실시간 교통 이벤트 관리 및 클러스터링 기법

Realtime Traffic Event Management and Clustering Method

김 보 성, 최 도 진, 송 석 일
한국교통대학교

Bo-sung Kim, Do-jin Choi, Seokil Song
Korea National University of Transportation

요약

본 논문에서는 운행중인 차량이 수집한 위치별 교통 이벤트 (지체, 정체, 사고, 노면상태 등)를 다른 운행 차량과 실시간으로 공유하여 안전운행 서비스를 제공하기 위한 방법을 제안한다. 운행중인 차량은 차량내의 스마트 기기나 전용 기기를 이용해 수집한 교통이벤트를 실시간으로 서버로 전송하고 서버는 전송된 교통이벤트를 위치별, 시간별로 색인하고 중복된 교통이벤트를 분류하여 저장한다. 이런 모든 과정은 처리 속도 향상을 위해 Spark의 RDD를 이용해서 인-메모리에서 처리된다.

1. 서론

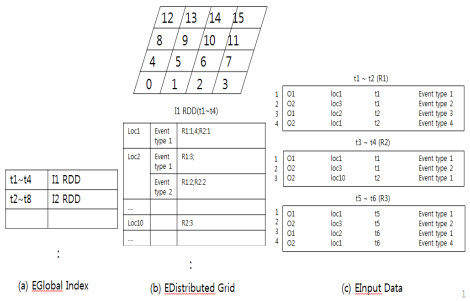
통신이 가능한 블랙박스나 스마트 폰과 같은 스마트 기기를 장착하는 차량이 급격히 증가하면서 이를 이용한 다양한 교통안전 및 교통편의 서비스들이 제공되고 있다. 특히, 차량내 스마트 기기가 각종 센서를 이용해서 수집한 교통 이벤트 (지체, 정체, 사고, 노면 상태 등)를 관련 영상과 함께 공유하는 서비스도 등장하였다.

서비스를 이용하는 차량이 많을 경우 서버가 교통 이벤트를 영상과 함께 수집하여 다른 차량과 공유하기 위한 처리를 하는 것은 매우 큰 부하를 수반한다. 특히, 교통이벤트는 실시간성이 매우 중요하므로 서버가 과부하로 교통이벤트 처리를 지연하게 되면 교통이벤트의 공유는 의미가 없어진다. 또한, 차량 통행량이 많은 구간에서의 교통이벤트는 중복될 가능성이 매우 높다. 따라서, 서버는 수집한 교통이벤트를 분석하여 중복 이벤트를 빠르게 분류할수 있어야 한다.

이 논문에서는 빠른 교통이벤트 저장 및 처리를 위해 인-메모리 기반의 데이터 처리 플랫폼인 Spark 및 Spark Streaming을 이용하는 방법을 제안한다. Spark Streaming을 이용하여 실시간으로 전송되는 교통이벤트 스트림을 위치와 시간에 따라 색인하여 저장한다. 현재 이동중인 차량의 위치에 따라 해당 교통이벤트를 빠르게 검색하여 차량에 전송할 수 있도록 한다. 또한, 실시간으로 전송되는 교통 이벤트의 발생 시간, 위치, 내용을 이용하여 동일 이벤트를 분류하여 불필요한 교통 이벤트를 차량들에 제공하지 않도록 한다.

2. 제안하는 교통이벤트 처리 기법

그림 1의 (a)는 본 Event Input에 대한 알고리즘이다. 이는 일정한 간격으로 Batch 처리하게 된다. 먼저 Message들이 들어오면 Message의 Type을 확인한다. 확인한 Type이 Event Tyep이면 그림2의 (c)와 같이 Grid Indexing을 하기 위한 EInput Data List에 해당 Message들의 data들을 차례대로 등록하게 된다. 등록된 Message는 두 번째 EInput Data을 Indexing하기위한 EDistributed Gird Hash Table에 등록하기 위하여 같은 시간 간격 때에 해당 Grid에 같은 Event가 등록이 되어있는지 확인을 하게 된다. 등록이 되어 있다면 기존 List안에 아래 그림 2의 (b) Loc1과 같이 데이터를 추가하고 같이 이벤트가 없다면 해당 Grid를 key로 하여 Hash Table에 Value으로 Event Type과 EInput Data의 key을 등록하게 된다. 그 다음으로 그림 2의 (a)와 같이, EGlobal Indexing을 위한 Hash Table에 시간대 간격을 key로 하고 해당 EDistributed Gird Hash Table을 찾을 수 있는 key을 value으로 등록한다. RDD형태로 해당 Hash Table들을 변환하게 된다. Grid Indexing을 위한 Data 입력이 완료되면 현재 발생되고 있는 Event들 등록되는 Running Event Hash Table에 해당 Data을 추가한다. 여기서 동일한 Event가 있다면 Event 후보로 추가 및 최근 Event가 들어온 시간을 나타내는 Last time을 수정한 뒤 다시 RDD 형태로 유지 하게 된다.



▶▶ 그림 1. Grid Index 구조

▶▶ 그림 2. 교통 이벤트 처리 알고리즘

그림 1의 (b)는 현재 진행 중인 Event 클러스터링에 대한 알고리즘이다. 이는 일정 시간 간격으로 Running Event Hash Table에 있는 각각의 Event Group의 가장 마지막에 들어온 시간을 확인한다. 설정된 시간 안에 Event Data가 들어오지 않았다면 해당 Event Group은 종료된 Event이라고 생각한다. 이 Event Group은 Finished Event Hash Table에 등록되고 Running Event Hash Table에서 삭제된다.

Grid Num1	Event type 1	gridnum : Int,latitudes : Double,longitudes : Double , evenType : String, starttime : Long, lasttime : Long ,masterEven : String, eventlist : List[String])
Grid Num2	Event type 1	
	Event type 2	
....		

(a) Running Event Group

Grid + Event Type + last Time 1	gridnum : Int,latitudes : Double,longitudes : Double , evenType : String, starttime : Long, lasttime : Long ,masterEven : String, eventlist : List[String])
Grid + Event Type + last Time 2	
...	

(b) Finished Event Group

▶▶ 그림 3. Event Guoup 구조

그림 3은 본론에서 설명한 Running Event Group과 Finished Event Group의 안에 Event Data가 어떤 형식으로 저장되는지 보여준다.

3. 결론

본 논문에서 Spark Streaming을 이용하여 실시간으로 교통 이벤트를 저장, 색인, 중복 처리하는 방법을 제안하였다. 제안하는 방법은 운행중인 차량이 수집한 교통이벤트를 실시간으로 다른 차량과 공유하여 안전 운행을 도와줄수 있다.

▌참 고 문 헌▐

[1] Resilient Distributed Datasets: A Fault-Tolerant Abstraction for In-Memory Cluster Computing. Matei Zaharia, Mosharaf Chowdhury, Tathagata Das, Ankur Dave, Justin Ma, Murphy McCauley, Michael J. Franklin, Scott Shenker, Ion Stoica. NSDI 2012. April 2012. Best Paper Award and Honorable Mention for Community Award.

[2] Spark: Cluster Computing with Working Sets. Matei Zaharia, Mosharaf Chowdhury, Michael J. Franklin, Scott Shenker, Ion Stoica. HotCloud 2010. June 2010.

[3] Shark: Fast Data Analysis Using Coarse-grained Distributed Memory (demo). Cliff Engle, Antonio Lupher, Reynold Xin, Matei Zaharia, Haoyuan Li, Scott Shenker, Ion Stoica. SIGMOD 2012. May 2012. Best Demo Award.

[4] Shark: SQL and Rich Analytics at Scale. Reynold Xin, Joshua Rosen, Matei Zaharia, Michael J. Franklin, Scott Shenker, Ion Stoica. Technical Report UCB/EECS-2012-214. November 2012.