트위터 분석을 위한 분산 시스템 설계 및 구현

윤진영, 김석중, 이범석, 황병연
가톨릭대학교 컴퓨터공학과
e-mail: {my_sk_test, typicalkorean, bumsuk, byhwang}@catholic.ac.kr

DiSAnT: Design and Implementation of Distributed System for Analysing Twitter

Jinyoung Yoon, Sukjoong Kim, Bumsuk Lee, Byung-Yeon Hwang
Dept. of Computer Science and Engineering
The Catholic University of Korea

요 약

트위터는 대표적인 소셜 네트워크 서비스이며 스마트 기기의 발달로 사용자 수뿐만 아니라 생성되는 트윗의 수도 지속적으로 늘고 있다. 또한 트위터는 인증과정을 통하여 API 요청을 제한해 데이터의 수집이 어렵기 때문에 트위터 기반 연구를 위해서는 빅 데이터를 처리하기 위한 분산처리 기술이 요구된다. 본 논문에서는 네트워크로 연결된 다수의 클라이언트를 이용해 계정과 트윗의 수집에 용이하고 수집한 데이터를 분석할 수 있는 기능까지 추가한 분산처리 시스템인 DiSAnT을 소개한다.

1. 서론

2006년 처음 서비스를 시작한 트위터는 전 세계에 3억 명 이상의 사용자를 가진 대표적인 소셜 네트워크 서비스 업체로 성장하였다. 입력할 수 있는 트윗의 길이는 텍스트 140자로 제한되어있으며, 사용자들은 트윗 작성을 통해 자신의 의견을 표현하고 정보를 공유할 수 있다.

이처럼 단순한 구조의 웹서비스는 스마트 기기의 빠른 성장과 함께 많은 사용자를 보유하게 됨으로써 관련 산업뿐만 아니라 소셜 네트워크 분석 연구를 진행하는 학자들에게서도 큰 주목을 받게 되었다. 최근에는 트위터에서 생성되는 트윗의 수가 매일 3억 개 이상으로 크게 늘어남에 따라, 소셜 네트워크 분석을 위해서 빅 데이터(big data)를 효율적으로 처리하기 위한 분산처리 시스템의 개발이 필요해졌다. 분산처리기술이란 네트워크에 연결된 다수의 컴퓨터를 이용하여 거대한 계산 문제를 해결하려는 분산처리 모델을 말한다[1]. 소셜 네트워크 분석 외에도 생물정보학, 의학 등 다양한 분야의 연구에서 빅 데이터를 처리하기 위해 분산처리기술을 사용하고 있다. 이러한 분산처리기술 중 최근 주목받고 있는 것이 아파치 연구재단의 하둡(Hadoop) 프로젝트이다. 하둡파일시스템(HDFS: Hadoop Distributed File System)은 각각의 데이터노드(Datanode)를 가진 다수의 서버를 이용하여 빅 데이터를 나누어 처리한다[2]. 모든 데이터를 가진 서버의 입장에서 소셜 네트워크 분석을 한다면 하둡의 분산된 저장구조가 유용하지만, 우리와 같은 클라이언트의 입장에서는 스트림 형태로 들어오는 빅 데이터를 실시간 처리해야하므로 분산처리 기능을 극대화 하는 것이 더 효율적일 수 있다.

본 논문에서는 트위터에서 데이터를 수집하고, 수집한 데이터를 분석할 수 있는 DiSAnT 시스템을 소개한다. 구현한 DiSAnT 시스템은 네트워크로 연결된 다수의 클라이언트로 이루어져있으며 클라이언트의 수는 필요에 따라 증감할 수 있다. 또한 계정수집모듈, 트윗수집모듈, 데이터 분석모듈로 구성되어있고, 각 모듈은 분산된 클라이언트에서 메인 시스템의 명령을 통해 데이터 수집 및 분석 작업을 수행한다. 본 논문의 의의는 다음과 같이 요약할 수 있다. 1) 빅 데이터를 가진 서버에 대해 제한적 접근만 가능한 클라이언트의 입장에서 효율적인 분산처리 시스템을 소개한다. 2) 실제로 데이터를 수집하고 분석함으로써 구현한 DiSAnT 시스템이 소셜 네트워크 분석에 적절한 구조와 성능을 가지고 있음을 확인한다.

논문의 구성은 다음과 같다. 2장에서는 관련연구로써 하둡 분산파일시스템과 트위터의 인증에 관한 내용을 다뤄 분산처리의 필요성을 설명한다. 3장에서는 트위터 데이터의 분석을 위해 구현한 DiSAnT 시스템의 구조와 기능에 대해 소개하고 4장에서 결론 및 향후 연구 방향을 제시한다.

2. 관련연구

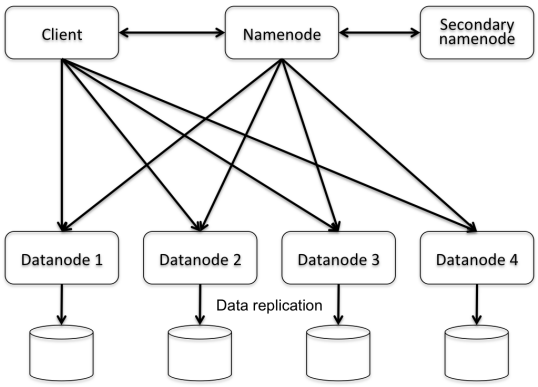
2.1 하둡 분산파일 시스템

트위터는 다량의 데이터가 스트림 형태로 발생하고 OAuth(Open Authorization) 프로토콜을 이용해 인증된 클라이언트의 요청을 제한하기 때문에 다수의 클라이언트를 이용한 분산처리기술의 적용이 필수적이다. 대표적인 분산 데이터 저장 기법과 분산 병렬 처리 기법은 Google의

※ 이 논문은 2011년도 정부(교육과학기술부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임(No. 2011-0009407).

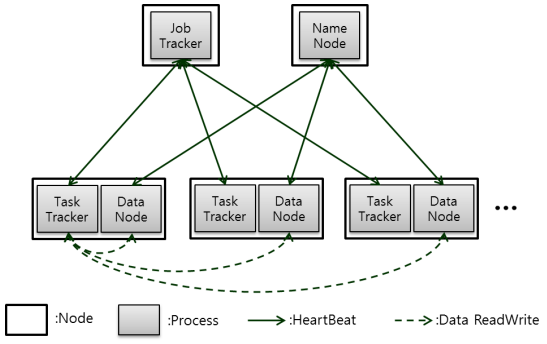
GFS(Google File System)[3]와 MapReduce[4]가 있으나, 최근에는 아파치의 하둡 프로젝트가 다양한 연구[5,6]에 적용되고 있다.

하둡은 분산파일시스템인 하둡파일시스템과 분산처리 시스템인 맵리듀스(MapReduce)로 구성되어 있다. 하둡파일시스템과 맵리듀스 모두 Master/Slave 구조인데, 하둡파일시스템에서는 이들 각각을 Namenode와 데이터노드라고 부르며 맵리듀스에서는 각각 잡트래커(Job Tracker)와 태스크트래커(Task Tracker)라고 부른다. 그림 1은 하둡파일시스템의 구조를 보여준다. Namenode는 Master로써 파일시스템의 메타데이터이고 실제 데이터는 여러 대의 데이터노드에 분산해 저장한다. Secondary namenode는 백업 노드로 사용되어지고, 각 데이터노드는 각각 하나의 저장노드(Storage node)를 가진다.



(그림 1) 하둡파일시스템 구조

맵리듀스는 그림 2와 같이 하둡파일시스템에 의해 분산 저장된 데이터를 여러 대의 태스크트래커에서 병렬로 처리하는 구조를 갖는다.

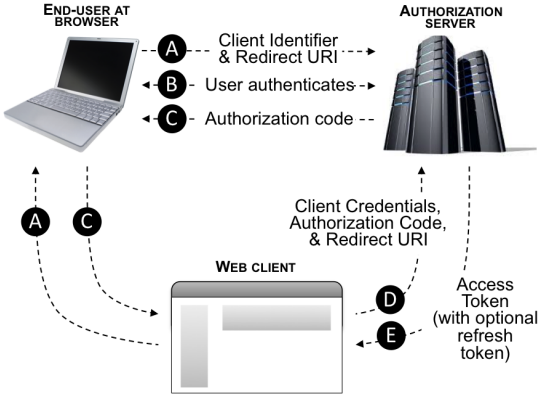


(그림 2) 맵리듀스 구조

각각의 데이터노드마다 태스크트래커가 존재하며 모든 태스크트래커를 하나의 잡트래커가 지속적으로 감시한다. 잡트래커가 자동으로 작업을 할당하고 결과를 통합해 주기 때문에 빅 데이터를 처리함에 있어 용이하다. 또한 기술이 오픈소스로 공개되어있고 처리과정을 API로 제공하기 때문에 응용프로그램의 개발이 쉽다는 장점이 있다.

2.2 OAuth 인증과 트위터 API

트위터 기반 플랫폼이나 응용프로그램을 개발하는 대부분의 연구에서는 트위터에서 제공하는 API를 사용한다. 트위터 API에는 사용자(User) 정보, 타임라인(Timeline) 등을 포함하는 Rest API와 검색 관련 Search API, 그리고 Streaming API가 있다. API의 실행결과는 HTTP status code에 반영되어, 만약 존재하지 않는 API를 실행하려고 할 경우 '404 Not Found' 에러가 발생한다[7].



(그림 3) OAuth 2.0의 흐름도(A-E: 인증순서)

트위터의 API를 사용하기 위해서는 OAuth 프로토콜 기술을 이용하여 인증과정을 거쳐야 한다. OAuth는 유저 이름과 패스워드 대신 규약에 따라 정해진 순서로 입수한 토큰을 사용해서 인증하는 방식이다[8]. 그림 3과 같은 구조로 인증과정을 거친 후, 시간당 제한된 수의 HTTP 요청을 처리할 수 있다. 이러한 구조는 사용자 인증 기능 뿐만 아니라 트위터가 자사의 데이터를 보호하고 무분별하게 발생하는 트래픽을 줄이려는 목적을 가진다. 그러나 빅 데이터를 필요로 하는 트위터 기반의 연구 플랫폼을 개발할 때는 제약이 된다.

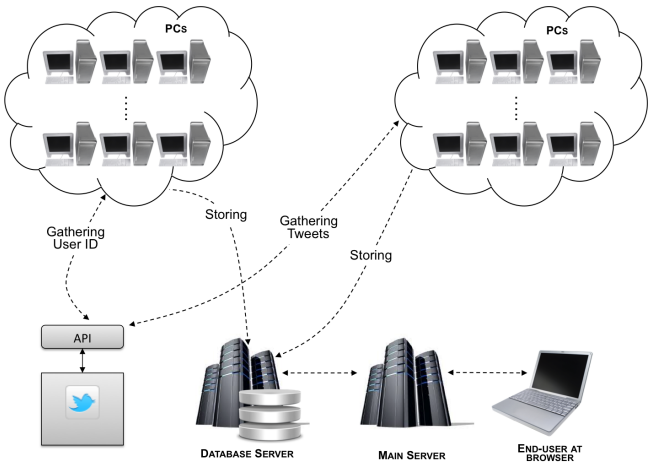
본 논문에서 제안하는 DiSAnT 시스템은 모든 데이터를 가진 서버의 입장이 아니라, 클라이언트 입장에서 스트림 형태로 저장되는 빅 데이터를 실시간으로 처리하기 위한 시스템이므로 데이터를 분산하여 저장하는 하둡을 사용하지 않았다. 따라서 데이터 수집을 위해 다수의 클라이언트 PC에서 트위터 API에 접속한다는 점에서 성형(star) 구조를 가진다. 클라이언트 PC에 개별적으로 데이터베이스를 두지 않고 별도의 데이터베이스 서버를 두어 관리하며 수집한 데이터를 처리한 다음 그 결과물을 데이터베이스에 저장한다.

3. DiSAnT: 트위터 분석을 위한 분산 시스템

3.1 DiSAnT 구조

트위터 분석 시스템을 개발하기 위해서는 데이터를 수집하고 필요한 형태로 정제하는 과정이 필요하다. 따라서 사용자의 계정과 작성된 트윗을 수집하는 단계를 거친 후

에 데이터 분석을 수행할 수 있다. 데이터 분석 단계에서는 수집된 계정의 프로필 위치정보와 각 계정별로 추출한 트윗의 위치정보를 바탕으로 두 GPS 좌표사이의 관계를 분석한다.



(그림 4) DiSAnT의 구조

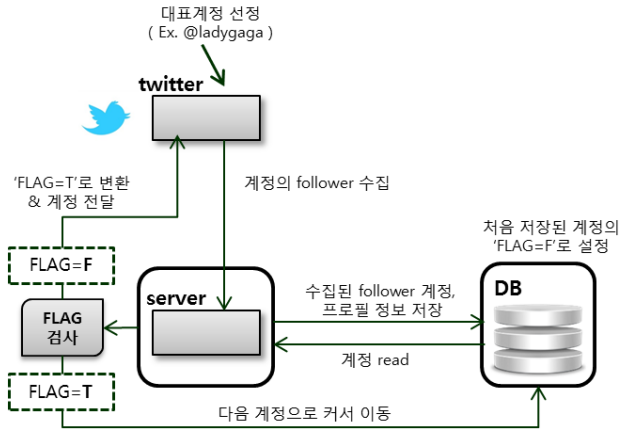
DiSAnT는 트위터로부터 빅 데이터를 처리하기 위해 다수의 클라이언트로 구성된 분산처리 구조로 이루어져 있으며, 그림 4와 같이 트위터 계정 수집과 트윗 수집을 위해 PC들을 두 그룹으로 나눴다. 두 그룹은 개념적으로 분류한 것이며, 처리할 데이터의 양에 따라 사용자 계정 수집 또는 트윗 수집을 위해 사용하는 PC의 수를 조절한다. 그룹에 속한 PC는 모두 개별적인 클라이언트로써 OAuth 인증을 받아 트위터 API에 접속하고, 받아온 데이터를 처리하여 데이터베이스 서버에 저장한다.

3.2 트위터 분석 프로그램의 구조

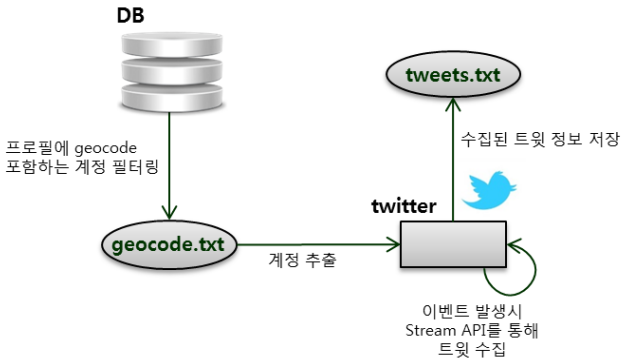
구현된 시스템의 분석 모듈을 시험해보기 위해 트위터의 사용자 프로필에 입력된 위치정보의 신뢰성에 대해 분석하였다. 이 분석에서는 사용자가 어느 위치에서 가장 트윗을 많이 남겼으며, 그 위치들 중에서 어느 위치가 프로필의 위치정보와 매치(match)되는지를 판단한다. 이에 앞서 계정 수집과 각 계정이 작성한 트윗의 수집이 선행되어야하며, 3.1절의 분산처리 구조에서 보인 두 개념적 그룹을 각각 계정수집모듈과 트윗수집모듈로 나타내었다.

그림 5는 트위터 API를 통해 계정을 수집하는 과정을 보여주고 있다. 설정된 기준 계정 하나를 시작으로 그 계정의 팔로워들을 수집하면서 데이터베이스에 계정과 프로필 정보인 위치정보, 언어 등을 저장한다. 이 과정에서는 FLAG를 통해 중복된 계정의 팔로우 수집을 피한다.

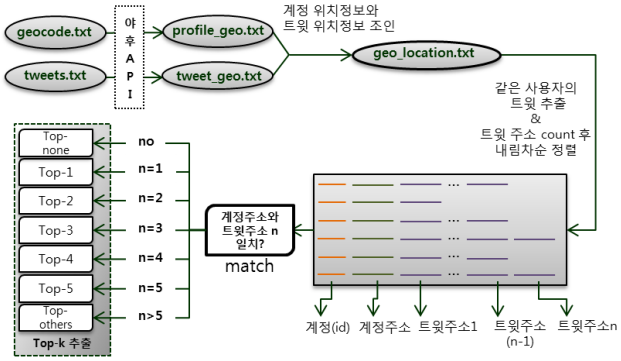
트윗을 수집하는 과정은 계정을 수집하는 과정과 비슷하다. 수집한 계정 중 프로필에 geocode 정보를 가진 사용자를 필터링한 다음 Streaming API를 통해 트윗과 관련된 정보를 수집한다. 그림 6에서 Streaming API는 사용자가 트윗을 작성하는 등의 이벤트가 발생할 때마다 실시간으로 새로운 정보를 받아온다.



(그림 5) 계정수집모듈



(그림 6) 트윗 수집 모듈



(그림 7) top-k 분석 모듈

마지막으로 야후 API를 이용하여 프로필의 위치와 트윗의 geocode를 주소 형태로 변환한 다음, 위치정보의 신뢰성 분석을 수행하였다. 계정별로 작성한 트윗을 정렬하여 프로필에 남긴 주소와 각 트윗에 남겨진 주소가 어느 정도 일치하는지를 판단해 프로필의 위치정보가 얼마나 신뢰성을 갖는지를 평가한다. 예를 들어 어떤 사용자가 프로필에는 경기도 부천시라고 위치를 작성하고, 사용자가 남긴 많은 트윗들 중 부천시에서 남긴 트윗이 가장 많다면 그 사용자는 top-k중 top-1에 해당한다. 그림 7에서 보이는 것처럼 사용자가 top-1부터 top-5, top-others,

top-none 중 어디에 속하는지를 계정과 트윗의 위치정보를 통하여 분석한다. top-none은 프로필에 입력한 위치에서 한 번도 트윗을 작성하지 않은 사용자를 말하며 top-others는 top-6 이후를 통틀어 나타낸다. 결론적으로, top-k에서 k의 숫자가 1이면 사용자는 처음 등록한 프로필 상의 위치에서 트윗을 가장 많이 작성하는 사용자로 이해할 수 있다.

3.3 프로그램 실행화면

이번 절에서는 DiSAnT의 실행모습을 보여준다. 각 모듈을 하나로 통합하여 그림 8과 같이 프로그램을 구현하였다. 각 기능의 결과를 텍스트 파일로 저장할 수 있으며, 그림 8(a)는 기준 계정을 설정한 후 계정을 수집하는 기능을 나타내는 화면이고, 그림 8(b)는 수집한 계정을 이용하여 해당 계정의 사용자가 작성한 트윗을 수집해 저장하는 실행모습이다. 마지막 그림 8(c)는 프로필 위치정보의 신뢰성을 분석하는 화면이다.

(a)

(b)

(c)

(그림 8) DiSAnT 실행화면

4. 결론 및 향후 연구 계획

본 논문에서는 트위터에서 발생하는 빅 데이터를 효율적으로 수집하고 분석할 수 있는 분산처리 기반의 DiSAnT 시스템을 소개하였다. 트위터와 같은 소셜 네트워크 서비스는 사용자가 증가하고 데이터가 스트림 형태로 생성되기 때문에 많은 트위터 기반의 연구들은 빅 데이터를 효율적으로 처리해야한다는 과제를 안고 있다. 트위터의 데이터를 처리하는데 가장 큰 어려움은 트위터 측에서 요청을 제한한다는 점에 있다. DiSAnT는 3가지 기능으로 세분화되어 처리되며, 계정 수집과 트윗 수집의 과정에서는 다수의 클라이언트 PC가 개념적인 두 그룹으로 나뉘어져 데이터를 수집하고 분석을 수행한다. 따라서 대용량이라는 특성과 요청에 제한이 있다는 특성을 가진 트위터 데이터의 처리를 용이하게 한다.

향후 연구 계획은 현재 계정과 트윗 수집이라는 기능을 가진 DiSAnT에 데이터의 위치를 분석하는 모듈을 추가한 것과 같이 다양한 분석 모듈을 추가하여 분석기능을 강화하는 것이다. 또한 최근 트위터 연구에서 주목받고 있는 오피니언 마이닝(Opinion Mining)을 적용하여, 수집한 트윗을 분석해 사용자의 감정 상태나 의견의 긍정/부정적 성격을 판단하는 기능을 추가한다면 플랫폼의 의미가 한층 더 높아질 것이다.

참고문헌

[1] Wikipedia, “Distributed Computing,” http://en.wikipedia.org/wiki/Distributed_computing, 2012.

[2] Apache Software Foundation, “Hadoop HDFS,” http://hadoop.apache.org/hdfs/, 2011.

[3] J. Dean and S. Ghemawat, “MapReduce: Simplified Data Processing on Large Clusters,” Communication of the ACM, Vol. 51, No. 1, Jan. 2008.

[4] S. Ghemawat, H. Gobioff, and S. Leung, “The Google File System,” In Proceedings of the 19th ACM Symposium on Operating Systems Principles, 2003.

[5] M. F. Husain, P. Doshi, L. Khan, and B. Thuraisingham, “Storage and Retrieval of Large RDF Graph Using Hadoop and MapReduce,“ In Proceedings of the 1st International Conference on Cloud Computing, pp. 680-688, Dec. 2009.

[6] A. Thusoo, J. S. Sarma, N. Jain, Z. Shao, P. Chakka, N. Zhang, S. Antony, H. Liu, and R. Murthy, “Hive - A Petabyte Scale Data Warehouse Using Hadoop,” In Proceedings of the 26th International Conference on Data Engineering, pp. 996-1005, 2010.

[7] C. Peri, The Twitter API, Pearson Education Inc., 2011.

[8] M. A. Russell, Mining the Social Web, O’Reilly, 2011.