입력되는 공간 데이터 스트림의 유동적인 데이터 크기나 입력 빈도의 변화 및 공간 정보로서의 특성 등이 고려되어야 한다.

맵 리듀스 환경에서 Hoeffding 색인 알고리즘을 이용한 데이터 스트림 처리 기법은 Hoeffding 경계에 따라 노드를 분할하고, Hoeffding 경계를 만족하는 최소 개수의 튜플을 선택할 수 있는 자원 스케쥴링 방법을 제안하였다[12]. 맵리듀스를 기반으로 스트림 데이터 처리를 위한 스트림 할당 및 분할 기법은 맵태스크에서 스트림 데이터의 수신 속도, 큐의 이용률, 데이터 처리 속도 등의 부하 요소를 분석하여 특정 맵태스크에서의 부하 여

부에 따라 해당 맵태스크의 데이터 스트림의 일부를 부하가 적은 다른 노드로 분할 전송하거나 분할 전송받은 다른 맵태스크의 상태에 따라 병합하는 기법을 제시하고 있다[13]. 맵리듀스 스타일의 플랫폼 기반의 동적 자원 관리 기법에서는 데이터 처리에 따른 지연을 줄이기 위해 데이터 전송 수를 최소화하고 실행 작업에 대한 가용성을 보장하는 방법을 제시하고 있다[14]. 그러나 맵리듀스 플랫폼에서 실시간 응답성을 개선하기 위한 이러한 연구들에서는 공간 데이터 스트림에 대한 공간적 특성을 고려하지 않고 있다.

공간 데이터 특성을 고려한 스트림 데이터 부하 관리에 대한 연구로 그리드 해쉬를 이용하여 공간 스트림 윈도우 집계질의의 정확도를 향상하는 연구[16]에서는 공간 질의의 공간적 이용도에 차등적인 샘플링을 통해 질의에 이용될 확률이 낮은 데이터를 제한하는 부하 관리 방법을 제시하고 있으나, 이 기법은 부하 제한을 위해 그리드 셀의 가중치만을 이용함으로써 공간 데이터 스트림의 유입 변화와 처리율을 고려하지 않고 있다. 공간 질의 정확도 향상을 위한 선-필터링을 이용한 후-부하제한 기법[17]에서는 스트림 큐에 불필요하게 가중되는 부하를 선제적으로 제한하고, 부하가 발생할 경우 공간 중요도와 데이터 중요도를 분석하여 추가 부하 제한을 수행한다. 그러나 이 기법은 선제적 부하 제한을 위해 스트림 데이터의 유입 변화나 데이터의 이용률을 반영하지 않아 부하 제한의 효과와 질의 처리 정확도를 보장할 수 없다는 문제가 있다. 공간 데이터 스트림의 입력 빈도와 데이터 밀집도를 이용하여 동적으로 부하를 제한하는 기법[18]은 질의 영역을 공간 분할하여 그리드 구조로 구성하고, 입력되는 공간 데이터 스트림의 유입 빈도의 변화율과 공간 연산 선택도를 분석하여 해당 질의 공간에서의 부하를 산출하며, 질의 공간에 대한 부하율에 따라 동적으로 부하 제한의 정도를 결정한다. 그러나 이 기법에서는 실제 공간 데이터 스트림 처리 환경에서 발생할 수 있는 중복 또는 무의미한 데이터에 대한 입력을 차단할 수 없고, 공간 영역에 대한 부하 정도를 산출하기 위한 연산 처리 비용에 대해 고려하지 않고 있다.

3. 제안 부하관리 기법

이 장에서는 맵리듀스 온라인을 확장한 공간 빅 데이터 플랫폼에서 쿼드 트리를 이용하여 전체 질의 공간 영역을 분할하여 각 공간 영역에 대응하는 맵태스크에서의 부하 정도를 분석하여 부하를 관리하는 방법에 대해 기술한다.

쿼드 트리는 d 차원의 노드를 2^d개의 하위 공간으로 분할하여 자식 노드들을 표현하는 색인으로 대용량 공간 데이터에 대한 빠른 검색을 제공하므로 공간 데이터 및 이미지 처리 분야에 널리 활용되고 있다. 제안 기법에서 쿼드 트리의 한 노드는 입력 공간 데이터 스트림이 맵 단계로 전달되어야 하는 사전 정의된 공간 영역을 의미한다.

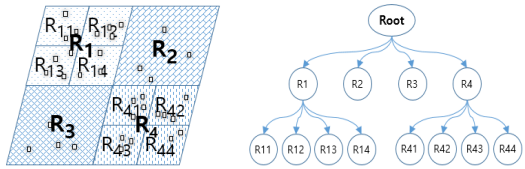

Fig. 2. An example of Quad tree
(a)Space partitioning (b)Quad tree structure

Fig. 2는 제안 기법에서 쿼드 트리를 이용하여 전체 질의 공간을 분할한 예시로, 2차원 공간에 대해 공간 분할 및 쿼드 트리 구조를 나타낸다.

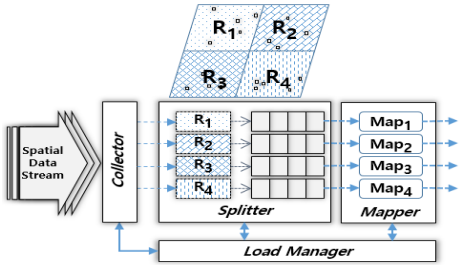

Fig. 3. Run-time architecture for spatial big-data stream processing

다양한 데이터 소스로부터 유입되는 공간 빅 데이터를 처리하기 위한 런타임 구조는 Fig. 3과 같다. 콜렉터(Collector) 모듈은 위치 정보를 갖는 센서 데이터, 소셜 데이터 등 다양한 형태의 데이터 소스로부터 공간 데이터 스트림을 수집한다. 이때 콜렉터 모듈은 부하 관리기(Load Manager)로부터 사전 분할된 쿼드 트리상의 분할 영역 정보를 확인하여 입력 데이터를 공간 영역에 대응하는 분배기(Splitter)의 스트림 큐로 전달한다. 매퍼(Mapper)는 개별 맵 단계를 수행하기 위해 분배기의 대

응하는 스트림 큐에 누적된 데이터를 가져와서 처리하며, 부하 관리기(Load Manager)는 데이터 전달 과정에서 발생하는 부하를 감지하고 필요시 분산하는 역할을 수행한다.

이러한 공간 빅 데이터 스트림 처리 과정에서의 부하는 콜렉터에서 분배기로 전달할 때 특정 영역에 입력 공간 데이터가 집중되거나 매퍼의 맵 단계에서 입력된 공간 데이터가 질의 처리 지연으로 인해 이용되지 않은 상태로 계속 유지되는 경우로 구분한다.

콜렉터에서 분배기의 스트림 큐로 전송되는 데이터의 부하를 측정하기 위해 분배기의 개별 스트림 큐에 입력되는 입력 데이터의 변화, 입력 데이터 양, 입력 데이터 크기 변화에 따라 데이터 입력율을 산출하고, 그 결과로부터 해당 스트림 큐에 대한 부하 관리 여부를 판단할 수 있는 부하수치를 계산한다.

분배기에서 임의의 스트림 큐로 유입되는 데이터에 대한 입력율(I_i)은 수식 (1)과 같이 산출된다.

$$I_i = \lceil (F_v(D_i) + D_i^c) \times A(D_i) \div C_i \rceil \quad (1)$$

수식 (1)에서 I_i는 콜렉터에서 분배기의 스트림 큐로 데이터 소스 D_i에 대한 데이터 입력 빈도의 변화율 $F_v(D_i)$, 최근 데이터 스트림의 입력 양 D_i^c, D_i로부터 분배기의 스트림 큐로 유입되는 데이터 크기 $A(D_i)$, 입력 데이터의 변화량이나 크기를 산정하기 위해 데이터 소스에 설정되는 임의값 C_i으로부터 계산된다. 위 수식에서 $F_v(D_i)$는 과거 입력 데이터 스트림의 변화량에 대한 평균값으로부터 계산된 예측 보정값과 직전 데이터 스트림의 입력 빈도와 현재 입력 빈도에 대한 절대치를 합산하여 유도한 값이다.

데이터 소스 D_i에 대해 산출된 유입율 I_i로부터 분배기 내 해당 분할영역에서 부하여부를 판단하기 위한 부하 수치 LF_i을 수식 (2)를 통해 결정한다.

$$LF_i = \left\lceil I_i / \sum_{k=1}^{n} I_k \right\rceil \quad (2)$$

분배기의 특정 분할 영역에 유입되는 데이터의 빈도에 부하수치 LF_i는 비례한다. 분배기의 임의 영역에 부하 수치가 증가하여 해당 분할 영역을 분할해야 할 경우 기준 임계값이 설정되어야 하며, 이러한 분할 기준 임계값은 수식 (3)을 통해 산출된다.

$$PF_i = \alpha(M_i + LF_i) \quad (3)$$

수식 (3)에서 α는 임의의 상수 값이고, M_i는 데이터 소스 D_i로부터 갱신 주기 동안 스트림 큐로 유입되는 데이터의 값으로부터 유도된 평균값이다.

쿼드 트리를 통해 분할한 공간 영역에 대응하는 분배기 스트림 큐의 부하 정도를 판단하기 위해 사용자의 요구에 따라 임의로 스트림 큐 별로 유입주기를 설정할 수 있다.

Fig. 3에서 쿼드 트리상의 R_1 영역에 대응하는 분배기의 R_1 스트림 큐에 데이터 유입이 급증할 경우 산출된 PF_1으로부터 R_1 영역의 분할에 대한 부하 분산 여부를 판단할 수 있다.

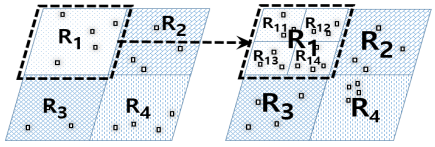

Fig. 4. Space partitioning of spatial region R1 on quad tree for load balancing

Fig. 4는 R_1 영역에 대한 부하 증가로 인해 산출된 PF_1가 부하 기준을 초과하여 부하 분산이 요구되면 쿼드 트리 상에서 R_1 영역을 $(R_{11}, R_{12}, R_{13}, R_{14})$로 격자 분할을 수행 결과를 나타낸다. 쿼드 트리에서 분할이 요구되면 분배기 내에서는 신규 분할된 영역에 대응하도록 스트림 큐가 재구성되어야 한다.

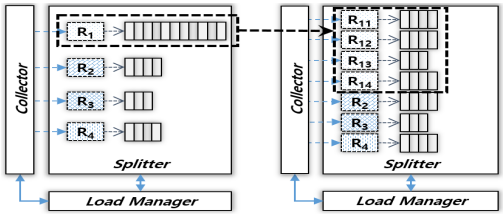

Fig. 5. Stream queues rebuilding on splitter

Fig. 5는 R_1 영역 분할 요구에 따라 쿼드 트리의 분할에 이어 분배기 내에서 R_1 영역에 대응하는 스트림 큐를 재구성하여 확장하여 스트림 데이터를 분산하여 유지하는 방법을 나타낸다.

제안 시스템에서는 쿼드 트리로 구성된 공간 분할 영역에 대응하는 공간 데이터 스트림이 맵 단계의 내부 버

퍼 큐에 전달되어 맵 함수를 통해 공간 객체에 대한 연산을 포함한 질의를 지원한다. Fig. 6은 쿼드 트리로 구성된 공간 분할 영역에 존재하는 공간 데이터 스트림 관리 구조와 질의의 대상이 되는 공간 영역의 상관 관계를 나타낸다.

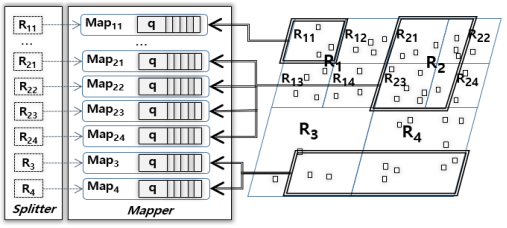

Fig. 6. Correlated information between spatial query region and stream queues

Fig. 6에서 질의 공간 영역에는 다수의 쿼트 트리 분할 영역이 포함될 수 있다. 맵 단계의 버퍼 큐에서는 큐 식별자, 레코드 수, 가중치 정보를 관리하며, 각 버퍼 큐 단위로 정의된 시간 구간 동안에 입력되는 공간 데이터 스트림에 대한 유입 빈도, 공간 연산 선택도 및 공간 연산 처리 시간으로부터 해당 질의 공간 영역에서의 부하 정도를 산정한다.

버퍼 큐 q_i의 유입 빈도는 주기 별로 q_i의 유입 빈도 변화율과 실제 유입 빈도를 이용하여 계산되며, 산출 수식은 (4)와 같다.

$$F(q_i) = [Avg(F^{p's}(q_i)) + (|q_i^p - q_i^c|)] + q_i^c \qquad (4)$$

q_i의 유입 빈도 변화율은 과거 입력 빈도 변화의 평균값 $Avg(F^{p's}(q_i))$와 직전 유입 빈도와 현재 유입 빈도의 절대 차이 $(|q_i^p - q_i^c|)$을 합산하여 유도한다. q_i^c는 현재 유입량을 나타내며, 유입 빈도 변화율과 합산하여 q_i의 유입 빈도가 산출된다.

버퍼 큐의 공간 데이터 스트림은 공간 연산이나 조인 연산의 입력으로 처리되며, 질의 공간 영역에 대한 연산 선택도는 수식 (5)를 이용한다.

$$S(O_i) = [\frac{1}{n}\sum_{1}^{n} S^{p'}(O_i) + |S^p(O_i) - S^c(O_i)|] + S^c(O_i) \qquad (5)$$

수식 (5)에서 연산자 O_i에 대한 선택도 $S(O_i)$는 변화율 보정을 위한 과거 선택도 $S^{p'}(O_i)$의 평균값과 직전 선택도 $S^p(O_i)$와 현재 선택도 $S^c(O_i)$ 차이의 합에 현재 선택도 $S^c(O_i)$를 합산하여 유도된다.

공간 데이터 스트림에 대한 연산 처리 소요 시간은 설정 주기에 따른 과거 데이터의 처리 시간을 이용하며, 수식 (6)에 따라 계산된다.

$$T(q_i) = \frac{1}{n}\sum_{1}^{n}(T^p(q_i)) \qquad (6)$$

수식 (6)에서 q_i에 대한 연산 처리 소요 시간 $T(q_i)$는 q_i에서의 과거 연산 처리 소요 시간 $T^p(q_i)$들의 평균값으로 산출한다.

질의 공간 영역에서의 부하 정도는 q_i에 대해 유도된 유입 빈도, 연산 선택도, 연산 처리 소요 시간으로부터 수식 (7)으로 산출된다.

$$LFq_i = \sum_{i=1}^{n} F(q_i) \cdot S(O_i) \cdot T(q_i) \qquad (7)$$

수식 (7)에서 질의 공간 영역의 연산자 수가 n일 때, 해당 공간의 부하 정도는 연산자 선택도 $S(O_i)$와 스트림 유입 빈도 $F(q_i)$의 누계 그리고 연산 처리 시간 $T(q_i)$의 곱으로 계산된다.

부하 수치 LFq_i는 임의 변경 가능한 자연수 k를 증감 단위로 상한값 u이하로 설정된다. Fig. 6에서 공간 객체가 다수의 쿼드 트리 분할 영역에 겹칠 수 있다. 이러한 공간 객체가 쿼드 트리의 분할 영역의 내부에 존재하거나 경계에 걸치는 수를 On_i, On_i 값들 가운데 최대값이 M이라면, $uk < M$일 때 k는 M/u로 결정된다. 이로부터 쿼드 트리의 분할 영역에서의 부하 수치 LFq_i는 $\lceil On_i/k \rceil$로부터 계산한다. LFq_i가 낮을수록 질의 처리 공간 영역에 존재하는 객체의 연산 참여가 낮다는 것을 의미한다. 이에 버퍼 큐에서는 유입 빈도 상승에 비례하여 부하 분산의 요인도 커지게 된다. 제안 기법에서는 질의 처리 지연 등으로 인해 특정 버퍼 큐의 공간 데이터 스트림이 소비되지 못하고 적체되어 부하 상황으로 판단되면 부하를 야기한 버퍼 큐의 공간 데이터 스트림을 질의 공간 영역을 공유하는 다른 버퍼 큐로 재분배하여 부하를 분산시킨다.

Fig. 7에서는 맵 단계에서 Map_{23}의 버퍼 큐에 부하가 발생한 경우 해당 맵의 공간 데이터 스트림을 질의 공간 영역을 공유하는 Map_{21}, Map_{22}, Map_{24}로 분산하여 처리한다.

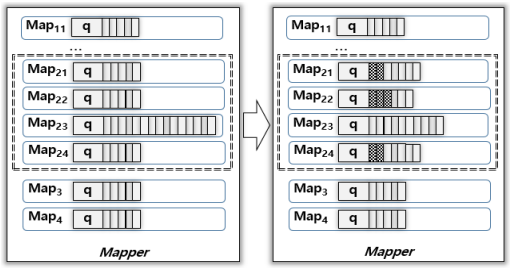

Fig. 7. Stream Data distribution adjustment on buffer queues of Mapper

4. 성능 평가

4.1 실험 환경

본 실험은 CentOS 6.7 64비트 운영체제, 16GB 메모리, Intel Xeon 3.5Ghz CPU, 4TB 디스크 시스템 1대의 마스터 노드와 CentOS 6.7 64비트 운영체제, 8GB 메모리, Intel Core i5 2.7Ghz CPU, 2TB 디스크 시스템 7대의 슬레이브 노드를 하나의 기가비트 네트워크로 연결하였으며, Hadoop 2.6.0과 HOP(Hadoop Online Prototype) 0.2를 설치하여 실험 환경을 구축하였다. 실험에서 입력 데이터는 식별자, 이차원 좌표, 생성 시간을 구조로 하는 16바이트 크기의 이동 객체로 주어진 공간 범위 내에서 설정된 단위 시간에 따라 생성한다. 쿼드 트리 노드에 대응하는 분배기의 스트림 큐와 맵 단계의 버퍼 큐의 최대 허용 용량은 8MB로 제한하였다. 실험에서 사용된 공간 영역의 경위도 좌표는 서울시 일원의 (37.611564, 126.935947), (37.611564, 127.071903), (37.548440, 126.935947), (37.548440, 127.071903)이며, 공간 데이터 스트림 데이터는 제시된 공간영역 내의 이차원 공간 좌표로 설정하였다.

4.2 성능 평가

공간 빅데이터 환경에서 실시간 데이터 스트림 처리를 지원하는 직접 비교 가능한 부하 관리 기법이 미제시되고 있어, 본 논문에서의 성능 평가는 제안 기법을 바탕으로 입력되는 공간 데이터 수, 공간 데이터 분포, 공간 질의와 비공간 질의를 달리하여 다양한 실험 환경을 구성하고 성능을 분석하였다.

공간 데이터 스트림의 입력 수의 변화에 따른 분배기의 스트림 큐에서의 데이터 누적량 실험에서는 초당 입력되는 데이터 수를 증가시키면서 스트림 큐에서의 누적 데이터 수를 관찰한다. 이때 스트림 큐에서 부하 분산이 수행되는 시점은 큐 용량의 90%로 설정하였다.

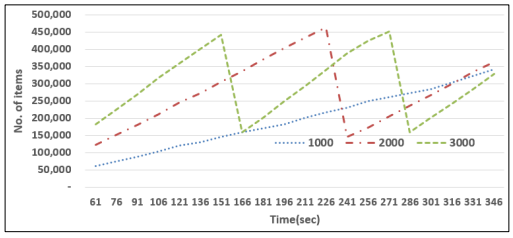

Fig. 8. Load distribution of number of tuples in stream queues of splitter according to time flow

이 실험은 Fig. 4에서 설명한 바와 같이 특정 공간 영역에 대한 입력이 급격히 증가하여 부하가 감지되면 쿼드 트리에서 공간 영역 분할을 수행하고 스트림 큐를 추가 구성하여 부하를 분산하는 내용이다. Fig. 8에서 분배기의 스트림 큐에 유입되는 데이터가 초당 1000개인 경우 완만한 상승 곡선을 보이고 있으며, 부하 관리가 요구되는 시점인 누적량 기준으로 450,000개에는 이르지 않는다. 초당 입력 수가 2000개인 경우와 3000개인 경우에는 각각 226초 구간, 151초 구간 및 271초 구간에서 부하가 감지되어 부하 분산을 수행함으로써 스트림 큐의

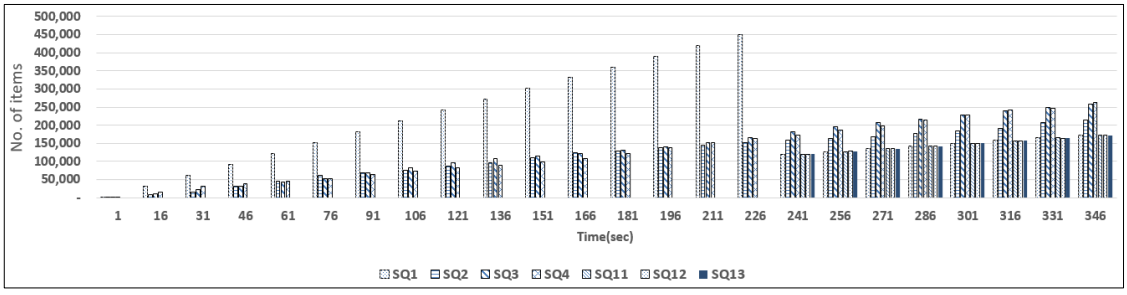

Fig. 9. Accumulated number of tuples and load sharing in stream queues of splitter according to time flow

누적량이 재조정되는 것을 확인할 수 있다.

분배기에서 부하 분산이 수행되면 해당 공간 영역은 4분할되며 각 공간 영역에 대응하는 스트림 큐가 재구성되어 부하를 공유하게 된다. 이에 대해 부하 분산에 따라 각 스트림 큐의 누적된 데이터량이 조정되는 과정을 이어서 실험하였다. 실험에서는 초기 공간 영역을 4분할하고 각기 대응하는 스트림 큐 SQ1 ~ SQ4를 대상으로 SQ1에 초당 3000개, SQ2 ~ SQ4에는 합산하여 초당 3000개의 임의 데이터를 비균일하게 생성하여 입력하였다. 이때 스트림 큐에서 매퍼의 버퍼 큐로 전달되어 소비되는 데이터는 SQ1의 경우 초당 1000개, SQ2 ~ SQ4의 경우 합산하여 초당 1000개를 설정하여 질의 처리 속도가 일정하게 유지되는 상황을 반영하였다.

Fig. 9에서 각 스트림 큐에는 지속적으로 데이터 입력으로 선형적인 데이터 누적 수치를 확인할 수 있으며, SQ1의 누적 데이터량이 최대 적재 용량의 90% 지점에 이르러 부하 지점이 되는 226초 구간에 이르기까지 SQ11 ~ SQ13은 예비 자원으로 대기 상태에 머문다. 부하 발생 지점을 지나면 부하 상태인 SQ1의 누적 데이터를 4분할하여 SQ11 ~ SQ13와 공유 분산 처리함으로써 SQ1의 부하가 해소되는 것을 알 수 있다. 이후 지속적인 스트림 큐에 대한 입출력으로 누적량이 상승하는 흐름을 보인다.

분배기의 스트림 큐로부터 매퍼의 버퍼 큐로 전달된 데이터는 공간 질의 처리를 위한 입력으로 활용된다. 공간 영역 질의 범위의 변화에 따른 질의 처리 성능에 대한 실험에서는 질의의 대상이 되는 공간 영역에 대해 쿼드 트리의 적용 유무 및 쿼드 트리를 적용할 때 입력 데이터 분포가 균일한 경우와 비균일한 경우로 구분하여 수행하였다. 이때 공간 질의 처리를 위해 입력되는 데이터는 초당 2000개로 설정하였다.

Fig. 10에서 공간 영역 질의(spatial range query)의 대상이 되는 공간 영역(spatial region)의 크기 증가에 따라 질의 처리 성능 결과를 쿼드 트리를 사용하지 않고 경우를 기준으로 입력 데이터가 균일 분포와 비균일 분포로 수행한 결과를 상대적인 수치로 표현하였다. 실험 결과에서 쿼드 트리를 적용하여 공간 영역 질의를 처리한 질의 처리 성능이 쿼드 트리를 적용하지 않은 경우에 비해 공간 질의 영역의 크기가 증가함에 따라 질의 처리 성능이 완만하게 근접하는 결과를 보이나, 전체적으로 비균일 분포의 경우 19.9%, 균일 분포의 경우 23.7% 차이로 성능 우위를 보였다. 이는 과도하게 질의 공간 영역에 확대되는 경우 쿼드 트리를 이용한 질의 처리의 장점이 감소하기 때문이다.

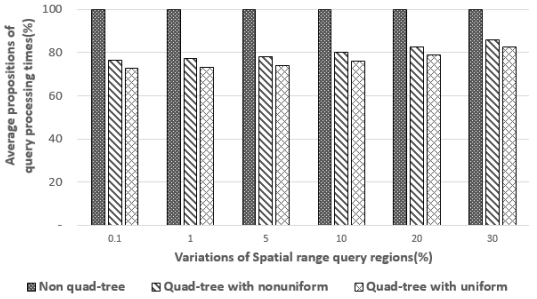

Fig. 10. Query processing time by increasing spatial range query region

질의 처리에 따른 부하 정도는 매퍼의 버퍼 큐에 유입되는 공간 데이터 스트림의 빈도와 연산 처리 시간 및 선택도로 판단한다. 이 실험은 공간 영역 질의 처리 과정에서 발생한 부하를 감지하고 이를 분산하기 위해 부하를 야기한 매퍼의 버퍼 큐 데이터를 해당 질의의 공간 영역을 공유하는 다른 버퍼 큐들을 대상으로 재조정하였다. 실험에서는 Fig. 7과 같이 질의 영역을 공유하는 4개 매퍼의 버퍼 큐들인 MQ21 ~ MQ24를 구성하고, 초당 평균 3974개의 입력 가운데 MQ3에는 초당 50%, 나머지 버퍼 큐들에는 초당 50%를 균등 분할하여 입력되었다.

Fig. 11에서 시간 흐름에 따라 부하 수준을 나타내는 그래프에서 버퍼 큐에 입력되는 전체 데이터 량은 일정한 수준으로 유지되고 있다. 질의 처리 수준은 200초 구간을 지나면서 급격하게 처리율이 떨어지기 시작하고, 230초 구간에서 부하 발생 시점으로 판단하여 이후 부하 분산을 시행한다. 동일 시간 흐름에서 MQ21 ~ MQ34에서 누적되는 데이터 량을 살펴보면, MQ23의 누적량은 지속적으로 증가하며 나머지 버퍼 큐들은 입력이 질의 처리로 연계되어 누적량이 미비한 수준을 보인다. 230초 구간을 지나면서 MQ3의 누적 데이터들은 다른 버퍼 큐들과 공유하여 부하 분산이 시작되고 235초 구간을 지나면서 부하 분산이 완료되며, 이는 부하 정도를 나타내는 그래프와 일치하는 결과를 보이고 있다. 이러한 결과는 입력 빈도가 일정하고, 공간 영역 질의에 주어진 연산에 대한 선택도는 여과(filtering) 단계를 통해 질의 결과로의 포함 여부를 판단할 수 있으므로 그 변화가 일정하므로, 이 실험에서 부하를 야기하는 가장 큰 요

인은 버퍼 큐에 유입되는 유지되는 데이터의 량에서 기인한 것으로 분석된다.

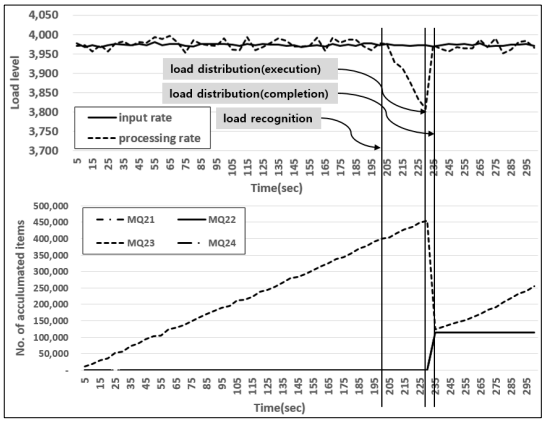

Fig. 11. Load recognition and distribution with other buffer queues sharing the same spatial query region

5. 결론

본 논문에서는 공간 빅 데이터 플랫폼에서 위치 정보와 연계된 대량의 실시간 데이터 스트림의 처리를 지원하기 위해 맵리듀스 온라인을 확장하여 공간 데이터의 특성을 반영하여 플랫폼에서 공간 질의 처리 성능의 수준을 보장할 수 있는 동적 부하 관리 기법을 제안하였다. 제안 기법에서는 쿼드 트리를 이용하여 공간 영역을 분할하고 플랫폼에 유입되는 공간 스트림 데이터를 해당 공간 분할 영역에 해당하는 분배기의 스트림 큐에서 유지하고, 특정 공간 영역에 데이터 유입 빈도가 급증할 경우 쿼드 트리를 분할하고 대응하는 분배기의 스트림 큐를 추가로 구성하여 부하를 분산하여 처리하였다. 분배기의 스트림 큐의 데이터는 이후 매퍼의 버퍼 큐로 전달되어 질의 처리에 활용되며 질의 처리 지연이나 입력 빈도의 증가로 인해 처리율이 떨어져 부하가 발생하면 질의 공간 영역을 공유하는 버퍼 큐들로 부하를 분산하였다. 이를 통해 공간 빅 데이터 플랫폼에서 공간 데이터 스트림 처리로 인해 야기될 수 있는 부하 문제를 해소할 수 있음을 실험을 통해 설명하였다. 향후 연구로는 공간 연산을 포함하여 비공간 연산이 복합적으로 사용되는 질의에서 연산의 선택도를 산출하는 방법을 통해 부하 분산에 대한 최적화 연구가 필요하다.

References

[1] J. Abdul, M. Alkathiri and M. B. Potdar, “Geospatial Hadoop (GS-Hadoop) an efficient mapreduce based engine for distributed processing of shapefiles”, *Advances in Computing, Communication, & Automation (ICACCA) (Fall), International Conference*, pp. 1-7, 2016.
DOI: https://doi.org/10.1109/ICACCAF.2016.7748956

[2] J. M. Park, M. H. Lee, D. B. Shin and J. W. Ahn, “Deduction of the Policy Issues for Activating the Geo-Spatial Big Data Services”, *Journal of Korea Spatial Information Society,* vol. 23, no. 6, pp. 19-29, 2015.
DOI: https://doi.org/10.12672/ksis.2015.23.6.019

[3] A. Aji, H. Vo, W. Fusheng, R. Lee, X. Zhang and J. Saltz, “Hadoop-GIS: A High Performance Spatial Data Warehousing System over MapReduce”, *VLDB Endowment*, vol. 6, no. 11, pp. 1009-1020, 2013.
DOI: https://doi.org/10.14778/2536222.2536227

[4] A. Eldawy, Alarabi, and M. F. Mokbel, “SpatialHaddop: A MapReduce Framework for Spatial Data”, *Data Engineering (ICDE), 2015 IEEE 31st International Conference on 2015*, pp. 1352-1363, 2015.
DOI: https://doi.org/10.1109/ICDE.2015.7113382

[5] In-Hak Joo, “Spatial Big Data Query Processing System Supporting SQL-based Query Language in Hadoop”, *Journal of Korea institute of information,* electronics, and communication technology vol. 10, no. 1, pp. 1-8, 2017.

[6] G. H. Kim, J. H. Yoon, C. M. Jun and H. C. Jung, “Providing Service Model Based on Concept and Requirements of Spatial Big Data”, *Journal of the Korean Society for Geospatial Information Science. vol. 24,* no. 4, pp. 89-96, 2016.
DOI: https://doi.org/10.7319/kogsis.2016.24.4.089

[7] Apache Hadoop, http://hadoop.apache.org/

[8] J. Dean, S. Ghemawat, “MapReduce: Simplified Data Processing on Large Clusters”, *Proc. of the 6th Symposium on Operating Systems Design and Implementation*, pp. 137-150, 2004.

[9] A. M. Aly, A. Sallam, B. M. Gnanasekaran, L. Nguyen-Dinh, W. G. Aref, M. Ouzzani and A. Ghafoor, “M3: Stream Processing on Main-Memory MapReduce”, *Data Engineering (ICDE), IEEE 28th International Conference*, pp. 1253-1256, 2012.
DOI: https://doi.org/10.1109/ICDE.2012.120

[10] D. Jeong, S. Jeon and B. Hong, “A Study on MapReduce Processing for Multi-dimensional Continuous Query”, *Lecture notes in computer science*, vol. 7827, pp. 74-78, 2013.
DOI: https://doi.org/10.1007/978-3-642-40270-8_6

[11] D. Yang, J. Cao, S. Wu and J. Wang, “Progressive online aggregation in a distributed stream system”, *The Journal of systems and software,* vol. 102, pp. 146-157, 2015.
DOI: https://doi.org/10.1007/978-3-642-40270-8_6

[12] X. Song, J. Gao, J. Ma, S. Niu and H. He, “HTME: A data streams processing strategy based on Hoeffding tree in MapReduce environment”, *Intelligent Control and*

Automation(WCICA), pp. 1042-1045, 2016.

[13] S. Park, W. Ryu, B. Hong and J Kwon, "MapReduce-based Stream Assigning and Splitting Technique for Stream Data Processing", *Journal of KIISE,* vol. 19, no. 8, pp. 439-443, 2013.

[14] K. Madsen and Y. Zhou, "Dynamic resource management in a MapReduce-style platform for fast data processing", *Data Engineering Workshops(ICDEW), 31st IEEE International Conference*, pp. 10-13, 2015.
DOI: https://doi.org/10.1109/ICDEW.2015.7129537

[15] T. Condie, N. Conway, P. ALvaro and J.M. Hellerstein, "MapReduce Online", NSDI'10, 2010.

[16] S. Baek, D. Lee, G. Kim, W. Chung and H. Bae, "Load Shedding Method based on Grid Hash to Improve Accuracy of Spatial Sliding Window Aggregate Queries", *Journal of KSIS,* vol. 11, no. 2, pp. 89-98, 2009.

[17] H. Kim, S. Baek, D. Lee, G. Kim, H. Bae, "Pre-filtering based Post-Load Shedding Method for Improving Spatial Query Accuracy in GeoSensor Environment", *Journal of KSISS,* vol. 12, no. 1, pp. 18-27, 2010.

[18] W. Jeong, "Dynamic Load Shedding Scheme based on Input Rate of Spatial Data Stream and Data Density", *Journal of KAIS,* vol. 16, no. 3, pp. 2158-2164, 2015.
DOI: https://doi.org/10.5762/KAIS.2015.16.3.2158

[19] R. A. Finkel and J. L. Bentley, "Quad trees a data structure for retrieval on composite keys", Acta informatica, vol. 4, no. 1, pp. 1-9, 1974.
DOI: https://doi.org/10.1007/BF00288933

정 원 일(Weonil Jeong) **[정회원]**

- 2004년 8월 : 인하대학교 일반대학원 컴퓨터정보공학과(공학박사)
- 2004년 7월 ~ 2006년 7월 : 한국전자통신연구원 선임연구원
- 2013년 1월 ~ 2014년 2월 : Univ. of Ohio Research Scholar
- 2007년 3월 ~ 현재 : 호서대학교 컴퓨터정보공학부 부교수

<관심분야>
공간빅데이터, 공간데이터스트림, 시스템보안, 공학교육