http://dx.doi.org/10.3745/KIPSTD.2012.19D.4.271

TK-Indexing : NoSQL 기반 SNS 데이터 색인 기법

심 형 남[†] · 김 정 동[††] · 설 광 수[†††] · 백 두 권[††††]

요 약

현재 소셜 네트워크 서비스(Social Network Service: SNS)의 이용자 수가 늘어나면서 SNS에서 생성되는 콘텐츠 데이터의 양도 기하급수적으로 늘어나고 있다. 이러한 SNS는 개인의 근황, 관심사를 전달하기 위해 사용하고, 친목도모, 엔터테인먼트, 제품 마케팅, 최신 뉴스 공유, 1인 미디어 등 다양한 목적으로 활용하고 있다. SNS가 스마트폰에서 사용 가능해지면서 사용자들은 언제, 어디서나 실시간으로 사회의 주요쟁점이나 사회구성원들의 주 관심사와 같은 콘텐츠를 기존 미디어 매체보다 빠르게 생성하고 확산시킨다. 기존 웹 콘텐츠 색인 기법은 색인대상이 다양하고 정확성에 중점을 두어 색인하므로 실시간으로 대량 생성되는 SNS 콘텐츠를 색인하는 기법으로 한계가 있다. 이러한 문제를 해결하기 위하여 관계형 DBMS기반 실시간 색인 기법이 있으나 색인대상의 축소와 색인 절차의 복잡성이 높다는 단점이 있다. 따라서 본 논문에서는 실시간으로 생성된 SNS콘텐츠를 색인하기 위하여 NoSQL기반 SNS 콘텐츠 생성시간과 키워드를 각각 색인하는 TK-Indexing 기법을 제안하여 기존 색인 기법의 복잡성을 개선한다.

키워드 : NoSQL, Indexing Method, MapReduce, Social Network Service, Real-time Search

TK-Indexing : An Indexing Method for SNS Data Based on NoSQL

Hyung-Nam Shim[†] · Jeong-Dong Kim[††] · Kwang-Soo Seol[†††] · Doo-Kwon Baik[††††]

ABSTRACT

Currently, contents generated by SNS services are increasing exponentially, as the number of SNS users increase. The SNS is commonly used to post personal status and individual interests. Also, the SNS is applied in socialization, entertainment, product marketing, news sharing, and single person journalism. As SNS services became available on smart phones, the users of SNS services can generate and spread the social issues and controversies faster than the traditional media. The existing indexing methods for web contents have limitation in terms of real-time indexing for SNS contents, as they usually focus on diversity and accuracy of indexing. To overcome this problem, there are real-time indexing techniques based on RDBMSs. However, these techniques suffer from complex indexing procedures and reduced indexing targets. In this regard, we introduce the TK-Indexing method to improve the previous indexing techniques. Our method indexes the generation time of SNS contents and keywords by way of NoSQL to indexing SNS contents in real-time.

Keywords : NoSQL, Indexing Method, MapReduce, Social Network Service, Real-time Search

1. 서 론

소셜 네트워크 서비스(Social Network Service: SNS) 초기에 이용자들은 개인의 근황, 관심사를 이용자와 개인적 친분이 있는 이용자들과 공유하는 목적으로 사용하였다. 그러나 이용자 수가 늘어나고 친목도모, 엔터테인먼트 외에 불특정 다수의 이용자에게 제품 마케팅, 최신 뉴스 공유, 1인 미디어 등 다양한 목적으로 활용하면서 SNS에서 생성되는 콘텐츠 데이터의 양이 기하급수적으로 늘어나고 있다[1].

최근 Facebook[2], Twitter[3], me2day[13]와 같은 SNS가 스마트폰에서 사용 가능해지면서 사용자들은 언제, 어디서나 실시간으로 사회의 주요쟁점이나 사회구성원들의 주 관심사와 같은 최신콘텐츠를 빠르게 생성 및 검색한다. SNS는 SNS 사용자가 생성한 콘텐츠를 사용자의 친구들 또는 다수의 다른 회원에게 공개하고 공개된 콘텐츠 정보를 획득한 다른 회원들은 이를 다시 다른 회원들에게 공유하면서 빠른 시간 안에 콘텐츠 정보를 확산시킨다. 이런 특징을 기반으로 SNS는 전통적인 미디어매체인 신문, TV 뉴스, 인터넷 뉴스, 개인 블로그보다도 빠르게 실시간 뉴스를 확산시

※ 이 논문은 2011년 정부(교육과학기술부)의 재원으로 한국연구재단(NRF-2011-0025588)과 중소기업청에서 지원하는 2011년도 산학연공동기술개발사업(C0034549)의 지원을 받아 수행된 연구 결과물임. 이 연구에 참여한 연구자는 '2단계 BK21사업'의 지원을 받았음.
† 준 회 원 : 고려대학교 컴퓨터·전파통신공학과 석사과정
†† 준 회 원 : 고려대학교 컴퓨터·전파통신공학과 박사과정
††† 준 회 원 : 고려대학교 컴퓨터·전파통신공학과 석·박통합과정
†††† 종신회원 : 고려대학교 컴퓨터·전파통신공학과 교수
논문접수 : 2012년 5월 3일
수 정 일 : 1차 2012년 6월 7일, 2차 2012년 7월 9일
심사완료 : 2012년 7월 9일
* Corresponding Author : Doo-Kwon Baik(baikdk@korea.ac.kr)

킨다. 기존 미디어매체 보다 빠르게 최신 뉴스를 생성하고 확산시키기 때문에 인터넷 사용자들은 최신 뉴스를 검색하기 위하여 웹 기반검색과 함께 SNS기반 검색을 사용한다. 기존 웹 서비스 콘텐츠와 SNS 콘텐츠는 콘텐츠의 성격 및 구조적 차이를 가지기 때문에 SNS 콘텐츠를 검색하기 위한 연구가 활발히 진행되고 있다[4-5].

웹 서비스 콘텐츠의 검색방식은 검색할 대상을 텍스트, 이미지, 영상, 파일 등으로 확장하여 질의에 정확한 모든 정보를 검색하여 사용자에게 보여준다. 이를 위해서는 수많은 메타데이터를 기반으로 정교한 검색을 구현할 수 있는가에 초점을 맞춘다. 검색을 위해 크롤링을 이용하여 데이터를 수집하고 파일단위로 색인한 후 여러 메타데이터를 사용, 데이터의 유용성을 랭킹화한 후 다시 색인을 하는 절차를 가진다. 웹 서비스 콘텐츠는 일정 시간이 지난 후에 검색이 가능하여 실시간으로 생성된 콘텐츠를 검색하는데 적합하지 않다. 일단 실시간으로 생성된 최신 콘텐츠를 최대한 빠른 시간 안에 검색을 하기 위해서는 SNS 사용자가 콘텐츠를 생성한 후 실시간으로 검색이 되어야한다. 또한 기존 웹 문서들이 일정시간 동안 생성되는 것과는 달리 SNS 데이터는 짧은 시간에 방대한 양의 데이터가 생성되기 때문에 관계형 DBMS를 이용하는 기존 검색방법으로 실시간 대용량 데이터를 처리하는데 한계가 있다. 또한 SNS 콘텐츠는 웹 서비스 콘텐츠와 비교하여 색인요소와 랭킹요소가 부족하기 때문에 차별화된 색인, 검색 방법이 요구된다[6].

따라서 본 논문은 최신 생성된 SNS 콘텐츠를 실시간으로 검색하기 위해 SNS 콘텐츠 생성시간과 키워드를 각각 색인하는 2중 색인 기법을 제안한다. 또한 실시간으로 생성된 SNS 콘텐츠의 메타데이터와 내용을 효율적으로 처리하기 위해 SNS 콘텐츠 데이터 구조와 색인 데이터 구조를 제안한다. 그리고 대용량의 SNS 콘텐츠 데이터를 색인하기 위하여 NoSQL(Not Only SQL)을 데이터 저장소로 활용한다.

본 논문의 구성은 다음과 같다. 제 2장에서는 관련연구에 대하여 언급하고 제 3장에서는 이 논문에서 제안하는 TK-Indexing 기법(Timeline-Keyword Indexing) 모델 정의를 위한 데이터구조와 프로세서의 전반적인 절차를 기술한다. 제 4장에서는 제안 모델의 구현과정을 기술하고 제 5장에서는 기존의 색인 기법들과 비교 평가하여 색인 범위 및 색인 절차의 복잡성에 대해 정량적 평가와 제안하는 시스템에 대한 정성적 평가를 기술한다. 마지막으로 제 6장에서는 결론 및 향후 연구를 제시한다.

2. 관련 연구

2.1 SNS 실시간 검색

데이터베이스 시스템에서 인덱스는 효율적으로 질의(query) 처리하기 위해서 생성된다. 인덱스 처리는 대부분 전체 데이터를 처리하는 대신에 질의와의 높은 연관을 가지고 있는 데이터를 색인 처리를 한다. 대표적인 기법으로 몇 개의 Top-k[14-15] 질의에 대해 질의 결과를 구하고 이를 실체화된 뷰(view)로 활용하는 뷰기반이 있다[7, 12]. 이 방법은 질의와 유사한 뷰가 존재할 경우, 질의 처리 성능이 좋지만 그렇지 않은 경우, 질의 처리 성능이 매우 떨어진다.

TI(Tweet Index)도 Top-k기반으로 질의 결과에 해당하는 Tweet만 실시간 색인하고 나머지는 주기적으로 batch 색인하여 색인의 효율을 높였으나 트위터에만 적용되는 리트윗과 같은 요소를 랭킹요소로 랭킹함수를 추가하였으며 검색의 질을 높이기 위해 많은 랭킹함수가 추가되어 있다[8].

2.2 Not Only SQL (NoSQL)

특히 SNS 기업의 발전으로 데이터가 빠르게 증가함으로서 대용량 데이터를 효율적으로 처리해야 한다. 그러나 현재 데이터를 처리하는 관계형 DBMS는 대용량 데이터를 처리하는 속도가 매우 떨어진다. 현재 대용량 데이터들은 기존에 생성된 데이터들 예를 들어 회계, 문서, 인적사항 등의 데이터구조와는 다른 구조의 데이터가 생성된다. 이런 데이터를 분석, 처리하기 위해서는 기존의 방식과는 다른 데이터 저장소가 필요하다.

NoSQL은 기존 관계형 DBMS 한계를 극복하기 위한 데이터 저장소의 새로운 형태로 수평적 확장성을 가지고 있으며 관계형 데이터 구조가 아니기 때문에 조인(join)연산이 없고 고정된 스키마를 갖지 않음이 특징이다.

대표적인 NoSQL 제품은 구글의 BigTable[9], 아마존의 Dynamo[10]등이 오픈소스로 제공이 되어 같은 대용량데이터를 처리하기 위한 고비용 고사양의 관계형 DBMS장비와 비교할 때 경제적인 이점이 있다.

SQL을 지원하기 않기 때문에 편의성이 부족하고 트랜잭션을 지원하지 않기 때문에 정합성의 오류가 발생한다. 그러나 SNS 실시간 검색은 다른 요소보다 실시간으로 콘텐츠를 검색하여 처리하는 것이 최우선으로 중요하기 때문에 관계형 DBMS보다 높은 처리속도를 가지는 NoSQL을 이용하는 것이 성능향상에 좋다[16].

2.3 Map/Reduce

Map/Reduce[11]는 대용량 데이터를 처리하는 분산/병렬 시스템 지원을 목적으로 구글이 개발한 프레임워크 및 프로그래밍 모델로서 단일 마스터 노드와 다수의 슬레이브 노드로 구성되며 마스터 노드가 Map, Reduce task 단위로 슬레이브 노드에 할당하여 대용량 데이터를 처리한다. 그리고 비공유 구조로 클러스터에서 각 노드 고유의 CPU, 메모리, 하드 디스크로 구성되며 네트워크로 연결하여 확장성이 용이하며 저가의 범용 컴퓨터로 구성 가능하다. 또한 작업을 완료한 슬레이브 노드가 실패하여 접근할 수 없는 경우, 똑같은 작업을 다른 슬레이브 노드가 재수행하며 작업을 완료한 슬레이브 노드가 실패하여도 이미 그 결과 값이 저장되어 재수행을 하지 않아 병렬 시스템 중 몇 개에 문제가 있더라도 전체 시스템에 영향을 미치지 않는다.

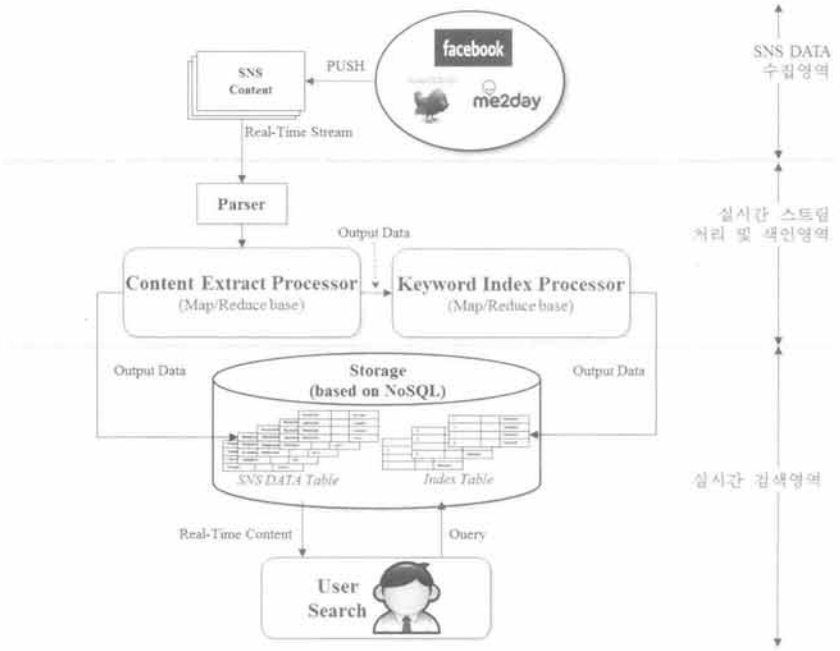

그림 1. 실시간 검색을 위한 제안 모델
Fig. 1. Proposed model for real-time search

프로그래밍 모델은 Map함수와 Reduce함수의 조합을 통해서 분산/병렬 시스템을 운영하는데 Map함수는 <Key, Value> 형식의 중간 데이터를 생성하고 Reduce함수는 Map함수에서 출력한 결과 값이 내부적으로 정렬되어 생성된 중간 결과 값 <Key, List<Value>>을 입력 값으로 하여 각 Reduce에서 생성된 결과 값을 병합하여 최종 결과 데이터로 출력한다.

3. NoSQL 기반 SNS 데이터 색인 기법

이 장에서는 NoSQL 기반 SNS데이터 색인 기법 모델을 설명하고 색인 기법을 위한 데이터 구조와 주요 프로세서를 설명한다.

3.1 제안 모델

제안하는 모델은 크게 "SNS 데이터 수집영역", "실시간 스트림 처리 및 색인영역", "실시간 검색영역"으로 크게 세 부분으로 구성한다.

- SNS DATA 수집영역 : SNS를 이용하는 사용자가 공개 콘텐츠를 생성 시 푸쉬(Push)기술 방식으로 콘텐츠를 실시간으로 받아 실시간 스트림으로 데이터를 보내기 전 데이터 수집 단계이다.
- 실시간 스트림 처리 및 색인영역: 실시간 스트림 방식으로 전달되는 SNS 콘텐츠를 실시간 정보에 맞는 데이터만 추출하기 위해 전단계로 파싱처리 한다. 파싱처리를 통해 의미 없는 정보와 태그를 제거한 데이터를 콘텐츠 추출 프로세서에서 Map/Reduce 프로그래밍 기반의 분산처리를 통해 타임라인(Time Line)순으로 정렬하게 된다. 이렇게 처리한 데이터로 NoSQL에 SNS DATA 테이블을 생성하고 키워드 인덱스 프로세서는 콘텐츠 추출 프로세서에서 생성된 데이터를 Map/Reduce 프로그래밍 기반 분산처리를 통해 키워드로 색인을 하고 NoSQL에 Index 테이블을 생성한다.
- 실시간 검색영역: 사용자가 키워드로 질의하면 NoSQL의 Index 테이블에서 해당 검색어를 포함한 Content ID를 SNS 데이터 테이블에서 검색하여 해당 내용을 타임라인(Time Line)순으로 사용자에게 전달한다.

이러한 제안 모델을 기반으로 SNS 데이터 생성시 실시간으로 색인이 가능하다.

3.2 TK-Indexing 데이터 구조

기존 검색과 달리 검색 대상이 문서가 아닌 SNS 데이터이므로 기존 색인 구조와 다른 데이터 구조가 필요하다. 일반적으로 SNS 데이터는 콘텐츠를 작성한 작성자의 ID, 성명, 콘텐츠를 포함하여 각각 SNS 마다 메타데이터가 존재한다. 각각의 메타데이터를 포함하는 데이터 구조를 생성할 경우 다양한 랭킹 함수를 이용하여 데이터 검색의 질을 향상시킬 수 있으나 SNS 마다 특징이 다르며 본 논문에서 다루고자 하는 최신 정보를 실시간으로 검색하기 위한 색인요소로 불필요하다. 실시간으로 생성되는 대용량 SNS 데이터를 처리하기 위해서는 데이터의 크기를 줄여야 한다. 본 논문의 데이터 구조는 SNS 데이터의 가장 큰 특징이라고 할 수 있는 사용자간의 관계 데이터를 제외 했으며 이를 통해서 알 수 있는 사용자의 영향력 척도를 제외 했다. 사용자간의 관계 데이터는 사용자간의 사회적인 관계를 나타내는 요소로 데이터의 실시간성을 보여주는 요소는 아니다.

데이터 색인요소로 SNS 콘텐츠 작성시간과 콘텐츠 내용을 포함했다. SNS 콘텐츠는 기존 문서 데이터와 다르게 비교적 적은 데이터(트위터에서는 140자 이내)를 가지며 그 용량이 제한적이다. 그리고 각 SNS 콘텐츠 생성 시간을 포함하여 작성 시간을 기준으로 검색 가능한 요소를 포함한다. TK-Indexing 색인 모델은 기존 색인 구조와 다른 2개의 데이터 구조를 포함한다. 첫 번째로 SNS 데이터 구조이다. SNS 데이터 구조는 표 1과 같다.

표 1. SNS 데이터 구조
Table 1. An SNS data structure

이름	내용
Content ID	Primary Key
Time	SNS 콘텐츠가 생성되는 시간
Content	콘텐츠 내용
User ID	콘텐츠 작성자 ID

Content ID 요소는 하나의 SNS 데이터 구조를 식별하기 위한 키 값으로 생성한다. Time 요소는 실제 tweet 또는 페이스북의 담벼락 등 각 SNS에서 콘텐츠들이 작성된 시간을 저장한다. Content는 SNS 이용자가 SNS에 작성한 글로 여러 데이터 중에서 실제 내용에 해당한다. 마지막으로 User ID 요소는 SNS에 게시글을 작성한 작성자 본인을 구별하기 위한 글이다. SNS 실시간 검색은 개인이 콘텐츠를 생성함으로서 콘텐츠를 생성한 작성자도 검색에 포함되어야 한다.

두 번째로 Keyword 데이터 구조이다. 표 2는 Keyword 데이터 구조를 보여준다.

표 2. Keyword 데이터 구조
Table 2. An Keyword data structure

이름	내용
Word ID	Primary Key
Word	키워드
Content ID	키워드가 포함된 SNS 데이터 ID

Keyword 데이터 구조는 사용자가 키워드 검색을 하는데 필수적인 데이터 구조로 SNS 데이터 구조로 이루어진 데이터 집합을 색인하기 위한 데이터 구조이다. Word ID 요소는 하나의 Keyword 데이터 구조를 식별하기 위한 키 값으로 생성하고 Word 요소는 SNS 데이터 구조 집합에서 Content 요소에서 각각 단어들 중 하나의 단어를 의미한다. 마지막으로 ID 요소는 Word 요소의 Keyword를 포함하고 있는 SNS 데이터 구조 집합에서의 Content ID 값을 의미한다.

3.3 TK-Indexing 시스템

본 논문의 기본적인 색인 절차는 SNS의 특성상 많은 사용자가 한 순간에 대량의 데이터를 발생하는 대량의 SNS 게시물과 메타데이터 중에서 그 내용과 작성된 시간을 기반으로 색인한다. 이런 데이터들을 실시간으로 색인 처리를 하기 위해서 Map/Reduce 기반의 분산처리가 유용하다. Map/Reduce 기반의 분산처리를 이용한 TK-Indexing 시스템은 전처리 부분과 2개의 프로세스로 구성한다. 각각 Parser와 Content Extract Processor 그리고 Keyword Index Processor로서 각각은 SNS에서 실시간으로 최신 정보를 추출하여 실시간 색인이 가능하도록 한다.

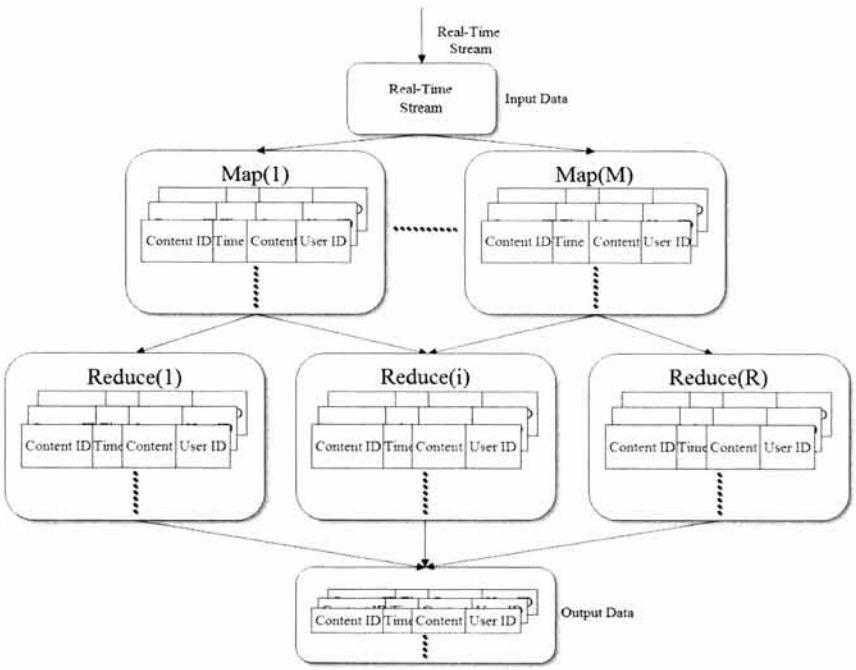

그림 2. Content Extract Processor
Fig. 2. Content Extract Processor

- Parser : 그림 1에서 SNS는 실시간으로 각각의 SNS를 이용하는 이용자들의 공개 콘텐츠들을 수집영역으로 보내게 된다. 이때 각각의 SNS들은 각각의 데이터들을 명명한다. 또한 본 논문에서 하고자하는 SNS 실시간 검색과는 거리가 먼 메타 데이터를 포함하고 있다. 데이터들은 주로 태그와 데이터로 표현되거나 JSON 포맷으로 데이터가 구성되어진다. 이런 데이터들 중에서 앞서 정의한 실시간 검색에 필요한 요소 SNS에서 사용자가 내용을 작성한 시간과 게시물의 내용 마지막으로 게시자의 아이디를 1차적으로 선별하는 전처리 작업이 필요하다. 본 논문에서는 각각의 SNS 가 푸쉬 해온 데이터를 Parser를 통해서 이 작업을 처리한다.
- Content Extract Processor : 그림 2에서와 같이 1차적으로 전처리가 된 SNS 데이터들은 Content Extract Processor로 대용량의 데이터가 실시간 입력이 된다. Content Extract Processor는 효율적인 병렬/분산처리가 가능하도록 입력된 데이터들을 일정 크기로 나눈다. 그리고 Content Extract Processor는 M개의 Map과 R개의 Reduce를 생성한다.

표 3. Content Extract Processor Map 함수
Table 3. Content Extract Processor Map function

```
Algorithm 1.
Content Extract Processer Map function
1: for ∀p_i ∈ P do /*P = ParsDataSet*/
2:  key←hash key
2:  value←p_i
3:  Map(key, value){
4:      Content ID = key
5:      Time = value.Time
6:      Content = value.Content
7:      User ID = value.User ID
8:      output key = (content ID)
9:      output value = (Time, Content, User ID)
10:     Emit(output key, output value)
11: }
12: end for
```

Content Extract Processor의 Map 함수는 할당된 데이터를 입력받는다. 그 후 <key, value> 값 형태로 데이터를 파싱한다. Key 값은 SNS 데이터 구조의 Content ID가 되고, value는 {Time, Content, User ID}가 된다. Content ID는 검색의 효율을 높이기 위해서 해쉬키가 할당된다. <key, value> 의 인자 값을 전달하면 Map 함수는 결과 값으로 중간 <key, list<value>> 데이터를 생성한다. Content Extract Processor의 Map 함수 알고리즘은 표 3과 같다. Content Extract Processor에서 할당된 reduce는 각 Map함수에서 생성된 중간결과 값을 입력 받는다. Content Extract Processor의 Reduce 함수는 value에 Time을 key로 하여 입력된 <key, list<<value>>를 기준시간 T로 정렬을 하게 된다. Content Extract Processor의 Reduce 함수 알고리즘은 표 4와 같다.

표 4. Content Extract Processor Reduce 함수
Table 4. Content Extract Processor Reduce function

```
Algorithm 2.
Content Extract Processer Reduce function
1: for ∀e_i ∈ E do /* E = EmitDataSet*/
2:  key←e.key
2:  value←e.value
3:  Reduce(key, value){
4:      if value.t < T then /* T는 기준시간 */
5:       g_i.key = key /*g는 그룹*/
6:       g_i.value = value
7:       output key = g_i.key
8:       output value = g_i.value
9:       Emit(output.key, output value)
10:     else
11:      g_j.key = key
12:      g_j.value = value
13:      output key = g_j.key
14:      output value = g_j.value
15:      Emit(output.key, output value)
16:     end if
17: }
18: end for
```

Content Extract Processor의 Reduce는 각 Reduce 함수에서 생성된 결과 값을 병합하여 최종 결과 데이터를 생성한다. 그리고 마지막으로 NoSQL 기반의 데이터베이스 스토리지에 SNS DATA 테이블에 최종 데이터를 입력한다.

- Keyword Index Processor : 그림 3과 같이 Keyword Index Processor는 Content Extract Processor를 통해 생성시간기준으로 정렬된 SNS 데이터를 입력받는다. 입력받은 SNS 데이터에서 각 Content의 키워드를 추출한다. 키워드를 추출하기 위해서 Content를 어절 단위로 나누고 나누어진 어절에서 불용어를 제거한다. SNS 데이터를 Content ID와 Content에 포함된 단어들의 집합으로 생성한다. 그 후 효율적인 병렬/ 분산처리가 가능하도록 입력된 데이터들을 일정한 크기로 나눈다.

Keyword Index Processor는 M개의 map과 R개의 reduce를 생성한다. Keyword Index Processor Map함수 알고리즘은 표 5와 같이 입력된 데이터를 <key, value> 값으로 파싱한다. key 값은 SNS 데이터의 Content ID이고 value는 Content에 포함된 단어이다. Keyword Index Processor Map 함수는 중간 데이터로 Content ID가 key값이고 Content가 포함하고 있는 단어들이 Value값을 생성한다.

Keyword Index Processer Reduce함수는 Keyword Index Processer Map 함수에서 생성된 결과 값이 내부적으로 정렬되어 생성된 중간 결과 값 <key, list<value>>을 입력 값으로 한다. Keyword Index Processer Reduce 함수는 value에 word를 key로 하여 inverted 인덱스를 한다. 일차적으로 <word, Content ID> 형태에서 효율적인 색인을 위해 output key 값은 해쉬키를 가지고 output value는 {word, Content ID}를 갖는 <key, value> 리스트로 정렬을 하게 된다. Keyword Index Processor Reduce 함수 알고리즘은 표 6과 같다.

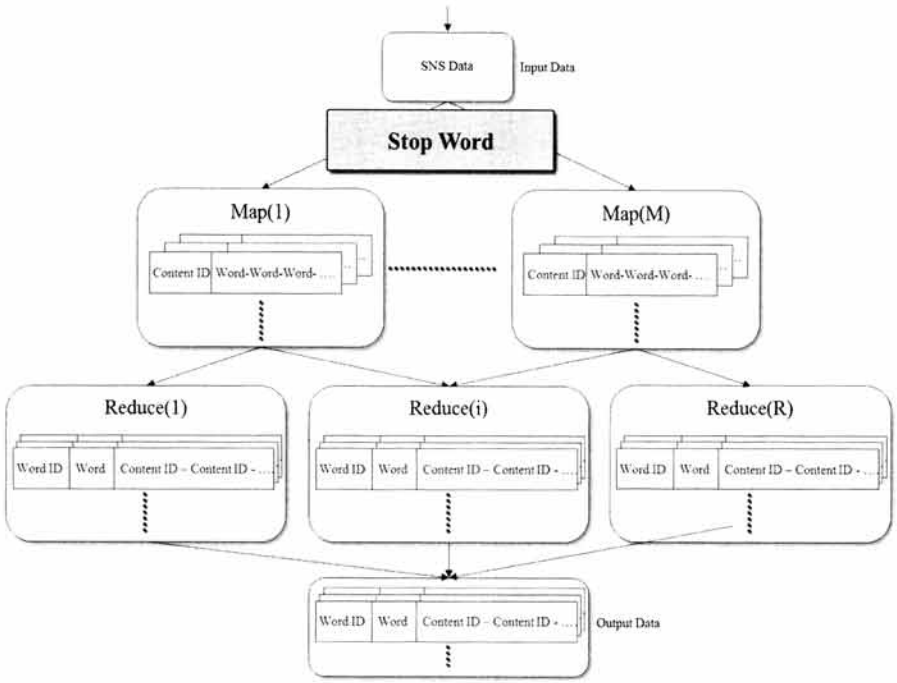

그림 3. Keyword Index Processor
Fig. 3. Keyword Index Processor

표 5. Keyword Index Processor Map 함수
Table 5. Keyword Index Processor Map function

```
Algorithm 3.
Keyword Index Processer Map function
1: for ∀s_i ∈ S do /*S = SNS DataSet*/
2:  key ← s_i.Content ID
2:  value←keyword
3:  Map(key, value){
4:      ID = key
5:      Word = value
6:      output key = (ID)
7:      output value = (word)
8:    Emit(output key, output value)
9: }
10: end for
```

표 6. Keyword Index Processor Reduce 함수
Table 6. Keyword Index Processor Reduce function

```
Algorithm 4.
Keyword Index Processer Reduce function
1: for ∀e_i ∈ E do /* E = EmitDataSet*/
2:  key←e.key
2:  value←e.value
3:  Reduce(key, value){
4:    inverted index by value
5:    output key ← hash key
6:    output value = (value, key)
7:  Emit(output key, output value)
8: }
9: end for
```

Keyword Index Processor는 최종 정렬된 데이터를 NoSQL 기반의 데이터베이스 스토리지에 Index 테이블을 생성하고 저장한다.

- Query Processor : Content Extract Processor와 Keyword Index Processor로 처리된 SNS Content는 각각 SNS 데이터 테이블과 Index 테이블로 저장되어 관리한다. Index 테이블은 하나의 Keyword가 포함되어 있는 다수의 SNS DATA Content를 포함한다. 관계형 데이터베이스 관리시스템에서는 복잡한 Join연산을 유발하여 실시간 검색의 질을 떨어뜨리게 된다. 본 논문에서는 NoSQL기반의 비정규화 데이터베이스 모델과 실시간 SNS 검색을 사용하는 사용자의 Query를 처리하기 위한 Query 프로세서가 존재 한다. 그림 4과 같이 Query 프로세서는 사용자에게 질의어를 입력받고 질의어를 Index 테이블에서 검색 후 질의어가 포함된 Content ID list를 결과값으로 받는다. 그 후 Content ID를 검색하여 해당 Content를 사용자에게 출력한다.

4. 구 현

이 장에서는 제안한 모델이 기존 모델과 비교하여 SNS 데이터를 실시간으로 처리하는데 있어 복잡성과 손실률이 낮음을 보인다. 구현은 그림 5와 같이 Windows 7 Enterprise K, Intel(R) Core(TM) i3-2100 3.10Hz, 2.0 GB Memory, Microsoft .NET Framework 4.030319 RTMRel, MongoDB win32-i386-2.0.3 환경에서 수행하였다.

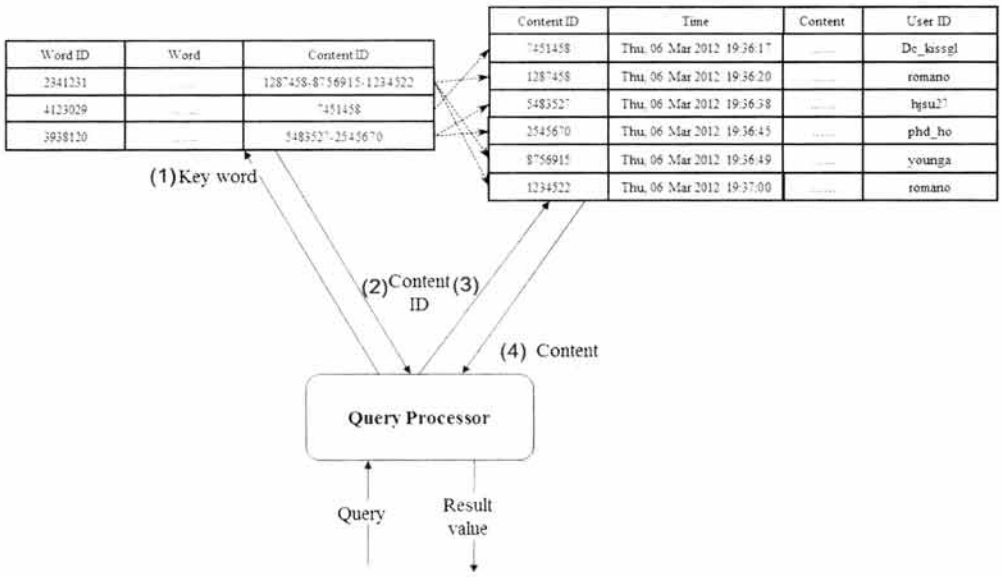

그림 4. Query Processor
Fig. 4. Query Processor

4.1 웹 클라이언트

웹 클라이언트는 SNS 데이터를 수집하기 위한 모듈이다. 공개되어 있는 SNS 중 오픈되어 있는 Twitter의 SNS 데이터를 이용했다. 실시간으로 SNS 데이터를 수집하기 위해 Twitter API를 이용한 모듈이 필요로 했다. 웹 클라이언트는 인터넷에 접속하여 리소스에 연결하는 객체와 이를 통해서 Twitter API를 요청하고 응답을 받을 수 있다. 또한 Twitter API를 통해서 실시간으로 받은 SNS데이터를 스트림으로 변환하는 객체와 이를 인코딩하는 객체가 포함되어 있다. 수집된 SNS 데이터는 JSON형식인데 웹 클라이언트에서 일정량의 데이터를 수집하면 이를 파싱하기 위하여 JSON 파서 클래스를 호출한다.

그림 5. 구현 화면
Fig. 5. Implementation of the screen

4.2 JSON 파서

본 논문에서 사용하는 Twitter의 SNS 데이터는 JSON형식으로 구성되어 있다. JSON 형식의 Twitter SNS 데이터들은 복잡한 JSON 객체의 집합으로 이루어져 있으며 객체는 Key/value 형식으로 이루어져 있다. 특히 Value는 집합의 구조로 이루어진 경우도 있다. JSON 파서 객체는 복잡한 SNS 데이터에서 작성자 ID, 콘텐츠, SNS 데이터 작성시간을 텍스트 형식으로 추출하는 기능을 가진다.

4.3 SNS 데이터

SNS 데이터 구조에 맞게 JSON파서를 통해 가공된 데이터를 처리하는 클래스이다. 주요 기능으로 해쉬키를 생성하여 ID 값을 입력하며 SNS 데이터 작성시간이 그리니치 평균시로 되어 있기 때문에 이를 한국시간에 맞추는 기능을 가지고 있으며 나머지 텍스트 데이터를 입력하는 기능을 가진다. 그림 6은 추출된 SNS 데이터 결과이다. 그림 6에서 _id는 해쉬키이며 Time은 Content가 생성된 시간이다. 그리고 User ID는 Content를 작성한 사용자의 ID이며 Content는 작성된 내용이다.

4.4 불용어 분석과 SNS 문서

SNS 데이터 구조에 입력된 Content 텍스트에서 단어만을 추출하는 클래스이다. 주요 기능으로 먼저 문장을 음절단위로 나눈다. 음절 단위로 나누어진 문장에서 명사만을 추출하기 위해서 조사 밑 동사 사전을 구축한다. 음절단위의 문단을 조사 밑 동사 사전과 비교하여 음절에서 조사 및 동사부분을 삭제한다. 최종적으로 한 문장이 가지고 있는 명사를 모으는 기능을 한다. SNS 문서 클래스는 SNS 데이터 ID와 불용어 분석기를 통해 생성된 SNS 데이터의 keyword집합들로 이뤄지는 객체를 생성한다. 그림 7은 불용

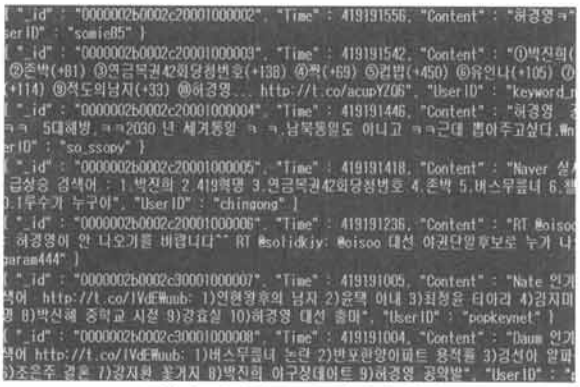

그림 6. 추출된 SNS 데이터 화면
Fig. 6. The extracted data screen SNS

그림 7. 불용어가 제거된 콘텐츠 결과 화면
Fig. 7. The stopword has been removed, the resulting screen content

그림 8. Keyword 색인 결과 화면
Fig. 8. Keyword Index Results screen

어가 제거된 콘텐츠 결과 화면으로 콘텐츠가 포함한 단어들이 ID를 기준으로 정렬되어 있으며 그림 8은 Keyword로 색인된 결과로 단어가 포함되어 있는 ID들이 하나일 경우 value로 2개 이상일 경우 title로 집합형태임을 보인다.

4.5 MongoDB

MongoDB 클래스는 NoSQL의 일종인 MongoDB 데이터베이스와 어플리케이션과의 데이터 통신이 주 목적이다. 먼저 데이터베이스와 통신을 하는 모듈과 Collection을 생성하는 모듈이 있으며 SNS 데이터와 SNS 문서 클래스를 각각 시간과 키워드 기반의 색인 처리를 하는 모듈을 포함한다.

5. 시험 및 평가

이 장에서는 실시간 검색을 위한 NoSQL기반의 TK-Indexing 기법이 기존 DBMS를 사용한 TI[8] 검색기법과 비교를 통해 제안한 기법의 성능을 정성적으로 평가하였으며 제안기법과 관계형 DBMS의 Full-Text 색인 속도를 정량적으로 평가한다.

5.1 프로세스 복잡도

표 7은 제안 모델과 기존 검색기법에서 실시간 검색의 복잡성을 정성적으로 평가한다. 제안 모델은 색인 전처리로 파싱처리가 있다 이는 기존 모델도 마찬가지로 필수적으로 필요한 처리이다. 그러나 기존 모델은 추가적으로 질의분석기를 통한 Top-K 선별 모듈이 필요하며 또한 Top-K를 조절하는 모듈이 추가적으로 필요하다 제안 모델은 시간과 키워드 중심의 2중 색인을 하는 반면에 기존 모델은 Full-Text 색인 처리를 한다. 색인 처리 후에도 검색의 질을 높이기 위해서 User page 랭킹 모듈과 인기 토픽 계산 모듈, 시간기반 랭킹 모듈 등이 필요로 한다. 그러나 제안 모델에서는 색인 후 처리가 따로 필요하지 않다. 이를 통해 검색에 필요한 연산을 줄여 복잡성을 낮은 장점이 있다.

표 7. 시스템 Process 모듈 개수
Table 7. Process number of system modules

	제안 모델	기존 모델
색인 전처리	1	3
색인 처리	2	1
색인 후 처리	0	3
합계	3	7

5.2 데이터 손실률

기존모델은 Top-K방식으로 처리하는 SNS 데이터의 전체 크기를 줄여 데이터 색인 처리 속도를 높인다. 그러나 상위 질의어에 해당하는 데이터만을 우선적으로 색인을 하기 때문에 전체 데이터 중에 손실되는 데이터의 비율이 표 8과 같다. 제안 모델은 실시간으로 발생된 전체 SNS 데이터를 모두 색인하여 데이터 손실률이 없는 장점이 있다.

표 8. SNS 데이터 색인 비율
Talble 8. SNS Data Index Ratio

Top-k	제안 모델	기존 모델
10	100 %	15%
20	100 %	24%
30	100 %	39%
40	100 %	47%
50	100 %	52%
60	100 %	59%
70	100 %	61%
80	100 %	63%
90	100 %	67%
100	100 %	69%

5.3 색인 속도

제안 모델과 기존모델의 색인 속도를 비교평가 하였다 기존 모델은 관계형 DBMS를 이용하여 Full-Text 색인을 하였다. 그림 9와 같이 제안 모델의 색인처리 속도가 기존 모델의 색인처리 속도보다 빠르다.

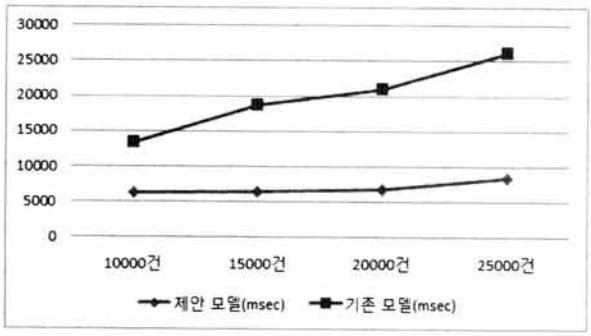

그림 9. 색인 처리속도
Fig. 9. Index Processing Speed

6. 결론 및 향후연구

본 논문에서는 SNS 실시간 콘텐츠를 색인하기 위해 시간 요소와 키워드 요소를 색인하는 2중 색인 기법을 및 효율적인 색인을 위해서 Map/Reduce를 활용 NoSQL 기반의 색인 모델을 제안했다.

제안한 TK-Indexing 기법은 기존색인 방법과 비교하여 실시간성, 복잡성에서 효과적인 색인 방안을 제시하였다. 또한 대용량 처리를 하기 위한 데이터 구조를 제시함으로 빠르게 증가하는 SNS 데이터를 실시간 처리한다.

향후 연구로는 랭킹 기법에 대한 연구와 제안한 모델에 검색의 품질을 향상시킬 수 있는 방향에 대해 모색이 요구된다.

참 고 문 헌

[1] Jansen, B.j., et al, "Real Time search on the web: Queries, topics, and economic value", Information Processing and Management, Vol.47, Issue.4, pp.491-506, 2011.

[2] Facebook, http://www.facebook.com, 2012.

[3] Twitter, http://www.twitter.com, 2012.

[4] David Geer, "Is It Really Time for Real-Time Search", IEEE Computer Society, Vol.43, Issue.3, pp.16-19, 2010.

[5] Shim H, Kim J, Baik D, "TK-Indexing method based on NoSQL for real-time search", KIISE 2011 Fall Conference, Vol.28, Issue2, Seoul National Univ., 2011.

[6] Bernard J.jansen, "Real time search user behavior", In: CHI Extended Abstracts, pp.3961-3966, 2010.

[7] Das, G., Gunopulos, D., Koudas, N., and Tsirogiannis, D, "Answering top-k queries using Views", International Conference on Very Large Data Bases, pp.451-462, 2006.

[8] Chun chen, Feng Li, Beng chin Ooi, Sai Wu, "TI: An Efficient Indexing Mechanism for Real-Time Search on Tweets", SIGMOD/PODS 2011, ACM Press(2011), pp.649-660, 2011.

[9] Fay Chang, Jeffrey Dean, Sanjay Ghemawat, et al., "Bigtable: A distributed storage system for structured data", OSDI'06 Proceedings of the 7th conference on Symposium on Operating Systems Design and Implementation, 2006.

[10] G. DeCandia, D.Hastorun, M. Jampani, G. Kakulapati, A.Lakshman, A. Pilchin, S. Sivasubramanian, P. Vosshall, and W. Vogels, "Dynamo: Amazon's Highly Available Key-Value Store", ACM Symposium on Operating Systems Principles, 2007.

[11] Jeffrey Dean, Sanjay Ghemawat, "MapReduce: simplified data processing on large clusters" the 6th conference on Symposium on Operating Systems Design & Implementation, 2004.

[12] R. Chirkova, C. Li, and J. Li. "Answering queries using materialized views with minimum size", The VLDB Journal, 15(3):191-210, 2006.

[13] me2day, http://www.me2day.net, 2012.

[14] V. Hristidis and Y. Papakonstantinou, "Algorithms and applications for answering ranked queries using ranked views," The VLDB Journal, Vol.13, No.1, 2004.

[15] C. Li, K. C.-C. chang, I. F. Ilyas, and S. Song, "RankSQL: Query Algebra and Opimization for Realational Top-k Queries", In Proc. Int'l Conf. on Management of Data, ACM SIGMOD, Baltimore, Maryland, June, 2005.

[16] B. F. Cooper, A. Silberstein, E. Tam, R. Ramakrishnan, and R. Sears, "Benchmarking cloud serving systems with YCSB", In SoCC '10: Proceedings of the 1st ACM symposium on Cloud computing, pp.143-154, 2010.

심 형 남

e-mail : romano@korea.ac.kr

2010년 수원대학교 인터넷정보공학과 (공학사)

2010년~현 재 고려대학교 컴퓨터·전파통신공학과 석사과정

관심분야 : 데이터베이스, 클라우드 컴퓨팅, NoSQL 등

김 정 동

e-mail : kjd4u@korea.ac.kr

2008년 고려대학교 컴퓨터학과(이학석사)

2008년~현 재 고려대학교 컴퓨터·전파통신공학과 박사과정

관심분야 : 메타데이터, 데이터 통합, 시맨틱 웹, 온톨로지, 소셜 네트워크, 상황인지

설 광 수

e-mail : seolks@korea.ac.kr

2012년 고려대학교 컴퓨터정보학과 (공학사)

2012년~현 재 고려대학교 컴퓨터·전파통신공학과 석·박통합과정

관심분야 : 소셜 네트워크 서비스, 시맨틱 웹, 온톨로지

백 두 권

e-mail : baikdk@korea.ac.kr

1974년 고려대학교 수학과(학사)

1977년 고려대학교 산업공학과(석사)

1983년 Wayne State Univ. 전산학과(석사)

1985년 Wayne State Univ. 전산학과(박사)

1986년~현 재 고려대학교 컴퓨터·전파통신공학과 교수

관심분야 : 데이터 모델링, 시뮬레이션, 데이터 공학, 소프트웨어 공학, 프로젝트 관리